云数据库 GaussDB

参考（分布式_V2.0-10.1.0）

文档版本 01

发布日期 2026-10-21

华为云计算技术有限公司

版权所有 © 华为云计算技术有限公司 2026。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 SQL 参考 1
1.1 GaussDB SQL 1
1.2 关键字 2
1.3 数据类型 36
1.3.1 数值类型 36
1.3.2 货币类型 47
1.3.3 布尔类型 48
1.3.4 字符类型 49
1.3.5 二进制类型 53
1.3.6 日期/时间类型 56
1.3.7 几何类型 79
1.3.8 网络地址类型 82
1.3.9 位串类型 83
1.3.10 UUID 类型 84
1.3.11 文本检索类型 85
1.3.12 JSON/JSONB 类型 88
1.3.13 HLL 数据类型 92
1.3.14 范围类型 96
1.3.15 对象标识符类型 99
1.3.16 伪类型 101
1.3.17 账本数据库使用的数据类型 102
1.3.18 XML 类型 103
1.3.19 XMLTYPE 类型 105
1.3.20 aclitem 类型 106
1.3.21 数组类型 107
1.3.22 向量化引擎支持的数据类型 112
1.3.23 ROWID 类型 117
1.3.24 向量数据类型 124
1.4 字符集和字符序 125
1.4.1 客户端连接的字符集和字符序 126
1.4.2 数据库级字符集和字符序 127
1.4.3 模式级字符集和字符序 127
1.4.4 表级字符集和字符序 128

 版权所有 © 华为云计算技术有限公司

1.4.5 列级字符集和字符序 130
1.4.6 字符类型表达式的字符集和字符序 130
1.4.7 字符集和字符序合并规则 131
1.5 常量和宏 133
1.6 函数和操作符 134
1.6.1 逻辑操作符 137
1.6.2 比较操作符 137
1.6.3 字符处理函数和操作符 138
1.6.4 二进制字符串函数和操作符 197
1.6.5 位串函数和操作符 200
1.6.6 模式匹配操作符 202
1.6.7 数字操作函数和操作符 209
1.6.8 时间和日期处理函数和操作符 228
1.6.9 类型转换函数 298
1.6.10 几何函数和操作符 330
1.6.11 网络地址函数和操作符 340
1.6.12 文本检索函数和操作符 345
1.6.13 JSON/JSONB 函数和操作符 352
1.6.14 HLL 函数和操作符 379
1.6.15 SEQUENCE 函数 391
1.6.16 数组函数和操作符 394
1.6.17 范围函数和操作符 403
1.6.18 聚集函数 439
1.6.19 窗口函数 461
1.6.20 安全函数 474
1.6.21 账本数据库的函数 486
1.6.22 密态函数和操作符 488
1.6.23 返回集合的函数 498
1.6.24 重载查询函数 500
1.6.25 条件表达式函数 504
1.6.26 系统信息函数 510
1.6.27 系统管理函数 553
1.6.27.1 配置设置函数 554
1.6.27.2 通用文件访问函数 554
1.6.27.3 服务器信号函数 556
1.6.27.4 备份恢复控制函数 559
1.6.27.5 双集群容灾控制函数 568
1.6.27.6 双集群容灾查询函数 574
1.6.27.7 快照同步函数 578
1.6.27.8 数据库对象函数 578
1.6.27.9 咨询锁函数 585
1.6.27.10 逻辑复制函数 588

 版权所有 © 华为云计算技术有限公司

1.6.27.11 段页式存储函数 627
1.6.27.12 bucket 系统函数 648
1.6.27.13 Undo 系统函数 658
1.6.27.14 HTAP 系统函数 676
1.6.27.15 其它函数 703
1.6.28 SPM 计划管理函数 765
1.6.29 统计信息函数 768
1.6.30 触发器函数 872
1.6.31 HashFunc 函数 874
1.6.32 提示信息函数 881
1.6.33 故障注入系统函数 882
1.6.34 重分布函数 882
1.6.35 分布列推荐函数 883
1.6.36 内部函数 887
1.6.37 敏感数据发现函数 898
1.6.38 动态数据脱敏函数 899
1.6.39 层次递归查询函数 900
1.6.40 hotkey 特性函数 901
1.6.41 Global SysCache 特性函数 901
1.6.42 数据损坏检测修复函数 903
1.6.43 Multixact 回收函数 924
1.6.44 XML 类型函数 924
1.6.45 XMLTYPE 类型函数 942
1.6.46 数据透视函数 953
1.6.47 Global Plsql Cache 特性函数 954
1.6.48 通用标识符函数 959
1.6.49 SQL 限流函数 960
1.6.50 SQL 工具函数 966
1.6.51 多租数据库函数 975
1.6.52 全局临时表函数 975
1.6.53 DATABASE LINK 函数 979
1.6.54 GSMonitor 系统资源统计函数 983
1.6.55 排序计算函数 988
1.6.56 自治事务特性函数 994
1.6.57 DSL 规则管理函数 998
1.6.58 ROWID 类型函数 1001
1.6.59 失效重编译函数 1002
1.6.60 其他系统函数 1004
1.6.61 废弃函数 1072
1.6.62 向量数据函数与操作符 1074
1.6.62.1 向量距离计算接口 1074
1.6.62.2 向量操作函数接口 1075

 版权所有 © 华为云计算技术有限公司

1.6.62.3 向量函数和操作符 1080
1.6.62.4 相似文档排序召回检索函数和操作符 1090
1.7 表达式 1092
1.7.1 简单表达式 1092
1.7.2 条件表达式 1094
1.7.3 子查询表达式 1099
1.7.4 数组表达式 1101
1.7.5 行表达式 1103
1.7.6 时间间隔表达式 1104
1.8 伪列 1106
1.9 类型转换 1110
1.9.1 概述 1110
1.9.2 操作符 1111
1.9.3 函数 1113
1.9.4 值存储 1115
1.9.5 UNION，CASE 和相关构造 1116
1.10 系统操作 1122
1.11 事务控制 1123
1.12 全文检索 1123
1.12.1 介绍 1123
1.12.1.1 全文检索概述 1123
1.12.1.2 文档概念 1124
1.12.1.3 基本文本匹配 1125
1.12.1.4 配置 1125
1.12.2 表和索引 1126
1.12.2.1 搜索表 1126
1.12.2.2 创建索引 1127
1.12.3 控制文本搜索 1129
1.12.3.1 解析文档 1129
1.12.3.2 解析查询 1130
1.12.3.3 排序查询结果 1131
1.12.3.4 高亮搜索结果 1133
1.12.4 附加功能 1134
1.12.4.1 处理 tsvector 1134
1.12.4.2 处理查询 1135
1.12.4.3 查询重写 1135
1.12.4.4 收集文献统计 1136
1.12.5 解析器 1137
1.12.6 词典 1140
1.12.6.1 词典概述 1140
1.12.6.2 停用词 1141
1.12.6.3 Simple 词典 1141

 版权所有 © 华为云计算技术有限公司

1.12.6.4 Snowball 词典 1142
1.12.7 配置示例 1142
1.12.8 测试和调试文本搜索 1143
1.12.8.1 分词器测试 1143
1.12.8.2 解析器测试 1144
1.12.8.3 词典测试 1145
1.12.9 限制约束 1145
1.13 SQL 语法 1146
1.13.1 SQL 语法格式说明 1146
1.13.2 DCL 语法一览表 1146
1.13.3 DDL 语法一览表 1147
1.13.4 DML 语法一览表 1160
1.13.5 其他语法一览表 1162
1.13.6 Online DDL 及其操作 1164
1.13.7 A 1166
1.13.7.1 ABORT 1167
1.13.7.2 ALTER AGGREGATE 1168
1.13.7.3 ALTER ASYNC ENCRYPTION KEY ROTATION 1169
1.13.7.4 ALTER AUDIT POLICY 1170
1.13.7.5 ALTER COLUMN ENCRYPTION KEY 1174
1.13.7.6 ALTER COORDINATOR 1174
1.13.7.7 ALTER DATABASE 1176
1.13.7.8 ALTER DATABASE LINK 1181
1.13.7.9 ALTER DEFAULT PRIVILEGES 1183
1.13.7.10 ALTER DIRECTORY 1185
1.13.7.11 ALTER FOREIGN DATA WRAPPER 1186
1.13.7.12 ALTER FOREIGN TABLE 1187
1.13.7.13 ALTER FUNCTION 1189
1.13.7.14 ALTER GLOBAL CONFIGURATION 1194
1.13.7.15 ALTER GROUP 1196
1.13.7.16 ALTER INDEX 1197
1.13.7.17 ALTER LANGUAGE 1204
1.13.7.18 ALTER MASKING POLICY 1204
1.13.7.19 ALTER MATERIALIZED VIEW 1208
1.13.7.20 ALTER NODE 1210
1.13.7.21 ALTER NODE GROUP 1214
1.13.7.22 ALTER PACKAGE 1216
1.13.7.23 ALTER PROCEDURE 1217
1.13.7.24 ALTER RESOURCE LABEL 1221
1.13.7.25 ALTER RESOURCE POOL 1223
1.13.7.26 ALTER ROLE 1225
1.13.7.27 ALTER ROW LEVEL SECURITY POLICY 1229

 版权所有 © 华为云计算技术有限公司

1.13.7.28 ALTER SCHEMA 1231
1.13.7.29 ALTER SEQUENCE 1235
1.13.7.30 ALTER SERVER 1237
1.13.7.31 ALTER SESSION 1239
1.13.7.32 ALTER SYNONYM 1242
1.13.7.33 ALTER SYSTEM KILL SESSION 1244
1.13.7.34 ALTER TABLE 1245
1.13.7.35 ALTER TABLE PARTITION 1275
1.13.7.36 ALTER TABLE SUBPARTITION 1291
1.13.7.37 ALTER TABLESPACE 1301
1.13.7.38 ALTER TEXT SEARCH CONFIGURATION 1304
1.13.7.39 ALTER TEXT SEARCH DICTIONARY 1308
1.13.7.40 ALTER TRIGGER 1310
1.13.7.41 ALTER TYPE 1311
1.13.7.42 ALTER USER 1316
1.13.7.43 ALTER VIEW 1319
1.13.7.44 ANALYZE | ANALYSE 1323
1.13.7.45 AUTOHINT 1330
1.13.7.46 AUTOHINT DROP MODEL 1332
1.13.7.47 AUTOHINT PURGE 1333
1.13.8 B 1333
1.13.8.1 BEGIN 1333
1.13.9 C 1335
1.13.9.1 CALL 1335
1.13.9.2 CHECKPOINT 1338
1.13.9.3 CLEAN CONNECTION 1338
1.13.9.4 CLOSE 1340
1.13.9.5 CLUSTER 1341
1.13.9.6 COMMENT 1344
1.13.9.7 COMMIT | END 1347
1.13.9.8 COMMIT PREPARED 1348
1.13.9.9 COPY 1349
1.13.9.10 CREATE AGGREGATE 1369
1.13.9.11 CREATE AUDIT POLICY 1372
1.13.9.12 CREATE BARRIER 1376
1.13.9.13 CREATE CLIENT MASTER KEY 1376
1.13.9.14 CREATE COLUMN ENCRYPTION KEY 1379
1.13.9.15 CREATE CONVERSION 1380
1.13.9.16 CREATE DATABASE 1381
1.13.9.17 CREATE DATABASE LINK 1392
1.13.9.18 CREATE DIRECTORY 1394
1.13.9.19 CREATE EXTENSION 1395

 版权所有 © 华为云计算技术有限公司

1.13.9.20 CREATE FOREIGN DATA WRAPPER....1397
1.13.9.21 CREATE FOREIGN TABLE....1398
1.13.9.22 CREATE FUNCTION....1410
1.13.9.23 CREATE GLOBAL INDEX....1420
1.13.9.24 CREATE GROUP....1427
1.13.9.25 CREATE INCREMENTAL MATERIALIZED VIEW....1428
1.13.9.26 CREATE INDEX....1430
1.13.9.27 CREATE LANGUAGE....1449
1.13.9.28 CREATE MASKING POLICY....1449
1.13.9.29 CREATE MATERIALIZED VIEW....1453
1.13.9.30 CREATE MODEL....1455
1.13.9.31 CREATE NODE....1456
1.13.9.32 CREATE NODE GROUP....1460
1.13.9.33 CREATE PACKAGE....1461
1.13.9.34 CREATE PROCEDURE....1464
1.13.9.35 CREATE RESOURCE LABEL....1470
1.13.9.36 CREATE RESOURCE POOL....1472
1.13.9.37 CREATE ROLE....1475
1.13.9.38 CREATE ROW LEVEL SECURITY POLICY....1483
1.13.9.39 CREATE SCHEMA....1487
1.13.9.40 CREATE SECURITY LABEL....1490
1.13.9.41 CREATE SEQUENCE....1492
1.13.9.42 CREATE SERVER....1495
1.13.9.43 CREATE SYNONYM....1497
1.13.9.44 CREATE TABLE....1500
1.13.9.45 CREATE TABLESPACE....1531
1.13.9.46 CREATE TABLE AS....1534
1.13.9.47 CREATE TABLE PARTITION....1541
1.13.9.48 CREATE TABLE PARTITION | SUBPARTITION AS....1557
1.13.9.49 CREATE TABLE SUBPARTITION....1564
1.13.9.50 CREATE TEXT SEARCH CONFIGURATION....1577
1.13.9.51 CREATE TEXT SEARCH DICTIONARY....1578
1.13.9.52 CREATE TRIGGER....1584
1.13.9.53 CREATE TYPE....1590
1.13.9.54 CREATE USER....1598
1.13.9.55 CREATE VIEW....1603
1.13.9.56 CREATE WEAK PASSWORD DICTIONARY....1610
1.13.9.57 CURSOR....1612
1.13.10 D....1614
1.13.10.1 DEALLOCATE....1614
1.13.10.2 DECLARE....1615
1.13.10.3 DELETE....1618

 版权所有 © 华为云计算技术有限公司

1.13.10.4 DO 1624
1.13.10.5 DROP AGGREGATE 1625
1.13.10.6 DROP AUDIT POLICY 1626
1.13.10.7 DROP CLIENT MASTER KEY 1627
1.13.10.8 DROP COLUMN ENCRYPTION KEY 1628
1.13.10.9 DROP DATABASE 1629
1.13.10.10 DROP DATABASE LINK 1630
1.13.10.11 DROP DIRECTORY 1631
1.13.10.12 DROP EXTENSION 1632
1.13.10.13 DROP FOREIGN DATA WRAPPER 1633
1.13.10.14 DROP FOREIGN TABLE 1634
1.13.10.15 DROP FUNCTION 1635
1.13.10.16 DROP GLOBAL CONFIGURATION 1636
1.13.10.17 DROP GROUP 1637
1.13.10.18 DROP INDEX 1638
1.13.10.19 DROP LANGUAGE 1640
1.13.10.20 DROP MASKING POLICY 1640
1.13.10.21 DROP MATERIALIZED VIEW 1641
1.13.10.22 DROP MODEL 1642
1.13.10.23 DROP NODE 1643
1.13.10.24 DROP NODE GROUP 1644
1.13.10.25 DROP OWNED 1645
1.13.10.26 DROP PACKAGE 1646
1.13.10.27 DROP PROCEDURE 1647
1.13.10.28 DROP RESOURCE LABEL 1647
1.13.10.29 DROP RESOURCE POOL 1649
1.13.10.30 DROP ROLE 1650
1.13.10.31 DROP ROW LEVEL SECURITY POLICY 1650
1.13.10.32 DROP SCHEMA 1651
1.13.10.33 DROP SECURITY LABEL 1653
1.13.10.34 DROP SEQUENCE 1653
1.13.10.35 DROP SERVER 1654
1.13.10.36 DROP SYNONYM 1655
1.13.10.37 DROP TABLE 1656
1.13.10.38 DROP TABLESPACE 1657
1.13.10.39 DROP TEXT SEARCH CONFIGURATION 1658
1.13.10.40 DROP TEXT SEARCH DICTIONARY 1659
1.13.10.41 DROP TRIGGER 1660
1.13.10.42 DROP TYPE 1663
1.13.10.43 DROP USER 1664
1.13.10.44 DROP VIEW 1666
1.13.10.45 DROP WEAK PASSWORD DICTIONARY 1667

 版权所有 © 华为云计算技术有限公司

1.13.11 E..... 1667
1.13.11.1 EXECUTE..... 1668
1.13.11.2 EXECUTE DIRECT..... 1669
1.13.11.3 EXPDP DATABASE..... 1671
1.13.11.4 EXPDP TABLE..... 1671
1.13.11.5 EXPLAIN..... 1672
1.13.11.6 EXPLAIN PLAN..... 1681
1.13.11.7 EXPLAIN AUTOHINT..... 1683
1.13.12 F..... 1684
1.13.12.1 FETCH..... 1684
1.13.13 G..... 1687
1.13.13.1 GENERATED UPDATE SYSTEM OBJECT..... 1687
1.13.13.2 GRANT..... 1688
1.13.14 I..... 1699
1.13.14.1 IMPDP DATABASE CREATE..... 1700
1.13.14.2 IMPDP RECOVER..... 1700
1.13.14.3 IMPDP TABLE..... 1701
1.13.14.4 IMPDP TABLE PREPARE..... 1701
1.13.14.5 INSERT..... 1702
1.13.14.6 INSERT ALL..... 1718
1.13.15 L..... 1726
1.13.15.1 LOAD DATA..... 1726
1.13.15.2 LOCK..... 1733
1.13.15.3 LOCK BUCKETS..... 1737
1.13.16 M..... 1738
1.13.16.1 MARK BUCKETS..... 1738
1.13.16.2 MERGE INTO..... 1739
1.13.16.3 MOVE..... 1741
1.13.17 P..... 1743
1.13.17.1 PREDICT BY..... 1743
1.13.17.2 PREPARE..... 1743
1.13.17.3 PREPARE TRANSACTION..... 1744
1.13.17.4 PURGE..... 1745
1.13.18 R..... 1747
1.13.18.1 REASSIGN OWNED..... 1747
1.13.18.2 REFRESH INCREMENTAL MATERIALIZED VIEW..... 1748
1.13.18.3 REFRESH MATERIALIZED VIEW..... 1749
1.13.18.4 REFRESH SYSTEM OBJECT..... 1750
1.13.18.5 REINDEX..... 1751
1.13.18.6 RELEASE SAVEPOINT..... 1755
1.13.18.7 RENAME TABLE..... 1756
1.13.18.8 REPLACE..... 1757

 版权所有 © 华为云计算技术有限公司

1.13.18.9 RESET 1761
1.13.18.10 REVOKE 1763
1.13.18.11 ROLLBACK 1767
1.13.18.12 ROLLBACK PREPARED 1768
1.13.18.13 ROLLBACK TO SAVEPOINT 1769
1.13.19 S 1771
1.13.19.1 SAVEPOINT 1771
1.13.19.2 SECURITY LABEL ON 1772
1.13.19.3 SELECT 1773
1.13.19.3.1 简单查询 1797
1.13.19.3.2 条件查询 1798
1.13.19.3.3 分组查询 1799
1.13.19.3.4 分页查询 1801
1.13.19.3.5 分区查询 1802
1.13.19.3.6 连接查询 1802
1.13.19.3.7 子查询 1804
1.13.19.3.8 层次查询 1806
1.13.19.3.9 复合查询 1808
1.13.19.3.10 行转列与列转行 1809
1.13.19.3.11 窗口函数查询 1809
1.13.19.3.12 简化版查询 1810
1.13.19.3.13 Hint 查询 1810
1.13.19.4 SELECT INTO 1811
1.13.19.5 SET 1813
1.13.19.6 SET CONSTRAINTS 1816
1.13.19.7 SET ROLE 1817
1.13.19.8 SET SESSION AUTHORIZATION 1818
1.13.19.9 SET TRANSACTION 1820
1.13.19.10 SHOW 1821
1.13.19.11 SHUTDOWN 1823
1.13.19.12 START TRANSACTION 1824
1.13.20 T 1826
1.13.20.1 TIMECAPSULE TABLE 1826
1.13.20.2 TRUNCATE 1830
1.13.21 U 1833
1.13.21.1 UPDATE 1833
1.13.21.1.1 伪更新分布列 1841
1.13.22 V 1846
1.13.22.1 VACUUM 1846
1.13.22.2 VALUES 1851
1.14 附录 1853
1.14.1 扩展函数 1853

 版权所有 © 华为云计算技术有限公司

1.14.2 美元引用的字符串常量 1853
1.14.3 行表达式函数白名单 1854
2 存储过程 1920
2.1 存储过程 1920
2.2 数据类型 1920
2.3 数据类型转换 1920
2.4 数组、集合和 record 1922
2.4.1 数组 1922
2.4.1.1 数组类型的使用 1922
2.4.1.2 数组支持的函数 1924
2.4.2 集合 1930
2.4.2.1 集合类型的使用 1930
2.4.2.2 集合支持的函数 1935
2.4.3 record 1948
2.5 声明语法 1953
2.5.1 基本结构 1953
2.5.2 匿名块 1953
2.5.3 子程序 1955
2.5.3.1 独立的子程序 1956
2.5.3.2 包内的子程序 1956
2.5.3.3 嵌套的子程序 1956
2.6 基本语句 1959
2.6.1 定义变量 1959
2.6.2 赋值语句 1961
2.6.3 调用语句 1963
2.7 动态语句 1965
2.7.1 执行动态查询语句 1965
2.7.2 执行动态非查询语句 1968
2.7.3 动态调用存储过程 1969
2.7.4 动态调用匿名块 1971
2.8 控制语句 1974
2.8.1 返回语句 1974
2.8.1.1 RETURN 1974
2.8.1.2 RETURN NEXT 及 RETURN QUERY 1974
2.8.2 条件语句 1976
2.8.3 循环语句 1978
2.8.4 分支语句 1983
2.8.5 空语句 1984
2.8.6 错误捕获语句 1985
2.8.6.1 获取有关错误的信息 1988
2.8.6.2 获取当前的执行信息 1989
2.8.7 GOTO 语句 1989

 版权所有 © 华为云计算技术有限公司

2.9 事务语句 1991
2.10 其他语句 2000
2.10.1 锁操作 2000
2.10.2 游标操作 2000
2.11 游标 2001
2.11.1 游标概述 2001
2.11.2 显式游标 2001
2.11.3 隐式游标 2006
2.11.4 游标循环 2007
2.12 高级包 2008
2.12.1 基础接口 2008
2.12.1.1 PKG_SERVICE 2008
2.12.1.2 PKG_UTIL 2018
2.12.1.3 DBE_XML 2057
2.12.2 二次封装接口 2079
2.12.2.1 DBE_APPLICATION_INFO 2079
2.12.2.2 DBE_COMPRESSION 2082
2.12.2.3 DBE_DESCRIBE 2087
2.12.2.4 DBE_FILE 2093
2.12.2.5 DBE_HEAT_MAP 2114
2.12.2.6 DBE_ILM 2116
2.12.2.7 DBE_ILM_ADMIN 2119
2.12.2.8 DBE_LICENSE 2124
2.12.2.9 DBE_LOB 2127
2.12.2.10 DBE_MATCH 2152
2.12.2.11 DBE_OBFUSCATION_TOOLKIT 2153
2.12.2.12 DBE_OUTPUT 2161
2.12.2.13 DBE_RANDOM 2166
2.12.2.14 DBE_RAW 2167
2.12.2.15 DBE_SCHEDULER 2181
2.12.2.16 DBE_SESSION 2213
2.12.2.17 DBE_SQL 2216
2.12.2.18 DBE_STATS 2254
2.12.2.19 DBE_TASK 2331
2.12.2.20 DBE_UTILITY 2341
2.12.2.21 DBE_XMLDOM 2355
2.12.2.22 DBE_XMLGEN 2405
2.12.2.23 DBE_XMLPARSER 2413
2.12.2.24 PRVT_ILM 2420
2.12.3 内部函数 2422
2.13 Retry 管理 2424
2.14 PACKAGE 2424

 版权所有 © 华为云计算技术有限公司

2.15 调试 2425
2.16 关键字 2429
3 兼容性说明 2435
3.1 GaussDB 数据库兼容性概述 2435
3.2 Oracle 兼容性说明 2435
3.2.1 Oracle 数据库兼容性概述 2435
3.2.2 SQL 的基本元素 2435
3.2.2.1 数据类型 2436
3.2.2.2 数据类型比较规则 2443
3.2.2.3 字面量 2444
3.2.2.4 格式模型 2444
3.2.2.5 空值 2445
3.2.2.6 注释 2445
3.2.2.7 数据库对象 2445
3.2.2.8 数据库对象名称和限定符 2447
3.2.2.9 SQL 语句中的引用架构对象和部件的语法 2447
3.2.3 伪列 2448
3.2.4 操作符 2449
3.2.5 表达式 2452
3.2.6 条件 2456
3.2.7 驱动 2457
3.2.7.1 JDBC 2457
3.2.7.1.1 Array 2457
3.2.7.1.2 Struct 2470
3.2.8 常见的 SQL DDL 子句 2480
3.2.9 SQL 查询和子查询 2488
3.2.10 PL/SQL 语言 2492
3.2.10.1 PL/SQL 基本语法 2492
3.2.10.2 数据类型兼容性 2495
3.2.10.3 控制语句 2495
3.2.10.4 集合和 Record 2497
3.2.10.5 静态 SQL 2501
3.2.10.6 动态 SQL 2505
3.2.10.7 Trigger 2505
3.2.11 系统函数 2513
3.2.11.1 单行函数 2513
3.2.11.2 其它函数 2548
3.2.12 系统视图 2550
3.2.13 高级包 2556
3.3 MySQL 兼容性说明 2602
3.3.1 MySQL 数据库兼容性概述 2602
3.3.2 MySQL 兼容性 M-Compatibility 模式 2604

 版权所有 © 华为云计算技术有限公司

3.3.2.1 数据类型 2604
3.3.2.1.1 数值数据类型 2604
3.3.2.1.2 日期与时间数据类型 2606
3.3.2.1.3 字符串数据类型 2608
3.3.2.1.4 二进制数据类型 2610
3.3.2.1.5 数据类型支持的属性 2613
3.3.2.1.6 数据类型转换 2614
3.3.2.2 系统函数 2620
3.3.2.2.1 系统函数兼容性概述 2620
3.3.2.2.2 流程控制函数 2622
3.3.2.2.3 日期和时间函数 2622
3.3.2.2.4 字符串函数 2628
3.3.2.2.5 强制转换函数 2632
3.3.2.2.6 加密函数 2633
3.3.2.2.7 比较函数 2634
3.3.2.2.8 聚合函数 2635
3.3.2.2.9 数字操作函数 2637
3.3.2.2.10 其他函数 2638
3.3.2.3 操作符 2638
3.3.2.4 字符集 2648
3.3.2.5 排序规则 2649
3.3.2.6 事务 2650
3.3.2.7 SQL 2654
3.3.2.7.1 SQL 兼容性概述 2654
3.3.2.7.2 关键字 2656
3.3.2.7.3 标识符 2656
3.3.2.7.4 DDL 2658
3.3.2.7.5 DML 2675
3.3.2.7.6 DCL 2709
3.3.2.7.7 其他语句 2710
3.3.2.7.8 用户与权限 2711
3.3.2.7.9 系统表和系统视图 2717
3.3.3 MySQL 兼容性 MYSQL 模式 2720
3.3.3.1 数据类型 2720
3.3.3.1.1 数值数据类型 2720
3.3.3.1.2 日期与时间数据类型 2727
3.3.3.1.3 字符串数据类型 2738
3.3.3.1.4 二进制数据类型 2741
3.3.3.1.5 JSON 数据类型 2743
3.3.3.1.6 数据类型支持的属性 2744
3.3.3.1.7 数据类型转换 2744
3.3.3.2 系统函数 2747

 版权所有 © 华为云计算技术有限公司

3.3.3.2.1 系统函数兼容性概述 2747
3.3.3.2.2 流量控制函数 2747
3.3.3.2.3 日期和时间函数 2750
3.3.3.2.4 字符串函数 2760
3.3.3.2.5 强制转换函数 2764
3.3.3.2.6 加密函数 2764
3.3.3.2.7 JSON 函数 2765
3.3.3.2.8 聚合函数 2767
3.3.3.2.9 数字操作函数 2768
3.3.3.2.10 其他函数 2769
3.3.3.3 操作符 2769
3.3.3.4 字符集 2770
3.3.3.5 排序规则 2771
3.3.3.6 SQL 2771
3.3.3.6.1 DDL 2772
3.3.3.6.2 DML 2780
3.3.3.6.3 DCL 2793
3.3.3.7 驱动 2793
3.3.3.7.1 JDBC 接口参考 2794
4 工具参考 2795
4.1 下载并安装工具 2795
4.2 数据库连接工具 2796
4.2.1 gsql 连接数据库 2796
4.2.1.1 gsql 使用指导 2796
4.2.1.2 元命令参考 2813
4.2.1.3 FAQ 2841
4.3 数据导入导出工具 2846
4.3.1 数据导入导出工具概览 2846
4.3.2 gs_dump 导出数据库信息 2851
4.3.3 gs_dumpall 导出所有数据库信息 2867
4.3.4 gs_restore 导入数据 2874
5 日志参考 2882
5.1 日志类型简介 2882
5.2 系统日志 2884
5.3 操作日志 2892
5.4 审计日志 2897
5.5 WAL 日志 2899
5.6 内存日志 2902
6 数据库运行参数说明 2910
6.1 查看参数 2910
6.2 设置参数 2911

 版权所有 © 华为云计算技术有限公司

6.3 GUC 参数说明 2915
6.3.1 GUC 使用说明 2915
6.3.2 文件位置 2939
6.3.3 连接和认证 2941
6.3.3.1 连接设置 2941
6.3.3.2 安全和认证 2952
6.3.3.3 通信库参数 2968
6.3.4 资源消耗 2978
6.3.4.1 内存 2978
6.3.4.2 磁盘空间 3002
6.3.4.3 内核资源使用 3003
6.3.4.4 基于开销的清理延迟 3004
6.3.4.5 后端写线程 3007
6.3.4.6 异步 I/O 3013
6.3.5 数据并行导入导出 3015
6.3.6 预写式日志 3022
6.3.6.1 设置 3022
6.3.6.2 检查点 3033
6.3.6.3 日志回放 3038
6.3.6.4 归档 3048
6.3.7 双机复制 3048
6.3.7.1 发送端服务器 3049
6.3.7.2 主服务器 3066
6.3.7.3 备服务器 3077
6.3.7.4 逻辑复制 3088
6.3.8 查询规划 3092
6.3.8.1 优化器方法配置 3092
6.3.8.2 优化器开销常量 3119
6.3.8.3 基因查询优化器 3122
6.3.8.4 基于 DSL 的规则引擎 3125
6.3.8.5 其他优化器选项 3126
6.3.9 错误报告和日志 3182
6.3.9.1 记录日志的位置 3182
6.3.9.2 记录日志的时间 3187
6.3.9.3 记录日志的内容 3190
6.3.9.4 使用 CSV 格式写日志 3203
6.3.10 告警上报 3205
6.3.11 运行时统计 3207
6.3.11.1 查询和索引统计收集器 3207
6.3.11.2 热点 key 统计 3213
6.3.12 负载管理 3213
6.3.13 自动清理 3235

 版权所有 © 华为云计算技术有限公司

6.3.14 客户端连接缺省设置 3246
6.3.14.1 语句行为 3246
6.3.14.2 区域和格式化 3254
6.3.14.3 其他缺省 3265
6.3.15 锁管理 3268
6.3.16 版本和平台兼容性 3276
6.3.16.1 历史版本兼容性 3276
6.3.16.2 平台和客户端兼容性 3283
6.3.16.3 云服务产品版本号 3406
6.3.17 容错性 3407
6.3.18 连接池参数 3409
6.3.19 集群事务 3415
6.3.20 双集群复制参数 3428
6.3.21 开发人员选项 3432
6.3.22 审计 3448
6.3.22.1 审计开关 3448
6.3.22.2 用户和权限审计 3452
6.3.22.3 操作审计 3456
6.3.23 事务监控 3464
6.3.24 GTM 相关参数 3465
6.3.25 升级参数 3481
6.3.26 其它选项 3482
6.3.27 等待事件 3497
6.3.28 Query 3498
6.3.29 系统性能快照 3510
6.3.30 黑匣子相关参数 3520
6.3.31 安全配置 3521
6.3.32 HyperLogLog 3531
6.3.33 用户自定义函数 3533
6.3.34 定时任务 3534
6.3.35 线程池 3535
6.3.36 全文检索 3543
6.3.37 备份恢复 3545
6.3.38 Undo 3547
6.3.39 DCF 参数设置 3549
6.3.40 闪回相关参数 3563
6.3.41 回滚相关参数 3564
6.3.42 AI 特性 3565
6.3.43 Global SysCache 参数 3577
6.3.44 分布式备机读参数 3578
6.3.45 备机数据修复 3580
6.3.46 账本数据库参数 3581

 版权所有 © 华为云计算技术有限公司

6.3.47 在线创建索引.....3582
6.3.48 在线 DDL.....3583
6.3.49 数据生命周期管理-OLTP 表压缩.....3583
6.3.50 session 级事务下推及相关参数.....3587
6.3.51 Enhanced Toast.....3589
6.3.52 全局临时表.....3589
6.3.53 单节点表.....3591
6.3.54 HTAP 混合负载.....3592
6.3.55 预留参数.....3599
6.3.56 向量数据库参数.....3601
6.4 CM 参数说明.....3606
6.4.1 cm_agent 参数.....3607
6.4.2 cm_server 参数.....3631
7 系统表和系统视图.....3661
7.1 系统表和系统视图概述.....3661
7.2 系统表.....3661
7.2.1 分区表.....3661
7.2.1.1 PG_PARTITION.....3662
7.2.1.2 GS_SUBPARTITION_TEMPLATE.....3664
7.2.2 负载管理.....3664
7.2.2.1 GS_ABNORMAL_SQL_INFO.....3664
7.2.2.2 GS_ABNORMAL_SQL_WHITE_LIST.....3665
7.2.2.3 PG_APP_WORKLOADGROUP_MAPPING.....3665
7.2.3 OLTP 表压缩.....3665
7.2.3.1 GS_ILM.....3666
7.2.3.2 GS_ILM_JOBDETAIL.....3666
7.2.3.3 GS_ILM_OBJECT.....3667
7.2.3.4 GS_ILM_PARAM.....3668
7.2.3.5 GS_ILM_POLICY.....3670
7.2.3.6 GS_ILM_TASK.....3671
7.2.3.7 GS_ILM_TASKDETAIL.....3671
7.2.3.8 GS_ILM_TICKER.....3672
7.2.3.9 GS_ILM_TURBO_OBJECT.....3672
7.2.4 密态等值查询.....3673
7.2.4.1 GS_CLIENT_GLOBAL_KEYS.....3673
7.2.4.2 GS_CLIENT_GLOBAL_KEYS_ARGS.....3674
7.2.4.3 GS_COLUMN_KEYS.....3674
7.2.4.4 GS_COLUMN_KEYS_ARGS.....3675
7.2.4.5 GS_ENCRYPTED_COLUMNS.....3675
7.2.4.6 GS_ENCRYPTED_PROC.....3676
7.2.5 通信.....3676
7.2.5.1 PGXC_NODE.....3676

 版权所有 © 华为云计算技术有限公司

7.2.6 账本数据库 3678
7.2.6.1 GS_GLOBAL_CHAIN 3678
7.2.7 SPM 计划管理 3679
7.2.7.1 GS_SPM_SQL 3679
7.2.7.2 GS_SPM_PARAM 3679
7.2.7.3 GS_SPM_BASELINE 3680
7.2.7.4 GS_SPM_EVOLUTION 3681
7.2.7.5 GS_SPM_ID_HASH_JOIN 3682
7.2.8 AI 3682
7.2.8.1 GS_MODEL_WAREHOUSE 3682
7.2.8.2 GS_OPT_MODEL 3683
7.2.8.3 GS_ABO_MODEL_STATISTIC 3683
7.2.8.4 GS_ABO_HISTORY 3684
7.2.9 审计 3685
7.2.9.1 GS_AUDITING_POLICY 3685
7.2.9.2 GS_AUDITING_POLICY_ACCESS 3686
7.2.9.3 GS_AUDITING_POLICY_FILTERS 3686
7.2.9.4 GS_AUDITING_POLICY_PRIVILEGES 3687
7.2.10 用户和权限管理 3687
7.2.10.1 GS_DB_PRIVILEGE 3687
7.2.10.2 PG_DB_ROLE_SETTING 3688
7.2.10.3 PG_DEFAULT_ACL 3688
7.2.10.4 PG_RLSPOLICY 3689
7.2.10.5 PG_SECLABEL 3689
7.2.10.6 PG_SHSECLABEL 3690
7.2.10.7 PG_USER_MAPPING 3690
7.2.10.8 PG_USER_STATUS 3691
7.2.11 连接和认证 3692
7.2.11.1 PG_AUTHID 3692
7.2.11.2 PG_AUTH_HISTORY 3694
7.2.11.3 PG_AUTH_MEMBERS 3694
7.2.12 动态脱敏 3695
7.2.12.1 GS_MASKING_POLICY 3695
7.2.12.2 GS_MASKING_POLICY_ACTIONS 3695
7.2.12.3 GS_MASKING_POLICY_FILTERS 3696
7.2.13 DATABASE LINK 3696
7.2.13.1 GS_DATABASE_LINK 3696
7.2.14 物化视图 3697
7.2.14.1 GS_MATVIEW 3697
7.2.14.2 GS_MATVIEW_DEPENDENCY 3698
7.2.15 逻辑解码 3698
7.2.15.1 GS_LOGICAL_ATTRDEF 3698

 版权所有 © 华为云计算技术有限公司

7.2.15.2 GS_LOGICAL_ATTRIBUTE 3699
7.2.15.3 GS_LOGICAL_AUTHID 3700
7.2.15.4 GS_LOGICAL_CLASS 3700
7.2.15.5 GS_LOGICAL_CONSTRAINT 3701
7.2.15.6 GS_LOGICAL_DATABASE 3704
7.2.15.7 GS_LOGICAL_DICTIONARY 3704
7.2.15.8 GS_LOGICAL_INDEX 3705
7.2.15.9 GS_LOGICAL_NAMESPACE 3706
7.2.15.10 GS_LOGICAL_PARTITION 3706
7.2.15.11 GS_LOGICAL_TYPE 3707
7.2.15.12 GS_SQLAPPLY_SKIP 3709
7.2.15.13 GS_TXN_LSN_TIME 3710
7.2.15.14 GS_LOGICAL_SET 3711
7.2.15.15 GS_LOGICAL_ENUM 3711
7.2.15.16 GS_SQLAPPLY_HISTORY 3712
7.2.16 基于 DSL 的规则引擎 3712
7.2.16.1 GS_DSL_REWRITERULE 3712
7.2.17 HTAP 3714
7.2.17.1 GS_IMCV 3714
7.2.18 其他系统表 3715
7.2.18.1 GS_ASP 3715
7.2.18.2 GS_COPY_ERROR_OLD 3717
7.2.18.3 GS_COPY_SUMMARY_LOG 3718
7.2.18.4 GS_COPY_ERROR_LOG 3719
7.2.18.5 GS_DEP_SOURCE 3720
7.2.18.6 GS_DEPENDENCIES 3721
7.2.18.7 GS_DEPENDENCIES_OBJ 3721
7.2.18.8 GS_GLOBAL_CONFIG 3722
7.2.18.9 GS_JOB_ATTRIBUTE 3722
7.2.18.10 GS_JOB_ARGUMENT 3723
7.2.18.11 GS_PACKAGE 3723
7.2.18.12 GS_PLAN_TRACE 3724
7.2.18.13 GS_POLICY_LABEL 3724
7.2.18.14 GS_RECYCLEBIN 3725
7.2.18.15 GS_SECURITY_LABEL 3726
7.2.18.16 GS_SQL_PATCH 3727
7.2.18.17 GS_STATISTIC_EXT_HISTORY 3728
7.2.18.18 GS_STATISTIC_HISTORY 3729
7.2.18.19 GS_SYSTEM_OBJECTS_VERSION 3731
7.2.18.20 GS_TABLESTATS_HISTORY 3732
7.2.18.21 GS_TS_DICT_DEFINITION 3732
7.2.18.22 GS_TXN_SNAPSHOT 3733

 版权所有 © 华为云计算技术有限公司

7.2.18.23 GS_UID....3733
7.2.18.24 GS_WORKLOAD_RULE....3733
7.2.18.25 PG_AGGREGATE....3734
7.2.18.26 PG_AM....3735
7.2.18.27 PG_AMOP....3738
7.2.18.28 PG_AMPROC....3739
7.2.18.29 PG_ATTRDEF....3740
7.2.18.30 PG_ATTRIBUTE....3740
7.2.18.31 PG_CAST....3742
7.2.18.32 PG_CLASS....3743
7.2.18.33 PG_COLLATION....3747
7.2.18.34 PG_CONSTRAINT....3748
7.2.18.35 PG_CONVERSION....3750
7.2.18.36 PG_DATABASE....3751
7.2.18.37 PG_DEPEND....3752
7.2.18.38 PG_DESCRIPTION....3753
7.2.18.39 PG_DIRECTORY....3754
7.2.18.40 PG_ENUM....3754
7.2.18.41 PG_EXTENSION....3755
7.2.18.42 PG_EXTENSION_DATA_SOURCE....3755
7.2.18.43 PG_FOREIGN_DATA_WRAPPER....3756
7.2.18.44 PG_FOREIGN_SERVER....3757
7.2.18.45 PG_FOREIGN_TABLE....3757
7.2.18.46 PG_HASHBUCKET....3757
7.2.18.47 PG_INDEX....3758
7.2.18.48 PG_INHERITS....3760
7.2.18.49 PG_JOB....3761
7.2.18.50 PG_JOB_PROC....3762
7.2.18.51 PG_LANGUAGE....3763
7.2.18.52 PG_NAMESPACE....3764
7.2.18.53 PG_OBJECT....3764
7.2.18.54 PG_OPCLASS....3766
7.2.18.55 PG_OPERATOR....3766
7.2.18.56 PG_OPFAMILY....3767
7.2.18.57 PG_PLTEMPLATE....3768
7.2.18.58 PG_PROC....3768
7.2.18.59 PG_RANGE....3772
7.2.18.60 PG_REPLICATION_ORIGIN....3772
7.2.18.61 PG_RESOURCE_POOL....3773
7.2.18.62 PG_REWRITE....3774
7.2.18.63 PG_SET....3775
7.2.18.64 PG_SHDEPEND....3775

版权所有 © 华为云计算技术有限公司

7.2.18.65 PG_SHDESCRIPTION....3776
7.2.18.66 PG_STATISTIC....3776
7.2.18.67 PG_STATISTIC_EXT....3778
7.2.18.68 PG_SYNONYM....3780
7.2.18.69 PG_TABLESPACE....3780
7.2.18.70 PG_TRIGGER....3781
7.2.18.71 PG_TS_CONFIG....3783
7.2.18.72 PG_TS_CONFIG_MAP....3783
7.2.18.73 PG_TS_DICT....3784
7.2.18.74 PG_TS_PARSER....3784
7.2.18.75 PG_TS_TEMPLATE....3785
7.2.18.76 PG_TYPE....3785
7.2.18.77 PGXC_CLASS....3789
7.2.18.78 PGXC_GROUP....3790
7.2.18.79 PGXC_SLICE....3791
7.2.18.80 PLAN_TABLE_DATA....3791
7.2.18.81 STATEMENT_HISTORY....3793
7.2.18.82 STREAMING_STREAM....3800
7.2.18.83 STREAMING_CONT_QUERY....3801
7.3 系统视图....3801
7.3.1 分区表....3802
7.3.1.1 ADM_IND_PARTITIONS....3802
7.3.1.2 ADM_IND_SUBPARTITIONS....3804
7.3.1.3 ADM_PART_COL_STATISTICS....3806
7.3.1.4 ADM_PART_INDEXES....3808
7.3.1.5 ADM_PART_TABLES....3809
7.3.1.6 ADM_SUBPART_KEY_COLUMNS....3811
7.3.1.7 ADM_TAB_PARTITIONS....3812
7.3.1.8 ADM_TAB_SUBPARTITIONS....3815
7.3.1.9 DB_IND_PARTITIONS....3815
7.3.1.10 DB_IND_SUBPARTITIONS....3818
7.3.1.11 DB_PART_COL_STATISTICS....3820
7.3.1.12 DB_PART_INDEXES....3821
7.3.1.13 DB_PART_KEY_COLUMNS....3822
7.3.1.14 DB_PART_TABLES....3823
7.3.1.15 DB_SUBPART_KEY_COLUMNS....3825
7.3.1.16 DB_TAB_PARTITIONS....3826
7.3.1.17 DB_TAB_SUBPARTITIONS....3828
7.3.1.18 MY_IND_PARTITIONS....3831
7.3.1.19 MY_IND_SUBPARTITIONS....3833
7.3.1.20 MY_PART_COL_STATISTICS....3835
7.3.1.21 MY_PART_INDEXES....3837

 版权所有 © 华为云计算技术有限公司

7.3.1.22 MY_PART_KEY_COLUMNS 3837
7.3.1.23 MY_PART_TABLES 3838
7.3.1.24 MY_SUBPART_KEY_COLUMNS 3840
7.3.1.25 MY_TAB_PARTITIONS 3841
7.3.1.26 MY_TAB_SUBPARTITIONS 3843
7.3.1.27 GS_STATIO_ALL_PARTITIONS 3846
7.3.1.28 GS_STAT_XACT_ALL_PARTITIONS 3847
7.3.1.29 GS_STAT_ALL_PARTITIONS 3847
7.3.2 负载管理 3849
7.3.2.1 GS_WLM_CGROUP_INFO 3849
7.3.2.2 GS_WLM_REBUILD_USER_RESOURCE_POOL 3849
7.3.2.3 GS_WLM_USER_INFO 3849
7.3.2.4 GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO 3850
7.3.2.5 GLOBAL_WLM_RESPOOL_CONNECTION_INFO 3851
7.3.2.6 GLOBAL_WLM_RESPOOL_CPU_INFO 3851
7.3.2.7 GLOBAL_WLM_RESPOOL_IO_INFO 3852
7.3.2.8 GLOBAL_WLM_RESPOOL_MEMORY_INFO 3853
7.3.2.9 PGXC_WLM_REBUILD_USER_RESPOOL 3854
7.3.3 OLTP 表压缩 3854
7.3.3.1 GS_ADM_ILMDATAMOVEMENTPOLICIES 3854
7.3.3.2 GS_ADM_ILMOBJECTS 3855
7.3.3.3 GS_ADM_ILMPOLICIES 3856
7.3.3.4 GS_ADM_ILMEVALUATIONDETAILS 3857
7.3.3.5 GS_ADM_ILMPARAMETERS 3858
7.3.3.6 GS_ADM_ILMRESULTS 3858
7.3.3.7 GS_ADM_ILMTASKS 3859
7.3.3.8 GS_MY_ILMEVALUATIONDETAILS 3859
7.3.3.9 GS_MY_ILMRESULTS 3860
7.3.3.10 GS_MY_ILMTASKS 3861
7.3.3.11 GS_MY_ILMDATAMOVEMENTPOLICIES 3862
7.3.3.12 GS_MY_ILMOBJECTS 3862
7.3.3.13 GS_MY_ILMPOLICIES 3863
7.3.4 通信 3864
7.3.4.1 COMM_CLIENT_INFO 3864
7.3.4.2 GLOBAL_COMM_CLIENT_INFO 3865
7.3.4.3 GS_COMM_LISTEN_ADDRESS_EXT_INFO 3865
7.3.4.4 GS_GET_LISTEN_ADDRESS_EXT_INFO 3866
7.3.4.5 GS_LIBCOMM_FD_INFO 3866
7.3.4.6 GS_LIBCOMM_MEMORY_INFO 3867
7.3.4.7 GS_STATIC_THREADPOOL_CTRL_STATUS 3867
7.3.4.8 PG_COMM_DELAY 3868
7.3.4.9 PG_COMM_RECV_STREAM 3869

 版权所有 © 华为云计算技术有限公司

7.3.4.10 PG_COMM_SEND_STREAM 3869
7.3.4.11 PG_COMM_STATUS 3870
7.3.4.12 PG_GET_INVALID_BACKENDS 3871
7.3.4.13 PG_POOLER_STATUS 3871
7.3.4.14 PGXC_COMM_DELAY 3872
7.3.4.15 PGXC_COMM_RECV_STREAM 3873
7.3.4.16 PGXC_COMM_SEND_STREAM 3874
7.3.4.17 PGXC_COMM_STATUS 3875
7.3.5 段页式存储 3875
7.3.5.1 GLOBAL_GS_SEG_DATAFILES 3875
7.3.5.2 GLOBAL_GS_SEG_EXTENTS 3876
7.3.5.3 GLOBAL_GS_SEG_SEGMENTS 3877
7.3.5.4 GLOBAL_GS_SEG_SPC_EXTENTS 3879
7.3.5.5 GLOBAL_GS_SEG_SPC_SEGMENTS 3881
7.3.5.6 GS_SEG_DATAFILES 3882
7.3.5.7 GS_SEG_DATAFILE_LAYOUT 3883
7.3.5.8 GS_SEG_EXTENTS 3884
7.3.5.9 GS_SEG_SEGMENTS 3885
7.3.5.10 GS_SEG_SEGMENT_LAYOUT 3887
7.3.5.11 GS_SEG_SPC_EXTENTS 3887
7.3.5.12 GS_SEG_SPC_SEGMENTS 3888
7.3.5.13 GS_SEG_SPC_REMAIN_EXTENTS 3890
7.3.5.14 GS_SEG_SPC_REMAIN_SEGMENTS 3891
7.3.6 SPM 计划管理 3892
7.3.6.1 GS_SPM_SQL_BASELINE 3892
7.3.6.2 GS_SPM_SQL_PARAM 3893
7.3.6.3 GS_SPM_SQL_EVOLUTION 3894
7.3.6.4 GS_SPM_SYS_BASELINE 3895
7.3.7 审计 3896
7.3.7.1 ADM_AUDIT_OBJECT 3896
7.3.7.2 ADM_AUDIT_SESSION 3898
7.3.7.3 ADM_AUDIT_STATEMENT 3899
7.3.7.4 ADM_AUDIT_TRAIL 3902
7.3.7.5 GS_AUDITING 3904
7.3.7.6 GS_AUDITING_ACCESS 3904
7.3.7.7 GS_AUDITING_PRIVILEGE 3905
7.3.8 用户和权限管理 3906
7.3.8.1 ADM_COL_PRIVS 3906
7.3.8.2 ADM_ROLE_PRIVS 3906
7.3.8.3 ADM_ROLES 3907
7.3.8.4 ADM_SYS_PRIVS 3907
7.3.8.5 ADM_TAB_PRIVS 3908

 版权所有 © 华为云计算技术有限公司

7.3.8.6 ADM_USERS 3909
7.3.8.7 DB_COL_PRIVS 3911
7.3.8.8 DB_DIRECTORIES 3911
7.3.8.9 DB_TAB_PRIVS 3912
7.3.8.10 DB_USERS 3913
7.3.8.11 GS_DB_PRIVILEGES 3913
7.3.8.12 GS_LABELS 3913
7.3.8.13 MY_COL_PRIVS 3914
7.3.8.14 MY_ROLE_PRIVS 3914
7.3.8.15 MY_SYS_PRIVS 3915
7.3.8.16 PG_RLSPOLICIES 3916
7.3.8.17 GS_RLSPOLICIES 3916
7.3.8.18 PG_ROLES 3917
7.3.8.19 GS_ROLES 3919
7.3.8.20 PG_SECLABELS 3921
7.3.8.21 GS_SECLABELS 3922
7.3.8.22 PG_SHADOW 3923
7.3.8.23 GS_SHADOW 3924
7.3.8.24 PG_USER 3926
7.3.8.25 GS_USER 3927
7.3.8.26 PG_USER_MAPPINGS 3929
7.3.8.27 ROLE_ROLE_PRIVS 3929
7.3.8.28 ROLE_SYS_PRIVS 3930
7.3.8.29 ROLE_TAB_PRIVS 3931
7.3.9 动态脱敏 3931
7.3.9.1 GS_MASKING 3931
7.3.10 透明加密 3932
7.3.10.1 PG_TDE_INFO 3932
7.3.11 DATABASE LINK 3932
7.3.11.1 GS_DB_LINKS 3932
7.3.11.2 V$DBLINK 3933
7.3.12 物化视图 3934
7.3.12.1 GS_MATVIEWS 3934
7.3.13 其他系统视图 3935
7.3.13.1 ADM_ARGUMENTS 3935
7.3.13.2 ADM_COL_COMMENTS 3937
7.3.13.3 ADM_COLL_TYPES 3937
7.3.13.4 ADM_CONS_COLUMNS 3938
7.3.13.5 ADM_CONSTRAINTS 3939
7.3.13.6 ADM_DATA_FILES 3940
7.3.13.7 ADM_DEPENDENCIES 3940
7.3.13.8 ADM_DIRECTORIES 3941

 版权所有 © 华为云计算技术有限公司

7.3.13.9 ADM_HIST_SNAPSHOT 3941
7.3.13.10 ADM_HIST_SQL_PLAN 3942
7.3.13.11 ADM_HIST_SQLSTAT 3944
7.3.13.12 ADM_HIST_SQLTEXT 3945
7.3.13.13 ADM_IND_COLUMNS 3946
7.3.13.14 ADM_IND_EXPRESSIONS 3946
7.3.13.15 ADM_INDEXES 3947
7.3.13.16 ADM_OBJECTS 3949
7.3.13.17 ADM_PROCEDURES 3951
7.3.13.18 ADM_SCHEDULER_JOB_ARGS 3952
7.3.13.19 ADM_SCHEDULER_JOBS 3953
7.3.13.20 ADM_SCHEDULER_PROGRAM_ARGS 3956
7.3.13.21 ADM_SCHEDULER_PROGRAMS 3957
7.3.13.22 ADM_SCHEDULER_RUNNING_JOBS 3958
7.3.13.23 ADM_SEGMENTS 3959
7.3.13.24 ADM_SEQUENCES 3960
7.3.13.25 ADM_SOURCE 3961
7.3.13.26 ADM_SYNONYMS 3961
7.3.13.27 ADM_TAB_COL_STATISTICS 3962
7.3.13.28 ADM_TAB_COLS 3963
7.3.13.29 ADM_TAB_COLUMNS 3966
7.3.13.30 ADM_TAB_COMMENTS 3968
7.3.13.31 ADM_TAB_HISTOGRAMS 3969
7.3.13.32 ADM_TAB_STATISTICS 3969
7.3.13.33 ADM_TAB_STATS_HISTORY 3971
7.3.13.34 ADM_TABLES 3971
7.3.13.35 ADM_TABLESPACES 3976
7.3.13.36 ADM_TRIGGERS 3978
7.3.13.37 ADM_TYPE_ATTRS 3979
7.3.13.38 ADM_TYPES 3980
7.3.13.39 ADM_VIEWS 3981
7.3.13.40 DB_ARGUMENTS 3982
7.3.13.41 DB_ALL_TABLES 3984
7.3.13.42 DB_COL_COMMENTS 3984
7.3.13.43 DB_COLL_TYPES 3985
7.3.13.44 DB_CONS_COLUMNS 3986
7.3.13.45 DB_CONSTRAINTS 3986
7.3.13.46 DB_DEPENDENCIES 3987
7.3.13.47 DB_IND_COLUMNS 3987
7.3.13.48 DB_IND_EXPRESSIONS 3988
7.3.13.49 DB_INDEXES 3988
7.3.13.50 DB_OBJECTS 3991

 版权所有 © 华为云计算技术有限公司

7.3.13.51 DB_PROCEDURES....3993
7.3.13.52 DB_SCHEDULER_JOB_ARGS....3993
7.3.13.53 DB_SCHEDULER_PROGRAM_ARGS....3994
7.3.13.54 DB_SEQUENCES....3994
7.3.13.55 DB_SOURCE....3995
7.3.13.56 DB_SYNONYMS....3996
7.3.13.57 DB_TAB_COL_STATISTICS....3997
7.3.13.58 DB_TAB_COLUMNS....3998
7.3.13.59 DB_TAB_COMMENTS....4000
7.3.13.60 DB_TAB_HISTOGRAMS....4001
7.3.13.61 DB_TAB_MODIFICATIONS....4001
7.3.13.62 DB_TAB_STATS_HISTORY....4002
7.3.13.63 DB_TABLES....4003
7.3.13.64 DB_TRIGGERS....4007
7.3.13.65 DB_TYPES....4007
7.3.13.66 DB_VIEWS....4008
7.3.13.67 DICT....4010
7.3.13.68 DICTIONARY....4010
7.3.13.69 DUAL....4010
7.3.13.70 DV_SESSIONS....4011
7.3.13.71 DV_SESSION_LONGOPS....4011
7.3.13.72 GET_GLOBAL_PREPARED_XACTS....4011
7.3.13.73 GLOBAL_BAD_BLOCK_INFO....4012
7.3.13.74 GLOBAL_CLEAR_BAD_BLOCK_INFO....4013
7.3.13.75 GLOBAL_SQL_PATCH....4013
7.3.13.76 GLOBAL_STAT_HOTKEYS_INFO....4014
7.3.13.77 GLOBAL_WAL_SENDER_STATUS....4014
7.3.13.78 GS_ALL_CONTROL_GROUP_INFO....4017
7.3.13.79 GS_ALL_PREPARED_STATEMENTS....4017
7.3.13.80 GS_AUTONOMOUS_TRANSACTION_DETAIL....4018
7.3.13.81 GS_BUCKET_LOCKS....4019
7.3.13.82 GS_GET_CONTROL_GROUP_INFO....4020
7.3.13.83 GS_GLC_MEMORY_DETAIL....4021
7.3.13.84 GS_GLOBAL_ARCHIVE_STATUS....4022
7.3.13.85 GS_GSC_MEMORY_DETAIL....4022
7.3.13.86 GS_GTT_ATTACHED_PIDS....4022
7.3.13.87 GS_GLOBAL_AUTONOMOUS_TRANSACTION_DETAIL....4023
7.3.13.88 GS_GLOBAL_GTT_ATTACHED_PIDS....4023
7.3.13.89 GS_GLOBAL_DIST_GTT_REMAIN_SESSIONS....4024
7.3.13.90 GS_LOGICALSTANDBY_SKIP....4024
7.3.13.91 GS_LOGICALSTANDBY_SKIP_TRANSACTION....4025
7.3.13.92 GS_LSC_MEMORY_DETAIL....4025

 版权所有 © 华为云计算技术有限公司

7.3.13.93 GS_MY_PLAN_TRACE 4026
7.3.13.94 GS_SESSION_ALL_SETTINGS 4026
7.3.13.95 GS_SQL_COUNT 4026
7.3.13.96 GS_SYSTEM_OBJECT_VERSION_DETAILS 4029
7.3.13.97 GS_THRDSTREAM_WORKER_INFO 4030
7.3.13.98 GS_TOTAL_NODEGROUP_MEMORY_DETAIL 4031
7.3.13.99 GS_WORKLOAD_RULE_STAT 4032
7.3.13.100 GV_INSTANCE 4033
7.3.13.101 GV_SESSION 4034
7.3.13.102 MPP_TABLES 4039
7.3.13.103 MY_AUDIT_TRAIL 4040
7.3.13.104 MY_COL_COMMENTS 4042
7.3.13.105 MY_COLL_TYPES 4043
7.3.13.106 MY_CONS_COLUMNS 4044
7.3.13.107 MY_CONSTRAINTS 4044
7.3.13.108 MY_DEPENDENCIES 4045
7.3.13.109 MY_IND_COLUMNS 4046
7.3.13.110 MY_IND_EXPRESSIONS 4046
7.3.13.111 MY_INDEXES 4047
7.3.13.112 MY_JOBS 4050
7.3.13.113 MY_OBJECTS 4052
7.3.13.114 MY_PROCEDURES 4053
7.3.13.115 MY_SCHEDULER_JOB_ARGS 4054
7.3.13.116 MY_SCHEDULER_JOBS 4055
7.3.13.117 MY_SCHEDULER_PROGRAM_ARGS 4058
7.3.13.118 MY_SEQUENCES 4059
7.3.13.119 MY_SOURCE 4060
7.3.13.120 MY_SYNONYMS 4060
7.3.13.121 MY_TAB_COL_STATISTICS 4061
7.3.13.122 MY_TAB_COLUMNS 4062
7.3.13.123 MY_TAB_COMMENTS 4065
7.3.13.124 MY_TAB_HISTOGRAMS 4065
7.3.13.125 MY_TAB_MODIFICATIONS 4066
7.3.13.126 MY_TAB_STATS_HISTORY 4067
7.3.13.127 MY_TAB_STATISTICS 4067
7.3.13.128 MY_TABLES 4069
7.3.13.129 MY_TABLESPACES 4074
7.3.13.130 MY_TRIGGERS 4076
7.3.13.131 MY_TYPE_ATTRS 4077
7.3.13.132 MY_TYPES 4078
7.3.13.133 MY_VIEWS 4079
7.3.13.134 NLS_DATABASE_PARAMETERS 4081

 版权所有 © 华为云计算技术有限公司

7.3.13.135 NLS_INSTANCE_PARAMETERS....4081
7.3.13.136 PG_AVAILABLE_EXTENSION_VERSIONS....4081
7.3.13.137 PG_AVAILABLE_EXTENSIONS....4082
7.3.13.138 PG_CONTROL_GROUP_CONFIG....4082
7.3.13.139 PG_CURSORS....4083
7.3.13.140 PG_EXT_STATS....4083
7.3.13.141 PG_GET_SENDERS_CATCHUP_TIME....4085
7.3.13.142 PG_GROUP....4086
7.3.13.143 PG_GTT_STATS....4086
7.3.13.144 PG_GTT_RELSTATS....4087
7.3.13.145 PG_GTT_ATTACHED_PIDS....4087
7.3.13.146 PG_INDEXES....4088
7.3.13.147 PG_LOCKS....4088
7.3.13.148 PG_NODE_ENV....4090
7.3.13.149 PG_OS_THREADS....4090
7.3.13.150 PG_PREPARED_STATEMENTS....4091
7.3.13.151 PG_PREPARED_XACTS....4091
7.3.13.152 PG_REPLICATION_SLOTS....4092
7.3.13.153 PG_RULES....4093
7.3.13.154 PG_RUNNING_XACTS....4094
7.3.13.155 PG_SESSION_IOSTAT....4094
7.3.13.156 PG_SETTINGS....4095
7.3.13.157 PG_SHARED_MEMORY_DETAIL....4096
7.3.13.158 PG_STATS....4097
7.3.13.159 PG_STAT_ACTIVITY....4098
7.3.13.160 PG_STAT_ACTIVITY_NG....4101
7.3.13.161 PG_STAT_ALL_INDEXES....4104
7.3.13.162 PG_STAT_ALL_TABLES....4104
7.3.13.163 PG_STAT_BAD_BLOCK....4105
7.3.13.164 PG_STAT_BGWRITER....4106
7.3.13.165 PG_STAT_DATABASE....4107
7.3.13.166 PG_STAT_DATABASE_CONFLICTS....4108
7.3.13.167 PG_STAT_REPLICATION....4109
7.3.13.168 PG_STAT_SYS_INDEXES....4110
7.3.13.169 PG_STAT_SYS_TABLES....4110
7.3.13.170 PG_STAT_USER_FUNCTIONS....4112
7.3.13.171 PG_STAT_USER_INDEXES....4112
7.3.13.172 PG_STAT_USER_TABLES....4112
7.3.13.173 PG_STAT_XACT_ALL_TABLES....4114
7.3.13.174 PG_STAT_XACT_SYS_TABLES....4114
7.3.13.175 PG_STAT_XACT_USER_FUNCTIONS....4115
7.3.13.176 PG_STAT_XACT_USER_TABLES....4115

 版权所有 © 华为云计算技术有限公司

7.3.13.177 PG_STATIO_ALL_INDEXES....4116
7.3.13.178 PG_STATIO_ALL_SEQUENCES....4116
7.3.13.179 PG_STATIO_ALL_TABLES....4117
7.3.13.180 PG_STATIO_SYS_INDEXES....4117
7.3.13.181 PG_STATIO_SYS_SEQUENCES....4118
7.3.13.182 PG_STATIO_SYS_TABLES....4118
7.3.13.183 PG_STATIO_USER_INDEXES....4119
7.3.13.184 PG_STATIO_USER_SEQUENCES....4119
7.3.13.185 PG_STATIO_USER_TABLES....4120
7.3.13.186 PG_TABLES....4120
7.3.13.187 PG_THREAD_WAIT_STATUS....4121
7.3.13.188 PG_TIMEZONE_ABBREVS....4171
7.3.13.189 PG_TIMEZONE_NAMES....4172
7.3.13.190 PG_TOTAL_MEMORY_DETAIL....4172
7.3.13.191 PG_TOTAL_USER_RESOURCE_INFO....4172
7.3.13.192 PG_TOTAL_USER_RESOURCE_INFO_OID....4174
7.3.13.193 PG_VARIABLE_INFO....4175
7.3.13.194 PG_VIEWS....4176
7.3.13.195 PGXC_COPY_ERROR_LOG....4176
7.3.13.196 PGXC_GET_STAT_ALL_TABLES....4177
7.3.13.197 PGXC_GET_TABLE_SKEWNESS....4178
7.3.13.198 PGXC_NODE_ENV....4179
7.3.13.199 PGXC_OS_THREADS....4179
7.3.13.200 PGXC_PREPARED_XACTS....4179
7.3.13.201 PGXC_RUNNING_XACTS....4180
7.3.13.202 PGXC_STAT_ACTIVITY....4180
7.3.13.203 PGXC_STAT_BAD_BLOCK....4183
7.3.13.204 PGXC_SQL_COUNT....4184
7.3.13.205 PGXC_THREAD_WAIT_STATUS....4184
7.3.13.206 PGXC_TOTAL_MEMORY_DETAIL....4186
7.3.13.207 PGXC_VARIABLE_INFO....4188
7.3.13.208 PLAN_TABLE....4189
7.3.13.209 PV_FILE_STAT....4191
7.3.13.210 PV_INSTANCE_TIME....4191
7.3.13.211 PV_OS_RUN_INFO....4192
7.3.13.212 PV_REDO_STAT....4192
7.3.13.213 PV_SESSION_MEMORY....4193
7.3.13.214 PV_SESSION_MEMORY_CONTEXT....4193
7.3.13.215 PV_SESSION_MEMORY_DETAIL....4194
7.3.13.216 PV_SESSION_STAT....4195
7.3.13.217 PV_SESSION_TIME....4196
7.3.13.218 PV_THREAD_MEMORY_CONTEXT....4197

 版权所有 © 华为云计算技术有限公司

7.3.13.219 PV_TOTAL_MEMORY_DETAIL....4197
7.3.13.220 SYS_DUMMY....4199
7.3.13.221 V_INSTANCE....4199
7.3.13.222 V_MYSTAT....4201
7.3.13.223 V_SESSION....4201
7.3.13.224 V$GLOBAL_OPEN_CURSOR....4206
7.3.13.225 V$GLOBAL_TRANSACTION....4207
7.3.13.226 V$LOCK....4208
7.3.13.227 V$NLS_PARAMETERS....4209
7.3.13.228 V$OPEN_CURSOR....4210
7.3.13.229 V$SESSION_WAIT....4211
7.3.13.230 V$SYSSTAT....4213
7.3.13.231 V$SYSTEM_EVENT....4213
7.3.13.232 V$VERSION....4214
7.3.14 废弃系统视图/系统表....4214
8 Schema....4215
8.1 Information Schema....4216
8.1.1 INFORMATION_SCHEMA._PG_FOREIGN_DATA_WRAPPERS....4217
8.1.2 INFORMATION_SCHEMA._PG_FOREIGN_SERVERS....4218
8.1.3 INFORMATION_SCHEMA._PG_FOREIGN_TABLE_COLUMNS....4218
8.1.4 INFORMATION_SCHEMA._PG_FOREIGN_TABLES....4219
8.1.5 INFORMATION_SCHEMA._PG_USER_MAPPINGS....4219
8.1.6 INFORMATION_SCHEMA.INFORMATION_SCHEMA_CATALOG_NAME....4220
8.1.7 INFORMATION_SCHEMA.TRIGGERS....4220
8.2 DBE_PERF Schema....4222
8.2.1 OS....4222
8.2.1.1 DBE_PERF.OS_RUNTIME....4222
8.2.1.2 DBE_PERF.GLOBAL_OS_RUNTIME....4222
8.2.1.3 DBE_PERF.OS_THREADS....4223
8.2.1.4 DBE_PERF.GLOBAL_OS_THREADS....4223
8.2.1.5 DBE_PERF.PERF_QUERY....4224
8.2.1.6 DBE_PERF.SHAKING_COLLECT_STATUS....4224
8.2.2 Instance....4224
8.2.2.1 DBE_PERF.INSTANCE_TIME....4224
8.2.2.2 DBE_PERF.GLOBAL_INSTANCE_TIME....4225
8.2.3 Memory....4226
8.2.3.1 DBE_PERF.MEMORY_NODE_DETAIL....4226
8.2.3.2 DBE_PERF.GLOBAL_MEMORY_NODE_DETAIL....4228
8.2.3.3 DBE_PERF.MEMORY_NODE_NG_DETAIL....4230
8.2.3.4 DBE_PERF.SHARED_MEMORY_DETAIL....4231
8.2.3.5 DBE_PERF.GLOBAL_SHARED_MEMORY_DETAIL....4231
8.2.3.6 DBE_PERF.TRACK_MEMORY_CONTEXT_DETAIL....4231

版权所有 © 华为云计算技术有限公司

8.2.4 File 4232
8.2.4.1 DBE_PERF.FILE_IOSTAT 4232
8.2.4.2 DBE_PERF.SUMMARY_FILE_IOSTAT 4232
8.2.4.3 DBE_PERF.GLOBAL_FILE_IOSTAT 4233
8.2.4.4 DBE_PERF.FILE_REDO_IOSTAT 4234
8.2.4.5 DBE_PERF.SUMMARY_FILE_REDO_IOSTAT 4234
8.2.4.6 DBE_PERF.GLOBAL_FILE_REDO_IOSTAT 4235
8.2.4.7 DBE_PERF.LOCAL_REL_IOSTAT 4235
8.2.4.8 DBE_PERF.SUMMARY_REL_IOSTAT 4236
8.2.4.9 DBE_PERF.GLOBAL_REL_IOSTAT 4236
8.2.5 Object 4236
8.2.5.1 DBE_PERF.STAT_USER_TABLES 4237
8.2.5.2 DBE_PERF.SUMMARY_STAT_USER_TABLES 4238
8.2.5.3 DBE_PERF.GLOBAL_STAT_USER_TABLES 4239
8.2.5.4 DBE_PERF.STAT_USER_INDEXES 4240
8.2.5.5 DBE_PERF.SUMMARY_STAT_USER_INDEXES 4241
8.2.5.6 DBE_PERF.GLOBAL_STAT_USER_INDEXES 4241
8.2.5.7 DBE_PERF.STAT_SYS_TABLES 4242
8.2.5.8 DBE_PERF.SUMMARY_STAT_SYS_TABLES 4243
8.2.5.9 DBE_PERF.GLOBAL_STAT_SYS_TABLES 4244
8.2.5.10 DBE_PERF.STAT_SYS_INDEXES 4245
8.2.5.11 DBE_PERF.SUMMARY_STAT_SYS_INDEXES 4246
8.2.5.12 DBE_PERF.GLOBAL_STAT_SYS_INDEXES 4246
8.2.5.13 DBE_PERF.STAT_ALL_TABLES 4247
8.2.5.14 DBE_PERF.SUMMARY_STAT_ALL_TABLES 4248
8.2.5.15 DBE_PERF.GLOBAL_STAT_ALL_TABLES 4249
8.2.5.16 DBE_PERF.STAT_ALL_INDEXES 4250
8.2.5.17 DBE_PERF.SUMMARY_STAT_ALL_INDEXES 4251
8.2.5.18 DBE_PERF.GLOBAL_STAT_ALL_INDEXES 4251
8.2.5.19 DBE_PERF.STAT_DATABASE 4252
8.2.5.20 DBE_PERF.SUMMARY_STAT_DATABASE 4253
8.2.5.21 DBE_PERF.GLOBAL_STAT_DATABASE 4254
8.2.5.22 DBE_PERF.STAT_DATABASE_CONFLICTS 4256
8.2.5.23 DBE_PERF.SUMMARY_STAT_DATABASE_CONFLICTS 4256
8.2.5.24 DBE_PERF.GLOBAL_STAT_DATABASE_CONFLICTS 4256
8.2.5.25 DBE_PERF.STAT_XACT_ALL_TABLES 4257
8.2.5.26 DBE_PERF.SUMMARY_STAT_XACT_ALL_TABLES 4258
8.2.5.27 DBE_PERF.GLOBAL_STAT_XACT_ALL_TABLES 4258
8.2.5.28 DBE_PERF.STAT_XACT_SYS_TABLES 4259
8.2.5.29 DBE_PERF.SUMMARY_STAT_XACT_SYS_TABLES 4259
8.2.5.30 DBE_PERF.GLOBAL_STAT_XACT_SYS_TABLES 4260
8.2.5.31 DBE_PERF.STAT_XACT_USER_TABLES 4261

 版权所有 © 华为云计算技术有限公司

8.2.5.32 DBE_PERF.SUMMARY_STAT_XACT_USER_TABLES 4261
8.2.5.33 DBE_PERF.GLOBAL_STAT_XACT_USER_TABLES 4262
8.2.5.34 DBE_PERF.STAT_XACT_USER_FUNCTIONS 4263
8.2.5.35 DBE_PERF.SUMMARY_STAT_XACT_USER_FUNCTIONS 4263
8.2.5.36 DBE_PERF.GLOBAL_STAT_XACT_USER_FUNCTIONS 4263
8.2.5.37 DBE_PERF.STAT_BAD_BLOCK 4264
8.2.5.38 DBE_PERF.SUMMARY_STAT_BAD_BLOCK 4264
8.2.5.39 DBE_PERF.GLOBAL_STAT_BAD_BLOCK 4265
8.2.5.40 DBE_PERF.STAT_USER_FUNCTIONS 4265
8.2.5.41 DBE_PERF.SUMMARY_STAT_USER_FUNCTIONS 4266
8.2.5.42 DBE_PERF.GLOBAL_STAT_USER_FUNCTIONS 4266
8.2.6 Workload 4267
8.2.6.1 DBE_PERF.WORKLOAD_SQL_COUNT 4267
8.2.6.2 DBE_PERF.SUMMARY_WORKLOAD_SQL_COUNT 4268
8.2.6.3 DBE_PERF.WORKLOAD_TRANSACTION 4270
8.2.6.4 DBE_PERF.SUMMARY_WORKLOAD_TRANSACTION 4271
8.2.6.5 DBE_PERF.GLOBAL_WORKLOAD_TRANSACTION 4272
8.2.6.6 DBE_PERF.WORKLOAD_SQL_ELAPSE_TIME 4273
8.2.6.7 DBE_PERF.SUMMARY_WORKLOAD_SQL_ELAPSE_TIME 4274
8.2.6.8 DBE_PERF.USER_TRANSACTION 4275
8.2.6.9 DBE_PERF.GLOBAL_USER_TRANSACTION 4276
8.2.7 Session/Thread 4277
8.2.7.1 DBE_PERF.SESSION_STAT 4277
8.2.7.2 DBE_PERF.GLOBAL_SESSION_STAT 4278
8.2.7.3 DBE_PERF.SESSION_TIME 4278
8.2.7.4 DBE_PERF.GLOBAL_SESSION_TIME 4278
8.2.7.5 DBE_PERF.SESSION_MEMORY 4279
8.2.7.6 DBE_PERF.GLOBAL_SESSION_MEMORY 4279
8.2.7.7 DBE_PERF.SESSION_MEMORY_DETAIL 4279
8.2.7.8 DBE_PERF.GLOBAL_SESSION_MEMORY_DETAIL 4280
8.2.7.9 DBE_PERF.SESSION_STAT_ACTIVITY 4280
8.2.7.10 DBE_PERF.GLOBAL_SESSION_STAT_ACTIVITY 4282
8.2.7.11 DBE_PERF.THREAD_WAIT_STATUS 4285
8.2.7.12 DBE_PERF.GLOBAL_THREAD_WAIT_STATUS 4285
8.2.7.13 DBE_PERF.LOCAL_THREADPOOL_STATUS 4286
8.2.7.14 DBE_PERF.GLOBAL_THREADPOOL_STATUS 4287
8.2.7.15 DBE_PERF.SESSION_CPU_RUNTIME 4287
8.2.7.16 DBE_PERF.SESSION_MEMORY_RUNTIME 4288
8.2.7.17 DBE_PERF.STATEMENT_IOSTAT_COMPLEX_RUNTIME 4289
8.2.7.18 DBE_PERF.LOCAL_ACTIVE_SESSION 4289
8.2.7.19 DBE_PERF.GLOBAL_ACTIVE_SESSION 4291
8.2.8 Transaction 4293

 版权所有 © 华为云计算技术有限公司

8.2.8.1 DBE_PERF.TRANSACTIONS_RUNNING_XACTS....4293
8.2.8.2 DBE_PERF.SUMMARY_TRANSACTIONS_RUNNING_XACTS....4294
8.2.8.3 DBE_PERF.GLOBAL_TRANSACTIONS_RUNNING_XACTS....4295
8.2.8.4 DBE_PERF.TRANSACTIONS_PREPARED_XACTS....4295
8.2.8.5 DBE_PERF.SUMMARY_TRANSACTIONS_PREPARED_XACTS....4296
8.2.8.6 DBE_PERF.GLOBAL_TRANSACTIONS_PREPARED_XACTS....4296
8.2.9 Query....4297
8.2.9.1 DBE_PERF.STATEMENT....4297
8.2.9.2 DBE_PERF.SUMMARY_STATEMENT....4301
8.2.9.3 DBE_PERF.STATEMENT_COUNT....4305
8.2.9.4 DBE_PERF.GLOBAL_STATEMENT_COUNT....4307
8.2.9.5 DBE_PERF.SUMMARY_STATEMENT_COUNT....4309
8.2.9.6 DBE_PERF.STATEMENT_RESPONSETIME_PERCENTILE....4312
8.2.9.7 DBE_PERF.STATEMENT_HISTORY....4312
8.2.9.8 DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY....4318
8.2.9.9 DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY_TABLE....4322
8.2.9.10 DBE_PERF.STATEMENT_WLMSTAT_COMPLEX_RUNTIME....4322
8.2.9.11 DBE_PERF.SESSION_SQL_MEMORY....4324
8.2.9.12 DBE_PERF.GLOBAL_SESSION_SQL_MEMORY....4324
8.2.9.13 DBE_PERF.GS_PLANCACHE_STAT....4325
8.2.9.14 DBE_PERF.GS_DIST_PLANCACHE_STAT....4326
8.2.10 Cache/IO....4327
8.2.10.1 DBE_PERF.STATIO_USER_TABLES....4327
8.2.10.2 DBE_PERF.SUMMARY_STATIO_USER_TABLES....4328
8.2.10.3 DBE_PERF.GLOBAL_STATIO_USER_TABLES....4328
8.2.10.4 DBE_PERF.STATIO_USER_INDEXES....4329
8.2.10.5 DBE_PERF.SUMMARY_STATIO_USER_INDEXES....4330
8.2.10.6 DBE_PERF.GLOBAL_STATIO_USER_INDEXES....4330
8.2.10.7 DBE_PERF.STATIO_USER_SEQUENCES....4330
8.2.10.8 DBE_PERF.SUMMARY_STATIO_USER_SEQUENCES....4331
8.2.10.9 DBE_PERF.GLOBAL_STATIO_USER_SEQUENCES....4331
8.2.10.10 DBE_PERF.STATIO_SYS_TABLES....4332
8.2.10.11 DBE_PERF.SUMMARY_STATIO_SYS_TABLES....4332
8.2.10.12 DBE_PERF.GLOBAL_STATIO_SYS_TABLES....4333
8.2.10.13 DBE_PERF.STATIO_SYS_INDEXES....4334
8.2.10.14 DBE_PERF.SUMMARY_STATIO_SYS_INDEXES....4334
8.2.10.15 DBE_PERF.GLOBAL_STATIO_SYS_INDEXES....4335
8.2.10.16 DBE_PERF.STATIO_SYS_SEQUENCES....4335
8.2.10.17 DBE_PERF.SUMMARY_STATIO_SYS_SEQUENCES....4336
8.2.10.18 DBE_PERF.GLOBAL_STATIO_SYS_SEQUENCES....4336
8.2.10.19 DBE_PERF.STATIO_ALL_TABLES....4336
8.2.10.20 DBE_PERF.SUMMARY_STATIO_ALL_TABLES....4337

 版权所有 © 华为云计算技术有限公司

8.2.10.21 DBE_PERF.GLOBAL_STATIO_ALL_TABLES 4338
8.2.10.22 DBE_PERF.STATIO_ALL_INDEXES 4339
8.2.10.23 DBE_PERF.SUMMARY_STATIO_ALL_INDEXES 4339
8.2.10.24 DBE_PERF.GLOBAL_STATIO_ALL_INDEXES 4340
8.2.10.25 DBE_PERF.STATIO_ALL_SEQUENCES 4340
8.2.10.26 DBE_PERF.SUMMARY_STATIO_ALL_SEQUENCES 4341
8.2.10.27 DBE_PERF.GLOBAL_STATIO_ALL_SEQUENCES 4341
8.2.11 Comm 4342
8.2.11.1 DBE_PERF.COMM_DELAY 4342
8.2.11.2 DBE_PERF.GLOBAL_COMM_DELAY 4342
8.2.11.3 DBE_PERF.COMM_RECV_STREAM 4343
8.2.11.4 DBE_PERF.GLOBAL_COMM_RECV_STREAM 4343
8.2.11.5 DBE_PERF.COMM_SEND_STREAM 4344
8.2.11.6 DBE_PERF.GLOBAL_COMM_SEND_STREAM 4345
8.2.11.7 DBE_PERF.COMM_STATUS 4346
8.2.11.8 DBE_PERF.GLOBAL_COMM_STATUS 4346
8.2.11.9 DBE_PERF.GLOBAL_COMM_CHECK_CONNECTION_STATUS 4347
8.2.12 Utility 4348
8.2.12.1 DBE_PERF.REPLICATION_STAT 4348
8.2.12.2 DBE_PERF.GLOBAL_REPLICATION_STAT 4349
8.2.12.3 DBE_PERF.REPLICATION_SLOTS 4350
8.2.12.4 DBE_PERF.GLOBAL_REPLICATION_SLOTS 4350
8.2.12.5 DBE_PERF.BGWRITER_STAT 4351
8.2.12.6 DBE_PERF.GLOBAL_BGWRITER_STAT 4352
8.2.12.7 DBE_PERF.POOLER_STATUS 4353
8.2.12.8 DBE_PERF.POOLER_AGENT 4354
8.2.12.9 DBE_PERF.GLOBAL_CKPT_STATUS 4354
8.2.12.10 DBE_PERF.GLOBAL_DOUBLE_WRITE_STATUS 4355
8.2.12.11 DBE_PERF.GLOBAL_PAGEWRITER_STATUS 4355
8.2.12.12 DBE_PERF.GLOBAL_POOLER_STATUS 4356
8.2.12.13 DBE_PERF.GLOBAL_RECORD_RESET_TIME 4357
8.2.12.14 DBE_PERF.GLOBAL_REDO_STATUS 4357
8.2.12.15 DBE_PERF.GLOBAL_RECOVERY_STATUS 4358
8.2.12.16 DBE_PERF.CLASS_VITAL_INFO 4359
8.2.12.17 DBE_PERF.USER_LOGIN 4359
8.2.12.18 DBE_PERF.SUMMARY_USER_LOGIN 4360
8.2.12.19 DBE_PERF.GLOBAL_GET_BGWRITER_STATUS 4360
8.2.12.20 DBE_PERF.GLOBAL_SINGLE_FLUSH_DW_STATUS 4361
8.2.12.21 DBE_PERF.GLOBAL_CANDIDATE_STATUS 4361
8.2.12.22 DBE_PERF.PARALLEL_DECODE_STATUS 4362
8.2.12.23 DBE_PERF.GLOBAL_PARALLEL_DECODE_STATUS 4362
8.2.12.24 DBE_PERF.PARALLEL_DECODE_THREAD_INFO 4363

 版权所有 © 华为云计算技术有限公司

8.2.12.25 DBE_PERF.GLOBAL_PARALLEL_DECODE_THREAD_INFO....4363
8.2.12.26 DBE_PERF.GLOBAL_ADIO_COMPLETER_STATUS....4364
8.2.12.27 DBE_PERF.GLOBAL_AIO_SLOT_USAGE_STATUS....4364
8.2.12.28 DBE_PERF.GLOBAL_XLOG_REDO_STAT....4365
8.2.12.29 DBE_PERF.EMERGENCY_OPERATION_RECORDS....4365
8.2.12.30 DBE_PERF.KERNEL_INFO....4366
8.2.13 Lock....4368
8.2.13.1 DBE_PERF.LOCKS....4368
8.2.13.2 DBE_PERF.GLOBAL_LOCKS....4369
8.2.14 Wait Events....4370
8.2.14.1 DBE_PERF.WAIT_EVENTS....4371
8.2.14.2 DBE_PERF.GLOBAL_WAIT_EVENTS....4371
8.2.14.3 DBE_PERF.WAIT_EVENT_INFO....4372
8.2.14.4 DBE_PERF.GLOBAL_DATANODES_WAIT_EVENTS....4384
8.2.15 Configuration....4385
8.2.15.1 DBE_PERF.CONFIG_SETTINGS....4385
8.2.15.2 DBE_PERF.GLOBAL_CONFIG_SETTINGS....4386
8.2.16 Operator....4387
8.2.16.1 DBE_PERF.OPERATOR_HISTORY_TABLE....4387
8.2.16.2 DBE_PERF.OPERATOR_RUNTIME....4388
8.2.16.3 DBE_PERF.GLOBAL_OPERATOR_HISTORY....4389
8.2.16.4 DBE_PERF.GLOBAL_OPERATOR_HISTORY_TABLE....4391
8.2.16.5 DBE_PERF.GLOBAL_OPERATOR_RUNTIME....4391
8.2.17 Workload Manager....4392
8.2.17.1 DBE_PERF.WLM_CGROUP_CONFIG....4392
8.2.17.2 DBE_PERF.WLM_CLUSTER_RESOURCE_RUNTIME....4393
8.2.17.3 DBE_PERF.WLM_CONTROLGROUP_CONFIG....4393
8.2.17.4 DBE_PERF.WLM_CONTROLGROUP_NG_CONFIG....4394
8.2.17.5 DBE_PERF.WLM_RESOURCEPOOL_RUNTIME....4395
8.2.17.6 DBE_PERF.WLM_USER_RESOURCE_CONFIG....4395
8.2.17.7 DBE_PERF.WLM_USER_RESOURCE_RUNTIME....4396
8.2.17.8 DBE_PERF.WLM_WORKLOAD_HISTORY_INFO....4396
8.2.17.9 DBE_PERF.WLM_WORKLOAD_RUNTIME....4397
8.2.17.10 DBE_PERF.GLOBAL_WLM_WORKLOAD_RUNTIME....4398
8.2.17.11 DBE_PERF.LOCAL_IO_WAIT_INFO....4399
8.2.17.12 DBE_PERF.GLOBAL_IO_WAIT_INFO....4400
8.2.18 Global Plancache....4400
8.2.18.1 DBE_PERF.LOCAL_PLANCACHE_STATUS....4400
8.2.18.2 DBE_PERF.GLOBAL_PLANCACHE_STATUS....4402
8.2.18.3 DBE_PERF.LOCAL_PREPARE_STATEMENT_STATUS....4402
8.2.18.4 DBE_PERF.GLOBAL_PREPARE_STATEMENT_STATUS....4402
8.2.18.5 DBE_PERF.LOCAL_PLANCACHE_CLEAN....4402

 版权所有 © 华为云计算技术有限公司

8.2.18.6 DBE_PERF.GLOBAL_PLANCACHE_CLEAN....4402
8.2.19 RTO & RPO....4402
8.2.19.1 DBE_PERF.GLOBAL_RTO_STATUS....4403
8.2.19.2 DBE_PERF.GLOBAL_STREAMING_HADR_RTO_AND_RPO_STAT....4403
8.2.20 AI Watchdog....4403
8.2.20.1 DBE_PERF.AI_WATCHDOG_MONITOR_STATUS....4404
8.2.20.2 DBE_PERF.AI_WATCHDOG_DETECTION_WARNINGS....4405
8.2.20.3 DBE_PERF.AI_WATCHDOG_PARAMETERS....4405
8.2.20.4 DBE_PERF.AI_WATCHDOG_FTASK_STATUS....4406
8.2.21 Trace Function....4406
8.2.21.1 DBE_PERF.pl_trace_func_start....4406
8.2.21.2 DBE_PERF.pl_trace_func_end....4407
8.2.21.3 DBE_PERF.pl_trace_info_clean....4407
8.2.21.4 DBE_PERF.pl_get_trace_info....4407
8.2.21.5 DBE_PERF.pl_get_global_trace_info....4408
8.2.22 废弃视图....4408
8.2.22.1 Query....4408
8.2.22.2 Operator....4409
8.3 WDR Snapshot Schema....4409
8.3.1 WDR Snapshot 原信息....4409
8.3.1.1 SNAPSHOT.SNAPSHOT....4409
8.3.1.2 SNAPSHOT.TABLES_SNAP_TIMESTAMP....4410
8.3.1.3 SNAPSHOT.SNAP_SEQ....4410
8.3.1.4 SNAPSHOT.SNAP_QUERY....4410
8.3.1.5 SNAPSHOT.SNAP_PLAN....4411
8.3.2 WDR Snapshot 数据表....4412
8.4 DBE_PLDEBUGGER Schema....4412
8.4.1 DBE_PLDEBUGGER.turn_on....4415
8.4.2 DBE_PLDEBUGGER.turn_off....4416
8.4.3 DBE_PLDEBUGGER.local_debug_server_info....4416
8.4.4 DBE_PLDEBUGGER.attach....4417
8.4.5 DBE_PLDEBUGGER.info_locals....4417
8.4.6 DBE_PLDEBUGGER.next....4417
8.4.7 DBE_PLDEBUGGER.continue....4418
8.4.8 DBE_PLDEBUGGER.abort....4418
8.4.9 DBE_PLDEBUGGER.print_var....4419
8.4.10 DBE_PLDEBUGGER.info_code....4419
8.4.11 DBE_PLDEBUGGER.step....4419
8.4.12 DBE_PLDEBUGGER.add_breakpoint....4420
8.4.13 DBE_PLDEBUGGER.delete_breakpoint....4420
8.4.14 DBE_PLDEBUGGER.info_breakpoints....4420
8.4.15 DBE_PLDEBUGGER.backtrace....4421

 版权所有 © 华为云计算技术有限公司

8.4.16 DBE_PLDEBUGGER.enable_breakpoint 4421
8.4.17 DBE_PLDEBUGGER.disable_breakpoint 4421
8.4.18 DBE_PLDEBUGGER.finish 4422
8.4.19 DBE_PLDEBUGGER.set_var 4422
8.4.20 DBE_PLDEBUGGER.error_backtrace 4422
8.4.21 DBE_PLDEBUGGER.error_end 4423
8.4.22 DBE_PLDEBUGGER.error_info_locals 4423
8.5 DBE_PLDEVELOPER 4424
8.5.1 DBE_PLDEVELOPER.GS_SOURCE 4424
8.6 DBE_COMPRESSION Schema 4424
8.7 DBE_HEAT_MAP Schema 4425
8.8 DBE_ILM Schema 4425
8.9 DBE_ILM_ADMIN Schema 4426
8.10 PRVT_ILM Schema 4426
9 WDR 报告 4429
9.1 Database Stat 4431
9.2 Load Profile 4432
9.3 Instance Efficiency Percentages 4433
9.4 Top 10 Events by Total Wait Time 4433
9.5 Wait Classes by Total Wait Time 4433
9.6 Host CPU 4434
9.7 IO Profile 4434
9.8 Memory Statistics 4435
9.9 Time Model 4435
9.10 SQL Statistics 4436
9.11 Wait Events 4438
9.12 Cache IO Stats 4439
9.13 Utility status 4440
9.14 Object Stats 4441
9.15 Configuration settings 4443
9.16 SQL Detail 4444
10 ASP 报告 4445

 版权所有 © 华为云计算技术有限公司

1 SQL 参考

1.1 GaussDB SQL

什么是 SQL

SQL是用于访问和处理数据库的标准计算机语言。

SQL提供了各种任务的语句，包括：

- 查询数据。
- 在表中插入、更新和删除行。
- 创建、替换、更改和删除对象。
- 控制对数据库及其对象的访问。
- 保证数据库的一致性和完整性。

SQL语言由用于处理数据库和数据库对象的命令和函数组成。该语言还会强制实施有关数据类型、表达式和文本使用的规则。因此在**SQL参考**章节，除了SQL语法参考外，还介绍了有关数据类型、表达式、函数和操作符等信息。

SQL 发展简史

SQL发展简史如下：

- 1986年，ANSI X3.135-1986，ISO/IEC 9075:1986，SQL-86
- 1989年，ANSI X3.135-1989，ISO/IEC 9075:1989，SQL-89
- 1992年，ANSI X3.135-1992，ISO/IEC 9075:1992，SQL-92
- 1999年，ISO/IEC 9075:1999，SQL:1999
- 2003年，ISO/IEC 9075:2003，SQL:2003
- 2011年，ISO/IEC 9075:2011，SQL:2011
- 2016年，ISO/IEC 9075:2016，SQL:2016
- 2019年，ISO/IEC 9075:2019，SQL:2019

版权所有 © 华为云计算技术有限公司

GaussDB 支持的 SQL 标准

GaussDB默认支持SQL:2016的大部分特性。

1.2 关键字

SQL里有保留关键字和非保留关键字之分，常见的SQL关键字请参见**表1-1**。根据标准，保留关键字绝不能用做其他标识符。非保留关键字只是在特定的环境里有特殊的含义，而在其他环境里是可以用做标识符的。

须知

- 目前“非保留”关键字在作为数据库对象的标识符时存在如下限制：
  - 不支持直接作为列别名使用，即类似SELECT 1 ABORT的用法会导致错误。
  - 对于ENTITYESCAPING、NOENTITYESCAPING以及WELLFORMED关键字，不带双引号时不支持作为表名、列名、表别名、列别名以及函数名的标识符。
  - 不带双引号的RAW关键字不支持作为表名和函数名的标识符。
  - 不带双引号的SET关键字不支持作为表别名的标识符，即类似SELECT * FROM T1 SET的用法均会导致错误。
  - 不带双引号的BEGIN、BY、CLOSE、CURSOR、DECLARE、DELETE、EXECUTE、FUNCTION、IF、IMMEDIATE、INSERT、LOOP、MOVE、OF、REF、RELEASE、RETURN、SAVEPOINT、STRICT、TYPE以及UPDATE等关键字不支持作为变量名使用。
  - 用SYS_REFCURSOR关键字作为数据库对象的标识符时，如果不附带双引号，则创建名为REFCURSOR的数据库对象，如果附带了双引号，则创建名为SYS_REFCURSOR的数据库对象。
- 与“非保留”关键字类似，“非保留（不能是函数或类型）”关键字不支持直接作为列别名使用。
- 对于未带有双引号的“保留”关键字CURRENT_TIMESTAMP而言，不允许作为函数名。
- CONVERT、TIMESTAMPADD、ADDDATE、DATE_ADD、SUBDATE、DATE_SUB、GET_FORMAT仅在MYSQL兼容和M兼容模式作为关键字；JSON_OBJECT、LAST_DAY、IFNULL仅在MYSQL兼容模式作为关键字。
- NOORDER仅在ORA兼容模式作为关键字。

标识符命名规范

标识符的命名需要遵守如下规范：

- 标识符需要为小写字母（a-z）、大写字母(A-Z)、下划线（_）、数字（0-9）或美元符号（$）。
- 标识符必须以字母或下划线开头。

  说明

  - 此命名规范为建议项，非强制项。
  - 特殊情况下可以使用双引号规避特殊字符报错。

版权所有 © 华为云计算技术有限公司

SQL 关键字

表 1-1 SQL 关键字

关键字	GaussDB	SQL:1999	SQL-92
ABORT	非保留	-	-
ABS	-	非保留	-
ABSOLUTE	非保留	保留	保留
ACCESS	非保留	-	-
ACCOUNT	非保留	-	-
ACTION	非保留	保留	保留
ADA	-	非保留	非保留
ADD	非保留	保留	保留
ADDDATE	非保留	-	-
ADMIN	非保留	保留	-
ADVANCED	非保留	-	-
AFTER	非保留	保留	-
AGGREGATE	非保留	保留	-
ALGORITHM	非保留	-	-
ALIAS	-	保留	-
ALL	保留	保留	保留
ALLOCATE	-	保留	保留
ALSO	非保留	-	-
ALTER	非保留	保留	保留
ALWAYS	非保留	-	-
ANALYSE	保留	-	-
ANALYZE	保留	-	-
AND	保留	保留	保留
ANY	保留	保留	保留
APP	非保留	-	-
APPEND	非保留	-	-
ARCHIVE	非保留	-	-
ARE	-	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
ARRAY	保留	保留	-
AS	保留	保留	保留
ASC	保留	保留	保留
ASENSITIVE	-	非保留	-
ASSERTION	非保留	保留	保留
ASSIGNMENT	非保留	非保留	-
ASYMMETRIC	保留	非保留	-
AT	非保留	保留	保留
ATOMIC	-	非保留	-
ATTRIBUTE	非保留	-	-
AUDIT	非保留	-	-
AUTHID	保留	-	-
AUTHORIZATION	保留（可以是函数或类型）	保留	保留
AUTO	非保留	-	-
AUTO_INCREMENT	非保留	-	-
AUTOEXTEND	非保留	-	-
AUTOHINT	非保留	-	-
AUTOMAPPED	非保留	-	-
AVG	-	非保留	保留
BACKWARD	非保留	-	-
BAD_PATH	非保留	-	-
BARRIER	非保留	-	-
BEFORE	非保留	保留	-
BEGIN	非保留	保留	保留
BEGIN_NON_ANOYBLOCK	非保留	-	-
BETWEEN	非保留（不能是函数或类型）	非保留	保留
BIGINT	非保留（不能是函数或类型）	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
BINARY	保留（可以是函数或类型）	保留	-
BINARY_DOUBLE	非保留（不能是函数或类型）	-	-
BINARY_INTEGER	非保留（不能是函数或类型）	-	-
BIT	非保留（不能是函数或类型）	保留	保留
BIT_LENGTH	-	非保留	保留
BITVAR	-	非保留	-
BLANKS	非保留	-	-
BLOB	非保留	保留	-
BLOCKCHAIN	非保留	-	-
BODY	非保留	-	-
BOOLEAN	非保留（不能是函数或类型）	保留	-
BOTH	保留	保留	保留
BREADTH	-	保留	-
BUCKETS	保留	-	-
BY	非保留	保留	保留
BYTE	非保留	-	-
BYTEAWITHOUTORDER	非保留（不能是函数或类型）	-	-
BYTEAWITHOUTORDERWITHEQUAL	非保留（不能是函数或类型）	-	-
C	-	非保留	非保留
CACHE	非保留	-	-
CALL	非保留	保留	-
CALLED	非保留	非保留	-
CANCELABLE	非保留	-	-
CARDINALITY	-	非保留	-
CASCADE	非保留	保留	保留
CASCADED	非保留	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
CASE	保留	保留	保留
CAST	保留	保留	保留
CATALOG	非保留	保留	保留
CATALOG_NAME	-	非保留	非保留
CHAIN	非保留	非保留	-
CHANGE	非保留	-	-
CHAR	非保留（不能是函数或类型）	保留	保留
CHAR_LENGTH	-	非保留	保留
CHARACTER	非保留（不能是函数或类型）	保留	保留
CHARACTER_LENGTH	-	非保留	保留
CHARACTER_SET_CATALOG	-	非保留	非保留
CHARACTER_SET_NAME	-	非保留	非保留
CHARACTER_SET_SCHEMA	-	非保留	非保留
CHARACTERISTICS	非保留	-	-
CHARACTERSET	非保留	-	-
CHARSET	非保留	-	-
CHECK	保留	保留	保留
CHECKED	-	非保留	-
CHECKPOINT	非保留	-	-
CLASS	非保留	保留	-
CLASS_ORIGIN	-	非保留	非保留
CLEAN	非保留	-	-
CLIENT	非保留	-	-
CLIENT_MASTER_KEY	非保留	-	-
CLIENT_MASTER_KEYS	非保留	-	-
CLOB	非保留	保留	-
CLOSE	非保留	保留	保留
CLUSTER	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
COALESCE	非保留（不能是函数或类型）	非保留	保留
COBOL	-	非保留	非保留
COLLATE	保留	保留	保留
COLLATION	保留（可以是函数或类型）	保留	保留
COLLATION_CATALOG	-	非保留	非保留
COLLATION_NAME	-	非保留	非保留
COLLATION_SCHEMA	-	非保留	非保留
COLUMN	保留	保留	保留
COLUMN_ENCRYPTION_KEY	非保留	-	-
COLUMN_ENCRYPTION_KEYS	非保留	-	-
COLUMN_NAME	-	非保留	非保留
COLUMNS	非保留	-	-
COLVIEW	非保留	-	-
COMMAND_FUNCTION	-	非保留	非保留
COMMAND_FUNCTION_CODE	-	非保留	-
COMMENT	非保留	-	-
COMMENTS	非保留	-	-
COMMIT	非保留	保留	保留
COMMITTED	非保留	非保留	非保留
COMPACT	保留（可以是函数或类型）	-	-
COMPATIBLE_ILLEGAL_CHARS	非保留	-	-
COMPILE	非保留	-	-
COMPLETE	非保留	-	-
COMPLETION	非保留	保留	-
COMPRESS	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
CONCURRENTLY	保留（可以是函数或类型）	-	-
CONDITION	非保留	-	-
CONDITION_NUMBER	-	非保留	非保留
CONFIGURATION	非保留	-	-
CONFLICT	非保留	-	-
CONNECT	非保留	保留	保留
CONNECTION	非保留	保留	保留
CONNECTION_NAME	-	非保留	非保留
CONSTANT	非保留	-	-
CONSTRAINT	保留	保留	保留
CONSTRAINT_CATALOG	-	非保留	非保留
CONSTRAINT_NAME	-	非保留	非保留
CONSTRAINT_SCHEMA	-	非保留	非保留
CONSTRAINTS	非保留	保留	保留
CONSTRUCTOR	-	保留	-
CONTAINING	非保留	-	-
CONTAINS	-	非保留	-
CONTENT	非保留	-	-
CONTINUE	非保留	保留	保留
CONVERSION	非保留	-	-
CONVERT	非保留	非保留	保留
COORDINATOR	非保留	-	-
COORDINATORS	非保留	-	-
COPY	非保留	-	-
CORRESPONDING	-	保留	保留
COST	非保留	-	-
COUNT	-	非保留	保留
CREATE	保留	保留	保留
CROSS	保留（可以是函数或类型）	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
CROSSBUCKET	保留	-	-
CSN	保留（可以是函数或类型）	-	-
CSV	非保留	-	-
CUBE	非保留	保留	-
CURRENT	非保留	保留	保留
CURRENT_CATALOG	保留	-	-
CURRENT_DATE	保留	保留	保留
CURRENT_PATH	-	保留	-
CURRENT_ROLE	保留	保留	-
CURRENT_SCHEMA	保留（可以是函数或类型）	-	-
CURRENT_TIME	保留	保留	保留
CURRENT_TIMESTAMP	保留	保留	保留
CURRENT_USER	保留	保留	保留
CURSOR	非保留	保留	保留
CURSOR_NAME	-	非保留	非保留
CYCLE	非保留	保留	-
DATA	非保留	保留	非保留
DATABASE	非保留	-	-
DATAFILE	非保留	-	-
DATANODE	非保留	-	-
DATANODES	非保留	-	-
DATATYPE_CL	非保留	-	-
DATE	非保留（不能是函数或类型）	保留	保留
DATE_ADD	非保留	-	-
DATE_FORMAT	非保留	-	-
DATE_SUB	非保留	-	-
DATETIME	非保留	-	-
DATETIME_INTERVAL_CODE	-	非保留	非保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
DATETIME_INTERVAL_PRECISION	-	非保留	非保留
DAY	非保留	保留	保留
DAY_HOUR	非保留	-	-
DAY_MICROSECOND	非保留	-	-
DAY_MINUTE	非保留	-	-
DAY_SECOND	非保留	-	-
DAYS	非保留	-	-
DB4AISHOT	非保留	-	-
DBCOMPATIBILITY	非保留	-	-
DBTIMEZONE	保留	-	-
DEALLOCATE	非保留	保留	保留
DEC	非保留（不能是函数或类型）	保留	保留
DECIMAL	非保留（不能是函数或类型）	保留	保留
DECLARE	非保留	保留	保留
DECODE	非保留（不能是函数或类型）	-	-
DEFAULT	保留	保留	保留
DEFAULTS	非保留	-	-
DEFERRABLE	保留	保留	保留
DEFERRED	非保留	保留	保留
DEFINED	-	非保留	-
DEFINER	非保留	非保留	-
DELETE	非保留	保留	保留
DELETE_ALL	非保留	-	-
DELIMITER	非保留	-	-
DELIMITERS	非保留	-	-
DELTA	非保留	-	-
DELTAMERGE	保留（可以是函数或类型）	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
DEPTH	-	保留	-
DEREF	-	保留	-
DESC	保留	保留	保留
DESCRIBE	-	保留	保留
DESCRIPTOR	-	保留	保留
DESTROY	-	保留	-
DESTRUCTOR	-	保留	-
DETERMINISTIC	非保留	保留	-
DIAGNOSTICS	-	保留	保留
DICTIONARY	非保留	保留	-
DIRECT	非保留	-	-
DIRECTORY	非保留	-	-
DISABLE	非保留	-	-
DISABLE_ALL	非保留	-	-
DISCARD	非保留	-	-
DISCARD_PATH	非保留	-	-
DISCONNECT	非保留	保留	保留
DISPATCH	-	非保留	-
DISTINCT	保留	保留	保留
DISTRIBUTE	非保留	-	-
DISTRIBUTED	非保留	-	-
DISTRIBUTION	非保留	-	-
DO	保留	-	-
DOCUMENT	非保留	-	-
DOMAIN	非保留	保留	保留
DOUBLE	非保留	保留	保留
DROP	非保留	保留	保留
DUMPFILE	非保留	-	-
DUPLICATE	非保留	-	-
DYNAMIC	-	保留	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
DYNAMIC_FUNCTION	-	非保留	非保留
DYNAMIC_FUNCTION_CODE	-	非保留	-
EACH	非保留	保留	-
ELASTIC	非保留	-	-
ELSE	保留	保留	保留
EMPTY	非保留	-	-
ENABLE	非保留	-	-
ENABLE_ALL	非保留	-	-
ENCLOSED	非保留	-	-
ENCODING	非保留	-	-
ENCRYPTED	非保留	-	-
ENCRYPTED_VALUE	非保留	-	-
ENCRYPTION	非保留	-	-
ENCRYPTION_TYPE	非保留	-	-
END	保留	保留	保留
END-EXEC	-	保留	保留
ENDS	非保留	-	-
ENFORCED	非保留	-	-
ENGINE	非保留	-	-
ENTITYESCAPING	非保留	-	-
ENUM	非保留	-	-
EOL	非保留	-	-
EQUALS	-	保留	-
ERROR	非保留	-	-
ERRORS	非保留	-	-
ESCAPE	非保留	保留	保留
ESCAPED	非保留	-	-
ESCAPING	非保留	-	-
EVALNAME	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
EVENT	非保留	-	-
EVENTS	非保留	-	-
EVERY	非保留	保留	-
EXCEPT	保留	保留	保留
EXCEPTION	-	保留	保留
EXCHANGE	非保留	-	-
EXCLUDE	非保留	-	-
EXCLUDED	非保留	-	-
EXCLUDING	非保留	-	-
EXCLUSIVE	非保留	-	-
EXEC	-	保留	保留
EXECUTE	非保留	保留	保留
EXISTING	-	非保留	-
EXISTS	非保留（不能是函数或类型）	非保留	保留
EXPDP	非保留	-	-
EXPIRED	非保留	-	-
EXPLAIN	非保留	-	-
EXTEND	非保留	-	-
EXTENSION	非保留	-	-
EXTERNAL	非保留	保留	保留
EXTRACT	非保留（不能是函数或类型）	非保留	保留
FALSE	保留	保留	保留
FAMILY	非保留	-	-
FAST	非保留	-	-
FEATURES	非保留	-	-
FENCED	保留	-	-
FETCH	保留	保留	保留
FIELDS	非保留	-	-
FILEHEADER	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
FILL_MISSING_FIELDS	非保留	-	-
FILLER	非保留	-	-
FILTER	非保留	-	保留
FINAL	-	非保留	-
FINISH	非保留	-	-
FIRST	非保留	保留	保留
FIXED	非保留	-	保留
FLOAT	非保留（不能是函数或类型）	保留	保留
FOLLOWING	非保留	-	-
FOR	保留	保留	保留
FORCE	保留	-	-
FOREIGN	保留	保留	保留
FORMATTER	非保留	-	-
FORTRAN	-	非保留	非保留
FORWARD	非保留	-	-
FOUND	-	保留	保留
FREE	-	保留	-
FREEZE	保留（可以是函数或类型）	-	-
FROM	保留	保留	保留
FULL	保留（可以是函数或类型）	保留	保留
FUNCTION	非保留	保留	-
FUNCTIONS	非保留	-	-
G	-	非保留	-
GENERAL	-	保留	-
GENERATED	非保留	非保留	-
GET	-	保留	保留
GET_FORMAT	非保留	-	-
GLOBAL	非保留	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
GO	-	保留	保留
GOTO	-	保留	保留
GRANT	保留	保留	保留
GRANTED	非保留	非保留	-
GREATEST	非保留（不能是函数或类型）	-	-
GROUP	保留	保留	保留
GROUPING	非保留（不能是函数或类型）	保留	-
GSIUSABLE	非保留	-	-
GSIVALID	非保留	-	-
GSIWAITALL	非保留	-	-
HANDLER	非保留	-	-
HAVING	保留	保留	保留
HDFSDIRECTORY	保留（可以是函数或类型）	-	-
HEADER	非保留	-	-
HIERARCHY	-	非保留	-
HIGH	非保留	-	-
HOLD	非保留	非保留	-
HOST	-	保留	-
HOUR	非保留	保留	保留
HOUR_MICROSECOND	非保留	-	-
HOUR_MINUTE	非保留	-	-
HOUR_SECOND	非保留	-	-
IDENTIFIED	非保留	-	-
IDENTITY	非保留	保留	保留
IF	非保留	-	-
IFNULL	非保留	-	-
IGNORE	非保留	保留	-
IGNORE_EXTRA_DATA	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
ILIKE	保留（可以是函数或类型）	-	-
ILM	非保留	-	-
ILM_PIDX_LIST	非保留	-	-
IMCVTABLE	非保留	-	-
IMMEDIATE	非保留	保留	保留
IMMUTABLE	非保留	-	-
IMPDP	非保留	-	-
IMPLEMENTATION	-	非保留	-
IMPLICIT	非保留	-	-
IN	保留	保留	保留
INCLUDE	非保留	-	-
INCLUDING	非保留	-	-
INCREMENT	非保留	-	-
INCREMENTAL	非保留	-	-
INDEX	非保留	-	-
INDEXES	非保留	-	-
INDICATOR	-	保留	保留
INFILE	非保留	-	-
INFIX	-	非保留	-
INHERIT	非保留	-	-
INHERITS	非保留	-	-
INITIAL	非保留	-	-
INITIALIZE	-	保留	-
INITIALLY	保留	保留	保留
INITRANS	非保留	-	-
INLINE	非保留	-	-
INNER	保留（可以是函数或类型）	保留	保留
INOUT	非保留（不能是函数或类型）	保留	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
INPUT	非保留	保留	保留
INSENSITIVE	非保留	非保留	保留
INSERT	非保留	保留	保留
INSTANCE	-	非保留	-
INSTANTIABLE	-	非保留	-
INSTEAD	非保留	-	-
INT	非保留（不能是函数或类型）	保留	保留
INTEGER	非保留（不能是函数或类型）	保留	保留
INTERNAL	非保留	-	-
INTERSECT	保留	保留	保留
INTERVAL	非保留（不能是函数或类型）	保留	保留
INTO	保留	保留	保留
INVISIBLE	非保留	-	-
INVOKER	非保留	非保留	-
IP	非保留	-	-
IS	保留	保留	保留
ISNULL	非保留	-	-
ISOLATION	非保留	保留	保留
ITERATE	-	保留	-
JOIN	保留（可以是函数或类型）	保留	保留
JSON_OBJECT	非保留	-	-
K	-	非保留	-
KEY	非保留	保留	保留
KEY_MEMBER	-	非保留	-
KEY_PATH	非保留	-	-
KEY_STORE	非保留	-	-
KEY_TYPE	-	非保留	-
KILL	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
LABEL	非保留	-	-
LANGUAGE	非保留	保留	保留
LARGE	非保留	保留	-
LAST	非保留	保留	保留
LAST_DAY	非保留	-	-
LATERAL	-	保留	-
LC_COLLATE	非保留	-	-
LC_CTYPE	非保留	-	-
LEADING	保留	保留	保留
LEAKPROOF	非保留	-	-
LEAST	非保留（不能是函数或类型）	-	-
LEFT	保留（可以是函数或类型）	保留	保留
LENGTH	-	非保留	非保留
LESS	保留	保留	-
LEVEL	非保留	保留	保留
LIKE	保留（可以是函数或类型）	保留	保留
LIMIT	保留	保留	-
LINES	非保留	-	-
LINK	非保留	-	-
LIST	非保留	-	-
LISTEN	非保留	-	-
LNNVL	非保留（不能是函数或类型）	-	-
LOAD	非保留	-	-
LOAD_BAD	非保留	-	-
LOAD_DISCARD	非保留	-	-
LOAD_INTEGER_LEN	非保留	-	-
LOAD_SESSION_ID	非保留	-	-
LOAD_SMALLINT_LEN	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
LOAD_UNFIXED_LEN	非保留	-	-
LOAD_UNFIXED_START_POS	非保留	-	-
LOCAL	非保留	保留	保留
LOCALTIME	保留	保留	-
LOCALTIMESTAMP	保留	保留	-
LOCATION	非保留	-	-
LOCATOR	-	保留	-
LOCK	非保留	-	-
LOCKED	非保留	-	-
LOG	非保留	-	-
LOGGING	非保留	-	-
LOGIN_ANY	非保留	-	-
LOGIN_FAILURE	非保留	-	-
LOGIN_SUCCESS	非保留	-	-
LOGOUT	非保留	-	-
LOOP	非保留	-	-
LOW	非保留	-	-
LOWER	-	非保留	保留
M	-	非保留	-
MAP	-	保留	-
MAPPING	非保留	-	-
MARK	非保留	-	-
MASKING	非保留	-	-
MASTER	保留	-	-
MATCH	非保留	保留	保留
MATCHED	非保留	-	-
MATERIALIZED	非保留	-	-
MAX	-	非保留	保留
MAXEXTENTS	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
MAXSIZE	非保留	-	-
MAXTRANS	非保留	-	-
MAXVALUE	保留	-	-
MEDIUM	非保留	-	-
MEDIUMINT	非保留（不能是函数或类型）	-	-
MERGE	非保留	-	-
MESSAGE_LENGTH	-	非保留	非保留
MESSAGE_OCTET_LENGTH	-	非保留	非保留
MESSAGE_TEXT	-	非保留	非保留
METHOD	-	非保留	-
MICROSECOND	非保留	-	-
MIN	-	非保留	保留
MINEXTENTS	非保留	-	-
MINUS	保留	-	-
MINUTE	非保留	保留	保留
MINUTE_MICROSECOND	非保留	-	-
MINUTE_SECOND	非保留	-	-
MINVALUE	非保留	-	-
MOD	-	非保留	-
MODE	非保留	-	-
MODEL	非保留	-	-
MODIFICATION	非保留	-	-
MODIFIES	-	保留	-
MODIFY	保留	保留	-
MODULE	-	保留	保留
MONTH	非保留	保留	保留
MONTHS	非保留	-	-
MORE	-	非保留	非保留
MOVE	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
MOVEMENT	非保留	-	-
MUMPS	-	非保留	非保留
NAME	非保留	非保留	非保留
NAMES	非保留	保留	保留
NATIONAL	非保留（不能是函数或类型）	保留	保留
NATURAL	保留（可以是函数或类型）	保留	保留
NCHAR	非保留（不能是函数或类型）	保留	保留
NCLOB	-	保留	-
NEW	-	保留	-
NEXT	非保留	保留	保留
NO	非保留	保留	保留
NOCACHE	非保留	-	-
NOCOLVIEW	非保留	-	-
NOCOMPRESS	非保留	-	-
NOCYCLE	保留	-	-
NODE	非保留	-	-
NOENTITYESCAPING	非保留	-	-
NOEXTEND	非保留	-	-
NOLOGGING	非保留	-	-
NOMAXVALUE	非保留	-	-
NOMINVALUE	非保留	-	-
NONE	非保留（不能是函数或类型）	保留	-
NOORDER	非保留	-	-
NOSCALE	非保留	-	-
NOT	保留	保留	保留
NOTHING	非保留	-	-
NOTIFY	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
NOTNULL	保留（可以是函数或类型）	-	-
NOW	非保留	-	-
NOWAIT	非保留	-	-
NULL	保留	保留	保留
NULLABLE	-	非保留	非保留
NULLCOLS	非保留	-	-
NULLIF	非保留（不能是函数或类型）	非保留	保留
NULLS	非保留	-	-
NUMBER	非保留（不能是函数或类型）	非保留	非保留
NUMERIC	非保留（不能是函数或类型）	保留	保留
NUMSTR	非保留	-	-
NVARCHAR2	非保留（不能是函数或类型）	-	-
NVL	非保留（不能是函数或类型）	-	-
NVL2	非保留（不能是函数或类型）	-	-
OBJECT	非保留	保留	-
OCTET_LENGTH	-	非保留	保留
OF	非保留	保留	保留
OFF	非保留	保留	-
OFFLINE	保留	-	-
OFFSET	保留	-	-
OIDS	非保留	-	-
OLD	-	保留	-
ON	保留	保留	保留
ONLINE	保留	-	-
ONLY	保留	保留	保留
OPEN	-	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
OPERATION	-	保留	-
OPERATOR	非保留	-	-
OPTIMIZATION	非保留	-	-
OPTION	非保留	保留	保留
OPTIONALLY	非保留	-	-
OPTIONS	非保留	非保留	-
OR	保留	保留	保留
ORDER	保留	保留	保留
ORDINALITY	非保留	保留	-
OUT	非保留（不能是函数或类型）	保留	-
OUTER	保留（可以是函数或类型）	保留	保留
OUTFILE	非保留	-	-
OUTPUT	-	保留	保留
OVER	非保留	-	-
OVERLAPS	保留（可以是函数或类型）	非保留	保留
OVERLAY	非保留（不能是函数或类型）	非保留	-
OVERRIDING	-	非保留	-
OWNED	非保留	-	-
OWNER	非保留	-	-
PACKAGE	非保留	-	-
PACKAGES	非保留	-	-
PAD	-	保留	保留
PARAMETER	-	保留	-
PARAMETER_MODE	-	非保留	-
PARAMETER_NAME	-	非保留	-
PARAMETER_ORDINAL_POSITION	-	非保留	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
PARAMETER_SPECIFIC_CATALOG	-	非保留	-
PARAMETER_SPECIFIC_NAME	-	非保留	-
PARAMETER_SPECIFIC_SCHEMA	-	非保留	-
PARAMETERS	-	保留	-
PARSER	非保留	-	-
PARTIAL	非保留	保留	保留
PARTITION	非保留	-	-
PARTITIONS	非保留	-	-
PASCAL	-	非保留	非保留
PASSING	非保留	-	-
PASSWORD	非保留	-	-
PATH	-	保留	-
PCTFREE	非保留	-	-
PER	非保留	-	-
PERCENT	非保留	-	-
PERFORMANCE	保留	-	-
PERM	非保留	-	-
PIVOT	非保留	-	-
PLACING	保留	-	-
PLAN	非保留	-	-
PLANS	非保留	-	-
PLI	-	非保留	非保留
POLICY	非保留	-	-
POOL	非保留	-	-
POPULATE	非保留	-	-
POSITION	非保留（不能是函数或类型）	非保留	保留
POSTFIX	-	保留	-
PRECEDING	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
PRECISION	非保留（不能是函数或类型）	保留	保留
PREDICT	非保留	-	-
PREFERRED	非保留	-	-
PREFIX	非保留	保留	-
PREORDER	-	保留	-
PREPARE	非保留	保留	保留
PREPARED	非保留	-	-
PRESERVE	非保留	保留	保留
PRIMARY	保留	保留	保留
PRIOR	非保留	保留	保留
PRIORER	保留	-	-
PRIORITY	非保留	-	-
PRIVATE	非保留	-	-
PRIVILEGE	非保留	-	-
PRIVILEGES	非保留	保留	保留
PROCEDURAL	非保留	-	-
PROCEDURE	保留	保留	保留
PROFILE	非保留	-	-
PUBLIC	非保留	保留	保留
PUBLISH	非保留	-	-
PURGE	非保留	-	-
QUARTER	非保留	-	-
QUERY	非保留	-	-
QUOTE	非保留	-	-
RANDOMIZED	非保留	-	-
RANGE	非保留	-	-
RATIO	非保留	-	-
RAW	非保留	-	-
READ	非保留	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
READS	-	保留	-
REAL	非保留（不能是函数或类型）	保留	保留
REASSIGN	非保留	-	-
REBUILD	非保留	-	-
RECHECK	非保留	-	-
RECOVER	非保留	-	-
RECURSIVE	非保留	保留	-
RECYCLEBIN	保留（可以是函数或类型）	-	-
REDISANYVALUE	非保留	-	-
REF	非保留	保留	-
REFERENCES	保留	保留	保留
REFERENCING	-	保留	-
REFRESH	非保留	-	-
REGEXP	保留（可以是函数或类型)	-	-
REGEXP_LIKE	非保留（不能是函数或类型）	-	-
REINDEX	非保留	-	-
REJECT	保留	-	-
RELATIVE	非保留	保留	保留
RELEASE	非保留	-	-
RELOPTIONS	非保留	-	-
REMOTE	非保留	-	-
REMOVE	非保留	-	-
RENAME	非保留	-	-
REPEATABLE	非保留	非保留	非保留
REPLACE	非保留	-	-
REPLICA	非保留	-	-
RESET	非保留	-	-
RESIZE	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
RESOURCE	非保留	-	-
RESPECT	非保留	-	-
RESTART	非保留	-	-
RESTRICT	非保留	保留	保留
RESULT	-	保留	-
RETURN	非保留	保留	-
RETURNED_LENGTH	-	非保留	非保留
RETURNED_OCTET_LENGTH	-	非保留	非保留
RETURNED_SQLSTATE	-	非保留	非保留
RETURNING	保留	-	-
RETURNS	非保留	保留	-
REUSE	非保留	-	-
REVOKE	非保留	保留	保留
RIGHT	保留（可以是函数或类型）	保留	保留
RLIKE	保留（可以是函数或类型)	-	-
ROLE	非保留	保留	-
ROLES	非保留	-	-
ROLLBACK	非保留	保留	保留
ROLLUP	非保留	保留	-
ROTATION	非保留	-	-
ROUTINE	-	保留	-
ROUTINE_CATALOG	-	非保留	-
ROUTINE_NAME	-	非保留	-
ROUTINE_SCHEMA	-	非保留	-
ROW	非保留（不能是函数或类型）	保留	-
ROW_COUNT	-	非保留	非保留
ROWNUM	保留	-	-
ROWS	非保留	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
ROWTYPE	非保留	-	-
RULE	非保留	-	-
SAMPLE	非保留	-	-
SAVEPOINT	非保留	保留	-
SCALE	非保留	非保留	非保留
SCHEDULE	非保留	-	-
SCHEMA	非保留	保留	保留
SCHEMA_NAME	-	非保留	非保留
SCOPE	-	保留	-
SCROLL	非保留	保留	保留
SEARCH	非保留	保留	-
SECOND	非保留	保留	保留
SECOND_MICROSECOND	非保留	-	-
SECTION	-	保留	保留
SECURITY	非保留	非保留	-
SELECT	保留	保留	保留
SELF	-	非保留	-
SENSITIVE	-	非保留	-
SEPARATOR	非保留	-	-
SEQUENCE	非保留	保留	-
SEQUENCES	非保留	-	-
SERIALIZABLE	非保留	非保留	非保留
SERVER	非保留	-	-
SERVER_NAME	-	非保留	非保留
SESSION	非保留	保留	保留
SESSION_USER	保留	保留	保留
SESSIONTIMEZONE	保留	-	-
SET	非保留	保留	保留
SETOF	非保留（不能是函数或类型）	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
SETS	非保留	保留	-
SHARE	非保留	-	-
SHIPPABLE	非保留	-	-
SHOW	非保留	-	-
SHRINK	保留	-	-
SHUTDOWN	非保留	-	-
SIBLINGS	非保留	-	-
SIGNED	非保留（不能是函数或类型）	-	-
SIMILAR	保留（可以是函数或类型）	非保留	-
SIMPLE	非保留	非保留	-
SIZE	非保留	保留	保留
SKIP	非保留	-	-
SLAVE	非保留	-	-
SLICE	非保留	-	-
SMALLDATETIME	非保留（不能是函数或类型）	-	-
SMALLDATETIME_FORMAT	非保留	-	-
SMALLINT	非保留（不能是函数或类型）	保留	保留
SNAPSHOT	非保留	-	-
SOME	保留	保留	保留
SOURCE	非保留	非保留	-
SPACE	非保留	保留	保留
SPECIFIC	-	保留	-
SPECIFIC_NAME	-	非保留	-
SPECIFICATION	非保留	-	-
SPECIFICTYPE	-	保留	-
SPILL	非保留	-	-
SPLIT	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
SQL	-	保留	保留
SQLCODE	-	-	保留
SQLERROR	-	-	保留
SQLEXCEPTION	-	保留	-
SQLSTATE	-	保留	保留
SQLWARNING	-	保留	-
STABLE	非保留	-	-
STANDALONE	非保留	-	-
START	非保留	保留	-
STARTING	非保留	-	-
STARTS	非保留	-	-
STATE	-	保留	-
STATEMENT	非保留	保留	-
STATEMENT_ID	非保留	-	-
STATIC	-	保留	-
STATISTICS	非保留	-	-
STDIN	非保留	-	-
STDOUT	非保留	-	-
STORAGE	非保留	-	-
STORE	非保留	-	-
STORED	非保留	-	-
STR	非保留	-	-
STRATIFY	非保留	-	-
STRICT	非保留	-	-
STRIP	非保留	-	-
STRUCTURE	-	保留	-
STYLE	-	非保留	-
SUBCLASS_ORIGIN	-	非保留	非保留
SUBDATE	非保留	-	-
SUBLIST	-	非保留	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
SUBPARTITION	非保留	-	-
SUBPARTITIONS	非保留	-	-
SUBSTR	非保留	-	-
SUBSTRING	非保留（不能是函数或类型）	非保留	保留
SUM	-	非保留	保留
SYMMETRIC	保留	非保留	-
SYNONYM	非保留	-	-
SYS_REFCURSOR	非保留	-	-
SYSDATE	保留	-	-
SYSID	非保留	-	-
SYSTEM	非保留	非保留	-
SYSTEM_USER	-	保留	保留
TABLE	保留	保留	保留
TABLE_COMPRESS_CLAUSE	非保留	-	-
TABLE_NAME	-	非保留	非保留
TABLES	非保留	-	-
TABLESAMPLE	保留（可以是函数或类型）	-	-
TABLESPACE	非保留	-	-
TARGET	非保留	-	-
TEMP	非保留	-	-
TEMPLATE	非保留	-	-
TEMPORARY	非保留	保留	保留
TERMINATE	-	保留	-
TERMINATED	非保留	-	-
TEXT	非保留	-	-
THAN	非保留	保留	-
THEN	保留	保留	保留
TIME	非保留（不能是函数或类型）	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
TIME_FORMAT	非保留	-	-
TIMECAPSULE	保留（可以是函数或类型）	-	-
TIMESTAMP	非保留（不能是函数或类型）	保留	保留
TIMESTAMP_FORMAT	非保留	-	-
TIMESTAMPADD	非保留	-	-
TIMESTAMPDIFF	非保留（不能是函数或类型）	-	-
TIMEZONE_HOUR	-	保留	保留
TIMEZONE_MINUTE	-	保留	保留
TINYINT	非保留（不能是函数或类型）	-	-
TO	保留	保留	保留
TRAILING	保留	保留	保留
TRANSACTION	非保留	保留	保留
TRANSACTION_ACTIVE	-	非保留	-
TRANSACTIONS_COMMITTED	-	非保留	-
TRANSACTIONS_ROLLED_BACK	-	非保留	-
TRANSFORM	非保留	非保留	-
TRANSFORMS	-	非保留	-
TRANSLATE	-	非保留	保留
TRANSLATION	-	保留	保留
TREAT	非保留（不能是函数或类型）	保留	-
TRIGGER	非保留	保留	-
TRIGGER_CATALOG	-	非保留	-
TRIGGER_NAME	-	非保留	-
TRIGGER_SCHEMA	-	非保留	-
TRIM	非保留（不能是函数或类型）	非保留	保留

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
TRUE	保留	保留	保留
TRUNCATE	非保留	-	-
TRUSTED	非保留	-	-
TSFIELD	非保留	-	-
TSTAG	非保留	-	-
TSTIME	非保留	-	-
TYPE	非保留	非保留	非保留
TYPES	非保留	-	-
UNBOUNDED	非保留	-	-
UNCOMMITTED	非保留	非保留	非保留
UNDER	-	保留	-
UNENCRYPTED	非保留	-	-
UNION	保留	保留	保留
UNIQUE	保留	保留	保留
UNKNOWN	非保留	保留	保留
UNLIMITED	非保留	-	-
UNLISTEN	非保留	-	-
UNLOCK	非保留	-	-
UNLOGGED	非保留	-	-
UNNAMED	-	非保留	非保留
UNNEST	-	保留	-
UNPIVOT	非保留	-	-
UNSIGNED	非保留（不能是函数或类型）	-	-
UNTIL	非保留	-	-
UNUSABLE	非保留	-	-
UPDATE	非保留	保留	保留
UPPER	-	非保留	保留
USAGE	-	保留	保留
USEEOF	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
USER	保留	保留	保留
USER_DEFINED_TYPE_CATALOG	-	非保留	-
USER_DEFINED_TYPE_NAME	-	非保留	-
USER_DEFINED_TYPE_SCHEMA	-	非保留	-
USING	保留	保留	保留
VACUUM	非保留	-	-
VALID	非保留	-	-
VALIDATE	非保留	-	-
VALIDATION	非保留	-	-
VALIDATOR	非保留	-	-
VALUE	非保留	保留	保留
VALUES	非保留（不能是函数或类型）	保留	保留
VARCHAR	非保留（不能是函数或类型）	保留	保留
VARCHAR2	非保留（不能是函数或类型）	-	-
VARIABLE	-	保留	-
VARIABLES	非保留	-	-
VARIADIC	保留	-	-
VARYING	非保留	保留	保留
VCGROUP	非保留	-	-
VERBOSE	保留（可以是函数或类型）	-	-
VERIFY	保留	-	-
VERSION	非保留	-	-
VIEW	非保留	保留	保留
VISIBLE	非保留	-	-
VOLATILE	非保留	-	-
WAIT	非保留	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
WEAK	非保留	-	-
WEEK	非保留	-	-
WELLFORMED	非保留	-	-
WHEN	保留	保留	保留
WHENEVER	-	保留	保留
WHERE	保留	保留	保留
WHITESPACE	非保留	-	-
WINDOW	保留	-	-
WITH	保留	保留	保留
WITHIN	非保留	-	-
WITHOUT	非保留	保留	-
WORK	非保留	保留	保留
WORKLOAD	非保留	-	-
WRAPPER	非保留	-	-
WRITE	非保留	保留	保留
XML	非保留	-	-
XMLATTRIBUTES	非保留（不能是函数或类型）	-	-
XMLCONCAT	非保留（不能是函数或类型）	-	-
XMLELEMENT	非保留（不能是函数或类型）	-	-
XMLEXISTS	非保留（不能是函数或类型）	-	-
XMLFOREST	非保留（不能是函数或类型）	-	-
XMLNAMESPACES	非保留	-	-
XMLPARSE	非保留（不能是函数或类型）	-	-
XMLPI	非保留（不能是函数或类型）	-	-
XMLROOT	非保留（不能是函数或类型）	-	-

版权所有 © 华为云计算技术有限公司

关键字	GaussDB	SQL:1999	SQL-92
XMLSERIALIZE	非保留（不能是函数或类型）	-	-
XMLTABLE	非保留	-	-
XMLQUERY	非保留	-	-
XMLCAST	非保留	-	-
XMLTYPE	非保留（不能是函数或类型）	-	-
YEAR	非保留	保留	保留
YEAR_MONTH	非保留	-	-
YEARS	非保留	-	-
YES	非保留	-	-
ZEROFILL	非保留（不能是函数或类型）	-	-
ZONE	非保留	保留	保留

下表所示字段在建表时禁止作为列名。

CTID	XMIN	CMIN	XMAX	CMAX
TABLEOID	XC_NODE_ID	XC_NODE_HASH	TABLEBUCKETID	GS_TUPLE_UID

1.3 数据类型

数据类型是数据的一个基本属性，用于区分不同类型的数据。不同的数据类型所占的存储空间不同，能够进行的操作也不相同。数据库中的数据存储在数据表中。数据表中的每一列都定义了数据类型，用户存储数据时，须遵从这些数据类型的属性，否则可能会出错。

GaussDB支持某些数据类型间的隐式转换，具体转换关系请参见**PG_CAST**。

1.3.1 数值类型

表1-2列出了所有的可用类型。数字操作符和相关内置函数请参见**数字操作函数和操作符**。

版权所有 © 华为云计算技术有限公司

表 1-2 整数类型

名称	描述	存储空间	范围
TINYINT [UNSIGNED]	微整数，带符号别名为INT1，无符号别名为UINT1。UNSIGNED类型仅支持在M或MYSQL兼容模式中使用。	1字节	• 带符号的范围是0 ~ +255。 • 无符号的范围是0 ~ +255。 • 当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，带符号的范围是-128 ~ +127。
SMALLINT [UNSIGNED]	小范围整数，带符号别名为INT2，无符号别名为UINT2。UNSIGNED类型仅支持在M或MYSQL兼容模式中使用。	2字节	• 带符号的范围是-32,768 ~ +32,767。 • 无符号的范围是0 ~ +65,535。
MEDIUMINT [UNSIGNED]	中等范围整数，带符号别名为INT4，无符号别名为UINT4。UNSIGNED类型仅支持在M或MYSQL兼容模式中使用。	4字节	• 带符号的范围是-2,147,483,648 ~ +2,147,483,647。 • 无符号的范围是0 ~ +4,294,967,295。
INTEGER [UNSIGNED]	常用的整数，带符号别名为INT4，无符号别名为UINT4。UNSIGNED类型仅支持在M或MYSQL兼容模式中使用。	4字节	• 带符号的范围是-2,147,483,648 ~ +2,147,483,647。 • 无符号的范围是0 ~ +4,294,967,295。
BINARY_INTEGER	常用的整数，带符号别名为INTEGER。	4字节	-2,147,483,648 ~ +2,147,483,647。
BIGINT [UNSIGNED]	大范围的整数，带符号别名为INT8，无符号别名为UINT8。UNSIGNED类型仅支持在M或MYSQL兼容模式中使用。	8字节	• 带符号的范围是-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807。 • 无符号的范围是0 ~ +18,446,744,073,709,551,615。
INT16 [SIGNED]	十六字节的大范围整数，目前不支持用户用于建表等使用。	16字节	-170,141,183,460,469,231,731,687,303,715,884,105,728 ~ +170,141,183,460,469,231,731,687,303,715,884,105,727。

版权所有 © 华为云计算技术有限公司

说明

- 当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，整型支持设置显示宽度和指定ZEROFILL属性。
- 显示宽度并不限制列所能存储的值的范围，也不影响超出显示宽度的值的正确显示。例如，定义为SMALLINT(3)的列可存储SMALLINT类型的完整范围（-32768至32767），即使某些值的位数超过三位，也会完整显示，而不会被截断或改变。
- 当显示宽度与ZEROFILL属性配合使用时，显示时会在数值的左侧填充零，直到其总长度达到指定的显示宽度。例如，对于定义为INT(4) ZEROFILL的列，插入值为5的数据，查询结果将显示为0005。
- 如果为数字列指定ZEROFILL，将自动添加UNSIGNED属性。
- 如果不为数字列指定ZEROFILL，仅指明宽度信息，则宽度信息在表结构描述中不显示。
- 如果指定UNSIGNED，将不允许使用负值。
- TINYINT、SMALLINT、INTEGER、BIGINT和INT16类型存储各种范围的数字，即整数。如果存储超出范围以外的数值将会导致错误。
- INTEGER是最常用的类型，因为它提供了在范围、存储空间、性能之间的最佳平衡。一般只有取值范围确定不超过SMALLINT的情况下，才会使用SMALLINT类型。而只有在INTEGER的范围不够的时候才使用BIGINT，因为INTEGER的处理速度相对快得多。
- 当在整数值（其中一个是UNSIGNED类型）之间使用减号、加号、乘号时，结果是无符号。
- INT1/UINT1/UINT2/UINT4/UINT8的+、-、*运算允许返回值超过类型本身范围，INT2/INT4/INT8的+、-、*运算不允许返回值超过类型本身范围。
- UNSIGNED类型不支持与SET数据类型间的转换，避免对UNSIGNED类型与SET类型计算或比较。
- 当sql_compatibility = 'MYSQL'时，非数值类型的字符输入，会自动截断或返回数值0。

示例：

```
--创建数据库。
gaussdb=# CREATE DATABASE b_database dbcompatibility = 'MYSQL';
gaussdb=# \c b_database

--创建具有TINYINT类型数据的表。
b_database=# CREATE TABLE int_type_t1
        (
         IT_COL1 TINYINT,
         IT_COL2 TINYINT UNSIGNED
        );

--插入数据。
b_database=# INSERT INTO int_type_t1 VALUES(10,20);

--查看数据。
b_database=# SELECT * FROM int_type_t1;
it_col1 | it_col2
--------+---------
     10 |      20
(1 row)

--删除表。
b_database=# DROP TABLE int_type_t1;

--创建具有TINYINT,INTEGER,BIGINT类型数据的表。
b_database=# CREATE TABLE int_type_t2
(
    a TINYINT,
    b TINYINT,
    c INTEGER,
    d INTEGER UNSIGNED,
    e BIGINT,
    f BIGINT UNSIGNED
);
```


版权所有 © 华为云计算技术有限公司

```
--插入数据。
b_database=# INSERT INTO int_type_t2 VALUES(100, 10, 1000, 10000, 200, 2000);

--查看数据。
b_database=# SELECT * FROM int_type_t2;
  a  | b  |  c   |   d    | e  |  f
-----+----+------+--------+----+------
 100 | 10 | 1000 |  10000 | 200 | 2000
(1 row)

--删除表。
b_database=# DROP TABLE int_type_t2;

--删除数据库。
b_database=# \c postgres
gaussdb=# DROP DATABASE b_database;
```

表 1-3 任意精度类型

名称	描述	存储空间	范围
NUMERIC[(p[,s])], DECIMAL[(p[,s])]	• 精度p取值范围为[1,1000]，标度s取值范围为[0,p]。 • 当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，在未指定精度和标度的情况下，默认精度p为10，标度s为0。numeric不带括号时，只在设置表列数据类型的时候指定为numeric(10,0)，其他场景默认numeric类型按原来范围使用。 **说明** • p为总位数，s为小数位数。 • 当插入数据小数位数超过定义标度s时，四舍五入到定义标度。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	未指定精度的情况下，小数点前最大131,072位，小数点后最大16,383位。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间	范围
NUMBER[(p[,s])]	NUMERIC类型的别名。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	未指定精度的情况下，小数点前最大131,072位，小数点后最大16,383位。

示例：

```
--创建表。
gaussdb=# CREATE TABLE decimal_type_t1
(
    DT_COL1 DECIMAL(10,4)
);

--插入数据。
gaussdb=# INSERT INTO decimal_type_t1 VALUES(123456.122331);

--查询表中的数据。
gaussdb=# SELECT * FROM decimal_type_t1;
   dt_col1
-------------
 123456.1223
(1 row)

--删除表。
gaussdb=# DROP TABLE decimal_type_t1;

--创建表。
gaussdb=# CREATE TABLE numeric_type_t1
(
    NT_COL1 NUMERIC(10,4)
);

--插入数据。
gaussdb=# INSERT INTO numeric_type_t1 VALUES(123456.12354);

--查询表中的数据。
gaussdb=# SELECT * FROM numeric_type_t1;
   nt_col1
-------------
 123456.1235
(1 row)

--插入数据。
gaussdb=# INSERT INTO numeric_type_t1 VALUES(10000.12355);

--查询表中的数据。
gaussdb=# SELECT * FROM numeric_type_t1;
   nt_col1
-------------
 123456.1235
  10000.1236
(2 rows)

--删除表。
gaussdb=# DROP TABLE numeric_type_t1;
```


版权所有 © 华为云计算技术有限公司

说明

- 与整数类型相比，任意精度类型需要更大的存储空间，其存储效率、运算效率以及压缩比效果都要差一些。在进行数值类型定义时，优先选择整数类型。当数值超出整数可表示最大范围时，再选用任意精度类型。
- 使用NUMERIC/DECIMAL进行列定义时，建议指定该列的精度p以及标度s。
- 当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，NUMERIC[(p[,s])]、DECIMAL[(p[,s])]类型的输入输出范围规格有差异。
  - 输入格式：支持数字或数字字符串输入格式。
    - sql_mode参数值包含“strict_trans_tables”时，非法输入或者输入超过范围时，会报错error。
    - sql_mode参数值不包含“strict_trans_tables”时，非法输入或者输入超过范围时，会报warning信息，并返回0值或截断后的值。
  - 输出格式：DECIMAL[(p[,s])]类型当没有指定精度和标度时与NUMERIC[(p[,s])]没有指定精度和标度时的部分输出格式存在差异。如NUMERIC无精度和标度时在类型转换时会保留小数部分，参照未设置参数的场景的输出格式，DECIMAL无精度和标度时在类型转换时则会按照精度为10，标度为0转换。

表 1-4 序列整型

名称	描述	存储空间	范围
SMALLSERIAL	二字节序列整型。	2字节。	-32,768 ~ +32,767。
SERIAL	四字节序列整型。	4字节。	-2,147,483,648 ~ +2,147,483,647。
BIGSERIAL	八字节序列整型。	8字节。	-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807。

示例：

```
--创建表。
gaussdb=# CREATE TABLE smallserial_type_tab(a SMALLSERIAL);

--插入数据。
gaussdb=# INSERT INTO smallserial_type_tab VALUES(default);

--再次插入数据。
gaussdb=# INSERT INTO smallserial_type_tab VALUES(default);

--查看数据。
gaussdb=# SELECT * FROM smallserial_type_tab;
 a
---
 1
 2
(2 rows)

--创建表。
gaussdb=# CREATE TABLE serial_type_tab(b SERIAL);

--插入数据。
gaussdb=# INSERT INTO serial_type_tab VALUES(default);

--再次插入数据。
gaussdb=# INSERT INTO serial_type_tab VALUES(default);
```


版权所有 © 华为云计算技术有限公司

```
--查看数据。
gaussdb=# SELECT * FROM serial_type_tab;
 b
---
 1
 2
(2 rows)

--创建表。
gaussdb=# CREATE TABLE bigserial_type_tab(c BIGSERIAL);

--插入数据。
gaussdb=# INSERT INTO bigserial_type_tab VALUES(default);

--再次插入数据。
gaussdb=# INSERT INTO bigserial_type_tab VALUES(default);

--查看数据。
gaussdb=# SELECT * FROM bigserial_type_tab;
 c
---
 1
 2
(2 rows)

--删除表。
gaussdb=# DROP TABLE smallserial_type_tab;

gaussdb=# DROP TABLE serial_type_tab;

gaussdb=# DROP TABLE bigserial_type_tab;
```

说明

SMALLSERIAL、SERIAL和BIGSERIAL类型不是真正的类型，只是为在表中设置唯一标识做的概念上的便利。因此，创建一个整数字段，设置其缺省数值从一个序列发生器读取。使用NOT NULL约束以确保NULL不会被插入。在多数情况下，用户可能还希望附加UNIQUE或PRIMARY KEY约束，避免插入重复的数值。最后，序列发生器将从属于该字段，当该字段或表被删除时，序列发生器会同时删除。目前只支持在创建表时候指定SERIAL列，不可以在已有的表中增加SERIAL列。另外，临时表不支持创建SERIAL列，因为SERIAL不是真正的类型，所以不可以将表中存在的列类型转化为SERIAL。

浮点类型如**表1-5**所示。

说明

- REAL、FLOAT、DEC[(p[,s])]数据类型，在sql_compatibility = 'MYSQL'时，设置b_format_version = '5.7'和b_format_dev_version = 's1'后的行为见**表1-5**具体描述。
- **表1-5**中描述的p为精度，表示整数位最低可以接受的总位数；s为小数位位数。

表 1-5 浮点类型

名称	描述	存储空间	范围
REAL, FLOAT4	单精度浮点数，存在精度误差。 REAL数据类型在满足**说明**的场景下，映射为双精度浮点数FLOAT8，使用场景参考FLOAT8。	4字节。	-3.402E+38~+3.402E+38，6位十进制数字精度。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间	范围
REAL(p,s)	精度p取值范围为[1,1000]，标度s取值范围为[0,p]。 未指定精度和标度的情况下，默认精度p为10，标度s为0。 该类型映射为NUMERIC，使用场景参考NUMERIC。 该类型需要在sql_compatibility = 'MYSQL'时，设置b_format_version = '5.7'和b_format_dev_version = 's1'后才能使用。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	在精度和标度指定最大的情况下，小数点前最大131,072位，小数点后最大16,383位。
DOUBLE PRECISION, FLOAT8	双精度浮点数，存在精度误差。	8字节。	-1.79E+308~+1.79E+308，15位十进制数字精度。
DOUBLE	双精度浮点数，存在精度误差。 该类型映射为双精度浮点数FLOAT8，使用场景参考FLOAT8。 该类型需要在sql_compatibility = 'MYSQL'时，设置b_format_version = '5.7'和b_format_dev_version = 's1'后才能使用。	8字节。	-1.79E+308~+1.79E+308，15位十进制数字精度。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间	范围
DOUBLE(p, s)	精度p取值范围为[1,1000]，标度s取值范围为[0,p]。 未指定精度和标度的情况下，默认精度p为10，标度s为0。 该类型映射为NUMERIC，使用场景参考NUMERIC。 该类型需要在sql_compatibility = 'MYSQL'时，设置b_format_version = '5.7'和b_format_dev_version = 's1'后才能使用。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	在精度和标度指定最大的情况下，小数点前最大131,072位，小数点后最大16,383位。
FLOAT[(p)]	浮点数，存在精度误差。精度p取值范围为[1,53]。 根据精度p不同选择，p<=24时映射为REAL类型，p>24时映射为DOUBLE PRECISION类型，不指定精度时，映射为DOUBLE PRECISION类型。 在满足**说明**中的场景下，精度p<=24时映射为FLOAT4类型，p>24时映射为DOUBLE PRECISION类型，不指定精度时，映射为FLOAT4类型。	4字节或8字节。	-

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间	范围
FLOAT(p,s)	精度p取值范围为[1,1000]，标度s取值范围为[0,p]。 在未指定精度和标度的情况下，默认精度p为10，标度s为0。 该类型映射为NUMERIC，使用场景参考NUMERIC。 该类型需要在sql_compatibility = 'MYSQL'时，设置b_format_version = '5.7'和b_format_dev_version = 's1'后才能使用。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	在精度和标度指定最大的情况下，小数点前最大131,072位，小数点后最大16,383位。
BINARY_DOUBLE	是DOUBLE PRECISION的别名，为兼容Oracle数据库类型。	8字节。	-1.79E+308~+1.79E+308，15位十进制数字精度。
DEC[(p[,s])]	精度p取值范围为[1,1000]，标度s取值范围为[0,p]。 在满足**说明**中的场景且未指定精度和标度的情况下，默认精度p为10，标度s为0。 该类型映射为NUMERIC，使用场景参考NUMERIC。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	在精度和标度指定最大的情况下，小数点前最大131,072位，小数点后最大16,383位。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间	范围
INTEGER[(p[,s])]	精度p取值范围为[1,1000]，标度s取值范围为[0,p]。 在未指定精度和标度的情况下，默认精度p为10，标度s为0。 未指定精度和标度的情况下，该类型映射为INTEGER。指定精度和标度的情况下，该类型映射为NUMERIC。	用户声明精度。每四位（十进制位）占用两个字节，然后在整个数据上加上八个字节的额外开销。	在精度和标度指定最大的情况下，小数点前最大131,072位，小数点后最大16,383位。 未指定精度和标度的情况下，范围是-2,147,483,648 ~ +2,147,483,647。

说明

- 二进制浮点数据类型REAL、FLOAT4、DOUBLE、DOUBLE PRECISION、FLOAT8、FLOAT[(p)]和BINARY_DOUBLE为不精确的数值类型，其内部存储为近似值，因此存储和检索时可能会显示轻微的差异。当用户在使用二进制浮点数据类型时需要注意以下几点：
  - 精确存储和计算：如果需要精确存储和计算（例如货币金额），请改用精确的数据类型（例如numeric）。
  - 复杂计算：若使用不精确的数据类型执行复杂计算以获得重要数据，需要仔细评估其结果。
  - 浮点数比较：比较两个浮点数是否相等的结果可能与预期存在差异。
  - 下溢错误：如果一个浮点数过于接近零，反而无法精确表示，会导致下溢错误。
- 关于浮点类型的精度，目前只能保证直接读取时的精度位数。涉及分布式计算时，由于计算执行在各个DN节点上，并且最终汇聚到一个CN节点，因此误差可能会随计算节点数量增加而被放大。
- 当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，FLOAT[(p[,s])]、DOUBLE[(p[,s])]、DEC[(p[,s])]、REAL[(p[,s])]类型的输入输出范围规格有差异。
  - 输入格式：支持数字或数字字符串输入格式。
    - sql_mode参数值包含“strict_trans_tables”时，非法输入或者输入超过范围时，会报错error。
    - sql_mode参数值不包含“strict_trans_tables”时，非法输入或者输入超过范围时，会报warning信息，并返回0值或截断后的值。
  - 输出格式：DEC[(p[,s])]类型当没有指定精度和标度时与其映射的NUMERIC[(p[,s])]类型在没有指定精度和标度时的部分输出格式场景存在差异。如NUMERIC无精度和标度时在类型转换时会保留小数部分，参照未设置参数的场景的输出格式，DEC无精度和标度时在类型转换时则会按照精度为10，标度为0转换。

示例：

```
--创建表。
gaussdb=# CREATE TABLE float_type_t2
(
    FT_COL1 INTEGER,
    FT_COL2 FLOAT4,
    FT_COL3 FLOAT8,
    FT_COL4 FLOAT(3),
    FT_COL5 BINARY_DOUBLE,
```


版权所有 © 华为云计算技术有限公司

```
    FT_COL6 DECIMAL(10,4),
    FT_COL7 INTEGER(6,3)
)DISTRIBUTE BY HASH ( ft_col1);

--插入数据。
gaussdb=# INSERT INTO float_type_t2 VALUES(10,10.365456,123456.1234,10.3214, 321.321, 123.123654,
123.123654);

--查看数据。
gaussdb=# SELECT * FROM float_type_t2 ;
 ft_col1 | ft_col2 |   ft_col3    | ft_col4 | ft_col5 | ft_col6  | ft_col7
---------+---------+--------------+---------+---------+----------+---------
      10 | 10.3655 | 123456.1234 | 10.3214 | 321.321 | 123.1237 | 123.124
(1 row)

--删除表。
gaussdb=# DROP TABLE float_type_t2;

--示例：REAL(p,s)、FLOAT(p,s)、DOUBLE、DOUBLE(p,s)类型。
gaussdb=# CREATE DATABASE gaussdb_m  WITH dbcompatibility  'MYSQL';
gaussdb=# \c gaussdb_m
--设置兼容版本控制参数。
gaussdb_m=# SET b_format_version='5.7';
gaussdb_m=# SET b_format_dev_version='s1';
--创建表。
gaussdb_m=# CREATE TABLE t1(a real(10,2), b float(10,2), c double, d double(10,2));
--插入数据。
gaussdb_m=# INSERT INTO t1 VALUES (1000.12, 2000.23, 3000.34, 4000.45);
--查询表中数据。
gaussdb_m=# SELECT * FROM t1;
    a    |    b    |    c    |    d
---------+---------+---------+---------
 1000.12 | 2000.23 | 3000.34 | 4000.45
(1 row)

--删除表和数据库。
gaussdb_m=# DROP TABLE t1;
gaussdb_m=# \c postgres;
gaussdb=# DROP DATABASE gaussdb_m;

--重置参数。
gaussdb=# RESET ALL;
```

1.3.2 货币类型

货币类型用于存储带有固定小数精度的货币金额。

表1-6中显示的范围基于两位小数精度。该类型支持多种输入格式，包括整型、浮点型或者典型的货币格式（如“$1,000.00”）。根据区域字符集，输出一般是最后一种形式。

表 1-6 货币类型

名称	描述	存储空间	范围
money	货币金额。	8字节	-92233720368547758.08 ~ +92233720368547758.07

numeric、int和bigint类型的值可以直接转换为money类型。对于real和double precision类型，建议先转换为numeric类型，再转换为money类型，例如：

```
gaussdb=# SELECT '12.34'::float8::numeric::money;
 money
```


版权所有 © 华为云计算技术有限公司

```
--------
 $12.34
(1 row)
```

这种用法是不推荐的。由于浮点数存在精度丢失风险，因此不应该用来处理货币类型。

money类型的值可以转换为numeric类型而不丢失精度。转换为其他类型可能丢失精度，并且必须通过以下两步来完成：

```
gaussdb=# SELECT '52093.89'::money::numeric::float8;
  float8
----------
 52093.89
(1 row)
```

当两个money类型的值进行除法运算时，将返回double precision类型的结果（即一个纯数字，而不是money类型）；货币单位会在运算过程中相互抵消。

1.3.3 布尔类型

GaussDB支持的布尔类型如表1-7所示。

表 1-7 布尔类型

名称	描述	存储空间	取值
BOOLEAN	布尔类型。	1字节	• true：真 • false：假 • null：未知（unknown）

- “真”值的有效输入文本值是：
  TRUE、't'、'true'、'y'、'yes'、'1'、'TRUE'、true、'on'、以及所有非0数值。
- “假”值的有效输入文本值是：
  FALSE、'f'、'false'、'n'、'no'、'0'、0、'FALSE'、false、'off'。

查询结果为布尔类型时，仅显示t/f。

使用TRUE和FALSE是比较规范的用法（也是SQL兼容的用法）。

示例

```
--创建表。
gaussdb=# CREATE TABLE bool_type_t1
(
    BT_COL1 BOOLEAN,
    BT_COL2 TEXT
)DISTRIBUTE BY HASH(BT_COL2);

--插入数据。
gaussdb=# INSERT INTO bool_type_t1 VALUES (TRUE, 'sic est');

gaussdb=# INSERT INTO bool_type_t1 VALUES (FALSE, 'non est');

--查看数据。
gaussdb=# SELECT * FROM bool_type_t1;
 bt_col1 | bt_col2
---------+---------
```


版权所有 © 华为云计算技术有限公司

```
 t       | sic est
 f       | non est
(2 rows)

gaussdb=# SELECT * FROM bool_type_t1 WHERE bt_col1 = 't';
 bt_col1 | bt_col2
---------+---------
 t       | sic est
(1 row)

--删除表。
gaussdb=# DROP TABLE bool_type_t1;
```

1.3.4 字符类型

GaussDB支持的字符类型如**表1-8**所示。字符串操作符和相关的内置函数请参见**字符处理函数和操作符**。

表 1-8 字符类型

名称	描述	存储空间
CHAR(n) CHARACTER(n) NCHAR(n)	定长字符串，长度不足时补充空格。其中，n表示字节长度，若未指定精度n，则默认为1。	n最大为10485760（即10MB）。
	当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'、b_format_dev_version = 's1'后： ● 类型：n指字符长度，区间[0,10485760]，若未指定精度n，则默认为1。定长字符串，长度不足时补充空格。 ● 输入： – 作为表列和临时变量的数据类型：输入字符串字符个数在范围内可以正常输入，sql_mode参数值包含“strict_trans_tables”时，超过范围直接报错，否则按n最大字符长度截断处理并告警。 ● 输出： – 作为表列和临时变量的数据类型：sql_mode参数值包含“pad_char_to_full_length”时，输出带有尾部空格的字符串，否则输出不带尾部空格的字符串。 – 作为函数参数和返回值，存储过程的参数：不支持长度校验（例如：自定义函数入参为CHAR(5)类型，输入一个字符串'123456'，不校验长度，可以直接传入参数）。	最大为10485760个字符。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
	当sql_compatibility = 'ORA'时： ● CHAR(n)和CHARACTER(n)可以在长度参数n后面显式指定n的单位，仅支持指定byte或char作为单位，其中byte表示n的单位为字节，char表示n的单位为字符，默认为byte。 ● NCHAR(n)不支持指定长度参数n单位的用法。 ● 在单字节字符集下，即使指定char作为单位，n的单位仍为字节，例如在SQL_ASCII字符集下，指定CHAR(1 char)，输入一个中文字符时会报错。 ● 由于类型存储空间限制为10MB，因此当指定CHAR(n char)或CHARACTER(n char)时，并不代表类型实际可以容纳下n个字符。 – 触发存储空间限制时，类型实际可以容纳的字符数取决于当前字符集，例如在GBK字符集下，指定CHAR(10485760 char)时，实际最多可以容纳5242880个中文字符。 – 定长字符串补足空格时，将至多补至存储空间上限。	最大为10MB。
VARCHAR(n) CHARACTER VARYING(n)	变长字符串。PG兼容模式下，n表示字符长度。其他兼容模式下，n表示字节长度。	n最大为10MB。 不带n时，最大存储字节长度为1GB-85-4(存储长度参数的空间)-其余列长度，比如(a int, b varchar, c int)，varchar最大长度为1GB-85-4(存储长度参数的空间)-4(a列int的长度)-4(c列int的长度)=1,073,741,727。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
	当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'、b_format_dev_version = 's1'后： ● n指字符长度，区间[0,10485760]，若未指定精度n，默认不限制长度，长度与TEXT类型相同。 ● 输入： – 作为表列和临时变量的数据类型：输入字符串字符个数在范围内可以正常输入，sql_mode参数值包含“strict_trans_tables”时，超过范围直接报错，否则按n最大字符长度截断处理并告警。 ● 输出：原字符串输出。	n最大为10485760。 不带n时，最大存储字节长度为1GB-85-4(存储长度参数的空间)-其余列长度，比如(a int, b varchar, c int)，varchar最大长度为1GB-85-4(存储长度参数的空间)-4(a列int的长度)-4(c列int的长度)=1,073,741,727。
	当sql_compatibility = 'ORA'时： ● 可以在长度参数n后面显式指定n的单位，仅支持指定byte或char作为单位，其中byte表示n的单位为字节，char表示n的单位为字符，默认为byte。 ● 在单字节字符集下，即使指定char作为单位，n的单位仍为字节，例如在SQL_ASCII字符集下，指定VARCHAR(1 char)，插入一个中文字符时会报错。 ● 由于类型存储空间限制为10MB，因此当指定VARCHAR(n char)或CHARACTER VARYING(n char)时，并不代表类型实际可以容纳下n个字符。 触发存储空间限制时，类型实际可以容纳的字符数取决于当前字符集，例如在GBK字符集下，指定VARCHAR(10485760 char)时，实际最多可以容纳5242880个中文字符。	最大为10MB。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
VARCHAR2(n)	变长字符串。是VARCHAR(n)类型的别名。	n最大为10485760（即10MB）。 不带n时，最大存储字节长度为1GB-85-4(存储长度参数的空间)-其余列长度，比如(a int, b varchar2, c int)，varchar2最大长度为1GB-85-4(存储长度参数的空间)-4(a列int的长度)-4(c列int的长度)=1,073,741,727。
NVARCHAR2(n)	变长字符串。在SQL_ASCII字符集下，n表示的是字节；在非SQL_ASCII字符集下，n表示的是字符。	n最大为10MB。 不带n时，最大存储字节长度为1GB-85-4(存储长度参数的空间)-其余列长度，比如(a int, b nvarchar2, c int)，nvarchar2最大长度为1GB-85-4(存储长度参数的空间)-4(a列int的长度)-4(c列int的长度)=1,073,741,727。
TEXT	变长字符串。	最大存储字节长度为1GB-85-4(存储长度参数的空间)-其余列长度，比如(a int, b text, c int)，text最大长度为1GB-85-4(存储长度参数的空间)-4(a列int的长度)-4(c列int的长度)=1,073,741,727。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
CLOB	文本大对象。	最大为1GB-1，但还需要考虑到列描述头信息的大小，以及列所在元组的大小限制（也小于1GB-1），因此CLOB类型最大值可能小于1GB-1。
TINYTEXT MEDIUMTEXT LONGTEXT	当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'、b_format_dev_version = 's1'后可以使用这些类型。 类型转换成TEXT类型，使用场景与TEXT类型相同。	最大为1GB-1，但还需要考虑到列描述头信息的大小，以及列所在元组的大小限制（也小于1GB-1），因此TEXT类型最大值可能小于1GB-1。

说明

- 除了每列的大小限制以外，每个元组的总大小也不可超过1GB-1字节，主要受列的控制头信息、元组控制头信息以及元组中是否存在NULL字段等影响。
- NCHAR为bpchar类型的别名，VARCHAR2(n)为VARCHAR(n)类型的别名。
- 在ORA兼容模式下，默认将接收到的空字符串转换为null。

GaussDB还支持另外两种定长字符类型，如表1-9所示。name类型只用在内部系统表中，作为存储标识符，不建议普通用户使用。该类型长度当前定为64字节（63可用字符加结束符）。类型“char”只用了一个字节的存储空间，在系统内部主要用于系统表，主要作为简单化的枚举类型使用。

表 1-9 特殊字符类型

名称	描述	存储空间
name	用于对象名的内部类型。	64字节。
"char"	单字节内部类型。	1字节。

1.3.5 二进制类型

GaussDB支持的二进制类型如表1-10所示。

版权所有 © 华为云计算技术有限公司

表 1-10 二进制类型

名称	描述	存储空间
BLOB	二进制大对象。 目前BLOB支持的外部存取接口仅有以下几种： • DBE_LOB.GET_LENGTH • DBE_LOB.READ • DBE_LOB.WRITE • DBE_LOB.WRITE_APPEND • DBE_LOB.COPY • DBE_LOB.ERASE 这些接口的详细说明请参见**DBE_LOB**。	最大为1GB-1，但还需要考虑到列描述头信息的大小，以及列所在元组的大小限制（也小于1GB-1），因此BLOB类型最大值可能小于1GB-1。
	当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，BLOB类型映射为BYTEA类型，别名为BYTEA。	具体存储规格参考BYTEA类型。
TINYBLOB MEDIUMBLOB LONGBLOB	二进制大对象。	具体存储规格参考BYTEA类型。
	只有当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，可以使用此类型，类型映射为BYTEA类型，别名为BYTEA。	
RAW	变长的十六进制类型。	4字节加上实际的二进制字符串。最大为1GB-1，但还需要考虑到列描述头信息的大小，以及列所在元组的大小限制（也小于1GB-1），因此类型最大值可能小于1GB-1。
BYTEA	变长的二进制字符串。	4字节加上实际的二进制字符串。最大为1GB-1，但还需要考虑到列描述头信息的大小，以及列所在元组的大小限制（也小于1GB-1），因此类型最大值可能小于1GB-1。

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
BYTEAWITHOUTORDERWITHEQUALCOL	变长的二进制字符串（密态特性新增的类型，如果加密列的加密类型指定为确定性加密，则该列的实际类型为BYTEAWITHOUTORDERWITHEQUALCOL），元命令打印加密表将显示原始数据类型。	4字节加上实际的二进制字符串。最大为1GB减去53字节（即1073741771字节）。
BYTEAWITHOUTORDERCOL	变长的二进制字符串（密态特性新增的类型，如果加密列的加密类型指定为随机加密，则该列的实际类型为BYTEAWITHOUTORDERCOL），元命令打印加密表将显示原始数据类型。	4字节加上实际的二进制字符串。最大为1GB减去53字节（即1073741771字节）。
_BYTEAWITHOUTORDERWITHEQUALCOL	变长的二进制字符串，密态特性新增的类型。	4字节加上实际的二进制字符串。最大为1GB减去53字节（即1073741771字节）。
_BYTEAWITHOUTORDERCOL	变长的二进制字符串，密态特性新增的类型。	4字节加上实际的二进制字符串。最大为1GB减去53字节（即1073741771字节）。

说明

- 除了每列的大小限制外，每个元组的总大小也不可超过1GB-1字节。
- 不支持直接使用BYTEAWITHOUTORDERWITHEQUALCOL、BYTEAWITHOUTORDERCOL、_BYTEAWITHOUTORDERWITHEQUALCOL和_BYTEAWITHOUTORDERCOL类型创建表。
- RAW(n)中的n表示字节长度建议值，不会用于校验输入RAW类型的字节长度。
- 当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB类型的表现规格为BYTEA类型，如查询表结构显示为BYTEA类型。

示例:

```
--创建表。
gaussdb=# CREATE TABLE blob_type_t1
(
    BT_COL1 INTEGER,
    BT_COL2 BLOB,
    BT_COL3 RAW,
    BT_COL4 BYTEA
) DISTRIBUTE BY REPLICATION;

--插入数据。
gaussdb=# INSERT INTO blob_type_t1 VALUES(10,empty_blob(),
HEXTORAW('DEADBEEF'),E'\\xDEADBEEF');

--查询表中的数据。
gaussdb=# SELECT * FROM blob_type_t1;
```


版权所有 © 华为云计算技术有限公司

```
 bt_col1 | bt_col2 | bt_col3  |  bt_col4
---------+---------+----------+------------
      10 |         | DEADBEEF | \xdeadbeef
(1 row)

--删除表。
gaussdb=# DROP TABLE blob_type_t1;

--示例：TINYBLOB MEDIUMBLOB LONGBLOB二进制大对象类型。
gaussdb=# CREATE DATABASE gaussdb_m WITH dbcompatibility  'MYSQL';
gaussdb=# \c gaussdb_m
--设置兼容版本控制参数。
gaussdb_m=# SET b_format_version='5.7';
gaussdb_m=# SET b_format_dev_version='s1';
--创建表。
gaussdb_m=# CREATE TABLE t1(num int, a tinyblob, b blob, m mediumblob, l longblob);
--插入数据。
gaussdb_m=# INSERT INTO t1 VALUES (1, 'tinyblobtest', 'blobtest', 'mediumblobtest', 'longblobtest');
--查询表中数据。
gaussdb_m=# SELECT * FROM t1;
 num |      a       |    b     |       m        |      l
-----+--------------+----------+----------------+--------------
   1 | tinyblobtest | blobtest | mediumblobtest | longblobtest
(1 row)

--删除表和数据库。
gaussdb_m=# DROP TABLE t1;
gaussdb_m=# \c postgres;
gaussdb=# DROP DATABASE gaussdb_m;
--重置参数。
gaussdb=# RESET ALL;
```

1.3.6 日期/时间类型

GaussDB支持的日期/时间类型请参见**表1-11**。该类型的操作符和内置函数请参见**时间和日期处理函数和操作符**。

说明

- 如果其他的数据库时间格式和GaussDB的时间格式不一致，可通过修改配置参数DateStyle的值来保持一致。
- 当前日期表示法采用ISO标准，年由4位数组成，以公历公元1年为0001年，以公元前1年为0000年，公元前2年为-0001年，其他以此类推。
- 在ORA兼容性下，DATE类型在GUC参数mapping_date_to_datea取值为off时，使用TIMESTAMP(0) WITHOUT TIME ZONE代替，而该GUC参数取值为on时，使用DATEA类型代替。两类型间主要差异总结如下：
  - DATEA类型和TIMESTAMP(0)类型的OID不同，DATEA类型为GaussDB内部实现的一套全新数据类型，有对应配套的操作符、转换函数等。
  - DATEA类型和TIMESTAMP(0)类型支持的范围不同。
  - 输入DATEA类型时，需要对字符串进行格式化校验，只有满足nls_date_format格式才可以转换为DATEA类型，而TIMESTAMP(0)类型则仅支持几种固定格式的输入。DATEA类型不支持特殊值输入，如：NOW、EPOCH、TODAY等。
  - 在驱动使用方面，TIMESTAMP(0)类型的元数据（包含OID、语言数据类型等）和DATEA类型的元数据不同。在JDBC中设置预编译参数时，对于TIMESTAMP(0)类型，需要使用SetTimestamp或者GetTimestamp；对于DATEA类型，则需要使用SetDate或者GetDate。
  - 在COPY导入数据时，可以设置WITH选项，设置导入数据时按照何种格式解析。对于TIMESTAMP(0)类型，需要使用TIMESTAMP_FORMAT指定；对于DATEA类型，则需要使用DATEA_FORMAT指定。具体请参见**语法格式**。

版权所有 © 华为云计算技术有限公司

表 1-11 日期/时间类型

名称	描述	存储空间
DATE	日期和时间。 最小值：公元前4713年，4714-11-24 BC。最大值：公元5874897年，5874897-12-31AD。 在设置sql_compatibility = 'MYSQL'、b_format_version = '5.7'和b_format_dev_version = 's1'三个参数后，输入和输出格式、范围的规格有差异： ● 输入格式：支持有分隔符YYYY-MM-DD和无分隔符YYYYMMDD的输入格式。 – 有分隔符场景下，日期的年、月、日之间不支持“+”、“:”作为分隔符（某些分隔符混合使用场景下不支持，规格继承已有实现。如：date '2020-12?12'），输入格式不受DateStyle参数的影响。 – 无分隔符场景下，输入不完整时可能不会报错，但是输出结果不一定正确。 ● 输出格式：仅支持YYYY-MM-DD，且不受DateStyle参数的影响。 ● 范围：4713-01-01 BC ~ 5874897-12-31 AD。 – sql_mode参数值包含“strict_trans_tables”时，非法输入或者超过范围，会报错。 – sql_mode参数值不包含“strict_trans_tables”时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值，如'0000-00-10' 转换为：0002-12-10 BC。 非法输入或者超过范围时，会报warning信息，并返回0000-00-00 00:00:00值。	4字节（兼容模式ORA下存储空间大小为8字节）

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
	说明 ORA兼容性下，mapping_date_to_datea开关关闭时，数据类型DATE会被替换为TIMESTAMP(0) WITHOUT TIME ZONE。 ORA兼容性下，mapping_date_to_datea开关打开时，数据类型DATE会被替换为DATEA。	

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
TIME [(p)] [WITHOUT TIME ZONE]	用于表示一天内的时间，不带时区。 p表示小数点后的精度，取值范围为0~6。 最小值：00:00:00。最大值：24:00:00。 在设置sql_compatibility = 'MYSQL'、b_format_version = '5.7'和b_format_dev_version = 's1'三个参数后，不限定于表示一天内的时间，如可以表示持续时间、相对时间等概念，且格式、范围、精度的规格有差异： ● 输入格式：支持有分隔符[D] hh:mi:ss.ff和无分隔符 hhmiss.ff的输入格式。D是表示天数，整型数（支持负数），可选。输出时会按照1天24小时的关系换算成小时数，加hh后显示。只支持冒号“:”作为分隔符。 ● 输出格式：hh:mi:ss.ff。在显示的时候会自动忽略小数部分末尾的所有零。 ● 范围：-838:59:59.000000 ~ 838:59:59.000000。 – sql_mode参数值包含“strict_trans_tables”时，非法输入返回00:00:00，超过范围会报错。 – sql_mode参数值不包含“strict_trans_tables”时，非法输入会返回 00:00:00；合法输入但是超过范围，会返回最近的边界值。如：838:59:59.000001会返回838:59:59，而-838:59:59.000001会返回-838:59:59。 ● 精度：p表示小数点后的精度，取值范围为0~6。当p设置超过6时，按照p为6进行处理。 – 作为表列的数据类型时，缺省精度为0。 – 作为表达式使用（如time '10:10:10.123456'）时，缺省的精度为6。	8字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
TIME [(p)] [WITH TIME ZONE]	用于表示一天内的时间，携带时区。 p表示小数点后的精度，取值范围为0~6。 最小值：00:00:00+15:59。最大值：24:00:00-15:59。	12字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
TIMESTAMP[(p)] [WITHOUT TIME ZONE]	日期和时间，不携带时区。 p表示小数点后的精度，取值范围为0~6。 最小值：公元前4713年，4714-11-24 BC 00:00:00.000000。最大值：公元294277年，294277-01-09 AD 04:00:54.775807，最大值等效infinity。 在设置sql_compatibility = 'MYSQL'、b_format_version = '5.7'和b_format_dev_version = 's1'三个参数后，会被替换为TIMESTAMP[(p)] [WITH TIME ZONE]类型，但格式、范围、精度的规格有差异： ● 输入格式：支持有分隔符YYYY-MM-DD hh:mi:ss.ff+timezone和无分隔符YYYYMMDDhhmiss.ff的输入格式。 – 有分隔符的情况下，日期年、月、日之间不支持“+”、“:”作为分隔符（某些分隔符混合使用场景下不支持，规格继承已有实现。如：timestamp '2020-12?12 00:00:00'），时间部分只支持冒号“:”作为分隔符。输入格式不受DateStyle参数的影响。 – 无分隔符且输入不完整的场景下，可能不会报错，但是输出结果不一定正确。 – 年份只有2位数时，00~69对应范围为2000~2069；70~99对应范围为1970~1999。 – 年份只有1位数时，如：timestamp '1-1-1 00:00:00' 结果为'0001-01-01 00:00:00'。 ● 输出格式：仅支持YYYY-MM-DD hh:mi:ss.ff格式，且不受DateStyle参数的影响。在显示的时候会自动忽略时间部分小数部分末尾的所有零。 ● 范围： – sql_mode参数值包含“strict_trans_tables”时，范围与TIMESTAMP[(p)] WITH TIME ZONE一致；非法输入或者超过范围时，会报错。	8字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
	– sql_mode参数值不包含“strict_trans_tables”时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值，如'0000-00-10 00:00:00'转换为：0002-12-10 00:00:00 BC。非法输入或者超过范围时，会报warning信息，并返回0000-00-00 00:00:00值。 ● 精度：p表示小数点后的精度，取值范围为0~6。当p设置超过6时，按照p为6进行处理。 – 作为表列的数据类型时，缺省精度为0。 – 作为表达式使用（如timestamp '2000-01-01 00:00:00.123456'）时，缺省的精度为6。 ● 时区：支持的范围：-15:59~+15:59，或使用时区名称（请参考系统视图PG_TIMEZONE_NAMES），与TIMESTAMP[(p)] WITH TIME ZONE类型的时区规格一致。将指定时区的时间值转换为UTC-0时区的时间。显示时，再将UTC-0时区的时间转换为当前服务器时区（SHOW TIME ZONE可以查看服务器时区）的时间。因此，通过SET TIME ZONE语句更改服务器时区时，显示的结果也会变更。	

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
TIMESTAMP[(p)] [WITH TIME ZONE]	日期和时间，带时区。TIMESTAMP的别名为TIMESTAMPTZ。 p表示小数点后的精度，取值范围为0~6。 最小值：公元前4713年，4714-11-24 BC 00:00:00.000000 +0。最大值：公元294277年，294277-01-09 AD 04:00:54.775807 +0，最大值等效infinity。需要注意的是，输入的原始日期时间、将其转换成UTC-0时区的日期时间以及转换成当前服务器时区的日期时间，三者均不能大于294277-01-09 AD 04:00:54.775807或小于4714-11-24 BC 00:00:00.000000，否则在使用过程中有可能出现日期时间超出范围的报错。 时区更新：部分国家或地区因为政治、经济、战争等因素经常会更新时区信息，数据库系统也因此需要同步修改时区文件以确保时间内容的正确性。 GaussDB时区类型目前只涉及timestamp with timezone，当新的时区文件生效时，不会对已有的数据进行变更，新数据会随时区文件信息进行同步调整。	8字节
SMALLDATETIME	日期和时间，不带时区。 精确到分钟，秒位大于等于30秒进一位。 最小值：公元前4713年，4714-11-24 BC 00:00:00。最大值：公元294277年，294277-01-09 AD 04:00:00。	8字节
INTERVAL DAY (l) TO SECOND (p)	时间间隔，X天X小时X分X秒。 • l：天数的精度，取值范围为0~6。兼容性考虑，目前未实现具体功能。 • p：秒数的精度，取值范围为0~6。小数末尾的零不显示。	16字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
INTERVAL [FIELDS] [(p)]	时间间隔。 ● FIELDS：可以是YEAR，MONTH，DAY，HOUR，MINUTE，SECOND，DAY TO HOUR，DAY TO MINUTE，DAY TO SECOND，HOUR TO MINUTE，HOUR TO SECOND，MINUTE TO SECOND。 ● p：秒数的精度，取值范围为0~6，且fields为SECOND，DAY TO SECOND，HOUR TO SECOND或MINUTE TO SECOND时，参数p才有效。小数末尾的零不显示。	12字节
reltime	相对时间间隔。 ● 格式为：X years X mons X days XX:XX:XX。 ● 采用儒略历计时，规定一年为365.25天，一个月为30天，计算输入值对应的相对时间间隔。	4字节
abstime	日期和时间。 ● 格式为：YYYY-MM-DD hh:mi:ss +timezone。 ● 取值范围为当前时区的1901-12-13 20:45:53~2038-01-18 23:59:59，精度为秒。插入超出范围的数据会转换为invalid特殊值存储。	4字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
datetime[(p)]	日期和时间。 仅在设置sql_compatibility = 'MYSQL'、b_format_version = '5.7'以及b_format_dev_version = 's1'三个参数后生效，datetime[(p)]类型会被替换为TIMESTAMP[(p)] WITHOUT TIME ZONE类型，但格式、范围、精度、时区处理的规格有差异： ● 输入格式：支持有分隔符YYYY-MM-DD hh:mi:ss.ff+timezone和无分隔符YYYYMMDDhhmiss.ff的输入格式。 – 有分隔符的情况下，日期年、月、日之间不支持“+”、“:”作为分隔符（某些分隔符混合使用场景下不支持，规格继承已有实现。如：datetime '2020-12?12 00:00:00'），时间部分只支持冒号“:”作为分隔符。输入格式不受DateStyle参数的影响。 – 无分隔符且输入不完整的场景下，可能不会报错，但是输出结果不一定正确。 – 年份只有2位数时，00-69对应范围为2000-2069；70-99对应范围为：1970-1999。 – 年份只有1位数时，如：datetime '1-1-1 00:00:00' 结果为'0001-01-01 00:00:00'。 ● 输出格式：仅支持YYYY-MM-DD hh:mi:ss.ff格式，且不受DateStyle参数的影响。在显示的时候会自动忽略时间部分小数部分末尾的所有零。 ● 范围： – sql_mode参数值包含“strict_trans_tables”时，范围与TIMESTAMP[(p)] WITHOUT TIME ZONE一致，非法输入或者超过范围时，会报错。 – sql_mode参数值不包含“strict_trans_tables”时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依	8字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
	次转换为合法的值，如'0000-00-10 00:00:00'转换为：0002-12-10 00:00:00 BC。非法输入或者超过范围时，会报warning信息，并返回0000-00-00 00:00:00值。 ● 精度：p表示小数点后的精度，取值范围为0~6。给定精度超过6时，按照6处理。 – 作为表列的数据类型时，缺省精度为0。 – 作为表达式使用（如：datetime '2000-01-01 00:00:00.123456'）时，缺省的精度为6。 ● 时区：支持的范围：-15:59 ~ +15:59，或使用时区名称（请参考系统视图PG_TIMEZONE_NAMES），与TIMESTAMP[(p)] WITH TIME ZONE类型的时区规格一致。将指定时区的时间值转换为当前服务器时区（SHOW TIME ZONE可以查看服务器时区）的时间。如果将转换后的日期时间存入表中，此值不会再随服务器时区的变更而再做时区转换。	

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间
year[(w)]	年份。 仅在设置sql_compatibility = 'MYSQL'参数后生效。 ● 输入格式： – 4位纯数字字符串：'1901'~'2155'，表示从1901年到2155年。 – 4位数字：1901~2155，表示从1901年到2155年。 – 1位或者2位的纯数字字符串：'0'~'99'，其中，'0'~'69'表示从2000年到2069年，'70'~'99'表示为从1970年到1999年。'0'、'00'表示为2000年。 – 1位或者2位的数字：0~99，其中，1~69表示从2001年到2069年，70~99表示为从1970年到1999年。0表示为0000。 – 其他时间函数的返回值，如now()。 ● 输出格式：仅支持YYYY格式。w表示输出格式的位数，只支持4位。若不指定，默认为4位。 ● 范围：1901~2155。 – sql_mode参数值包含“strict_trans_tables”时，非法输入或者超过范围，会报错。 – sql_mode参数值不包含“strict_trans_tables”时，会将非法值转为0000插入，并提示告警。	1字节

版权所有 © 华为云计算技术有限公司

名称	描述	存储空间

DATEA	日期和时间。
	仅在设置sql_compatibility = 'ORA'参数后生效。
	最小值：公元前4712年1月1日0时0分0秒。
	最大值：公元9999年12月31日23时59分59秒。
	输入格式：输入格式必须符合参数nls_date_format设置的时间格式，nls_date_format支持的格式请参考**表1-12**。
	• 支持'-'、'/'、','、'.'、'%'、':'或用""包含的字符串作为分隔符。
	• 无分隔符且输入不完整的场景下，可能不会报错，但是输出结果不一定正确。
	• 若输入格式包含YY时，DATEA优先尝试两位年份解析，失败后尝试四位年份解析，最终错误信息以四位年份解析失败返回的信息为准。
	输出格式：仅支持YYYY-MM-DD HH24:MI:SS AD/BC的时间格式，且不受DateStyle参数的影响。
	范围：公元前4712年1月1日0时0分0秒 ~ 公元9999年12月31日23时59分59秒， 非法输入或输入超过范围，会报错。
	精度：不保留小数点后精度，最小精度为秒。
	说明
	• 在ORA兼容性下，mapping_date_to_datea开关开启时，用于替换DATE类型。
	• DATEA类型用于内部功能实现，不建议用户直接使用。
	• DATEA类型不支持作为自定义range类型。
	• 输入格式若不满足nls_date_format设置的时间格式可能产生以下结果：
	  • 出现报错提示，提示输入的格式不符合要求。
	  • 输入成功，但输入的时间与预期结果不符。
		8字节

版权所有 © 华为云计算技术有限公司

说明

- 时间类型的数据在显示的时候会自动忽略末尾的所有零。
- 精度p默认取值为6。
- 对于INTERVAL类型，日期和时间在系统内部分别用int32和double类型存储，所以两者的取值范围和对应数据类型的取值范围一致。
- 插入时间超出范围的时候，系统可能不报错，但不保证行为正常。
- 在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，缺省的DATE值由以下确定：
  - 年通过SYSDATE返回当年。
  - 月通过SYSDATE返回当月。
  - 日返回01（月份的第一天）。
  - 小时，分钟，秒都是0。

示例：

```
--创建表。
gaussdb=# CREATE TABLE date_type_tab(coll date);

--插入数据。
gaussdb=# INSERT INTO date_type_tab VALUES (date '12-10-2010');

--查看数据。
gaussdb=# SELECT * FROM date_type_tab;
    coll
------------
  2010-12-10
(1 row)

--删除表。
gaussdb=# DROP TABLE date_type_tab;

--创建表。
gaussdb=# CREATE TABLE time_type_tab (da time without time zone ,dai time with time zone,dfgh
timestamp without time zone,dfga timestamp with time zone, vbg smalldatetime);

--插入数据。
gaussdb=# INSERT INTO time_type_tab VALUES ('21:21:21','21:21:21 pst','2010-12-12','2013-12-11
pst','2003-04-12 04:05:06');

--查看数据。
gaussdb=# SELECT * FROM time_type_tab;
    da    |     dai     |        dfgh         |          dfga          |         vbg
----------+-------------+---------------------+------------------------+---------------------
 21:21:21 | 21:21:21-08 | 2010-12-12 00:00:00 | 2013-12-11 16:00:00+08 | 2003-04-12 04:05:00
(1 row)

--删除表。
gaussdb=# DROP TABLE time_type_tab;

--创建表。
gaussdb=# CREATE TABLE day_type_tab (a int,b INTERVAL DAY(3) TO SECOND (4));

--插入数据。
gaussdb=# INSERT INTO day_type_tab VALUES (1, INTERVAL '3' DAY);

--查看数据。
gaussdb=# SELECT * FROM day_type_tab;
 a |   b
---+--------
 1 | 3 days
(1 row)

--删除表。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE day_type_tab;

--创建表。
gaussdb=# CREATE TABLE year_type_tab(a int, b interval year (6));

--插入数据。
gaussdb=# INSERT INTO year_type_tab VALUES(1,interval '2' year);

--查看数据。
gaussdb=# SELECT * FROM year_type_tab;
 a |     b
---+---------
 1 | 2 years
(1 row)

gaussdb=# SELECT TIME 'allballs';
   time
----------
 00:00:00
(1 row)

--删除表。
gaussdb=# DROP TABLE year_type_tab;

-- ORA兼容模式下，当mapping_date_to_datea=on时，datea数据类型示例。
-- 创建ORA兼容模式下的数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
CREATE DATABASE

-- 切换到ORA兼容模式下的数据库。
gaussdb=# \c ora_db

-- 查看参数mapping_date_to_datea。
ora_db=# SHOW mapping_date_to_datea;
 mapping_date_to_datea
-----------------------
 on
(1 row)

-- 创建表。
ora_db=# CREATE TABLE datea_type_tab(a int, b date);

-- 设置nls_date_format参数。
ora_db=# SET nls_date_format='YYMMDD HH:MI:SS';
SET

-- 插入数据。
ora_db=# INSERT INTO datea_type_tab VALUES(1, '20210512 05:52:23'::date); --YY按照4位年份解析。
INSERT 0 1
ora_db=# INSERT INTO datea_type_tab VALUES(2, '201012 11:12:13'::date); --YY按照2位年份解析。
INSERT 0 1

ora_db=# INSERT INTO datea_type_tab VALUES(3, '251126 00:00:00'::date); --最终错误信息来自YY按照四位年
份解析失败时产生的信息。
ERROR:  The data of "month = 26" is invalid. The value must be between 1 and 12.
LINE 1: INSERT INTO datea_type_tab VALUES(3, '251126 00:00:00'::date...
                                          ^
-- 查看数据。
ora_db=# SELECT * FROM datea_type_tab ORDER BY a;
 a |           b
---+-----------------------
 1 | 2021-05-12 05:52:23 AD
 2 | 2020-10-12 11:12:13 AD
(2 rows)

-- 删除表。
ora_db=# DROP TABLE datea_type_tab;
ora_db=# \c postgres
```


版权所有 © 华为云计算技术有限公司

```
-- 删除数据库。
gaussdb=# DROP DATABASE ora_db;
DROP DATABASE

-- MYSQL兼容模式下datetime和timestamp数据类型示例。
-- 创建MYSQL兼容模式下的数据库。
-- 切换到MYSQL兼容模式下的数据库。
gaussdb=# CREATE DATABASE gaussdb_m dbcompatibility='MYSQL';
gaussdb=# \c gaussdb_m;

-- 设置兼容版本控制参数。
gaussdb_m=# SET b_format_version = '5.7';
gaussdb_m=# SET b_format_dev_version = 's1';

-- 创建表。
gaussdb_m=# CREATE TABLE datetime_typ_tab(col1 datetime, col2 timestamp);

-- 插入数据。
gaussdb_m=# INSERT INTO datetime_typ_tab VALUES ('2003-04-12 04:05:06+09:00', '2003-04-12
04:05:06+09:00');

-- 查看数据。
gaussdb_m=# SELECT * FROM datetime_typ_tab;
        col1         |        col2
---------------------+---------------------
 2003-04-12 03:05:06 | 2003-04-12 03:05:06
(1 row)

-- 删除表。
gaussdb_m=# DROP TABLE datetime_typ_tab;

-- 创建表。
gaussdb_m=# CREATE TABLE year_typ_tab(col1 year, col2 year(4));

-- 插入数据。
gaussdb_m=# INSERT INTO year_typ_tab VALUES ('2023', now());

-- 查看数据。
gaussdb_m=# SELECT * FROM year_typ_tab;
 col1 | col2
------+------
 2023 | 2023
(1 row)

-- 删除表和数据库。
gaussdb_m=# DROP TABLE year_typ_tab;
gaussdb_m=# \c postgres;
gaussdb=# DROP DATABASE gaussdb_m;

-- 重置参数。
gaussdb=# RESET ALL;
```

表 1-12 nls_date_format 支持的格式

格式	描述
A.D.	公元后。
AD	公元后。
A.M.	上午。
AM	上午。
B.C.	公元前。
BC	公元前。

版权所有 © 华为云计算技术有限公司

格式	描述
CC	世纪（2位），21世纪从2001-01-01开始。
DAY	全长大写日期名（空白填充为9字符）。
DL	返回指定长日期形式。
DS	返回指定短日期。
DDD	一年里的日数（001~366）。
DD	一个月里的日数（01~31）。
DY	缩写大写日期名（3字符）。
D	一周里的日数（1~7；周日是1）。
FX	强制函数严格按照指定的格式解析或格式化日期。
HH24	一天的小时数（00~23）。
HH12	一天的小时数（01~12）。
HH	一天的小时数（01~12）。
IW	ISO一年里的周数（第一个星期四在第一周里）。
IYYY	ISO年（4位或更多位）。
IYY	ISO年的最后三位。
IY	ISO年的最后两位。
I	ISO年的最后一位。
J	儒略日（自公元前4713年1月1日以后的天数）。
MI	分钟（00~59）。
MM	月份数（01~12）。
MONTH	全长大写月份名（空白填充为9字符）。
MON	大写缩写月份名（3字符）。
P.M.	下午。
PM	下午。
Q	季度。
RM	罗马数字的月份（I~XII，其中I=JAN）（大写）。
RRRR	返回日期的4位年份。
RR	返回日期的2位年份。
SSSSS	午夜后的秒（0~86399）。
SCC	世纪，公元前。

版权所有 © 华为云计算技术有限公司

格式	描述
SS	秒（00~59）。
SYEAR	返回字符型年，公元前会显示-。
SYYYY	返回数字型年，公元前会显示-。
TS	返回指定时间格式。
WW	一年里的周数（1~53）（第一周从该年的第一天开始）。
W	一个月里的周数（1~5）（第一周从该月第一天开始）。
Y,YYY	带逗号的年（4和更多位）。
YEAR	返回字符型年。
YYYY	年（4和更多位）。
YYY	年的后三位。
YY	年的后两位。
Y	年的最后一位。

说明

GaussDB中，nls_date_format参数值支持使用TH后缀，该后缀的作用是在解析输入的时间字符串时，跳过格式指定部分的两个字节。支持TH后缀的格式包括：HH、HH12、HH24、MI、SS、SSSSS、MM、DDD、DD、D、Y,YYY、YYYY、YYY、YY、Y、RRRR、RR、J、SYYYY。当nls_date_format参数值中包含TH后缀且其不处于字符串末尾时，输入的时间需要预留两个字节，否则会解析错误或者结果异常。例如：

- 正常解析。

```
gaussdb=# select to_datea('2000TH-10-24', 'YYYYTH-MM-DD');  -- 当TH位于YYYY格式后，输入的2000年份后需预留两个字节（可为任意字符）。
to_datea
-------------------------
2000-10-24 00:00:00 AD
(1 row)

gaussdb=# select to_datea('2000TH-10-24', 'YYYYTH-MM-DDTH');  -- 当TH位于nls_date_format参数末尾，可正常解析。
       to_datea
-------------------------
 2000-10-24 00:00:00 AD
(1 row)
```

- 解析错误。

```
gaussdb=# select to_datea('2000-10-24', 'YYYYTH-MM-DD');
ERROR:  The data of "month = 0" is invalid. The value must be between 1 and 12.
CONTEXT:  referenced column: to_datea
```

- 结果异常。

```
gaussdb=# select to_datea('2000-11-24', 'YYYYTH-MM-DD');  -- 当TH位于YYYY格式后，由于输入的2000年份后未预留2个字节，解析时将“-1”跳过，MM解析为1月，结果异常。
       to_datea
-------------------------
 2000-01-24 00:00:00 AD
(1 row)
```


版权所有 © 华为云计算技术有限公司

日期输入

日期和时间的输入几乎可以是任何合理的格式，包括ISO-8601格式、SQL-兼容格式或者其它的格式。系统支持按照日、月、年的顺序自定义日期输入。如果把DateStyle参数设置为MDY就按照“月-日-年”解析，设置为DMY就按照“日-月-年”解析，设置为YMD就按照“年-月-日”解析。

日期的文本输入需要加单引号包围，语法如下：

```
type [ ( p ) ] 'value'
```

可选的精度声明中的p是一个整数，表示在秒域中小数部分的位数。表1-13显示了date类型的输入格式。

表 1-13 日期输入格式

例子	描述
1999-01-08	ISO 8601格式（建议格式），任何方式下都是1999年1月8日。
January 8, 1999	在任何datestyle输入模式下都无歧义。
1/8/1999	有歧义，在MDY模式下是1月8日，在DMY模式下是8月1日。
1/18/1999	MDY模式下是1月18日，其它模式下被拒绝。
01/02/03	• MDY模式下的2003年1月2日。 • DMY模式下的2003年2月1日。 • YMD模式下的2001年2月3日。
1999-Jan-08	任何模式下都是1月8日。
Jan-08-1999	任何模式下都是1月8日。
08-Jan-1999	任何模式下都是1月8日。
99-Jan-08	YMD模式下是1月8日，否则错误。
08-Jan-99	1月8日，除了在YMD模式下是错误的之外。
Jan-08-99	1月8日，除了在YMD模式下是错误的之外。
19990108	ISO 8601格式，任何模式下都是1999年1月8日。
990108	ISO 8601格式，任何模式下都是1999年1月8日。
1999.008	年和年里的第几天。
J2451187	儒略日。
January 8, 99 BC	公元前99年。

示例：

```
--创建表。
gaussdb=# CREATE TABLE date_type_tab(coll date);
```


版权所有 © 华为云计算技术有限公司

```
--插入数据。
gaussdb=# INSERT INTO date_type_tab VALUES (date '12-10-2010');

--查看数据。
gaussdb=# SELECT * FROM date_type_tab;
        coll
---------------------
 2010-12-10
(1 row)

--查看日期格式。
gaussdb=# SHOW datestyle;
 DateStyle
-----------
 ISO, MDY
(1 row)

--设置日期格式。
gaussdb=# SET datestyle='YMD';
SET

--插入数据。
gaussdb=# INSERT INTO date_type_tab VALUES(date '2010-12-11');

--查看数据。
gaussdb=# SELECT * FROM date_type_tab;
        coll
---------------------
 2010-12-10
 2010-12-11
(2 rows)

--删除表。
gaussdb=# DROP TABLE date_type_tab;
```

时间

时间类型包括time [(p)] without time zone和time [(p)] with time zone。如果只写time等效于time without time zone。

如果在time without time zone类型的输入中声明了时区，则会忽略这个时区。

时间输入类型的详细信息请参见**表1-14**，时区输入类型的详细信息请参见**表1-15**。

表 1-14 时间输入

例子	描述
05:06.8	ISO 8601
4:05:06	ISO 8601
4:05	ISO 8601
040506	ISO 8601
4:05 AM	与04:05一样，输入小时数必须<= 12。
4:05 PM	与16:05一样，输入小时数必须<= 12。
04:05:06.789-8	ISO 8601
04:05:06-08:00	ISO 8601

版权所有 © 华为云计算技术有限公司

例子	描述
04:05-08:00	ISO 8601
040506-08	ISO 8601
04:05:06 PST	缩写的时区。
2003-04-12 04:05:06 America/New_York	用名称声明的时区。

表 1-15 时区输入

例子	描述
PST	太平洋标准时间（Pacific Standard Time）。
America/New_York	完整时区名称。
-8:00	ISO 8601与PST的偏移。
-800	ISO 8601与PST的偏移。
-8	ISO 8601与PST的偏移。

示例：

```
gaussdb=# SELECT time '04:05:06';
   time
----------
 04:05:06
(1 row)

gaussdb=# SELECT time '04:05:06 PST';
   time
----------
 04:05:06
(1 row)

gaussdb=# SELECT time with time zone '04:05:06 PST';
   timetz
-------------
 04:05:06-08
(1 row)
```

特殊值

GaussDB支持几个特殊值，在读取的时候将被转换成普通的日期/时间值，请参考**表 1-16**。

版权所有 © 华为云计算技术有限公司

表 1-16 特殊值

输入字符串	适用类型	描述
epoch	date，timestamp	1970-01-01 00:00:00+00 （Unix系统零时）。
infinity	timestamp	无穷大，比任何其他时间戳都晚。
-infinity	timestamp	无穷小，比任何其他时间戳都早。
now	date，time，timestamp	当前事务的开始时间。
today	date，timestamp	今日午夜零时。
tomorrow	date，timestamp	明日午夜零时。
yesterday	date，timestamp	昨日午夜零时。
allballs	time	00:00:00.00 UTC

示例：

```
--创建表。
gaussdb=# CREATE TABLE realtime_type_special(col1 varchar(20), col2 date, col3 timestamp, col4 time);

--插入数据。
gaussdb=# INSERT INTO realtime_type_special VALUES('epoch', 'epoch', 'epoch', NULL);
gaussdb=# INSERT INTO realtime_type_special VALUES('now', 'now', 'now', 'now');
gaussdb=# INSERT INTO realtime_type_special VALUES('today', 'today', 'today', NULL);
gaussdb=# INSERT INTO realtime_type_special VALUES('tomorrow', 'tomorrow', 'tomorrow', NULL);
gaussdb=# INSERT INTO realtime_type_special VALUES('yesterday', 'yesterday', 'yesterday', NULL);

--查看数据。
gaussdb=# SELECT * FROM realtime_type_special;
   col1    |        col2         |            col3            |      col4
-----------+---------------------+----------------------------+-----------------
 epoch     | 1970-01-01 00:00:00 | 1970-01-01 00:00:00        |
 now       | 2023-02-27 11:38:13 | 2023-02-27 11:38:13.032815 | 11:38:13.032815
 today     | 2023-02-27 00:00:00 | 2023-02-27 00:00:00        |
 tomorrow  | 2023-02-28 00:00:00 | 2023-02-28 00:00:00        |
 yesterday | 2023-02-26 00:00:00 | 2023-02-26 00:00:00        |
(5 rows)

gaussdb=# SELECT * FROM realtime_type_special WHERE col3 < 'infinity';
   col1    |        col2         |            col3            |      col4
-----------+---------------------+----------------------------+-----------------
 epoch     | 1970-01-01 00:00:00 | 1970-01-01 00:00:00        |
 now       | 2023-02-27 11:38:13 | 2023-02-27 11:38:13.032815 | 11:38:13.032815
 today     | 2023-02-27 00:00:00 | 2023-02-27 00:00:00        |
 tomorrow  | 2023-02-28 00:00:00 | 2023-02-28 00:00:00        |
 yesterday | 2023-02-26 00:00:00 | 2023-02-26 00:00:00        |
(5 rows)

gaussdb=# SELECT * FROM realtime_type_special WHERE col3 > '-infinity';
   col1    |        col2         |            col3            |      col4
-----------+---------------------+----------------------------+-----------------
 epoch     | 1970-01-01 00:00:00 | 1970-01-01 00:00:00        |
 now       | 2023-02-27 11:38:13 | 2023-02-27 11:38:13.032815 | 11:38:13.032815
 today     | 2023-02-27 00:00:00 | 2023-02-27 00:00:00        |
 tomorrow  | 2023-02-28 00:00:00 | 2023-02-28 00:00:00        |
 yesterday | 2023-02-26 00:00:00 | 2023-02-26 00:00:00        |
(5 rows)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM realtime_type_special WHERE col3 > 'now';
   col1   |        col2         |        col3         | col4
----------+---------------------+---------------------+------
 tomorrow | 2023-02-28 00:00:00 | 2023-02-28 00:00:00 |
(1 row)

gaussdb=# SELECT * FROM realtime_type_special WHERE col3 = 'today';
 col1  |        col2         |        col3         | col4
-------+---------------------+---------------------+------
 today | 2023-02-27 00:00:00 | 2023-02-27 00:00:00 |
(1 row)

gaussdb=# SELECT * FROM realtime_type_special WHERE col3 = 'tomorrow';
   col1   |        col2         |        col3         | col4
----------+---------------------+---------------------+------
 tomorrow | 2023-02-28 00:00:00 | 2023-02-28 00:00:00 |
(1 row)

gaussdb=# SELECT * FROM realtime_type_special WHERE col3 > 'yesterday';
   col1   |        col2         |            col3             |      col4
----------+---------------------+-----------------------------+-----------------
 now      | 2023-02-27 11:38:13 | 2023-02-27 11:38:13.032815 | 11:38:13.032815
 today    | 2023-02-27 00:00:00 | 2023-02-27 00:00:00         |
 tomorrow | 2023-02-28 00:00:00 | 2023-02-28 00:00:00         |
(3 rows)

--删除表。
gaussdb=# DROP TABLE realtime_type_special;
```

时间段输入

reltime的输入方式可以采用任何合法的时间段文本格式，包括数字形式（含负数和小数）及时间形式，其中时间形式的输入支持SQL标准格式、ISO-8601格式等。另外，文本输入需要加单引号。

时间段输入的详细信息请参考**表1-17**。

表 1-17 时间段输入

输入示例	输出结果	描述
60	2 mons	采用数字表示时间段，默认单位是day，可以是小数或负数。特别的，负数时间段，在语义上，可以理解为“早于多久”。
31.25	1 mons 1 days 06:00:00	
-365	-12 mons -5 days	
1 years 1 mons 8 days 12:00:00	1 years 1 mons 8 days 12:00:00	采用POSTGRES格式表示时间段，可以正负混用，不区分大小写，输出结果为将输入时间段计算并转换得到的简化POSTGRES格式时间段。
-13 months -10 hours	-1 years -25 days -04:00:00	
-2 YEARS +5 MONTHS 10 DAYS	-1 years -6 mons -25 days -06:00:00	

版权所有 © 华为云计算技术有限公司

输入示例	输出结果	描述
P-1.1Y10M	-3 mons -5 days -06:00:00	采用ISO-8601格式表示时间段，可以正负混用，不区分大小写，输出结果为将输入时间段计算并转换得到的简化POSTGRES格式时间段。
-12H	-12:00:00	

示例：

```
--创建表。
gaussdb=# CREATE TABLE reltime_type_tab(col1 character(30), col2 reltime);

--插入数据。
gaussdb=# INSERT INTO reltime_type_tab VALUES ('90', '90');
gaussdb=# INSERT INTO reltime_type_tab VALUES ('-366', '-366');
gaussdb=# INSERT INTO reltime_type_tab VALUES ('1975.25', '1975.25');
gaussdb=# INSERT INTO reltime_type_tab VALUES ('-2 YEARS +5 MONTHS 10 DAYS', '-2 YEARS +5
MONTHS 10 DAYS');
gaussdb=# INSERT INTO reltime_type_tab VALUES ('30 DAYS 12:00:00', '30 DAYS 12:00:00');
gaussdb=# INSERT INTO reltime_type_tab VALUES ('P-1.1Y10M', 'P-1.1Y10M');

--查看数据。
gaussdb=# SELECT * FROM reltime_type_tab;
              col1              |                col2
--------------------------------+-------------------------------------
 90                             | 3 mons
 -366                           | -1 years -18:00:00
 1975.25                        | 5 years 4 mons 29 days
 -2 YEARS +5 MONTHS 10 DAYS     | -1 years -6 mons -25 days -06:00:00
 30 DAYS 12:00:00               | 1 mon 12:00:00
 P-1.1Y10M                      | -3 mons -5 days -06:00:00
(6 rows)

--删除表。
gaussdb=# DROP TABLE reltime_type_tab;
```

1.3.7 几何类型

GaussDB支持的几何类型请参见**表1-18**。最基本的类型是点（point），点是其它几何类型的基础。

表 1-18 几何类型

名称	存储空间	说明	表现形式
point	16字节	平面中的点	(x,y)
lseg	32字节	（有限）线段	((x1,y1),(x2,y2))
box	32字节	矩形	((x1,y1),(x2,y2))
path	16+16n字节	闭合路径（与多边形相似）	((x1,y1),...)
path	16+16n字节	开放路径	[(x1,y1),...]
polygon	40+16n字节	多边形（与闭合路径相似）	((x1,y1),...)

版权所有 © 华为云计算技术有限公司

名称	存储空间	说明	表现形式
circle	24字节	圆	<(x,y),r>（圆心和半径）

GaussDB提供了一系列的函数和操作符用来进行各种几何计算，包括拉伸、转换、旋转和计算相交等操作。详细信息请参考**几何函数和操作符**。

点

点（point）是几何类型的基本二维构造单位。用以下语法描述point的数值：

```
( x , y )
x , y
```

x和y是用浮点数表示的点的坐标，点的数值类型为float8类型。

点的输出使用第一种语法。

示例：

```
gaussdb=# SELECT point(1.1, 2.2);
   point
-----------
 (1.1,2.2)
(1 row)
```

线段

线段（lseg）是用一对点来表示的。用以下语法描述lseg的数值：

```
[ ( x1 , y1 ) , ( x2 , y2 ) ]
( ( x1 , y1 ) , ( x2 , y2 ) )
( x1 , y1 ) , ( x2 , y2 )
x1 , y1   ,   x2 , y2
```

(x1,y1)和(x2,y2)表示线段的端点，点的数值类型为float8类型。

线段的输出使用第一种语法。

示例：

```
gaussdb=# SELECT lseg(point(1.1, 2.2), point(3.3, 4.4));
         lseg
-----------------------
 [(1.1,2.2),(3.3,4.4)]
(1 row)
```

矩形

矩形（box）是用一对对角点来表示的。用以下语法描述box的值：

```
( ( x1 , y1 ) , ( x2 , y2 ) )
( x1 , y1 ) , ( x2 , y2 )
x1 , y1   ,   x2 , y2
```

(x1,y1)和(x2,y2)表示矩形的一对对角点，点的数值类型为float8类型。

矩形的输出使用第二种语法。

版权所有 © 华为云计算技术有限公司

任意两个对角点都可以使用，但会以右上角端点在前、左下角端点在后的顺序重新排序并存储。

示例：

```
gaussdb=# SELECT box(point(1.1, 2.2), point(3.3, 4.4));
        box
---------------------
 (3.3,4.4),(1.1,2.2)
(1 row)
```

路径

路径（path）由一系列连接的点组成。可分为开放路径和闭合路径，开放路径表示路径的首尾点不连接，而闭合路径则认为首尾点存在连接关系。

用下面的语法描述path的数值：

```
[ ( x1 , y1 ) , ... , ( xn , yn ) ]
( ( x1 , y1 ) , ... , ( xn , yn ) )
( x1 , y1 ) , ... , ( xn , yn )
( x1 , y1   , ... ,  xn , yn )
x1 , y1   , ... ,  xn , yn
```

点表示组成路径的线段的端点，点的数值类型为float8类型。方括号（[]）表示一个开放的路径，圆括号（()）表示一个闭合的路径。当最外层的括号被省略时（如使用第三到第五种语法），会被假定为闭合的路径。

路径的输出使用第一种或第二种语法。

示例：

```
gaussdb=# SELECT path(polygon '((0,0),(1,1),(2,0))');
        path
---------------------
 ((0,0),(1,1),(2,0))
(1 row)
```

多边形

多边形（polygon）由一系列点表示（多边形的顶点）。多边形可以认为是与闭合路径一样，但存储方式不一样，并且多边形有自己的一套支持函数。

用下面的语法描述polygon的数值：

```
( ( x1 , y1 ) , ... , ( xn , yn ) )
( x1 , y1 ) , ... , ( xn , yn )
( x1 , y1   , ... ,  xn , yn )
x1 , y1   , ... ,  xn , yn
```

点表示多边形的顶点，点的数值类型为float8类型。

多边形的输出使用第一种语法。

示例：

```
gaussdb=# SELECT polygon(box '((0,0),(1,1))');
         polygon
----------------------------
 ((0,0),(0,1),(1,1),(1,0))
(1 row)
```

 版权所有 © 华为云计算技术有限公司

圆

圆（circle）由一个圆心和半径表示。用下面的格式描述circle的数值：

```
< ( x , y ) , r >
( ( x , y ) , r )
( x , y ) , r
x , y   , r
```

(x,y)表示圆心，r表示半径，点的数值类型为float8类型。

圆的输出使用第一种格式。

示例：

```
gaussdb=# SELECT circle(point(0,0),1);
  circle
----------
 <(0,0),1>
(1 row)
```

1.3.8 网络地址类型

GaussDB提供用于存储IPv4、IPv6、MAC地址的数据类型如**表1-19**所示，相较于纯文本类型，这些类型提供了输入错误检查和特殊的操作和功能（请参见**网络地址函数和操作符**）。

表 1-19 网络地址类型

名称	存储空间	描述
cidr	7或19字节	IPv4或IPv6网络。
inet	7或19字节	IPv4或IPv6主机和网络。
macaddr	6字节	MAC地址。

在对inet或cidr数据类型进行排序的时候，IPv4地址总是排在IPv6地址前，包括那些封装或者是映射在IPv6地址里的IPv4地址，比如::10.2.3.4或::ffff:10.4.3.2。

cidr

cidr（无类别域间路由，Classless Inter-Domain Routing）类型，保存一个IPv4或IPv6网络地址，如**表1-20**所示。声明网络格式为address/y，address表示IPv4或者IPv6地址，y表示子网掩码的二进制位数。如果省略y，则掩码部分使用已有类别的网络编号系统进行计算，但要求输入的数据已经包括了确定掩码所需的所有字节。

表 1-20 cidr 类型输入举例

cidr输入	cidr输出	abbrev（cidr）
192.168.100.128/25	192.168.100.128/25	192.168.100.128/25
192.168/24	192.168.0.0/24	192.168.0/24
192.168/25	192.168.0.0/25	192.168.0.0/25

版权所有 © 华为云计算技术有限公司

cidr输入	cidr输出	abbrev（cidr）
192.168.1	192.168.1.0/24	192.168.1/24
192.168	192.168.0.0/24	192.168.0/24
10.1.2	10.1.2.0/24	10.1.2/24
10.1	10.1.0.0/16	10.1/16
10	10.0.0.0/8	10/8
10.1.2.3/32	10.1.2.3/32	10.1.2.3/32
2001:4f8:3:ba::/64	2001:4f8:3:ba::/64	2001:4f8:3:ba::/64
2001:4f8:3:ba:2e0:81ff:fe22:d1f1/128	2001:4f8:3:ba:2e0:81ff:fe22:d1f1/128	2001:4f8:3:ba:2e0:81ff:fe22:d1f1
::ffff:127.0.0.0/120	::ffff:127.0.0.0/120	::ffff:127.0.0.0/120
::ffff:127.0.0.0/128	::ffff:127.0.0.0/128	::ffff:127.0.0.0/128

inet

inet类型用于在一个数据区域内存储主机的IPv4或IPv6地址，以及一个可选子网。主机地址中网络地址的位数表示子网（“子网掩码”）。如果子网掩码是32并且是IPv4地址，则这个值不表示任何子网，只表示一台主机。在IPv6里，地址长度是128位，因此128位表示唯一的主机地址。

该类型的输入格式为address/y，其中address表示IPv4或IPv6地址，y是子网掩码的二进制位数。如果省略/y，则子网掩码对IPv4是32，对IPv6是128，所以该值表示只有一台主机。如果该值表示只有一台主机，/y将不会显示。

inet和cidr类型之间的基本区别在于，inet接受子网掩码，而cidr不接受。

macaddr

macaddr类型存储MAC地址，也就是以太网卡硬件地址（尽管MAC地址还用于其它用途）。可以接受下列格式：

```
'08:00:2b:01:02:03'
'08-00-2b-01-02-03'
'08002b:010203'
'08002b-010203'
'0800.2b01.0203'
'08002b010203'
```

以上示例都表示同一个地址。对于数据位a到f，不区分大小写，输出时都是以第一种形式展示。

1.3.9 位串类型

位串是一串二进制数字（1和0）组成的字符串，可以用于存储位掩码。

GaussDB支持两种位串类型：bit(n)和bit varying(n)，其中n是一个正整数，最大取值为83886080，相当于10M的容量。

版权所有 © 华为云计算技术有限公司

bit类型的数据必须准确匹配长度n，存储长度不为n的数据会报错。bit varying类型的数据是最长为n的变长类型，长度超过n时会被拒绝。一个没有指定长度的bit等效于bit(1)，没有指定长度的bit varying表示没有长度限制。

说明

- 如果用户明确地把一个位串数值转换成bit(n)，则此位串右边超过n位的内容将被截断，或者在位串不足n位时右边补齐零到n位，不会抛出错误。
- 如果用户明确地把一个位串数值转换成bit varying(n)，但位串超过了n位，则位串的右边将被截断。
- 使用ADMS平台8.1.3-200驱动版本及之前版本时，写入bit类型需要用::bit varying进行类型转换，否则可能出现异常报错。

```
--创建表。
gaussdb=# CREATE TABLE bit_type_t1
(
    BT_COL1 INTEGER,
    BT_COL2 BIT(3),
    BT_COL3 BIT VARYING(5)
) DISTRIBUTE BY REPLICATION;

--插入数据。
gaussdb=# INSERT INTO bit_type_t1 VALUES(1, B'101', B'00');

--插入数据的长度不符合类型的标准会报错。
gaussdb=# INSERT INTO bit_type_t1 VALUES(2, B'10', B'101');
ERROR:  bit string length 2 does not match type bit(3)
CONTEXT:  referenced column: bt_col2

--将不符合类型长度的数据进行转换。
gaussdb=# INSERT INTO bit_type_t1 VALUES(2, B'10'::bit(3), B'101');

--查看数据。
gaussdb=# SELECT * FROM bit_type_t1;
 bt_col1 | bt_col2 | bt_col3
---------+---------+---------
       1 | 101     | 00
       2 | 100     | 101
(2 rows)

--删除表。
gaussdb=# DROP TABLE bit_type_t1;
```

1.3.10 UUID 类型

UUID数据类型用于存储符合RFC 4122、ISO/IEC 9834-8:2005以及相关标准定义的通用唯一标识符（UUID）。该标识符是一个由算法产生的128位标识符（大小为16字节），确保不会在已知的模块中产生相同的标识符。

因此，对分布式系统而言，这种标识符比序列能更好的保证唯一性，因为序列只能在单一数据库中保证是唯一。

标准格式的UUID由32个小写十六进制数字组成，大小为128位，通过连字符分为几组，一组8个数字+三组4个数字+一组12个数字，典型示例如下：

```
a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11
```

GaussDB同样支持以其他方式输入：大写字母和数字、由花括号包围的标准格式、省略部分或所有连字符、在任意一组四个数字之后加一个连字符。示例：

```
A0EEBC99-9C0B-4EF8-BB6D-6BB9BD380A11
{a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11}
a0eebc999c0b4ef8bb6d6bb9bd380a11
a0ee-bc99-9c0b-4ef8-bb6d-6bb9-bd38-0a11
```

 版权所有 © 华为云计算技术有限公司

默认是以标准格式输出。

示例

```
-- 生成一个UUID类型的序列号
gaussdb=# SELECT uuid();
                 uuid
--------------------------------------
 846b9d00-172d-c63d-4a03-e8b4700370e0
(1 row)

-- 创建表
gaussdb=# CREATE TABLE uuid_test(id int, test uuid) DISTRIBUTE BY HASH(test);

-- 插入数据，使用系统生成的uuid插入数据
gaussdb=# INSERT INTO uuid_test VALUES(1, uuid()::uuid);

-- 插入数据，使用示例格式插入数据
gaussdb=# INSERT INTO uuid_test VALUES(2, 'A0EEBC99-9C0B-4EF8-BB6D-6BB9BD380A11');
gaussdb=# INSERT INTO uuid_test VALUES(3, '{a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11}');
gaussdb=# INSERT INTO uuid_test VALUES(4, 'a0eebc999c0b4ef8bb6d6bb9bd380a11');
gaussdb=# INSERT INTO uuid_test VALUES(5, 'a0ee-bc99-9c0b-4ef8-bb6d-6bb9-bd38-0a11');

-- 查看数据，输出时以标准格式输出
gaussdb=# SELECT * FROM uuid_test;
 id |                 test
----+--------------------------------------
  1 | 1b52051c-1731-c63d-4a03-e8b4700370e0
  2 | a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11
  3 | a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11
  4 | a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11
  5 | a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11
(5 rows)

-- 删除表
gaussdb=# DROP TABLE uuid_test;
```

1.3.11 文本检索类型

GaussDB提供了两种数据类型用于支持全文检索。tsvector类型是一种被全文检索优化后的文档数据类型，tsquery类型表示文本查询类型。

tsvector

tsvector类型表示一个检索单元，通常是一个数据库表中一行的文本字段或者这些字段的组合，tsvector类型的值是一个标准词位的有序列表，标准词位就是把同一个词的变型体都标准化，在输入的同时会自动排序和去重，如下所示。

```
gaussdb=# SELECT 'a fat cat sat on a mat and ate a fat rat'::tsvector;
                    tsvector
-----------------------------------------------------
 'a' 'and' 'ate' 'cat' 'fat' 'mat' 'on' 'rat' 'sat'
(1 row)
```

如果词条中需要包含空格或标点符号，可以用引号标记：

```
gaussdb=# SELECT $$the lexeme '    ' contains spaces$$::tsvector;
                tsvector
-------------------------------------------
 '    ' 'contains' 'lexeme' 'spaces' 'the'
(1 row)
```

如果在词条中需要使用引号，可以使用双$$符号作为标记：

```
gaussdb=# SELECT $$the lexeme 'Joe''s' contains a quote$$::tsvector;
                  tsvector
```


版权所有 © 华为云计算技术有限公司

```
------------------------------------------------
 'Joe''s' 'a' 'contains' 'lexeme' 'quote' 'the'
(1 row)
```

词条位置常量也可以放到词汇中：

```
gaussdb=# SELECT 'a:1 fat:2 cat:3 sat:4 on:5 a:6 mat:7 and:8 ate:9 a:10 fat:11 rat:12'::tsvector;
                               tsvector
-------------------------------------------------------------------------------
 'a':1,6,10 'and':8 'ate':9 'cat':3 'fat':2,11 'mat':7 'on':5 'rat':12 'sat':4
(1 row)
```

位置常量通常表示文档中源字的位置。位置信息可以用于排名。位置常量的范围是1到16383；更大的数字会被自动设置为16383。相同词的重复位置会被忽略掉。

拥有位置的词条可以进一步用一个权重来标记，可以是A、B、C或D。默认值是D，在输出中不显示：

```
gaussdb=# SELECT 'a:1A fat:2B,4C cat:5D'::tsvector;
          tsvector
----------------------------
 'a':1A 'cat':5 'fat':2B,4C
(1 row)
```

权重可以用来反映文档结构，如：标记标题与主体文字的权重区别。全文检索排序函数可以为不同的权重标记分配不同的优先级。

tsvector类型本身不执行任何单词规范化，它假设给定的单词已经适当地进行了规范化。下面的示例是tsvector类型标准用法。如：

```
gaussdb=# SELECT 'The Fat Rats'::tsvector;
      tsvector
--------------------
 'Fat' 'Rats' 'The'
(1 row)
```

对于大多数英文文本搜索应用程序来说，上述单词被认为是非规范化的，但tsvector不会进行规范化处理，所以需要通过to_tsvector函数对这些单词进行规范化处理：

```
gaussdb=# SELECT to_tsvector('english', 'The Fat Rats');
   to_tsvector
-----------------
 'fat':2 'rat':3
(1 row)
```

上述示例中的“The”是停用词。一些词会被识别为“停用词”，因为它们出现得太频繁导致在搜索中起不到作用，所以搜索时停用词会被忽略，to_tsvector函数做规范化处理时会将停用词去除。但是，停用词会影响tsvector中单词的位置，进而会影响排名，上述示例中缺失的位置1就是因为停用词。

tsquery

tsquery类型表示一个检索条件，存储用于检索的词汇，并且支持使用如下操作符：

- 布尔操作符：&（AND），|（OR）和!（NOT）。
- 词组操作符（PHRASE）：<->或<n>，<n>表示两个词在文档中的距离为n（允许前导零），<->等价于<1>。解析词组操作符需要GUC参数system_function_version包含parse_phrase_fix选项。

圆括号可以被用来强制对操作符分组。如果没有圆括号，则!（NOT）的优先级最高，其次是<->/<n>（PHRASE），然后是&（AND），最后是|（OR）。

```
gaussdb=# SELECT 'fat & rat'::tsquery;
   tsquery
```


版权所有 © 华为云计算技术有限公司

```
---------------
 'fat' & 'rat'
(1 row)

gaussdb=# SELECT 'fat & (rat | cat)'::tsquery;
          tsquery
----------------------------
 'fat' & ( 'rat' | 'cat' )
(1 row)

gaussdb=# SELECT 'fat & rat & ! cat'::tsquery;
        tsquery
------------------------
 'fat' & 'rat' & !'cat'
(1 row)

gaussdb=# SELECT 'fat <-> cat'::tsquery;
     tsquery
-----------------
 'fat' <-> 'cat'
(1 row)
```

tsquery中的词汇可以用一个或多个权重字母标记，这将限制它们只能和具有相应权重之一的tsvector词位匹配；如果tsvector的对应词位没有位置信息，则tsquery中的权重限制将被忽略：

```
gaussdb=# SELECT 'fat:ab & cat'::tsquery;
     tsquery
------------------
 'fat':AB & 'cat'
(1 row)
gaussdb=# SELECT 'fat:1C'::tsvector @@ 'fat:ab'::tsquery AS RESULT;
 result
--------
 f
(1 row)

gaussdb=# SELECT 'fat'::tsvector @@ 'fat:ab'::tsquery AS RESULT;
 result
--------
 t
(1 row)
```

另外，tsquery中的词汇可以用*标记指定前缀匹配：

```
gaussdb=# SELECT 'super:*'::tsquery;
  tsquery
-----------
 'super':*
(1 row)
```

这个查询可以匹配tsvector中以“super”开始的任意单词。

请注意，to_tsquery也会使用与其他函数相同的方式进行规范化，以下示例结果为真：

```
gaussdb=# SELECT to_tsvector( 'seriousness' ) @@ to_tsquery( 'series:*' ) AS RESULT;
  result
----------
 t
(1 row)
```

因为series经过处理后得到seri，这样就能匹配seriousness了：

```
gaussdb=# SELECT to_tsquery('series:*');
 to_tsquery
------------
 'seri':*
(1 row)
```


版权所有 © 华为云计算技术有限公司

需要注意的是，tsquery类型本身不执行任何单词规范化，所以在转换为tsquery类型之前，需要对单词进行规范化。函数to_tsquery用于执行这种规范化：

```
gaussdb=# SELECT to_tsquery('Fat:ab & Cats');
    to_tsquery
------------------
 'fat':AB & 'cat'
(1 row)
```

1.3.12 JSON/JSONB 类型

JSON（JavaScript Object Notation）数据，可以是一个单独的标量，也可以是数组或键值对象，其中数组和对象可以统称为容器（container）：

- 标量（scalar）：单一的数字、布尔值、字符串和null都可以称作标量。
- 数组（array）：[]结构，里面存放的元素可以是任意类型的JSON，并且不要求数组内所有元素都是同一类型。
- 对象（object）：{}结构，存储key:value的键值对，键必须是用双引号""包裹起来的字符串，值可以是任意类型的JSON，键重复时，以最后一个键值为准。

GaussDB提供JSON和JSONB两种数据类型存储JSON数据。其中：

- JSON类型存储输入文本的完整副本，使用时再去解析，所以它会保留输入的空格、重复键以及顺序等。
- JSONB类型以二进制格式存储，解析时会删除与语义无关的细节和重复键，对键值进行重新排序，使用时无需再次解析。

两种数据类型其实都是JSON，接受相同的字符串作为输入，主要差别在于效率。JSON数据类型存储输入文本的完整副本，处理函数必须在每个执行上重新解析；而JSONB数据以解析的二进制格式存储，由于添加了解析机制，因此在输入上稍微慢些，但是在处理上明显更快，因为其不需要重新解析。同时由于JSONB类型存在解析后的格式归一化等操作，同等的语义下只会有一种格式，因此可以更好地支持很多其他额外的操作，比如按照一定的规则进行大小比较等。此外，JSONB也支持索引，这也是一个显著优势。

输入格式

输入必须是一个符合JSON数据格式的字符串，此字符串用单引号''声明。

null（null-json）：仅null，全小写。

```
gaussdb=# SELECT 'null'::json;   -- suc
 json
------
 null
(1 row)

gaussdb=# SELECT 'NULL'::jsonb;  -- err
ERROR:  invalid input syntax for type json
```

数字（num-json）：正负整数、小数、0，支持科学计数法。

```
gaussdb=# SELECT '1'::json;
 json
------
 1
(1 row)

gaussdb=# SELECT '-1.5'::json;
 json
```

 版权所有 © 华为云计算技术有限公司

```
------
 -1.5
(1 row)

gaussdb=# SELECT '-1.5e-5'::jsonb, '-1.5e+2'::jsonb;
   jsonb   | jsonb
-----------+-------
 -0.000015 | -150
(1 row)

gaussdb=# SELECT '001'::json, '+15'::json, 'NaN'::json;  -- 不支持多余的前导0，正数的+号，以及NaN和
infinity。
ERROR:  invalid input syntax for type json
```

布尔（bool-json）：仅true、false，全小写。

```
gaussdb=# SELECT 'true'::json;
 json
------
 true
(1 row)

gaussdb=# SELECT 'false'::jsonb;
 jsonb
-------
 false
(1 row)
```

字符串（str-json）：必须是加双引号的字符串。

```
gaussdb=# SELECT '"a"'::json;
 json
------
 "a"
(1 row)

gaussdb=# SELECT '"abc"'::jsonb;
 jsonb
-------
 "abc"
(1 row)
```

数组（array-json）：使用中括号[]包裹，满足数组书写条件。数组内元素类型可以是任意合法的JSON，且不要求类型一致。

```
gaussdb=# SELECT '[1, 2, "foo", null]'::json;
        json
---------------------
 [1, 2, "foo", null]
(1 row)

gaussdb=# SELECT '[]'::json;
 json
------
 []
(1 row)

gaussdb=# SELECT '[1, 2, "foo", null, [[]], {}]'::jsonb;
             jsonb
-------------------------------
 [1, 2, "foo", null, [[]], {}]
(1 row)
```

对象（object-json）：使用大括号{}包裹，键必须是满足JSON字符串规则的字符串，值可以是任意合法的JSON。

```
gaussdb=# SELECT '{}'::json;
 json
------
```

 版权所有 © 华为云计算技术有限公司

```
 {}
(1 row)

gaussdb=# SELECT '{"a": 1, "b": {"a": 2,  "b": null}}'::json;
                json
-------------------------------------
 {"a": 1, "b": {"a": 2,  "b": null}}
(1 row)

gaussdb=# SELECT '{"foo": [true, "bar"], "tags": {"a": 1, "b": null}}'::jsonb;
                          jsonb
--------------------------------------------------------
 {"foo": [true, "bar"], "tags": {"a": 1, "b": null}}
(1 row)
```

须知

- 区分 'null'::json 和 null::json 是两个不同的概念，类似于字符串 str="null" 和 str=null。
- 对于数字，当使用科学计数法时，JSONB类型会将其展开，而JSON会精确复制输入。

JSONB 高级特性

- 注意事项
  - 不支持作为分区键。
  - 不支持外表。

JSON和JSONB的主要差异在于存储方式上的不同，JSONB存储的是解析后的二进制格式，能够体现JSON的层次结构，更方便直接访问等，因此JSONB相较于JSON具有更多高级特性。

- 格式归一化
  - 对于输入的object-json字符串，解析为jsonb二进制格式后，会自动去除与语义上无关紧要的细节，如空格等：

    ```
    gaussdb=# SELECT '   [1, " a ", {"a"   :1    }]  '::jsonb;
           jsonb
    ----------------------
     [1, " a ", {"a": 1}]
    (1 row)
    ```

  - 对于object-json，会删除重复的键值，只保留最后一个出现的，如：

    ```
    gaussdb=# SELECT '{"a" : 1, "a" : 2}'::jsonb;
      jsonb
    ----------
     {"a": 2}
    (1 row)
    ```

  - 对于object-json，键值会重新进行排序，排序规则为：长度长的在后、长度相等则ascii码大的在后，如：

    ```
    gaussdb=# SELECT '{"aa" : 1, "b" : 2, "a" : 3}'::jsonb;
              jsonb
    ----------------------------
    {"a": 3, "b": 2, "aa": 1}
    (1 row)
    ```

- 大小比较

  由于经过了格式归一化，保证了同一种语义下的JSONB只会有一种存在形式，因此按照制定的规则，可以比较大小。

版权所有 © 华为云计算技术有限公司

  - 首先比较类型：object-jsonb > array-jsonb > bool-jsonb > num-jsonb > str-jsonb > null-jsonb
  - 同类型则比较内容：
    - str-jsonb类型：依据text比较的方法，使用数据库默认排序规则进行比较，返回值正数代表大于，负数代表小于，0表示相等。
    - num-jsonb类型：数值比较。
    - bool-jsonb类型：true > false。
    - array-jsonb类型：长度长的 > 长度短的，长度相等则依次比较每个元素。
    - object-jsonb类型：长度长的 > 长度短的，长度相等则依次比较每个键值对，先比较键，然后比较值。

    说明

    object-jsonb类型比较时，使用的是格式整理后的最终结果进行比较，因此相对于直接的输入未必会很直观。

- 创建索引、主键
  - BTREE索引

    JSONB类型支持创建btree索引，支持创建主键。
- 包含存在

  查询一个JSON之中是否包含某些元素，或者某些元素是否存在于某个JSON中是JSONB的一个重要能力。

```
-- 简单的标量/原始值只包含相同的值。
gaussdb=# SELECT '"foo"'::jsonb @> '"foo"'::jsonb;
 ?column?
----------
 t
(1 row)

-- 左侧数组包含了右侧字符串。
gaussdb=# SELECT '[1, "aa", 3]'::jsonb ? 'aa';
 ?column?
----------
 t
(1 row)

-- 左侧数组包含了右侧的数组所有元素，顺序、重复不重要。
gaussdb=# SELECT '[1, 2, 3]'::jsonb @> '[1, 3, 1]'::jsonb;
 ?column?
----------
 t
(1 row)

-- 左侧object-json包含了右侧object-json的所有键值对。
gaussdb=# SELECT '{"product": "PostgreSQL", "version": 9.4, "jsonb":true}'::jsonb @>
'{"version":9.4}'::jsonb;
 ?column?
----------
 t
(1 row)

-- 左侧数组并没有包含右侧的数组所有元素，因为左侧数组的三个元素为1、2、[1,3]，右侧的为1、3。
gaussdb=# SELECT '[1, 2, [1, 3]]'::jsonb @> '[1, 3]'::jsonb;
 ?column?
----------
 f
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# SELECT '{"foo": {"bar": "baz"}}'::jsonb @> '{"bar": "baz"}'::jsonb;
 ?column?
----------
 f
(1 row)
```

相关的操作符请参见**JSON/JSONB函数和操作符**。

- 函数和操作符

  JSON/JSONB类型相关支持的函数和操作符请参见**JSON/JSONB函数和操作符**。

  说明

  GIN索引可以高效搜索出现在大量JSONB文档（datums）中的键或键值对。提供了两个GIN操作符类（jsonb_ops、jsonb_hash_ops），提供了不同的性能和灵活性取舍。默认的GIN操作符类支持使用@>、<@、?、?&和?|操作符查询，非默认的GIN操作符类jsonb_hash_ops只支持索引@>、<@操作符。

  GIN索引及JSON\JSONB聚集函数所用到的内部函数：

  - gin_compare_jsonb
  - gin_consistent_jsonb
  - gin_consistent_jsonb_hash
  - gin_extract_jsonb
  - gin_extract_jsonb_hash
  - gin_extract_jsonb_query
  - gin_extract_jsonb_query_hash
  - gin_triconsistent_jsonb
  - gin_triconsistent_jsonb_hash

1.3.13 HLL 数据类型

HLL（HyperLoglog）是一种用于统计数据集中唯一值个数的高效近似算法。它有着计算速度快、节省空间的特点，不需要直接存储集合本身，而是存储一种名为HLL的数据结构。每当有新数据需要参与统计时，数据库会对数据进行哈希计算并插入到HLL中，最后就可以根据HLL得到结果。

HLL与其他算法的比较请参见**表1-21**。

表 1-21 HLL 与其他算法比较

项目	Sort算法	Hash算法	HLL
时间复杂度	O(nlogn)	O(n)	O(n)
空间复杂度	O(n)	O(n)	log(logn)
误差率	0	0	≈0.8%
所需存储空间	原始数据大小	原始数据大小	默认规格下最大16kB

HLL在计算速度和存储效率方面都具有优势。在时间复杂度上，Sort算法需要花费至少O(nlogn)的时间进行排序，而Hash算法和HLL均能通过O(n)的线性时间完成统计。但在存储效率方面，Sort算法和Hash算法都需要先存储原始数据再进行统计，会导致存

版权所有 © 华为云计算技术有限公司

储空间消耗巨大，而HLL不需要存储原始数据，只需要维护HLL数据结构，能节省很多存储空间，默认规格下HLL数据结构的最大空间约为16kB。

须知

- 当前默认规格下可计算的最大唯一值个数约为1.1e+15个，误差率约为0.8%。用户应注意如果计算结果超过当前规格下最大唯一值个数会导致计算结果误差率变大，或导致计算失败并报错。
- 用户在首次使用该特性时，应该对业务的唯一值做评估，选取适当的配置参数并做验证，以确保精度符合要求：
  - 当前默认规格下，可以计算的最大唯一值个数为1.1e+15，如果计算得到的唯一值个数为NaN，需要调整log2m，或者采用其他算法计算唯一值个数。
  - 虽然hash算法存在极低的哈希冲突概率，但是建议用户在首次使用时选取2~3个不同的哈希种子进行验证，若各种子计算结果差异较小，则可以从中任选一个作为哈希种子。

HLL中主要的数据结构，请参见**表1-22**。

表 1-22 HyperLogLog 中主要数据结构

数据类型	功能描述
HLL	HLL头部为27字节长度字段，默认规格下数据段长度0~16kB，可直接计算得到唯一值。

创建HLL数据类型时，可支持0~4个参数入参，具体的参数含义与参数规格同函数hll_empty一致。第一个参数为log2m，表示分桶数的对数值，取值范围为10~16；第二个参数为log2explicit，表示Explicit模式的阈值大小，取值范围为0~12；第三个参数为log2sparse，表示Sparse模式的阈值大小，取值范围为0~14；第四个参数为duplicatecheck，表示是否启用duplicatecheck，取值范围为0~1。当入参输入值为-1时，会采用默认值设定HLL的参数。可通过“\d”或“\d+”查看HLL类型的参数。

说明

创建HLL数据类型时，入参的行为不同会导致结果不同：

- 创建HLL类型时对应入参不输入或输入-1时，采用默认值设定对应的HLL参数。
- 输入合法范围的入参时，对应HLL参数采用输入值。
- 输入不合法范围的入参时，创建HLL类型报错。

```
-- 创建hll类型的表，不指定入参。
gaussdb=# CREATE TABLE t1 (id integer, set hll);
gaussdb=# \d t1
      Table "public.t1"
 Column |  Type   | Modifiers
--------+---------+-----------
 id     | integer |
 set    | hll     |

-- 创建hll类型的表，指定前两个入参，后两个采用默认值。
gaussdb=# CREATE TABLE t2 (id integer, set hll(12,4));
gaussdb=# \d t2
         Table "public.t2"
 Column |      Type      | Modifiers
```


版权所有 © 华为云计算技术有限公司

```
--------+----------------+-----------
 id     | integer        |
 set    | hll(12,4,12,0) |

-- 创建hll类型的表，指定第三个入参，其余采用默认值。
gaussdb=# CREATE TABLE t3(id int, set hll(-1,-1,8,-1));
gaussdb=# \d t3
          Table "public.t3"
 Column |      Type      | Modifiers
--------+----------------+-----------
 id     | integer        |
 set    | hll(14,10,8,0) |

-- 创建hll类型的表，指定入参不合法报错。
gaussdb=# CREATE TABLE t4(id int, set hll(5,-1));
ERROR:  log2m = 5 is out of range, it should be in range 10 to 16, or set -1 as default

-- 删除已创建的hll类型的表。
gaussdb=# DROP TABLE t1,t2,t3;
DROP TABLE
```

说明

向包含HLL类型的表插入HLL对象时，HLL类型的设定参数须同插入对象的设定参数一致，否则报错。

```
-- 创建带有hll类型的表。
gaussdb=# CREATE TABLE t1(id integer, set hll(14));

-- 向表中插入hll对象，参数一致，成功。
gaussdb=# INSERT INTO t1 VALUES (1, hll_empty(14,-1));

-- 向表中插入hll对象，参数不一致，失败。
gaussdb=# INSERT INTO t1(id, set) VALUES (1, hll_empty(14,5));
ERROR:  log2explicit does not match: source is 5 and dest is 10

-- 删除表。
gaussdb=# DROP TABLE t1;
```

HLL的应用场景。

- 场景1：“Hello World”

  通过以下示例说明如何使用hll数据类型：

  ```
  -- 创建带有hll类型的表。
  gaussdb=# CREATE TABLE helloworld (id integer, set hll);

  -- 向表中插入空的hll。
  gaussdb=# INSERT INTO helloworld(id, set) VALUES (1, hll_empty());

  -- 把整数经过哈希计算加入到hll中。
  gaussdb=# UPDATE helloworld SET set = hll_add(set, hll_hash_integer(12345)) WHERE id = 1;

  -- 把字符串经过哈希计算加入到hll中。
  gaussdb=# UPDATE helloworld SET set = hll_add(set, hll_hash_text('hello world')) WHERE id = 1;

  -- 得到hll中的distinct值。
  gaussdb=# SELECT hll_cardinality(set) FROM helloworld WHERE id = 1;
   hll_cardinality
  -----------------
                 2
  (1 row)

  -- 删除表。
  gaussdb=# DROP TABLE helloworld;
  ```

- 场景2：“网站访客数量统计”

  通过以下示例说明hll如何统计在一段时间内访问网站的不同用户数量：

  ```
  -- 创建原始数据表，表示某个用户在某个时间访问过网站。
  gaussdb=# CREATE TABLE facts (
  ```


版权所有 © 华为云计算技术有限公司

```
    date            date,
    user_id         integer
);

-- 构造数据，表示一天中有哪些用户访问过网站。
gaussdb=# INSERT INTO facts VALUES ('2019-02-20', generate_series(1,100));
gaussdb=# INSERT INTO facts VALUES ('2019-02-21', generate_series(1,200));
gaussdb=# INSERT INTO facts VALUES ('2019-02-22', generate_series(1,300));
gaussdb=# INSERT INTO facts VALUES ('2019-02-23', generate_series(1,400));
gaussdb=# INSERT INTO facts VALUES ('2019-02-24', generate_series(1,500));
gaussdb=# INSERT INTO facts VALUES ('2019-02-25', generate_series(1,600));
gaussdb=# INSERT INTO facts VALUES ('2019-02-26', generate_series(1,700));
gaussdb=# INSERT INTO facts VALUES ('2019-02-27', generate_series(1,800));

-- 创建表并指定列为hll。
gaussdb=# CREATE TABLE daily_uniques (
   date            date UNIQUE,
   users           hll
);

-- 根据日期把数据分组，并把数据插入到hll中。
gaussdb=# INSERT INTO daily_uniques(date, users)
   SELECT date, hll_add_agg(hll_hash_integer(user_id))
   FROM facts
   GROUP BY 1;

-- 计算每一天访问网站不同用户数量。
gaussdb=# SELECT date, hll_cardinality(users) FROM daily_uniques ORDER BY date;
   date    | hll_cardinality
------------+------------------
 2019-02-20 |              100
 2019-02-21 | 200.217913059312
 2019-02-22 |  301.76494508014
 2019-02-23 | 400.862858326446
 2019-02-24 | 502.626933349694
 2019-02-25 | 601.922606454213
 2019-02-26 | 696.602316769498
 2019-02-27 | 798.111731634412
(8 rows)

-- 计算在2019.02.20到2019.02.26一周中有多少不同用户访问过网站。
gaussdb=# SELECT hll_cardinality(hll_union_agg(users)) FROM daily_uniques WHERE date >=
'2019-02-20'::date AND date <= '2019-02-26'::date;
 hll_cardinality
------------------
 696.602316769498
(1 row)

-- 计算昨天访问过网站而今天没访问网站的用户数量。
gaussdb=# SELECT date, (#hll_union_agg(users) OVER two_days) - #users AS lost_uniques FROM
daily_uniques WINDOW two_days AS (ORDER BY date ASC ROWS 1 PRECEDING);  --默认兼容性（M兼
容性）结果如下：
   date    | lost_uniques
------------+--------------
 2019-02-20 |            0
 2019-02-21 |            0
 2019-02-22 |            0
 2019-02-23 |            0
 2019-02-24 |            0
 2019-02-25 |            0
 2019-02-26 |            0
 2019-02-27 |            0
(8 rows)

-- 删除表。
gaussdb=# DROP TABLE facts;
gaussdb=# DROP TABLE daily_uniques;
```

- 场景3：“插入数据不满足hll数据结构要求”

版权所有 © 华为云计算技术有限公司

当用户给hll类型的字段插入数据的时候，必须保证插入的数据满足hll数据结构要求，如果解析后不满足就会报错。如下示例中：插入数据'E\\1234'时，该数据不满足hll数据结构要求，不能解析成功因此失败报错。

```
gaussdb=# CREATE TABLE test(id integer, set hll);
gaussdb=# INSERT INTO test VALUES(1, 'E\\1234');
ERROR:  not a hll type, size=6 is not enough
gaussdb=# DROP TABLE test;
```

1.3.14 范围类型

范围类型是表示某种元素类型（称为范围的subtype）的值范围的数据类型。例如，timestamp的范围可用于表示会议室的预定时间段，此时数据类型是tsrange（“timestamp range”的简写），而timestamp是其subtype。subtype必须具备有序性，这样才能明确判断一个元素值是在范围值之内、之前还是之后。

范围类型具有显著优势，它们能将多个元素值统一表达为单一范围值，并清晰表达如范围重叠等概念，如用于时间安排的时间和日期范围，以及价格范围、仪器的量程等场景都是可以表示的。

内建范围类型

系统提供有下列内建范围类型：

- int4range — integer的范围
- int8range — bigint的范围
- numrange — numeric的范围
- tsrange — 不带时区的timestamp的范围
- tstzrange — 带时区的timestamp的范围
- daterange — date的范围

此外，还可以自定义范围类型，具体请参见**CREATE TYPE**。

例子

```
-- 创建表格并插入数据。
gaussdb=# CREATE TABLE reservation (room int, during tsrange);
gaussdb=# INSERT INTO reservation VALUES (1108, '[2010-01-01 14:30, 2010-01-01 15:30)');

-- 判断是否包含。
gaussdb=# SELECT int4range(10, 20) @> 3;
 ?column?
----------
 f
(1 row)

-- 判断是否重叠。
gaussdb=# SELECT numrange(11.1, 22.2) && numrange(20.0, 30.0);
 ?column?
----------
 t
(1 row)

-- 抽取上界。
gaussdb=# SELECT upper(int8range(15, 25));
 upper
-------
    25
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
-- 计算交集。
gaussdb=# SELECT int4range(10, 20) * int4range(15, 25);
 ?column?
----------
 [15,20)
(1 row)

-- 判断范围是否为空。
gaussdb=# SELECT isempty(numrange(1, 5));
 isempty
---------
 f
(1 row)

-- 删除表格。
gaussdb=# DROP TABLE reservation;
```

范围类型上的操作符和函数的完整列表请参见**范围函数和操作符**。

说明

在ORA兼容模式下，开启mapping_date_to_datea参数后，DATE类型会映射为DATEA类型，DATEA类型不支持作为范围类型，daterange函数不支持DATEA类型作为入参。

包含和排除边界

每一个非空范围都有两个边界，下界和上界。上下界之间的所有点都被包括在范围内。一个包含边界意味着边界点本身也被包括在范围内，而一个排除边界意味着边界点不被包括在范围内。

在一个范围的文本形式中，包含下界被表达为“[”，不包含下界被表达为“(”；包含上界被表达为“]”，不包含上界被表达为“)”（具体请参见**范围输入/输出**）。

函数lower_inc和upper_inc分别用于检测范围值的上下界。

无限（无界）范围

一个范围的下界可以被忽略，意味着所有小于上界的值都被包括在范围中。同样，如果范围的上界被忽略，那么所有比下界大的值都被包括在范围中。如果上下界都被忽略，该元素类型的所有值都被认为在该范围中。当不设置范围的上界和下界时，即上界为正无穷大，下界为负无穷大，该范围为无限（无界）范围。

具有infinity概念的元素类型可以作为显式边界值。例如，时间戳范围[today,infinity)表示不包括特殊的timestamp值infinity，类似于[today,)，时间戳范围[today,infinity]表示包括特殊的timestamp值infinity，类似于[today,]。

函数lower_inf和upper_inf分别用于检测范围的无限上下界。

范围输入/输出

一个范围值的输入必须遵循以下模式之一：

```
(lower-bound,upper-bound)
(lower-bound,upper-bound]
[lower-bound,upper-bound)
[lower-bound,upper-bound]
empty
```

一个范围值的输出必须遵循以下模式之一：

```
[lower-bound,upper-bound)
empty
```


版权所有 © 华为云计算技术有限公司

圆括号或方括号用于指示上下界是否被包含或排除。empty表示一个空范围（一个不包含点的范围）。

lower-bound可以是作为subtype的一个合法输入的字符串，或者是空（表示没有下界）。同样地，upper-bound可以是作为subtype的一个合法输入的字符串，或者是空（表示没有上界）。

当边界值包含圆括号、方括号、逗号、双引号或反斜杠时，必须使用双引号进行引用，否则这些字符会被认作范围语法的一部分。若要在引用的边界值中包含双引号或反斜杠，需在其前面添加一个反斜杠进行转义（双引号引用的边界值中的连续两个双引号表示一个双引号字符，这与SQL字符串中的单引号转义逻辑一致）。此外，所有可能被解析为范围语法的字符都应进行转义。空字符串边界值需要写成""，什么都不写表示一个无限边界。

范围值前后允许有空格，但是圆括号或方括号之间的任何空格会被当做上下界值的一部分（该空格是否具有语义取决于元素类型）。

示例：

```
-- 包含3，不包含7，并且包含3和7之间的所有点。
gaussdb=# SELECT '[3,7)'::int4range;
 int4range
-----------
 [3,7)
(1 row)
-- 既不包含3，也不包含7，但是包含3和7之间的所有点。
gaussdb=# SELECT '(3,7)'::int4range;
 int4range
-----------
 [4,7)
(1 row)
-- 仅包含单个点4。
gaussdb=# SELECT '[4,4]'::int4range;
 int4range
-----------
 [4,5)
(1 row)
-- 不包含任何点（并且将被标准化为'空'）。
gaussdb=# SELECT '[4,4)'::int4range;
 int4range
-----------
 empty
(1 row)
```

构造范围

每种范围类型都有一个与其同名的构造器函数。使用构造器函数通常比写一个范围文字常数更方便，因为它可以避免对边界值的额外引用。构造器函数接收两个或三个参数：

- 两个参数的形式：按标准格式构造范围（下界是包含的，上界是排除的）。
- 三个参数的形式：按第三个参数指定的边界类型构造范围，第三个参数必须为以下字符串之一：“()”、“(]”、“[)”或者“[]”。

示例：

```
-- 完整形式是：下界、上界以及指示边界包含性/排除性的文本参数。
gaussdb=# SELECT numrange(1.0, 14.0, '(]');
  numrange
------------
 (1.0,14.0]
(1 row)
-- 如果第三个参数被忽略，则默认为'[)'。
```

 版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT numrange(1.0, 14.0);
  numrange
------------
 [1.0,14.0)
(1 row)
-- 对于离散范围类型（如 int8range），尽管第三个参数指定为'(]'，显示时该值将被转换成标准形式。
gaussdb=# SELECT int8range(1, 14, '(]');
 int8range
-----------
 [2,15)
(1 row)
-- 使用NULL作为边界值将导致对应边界是无界的。
gaussdb=# SELECT numrange(NULL, 2.2);
 numrange
----------
 (,2.2)
(1 row)
```

离散范围类型

一种范围的元素类型具有明确的“步长”定义，例如integer或date。在这些类型中，如果两个元素之间不存在合法值，则可将它们称为相邻元素。这与连续范围不同，连续范围中通常可以在两个给定值之间标识其他元素值。例如，numeric类型上的范围就是连续的，timestamp类型上的范围也是（尽管timestamp具有有限的精度，并且在理论上可以被视为离散的，但通常并不关心它的步长，因此也可以被视为连续的）。

离散范围类型的另一个特征是每一个元素值都有一个明确的“下一个”或“上一个”值。基于这种特性，通过使用原来给定的下一个或上一个元素值来取代它，就可以在一个范围界限的包含和排除属性之间转换。例如，在整数范围类型中，[4,8]和(3,9)表示相同的值集合，但对于numeric上的范围就并非如此。

一个离散范围类型应该具有一个正规化函数，即明确元素类型的指定步长。正规化函数的作用是将具有相同值集合的不同范围表达式转换为统一格式，特别是与包含或者排除界限一致。若未指定正规化函数，那么具有不同格式的范围将总是会被视为不等，即使它们实际上是表达相同的一组值。

内建的范围类型int4range、int8range和daterange采用标准的包含下界排除上界的格式（即[)）。但用户定义的范围类型可以使用其他形式。

索引

B-树索引可以在范围类型的表列上创建。对于这类索引，只有等值操作具有实际意义。虽然对范围值定义了使用<和>操作符时的B-树排序顺序，但该排序顺序较为随机，在实际应用中使用价值有限。范围类型的B-树支持主要是为了允许在查询内部进行排序，而不是创建真正的索引。

1.3.15 对象标识符类型

GaussDB在内部使用对象标识符（OID）作为系统表的主键，不建议在用户创建的表中将OID字段作为主键使用。OID类型采用四字节无符号整数实现，系统不会自动为用户创建的表添加OID字段。对象标识符类型如**表1-23**所示。

表 1-23 对象标识符类型

名称	引用	描述	示例
OID	-	以数字形式表示的对象标识符。	564182

版权所有 © 华为云计算技术有限公司

名称	引用	描述	示例
CID	-	32位命令标识符。系统字段cmin和cmax的数据类型。	-
XID	-	64位事务标识符。系统字段xmin和xmax的数据类型。	-
TID	-	行标识符。系统表字段ctid的数据类型。行ID是一对数值（块号，块内的行索引），标识该行在其所在表内的物理位置。	-
REGCONFIG	pg_ts_config	文本搜索配置。	english
REGDICTIONARY	pg_ts_dict	文本搜索字典。	simple
REGOPER	pg_operator	操作符名。	-
REGOPERATOR	pg_operator	带参数类型的操作符。	*(integer,integer)或-(NONE,integer)
REGPROC	pg_proc	函数名称。	sum
REGPROCEDURE	pg_proc	带参数类型的函数。	sum(int4)
REGCLASS	pg_class	关系名。	pg_type
REGTYPE	pg_type	数据类型名。	integer

OID类型：主要作为数据库系统表中字段使用。

示例：

```
gaussdb=# SELECT oid FROM pg_class WHERE relname = 'pg_type';
 oid
------
 1247
(1 row)
```

OID别名类型REGCLASS：主要用于对象OID值的简化查找。

示例：

```
gaussdb=# SELECT attrelid,attname,atttypid,attstattarget FROM pg_attribute WHERE attrelid =
'pg_type'::REGCLASS;
 attrelid |  attname   | atttypid | attstattarget
----------+------------+----------+---------------
     1247 | xc_node_id     |       23 |             0
     1247 | tableoid       |       26 |             0
     1247 | cmax           |       29 |             0
     1247 | xmax           |       28 |             0
     1247 | cmin           |       29 |             0
     1247 | xmin           |       28 |             0
     1247 | oid            |       26 |             0
```


版权所有 © 华为云计算技术有限公司

```
 1247 | ctid          |      27 |           0
 1247 | typname       |      19 |          -1
 1247 | typnamespace  |      26 |          -1
 1247 | typowner      |      26 |          -1
 1247 | typlen        |      21 |          -1
 1247 | typbyval      |      16 |          -1
 1247 | typtype       |      18 |          -1
 1247 | typcategory   |      18 |          -1
 1247 | typispreferred |      16 |          -1
 1247 | typisdefined  |      16 |          -1
 1247 | typdelim      |      18 |          -1
 1247 | typrelid      |      26 |          -1
 1247 | typelem       |      26 |          -1
 1247 | typarray      |      26 |          -1
 1247 | typinput      |      24 |          -1
 1247 | typoutput     |      24 |          -1
 1247 | typreceive    |      24 |          -1
 1247 | typsend       |      24 |          -1
 1247 | typmodin      |      24 |          -1
 1247 | typmodout     |      24 |          -1
 1247 | typanalyze    |      24 |          -1
 1247 | typalign      |      18 |          -1
 1247 | typstorage    |      18 |          -1
 1247 | typnotnull    |      16 |          -1
 1247 | typbasetype   |      26 |          -1
 1247 | typtypmod     |      23 |          -1
 1247 | typndims      |      23 |          -1
 1247 | typcollation  |      26 |          -1
 1247 | typdefaultbin |     194 |          -1
 1247 | typdefault    |      25 |          -1
 1247 | typacl        |    1034 |          -1
 1247 | typelemmod    |      23 |          -1
(39 rows)
```

1.3.16 伪类型

GaussDB数据类型中包含一系列特殊用途的类型，这些类型被归类为伪类型，如**表1-24**所示。伪类型不能作为字段的数据类型，但是可以用于声明函数的参数或结果类型。

当一个函数不仅简单地接受并返回某种SQL数据类型时，伪类型非常有用。**表1-24**列出了所有的伪类型。

表 1-24 伪类型

名称	描述
any	表示函数接受任何输入数据类型。
anyelement	表示函数接受任何数据类型。
anyarray	表示函数接受任意数组数据类型。
anynonarray	表示函数接受任意非数组数据类型。
anyenum	表示函数接受任意枚举数据类型。
anyrange	表示函数接受任意范围数据类型。
cstring	表示函数接受或者返回一个空结尾的C字符串。
internal	表示函数接受或者返回一种服务器内部的数据类型。
language_handler	声明一个过程语言调用句柄返回language_handler。

版权所有 © 华为云计算技术有限公司

名称	描述
fdw_handler	声明一个外部数据封装器返回fdw_handler。
record	标识函数返回一个未声明的行类型。
trigger	声明一个触发器函数返回trigger。
void	表示函数不返回数值。
opaque	一个已经过时的类型，以前用于所有上面这些用途。

用C语言编写的函数（无论是内置的还是动态装载的）都可以将任意伪数据类型作为输入或输出参数。当使用伪类型作为参数类型时，用户需要确保函数能正常运行。

用过程语言编写的函数只能使用实现语言允许的伪类型。目前，过程语言都不允许使用伪类型作为参数类型，并且只允许使用void和record作为返回类型。一些多态的函数还支持使用anyelement、anyarray、anynonarray、anyenum和anyrange类型。

每一个被声明为anyelement的位置（参数或返回值）都允许具有任意特定的实际数据类型，但是在任何给定的查询中必须全部是相同的实际类型。

伪类型internal用于声明只能在数据库系统内部调用的函数，这些函数不能直接在SQL查询里调用。如果函数至少有一个internal类型的参数，则不能从SQL里调用。建议不要创建任何声明返回internal类型的函数，除非其至少有一个internal类型的参数。

示例：

```
--创建表。
gaussdb=# CREATE TABLE t1 (a int);

--插入两条数据。
gaussdb=# INSERT INTO t1 values(1),(2);

--创建函数showall()。
gaussdb=# CREATE OR REPLACE FUNCTION showall() RETURNS SETOF record
AS $$ SELECT count(*) from t1; $$
LANGUAGE SQL;

--调用函数showall()。
gaussdb=# SELECT showall();
 showall
---------
 (2)
(1 row)

--删除函数。
gaussdb=# DROP FUNCTION showall();

--删除表。
gaussdb=# DROP TABLE t1;
```

1.3.17 账本数据库使用的数据类型

账本数据库使用HASH16数据类型存储行级hash摘要或表级hash摘要，使用HASH32数据类型存储全局hash摘要或者历史表校验hash。

版权所有 © 华为云计算技术有限公司

表 1-25 账本数据库 HASH 类型

名称	描述	存储空间	范围
HASH16	以无符号64位整数存储。	8字节	0 ~ +18446744073709551615
HASH32	以包含16个无符号整型元素数的组存储。	16字节	16个元素的无符号整型数组能够包含的取值范围。

HASH16数据类型用于在账本数据库中存储行级或表级hash摘要，在获得长度为16个字符的十六进制字符串的hash序列后，系统会调用hash16in函数，将该序列转换为一个无符号64位整数，并存储到HASH16类型变量中。示例如下：

```
十六进制字符串：e697da2eaa3a775b  对应的无符号64位整数：16615989244166043483
十六进制字符串：ffffffffffffffff  对应的无符号64位整数：18446744073709551615
```

示例：

```
gaussdb=# CREATE TABLE hash16test(id int, data hash16);
gaussdb=# INSERT INTO hash16test VALUES (1, 'e697da2eaa3a775b');
gaussdb=# SELECT * FROM hash16test;
 id |              data
----+----------------------------------
  1 | e697da2eaa3a775b
(1 row)
gaussdb=# DROP TABLE hash16test;
```

HASH32数据类型用于在账本数据库中存储全局hash摘要或者历史表校验hash，在获得长度为32个字符的十六进制字符串的hash序列后，系统会调用hash32in函数，将该序列转换为一个包含16个无符号整型元素的数组。示例如下：

```
十六进制字符串：685847ed1fe38e18f6b0e2b18c00edee
对应的HASH32数组：[104,88,71,237,31,227,142,24,246,176,226,177,140,0,237,238]
```

示例：

```
gaussdb=# CREATE TABLE hash32test(id int, data hash32);
gaussdb=# INSERT INTO hash32test VALUES (1, '685847ed1fe38e18f6b0e2b18c00edee');
gaussdb=# SELECT * FROM hash32test;
 id |              data
----+----------------------------------
  1 | 685847ed1fe38e18f6b0e2b18c00edee
(1 row)
gaussdb=# DROP TABLE hash32test;
```

1.3.18 XML 类型

XML数据类型可以用于存储格式良好的、遵循XML标准定义的“文档”以及“内容”片段，该类型通过引用更广泛的“DOCUMENT NODE”XQUERY和XPATH数据模型来定义，允许内容片段包含多个顶层元素或字符节点，通过表达式XMLVALUE IS DOCUMENT可判断特定XML值是完整文档还是文档片段。其内部格式和TEXT类型相同，但与直接在TEXT域中存储XML数据相比，XML数据类型的优势在于它支持基于LIBXML2提供的标准XML操作函数和XML规范性检查。XML类型相关函数请参见**XML类型函数**。

XML解析器将XML文档转换为XML DOM对象。DOM（Document Object Model 文档对象模型）定义了访问和操作文档的标准方法。XML DOM（XML Document Object Model）定义了访问和操作XML文档的标准方法。XML DOM将XML文档视为树结构。

版权所有 © 华为云计算技术有限公司

所有元素都可以通过DOM树来访问，可以修改或删除它们的内容，并创建新的元素，一个元素、元素的文本、以及元素的属性都被视为节点。

XML底层使用和TEXT类型一样的数据结构进行存储，最大为1GB。

示例：

```
gaussdb=# CREATE TABLE xmltest ( id int, data xml );
gaussdb=# INSERT INTO xmltest VALUES (1, 'one');
gaussdb=# INSERT INTO xmltest VALUES (2, XMLCONCAT(('<?xml version="1.0" encoding="UTF8"
standalone="no"?><bar>foo</bar>'), ('<?xml version="1.0" encoding="UTF8" standalone="no"?><bar>foo</
bar>')));
gaussdb=# SELECT * FROM xmltest ORDER BY 1;
 id |                            data
----+-----------------------------------------------------------------
  1 | one
  2 | <?xml version="1.0" standalone="no"?><bar>foo</bar><bar>foo</bar>
(2 rows)
gaussdb=# DROP TABLE xmltest;
```

XML类型不支持以下操作：

- 逻辑表达式and、or、not。
- 作为非XML操作函数的系统函数的入参。
- 作为分布键、分区键、二级分区键、主键或唯一约束。
- XML相关的隐式类型转换，包括字符串和XML类型之间的隐式转换。
- 数组表达式、行表达式、子查询表达式、TABLE OF和TABLE OF INDEX。
- 使用XML字段创建普通索引、unique索引、global索引、local索引或部分索引。
- 比较表达式>、<、>=、<=、=、<>、!=、^=、between and、is distinct from、is not distinct from、<=>。
- 条件表达式decode、nullif、greatest、least。
- 作为distinct、group by、order by表达式的参数。
- 聚合函数sum、max、min、avg、list_agg、corr、covar_pop、covar_samp、stddev、stddev_pop、stddev_samp、var_pop、var_samp、variance、bit_and、bit_or、bool_and、bool_or、every、regr_avgx、regr_avgy、regr_count、regr_intercept、regr_r2、regr_slope、regr_sxx、regr_sxy、regr_syy、rank、spread。
- 不支持ODBC相关绑定传参接口。

XML类型支持如下操作：

- 物理备份恢复。
- 空值判断is null、is not null。
- 条件表达式case、coalesce。
- 全局临时表和本地临时表。
- 强制类型转换。
- 表达式索引。
- XML类型的输入值需要符合XML格式标准。
- 支持gs_dump导出和gs_restore导入操作。
- 并行查询，支持Astore和Ustore存储引擎。
- 作为自定义函数的入参、出参、自定义变量和返回值。

版权所有 © 华为云计算技术有限公司

- 作为存储过程的入参、出参、自定义变量和返回值。支持自治事务的存储过程。
- 字符处理函数quote_literal(string text)（需显式转换为字符类型）、quote_nullable(string text)（需显式转换为字符类型）。
- 聚集函数count、array_agg、checksum（需显式转换为字符类型）、string_agg（需显式转换为字符类型）。
- 自定义类型中复合类型涉及XML类型的增删改查，且与普通表中的XML字段一样，需要按照XML的语法插入和修改。
- 支持JDBC和ODBC对XML数据类型进行操作，支持对该字段执行SELECT、UPDATE、INSERT或DELETE操作，使用SQL语法输入XML值，使用ResultSet类的getSQLXML方法获取XML值。支持JDBC相关绑定传参接口，可使用PreparedStatement预处理语句接口中的setSQLXML方法和ResultSet执行结果集接口中的getSQLXML(int columnIndex)方法。

  调用流程：需要使用java.sql.SQLXML接口类构造XML对象，再设置对象类型为Oid.XML，然后将类型ID和XML值发送到服务端，从服务端获取返回结果后，先调用ResultSet.getString，再通过java.sql.SQLXML接口类将获取的字符串构造成XML对象，此时会再次检查其内容是否符合XML标准格式。所以XML值也可以通过ResultSet.getString直接获取字符串对象。

1.3.19 XMLTYPE 类型

XMLTYPE数据类型用于存储格式良好的、遵循XML标准定义的“文档”。其内部采用字符串格式存储数据，与直接在TEXT域中存储XML数据相比，XMLTYPE数据类型的优势在于其支持基于LIBXML2提供的标准XML操作函数和XML规范性检查。XMLTYPE类型相关函数请参见**XMLTYPE类型函数**。

XML解析器将XML文档转换为XML DOM对象。DOM（Document Object Model 文档对象模型）定义了访问和操作文档的标准方法。XML DOM（XML Document Object Model）定义了访问和操作XML文档的标准方法。XML DOM将XML文档视为树结构。所有元素都可以通过DOM树来访问，可以修改或删除它们的内容，并创建新的元素，一个元素、元素的文本、以及元素的属性都被视为节点。最大为1GB。

示例：

```
gaussdb=# CREATE TABLE xmltypetest(id int, data xmltype);
gaussdb=# INSERT INTO xmltypetest VALUES (1, '<ss/>');
gaussdb=# INSERT INTO xmltypetest VALUES (2, '<xx/>');
gaussdb=# SELECT * FROM xmltypetest ORDER BY 1;
 id | data
----+-------
  1 | <ss/>
  2 | <xx/>
(2 rows)
gaussdb=# DROP TABLE xmltypetest;
```

XMLTYPE类型不支持以下操作：

- 逻辑表达式and、or、not。
- 作为非XMLTYPE操作函数的系统函数的入参。
- 作为分布键、分区键、二级分区键、主键或唯一约束。
- XMLTYPE相关的隐式类型转换，包括字符串和XMLTYPE类型之间的隐式转换。
- 数组表达式、行表达式、子查询表达式、TABLE OF和TABLE OF INDEX。
- 使用XMLTYPE字段创建普通索引、unique索引、global索引、local索引或部分索引。

版权所有 © 华为云计算技术有限公司

- 比较表达式>、<、>=、<=、=、<>、!=、^=、between and、is distinct from、is not distinct from、<=>。
- 条件表达式decode、nullif、greatest、least。
- 作为distinct、group by、order by表达式的参数。
- 聚合函数sum、max、min、avg、list_agg、corr、covar_pop、covar_samp、stddev、stddev_pop、stddev_samp、var_pop、var_samp、variance、bit_and、bit_or、bool_and、bool_or、every、regr_avgx、regr_avgy、regr_count、regr_intercept、regr_r2、regr_slope、regr_sxx、regr_sxy、regr_syy、rank、spread。
- 不支持ODBC相关绑定传参接口。

XMLTYPE类型支持如下操作：

- 物理备份恢复。
- 空值判断is null、is not null。
- 条件表达式case、coalesce。
- 全局临时表和本地临时表。
- 强制类型转换。
- 表达式索引。
- XMLTYPE类型的输入值需要符合XML格式标准。
- 支持gs_dump导出和gs_restore导入操作。
- 并行查询，支持Astore和Ustore存储引擎。
- 作为自定义函数的入参、出参、自定义变量和返回值。
- 作为存储过程的入参、出参、自定义变量和返回值。支持自治事务的存储过程。
- 字符处理函数quote_literal(string text)（需显式转换为字符类型）、quote_nullable(string text)（需显式转换为字符类型）。
- 聚集函数count、array_agg、checksum（需显式转换为字符类型）、string_agg（需显式转换为字符类型）。
- 自定义类型中复合类型涉及XMLTYPE类型的增删改查，且与普通表中的XMLTYPE字段一样，需要按照XMLTYPE的语法插入和修改。
- 支持对该字段执行SELECT、UPDATE、INSERT或DELETE操作，使用SQL语法输入XMLTYPE值。

可以创建名称为xmltype的Schema，用户可以在该Schema下创建函数，但是不能通过schema.func()这种方式调用Schema下定义的函数。

XMLTYPE类型在输出时，会将XML文本声明中encoding属性修改为当前数据库编码。

1.3.20 aclitem 类型

aclitem数据类型用于存储对象权限信息。其内部实现基于int类型，支持的格式为“*user1=privs/user2*”。

aclitem[]数据类型为多个aclitem元素组成的数组，支持的格式为“{*user1=privs1/user3*，*user2=privs2/user3*}”。

其中user1、user2和user3为数据库中已存在的用户/角色名，privs为数据库中支持的权限（具体请参见**表7-105**）。

示例：

版权所有 © 华为云计算技术有限公司

```
--创建相应用户。
gaussdb=# CREATE USER user1 WITH PASSWORD '********';
gaussdb=# CREATE USER user2 WITH PASSWORD '********';
gaussdb=# CREATE USER omm WITH PASSWORD '********';

--新建一张数据表table_acl，有三个字段，类型分别为int、aclitem、aclitem[]。
gaussdb=# CREATE TABLE table_acl (id int,priv aclitem,privs aclitem[]);

--向数据表table_acl插入一条内容为(1,'user1=arw/omm','{omm=d/user2,omm=w/omm}')的数据。
gaussdb=# INSERT INTO table_acl VALUES (1,'user1=arw/omm','{omm=d/user2,omm=w/omm}');

--向数据表table_acl再插入一条内容为(2,'user1=aw/omm','{omm=d/user2}')的数据。
gaussdb=# INSERT INTO table_acl VALUES (2,'user1=aw/omm','{omm=d/user2}');
gaussdb=# SELECT * FROM  table_acl;
 id |     priv      |          privs
----+---------------+-------------------------
  1 | user1=arw/omm | {omm=d/user2,omm=w/omm}
  2 | user1=aw/omm  | {omm=d/user2}
(2 rows)

--删除表和用户。
gaussdb=# DROP USER user1;
gaussdb=# DROP USER user2;
gaussdb=# DROP USER omm;
gaussdb=# DROP TABLE table_acl;
```

1.3.21 数组类型

数组类型可以用来存储具有相同类型的若干元素。数组类型使用和相关函数请参见**数组**。

数组类型的定义

数组类型一般通过在数组元素的数据类型名称后面加上方括号“[]”进行定义。

示例一，创建一个名为sal_emp的表，拥有一个表示雇员姓名类型为text的列（name），一个表示雇员季度工资的数组且元素类型为integer的列（pay_by_quarter），一个表示雇员手机号码的数组且元素类型为varchar(11)的列（phone_numbers）：

```
gaussdb=# CREATE TABLE sal_emp (
  name            text,
  pay_by_quarter  integer[],
  phone_numbers   varchar(11)[]
);
gaussdb=# DROP TABLE sal_emp;
```

示例二，以其他方式定义一个数组类型，具体定义方法和定义行为参考示例中的注释：

```
gaussdb=# CREATE TABLE sal_emp (
  name             text,
  pay_by_quarter1  integer[][],      -- int类型的二维数组。
  pay_by_quarter2  integer[3],       -- int类型的一维数组，尺寸大小为3。
  pay_by_quarter3  integer[3][3],    -- int类型的二维数组，每一维尺寸大小为3。
  pay_by_quarter4  integer ARRAY,    -- int类型的一维数组。
  pay_by_quarter5  integer ARRAY[3]  -- int类型的一维数组，尺寸大小为3。
);
gaussdb=# DROP TABLE sal_emp;
```


版权所有 © 华为云计算技术有限公司

说明

- 支持在数组类型定义中指定维数，但该定义仅作为类型声明的一部分，不会对实际存储的数组维数进行约束。
- 支持在数组类型定义中指定尺寸，但该定义仅作为类型声明的一部分，不会对实际存储的数组尺寸进行约束。
- 数组实际存储的数据最多支持6维。
- 数组元素个数限制如下：
  - 元素个数最大为134217727个。
  - 所有元素加起来最大存储空间不超过1GB - 1字节，即1073741823字节。

数组构造器

数组构造器是用于构建数组值的表达式。简单的数组构造器由关键词ARRAY、“[”、数组元素值的表达式列表（用逗号分隔）以及最后的“]”组成。示例如下：

```
gaussdb=# SELECT ARRAY[1, 2, 3 + 4];
  array
---------
 {1,2,7}
(1 row)
```

默认情况下，数组的元素类型是成员表达式的公共类型，遵循与UNION或CASE结构（详情请参见**UNION，CASE和相关构造**）相同的类型推导规则。可以通过显式类型转换将数组构造为所需的数据类型，示例如下：

```
gaussdb=# SELECT ARRAY[1, 2, 3]::varchar[];
  array
---------
 {1,2,3}
(1 row)

gaussdb=# SELECT ARRAY['a', 'b', 'c']::varchar[];
  array
---------
 {a,b,c}
(1 row)
```

除预置的基础类型外，record类型和表类型也可以定义其数组类型，示例：

```
gaussdb=# CREATE TYPE rec IS (c1 int, c2 int);
gaussdb=# SELECT ARRAY[(1, 1), (2, 2)]::rec[];
       array
-------------------
 {"(1,1)","(2,2)"}
(1 row)

gaussdb=# CREATE TABLE tab (c1 int, c2 int);
gaussdb=# SELECT ARRAY[(1, 1), (2, 2)]::tab[];
       array
-------------------
 {"(1,1)","(2,2)"}
(1 row)

gaussdb=# DROP TYPE rec;
gaussdb=# DROP TABLE tab;
```

由于数组必须具有明确的类型，因此在构造空数组时，必须明确指定所需类型，示例：

```
gaussdb=# SELECT ARRAY[]::int[];
 array
-------
```


版权所有 © 华为云计算技术有限公司

```
 {}
(1 row)
```

也可以从子查询的结果中构造一个数组。此时，数组构造器由关键字ARRAY后拼接用圆括号包裹的子查询构成，子查询必须只返回一个单独的字段。生成的一维数组将为子查询里每行结果生成一个元素，元素类型匹配子查询的输出字段。示例：

```
gaussdb=# SELECT ARRAY(SELECT generate_series(1, 6));
    array
---------------
 {1,2,3,4,5,6}
(1 row)
```

多维数组值可以通过嵌套数组构造器的方法创建。内层构造器中的ARRAY关键字可以省略。比如，以下两个示例将产生相同结果：

```
gaussdb=# SELECT ARRAY[ARRAY[1,2], ARRAY[3,4]];
    array
---------------
 {{1,2},{3,4}}
(1 row)

gaussdb=# SELECT ARRAY[[1,2], [3,4]];
    array
---------------
 {{1,2},{3,4}}
(1 row)
```

说明

- 同层的内层构造器必须生成同维的子数组。
- 任何应用于外层ARRAY构造器的类型转换将自动应用到所有内层构造器。

数组类型的字符串输入

要把一个数组值写成一个文字常数（常量输入），将元素值用花括号括起并用逗号分隔。因此，一个数组常量的一般格式如下：

```
'{ val1 delim val2 delim ... }'
```

上述格式中的delim是元素类型的分隔符，记录在pg_type表的typdelim列中。每个val可以是数组元素类型的一个常量，也可以是一个子数组。示例如下：

```
gaussdb=# SELECT '{1, 2, 3}'::int[] AS RESULT;
 result
---------
 {1,2,3}
(1 row)

gaussdb=# SELECT '{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}'::int[] AS RESULT;
         result
---------------------------
 {{1,2,3},{4,5,6},{7,8,9}}
(1 row)
```

在任意元素值周围均可以使用双引号，并且在元素值包含逗号或花括号等一些特殊字符时必须使用双引号。

如下示例表示有一个varchar类型的数组，且一共有7个varchar元素，元素依次为：

- 包含一个空格的字符串。
- 值为“NULL”的字符串。
- 值为NULL的字符串。

版权所有 © 华为云计算技术有限公司

- 包含一个\字符的字符串。
- 包含一个{字符的字符串。
- 包含一个}字符的字符串。
- 包含一个,字符的字符串。

```
gaussdb=# SELECT '{" ", "NULL", null, "\\", "{", "}", ","}'::varchar[] AS RESULT;
              result
-----------------------------------
 {" ","NULL",NULL,"\\","{","}",","}
(1 row)
```

说明

- 对于数组字符串常量输入，如果数组元素值是空字符串或者包含花括号、分隔符、双引号、反斜杠、空白或者匹配关键字NULL，则这些元素需要使用双引号包裹，在元素值中包含的双引号和反斜杠时，需要额外添加一个反斜杠。
- 关键字NULL不区分大小写。
- 输入会自动忽略没有使用双引号的空白。
- 通常不建议使用字符串常量的方式创建数组，推荐使用ARRAY构造器。

数组类型的字符串输出

一个数组值的输出表现形式由该数组元素类型的输出再加上一些标明该数组结构的修饰组成。这些修饰由围绕在数组值周围的花括号（“{”和“}”）加上相邻项之间的分隔字符组成。在多维数组里， 每个维都有自己级别的花括号，并且在同级相邻的花括号项之间包含分隔符。

数组类型数据包含特殊字符（下述说明中的字符），字符串输出示例：

```
gaussdb=# SELECT ARRAY['{', '}', 'hello, world', '"', '\', '  ', NULL] AS RESULT;
                    array
----------------------------------------------
 {"{","}","hello, world","\"","\\","  ",NULL}
(1 row)
```

说明

对于数组字符串常量输出，当数组元素值是空字符串，或者包含花括号、分隔符、双引号、反斜杠、空白，或者元素为NULL时，这些元素输出时会自动包裹在双引号中，双引号和反斜杠则会通过反斜杠转义并额外输出一个反斜杠。

数组类型的使用

数组类型的使用示例如下：

```
-- 创建有数组类型列的表,并插入一些数据。
gaussdb=# CREATE TABLE orders (
  name  varchar,
  items varchar[]
);
gaussdb=# INSERT INTO orders VALUES('a', ARRAY['苹果', '橘子', '梨']);
gaussdb=# INSERT INTO orders VALUES('b', ARRAY['矿泉水', '可乐', '雪碧']);
gaussdb=# INSERT INTO orders VALUES('c', ARRAY['鼠标', '键盘', '耳机']);
gaussdb=# INSERT INTO orders VALUES('d', '{白菜, 土豆, 茄子}');

-- 查询数据。
gaussdb=# SELECT * FROM orders ORDER BY name;
 name |       items
------+--------------------
 a    | {苹果,橘子,梨}
```

 版权所有 © 华为云计算技术有限公司

```
 b    | {矿泉水,可乐,雪碧}
 c    | {鼠标,键盘,耳机}
 d    | {白菜,土豆,茄子}
(4 rows)

-- 访问数组元素。
gaussdb=# SELECT items[1] FROM orders ORDER BY name;
 items
--------
 苹果
 矿泉水
 鼠标
 白菜
(4 rows)

-- 访问元素超过范围或者访问下标为NULL时会返回NULL。
gaussdb=# SELECT items[4] FROM orders ORDER BY name;
 items
-------




(4 rows)

gaussdb=# SELECT items[null] FROM orders ORDER BY name;
 items
-------




(4 rows)

-- 访问子数组。
gaussdb=# SELECT items[1:2] FROM orders ORDER BY name;
     items
---------------
 {苹果,橘子}
 {矿泉水,可乐}
 {鼠标,键盘}
 {白菜,土豆}
(4 rows)

-- 更新整个数组。
gaussdb=# UPDATE orders SET items = ARRAY['香蕉', '西瓜', '草莓'] WHERE name = 'a';
gaussdb=# SELECT items FROM orders WHERE name = 'a';
      items
------------------
 {香蕉,西瓜,草莓}
(1 row)

-- 更新数组的元素。
gaussdb=# UPDATE orders SET items[1] = '芒果' WHERE name = 'a';
gaussdb=# SELECT items FROM orders WHERE name = 'a';
      items
------------------
 {芒果,西瓜,草莓}
(1 row)

-- 更新数组的元素片段。
gaussdb=# UPDATE orders SET items[1:2] = ARRAY['电脑', '手机'] WHERE name = 'c';
gaussdb=# SELECT items FROM orders WHERE name = 'c';
      items
------------------
 {电脑,手机,耳机}
(1 row)

-- 添加数组元素，所有位于原数组最后一个元素和这个新元素之间的未赋值元素都将设置为NULL。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# UPDATE orders SET items[4] = '显示器' WHERE name = 'c';
gaussdb=# SELECT items FROM orders WHERE name = 'c';
          items
------------------------
 {电脑,手机,耳机,显示器}
(1 row)

gaussdb=# UPDATE orders SET items[6] = '显示器2' WHERE name = 'c';
gaussdb=# SELECT items FROM orders WHERE name = 'c';
                items
-------------------------------------
 {电脑,手机,耳机,显示器,NULL,显示器2}
(1 row)

-- 删除表。
gaussdb=# DROP TABLE orders;
```

1.3.22 向量化引擎支持的数据类型

向量化引擎支持的数据类型如表1-26所示。

表 1-26 向量化引擎支持的数据类型

类别	数据类型	长度
Numeric Types	tinyint [unsigned]	1
	smallint [unsigned]	2
	mediumint [unsigned]	3
	integer [unsigned]	4
	bigint [unsigned]	8
	int16 [signed]	16
	decimal	-1
	numeric	-1
	real	4
	double precision	8
	smallserial	2
	serial	4
	bigserial	8
	largeserial	-1
Monetary Types	money	8
Boolean Types	boolean	1
Character Types	character varying(n), varchar(n)	-1
	character(n), char(n),nchar(n)	n

版权所有 © 华为云计算技术有限公司

类别	数据类型	长度
	character、char	1
	text	-1
	varchar2(n)	-1
	nvarchar2(n)	-1
	clob	-1
Binary Types	blob	-1
	raw	-1
	bytea	-1
	byteawithoutorderwithequalcol	-1
	byteawithoutordercol	-1
Bit String Types	bit(n)	n
	bit varying(n)	-1
Date/Time Types	timestamp with time zone	8
	timestamp without time zone	8
	date	4
	time without time zone	8
	time with time zone	12
	smalldatetime	8
	interval	16
	reltime	4
	abstime	4
	datea	8
Network Address Types	cidr	7 or 19
	inet	7 or 19
Text Retrieval Types	tsvector	-1
	tsquery	-1
JSON Types	json	-1
	jsonb	-1

版权所有 © 华为云计算技术有限公司

类别	数据类型	长度
HyperLoglog Types	hll	-1
Range Types	int4range	-1
	int8range	-1
	numrange	-1
	tsrange	-1
	tstzrange	-1
	daterange	-1
Object Identifier Types	OID	4
	CID	4
	XID	8
	TID	8
	regconfig	4
	regdictionary	4
	regoper	4
	regoperator	4
	regproc	4
	regprocedure	4
	regclass	4
	regtype	4
Pseudo Types	anyelement	4
	anynonarray	4
	anyrange	-1
	cstring	-2
	record	-1
	void	4
XML Types	xml	-1
XMLTYPE Types	xmltype	-1
Hash Types	hash16	8
Array Types	tinyint[]	-1
	smallint[]	-1

版权所有 © 华为云计算技术有限公司

类别	数据类型	长度
	mediumint[]	-1
	integer[]	-1
	bigint[]	-1
	int16[]	-1
	decimal[]	-1
	numeric[]	-1
	real[]	-1
	double precision[]	-1
	money[]	-1
	boolean[]	-1
	varchar[]	-1
	char(n)[]	-1
	char[]	-1
	text[]	-1
	varchar2(n)[]	-1
	nvarchar2(n)[]	-1
	clob[]	-1
	blob[]	-1
	raw[]	-1
	bytea[]	-1
	byteawithoutorderwithequalcol[]	-1
	byteawithoutordercol[]	-1
	bit(n)[]	-1
	bit varying(n)[]	-1
	timestamp with time zone[]	-1
	timestamp without time zone[]	-1
	date[]	-1
	time without time zone[]	-1

版权所有 © 华为云计算技术有限公司

类别	数据类型	长度
	time with time zone[]	-1
	smalldatetime[]	-1
	interval[]	-1
	reltime[]	-1
	abstime[]	-1
	datea[]	-1
	cidr[]	-1
	inet[]	-1
	tsvector[]	-1
	tsquery[]	-1
	json[]	-1
	jsonb[]	-1
	hll[]	-1
	int4range[]	-1
	int8range[]	-1
	numrange[]	-1
	tsrange[]	-1
	tstzrange[]	-1
	daterange[]	-1
	OID[]	-1
	CID[]	-1
	XID[]	-1
	TID[]	-1
	regconfig[]	-1
	regdictionary[]	-1
	regoper[]	-1
	regoperator[]	-1
	regproc[]	-1
	regprocedure[]	-1
	regclass[]	-1

版权所有 © 华为云计算技术有限公司

类别	数据类型	长度
	regtype[]	-1
	cstring[]	-1
	record[]	-1
	xml[]	-1
	xmltype[]	-1
	hash16[]	-1
	hash32[]	-1
	aclitem[]	-1
	name[]	-1
	macaddr[]	-1
	UUID[]	-1
	point[]	-1
	lseg[]	-1
	box[]	-1
	path[]	-1
	polygon[]	-1
	circle[]	-1
Vector Types	floatvector	-1
	boolvector	-1
Other Types	enum	enum 类型由若干个标签构成的列表，每一个标签值都是一个非空字符串，且字符串长度必须不超过63个字节。

说明

enum类型当前仅支持如下方式创建，详情请参见**CREATE TYPE**。

```
CREATE TYPE name AS ENUM (['label' [, ...]]);
```

1.3.23 ROWID 类型

ROWID数据类型用于标识表中的数据行，包含定位该行所需的关键信息，可实现对表中数据行的快速定位。

ROWID数据类型由29个BASE64编码的字符组成，各个字符串的含义如**表1-27**所示：

版权所有 © 华为云计算技术有限公司

表 1-27 ROWID 类型字段含义

字段	含义	字符长度	取值范围（BASE64编码）
reserve_flag	保留位。	6	当前版本0（AAAAAA）。
node_slice	节点的分片编号。	3	最小值0（AAA），最大值65535（P//），实际最大值为集群分片数量。
bucket_id	hash bucket表的bucket编号。	3	最小值0（AAA），最大值32767（H//），实际最大值为表的最大bucket编号。对于非hash bucket表而言，该字段值固定为65535（P//）。
table_oid	表的oid。	6	最小值0（AAAAAA），最大值4294967295（D/////）。
row_no	行的编号。	11	最小值0（AAAAAAAAAAA），最大值9223372036854775807（H//////////）。

说明

ROWID类型不支持以下操作：

- ROWID类型不支持作为表的分区键、分布键。
- ROWID类型不支持HTAP。
- ROWID类型仅支持ORA兼容性数据库。
- ROWID类型不支持数据脱敏策略。
- ROWID类型不可作为自定义范围类型的subtype。详细可见**范围类型**章节内容。
- ROWID类型仅支持JDBC驱动。

示例

```
gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c ora_test
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora_test" as user "$user".

-- 创建表
ora_test=# CREATE TABLE rowid_test(id INT, rwd ROWID) DISTRIBUTE BY HASH(id);
CREATE TABLE

-- 插入数据，值为NULL
ora_test=# INSERT INTO rowid_test VALUES(1, NULL);
INSERT 0 1

-- 插入数据，满足ROWID格式
ora_test=# INSERT INTO rowid_test VALUES(2, 'AAAAAAAAAP//BBBBBAAAAAAAAAAAC');
INSERT 0 1
ora_test=# INSERT INTO rowid_test VALUES(3, 'AAAAAAAAAP//BBBd+AAAAAAAA901/');
INSERT 0 1
ora_test=# INSERT INTO rowid_test VALUES(4, 'AAAAAAAAAP//A12i+AAAAAAAAItn/');
INSERT 0 1
ora_test=# INSERT INTO rowid_test VALUES(5, 'AAAAAAAAAP//CC+uJAAAAAAGASIi9');
INSERT 0 1
```


版权所有 © 华为云计算技术有限公司

```
-- 插入数据，不满足ROWID格式
ora_test=# INSERT INTO rowid_test VALUES(6, 'AAAAAAAAAP//BBBd+AAAAAAAA@@@@');
ERROR:  Invalid rowid.
LINE 1: INSERT INTO rowid_test VALUES(6, 'AAAAAAAAAP//BBBd+AAAAAAAA@...
                                         ^
CONTEXT:  referenced column: rwd

-- 插入数据，不满足ROWID值范围
ora_test=# INSERT INTO rowid_test VALUES(6, 'AAAAAAAAAP/////d+AAAAAAAA901/');
ERROR:  Invalid rowid.
LINE 1: INSERT INTO rowid_test VALUES(6, 'AAAAAAAAAP/////d+AAAAAAAA9...
                                         ^
CONTEXT:  referenced column: rwd

-- 查看数据
ora_test=# SELECT * FROM rowid_test;
 id |              rwd
----+-------------------------------
  1 |
  2 | AAAAAAAAAP//BBBBBAAAAAAAAAAAC
  3 | AAAAAAAAAP//BBBd+AAAAAAAA901/
  4 | AAAAAAAAAP//A12i+AAAAAAAAItn/
  5 | AAAAAAAAAP//CC+uJAAAAAAGASIi9
(5 rows)

-- 删除表
ora_test=# DROP TABLE IF EXISTS rowid_test;
DROP TABLE
ora_test=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "$user".
gaussdb=# DROP DATABASE ora_test;
DROP DATABASE
```

操作符

ROWID类型支持比较操作符（=、<>、<、>、<=、>=），当与不支持隐式转换成ROWID的数据类型进行比较操作时，报错“Inconsistent datatypes”。比较操作符详情请参见**比较操作符**章节。

ROWID类型的比较规则是按照node_slice、table_oid、bucket_id、row_no字段数值依次进行比较。

示例

```
gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c ora_test
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora_test" as user "$user".

-- ROWID类型比较
ora_test=# CREATE OR REPLACE FUNCTION func1 RETURN ROWID AS
ora_test$# DECLARE
ora_test$#     a ROWID;
ora_test$#     b ROWID;
ora_test$# BEGIN
ora_test$#     a := 'AAAAAAAAAP//ABPH//AAAAAAAACAB';
ora_test$#     b := 'AAAAAAAAAP//ABPH//AAAAAAAACAA';
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= b)::text);
ora_test$#     RETURN a;
ora_test$# END;
ora_test$# /
```


版权所有 © 华为云计算技术有限公司

```
CREATE FUNCTION
ora_test=# SELECT func1();
false
true
true
false
false
true
            func1
------------------------------
 AAAAAAAAAP//ABPH//AAAAAAAACAB
(1 row)

-- ROWID类型与其他可隐式转换类型的比较
ora_test=# CREATE OR REPLACE FUNCTION func1 RETURN ROWID AS
ora_test$# DECLARE
ora_test$#     a ROWID;
ora_test$#     b TEXT;
ora_test$#     c VARCHAR(30);
ora_test$#     d VARCHAR2;
ora_test$#     e CHAR(29);
ora_test$#     F NVARCHAR2;
ora_test$# BEGIN
ora_test$#     --数据准备
ora_test$#     a:='AAAAAAAAAP//ABPH//AAAAAAAACAB';
ora_test$#     b:='AAAAAAAAAP//ABPH//AAAAAAAACAA';
ora_test$#     c:='AAAAAAAAAP//ABPH//AAAAAAAACAA';
ora_test$#     d:='AAAAAAAAAP//ABPH//AAAAAAAACAA';
ora_test$#     e:='AAAAAAAAAP//ABPH//AAAAAAAACAA';
ora_test$#     F:='AAAAAAAAAP//ABPH//AAAAAAAACAA';
ora_test$#
ora_test$#     --与text比较
ora_test$#     DBE_OUTPUT.PRINT_LINE('text:');
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= b)::text);
ora_test$#
ora_test$#     --与varchar比较
ora_test$#     DBE_OUTPUT.PRINT_LINE('varchar:');
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = c)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> c)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > c)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < c)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= c)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= c)::text);
ora_test$#
ora_test$#     --与varchar2比较
ora_test$#     DBE_OUTPUT.PRINT_LINE('varchar2:');
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = d)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> d)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > d)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < d)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= d)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= d)::text);
ora_test$#     --与char比较
ora_test$#     DBE_OUTPUT.PRINT_LINE('char:');
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = e)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> e)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > e)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < e)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= e)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= e)::text);
ora_test$#
ora_test$#     --与nvarchar2比较
ora_test$#     DBE_OUTPUT.PRINT_LINE('nvarchar2:');
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = f)::text);
```


版权所有 © 华为云计算技术有限公司

```
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> f)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > f)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < f)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= f)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= f)::text);
ora_test$#
ora_test$#     RETURN a;
ora_test$# END;
ora_test$# /
CREATE FUNCTION
ora_test=# SELECT func1();
text:
false
true
true
false
false
true
varchar:
false
true
true
false
false
true
varchar2:
false
true
true
false
false
true
char:
false
true
true
false
false
true
nvarchar2:
false
true
true
false
false
true
            func1
-------------------------------
 AAAAAAAAAP//ABPH//AAAAAAAACAB
(1 row)

-- ROWID类型与其他不可隐式转换类型的比较
ora_test=# CREATE OR REPLACE FUNCTION func1 RETURN ROWID AS
ora_test$# DECLARE
ora_test$#     a ROWID;
ora_test$#     b INT;
ora_test$# BEGIN
ora_test$#     --数据准备
ora_test$#     a := 'AAAAAAAAAP//ABPH//AAAAAAAACAB';
ora_test$#     b := 1;
ora_test$#
ora_test$#     --与int比较
ora_test$#     DBE_OUTPUT.PRINT_LINE((a = b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <> b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a > b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a < b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a <= b)::text);
ora_test$#     DBE_OUTPUT.PRINT_LINE((a >= b)::text);
ora_test$#     RETURN a;
```


版权所有 © 华为云计算技术有限公司

```
ora_testb$# END;
ora_test$# /
CREATE FUNCTION

ora_test=# SELECT func1();
ERROR:  Inconsistent datatypes: expected rowid got integer
CONTEXT:  SQL statement "CALL dbe_output.print_line((a = b)::text)"
PL/pgSQL function func1() line 11 at PERFORM
referenced column: func1

ora_test=# DROP FUNCTION IF EXISTS func1();
DROP FUNCTION
ora_test=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "$user".
gaussdb=# DROP DATABASE ora_test;
DROP DATABASE
```

索引

ROWID类型普通列支持BTREE/UBTREE索引，详情请参见**CREATE INDEX**章节内容。在使用ROWID类型普通列进行JOIN操作时，支持HASH JOIN算子，详情请参见**连接查询**章节内容。

示例

```
gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c ora_test
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora_test" as user "$user".

ora_test=# CREATE TABLE t1(b ROWID, c INT) WITH (storage_type=ustore);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# CREATE TABLE t2(b ROWID, c INT) WITH (storage_type=astore);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# INSERT INTO t1 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAAC', generate_series(1,100));
INSERT 0 100
ora_test=# INSERT INTO t2 VALUES('AAAAAAAAAP//BBBBBAAAAACdAAAAC', generate_series(1,100));
INSERT 0 100
ora_test=# --BTREE索引
ora_test=# CREATE INDEX t1_rowid_index ON t1 USING BTREE (b);
NOTICE:  Index type btree is converted to index type ubtree.
CREATE INDEX
ora_test=# CREATE INDEX t2_rowid_index ON t2 USING BTREE (b rowid_ops);
CREATE INDEX
ora_test=# \d t1
      Table "public.t1"
 Column |  Type   | Modifiers
--------+---------+-----------
 b      | rowid   |
 c      | integer |
Indexes:
    "t1_rowid_index" ubtree (b) WITH (storage_type=USTORE) TABLESPACE pg_default

ora_test=# \d t2
      Table "public.t2"
 Column |  Type   | Modifiers
--------+---------+-----------
 b      | rowid   |
 c      | integer |
Indexes:
    "t2_rowid_index" btree (b) TABLESPACE pg_default
```


版权所有 © 华为云计算技术有限公司

```
ora_test=# DROP INDEX t1_rowid_index;
DROP INDEX
ora_test=# DROP INDEX t2_rowid_index;
DROP INDEX
ora_test=# CREATE INDEX t1_rowid_index ON t1 USING UBTREE (b);
CREATE INDEX
ora_test=# CREATE INDEX t2_rowid_index on t2 USING UBTREE (b rowid_ops);
NOTICE:  Index type ubtree is converted to index type btree.
CREATE INDEX
ora_test=# \d t1
      Table "public.t1"
 Column |  Type   | Modifiers
--------+---------+-----------
 b      | rowid   |
 c      | integer |
Indexes:
    "t1_rowid_index" ubtree (b) WITH (storage_type=USTORE) TABLESPACE pg_default

ora_test=# \d t2
      Table "public.t2"
 Column |  Type   | Modifiers
--------+---------+-----------
 b      | rowid   |
 c      | integer |
Indexes:
    "t2_rowid_index" btree (b) TABLESPACE pg_default

ora_test=# DROP INDEX t1_rowid_index;
DROP INDEX
ora_test=# DROP INDEX t2_rowid_index;
DROP INDEX
ora_test=# ANALYZE t1;
ANALYZE
ora_test=# ANALYZE t2;
ANALYZE
--HASH JOIN 算子
ora_test=# EXPLAIN VERBOSE SELECT * FROM t1 JOIN t2 ON t1.b = t2.b;
                                 QUERY PLAN
-----------------------------------------------------------------------------------------
 Streaming (type: GATHER)  (cost=11.69..14.48 rows=2 width=56)
   Output: t1.b, t1.c, t2.b, t2.c
   Node/s: All datanodes
   ->  Hash Join  (cost=11.62..14.38 rows=2 distinct=[200, 1] width=56)
         Output: t1.b, t1.c, t2.b, t2.c, (Expression Flatten Optimized)
         Hash Cond: (t1.b = t2.b), (Expression Flatten Optimized)
         ->  Seq Scan on public.t1  (cost=0.00..2.50 rows=100 width=28)
               Output: t1.b, t1.c
               Distribute Key: t1.c
         ->  Hash  (cost=10.37..10.37 rows=200 width=28)
               Output: t2.b, t2.c
               ->  Streaming(type: BROADCAST)  (cost=0.00..10.37 rows=200 width=28)
                     Output: t2.b, t2.c
                     Spawn on: All datanodes
                     Consumer Nodes: All datanodes
                     ->  Seq Scan on public.t2  (cost=0.00..2.50 rows=100 width=28)
                           Output: t2.b, t2.c
                           Distribute Key: t2.c
(18 rows)
ora_test=# DROP TABLE IF EXISTS t1;
DROP TABLE
ora_test=# DROP TABLE IF EXISTS t2;
DROP TABLE
ora_test=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "$user".
gaussdb=# DROP DATABASE ora_test;
DROP DATABASE
```


版权所有 © 华为云计算技术有限公司

类型转换

ROWID类型和字符串类型允许进行隐式类型转换。

示例

```
gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c ora_test
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora_test" as user "$user".

ora_test=# CREATE TABLE t1(b ROWID, c int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# INSERT INTO t1 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::BPCHAR(29));
INSERT 0 1
ora_test=# INSERT INTO t1 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::VARCHAR(29));
INSERT 0 1
ora_test=# INSERT INTO t1 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::NVARCHAR2(29));
INSERT 0 1
ora_test=# INSERT INTO t1 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::TEXT);
INSERT 0 1
ora_test=# CREATE TABLE t2(b BPCHAR(29));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'b' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# INSERT INTO t2 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::ROWID);
INSERT 0 1
ora_test=# CREATE TABLE t3(b VARCHAR(29));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'b' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# INSERT INTO t3 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::ROWID);
INSERT 0 1
ora_test=# CREATE TABLE t4(b NVARCHAR2(29));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'b' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# INSERT INTO t4 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::ROWID);
INSERT 0 1
ora_test=# CREATE TABLE t5(b TEXT);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'b' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_test=# INSERT INTO t5 VALUES('AAAAAAAAAP//BBBBBAAAAAAAAAAAC'::ROWID);
INSERT 0 1
gaussdb=# DROP TABLE IF EXISTS t1;
DROP TABLE
gaussdb=# DROP TABLE IF EXISTS t2;
DROP TABLE
gaussdb=# DROP TABLE IF EXISTS t3;
DROP TABLE
gaussdb=# DROP TABLE IF EXISTS t4;
DROP TABLE
gaussdb=# DROP TABLE IF EXISTS t5;
DROP TABLE
ora_test=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "$user".
gaussdb=# DROP DATABASE ora_test;
DROP DATABASE
```

1.3.24 向量数据类型

向量数据类型包含floatvector和boolvector两种类型。

版权所有 © 华为云计算技术有限公司

- floatvector数据类型是指多维数据中含有的数据为float类型，例如[1.0,3.0,11.0,110.0,62.0,22.0,4.0]。

  说明

  - floatvector成员仅支持单精度。单精度范围为-3.402E+38~+3.402E+38，6位十进制数字精度。
  - floatvector不支持NULL、Nan、Inf作为元素，当向量中含有NULL值，数据库会报错。
  - floatvector不能为NULL，当插入、更新或转换NULL值作为向量数据时，数据库会报错。
  - 在某些场景下（执行计划走stream计划时），floatvector成员存储数字精度受GUC参数extra_float_digits控制，这可能导致相同数据在计算时结果不一致，设置extra_float_digits = 3可规避此问题。
- boolvector数据类型是指多维数据中含有的数据为bool类型，例如[0,0,0,0,0,1,0,0,1,0,0,0]。

  说明

  - boolvector成员可使用t(T)\f(F)、y(Y)\n(N)、1\0、true\false、yes\no、on\off等方式表达布尔型数据。
  - boolvector不支持NULL、Nan、Inf作为元素，当向量中含有NULL值，数据库会报错。
  - boolvector不能为NULL，当插入、更新或转换NULL值作为向量数据时，数据库会报错。

向量类型的使用

向量类型的使用示例如下：

```
-- 创建包含向量类型的表，同时设定数据维度。建表时向量类型必须要指定维度。
gaussdb=# CREATE TABLE t1(id int unique, repr floatvector(4));
gaussdb=# CREATE TABLE t2(id int unique, repr boolvector(3));

-- 插入数据。
gaussdb=# INSERT INTO t1 VALUES(0, '[30, 12, 12, 25]');
gaussdb=# INSERT INTO t2 VALUES(1, '[1, 0, 1]');

-- 删除表。
gaussdb=# DROP TABLE t1;
gaussdb=# DROP TABLE t2;
```

1.4 字符集和字符序

字符集（Character Set）是字符的编码规则，字符序（Collation）是字符的排序规则，本章主要对GaussDB在MYSQL兼容模式（即sql_compatibility = 'MYSQL'）下的字符集、字符序进行介绍，以下介绍的字符集、字符序规则和语法仅在MYSQL兼容模式下支持。

GaussDB目前所支持的字符集可以参考**CREATE DATABASE**章节中ENCODING部分介绍，所支持的字符序可以通过查询系统表PG_COLLATION获取。

其中，部分字符集在MYSQL模式下存在默认字符序，可以参考**表1-398**。

字符集和字符序的说明如下：

- 每个字符集都有一个或多个字符序，只有一个字符序为字符集的默认字符序。
- 每个字符序仅有一个相关联的字符集。
- 相同数据使用不同的字符序排序时，结果可能会产生差异。

版权所有 © 华为云计算技术有限公司

- 在GaussDB中，utf8mb4与utf8为同一字符集。
- 在GaussDB的MYSQL模式（即sql_compatibility = 'MYSQL'）下，binary字符集与SQL_ASCII字符集为同一字符集。
- 建议将表字段和server_encoding配置为相同字符集，以避免转码带来的性能损耗。

GaussDB数据库支持以下功能：

- 支持多种字符集存储字符串。
- 支持使用字符序比较字符串。
- 支持数据库级、模式级、表级、字段级字符集和字符序。

说明

暂不支持在同一服务器、同一数据库、同一张表、同一条SQL语句中混合使用不同字符集和字符序的字符串。

1.4.1 客户端连接的字符集和字符序

含有字符集属性的数据在服务端和客户端之间传输时，会自动转换编码。服务端接收到客户端发送的SQL语句后，会将其由客户端字符集client_encoding向数据库字符集server_encoding转换编码。查询结果数据发送到客户端之前，也会将数据向客户端字符集client_encoding转换编码。

系统参数说明

- **server_encoding**

  创建数据库时指定的字符集，详见**CREATE DATABASE**章节。

- **client_encoding**

  客户端的字符集，可以通过SET NAMES语句修改，详见**SET**章节，参数说明请参见**client_encoding**。

- **character_set_connection**

  SQL语句中未指定字符集的字符串常量的默认字符集，参数说明请参见**character_set_connection**。

- **collation_connection**

  SQL语句中未指定字符序的字符串常量的默认字符序，参数说明请参见**collation_connection**。

- **character_set_results**

  返回结果的字符集，参数说明请参见**character_set_results**。

说明

- 对于一个由非字符类型对象转换为字符类型的表达式，其结果的字符集和字符序为character_set_connection和collation_connection。
- 字符类型的绑定参数的字符集和字符默认为系统参数character_set_connection和collation_connection设置的值，绑定参数输入任何值均将认作上述字符集。
- 字符数据转换编码的过程中会校验字符的编码，如果不符合，将会抛出异常，返回ERROR信息。

版权所有 © 华为云计算技术有限公司

1.4.2 数据库级字符集和字符序

创建一个新的数据库时，可以指定数据库的字符集字符序，创建数据库操作请参见**CREATE DATABASE**。

```
CREATE DATABASE [IF NOT EXISTS] database_name
            [ ENCODING [=] encoding ] |
            [ LC_COLLATE [=] lc_collate ] |
            [ LC_CTYPE [=] lc_ctype ] |;
```

参数说明：

- **database_name**

  数据库名称。

  取值范围：字符串，要符合标识符的命名规范。

- **ENCODING [=] encoding**

  指定新数据库使用的字符编码，可以是字符串（如'SQL_ASCII'）、整数编号。

- **LC_COLLATE [=] 'lc_collate'**

  指定新数据库使用的字符集。例如，通过lc_collate = 'zh_CN.gbk'设定该参数。

  该参数会影响字符串的排序行为（包括使用ORDER BY的排序结果，以及在文本列上使用索引的排序顺序）。默认使用模板数据库的排序规则。

  取值范围：操作系统支持的字符集。

- **LC_CTYPE [=] 'lc_ctype'**

  指定新数据库使用的字符分类。例如，通过lc_ctype = 'zh_CN.gbk'设定该参数。该参数的使用会影响到字符的分类，包括大小写字母和数字。默认使用模板数据库的字符分类。

  取值范围：操作系统支持的字符分类。

  说明

  - 所有模式均可使用数据库级字符集、字符序语法，详细语法请参见**CREATE DATABASE**。
  - LC_COLLATE、LC_CTYPE参数不支持指定MYSQL模式特有的字符序，参数的取值范围取决于本地环境支持的字符集，可通过locale -a查看。

1.4.3 模式级字符集和字符序

创建模式并指定默认字符集和字符序，创建模式请参见**CREATE SCHEMA**。

```
CREATE SCHEMA schema_name
 [ [DEFAULT] CHARACTER SET | CHARSET [ = ] default_charset ]
 [ [DEFAULT] COLLATE [ = ] default_collation ];
```

修改模式的默认字符集、字符序属性，修改模式请参见**ALTER SCHEMA**。

```
ALTER SCHEMA schema_name
    [ [DEFAULT] CHARACTER SET | CHARSET [ = ] default_charset ] [ [DEFAULT] COLLATE [ = ]
default_collation ];
```

参数说明：

- **schema_name**

  模式名称。

  取值范围：字符串，要符合**标识符命名规范**。

版权所有 © 华为云计算技术有限公司

- **default_charset**

  指定模式的默认字符集，单独指定此参数时，模式的默认字符序将被设置为指定字符集的默认字符序。

- **default_collation**

  指定模式的默认字符序，单独指定此参数时，模式的默认字符集将被设置为指定字符序对应的字符集。

GaussDB通过以下方式选择模式的字符集和字符序：

- 如果同时指定了default_charset和default_collation，则使用字符集default_charset和字符序default_collation，且default_charset和default_collation需要对应，否则会产生报错。
- 如果仅指定了default_charset，则使用字符集default_charset及其默认字符序。
- 如果仅指定了default_collation，则使用default_collation字符序和其对应的字符集。
- 如果既不指定default_charset也不指定default_collation，则该模式没有默认字符集和默认字符序。

说明

- 仅设置default_charset时，仅支持指定为带有默认字符序的字符集，如果指定的字符集没有默认字符序则报错。
- default_collation仅支持指定为MYSQL模式（即sql_compatibility = 'MYSQL'）下的字符序，指定其他字符序报错。
- 暂不支持新建模式的字符集与数据库的server_encoding不同。

示例：

```
-- 仅设置字符集，字符序为字符集的默认字符序
gaussdb=# CREATE SCHEMA test CHARSET utf8;

-- 仅设置字符序，字符集为字符序关联的字符集
gaussdb=# CREATE SCHEMA test COLLATE utf8_bin;

-- 同时设置字符集与字符序，字符集和字符序需对应
gaussdb=# CREATE SCHEMA test CHARSET utf8 COLLATE utf8_bin;

-- 将test的默认字符集修改为utf8mb4，默认字符序修改为utf8mb4_bin。
gaussdb=# ALTER SCHEMA test CHARSET utf8mb4 COLLATE utf8mb4_bin;
```

1.4.4 表级字符集和字符序

设置表的默认字符集和默认字符序，创建表操作请参见**CREATE TABLE**。

```
CREATE TABLE table_name (column_list)
    [ [DEFAULT] CHARACTER SET | CHARSET [ = ] default_charset ]
    [ [DEFAULT] COLLATE [ = ] default_collation ]
```

修改表的默认字符集和默认字符序，修改不会影响表中已存在的列，修改表操作请参见**ALTER TABLE**。

```
ALTER TABLE table_name
    [ [DEFAULT] CHARACTER SET | CHARSET [ = ] default_charset ]
    [ [DEFAULT] COLLATE [ = ] default_collation ]
```


版权所有 © 华为云计算技术有限公司

说明

暂不支持修改表的默认字符集和默认字符序为指定的值，同时将表中所有字符类型的字段的字符集和字符序设置为指定值，并将字段数据转换为新字符集编码。语法如下：

```
ALTER TABLE table_name
  CONVERT TO CHARACTER SET | CHARSET charset [ COLLATE collation ]
```

参数说明

- **table_name**

  表名称。

- **default_charset**

  指定表的默认字符集，单独指定此参数时，表的默认字符序将被设置为指定字符集的默认字符序。

- **default_collation**

  指定表的默认字符序，单独指定此参数时，表的默认字符集将被设置为指定字符序对应的字符集。

GaussDB通过以下方式选择表的字符集和字符序：

- 如果同时指定了default_charset和default_collation，则使用字符集default_charset和字符序default_collation，且default_charset和default_collation需要匹配，否则会产生报错。
- 如果仅指定了default_charset，则使用字符集default_charset及其默认字符序。
- 如果仅指定了default_collation，则使用default_collation字符序及其对应的字符集。
- 如果未指定default_charset和default_collation，则使用该表所在的模式的默认字符集和默认字符序。

说明

- 仅设置default_charset时，仅支持指定为带有默认字符序的字符集，如果指定的字符集没有默认字符序则报错。
- default_collation仅支持指定为MYSQL模式（即sql_compatibility = 'MYSQL'）下的字符序，指定其他字符序报错。
- 如果表的默认字符序为binary，则表中未指定字符序的文本类型会转换为对应的二进制类型，且字符序设置为binary。
- 暂不支持设置表的默认字符集与数据库的server_encoding不同。
- 暂不支持将表中字符类型字段的数据转化编码。

示例：

```
-- 仅设置字符集，字符序为字符集的默认字符序
gaussdb=# CREATE TABLE test(c1 text) CHARSET utf8;

-- 仅设置字符序，字符集为字符序关联的字符集
gaussdb=# CREATE TABLE test(c1 text) COLLATE utf8_bin;

-- 同时设置字符集与字符序，字符集和字符序需对应
gaussdb=# CREATE TABLE test(c1 text)  CHARSET utf8 COLLATE utf8_bin;

-- 修改表的默认字符集为utf8mb4，默认字符序为utf8mb4_bin
gaussdb=# ALTER TABLE test CHARSET utf8mb4 COLLATE utf8mb4_bin;

-- 删除表。
gaussdb=# DROP TABLE test;
```


版权所有 © 华为云计算技术有限公司

1.4.5 列级字符集和字符序

每个字符串类型（即类型为CHAR、VARCHAR、TEXT等字符串类型）的列都可以配置列级字符集和字符序。

```
CREATE TABLE table_name (
    column_name data_type
 [ CHARACTER SET | CHARSET charset ]
    [ COLLATE collation ]
);
```

参数说明：

- **table_name**

  表名称。

- **data_type**

  字段的数据类型，仅字符串类型支持配置字符集和字符序。

- **CHARACTER SET | CHARSET charset**

  指定列的字符集，单独指定时会将列的字符序设置为指定字符集的默认字符序。

- **COLLATE collation**

  COLLATE子句指定列的字符序（该列的数据类型必须支持字符序）。如果没有指定，则使用默认的排序规则。

GaussDB通过以下方式选择表字段的字符集和字符序：

- 如果同时指定了charset和collation，则使用字符集charset和字符序collation，且charset和collation需要对应，否则会产生报错。
- 如果仅指定了charset，则使用字符集charset及其默认字符序。
- 如果仅指定了collation，则使用与collation指定的字符序及其对应的字符集。
- 如果既不指定charset也不指定collation，则使用表的默认字符集和默认字符序。

说明

- 仅设置charset时，仅支持指定为带有默认字符序的字符集，如果指定的字符集没有默认字符序则报错。
- collation仅支持指定为MYSQL模式下的字符序，指定其他字符序报错。
- 如果表字段为文本类型且指定的字符序为binary，则文本类型会转换为对应的二进制类型，字符序为指定的binary字符序。
- 分区表的分区键的字符集必须与数据库字符集相同。
- 暂不支持设置表字段的默认字符集与数据库字符集server_encoding不同。

示例：

```
-- 仅设置字符集，字符序为字符集的默认字符序。
gaussdb=# CREATE TABLE test(c1 text CHARSET utf8);

-- 仅设置字符序，字符集为字符序关联的字符集。
gaussdb=# CREATE TABLE test(c1 text COLLATE utf8_bin);

-- 同时设置字符集与字符序，字符集和字符序需对应。
gaussdb=# CREATE TABLE test(c1 text CHARSET utf8 COLLATE utf8_bin);
```

1.4.6 字符类型表达式的字符集和字符序

每一个字符类型的表达式都含有字符集和字符序属性。

版权所有 © 华为云计算技术有限公司

在MYSQL模式（即sql_compatibility = 'MYSQL'）的数据库下，当设置参数b_format_version='5.7'、b_format_dev_version='s2'时，字符串常量的默认字符集和字符序由系统参数character_set_connection和collation_connection决定。未设置时，其默认字符集与数据库字符集server_encoding相同，其默认字符序为default。

字符集语法：

GaussDB暂不支持用以下语法指定字符串常量的字符集。

```
[_charset_name]'string'
```

字符序语法：

其他字符串类型的表达式也可以指定字符序。

```
EXPRESSION [COLLATE collation_name]
```

参数说明：

COLLATE collation_name

指定字符序的名称，用于设置该字符串的字符序属性。

- 表达式的数据类型必须是支持字符序的数据类型。
- 指定的字符序必须是表达式的字符集允许的字符序。

示例：

```
--使用COLLATE语句指定字符序
gaussdb=# SELECT 'a' COLLATE utf8mb4_general_ci = 'A';
```

1.4.7 字符集和字符序合并规则

在MYSQL兼容模式（即sql_compatibility = 'MYSQL'）的数据库下，且设置参数b_format_version='5.7'和b_format_dev_version='s2'时，系统会根据一定优先级处理不同字符集和字符序的表达式，从而确定字符串比较运算时使用的字符序和表达式的字符集。

字符序优先级

不同表达式的字符序优先级由高到低排列如下：

- COLLATE语法拥有最高优先级。
- 存在字符序冲突的表达式（如：两个不同字符序的字符串拼接表达式）。
- 支持字符序的数据类型的列、用户自定义变量、存储过程参数、CASE表达式等。
- 特定的系统函数（如：version()和opengauss_version()函数表达式）。
- 字符串常量和绑定参数。
- NULL表达式。
- 如果一个表达式的数据类型不支持字符序，那么这个表达式的优先级最低。

当两个表达式字符序不同时，使用字符序优先级最高的表达式的字符序。

示例：

```
gaussdb=# CREATE TABLE t_utf8(c1 varchar(16) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin);
gaussdb=# INSERT INTO t_utf8 VALUES('STRING');

-- 比较时使用utf8mb4_bin字符序，结果为false。
gaussdb=# SELECT c1 = 'string' AS result FROM t_utf8;
```

 版权所有 © 华为云计算技术有限公司

```
 result
--------
 f
(1 row)

-- 比较时使用utf8mb4_general_ci字符序，结果为true。
gaussdb=# SELECT c1 = 'string' COLLATE utf8mb4_general_ci AS result FROM t_utf8;
 result
--------
 t
(1 row)

-- 将绑定参数“$1”的字符序定义为collation_connection。
gaussdb=# PREPARE test_collation(text) AS SELECT c1 = $1 AS result FROM t_utf8;

-- 绑定参数字符序与字符串常量同级别，即使传入的表达式含有显式的字符序，比较时仍然采用c1的字符序。
gaussdb=# EXECUTE test_collation('string' COLLATE utf8mb4_general_ci);
 result
--------
 f
(1 row)

-- CASE表达式与c1列同级别，即使表达式含有显式的字符序，比较时仍然采用c1的字符序，二者不相等，输出“same level”。
gaussdb=# SELECT CASE 'string' COLLATE utf8mb4_general_ci WHEN c1 THEN 'different level' ELSE 'same level' END AS result FROM t_utf8;
   result
------------
 same level
(1 row)

-- IN子查询与c1列同级别，即使表达式含有显式的字符序，比较时仍然采用c1的字符序，二者不相等。
gaussdb=# SELECT c1 FROM t_utf8 WHERE c1 in (SELECT 'string' COLLATE utf8mb4_general_ci);
 c1
----
(0 rows)
```

当两个相同优先级的表达式字符序不同时，采用以下方式处理：

- 如果两者字符集相同，优先使用后缀为_bin的字符序。
- 如果两者字符集相同，优先不使用default字符序。
- 如果不符合上述情况，两个表达式将被标记为字符序冲突，字符序将被标记为无效。
  - 因COLLATE语法指定同字符集不同字符序产生的冲突，将产生异常。
  - 若冲突的两个字符序均为MYSQL模式（即sql_compatibility = 'MYSQL'）下支持的字符序，将产生异常。
  - 产生冲突的字符集与数据库字符集server_encoding不同时，将产生异常。
  - 无效字符序被用作排序运算时（如：>、<等），将产生异常。
  - 字符串等值比较时，若字符序无效，将直接作为二进制比较是否相等。

示例：

```
gaussdb=# CREATE TABLE t_utf8mb4_charset(
    c_utf8_bin varchar(16) character set utf8mb4 collate utf8mb4_bin,
    c_utf8_uni varchar(16) character set utf8mb4 collate utf8mb4_unicode_ci,
    c_utf8_gen varchar(16) character set utf8mb4 collate utf8mb4_general_ci);
gaussdb=# INSERT INTO t_utf8mb4_charset VALUES('STRING', 'String', 'string');

-- 优先使用utf8mb4_bin字符序比较，结果为false。
gaussdb=# SELECT c_utf8_bin = c_utf8_uni FROM t_utf8mb4_charset;

-- 字符序冲突，进行二进制比较，结果为false。
gaussdb=# SELECT c_utf8_uni = c_utf8_gen FROM t_utf8mb4_charset;
```


版权所有 © 华为云计算技术有限公司

```
-- 显式指定的字符序冲突，抛出异常。
gaussdb=# SELECT c_utf8_uni COLLATE utf8mb4_unicode_ci = c_utf8_gen COLLATE utf8mb4_general_ci
FROM t_utf8mb4_charset;
```

说明

- 只有字符串类型（不包括"char"、name、clob）的对象和表达式的字符集可以与数据库的字符集不同。
- ARRAY、XML、JSON、TSVECTOR等数据类型含有文本数据，这些数据类型的对象和表达式中的文本数据的字符集必须与数据库字符集一致。
- 在字符集和字符序合并规则中，认为C、POSIX、DEFAULT字符序对应的字符集为server_encoding。

1.5 常量和宏

GaussDB支持的常量和宏请参见**表1-28**。

表 1-28 常量和宏

参数	描述	示例
CURRENT_CATALOG	当前数据库。	testdb=# SELECT CURRENT_CATALOG; current_database ------------------ testdb (1 row)
CURRENT_ROLE	当前用户。	gaussdb=# SELECT CURRENT_ROLE; current_user -------------- omm (1 row)
CURRENT_SCHEMA	当前数据库模式。	gaussdb=# SELECT CURRENT_SCHEMA; current_schema ---------------- public (1 row)
CURRENT_USER	当前用户。	gaussdb=# SELECT CURRENT_USER; current_user -------------- omm (1 row)
LOCALTIMESTAMP	当前会话时间（无时区）。	gaussdb=# SELECT LOCALTIMESTAMP; timestamp ---------------------------- 2015-10-10 15:37:30.968538 (1 row)
NULL	空值。	-
SESSION_USER	当前系统用户。	gaussdb=# SELECT SESSION_USER; session_user -------------- omm (1 row)
SYSDATE	当前系统日期。	gaussdb=# SELECT SYSDATE; sysdate --------------------- 2015-10-10 15:48:53 (1 row)

版权所有 © 华为云计算技术有限公司

参数	描述	示例
USER	当前用户，此用户为CURRENT_USER的别名。	gaussdb=# SELECT USER; current_user --------------- omm (1 row)

1.6 函数和操作符

操作符可以对一个或多个操作数进行处理，其位置可能处于操作数之前、之后或者两个操作数中间。处理完成之后，返回处理结果。

函数是对一些业务逻辑的封装，以完成特定的功能。函数可以有参数，也可以没有参数。函数具有返回类型，执行结束后，会返回执行结果。

对于系统函数，用户可以进行修改，但是修改之后系统函数的语义可能会发生改变，从而导致系统控制紊乱。正常情况下不允许用户手动修改系统函数。

函数分为系统函数和语法函数。存在部分函数既是系统函数又是语法函数，具体请参见表1-29。

- 系统函数：调用注册机制，注册到pg_proc系统表中，通过查询pg_proc系统表能够获取系统函数元信息。
- 语法函数：由SQL语法规则直接定义，具有专属的语法格式。语法函数列表请参见表1-29。

表 1-29 语法函数列表

函数名称	ORA兼容模式	MYSQL兼容模式	其他兼容模式
adddate	不涉及	语法函数&系统函数	语法函数&系统函数
cast	语法函数	语法函数	语法函数
coalesce	语法函数	语法函数	语法函数
collation for	语法函数	语法函数	语法函数
convert	系统函数	语法函数&系统函数	语法函数&系统函数
current_catalog	语法函数	语法函数	语法函数
current_date	语法函数	语法函数	语法函数
current_role	语法函数	语法函数	语法函数
current_schema	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
current_time	语法函数	语法函数	语法函数

版权所有 © 华为云计算技术有限公司

函数名称	ORA兼容模式	MYSQL兼容模式	其他兼容模式
current_timestamp	语法函数&系统函数	语法函数	语法函数
current_user	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
curtime	不涉及	系统函数	不涉及
date_add	不涉及	语法函数&系统函数	语法函数&系统函数
date_sub	不涉及	语法函数&系统函数	语法函数&系统函数
db_jsobj	不涉及	语法函数	语法函数
dbtimezone	语法函数&系统函数	不涉及	不涉及
default	不涉及	语法函数	不涉及
extract	语法函数	语法函数	语法函数
get_format	不涉及	语法函数&系统函数	语法函数&系统函数
greatest	语法函数	语法函数	语法函数
ifnull	不涉及	语法函数	不涉及
last_day	系统函数	语法函数&系统函数	语法函数&系统函数
least	语法函数	语法函数	语法函数
lnnvl	语法函数	不涉及	不涉及
localtime	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
localtimestamp	语法函数	语法函数	语法函数
now	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
nullif	语法函数	语法函数	语法函数
nvl	语法函数	语法函数	语法函数
nvl2	语法函数	不涉及	不涉及
overlay	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
position	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数

版权所有 © 华为云计算技术有限公司

函数名称	ORA兼容模式	MYSQL兼容模式	其他兼容模式
regexp_like	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
rownum	语法函数	语法函数	语法函数
sessiontimezone	语法函数	不涉及	不涉及
session_user	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
subdate	不涉及	语法函数&系统函数	语法函数&系统函数
substr	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
substring	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
sysdate	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
timestampadd	不涉及	语法函数	语法函数
timestampdiff	不涉及	语法函数	不涉及
treat	语法函数	语法函数	语法函数
trim	语法函数	语法函数	语法函数
user	语法函数	语法函数	语法函数
utc_date	不涉及	系统函数	不涉及
utc_time	不涉及	系统函数	不涉及
utc_timestamp	不涉及	系统函数	不涉及
xmlcast	语法函数	不涉及	不涉及
xmlconcat	语法函数	语法函数	语法函数
xmlelement	语法函数	语法函数	语法函数
xmlexists	语法函数&系统函数	语法函数&系统函数	语法函数&系统函数
xmlforest	语法函数	语法函数	语法函数
xmlparse	语法函数	语法函数	语法函数
xmlpi	语法函数	语法函数	语法函数
xmlquery	语法函数	不涉及	不涉及
xmlroot	语法函数	语法函数	语法函数
xmlserialize	语法函数	语法函数	语法函数

版权所有 © 华为云计算技术有限公司

函数名称	ORA兼容模式	MYSQL兼容模式	其他兼容模式
year	不涉及	系统函数	不涉及

说明

当GUC参数behavior_compat_options含有'enable_funcname_with_argsname'选项时，投影别名显示完整函数。

1.6.1 逻辑操作符

常用的逻辑操作符有AND、OR和NOT，运算结果有三个值，分别为TRUE、FALSE和NULL，其中NULL代表未知。运算优先级顺序为：NOT>AND>OR。

运算规则请参见表1-30，表中的a和b代表逻辑表达式。

表 1-30 运算规则表

a	b	a AND b的结果	a OR b的结果	NOT a的结果
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
TRUE	NULL	NULL	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	NULL	FALSE	NULL	TRUE
NULL	NULL	NULL	NULL	NULL

说明

- 操作符AND和OR具有交换性，即交换左右两个操作数，不影响其结果。
- 不支持对XML类型数据进行操作。

1.6.2 比较操作符

大部分数据类型都可用比较操作符进行比较，并返回一个布尔类型的值。

比较操作符均为双目操作符，被比较的两个数据类型需要为相同的数据类型或者是可以进行隐式转换的类型。

GaussDB提供的比较操作符请参见表1-31。

表 1-31 比较操作符

操作符	描述
<	小于

版权所有 © 华为云计算技术有限公司

操作符	描述
>	大于
<=	小于或等于
>=	大于或等于
=	等于
<> 或 !=或^=	不等于

说明

- 当比较操作符（<=、<>、>=、^=）中间存在空格时，也可以识别成没有空格进行正常操作。!=中间存在空格时，!会被识别为阶乘，可能会导致结果与预期不一致。
- 比较操作符可以用于所有相关的数据类型。所有比较操作符都是双目操作符，返回布尔类型数值。不等号的计算优先级高于等号。当输入的数据类型不同且无法隐式转换时，比较操作将会失败。例如像1<2<3这样的表达式是非法的，因为布尔值和3之间无法用小于号（<）比较。
- 上述每种操作符在PG_PROC系统表中都有对应的函数，如果其对应的函数的proleakproof属性值为f，则表示该函数不是防数据泄露的。如果用户仅拥有视图权限而未获得该视图所对应表的权限，在查询该视图时，可能存在查询计划不是最优的问题。
- 不支持XML类型数据。
- 如果被比较的两个数据类型为不同的数据类型或无法进行隐式转换的类型，会先将两个数据类型先隐式转换为第三种可转换的数据类型，再进行比较，此时比较结果可能不具有实际意义。

  示例：

```
gaussdb=# EXPLAIN verbose SELECT to_datea('2010-10-10','yyyy-mm-dd') > 1.5;
                                  QUERY PLAN
-------------------------------------------------------------------------------
 Result  (cost=0.00..0.02 rows=1 width=0)
   Output: ((to_datea('2010-10-10'::text, 'yyyy-mm-dd'::text))::text > '1.5'::text)
(2 rows)
```

1.6.3 字符处理函数和操作符

GaussDB提供的字符处理函数和操作符主要用于字符串与字符串、字符串与非字符串之间的连接，以及字符串的模式匹配操作。字符串处理函数除了length相关函数，其他函数和操作符不支持大于1GB的clob作为参数。

说明

若进行字符比较时，存在输入参数为非法字符，统一默认返回false。

- bin(number)

  描述：返回给定数字的二进制值表示的字符串。

  参数：如表1-32所示。

版权所有 © 华为云计算技术有限公司

表 1-32 bin 参数说明

参数名称	类型	描述
number	• 整数类型：tinyint、smallint、mediumint、int、bigint。 • 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 • 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 • 浮点类型：float、real、double。 • 定点类型：numeric、decimal、dec。 • 布尔类型：boolean。	表示给定数字。

返回值类型：text

示例：

```
gaussdb=# SELECT bin(5);
  bin
--------
  101
(1 row)
```

说明

- 该函数仅在参数sql_compatibility='MYSQL'时生效。
- 如果入参number是小数，将其向下取整。
- 如果入参number的绝对值超过bigint unsigned类型的最大值，该参数将被转换为其上限值。

- bit_length(string)

  描述：返回输入字符串的位数。

  返回值类型：int

  示例：

```
gaussdb=# SELECT bit_length('world');
 bit_length
------------
```


版权所有 © 华为云计算技术有限公司

```
      40
(1 row)
```

- btrim(string text [, characters text])

  描述：从string开头和结尾删除仅包含characters中字符（缺省时为空白）的最长字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT btrim('sring' , 'ing');
 btrim
-------
 sr
(1 row)
```

- char_length(string)或character_length(string)

  描述：返回输入字符串中的字符个数。

  返回值类型：int

  示例：

```
gaussdb=# SELECT char_length('hello');
 char_length
-------------
           5
(1 row)
```

- dump(expr[, return_fmt [, start_position [, length]]])

  描述：返回输入表达式的数据类型代码、字节长度和内部表示形式。

  参数：return_fmt指定内部表现形式的进制，start_position指定从第几个字节开始，length表示读取的长度。

  返回值类型：text

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- elt(pos, str1, str2,...)

  描述：用于返回指定位置的字符串，该函数接受多个字符串参数，并根据位置参数pos返回对应的字符串。

  参数：如**表1-33**所示。

版权所有 © 华为云计算技术有限公司

表 1-33 elt 参数说明

参数名称	类型	描述
pos	• 整数类型：tinyint、smallint、mediumint、int、bigint。 • 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned。 • 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 • 浮点类型：float、real、double。 • 定点类型：numeric、decimal、dec。 • 布尔类型：boolean。	表示指定位置的参数。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述
str1, str2,...	• 整数类型：tinyint、smallint、mediumint、int、bigint。 • 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 • 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext。 • 浮点类型：float、real、double。 • 定点类型：numeric、decimal、dec。 • 布尔类型：boolean。 • 大对象类型：tinyblob、blob、mediumblob、longblob。 • 日期类型：datetime、timestamp、date、time。	表示字符串列表。

返回值类型：text

示例：

```
gaussdb=# SELECT elt(3, 'a', 'b', 'c');
 elt
-------
   c
(1 row)
```

说明

- 该函数仅在参数sql_compatibility='MYSQL'时生效。
- 如果入参pos小于1或者超过参数的数量，则返回NULL。

- field(str, str1, str2, str3,...)

  描述：field函数返回str在参数列表（str1,str2,str3,...）中的位置序号，位置计数从1开始递增。若未找到匹配项，则返回0；当str参数为NULL时，直接返回0。该函数根据入参的类型采用不同比较规则：

版权所有 © 华为云计算技术有限公司

- 当所有入参均为数字类型时，按照数值规则进行比较。
- 当所有入参均为为非数字类型时，按照字符串规则进行比较。
- 当所有入参存在数字和非数字混合时，按照double类型进行比较。

参数：如**表1-34**所示。

表 1-34 field 参数说明

参数名称	类型	描述
str, str1, str2, str3,...	• 整数类型：tinyint、smallint、mediumint、int、bigint、tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 • boolean类型：true、false。 • 浮点类型：float(p)、float、real、double，float(m, d)、double(m, d)、real(m, d)。 • 定点类型：numeric、decimal、dec。 • 文本类型：tinytext、text、mediumtext、longtext。 • 字符串类型：char、varchar。 • 大对象类型：tinyblob、blob、mediumblob、longblob。 • 日期类型：datetime、timestamp、date、time。	表示字符串列表。

返回值类型：bigint

示例：

```
gaussdb=# SELECT field( 'abc','1',1,'abc','abcd' );
 field
-------
     3
(1 row)
```

说明

- field函数仅在参数sql_compatibility='MYSQL'时生效。
- 自b_format_version='5.7'和b_format_dev_version='s1'版本起，sql_mode参数pad_char_to_full_length控制对char类型尾部填充空格，会影响field的比较结果，具体请参见**表1-8**。
- 自b_format_version='5.7'和b_format_dev_version='s1'版本起，字符类型、二进制类型、数值类型、日期时间类型行为兼容M，会影响field的比较结果，具体请参见**数据类型**。对于数值类型中的浮点类型，由于连接参数设置不同，精度可能与M有差异，不建议使用该场景，或使用numeric类型代替。
- 自b_format_version='5.7'和b_format_dev_version='s2'版本起，支持字符转义和常量字符串获取字符序，字符序会影响field的比较结果，具体请参见**SET**章节SET NAMES语法，字符类型不同字符序间的合并规则请参见**字符集和字符序合并规则**。

- insert(str1, pos, len, str2)

版权所有 © 华为云计算技术有限公司

描述：返回字符串str1的处理结果，子字符串起始于pos位置，将len个字符长的字符串替代为str2。任意传参为NULL，则返回值为NULL。

参数：如表1-35所示。

表 1-35 insert 参数说明

参数名称	类型	描述
pos、len	• 整数类型：tinyint、smallint、mediumint、int、bigint。 • 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 • 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串。 • 浮点类型：float、real、double。 • 定点类型：numeric、decimal、dec。	pos表示指定位置的参数；len表示替换的长度。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述
str1、str2	• NULL。 • 整型：tinyint、smallint、mediumint、int、bigint; tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned；但整型数值超过正负81个9，返回值与MySQL有差异。此差异为GaussDB与MySQL固有差异，非本函数导致。 • 浮点型、定点型：float、real、double；numeric、decimal、dec；但当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，此为固有显示差异。非本函数所致。 • 字符串类型：char、varchar。 • 文本类型：tinytext、text、mediumtext、longtext。 • 大对象类型：tinyblob、blob、mediumblob、longblob。 • 日期类型：datetime,、timestamp、date、time。	字符串。

返回值类型：函数返回值是text类型(s1和s2都是text类型)或bytea类型(s1或s2任意参数为bytea类型)。

示例：

```
gaussdb=# SELECT INSERT('abcdef',2,3,'gg');
 insert
--------
 aggef
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

- insert函数仅在参数sql_compatibility='MYSQL'时生效；
- Int64类型传参有范围限制，当超出-9223372036854775808~9223372036854775807范围会直接报错。MySQL对数值类型传参范围无限制，异常会告警，按照上限或下限数值处理；字符串传参有限制，入参text类型字符串长度最大为2^30-5字节，入参bytea类型字符串长度最大为2^30-512字节。

- instr(text, text, int, int)

  描述：该函数返回在第一个参数中，从第三个参数指定的起始位置开始，匹配到第二个参数第n次出现的位置（n为第四个参数的值）。其中第三个参数表示开始匹配的起始位置，第四个参数表示匹配的次数。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT instr( 'abcdabcdabcd', 'bcd', 2, 2 );
   instr
  -------
      6
  (1 row)
  ```

- instrb(text, text, int, int)

  描述：该函数返回在第一个参数中，从第三个参数指定的起始位置开始，匹配到第二个参数第n次出现的位置（n为第四个参数的值）。其中，第三个参数表示开始匹配的起始位置，第四个参数表示匹配的次数。与instr函数不同的是，instrb固定以字节为单位，不受所使用的字符集影响。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT instrb( 'abcdabcdabcd', 'bcd', 2, 2 );
   instrb
  -------
      6
  (1 row)
  ```

  说明

  - 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。
  - 参数int1、int2入参若为小数则不会被四舍五入，而是被截断。

- lengthb(text/bpchar)

  描述：获取指定字符串的字节数。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT lengthb('hello');
   lengthb
  ---------
        5
  (1 row)
  ```

- lengthc(text/bpchar)

  描述：返回指定字符串的长度，以Unicode字符编码规则计算。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT lengthc('hello');
   lengthc
  ---------
  ```


版权所有 © 华为云计算技术有限公司

```
     5
(1 row)
```

- left(str text, n int)

  描述：返回字符串的前n个字符。当n是负数时，返回除最后|n|个字符以外的所有字符。

  返回值类型：text

  示例：

```
gaussdb=# SELECT left('abcde', 2);
 left
------
 ab
(1 row)
```

- length(string bytea, encoding name)

  描述：指定encoding编码格式的string的字符数。在这个编码格式中，string需要为有效的。

  返回值类型：int

  示例：

```
gaussdb=# SELECT length('jose', 'UTF8');
 length
--------
      4
(1 row)
```

  说明

  如果是查询bytea类型的长度，指定utf8编码时，最大长度只能为536870888。

- locate(substr, str[, pos])

  描述：返回字符串substr在字符串str中第一次出现的位置，从位置pos（默认为1）开始算起。如果在str中找不到substr，则此函数返回0。

  参数：如表1-36所示。

表 1-36 locate 参数说明

参数名称	类型	描述
substr，str	• 整型：tinyint、smallint、mediumint、int、bigint、tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 • 浮点型：float、double。 • 任意精度类型：numeric。 • 字符类型：char、varchar、text。 • 二进制类型：bytea、blob。 • 日期/时间类型：date、time、datetime、timestamp。	• substr必需，表示要搜索的子字符串。 • str必需，表示将被搜索的字符串。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述
pos	• 整型：tinyint、smallint、mediumint、int、tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 • 浮点型：float、double。 • 任意精度类型：numeric。 • 字符类型：char、varchar、text。 • 布尔类型：true、false。	pos可选，表示搜索的起始位置。

返回值类型：int

示例：

```
gaussdb=# SELECT locate( 'b','abcabc');
 locate
--------
      2
(1 row)

gaussdb=# SELECT locate( 'b','abcabc',3);
 locate
--------
      5
(1 row)
```

说明

locate函数仅在参数sql_compatibility='MYSQL'时生效。

- lpad_s(string text, length int [, fill text])

  描述：通过填充字符fill（缺省时为空白），把string填充为length长度。如果string已经比length长则报错。

  返回值类型：text

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：

  - 参数length表示字符串显示长度，单个字符的显示长度按照ORA兼容处理。
  - 函数执行过程中出现length剩余长度为1且下一个字符显示长度为2，在字符串左侧添加一个空格字符。
  - 参数length入参若为小数则不会被四舍五入，而是会向下取整。
  - 参数string与fill不支持不符合编码规范的输入。

  在其他情况下：

  - 参数length表示字符串中字符总长度，单个字符的长度固定为1。
  - 参数length入参若为小数则被四舍五入。
  - 参数string与fill不支持不符合编码规范的输入。

- lpad(string text, length int [, fill text])

  描述：通过填充字符fill（缺省时为空白），把string填充为length长度。如果string已经比length长则将其尾部截断。

版权所有 © 华为云计算技术有限公司

返回值类型：text

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：

- 参数length表示字符串显示长度，单个字符的显示长度按照ORA兼容处理。
- 函数执行过程中出现length剩余长度为1且下一个字符显示长度为2，在字符串左侧添加一个空格字符。
- 参数length入参若为小数则不会被四舍五入，而是会向下取整。
- 参数string与fill不支持不符合编码规范的输入。

在其他情况下：

- 参数length表示字符串中字符总长度，单个字符的长度固定为1。
- 参数length入参若为小数则被四舍五入。
- 参数string与fill不支持不符合编码规范的输入。

- make_set(bits, str1, str2,...)

  描述：str1对应bits入参在比特形式下的位0，str2对应位1，以此类推。对应比特形式下位为1，则将对应的str参数追加到结果中，以逗号作为分隔符返回。

  参数：如**表1-37**所示。

版权所有 © 华为云计算技术有限公司

表 1-37 make_set 参数说明

参数名称	类型	描述
bits	bits入参支持： ● NULL。 ● 整型：tinyint、smallint、mediumint、int、bigint; tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned。 ● 浮点、定点型：float、real、double；numeric、decimal、dec。 ● 比特类型：bit。GaussDB未完善支持bit，代码设计中当入参1为十六进制格式（如“x'FFF'”）且长度为奇数时，MySQL会报错，但GaussDB处理方式不同。 ● 字符串类型：char、varchar；仅支持纯数字整数字符串（范围在bigint内）；带小数的字符串由于取整规则与MySQL存在差异，返回结果存在差异。数字字母拼接字符串不支持。 ● 文本类型：tinytext、text、mediumtext、longtext；仅支持纯数字整数字符串（范围在bigint内），超出范围返回结果与MySQL存在差异；带小数的字符串由于取整规则与MySQL存在差异，返回结果存在差异。数字字母拼接文本类型不支持。 ● 大对象类型：tinyblob、blob、mediumblob、longblob。	通过该参数所对应的二进制的每一位控制是否包含后续的str参数。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述
str1, str2,...	str可变入参支持： ● NULL。 ● 整型：tinyint、smallint、mediumint、int、bigint; tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned；但当整型数值超过正负81个9时，返回值与MySQL有差异。此差异为GaussDB与MySQL固有差异，非本函数导致。 ● 浮点型、定点型：float、real、double；numeric、decimal、dec；但当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，会以科学计数法打印，此为固有显示差异，非本函数所致。 ● 字符串类型：char、varchar。 ● 文本类型：tinytext、text、mediumtext、longtext。 ● 大对象类型：tinyblob、blob、mediumblob、longblob。 ● 日期类型：datetime、timestamp、date、time。	根据bits参数的二进制位决定是否将str1,str2,...中的某些字符串按顺序追加到返回结果中。

返回值类型：text

示例：

```
gaussdb=# SELECT make_set(7,'a','b','c','d');
 make_set
----------
 a,b,c
(1 row)
gaussdb=# SELECT
make_set(b'1010101010101101010101010101101010101010101101010101010101101010101010','1','2','3','4','
5','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24','25','26','27','28','29','30','31','3
2','33','34','35','36','37','38','39','40','41','42','43','44','45','46','47','48','49','50','51','52','53','54','55','56','57','
58','59','60','61','62','63','64','65','66','67','68','69');
                                                  make_set
-------------------------------------------------------------------------------------------------------------
 2,4,6,8,10,12,13,15,17,19,21,23,25,26,28,30,32,34,36,38,39,41,43,45,47,49,51,52,54,56,58,60,62,64
(1 row)
```

说明

- make_set函数仅在参数sql_compatibility='MYSQL'时生效。
- bits参数为整型时，最大范围支持到int128，低于MySQL范围。
- bits参数为日期类型datetime、timestamp、date、time，由于时间类型转整型与MySQL存在差异，非本函数所致，目前均不做支持。
- bit类型或boolean类型由于此类数据类型GaussDB与MySQL存在差异，返回结果导致的差异为GaussDB与MySQL固有差异,非本函数导致。bits入参为boolean类型，str入参为bit类型与boolean类型均不做支持。

版权所有 © 华为云计算技术有限公司

- notlike(bytea, bytea)、notlike(name, text)、notlike(text, text)

  描述：该函数为重载函数，其功能为比较两个输入参数是否为不同的数据类型。

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT notlike(1,2);
   notlike
--------------
          t
(1 row)
gaussdb=# SELECT notlike(1,1);
   notlike
--------------
          f
(1 row)
```

- octet_length(string)

  描述：返回输入字符串中的字节数。

  返回值类型：int

  示例：

```
gaussdb=# SELECT octet_length('jose');
 octet_length
--------------
          4
(1 row)
```

- overlay(string placing string FROM int [for int])

  描述：替换子字符串。FROM int表示从第一个string的第几个字符开始替换，for int表示替换第一个string的字符数目。

  返回值类型：text

  示例：

```
gaussdb=# SELECT overlay('hello' placing 'world' FROM 2 for 3 );
 overlay
---------
 hworldo
(1 row)
```

- position(substring in string)

  描述：指定子字符串的位置。字符串区分大小写。

  返回值类型：int，字符串不存在时返回0。

  示例：

```
gaussdb=# SELECT position('ing' in 'string');
 position
----------
        4
(1 row)
```

- pg_client_encoding()

  描述：当前客户端编码名称。

  返回值类型：name

  示例：

```
gaussdb=# SELECT pg_client_encoding();
 pg_client_encoding
--------------------
 UTF8
(1 row)
```

- quote(str)

版权所有 © 华为云计算技术有限公司

描述：在字符串str两端分别添加单引号后进行输出，同时可将str中的特殊字符进行转义，包括反斜线（\\）、单引号（\'）、ASCII NUL（\0）或Control+Z（\Z）。如果参数为NULL，则返回值为字符串NULL，而不添加单引号。

参数：如**表1-38**所示。

表 1-38 quote 参数说明

参数名称	类型	描述
str	• NULL。 • 整型：tinyint、smallint、mediumint、int、bigint; tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned；但整型数值超过正负81个9，返回值与MySQL有差异。此差异为GaussDB与MySQL固有差异，非本函数导致。 • 浮点型、定点型：float、real、double；numeric、decimal、dec；但当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，以科学计数法打印，此为固有显示差异。非本函数所致。 • 字符串类型：char、varchar。 • 文本类型：tinytext、text，mediumtext、longtext。 • 大对象类型：tinyblob、blob、mediumblob、longblob。 • 日期类型：datetime、timestamp、date、time。	在字符串str两端分别添加单引号后进行输出，同时可将str中的特殊字符进行转义，包括反斜线（\\）、单引号（\'）、ASCII NUL（\0）或Control+Z（\Z）。

版权所有 © 华为云计算技术有限公司

返回值类型：text

示例：

```
gaussdb=# SELECT quote('hello\' world');
    quote
--------------
 'hello\' world'
(1 row)
```

说明

- 函数仅在参数sql_compatibility='MYSQL'时生效。
- 开启GUC参数：
  ```
  SET standard_conforming_strings=off;
  SET escape_string_warning=off;
  SET backslash_quote=on;
  ```
- 已知str字符串中含有\Z、\r、\%、_，GaussDB未进行转义，与MySQL存在差异。此外，斜线后跟部分数字也会引起差异，如“\563”。这些差异源于GaussDB与MySQL在转义字符处理上的不同，与本函数无关。
- str字符串中的\b，输出结果表现形式与MySQL有差异。此为GaussDB与MySQL的固有差异，与本函数无关。
- str字符串中含有\0时，GaussDB由于UTF-8字符集不识别该字符，输入不成功。此为GaussDB与MySQL的固有差异，与本函数无关。
- str为bit或boolean类型时，由于GaussDB与MySQL此类型目前有差异，暂不支持此类类型。
- GaussDB最大支持1GB数据传输，str入参长度最大支持536870908，函数返回结果字符串最大支持1GB。

- quote_ident(string text)

  描述：返回适用于SQL语句的标识符形式（使用适当的引号进行界定）。该函数仅在必要时为字符串添加引号（字符串包含非标识符字符或者会转换大小写的字符）。返回值中嵌入的引号需要写两次。

  参数：入参为一个字符串，将被当作SQL标识符使用。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT quote_ident('hello world');
   quote_ident
  --------------
   "hello world"
  (1 row)
  ```

- quote_literal(string text)

  描述：返回适用于在SQL语句里当作文本使用的形式（使用适当的引号进行界定）。支持显式转换成字符类型后的XML类型数据。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT quote_literal('hello');
   quote_literal
  ---------------
   'hello'
  (1 row)
  ```

  如果出现如下写法，text文本将进行转义。

  ```
  gaussdb=# SELECT quote_literal(E'O\'hello');
   quote_literal
  ---------------
   'O''hello'
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

如果出现如下写法，反斜杠会写入两次。

```
gaussdb=# SELECT quote_literal('O\hello');
 quote_literal
---------------
 E'O\\hello'
(1 row)
```

如果参数为NULL，返回空。如果参数可能为null，通常使用函数quote_nullable更合适。

```
gaussdb=# SELECT quote_literal(NULL);
 quote_literal
---------------

(1 row)
```

- quote_literal(value anyelement)

  描述：将给定的值强制转换为text，加上引号作为文本。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT quote_literal(42.5);
   quote_literal
  ---------------
   '42.5'
  (1 row)
  ```

  如果出现如下写法，定值将进行转义。

  ```
  gaussdb=# SELECT quote_literal(E'O\'42.5');
   quote_literal
  ---------------
   '0''42.5'
  (1 row)
  ```

  如果出现如下写法，反斜杠会写入两次。

  ```
  gaussdb=# SELECT quote_literal('O\42.5');
   quote_literal
  ---------------
   E'O\\42.5'
  (1 row)
  ```

- quote_nullable(string text)

  描述：返回适用于在SQL语句里当作字符串使用的形式（使用适当的引号进行界定）。支持显式转换成字符类型后的XML类型数据。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT quote_nullable('hello');
   quote_nullable
  ----------------
   'hello'
  (1 row)
  ```

  如果出现如下写法，text文本将进行转义。

  ```
  gaussdb=# SELECT quote_nullable(E'O\'hello');
   quote_nullable
  ----------------
   'O''hello'
  (1 row)
  ```

  如果出现如下写法，反斜杠会写入两次。

  ```
  gaussdb=# SELECT quote_nullable('O\hello');
   quote_nullable
  ----------------
   E'O\\hello'
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

如果参数为NULL，返回NULL。

```
gaussdb=# SELECT quote_nullable(NULL);
 quote_nullable
----------------
 NULL
(1 row)
```

- quote_nullable(value anyelement)

  描述：将给定的参数值转换为text，加上引号作为文本。入参类型为anyelement。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT quote_nullable(42.5);
   quote_nullable
  ----------------
   '42.5'
  (1 row)
  ```

  如果出现如下写法，定值将进行转义。

  ```
  gaussdb=# SELECT quote_nullable(E'O\'42.5');
   quote_nullable
  ----------------
   'O''42.5'
  (1 row)
  ```

  如果出现如下写法，反斜杠会写入两次。

  ```
  gaussdb=# SELECT quote_nullable('O\42.5');
   quote_nullable
  ----------------
   E'O\\42.5'
  (1 row)
  ```

  如果参数为NULL，返回NULL。

  ```
  gaussdb=# SELECT quote_nullable(NULL);
   quote_nullable
  ----------------
   NULL
  (1 row)
  ```

- space(count)

  描述：返回由指定数量的空格组成的字符串。

  参数：如**表1-39**所示。

版权所有 © 华为云计算技术有限公司

表 1-39 space 参数说明

参数名称	类型	描述
count	• 整数类型：tinyint、smallint、mediumint、int、bigint。 • 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned。 • 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext（仅支持纯数字整数字符串，且整数范围在bigint范围内）。 • 浮点类型：float、real、double。 • 定点类型：numeric、decimal、dec。 • 布尔类型：boolean。	表示空格数量。

返回值类型：text

示例：

```
gaussdb=# SELECT space(5);
 space
--------

(1 row)
```

说明

- space函数仅在参数sql_compatibility='MYSQL'时生效。
- 如果入参count小于1073741819且大于0，返回count个空格字符串，否则返回空字符串。
- 当入参类型为浮点类型或定点类型时，若参数值为非整数，此函数的行为是向下取整。
- 当入参类型为布尔类型时，若参数值为true，则将其转换为1；若参数值为false，则将其转换为0。

- substring_inner(string [from int] [for int])

  描述：截取子字符串，from int表示从第几个字符开始截取，for int表示截取几个字节。

  返回值类型：text

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT substring_inner('adcde', 2,3);
 substring_inner
-----------------
 dcd
(1 row)
```

- substring(string [from int] [for int])

  描述：截取子字符串，from int表示从第几个字符开始截取，for int表示截取几个字节。

  返回值类型：text

  示例：

```
gaussdb=# SELECT substring('Thomas' from 2 for 3);
 substring
-----------
 hom
(1 row)
```

  说明

  此函数在MYSQL模式数据库中，且GUC参数 b_format_version = '5.7'和 b_format_dev_version = 's1'时，[from int]可以为负数，此时表示从字符串末尾向前数的第几个字符。如果不设置参数，[from int]为负数时，结果为空。

- substring(string, pos, len)

  描述：截取子字符串。

  参数：如**表1-40**所示。

  表 1-40 substring(string, pos, len)参数说明

参数名称	类型	描述	取值范围
string	text	需要被截取的字符串。	-
pos	int	需要被截取字符串的起点。	绝对值小于字符串长度。
len	int	需要被截取字符串的长度。	大于0。

  返回值类型：text

  示例：

```
gaussdb=# SELECT substring('substrteststring', -5, 5);
 substring
-----------
 tring
(1 row)
```

  说明

  此函数在MYSQL模式数据库中，且GUC参数b_format_version = '5.7'和 b_format_dev_version = 's1'时，pos可以为负数，此时表示从字符串字符串末尾向前数的第几个字符。如果不设置参数，pos为负数时，结果为空。

- substring_index(str, split, index)

  描述：此函数返回第index个被split分隔的str字符串的左侧或右侧的内容，若index为正数，则返回左侧内容；若为负数，则返回右侧内容。

  参数：如**表1-41**所示。

版权所有 © 华为云计算技术有限公司

表 1-41 substring_index 参数说明

参数名称	类型	描述
str	text	待处理的源字符串。
split	text	分隔符字符串。
index	int	分隔符字符串的位置索引（从1开始），正数表示左侧，负数表示右侧。

返回值类型：text

示例：

```
gaussdb=# SELECT substring_index('Test1splitTest2splitTest3splitTest4', 'split', 2);
 substring_index
-----------------
 Test1splitTest2
(1 row)

gaussdb=# SELECT substring_index('Test1splitTest2splitTest3splitTest4', 'split', -2);
 substring_index
-----------------
 Test3splitTest4
(1 row)
```

- substring(string FROM pattern)

  描述：截取匹配POSIX正则表达式的子字符串。若未找到匹配项，则返回空值；否则返回文本中匹配模式的部分内容。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substring('Thomas' FROM '...$');
   substring
  -----------
   mas
  (1 row)
  gaussdb=# SELECT substring('foobar' FROM 'o(.)b');
   result
  --------
   o
  (1 row)
  gaussdb=# SELECT substring('foobar' FROM '(o(.)b)');
   result
  --------
   oob
  (1 row)
  ```

  说明

  - 如果POSIX正则表达式模式包含任何圆括号，那么将返回匹配第一对子表达式（对应第一个左圆括号）中的文本。若需使用圆括号而不触发此行为，建议在整个表达式外层添加一对圆括号。
  - 该函数在sql_compatibility = 'MYSQL'，且GUC参数b_format_version = '5.7'、b_format_dev_version = 's1'时，其使用方式与**substr(str FROM pos)**一致，输出结果和行为也和**substr(str FROM pos)**函数相同。

- substring(string from pattern for escape)

  描述：截取匹配SQL正则表达式的子字符串。声明的模式必须匹配整个数据串，否则函数失败并返回空值。为了标识在成功时应返回的模式部分，模式需要包含

版权所有 © 华为云计算技术有限公司

两个逃逸字符引用的部分，并且逃逸字符后需添加双引号（"）。匹配这两个标记之间的模式的文本将被返回。

返回值类型：text

示例：

```
gaussdb=# SELECT substring('Thomas' from '%#"o_a#"_' for '#');
 substring
-----------
 oma
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且GUC参数b_format_version = 5.7'、b_format_dev_version = 's1'时，其使用方式与**substr(str FROM pos FOR len)**函数一致，输出结果和行为也和**substr(str FROM pos FOR len)**函数相同。

- rawcat(raw, raw)

  描述：字符串拼接函数。

  返回值类型：raw

  示例：

  ```
  gaussdb=# SELECT rawcat('ab','cd');
   rawcat
  --------
   ABCD
  (1 row)
  ```

- regexp_substr(string text, pattern text [, position int [, occurrence int [, flags text]]])

  描述：正则表达式的抽取子串函数。与substr功能相似，正则表达式出现多个并列的括号时，也全部处理。

  参数：

  - string：用于匹配的源字符串。
  - pattern：用于匹配的正则表达式模式串。
  - position：可选参数，表示从源字符串的第几个字符开始匹配，默认值为1。
  - occurrence：可选参数，表示抽取第几个满足匹配的子串，默认值为1。
  - flags：可选参数，包含零个或多个改变函数匹配行为的单字母标记。flags支持的选项值及含义描述如**表1-42**所示。

  表 1-42 flags 支持的选项值

选项	描述
'b'	按照无扩展的BRE规则匹配。
'c'	大小写敏感匹配。
'e'	按照扩展的ERE规则匹配。
'i'	大小写不敏感匹配。
'm'	多行模式匹配。flags中包含'm' 时，按照多行模式匹配，否则按照单行模式匹配。

版权所有 © 华为云计算技术有限公司

选项	描述
'n'	n选项的含义和GUC参数behavior_compat_options及数据库当前的兼容模式有关： 数据库SQL语法兼容模式为ORA或MYSQL，且GUC参数behavior_compat_options值包含aformat_regexp_match时，n选项表示“.”能够匹配换行符('\n')；flags未指定'n'选项时，“.”不会匹配换行符。 其他情况下，'n'选项和'm'选项的含义相同。
'p'	部分新行敏感的匹配，影响'.'和方括号表达式，和新行敏感的匹配('m'或'n')一样，但是不影响^和$。
'q'	普通字符匹配。
's'	单行模式匹配，含义与'm'、'n'相反。
't'	紧凑模式匹配，空白符匹配自身。
'w'	逆部分新行匹配，与'p'含义相反。
'x'	宽松模式匹配，忽略空白符。

返回值类型：text

示例：

```
gaussdb=# SELECT regexp_substr('str','[ac]');
 regexp_substr
---------------

(1 row)

gaussdb=# SELECT regexp_substr('foobarbaz', 'b(..)', 3, 2) AS RESULT;
 result
--------
 baz
(1 row)
```

- regexp_count(string text, pattern text [, position int [, flags text]])

  描述：获取满足匹配的子串个数。

  参数：

  - string：用于匹配的源字符串。
  - pattern：用于匹配的正则表达式模式串。
  - position：表示从源字符串的第几个字符开始匹配，为可选参数，默认值为1。
  - flags：可选参数，包含零个或多个改变函数匹配行为的单字母标记。flags支持的选项值及含义描述如**表1-42**所示。

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下，以'\'结尾的pattern参数为合法的。

  返回值类型：int

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT regexp_count('foobarbaz','b(..)', 5) AS RESULT;
result
--------
1
(1 row)
```

- regexp_instr(string text, pattern text [, position int [, occurrence int [, return_opt int [, flags text]]]])

  描述：获取满足匹配条件的子串位置（从1开始）。如果没有匹配的子串，则返回0。

  参数：

  - string：用于匹配的源字符串。
  - pattern：用于匹配的正则表达式模式串。
  - position：可选参数，表示从源字符串的第几个字符开始匹配，默认值为1。
  - occurrence：可选参数，表示获取第occurrence个匹配子串的位置，默认值为1。
  - return_opt：可选参数，用于控制返回匹配子串的首字符位置还是尾字符位置。取值为0时，返回匹配子串的第一个字符的位置（从1开始计算），取值为大于0的值时，返回匹配子串的尾字符的下一个字符的位置。默认值为0。
  - flags：可选参数，包含零个或多个改变函数匹配行为的单字母标记。flags支持的选项值及含义描述如**表1-42**所示。

  返回值类型：int

  示例：

```
gaussdb=# SELECT regexp_instr('foobarbaz','b(..)', 1, 1, 0) AS RESULT;
result
--------
4
(1 row)

gaussdb=# SELECT regexp_instr('foobarbaz','b(..)', 1, 2, 0) AS RESULT;
result
--------
7
(1 row)
```

- regexp_matches(string text, pattern text [, flags text])

  描述：返回string中所有匹配POSIX正则表达式的子字符串。如果pattern不匹配，该函数不返回行。

  - 如果模式不包含圆括号子表达式，则每一个被返回的行都是一个单一元素的文本数组，其中包括匹配整个模式的子串。
  - 如果模式包含圆括号子表达式，该函数返回一个文本数组，它的第n个元素是匹配模式的第n个圆括号子表达式的子串。

  参数：flags参数为可选参数，包含零个或多个改变函数行为的单字母标记。i表示进行大小写无关的匹配，g表示替换每一个匹配的子字符串而不仅仅是第一个。

版权所有 © 华为云计算技术有限公司

须知

如果提供了最后一个参数，但参数值是空字符串（''），且数据库SQL兼容模式设置为ORA的情况下，返回结果将为空集。这是因为ORA兼容模式将''作为NULL处理，避免此类行为的方式有如下几种：

- 将数据库SQL兼容模式改为TD；
- 不提供最后一个参数，或最后一个参数不为空字符串。

返回值类型：setof text[]

示例：

```
gaussdb=# SELECT regexp_matches('foobarbequebaz', '(bar)(beque)');
 regexp_matches
----------------
 {bar,beque}
(1 row)
gaussdb=# SELECT regexp_matches('foobarbequebaz', 'barbeque');
 regexp_matches
----------------
 {barbeque}
(1 row)
gaussdb=# SELECT regexp_matches('foobarbequebazilbarfbonk', '(b[^b]+)(b[^b]+)', 'g');
 regexp_matches
----------------
 {bar,beque}
 {bazil,barf}
(2 rows)
```

- regexp_match(string text, pattern text [, flags text])

  描述：返回一个字符串数组，为POSIX正则表达式与字符串首次匹配所得到的子串。如果pattern不匹配，该函数返回空行。

  - 如果模式不包含圆括号子表达式，则每一个被返回的行都是一个单一元素的文本数组，其中包括匹配整个模式的子串。
  - 如果模式包含圆括号子表达式，该函数返回一个文本数组，它的第n个元素是匹配模式的第n个圆括号子表达式的子串。

  参数：flags参数为可选参数，不带flags默认使用紧语法匹配。flags参数包含零个或多个改变函数行为的单字母标记，支持的标记如下表所示。

表 1-43 flags 参数说明

选项	描述
'b'	POSIX正则表达式变为BRE（基础正则表达式）。
'c'	大小写敏感的匹配。
'e'	POSIX正则表达式变为ERE（扩展正则表达式）。
'i'	大小写不敏感的匹配，即在匹配时不区分大小写。
'n'	新行敏感的匹配，将字符串视为多行，^和$匹配每行的开头和结尾，（.）不可以匹配换行。
'm'	新行敏感的匹配，将字符串视为多行，^和$匹配每行的开头和结尾，（.）不可以匹配换行，n的匹配模式一致。

版权所有 © 华为云计算技术有限公司

选项	描述
'p'	部分新行敏感的匹配，单行的开头和结尾，（.）不能匹配换行符，^只能匹配第一行开头，$只能匹配最后一行末尾。
'q'	将正则表达式视为一个文本字符串，所有都是普通字符。
's'	非新行敏感匹配，即将字符串视为单行，^和$匹配单行的开头和结尾，此模式下，正则表达式中的点号（.）可以匹配任何字符，包括换行符。
't'	紧语法匹配。
'w'	逆部分新行敏感的匹配，正则表达式中^匹配每行的开头，$只能匹配最后一行的结尾，（.）可以匹配任何字符，包括换行符。
'x'	扩展语法匹配，忽略RE中的空白字符和注释。以下三种场景除外： ● 前置了\的空白字符或#将被保留。 ● 方括号中的空白字符或#将被保留。 ● 在多字符符号中不能出现空白和注释，例如“(?:”。

须知

如果参数中存在空字符串（''），且数据库兼容模式设置为ORA的情况下，会导致返回结果为NULL，原因是ORA兼容模式下空字符串将''作为NULL处理。

返回值类型：setof text[]

示例：

```
gaussdb=# SELECT regexp_match('foobarbequebaz', '(bar)(beque)');
 regexp_match
--------------
 {bar,beque}
(1 row)
gaussdb=# SELECT (regexp_match('foobarbequebaz', 'bar.*que'))[1];
 regexp_match
--------------
 barbeque
(1 row)
gaussdb=# SELECT regexp_match('Learning #PostgreSQL', 'R', 'c');
 regexp_match
--------------


(1 row)
gaussdb=# SELECT regexp_match('hello world', 'h e l l o', 'x');
 regexp_match
--------------
 {hello}
(1 row)
```

- regexp_split_to_table(string text, pattern text [, flags text])

  描述：用POSIX正则表达式作为分隔符，分隔string。如果没有与pattern的匹配，该函数返回string。当存在至少一个匹配时，每个匹配项返回从上一个匹配的结束位置（或字符串的起始位置）到当前匹配的起始位置之间的文本。当没有更多匹配时，将返回从上一次匹配的结束位置到字符串末尾之间的文本。

版权所有 © 华为云计算技术有限公司

参数：flags参数包含零个或多个改变函数行为的单字母标记。i表示进行大小写无关的匹配，默认替换每一个匹配的子字符串，而不仅仅是替换第一个子字符串。

返回值类型：setof text

示例：

```
gaussdb=# SELECT regexp_split_to_table('hello world', E'\\s+');
 regexp_split_to_table
-----------------------
 hello
 world
(2 rows)
```

- regexp_split_to_array(string text, pattern text [, flags text])

  描述：用POSIX正则表达式作为分隔符，分隔string。和regexp_split_to_table函数相似，但本函数会将分隔结果以一个text数组的形式返回。

  返回值类型：text[]

  示例：

```
gaussdb=# SELECT regexp_split_to_array('hello world', E'\\s+');
 regexp_split_to_array
-----------------------
 {hello,world}
(1 row)
```

- repeat(string text, number int)

  描述：将string重复number次。

  返回值类型：text

  示例：

```
gaussdb=# SELECT repeat('Pg', 4);
  repeat
----------
 PgPgPgPg
(1 row)
```

  说明

  由于数据库内存分配机制限制单次内存分配不可超过1GB，因此number最大值不应超过(1G-x)/lengthb(string) - 1。x为头信息长度，通常大于4字节，其具体值在不同的场景下存在差异。

- replace(string text, from text, to text)

  描述：将字符串string中出现的所有子字符串from的内容替换成子字符串to的内容。

  返回值类型：text

  示例：

```
gaussdb=# SELECT replace('abcdefabcdef', 'cd', 'XXX');
    replace
----------------
 abXXXefabXXXef
(1 row)
```

- replace(string, substring)

  描述：删除字符串string中出现的所有子字符串substring的内容。

  参数：string类型为text；substring类型为text。

  返回值类型：text

  示例：

```
gaussdb=# SELECT replace('abcdefabcdef', 'cd');
    replace
```

 版权所有 © 华为云计算技术有限公司

```
----------------
 abefabef
(1 row)
```

- reverse(str)

  描述：返回颠倒的字符串（按字符颠倒）。

  返回值类型：text

  示例：

```
gaussdb=# SELECT reverse('abcde');
 reverse
---------
 edcba
(1 row)
```

- right(str text, n int)

  描述：返回字符串中的后n个字符。当n是负值时，返回除前|n|个字符以外的所有字符。

  返回值类型：text

  示例：

```
gaussdb=# SELECT right('abcde', 2);
 right
-------
 de
(1 row)

gaussdb=# SELECT right('abcde', -2);
 right
-------
 cde
(1 row)
```

- rpad(string text, length int [, fill text])

  描述：使用填充字符fill（缺省时为空白），将string填充至length长度。当string长度超过length长度时，则将其从尾部截断。

  返回值类型：text

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：

  - 参数length表示字符串显示长度，单个字符的显示长度按照与Oracle数据库一致。
  - 函数执行过程中出现length剩余长度为1且下一个字符显示长度为2，在字符串右侧添加一个空格字符。
  - 参数length入参若为小数则不会被四舍五入，而是被截断。
  - 参数string与fill不支持不符合编码规范的输入。
  - 参数string有长度限制，若填充后大小超出1G会报错。

  在其他情况下：

  - 参数length表示字符串中字符总长度，单个字符的长度固定为1。
  - 参数string有长度限制，若填充后大小超出1G会报错。
  - 参数length入参若为小数则被四舍五入。
  - 参数string与fill不支持不符合编码规范的输入。

- substrb(text, int, int)

  描述：提取子字符串。第一个int参数表示提取的起始位置，可以是正数或负数：正数表示从字符串开头开始计算，负数表示从字符串末尾开始计算；第二个int参数表示要提取的字符个数。

版权所有 © 华为云计算技术有限公司

返回值类型：text

示例：

```
gaussdb=# SELECT substrb('string',2,3);
 substrb
---------
 tri
(1 row)
```

- substrb(text, int)

  描述：提取子字符串，int表示提取的起始位置。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substrb('string',2);
   substrb
  ---------
   tring
  (1 row)
  ```

- substr(text, int)/substr(str FROM pos)

  描述：其中str为目标字符串，pos为字符串中的位置。输出str从pos位置开始到字符结束的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substr('stringtest' FROM 4);
   substr
  ---------
   ingtest
  (1 row)

  gaussdb=# SELECT substr('stringtest', 4);
   substr
  ---------
   ingtest
  (1 row)
  ```

- substr(str FROM pos for len)

  描述：提取子字符串，第一个int表示提取的起始位置，第二个表示提取几位字符。pos可以是负数，如果为负数，则从后往前提取。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substr('teststring' FROM 5 for 2);
   substr
  --------
   st
  (1 row)
  ```

- substr(bytea, from, count)

  描述：从参数bytea中抽取子字符串。from表示抽取的起始位置，count表示抽取的子字符串长度。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substr('string',2,3);
   substr
  --------
   tri
  (1 row)
  ```

- string || string

版权所有 © 华为云计算技术有限公司

描述：连接字符串。

返回值类型：text

示例：

```
gaussdb=# SELECT 'MPP'||'DB' AS RESULT;
 result
--------
 MPPDB
(1 row)
```

- string || non-string或non-string || string

  描述：连接字符串和非字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT 'Value: '||42 AS RESULT;
    result
  -----------
   Value: 42
  (1 row)
  ```

- split_part(string text, delimiter text, field int)

  描述：根据delimiter分隔string返回生成的第field个子字符串（以出现第一个delimiter的text为基础）。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT split_part('abc~@~def~@~ghi', '~@~', 2);
   split_part
  ------------
   def
  (1 row)
  ```

- strpos(string, substring)

  描述：指定的子字符串的位置。和position(substring in string)相同，不过参数顺序相反。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT strpos('source', 'rc');
   strpos
  --------
       4
  (1 row)
  ```

- to_hex(number int or bigint)

  描述：把number转换成十六进制表现形式。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT to_hex(2147483647);
    to_hex
  ----------
   7fffffff
  (1 row)
  ```

- translate(string text, from text, to text)

  描述：将string中包含的任何匹配from中的字符替换为to中对应位置的字符。当from的长度超过to时，string中匹配to长度之外的from位置（即from中超出to长度的字符）所对应的字符将被删除。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT translate('12345', '143', 'ax');
 translate
-----------
 a2x5
(1 row)
```

- length(string)

  描述：获取参数string中字符的数目。当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'、b_format_dev_version = 's1'后，string为字符串类型或文本类型时，返回string的字节数。

  返回值类型：integer

- lengthb(string)

  描述：获取参数string中字节的数目。与字符集有关，同样的中文字符，在GBK与UTF8中，返回的字节数不同。

  返回值类型：integer

  示例：

  ```
  gaussdb=# SELECT lengthb('ABCDEF');
   lengthb
  ---------
         7
  (1 row)
  ```

- substr(string, from)

  描述：从参数string中提取子字符串，from表示提取的起始位置：

  – from为0时，按1处理。
  – from为正数时，提取从from到末尾的所有字符。
  – from为负数时，提取字符串的后n个字符，n为from的绝对值。

  返回值类型：text

  示例：

  – from为正数时：

    ```
    gaussdb=# SELECT substr('ABCDEF',2);
     substr
    --------
     BCDEF
    (1 row)
    ```

  – from为负数时：

    ```
    gaussdb=# SELECT substr('ABCDEF',-2);
     substr
    --------
     EF
    (1 row)
    ```

- substr(string, from, count)

  描述：从参数string中提取子字符串，count表示提取的子字符串长度，from表示提取的起始位置。

  – from为0时，按1处理。
  – from为正数时，提取从from开始的count个字符。
  – from为负数时，提取从倒数第n个开始的count个字符，n为from的绝对值。
  – count小于1时，返回null。

  返回值类型：text

  示例：

版权所有 © 华为云计算技术有限公司

  - from为正数时：
    ```
    gaussdb=# SELECT substr('ABCDEF',2,2);
     substr
    --------
     BC
    (1 row)
    ```
  - from为负数时：
    ```
    gaussdb=# SELECT substr('ABCDEF',-3,2);
     substr
    --------
     DE
    (1 row)
    ```

- substrb(string, from)

  描述：该函数和substr(string, from)函数功能一致，但是计算单位为字节。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substrb('ABCDEF',-2);
   substrb
  ---------
   EF
  (1 row)
  ```

- substrb(string, from, count)

  描述：该函数和substr(string, from, count)函数功能一致，但是计算单位为字节。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT substrb('ABCDEF',2,2);
   substrb
  ---------
   BC
  (1 row)
  ```

- to_single_byte(char)

  描述：将字符串中所有多字节字符转换为单字节字符。

  返回值类型：text

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- to_multi_byte(char)

  描述：将字符串中所有单字节字符转换为多字节字符。

  返回值类型：text

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- trim([leading |trailing |both] [characters] FROM string)

  描述：从字符串string的开头、结尾或两边删除只包含characters中字符（缺省时为空白）的最长的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT trim(BOTH 'x' FROM 'xTomxx');
   btrim
  ```


版权所有 © 华为云计算技术有限公司

```
-------
 Tom
(1 row)
gaussdb=# SELECT trim(LEADING 'x' FROM 'xTomxx');
 ltrim
-------
 Tomxx
(1 row)
gaussdb=# SELECT trim(TRAILING 'x' FROM 'xTomxx');
 rtrim
-------
 xTom
(1 row)
```

- rtrim(string [, characters])

  描述：从字符串string的结尾删除仅包含characters中字符（缺省时为空白）的最长的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT rtrim('TRIMxxxx','x');
   rtrim
  -------
   TRIM
  (1 row)
  ```

- ltrim(string [, characters])

  描述：从字符串string的开头删除仅包含characters中字符（缺省时为空白）的最长的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT ltrim('xxxxTRIM','x');
   ltrim
  -------
   TRIM
  (1 row)
  ```

- upper(string)

  描述：把字符串转换为大写。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT upper('tom');
   upper
  -------
   TOM
  (1 row)
  ```

- lower(string)

  描述：把字符串转换为小写。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT lower('TOM');
   lower
  -------
   tom
  (1 row)
  ```

- nls_upper(string [, nlsparam])

版权所有 © 华为云计算技术有限公司

描述：把字符串转换为大写，可以指定排序规则来处理某些国家语言的特殊大写转换规则。nlsparam的格式为'nls_sort=sort_name'，其中sort_name替换为具体的排序规则名。当不输入nlsparam参数时，该函数完全等同于upper。

返回值类型：text

示例：

```
gaussdb=# SELECT nls_upper('große');
 nls_upper
-----------
 GROßE
(1 row)
gaussdb=# SELECT nls_upper('große', 'nls_sort = XGerman');
 nls_upper
-----------
 GROSSE
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下支持使用。

- nls_lower(string [, nlsparam])

  描述：把字符串转换为小写，可以指定排序规则来处理某些国家语言的特殊小写转换规则。nlsparam的格式为'nls_sort=sort_name'，其中sort_name替换为具体的排序规则名。当不输入nlsparam参数时，该函数完全等同于lower。

  返回值类型：text

  示例：

```
gaussdb=# SELECT nls_lower('INDIVISIBILITY');
   nls_lower
----------------
 indivisibility
(1 row)
gaussdb=# SELECT nls_lower('INDIVISIBILITY', 'nls_sort = XTurkish');
   nls_lower
----------------
 ındıvısıbılıty
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下支持使用。

- instr(string, substring[,position, occurrence])

  描述：从字符串string的position（缺省时为1）所指的位置开始查找，返回第occurrence（缺省时为1）次出现子串substring的位置。

  - 当position为0时，返回0。
  - 当position为负数时，从字符串倒数第n个字符往前逆向搜索。n为position的绝对值。

  本函数以字符为计算单位，如一个汉字为一个字符。

  返回值类型：integer

  示例：

```
gaussdb=# SELECT instr('corporate floor','or', 3);
 instr
-------
     5
(1 row)
gaussdb=# SELECT instr('corporate floor','or',-3,2);
 instr
```


版权所有 © 华为云计算技术有限公司

```
-------
     2
(1 row)
```

- initcap(string)

  描述：将字符串中的每个单词的首字母转换为大写，其他字母转换为小写。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT initcap('hi THOMAS');
    initcap
  -----------
   Hi Thomas
  (1 row)
  ```

  说明

  此函数在开启参数a_format_version值为10c和a_format_dev_version值为s2的情况下，如中文等无大小写区分的字符后接区分大小写的字符时，首字母会转换为大写，其他字母转换为小写。因此建议开启参数a_format_version值为10c和a_format_dev_version值为s2。

- ascii(string)

  描述：参数string的第一个字符的ASCII码。

  返回值类型：integer

  示例：

  ```
  gaussdb=# SELECT ascii('xyz');
   ascii
  -------
    120
  (1 row)
  ```

- ascii2(string)

  描述：参数string的第一个字符在数据库字符集中的十进制编码。

  返回值类型：bigint

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- asciistr(string)

  描述：把字符串string中非ASCII字符转换为\XXXX形式，其中XXXX表示UTF-16代码单元。

  返回值类型：text

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- unistr(string)

  描述：将字符串中的编码序列转换成对应字符，其他字符保持不变。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

说明

- 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。
- '\'后必须接4位16进制字符表示编码序列，或者接另一个'\'表示输入单个'\'字符。
- 入参是时间类型时，时间类型会隐式转换成字符串类型。

- vsize(expr)

  描述：返回输入表达式的字节数。

  返回值类型：int

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- replace(string varchar, search_string varchar, replacement_string varchar)

  描述：将字符串string中所有子字符串search_string替换成子字符串replacement_string。

  返回值类型：text

  示例：

```
gaussdb=# SELECT replace('jack and jue','j','bl');
    replace
----------------
 black and blue
(1 row)
```

- concat(str1, str2)

  描述：将字符串str1和str2连接并返回。该函数会调用data type的输出函数，返回值不确定，导致优化器在生成计划时不能提前计算结果。如果对性能有要求，建议用 || 替代。

  说明

  - 在sql_compatibility = 'MYSQL'的情况下，参数str1或str2为NULL会导致返回结果为NULL。
  - 该函数的返回值类型为变长类型，和表中数据比较时，会因为拼接结果丢失字符串长度，导致比较结果不相等。

  返回值类型：text

  示例：

```
gaussdb=# SELECT concat('Hello', ' World!');
    concat
--------------
 Hello World!
(1 row)
gaussdb=# SELECT concat('Hello', NULL);
 concat
--------

(1 row)
gaussdb=# CREATE TABLE test_space(c char(10));
CREATE TABLE
gaussdb=# INSERT INTO test_space values('a');
INSERT 0 1
-- 填充空格后仍然是定长字符串，预期可以查找到结果
gaussdb=# SELECT * FROM test_space WHERE c = 'a ';
     c
------------
 a
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
-- 拼接结果为变长字符串，比对失败，找不到结果
gaussdb=# SELECT * FROM test_space WHERE c = 'a' || ' ';
 c
---
(0 rows)
```

- chr(integer)

  描述：如果为UTF-8字符集，将输入作为unicode编码，返回一个UTF-8的字符，其他字符集给出ASCII码的字符。

  返回值类型：text

  说明

  - 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：参数integer入参若为小数则不会被四舍五入，而是被截断。
  - 如需设置、修改字符集与字符序，请参考**字符集和字符序**章节。

  示例：

```
gaussdb=# SELECT chr(65);
 chr
-----
 A
(1 row)

-- UTF-8字符集的情况下
gaussdb=# SELECT chr(19968);
 chr
-----
 一
(1 row)
```

- chr(cvalue int|bigint)

  描述：将cvalue转换为对应字节序列的字符返回。

  参数：cvalue支持类型可以转换为int或bigint的类型，取值范围为[0, 2^32 - 1]（对应unsigned int的范围），根据输入n的大小返回由1-4个字节组成的字符数组。其中在不同字符集中所返回的字节数组是相同的，但由于编码规则的不同会造成返回字符串的结果依赖于字符集编码。当字符集为单字节编码时，会先对cvalue进行mod 256运算后，返回对应的ASCII码字符。

  - 当输入的cvalue其中的某个字节为0的时候，输出会在该处截断。
  - 当输入不符合当前字符集的编码规则时会报错。
  - 当输入为NULL或者0时返回NULL。

  返回值类型：text

  示例：

```
gaussdb=# SELECT chr(65);
 chr
-----
 A
(1 row)
gaussdb=# CREATE DATABASE gaussdb_o WITH DBCOMPATIBILITY 'ORA';
gaussdb=# \c gaussdb_o
gaussdb_o=# SET a_format_version='10c';
gaussdb_o=# SET a_format_dev_version = 's1';
gaussdb_o=# SELECT chr(16705);
 chr
-----
 AA
(1 row)

-- 输出被截断
```


版权所有 © 华为云计算技术有限公司

```
gaussdb_o=# SELECT chr(4259905);
 chr
-----
 A
(1 row)
gaussdb_o=# \c postgres
gaussdb=#  DROP DATABASE gaussdb_o;
```

说明

在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下，此函数功能为根据输入大小返回对应字符序列：当前数据库编码字符集为多字节编码字符集时返回值为1-4个字节；当前数据库编码字符集为单字节编码字符集时返回值为将输入值通过mod 256运算后得到的单个字节。否则功能为：若当前数据库编码字符集为UTF-8字符集，则将输入作为unicode编码并返回一个UTF-8字符，若当前数据库编码字符集为其他字符集则返回ASCII码字符。

- nchr(cvalue int|bigint)

  描述：nchr(cvalue int|bigint)返回入参在国家字符集中对应的字符。国家字符集由GUC参数nls_nchar_characterset控制，仅支持AL16UTF16和UTF8：

  - 如果nls_nchar_characterset为AL16UTF16，入参超过两字节，会进行截断，保留低两位字节。
  - 如果nls_nchar_characterset为UTF8，入参超过三字节，会被当作0处理。

  该函数在ORA兼容模式数据库中且GUC参数a_format_version值为10c和a_format_dev_version值为s4的情况下有效。

  参数：cvalue支持可以转换成int或bigint的类型，取值范围为[0, 2^32 - 1]（对应unsigned int的范围）。入参若有小数部分，小数部分会被去掉。

  返回值类型：NVARCHAR2

  说明

  - 函数的返回值字节长度与Oracle数据库不一致。
  - 函数返回值受限于数据库字符集，在不同的数据库字符集下，如果没有对应的UTF8转数据库字符集的映射表，或者映射表中不存在该入参对应的UTF8编码时，表明当前数据库字符集不支持该UTF8字符，此时该函数的返回结果与Oracle数据库不一致。
  - 当前数据库字符集不支持入参对应的UTF8字符，或者国家字符集为UTF8时但是入参不符合UTF8格式，这两种情况会返回入参对应的字节数组。单个字节在[0x00-0x7F]范围，会返回一个ASCII码字符，在[0x80-0xFF]范围，会返回“?”。
  - 如需设置、修改字符集与字符序，请参考**字符集和字符序**章节。

- regexp_substr(source_char, pattern)

  描述：正则表达式的提取子串函数。SQL语法兼容ORA和MYSQL的情况下，GUC参数behavior_compat_options的值包含aformat_regexp_match时，“.”不能匹配 '\n' 字符；不包含aformat_regexp_match时，“.”能够匹配'\n'字符。

  参数：

  - source_char：用于匹配的源字符串。
  - pattern：用于匹配的正则表达式模式串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT regexp_substr('500 Hello World, Redwood Shores, CA', ',[^,]+,')
"REGEXPR_SUBSTR";
  REGEXPR_SUBSTR
--------------------
 , Redwood Shores,
(1 row)
```


版权所有 © 华为云计算技术有限公司

- regexp_replace(string, pattern, replacement [,flags])

  描述：替换匹配POSIX正则表达式的子字符串。 未匹配pattern时，返回原始的string串。 当存在匹配时，返回的string串中的匹配子串将被replacement串替换。replacement串可以包含\n， 其中\n是1到9， 表明string串匹配模式里第n个圆括号子表达式的子串应该被插入， 并且它可以包含\&表示应该插入匹配整个模式的子串。

  参数：

  - string：用于匹配的源字符串。
  - pattern：用于匹配的正则表达式模式串。
  - flags：可选参数，包含零个或多个改变函数匹配行为的单字母标记。g表示将所有匹配的子字符串替换为指定内容，而非仅替换第一个匹配项。其他flags支持的选项值及含义描述如**表1-42**所示。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT regexp_replace('Thomas', '.[mN]a.', 'M');
   regexp_replace
  ----------------
   ThM
  (1 row)
  gaussdb=# SELECT regexp_replace('foobarbaz','b(..)', E'X\\1Y', 'g') AS
  RESULT;
     result
  -------------
   fooXarYXazY
  (1 row)
  ```

- regexp_replace(string text, pattern text [, replacement text [, position int [, occurrence int [, flags text]]]])

  描述：替换匹配POSIX正则表达式的子字符串。未匹配pattern时，返回原始的string串。当存在匹配时，返回的string串中的匹配子串将被replacement串替换。

  参数：

  - string：用于匹配的源字符串。
  - pattern：用于匹配的正则表达式模式串。
  - replacement：可选参数，用于替换匹配子串的字符串。如果不给定参数值或者为null，表示用空串替换。
  - position：可选参数，表示从源字符串的第几个字符开始匹配，默认值为1。
  - occurrence：可选参数，表示替换第occurrence个匹配的子串。默认值为1，表示替换匹配到的第一个子串。
  - flags：可选参数，包含零个或多个改变函数匹配行为的单字母标记。flags支持的选项值及含义描述如**表1-42**所示。

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下，occurrence参数默认值为0，表示替换所有匹配到的子串。并且以'\'字符结尾的pattern参数为合法的。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT regexp_replace('foobarbaz','b(..)', E'X\\1Y', 2, 2, 'n') AS RESULT;
  result
  ------------
  ```


版权所有 © 华为云计算技术有限公司

```
foobarXazY
(1 row)
```

- concat_ws(sep text, str"any" [, str"any" [, ...]])

  描述：以第一个参数为分隔符，连接第二个以后的所有参数。NULL参数被忽略。

  须知

  - 如果第一个参数值是NULL，会导致返回结果为NULL。
  - 如果第一个参数值是空字符串（''），且数据库兼容模式设置为ORA的情况下，会导致返回结果为NULL。这是因为ORA兼容模式将''作为NULL处理，为避免此类情况，建议将数据库兼容模式改为MYSQL、TD或者PG。

  返回值类型：text

  示例：

```
gaussdb=# SELECT concat_ws(',', 'ABCDE', 2, NULL, 22);
 concat_ws
------------
 ABCDE,2,22
(1 row)
```

- nlssort(string text, sort_method text)

  描述：以sort_method指定的排序方式返回字符串在该排序模式下的编码值，该编码值可用于排序，其决定了string在这种排序模式下的先后位置。目前支持的sort_method为'nls_sort=schinese_pinyin_m'和'nls_sort=generic_m_ci'。其中，'nls_sort=generic_m_ci'仅支持纯英文不区分大小写排序。

  参数：string类型为text；sort_method类型为text。

  返回值类型：text

  示例：

```
gaussdb=# CREATE TABLE test(a text);

gaussdb=# INSERT INTO test(a) VALUES ('abC');

gaussdb=# INSERT INTO test(a) VALUES ('abC');

gaussdb=# INSERT INTO test(a) VALUES ('abc');

gaussdb=# SELECT * FROM test ORDER BY nlssort(a,'nls_sort=schinese_pinyin_m');
   a
--------
 abc
 abC
 abC
(3 rows)

gaussdb=# SELECT * FROM test ORDER BY nlssort(a, 'nls_sort=generic_m_ci');
   a
--------
 abC
 abc
 abC
(3 rows)

gaussdb=# DROP TABLE test;
```

- convert(string bytea, src_encoding name, dest_encoding name)

  描述：以dest_encoding指定的目标编码方式转换字符串string。src_encoding指定源编码方式，在该编码下，string必须是合法的。

版权所有 © 华为云计算技术有限公司

返回值类型：bytea

示例：

```
gaussdb=# SELECT convert('text_in_utf8', 'UTF8', 'GBK');
         convert
-----------------------------
 \x746578745f696e5f75746638
(1 row)
```

说明

如果源编码到目标编码方式的转换规则不存在，则字符串不进行任何转换直接返回(如GBK和LATIN1之间无转换规则），具体转换规则可以通过查询系统表PG_CONVERSION获得。server_encoding在数据库初始化时指定，决定数据库的字符编码方式。

示例：

```
gaussdb=# SHOW server_encoding;
 server_encoding
-----------------
 LATIN1
(1 row)

gaussdb=# SELECT convert_from('some text', 'GBK');
 convert_from
--------------
 some text
(1 row)

db_latin1=# SELECT convert_to('some text', 'GBK');
      convert_to
----------------------
 \x736f6d652074657874
(1 row)

db_latin1=# SELECT convert('some text', 'GBK', 'LATIN1');
       convert
----------------------
 \x736f6d652074657874
(1 row)
```

- convert(expr, using transcoding_name)

  描述：接收一个参数，可以是字符也可以是数字，将其转换为一个字符编码为transcoding_name类型的字符串并返回。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT convert('asdas' using 'gbk');
   convert
  ---------
   asdas
  (1 row)
  ```

  说明

  此函数仅在MYSQL模式数据库中生效。

- convert_from(string bytea, src_encoding name)

  描述：以数据库的字符编码方式转换字符串string。src_encoding指定源编码方式，在该编码下，string必须是合法的。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT convert_from('text_in_utf8', 'UTF8');
   convert_from
  --------------
  ```


版权所有 © 华为云计算技术有限公司

```
 text_in_utf8
(1 row)
```

- convert_to(string text, dest_encoding name)

  描述：将字符串转换为dest_encoding的编码格式。

  返回值类型：bytea

  示例：

```
gaussdb=# SELECT convert_to('some text', 'UTF8');
     convert_to
----------------------
 \x736f6d652074657874
(1 row)
```

- string [NOT] LIKE pattern [ESCAPE escape-character]

  描述：模式匹配函数。

  - 当pattern不包含百分号或者下划线，该模式仅代表它本身，此时LIKE的行为等同于等号操作符。pattern中的下划线（_）匹配任何单个字符；而一个百分号可（%）匹配零或多个任何字符。
  - 若需匹配下划线（_）或者百分号（%），需要在pattern中相应的字符前导逃逸字符。缺省的逃逸字符是反斜杠（\），但是用户可以用ESCAPE子句自定义逃逸字符。在匹配逃逸字符本身时，需要输入两个逃逸字符。

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT 'AA_BBCC' LIKE '%A@_B%' ESCAPE '@' AS RESULT;
 result
--------
 t
(1 row)
gaussdb=# SELECT 'AA_BBCC' LIKE '%A@_B%' AS RESULT;
 result
--------
 f
(1 row)
gaussdb=# SELECT 'AA@_BBCC' LIKE '%A@_B%' AS RESULT;
 result
--------
 t
(1 row)
```

- regexp_like(source_string, pattern [, match_parameter])

  描述：正则表达式的模式匹配函数。source_string为源字符串，pattern为正则表达式匹配模式。 match_parameter为匹配选项，可取值为：

  - 'i'：大小写不敏感。
  - 'c'：大小写敏感。
  - 'm'：将source_string视为多行。

  若忽略match_parameter选项，默认为大小写敏感，“.”不匹配换行符，source_string视为单行。当输入不支持的match_parameter选项，会产生提示信息“INFO: illegal argument for function”，并返回结果false。

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT regexp_like('ABC', '[A-Z]');
 regexp_like
-------------
 t
(1 row)
gaussdb=# SELECT regexp_like('ABC', '[D-Z]');
 regexp_like
```


版权所有 © 华为云计算技术有限公司

```
-------------
 f
(1 row)
gaussdb=# SELECT regexp_like('ABC', '[A-Z]','i');
 regexp_like
-------------
 t
(1 row)
gaussdb=# SELECT regexp_like('ABC', '[A-Z]');
 regexp_like
-------------
 t
(1 row)
```

- format(formatstr text [, str"any" [, ...]])

  描述：格式化字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT format('Hello %s, %1$s', 'World');
      format
--------------------
 Hello World, World
(1 row)
```

- md5(string)

  描述：对string进行MD5加密处理，并以16进制数作为返回值。

  说明

  MD5加密算法安全性低，存在安全风险，建议使用更安全的加密算法。

  返回值类型：text

  示例：

```
gaussdb=# SELECT md5('ABC');
               md5
----------------------------------
 902fbdd2b1df0c4f70b4a5d23525e932
(1 row)
```

- sha(string) / sha1(string)

  描述：对string进行SHA1加密处理，并以16进制数作为返回值，sha和sha1函数功能相同。

  说明

  - SHA1加密算法安全性低，存在安全风险，不建议使用。
  - 该函数仅在GaussDBMYSQL兼容模式下（即sql_compatibility = 'MYSQL'）有效。

  返回值类型：text

  示例：

```
gaussdb=# SELECT sha('ABC');
                   sha
------------------------------------------
 3c01bdbb26f358bab27f267924aa2c9a03fcfdb8
(1 row)
gaussdb=# SELECT sha1('ABC');
                   sha1
------------------------------------------
 3c01bdbb26f358bab27f267924aa2c9a03fcfdb8
(1 row)
```

- sha2(string, hash_length)

  描述：对string进行SHA2加密处理，并以16进制数作为返回值。

版权所有 © 华为云计算技术有限公司

参数：hash_length对应相应的SHA2算法，可选值为 0(SHA-256)、224(SHA-224)、256(SHA-256)、384(SHA-384)、512(SHA-512)，其他值将返回NULL。

说明

- SHA224加密算法安全性低，存在安全风险，不建议使用。
- SHA2函数会在日志中记录哈希的明文，因此不建议用户使用该函数加密密钥等敏感信息。
- 该函数仅在GaussDBMYSQL兼容模式下（即sql_compatibility = 'MYSQL'）有效。

返回值类型：text

示例：

```
gaussdb=# SELECT sha2('ABC',224);
                            sha2
------------------------------------------------------------
 107c5072b799c4771f328304cfe1ebb375eb6ea7f35a3aa753836fad
(1 row)
gaussdb=# SELECT sha2('ABC',256);
                              sha2
----------------------------------------------------------------
 b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
(1 row)
gaussdb=# SELECT sha2('ABC',0);
                              sha2
----------------------------------------------------------------
 b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
(1 row)
```

- decode(string text, format text)

  描述：将二进制数据从文本数据中解码。

  返回值类型：bytea

  示例：

  ```
  gaussdb=# SELECT decode('MTIzAAE=', 'base64');
     decode
  -------------
   \x3132330001
  (1 row)
  ```

- similar_escape(pat text, esc text)

  描述：将一个SQL:2008风格的正则表达式转换为POSIX风格。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT similar_escape('\s+ab','2');
   similar_escape
  ----------------
   ^(?:\\s+ab)$
  (1 row)
  ```

- find_in_set(text, set)

  描述：查找给定成员在集合中的位置，从1开始计数。如果没有找到，返回0。分布式暂不支持SET数据类型，此函数执行时会报错。

  返回值类型：int2

- find_in_set(str, strlist)

  描述：查询字段strlist中是否包含str的结果，如有则返回str在strlist中的位置。如果字符串str不在strlist中或者strlist为空字符串，则返回值为0。

  参数：如**表1-44**所示。

版权所有 © 华为云计算技术有限公司

表 1-44 find_in_set(str, strlist)参数说明

参数名称	类型	描述
str	text	str为用户需要查询的字符串。
strlist	text	strlist是以“,”为分隔符，不同字符串通过分隔符组合在一起的字符串集合，strlist本身也为字符串。

返回值类型：int

示例：

```
gaussdb=# SELECT find_in_set('ee','a,ee,c');
 find_in_set
-------------
           2
(1 row)
```

- encode(data bytea, format text)

  描述：将二进制数据编码为文本数据。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT encode(E'123\\000\\001', 'base64');
    encode
  ----------
   MTIzAAE=
  (1 row)
  ```

- strcmp(expr1, expr2)

  描述：根据当前字符序比较两个输入字符串，字符串相等返回0，第一个字符串小于第二个字符串返回-1，否则返回1。

  参数：如表1-45所示。

版权所有 © 华为云计算技术有限公司

表 1-45 strcmp 参数说明

参数名称	类型	描述
expr1/expr2	• 字符类型：CHAR、VARCHAR、NVARCHAR2、TEXT。 • 二进制类型：BYTEA。 • 数值类型：TINYINT [UNSIGNED]、SMALLINT [UNSIGNED]、INTEGER [UNSIGNED]、BIGINT [UNSIGNED]、FLOAT4、FLOAT8、NUMERIC。 • 日期时间类型：DATE、TIME WITHOUT TIME ZONE、DATETIME、TIMESTAMPTZ。	表示参与比较的字符串。

返回值类型：INTEGER

示例：

```
-- 切换MYSQL兼容模式数据库。
gaussdb=# CREATE DATABASE gaussdb_m dbcompatibility='MYSQL';
gaussdb=# \c gaussdb_m

-- 设置兼容版本控制参数，开启常量字符串指定字符序功能。
gaussdb_m=# SET b_format_version='5.7';
gaussdb_m=# SET b_format_dev_version='s2';

gaussdb_m=# SELECT strcmp('abc', 'ABC');
 strcmp
--------
      0
(1 row)

gaussdb_m=# SELECT strcmp('abc     ', 'abc');
 strcmp
--------
      0
(1 row)

gaussdb_m=# SELECT strcmp('1', 1);
 strcmp
--------
      0
(1 row)

gaussdb_m=# SELECT strcmp(123, 2);
 strcmp
--------
```


版权所有 © 华为云计算技术有限公司

```
   -1
(1 row)
```

说明

- strcmp函数仅在sql_compatibility='MYSQL'时有效；
- b_format_version='5.7'和b_format_dev_version='s1'版本开始，sql_mode参数"pad_char_to_full_length"控制对CHAR类型尾部填充空格，会影响strcmp的比较结果，具体请参见**表1-8**。
- b_format_version='5.7'和b_format_dev_version='s1'版本开始，字符类型、二进制类型、数值类型、日期时间类型行为兼容M，会影响strcmp的比较结果，具体请参见**数据类型**。对于数值类型中的浮点类型，由于连接参数设置不同，精度可能与M有差异，不建议使用该场景，或使用NUMERIC类型代替。
- b_format_version='5.7'和b_format_dev_version='s2'版本开始，支持字符转义和常量字符串获取字符序，字符序会影响strcmp的比较结果，具体请参见**SET**章节SET NAMES语法，字符类型不同字符序间的合并规则请参见**字符集和字符序合并规则**。

说明

- 若字符串中存在换行符，如字符串由一个换行符和一个空格组成，在GaussDB中LENGTH和LENGTHB的值为2。
- 对于CHAR(n) 类型，GaussDB中n是指字符个数。因此，对于多字节编码的字符集，LENGTHB函数返回的长度可能大于n。
- GaussDB支持多种类型的数据库，目前有4种，分别是ORA类型、MYSQL类型、TD类型以及PG类型。ORA的词法分析器与另外三种不一样，在ORA中空字符串会被当作是NULL。所以，当使用ORA类型的数据库时，假如上述字符操作函数中有空字符串作为参数，会出现没有输出的情况。例如：

```
gaussdb=# SELECT translate('12345','123','');
translate
-----------

(1 row)
```

  这是因为数据库在调用相应的函数进行处理前，会判断所输入的参数中是否含有NULL，假如有，则不会调用相应的函数，因此会没有输出。而在PG模式下，字符串的处理方式与postgresql保持一致，因此不会有上述问题产生。

扩展函数和操作符

- pkg_bpchar_opc()

  描述：扩展接口，用于新增bpchar和text或者text和bpchar策略比较操作符，解决bpchar类型和text类型数据比较无法命中索引的问题。仅初始用户可以安装扩展。

  示例：

  - bpchar类型和text类型比较时（初始状态，前向兼容）：

```
/*
建表和数据初始化。
*/
gaussdb=# CREATE TABLE logs_nchar(log_id nchar(16), log_message text);
gaussdb=# INSERT INTO logs_nchar SELECT GENERATE_SERIES(1,100000),MD5(RANDOM());
gaussdb=# INSERT INTO logs_nchar VALUES ('FE306991300002  ','002');
gaussdb=# CREATE INDEX idx_nchar_logid ON logs_nchar(log_id);
gaussdb=# VACUUM ANALYZE logs_nchar;

/*
目的就是获得所有节点的详细执行计划。
*/
gaussdb=# SET max_datanode_for_plan = 1;
SET

/*
```


版权所有 © 华为云计算技术有限公司

```
当没有安装扩展，对nchar和text比较时，由于没有bpchar和text索引操作符，nchar会隐式转换为text,即定长字符类型转换成变长字符类型，导致执行计划发生变化，从而无法命中索引。
*/
gaussdb=# EXPLAIN SELECT * FROM logs_nchar WHERE log_id = RPAD(TRIM('FE306991300002 '),16,' ');
                                                 QUERY
PLAN
-------------------------------------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT log_id, log_message FROM public.logs_nchar WHERE log_id::text = rpad(btrim('FE306991300002  '::text), 16, ' '::text)
 Datanode Name: datanode1
   Seq Scan on logs_nchar  (cost=0.00..2539.01 rows=500 width=50)
     Filter: ((log_id)::text = 'FE306991300002  '::text)

(8 rows)

/*
表logs_nchar里log_id字段类型是nchar(16)，插入数据为'FE306991300002  '，隐式转换成text类型，进行比较时，会把后面空格去掉进行比较，即'FE306991300002'='FE306991300002  '，所以不命中数据。
*/
gaussdb=# SELECT * FROM logs_nchar WHERE log_id = RPAD(TRIM('FE306991300002  '),16,' ');
 log_id | log_message
--------+-------------
(0 rows)

/*
删除表。
*/
gaussdb=# DROP TABLE logs_nchar;
```

- bpchar类型和text类型比较时（安装pkg_bpchar_opc扩展，与ORA保持一致）：

```
/*
建表和数据初始化。
*/
gaussdb=# CREATE TABLE logs_nchar(log_id nchar(16), log_message text);
gaussdb=# INSERT INTO logs_nchar SELECT GENERATE_SERIES(1,100000),MD5(RANDOM());
gaussdb=# INSERT INTO logs_nchar VALUES ('FE306991300002  ','002');
gaussdb=# CREATE INDEX idx_nchar_logid ON logs_nchar(log_id);
gaussdb=# VACUUM ANALYZE logs_nchar;

/*
系统管理员安装pkg_bpchar_opc扩展，数据库增加了bpchar和text类型比较操作符，以及索引相关内容。
*/
gaussdb=# CREATE EXTENSION pkg_bpchar_opc;
CREATE EXTENSION

gaussdb=# SET max_datanode_for_plan = 1;
SET

gaussdb=# EXPLAIN SELECT * FROM logs_nchar WHERE log_id = RPAD(TRIM('FE306991300002 '),16,' ');
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT log_id, log_message FROM public.logs_nchar WHERE log_id = rpad(btrim('FE306991300002  '::text), 16, ' '::text)
 Datanode Name: datanode1
   [Bypass]
   Index Scan using idx_nchar_logid on logs_nchar  (cost=0.00..8.27 rows=1 width=50)
     Index Cond: (log_id = 'FE306991300002  '::text)
```


版权所有 © 华为云计算技术有限公司

```
(9 rows)

/*
此时，log_id隐式转换为bpchar类型时，和text类型比较时，能找到比较操作符以及索引信息，能命中索引。
*/
gaussdb=# SELECT * FROM logs_nchar WHERE log_id = RPAD(TRIM('FE306991300002  '),16,' ');
      log_id      | log_message
------------------+-------------
 FE306991300002   | 002
(1 row)

/*
删除表和扩展。
*/
gaussdb=# DROP TABLE logs_nchar;
gaussdb=# DROP EXTENSION pkg_bpchar_opc;
```

- text类型和bpchar类型比较时（初始状态，前向兼容）：

```
/*
建表和数据初始化。
*/
gaussdb=# CREATE TABLE logs_text(log_id nchar(16), log_message text);
gaussdb=# INSERT INTO logs_text SELECT GENERATE_SERIES(1,100000),MD5(RANDOM());
gaussdb=# INSERT INTO logs_text VALUES ('FE306991300002  ','002');
gaussdb=# CREATE INDEX idx_text_logid ON logs_text(log_id);
gaussdb=# VACUUM ANALYZE logs_text;

gaussdb=# SET max_datanode_for_plan = 1;
SET

gaussdb=# EXPLAIN SELECT * FROM logs_text WHERE log_id = 'FE306991300002  '::bpchar;
                                        QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT log_id, log_message FROM public.logs_text WHERE log_id =
'FE306991300002  '::bpchar::text
 Datanode Name: datanode1
   [Bypass]
   Index Scan using idx_text_logid on logs_text  (cost=0.00..8.27 rows=1 width=38)
     Index Cond: (log_id = 'FE306991300002'::text)

(9 rows)

gaussdb=# SELECT * FROM logs_text WHERE log_id = 'FE306991300002  '::bpchar;
 log_id | log_message
--------+-------------
(0 rows)

gaussdb=# SELECT * FROM logs_text WHERE log_id = 'FE306991300002  '::text;
      log_id      | log_message
------------------+-------------
 FE306991300002   | 002
(1 row)

/*
删除表。
*/
gaussdb=# DROP TABLE logs_text;
```

- text类型和bpchar类型比较时（安装pkg_bpchar_opc扩展，与ORA保持一致）：

```
/*
建表和数据初始化。
*/
gaussdb=# CREATE TABLE logs_text(log_id nchar(16), log_message text);
gaussdb=# INSERT INTO logs_text SELECT GENERATE_SERIES(1,100000),MD5(RANDOM());
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO logs_text VALUES ('FE306991300002  ','002');
gaussdb=# CREATE INDEX idx_text_logid on logs_text(log_id);
gaussdb=# VACUUM ANALYZE logs_text;

gaussdb=# CREATE EXTENSION pkg_bpchar_opc;
CREATE EXTENSION

gaussdb=# SET max_datanode_for_plan = 1;
SET

gaussdb=# EXPLAIN SELECT * FROM logs_text t1 WHERE t1.log_id ='FE306991300002  '::bpchar;
                                        QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT log_id, log_message FROM public.logs_text t1 WHERE log_id =
'FE306991300002  '::bpchar
 Datanode Name: datanode1
   [Bypass]
   Index Scan using idx_text_logid on logs_text t1  (cost=0.00..8.27 rows=1 width=38)
     Index Cond: (log_id = 'FE306991300002  '::bpchar)

(9 rows)

gaussdb=# SELECT * FROM logs_text WHERE log_id = 'FE306991300002  '::bpchar;
      log_id      | log_message
------------------+-------------
 FE306991300002   | 002
(1 row)

gaussdb=# SELECT * FROM logs_text WHERE log_id = 'FE306991300002  '::text;
      log_id      | log_message
------------------+-------------
 FE306991300002   | 002
(1 row)

/*
删除表和扩展。
*/
gaussdb=# DROP TABLE logs_text;
gaussdb=# DROP EXTENSION pkg_bpchar_opc;
```

- hash join、text类型和bpchar类型比较时（初始状态，前向兼容）：

```
/*
建表和数据初始化。
*/
gaussdb=# CREATE TABLE logs_varchar2(log_id varchar2, log_message text);
gaussdb=# INSERT INTO logs_varchar2 VALUES ('FE306991300002  ','002');
gaussdb=# INSERT INTO logs_varchar2 VALUES ('FE306991300003  ','003');
gaussdb=# INSERT INTO logs_varchar2 VALUES ('FE306991300004  ','004');
gaussdb=# VACUUM ANALYZE logs_varchar2;

gaussdb=# CREATE TABLE logs_char(log_id char(16), log_message text);
gaussdb=# INSERT INTO logs_char VALUES ('FE306991300002  ','002');
gaussdb=# INSERT INTO logs_char VALUES ('FE306991300003  ','003');
gaussdb=# INSERT INTO logs_char VALUES ('FE306991300004  ','004');
gaussdb=# VACUUM ANALYZE logs_char;

gaussdb=# SET max_datanode_for_plan = 1;
SET
gaussdb=# SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = t2.log_id;
      log_id      | log_message |      log_id      | log_message
------------------+-------------+------------------+-------------
 FE306991300002   | 002         | FE306991300002   | 002
 FE306991300003   | 003         | FE306991300003   | 003
 FE306991300004   | 004         | FE306991300004   | 004
(3 rows)

gaussdb=# EXPLAIN SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = t2.log_id;
```


版权所有 © 华为云计算技术有限公司

```
                                                                    QUERY
PLAN
------------------------------------------------------------------------------------------------------------
----------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT t1.log_id, t1.log_message, t2.log_id, t2.log_message FROM
public.logs_varchar2 t1, public.logs_char t2 WHERE t1.log_id::bpchar = t2.log_id
 Datanode Name: datanode1
   Hash Join  (cost=1.07..2.14 rows=3 width=42)
     Hash Cond: ((t1.log_id)::bpchar = t2.log_id)
     ->  Seq Scan on logs_varchar2 t1  (cost=0.00..1.03 rows=3 width=21)
     ->  Hash  (cost=1.03..1.03 rows=3 width=21)
           ->  Seq Scan on logs_char t2  (cost=0.00..1.03 rows=3 width=21)

(11 rows)

gaussdb=# SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = 'FE306991300002
';
      log_id      | log_message |      log_id      | log_message
------------------+-------------+------------------+-------------
 FE306991300002   | 002         | FE306991300002   | 002
 FE306991300002   | 002         | FE306991300003   | 003
 FE306991300002   | 002         | FE306991300004   | 004
(3 rows)

/*
删除表。
*/
gaussdb=# DROP TABLE logs_varchar2;
gaussdb=# DROP TABLE logs_char;
```

– hash join、text类型和bpchar类型比较时（安装pkg_bpchar_opc扩展，与ORA保持一致）：

```
/*
建表和数据初始化。
*/
gaussdb=# CREATE TABLE logs_varchar2(log_id varchar2, log_message text);
gaussdb=# INSERT INTO logs_varchar2 VALUES ('FE306991300002  ','002');
gaussdb=# INSERT INTO logs_varchar2 VALUES ('FE306991300003  ','003');
gaussdb=# INSERT INTO logs_varchar2 VALUES ('FE306991300004  ','004');
gaussdb=# VACUUM ANALYZE logs_varchar2;

gaussdb=# CREATE TABLE logs_char(log_id char(16), log_message text);
gaussdb=# INSERT INTO logs_char VALUES ('FE306991300002  ','002');
gaussdb=# INSERT INTO logs_char VALUES ('FE306991300003  ','003');
gaussdb=# INSERT INTO logs_char VALUES ('FE306991300004  ','004');
gaussdb=# VACUUM ANALYZE logs_char;

gaussdb=# CREATE EXTENSION pkg_bpchar_opc;
CREATE EXTENSION

gaussdb=# SET max_datanode_for_plan = 1;
SET

/*
不推荐写法，由于t1表的log_id的varchar2类型，受安装扩展影响，隐式转换为text类型，和t2表的
log_id比较，t2表的log_id类型会由char隐式转换为bpchar类型，此时log_id后面的空格会被数据库去
掉，即'FE306991300002'='FE306991300002  '，所以不命中数据。
*/
/*
错误示例：
*/
gaussdb=# SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = t2.log_id;
 log_id | log_message | log_id | log_message
--------+-------------+--------+-------------
(0 rows)

gaussdb=# EXPLAIN SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = t2.log_id;
```

```
                                                        QUERY
PLAN
----------------------------------------------------------------------------------------------------------
--------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT t1.log_id, t1.log_message, t2.log_id, t2.log_message FROM
public.logs_varchar2 t1, public.logs_char t2 WHERE t1.log_id::text = t2.log_id
 Datanode Name: datanode1
   Hash Join  (cost=1.07..2.14 rows=3 width=42)
     Hash Cond: ((t1.log_id)::text = t2.log_id)
     ->  Seq Scan on logs_varchar2 t1  (cost=0.00..1.03 rows=3 width=21)
     ->  Hash  (cost=1.03..1.03 rows=3 width=21)
           ->  Seq Scan on logs_char t2  (cost=0.00..1.03 rows=3 width=21)

(11 rows)

gaussdb=# SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = 'FE306991300002
';
      log_id      | log_message |      log_id      | log_message
------------------+-------------+------------------+-------------
 FE306991300002   | 002         | FE306991300002   | 002
 FE306991300002   | 002         | FE306991300003   | 003
 FE306991300002   | 002         | FE306991300004   | 004
(3 rows)

/*
推荐写法，为了避免t1表的log_id的数据类型转换成text类型时保留空格，导致与t2表的log_id比较无
法命中数据，可以将t1表类型强转成没安装扩展前的bpchar类型，即
'FE306991300002'='FE306991300002'，以确保命中数据。
*/
/*
正确示例：
*/
gaussdb=# SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id::bpchar = t2.log_id;
      log_id      | log_message |      log_id      | log_message
------------------+-------------+------------------+-------------
 FE306991300002   | 002         | FE306991300002   | 002
 FE306991300003   | 003         | FE306991300003   | 003
 FE306991300004   | 004         | FE306991300004   | 004
(3 rows)

/*
执行计划和没安装扩展前是一致的。
*/
gaussdb=# EXPLAIN SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id::bpchar =
t2.log_id;
                                                         QUERY
PLAN
----------------------------------------------------------------------------------------------------------
----------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT t1.log_id, t1.log_message, t2.log_id, t2.log_message FROM
public.logs_varchar2 t1, public.logs_char t2 WHERE t1.log_id::bpchar = t2.log_id
 Datanode Name: datanode1
   Hash Join  (cost=1.07..2.14 rows=3 width=42)
     Hash Cond: ((t1.log_id)::bpchar = t2.log_id)
     ->  Seq Scan on logs_varchar2 t1  (cost=0.00..1.03 rows=3 width=21)
     ->  Hash  (cost=1.03..1.03 rows=3 width=21)
           ->  Seq Scan on logs_char t2  (cost=0.00..1.03 rows=3 width=21)

(11 rows)

gaussdb=# SELECT * FROM logs_varchar2 t1, logs_char t2 WHERE t1.log_id = 'FE306991300002
';
      log_id      | log_message |      log_id      | log_message
```

```
------------------+-------------+------------------+-------------
 FE306991300002   | 002         | FE306991300002   | 002
 FE306991300002   | 002         | FE306991300003   | 003
 FE306991300002   | 002         | FE306991300004   | 004
(3 rows)

/*
索引的bpchar与text隐式转换。
*/
gaussdb=# CREATE DATABASE ora_compatible_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE
gaussdb=# \c ora_compatible_db
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora_compatible_db" as user "tom".
ora_compatible_db=# CREATE EXTENSION pkg_bpchar_opc;
CREATE EXTENSION
ora_compatible_db=# CREATE TABLE t1_1(c1 char(5), c2 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by
default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_compatible_db=# ALTER TABLE t1_1 ADD CONSTRAINT t1_1_pkey PRIMARY KEY (c1) WITH
(storage_type=USTORE);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "t1_1_pkey" for table
"t1_1"
ALTER TABLE
ora_compatible_db=# INSERT INTO t1_1 VALUES('abc  ', 0);
INSERT 0 1
ora_compatible_db=# CREATE TABLE t1_2(c1 varchar(5));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by
default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
ora_compatible_db=# INSERT INTO t1_2 VALUES('abc  ');
INSERT 0 1
ora_compatible_db=# CREATE /*+INDEXSCAN (t1_1 t1_1_pkey)*/ c1 FROM t1_1 WHERE c2 = 0
AND c1 IN (SELECT c1 FROM t1_2);
  c1
-------
 abc
(1 row)
ora_compatible_db=# SET max_datanode_for_plan = 64;
SET
ora_compatible_db=# EXPLAIN CREATE /*+INDEXSCAN (t1_1 t1_1_pkey)*/ c1 FROM t1_1
WHERE c2 = 0 AND c1 IN (SELECT c1 FROM t1_2);
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes

 Remote SQL: SELECT/*+ IndexScan(t1_1 t1_1_pkey)*/ c1 FROM public.t1_1 WHERE c2 = 0 AND
(c1 IN (SELECT t1_2.c1 FROM public.t1_2))
 Datanode Name: dist1_datanode1
   Nested Loop  (cost=28.91..93.76 rows=4 width=24)
     ->  HashAggregate  (cost=28.91..30.91 rows=200 width=24)
           Group By Key: (t1_2.c1)::text
           ->  Seq Scan on t1_2  (cost=0.00..25.13 rows=1513 width=24)
     ->  Index Scan using t1_1_pkey on t1_1  (cost=0.00..0.30 rows=1 width=24)
           Index Cond: (c1 = (t1_2.c1)::text)
           Filter: (c2 = 0)

(13 rows)
/*
删除表和扩展。
*/
gaussdb=# DROP TABLE t1_1;
gaussdb=# DROP TABLE t1_2;
gaussdb=# DROP TABLE logs_varchar2;
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE logs_char;
gaussdb=# DROP EXTENSION pkg_bpchar_opc;
```

说明

- 为了解决bpchar类型（包含多个后补空格）与text类型做等值匹配操作的时候无法正常匹配数据以及索引问题。
- 涉及ubtree, btree，比较操作符包含： >、>=、<、<=、 <>。
- 影响面涉及字符类型之间的隐式转换，例如：变长和定长数据类型比较时，变长会优先转换为text类型，而不是最初的bpchar类型。
- 默认不开启pkg_bpchar_opc扩展。检验扩展启用状态可通过查询系统表PG_EXTENSION，存在对应记录则表示扩展已启用，没有则表示未开启。关闭扩展时，保持前向兼容，开启扩展时，保持与ORA数据库兼容等。开启pkg_bpchar_opc扩展后，需要设置max_datanode_for_plan参数（该参数可以控制生成FQS计划时设置显示DN上执行计划的个数），根据自身需求去设置。以设置max_datanode_for_plan参数为64为例，命令如下：
  ```
  SET max_datanode_for_plan = 64;
  ```
- 示例中所用的表结构都是以log_id为索引，共有log_id，log_message两个字段，表名下划线后面接的是log_id字段类型（例如：表名为logs_text,则log_id字段类型为text）。

表 1-46 pkg_bpchar_opc 支持的函数说明

接口名称	描述
pg_catalog.bpchar_text_lt	bpchar类型和text类型比较，左边数值是否小于右边的数值。
pg_catalog.bpchar_text_le	bpchar类型和text类型比较，左边数值是否小于等于右边的数值。
pg_catalog.bpchar_text_eq	bpchar类型和text类型比较，左边数值是否等于右边的数值。
pg_catalog.bpchar_text_ge	bpchar类型和text类型比较，左边数值是否大于等于右边的数值。
pg_catalog.bpchar_text_gt	bpchar类型和text类型比较，左边数值是否大于右边的数值。
pg_catalog.bpchar_text_ne	bpchar类型和text类型比较，左边数值是否不等于右边的数值。
pg_catalog.bpchar_text_cmp	bpchar类型和text类型的索引支持比较函数。
pg_catalog.text_bpchar_lt	text类型和bpchar 类型比较，左边数值是否小于右边的数值。
pg_catalog.text_bpchar_le	text类型和bpchar类型比较，左边数值是否小于等于右边的数值。
pg_catalog.text_bpchar_eq	text类型和bpchar类型比较，左边数值是否等于右边的数值。
pg_catalog.text_bpchar_ge	text类型和bpchar类型比较，左边数值是否大于等于右边的数值。

版权所有 © 华为云计算技术有限公司

接口名称	描述
pg_catalog.text_bpchar_gt	text类型和bpchar类型比较，左边数值是否大于右边的数值。
pg_catalog.text_bpchar_ne	text类型和bpchar类型比较，左边数值是否不等于右边的数值。
pg_catalog.text_bpchar_cmp	text类型和bpchar类型的索引支持比较函数。
pg_catalog.hashbpchartext	bpchar类型和text类型的hash支持比较函数。
pg_catalog.hashtextbpchar	text类型和bpchar类型的hash支持比较函数。

- bpcharlikebpchar(BPCHAR, BPCHAR)

  描述：判断第一个入参BPCHAR字符串是否LIKE第二个入参BPCHAR字符串。用于新增BPCHAR类型和BPCHAR类型的LIKE操作符。启用“~~”操作符需在GUC参数behavior_compat_options的取值中包含enable_bpcharlikebpchar_compare配置项。

  返回值类型：boolean，其取值范围为：

  - t：表示两个BPCHAR类型参数匹配。
  - f：表示两个BPCHAR类型参数不匹配。

  示例：

```
gaussdb=#  SELECT bpcharlikebpchar('455'::BPCHAR(10), '455  '::BPCHAR);
 bpcharlikebpchar
------------------
 f
(1 row)
gaussdb=#  SELECT bpcharlikebpchar('455 '::BPCHAR(10), '455 '::BPCHAR(10));
 bpcharlikebpchar
------------------
 t
(1 row)
gaussdb=#  SELECT bpcharlikebpchar('455 '::BPCHAR(10), '455'::BPCHAR(10));
 bpcharlikebpchar
------------------
 t
(1 row)
gaussdb=#  SELECT bpcharlikebpchar('455 '::BPCHAR(10), '455'::BPCHAR(11));
 bpcharlikebpchar
------------------
 f
(1 row)
gaussdb=#  CREATE TABLE op_test (
col BPCHAR(2) DEFAULT NULL
 );
CREATE TABLE
gaussdb=#  CREATE INDEX op_index ON op_test(col);
CREATE INDEX
gaussdb=#  INSERT INTO op_test VALUES ('a');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('1');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('11');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('12');
INSERT 0 1
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=#  INSERT INTO op_test VALUES ('sd');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('aa');
INSERT 0 1
gaussdb=#  SHOW behavior_compat_options;
 behavior_compat_options
-------------------------
(1 row)
--当behavior_compat_options不包含enable_bpcharlikebpchar_compare时，则未启用最新
bpcharlikebpchar操作符，其bpchar匹配bpchar返回结果集非预期（正常应返回全部数据）。
gaussdb=# EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY
col;
               QUERY PLAN
-----------------------------------------------
 Streaming (type: GATHER)
   Merge Sort Key: col
   Node/s: All datanodes
   ->  Sort
         Sort Key: col
         ->  Seq Scan on op_test
               Filter: (col !~~ (col)::text)
(7 rows)
gaussdb=#  SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col;
 col
-----
 11
 12
 aa
 sd
(4 rows)
gaussdb=#  SET behavior_compat_options = 'enable_bpcharlikebpchar_compare';
SET
gaussdb=#  SHOW behavior_compat_options;
     behavior_compat_options
---------------------------------
 enable_bpcharlikebpchar_compare
(1 row)
--开启参数后，将启用最新bpcharlikebpchar操作符，其匹配时返回行为符合预期。
gaussdb=#  EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY
col;
            QUERY PLAN
-----------------------------------
 Streaming (type: GATHER)
   Merge Sort Key: col
   Node/s: All datanodes
   ->  Sort
         Sort Key: col
         ->  Seq Scan on op_test
               Filter: (col ~~ col)
(7 rows)
gaussdb=#  SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col;
 col
-----
 1
 11
 12
 a
 aa
 sd
(6 rows)
gaussdb=#  DROP TABLE op_test;
DROP TABLE
```


版权所有 © 华为云计算技术有限公司

说明

- 仅在数据库兼容性参数SQL_COMPATIBILITY为ORA时，设置GUC参数behavior_compat_options中含有enable_bpcharlikebpchar_compare字段，则可以使用bpcharlikebpchar的匹配。
- 启用该特性，会影响BPCHAR类型进行LIKE模式匹配时的结果集与执行计划。
- GUC参数behavior_compat_options中不含有enable_bpcharlikebpchar_compare字段，表示关闭该特性；GUC参数behavior_compat_options中含有enable_bpcharlikebpchar_compare字段，表示开启该特性。
- 开启新特性后，定长匹配定长（bpchar匹配bpchar），需要指定左右两侧参数长度相同；模式匹配时，注意模式列长度与强制转换长度相同，避免过长后填补空格导致结果与预期存在差异。

- bpcharnlikebpchar(BPCHAR, BPCHAR)

  描述：判断第一个入参BPCHAR字符串是否NOT LIKE第二个入参BPCHAR字符串。用于新增BPCHAR类型和BPCHAR类型的NOT LIKE操作符。启用“!~~”操作符需在GUC参数behavior_compat_options的取值中包含enable_bpcharlikebpchar_compare配置项。

  返回值类型：boolean，其取值范围为：

  - t：表示两个BPCHAR类型参数不匹配。
  - f：表示两个BPCHAR类型参数匹配。

  示例：

```
gaussdb=#  SELECT bpcharnlikebpchar('455 '::BPCHAR(10), '455'::BPCHAR(11));
 bpcharnlikebpchar
-------------------
 t
(1 row)

gaussdb=#  SELECT bpcharnlikebpchar('455 '::BPCHAR(10), '455'::BPCHAR(10));
 bpcharnlikebpchar
-------------------
 f
(1 row)

gaussdb=#  SELECT bpcharnlikebpchar('455 '::BPCHAR(10), '455 '::BPCHAR);
 bpcharnlikebpchar
-------------------
 t
(1 row)
gaussdb=#  CREATE TABLE op_test (
 col BPCHAR(2) DEFAULT NULL
);
CREATE TABLE
gaussdb=#  CREATE INDEX op_index ON op_test(col);
CREATE INDEX
gaussdb=#  INSERT INTO op_test VALUES ('a');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('1');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('11');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('12');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('sd');
INSERT 0 1
gaussdb=#  INSERT INTO op_test VALUES ('aa');
INSERT 0 1
gaussdb=#  SHOW behavior_compat_options;
 behavior_compat_options
-------------------------
(1 row)
--当behavior_compat_options不包含enable_bpcharlikebpchar_compare时，则未启用最新
```


版权所有 © 华为云计算技术有限公司

```
bpcharnlikebpchar操作符，其BPCHAR匹配BPCHAR返回结果集非预期（正常应返回0条数据）。
gaussdb=#  SELECT * FROM op_test WHERE col NOT LIKE col::BPCHAR ORDER BY col;
 col
-----
 1
 a
(2 rows)

gaussdb=#  EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col NOT LIKE col::BPCHAR ORDER
BY col;
               QUERY PLAN
-------------------------------------------
 Streaming (type: GATHER)
   Merge Sort Key: col
   Node/s: All datanodes
   ->  Sort
         Sort Key: col
         ->  Seq Scan on op_test
               Filter: (col !~~ (col)::text)
(7 rows)
gaussdb=#  SET behavior_compat_options = 'enable_bpcharlikebpchar_compare';
SET
gaussdb=#  SHOW behavior_compat_options;
     behavior_compat_options
---------------------------------
 enable_bpcharlikebpchar_compare
(1 row)
--开启参数后，将启用最新bpcharnlikebpchar操作符，其匹配时返回行为符合预期。

gaussdb=#  SELECT * FROM op_test WHERE col NOT LIKE col::BPCHAR ORDER BY col;
 col
-----
(0 rows)

gaussdb=#  EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col NOT LIKE col::BPCHAR ORDER
BY col;
            QUERY PLAN
-------------------------------------
 Streaming (type: GATHER)
   Merge Sort Key: col
   Node/s: All datanodes
   ->  Sort
         Sort Key: col
         ->  Seq Scan on op_test
               Filter: (col !~~ col)
(7 rows)
gaussdb=#  DROP TABLE op_test;
DROP TABLE
```

说明

- 仅在数据库兼容性参数SQL_COMPATIBILITY为ORA时，设置GUC参数behavior_compat_options中含有enable_bpcharlikebpchar_compare字段，则可以使用bpcharlikebpchar的匹配。
- 启用该特性，会影响BPCHAR类型进行NOT LIKE模式匹配时的结果集与执行计划。
- GUC参数behavior_compat_options中不含有enable_bpcharlikebpchar_compare字段，表示关闭该特性；GUC参数behavior_compat_options中含有enable_bpcharlikebpchar_compare字段，表示开启该特性。
- 开启新特性后，定长匹配定长（bpchar匹配bpchar），需要指定左右两侧参数长度相同；模式匹配时，注意模式列长度与强制转换长度相同，避免过长后填补空格导致结果与预期存在差异。

版权所有 © 华为云计算技术有限公司

1.6.4 二进制字符串函数和操作符

字符串操作符

SQL定义了一些字符串函数，在这些函数里使用关键字而不是逗号来分隔参数。

- octet_length(string)
  描述：二进制字符串中的字节数。
  返回值类型：int
  示例：

```
gaussdb=# SELECT octet_length(E'jo\\000se'::bytea) AS RESULT;
 result
--------
      5
(1 row)
```

- overlay(string placing string from int [for int])
  描述：替换子串。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT overlay(E'Th\\000omas'::bytea placing E'\\002\\003'::bytea from 2 for 3) AS
RESULT;
     result
----------------
 \x5402036d6173
(1 row)
```

- position(substring in string)
  描述：特定子字符串的位置。
  返回值类型：int
  示例：

```
gaussdb=# SELECT position(E'\\000om'::bytea in E'Th\\000omas'::bytea) AS RESULT;
 result
--------
      3
(1 row)
```

- substring(string [from int] [for int])
  描述：截取子串。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT substring(E'Th\\000omas'::bytea from 2 for 3) AS RESULT;
  result
-----------
 \x68006f
(1 row)
```

- substr(bytea [from int] [for int])
  描述：截取子串。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT substr(E'Th\\000omas'::bytea,2, 3) AS RESULT;
  result
-----------
 \x68006f
(1 row)
```


版权所有 © 华为云计算技术有限公司

- trim([both] bytes from string)
  描述：从string的开头和结尾删除仅包含bytes中字节的最长字符串。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT trim(E'\\000'::bytea from E'\\000Tom\\000'::bytea) AS RESULT;
 result
----------
 \x546f6d
(1 row)
```

二进制字符串函数

GaussDB提供了函数调用所使用的常用语法。

- btrim(string bytea, bytes bytea)
  描述：从string的开头和结尾删除仅包含bytes中字节的最长的字符串。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT btrim(E'\\000trim\\000'::bytea, E'\\000'::bytea) AS RESULT;
  result
------------
 \x7472696d
(1 row)
```

- get_bit(string, offset)
  描述：从字符串中抽取位。
  返回值类型：int
  示例：

```
gaussdb=# SELECT get_bit(E'Th\\000omas'::bytea, 45) AS RESULT;
 result
--------
      1
(1 row)
```

  说明

  第一个参数仅支持bit或bytea类型。

- get_byte(string, offset)
  描述：从字符串中抽取字节。
  返回值类型：int
  示例：

```
gaussdb=# SELECT get_byte(E'Th\\000omas'::bytea, 4) AS RESULT;
 result
--------
    109
(1 row)
```

- set_bit(string, offset, newvalue)
  描述：设置字符串中的位。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT set_bit(E'Th\\000omas'::bytea, 45, 0) AS RESULT;
      result
------------------
```


版权所有 © 华为云计算技术有限公司

```
 \x5468006f6d4173
(1 row)
```

说明

第一个参数仅支持bit或bytea类型。

- set_byte(string, offset, newvalue)
  描述：设置字符串中的字节。
  返回值类型：bytea
  示例：

```
gaussdb=# SELECT set_byte(E'Th\\000omas'::bytea, 4, 64) AS RESULT;
     result
------------------
 \x5468006f406173
(1 row)
```

- rawcmp(raw, raw)
  描述：raw数据类型比较函数。
  返回值类型：integer
- raweq(raw, raw)
  描述：raw数据类型比较函数。
  返回值类型：boolean
- rawge(raw, raw)
  描述：raw数据类型比较函数。
  返回值类型：boolean
- rawgt(raw, raw)
  描述：raw数据类型比较函数。
  返回值类型：boolean
- rawin(cstring)
  描述：raw数据类型解析函数。
  返回值类型：bytea
- rawle(raw, raw)
  描述：raw数据类型解析函数。
  返回值类型：boolean
- rawlike(raw, raw)
  描述：raw数据类型解析函数。
  返回值类型：boolean
- rawlt(raw, raw)
  描述：raw数据类型解析函数。
  返回值类型：boolean
- rawne(raw, raw)
  描述：比较raw类型是否相同。
  返回值类型：boolean
- rawnlike(raw, raw)
  描述：比较raw类型与模式是否不匹配。

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- rawout(bytea)
  描述：raw类型的输出接口。
  返回值类型：cstring
- rawsend(raw)
  描述：转换bytea类型为二进制类型。
  返回值类型：bytea
- rawtohex(text)
  描述：raw格式转换为十六进制。
  返回值类型：text

1.6.5 位串函数和操作符

位串操作符

除了常用的比较操作符之外，还可以使用以下的操作符。&，|和#的位串操作数必须等长。在位移时，保留原始的位串长度（并以0填充）。

- ||
  描述：位串之间进行连接。
  示例：

```
gaussdb=# SELECT B'10001' || B'011' AS RESULT;
 result
----------
 10001011
(1 row)
```

  说明

  - 单字段内部连续连接操作不能超过180次，若超过需拆分为多个连续连接的字符串，它们之间再执行连接操作。
    例如：str1||str2||str3||str4 拆分为 (str1||str2)||(str3||str4)。
  - ORA兼容模式下位串中包含空字符串，会忽略空字符串连接其他字符串，其他兼容模式下则会返回空串。
    例如：str1||NULL||str2，ORA兼容模式下返回str1str2，其他兼容模式下返回NULL。

- &
  描述：位串之间进行“与”操作。
  示例：

```
gaussdb=# SELECT B'10001' & B'01101' AS RESULT;
 result
--------
 00001
(1 row)
```

- |
  描述：位串之间进行“或”操作。
  示例：

```
gaussdb=# SELECT B'10001' | B'01101' AS RESULT;
 result
--------
 11101
(1 row)
```


版权所有 © 华为云计算技术有限公司

- #

  描述：位串之间如果不一致进行“或”操作。如果两个位串中对应位置都为1或者0，则该位置返回为0。

  示例：

```
gaussdb=# SELECT B'10001' # B'01101' AS RESULT;
 result
--------
 11100
(1 row)
```

- ~

  描述：位串之间进行“非”操作。

  示例：

```
gaussdb=# SELECT ~B'10001'AS RESULT;
 result
----------
 01110
(1 row)
```

- <<

  描述：位串进行左移操作。

  示例：

```
gaussdb=# SELECT B'10001' << 3 AS RESULT;
 result
----------
 01000
(1 row)
```

- >>

  描述：位串进行右移操作。

  示例：

```
gaussdb=# SELECT B'10001' >> 2 AS RESULT;
 result
----------
 00100
(1 row)
```

下面的SQL标准函数除了可以用于字符串之外，也可以用于位串：length、bit_length、octet_length、position、substring、overlay。

下面的函数用于位串和二进制字符串：get_bit、set_bit。当用于位串时，这些函数位数从字符串的第一位（最左边）开始计数，其中第一位对应0位。

另外，可以在整数和bit之间来回转换。示例：

```
gaussdb=# SELECT 44::bit(10) AS RESULT;
   result
------------
 0000101100
(1 row)

gaussdb=# SELECT 44::bit(3) AS RESULT;
 result
--------
 100
(1 row)

gaussdb=# SELECT cast(-44 as bit(12)) AS RESULT;
    result
--------------
 111111010100
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT '1110'::bit(4)::integer AS RESULT;
 result
--------
     14
(1 row)

gaussdb=# SELECT substring('10101111'::bit(8), 2);
 substring
-----------
 0101111
(1 row)
```

说明

只是转换为“bit”的意思是转换成bit(1)，因此只会转换成整数的最低位。

1.6.6 模式匹配操作符

数据库提供了三种独立的实现模式匹配的方法：SQL LIKE操作符、SIMILAR TO操作符和POSIX-风格的正则表达式。除了这些基本的操作符外，还有一些函数可用于提取或替换匹配子串并在匹配位置分离一个串。

- LIKE

  描述：判断字符串是否能匹配上LIKE后的模式字符串。如果字符串与提供的模式匹配，则LIKE表达式返回为真（NOT LIKE表达式返回假），否则返回为假（NOT LIKE表达式返回真）。

  匹配规则：

  - 此操作符仅当模式匹配整个串时才能成功。若需匹配串内任意位置的字符序列，该模式必须以百分号开始和结束。
  - 下划线（_）代表（匹配）任何单个字符； 百分号（%）代表任意串的通配符。
  - 要匹配文本里的下划线或者百分号，在提供的模式里相应字符必须前导逃逸字符。逃逸字符的作用是禁用元字符的特殊含义，缺省的逃逸字符是反斜线，也可以用ESCAPE子句指定一个不同的逃逸字符。
  - 要匹配逃逸字符本身，需要输入两个逃逸字符。例如要写一个包含反斜线的模式常量，需要在SQL语句中输入两个反斜线。

    说明

    - 参数standard_conforming_strings设置为off时，在文串常量中的任何反斜线需进行双写处理。因此，要在模式中匹配单个反斜线，SQL语句中需输入四个反斜线。（可通过使用ESCAPE选择不同的逃逸字符来避免这种情况，此操作将使反斜线不再是LIKE的特殊字符。但仍然是字符文本分析器的特殊字符，所以需要两个反斜线。）
    - 在MYSQL模式下，可以通过写ESCAPE ''的方式不选择逃逸字符，此操作可以有效地禁用逃逸机制，但无法消除下划线和百分号在模式中的特殊含义。
    - 在ORA模式下，参数standard_conforming_strings设置为on时，可以设置behavior_compat_options为ignore_like_default_escape，如果未显式指定ESCAPE '\'，则禁用反斜线（\）的特殊含义，匹配单个反斜线（\）字符。
      - 该选项仅对关键字（LIKE、NOT LIKE、ILIKE、NOT ILIKE）生效，对操作符（~~、!~~、~~*、!~~*）不生效。
      - 当模式串包含 ANY、SOME 或 ALL 子句时，此选项将不生效。
  - 关键字ILIKE可以用于替换LIKE，区别是LIKE大小写敏感，ILIKE大小写不敏感。
  - 操作符~~等效于LIKE，操作符~~*等效于ILIKE。

 版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT 'abc' LIKE 'abc' AS RESULT;
 result
-----------
 t
(1 row)
gaussdb=# SELECT 'abc' LIKE 'a%' AS RESULT;
 result
-----------
 t
(1 row)
gaussdb=# SELECT 'abc' LIKE '_b_' AS RESULT;
 result
-----------
 t
(1 row)
gaussdb=# SELECT 'abc' LIKE 'c' AS RESULT;
 result
-----------
 f
(1 row)
```

- SIMILAR TO

  描述：SIMILAR TO操作符根据自己的模式是否匹配给定串而返回真或者假。该操作符与LIKE语法相似，但采用SQL标准定义的正则表达式进行模式匹配。

  匹配规则：

  - 和LIKE相同，此操作符仅当模式匹配整个串时才能成功。若需匹配串中任何位置的字符序列，模式需要以百分号作为起始符和终止符。
  - 下划线（_）代表（匹配）任何单个字符； 百分号（%）代表任意串的通配符。
  - SIMILAR TO也支持以下从POSIX正则表达式借用的模式匹配元字符。

元字符	含义
|	表示选择（两个候选之一）。
*	表示重复前面的项零次或更多次。
+	表示重复前面的项一次或更多次。
?	表示重复前面的项零次或一次。
{m}	表示重复前面的项刚好m次。
{m,}	表示重复前面的项m次或更多次。
{m,n}	表示重复前面的项至少m次并且不超过n次。
()	把多个项组合成一个逻辑项。
[...]	与POSIX正则表达式相同，表示声明一个字符类。

  - 前导逃逸字符可以禁用所有这些元字符的特殊含义。逃逸字符的使用规则和LIKE相同。

  正则表达式函数：

  支持使用函数**substring(string from pattern for escape)**截取匹配SQL正则表达式的子字符串。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT 'abc' SIMILAR TO 'abc' AS RESULT;
 result
------------
 t
(1 row)
gaussdb=# SELECT 'abc' SIMILAR TO 'a' AS RESULT;
 result
------------
 f
(1 row)
gaussdb=# SELECT 'abc' SIMILAR TO '%(b|d)%' AS RESULT;
 result
------------
 t
(1 row)
gaussdb=# SELECT 'abc' SIMILAR TO '(b|c)%'  AS RESULT;
 result
------------
 f
(1 row)
```

- POSIX正则表达式

  描述：正则表达式是一个字符序列，定义一个串集合（一个正则集）的缩写。 如果一个串是正则表达式描述的正则集中的一员时， 则说明这个串匹配该正则表达式。 POSIX正则表达式提供了比LIKE和SIMILAR TO操作符更强大的含义。**表1-47**列出了所有可用于POSIX正则表达式模式匹配的操作符。

  表 1-47 正则表达式匹配操作符

操作符	描述	例子
~	匹配正则表达式，大小写敏感	'thomas' ~ '.*thomas.*'
~*	匹配正则表达式，大小写不敏感	'thomas' ~* '.*Thomas.*'
!~	不匹配正则表达式，大小写敏感	'thomas' !~ '.*Thomas.*'
!~*	不匹配正则表达式，大小写不敏感	'thomas' !~* '.*vadim.*'

  匹配规则：

  - 与LIKE不同，正则表达式允许匹配串里的任何位置，除非该正则表达式显式地挂接在串的开头或者结尾。
  - 除了上文提到的元字符外， POSIX正则表达式还支持下列模式匹配元字符。

元字符	含义
^	表示串开头的匹配
$	表示串末尾的匹配
.	匹配任意单个字符

版权所有 © 华为云计算技术有限公司

正则表达式函数：POSIX正则表达式支持以下函数。

- **substring(string from pattern)**函数提供了抽取一个匹配POSIX正则表达式模式的子串的方法。
- **regexp_count(string text, pattern text [, position int [, flags text]])**函数提供了获取匹配POSIX正则表达式模式的子串数量的功能。
- **regexp_instr(string text, pattern text [, position int [, occurrence int [, return_opt int [, flags text]]]])**函数提供了获取匹配POSIX正则表达式模式子串位置的功能。
- **regexp_substr(string text, pattern text [, position int [, occurrence int [, flags text]]])**函数提供了抽取一个匹配POSIX正则表达式模式的子串的方法。
- **regexp_replace(string, pattern, replacement [,flags])**函数提供了将匹配POSIX正则表达式模式的子串替换为新文本的功能。
- **regexp_matches(string text, pattern text [, flags text])**函数返回一个文本数组，该数组由匹配一个POSIX正则表达式模式得到的所有被捕获子串构成。
- **regexp_split_to_table(string text, pattern text [, flags text])**函数把一个POSIX正则表达式模式当作一个定界符来分离一个串。
- **regexp_split_to_array(string text, pattern text [, flags text])**和regexp_split_to_table类似，是一个正则表达式分离函数，不过它的结果以一个text数组的形式返回。

  说明

  正则表达式分离函数会忽略零长度的匹配，这种匹配发生在串的开头或结尾或者正好发生在前一个匹配之后。这和正则表达式匹配的严格定义是相悖的，后者由regexp_matches实现，但是通常前者是实际中最常用的行为。

示例：

```
gaussdb=# SELECT 'abc' ~ 'Abc' AS RESULT;
result
--------
 f
(1 row)
gaussdb=# SELECT 'abc' ~* 'Abc' AS RESULT;
 result
--------
 t
(1 row)
gaussdb=# SELECT 'abc' !~ 'Abc' AS RESULT;
 result
--------
 t
(1 row)
gaussdb=# SELECT 'abc'!~* 'Abc' AS RESULT;
 result
--------
 f
(1 row)
gaussdb=# SELECT 'abc' ~ '^a' AS RESULT;
 result
--------
 t
(1 row)
gaussdb=# SELECT 'abc' ~ '(b|d)'AS RESULT;
 result
--------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT 'abc' ~ '^(b|c)'AS RESULT;
 result
--------
 f
(1 row)
```

虽然大部分的正则表达式搜索能够快速执行，但是仍可能被人为地处理成需要任意长的时间和任意量的内存。不建议从非安全模式来源接受正则表达式搜索模式，如果必须这样做，建议设置语句超时限制。使用SIMILAR TO模式的搜索具有同样的安全性危险， 因为SIMILAR TO提供了很多和POSIX-风格正则表达式相同的能力。LIKE搜索比其他两种选项简单得多，因此在接受非安全模式来源搜索时要更为安全。

- [NOT] REGEXP/ [NOT] RLIKE

  描述：REGEXP操作符用于正则表达式匹配，符合POSIX正则匹配规则。根据自己的模式是否匹配给定字符串而返回真或者假。

  下表描述了正则表达式操作符：

操作符名称	描述	语法说明
REGEXP	字符串是否与正则表达式匹配	expr REGEXP pat
RLIKE	字符串是否与正则表达式匹配（同REGEXP）	expr RLIKE pat
NOT REGEXP	字符串是否与正则表达式不匹配	expr NOT REGEXP pat
NOT RLIKE	字符串是否与正则表达式不匹配（同NOT REGEXP）	expr NOT RLIKE pat

  匹配规则：

  - 正则表达式允许匹配字符串中的任何位置，除非该正则表达式显式地拼接在字符串的起始位置或结尾位置。
  - REGEXP操作符支持的模式匹配元字符与POSIX正则表达式支持的模式匹配元字符一致。
  - 操作符支持的转义字符匹配包括：

转义字符	描述
\b	退格键。
\f	换页符，如C语言。
\n	换行符，如C语言。
\r	回车符，如C语言。
\t	水平制表符，如C语言。
\uwxyz	（其中wxyz正好是四个十六进制数字）十六进制值为0xwxyz的字符。

版权所有 © 华为云计算技术有限公司

转义字符	描述
\xhhh	（其中hhh是十六进制数字的任何序列）十六进制值为0xhhh的字符。
\xy	（其中xy正好是两个八进制数字）八进制值为0xy的字符。
\xyz	（其中xyz正好是三个八进制数字）八进制值为0xyz的字符。

- 模式字符串pat范围匹配：[a-dX]，[^a-dX]

  [a-dX]表示匹配任何a、b、c、d或X的字符。[^a-dX]表示匹配不是a、b、c、d或X的字符。

  两个字符之间的“-”构成一个范围，表示范围内所有字符匹配。要想包含文字字符“]”，该文字字符必须紧跟在第一个方括号“[”之后。要想包含文字字符“-”，它必须在第一个方括号“[”之后或在结束的方括号“]”之前。对于[]内未定义任何特殊含义的任何字符，与其本身匹配。

  示例：

  ```
  gaussdb=# SELECT 'abd' REGEXP 'a[bc]d' AS RESULT;
   result
  --------
   t
  (1 row)
  gaussdb=# SELECT 'aed' REGEXP 'a[^bc]d' AS RESULT;
   result
  --------
   t
  (1 row)
  gaussdb=# SELECT 'a-' REGEXP 'a[-b]' AS RESULT;
   result
  --------
   t
  (1 row)
  gaussdb=# SELECT 'aX]bc' REGEXP '^[]a-dXYZ]*$' AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- 模式字符串pat中[.characters.]匹配元素的字符序列。在括号表达式中（使用“[”和“]”），匹配用于校对元素的字符序列。字符为单个字符或诸如“space”等字符名。在文件regex/regc_locale中，可找到字符名称的完整列表。

  示例：

  ```
  gaussdb=# SELECT ' ' REGEXP '[[.space.]]' AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- 模式字符串pat中[=character_class=]字符匹配的字符类。在方括号表达式中编写，[=character_class=]代表等价类。它匹配具有相同排序校对值得所有字符，包括其本身。例如，如果o和(+)是等同类的成员，则[[=o=]]、[[=(+)=]]和[o(+)]是同义词。等同类不得用作范围的端点。
- 模式字符串pat中[:character_class:]字符匹配的字符类。在方括号表达式中编写（使用“[]”），[:character_class:]表示匹配属于该类的所有字符的字符类。特定区域可能会提供其他类名。字符类不得用作范围的端点。下表列出了标准类名。当涉及\字符时，需按说明配置相应的参数。

版权所有 © 华为云计算技术有限公司

字符类名	描述	字符范围
alnum	文字数字字符	[0-9a-zA-Z]
alpha	文字字符	[a-zA-Z]
blank	空白字符	空白字符[\t]
cntrl	控制字符	[\x01-\x1F]
digit	数字字符	[0-9]
graph	图形字符	[^\x01-\x20]
lower	小写文字字符	[a-z]
print	图形字符	[^\x01-\x20]
punct	标点字符	[-!"#$%&'( )*+,./:;<=>?@[\ \]^_`{|}~]
space	空格、制表符、新行、回车	[\n\r\t\x0B]
upper	大写文字字符	[A-Z]
xdigit	十六进制数字字符	[0-9a-fA-F]

示例：

```
gaussdb=# SELECT '\n' REGEXP '[[:space:]]' AS RESULT;
 result
--------
 t
(1 row)
```

- 模式字符串pat中[[:<:]], [[:>:]]匹配字符串的开始和结束位置。

  示例：

  ```
  gaussdb=# SELECT 'a word a' REGEXP '[[:<:]]word[[:>:]]' AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- 匹配特殊字符的文字实例，需要在特殊字符前加入两个反斜线，包括（、）、"、^、+、？特殊字符。特别地，对与单引号的匹配，需要在原字符串写'\''，与模式字符串中'\\\''匹配。当涉及\字符时，需按说明配置相应的参数。

  示例

  ```
  gaussdb=# SELECT 'a+b' REGEXP 'a\\+b' AS RESULT;
   result
  --------
   t
  (1 row)
  gaussdb=# SELECT '\'' REGEXP '\\\'' AS RESULT;
   result
  --------
   t
  (1 row)
  gaussdb=# SELECT '\\' REGEXP '\\\\' AS RESULT;
  ```


版权所有 © 华为云计算技术有限公司

```
 result
--------
 t
(1 row)
```

- 当设置了区分大小写的字符集字符序时，正则匹配也会区分大小写。

  示例：

```
gaussdb=# SELECT 'abc' REGEXP 'ABC' COLLATE utf8mb4_bin AS RESULT;
 result
--------
 f
(1 row)
```

- 入参Expr或pat为NULL时，返回空值。

```
gaussdb=# SELECT NULL REGEXP '*' AS RESULT;
 result
--------

(1 row)
gaussdb=# SELECT '-' REGEXP NULL AS RESULT;
 result
--------

(1 row)
```

说明

- 此操作符仅在MYSQL模式数据库下支持，且仅在sql_compatibility='MYSQL'有效，设置参数b_format_version='5.7'后，REGEXP操作符与~*操作符等效，NOT REGEXP操作符与!~*操作符等效。
- 当设置b_format_dev_version为s2时，默认standard_conforming_strings=off且escape_string_warning=off，此时在文串常量中输入的任何反斜线都需要被重复输入。因此，写一个匹配单个反斜线的模式，实际上要在语句中输入四个反斜线。
- 此操作符仅支持字符串类型、文本类型、以及可以隐式转换成字符串类型和文本类型的其他数据类型，不支持bytea类型。

1.6.7 数字操作函数和操作符

数字操作符

- +

  描述：加。

  示例：

```
gaussdb=# SELECT 2+3 AS RESULT;
 result
--------
      5
(1 row)
```

- -

  描述：减。

  示例：

```
gaussdb=# SELECT 2-3 AS RESULT;
 result
--------
     -1
(1 row)
```

- *

  描述：乘。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT 2*3 AS RESULT;
 result
--------
      6
(1 row)
```

- /

  描述：除（除法操作符不会取整）。

  示例：

```
gaussdb=# SELECT 4/2 AS RESULT;
 result
--------
      2
(1 row)
gaussdb=# SELECT 4/3 AS RESULT;
       result
------------------
 1.33333333333333
(1 row)
```

- +/-

  描述：正/负。

  示例：

```
gaussdb=# SELECT -2 AS RESULT;
 result
--------
     -2
(1 row)
```

- %

  描述：模（求余）。

  示例：

```
gaussdb=# SELECT 5%4 AS RESULT;
 result
--------
      1
(1 row)
```

- @

  描述：绝对值。

  示例：

```
gaussdb=# SELECT @ -5.0 AS RESULT;
 result
--------
    5.0
(1 row)
```

- ^

  描述：幂（指数运算）。

  示例：

```
gaussdb=# SELECT 2.0^3.0 AS RESULT;
       result
--------------------
 8.0000000000000000
(1 row)
```

- |/

  描述：平方根。

  示例：

```
gaussdb=# SELECT |/ 25.0 AS RESULT;
 result
```


版权所有 © 华为云计算技术有限公司

```
--------
     5
(1 row)
```

- ||/

  描述：立方根。

  示例：

  ```
  gaussdb=# SELECT ||/ 27.0 AS RESULT;
   result
  --------
       3
  (1 row)
  ```

- !

  描述：阶乘。

  示例：

  ```
  gaussdb=# SELECT 5! AS RESULT;
   result
  --------
     120
  (1 row)
  ```

- !!

  描述：阶乘（前缀操作符）。

  示例：

  ```
  gaussdb=# SELECT !!5 AS RESULT;
   result
  --------
     120
  (1 row)
  ```

- &

  描述：二进制AND。

  示例：

  ```
  gaussdb=# SELECT 91&15  AS RESULT;
   result
  --------
      11
  (1 row)
  ```

- |

  描述：二进制OR。

  示例：

  ```
  gaussdb=# SELECT 32|3  AS RESULT;
   result
  --------
      35
  (1 row)
  ```

- #

  描述：二进制XOR。

  示例：

  ```
  gaussdb=# SELECT 17#5  AS RESULT;
   result
  --------
      20
  (1 row)
  ```

- ~

  描述：二进制NOT。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT ~1 AS RESULT;
 result
--------
     -2
(1 row)
```

- <<

  描述：二进制左移。

  示例：

```
gaussdb=# SELECT 1<<4 AS RESULT;
 result
--------
     16
(1 row)
```

- >>

  描述：二进制右移。

  示例：

```
gaussdb=# SELECT 8>>2 AS RESULT;
 result
--------
      2
(1 row)
```

数字操作函数

- abs(x)

  描述：绝对值。

  返回值类型：和输入相同。

  示例：

```
gaussdb=# SELECT abs(-17.4);
 abs
------
 17.4
(1 row)
```

- acos(x)

  描述：反余弦。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT acos(-1);
       acos
------------------
 3.14159265358979
(1 row)
```

- asin(x)

  描述：反正弦。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT asin(0.5);
       asin
------------------
 .523598775598299
(1 row)
```


版权所有 © 华为云计算技术有限公司

- atan(x)
  描述：反正切。
  返回值类型：double precision
  示例：

```
gaussdb=# SELECT atan(1);
       atan
------------------
 .785398163397448
(1 row)
```

- atan2(y, x)
  描述：y/x的反正切。
  返回值类型：double precision
  示例：

```
gaussdb=# SELECT atan2(2, 1);
      atan2
------------------
 1.10714871779409
(1 row)
```

- bitand(integer, integer)
  描述：计算两个数字与运算(&)的结果。
  返回值类型：bigint类型数字。
  示例：

```
gaussdb=# SELECT bitand(127, 63);
 bitand
--------
     63
(1 row)
```

- cbrt(dp)
  描述：立方根。
  返回值类型：double precision
  示例：

```
gaussdb=# SELECT cbrt(27.0);
 cbrt
------
    3
(1 row)
```

- ceil(x)
  描述：不小于参数的最小的整数。
  返回值类型：整数。
  示例：

```
gaussdb=# SELECT ceil(-42.8);
 ceil
------
  -42
(1 row)
```

- ceiling(dp or numeric)
  描述：不小于参数的最小整数（ceil的别名）。
  返回值类型：与输入相同。
  示例：

```
gaussdb=# SELECT ceiling(-95.3);
 ceiling
```

 版权所有 © 华为云计算技术有限公司

```
---------
    -95
(1 row)
```

- cos(x)

  描述：余弦。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT cos(-3.1415927);
        cos
--------------------
 -.999999999999999
(1 row)
```

- cosh(x)

  描述：双曲余弦。

  返回值类型：dp or numeric，不考虑隐式类型转换的情况下与输入相同。

  示例：

```
gaussdb=# SELECT cosh(4);
        cosh
--------------------
 27.3082328360165
(1 row)
```

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- cot(x)

  描述：余切。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT cot(1);
       cot
-------------------
 .642092615934331
(1 row)
```

- degrees(dp)

  描述：把弧度转为角度。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT degrees(0.5);
     degrees
-------------------
 28.6478897565412
(1 row)
```

- div(y numeric, x numeric)

  描述：y除以x的商的整数部分。

  返回值类型：numeric

  示例：

```
gaussdb=# SELECT div(9,4);
 div
-----
   2
(1 row)
```


版权所有 © 华为云计算技术有限公司

- exp(x)

  描述：自然指数。

  返回值类型：与输入相同。

  示例：

```
gaussdb=# SELECT exp(1.0);
        exp
--------------------
 2.7182818284590452
(1 row)
```

- floor(x)

  描述：不大于参数的最大整数。

  返回值类型：与输入相同。

  示例：

```
gaussdb=# SELECT floor(-42.8);
 floor
-------
   -43
(1 row)
```

- int1(in)

  描述：将输入的text参数转换为int1类型值并返回。

  返回值类型：int1

  示例：

```
gaussdb=# SELECT int1('123');
 int1
------
 123
(1 row)
gaussdb=# SELECT int1('1.1');
 int1
------
   1
(1 row)
```

- int2(in)

  描述：将输入参数转换为int2类型值并返回。支持的入参类型包括：bigint、float4、float8、int16、integer、numeric、real、text。

  返回值类型：int2

  示例：

```
gaussdb=# SELECT int2('1234');
 int2
------
 1234
(1 row)
gaussdb=# SELECT int2(25.3);
 int2
------
   25
(1 row)
```

- int4(in)

  描述：将输入参数转换为int4类型值并返回。支持的入参类型包括：bit、boolean、char、double precision、int16、numeric、real、smallint、text。

  返回值类型：int4

  示例：

```
gaussdb=# SELECT int4('789');
 int4
```


版权所有 © 华为云计算技术有限公司

```
------
 789
(1 row)
gaussdb=# SELECT int4(99.9);
 int4
------
  100
(1 row)
```

- int8(in)

  描述：将输入参数转换为int8类型值并返回。支持的入参类型包括：bit、double precision、int16、integer、numeric、oid、real、smallint、text。

  返回值类型：int8

  示例：

```
gaussdb=# SELECT int8('789');
 int8
------
 789
(1 row)
gaussdb=# SELECT int8(99.9);
 int8
------
  99
(1 row)
```

- float4(in)

  描述：将输入参数转换为float4类型值并返回。支持的入参类型包括：bigint、double precision、int16、integer、numeric、smallint、text。

  返回值类型：float4

  示例：

```
gaussdb=# SELECT float4('789');
 float4
--------
   789
(1 row)

gaussdb=# SELECT float4(99.9);
 float4
--------
  99.9
(1 row)
```

- float8(in)

  描述：将输入参数转换为float8类型值并返回。支持的入参类型包括：bigint、int16、integer、numeric、real、smallint、text。

  返回值类型：float8

  示例：

```
gaussdb=# SELECT float8('789');
 float8
--------
   789
(1 row)

gaussdb=# SELECT float8(99.9);
 float8
--------
  99.9
(1 row)
```

- int16(in)

版权所有 © 华为云计算技术有限公司

描述：将输入参数转换为int16类型值并返回。支持的入参类型包括：bigint、boolean、double precision、 integer、numeric、oid、real、smallint、tinyint。

返回值类型：int16

示例：

```
gaussdb=# SELECT int16('789');
 int16
--------
    789
(1 row)

gaussdb=# SELECT int16(99.9);
 int16
--------
   100
(1 row)
```

- numeric(in)

  描述：将输入参数转换为numeric类型值并返回。支持的入参类型包括：bigint、boolean、double precision、int16、integer、money、real、smallint。

  返回值类型：numeric

  示例：

```
gaussdb=# SELECT "numeric"('789');
 numeric
---------
     789
(1 row)

gaussdb=# SELECT "numeric"(99.9);
 numeric
---------
    99.9
(1 row)
```

- oid(in)

  描述：将输入参数转换为oid类型值并返回。支持的入参类型包括：bigint、int16。

  返回值类型：oid

- radians(dp)

  描述：把角度转为弧度。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT radians(45.0);
     radians
------------------
 .785398163397448
(1 row)
```

- random()

  描述：0.0到1.0之间的随机数。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT random();
      random
------------------
 .824823560658842
(1 row)
```

 版权所有 © 华为云计算技术有限公司

说明

建议不要在安全场景下使用该函数。

- rand([seed])

  描述：可以无入参，也可以接受一个bigint类型的seed入参。返回0到1之间的随机数，如果指定seed，则返回随机数种子seed的随机值。

  参数：bigint类型，一个随机数种子。

  返回值类型：double

  示例：

```
b_compatible_db=# SELECT rand();
       rand
------------------
 .327476012520492
(1 row)

b_compatible_db=# SELECT rand(12321);
       rand
------------------
 .326073104515672
(1 row)
```

  说明

  - 该函数在数据库参数sql_compatibility = 'MYSQL'时生效。
  - 建议不要在安全场景下使用该函数。

- multiply(x double precision or text, y double precision or text)

  描述：x和y的乘积。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT multiply(9.0, '3.0');
    multiply
--------------------
              27
(1 row)
gaussdb=# SELECT multiply('9.0', 3.0);
    multiply
--------------------
              27
(1 row)
```

- ln(x)

  描述：自然对数。

  返回值类型：与输入相同。

  示例：

```
gaussdb=# SELECT ln(2.0);
        ln
--------------------
 .6931471805599453
(1 row)
```

- log(x)

  描述：以10为底的对数。

  返回值类型：与输入相同。

  示例：

```
gaussdb=# SELECT log(100.0);
       log
```


版权所有 © 华为云计算技术有限公司

```
--------------------
 2.0000000000000000
(1 row)
```

- log(b numeric, x numeric)

  描述：以b为底的对数。

  返回值类型：numeric

  示例：

```
gaussdb=# SELECT log(2.0, 64.0);
        log
--------------------
 6.0000000000000000
(1 row)
```

- log2(x)

  描述：以2为底的对数。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT log2(2);
 log2
------
    1
(1 row)
```

  说明

  该函数在数据库参数sql_compatibility = 'MYSQL'时生效。

- log10(x)

  描述：以10为底的对数。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT log10(10);
 log10
-------
     1
(1 row)
```

  说明

  该函数在数据库参数sql_compatibility = 'MYSQL'时生效。

- mod(x, y)

  描述：x/y的余数（模）如果x是0，则返回y。

  返回值类型：在入参为(text, int)类型时，返回NUMERIC类型，其余类型入参时，返回值类型与参数类型相同。

  示例：

```
gaussdb=# SELECT mod(9,4);
 mod
-----
   1
(1 row)
gaussdb=# SELECT mod(9,0);
 mod
-----
   9
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

此函数在ORA兼容模式数据库下且在参数a_format_version值为10c和a_format_dev_version值为s6的情况下，mod函数支持mod(text, int)重载，在以(text, int)为入参类型调用mod函数时，会优先选择mod(text, int)。

```
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s6';
SET
gaussdb=# SELECT mod('170141183460469231731687303715884105728', 7);
 mod
-----
   2
(1 row)
```

- pi()

  描述：“π”常量。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT pi();
        pi
-------------------
 3.14159265358979
(1 row)
```

- power(a double precision or numeric, b double precision or numeric)

  描述：a的b次幂。

  返回值类型：double precision or numeric。不考虑隐式类型转换的情况下，当两个入参均为numeric类型，返回numeric类型；如果其中任何一个入参为double precision，返回double precision类型。

  示例：

```
gaussdb=# SELECT power(9.0, 3.0);
        power
----------------------
 729.0000000000000000
(1 row)
```

- remainder(x, y)

  描述：x/y的余数，如果y是0，则返回报错信息。

  返回值类型：与输入相同（float4、float8或者numeric类型）。

  示例：

```
gaussdb=# SELECT remainder(11,4);
 remainder
-----------
        -1
(1 row)
gaussdb=# SELECT remainder(9,0);
ERROR: division by zero
CONTEXT:  referenced column: remainder
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- round(x)

  描述：离输入参数最近的整数。

  返回值类型：与输入相同（double precision或者numeric类型）。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT round(42.4);
 round
-------
    42
(1 row)

gaussdb=# SELECT round(42.6);
 round
-------
    43
(1 row)
```

说明

float/double类型的输出结果可能会出现-0（trunc、ceil等函数同样会出现此种情形。在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下结果为0），例如：

```
gaussdb=# SELECT round(-0.2::float8);
 round
-------
    -0
(1 row)
```

- round(v numeric, s int)

  描述：保留小数点后s位，s后一位进行四舍五入。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SELECT round(42.4382, 2);
   round
  -------
   42.44
  (1 row)
  ```

  说明

  - 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下，s截断为整数，否则s四舍五入为整数。
  - 在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，round函数支持round(timestamp, text)重载，在以(text, text)或(text, '')为入参调用round函数时会优先选择round(timestamp, text)。

- setseed(dp)

  描述：为随后的random()调用设置种子，种子取值范围为[-1.0, 1.0]。

  返回值类型：void

  示例：

  ```
  gaussdb=# SELECT setseed(0.54823);
   setseed
  ---------

  (1 row)
  ```

  说明

  建议不要在安全场景下使用该函数。

- sign(x)

  描述：输出此参数的符号。

  返回值类型：-1表示负数，0表示0，1表示正数。

  示例：

  ```
  gaussdb=# SELECT sign(-8.4);
   sign
  ```

 版权所有 © 华为云计算技术有限公司

```
------
  -1
(1 row)
```

- sin(x)

  描述：正弦。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT sin(1.57079);
       sin
------------------
 .999999999979986
(1 row)
```

- sinh(x)

  描述：双曲正弦。

  返回值类型：dp or numeric，不考虑隐式类型转换的情况下与输入相同。

  示例：

```
gaussdb=# SELECT sinh(4);
       sinh
------------------
 27.2899171971277
(1 row)
```

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- sqrt(x)

  描述：平方根。

  返回值类型：与输入相同。

  示例：

```
gaussdb=# SELECT sqrt(2.0);
       sqrt
-------------------
 1.414213562373095
(1 row)
```

  说明

  此函数在GUC参数gs_format_behavior_compat_options值为'sqrt_karatsuba'时会使用Karatsuba sqrt平方根算法计算，否则使用牛顿迭代算法计算。Karatsuba sqrt平方根算法的性能更快，在极少数场景下精度和牛顿迭代算法有差别。

- tan(x)

  描述：正切。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT tan(20);
       tan
------------------
 2.23716094422474
(1 row)
```

- tanh(x)

  描述：双曲正切。

  返回值类型：与输入相同（double precision或者numeric类型）。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT tanh(0.1);
        tanh
-------------------------------------------
0.099667994624955817171830508367835218353 89
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- trunc(x)

  描述：截断（取整数部分）。

  返回值类型：与输入相同。

  示例：

  ```
  gaussdb=# SELECT trunc(42.8);
   trunc
  -------
     42
  (1 row)
  ```

- trunc(v numeric, s int)

  描述：截断为s位小数。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SELECT trunc(42.4382, 2);
   trunc
  -------
   42.43
  (1 row)
  ```

  说明

  ORA兼容模式下，参数a_format_version值为10c和a_format_dev_version值为s1时，参数s若入参为小数则不会被四舍五入，而是被截断。

- width_bucket(op numeric, b1 numeric, b2 numeric, count int)

  描述：根据等间隔分桶原则，确定操作数op在由count个等宽区间构成的[b1,b2]范围内所属桶的序号。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT width_bucket(5.35, 0.024, 10.06, 5);
   width_bucket
  --------------
             3
  (1 row)
  ```

- width_bucket(op dp, b1 dp, b2 dp, count int)

  描述：根据等间隔分桶原则，确定操作数op在由count个等宽区间构成的[b1,b2]范围内所属桶的序号。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT width_bucket(5.35, 0.024, 10.06, 5);
   width_bucket
  --------------
             3
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- smgrne(a smgr, b smgr)
  描述：比较两个smgr类型整数是否不相等。
  返回值类型：boolean
- smgreq(a smgr, b smgr)
  描述：比较两个smgr类型整数是否相等。
  返回值类型：boolean
- int1abs(tinyint)
  描述：返回uint8类型数据的绝对值。
  返回值类型：tinyint
- int1and(tinyint, tinyint)
  描述：返回两个uint8类型数据按位与的结果。
  返回值类型：tinyint
- int1cmp(tinyint, tinyint)
  描述：返回两个uint8类型数据比较的结果，若第一个参数大，则返回1；若第二个参数大，则返回-1；若相等，则返回0。
  返回值类型：integer
- int1div(tinyint, tinyint)
  描述：返回两个uint8类型数据相除的结果，结果为tinyint类型。
  返回值类型：tinyint
- int1eq(tinyint, tinyint)
  描述：比较两个uint8类型数据是否相等。
  返回值类型：boolean
- int1ge(tinyint, tinyint)
  描述：判断两个uint8类型数据是否第一个参数大于等于第二个参数。
  返回值类型：boolean
- int1gt(tinyint, tinyint)
  描述：无符号一字节整数做大于运算。
  返回值类型：boolean
- int1larger(tinyint, tinyint)
  描述：返回无符号一字节整数中较大值。
  返回值类型：tinyint
- int1le(tinyint, tinyint)
  描述：无符号一字节整数做小于等于运算。
  返回值类型：boolean
- int1lt(tinyint, tinyint)
  描述：无符号一字节整数做小于运算。
  返回值类型：boolean
- int1smaller(tinyint, tinyint)
  描述：返回两个无符号一字节整数中较小的数。
  返回值类型：tinyint

版权所有 © 华为云计算技术有限公司

- int1inc(tinyint)
  描述：无符号一字节整数加一。
  返回值类型：tinyint
- int1mi(tinyint, tinyint)
  描述：无符号一字节整数做差运算。
  返回值类型：tinyint
- int1mod(tinyint, tinyint)
  描述：无符号一字节整数做取余运算。
  返回值类型：tinyint
- int1mul(tinyint, tinyint)
  描述：无符号一字节整数做乘法运算。
  返回值类型：tinyint
- int1ne(tinyint, tinyint)
  描述：无符号一字节整数做不等于运算。
  返回值类型：boolean
- int1pl(tinyint, tinyint)
  描述：无符号一字节整数做加法运算。
  返回值类型：tinyint
- int1um(tinyint)
  描述：无符号一字节数取相反数并返回有符号二字节整数。
  返回值类型：smallint
- int1xor(tinyint, tinyint)
  描述：无符号一字节整数进行异或运算。
  返回值类型：tinyint
- cash_div_int1(money, tinyint)
  描述：对money类型进行除法运算。
  返回值类型：money
- cash_mul_int1(money, tinyint)
  描述：对money类型进行乘法运算。
  返回值类型：money
- int1not(tinyint)
  描述：无符号一字节整数二进制位翻转。
  返回值类型：tinyint
- int1or(tinyint, tinyint)
  描述：无符号一字节整数进行或运算。
  返回值类型：tinyint
- int1shl(tinyint, integer)
  描述：无符号一字节整数左移指定位数。
  返回值类型：tinyint
- int1shr(tinyint, integer)

 版权所有 © 华为云计算技术有限公司

描述：无符号一字节整数右移指定位数。

返回值类型：tinyint

- analyze_tgtype_for_type(n smallint)

  描述：用于解析pg_trigger.tgtype，按位解析n，并返回before each row、 after each row、before statement、after statement、instead of中的一个。

  返回值类型：varchar2(16)

- analyze_tgtype_for_event(n smallint)

  描述：用于解析pg_trigger.tgtype，按位解析n，并返回insert、update、delete、truncate中的一个或多个。

  返回值类型：varchar2(246)

- nanvl(n2, n1)

  描述：输入两个参数，要求为数字类型或可以被隐式转化为数字类型的非数字类型；若第一个参数n2为NaN，返回n1，否则返回n2。

  返回值类型：入参中优先级最高的类型，优先级double precision>float4>numeric

  示例：

  ```
  gaussdb=# SELECT nanvl('NaN', 1.1);
   nanvl
  -------
    1.1
  (1 row)
  ```

  说明

  此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- numeric_eq_text(numeric, text)

  描述：判断numeric变量是否等于text变量转为numeric后的值。

  返回值类型：boolean

  示例：

  ```
  gaussdb=# SELECT numeric_eq_text(1, '1');
   numeric_eq_text
  -----------------
   t
  (1 row)
  ```

- numeric_ne_text(numeric, text)

  描述：判断numeric变量是否不等于text变量转为numeric后的值。

  返回值类型：boolean

- numeric_gt_text(numeric, text)

  描述：判断numeric变量是否大于text变量转为numeric后的值。

  返回值类型：boolean

- numeric_ge_text(numeric, text)

  描述：判断numeric变量是否大于等于text变量转为numeric后的值。

  返回值类型：boolean

- numeric_lt_text(numeric, text)

  描述：判断numeric变量是否小于text变量转为numeric后的值。

  返回值类型：boolean

版权所有 © 华为云计算技术有限公司

- numeric_le_text(numeric, text)
  描述：判断numeric变量是否小于等于text变量转为numeric后的值。
  返回值类型：boolean
- text_eq_numeric(text, numeric)
  描述：判断text变量转为numeric后的值是否等于numeric变量。
  返回值类型：boolean
  示例：

```
gaussdb=# SELECT text_eq_numeric('1', 1);
 text_eq_numeric
-----------------
 t
(1 row)
```

- text_ne_numeric(text, numeric)
  描述：判断text变量转为numeric后的值是否不等于numeric变量。
  返回值类型：boolean
- text_gt_numeric(text, numeric)
  描述：判断text变量转为numeric后的值是否大于numeric变量。
  返回值类型：boolean
- text_ge_numeric(text, numeric)
  描述：判断text变量转为numeric后的值是否大于等于numeric变量。
  返回值类型：boolean
- text_lt_numeric(text, numeric)
  描述：判断text变量转为numeric后的值是否小于numeric变量。
  返回值类型：boolean
- text_le_numeric(text, numeric)
  描述：判断text变量转为numeric后的值是否小于等于numeric变量。
  返回值类型：boolean
- bigint_eq_text(bigint, text)
  描述：判断bigint变量是否等于text变量转为bigint后的值。
  返回值类型：boolean

```
gaussdb=# SELECT bigint_eq_text(1, '1');
 bigint_eq_text
----------------
 t
(1 row)
```

- bigint_ne_text(bigint, text)
  描述：判断bigint变量是否不等于text变量转为bigint后的值。
  返回值类型：boolean
- bigint_gt_text(bigint, text)
  描述：判断bigint变量是否大于text变量转为bigint后的值。
  返回值类型：boolean
- bigint_ge_text(bigint, text)
  描述：判断bigint变量是否大于等于text变量转为bigint后的值。
  返回值类型：boolean

版权所有 © 华为云计算技术有限公司

- bigint_lt_text(bigint, text)
  描述：判断bigint变量是否小于text变量转为bigint后的值。
  返回值类型：boolean
- bigint_le_text(bigint, text)
  描述：判断bigint变量是否小于等于text变量转为bigint后的值。
  返回值类型：boolean
- text_eq_bigint(text, bigint)
  描述：判断text变量转为bigint后的值是否等于bigint变量。
  返回值类型：boolean
  示例：

```
gaussdb=# SELECT text_eq_bigint('1', 1);
 text_eq_bigint
----------------
 t
(1 row)
```

- text_ne_bigint(text, bigint)
  描述：判断text变量转为bigint后的值是否不等于bigint变量。
  返回值类型：boolean
- text_gt_bigint(text, bigint)
  描述：判断text变量转为bigint后的值是否大于bigint变量。
  返回值类型：boolean
- text_ge_bigint(text, bigint)
  描述：判断text变量转为bigint后的值是否大于等于bigint变量。
  返回值类型：boolean
- text_lt_bigint(text, bigint)
  描述：判断text变量转为bigint后的值是否小于bigint变量。
  返回值类型：boolean
- text_le_bigint(text, bigint)
  描述：判断text变量转为bigint后的值是否小于等于bigint变量。
  返回值类型：boolean

1.6.8 时间和日期处理函数和操作符

时间日期操作符

时间日期操作符如**表1-48**所示。

版权所有 © 华为云计算技术有限公司

说明

用户在使用时间和日期操作符时，应使用明确的类型前缀修饰对应的操作数，以确保数据库在解析操作数时能够与用户预期一致，避免产生非预期的结果。

举例：以下示例没有明确数据类型，就会出现异常错误。

```
gaussdb=# SELECT date '2001-10-01' - '7' AS RESULT;
ERROR:
GAUSS-10416: invalid input syntax for type timestamp: "7"
SQLSTATE: 22007
LINE 1: SELECT date '2001-10-01' - '7' AS RESULT;
                                   ^
CONTEXT:  referenced column: result
```

在ORA兼容模式下，DATEA类型支持与未明确类型的字符串（整数字符串或浮点数字符串）进行算术运算，运算结果的返回值类型为DATEA，仅支持+、-操作符。

```
gaussdb=# SET nls_date_format='yyyy-mm-dd hh24:mi:ss';
SET
gaussdb=# SELECT '2024-1-2'::DATEA + '0.00000579';
       ?column?
------------------------
 2024-01-02 00:00:01 AD
(1 row)

gaussdb=# SELECT '0.00000579' + '2024-1-2'::DATEA;
       ?column?
------------------------
 2024-01-02 00:00:01 AD
(1 row)

gaussdb=# SELECT '2024-1-2'::DATEA + '1';
       ?column?
------------------------
 2024-01-03 00:00:00 AD
(1 row)

gaussdb=# SELECT '2024-1-2'::DATEA - '0.00000579';
       ?column?
------------------------
 2024-01-01 23:59:59 AD
(1 row)

gaussdb=# SELECT '2' - '2024-1-2'::DATEA;
ERROR:  Invalid data exists in a string for matching date.
LINE 1: SELECT '2' - '2024-1-2'::DATEA;
               ^
```

在ORA兼容模式下，支持未明确类型（符合时间格式）减DATEA类型。

```
gaussdb=# SET nls_date_format='yyyy-mm-dd hh24:mi:ss';
SET
gaussdb=# SELECT '2024-1-3' - '2024-1-2'::DATEA;
 ?column?
----------
        1
(1 row)
```


版权所有 © 华为云计算技术有限公司

表 1-48 时间和日期操作符

操作符	示例
+	gaussdb=# SELECT date '2001-9-28' + integer '7' AS RESULT; result ------------ 2001-10-05 (1 row) **说明** 在ORA兼容模式下，查询结果为2001-10-05 00:00:00。
	gaussdb=# SELECT date '2001-09-28' + interval '1 hour' AS RESULT; result --------------------- 2001-09-28 01:00:00 (1 row)
	gaussdb=# SELECT date '2001-09-28' + time '03:00' AS RESULT; result --------------------- 2001-09-28 03:00:00 (1 row)
	gaussdb=# SELECT interval '1 day' + interval '1 hour' AS RESULT; result ---------------- 1 day 01:00:00 (1 row)
	gaussdb=# SELECT timestamp '2001-09-28 01:00' + interval '23 hours' AS RESULT; result --------------------- 2001-09-29 00:00:00 (1 row)
	gaussdb=# SELECT time '01:00' + interval '3 hours' AS RESULT; result ---------- 04:00:00 (1 row)
	gaussdb=# SET nls_date_format='yyyy-mm-dd hh24:mi:ss'; SET gaussdb=# SELECT '2024-1-2'::DATEA - '2024-1-1'::DATEA AS RESULT; result -------- 1 (1 row)
-	gaussdb=# SELECT date '2001-10-01' - date '2001-09-28' AS RESULT; result -------- 3 (1 row)
	gaussdb=# SELECT date '2001-10-01' - integer '7' AS RESULT; result --------------------- 2001-09-24 00:00:00 (1 row)
	gaussdb=# SELECT date '2001-09-28' - interval '1 hour' AS RESULT; result --------------------- 2001-09-27 23:00:00 (1 row)

版权所有 © 华为云计算技术有限公司

操作符	示例
	gaussdb=# SELECT time '05:00' - time '03:00' AS RESULT; result ---------- 02:00:00 (1 row)
	gaussdb=# SELECT time '05:00' - interval '2 hours' AS RESULT; result ---------- 03:00:00 (1 row)
	gaussdb=# SELECT timestamp '2001-09-28 23:00' - interval '23 hours' AS RESULT; result --------------------- 2001-09-28 00:00:00 (1 row)
	gaussdb=# SELECT interval '1 day' - interval '1 hour' AS RESULT; result ---------- 23:00:00 (1 row)
	gaussdb=# SELECT timestamp '2001-09-29 03:00' - timestamp '2001-09-27 12:00' AS RESULT; result ---------------- 1 day 15:00:00 (1 row)
	gaussdb=# SET nls_date_format='yyyy-mm-dd hh24:mi:ss'; SET gaussdb=# SELECT '2024-1-2'::DATEA - '2024-1-1'::DATEA AS RESULT; result -------- 1 (1 row) **说明** 在DATEA类型减DATEA类型的场景中，由于double类型精度问题，当前仅支持15位有效数字，且第15位数字可能因为第16位数字的精度问题而四舍五入错误。例如： SELECT '2020-10-02 00:00:00'::datea - '2020-10-01 00:00:34'::datea; 结果手动计算应该为：0.99960648148148148..... 但是因为double精度问题，实际计算结果会变成：0.999606481481482第16位出现了进位现象。
*	gaussdb=# SELECT 900 * interval '1 second' AS RESULT; result ---------- 00:15:00 (1 row)
	gaussdb=# SELECT 21 * interval '1 day' AS RESULT; result --------- 21 days (1 row)
	gaussdb=# SELECT double precision '3.5' * interval '1 hour' AS RESULT; result ---------- 03:30:00 (1 row)

版权所有 © 华为云计算技术有限公司

操作符	示例
/	gaussdb=# SELECT interval '1 hour' / double precision '1.5' AS RESULT; result ---------- 00:40:00 (1 row)

说明

在ORA兼容模式下支持：

- 左操作数类型DATEA加右操作数类型TIME WITHOUT TIME ZONE、INTERVAL DAY TO SECOND、INTERVAL DAY以及RELTIME类型，左右操作数可交换。
- 左操作数类型DATEA减右操作数类型TIME WITHOUT TIME ZONE、TIMESTAMP WITH TIME ZONE、INTERVAL DAY TO SECOND、INTERVAL DAY、RELTIME、SMALLDATETIME以及ABSTIME类型。
- 左操作数类型TIMESTAMP WITHOUT TIME ZONE、TIMESTAMP WITH TIME ZONE、SMALLDATETIME以及ABSTIME类型减右操作数类型DATEA。

时间/日期函数

- age(timestamp, timestamp)

  描述：将两个参数相减，并以年、月、日作为返回值。若相减值为负，则函数返回亦为负，入参可以都带timezone或都不带timezone。

  返回值类型：interval

  示例：

  ```
  gaussdb=# SELECT age(timestamp '2001-04-10', timestamp '1957-06-13');
            age
  --------------------------
   43 years 9 mons 27 days
  (1 row)
  ```

- age(timestamp)

  描述：当前SQL执行开始时刻的系统时间和参数相减，入参可以都带timezone或都不带timezone。

  返回值类型：interval

  示例：

  ```
  gaussdb=# SELECT age(timestamp '1957-06-13');
            age
  --------------------------
   60 years 2 mons 18 days
  (1 row)
  ```

- clock_timestamp()

  描述：返回当前函数被调用时系统时间的时间戳。该函数为volatile函数，每次扫描都会获取最新的时间戳，因此在一次查询中每次调用该函数的结果不相同。

  返回值类型：timestamp with time zone

  示例：

  ```
  gaussdb=# SELECT clock_timestamp();
         clock_timestamp
  -------------------------------
   2017-09-01 16:57:36.636205+08
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- current_date

  描述：返回当前SQL语句启动时的系统日期。

  返回值类型：date

  示例：

```
gaussdb=# SELECT current_date;
    date
------------
 2017-09-01
(1 row)
-- ORA兼容模式下，打开GUC参数a_format_date_timestamp时
gaussdb=# SET a_format_date_timestamp=on;
SET
gaussdb=# SELECT current_date;
    current_date
---------------------
 2023-11-24 11:25:09
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时，返回值为datea类型
gaussdb=# SELECT current_date;
      current_date
-------------------------
 2025-05-21 11:31:15 AD
(1 row)
```

  说明

  该函数在sql_compatibility = 'ORA'时，有以下表现：

  - GUC参数mapping_date_to_datea值为on的情况下，返回当前SQL语句启动的系统时间的时间戳，返回值类型为datea，列名为current_date。
  - GUC参数mapping_date_to_datea值为off的情况下遵循以下规则：
    - GUC参数a_format_date_timestamp值为on的情况下，返回当前SQL语句启动的系统时间的时间戳。返回值类型为timestamp without time zone，保留到秒，列名为current_date。
    - GUC参数a_format_date_timestamp值为off的情况下，返回事务开启时的系统时间。如果参数a_format_version值为10c且a_format_dev_version值为s2，返回值类型为timestamp without time zone，保留到秒，列名为timestamp。否则，返回值类型为date，列名为date，保留到天。

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有如下行为：

  - current_date支持带括号调用。
  - current_date实际调用的函数为curdate，可以用“\df curdate”指令查询函数详细入参和返回值。

- current_time()

  描述：当前事务开始时刻的系统时间。在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时，返回当前SQL语句执行开始时刻的系统时间。

  返回值类型：time with time zone；在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时，类型是time without time zone。

  示例：

```
gaussdb=# SELECT current_time;
      timetz
--------------------
 16:58:07.086215+08
(1 row)
--MYSQL模式下打开参数时
gaussdb_m=# SELECT current_time;
```


版权所有 © 华为云计算技术有限公司

```
 current_time
--------------
 15:14:00
(1 row)
```

- current_time([precision])

  描述：返回当前SQL语句执行开始时刻的系统时间。

  参数：表示精度（秒后小数位数），int类型，范围为[0,6]，缺省时默认值为0，无效精度值都会报错。

  返回值类型：time without time zone

  实现方式：映射成系统函数curtime。

  示例：

```
gaussdb_m=# SELECT current_time();
 current_time
--------------
 15:14:05
(1 row)
gaussdb_m=# SELECT current_time(3);
 current_time
--------------
 15:14:08.433
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。current_time实际调用的函数为curtime，可以用“\df curtime”指令查询函数详细入参和返回值。

- current_timestamp

  描述：当前日期及时间。语句级别时间，同一个语句内返回结果不变。

  返回值类型：MYSQL模式库中的5.7版本下（sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s1'）返回类型为timestamp without time zone，其他场景下返回类型为timestamp with time zone。

  示例：

```
gaussdb=# SELECT current_timestamp;
       pg_systimestamp
-------------------------------
 2017-09-01 16:58:19.22173+08
(1 row)

-- MYSQL模式库中的5.7版本下
gaussdb_m=# SELECT current_timestamp;
     timestamp
---------------------
 2023-08-21 15:08:24
(1 row)
-- ORA模式下,打开GUC参数a_format_date_timestamp时
gaussdb=# SET a_format_date_timestamp=on;
SET
gaussdb=# SELECT current_timestamp;
       current_timestamp
---------------------------------
 2023-11-24 11:31:04.895312+08
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

- 该函数在sql_compatibility = 'ORA'下，或者sql_compatibility = 'MYSQL'下且未设置b_format_version、b_format_dev_version参数时，返回结果为最近一条SQL执行开始时刻的系统时间的时间戳，返回值类型为timestamp with time zone，且GUC参数a_format_date_timestamp值为on的情况下，列名为current_timestamp。
- 在PL/SQL中，简单的赋值语句如：time1 := current_timestamp，被认为是表达式，所以会返回上一条SQL语句启动的系统时间的时间戳。
- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有如下行为：
  - 返回类型为timestamp without time zone。
  - 返回结果的精度为0。
  - 返回的结果为当前SQL语句启动时的系统时间的时间戳。
  - 该函数通过TYPE转化实现，无实际注册函数，因此通过gsql工具的\df+等相关命令查看时，显示的是其他兼容模式下的函数信息，而不是MYSQL兼容模式5.7版本中的函数信息。
  - 当在一个匿名块内多次调用该函数时，整个匿名块被视为一条语句，因此每次调用该函数都会返回相同的结果。

- current_timestamp()

  描述：返回结果为当前SQL语句启动时的系统时间。

  返回值类型：timestamp without time zone

  示例：

```
gaussdb=# SELECT current_timestamp();
     timestamp
---------------------
 2023-08-21 14:34:30
(1 row)
```

  说明

  该函数仅支持在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时使用，且有如下行为：

  - 返回类型为timestamp without time zone。
  - 返回结果的精度为0。
  - 返回的结果为当前SQL执行开始时刻的系统时间的时间戳。
  - 该函数通过TYPE转化实现，无实际注册函数，因此通过gsql工具的\df+等相关命令查看的为其他模式下的函数信息，而不是MYSQL兼容模式的5.7版本中的函数信息。

- current_timestamp(precision)

  描述：当前日期及时间，并将结果的微秒圆整为指定小数位。

  参数：表示精度（秒后小数位数），int类型，取值范围为[0,6]，缺省时默认值为0，如果超过6，则发出告警，并输出精度为6时的时间值；其他无效值将导致报错。

  返回值类型：MYSQL模式库中的5.7版本下（sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s1'）返回类型为timestamp without time zone，其他场景下返回类型为timestamp with time zone。

  示例：

```
gaussdb=# SELECT current_timestamp(1);
      timestamptz
-------------------------------
 2017-09-01 16:58:19.2+08
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
-- MYSQL模式库中的5.7版本下
gaussdb_m=# SELECT current_timestamp(1);
      timestamp
----------------------
 2023-08-21 15:09:35.3
(1 row)

-- ORA模式下打开GUC参数a_format_date_timestamp时
gaussdb=# SET a_format_date_timestamp=on;
SET
gaussdb=# SELECT current_timestamp(6);
      current_timestamp
-------------------------------
 2023-11-24 11:35:57.268592+08
(1 row)
-- 在ORA兼容模式数据库中且参数a_format_version值为10c、a_format_dev_version值为s2的情况下，
precision参数支持numeric类型的整值
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s2';
SET
gaussdb=# SELECT current_timestamp(6.0);
     current_timestamp
-------------------------------
 2023-11-25 11:14:48.74763+08
(1 row)
```

说明

- 微秒末位的0不显示，如2017-09-01 10:32:19.212000输出显示为2017-09-01 10:32:19.212。
- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有如下行为：
  - 返回类型为timestamp without time zone。
  - 返回的结果为当前SQL执行开始时刻的系统时间的时间戳。
  - 该函数通过TYPE转化实现，无实际注册函数，因此通过gsql工具的\df+等相关命令查看的为其他模式下的函数信息，而不是MYSQL兼容模式的5.7版本中的函数信息。
- 该函数在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为on的情况下有如下行为：
  - 返回值类型为timestamp with time zone，列名为current_timestamp。
  - 返回的结果为当前SQL执行开始时刻的系统时间的时间戳。
  - 当参数a_format_version值为10c、a_format_dev_version值为s2时，precision参数支持numeric类型的整值，否则仅支持int输入。
- 该函数在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为off的情况下，当入参为不带小数点的整值时，返回的结果为事务开启的系统的日期和时间；当入参为带小数点的整值时，返回结果为当前SQL执行开始时刻的系统时间的时间戳。

- pg_systimestamp()

  描述：当前日期和时间（当前语句的开始）。

  返回值类型：timestamp with time zone

  示例：

```
gaussdb=# SELECT pg_systimestamp();
       pg_systimestamp
-------------------------------
 2015-10-14 11:21:28.317367+08
(1 row)
```

- systimestamp()

  描述：获取当前系统日期和时间。

版权所有 © 华为云计算技术有限公司

返回值类型：timestamp with time zone

示例：

```
gaussdb=# SELECT systimestamp();
        systimestamp
-------------------------------
 2024-04-16 15:55:21.614342+08
(1 row)
-- behavior_compat_options参数设置sys_function_without_brackets后支持不带括号调用
gaussdb=# SET behavior_compat_options='sys_function_without_brackets';
gaussdb=# SELECT systimestamp;
        systimestamp
-------------------------------
 2024-04-16 15:55:21.614342+08
(1 row)
```

说明

该函数仅在sql_compatibility = 'ORA'时生效。

- systimestamp(precision)

  描述：获取当前系统日期及时间，并将结果的微秒部分按照指定precision进行显示。

  参数：precision表示精度（秒后小数位数）。INT类型，取值范围[0,6]，默认值为6，当超过6时提示WARNING，输出精度为6的时间值。

  返回值类型：timestamp with time zone

  示例：

```
gaussdb=# SELECT systimestamp(6);
        systimestamp
-------------------------------
 2024-04-16 15:55:21.614342+08
(1 row)
gaussdb=# SELECT systimestamp(9);
WARNING:  TIMESTAMP(9) WITH TIME ZONE precision reduced to maximum allowed, 6
CONTEXT:  referenced column: systimestamp
        systimestamp
-------------------------------
 2024-04-16 15:59:49.492505+08
(1 row)
gaussdb=# SELECT systimestamp(5.0);
        systimestamp
-------------------------------
 2024-04-16 16:00:23.48079+08
(1 row)
```

说明

该函数仅在sql_compatibility = 'ORA'时生效。

- date_part(text, timestamp)

  描述：获取日期或者时间值中子域的值，例如年或者小时的值。等效于extract(field from timestamp)。

  参数：timestamp类型为abstime、date、interval、reltime、time with time zone、time without time zone、timestamp with time zone、timestamp without time zone、datea（仅在ORA兼容性下生效）。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT date_part('hour', timestamp '2001-02-16 20:38:40');
 date_part
-----------
        20
(1 row)
```


版权所有 © 华为云计算技术有限公司

- date_part(text, interval)

  描述：获取月份的值。如果大于12，则取与12的模。等效于extract(field from timestamp)。

  返回值类型：double precision

  示例：

```
gaussdb=# SELECT date_part('month', interval '2 years 3 months');
 date_part
-----------
         3
(1 row)
```

- date_trunc(text, timestamp)

  描述：截取到参数text指定的精度。

  返回值类型：interval、timestamp with time zone、timestamp without time zone

  示例：

```
gaussdb=# SELECT date_trunc('hour', timestamp  '2001-02-16 20:38:40');
     date_trunc
---------------------
 2001-02-16 20:00:00
(1 row)
```

- trunc(timestamp)

  描述：默认按天截取。

  参数：timestamp类型为timestamp with time zone、timestamp without time zone、datea（仅在ORA兼容性下生效）。

  返回值类型： timestamp without time zone、datea（仅在ORA兼容性下生效）

  示例：

```
gaussdb=# SELECT trunc(timestamp  '2001-02-16
20:38:40');
                      trunc
---------------------
2001-02-16 00:00:00
(1 row)

gaussdb=# SELECT trunc(timestamp with time zone '2001-02-16 20:38:40');
        trunc
---------------------
 2001-02-16 00:00:00
(1 row)
```

- trunc(arg1, arg2)

  描述：截取到arg2指定的精度。

  参数：

  - arg1类型：interval、timestamp with time zone、timestamp without time zone、datea（仅在ORA兼容性下生效）
  - arg2类型：text

  arg1为datea类型时，arg2支持的单位列表如**表1-49**所示。

  arg1为其余支持类型时，当sql_compatibility = 'ORA' 且参数a_format_version值为10c、a_format_dev_version值为s1时，arg2支持的单位列表如**表1-49**所示，其他情况下，arg2支持的单位列表如**表1-50**所示。

版权所有 © 华为云计算技术有限公司

表 1-49 单位映射表 1

单位	含义
cc、scc	世纪
syyyy、yyyy、year、syear、yyy、yy、y	年
iyyy、iy、i	由ISO 8601标准定义的包含日历周的年份
q	季度
month、mon、mm、rm	月
ww	返回一个星期内与年份的第一天相同（星期相同）的那一天
iw	与ISO 8601标准定义的日历周的第一天相同的星期
w	返回一个星期内与月份第一天相同（星期相同）的那一天
ddd、dd、j	天
day、dy、d	一周的开始日期（以星期日作为每周的第一天）
hh、hh12、hh24	小时
mi	分钟

表 1-50 单位映射表 2

单位	含义
mil、mils、millennia、millennium	千禧年
c、cc、cent、centuries、century	世纪
dec、decade、decades、decs	年代
y、year、years、yr、yrs、yyyy	年
q、qtr、quarter	季度
mm、mon、mons、month、months	月
w、week、weeks	周
j、d、day、days、dd、ddd	天
h、hh、hour、hours、hr、hrs	小时
m、mi、min、mins、minute、minutes	分钟

版权所有 © 华为云计算技术有限公司

单位	含义
s、sec、second、seconds、secs	秒
millisecon、ms、msec、msecond、mseconds、msecs	毫秒
microsecon、us、usec、usecond、useconds、usecs	微秒

返回值类型：interval、timestamp with time zone、timestamp without time zone、datea（仅在ORA兼容性下生效）。

示例：

- arg1为datea类型

```
-- 2022年第一个周一为1月3日
gaussdb=# SELECT trunc(to_date('2022-12-30', 'YYYY-MM-
DD'),'iy');
                      trunc
-------------------------
 2022-01-03 00:00:00 AD
(1 row)

-- 2022年1月1日为周六，因此返回输入日期之前的第一个周六
gaussdb=# SELECT trunc(to_date('2022-12-30', 'YYYY-MM-DD'),'ww');
trunc
-------------------------
 2022-12-24 00:00:00 AD
(1 row)

-- 2022年12月30日这周，一周的开始日期（周一）为12月26日
gaussdb=# SELECT trunc(to_date('2022-12-30', 'YYYY-MM-DD'),'iw');
trunc
-------------------------
 2022-12-26 00:00:00 AD
(1 row)

-- 2022年12月1日为周四，因此返回输入日期前的第一个周四
gaussdb=# SELECT trunc(to_date('2022-12-30', 'YYYY-MM-DD'),'w');
trunc
-------------------------
 2022-12-29 00:00:00 AD
(1 row)

-- 2022年12月30这周，一周的开始日期（周日）为12月25日
gaussdb=# SELECT trunc(to_date('2022-12-30', 'YYYY-MM-DD'),'day');
trunc
-------------------------
 2022-12-25 00:00:00 AD
(1 row)
```

- arg1为其余支持类型

```
gaussdb=# SELECT trunc(timestamp  '2001-02-16 20:38:40',
'hour');
                      trunc
---------------------
2001-02-16 20:00:00
(1 row)
```

- round(arg1, arg2)

  描述：四舍五入到arg2指定的精度。

  参数：

版权所有 © 华为云计算技术有限公司

- arg1类型：timestamp without time zone、datea（仅在ORA兼容性下生效）
- arg2类型：text

返回值类型：timestamp without time zone、datea（仅在ORA兼容性下生效）

示例：

```
gaussdb=# SELECT round(timestamp  '2001-02-16 20:38:40',
'hour');
                round
---------------------
2001-02-16 21:00:00
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时
gaussdb=# SET nls_date_format = 'YYYY-MM-DD HH24:MI:SS';
gaussdb=# SELECT round(datea '2001-02-16 20:38:40', 'yyyy');
        round
------------------------
 2001-01-01 00:00:00 AD
(1 row)
```

说明

- 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。
- 当arg1入参类型为datea时，arg2支持指定的单位参考trunc函数，见**表1-49**。

- daterange(arg1, arg2)

  描述：获取时间边界信息。

  参数：arg1和arg2的类型均为date。

  返回值类型：daterange

  示例：

```
gaussdb=# SELECT daterange('2000-05-06','2000-08-08');
       daterange
-------------------------
 [2000-05-06,2000-08-08)
(1 row)
```

  说明

  在ORA兼容模式下：

  - 开启mapping_date_to_datea参数后，DATE类型会映射为DATEA类型，daterange函数不支持DATEA类型作为入参。
  - 关闭mapping_date_to_datea参数后，DATE类型会映射为TIMESTAMP(0) WITHOUT TIME ZONE类型，daterange函数不支持TIMESTAMP(0) WITHOUT TIME ZONE类型作为入参。

- daterange(arg1, arg2, text)

  描述：获取时间边界信息。

  参数：

  - arg1为时间范围的起始日期，类型为date。
  - arg2为时间范围的结束日期，类型为date。
  - text为边界符号字符串，类型为text，必须是字符串 “()”、“(]”、“[)”或 “[]” 之一，用于指定范围两端是否包含边界值。其中“[”代表包含起始日期，“(”代表不包含起始日期，“]”代表包含结束日期，“)”代表不包含结束日期。

  返回值类型：daterange

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT daterange('2000-05-06','2000-08-08','[]');
       daterange
-------------------------
 [2000-05-06,2000-08-09)
(1 row)
```

说明

在ORA兼容模式下：

- 开启mapping_date_to_datea参数后，DATE类型会映射为DATEA类型，daterange函数不支持DATEA类型作为入参。
- 关闭mapping_date_to_datea参数后，DATE类型会映射为TIMESTAMP(0) WITHOUT TIME ZONE类型，daterange函数不支持TIMESTAMP(0) WITHOUT TIME ZONE类型作为入参。

- isfinite(date)

  描述：判断日期是否为有限值，是则返回t，否则返回f。

  返回值类型：Boolean

  示例：

  ```
  gaussdb=# SELECT isfinite(date '2001-02-16');
   isfinite
  ----------
   t
  (1 row)
  gaussdb=# SELECT isfinite(date 'infinity');
   isfinite
  ----------
   f
  (1 row)
  ```

- isfinite(timestamp)

  描述：判断时间戳是否为有限值，是则返回t，否则返回f。

  返回值类型：Boolean

  示例：

  ```
  gaussdb=# SELECT isfinite(timestamp '2001-02-16 21:28:30');
   isfinite
  ----------
   t
  (1 row)
  gaussdb=# SELECT isfinite(timestamp 'infinity');
   isfinite
  ----------
   f
  (1 row)
  ```

- isfinite(interval)

  描述：判断时间间隔是否为有限值，是则返回t，暂不支持返回f，输入'infinity'会报错。

  返回值类型：Boolean

  示例：

  ```
  gaussdb=# SELECT isfinite(interval '4 hours');
   isfinite
  ----------
   t
  (1 row)
  ```

- justify_days(interval)

  描述：将时间间隔调整为以月（30天为一月）为单位。

版权所有 © 华为云计算技术有限公司

返回值类型：interval

示例：

```
gaussdb=# SELECT justify_days(interval '35 days');
 justify_days
--------------
 1 mon 5 days
(1 row)
```

- justify_hours(interval)

  描述：将时间间隔调整为以天（24小时为一天）为单位。

  返回值类型：interval

  示例：

  ```
  gaussdb=# SELECT justify_hours(interval '27 hours');
   justify_hours
  ---------------
   1 day 03:00:00
  (1 row)
  ```

- justify_interval(interval)

  描述：结合justify_days和justify_hours，调整interval。

  返回值类型：interval

  示例：

  ```
  gaussdb=# SELECT justify_interval(interval '1 mon -1 hour');
   justify_interval
  ------------------
   29 days 23:00:00
  (1 row)
  ```

- localtime()

  描述：当前事务开始时刻的系统时间。在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时返回当前SQL执行开始时刻的系统日期和时间。

  返回值类型：time，在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时返回类型timestamp without time zone。

  示例：

  ```
  gaussdb=# SELECT localtime AS RESULT;
      result
  ----------------
   16:05:55.664681
  (1 row)
  -- MYSQL模式，打开兼容性参数后
  gaussdb_m=# select localtime;
       localtime
  ---------------------
   2023-08-21 15:21:57
  (1 row)
  ```

- localtime([precision])

  描述：返回当前SQL执行开始时刻的系统日期和时间。

  参数：表示精度（秒后小数位数），int类型，范围为[0,6]，缺省时默认值为0，无效值都报错。

  返回值类型：timestamp without time zone

  实现方式：注册系统函数localtime。

  示例：

  ```
  gaussdb_m=# SELECT localtime();
       localtime
  ```


版权所有 © 华为云计算技术有限公司

```
---------------------
 2023-08-21 15:23:49
(1 row)
gaussdb_m=# SELECT localtime(3);
       localtime
-------------------------
 2023-08-21 15:23:51.965
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- localtimestamp

  描述：当前日期及时间。

  返回值类型：timestamp

  示例：

```
gaussdb=# SELECT localtimestamp;
        timestamp
----------------------------
 2017-09-01 17:03:30.781902
(1 row)
-- MYSQL兼容模式下，打开兼容性参数
gaussdb_m=# SELECT localtimestamp;
     timestamp
---------------------
 2023-08-21 15:27:55
(1 row)
-- ORA兼容模式下打开GUC参数a_format_date_timestamp时
gaussdb=# SET a_format_date_timestamp=on;
SET
gaussdb=# SELECT localtimestamp;
      localtimestamp
----------------------------
 2023-11-24 11:38:25.633231
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有如下行为：
  - 返回当前SQL执行开始时刻的系统日期和时间。
  - 返回类型timestamp without time zone，列名为timestamp。
- 该函数在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为on的情况下有如下行为：
  - 返回值类型为timestamp without time zone，列名为localtimestamp。
  - 返回的结果为当前SQL执行开始时刻的系统时间的时间戳。
  - 防止优化器提前获取常数时间，造成gplan场景下时间获取不正确。
- 该函数在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为off的情况下，返回的结果为事务开启时的系统的日期和时间。

- localtimestamp[([precision])]

  描述：当前日期及时间。

  参数：表示精度（秒后小数位数），int类型，取值范围为[0,6]，缺省时默认值为0，如果超过6，则发出告警，并输出精度为6时的时间值；其他无效值将导致报错。

  返回值类型：timestamp without time zone

  示例：

版权所有 © 华为云计算技术有限公司

```
-- 带括号且无入参调用仅在MYSQL兼容模式下支持
gaussdb_m=# SELECT localtimestamp();
     timestamp
---------------------
 2023-08-21 15:27:59
(1 row)
gaussdb_m=# SELECT localtimestamp(3);
       timestamp
-------------------------
 2023-08-21 15:28:02.445
(1 row)
-- ORA兼容模式下，打开GUC参数a_format_date_timestamp时
gaussdb=# SET a_format_date_timestamp=on;
SET
gaussdb=# SELECT localtimestamp(6);
      localtimestamp
----------------------------
 2023-11-24 11:41:14.086227
(1 row)
-- 在ORA兼容模式数据库中且参数a_format_version值为10c、a_format_dev_version值为s2的情况下，
precision参数支持numeric类型的整值
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s2';
SET
gaussdb=# SELECT localtimestamp(6.0);
      localtimestamp
----------------------------
 2023-11-24 11:42:45.642167
(1 row)
```

说明

- 微秒末位的0不显示，如2017-09-01 10:32:19.212000输出显示为2017-09-01 10:32:19.212。
- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时，返回当前SQL执行开始时刻的系统日期和时间，支持带括号且无入参调用。
- 该函数在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为on的情况下，有以下行为：
  - 返回当前SQL执行开始时刻的系统时间的时间戳。
  - 返回值类型为timestamp without time zone，列名为localtimestamp。
  - 当参数a_format_version值为10c、a_format_dev_version值为s2时，precision参数支持numeric类型的整值，否则仅支持int输入。
- 该函数在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为off的情况下，返回的结果为事务开启时的系统的日期和时间。

- maketime(hour, minute, second)

  描述：通过入参hour、minute、second生成对应time类型时间，其中三个入参类型分别为bigint、bigint和numeric。

  返回值类型：time

  示例：

```
gaussdb=# SELECT maketime(8, 15, 26.53);
  maketime
-------------
 08:15:26.53
(1 row)

gaussdb=# SELECT maketime(-888, 15, 26.53);
  maketime
------------
 -838:59:59
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

该函数仅支持sql_compatibility = 'MYSQL'，GUC参数b_format_version='5.7'、b_format_dev_version='s1'时使用，且有如下行为：

- 当参数满足如下任一条件时，函数返回NULL。
  - 入参minute<0或minute>=60。
  - 入参second<0或second>=60。
  - 任一参数为NULL。
- 返回的time类型结果保留6位小数，若second超出六位小数，则按照四舍五入进位。
- 返回time类型值要求在[-838:59:59, 838:59:59]中。若超出范围，则根据hour的正负来返回指定的边界值。
- maketime不支持自嵌套。

- now()

  描述：当前事务的开始时刻的系统的日期及时间，同一个事务内返回结果相同。仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时返回当前SQL执行开始时刻的系统日期和时间。

  返回值类型：timestamp with time zone，在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时返回类型timestamp without time zone。

  示例：

```
gaussdb=# SELECT now();
              now
-------------------------------
 2017-09-01 17:03:42.549426+08
(1 row)
--MYSQL模式下，打开兼容性参数
gaussdb_m=# SELECT now();
      timestamp
---------------------
 2023-08-21 17:17:42
(1 row)
```

- now(precision)

  描述：返回当前SQL执行开始时刻的系统日期和时间。

  参数：表示精度（秒后小数位数），int类型，取值范围为[0,6]，缺省时默认值为0，如果超过6，则发出告警，并输出精度为6时的时间值；其他无效值将导致报错。

  返回值类型：timestamp without time zone

  实现方式：注册系统函数now_b。

  示例：

```
gaussdb_m=# SELECT now(3);
       timestamp
-------------------------
 2023-08-21 17:17:48.819
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- now_b(precision)

  描述：返回当前SQL语句执行开始时刻的系统日期和时间，等价于now函数，此函数为内部实现函数，不建议使用。

版权所有 © 华为云计算技术有限公司

参数：precision表示精度（秒后小数位数），int类型，取值范围为[0,6]，如果超过6，则发出告警，并输出精度为6时的时间值。

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- timenow()

  描述：返回当前SQL执行开始时刻的系统日期和时间。

  返回值类型：abstime

  示例：

```
gaussdb=# SELECT timenow();
       timenow
------------------------
 2020-06-23 20:36:56+08
(1 row)
```

- dbtimezone

  描述：当前数据库的时区。

  返回值类型：text

  示例：

```
gaussdb=# SELECT dbtimezone;
       dbtimezone
------------------------
 PRC
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- numtodsinterval(num, interval_unit)

  描述：将数字转换为interval类型。num为numeric类型数字，interval_unit为固定格式字符串（'DAY' | 'HOUR' | 'MINUTE' | 'SECOND'）。

  可以通过设置GUC参数intervalstyle为oracle，兼容该函数interval输出格式。

  返回值类型：interval

  示例：

```
gaussdb=# SELECT numtodsinterval(100, 'HOUR');
 numtodsinterval
-----------------
 100:00:00
(1 row)

gaussdb=# SET intervalstyle = oracle;
SET
gaussdb=# SELECT numtodsinterval(100, 'HOUR');
       numtodsinterval
-------------------------------
 +000000004 04:00:00.000000000
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下：当参数interval_unit为 'DAY' 时，参数num超过1000000000会报错。

- numtoyminterval(num, interval_unit)

版权所有 © 华为云计算技术有限公司

描述：将数字转换为interval类型。num为numeric类型数字，interval_unit为固定格式字符串（'YEAR' | 'MONTH'）。

可以通过设置GUC参数intervalstyle为oracle，兼容该函数interval输出格式。

返回值类型：interval

示例：

```
gaussdb=# \c gaussdb_o;
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "gaussdb_o" as user "omm".
gaussdb_o=# SET a_format_version='10c';
SET
gaussdb_o=# SET a_format_dev_version='s2';
SET
gaussdb=# SELECT numtoyminterval(100, 'MONTH');
 numtoyminterval
-----------------
 8 years 4 mons
(1 row)
gaussdb_o=# SET intervalstyle = oracle;
SET
gaussdb_o=# SELECT numtoyminterval(100, 'MONTH');
 numtoyminterval
-----------------
 8-4
(1 row)

gaussdb_o=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "omm".
gaussdb=# DROP DATABASE gaussdb_o;
DROP DATABASE
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下有效。

- new_time(date, timezone1,timezone2)

  描述：当timezone1所表示时区的日期时间为date的时候，返回此时timezone2所表示时区的日期时间值。

  返回值类型：timestamp、datea（仅在ORA兼容性下生效）

  示例：

```
gaussdb=# SELECT new_time('1997-10-10','AST','EST');
      new_time
---------------------
 1997-10-09 23:00:00
(1 row)
gaussdb=# SELECT new_time(to_timestamp ('10-sep-02 14:10:10.123000','dd-mon-rr hh24:mi:ss.ff'),
'ast', 'pst');
        new_time
-------------------------
 2002-09-10 10:10:10.123
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时
gaussdb=# SET nls_date_format = 'YYYY-MM-DD HH24:MI:SS';
gaussdb=# SELECT new_time(datea '2001-02-16 20:38:40', 'AST', 'EST');
        new_time
------------------------
 2001-02-16 19:38:40 AD
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

- 该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型，此时返回值为datea类型。
- 该函数在ORA兼容模式数据库中、参数a_format_version值为10c、a_format_dev_version值为s2，且GUC参数mapping_date_to_datea值为off的情况下，数据库会将入参中的DATE类型解析为TIMESTAMP(0) WITHOUT TIME ZONE类型，此时返回值为TIMESTAMP类型。

- sessiontimezone()

  描述：当前会话的时区，无入参。

  返回值类型：text

  示例：

```
gaussdb=# SELECT sessiontimezone;
 session_time_zone
-------------------
 PST8PDT
(1 row)
gaussdb=# SELECT lower(sessiontimezone);
   lower
-----------
 @ 8 hours
(1 row)
```

说明

- 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下生效。
- 当set session time zone的值为GMT+08:00/GMT-08:00格式时，正值的偏移量被用于格林威治以西的位置，例如GMT+08:00表示西八区，GMT-08:00表示东八区。

- sys_extract_utc(timestamp| timestamptz)

  描述：从具有时区偏移量或时区区域名称的日期时间值中提取UTC（协调世界时-以前称为格林威治平均时间）。如果未指定时区，则日期时间与会话时区关联。入参有timestamp和timestamptz两种形式。

  返回值类型：timestamp

  示例：

```
gaussdb=# SELECT sys_extract_utc(timestamp '2000-03-28 11:30:00.00');
   sys_extract_utc
---------------------
 2000-03-28 03:30:00
(1 row)
gaussdb=# SELECT sys_extract_utc(timestamptz '2000-03-28 11:30:00.00 -08:00');
   sys_extract_utc
---------------------
 2000-03-28 19:30:00
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下生效。

- tz_offset('time_zone_name' | '(+/-)hh:mi' | SESSIONTIMEZONE | DBTIMEZONE)

  描述：入参有以上四种形式，返回入参所表示时区的UTC偏移量。

  返回值类型：text

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT tz_offset('us/pacific');
 tz_offset
-----------
 -08:00
(1 row)
gaussdb=# SELECT tz_offset(sessiontimezone);
 tz_offset
-----------
 +08:00
(1 row)
```

说明

此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s2的情况下生效。

- pg_sleep(seconds)

  描述：服务器线程延迟时间，单位为秒。当数据库调用该函数时，会获取相应的事务快照，相当于一个长事务，如果入参时间过长可能导致数据库oldestxmin无法推进，影响表的回收和查询性能。

  返回值类型：void

  示例：

```
gaussdb=# SELECT pg_sleep(10);
 pg_sleep
----------


(1 row)
```

- statement_timestamp()

  描述：返回当前日期和时间（当前语句的开始）。

  返回值类型：timestamp with time zone

  示例：

```
gaussdb=# SELECT statement_timestamp();
     statement_timestamp
-------------------------------
 2017-09-01 17:04:39.119267+08
(1 row)
```

- sysdate

  描述：返回当前SQL执行时刻的系统日期和时间。

  返回值类型：timestamp、datea（仅在ORA兼容性下生效）

  示例：

```
gaussdb=# SELECT sysdate;
       sysdate
---------------------
 2017-09-01 17:04:49
(1 row)
--开启fixed_date时返回的结果。
gaussdb=# SET fixed_date='2024-05-15 11:47:02';
SET
gaussdb=# SELECT sysdate;
       timestamp
---------------------
 2024-05-15 11:47:02
(1 row)
--不支持负数时间设置，若需要表示为公元前时间，可在时间后加上'BC'。
gaussdb=# SET fixed_date = '2012-01-01 00:00:00 BC';
SET
gaussdb=# SELECT sysdate;
       timestamp
------------------------
 2012-01-01 00:00:00 BC
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时，返回值为datea类型。
gaussdb=# SELECT sysdate;
      sysdate_a
------------------------
 2025-05-21 11:37:26 AD
(1 row)
```

说明

该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，会映射为sysdate_a函数，返回值将变更为datea类型。

- sysdate([precision])

  描述：返回函数执行时刻的系统日期和时间。

  参数：表示时间精度，int类型，范围为[0,6]，缺省时默认值为0。

  返回值类型：timestamp without time zone

  示例：

```
gaussdb_m=# SELECT sysdate();
     sysdate()
---------------------
 2023-08-21 17:17:42
(1 row)
gaussdb_m=# SELECT sysdate(3);
       sysdate(3)
-------------------------
 2023-08-21 17:17:48.819
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- current_sysdate()

  描述：返回当前SQL执行开始时刻的系统日期和时间。

  返回值类型：timestamp

  示例：

```
gaussdb=# SELECT current_sysdate();
   current_sysdate
---------------------
 2023-06-20 20:09:02
(1 row)
```

- timeofday()

  描述：返回当前函数被调用时的系统时间的时间戳（像clock_timestamp，但是返回时为text）。

  返回值类型：text

  示例：

```
gaussdb=# SELECT timeofday();
              timeofday
--------------------------------------
 Fri Sep 01 17:05:01.167506 2017 CST
(1 row)
```

- transaction_timestamp()

  描述：当前事务开始的系统的日期及时间。

  返回值类型：timestamp with time zone，在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时返回timestamp without time zone。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT transaction_timestamp();
    transaction_timestamp
-------------------------------
 2017-09-01 17:05:13.534454+08
(1 row)
--MYSQL模式打开兼容性开关
gaussdb=#  SELECT transaction_timestamp();
  transaction_timestamp
----------------------------
 2023-09-07 09:32:09.728998
(1 row)
```

- add_months(d,n)

  描述：用于计算时间点d再加上n个月的时间。

  参数：

  - d：timestamp类型的值，以及可以隐式转换为timestamp类型的值。
  - n：INTEGER类型的值，以及可以隐式转换为INTEGER类型的值。

  返回值类型：timestamp、datea（仅在ORA兼容性下生效）

  示例：

  ```
  gaussdb=# SELECT add_months(to_date('2017-5-29', 'yyyy-mm-dd'), 11) FROM sys_dummy;
      add_months
  ---------------------
   2018-04-29 00:00:00
  (1 row)
  -- ORA兼容模式下，打开GUC参数mapping_date_to_datea时
  gaussdb=# SET nls_date_format = 'YYYY-MM-DD HH24:MI:SS';
  gaussdb=# SELECT add_months(datea '2020-11-12 11:12:13', 3);
        add_months
  ------------------------
   2021-02-12 11:12:13 AD
  (1 row)
  ```

  说明

  - 该函数在sql_compatibility='ORA'时，且GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型，此时返回值为datea类型。
  - 该函数在GUC参数mapping_date_to_datea值为off的情况下，数据库会将入参中的DATE类型解析为TIMESTAMP(0) WITHOUT TIME ZONE类型，此时返回值为TIMESTAMP类型。当该函数入参类型被解析为TIMESTAMP(0) WITHOUT TIME ZONE时，函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下，存在以下行为：
    - 当返回结果大于公元9999年时，会产生报错。
    - 当参数n入参为小数时，函数不会对小数位进行四舍五入操作，而是直接被截断。
  - 在sql_compatibility='ORA'时，如果参数为datea类型（如：add_months(to_datea('2025-2-28', 'yyyy-mm-dd'),1)场景），当设置behavior_compat_options的值包含end_month_calculate时，即SET behavior_compat_options = 'end_month_calculate'，返回结果为'2025-03-31 00:00:00 AD'；否则，返回结果为'2025-03-28 00:00:00 AD'。

- last_day(d)

  描述：用于计算时间点d当月最后一天的时间。

  返回值类型：timestamp、datea（仅在ORA兼容性下生效）

  示例：

  ```
  gaussdb=# SELECT last_day(to_date('2017-01-01', 'yyyy-mm-dd')) AS cal_result;
      cal_result
  ---------------------
   2017-01-31 00:00:00
  ```


版权所有 © 华为云计算技术有限公司

```
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时
gaussdb=# SET nls_date_format = 'YYYY-MM-DD HH24:MI:SS';
gaussdb=# SELECT last_day(datea '2020-11-12 11:12:13');
       last_day
-----------------------
 2020-11-30 11:12:13 AD
(1 row)
```

说明

- 该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型，此时返回值为datea类型。
- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时，使用last_day函数将会调用内置函数b_db_last_day，入参类型可为TEXT、DATE、DATETIME和TIME数据类型。返回值为date类型，支持输入datetime样式数字。

- months_between(d1, d2)

  描述：用于计算时间点d1和时间点d2的月份差值，如果两个日期都是月末或天数相同，则返回整数，否则返回值带小数，按31天/月计算。

  返回值类型：numeric

  示例：

```
gaussdb=# SELECT months_between(to_date('2022-10-31', 'yyyy-mm-dd'), to_date('2022-09-30',
'yyyy-mm-dd'));
 months_between
----------------
              1
(1 row)

gaussdb=# SELECT months_between(to_date('2022-10-30', 'yyyy-mm-dd'), to_date('2022-09-30',
'yyyy-mm-dd'));
 months_between
----------------
              1
(1 row)

gaussdb=# SELECT months_between(to_date('2022-10-29', 'yyyy-mm-dd'), to_date('2022-09-30',
'yyyy-mm-dd'));
    months_between
-----------------------
 .96774193548387096774
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时，会把date类型映射为datea类型
gaussdb=# SET nls_date_format = 'YYYY-MM-DD HH24:MI:SS';
gaussdb=# SELECT months_between('2020-11-12 11:12:13'::date, '2020-11-13 11:12:13'::date);
     months_between
------------------------
 -.03225806451612903226
(1 row)
```

说明

- 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。
- 当在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型。

- next_day(x,y)

  描述：用于计算时间点x开始的下一个星期几（y）的时间。

  返回值类型：timestamp、datea（仅在ORA兼容性下生效）

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT next_day(timestamp '2017-05-25 00:00:00','Sunday')AS cal_result;
    cal_result
---------------------
 2017-05-28 00:00:00
(1 row)
gaussdb=# SELECT next_day(timestamp '2024-01-17 00:00:00', 7.9999999);
ERROR:  Week day out of range, the range is 1 ... 7
CONTEXT:  referenced column: next_day
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s1';
SET
gaussdb=# SELECT next_day(timestamp '2024-01-17 00:00:00', 7.9999999);
     next_day
---------------------
 2024-01-20 00:00:00
(1 row)
-- ORA兼容模式下，打开GUC参数mapping_date_to_datea时
gaussdb=# SET nls_date_format = 'YYYY-MM-DD HH24:MI:SS';
gaussdb=# SELECT next_day(datea '2020-11-12 11:12:13', 'Monday');
       next_day
------------------------
 2020-11-16 11:12:13 AD
(1 row)
```

说明

- 此函数在ORA兼容模式数据库中参数a_format_version值为10c和a_format_dev_version值为s1的情况下，且第二个参数为浮点数时，其将舍弃小数部分，否则将按照四舍五入计算。
- 此函数在ORA兼容模式数据库中，当参数a_format_version值为10c和a_format_dev_version值为s1时，第一个参数的类型才可以是TIMESTAMP WITH TIME ZONE。
- 该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型。
- next_day第二个参数支持缩写(不区分大小写)，包括SUN、MON、TUE、WED、THU、THUR、FRI、SAT。

- tinterval(abstime, abstime)

  描述：用两个具体时间点创建时间间隔。

  返回值类型：tinterval

  示例：

```
gaussdb=# CALL tinterval(abstime 'May 10, 1947 23:59:12', abstime 'Mon May  1 00:30:30 1995');
                    tinterval
----------------------------------------------------
 ["1947-05-10 23:59:12+08" "1995-05-01 00:30:30+08"]
(1 row)
```

- tintervalend(tinterval)

  描述：返回tinterval的结束时间。

  返回值类型：abstime

  示例：

```
gaussdb=# SELECT tintervalend('["Sep 4, 1983 23:59:12" "Oct4, 1983 23:59:12"]');
     tintervalend
------------------------
 1983-10-04 23:59:12+08
(1 row)
```

- tintervalrel(tinterval)

  描述：计算并返回tinterval的相对时间。

  返回值类型：reltime

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT tintervalrel('["Sep 4, 1983 23:59:12" "Oct4, 1983 23:59:12"]');
 tintervalrel
--------------
 1 mon
(1 row)
```

- smalldatetime_ge(smalldatetime, smalldatetime)
  描述：判断是否第一个参数大于等于第二个参数。
  返回值类型：boolean
- smalldatetime_cmp(smalldatetime, smalldatetime)
  描述：对比smalldatetime是否相等。
  返回值类型：integer
- smalldatetime_eq(smalldatetime, smalldatetime)
  描述：对比smalldatetime是否相等。
  返回值类型：boolean
- smalldatetime_gt(smalldatetime, smalldatetime)
  描述：判断是否第一个参数大于第二个参数。
  返回值类型：boolean
- smalldatetime_hash(smalldatetime)
  描述：计算smalldatetime值对应的哈希值。
  返回值类型：integer
- smalldatetime_in(cstring, oid, integer)
  描述：将输入转换为smalldatetime值。
  返回值类型：smalldatetime
- smalldatetime_larger(smalldatetime, smalldatetime)
  描述：返回较大的smalldatetime值。
  返回值类型：smalldatetime
- smalldatetime_le(smalldatetime, smalldatetime)
  描述：判断是否第一个参数小于等于第二个参数。
  返回值类型：boolean
- smalldatetime_lt(smalldatetime, smalldatetime)
  描述：判断是否第一个参数小于第二个参数。
  返回值类型：boolean
- smalldatetime_ne(smalldatetime, smalldatetime)
  描述：比较两个smalldatetime值是否不相等。
  返回值类型：boolean
- smalldatetime_out(smalldatetime)
  描述：将smalldatetime值转换为外部字符串形式。
  返回值类型：cstring
- smalldatetime_send(smalldatetime)
  描述：将smalldatetime值转换为二进制格式。
  返回值类型：bytea

版权所有 © 华为云计算技术有限公司

- smalldatetime_smaller(smalldatetime, smalldatetime)
  描述：返回较小的一个smalldatetime值。
  返回值类型：smalldatetime
- smalldatetime_to_abstime(smalldatetime)
  描述：将smalldatetime值转换为abstime值。
  返回值类型：abstime
- smalldatetime_to_time(smalldatetime)
  描述：将smalldatetime值转换为time值。
  返回值类型：time without time zone
- smalldatetime_to_timestamp(smalldatetime)
  描述：将smalldatetime值转换为timestamp值。
  返回值类型：timestamp without time zone
- smalldatetime_to_timestamptz(smalldatetime)
  描述：将smalldatetime值转换为timestamptz值。
  返回值类型：timestamp with time zone
- smalldatetime_to_varchar2(smalldatetime)
  描述：将smalldatetime值转换为varchar2值。
  返回值类型：character varying

版权所有 © 华为云计算技术有限公司

说明

获取当前时间有多种方式，请根据实际业务从场景选择合适的接口：

- 以下接口按照当前事务的开始时刻返回值：
  ```
  CURRENT_DATE CURRENT_TIME
  CURRENT_TIME(precision)
  CURRENT_TIMESTAMP(precision)
  LOCALTIME
  LOCALTIMESTAMP
  LOCALTIME(precision)
  LOCALTIMESTAMP(precision)
  transaction_timestamp()
  now()
  ```
  其中CURRENT_TIME和CURRENT_TIMESTAMP(precision)传递带有时区的值；LOCALTIME和LOCALTIMESTAMP传递的值不带时区。CURRENT_TIME、LOCALTIME和LOCALTIMESTAMP可以指定精度参数，这会导致结果在秒字段中四舍五入到小数位数。如果没有精度参数，结果将被给予所能得到的全部精度。

  因为这些函数全部都按照当前事务的开始时刻返回结果，所以它们的值在事务运行的整个期间内都不改变。可以认为这是一个特性：目的是为了允许一个事务在“当前”时间上有一致的概念，这样在同一个事务里的多个修改可以保持同样的时间戳。其中transaction_timestamp()等价于CURRENT_TIMESTAMP(precision)，表示当前语句所在事务的开启时间。now()等效于transaction_timestamp()。

  需要特别注意，当事务开启时，在sql_compatibility = 'ORA'，且GUC参数a_format_date_timestamp值为on的情况下，CURRENT_DATE、CURRENT_TIMESTAMP(precision)、LOCALTIMESTAMP、LOCALTIMESTAMP(precision)函数返回的结果为当前SQL执行开始时刻的时间戳，关闭GUC参数a_format_date_timestamp，则返回的结果为事务开启日期或日期及时间。

  在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时CURDATE、CURRENT_DATE、CURTIME、CURRENT_TIME、CURRENT_TIMESTAMP、NOW、LOCALTIME、LOCALTIMESTAMP、UTC_TIME、UTC_TIMESTAMP都返回SQL语句开始执行的时间，SYSDATE返回函数调用开始的时间。transaction_timestamp()依然表示事务开始时间，返回值类型改为timestamp without time zone。
- 以下接口返回当前语句开始时间：
  ```
  statement_timestamp()
  ```
  statement_timestamp()返回当前语句的开始时刻（更准确的说是收到客户端最后一条命令的时间）。statement_timestamp()和transaction_timestamp()在一个事务的第一条命令期间返回值相同，但是在随后的命令中却不一定相同。
- 以下接口返回函数被调用时的真实当前时间：
  ```
  clock_timestamp()
  timeofday()
  ```
  clock_timestamp()返回真正的当前时间，因此它的值甚至在同一条SQL命令中都会变化。timeofday()和clock_timestamp()相似，timeofday()也返回真实的当前时间，但是它的结果是一个格式化的text串，而不是timestamp with time zone值。

- convert_tz(dt, from_tz, to_tz)

  描述：将时间日期值dt从from_tz时区转换到to_tz时区。

  参数：如**表1-51**所示。

版权所有 © 华为云计算技术有限公司

表 1-51 convert_tz 参数说明

参数名称	类型	描述	取值范围
dt	datetime、date、text、numeric	时间日期值。	[0000-01-01 00:00:00.000000, 9999-12-31 23:59:59.999999]。
from_tz	text	源时区，表示dt当前所属的时区。	• 偏移量格式：[-15:59, 15:00]，如'+08:00'。表示相较于UTC时间的偏移。 • 命名时区：如'MET'、'UTC'等。表示时区名称，具体请参见 **PG_TIMEZONE_NAMES** 系统视图。
to_tz	text	目标时区，表示转换后的时区。	

返回值类型：datetime

示例：

```
gaussdb=# SET b_format_dev_version = 's1';
SET
gaussdb=# SET b_format_version = '5.7';
SET
gaussdb=# SELECT convert_tz(cast('2023-01-01 10:10:10' as datetime), '+00:00', '+01:00');
    convert_tz
---------------------
 2023-01-01 11:10:10
(1 row)
gaussdb=# SELECT convert_tz(cast('2023-01-01' as date), '+00:00', '+01:00');
    convert_tz
---------------------
 2023-01-01 01:00:00
(1 row)
gaussdb=# SELECT convert_tz('2023-01-01 10:10:10', '+00:00', '+01:00');
    convert_tz
---------------------
 2023-01-01 11:10:10
(1 row)
gaussdb=# SELECT convert_tz('2023-01-01', '+00:00', '+01:00');
    convert_tz
---------------------
 2023-01-01 01:00:00
(1 row)
gaussdb=# SELECT convert_tz(20230101101010, '+00:00', '+01:00');
    convert_tz
---------------------
 2023-01-01 11:10:10
(1 row)
gaussdb=# SELECT convert_tz(20230101, '+00:00', '+01:00');
    convert_tz
---------------------
 2023-01-01 01:00:00
(1 row)
gaussdb=# SELECT convert_tz('2023-01-01 10:10:10', 'UTC', 'PRC');
    convert_tz
---------------------
 2023-01-01 18:10:10
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- sec_to_time(seconds)

  描述：将seconds表示的秒数转换为time类型的时间。

  参数：如表1-52所示。

  表 1-52 sec_to_time 参数说明

参数名称	类型	描述	取值范围
seconds	numeric、text类型。	秒数。	[-3020399, +3020399]，对应time类型范围[-838:59:59, 838:59:59]，对于越界输入会截断到边界值。

  返回值类型：time without time zone

  示例：

```
gaussdb=# SET b_format_dev_version = 's1';
SET
gaussdb=# SET b_format_version = '5.7';
SET
gaussdb=# SELECT sec_to_time(2000);
 sec_to_time
-------------
 00:33:20
(1 row)
gaussdb=# SELECT sec_to_time('-2000');
 sec_to_time
-------------
 -00:33:20
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- adddate(date, INTERVAL val unit)

  描述：将日期加上一定的时间间隔，返回一个新的日期。

  参数：如表1-53所示。

  表 1-53 adddate(date, INTERVAL val unit)参数说明

参数名称	类型	描述	取值范围
date	时间类型表达式、text、datetime、date或time等类型	要加上时间间隔的日期。	和类型范围一致。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述	取值范围
val	整数、浮点数、字符串等类型	要加上的时间间隔。	和类型范围一致。
unit	关键字	时间间隔的单位。	YEAR、QUARTER、MONTH、WEEK、DAY、HOUR、MINUTE、SECOND、MICROSECOND等，具体请参见**时间间隔表达式**。

返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE

示例：

```
gaussdb=# SELECT adddate('2018-05-01', INTERVAL 1 DAY);
  adddate
------------
 2018-05-02
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 在PREPARE语句中，adddate的第二个参数为interval表达式，当使用参数$1完全替代时，会导致结果非预期，例如：PREPARE p1 AS SELECT adddate('2023-01-01 10:00:00', $1); EXECUTE p1(interval 13 hour);该用例返回非预期结果为'2023-01-01 10:00:00'。

- adddate(expr, days)

  描述：将日期加上一定的天数值，返回一个新的日期。

  参数：

  - expr：指定开始的日期时间，时间类型表达式，TEXT、DATE、DATETIME或TIME等类型。
  - days：需要加上的天数，int类型。

  返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE

  示例：

  ```
  gaussdb=# SELECT ADDDATE('2018-05-01', 1);
    adddate
  ------------
   2018-05-02
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- curdate()

  描述：返回本地函数调用开始时刻的系统日期。支持在同一连接内修改时区，返回的日期受时区影响。

  返回值类型：date

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT curdate();
  curdate
------------
 2023-08-10
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- curtime([precision])

  描述：返回SQL调用开始时刻的系统时间。

  参数：表示精度（秒后小数的位数），int类型，范围为[0,6]，缺省时默认值为0，如果提供的值能够转换为范围内的整型值，函数将返回对应精度的时间值，否则将报错。

  返回值类型：time without time zone

  实现方式：注册系统函数curtime。

  示例：

```
gaussdb=# SELECT curtime(3);
   curtime
--------------
 16:59:57.203
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- date_add(date,INTERVAL val unit)

  描述：向指定日期添加一段时间，并返回计算结果。

  参数：参数介绍如**表1-54**所示。

表 1-54 date_add(date,INTERVAL val unit)参数说明

参数名称	类型	描述	取值范围
date	时间类型表达式、text、datetime、date或time等类型	要加上时间间隔的日期。	和类型范围一致。
val	整数、浮点数、字符串等类型	要加上的时间间隔。	和类型范围一致。
unit	关键字	时间间隔的单位。	YEAR、QUARTER、MONTH、WEEK、DAY、HOUR、MINUTE、SECOND、MICROSECOND等，详见**时间间隔表达式**。

返回值类型：单个返回值，具体返回类型描述如**表1-55**所示。

版权所有 © 华为云计算技术有限公司

表 1-55 date_add(date,INTERVAL val unit)返回值类型说明

返回值类型	描述
TEXT	当date入参为text类型。
DATE	当date入参为date类型，且unit入参大于等于day（如week、month）。
TIME WITHOUT TIMEZONE	当date入参为time类型。
DATETIME	当date入参为datetime类型；或当date入参为date类型，且unit入参小于day（如hour、second）。

示例：

```
gaussdb=#  SELECT date_add('2018-05-01', INTERVAL 1 DAY);
 date_add
------------
 2018-05-02
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 在PREPARE语句中，date_add的第二个参数为interval表达式，不支持使用参数$1替代使用，例如：PREPARE p1 AS SELECT date_add('2023-01-01 10:00:00', $1);

- date_add(expr, days)

  描述：将日期加上一定的天数值，返回一个新的日期。

  参数：

  - expr：指定开始的日期时间，时间类型表达式，TEXT、DATE、DATETIME或TIME等类型。
  - days：需要加上的天数，int类型。

  返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE

  示例：

  ```
  gaussdb=# SELECT DATE_ADD('2018-05-01', 1);
   date_add
  ------------
   2018-05-02
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- date_format(date, format)

  描述：将日期时间以指定格式输出。

  参数：参数介绍如**表1-56**所示。

版权所有 © 华为云计算技术有限公司

表 1-56 date_format 参数说明

参数名称	类型	描述	取值范围
date	时间类型表达式、TEXT、DATETIME、DATE或TIME等类型	需要格式化的日期。	和类型范围一致。
format	text	格式化字符串。	如表1-57所示。

format参数列表如表1-57所示。

表 1-57 format 的取值及含义

format取值	含义
%a	星期的缩写（Sun~Sat）。
%b	月份的缩写（Jan~Dec）。
%c	月份数字（1~12）。
%D	带有英语前缀的月份中的每天（1st, 2nd, 3rd, …）。
%d	月份中的每天的两位数字表示（01~31）。
%e	月份中的每天的数字表示（1~31）。
%f	微秒（000000~999999）。
%H	小时（00~23）。
%h	小时（01~12）。
%I	小时（01~12）。
%i	分钟（00~59）。
%j	一年中的每天（001~366）。
%k	小时（0~23）。
%l	小时（1~12）。
%M	月份名称（January~December）。
%m	两位数字月份（01~12）。
%p	AM或者PM。
%r	十二小时制时间（hh:mm:ss后跟AM或PM）。
%S	秒（00~59）。

版权所有 © 华为云计算技术有限公司

format取值	含义
%s	秒（00~59）。
%T	二十四小时制时间（hh:mm:ss）。
%U	一年中的星期（00~53），每周的开始是星期天。
%u	一年中的星期（00~53），每周的开始是星期一。
%V	一年中的星期（01~53），每周的开始是星期天。
%v	一年中的星期（01~53），每周的开始是星期一。
%W	星期的名称（Sunday~Saturday）。
%w	星期中的每天（0~6）。其中，星期天用0表示，星期六用6表示。
%X	一年中的星期，每周的开始是星期天，四位数字，用于%V。
%x	一年中的星期，每周的开始是星期一，四位数字，用于%v。
%Y	四位数字年份。
%y	两位数字年份。

返回值类型：text

示例：

```
gaussdb=# SELECT date_format('2023-10-11 12:13:14.151617','%b %c %M %m');
    date_format
--------------------
 Oct 10 October 10
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- date_sub(date, INTERVAL val unit)

  描述：将日期减上一定的时间间隔，返回一个新的日期。

  参数：参数介绍如表1-58所示。

  表 1-58 date_sub(date, INTERVAL val unit)参数说明

参数名称	类型	描述	取值范围
date	时间类型表达式、text、datetime、date或time等类型	要加上时间间隔的日期。	和类型范围一致。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述	取值范围
val	整数、浮点数、字符串等类型	要加上的时间间隔。	和类型范围一致。
unit	关键字	时间间隔的单位。	YEAR、QUARTER、MONTH、WEEK、DAY、HOUR、MINUTE、SECOND、MICROSECOND等，具体请参见**时间间隔表达式**。

返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE。

示例：

```
gaussdb=#  SELECT date_sub('2018-05-01', INTERVAL 1 YEAR);
  date_sub
------------
 2017-05-01
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 在PREPARE语句中，date_sub的第二个参数为interval表达式，不支持使用参数$1替代使用，例如：PREPARE p1 AS SELECT date_sub('2023-01-01 10:00:00', $1);

- date_sub(expr, days)

  描述：指定开始的日期时间和要从开始的日期时间减去的天数值，返回相减后的日期结果值。

  参数：

  – expr：指定开始的日期时间，时间类型表达式，TEXT、DATE、DATETIME或TIME等类型。

  – days：需要减去的天数，int类型。

  返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE

  示例：

  ```
  gaussdb=#  SELECT date_sub('2023-1-1', 20);
    date_sub
  ------------
   2022-12-12
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- datediff(expr1, expr2)

  描述：返回两个时间表达式之间相差的天数。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：int

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=#  SELECT datediff('2021-11-12','2021-11-13');
 datediff
----------
       -1
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- day()/dayofmonth()

  描述：提取日期时间的天数部分，将结果返回。dayofmonth()是day()的别名，功能完全相同。

  参数：入参为指定提取的日期时间，是时间类型表达式、text、datetime、date或time等类型。

  返回值类型：int

  示例：

```
gaussdb=# SELECT day('2023-01-02');
 day
-----
   2
(1 row)
gaussdb=# SELECT dayofmonth('23-05-22');
 dayofmonth
------------
         22
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- dayname()

  描述：返回日期的星期几名称。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。

  返回值类型：text

  示例：

```
gaussdb=# SELECT dayname('2023-10-11');
  dayname
-----------
 Wednesday
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 返回值使用的语言由GUC参数lc_time_names控制。

- dayofweek()

  描述：返回日期的工作日索引（1=周日，2=周一，…，7=周六）。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：bigint

  示例：

```
gaussdb=# SELECT dayofweek('2023-04-16');
 dayofweek
```


版权所有 © 华为云计算技术有限公司

```
-----------
         1
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- dayofyear()

  描述：返回一年中的第几天。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。

  返回值类型：int，范围1-366。

  示例：

```
gaussdb=# SELECT dayofyear('2000-12-31');
 dayofyear
-----------
       366
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- extract(unit FROM date)

  描述：提取时间的一部分 ，此函数在其他模式可以使用，但行为差异较大。当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'时该函数的表现如下所述。

  参数：

  – unit：text。具体的取值类型如**表1-59**所示。
  – date：date/time/datetime/text/datea（仅在ORA兼容性下生效）等时间类型表达式。

  返回值类型：bigint

表 1-59 unit 的取值及含义

UNIT 取值	含义
MICROSECOND	微秒
SECOND	秒
MINUTE	分钟
HOUR	小时
DAY	天数
WEEK	周数
MONTH	月份
QUARTER	季度
YEAR	年份

版权所有 © 华为云计算技术有限公司

UNIT 取值	含义
SECOND_MICROSECOND	秒、微秒的拼接值
MINUTE_MICROSECOND	分钟、秒、微秒的拼接值
MINUTE_SECOND	分钟、秒的拼接值
HOUR_MICROSECOND	小时、分钟、秒、微秒的拼接值
HOUR_SECOND	小时、分钟、秒的拼接值
HOUR_MINUTE	小时、分钟的拼接值
DAY_MICROSECOND	天数、小时、分钟、秒、微秒的拼接值
DAY_SECOND	天数、小时、分钟、秒的拼接值
DAY_MINUTE	天数、分钟的拼接值
DAY_HOUR	天数、小时的拼接值
YEAR_MONTH	年份、月份的拼接值
EPOCH	表示自1970-01-01 00:00:00-00 UTC以来的秒数或时间间隔的总秒数

示例：

```
gaussdb=# SELECT extract(YEAR FROM '2023-10-11');
 extract
---------
    2023
(1 row)

gaussdb=# SELECT extract(QUARTER FROM '2023-10-11');
 extract
---------
       4
(1 row)

gaussdb=# SELECT extract(MONTH FROM '2023-10-11');
 extract
---------
      10
(1 row)

gaussdb=# SELECT extract(WEEK FROM '2023-10-11');
 extract
---------
      41
(1 row)

gaussdb=# SELECT extract(DAY FROM '2023-10-11');
 extract
---------
      11
(1 row)

gaussdb=# SELECT extract(HOUR FROM '2023-10-11 12:13:14');
 extract
---------
```


版权所有 © 华为云计算技术有限公司

```
    12
(1 row)
```

说明

- 在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7以及b_format_dev_version值为s1时调用extract函数，实际调用的函数为b_extract；在其他情况中，实际调用的函数为date_part，可以用“\df b_extract”等指令查询函数详细入参与返回值。
- 该函数在sql_compatibility = 'ORA'时：
  - GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型，此时unit仅支持年(YEAR)、月(MONTH)、日(DAY)，指定其他unit会报错。
  - 入参为datea类型时，无论是否对GUC参数mapping_date_to_datea的值进行设置（on或off），此时unit仅支持年(YEAR)、月(MONTH)、日(DAY)，指定其他unit会报错。

GUC参数default_week_format用于处理特殊周问题，默认值为0。详细描述如表1-60所示。

表 1-60 default_week_format 说明

default_week_format	一周的第一天	范围	哪一周是第一周
0	周日	0~53	今年有周日的周。
1	周一	0~53	今年有4天及以上的周。
2	周日	1~53	今年有周日的周。
3	周一	1~53	今年有4天及以上的周。
4	周日	0~53	今年有4天及以上的周。
5	周一	0~53	今年有周一的周。
6	周日	1~53	今年有4天及以上的周。
7	周一	1~53	今年有周一的周。

- from_days(days)

  描述：给定一个天数，返回相应的日期值。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。

  返回值类型：date

  示例：

```
gaussdb=# SELECT from_days(36524);--0099-12-31
 from_days
------------
 0099-12-31
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 对于空字符和0字符，将作为0处理；对于无法正确转换为bigint的入参，将报错。
- 入参小于366时，返回日期为0000-00-00。

- from_unixtime(unix_timestamp[,format])

  描述：将Unix时间戳转换为日期时间格式的函数。Unix时间戳是指从1970年1月1日08:00:00 UTC到指定时间的秒数。

  参数：

  – unix_timestamp：unix时间戳，numeric类型。

  – format：时间格式，text类型。

  返回值类型：text/datetime

  示例：

```
gaussdb=# SELECT from_unixtime(1111885200);
    from_unixtime
---------------------
 2005-03-27 09:00:00
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- get_format({DATE | TIME | DATETIME | TIMESTAMP}, {'EUR' | 'USA' | 'JIS' | 'ISO' | 'INTERNAL'})

  描述： 用于获取指定时间类型在某种地区标准下对应的日期/时间格式字符串。

  参数：

  – DATE|TIME|DATETIME|TIMESTAMP：表示时间类型，为关键字。

  – 'EUR'|'USA'|'JIS'|'ISO'|'INTERNAL'：五种时间格式，text类型。

  返回值类型：text

  示例：

```
gaussdb=# SELECT get_format(date, 'eur');
 get_format
------------
 %d.%m.%y
(1 row)
gaussdb=# SELECT get_format(date, 'usa');
 get_format
------------
 %m.%d.%y
(1 row)
```

  说明

  - 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
  - 在PREPARE语句中，get_format的第一个参数为关键字，不支持使用参数$1替代使用，例如：PREPARE p1 AS SELECT get_format($1, 'usa');

- hour()

  描述：输入一个时间类型，返回对应的时间的小时部分。

  参数：时间类型表达式、text、datetime、date或time等类型。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

示例：

```
gaussdb=# SELECT hour('10:10:10.1');
 hour
------
   10
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- makedate(year,dayofyear)

  描述：根据年份和天数值返回日期。

  参数：

  - year：参数类型为bigint。
  - dayofyear：参数类型为bigint，该年的第若干天，允许跨年，小于等于0时返回null。

  返回值类型：date

  示例：

  ```
  gaussdb=# SELECT makedate(2000, 60);
    makedate
  ------------
   2000-02-29
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- microsecond()

  描述：输入一个时间类型，返回对应的时间的微秒部分。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：bigint

  示例：

  ```
  gaussdb=# SELECT microsecond('2023-5-5 10:10:10.24485');
   microsecond
  -------------
       244850
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- minute()

  描述：输入一个时间类型，返回对应时间的分钟部分。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：bigint

  示例：

  ```
  gaussdb=# SELECT minute(time'10:10:10');
   minute
  --------
  ```


版权所有 © 华为云计算技术有限公司

```
   10
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- month()

  描述：返回指定日期时间中的月份。

  参数：入参为指定提取的日期时间，是时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。

  返回值类型：int

  示例：

```
gaussdb=# SELECT month('2021-11-30');
 month
-------
   11
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- monthname()

  描述：返回日期的月份全名。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。

  返回值类型：text

  示例：

```
gaussdb=# SELECT monthname('2023-02-28');
 monthname
-----------
 February
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 返回值使用的语言由GUC参数lc_time_names控制。

- period_add(period, month_number)

  描述：在指定时间段上加上指定的月数，将结果作为时间段返回。

  参数：

  - period：参数类型为bigint，使用YYYYMM或YYMM格式表示的日期。
  - month_number：参数类型为bigint，需要加上的月数，允许为负数。

  返回值类型：bigint，使用YYYYMM格式表示的日期。

  示例：

```
gaussdb=# SELECT period_add(202205, -12);
 period_add
------------
   202105
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- period_diff(p1, p2)

  描述：计算两个时间段的月份差。

  参数：参数类型为bigint，两个参数均使用YYMM或YYYYMM格式来表示时间段。

  返回值类型：bigint，月份差。

  示例：

```
gaussdb=# SELECT period_diff('202101', '202102');
 period_diff
-------------
          -1
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- second()

  描述：输入一个时间类型，返回对应的时间的秒部分。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：bigint

  示例：

```
gaussdb=# SELECT second('2023-5-5 10:10:10');
 second
--------
     10
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- quarter()

  描述：输入一个日期类型，返回对应的日期的季度部分。

  参数：入参为指定提取的日期时间，是时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。

  返回值类型：bigint

  示例：

```
gaussdb=# SELECT quarter('2012-1-1');
 quarter
---------
       1
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- str_to_date(str, format)

  描述：将指定的字符串根据指定日期格式转为日期/时间。

  参数：

版权所有 © 华为云计算技术有限公司

- str：text类型，需要格式化成日期的字符串。
- format：text类型，格式化字符串。format参数列表如**表1-61**所示。

表 1-61 format 的取值及含义

format取值	含义
%a	星期的缩写（Sun~Sat）。
%b	月份的缩写（Jan~Dec）。
%c	月份数字（1~12）。
%D	带有英语前缀的月份中的每天（1st, 2nd, 3rd, …）。
%d	月份中的每天的两位数字表示（01~31）。
%e	月份中的每天的数字表示（1~31）。
%f	微秒（000000~999999）。
%H	小时（00~23）。
%h	小时（01~12）。
%I	小时（01~12）。
%i	分钟（00~59）。
%j	一年中的每天（001~366）。
%k	小时（0~23）。
%l	小时（1~12）。
%M	月份名称（January~December）。
%m	两位数字月份（01~12）。
%p	AM或者PM。
%r	十二小时制时间（hh:mm:ss后跟AM或PM）。
%S	秒（00~59）。
%s	秒（00~59）。
%T	二十四小时制时间（hh:mm:ss）。
%U	一年中的星期（00~53），每周的开始是星期天。
%u	一年中的星期（00~53），每周的开始是星期一。
%V	一年中的星期（01~53），每周的开始是星期天。
%v	一年中的星期（01~53），每周的开始是星期一。
%W	星期的名称（Sunday~Saturday）。

版权所有 © 华为云计算技术有限公司

format取值	含义
%w	星期中的每天（0~6）。其中，星期天用0表示，星期六用6表示。
%X	一年中的星期，每周的开始是星期天，四位数字，用于%V。
%x	一年中的星期，每周的开始是星期一，四位数字，用于%v。
%Y	四位数字年份。
%y	两位数字年份。

返回值类型：text

示例：

```
gaussdb=# SELECT str_to_date('May 1, 2013','%M %d,%Y');--2013-05-01
 str_to_date
-------------
 2013-05-01
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 只支持返回YYYY-MM-DD格式的时间。
- 对于包含0年0月0日的时间或者只包含时分秒的时间，会告警并返回NULL。

- subdate(expr, days)

  描述：指定开始的日期时间和要从开始的日期时间减去的天数值，返回相减后的日期结果值。

  参数：

  – expr：指定开始的日期时间，时间类型表达式，TEXT、DATE、DATETIME或TIME等类型。

  – days：需要减去的天数，int类型。

  返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE

  示例：

  ```
  gaussdb=# SELECT subdate('2023-1-1', 20);
    subdate
  ------------
   2022-12-12
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- subdate(expr, INTERVAL val unit)

  描述：指定开始的日期时间和要从开始的日期时间减去的时间间隔，返回相减后的日期结果值。

  参数：如**表1-62**所示。

版权所有 © 华为云计算技术有限公司

表 1-62 subdate(expr,INTERVAL val unit)参数说明

参数名称	类型	描述	取值范围
expr	时间类型表达式、text、datetime、date或time等类型	指定开始的日期时间。	和类型范围一致。
val	整数、浮点数、字符串等类型	指定要减去的时间间隔。	和类型范围一致。
unit	关键字		YEAR、QUARTER、MONTH、WEEK、DAY、HOUR、MINUTE、SECOND、MICROSECOND等，具体请参见**时间间隔表达式**。

返回值类型：TEXT、DATE、DATETIME或TIME WITHOUT TIMEZONE

示例：

```
gaussdb=# SELECT subdate('2018-05-01', INTERVAL 1 YEAR);
  subdate
------------
 2017-05-01
(1 row)
```

说明

- 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
- 在PREPARE语句中，subdate的第二个参数为interval表达式，当使用参数$1完全替代时，会导致结果非预期，例如：PREPARE p1 AS SELECT subdate('2023-01-01 10:00:00', $1); EXECUTE p1(interval 13 hour);该用例返回非预期结果为'2023-01-01 10:00:00'。

- subtime(expr1, expr2)

  描述：返回时间或时间日期表达式expr1减去时间表达式expr2后的时间点。

  参数：expr1为TIMESTAMP WITHOUT TIME ZONE/TIME类型的表达式，expr2为TIME表达式。

  返回值类型：TEXT、TIMESTAMP WITHOUT TIME ZONE或TIME。返回值类型与expr1类型有关，两入参存在TEXT类型则返回TEXT，解析为TIMESTAMP WITHOUT TIME ZONE则返回TIMESTAMP WITHOUT TIME ZONE，解析为TIME则返回TIME。

  示例：

  ```
  gaussdb=# SELECT subtime('2000-03-01 20:59:59', '22:58');
        subtime
  ---------------------
   2000-02-29 22:01:59
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- addtime(expr1, expr2)

  描述：返回时间或时间日期表达式expr1与时间表达式expr2相加后的时间点，返回值的格式与expr1保持一致。

  参数：expr1为timestamp without time zone/time类型的表达式，expr2为time表达式。

  返回值类型：text、timestamp without time zone或time。返回值类型与expr1类型有关，解析为timestamp without time zone则返回timestamp without time zone，解析为time则返回time。

  示例：

```
gaussdb=# SELECT addtime('2000-03-01 20:59:59', '00:00:01');
      addtime
---------------------
 2000-03-01 21:00:00
(1 row)
```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- time_format(time, format)

  描述：根据格式说明符format格式化time入参。

  参数：

  - time为时间类型表达式、text、datetime、date或time等类型。
  - format为text类型，支持格式如表1-63所示。

  表 1-63 format 的取值及含义

format取值	含义
%f	微秒（000000~999999）。
%H	小时（00~23）。
%h、%I	小时（00~12）。
%l	小时（0~12）。
%k	小时（0~838）。
%i	分钟（00~59）。
%p	AM or PM。
%r	时间为12小时AM或PM格式（hh：mm：ss AM / PM）。
%S、%s	秒（00~59）。
%T	24小时格式的时间（hh：mm：ss）。

版权所有 © 华为云计算技术有限公司

format取值	含义
%a、%b、%D、%j、%M、%U、%u、%V、%v、%W、%w、%X、%x	NULL。
%c、%e	0。
%d、%m、%y	00。
%Y	0000。
%其他字符/其他字符，如%A/A	返回字符本身，如A。
%单个字符+字符串s	解析（%单个字符），拼接s。

返回值类型：text

示例：

```
gaussdb=# SELECT time_format('25:30:30', '%T|%r|%H|%h|%I|%i|%S|%f|%p|%k');
                time_format
-----------------------------------------------------
 25:30:30|01:30:30 AM|25|01|01|30|30|000000|AM|25
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- time_to_sec()

  描述：将time入参转化为秒数。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。时间表达式按照TIME来解析。

  返回值类型：int

  示例：

```
gaussdb=# SELECT time_to_sec('00:00:01');
 time_to_sec
-------------
           1
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- timediff()

  描述：计算两个时间之间的差值，并返回一个时间间隔。

  参数：有两个参数，为时间类型表达式、text、datetime、date或time等类型。

  返回值类型：TIME WITHOUT TIMEZONE

  示例：

```
gaussdb=# SELECT timediff(date'2022-12-30',20221229);
 timediff
----------
 24:00:00
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- timestampadd(unit, interval, datetime_expr)

  描述：返回一个新的时间戳，该时间戳是通过将unit的多个interval添加到datetime_expr来计算的。

  参数：如**表1-64**所示。

  表 1-64 timestampadd 参数说明

参数	类型	描述	取值范围
unit	关键字	时间间隔单位。	YEAR、QUARTER、MONTH、WEEK、DAY、HOUR、MINUTE、SECOND、MICROSECOND、Y、MM（月）、D、H、M（分钟）、S、US、YRS、QTR、MON、HRS、MIN、YEARS、WEEKS或HOURS等。
interval	numeric	时间间隔数值。	和类型范围一致。
datetime_expr	时间类型表达式、text、datetime、date或time等类型	需要变更的时间日期。若为text类型则返回text，为time则返回time，其他情况返回datetime。	和类型范围一致。

  返回值类型：DATETIME、TEXT、TIME WITHOUT TIMEZONE

  示例：

```
gaussdb=# SELECT timestampadd(DAY,-2,'2022-07-27');
 timestampadd
--------------
 2022-07-25
(1 row)
```

  说明

  - 该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。
  - timestampadd实际注册的函数为timestamp_add，可以用“\df timestamp_add”等指令查询函数详细入参与返回值。
  - 在PREPARE语句中，timestampadd的第一个参数为关键字，不支持使用参数$1替代使用，例如：PREPARE p1 AS SELECT timestampadd($1, -2, '2023-01-01');

- to_days()

  描述：返回指定日期自0000年1月1日以来的天数。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。时间表达式按照DATE来解析。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

示例：

```
gaussdb=# SELECT to_days('2000-1-1');
 to_days
---------
  730485
(1 row)
```

说明

该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- to_seconds()

  描述：返回入参自公元0年以来的秒数。

  参数：入参为时间类型表达式、TEXT、DATETIME、DATE或TIME等类型。时间表达式按照datetime来解析。

  返回值类型：bigint

  示例：

  ```
  gaussdb=# SELECT to_seconds('2009-11-29 13:43:32');
   to_seconds
  -------------
   63426721412
  (1 row)
  ```

  说明

  该函数在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时有效。

- unix_timestamp([date])

  描述：返回一个UNIX时间戳，表示自“1970-01-01 08:00”UTC以来的秒数。若无入参，则默认值为调用函数时的datetime时间戳。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SELECT unix_timestamp('2022-12-22');
   unix_timestamp
  ----------------
      1671638400
  (1 row)
  ```

  说明

  该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- utc_date()

  描述：将函数执行时的当前UTC日期作为“YYYY-MM-DD”格式的值返回。

  返回值类型：date

  示例：

  ```
  gaussdb=# SELECT utc_date();
    utc_date
  ------------
   2023-08-10
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

说明

该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- utc_time([precision])

  描述：以“HH:MM:SS”格式返回函数执行时的当前UTC时间。

  参数：表示时间精度，int类型，范围为[0,6]，缺省时默认值为0。

  返回值类型：time without time zone

  示例：

```
gaussdb=# SELECT utc_time();
 utc_time
----------
 11:47:53
(1 row)
```

  说明

  该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- utc_timestamp([precision])

  描述：以“YYYY-MM-DD HH:MM:SS”格式返回函数执行时的当前UTC时间戳。

  参数：表示时间精度，int类型，范围为[0,6]，缺省时默认值为0。

  返回值类型：timestamp without time zone

  示例：

```
gaussdb=# SELECT utc_timestamp();
    utc_timestamp
---------------------
 2023-08-21 11:51:19
(1 row)
```

  说明

  该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- week(date[, mode])

  描述：返回日期所在的周数。

  参数：

  - date表示指定日期时间，为时间类型表达式、text、datetime、date或time等类型。
  - 可选参数mode的含义如**表1-65**所示，缺省时则为0。

  表 1-65 mode 说明

mode	一周的第一天	范围	哪一周是第一周
0	周日	0~53	今年有周日的周。
1	周一	0~53	今年有4天及以上的周。
2	周日	1~53	今年有周日的周。

版权所有 © 华为云计算技术有限公司

mode	一周的第一天	范围	哪一周是第一周
3	周一	1~53	今年有4天及以上的周。
4	周日	0~53	今年有4天及以上的周。
5	周一	0~53	今年有周一的周。
6	周日	1~53	今年有4天及以上的周。
7	周一	1~53	今年有周一的周。

返回值类型：bigint

示例：

```
gaussdb=# SELECT week(date'2000-01-01', 1);
 week
------
    0
(1 row)

gaussdb=# SELECT week('2000-01-01', 2);
 week
------
   52
(1 row)
```

说明

该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- weekday()

  描述：返回一个日期的工作日索引值，即星期一为0，星期二为1，星期三为2，星期四为3，星期五为4，星期六为5，星期日为6。

  参数：时间类型表达式、text、datetime、date或time等类型。

  返回值类型：bigint

  示例：

```
gaussdb=# SELECT weekday('1970-01-01 12:00:00');
 weekday
---------
       3
(1 row)
```

  说明

  该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- weekofyear(date)

  描述：返回日期时间的日历周，范围1~53。该函数等同于week(date, 3)，具体请参见**week(date[,mode])**。

  参数：表示指定日期时间，为时间类型表达式、text、datetime、date或time等类型。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

示例：

```
gaussdb=# SELECT weekofyear('1970-05-22');
 weekofyear
------------
         21
(1 row)
```

说明

该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- year()

  描述：提取日期时间的年份部分，将结果返回。

  参数：入参为指定提取的日期时间，是时间类型表达式、text、datetime、date或time等类型。

  返回值类型：int

  示例：

```
gaussdb=# SELECT year('23-05-22');
 year
------
 2023
(1 row)
```

说明

该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

- yearweek(date[, mode])

  描述：返回日期的年份和周数。

  参数：

  - date表示指定日期时间，为DATE、DATETIME、TIMESTAMP、TIME、TEXT、NUMERIC等类型。
  - 可选参数mode的含义如**表1-66**所示，缺省时则为0。

表 1-66 mode 说明

mode	一周的第一天	范围	哪一周是第一周
0	周日	0~53	今年有周日的周。
1	周一	0~53	今年有4天及以上的周。
2	周日	1~53	今年有周日的周。
3	周一	1~53	今年有4天及以上的周。
4	周日	0~53	今年有4天及以上的周。
5	周一	0~53	今年有周一的周。

版权所有 © 华为云计算技术有限公司

mode	一周的第一天	范围	哪一周是第一周
6	周日	1~53	今年有4天及以上的周。
7	周一	1~53	今年有周一的周。

返回值类型：bigint

示例：

```
gaussdb=# SELECT yearweek(datetime'2000-01-01', 3);
 yearweek
----------
   199952
(1 row)
```

说明

该函数仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7和b_format_dev_version值为s1时有效。

表1-67显示了可以用于截断日期和时间值的模板。

表 1-67 用于日期/时间截断的模式

类别	模式	描述
微秒	MICROSECON	截断日期/时间，精确到微秒（000000~999999）
	US	
	USEC	
	USECOND	
毫秒	MILLISECON	截断日期/时间，精确到毫秒（000~999）
	MS	
	MSEC	
	MSECOND	
秒	S	截断日期/时间，精确到秒（00~59）
	SEC	
	SECOND	
分钟	M	截断日期/时间，精确到分钟（00~59）
	MI	
	MIN	
	MINUTE	
小时	H	截断日期/时间，精确到小时（00~23）

版权所有 © 华为云计算技术有限公司

类别	模式	描述
	HH	
	HOUR	
	HR	
天	D	截断日期/时间，精确到天（01-01~12-31）
	DAY	
	DD	
	DDD	
	J	
周	W	截断日期/时间，精确到周（本周的第一天）
	WEEK	
月	MM	截断日期/时间，精确到月（本月的第一天）
	MON	
	MONTH	
季度	Q	截断日期/时间，精确到季度（本季度的第一天）
	QTR	
	QUARTER	
年	Y	截断日期/时间，精确到年（本年的第一天）
	YEAR	
	YR	
	YYYY	
十年	DEC	截断日期/时间，精确到十年（本十年的第一天）
	DECADE	
世纪	C	截断日期/时间，精确到世纪（本世纪的第一天）
	CC	
	CENT	
	CENTURY	
千年	MIL	截断日期/时间，精确到千年（本千年的第一天）
	MILLENNIA	
	MILLENNIUM	

版权所有 © 华为云计算技术有限公司

表 1-68 用于时间截断和时间四舍五入的参数

类别	模式	描述
分钟	M	截断或四舍五入日期/时间，精确到分钟（00~59）
	MI	
	MIN	
	MINUTE	
小时	H	截断或四舍五入日期/时间，精确到小时（00~23）
	HH	
	HOUR	
	HR	
	HH12	
	HH24	
天	DD	截断或四舍五入日期/时间，精确到天（01-01~12-31）
	DDD	
	J	
ISO周	IW	截断或四舍五入日期/时间，精确到周（本周的第一天,第一天为周一）
周	DAY	截断或四舍五入日期/时间，精确到周（本周的第一天,第一天为周日）
	DY	
	D	
月周	W	截断或四舍五入日期/时间，精确到周（本周的第一天,第一天为本月第一天的周数）
年周	WW	截断或四舍五入日期/时间，精确到周（本周的第一天,第一天为本年第一天的周数）
月	MM	截断或四舍五入日期/时间，精确到月（本月的第一天）
	MON	
	MONTH	
	RM	
季度	Q	截断或四舍五入日期/时间，精确到季度（本季度的第一天）
	QTR	
	QUARTER	

版权所有 © 华为云计算技术有限公司

类别	模式	描述
年	Y	截断或四舍五入日期/时间，精确到年（本年的第一天）
	YEAR	
	YR	
	YYYY	
	SYYYY	
	YYY	
	YY	
	SYEAR	
十年	DEC	截断或四舍五入日期/时间，精确到十年（本十年的第一天）
	DECADE	
世纪	C	截断或四舍五入日期/时间，精确到世纪（本世纪的第一天）
	CC	
	CENT	
	CENTURY	
	SCC	
千年	MIL	截断或四舍五入日期/时间，精确到千年（本千年的第一天）
	MILLENNIA	
	MILLENNIUM	

说明

表1-68中行为仅在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。

- timestamp_diff(text, timestamp, timestamp)

  描述：计算两个日期时间之间的差值，截取到参数text指定的精度。

  返回值类型：bigint

  示例：

```
gaussdb=# SELECT timestamp_diff('year','2018-01-01','2020-04-01');
 timestamp_diff
----------------
              2
(1 row)
gaussdb=# SELECT timestamp_diff('month','2018-01-01','2020-04-01');
 timestamp_diff
----------------
             27
(1 row)
gaussdb=# SELECT timestamp_diff('quarter','2018-01-01','2020-04-01');
 timestamp_diff
```


版权所有 © 华为云计算技术有限公司

```
----------------
           9
(1 row)
gaussdb=# SELECT timestamp_diff('week','2018-01-01','2020-04-01');
 timestamp_diff
----------------
         117
(1 row)
gaussdb=# SELECT timestamp_diff('day','2018-01-01','2020-04-01');
 timestamp_diff
----------------
         821
(1 row)
gaussdb=# SELECT timestamp_diff('hour','2018-01-01 10:10:10','2018-01-01 12:12:12');
 timestamp_diff
----------------
           2
(1 row)
gaussdb=# SELECT timestamp_diff('minute','2018-01-01 10:10:10','2018-01-01 12:12:12');
 timestamp_diff
----------------
         122
(1 row)
gaussdb=# SELECT timestamp_diff('second','2018-01-01 10:10:10','2018-01-01 10:12:12');
 timestamp_diff
----------------
         122
(1 row)
gaussdb=# SELECT timestamp_diff('microsecond','2018-01-01 10:10:10','2018-01-01 10:12:12');
 timestamp_diff
----------------
    122000000
(1 row)
```

TIMESTAMPDIFF

TIMESTAMPDIFF(unit , timestamp_expr1, timestamp_expr2)

描述：该函数用于计算两个时间之间(timestamp_expr2-timestamp_expr1)的差值，并以unit形式返回结果。等效于timestamp_diff(text, timestamp, timestamp)。

参数：timestamp_expr1、timestamp_expr2为时间类型表达式、text、datetime、date或time等类型。unit表示的是两个日期差的单位。

返回值类型：bigint

说明

该函数仅在MYSQL模式数据库中（即sql_compatibility = 'MYSQL' 时）有效。

- 当参数b_format_version值为5.7、b_format_dev_version值为s1时，实际调用的函数为b_timestampdiff。
- 当未设置参数b_format_version和b_format_dev_version时，实际调用的函数为timestamp_diff，可以用“\df b_timestampdiff”等指令查询函数的详细入参与返回值信息。

- year

  年份。

  ```
  gaussdb=# SELECT TIMESTAMPDIFF(YEAR, '2018-01-01', '2020-01-01');
   timestamp_diff
  ----------------
             2
  (1 row)
  ```

- quarter

版权所有 © 华为云计算技术有限公司

季度。

```
gaussdb=# SELECT TIMESTAMPDIFF(QUARTER, '2018-01-01', '2020-01-01');
 timestamp_diff
----------------
              8
(1 row)
```

- month

  月份。

```
gaussdb=# SELECT TIMESTAMPDIFF(MONTH, '2018-01-01', '2020-01-01');
 timestamp_diff
----------------
             24
(1 row)
```

- week

  星期。

```
gaussdb=# SELECT TIMESTAMPDIFF(WEEK, '2018-01-01', '2020-01-01');
 timestamp_diff
----------------
            104
(1 row)
```

- day

  天。

```
gaussdb=# SELECT TIMESTAMPDIFF(DAY, '2018-01-01', '2020-01-01');
 timestamp_diff
----------------
            730
(1 row)
```

- hour

  小时。

```
gaussdb=# SELECT TIMESTAMPDIFF(HOUR, '2020-01-01 10:10:10', '2020-01-01 11:11:11');
 timestamp_diff
----------------
              1
(1 row)
```

- minute

  分钟。

```
gaussdb=# SELECT TIMESTAMPDIFF(MINUTE, '2020-01-01 10:10:10', '2020-01-01 11:11:11');
 timestamp_diff
----------------
             61
(1 row)
```

- second

  秒。

```
gaussdb=# SELECT TIMESTAMPDIFF(SECOND, '2020-01-01 10:10:10', '2020-01-01 11:11:11');
 timestamp_diff
----------------
           3661
(1 row)
```

- microseconds

  秒域（包括小数部分）乘以1,000,000。

```
gaussdb=# SELECT TIMESTAMPDIFF(MICROSECOND, '2020-01-01 10:10:10.000000', '2020-01-01
10:10:10.111111');
 timestamp_diff
----------------
```


版权所有 © 华为云计算技术有限公司

```
       111111
(1 row)
```

EXTRACT

EXTRACT(field FROM source)

描述：extract函数从日期或时间的数值里提取子域，比如年、小时等。

参数：

- source是一个timestamp、time或interval类型的值表达式（类型为date的表达式可以转换为timestamp，因此也可使用）。
- field是一个标识符或者字符串，它指定从源数据中提取的域。

返回值类型：double precision

field的取值范围如下所示。

- century

  世纪。

  第一个世纪从0001-01-01 00:00:00 AD开始。这个定义适用于所有使用阳历的国家。没有0世纪，直接从公元前1世纪到公元1世纪。

  ```
  gaussdb=# SELECT EXTRACT(CENTURY FROM TIMESTAMP '2000-12-16 12:21:13');
   date_part
  -----------
         20
  (1 row)
  ```

- day
  - 如果source为timestamp，表示月份里的日期（1~31）。

    ```
    gaussdb=# SELECT EXTRACT(DAY FROM TIMESTAMP '2001-02-16 20:38:40');
     date_part
    -----------
          16
    (1 row)
    ```

  - 如果source为interval，表示天数。

    ```
    gaussdb=# SELECT EXTRACT(DAY FROM INTERVAL '40 days 1 minute');
     date_part
    -----------
          40
    (1 row)
    ```

- decade

  年份除以10。

  ```
  gaussdb=# SELECT EXTRACT(DECADE FROM TIMESTAMP '2001-02-16 20:38:40');
   date_part
  -----------
        200
  (1 row)
  ```

- dow

  每周的星期几，星期天（0）到星期六（6）。

  ```
  gaussdb=# SELECT EXTRACT(DOW FROM TIMESTAMP '2001-02-16 20:38:40');
   date_part
  -----------
          5
  (1 row)
  ```

- doy

  一年的第几天（1~365/366）。

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT EXTRACT(DOY FROM TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
        47
(1 row)
```

- epoch
  - 如果source为timestamp with time zone，表示自1970-01-01 00:00:00-00 UTC以来的秒数（结果可能是负数）；

    如果source为date和timestamp，表示自1970-01-01 00:00:00-00当地时间以来的秒数；

    如果source为interval，表示时间间隔的总秒数。

    ```
    gaussdb=# SELECT EXTRACT(EPOCH FROM TIMESTAMP WITH TIME ZONE '2001-02-16
    20:38:40.12-08');
      date_part
    --------------
     982384720.12
    (1 row)
    gaussdb=# SELECT EXTRACT(EPOCH FROM INTERVAL '5 days 3 hours');
     date_part
    -----------
        442800
    (1 row)
    ```

  - 将epoch值转换为时间戳的方法。

    ```
    gaussdb=# SELECT TIMESTAMP WITH TIME ZONE 'epoch' + 982384720.12 * INTERVAL '1
    second' AS RESULT;
              result
    ---------------------------
     2001-02-17 12:38:40.12+08
    (1 row)
    ```

- hour

  小时域（0~23）。

  ```
  gaussdb=# SELECT EXTRACT(HOUR FROM TIMESTAMP '2001-02-16 20:38:40');
   date_part
  -----------
          20
  (1 row)
  ```

- isodow

  一周的第几天（1~7）。星期一为1，星期天为7。

  说明

  除了星期天外，isodow的值都与dow相同。

  ```
  gaussdb=# SELECT EXTRACT(ISODOW FROM TIMESTAMP '2001-02-18 20:38:40');
   date_part
  -----------
           7
  (1 row)
  ```

- isoyear

  日期中的ISO 8601标准年（不适用于间隔）。

  ISO年的第一周以星期一开始，且必须包含1月4日。因此1月初或12月下旬的日期，其对应的ISO年可能与公历年不同。详细信息请参见**week**。

  ```
  gaussdb=# SELECT EXTRACT(ISOYEAR FROM DATE '2006-01-01');
   date_part
  -----------
        2005
  (1 row)
  gaussdb=# SELECT EXTRACT(WEEK FROM TIMESTAMP '2006-01-01 00:00:40');
   date_part
  ```


版权所有 © 华为云计算技术有限公司

```
-----------
       52
(1 row)
gaussdb=# SELECT EXTRACT(ISOYEAR FROM DATE '2006-01-02');
 date_part
-----------
      2006
(1 row)
gaussdb=# SELECT EXTRACT(WEEK FROM TIMESTAMP '2006-01-02 00:00:40');
 date_part
-----------
         1
(1 row)
```

- microseconds

  秒域（包括小数部分）乘以1,000,000。

```
gaussdb=# SELECT EXTRACT(MICROSECONDS FROM TIME '17:12:28.5');
 date_part
-----------
  28500000
(1 row)
```

- millennium

  千年。

  20世纪（19xx年）里面的年份在第二个千年里。第三个千年从2001年1月1日零时开始。

```
gaussdb=# SELECT EXTRACT(MILLENNIUM FROM TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
         3
(1 row)
```

- milliseconds

  秒域（包括小数部分）乘以1000。请注意它包括完整的秒。

```
gaussdb=# SELECT EXTRACT(MILLISECONDS FROM TIME '17:12:28.5');
 date_part
-----------
     28500
(1 row)
```

- minute

  分钟域（0~59）。

```
gaussdb=# SELECT EXTRACT(MINUTE FROM TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
        38
(1 row)
```

- month

  如果source为timestamp，表示一年里的月份数（1~12）。

```
gaussdb=# SELECT EXTRACT(MONTH FROM TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
         2
(1 row)
```

  如果source为interval，表示月的数目，然后对12取模（0~11）。

```
gaussdb=# SELECT EXTRACT(MONTH FROM INTERVAL '2 years 13 months');
 date_part
-----------
         1
(1 row)
```

- quarter

版权所有 © 华为云计算技术有限公司

该天所在的该年的季度（1~4）。

```
gaussdb=# SELECT EXTRACT(QUARTER FROM TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
         1
(1 row)
```

- second

  秒域，包括小数部分（0~59）。

```
gaussdb=# SELECT EXTRACT(SECOND FROM TIME '17:12:28.5');
 date_part
-----------
      28.5
(1 row)
```

- timezone

  与UTC的时区偏移量，单位为秒。正数对应UTC以东的时区，负数对应UTC以西的时区。

- timezone_hour

  时区偏移量的小时部分。

- timezone_minute

  时区偏移量的分钟部分。

- week

  该日期在所属年份中是第几周。ISO 8601定义一年的第一周包含该年的一月四日（ISO-8601 的周从星期一开始）。换言之，一年的第一个星期四在第一周。

  在ISO定义里，一月的开始几天可能是前一年的第52或者第53周，十二月的后几天可能是下一年第一周。比如，2006-01-01是2005年的第52周，而2006-01-02是2006年的第1周。建议**isoyear**字段和week一起使用以得到一致的结果。

```
gaussdb=# SELECT EXTRACT(ISOYEAR FROM DATE '2006-01-01');
 date_part
-----------
      2005
(1 row)
gaussdb=# SELECT EXTRACT(WEEK FROM TIMESTAMP '2006-01-01 00:00:40');
 date_part
-----------
        52
(1 row)
gaussdb=# SELECT EXTRACT(ISOYEAR FROM DATE '2006-01-02');
 date_part
-----------
      2006
(1 row)
gaussdb=# SELECT EXTRACT(WEEK FROM TIMESTAMP '2006-01-02 00:00:40');
 date_part
-----------
         1
(1 row)
```

- year

  年份域。

```
gaussdb=# SELECT EXTRACT(YEAR FROM TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
      2001
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

该函数在sql_compatibility = 'ORA'时：

- GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型，此时unit仅支持年(YEAR)、月(MONTH)、日(DAY)，指定其他unit会报错。
- 入参为datea类型时，无论是否对GUC参数mapping_date_to_datea的值进行设置（on或off），此时unit仅支持年(YEAR)、月(MONTH)、日(DAY)，指定其他unit会报错。

date_part

date_part('field', source)

描述：date_part函数是基于传统的Ingres函数实现的（该函数等效于SQL标准函数extract）。

参数：field参数是一个字符串，而不是一个名称。有效的field与extract一样，详细信息请参见**EXTRACT**。

示例：

```
gaussdb=# SELECT date_part('day', TIMESTAMP '2001-02-16 20:38:40');
 date_part
-----------
        16
(1 row)
gaussdb=# SELECT date_part('hour', INTERVAL '4 hours 3 minutes');
 date_part
-----------
         4
(1 row)
```

表1-69显示了可以用于格式化日期和时间值的格式。

表 1-69 用于日期/时间格式化的格式

类别	格式	描述
小时	HH	一天的小时数（01~12）
	HH12	一天的小时数（01~12）
	HH24	一天的小时数（00~23）
分钟	MI	分钟（00~59）
秒	SS	秒（00~59）
	FF	微秒（000000~999999）
	FF1	微秒（0~9）
	FF2	微秒（00~99）
	FF3	微秒（000~999）
	FF4	微秒（0000~9999）
	FF5	微秒（00000~99999）

版权所有 © 华为云计算技术有限公司

类别	格式	描述
	FF6	微秒（000000~999999）
	SSSSS	午夜后的秒（0~86399）
上、下午	AM或A.M.	上午标识
	PM或P.M.	下午标识
年	Y,YYY	带逗号的年（4和更多位）
	SYYYY	公元前四位年
	YYYY	年（4和更多位）
	YYY	年的后三位
	YY	年的后两位
	Y	年的最后一位
	IYYY	ISO年（4位或更多位）
	IYY	ISO年的最后三位
	IY	ISO年的最后两位
	I	ISO年的最后一位
	RR	年的后两位（可在21世纪存储20世纪的年份）
	RRRR	可接收4位年或两位年。若是两位，则和RR的返回值相同，若是四位，则和YYYY相同。
	• BC或B.C. • AD或A.D.	纪元标识。BC（公元前），AD（公元后）
月	MONTH	全长大写月份名（空白填充为9字符）
	MON	大写缩写月份名（3字符）
	MM	月份数（01~12）
	RM	罗马数字的月份（I-XII；I=JAN）（大写）
天	DAY	全长大写日期名（空白填充为9字符）
	DY	缩写大写日期名（3字符）
	DDD	一年里的日（001~366）
	DD	一个月里的日（01~31）
	D	一周里的日（1~7；周日是1）
周	W	一个月里的周数（1~5）（第一周从该月第一天开始）

版权所有 © 华为云计算技术有限公司

类别	格式	描述
	WW	一年里的周数（1~53）（第一周从该年的第一天开始）
	IW	ISO一年里的周数（第一个星期四在第一周里）
世纪	CC	世纪（2位）（21 世纪从 2001-01-01 开始）
儒略日	J	儒略日（自公元前4713年1月1日来的天数）
季度	Q	季度

在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下新增用于格式化日期和时间值的模式：

表 1-70 新增用于格式化日期和时间值的模式

类别	模式	描述
世纪	SCC	世纪标识，公元前会显示-
年	SYYYY	返回数字型年，公元前会显示-
	RR	返回日期的2位年份
	RRRR	返回日期的4位年份
	YEAR	返回字符型年
	SYEAR	返回字符型年，公元前会显示-
日期格式	DL	返回指定长日期形式
	DS	返回指定短日期
	TS	返回指定时间格式
秒	FF7	微秒（0000000~9999990）
	FF8	微秒（00000000~99999900）
	FF9	微秒（000000000~999999000）

版权所有 © 华为云计算技术有限公司

说明

- 该函数在sql_compatibility = 'ORA'时：
  - GUC参数mapping_date_to_datea值为on的情况下，数据库会将入参中的date类型解析为datea类型，此时格式仅支持年(YEAR)、月(MONTH)、日(DAY)，指定其他格式会报错。
  - 入参为datea类型时，无论是否对GUC参数mapping_date_to_datea的值进行设置（on或off），此时格式仅支持年(YEAR)、月(MONTH)、日(DAY)，指定其他格式会报错。
- 此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：
  - to_date, to_timestamp函数支持FX模式（输入和模式严格对应），支持X模式（小数点）。
  - 输入模式不能出现超过一次，表示相同信息的模式不能同时出现。如SYYYY和BC不能同时出现。
  - 模式大小写不敏感。
  - 建议输入和模式之间使用分隔符，否则不保证行为与O完全一致。
  - GaussDB采用严格匹配策略，fmt中的每个格式元素都必须在输入字符串中有对应内容，否则报错；Oracle在字符串到日期转换时，对格式模型中的可选元素（如FF）采用宽松匹配策略，若输入字符串中缺少该部分，会自动跳过并继续匹配后续元素。该差异影响to_date、to_timestamp、to_timestamp_tz函数在不指定fmt或fmt中包含FF等可选格式元素场景下的行为。
- 上表中RR计算年的规则如下：
  - 输入的两位年份在00~49之间：

    当前年份的后两位在00~49之间，返回值年份的前两位和当前年份的前两位相同。

    当前年份的后两位在50~99之间，返回值年份的前两位是当前年份的前两位加1。
  - 输入的两位年份在50~99之间：

    当前年份的后两位在00~49之间，返回值年份的前两位是当前年份的前两位减1。

    当前年份的后两位在50~99之间，返回值年份的前两位和当前年份的前两位相同。

time_window

time_window(window_width interval, time timestamptz, [origin timestamptz])

描述：按照指定的时间窗口大小对指定时间戳列与原始时间戳的差值进行取模运算。

参数：如表1-71所示。

返回值类型：timestamptz

表 1-71 time_window 参数说明

参数名称	类型	描述
window_width	interval	时间窗口。
time	timestamp(tz)	时间戳列。
origin	timestamp(tz)	原始的时间戳。默认为2000-01-03 00:00:00

版权所有 © 华为云计算技术有限公司

说明

- window_width单位可以为microsecond、millisecond、second、min、hour、day、week以及正数（默认单位为day）。
- window_width以及time参数类型正确不能为空值。

示例：

```
gaussdb=# SELECT
    time_window(interval '1 min', time),
    time_window(interval '3 min', time,  '2019-07-11 00:02:05'::timestamptz)
        FROM
    (VALUES ('2019-07-12 00:00:00'::timestamptz),
          ('2019-07-12 00:01:01'::timestamptz)) v(time);

     time_window       |      time_window
-----------------------+-----------------------
 2019-07-12 00:00:00+08 | 2019-07-11 23:59:05+08
 2019-07-12 00:01:00+08 | 2019-07-11 23:59:05+08
(2 rows)
```

1.6.9 类型转换函数

函数类型转换

- cash_words(money)

  描述：类型转换函数，将money转换为text后输出。

  示例：

  ```
  gaussdb=# SELECT cash_words('1.23');
              cash_words
  -----------------------------------
   One dollar and twenty three cents
  (1 row)
  ```

- convert(expr, type)

  描述：类型转换函数，将expr转换成type指定的类型。

  参数：第一个参数为一个任意值，第二个参数为类型名称。

  返回值类型：返回值类型为输入的type类型。

  示例：

  ```
  gaussdb=# SELECT convert(12.5, text);
   text
  ------
   12.5
  (1 row)
  ```

  说明

  此函数仅在MYSQL模式数据库中生效。

- cast(x as y [DEFAULT z ON CONVERSION ERROR][,fmt])

  描述：类型转换函数，将x转换成y指定的类型。当sql_compatibility = 'MYSQL'时，设置参数b_format_version = '5.7'、b_format_dev_version = 's1'后，若y是char类型，则该函数会将x转换成varchar类型。

  参数：

  - x为该函数中待转换的数据，其类型和值需要支持转换为目标类型y。
  - DEFAULT z ON CONVERSION ERROR为第一个参数的可选语法，当尝试将x转换成y指定的类型失败时，则将z转换成y指定的类型。

版权所有 © 华为云计算技术有限公司

- fmt为可选参数，当y是以下数据类型时可以指定fmt参数：
  - int1/int2/int4/int8/int16/float4/float8/numeric：则fmt的作用与to_number(expr [,fmt])函数相同。
  - date/timestamp/timestamp with time zone/datea（只在ORA兼容性下生效）：则fmt的作用与to_date(string [,fmt])/to_timestamp(string [,fmt]) /to_timestamp_tz(string [,fmt])函数相同。

示例：

```
gaussdb=# SELECT cast('22-oct-1997' as timestamp);
     timestamp
---------------------
 1997-10-22 00:00:00
(1 row)

gaussdb=# SELECT cast('22-ocX-1997' as timestamp DEFAULT '22-oct-1997' ON CONVERSION
ERROR, 'DD-Mon-YYYY');
     timestamp
---------------------
 1997-10-22 00:00:00
(1 row)

gaussdb=# CREATE DATABASE gaussdb_m  WITH dbcompatibility  'MYSQL';
gaussdb=# \c gaussdb_m
-- 设置兼容版本控制参数
gaussdb_m=# SET b_format_version='5.7';
gaussdb_m=# SET b_format_dev_version='s1';
gaussdb_m=# SELECT cast('aaa' as char);
 varchar
---------
 aaa
(1 row)
```

说明

- 此函数在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，支持DEFAULT z ON CONVERSION ERROR的可选语法及fmt可选参数。
- 在ORA兼容模式下，且参数mapping_date_to_datea打开时，若函数参数fmt缺省，x转换为datea类型时需要符合nls_date_format设置的时间格式。
- 在ORA兼容模式下，且参数mapping_date_to_datea打开时，若函数参数fmt缺省，datea类型转为字符类型时，将按照nls_date_format设置的格式转为字符类型。
- 当y为datea类型，x的类型仅支持text、bpchar、varchar、clob、nvarchar2、timestamp、timestamp with time zone类型。

- cast(x AS {SIGNED | UNSIGNED} [INT | INTEGER])

  描述：类型转换函数，将x转换成BIGINT SIGNED或BIGINT UNSIGNED类型。

  参数：x是该函数中待转换的数据，其类型和值需要支持转换为目标类型INT。

  返回值类型：BIGINT SIGNED或BIGINT UNSIGNED

  示例：

```
gaussdb=# SELECT CAST(12 AS UNSIGNED);
 uint8
-------
 12
(1 row)
```

- hextoraw(text)

  描述：将一个十六进制构成的字符串转换为raw类型。

  返回值类型：raw

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT hextoraw('7D');
 hextoraw
----------
 7D
(1 row)
```

- numtoday(numeric)

  描述：将数字类型的值转换为指定格式的时间戳。

  返回值类型：timestamp

  示例：

```
gaussdb=# SELECT numtoday(2);
 numtoday
----------
 2 days
(1 row)
```

- rawtohex(string)

  描述：将一个二进制构成的字符串转换为十六进制的字符串。结果为输入字符的ASCII码，以十六进制表示。

  返回值类型：varchar

  示例：

```
gaussdb=# SELECT rawtohex('1234567');
    rawtohex
-----------------
 31323334353637
(1 row)
```

- to_blob(raw)

  描述：将RAW类型转成BLOB类型。

  返回值类型：blob

  示例：

```
gaussdb=# SELECT to_blob('0AADD343CDBBD'::RAW(10));
    to_blob
-----------------
 00AADD343CDBBD
(1 row)
```

  说明

  在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_blob函数。

- to_bigint(varchar)

  描述：将字符类型转换为bigint类型。

  返回值类型：bigint

  示例：

```
gaussdb=# SELECT to_bigint('123364545554455');
    to_bigint
-----------------
 123364545554455
(1 row)
```

- to_binary_double(expr)

  描述：将expr转换成float8类型的值。

  参数：expr支持number、float4、float8数值类型，以及可以隐式转换为数值类型的字符串。

  返回值类型：float8

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT to_binary_double('12345678');
 to_binary_double
------------------
         12345678
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_double函数。

- to_binary_double(expr, fmt)

  描述：将expr经过指定的fmt匹配后转换成float8类型的数值。

  参数：expr/fmt支持char、nchar、varchar2、nvarchar2类型的字符串，expr还支持可以隐式转换为字符串类型的数值类型。

  返回值类型：float8

  示例：

```
gaussdb=# SELECT to_binary_double('1,2,3', '9,9,9');
 to_binary_double
------------------
              123
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_double函数。

- to_binary_double(expr DEFAULT return_value ON CONVERSION ERROR)

  描述：将expr转换成float8类型的值，若失败则返回默认值return_value。

  参数：expr支持number、float4、float8数值类型，以及可以隐式转换为字符串的数值类型。当expr为非数值类型或非字符串类型时，则会报错。

  返回值类型：float8

  示例：

```
gaussdb=# SELECT to_binary_double(1e2 DEFAULT 12 ON CONVERSION ERROR);
 to_binary_double
------------------
              100
(1 row)

gaussdb=# SELECT to_binary_double('aa' DEFAULT 12 ON CONVERSION ERROR);
 to_binary_double
------------------
               12
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_double函数。

- to_binary_double(expr DEFAULT return_value ON CONVERSION ERROR, fmt)

  描述：将expr经过指定的fmt匹配后转换成float8类型的数值，若失败则返回默认值return_value。

  参数：expr/fmt支持char、nchar、varchar2、nvarchar2类型的字符串，expr还支持可以隐式转换为字符串的数值类型。

  返回类型：float8

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT to_binary_double('12-' DEFAULT 10 ON CONVERSION ERROR, '99S');
 to_binary_double
------------------
              -12
(1 row)

gaussdb=# SELECT to_binary_double('aa-' DEFAULT 12 ON CONVERSION ERROR, '99S');
 to_binary_double
------------------
               12
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_double函数。

- to_binary_float(expr)

  描述：将expr转换成float4类型的值。

  参数：expr支持number、float4、float8数值类型，以及可以隐式转换为数值类型的字符串。

  返回值类型：float4

  示例：

```
gaussdb=# SELECT to_binary_float('12345678');
 to_binary_float
------------------
     1.23457e+07
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_float函数。

- to_binary_float(expr, fmt)

  描述：将expr经过指定的fmt匹配后转换成float4类型的数值。

  参数：expr/fmt支持char、nchar、varchar2、nvarchar2类型的字符串，expr还支持可以隐式转换为字符串类型的数值类型。

  返回值类型：float4

  示例：

```
gaussdb=# SELECT to_binary_float('1,2,3', '9,9,9');
 to_binary_float
------------------
             123
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_float函数。

- to_binary_float(expr DEFAULT return_value ON CONVERSION ERROR)

  描述：将expr转换成float4类型的值，若失败则返回默认值return_value。

  参数：expr支持number、float4、float8数值类型，以及可以隐式转换为字符串的数值类型，当expr为非数值类型或非字符串类型时，则会报错。

  返回值类型：float4

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT to_binary_float(1e2 DEFAULT 12 ON CONVERSION ERROR);
 to_binary_float
-----------------
             100
(1 row)

gaussdb=# SELECT to_binary_float('aa' DEFAULT 12 ON CONVERSION ERROR);
 to_binary_float
-----------------
              12
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_float函数。

- to_binary_float(expr DEFAULT return_value ON CONVERSION ERROR, fmt)

  描述：将expr经过指定的fmt匹配后转换成float4类型的数值，若失败则返回默认值return_value。

  参数：expr/fmt支持char、nchar、varchar2、nvarchar2类型的字符串，expr还支持可以隐式转换为字符串类型的数值类型。

  返回类型：float4

  示例：

```
gaussdb=# SELECT to_binary_float('12-' DEFAULT 10 ON CONVERSION ERROR, '99S');
 to_binary_float
-----------------
             -12
(1 row)

gaussdb=# SELECT to_binary_float('aa-' DEFAULT 12 ON CONVERSION ERROR, '99S');
 to_binary_float
-----------------
              12
(1 row)
```

说明

在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_binary_float函数。

- to_char(datetime/interval [, fmt])

  描述：将一个DATE、DATEA（只在ORA兼容性下生效）、TIMESTAMP、TIMESTAMP WITH TIME ZONE或者TIMESTAMP WITH LOCAL TIME ZONE类型的DATETIME或者INTERVAL值按照fmt（用于格式化日期和时间值的模板）指定的格式转换为TEXT类型。

  - fmt可以为以下几类：日期、时间、星期、季度和世纪。每类都可以有不同的模板，模板之间可以合理组合，常见的模板有：HH、MI、SS、YYYY、MM、DD，详情请参见**表1-69**。
  - fmt可以有修饰词，常用的修饰词是FM，可以用来判断排查前导的零或尾随的空白。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char(current_timestamp,'HH12:MI:SS');
 to_char
----------
 10:19:26
(1 row)
gaussdb=# SELECT to_char(current_timestamp,'FMHH12:FMMI:FMSS');
 to_char
```


版权所有 © 华为云计算技术有限公司

```
----------
 10:19:46
(1 row)
-- ORA兼容模式下mapping_date_to_datea打开，current_date返回datea类型。
test_db_a=# SELECT to_char(current_date);
  to_char
------------
 05-11-2024
(1 row)
test_db_a=# SELECT to_char(current_date, 'YYYY-MM-DD HH24:MI:SS AD');
        to_char
------------------------
 2024-11-05 14:07:21 AD
(1 row)
```

说明

在ORA兼容模式下，且参数mapping_date_to_datea打开时，若函数参数fmt缺省，datea类型转为text类型将按照nls_date_format设置的时间格式转为字符串。

- to_char(double precision/real, text)

  描述：将浮点类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char(125.8::real, '999D99');
 to_char
---------
  125.80
(1 row)
```

- to_char(numeric/smallint/integer/bigint/double precision/real[, fmt])

  描述：将一个整型或者浮点类型的值转换为指定格式的字符串。

  - 可选参数fmt可以为以下几类：十进制字符、“分组”符、正/负号和货币符号，每类都可以有不同的模板，模板之间可以合理组合，常见的模板有：9、0、,（千分隔符）、.（小数点），详情如**表1-72**所示。
  - 模板可以有类似FM的修饰词，但FM不判断排查由模板0指定而输出的0。
  - 要将整型类型的值转换成对应16进制值的字符串，使用模板X或x。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char(1485,'9,999');
 to_char
---------
  1,485
(1 row)
gaussdb=# SELECT to_char( 1148.5,'9,999.999');
  to_char
------------
  1,148.500
(1 row)
gaussdb=# SELECT to_char(148.5,'990999.909');
   to_char
-------------
    0148.500
(1 row)
gaussdb=# SELECT to_char(123,'XXX');
 to_char
---------
   7B
(1 row)
```


版权所有 © 华为云计算技术有限公司

表 1-72 number 类型 fmt 参数

模式	描述
, (comma)	分组（千）分隔符。
. (period)	小数点。
$	指定位置输出$。
0	带前导零的值。
9	带有指定数值位数的值。
B	当整数部分是0时返回空格。
C	货币符号（使用区域设置）。
D	小数点（使用区域设置）。
EEEE	科学计数法。
G	分组分隔符（使用区域设置）。
L	货币符号（使用区域设置）。
MI	在指明的位置的负号（如果数字<0）。
PR	尖括号内负值。
RN	罗马数字（输入在1和3999之间）。
S	带符号的数值（使用区域设置）。
TM	标准数值与科学计数法。
TM9	标准数值与科学计数法。
TME	标准数值与科学计数法。
U	双货币符号（使用区域设置）。
V	移动指定位（小数）。
PL	在指明的位置的正号（如果数字 > 0）。
SG	在指明的位置的正/负号。
TH或th	序数后缀。

说明

此函数在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，支持$、C、TM、TM9、TME、U格式。同时在该参数下，不支持TH、PL、SG格式的fmt。

- to_char(interval, text)

  描述：将时间间隔类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT to_char(interval '15h 2m 12s', 'HH24:MI:SS');
 to_char
----------
 15:02:12
(1 row)
```

- to_char(integer, text)

  描述：将整数类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char(125, '999');
 to_char
---------
  125
(1 row)
```

- to_char(set)

  描述：将SET类型的值转换为字符串。分布式暂不支持SET数据类型。

  返回值类型：text

- to_char(numeric, text)

  描述：将数字类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char(-125.8, '999D99S');
 to_char
---------
 125.80-
(1 row)
```

- to_char(string)

  描述：将CHAR、VARCHAR、VARCHAR2、CLOB类型转换为TEXT类型。如使用该函数对CLOB类型进行转换，且待转换CLOB类型的值超出目标类型的范围，则会报错。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char('01110');
 to_char
---------
 01110
(1 row)
```

- to_nvarchar2(numeric)

  描述：转换为nvarchar2类型。

  返回值类型：nvarchar2

- to_char(timestamp, text)

  描述：将时间戳类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_char(current_timestamp, 'HH12:MI:SS');
 to_char
----------
 10:55:59
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

- 在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，to_char函数会对错误的fmt报错。
- 如果不是在兼容模式下，to_char函数对于错误的fmt会原样输出，如fmt为FF10，会匹配到FF1进行格式化输出，然后原样输出0。

- to_nchar(datetime/interval [, fmt])

  描述：将一个DATE、DATEA（只在ORA兼容性下生效）、TIMESTAMP、TIMESTAMP WITH TIME ZONE或者TIMESTAMP WITH LOCAL TIME ZONE类型的DATETIME或者INTERVAL值按照fmt指定的格式转换为TEXT类型。

  - 可选参数fmt可以为以下几类：日期、时间、星期、季度和世纪。每类都可以有不同的模板，模板之间可以合理组合，常见的模板有：HH、MI、SS、YYYY、MM、DD，详情请参见**表1-69**。
  - 模板可以有修饰词，常用的修饰词是FM，可以用来判断排查前导的零或尾随的空白。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_nchar(current_timestamp,'HH12:MI:SS');
 to_nchar
----------
 10:19:26
(1 row)
gaussdb=# SELECT to_nchar(current_timestamp,'FMHH12:FMMI:FMSS');
 to_nchar
----------
 10:19:46
(1 row)
```

  说明

  - 在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_nchar函数。
  - 在ORA兼容模式下，且参数mapping_date_to_datea打开时，若函数参数fmt缺省，datea类型转为text类型将按照nls_date_format设置的时间格式转为字符串。

- to_nchar(double precision/real, text)

  描述：将浮点类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_nchar(125.8::real, '999D99');
 to_nchar
----------
  125.80
(1 row)
```

- to_nchar(numeric/smallint/integer/bigint/double precision/real[, fmt])

  描述：将一个整型或者浮点类型的值转换为指定格式的字符串。

  - 可选参数fmt可以为以下几类：十进制字符、“分组”符、正/负号和货币符号，每类都可以有不同的模板，模板之间可以合理组合，常见的模板有：9、0、,（千分隔符）、.（小数点），详情请参见**表1-72**。
  - 模板可以有类似FM的修饰词，但FM不判断排查由模板0指定而输出的0。
  - 要将整型类型的值转换成对应16进制值的字符串，使用模板X或x。

  返回值类型：text

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT to_nchar(1485,'9,999');
 to_nchar
----------
  1,485
(1 row)
gaussdb=# SELECT to_nchar( 1148.5,'9,999.999');
  to_nchar
------------
  1,148.500
(1 row)
gaussdb=# SELECT to_nchar(148.5,'990999.909');
   to_nchar
--------------
    0148.500
(1 row)
gaussdb=# SELECT to_nchar(123,'XXX');
 to_nchar
----------
   7B
(1 row)
```

说明

此函数仅在ORA兼容模式中（即sql_compatibility='ORA'）、且参数a_format_version值为10c和a_format_dev_version值为s2时生效。开启参数时支持$、C、TM、TM9、TME、U格式。同时在该参数下，不支持TH、PL、SG格式的fmt。

- to_nchar(interval, text)

  描述：将时间间隔类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT to_nchar(interval '15h 2m 12s', 'HH24:MI:SS');
   to_nchar
  ----------
   15:02:12
  (1 row)
  ```

- to_nchar(integer, text)

  描述：将整数类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT to_nchar(125, '999');
   to_nchar
  ----------
   125
  (1 row)
  ```

- to_nchar(set)

  描述：将SET类型的值转换为字符串。分布式暂不支持SET数据类型。

  返回值类型：text

- to_nchar(numeric, text)

  描述：将数字类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT to_nchar(-125.8, '999D99S');
   to_nchar
  ----------
   125.80-
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- to_nchar(string)

  描述：将CHAR、VARCHAR、VARCHAR2、CLOB类型转换为TEXT类型。如使用该函数对CLOB类型进行转换，且待转换CLOB类型的值超出目标类型的范围，则会报错。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_nchar('01110');
 to_nchar
----------
 01110
(1 row)
```

- to_nchar(timestamp, text)

  描述：将时间戳类型的值转换为指定格式的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT to_nchar(current_timestamp, 'HH12:MI:SS');
 to_nchar
----------
 10:55:59
(1 row)
```

  说明

  在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，支持to_nchar函数。

- to_clob(char/nchar/varchar/varchar2/nvarchar2/text/raw)

  描述：将RAW类型或者文本字符集类型CHAR、NCHAR、VARCHAR、VARCHAR2、NVARCHAR2、TEXT转成CLOB类型。

  返回值类型：clob

  示例：

```
gaussdb=# SELECT to_clob('ABCDEF'::RAW(10));
 to_clob
---------
 ABCDEF
(1 row)
gaussdb=# SELECT to_clob('hello111'::CHAR(15));
 to_clob
----------
 hello111
(1 row)
gaussdb=# SELECT to_clob('gauss123'::NCHAR(10));
 to_clob
----------
 gauss123
(1 row)
gaussdb=# SELECT to_clob('gauss234'::VARCHAR(10));
 to_clob
----------
 gauss234
(1 row)
gaussdb=# SELECT to_clob('gauss345'::VARCHAR2(10));
 to_clob
----------
 gauss345
(1 row)
gaussdb=# SELECT to_clob('gauss456'::NVARCHAR2(10));
 to_clob
----------
```


版权所有 © 华为云计算技术有限公司

```
 gauss456
(1 row)
gaussdb=# SELECT to_clob('World222!'::TEXT);
  to_clob
-----------
 World222!
(1 row)
```

- to_date(text)

  描述：将文本类型的值转换为指定格式的时间戳。在参数a_format_version值为10c、a_format_dev_version值为s1、mapping_date_to_datea值为off的情况下，按参数nls_timestamp_format所指定的格式转换；若不指定参数a_format_version和a_format_dev_version，则按照固定fmt = 'YYYY-MM-DD HH24:MI:SS'进行转换。

  - 格式一：无分隔符日期，如20150814，需要包括完整的年月日。
  - 格式二：带分隔符日期，如2014-08-14，分隔符可以是单个任意非数字字符。

  返回值类型：timestamp without time zone、datea（仅在ORA兼容下生效）

  示例：

  ```
  -- 若在ORA兼容下，需保证输入的文本类型格式满足NLS_DATE_FORMAT格式，否则会报错
  gaussdb=# SELECT to_date('2015-08-14');
        to_date
  ---------------------
   2015-08-14 00:00:00
  (1 row)
  ```

  说明

  - 该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，返回值为datea类型。
  - to_date函数为stable类型，而不是immutable，因此无法作为生成列/表达式索引。

- to_date(text, text)

  描述：将字符串类型的值转换为指定格式的日期。

  返回值类型：timestamp without time zone、datea（仅在ORA兼容下生效）

  示例：

  ```
  gaussdb=# SELECT to_date('05 Dec 2000', 'DD Mon YYYY');
        to_date
  ---------------------
   2000-12-05 00:00:00
  (1 row)
  ```

  说明

  - 该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，返回值为datea类型。
  - to_date函数为stable类型，而不是immutable，因此无法作为生成列/表达式索引。

- to_date(text [DEFAULT return_value ON CONVERSION ERROR [, fmt]])

  描述：将字符串text按fmt指定的格式转换成DATE类型的值。不指定fmt时，在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，按参数nls_timestamp_format所指定的格式转换；若未指定参数a_format_version和a_format_dev_version，则按照固定fmt = 'YYYY-MM-DD HH24:MI:SS'进行转换。

  参数：

  - text：任何计算结果为CHAR、VARCHAR2、NCHAR、NVARCHAR2、TEXT类型字符串的表达式。输入null，返回null。

版权所有 © 华为云计算技术有限公司

- DEFAULT return_value ON CONVERSION ERROR：可选参数，在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，用于当text转换DATE类型失败时指定返回值return_value。return_value可以是表达式或者绑定的变量，需要能够转换为CHAR、VARCHAR2、NCHAR、NVARCHAR2、TEXT类型或者是null。return_value转换为DATE类型数据的方法与text转换为DATE类型数据相同，如果return_value转换为DATE类型数据失败，该函数报错。
- fmt：可选参数。指定text的日期时间模型格式。缺省时，text需要符合默认的日期格式的要求。fmt指定为J时，text需要为整数。

返回值类型：timestamp without time zone、datea（只在ORA兼容下生效）

示例：

```
-- 若在ORA兼容下，若入参仅为一个文本类型时，需保证输入的文本类型格式满足NLS_DATE_FORMAT格式，否则会报错
gaussdb=# SELECT to_date('2015-08-14');
      to_date
---------------------
 2015-08-14 00:00:00
(1 row)
gaussdb=# SELECT to_date('05 Dec 2000', 'DD Mon YYYY');
      to_date
---------------------
 2000-12-05 00:00:00
(1 row)
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s1';
SET
gaussdb=# SHOW nls_timestamp_format;
   nls_timestamp_format
----------------------------
 DD-Mon-YYYY HH:MI:SS.FF AM
(1 row)
gaussdb=# SELECT to_date('12-jan-2022' DEFAULT '12-apr-2022' ON CONVERSION ERROR);
      to_date
---------------------
 2022-01-12 00:00:00
(1 row)
gaussdb=# SELECT to_date('12-ja-2022' DEFAULT '12-apr-2022' ON CONVERSION ERROR);
      to_date
---------------------
 2022-04-12 00:00:00
(1 row)
gaussdb=# SELECT to_date('2022-12-12' DEFAULT '2022-01-01' ON CONVERSION ERROR, 'yyyy-mm-dd');
      to_date
---------------------
 2022-12-12 00:00:00
(1 row)
```

说明

- 在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，年份的输入超过9999时，系统可能不报错。如to_date('99999-12-12', 'YYYY-MM-DD HH24:MI:SS')结果为9999-09-12 12:00:00。年输入超过9999时，超过4位后面的数字会解析为下一个fmt，to_timestamp同样有此限制。
- 该函数在sql_compatibility='ORA'，且GUC参数mapping_date_to_datea值为on的情况下，返回值为DATEA类型。此时to_date将字符串text按fmt指定的格式转换成DATEA类型的值。不指定fmt时，按参数nls_date_format所指定的格式转换。
- to_date函数为stable类型，而不是immutable，因此无法作为生成列/表达式索引。

- to_number(expr [, fmt])

版权所有 © 华为云计算技术有限公司

描述：将expr按指定格式转换为一个NUMBER类型的值。类型转换格式请参见**表1-75**。转换十六进制字符串为十进制数字时，最多支持16个字节的十六进制字符串转换为无符号数；同时格式字符串中不允许出现除x或X以外的其他字符，否则报错。

返回值类型：number

示例：

```
gaussdb=# SELECT to_number('12,454.8-', '99G999D9S');
 to_number
-----------
  -12454.8
(1 row)
```

- to_number(text, text)

  描述：将字符串类型的值转换为指定格式的数字。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SELECT to_number('12,454.8-', '99G999D9S');
   to_number
  -----------
    -12454.8
  (1 row)
  ```

- to_number(expr [DEFAULT return_value ON CONVERSION ERROR [, fmt]])

  描述：将字符串expr根据指定fmt格式转换成numeric类型的值。不指定fmt时，expr需要为能直接转换成numeric的字符串, 例: '123' , '1e2'。类型转换格式请参见**表1-76**。

  参数：

  - expr：支持的类型有CHAR、VARCHAR2、NCHAR、NVARCHAR2、TEXT、INT、FLOAT等可以转换成字符串类型的表达式。输入null，返回null。
  - DEFAULT return_value ON CONVERSION ERROR：可选参数，在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，用于当expr转换numeric类型失败时指定返回值return_value。return_value同expr一样可以是任何能转换成字符串的类型。return_value的转换方式和expr相同也是根据fmt格式进行转换，会先校验return_value是否会转换失败，如果失败则该函数报错。
  - fmt：可选参数。指定expr的转换格式。任意入参为null，则返回null。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SET a_format_version='10c';
  gaussdb=# SET a_format_dev_version='s1';

  gaussdb=# SELECT to_number('1e2');
   to_number
  -----------
         100
  (1 row)

  gaussdb=# SELECT to_number('123.456');
   to_number
  -----------
     123.456
  (1 row)

  gaussdb=# SELECT to_number('123', '999');
   to_number
  -----------
         123
  ```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# SELECT to_number('123-', '999MI');
 to_number
-----------
      -123
(1 row)

gaussdb=# SELECT to_number('123' DEFAULT '456-' ON CONVERSION ERROR, '999MI');
 to_number
-----------
      -456
(1 row)
```

- to_timestamp(double precision)

  描述：把Unix纪元转换成时间戳。

  返回值类型：timestamp with time zone

  示例：

```
gaussdb=# SELECT to_timestamp(1284352323);
      to_timestamp
------------------------
 2010-09-13 12:32:03+08
(1 row)
```

- to_timestamp(string [,fmt])

  描述：将字符串string按fmt指定的格式转换成时间戳类型的值。不指定fmt时，按参数nls_timestamp_format所指定的格式转换。GaussDB的to_timestamp中，

  - 如果输入的年份YYYY=0，系统报错。
  - 如果输入的年份YYYY<0，在fmt中指定SYYYY，则正确输出公元前绝对值n的年份。

  fmt中出现的字符必须与日期/时间格式化的模式相匹配，否则报错。

  返回值类型：timestamp without time zone

  示例：

```
gaussdb=# SHOW nls_timestamp_format;
    nls_timestamp_format
-----------------------------
 DD-Mon-YYYY HH:MI:SS.FF AM
(1 row)

gaussdb=# SELECT to_timestamp('12-sep-2014');
    to_timestamp
---------------------
 2014-09-12 00:00:00
(1 row)
gaussdb=# SELECT to_timestamp('12-Sep-10 14:10:10.123000','DD-Mon-YY HH24:MI:SS.FF');
      to_timestamp
-------------------------
 2010-09-12 14:10:10.123
(1 row)
gaussdb=# SELECT to_timestamp('-1','SYYYY');
      to_timestamp
------------------------
 0001-01-01 00:00:00 BC
(1 row)
gaussdb=# SELECT to_timestamp('98','RR');
    to_timestamp
---------------------
 1998-01-01 00:00:00
(1 row)
gaussdb=# SELECT to_timestamp('01','RR');
    to_timestamp
---------------------
```


版权所有 © 华为云计算技术有限公司

```
2001-01-01 00:00:00
(1 row)
```

说明

- 在参数a_format_version值为10c且a_format_dev_version值为s1的情况下fmt支持FF[7-9]，在FF[7-9]的情况下允许转换string中的对应位置长度小于等于FF紧跟的数字，但最终转换结果长度最大保留6位。
- 不支持current_timestamp函数返回结果作为string参数。
- PG兼容模式下，当string比fmt格式长时（例如：to_timestamp('2024-01-05 14:52:06.694752', 'yyyy-MM-dd hh24:mi:ss')），按照fmt格式的长度自动对string进行截取并舍弃长度超出部分。
- PG兼容模式下，开启GUC参数format_pg_to_timestamp后，如果存在不支持的格式，会自动跳过，当前版本不支持FF、FF7、FF8、FF9、RR、RRRR、SYYYY、x、X等格式；支持早于294277-01-01 00:00:00之前的时间日期和负数年份（转化成公元前）；FF1-FF6会将超出精确度的毫秒四舍五入；当时间日期中相同项冲突时返回一个格式正确的时间不会报错，具体冲突行为如**表1-73**所示。

表 1-73 to_timestamp fmt 冲突

冲突场景	描述
SSSSS与时间冲突	保留时间（分钟，小时）中的值以及SSSSS中的秒数部分的值。
W，J与时间日期冲突	优先级：J，W，mmdd按顺序进行覆盖。 W：未指定时，默认为星期三。 默认行为：W覆盖mmdd。
SSSSS与SS冲突	保留SS，舍弃SSSSS。
DDD与日期冲突	舍弃DDD。

- to_timestamp(text [DEFAULT return_value ON CONVERSION ERROR [, fmt]])

  描述：将字符串text按fmt指定的格式转换成TIMESTAMP类型的值。不指定fmt时，在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，按参数nls_timestamp_format所指定的格式转换；若未指定参数a_format_version和a_format_dev_version，则按照固定fmt = 'yyyy-mm-dd hh24-mi-ss'进行转换。

  参数：

  - text：任何计算结果为CHAR、VARCHAR2、NCHAR、NVARCHAR2、TEXT类型字符串的表达式。输入null，返回null。
  - DEFAULT return_value ON CONVERSION ERROR：可选参数，在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，用于当text转换TIMESTAMP类型失败时指定返回值return_value。return_value可以是表达式或者绑定的变量，需可以转换为CHAR、VARCHAR2、NCHAR、NVARCHAR2、TEXT类型或者是null。return_value转换为timestamp类型数据的方法与text转换为timestamp类型数据相同，如果return_value转换为timestamp类型数据失败，该函数报错。
  - fmt：可选参数。指定text的日期时间模型格式。缺省时，text需要符合默认的日期格式的要求。fmt指定为J时，text需要为整数。

  返回值类型：timestamp without time zone

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s1';
SET
gaussdb=# SELECT to_timestamp('11-Sep-11' DEFAULT '12-Sep-10 14:10:10.123000' ON
CONVERSION ERROR,'DD-Mon-YY HH24:MI:SS.FF');
   to_timestamp
---------------------
 2011-09-11 00:00:00
(1 row)
gaussdb=# SELECT to_timestamp('12-Sep-10 14:10:10.123000','DD-Mon-YY HH24:MI:SSXFF');
     to_timestamp
-------------------------
 2010-09-12 14:10:10.123
(1 row)
```

- to_timestamp(text, text)

  描述：将字符串类型的值转换为指定格式的时间戳。

  返回值类型：timestamp

  示例：

  ```
  gaussdb=# SELECT to_timestamp('05 Dec 2000', 'DD Mon YYYY');
     to_timestamp
  ---------------------
   2000-12-05 00:00:00
  (1 row)
  ```

- to_timestamp_tz(string [DEFAULT return_value ON CONVERSION ERROR] [,fmt])

  描述：将字符串string按fmt指定的格式转换成带时区时间戳类型的值。不指定fmt时，按参数nls_timestamp_tz_format所指定的格式转换。

  参数：

  - DEFAULT return_value ON CONVERSION ERROR：可选参数。当string转换成timestamp with time zone类型失败时，则将return_value转换timestamp with time zone类型。
  - fmt：可选参数。指定string的日期时间模型格式。

  返回值类型：timestamp with time zone

  示例：

  ```
  gaussdb=# SELECT to_timestamp_tz('05 DeX 2000' DEFAULT '05 Dec 2001' ON CONVERSION ERROR,
  'DD Mon YYYY');
     to_timestamp_tz
  --------------------------
   2001-12-05 00:00:00+08:00
  (1 row)
  ```

  说明

  此函数在参数a_format_version值为10c且a_format_dev_version值为s1的情况下有效。

- to_timestamp_tz(string [DEFAULT return_value ON CONVERSION ERROR], fmt, nlsparam)

  描述：将字符串string按fmt指定的格式转换成带时区时间戳类型的值。当string转换成timestamp with time zone类型失败时，则将return_value转换timestamp with time zone类型。

  参数：如**表1-74**所示。

  返回值类型：timestamp with time zone

版权所有 © 华为云计算技术有限公司

表 1-74 to_timestamp_tz(string [DEFAULT return_value ON CONVERSION ERROR], fmt, nlsparam)参数说明

参数	类型	描述
string	text	被用来转换为带时区时间戳类型的字符串。
return_value	text	当string转换成timestamp with time zone类型失败时，则将return_value转换timestamp with time zone类型。
fmt	text	指定参数string的日期时间模型格式。
nlsparam	text	指定参数string中月份和日名称所用的语言，形式为'nls_date_language=language'，目前language只支持ENGLISH和AMERICAN。正确使用nlsparam参数的结果和省略nlsparam参数的结果保持一致。

示例：

```
gaussdb=# SELECT to_timestamp_tz('05 DeX 2000' DEFAULT '05 Dec 2001' ON CONVERSION ERROR,
'DD Mon YYYY','nls_date_language=AMERICAN');
    to_timestamp_tz
---------------------------
 2001-12-05 00:00:00+08:00
(1 row)
```

说明

该函数在参数a_format_version值为10c且a_format_dev_version值为s4的情况下有效。

- to_dsinterval(text)

  描述：将字符转换为interval类型。支持SQL兼容格式与ISO格式。

  返回值类型：interval

  示例：

```
gaussdb=# SELECT to_dsinterval('12 1:2:3.456');
    to_dsinterval
----------------------
 12 days 01:02:03.456
(1 row)

gaussdb=# SELECT to_dsinterval('P3DT4H5M6S');
  to_dsinterval
-----------------
 3 days 04:05:06
(1 row)
```

说明

此函数在参数a_format_version值为10c且a_format_dev_version值为s2的情况下有效。

- to_yminterval(text)

  描述：将字符转换为interval类型。支持SQL兼容格式与ISO格式。

  返回值类型：interval

  示例：

```
gaussdb=# SELECT to_yminterval('1-1');
 to_yminterval
-----------------
 1 year 1 mon
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# SELECT to_yminterval('P13Y3M4DT4H2M5S');
 to_yminterval
-----------------
 13 years 3 mons
(1 row)
```

说明

此函数在参数a_format_version值为10c且a_format_dev_version值为s2的情况下有效。

表 1-75 数值格式化的模板模式

模式	描述
9	带有指定数值位数的值。
0	带前导零的值。
.（句点）	小数点。
,（逗号）	分组（千）分隔符。
PR	尖括号内负值。
S	带符号的数值（使用区域设置）。
L	货币符号（使用区域设置）。
D	小数点（使用区域设置）。
G	分组分隔符（使用区域设置）。
MI	在指明的位置的负号（如果数字<0）。
PL	在指明的位置的正号（如果数字>0）。
SG	在指明的位置的正/负号。
RN	罗马数字（输入在1和3999之间）。
TH或th	序数后缀。
V	移动指定位（小数）。
x或X	16进制转换10进制标识符。

表 1-76 to_number 数值格式化的模板模式

模式	描述
9	匹配一个数字，9的数量可以大于等于expr中对应位置的数字的数量。
0	严格匹配一个数字，0的数量要等于expr中数字的数量。
5	匹配一个0或5的数字。

版权所有 © 华为云计算技术有限公司

模式	描述
.（句点）	指定位置的小数点。
,（逗号）	指定位置的分组（千）分隔符，可以在fmt中指定多个逗号。
B	前导空白。
PR	尖括号（<>）中对应负值。空白对应正值。
S	前导减号（-）的负值，返回带有前导加号（+）的正值。返回带有尾随减号（-）的负值，返回带有尾随加号（+）的正值。
MI	尾随减号（-）的负值，尾随空白的正值。
$	前导美元符号。
L	本地货币符号。
C	指定位置ISO货币符号。
U	双币符号。
D	小数点（使用区域设置）。
G	分组分隔符（ISO标准），可以在fmt中指定多个逗号。
RN / rn	罗马数字（输入在1和3999之间），to_number不支持该fmt。
V	to_number不支持该fmt。
X / x	十六进制与十进制转换。
TM	to_number不支持。
FM	位于fmt中的起始位置，用于去除前导空格并保留占位符的0，对to_number的结果无实际影响。
EEEE	按照科学计数法模型转换。

说明

在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，fmt功能参考**表1-76**，否则如**表1-75**所示。**表1-76**中国际化ISO的fmt功能受参数lc_monetary和lc_numeric影响，具体表现为：不同的lc_monetary取值会影响货币符号；不同的lc_numeric取值则会影响小数点符号与千位分隔符的显示规则。

- cast_varchar2_to_raw_for_histogram(varchar2)

  描述：将varchar2类型转换为raw类型输出。

  返回值类型：raw
- abstime_text(abstime)

  描述：将abstime类型转换为text类型输出。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

- abstime_to_smalldatetime(abstime)
  描述：将abstime类型转换为smalldatetime类型。
  返回值类型：smalldatetime
- bigint_tid(bigint)
  描述：将bigint类型转换为tid类型。
  返回值类型：tid
- bool_int1(boolean)
  描述：将boolean类型转换为int1类型。
  返回值类型：tinyint
- bool_int2(boolean)
  描述：将boolean类型转换为int2类型。
  返回值类型：smallint
- bool_int8(boolean)
  描述：将boolean类型转换为int8类型。
  返回值类型：bigint
- bpchar_date(character)
  描述：将字符串转换为日期类型。
  返回值类型：date
- bpchar_float4(character)
  描述：将字符串转换为float4类型。
  返回值类型：real
- bpchar_float8(character)
  描述：将字符串转换为float8类型。
  返回值类型：double precision
- bpchar_int4(character)
  描述：将字符串转换为int4类型。
  返回值类型：integer
- bpchar_int8(character)
  描述：将字符串转换为int8类型。
  返回值类型：bigint
- bpchar_numeric(character)
  描述：将字符串转换为numeric类型。
  返回值类型：numeric
- bpchar_timestamp(character)
  描述：将字符串转换为时间戳类型。
  返回值类型：timestamp without time zone
- bpchar_to_smalldatetime(character)
  描述：将字符串转换为smalldatetime类型。
  返回值类型：smalldatetime
- complex_array_in(cstring, oid, int2vector)

版权所有 © 华为云计算技术有限公司

描述：将外部complex_array类型转换为内部anyarray数组类型。
返回值类型：anyarray

- date_bpchar(date)
  描述：将date类型转换为bpchar类型。
  返回值类型：character
- date_text(date)
  描述：将date类型转换为text类型。
  返回值类型：text
- date_varchar(date)
  描述：将date类型转换为varchar类型。
  返回值类型：character varying
- f4toi1(real)
  描述：将float4类型强转为uint8类型。
  返回值类型：tinyint
- f8toi1(double precision)
  描述：将float8类型强转为uint8类型。
  返回值类型：tinyint
- float4_bpchar(real)
  描述：将float4类型转换为bpchar类型。
  返回值类型：character
- float4_text(real)
  描述：将float4类型转换为text类型。
  返回值类型：text
- float4_varchar(real)
  描述：将float4类型转换为varchar类型。
  返回值类型：character varying
- float8_bpchar(double precision)
  描述：将float8类型转换为bpchar类型。
  返回值类型：character
- float8_interval(double precision)
  描述：将float8类型转换为interval类型。
  返回值类型：interval
- float8_text(double precision)
  描述：将float8类型转换为text类型。
  返回值类型：text
- float8_varchar(double precision)
  描述：将float8类型转换为varchar类型。
  返回值类型：character varying
- i1tof4(tinyint)
  描述：将uint8类型转换为float4类型。

版权所有 © 华为云计算技术有限公司

返回值类型：real

- i1tof8(tinyint)
  描述：将uint8类型转换为float8类型。
  返回值类型：double precision
- i1toi2(tinyint)
  描述：将uint8类型转换为int16类型。
  返回值类型：smallint
- i1toi4(tinyint)
  描述：将uint8类型转换为int32类型。
  参数：tinyint
  返回值类型：integer
- i1toi8(tinyint)
  描述：将uint8类型转换为int64类型。
  返回值类型：bigint
- i2toi1(smallint)
  描述：将int16类型转换为uint8类型。
  返回值类型：tinyint
- i4toi1(integer)
  描述：将int32类型转换为uint8类型。
  参数：integer
  返回值类型：tinyint
- i8toi1(bigint)
  描述：将int64类型转换为uint8类型。
  返回值类型：tinyint
- int1_avg_accum(bigint[], tinyint)
  描述：将第二个uint8类型参数，加入到第一个参数中，第一个参数为bigint类型数组。
  返回值类型：bigint[]
- int1_bool(tinyint)
  描述：将uint8类型转换为boolean类型。
  返回值类型：boolean
- int1_bpchar(tinyint)
  描述：将uint8类型转换为bpchar类型。
  返回值类型：character
- int1_mul_cash(tinyint, money)
  描述：返回一个int8类型参数和一个money类型参数的乘积，返回值为money类型。
  返回值类型：money
- int1_numeric(tinyint)
  描述：将uint8类型转换为numeric类型。
  返回值类型：numeric

版权所有 © 华为云计算技术有限公司

- int1_nvarchar2(tinyint)
  描述：将uint8类型转换为nvarchar2类型。
  返回值类型：nvarchar2
- int1_text(tinyint)
  描述：将uint8类型转换为text类型。
  返回值类型：text
- int1_varchar(tinyint)
  描述：将uint8类型转换为varchar类型。
  返回值类型：character varying
- int1in(cstring)
  描述：将字符串转换为无符号一字节整数类型。
  返回值类型：tinyint
- int1out(tinyint)
  描述：将无符号一字节整数类型转换为字符串。
  返回值类型：cstring
- int1up(tinyint)
  描述：输入整数转换为无符号一字节整数类型。
  返回值类型：tinyint
- int2_bool(smallint)
  描述：将有符号二字节整数类型转换为boolean类型。
  返回值类型：boolean
- int2_bpchar(smallint)
  描述：将有符号二字节整数类型转换为bpchar类型。
  返回值类型：character
- int2_text(smallint)
  描述：将有符号二字节整数类型转换为text类型。
  返回值类型：text
- int2_varchar(smallint)
  描述：将有符号二字节整数类型转换为varchar类型。
  返回值类型：character varying
- int4_bpchar(integer)
  描述：将有符号四字节整数类型转换为bpchar类型。
  返回值类型：character
- int4_text(integer)
  描述：将有符号四字节整数类型转换为text类型。
  返回值类型：text
- int4_varchar(integer)
  描述：将有符号四字节整数类型转换为varchar类型。
  返回值类型：character varying
- int8_bool(bigint)

版权所有 © 华为云计算技术有限公司

描述：将有符号八字节整数类型转换为boolean类型。
返回值类型：boolean

- int8_bpchar(bigint)
  描述：将有符号八字节整数类型转换为bpchar类型。
  返回值类型：character
- int8_text(bigint)
  描述：将有符号八字节整数类型转换为text类型。
  返回值类型：text
- int8_varchar(bigint)
  描述：将有符号八字节整数类型转换为varchar类型。
  返回值类型：character varying
- intervaltonum(interval)
  描述：将内部数据类型日期转换为numeric类型。
  返回值类型：numeric
- numeric_bpchar(numeric)
  描述：将numeric类型转换为bpchar类型。
  返回值类型：character
- numeric_int1(numeric)
  描述：将numeric类型转换为有符号一字节整数类型。
  返回值类型：tinyint
- numeric_text(numeric)
  描述：将numeric类型转换为text类型。
  返回值类型：text
- numeric_varchar(numeric)
  描述：将numeric类型转换为varchar类型。
  返回值类型：character varying
- nvarchar2in(cstring, oid, integer)
  描述：将c字符串转换为varchar类型。
  返回值类型：nvarchar2
- nvarchar2out(nvarchar2)
  描述：将text类型转换为c字符串。
  返回值类型：cstring
- nvarchar2send(nvarchar2)
  描述：将varchar类型转换为二进制。
  返回值类型：bytea
- oidvectorin_extend(cstring)
  描述：将字符串转化为oidvector类型。
  返回值类型：oidvector_extend
- oidvectorout_extend(oidvector_extend)
  描述：将oidvector类型转换为字符串。

版权所有 © 华为云计算技术有限公司

返回值类型：cstring

- oidvectorsend_extend(oidvector_extend)
  描述：将oidvector类型转换为字符串。
  返回值类型：bytea
- reltime_text(reltime)
  描述：将reltime类型转换为text类型。
  返回值类型：text
- text_date(text)
  描述：将text类型转换为date类型。
  返回值类型：date
- text_float4(text)
  描述：将text类型转换为float4类型。
  返回值类型：real
- text_float8(text)
  描述：将text类型转换为float8类型。
  返回值类型：double precision
- text_int1(text)
  描述：将text类型转换为int1类型。
  返回值类型：tinyint
- text_int2(text)
  描述：将text类型转换为int2类型。
  返回值类型：smallint
- text_int4(text)
  描述：将text类型转换为int4类型。
  返回值类型：integer
- text_int8(text)
  描述：将text类型转换为int8类型。
  返回值类型：bigint
- text_numeric(text)
  描述：将text类型转换为numeric类型。
  返回值类型：numeric
- text_timestamp(text)
  描述：将text类型转换为timestamp类型。
  返回值类型：timestamp without time zone
- time_text(time without time zone)
  描述：将time类型转换为text类型。
  返回值类型：text
- timestamp_text(timestamp without time zone)
  描述：将timestamp类型转换为text类型。
  返回值类型：text

版权所有 © 华为云计算技术有限公司

- timestamp_to_smalldatetime(timestamp without time zone)
  描述：将timestamp类型转换为smalldatetime类型。
  返回值类型：smalldatetime
- timestamp_varchar(timestamp without time zone)
  描述：将timestamp类型转换为varchar类型。
  返回值类型：character varying
- timestamptz_to_smalldatetime(timestamp with time zone)
  描述：将timestamptz类型转换为smalldatetime类型。
  返回值类型：smalldatetime
- timestampzone_text(timestamp with time zone)
  描述：将timestamptz类型转换为text类型。
  返回值类型：text
- timetz_text(time with time zone)
  描述：将timetz类型转换为text类型。
  返回值类型：text
- to_integer(character varying)
  描述：转换为integer类型。
  返回值类型：integer
- to_interval(character varying)
  描述：转换为interval类型。
  返回值类型：interval
- to_numeric(character varying)
  描述：转换为numeric类型。
  返回值类型：numeric
- to_text(smallint)
  描述：转换为text类型。
  返回值类型：text
- to_ts(character varying)
  描述：转换为timestamptz类型。
  返回值类型：timestamp without time zone
- to_varchar2(timestamp without time zone)
  描述：转换为varchar2类型。
  返回值类型：character varying
- varchar_blob(character varying)
  描述：将varchar类型转换为blob类型。
  返回值类型：blob

须知

此函数仅在ORA兼容模式下支持，其余模式会报错。

版权所有 © 华为云计算技术有限公司

- varchar_date(character varying)
  描述：将varchar类型转换为date类型。
  返回值类型：date
- varchar_float4(character varying)
  描述：将varchar类型转换为float4类型。
  返回值类型：real
- varchar_float8(character varying)
  描述：将varchar类型转换为float8类型。
  返回值类型：double precision
- varchar_int4(character varying)
  描述：将varchar类型转换为int4类型。
  返回值类型：integer
- varchar_int8(character varying)
  描述：将varchar类型转换为int8类型。
  返回值类型：bigint
- varchar_numeric(character varying)
  描述：将varchar类型转换为numeric类型。
  返回值类型：numeric
- varchar_timestamp(character varying)
  描述：将varchar类型转换为timestamp类型。
  返回值类型：timestamp without time zone
- varchar2_to_smlldatetime(character varying)
  描述：将varchar2类型转换为smlldatetime类型。
  返回值类型：smalldatetime
- xidout4(xid32)
  描述：将xid类型输出为4字节数字类型。
  返回值类型：cstring
- xidsend4(xid32)
  描述：将xid类型转换为二进制格式。
  返回值类型：bytea
- bytea_blob(any)
  描述：将bytea类型的数据转换为blob类型，当前只支持bytea类型的转换，其他类型暂不支持。
  返回值类型：blob
- datea_text(datea)
  描述：将datea类型转换为text类型。
  返回值类型：text
- datea_clob(datea)
  描述：将datea类型转换为clob类型。
  返回值类型：clob

版权所有 © 华为云计算技术有限公司

- datea_varchar(datea)
  描述：将datea类型转换为character varying类型。
  返回值类型：character varying
- datea_nvarchar2(datea)
  描述：将datea类型转换为nvarchar2类型。
  返回值类型：nvarchar2
- datea_bpchar(datea)
  描述：将datea类型转换为bpchar类型。
  返回值类型：character
- text_datea(text)
  描述：将text类型转换为datea类型。
  返回值类型：datea
- clob_datea(clob)
  描述：将clob类型转换为datea类型。
  返回值类型：datea
- varchar_datea(character varying)
  描述：将character varying类型转换为datea类型。
  返回值类型：datea
- bpchar_datea(character)
  描述：将character类型转换为datea类型。
  返回值类型：datea
- datea_timestamp(datea)
  描述：将datea类型转换为timestamp类型。
  返回值类型：timestamp
- timestamp_datea(timestamp)
  描述：将timestamp类型转换为datea类型。
  返回值类型：datea
- datea_timestamptz(datea)
  描述：将datea类型转为timestamptz类型。
  返回值类型：timestamptz
- timestamptz_datea(timestamptz)
  描述：将timestamptz类型转换为datea类型。
  返回值类型：datea
- nvarchar2_datea(nvarchar2)
  描述：将nvarchar2类型转换为datea类型。
  返回值类型：datea

  说明

  datea相关转换函数仅在ORA兼容性下可以使用。
- chartorowid(character/character varying/nvarchar2/text)
  描述：将character/character varying/nvarchar2/text类型转换为rowid类型。

版权所有 © 华为云计算技术有限公司

返回值类型：rowid

- rowidtobpchar(rowid)
  描述：将rowid类型转换为character类型。
  返回值类型：character
- rowidtochar(rowid)
  描述：将rowid类型转换为character varying类型。
  返回值类型：character varying
- rowidtonchar(rowid)
  描述：将rowid类型转换为nvarchar2类型。
  返回值类型：nvarchar2
- rowidtotext(rowid)
  描述：将rowid类型转换为text类型。
  返回值类型：text

  说明

  rowid相关转换函数仅在ORA兼容性下可以使用。

- treat(expr AS [JSON | REF] schema.type)
  描述：将expr转化为AS后关键字指定的类型（JSON或输入的用户自定义类型）。
  返回值类型：JSON或输入的用户自定义类型。
  示例：

```
gaussdb=# CREATE TABLE json_doc(data CLOB);
gaussdb=# INSERT INTO json_doc VALUES('{"name":"a"}');
gaussdb=# SELECT treat(data as json) FROM json_doc;
    json
--------------
 {"name":"a"}
(1 row)
gaussdb=# DROP TABLE json_doc;
DROP TABLE
```

- nesttable_to_array(anynesttable)
  描述：将一个无索引的集合类型转换成具有相同元素的数组类型。
  返回值类型：anyarray
  示例：

```
gaussdb=# CREATE OR REPLACE PROCEDURE p1 IS
gaussdb$#     type t1 IS table of int;
gaussdb$#     v1 t1 := t1(1, 2, 3);
gaussdb$#     v2 int[] := cast(v1 as int[]);
gaussdb$# BEGIN
gaussdb$#     raise info '%', v2;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# CALL p1();
INFO:  {1,2,3}
 p1
----


(1 row)

gaussdb=# CREATE TYPE t1 is table of int;
CREATE TYPE
gaussdb=# SELECT cast(t1(1, 2, 3) AS int[]) result;
 result
```


版权所有 © 华为云计算技术有限公司

```
---------
 {1,2,3}
(1 row)
gaussdb=# DROP PROCEDURE p1;
DROP PROCEDURE

gaussdb=# DROP TYPE t1;
DROP TYPE
```

- indexbytableint_to_array(anyindexbytable)

  描述：将一个索引类型为integer的集合类型转换成具有相同元素的数组类型。

  返回值类型：anyarray

  示例：

```
gaussdb=# CREATE OR REPLACE PACKAGE pkg1 IS
gaussdb$#     type t1 IS table of int index by int;
gaussdb$#     PROCEDURE p1();
gaussdb$# END pkg1;
gaussdb$# /
CREATE PACKAGE
gaussdb=#
gaussdb=# CREATE OR REPLACE PACKAGE BODY pkg1 IS
gaussdb$#     PROCEDURE p1() IS
gaussdb$#         v1 t1 := t1(1 => 1, 2 => 2, 3 => 3);
gaussdb$#         v2 int[];
gaussdb$#     BEGIN
gaussdb$#         v2 := cast(v1 as int[]);
gaussdb$#         RAISE INFO '%', v2;
gaussdb$#     END;
gaussdb$# END pkg1;
gaussdb$# /
CREATE PACKAGE BODY
gaussdb=#
gaussdb=# CALL pkg1.p1();
INFO:  {1,2,3}
 p1
----

(1 row)

gaussdb=# SELECT indexbytableint_to_array(pkg1.t1(1 => 1, 2 => 2, 3 => 3));
 indexbytableint_to_array
--------------------------
 {1,2,3}
(1 row)

gaussdb=# DROP PACKAGE pkg1;
DROP PACKAGE
```

编码类型转换

convert_to_nocase(text, text)

描述：将字符串转换为指定的编码类型后输出。

返回值类型：bytea

示例：

```
gaussdb=# SELECT convert_to_nocase('12345', 'GBK');
 convert_to_nocase
-------------------
 \x3132333435
(1 row)
```


版权所有 © 华为云计算技术有限公司

1.6.10 几何函数和操作符

几何操作符

- +

  描述：平移。

  示例：

```
gaussdb=# SELECT box '((0,0),(1,1))' + point '(2.0,0)' AS RESULT;
   result
-------------
 (3,1),(2,0)
(1 row)
```

- -

  描述：平移。

  示例：

```
gaussdb=# SELECT box '((0,0),(1,1))' - point '(2.0,0)' AS RESULT;
    result
---------------
 (-1,1),(-2,0)
(1 row)
```

- *

  描述：伸展/旋转。

  示例：

```
gaussdb=# SELECT box '((0,0),(1,1))' * point '(2.0,0)' AS RESULT;
   result
-------------
 (2,2),(0,0)
(1 row)
```

- /

  描述：收缩/旋转。

  示例：

```
gaussdb=# SELECT box '((0,0),(2,2))' / point '(2.0,0)' AS RESULT;
   result
-------------
 (1,1),(0,0)
(1 row)
```

- #

  描述：两个图形交面。

  示例：

```
gaussdb=# SELECT box '((1,-1),(-1,1))' # box '((1,1),(-2,-2))' AS RESULT;
 result
---------------
 (1,1),(-1,-1)
(1 row)
```

- #

  描述：图形的路径数目或多边形顶点数。

  示例：

```
gaussdb=# SELECT # path'((1,0),(0,1),(-1,0))' AS RESULT;
 result
--------
      3
(1 row)
```


版权所有 © 华为云计算技术有限公司

- @-@

  描述：图形的长度或者周长。

  示例：

```
gaussdb=# SELECT @-@ path '((0,0),(1,0))' AS RESULT;
 result
--------
      2
(1 row)
```

- @@

  描述：图形的中心。

  示例：

```
gaussdb=# SELECT @@ circle '((0,0),10)' AS RESULT;
 result
--------
 (0,0)
(1 row)
```

- <->

  描述：两个图形之间的距离。

  示例：

```
gaussdb=# SELECT circle '((0,0),1)' <-> circle '((5,0),1)' AS RESULT;
 result
--------
      3
(1 row)
```

- &&

  描述：两个图形是否重叠（有一个共同点就为真）。

  示例：

```
gaussdb=# SELECT box '((0,0),(1,1))' && box '((0,0),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- <<

  描述：图形是否全部在另一个图形的左边（没有相同的横坐标）。

  示例：

```
gaussdb=# SELECT circle '((0,0),1)' << circle '((5,0),1)' AS RESULT;
 result
--------
 t
(1 row)
```

- >>

  描述：图形是否全部在另一个图形的右边（没有相同的横坐标）。

  示例：

```
gaussdb=# SELECT circle '((5,0),1)' >> circle '((0,0),1)' AS RESULT;
 result
--------
 t
(1 row)
```

- &<

  描述：图形的最右边是否不超过在另一个图形的最右边。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT box '((0,0),(1,1))' &< box '((0,0),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- &>

  描述：图形的最左边是否不超过在另一个图形的最左边。

  示例：

```
gaussdb=# SELECT box '((0,0),(3,3))' &> box '((0,0),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- <<|

  描述：图形是否全部在另一个图形的下方（没有相同的纵坐标）。

  示例：

```
gaussdb=# SELECT box '((0,0),(3,3))' <<| box '((3,4),(5,5))' AS RESULT;
 result
--------
 t
(1 row)
```

- |>>

  描述：图形是否全部在另一个图形的上方（没有相同的纵坐标）。

  示例：

```
gaussdb=# SELECT box '((3,4),(5,5))' |>> box '((0,0),(3,3))' AS RESULT;
 result
--------
 t
(1 row)
```

- &<|

  描述：图形的上边界是否不超过另一个图形的上边界。

  示例：

```
gaussdb=# SELECT box '((0,0),(1,1))' &<| box '((0,0),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- |&>

  描述：图形的下边界是否不超过另一个图形的下边界。

  示例：

```
gaussdb=# SELECT box '((0,0),(3,3))' |&> box '((0,0),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- <^

  描述：图形是否低于另一个图形（允许两个图形有接触）。

  示例：

```
gaussdb=# SELECT box '((0,0),(-3,-3))' <^ box '((0,0),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

- >^

  描述：图形是否高于另一个图形（允许两个图形有接触）。

  示例：

```
gaussdb=# SELECT box '((0,0),(2,2))' >^ box '((0,0),(-3,-3))'  AS RESULT;
 result
--------
 t
(1 row)
```

- ?#

  描述：两个图形是否相交。

  示例：

```
gaussdb=# SELECT lseg '((-1,0),(1,0))' ?# box '((-2,-2),(2,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- ?-

  描述：图形是否处于水平位置。

  示例：

```
gaussdb=# SELECT ?- lseg '((-1,0),(1,0))' AS RESULT;
 result
--------
 t
(1 row)
```

- ?-

  描述：图形是否水平对齐。

  示例：

```
gaussdb=# SELECT point '(1,0)' ?- point '(0,0)' AS RESULT;
 result
--------
 t
(1 row)
```

- ?|

  描述：图形是否处于竖直位置。

  示例：

```
gaussdb=# SELECT ?| lseg '((-1,0),(1,0))' AS RESULT;
 result
--------
 f
(1 row)
```

- ?|

  描述：图形是否竖直对齐。

  示例：

```
gaussdb=# SELECT point '(0,1)' ?| point '(0,0)' AS RESULT;
 result
--------
 t
(1 row)
```

- ?-|

  描述：两条线是否垂直。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT lseg '((0,0),(0,1))' ?-| lseg '((0,0),(1,0))' AS RESULT;
 result
--------
 t
(1 row)
```

- ?||

  描述：两条线是否平行。

  示例：

```
gaussdb=# SELECT lseg '((-1,0),(1,0))' ?|| lseg '((-1,2),(1,2))' AS RESULT;
 result
--------
 t
(1 row)
```

- @>

  描述：图形是否包含另一个图形。

  示例：

```
gaussdb=# SELECT circle '((0,0),2)' @> point '(1,1)' AS RESULT;
 result
--------
 t
(1 row)
```

- <@

  描述：图形是否被包含于另一个图形。

  示例：

```
gaussdb=# SELECT point '(1,1)' <@ circle '((0,0),2)' AS RESULT;
 result
--------
 t
(1 row)
```

- ~=

  描述：两个图形是否相同。

  示例：

```
gaussdb=# SELECT polygon '((0,0),(1,1))' ~= polygon '((1,1),(0,0))' AS RESULT;
 result
--------
 t
(1 row)
```

几何函数

- area(object)

  描述：计算图形的面积。

  返回类型：double precision

  示例：

```
gaussdb=# SELECT area(box '((0,0),(1,1))') AS RESULT;
 result
--------
      1
(1 row)
```

- center(object)

  描述：计算图形的中心。

  返回类型：point

  示例：

 版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT center(box '((0,0),(1,2))') AS RESULT;
 result
---------
 (0.5,1)
(1 row)
```

- diameter(circle)
  描述：计算圆的直径。
  返回类型：double precision
  示例：

```
gaussdb=# SELECT diameter(circle '((0,0),2.0)') AS RESULT;
 result
--------
      4
(1 row)
```

- height(box)
  描述：矩形的竖直高度。
  返回类型：double precision
  示例：

```
gaussdb=# SELECT height(box '((0,0),(1,1))') AS RESULT;
 result
--------
      1
(1 row)
```

- isclosed(path)
  描述：图形是否为闭合路径。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT isclosed(path '((0,0),(1,1),(2,0))') AS RESULT;
 result
--------
 t
(1 row)
```

- isopen(path)
  描述：图形是否为开放路径。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT isopen(path '[(0,0),(1,1),(2,0)]') AS RESULT;
 result
--------
 t
(1 row)
```

- length(object)
  描述：计算图形的长度。
  返回类型：double precision
  示例：

```
gaussdb=# SELECT length(path '((-1,0),(1,0))') AS RESULT;
 result
--------
      4
(1 row)
```

- npoints(path)
  描述：计算路径的顶点数。

版权所有 © 华为云计算技术有限公司

返回类型：int

示例：

```
gaussdb=# SELECT npoints(path '[(0,0),(1,1),(2,0)]') AS RESULT;
 result
--------
      3
(1 row)
```

- npoints(polygon)

  描述：计算多边形的顶点数。

  返回类型：int

  示例：

  ```
  gaussdb=# SELECT npoints(polygon '((1,1),(0,0))') AS RESULT;
   result
  --------
       2
  (1 row)
  ```

- pclose(path)

  描述：把路径转换为闭合路径。

  返回类型：path

  示例：

  ```
  gaussdb=# SELECT pclose(path '[(0,0),(1,1),(2,0)]') AS RESULT;
        result
  ---------------------
   ((0,0),(1,1),(2,0))
  (1 row)
  ```

- popen(path)

  描述：把路径转换为开放路径。

  返回类型：path

  示例：

  ```
  gaussdb=# SELECT popen(path '((0,0),(1,1),(2,0))') AS RESULT;
        result
  ---------------------
   [(0,0),(1,1),(2,0)]
  (1 row)
  ```

- radius(circle)

  描述：计算圆的半径。

  返回类型：double precision

  示例：

  ```
  gaussdb=# SELECT radius(circle '((0,0),2.0)') AS RESULT;
   result
  --------
       2
  (1 row)
  ```

- width(box)

  描述：计算矩形的水平尺寸。

  返回类型：double precision

  示例：

  ```
  gaussdb=# SELECT width(box '((0,0),(1,1))') AS RESULT;
   result
  --------
       1
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

几何类型转换函数

- box(circle)

  描述：将圆转换成矩形。

  返回类型：box

  示例：

```
gaussdb=# SELECT box(circle '((0,0),2.0)') AS RESULT;
                                result
-----------------------------------------------------------------------
 (1.41421356237309,1.41421356237309),(-1.41421356237309,-1.41421356237309)
(1 row)
```

- box(point, point)

  描述：将点转换成矩形。

  返回类型：box

  示例：

```
gaussdb=# SELECT box(point '(0,0)', point '(1,1)') AS RESULT;
   result
-------------
 (1,1),(0,0)
(1 row)
```

- box(polygon)

  描述：将多边形转换成矩形。

  返回类型：box

  示例：

```
gaussdb=# SELECT box(polygon '((0,0),(1,1),(2,0))') AS RESULT;
   result
-------------
 (2,1),(0,0)
(1 row)
```

- circle(box)

  描述：矩形转换成圆。

  返回类型：circle

  示例：

```
gaussdb=# SELECT circle(box '((0,0),(1,1))') AS RESULT;
            result
-------------------------------
 <(0.5,0.5),0.707106781186548>
(1 row)
```

- circle(point, double precision)

  描述：将圆心和半径转换成圆。

  返回类型：circle

  示例：

```
gaussdb=# SELECT circle(point '(0,0)', 2.0) AS RESULT;
  result
-----------
 <(0,0),2>
(1 row)
```

- circle(polygon)

  描述：将多边形转换成圆。

  返回类型：circle

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT circle(polygon '((0,0),(1,1),(2,0))') AS RESULT;
                result
-----------------------------------------
 <(1,0.333333333333333),0.924950591148529>
(1 row)
```

- lseg(box)

  描述：矩形对角线转化成线段。

  返回类型：lseg

  示例：

```
gaussdb=# SELECT lseg(box '((-1,0),(1,0))') AS RESULT;
    result
----------------
 [(1,0),(-1,0)]
(1 row)
```

- lseg(point, point)

  描述：点转换成线段。

  返回类型：lseg

  示例：

```
gaussdb=# SELECT lseg(point '(-1,0)', point '(1,0)') AS RESULT;
    result
----------------
 [(-1,0),(1,0)]
(1 row)
```

- slope(point, point)

  描述：计算两个点构成直线的斜率。

  返回类型：double

  示例：

```
gaussdb=# SELECT slope(point '(1,1)', point '(0,0)') AS RESULT;
 result
--------
      1
(1 row)
```

- path(polygon)

  描述：多边形转换成路径。

  返回类型：path

  示例：

```
gaussdb=# SELECT path(polygon '((0,0),(1,1),(2,0))') AS RESULT;
      result
--------------------
 ((0,0),(1,1),(2,0))
(1 row)
```

- point(double precision, double precision)

  描述：节点。

  返回类型：point

  示例：

```
gaussdb=# SELECT point(23.4, -44.5) AS RESULT;
   result
--------------
 (23.4,-44.5)
(1 row)
```

- point(box)

  描述：矩形的中心。

版权所有 © 华为云计算技术有限公司

返回类型：point

示例：

```
gaussdb=# SELECT point(box '((-1,0),(1,0))') AS RESULT;
 result
--------
 (0,0)
(1 row)
```

- point(circle)

  描述：圆心。

  返回类型：point

  示例：

  ```
  gaussdb=# SELECT point(circle '((0,0),2.0)') AS RESULT;
   result
  --------
   (0,0)
  (1 row)
  ```

- point(lseg)

  描述：线段的中心。

  返回类型：point

  示例：

  ```
  gaussdb=# SELECT point(lseg '((-1,0),(1,0))') AS RESULT;
   result
  --------
   (0,0)
  (1 row)
  ```

- point(polygon)

  描述：多边形的中心。

  返回类型：point

  示例：

  ```
  gaussdb=# SELECT point(polygon '((0,0),(1,1),(2,0))') AS RESULT;
         result
  ----------------------
   (1,0.333333333333333)
  (1 row)
  ```

- polygon(box)

  描述：矩形转换成4点多边形。

  返回类型：polygon

  示例：

  ```
  gaussdb=# SELECT polygon(box '((0,0),(1,1))') AS RESULT;
           result
  ---------------------------
   ((0,0),(0,1),(1,1),(1,0))
  (1 row)
  ```

- polygon(circle)

  描述：圆转换成12点多边形。

  返回类型：polygon

  示例：

  ```
  gaussdb=# SELECT polygon(circle '((0,0),2.0)') AS RESULT;

  result

  ------------------------------------------------------------------------------------------------------------------------
  ```


版权所有 © 华为云计算技术有限公司

```
-----------------------------------------------------------------------------------------------------------------------------------------------------------
 ((-2,0),(-1.73205080756888,1),(-1,1.73205080756888),(-1.22464679914735e-16,2),
(1,1.73205080756888),(1.73205080756888,1),(2,2.44929359829471e-16),
(1.73205080756888,-0.999999999999999),(1,-1.73205080756888),(3.67394039744206e-16,-2),
(-0.999999999999999,-1.73205080756888),(-1.73205080756888,-1))
(1 row)
```

- polygon(npts, circle)

  描述：圆转换成npts点多边形。

  返回类型：polygon

  示例：

```
gaussdb=# SELECT polygon(12, circle '((0,0),2.0)') AS RESULT;

result

-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
 ((-2,0),(-1.73205080756888,1),(-1,1.73205080756888),(-1.22464679914735e-16,2),
(1,1.73205080756888),(1.73205080756888,1),(2,2.44929359829471e-16),
(1.73205080756888,-0.999999999999999),(1,-1.73205080756888),(3.67394039744206e-16,-2),
(-0.999999999999999,-1.73205080756888),(-1.73205080756888,-1))
(1 row)
```

- polygon(path)

  描述：路径转换成多边形。

  返回类型：polygon

  示例：

```
gaussdb=# SELECT polygon(path '((0,0),(1,1),(2,0))') AS RESULT;
      result
---------------------
 ((0,0),(1,1),(2,0))
(1 row)
```

1.6.11 网络地址函数和操作符

cidr 和 inet 操作符

操作符<<，<<=，>>，>>=对子网进行测试。这些操作符仅比较两个地址的网络部分（忽略任何主机部分），以确定其中一个网络是等于另外一个网络，还是另外一个网络的子网。

- <

  描述：小于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' < inet '192.168.1.6' AS RESULT;
 result
--------
 t
(1 row)
```

- <=

  描述：小于或等于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' <= inet '192.168.1.5' AS RESULT;
 result
--------
```


版权所有 © 华为云计算技术有限公司

```
 t
(1 row)
```

- =

  描述：等于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' = inet '192.168.1.5' AS RESULT;
 result
--------
 t
(1 row)
```

- >=

  描述：大于或等于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' >= inet '192.168.1.5' AS RESULT;
 result
--------
 t
(1 row)
```

- >

  描述：大于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' > inet '192.168.1.4' AS RESULT;
 result
--------
 t
(1 row)
```

- <>

  描述：不等于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' <> inet '192.168.1.4' AS RESULT;
 result
--------
 t
(1 row)
```

- <<

  描述：包含于。

  示例：

```
gaussdb=# SELECT inet '192.168.1.5' << inet '192.168.1/24' AS RESULT;
 result
--------
 t
(1 row)
```

- <<=

  描述：包含于或等于。

  示例：

```
gaussdb=# SELECT inet '192.168.1/24' <<= inet '192.168.1/24' AS RESULT;
 result
--------
 t
(1 row)
```

- >>

  描述：包含。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT inet '192.168.1/24' >> inet '192.168.1.5' AS RESULT;
 result
--------
 t
(1 row)
```

- >>=

  描述：包含或等于。

  示例：

```
gaussdb=# SELECT inet '192.168.1/24' >>= inet '192.168.1/24' AS RESULT;
 result
--------
 t
(1 row)
```

- ~

  描述：位非。

  示例：

```
gaussdb=# SELECT ~ inet '192.168.1.6' AS RESULT;
    result
---------------
 63.87.254.249
(1 row)
```

- &

  描述：两个网络地址的每一位都进行“与”操作。

  示例：

```
gaussdb=# SELECT inet '192.168.1.6' & inet '10.0.0.0' AS RESULT;
 result
---------
 0.0.0.0
(1 row)
```

- |

  描述：两个网络地址的每一位都进行“或”操作。

  示例：

```
gaussdb=# SELECT inet '192.168.1.6' | inet '10.0.0.0' AS RESULT;
   result
-------------
 202.168.1.6
(1 row)
```

- +

  描述：加。

  示例：

```
gaussdb=# SELECT inet '192.168.1.6' + 25 AS RESULT;
    result
--------------
 192.168.1.31
(1 row)
```

- -

  描述：减。

  示例：

```
gaussdb=# SELECT inet '192.168.1.43' - 36 AS RESULT;
   result
-------------
 192.168.1.7
(1 row)
```


版权所有 © 华为云计算技术有限公司

- -

  描述：减。

  示例：

```
gaussdb=# SELECT inet '192.168.1.43' - inet '192.168.1.19' AS RESULT;
 result
--------
     24
(1 row)
```

cidr 和 inet 函数

函数abbrev，host，text主要是为了提供可选的显示格式。

- abbrev(inet)

  描述：缩写显示格式文本。

  返回类型：text

  示例：

```
gaussdb=# SELECT abbrev(inet '10.1.0.0/16') AS RESULT;
   result
-------------
 10.1.0.0/16
(1 row)
```

- abbrev(cidr)

  描述：缩写显示格式文本。

  返回类型：text

  示例：

```
gaussdb=# SELECT abbrev(cidr '10.1.0.0/16') AS RESULT;
 result
---------
 10.1/16
(1 row)
```

- broadcast(inet)

  描述：网络广播地址。

  返回类型：inet

  示例：

```
gaussdb=# SELECT broadcast('192.168.1.5/24') AS RESULT;
      result
------------------
 192.168.1.255/24
(1 row)
```

- family(inet)

  描述：抽取地址族，4为IPv4，6为IPv6。

  返回类型：int

  示例：

```
gaussdb=# SELECT family('127.0.0.1') AS RESULT;
 result
--------
      4
(1 row)
```

- host(inet)

  描述：将主机地址类型抽出为文本。

  返回类型：text

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT host('192.168.1.5/24') AS RESULT;
   result
-------------
 192.168.1.5
(1 row)
```

- hostmask(inet)

  描述：为网络构造主机掩码。

  返回类型：inet

  示例：

  ```
  gaussdb=# SELECT hostmask('192.168.23.20/30') AS RESULT;
   result
  ---------
   0.0.0.3
  (1 row)
  ```

- masklen(inet)

  描述：抽取子网掩码长度。

  返回类型：int

  示例：

  ```
  gaussdb=# SELECT masklen('192.168.1.5/24') AS RESULT;
   result
  --------
      24
  (1 row)
  ```

- netmask(inet)

  描述：为网络构造子网掩码。

  返回类型：inet

  示例：

  ```
  gaussdb=# SELECT netmask('192.168.1.5/24') AS RESULT;
     result
  ---------------
   255.255.255.0
  (1 row)
  ```

- network(inet)

  描述：抽取地址的网络部分。

  返回类型：cidr

  示例：

  ```
  gaussdb=# SELECT network('192.168.1.5/24') AS RESULT;
      result
  ----------------
   192.168.1.0/24
  (1 row)
  ```

- set_masklen(inet, int)

  描述：为inet数值设置子网掩码长度。

  返回类型：inet

  示例：

  ```
  gaussdb=# SELECT set_masklen('192.168.1.5/24', 16) AS RESULT;
      result
  ----------------
   192.168.1.5/16
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- set_masklen(cidr, int)

  描述：为cidr数值设置子网掩码长度。

  返回类型：cidr

  示例：

```
gaussdb=# SELECT set_masklen('192.168.1.0/24'::cidr, 16) AS RESULT;
    result
----------------
 192.168.0.0/16
(1 row)
```

- text(inet)

  描述：把IP地址和掩码长度抽取为文本。

  返回类型：text

  示例：

```
gaussdb=# SELECT text(inet '192.168.1.5') AS RESULT;
    result
----------------
 192.168.1.5/32
(1 row)
```

任何cidr值都能以显式或者隐式的方式转换为inet值，因此上述能够操作inet值的函数也同样能够操作cidr值。inet值也可以转换为cidr值，此时inet子网掩码右侧的所有位都将转换为零，以创建一个有效的cidr值。另外，用户还可以使用常规的类型转换语法将一个文本字符串转换为inet或cidr值。例如：inet(expression)或colname::cidr。

macaddr 函数

trunc(macaddr)

描述：函数trunc(macaddr)返回一个MAC地址，该地址的最后三个字节设置为零。

返回类型：macaddr

示例：

```
gaussdb=# SELECT trunc(macaddr '12:34:56:78:90:ab') AS RESULT;
     result
-------------------
 12:34:56:00:00:00
(1 row)
```

说明

macaddr类型还支持标准关系操作符（>，<=等）用于词法排序，和按位运算符（~，&和|）非，与和或。

1.6.12 文本检索函数和操作符

文本检索操作符

- @@

  描述：tsvector类型的词汇与tsquery类型的词汇是否匹配。

  示例：

```
gaussdb=# SELECT to_tsvector('fat cats ate rats') @@ to_tsquery('cat & rat') AS RESULT;
 result
--------
 t
(1 row)
```

 版权所有 © 华为云计算技术有限公司

- @@@

  描述：@@的同义词。

  示例：

```
gaussdb=# SELECT to_tsvector('fat cats ate rats') @@@ to_tsquery('cat & rat') AS RESULT;
 result
--------
 t
(1 row)
```

- ||

  描述：连接两个tsvector类型的词汇。

  示例：

```
gaussdb=# SELECT 'a:1 b:2'::tsvector || 'c:1 d:2 b:3'::tsvector AS RESULT;
          result
---------------------------
 'a':1 'b':2,5 'c':3 'd':4
(1 row)
```

- &&

  描述：将两个tsquery类型的词汇进行“与”操作。

  示例：

```
gaussdb=# SELECT 'fat | rat'::tsquery && 'cat'::tsquery AS RESULT;
          result
----------------------------
 ( 'fat' | 'rat' ) & 'cat'
(1 row)
```

- ||

  描述：将两个tsquery类型的词汇进行“或”操作。

  示例：

```
gaussdb=# SELECT 'fat | rat'::tsquery || 'cat'::tsquery AS RESULT;
        result
-----------------------
 'fat' | 'rat' | 'cat'
(1 row)
```

- !!

  描述：tsquery类型词汇的非关系。

  示例：

```
gaussdb=# SELECT !! 'cat'::tsquery AS RESULT;
 result
--------
 !'cat'
(1 row)
```

- <->

  描述：将两个tsquery类型的词汇进行“词组”操作。

  示例：

```
gaussdb=# SELECT 'fat'::tsquery <-> 'cat'::tsquery AS RESULT;
    result
-----------------
 'fat' <-> 'cat'
(1 row)
```

- @>

  描述：一个tsquery类型的词汇是否包含另一个tsquery类型的词汇。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT 'cat'::tsquery @> 'cat & rat'::tsquery AS RESULT;
 result
--------
 f
(1 row)
```

- <@

  描述：一个tsquery类型的词汇是否被包含另一个tsquery类型的词汇。

  示例：

```
gaussdb=# SELECT 'cat'::tsquery <@ 'cat & rat'::tsquery AS RESULT;
 result
--------
 t
(1 row)
```

除了上述的操作符，还为tsvector类型和tsquery类型的数据定义了普通的B-tree比较操作符（=，<等）。

文本检索函数

- get_current_ts_config()

  描述：获取文本检索的默认配置。

  返回类型：regconfig

  示例：

```
gaussdb=# SELECT get_current_ts_config();
 get_current_ts_config
-----------------------
 english
(1 row)
```

- length(tsvector)

  描述：tsvector类型词汇的单词数。

  返回类型：integer

  示例：

```
gaussdb=# SELECT length('fat:2,4 cat:3 rat:5A'::tsvector);
 length
--------
      3
(1 row)
```

- numnode(tsquery)

  描述：tsquery类型的单词加上操作符的数量。

  返回类型：integer

  示例：

```
gaussdb=# SELECT numnode('(fat & rat) | cat'::tsquery);
 numnode
---------
       5
(1 row)
```

- plainto_tsquery([config regconfig,] query text)

  描述：产生tsquery类型的词汇，使用&（AND）作为连接操作符，并忽略标点。

  返回类型：tsquery

  示例：

```
gaussdb=# SELECT plainto_tsquery('english', 'The Fat Rats');
 plainto_tsquery
-----------------
```


版权所有 © 华为云计算技术有限公司

```
 'fat' & 'rat'
(1 row)
```

- phraseto_tsquery([config regconfig,] query text)

  描述：产生tsquery类型的词汇，使用<->（PHRASE）作为连接操作符，并忽略标点。config为token配置词典，详情请参见**词典**。

  返回类型：tsquery

  示例：

```
gaussdb=# SELECT phraseto_tsquery('english', 'The Fat Rats');
 phraseto_tsquery
------------------
 'fat' <-> 'rat'
(1 row)
```

- tsquery_phrase(query1 tsquery, query2 tsquery [, distance regdistance])

  描述：使用词组操作符（PHRASE）连接tsquery类型。query1和query2可使用text类型数据，会尝试转换为tsquery类型。distance可指定query1和query2的距离。

  返回类型：tsquery

  示例：

```
--不指定距离，默认为1。
gaussdb=# SELECT tsquery_phrase(to_tsquery('fat'), to_tsquery('cat'));
 tsquery_phrase
-----------------
 'fat' <-> 'cat'
(1 row)

--指定距离。
gaussdb=# SELECT tsquery_phrase(to_tsquery('fat'), to_tsquery('cat'), 2);
 tsquery_phrase
-----------------
 'fat' <2> 'cat'
(1 row)
```

- querytree(query tsquery)

  描述：获取tsquery类型的词汇可加索引的部分。

  返回类型：text

  示例：

```
gaussdb=# SELECT querytree('foo & ! bar'::tsquery);
 querytree
-----------
 'foo'
(1 row)
```

- setweight(tsvector, "char")

  描述：给tsvector类型的每个元素分配权重值，权重的取值为：'A'、'B'、'C'、'D'。

  返回类型：tsvector

  示例：

```
gaussdb=# SELECT setweight('fat:2,4 cat:3 rat:5B'::tsvector, 'A');
          setweight
-------------------------------
 'cat':3A 'fat':2A,4A 'rat':5A
(1 row)
```

- strip(tsvector)

  描述：删除tsvector类型单词中的位置信息和权重值。

  返回类型：tsvector

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT strip('fat:2,4 cat:3 rat:5A'::tsvector);
      strip
-------------------
 'cat' 'fat' 'rat'
(1 row)
```

- to_tsquery([config regconfig ,] query text)

  描述：标准化单词，并转换为tsquery类型。具体功能介绍详见**解析查询**。

  返回类型：tsquery

  示例：

```
gaussdb=# SELECT to_tsquery('english', 'The & Fat & Rats');
  to_tsquery
---------------
 'fat' & 'rat'
(1 row)
```

- to_tsvector([config regconfig ,] document text)

  描述：将文档内容标准化，去除停用词，排序并去重后生成tsvector类型数据。

  返回类型：tsvector

  示例：

```
gaussdb=# SELECT to_tsvector('english', 'The Fat Rats');
   to_tsvector
-----------------
 'fat':2 'rat':3
(1 row)
```

- to_tsvector_for_batch([config regconfig ,] document text)

  描述：将文档内容标准化，去除停用词，排序并去重后生成tsvector类型数据。

  返回类型：tsvector

  示例：

```
gaussdb=# SELECT to_tsvector_for_batch('english', 'The Fat Rats');
   to_tsvector
-----------------
 'fat':2 'rat':3
(1 row)
```

- ts_headline([config regconfig,] document text, query tsquery [, options text])

  描述：高亮显示查询的匹配项。具体功能介绍详见**高亮搜索结果**。

  返回类型：text

  示例：

```
gaussdb=# SELECT ts_headline('x y z', 'z'::tsquery);
 ts_headline
--------------
 x y <b>z</b>
(1 row)
```

- ts_rank([weights float4[],] vector tsvector, query tsquery [, normalization integer])

  描述：文档查询排名。具体功能介绍详见**排序查询结果**。

  返回类型：float4

  示例：

```
gaussdb=# SELECT ts_rank('hello world'::tsvector, 'world'::tsquery);
 ts_rank
----------
```


版权所有 © 华为云计算技术有限公司

```
 .0607927
(1 row)
```

- ts_rank_cd([weights float4[],] vector tsvector, query tsquery [, normalization integer])

  描述：排序文件查询使用覆盖密度。具体功能介绍详见**排序查询结果**。

  返回类型：float4

  示例：

```
gaussdb=# SELECT ts_rank_cd('hello world'::tsvector,'world'::tsquery);
 ts_rank_cd
------------
          0
(1 row)
```

- ts_rewrite(query tsquery, target tsquery, substitute tsquery)

  描述：替换目标tsquery类型的单词。具体功能介绍详见**查询重写**。

  返回类型：tsquery

  示例：

```
gaussdb=# SELECT ts_rewrite('a & b'::tsquery, 'a'::tsquery, 'foo|bar'::tsquery);
       ts_rewrite
-------------------------
 'b' & ( 'foo' | 'bar' )
(1 row)
```

- ts_rewrite(query tsquery, select text)

  描述：使用SELECT命令的结果替代目标中tsquery类型的单词。具体功能介绍详见**查询重写**。

  返回类型：tsquery

  示例：

```
gaussdb=# SELECT ts_rewrite('world'::tsquery, 'select ''world''::tsquery, ''hello''::tsquery');
 ts_rewrite
------------
 'hello'
(1 row)
```

文本检索调试函数

须知

该功能为内部使用功能，不建议用户使用。

- ts_debug([config regconfig,] document text, OUT alias text, OUT description text, OUT token text, OUT dictionaries regdictionary[], OUT dictionary regdictionary, OUT lexemes text[])

  描述：测试一个配置。

  返回类型：setof record

  示例：

```
gaussdb=# SELECT ts_debug('english', 'The Brightest supernovaes');
                                   ts_debug
-------------------------------------------------------------------------------
 (asciiword,"Word, all ASCII",The,{english_stem},english_stem,{})
 (blank,"Space symbols"," ",{},,)
 (asciiword,"Word, all ASCII",Brightest,{english_stem},english_stem,{brightest})
 (blank,"Space symbols"," ",{},,)
```


版权所有 © 华为云计算技术有限公司

```
 (asciiword,"Word, all ASCII",supernovaes,{english_stem},english_stem,{supernova})
(5 rows)
```

- ts_lexize(dict regdictionary, token text)

  描述：测试一个数据字典。

  返回类型：text[]

  示例：

  ```
  gaussdb=# SELECT ts_lexize('english_stem', 'stars');
   ts_lexize
  -----------
   {star}
  (1 row)
  ```

- ts_parse(parser_name text, document text, OUT tokid integer, OUT token text)

  描述：测试一个解析。

  返回类型：setof record

  示例：

  ```
  gaussdb=# SELECT ts_parse('default', 'foo - bar');
   ts_parse
  -----------
   (1,foo)
   (12," ")
   (12,"- ")
   (1,bar)
  (4 rows)
  ```

- ts_parse(parser_oid oid, document text, OUT tokid integer, OUT token text)

  描述：测试一个解析。

  返回类型：setof record

  示例：

  ```
  gaussdb=# SELECT ts_parse(3722, 'foo - bar');
   ts_parse
  -----------
   (1,foo)
   (12," ")
   (12,"- ")
   (1,bar)
  (4 rows)
  ```

- ts_token_type(parser_name text, OUT tokid integer, OUT alias text, OUT description text)

  描述：获取分析器定义的记号类型。

  返回类型：setof record

  示例：

  ```
  gaussdb=# SELECT ts_token_type('default');
                          ts_token_type
  ---------------------------------------------------------------
   (1,asciiword,"Word, all ASCII")
   (2,word,"Word, all letters")
   (3,numword,"Word, letters and digits")
   (4,email,"Email address")
   (5,url,URL)
   (6,host,Host)
   (7,sfloat,"Scientific notation")
   (8,version,"Version number")
   (9,hword_numpart,"Hyphenated word part, letters and digits")
   (10,hword_part,"Hyphenated word part, all letters")
   (11,hword_asciipart,"Hyphenated word part, all ASCII")
   (12,blank,"Space symbols")
  ```


版权所有 © 华为云计算技术有限公司

```
 (13,tag,"XML tag")
 (14,protocol,"Protocol head")
 (15,numhword,"Hyphenated word, letters and digits")
 (16,asciihword,"Hyphenated word, all ASCII")
 (17,hword,"Hyphenated word, all letters")
 (18,url_path,"URL path")
 (19,file,"File or path name")
 (20,float,"Decimal notation")
 (21,int,"Signed integer")
 (22,uint,"Unsigned integer")
 (23,entity,"XML entity")
(23 rows)
```

- ts_token_type(parser_oid oid, OUT tokid integer, OUT alias text, OUT description text)

  描述：获取分析器定义的记号类型。

  返回类型：setof record

  示例：

```
gaussdb=# SELECT ts_token_type(3722);
                        ts_token_type
------------------------------------------------------------------
 (1,asciiword,"Word, all ASCII")
 (2,word,"Word, all letters")
 (3,numword,"Word, letters and digits")
 (4,email,"Email address")
 (5,url,URL)
 (6,host,Host)
 (7,sfloat,"Scientific notation")
 (8,version,"Version number")
 (9,hword_numpart,"Hyphenated word part, letters and digits")
 (10,hword_part,"Hyphenated word part, all letters")
 (11,hword_asciipart,"Hyphenated word part, all ASCII")
 (12,blank,"Space symbols")
 (13,tag,"XML tag")
 (14,protocol,"Protocol head")
 (15,numhword,"Hyphenated word, letters and digits")
 (16,asciihword,"Hyphenated word, all ASCII")
 (17,hword,"Hyphenated word, all letters")
 (18,url_path,"URL path")
 (19,file,"File or path name")
 (20,float,"Decimal notation")
 (21,int,"Signed integer")
 (22,uint,"Unsigned integer")
 (23,entity,"XML entity")
(23 rows)
```

- ts_stat(sqlquery text, [weights text,] OUT word text, OUT ndoc integer, OUT nentry integer)

  描述：获取tsvector列的统计数据。

  返回类型：setof record

  示例：

```
gaussdb=# SELECT ts_stat('select ''hello world''::tsvector');
   ts_stat
-------------
 (world,1,1)
 (hello,1,1)
(2 rows)
```

1.6.13 JSON/JSONB 函数和操作符

JSON/JSONB数据类型参考**JSON/JSONB类型**，操作符如**表1-77**和**表1-78**所示。

版权所有 © 华为云计算技术有限公司

表 1-77 JSON/JSONB 通用操作符

操作符	左操作数类型	右操作数类型	返回类型	描述	示例
->	Array-json(b)	int	json(b)	获得array-json元素。下标不存在返回空。	SELECT '[{"a":"foo"},{"b":"bar"},{"c":"baz"}]'::json->2; ?column? ---------- {"c":"baz"} (1 row)
->	object-json(b)	text	json(b)	通过键获得值。不存在则返回空。	SELECT '{"a": {"b":"foo"}}'::json->'a'; ?column? ---------- {"b":"foo"} (1 row)
->>	Array-json(b)	int	text	获得JSON数组元素。下标不存在返回空。	SELECT '[1,2,3]'::json->>2; ?column? ---------- 3 (1 row)
->>	object-json(b)	text	text	通过键获得值。不存在则返回空。	SELECT '{"a":1,"b":2}'::json->>'b'; ?column? ---------- 2 (1 row)
#>	container-json(b)	text[]	json(b)	获取在指定路径的JSON对象，路径不存在则返回空。	SELECT '{"a": {"b":{"c": "foo"}}}'::json #>'{a,b}'; ?column? -------------- {"c": "foo"} (1 row)
#>>	container-json(b)	text[]	text	获取在指定路径的JSON对象，路径不存在则返回空。	SELECT '{"a":[1,2,3],"b":[4,5,6]}'::json #>>'{a,2}'; ?column? ---------- 3 (1 row)

注意

对于#>和#>>操作符，当给出的路径无法查找到数据时，不会报错，会返回空。

表 1-78 JSONB 额外支持操作符

操作符	右操作数类型	描述	例子
@>	jsonb	左边的JSON的顶层是否包含右边JSON的顶层所有项。	'{"a":1, "b":2}'::jsonb @> '{"b":2}'::jsonb

版权所有 © 华为云计算技术有限公司

操作符	右操作数类型	描述	例子
<@	jsonb	左边的JSON的所有项是否全部存在于右边JSON的顶层。	'{"b":2}'::jsonb <@ '{"a":1, "b":2}'::jsonb
?	text	键/元素的字符串是否存在于JSON值的顶层。	'{"a":1, "b":2}'::jsonb ? 'b'
?|	text[]	这些数组字符串中的任何一个是否作为顶层键存在。	'{"a":1, "b":2, "c":3}'::jsonb ?| array['b', 'c']
?&	text[]	是否所有这些数组字符串都作为顶层键存在。	'["a", "b"]'::jsonb ?& array['a', 'b']
=	jsonb	判断两个jsonb的大小关系，同函数jsonb_eq。	/
<>	jsonb	判断两个jsonb的大小关系，同函数jsonb_ne。	/
<	jsonb	判断两个jsonb的大小关系，同函数jsonb_lt。	/
>	jsonb	判断两个jsonb的大小关系，同函数jsonb_gt。	/
<=	jsonb	判断两个jsonb的大小关系，同函数jsonb_le。	/
>=	jsonb	判断两个jsonb的大小关系，同函数jsonb_ge。	/

说明

GIN索引可以有效地搜索大量JSONB文档（datums）中的键或者键值对。提供了两个GIN操作符类（jsonb_ops和jsonb_hash_ops），以供不同的性能和灵活性取舍。

- 缺省的GIN操作符类jsonb_ops：支持使用@>、?、 ?&和?|操作符进行查询。
- 非缺省的GIN操作符类jsonb_hash_ops：仅支持索引@>操作符进行查询。

JSON/JSONB 支持的函数

- array_to_json(anyarray [, pretty_bool])

  描述：返回JSON类型的数组。将一个多维数组组成一个JSON数组。如果pretty_bool为true，将在一维元素之间添加换行符。

版权所有 © 华为云计算技术有限公司

返回类型：json

示例：

```
gaussdb=# SELECT array_to_json('{{1,5},{99,100}}'::int[]);
  array_to_json
-----------------
 [[1,5],[99,100]]
(1 row)
```

- row_to_json(record [, pretty_bool])

  描述：返回JSON类型的行。如果pretty_bool为true，将在第一级元素之间添加换行符。

  返回类型：json

  示例：

  ```
  gaussdb=# SELECT row_to_json(row(1,'foo'));
      row_to_json
  ---------------------
   {"f1":1,"f2":"foo"}
  (1 row)
  ```

- json_array_element(array-json, integer)、jsonb_array_element(array-jsonb, integer)

  描述：同操作符`->`, 返回数组中指定下标的元素。

  返回类型：json、jsonb

  示例：

  ```
  gaussdb=# SELECT json_array_element('[1,true,[1,[2,3]],null]',2);
   json_array_element
  --------------------
   [1,[2,3]]
  (1 row)
  ```

- json_array_element_text(array-json, integer)、jsonb_array_element_text(array-jsonb, integer)

  描述：同操作符`->>`, 返回数组中指定下标的元素。

  返回类型：text、text

  示例：

  ```
  gaussdb=# SELECT json_array_element_text('[1,true,[1,[2,3]],null]',2);
  json_array_element_text
  ------------------------
   [1,[2,3]]
  (1 row)
  ```

- json_object_field(object-json, text)、jsonb_object_field(object-jsonb, text)

  描述：同操作符`->`, 返回对象中指定键对应的值。

  返回类型：json、json

  示例：

  ```
  gaussdb=# SELECT json_object_field('{"a": {"b":"foo"}}','a');
   json_object_field
  -------------------
   {"b":"foo"}
  (1 row)
  ```

- json_object_field_text(object-json, text)、jsonb_object_field_text(object-jsonb, text)

  描述：同操作符`->>`, 返回对象中指定键对应的值。

  返回类型：text、text

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT json_object_field_text('{"a": {"b":"foo"}}','a');
json_object_field_text
----------------------
 {"b":"foo"}
(1 row)
```

- json_extract_path(json, VARIADIC text[])、jsonb_extract_path((jsonb, VARIADIC text[])

  描述：等价于操作符`#>`。根据第二个参数所指的路径，查找json，并返回。

  返回类型：json、jsonb

  示例：

```
gaussdb=# SELECT json_extract_path('{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}', 'f4','f6');
 json_extract_path
-------------------
 "stringy"
(1 row)
```

- json_extract_path_op(json, text[])、jsonb_extract_path_op(jsonb, text[])

  描述：同操作符`#>`。根据第二个参数所指的路径，查找json，并返回。

  返回类型：json、jsonb

  示例：

```
gaussdb=# SELECT json_extract_path_op('{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}', ARRAY['f4','f6']);
 json_extract_path_op
----------------------
 "stringy"
(1 row)
```

- json_extract_path_text(json, VARIADIC text[])、jsonb_extract_path_text((jsonb, VARIADIC text[])

  描述：等价于操作符`#>>`。根据第二个参数所指的路径，查找json，并返回。

  返回类型：text、text

  示例：

```
gaussdb=# SELECT json_extract_path_text('{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}', 'f4','f6');
 json_extract_path_text
------------------------
 stringy
(1 row)
```

- json_extract_path_text_op(json, text[])、jsonb_extract_path_text_op(jsonb, text[])

  描述：同操作符`#>>`。根据第二个参数所指的路径，查找json，并返回。

  返回类型：text、text

  示例：

```
gaussdb=# SELECT json_extract_path_text_op('{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}',
ARRAY['f4','f6']);
 json_extract_path_text_op
---------------------------
 stringy
(1 row)
```

- json_array_elements(array-json)、jsonb_array_elements(array-jsonb)

  描述：拆分数组，每一个元素返回一行。

  返回类型：json、jsonb

  示例：

```
gaussdb=# SELECT json_array_elements('[1,true,[1,[2,3]],null]');
 json_array_elements
----------------------
 1
 true
```


版权所有 © 华为云计算技术有限公司

```
 [1,[2,3]]
 null
(4 rows)
```

- json_array_elements_text(array-json)、jsonb_array_elements_text(array-jsonb)

  描述：拆分数组，每一个元素返回一行。

  返回类型：text、text

  示例：

```
gaussdb=# SELECT * FROM  json_array_elements_text('[1,true,[1,[2,3]],null]');
   value
-----------
 1
 true
 [1,[2,3]]

(4 rows)
```

- json_array_length(array-json)、jsonb_array_length(array-jsonb)

  描述：返回数组长度。

  返回类型：integer

  示例：

```
gaussdb=# SELECT json_array_length('[1,2,3,{"f1":1,"f2":[5,6]},4,null]');
 json_array_length
-------------------
                 6
(1 row)
```

- json_each(object-json)、jsonb_each(object-jsonb)

  描述：将对象的每个键值对拆分转换成一行两列。

  返回类型：setof(key text, value json)、setof(key text, value jsonb)

  示例：

```
gaussdb=# SELECT * FROM  json_each('{"f1":[1,2,3],"f2":{"f3":1},"f4":null}');
 key |  value
-----+----------
 f1  | [1,2,3]
 f2  | {"f3":1}
 f4  | null
(3 rows)
```

- json_each_text(object-json)、jsonb_each_text(object-jsonb)

  描述：将对象的每个键值对拆分转换成一行两列。

  返回类型：setof(key text, value text)、setof(key text, value text)

  示例：

```
gaussdb=# SELECT * FROM  json_each_text('{"f1":[1,2,3],"f2":{"f3":1},"f4":null}');
 key |  value
-----+----------
 f1  | [1,2,3]
 f2  | {"f3":1}
 f4  |
(3 rows)
```

- json_object_keys(object-json)、jsonb_object_keys(object-jsonb)

  描述：返回对象中顶层的所有键。

  返回类型：SETOF text

  示例：

```
gaussdb=# SELECT json_object_keys('{"f1":"abc","f2":{"f3":"a", "f4":"b"}, "f1":"abcd"}');
 json_object_keys
------------------
 f1
 f2
```


版权所有 © 华为云计算技术有限公司

```
 f1
(3 rows)
```

```
--jsonb_object_keys函数会有去重操作
gaussdb=# SELECT jsonb_object_keys('{"f1":"abc","f2":{"f3":"a", "f4":"b"}, "f1":"abcd"}');
 jsonb_object_keys
-------------------
 f1
 f2
(2 rows)
```

- json_populate_record(anyelement, object-json [, bool])、jsonb_populate_record(anyelement, object-jsonb [, bool])

  描述：将第二个参数中的每个键值对进行拆分，以键作为列名，与第一个参数（必须为复合类型）中的列名进行匹配查找，并将其值填充到第一个参数的格式中。

  返回类型：anyelement、anyelement

  示例：

```
gaussdb=# CREATE TYPE jpop AS (a text, b int, c bool);
CREATE TYPE
gaussdb=# SELECT * FROM json_populate_record(null::jpop,'{"a":"blurfl","x":43.2}');
   a    | b | c
--------+---+---
 blurfl |   |
(1 row)
```

```
gaussdb=# SELECT * FROM json_populate_record((1,1,null)::jpop,'{"a":"blurfl","x":43.2}');
   a    | b | c
--------+---+---
 blurfl | 1 |
(1 row)
gaussdb=# DROP TYPE jpop;
DROP TYPE
```

- json_populate_recordset(anyelement, array-json [, bool])、jsonb_populate_recordset(anyelement, array-jsonb [, bool])

  描述：参考上述函数json_populate_record、jsonb_populate_record，对第二个参数的数组中的每一个元素进行上述参考函数的操作，因此这也要求第二个参数的数组中的每个元素都是object-json类型的。

  返回类型：setof anyelement、setof anyelement

  示例：

```
gaussdb=# CREATE TYPE jpop AS (a text, b int, c bool);
CREATE TYPE
gaussdb=# SELECT * FROM json_populate_recordset(null::jpop, '[{"a":1,"b":2},{"a":3,"b":4}]');
 a | b | c
---+---+---
 1 | 2 |
 3 | 4 |
(2 rows)
gaussdb=# DROP TYPE jpop;
DROP TYPE
```

- json_typeof(json)、jsonb_typeof(jsonb)

  描述：检测json类型。

  返回类型：text、text

  示例：

```
gaussdb=# SELECT value, json_typeof(value)  FROM (values (json '123.4'), (json '"foo"'), (json 'true'),
(json 'null'), (json '[1, 2, 3]'), (json '{"x":"foo", "y":123}'), (NULL::json))  AS data(value);
        value         | json_typeof
----------------------+-------------
 123.4                | number
 "foo"                | string
 true                 | boolean
```


版权所有 © 华为云计算技术有限公司

```
 null                 | null
 [1, 2, 3]            | array
 {"x":"foo", "y":123} | object
                      |
(7 rows)
```

- json_build_array([VARIADIC "any"])

  描述：从一个可变参数列表构造出一个JSON数组。

  返回类型：array-json

  示例：

```
gaussdb=# SELECT json_build_array('a',1,'b',1.2,'c',true,'d',null,'e',json '{"x": 3, "y": [1,2,3]}','');
                          json_build_array
-------------------------------------------------------------------------------
 ["a", 1, "b", 1.2, "c", true, "d", null, "e", {"x": 3, "y": [1,2,3]}, null]
(1 row)
```

- json_build_object([VARIADIC "any"])

  描述：从一个可变参数列表构造出一个JSON对象，其入参必须为偶数个，两两一组组成键值对。注意键的值不可为NULL。

  返回类型：object-json

  示例：

```
gaussdb=# SELECT json_build_object(1,2);
 json_build_object
-------------------
 {"1" : 2}
(1 row)
```

- jsonb_build_object([VARIADIC "any"])

  描述：从一个可变参数列表构造出一个JSONB对象，其入参必须为偶数个，两两一组组成键值对。注意键的值不可为NULL。

  返回类型：object-jsonb

  须知

  - 当可变参数列表的元素存在空字符串（''）时，若数据库SQL兼容模式为ORA，则对应的元素位置返回结果为NULL；若数据库SQL兼容模式为PG, 则对应的元素位置返回结果为空字符串。这是因为ORA兼容模式时空字符串作为NULL处理。
  - 当可变参数列表元素存在DATE类型时，若数据库SQL兼容模式为ORA，在开关mapping_date_to_datea关闭时，DATE会被替换为TIMESTAMP(0) WITHOUT TIME ZONE类型。在开关mapping_date_to_datea打开时，DATE会被替换为DATEA类型。因此对应元素位置返回结果格式受mapping_date_to_datea开关是否打开影响；若数据库SQL兼容模式为PG, 则对应的元素位置返回结果不包含时分秒。

  示例：

```
gaussdb=# SELECT jsonb_build_object('a',1,'b',1.2,'c',true,'d',null,'e',json '{"x": 3, "y": [1,2,3]}');
                         jsonb_build_object
-------------------------------------------------------------------------------
 {"a": 1, "b": 1.2, "c": true, "d": null, "e": {"x": 3, "y": [1, 2, 3]}}
(1 row)

gaussdb=# SELECT jsonb_build_object();
 jsonb_build_object
--------------------
 {}
(1 row)
```


版权所有 © 华为云计算技术有限公司

- json_to_record(object-json, bool)、json_to_record(object-json)

  描述：正如所有返回record的函数一样，调用者必须用一个AS子句显式地定义记录的结构。会将object-json的键值对进行拆分重组，把键当做列名，去匹配填充AS显式指定的记录的结构，键值需要符合对应数据类型的结构。对于时间类型date，ORA兼容模式下，在开关mapping_date_to_datea关闭时，相当于TIMESTAMP(0) WITHOUT TIME ZONE，因此与PG兼容模式相比，在显示上会多出00:00:00时分秒字符串。

  参数：第一个重载函数的bool入参表示是否支持对象嵌套，true为支持，false为不支持。当只有一个入参时，默认支持嵌套，且入参仅有一个时可将嵌套json对象转换为行类型及支持方括号格式数组的解析。

  返回类型：record

  示例：

```
gaussdb=# SELECT * FROM json_to_record('{"a":1,"b":"foo","c":"bar"}',true) AS x(a int, b text, d text);
  a |  b  | d
 ---+-----+---
  1 | foo |
 (1 row)
gaussdb=# SELECT * FROM json_to_record('{"a": {"x": 1, "y": 2},"b":"foo","c":[1, 2]}') AS x(a json, b
text, c int[]);
         a         |  b  |   c
 ------------------+-----+-------
  {"x": 1, "y": 2} | foo | {1,2}
 (1 row)
```

- json_to_recordset(array-json, bool)

  描述：参考函数json_to_record，对数组内每个元素，执行上述函数的操作，因此这要求数组内的每个元素都是object-json，bool入参也可参考函数json_to_record，true为支持解析嵌套对象，false为不支持，即是否支持json对象中元素的value值也是一个json对象。

  返回类型：setof record

  示例：

```
gaussdb=# SELECT * FROM json_to_recordset('[{"a":1,"b":"foo","d":false},
{"a":2,"b":"bar","c":true}]',false) AS x(a int, b text, c boolean);
 a |  b  | c
---+-----+---
 1 | foo |
 2 | bar | t
(2 rows)
```

- json_object(text[])、json_object(text[], text[])

  描述：从一个文本数组构造一个object-json。这是个重载函数，当入参为一个文本数组的时候，其数组长度必须为偶数，数组内的成员被当作交替出现的键值对。两个文本数组的时候，第一个数组认为是键，第二个认为是值，两个数组长度必须相等。键的值不可为NULL。

  返回类型：object-json

  示例：

```
gaussdb=# SELECT json_object('{a,1,b,2,3,NULL,"d e f","a b c"}');
                      json_object
-------------------------------------------------------
 {"a" : "1", "b" : "2", "3" : null, "d e f" : "a b c"}
(1 row)
gaussdb=# SELECT json_object('{a,b,"a b c"}', '{a,1,1}');
              json_object
----------------------------------------
 {"a" : "a", "b" : "1", "a b c" : "1"}
(1 row)
```

- json_object([VARIADIC "any"])

版权所有 © 华为云计算技术有限公司

描述：从一个可变参数列表构造出一个json对象，其入参必须为偶数个，两两一组组成键值对。键的值为null报错；参数格式为奇数报错。

参数：可变参数列表，入参是一个或多个成对的key-value组合。

返回类型：json

示例：

```
gaussdb=# SELECT json_object('d',2,'c','name','b',true,'a',2,'a',NULL,'d',1);
               json_object
-----------------------------------------
 {"a": 2, "b": true, "c": "name", "d": 2}
(1 row)

gaussdb=# SELECT json_object('d',2,true,'name','b',true,'a',2,'aa', current_timestamp);
                                  json_object
-------------------------------------------------------------------------------
 {"1": "name", "a": 2, "b": true, "d": 2, "aa": "2023-08-12 11:28:13.385958"}
(1 row)
```

说明

此函数在MYSQL模式数据库中，且GUC参数 b_format_version = '5.7'和 b_format_dev_version = 's1'时生效，生效时会替换原有json_object行为。

- json_append/json_array_append(json, [VARIADIC "any"])

  描述：接受一个json入参和一个可变参数列表。从一个可变参数列表构造出若干对json_path和value对，向json对象中指定的路径追加一个值，并返回修改后的json。json_append与json_array_append完全一致。如果有任何参数为null返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：详情请参见**表1-79**。

  表 1-79 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	需要被修改的json值。如果传入其他非限定的类型，执行时将报错。	符合json格式。
[VARIADIC "any"]	可变any数组	可变参数列表，包含可变个json_path-value对。json_path-value对将从可变参数列表中构造。	json_path必须是一个有效的路径表达式，value可以是任意值。

  返回类型：json

  示例：

```
gaussdb=# SELECT json_array_append('[1, [2, 3]]', '$[1]', 4, '$[0]', false, '$[0]', null, '$[0]',
current_timestamp);
                    json_array_append
```


版权所有 © 华为云计算技术有限公司

```
-----------------------------------------------------------------
 [[1, false, null, "2023-08-12 14:27:16.142355+08"], [2, 3, 4]]
(1 row)
```

- json_array([VARIADIC "any"])

  描述：从一个可变参数列表构造出一个数组，并返回数组形式的json。如果该函数没有任何参数，则返回一个空的数组形式的json。

  参数：入参为可变参数列表，这些列表中的值可以是任意类型。

  返回类型：json

  示例：

```
--不输入入参，返回一个空的数组形式的json
gaussdb=# SELECT json_array();
))::numeric >= $0) json_array
------------
 []
(1 row)

--入参可以是任意类型
gaussdb=# SELECT json_array(TRUE, FALSE, NULL, 114, 'text', current_timestamp);
                          json_array
---------------------------------------------------------------------
 [true, false, null, 114, "text", "2023-08-12 15:17:34.979294+08"]
(1 row)
```

- json_array_insert(json, [VARIADIC "any"])

  描述：从一个可变参数列表中，构造一个或者多个json_path和value的路径-值对，向json对象中的json_path指定数组路径插入一个值value并返回新的json。如果指定的路径上有值，则在该路径上插入，并将已存在的值后移。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如**表1-80**所示。

  表 1-80 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	需要修改的json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
[VARIADIC "any"]	可变any数组	可变参数列表，可以支持一个或多个json_path-value对。	json_path必须是一个有效的路径表达式，value可以是任意值。

  返回类型：json

  示例：

版权所有 © 华为云计算技术有限公司

```
--一对json_path-value用例
gaussdb=# SELECT json_array_insert('[1, [2, 3]]', '$[1]', 4);
 json_array_insert
-------------------
 [1, 4, [2, 3]]
(1 row)

--多个json_path-value用例
gaussdb=# SELECT json_array_insert('{"x": 1, "y": [1, 2]}', '$.y[0]', NULL, '$.y[0]', 123, '$.y[3]',
current_timestamp);
                        json_array_insert
------------------------------------------------------------------
 {"x": 1, "y": [123, null, 1, "2023-08-14 14:54:12.85087+08", 2]}
(1 row)
```

- json_contains(target_json, candidate_json[, json_path])

  描述：入参为两个json对象和一个可选的json_path路径。如果不选定json路径，则判断target_json是否包含了candidate_json。如果选定路径，则在target_json指定路径json_path指向的json中，判断是否包含candidate_json。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如**表1-81**所示。

  表 1-81 参数说明

参数名称	类型	描述	取值范围
target_json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	目标json对象，需要判断candidate_json是否被该json包含。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
candidate_json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	json子对象，需要判断是否被target_json包含。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
json_path	text	可选参数，指定json的路径json_path。如果指定，则使用target_json的json_path指定路径判断包含关系。	json_path必须是一个有效的路径表达式。

版权所有 © 华为云计算技术有限公司

返回类型：bigint类型，如果包含返回1，如果不包含返回0。

示例：

```
--不指定路径
gaussdb=# SELECT json_contains('[1, 2, {"x": 3}]', '{"x":3}');
 json_contains
---------------
             1
(1 row)

--指定路径
gaussdb=# SELECT json_contains('[1, 2, {"x": 3},[1,2,3,4]]', '2','$[1]');
 json_contains
---------------
             1
(1 row)

gaussdb=# SELECT json_contains('[1, 2, {"x": 3},[1,2,3,4]]', '1','$[1]');
 json_contains
---------------
             0
(1 row)
```

- json_contains_path(json, mode_str, [VARIADIC text])

  描述：该函数用于检查一个json对象在指定的路径上是否有值，这个路径可以是多个。第一个入参为json对象，第二个入参为字符串“one”或“all”，它指示是否检查所有的路径。第三个入参为一个可变参数列表，所有的json_path路径将会从这个可变参数列表中构造。如果路径根据模式判断存在返回1，如果不存在返回0。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式则会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如**表1-82**所示。

  表 1-82 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	需要传入的json对象。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
mode_str	text	字符串“one”或者“all”，如果是“one”，其中一个路径存在返回1，否则返回0；如果是“all”，所有路径存在返回1，否则返回0。	必须为字符串“one”或“all”，不区分大小写。
[VARIADIC text]	可变text数组	可变参数列表，从此列表中构造所有的json_path路径。	json_path必须为有效的路径表达式。

版权所有 © 华为云计算技术有限公司

返回类型：bigint类型

示例：

```
--all场景，需要所有路径都存在
gaussdb=# SELECT json_contains_path('[1, 2, {"x": 3}]', 'all', '$[0]', '$[1]', '$[2]');
 json_contains_path
--------------------
                  1
(1 row)

gaussdb=# SELECT json_contains_path('[1, 2, {"x": 3}]', 'all', '$[0]', '$[1]', '$[6]');
 json_contains_path
--------------------
                  0
(1 row)

--one场景，其中一个路径存在即可
gaussdb=# SELECT json_contains_path('[1, 2, {"x": 3}]', 'one', '$[0]', '$[1]', '$[5]');
 json_contains_path
--------------------
                  1
(1 row)
```

- json_depth(json)

  描述：此函数用于返回传入json对象的最大深度。入参为null，返回null。

  参数：入参为一个需要查询深度的json对象，仅支持JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2类型；若传入其他非限定类型或无效的json，执行时将报错。

  返回类型：int

  示例：

  ```
  --空数组的深度为1
  gaussdb=# SELECT json_depth('[]');
   json_depth
  ------------
           1
  (1 row)

  gaussdb=# SELECT json_depth('{"s":1, "x":2,"y":[1]}');
   json_depth
  ------------
           3
  (1 row)
  ```

- json_extract(json, [VARIADIC text])

  描述：入参为一个json对象，和若干个json_path路径，json路径从可变参数列表中构造，此函数在json对象中提取指定json_path路径的数据，并将其组合成一个json数组返回。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式则会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如**表1-83**所示。

版权所有 © 华为云计算技术有限公司

表 1-83 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	需要提取路径的json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
[VARIADIC text]	可变text数组	可变参数列表，可以支持一个或多个json_path路径。	json_path必须是一个有效的路径表达式。

返回类型：json

示例：

```
--提取一个路径的值
gaussdb=# SELECT json_extract('[1, 2, {"x": 3}]', '$[2]');
 json_extract
--------------
 {"x": 3}
(1 row)

--提取多个路径的值
gaussdb=# SELECT json_extract('["a", ["b", "c"], "d"]', '$[1]', '$[2]', '$[3]');
   json_extract
--------------------
 [["b", "c"], "d"]
(1 row)
```

- json_insert(json, [VARIADIC any])

  描述：此函数在json对象中的json_path路径指定的位置插入一个value值，仅当路径不存在时才执行插入操作。入参为一个json对象，若干个json_path路径和需要插入的值，json_path和值必须两两配对。json和json_path中存在null值返回null。并且json_path中不能存在“*”、“**”通配符，否则报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如**表1-84**所示。

版权所有 © 华为云计算技术有限公司

表 1-84 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	待插入的json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
[VARIADIC any]	可变any数组	可变参数列表，可以支持一个或多个json_path-value对。	json_path必须是一个有效的路径表达式，value可以是任意值。

返回类型：json

示例：

```
gaussdb=# SELECT json_insert('[1, [2, 3], {"a": [4, 5]}]', '$[3]', 2);
         json_insert
-------------------------------
 [1, [2, 3], {"a": [4, 5]}, 2]
(1 row)

gaussdb=# SELECT json_insert('[1, [2, 3], {"a": [4, 5]}]', '$[10]', 10,'$[5]', 5);
           json_insert
-----------------------------------
 [1, [2, 3], {"a": [4, 5]}, 10, 5]
(1 row)
```

- json_keys(json[, json_path])

  描述：入参为一个json对象，和一个可选的json_path路径。如果没有传入json_path路径，此函数返回json对象中最上层对象的成员key值的json数组。如果传入了json_path路径，此函数返回该路径对应的json对象中最上层的成员key值的json数组。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。

  参数：如表1-85所示。

版权所有 © 华为云计算技术有限公司

表 1-85 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	一个json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
json_path	text	可选的，一个json_path路径。	json_path必须是一个有效的路径表达式。

返回类型：json

示例：

```
gaussdb=# SELECT json_keys('{"x": 1, "y": 2, "z": 3}');
   json_keys
-----------------
 ["x", "y", "z"]
(1 row)

gaussdb=# SELECT json_keys('[1,2,3,{"name":"Tom"}]','$[3]');
 json_keys
-----------
 ["name"]
(1 row)
```

- json_length(json[, json_path])

  描述：入参为一个json对象，和一个可选的json_path路径。如果没有传入json_path路径，此函数返回传入json对象的长度。如果传入了json_path路径，此函数返回该路径对应的json对象的长度。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。

  参数：如表1-86所示。

表 1-86 参数说明

参数名称	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	一个json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述	取值范围
json_path	text	可选的，一个json_path路径。	json_path必须是一个有效的路径表达式。

返回类型：int

示例：

```
gaussdb=# SELECT json_length('[1,2,3,4,5]');
 json_length
-------------
           5
(1 row)

gaussdb=# SELECT json_length('{"name":"Tom", "age":24, "like":"football"}');
 json_length
-------------
           3
(1 row)
```

- json_merge([VARIADIC any])

  描述：此函数将所有传入的json对象进行合并，并返回合并后的结果。入参中有null，返回null。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：为可变参数列表，用于构造多个json对象。入参由多个json对象组成，仅支持JSON、JSONB、CSTRING、TEXT、 "char"、VARCHAR、 BPCHAR、NVARCHAR2类型，若传入其他非限定类型，执行时将报错。

  返回类型：json

  示例：

  ```
  gaussdb=# SELECT json_merge('[1, 2]','[2]');
   json_merge
  ------------
   [1, 2, 2]
  (1 row)

  gaussdb=# SELECT json_merge('{"b":"2"}','{"a":"1"}','[1,2]');
            json_merge
  -------------------------------
   [{"a": "1", "b": "2"}, 1, 2]
  (1 row)
  ```

- json_quote(json)

  描述：此函数使用双引号将入参包装成一个字符串形式的json类型并返回。

  参数：入参为一个json对象，仅支持JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2类型，若传入其他非限定类型，执行时将报错。

  返回类型：json

  示例：

  ```
  gaussdb=# SELECT json_quote('gauss');
   json_quote
  ------------
   "gauss"
  (1 row)
  ```

- json_unquote(json)

  描述：此函数将取消入参的双引号引用，并返回取消后的字符串。

版权所有 © 华为云计算技术有限公司

参数：入参为一个json对象，仅支持JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2类型，若传入其他非限定类型，执行时将报错。

返回类型：text

示例：

```
gaussdb=# SELECT json_unquote('"gauss"');
 json_unquote
--------------
 gauss
(1 row)
```

- json_remove(json, [VARIADIC text])

  描述：入参为一个json对象，和若干个需要被删除的json_path路径，这些json_path路径从可变参数列表中构造。此函数将删除json对象中指定json_path路径的值，并返回删除后的json。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如表1-87所示。

  表 1-87 参数说明

参数说明	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	一个json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
[VARIADIC text]	可变text数组	可变参数列表，可以支持一个或多个json_path路径。	json_path必须是一个有效的路径表达式。

  返回类型：json

  示例：

```
gaussdb=# SELECT json_remove('[0, 1, 2, [3, 4]]', '$[0]', '$[2]');
 json_remove
-------------
 [1, 2]
(1 row)

gaussdb=# SELECT json_remove('[0, 1, 2, [3, 4]]', '$[0]', '$[0]','$[0]');
 json_remove
-------------
 [[3, 4]]
(1 row)
```

- json_replace(json, [VARIADIC any])

  描述：入参为一个json对象，和若干个json_path路径和需要替换的值，json_path和值必须两两配对；此函数在一个json对象中使用指定的值来替换指定json_path

版权所有 © 华为云计算技术有限公司

路径上的数据，并返回修改后的json。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

参数：如表1-88所示。

表 1-88 参数说明

参数说明	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	一个json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
[VARIADIC any]	可变any入参	可变参数列表，可以支持一个或多个json_path-value对。	json_path必须是一个有效的路径表达式，value可以是任意值。

返回类型：json

示例：

```
gaussdb=# SELECT json_replace('{"x": 1}', '$.x', 'true');
 json_replace
---------------
 {"x": "true"}
(1 row)

gaussdb=# SELECT json_replace('{"x": 1}', '$.x', true, '$.x', 123, '$.x', 'asd', '$.x', null);
 json_replace
--------------
 {"x": null}
(1 row)
```

- json_search(json, mode_str, search_str, escape_char, json_path)

  描述：函数返回一个给定字符串在一个json对象中的路径。它返回一个路径字符串或者由多个路径组成的数组。如果参数json、search_str或json_path任意一个是null，返回null。如果json格式错误或者json_path不是有效的路径表达式则会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如表1-89所示。

版权所有 © 华为云计算技术有限公司

表 1-89 参数说明

参数说明	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	目标的json。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
mode_str	text	字符串“one”或者“all”，如果是“one”，表示只获取第一个匹配到的路径；如果是“all”，则会匹配所有符合的路径。	必须是字符串“one”或者“all”，不区分大小写。
search_str	text	被搜索的字符串，正常情况下为全字匹配，但是可以使用通配符：%: 匹配任意数量的任意字符，_: 匹配一个任意字符。	-
escape_char	text	可选的，如果search_str中包含通配符，则该参数定义为通配符的转义字符。默认为'\'。若在通配符前加上该字符，则不将该字符看做通配符。	可以是任意单个字符。
json_path	text	可选的，若指定json_path，则在该路径下搜索。	json_path必须是一个有效的路径表达式。

返回类型：text

示例：

```
--all模式
gaussdb=# SELECT json_search('{"a":"abc","b":"abc"}','all','abc');
  json_search
----------------
 ["$.a", "$.b"]
(1 row)

--one模式
gaussdb=# SELECT json_search('{"a":"abc","b":"abc"}','one','abc');
 json_search
--------------
 "$.a"
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
--使用默认转义字符
gaussdb=# SELECT json_search('{"a":"abc","b":"a%c"}','one','a\%c');
 json_search
-------------
 "$.b"
(1 row)
```

- json_set(json, [VARIADIC any])

  描述：入参为一个json对象，若干个json_path路径和需要设置的值，json_path和值必须两两配对；此函数在一个json对象中使用指定的值来更新指定json_path路径上的数据，并返回修改后的json；如果指定的路径不存在值，则在对应的路径上插入值。json和json_path路径中有null值返回null，如果json格式错误或者json_path不是有效的路径表达式或者json_path包含“*”、“**”会报错。如果有null值和格式错误场景同时在可变参数列表中，则按异常的先后顺序处理，先有null值返回null，先有报错则报错。

  参数：如**表1-90**所示。

  表 1-90 参数说明

参数说明	类型	描述	取值范围
json	JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2	一个json值。如果传入其他非限定的类型，执行时将报错。	必须满足json的格式。
[VARIADIC any]	可变any数组	可变参数列表，可以支持一个或多个json_path-value对。	json_path必须是一个有效的路径表达式，value可以是任意值。

  返回类型：json

  示例：

```
gaussdb=# SELECT json_set('{"s":3}','$.s','d');
  json_set
------------
 {"s": "d"}
(1 row)

gaussdb=# SELECT json_set('{"s":3}','$.a','d','$.a','1');
      json_set
--------------------
 {"a": "1", "s": 3}
(1 row)
```

- json_type(json)

  描述：函数返回一个字符串，它代表了给定的json对象的类型。入参为null返回null。

版权所有 © 华为云计算技术有限公司

参数：一个json对象，仅支持JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2类型，若传入其他非限定类型，执行时将报错。

返回类型：text，详情请参见**表1-91**。

表 1-91 json_type 返回值

json对象中元素的类型	返回值
数组类型	ARRAY
对象类型	OBJECT
字符串类型	STRING
数值类型	NUMBER
boolean类型	BOOLEAN
NULL	NULL

示例：

```
gaussdb=# SELECT json_type('{"w":{"2":3},"2":4}');
 json_type
-----------
 OBJECT
(1 row)

gaussdb=# SELECT json_type('[1,2,2,3,3,4,4,4,4,4,4,4,4]');
 json_type
-----------
 ARRAY
(1 row)
```

- json_valid(json)

  描述：函数返回0和1来指示给定的参数是否是一个有效的json。入参为null返回null。

  参数：一个json对象，仅支持JSON、JSONB、CSTRING、TEXT、"char"、VARCHAR、BPCHAR、NVARCHAR2类型，若传入其他非null值的限定类型，执行时将始终返回0。

  返回类型：bigint

  示例：

```
gaussdb=# SELECT json_valid('{"name":"Tom"}');
 json_valid
------------
          1
(1 row)


gaussdb=# SELECT json_valid('[1,23,4,5,5]');
 json_valid
------------
          1
(1 row)

gaussdb=# SELECT json_valid('[1,23,4,5,5]}');
 json_valid
------------
```


版权所有 © 华为云计算技术有限公司

```
        0
(1 row)
```

- json_agg(any)

  描述：将值聚集为json数组。

  返回类型：array-json

  示例：

```
gaussdb=# CREATE TABLE classes(name varchar, score int);
CREATE TABLE
gaussdb=# INSERT INTO classes VALUES('A',2);
INSERT 0 1
gaussdb=# INSERT INTO classes VALUES('A',3);
INSERT 0 1
gaussdb=# INSERT INTO classes VALUES('D',5);
INSERT 0 1
gaussdb=# INSERT INTO classes VALUES('D',null);
INSERT 0 1
gaussdb=# SELECT * FROM classes;
 name | score
------+-------
 A    |     2
 A    |     3
 D    |     5
 D    |
(4 rows)

gaussdb=# SELECT name, json_agg(score) score FROM classes GROUP BY name ORDER BY name;
 name |   score
------+-----------
 A    | [2, 3]
 D    | [5, null]
(2 rows)

gaussdb=# DROP TABLE classes;
DROP TABLE
```

- json_object_agg(any, any)

  描述：将值聚集为json对象。

  返回类型：object-json

  示例：

```
gaussdb=# CREATE TABLE classes(name varchar, score int);
CREATE TABLE
gaussdb=# INSERT INTO classes VALUES('A',2);
INSERT 0 1
gaussdb=# INSERT INTO classes VALUES('A',3);
INSERT 0 1
gaussdb=# INSERT INTO classes VALUES('D',5);
INSERT 0 1
gaussdb=# INSERT INTO classes VALUES('D',null);
INSERT 0 1
gaussdb=# SELECT * FROM classes;
 name | score
------+-------
 A    |     2
 A    |     3
 D    |     5
 D    |
(4 rows)

gaussdb=# SELECT json_object_agg(name, score) FROM classes GROUP BY name ORDER BY name;
    json_object_agg
-------------------------
 { "A" : 2, "A" : 3 }
 { "D" : 5, "D" : null }
(2 rows)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE classes;
DROP TABLE
```

- jsonb_contained(jsonb, jsonb)

  描述：同操作符 `<@`, 判断第一个参数中的所有元素是否在第二个参数的顶层存在。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_contained('[1,2,3]', '[1,2,3,4]');
 jsonb_contained
-----------------
  t
 (1 row)
```

- jsonb_contains(jsonb, jsonb)

  描述：同操作符 `@>`, 判断第二个参数中的所有元素是否在第一个参数的顶层存在。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_contains('[1,2,3,4]', '[1,2,3]');
 jsonb_contains
----------------
 t
(1 row)
```

- jsonb_exists(jsonb, text)

  描述：同操作符 `?`, 判断字符串的第二个参数是否以key/elem/scalar的形式存在于第一个参数的顶层。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_exists('["1",2,3]', '1');
 jsonb_exists
--------------
 t
(1 row)
```

- jsonb_exists_all(jsonb, text[])

  描述：同操作符 `?&`, 判断字符串数组的第二个参数中的所有元素，是否都以key/elem/scalar的形式存在于第一个参数的顶层。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_exists_all('["1","2",3]', '{1, 2}');
 jsonb_exists_all
------------------
 t
(1 row)
```

- jsonb_exists_any(jsonb, text[])

  描述：同操作符 `?|`, 判断参数二中是否有任意元素，以key/elem/scalar的形式存在于参数一的顶层。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_exists_any('["1","2",3]', '{1, 2, 4}');
 jsonb_exists_any
------------------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

- jsonb_cmp(jsonb, jsonb)

  描述：比较大小，正数代表大于，负数代表小于，0表示相等。

  返回类型：integer

  示例：

```
gaussdb=# SELECT jsonb_cmp('["a", "b"]', '{"a":1, "b":2}');
 jsonb_cmp
-----------
          -1
(1 row)
```

- jsonb_eq(jsonb, jsonb)

  描述：同操作符 `=`, 比较两个值的大小是否相等。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_eq('["a", "b"]', '{"a":1, "b":2}');
 jsonb_eq
----------
 f
(1 row)
```

- jsonb_ne(jsonb, jsonb)

  描述：同操作符 `<>`, 比较两个值的大小是否不相等。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_ne('["a", "b"]', '{"a":1, "b":2}');
 jsonb_ne
----------
 t
(1 row)
```

- jsonb_gt(jsonb, jsonb)

  描述：同操作符 `>`, 比较第一个参数的值是否大于第二个参数的值。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_gt('["a", "b"]', '{"a":1, "b":2}');
 jsonb_gt
----------
 f
(1 row)
```

- jsonb_ge(jsonb, jsonb)

  描述：同操作符 `>=`, 比较第一个参数的值是否大于等于第二个参数的值。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_ge('["a", "b"]', '{"a":1, "b":2}');
 jsonb_ge
----------
 f
(1 row)
```

- jsonb_lt(jsonb, jsonb)

  描述：同操作符 `<`, 比较第一个参数的值是否小于第二个参数的值。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_lt('["a", "b"]', '{"a":1, "b":2}');
 jsonb_lt
----------
```


版权所有 © 华为云计算技术有限公司

```
 t
(1 row)
```

- jsonb_le(jsonb, jsonb)

  描述：同操作符 `<=`, 比较第一个参数的值是否小于等于第二个参数的值。

  返回类型：boolean

  示例：

```
gaussdb=# SELECT jsonb_le('["a", "b"]', '{"a":1, "b":2}');
 jsonb_le
----------
 t
(1 row)
```

- to_json(anyelement)

  描述：把参数转换为`json`。

  返回类型：json

  示例：

```
gaussdb=# SELECT to_json('{1,5}'::text[]);
  to_json
-----------
 ["1","5"]
(1 row)
```

- to_jsonb(anyelement)

  描述：把输入的anyelement参数转换为`jsonb`类型。

  返回类型：jsonb

  须知

  - 当参数为空字符串（''）时，若数据库SQL兼容模式为ORA，则返回结果为NULL；若数据库SQL兼容模式为PG, 则返回结果为空字符串。这是因为在ORA兼容模式下，空字符串被当作NULL处理。
  - 当参数为DATE类型时，若数据库SQL兼容模式为ORA，则返回结果包含时分秒；若数据库SQL兼容模式为PG, 则返回结果不包含时分秒。这是因为在ORA兼容模式下，当开关mapping_date_to_datea关闭时，DATE会被替换为TIMESTAMP(0) WITHOUT TIME ZONE。

  示例：

```
gaussdb=# SELECT to_jsonb(ARRAY[1, 2, 3, 4]);
   to_jsonb
---------------
 [1, 2, 3, 4]
(1 row)
```

- jsonb_hash(jsonb)

  描述：对jsonb进行hash运算。

  返回类型：integer

  示例：

```
gaussdb=# SELECT jsonb_hash('[1,2,3]');
 jsonb_hash
------------
 -559968547
(1 row)
```

- 其他函数

  描述：json\jsonb聚集函数所用到的内部函数。

版权所有 © 华为云计算技术有限公司

```
json_agg_transfn
json_agg_finalfn
json_object_agg_transfn
json_object_agg_finalfn
```

1.6.14 HLL 函数和操作符

哈希函数

- hll_hash_boolean(bool)

  描述：对bool类型数据计算哈希值。

  返回值类型：hll_hashval

  示例：

```
gaussdb=# SELECT hll_hash_boolean(FALSE);
 hll_hash_boolean
---------------------
 -5451962507482445012
(1 row)
```

- hll_hash_boolean(bool, int32)

  描述：设置hash seed（即改变哈希策略）并对bool类型数据计算哈希值。

  返回值类型：hll_hashval

  示例：

```
gaussdb=# SELECT hll_hash_boolean(FALSE, 10);
 hll_hash_boolean
--------------------
 -1169037589280886076
(1 row)
```

- hll_hash_smallint(smallint)

  描述：对smallint类型数据计算哈希值。

  返回值类型：hll_hashval

  示例：

```
gaussdb=# SELECT hll_hash_smallint(100::smallint);
 hll_hash_smallint
---------------------
 962727970174027904
(1 row)
```

  说明

  数值大小相同的参数使用不同数据类型的哈希函数计算，最后结果会不一样，因为不同类型哈希函数会选取不同的哈希计算策略。

- hll_hash_smallint(smallint, int32)

  描述：设置hash seed（即改变哈希策略）同时对smallint类型数据计算哈希值。

  返回值类型：hll_hashval

  示例：

```
gaussdb=# SELECT hll_hash_smallint(100::smallint, 10);
 hll_hash_smallint
---------------------
 -9056177146160443041
(1 row)
```

- hll_hash_integer(integer)

  描述：对integer类型数据计算哈希值。

  返回值类型：hll_hashval

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT hll_hash_integer(0);
  hll_hash_integer
----------------------
 5156626420896634997
(1 row)
```

- hll_hash_integer(integer, int32)

  描述：对integer类型数据计算哈希值，并设置hashseed（即改变哈希策略）。

  返回值类型：hll_hashval

  示例：

  ```
  gaussdb=# SELECT hll_hash_integer(0, 10);
   hll_hash_integer
  --------------------
   -5035020264353794276
  (1 row)
  ```

- hll_hash_bigint(bigint)

  描述：对bigint类型数据计算哈希值。

  返回值类型：hll_hashval

  示例：

  ```
  gaussdb=# SELECT hll_hash_bigint(100::bigint);
    hll_hash_bigint
  ---------------------
   -2401963681423227794
  (1 row)
  ```

- hll_hash_bigint(bigint, int32)

  描述：对bigint类型数据计算哈希值，并设置hashseed（即改变哈希策略）。

  返回值类型：hll_hashval

  示例：

  ```
  gaussdb=# SELECT hll_hash_bigint(100::bigint, 10);
    hll_hash_bigint
  ---------------------
   -2305749404374433531
  (1 row)
  ```

- hll_hash_bytea(bytea)

  描述：对bytea类型数据计算哈希值。

  返回值类型：hll_hashval

  示例：

  ```
  gaussdb=# SELECT hll_hash_bytea(E'\\x');
   hll_hash_bytea
  ----------------
   0
  (1 row)
  ```

- hll_hash_bytea(bytea, int32)

  描述：对bytea类型数据计算哈希值，并设置hashseed（即改变哈希策略）。

  返回值类型：hll_hashval

  示例：

  ```
  gaussdb=# SELECT hll_hash_bytea(E'\\x', 10);
    hll_hash_bytea
  ---------------------
   7233188113542599437
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- hll_hash_text(text)
  描述：对text类型数据计算哈希值。
  返回值类型：hll_hashval
  示例：

```
gaussdb=# SELECT hll_hash_text('AB');
    hll_hash_text
----------------------
 -5666002586880275174
(1 row)
```

- hll_hash_text(text, int32)
  描述：对text类型数据计算哈希值, 并设置hashseed（即改变哈希策略）。
  返回值类型：hll_hashval
  示例：

```
gaussdb=# SELECT hll_hash_text('AB', 10);
hll_hash_text
----------------------
-2215507121143724132
(1 row)
```

- hll_hash_any(anytype)
  描述：对任意类型数据计算哈希值。
  返回值类型：hll_hashval
  示例：

```
gaussdb=# SELECT hll_hash_any(1);
     hll_hash_any
----------------------
 -1316670585935156930
(1 row)

gaussdb=# SELECT hll_hash_any('08:00:2b:01:02:03'::macaddr);
     hll_hash_any
----------------------
 -3719950434455589360
(1 row)
```

- hll_hash_any(anytype, int32)
  描述：对任意类型数据计算哈希值，并设置hashseed（即改变哈希策略）。
  返回值类型：hll_hashval
  示例：

```
gaussdb=# SELECT hll_hash_any(1, 10);
     hll_hash_any
----------------------
 7048553517657992351
(1 row)
```

- hll_hashval_eq(hll_hashval, hll_hashval)
  描述：比较两个hll_hashval类型数据是否相等。
  返回值类型：bool
  示例：

```
gaussdb=# SELECT hll_hashval_eq(hll_hash_integer(1), hll_hash_integer(1));
 hll_hashval_eq
----------------
 t
(1 row)
```

- hll_hashval_ne(hll_hashval, hll_hashval)

 版权所有 © 华为云计算技术有限公司

描述：比较两个hll_hashval类型数据是否不相等。

返回值类型：bool

示例：

```
gaussdb=# SELECT hll_hashval_ne(hll_hash_integer(1), hll_hash_integer(1));
 hll_hashval_ne
----------------
 f
(1 row)
```

日志函数

hll主要存在三种模式Explicit、Sparse和Full。当数据规模比较小的时候会使用Explicit模式，这种模式下distinct值的计算是没有误差的；随着distinct值越来越多，hll会先后转换为Sparse模式和Full模式，这两种模式在计算结果上没有任何区别，只影响hll函数的计算效率和hll对象的存储空间。以下函数可以用于查看hll的一些参数。

- hll_print(hll)

  描述：打印hll的一些debug参数信息。

  示例：

  ```
  gaussdb=# SELECT hll_print(hll_empty());
                                  hll_print
  ------------------------------------------------------------------------------
   type=1(HLL_EMPTY), log2m=14, log2explicit=10, log2sparse=12, duplicatecheck=0
  (1 row)
  ```

- hll_type(hll)

  描述：查看当前hll的类型。

  返回值具体含义如下：

  - 0表示HLL_UNINIT，未初始化的hll对象。
  - 1表示HLL_EMPTY，hll空对象。
  - 2表示HLL_EXPLICIT，Explicit模式的hll对象。
  - 3表示HLL_SPARSE，Sparse模式的hll对象。
  - 4表示HLL_FULL，Full模式的hll对象。
  - 5表示HLL_UNDEFINED，不合法的hll对象。

  示例：

  ```
  gaussdb=# SELECT hll_type(hll_empty());
   hll_type
  ----------
         1
  (1 row)
  ```

- hll_log2m(hll)

  描述：查看当前hll数据结构中的log2m数值，log2m是分桶数的对数值，此值会影响最后hll计算distinct误差率，误差率计算公式为±1.04/√(2 ^ log2m)。当显式指定log2m的取值为10-16之间时，hll会设置分桶数为2^{log2m}。当显式指定log2explicit为-1时，会采用内置默认值。

  示例：

  ```
  gaussdb=# SELECT hll_log2m(hll_empty());
   hll_log2m
  -----------
         14
  (1 row)

  gaussdb=# SELECT hll_log2m(hll_empty(10));
  ```


版权所有 © 华为云计算技术有限公司

```
 hll_log2m
-----------
        10
(1 row)

gaussdb=# SELECT hll_log2m(hll_empty(-1));
 hll_log2m
-----------
        14
(1 row)
```

- hll_log2explicit(hll)

  描述：查看当前hll数据结构中的log2explicit数值。hll通常会由Explicit模式到Sparse模式再到Full模式，这个过程称为promotion hierarchy策略。可以通过调整log2explicit值的大小改变策略，比如log2explicit为0的时候就会跳过Explicit模式而直接进入Sparse模式。当显式指定log2explicit的取值为1-12之间时，hll会在数据段长度超过$2^{log2explicit}$时转为Sparse模式。当显式指定log2explicit为-1时，会采用内置默认值。

  示例：

```
gaussdb=# SELECT hll_log2explicit(hll_empty());
 hll_log2explicit
------------------
              10
(1 row)

gaussdb=# SELECT hll_log2explicit(hll_empty(12, 8));
 hll_log2explicit
------------------
               8
(1 row)

gaussdb=# SELECT hll_log2explicit(hll_empty(12, -1));
 hll_log2explicit
------------------
              10
(1 row)
```

- hll_log2sparse(hll)

  描述：查看当前hll数据结构中的log2sparse数值。hll通常会由Explicit模式到Sparse模式再到Full模式，这个过程称为promotion hierarchy策略。可以通过调整log2sparse值的大小改变策略，比如log2sparse为0的时候就会跳过Sparse模式而直接进入Full模式。当显式指定log2sparse的取值为1-14之间时，hll会在数据段长度超过$2^{log2sparse}$时转为Full模式。当显式指定log2sparse为-1时，会采用内置默认值。

  示例：

```
gaussdb=# SELECT hll_log2sparse(hll_empty());
 hll_log2sparse
----------------
             12
(1 row)

gaussdb=# SELECT hll_log2sparse(hll_empty(12, 8, 10));
 hll_log2sparse
----------------
             10
(1 row)

gaussdb=# SELECT hll_log2sparse(hll_empty(12, 8, -1));
 hll_log2sparse
----------------
             12
(1 row)
```


版权所有 © 华为云计算技术有限公司

- hll_duplicatecheck(hll)

  描述：是否启用duplicatecheck，0是关闭，1是开启。默认关闭，对于有较多重复值出现的情况，可以开启以提高效率。当显式指定duplicatecheck为-1时，会采用内置默认值。

  示例：

```
gaussdb=# SELECT hll_duplicatecheck(hll_empty());
 hll_duplicatecheck
--------------------
                  0
(1 row)

gaussdb=# SELECT hll_duplicatecheck(hll_empty(12, 8, 10, 1));
 hll_duplicatecheck
--------------------
                  1
(1 row)

gaussdb=# SELECT hll_duplicatecheck(hll_empty(12, 8, 10, -1));
 hll_duplicatecheck
--------------------
                  0
(1 row)
```

功能函数

- hll_empty()

  描述：创建一个空的hll。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_empty();
                                hll_empty
--------------------------------------------------------------------
 \x484c4c00000000002b050000000000000000000000000000000000000000
(1 row)
```

- hll_empty(int32 log2m)

  描述：创建空的hll并指定参数log2m，取值范围是10到16。若输入-1，则采用内置默认值。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_empty(10);
                                hll_empty
--------------------------------------------------------------------
 \x484c4c00000000002b040000000000000000000000000000000000000000
(1 row)

gaussdb=# SELECT hll_empty(-1);
                                hll_empty
--------------------------------------------------------------------
 \x484c4c00000000002b050000000000000000000000000000000000000000
(1 row)
```

- hll_empty(int32 log2m, int32 log2explicit)

  描述：创建空的hll并依次指定参数log2m、log2explicit。log2explicit取值范围是0到12，0表示直接跳过Explicit模式。该参数可以用来设置Explicit模式的阈值大小，在数据段长度达到$2^{log2explicit}$后切换为Sparse模式或者Full模式。若输入-1，则log2explicit采用内置默认值。

  返回值类型: hll

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT hll_empty(10, 4);
                                                            hll_empty
------------------------------------------------------------------------------------------------------------------
 \x484c4c0000000000130400000000000000000000000000000000000000000000
(1 row)

gaussdb=# SELECT hll_empty(10, -1);
                                                            hll_empty
------------------------------------------------------------------------------------------------------------------
 \x484c4c00000000002b040000000000000000000000000000000000000000000000
(1 row)
```

- hll_empty(int32 log2m, int32 log2explicit, int64 log2sparse)

  描述：创建空的hll并依次指定参数log2m、log2explicit、log2sparse。log2sparse取值范围是0到14，0表示直接跳过Sparse模式。该参数可以用来设置Sparse模式的阈值大小，在数据段长度达到$2^{log2sparse}$后切换为Full模式。若输入-1，则log2sparse采用内置默认值。

  返回值类型：hll

  示例：

  ```
  gaussdb=# SELECT hll_empty(10, 4, 8);
                                                              hll_empty
  ------------------------------------------------------------------------------------------------------------------
   \x484c4c0000000000120400000000000000000000000000000000000000000000
  (1 row)

  gaussdb=# SELECT hll_empty(10, 4, -1);
                                                              hll_empty
  ------------------------------------------------------------------------------------------------------------------
   \x484c4c0000000000130400000000000000000000000000000000000000000000
  (1 row)
  ```

- hll_empty(int32 log2m, int32 log2explicit, int64 log2sparse, int32 duplicatecheck)

  描述：创建空的hll并依次指定参数log2m、log2explicit、log2sparse、duplicatecheck。duplicatecheck取0或者1，表示是否开启该模式，默认情况下该模式会关闭。若输入-1，则duplicatecheck采用内置默认值。

  返回值类型：hll

  示例：

  ```
  gaussdb=# SELECT hll_empty(10, 4, 8, 0);
                                                              hll_empty
  ------------------------------------------------------------------------------------------------------------------
   \x484c4c0000000000120400000000000000000000000000000000000000000000
  (1 row)

  gaussdb=# SELECT hll_empty(10, 4, 8, -1);
                                                              hll_empty
  ------------------------------------------------------------------------------------------------------------------
   \x484c4c0000000000120400000000000000000000000000000000000000000000
  (1 row)
  ```

- hll_add(hll, hll_hashval)

  描述：把hll_hashval加入到hll中。

  返回值类型：hll

  示例：

  ```
  gaussdb=# SELECT hll_add(hll_empty(), hll_hash_integer(1));
                                  hll_add
  ------------------------------------------------------------------------------
   \x484c4c08000002002b0900000000000000f03f3e2921ff133fbaed3e2921ff133fbaed00
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- hll_add_rev(hll_hashval, hll)

  描述：把hll_hashval加入到hll中，和hll_add功能一样，只是参数位置进行了交换。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_add_rev(hll_hash_integer(1), hll_empty());
                                hll_add_rev
---------------------------------------------------------------------------
 \x484c4c08000002002b0900000000000000f03f3e2921ff133fbaed3e2921ff133fbaed00
(1 row)
```

- hll_eq(hll, hll)

  描述：比较两个hll是否相等。

  返回值类型：bool

  示例：

```
gaussdb=# SELECT hll_eq(hll_add(hll_empty(), hll_hash_integer(1)), hll_add(hll_empty(),
hll_hash_integer(2)));
 hll_eq
--------
 f
(1 row)
```

- hll_ne(hll, hll)

  描述：比较两个hll是否不相等。

  返回值类型：bool

  示例：

```
gaussdb=# SELECT hll_ne(hll_add(hll_empty(), hll_hash_integer(1)), hll_add(hll_empty(),
hll_hash_integer(2)));
 hll_ne
--------
 t
(1 row)
```

- hll_cardinality(hll)

  描述：计算hll的distinct值。

  返回值类型：int

  示例：

```
gaussdb=# SELECT hll_cardinality(hll_empty() || hll_hash_integer(1));
 hll_cardinality
-----------------
               1
(1 row)
```

- hll_union(hll, hll)

  描述：把两个hll数据结构union成一个。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_union(hll_add(hll_empty(), hll_hash_integer(1)), hll_add(hll_empty(),
hll_hash_integer(2)));
                                    hll_union
-------------------------------------------------------------------------------------------

\x484c4c10002000002b0900000000000000004000000000000000000b3ccc49320cca1ae3e2921ff133fba
ed00
(1 row)
```


版权所有 © 华为云计算技术有限公司

聚合函数

- hll_add_agg(hll_hashval)

  描述：把哈希后的数据按照分组放到hll中。

  返回值类型：hll

  示例：

```
--准备数据
gaussdb=# CREATE TABLE t_id(id int);
gaussdb=# INSERT INTO t_id VALUES(generate_series(1,500));
gaussdb=# CREATE TABLE t_data(a int, c text);
gaussdb=# INSERT INTO t_data SELECT mod(id,2), id FROM t_id;

--创建表并指定列为hll
gaussdb=# CREATE TABLE t_a_c_hll(a int, c hll);

--根据a列group by对数据分组，把各组数据加到hll中
gaussdb=# INSERT INTO t_a_c_hll SELECT a, hll_add_agg(hll_hash_text(c)) FROM t_data GROUP BY a;

--得到每组数据中hll的Distinct值
gaussdb=# SELECT a, #c AS cardinality FROM t_a_c_hll ORDER BY a;
 a |    cardinality
---+-------------------
 0 | 247.862354346299
 1 | 250.908710610377
(2 rows)
```

- hll_add_agg(hll_hashval, int32 log2m)

  描述：把哈希后的数据按照分组放到hll中， 并指定参数log2m，取值范围是10到16。若输入-1或者NULL，则采用内置默认值。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_cardinality(hll_add_agg(hll_hash_text(c), 12)) FROM t_data;
 hll_cardinality
------------------
 497.965240179228
(1 row)
```

- hll_add_agg(hll_hashval, int32 log2m, int32 log2explicit)

  描述：把哈希后的数据按照分组放到hll中，依次指定参数log2m、log2explicit。log2explicit取值范围是0到12，0表示直接跳过Explicit模式。该参数可以用来设置Explicit模式的阈值大小，在数据段长度达到$2^{log2explicit}$后切换为Sparse模式或者Full模式。若输入-1或者NULL，则log2explicit采用内置默认值。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_cardinality(hll_add_agg(hll_hash_text(c), NULL, 1)) FROM t_data;
 hll_cardinality
------------------
 498.496062953313
(1 row)
```

- hll_add_agg(hll_hashval, int32 log2m, int32 log2explicit, int64 log2sparse)

  描述：把哈希后的数据按照分组放到hll中， 依次指定参数log2m、log2explicit、log2sparse。log2sparse取值范围是0到14，0表示直接跳过Sparse模式。该参数可以用来设置Sparse模式的阈值大小，在数据段长度达到$2^{log2sparse}$后切换为Full模式。若输入-1或者NULL，则log2sparse采用内置默认值。

  返回值类型：hll

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT hll_cardinality(hll_add_agg(hll_hash_text(c), NULL, 6, 10)) FROM t_data;
 hll_cardinality
------------------
 498.496062953313
(1 row)
```

- hll_add_agg(hll_hashval, int32 log2m, int32 log2explicit, int64 log2sparse, int32 duplicatecheck)

  描述：把哈希后的数据按照分组放到hll中, 依次指定参数log2m、log2explicit、log2sparse、duplicatecheck，duplicatecheck取值范围是0或者1，表示是否开启该模式，默认情况下该模式会关闭。若输入-1或者NULL，则duplicatecheck采用内置默认值。

  返回值类型：hll

  示例：

```
gaussdb=# SELECT hll_cardinality(hll_add_agg(hll_hash_text(c), NULL, 6, 10, -1)) FROM t_data;
 hll_cardinality
------------------
 498.496062953313
(1 row)
```

- hll_union_agg(hll)

  描述：将多个hll类型数据union成一个hll。

  返回值类型：hll

  示例：

```
--将各组中的hll数据union成一个hll，并计算distinct值。
gaussdb=# SELECT #hll_union_agg(c) AS cardinality FROM t_a_c_hll;
   cardinality
------------------
 498.496062953313
(1 row)

--删除表
gaussdb=# DROP TABLE t_id;
gaussdb=# DROP TABLE t_data;
gaussdb=# DROP TABLE t_a_c_hll;
```

  说明

  注意：当两个或者多个hll数据结构做union的时候，必须要保证其中每一个hll里面的精度参数一样，否则将不可以进行union。同样的约束也适用于函数hll_union(hll,hll)。

废弃函数

由于版本升级，HLL（HyperLogLog）有一些旧的函数废弃，用户可以用类似的函数进行替代。

- hll_schema_version(hll)

  描述：查看当前hll中的schema version。旧版本schema version是常值1，用来进行hll字段的头部校验，重构后的hll在头部增加字段“HLL”进行校验，schema version不再使用。

- hll_regwidth(hll)

  描述：查看hll数据结构中桶的位数大小。旧版本桶的位数regwidth取值1~5，会存在较大的误差，也限制了基数估计上限。 重构后regwidth为固定值6，不再使用regwidth变量。

- hll_expthresh(hll)

  描述：得到当前hll中expthresh大小。采用hll_log2explicit(hll)替代类似功能。

版权所有 © 华为云计算技术有限公司

- hll_sparseon(hll)

  描述：是否启用Sparse模式。采用hll_log2sparse(hll)替代类似功能，0表示关闭Sparse模式。

内置函数

HLL（HyperLogLog）有一系列内置函数用于内部对数据进行处理，一般情况下用户不需要熟知这些函数的使用。详情见表1-92。

表 1-92 内置函数

函数名称	功能描述
hll_in	以string格式接收hll数据。
hll_out	以string格式发送hll数据。
hll_recv	以bytea格式接收hll数据。
hll_send	以bytea格式发送hll数据。
hll_trans_in	以string格式接收hll_trans_type数据。
hll_trans_out	以string格式发送hll_trans_type数据。
hll_trans_recv	以bytea形式接收hll_trans_type数据。
hll_trans_send	以bytea形式发送hll_trans_type数据。
hll_typmod_in	接收typmod类型数据。
hll_typmod_out	发送typmod类型数据。
hll_hashval_in	接收hll_hashval类型数据。
hll_hashval_out	发送hll_hashval类型数据。
hll_add_trans0	类似于hll_add所提供的功能，初始化时无指定入参，通常在分布式聚合运算的第一阶段DN上使用。
hll_add_trans1	类似于hll_add所提供的功能，初始化时指定一个入参，通常在分布式聚合运算的第一阶段DN上使用。
hll_add_trans2	类似于hll_add所提供的功能，初始化时指定两个入参，通常在分布式聚合运算的第一阶段DN上使用。
hll_add_trans3	类似于hll_add所提供的功能，初始化时指定三个入参，通常在分布式聚合运算的第一阶段DN上使用。
hll_add_trans4	类似于hll_add所提供的功能，初始化时指定四个入参，通常在分布式聚合运算的第一阶段DN上使用。
hll_union_trans	类似hll_union所提供的功能，在分布式聚合运算的第一阶段DN上使用。
hll_union_collect	类似于hll_union所提供的功能，在分布式聚合运算第二阶段CN上使用，汇总各个DN上的结果。

版权所有 © 华为云计算技术有限公司

函数名称	功能描述
hll_pack	在分布式聚合运算第三阶段CN上使用，把自定义hll_trans_type类型最后转换成hll类型。
hll	用于hll类型转换成hll类型，根据输入参数会设定指定参数。
hll_hashval	用于bigint类型转换成hll_hashval类型。
hll_hashval_int4	用于int4类型转换成hll_hashval类型。

操作符

- =

  描述：比较hll或hll_hashval的值是否相等。

  返回值类型：bool

  示例：

```
--hll
gaussdb=# SELECT (hll_empty() || hll_hash_integer(1)) = (hll_empty() || hll_hash_integer(1));
 ?column?
----------
 t
(1 row)

--hll_hashval
gaussdb=# SELECT hll_hash_integer(1) = hll_hash_integer(1);
 ?column?
----------
 t
(1 row)
```

- <> or !=

  描述：比较hll或hll_hashval是否不相等。

  返回值类型：bool

  示例：

```
--hll
gaussdb=# SELECT (hll_empty() || hll_hash_integer(1)) <> (hll_empty() || hll_hash_integer(2));
 ?column?
----------
 t
(1 row)

--hll_hashval
gaussdb=# SELECT hll_hash_integer(1) <> hll_hash_integer(2);
 ?column?
----------
 t
(1 row)
```

- ||

  描述：可代表hll_add, hll_union, hll_add_rev三个函数的功能。

  返回值类型：hll

  示例：

```
--hll_add
gaussdb=# SELECT hll_empty() || hll_hash_integer(1);
                                 ?column?
--------------------------------------------------------------------------------
```


版权所有 © 华为云计算技术有限公司

```
 \x484c4c08000002002b0900000000000000f03f3e2921ff133fbaed3e2921ff133fbaed00
(1 row)

--hll_add_rev
gaussdb=# SELECT hll_hash_integer(1) || hll_empty();
                              ?column?
-------------------------------------------------------------------------------
 \x484c4c08000002002b0900000000000000f03f3e2921ff133fbaed3e2921ff133fbaed00
(1 row)

--hll_union
gaussdb=# SELECT (hll_empty() || hll_hash_integer(1)) || (hll_empty() || hll_hash_integer(2));
                                        ?column?
-----------------------------------------------------------------------------------------------

\x484c4c10002000002b0900000000000000004000000000000000000b3ccc49320cca1ae3e2921ff133fba
ed00
(1 row)
```

- #

  描述：计算出hll的distinct值，同hll_cardinality函数。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT #(hll_empty() || hll_hash_integer(1));
   ?column?
  ----------
         1
  (1 row)
  ```

1.6.15 SEQUENCE 函数

序列函数为用户从序列对象中获取后续的序列值提供了简单的多用户安全的方法。

- nextval(regclass)

  描述：递增序列并返回新值。

  说明

  - 为了避免从同一个序列获取值的并发事务被阻塞， nextval操作不会回滚；即一旦值被抓取就认为其已经被使用过，并且不会再被返回。 即使该操作处于事务中，当事务之后中断，或者如果调用查询结束不使用该值，也是如此。这种情况将在指定值的顺序中留下未使用的“空洞”。 因此，GaussDB序列对象不能用于获得“无间隙”序列。
  - 如果nextval被下推到DN上时，各个DN会自动连接GTM，请求next values值，例如（insert into t1 select xxx，t1某一列需要调用nextval函数），由于GTM上有最大连接数为8192的限制，而这类下推语句会导致消耗过多的GTM连接数，因此对于这类语句的并发数目限制为7000（其它语句需要占用部分连接）/集群DN数目。
  - 如果插入语句中同时包含SRF（Set-Returning-Functions）函数和序列自增列，则序列的值会跳过一位（即跳号）。SRF函数以generate_series为例：

    ```
    CREATE TABLE t1 (a int, b serial);
    INSERT INTO t1(a) VALUES(generate_series(1,1)); -- 插入数据b列为1
    INSERT INTO t1(a) VALUES(generate_series(1,1));  -- 插入数据b列为3，跳号2
    ```

  - 对于rowid系统列的序列，nextval()仅支持在维护模式下操作。

  返回类型：numeric

  nextval函数有两种调用方式（其中第二种调用方式兼容ORA的语法，目前不支持Sequence命名中有特殊字符"."的情况），如下：

  ```
  gaussdb=# CREATE SEQUENCE seqDemo;

  --示例1:
  gaussdb=# SELECT nextval('seqDemo');
   nextval
  ```


版权所有 © 华为云计算技术有限公司

```
---------
       1
(1 row)

--示例2:
gaussdb=# SELECT seqDemo.nextval;
 nextval
---------
       2
(1 row)
gaussdb=# DROP SEQUENCE seqDemo;
```

- currval(regclass)

  描述：返回当前会话里最近一次nextval返回的数值。如果当前会话还没有调用过指定的sequence的nextval，那么调用currval将会报错。需要注意的是，这个函数在默认情况下是不支持的，需要通过设置enable_beta_features为true之后，才能使用这个函数。同时在设置enable_beta_features为true之后，nextval()函数将不支持下推。

  返回类型：numeric

  currval函数有两种调用方式（其中第二种调用方式兼容ORA的语法，目前不支持Sequence命名中有特殊字符"."的情况），如下：

```
gaussdb=# CREATE SEQUENCE seq1;
gaussdb=# SELECT nextval('seq1');
gaussdb=# SET enable_beta_features = true;

--示例1:
gaussdb=# SELECT currval('seq1');
 currval
---------
       1
(1 row)

--示例2:
gaussdb=# SELECT seq1.currval seq1;
 seq1
------
    1
(1 row)
gaussdb=# DROP SEQUENCE seq1;
gaussdb=# SET enable_beta_features = false;
```

- lastval()

  描述：返回当前会话里最近一次nextval返回的数值。这个函数等效于currval，只是它不用序列名为参数，它抓取当前会话里面最近一次nextval使用的序列。如果当前会话还没有调用过nextval，那么调用lastval将会报错。

  需要注意的是，这个函数在默认情况下是不支持的，需要通过设置enable_beta_features或者lastval_supported为true之后，才能使用这个函数。同时这种情况下，nextval()函数将不支持下推。

  返回类型：numeric

  示例：

```
gaussdb=# CREATE SEQUENCE seq1;
gaussdb=# SELECT nextval('seq1');
gaussdb=# SET enable_beta_features = true;
gaussdb=# SELECT lastval();
 lastval
---------
       1
(1 row)
gaussdb=# DROP SEQUENCE seq1;
gaussdb=# SET enable_beta_features = false;
```


版权所有 © 华为云计算技术有限公司

- setval(regclass, bigint)

  描述：设置序列的当前数值。

  返回类型：numeric

  示例：

  ```
  gaussdb=# CREATE SEQUENCE seqDemo;
  gaussdb=# SELECT nextval('seqDemo');
  gaussdb=# SELECT setval('seqDemo',3);
   setval
  --------
       3
  (1 row)
  gaussdb=# DROP SEQUENCE seqDemo;
  ```

- setval(regclass, numeric, Boolean)

  描述：设置序列的当前数值以及is_called标志。

  返回类型：numeric

  示例：

  ```
  gaussdb=# CREATE SEQUENCE seqDemo;
  gaussdb=# SELECT nextval('seqDemo');
  gaussdb=# SELECT setval('seqDemo',5,true);
   setval
  --------
       5
  (1 row)
  gaussdb=# DROP SEQUENCE seqDemo;
  ```

  说明

  - 使用该函数后：新值在GTM上会立即生效；当cache缓存值大于session_sequence_cache缓存值时，当前会话会继续获取缓存直到cache缓存值耗尽；当cache缓存值小于等于session_sequence_cache缓存值时，新值在当前会话会立即生效，但如果其他会话有缓存的序列值，只能等到缓存值用尽新值才会生效。为避免序列值冲突，应谨慎使用该函数。
  - 由于序列是非事务的，使用该函数所造成的改变不会由于事务的回滚而撤销。
  - 对于rowid系统列的序列，该函数仅支持在维护模式下使用。

- pg_sequence_last_value(sequence_oid oid, OUT cache_value int16, OUT last_value int16)

  描述：获取指定sequence的参数，包含缓存值和当前值。

  返回类型：int16、int16

- gs_get_sequence_parameters(sequence_oid, OUT start_value int16, OUT minimum_value int16, OUT maximum_value int16, OUT increment int16, OUT cycle_option Boolean, OUT order_option Boolean)

  描述：获取指定sequence的参数，包含起始值、最小值、最大值、步长、循环选项以及有序选项。

  返回类型：int16、int16、int16、int16、boolean、boolean

- gs_get_sequence_last_value(sequence_oid oid, OUT cache_value int16, OUT last_value int16)

  描述：获取指定sequence的参数，包含缓存值以及当前值。

  返回类型：int16、int16

- get_global_last_value_from_gtm(sequence_regclass regclass, OUT is_called boolean, OUT last_value numeric)

  描述：从GTM获取序列是否调用过的标记位和已申请的序列最新值。

版权所有 © 华为云计算技术有限公司

参数：表示待查询序列的序列名或OID。

返回类型：boolean、numeric

说明

- 从不支持get_global_last_value_from_gtm的版本升级到支持get_global_last_value_from_gtm的版本时，在升级观察期期间，不可使用此函数。
- get_global_last_value_from_gtm函数每次调用都需要与GTM进行通信，不建议频繁执行。
- 序列已申请的最新值，不一定表示已使用的最大值。如果序列的cache大于1，且节点申请的序列值没有用完，这种情况将会留下未使用的“空洞”。

1.6.16 数组函数和操作符

数组操作符

- =

  描述：两个数组是否相等。

  示例：

```
gaussdb=# SELECT ARRAY[1.1,2.1,3.1]::int[] = ARRAY[1,2,3] AS RESULT ;
 result
--------
 t
(1 row)
```

- <>

  描述：两个数组是否不相等。

  示例：

```
gaussdb=# SELECT ARRAY[1,2,3] <> ARRAY[1,2,4] AS RESULT;
 result
--------
 t
(1 row)
```

- <

  描述：一个数组是否小于另一个数组。

  示例：

```
gaussdb=# SELECT ARRAY[1,2,3] < ARRAY[1,2,4] AS RESULT;
 result
--------
 t
(1 row)
```

- >

  描述：一个数组是否大于另一个数组。

  示例：

```
gaussdb=# SELECT ARRAY[1,4,3] > ARRAY[1,2,4] AS RESULT;
 result
--------
 t
(1 row)
```

- <=

  描述：一个数组是否小于或等于另一个数组。

  示例：

```
gaussdb=# SELECT ARRAY[1,2,3] <= ARRAY[1,2,3] AS RESULT;
 result
```


版权所有 © 华为云计算技术有限公司

```
--------
 t
(1 row)
```

- >=

  描述：一个数组是否大于或等于另一个数组。

  示例：

```
gaussdb=# SELECT ARRAY[1,4,3] >= ARRAY[1,4,3] AS RESULT;
 result
--------
 t
(1 row)
```

- @>

  描述：一个数组是否包含另一个数组。

  示例：

```
gaussdb=# SELECT ARRAY[1,4,3] @> ARRAY[3,1] AS RESULT;
 result
--------
 t
(1 row)
```

- <@

  描述：一个数组是否被包含于另一个数组。

  示例：

```
gaussdb=# SELECT ARRAY[2,7] <@ ARRAY[1,7,4,2,6] AS RESULT;
 result
--------
 t
(1 row)
```

- &&

  描述：一个数组是否和另一个数组重叠（有共同元素）。

  示例：

```
gaussdb=# SELECT ARRAY[1,4,3] && ARRAY[2,1] AS RESULT;
 result
--------
 t
(1 row)
```

- ||

  描述：数组与数组进行连接。

  示例：

```
gaussdb=# SELECT ARRAY[1,2,3] || ARRAY[4,5,6] AS RESULT;
    result
---------------
 {1,2,3,4,5,6}
(1 row)
gaussdb=# SELECT ARRAY[1,2,3] || ARRAY[[4,5,6],[7,8,9]] AS RESULT;
          result
-----------------------------
 {{1,2,3},{4,5,6},{7,8,9}}
(1 row)
```

- ||

  描述：元素与数组进行连接。

  示例：

```
gaussdb=# SELECT 3 || ARRAY[4,5,6] AS RESULT;
  result
-----------
```


版权所有 © 华为云计算技术有限公司

```
 {3,4,5,6}
(1 row)
```

- ||

  描述：数组与元素进行连接。

  示例：

```
gaussdb=# SELECT ARRAY[4,5,6] || 7 AS RESULT;
  result
-----------
 {4,5,6,7}
(1 row)
```

数组比较是使用默认的B-tree比较函数对所有元素逐一进行比较的。多维数组的元素按照行顺序进行访问。如果两个数组的内容相同但维数不等，决定排序顺序的首要因素是维数。

数组函数

- array_append(anyarray, anyelement)

  描述：向数组末尾添加元素，只支持一维数组。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_append(ARRAY[1,2], 3) AS RESULT;
 result
---------
 {1,2,3}
(1 row)
```

- array_prepend(anyelement, anyarray)

  描述：向数组开头添加元素，只支持一维数组。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_prepend(1, ARRAY[2,3]) AS RESULT;
 result
---------
 {1,2,3}
(1 row)
```

- array_cat(anyarray, anyarray)

  描述：连接两个数组，支持多维数组。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_cat(ARRAY[1,2,3], ARRAY[4,5]) AS RESULT;
   result
-------------
 {1,2,3,4,5}
(1 row)

gaussdb=# SELECT array_cat(ARRAY[[1,2],[4,5]], ARRAY[6,7]) AS RESULT;
      result
---------------------
 {{1,2},{4,5},{6,7}}
(1 row)
```

- array_union(anyarray, anyarray)

  描述：连接两个数组，只支持一维数组。有入参为NULL时返回另一个入参。

  返回类型：anyarray

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT array_union(ARRAY[1,2,3], ARRAY[3,4,5]) AS RESULT;
   result
-------------
 {1,2,3,3,4,5}
(1 row)

gaussdb=# SELECT array_union(ARRAY[1,2,3], NULL) AS RESULT;
 result
---------
 {1,2,3}
(1 row)
```

- array_union_distinct(anyarray, anyarray)

  描述：连接两个数组，并去重，只支持一维数组。有入参为NULL时返回另一个入参。不为NULL的入参仅支持一维数组，多维数组会报错。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_union_distinct(ARRAY[1,2,3], ARRAY[3,4,5]) AS RESULT;
   result
-------------
 {1,2,3,4,5}
(1 row)

gaussdb=# SELECT array_union_distinct(ARRAY[1,2,3], NULL) AS RESULT;
 result
---------
 {1,2,3}
(1 row)
```

- array_intersect(anyarray, anyarray)

  描述：两个数组取交集，只支持一维数组。有入参为NULL时返回NULL。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_intersect(ARRAY[1,2,3], ARRAY[3,4,5]) AS RESULT;
   result
-------------
 {3}
(1 row)

gaussdb=# SELECT array_intersect(ARRAY[1,2,3], NULL) AS RESULT;
 result
--------

(1 row)
```

- array_intersect_distinct(anyarray, anyarray)

  描述：两个数组取交集，并去重，只支持一维数组。有入参为NULL时返回NULL。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_intersect_distinct(ARRAY[1,2,2], ARRAY[2,2,4,5]) AS RESULT;
   result
-------------
 {2}
(1 row)

gaussdb=# SELECT array_intersect_distinct(ARRAY[1,2,3], NULL) AS RESULT;
 result
--------

(1 row)
```


版权所有 © 华为云计算技术有限公司

- array_except(anyarray, anyarray)

  描述：两个数组取差，只支持一维数组。第一个入参为NULL时返回NULL, 第二个入参为NULL时返回第一个入参。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_except(ARRAY[1,2,3], ARRAY[3,4,5]) AS RESULT;
   result
-------------
 {1,2}
(1 row)

gaussdb=# SELECT array_except(ARRAY[1,2,3], NULL) AS RESULT;
 result
---------
 {1,2,3}
(1 row)

gaussdb=# SELECT array_except(NULL, ARRAY[3,4,5]) AS RESULT;
 result
--------

(1 row)
```

- array_except_distinct(anyarray, anyarray)

  描述：两个数组取差，并去重，只支持一维数组。第一个入参为NULL时返回NULL, 第二个入参为NULL时返回第一个入参（第一个入参仅支持一维数组，多维数组会报错）。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_except_distinct(ARRAY[1,2,2,3], ARRAY[3,4,5]) AS RESULT;
   result
-------------
 {1,2}
(1 row)

gaussdb=# SELECT array_except_distinct(ARRAY[1,2,3], NULL) AS RESULT;
 result
---------
 {1,2,3}
(1 row)

gaussdb=# SELECT array_except_distinct(NULL, ARRAY[3,4,5]) AS RESULT;
 result
--------

(1 row)
```

- array_ndims(anyarray)

  描述：返回数组的维数。

  返回类型：int

  示例：

```
gaussdb=# SELECT array_ndims(ARRAY[[1,2,3], [4,5,6]]) AS RESULT;
 result
--------
      2
(1 row)
```

- array_dims(anyarray)

  描述：返回数组各个维度中的下界和上界。

  返回类型：text

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT array_dims(ARRAY[[1,2,3], [4,5,6]]) AS RESULT;
   result
------------
 [1:2][1:3]
(1 row)
```

- array_length(anyarray, int)

  描述：返回数组维度的长度。

  返回类型：int

  示例：

```
gaussdb=# SELECT array_length(array[1,2,3], 1) AS RESULT;
 result
--------
      3
(1 row)
```

- array_lower(anyarray, int)

  描述：返回数组维数的下界。

  返回类型：int

  示例：

```
gaussdb=# SELECT array_lower('[0:2]={1,2,3}'::int[], 1) AS RESULT;
 result
--------
      0
(1 row)
```

- array_sort(anyarray)

  描述：返回从小到大排列好的数组。只支持一维anyarray数组排序，多维数组返回NULL值。目前暂不支持record、xml、xmltype、json类型数组排序。

  返回类型：anyarray

  示例：

```
gaussdb=# SELECT array_sort(ARRAY[5,1,3,6,2,7]) AS RESULT;
 result
-------------
{1,2,3,5,6,7}
(1 row)
gaussdb=# SELECT array_sort(array[array[1,23], array[1,34]]);
 array_sort
-------------
 {NULL,NULL}
(1 row)
```

- array_upper(anyarray, int)

  描述：返回数组维数的上界。

  返回类型：int

  示例：

```
gaussdb=# SELECT array_upper(ARRAY[1,8,3,7], 1) AS RESULT;
 result
--------
      4
(1 row)
```

- array_to_string(anyarray, text [, text])

  描述：使用第一个text作为数组的新分隔符，使用第二个text替换数组值为NULL的值。

  返回类型：text

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT array_to_string(ARRAY[1, 2, 3, NULL, 5], ',', '*') AS RESULT;
 result
-----------
 1,2,3,*,5
(1 row)
```

- array_delete(anyarray)

  描述：清空数组中的元素并返回一个同类型的空数组。

  返回类型：anyarray

  示例：

  ```
  gaussdb=# SELECT array_delete(ARRAY[1,8,3,7]) AS RESULT;
   result
  --------
   {}
  (1 row)
  ```

- array_deleteidx(anyarray, int)

  描述：从数组中删除指定下标的元素并返回剩余元素组成的数组。

  返回类型：anyarray

  示例：

  ```
  gaussdb=# SELECT array_deleteidx(ARRAY[1,2,3,4,5], 1) AS RESULT;
   result
  -----------
   {2,3,4,5}
  (1 row)
  ```

  说明

  array_deleteidx(anyarray, int)此函数在参数a_format_version值为10c和a_format_dev_version值为s1的情况下被禁用。

- array_extendnull(anyarray, int)

  描述：往数组尾部添加指定个数空元素。

  返回类型：anyarray

  示例：

  ```
  gaussdb=# SELECT array_extendnull(ARRAY[1,8,3,7],1) AS RESULT;
     result
  --------------
  {1,8,3,7,null}
  (1 row)
  ```

- array_extendnull(anyarray, int, int)

  描述：往数组尾部添加指定个数的指定索引的元素。

  返回类型：anyarray

  示例：

  ```
  gaussdb=# SELECT array_extendnull(ARRAY[1,8,3,7],2,2) AS RESULT;
     result
  --------------
  {1,8,3,7,8,8}
  (1 row)
  ```

  说明

  array_extendnull(anyarray, int, int)此函数在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。

- array_trim(anyarray, int)

  描述：从数组尾部移除指定数量的元素。

版权所有 © 华为云计算技术有限公司

返回类型：anyarray

示例：

```
gaussdb=# SELECT array_trim(ARRAY[1,8,3,7],1) AS RESULT;
 result
---------
 {1,8,3}
(1 row)
```

- array_exists(anyarray, int)

  描述：检查第二个参数是否是数组的合法下标。

  返回类型：boolean

  示例：

  ```
  gaussdb=# SELECT array_exists(ARRAY[1,8,3,7],1) AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- array_next(anyarray, int)

  描述：根据第二个入参返回数组中指定下标元素的下一个元素的下标。

  返回类型：int

  示例：

  ```
  gaussdb=# SELECT array_next(ARRAY[1,8,3,7],1) AS RESULT;
   result
  --------
       2
  (1 row)
  ```

- array_prior(anyarray, int)

  描述：根据第二个入参返回数组中指定下标元素的上一个元素的下标。

  返回类型：int

  示例：

  ```
  gaussdb=# SELECT array_prior(ARRAY[1,8,3,7],2) AS RESULT;
   result
  --------
    1
  (1 row)
  ```

- string_to_array(text, text [, text])

  描述：使用第二个text指定分隔符，使用第三个可选的text作为NULL值替换模板，如果分隔后的子串与第三个可选的text完全匹配，则将其替换为NULL。

  返回类型：text[]

  示例：

  ```
  gaussdb=# SELECT string_to_array('xx~^~yy~^~zz', '~^~', 'yy') AS RESULT;
     result
  --------------
   {xx,NULL,zz}
  (1 row)
  gaussdb=# SELECT string_to_array('xx~^~yy~^~zz', '~^~', 'y') AS RESULT;
    result
  ------------
   {xx,yy,zz}
  (1 row)
  ```

  在string_to_array中，如果分隔符参数是NULL，输入字符串中的每个字符将在结果数组中变成一个独立的元素。如果分隔符是一个空白字符串，则整个输入的字符串将变为一个元素的数组。否则输入字符串将在每个分隔字符串处分开。

版权所有 © 华为云计算技术有限公司

在string_to_array中，如果省略NULL字符串参数或为NULL，将字符串中没有输入内容的子串替换为NULL。

在array_to_string中，如果省略NULL字符串参数或为NULL，运算中将跳过在数组中的任何NULL元素，并且不会在输出字符串中出现。

- unnest(anyarray)

  描述：扩大一个数组为一组行。

  返回类型：setof anyelement

  示例：

```
gaussdb=# SELECT unnest(ARRAY[1,2]) AS RESULT;
 result
--------
      1
      2
(2 rows)
```

- _pg_keysequal

  描述：判断两个smallint数组是否相同。

  参数：smallint[], smallint[]

  返回值类型：boolean

  说明

  此函数存在于information_schema命名空间。

- cardinality(anyarray)

  描述：返回数组中各个维度上元素的总数，如果数组为空则返回0。

  返回类型：integer

  示例：

```
gaussdb=# SELECT cardinality(array[[1, 2], [3, 4]]);
 cardinality
-------------
           4
(1 row)
```

- array_positions(anyarray, anyelement)

  描述：返回作为第一个参数给出的数组中所有出现的第二个参数的下标的数组。

  返回类型：int[]

  示例：

```
gaussdb=# SELECT array_positions(array[1, 2, 3, 1], 1) AS RESULT;
 result
--------
 {1,4}
(1 row)
```

  说明

  - 数组必须是一维的。
  - 第二个参数可以设置为NULL。
  - 如果数组中找不到第二个参数，返回空数组。

版权所有 © 华为云计算技术有限公司

1.6.17 范围函数和操作符

范围操作符

- =

  描述：等于。

  示例：

```
gaussdb=# SELECT int4range(1,5) = '[1,4]'::int4range AS RESULT;
 result
--------
 t
(1 row)
```

- <>

  描述：不等于。

  示例：

```
gaussdb=# SELECT numrange(1.1,2.2) <> numrange(1.1,2.3) AS RESULT;
 result
--------
 t
(1 row)
```

- <

  描述：小于。

  示例：

```
gaussdb=# SELECT int4range(1,10) < int4range(2,3) AS RESULT;
 result
--------
 t
(1 row)
```

- >

  描述：大于。

  示例：

```
gaussdb=# SELECT int4range(1,10) > int4range(1,5) AS RESULT;
 result
--------
 t
(1 row)
```

- <=

  描述：小于或等于。

  示例：

```
gaussdb=# SELECT numrange(1.1,2.2) <= numrange(1.1,2.2) AS RESULT;
 result
--------
 t
(1 row)
```

- >=

  描述：大于或等于。

  示例：

```
gaussdb=# SELECT numrange(1.1,2.2) >= numrange(1.1,2.0) AS RESULT;
 result
--------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

- @>

  描述：包含范围。

  示例：

  ```
  gaussdb=# SELECT int4range(2,4) @> int4range(2,3) AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- @>

  描述：包含元素。

  示例：

  ```
  gaussdb=# SELECT '[2011-01-01,2011-03-01)'::tsrange @> '2011-01-10'::timestamp AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- <@

  描述：范围包含于。

  示例：

  ```
  gaussdb=# SELECT int4range(2,4) <@ int4range(1,7) AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- <@

  描述：元素包含于。

  示例：

  ```
  gaussdb=# SELECT 42 <@ int4range(1,7) AS RESULT;
   result
  --------
   f
  (1 row)
  ```

- &&

  描述：重叠（有共同点）。

  示例：

  ```
  gaussdb=# SELECT int8range(3,7) && int8range(4,12) AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- <<

  描述：范围值是否比另一个范围值的最小值还小（没有交集）。

  示例：

  ```
  gaussdb=# SELECT int8range(1,10) << int8range(100,110) AS RESULT;
   result
  --------
   t
  (1 row)
  ```

- >>

  描述：范围值是否比另一个范围值的最大值还大（没有交集）。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT int8range(50,60) >> int8range(20,30) AS RESULT;
 result
--------
 t
(1 row)
```

- &<

  描述：范围值的最大值是否不超过另一个范围值的最大值。

  示例：

```
gaussdb=# SELECT int8range(1,20) &< int8range(18,20) AS RESULT;
 result
--------
 t
(1 row)
```

- &>

  描述：范围值的最小值是否不小于另一个范围值的最小值。

  示例：

```
gaussdb=# SELECT int8range(7,20) &> int8range(5,10) AS RESULT;
 result
--------
 t
(1 row)
```

- -|-

  描述：相邻。

  示例：

```
gaussdb=# SELECT numrange(1.1,2.2) -|- numrange(2.2,3.3) AS RESULT;
 result
--------
 t
(1 row)
```

- +

  描述：并集。

  示例：

```
gaussdb=# SELECT numrange(5,15) + numrange(10,20) AS RESULT;
 result
--------
 [5,20)
(1 row)
```

- *

  描述：交集。

  示例：

```
gaussdb=# SELECT int8range(5,15) * int8range(10,20) AS RESULT;
 result
---------
 [10,15)
(1 row)
```

- -

  描述：差集。

  示例：

```
gaussdb=# SELECT int8range(5,15) - int8range(10,20) AS RESULT;
 result
--------
 [5,10)
(1 row)
```

 版权所有 © 华为云计算技术有限公司

简单的比较操作符<，>，<=和>=先比较下界，只有下界相等时才比较上界。

<<、>>和-|-操作符当包含空范围时也会返回false；即不认为空范围在其他范围之前或之后。

并集和差集操作符的执行结果无法包含两个不相交的子范围。

范围函数

如果范围是空或者需要的界限是无穷的，lower和upper函数将返回null。lower_inc、upper_inc、lower_inf和upper_inf函数均对空范围返回false。

- numrange(numeric, numeric, [text])

  描述：表示一个范围。

  返回类型：范围元素类型

  示例：

```
gaussdb=# SELECT numrange(1.1,2.2) AS RESULT;
 result
--------
[1.1,2.2)
(1 row)
gaussdb=# SELECT numrange(1.1,2.2, '()') AS RESULT;
 result
--------
(1.1,2.2)
(1 row)
```

- lower(anyrange)

  描述：范围的下界。

  返回类型：范围元素类型

  示例：

```
gaussdb=# SELECT lower(numrange(1.1,2.2)) AS RESULT;
 result
--------
    1.1
(1 row)
```

- upper(anyrange)

  描述：范围的上界。

  返回类型：范围元素类型

  示例：

```
gaussdb=# SELECT upper(numrange(1.1,2.2)) AS RESULT;
 result
--------
    2.2
(1 row)
```

- isempty(anyrange)

  描述：范围是否为空。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT isempty(numrange(1.1,2.2)) AS RESULT;
 result
--------
 f
(1 row)
```


版权所有 © 华为云计算技术有限公司

- lower_inc(anyrange)
  描述：是否包含下界。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT lower_inc(numrange(1.1,2.2)) AS RESULT;
 result
--------
 t
(1 row)
```

- upper_inc(anyrange)
  描述：是否包含上界。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT upper_inc(numrange(1.1,2.2)) AS RESULT;
 result
--------
 f
(1 row)
```

- lower_inf(anyrange)
  描述：下界是否为无穷。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT lower_inf('(,)'::daterange) AS RESULT;
 result
--------
 t
(1 row)
```

- upper_inf(anyrange)
  描述：上界是否为无穷。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT upper_inf('(,)'::daterange) AS RESULT;
 result
--------
 t
(1 row)
```

- elem_contained_by_range(anyelement, anyrange)
  描述：判断元素是否在范围内。
  返回类型：Boolean
  示例：

```
gaussdb=# SELECT elem_contained_by_range('2', numrange(1.1,2.2));
 elem_contained_by_range
-------------------------
 t
(1 row)
```

  说明

  在ORA兼容模式下，开启mapping_date_to_datea参数后，date类型会映射为datea类型，datea类型不支持作为范围类型，因此该函数不支持datea类型作为入参。

- in_range_int16_int16(int16, int16, int16, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_int16_int16(1::int16, 1::int16, 1::int16, true, true);
 in_range_int16_int16
----------------------
 f
(1 row)

gaussdb=# SELECT in_range_int16_int16(1::int16, 4::int16, 2::int16, true, true);
 in_range_int16_int16
----------------------
 t
(1 row)
```

- in_range_int8_int16(int8, int8, int16, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int8_int16(1::int8, 1::int8, 1::int16, true, true);
   in_range_int8_int16
  ```


版权所有 © 华为云计算技术有限公司

```
---------------------
 f
(1 row)

gaussdb=# SELECT in_range_int8_int16(1::int8, 4::int8, 2::int16, true, true);
 in_range_int8_int16
---------------------
 t
(1 row)
```

- in_range_int4_int16(int4, int4, int16, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_int4_int16(1::int4, 1::int4, 1::int16, true, true);
 in_range_int4_int16
---------------------
 f
(1 row)

gaussdb=# SELECT in_range_int4_int16(1::int4, 4::int4, 2::int16, true, true);
 in_range_int4_int16
---------------------
 t
(1 row)
```

- in_range_int2_int16(int2, int2, int16, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。

 版权所有 © 华为云计算技术有限公司

    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_int2_int16(1::int2, 1::int2, 1::int16, true, true);
 in_range_int2_int16
---------------------
 f
(1 row)

gaussdb=# SELECT in_range_int2_int16(1::int2, 4::int2, 2::int16, true, true);
 in_range_int2_int16
---------------------
 t
```

- in_range_int1_int16(int1, int1, int16, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

```
gaussdb=# SELECT in_range_int1_int16(1::int1, 1::int1, 1::int16, true, true);
 in_range_int1_int16
---------------------
 f
(1 row)

gaussdb=# SELECT in_range_int1_int16(1::int1, 4::int1, 2::int16, true, true);
 in_range_int1_int16
---------------------
 t
(1 row)
```

- in_range_float8_float8(float8, float8, float8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。

版权所有 © 华为云计算技术有限公司

- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_float8(1.0::float8, 1.9::float8, 1.0::float8, true, true);
 in_range_float8_float8
------------------------
 f
(1 row)
```

- in_range_float4_float8(float4, float4, float8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_float8(1.0::float4, 1.9::float4, 1.0::float8, true, true);
   in_range_float4_float8
  ------------------------
   f
  (1 row)
  ```

- in_range_numeric_numeric(numeric, numeric, numeric, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。

版权所有 © 华为云计算技术有限公司

- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_numeric_numeric(1.0::numeric(10,4), 1.9::numeric(10,4),
1.0::numeric(10,4), true, true);
 in_range_numeric_numeric
--------------------------
 f
(1 row)
```

- in_range_interval_interval(interval, interval, interval, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_interval_interval(interval '13' hour, interval '10' hour, interval '3' hour,
  false, true);
   in_range_interval_interval
  ----------------------------
   t
  (1 row)
  ```

- in_range_timestamp_interval(timestamp, timestamp, interval, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

版权所有 © 华为云计算技术有限公司

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

```
gaussdb=# SELECT in_range_timestamp_interval('12-10-2010 11:25:00', '10-11-2010 11:26:00', interval
'60' day, false, true);
 in_range_timestamp_interval
-----------------------------
 t
(1 row)
```

- in_range_timestamptz_interval(timestamptz, timestamptz, interval, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_timestamptz_interval('12-10-2010 11:25:00+00', '10-11-2010
  11:26:00+00', interval '60' day, false, true);
   in_range_timestamptz_interval
  -------------------------------
   t
  (1 row)
  ```

- in_range_time_interval(time, time, interval, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

版权所有 © 华为云计算技术有限公司

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_time_interval('00:00:00', '12:00:00', interval '13' hour, true, true);
 in_range_time_interval
------------------------
 f
(1 row)
```

- in_range_timetz_interval(timetz, timetz, interval, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_timetz_interval('00:00:00+00', '12:00:00+00', interval '13' hour, true, true);
 in_range_timetz_interval
--------------------------
 f
(1 row)
```

- in_range_int8_numeric(int8, int8, numeric, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

版权所有 © 华为云计算技术有限公司

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_int8_numeric(1::int8, 2::int8, 0.9::numeric(10,4), true, true);
 in_range_int8_numeric
-----------------------
 t
(1 row)
```

- in_range_int4_numeric(int4, int4, numeric, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_int4_numeric(1::int4, 2::int4, 0.9::numeric(10,4), true, true);
 in_range_int4_numeric
-----------------------
 t
(1 row)
```

- in_range_int2_numeric(int2, int2, numeric, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

版权所有 © 华为云计算技术有限公司

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_int2_numeric(1::int2, 2::int2, 0.9::numeric(10,4), true, true);
 in_range_int2_numeric
-----------------------
 t
(1 row)
```

- in_range_int1_numeric(int1, int1, numeric, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_int1_numeric(1::int1, 2::int1, 0.9::numeric(10,4), true, true);
 in_range_int1_numeric
-----------------------
 t
(1 row)
```

- in_range_numeric_float4(numeric, numeric, float4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

版权所有 © 华为云计算技术有限公司

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_numeric_float4(1.0::numeric(10,4), 1.9::numeric(10,4), 1.0::float4, true,
true);
 in_range_numeric_float4
-------------------------
 f
(1 row)
```

- in_range_numeric_float8(numeric, numeric, float8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_numeric_float8(1.0::numeric(10,4), 1.9::numeric(10,4), 1.0::float8, true,
true);
 in_range_numeric_float8
-------------------------
 f
(1 row)
```

- in_range_int8_float8(int8, int8, float8, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_int8_float8(1::int8, 2::int8, 0.9::float8, true, true);
 in_range_int8_float8
----------------------
 t
(1 row)
```

- in_range_int4_float8(int4, int4, float8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int4_float8(1::int4, 2::int4, 0.9::float8, true, true);
   in_range_int4_float8
  ----------------------
   t
  (1 row)
  ```

- in_range_int2_float8(int2, int2, float8, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_int2_float8(1::int2, 2::int2, 0.9::float8, true, true);
 in_range_int2_float8
----------------------
 t
(1 row)
```

- in_range_int1_float8(int1, int1, float8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int1_float8(1::int1, 2::int1, 0.9::float8, true, true);
   in_range_int1_float8
  ----------------------
   t
  (1 row)
  ```

- in_range_float8_int16(float8, float8, int16, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_int16(1.0::float8, 1.9::float8, 1::int16, true, true);
 in_range_float8_int16
-----------------------
 f
(1 row)
```

- in_range_float4_int16(float4, float4, int16, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_int16(1.0::float4, 1.9::float4, 1::int16, true, true);
   in_range_float4_int16
  -----------------------
   f
  (1 row)
  ```

- in_range_float8_int8(float8, float8, int8, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_int8(1.0::float8, 1.9::float8, 1::int8, true, true);
 in_range_float8_int8
----------------------
 f
(1 row)
```

- in_range_float4_int8(float4, float4, int8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_int8(1.0::float4, 1.9::float4, 1::int8, true, true);
   in_range_float4_int8
  ----------------------
   f
  (1 row)
  ```

- in_range_float8_int4(float8, float8, int4, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_int4(1.0::float8, 1.9::float8, 1::int4, true, true);
 in_range_float8_int4
----------------------
 f
(1 row)
```

- in_range_float4_int4(float4, float4, int4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_int4(1.0::float4, 1.9::float4, 1::int4, true, true);
   in_range_float4_int4
  ----------------------
   f
  (1 row)
  ```

- in_range_float8_int2(float8, float8, int2, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_int2(1.0::float8, 1.9::float8, 1::int2, true, true);
 in_range_float8_int2
----------------------
 f
(1 row)
```

- in_range_float4_int2(float4, int2, int2, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_int2(1.0::float4, 1.9::float4, 1::int2, true, true);
   in_range_float4_int2
  ----------------------
   f
  (1 row)
  ```

- in_range_float8_int1(float8, float8, int1, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_int1(1.0::float8, 1.9::float8, 1::int1, true, true);
 in_range_float8_int1
----------------------
 f
(1 row)
```

- in_range_float4_int1(float4, float4, int1, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_int1(1.0::float4, 1.9::float4, 1::int1, true, true);
   in_range_float4_int1
  ----------------------
   f
  (1 row)
  ```

- in_range_float8_float4(float8, float8, float4, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_float4(1.0::float8, 1.9::float8, 1.0::float4, true, true);
 in_range_float8_float4
------------------------
 f
(1 row)
```

- in_range_float4_float4(float4, float4, float4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_float4(1.0::float4, 1.9::float4, 1.0::float4, true, true);
   in_range_float4_float4
  ------------------------
   f
  (1 row)
  ```

- in_range_float8_numeric(float8, float8, numeric, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_float8_numeric(1.0::float8, 1.9::float8, 1.0::numeric(10,4), true, true);
 in_range_float8_numeric
-------------------------
 f
(1 row)
```

- in_range_float4_numeric(float4, float4, numeric, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_float4_numeric(1.0::float4, 1.9::float4, 1.0::numeric(10,4), true, true);
   in_range_float4_numeric
  -------------------------
   f
  (1 row)
  ```

- in_range_numeric_int16(numeric, numeric, int16, boolean, boolean)

版权所有 © 华为云计算技术有限公司

描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

参数：

- 第一个参数：需要被判断是否在范围内的数。
- 第二个参数：当前范围的起始值。
- 第三个参数：范围值，必须为正数。
- 第四个参数：当前操作为减操作或加操作。
  - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
  - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
- 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
  - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
  - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

返回类型：Boolean

示例：

```
gaussdb=# SELECT in_range_numeric_int16(1.0::numeric(10,4), 1.9::numeric(10,4), 1.0::int16, true,
true);
 in_range_numeric_int16
------------------------
 f
(1 row)
```

- in_range_numeric_int8(numeric, numeric, int8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT in_range_numeric_int8(1.0::numeric(10,4), 1.9::numeric(10,4), 1.0::int8, true, true);
 in_range_numeric_int8
------------------------
 f
(1 row)
```


版权所有 © 华为云计算技术有限公司

- in_range_numeric_int4(numeric, numeric, int4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_numeric_int4(1.0::numeric(10,4), 1.9::numeric(10,4), 1.0::int4, true, true);
   in_range_numeric_int4
  -----------------------
   f
  (1 row)
  ```

- in_range_numeric_int1(numeric, numeric, int1, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_numeric_int1(1.0::numeric(10,4), 1.9::numeric(10,4), 1.0::int1, true, true);
   in_range_numeric_int1
  -----------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int8_int8(int8, int8, int8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int8_int8(1::int8, 1::int8, 1::int8, true, true);
   in_range_int8_int8
  --------------------
   f
  (1 row)
  ```

- in_range_int4_int8(int4, int4, int8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int4_int8(1::int4, 1::int4, 1::int8, true, true);
   in_range_int4_int8
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int2_int8(int2, int2, int8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int2_int8(1::int2, 1::int2, 1::int8, true, true);
   in_range_int2_int8
  --------------------
   f
  (1 row)
  ```

- in_range_int1_int8(int1, int1, int8, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int1_int8(1::int1, 1::int1, 1::int8, true, true);
   in_range_int1_int8
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int8_int4(int8, int8, int4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int8_int4(1::int8, 1::int8, 1::int4, true, true);
   in_range_int8_int4
  --------------------
   f
  (1 row)
  ```

- in_range_int4_int4(int4, int4, int4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int4_int4(1::int4, 1::int4, 1::int4, true, true);
   in_range_int4_int4
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int2_int4(int2, int2, int4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int2_int4(1::int2, 1::int2, 1::int4, true, true);
   in_range_int2_int4
  --------------------
   f
  (1 row)
  ```

- in_range_int1_int4(int1, int1, int4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int1_int4(1::int1, 1::int1, 1::int4, true, true);
   in_range_int1_int4
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int8_int2(int8, int8, int2, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int8_int2(1::int8, 1::int8, 1::int2, true, true);
   in_range_int8_int2
  --------------------
   f
  (1 row)
  ```

- in_range_int4_int2(int4, int4, int2, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int4_int2(1::int4, 1::int4, 1::int2, true, true);
   in_range_int4_int2
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int2_int2(int2, int2, int2, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int2_int2(1::int2, 1::int2, 1::int2, true, true);
   in_range_int2_int2
  --------------------
   f
  (1 row)
  ```

- in_range_int1_int2(int1, int1, int2, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int1_int2(1::int1, 1::int1, 1::int2, true, true);
   in_range_int1_int2
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int8_int1(int8, int8, int1, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int8_int1(1::int8, 1::int8, 1::int1, true, true);
   in_range_int8_int1
  --------------------
   f
  (1 row)
  ```

- in_range_int4_int1(int4, int4, int1, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int4_int1(1::int4, 1::int4, 1::int1, true, true);
   in_range_int4_int1
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int2_int1(int2, int2, int1, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int2_int1(1::int2, 1::int2, 1::int1, true, true);
   in_range_int2_int1
  --------------------
   f
  (1 row)
  ```

- in_range_int1_int1(int1, int1, int1, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int1_int1(1::int1, 1::int1, 1::int1, true, true);
   in_range_int1_int1
  --------------------
   f
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int8_float4(int8, int8, float4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int8_float4(1::int8, 2::int8, 0.9::float4, true, true);
   in_range_int8_float4
  ----------------------
   t
  (1 row)
  ```

- in_range_int4_float4(int4, int4, float4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int4_float4(1::int4, 2::int4, 0.9::float4, true, true);
   in_range_int4_float4
  ----------------------
   t
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- in_range_int2_float4(int2, int2, float4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int2_float4(1::int2, 2::int2, 0.9::float4, true, true);
   in_range_int2_float4
  ----------------------
   t
  (1 row)
  ```

- in_range_int1_float4(int1, int1, float4, boolean, boolean)

  描述：判断第一个参数，是否在第二个参数加/减第三个参数范围内。

  参数：

  - 第一个参数：需要被判断是否在范围内的数。
  - 第二个参数：当前范围的起始值。
  - 第三个参数：范围值，必须为正数。
  - 第四个参数：当前操作为减操作或加操作。
    - TRUE：第二个参数减第三个参数再与第一个参数比较大小。
    - FALSE：第二个参数加第三个参数再与第一个参数比较大小。
  - 第五个参数：第二个参数和第三个参数的运算结果与第一个参数做比较。
    - TRUE：判断结果是否大于第一个参数。若结果大于等于第一个参数则函数返回值为TRUE，反之为FALSE。
    - FALSE：判断结果是否小于等于第一个参数。若结果小于等于第一个参数则函数返回值为TRUE，反之为FALSE。

  返回类型：Boolean

  示例：

  ```
  gaussdb=# SELECT in_range_int1_float4(1::int1, 2::int1, 0.9::float4, true, true);
   in_range_int1_float4
  ----------------------
   t
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

1.6.18 聚集函数

聚集函数

- sum(expression)

  描述：所有输入行的expression总和。

  返回类型：

  通常情况下输入数据类型和输出数据类型是相同的，但以下情况会发生类型转换：

  - 对于SMALLINT或INT输入，输出类型为BIGINT。
  - 对于BIGINT输入，输出类型为NUMBER 。
  - 对于浮点数输入，输出类型为DOUBLE PRECISION。

  示例：

```
gaussdb=# CREATE TABLE tab(a int);
CREATE TABLE
gaussdb=# INSERT INTO tab VALUES(1);
INSERT 0 1
gaussdb=# INSERT INTO tab VALUES(2);
INSERT 0 1
gaussdb=# SELECT sum(a) FROM tab;
 sum
-----
   3
(1 row)
gaussdb=# DROP TABLE tab;
```

- max(expression)

  描述：所有输入行中expression的最大值。

  参数：类型为任意数组、数值、字符串、日期/时间类型、ROWID类型。

  返回类型：与参数数据类型相同

  示例：

```
gaussdb=# CREATE TABLE max_t1(a int, b int);

gaussdb=# INSERT INTO max_t1 VALUES(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT MAX(a) FROM max_t1;
 max
-----
   4
(1 row)

gaussdb=# DROP TABLE max_t1;
```

- min(expression)

  描述：所有输入行中expression的最小值。

  参数：类型为任意数组、数值、字符串、日期/时间类型、ROWID类型。

  返回类型：与参数数据类型相同

  示例：

```
gaussdb=# CREATE TABLE min_t1(a int, b int);

gaussdb=# INSERT INTO min_t1 VALUES(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT MIN(a) FROM min_t1;
 min
-----
   1
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# DROP TABLE min_t1;
```

- avg(expression)

  描述：所有输入值的均值（算术平均）。

  返回类型：

  - 对于任何为整数类型或可隐式转换成整数类型的输入参数，函数返回结果为NUMBER类型。
  - 对于任何为浮点类型或可隐式转换成浮点类型的输入参数，函数返回结果为DOUBLE PRECISION类型。
  - 其他情况和输入参数的数据类型相同。

  示例：

```
gaussdb=# CREATE TABLE avg_t1(a int, b int);

gaussdb=# INSERT INTO avg_t1 VALUES(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT AVG(a) FROM avg_t1;
       avg
--------------------
 2.5000000000000000
(1 row)

gaussdb=# DROP TABLE avg_t1;
```

- count(expression)

  描述：返回表中满足expression不为NULL的行数。

  返回类型：BIGINT

  支持对XML类型数据操作。

  示例：

```
gaussdb=# CREATE TABLE count_t1(a int, b int);

gaussdb=# INSERT INTO count_t1 VALUES (NULL,1),(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT COUNT(a) FROM count_t1;
 count
-------
     4
(1 row)

gaussdb=# DROP TABLE count_t1;
```

- count(*)

  描述：返回表中的记录行数。

  返回类型：BIGINT

  支持对XML类型数据操作。

  示例：

```
gaussdb=# CREATE TABLE count_t1(a int, b int);

gaussdb=# INSERT INTO count_t1 VALUES (NULL,1),(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT COUNT(*) FROM count_t1;
 count
-------
     5
(1 row)

gaussdb=# DROP TABLE count_t1;
```


版权所有 © 华为云计算技术有限公司

- KEEP(DENSE_RANK { FIRST | LAST } ORDER BY expr)

  描述：将查询结果的数据行按指定规则进行排序，返回第一组或者最后一组的值，可用于聚集函数或窗口函数。

  语法：

  ```
  aggregate_function KEEP ( DENSE_RANK { FIRST | LAST }  ORDER BY expr [ DESC | ASC ] [ NULLS { FIRST | LAST } ]
      [, expr [ DESC | ASC ] [ NULLS { FIRST | LAST } ]]...)
  ```

  返回值类型: 在aggregate_function中指定列的数据类型。

  说明

  - aggregate_function为聚集函数，当前仅支持：MIN、MAX、SUM、AVG、COUNT、VARIANCE和STDDEV。
  - KEEP关键字用于语义清晰，限定于聚集函数/窗口函数。
  - DENSE_RANK FIRST或DENSE_RANK LAST表示仅聚集最小（FIRST）或最大（LAST）密集排序后的行。
  - 不支持生成向量化计划执行 。

  示例：

  ```
  gaussdb=# CREATE TABLE keep_t1(a int, b int, c int);

  gaussdb=# INSERT INTO keep_t1 values(2, 111, 333), (2, 11, 33), (2, 11, 3), (1, 22, 55), (1, 2, 5);

  gaussdb=#  SELECT MIN(b) KEEP(DENSE_RANK FIRST ORDER BY a) MIN_FIRST,
          MIN(b) KEEP(DENSE_RANK LAST ORDER BY a) MIN_LAST,
          MAX(b) KEEP(DENSE_RANK FIRST ORDER BY a) MAX_FIRST,
          MAX(b) KEEP(DENSE_RANK LAST ORDER BY a) MAX_LAST FROM keep_t1;
   min_first | min_last | max_first | max_last
  -----------+----------+-----------+----------
          2 |       11 |        22 |      111
  (1 row)

  gaussdb=# DROP TABLE keep_t1;
  ```

- default(column_name)

  描述：获取表字段的默认值。

  返回值类型：text

  示例：

  ```
  --创建兼容性MYSQL数据库。
  gaussdb=# CREATE DATABASE gaussdb_m WITH DBCOMPATIBILITY 'MYSQL';

  gaussdb=# \c gaussdb_m

  gaussdb_m=# CREATE TABLE t1(id int DEFAULT 100, name varchar(20) DEFAULT 'tt');

  gaussdb_m=# INSERT INTO t1 VALUES(1,'test');

  --执行查询。
  gaussdb_m=# SELECT default(id), default(name) FROM t1;
   default | default
  ---------+---------
       100 | tt
  (1 row)

  --删除数据库。
  gaussdb_m=# \c postgres
  gaussdb=# DROP DATABASE gaussdb_m;
  ```


版权所有 © 华为云计算技术有限公司

说明

- default函数仅在参数sql_compatibility='MYSQL'时生效。
- 表字段中不存在默认值时，default函数返回空值。
- 表字段是隐藏列（如xmin、cmin）时，default函数返回空值。
- 表字段是自增列时，default函数返回0。
- GaussDB支持分区表、临时表、多表连接查询默认值。
- GaussDB支持查询列名包含字符串值节点（表示名称）和A_Star节点（表示出现“*”），如default(tt.t4.id)和default(tt.t4.*)。
- GaussDB创建字段默认值时，如果没有检验字段类型的范围，使用default函数可能会报错。
- 字段的默认值是函数表达式时，GaussDB的default函数返回建表时字段的default表达式的计算值。

- array_agg(expression)

  描述：将所有输入值（包括空）连接成一个数组。

  返回类型：参数类型的数组

  支持对XML类型数据操作。

  示例：

```
gaussdb=# CREATE TABLE array_agg_t1(a int, b int);

gaussdb=# INSERT INTO array_agg_t1 VALUES (NULL,1),(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT ARRAY_AGG(a) FROM array_agg_t1;
  array_agg
----------------
 {NULL,1,2,3,4}
(1 row)

gaussdb=# DROP TABLE array_agg_t1;
```

- string_agg(expression, delimiter)

  描述：将输入值连接成为一个字符串，用分隔符分开。

  返回类型：和参数数据类型相同。

  支持对显示转换成字符类型后的XML类型数据操作。

  示例：

```
gaussdb=# CREATE TABLE string_agg_t1(a int, b int);

gaussdb=# INSERT INTO string_agg_t1 VALUES (NULL,1),(1,2),(2,3),(3,4),(4,5);

gaussdb=# SELECT STRING_AGG(a,';') FROM string_agg_t1;
 string_agg
------------
 1;2;3;4
(1 row)

gaussdb=# DROP TABLE string_agg_t1;
```

- listagg([DISTINCT | ALL] expression [, delimiter]) [WITHIN GROUP(ORDER BY order-list)]

  描述：将聚集列数据按WITHIN GROUP指定的排序方式排列，并用delimiter指定的分隔符拼接成一个字符串。

  - DISTINCT：可选。表示对每行拼接后结果进行去重，仅支持ORA兼容模式。
  - expression：必选。指定聚集列名或基于列的有效表达式，不支持VARIADIC参数。

版权所有 © 华为云计算技术有限公司

- delimiter：可选。指定分隔符，可以是字符串常数或基于分组列的确定性表达式，缺省时表示分隔符为空。
- order-list：可选。指定分组内的排序方式。

返回类型：text

说明

listagg为兼容Oracle 11g2的列转行聚集函数，可以指定OVER子句用作窗口函数。为了避免与listagg本身WITHIN GROUP子句的ORDER BY造成不同的含义/行为时，listagg用作窗口函数时，OVER子句不支持ORDER BY的窗口排序或窗口框架。

示例：

聚集列是文本字符集类型。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b text);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,'a1'),(1,'b2'),(1,'c3'),(2,'d4'),(2,'e5'),(3,'f6');

gaussdb=# SELECT a,LISTAGG(b,';') WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a | listagg
---+---------
 1 | b2;c3
 2 | d4;e5
 3 | f6
   | a1
(4 rows)

gaussdb=# SELECT a,LISTAGG(b,';') FROM listagg_t1 GROUP BY a ORDER BY a;
 a | listagg
---+---------
 1 | b2;c3
 2 | d4;e5
 3 | f6
   | a1
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

聚集列是整型。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b int);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,1),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT a,LISTAGG(b,';') WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a | listagg
---+---------
 1 | 2;3
 2 | 4;5
 3 | 6
   | 1
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

聚集列是浮点类型。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b float);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,1.111),(1,2.222),(1,3.333),(2,4.444),(2,5.555),
(3,6.666);

gaussdb=# SELECT a,LISTAGG(b,';') WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a |      listagg
---+-------------------
 1 | 2.222000;3.333000
```


版权所有 © 华为云计算技术有限公司

```
 2 | 4.444000;5.555000
 3 | 6.666000
   | 1.111000
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

聚集列是时间类型。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b timestamp);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,'2000-01-01'),(1,'2000-02-02'),(1,'2000-03-03'),
(2,'2000-04-04'),(2,'2000-05-05'),(3,'2000-06-06');

gaussdb=# SELECT a,LISTAGG(b,';') WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a |                 listagg
---+-----------------------------------------
 1 | 2000-02-02 00:00:00;2000-03-03 00:00:00
 2 | 2000-04-04 00:00:00;2000-05-05 00:00:00
 3 | 2000-06-06 00:00:00
   | 2000-01-01 00:00:00
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

聚集列是时间间隔类型。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'3 days'),(2,'4 days'),(2,'5
days'),(3,'6 days');

gaussdb=# SELECT a,LISTAGG(b,';') WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a |    listagg
---+---------------
 1 | 2 days;3 days
 2 | 4 days;5 days
 3 | 6 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

聚集列是ROWID类型。

```
gaussdb=# CREATE TABLE test_table_1(id int, name varchar(29), rwd rowid);
CREATE TABLE
gaussdb=# INSERT INTO test_table_1 VALUES (1, 'AAAAAAAAAP//AAAAAAAAAAAAAAAAAA',
'AAAAAAAAAP//AAAAAAAAAAAAAAAAAA');
INSERT 0 1
gaussdb=# INSERT INTO test_table_1 VALUES (2, 'AAAAAAAAAP//AAAAAAAAAAAAAAAABA',
'AAAAAAAAAP//AAAAAAAAAAAAAAAABA');
INSERT 0 1
gaussdb=# INSERT INTO test_table_1 VALUES (3, 'AAAAAAAAAP//AAAAAAAAAAAAAAACAA',
'AAAAAAAAAP//AAAAAAAAAAAAAAACAA');
INSERT 0 1
gaussdb=# -- 5.listagg函数
gaussdb=# SELECT LISTAGG(rwd, ';') WITHIN GROUP(ORDER BY rwd) FROM test_table_1;
                                    listagg
-------------------------------------------------------------------------------------
 AAAAAAAAAP//AAAAAAAAAAAAAAAAAAAAAAAAAAAP//AAAAAAAAAAAAAAABAAAAAAAAAAP//
AAAAAAAAAAAAAACAA
(1 row)
gaussdb=# DROP TABLE test_table_1;
DROP TABLE
```

分隔符缺省时，默认为空。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'3 days'),(2,'4 days'),(2,'5
```


版权所有 © 华为云计算技术有限公司

```
days'),(3,'6 days');

gaussdb=# SELECT a,LISTAGG(b) WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a ORDER
BY a;
 a |   listagg
---+-------------
 1 | 2 days3 days
 2 | 4 days5 days
 3 | 6 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

使用distinct去重。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval);

gaussdb=# INSERT INTO listagg_t1 VALUES(1,'4 days'),(1,'3 days'),(2,'1 days'),(1,'5 days'),(3,'2 days'),
(1,'4 days'),(NULL,'1 days'),(2,'1 days');

gaussdb=# SELECT a,listagg(distinct b,',') FROM listagg_t1 GROUP BY a ORDER BY a;
 a |       listagg
---+---------------------
 1 | 3 days,4 days,5 days
 2 | 1 day
 3 | 2 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

使用WITHIN GROUP(ORDER BY order-list)排序。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval, c int);

gaussdb=# INSERT INTO listagg_t1 VALUES(1,'4 days',4),(1,'3 days',3),(2,'1 days',1),(1,'5 days',1),(3,'2
days',1),(1,'4 days',2),(NULL,'1 days',1),(2,'1 days',2);

gaussdb=# SELECT a,listagg(distinct b,',') WITHIN GROUP(ORDER BY b) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a |       listagg
---+---------------------
 1 | 3 days,4 days,5 days
 2 | 1 day
 3 | 2 days
   | 1 day
(4 rows)

gaussdb=# SELECT a,listagg(distinct b,',') WITHIN GROUP(ORDER BY c) FROM listagg_t1 GROUP BY a
ORDER BY a;
 a |       listagg
---+---------------------
 1 | 5 days,4 days,3 days
 2 | 1 day
 3 | 2 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

不使用WITHIN GROUP(ORDER BY order-list)排序，拼接结果无法保证有序。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval, c int);

gaussdb=# INSERT INTO listagg_t1 VALUES(1,'4 days',4),(1,'3 days',3),(2,'1 days',1),(1,'5 days',1),(3,'2
days',1),(1,'4 days',2),(NULL,'1 days',1),(2,'1 days',2);

gaussdb=# SELECT a,listagg(distinct b,',') FROM listagg_t1 GROUP BY a ORDER BY a;
 a |       listagg
---+---------------------
 1 | 3 days,4 days,5 days
 2 | 1 day
```


版权所有 © 华为云计算技术有限公司

```
 3 | 2 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE listagg_t1;
```

listagg作为窗口函数时，listagg列为对应分组的有序聚集。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval);

gaussdb=# INSERT INTO listagg_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'3 days'),(2,'4 days'),(2,'5
days'),(3,'6 days');

gaussdb=# SELECT a,LISTAGG(b) WITHIN GROUP(ORDER BY b) OVER(PARTITION BY a) FROM
listagg_t1;
 a |   listagg
---+--------------
 1 | 2 days3 days
 1 | 2 days3 days
 2 | 4 days5 days
 2 | 4 days5 days
 3 | 6 days
   | 1 day
(6 rows)

gaussdb=# DROP TABLE listagg_t1;
```

listagg作为窗口函数时，使用distinct去重，仅支持ORA兼容模式。

```
gaussdb=# CREATE TABLE listagg_t1(a int, b interval, c int);

gaussdb=# INSERT INTO listagg_t1 VALUES(1,'3 days',8),(1,'4 days',7),(2,'1 days',6),(1,'5 days',5),(3,'2
days',4),(1,'4 days',3),(NULL,'1 days',2),(2,'1 days',1);

gaussdb=# SELECT a,listagg(distinct b,',') over(partition by a) FROM listagg_t1 ORDER BY a;
 a |       listagg
---+----------------------
 1 | 3 days,4 days,5 days
 1 | 3 days,4 days,5 days
 1 | 3 days,4 days,5 days
 1 | 3 days,4 days,5 days
 2 | 1 day
 2 | 1 day
 3 | 2 days
   | 1 day
(8 rows)

gaussdb=# SELECT a,listagg(distinct b,',') WITHIN GROUP(ORDER BY b) over(partition by a) FROM
listagg_t1 ORDER BY a;
 a |       listagg
---+----------------------
 1 | 3 days,4 days,5 days
 1 | 3 days,4 days,5 days
 1 | 3 days,4 days,5 days
 1 | 3 days,4 days,5 days
 2 | 1 day
 2 | 1 day
 3 | 2 days
   | 1 day
(8 rows)

gaussdb=# SELECT a,listagg(distinct b,',') WITHIN GROUP(ORDER BY c) over(partition by a) FROM
listagg_t1 ORDER BY a;
 a |       listagg
---+----------------------
 1 | 4 days,5 days,3 days
 1 | 4 days,5 days,3 days
 1 | 4 days,5 days,3 days
 1 | 4 days,5 days,3 days
 2 | 1 day
 2 | 1 day
 3 | 2 days
```


版权所有 © 华为云计算技术有限公司

```
  | 1 day
(8 rows)

gaussdb=# DROP TABLE listagg_t1;
```

- group_concat([DISTINCT | ALL] expression [,expression ...] [ORDER BY { expression [[ASC | DESC | USING operator] | nlssort_expression_clause] [NULLS { FIRST | LAST }] } [,...]] [SEPARATOR str_val])

  描述：参数数量不定，可对多列进行拼接，将聚集列数据按照ORDER BY指定的排序方式排列，并用separator指定的分隔符拼接成一个字符串, 不支持作为窗口函数使用。

  - DISTINCT：可选，表示对每行拼接后结果进行去重。
  - expression：必选，指定聚集列名或基于列的有效表达式。
  - ORDER BY: 可选，后跟可变数量表达式及排序规则。group_concat函数中不支持（ORDER BY + 数字）形式。
  - SEPARATOR子句: 可选，后跟字符或字符串，分组中相邻两行表达式结果使用此分隔符拼接。若不指定，默认使用英文逗号‘,’。
  - 当同时指定DISTINCT和ORDER BY时，ORDER BY表达式必须在distinct表达式中，否则报错。
  - 使用参数group_concat_max_len限制GROUP_CONCAT最大返回长度，超长截断，目前能返回的最大长度是1073741823。

  返回类型：text

  示例：

  使用separator指定分隔符为';'。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b int);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,1),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT a,group_concat(b separator ';') FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+--------------
 1 | 2;3
 2 | 4;5
 3 | 6
   | 1
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

  分隔符缺省时，默认为','。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b int);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,1),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT a,group_concat(a,b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+--------------
 1 | 12,13
 2 | 24,25
 3 | 36
   |
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

  聚集列是文本字符集类型。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b text);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,'a1'),(1,'b2'),(1,'c3'),(2,'d4'),(2,'e5'),(3,'f6');
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT a,group_concat(a,b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+--------------
 1 | 1b2,1c3
 2 | 2d4,2e5
 3 | 3f6
   |
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

聚集列是整型。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b int);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,group_concat(b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+--------------
 1 | 2,3
 2 | 4,5
 3 | 6
   | 1
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

聚集列是浮点类型。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b float);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,1.11),(1,2.22),(1,3.33),(2,4.44),(2,5.55),
(3,6.66);

gaussdb=# SELECT a,group_concat(b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+--------------
 1 | 3,2,2,3
 2 | 6,4,5,4
 3 | 7,6
   | 1,1
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

聚集列是时间类型。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b timestamp);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,'2000-01-01'),(1,'2000-02-02'),
(1,'2000-03-03'),(2,'2000-04-04'),(2,'2000-05-05'),(3,'2000-06-06');

gaussdb=# SELECT a,group_concat(b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a |              group_concat
---+-----------------------------------------
 1 | 2000-02-02 00:00:00,2000-03-03 00:00:00
 2 | 2000-04-04 00:00:00,2000-05-05 00:00:00
 3 | 2000-06-06 00:00:00
   | 2000-01-01 00:00:00
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

聚集列是二进制类型。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b bytea);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,'1'),(1,'2'),(1,'3'),(2,'4'),(2,'5'),(3,'6');

gaussdb=# SELECT a,group_concat(b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+--------------
```


版权所有 © 华为云计算技术有限公司

```
 1 | \x32,\x33
 2 | \x34,\x35
 3 | \x36
   | \x31
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

聚集列是时间间隔类型。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b interval);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'3 days'),(2,'4 days'),
(2,'5 days'),(3,'6 days');

gaussdb=# SELECT a,group_concat(b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+---------------
 1 | 2 days,3 days
 2 | 4 days,5 days
 3 | 6 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

使用distinct去重。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b interval);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'2 days'),(1,'3 days'),
(1,'3 days'),(2,'4 days'),(2,'5 days'),(3,'6 days');

gaussdb=# SELECT a,group_concat(distinct b) FROM group_concat_t1 GROUP BY a ORDER BY a;
 a | group_concat
---+---------------
 1 | 2 days,3 days
 2 | 4 days,5 days
 3 | 6 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

使用ORDER BY排序。

```
gaussdb=# CREATE TABLE group_concat_t1(a int, b interval);

gaussdb=# INSERT INTO group_concat_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'3 days'),(2,'4 days'),
(2,'5 days'),(3,'6 days');

gaussdb=# SELECT a,group_concat(b ORDER BY b desc) FROM group_concat_t1 GROUP BY a ORDER
BY a;
 a | group_concat
---+---------------
 1 | 3 days,2 days
 2 | 5 days,4 days
 3 | 6 days
   | 1 day
(4 rows)

gaussdb=# DROP TABLE group_concat_t1;
```

- wm_concat(expression)

  描述：将指定的列/列的表达式数据用逗号（,）连接成一个字符串进行输出。

  返回类型：text

  示例：

```
gaussdb=# CREATE TABLE wm_concat_t1(a int, b interval);
CREATE TABLE
gaussdb=# INSERT INTO wm_concat_t1 VALUES (NULL,'1 days'),(1,'2 days'),(1,'3 days'),(2,'4 days'),
```


版权所有 © 华为云计算技术有限公司

```
(2,'5 days'),(3,'6 days');
INSERT 0 6
gaussdb=# SELECT wm_concat(a) FROM wm_concat_t1;
 wm_concat
-----------
 1,1,2,2,3
(1 row)
gaussdb=# SELECT wm_concat(b) FROM wm_concat_t1;
                wm_concat
------------------------------------------
 1 day,2 days,3 days,4 days,5 days,6 days
(1 row)
gaussdb=# DROP TABLE wm_concat_t1;
```

说明

该函数的返回值为text，拼接结果无序。

- covar_pop(Y, X)

  描述：总体协方差。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE covar_pop_t1(a int, b int);

gaussdb=# INSERT INTO covar_pop_t1 VALUES (NULL,11),(11,21),(11,31),(21,41),(21,51),(31,61);

gaussdb=# SELECT COVAR_POP(a,b) FROM covar_pop_t1;
 covar_pop
-----------
       100
(1 row)

gaussdb=# DROP TABLE covar_pop_t1;
```

- covar_samp(Y, X)

  描述：样本协方差。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE covar_samp_t1(a int, b int);

gaussdb=# INSERT INTO covar_samp_t1 VALUES (NULL,11),(11,21),(11,31),(21,41),(21,51),(31,61);

gaussdb=# SELECT COVAR_SAMP(a,b) FROM covar_samp_t1;
 covar_samp
------------
        125
(1 row)

gaussdb=# DROP TABLE covar_samp_t1;
```

- stddev_pop(expression)

  描述：总体标准差。

  返回类型：对于浮点类型的输入返回double precision，其他输入返回numeric。

  示例：

```
gaussdb=# CREATE TABLE stddev_pop_t1(a int, b int);

gaussdb=# INSERT INTO stddev_pop_t1 VALUES (NULL,11),(11,21),(11,31),(21,41),(21,51),(31,61);

gaussdb=# SELECT STDDEV_POP(a) FROM stddev_pop_t1;
    stddev_pop
--------------------
 7.4833147735478828
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# DROP TABLE stddev_pop_t1;
```

- stddev_samp(expression)

  描述：样本标准差。

  返回类型：对于浮点类型的输入返回double precision，其他输入返回numeric。

  示例：

```
gaussdb=# CREATE TABLE stddev_samp_t1(a int, b int);

gaussdb=# INSERT INTO stddev_samp_t1 VALUES (NULL,11),(11,21),(11,31),(21,41),(21,51),(31,61);

gaussdb=# SELECT STDDEV_SAMP(a) FROM stddev_samp_t1;
   stddev_samp
--------------------
 8.3666002653407555
(1 row)

gaussdb=# DROP TABLE stddev_samp_t1;
```

- var_pop(expression)

  描述：总体方差（总体标准差的平方）。

  返回类型：对于浮点类型的输入返回double precision类型，其他输入返回numeric类型。

  示例：

```
gaussdb=# CREATE TABLE var_pop_t1(a int, b int);

gaussdb=# INSERT INTO var_pop_t1 VALUES (NULL,11),(11,21),(11,31),(21,41),(21,51),(31,61);

gaussdb=# SELECT VAR_POP(a) FROM var_pop_t1;
       var_pop
---------------------
 56.0000000000000000
(1 row)

gaussdb=# DROP TABLE var_pop_t1;
```

- var_samp(expression)

  描述：样本方差（样本标准差的平方）。

  返回类型：对于浮点类型的输入返回double precision类型，其他输入返回numeric类型。

  示例：

```
gaussdb=# CREATE TABLE var_samp_t1(a int, b int);

gaussdb=# INSERT INTO var_samp_t1 VALUES (NULL,11),(11,21),(11,31),(21,41),(21,51),(31,61);

gaussdb=# SELECT VAR_SAMP(a) FROM var_samp_t1;
      var_samp
---------------------
 70.0000000000000000
(1 row)

gaussdb=# DROP TABLE var_samp_t1;
```

- bit_and(expression)

  描述：所有非NULL输入值的按位与(AND)，如果全部输入值皆为NULL，那么结果也为NULL 。

  返回类型：和参数数据类型相同。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE TABLE bit_and_t1(a int, b int);

gaussdb=# INSERT INTO bit_and_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT BIT_AND(a) FROM bit_and_t1;
 bit_and
---------
       0
(1 row)

gaussdb=# DROP TABLE bit_and_t1;
```

- bit_or(expression)

  描述：所有非NULL输入值的按位或(OR)，如果全部输入值皆为NULL，那么结果也为NULL。

  返回类型：和参数数据类型相同。

  示例：

```
gaussdb=# CREATE TABLE bit_or_t1(a int, b int);

gaussdb=# INSERT INTO bit_or_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT BIT_OR(a) FROM bit_or_t1;
 bit_or
--------
      3
(1 row)

gaussdb=# DROP TABLE bit_or_t1;
```

- bool_and(expression)

  描述：如果所有输入值都是真，则为真，否则为假。

  返回类型：bool

  示例：

```
gaussdb=# SELECT bool_and(100 <2500);
 bool_and
----------
 t
(1 row)
```

- bool_or(expression)

  描述：如果所有输入值只要有一个为真，则为真，否则为假。

  返回类型：bool

  示例：

```
gaussdb=# SELECT bool_or(100 <2500);
 bool_or
----------
 t
(1 row)
```

- corr(Y, X)

  描述：相关系数。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE corr_t1(a int, b int);

gaussdb=# INSERT INTO corr_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT CORR(a,b) FROM corr_t1;
       corr
------------------
```


版权所有 © 华为云计算技术有限公司

```
 .944911182523068
(1 row)

gaussdb=# DROP TABLE corr_t1;
```

- every(expression)

  描述：等效于bool_and。

  返回类型：bool

  示例：

```
gaussdb=# SELECT every(100 <2500);
 every
-------
 t
(1 row)
```

- rank(expression)

  描述：根据expression对不同组内的元组进行跳跃排序。

  返回类型：BIGINT

  示例：

```
gaussdb=# CREATE TABLE rank_t1(a int, b int);

gaussdb=# INSERT INTO rank_t1 VALUES(NULL,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,RANK() OVER(PARTITION BY a ORDER BY b) FROM rank_t1;
 a | b | rank
---+---+------
 1 | 2 |    1
 1 | 3 |    2
 2 | 4 |    1
 2 | 5 |    2
 3 | 6 |    1
   | 1 |    1
(6 rows)

gaussdb=# DROP TABLE rank_t1;
```

- regr_avgx(Y, X)

  描述：自变量的平均值 (sum(X)/Y)。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_t1(a int, b int);

gaussdb=# INSERT INTO regr_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_AVGX(a,b) FROM regr_t1;
 regr_avgx
-----------
         4
(1 row)

gaussdb=# DROP TABLE regr_t1;
```

- regr_avgy(Y, X)

  描述：因变量的平均值 (sum(Y)/X)。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_avgy_t1(a int, b int);

gaussdb=# INSERT INTO regr_avgy_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT REGR_AVGY(a,b) FROM regr_avgy_t1;
 regr_avgy
-----------
      1.8
(1 row)

gaussdb=# DROP TABLE regr_avgy_t1;
```

- regr_count(Y, X)

  描述：两个表达式都不为NULL的输入行数。

  返回类型：bigint

  示例：

```
gaussdb=# CREATE TABLE regr_count_t1(a int, b int);

gaussdb=# INSERT INTO regr_count_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_COUNT(a,b) FROM regr_count_t1;
 regr_count
------------
         5
(1 row)

gaussdb=# DROP TABLE regr_count_t1;
```

- regr_intercept(Y, X)

  描述：根据所有输入的点(X, Y)按照最小二乘法拟合成一个线性方程，然后返回该直线的Y轴截距。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_intercept_t1(a int, b int);

gaussdb=# INSERT INTO regr_intercept_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_INTERCEPT(b,a) FROM regr_intercept_t1;
  regr_intercept
------------------
 .785714285714286
(1 row)

gaussdb=# DROP TABLE regr_intercept_t1;
```

- regr_r2(Y, X)

  描述：相关系数的平方。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_r2_t1(a int, b int);

gaussdb=# INSERT INTO regr_r2_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_R2(b,a) FROM regr_r2_t1;
     regr_r2
------------------
 .892857142857143
(1 row)

gaussdb=# DROP TABLE regr_r2_t1;
```

- regr_slope(Y, X)

  描述：根据所有输入的点(X, Y)按照最小二乘法拟合成一个线性方程， 然后返回该直线的斜率。

  返回类型：double precision

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CREATE TABLE regr_slope_t1(a int, b int);

gaussdb=# INSERT INTO regr_slope_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_SLOPE(b,a) FROM regr_slope_t1;
    regr_slope
------------------
 1.78571428571429
(1 row)

gaussdb=# DROP TABLE regr_slope_t1;
```

- regr_sxx(Y, X)

  描述：sum(Y^2) - sum(X)^2/N （自变量的“平方和”）。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_sxx_t1(a int, b int);

gaussdb=# INSERT INTO regr_sxx_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_SXX(b,a) FROM regr_sxx_t1;
 regr_sxx
----------
      2.8
(1 row)

gaussdb=# DROP TABLE regr_sxx_t1;
```

- regr_sxy(Y, X)

  描述：sum(X*Y) - sum(X) * sum(Y)/N （自变量和因变量的“乘方积”）。

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_sxy_t1(a int, b int);

gaussdb=# INSERT INTO regr_sxy_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_SXY(b,a) FROM regr_sxy_t1;
 regr_sxy
----------
        5
(1 row)

gaussdb=# DROP TABLE regr_sxy_t1;
```

- regr_syy(Y, X)

  描述：sum(Y^2) - sum(X)^2/N（因变量的"平方和"）

  返回类型：double precision

  示例：

```
gaussdb=# CREATE TABLE regr_syy_t1(a int, b int);

gaussdb=# INSERT INTO regr_syy_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT REGR_SYY(b,a) FROM regr_syy_t1;
 regr_syy
----------
       10
(1 row)

gaussdb=# DROP TABLE regr_syy_t1;
```

- stddev(expression)

 版权所有 © 华为云计算技术有限公司

描述：stddev_samp的别名。

返回类型：对于浮点类型的输入返回double precision，其他输入返回numeric。

示例：

```
gaussdb=# CREATE TABLE stddev_t1(a int, b int);

gaussdb=# INSERT INTO stddev_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT STDDEV(a) FROM stddev_t1;
       stddev
----------------------
 .83666002653407554798
(1 row)

gaussdb=# DROP TABLE stddev_t1;
```

- variance(expression,expression)

  描述：var_samp的别名。

  返回类型：对于浮点类型的输入返回double precision类型，其他输入返回numeric类型。

  示例：

```
gaussdb=# CREATE TABLE variance_t1(a int, b int);

gaussdb=# INSERT INTO variance_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT VARIANCE(a) FROM variance_t1;
      variance
----------------------
 .70000000000000000000
(1 row)

gaussdb=# DROP TABLE variance_t1;
```

- spread()

  描述：该函数用于计算某段时间内最大和最小值差值。

  参数：real，如**表1-93**所示。

  返回值类型：real

  表 1-93 spread 函数参数说明

参数名	类型	描述
field	数值型	需要计算的列。

  说明

  - 每个分组内如果少于2个元组，返回结果为0，不要和over语句混用。
  - spread函数无法进行下推，需要在CN节点拉取数据进行计算，如果数据量很大，会造成数据下盘，影响性能。

  示例：

```
gaussdb=# SELECT
gaussdb-#       SPREAD(value)
gaussdb-# FROM
gaussdb-#       (VALUES ('2019-07-12 00:00:00'::timestamptz, 1),('2019-07-12 00:01:00'::timestamptz,
2),('2019-07-12 00:02:00'::timestamptz, 3)) v(time,value);
 spread
--------
```


版权所有 © 华为云计算技术有限公司

```
    2
(1 row)
```

- pivot_func(anyelement)

  描述：返回某列中唯一不为NULL的值，如果有超过两个非NULL值则报错。其中value为输入参数，可以为任意类型。

  返回类型：与输入参数类型相同。

  说明

  该聚合函数主要作为pivot语法内部使用。

```
gaussdb=# CREATE TABLE pivot_func_t1(a int, b int);

gaussdb=# INSERT INTO pivot_func_t1 VALUES (NULL,11),(1,2);

gaussdb=# SELECT PIVOT_FUNC(a) FROM pivot_func_t1;
 pivot_func
------------
          1
(1 row)

gaussdb=# DROP TABLE pivot_func_t1;
```

- checksum(expression)

  描述：返回所有输入值的CHECKSUM值。使用该函数可以用来验证GaussDB数据库（不支持GaussDB之外的其他数据库）的备份恢复或者数据迁移操作前后表中的数据是否相同。在备份恢复或者数据迁移操作前后都需要用户通过手工执行SQL命令的方式获取执行结果，通过对比获取的执行结果判断操作前后表中的数据是否相同。

  说明

  - 对于大表，CHECKSUM函数可能会需要很长时间。
  - 如果某两表的CHECKSUM值不同，则表明两表的内容是不同的。由于CHECKSUM函数中使用散列函数不能保证无冲突，因此两个不同内容的表可能会得到相同的CHECKSUM值，存在这种情况的可能性较小。对于列进行的CHECKSUM也存在相同的情况。
  - 对于时间类型timestamp, timestamptz和smalldatetime，计算CHECKSUM值时请确保时区设置一致。

  - 若计算某列的CHECKSUM值，且该列类型可以默认转为TEXT类型，则expression为列名。
  - 若计算某列的CHECKSUM值，且该列类型不能默认转为TEXT类型，则expression为列名::TEXT。
  - 若计算所有列的CHECKSUM值，则expression为表名::TEXT。

  可以默认转换为TEXT类型的类型包括：char, name, int8, int2, int1, int4, raw, pg_node_tree, float4, float8, bpchar, varchar, nvarchar2, date, datea（仅在ORA兼容性生效）, timestamp, timestamptz, numeric, smalldatetime，其他类型需要强制转换为TEXT，例如XML类型。

  返回类型：numeric。

  示例：

  表中可以默认转为TEXT类型的某列的CHECKSUM值。

```
gaussdb=# CREATE TABLE checksum_t1(a int, b int);

gaussdb=# INSERT INTO checksum_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT CHECKSUM(a) FROM checksum_t1;
  checksum
```


版权所有 © 华为云计算技术有限公司

```
-------------
 18126842830
(1 row)

gaussdb=# DROP TABLE checksum_t1;
```

表中不能默认转为TEXT类型的某列的CHECKSUM值。注意此时CHECKSUM参数是列名::TEXT。

```
gaussdb=# CREATE TABLE checksum_t1(a int, b int);

gaussdb=# INSERT INTO checksum_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT CHECKSUM(a::TEXT) FROM checksum_t1;
 checksum
-------------
 18126842830
(1 row)

gaussdb=# DROP TABLE checksum_t1;
```

表中所有列的CHECKSUM值。注意此时CHECKSUM参数是表名::TEXT，且表名前不加Schema。

```
gaussdb=# CREATE TABLE checksum_t1(a int, b int);

gaussdb=# INSERT INTO checksum_t1 VALUES (NULL,11),(1,2),(1,3),(2,4),(2,5),(3,6);

gaussdb=# SELECT CHECKSUM(checksum_t1::TEXT) FROM checksum_t1;
 checksum
-------------
 11160522226
(1 row)

gaussdb=# DROP TABLE checksum_t1;
```

- sample(value_array anyarray, N int)

  描述：对给定的列，返回其中随机N个值。

  参数：参照如**表1-94**所示。

  返回值类型：anyelement

  表 1-94 sample 函数参数说明

参数名	类型	描述
value_array	anyarray	查询列的数组。
N	int	随机采样个数。

  说明

  - N需要为正数，且不大于value_array长度。
  - 不能和over子句一起使用。
  - 不支持对数据量超过int8类型取值范围的数组使用。

  示例：

```
gaussdb=# SELECT
gaussdb-#     sample(array_agg(value), 1)
gaussdb-# FROM
gaussdb-#     (VALUES ('2019-07-12 00:00:00'::timestamptz, 1),('2019-07-12 00:01:00'::timestamptz,
2),('2019-07-12 00:02:00'::timestamptz, 3)) v(time,value);
 sample
```


版权所有 © 华为云计算技术有限公司

```
--------
     3
(1 row)
```

- mode() within group (order by value anyelement)

  描述：对给定的列，返回出现频率最高的值，如果多个值频率相同，返回这些值中最小的值。

  参数：参照如表1-95所示。

  返回值类型：anyelement

  表 1-95 mode 函数参数说明

参数名	类型	描述
value	anyelement	查询列。

  说明

  - 需要与within group一起使用，无within group语句，会报错，该函数参数放在group的order by后面。
  - 不能和over子句一起使用。

  示例：

```
gaussdb=# SELECT
gaussdb-#     mode() within group (order by value)
gaussdb-# FROM
gaussdb-#     (VALUES ('2019-07-12 00:00:00'::timestamptz, 1),('2019-07-12 00:01:00'::timestamptz, 2),('2019-07-12 00:02:00'::timestamptz, 3)) v(time,value);
 mode
------
    1
(1 row)
```

- integral(time_array timestamptz[], value_array double precision[], unit interval)

  描述：对给定的时间和某个列，返回该列在时间下的梯形面积积分。

  参数：参照如表1-96所示。

  返回值类型：float

  表 1-96 integral 函数参数说明

参数名	类型	描述
time_array	timestamptz[]	查询时间数组。
value_array	double precision[]	查询列的数组。
unit	interval	时间单位。

版权所有 © 华为云计算技术有限公司

说明

- 时间数组长度和value数组长度需要相同，否则报错。
- 不能和over子句一起使用。
- 不支持对数据量超过int类型取值范围的数组使用。

示例：

```
gaussdb=# SELECT
gaussdb-#     integral(array_agg(time), array_agg(value), interval '1 second')
gaussdb-# FROM
gaussdb-#     (VALUES ('2019-07-12 00:00:00'::timestamptz, 1),('2019-07-12 00:01:00'::timestamptz,
2),('2019-07-12 00:02:00'::timestamptz, 3)) v(time,value);
    integral
------------------
 239.999999999999
(1 row)
```

聚集函数嵌套

描述：对聚集函数分组计算的结果再进行一次聚集函数操作。

一般可描述为：

```
SELECT AGG1(AGG2(column_name1)) FROM table_name GROUP BY column_name2;
```

等价为：

```
SELECT AGG1(value) FROM (SELECT AGG2(column_name1) value FROM table_name GROUP BY
column_name2);
```

其中：

- AGG1()：表示外层聚集函数。
- AGG2()：表示内层聚集函数。
- table_name：表示表名。
- column_name1、column_name2：表示列名。
- value：表示内层聚集函数结果的别名。

整体含义可描述为：将内层聚集函数AGG2()分组计算的结果作为外层聚集函数AGG1()的输入再计算一次。

说明

1. 嵌套的聚集函数应位于SELECT和FROM之间，否则无意义。
2. 使用嵌套聚集函数的SELECT语句应包含GROUP BY子句。
3. 与嵌套聚集函数同时被SELECT的，应同为嵌套聚集函数，或为常量表达式。其中常量表达式不支持CASE WHEN表达式或逻辑上等价于CASE WHEN的函数（如：decode函数、ORA兼容模式下的nvl2函数、MYSQL兼容模式下的if函数）。
4. 聚集函数仅支持一次聚集函数嵌套操作。
5. 当前支持以下聚集函数之间的嵌套：avg、max、min、sum、var_pop、var_samp、variance、stddev_pop、stddev_samp、stddev、median、regr_sxx、regr_syy、regr_sxy、regr_avgx、regr_avgy、regr_r2、regr_slope、regr_intercept、covar_pop、covar_samp、corr和listagg。
6. 内层聚集函数的返回结果类型应符合外层聚集函数的参数类型。

示例：

```
gaussdb=# CREATE TABLE test1 (id INT,val INT);
CREATE TABLE
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO test1 VALUES (1, 1);
INSERT 0 1
gaussdb=# INSERT INTO test1 VALUES (1, null);
INSERT 0 1
gaussdb=# INSERT INTO test1 VALUES (2, 10);
INSERT 0 1
gaussdb=# INSERT INTO test1 VALUES (2, 55);
INSERT 0 1

gaussdb=# SELECT SUM(MIN(val)) FROM test1 GROUP BY id;
 sum
-----
  11
(1 row)

gaussdb=# DROP TABLE test1;
DROP TABLE
```

1.6.19 窗口函数

窗口函数

窗口函数与OVER语句一起使用。OVER语句用于对数据进行分组，并对组内元素进行排序。窗口函数用于给组内的值生成序号。

说明

窗口函数中的order by后面必须跟字段名，若order by后面跟数字，该数字会被按照常量处理，因此对目标列没有起到排序的作用。

当父查询中包含子查询中的窗口函数的过滤条件时，支持将父查询中的过滤条件下推到子查询中。

注意

1. 只支持将父查询中关于窗口函数的<、<=、=过滤条件下推到子查询中。
2. 限制条件范围的上边界可以是常量、常量表达式、参数、非VOLATILE类型的函数、非相关子链接。
3. 只支持**ROW_NUMBER()**、**RANK()**、**DENSE_RANK()**三个窗口函数。

示例：

```
--创建表。
gaussdb=# CREATE TABLE t2 (c1 INT, c2 INT);
--导入数据。
gaussdb=# INSERT INTO t2 SELECT generate_series, generate_series FROM generate_series(1, 1000000);
--执行查询，查询正常无报错，且执行结果正确。
gaussdb=# EXPLAIN ANALYZE SELECT nc1 FROM (
SELECT row_number() over() rid,
t2.c1 nc1
FROM t2
) WHERE rid BETWEEN 1 AND (1 + 10 - 1);
                                                  QUERY PLAN
---------------------------------------------------------------------------------------------------------------------
Streaming (type: GATHER)  (cost=0.12..10.64 rows=5 width=4) (actual time=10.201..12.766 rows=10
loops=1)
   Node/s: All datanodes
   ->  Subquery Scan on __unnamed_subquery__  (cost=0.00..10.45 rows=5 width=4) (actual
time=[5.310,6.092]..[5.310,6.092], rows=10)
         Filter: (__unnamed_subquery__.rid >= 1)
         ->  WindowAgg  (cost=0.00..10.25 rows=16 width=4) (actual time=[5.262,5.873]..[5.262,5.873],
```


版权所有 © 华为云计算技术有限公司

```
rows=10)
            row_number_filter: (row_number() OVER () <= 10)
            ->  Streaming(type: BROADCAST)  (cost=0.00..10.29 rows=32 width=4) (actual
time=[5.162,5.682]..[5.162,5.682], rows=11)
                  Spawn on: All datanodes
                  ->  WindowAgg  (cost=0.00..10.25 rows=16 width=4) (actual time=[1.759,1.815]..
[1.786,1.844], rows=20)
                        row_number_filter: (row_number() OVER () <= 10)
                        ->  Seq Scan on t2  (cost=0.00..13.13 rows=20 width=4) (actual time=[1.711,1.716]..
[1.743,1.747], rows=22)
 Total runtime: 13.428 ms
(12 rows)
--环境清理，避免数据泄露。
gaussdb=# DROP TABLE t2;
```

- RANK()

  描述：RANK函数为各组内值生成跳跃排序序号，其中，相同的值具有相同序号。

  返回值类型：BIGINT

  示例：

  ```
  gaussdb=# CREATE TABLE rank_t1(a int, b int);

  gaussdb=# INSERT INTO rank_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

  gaussdb=# SELECT a,b,RANK() OVER(PARTITION BY a ORDER BY b) FROM rank_t1;
   a | b | rank
  ---+---+------
   1 | 1 |    1
   1 | 1 |    1
   1 | 2 |    3
   1 | 3 |    4
   2 | 4 |    1
   2 | 5 |    2
   3 | 6 |    1
  (7 rows)

  gaussdb=# DROP TABLE rank_t1;
  ```

- ROW_NUMBER()

  描述：ROW_NUMBER函数为各组内值生成连续排序序号，其中，相同的值其序号也不相同。

  返回值类型：BIGINT

  示例：

  ```
  gaussdb=# CREATE TABLE row_number_t1(a int, b int);

  gaussdb=# INSERT INTO row_number_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

  gaussdb=# SELECT a,b,ROW_NUMBER() OVER(PARTITION BY a ORDER BY b) FROM row_number_t1;
   a | b | row_number
  ---+---+------------
   1 | 1 |          1
   1 | 1 |          2
   1 | 2 |          3
   1 | 3 |          4
   2 | 4 |          1
   2 | 5 |          2
   3 | 6 |          1
  (7 rows)

  gaussdb=# DROP TABLE row_number_t1;
  ```

- DENSE_RANK()

  描述：DENSE_RANK函数为各组内值生成连续排序序号，其中，相同的值具有相同序号。

  返回值类型：BIGINT

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CREATE TABLE dense_rank_t1(a int, b int);

gaussdb=# INSERT INTO dense_rank_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,DENSE_RANK() OVER(PARTITION BY a ORDER BY b) FROM dense_rank_t1;
 a | b | dense_rank
---+---+------------
 1 | 1 |          1
 1 | 1 |          1
 1 | 2 |          2
 1 | 3 |          3
 2 | 4 |          1
 2 | 5 |          2
 3 | 6 |          1
(7 rows)

gaussdb=# DROP TABLE dense_rank_t1;
```

- PERCENT_RANK()

  描述：PERCENT_RANK函数为各组内对应值生成相对序号，即根据公式 (rank - 1) / (totalrows - 1)计算所得的值。其中rank为该值依据RANK函数所生成的对应序号，totalrows为该分组内的总元素个数。

  返回值类型：DOUBLE PRECISION

  示例：

```
gaussdb=# CREATE TABLE percent_rank_t1(a int, b int);

gaussdb=# INSERT INTO percent_rank_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,PERCENT_RANK() OVER(PARTITION BY a ORDER BY b) FROM percent_rank_t1;
 a | b |    percent_rank
---+---+-----------------
 1 | 1 |               0
 1 | 1 |               0
 1 | 2 | .666666666666667
 1 | 3 |               1
 2 | 4 |               0
 2 | 5 |               1
 3 | 6 |               0
(7 rows)

gaussdb=# DROP TABLE percent_rank_t1;
```

- CUME_DIST()

  描述：CUME_DIST函数为各组内对应值生成累积分布序号。即根据公式(小于等于当前值的数据行数)/(该分组总行数totalrows)计算所得的相对序号。

  返回值类型：DOUBLE PRECISION

  示例：

```
gaussdb=# CREATE TABLE cume_dist_t1(a int, b int);

gaussdb=# INSERT INTO cume_dist_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,CUME_DIST() OVER(PARTITION BY a ORDER BY b) FROM cume_dist_t1;
 a | b | cume_dist
---+---+-----------
 1 | 1 |        .5
 1 | 1 |        .5
 1 | 2 |       .75
 1 | 3 |         1
 2 | 4 |        .5
 2 | 5 |         1
 3 | 6 |         1
(7 rows)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE cume_dist_t1;
```

- NTILE(num_buckets integer)

  描述：NTILE函数根据num_buckets integer将有序的数据集合平均分配到num_buckets所指定数量的桶中，并将桶号分配给每一行。分配时应尽量做到平均分配。

  返回值类型：INTEGER

  示例：

```
gaussdb=# CREATE TABLE ntile_t1(a int, b int);

gaussdb=# INSERT INTO ntile_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,NTILE(2) OVER(PARTITION BY a ORDER BY b) FROM ntile_t1;
 a | b | ntile
---+---+-------
 1 | 1 |     1
 1 | 1 |     1
 1 | 2 |     2
 1 | 3 |     2
 2 | 4 |     1
 2 | 5 |     2
 3 | 6 |     1
(7 rows)

gaussdb=# DROP TABLE ntile_t1;
```

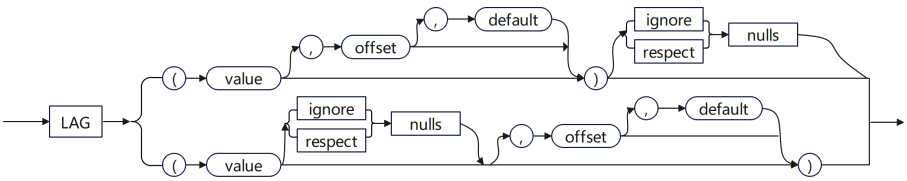
- LAG

  描述：LAG函数为各组内对应值生成滞后值。即将当前值对应的行数向前偏移offset位后所得行的value值作为序号。若经过向前偏移后行数不存在，则对应结果取为default值。若无指定，在默认情况下，offset取值为1，default值取值为NULL。default值的类型需要与value值的类型保持一致。

  语法：

```
LAG(value any [, offset integer [, default any ]])
LAG(value any ignore|respect nulls [, offset integer [, default any ]])
LAG(value any [, offset integer [, default any ]]) ignore|respect nulls
```

图 1-1 LAG 语法图



LAG函数的语法图如图1-1所示。ignore|respect nulls表示是否将NULL包含在向前偏移值中。若无指定，则默认是respect nulls。若设置为ignore nulls，当value为NULL时，则将其不包含在向前偏移值中。如果开启了ignore nulls功能，则该函数会存在性能劣化。

返回值类型：与参数数据类型相同。

示例1：不开启ignore nulls功能，设置offset=3, default=null

```
-- 建表并插入数据
gaussdb=# CREATE TABLE ta1 (hire_date date, last_name varchar(20), department_id int);
CREATE TABLE
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO ta1 values('07-DEC-02', 'Raphaely', 30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('24-JUL-05', 'Tobias',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('24-DEC-05', 'Baida',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('18-MAY-03', 'Khoo', 30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('15-NOV-06', 'Himuro',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-AUG-07', 'Colmenares',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-MAY-07', 'yq',  11);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-MAY-08', 'zi',  11);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('', 'yq1',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values(null, 'yq2',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-DEC-07', 'yq3',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values(null, null,  11);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values(null, null,  11);
INSERT 0 1

-- 调用LAG,指定offset=3, default=null
gaussdb=# SELECT hire_date, last_name, department_id, lag(hire_date, 3, null)  OVER (PARTITION BY
department_id ORDER BY last_name) AS "NextHired" FROM ta1 ORDER BY department_id;
     hire_date      | last_name  | department_id |      NextHired
--------------------+------------+---------------+---------------------
 2007-05-10 00:00:00 | yq         |            11 |
 2008-05-10 00:00:00 | zi         |            11 |
                     |            |            11 |
                     |            |            11 | 2007-05-10 00:00:00
 2005-12-24 00:00:00 | Baida      |            30 |
 2007-08-10 00:00:00 | Colmenares |            30 |
 2006-11-15 00:00:00 | Himuro     |            30 |
 2003-05-18 00:00:00 | Khoo       |            30 | 2005-12-24 00:00:00
 2002-12-07 00:00:00 | Raphaely   |            30 | 2007-08-10 00:00:00
 2005-07-24 00:00:00 | Tobias     |            30 | 2006-11-15 00:00:00
                     | yq1        |            30 | 2003-05-18 00:00:00
                     | yq2        |            30 | 2002-12-07 00:00:00
 2007-12-10 00:00:00 | yq3        |            30 | 2005-07-24 00:00:00
(13 rows)
```

示例2：开启ignore nulls功能，设置offset=3, default='01-JAN-00'

```
gaussdb=# SELECT hire_date, last_name, department_id, lag(hire_date, 3, '01-JAN-00') ignore nulls
OVER (PARTITION BY department_id ORDER BY last_name) AS "NextHired"
 FROM ta1 ORDER BY department_id;
     hire_date      | last_name  | department_id |      NextHired
--------------------+------------+---------------+---------------------
 2007-05-10 00:00:00 | yq         |            11 | 2000-01-01 00:00:00
 2008-05-10 00:00:00 | zi         |            11 | 2000-01-01 00:00:00
                     |            |            11 | 2000-01-01 00:00:00
                     |            |            11 | 2000-01-01 00:00:00
 2005-12-24 00:00:00 | Baida      |            30 | 2000-01-01 00:00:00
 2007-08-10 00:00:00 | Colmenares |            30 | 2000-01-01 00:00:00
 2006-11-15 00:00:00 | Himuro     |            30 | 2000-01-01 00:00:00
 2003-05-18 00:00:00 | Khoo       |            30 | 2005-12-24 00:00:00
 2002-12-07 00:00:00 | Raphaely   |            30 | 2007-08-10 00:00:00
 2005-07-24 00:00:00 | Tobias     |            30 | 2006-11-15 00:00:00
                     | yq1        |            30 | 2003-05-18 00:00:00
                     | yq2        |            30 | 2003-05-18 00:00:00
 2007-12-10 00:00:00 | yq3        |            30 | 2003-05-18 00:00:00
(13 rows)

-- 删除表
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE ta1;
DROP TABLE
```

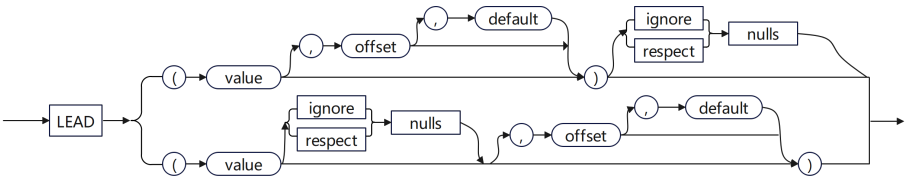
- LEAD

  描述：LEAD函数为各组内对应值生成提前值。即当前值对应的行数向后偏移offset位后所得行的value值作为序号。若经过向后偏移后行数超过当前组内的总行数，则对应结果取为default值。若无指定，在默认情况下，offset取为1，default值取为NULL。default值的类型需要与value值的类型保持一致。

  语法：

```
LEAD(value any [, offset integer [, default any ]])
LEAD(value any ignore|respect nulls [, offset integer [, default any ]])
LEAD(value any [, offset integer [, default any ]]) ignore|respect nulls
```

图 1-2 LEAD 语法图



LEAD函数的语法图如图1-2所示。ignore|respect nulls表示是否将NULL包含在向后偏移值中。若无指定，则默认是respect nulls。若设置为ignore nulls，当value为NULL时，则将其不包含在向后偏移值中。如果开启了ignore nulls功能，则该函数会存在性能劣化。

返回值类型：与参数数据类型相同。

示例1：不开启ignore nulls功能，设置offset=2，不指定default

```
-- 建表并插入数据
gaussdb=# CREATE TABLE ta1 (hire_date date, last_name varchar(20), department_id int);
CREATE TABLE
gaussdb=# INSERT INTO ta1 values('07-DEC-02', 'Raphaely', 30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('24-JUL-05', 'Tobias',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('24-DEC-05', 'Baida',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('18-MAY-03', 'Khoo', 30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('15-NOV-06', 'Himuro',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-AUG-07', 'Colmenares',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-MAY-07', 'yq',  11);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-MAY-08', 'zi',  11);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('', 'yq1',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values(null, 'yq2',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values('10-DEC-07', 'yq3',  30);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values(null, null,  11);
INSERT 0 1
gaussdb=# INSERT INTO ta1 values(null, null,  11);
INSERT 0 1
```


版权所有 © 华为云计算技术有限公司

```
-- 调用LEAD,指定offset=2
gaussdb=# SELECT hire_date, last_name, department_id, lead(hire_date, 2) OVER (PARTITION BY
department_id ORDER BY last_name) AS "NextHired" FROM ta1 ORDER BY department_id;
     hire_date      | last_name  | department_id |      NextHired
--------------------+------------+---------------+---------------------
 2007-05-10 00:00:00 | yq         |            11 |
 2008-05-10 00:00:00 | zi         |            11 |
                     |            |            11 |
                     |            |            11 |
 2005-12-24 00:00:00 | Baida      |            30 | 2006-11-15 00:00:00
 2007-08-10 00:00:00 | Colmenares |            30 | 2003-05-18 00:00:00
 2006-11-15 00:00:00 | Himuro     |            30 | 2002-12-07 00:00:00
 2003-05-18 00:00:00 | Khoo       |            30 | 2005-07-24 00:00:00
 2002-12-07 00:00:00 | Raphaely   |            30 |
 2005-07-24 00:00:00 | Tobias     |            30 |
                     | yq1        |            30 | 2007-12-10 00:00:00
                     | yq2        |            30 |
 2007-12-10 00:00:00 | yq3        |            30 |
(13 rows)
```

示例2：开启ignore nulls功能，设置offset=2，default='01-JAN-00'

```
gaussdb=# SELECT hire_date, last_name, department_id, lead(hire_date, 2, '01-JAN-00') ignore nulls
OVER (PARTITION BY department_id ORDER BY last_name) AS "NextHired" FROM ta1 ORDER BY
department_id;
     hire_date      | last_name  | department_id |      NextHired
--------------------+------------+---------------+---------------------
 2007-05-10 00:00:00 | yq         |            11 | 2000-01-01 00:00:00
 2008-05-10 00:00:00 | zi         |            11 | 2000-01-01 00:00:00
                     |            |            11 | 2000-01-01 00:00:00
                     |            |            11 | 2000-01-01 00:00:00
 2005-12-24 00:00:00 | Baida      |            30 | 2006-11-15 00:00:00
 2007-08-10 00:00:00 | Colmenares |            30 | 2003-05-18 00:00:00
 2006-11-15 00:00:00 | Himuro     |            30 | 2002-12-07 00:00:00
 2003-05-18 00:00:00 | Khoo       |            30 | 2005-07-24 00:00:00
 2002-12-07 00:00:00 | Raphaely   |            30 | 2007-12-10 00:00:00
 2005-07-24 00:00:00 | Tobias     |            30 | 2000-01-01 00:00:00
                     | yq1        |            30 | 2000-01-01 00:00:00
                     | yq2        |            30 | 2000-01-01 00:00:00
 2007-12-10 00:00:00 | yq3        |            30 | 2000-01-01 00:00:00
(13 rows)

-- 删除表
gaussdb=# DROP TABLE ta1;
DROP TABLE
```

- FIRST_VALUE()

  描述：FIRST_VALUE函数获取各组内的第一个值作为返回结果。

  返回值类型：与参数数据类型相同。

  语法：

  ```
  LAST_VALUE(value any)
  LAST_VALUE(value any IGNORE|RESPECT NULLS)
  LAST_VALUE(value any) IGNORE|RESPECT NULLS
  ```

  IGNORE|RESPECT NULLS：表示是否要忽略NULL值，其中RESPECT NULLS是缺省值。若开启IGNORE NULLS选项，当查询的组内第一个值是NULL时，会继续往后查询，直到查询到一个非NULL值时进行返回；如果一直向后查询到当前组的结尾仍为NULL，则返回NULL。

  示例1：不开启IGNORE NULLS，默认不忽略NULL值。

  ```
  gaussdb=# CREATE TABLE first_value_t1(a int, b int);

  gaussdb=# INSERT INTO first_value_t1 VALUES(1, NULL),(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

  gaussdb=# SELECT a,b,FIRST_VALUE(b) OVER(PARTITION BY a ORDER BY b) FROM first_value_t1;
   a | b | first_value
  ---+---+-------------
   1 | 1 |           1
  ```


版权所有 © 华为云计算技术有限公司

```
 1 | 1 |          1
 1 | 2 |          1
 1 | 3 |          1
 1 |   |          1
 2 | 4 |          4
 2 | 5 |          4
 3 | 6 |          6
(8 rows)
gaussdb=# DROP TABLE first_value_t1;
```

示例2：开启IGNORE NULLS。

```
gaussdb=# CREATE TABLE first_value_t1(a int, b int);

gaussdb=# INSERT INTO first_value_t1 VALUES(1, NULL),(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,FIRST_VALUE(b) IGNORE NULLS OVER(PARTITION BY a ORDER BY b DESC)
FROM first_value_t1;
 a | b | first_value
---+---+-------------
 1 |   |
 1 | 3 |           3
 1 | 2 |           3
 1 | 1 |           3
 1 | 1 |           3
 2 | 5 |           5
 2 | 4 |           5
 3 | 6 |           6
(8 rows)

-- 对比不加IGNORE NULLS
gaussdb=# SELECT a,b,FIRST_VALUE(b) OVER (PARTITION BY a ORDER BY b DESC) FROM
first_value_t1;
 a | b | first_value
---+---+-------------
 1 |   |
 1 | 3 |
 1 | 2 |
 1 | 1 |
 1 | 1 |
 2 | 5 |           5
 2 | 4 |           5
 3 | 6 |           6
(8 rows)

gaussdb=# DROP TABLE first_value_t1;
```

- LAST_VALUE()

  描述：LAST_VALUE函数获取各组内的最后一个值作为返回结果。

  返回值类型：与参数数据类型相同。

  语法：

  ```
  LAST_VALUE(value any)
  LAST_VALUE(value any IGNORE|RESPECT NULLS)
  LAST_VALUE(value any) IGNORE|RESPECT NULLS
  ```

  IGNORE|RESPECT NULLS：表示是否要忽略NULL值。其中RESPECT NULLS是缺省值。若开启IGNORE NULLS选项，当查询到的组内最后一个值是NULL时，会继续向前查询，直到查询到一个非NULL值时进行返回；如果一直向前查询到当前组的开头仍是NULL值，则返回NULL。

  示例1：不开启IGNORE NULLS，默认不忽略NULL值。

  ```
  gaussdb=# CREATE TABLE last_value_t1(a int, b int);

  gaussdb=# INSERT INTO last_value_t1 VALUES(1, NULL),(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

  gaussdb=# SELECT a,b,LAST_VALUE(b) OVER(PARTITION BY a ORDER BY b) FROM last_value_t1;
   a | b | last_value
  ---+---+------------
  ```


版权所有 © 华为云计算技术有限公司

```
 1 | 1 |         1
 1 | 1 |         1
 1 | 2 |         2
 1 | 3 |         3
 1 |   |
 2 | 4 |         4
 2 | 5 |         5
 3 | 6 |         6
(8 rows)
gaussdb=# DROP TABLE last_value_t1;
```

示例2：开启IGNORE NULLS。

```
gaussdb=# CREATE TABLE last_value_t1(a int, b int);

gaussdb=# INSERT INTO last_value_t1 VALUES(1, NULL),(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,LAST_VALUE(b) IGNORE NULLS OVER(PARTITION BY a ORDER BY b) FROM
last_value_t1;
 a | b | last_value
---+---+------------
 1 | 1 |          1
 1 | 1 |          1
 1 | 2 |          2
 1 | 3 |          3
 1 |   |          3
 2 | 4 |          4
 2 | 5 |          5
 3 | 6 |          6
(8 rows)

gaussdb=# DROP TABLE last_value_t1;
```

- NTH_VALUE(value any, nth integer)

  描述：NTH_VALUE函数返回该组内的第nth行作为结果。若该行不存在，则默认返回NULL。

  返回值类型：与参数数据类型相同。

  语法：

  ```
  NTH_VALUE(value any, nth integer)
  NTH_VALUE(value any, nth integer) IGNORE|RESPECT NULLS
  ```

  IGNORE|RESPECT NULLS：表示是否要忽略NULL值。其中RESPECT NULLS是缺省值。若开启IGNORE NULLS选项，会在每个组从前向后查找非NULL值，直到查询到第nth个非NULL值时进行返回；如果查找到当前组的结尾都没有nth个非NULL值，则返回NULL。

  示例1：不开启IGNORE NULLS，默认不忽略NULL值。

  ```
  gaussdb=# CREATE TABLE nth_value_t1(a int, b int);

  gaussdb=# INSERT INTO nth_value_t1 VALUES(1, NULL),(1, NULL),(1, NULL),(1,1),(1,1),(1, 2),(1, 3),
  (2, 4),(2, 5),(3,6);

  gaussdb=# SELECT a,b,NTH_VALUE(b, 2) OVER(PARTITION BY a ORDER BY b) FROM nth_value_t1;
   a | b | nth_value
  ---+---+-----------
   1 | 1 |         1
   1 | 1 |         1
   1 | 2 |         1
   1 | 3 |         1
   1 |   |         1
   1 |   |         1
   1 |   |         1
   2 | 4 |
   2 | 5 |         5
   3 | 6 |
  (10 rows)

  gaussdb=# DROP TABLE nth_value_t1;
  ```


版权所有 © 华为云计算技术有限公司

示例2：开启IGNORE NULLS。

```
gaussdb=# CREATE TABLE nth_value_t1(a int, b int);

gaussdb=# INSERT INTO nth_value_t1 VALUES(1, NULL),(1, NULL),(1, NULL),(1,1),(1,1),(1, 2),(1, 3),
(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,NTH_VALUE(b, 2) IGNORE NULLS OVER(PARTITION BY a ORDER BY b DESC)
FROM nth_value_t1;
 a | b | nth_value
---+---+-----------
 1 |   |
 1 |   |
 1 |   |
 1 | 3 |
 1 | 2 |         2
 1 | 1 |         2
 1 | 1 |         2
 2 | 5 |
 2 | 4 |         4
 3 | 6 |
(10 rows)

-- 对比不加IGNORE NULLS
gaussdb=# SELECT a,b,NTH_VALUE(b, 2) OVER(PARTITION BY a ORDER BY b DESC) FROM
nth_value_t1;
 a | b | nth_value
---+---+-----------
 1 |   |
 1 |   |
 1 |   |
 1 | 3 |
 1 | 2 |
 1 | 1 |
 1 | 1 |
 2 | 5 |
 2 | 4 |         4
 3 | 6 |
(10 rows)

gaussdb=# DROP TABLE nth_value_t1;
```

- delta()

  描述：返回当前行和前一行的差值。

  参数：numeric，如**表1-97**所示。

  返回值类型：numeric

  表 1-97 delta 函数参数说明

参数名	类型	描述
field	数值型	需要计算的列。

版权所有 © 华为云计算技术有限公司

说明

- 当delta()函数作为窗口函数时，需与over()语句共同使用，且over()语句中rows语句不会改变delta()函数结果。如：delta(value) over(order by time rows 1 preceding)和delta(value) over(order by time rows 3 preceding)返回的结果一致。
- delta()函数内只支持可以进行计算的数值类型，不支持字符串类型。
- delta()窗口函数over()中应该固定为rows 1 preceding，指定是与前一行的差值；如果是其他写法，如下:
  1. 如果over()中用户使用ORDER BY子句，计算的为排序后每行与第一行的数据。
  2. 如果over()中用户使用PARTITION BY子句，计算的为每个分组最后行与第一行的数据。
- delta()函数无法进行下推，需要在CN节点拉取数据进行计算，如果数据量很大，会造成数据下盘，影响性能。

示例：

```
gaussdb=# SELECT
gaussdb-#       delta(value) over (rows 1 preceding)
gaussdb-# FROM
gaussdb-#       (VALUES ('2019-07-12 00:00:00'::timestamptz, 1),('2019-07-12 00:01:00'::timestamptz,
2),('2019-07-12 00:02:00'::timestamptz, 3)) v(time,value);
 delta
-------
     0
     1
     1
(3 rows)
```

- RATIO_TO_REPORT(column_name)

  描述：计算某一列的值占其所属分组中所有值总和的比例。

  参数：数值类型，或任意可以隐式转换成数值类型的类型。

  返回值类型：入参为float4和float8，返回值类型与入参一致；其余入参类型返回numeric类型。

  说明

  RATIO_TO_REPORT(column_name)与OVER()一起使用时，OVER()中入参只支持PARTITION BY和NULL。

  示例1：

```
gaussdb=# CREATE TABLE ratio_to_report_t1(a int, b int);

gaussdb=# INSERT INTO ratio_to_report_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# SELECT a,b,RATIO_TO_REPORT(b) OVER(PARTITION BY a) FROM ratio_to_report_t1;
 a | b |     ratio_to_report
---+---+-------------------------
 1 | 1 |  .14285714285714285714
 1 | 1 |  .14285714285714285714
 1 | 2 |  .28571428571428571429
 1 | 3 |  .42857142857142857143
 2 | 4 |  .44444444444444444444
 2 | 5 |  .55555555555555555556
 3 | 6 | 1.00000000000000000000
(7 rows)

gaussdb=# DROP TABLE ratio_to_report_t1;
```

  示例2：与其它函数嵌套使用。

```
gaussdb=# CREATE TABLE ratio_to_report_t1(a int, b int);

gaussdb=# INSERT INTO ratio_to_report_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT a,b,TO_CHAR(RATIO_TO_REPORT(b) OVER(PARTITION BY a), '$999eeee') FROM
ratio_to_report_t1;
 a | b | to_char
---+---+---------
 1 | 1 |  1e-01
 1 | 1 |  1e-01
 1 | 2 |  3e-01
 1 | 3 |  4e-01
 2 | 4 |  4e-01
 2 | 5 |  6e-01
 3 | 6 |  1e+00
(7 rows)

gaussdb=# DROP TABLE ratio_to_report_t1;
```

示例3：存储过程调用。

```
gaussdb=# CREATE TABLE ratio_to_report_t1(a int, b int);

gaussdb=# INSERT INTO ratio_to_report_t1 VALUES(1,1),(1,1),(1, 2),(1, 3),(2, 4),(2, 5),(3,6);

gaussdb=# CREATE OR REPLACE PROCEDURE ratio_to_report_proc IS CURSOR cur_1 IS SELECT
a,b,RATIO_TO_REPORT(b) OVER(PARTITION BY a) FROM ratio_to_report_t1;
BEGIN
FOR cur IN cur_1 LOOP
RAISE INFO '%', cur.ratio_to_report;
END LOOP;
END;
/

gaussdb=# CALL RATIO_TO_REPORT_PROC();
INFO:  .14285714285714285714
INFO:  .14285714285714285714
INFO:  .28571428571428571429
INFO:  .42857142857142857143
INFO:  .44444444444444444444
INFO:  .55555555555555555556
INFO:  1.00000000000000000000
 ratio_to_report_proc
----------------------

(1 row)

gaussdb=# DROP PROCEDURE ratio_to_report_proc;

gaussdb=# DROP TABLE ratio_to_report_t1;
```

- KEEP(DENSE_RANK { FIRST | LAST } ORDER BY expr)

  描述：将查询结果的数据行按指定规则进行排序，返回第一组或者最后一组的值，可用于聚集函数或窗口函数。

  语法：

```
aggregate_function KEEP ( DENSE_RANK { FIRST | LAST }  ORDER BY expr [ DESC | ASC ] [ NULLS
{ FIRST | LAST } ]
     [, expr [ DESC | ASC ] [ NULLS { FIRST | LAST } ]]...)
OVER ( [query_partition_clause] ) ]
```

  返回值类型：在aggregate_function中指定列的数据类型。

  说明

  - aggregate_function为聚集函数，当前仅支持：MIN、MAX、SUM、AVG、COUNT、VARIANCE和STDDEV。
  - KEEP关键字用于语义清晰，限定于聚集函数/窗口函数。
  - DENSE_RANK FIRST或DENSE_RANK LAST表示仅聚集最小（FIRST）或最大（LAST）密集排序后的行。
  - 不支持生成向量化计划执行 。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CREATE TABLE keep_t1(a int, b int, c int, d int);

gaussdb=# INSERT INTO keep_t1 values(2, 111, 333, 10), (2, 11, 33, 10), (2, 11, 3, 1), (1, 22, 55, 10),
(1, 2, 5, 100);

gaussdb=#  SELECT MIN(d) KEEP(DENSE_RANK FIRST ORDER BY b) OVER (PARTITION BY a)
MIN_FIRST,
   MIN(b) KEEP(DENSE_RANK LAST ORDER BY b) OVER (PARTITION BY a) MIN_LAST,
   MAX(b) KEEP(DENSE_RANK FIRST ORDER BY b) OVER (PARTITION BY a) MAX_FIRST,
   MAX(b) KEEP(DENSE_RANK LAST ORDER BY b) OVER (PARTITION BY a) MAX_LAST FROM keep_t1;
 min_first | min_last | max_first | max_last
-----------+----------+-----------+----------
       100 |       22 |         2 |       22
       100 |       22 |         2 |       22
         1 |      111 |        11 |      111
         1 |      111 |        11 |      111
         1 |      111 |        11 |      111
(5 rows)
gaussdb=# DROP TABLE keep_t1;
```

- fill(value anyelement)

  描述：对每个分组为空的数据进行插值补充，fill可以对现有数据中的空值进行补充，遵从如下规则：

  - 若当前为空数据，返回上一个不为空的值。
  - 若当前不为空数据，返回当前值。

  参数：参数说明如**表1-98**所示

  返回值类型：anyelement

表 1-98 fill 函数参数说明

参数名	类型	描述
Value	anyelement	填充的值。

说明

- fill需要与OVER语句一起使用。
- 核心算法通过COALESCE函数实现。请参见**条件表达式函数**章节中介绍。

示例：

```
gaussdb=# SELECT
gaussdb-#    time,
gaussdb-#    CASE WHEN value is not null THEN value else fill(value) over(partition by time) END
gaussdb-# FROM
gaussdb-#    (VALUES ('2019-07-12 00:00:00'::timestamptz, 1),
gaussdb(#            ('2019-07-12 00:00:00'::timestamptz, NULL),
gaussdb(#            ('2019-07-12 00:02:00'::timestamptz, 3)) v(time,value);
          time          | fill
------------------------+------
 2019-07-12 00:00:00+08 |    1
 2019-07-12 00:00:00+08 |    1
 2019-07-12 00:02:00+08 |    3
(3 rows)
```


版权所有 © 华为云计算技术有限公司

1.6.20 安全函数

安全函数

- gs_encrypt_aes128(encryptstr,keystr)

  描述：使用基于keystr派生出的密钥对encryptstr字符串进行加密，返回加密后的字符串。keystr的长度范围为8~16字节，并且至少包含大写字母、小写字母、数字和特殊字符中的三种字符。支持的加密数据类型：目前数据库支持的数值类型，字符类型，二进制类型中的RAW，日期/时间类型中的DATE、TIMESTAMP、SMALLDATETIME。

  返回值类型：text

  返回值长度：至少为92字节，不超过4*[(Len+68)/3]字节，其中Len为加密前数据长度（单位为字节）。

  示例：

  ```
  gaussdb=# SELECT gs_encrypt_aes128('MPPDB','1234@abc');
                                      gs_encrypt_aes128
  ----------------------------------------------------------------------------------------------
   OF1g3+70oeqFfyKiWlpxfYxPnpeitNc6+7nAe02Ttt37fZF8Q+bbEYhdw/YG+0c9tHKRWM6OcTzlB3HnqvX
  +1d8Bflo=
  (1 row)
  ```

  说明

  由于该函数的执行过程需要传入加密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史。即无法在gsql里通过上下翻页功能找到该函数的执行历史。

- gs_decrypt_aes128(decryptstr,keystr)

  描述：使用基于keystr派生出的密钥对decryptstr字符串进行解密，返回解密后的字符串。解密使用的keystr必须保证与加密时使用的keystr一致才能正确解密。keystr不得为空。

  说明

  此函数需要结合gs_encrypt_aes128加密函数共同使用。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT gs_decrypt_aes128('OF1g3+70oeqFfyKiWlpxfYxPnpeitNc6+7nAe02Ttt37fZF8Q
  +bbEYhdw/YG+0c9tHKRWM6OcTzlB3HnqvX+1d8Bflo=','1234@abc');
   gs_decrypt_aes128
  -------------------
   MPPDB
  (1 row)
  ```

  说明

  由于该函数的执行过程需要传入加密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史；即无法在gsql里通过上下翻页功能找到该函数的执行历史。

- aes_encrypt(str, key_str, init_vector)

  描述：基于AES算法，使用加密口令key_str和初始化向量init_vector对字符串str进行加密。

  参数解释：

  - str: 需要被加密的字符串。若str为NULL，函数返回NULL。
  - key_str: 加密口令。若key_str为NULL，函数返回NULL。为了安全，对于128bit/192bit/256bit的密钥长度（由块加密模式block_encryption_mode确定），建议用户使用128bit/192bit/256bit的安全随机数作为密钥字符串。

 版权所有 © 华为云计算技术有限公司

- init_vector: 为需要它的块加密模式提供初始化变量，长度大于等于16字节（大于16字节的部分会被自动忽略）。str和key_str均不为NULL时，该参数不可为NULL，否则报错。为了安全，建议用户在OFB模式下，保证每次加密IV值的唯一性；在CBC模式和CFB模式下，保证每次加密的IV值不可被预测。

返回值类型：text

示例：

```
gaussdb=# SELECT aes_encrypt('huwei123','123456vfhex4dyu,vdaladhjsadad','1234567890123456');
 aes_encrypt
-------------
 u*8\x05c?0
(1 row)
```

说明

- 该函数仅在GaussDB兼容MYSQL类型时（即sql_compatibility = 'MYSQL'）有效，其他类型不支持该函数。
- 由于该函数的执行过程需要传入加密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史，即无法在gsql里通过上下翻页功能找到该函数的执行历史。
- 在存储过程的相关操作中需尽量避免调用该函数，避免敏感参数信息在日志中泄露的风险。同时建议用户在使用包含该函数的存储过程相关操作时，将该函数的参数信息过滤后再提供给外部维护人员定位，日志使用完后请及时删除。
- 在打开debug_print_plan开关的情况下需尽量避免调用该函数，避免敏感参数信息在日志中泄露的风险。同时建议用户在打开debug_print_plan开关生成的日志中对该函数的参数信息进行过滤后再提供给外部维护人员定位，日志使用完后请及时删除。
- 由于SQL_ASCII设置与其他设置表现得相当不同。如果服务器字符集是SQL_ASCII，服务器把字节值0~127根据 ASCII标准解释，而字节值128~255则当作无法解析的字符。如果设置为SQL_ASCII，就不会有编码转换。该函数调用openssl三方库返回的数据的编码为非ASCII数据，因此当数据库服务端字符集设置为SQL_ASCII时，客户端编码也需设置为SQL_ASCII ，否则会报错。因为数据库不会帮助转换或者校验非ASCII字符。

- aes_decrypt(pass_str, key_str, init_vector)

  描述：基于AES算法，使用解密口令key_str和初始化向量init_vector对字符串str进行解密。

  参数解释：

  - pass_str: 需要被解密的字符串。若pass_str为NULL，函数返回NULL。
  - key_str: 解密口令。若key_str为NULL，函数返回NULL。为了安全，对于128bit/192bit/256bit的密钥长度（由块加密模式block_encryption_mode确定），建议用户使用128bit/192bit/256bit的安全随机数作为密钥字符串。
  - init_vector: 为需要它的块解密模式提供初始化变量，长度大于等于16字节（大于16字节的部分会被自动忽略）。pass_str和key_str均不为NULL时，该参数不可为NULL，否则报错。为了安全，建议用户在OFB模式下，保证每次加密IV值的唯一性；在CBC模式和CFB模式下，保证每次加密的IV值不可被预测。

  返回值类型：text

  示例：

```
gaussdb=# SELECT
aes_decrypt(aes_encrypt('huwei123','123456vfhex4dyu,vdaladhjsadad','1234567890123456'),'123456vf
hex4dyu,vdaladhjsadad','1234567890123456');
 aes_decrypt
-------------
 huwei123
(1 row)
```

 版权所有 © 华为云计算技术有限公司

说明

- 该函数仅在GaussDB兼容MYSQL类型时（即sql_compatibility = 'MYSQL'）有效，其他类型不支持该函数。
- 由于该函数的执行过程需要传入解密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史；即无法在gsql里通过上下翻页功能找到该函数的执行历史。
- 在存储过程的相关操作中需尽量避免调用该函数，避免敏感参数信息在日志中泄露的风险。同时建议用户在使用包含该函数的存储过程相关操作时，将该函数的参数信息过滤后再提供给外部维护人员定位，日志使用完后请及时删除。
- 在打开debug_print_plan开关的情况下需尽量避免调用该函数，避免敏感参数信息在日志中泄露的风险。同时建议用户在打开debug_print_plan开关生成的日志中对该函数的参数信息进行过滤后再提供给外部维护人员定位，日志使用完后请及时删除。
- 若想成功解密，需要保证block_encryption_mode，key_str，iv值与加密时一致。
- 由于编码差异，不支持从gsql客户端直接复制加密后的数据进行解密，此场景解密出的结果不一定是加密前的字符串。
- 由于SQL_ASCII设置与其他设置表现得相当不同。如果服务器字符集是SQL_ASCII，服务器把字节值0~127根据 ASCII标准解释，而字节值128~255则当作无法解析的字符。如果设置为SQL_ASCII，就不会有编码转换。该函数调用openssl三方库返回的数据的编码为非ASCII数据，因此当数据库服务端字符集设置为SQL_ASCII时，客户端编码也需设置为SQL_ASCII ，否则会报错。因为数据库不会帮助转换或者校验非ASCII字符。

- gs_digest(input_string, hash_algorithm)

  描述：使用指定的哈希算法，对输入的字符串计算哈希，并且以十六进制数作为返回值。

  参数解释：

  – input_string：需要被计算哈希的字符串，不能为NULL。
  – hash_algorithm：指定的哈希计算算法，当前支持SHA256，SHA384，SHA512和SM3，支持大写和小写。使用不支持的哈希算法则会报错。

  返回值类型：text

  示例：

```
gaussdb=# SELECT pg_catalog.gs_digest('gaussdb', 'sha256');
                            gs_digest
------------------------------------------------------------------
 4dc50d746f4e04f9b446986b34a0050e358fbfb8bc1fba314c54b52a417b0b8e
(1 row)
```

- gs_password_deadline()

  描述：显示当前账户密码离过期还距离多少天。

  返回值类型：interval

  示例：

```
gaussdb=# SELECT gs_password_deadline();
  gs_password_deadline
------------------------
 83 days 17:44:32.196094
(1 row)
```

- gs_password_notifytime()

  描述：显示账户密码到期前提醒的天数。

  返回值类型：int32

- gs_password_lifetime(roleName)

  描述：显示查询账户密码离过期还有多少时间。

  参数：需要查询的账户名，不传参数则默认为当前登录的账户。

版权所有 © 华为云计算技术有限公司

返回值类型：string

示例：

如果账户没有设置密码有效期，返回结果如下：

```
gaussdb=# SELECT gs_password_lifetime('user2');
 gs_password_lifetime
----------------------
 -
(1 row)
```

如果账户设置密码有效期，返回结果如下：

```
gaussdb=# SELECT gs_password_lifetime('user1');
 gs_password_lifetime
----------------------
 9 days 23:38:30.509614
(1 row)
```

说明

gs_password_deadline()查询的是password_effect_time设置的密码到期时间，gs_password_lifetime()是账户密码设置的到期时间。

- login_audit_messages(BOOLEAN)

  描述：查看登录用户的登录信息。

  返回值类型：元组

  示例：

  - 查看上一次登录成功的日期、时间和IP等信息。

    ```
    gaussdb=> SELECT * FROM login_audit_messages(true);
     username | database |       logintime        |    mytype     | result | client_conninfo
    ----------+----------+------------------------+---------------+--------+-----------------
     omm      | postgres | 2020-06-29 21:56:40+08 | login_success | ok     | gsql@[local]
    (1 row)
    ```

  - 查看自从上一次登录成功以来登录失败的尝试次数、日期和时间。

    ```
    gaussdb=>  SELECT * FROM login_audit_messages(false);
     username | database |       logintime        |    mytype    | result |  client_conninfo
    ----------+----------+------------------------+--------------+--------+-------------------
     omm      | postgres | 2020-06-29 21:57:55+08 | login_failed | failed | [unknown]@[local]
     omm      | postgres | 2020-06-29 21:57:53+08 | login_failed | failed | [unknown]@[local]
    (2 rows)
    ```

- login_audit_messages_pid()

  描述：查看登录用户的登录信息。与login_audit_messages的区别在于结果基于当前backendid向前查找。所以不会因为同一用户的后续登录，而影响本次登录的查询结果。也就是查询不到该用户后续登录的信息。

  返回值类型：元组

  说明

  在开启线程池的情况下，由于线程切换，同一session中获取到的backendid可能会发生变化，因此会造成多次调用该函数返回值不一致的情况。不建议用户在开启线程池的情况下调用此函数。

  示例：

  - 查看上一次登录成功的日期、时间和IP等信息。

    ```
    gaussdb=> SELECT * FROM login_audit_messages_pid(true);
     username | database |       logintime        |    mytype     | result | client_conninfo |    backendid
    ----------+----------+------------------------+---------------+--------+-----------------+-----------------
     omm      | postgres | 2020-06-29 21:56:40+08 | login_success | ok     | gsql@[local]    |
    139823109633792
    (1 row)
    ```

  - 查看自从上一次登录成功以来登录失败的尝试次数、日期和时间。

版权所有 © 华为云计算技术有限公司

```
gaussdb=> SELECT * FROM login_audit_messages_pid(false);
 username | database |       logintime        |    mytype    | result |  client_conninfo  |    backendid
----------+----------+------------------------+--------------+--------+-------------------+-----------------
 omm      | postgres | 2020-06-29 21:57:55+08 | login_failed | failed | [unknown]@[local] |
139823109633792
 omm      | postgres | 2020-06-29 21:57:53+08 | login_failed | failed | [unknown]@[local] |
139823109633792
(2 rows)
```

- inet_server_addr()

  描述：显示服务器IP信息。

  返回值类型：inet

  示例：

  ```
  gaussdb=# SELECT inet_server_addr();
   inet_server_addr
  ------------------
   10.10.0.13
  (1 row)
  ```

  说明

  - 上面是以客户端在10.10.0.50上，服务器端在10.10.0.13上为例。
  - 如果是通过本地连接，使用此接口显示为空。

- inet_client_addr()

  描述：显示客户端IP信息。

  返回值类型：inet

  示例：

  ```
  gaussdb=# SELECT inet_client_addr();
   inet_client_addr
  ------------------
   10.10.0.50
  (1 row)
  ```

  说明

  - 上面是以客户端在10.10.0.50上，服务器端在10.10.0.13上为例。
  - 如果是通过本地连接，使用此接口显示为空。

- gs_query_audit(starttime, endtime, audit_log)

  描述：查看当前CN节点传统审计日志。

  参数：timestamptz starttime，timestamptz endtime[,audit_log]

  返回值类型：record

  函数返回字段如**表1-99**所示。

  表 1-99 gs_query_audit 返回字段信息

名称	类型	描述
time	timestamp with time zone	操作时间
type	text	操作类型
result	text	操作结果
userid	oid	用户id

版权所有 © 华为云计算技术有限公司

名称	类型	描述
username	text	执行操作的用户名
database	text	数据库名称
client_conninfo	text	客户端连接信息
object_name	text	操作对象名称
detail_info	text	执行操作详细信息
node_name	text	节点名称
thread_id	text	线程id
local_port	text	本地端口
remote_port	text	远端端口

示例：

```
gaussdb=# SELECT * FROM gs_query_audit('2025-01-14 11:35:41','2025-01-14 11:35:42');
          time          |     type      | result | userid | username | database | client_conninfo | object_name
|                       detail_info                        |  node_name   |            thread_id
        | local_port | remote_port
------------------------+---------------+--------+--------+----------+----------+-----------------+-------------
+-----------------------------------------------------------------------+--------------+-----------------------
---------+------------+-------------
 2025-01-14 11:35:41+08 | login_success | ok     | 10     |  omm     | postgres | gsql@[local]    |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941594432 | 45002      | null
 2025-01-14 11:35:41+08 | user_logout   | ok     | 10     |  omm     | postgres | gsql@[local]    |
postgres    | logout db(postgres) success                                     | coordinator1 |
139779518232320@790140941676326 | 45002      | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     |  omm     | postgres | gsql@[local]    |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941809559 | 45002      | null
(3 rows)
```

- pg_query_audit(starttime, endtime, audit_log)

  描述：查看当前CN节点传统审计日志。

  参数：timestamptz starttime，timestamptz endtime[,audit_log]

  返回值类型：record

  函数返回字段如**表1-100**所示。

  表 1-100 pg_query_audit 返回字段信息

名称	类型	描述
time	timestamp with time zone	操作时间
type	text	操作类型
result	text	操作结果

版权所有 © 华为云计算技术有限公司

名称	类型	描述
userid	oid	用户id
username	text	执行操作的用户名
database	text	数据库名称
client_conninfo	text	客户端连接信息
object_name	text	操作对象名称
detail_info	text	执行操作详细信息
node_name	text	节点名称
thread_id	text	线程id
local_port	text	本地端口
remote_port	text	远端端口

示例：

```
gaussdb=# SELECT * FROM pg_query_audit('2025-01-14 11:35:41','2025-01-14 11:35:42');
         time          |     type      | result | userid | username | database | client_conninfo | object_name
|                        detail_info                         |  node_name   |           thread_id      | local_port |
remote_port
-----------------------+---------------+--------+--------+----------+----------+-----------------+-------------
+------------------------------------------------------------------+--------------+--------------------------------
+------------+-------------
 2025-01-14 11:35:41+08 | login_success | ok     | 10     |  omm     | postgres | gsql@[local]    |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941594432 | 45002       | null
 2025-01-14 11:35:41+08 | user_logout   | ok     | 10     |  omm     | postgres | gsql@[local]    |
postgres    | logout db(postgres) success                                     | coordinator1 |
139779518232320@790140941676326 | 45002       | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     |  omm     | postgres | gsql@[local]    |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941809559 | 45002       | null
(3 rows)
```

- gs_query_audit_dist(starttime, endtime)

  描述：查看所有CN节点传统审计日志。函数返回字段同gs_query_audit函数。

  参数：timestamptz starttime，timestamptz endtime

  返回值类型：record

  示例：

```
gaussdb=# SELECT * FROM gs_query_audit_dist('2025-01-14 11:35:41','2025-01-14 11:35:42');
         time          |     type      | result | userid | username | database | client_conninfo | object_name
|                        detail_info                         |  node_name   |           thread_id       | local_port |
remote_port
-----------------------+---------------+--------+--------+----------+----------+-----------------+-------------
+------------------------------------------------------------------+--------------+--------------------------------
+------------+-------------
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | gsql@[local]    |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941594432 | 45002       | null
 2025-01-14 11:35:41+08 | user_logout   | ok     | 10     | omm      | postgres | gsql@[local]    |
postgres    | logout db(postgres) success                                     | coordinator1 |
139779518232320@790140941676326 | 45002       | null
```


版权所有 © 华为云计算技术有限公司

```
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941809559 | 45002      | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator2 |
140242642269952@790140941703387 | 45004      | null
 2025-01-14 11:35:41+08 | user_logout   | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | logout db(postgres) success                                     | coordinator2 |
140242642269952@790140941784961 | 45004      | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | coordinator1@::1 |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator2 |
140242642269952@790140941846544 | 45004      | 49014
(6 rows)
```

- pgxc_query_audit(starttime, endtime)

  描述：查看所有CN节点传统审计日志。函数返回字段同gs_query_audit函数。

  参数：timestamptz starttime，timestamptz endtime

  返回值类型：record

  示例：

```
gaussdb=# SELECT * FROM pgxc_query_audit('2025-01-14 11:35:41','2025-01-14 11:35:42');
          time          |     type      | result | userid | username | database | client_conninfo | object_name
|                        detail_info                         |  node_name   |           thread_id       | local_port |
remote_port
------------------------+---------------+--------+--------+----------+----------+-----------------+-------------
+------------------------------------------------------------------+--------------+---------------------------------
+------------+-------------
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941594432 | 45002      | null
 2025-01-14 11:35:41+08 | user_logout   | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | logout db(postgres) success                                     | coordinator1 |
139779518232320@790140941676326 | 45002      | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator1 |
139779518232320@790140941809559 | 45002      | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator2 |
140242642269952@790140941703387 | 45004      | null
 2025-01-14 11:35:41+08 | user_logout   | ok     | 10     | omm      | postgres | gsql@[local]     |
postgres    | logout db(postgres) success                                     | coordinator2 |
140242642269952@790140941784961 | 45004      | null
 2025-01-14 11:35:41+08 | login_success | ok     | 10     | omm      | postgres | coordinator1@::1 |
postgres    | login db(postgres) success,the current user is:omm, SSL=off     | coordinator2 |
140242642269952@790140941846544 | 45004      | 49014
(6 rows)
```

- gs_query_unified_audit(starttime, endtime, audit_log)

  描述：查看所有CN节点统一审计日志。

  参数：timestamptz starttime，timestamptz endtime[,audit_log text]

  返回值类型：record

  函数返回字段如下：

  表 1-101 统一审计日志参数

名称	描述
time	操作时间。
type	操作类型。
result	操作结果。

版权所有 © 华为云计算技术有限公司

名称	描述
userid	用户ID。
username	执行操作的用户名。
database	数据库名称。
client_conninfo	客户端连接信息。
object_name	操作对象名称。
detail_info	执行操作详细信息。
node_name	节点名称。
thread_id	线程ID。
local_port	本地端口。
remote_port	远端端口。
policy_id	统一审计策略ID。
unified_audit_type	统一审计策略类型。
unified_audit_policy	统一审计策略信息。

示例：

```
gaussdb=# SELECT * FROM gs_query_unified_audit(sysdate-1,sysdate+1) WHERE type='audit_policy';
          time          |     type     | result | userid | username | database | client_conninfo |
object_name               |                        detail_info                         |  node_name   |
         thread_id          | local_port | remote_port | policy_id | unified_audit_type |
unified_audit_policy
------------------------+--------------+--------+--------+----------+----------+-----------------
+-----------------------------------------+------------------------------------------------------------------
+--------------+--
-------------------------------+------------+-------------+-----------+--------------------
+-----------------------------------------------------------------------
 2025-05-07 15:17:17+08 | audit_policy | ok     | 16397  | user001  | postgres | gsql@[local]     | table:
[public.table_security_auditing] | INSERT INTO table_security_auditing VALUES(1,'22');          |
coordinator1 | 1
40658670036736@799917437811493 | 45102      | 46946       | 16421     | access             | INSERT
 2025-05-07 15:17:17+08 | audit_policy | ok     | 16397  | user001  | postgres | gsql@[local]     | table:
[public.table_security_auditing] | UPDATE table_security_auditing SET name=234123 WHERE id=1;   |
coordinator1 | 1
40658670036736@799917437817363 | 45102      | 46946       | 16421     | access             | UPDATE
 2025-05-07 15:17:17+08 | audit_policy | ok     | 16397  | user001  | postgres | gsql@[local]     | table:
[public.table_security_auditing] | DELETE FROM table_security_auditing WHERE id=1;               |
coordinator1 | 1
40658670036736@799917437820418 | 45102      | 46946       | 16421     | access             | DELETE
 2025-05-07 15:17:19+08 | audit_policy | ok     | 16397  | user001  | postgres | gsql@[local]     | table:
[public.table_security_auditing] | TRUNCATE TABLE table_security_auditing;                       |
coordinator1 | 1
40658670036736@799917439207152 | 45102      | 46946       | 16421     | access             | TRUNCATE
 2025-05-07 15:17:23+08 | audit_policy | ok     | 16397  | user001  | postgres | gsql@[local]     |
public.table_security_auditing           | GRANT INSERT ON TABLE table_security_auditing TO user002;
| coordinator1 | 1
40658670036736@799917443968269 | 45102      | 46946       | 16406     | privilege          | GRANT ON
TABLE postgres.public.table_security_auditing TO user002
```


版权所有 © 华为云计算技术有限公司

```
 2025-05-07 15:17:25+08 | audit_policy | ok     | 16397  | user001  | postgres | gsql@[local]    |
public.table_security_auditing          | REVOKE INSERT ON TABLE table_security_auditing FROM
user002; | coordinator1 | 1
40658670036736@799917445034551 | 45102      | 46946      | 16406     | privilege          | REVOKE
ON TABLE postgres.public.table_security_auditing FROM user002
(6 rows)
```

- gs_delete_audit(starttime, endtime)

  描述：删除指定时间段的审计日志。

  参数：timestamptz starttime，timestamptz endtime

  返回值类型：void

  示例：

  ```
  gaussdb=# SELECT * FROM gs_delete_audit('2025-01-14 00:00:00','2025-01-14 00:01:00');
   gs_delete_audit
  -----------------

  (1 row)
  ```

- pg_delete_audit(starttime, endtime)

  描述：删除指定时间段的审计日志。

  参数：timestamptz starttime，timestamptz endtime

  返回值类型：void

  示例：

  ```
  gaussdb=# SELECT * FROM pg_delete_audit('2025-01-14 00:00:00','2025-01-14 00:01:00');
   pg_delete_audit
  -----------------

  (1 row)
  ```

- alldigitsmasking()

  描述：脱敏策略的内部函数，对所有字符进行脱敏。

  参数：col text, letter character default '0'

  返回值类型：text

- creditcardmasking()

  描述：脱敏策略的内部函数，对所有信用卡信息进行脱敏。

  参数：col text, letter character default 'x'

  返回值类型：text

- randommasking()

  描述：脱敏策略的内部函数，使用随机策略。

  参数：col text,

  返回值类型：text

- fullemailmasking()

  描述：脱敏策略的内部函数，对出现最后一个'.'之前的文本（除'@'之外）进行脱敏。

  参数：col text, letter character default 'x'

  返回值类型：text

- basicemailmasking()

  描述：脱敏策略的内部函数，对出现第一个'@'之前的文本进行脱敏。

  参数：col text, letter character default 'x'

  返回值类型：text

版权所有 © 华为云计算技术有限公司

- shufflemasking()

  描述：脱敏策略的内部函数，对字符进行乱序排列。

  参数：col text

  返回值类型：text

- regexpmasking()

  描述：脱敏策略的内部函数，对字符进行正则表达式替换。

  参数：col text, reg text, replace_text text, pos INTEGER default 0, reg_len INTEGER default -1

  返回值类型：text

- gs_encrypt(encryptstr, keystr, encrypttype)

  描述：根据encrypttype，以keystr为密钥对encryptstr字符串进行加密，返回加密后的字符串。

  返回值类型：text

参数	类型	描述	取值范围
encryptstr	text	需要加密的数据	-
keystr	text	加密口令	8~16字节，至少包含3种字符（大写字母、小写字母、数字、特殊字符）
encrypttype	text	加解密类型（不区分大小写）	aes128、sm4、aes128_cbc_sha256、aes256_cbc_sha256、aes128_gcm_sha256、aes256_gcm_sha256、sm4_ctr_sm3

  示例：

```
gaussdb=#  SELECT gs_encrypt('MPPDB', 'Asdf1234', 'sm4');
       gs_encrypt
 ------------------------------
 ZBzOmaGA4Bb+coyucJ0B8AkIShqc
(1 row)
```

  说明

  由于该函数的执行过程需要传入加密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史。即无法在gsql里通过上下翻页功能找到该函数的执行历史。

  aes128、sm4为兼容旧版本参数。aes128加解密使用AES128 CBC模式，使用SHA1进行完整性校验。sm4加解密算法使用SM4 CTR模式，不进行完整性校验。

- gs_decrypt(decryptstr, keystr, decrypttype)

  描述：根据decrypttype，以keystr为解密口令对decryptstr字符串进行解密，返回解密后的字符串。解密使用的decrypttype 及keystr必须保证与加密时使用的encrypttype 及keystr一致才能正常解密。keystr不得为空。

  此函数需要结合gs_encrypt加密函数共同使用。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

参数	类型	描述	取值范围
decryptstr	text	需要解密的数据	-
keystr	text	解密口令	8~16字节，至少包含3种字符（大写字母、小写字母、数字、特殊字符）
decrypttype	text	加解密类型（不区分大小写）	aes128、sm4、aes128_cbc_sha256、aes256_cbc_sha256、aes128_gcm_sha256、aes256_gcm_sha256、sm4_ctr_sm3

示例：

```
gaussdb=# SELECT gs_decrypt('ZBzOmaGA4Bb+coyucJ0B8AkIShqc', 'Asdf1234', 'sm4');
 gs_decrypt
------------
 MPPDB
(1 row)
```

说明

由于该函数的执行过程需要传入解密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史；即无法在gsql里通过上下翻页功能找到该函数的执行历史。

aes128、sm4为兼容旧版本参数。aes128加解密使用AES128 CBC模式，使用SHA1进行完整性校验。sm4加解密算法使用SM4 CTR模式，不进行完整性校验。

- gs_encrypt_bytea(encryptstr, keystr, encrypttype)

  描述：根据encrypttype，以keystr为加密口令对encryptstr字符串进行加密，返回加密后的字符串。

  返回值类型：bytea

参数	类型	描述	取值范围
encryptstr	text	需要加密的数据	-
keystr	text	加密口令	8~16字节，至少包含3种字符（大写字母、小写字母、数字、特殊字符）
encrypttype	text	加解密类型（不区分大小写）	aes128_cbc_sha256、aes256_cbc_sha256、aes128_gcm_sha256、aes256_gcm_sha256、sm4_ctr_sm3

示例：

```
gaussdb=#  SELECT gs_encrypt_bytea('MPPDB', 'Asdf1234', 'sm4_ctr_sm3');
                                                                                                                                gs_encrypt_bytea
 ---------------------------------------------------------------------------------------------------------------------------------------------------------

\x90e286971c2c70410def0a2814af4ac44c737926458b66271d9d1547bc937395ca018d7755672fa9dc3cdc6ec4a76001dc0e137f3bc5c8a5c51143561f1d09a848bfdebfec5e
(1 row)
```


版权所有 © 华为云计算技术有限公司

说明

由于该函数的执行过程需要传入加密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史，即无法在gsql里通过上下翻页功能找到该函数的执行历史。

- gs_decrypt_bytea(decryptstr, keystr, decrypttype)

  描述：根据decrypttype，以keystr为解密口令对decryptstr字符串进行解密，返回解密后的字符串。解密使用的decrypttype及keystr必须保证与加密时使用的encrypttype 及keystr一致才能正常解密。keystr不得为空。

  此函数需要结合gs_encrypt_bytea加密函数共同使用。

  返回值类型：text

参数	类型	描述	取值范围
decryptstr	bytea	需要解密的数据	-
keystr	text	解密口令	8~16字节，至少包含3种字符（大写字母、小写字母、数字、特殊字符）
decrypttype	text	加解密类型（不区分大小写）	aes128_cbc_sha256、aes256_cbc_sha256、aes128_gcm_sha256、aes256_gcm_sha256、sm4_ctr_sm3

  示例：

```
gaussdb=# SELECT
gs_decrypt_bytea('\x90e286971c2c70410def0a2814af4ac44c737926458b66271d9d1547bc937395ca018
d7755672fa9dc3cdc6ec4a76001dc0e137f3bc5c8a5c51143561f1d09a848bfdebfec5e', 'Asdf1234',
'sm4_ctr_sm3');
 gs_decrypt_bytea
------------------
 MPPDB
(1 row)
```

说明

由于该函数的执行过程需要传入解密口令，为了安全起见，gsql工具不会将包含该函数名字样的SQL记录入执行历史；即无法在gsql里通过上下翻页功能找到该函数的执行历史。

1.6.21 账本数据库的函数

- get_dn_hist_relhash(text, text)

  描述：返回指定防篡改用户表的表级数据hash值。该函数仅供分布式节点间或通过EXECUTE DIRECT ON调用，所有用户直接调用均提示权限不足。

  参数类型：text

  返回值类型：hash16

- ledger_hist_check(text, text)

  描述：校验指定防篡改用户表的表级数据hash值与其对应历史表hash一致性。

  参数类型：text

  返回值类型：Boolean

  示例：

版权所有 © 华为云计算技术有限公司

```
--创建schema。
gaussdb=# CREATE SCHEMA ledgernsp WITH BLOCKCHAIN;
--创建表。
gaussdb=# CREATE TABLE ledgernsp.tab(a int, b text);
--插入数据。
gaussdb=# INSERT INTO ledgernsp.tab values(generate_series(1, 10000), 'test');
gaussdb=# SELECT ledger_hist_check('ledgernsp','tab');
 ledger_hist_check
-------------------
 t
(1 row)
```

- ledger_hist_repair(text, text)

  描述：修复指定防篡改用户表对应的历史表hash值，使之与用户表hash一致，返回hash差值。

  参数类型：text

  返回值类型：hash16

  示例：

```
gaussdb=# SELECT ledger_hist_repair('ledgernsp','tab');
 ledger_hist_repair
--------------------
 0000000000000000
(1 row)
```

- ledger_hist_archive(text, text)

  描述：归档指定防篡改用户表对应的历史表至审计日志目录中hist_back文件夹下。如果用户历史表名称中带有'/'符号，归档所生成的文件名中将会把'/'替换为'_'。

  参数类型：text

  返回值类型：Boolean

  示例：

```
gaussdb=# EXECUTE DIRECT ON (datanode1) 'select pg_catalog.ledger_hist_archive(''ledgernsp'',
''tab'');';
 ledger_hist_archive
---------------------
 t
(1 row)
```

- ledger_gchain_check(text, text)

  描述：校验指定防篡改用户表对应的历史表hash与全局历史表对应的relhash一致性。

  参数类型：text

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT ledger_gchain_check('ledgernsp','tab');
 ledger_gchain_check
---------------------
 t
(1 row)
```

- ledger_gchain_repair(text, text)

  描述：修复指定防篡改用户表在全局历史表中的relhash，使之与其历史表hash一致，返回指定表的hash总和。

  参数类型：text

  返回值类型：hash16

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT ledger_gchain_repair('ledgernsp','tab');
 ledger_gchain_repair
----------------------
 da30c1260af5be50
(1 row)
```

- ledger_gchain_archive(void)

  描述：归档全局历史表至审计日志目录中hist_back文件夹下。

  参数类型：void

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT ledger_gchain_archive();
 ledger_gchain_archive
-----------------------
 t
(1 row)
--清理环境
gaussdb=# DROP TABLE ledgernsp.tab;
gaussdb=# DROP SCHEMA ledgernsp;
```

- hash16in(cstring)

  描述：将输入16进制字符串转化成内部hash16形式。

  参数类型：cstring

  返回值类型：hash16

- hash16out(hash16)

  描述：将内部hash16类型的数据转码转化为16进制cstring类型。

  参数类型：hash16

  返回值类型：cstring

- hash32in(cstring)

  描述：将输入的16进制字符串(32个字符)转化成内部类型hash32形式。

  参数类型：cstring

  返回值类型：hash32

- hash32out(hash32)

  描述：将内部hash32类型的数据转码转化为16进制cstring类型。

  参数类型：hash32

  返回值类型：cstring

1.6.22 密态函数和操作符

- byteawithoutorderwithequalcolin(cstring)

  描述：将输入转码转化成内部byteawithoutorderwithequalcol形式。

  参数类型：cstring

  返回值类型：byteawithoutorderwithequalcol

- byteawithoutorderwithequalcolout(byteawithoutorderwithequalcol)

  描述：将内部byteawithoutorderwithequalcol类型的数据转码转化为cstring类型。

  参数类型：byteawithoutorderwithequalcol

  返回值类型：cstring

版权所有 © 华为云计算技术有限公司

- byteawithoutorderwithequalcolsend(byteawithoutorderwithequalcol)
  描述：将byteawithoutorderwithequalcol类型的数据转码转化为bytea类型。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：bytea
- byteawithoutorderwithequalcolrecv(internal)
  描述：将internal类型的数据转码转化为byteawithoutorderwithequalcol类型。
  参数类型：internal
  返回值类型：byteawithoutorderwithequalcol
- byteawithoutorderwithequalcoltypmodin(cstring[])
  描述：将cstring[]类型的数据转码转化为byteawithoutorderwithequalcol类型。
  参数类型：cstring[]
  返回值类型：int4
- byteawithoutorderwithequalcoltypmodout(int4)
  描述：将int4类型的数据转码转化为cstring类型。
  参数类型：int4
  返回值类型：cstring
- byteawithoutordercolin(cstring)
  描述：将输入转码转化成内部byteawithoutordercolin形式。
  参数类型：cstring
  返回值类型：byteawithoutordercol
- byteawithoutordercolout(byteawithoutordercol)
  描述：将内部byteawithoutordercol类型的数据转码转化为cstring类型。
  参数类型：byteawithoutordercol
  返回值类型：cstring
- byteawithoutordercolsend(byteawithoutordercol)
  描述：将byteawithoutordercol类型的数据转码转化为bytea类型。
  参数类型：byteawithoutordercol
  返回值类型：bytea
- byteawithoutordercolrecv(internal)
  描述：将internal类型的数据转码转化为byteawithoutordercol类型。
  参数类型：internal
  返回值类型：byteawithoutordercol
- byteawithoutorderwithequalcolcmp(byteawithoutorderwithequalcol, byteawithoutorderwithequalcol)
  描述：比较两个byteawithoutorderwithequalcol类型的数据大小，若第一个参数小于第二个参数，返回-1；若等于，返回0；若大于，则返回1。
  参数类型：byteawithoutorderwithequalcol, byteawithoutorderwithequalcol
  返回值类型：int4
- byteawithoutorderwithequalcolcmpbytear(byteawithoutorderwithequalcol, bytea)
  描述：比较byteawithoutorderwithequalcol和bytea数据大小，若第一个参数小于第二个参数，返回-1；若等于，返回0；若大于，则返回1。

版权所有 © 华为云计算技术有限公司

参数类型：byteawithoutorderwithequalcol, bytea

返回值类型：int4

- byteawithoutorderwithequalcolcmpbyteal(bytea, byteawithoutorderwithequalcol)

  描述：比较bytea和byteawithoutorderwithequalcol数据大小，若第一个参数小于第二个参数，返回-1；若等于，返回0；若大于，则返回1。

  参数类型：bytea、byteawithoutorderwithequalcol

  返回值类型：int4

- byteawithoutorderwithequalcoleq(byteawithoutorderwithequalcol, byteawithoutorderwithequalcol)

  描述：比较两个byteawithoutorderwithequalcol类型的数据是否相同，相同则返回true，否则返回false。

  参数类型：byteawithoutorderwithequalcol、byteawithoutorderwithequalcol

  返回值类型：bool

- byteawithoutorderwithequalcoleqbyteal(bytea, byteawithoutorderwithequalcol)

  描述：比较bytea和byteawithoutorderwithequalcol数据是否相同，相同则返回true，否则返回false。

  参数类型：bytea，byteawithoutorderwithequalcol

  返回值类型：bool

- byteawithoutorderwithequalcoleqbytear(byteawithoutorderwithequalcol, bytea)

  描述：比较byteawithoutorderwithequalcol和bytea数据是否相同，相同则返回true，否则返回false。

  参数类型：byteawithoutorderwithequalcol, bytea

  返回值类型：bool

- byteawithoutorderwithequalcolne(byteawithoutorderwithequalcol, byteawithoutorderwithequalcol)

  描述：比较两个byteawithoutorderwithequalcol类型的数据是否不相同，不相同则返回true，否则返回false。

  参数类型：byteawithoutorderwithequalcol, byteawithoutorderwithequalcol

  返回值类型：bool

- byteawithoutorderwithequalcolnebyteal(bytea, byteawithoutorderwithequalcol)

  描述：比较bytea和byteawithoutorderwithequalcol数据是否相同，相同则返回true，否则返回false。

  参数类型：bytea，byteawithoutorderwithequalcol

  返回值类型：bool

- byteawithoutorderwithequalcolnebytear(byteawithoutorderwithequalcol, bytea)

  描述：比较byteawithoutorderwithequalcol和bytea数据是否不相同，相同则返回true，否则返回false。

  参数类型：byteawithoutorderwithequalcol, bytea

  返回值类型：bool

版权所有 © 华为云计算技术有限公司

- hll_hash_byteawithoutorderwithequalcol(byteawithoutorderwithequalcol)
  描述：返回byteawithoutorderwithequalcol的hll哈希值。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：hll_hashval
- tee_sortsupport(internal)
  描述：排序辅助函数，执行排序时供内部调用，不支持用户调用。
  参数类型：internal
  返回值类型：void
- teegtsel(internal, oid, internal, integer)
  描述：密文比较大于和大于等于选择率函数。返回默认值0.3333333333333333。不支持用户调用。
  参数类型：internal, oid, internal, integer
  返回值类型：double precision
- teeltsel(internal, oid, internal, integer)
  描述：密文比较小于和小于等于选择率函数，返回默认值0.3333333333333333。不支持用户调用。
  参数类型：internal, oid, internal, integer
  返回值类型：double precision
- teelikesel(internal, oid, internal, integer)
  描述：密文模糊匹配选择率函数。返回默认值0.005。不支持用户调用。
  参数类型：internal, oid, internal, integer
  返回值类型：double precision
- teenlikesel(internal, oid, internal, integer)
  描述：密文不能模糊匹配选择率函数，返回默认值0.005。不支持用户调用。
  参数类型：internal, oid, internal, integer
  返回值类型：double precision
- sum(byteawithoutorderwithequalcol)
  描述：密文求和聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- avg(byteawithoutorderwithequalcol)
  描述：密文求均值聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- min(byteawithoutorderwithequalcol)
  描述：密文求极小值聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- max(byteawithoutorderwithequalcol)
  描述：密文求极大值聚合运算。
  参数类型：byteawithoutorderwithequalcol

版权所有 © 华为云计算技术有限公司

返回值类型：byteawithoutorderwithequalcol

- stddev_samp(byteawithoutorderwithequalcol)
  描述：密文样本标准差聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- stddev_pop(byteawithoutorderwithequalcol)
  描述：密文总体标准差聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- var_samp(byteawithoutorderwithequalcol)
  描述：密文样本方差聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- var_pop(byteawithoutorderwithequalcol)
  描述：密文总体方差聚合运算。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- tee_trans(byteawithoutorderwithequalcol, byteawithoutorderwithequalcol)
  描述：密文聚合运算内部处理函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- tee_collect(byteawithoutorderwithequalcol, byteawithoutorderwithequalcol)
  描述：密文聚合运算内部处理函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- tee_final(byteawithoutorderwithequalcol)
  描述：密文聚合运算内部处理函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_deterministic(text, oid)
  描述：将text类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id，当前版本暂不支持。
  参数类型：text, oid
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_deterministic(int1, oid)
  描述：将int1类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。
  参数类型：int1, oid
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_deterministic(int2, oid)

版权所有 © 华为云计算技术有限公司

描述：将int2类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

参数类型：int2, oid

返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(int4, oid)

  描述：将int4类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：int4, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(int8, oid)

  描述：将int8类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：int8, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(float4, oid)

  描述：将float4类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：float4, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(float8, oid)

  描述：将float8类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：float8, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(numeric, oid)

  描述：将numeric类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：numeric, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(clob, oid)

  描述：将clob类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：clob, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(varchar, oid)

  描述：将varchar类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：varchar, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(nvarchar2, oid)

  描述：将nvarchar2类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：nvarchar2, oid

版权所有 © 华为云计算技术有限公司

返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(bpchar, oid)

  描述：将bpchar类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：bpchar, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_deterministic(bytea, oid)

  描述：将bytea类型的明文数据加密至密态等值结构的密文，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：bytea, oid

  返回值类型：byteawithoutorderwithequalcol

- ce_decrypt_deterministic(byteawithoutorderwithequalcol, oid)

  描述：将密态等值结构的密文解密至text类型的明文数据，密钥oid为系统表gs_column_keys中的column_key_distributed_id。当前版本暂不支持。

  参数类型：byteawithoutorderwithequalcol, oid

  返回值类型：text

说明

以下以ce_encrypt/ce_decrypt开头的函数存在于pg_catalog命名空间。

- ce_encrypt_text(text, int4, boolean, int4, internal)

  描述：全密态类型转换加密函数。不支持用户调用。

  参数类型：text, int4, boolean, int4, internal

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_varchar(varchar, int4, boolean, int4, internal)

  描述：全密态类型转换加密函数。不支持用户调用。

  参数类型：varchar, int4, boolean, int4, internal

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_nvarchar2(nvarchar2, int4, boolean, int4, internal)

  描述：全密态类型转换加密函数。不支持用户调用。

  参数类型：nvarchar2, int4, boolean, int4, internal

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_bpchar(bpchar, int4, boolean, int4, internal)

  描述：全密态类型转换加密函数。不支持用户调用。

  参数类型：bpchar, int4, boolean, int4, internal

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_int1(int1, int4, boolean, int4, internal)

  描述：全密态类型转换加密函数。不支持用户调用。

  参数类型：int1, int4, boolean, int4, internal

  返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_int2(int2, int4, boolean, int4, internal)

  描述：全密态类型转换加密函数。不支持用户调用。

  参数类型：int2, int4, boolean, int4, internal

版权所有 © 华为云计算技术有限公司

返回值类型：byteawithoutorderwithequalcol

- ce_encrypt_int4(int4, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：int4, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_int8(int8, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：int8, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_float4(float4, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：float4, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_float8(float8, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：float8, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_numeric(numeric, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：numeric, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_bytea(bytea, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：bytea, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_encrypt_clob(clob, int4, boolean, int4, internal)
  描述：全密态类型转换加密函数。不支持用户调用。
  参数类型：clob, int4, boolean, int4, internal
  返回值类型：byteawithoutorderwithequalcol
- ce_decrypt_text(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：text
- ce_decrypt_varchar(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：varchar
- ce_decrypt_nvarchar2(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。

版权所有 © 华为云计算技术有限公司

参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
返回值类型：nvarchar2

- ce_decrypt_bpchar(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：bpchar
- ce_decrypt_int1(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：int1
- ce_decrypt_int2(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：int2
- ce_decrypt_int4(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：int4
- ce_decrypt_int8(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：int8
- ce_decrypt_float4(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：float4
- ce_decrypt_float8(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：float8
- ce_decrypt_numeric(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal
  返回值类型：numeric
- ce_decrypt_bytea(byteawithoutorderwithequalcol, int4, boolean, int4, internal)
  描述：全密态类型转换解密函数。不支持用户调用。
  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal

版权所有 © 华为云计算技术有限公司

返回值类型：bytea

- ce_decrypt_clob(byteawithoutorderwithequalcol, int4, boolean, int4, internal)

  描述：全密态类型转换解密函数。不支持用户调用。

  参数类型：byteawithoutorderwithequalcol, int4, boolean, int4, internal

  返回值类型：clob

示例

byteawithoutorderwithequalcolin、byteawithoutorderwithequalcolout等密态等值函数为数据库内核中数据类型byteawithoutorderwithequalcol指定的in、out、send、recv等读写格式转换函数，具体可参考bytea类型的byteain、byteaout等函数，但会对本地的cek进行验证，需要密文字段中有本地存在的cekoid才能执行成功。

```
-- 例如使用密态客户端连接数据库,创建加密表int_type，int_col2为其加密列。
gaussdb=# \key_info keyType=user_token,password=*********
gaussdb=# CREATE CLIENT MASTER KEY MyCMK WITH (KEY_STORE = user_token, ALGORITHM =
AES_256_CBC);
gaussdb=# CREATE COLUMN ENCRYPTION KEY cek1 WITH VALUES (CLIENT_MASTER_KEY = MyCMK,
ALGORITHM = AEAD_AES_256_CBC_HMAC_SHA256);
gaussdb=# CREATE TABLE IF NOT EXISTS int_type (
id INT,
int_col1 TINYINT,
int_col2 smallint ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = cek1, ENCRYPTION_TYPE =
DETERMINISTIC)
);
CREATE TABLE
gaussdb=# INSERT INTO int_type VALUES (2, 100, 200);
INSERT 0 1

-- 使用非密态客户端连接数据库，查询加密列密文。
gaussdb=# SELECT int_col2 FROM int_type;
                                                          int_col2
--------------------------------------------------------------------------------------------------------------------------------
-----------------------

\x01c35301bf421c8edf38c34704bcc82838742917778ccb402a1b7452ad4a6ac7371acc0ac33100000035fe3424
919854c86194f1aa5bb4e1ca656e8fc6d05324a1419b69f488bdc3c6
(1 row)

-- 将加密列密文当做byteawithoutorderwithequalcolin入参，格式从cstring输入转码转化成内部
byteawithoutorderwithequalcol形式。
gaussdb=# SELECT
byteawithoutorderwithequalcolin('\x01c35301bf421c8edf38c34704bcc82838742917778ccb402a1b7452ad4a
6ac7371acc0ac33100000035fe3424919854c86194f1aa5bb4e1ca656e8fc6d05324a1419b69f488bdc3c6');
                                                byteawithoutorderwithequalcolin
--------------------------------------------------------------------------------------------------------------------------------
-----------------------

\x01c35301bf421c8edf38c34704bcc82838742917778ccb402a1b7452ad4a6ac7371acc0ac33100000035fe3424
919854c86194f1aa5bb4e1ca656e8fc6d05324a1419b69f488bdc3c6
(1 row)
```

由于byteawithoutorderwithequalcolin等的实现会对cek进行查找，并且判断是否为正常加密后的数据类型。

因此如果用户输入数据的格式不是加密后的数据格式，并且在本地不存在对应cek的情况下，会返回错误。

```
gaussdb=# SELECT * FROM
byteawithoutorderwithequalcolsend('\x907219912381298461289346129'::byteawithoutorderwithequalcol);
ERROR:  cek with OID 596711794 not found
LINE 1: SELECT * FROM byteawithoutorderwithequalcolsend('\x907219912...
                                                         ^
gaussdb=# SELECT * FROM byteawithoutordercolout('\x90721901999999999999912381298461289346129');
```

```
ERROR:  cek with OID 2566986098 not found
LINE 1: SELECT * FROM byteawithoutordercolout('\x9072190199999999999...

                                    ^
gaussdb=# SELECT * FROM
byteawithoutorderwithequalcolsend('\x907219019999999999999912381298461289346129'::byteawithoutorde
rwithequalcol);
ERROR:  cek with OID 2566986098 not found
LINE 1: SELECT * FROM byteawithoutorderwithequalcolsend('\x907219019...
                                              ^
```

1.6.23 返回集合的函数

说明

- 对于任意函数的输入参数为返回集合的函数（set return function，简称srf函数）时，会对任意函数入参列表的各个表达式进行校验，当出现多个srf函数时会报错。

  例如：SELECT div(srf(n,1),srf(m,2));。
- 在SELECT语句中的target list中，当存在srf函数时，会返回多行数据。
- 在SELECT语句中的target list中，当存在多层嵌套场景时，srf(n, srf(m, 1))会返回n*m行的结果。即使存在strict的非srf函数，也不会对strict函数进行优化。

  例如：SELECT srf(n, div(srf(m, 1), NULL));div为strict函数，当其输入参数不包含srf函数时，div函数在输入参数包含NULL的情况下，会返回NULL；而当其输入参数包含srf函数时，不会对strict函数进行优化，返回m行NULL。
- 当srf函数的输入参数包含返回值为NULL的表达式时，对于strict的srf函数，会返回0行；而非strict的srf函数则会返回指定行数的NULL。

  例如：srf(10, 'a')表示返回10行'a'。如果传入srf(10, NULL)，strict的srf函数将返回0行，而非strict的srf函数将返回10行NULL。

序列号生成函数

- generate_series(start, stop)

  描述：生成一个数值序列，从start到stop，步长为1。

  参数类型：int、bigint、numeric

  返回值类型：setof int、setof bigint、setof numeric（与参数类型相同）
- generate_series(start, stop, step)

  描述：生成一个数值序列，从start到stop，步长为step。

  参数类型：int、bigint、numeric

  返回值类型：setof int、setof bigint、setof numeric（与参数类型相同）
- generate_series(start, stop, step interval)

  描述：生成一个数值序列，从start到stop，步长为step。

  参数类型：timestamp或timestamp with time zone

  返回值类型：setof timestamp或setof timestamp with time zone（与参数类型相同）

如果step是正数且start大于stop，则返回零行。相反，如果step是负数且start小于stop，则也返回零行。如果输入是NULL，同样产生零行。如果step为零则是一个错误。

示例：

```
gaussdb=# SELECT * FROM generate_series(2,4);
 generate_series
-----------------
```

 版权所有 © 华为云计算技术有限公司

```
            2
            3
            4
(3 rows)

gaussdb=# SELECT * FROM generate_series(5,1,-2);
 generate_series
-----------------
            5
            3
            1
(3 rows)

gaussdb=# SELECT * FROM generate_series(4,3);
 generate_series
-----------------
(0 rows)

--这个示例应用于date-plus-integer操作符。
gaussdb=# SELECT current_date + s.a AS dates FROM generate_series(0,14,7) AS s(a);
   dates
------------
 2017-06-02
 2017-06-09
 2017-06-16
(3 rows)

gaussdb=# SELECT * FROM generate_series('2008-03-01 00:00'::timestamp, '2008-03-04 12:00', '10 hours');
   generate_series
---------------------
 2008-03-01 00:00:00
 2008-03-01 10:00:00
 2008-03-01 20:00:00
 2008-03-02 06:00:00
 2008-03-02 16:00:00
 2008-03-03 02:00:00
 2008-03-03 12:00:00
 2008-03-03 22:00:00
 2008-03-04 08:00:00
(9 rows)
```

- gs_search_function_with_name(funcname)

  描述：获取函数名为funcname的函数OID。

  参数类型：cstring

  返回值类型：setof oid

  ```
  --基本用法。
  CREATE OR REPLACE FUNCTION proc_plpgsql(a int,b int, c int)
  RETURNS int AS $BODY$
  DECLARE
  BEGIN
  RETURN $1 + $2;
  END;
  $BODY$ language plpgsql;
  CREATE OR REPLACE FUNCTION proc_plpgsql(int,int)
  RETURNS int AS $BODY$
  DECLARE
  BEGIN
  RETURN $1 + $2;
  END;
  $BODY$ language plpgsql;
  gaussdb=# SELECT gs_search_function_with_name('proc_plpgsql');
   gs_search_function_with_name
  ------------------------------
                         16776
                         24576
  (2 rows)
  ```


版权所有 © 华为云计算技术有限公司

下标生成函数

- generate_subscripts(array anyarray, dim int)

  描述：生成一系列包括给定数组的下标。

  返回值类型：setof int

- generate_subscripts(array anyarray, dim int, reverse boolean)

  描述：生成一系列包括给定数组的下标。当reverse为真时，该系列则以相反的顺序返回。

  返回值类型：setof int

generate_subscripts是一个为给定数组中的指定维度生成有效下标集的函数。如果数组中没有所请求的维度或者NULL数组，返回零行（但是会给数组元素为空的返回有效下标）。示例：

```
--基本用法。
gaussdb=# SELECT generate_subscripts('{NULL,1,NULL,2}'::int[], 1) AS s;
 s
---
 1
 2
 3
 4
(4 rows)
--unnest一个2D数组。
gaussdb=# CREATE OR REPLACE FUNCTION unnest2(anyarray)
RETURNS SETOF anyelement AS $$
SELECT $1[i][j]
   FROM generate_subscripts($1,1) g1(i),
        generate_subscripts($1,2) g2(j);
$$ LANGUAGE sql IMMUTABLE;

gaussdb=# SELECT * FROM unnest2(ARRAY[[1,2],[3,4]]);
 unnest2
---------
       1
       2
       3
       4
(4 rows)

--删除函数。
gaussdb=# DROP FUNCTION unnest2;
```

1.6.24 重载查询函数

- gs_search_function_with_name_and_arg(funcname_and_argtypes cstring)

  描述：通过函数名和函数入参类型，返回实际调用的具体的重载函数信息。

  返回值类型：text

  说明

  - 此函数仅在ORA兼容模式下支持。
  - 如需查询入参包含NULL的函数的重载信息，须使用UNKNOWN替代NULL。
  - 当函数入参为窗口函数时，须在最后加上over()。
  - 部分函数通过语法关键字实现，无法通过该函数进行查询，如需了解对应关系，请直接参见**表1-102**。
  - 如果要查找的函数发生变更或者函数选择逻辑发生变更，该函数的结果可能发生变化。函数选择逻辑详见**函数**。

版权所有 © 华为云计算技术有限公司

表 1-102 语法函数内部函数调用关系

语法函数	内部调用函数
CURRENT_CATALOG	current_database()
CURRENT_DATE	优先级从上往下匹配： • 当a_format_date_timestamp=on、mapping_date_to_datea=off时，对应pg_systimestamp()::timestamp(0)。 • 当mapping_date_to_datea=on时，对应pg_systimestamp()::datea。 • 当a_format_version='10c'、a_format_dev_version='s2'以上时，对应'now'::text::timestamp(0)。 • 其他场景对应text_date('now'::text)。
CURRENT_ROLE	current_user()
CURRENT_SCHEMA	current_schema()
CURRENT_TIME[(n)]	'now'::text::timetz[(n)]
CURRENT_TIMESTAMP	• CURRENT_TIMESTAMP，对应pg_systimestamp()。 • CURRENT_TIMESTAMP(int) – 当a_format_date_timestamp=on时，对应pg_systimestamp()::timestamptz(int)。 – 当a_format_date_timestamp=off时，对应'now'::text::timestamptz(int)。 • CURRENT_TIMESTAMP(float)，仅a_format_version='10c'、a_format_dev_version='s2'以上时支持，对应current_timestamp(numeric)。
CURRENT_USER	current_user()
DBTIMEZONE	仅a_format_version='10c'、a_format_dev_version='s2'以上时支持，对应dbtimezone()。
DECODE	• 参数个数等于2时，DECODE(A, B)对应decode(A, B)。 • 参数个数大于2时，构造成case，请参见**UNION，CASE和相关构造**。
EXTRACT (field FROM source)	date_part(field, source)
GROUP_CONCAT(arg1[, …] SEPARATOR s)	group_concat(s, arg1[, …])
LOCALTIME[(n)]	'now'::text::time[(n)]

版权所有 © 华为云计算技术有限公司

语法函数	内部调用函数
LOCALTIMESTAMP	• LOCALTIMESTAMP – 当a_format_date_timestamp=on时，对应pg_systimestamp()::timestamp。 – 当a_format_date_timestamp=off时，对应'now'::text::timestamp。 • LOCALTIMESTAMP(int) – 当a_format_date_timestamp=on时，对应pg_systimestamp()::timestamp(int)。 – 当a_format_date_timestamp=off时，对应'now'::text::timestamp(int)。 • LOCALTIMESTAMP(float)，仅a_format_version='10c'、a_format_dev_version='s2'以上时支持。 – 当a_format_date_timestamp=on时，对应current_timestamp(numeric)::timestamp。 – 当a_format_date_timestamp=off时，无对应关系。
NOW()	now()
OVERLAY	• OVERLAY(A PLACING B FROM C FOR D)，对应overlay(A, B, C, D)。 • OVERLAY(A PLACING B FROM C)，对应overlay(A, B, C)。
POSITION (A IN B)	position(B, A)
SESSION_USER	session_user()
SESSIONTIMEZONE	session_time_zone()
SUBSTR	• SUBSTR(str, pos)，对应substr(str, pos)。 • SUBSTR(str FROM pos)，对应substr(str, pos)。 • SUBSTR(str, pos, len)，对应substr(str, pos, len)。 • SUBSTR(str FROM pos FOR len)，对应substr(str, pos, len)。 • SUBSTR(str FOR len FROM pos)，对应substr(str, pos, len)。 • SUBSTR(str FOR len)，对应substr(str, 1::int4, len::int4)。

版权所有 © 华为云计算技术有限公司

语法函数	内部调用函数
SUBSTRING	• SUBSTRING(str, pos)，对应substring(str, pos)。 • SUBSTRING(str FROM pos)，对应substring(str, pos)。 • SUBSTRING(str, pos, len)，对应substring(str, pos, len)。 • SUBSTRING(str FROM pos FOR len)，对应substring(str, pos, len)。 • SUBSTRING(str FOR len FROM pos)，对应substring(str, pos, len)。 • SUBSTRING(str FOR len)，对应substring(str, 1::int4, len::int4)。
SYSDATE	• 当mapping_date_to_datea=off时，对应current_sysdate()。 • 当mapping_date_to_datea=on时，对应sysdate_a()。
TRIM(BOTH [A | A [FROM]] B)	btrim(B[, A])
TRIM(LEADING [A | A [FROM]] B)	ltrim(B[, A])
TRIM(TRAILING [A | A [FROM]] B)	rtrim(B[, A])
USER	current_user()
XMLEXISTS(A PASSING [BY REF] B [BY REF])	xmlexists(A, B)

说明

对于存在内部映射函数的语法函数，查询结果的投影别名实际显示的是其内部调用的实现函数名称，而非语法关键字本身。

```
--基本用法。
gaussdb=# SELECT gs_search_function_with_name_and_arg('abs(int)');
     gs_search_function_with_name_and_arg
-----------------------------------------------
 pg_catalog.int4 pg_catalog.abs(pg_catalog.int4)
(1 row)

gaussdb=# SELECT gs_search_function_with_name_and_arg('pg_catalog.avg(int16)');
       gs_search_function_with_name_and_arg
-------------------------------------------------------
 pg_catalog.float8 pg_catalog.avg(pg_catalog.float8)
(1 row)

gaussdb=# SELECT gs_search_function_with_name_and_arg('lag(datea, int, unknown) over()');
                            gs_search_function_with_name_and_arg
-------------------------------------------------------------------------------------------------
 pg_catalog.anyelement pg_catalog.lag(pg_catalog.anyelement, pg_catalog.int4,
```


版权所有 © 华为云计算技术有限公司

```
pg_catalog.anyelement)
(1 row)
```

- gs_search_operator_with_name_and_arg(ltype cstring, opname cstring, rtype cstring)

  描述：通过操作符的名称和左右操作数的数据类型，返回实际调用的具体的重载操作符信息。

  返回值类型：text

  说明

  - 此函数仅在ORA兼容模式下支持。
  - 仅支持查找二元操作符。
  - 如需查询入参包含NULL的操作符的重载信息，须使用UNKNOWN替代NULL。
  - 如需查询!=或^=操作符的重载信息，须使用<>进行替代。
  - 如果要查找的操作符发生变更或者操作符选择逻辑发生变更，该函数的结果可能发生变化。操作符选择逻辑详见**操作符**。

```
--基本用法。
gaussdb=# SELECT gs_search_operator_with_name_and_arg('int', '+', 'int');
             gs_search_operator_with_name_and_arg
-----------------------------------------------------------------
 pg_catalog.int4 (pg_catalog.int4 pg_catalog.+ pg_catalog.int4)
(1 row)

gaussdb=# SELECT gs_search_operator_with_name_and_arg('int', '+', 'float4');
                gs_search_operator_with_name_and_arg
-------------------------------------------------------------------
 pg_catalog.float8 (pg_catalog.float8 pg_catalog.+ pg_catalog.float4)
(1 row)
```

1.6.25 条件表达式函数

条件表达式函数

- coalesce(expr1, expr2, ..., exprn)

  描述：返回参数列表中第一个非NULL的参数值。该函数等价于CASE WHEN expr1 IS NOT NULL THEN expr1 ELSE expr2 END。

  示例：

```
gaussdb=# SELECT coalesce(NULL,'hello');
 coalesce
----------
 hello
(1 row)
```

  备注：

  - 如果表达式列表中的所有表达式都等于NULL，则本函数返回NULL。
  - 它常用于在显示数据时用缺省值替换NULL。
  - 和CASE表达式一样，COALESCE不会计算不需要用来判断结果的参数；即在第一个非空参数右边的参数不会被计算。

- decode(base_expr, compare1, value1, compare2,value2, … default)

  描述：把base_expr与后面的每个compare(n) 进行比较，如果匹配返回相应的value(n)。如果没有发生匹配，则返回default。

  示例：

```
gaussdb=# SELECT decode('A','A',1,'B',2,0);
 case
```


版权所有 © 华为云计算技术有限公司

```
------
 1
(1 row)
```

备注：不支持对xml数据类型的操作。

- nullif(expr1, expr2)

  描述：当且仅当expr1和expr2相等时，NULLIF才返回NULL，否则它返回expr1。该函数逻辑上等价于CASE WHEN expr1 = expr2 THEN NULL ELSE expr1 END。

  说明

  nullif(expr1, expr2)函数为映射函数，故pg_proc系统表中无法查到对应函数定义。

  示例：

  ```
  gaussdb=# SELECT nullif('hello','world');
   nullif
  --------
   hello
  (1 row)
  ```

  备注：不支持对xml数据类型的操作。

  如果两个参数的数据类型不同，则：

  – 若两种数据类型之间存在隐式转换，则以其中优先级较高的数据类型为基准将另一个参数隐式转换成该类型，转换成功则进行计算，转换失败则返回错误。如：

    ```
    gaussdb=# SELECT nullif('1234'::VARCHAR,123::INT4);
     nullif
    --------
       1234
    (1 row)
    gaussdb=# SELECT nullif('1234'::VARCHAR,'2012-12-24'::DATE);
    ERROR:  invalid input syntax for type timestamp: "1234"
    ```

  – 若两种数据类型之间不存在隐式转换，则返回错误。如：

    ```
    gaussdb=# SELECT nullif(1::bit, '1'::MONEY);
    ERROR:  operator does not exist: bit = money
    LINE 1: SELECT nullif(1::bit, '1'::MONEY);
                  ^
    HINT:  No operator matches the given name and argument type(s). You might need to add explicit type casts.
    CONTEXT:  referenced column: nullif
    ```

- nvl(expr1 , expr2)

  描述：

  – 如果expr1为NULL，则返回expr2。

  – 如果expr1非NULL，则返回expr1。

  示例：

  ```
  gaussdb=# SELECT nvl('hello','world');
    nvl
  -------
   hello
  (1 row)
  ```

  备注：参数expr1和expr2可以为任意类型，当NVL的两个参数不属于同类型时，看第二个参数是否可以向第一个参数进行隐式转换，如果可以则返回第一个参数类型。如果第二个参数不能向第一个参数进行隐式转换而第一个参数可以向第二个参数进行隐式转换，则返回第二个参数的类型。如果两个参数之间不存在隐式类型转换并且也不属于同一类型则报错。

- nvl2(expr1 , expr2，expr3)

  描述：

版权所有 © 华为云计算技术有限公司

- 如果expr1为NULL，则返回expr3。
- 如果expr1非NULL，则返回expr2。

说明

此函数在参数ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下有效。

示例：

```
gaussdb=# SELECT nvl2('hello','world','other');
 case
-------
 world
(1 row)
```

备注：参数expr2和expr3可以为任意类型， 当NVL2的后面两个参数不属于同类型时，看expr3参数是否可以向expr2参数进行隐式转换，如果不能隐式转换，会返回错误。如果第一个参数是数值类型，函数将第一个参数和其他参数都转换为numeric类型，然后进行比较，对于不能转换的，提示出错信息；第一个参数是其他类型的，函数将其他参数都转换为第一个参数的类型进行比较，对于不能转换的，提示出错信息。

- greatest(expr1 [, ...])

  描述：获取并返回参数列表中值最大的表达式的值。

  示例：

```
gaussdb=# SELECT greatest(1*2,2-3,4-1);
 greatest
----------
        3
(1 row)
gaussdb=# SELECT greatest('HARRY', 'HARRIOT', 'HAROLD');
 greatest
----------
 HARRY
(1 row)
```

  备注：不支持对xml数据类型的操作。

  说明

  此函数在参数ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下：

  1. 如果参数中有任意一个参数的值为null，函数返回null。
  2. 如果第一个参数是数值类型，函数将第一个参数和其他参数都转换为numeric类型，然后进行比较，对于不能转换的，提示出错信息；第一个参数是其他类型的，函数将其他参数都转换为第一个参数的类型进行比较，对于不能转换的，提示出错信息。

- least(expr1 [, ...])

  描述：获取并返回参数列表中值最小的表达式的值。

  示例：

```
gaussdb=# SELECT least(1*2,2-3,4-1);
 least
-------
    -1
(1 row)
gaussdb=# SELECT least('HARRY','HARRIOT','HAROLD');
 least
--------
 HAROLD
(1 row)
```

  备注：不支持对xml数据类型的操作。

版权所有 © 华为云计算技术有限公司

说明

此函数在ORA兼容模式下，且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：

1. 如果参数中有任意一个参数的值为null，函数返回null。
2. 如果第一个参数是数值类型，函数将第一个参数和其他参数都转换为numeric类型，然后进行比较，对于不能转换的，提示出错信息；第一个参数是其他类型的，函数将其他参数都转换为第一个参数的类型进行比较，对于不能转换的，提示出错信息。

- EMPTY_BLOB()

  描述：使用EMPTY_BLOB在INSERT或UPDATE语句中初始化一个BLOB变量，取值为NULL。

  返回值类型：BLOB

  示例：

  ```
  --新建表
  gaussdb=# CREATE TABLE blob_tb(b blob,id int) DISTRIBUTE BY REPLICATION;
  --插入数据
  gaussdb=# INSERT INTO blob_tb VALUES (empty_blob(),1);
  --删除表
  gaussdb=# DROP TABLE blob_tb;
  ```

  备注：使用DBE_LOB.GET_LENGTH求得的长度为0。

- EMPTY_CLOB()

  描述：使用EMPTY_CLOB在INSERT或UPDATE语句中初始化一个CLOB变量，取值为空。

  说明

  此函数在ORA兼容模式下，且参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。

  返回值类型：CLOB

  示例：

  ```
  --新建表
  gaussdb=# CREATE TABLE clob_tb(c clob,id int);
  --插入数据
  gaussdb=# INSERT INTO clob_tb VALUES (empty_clob(),1);
  --删除表
  gaussdb=# DROP TABLE clob_tb;
  ```

  备注：使用DBE_LOB.GET_LENGTH求得的长度为0。

- lnnvl(condition)

  描述：lnnvl用于某个查询语句的WHERE子句中，如果条件为true就返回false，如果条件为unknown或者false，就返回true。

  condition：必须为逻辑表达式。但不能用于复合条件如AND、OR或者BETWEEN。

  返回类型：Boolean

  示例：

  ```
  --新建表
  gaussdb=# CREATE TABLE student_demo (name VARCHAR2(20), grade  NUMBER(10,2));
  CREATE TABLE

  --插入数据
  gaussdb=# INSERT INTO student_demo VALUES ('name0',0);
  INSERT 0 1
  gaussdb=# INSERT INTO student_demo VALUES ('name1',1);
  INSERT 0 1
  gaussdb=# INSERT INTO student_demo VALUES ('name2',2);
  ```


版权所有 © 华为云计算技术有限公司

```
INSERT 0 1

--调用lnnvl
gaussdb=# SELECT * FROM student_demo WHERE LNNVL(name = 'name1');
 name  | grade
-------+-------
 name0 |  0.00
 name2 |  2.00
(2 rows)

--删除表
gaussdb=# drop table student_demo;
DROP TABLE
```

说明

此函数在ORA兼容模式下，且参数a_format_version值为10c和a_format_dev_version值为s2的情况下，才支持lnnvl函数。

- isnull(expr)

  描述：判断expr是否为NULL，如果expr为NULL，返回true，否则返回false。isnull函数在全模式下兼容。

  参数：支持任意类型输入。

  返回类型：Boolean

  示例：

```
gaussdb=# SELECT isnull(null);
 isnull
--------
 t
(1 row)

gaussdb=# SELECT isnull(1);
 isnull
--------
 f
(1 row)
```

- if(expr1, expr2, expr3)

  描述：判断expr1的值，当expr1为true时返回expr2的值，否则返回expr3的值。该函数逻辑上等价于CASE WHEN expr1 THEN expr2 ELSE expr3 END。

  参数：如**表1-103**所示。

  表 1-103 参数类型说明

参数名	合法入参类型	描述
expr1	boolean	用于判断if()函数选择返回值。
expr2	任意类型	当expr1为true时，返回值为expr2。
expr3	任意类型	当expr1为false时，返回值为expr3。

  返回值类型：与入参类型有关，输出规则见说明。

版权所有 © 华为云计算技术有限公司

说明

- if(expr1, expr2, expr3)仅在MYSQL兼容模式下支持。
- 当设置兼容性参数b_format_version=''和b_format_dev_version=''时，if(expr1, expr2, expr3)函数输出结果类型与CASE WHEN expr1 THEN expr2 ELSE expr3 END输出结果类型一致；当设置兼容性参数b_format_version='5.7'和b_format_dev_version='s1'后返回值类型推导遵从以下规则，规则优先级由高到低排列：
  - 两入参类型相同时，返回值类型与入参类型相同；
  - 其中一个入参为SET类型时，返回TEXT类型；
  - 其中一个入参为BLOB类型时，返回BLOB类型；
  - 其中一个入参为TEXT类型时，返回TEXT类型；
  - 其中一个入参为STRING类型时，返回TEXT类型；
  - 其中一个入参为时间类型时，另一个也为时间类型则返回TIMESTAMPTZ，否则返回TEXT类型；
  - 两入参均为有符号整型时，若其中一个入参为BIGINT类型则返回BIGINT类型，否则返回INT类型；
  - 两入参均为无符号整型时，若其中一个入参为UNSIGNED BIGINT类型则返回UNSIGNED BIGINT类型，否则返回UNSIGNED INT类型；
  - 两入参分别为浮点型与整型时，返回浮点型；
  - 除以上提到的NUMERIC类型组合外，其余NUMERIC类型组合均返回NUMERIC类型；
  - 若入参类型与返回值类型不存在隐式转换函数，则报错。
- expr1入参为NULL时返回结果与expr1为FALSE场景返回结果一致。
- if(expr1, expr2, expr3)函数为映射函数，故pg_proc系统表中无法查到对应函数定义。

示例：

```
--expr1为表达式。
gaussdb=# select if(2>3, 'true', 'false');
 case
-------
 false
(1 row)

--expr1入参为NULL时，与false场景结果一致，返回expr3的值。
gaussdb=# select if(null, 'not null', 'is null');
  case
---------
 is null
(1 row)
```

- ifnull(expr1, expr2)

  描述：若expr1为NULL返回expr2的值，否则返回expr1的值。该函数逻辑上等价于nvl(expr1, expr2)。

  参数：任意类型。

  返回值类型：与入参类型有关，输出规则见说明。

版权所有 © 华为云计算技术有限公司

说明

- ifnull(expr1, expr2)仅在MYSQL兼容模式下支持。
- 当设置兼容性参数b_format_version=''和b_format_dev_version=''时，ifnull(expr1, expr2)函数输出结果类型与nvl(expr1, expr2)输出结果类型一致；当设置兼容性参数b_format_version='5.7'和b_format_dev_version='s1'后返回值类型推导遵从以下规则，规则优先级由高到低排列：
  - 两入参类型相同时，返回值类型与入参类型相同；
  - 其中一个入参为SET类型时，返回TEXT类型；
  - 其中一个入参为BLOB类型时，返回BLOB类型；
  - 其中一个入参为TEXT类型时，返回TEXT类型；
  - 其中一个入参为STRING类型时，返回TEXT类型；
  - 其中一个入参为时间类型时，另一个也为时间类型则返回TIMESTAMPTZ，否则返回TEXT类型；
  - 两入参类型其中一个为float4类型另一个为NUMERIC范畴中任一类型，返回值为double类型。
  - 两入参均为NUMERIC类型时，返回两个入参类型中精度较高的类型，例如入参类型分别为TINYINT和INT时，返回值类型推导为INT型；
  - 若入参类型与返回值类型不存在隐式转换函数，则报错。
- 两入参同时为NULL时，返回值为NULL，否则返回第一个不为空的值。
- ifnull(expr1, expr2)函数为映射函数，故pg_proc系统表中无法查到对应函数定义。

示例：

```
--入参为空串，输出为空串而非NULL。
gaussdb=# SELECT ifnull('', null) is null as a;
 a
---
 f
(1 row)

--两个入参都为NULL，输出为NULL。
gaussdb=# SELECT ifnull(null, null) is null as a;
 a
---
 t
(1 row)

--输入NULL和字符串，输出结果为第一个非空的值。
gaussdb=# SELECT ifnull(null, 'A') as a;
 a
---
 A
(1 row)
```

1.6.26 系统信息函数

会话信息函数

- SYS_CONTEXT()

  描述：返回当前时刻与上下文命名空间'namespace'关联的参数'parameter'的值，该函数功能在第一个入参为'userenv'时与**USERENV()**功能一致。

  当前支持的第二个入参参数：

  'current_edition_name'、'current_schema'、'current_sql'、'current_user'、'current_userid'、'database_role'、'ip_address'、'is_apply_server'、'isdba'、'nls_currency'、'nls_date_format'、'nls_date_language'、'server_host'、'service_name'、'session_user'、'session_userid'、'db_name'、

版权所有 © 华为云计算技术有限公司

'db_unique_name'、'instance'、'instance_name'、'current_edition_id'、'current_schemaid'、'lang'、'language'、'sid'

返回值类型：text

示例：

```
gaussdb=# SELECT SYS_CONTEXT('userenv','NLS_CURRENCY');
 sys_context
-------------
 $
(1 row)

gaussdb=# SELECT SYS_CONTEXT('userenv','NLS_DATE_FORMAT');
  sys_context
---------------
 ISO, MDY
(1 row)

gaussdb=# SELECT SYS_CONTEXT('userenv','NLS_DATE_LANGUAGE');
 sys_context
-------------
 en_US.UTF-8
(1 row)
```

- USERENV()

  描述：返回当前时刻与上下文命名空间'namespace'关联的参数'parameter'的值，该函数功能与**SYS_CONTEXT()**在第一个入参为'userenv'时功能一致。

  当前支持的入参：

  'current_edition_name'、'current_schema'、'current_sql'、'current_user'、'current_userid'、'database_role'、'ip_address'、'is_apply_server'、'isdba'、'nls_currency'、'nls_date_format'、'nls_date_language'、'server_host'、'service_name'、'session_user'、'session_userid'、'db_name'、'db_unique_name'、'instance'、'instance_name'、'current_edition_id'、'current_schemaid'、'lang'、'language'、'sid'

  当前不支持的入参：

  'action'、'is_application_root'、'is_application_pdb'、'audited_cursorid'、'authenticated_identity'、'authentication_data'、'authentication_method'、'cdb_domain'、'cdb_name'、'client_identifier'、'con_id'、'con_name'、'current_sql_length'、'db_domain'、'db_supplemental_log_level'、'dblink_info'、'drain_status'、'entryid'、'enterprise_identity'、'fg_job_id'、'global_uid'、'identification_type'、'is_dg_rolling_upgrade'、'ldap_server_type'、'module'、'network_protocol'、'nls_calendar'、'nls_sort'、'nls_territory'、'oracle_home'、'os_user'、'platform_slash'、'policy_invoker'、'proxy_enterprise_identity'、'proxy_user'、'proxy_userid'、'scheduler_job'、'session_edition_id'、'session_edition_name'、'sessionid'、'statementid'、'terminal'、'unified_audit_sessionid'、'session_default_collation'、'client_info'、'bg_job_id'、'client_program_name'、'current_bind'、'global_context_memory'、'host'、'current_sqln'

  返回值类型：text

  示例：

```
gaussdb=# SELECT USERENV('sid');
 userenv
---------
 955
(1 row)

gaussdb=# SELECT USERENV('instance');
 userenv
---------
```


版权所有 © 华为云计算技术有限公司

```
 14524
(1 row)

gaussdb=# SELECT USERENV('action');
 userenv
---------

(1 row)

gaussdb=# SELECT USERENV('actijj');
ERROR:  The configuration parameter actijj cannot be identified.
CONTEXT:  PL/pgSQL function userenv(text) line 114 at RETURN
referenced column: userenv
```

说明

当入参为当前不支持的入参时，则返回NULL；入参为非法入参时报错。

- current_catalog()

  描述：当前数据库的名称（在标准SQL中称"catalog"）。

  返回值类型：name

  示例：

```
testdb=# SELECT current_catalog;
 current_database
------------------
 testdb
(1 row)
```

- current_database()

  描述：当前数据库的名称。

  返回值类型：name

  示例：

```
testdb=# SELECT current_database();
 current_database
------------------
 testdb
(1 row)
```

- current_query()

  描述：由客户端提交的当前执行语句（可能包含多个声明）。

  返回值类型：text

  示例：

```
gaussdb=# SELECT current_query();
     current_query
--------------------------
 SELECT current_query();
(1 row)
```

- current_schema[()]

  描述：当前模式的名称。

  返回值类型：name

  示例：

```
gaussdb=# SELECT current_schema();
 current_schema
----------------
 public
(1 row)
```


版权所有 © 华为云计算技术有限公司

备注：current_schema返回在搜索路径中第一个顺位有效的模式名。（如果搜索路径为空则返回NULL，没有有效的模式名也返回NULL）。如果创建表或者其他命名对象时没有声明目标模式，则将使用这些对象的模式。

- current_schemas(Boolean)

  描述：搜索路径中的模式名称。

  返回值类型：name[]

  示例：

```
gaussdb=# SELECT current_schemas(true);
  current_schemas
--------------------
 {pg_catalog,public}
(1 row)
```

  备注：

  current_schemas(Boolean)返回搜索路径中所有模式名称的数组。布尔选项决定像pg_catalog这样隐含包含的系统模式是否包含在返回的搜索路径中。

  说明

  搜索路径可以通过运行时设置更改。命令是：

```
SET search_path TO schema [, schema, ...]
```

- database()

  描述：返回当前的Schema的名称。

  返回值类型：name

  示例：

```
gaussdb=# SELECT database();
 database
----------
 public
(1 row)
```

  说明

  此函数在MYSQL模式数据库中，且GUC参数b_format_version = '5.7'和b_format_dev_version = 's1'时生效。

- current_user()

  描述：当前执行环境下的用户名。

  返回值类型：name

  示例：

```
gaussdb=# SELECT current_user;
 current_user
--------------
 omm
(1 row)
```

  备注：current_user用于权限检查的用户标识。通常表示会话用户，可以通过**SET ROLE**进行修改。在函数执行的过程中，受SECURITY DEFINER属性影响。

- definer_current_user()

  描述：当前执行环境下的用户名。

  返回值类型：name

  示例：

```
gaussdb=# SELECT definer_current_user();
 definer_current_user
----------------------
```


版权所有 © 华为云计算技术有限公司

```
 omm
(1 row)
```

备注：通常definer_current_user和current_user结果相同，但在存储过程中执行该函数会返回定义当前存储过程的用户名。

- pg_current_sessionid()

  描述：当前执行环境下的会话ID。

  返回值类型：text

  示例：

```
gaussdb=# SELECT pg_current_sessionid();
   pg_current_sessionid
----------------------------
 1579228402.140190434944768
(1 row)
```

  备注：pg_current_sessionid()是用于获取当前执行环境下的会话ID。其组成结构为：时间戳.会话ID，当线程池模式开启（enable_thread_pool=on）时，会话ID为SessionID；而线程池模式关闭时，会话ID实际为线程ID。

- pg_current_sessid()

  描述：当前执行环境下的会话ID。

  返回值类型：text

  示例：

```
gaussdb=# select pg_current_sessid();
pg_current_sessid
-------------------
140308875015936
(1 row)
```

  备注：在线程池模式下获得当前会话的会话ID，非线程池模式下获得当前会话对应的后台线程ID。

- pg_current_userid()

  描述：当前用户ID。

  返回值类型：text

  示例：

```
gaussdb=# SELECT pg_current_userid();
pg_current_userid
-------------------
10
(1 row)
```

- tablespace_oid_name(oid)

  描述：根据表空间oid，查找表空间名称。

  返回值类型：text

  示例：

```
gaussdb=# select tablespace_oid_name(1663);
 tablespace_oid_name
---------------------
 pg_default
(1 row)
```

- inet_client_addr()

  描述：连接的远端地址。inet_client_addr返回当前客户端的IP地址。

  说明

  此函数只有在远程连接模式下有效。

版权所有 © 华为云计算技术有限公司

返回值类型：inet

示例：

```
gaussdb=# SELECT inet_client_addr();
 inet_client_addr
------------------
 10.10.0.50
(1 row)
```

- inet_client_port()

  描述：连接的远端端口。inet_client_port返回当前客户端的端口号。

  说明

  此函数只有在远程连接模式下有效。

  返回值类型：int

  示例：

```
gaussdb=# SELECT inet_client_port();
 inet_client_port
------------------
            33143
(1 row)
```

- inet_server_addr()

  描述：连接的本地地址。inet_server_addr返回服务器接收当前连接用的IP地址。

  说明

  此函数只有在远程连接模式下有效。

  返回值类型：inet

  示例：

```
gaussdb=# SELECT inet_server_addr();
 inet_server_addr
------------------
 10.10.0.13
(1 row)
```

- inet_server_port()

  描述：连接的本地端口。inet_server_port返回接收当前连接的端口号。如果是通过Unix-domain socket连接的，则所有这些函数都返回NULL。

  说明

  此函数只有在远程连接模式下有效。

  返回值类型：int

  示例：

```
gaussdb=# SELECT inet_server_port();
 inet_server_port
------------------
            8000
(1 row)
```

- pg_backend_pid()

  描述：当前会话连接的服务线程的线程ID。

  返回值类型：int

  示例：

```
gaussdb=# SELECT pg_backend_pid();
 pg_backend_pid
----------------
```


版权所有 © 华为云计算技术有限公司

```
 140229352617744
(1 row)
```

- pg_conf_load_time()

  描述：配置加载时间。pg_conf_load_time返回最后加载服务器配置文件的时间戳。

  返回值类型：timestamp with time zone

  示例：

```
gaussdb=# SELECT pg_conf_load_time();
     pg_conf_load_time
-------------------------------
 2017-09-01 16:05:23.89868+08
(1 row)
```

- pg_my_temp_schema()

  描述：会话的临时模式的OID，不存在则为0。

  返回值类型：oid

  示例：

```
gaussdb=# SELECT pg_my_temp_schema();
 pg_my_temp_schema
-------------------
                 0
(1 row)
```

  备注：pg_my_temp_schema返回当前会话中临时模式的OID，如果不存在（没有创建临时表）的话则返回0。如果给定的OID是其它会话中临时模式的OID，pg_is_other_temp_schema则返回true。

- pg_is_other_temp_schema(oid)

  描述：是否为另一个会话的临时模式。

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT pg_is_other_temp_schema(25356);
 pg_is_other_temp_schema
-------------------------
 f
(1 row)
```

- pg_listening_channels()

  描述：会话正在侦听的信道名称。

  返回值类型：setof text

  示例：

```
gaussdb=# SELECT pg_listening_channels();
 pg_listening_channels
-----------------------
(0 rows)
```

  备注：pg_listening_channels返回当前会话正在侦听的一组信道名称。

- pg_postmaster_start_time()

  描述：服务器启动时间。pg_postmaster_start_time返回服务器启动时的timestamp with time zone。

  返回值类型：timestamp with time zone

  示例：

```
gaussdb=# SELECT pg_postmaster_start_time();
   pg_postmaster_start_time
-------------------------------
```


版权所有 © 华为云计算技术有限公司

```
 2017-08-30 16:02:54.99854+08
(1 row)
```

- sessionid2pid()

  描述：从sessionid中得到pid信息（如: pv_session_stat中sessid列）。

  返回值类型：int8

  示例：

  ```
  gaussdb=# select sessionid2pid(sessid::cstring) from pv_session_stat limit 2;
    sessionid2pid
  -----------------
   139973107902208
   139973107902208
  (2 rows)
  ```

- session_context('namespace' , 'parameter')

  描述：获取并返回指定namespace下参数parameter的值。

  返回值类型：VARCHAR

  示例：

  ```
  gaussdb=# SELECT session_context('USERENV', 'CURRENT_SCHEMA');
   session_context
  -----------------
   public
  (1 row)
  ```

  根据当前所在的实际schema而变化。

  备注：目前仅支持SESSION_CONTEXT('USERENV', 'CURRENT_SCHEMA') 和SESSION_CONTEXT('USERENV', 'CURRENT_USER')两种格式。

- pg_trigger_depth()

  描述：触发器的嵌套层次。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT pg_trigger_depth();
   pg_trigger_depth
  ------------------
                 0
  (1 row)
  ```

- opengauss_version()

  描述：引用的openGauss内核版本信息。

  返回值类型：text

  使用示例如下，查询结果中的*x.x.x*请以实际输出为准：

  ```
  gaussdb=# SELECT opengauss_version();
   opengauss_version
  -------------------
   x.x.x
  (1 row)
  ```

- gs_deployment()

  描述：当前系统的部署形态信息，对于分布式系统来说返回的是“Distribute”。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT gs_deployment();
   gs_deployment
  ---------------
   Distribute
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- session_user

  描述：会话用户名。

  返回值类型：name

  示例：

  ```
  gaussdb=# SELECT session_user;
   session_user
  --------------
   omm
  (1 row)
  ```

  备注：session_user通常是连接当前数据库的初始用户，不过系统管理员可以用**SET SESSION AUTHORIZATION**修改这个设置。

- user

  描述：等价于current_user。

  返回值类型：name

  示例：

  ```
  gaussdb=# SELECT user;
   current_user
  --------------
   omm
  (1 row)

  -- 创建ORA兼容模式数据库，设置GUC参数a_format_func_col_name='user'。
  gaussdb=# SET a_format_func_col_name = 'user';
  SET

  gaussdb=# SELECT user;
   user
  ------
   omm
  (1 row)
  ```

  说明

  在ORA兼容模式下，设置GUC参数a_format_func_col_name='user'，user函数返回的投影列名为user。

- get_shard_oids_byname()

  描述：输入node的名字返回node的oid。

  返回值类型：oid

  示例：

  ```
  gaussdb=# SELECT get_shard_oids_byname('datanode1');
   get_shard_oids_byname
  -----------------------
   {16385}
  (1 row)
  ```

- getpgusername()

  描述：获取数据库用户名。

  返回值类型：name

  示例：

  ```
  gaussdb=# SELECT getpgusername();
   getpgusername
  ---------------
   GaussDB_userna
  (1 row)
  ```

- getdatabaseencoding()

版权所有 © 华为云计算技术有限公司

描述：获取数据库编码方式。

返回值类型：name

示例：

```
gaussdb=# SELECT getdatabaseencoding();
 getdatabaseencoding
---------------------
 SQL_ASCII
(1 row)
```

- version()

  描述：版本信息。version返回一个描述服务器版本信息的字符串。

  返回值类型：text

  示例：

```
gaussdb=# SELECT version();
                                                                                    version
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 gaussdb (GaussDB XXX.XXX.XXX build f521c606) compiled at 2021-09-16 14:55:22 last mr 6385 
release
(1 row)
```

- working_version_num()

  描述：版本序号信息。返回一个系统兼容性有关的版本序号。

  返回值类型：int

  示例：

```
gaussdb=# SELECT working_version_num();
 working_version_num
---------------------
               92231
(1 row)
```

- get_hostname()

  描述：返回当前节点的hostname。

  返回值类型：text

  示例：

```
gaussdb=# SELECT get_hostname();
 get_hostname
--------------
 linux-user
(1 row)
```

- get_nodename()

  描述：返回当前节点的名字。

  返回值类型：text

  示例：

```
gaussdb=# SELECT get_nodename();
 get_nodename
--------------
 coordinator1
(1 row)
```

- get_nodeinfo(text)

  描述：根据检索的属性返回对应的节点信息的值，可检索属性目前包括node_name和node_type。

  返回值类型：text

  示例：

```
gaussdb=# SELECT get_nodeinfo('node_type');
 get_nodeinfo
--------------
 DIS_CN
(1 row)
gaussdb=# SELECT get_nodeinfo('node_name');
 get_nodeinfo
--------------
 coordinator1
(1 row)
```

- get_schema_oid(cstring)

  描述：返回查询Schema的oid。

  返回值类型：oid

  示例：

  ```
  gaussdb=# SELECT get_schema_oid('public');
   get_schema_oid
  ----------------
            2200
  (1 row)
  ```

- pgxc_parse_clog(OUT xid int8, OUT nodename text, OUT status text)

  描述：返回当前集群中普通表所有事务的状态。

  返回值类型：set of record

  示例：

  ```
  gaussdb=# SELECT pgxc_parse_clog();
   pgxc_parse_clog
  ----------------
  (0,dn_6004_6005_6006,INPROGRESS)
  (1,dn_6004_6005_6006,COMMITTED)
  (2,dn_6004_6005_6006,INPROGRESS)
  (3 row)
  ```

- pgxc_parse_clog(IN int2 bucketid, OUT xid int8, OUT nodename text, OUT status text)

  描述：返回当前集群中指定bucketid的所有事务的状态，入参是hash bucket表bucketid，取值范围[-1,1023]，普通表取值-1。

  返回值类型：set of record

  示例：

  ```
  gaussdb=# SELECT pgxc_parse_clog('-1');
   pgxc_parse_clog
  ----------------
  (0,dn_6004_6005_6006,INPROGRESS)
  (1,dn_6004_6005_6006,COMMITTED)
  (2,dn_6004_6005_6006,INPROGRESS)
  (3 row)
  ```

- pgxc_prepared_xact()

  描述：返回集群中处于prepared阶段事务GID列表。

  返回值类型：set of text

  示例：

  ```
  gaussdb=# SELECT pgxc_prepared_xact();
   pgxc_prepared_xact
  --------------------
  (0 row)
  ```

- pgxc_xacts_iscommitted()

  描述：返回集群中指定事务xid的事务的状态。t代表committed；f代表aborted；null代表others。需要sysadmin或者monadmin权限执行。

版权所有 © 华为云计算技术有限公司

返回值类型：set of record

示例：

```
gaussdb=# SELECT pgxc_xacts_iscommitted(1);
 pgxc_xacts_iscommitted
--------------------
(dn_6004_6005_6006,t)
(cn_5001,t)
(cn_5002,t)
(dn_6001_6002_6003,t)
(4 row)
```

- pgxc_total_memory_detail()

  描述：显示集群内存使用情况。需要sysadmin或者monadmin权限执行。

  说明

  若GUC参数enable_memory_limit=off，该函数不能使用。

  返回值类型：set of pv_total_memory_detail

  示例：

```
gaussdb=# SELECT pgxc_total_memory_detail();
pgxc_total_memory_detail
--------------------
(dn_6004_6005_6006,max_process_memory,81920)
(dn_6004_6005_6006,process_used_memory,72747)
(dn_6004_6005_6006,max_dynamic_memory,12096)
(dn_6004_6005_6006,dynamic_used_memory,1530)
(4 row)
```

- pv_total_memory_detail()

  描述：统计当前数据库节点使用内存的信息，单位为MB。

  说明

  若GUC参数**enable_memory_limit**=off，该函数不能使用。

  返回值类型：record

表 1-104 返回值说明

名称	类型	描述
nodename	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memorytype	text	内存类型，包括以下几种： ● max_process_memory：GaussDB集群实例所占用的内存大小。 ● process_used_memory：GaussDB线程所使用的内存大小。 ● max_dynamic_memory：最大动态内存。 ● dynamic_used_memory：已使用的动态内存。 ● dynamic_peak_memory：内存的动态峰值。 ● dynamic_used_shrctx：最大动态共享内存上下文。 ● dynamic_peak_shrctx：共享内存上下文的动态峰值。 ● max_shared_memory：最大共享内存。 ● shared_used_memory：已使用的共享内存。 ● high_backend_thread_memory：高优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● high_backend_thread_max_memory：高优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_memory：中优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_max_memory：中优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_memory ： 低优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_max_memory ： 低优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● max_sctpcomm_memory：通信库所允许使用的最大内存。 ● sctpcomm_used_memory：通信库已使用的内存大小。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
		• sctpcomm_peak_memory：通信库的内存峰值。 • other_used_memory：其他已使用的内存大小。
memorymbytes	integer	内存类型分配内存的大小。

- get_client_info()
  描述：返回客户端信息。
  返回值类型：record
- pv_session_memory_detail()
  描述：返回会话的内存使用情况，以MemoryContext节点来统计。
  返回值类型：record

表 1-105 返回值说明

名称	类型	描述
sessid	text	• 关闭线程池（enable_thread_pool = off）时该字段表示线程启动时间+session标识（字符串信息为timestamp.sessionid）。 • 开启线程池（enable_thread_pool = on）时，内存上下文是线程级别的，则对应的该字段表示线程启动时间+线程标识（字符串信息为timestamp.threadid），内存上下文是session级别的，则对应的该字段表示线程启动时间+session标识（字符串信息为timestamp.sessionid）。
threadid	bigint	会话绑定的线程标识。
contextname	text	内存上下文名称。
level	smallint	当前上下文在整体内存上下文中的层级。
parent	text	父内存上下文名称。
totalsize	bigint	当前内存上下文的内存总数，单位Byte。
freesize	bigint	当前内存上下文中已释放的内存总数，单位Byte。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
usedsize	bigint	当前内存上下文中已使用的内存总数，单位Byte；“TempSmallContextGroup”内存上下文中该字段含义为统计计数。

访问权限查询函数

DDL类权限ALTER、DROP、COMMENT、INDEX、VACUUM属于所有者固有的权限，隐式拥有。

说明

对于上述隐式拥有的权限，所有者以及继承所有者权限的用户无法直接通过以下访问权限查询函数获取，但实际所有者和继承所有者权限的用户有该权限，执行无影响。如希望通过以下查询函数获取权限，需要通过GRANT显示授权后查询。

以下访问权限查询函数仅表示用户是否具有某对象上的某种对象权限，即返回记录在系统表acl字段中的对象权限拥有情况。

- has_any_column_privilege(user, table, privilege)
  描述：指定用户是否有访问表任何列的权限。

  表 1-106 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或id。
table	text，oid	表	表名称或表id。
privilege	text	权限	• SELECT：允许对指定表任何列执行SELECT语句。 • INSERT：允许对指定表任何列执行INSERT语句。 • UPDATE：允许对指定表任何列任意字段执行UPDATE语句。 • REFERENCES：允许创建一个外键约束（分布式场景暂不支持）。 • COMMENT：允许对指定表任何列执行COMMENT语句。

  返回类型：Boolean

  示例：

```
gaussdb=# CREATE USER test_user PASSWORD '********';
gaussdb=# CREATE TABLE test_table(col int);
gaussdb=# SELECT has_any_column_privilege('test_user','test_table','INSERT');
has_any_column_privilege
```


版权所有 © 华为云计算技术有限公司

```
--------------------------
f
(1 row)

gaussdb=# DROP USER test_user;
```

- has_any_column_privilege(table, privilege)

  描述：当前用户是否有访问表任何列的权限，合法参数类型见**表1-106**。

  返回类型：Boolean

  备注：has_any_column_privilege检查用户是否以特定方式访问表的任何列。其参数可能与has_table_privilege类似，区别在于访问权限类型必须是SELECT、INSERT、UPDATE或REFERENCES的一些组合。

  示例：

```
gaussdb=# SELECT has_any_column_privilege('test_table','INSERT');
has_any_column_privilege
--------------------------
t
(1 row)

gaussdb=# DROP TABLE test_table;
```

  说明

  拥有表的表级别权限则隐含的拥有该表每列的列级权限，因此如果与has_table_privilege参数相同，has_any_column_privilege总是返回true。但是如果授予至少一列的列级权限也返回成功。

- has_column_privilege(user, table, column, privilege)

  描述：指定用户是否有访问列的权限。

  表 1-107 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名称或id。
table	text，oid	表名	表名称或id。
column	text，smallint	列名	列名称或属性号。
privilege	text	权限	• SELECT：允许对表的指定列执行SELECT语句。 • INSERT：允许对表的指定列执行INSERT语句。 • UPDATE：允许对表的指定列执行UPDATE语句。 • REFERENCES：允许创建一个外键约束（分布式场景暂不支持）。 • COMMENT：允许对表的指定列执行COMMENT语句。

  返回类型：Boolean

版权所有 © 华为云计算技术有限公司

- has_column_privilege(table, column, privilege)
  描述：当前用户是否有访问列的权限，合法参数类型见表1-107。
  返回类型：Boolean
  备注：has_column_privilege检查用户是否以特定方式访问一列。其参数类似于has_table_privilege，可以通过列名或属性号添加列。想要的访问权限类型必须是SELECT、INSERT、UPDATE或REFERENCES的一些组合。

  说明

  拥有表的表级别权限则隐含的拥有该表每列的列级权限。

- has_cek_privilege(user, cek, privilege)
  描述：指定用户是否有访问列加密密钥CEK的权限。

  表 1-108 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或id。
cek	text，oid	列加密密钥	列加密密钥名称或id。
privilege	text	权限	• USAGE：允许使用指定列加密密钥。 • DROP：允许删除指定列加密密钥。

  返回类型：Boolean

- has_cmk_privilege(user, cmk, privilege)
  描述：指定用户是否有访问客户端加密主密钥CMK的权限。

  表 1-109 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或id。
cmk	text，oid	客户端加密主密钥	客户端加密主密钥名称或id。
privilege	text	权限	• USAGE：允许使用指定客户端加密主密钥。 • DROP：允许删除指定客户端加密主密钥。

  返回类型：Boolean

- has_database_privilege(user, database, privilege)
  描述：指定用户是否有访问数据库的权限。

版权所有 © 华为云计算技术有限公司

表 1-110 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或id。
database	text，oid	数据库	数据库名字或id。
privilege	text	权限	• CREATE：对于数据库，允许在数据库里创建新的模式。 • TEMPORARY：允许在使用数据库的时候创建临时表。 • TEMP：允许在使用数据库的时候创建临时表。 • CONNECT：允许用户连接到指定的数据库。 • ALTER：允许用户修改指定对象的属性。 • DROP：允许用户删除指定的对象。 • COMMENT：允许用户定义或修改指定对象的注释。

返回类型：Boolean

- has_database_privilege(database, privilege)

  描述：当前用户是否有访问数据库的权限，合法参数类型见**表1-110**。

  返回类型：Boolean

  备注：has_database_privilege检查用户是否能以在特定方式访问数据库。其参数类似has_table_privilege。访问权限类型必须是CREATE、CONNECT、TEMPORARY或TEMP（等价于TEMPORARY）的一些组合。

- has_directory_privilege(user, directory, privilege)

表 1-111 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或id。
directory	text，oid	目录	目录名字或者oid。
privilege	text	权限	• READ：允许对该目录进行读操作。 • WRITE:允许对该目录进行写操作。

版权所有 © 华为云计算技术有限公司

描述：指定用户是否有访问directory的权限。

返回类型：Boolean

- has_directory_privilege(directory, privilege)

  描述：当前用户是否有访问directory的权限，合法参数类型见表1-111。

  返回类型：Boolean

- has_foreign_data_wrapper_privilege(user, fdw, privilege)

  表 1-112 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或者id。
fdw	text，oid	外部数据封装器	外部数据封装器名字或id。
privilege	text	权限	USAGE：允许访问外部数据封装器。

  描述：指定用户是否有访问外部数据封装器的权限。

  返回类型：Boolean

- has_foreign_data_wrapper_privilege(fdw, privilege)

  描述：当前用户是否有访问外部数据封装器的权限，合法参数类型见表1-112。

  返回类型：Boolean

  备注：has_foreign_data_wrapper_privilege检查用户是否能以特定方式访问外部数据封装器。其参数类似has_table_privilege。访问权限类型必须是USAGE。

- has_function_privilege(user, function, privilege)

  表 1-113 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或者id
function	text，oid	函数	函数名称或id
privilege	text	权限	EXECUTE：允许使用指定的函数，以及利用这些函数实现的操作符。 • ALTER：允许用户修改指定对象的属性。 • DROP：允许用户删除指定的对象。 • COMMENT：允许用户定义或修改指定对象的注释。

  描述：指定用户是否有访问函数的权限。

版权所有 © 华为云计算技术有限公司

返回类型：Boolean

- has_function_privilege(function, privilege)

  描述：当前用户是否有访问函数的权限。合法参数类型见**表1-113**。

  返回类型：Boolean

  备注：has_function_privilege检查一个用户是否能以指定方式访问一个函数。其参数类似has_table_privilege。使用文本字符而不是OID声明一个函数时，允许输入的类型和regprocedure数据类型一样（请参考**对象标识符类型**）。访问权限类型必须是EXECUTE。

- has_language_privilege(user, language, privilege)

  表 1-114 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	用户名字或id。
language	text，oid	语言	语言名称或id。
privilege	text	权限	USAGE：对于过程语言，允许用户在创建函数的时候指定过程语言。

  描述：指定用户是否有访问语言的权限。

  返回类型：Boolean

- has_language_privilege(language, privilege)

  描述：当前用户是否有访问语言的权限。合法参数类型见**表1-114**。

  返回类型：Boolean

  备注：has_language_privilege检查用户是否能以特定方式访问一个过程语言。其参数类似has_table_privilege。访问权限类型必须是USAGE。

- has_nodegroup_privilege(user, nodegroup, privilege)

  描述：检查用户是否有集群节点访问权限。

  返回类型：Boolean

  表 1-115 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	已有的用户名称或id。
nodegroup	text，oid	集群节点	已有的集群节点。

版权所有 © 华为云计算技术有限公司

参数名	合法入参类型	描述	取值范围
privilege	text	权限	• USAGE：对于子集群，对包含在指定模式中的对象有访问权限时，USAGE允许访问指定子集群下的表对象。 • CREATE：对于子集群，允许在子集群中创建表对象。 • COMPUTE：针对计算子集群，允许用户在具有compute权限的计算子集群上进行弹性计算。 • ALTER：允许用户修改指定对象的属性。 • DROP：允许用户删除指定的对象。

- has_nodegroup_privilege(nodegroup, privilege)
  描述：检查用户是否有集群节点访问权限。
  返回类型：Boolean
- has_schema_privilege(user, schema, privilege)
  描述：指定用户是否有访问模式的权限。
  返回类型：Boolean
- has_schema_privilege(schema, privilege)
  描述：当前用户是否有访问模式的权限。
  返回类型：Boolean
  备注：has_schema_privilege检查用户是否能以特定方式访问一个模式。其参数类似has_table_privilege。访问权限类型必须是CREATE、USAGE、ALTER、DROP或COMMENT的一些组合。当检查类型里包含CREATE权限且被检查的Schema是用户同名的Schema时，由于同名Schema的特殊约束，必须拥有Schema的OWNER权限，该函数才会返回TRUE。
- has_sequence_privilege(user, sequence, privilege)
  描述：指定用户是否有访问序列的权限。
  返回类型：Boolean

表 1-116 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name，oid	用户	已有的用户名称或id。
sequence	text，oid	序列	已有的序列名称或id。

版权所有 © 华为云计算技术有限公司

参数名	合法入参类型	描述	取值范围
privilege	text	权限	• USAGE：对于序列，USAGE允许使用nextval函数。 • SELECT：允许创建序列。 • UPDATE：允许执行UPDATE语句。 • ALTER：允许用户修改指定对象的属性。 • DROP：允许用户删除指定的对象。 • COMMENT：允许用户定义或修改指定对象的注释。

- has_sequence_privilege(sequence, privilege)

  描述：指定当前用户是否有访问序列的权限。

  返回类型：Boolean
- has_server_privilege(user, server, privilege)

  描述：指定用户是否有访问外部服务的权限。

  返回类型：Boolean
- has_server_privilege(server, privilege)

  描述：当前用户是否有访问外部服务的权限。

  返回类型：Boolean

  备注：has_server_privilege检查用户是否能以指定方式访问一个外部服务器。其参数类似has_table_privilege。访问权限类型必须是USAGE、ALTER、DROP或COMMENT之一的值。
- has_table_privilege(user, table, privilege)

  描述：指定用户是否有访问表的权限。

  返回类型：Boolean
- has_table_privilege(table, privilege)

  描述：当前用户是否有访问表的权限。

  返回类型：Boolean

  备注：has_table_privilege检查用户是否以特定方式访问表。用户可以通过名称或OID（pg_authid.oid）来指定，public表明PUBLIC伪角色，或如果缺省该参数，则使用current_user。该表可以通过名称或者OID声明。如果用名称声明，则在必要时可以用模式进行修饰。如果使用文本字符串来声明所希望的权限类型，这个文本字符串必须是SELECT、INSERT、UPDATE、DELETE、TRUNCATE、REFERENCESTRIGGER、ALTER、DROP、COMMENT、INDEX或VACUUM之一的值。可以给权限类型添加WITH GRANT OPTION，用来测试权限是否拥有授权选项。也可以用逗号分隔列出的多个权限类型，如果拥有任何所列出的权限，则结果为true。

  示例：

  ```
  gaussdb=# CREATE SCHEMA tpcds;
  gaussdb=# CREATE TABLE tpcds.web_site(col int);
  ```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE USER test_user PASSWORD '********';
gaussdb=# SELECT has_table_privilege('tpcds.web_site', 'select');
 has_table_privilege
---------------------
 t
(1 row)

gaussdb=# SELECT has_table_privilege('test_user', 'tpcds.web_site', 'select,INSERT WITH GRANT
OPTION ');
 has_table_privilege
---------------------
 f
(1 row)

gaussdb=# DROP USER test_user;
gaussdb=# DROP TABLE tpcds.web_site;
gaussdb=# DROP SCHEMA tpcds;
```

- has_tablespace_privilege(user, tablespace, privilege)

  描述：指定用户是否有访问表空间的权限。

  返回类型：Boolean

- has_tablespace_privilege(tablespace, privilege)

  描述：当前用户是否有访问表空间的权限。

  返回类型：Boolean

  备注：has_tablespace_privilege检查用户是否能以特定方式访问一个表空间。其参数类似has_table_privilege。访问权限类型必须是CREATE、ALTER、DROP或COMMENT之一的值。

- pg_has_role(user, role, privilege)

  描述：指定用户是否有角色的权限。

  返回类型：Boolean

- pg_has_role(role, privilege)

  描述：当前用户是否有角色的权限。

  返回类型：Boolean

  备注：pg_has_role检查用户是否能以特定方式访问一个角色。其参数类似has_table_privilege，除了public不能用做用户名。访问权限类型必须是MEMBER或USAGE的一些组合。 MEMBER表示的是角色中的直接或间接成员关系（也就是SET ROLE的权限），而USAGE表示无需通过SET ROLE也直接拥有角色的使用权限。

- has_any_privilege(user, privilege)

  描述：指定用户是否有某项ANY权限，若同时查询多个权限，只要具有其中一个则返回true。

  返回类型：Boolean

表 1-117 参数类型说明

参数名	合法入参类型	描述	取值范围
user	name	用户	已存在的用户名。

版权所有 © 华为云计算技术有限公司

参数名	合法入参类型	描述	取值范围
privilege	text	ANY权限	可选取值： • CREATE ANY TABLE [WITH ADMIN OPTION] • ALTER ANY TABLE [WITH ADMIN OPTION] • DROP ANY TABLE [WITH ADMIN OPTION] • SELECT ANY TABLE [WITH ADMIN OPTION] • INSERT ANY TABLE [WITH ADMIN OPTION] • UPDATE ANY TABLE [WITH ADMIN OPTION] • DELETE ANY TABLE [WITH ADMIN OPTION] • TRUNCATE ANY TABLE [WITH ADMIN OPTION] • CREATE ANY SEQUENCE [WITH ADMIN OPTION] • CREATE ANY INDEX [WITH ADMIN OPTION] • CREATE ANY FUNCTION [WITH ADMIN OPTION] • EXECUTE ANY FUNCTION [WITH ADMIN OPTION] • CREATE ANY PACKAGE [WITH ADMIN OPTION] • EXECUTE ANY PACKAGE [WITH ADMIN OPTION] • CREATE ANY TYPE [WITH ADMIN OPTION]

模式可见性查询函数

每个函数执行检查数据库对象类型的可见性。对于函数和操作符，如果在前面的搜索路径中没有相同的对象名称和参数的数据类型，则此对象是可见的。对于操作符类，则要同时考虑名称和相关索引的访问方法。

所有这些函数都需要使用OID来标识需要检查的对象。如果用户想通过名称测试对象，则使用OID别名类型（regclass、regtype、regprocedure、regoperator、regconfig或regdictionary）将会很方便。

版权所有 © 华为云计算技术有限公司

比如，如果一个表所在的模式在搜索路径中，并且在前面的搜索路径中没有同名的表，则这个表是可见的。它等效于表可以不带明确模式修饰进行引用。比如，要列出所有可见表的名称：

```
gaussdb=# SELECT relname FROM pg_class WHERE pg_table_is_visible(oid);
```

- pg_collation_is_visible(collation_oid)
  描述：该排序是否在搜索路径中可见。
  返回类型：Boolean
- pg_conversion_is_visible(conversion_oid)
  描述：该转换是否在搜索路径中可见。
  返回类型：Boolean
- pg_function_is_visible(function_oid)
  描述：该函数是否在搜索路径中可见。
  返回类型：Boolean
- pg_opclass_is_visible(opclass_oid)
  描述：该操作符类是否在搜索路径中可见。
  返回类型：Boolean
- pg_operator_is_visible(operator_oid)
  描述：该操作符是否在搜索路径中可见。
  返回类型：Boolean
- pg_opfamily_is_visible(opclass_oid)
  描述：该操作符族是否在搜索路径中可见。
  返回类型：Boolean
- pg_table_is_visible(table_oid)
  描述：该表是否在搜索路径中可见。
  返回类型：Boolean
- pg_ts_config_is_visible(config_oid)
  描述：该文本检索配置是否在搜索路径中可见。
  返回类型：Boolean
- pg_ts_dict_is_visible(dict_oid)
  描述：该文本检索词典是否在搜索路径中可见。
  返回类型：Boolean
- pg_ts_parser_is_visible(parser_oid)
  描述：该文本搜索解析是否在搜索路径中可见。
  返回类型：Boolean
- pg_ts_template_is_visible(template_oid)
  描述：该文本检索模板是否在搜索路径中可见。
  返回类型：Boolean
- pg_type_is_visible(type_oid)
  描述：该类型（或域）是否在搜索路径中可见。
  返回类型：Boolean

 版权所有 © 华为云计算技术有限公司

系统表信息函数

- format_type(type_oid, typemod)

  描述：获取数据类型的SQL名称。

  返回类型：text

  备注：format_type通过某个数据类型的OID以及可能的修饰词，返回其SQL名称。如果不知道具体的修饰词，则在修饰词的位置传入NULL。修饰词一般只对有长度限制的数据类型有意义。format_type所返回的SQL名称中包含数据类型的长度值，其大小是：实际存储长度len - sizeof(int32)，单位字节。原因是数据存储时需要32位的空间来存储用户对数据类型的自定义长度信息，即实际存储长度要比用户定义长度多4个字节。在下例中，format_type返回的SQL名称为“character varying(6)”，6表示varchar类型的长度值是6字节，因此该类型的实际存储长度为10字节。

  ```
  gaussdb=# SELECT format_type((SELECT oid FROM pg_type WHERE typname='varchar'), 10);
      format_type
  ----------------------
   character varying(6)
  (1 row)
  ```

- pg_check_authid(role_oid)

  描述：检查是否存在给定oid的角色名。

  返回类型：Boolean

  ```
  gaussdb=# select pg_check_authid(1);
  pg_check_authid
  -----------------
  f
  (1 row)
  ```

- pg_describe_object(catalog_id, object_id, object_sub_id)

  描述：获取数据库对象的描述。

  返回类型：text

  备注：pg_describe_object返回由目录OID，对象OID和一个（或许0个）子对象ID指定的数据库对象的描述。这有助于确认存储在pg_depend系统表中对象的身份。

- pg_get_constraintdef(constraint_oid)

  描述：获取约束的定义。

  返回类型：text

- pg_get_constraintdef(constraint_oid, pretty_bool)

  描述：获取约束的定义。

  返回类型：text

  备注：pg_get_constraintdef和pg_get_indexdef分别从约束或索引上使用创建命令进行重构。

- pg_get_expr(pg_node_tree, relation_oid)

  描述：反编译表达式的内部形式，假设其中的任何Vars都引用第二个参数指定的关系。

  返回类型：text

- pg_get_expr(pg_node_tree, relation_oid, pretty_bool)

  描述：反编译表达式的内部形式，假设其中的任何Vars都引用第二个参数指定的关系。

  返回类型：text

版权所有 © 华为云计算技术有限公司

备注：pg_get_expr反编译一个独立表达式的内部形式，比如一个字段的缺省值。便于检查系统表的内容。如果表达式可能包含关键字，则指定关键字引用相关的OID作为第二个参数；如果没有关键字，用零表示即可。

- pg_get_functiondef(func_oid)

  描述：获取函数的定义。

  返回类型：text

  示例：

```
gaussdb=# SELECT * FROM pg_get_functiondef(598);
 headerlines |                  definition
-------------+-----------------------------------------------
           4 | CREATE OR REPLACE FUNCTION pg_catalog.abbrev(inet)+
             |  RETURNS text                                 +
             |  LANGUAGE internal                            +
             |  IMMUTABLE STRICT NOT FENCED NOT SHIPPABLE    +
             | AS $function$inet_abbrev$function$            +
             |
(1 row)
```

- pg_get_function_arguments(func_oid)

  描述：获取函数定义的参数列表（带默认值）。

  返回类型：text

  备注：pg_get_function_arguments返回一个函数的参数列表，需要在CREATE FUNCTION中使用这种格式。

- pg_get_function_identity_arguments(func_oid)

  描述：获取参数列表来确定一个函数 (不带默认值)。

  返回类型：text

  备注：pg_get_function_identity_arguments返回需要的参数列表用来标识函数，这种形式需要在ALTER FUNCTION中使用，并且这种形式省略了默认值。

- pg_get_function_result(func_oid)

  描述：获取函数的RETURNS子句。

  返回类型：text

  备注：pg_get_function_result为函数返回适当的RETURNS子句。

- pg_get_indexdef(index_oid)

  描述：获取索引的CREATE INDEX命令。

  返回类型：text

  示例：

```
gaussdb=# CREATE TABLE test3 (a int, b int);
gaussdb=# CREATE INDEX test3_b_index ON test3 USING btree(b) TABLESPACE pg_default;
gaussdb=# SELECT pg_get_indexdef(oid) FROM (SELECT oid FROM pg_class WHERE relname =
'test3_b_index') tmp;
                                    pg_get_indexdef
-----------------------------------------------------------------------------------------------------------
 CREATE INDEX test3_b_index ON test3 USING ubtree (b) WITH (storage_type=USTORE) TABLESPACE
pg_default
(1 row)
```

- pg_get_indexdef(index_oid, dump_schema_only)

  描述：获取索引的CREATE INDEX命令，仅用于dump场景。当前版本dump_schema_only参数取值对函数输出结果无影响。

  返回类型：text

  示例：

```
gaussdb=# SELECT pg_get_indexdef(oid, true) FROM (SELECT oid FROM pg_class WHERE relname =
'test3_b_index') tmp;
```


版权所有 © 华为云计算技术有限公司

```
                                    pg_get_indexdef
-----------------------------------------------------------------------------------------------------------
 CREATE INDEX test3_b_index ON test3 USING ubtree (b) WITH (storage_type=USTORE) TABLESPACE
pg_default
(1 row)
```

- pg_get_indexdef(index_oid, column_no, pretty_bool)

  描述：获取索引的CREATE INDEX命令，如果column_no不为零，则只获取一个索引字段的定义。

  返回类型：text

  示例：

```
gaussdb=# SELECT pg_get_indexdef(oid, 0, false) FROM (SELECT oid FROM pg_class WHERE relname
= 'test3_b_index') tmp;
                                    pg_get_indexdef
-----------------------------------------------------------------------------------------------------------
 CREATE INDEX test3_b_index ON test3 USING ubtree (b) WITH (storage_type=USTORE) TABLESPACE
pg_default
(1 row)


gaussdb=# SELECT pg_get_indexdef(oid, 1, false) FROM (SELECT oid FROM pg_class WHERE relname
= 'test3_b_index') tmp;
 pg_get_indexdef
-----------------
 b
(1 row)

gaussdb=# DROP INDEX test3_b_index;
gaussdb=# DROP TABLE test3;
```

  备注：pg_get_idexdef为函数返回一个完整的CREATE OR REPLACE INDEX语句。

- pg_get_keywords()

  描述：获取SQL关键字和类别列表。

  返回类型：setof record

  备注：pg_get_keywords返回一组关于描述服务器识别SQL关键字的记录。word列包含关键字。catcode列包含一个分类代码：U表示通用的、C表示列名、T表示类型或函数名、R表示保留。catdesc列包含了一个可能本地化描述分类的字符串。不同兼容性模式下，关键字的范围与类型不完全一致，pg_get_keywords的返回结果也有所差异。pg_get_keywords函数返回的记录不会包含disable_keyword_options参数中配置的选项，disable_keyword_options参数说明具体请参见**平台和客户端兼容性**。

- pg_get_ruledef(rule_oid)

  描述：获取规则的CREATE RULE命令。

  返回类型：text

- pg_get_ruledef(rule_oid, pretty_bool)

  描述：获取规则的CREATE RULE命令。

  返回类型：text

- pg_get_userbyid(role_oid)

  描述：获取给定OID的角色名。

  返回类型：name

  备注：pg_get_userbyid通过角色的OID抽取对应的用户名。

- pg_check_authid(role_id)

  描述：通过role_id检查用户是否存在。

  返回类型：text

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT pg_check_authid(20);
pg_check_authid
-----------------
f
(1 row)
```

- pg_get_viewdef(view_name)

  描述：为视图获取底层的SELECT命令。

  返回类型：text

- pg_get_viewdef(view_name, pretty_bool)

  描述：为视图获取底层的SELECT命令，如果pretty_bool为true，行字段可以包含80列。

  返回类型：text

  备注：pg_get_viewdef重构出定义视图的SELECT查询。这些函数通常都有两种形式，其中带有pretty_bool参数且参数为true时，是“适合打印”的结果，这种格式更容易读。另一种是缺省的格式，更有可能被将来的不同版本用同样的方法解释。如果是用于转储，那么尽可能避免使用适合打印的格式。给pretty-print参数传递false生成的结果和没有这个参数的变种生成的结果完全一样。

- pg_get_viewdef(view_oid)

  描述：为视图获取底层的SELECT命令。

  返回类型：text

- pg_get_viewdef(view_oid, pretty_bool)

  描述：为视图获取底层的SELECT命令，如果pretty_bool为true，行字段可以包含80列。

  返回类型：text

- pg_get_viewdef(view_oid, wrap_column_int)

  描述：为视图获取底层的SELECT命令；行字段被换到指定的列数，打印是隐含的。

  返回类型：text

- pg_get_tabledef(table_oid)

  描述：根据table_oid获取表定义。

  返回类型：text

  示例：

```
gaussdb=# CREATE TABLE t1 (c1 serial);
gaussdb=# SELECT pg_get_tabledef(oid) FROM (SELECT oid FROM pg_class WHERE relname = 't1')
tmp;
                          pg_get_tabledef
------------------------------------------------------------------------------
 SET search_path = public;                                                   +
 CREATE TABLE t1 (                                                           +
    c1 integer DEFAULT nextval('t1_c1_seq1790791646'::regclass) NOT NULL+
 )                                                                           +
 WITH (orientation=row, compression=no, storage_type=USTORE, segment=off)+
 DISTRIBUTE BY HASH(c1)                                                      +
 TO GROUP group_version1;
(1 row)
```

- pg_get_tabledef(table_name)

  描述：根据table_name获取表定义。

  返回类型：text

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM pg_get_tabledef('t1');
                          pg_get_tabledef
------------------------------------------------------------------------------
 SET search_path = public;                                                   +
 CREATE TABLE t1 (                                                           +
     c1 integer DEFAULT nextval('t1_c1_seq1790791646'::regclass) NOT NULL+
 )                                                                           +
 WITH (orientation=row, compression=no, storage_type=USTORE, segment=off)+
 DISTRIBUTE BY HASH(c1)                                                      +
 TO GROUP group_version1;
(1 row)
```

备注：pg_get_tabledef重构出表定义的CREATE语句，包含了表定义本身、索引信息、comments和ilm策略信息。对于表对象依赖的group、schema、tablespace、server等信息，需要用户自己创建。如果ilm策略信息是继承自父对象，pg_get_tabledef不会返回该数据对象的ilm策略信息创建语句。

- pg_options_to_table(reloptions)

  描述：获取存储选项名称/值对的集合。

  返回类型：setof record

  备注：pg_options_to_table当通过pg_class.reloptions或pg_attribute.attoptions时返回存储选项名称/值对（option_name/option_value）的集合。

- pg_tablespace_databases(tablespace_oid)

  描述：获取在指定的表空间中有对象的数据库OID集合。

  返回类型：setof oid

  备注：pg_tablespace_databases允许检查表空间的状况，返回在该表空间中保存了对象的数据库OID集合。如果这个函数返回数据行，则该表空间就是非空的，因此不能删除。要显示该表空间中的特定对象，用户需要连接pg_tablespace_databases标识的数据库与查询pg_class系统表。

- pg_tablespace_location(tablespace_oid)

  描述：获取表空间所在的文件系统的路径。

  返回类型：text

- pg_typeof(any)

  描述：获取任何值的数据类型。

  返回类型：regtype

  备注：pg_typeof返回传递给它的值的数据类型OID。这可能有助于故障排除或动态构造SQL查询。声明此函数返回regtype，这是一个OID别名类型（请参考**对象标识符类型**）；这意味着它是一个为了比较而显示类型名称的OID。

  示例：

  ```
  gaussdb=# SELECT pg_typeof(33);
   pg_typeof
  -----------
   integer
  (1 row)

  gaussdb=# SELECT typlen FROM pg_type WHERE oid = pg_typeof(33);
   typlen
  --------
       4
  (1 row)
  ```

- collation for (any)

  描述：获取参数的排序。

  返回类型：text

  备注：表达式collation for返回传递给它的值的排序。示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT collation for (description) FROM pg_description LIMIT 1;
 pg_collation_for
------------------
 "default"
(1 row)
```

值可能是引号括起来的并且模式限制的。如果没有为参数表达式排序，则返回一个null值。如果参数不是排序的类型，则提示一个错误。

- getdistributekey(table_name)

  描述：获取一个hash表的分布列。

  返回类型：text

  示例：

  ```
  gaussdb=# SELECT getdistributekey('t1');
   getdistributekey
  ------------------
   c1
  (1 row)
  ```

- pg_extension_update_paths(name)

  描述：返回指定扩展的版本更新路径，本函数仅系统管理员可调用。

  返回类型：text（source text）, text（target text）, text（path text）

- pg_get_serial_sequence(tablename, colname)

  描述：获取对应表名和列名上的序列。

  返回类型：text

  示例：

  ```
  gaussdb=# SELECT * FROM pg_get_serial_sequence('t1', 'c1');
     pg_get_serial_sequence
  ----------------------------
   public.t1_c1_seq1790791646
  (1 row)
  ```

- pg_sequence_parameters(sequence_oid, OUT start_value int16, OUT minimum_value int16, OUT maximum_value int16, OUT increment int16, OUT cycle_option Boolean, OUT order_option Boolean)

  描述：获取指定sequence的参数，包含起始值、最小值、最大值、步长、循环选项以及有序选项。

  返回类型：int16，int16，int16，int16，Boolean，Boolean

  示例：

  ```
  -- 使用上方创建表t1时自动生成的序列名称。
  gaussdb=# SELECT oid FROM pg_class WHERE relname = 't1_c1_seq1790791646';
    oid
  -------
   17120
  (1 row)

  -- 使用上方查询出来的OID。
  gaussdb=# SELECT * FROM  pg_sequence_parameters(17120);
   start_value | minimum_value |     maximum_value     | increment | cycle_option | order_option
  -------------+---------------+-----------------------+-----------+--------------+--------------
             1 |             1 | 9223372036854775807 |         1 | f            | f
  (1 row)
  ```

- pgxc_get_variable_info()

  描述：获取节点上的变量值，包括nodeName、nextOid、nextXid、oldestXid、xidVacLimit、oldestXidDB、lastExtendCSNLogpage、startExtendCSNLogpage、nextCommitSeqNo、latestCompleteXid,startupMaxXid。

版权所有 © 华为云计算技术有限公司

返回类型：set of pg_variable_info

示例：

```
gaussdb=# SELECT pgxc_get_variable_info( );
                    pgxc_get_variable_info
-----------------------------------------------------------------------
(dn_6004_6005_6006,25617,141396349,2073,20000002073,15808,138111,0,127154152,141396348,104
433004)
(1 row)
```

- gs_get_index_status(schema_name, index_name)

  描述：获取所有节点上的索引状态信息，包括索引是否可插入以及索引是否可用，主要用于在线创建索引过程中或者构建失败的情况下，确认索引状态。返回值包括节点名称node_name、索引是否可插入状态indisready、索引是否可以用状态indisvalid，只有当所有节点上索引的indisready和indisvalid状态均为true，且索引状态没有被修改为unusable的情况下，当前索引才可用。

  返回类型：text，boolean，boolean

  示例：

```
gaussdb=# CREATE INDEX index1 ON t1(c1);
gaussdb=# SELECT * FROM gs_get_index_status('public', 'index1');
   node_name    | indisready | indisvalid
----------------+------------+------------
 datanode1      | t          | t
 datanode2      | t          | t
 coordinator1   | t          | t
(3 row)

gaussdb=# DROP INDEX index1;
gaussdb=# DROP TABLE t1;
```

注释信息函数

- col_description(table_oid, column_number)

  描述：获取一个表字段的注释。

  返回类型：text

  备注：col_description返回一个表中字段的注释，通过表OID和字段号来声明。

- obj_description(object_oid, catalog_name)

  描述：获取一个数据库对象的注释。

  返回类型：text

  备注：带有两个参数的obj_description返回一个数据库对象的注释，该对象是通过其OID和其所属的系统表名称声明。比如，obj_description(123456,'pg_class')将返回OID为123456的表的注释。只带一个参数的obj_description只要求对象OID。

  obj_description不能用于表字段，因为字段没有自己的OID。

- obj_description(object_oid)

  描述：获取一个数据库对象的注释。

  返回类型：text

- shobj_description(object_oid, catalog_name)

  描述：获取一个共享数据库对象的注释。

  返回类型：text

  备注：shobj_description和obj_description差不多，不同之处仅在于前者用于共享对象。一些系统表是通用于集群中所有数据库的全局表，因此这些表的注释也是全局存储的。

版权所有 © 华为云计算技术有限公司

事务 ID 和快照

内部事务ID类型（xid）是64位。这些函数使用的数据类型txid_snapshot，存储在特定时刻事务ID可见性的信息。其组件描述如表1-118所示。

表 1-118 快照组件

名称	描述
xmin	最早的事务ID（txid）仍然活动。所有较早事务将是已经提交可见的，或者是直接回滚。
xmax	作为尚未分配的txid。所有大于或等于此txids的都是尚未开始的快照时间，因此不可见。
xip_list	当前快照中活动的txids。这个列表只包含在xmin和xmax之间活动的txids，有可能活动的txids高于xmax。介于大于等于xmin、小于xmax，并且不在这个列表中的txid，在这个时间快照已经完成的，因此按照提交状态查看它是可见还是回滚。这个列表不包含子事务的txids。

txid_snapshot的文本表示为：xmin:xmax:xip_list。

示例：10:20:10,14,15意思为：xmin=10, xmax=20, xip_list=10, 14, 15。

以下的函数在一个输出形式中提供服务器事务信息。这些函数的主要用途是为了确定在两个快照之间有哪些事务提交。

- pgxc_is_committed(transaction_id)

  描述：如果普通表提交或忽略给定的XID（gxid）。NULL表示的状态是未知的（运行、准备、冻结等）。如果在维护模式（GUC参数xc_maintenance_mode为on时）下，会遍历所有bucketid和普通表对应的xid状态，如果有状态冲突，例如同一个xid，在bucket1状态为提交，在bucket2为回滚则报错。该模式谨慎打开，一般供维护人员排查问题使用，一般用户不应使用该模式。

  返回类型：Boolean

- pgxc_is_committed(transaction_id, bucketid)

  描述：如果指定bucketid提交或忽略给定的XID（gxid）。NULL表示的状态是未知的（运行、准备、冻结等），bucketid是hash bucket表的物理bucketid，普通表为-1。如果在维护模式(GUC参数xc_maintenance_mode为on时)下，指定bucketid为-1，会遍历所有bucketid和普通表对应的xid状态，如果有状态冲突，例如同一个xid，在bucket1状态为提交，在bucket2为回滚则报错。该模式谨慎打开，一般供维护人员排查问题使用，一般用户不应使用该模式。

  返回类型：Boolean

- txid_current()

  描述：获取当前事务ID。

  返回类型：bigint

- gs_txid_oldestxmin()

  描述：获取当前最小事务id的值oldestxmin。

  返回类型：bigint

- txid_current_snapshot()

版权所有 © 华为云计算技术有限公司

描述：获取当前快照。

返回类型：txid_snapshot

- txid_snapshot_xip(txid_snapshot)

  描述：在快照中获取正在进行的事务ID。

  返回类型：setof bigint

- txid_snapshot_xmax(txid_snapshot)

  描述：获取快照的xmax。

  返回类型：bigint

- txid_snapshot_xmin(txid_snapshot)

  描述：获取快照的xmin。

  返回类型：bigint

- txid_visible_in_snapshot(bigint, txid_snapshot)

  描述：在快照中事务ID是否可见（不使用子事务ID）。

  返回类型：Boolean

- get_local_prepared_xact()

  描述：获取当前节点两阶段残留事务信息，包括事务id，两阶段gid名称，prepared的时间，owner的oid，database的oid及当前节点的node_name。

  返回类型：xid, text, timestamptz, oid, text

- get_remote_prepared_xacts()

  描述：获取所有远程节点两阶段残留事务信息，包括事务id，两阶段gid名称，prepared的时间，owner的名称，database的名称及node_name。

  返回类型：xid, text, timestamptz, name, text

- global_clean_prepared_xacts(text, text)

  描述：并发清理两阶段残留事务，仅gs_clean工具调用该函数进行清理，其他情况下调用均返回false。

  返回类型：Boolean

- pgxc_stat_get_wal_senders()

  描述：返回集群中所有主DN的发送日志的信息和其对应的备DN的接收日志的信息。仅支持system admin或monitor admin权限用户使用。

  返回值如**表1-119**所示。

表 1-119 pgxc_stat_get_wal_senders 返回参数说明

字段名	描述
nodename	实例名。
sender_pid	发送日志的线程pid。
local_role	实例角色。
peer_role	接收端的实例的角色。
peer_state	接收端的实例的状态。
state	实例间同步的状态。

版权所有 © 华为云计算技术有限公司

字段名	描述
sender_sent_location	发送端发送日志的位置。
sender_write_location	发送端写日志的位置。
sender_flush_location	发送端刷盘日志的位置。
sender_replay_location	当前实例的日志位置。如果是主DN，则该位置和sender_flush_location相同；否则该位置为当前实例日志回放到的位置。
receiver_received_location	接收端日志接收到的位置。
receiver_write_location	接收端日志写的位置。
receiver_flush_location	接收端日志刷盘的位置。
receiver_replay_location	接收端日志回放的位置。

- pgxc_stat_get_wal_senders_status()

  描述：返回所有节点事务日志接收状态。仅支持system admin或monitor admin权限用户使用。

  返回值如表1-120所示。

  表 1-120 pgxc_stat_get_wal_senders_status 返回参数说明

字段名	描述
nodename	主节点名。
source_ip	主节点IP。
source_port	主节点端口。
dest_ip	备节点IP。
dest_port	备节点端口。
sender_pid	发送线程PID。
local_role	主节点类型。
peer_role	备节点类型。
peer_state	备节点状态。
state	wal sender状态。

版权所有 © 华为云计算技术有限公司

字段名	描述
sender_sent_location	主节点发送位置。
sender_write_location	主节点落盘位置。
sender_replay_location	主节点redo位置。
receiver_received_location	备节点接收位置。
receiver_write_location	备节点落盘位置。
receiver_flush_location	备节点flush磁盘位置。
receiver_replay_location	备节点redo位置。

- gs_get_next_xid_csn()
  描述：返回全局所有节点上的next_xid和next_csn值。
  返回值如：**表1-121**所示。

  表 1-121 gs_get_next_xid_csn 返回参数说明

字段名	描述
nodename	节点名称。
next_xid	当前节点下一个事务id号。
next_csn	当前节点下一个csn号。

- pg_control_system()
  描述：返回系统控制文件状态。
  返回类型：SETOF record
- pg_control_checkpoint()
  描述：返回系统检查点状态。
  返回类型：SETOF record
- get_prepared_pending_xid()
  描述：当恢复完成时，返回nextxid。
  返回值类型：text
- pg_clean_region_info()
  描述：清理regionmap。
  返回值类型：character varying
- pg_get_replication_slot_name()
  描述：获取slot name。
  返回值类型：text
- pg_get_running_xacts()
  描述：获取运行中的xact。

版权所有 © 华为云计算技术有限公司

返回值类型：handle integer, gxid xid, state tinyint, node text, xmin xid, vacuum boolean, timeline bigint, prepare_xid xid, pid bigint, next_xid xid，dbid oid。

- pg_get_variable_info()

  描述：获取共享内存变量cache。

  返回值类型：node_name text, nextOid oid, nextXid xid, oldestXid xid, xidVacLimit xid, oldestXidDB oid, lastExtendCSNLogpage xid, startExtendCSNLogpage xid, nextCommitSeqNo xid, latestCompletedXid xid, startupMaxXid xid

- pg_get_xidlimit()

  描述：从共享内存获取事务id信息。

  返回值类型：nextXid xid, oldestXid xid, xidVacLimit xid, xidWarnLimit xid, xidStopLimit xid, xidWrapLimit xid, oldestXidDB oid

- pg_relation_compression_ratio(text)

  描述：查询表压缩率，默认返回1.0。

  参数：text

  返回值类型：real

- pg_relation_with_compression(text)

  描述：查询表是否压缩。

  参数：text

  返回值类型：boolean

- pg_stat_file_recursive(location text[, get_turbo_info boolean])

  描述：列出路径下所有文件，仅支持数据库初始用户执行。

  参数：location text[, get_turbo_info boolean]

  返回值类型：path text, filename text, size bigint, isdir boolean[, isturbocompress boolean]

- pg_stat_get_activity_for_temptable()

  描述：返回临时表相关的后台线程的记录。

  返回值类型：datid oid, timelineid integer, tempid integer, sessionid bigint

- pg_stat_get_activity_ng(pid bigint)

  描述：返回nodegroup相关的后台线程的记录。

  参数：pid bigint

  返回值类型：datid oid, pid bigint, sessionid bigint, node_group text

- pg_stat_get_cgroup_info()

  描述：返回cgroup信息。

  返回值类型：cgroup_name text, percent integer, usage_percent integer, shares bigint, usage bigint, cpuset text, relpath text, valid text, node_group text

- pg_stat_get_realtime_info_internal(oid, oid, bigint, cstring, oid)

  描述：返回实时信息，当前该接口已不可用，返回FailedToGetSessionInfo。

  参数：oid, oid, bigint, cstring, oid

  返回值类型：text

- pg_stat_get_wlm_session_iostat_info()

版权所有 © 华为云计算技术有限公司

描述：返回会话负载I/O信息。

返回值类型：threadid bigint, maxcurr_iops integer, mincurr_iops integer, maxpeak_iops integer, minpeak_iops integer, iops_limits integer, io_priority integer, curr_io_limits integer

- pg_test_err_contain_err(integer)

  描述：测试错误类型和返回信息。

  参数：integer

  返回值类型：void

- pv_session_memory_detail_tp()

  描述：返回会话的内存使用情况，参考pv_session_memory_detail。

  返回值类型：sessid text, sesstype text, contextname text, level smallint, parent text, totalsize bigint, freesize bigint, usedsize bigint

- gs_get_table_distribution(table_name text, schema_name text)

  描述：返回表数据在各个数据节点的分布情况。

  参数：table_name text, schema_name text

  返回值类型：text

- pv_builtin_functions()

  描述：查看所有内置系统函数信息。

  返回值类型：proname name, pronamespace oid, proowner oid, prolang oid, procost real, prorows real, provariadic oid, protransform regproc, proisagg boolean, proiswindow boolean, prosecdef boolean, proleakproof boolean, proisstrict boolean, proretset boolean, provolatile "char", pronargs smallint, pronargdefaults smallint, prorettype oid, proargtypes oidvector, proallargtypes integer[], proargmodes "char"[], proargnames text[], proargdefaults pg_node_tree, prosrc text, probin text, proconfig text[], proacl aclitem[], prodefaultargpos int2vector, fencedmode boolean, proshippable boolean, propackage boolean, oid oid

- pv_thread_memory_detail()

  描述：返回各线程的内存信息。

  返回值类型：threadid text, tid bigint, thrdtype text, contextname text, level smallint, parent text, totalsize bigint, freesize bigint, usedsize bigint

- pg_shared_memory_detail()

  描述：返回所有已产生的共享内存上下文的使用信息，各列描述请参考**DBE_PERF.SHARED_MEMORY_DETAIL**。

  返回值类型：contextname text, level smallint, parent text, totalsize bigint, freesize bigint, usedsize bigint

- pgxc_get_running_xacts()

  描述：返回集群中各个节点运行事务的信息，字段内容和PGXC_RUNNING_XACTS视图（请参见**PGXC_RUNNING_XACTS**）相同。只有system admin和monitor admin用户有权限查看。

  返回值类型：setof record

- pgxc_snapshot_status()

  描述：在GTM模式下，返回GTM中关键内存信息状态，用来支持问题定位，GTM-Free、GTM-Lite不支持本函数。

版权所有 © 华为云计算技术有限公司

返回值类型：xmin xid, xmax xid, xcnt int, oldestxmin xid, next_xid xid, timeline int, active_thread_num int, max_active_thread_num int, snapshot_num int, snapshot_totalsize bigint

返回值描述如：表1-122所示。

表 1-122 pgxc_snapshot_status 返回参数说明

字段名	描述
xmin	GTM上当前最小的活跃事务id。
xmax	GTM上当前提交的最大的事务id + 1，大于等于该值的事务id是活跃的。
xcnt	GTM上当前活跃事务个数。
oldestxmin	GTM上最老被访问的事务id号。
next_xid	GTM上下一个分配的事务id号。
timeline	GTM上当前的时间线。
active_thread_num	GTM上当前活跃的工作线程数。
max_active_thread_num	GTM上1分钟内工作线程数峰值。
snapshot_num	GTM上1分钟内下发的快照个数。
snapshot_totalsize	GTM上1分钟内下发快照总大小。

- get_gtm_lite_status()

  描述：返回GTM上的backupXid和csn号，用来支持问题定位，GTM-FREE模式下不支持使用本系统函数。

  返回值如表1-123所示。

  表 1-123 get_gtm_lite_status 返回参数说明

字段名	描述
backup_xid	GTM上的备份Xid值。
csn	GTM当前下发的最新的csn号。

- adm_hist_snapshot_func()

  描述：返回快照执行时间相关信息，访问该函数需要打开enable_wdr_snapshot参数，并且需要snapshot schema， snapshot表和tables_snap_timestamp表的访问权限。

  返回值类型：snap_id bigint, dbid oid, begin_interval_time timestamp(3), end_interval_time timestamp(3), flush_elapsed interval day(5) to second(1), begin_interval_time_tz timestamp(3) with time zone, end_interval_time_tz timestamp(3) with time zone

- global_sql_patch_func()

版权所有 © 华为云计算技术有限公司

描述：全局各个节点上的SQL PATCH信息，用于返回global_sql_patch视图的结果。

返回值如表1-124所示。

表 1-124 global_sql_patch_func 返回参数说明

名称	类型	描述
node_name	text	SQLpatch所在节点名。
patch_name	name	PATCH名称。
unique_sql_id	bigint	查询全局唯一ID。
owner	oid	PATCH的创建用户ID。
enable	boolean	PATCH是否生效。
status	"char"	SQL PATCH是基于Unique SQL ID还是SQL HASH创建的，'d'代表Unique SQL ID，'h'代表SQL HASH。
abort	boolean	是否是AbortHint。
hint_string	text	Hint文本。
description	text	PATCH的备注。
parent_unique_sql_id	bigint	PATCH生效的SQL语句的外层语句的全局唯一ID，存储过程外的语句该值为0，存储过程内的语句该值为调用该存储过程语句的全局唯一ID。
original_query	text	原始语句。

- gs_get_current_version()
  描述：依据当前编译宏返回当前编译模式，返回'M'。
  返回值类型：char
- gs_get_kernel_info()
  描述：全局各个CN、DN主节点上的事务相关的信息。
  返回值如表1-125所示。

表 1-125 gs_get_kernel_info 返回参数说明

名称	类型	描述
node_name	text	节点名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
module	text	模块名。包括： ● XACT（事务模块）。 ● STANDBY（备机模块）。 ● UNDO（undo模块）。 ● HOTPATH（热补丁模块）。 ● FULL_SQL。 ● LOCK（锁模块）。 ● SQL。 ● SEC （安全模块）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
name	text	探查内存态关键数据名字。包括： ● startup_max_xid（线程启动时最大的xid值）。 ● recent_local_xmin（本地活跃事务最小xid值）recent_global_xmin（全局活跃事务最小xid值）。 ● standby_xmin（备机活跃事务最小xid值）。 ● standby_redo_cleanup_xmin（备机redo时cleanup日志最小xid值）。 ● standby_redo_cleanup_xmin_lsn（备机redo时cleanup日志最小xid的LSN值）。 ● local_csn_min（本地活跃事务最小CSN值）。 ● replication_slot_xmin（复制槽的最小xid值）。 ● replication_slot_catalog_xmin（catalog复制槽最小xid值）。 ● global_recycle_xid（全局undo回收事务的最小xid值）。 ● global_frozen_xid（全局冻结事务的最小的xid值）。 ● recent_xmin（当前会话活跃事务最小xid值）。 ● next_csn（下一个事务的CSN值）。 ● hotpatch_additional_info（热补丁预留字段）。 ● stmt_session_discard_records（内核支持全量SQL因slot满丢弃数据量）。 ● stmt_shm_flush_discard_records（内核支持全量SQL因ringbuf满丢弃数据量）。 ● idle_in_trans_timeout_records（内核支持空闲事务超时次数累计数据量）。 ● total_lock_usage（LOCK：当前通过主锁表申请锁的数量）。 ● total_lock_usage_rate（LOCK：主锁表空间的使用率）。 ● fastpath_strong_relation_locks_count（LOCK：存在FASTPATH强锁冲突的分区及该分区对应的强锁数量）。 ● fastpath_failure_rate（LOCK：由于对应分区存在强锁冲突导致无法通过FASTPATH申请锁的比例）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
		• lock_acquire_lack_fastpath_count（LOCK：由于FASTPATH槽位占满导致通过FASTPATH申请锁失败的总次数）。 • n_soft_parse（SQL：软解析次数）。 • n_hard_parse（SQL：硬解析次数）。 • hsm_access_success_count（SEC：KMS 服务访问成功次数）。 • hsm_access_failure_count（SEC：KMS 服务访问失败次数）。
value	text	探查内存态关键数据值。

示例：

```
gaussdb=# SELECT * FROM gs_get_kernel_info();
  node_name   | module |              name                    |  value
--------------+--------+--------------------------------------+---------
 coordinator1 | XACT   | startup_max_xid                      | 16488
 coordinator1 | XACT   | recent_local_xmin                    | 16504
 coordinator1 | XACT   | recent_global_xmin                   | 16503
 coordinator1 | STANDBY | standby_xmin                        | 0
 coordinator1 | STANDBY | standby_redo_cleanup_xmin           | 0
 coordinator1 | STANDBY | standby_redo_cleanup_xmin_lsn       | 0/0
 coordinator1 | XACT   | local_csn_min                        | 6014225
 coordinator1 | XACT   | replication_slot_xmin                | 0
 coordinator1 | XACT   | replication_slot_catalog_xmin        | 0
 coordinator1 | UNDO   | global_recycle_xid                   | 16501
 coordinator1 | XACT   | global_frozen_xid                    | 0
 coordinator1 | XACT   | recent_xmin                          | 16504
 coordinator1 | XACT   | next_csn                             | 6014226
 coordinator1 | HOTPATH | hotpatch_additional_info            |
 coordinator1 | FULL_SQL| stmt_session_discard_records        | 0
 coordinator1 | FULL_SQL| stmt_shm_flush_discard_records      | 0
 coordinator1 | XACT   | idle_in_trans_timeout_records        | 0
 coordinator1 | LOCK   | total_lock_usage                     | 0
 coordinator1 | LOCK   | total_lock_usage_rate                | 0.00%
 coordinator1 | LOCK   | fastpath_strong_relation_locks_count |
 coordinator1 | LOCK   | fastpath_failure_rate                | 0.00%
 coordinator1 | LOCK   | lock_acquire_lack_fastpath_count     | 0
 coordinator1 | SQL    | n_soft_parse                         | 21809
 coordinator1 | SQL    | n_hard_parse                         | 234111
 coordinator1 | SEC    | hsm_access_success_count             | 0
 coordinator1 | SEC    | hsm_access_failure_count             | 0
 datanode1    | XACT   | startup_max_xid                      | 16488
 datanode1    | XACT   | recent_local_xmin                    | 15805
 datanode1    | XACT   | recent_global_xmin                   | 15805
 datanode1    | STANDBY | standby_xmin                        | 0
 datanode1    | STANDBY | standby_redo_cleanup_xmin           | 0
 datanode1    | STANDBY | standby_redo_cleanup_xmin_lsn       | 0/0
 datanode1    | XACT   | local_csn_min                        | 6014225
 datanode1    | XACT   | replication_slot_xmin                | 0
 datanode1    | XACT   | replication_slot_catalog_xmin        | 0
 datanode1    | UNDO   | global_recycle_xid                   | 15805
 datanode1    | XACT   | global_frozen_xid                    | 0
 datanode1    | XACT   | recent_xmin                          | 15805
 datanode1    | XACT   | next_csn                             | 6014226
 datanode1    | HOTPATH | hotpatch_additional_info            |
 datanode1    | FULL_SQL| stmt_session_discard_records        | 0
 datanode1    | FULL_SQL| stmt_shm_flush_discard_records      | 0
 datanode1    | XACT   | idle_in_trans_timeout_records        | 0
```


版权所有 © 华为云计算技术有限公司

```
datanode1    | LOCK    | total_lock_usage                    | 0
datanode1    | LOCK    | total_lock_usage_rate               | 0.00%
datanode1    | LOCK    | fastpath_strong_relation_locks_count |
datanode1    | LOCK    | fastpath_failure_rate               | 0.00%
datanode1    | LOCK    | lock_acquire_lack_fastpath_count    | 0
datanode1    | SQL     | n_soft_parse                        | 0
datanode1    | SQL     | n_hard_parse                        | 8958
datanode1    | SEC     | hsm_access_success_count            | 0
datanode1    | SEC     | hsm_access_failure_count            | 0
(52 row)
```

- gs_get_single_node_group(object_type, object_name1, object_name2, queryonly)

  描述：用户可以根据该接口获取目标表或者目标Schema的默认单节点信息。

  参数：如表1-126所示。

  返回值类型：text

  表 1-126 gs_get_single_node_group 参数说明

参数	类型	描述	取值范围
object_type	text	对象类型	• schema • table
object_name1	text	模式名称	已存在的模式名称。
object_name2	text	表名称	• 表名称。 • NULL，默认值为NULL。
queryonly	bool	是否仅查询标志	• true，默认值为true。 • false。

  示例：

```
gaussdb=# CREATE SCHEMA s1 SET NODE GROUP sys_single_group1;

gaussdb=# CREATE TABLE single_table(c1 int, c2 int) TO GROUP sys_single_group2;

gaussdb=# SELECT gs_get_single_node_group('schema', 's1');
 gs_get_single_node_group
--------------------------
 sys_single_group1
(1 row)

gaussdb=# SELECT gs_get_single_node_group('table', 'public', 'single_table');
 gs_get_single_node_group
--------------------------
 sys_single_group2
(1 row)

gaussdb=# DROP TABLE single_table;
DROP TABLE
gaussdb=# DROP SCHEMA s1 CASCADE;
DROP SCHEMA
```

1.6.27 系统管理函数

版权所有 © 华为云计算技术有限公司

1.6.27.1 配置设置函数

配置设置函数用于查询以及修改运行时配置参数。

- current_setting(setting_name)

  描述：用于以查询形式获取setting_name的当前值。和SQL语句SHOW是等效的。

  返回值类型：text

  示例：

```
gaussdb=# SELECT current_setting('datestyle');

 current_setting
-----------------
 ISO, MDY
(1 row)
```

- set_working_grand_version_num_manually(tmp_version)

  描述：通过切换授权版本号来更新和升级GaussDB数据库的新特性。

  返回值类型：void

- shell_in(type)

  描述：用于处理shell类型的输入路由（即那些尚未填充的类型）。

  返回值类型：void

- shell_out(type)

  描述：用于处理shell类型的输出路由（即那些尚未填充的类型）。

  返回值类型：void

- set_config(setting_name, new_value, is_local)

  描述：用于设置指定的配置参数，并返回设置后的参数值。该函数不允许对ROLE参数进行设置。

  参数：

  - setting_name为要设置的参数名称。
  - new_value为要设置的参数值。
  - is_local决定设置后的参数值作用范围。
    - 如果is_local参数为true，则new_value将仅应用于当前事务。
    - 如果is_local参数为false，则与SQL语句SET等效，new_value将应用于当前会话。

  返回值类型：text

  示例：

```
gaussdb=# SELECT set_config('log_statement_stats', 'off', false);

 set_config
------------
 off
(1 row)
```

1.6.27.2 通用文件访问函数

通用文件访问函数提供了对数据库服务器上的文件的本地访问接口。只有数据库集群目录和log_directory目录里面的文件可以访问。使用相对路径访问集群目录里面的文

版权所有 © 华为云计算技术有限公司

件，以及匹配log_directory配置而设置的路径访问日志文件。只有数据库初始化用户才能使用这些函数。

- pg_ls_dir(dirname text)

  描述：列出目录中的文件。

  返回值类型：setof text

  备注：pg_ls_dir返回指定目录里面的除了特殊项“.”和“..”之外所有名称。

  示例：

```
gaussdb=# SELECT pg_ls_dir('./');
      pg_ls_dir
----------------------
 .gaussdb.conf.swp
 gaussdb.conf
 pg_tblspc
 PG_VERSION
 gs_ident.conf
 core
 server.crt
 pg_serial
 pg_twophase
 gaussdb.conf.lock
 pg_stat_tmp
 pg_notify
 pg_subtrans
 pg_ctl.lock
 pg_xlog
 pg_clog
 base
 pg_snapshots
 postmaster.opts
 postmaster.pid
 server.key.rand
 server.key.cipher
 pg_multixact
 pg_errorinfo
 server.key
 gs_hba.conf
 pg_replslot
 .gs_hba.conf.swp
 cacert.pem
 gs_hba.conf.lock
 global
 gaussdb.state
(32 rows)
```

- pg_read_file(filename text, offset bigint, length bigint)

  描述：返回一个文本文件的内容，从offset开始，最多返回length字节（如果先达到文件结尾，则小于这个数值）。如果offset是负数，则它是相对于文件结尾回退的长度。如果省略了offset和length，则返回整个文件。

  返回值类型：text

  示例：

```
gaussdb=# SELECT pg_read_file('postmaster.pid',0,100);
             pg_read_file
-----------------------------------------
 53078                                  +
 /srv/BigData/testdir/data1/coordinator+
 1500022474                             +
 8000                                   +
 /var/run/FusionInsight                 +
 localhost                              +
  2
(1 row)
```


版权所有 © 华为云计算技术有限公司

- pg_read_binary_file(filename text [, offset bigint, length bigint,missing_ok boolean[, compress boolean]])

  描述：返回一个二进制文件的内容，该函数的功能与pg_read_file类似，但其返回值为bytea类型，因此不会执行编码检查。可以结合convert_from函数，使用pg_read_binary_file函数读取指定编码的一个文件。

  返回值类型：bytea

  ```
  gaussdb=# SELECT convert_from(pg_read_binary_file('filename'), 'UTF8');
  ```

- pg_read_binary_file_blocks(path text, block_begin bigint, block_count bigint, bitmap bigint)

  描述：仅用于读取一个跨页透明压缩文件的指定页面区域的二进制内容，该函数仅支持数据库初始用户调用。

  返回值类型：bytea

- pg_stat_file(filename text)

  描述：返回一个文本文件的状态信息，其中包含：文件大小、最后访问时间戳、最后更改时间戳、最后文件状态修改时间戳以及标识传入参数是否为目录的Boolean值。

  返回值类型：record

  典型的用法：

  ```
  gaussdb=# SELECT * FROM pg_stat_file('filename');
  gaussdb=# SELECT (pg_stat_file('filename')).modification;
  ```

  示例：

  ```
  gaussdb=# SELECT convert_from(pg_read_binary_file('postmaster.pid'), 'UTF8');
              convert_from
  ---------------------------------------
   4881                                 +
   /srv/BigData/gaussdb/data1/coordinator+
   1496308688                           +
   25108                                +
   /opt/huawei/Bigdata/gaussdb/gaussdb_tmp +
   *                                    +
    25108001  43352069                  +

  (1 row)
  gaussdb=# SELECT * FROM pg_stat_file('postmaster.pid');

   size |         access         |      modification      |         change
  | creation | isdir
  ------+------------------------+------------------------+------------------------
  +----------+-------
    117 | 2017-06-05 11:06:34+08 | 2017-06-01 17:18:08+08 | 2017-06-01 17:18:08+08
  |          | f
  (1 row)
  gaussdb=# SELECT (pg_stat_file('postmaster.pid')).modification;
       modification
  ------------------------
   2017-06-01 17:18:08+08
  (1 row)
  ```

1.6.27.3 服务器信号函数

服务器信号函数向其他服务器线程发送控制信号。仅系统管理员有权执行以下函数。

- pg_cancel_backend(pid int)

  描述：取消一个后端线程正在执行的语句。

  返回值类型：Boolean

版权所有 © 华为云计算技术有限公司

备注：pg_cancel_backend向由pid标识的后端线程发送一个查询取消（SIGINT）信号。一个活动的后端线程的PID可以从PG_STAT_ACTIVITY视图（请参见**PG_STAT_ACTIVITY**）的pid字段找到，或者在服务器上用ps列出数据库线程。具有SYSADMIN权限的用户，后端线程所连接的数据库的属主，后端线程的属主或者继承了内置角色gs_role_signal_backend权限的用户有权使用该函数。

- pg_cancel_session(pid bigint, sessionid bigint)

  描述：线程池模式下，取消一个活跃状态会话正在执行的语句。

  返回值类型：Boolean

  备注：pg_cancel_session的入参可以通过pg_stat_activity中的pid字段和sessionid的字段查询，可以取消线程池模式下，活跃状态会话正在执行的语句。当入参pid和sessionid相同，且均为线程id时，功能和pg_cancel_backend相同。

- pg_cancel_invalid_query()

  描述：取消一个后端的无效查询。

  返回值类型：Boolean

  备注：只有系统管理员才有权限取消连接到降级的GTM的后端中运行的查询。

- pg_reload_conf()

  描述：该函数向服务器发送一个SIGHUP信号，触发所有服务器线程重新加载配置文件。

  返回值类型：Boolean

- pg_rotate_logfile()

  描述：滚动服务器的日志文件。

  返回值类型：Boolean

  备注：pg_rotate_logfile给日志文件管理器发送信号，使之立即切换到一个新的输出文件。这个函数只有在redirect_stderr用于日志输出的时候才有用，否则不会产生日志文件管理器子线程。

- pg_terminate_session(pid bigint, sessionid bigint)

  描述：线程池模式下，终止一个后台会话。

  返回值类型：Boolean

  备注：本函数的入参可以通过pg_stat_activity中的pid字段和sessionid的字段查询。具有SYSADMIN权限的用户、会话所连接的数据库的属主、会话的属主或者继承了内置角色gs_role_signal_backend权限的用户有权使用该函数。若执行该函数后未成功终止目标会话，则会强制关闭该会话和客户端的socket连接。

版权所有 © 华为云计算技术有限公司

须知

- 该函数根据入参pid与sessionid的关系区分两种执行模式，具体说明如下：
  - 参数相同时（pid=sessionid，且为线程id），此时函数按线程粒度执行终止操作，支持终止非线程池线程、活跃状态的线程池线程；不支持终止非活跃状态的线程池线程。
  - 参数不同时（pid ≠ sessionid），此时函数按会话粒度执行终止操作，支持终止活跃状态的会话，清理非活跃状态会话以及清理其对应的客户端socket连接。
- 在以下场景中，执行该函数可能无法保证立即终止目标线程或会话：
  - 线程或会话正处于等待锁的状态。
  - 因硬件故障（如存储异常、网络中断、资源不可用等）导致线程或会话阻塞或无响应。
  - 会话处于网络 I/O（如 recv/send）状态。
  - 系统CPU负载高、线程池满场景下，会话处于idle状态。

- pg_terminate_active_session_socket(pid int64, sessionid int64)

  描述：关闭一个活跃session和客户端的socket连接。

  返回值类型：Boolean

  备注：如果成功，函数返回true，否则返回false。具有SYSADMIN权限的用户，后端线程所连接的数据库的属主，后端线程的属主或者继承了内置角色gs_role_signal_backend权限的用户有权使用该函数。

- pg_terminate_backend(pid int)

  描述：终止一个后端线程。仅系统管理员和线程所有者可执行该函数。

  返回值类型：Boolean

  备注：如果成功，函数返回true，否则返回false。具有SYSADMIN权限的用户，后端线程所连接的数据库的属主，后端线程的属主或者继承了内置角色gs_role_signal_backend权限的用户有权使用该函数。若执行该函数后未成功终止目标会话，则会强制关闭该会话和客户端的socket连接。

须知

- 该函数支持终止非线程池线程及处于活跃状态的线程池线程，但无法对处于非活跃状态的线程池线程执行终止操作。
- 在以下场景中，执行该函数可能无法保证立即终止目标线程：
  - 线程正处于等待锁的状态。
  - 因硬件故障（如磁盘异常、资源不可用等）导致线程阻塞或无响应。

示例：

```
gaussdb=# SELECT pid from pg_stat_activity;
       pid
-----------------
 140657876268816
(1 rows)

gaussdb=# SELECT pg_terminate_backend(140657876268816);
 pg_terminate_backend
```


版权所有 © 华为云计算技术有限公司

```
----------------------
 t
(1 row)
```

- gs_cancel_invalid_gtm_query(cstring)

  描述：断开与指定地址的GTM连接的所有会话。函数参数为逗号分隔的字符串，分隔后的每个子串格式为IP[:port]（中括号表示可选）。

  返回值类型：void

  备注：只有系统管理员才有权限执行该函数。该函数仅由CM组件在特定流程中自动调用，不建议用户手动进行调用。

- terminate_session_has_temp_file()

  描述：中断所有正在打开临时文件的会话。

  返回值类型：void

  备注：仅系统管理员有权限执行该函数。该函数仅由CM组件在特定流程中自动调用，不建议用户手动进行调用。

1.6.27.4 备份恢复控制函数

备份控制函数

备份控制函数可帮助进行在线备份。

- pg_create_restore_point(name text)

  描述：为执行恢复创建一个命名点（需要管理员角色）。

  返回值类型：text

  备注：pg_create_restore_point创建了一个可以用作恢复目的、有命名的事务日志记录，并返回相应的事务日志位置。在恢复过程中，recovery_target_name可以通过这个名称定位对应的日志恢复点，并从此处开始执行恢复操作。避免使用相同的名称创建多个恢复点，因为恢复操作将在第一个匹配（恢复目标）的名称上停止。

- pg_current_xlog_location()

  描述：获取当前事务日志的写入位置。

  返回值类型：text

  备注：pg_current_xlog_location使用与前面那些函数相同的格式显示当前事务日志的写入位置。如果是只读操作，不需要系统管理员权限。

- pg_current_xlog_insert_location()

  描述：获取当前事务日志的插入位置。

  返回值类型：text

  备注：pg_current_xlog_insert_location显示当前事务日志的插入位置。插入点是事务日志在某个瞬间的“逻辑终点”，而实际的写入位置则是从服务器内部缓冲区写出时的终点。写入位置是可以从服务器外部检测到的终点，如果要归档部分完成事务日志文件，则该操作即可实现。插入点主要用于服务器调试目的。如果是只读操作，不需要系统管理员权限。

- gs_current_xlog_insert_end_location()

  描述：获取当前事务日志的插入位置。

  返回值类型：text

  备注：gs_current_xlog_insert_end_location显示当前事务日志的实际插入位置。

版权所有 © 华为云计算技术有限公司

- pg_start_backup(label text [, fast boolean])

  描述：开始执行在线备份（需要管理员角色、复制的角色或运维管理员角色打开operation_mode）。以gs_roach开头的label串为保留命名串，只能由内部备份工具GaussRoach使用。

  返回值类型：text

  备注：pg_start_backup接受一个用户定义的备份标签（通常这是备份转储文件存放地点的名称）。这个函数向数据库集群的数据目录写入一个备份标签文件，然后以文本方式返回备份的事务日志起始位置。该函数务必配合pg_stop_backup()使用，单独调用该函数会导致backup_label残留，后面再进行WAL回放时会根据backup_label读取checkpoint，如果该checkpoint对应的WAL文件已经被回收，则会导致数据库无法正常启动。

  ```
  gaussdb=# SELECT pg_start_backup('label_goes_here',true);
   pg_start_backup
  -----------------
   0/3000020
  (1 row)
  ```

- pg_stop_backup()

  描述：完成执行在线备份。需要管理员角色、复制的角色执行或运维管理员角色打开operation_mode。

  返回值类型：text

  备注：pg_stop_backup删除pg_start_backup创建的标签文件，并且在事务日志归档区里创建一个备份历史文件。这个历史文件包含给予pg_start_backup的标签、备份的事务日志起始与终止位置、备份的起始和终止时间。返回值是备份的事务日志终止位置。计算出终止位置后，当前事务日志的插入点将自动前进到下一个事务日志文件，结束的事务日志文件可以被立即归档从而完成备份。

- pg_switch_xlog()

  描述：切换到一个新的事务日志文件。需要管理员角色或运维管理员角色打开operation_mode。

  返回值类型：text

  备注：pg_switch_xlog移动到下一个事务日志文件，以允许将当前日志文件归档（假定使用连续归档）。返回值是刚完成的事务日志文件的事务日志结束位置+1。如果从最后一次事务日志切换以来没有活动的事务日志，则pg_switch_xlog不进行移动操作，直接返回当前事务日志文件的开始位置。

- pg_xlogfile_name(location text)

  描述：将事务日志的位置字符串转换为文件名。

  返回值类型：text

  备注：pg_xlogfile_name仅抽取事务日志文件名称。如果给定的事务日志位置恰好位于事务日志文件的交界上，这两个函数都返回前一个事务日志文件的名称。这对于管理事务日志归档来说是非常有利的，因为前一个文件是当前最后一个需要归档的文件。

- pg_xlogfile_name_offset(location text)

  描述：将事务日志的位置字符串转换为文件名并返回在文件中的字节偏移量。

  返回值类型：text,integer

  备注：可以使用pg_xlogfile_name_offset从前述函数的返回结果中抽取相应的事务日志文件名称和字节偏移量。例如：

  ```
  gaussdb=# SELECT * FROM pg_xlogfile_name_offset(pg_stop_backup());
  NOTICE:  pg_stop_backup cleanup done, waiting for required WAL segments to be archived
  NOTICE:  pg_stop_backup complete, all required WAL segments have been archived
  ```


版权所有 © 华为云计算技术有限公司

```
         file_name         | file_offset
---------------------------+-------------
000000010000000000000003 |         272
(1 row)
```

- pg_xlog_location_diff(location text, location text)

  描述：计算两个事务日志位置之间在字节上的区别。

  返回值类型：numeric

- pg_cbm_start_tracked_location()

  描述：用于查询cbm解析的起始lsn位置。

  返回值类型：text

- pg_cbm_tracked_location()

  描述：用于查询cbm解析到的lsn位置。

  返回值类型：text

- pg_cbm_get_merged_file(startLSNArg text, endLSNArg text)

  描述：用于将指定lsn范围之内的cbm文件合并成一个cbm文件，并返回合并完的cbm文件名。

  返回值类型：text

  备注：必须是系统管理员或运维管理员才能获取cbm合并文件。

- pg_cbm_get_changed_block(startLSNArg text, endLSNArg text)

  描述：用于将指定lsn范围之内的cbm文件合并成一个表，并返回表的各行记录。

  返回值类型：records

  备注：pg_cbm_get_changed_block返回的表字段包含：合并起始的lsn、合并截止的lsn、表空间oid、库oid、表的relfilenode、表的fork number、表是否为系统表、表是否被删除、表是否被创建、表是否被截断、表被截断后的页面数、有多少页被修改以及被修改的页号的列表。

- gs_cbm_get_changed_block(startLSNArg text, endLSNArg text)

  描述：用于将指定LSN范围之内的cbm文件合并成一个表，并返回表的各行记录。

  返回值类型：records

  备注：gs_cbm_get_changed_block返回的表字段包含：合并起始的LSN、合并截止的lsn、表空间oid、库oid、表的relfilenode、表的fork number、表是否为系统表、表是否被删除、表是否被创建、表是否被截断、表被截断后的页面数、有多少页被修改以及被修改的页号的列表，以及进行该操作的事务xid（仅记录创建、删除数据库，创建、删除表空间操作）。

- pg_cbm_recycle_file(targetLSNArg text)

  描述：删除不再使用的cbm文件，并返回删除后的第一条lsn。

  返回值类型：text

- pg_cbm_force_track(targetLSNArg text,timeOut int)

  描述：强制执行一次cbm追踪到指定的Xlog位置，并返回实际追踪结束点的Xlog位置。

  返回值类型：text

- pg_enable_delay_ddl_recycle()

  描述：开启延迟DDL功能，并返回开启点的Xlog位置。需要管理员角色或运维管理员角色打开operation_mode。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

- pg_disable_delay_ddl_recycle(barrierLSNArg text, isForce bool)

  描述：关闭延迟DDL功能，并返回本次延迟DDL生效的Xlog范围。需要管理员角色或运维管理员角色打开operation_mode。

  返回值类型：records

- pg_enable_delay_xlog_recycle()

  描述：开启延迟Xlog回收功能，cn修复使用。需要管理员角色或运维管理员角色打开operation_mode。

  返回值类型：void

- pg_disable_delay_xlog_recycle()

  描述：关闭延迟Xlog回收功能，cn修复使用。需要管理员角色或运维管理员角色打开operation_mode。

  返回值类型：void

- pg_cbm_rotate_file(rotate_lsn text)

  描述：等待cbm解析到rotate_lsn之后，强制切换文件，在build期间调用。

  返回值类型：void

- gs_roach_stop_backup(backupid text)

  描述：停止一个内部备份工具GaussRoach开启的备份。与pg_stop_backup系统函数类似，但更轻量。

  返回值类型：text，内容为当前日志的插入位置。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_roach_enable_delay_ddl_recycle(backupid name)

  描述：开启延迟DDL功能，并返回开启点的日志位置。与pg_enable_delay_ddl_recycle系统函数类似，但更轻量。并且，通过传入不同的backupid，可以支持并发打开延迟DDL。以gs_roach开头的backupid为内部名，只能由内部备份工具GaussRoach使用，执行时需要设置application_name='gs_roach'。

  返回值类型：text，内容为返回开启点的日志位置。

- gs_roach_disable_delay_ddl_recycle(backupid text)

  描述：关闭延迟DDL功能，并返回本次延迟DDL生效的日志范围。与pg_enable_delay_ddl_recycle系统函数类似，但更轻量。并且，通过传入不同的backupid，可以支持并发关闭延迟DDL功能。以gs_roach开头的backupid为内部名，只能由内部备份工具GaussRoach使用，执行时需要设置application_name='gs_roach'。

  返回值类型：records，内容为本次延迟DDL生效的日志范围。

- gs_roach_switch_xlog(request_ckpt bool)

  描述：切换当前使用的日志段文件，并且，如果request_ckpt 为true，则触发一个全量检查点。

  返回值类型：text，内容为切段日志的位置。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_block_dw_io(timeout int, identifier text)

  描述：阻塞双写页面刷盘。

  参数说明：

版权所有 © 华为云计算技术有限公司

  - timeout

    阻塞时长。

    取值范围：[0, 3600]（秒），0为阻塞时长为0。
  - identifier

    此次操作的标识。

    取值范围：字符串，不支持除大小写字母、数字以及下划线(_)以外的字符。

  返回值类型：Boolean

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_is_dw_io_blocked()

  描述：查看当前双写页面刷盘是否被阻塞，如果处于阻塞中则返回true。

  返回值类型：Boolean

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_pitr_advance_last_updated_barrier()

  描述：在PITR模式下，强制推进上次上传到OBS/NAS介质中的全局最大已归档恢复点到当前点，无入参。

  返回值类型：text

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。同时只能在分布式的第一个normal CN上使用才有效。返回值为当前推进到的最新本地最大已归档恢复点。

- gs_pitr_clean_local_barrier_files('delete_timestamp')

  描述：清理本地缓存的barrier记录文件。

  参数范围：delete_timestamp参数类型为text，为linux时间戳，长度为10位。

  返回值类型：text

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。返回的结果是删除后本地最老barrier文件的开始时间戳。

- gs_get_barrier_lsn(barrier_name text)

  描述：获取备份创建的barrier对应的lsn。

  返回值类型：text

  备注：当前入参仅支持gs_roach_full和gs_roach_inc。调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_gbr_relation_associated_filenode(schemaName name, tableName name)

  描述：返回与输入table相关的全部索引、sequence、分区、toast、toast index的relfilenode。

  返回值类型：records

  备注：gs_gbr_relation_associated_filenode返回的字段包含文件类型relkind、文件所在namespace、文件对应的relation name、文件所在的database oid、文件所在的tablespace oid以及文件的relfilenode。

- gs_pitr_stop_archive_manually(archive_slot_name cstring)

  描述：触发单个CN/DN节点停止正在进行的归档任务（仅对介质REMOTE有效），函数返回停止归档是否成功。

版权所有 © 华为云计算技术有限公司

返回值类型：text

说明：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。函数为内部调用函数，不允许用户调用。

- pg_create_physical_replication_slot_extern(slotname text, dummy_standby bool, extra_content text, need_recycle_xlog bool)

  描述：创建OBS/NAS归档槽或备份槽。对于归档槽，slotname 为归档槽/恢复槽的slotname ，主备必须使用同一个slotname 。dummy_standby是预留参数。extra_content包含了归档槽的一些信息。对于OBS归档槽，其格式为"OBS;obs_server_ip;obs_bucket_name;obs_ak;obs_sk;archive_path;is_recovery;is_vote_replicate"，OBS表示归档槽的归档的介质，obs_server_ip为obs的IP，obs_bucket_name为obs的桶名，obs_ak为obs的ak，obs_sk为obs的sk，archive_path为归档的路径，is_recovery标志是归档槽还是恢复槽，0表示是归档槽，1表示是恢复槽。is_vote_replicate标志是否是投票副本优先，0表示同步备机归档优先，1表示投票副本归档优先，当前版本该字段为预留字段，暂未适配。对于NAS归档槽，其格式为"NAS;archive_path;is_recovery;is_vote_replicate"，相比OBS归档槽，缺少了OBS相关的配置信息，其余字段意义相同。

  如果是不指定OBS或NAS介质的话，默认指定的是OBS归档槽，其extra_content格式为"obs_server_ip;obs_bucket_name;obs_ak;obs_sk;archive_path;is_recovery;is_vote_replicate"。

  need_recycle_xlog标志创建归档槽时是否回收旧的归档日志，true表示回收，false表示不回收。

  对于备份槽，slotname为"gs_roach_"开头的槽，一般由备份流程创建。dummy_standby是预留参数。extra_content可以为NULL或者合法的lsn字符串。need_recycle_xlog不影响备份槽的功能。如果extra_content传入NULL，使用checkpoint redo点作为备份槽的初始位置。如果extra_content传入合法lsn字符串，且该lsn比当前checkpoint redo点小，则使用该lsn作为备份槽的初始位置；如果字符串为非法lsn，则创建失败。

  返回值类型：records包含slotname和xlog_position。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。目前不支持创建多归档槽。

  例如：

  创建OBS归档槽：

  其中obs_server_ip、obs_bucket_name、obs_ak、obs_sk为OBS的IP地址、桶名、AK、SK信息，gaussdb_uuid/dn1为日志在OBS对应桶下的归档路径，使用时需要根据实际信息进行替换。

  ```
  gaussdb=# SELECT * FROM pg_create_physical_replication_slot_extern('uuid', false,
  'OBS;obs_server_ip;obs_bucket_name;obs_ak;obs_sk;gaussdb_uuid/dn1;0;0', false);
   slotname | xlog_position
  ----------+---------------
   uuid     |
  (1 row)
  ```

  创建NAS归档槽：

  其中/data/nas/media/gaussdb_uuid/dn1为日志在NAS上对应的归档路径，使用时需要根据实际信息进行替换。

  ```
  gaussdb=# SELECT * FROM pg_create_physical_replication_slot_extern('uuid', false, 'NAS;/data/nas/
  media/gaussdb_uuid/dn1;0;0', false);
   slotname | xlog_position
  ----------+---------------
   uuid     |
  ```

  创建备份槽：

版权所有 © 华为云计算技术有限公司

使用pg_create_physical_replication_slot_extern函数创建备份槽时，需要先执行set application_name='gs_roach';。因为该函数属于备份相关的运维操作，不建议直接使用。

```
gaussdb=# SELECT * FROM pg_catalog.pg_create_physical_replication_slot_extern('gs_roach_full',
false, NULL, false);
   slotname    | xlog_position
---------------+---------------
 gs_roach_full | 0/3F883530
```

- gs_set_obs_delete_location(delete_location text)

  描述：设置obs归档日志可删除的位置。delete_location实际为Log Sequence Number（LSN），该位置之前的日志已经完成回放并且落盘，可以在obs上进行删除。

  返回值类型：xlog_file_name text，表明此次可删除点所在的日志文件名。无论obs删除是否成功，该值都会正常返回。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

  ```
  gaussdb=# SELECT gs_set_obs_delete_location('0/54000000');
   gs_set_obs_delete_location
  ----------------------------
   000000010000000000000054_00
  (1 row)
  ```

- gs_set_obs_delete_location_with_slotname(cstring, cstring)

  描述：设置指定归档槽obs归档日志可删除的位置。第一个参数实际为Log Sequence Number（LSN），该位置之前的日志已经完成回放并且落盘，可以在obs上进行删除，第二个参数为归档槽的名称。

  返回值类型：xlog_file_name text，表明此次可删除点所在的日志文件名。无论obs删除是否成功，该值都会正常返回。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_get_global_barrier_status()

  描述：gs_get_global_barrier_status用以查询已在obs完成归档的最新global barrier。

  返回值类型：text

  global_barrier_id：全局最新barrier ID。

  global_achive_barrier_id：全局最新归档barrier ID。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_get_global_barriers_status()

  描述：gs_get_global_barriers_status用以查询已在obs完成归档的最新global barrier。

  返回值类型：text

  slot_name：槽位名。

  global_barrier_id：全局最新barrier ID。

  global_achive_barrier_id：全局最新归档barrier ID。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_set_archive_standby_name_on_primary(text)

版权所有 © 华为云计算技术有限公司

描述：gs_set_archive_standby_name_on_primary用来指定归档备机，设置后将由指定的备机执行归档操作，DCF模式下不支持指定备机。输入为备机名称，如：dn_6002。

返回值类型：BOOLEAN

true表示设置成功，false表示设置失败。

恢复控制函数

恢复信息函数提供了当前备机状态的信息。这些函数可能在恢复期间或正常运行中执行。

- pg_is_in_recovery()

  描述：如果恢复仍然在进行中则返回true。

  返回值类型：Boolean

- pg_last_xlog_receive_location()

  描述：获取最后接收事务日志的位置并通过流复制将其同步到磁盘。当流复制正在进行时，事务日志将持续递增。如果恢复已完成，则最后一次获取的WAL记录会被静态保持并在恢复过程中同步到磁盘。如果流复制不可用，或还没有开始，这个函数返回NULL。

  返回值类型：text

- pg_last_xlog_replay_location()

  描述：获取最后一个事务日志在恢复时重放的位置。如果恢复仍在进行，事务日志将持续递增。如果已经完成恢复，则将保持在恢复期间最后接收WAL记录的值。如果未进行恢复但服务器正常启动时，则这个函数返回NULL。

  返回值类型：text

- pg_last_xact_replay_timestamp()

  描述：获取最后一个事务在恢复时重放的时间戳。这是为在主节点上生成事务提交或终止WAL记录的时间。如果在恢复时没有事务重放，则这个函数返回NULL。如果恢复仍在进行，则事务日志将持续递增。如果恢复已经完成，则将保持在恢复期间最后接收WAL记录的值。如果服务器无需恢复就已正常启动，则这个函数返回NULL。

  返回值类型：timestamp with time zone

恢复控制函数控制恢复的线程。这些函数可能只在恢复时被执行。

- pg_is_xlog_replay_paused()

  描述：如果恢复暂停则返回true。

  返回值类型：Boolean

- pg_xlog_replay_pause()

  描述：立即暂停恢复。

  返回值类型：void

- pg_xlog_replay_resume()

  描述：如果恢复处于暂停状态，则重新启动。

  返回值类型：void

- gs_get_active_archiving_standby()

  描述：查询同一分片内归档备机的信息。返回备机名，备机归档位置和已归档日志个数。

版权所有 © 华为云计算技术有限公司

返回值类型：text，int

- gs_pitr_get_warning_for_xlog_force_recycle()

  描述：查询开启归档后是否因归档槽不推进日志大量堆积导致日志被回收。

  返回值类型：Boolean

- gs_pitr_clean_history_global_barriers(stop_barrier_timestamp cstring)

  描述：清理指定时间之前所有barrier记录。返回时间最远的barrier记录。入参为cstring类型，linux时间戳。需要管理员角色或运维管理员角色执行。

  返回值类型：text

- gs_pitr_archive_slot_force_advance(stop_barrier_timestamp cstring)

  描述：强制推进归档槽，并清理不需要的barrier记录。返回新的归档槽位置。入参为cstring类型，linux时间戳。需要管理员角色或运维管理员角色执行。

  返回值类型：text

- gs_recent_barrier_buffer_info(start_time text, end_time text)

  描述：根据用户输入的时间范围，进行相应的barrier信息查询，获取time_stamp、CSN、LSN和standard_time。

  返回值类型：records

  说明：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限。输入参数start_time和end_time采用“年-月-日 时间”格式，其中时间采用clock格式。查询最大时间跨度为1天，超出跨度约束，根据查询起始时间将结束时间自动转换为极限边界进行查询。例如：

```
gaussdb=# SELECT * FROM gs_recent_barrier_buffer_info('2024-01-15 23:27:50', '2024-01-15
23:28:00');
 timestamp |       lsn        |   csn    |    standard_time
-----------+------------------+----------+---------------------
 1705332470 | 00000000/15FFBBA0 | 41020421 | 2024-01-15 23:27:50
 1705332471 | 00000000/15FFBDF0 | 41020422 | 2024-01-15 23:27:51
 1705332472 | 00000000/15FFC058 | 41020423 | 2024-01-15 23:27:52
 1705332472 | 00000000/15FFC0F8 | 41020424 | 2024-01-15 23:27:52
 1705332473 | 00000000/15FFC348 | 41020425 | 2024-01-15 23:27:53
 1705332474 | 00000000/15FFC598 | 41020426 | 2024-01-15 23:27:54
 1705332475 | 00000000/15FFC638 | 41020427 | 2024-01-15 23:27:55
 1705332476 | 00000000/15FFC888 | 41020428 | 2024-01-15 23:27:56
 1705332476 | 00000000/15FFDC80 | 41020433 | 2024-01-15 23:27:56
 1705332477 | 00000000/15FFDD20 | 41020434 | 2024-01-15 23:27:57
 1705332478 | 00000000/15FFDF70 | 41020435 | 2024-01-15 23:27:58
 1705332479 | 00000000/15FFE1D8 | 41020436 | 2024-01-15 23:27:59
 1705332480 | 00000000/15FFE278 | 41020437 | 2024-01-15 23:28:00
 1705332480 | 00000000/15FFE4C8 | 41020438 | 2024-01-15 23:28:00
(14 rows)
```

- gs_show_obs_media_files(slot_name cstring, src cstring, offset int32, limit int32)

  描述：根据用户输入的归档槽（slot_name）和OBS目录地址（src），查询OBS文件列表。

  返回值类型：records

  说明：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限。Offset为查询结果偏移，limit为输出最大行数，查询src下所有文件。例如：

```
gaussdb=# SELECT gs_show_obs_media_files('ssh','cn_5001/pg_xlog',1, 5);
                                gs_show_obs_media_files
-----------------------------------------------------------------------------------------------------
 (wstdist_ssh/archive/cn_5001/pg_xlog/
000000010000000000000010_01_01_00000000_00000000_00000003)
 (wstdist_ssh/archive/cn_5001/pg_xlog/
000000010000000000000010_02_01_00000000_00000000_00000003)
 (wstdist_ssh/archive/cn_5001/pg_xlog/
```


版权所有 © 华为云计算技术有限公司

```
000000010000000000000010_03_01_00000000_00000000_00000003)
 (wstdist_ssh/archive/cn_5001/pg_xlog/
000000010000000000000011_00_01_00000000_00000000_00000003)
 (wstdist_ssh/archive/cn_5001/pg_xlog/
000000010000000000000011_01_01_00000000_00000000_00000003)
(5 rows)
```

- gs_upload_obs_media_file(slot_name cstring, src cstring, dest cstring, is_forced bool)

  描述：根据用户输入的归档槽（slot_name）、上传文件原地址（src）、OBS地址（dest）和是否强制上传（is_forced），上传OBS文件。

  返回值类型：void

  说明：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限。原文件目录仅允许为$GAUSSLOG目录。例如：

  ```
  gaussdb=# SELECT * FROM gs_upload_obs_media_file('ssh', '/data/gauss/log/stwang/test/
  000000010000000000000019_02_01_00000000_00000000_00000003', 'cn_5001/pg_xlog/
  000000010000000000000019_02_01_00000000_00000000_00000003', true);
   gs_upload_obs_media_file
  ---------------------------

  (1 row)
  ```

- gs_download_obs_media_file(slot_name cstring, src cstring, dest cstring)

  描述：根据用户输入的归档槽（slot_name）、下载原地址（src）和本地目标地址（dest），下载OBS文件。

  返回值类型：void

  说明：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限。下载目录仅允许为$GAUSSLOG目录。例如：

  ```
  gaussdb=# SELECT * FROM gs_download_obs_media_file('ssh','cn_5001/pg_xlog/
  000000010000000000000019_02_01_00000000_00000000_00000003','/data/gauss/log/stwang/test');
   gs_download_obs_media_file
  -----------------------------

  (1 row)
  ```

当恢复暂停时，没有发生数据库更改。如果是在热备里，所有新的查询将看到一致的数据库快照，并且不会有进一步的查询冲突产生，直到恢复继续。

如果不能使用流复制，则暂停状态将无限的延续。当流复制正在进行时，将连续接收WAL记录，最终将填满可用磁盘空间，这个进度取决于暂停的持续时间，WAL生成的速度和可用的磁盘空间。

1.6.27.5 双集群容灾控制函数

- gs_streaming_dr_in_switchover()

  描述：基于流式复制的异地容灾解决方案中主集群在执行计划内switchover过程中截断业务的接口。

  返回值类型：Boolean，表明此次业务截断是否成功，是否可以正常进行switchover流程。

- gs_set_dr_dest_name()

  描述：用于配置日志发送端集群的所有日志接收端集群实例ID，入参为与日志发送端集群直接连接的所有日志接收端集群实例ID拼接成的字符串，并以','作为间隔符；设置成功返回t，否则返回f。

  入参类型：text，当入参为0时表示未配置日志接收端集群实例ID信息。容灾流程中会自动设置，不建议用户进行修改。

版权所有 © 华为云计算技术有限公司

须知

日志接收端集群实例ID不可为“invalid_cluster_uuid”，且字符长度不可超过36位。

返回值类型：boolean

示例

```
gaussdb=# SELECT * FROM gs_set_dr_dest_name('cluster1,cluster2');
 gs_set_dr_dest_name
---------------------
 t
(1 row)
```

- gs_get_dr_dest_name()

  描述：用于查看通过gs_set_dr_dest_name配置的所有日志接收端集群实例ID信息。

  返回值类型：text，返回值为0时表示配置的日志接收端集群实例ID信息为空。

  示例

```
gaussdb=# SELECT * FROM gs_get_dr_dest_name();
 gs_get_dr_dest_name
---------------------
 cluster1,cluster2
(1 row)
```

- gs_set_hadr_link_info()

  描述：用于容灾链路信息的设置，入参是JSON格式的text；设置成功返回"success"，失败则返回"failed"。容灾流程中会自动设置，不建议用户进行修改。

  返回值类型：text

表 1-127 gs_set_hadr_link_info 参数说明

参数名	类型	描述
meta_version	string	容灾链路JSON格式的版本号。
version	string	容灾链路信息的版本号。
replication_info	string	存放容灾链路信息，包含多条链路信息的JSON数组。
dest_name	string	灾备集群实例ID。
dest_type	string	控制容灾链路的类型。 取值类型：枚举值 取值范围：physical_standby_cluster/logical_standby_cluster/gaussstor 默认值：physical_standby_cluster

版权所有 © 华为云计算技术有限公司

参数名	类型	描述
mode	string	控制容灾链路日志同步的模式。 取值类型：枚举值 取值范围： ● sync：该链路同步等待日志接收端备机持久化redo日志到磁盘，确保无数据丢失，会影响主机性能。 ● semisync：该链路优先同步传输，若发生断链超时或RPO异常时则降级为异步。 ● async：异步传输，优先主机性能。 默认值：async
net_timeout	string	半同步模式下超时阈值。 取值范围：[0,max] 取值单位：秒 默认值：0
rpo_threshold	int	半同步模式下RPO阈值。 取值范围：[0,max] 取值单位：秒 默认值：0
compression	string	控制是否打开压缩。 取值类型：枚举值 取值范围：enable/disable 默认值：enable
compression_algo_setting	string	控制压缩算法，当前不支持修改，此参数使用默认值"lz4,1"。
replication_rate_limit	int/string	控制容灾链路的限制速度，带单位：K、M、G，例如：1024K、1M、10M、2048G。 取值单位：Byte 取值范围：[-1、0、1MB ~ 2TB] ● 整型：-1、0、1048576 ~ 2199023255552 ● 字符串：-1、0、1048576 ~ 2199023255552 默认值：-1
rate_burst	int	控制允许单次发送的最大数据包大小，为replication_rate_limit的倍数关系。 取值范围：[1, 5] 默认值：2
conn_info	string	包含连接串和访问权限信息。详见下面示例的具体内容。

版权所有 © 华为云计算技术有限公司

示例

```
gaussdb=# SELECT gs_set_hadr_link_info('{"meta_version":"1","version":"3","replication_info":
[{"dest_name":"clusterb","dest_type":"physical_standby_cluster","mode":"async","net_timeout":0,"rpo_t
hreshold":0,"compression":"enable","compression_algo_setting":"lz4,1","replication_rate_limit":-1,"rate_
burst":2,"conn_info":{"shards":{"replconninfo":{"shard_1":{"localinfo":{"dn_6001":
{"h":"127.0.0.1","p":40001,"hbp":40005},"dn_6002":{"h":"127.0.0.1","p":40001,"hbp":40005},"dn_6003":
{"h":"127.0.0.1","p":40001,"hbp":40005}},"remoteinfo":[{"h":"127.0.0.1","p":40001,"hbp":40005},
{"h":"127.0.0.1","p":40001,"hbp":40005},{"h":"127.0.0.1","p":40001,"hbp":40005}]},"shard_2":
{"localinfo":{"dn_6006":{"h":"127.0.0.1","p":40021,"hbp":40025},"dn_6004":
{"h":"127.0.0.1","p":40021,"hbp":40025},"dn_6005":
{"h":"127.0.0.1","p":40021,"hbp":40025}},"remoteinfo":[{"h":"127.0.0.1","p":40021,"hbp":40025},
{"h":"127.0.0.1","p":40021,"hbp":40025},{"h":"127.0.0.1","p":40021,"hbp":40025}]},"shard_3":
{"localinfo":{"dn_6008":{"h":"127.0.0.1","p":40041,"hbp":40045},"dn_6009":
{"h":"127.0.0.1","p":40041,"hbp":40045},"dn_6007":
{"h":"127.0.0.1","p":40041,"hbp":40045}},"remoteinfo":[{"h":"127.0.0.1","p":40041,"hbp":40045},
{"h":"127.0.0.1","p":40041,"hbp":40045},{"h":"127.0.0.1","p":40041,"hbp":40045}]}},"hba":
[{"type":"host","database":"all","user":"rdsAdmin","method":"trust"},
{"type":"host","database":"replication","user":"stream_user","method":"sha256"}]},"cn":{"replconninfo":
{"cn_5001":{"h":"127.0.0.1","p":9081,"hbp":9085,"remoteinfo":
[{"h":"127.0.0.1","p":9081,"hbp":9085}]},"cn_5002":{"h":"127.0.0.1","p":9081,"hbp":9085,"remoteinfo":
[{"h":"127.0.0.1","p":9081,"hbp":9085}]},"cn_5003":{"h":"127.0.0.1","p":9081,"hbp":9085,"remoteinfo":
[{"h":"127.0.0.1","p":9081,"hbp":9085}]}},"hba":
[{"type":"host","database":"all","user":"rdsAdmin","method":"trust"},
{"type":"host","database":"replication","user":"stream_user","method":"sha256"}]}}}]} ');
 gs_set_hadr_link_info
-----------------------
 success
(1 row)
```

- gs_get_hadr_link_info()

  描述：用于查看当前所有容灾链路的信息。

  返回值类型：record

  示例

```
gaussdb=# SELECT * FROM gs_get_hadr_link_info();
-[ RECORD 1 ]------------
+-------------------------------------------------------------------------------------------------------------------------------------------------------
--------
dest_name               | clusterb
dest_type               | physical_standby_cluster
mode                    | semisync
net_timeout             | 60
rpo_threshold           | 10
compression             | enable
compression_algo_setting | lz4,1
replication_rate_limit  | 0
rate_burst              | 2
replconninfo            | localhost=127.0.0.1 localport=20001 localheartbeatport=20005
remotehost=127.0.0.1 remoteport=20025 remoteheartbeatport=20029
                        | localhost=127.0.0.1 localport=20001 localheartbeatport=20005
remotehost=127.0.0.1 remoteport=20033 remoteheartbeatport=20037
                        | localhost=127.0.0.1 localport=20001 localheartbeatport=20005
remotehost=127.0.0.1 remoteport=20041 remoteheartbeatport=20045
                        |
hba                     | local all all trust
                        | local all all trust
                        | local all all trust
                        |
```

- gs_get_hadr_link_version()

  描述：用于获取容灾链路信息的版本号。

  返回值类型：int8

  示例

```
gaussdb=# SELECT * FROM gs_get_hadr_link_version();
 link_info_version
-------------------
```


版权所有 © 华为云计算技术有限公司

```
          100
(1 row)
```

- gs_get_hadr_sync_status()
  描述：用于查看当前所有容灾链路日志同步的状态信息。
  返回值类型：record

  表 1-128 gs_get_hadr_sync_status 参数说明

参数类型	参数名	类型	描述
输出参数	dest_name	text	容灾链路日志接收端集群ID。
输出参数	sync_state	text	当前容灾链路实际的同步状态。
输出参数	current_rpo	int	当前容灾链路的RPO时间（单位：秒）。
输出参数	sender_sent_location	text	容灾链路日志发送端发送日志位置。
输出参数	sender_write_location	text	容灾链路日志发送端写入到文件系统的日志位置。
输出参数	sender_flush_location	text	容灾链路日志发送端flush磁盘的日志位置。
输出参数	sender_replay_location	text	容灾链路日志发送端回放日志位置。
输出参数	receiver_received_location	text	容灾链路日志接收端接收日志位置。
输出参数	receiver_flush_location	text	容灾链路日志接收端日志flush磁盘位置。
输出参数	last_disconnect_time	timestamptz	最近一次容灾链路断链的时间。
输出参数	last_upgrade_to_sync_time	timestamptz	半同步模式下，最近一次升级为同步模式的时间。
输出参数	last_downgrade_to_async_time	timestamptz	半同步模式下，最近一次降级为异步模式的时间。
输出参数	mode	text	容灾链路日志同步模式的配置值。
输出参数	net_timeout	text	容灾链路半同步模式的超时阈值。
输出参数	rpo_threshold	text	容灾链路半同步模式的RPO阈值。

示例

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM gs_get_hadr_sync_status();
-[ RECORD 1 ]----------------+-----------------------------
dest_name                    | clusterb
sync_state                   | Sync
current_rpo                  | 0
sender_sent_location         | 0/11380B20
sender_write_location        | 0/11380B20
sender_flush_location        | 0/11380B20
sender_replay_location       | 0/11380B20
receiver_received_location   | 0/11380B20
receiver_flush_location      | 0/11380B20
last_disconnect_time         | 2026-01-15 20:35:42.18516+08
last_upgrade_to_sync_time    | 2026-01-15 21:29:55.147578+08
last_downgrade_to_async_time | 2026-01-15 20:36:42.187545+08
mode                         | semisync
net_timeout                  | 60
rpo_threshold                | 10
```

- gs_stat_get_replication_limit()

  描述：用于查询当前集群所有链路的限流配置参数值，最近1s、最近5s和最近60s的平均带宽使用情况。

  返回值类型：record

  表 1-129 gs_stat_get_replication_limit 参数说明

参数类型	参数名	类型	描述
输出参数	dest_name	text	该发送链路的目标集群名称。
输出参数	replication_rate_limit	text	该链路限流配置的带宽上限。
输出参数	rate_burst	text	该链路限流配置的突发流量为带宽上限的倍数，允许瞬时产生的最大流量。
输出参数	current_usage	text	最近1s链路使用的平均带宽统计，单位Byte/s。
输出参数	average_usage_5s	text	最近5s链路使用的平均带宽统计，单位Byte/s。
输出参数	average_usage_60s	text	最近60s链路使用的平均带宽统计，单位Byte/s。

  示例

```
gaussdb=# SELECT * FROM gs_stat_get_replication_limit();
-[ RECORD 1 ]----------+---------
dest_name              | clusterb
replication_rate_limit | -1
rate_burst             | 2
current_usage          | 0
average_usage_5s       | 0
average_usage_60s      | 0
-[ RECORD 2 ]----------+---------
dest_name              | clusterc
replication_rate_limit | -1
rate_burst             | 2
```


版权所有 © 华为云计算技术有限公司

```
current_usage          | 0
average_usage_5s       | 0
average_usage_60s      | 0
```

1.6.27.6 双集群容灾查询函数

- gs_get_local_barrier_status()

  描述：基于流式复制的异地容灾特性开启后，主集群和灾备集群进行日志同步，通过barrier日志在主集群的落盘，在灾备集群的回放来确定主集群归档日志进度与灾备集群日志回放进度。gs_get_local_barrier_status用于查询灾备集群每个节点当前的日志回放情况。

  返回值类型：text

  barrier_id：灾备集群某节点当前回放到的最新barrier ID。

  barrier_lsn：灾备集群某节点当前回放到的最新barrier ID的Log Sequence Number（LSN）。

  archive_lsn：灾备集群某节点当前已获得归档日志的位置，该参数当前未生效。

  flush_lsn：灾备集群某节点当前已完成刷盘日志位置。

- gs_hadr_in_recovery()

  描述：基于流式复制的异地容灾特性开启后，查询当前节点是否处于基于目标barrier的日志恢复中，还在恢复中返回true。只有完成日志恢复，才会启动switchover流程中的灾备集群升为生产集群的步骤，需要系统管理员角色执行。

  返回值类型：Boolean

  说明

  该函数只有在容灾集群启动计划内switchover时使用。

- gs_streaming_dr_get_switchover_barrier()

  描述：基于流式复制的异地容灾解决方案中，查询灾备集群参与容灾的CN与首备DN实例是否已接收到switchover barrier日志并完成回放，已完成返回true。灾备集群只有在所有DN实例都完成switchover barrier日志回放，才会启动switchover流程中的灾备数据库实例升为生产数据库实例的步骤（需要系统管理员角色）。

  返回值类型：Boolean

  备注：该函数只有在流式容灾解决方案中容灾数据库实例启动计划内switchover时使用。

- gs_streaming_dr_service_truncation_check()

  描述：基于流式复制的异地容灾解决方案中，查询主集群参与容灾的CN与主DN实例是否已完成switchover barrier日志发送，已完成返回true。只有完成日志发送，才会启动switchover流程中的生产数据库实例降为灾备数据库实例的步骤（需要系统管理员角色）。

  返回值类型：Boolean

  备注：该函数只有在容灾数据库实例启动计划内switchover时使用。

- gs_hadr_local_rto_and_rpo_stat()

  描述：显示流式容灾的本地节点数据库实例和灾备数据库实例日志流控信息（如果在没有参与流式容灾的节点执行，如备DN或部分CN节点，则可能返回空）。

  返回值类型：record，具体各个字段的类型和含义如**表1-130**所示。

版权所有 © 华为云计算技术有限公司

表 1-130 gs_hadr_local_rto_and_rpo_stat 参数说明

参数	类型	描述
hadr_sender_node_name	text	节点的名称，包含主数据库实例和备数据库实例首备。
hadr_receiver_node_name	text	备数据库实例首备名称。
source_ip	text	主数据库实例主DN IP地址。
source_port	int	主数据库实例主DN通信端口。
dest_ip	text	备数据库实例首备DN IP地址。
dest_port	int	备数据库实例首备DN通信端口。
current_rto	int	流控的信息，当前主备数据库实例的日志RTO时间（单位：秒）。
target_rto	int	流控的信息，目标主备数据库实例间的RTO时间（单位：秒）。
current_rpo	int	流控的信息，当前主备数据库实例的日志RPO时间（单位：秒）。
target_rpo	int	流控的信息，目标主备数据库实例间的RPO时间（单位：秒）。
rto_sleep_time	int	RTO流控信息，为了达到目标RTO，预期主机walsender所需要的睡眠时间（单位：微秒）。
rpo_sleep_time	int	RPO流控信息，为了达到目标RPO，预期主机xlogInsert所需要的睡眠时间（单位：微秒）。

- gs_hadr_remote_rto_and_rpo_stat()

  描述：显示流式容灾的其他所有分片或CN数据库实例和灾备数据库实例日志流控信息（一般在CN节点执行；如果在DN节点执行，可能返回为空）。

  返回值类型：record，具体各个字段的类型和含义如**表1-131**所示。

表 1-131 gs_hadr_remote_rto_and_rpo_stat

参数	类型	描述
hadr_sender_node_name	text	节点的名称，包含主数据库实例和备数据库实例首备。
hadr_receiver_node_name	text	备数据库实例首备名称。
source_ip	text	主数据库实例主DN IP地址。
source_port	int	主数据库实例主DN通信端口。
dest_ip	text	备数据库实例首备DN IP地址。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
dest_port	int	备数据库实例首备DN通信端口。
current_rto	int	流控的信息，当前主备数据库实例的日志RTO时间（单位：秒）。
target_rto	int	流控的信息，目标主备数据库实例间的RTO时间（单位：秒）。
current_rpo	int	流控的信息，当前主备数据库实例的日志RPO时间（单位：秒）。
target_rpo	int	流控的信息，目标主备数据库实例间的RPO时间（单位：秒）。
rto_sleep_time	int	RTO流控信息，为了达到目标RTO，预期主机walsender所需要的睡眠时间（单位：微秒）。
rpo_sleep_time	int	RPO流控信息，为了达到目标RPO，预期主机xlogInsert所需要的睡眠时间（单位：微秒）。

- gs_is_wal_truncate_complete()

  描述：用于查询灾备集群升主时，日志在线截断是否完成，已完成返回true。每次查询到已完成后会重置该值。

  返回值类型：Boolean

  备注：该函数只有在灾备集群升主时使用。

- gs_hadr_local_replay_delay_stat()

  描述：灾备集群延迟回放场景下，显示灾备集群barrier接收、barrier回放以及延迟时间等信息。

  返回值类型：record

  表 1-132 gs_hadr_local_replay_delay_stat 返回参数说明

参数名	类型	描述
node_name	text	实例名。
receive_barrier	text	接收携带时间戳的日志信息。
receive_barrier_lsn	text	接收携带时间戳的日志LSN。
receive_time	timestamp with time zone	解析接收时间戳。
replay_barrier	text	回放携带时间戳的日志信息。
replay_barrier_lsn	text	回放携带时间戳的日志LSN。
replay_time	timestamp with time zone	解析回放时间戳。

版权所有 © 华为云计算技术有限公司

参数名	类型	描述
delay_replay	int8	延迟回放配置值。
current_remain_interval	int8	延迟实际间隔。

示例：

```
gaussdb=# SELECT * FROM gs_hadr_local_replay_delay_stat();
  node_name   |             receive_barrier             | receive_barrier_lsn |        receive_time        |
replay_barrier              | replay_barrier_lsn |         replay_time         | delay_replay |
current_remain_interval
--------------+-----------------------------------------+---------------------+----------------------------
+-----------------------------------------+--------------------+----------------------------+--------------
+------------------------
 dn_6001_6002 | csn_00000000000000023117_1772440335494 | 00000000/1225B4B0   | 2026-03-02
16:32:15.494+08 | csn_00000000000000023111_1772440331492 | 00000000/1225B320  | 2026-03-02
16:32:11.492+08 |            0 |                        4
(1 row)
```

- disaster_cluster_run_mode()

  描述：用于在连接时查询连接的是主集群还是灾备集群。查询该函数主集群返回0，灾备集群返回1。

  返回值类型：int

  示例：

  ```
  gaussdb=# SELECT pg_catalog.disaster_cluster_run_mode();
  ----------------
              0
  (1 row)
  ```

- pgxc_disaster_read_node()

  描述：用于在连接时返回容灾集群相关的集群信息。

  返回值类型：record，具体各个字段的类型和含义如**表1-133**所示。

  备注：主集群查询会报错，仅用于容灾集群查询。

  表 1-133 pgxc_disaster_read_node

参数	类型	描述
node_name	name	节点的名称。
node_type	char	节点的类型：D代表首备数据节点，S代表级联备数据节点，C代表查询节点。
node_host	name	节点的通讯IP地址。
node_port	int	节点的通讯端口。
node_host1	name	节点的通讯虚拟IP地址。
node_port1	int	节点的通讯虚拟端口。
nodeis_active	boolean	节点是否正常：正常为true，异常为false。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT * from pg_catalog.pgxc_disaster_read_node();
  node_name   | node_type | node_host  | node_port | node_host1 | node_port1 | nodeis_active
--------------+-----------+------------+-----------+------------+------------+---------------
 dn_6001_6002 | D         | 8.92.0.126 |      4000 | 8.92.0.126 |       4000 | t
 dn_6003_6004 | D         | 8.92.0.128 |      4020 | 8.92.0.128 |       4020 | t
 dn_6005_6006 | D         | 8.92.0.127 |      4040 | 8.92.0.127 |       4040 | t
 dn_6007_6008 | D         | 8.92.0.68  |      4060 | 8.92.0.68  |       4060 | t
 dn_6001_6002 | S         | 8.92.0.127 |      4000 | 8.92.0.127 |       4000 | t
 dn_6003_6004 | S         | 8.92.0.68  |      4020 | 8.92.0.68  |       4020 | t
 dn_6005_6006 | S         | 8.92.0.126 |      4040 | 8.92.0.126 |       4040 | t
 dn_6007_6008 | S         | 8.92.0.128 |      4060 | 8.92.0.128 |       4060 | t
 cn_5001      | C         | 8.92.0.128 |      9080 | 10.0.0.128 |       9080 | t
 cn_5002      | C         | 8.92.0.127 |      9080 | 10.0.0.127 |       9080 | t
(10 rows)
```

1.6.27.7 快照同步函数

快照同步函数是导出当前快照的标识符。

- pg_export_snapshot()

  描述：保存当前事务的快照系统信息，同时向客户端返回该快照的标识符（Snapshot ID）。

  返回值类型：text

  备注：函数pg_export_snapshot保存当前的快照并返回一个文本字符串标识此快照。这个字符串必须传递给想要导入快照的客户端。可用在set transaction snapshot snapshot_id时导入snapshot，但是应用的前提是该事务设置了SERIALIZABLE或REPEATABLE READ隔离级别。而GaussDB目前不支持这两种隔离级别。该函数的输出不可用做set transaction snapshot的输入。

- pg_export_snapshot_and_csn()

  描述：保存当前事务的快照系统信息，同时向客户端返回该快照的标识符（Snapshot ID）以及对应的CSN（Commit Sequence Number，提交序列号）。比pg_export_snapshot()多返回一列CSN，表示当前快照的CSN。

  返回值类型：text

1.6.27.8 数据库对象函数

数据库对象尺寸函数

数据库对象尺寸函数计算数据库对象使用的实际磁盘空间。

- pg_column_size(any)

  描述：存储一个指定的数值需要的字节数（可能压缩过）。

  返回值类型：int

  备注：pg_column_size显示用于存储某个独立数据值的空间。

  ```
  gaussdb=# SELECT pg_column_size(1);
   pg_column_size
  ----------------
                4
  (1 row)
  ```

- pg_database_size(oid)

  描述：指定OID代表的数据库使用的磁盘空间。

  返回值类型：bigint

版权所有 © 华为云计算技术有限公司

- pg_database_size(name)

  描述：指定名称的数据库使用的磁盘空间。

  返回值类型：bigint

  备注：pg_database_size接受一个数据库的OID或者名称，然后返回该对象使用的全部磁盘空间。

  示例：

```
gaussdb=# CREATE DATABASE testdb;
gaussdb=# SELECT pg_database_size('testdb');
 pg_database_size
------------------
         51590112
(1 row)
gaussdb=# DROP DATABASE testdb;
```

  说明

  pg_database_size(oid/name)函数的查询耗时与该oid/name所对应数据库中的对象文件数目相关，基本呈线性关系。如需进行耗时评估，可参考以下步骤：

  1. 由于耗时与软硬件环境有关，因此需要以某个库（所选择库的对象文件数目不宜过多）的测试结果为基准。根据查询耗时和库中对象文件数目，计算出单个对象文件的耗时database_size_t。
  2. 使用select oid,* from pg_database where datname = 'database';查询获取需要查询库的OID，在表空间base和pg_tblspc目录下统计该OID目录下的所有文件数目database_obj_num。如果该目录中包含子目录，则需要进入查看对象文件数目，依次类推；使用ls | wc -l查询目录下文件数目。
  3. 根据基准测试结果中的单文件耗时database_size_t和数据库下所有对象文件数目database_obj_num计算预估总耗时，总耗时 = database_size_t * database_obj_num。如果预估时延过长，需考虑调整执行超时阈值，避免出现不必要的告警。

- get_db_source_datasize()

  描述：估算当前数据库非压缩态的数据总容量。

  返回值类型：bigint

  备注：调用该函数前需要做analyze。

  示例：

```
gaussdb=# analyze;
ANALYZE
gaussdb=# SELECT get_db_source_datasize();
 get_db_source_datasize
------------------------
           35384925667
(1 row)
```

- pg_relation_size(relation regclass, fork text)

  描述：指定表或索引的指定分叉树（'main'，'fsm'或'vm'）使用的磁盘空间。

  返回值类型：bigint

- pg_relation_size(relation regclass)

  描述：pg_relation_size(..., 'main')的简写。

  返回值类型：bigint

  备注：pg_relation_size接受一个表、索引、压缩表的OID或者名称，然后返回它们的字节大小。

- pg_partition_size(oid, oid)

  描述：指定OID代表的分区使用的磁盘空间。其中，第一个oid为表的OID，第二个oid为分区的OID。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

- pg_partition_size(text, text)

  描述：指定名称的分区使用的磁盘空间。其中，第一个text为表名，第二个text为分区名。

  返回值类型：bigint

- pg_partition_indexes_size(oid, oid)

  描述：指定OID代表的分区的索引使用的磁盘空间。其中，第一个oid为表的OID，第二个oid为分区的OID。

  返回值类型：bigint

- pg_partition_indexes_size(text, text)

  描述：指定名称的分区的索引使用的磁盘空间。其中，第一个text为表名，第二个text为分区名。

  返回值类型：bigint

- pg_indexes_size(regclass)

  描述：附加到指定表的索引使用的总磁盘空间。

  返回值类型：bigint

- pg_size_pretty(bigint)

  描述：将以64位整数表示的字节值转换为具有单位的易读格式。

  返回值类型：text

- pg_size_pretty(numeric)

  描述：将以数值表示的字节值转换为具有单位的易读格式。

  返回值类型：text

  备注：pg_size_pretty用于把其他函数的结果格式化成一种易读的格式，可以根据情况使用kB 、MB 、GB 、TB。

- pg_table_size(regclass)

  描述：指定的表使用的磁盘空间，不计索引（但是包含TOAST，自由空间映射和可见性映射）。

  返回值类型：bigint

- pg_tablespace_size(oid)

  描述：指定OID代表的表空间使用的磁盘空间。

  返回值类型：bigint

- pg_tablespace_size(name)

  描述：指定名称的表空间使用的磁盘空间。

  返回值类型：bigint

  备注：

  pg_tablespace_size接受一个数据库的OID或者名称，然后返回该对象使用的全部磁盘空间。

- pg_total_relation_size(regclass)

  描述：指定的表使用的总磁盘空间，包括所有的索引和TOAST数据。

  返回值类型：bigint

- dist_relation_size(nspname name, relname name)

  描述：指定模式下的普通表在每个DN上的relpage数。

版权所有 © 华为云计算技术有限公司

返回值类型：float8

表 1-134 参数说明

入参	类型	含义
nspname	name	表所在的模式名。
relname	name	表名。

1CN6DN数据库环境，可以看到6个DN上的relpage数：

```
gaussdb=# CREATE TABLE tab(col int);
-- node_name与当前环境的节点名称保持一致。
gaussdb=# SELECT * FROM dist_relation_size('public','tab');
 node_name | relpages
-----------+----------
 datanode1 |        0
 datanode2 |        0
 datanode3 |        0
 datanode4 |        0
 datanode6 |        0
 datanode5 |        0
(6 rows)

gaussdb=# DROP TABLE tab;
```

说明

- 该函数不支持在升级观察期使用。
- 该函数不支持在DN上调用。

- dist_partition_size(nspname name, relname name, subname name)
  描述：指定模式下的分区表的指定分区在每个DN上的relpage数。
  返回值类型：float8

  表 1-135 参数说明

入参	类型	含义
nspname	name	表所在的模式名。
relname	name	表名。
subname	name	需要查询的分区名。

  1CN6DN数据库环境，可以看到6个DN上的relpage数：

  ```
  gaussdb=# CREATE TABLE test_part_table(col int)
  PARTITION BY RANGE(col)
  (
      PARTITION p1 VALUES LESS THAN (10),
      PARTITION p2 VALUES LESS THAN (20),
      PARTITION p3 VALUES LESS THAN (MAXVALUE)
  );
  -- node_name与当前环境的节点名称保持一致。
  gaussdb=# SELECT * FROM dist_partition_size('public','test_part_table','p1') ORDER BY node_name;
   node_name | relpages
  ```


版权所有 © 华为云计算技术有限公司

```
-----------+----------
 datanode1 |        0
 datanode2 |        0
 datanode3 |        0
 datanode4 |        0
 datanode5 |        0
 datanode6 |        0
(6 rows)

gaussdb=# DROP TABLE test_part_table;
```

说明

- 该函数不支持在升级观察期使用。
- 该函数不支持在DN上调用。

- datalength(any)

  描述：计算一个指定的数据需要的字节数（不考虑数据的管理空间和数据压缩，数据类型转换等情况）。

  返回值类型：int

  备注：datalength用于计算某个独立数据值的空间。

  示例：

```
gaussdb=# SELECT datalength(1);
datalength
------------
4
(1 row)
```

  目前支持的数据类型及计算方式见下表：

数据类型			存储空间
数值类型	整数类型	TINYINT	1
		SMALLINT	2
		INTEGER	4
		BINARY_INTEGER	4
		BIGINT	8
	任意精度型	DECIMAL	每4位十进制数占两个字节，小数点前后数字分别计算。
		NUMERIC	每4位十进制数占两个字节，小数点前后数字分别计算。
		NUMBER	每4位十进制数占两个字节，小数点前后数字分别计算。
	序列整型	SMALLSERIAL	2
		SERIAL	4
		BIGSERIAL	8

版权所有 © 华为云计算技术有限公司

数据类型			存储空间
	浮点类型	FLOAT4	4
		DOUBLE PRECISION	8
		FLOAT8	8
		BINARY_DOUBLE	8
		FLOAT[(p)]	每4位十进制数占两个字节，小数点前后数字分别计算。
		DEC[(p[,s])]	每4位十进制数占两个字节，小数点前后数字分别计算。
		INTEGER[(p[,s])]	每4位十进制数占两个字节，小数点前后数字分别计算。
布尔类型	布尔类型	BOOLEAN	1
字符类型	字符类型	CHAR	1
		CHAR(n)	n
		CHARACTER(n)	n
		NCHAR(n)	n
		VARCHAR(n)	字符实际字节数。
		CHARACTER VARYING(n)	字符实际字节数。
		VARCHAR2(n)	字符实际字节数。
		NVARCHAR2(n)	字符实际字节数。
		TEXT	字符实际字节数。
		CLOB	字符实际字节数。
时间类型	时间类型	DATE	8
		DATEA（仅在ORA模式下支持）	8
		TIME	8
		TIMETZ	12
		TIMESTAMP	8
		TIMESTAMPTZ	8
		SMALLDATETIME	8

版权所有 © 华为云计算技术有限公司

数据类型			存储空间
		INTERVAL DAY TO SECOND	16
		INTERVAL	16
		RELTIME	4
		ABSTIME	4
		TINTERVAL	12
ROWID类型	ROWID类型	ROWID	24

数据库对象位置函数

- pg_relation_filenode(relation regclass)

  描述：指定关系的文件节点数。

  返回值类型：oid

  备注：pg_relation_filenode接受一个表、索引、序列或压缩表的OID或者名称，并且返回当前分配给它的“filenode”数。文件节点是关系使用的文件名称的基本组件。对大多数表来说，结果和pg_class.relfilenode相同，但对确定的系统目录来说， relfilenode为0而且这个函数必须用来获取正确的值。如果传递一个没有存储的关系，比如一个视图，那么这个函数返回NULL。

- pg_relation_filepath(relation regclass)

  描述：指定关系的文件路径名。只能用于非段页式关系。

  返回值类型：text

  备注：pg_relation_filepath类似于pg_relation_filenode，但是它返回关系的整个文件路径名（相对于数据库集群的数据目录PGDATA）。

  段页式关系建议使用段页式相关函数或视图，例如：

  ```
  SELECT e.*, f.file_name
  FROM gs_seg_extents e, gs_seg_datafiles f
  WHERE e.tablespace_name = f.tablespace_name AND e.bucketnode = f.bucketnode AND e.file_id = f.file_id AND e.forknum = f.forknum;
  ```

- get_large_table_name(relfile_node text, threshold_size_gb int8)

  描述：根据表的文件编码（relfile_node）查询对应的表大小（单位为GB）是否超过阈值（threshold_size_gb），如果超过则返回模式名和表名（形式为schemaname.tablename）， 否则返回字符串'null'。

  返回值类型：text

- pg_filenode_relation(tablespace oid, filenode oid)

  描述：获取对应的tablespace和relfilenode所对应的表名。

  返回类型：regclass

- pg_partition_filenode(partition_oid)

  描述：获取到指定分区表的oid锁对应的filenode。

  返回类型：oid

- pg_partition_filepath(partition_oid)

版权所有 © 华为云计算技术有限公司

描述：指定分区的文件路径名。只能用于非段页式关系。

返回值类型：text

备注：段页式关系建议使用段页式相关函数或视图，例如：

```
SELECT e.*, f.file_name
FROM gs_seg_extents e, gs_seg_datafiles f
WHERE e.tablespace_name = f.tablespace_name AND e.bucketnode = f.bucketnode AND e.file_id =
f.file_id AND e.forknum = f.forknum;
```

回收站对象函数

- gs_is_recycle_object(classid, objid, objname)

  描述：判断是否为回收站对象。分布式不支持该函数。

  返回值类型：Boolean

- gs_is_recycle_obj(classid, objid, objname)

  描述：判断是否为回收站对象。其中classid为对象所在的系统表的OID，objid为对象的OID，objname为对象的名称。

  返回值类型：Boolean

1.6.27.9 咨询锁函数

咨询锁函数用于管理咨询锁（Advisory Lock）。

- pg_advisory_lock(key bigint)

  描述：获取会话级别的排他咨询锁。

  返回值类型：void

  备注：pg_advisory_lock锁定应用程序定义的资源，该资源可以用一个64位或两个不重叠的32位键值标识。如果已经有另外的会话锁定了该资源，则该函数将阻塞到该资源可用为止。这个锁是排他的。多个锁定请求将会被压入栈中，因此，如果同一个资源被锁定了三次，它必须被解锁三次以将资源释放给其他会话使用。

- pg_advisory_lock(key1 int, key2 int)

  描述：获取会话级别的排他咨询锁。

  返回值类型：void

  备注：只允许sysadmin对键值对(65535, 65535)加会话级别的排他咨询锁，普通用户无权限。

- pg_advisory_lock(lock_id int4, lock_id int4, database_name Name)

  描述：通过传入锁ID和数据库名字，获取指定数据库的排他咨询锁。

  返回值类型：void

- pg_advisory_lock_shared(key bigint)

  描述：获取会话级别的共享咨询锁。

  返回值类型：void

- pg_advisory_lock_shared(key1 int, key2 int)

  描述：获取会话级别的共享咨询锁。

  返回值类型：void

  备注：pg_advisory_lock_shared类似于pg_advisory_lock，不同之处仅在于共享锁会话可以和其他请求共享锁的会话共享资源，但排他锁除外。

 版权所有 © 华为云计算技术有限公司

- pg_advisory_unlock(key bigint)

  描述：释放会话级别的排他咨询锁。

  返回值类型：Boolean

- pg_advisory_unlock(key1 int, key2 int)

  描述：释放会话级别的排他咨询锁。

  返回值类型：Boolean

  备注：pg_advisory_unlock释放先前取得的排他咨询锁。如果释放成功则返回true。如果实际上并未持有指定的锁，将返回false并在服务器中产生一条SQL警告信息。

- pg_advisory_unlock(lock_id int4, lock_id int4, database_name Name)

  描述：通过传入锁ID和数据库名字，释放指定数据库上的排他咨询锁。

  返回值类型：Boolean

  备注：如果释放成功则返回true；如果未持有锁，则返回false。

- pg_advisory_unlock_shared(key bigint)

  描述：释放会话级别的共享咨询锁。

  返回值类型：Boolean

- pg_advisory_unlock_shared(key1 int, key2 int)

  描述：释放会话级别的共享咨询锁。

  返回值类型：Boolean

  备注：pg_advisory_unlock_shared类似于pg_advisory_unlock，不同之处在于该函数释放的是共享咨询锁。

- pg_advisory_unlock_all()

  描述：释放当前会话持有的所有咨询锁。

  返回值类型：void

  备注：pg_advisory_unlock_all将会释放当前会话持有的所有咨询锁，该函数在会话结束的时候被隐含调用，即使客户端异常地断开连接也是一样。

- pg_advisory_xact_lock(key bigint)

  描述：获取事务级别的排他咨询锁。

  返回值类型：void

- pg_advisory_xact_lock(key1 int, key2 int)

  描述：获取事务级别的排他咨询锁。

  返回值类型：void

  备注：pg_advisory_xact_lock类似于pg_advisory_lock，不同之处在于锁是自动在当前事务结束时释放，而且不能被显式地释放。只允许sysadmin对键值对(65535, 65535)加事务级别的排他咨询锁，普通用户无权限。

- pg_advisory_xact_lock_shared(key bigint)

  描述：获取事务级别的共享咨询锁。

  返回值类型：void

- pg_advisory_xact_lock_shared(key1 int, key2 int)

  描述：获取事务级别的共享咨询锁。

  返回值类型：void

版权所有 © 华为云计算技术有限公司

备注：pg_advisory_xact_lock_shared类似于pg_advisory_lock_shared，不同之处在于锁是在当前事务结束时自动释放，而且不能被显式地释放。

- pg_try_advisory_lock(key bigint)

  描述：尝试获取会话级排他咨询锁。

  返回值类型：Boolean

  备注：pg_try_advisory_lock类似于pg_advisory_lock，不同之处在于该函数不会阻塞以等待资源的释放。它要么立即获得锁并返回true，要么返回false表示目前不能锁定。

- pg_try_advisory_lock(key1 int, key2 int)

  描述：尝试获取会话级排他咨询锁。

  返回值类型：Boolean

  备注：只允许sysadmin对键值对(65535, 65535)加会话级别的排他咨询锁，普通用户无权限。

- pg_try_advisory_lock_shared(key bigint)

  描述：尝试获取会话级共享咨询锁。

  返回值类型：Boolean

- pg_try_advisory_lock_shared(key1 int, key2 int)

  描述：尝试获取会话级共享咨询锁。

  返回值类型：Boolean

  备注：pg_try_advisory_lock_shared类似于pg_try_advisory_lock，不同之处在于该函数尝试获得共享锁而不是排他锁。

- pg_try_advisory_xact_lock(key bigint)

  描述：尝试获取事务级别的排他咨询锁。

  返回值类型：Boolean

- pg_try_advisory_xact_lock(key1 int, key2 int)

  描述：尝试获取事务级别的排他咨询锁。

  返回值类型：Boolean

  备注：pg_try_advisory_xact_lock类似于pg_try_advisory_lock，不同之处在于如果得到锁，在当前事务的结束时自动释放，而且不能被显式地释放。只允许sysadmin对键值对(65535, 65535)加事务级别的排他咨询锁，普通用户无权限。

- pg_try_advisory_xact_lock_shared(key bigint)

  描述：尝试获取事务级别的共享咨询锁。

  返回值类型：Boolean

- pg_try_advisory_xact_lock_shared(key1 int, key2 int)

  描述：尝试获取事务级别的共享咨询锁。

  返回值类型：Boolean

  备注：pg_try_advisory_xact_lock_shared类似于pg_try_advisory_lock_shared，不同之处在于如果得到锁，在当前事务结束时自动释放，而且不能被显式地释放。

- lock_cluster_ddl()

  描述：尝试对集群内所有存活的CN节点获取会话级别的排他咨询锁。

  返回值类型：Boolean

版权所有 © 华为云计算技术有限公司

备注：只允许sysadmin调用，普通用户无权限。

- unlock_cluster_ddl()
  描述：尝试对CN节点会话级别的排他咨询锁。
  返回值类型：Boolean
  备注：只允许sysadmin调用，普通用户无权限。
- pg_rel_advisory_lock(schema_name cstring, relname cstring)
  描述：获取会话级别的表上的咨询锁。
  返回值类型：void
  备注：pg_rel_advisory_lock仅供内部调用，不建议用户使用。
- pg_rel_advisory_unlock(schema_name cstring, relname cstring)
  描述：释放会话级别的表上的咨询锁。
  返回值类型：void
  备注：pg_rel_advisory_unlock仅供内部调用，不建议用户使用。

1.6.27.10 逻辑复制函数

说明

使用逻辑复制函数时，需要设置GUC参数wal_level为logical。具体配置请参考《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写） > 使用SQL函数接口进行逻辑解码”章节。

- pg_create_logical_replication_slot('slot_name', 'plugin_name', 'output_order')
  描述：创建逻辑复制槽。
  参数说明：
  - slot_name
    流复制槽名称。
    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。
  - plugin_name
    插件名称。
    取值范围：字符串类型，支持取值包括：
    - 通用插件：mppdb_decoding、sql_decoding。
    - 仅并行解码支持：parallel_binary_decoding、parallel_json_decoding、parallel_text_decoding。
  - output_order
    复制槽解码结果输出顺序，该参数为可选参数。
    取值范围：0或1，在DN上默认值为0， 在CN上默认值为1。
    - 0：设为0时，复制槽解码结果按照事务的COMMIT LSN排序。此时复制槽的confirmed_csn为0，此复制槽称为LSN序复制槽。
    - 1：设为1时，复制槽解码结果按照事务的CSN排序。此时复制槽的confirmed_csn为非0值，此复制槽称为CSN序复制槽，此类复制槽的confirmed_flush值无意义。

版权所有 © 华为云计算技术有限公司

说明

- CSN序复制槽仅分布式强一致解码使用，非分布式强一致解码要求使用LSN序复制槽。
- 创建复制槽时，如果output_order输入的数值为小数，会将其四舍五入为整数。

返回值类型：name, text

示例：

```
gaussdb=# SELECT * FROM pg_create_logical_replication_slot('slot_lsn','mppdb_decoding',0);
 slotname | xlog_position
----------+---------------
 slot_lsn | 0/6D08B58
(1 row)

gaussdb=# SELECT * FROM pg_create_logical_replication_slot('slot_csn','mppdb_decoding',1);
 slotname | xlog_position
----------+---------------
 slot_csn | 0/59AD800
(1 row)
```

备注：第一个返回值表示slot_name，第二个返回值在LSN序复制槽和CSN序复制槽下有不同含义。对于LSN序复制槽，该值为复制槽的confirmed_flush，表示COMMIT LSN小于等于该值的事务之后不会被解码输出；对于CSN序复制槽，该值为复制槽的confirmed_csn，表示CSN小于等于该值的事务之后不会被解码输出。调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。此函数目前只支持在CN或主DN上调用。

注意

当该函数在CN上执行时，如果output_order值为1，会在所有CN和主DN上创建同名复制槽，如果某些DN上已存在同名复制槽，会在除当前CN节点外，其他不存在同名复制槽的CN和主DN节点上创建CSN序逻辑复制槽，同时报错复制槽已存在。因此需删除其他CN和主DN上已有的同名复制槽（如为LSN序复制槽则需在对应节点手动删除）再在CN上重新创建。如果在CN上创建复制槽时指定output_order为0，则不会在DN上创建复制槽。CN上创建的复制槽仅用于标识DN上存在相关复制槽。

- pg_create_physical_replication_slot('slot_name', 'isDummyStandby')

  描述：创建新的物理复制槽。

  参数说明：

  - slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。以gs_roach开头的复制槽名为内部保留槽名，只能由内部备份工具GaussRoach使用，执行时需要设置application_name='gs_roach'。

  - isDummyStandby

    预留参数。

    类型：Boolean

  返回值类型：name，text

版权所有 © 华为云计算技术有限公司

说明

- 调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。
- 由于该函数创建的物理复制槽没有restart_lsn，会被认为是无效槽，在做checkpoint时会被自动删除。

- pg_drop_replication_slot('slot_name')

  描述：删除流复制槽。

  参数说明：

  – slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。以gs_roach开头的复制槽名为内部保留槽名，只能由内部备份工具GaussRoach使用，执行时需要设置application_name='gs_roach'。

  返回值类型：void

  说明

  - 调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。
  - 删除数据库时不会删除该库上的逻辑复制槽。该库上的逻辑复制槽需要手动删除，否则将会导致Xlog和系统表历史版本无法回收。
  - 在CN上删除逻辑复制槽时，若为LSN序逻辑复制槽，则仅删除当前节点复制槽，其他节点同名复制槽不受影响；否则只要其他节点有残留同名CSN序逻辑复制槽，执行删除时不会因为某些节点不存在复制槽而报错，同时所有节点的同名复制槽会被成功删除；如果任何节点均不存在该复制槽，则报错。
  - 如果当前CN节点残留LSN序逻辑复制槽，同时其他某些节点上残留同名CSN序逻辑复制槽，则在当前CN节点上执行删除复制槽操作仅会删除本地LSN序逻辑复制槽，待删除完成再次执行删除操作方可删除其他节点的同名复制槽。

- pg_logical_slot_peek_changes('slot_name', 'upto_lsn', upto_nchanges, 'options_name', 'options_value')

  描述：解码但不推进流复制槽（下次解码可以再次获取本次解出的数据）。

  参数说明：

  – slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  – upto_lsn

    在CSN序逻辑复制槽上代表日志的CSN，表示解码直到小于等于此CSN的事务日志解码完毕（可能会解码一个CSN大于指定CSN的事务）；在LSN序复制槽上代表日志的LSN，表示解码直到COMMIT LSN大于等于该LSN的第一个事务解码完毕。

    取值范围：字符串（代表十六进制格式表示的uint64，在正中间用'/'分隔，左右两边各为一个uint32，如某个uint32为0则显示0），如'1/2AAFC60'、'0/A060'或'3A/0'。为NULL时表示不对解码截止的日志位置做限制。

  – upto_nchanges

    解码条数（包含begin和commit）。假设一共有三条事务，分别包含3、5、7条记录，如果upto_nchanges为4，那么会解码出前两个事务共8条记录。解

版权所有 © 华为云计算技术有限公司

码完第二条事务时发现解码条数记录大于等于upto_nchanges，会停止解码。

取值范围：非负整数。

说明

upto_lsn和upto_nchanges中任一参数达到限制，解码都会结束。

- options：此项为可选参数，由一系列options_name和options_value一一对应组成。
  - include-xids

    解码出的data列是否包含xid信息。

    取值范围：bool型，默认值为true。
    - false：设为false时，解码出的data列不包含xid信息。
    - true：设为true时，解码出的data列包含xid信息。
  - skip-empty-xacts

    解码时是否忽略空事务信息。

    取值范围：bool型，默认值为false。
    - false：设为false时，解码时不忽略空事务信息。
    - true：设为true时，解码时会忽略空事务信息。
  - include-timestamp

    解码信息是否包含commit时间戳。

    取值范围：bool型，默认值为true。
    - false：设为false时，解码信息不包含commit时间戳。
    - true：设为true时，解码信息包含commit时间戳。
  - only-local

    是否仅解码本地日志。

    取值范围：Boolean型，默认值为true。
    - false：设为false时，解码非本地日志和本地日志。
    - true：设为true时，仅解码本地日志。
  - force-binary

    是否以二进制格式输出解码结果

    取值范围：Boolean型，默认值为false。
    - false：设为false时，以文本格式输出解码结果。
    - true：暂不支持设置。
  - white-table-list

    白名单参数，包含需要进行解码的schema和表名。

    取值范围：包含白名单中表名的字符串，不同的表以','为分隔符进行隔离；使用'*'来模糊匹配所有情况；schema名和表名间以'.'分割，不允许存在任意空白符。例：

    ```
    SELECT * FROM pg_logical_slot_peek_changes('slot1', NULL, 4096, 'white-table-list', 'public.t1,public.t2');
    ```


版权所有 © 华为云计算技术有限公司

- skip-xid-list

  事务黑名单参数，包含需要跳过解码的事务xid。升级观察期不支持跳过事务。

  取值范围：3~18446744073709551615（即uint64最大值）。包含事务xid的字符串，不同的事务以'/'为分隔符进行隔离；不允许存在任意空白符。例：

  ```
  SELECT * FROM pg_logical_slot_peek_changes('slot1', NULL, 4096, 'skip-xid-list',
  '1234/2345');
  ```

- max-txn-in-memory

  内存管控参数，单位为MB，单个事务占用内存大于该值即进行落盘。

  取值范围：0~100的整型，默认值为0，即不开启此种管控。

- max-reorderbuffer-in-memory

  内存管控参数，单位为GB，拼接-发送线程中正在拼接的事务总内存（包含缓存）大于该值则对当前解码事务进行落盘。

  取值范围：0~100的整型，默认值为0，即不开启此种管控。

- desc-memory-limit

  内存管控参数，单位为MB，逻辑解码任务维护的表元信息总内存大于该值时，触发淘汰机制清理部分表元信息。

  取值范围：10~1048576的整型，默认值为200。

- include-user

  事务的BEGIN逻辑日志是否输出事务的用户名字。

  取值范围：Boolean型，默认值为false。

  - false：设为false时，事务的BEGIN逻辑日志不输出事务的用户名字。
  - true：设为true时，事务的BEGIN逻辑日志输出事务的用户名字。

- exclude-userids

  黑名单用户的OID参数。

  取值范围：指定黑名单用户的OID，多个OID通过','分隔，不校验用户OID是否存在。

- exclude-users

  黑名单用户的名字参数。

  取值范围：指定黑名单用户的名字，多个名字通过','分隔；通过dynamic-resolution设置是否动态解析识别用户名字。若解码报错用户不存在而中断，在确定日志产生时刻不存在对应的黑名单用户，可以通过配置dynamic-resolution成true或者从用户黑名单中删除报错用户名字来启动解码继续获取逻辑日志。

- enable-ddl-decoding

  逻辑解码控制参数，用于控制是否开启DDL语句的逻辑解码。

  取值范围：bool型，默认值为false。

  - false：设为false时，不开启DDL语句的逻辑解码。

版权所有 © 华为云计算技术有限公司

- true：设为true时，开启DDL语句的逻辑解码。

- enable-ddl-json-format

  逻辑解码控制参数，用于控制DDL的反解析流程以及输出形式。

  取值范围：bool型，默认值为false。

  - false：设为false时，传送TEXT格式的DDL反解析结果。
  - true：设为true时，传送JSON格式的DDL反解析结果。

- skip-generated-columns：

  逻辑解码控制参数，用于跳过存储生成列的输出。对UPDATE和DELETE的旧元组无效，相应元组始终会输出存储生成列。分布式版本暂不支持生成列语法，CN解码不支持配置此选项。

  取值范围：boolean型，默认值为false/off。

  - true/on：值为true/on时，不输出存储生成列的解码结果。
  - false/off：设为false/off时，输出存储生成列的解码结果。

  虚拟生成列不受此参数控制，DML的解码结果始终不会输出虚拟生成列。

- dynamic-resolution

  是否动态解析黑名单用户名字。

  取值范围：bool型，默认值为true。

  - false：设为false时，当解码观测到黑名单exclude-users中用户不存在时将会报错并退出逻辑解码。
  - true：设为true时，当解码观测到黑名单exclude-users中用户不存在时继续解码。

- decode-sequence：

  逻辑解码控制参数，用来指定是否输出sequence值的变更日志的解码结果。

  取值范围：boolean型，默认值为false。

  - true：暂不支持设置。
  - false：设为false时，不输出sequence值的变更日志的解码结果。

须知

解码选项decode-sequence当前仅允许设置为false，设置为true会在启动解码时报错退出。

- enable-decode-position:

  逻辑解码控制参数，该参数主要控制逻辑解码读取到解码位置日志时，是否发送该位置日志信息。

  取值范围：boolean型，默认值为false。

  - true：设为true时，输出位置日志。
  - false：设为false时，不输出位置日志。

版权所有 © 华为云计算技术有限公司

- enable-rowno:

  逻辑解码控制参数，该参数用以控制解码包含rowid系统列的无主键表且REPLICA IDENTITY选项明确设置为FULL或者实际效果等同于FULL时，是否输出INSERT/UPDATE/DELETE操作中新元组和旧元组的rowno系统列。使用该选项输出rowno系统列，请确保目标库中存在名称和类型相同的列，否则回放逻辑解码日志可能失败。如有疑问，请联系华为技术工程师提供技术支持。

  取值范围：boolean型

  - true：输出rowno系统列。
  - false：不输出rowno系统列。

  默认值：false

须知

- 该选项对sql_decoding插件无效。
- REPLICA IDENTITY选项的实际效果等同于FULL目前指的是Ustore表的NOTHING和DEFAULT选项，详情参见**REPLICA IDENTITY { DEFAULT | USING INDEX index_name | FULL | NOTHING }**选项的说明。

说明

其他配置选项可参考《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写）> 逻辑解码选项”章节。

返回值类型：text、xid、text

示例：

```
gaussdb=# EXECUTE DIRECT ON (datanode1) 'SELECT * FROM
pg_logical_slot_peek_changes(''slot_lsn'',NULL,4096,''skip-empty-xacts'',''on'')';
 location  | xid  |                                                             data
-----------+-------
+-----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------
 0/6D0B500 | 46914 | BEGIN 46914
 0/6D0B530 | 46914 | {"table_name":"public.t1","op_type":"INSERT","columns_name":
["a","b"],"columns_type":["integer","integer"],"columns_val":["3","1"],"old_keys_name":
[],"old_keys_type":[],"old_keys_val":[]}
 0/6D0B8B8 | 46914 | COMMIT 46914 (at 2023-02-22 17:29:31.090018+08) CSN 94034528
 0/6D0BB58 | 46915 | BEGIN 46915
 0/6D0BB88 | 46915 | {"table_name":"public.t1","op_type":"INSERT","columns_name":
["a","b"],"columns_type":["integer","integer"],"columns_val":["3","2"],"old_keys_name":
[],"old_keys_type":[],"old_keys_val":[]}
 0/6D0BF08 | 46915 | COMMIT 46915 (at 2023-02-22 17:31:30.672093+08) CSN 94034568
 0/6D0BF08 | 46916 | BEGIN 46916
 0/6D0BF38 | 46916 | {"table_name":"public.t1","op_type":"INSERT","columns_name":
["a","b"],"columns_type":["integer","integer"],"columns_val":["3","3"],"old_keys_name":
[],"old_keys_type":[],"old_keys_val":[]}
 0/6D0C218 | 46916 | COMMIT 46916 (at 2023-02-22 17:31:34.438319+08) CSN 94034570
(9 rows)

gaussdb=# EXECUTE DIRECT ON (datanode1) 'SELECT * FROM
pg_logical_slot_peek_changes(''slot_csn'',NULL,4096,''skip-empty-xacts'',''on'')';
 location  | xid  |                                                             data
-----------+-------
+-----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------
 0/0       | 46914 | BEGIN CSN: 94034528
```

```
 0/0       | 46914 | {"table_name":"public.t1","op_type":"INSERT","columns_name":
["a","b"],"columns_type":["integer","integer"],"columns_val":["3","1"],"old_keys_name":
[],"old_keys_type":[],"old_keys_val":[]}
 0/59ADA60 | 46914 | COMMIT 46914 (at 2023-02-22 17:29:31.090018+08) CSN 94034528
 0/59ADA60 | 46915 | BEGIN CSN: 94034568
 0/59ADA60 | 46915 | {"table_name":"public.t1","op_type":"INSERT","columns_name":
["a","b"],"columns_type":["integer","integer"],"columns_val":["3","2"],"old_keys_name":
[],"old_keys_type":[],"old_keys_val":[]}
 0/59ADA88 | 46915 | COMMIT 46915 (at 2023-02-22 17:31:30.672093+08) CSN 94034568
 0/59ADA88 | 46916 | BEGIN CSN: 94034570
 0/59ADA88 | 46916 | {"table_name":"public.t1","op_type":"INSERT","columns_name":
["a","b"],"columns_type":["integer","integer"],"columns_val":["3","3"],"old_keys_name":
[],"old_keys_type":[],"old_keys_val":[]}
 0/59ADA8A | 46916 | COMMIT 46916 (at 2023-02-22 17:31:34.438319+08) CSN 94034570
(9 rows)
```

备注：上述函数示例中，datanode1为需要解码的DN节点名称，slot_lsn/slot_csn为逻辑复制槽名称。

函数返回解码结果，每一条解码结果包含三列，对应上述返回值类型，分别表示LSN（LSN序复制槽）或CSN（CSN序复制槽）位置、xid和解码内容。其中location列代表CSN时，仅当解码到COMMIT日志时才会更新。调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。

说明

在CN上，该函数不支持在CSN序复制槽（confirmed_csn值不为0）上执行。

- pg_logical_slot_get_changes('slot_name', 'upto_lsn', upto_nchanges, 'options_name', 'options_value')

  描述：解码并推进流复制槽。

  参数说明：与pg_logical_slot_peek_changes一致，详细内容请参见**pg_logical_slot_peek_changes('slot_name', 'upto_lsn', upto_nchanges, 'options_name', 'options_value')**。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。

  说明

  在CN上，该函数不支持在CSN序复制槽（confirmed_csn值不为0）上执行。此函数在备DN调用会同步推进主DN上对应的逻辑复制槽。在备DN上执行该函数，推进主DN对应复制槽时需占用主DN的一个walsender。由于逻辑解码功能会为每个逻辑复制槽预留walsender，因此正常场景执行该函数会正常推进主DN的逻辑复制槽，如果短时间连续执行该函数会导致通知主DN推进失败，且没有报错。

- pg_logical_slot_peek_binary_changes('slot_name', 'upto_lsn', upto_nchanges, 'options_name', 'options_value')

  描述：以二进制格式解码且不推进流复制槽（下次解码可以再次获取本次解出的数据）。

  参数说明：

  – slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  – upto_lsn

    在CSN序逻辑复制槽上代表日志的CSN，表示解码直到小于等于此CSN的事务日志解码完毕（可能会解码一个CSN大于指定CSN的事务）；在LSN序复制槽

版权所有 © 华为云计算技术有限公司

上代表日志的LSN，表示解码直到COMMIT LSN大于等于该LSN的第一个事务解码完毕。

取值范围：字符串（代表十六进制格式表示的uint64，在正中间用'/'分隔，左右两边各为一个uint32，如某个uint32为0则显示0），如'1/2AAFC60'、'0/A060'或'3A/0'。为NULL时表示不对解码截止的日志位置做限制。

- upto_nchanges

  解码条数（包含begin和commit）。假设一共有三条事务，分别包含3、5、7条记录，如果upto_nchanges为4，那么会解码出前两个事务共8条记录。解码完第二条事务时发现解码条数记录大于等于upto_nchanges，会停止解码。

  取值范围：非负整数。

  说明

  upto_lsn和upto_nchanges中任一参数达到限制，解码都会结束。

- options：此项为可选参数，由一系列options_name和options_value一一对应组成。

  - include-xids

    解码出的data列是否包含xid信息。

    取值范围：Boolean型，默认值为true。

    - false：设为false时，解码出的data列不包含xid信息。
    - true：设为true时，解码出的data列包含xid信息。

  - skip-empty-xacts

    解码时是否忽略空事务信息。

    取值范围：Boolean型，默认值为false。

    - false：设为false时，解码时不忽略空事务信息。
    - true：设为true时，解码时会忽略空事务信息。

  - include-timestamp

    解码信息是否包含commit时间戳。

    取值范围：Boolean型，默认值为true。

    - false：设为false时，解码信息不包含commit时间戳。
    - true：设为true时，解码信息包含commit时间戳。

  - only-local

    是否仅解码本地日志。

    取值范围：Boolean型，默认值为true。

    - false：设为false时，解码非本地日志和本地日志。
    - true：设为true时，仅解码本地日志。

  - force-binary

    对此函数不生效。

    取值范围：Boolean型，均以二进制格式输出结果。

  - white-table-list

版权所有 © 华为云计算技术有限公司

白名单参数，包含需要进行解码的Schema和表名。

取值范围：包含白名单中表名的字符串，不同的表以','为分隔符进行隔离；使用'*'来模糊匹配所有情况；schema名和表名间以'.'分割，不允许存在任意空白符。例：select * from pg_logical_slot_peek_binary_changes('slot1', NULL, 4096, 'white-table-list', 'public.t1,public.t2');

- skip-xid-list

  事务黑名单参数，包含需要跳过解码的事务xid。升级观察期不支持跳过事务。

  取值范围：3~18446744073709551615（即uint64最大值）。包含事务xid的字符串，不同的事务以'/'为分隔符进行隔离；不允许存在任意空白符。例：

  ```
  SELECT * FROM pg_logical_slot_peek_changes('slot1', NULL, 4096, 'skip-xid-list', '1234/2345');
  ```

- max-txn-in-memory

  内存管控参数，单位为MB，单个事务占用内存大于该值即进行落盘。

  取值范围：0~100的整型，默认值为0，即不开启此种管控。

- max-reorderbuffer-in-memory

  内存管控参数，单位为GB，拼接-发送线程中正在拼接的事务总内存（包含缓存）大于该值则对当前解码事务进行落盘。

  取值范围：0~100的整型，默认值为0，即不开启此种管控。

- desc-memory-limit

  内存管控参数，单位为MB，逻辑解码任务维护的表元信息总内存大于该值时，触发淘汰机制清理部分表元信息。

  取值范围：10~1048576的整型，默认值为200。

- include-user

  事务的BEGIN逻辑日志是否输出事务的用户名字。

  取值范围：Boolean型，默认值为false。

  - false：设为false时，事务的BEGIN逻辑日志不输出事务的用户名字。
  - true：设为true时，事务的BEGIN逻辑日志输出事务的用户名字。

- exclude-userids

  黑名单用户的OID参数。

  取值范围：指定黑名单用户的OID，多个OID通过','分隔，不校验用户OID是否存在。

- exclude-users

  黑名单用户的名字参数。

  取值范围：指定黑名单用户的名字，多个名字通过逗号（,）分隔；通过dynamic-resolution设置是否动态解析识别用户名字。若解码报错用户不存在而中断，在确定日志产生时刻不存在对应的黑名单用户，可以通过配置dynamic-resolution为true或者从用户黑名单中删除报错用户名字来启动解码继续获取逻辑日志。

版权所有 © 华为云计算技术有限公司

- dynamic-resolution

  是否动态解析黑名单用户名字。

  取值范围：Boolean型，默认值为true。

  - false：设为false时，当解码观测到黑名单exclude-users中用户不存在时将会报错并退出逻辑解码。
  - true：设为true时，当解码观测到黑名单exclude-users中用户不存在时继续解码。

- decode-sequence：

  逻辑解码控制参数，用来指定是否输出sequence值的变更日志的解码结果。

  取值范围：boolean型，默认值为false。

  - true：暂不支持设置。
  - false：设为false时，不输出sequence值的变更日志的解码结果。

  须知

  解码选项decode-sequence当前仅允许设置为false，设置为true会在启动解码时报错退出。

- enable-decode-position:

  逻辑解码控制参数，该参数主要控制逻辑解码读取到解码位置日志时，是否发送该位置日志信息。

  取值范围：boolean型，默认值为false。

  - true：设为true时，输出位置日志。
  - false：设为false时，不输出位置日志。

- enable-rowno:

  逻辑解码控制参数，该参数用以控制解码包含rowid系统列的无主键表且REPLICA IDENTITY选项明确设置为FULL或者实际效果等同于FULL时，是否输出INSERT/UPDATE/DELETE操作中新元组和旧元组的rowno系统列。使用该选项输出rowno系统列，请确保目标库中存在名称和类型相同的列，否则回放逻辑解码日志可能失败。如有疑问，请联系华为技术工程师提供技术支持。

  取值范围：boolean型

  - true：输出rowno系统列。
  - false：不输出rowno系统列。

  默认值：false

版权所有 © 华为云计算技术有限公司

须知

- 该选项对sql_decoding插件无效。
- REPLICA IDENTITY选项的实际效果等同于FULL目前指的是Ustore表的NOTHING和DEFAULT选项，详情参见**REPLICA IDENTITY { DEFAULT | USING INDEX index_name | FULL | NOTHING }**选项的说明。

说明

某些配置选项在函数中仅能配置而不实际生效，可参考《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写） > 逻辑解码选项”章节。

返回值类型：text、xid、bytea

备注：函数返回解码结果，每一条解码结果包含三列，对应上述返回值类型，分别表示LSN位置、xid和二进制格式的解码内容。调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。

说明

在CN上，该函数不支持在CSN序复制槽（confirmed_csn值不为0）上执行。

- pg_logical_slot_get_binary_changes('slot_name', 'upto_lsn', upto_nchanges, 'options_name', 'options_value')

  描述：以二进制格式解码并推进流复制槽。

  参数说明：与pg_logical_slot_peek_binary_changes一致，详细内容请参见**pg_logical_slot_peek_bi...**。

  备注：调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。

  说明

  在CN上，该函数不支持在CSN序复制槽（confirmed_csn值不为0）上执行。该函数不支持在备DN上执行。

- pg_replication_slot_advance ('slot_name', 'upto_lsn')

  描述：直接推进流复制槽到指定位置（CSN序逻辑复制槽代表CSN，LSN序复制槽代表LSN），不输出解码结果。

  参数说明：

  - slot_name

    流复制槽名称。

    取值范围：字符串，不支持除小写字母，数字,以及“_”，“?”，“-”，“.”以外的字符，且不支持'.'或'..'单独作为复制槽名称。

  - upto_lsn

    在CSN序逻辑复制槽上代表推进到的日志CSN位置，下次解码时只会输出比该CSN大的事务结果。如果输入的CSN比当前流复制槽记录的confirmed_csn还要小，则直接返回；如果输入的CSN比当前可获取的最新CSN要大，则推进到当前可获取的最新CSN；如果输入的CSN比当前最新物理日志CSN小，则推进到第一个比输入的CSN大的有效日志位置。

    在LSN序复制槽上代表推进到的日志LSN位置，下次解码时只会输出提交位置比该LSN大的事务结果。如果输入的LSN比当前流复制槽记录的推进位置还要

版权所有 © 华为云计算技术有限公司

小，则报错；如果输入的LSN比当前最新物理日志LSN还要大，则推进到当前最新物理日志LSN；如果输入的LSN比当前最新物理日志LSN小，则推进到第一个比输入的LSN大的有效日志位置。

取值范围：字符串（代表十六进制格式表示的uint64，在正中间用'/'分隔，左右两边各为一个uint32，如某个uint32为0则显示0），如'1/2AAFC60'、'0/A060'或'3A/0'。为NULL时表示不对解码截止的日志位置做限制。

返回值类型：name, text

说明

- 返回值分别对应slot_name和实际推进到的位置（CSN或LSN）。调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。
- 在CN上，该函数不支持在CSN序复制槽（confirmed_csn值不为0）上执行。该函数在备DN上支持对备份复制槽的执行。该函数在备DN上支持对逻辑复制槽执行，同步推进主DN上对应的逻辑复制槽。在备DN上执行该函数，推进主DN对应复制槽时需占用主DN的一个walsender。由于逻辑解码功能会为每个逻辑复制槽预留walsender，因此正常场景执行该函数会正常推进主DN的逻辑复制槽，如果短时间连续执行该函数会导致通知主DN推进失败，且没有报错。

- pg_get_replication_slots()

  描述：获取复制槽列表。

  表 1-136 返回值说明

字段名称	数据类型	说明
slot_name	text	复制槽名称。
plugin	text	逻辑复制槽对应的输出插件名称。
slot_type	text	复制槽的类型： ● physical：物理复制槽。 ● logical：逻辑复制槽。
datoid	oid	复制槽所在的数据库OID。
active	boolean	复制槽是否为激活状态： ● f：未激活。 ● t：已激活。
xmin	xid	数据库须为复制槽保留的最早事务的事务号。
catalog_xmin	xid	数据库须为逻辑复制槽保留的最早的涉及系统表的事务的事务号。
restart_lsn	text	复制槽需要的最早Xlog的物理位置。
dummy_standby	boolean	预留参数。
confirmed_flush	text	客户端确认接收到的日志位置（逻辑复制槽专用）。
confirmed_csn	xid	客户端确认接收到的日志中最后一个事务对应的CSN（逻辑复制槽专用）。

版权所有 © 华为云计算技术有限公司

字段名称	数据类型	说明
dictionary_csn_min	xid	数据库须为逻辑复制槽保留的最早的涉及字典表的事务提交CSN。
slot_dictionary_type	text	逻辑复制槽的类型。
intra_cluster	boolean	该复制槽是否无需在集群之间进行同步： ● true：无需同步。 ● false：需同步。
remote_stream	boolean	该复制槽当前是否在解码远端日志流。

示例：

```
在DN上执行查询：
gaussdb=# SELECT * FROM pg_get_replication_slots();
 slot_name |     plugin     | slot_type | datoid | active | xmin | catalog_xmin | restart_lsn |
dummy_standby | confirmed_flush | confirmed_csn | dictionary_csn_min | slot_dictionary_type |
intra_cluster | remote_stream
-----------+----------------+-----------+--------+--------+------+--------------+-------------+---------------
+-----------------+---------------+--------------------+----------------------
 dn_6002   |              | physical |     0 | t      |      |              | 0/3622B528  | f             |
|             |                 |                    |
 f         |    f
 dn_6003   |              | physical |     0 | t      |      |              | 0/3622B528  | f             |
|             |                 |                    |
 f         |    f

 slot_lsn  | mppdb_decoding | logical   | 131072 | f      |      |       66658 | 0/36252350  | f             |
0/362523D0      |             |                 |                    |
 f         |    f
 slot_test | mppdb_decoding | logical   | 131072 | f      |      |       66658 | 0/36251718  | f
|             |     10025527 |                 |                    |
 f         |    f
(4 rows)
在CN上执行查询：
gaussdb=# SELECT * FROM pg_get_replication_slots();
 slot_name |     plugin     | slot_type | datoid | active | xmin | catalog_xmin | restart_lsn |
dummy_standby | confirmed_flush | confirmed_csn | dictionary_csn_min | slot_dictionary_type |
intra_cluster | remote_stream
-----------+----------------+-----------+--------+--------+------+--------------+-------------+---------------
+-----------------+---------------+--------------------+----------------------
 slot_test | mppdb_decoding | logical   | 139264 | f      |      |              |             | f             |
|             |                 |                    |
 f         |    f
(1 row)
```

须知

在DN上执行查询，LSN序逻辑复制槽的confirmed_csn查询结果为空，CSN序逻辑复制槽的confirmed_flush查询结果为空；在CN上执行查询，CSN序逻辑复制槽的catalog_xmin、restart_lsn、confirmed_flush和confirmed_csn均不予显示，查询结果为空。

- pg_logical_get_area_changes('start_lsn', 'upto_lsn', upto_nchanges, 'decoding_plugin', 'xlog_path', 'options_name', 'options_value')

  描述：指定lsn区间进行解码，或者指定Xlog文件进行解码。

版权所有 © 华为云计算技术有限公司

说明

约束条件如下：

- 当前网络和硬件环境正常。
- 单条元组大小建议不超过500MB，500MB~1GB之间会报错。
- 不支持数据页复制这种不落Xlog的日志解码。
- 调用接口时，要求日志级别wal_level=logical，且只有在wal_level=logical期间产生的日志文件才能被解析。如果使用的Xlog文件为非logical级别，则报错不解码。
- Xlog文件只能被完全同构的dn的某个副本解析，且数据库没有发生DDL操作和VACUUM FULL，以确保可以找到数据对应的元信息。
- 需注意一次不要读入过多Xlog文件，在指定范围解码不指定文件解码时，推荐指定一次一个Xlog文件的大小。一般情况下，一个Xlog文件解码过程中占用内存约为该Xlog文件大小的2~3倍。
- 不支持VACUUM FULL之前的日志解码。
- 不能解码扩容前的Xlog文件。
- enable-ddl不打开时：不支持toast类型、clob类型和blob类型的字段解码。当enable-ddl打开时：支持相关类型的解码。
- 此解码方式为根据Xlog文本记录数据进行解码，将可解码内容解码出来，不基于事务进行解码。因此不在此Xlog里的数据无法解码。若相关Xlog日志缺失，缺失日志也无法解码。
- enable-ddl不打开时：从解码点开始，如果未指定解码文件，会先检测从解码开始点到最新redo值之间是否发生ddl，如果发生ddl则全部不解码；如果指定解码文件，会同时检测解码文件的开始点到文件最后可读内容之间，以及数据目录下Xlog开始点到最新redo值之间是否发生ddl，检测到一条ddl则对所有表都不解码。
- enable-ddl不打开时：不支持toast类型、clob类型和blob类型的字段解码。当enable-ddl打开时：支持相关类型的解码。
- enable-ddl打开时：支持区间含有DDL的解码，并且可以解码输出DDL语句，输出DDL语句范围请参见《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写） > 逻辑解码支持DDL”章节。
- 为解析某个表的UPDATE和DELETE语句，需为此表配置REPLICA IDENTITY属性，在此表无主键时需要配置为FULL，具体配置方式请参见**ALTER TABLE**章节中“REPLICA IDENTITY { DEFAULT | USING INDEX index_name | FULL | NOTHING }”字段。
- M兼容模式下，float4类型的解码数据值会以实际存储值输出，行为同SQL查询存在显示上的差异。

备注：打开三权分立时，只有数据库初始用户可以调用；关闭三权分立时，需要具备系统管理员权限。

参数说明：

- start_lsn

  指定开始解码的lsn。

  取值范围：字符串（LSN，格式为xlogid/xrecoff），如'1/2AAFC60'。为NULL时表示不对解码起始的日志位置做限制。

- upto_lsn

  指定解码结束的lsn。

  取值范围：字符串（LSN，格式为xlogid/xrecoff），如'1/2AAFC60'。为NULL时表示不对解码截止的日志位置做限制。

- upto_nchanges

  解码条数（包含begin和commit）。假设一共有三条事务，分别包含3、5、7条记录，如果upto_nchanges为4，那么会解码出前两个事务共8条记录。解码完第二条事务时发现解码条数记录大于等于upto_nchanges，会停止解码。

版权所有 © 华为云计算技术有限公司

取值范围：非负整数。

说明

LSN和upto_nchanges中任一参数达到限制，解码都会结束。

- decoding_plugin

解码插件，指定解码内容输出格式的so插件。

取值范围：提供mppdb_decoding和sql_decoding两个解码插件。

- xlog_path

解码插件，指定解码文件的Xlog绝对路径。

取值范围：NULL 或者 Xlog文件绝对路径的字符串。

- options：此项为可选参数，由一系列options_name和options_value一一对应组成，可以缺省。

  - include-xids

    解码出的data列是否包含xid信息。

    取值范围：bool型，默认值为true。

    false：设为false时，解码出的data列不包含xid信息。

    true：设为true时，解码出的data列包含xid信息。

  - include-timestamp

    解码信息是否包含commit时间戳。

    取值范围：bool型，默认值为true。

    false：设为false时，解码信息不包含commit时间戳。

    true：设为true时，解码信息包含commit时间戳。

  - only-local

    是否仅解码本地日志。

    取值范围：Boolean型，默认值为true。

    false：设为false时，解码非本地日志和本地日志。

    true：设为true时，仅解码本地日志。

  - force-binary

    是否以二进制格式输出解码结果

    取值范围：Boolean型，默认值为false。

    false：设为false时，以文本格式输出解码结果。

    true：暂不支持设置。

  - white-table-list

    白名单参数，包含需要进行解码的Schema和表名。取值范围：包含白名单中表名的字符串，不同的表以','为分隔符进行隔离；使用'*'来模糊匹配所有情况；Schema名和表名间以'.'分割，不允许存在任意空白符。

  - white-database-list

    白名单参数，包含需要进行解码的database名。取值范围：包含白名单中database名的字符串，不同的database名以','为分隔符进行隔离；不允许存在任意空白符。如果不设置white-database-list时，只解码当前连接的database Xlog日志；如果设置white-database-list时，只解码

版权所有 © 华为云计算技术有限公司

white-database-list里的database Xlog日志。如果是对系统库解码时，也支持对系统库中普通表的数据找回。

- enable-ddl

  支持含有DDL语法的解码。从起始位置到当前最新日志之间，如果出现DDL语法，不影响功能使用。取值范围：0或1。

  设置为1：打开状态。该函数支持含有DDL语法的解码。

  设置为0：默认关闭状态。该函数不支持含有DDL语法的解码，如果检测到DDL语句，直接退出。

  若表结构发生过删除列操作，enable-ddl在打开状态时，解码数据的列名会被UNDEFINED占位符替代。

- data-limit

  解码输出数据量。单位：KB。

  取值范围：大于等于0的整数。默认值0，表示不限制解码结果大小。

  该函数入参data-limit与GUC中data-limit配合使用，解码超过data-limit大小结束。

  规则：

  1) GUC中data-limit设置为0，函数解码data-limit不设置或者设置为0，解码结果无上限。
  2) GUC中data-limit不设置，函数解码data-limit设置为0，解码结果上限为10GB。
  3) GUC中data-limit设置，函数解码data-limit不设置或者设置为0，解码结果上限为GUC中的data-limit。
  4) GUC中data-limit和函数data-limit同时设置，取两者较小值作为解码结果上限。

- xlog_dir：

  支持归档日志文件夹解码，且文件夹名称与WAL日志文件前8位同名。当且仅当enable-ddl = 1，输入的start_lsn 和 upto_lsn 为有效区间，可以进行解码。取值范围：文件夹的全路径。

  具体可参考《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写） > 逻辑解码数据找回功能”章节。

  ```
  gaussdb=# SELECT * FROM pg_logical_get_area_changes('1/F9FC24C8', '1/FA67B2A0',
  NULL, 'sql_decoding', NULL ,'enable-ddl','1','xlog_dir','/base/username/xlog_dir/00000000');
  ```

- decode-sequence：

  逻辑解码控制参数，用来指定是否输出sequence值的变更日志的解码结果。

  取值范围：boolean型，默认值为false。

  - true：暂不支持设置。
  - false：设为false时，不输出sequence值的变更日志的解码结果。

须知

解码选项decode-sequence当前仅允许设置为false，设置为true会在启动解码时报错。

版权所有 © 华为云计算技术有限公司

- enable-rowno:

  逻辑解码控制参数，该参数用以控制解码包含rowid系统列的无主键表且REPLICA IDENTITY选项明确设置为FULL或者实际效果等同于FULL时，是否输出INSERT/UPDATE/DELETE操作中新元组和旧元组的rowno系统列。使用该选项输出rowno系统列，请确保目标库中存在名称和类型相同的列，否则回放逻辑解码日志可能失败。如有疑问，请联系华为技术工程师提供技术支持。

  取值范围：boolean型

  - true：输出rowno系统列。
  - false：不输出rowno系统列。

  默认值：false

  须知

  - 该选项对sql_decoding插件无效。
  - REPLICA IDENTITY选项的实际效果等同于FULL目前指的是Ustore表的NOTHING和DEFAULT选项，详情参见**REPLICA IDENTITY { DEFAULT | USING INDEX index_name | FULL | NOTHING }**选项的说明。

  说明

  pg_logical_get_area_changes只对以上参数生效。其余参数max-txn-in-memory、include-user、exclude-userids、exclude-users、dynamic-resolution等（详见pg_logical_slot_peek_changes）不生效。

返回值类型：text、xid、text

示例：

```
-在CN上执行DML操作，在DN上可以查询出当前DN上执行的DML：
gaussdb=# SELECT * FROM pg_logical_get_area_changes('0/502E418', NULL, NULL, 'sql_decoding',
NULL);
 location  |  xid  |                                data
-----------+-------+-------------------------------------------------------------------------
 0/502E448 | 17365 | insert into public.t1 values (1, 1);
 0/502E5A0 | 17365 | COMMIT 17365 (at 2023-11-01 11:28:43.92526+08) 2010016
 0/502E5D0 | 17366 | delete from public.t1 where a = 1;insert into public.t1 values (1, 2);
 0/502E6D8 | 17366 | COMMIT 17366 (at 2023-11-01 11:28:45.889283+08) 2010017
 0/502E7B8 | 17367 | delete from public.t1 where a = 1;
 0/502E8B0 | 17367 | COMMIT 17367 (at 2023-11-01 11:28:48.301307+08) 2010018
(6 rows)
```

备注：函数返回解码结果，每一条解码结果包含三列，对应上述返回值类型，分别表示LSN位置、xid和解码内容。

- gs_get_parallel_decode_status()

  描述：监控各个解码线程的读取日志队列和解码结果队列的长度，以便定位并行解码性能瓶颈。

  返回值类型：text、int、text、text、text、int64、int64、TimestampTz

  示例：

```
gaussdb=# SELECT * FROM gs_get_parallel_decode_status();
 slot_name | parallel_decode_num | read_change_queue_length  | decode_change_queue_length |
reader_lsn | working_txn_cnt | working_txn_memory |      decoded_time
-----------+---------------------+---------------------------+----------------------------+-----------
+-----------------+-------------------+-----------------------
```


版权所有 © 华为云计算技术有限公司

```
 slot1     |                2 | queue0: 1005, queue1: 320 | queue0: 63, queue1: 748     | 0/1DCE2578
|            42 |          192927504 | 2023-01-10 11:18:22+08
(1 row)
```

备注：返回值的slot_name代表复制槽名，parallel_decode_num代表该复制槽的并行解码线程数，read_change_queue_length列出了每个解码线程读取日志队列的当前长度，decode_change_queue_length列出了每个解码线程解码结果队列的当前长度，reader_lsn表示当前reader线程读取的日志位置，working_txn_cnt表示当前拼接-发送线程中正在拼接的事务个数，working_txn_memory代表拼接-发送线程中拼接事务占用总内存（单位字节），decoded_time代表该复制槽最新解码到的WAL日志时间。

须知

decoded_time时间来自检查点日志和事务提交日志，存在一定误差。如果没有解码到任何前述包含时间的日志，则显示“2000-01-01 08:00:00+08”(依照数据库设置的时区而定)。

- gs_get_slot_decoded_wal_time(slot_name)

  描述：查看某个复制槽最新解码的WAL日志时间。

  参数说明：

  – slot_name：

    要查询的复制槽名称。

    取值范围：字符串，不支持除字母、数字以及“_”，“?”，“-”，“.”以外的字符。

  示例：

```
gaussdb=# SELECT * FROM gs_get_slot_decoded_wal_time('replication_slot');
 gs_get_slot_decoded_wal_time
------------------------------
 2023-01-10 11:25:22+08
(1 row)
```

  备注：返回一列值代表该复制槽最新解码到的WAL日志时间。

  须知

  - 返回的时间来自检查点日志和事务提交日志，存在一定误差。如果没有解码到任何前述包含时间的日志，则显示“2000-01-01 08:00:00+08”(依照数据库设置的时区而定)。查询一个当前不存在的逻辑复制槽的最新解码的WAL日志时间时，返回NULL，在gsql中NULL显示和设置有关，可以使用\pset null 'null'设置。
  - 不支持串行解码使用。

- gs_logical_parallel_decode_status('slot_name')

  描述：获取并行逻辑解码中活跃复制槽的解码统计信息，包含29行指标。

  指标定义如下：

```
Record - (stat_id int, stat_name TEXT, value TEXT)
```


版权所有 © 华为云计算技术有限公司

表 1-137 指标含义

指标名称	说明
slot_name	逻辑解码任务复制槽名称。
reader_lsn	逻辑解码日志位置。
wal_read_total_time	日志模块加载耗时，单位：微秒。
wal_wait_total_time	日志模块等待耗时，单位：微秒。
parser_total_time	reader线程处理耗时，单位：微秒。
decoder_total_time	各decoder线程处理耗时总值，单位：微秒。
sender_total_time	sender线程处理耗时，单位：微秒。
net_send_total_time	网络发送逻辑日志耗时，单位：微秒。
net_wait_total_time	网络等待发送逻辑日志耗时，单位：微秒。
net_send_total_bytes	网络发送逻辑日志字节数，单位：byte。
transaction_count	事务数量。
big_transaction_count	大事务数量。
max_transaction_tuples	事务操作元组的最大数量。
sent_transaction_count	发送事务数量（本库）。
spill_disk_transaction_count	落盘事务数量。
spill_disk_bytes	累计落盘字节数量，单位：byte。
spill_disk_count	落盘次数。
input_queue_full_count	各decode线程输入队列FULL次数总数。
output_queue_full_count	各decode线程输出队列FULL次数总数。
dml_count	各decode线程解码WAL日志中DML数量（本库）总数。
dml_filtered_count	各decode线程解码过滤WAL日志中DML数量（本库）总数。
toast_count	涉及TOAST表修改行数。
candidate_catalog_xmin	当前逻辑复制槽catalog_xmin候选点。

版权所有 © 华为云计算技术有限公司

指标名称	说明
candidate_xmin_lsn	LSN序逻辑复制槽推进catalog_xmin所需要的日志确认接收点。
candidate_restart_valid	LSN序逻辑复制槽推进restart_lsn所需要的日志确认接收点。
candidate_restart_lsn	当前逻辑复制槽restart_lsn候选点。
candidate_xmin_csn	CSN序逻辑复制槽推进catalog_xmin所需要的日志确认接收点。
candidate_restart_csn	CSN序逻辑复制槽推进restart_lsn所需要的日志确认接收点。
candidate_csn_min	当前逻辑复制槽dictionary_csn_min（参考函数**pg_get_replication_slot...**)的候选点。

参数说明：

- slot_name

  流复制槽名称。

  取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

返回值类型：int、text、text

示例：

```
gaussdb=# SELECT * FROM gs_logical_parallel_decode_status('replication_slot');
 stat_id |          stat_name           |        value
---------+------------------------------+----------------------
       1 | slot_name                    | replication_slot_lsn
       2 | reader_lsn                   | 0/1E249228
       3 | wal_read_total_time          | 4769541
       4 | wal_wait_total_time          | 4754879
       5 | parser_total_time            | 12727
       6 | decoder_total_time           | 215633
       7 | sender_total_time            | 41707
       8 | net_send_total_time          | 28435
       9 | net_wait_total_time          | 11918
      10 | net_send_total_bytes         | 931076
      11 | transaction_count            | 19
      12 | big_transaction_count        | 1
      13 | max_transaction_tuples       | 10000
      14 | sent_transaction_count       | 19
      15 | spill_disk_transaction_count | 0
      16 | spill_disk_bytes             | 0
      17 | spill_disk_count             | 0
      18 | input_queue_full_count       | 226
      19 | output_queue_full_count      | 13
      20 | dml_count                    | 10000
      21 | dml_filtered_count           | 0
      22 | toast_count                  | 0
      23 | candidate_catalog_xmin       | 83576
      24 | candidate_xmin_lsn           | 0/1E0992A0
      25 | candidate_restart_valid      | 0/1E0992A0
      26 | candidate_restart_lsn        | 0/1E0992A0
      27 | candidate_xmin_csn           | 0
      28 | candidate_restart_csn        | 0
```


版权所有 © 华为云计算技术有限公司

```
    29 | candidate_csn_min           | 17157
(29 rows)
```

备注：按照指标定义，指标应该满足下列约束关系：

wal_read_total_time >= wal_wait_total_time;

transaction_count >= big_transaction_count;

transaction_count >= sent_transaction_count;

transaction_count >= spill_disk_transaction_count;

dml_count >= dml_filtered_count;

dml_count >= toast_count;

如果spill_transaction_count == 0，那么 spill_disk_bytes == 0；

但由于严格保证需要频繁加锁解锁，将对性能造成较大影响，故上面约束关系在多线程情况下可能不满足。

transaction_count 统计的是所有库的事务数量。

sent_transaction_count统计的是本库的发送事务数量，因为非本库事务将不会被发送。

当传入不存在的slot_name时，函数不报错，返回值为空。

- gs_logical_parallel_decode_reset_status('slot_name')

  描述：重置gs_logical_parallel_decode_status('slot_name')中的指标。

  参数说明：

  – slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  返回值类型：text

  示例：

```
gaussdb=# SELECT * FROM gs_logical_parallel_decode_reset_status('replication_slot');
 gs_logical_parallel_decode_reset_status
-----------------------------------------
 OK
(1 row)

gaussdb=# SELECT * FROM gs_logical_parallel_decode_status('replication_slot');
 stat_id |          stat_name            |        value
---------+-------------------------------+---------------------
       1 | slot_name                     | replication_slot_lsn
       2 | reader_lsn                    | 0/1E2497F0
       3 | wal_read_total_time           | 0
       4 | wal_wait_total_time           | 0
       5 | parser_total_time             | 0
       6 | decoder_total_time            | 0
       7 | sender_total_time             | 0
       8 | net_send_total_time           | 0
       9 | net_wait_total_time           | 0
      10 | net_send_total_bytes          | 0
      11 | transaction_count             | 0
      12 | big_transaction_count         | 0
      13 | max_transaction_tuples        | 0
      14 | sent_transaction_count        | 0
      15 | spill_disk_transaction_count | 0
      16 | spill_disk_bytes              | 0
      17 | spill_disk_count              | 0
      18 | input_queue_full_count        | 0
      19 | output_queue_full_count       | 0
      20 | dml_count                     | 0
```


版权所有 © 华为云计算技术有限公司

```
    21 | dml_filtered_count           | 0
    22 | toast_count                 | 0
    23 | candidate_catalog_xmin      | 83594
    24 | candidate_xmin_lsn          | 0/1E249528
    25 | candidate_restart_valid     | 0/1E249528
    26 | candidate_restart_lsn       | 0/1E249528
    27 | candidate_xmin_csn          | 0
    28 | candidate_restart_csn       | 0
    29 | candidate_csn_min           | 17176
(29 rows)
```

备注：返回值为执行结果，传入不存在的slot_name时，函数不报错，返回值为invalid slot name。

不允许对正在进行observe的复制槽进行reset操作，异常信息按优先级如下：

a. slot_name为空，报错，显示：ERROR: inputString should not be NULL。

b. slot_name不为空，但不存在，不报错，显示：invalid slot name。

c. slot_name不为空，但当前slot_name对应的复制槽正在被采样（observing），不报错，显示：can't reset during observing! use gs_logical_decode_stop_observe to stop。

- gs_logical_decode_start_observe('slot_name', window, interval)

  描述：开启逻辑复制性能指标采样。分布式需要在DN上执行。

  参数说明：

  – slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  – window

    指定采样窗口大小

    取值范围：整数类型，最小2，最大1024,收集最近interval*window时间内采样数据。

  – interval

    性能监控的间隔，时间间隔类型，秒级。

    取值范围：时间间隔类型，最小1s，最大60s，收集最近interval*window时间内采样数据。

    返回值类型：text

  备注：返回值为执行结果，传入不存在的slot_name时，函数不报错，返回值为invalid slot name。

  不允许对已经开启observe的复制槽进行开启observe操作，异常信息按优先级如下：

  a. slot_name为空，报错，显示：ERROR: inputString should not be NULL。

  b. slot_name不为空，但不存在，不报错，显示：invalid slot name。

  c. 若window小于2，则显示：window has to be >= 2。

  d. 若window大于1024，则显示：window has to be <= 1024。

  e. 若interval小于1s，则显示：sample interval has to be >= 1s。

  f. 若interval大于60s，则显示：sample interval has to be <= 60s。

  g. slot_name不为空，但当前slot_name对应的复制槽已开启observe，不报错，显示：observe has started!

版权所有 © 华为云计算技术有限公司

- gs_logical_decode_stop_observe('slot_name')

  描述：停止逻辑复制性能指标采样。分布式需要在DN上执行。

  参数说明：

  - slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  - 返回值类型：text

  备注：返回值为执行结果，传入不存在的slot_name时，函数不报错，返回值为invalid slot name。

  不允许对已经停止observe的复制槽进行停止observe操作，异常信息按优先级如下：

  a. slot_name为空，报错，显示：ERROR: inputString should not be NULL。
  b. slot_name不为空，但不存在，不报错，显示：invalid slot name。
  c. slot_name不为空，但当前slot_name对应的复制槽已停止observe，不报错，显示：observe not started!

- gs_logical_decode_observe_data('slot_name')

  描述：展示逻辑复制性能指标采样原始数据。分布式需要在DN上执行。

  表 1-138 指标含义

指标名称	说明
slot_name	逻辑解码任务复制槽名称。
sample_time	性能指标采样时间，单位：微秒。
reader_lsn	逻辑解码日志位置。
wal_read_total_time	日志模块加载耗时，单位：微秒。
wal_wait_total_time	日志模块等待耗时，单位：微秒。
parser_total_time	reader线程处理耗时，单位：微秒。
decoder_total_time	各decoder线程处理耗时总值，单位：微秒。
sender_total_time	sender线程处理耗时，单位：微秒。
net_send_total_time	网络发送逻辑日志耗时，单位：微秒。
net_send_total_bytes	网络发送逻辑日志字节数，单位：byte。
transaction_count	事务数量。
big_transaction_count	大事务数量。
sent_transaction_count	发送事务数量（本数据库）。
spill_transaction_count	落盘事务数量。
spill_disk_bytes	累计落盘字节数量，单位：byte。

版权所有 © 华为云计算技术有限公司

参数说明：

- slot_name

  流复制槽名称。

  取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

- 返回值类型：setof record

示例：

```
gaussdb=# SELECT * FROM gs_logical_decode_observe_data('replication_slot');
 slot_name |          sample_time          | reader_lsn | wal_read_total_time | wal_wait_total_time |
parser_total_time | decoder_total_time | sender_total_time | net_send_total_time |
net_send_total_bytes | transaction_count | big_
transaction_count | sent_transaction_count | spill_transaction_count | spill_disk_bytes
-----------+-------------------------------+------------+---------------------+---------------------
+------------------+--------------------+------------------+--------------------+---------------------
+------------------+-----
------------------+------------------------+-------------------------+------------------
 repl      | 2023-01-12 20:09:40.416798+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:09:45.416849+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:09:50.417006+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:09:55.417057+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:00.417115+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:05.417165+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:10.417217+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:15.417271+08 |   49447976 |           776846657 |           776846244
|              65 | {46,11,11,6}       |         56 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:20.417342+08 |   49448264 |           836085882 |           836085442
|              67 | {46,11,11,8}       |         58 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:25.417433+08 |   49448264 |           836085882 |           836085442
|              67 | {46,11,11,8}       |         58 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:30.417526+08 |   49448264 |           836085882 |           836085442
|              67 | {46,11,11,8}       |         58 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:35.417532+08 |   49448264 |           836085882 |           836085442
|              67 | {46,11,11,8}       |         58 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:40.417644+08 |   49448264 |           836085882 |           836085442
|              67 | {46,11,11,8}       |         58 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
 repl      | 2023-01-12 20:10:45.417763+08 |   49448264 |           836085882 |           836085442
|              67 | {46,11,11,8}       |         58 |                0 |                   0 |                0 |
               0 |                      0 |                  0 |           0
(14 rows)
```

备注：传入不存在的slot_name时，函数不报错，返回值为空记录。

- gs_logical_decode_observe('slot_name')

  描述：展示逻辑复制性能指标数据。分布式需要在DN上执行。

版权所有 © 华为云计算技术有限公司

表 1-139 指标含义

指标名称	说明
slot_name	逻辑解码任务复制槽名称。
sample_time	性能指标采样时间。
logical_decode_rate	逻辑解码整体推进速率，单位：byte/s。
wal_read_rate	WAL日志读取阶段处理速率，单位：byte/s。
parser_rate	WAL日志解析阶段处理速率，单位：byte/s。
decoder_rate	解码阶段处理速率，单位：byte/s。
sender_rate	发送阶段处理速率，单位：byte/s。
net_send_rate	网络发送速率，单位：byte/s。

参数说明：

- slot_name

  流复制槽名称。

  取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

- 返回值类型：setof record

示例：

```
gaussdb=# SELECT * FROM gs_logical_decode_observe('replication_slot');
    slot_name     |          sample_time          | logical_decode_rate | wal_read_rate | parser_rate  |
decoder_rate  |  sender_rate  | net_send_rate
------------------+-------------------------------+---------------------+---------------+--------------
+----------------+---------------+---------------
 replication_slot | 2023-01-12 20:16:50.42448+08  | 0.000               | 0.000         | 0.000        |
0.000          | 0.000         | 0.000
 replication_slot | 2023-01-12 20:16:55.424537+08 | 0.000               | 0.000         | 0.000        |
0.000          | 0.000         | 0.000
 replication_slot | 2023-01-12 20:17:00.424641+08 | 0.000               | 0.000         | 0.000        |
0.000          | 0.000         | 0.000
 replication_slot | 2023-01-12 20:17:05.424645+08 | 0.000               | 0.000         | 0.000        |
0.000          | 0.000         | 0.000
 replication_slot | 2023-01-12 20:17:10.424795+08 | 0.000               | 0.000         | 0.000        |
0.000          | 0.000         | 0.000
 replication_slot | 2023-01-12 20:17:15.424848+08 | 0.000               | 0.000         | 0.000        |
0.000          | 0.000         | 0.000
 replication_slot | 2023-01-12 20:17:20.424849+08 | 57.600              | 699029.126    | 96000000.000 |
1152000000.000 | 144000000.000 | 0.000
 replication_slot | 2023-01-12 20:17:25.424959+08 | 0.000               | 699029.126    | 96000000.000 |
1152000000.000 | 144000000.000 | 0.000
 replication_slot | 2023-01-12 20:17:30.42496+08  | 0.000               | 699029.126    | 96000000.000 |
1152000000.000 | 144000000.000 | 0.000
 replication_slot | 2023-01-12 20:17:35.425059+08 | 0.000               | 699029.126    | 96000000.000 |
1152000000.000 | 144000000.000 | 0.000
```

备注：传入不存在的slot_name时，函数不报错，返回值为空记录。若分母为0，则返回最近一次有效数据。若分母不为0，分子为0，则返回0。

指标计算公式：

- logical_decode_rate = (reader_lsn1 - reader_lsn2) / (sample_time1 - sample_time2)

版权所有 © 华为云计算技术有限公司

  - wal_read_rate = (reader_lsn1 - reader_lsn2) / (wal_read_total_time1 - wal_read_total_time2) - (wal_wait_total_time1 - wal_wait_total_time2)
  - parser_rate = (reader_lsn1 - reader_lsn2) / (parser_total_time1 - parser_total_time2)
  - decoder_rate = (reader_lsn1 - reader_lsn2) / avg(decoder_total_time1[i]-decoder_total_time2[i])
  - sender_rate = (reader_lsn1 - reader_lsn2) / (sender_total_time1 - sender_total_time2) - (net_send_total_time1 - net_send_total_time2)
  - net_send_rate = (net_sent_bytes1 - net_sent_bytes2) / (net_send_total_time1 - net_send_total_time2) - (net_wait_total_time1 - net_wait_total_time2)

- gs_logical_decode_observe_status('slot_name')

  描述：查询指定逻辑解码任务的监控状态。分布式需要在DN上执行。

  参数说明：

  - slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  - 返回值类型：text

  示例：

```
gaussdb=# SELECT * FROM gs_logical_decode_observe_status('replication_slot');
 gs_logical_decode_observe_status
----------------------------------
 START
(1 row)

gaussdb=# SELECT * FROM gs_logical_decode_observe_status('replication_slot');
 gs_logical_decode_observe_status
----------------------------------
 invalid slot name
(1 row)

gaussdb=# SELECT * FROM gs_logical_decode_stop_observe('replication_slot');
 gs_logical_decode_stop_observe
--------------------------------
 OK
(1 row)

gaussdb=# SELECT * FROM gs_logical_decode_observe_status('replication_slot');
 gs_logical_decode_observe_status
----------------------------------
 STOP
(1 row)
```

  备注：返回值为逻辑解码任务的监控状态，传入不存在的slot_name时，函数不报错，返回值为invalid slot name。

- gs_get_parallel_decode_thread_info()

  描述：在并行解码所在的DN执行，返回当前DN上并行解码的线程信息。

  返回值类型：int64、text、text、int

  示例：

```
gaussdb=# SELECT * FROM gs_get_parallel_decode_thread_info();
    thread_id    | slot_name | thread_type | seq_number
-----------------+-----------+-------------+------------
 140335364699904 | slot1     | sender      |          1
 140335214098176 | slot1     | reader      |          1
```


版权所有 © 华为云计算技术有限公司

```
140335325312768 | slot1    | decoder    |         1
140335291750144 | slot1    | decoder    |         2
140335274968832 | slot1    | decoder    |         3
140335258187520 | slot1    | decoder    |         4
140335165404928 | slot2    | sender     |         1
140335022864128 | slot2    | reader     |         1
140335129818880 | slot2    | decoder    |         1
140335113037568 | slot2    | decoder    |         2
(10 rows)
```

备注：返回值thread_id代表线程id，slot_name代表复制槽名，thread_type表示线程种类（共三种，sender代表发送线程、reader代表读取线程、decoder代表解码线程），seq_number代表每个线程在当前复制槽中同种线程的序号。其中sender和reader在每个并行解码连接中均只有一个，因此序号均为1，decoder的序号从1排列到当前复制槽解码并行度。

- gs_get_logical_decode_parameter()

  描述：查询活跃的并行逻辑解码任务参数选项。

  表 1-140 参数含义

字段名称	返回值类型	说明
slot_name	text	复制槽名。
include_xids	bool	解码信息是否包含xid。
include_timestamp	bool	解码信息是否包含时间戳。
skip_empty_xacts	bool	解码结果是否忽略空事务。
only_local	bool	是否只解码当前数据库产生的数据。
include_username	bool	事务的BEGIN日志是否包含用户名。
enable_ddl_decoding	bool	是否开启DDL语句的解码。
enable_ddl_json_format	bool	是否开启DDL的反解析json输出形式。
max_txn_in_memory	int64	事务占用内存最大值（该参数已废弃，不生效）。
max_reorderbuffer_in_memory	int64	当前解码任务最大内存值。
decode_style	char	解码格式。
parallel_decode_num	int64	并行解码的decoder线程数量。
sending_batch	int64	是否批量发送。
parallel_queue_size	int64	并行逻辑解码线程间进行交互的队列长度。
enable_heartbeat	bool	是否输出心跳日志。
slice_id	int64	复制表分片id。
output_start_csn	int64	开始输出的CSN位置。

版权所有 © 华为云计算技术有限公司

字段名称	返回值类型	说明
output_start_lsn	int64	开始输出的LSN位置。
logical_reader_bind_cpu	int64	reader线程绑定CPU的编号。
logical_decoder_bind_cpu_index	int64	decoder线程绑定CPU的索引。
auto_advance	bool	是否自动推进。
skip_generated_columns	bool	是否跳过生成列的结果输出。
big_transaction_limit	int64	判断是否为大事务的内存使用限额（单位字节）。
exclUser	text	黑名单用户的名称列表。
table_white_list	text	白名单表参数。
desc_memory_limit	int64	逻辑解码任务维护的表元信息总内存（单位MB）。
max_logical_decode_memory	int64	并行解码任务的总内存（单位GB）。
max_control_memory_size	int64	参数max-logical-decode-memory的上限值（单位GB）。
max_queue_cache_num	int64	参数max-decode-to-sender-cache-num的上限值。
max_toast_memory_size	int64	读取Xlog中toast字段的内存使用上限值（单位MB）。
restart_lsn	int64	逻辑解码控制参数，用于指定解码开始点。
timezone_is_utc	bool	逻辑解码控制参数，用于控制携带时区的时间类型数据的输出。
enable_decode_position	bool	逻辑解码控制参数，该参数主要控制逻辑解码读取到解码位置日志时，是否发送该位置日志信息。
standby_connection	bool	仅流式解码设置，是否仅限制备机解码。
sender_timeout	int64	仅流式解码设置，GaussDB与客户端的心跳超时阈值（单位毫秒）。
change_log_max_len	int64	逻辑日志缓存长度上限参数，单位为字节。
max_decode_to_sender_cache_num	int64	并行解码日志的缓存条数阈值。

版权所有 © 华为云计算技术有限公司

字段名称	返回值类型	说明
output_order	bool	仅流式解码设置有效，代表是否使用CSN顺序输出解码结果。

示例：

```
gaussdb=# SELECT * FROM gs_get_logical_decode_parameter();
 slot_name |
configs
                  -----------
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-----------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
---------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
---------------------------
-------------------------------------------------------------------------------------
 slot_1    | include_xids: 1, include_timestamp: 0, skip_empty_xacts: 0, only_local: 1, include_username:
0, enable_ddl_decoding: 1, enable_ddl_json_format: 0, max_txn_in_memory: 3072,
max_reorderbuffer_in_memory: 1, decode_style: b, parallel_decode_num: 2,
sending_batch: 0, parallel_queue_size: 128, enable_heartbeat: 0, slice_id: -1, output_start_csn: 0,
output_start_lsn: 0, logical_reader_bind_cpu: -1, logical_decoder_bind_cpu_index: -1, auto_advance: 0,
skip_generated_columns: 0, big_transaction_limit: 10, desc_memory_limit: 200,
max_logical_decode_memory: 6, max_control_memory_size: 10, max_queue_cache_num: 16384,
max_toast_memory_size: 100,  restart_lsn: 274660168, timezone_is_utc: 1, enable_decode_position: 1,
standby_connection: 0, sender_timeout: 10, change_log_max_len: 8192,
max_decode_to_sender_cache_num: 10000, output_order: 0, exclUser: , table_white_list:
(1 row)
```

- gs_logical_get_transaction_table_info('start_lsn', 'upto_lsn', 'xid', 'options_name', 'options_value')

  描述：指定LSN区间和xid，解析Xlog获取表相关信息。打开三权分立时，仅数据库初始用户可以调用；关闭三权分立时，具备系统管理员权限的用户可以调用。

说明

约束条件如下：

- 当前网络和硬件环境正常。
- 调用函数时，要求日志级别wal_level=logical，且只有在wal_level=logical期间产生的日志文件才能被解析，如果使用的Xlog文件为非logical级别，则解析内容没有对应的值和类型，无其他影响。如果wal_level未被设置成logical级别，则报错不解析。
- 此解析方式为根据Xlog文本记录数据进行解析，因此不在此Xlog里的数据无法解析。若相关Xlog日志缺失，缺失日志也无法解析。
- enable_decode_parent_xid不打开时，从start_lsn点开始，只解析该事务相关的Xlog，其父事务、子事务等都不解析。
- enable_decode_parent_xid打开时，从start_lsn点开始解析，解析输出仅包含该子事务及后续产生的事务信息。因需解析至子事务方可确立父子关系，该子事务之前的父事务及同层级其他子事务信息不予输出。如果需要完整顶层事务信息，请将事务ID指定为顶层事务ID，并合理设置start_lsn。
- 不支持获取relfilenode已变化场景的表名称，例如表被TRUNCATE/DROP等。
- 不支持获取Schema、表名已经变化的历史版本名称。
- 执行函数时，输出文件大小超过100MB将被截断并报错。
- xid小于Xlog中事务最小值时不报错，返回文件内容为空。
- 当start_lsn检索范围过大，或设置为NULL（即从头遍历）时，Xlog解析期间可能因底层Xlog日志已被回收而中断，建议调整start_lsn范围后重新执行。

参数说明：

- start_lsn

  指定开始解析的LSN。

  取值范围：字符串（LSN，格式为xlogid/xrecoff），如'1/2AAFC60'。为NULL时表示不对解析起始的日志位置做限制。

- upto_lsn

  指定解析结束的LSN。

  取值范围：字符串（LSN，格式为xlogid/xrecoff），如'1/2AAFC60'。为NULL时表示不对解析截止的日志位置做限制。

- xid

  事务ID。

  取值范围：3~18446744073709551615（即uint64最大值）。

  说明

  如果用户设置的xid超过当前数据系统使用的xid的最大值，则会提示输入事务ID非法。

- options：可选参数，由一系列options_name和options_value一一对应组成，可以缺省。

  - enable_decode_parent_xid

    是否解析出的指定xid的父事务以及子事务信息。

    取值类型：布尔型。

    取值范围：

    - false：设为false时，解析出的指定事务的表信息。
    - true：设为true时，解析出指定事务的父事务以及相关子事务信息。

版权所有 © 华为云计算技术有限公司

默认值：false。

说明

仅505.2.0以后的版本产生的Xlog日志支持构建父子事务关系。

返回值类型：text

示例：

```
gaussdb=# SELECT * FROM gs_logical_get_transaction_table_info('0/202FC320', '0/20306EE8',
'221557', 'enable_decode_parent_xid', '1');
                                    output_filepath
-------------------------------------------------------------------------------------
 /home/omm/install/clusters/clusters_cent/cxz/cxz_datanode/221557_202fc320_20306ee8.log
(1 row)
解析结果如下:
=== DETAIL RECORDS ===
XID|TOP_XID|PARENT_XID|RELFILE_OID|START_LSN|END_LSN|CSN|TABLE_OID|SCHEMA_NAME|
TABLE_NAME|OPERATION|ADDITIONAL_INFO
9982031|9982031|0|324521|9/297251C8|9/29725260|9417216|324521|public|t_tran|INSERT|
9982032|9982031|9982031|324521|9/297254E8|9/29725578|9417216|324521|public|t_tran|INSERT|
9982034|9982031|9982032|324521|9/29725B80|9/29725C18|9417217|324521|public|t_tran|INSERT|
9982041|9982031|9982031|324521|9/29727940|9/297279D0|9417221|324521|public|t_tran|INSERT|
9982045|9982031|9982041|324521|9/2974C458|9/2974C4F0|9417224|324521|public|t_tran|INSERT|
9982047|9982031|9982041|324521|9/2974CC00|9/2974CC98|9417225|324521|public|t_tran|INSERT|
=== SUMMARY RECORDS ===
XID|TABLE_OID|SCHEMA_NAME|TABLE_NAME|IS_COMMITTED|INSERT_COUNT|UPDATE_COUNT|
DELETE_COUNT|DDL_COUNT
9982031|324521|public|t_tran|true|6|0|0|0
```

表 1-141 指标含义

指标名称	说明
XID	事务号。
TOP_XID	顶层事务号。
PARENT_XID	父事务号。
RELFILE_NODE	表文件对象标识符。
START_LSN	该事务起始LSN。
END_LSN	该事务结束LSN。
CSN	提交序列号。
TABLE_OID	表的对象标识符。
SCHEMA_NAME	表所属Schema。
TABLE_NAME	表名。
OPERATION	操作类型。
ADDITIONAL_INFO	DDL信息。
IS_COMMITTED	事务是否提交。
INSERT_COUNT	插入记录数。

版权所有 © 华为云计算技术有限公司

指标名称	说明
UPDATE_COUNT	更新记录数。
DELETE_COUNT	删除记录数。
DDL_COUNT	DDL语句数。

- gs_get_distribute_decode_status()

  描述：获取当前节点上分布式解码状态细节（以复制槽为粒度），需在CN上执行，DN上执行仅返回空。

  返回值类型：text、int、int、int64、xid、xid、text、text、text

```
gaussdb=# SELECT * FROM gs_get_distribute_decode_status();
 slot_name | logical_receiver_num | slice_num | walsender_thread_id | last_sent_csn |
last_confirmed_csn |       receiver_queue_length       |          connect_times          |
csn_receive_array
-----------+----------------------+-----------+---------------------+---------------+--------------------
+---------------------------------+---------------------------------
+-----------------------------------------------------
 slot1     |                    1 |         3 |     139958481843968 |       2012169 |             2010107 | queue0: 1,
queue1: 1, queue2: 0 | slice0: 2, slice1: 2, slice2: 2 | slice0: 2012244, slice1: 2012244, slice2: 2012244
(1 row)
```

  表 1-142 指标含义

指标名称	类型	说明
slot_name	text	复制槽名。
logical_receiver_num	int	分布式解码启动的receiver线程数量。
slice_num	int	集群分片数。
walsender_thread_id	int64	walsender线程id。
last_sent_csn	xid	最近发送的CSN。
last_confirmed_csn	xid	最后从客户端返回的已确认接收的CSN。
receiver_queue_length	text	各DN接收日志队列长度（拼接后以字符串格式显示）。
connect_times	text	各DN连接次数（拼接后以字符串格式显示）。
csn_receive_array	text	各DN获取到的最新CSN（拼接后以字符串格式显示）。

- gs_logical_decode_meta_cache_summary('slot_name')

  描述：获取并行逻辑解码中活跃复制槽的解码元信息缓存的统计信息，包括缓存对象数、版本数、总内存占用、缓存命中率、扫描系统表命中率等全局指标。

  参数说明：

  – slot_name

    流复制槽名称。

版权所有 © 华为云计算技术有限公司

取值类型：字符串。

取值范围：仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

表 1-143 指标含义

指标名称	说明
type	逻辑解码元信息的类型（支持table、partition、type、enum、auth、database）。
entry_num	元信息实体个数。
version_num	元信息实体版本总数（每个实体可能有多个版本）。
total_mem_size	元信息缓存总内存，单位：字节。
hit_count	缓存命中次数。
miss_count	缓存未命中次数。
hit_ratio	缓存命中率。
scan_count	扫描系统表次数。
scan_hit_count	扫描系统表命中次数。
scan_hit_ratio	扫描系统表命中率。
extra_info	其它信息。

示例：

```
gaussdb=# SELECT * FROM gs_logical_decode_meta_cache_summary('slot1');
   type    | entry_num | version_num | total_mem_size | hit_count | miss_count | hit_ratio | scan_count |
scan_hit_count | scan_hit_ratio | extra_info
-----------+-----------+-------------+----------------+-----------+------------+-----------+------------
+----------------+----------------+------------
 table     |        53 |          53 |         173056 |         0 |          0 |         0 |          0 |
           0 |
 partition |         0 |           0 |              0 |         0 |          0 |         0 |          0 |
           0 |              0 |              0 |
 type      |        14 |          14 |           1791 |         0 |          0 |         0 |          0 |
           0 |              0 |              0 |
 enum      |         0 |           0 |              0 |         0 |          0 |         0 |          0 |
           0 |              0 |              0 |
 auth      |         1 |           1 |            136 |         0 |          0 |         0 |          0 |
           0 |              0 |              0 |
 database  |         0 |           0 |              0 |         0 |          0 |         0 |          0 |
           0 |              0 |              0 |
(6 rows)
```

- gs_logical_decode_meta_cache_stats('slot_name', 'type', 'mode', 'limit_num')

  描述：获取并行逻辑解码中活跃复制槽的解码元信息缓存指定维度的TopN对象统计信息，支持按内存占用（memory）、版本链长度（version_cnt）、未命中次数（miss_count）、命中率（hit_ratio）维度查询。

  参数说明：

  – slot_name

    流复制槽名称。

    取值类型：字符串。

版权所有 © 华为云计算技术有限公司

取值范围：仅支持小写字母、数字以及“_”、“?”、“-”、“.”字符，且不支持“.”或“..”单独作为复制槽名称。

- type

  元信息类型名称。

  取值范围：支持table_oid、table_rfn、partition、type、enum、auth、database。

- mode

  查询维度信息。

  取值范围：支持memory、version_cnt、miss_count、hit_ratio。

- limit_num

  输出行数限制大小。

  取值类型：整型。

  取值范围：1 ~ 1000。

表 1-144 指标含义

指标名称	说明
oid	元信息实体的oid。
name	元信息实体的名称。
relfilenode	元信息实体的relfilenode。
version_cnt	元信息实体的版本个数。
invalid_version_cnt	元信息实体的失效版本个数。
total_mem_size	元信息实体的占用内存的总大小，单位：字节。
avg_version_size	元信息实体各版本占用内存的平均大小，单位：字节。
hit_count	元信息实体的命中次数。
miss_count	元信息实体的未命中次数。
hit_ratio	元信息实体的命中率。
create_time	元信息实体加载到内存中的时间。
evict_fail_count	元信息实体的淘汰失败次数。
evict_count	元信息实体的淘汰次数。
latest_evict_time	元信息实体的最近一次淘汰的时间。
warning_flags	元信息实体异常标记信息。
extra_info	其他信息。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT * FROM gs_logical_decode_meta_cache_stats('slot1', 'table_oid', 'memory', 20);
  oid  |                name                | relfilenode | version_cnt | invalid_version_cnt | total_mem_size |
avg_version_size | hit_count | miss_count | hit_ratio |          create_time          | evict_fail_count |
evict_count | latest_evict_time | warning_flags | extra_info
-------+------------------------------------+-------------+-------------+---------------------+----------------
+------------------+-----------+------------+-----------+-------------------------------+------------------
+-------------+-------------------+---------------+------------
 16609 | snap_summary_statement             |       16609 |           1 |                   0 |           6544
|           6544 |         3 |         0 |         1 | 2026-07-01 10:55:57.313943+08 |                0 |           0
|                |           0 |
 16616 | snap_global_statement_count        |       16616 |           1 |                   0 |           3424
|           3424 |         3 |         0 |         1 | 2026-07-01 10:55:57.31703+08  |                0 |           0
|                |           0 |
 16622 | snap_summary_statement_count       |       16622 |           1 |                   0 |           3424
|           3424 |         3 |         0 |         1 | 2026-07-01 10:55:57.319207+08 |                0 |           0
|                |           0 |
 16758 | snap_global_stat_all_tables        |       16758 |           1 |                   0 |           2904
|           2904 |         3 |         0 |         1 | 2026-07-01 10:55:54.406154+08 |                0 |           0
|                |           0 |
 16664 | snap_global_redo_status            |       16664 |           1 |                   0 |           2800
|           2800 |         0 |         0 |         0 | 2026-07-01 10:55:54.385334+08 |                0 |           0
|                |           0 |
 16764 | snap_summary_stat_all_tables       |       16764 |           1 |                   0 |           2800
|           2800 |         3 |         0 |         1 | 2026-07-01 10:55:54.408911+08 |                0 |           0
|                |           0 |
(6 rows)
```

须知

函数执行过程中需要访问解码元信息缓存的Hash表并获取锁，以保证统计数据一致性。执行过程中可能与逻辑解码线程存在锁竞争，对解码性能造成影响。建议避免频繁调用该函数，推荐在问题定位、性能分析等场景下使用。

- gs_logical_decode_meta_cache_obj_detail('slot_name', 'type', oid, 'mode', 'limit_num')

  描述：获取并行逻辑解码中活跃复制槽的解码元信息缓存指定对象的详细统计信息和版本链详情，包括版本数、有效区间（csnmin/csnmax）、内存占用、命中情况等。

  参数说明：

  - slot_name

    流复制槽名称。

    取值类型：字符串。

    取值范围：仅支持小写字母、数字以及“_”、“?”、“-”、“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  - type

    元信息类型名称。

    取值范围：支持table、partition、type、enum、auth、database。

  - oid

    查询元信息的oid。

  - mode

    查询结果输出模式。

    取值范围：支持memory、file。

版权所有 © 华为云计算技术有限公司

- limit_num

  输出行数限制大小。

  取值类型：整型。

  取值范围：当mode为memory时取值1 ~ 1000；当mode为file时取值1 ~ 2147483647。

表 1-145 指标含义

指标名称	说明
oid	元信息版本的oid。
relfilenode	元信息版本的relfilenode。
name	元信息版本的名称。
csnmin	元信息版本的csnmin信息。
csnmax	元信息版本的csnmax信息。
is_invalid	元信息版本是否有效。
mem_size	元信息版本的内存占用大小，单位：字节。
hit_count	元信息版本的命中次数。
refcnt	元信息版本的引用计数。
createtime	元信息版本加载到内存中的时间。
extra_info	其他信息，当mode=file时，表示输出文件路径。

示例：

```
gaussdb=# SELECT * FROM gs_logical_decode_meta_cache_obj_detail('slot1', 'table', 54921, 'memory',
100);
  oid   | relfilenode | name | csnmin | csnmax | is_invalid | mem_size | hit_count | refcnt |
createtime           | extra_info
-------+-------------+------+--------+--------+------------+----------+-----------+--------
+-----------------------------+------------
 54921 |       54921 | t1   | 409737 |      0 | f          |      408 |         0 |      0 | 2026-07-01
10:55:52.27585+08 |
(1 row)
gaussdb=# SELECT * FROM gs_logical_decode_meta_cache_obj_detail('slot1', 'table', 54921, 'file', 100);
 oid | relfilenode | name | csnmin | csnmax | is_invalid | mem_size | hit_count | refcnt | createtime
|                                    extra_info
-----+-------------+------+--------+--------+------------+----------+-----------+--------+------------
+-------------------------------------------------------------------------------------------
   0 |           0 |      |      0 |      0 | f          |        0 |         0 |      0 |            | <DataDir>/
meta_cache_slot1_1782874593.out
(1 row)
```

须知

函数执行过程中需要访问解码元信息缓存的Hash表并获取锁，以保证统计数据一致性。执行过程中可能与逻辑解码线程存在锁竞争，对解码性能造成影响。建议避免频繁调用该函数，推荐在问题定位、性能分析等场景下使用。

版权所有 © 华为云计算技术有限公司

- gs_get_distribute_decode_status_detail()

  描述：获取当前节点上分布式解码状态细节（以DN为粒度），需在CN上执行，DN上执行仅返回空。

  返回值类型：text、int、int64、int、int、xid

```
gaussdb=# SELECT * FROM gs_get_distribute_decode_status_detail();
 slot_name | slice_id |     thread_id     | queue_len | connect_times | received_csn
-----------+----------+-------------------+-----------+---------------+--------------
 slot1     |        0 | 139959895848704 |         1 |             2 |      2012244
 slot1     |        1 | 139959895848704 |         1 |             2 |      2012244
 slot1     |        2 | 139959895848704 |         0 |             2 |      2012244
(3 rows)
```

  表 1-146 指标含义

指标名称	类型	说明
slot_name	text	复制槽名。
slice_id	int	分片号（从0开始）。
thread_id	int64	分布式解码启动的receiver线程号。
queue_len	int	当前DN接收日志队列长度。
connect_times	int	各DN连接次数。
received_csn	xid	从当前DN获取的最新CSN。

- gs_logical_dictionary_baseline()

  描述：执行逻辑解码数据字典的存量数据基线化，执行成功返回耗时，执行失败返回失败原因。

  返回值类型：text

- gs_logical_dictionary_disabled()

  描述：关闭逻辑解码数据字典功能，停止增量DDL，执行成功返回OK，执行失败返回失败原因。

  返回值类型：text

  注意

  关闭数据字典功能之后，无法使用已有的数据字典模式复制槽继续解码，如需使用数据字典功能，须执行gs_logical_dictionary_baseline()函数，对逻辑数据字典重新进行基线化，然后重建逻辑复制槽。

- gs_logical_decode_lock()

  描述：逻辑解码组件升级管理加锁接口，加锁成功返回t，失败返回f。内部使用功能，不建议用户使用。

  参数说明：wait_time：加锁等待时间。取值范围：非负数，单位：秒；支持不传入参，不传入参，加锁等待时间为0秒，即不等待。

  返回值类型：boolean

- gs_logical_decode_unlock()

版权所有 © 华为云计算技术有限公司

描述：逻辑解码组件升级管理放锁接口，无入参，放锁成功返回t，失败返回f。内部使用功能，不建议用户使用。

返回值类型：boolean

- gs_add_logical_decoding_position_xlog()

  描述：执行该函数后会生成XLOG_STANDBY_DECODE_POSITION类型的Xlog日志。该日志会被当前数据库上的解码任务读取，用于反馈解码位置。具有sysadmin、replication或opradmin权限的用户才能调用该函数，否则调用失败。该函数执行频率依据数据库统计，按节点控制每个数据库调用该系统函数的频率，调用最小间隔为1秒。分布式场景也遵循此原则，每个CN单独控制调用频率。

  返回值类型：text

- gs_change_replication_slot_plugin('slot_name', 'plugin_name')

  描述：修改逻辑复制槽插件类型。

  参数说明：

  – slot_name

    流复制槽名称。

    取值范围：字符串，仅支持小写字母、数字以及“_”，“?”，“-”，“.”字符，且不支持“.”或“..”单独作为复制槽名称。

  – plugin_name

    插件名称。

    取值范围：字符串，当前支持mppdb_decoding、sql_decoding、parallel_binary_decoding、parallel_json_decoding、parallel_text_decoding。

    返回值类型：boolean，默认值为true。

  示例：

  查询当前复制槽信息：

```
gaussdb=# SELECT * FROM pg_get_replication_slots();
 slot_name |          plugin          | slot_type | datoid | active | xmin | catalog_xmin | restart_lsn |
dummy_standby | confirmed_flush | confirmed_csn | dictionary_csn_min | slot_dictionary_type
-----------+--------------------------+-----------+--------+--------+------+--------------+-------------
+--------------+-----------------+---------------+--------------------+----------------------
 slot1     | parallel_json_decoding   | logical   |  12834 | f      |      |              | 0/125A15E0  | f            |
0/125E84D0      |              |              7787 | dictionary table
 slot2     | parallel_binary_decoding | logical   |  12834 | f      |      |              | 0/E9C71B0   | f            |
0/E9CA638       |              |              3860 | dictionary table
 slot3     | parallel_text_decoding   | logical   |  12834 | f      |      |              | 0/E9D96D0   | f            | 0/
EEE6508       |              |              3867 | dictionary table
 slot4     | mppdb_decoding           | logical   |  12834 | f      |      |              | 0/E9AFC38   | f            | 0/
EEE6508       |              |              3853 | dictionary table
(4 rows)
```

  修改复制槽类型：

```
gaussdb=# SELECT * FROM gs_change_replication_slot_plugin('slot1', 'mppdb_decoding');
 gs_change_replication_slot_plugin
-----------------------------------
 t
(1 row)
```

  再次查询复制槽信息：

```
gaussdb=# SELECT * FROM pg_get_replication_slots();
 slot_name |          plugin          | slot_type | datoid | active | xmin | catalog_xmin | restart_lsn |
dummy_standby | confirmed_flush | confirmed_csn | dictionary_csn_min | slot_dictionary_type
-----------+--------------------------+-----------+--------+--------+------+--------------+-------------
+--------------+-----------------+---------------+--------------------+----------------------
```


版权所有 © 华为云计算技术有限公司

```
 slot1     | mppdb_decoding          | logical   |  12834 | f      |      |            | 0/125A15E0  | f           |
0/125E84D0      |              |             7787 | dictionary table
 slot2     | parallel_binary_decoding | logical   |  12834 | f      |      |            | 0/E9C71B0   | f           |
0/E9CA638       |              |             3860 | dictionary table
 slot3     | parallel_text_decoding   | logical   |  12834 | f      |      |            | 0/E9D96D0   | f           | 0/
EEE6508       |              |             3867 | dictionary table
 slot4     | mppdb_decoding          | logical   |  12834 | f      |      |            | 0/E9AFC38   | f           | 0/
EEE6508       |              |             3853 | dictionary table
(4 rows)
```

- gs_get_inter_cluster_version_info()

  描述：查询主备数据库的版本号、逻辑解码组件的版本以及复制类型（逻辑复制或者物理复制）。当前版本暂不支持。

  返回值类型：int、text、int、text、text

1.6.27.11 段页式存储函数

说明

在存储引擎下创建段页式表，需要在创建数据表时指定参数segment=on。例如：

```
CREATE TABLE t1(id int) WITH (segment=on, storage_type=astore);
CREATE TABLE t1(id int) WITH (segment=on, storage_type=ustore);
```

- 段页式存储函数相关字段取值说明：

  a. forknum：数据文件分支。

  取值范围：[0：mainfork；1：fsmfork； 2：vm fork]。

  b. file id：数据文件编号。

  取值范围：[1：元数据文件；2 ~ 5：数据文件]。

  c. blocks：扩展大小。

  取值范围：[1：1号文件；8：2号文件；128：3号文件；1024：4号文件；4096：5号文件]。

  d. file_block_id/head_block_id/block_id：物理页面在数据文件中的偏移页面号，以及其他含义为页面号的字段皆为此意。

  取值范围：[0 ~ 4,294,967,294]。

  e. page_ type：页面类型。

  取值范围：

  元页面：file head/file_header：文件头；spc head/spc_header：空间头；map head/map_header：映射头；map page/map_pages：映射页面；reverse pointer page/inverse pointer page/ip pages/：反向指针页面；segment head page/segment head：段头页面；level1 page：level1页面；data_pages/data extent：数据页面；fork head：分支头。

  数据页面：heap、uheap、btree、ubtree。

  未知页面：unknown(data extent)：全零段页式页面，无法判断页面类型；unknown(fsm indexurq)：fsm或indexurq 页面。

  f. contents：数据文件的存储内容。

  取值范围：[permanent：永久；unlogged：不记录日志；temporary：全局临时；temporary2：本地临时]。

- local_space_shrink(tablespacename TEXT, databasename TEXT)

  描述：当前节点上对指定段页式空间做物理空间收缩。目前只支持对当前连接的database做shrink。

  返回值：空

版权所有 © 华为云计算技术有限公司

- gs_space_shrink(tablespace int4, database int4, extent_type int4, forknum int4)

  描述：效果跟local_space_shrink类似，对指定段页式空间做物理空间收缩，但参数不同，传入的是tablespace和database的oid，extent_type为[2,5]的int值。extent_type为 1表示段页式元数据，目前不支持对元数据所在的物理文件做收缩。该函数仅限工具使用，不建议用户直接使用。

  返回值：空

- global_space_shrink(tablespacename TEXT, databasename TEXT)

  描述：在cn上执行，对整个集群上所有dn执行段页式存储空间压缩。

  注意：global_space_shrink 锁cluster，在此期间不能执行DDL操作。而local_space_shrink不会锁集群。

- gs_seg_dump_page(tablespace_name name, file_id int4, bucketnode int4, file_block_id bigint, forknum int4 default 0)

  描述：解析段页式的指定页面并返回解析内容。只有具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以执行。该函数为物理页面解析方式，每次返回一个页面的解析结果（返回的结果中不会包含实际的用户数据信息）。本函数不要求用户输入页面类型，在实现时，先尝试确定页面类型；如果不能确定，则输出可能的解析结果。

  参数说明：

名称	类型	描述
table_space_name	NAME	段对象所属的表空间。取值范围：合法存在的表空间name。
file_id	INTEGER	数据文件标识。取值范围：[1, 5]的int4值。
bucketnode	INTEGER	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
file_block_id	BIGINT	物理页面在数据文件中的偏移页面号。取值范围：[0~4294967294]。
forknum	INTEGER DEFAULT 0	数据文件fork号，默认为0。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。

  返回值类型：

版权所有 © 华为云计算技术有限公司

名称	类型	描述
page_type	TEXT	页面类型。取值范围： ● 业务页面：heap、uheap、btree、ubtree、gsivfflat_index、gsdiskann_index。 ● 段页式元页面：bucket main head、file head、spc head、map head、map page、reverse pointer page、segment head page、level1 page。 ● 未知页面： – unknown(data extent)：全零段页式页面, 无法判断页面类型。 – unknown(fsm indexurq)：fsm或indexurq页面。
result	TEXT	解析结果。

例如（下列操作需要在创建过段页式普通表后再执行）：

```
gaussdb=# SELECT * FROM gs_seg_dump_page('pg_default', 1, 1024, 4157);
    page_type     |                                  result
------------------+-----------------------------------------------------------------------------
 segment head page | Page information of block 4157/4157                                        +
                   |        pd_lsn: 0/2C90608 ,len 8 ,offset 0                                  +
                   |                                                                            +
                   |        pd_checksum: 0x8A7F, verify success,len 2, offset 8                 +
                   |        pd_flags:                                                           +
                   |        pd_lower: 24, empty, len 2, offset 12                               +
                   |        pd_upper: 8192, old, len 2, offset 14                               +
                   |        pd_special: 8192, size 0, len 2, offset 16                          +
                   |        Page size & version: 8192, 8, len 2, offset 18                      +
                   |        pd_xid_base: 0, len 8, offset 24 pd_multi_base: 0, len 8, offset 32+
                   |        pd_prune_xid: 0, len 4 ,offset 20                                   +
                   |                                                                            +
                   |        Segment head information on this page                               +
                   |              magic: 44414548544E454D                                       +
                   |              lsn is: 0/2C90540                                             +
                   |              nblocks: 0                                                    +
                   |              total_blocks: 8                                               +
                   |              reserved: 0                                                   +
                   |              Level 0 slots information on this page                        +
                   |                    The BlockNumber of level0 slots 0 is: 4157              +
                   |              fork head information on this page                            +
                   |                    4157(valid)                                             +
                   |                    4294967295(invalid)                                     +
                   |                    4294967295(invalid)
(1 row)
```

- gs_seg_dump_page(relid oid, bucketnode int, block_id bigint, partition bool default false, forknum int4 default 0)

  描述：在DN上执行，解析段页式的指定页面并返回解析内容。只有具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以执行。该函数为逻辑页面解析方式，每次返回一个页面的解析结果（返回的结果中不会包含实际的用户数据信息）。本函数不要求用户输入页面类型，在实现时，先尝试确定页面类型；如果不能确定，则输出可能的解析结果。

  参数说明：

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relid	OID	段对象标识。取值范围：合法存在的段页式对象oid，否则报错。
bucketnode	INTEGER	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
block_id	BIGINT	逻辑页面号。取值范围：[0~4294967294]。
partition	BOOLEAN DEFAULT FALSE	段对象是否为分区，默认为false。取值范围：bool，代表所传入的oid对应的对象是否为分区。
forknum	INTEGER DEFAULT 0	数据文件fork号，默认为0。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。

返回值说明：

名称	类型	描述
page_type	TEXT	页面类型。取值范围： ● 业务页面：heap、uheap、btree、ubtree。 ● 段页式元页面：bucket_main_head、file head、spc head、map head、map page、reverse pointer page、segment head page、level1 page。 ● 未知页面： – unknown(data extent)：全零段页式页面，无法判断页面类型。 – unknown(fsm indexurq)：fsm或indexurq页面。
result	TEXT	解析结果。

例如（下列操作需要在DN上执行，relid必须为合法的段页式对象在该DN上的oid，且该对象需要存在数据，才能查到页面信息）：

```
gaussdb=# SELECT * FROM gs_seg_dump_page(16788, 1024, 0);
 page_type |                                                result
-----------
```


版权所有 © 华为云计算技术有限公司

```
+-------------------------------------------------------------------------------------------------------------------------
----
 heap      | Page information of block
6021/6021                                                                                                              +
         |        pd_lsn: 0/4463418 ,len 8 ,offset 0                                                                     +
         |                                                                                                               +
         |        pd_checksum: 0xD4CD, verify success,len 2, offset
8                                                                                                                      +
         |        pd_flags:                                                                                              +
         |        pd_lower: 44, non-empty, len 2, offset
12                                                                                                                     +
         |        pd_upper: 8160, old, len 2, offset 14                                                                  +
         |        pd_special: 8192, size 0, len 2, offset 16                                                             +
         |        Page size & version: 8192, 6, len 2, offset 18
+
         |        pd_xid_base: 17049, len 8, offset 24 pd_multi_base: 0, len 8, offset
32                                                                                                                     +
         |        pd_prune_xid: 17049, len 4 ,offset 20                                                                  +
         |                                                                                                               +
         |        Heap tuple information on this page
+
         |                                                                                                               +
         |               Tuple #1 is normal: length 28, offset
8160                                                                                                                   +
         |                       (uint64)xmin/xmax/t_cid:
17052/0/0                                                                                                              +
         |                       (uint32)t_xmin/t_xmax: 3/3(check ilm flag to indicate whether t_xmin/
t_xmax is xid or ilm time)+
         |                       ctid:(block 0/0, offset 1)                                                              +
         |                       t_infomask: HEAP_XMAX_INVALID
HEAP_HAS_NO_UID                                                                                                        +
         |                       t_infomask2: Attrs Num: 1                                                               +
         |                       t_hoff: 24                                                                              +
         |                       t_bits: NNNNNNNN                                                                        +
         |        Summary (1 total):  1 normal, 0 unused, 0
dead                                                                                                                   +
         |                                                                                                               +
         | Normal Heap Page, special space is 0
+
         |                                                                                                               +
         |
(1 row)
```

- gs_seg_get_spc_location(tablespace_name NAME, bucketnode INTEGER, head_block_id BIGINT, block_id BIGINT)

  描述：给定段和逻辑页面号，计算物理位置。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
tablespace_name	NAME	段对象所属的表空间。
bucketnode	INTEGER	• 0~1023表示hashbucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
head_block_id	BIGINT	指定的段头页面号。
block_id	BIGINT	指定的逻辑页面号。

版权所有 © 华为云计算技术有限公司

返回值说明：

名称	类型	描述
extent_id	INTEGER	逻辑页面所在的逻辑扩展号。
extent_size	INTEGER	逻辑页面所在的逻辑扩展的大小。
file_id	INTEGER	物理页面所在的数据文件标识。取值范围：[1, 5]的int4值。
file_block_id	BIGINT	物理页面在数据文件中的偏移页面号。

例如（下列操作需要在DN上执行，需要提前在表空间下创建过段页式普通表并插入过数据）：

```
gaussdb=# SELECT * FROM gs_seg_get_spc_location('pg_default', 1024, 4157, 0);
 extent_id | extent_size | file_id | file_block_id
-----------+-------------+---------+---------------
         0 |           8 |       2 |          4157
(1 row)
```

- gs_seg_get_spc_location(relid OID, bucketnode INTEGER, block_id BIGINT, partition BOOLEAN DEFAULT FALSE，forknum INTEGER DEFAULT 0)

  描述：给定段和逻辑页面号，计算物理位置。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
relid	OID	段对象标识。
bucketnode	INTEGER	• 0~1023表示hashbucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
block_id	BIGINT	指定逻辑页面号。
partition	BOOLEAN DEFAULT FALSE	段对象是否为分区，默认为FALSE。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
forknum	INTEGER DEFAULT 0	段对象的分支，默认为0。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。

返回值说明：

名称	类型	描述
extent_id	INTEGER	逻辑页面所在的逻辑扩展号。
extent_size	INTEGER	逻辑页面所在的逻辑扩展的大小。
file_id	INTEGER	物理页面所在的数据文件标识。取值范围：[1, 5]的int4值。
file_block_id	BIGINT	物理页面在数据文件中的偏移页面号。

例如（下列操作需要在DN上执行，relid必须为合法的段页式对象在该DN上的oid，且该对象需要存在数据）：

```
gaussdb=# SELECT * FROM gs_seg_get_spc_location(24578,1024, 0);
 extent_id | extent_size | file_id | file_block_id
-----------+-------------+---------+---------------
         0 |           8 |       2 |          4157
(1 row)
```

- gs_seg_get_location(block_id BIGINT, as_extent BOOLEAN DEFAULT FALSE)

  描述：给定段和逻辑页面号，计算物理位置。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
block_id	BIGINT	指定逻辑页面号。
as_extent	BOOLEAN DEFAULT FALSE	参数block_id是否为扩展号，默认FALSE。

返回值说明：

名称	类型	描述
extent_id	BIGINT	扩展号。
extent_size	INTEGER	扩展的大小。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
extent_offset	INTEGER	扩展的偏移块号。
level0_slots_idx	INTEGER	扩展在段头level0槽位数组下标。
level1_slots_idx	INTEGER	扩展在段头level1槽位数组下标。
level1_page_offset	INTEGER	扩展在段头level1槽位页面的偏移。
segment_blocks	BIGINT	包含此页面或扩展的段的页面数。
relative_fno	INTEGER	包含此页面或扩展的相对文件编号。

例如：

```
gaussdb=# SELECT * FROM gs_seg_get_location(4157);
 extent_id | extent_size | extent_offset | level0_slots_idx | level1_slots_idx | level1_page_offset |
segment_blocks | relative_fno
-----------+-------------+---------------+------------------+------------------+--------------------
+----------------+--------------
       47 |         128 |            61 |               47 |                  |                  |         4158 |            3
(1 row)
```

- gs_seg_get_segment_layout()

  描述：输出段的静态布局。只支持管理员权限用户查询。

  返回值说明：

名称	类型	描述
version	TEXT	段页式版本。默认：1.0。
section_id	INTEGER	段划分的数据区号。
section_type	TEXT	段数据区扩展类型。 取值范围： ● meta：段头。 ● data：数据。
extent_size	INTEGER	扩展大小。单位为字节。
extent_page_count	INTEGER	扩展页面数。
extent_count_start	BIGINT	起始扩展号。
extent_count_end	BIGINT	终止扩展号。
total_size	BIGINT	段数据区的大小。单位为字节。

版权所有 © 华为云计算技术有限公司

例如：

```
gaussdb=# SELECT * FROM gs_seg_get_segment_layout();
 version | section_id | section_type | extent_size | extent_page_count | extent_count_start |
extent_count_end |   total_size
---------+------------+--------------+-------------+-------------------+--------------------+------------------
+----------------
 1.0     |          1 | meta         |        8192 |                 1 |                  0 |                0 |             8192
 1.0     |          2 | data         |       65536 |                 8 |                  1 |               16 |          1048576
 1.0     |          3 | data         |     1048576 |               128 |                 17 |              143 |        134217728
 1.0     |          4 | data         |     8388608 |              1024 |                144 |              255 |
1073741824
 1.0     |          5 | data         |    33554432 |              4096 |                256 |          1025255 |
34394366541824
(5 rows)
```

- gs_seg_get_datafile_layout()

  描述：查看1~5号数据文件的静态布局。只支持管理员权限用户查询。

  返回值说明：

名称	类型	描述
version	TEXT	段页式版本。默认：1.0。
seg_storage_type	TEXT	● segment表示普通段页式数据。 ● hashbucket表示hashbucket数据。
file_id	INTEGER	数据文件标识。取值范围：[1, 5]的int4值。
section_id	INTEGER	数据文件划分的数据区号。
section_type	TEXT	数据文件区类型。 取值范围： ● file_header表示文件头。 ● spc_header表示空间头。 ● map_header表示映射头。 ● map_pages表示映射页面。 ● ip_pages(inverse pointer pages)表示反向指针页面。 ● data_pages表示数据页面。
page_start	BIGINT	数据区域起始页面号。
page_end	BIGINT	数据区域结束页面号。
page_count	BIGINT	数据区域页面数。
total_size	BIGINT	数据区的大小。单位为字节。

例如：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM gs_seg_get_datafile_layout();
 version | seg_storage_type | file_id | section_id | section_type |  page_start  |   page_end   |
page_count  |   total_size
---------+------------------+---------+------------+--------------+--------------+--------------+-------------
+-----------------
 1.0     | segment          |       1 |          0 | file_header  |            0 |            0 |           1 |           8192
 1.0     | segment          |       1 |          1 | spc_header   |            1 |            1 |           1 |           8192
 1.0     | segment          |       1 |          2 | map_header   |            2 |            2 |           1 |           8192
 1.0     | segment          |       1 |          3 | map_pages    |            3 |           66 |          64 |         524288
 1.0     | segment          |       1 |          4 | ip_pages     |           67 |         4156 |        4090 |
33505280
 1.0     | segment          |       1 |          5 | data_pages   |         4157 |      4147260 |     4143104 |
33940307968

...(数据较多，仅展示部分)
```

- gs_seg_get_slice_layout(file_id INTEGER, bucketnode INTEGER,slice_id INTEGER)

  描述：输出给定数据文件分片的静态布局。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
file_id	INTEGER	数据文件标识。取值范围：[1, 5]的int4值。
bucketnode	INTEGER	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
slice_id	INTEGER	Slice文件标识。

  返回值说明：

名称	类型	描述
version	TEXT	段页式版本。默认：1.0。
section_id	INTEGER	数据文件划分的数据区号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
section_type	INTEGER	数据文件区类型。 取值范围： ● file_header表示文件头。 ● spc_header表示空间头。 ● map_header表示映射头。 ● map_pages表示映射页面。 ● ip_pages(inverse pointer pages)表示反向指针页面。 ● data_pages表示数据页面。
page_start	BIGINT	数据区域起始页面号。
page_end	BIGINT	数据区域结束页面号。
page_count	BIGINT	数据区域页面数。
total_size	BIGINT	数据区的大小。单位为字节。

例如：

```
gaussdb=# SELECT * FROM gs_seg_get_slice_layout(1,1024,0);
 version | section_id | section_type | page_start | page_end | page_count | total_size
---------+------------+--------------+------------+----------+------------+------------
 1.0     |          0 | file_header  |          0 |        0 |          1 |       8192
 1.0     |          1 | spc_header   |          1 |        1 |          1 |       8192
 1.0     |          2 | map_header   |          2 |        2 |          1 |       8192
 1.0     |          3 | map_pages    |          3 |       66 |         64 |     524288
 1.0     |          4 | ip_pages     |         67 |     4156 |       4090 |   33505280
 1.0     |          5 | data_pages   |       4157 |   131071 |     126915 | 1039687680
(6 rows)
```

- gs_seg_get_segment(tablespace_name NAME, bucketnode INTEGER, head_block_id BIGINT)

  描述：输出该表空间下段头文件里段头页面的段头信息。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
tablespace_name	NAME	段对象所属的表空间。
bucketnode	INTEGER	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
head_block_id	BIGINT	段头页面号。

版权所有 © 华为云计算技术有限公司

返回值说明：

名称	类型	描述
blocks	BIGINT	段对象的逻辑页面数。
total_blocks	BIGINT	段对象的物理页面数。
extents	INTEGER	段对象的逻辑扩展数。
total_extents	INTEGER	段对象的物理扩展数。
head_lsn	TEXT	段头lsn标识。
level0_slots	BIGINT[]	段扩展映射level0槽位数组。
level1_slots	BIGINT[]	段扩展映射level1槽位数组。
fork_head	BIGINT[]	段对象的分支段头数组。

例如（下列操作需要在创建过段页式普通表后再执行）：

```
gaussdb=# SELECT * FROM gs_seg_get_segment('pg_default', 1024, 4157);
 blocks | total_blocks | extents | total_extents | head_lsn | level0_slots | level1_slots |   fork_head
--------+--------------+---------+---------------+----------+--------------+--------------+---------------
      9 |           16 |       2 |             2 | 62211744 | {4157,4165}  | {}           | {4157,4158,-1}
(1 row)
```

- gs_seg_get_segment(relid OID, bucketnode INTEGER, partition BOOLEAN DEFAULT FALSE, forknum INTEGER DEFAULT 0)

  描述：给定relid，bucketnode等信息输出对应的段头信息。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
relid	OID	表对应的OID。
bucketnode	INTEGER	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
partition	BOOLEAN DEFAULT FALSE	段对象是否为分区，默认为false。 取值范围：bool，代表所传入的oid对应的对象是否为分区。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
forknum	INTEGER DEFAULT 0	数据文件fork号，默认为0。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。

返回值说明：

名称	类型	描述
blocks	BIGINT	段对象的逻辑页面数。
total_blocks	BIGINT	段对象的物理页面数。
extents	INTEGER	段对象的逻辑扩展数。
total_extents	INTEGER	段对象的物理扩展数。
head_lsn	TEXT	段头lsn标识。
level0_slots	BIGINT[]	段扩展映射level0槽位数组。
level1_slots	BIGINT[]	段扩展映射level1槽位数组。
fork_head	BIGINT[]	段对象的分支段头数组。

例如（下列操作需要在DN上执行，relid必须为查询的段页式对象在该DN上的oid）：

```
gaussdb=# SELECT * FROM gs_seg_get_segment(16768, 1024);
 blocks | total_blocks | extents | total_extents | head_lsn | level0_slots | level1_slots |   fork_head
--------+--------------+---------+---------------+----------+--------------+--------------+---------------
      9 |           16 |       2 |             2 | 62211744 | {4157,4165}  | {}           | {4157,4158,4294967295}
(1 row)
```

- gs_seg_get_extents(tablespace_name NAME, bucketnode INTEGER, head_block_id BIGINT)

  描述：输出该表空间下段头文件里段头页面的段对象的所有扩展，包含1号文件中的segment head、fork head、level1 page，2~5号文件中的data extent。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
tablespace_name	NAME	段对象所属的表空间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bucketnode	INTEGER	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
head_block_id	BIGINT	段头页面号。

返回值说明：

名称	类型	描述
extent_id	INTEGER	逻辑扩展号。
file_id	INTEGER	扩展所在的数据文件标识。取值范围：[1, 5]的int4值。
forknum	INTEGER	段对象的分支。 取值范围： • 0：main fork。 • 1：fsm fork。 • 2：vm fork。
block_id	BIGINT	扩展所在的数据文件中的起始页面号。
blocks	INTEGER	扩展大小。 取值范围： • 1表示1号文件。 • 8表示2号文件。 • 128表示3号文件。 • 1024表示4号文件。 • 4096表示5号文件。
usage_type	TEXT	Extent的使用类型。 取值范围： • segment head表示段头。 • fork head表示分支头。 • level1 page表示level页面。 • data extent表示数据扩展。

版权所有 © 华为云计算技术有限公司

例如（下列操作需要在创建过段页式普通表后再执行）：

```
gaussdb=# SELECT * FROM gs_seg_get_extents('pg_default', 1024, 4157);
 extent_id | file_id | forknum | block_id | blocks |  usage_type
-----------+---------+---------+----------+--------+--------------
           |       1 |       0 |     4157 |      1 | segment head
         0 |       2 |       0 |     4157 |      8 | data extent
         1 |       2 |       0 |     4165 |      8 | data extent
(3 rows)
```

- gs_seg_get_extents(relid OID, bucketnode INTEGER, partition BOOLEAN DEFAULT FALSE, forknum INTEGER DEFAULT 0)

  描述：输出对应段头文件里段头页面的段对象的所有扩展，包含1号文件中的segment head、fork head、level1 page，2~5号文件中的data extent。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
relid	OID	表对应的OID。
bucketnode	INTEGER	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
partition	BOOLEAN DEFAULT FALSE	段对象是否为分区，默认为false。 取值范围：boolean，代表所传入的oid对应的对象是否为分区。
forknum	INTEGER DEFAULT 0	数据文件fork号，默认为0。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。

  返回值说明：

名称	类型	描述
extent_id	INTEGER	逻辑扩展号。
file_id	INTEGER	扩展所在的数据文件标识。取值范围：[1, 5]的int4值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
forknum	INTEGER	段对象的分支。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。
block_id	BIGINT	扩展所在的数据文件中的起始页面号。
blocks	INTEGER	扩展大小。 取值范围： ● 1表示1号文件。 ● 8表示2号文件。 ● 128表示3号文件。 ● 1024表示4号文件。 ● 4096表示5号文件。
usage_type	TEXT	Extent的使用类型。 取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。

例如（下列操作中relid必须是合法的段页式对象的oid）：

```
gaussdb=# SELECT * FROM gs_seg_get_extents(16768, 1024);
 extent_id | file_id | forknum | block_id | blocks |  usage_type
-----------+---------+---------+----------+--------+--------------
           |       1 |       0 |     4157 |      1 | segment head
         0 |       2 |       0 |     4157 |      8 | data extent
         1 |       2 |       0 |     4165 |      8 | data extent
(3 rows)
```

- gs_seg_free_spc_remain_segment(tablespace_name NAME, head_file_id INTEGER, bucketnode INTEGER, head_block_id BIGINT)

  描述：释放指定表空间上的段页式残留段在1号文件上占用的页面。残留段可以在**GS_SEG_SPC_REMAIN_SEGMENTS**视图中查询。只支持管理员权限用户查询。且仅允许在主机执行。

版权所有 © 华为云计算技术有限公司

须知

- 该函数是一种兜底的离线段页式清理手段，当前版本已提供在线清理函数：gs_seg_auto_free_remain_segment()。
- 为保证查询到的残留段和残留扩展的一致性，以及残留清理的一致性，该函数和**GS_SEG_SPC_REMAIN_SEGMENTS**视图应在DDL\DML受限状态下执行。鉴于当前版本并未提供DML\DDL受限能力，用户在使用该特性时，应保证其执行环境处于DML\DDL受限状态。
- 同一节点的同一数据库上，不允许同时运行多个残留清理相关操作，包括手动执行该函数、手动执行在线清理操作以及由autovacuum调度执行的在线清理操作。

参数说明：

名称	类型	描述
tablespace_name	NAME	表空间名称。
head_file_id	INTEGER	段页式残留段的段头所在的数据文件标识。取值范围：[1]。
bucketnode	INTEGER	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
head_block_id	BIGINT	指定的段头页面号。

返回值说明：void

例如：

```
gaussdb=# SELECT * FROM gs_seg_free_spc_remain_segment('pg_default', 1, 1024, 4159);
 gs_seg_free_spc_remain_segment
--------------------------------

(1 row)
```

- gs_seg_auto_free_remain_segment()

  描述：释放本节点上指定表空间上的段页式残留段在1号文件上占用的页面。只支持管理员权限用户查询，且仅允许在主机执行。

版权所有 © 华为云计算技术有限公司

须知

- 该函数为在线的段页式清理手段。
- 该函数需要在参数enable_auto_segment_remain_cleanup取值为on时使用。开启方式参考**开发人员选项**。
- 同一节点的同一数据库上，不允许同时运行多个残留清理相关操作，包括手动执行该函数、手动执行离线清理操作以及由autovacuum调度执行的在线清理操作。

参数说明：void

返回值说明：void

例如：

```
gaussdb=# SELECT * FROM gs_seg_auto_free_remain_segment();
 gs_seg_auto_free_remain_segment
---------------------------------

(1 row)
```

- gs_seg_free_spc_remain_extent(tablespace_name NAME, file_id INTEGER, bucketnode INTEGER, forknum INTEGER, block_id BIGINT)

  描述：释放指定表空间上残留的段页式孤立扩展。残留的孤立扩展可以在**GS_SEG_SPC_REMAIN_EXTENTS**视图中查询。只支持管理员权限用户查询。且仅允许在主机执行。

须知

- 该函数为兜底的离线段页式清理手段，当前版本已提供在线清理函数：gs_seg_auto_free_remain_segment()。
- 同一节点的同一database上，不允许同时运行多个残留清理相关操作，包括手动执行该函数、手动执行在线清理操作以及由autovacuum调度执行的在线清理操作。

参数说明：

名称	类型	描述
tablespace_name	NAME	表空间名称。
file_id	INTEGER	段页式残留扩展所在数据文件标识。取值范围：[1~5]号文件。
bucketnode	INTEGER	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
forknum	INTEGER	数据文件分支编号。 取值范围： ● 0：main fork。 ● 1：fsm fork。 ● 2：vm fork。
block_id	BIGINT	指定的段头页面号。

返回值说明：void

例如：

```
gaussdb=# SELECT * FROM gs_seg_free_spc_remain_extent('pg_default', 1, 1024, 0, 4159);
 gs_seg_free_spc_remain_extent
-------------------------------

(1 row)
```

- gs_seg_get_datafiles(database_name NAME)

  描述：输出查看实例所有数据文件信息。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
database_name	NAME	数据库名称。默认为current_database()，代表当前数据库名称。

返回值说明：

名称	类型	描述
file_name	TEXT	数据文件名。例如：base/17467/2_fsm。
file_id	INTEGER	数据文件标识。取值范围：[1~5]号文件。
bucketnode	INTEGER	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
forknum	INTEGER	数据文件分支类型。
tablespace_name	NAME	数据文件所属的表空间名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
contents	TEXT	数据文件的存储内容。 取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
extent_size	INTEGER	数据文件的扩展大小。
meta_blocks	BIGINT	数据文件的已分配的元页面数。
data_blocks	BIGINT	数据文件的已分配的数据页面数。
total_blocks	BIGINT	数据文件的总物理页面数。
high_water_mark	BIGINT	数据文件使用页数的高水位线。
utilization	REAL	使用的block数占总block数的百分比。即(data_blocks+meta_blocks)/total_blocks。

例如（下列操作需要在创建过段页式表后再执行）：

```
gaussdb=# SELECT * FROM gs_seg_get_datafiles();
  file_name   | file_id | bucketnode | forknum | tablespace_name | contents  | extent_size | meta_blocks
| data_blocks | total_blocks | high_water_mark | utilization
--------------+---------+------------+---------+-----------------+-----------+-------------+-------------
+-------------+--------------+-----------------+-------------
 base/15949/1 |       1 |       1024 |       0 | pg_default      | permanent |           1 |        4157 |           1
|        16384 |      1048702976 | 2.24208e-44
 base/15949/2 |       2 |       1024 |       0 | pg_default      | permanent |           8 |        4157 |           8
|        16384 |      1048717312 | 2.24208e-44
(2 rows)
```

- gs_seg_get_spc_extents(tablespace_name NAME, file_id INTEGER, bucketnode INTEGER, forknum INTEGER, skip_unused BOOLEAN DEFAULT TRUE)

  描述：获取指定表空间所有扩展信息。只支持管理员权限用户查询。

  参数说明：

名称	类型	描述
tablespace_name	NAME	表空间名称。
file_id	INTEGER	数据文件标识。取值范围：[1, 5]的int4值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bucketnode	INTEGER	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
forknum	INTEGER	数据文件分支。 取值范围： • 0：main fork。 • 1：fsm fork。 • 2：vm fork。
skip_unused	BOOLEAN DEFAULT TRUE	是否只输出已分配的extent，默认为TRUE，代表只输出已分配的extent。

返回值说明：

名称	类型	描述
block_id	BIGINT	数据扩展的起始页面号。
blocks	INTEGER	数据扩展大小。 取值范围： • 1表示1号文件。 • 8表示2号文件。 • 128表示3号文件。 • 1024表示4号文件。 • 4096表示5号文件。
contents	TEXT	存储内容。 取值范围： • permanent表示永久。 • unlogged表示不记录日志。 • temporary表示全局临时。 • temporary2表示本地临时。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
in_used	TEXT	是否已分配。 取值范围： ● Y表示已分配。 ● N表示未分配。
mapblock_location	TEXT	扩展在map block中的位置。格式：(page_id, offset)。
head_file_id	INTEGER	段头所在数据文件标识。
head_block_id	BIGINT	段头页面号。
usage_type	TEXT	Extent的使用类型。 取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。
remain_flag	TEXT	是否为shrink残留扩展。 取值范围： ● Y表示是shrink残留扩展。 ● N表示不是shrink残留扩展。
special_data	INTEGER	扩展对应反向指针的特殊数据区。
ipblock_location	TEXT	扩展反向指针位置。格式：(block_id, offset)。

例如：

```
gaussdb=# SELECT * FROM gs_seg_get_spc_extents('pg_default', 1,1024, 0);
 block_id | blocks | contents  | in_used | mapblock_location | head_file_id | head_block_id |  usage_type
| remain_flag | special_data | ipblock_location
----------+--------+-----------+---------+-------------------+--------------+---------------+--------------
+-------------+--------------+------------------
    4157 |      1 | permanent | Y       | (4157,0)          |            1 |          4157 | segment head | N
|           0 | (67,0)
    4158 |      1 | permanent | Y       | (4157,1)          |            1 |          4158 | fork head    | N
|           1 | (67,1)
(2 rows)
```

1.6.27.12 bucket 系统函数

- gs_bucket_get_data_distribution(relation REGCLASS, forknum INTEGER, including_toast Boolean)

  描述：获取某个hashbucket表的数据分布情况。

版权所有 © 华为云计算技术有限公司

参数说明：

名称	类型	描述
relation	REGCLASS	表名。
forknum	INTEGER	数据文件分支，合法取值范围包括： ● 0：表示main fork。 ● 1：表示fsm fork。 ● 2：表示vm fork。 ● -1：表示其他所有forknum下的结果之和。
including_toast	Boolean	数据分布情况统计是否包含toast表。

返回值说明：

名称	类型	描述
node_name	TEXT	数据库节点名称。
bucketnode	INT2	bucket编号。
used_blocks	INT8	表在该bucket编号下已使用的数据块数量。
allocated_blocks	INT8	表在该bucket编号下已申请的数据块数量。

示例：

```
gaussdb=# select * from gs_bucket_get_data_distribution('bucket_table', 0, true);

 node_name | bucketnode | used_blocks | allocated_blocks
-----------+------------+-------------+------------------
 datanode1 |          0 | 1           | 8
 datanode1 |          1 | 1           | 8
 datanode1 |          2 | 1           | 8
 datanode1 |          3 | 1           | 8
 datanode1 |          4 | 1           | 8
 datanode1 |          5 | 1           | 8
 datanode1 |          6 | 1           | 8
 datanode1 |          7 | 1           | 8
 datanode1 |          8 | 1           | 8
 datanode1 |          9 | 1           | 8
 datanode1 |         10 | 1           | 8
 datanode1 |         11 | 1           | 8
 datanode1 |         12 | 1           | 8
 datanode1 |         13 | 1           | 8
 datanode1 |         14 | 1           | 8
 datanode1 |         15 | 1           | 8
 datanode1 |         16 | 1           | 8
 datanode1 |         17 | 1           | 8
 datanode1 |         18 | 1           | 8
 datanode1 |         19 | 1           | 8
 datanode1 |         20 | 1           | 8
 datanode1 |         21 | 1           | 8
```


版权所有 © 华为云计算技术有限公司

```
datanode1 |        22 | 1          | 8
datanode1 |        23 | 1          | 8
datanode1 |        24 | 1          | 8
...(数据较多，仅展示部分)
```

- gs_redis_get_plan(origin_group_id OID, target_group_id OID)

  描述：获取完整的迁移计划。只支持在扩容期间内部调用，且只支持在CN上调用。

  参数说明：

名称	类型	描述
origin_group_id	OID	源节点的node group。
target_group_id	OID	新节点的node group。

  返回值说明：

名称	类型	描述
sender_id	OID	源节点的node oid。
sender_name	CSTRING	源节点的节点名称。
receiver_id	OID	新节点的node oid。
receiver_name	CSTRING	新节点的节点名称。
bucket_number	INT4	bucket_list中包含的bucket个数。
bucket_list	OIDVECTOR_EXTEND	迁移计划涉及的bucket列表。

- gs_redis_get_bucket_statistics()

  描述：获取日志流传输状态。只支持在扩容期间内部调用。

  参数：void

  返回值说明：

名称	类型	描述
bucket_id	OID	bucket id
redis_state	INT1	bucket的扩容状态，0表示扩容未开始，1表示扩容基线数据已完成。
xlog_count	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）在原DN产生的Xlog数量。
sndr_latest_lsn	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）在原DN产生的最新LSN。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
parser_latest_lsn	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）被原DN的扩容相关线程解析到的最新LSN。
parser_latest_lsn_new	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）被原DN的扩容相关线程解析到bucketxlog的最新LSN。
rcvr_redo_latest_lsn	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）被新DN的扩容相关线程回放到的最新LSN。
rcvr_redo_latest_lsn_new	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）被新DN的扩容相关线程回放到bucketxlog的最新LSN。
rcvr_checkpoint	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）的checkpoint点。
rcvr_redo_start_lsn	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）回放开始的原始LSN。
rcvr_redo_end_lsn	INT8	bucket在当前database扩容开始后（即上次调用gs_redis_set_distributed_db后）回放结束的原始LSN。

- gs_redis_set_distributed_db(db_name CSTRING, bucketgroup SMALLINT)

  描述：设置当前正在hashbucket重分布的数据库。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：

  - db_name：本次要重分布的数据库库名。
  - bucketgroup：本次要重分布的bucketgroup，当前版本暂无任何作用。

  返回值说明：void

- gs_redis_parse_slicebucket_tranfer_file(plan_json CSTRING, type CSTRING)

  描述：用于获取slicebucket搬迁计划。只支持在扩容期间内部调用，且只支持在CN上调用，当前版本暂不支持。

  参数说明：

名称	类型	描述
plan_json	TEXT	用户指定的slice搬迁文件。
type	CSTRING	解析计划类型，可指定为'incremental'或'full'，默认为'full'。

  返回值类型：

版权所有 © 华为云计算技术有限公司

名称	类型	描述
idx	INT	需要搬迁的数据库序号。
database_name	CSTRING	需要搬迁的数据库名字。

- gs_redis_get_slicebucket_plan()

  描述：用于获取当前库的slicebucket迁移计划。只支持在扩容期间内部调用，且只支持在CN上调用，当前版本暂不支持。

  参数：void

  返回值说明：

名称	类型	描述
sender_id	OID	源节点的节点oid。
sender_name	CSTRING	源节点的节点名称。
receiver_id	OID	新节点的节点oid。
receiver_name	CSTRING	新节点的节点名称。
bucket_number	INT4	bucket_list中包含的bucket个数。
bucket_list	OIDVECTOR_EXTEND	迁移计划涉及的bucket列表。
bucketgroup	OID	bucketlist中的bucket所属的具体group，取值范围为0 ~ 4。

- gs_redis_hashbucket_update_segment_header(origin_group_id OID, target_group_id OID)

  描述：更新当前数据库所有hashbucket表的header。只支持在扩容期间内部调用，且只支持在CN上调用。

  参数说明：

名称	类型	描述
origin_group_id	OID	源节点的node group。
target_group_id	OID	新节点的node group。

  返回值类型：Boolean

- gs_redis_local_get_segment_header(table_name REGCLASS, bucketlist OIDVECTOR_EXTEND)

  描述：获取段页式表header。只支持在DN上调用。

  参数说明：

版权所有 © 华为云计算技术有限公司

名称	类型	描述
table_name	REGCLASS	表名
bucketlist	OIDVECTOR_EXTEND	bucket列表。

返回值类型：header_info

示例：

```
gaussdb=#  SELECT * FROM gs_redis_local_get_segment_header('mytable', '256');
 gs_redis_local_get_segment_header
-----------------------------------------------------------------------------------------------------------------------------------
4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,4294967295,...(数据较多，仅展示部分)
(1 row)
```

- gs_redis_local_update_segment_header(table_name REGCLASS, header_info CSTRING)

  描述：更新段页式表header。只支持在DN上调用。

  参数说明：

名称	类型	描述
table_name	REGCLASS	表名
header_info	CSTRING	段页式表header。

  返回值类型：void

  示例：

```
gaussdb=#  SELECT * FROM gs_redis_local_update_segment_header('mytable',
'4294967295,4294967295,4294967295,4294967295,....'); (数据较多，仅展示部分)
 gs_redis_local_update_segment_header
------------------------------------------
 0
(1 row)
```

- gs_redis_hashbucket_update_inverse_pointer(buckets TEXT, origin_dn_name TEXT, new_dn_name TEXT)

  描述：更新当前db所有hashbucket表本批次bucket的反向指针。只支持在扩容期间内部调用，且只支持在CN上调用。

  参数说明：

名称	类型	描述
buckets	TEXT	本次上线的bucket列表。
origin_dn_name	TEXT	源节点名称。
new_dn_name	TEXT	新节点名称。

版权所有 © 华为云计算技术有限公司

返回值类型：void

- gs_redis_local_get_inverse_pointer(buckets TEXT, origin_dn_name TEXT, new_dn_name TEXT)

  描述：获取反向指针。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：

名称	类型	描述
buckets	TEXT	本次上线的bucket列表。
origin_dn_name	TEXT	源节点名称。
new_dn_name	TEXT	新节点名称。

  返回值类型：void

- gs_redis_local_update_inverse_pointer(table_name TEXT, header_info TEXT, bucketlist TEXT)

  描述：记录更新反向指针的Xlog。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：

名称	类型	描述
table_name	REGCLASS	表名。
header_info	TEXT	段页式表header。
bucketlist	TEXT	bucket列表。

  返回值类型：void

- gs_redis_local_set_hashbucket_frozenxid

  描述：修改hashbucket表在系统表中的relfrozenxid64值。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：void

  返回值类型：void

- gs_redis_set_hashbucket_frozenxid(origin_group_id OID, target_group_id OID)

  描述：修改新DN节点hashbucket表在系统表中的relfrozenxid64值。只支持在扩容期间内部调用，且只支持在CN上调用。

  参数说明：

名称	类型	描述
origin_group_id	OID	源节点的node group。
target_group_id	OID	新节点的node group。

版权所有 © 华为云计算技术有限公司

返回值类型：void

- gs_redis_set_nextxid(xid BIGINT)

  描述：修改DN节点next_xid值。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：xid：预期next_xid的值

  返回值类型：void

- gs_redis_set_csn(csn BIGINT)

  描述：修改DN节点next_csn值。只支持在扩容期间内部调用，且只支持在GTM_FREE模式下和DN上调用。

  参数说明：

  – csn：预期next_csn的值。

  返回值类型：void

- gs_redis_check_bucket_flush(dn_array NAME[])

  描述：查询RTO回放私有buffer是否全部刷下去。只支持在扩容期间内部调用。

  参数说明：

  – dn_array：DN节点名称列表。

  返回值类型：Boolean

- gs_redis_get_flush_page_lsn(isclean bool)

  描述：查询bucket扩容回放的刷页信息。

  权限：只支持debug版本调用，只支持在扩容期间内部调用。

  参数说明：

  – isclean：BOOLEAN类型，是否清理刷页信息。

  返回值说明：

名称	类型	描述
node_name	TEXT	页面所在dn名。
space_id	OID	表空间id。
db_id	OID	数据库id。
file_id	OID	数据文件编号，取值范围1-5。
bucket_id	OID	bucket分区id，取值范围0-1023。
forknum	OID	数据文件分支，取值范围0-2。
blocknum	OID	物理页面在数据文件中的偏移页面号。
latest_flush_lsn	UINT8	刷脏页面最新的lsn。

- gs_redis_show_bucketxid(bucketid_list OIDVECTOR_EXTEND)

  描述：查询指定bucket对应的bucketxid。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：

版权所有 © 华为云计算技术有限公司

– bucketid_list：bucket列表。

返回值类型：CSTRING

- gs_redis_drop_bucket_files(plan_json CSTRING)

  描述：删除源节点已经物理搬迁的rangebucket文件。只支持在扩容期间内部调用，且只支持在CN上调用，当前版本暂不支持。

  参数说明：

名称	类型	描述
plan_json	CSTRING	用户指定的SLICE搬迁文件。

  返回值类型：BOOLEAN

- gs_redis_drop_bucket_files(origin_group_id OID, target_group_id OID)

  描述：删除源节点已经物理搬迁的文件。只支持在扩容期间内部调用，且只支持在CN上调用。

  参数说明：

名称	类型	描述
origin_group_id	OID	源节点的node group。
target_group_id	OID	新节点的node group。

  返回值类型：BOOLEAN

- gs_redis_local_drop_bucket_files(bucketlist CSTRING, bucketnum SMALLINT)

  描述：删除对应的bucket list。只支持在扩容期间内部调用，且只支持在DN上调用。

  参数说明：

名称	类型	描述
bucketlist	CSTRING	指定bucket列表。
bucketnum	SMALLINT	bucket的个数。

  返回值类型：Boolean

- gs_redis_local_drop_bucket_files(database_name NAME, bucketlist CSTRING, bucketnum SMALLINT)

  描述：用于rangebucket表删除对应db的bucket list。只支持在扩容期间内部调用，且只支持在DN上调用，当前版本暂不支持。

  参数说明：

名称	类型	描述
database_name	NAME	指定数据库名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bucketlist	CSTRING	指定bucket列表。
bucketnum	SMALLINT	bucket的个数。

返回值类型：Boolean

- gs_redis_slicebucket_update_segment_header()

  描述：更新当前数据库所有slicebucket表的header。只支持在扩容期间内部调用，且只支持在CN上调用，当前版本暂不支持。

  入参：void

  返回值类型：Boolean

- gs_redis_get_undo_state(mode INTEGER)

  描述：获取扩容重分布期间的UNDO回滚状态。仅支持在扩容期间内部调用。

  参数说明：

名称	类型	描述
mode	INTEGER	重分布期间判断UNDO回滚状态的模式，取值范围： ● 0：表示UNDO监控点next_xid前是否有待回滚的事务。 ● 1：表示是否有新增的小于监测点next_xid的回滚事务。 ● 2：表示是否有大于监测点next_xid的待回滚事务。

  返回值state为INT4类型，表示查询到的扩容重分布期间的UNDO回滚状态。根据参数值不同的返回值含义如下：

  - 当mode=0时：
    - state = 0表示无待回滚事务，xmin > 监控点next_xid。
    - state = 1表示无待回滚事务，xmin <= 监控点next_xid。
    - state = 2表示有待回滚事务。
  - 当mode=1时：
    - state=0表示无监控点next_xid之前的新增回滚事务。
    - state=1表示有监控点next_xid之前的新增回滚事务。
  - 当mode=2时：
    - state=0表示无监控点next_xid之后的待回滚事务。
    - state=1表示有监控点next_xid之后的待回滚事务。

版权所有 © 华为云计算技术有限公司

1.6.27.13 Undo 系统函数

如需确认当前回滚段使用的存储方式，可以通过查询gs_global_config系统表的undostoragetype字段确认：segpage表示段页式，预留参数，暂不支持；page表示页式。

示例：

```
gaussdb=# SELECT * FROM gsd_global_config where name like '%undostoragetype%';
      name       |  value
-----------------+---------
 undostoragetype |  page
(1 row)
```

说明

Undo系统函数DN上执行，不支持在CN上执行。

- gs_undo_meta(type, zoneId, location)

  描述：Undo模块元信息。

  参数：

  - type(元信息类型)
    - 0：表示UndoZone(Record) 对应的元信息。
    - 1：表示UndoZone(Transaction Slot) 对应的元信息。
    - 2：表示UndoSpace(Record) 对应的元信息。
    - 3：表示UndoSpace(Transaction Slot) 对应的元信息。
  - zoneId(UndoZone编号)
    - -1：表示所有UndoZone的元信息。
    - 0~1024*1024-1：表示对应ZoneID的元信息。
  - location(读取位置)
    - 0：表示从当前内存中读取。
    - 1：表示从物理文件中读取。

  返回值类型：record

  说明

  该系统函数仅支持回滚段使用页式存储方式。

  表 1-147 gs_undo_meta 参数说明

参数类型	参数名	类型	描述
输出参数	zoneId	oid	UndoZone的ID。
输出参数	persistType	oid	持久化级别。
输出参数	insert	text	下一条插入的Undo记录位置。
输出参数	discard	text	普通回收到的Undo记录位置。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	end	text	强制回收掉的Undo记录位置，小于它的Undo记录已经被回收。
输出参数	used	text	已经使用的Undo空间。
输出参数	lsn	text	修改UndoZone的LSN。
输出参数	pid	oid	UndoZone绑定的线程ID。

- gs_undo_translot(location, zoneId)
  描述：Undo事务槽信息。
  参数：
  - location(读取位置)
    - 0：表示从当前内存中读取。
    - 1：表示从物理文件中读取。
  - zoneId(UndoZone编号)
    - -1：表示所有UndoZone的元信息。
    - 0~1024*1024-1：表示对应ZoneID的元信息。

  返回值类型：record

  说明

  该系统函数仅支持回滚段使用页式存储方式。

表 1-148 gs_undo_translot 参数说明

参数类型	参数名	类型	描述
输出参数	groupId	oid	使用的UndoZone ID。
输出参数	xactId	text	事务ID。
输出参数	startUndoPtr	text	Transaction Slot对应事务起始插入Undo记录位置。
输出参数	endUndoPtr	text	Transaction Slot对应事务结束插入Undo记录位置。
输出参数	lsn	text	对应Transaction Slot指针。
输出参数	slot_states	oid	事务状态。 ● 0表示已经提交。 ● 1表示正在执行中。 ● 2表示回滚中。 ● 3表示回滚完成。

版权所有 © 华为云计算技术有限公司

- gs_stat_undo([bool init])
  描述：Undo统计信息。
  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

  表 1-149 gs_stat_undo 页式参数说明

参数类型	参数名	类型	描述
输入参数	init	bool	可选参数，是否清理统计信息并重新开始统计。
输出参数	curr_used_zone_count	int	当前使用的UndoZone数量。
输出参数	top_used_zones	text	前三个使用量最大的UndoZone信息，格式输出为： ● zoneId1：使用大小。 ● zoneId2：使用大小。 ● zoneId3：使用大小。
输出参数	curr_used_undo_size	int	当前使用的Undo总空间大小，单位为MB。
输出参数	undo_threshold	int	为GUC参数undo_space_limit_size * 80%计算的结果,单位为MB。
输出参数	global_recycle_xid	xid	当前Undo空间回收到的事务xid（小于该xid事务产生的Undo记录都已经被回收）。
输出参数	oldest_xmin	xid	最旧的活跃事务。
输出参数	total_undo_chain_len	int8	所有访问过的Undo链总长度。
输出参数	max_undo_chain_len	int8	最大访问过的Undo链长度。
输出参数	create_undo_file_count	uint32	创建的Undo文件数量统计。
输出参数	discard_undo_file_count	uint32	删除的Undo文件数量统计。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	info	text	如果入参为false，输出undo_space_limit_size、undo_limit_size_per_transaction 、undo_retention_time参数的合理化建议。 如果入参为true，即需要init（清理统计信息），仅提示‘The statistics have been initialized.’。

示例1：清除undo统计信息

```
gaussdb=# SELECT * FROM gs_stat_undo(true);
curr_used_zone_count |   top_used_zones    | curr_used_undo_size | undo_threshold | global_recycle_xid
| oldest_xmin | total_undo_chain_len | max_undo_chain_len | create_undo_file_coun
t | discard_undo_file_count |                info
----------------------+---------------------+---------------------+----------------+--------------------
+-------------+----------------------+--------------------+----------------------
--+-------------------------+-----------------------------------
                   3 | 0 : 0, 0 : 0, 0 : 0 |                   1 |         209715 |             15741 |       15741
|                  0 |                   0 |
2 |                     0 | The statistics have been initialized.
(1 row)
```

示例2：输出undo统计信息

```
gaussdb=# SELECT * FROM gs_stat_undo(false);
curr_used_zone_count |   top_used_zones    | curr_used_undo_size | undo_threshold | global_recycle_xid
| oldest_xmin | total_undo_chain_len | max_undo_chain_len | create_undo_file_coun
t | discard_undo_file_count |
            info

----------------------+---------------------+---------------------+----------------+--------------------
+-------------+----------------------+--------------------+----------------------
--+-------------------------
+-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------
-
                   3 | 0 : 0, 0 : 0, 0 : 0 |                   1 |         209715 |             16253 |       16253
|                  0 |                   0 |
2 |                     0 | Based on the statistic info from last initialization, undo_space_limit_size is
recommended to set >= 120953 blocks, undo_limit_size_per_transaction is
recommended to set <= 327150 blocks, undo_retention_time is recommended to set <= 259200
seconds. Since last initialization, max undo space used by a single transaction is 32715 blocks.
(1 row)
```

- gs_undo_record(undoptr)

  描述：Undo记录解析。

  参数：undoptr（undo记录指针）。

  返回值类型：record

  说明

  该系统函数仅支持回滚段使用页式存储方式。

- gs_undo_dump_parsepage_mv(relpath text, blkno bigint, reltype text, rmem boolean)

  描述：解析USTORE数据表磁盘页面的页头信息，每个元组的头部信息，标识位信息以及所有可以查询到undo历史版本信息。

版权所有 © 华为云计算技术有限公司

返回值类型：text
备注：必须是系统管理员或者运维管理人员才能执行此函数。

说明

该接口当前仅支持USTORE数据表。

表 1-150 gs_undo_dump_parsepage_mv 参数说明

参数类型	参数名	类型	描述
输入参数	relpath	text	USTORE表数据文件相对路径，相对路径格式为：tablespace name/database oid/relfilenode，例如base/16603/16384, 表对应数据文件的相对路径查找可以通过pg_relation_filepath('tablename')查询。
输入参数	blkno	int8	• -1 解析所有block页面。 • 0-MaxBlocNumber解析指定的block页面。
输入参数	reltype	text	表类型，目前仅支持USTORE数据表，取值为uheap。
输入参数	rmem	boolean	• false • true 目前仅支持false，从磁盘文件上解析对应的页面。
输出参数	output	text	解析结果文件的绝对路径。

- gs_undo_meta_dump_zone(zone_id int, read_memory boolean)
  描述：解析Undo模块中UndoZone的元信息。
  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

  表 1-151 gs_undo_meta_dump_zone 页式参数说明

参数类型	参数名	类型	描述
输入参数	zone_id	int	UndoZone编号： • -1：查询所有UndoZone。 • 0-1,048,575：查询对应zone_id编号的UndoZone元信息。
输入参数	read_memory	boolean	• true：从当前内存中读取。 • false：从物理文件中读取。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	zone_id	oid	UndoZone编号。
输出参数	persist_type	oid	持久化级别： ● 0：普通表。 ● 1：无日志表。 ● 2：临时表。
输出参数	insert	text	下一条插入的Undo记录位置。
输出参数	discard	text	普通回收到的Undo记录位置。
输出参数	forcediscard	text	强制回收掉Undo记录位置，小于它的Undo记录已经被回收。
输出参数	lsn	text	修改UndoZone的LSN。

示例：

```
gaussdb=# SELECT * FROM gs_undo_meta_dump_zone(-1,true);
 zone_id | persist_type | insert | discard | forcediscard |   lsn
---------+--------------+--------+---------+--------------+----------
       0 |            0 | 244577 | 244577  | 244577       | 43967224
       1 |            0 | 108    | 66      | 66           | 43967568
  349525 |            1 | 24     | 24      | 24           | 0
  349526 |            1 | 24     | 24      | 24           | 0
  699050 |            2 | 24     | 24      | 24           | 0
  699051 |            2 | 24     | 24      | 24           | 0
(6 rows)
```

- gs_undo_meta_dump_spaces(zone_id int, read_memory boolean)
  描述：解析Undo模块中Undo记录空间、Transaction Slot空间的元信息。
  返回值类型：record

  说明

  该系统函数仅支持回滚段使用页式存储方式。

表 1-152 gs_undo_meta_dump_spaces 参数说明

参数类型	参数名	类型	描述
输入参数	zone_id	int	UndoZone编号： ● -1：查询所有UndoZone。 ● 0-1,048,575：查询对应zone_id编号的UndoZone元信息。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输入参数	read_memory	boolean	• true：从当前内存中读取。 • false：从物理文件中读取。
输出参数	zone_id	int	UndoZone编号。
输出参数	undorecord_space_tail	text	UndoRecord空间的结尾位置。
输出参数	undorecord_space_head	text	UndoRecord空间的起始位置。
输出参数	undorecord_space_lsn	text	修改UndoRecord空间LSN。
输出参数	undoslot_space_tail	text	Transaction Slot空间的结尾位置。
输出参数	undoslot_space_head	text	Transaction Slot空间的起始位置。
输出参数	undoreslot_space_lsn	text	修改Transaction Slot空间LSN。

- gs_undo_meta_dump_slot(zone_id int, read_memory boolean)
  描述：解析Undo模块中Transaction Slot元信息。
  返回值类型：record

  说明

  该系统函数仅支持回滚段使用页式存储方式。

  表 1-153 gs_undo_meta_dump_slot 参数说明

参数类型	参数名	类型	描述
输入参数	zone_id	int	Undo zone编号： • -1：查询所有UndoZone。 • 0-1,048,575：查询对应zone_id编号的UndoZone元信息。
输入参数	read_memory	boolean	• true：从当前内存中读取。 • false：从物理文件中读取。
输出参数	zone_id	int	UndoZone编号。
输出参数	allocate	text	Undo Transaction Slot分配位置。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	recycle	text	Undo Transaction Slot回收位置。
输出参数	frozen_xid	text	Frozen Xid，用于可见性判断。
输出参数	global_frozen_xid	text	全局最小的Frozen Xid，小于该Xid的事务可见。
输出参数	recycle_xid	text	回收到的Xid，小于该Xid的事务被回收。
输出参数	global_recycle_xid	text	全局最小的Recycle Xid，小于该Xid的事务被回收。

- gs_undo_translot_dump_slot(zone_id int, read_memory boolean)
  描述：解析UndoZone中的Transaction Slot。
  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

  表 1-154 gs_undo_translot_dump_slot 页式参数说明

参数类型	参数名	类型	描述
输入参数	zone_id	int	UndoZone编号： ● -1：查询所有UndoZone。 ● 0-1,048,575：查询对应zone_id编号的UndoZone元信息。
输入参数	read_memory	boolean	● true：从当前内存中读取。 ● false：从物理文件中读取。
输出参数	zone_id	oid	UndoZone编号。
输出参数	slot_xid	text	事务ID。
输出参数	start_undoptr	text	本Transaction Slot起始Undo Record在本UndoZone内的位置。
输出参数	end_undoptr	text	本Transaction Slot结束Undo Record在本UndoZone内的位置。
输出参数	slot_ptr	text	Transaction Slot对应的位置。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	slot_states	oid	事务状态： ● 0：已提交。 ● 1：执行中。 ● 2：回滚中。 ● 3：回滚完成。

示例：

```
gaussdb=# SELECT * FROM gs_undo_translot_dump_slot(-1,true);
 zone_id |     slot_xid     |  start_undoptr   |   end_undoptr    |     slot_ptr     | slot_states
---------+------------------+------------------+------------------+------------------+-------------
       1 | 0000000000015758 | 0000000000000042 | 000000000000006C | 0000000000000038 |           0
(1 row)
```

- gs_undo_translot_dump_xid(slot_xid xid, read_memory boolean)

  描述：根据xid解析UndoZone中对应的Transaction Slot。

  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

  表 1-155 gs_undo_translot_dump_xid 页式参数说明

参数类型	参数名	类型	描述
输入参数	slot_xid	xid	需要查询的事务ID。
输入参数	read_memory	boolean	● true：从当前内存中读取。 ● false：从物理文件中读取。
输出参数	zone_id	oid	UndoZone编号。
输出参数	slot_xid	text	事务ID。
输出参数	start_undoptr	text	本Transaction Slot起始Undo Record在本UndoZone内的位置。
输出参数	end_undoptr	text	本Transaction Slot结束Undo Record在本UndoZone内的位置。
输出参数	slot_ptr	text	Transaction Slot对应的位置。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	slot_states	oid	事务状态： ● 0 已提交。 ● 1 执行中。 ● 2 回滚中。 ● 3 回滚完成。

示例：

```
gaussdb=# SELECT * FROM gs_undo_translot_dump_xid('15758',false);
 zone_id |     slot_xid     |  start_undoptr   |   end_undoptr    |     slot_ptr     | slot_states
---------+------------------+------------------+------------------+------------------+-------------
       1 | 0000000000015758 | 0000000000000042 | 000000000000006C | 0000000000000038 |           0
(1 row)
```

- gs_undo_dump_record(undoptr bigint)

  描述：解析给定URP的Undo Record信息。

  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

  表 1-156 gs_undo_dump_record 页式参数说明

参数类型	参数名	类型	描述
输入参数	undoptr	xid	需要解析的Undo记录起始位置,十进制格式。
输出参数	undoptr	xid	需要解析的Undo记录在本UndoZone内的起始位置。
输出参数	xactid	xid	事务ID。
输出参数	cid	text	Command Id。
输出参数	reloid	text	Relation oid。
输出参数	relfilenode	text	文件的Relfinode。
输出参数	utype	text	Undo记录类型。
输出参数	blkprev	text	同一个块前一条Undo记录的位置。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	blockno	text	块号。
输出参数	uoffset	text	Undo记录偏移。
输出参数	prevurp	text	前一条Undo记录位置。
输出参数	payloadlen	text	Undo记录数据部分长度。
输出参数	oldxactid	text	前一个事务ID。
输出参数	partitionoid	text	分区oid。
输出参数	tablespace	text	表空间。
输出参数	alreadyread_bytes	text	读取到的Undo记录长度。
输出参数	prev_undorec_len	text	前一条Undo记录长度。
输出参数	td_id	text	Transaction Directory的ID。
输出参数	reserved	text	Undo记录中存储的旧版本元组预留标识位。
输出参数	flag	text	Undo记录中存储的旧版本元组状态标识。
输出参数	flag2	text	Undo记录中存储的旧版本元组列数。
输出参数	t_hoff	text	Undo记录数据头的长度。

示例（在Undo记录没有被回收的前提下，入参undoptr可以通过gs_undo_translot_dump_slot函数的出参end_undoptr转换为10进制后进行查询）：

```
gaussdb=# SELECT * FROM gs_undo_dump_record('0000000000000042');
 undoptr |   xactid   | cid | reloid | relfilenode | utype |  blkprev   | blockno | uoffset | prevurp |
payloadlen | oldxactid | partitionoid | tablespace | alreadyread_byt
es | prev_undorec_len | td_id | reserved | flag | flag2 | t_hoff
---------+------------+-----+--------+-------------+-------+------------+---------+---------+---------
+------------+-----------+--------------+------------+----------------
---+------------------+-------+----------+------+-------+--------
      42 | 1073807360 | 0   | 0      | 108986369   | 0     | 1024786474 | 0       | 0       | 0       | 0         |
0         | 0            | 0          | 36
```


版权所有 © 华为云计算技术有限公司

```
  | 16390          | -1    | -1      | -1   | -1    | -1
(1 row)
```

- gs_undo_dump_xid(undo_xid xid)

  描述：根据xid解析Undo记录。

  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

  表 1-157 gs_undo_dump_xid 页式参数说明

参数类型	参数名	类型	描述
输入参数	undo_xid	xid	事务Xid。
输出参数	undoptr	xid	需要解析的Undo记录在本UndoZone内的起始位置。
输出参数	xactid	xid	事务ID。
输出参数	cid	text	Command Id。
输出参数	reloid	text	Relation oid。
输出参数	relfilenode	text	文件的Relfinode。
输出参数	utype	text	Undo记录类型。
输出参数	blkprev	text	同一个块前一条Undo记录的位置。
输出参数	blockno	text	块号。
输出参数	uoffset	text	Undo记录偏移。
输出参数	prevurp	text	前一条Undo记录位置。
输出参数	payloadlen	text	Undo记录数据部分长度。
输出参数	oldxactid	text	前一个事务ID。
输出参数	partitionoid	text	分区oid。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	tablespace	text	表空间。
输出参数	alreadyread_bytes	text	读取到的Undo记录长度。
输出参数	prev_undorec_len	text	前一条Undo记录长度。
输出参数	td_id	text	Transaction Directory的ID。
输出参数	reserved	text	Undo记录中存储的旧版本元组预留标识位。
输出参数	flag	text	Undo记录中存储的旧版本元组状态标识。
输出参数	flag2	text	Undo记录中存储的旧版本元组列数。
输出参数	t_hoff	text	Undo记录数据头的长度。

示例：

```
gaussdb=# SELECT * FROM gs_undo_dump_xid('15779');
 undoptr | xactid | cid | reloid | relfilenode | utype | blkprev | blockno | uoffset | prevurp | payloadlen |
oldxactid | partitionoid | tablespace | alreadyread_bytes | pr
ev_undorec_len | td_id | reserved | flag | flag2 | t_hoff
---------+--------+-----+--------+-------------+-------+---------+---------+---------+---------+------------
+-----------+--------------+------------+-------------------+---
---------------+-------+----------+------+-------+--------
    108 |  15779 | 0   | 16767  | 16767       | 64    | 0       | 0       | 3       | 0       | 0         | 0         |
0           | 1663       | 40                | 42
             | -1    | -1       | -1   | -1    | -1
(1 row)
```

- gs_verify_undo_record(type, start_idx，end_idx, location)

  描述：校验Undo记录，目前只支持磁盘校验模式。仅支持在业务非运行时执行离线校验，校验之前需要手动执行一次checkpoint落盘操作。

  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

版权所有 © 华为云计算技术有限公司

表 1-158 gs_verify_undo_record 页式参数说明

参数类型	参数名	类型	描述
输入参数	type	text	校验类型： ● 'urp'：校验给定urp区间的所有Undo记录。 ● 'zone'：校验给定Zone区间的所有Zone的所有Undo记录。
输入参数	start_idx	int8	开始位置： ● type为'urp'时，表示起始Undo记录位置。 ● type为'zone'时，表示起始UndoZone编号。
输入参数	end_idx	int8	结束位置： ● type为'urp'时，表示Undo记录结束位置。 ● type为'zone'时，表示结束UndoZone编号。
输入参数	location	bool	● 0：内存校验。 ● 1：磁盘校验。 目前该参数只支持输入为1。
输出参数	zone_id	int8	UndoZone编号。
输出参数	detail	text	校验出错信息。

示例1，校验urp为24的这条Undo记录：

```
gaussdb=# SELECT * FROM gs_verify_undo_record('urp', 24, 24, 1);
zone_id | detail
--------+--------
(0 rows)
```

示例2，从磁盘中校验zone0到zone2的所有Undo记录：

```
gaussdb=# SELECT * FROM gs_verify_undo_record('zone', 0, 2, 1);
zone_id | detail
--------+--------
(0 rows)
```

说明

调用此函数如有报错，请联系华为工程师处理。

- gs_verify_undo_slot(type, start_idx, end_idx, location)

  描述：校验Undo事务槽，目前只支持磁盘校验模式。仅支持在业务非运行时执行离线校验，校验之前需要手动执行一次checkpoint落盘操作。

  返回值类型：record

  说明

  当回滚段使用页式文件存储方式时，使用以下格式。

版权所有 © 华为云计算技术有限公司

表 1-159 gs_verify_undo_slot 页式参数说明

参数类型	参数名	类型	描述
输入参数	type	text	校验类型： ● 'zone'：校验给定zone区间的所有zone的所有事务槽。
输入参数	start_idx	int8	起始UndoZone编号。
输入参数	end_idx	int8	结束UndoZone编号。
输入参数	location	bool	● 0：内存校验。 ● 1：磁盘校验。 目前该参数只支持输入为1。
输出参数	zone_id	int8	UndoZone编号。
输出参数	detail	text	校验出错信息。

示例，从磁盘中校验zone0到zone2的所有事务槽记录：

```
gaussdb=# SELECT * FROM gs_verify_undo_slot('zone', 0, 2, 1);
zone_id | detail
---------+---------
(0 rows)
```

说明

调用此函数如有报错，请联系华为工程师处理。

- gs_verify_undo_meta(type, start_idx, end_idx, location)

  描述：校验Undo元信息，目前只支持磁盘校验模式。仅支持在业务非运行时执行离线校验，校验之前需要手动执行一次checkpoint落盘操作。

  返回值类型：record

  说明

  该系统函数仅支持回滚段使用页式存储方式。

表 1-160 gs_verify_undo_meta 参数说明

参数类型	参数名	类型	描述
输入参数	type	text	校验类型，type只能设置为'all'，表示校验给定Zone区间的所有Zone的所有meta信息。
输入参数	start_idx	int64	起始UndoZone编号。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输入参数	end_idx	int64	结束UndoZone编号。
输入参数	location	bool	● 0：内存校验。 ● 1：磁盘校验。 目前该参数只支持输入为1。
输出参数	zone_id	int64	UndoZone编号。
输出参数	detail	text	校验出错信息。

示例，从磁盘中校验zone0到zone2的所有meta信息记录：

```
gaussdb=# SELECT * FROM gs_verify_undo_meta('all', 0, 2, 1);
zone_id | detail
--------+--------
(0 rows)
```

说明

调用此函数如有报错，请联系华为工程师处理。

- gs_async_rollback_worker_status()

  描述：活跃异步回滚线程状态监控。

  返回值类型：record

表 1-161 gs_async_rollback_worker_status 参数说明

参数类型	参数名	类型	描述
输出参数	datid	oid	数据库ID。
输出参数	pid	int8	线程ID。
输出参数	sessionid	int8	会话ID。
输出参数	usesysid	oid	发起该线程的用户ID。
输出参数	state	int	线程当前状态： ● 0：未定义。 ● 1：空闲。 ● 2：运行中。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	rollback_start_time	timestamptz	线程启动时间戳。
输出参数	idx	oid	异步回滚线程在数组中的下标。
输出参数	xid	xid	正在回滚的事务xid。
输出参数	progress	text	该事务的回滚进度（当前已回滚的Undo记录条数/总Undo记录条数，以百分比形式显示）。

- gs_async_rollback_xact_status()
  描述：异步回滚任务哈希表监控。
  返回值类型：record

  表 1-162 gs_async_rollback_xact_status 参数说明

参数类型	参数名	类型	描述
输出参数	xid	xid	需要进行异步回滚的事务xid。
输出参数	start_undoptr	xid	该事务起始Undo记录指针。
输出参数	end_undoptr	xid	该事务结束Undo记录指针。
输出参数	dbid	uint32	该事务所在数据库id。
输出参数	slot_ptr	xid	该事务对应事务槽的指针。
输出参数	launched	bool	是否有对应的活跃的异步回滚线程。

- gs_undo_recycler_status()
  描述：异步回收线程状态监控。
  返回值类型：record

版权所有 © 华为云计算技术有限公司

表 1-163 gs_undo_recycler_status 参数说明

参数类型	参数名	类型	描述
输出参数	datid	oid	数据库ID。
输出参数	pid	int8	线程ID。
输出参数	sessionid	int8	会话ID。
输出参数	usesysid	oid	发起该线程的用户ID。
输出参数	state	int	线程当前状态： ● 0：未定义。 ● 1：空闲。 ● 2：运行中。
输出参数	backend_start	timestamptz	线程启动时间戳。
输出参数	total_recycle_time	xid	总回收时间。
输出参数	max_recycle_time	xid	最大回收时间。
输出参数	total_recycle_size	xid	总回收空间。
输出参数	total_recycle_count	xid	总回收次数。
输出参数	recycle_sleep_count	xid	睡眠次数。
输出参数	recycle_sleep_time	xid	睡眠总时间。
输出参数	max_recycle_sleep_time	xid	最长睡眠时间。
输出参数	last_recycle_timestamp	timestamptz	上次成功回收时间戳。
输出参数	last_update_global_recycle_xid_timestamp	timestamptz	globalRecycleXid上次推进时间戳。

- gs_undo_launcher_status()
  描述：异步回滚发起线程状态监控。

版权所有 © 华为云计算技术有限公司

返回值类型：record

表 1-164 gs_undo_launcher_status 参数说明

参数类型	参数名	类型	描述
输出参数	datid	oid	数据库ID。
输出参数	pid	int64	线程ID。
输出参数	sessionid	int64	会话ID。
输出参数	usesysid	oid	发起该线程的用户ID。
输出参数	state	int32	线程当前状态： ● 0：未定义。 ● 1：空闲。 ● 2：运行中。
输出参数	backend_start	timestamptz	线程启动时间戳。
输出参数	total_async_rollback_task_count	uint64	本节点数据库启动后发起异步回滚任务总个数。
输出参数	average_async_rollback_time	uint64	异步回滚任务平均耗时。
输出参数	max_async_rollback_time	uint64	异步回滚任务最长耗时。
输出参数	min_async_rollback_time	uint64	异步回滚任务最短耗时。

1.6.27.14 HTAP 系统函数

- gs_global_imcv_info()

  描述：该函数用于显示当前数据库内IMCV表的基本信息，例如列存数据的加载情况、内存/磁盘空间占用等。分布式下仅支持在CN节点执行，结果返回所有DN节点的IMCV表的基本信息。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员才有权限执行此函数。

表 1-165 gs_global_imcv_info 参数说明

参数	参数	类型	描述
输出参数	node_name	text	节点名。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	一级分区名。
输出参数	subpartition_name	name	二级分区名。
输出参数	schema_name	name	Schema名。
输出参数	user_name	name	用户名。
输出参数	imcvnattr	smallint	加载列数量。
输出参数	imcvkey	int2vector	加载的列数组。
输出参数	priority	smallint	加载优先级（目前优先级暂不生效）。 ● '1'：PRIORITY_NONE无优先级。 ● '2'：PRIORITY_LOW低优先级。 ● '3'：PRIORITY_HIGH高优先级。
输出参数	rowgroup_num	integer	预期的行组数（即完全加载后应有的行组数）。
输出参数	populated_rowgroup_num	integer	实际加载的行组数（完全加载后该值应等于rowgroup_num）。
输出参数	cu_num_in_mem	integer	内存中的IMCU数量。
输出参数	cu_mem_size	bigint	IMCU占用的内存大小，单位为字节。
输出参数	cu_num_in_disk	integer	磁盘中的IMCU数量。

版权所有 © 华为云计算技术有限公司

参数	参数	类型	描述
输出参数	cu_disk_size	bigint	IMCU占用的磁盘大小，单位为字节。

示例

```
htapdb=# SELECT * FROM gs_global_imcv_info();
  node_name   |        rel_name         |       partition_name        | subpartition_name | schema_name |
user_name | imcvnattr |   imcvkey   | priority | rowgroup_num | populated_rowgroup_num |
cu_num_in_mem | cu_mem_size | cu_num_in_disk | cu_disk_size
--------------+-------------------------+-----------------------------+-------------------+-------------+-----------
+-----------+-------------+----------+--------------+-----------------------+--------------+-------------
+----------------+--------------
 dn_6003_6004 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |                   | public      |
***       |        6 | 1 2 3 4 5 6 |        1 |            1 |                   1 |            7 |       24228 |             0
|              0
 dn_6003_6004 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      |
***       |        6 | 1 2 3 4 5 6 |        1 |            1 |                   1 |            7 |        8048 |             0
|              0
 dn_6003_6004 | htap_flush_by_name_part1 |                             |                   | public      | ***
|        6 | 1 2 3 4 5 6 |        1 |            0 |                   0 |            0 |            0 |             0 |             0
 dn_6001_6002 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |                   | public      |
***       |        6 | 1 2 3 4 5 6 |        1 |            0 |                   0 |            0 |            0 |             0
|              0
 dn_6001_6002 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      |
***       |        6 | 1 2 3 4 5 6 |        1 |            1 |                   1 |            7 |        8048 |             0
|              0
 dn_6001_6002 | htap_flush_by_name_part1 |                             |                   | public      | ***
|        6 | 1 2 3 4 5 6 |        1 |            0 |                   0 |            0 |            0 |             0 |             0
(6 rows)
```

- gs_htap_imcv_info()

  描述：该函数用于显示当前数据库内IMCV表的基本信息，例如列存数据的加载情况、内存/磁盘空间占用等。分布式下执行该函数返回当前节点的信息，查询所有节点信息可使用gs_global_imcv_info()函数。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。

表 1-166 gs_htap_imcv_info 参数说明

参数	参数	类型	描述
输出参数	relname	name	表名。
输出参数	partition_name	name	一级分区名。
输出参数	subpartition_name	name	二级分区名。

版权所有 © 华为云计算技术有限公司

参数	参数	类型	描述
输出参数	schema_name	name	Schema的名字。
输出参数	user_name	name	用户名。
输出参数	imcvnattr	smallint	加载列数量。
输出参数	imcvkey	int2vector	加载的列数组。
输出参数	priority	smallint	加载优先级。（目前优先级暂不生效） ● '1'：PRIORITY_NONE无优先级。 ● '2'：PRIORITY_LOW低优先级。 ● '3'：PRIORITY_HIGH高优先级。
输出参数	rowgroup_num	integer	预期的行组数（即完全加载后应有的行组数）。
输出参数	populated_rowgroup_num	integer	实际加载的行组数（完全加载后该值应等于rowgroup_num）。
输出参数	cu_num_in_mem	integer	内存中的IMCU数量。
输出参数	cu_mem_size	bigint	IMCU占用的内存大小，单位为字节。
输出参数	cu_num_in_disk	integer	磁盘中的IMCU数量。
输出参数	cu_disk_size	bigint	IMCU占用的磁盘大小，单位为字节。

示例：

```
htapdb=# SELECT * FROM gs_htap_imcv_info();
        rel_name         |       partition_name        | subpartition_name | schema_name | user_name |
imcvnattr |   imcvkey   | priority | rowgroup_num | populated_rowgroup_num | cu_num_in_mem |
cu_mem_size | cu_num_in_disk | cu_disk_size
-------------------------+-----------------------------+-------------------+-------------+-----------+-----------
+-------------+----------+--------------+------------------------+---------------+-------------+----------------
+--------------
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |                   | public      | ***       |         6
| 1 2 3 4 5 6 |         1 |           0 |                      0 |             0 |           0 |          0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      | ***       |         6
| 1 2 3 4 5 6 |         1 |           0 |                      0 |             0 |           0 |          0
 htap_flush_by_name_part1 |                             |                   | public      | ***       |         6 | 1 2 3 4 5 6
|         1 |           0 |                      0 |             0 |           0 |           0 |          0
(3 rows)
```

- gs_imcv_flush(reloid oid, partoid oid, parallelism integer, rebuild_mode integer)

  描述：对数据库中指定的IMCV表或者分区立即进行一次重建，并显示重建的结果。重建过程将刷新IMCV表，移除旧数据并同步当前表的非活跃数据。在CN节

版权所有 © 华为云计算技术有限公司

点执行时，重建所有节点上该表的IMCV；在DN节点执行时，重建该节点上该表的IMCV。

返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或运维管理员才有权限执行此函数。
- 该函数不允许在事务块内执行。
- 在执行并行加载/重建时，每个worker都会额外占用内存、CPU和I/O（开启落盘）等资源，请根据实际硬件资源、系统负载谨慎控制加载或重建的并行度。

表 1-167 gs_imcv_flush 参数说明

参数类型	参数名	类型	描述
输入参数	reloid	oid	表oid。
输入参数	partoid	oid	分区oid。当查询分区表所有分区或非分区表时，该值可传入0。
输入参数	parallelism	integer	并行度。即加载/重建任务利用的bgworker数量。取值范围为1~256。实际后台bgworker数量不足时，使用实际能够拉起的bgworker执行重建任务。
输入参数	rebuild_mode	integer	重建模式。取值范围： ● 0：表示非强制重建，仅当列存数据未同步到行存时会触发重建。 ● 1：表示强制重建，即无论列存数据是否为最新状态，都会触发重建。
输出参数	total	integer	实际处理的行组数。
输出参数	success	integer	成功处理的行组数。
输出参数	error	integer	处理失败的行组数，例如因为内存不足，未实际发生加载或者重建操作。
输出参数	skip	integer	跳过处理的行组数，例如数据足够新鲜，未发生重建操作。

示例：

```
htapdb=# SELECT * FROM gs_imcv_flush(20834, 20838,1,1);
 total | success | error | skip
-------+---------+-------+------
```


版权所有 © 华为云计算技术有限公司

```
1    | 1     | 0    | 0
(1 row)
```

- gs_imcv_flush(relname text, partname text, parallelism integer, rebuild_mode integer)

  描述：对数据库中指定的IMCV表或者分区立即进行一次重建，并显示重建的结果。重建过程将刷新IMCV表，移除旧数据并同步当前表的非活跃数据。在CN节点执行时，重建所有节点上该表的IMCV；在DN节点执行时，重建该节点上该表的IMCV。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或运维管理员才有权限执行此函数。
- 该函数不允许在事务块内执行。
- 在执行并行加载/重建时，每个worker都会额外占用内存、CPU和I/O（开启落盘）等资源，请根据实际硬件资源、系统负载谨慎控制加载或重建的并行度。

表 1-168 gs_imcv_flush 参数说明

参数类型	参数名	类型	描述
输入参数	relname	text	表名。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。可缺省，缺省时如果表包含分区，则处理所有分区。
输入参数	parallelism	integer	并行度。即加载/重建任务利用的bgworker数量。取值范围为1~256。实际后台bgworker数量不足时，使用实际能够拉起的bgworker执行重建任务。可缺省，缺省时默认并行度为1，即无并行。
输入参数	rebuild_mode	integer	重建模式。取值范围： ● 0：表示非强制重建，仅当列存数据未同步到行存时会触发重建。 ● 1：表示强制重建，即列存数据是否新鲜都会触发重建。 可缺省，缺省时代表非强制重建。
输出参数	total	integer	实际处理的行组数。
输出参数	success	integer	成功处理的行组数。
输出参数	error	integer	处理失败的行组数，例如因为内存不足，未实际发生加载或者重建操作。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	skip	integer	跳过处理的行组数，例如数据足够新鲜，未发生重建操作。

示例：

```
--缺省partname、parallelism、rebuild_mode。
htapdb=# SELECT * FROM gs_imcv_flush('htap_flush_by_name_part1');
 total | success | error | skip
-------+---------+-------+------
 2     | 2       | 0     | 0
(1 row)
--缺省parallelism、rebuild_mode。
htapdb=# SELECT * FROM gs_imcv_flush('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p1');
 total | success | error | skip
-------+---------+-------+------
 1     | 0       | 0     | 1
(1 row)
--缺省rebuild_mode。
htapdb=# SELECT * FROM gs_imcv_flush('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p1',1);
 total | success | error | skip
-------+---------+-------+------
 1     | 0       | 0     | 1
(1 row)
htapdb=# SELECT * FROM gs_imcv_flush('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p1',1,1);
 total | success | error | skip
-------+---------+-------+------
 1     | 1       | 0     | 0
(1 row)
```

- gs_global_htap_tmu_data(relname text, partname text, rowgroup_id integer)

  描述：查看实时事务单元TMU中的活跃事务数据，只能在分布式CN下执行，返回所有DN节点的信息。当rowgroup_id为0时，返回表中所有活跃事务数据；当rowgroup_id>0时，返回表中指定行组中的活跃事务数据。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员才有权限执行此函数。

表 1-169 gs_global_htap_tmu_data 参数说明

参数类型	参数名	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输入参数	rowgroup_id	integer	行组唯一标识。
输出参数	node_name	text	节点名。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	一级分区名，如果非一级分区，该字段为空。
输出参数	subpartition_name	name	二级分区名，如果非二级分区，该字段为空。
输出参数	schema_name	name	Schema名。
输出参数	rel_oid	oid	表oid。
输出参数	rowgroup_id	integer	行组唯一标识。
输出参数	page_id	integer	所查行组中页面id。
输出参数	flag	smallint	0代表非脏页，1代表脏页。
输出参数	xid	xid	操作该数据的事务id。

示例：

```
htapdb=# SELECT * FROM gs_global_htap_tmu_data('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p1', 0);
 node_name |          rel_name          |       partition_name        | subpartition_name | schema_name |
rel_oid | rowgroup_id | page_id | flag |  xid
-----------+----------------------------+-----------------------------+-------------------+-------------+---------
+-------------+---------+------+--------
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |                   | public      |
16389 |           1 |       0 |    0 | 203236
(1 row)
```

- gs_htap_tmu_data(relname text, partname text, rowgroup_id integer)

  描述：查看实时事务单元TMU中的活跃事务数据，返回当前节点表中活跃事务数据。当rowgroup_id为0时，返回表中所有活跃事务数据；当rowgroup_id>0时，返回表中指定行组的活跃事务数据。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或运维管理员才有权限执行此函数。

表 1-170 gs_htap_tmu_data 参数说明

参数类型	参数名	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输入参数	rowgroup_id	integer	行组唯一标识。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	一级分区名，如果非一级分区，该字段为空。
输出参数	subpartition_name	name	二级分区名，如果非二级分区，该字段为空。
输出参数	schema_name	name	Schema名。
输出参数	rel_oid	oid	表oid。
输出参数	rowgroup_id	integer	行组唯一标识。
输出参数	page_id	integer	所查行组中页面id。
输出参数	flag	smallint	0代表非脏页，1代表脏页。
输出参数	xid	xid	操作该数据的事务id。

示例：

```
htapdb=# SELECT * FROM gs_htap_tmu_data('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p2',0);
         rel_name          |       partition_name        | subpartition_name | schema_name | rel_oid |
rowgroup_id | page_id | flag | xid
---------------------------+-----------------------------+-------------------+-------------+---------+-------------
+---------+------+-----
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      |   16390 |
1 |       0 |    0 |   0
(1 row)
```

- gs_global_htap_tmu_chunk_meta(relname name, partname name, rowgroup_id integer)

  描述：查看实时事务单元TMU元数据信息，只能在分布式CN下执行，返回所有DN节点的信息。当rowgroup_id为0时，返回所有行组中的实时事务单元TMU数

版权所有 © 华为云计算技术有限公司

据信息；当rowgroup_id>0时，返回表中指定行组中的实时事务单元TMU数据信息；当rowgroup_id为-1时，统计所有行组中的实时事务单元TMU数据后以单条返回。

返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员才有权限执行此函数。

表 1-171 gs_global_htap_tmu_chunk_meta 参数说明

参数类型	参数名	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输入参数	rowgroup_id	integer	行组唯一标识。
输出参数	node_name	text	数据库节点名。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	一级分区名，如果非一级分区，该字段为空。
输出参数	subpartition_name	name	二级分区名，如果非二级分区，该字段为空。
输出参数	schema_name	name	Schema名。
输出参数	rel_oid	oid	表oid。
输出参数	rowgroup_id	integer	行组唯一标识。
输出参数	page_count	bigint unsigned	所查行组中页面数量。
输出参数	dirty_page_count	bigint unsigned	所查行组中脏页面数量。
输出参数	clean_page_count	bigint unsigned	所查行组中非脏页面数量。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	dirty_ratio	real	所查行组中脏页占全部页面数量比例。

示例：

```
htapdb=# SELECT * FROM gs_global_htap_tmu_chunk_meta('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p2', 0);
 node_name |         rel_name         |       partition_name        | subpartition_name | schema_name |
rel_oid | rowgroup_id | page_count | dirty_page_count | clean_page_count | dirty_ratio
-----------+--------------------------+-----------------------------+-------------------+-------------+---------
+-------------+------------+------------------+------------------+-------------
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      |
16390 |           1 |          1 |                0 |                1 |           0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      |
16390 |           1 |          1 |                0 |                1 |           0
(2 rows)
```

- gs_htap_tmu_chunk_meta(relname text, partname text, rowgroup_id integer)

  描述：查看实时事务单元TMU元数据信息，分布式下执行返回当前节点的信息。当rowgroup_id为0时，返回所有行组中的实时事务单元TMU数据信息；当rowgroup_id>0时，返回表中指定行组中的实时事务单元TMU数据信息；当rowgroup_id为-1时，统计所有行组中的实时事务单元TMU数据后以单条返回。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。

表 1-172 gs_htap_tmu_chunk_meta 参数说明

参数类型	参数名	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输入参数	rowgroup_id	integer	行组唯一标识。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	一级分区名，如果非一级分区，该字段为空。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	subpartition_name	name	二级分区名，如果非二级分区，该字段为空。
输出参数	schema_name	name	Schema名。
输出参数	rel_oid	oid	表oid。
输出参数	rowgroup_id	integer	行组唯一标识。
输出参数	page_count	bigint unsigned	所查行组中页面数量。
输出参数	dirty_page_count	bigint unsigned	所查行组中脏页面数量。
输出参数	clean_page_count	bigint unsigned	所查行组中非脏页面数量。
输出参数	dirty_ratio	real	所查行组中脏页占全部页面数量比例。

示例：

```
htapdb=# SELECT * FROM gs_htap_tmu_chunk_meta('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p2', 0);
        rel_name         |       partition_name        | subpartition_name | schema_name | rel_oid |
rowgroup_id | page_count | dirty_page_count | clean_page_count | dirty_ratio
-------------------------+-----------------------------+-------------------+-------------+---------+-------------
+------------+------------------+------------------+-------------
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public      |   16390 |
1 | 1          | 0                | 1                |                0
(1 row)
```

- gs_global_imcv_bgworker_status()

  描述：显示执行IMCV表重建的bgworker的状态信息，分布式下仅支持在CN节点执行，结果返回所有DN节点的信息。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员才有权限执行此函数。

版权所有 © 华为云计算技术有限公司

表 1-173 gs_global_imcv_bgworker_status 参数说明

参数类型	参数名	类型	描述
输出参数	node_name	text	数据库节点名。
输出参数	worker_pid	bigint	线程号。
输出参数	database_id	oid	数据库id。
输出参数	relation_id	oid	正在处理的IMCV表的oid。
输出参数	rowgroup_id	integer	行组唯一标识。
输出参数	worker_status	text	表示init（初始化）、running（进行中）、ready（空闲）三种线程工作状态。
输出参数	worker_type	text	任务类型，包含以下五种： ● background_invalid_worker：无任务在执行。 ● background_unknown_worker：未知任务。 ● background_rowgroup_rebuild_worker：重建行组。 ● background_async_populate_worker：异步加载。 ● background_cache_worker：缓存区任务。
输出参数	start_time	timestamp with time zone	线程启动的时间戳。

示例：

```
htapdb=# SELECT * FROM gs_global_imcv_bgworker_status();
 node_name |    worker_pid    | database_id | relation_id | rowgroup_id | worker_status |
worker_type         |          start_time
-----------+------------------+-------------+-------------+-------------+---------------+--------------------------
+-------------------------------
 datanode1 | 140067573069568 |       16384 |           0 |           0 | ready         |
background_invalid_worker | 2025-12-02 21:10:16.282394+08
 datanode1 | 140067645421312 |       16384 |           0 |           0 | ready         |
background_invalid_worker | 2025-12-02 21:10:12.267694+08
 datanode1 | 140067608721152 |           0 |           0 |           0 | ready         |
background_invalid_worker | 2025-12-02 21:10:12.332674+08
 datanode2 | 139803070822144 |       16384 |           0 |           0 | ready         |
background_invalid_worker | 2025-12-02 21:10:16.274911+08
 datanode2 | 139803270051584 |       16384 |           0 |           0 | ready         |
background_invalid_worker | 2025-12-02 21:10:16.284832+08
 datanode2 | 139803153659648 |           0 |           0 |           0 | ready         |
```


版权所有 © 华为云计算技术有限公司

```
background_invalid_worker | 2025-12-02 21:10:12.374876+08
(6 rows)
```

- gs_imcv_bgworker_status()

  描述：显示执行IMCV表重建的bgworker的状态信息，分布式下执行返回当前节点的信息。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。

表 1-174 gs_imcv_bgworker_status 参数说明

参数类型	参数名	类型	描述
输出参数	worker_pid	bigint	线程号。
输出参数	database_id	oid	数据库id。
输出参数	relation_id	oid	正在处理的IMCV表的oid。
输出参数	rowgroup_id	integer	行组唯一标识。
输出参数	worker_status	cstring	表示init（初始化）、running（进行中）、ready（空闲）三种线程工作状态。
输出参数	worker_type	cstring	任务类型，包含background_invalid_worker（无任务在执行）、background_unknown_worker（未知任务）、background_rowgroup_rebuild_worker（重建行组）、background_async_populate_worker（异步加载）、background_cache_worker（缓存区任务）五种任务类型。
输出参数	start_time	timestamp with time zone	线程启动的时间戳。

示例：

版权所有 © 华为云计算技术有限公司

```
htapdb=# SELECT * FROM gs_imcv_bgworker_status();
   worker_pid    | database_id | relation_id | rowgroup_id | worker_status |        worker_type
|          start_time
-----------------+-------------+-------------+-------------+---------------+---------------------------
+-------------------------------
 140067573069568 |       16384 |           0 |           0 | ready         | background_invalid_worker |
2025-12-02 21:10:16.282394+08
 140067645421312 |       16384 |           0 |           0 | ready         | background_invalid_worker |
2025-12-02 21:10:12.267694+08
 140067608721152 |           0 |           0 |           0 | ready         | background_invalid_worker |
2025-12-02 21:10:12.332674+08
(3 rows)
```

- gs_global_imcu_meta(rel_name text, partname text)

  描述：用于显示当前数据库下指定rel_name的IMCV表下的IMCU元信息，分布式下仅支持在CN节点执行，结果返回所有DN节点的信息。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员才有权限执行此函数。

表 1-175 gs_global_imcu_meta 参数说明

参数类型	参数	类型	描述
输入参数	relname	text	IMCV表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	IMCV分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输出参数	node_name	text	数据库节点名字。
输出参数	rel_name	name	IMCV表名。
输出参数	partition_name	name	IMCV一级分区名。
输出参数	subpartition_name	name	IMCV二级分区名。
输出参数	schema_name	name	Schema名。
输出参数	rowgroup_id	integer	IMCU所在行组id。
输出参数	row_count	integer	IMCU行数。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	imcu_size	integer	IMCU大小，单位字节。
输出参数	imcu_data_size	integer	IMCU数据大小（不包含IMCU元信息），单位字节。
输出参数	compressed_data_size	integer	IMCU数据压缩后的大小。
输出参数	compressed_ratio	real	IMCU数据的压缩比。
输出参数	compressed_type	text	IMCU的压缩算法： ● none：无IMCU数据，无法压缩。 ● uncompressed：未进行压缩。 ● delta_plus_rle：IMCU数据采用delta_plus_rle压缩算法进行压缩。 ● lz4_compressed：IMCU数据采用lz4压缩算法进行压缩。 ● dict_compressed：IMCU数据采用dict压缩算法进行压缩。
输出参数	imcu_mode	integer	IMCU加载列数据特征： ● 1：普通。 ● 2：全部为空。 ● 3：相同值。 ● 4：无最大最小值。 ● 8：有空行。 ● 16：无求和值。
输出参数	magic	integer	生成IMCU数据时的事务xid值，用于校验IMCU数据是否有效。
输出参数	attnum	smallint	IMCU加载列对应的attnum。
输出参数	scan_cnt	bigint	扫描IMCU的总次数。
输出参数	hit_cnt	bigint	命中内存态IMCU的次数。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	data_location	smallint	标记IMCU位于内存中、位于磁盘中或处于escape状态。 ● 0：内存。 ● 1：磁盘。 ● 2：escape状态，因内存不足，仅创建元数据，查询这部分数据时会回行存读取。

示例：

```
htapdb=# SELECT * FROM gs_global_imcu_meta('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p2');
 node_name |        rel_name        |       partition_name       | subpartition_name | schema_name |
rowgroup_id | row_count | imcu_size | imcu_data_size | compressed_data_size | compressed_ratio |
compressed_type | imcu_mode | magic | attnum | scan_cnt | hit_cnt | data_location
-----------+------------------------+----------------------------+-------------------+-------------
+-----------+-----------+----------------+----------------------+------------------+-----------------+-----------
+-------+--------+----------+---------+---------------
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |      4016 |           16 |                16 |              1 | uncompressed    |        1 |
0 |      0 |        0 |       0 |            0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
1 |        0 |       0 |             0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
2 |        0 |       0 |             0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
3 |        0 |       0 |             0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
4 |        0 |       0 |             0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |      4032 |           24 |                24 |              1 | uncompressed    |        4 |
0 |      5 |        0 |       0 |            0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
6 |        0 |       0 |             0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |      4016 |           16 |                16 |              1 | uncompressed    |        1 |
0 |      0 |        0 |       0 |            0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
1 |        0 |       0 |             0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
2 |        0 |       0 |             0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
3 |        0 |       0 |             0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
4 |        0 |       0 |             0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |      4032 |           24 |                24 |              1 | uncompressed    |        4 |
0 |      5 |        0 |       0 |            0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                   | public
|         1 |         1 |         0 |            0 |                 0 |              0 | none            |        3 |     0 |
6 |        0 |       0 |             0
(14 rows)
```


版权所有 © 华为云计算技术有限公司

备注：该示例每一行代表一个IMCU（其中attnum为0的列为CTID列）。

- gs_imcu_meta(rel_name text, partname text)

  描述：用于显示当前数据库下指定rel_name的IMCV表下的IMCU元信息，分布式下执行返回当前节点的信息。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。

表 1-176 gs_imcu_meta 参数说明

参数类型	参数	类型	描述
输入参数	relname	text	IMCV表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	IMCV分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输出参数	rel_name	name	IMCV表名。
输出参数	partition_name	name	IMCV一级分区名。
输出参数	subpartition_name	name	IMCV二级分区名。
输出参数	schema_name	name	Schema名。
输出参数	rowgroup_id	integer	IMCU所在行组id。
输出参数	row_count	integer	IMCU行数。
输出参数	imcu_size	integer	IMCU大小，单位字节。
输出参数	imcu_data_size	integer	IMCU数据大小（不包含IMCU元信息），单位字节。
输出参数	compressed_data_size	integer	IMCU数据压缩后的大小。
输出参数	compressed_ratio	real	IMCU数据的压缩比。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	compressed_type	text	IMCU的压缩算法： ● none：无IMCU数据，无法压缩。 ● uncompressed：未进行压缩。 ● delta_plus_rle：IMCU数据采用delta_plus_rle压缩算法进行压缩。 ● lz4_compressed：IMCU数据采用lz4压缩算法进行压缩。 ● dict_compressed：IMCU数据采用dict压缩算法进行压缩。
输出参数	imcu_mode	integer	IMCU加载列数据特征： ● 1：普通。 ● 2：全部为空。 ● 3：相同值。 ● 4：无最大最小值。 ● 8：有空行。 ● 16：无求和值。
输出参数	magic	integer	生成IMCU数据时的事务xid值，用于校验IMCU数据是否有效。
输出参数	attnum	smallint	IMCU加载列对应的attnum。
输出参数	scan_cnt	bigint unsigned	扫描IMCU的总次数。
输出参数	hit_cnt	bigint unsigned	命中内存态IMCU的次数。
输出参数	data_location	smallint	标记IMCU位于内存中、位于磁盘中或处于escape状态。 ● 0：内存。 ● 1：磁盘。 ● 2：escape状态，因内存不足，仅创建元数据，查询这部分数据时会回行存读取。

示例：

```
htapdb=# SELECT * FROM gs_imcu_meta('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p2');
      rel_name       |      partition_name       | subpartition_name | schema_name | rowgroup_id |
```


版权所有 © 华为云计算技术有限公司

```
row_count | imcu_size | imcu_data_size | compressed_data_size | compressed_ratio | compressed_type |
imcu_mode | magic | attnum | scan_cnt | hit_cnt | data_location
-------------------------+--------------------------+------------------+-------------+-------------+----------
+----------+---------------+--------------------+------------------+----------------+-----------+-------
+--------+----------+---------+---------------
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |     4016 |           16 |                16 |              1 | uncompressed   |         1 |     0 |     0 | 0
| 0      |           0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |        0 |            0 |                 0 |              0 | none           |         3 |     0 |     1 | 0       | 0
|           0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |        0 |            0 |                 0 |              0 | none           |         3 |     0 |     2 | 0       | 0
|           0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |        0 |            0 |                 0 |              0 | none           |         3 |     0 |     3 | 0       | 0
|           0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |        0 |            0 |                 0 |              0 | none           |         3 |     0 |     4 | 0       | 0
|           0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |     4032 |           24 |                24 |              1 | uncompressed   |         4 |     0 |     5 | 0
| 0      |           0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |                  | public     |         1 |
1 |        0 |            0 |                 0 |              0 | none           |         3 |     0 |     6 | 0       | 0
|           0
(7 rows)
```

备注：该示例每一行代表一个IMCU（其中attnum为0的列为CTID列）。

- gs_global_htap_file_status(db_name text)

  描述：查看HTAP列存IMCU落盘文件的使用信息，分布式下仅支持在CN节点执行，结果返回所有DN节点的信息。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员才有权限执行此函数。

表 1-177 gs_global_htap_file_status 参数说明

参数类型	参数	类型	描述
输入参数	db_name	text	数据库名字，输入为''或NULL表示查询所有数据库。
输出参数	node_name	text	数据库节点名字。
输出参数	db_oid	integer	数据库id。
输出参数	file_name	text	数据文件名。
输出参数	file_id	integer	数据文件标识，当前版本返回值只有2。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	extent_size	integer	数据文件的扩展大小。
输出参数	allocated_blocks	bigint	数据文件已分配给IMCU使用的页面数。
输出参数	total_blocks	bigint	数据文件在磁盘上的总物理页面数。
输出参数	high_water_mark	bigint	数据文件使用页数的高水位线。
输出参数	utilization	real	使用的block数占总block数的百分比。
输出参数	ext	text	扩展字段，当前版本为NULL。

示例：

```
htapdb=# SELECT * FROM gs_global_htap_file_status('htapdb');
 node_name | db_oid |        file_name          | file_id | extent_size | allocated_blocks | total_blocks |
high_water_mark | utilization | ext
-----------+--------+---------------------------+---------+-------------+------------------+--------------
+-----------------+-------------+-----
 datanode1 |  16384 | global/imcu/16384/2_imcu |       2 |           4 |                0 |            0 |
0 |           1 |
 datanode2 |  16384 | global/imcu/16384/2_imcu |       2 |           4 |                0 |            0 |
0 |           1 |
(2 rows)
```

- gs_htap_file_status(db_name text)

  描述：查看HTAP列存IMCU落盘文件的使用信息，分布式下执行返回当前节点的信息。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。

表 1-178 gs_htap_file_status 参数说明

参数类型	参数	类型	描述
输入参数	db_name	text	数据库名字，输入为''或NULL表示查询当前节点下所有数据库。
输出参数	db_oid	oid	数据库id。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	file_name	text	数据文件名。
输出参数	file_id	integer	数据文件标识，当前版本返回值只有2。
输出参数	extent_size	integer	数据文件的扩展大小。
输出参数	allocated_blocks	bigint	数据文件已分配给IMCU使用的页面数。
输出参数	total_blocks	bigint	数据文件在磁盘上的总物理页面数。
输出参数	high_water_mark	bigint	数据文件使用页数的高水位线。
输出参数	utilization	real	使用的block数占总block数的百分比。
输出参数	ext	text	扩展字段，当前版本为NULL。

示例：

```
htapdb=# SELECT * FROM gs_htap_file_status('htapdb');
 db_oid |        file_name          | file_id | extent_size | allocated_blocks | total_blocks | high_water_mark |
utilization | ext
--------+---------------------------+---------+-------------+------------------+--------------+-----------------
+-------------+-----
  16384 | global/imcu/16384/2_imcu |       2 |           4 |                0 |            0 |               0 |           1 |
(1 row)
```

- gs_global_htap_scan_hit_status(relname text, partname text)

  描述：用于查询表粒度的IMCU命中率信息，分布式下仅支持在CN节点执行，结果返回所有DN节点的信息。当reloid为0时，返回当前数据库下所有IMCV表的命中率信息；当reloid非0时，返回指定表中的命中率信息。

  返回值类型：record

  须知

  - 仅参数enable_htap=on时可用。
  - 仅系统管理员才有权限执行此函数。

版权所有 © 华为云计算技术有限公司

表 1-179 gs_global_htap_scan_hit_status 参数说明

参数类型	参数	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输出参数	node_name	text	数据库节点名。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	分区名。
输出参数	subpartition_name	name	二级分区名。
输出参数	schema_name	name	Schema名。
输出参数	column_number	integer	列序号。
输出参数	column_name	name	列名。
输出参数	imcu_in_cache_cnt	bigint unsigned	该列存放于缓存区的IMCU数。
输出参数	imcu_ave_scan_cnt	bigint unsigned	该列平均每个IMCU的使用次数。
输出参数	imcu_total_scan_cnt	bigint unsigned	该列IMCU总的使用次数。
输出参数	imcu_ave_hit_cnt	bigint unsigned	该列平均每个IMCU的缓存命中数。
输出参数	imcu_total_hit_cnt	bigint unsigned	该列IMCU总的缓存命中数。
输出参数	total_hit_percentage	real	该列IMCU总的缓存命中率。

示例：

```
htapdb=# SELECT * FROM gs_global_htap_scan_hit_status('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p2');
 node_name |       rel_name       |       partition_name       | subpartition_name | schema_name |
column_number | column_name | imcu_in_cache_cnt | imcu_ave_scan_cnt | imcu_total_scan_cnt |
imcu_ave_hit_cnt | imcu_total_hit_cnt | total_hit_percentage
-----------+----------------------+----------------------------+-------------------+-------------
+---------------+-------------+-------------------+-------------------+---------------------+------------------
```


版权所有 © 华为云计算技术有限公司

```
+--------------------+----------------------
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          0 | ctid       | 1             | 0             | 0             | 0             | 0
|                0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          1 | c1         | 1             | 0             | 0             | 0             | 0
|                0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          2 | c2         | 1             | 0             | 0             | 0             | 0
|                0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          3 | c3         | 1             | 0             | 0             | 0             | 0
|                0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          4 | c4         | 1             | 0             | 0             | 0             | 0
|                0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          5 | c5         | 1             | 0             | 0             | 0             | 0
|                0
 datanode1 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          6 | c6         | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          0 | ctid       | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          1 | c1         | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          2 | c2         | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          3 | c3         | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          4 | c4         | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          5 | c5         | 1             | 0             | 0             | 0             | 0
|                0
 datanode2 | htap_flush_by_name_part1 | htap_flush_by_name_part1_p2 |            | public
|          6 | c6         | 1             | 0             | 0             | 0             | 0
|                0
(14 rows)
```

- gs_htap_scan_hit_status(relname text, partname text)

  描述：用于查询表粒度的IMCU命中率信息，分布式下执行返回当前节点的信息。当relname与partname均为null或空时，返回当前节点下所有IMCV表的命中率信息；其他情况返回指定表或指定分区的命中率信息。

  返回值类型：record

  须知

  - 仅参数enable_htap=on时可用。
  - 仅系统管理员或者运维管理人员才有权限执行此函数。

版权所有 © 华为云计算技术有限公司

表 1-180 gs_htap_scan_hit_status 参数说明

参数类型	参数	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名，查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输出参数	rel_name	name	表名。
输出参数	partition_name	name	分区名。
输出参数	subpartition_name	name	二级分区名。
输出参数	schema_name	name	Schema名。
输出参数	column_number	integer	列序号。
输出参数	column_name	name	列名。
输出参数	imcu_in_cache_cnt	bigint unsigned	该列存放于缓存区的IMCU数。
输出参数	imcu_total_cnt	bigint unsigned	该列总的IMCU数。
输出参数	imcu_ave_scan_cnt	bigint unsigned	该列平均每个IMCU的使用次数。
输出参数	imcu_total_scan_cnt	bigint unsigned	该列IMCU总的使用次数。
输出参数	imcu_ave_hit_cnt	bigint unsigned	该列平均每个IMCU的缓存命中数。
输出参数	imcu_total_hit_cnt	bigint unsigned	该列IMCU总的缓存命中数。
输出参数	total_hit_percentage	real	该列IMCU总的缓存命中率。

示例：

```
gaussdb=# SELECT * FROM gs_htap_scan_hit_status('htap_flush_by_name_part1',
'htap_flush_by_name_part1_p1');
        rel_name         |       partition_name        | subpartition_name | schema_name | column_number
| column_name | imcu_in_cache_cnt | imcu_total_cnt | imcu_ave_scan_cnt | imcu_total_scan_cnt |
imcu_ave_hit_cnt | imcu_total_hit_cnt | tot
al_hit_percentage
-------------------------+-----------------------------+-------------------+-------------+---------------
```


版权所有 © 华为云计算技术有限公司

```
+-------------+------------------+----------------+------------------+--------------------+------------------
+-------------------+----
------------------
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          0 |
ctid        | 1             | 1           | 0             | 0             | 0           | 0             |
            0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          1 |
c1          | 1             | 1           | 0             | 0             | 0           | 0             |
            0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          2 |
c2          | 1             | 1           | 0             | 0             | 0           | 0             |
            0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          3 |
c3          | 1             | 1           | 0             | 0             | 0           | 0             |
            0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          4 |
c4          | 1             | 1           | 0             | 0             | 0           | 0             |
            0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          5 |
c5          | 1             | 1           | 0             | 0             | 0           | 0             |
            0
 htap_flush_by_name_part1 | htap_flush_by_name_part1_p1 |              | public     |          6 |
c6          | 1             | 1           | 0             | 0             | 0           | 0             |
            0
(7 rows)
```

- gs_imcu_size_estimation(relname text, partname text, colnum_name_array text[])

  描述：该函数返回该表指定分区指定列创建IMCV列存数据预估的内存使用值，DN节点执行时返回当前节点预估值，CN节点执行返回各个节点预估值。

  返回值类型：record

须知

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。
- 不支持备机执行。该函数约束与HTAP DDL保持一致，相关报错信息与实际执行DDL报错信息一致。

表 1-181 gs_imcu_size_estimation 参数说明

参数类型	参数名	类型	描述
输入参数	relname	text	表名。如果查询跨Schema的表信息，支持传入'schema.relname'形式。
输入参数	partname	text	分区名。查询分区表所有分区或非分区表时，该值可传入NULL或者''。
输入参数	colnum_name_array	text[]	列名数组。
输出参数	node_name	name	节点名。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	estimated_size	bigint unsigned	预估内存使用值，单位字节。

示例：

```
htapdb=# SELECT * FROM gs_imcu_size_estimation('estimate_htap', '', ARRAY['id']);
 node_name | estimated_size
-----------+----------------
 datanode1 | 28411
 datanode2 | 28089
(2 rows)
```

- gs_htap_data_maintenance(rel_oid oid, type integer)

  描述：列存数据的维护接口，提供两项能力（1）卸载列存数据（清理内存和磁盘空间）;（2）所有列存数据落盘的逃生手段。

  返回值类型：text

须知

该接口会直接卸载IMCV保护行存TP业务，会导致已经使用IMCV Scan的业务报错，使用前需充分确认影响。仅供因配置或其他原因导致IMCV内存等资源使用过量，集群又无法停机时，紧急逃生使用。

说明

- 仅参数enable_htap=on时可用。
- 仅系统管理员或者运维管理人员才有权限执行此函数。
- 该函数不允许在事务内执行。
- 分布式下该函数只能在DN执行，不支持在CN执行。

表 1-182 gs_htap_data_maintenance 参数说明

参数类型	参数	类型	描述
输入参数	rel_oid	oid	行存表oid，0表示处理当前DN节点的数据库下所有表。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输入参数	type	integer	处理类型，取值范围： ● 0：表示卸载对应表的所有IMCV数据，包括内存和落盘数据文件，仅保留元数据，gs_imcv系统表还存在信息，通过gs_imcv_status视图查询状态为'E'。入参rel_oid为0时，卸载所有表IMCV数据，落盘文件会被清理，磁盘空间也会释放。使用该接口卸载IMCV后，IMCV状态为'E'，此时无法使用HTAP相关功能，需要重新创建IMCV或执行VACUUM FULL将基表与对应分区的IMCV状态转为'F' 。 ● 1：该参数已废弃。 ● 2：表示将对应表的IMCV数据全部落盘。

示例：

```
htapdb=# SELECT * FROM gs_htap_data_maintenance(16824,0);
 gs_htap_data_maintenance
--------------------------
 Maintenance finish.
(1 row)
htapdb=# SELECT * FROM gs_htap_data_maintenance(16824,2);
 gs_htap_data_maintenance
--------------------------
 Maintenance finish.
(1 row)
```

1.6.27.15 其它函数

- pgxc_pool_check()
  描述：检查连接池中缓存的连接数据是否与pgxc_node一致。
  返回值类型：Boolean
- pgxc_pool_reload()
  描述：更新连接池中缓存的连接信息。
  返回值类型：Boolean
- reload_active_coordinator()
  描述：对所有存活的CN，更新连接池中缓存的连接信息。
  返回值类型：void
- pgxc_lock_for_backup()
  描述：为备份操作给集群加锁，这些备份是为在新增节点上做恢复。
  返回值类型：Boolean
  备注：调用该函数的用户需要具有SYSADMIN权限。

  说明

  pgxc_lock_for_backup是在使用gs_dump或gs_dumpall工具备份集群前，用来给集群加锁的。当给集群加锁后，不允许有改变系统结构的操作。该函数不影响DML语句。
- pg_pool_validate(clear bool, node_name text)

版权所有 © 华为云计算技术有限公司

描述：显示CN到节点node_name之间pooler中无效连接，当clear为true时清理无效连接。

返回值类型：record

- gs_validate_node_conn(validate_type cstring, validate_node_name cstring)

  描述：显示或清除当前节点接收来自指定节点的连接。

  返回值类型：record

  表 1-183 gs_validate_node_conn 函数参数说明

参数类型	参数名	类型	描述
输入参数	validate_type	cstring	指定入参类型： ● 'check'：用于显示当前节点接收来自指定节点的连接。 ● 'clear'：用于清除当前节点接收来自指定节点的连接。
输入参数	validate_node_name	cstring	指定节点名称。
输出参数	pid	bigint	显示当前节点接收来自指定节点连接的线程id（若是线程池线程则为0）。
输出参数	sessionid	bigint	显示当前节点接收来自指定节点连接的session id（若是非线程池线程则为0）。
输出参数	node_name	text	显示当前节点接收来自指定节点连接的节点名称。

- pgxc_pool_connection_status()

  描述：检查pooler连接状态是否正常。

  返回值类型：boolean

- pg_nodes_memory()

  描述：查看所有节点的内存占用。

  返回值类型：record

- table_skewness(text)

  描述：查看表数据在所有节点的占比。该函数不支持查询hashbucket表在所有节点的占比。

  参数：表示待查询表的表名，为text类型。

  返回值类型：record

- table_skewness(text, text, text)

  描述：查看表数据指定列在所有节点的占比。

  参数：表示待查询表的表名、指定列名、指定的表的记录数（默认值为0，查询所有记录），都为text类型。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

返回值说明：节点编号，指定列的数据行数，当前节点数据量相对总数据量的占比。

示例：

```
--返回't'表'a'字段前5行数据在节点上的分布。
gaussdb=# SELECT table_skewness('t', 'a',5);
 table_skewness
----------------
 (1,3,60.000%)
 (2,2,40.000%)
(2 rows)

--返回't'表'a'字段所有数据在节点上的分布。
gaussdb=# SELECT table_skewness('t', 'a');
 table_skewness
----------------
 (1,7,70.000%)
 (2,2,20.000%)
 (0,1,10.000%)
(3 rows)
```

- table_skewness_with_schema(text, text)

  描述：查看表数据在所有节点的占比，与table_skewness(text)作用相同。

  参数：表示待查询表的schema名称和表名，为text类型。

  返回值类型：record

- table_data_skewness(colrecord, type)

  描述：查看表数据所在节点。

  参数：

  - colrecord：表示待查询表的列名记录，为record类型。
  - type：hash分布类型，其取值请参见**PGXC_CLASS**系统表中pclocatortype字段的描述。

  返回值类型：smallint

  示例：

  ```
  --返回'test1'表的'index'字段数据所在的节点。
  gaussdb=# SELECT table_data_skewness(row(index), 'R') FROM test1;
   table_data_skewness
  ---------------------
                     4
                     3
                     1
                     2
  (4 rows)
  ```

- table_distribution(schemaname text, tablename text)

  描述：查看指定表在各个节点上占用的存储空间。

  参数：表示待查询表的模式名和表名，均为text类型。

  返回值类型：record

  说明

  - 使用本函数查询指定表存储分布信息，需要具备指定表的SELECT权限。
  - table_distribution性能比table_skewness更优，尤其是在大数据量场景下，请优先考虑使用table_distribution函数。
  - 当使用table_distribution并希望直观的看到空间占比时，可使用dnsize/(sum(dnsize) over ())的方式查看出具体的占比情况。

- table_distribution()

版权所有 © 华为云计算技术有限公司

描述：查看当前库中所有表在各节点的存储空间分布情况。

返回值类型：record

> **说明**
> - 使用本函数涉及全库表信息查询，需要具备管理员权限。
> - 当前基于table_distribution()函数，GaussDB提供PGXC_GET_TABLE_SKEWNESS视图（请参见**PGXC_GET_TABLE_SKEWNESS**）进行数据倾斜查询，建议在数据库中表数量（小于10000）较少的场景直接使用。

- plan_seed()

  描述：获取前一次查询语句的seed值（内部使用）。

  返回值类型：int

- pg_stat_get_env()

  描述：获取当前节点的环境变量信息，仅sysadmin和monitor admin可以访问。

  返回值类型：record

  示例：

```
gaussdb=# SELECT pg_stat_get_env();
                                                    pg_stat_get_env
----------------------------------------------------------------------------------------------------------------------------
 (coordinator1,localhost,144773,49100,/data1/GaussDB_Kernel_TRUNK/install,/data1/
GaussDB_Kernel_TRUNK/install/data/coordinator1,gs_log)
(1 row)
```

- pg_stat_get_thread()

  描述：提供当前节点下线程的状态信息，sysadmin和monitor admin用户可以查看所有线程的信息，普通用户只能查看本用户的线程信息。

  返回值类型：record

- pgxc_get_os_threads()

  描述：提供整个集群中所有正常节点下的线程状态信息。

  返回值类型：record

- pg_stat_get_sql_count()

  描述：提供当前节点中用户执行的SELECT/UPDATE/INSERT/DELETE/MERGE INTO语句的计数结果和DDL/DML/DCL语句的计数结果。DDL语句包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。DML语句包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。DCL语句包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。sysadmin和monitor admin用户可以查看所有用户的统计信息，普通用户可以查看本用户的统计信息。

  返回值类型：record

- pgxc_get_sql_count()

  描述：提供整个集群所有节点中所有用户执行的SELECT/UPDATE/INSERT/DELETE/MERGE INTO语句的计数结果和DDL/DML/DCL语句的计数结果，DDL语句包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。DML语句包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。DCL语句包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。

  返回值类型：record

- pgxc_get_node_env()

  描述：提供获取集群中所有节点的环境变量信息。

  返回值类型：record

- pgxc_disaster_read_set(text)

  描述：设置灾备集群的节点信息到ETCD上。仅灾备集群可用，仅初始用户可调用。

  返回值类型：Boolean

- pgxc_disaster_read_init()

  描述：初始化灾备可读的资源和状态信息。仅灾备集群可用，仅初始用户可调用。

  返回值类型：Boolean

- pgxc_disaster_read_clear()

  描述：清理灾备可读的资源和状态信息。仅灾备集群可用，仅初始用户可调用。

  返回值类型：Boolean

- pgxc_disaster_read_status()

  描述：提供灾备集群的节点信息，仅灾备集群可用。

  返回值类型：record

- disaster_read_set_cn_status()

  描述：更新灾备读中CN活跃状态信息。仅灾备集群可用。

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT
g_catalog.disaster_read_set_cn_status('440000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000');
disaster_read_set_cn_status
-----------------------------
 t
(1 row)
```

- disaster_read_del_cn_status()

  描述：清理灾备读中CN活跃状态信息。仅灾备集群可用。

  返回值类型：Boolean

  示例：

```
gaussdb=# SELECT pg_catalog.disaster_read_del_cn_status();
disaster_read_del_cn_status
-----------------------------
 t
(1 row)
```

- gs_switch_relfilenode()

  描述：交换两个表或分区的元信息（重分布工具内部使用，用户直接使用会有错误信息提示）。

  返回值类型：int

  说明

  此函数可能导致清空统计信息，建议在调用后重新收集统计信息。

- pg_catalog.plancache_clean()

  描述：清理当前节点上无人使用的全局计划缓存。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：boolean

- DBE_PERF.global_plancache_clean()

  描述：清理所有节点上无人使用的全局计划缓存。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：Boolean

- pg_catalog.global_plancache_set_invalid(hashkey, unique_sql_id)

  描述：支持通过bigint类型的入参hashkey和unique_sql_id手动失效本节点的GPC缓存计划。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：text

  示例：

```
-- 从dbe_perf.global_plancache_status视图中获取hash_value作为第一个入参，unique_sql_id作为第二个入参。
gaussdb=# SELECT * FROM pg_catalog.global_plancache_set_invalid(3928396421,3641111755);
 global_plancache_set_invalid
------------------------------
 Execution succeeded.
(1 row)
```

  说明

  - 不建议用户使用此函数，会导致硬解析次数变多，影响性能。
  - 函数不支持使unique_sql_id为0的计划失效。

- pg_catalog.global_plancache_show_plan(hashkey, unique_sql_id)

描述：通过bigint类型的入参hashkey和unique_sql_id，查看对应的GPC缓存计划。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。此函数不支持explain_perf_mode pretty模式。

返回值类型：text

示例：

```
-- 从dbe_perf.global_plancache_status视图中获取hash_value作为第一个入参，unique_sql_id作为第二个入参。
gaussdb=# SELECT * FROM pg_catalog.global_plancache_show_plan(3928396421,3641111755);
       global_plancache_show_plan
-----------------------------------------
 Result  (cost=0.00..0.01 rows=1 width=0)+

(1 row)
```

说明

函数不支持显示unique_sql_id为0的计划。

- copy_error_log_create()

  描述：由于COPY FROM容错机制所需要的错误表已变更为在初始化阶段创建的系统表，因此当前函数改为始终返回true，仅做兼容。

  返回值类型：Boolean

表 1-184 错误表 pg_catalog.pgxc_copy_error_log 信息

列名称	类型	描述
relname	character varying	表名称。以模式名.表名形式显示。
begintime	timestamp with time zone	出现数据格式错误的时间。
filename	character varying	出现数据格式错误的数据源文件名。
lineno	bigint	在数据源文件中，出现数据格式错误的行号。
rawrecord	text	在数据源文件中，出现数据格式错误的原始记录。
detail	text	详细错误信息。
custom_id	text	关联的copy_custom_id输入的值，表示本次导入的自定义标识信息。

- pgxc_copy_error_func(boolean need_union, boolean only_dn)

  描述：查询各节点上pg_catalog.gs_copy_error_log表中的内容并返回。返回值record中各字段的名称及数据类型与pg_catalog.gs_copy_error_log表中各字段名称及数据类型对应。

  返回值类型：set of record

版权所有 © 华为云计算技术有限公司

表 1-185 pgxc_copy_error_func 参数说明

参数	类型	描述
need_union	boolean	• true表示同时查询pg_catalog.gs_copy_error_log表和pg_catalog.gs_copy_error_old表中的数据并做union_all操作； • false表示仅查询pg_catalog.gs_copy_error_log表中的数据。
only_dn	boolean	• true表示只查询所有DN节点上的数据； • false表示查询所有节点上的数据。

- gs_copy_error_insert(cstring relname, cstring begintime, cstring filename, bigint lineno, cstring rawrecord, cstring detail, cstring custom_id)

  描述：向DN节点的pg_catalog.gs_copy_error_log表中插入数据。该函数仅供COPY内部调用，禁止用户主动调用。

  返回值类型：boolean

  表 1-186 gs_copy_error_insert 参数说明

参数	类型	描述
relname	cstring	表名称。以模式名.表名形式显示。
begintime	cstring	出现数据格式错误的时间。
filename	cstring	出现数据格式错误的数据源文件名。
lineno	bigint	在数据源文件中，出现数据格式错误的行号。
rawrecord	cstring	在数据源文件中，出现数据格式错误的原始记录。
detail	cstring	详细错误信息。
custom_id	cstring	关联的copy_custom_id输入的值，表示本次导入的自定义标识信息。

- gs_copy_summary_func(boolean need_union)

  描述：查询当前节点上pg_catalog.gs_copy_summary_log表中的内容并返回。目前不支持此函数。

  返回值类型：set of record

版权所有 © 华为云计算技术有限公司

表 1-187 gs_copy_summary_func 参数说明

参数	类型	描述
need_union	boolean	• true表示同时查询pg_catalog.gs_copy_summary_log表和pg_catalog.gs_copy_summary_old表中的数据并做union_all操作； • false表示仅查询pg_catalog.gs_copy_summary_log表中的数据。

- pg_stat_get_data_senders()

  描述：提供当前活跃的数据复制发送线程的详细信息。

  返回值类型：record

- textlen()

  描述：提供查询text的逻辑长度的方法。

  返回值类型：int

- threadpool_status()

  描述：显示线程池中工作线程及会话的状态信息。

  返回值类型：record

- get_local_active_session()

  描述：提供当前节点保存在内存中的历史活跃session状态的采样记录，sysadmin和monitor admin权限能查看当前节点所有的历史活跃session记录，普通用户查看本会话的历史活跃session记录。

  返回值类型：record

- dbe_perf.get_global_gs_asp(timestamp,timestamp)

  描述：提供所有节点保存在系统表gs_asp中的历史活跃session状态的采样记录。505.0版本新增event_start_time、current_xid、top_xid三个字段信息，由低于505.0版本升级至505.0或更高版本时，升级观察期无法查询。

  返回值类型：record

- get_wait_event_info()

  描述：提供wait event事件的具体信息。

  返回值类型：record

- dbe_perf.get_datanode_active_session(text)

  描述：提供从CN查询主DN上保存在内存中的历史活跃session状态的采样记录。505.0版本新增event_start_time、current_xid、top_xid三个字段信息，由低于505.0版本升级至505.0或更高版本时，升级观察期无法查询。

  返回值类型：record

  备注：该函数查询目标DN上DBE_PERF.LOCAL_ACTIVE_SESSION视图（请参见**DBE_PERF.LOCAL_ACTIVE_SESSION**）中的记录并和所有CN上的DBE_PERF.LOCAL_ACTIVE_SESSION视图中的记录进行匹配获取query string，所以会占用大量的内存。

- dbe_perf.get_datanode_active_session_hist(text,timestamp,timestamp)

版权所有 © 华为云计算技术有限公司

描述：提供从CN查询主DN上保存在系统表gs_asp中的历史活跃session状态的采样记录。505.0版本新增event_start_time、current_xid、top_xid三个字段信息，由低于505.0版本升级至505.0或更高版本时，升级观察期无法查询。

返回值类型：record

备注：该函数查询目标DN上指定时间段的gs_asp记录，如果指定时间段过长造成查询的记录过多，会耗费大量时间。

- generate_wdr_report(begin_snap_id bigint, end_snap_id bigint, report_type cstring, report_scope cstring, node_name cstring)

  描述：基于两个snapshot生成系统诊断报告，默认初始化用户或监控管理员用户可以访问。只可在系统库中查询到结果，用户库中无法查询。

  返回值类型：text

  表 1-188 generate_wdr_report 参数说明

参数	说明	取值范围
begin_snap_id	生成某段时间内性能诊断报告的开始snapshotid。	1~9,223,372,036,854,775,807
end_snap_id	结束snapshot的id，默认end_snap_id大于begin_snap_id。	1~9,223,372,036,854,775,807
report_type	指定生成report的类型。	• summary • detail • all，即同时包含summary和detail。
report_scope	指定生成report的范围。	• cluster：数据库级别的信息。 • node：节点级别的信息。
node_name	• 在report_scope指定为node时，需要把该参数指定为对应节点的名称，节点名称可以通过执行“SELECT * FROM pg_node_env;”查询。 • 在report_scope指定为cluster时，该值可以省略，或者指定为空或NULL。	• cluster：省略/空/NULL。 • node：GaussDB中的节点名称。

- generate_wdr_db_report(db_name cstring, begin_snap_id bigint, end_snap_id bigint, report_type cstring, report_scope cstring, node_name cstring)

  描述：基于两个snapshot生成库级系统诊断报告。需要在系统库下执行，默认初始化用户或monadmin用户可以访问。用户只可在系统库中查询到结果，用户库中无法查询。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

表 1-189 generate_wdr_db_report 参数说明

参数	说明	取值范围
db_name	指定database名称。	-
begin_snap_id	生成某段时间内性能诊断报告的开始snapshotid。	1~9,223,372,036,854,775,807
end_snap_id	结束snapshot的id，默认end_snap_id大于begin_snap_id。	1~9,223,372,036,854,775,807
report_type	指定生成report的类型。	• summary • detail • all，即同时包含summary和detail。
report_scope	指定生成report的范围。	• cluster：数据库级别的信息。 • node：节点级别的信息。
node_name	• 在report_scope指定为node时，需要把该参数指定为对应节点的名称，节点名称可以通过执行“SELECT * FROM pg_node_env;”查询。 • 在report_scope指定为cluster时，该值可以省略，或者指定为空或NULL。	• cluster：省略/空/NULL。 • node：GaussDB中的节点名称。

- create_wdr_snapshot()

  描述：手工生成系统诊断快照，该函数需要sysadmin权限，且只能在CCN上执行。

  返回值类型：text

- kill_snapshot()

  描述：kill后台的WDR snapshot线程，调用该函数的用户需要具有SYSADMIN权限或具有REPLICATION权限或继承了内置角色gs_role_replication的权限。

  返回值类型：void

- dbe_perf.generate_asp_report(start_time timestamp with time zone, end_time timestamp with time zone, datanode text, slot_count bigint)

  描述：基于时间戳和节点信息生成ASP诊断报告，monadmin用户可以访问。slot_count为指标展示时分割的时间段的个数。datanode可以指定某个分区DN（如：dn_6001_6002_6003），或者是某个DN节点（dn_6001、dn_6002、dn_6003）。该函数只能在CN上使用。

  返回值类型：text

- generate_asp_report(start_time timestamp with time zone, end_time timestamp with time zone, slot_count bigint, node_name text)

  描述：基于时间戳生成ASP诊断报告，monadmin用户可以访问。slot_count为指标展示时分割的时间段的个数。node_name用于生成指定节点的ASP报告，取值

版权所有 © 华为云计算技术有限公司

只能是当前CN的名称，可缺省。缺省时，生成当前节点的报告。该函数只能在CN上使用。

返回值类型：text

- dbe_perf.get_active_session_profile(start_ts timestamp with time zone, end_ts timestamp with time zone, need_final boolean,node_name text)

  描述：基于时间戳查询asp内存中与磁盘中的数据。need_final是否查询阻塞信息，默认值为false。node_name用于区分CN、主备DN，如（cn_5001、dn_6001或dn_6002），不可缺省。505.0版本新增event_start_time、current_xid、top_xid三个字段信息，由低于505.0版本升级至505.0或更高版本时，升级观察期无法查询。

  返回值类型：record

- capture_view_to_json(text,integer)

  描述：将视图的结果存入GUC: perf_directory所指定的目录，如果is_crossdb为1，则表示对于所有的database都会访问一次view；如果is_crossdb为0，则表示仅对当前database进行一次视图访问。该函数只有sysadmin和monitor admin用户可以执行。

  返回值类型：int

- reset_unique_sql(text,text,bigint)

  描述：用来清理CN/DN内存中的Unique SQL（需要sysadmin/monitor admin权限）。

  返回值类型：Boolean

  表 1-190 reset_unique_sql 参数说明

参数	类型	描述
scope	text	清理范围类型： ● 'GLOBAL'：清理所有的CN/DN节点，如果是'GLOBAL'，则只可以为CN节点执行此函数。 ● 'LOCAL'：清理本节点。
clean_type	text	● 'BY_USERID'：按用户ID来进行清理Unique SQL。 ● 'BY_CNID'：按CN的ID来进行清理Unique SQL。 ● 'ALL'：全部清理。
clean_value	int8	clean_type对应的清理值。 ● 如果clean_type为'BY_USERID'，则清理用户ID为clean_value的记录。 ● 如果clean_type为'BY_CNID'，则清理CN的ID为clean_value的记录。 ● 如果第二个参数为ALL，则第三个参数无效，可为任意值。

- wdr_xdb_query(db_name_str text, query text)

版权所有 © 华为云计算技术有限公司

描述：提供本地跨数据库执行query的能力。例如: 在连接到testdb库时，访问test库下的表。只有初始化用户才有权限执行。

```
SELECT col1 from wdr_xdb_query('dbname=test','select col1 from t1') as dd(col1 int);
```

返回值类型：record

- pg_wlm_jump_queue(pid int)

  描述：调整任务到CN队列的最前端。

  返回值类型：boolean

  - true：成功。
  - false：失败。

- gs_wlm_switch_cgroup(pid int, cgroup text)

  描述：调整作业的优先级到新控制组。

  返回值类型：boolean

  - true：成功。
  - false：失败。

- pv_session_memctx_detail(threadid tid, MemoryContextName text)

  描述：将线程tid的MemoryContextName内存上下文信息记录到“$GAUSSLOG/gs_log/${node_name}/dumpmem”目录下的“threadid_timestamp.log”文件中。其中threadid可通过查询PV_SESSION_MEMORY_DETAIL视图（请参见**PV_SESSION_MEMORY_DETAIL**）中的sessid字段获得。在正式发布的版本中仅接受MemoryContextName为空串（两个单引号表示输入为空串，即''）的输入，此时会记录所有的内存上下文信息，否则不会有任何操作。该函数需要管理员权限的用户才能执行。

  返回值类型：boolean

  - true：成功。
  - false：失败。

- pg_shared_memctx_detail(MemoryContextName text)

  描述：将MemoryContextName内存上下文信息记录到“$GAUSSLOG/gs_log/${node_name}/dumpmem”目录下的“threadid_timestamp.log”文件中。在正式发布版本中调用该函数不会有任何操作。该函数需要管理员权限的用户才能执行。

  返回值类型：boolean

  - true：成功。
  - false：失败。

- local_aio_completer_stat()

  描述：显示本实例中AIO Completer线程的相关统计信息。

  返回值类型：record

表 1-191 local_aio_completer_stat 返回值说明

参数	类型	描述
node_name	text	当前实例名称。
tid	int8	AIO Completer线程id。
thread_type	text	AIO Completer线程类型（读/写）。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
aio_submitted_num	int8	AIO Completer线程已提交的异步I/O请求数量。
aio_completed_num	int8	AIO Completer线程已完成的异步I/O请求数量。
aio_incompleted_num	int8	AIO Completer线程未完成的异步I/O请求数量。
slot_count_left	int8	空闲slot数量。

- local_aio_slot_usage_status()
  描述：显示本实例中异步I/O提交槽位的相关统计信息。
  返回值类型：record

  表 1-192 local_aio_slot_usage_status 返回值说明

参数	类型	描述
node_name	text	当前实例名称。
slot_id	int4	槽位id。
slot_type	char	槽位类型（r：读，w：写）。
status	bool	槽位占用状态。
buffer_id	int8	该槽位对应的buffer id。
relfilenode_blocknum	text	该槽位对应的buffer所在的物理页面位置。
lsn	int8	页面对应的LSN。
submitted_time	int8	页面异步提交时的时间。
elapsed_time	int8	页面已经等待的时间。

- remote_aio_completer_stat()
  描述：显示其他实例中AIO Completer线程的相关统计信息（仅在CN上使用）。
  返回值类型：record

  表 1-193 remote_aio_completer_stat 返回值说明

参数	类型	描述
node_name	text	其他实例名称。
tid	int8	AIO Completer线程id。
thread_type	text	AIO Completer线程类型（读/写）。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
aio_submitted_num	int8	AIO Completer线程已提交的异步I/O请求数量。
aio_completed_num	int8	AIO Completer线程已完成的异步I/O请求数量。
aio_incompleted_num	int8	AIO Completer线程未完成的异步I/O请求数量。
slot_count_left	int8	空闲slot数量。

- remote_aio_slot_usage_status()
  描述：显示其他实例中异步I/O提交槽位的相关统计信息。
  返回值类型：record

  表 1-194 remote_aio_slot_usage_status 返回值说明

参数	类型	描述
node_name	text	其他实例名称。
slot_id	int4	槽位id。
slot_type	char	槽位类型（r：读，w：写）。
status	bool	槽位占用状态。
buffer_id	int8	该槽位对应的buffer id。
relfilenode_blocknum	text	该槽位对应的buffer所在的物理页面位置。
lsn	int8	页面对应的LSN。
submitted_time	int8	页面异步提交时的时间。
elapsed_time	int8	页面已经等待的时间。

- gs_get_io_type()
  描述：显示本实例的I/O模式。
  返回值类型：text
  - BIO：表示当前实例正以BIO模式运行（即ADIO未打开）。
  - DIO：表示当前实例正以DIO模式运行（即ADIO已打开）。
  - BIO->DIO (In progress)：表示当前实例正在从BIO到DIO模式的切换过程中。
- local_bgwriter_stat()
  描述：显示本实例的bgwriter线程刷页信息。
  返回值类型：record

版权所有 © 华为云计算技术有限公司

表 1-195 local_bgwriter_stat 返回值说明

参数	类型	描述
node_name	text	当前实例名称。
bgwr_actual_flush_total_num	int8	自从上次重启到现在，bgwriter线程刷脏总数（单位：页）。
bgwr_last_flush_num	int4	自从上一轮刷脏到现在，bgwriter线程刷脏数量（单位：页）。
candidate_slots	int4	候选者队列中可用于缓存页面申请的数量。
get_buffer_from_list	int8	自从上次重启到现在，从候选者队列中获取缓存页面的数量（单位：页）。
get_buf_clock_sweep	int8	自从上次重启到现在，从clock sweep机制中获取缓存页面的数量（单位：页）。

- local_candidate_stat()

  描述：显示本实例的候选buffer链中页面个数，buffer淘汰信息，包含normal buffer pool和segment buffer pool。

  返回值类型：record

- gs_buffer_manager_info()

  描述：打印当前数据库从上一次调用该函数（如未执行过则为启动时）到现在的缓冲区的页面访问数和未命中数量，以及读取率（即1-命中率），以及智能淘汰算法给出淘汰队列的每个轮次的休眠时间。

  Total access: %lu, Total read: %lu, Read rate: %0.3f, Sleep time :%u

  返回值类型：text

- local_ckpt_stat()

  描述：显示本实例的检查点信息和各类日志刷页情况。

  返回值类型：record

- local_double_write_stat()

  描述：显示本实例的双写文件的情况。

  返回值类型：record

表 1-196 local_double_write_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
curr_dwn	int8	当前双写文件的序列号。
curr_start_page	int8	当前双写文件恢复起始页面。
file_trunc_num	int8	当前双写文件复用的次数。
file_reset_num	int8	当前双写文件写满后发生重置的次数。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
total_writes	int8	当前双写文件总的I/O次数。
low_threshold_writes	int8	低效率写双写文件的I/O次数（一次I/O刷页数量少于16页面）。
high_threshold_writes	int8	高效率写双写文件的I/O次数（一次I/O刷页数量多于一批，421个页面）。
total_pages	int8	当前刷页到双写文件区的总的页面个数。
low_threshold_pages	int8	低效率刷页的页面个数。
high_threshold_pages	int8	高效率刷页的页面个数。
file_id	int8	当前双写文件的id号。

- local_pagewriter_running_stat()
  描述：显示本实例批量刷脏线程的线程运行状态情况。
  返回值类型：record

  表 1-197 local_pagewriter_running_stat()参数说明

参数类型	参数	类型	描述
输出参数	nodename	text	实例名称。
输出参数	tid	int8	pagewriter线程id。
输出参数	type	text	线程类型。
输出参数	extre_start_loc	int8	极致刷脏推进的位置。
输出参数	extre_left_size	int4	极致刷脏上一轮跳过待刷的脏页数量。
输出参数	full_ckpt	bool	是否处于全量刷脏模式。
输出参数	shutdown_progress	bool	退出状态。
输出参数	buf_id_start	int4	扫描线程范围起始buf_id。
输出参数	buf_id_end	int4	扫描线程结束buf_id。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	standby_buf_start_id	int4	备机读扫描线程起始buf_id。
输出参数	standby_buf_end_id	int4	备机读扫描线程结束buf_id。
输出参数	exception_count	int4	线程历史累计异常数量。
输出参数	recent_exception_time	timestamptz	最近一次异常的时间戳。

- local_pagewriter_accumulate_stat()
  描述：显示本实例在输入时间范围内的批量刷脏线程的带宽和锁冲突统计情况。
  返回值类型：record

  表 1-198 local_pagewriter_accumulate_stat()参数说明

参数类型	参数	类型	描述
输入参数	last_time	int8	统计的时间长度(秒)。
输出参数	nodename	text	实例名称。
输出参数	tid	int8	pagewriter线程id。
输出参数	type	text	线程类型。
输出参数	bandwidth	float8	平均带宽(MB/s)。
输出参数	lock_conflict_count	int8	因为内容锁冲突跳过的页面数量。
输出参数	avg_pagewriter_sleep	float8	真实的平均刷脏周期。

- local_dirty_queue_advance_stat()
  描述：显示本实例脏页队列的推进情况。
  返回值类型：record

版权所有 © 华为云计算技术有限公司

表 1-199 local_dirty_queue_advance_stat()参数说明

参数类型	参数	类型	描述
输出参数	nodename	text	实例名称。
输出参数	dirty_queue_head	int8	脏页队列头的位置。
输出参数	dirty_queue_tail	int8	脏页队列尾的位置。
输出参数	dirty_queue_size	int8	脏页队列大小。
输出参数	dirty_queue_full	bool	脏页队列是否满。
输出参数	head_buffer	int4	脏页队列头的buffer值。
输出参数	head_buffer_lsn	text	脏页队列头的页面lsn。
输出参数	head_buffer_content_lock_id	int8	脏页队列头buffer的内容锁的lock_unique_id。
输出参数	head_buffer_io_lock_id	int8	脏页队列头buffer的I/O锁的lock_unique_id。
输出参数	extre_scan_current	int8	极致刷脏已经分配给刷脏线程的脏页队列的结束位置。

- local_single_flush_dw_stat()
  描述：显示本实例的单页面淘汰双写文件的情况。
  返回值类型：record
- local_pagewriter_stat()
  描述：显示本实例的刷页信息和检查点信息。
  返回值类型：record

表 1-200 local_pagewriter_stat()参数说明

参数类型	参数	类型	描述
输出参数	node_name	text	实例名称。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	pgwr_actual_flush_total_num	int8	从启动到当前时间刷脏页的总计数量。
输出参数	pgwr_last_flush_num	int4	上一批刷脏页数量。
输出参数	remain_dirty_page_num	int8	当前预计剩余的脏页数量。
输出参数	queue_head_page_rec_lsn	text	当前实例的脏页队列第一个脏页的recovery_lsn。
输出参数	queue_rec_lsn	text	当前实例的脏页队列的recovery_lsn。
输出参数	current_xlog_insert_lsn	text	当前实例Xlog写入的位置，备机查询时该值无效。
输出参数	ckpt_redo_point	text	当前实例的检查点。

- local_redo_stat()
  描述：显示本实例的备机的当前回放状态。
  返回值类型：record
  备注：返回的回放状态主要包括当前回放位置，回放最小恢复点位置等信息。
- local_recovery_status()
  描述：显示本实例的主机和备机的日志流控信息。
  返回值类型：record
- local_rto_status()
  描述：显示本实例的主机和备机的日志流控信息。
  返回值类型：record
- gs_wlm_switch_cgroup(sess_id int8, cgroup name)
  描述：切换指定会话的控制组。
  返回值类型：record
- comm_client_info()
  描述：用于查询单个节点活跃的客户端连接信息，返回结果解释见**COMM_CLIENT_INFO**。
  返回值类型：setof record
- pg_get_flush_lsn()
  描述：返回当前节点flush的Xlog位置。
  返回值类型：text
- pg_get_sync_flush_lsn()
  描述：返回当前节点多数派flush的Xlog位置。

版权所有 © 华为云计算技术有限公司

返回值类型：text

- pgxc_wlm_rebuild_user_resource_pool()

  描述：重新构建用户及资源池缓存信息。需要系统管理员权限才可以执行该函数。

  返回值类型：boolean

- locktag_decode(locktag text)

  描述：从locktag中解析锁的具体信息。

  示例：

```
gaussdb=# SELECT locktag_decode('271b:0:0:0:0:6');
              locktag_decode
-------------------------------------------
 locktype:transactionid, transactionid:10011
(1 row)
```

  返回值类型：text

- disable_conn(disconn_mode text, host text, port integer)

  描述：CM Agent处理CM Server下发的命令，在DN进行选主时设置该DN拒绝连接所有DN、强制连接某个DN或轮询连接所有DN。只有初始化用户和系统管理员才可以调用该函数。

  返回值类型：void

  表 1-201 disable_conn 参数说明

参数	类型	描述
disconn_mode	text	DN连接模式： ● 'prohibit_connection' - 拒绝连接所有DN。 ● 'specify_connection' - 强制连接某个DN。 ● 'polling_connection' - 轮询连接所有DN。
host	text	DN的IP。
port	integer	DN的端口号。

- dbe_perf.get_global_full_sql_by_timestamp(start_timestamp timestamp with time zone, end_timestamp timestamp with time zone)

  描述：获取集群级的全量SQL(Full SQL)信息。只可在系统库中查询到结果，用户库中无法查询。

  返回值类型：record

  表 1-202 dbe_perf.get_global_full_sql_by_timestamp 参数说明

参数	类型	描述
start_timestamp	timestamp with time zone	SQL启动时间范围的开始时间点。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
end_timestamp	timestamp with time zone	SQL启动时间范围的结束时间点。若start_timestamp>=end_timestamp，则函数执行报错。

- dbe_perf.get_global_slow_sql_by_timestamp(start_timestamp timestamp with time zone, end_timestamp timestamp with time zone)

  描述：获取集群级的慢SQL(Slow SQL)信息。只可在系统库中查询到结果，用户库中无法查询。

  返回值类型：record

  表 1-203 dbe_perf.get_global_slow_sql_by_timestamp 参数说明

参数	类型	描述
start_timestamp	timestamp with time zone	SQL启动时间范围的开始时间点。
end_timestamp	timestamp with time zone	SQL启动时间范围的结束时间点。若start_timestamp>=end_timestamp，则函数执行报错。

- statement_detail_decode(detail text, format text, pretty boolean)

  描述：解析全量/慢SQL语句中的details字段的信息。只可在系统库中查询到结果，用户库中无法查询。

  返回值类型：text

  表 1-204 statement_detail_decode 参数说明

参数	类型	描述
detail	text	SQL语句产生的事件的集合（不可读）。
format	text	解析输出格式，取值为plaintext。
pretty	boolean	当format为plaintext时，是否以优雅的格式展示： ● true表示通过“\n”分隔事件。 ● false表示通过“,”分隔事件。

- pgxc_get_csn(tid)

  描述：返回普通表的事务id对应的事务提交序号（CSN）。

  返回值类型：int8

- pgxc_get_csn(tid，bucketid)

  描述：返回给定bucketid的事务id对应的事务提交序号（CSN），入参bucketid指hashbucket表的物理bucketid，普通表为-1。

  返回值类型：int8

版权所有 © 华为云计算技术有限公司

- get_global_user_transaction()

  描述：返回所有节点上各用户的事务相关信息。

  返回值类型：node_name name、usename name、commit_counter bigint、rollback_counter bigint、resp_min bigint、resp_max bigint、resp_avg bigint、resp_total bigint、bg_commit_counter bigint、bg_rollback_counter bigint、bg_resp_min bigint、bg_resp_max bigint、bg_resp_avg bigint、bg_resp_total bigint、dbid oid

- pg_collation_for()

  描述：返回入参字符串对应的排序规则。

  参数：any（如果是常量必须进行显式类型转换）

  返回值类型：text

- pgxc_unlock_for_sp_database(databasename text)

  描述：释放指定数据库锁。

  参数：databasename为数据库名称。

  返回值类型：Boolean

  备注：调用该函数的用户需要具有SYSADMIN权限。

- pgxc_lock_for_sp_database(databasename text)

  描述：对指定的数据库加锁。

  参数：databasename为数据库名称。

  返回值类型：Boolean

  备注：调用该函数的用户需要具有SYSADMIN权限。

- pgxc_unlock_for_transfer(schemaname text)

  描述：释放用于数据传输（数据重分布）锁。

  参数：schemaname为模式名称。

  返回值类型：Boolean

  备注：调用该函数的用户需要具有SYSADMIN权限。

- pgxc_lock_for_transfer(schemaname text)

  描述：对数据库加锁，用于数据传输（数据重分布）。

  参数：schemaname为模式名称。

  返回值类型：Boolean

  备注：调用该函数的用户需要具有SYSADMIN权限。

- gs_comm_proxy_thread_status()

  描述：用于在集群配置用户态网络的场景下，代理通信库comm_proxy收发数据包统计。该函数在当前版本不支持。

  参数：nan

  返回值类型：record

- gs_catalog_attribute_records()

  描述：对于指定的系统表oid，返回该系统表对应的各个字段的定义。仅支持oid小于10000的普通系统表（不支持索引、toast表等）。

  参数：系统表oid。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- dynamic_func_control(scope text, function_name text, action text, "{params}" text[])

  描述：动态开启内置的功能，当前仅支持动态开启全量SQL。

  返回值类型：record

  表 1-205 dynamic_func_control 参数说明

参数	类型	描述
scope	text	动态开启功能的范围，当前仅支持'GLOBAL'和'LOCAL'。
function_name	text	功能的名称，当前仅支持'STMT'和'GSTRACE'。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
action	text	当function_name为'STMT'时，action仅支持TRACK/UNTRACK/LIST/CLEAN（所有用户均可以使用LIST选项，其他选项必须具有系统管理员或MONADMIN权限才能使用）： ● TRACK - 开始记录归一化SQL的全量SQL信息。 ● UNTRACK - 取消记录归一化SQL的全量SQL信息。 ● LIST - 列取当前TRACK的归一化SQL的信息。 ● CLEAN - 清理记录当前归一化SQL的信息。 当function_name为'GSTRACE'时，action仅支持TRACE_SQL/TRACE_FUNC/UNTRACE_SQL/UNTRACE_FUNC/UNTRACE_ALL_SQL/UNTRACE_ALL_FUNC/LIST_TRACE_SQL/LIST_TRACE_FUNC/SHOW_TRACED_FUNC/SHOW_MANGLED_FUNC/SHOW_SUBFUNC： ● TRACE_SQL - GSTrace注册指定跟踪的SQL语句。 ● TRACE_FUNC - GSTrace注册指定跟踪的函数。 ● UNTRACE_SQL - GSTrace取消注册指定跟踪的SQL语句。 ● UNTRACE_FUNC - GSTrace取消注册指定跟踪的函数。 ● UNTRACE_ALL_SQL - GSTrace取消全部SQL语句。 ● UNTRACE_ALL_FUNC - GSTrace取消全部函数。 ● LIST_TRACE_SQL - GSTrace查看配置的跟踪SQL语句。 ● LIST_TRACE_FUNC - GSTrace查看配置的跟踪函数。 ● SHOW_TRACED_FUNC - GSTrace查看正在跟踪的函数的状态。 ● SHOW_MANGLED_FUNC - GSTrace查看指定函数mangle后的名称。 ● SHOW_SUBFUNC - GSTrace指定函数的全部子函数。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
params	text[]	当function_name为'STMT'时，对应不同action的params设置如下： ● TRACK - '{"归一化SQLID", "L0/L1/L2"}' ● UNTRACK - '{"归一化SQLID"}' ● LIST - '{}' ● CLEAN - '{}' 当function_name为'GSTRACE'时，对应不同action的params设置如下： ● TRACE_SQL - 请参见gstrace_reg_sql()函数入参说明。 ● TRACE_FUNC - 请参见gstrace_reg_func()函数入参说明。 ● UNTRACE_SQL - 请参见gstrace_unreg_sql()函数入参说明。 ● UNTRACE_FUNC - 请参见gstrace_unreg_func()函数入参说明。 ● UNTRACE_ALL_SQL - 请参见gstrace_unreg_all_sql()函数入参说明。 ● UNTRACE_ALL_FUNC - 请参见gstrace_unreg_all_func()函数入参说明。 ● LIST_TRACE_SQL - 请参见gstrace_list_configured_sql()函数入参说明。 ● LIST_TRACE_FUNC - 请参见gstrace_list_configured_func()函数入参说明。 ● SHOW_TRACED_FUNC - 请参见gstrace_show_traced_func()函数入参说明。 ● SHOW_MANGLED_FUNC - 请参见gstrace_show_mangled_func()函数入参说明。 ● SHOW_SUBFUNC - 请参见gstrace_show_subfunc()函数入参说明。

- gs_parse_page_bypath(path text, blocknum bigint, relation_type text, read_memory boolean)

  描述：用于解析指定表页面，并返回存放解析内容的路径。

  返回值类型：text

  备注：必须是系统管理员或运维管理员才能执行此函数。对于未落盘的页面，由于磁盘上实际为空页，解析结果自然为空，且relation_type参数不生效。

版权所有 © 华为云计算技术有限公司

表 1-206 gs_parse_page_bypath 参数说明

参数	类型	描述
path	text	• 对于普通表，相对路径为：tablespace name/database oid/表的relfilenode(物理文件名)；例如：base/16603/16394。 • 对于普通表的visibility map，相对路径为：tablespace name/database oid/表的vm文件。例如：base/16603/16394_vm。 • 对于clog文件，相对路径为：pg_clog目录下的clog文件。例如：000000000000。 • 对于csnlog文件，相对路径为：pg_csnlog目录下的csnlog文件。例如：000000000000。 • 对于undo record文件，相对路径为：undo目录下的undo/UNDOPERSISTENCE/zoneid.segno。例如：undo/permanent/00000.0000009。 • 对于undo meta文件，相对路径为：undo目录下的undo/UNDOPERSISTENCE/zoneid.meta.segno。例如：undo/permanent/00000.meta.0000004。 • 表文件的相对路径可以通过pg_relation_filepath(table_name text)查找。分区表的路径可以查看pg_partition系统表和调用pg_partition_filepath(partition_oid)。 • 合法的path格式列举： – global/relNode。 – base/dbNode/relNode。 – pg_tblspc/spcNode/version_dir/dbNode/relNode。
blocknum	bigint	• -1：所有block的信息（强制从磁盘解析）。 • 0~MaxBlockNumber：对应block的信息。 • 对于BTree/UBTree索引，block 0为索引元页面。 • 对于undo record文件，采用逻辑块号，对应文件block大于等于segno*128，小于（segno+1）*128。 • 对于undo meta文件，采用逻辑块号，对应文件block大于等于segno*4，小于（segno+1）*4。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
relation_type	text	• heap（astore表）。 • uheap（ustore表）。 • btree（BTree索引）。 • ubtree（UBTree索引）。 • vm（astore普通表的visibility map）。 • fsm（行存表的freespace map文件）。 • clog（事务状态日志commit log）。 • csnlog（快照时间戳日志commit sequence number log）。 • undo_slot（事务槽信息）。 • undo_record（undo 记录信息）。 • indexurq（indexurq页面）。 • ugin_index (ugin索引) • gin_index (gin索引) • gist_index (gist索引) • cmap (跨页透明压缩元信息) • pg_dw_turbo_compress (跨页透明压缩双写区)
read_memory	boolean	• false，从磁盘文件解析。 • true，首先尝试从共享缓冲区中解析该页面；如果共享缓冲区中不存在，则从磁盘文件解析。

正常使用示例：

```
# 解析BTree索引文件中所有页面的信息内容
# 函数调用前请依据参数说明中的内容确保文件路径真实存在
gaussdb=# SELECT gs_parse_page_bypath('base/16603/16394', -1, 'btree', false);
                        gs_parse_page_bypath
---------------------------------------------------------------------
 gs_log_dir/dump/1663_16603_16394_-1.page
(1 row)

# 解析vm文件中所有block的可见性结果
gaussdb=# SELECT gs_parse_page_bypath('base/12828/16771_vm', -1, 'vm', false);
                         gs_parse_page_bypath
------------------------------------------------------------------------
 gs_log_dir/dump/1663_12828_16771_-1_vm.page
(1 row)

# 解析clog文件中0号block的commit log日志
gaussdb=# SELECT gs_parse_page_bypath('000000000000', 0, 'clog', false);
                        gs_parse_page_bypath
---------------------------------------------------------------------
 gs_log_dir/dump/000000000000.clog
(1 row)
```

异常场景报错示例：

```
# 入参blocknum超出取值范围时报错
gaussdb=# SELECT gs_parse_page_bypath('base/12828/16777', -10, 'heap', false);
```


版权所有 © 华为云计算技术有限公司

```
ERROR:  Blocknum should be between -1 and 4294967294.
CONTEXT:  referenced column: gs_parse_page_bypath
```

- gs_xlogdump_lsn(start_lsn text, end_lsn text)

  描述：用于解析指定lsn范围之内的Xlog日志，并返回存放解析内容的路径。可以通过pg_current_xlog_location()获取当前Xlog位置。

  参数：start_lsn为LSN起始位置， end_lsn为LSN结束位置。

  返回值类型：text

  备注：必须是系统管理员或运维管理员才能执行此函数。

- gs_xlogdump_xid(c_xid xid)

  描述：用于解析指定xid的Xlog日志，并返回存放解析内容的路径。可以通过txid_current()获取当前事务ID。

  参数：c_xid为事务ID。

  返回值类型：text

  备注：必须是系统管理员或运维管理员才能执行此函数。

- gs_xlogdump_tablepath(path text, blocknum bigint, relation_type text)

  描述：用于解析指定表页面对应的日志，并返回存放解析内容的路径。

  返回值类型：text

  备注：必须是系统管理员或运维管理员才能执行此函数。

表 1-207 gs_xlogdump_tablepath 参数说明

参数	类型	描述
path	text	● 对于普通表或段页式的普通表，相对路径为：tablespace name/database oid/表的relfilenode(物理文件名)。例如：base/16603/16394。 ● 对于段页式的hashbucket表，相对路径为：tablespace name/database oid/Segment Head的逻辑页号_b(bucketid)。例如：base/16603/16394_b1437。 ● 表文件的相对路径可以通过pg_relation_filepath(table_name text)查找。分区表的路径可以查看pg_partition系统表和调用pg_partition_filepath(partition_oid)。 ● 合法的path格式列举： – global/relNode – base/dbNode/relNode – pg_tblspc/spcNode/version_dir/dbNode/relNode 对于hashbucket表，在此格式基础上，路径末尾加上_b段页式的逻辑页号。
blocknum	bigint	● -1：所有block的信息（强制从磁盘解析）。 ● 0~MaxBlockNumber：对应block的信息。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
relation_type	text	● heap(astore 表) ● btree(BTree 索引) ● segment(段页式)

- gs_xlogdump_parsepage_tablepath(path text, blocknum bigint, relation_type text, read_memory boolean)

  描述：用于解析指定表页面和表页面对应的日志，并返回存放解析内容的路径。可以看做一次执行gs_parse_page_bypath和gs_xlogdump_tablepath。该函数执行的前置条件是表文件存在。如果想查看已删除的表的相关日志，请直接调用gs_xlogdump_tablepath。

  返回值类型：text

  备注：必须是系统管理员或运维管理员才能执行此函数。

  表 1-208 gs_xlogdump_parsepage_tablepath 参数说明

参数	类型	描述
path	text	● 对于普通表，相对路径为：tablespace name/database oid/表的relfilenode(物理文件名)；例如：base/16603/16394。 ● 表文件的相对路径可以通过pg_relation_filepath(table_name text)查找。分区表的路径可以查看pg_partition系统表和调用pg_partition_filepath(partition_oid)。 ● 合法的path格式列举： – global/relNode。 – base/dbNode/relNode。 – pg_tblspc/spcNode/version_dir/dbNode/relNode 对于hashbucket表，在此格式基础上，路径末尾加上_b段页式的逻辑页号。
blocknum	bigint	● -1：所有block的信息（强制从磁盘解析）。 ● 0~MaxBlockNumber：对应block的信息。
relation_type	text	● heap(astore 表) ● btree(BTree 索引) ● indexurq(indexurq 页面）
read_memory	boolean	● false，从磁盘文件解析。 ● true，首先尝试从共享缓冲区中解析该页面；如果共享缓冲区中不存在，则从磁盘文件解析。

- gs_shared_storage_xlogdump_lsn(start_lsn text, end_lsn text, xlog_file_path text)

版权所有 © 华为云计算技术有限公司

描述：用于解析Xlog日志共享盘的指定LSN范围之内的Xlog日志，并返回存放解析内容的路径。可以通过pg_current_xlog_location()获取当前Xlog位置。

返回值类型：text

备注：必须是系统管理员或运维管理员才能执行此函数。

表 1-209 gs_shared_storage_xlogdump_lsn 入参参数说明

参数	类型	描述
start_lsn	text	需要解析的Xlog的起始LSN。
end_lsn	text	需要解析的Xlog的结束LSN。
xlog_file_path	text	可选参数。需要解析的Xlog所在Xlog日志共享盘的路径。不指定时，解析当前CN/DN所有共享盘的Xlog。

使用示例：

```
# 支持两个入参，start_lsn与end_lsn
gaussdb=# SELECT * FROM
gs_shared_storage_xlogdump_lsn('00000000/01000000','00000000/19B08128');
                    output_filepath
------------------------------------------------------------------
 /var/lib/engine/data1/data/dn_6001/01000000_19b08128_6001.xlog
(1 row)

# 支持三个入参，start_lsn、end_lsn与xlog_file_path
gaussdb=# SELECT * FROM
gs_shared_storage_xlogdump_lsn('00000000/01000000','00000000/19B08128','/data/nfs/6001');
                    output_filepath
------------------------------------------------------------------
 /var/lib/engine/data1/data/dn_6001/01000000_19b08128_6001.xlog
(1 row)
```

- gs_shared_storage_ctlinfo()

  描述：用于获取当前CN/DN共享盘元数据相关信息。

  返回值类型：record

表 1-210 gs_shared_storage_ctlinfo 出参参数说明

参数名	类型	描述
xlog_file_path	text	Xlog日志共享盘的路径。
system_id	text	当前Xlog日志共享盘上存储的system_id值。
term	text	当前Xlog日志共享盘上存储的term值。
xlog_file_size	text	Xlog日志共享盘上存Xlog的空间大小。
insert_tail	text	Xlog日志共享盘上当前有效的最小LSN。
insert_head	text	Xlog日志共享盘上当前有效的最大LSN。
shared_storage_flush_lsn	text	Xlog日志共享盘上当前实际落盘LSN。

版权所有 © 华为云计算技术有限公司

使用示例：

```
gaussdb=# SELECT * FROM gs_shared_storage_ctlinfo();
 xlog_file_path |      system_id       | term | xlog_file_size |    insert_tail    |    insert_head    |
shared_storage_flush_lsn
----------------+----------------------+------+----------------+-------------------+-------------------
+--------------------------
 /data/nfs/6001 | 18020659526041888084 | 1    | 1073741824     | 00000000/01000000 |
00000000/19B08128 | 00000000/19B08128
(1 row)
```

- gs_index_verify(oid oid, uint32 blkno)

  描述：用于校验UBtree索引页面或者索引树上key的顺序是否正确。

  返回值类型：record

  表 1-211 gs_index_verify 参数说明

参数	类型	描述
oid	oid	● 索引文件relfilenode，可以通过SELECT relfilenode FROM pg_class where relname='name'查询，其中name表示对应的索引文件名字。
blkno	uint32	● 0：表示检验整个索引树上所有页面。 ● 大于0：表示校验页面编码等于blkno的索引页面。

- gs_index_recycle_queue(Oid oid, int type, uint32 blkno)

  描述：用于解析UBtree索引回收队列信息。

  返回值类型：record

  表 1-212 gs_index_recycle_queue 参数说明

参数	类型	描述
oid	oid	● 索引文件relfilenode,可以通过select relfilenode from pg_class where relname='name'查询，其中name表示对应的索引文件名字。
type	int	● 0：表示解析整个待回收队列。 ● 1：表示解析整个空页队列。 ● 2：表示解析单个页面。
blkno	uint32	回收队列页面编号，该参数只有在type=2的时候有效，blkno有效取值范围为1~4294967294。

- gs_stat_wal_entrytable(int64 idx)

  描述：用于输出Xlog中预写日志插入状态表的内容。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

表 1-213 gs_stat_wal_entrytable 参数说明

参数类型	参数名	类型	描述
输入参数	idx	int64	● -1：查询数组所有元素。 ● 0-最大值：具体某个数组元素内容。
输出参数	idx	uint64	记录对应数组中的下标。
输出参数	endlsn	uint64	记录的LSN标签。标识当前entry对应的Xlog是否已经完全复制到wal buffer中： ● 0：非COPIED。 ● 非0：COPIED。
输出参数	lrc	int32	记录对应的LRC。
输出参数	status	uint32	当前版本预留字段，查询结果显示为0。

- gs_walwriter_flush_position()
  描述：输出预写日志的刷新位置。
  返回值类型：record

  表 1-214 gs_walwriter_flush_position 参数说明

参数类型	参数名	类型	描述
输出参数	last_flush_status_entry	int32	Xlog flush上一个刷盘的tblEntry下标索引。
输出参数	last_scanned_lrc	int32	Xlog flush上一次扫描到的最后一个tblEntry记录的LRC。
输出参数	curr_lrc	int32	WALInsertStatusEntry状态表中LRC最新的使用情况，该LRC表示下一个Xlog记录写入时在WALInsertStatusEntry对应的LRC值。
输出参数	curr_byte_pos	uint64	Xlog记录写入WAL 文件，最新分配的位置，下一个Xlog记录插入点。
输出参数	prev_byte_size	uint32	上一个Xlog记录的长度。
输出参数	flush_result	uint64	当前全局Xlog刷盘的位置。
输出参数	send_result	uint64	当前主机上Xlog发送位置。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	shm_rqst_write_pos	uint64	共享内存中记录的XLogCtl中LogwrtRqst请求的write位置。
输出参数	shm_rqst_flush_pos	uint64	共享内存中记录的XLogCtl中LogwrtRqst请求的flush位置。
输出参数	shm_result_write_pos	uint64	共享内存中记录的XLogCtl中LogwrtResult的write位置。
输出参数	shm_result_flush_pos	uint64	共享内存中记录的XLogCtl中LogwrtResult的flush位置。
输出参数	curr_time	text	当前时间。

- gs_walwriter_flush_stat(int operation)

  描述：用于统计预写日志write与sync的次数频率与数据量，以及Xlog文件的信息。

  返回值类型：record

  表 1-215 gs_walwriter_flush_stat 参数说明

参数类型	参数名	类型	描述
输入参数	operation	int	● -1： 关闭统计开关(默认状态为关闭)。 ● 0：打开统计开关。 ● 1：查询统计信息。 ● 2：重置统计信息。
输出参数	write_times	uint64	Xlog调用write接口的次数。
输出参数	sync_times	uint64	Xlog调用sync接口次数。
输出参数	total_xlog_sync_bytes	uint64	Backend线程请求写入Xlog总量统计值。
输出参数	total_actual_xlog_sync_bytes	uint64	调用sync接口实际刷盘的Xlog总量统计值。
输出参数	avg_write_bytes	uint32	每次调用XLogWrite接口请求写的Xlog量。
输出参数	avg_actual_write_bytes	uint32	实际每次调用write接口写的Xlog量。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	avg_sync_bytes	uint32	平均每次请求sync的Xlog量。
输出参数	avg_actual_sync_bytes	uint32	实际每次调用sync刷盘Xlog量。
输出参数	total_write_time	uint64	调用write操作总时间统计(单位：us)。
输出参数	total_sync_time	uint64	调用sync操作总时间统计(单位：us)。
输出参数	avg_write_time	uint32	每次调用write接口平均时间(单位：us)。
输出参数	avg_sync_time	uint32	每次调用sync接口平均时间(单位：us)。
输出参数	curr_init_xlog_segno	uint64	当前最新创建的Xlog段文件编号。
输出参数	curr_open_xlog_segno	uint64	当前正在写的Xlog段文件编号。
输出参数	last_reset_time	text	上一次重置统计信息的时间。
输出参数	curr_time	text	当前时间。

- pg_ls_tmpdir()

  描述：返回默认表空间下临时目录（pgsql_tmp）中每个文件的名称、大小和最后修改时间。

  参数：nan

  返回值类型：record

  备注：必须是系统管理员或者监控管理员才能执行此函数。

参数类型	参数名	类型	描述
输出参数	name	text	文件名称。
输出参数	size	int8	文件大小（单位：byte）。
输出参数	modification	timestamptz	文件最后修改时间。

- pg_ls_tmpdir(oid)

  描述：返回指定表空间下临时目录（pgsql_tmp）中每个文件的名称、大小和最后修改时间。

  参数：oid

版权所有 © 华为云计算技术有限公司

返回值类型：record

备注：必须是系统管理员或者监控管理员才能执行此函数。

参数类型	参数名	类型	描述
输入参数	oid	oid	表空间id。
输出参数	name	text	文件名称。
输出参数	size	int8	文件大小（单位：byte）。
输出参数	modification	timestamptz	文件最后修改时间。

- pg_ls_waldir()

  描述：返回预写日志(WAL)目录中每个文件的名称、大小和最后修改时间。

  参数：nan

  返回值类型：record

  备注：必须是系统管理员或者监控管理员才能执行此函数。

参数类型	参数名	类型	描述
输出参数	name	text	文件名称。
输出参数	size	int8	文件大小（单位：byte）。
输出参数	modification	timestamptz	文件最后修改时间。

- gs_stat_anti_cache()

  描述：返回AntiCache的相关统计信息。

  参数：nan

  返回值类型：record

参数类型	参数名	类型	描述
输出参数	table_nums	int8	分区数量。
输出参数	table_init_size	int8	分区初始大小。
输出参数	table_max_size	int8	分区大小上限。
输出参数	anti_cache_upper_limit_size	int8	AntiCache大小上限。
输出参数	anti_cache_size	int8	实时AntiCache大小。
输出参数	anti_cache_max_table_size	int8	实时最大分区大小。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	anti_cache_min_table_size	int8	实时最小分区大小。

- gs_stat_vlog_buffer()
  描述：返回verifyLog buffer的相关统计信息。
  参数：nan
  返回值类型：record

参数类型	参数名	类型	描述
输出参数	vbuffer_write_offset	int8	vlog写入位置。
输出参数	vbuffer_flushed_offset	int8	vlog刷盘位置。
输出参数	max_vbuffer_flushed_value	int8	vlog下刷最大字节数。
输出参数	min_vbuffer_flushed_vlaue	int8	vlog下刷最小字节数。
输出参数	ave_vbuffer_flushed_vlaue	int8	vlog下刷平均字节数。
输出参数	vbuffer_flush_latency	int8	vlog下刷时延统计。

- gs_stat_vlog_related_io()
  描述：返回verifylog文件读写的io相关统计信息。
  参数：nan
  返回值类型：record

参数类型	参数名	类型	描述
输出参数	read_data_iops	float8	每秒读文件的页面数。
输出参数	vlog_read_iops	float8	每秒读vlog的页面数。
输出参数	vlog_write_iops	float8	每秒写vlog的页面数。

- gs_stat_vlog_file()
  描述：查询还有哪些Vlog（Verifylog）文件未回收。
  参数：nan
  返回值类型：record

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	last_recycled_segno	int8	已完成回收的最后一个Vlog文件段号。
输出参数	target_recycle_segno	int8	当前待回收的目标Vlog文件段号。
输出参数	max_segno	int8	当前最大的Vlog文件段号。

- pause_anti_cache_recycle(boolean pause)

  描述：设置是否暂停AntiCache回收。本函数仅供内部使用。

  参数：pause表示是否暂停AntiCache回收，类型为boolean。pause为true表示AntiCache暂停回收，pause为false表示AntiCache正常回收。

  返回值类型：text

  示例：

```
#在内部版本中，设置AntiCache正常回收
gaussdb=# select * from pause_anti_cache_recycle(false);
           result
----------------------------
 Anticache recycle resumed.
(1 row)

#在内部版本中，设置AntiCache暂停回收
gaussdb=# select * from pause_anti_cache_recycle(true);
          result
---------------------------
 Anticache recycle paused.
(1 row)

#在正式版本中，不支持设置
gaussdb=# select * from pause_anti_cache_recycle(false);
                        result
-------------------------------------------------------------
 This function is not allowed to use in release version.
(1 row)
```

- gs_write_term_log(void)

  描述：写入一条日志记录DN节点当前的term值。备DN节点返回false，主DN节点写入成功后返回true。

  返回值类型：Boolean

- gs_stat_space(bool init)

  描述：用于查询UStore中做Insert操作时拓展页面的状态。

  返回值类型：record

参数类型	参数名	类型	描述
输入参数	init	bool	是否重置已统计的数据。
输出参数	access_func	int8	relation_get_buffer_for_utuple接口访问总次数。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	cache_blk	int8	relation_get_buffer_for_utuple获取缓存次数。
输出参数	cache_succ	int8	relation_get_buffer_for_utuple获取缓存成功次数。
输出参数	nblk_first	int8	relation_get_buffer_for_utuple中第一次获取nblocks-1次数。
输出参数	nblk_first_succ	int8	relation_get_buffer_for_utuple中第一次获取nblocks-1成功次数。
输出参数	nblk_second	int8	relation_get_buffer_for_utuple中第二次获取nblocks-1次数。
输出参数	nblk_second_succ	int8	relation_get_buffer_for_utuple中第二次获取nblocks-1成功次数。
输出参数	fsm_first	int8	第一次访问FSM次数。
输出参数	fsm_first_success	int8	第一次访问FSM成功次数。
输出参数	fsm_rewrite	int8	FSM回写次数。
输出参数	fsm_second	int8	第二次访问FSM次数。
输出参数	fsm_second_success	int8	第二次访问FSM成功次数。
输出参数	prune_count	int8	relation_get_buffer_for_utuple中Prune次数。
输出参数	prune_space	int8	relation_get_buffer_for_utuple中Prune总空间。
输出参数	coprune_count	int8	联合清理执行次数。
输出参数	coprune_scan_blocks	int8	联合清理扫描总页数。
输出参数	coprune_prune_count	int8	联合清理Prune次数。
输出参数	coprune_prune_space	int8	联合清理Prune总空间。
输出参数	con_extend_count	int8	并发扩页数量。
输出参数	con_extend_time	int8	并发扩页总时间。
输出参数	single_extend_time	int8	单一扩页次数。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT * FROM gs_stat_space(false);
 access_func | cache_blk | cache_succ | nblk_first | nblk_first_succ | nblk_second | nblk_second_succ |
fsm_first | fsm_first_success | fsm_rewrite | fsm_
second | fsm_second_success | prune_count | prune_space | coprune_count | coprune_scan_blocks |
coprune_prune_count | coprune_prune_space | con_extend_count
 | con_extend_time | single_extend_count
-------------+-----------+------------+------------+-----------------+--------------+-------------------+-----------
+-------------------+-------------+-----
-------+--------------------+-------------+-------------+---------------+--------------------+--------------------
+--------------------+-----------------
-+-----------------+---------------------
   10082397 |  10082397 |   10082235 |        88 |             78 |       47021 |            47021 |       162
|              74 |      360996 |
360996 |           270948 |       6711 |          0 |        45497 |            222619 |                 0
|                 0 |            3675
 |       25542884 |              26791
(1 row)
```

备注：请重点关注cache_succ此值较小说明系统缓存失效，prune_space 此值较小表示UStore数据页页面清理机制可能存在问题，con_extend_time此值过高可能表明UStore并发拓页时耗时较高。

- gs_redo_upage(directory_path text, backup_path text, blocknum bigint, relation_type text, xlog_path text, lsn text)

  描述：用于将备份的特定UStore数据页面重放到指定LSN，并在重放期间校验页面，若检测到页面受损则直接落盘该坏块，返回落盘路径、页面LSN以及受损信息，否则重放至指定LSN并落盘页面后返回，必须是系统管理员或运维管理员才能执行此函数。

  返回值类型：record

参数类型	参数	类型	描述
输入参数	directory_path	text	备份文件所在目录。
输入参数	backup_path	text	备份表文件的相对路径，与备份文件所在目录拼成表文件的完整路径，例如，base/15635/12488，当备份不存在时，传入null。
输入参数	blocknum	bigint	0~MaxBlockNumber：对应页面的块号。
输入参数	relation_type	text	• uheap（Ustore数据页）。 • ubtree（Ustore索引页）。 • indexurq（Urq页面）。 • undo_record（Undo record页面）。 • undo_slot（Transaction slot页面）。
输入参数	xlog_path	text	归档日志目录的绝对路径。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输入参数	lsn	text	lsn由两个16进制数(各32 bits)组成，中间通过"/"隔开，例如：2/962D1DF8。如果是0则重放到最新版本。
输出参数	output_filename	text	文件的落盘路径与文件名。
输出参数	output_lsn	text	最后一次页面重放的LSN。
输出参数	corruption_desc	text	页面受损情况描述。

- gs_xlogdump_bylastlsn(last_lsn text, blocknum bigint, relation_type text)

  描述：传入一个页面LSN以及块号，解析LSN对应的WAL日志，并获取对应块号的last LSN继续解析，直到last LSN为0或者更老版本的WAL日志已被复用回收，并将解析后的日志落盘到指定路径，必须是系统管理员或运维管理员才能执行此函数。本系统函数不支持备机调用。

  返回值类型：text

参数类型	参数名	类型	描述
输入参数	last_lsn	text	解析指定页面的LSN，基于十六进制表示，如12BA/32CDEDDD，LSN可通过页面解析工具（gs_parse_page_bypath等）获取。
输入参数	blocknum	bigint	指定页面的逻辑块号。 参数范围：-1~MaxBlockNumber。 块号指定为-1时表示从WAL日志中获取默认块号。
输入参数	relation_type	text	指定解析页面的类型。 参数范围：uheap、ubtree、heap、btree、undo_record、undo_slot。
输出参数	output_filepath	text	WAL日志解析结果的落盘路径。

  示例：

```
# 获取页面LSN信息
# 函数调用前请依据参数说明中的内容确保文件路径真实存在
gaussdb=# SELECT * FROM gs_parse_page_bypath('base/15833/16768', 0, 'uheap', false);
                        output_filepath
-----------------------------------------------------------------
 data1/database/cluster/primary/data/1663_15833_16768_0.page
(1 row)
gaussdb=# SELECT * FROM gs_xlogdump_bylastlsn('0/4593570', -1, 'uheap');
                        output_filepath
-----------------------------------------------------------------
 /data1/database/cluster/primary/data/gs_log/dump/4593570_-1.xlog
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
gaussdb=# SELECT * FROM gs_xlogdump_bylastlsn('0/4593570', 0, 'ubtree');
ERROR:  The input lsn 0/4593570 related xlog is not ubtree.
```

- dbe_perf.get_full_sql_by_parent_id_and_timestamp(parent_id bigint, start_timestamp timestamp with time zone, end_timestamp timestamp with time zone)

  描述：根据parent_id获取某个时间段内，执行某个存储过程的数据库级的全量SQL及其子语句的记录。只可在系统库中查询到结果，用户库中无法查询。

  返回值类型：record

参数	类型	描述
parent_id	bigint	指定需要检索的存储过程的调用语句的unique_sql_id。
start_timestamp	timestamp with time zone	SQL启动时间范围的开始时间点。
end_timestamp	timestamp with time zone	SQL启动时间范围的结束时间点。

  示例：

```
gaussdb=# CREATE TABLE test(a int,b int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# INSERT INTO test values(1,1);
INSERT 0 1
gaussdb=# CREATE PROCEDURE mypro1() as num int;
gaussdb$# begin
gaussdb$# INSERT INTO test values(2,2);
gaussdb$# DELETE FROM test where a = 2;
gaussdb$# end;
gaussdb$# /
CREATE PROCEDURE

--打开参数，跟踪存储过程子语句。
gaussdb=# SET instr_unique_sql_track_type = 'all';
SET

--打开参数，dbe_perf.statement_history表生成全量语句记录。
gaussdb=# SET track_stmt_stat_level = 'L0,L0';
SET

gaussdb=# CALL mypro1();
 mypro1
--------

(1 row)

gaussdb=# SET track_stmt_stat_level = 'off,L0';
SET

gaussdb=# SET instr_unique_sql_track_type = 'top';
SET

--查询关键信息，作为函数参数使用。
gaussdb=# SELECT query,unique_query_id,start_time,finish_time FROM dbe_perf.statement_history;
                query                | unique_query_id |          start_time           |          finish_time
-------------------------------------+-----------------+-------------------------------
+-------------------------------
```


版权所有 © 华为云计算技术有限公司

```
 set track_stmt_stat_level = 'L0,L0'; |       636388010 | 2023-06-02 17:40:49.176155+08 | 2023-06-02
17:40:49.176543+08
 call mypro1();                       |       536458473 | 2023-06-02 17:40:59.028144+08 | 2023-06-02
17:40:59.032027+08
 delete from test where a = ?         |       583323884 | 2023-06-02 17:40:59.029955+08 | 2023-06-02
17:40:59.031577+08
 insert into test values(?,?)         |       769279931 | 2023-06-02 17:40:59.029219+08 | 2023-06-02
17:40:59.029947+08
(4 rows)

--通过外层语句的unique_query_id、外层语句的开始和结束时间作为参数，查询该时间段内，指定存储过
程及其子语句的信息。
gaussdb=# SELECT query FROM
dbe_perf.get_full_sql_by_parent_id_and_timestamp(536458473,'2023-06-02
17:40:59.028144+08','2023-06-02 17:40:59.032027+08');
            query
------------------------------
 call mypro1();
 delete from test where a = ?
 insert into test values(?,?)
(3 rows)

gaussdb=# DROP PROCEDURE mypro1();
DROP PROCEDURE
gaussdb=# DROP TABLE test;
DROP TABLE
```

- gs_index_dump_read(int8 reset, text out_type)

  描述：用于查询索引获取新页面时在循环队列中产生的buffer read信息和索引页面相同key从左到右遍历叶子页面的buffer read信息。

  返回值类型：record

参数类型	参数名	类型	描述
输入参数	reset	int8	• 0：将统计信息reset为初始值0，重新开始统计。 • 1：直接show出当前的统计信息。
输入参数	out_type	text	• urq：输出循环队列的统计信息。 • ubtree：输出索引页的统计信息。 • all：循环队列和索引页的统计信息全部输出。
输出参数	relfilenode	oid	统计到的最大buffer read值对应的索引relfilenode。
输出参数	max_count	int8	最大buffer read值。
输出参数	ave_count	int8	平均buffer read值。

版权所有 © 华为云计算技术有限公司

说明

- 该接口当前仅支持USTORE索引表。
- 该接口执行时，先reset清理、将记录全部置为0。再进行查询，直到下次统计到信息值之前，查询记录会一直为0。查询示例如下：

```
gaussdb=# SELECT * FROM gs_index_dump_read(0, 'all');
relfilenode | max_count | ave_count
--------------+-------------+-------------
|                  |
(1 row)
gaussdb=# SELECT * FROM gs_index_dump_read(1, 'all');
relfilenode | max_count | ave_count
--------------+-------------+-------------
0 |                0 |                0
0 |                0 |                0
(2 rows)
```

- pg_get_ilmdef(pidx integer)

  描述：行存压缩系统函数，根据输入的ilm策略索引返回对应的策略信息

  返回值类型：text

参数类型	参数名	类型	描述
输入参数	pidx	integer	ilm策略的序号。
输出参数	ilm_policy_info	text	指定ilm策略的策略信息。

- gs_shared_storage_flush_stat(int operation)

  描述：统计共享盘的写入数据量、写入速度、写入时间、读取数据量、读取速度、读取时间等。

  返回值类型：record

参数类型	参数名	类型	描述
输入参数	operation	int	● -1：关闭统计开关。 ● 0：打开统计开关（默认状态为打开）。 ● 1：查询统计信息。 ● 2：重置统计信息。 关闭统计开关后重新打开时会清理之前的统计信息。
输出参数	stat_switch	bool	统计开关是否打开。
输出参数	write_times	uint64	shared_storage_xlog_copy线程调用dorado_write_xlog接口的次数。
输出参数	avg_write_bytes	uint32	每次调用dorado_write_xlog接口请求写的Xlog量（单位：B）。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	avg_actual_write_bytes	uint32	实际每次调用dorado_write_xlog接口写的Xlog量（单位：B）。
输出参数	total_write_time	uint64	调用dorado_write_xlog操作总时间统计（单位：us）。
输出参数	avg_write_time	uint32	每次调用dorado_write_xlog接口平均时间（单位：us）。
输出参数	avg_write_speed	uint32	平均调用dorado_write_xlog接口请求写Xlog的速度（单位：KB/s）。
输出参数	avg_actual_write_speed	uint32	实际平均调用dorado_write_xlog接口请求写Xlog的速度（单位：KB/s）。
输出参数	total_write_sleep_time	uint64	shared_storage_xlog_copy线程总睡眠时间。
输出参数	read_times	uint64	walreceiver线程调用dorado_read_xlog接口的次数。
输出参数	avg_read_bytes	uint32	每次调用dorado_read_xlog接口请求写的Xlog量（单位：B）。
输出参数	avg_actual_read_bytes	uint32	实际每次调用dorado_read_xlog接口写的Xlog量（单位：B）。
输出参数	total_read_time	uint64	调用dorado_read_xlog操作总时间统计（单位：us）。
输出参数	avg_read_time	uint32	每次调用dorado_read_xlog接口平均时间（单位：us）。
输出参数	avg_read_speed	uint32	平均调用dorado_read_xlog接口请求写Xlog的速度（单位：KB/s）。
输出参数	avg_actual_read_speed	uint32	实际平均调用dorado_read_xlog接口请求写Xlog的速度（单位：KB/s）。
输出参数	total_read_sleep_time	uint64	walreceiver线程总睡眠时间。
输出参数	stat_start_time	timestamptz	本次统计数据开始时间，若无重置统计信息操作则为GaussDB程序开始运行时间，若有重置统计信息操作则为上次重置时间。
输出参数	stat_end_time	timestamptz	本次统计数据结束时间，若统计开关开启则为当前时间，若统计开关关闭则为上次关闭时间。

示例：

```
gaussdb=# SELECT * FROM gs_shared_storage_flush_stat(1);
 stat_switch | write_times | avg_write_bytes | avg_actual_write_bytes | total_write_time |
```


版权所有 © 华为云计算技术有限公司

```
avg_write_time | avg_write_speed | avg_actual_write_speed | total_write_sleep_time | read_times | avg
_read_bytes | avg_actual_read_bytes | total_read_time | avg_read_time | avg_read_speed |
avg_actual_read_speed | total_read_sleep_time |         stat_start_time         |          stat_end_time

-------------+-------------+-----------------+------------------------+------------------+----------------
+-----------------+-----------------------+-----------------------+------------+----
------------+-----------------------+-----------------+---------------+----------------+-----------------------
+-----------------------+-------------------------------+-----------------------
-------
 t           |       43644 | 8624            | 8624                  |        263922553 | 6047           | 1392           |
1392                  |            26918503404 |       7451 | 140
13          | 14013                 |         5321082 | 714           | 19163          | 19163
|                    0 | 2023-12-18 07:18:39.766855+08 | 2023-12-18 16:29:22.757
535+08
(1 row)
```

- gs_stat_freespace(Oid relid, Oid partid, char type, int sample_ratio)

  描述：通过统计信息或采样的方法估算表/分区表/索引的膨胀率，输出空间使用统计信息及膨胀率信息。

  返回值类型：record

参数类型	参数	类型	描述
输入参数	relation_oid	OID	要查询的数据表/分区表主表/索引的OID。
输入参数	partition_oid	OID	分区表/二级分区表/LOCAL分区索引的OID。 ● 如果relation_oid为非分区表/非LOCAL分区索引/GPI，本入参无意义。 ● 如果relation_oid为分区表/二级分区表/LOCAL分区索引，partition_oid取值为0，代表输出所有分区的信息。
输入参数	relation_type	CHAR	表类型： ● 'r'：表示查询目标表为行存数据表。 ● 'i'：表示查询目标表为行存索引表。
输入参数	sample_ratio	INT8	采样比例，取值范围为0~100。 ● 当sample_ratio取值为0时，INFO为通过统计信息估算出的膨胀率。 ● 当sample_ratio取值为1-100时，INFO为通过采样估算出的膨胀率。
输入参数	read_memory	BOOL	是否从内存读取页面： ● true：表示为从内存读取页面； ● false：表示为从磁盘读取页面。暂时只支持取值为 false。
输出参数	OID	OID	要查询的数据表/分区表/二级分区表/索引的OID。

版权所有 © 华为云计算技术有限公司

参数类型	参数	类型	描述
输出参数	INFO	TEXT	• 当sample_ratio=0时，INFO为估算页面膨胀率。 • 当sample_ratio>0时，INFO为页面使用率采样统计信息，包括页头长度、元组平均长度、平均normal元组数量、平均dead元组数量、平均unused元组数量、平均redirected元组数量、页面平均td数（Ustore数据表）、页面平均potential_freespace、实际freespace（页面upper - lower）、relation总页面数、通过采样信息估算数据表/索引占空间大小、通过采样信息估算数据表/索引膨胀率。

说明

- 该接口当前仅支持行存表，不支持时序表、段页式表、hashbucket表、本地临时表和全局临时表。对于不支持表类型上的索引同样不支持空间统计信息估算。
- CN节点查询该接口，查询结果仅代表本节点相关统计信息与膨胀率估算情况，不包含对应DN的相关信息。如需查询DN节点统计信息与膨胀率估算结果，需要连接DN节点进行查询。
- 备机上使用此接口时，当sample_ratio取值为0时，即根据统计信息估算，无法保证实时准确率。

示例：

```
-- 查询普通表信息
-- 建表语句：
gaussdb=# DROP TABLE IF EXISTS t;
NOTICE:  table "t" does not exist, skipping
DROP TABLE
gaussdb=# CREATE TABLE t(a int) WITH (storage_type=ustore);
CREATE TABLE
gaussdb=# INSERT INTO t VALUES(generate_series(1, 110000));
INSERT 0 110000
-- 更新统计信息：
gaussdb=# VACUUM ANALYZE t;
VACUUM
-- 估算膨胀率：
gaussdb=# SELECT * FROM gs_stat_freespace((SELECT oid FROM pg_class WHERE relname='t'), 0, 'r', 0, false);
  oid  |                                                                                                  info

-------
+-----------------------------------------------------------------------------------------------------------------
-------------------------
-------------------------------------------------------------------
 16735 | live_tuples: 110000, dead_tuples: 0, avg_tuple_len: 4, tuple_header_size: 11, rowptr_size: 440000, page_header_size: 56,  nblocks: 419, approximate_table_size: 2113464 (Bytes), approximate expansion rate: 1.62
(1 row)

-- 采样信息：
gaussdb=# SELECT * FROM gs_stat_freespace((SELECT oid FROM pg_class WHERE relname='t'), 0, 'r', 80, false);
  oid  |
                    info
```

```
-------
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------
----------------------------------------
 16732 | Average page stat info: page_header_size: 28, tuple has storage: 222.22, normal tuple:
222.22, dead tuple: 0.00, unused tuple: 0.00, redirected tuple: 0.00, td_count: 0.00, potential_freespace:
0.00, actual_freespace: 152.00. Freespace stat info: nblocks: 45, approximate_table_size: 342495
(Bytes), approximate expansion rate: 1.08.
(1 row)

-- 查询分区表单个分区信息
-- 建表语句：
gaussdb=# DROP TABLE IF EXISTS t;
DROP TABLE
gaussdb=# CREATE TABLE t(a int) WITH (storage_type=ustore) PARTITION BY range(a) (PARTITION
p1 VALUES LESS THAN(3000), PARTITION p2 VALUES LESS THAN(maxvalue));
CREATE TABLE
gaussdb=# INSERT INTO t VALUES(generate_series(1, 10000));
INSERT 0 10000

-- 更新统计信息：
gaussdb=# VACUUM ANALYZE t;
VACUUM

-- 查询 oid：
gaussdb=# SELECT oid FROM pg_class WHERE relname='t';
  oid
-------
 16802
(1 row)

gaussdb=# SELECT oid, relname, parttype FROM pg_partition;
  oid  | relname | parttype
-------+---------+----------
 16805 | t       | r
 16806 | p1      | p
 16807 | p2      | p
(3 rows)

-- 估算膨胀率：
gaussdb=# SELECT * FROM gs_stat_freespace(16802, 16806, 'r', 0, false);
  oid  |                                                                                                info

-------
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------
------------------------------------------------------------------
 16806 | live_tuples: 5998, dead_tuples: 0, avg_tuple_len: 4, tuple_header_size: 11, rowptr_size: 23992,
page_header_size: 56,  nblocks: 12, approximate_table_size: 114634 (Bytes), approximate expansion
rate: 0.86
(1 row)

-- 采样信息：
gaussdb=# SELECT * FROM gs_stat_freespace(16802, 16807, 'r', 0, false);
  oid  |                                                                                                info

-------
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------
------------------------------------------------------------------
 16807 | live_tuples: 14002, dead_tuples: 0, avg_tuple_len: 4, tuple_header_size: 11, rowptr_size: 56008,
page_header_size: 56,  nblocks: 27, approximate_table_size: 267550 (Bytes), approximate expansion
rate: 0.83
(1 row)

-- 查询分区表所有分区信息
-- 建表语句：
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE IF EXISTS t;
DROP TABLE
gaussdb=# CREATE TABLE t(a int) WITH (storage_type=ustore) PARTITION BY range(a) (PARTITION
p1 VALUES LESS THAN(3000), PARTITION p2 VALUES LESS THAN(maxvalue));
CREATE TABLE
gaussdb=# INSERT INTO t VALUES(generate_series(1, 10000));
INSERT 0 10000

-- 更新统计信息：
gaussdb=# VACUUM ANALYZE t;

-- 查询 oid：
gaussdb=# SELECT oid FROM pg_class WHERE relname='t';
gaussdb=#
  oid
-------
 16802
(1 row)

gaussdb=# SELECT oid, relname, parttype FROM pg_partition;
  oid  | relname | parttype
-------+---------+----------
 16805 | t       | r
 16806 | p1      | p
 16807 | p2      | p
(3 rows)

-- 估算膨胀率：
gaussdb=# SELECT * FROM gs_stat_freespace(16802, 0, 'r', 0, false);
  oid  |                                                                          info

-------
+-------------------------------------------------------------------------------------------------------------------
-------------------------
---------------------------------------------------------------
 16806 | live_tuples: 5998, dead_tuples: 0, avg_tuple_len: 4, tuple_header_size: 11, rowptr_size: 23992,
page_header_size: 56,  nblocks: 12, approximate_table_size: 114634 (Bytes), approximate expansion
rate: 0.86
 16807 | live_tuples: 14002, dead_tuples: 0, avg_tuple_len: 4, tuple_header_size: 11, rowptr_size: 56008,
page_header_size: 56,  nblocks: 27, approximate_table_size: 267550 (Bytes), approximate expansion
rate: 0.83
(2 rows)

-- 采样信息：
gaussdb=# SELECT * FROM gs_stat_freespace(16802, 0, 'r', 60, false);
  oid  |
                       info

-------
+-------------------------------------------------------------------------------------------------------------------
-------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------
-------------------------------------------
 16806 | Average page stat info: page_header_size: 56, tuple has storage: 249.92, normal tuple:
249.92, dead tuple: 0.00, unused tuple: 0.00, redirected tuple: 0.00, td_count: 4.00, potential_freespace:
1034.33, actual_freespace: 1034.33. Freespace stat info: nblocks: 12, approximate_table_size: 84636
(Bytes), approximate expansion rate: 1.16.
 16807 | Average page stat info: page_header_size: 56, tuple has storage: 259.30, normal tuple:
259.30, dead tuple: 0.00, unused tuple: 0.00, redirected tuple: 0.00, td_count: 4.00, potential_freespace:
771.70, actual_freespace: 771.70. Freespace stat info: nblocks: 27, approximate_table_size:
197532(Bytes), approximate expansion rate: 1.12.
(2 rows)

-- 查询普通索引信息
-- 建表语句：
gaussdb=# DROP TABLE IF EXISTS t;
DROP TABLE
gaussdb=# CREATE TABLE t(a int, b int, c varchar(1000)) WITH (storage_type=ustore);
```

```
CREATE TABLE
gaussdb=# CREATE INDEX ON t(a);
CREATE INDEX
gaussdb=# INSERT INTO t VALUES(generate_series(1, 100000), generate_series(1, 100000),
'kjgkjature');
INSERT 0 100000

-- 更新统计信息：
gaussdb=# VACUUM ANALYZE t;
VACUUM

-- 估算膨胀率：
gaussdb=# SELECT * FROM gs_stat_freespace((SELECT oid FROM pg_class WHERE relname='t_a_idx'),
0, 'i', 0, false);
  oid  |                                                                                                                    info

-------
+-------------------------------------------------------------------------------------------------------------------
-------------------------
----------------------------------------------------------------------------------------------------
 16776 | live_tuples: 100000, dead_tuples: 0, origin_avg_tuple_len 4, max_align_avg_tuple_len: 8,
tuple_header_size: 16, rowptr_size: 401548, page_header_size: 24, nblocks: 387,
approximate_table_size: 2838700 (Bytes), approximate expansion rate: 1.12
(1 row)

-- 采样信息：
gaussdb=# SELECT * FROM gs_stat_freespace((SELECT oid FROM pg_class WHERE relname='t_a_idx'),
0, 'i', 60, false);
  oid  |
                      info

-------
+-------------------------------------------------------------------------------------------------------------------
-------------------------
-------------------------------------------------------------------------------------------------------------------
-------------------------------
-------------------------------------------
 16776 | Average page stat info: page_header_size: 24, tuple has storage: 260.39, normal tuple:
260.39, dead tuple: 0.00, unused tuple: 0.02, redirected tuple: 0.00, td_count: 0.00, potential_freespace:
0.00, actual_freespace: 845.05. Freespace stat info: nblocks: 387, approximate_table_size:
3044529(Bytes), approximate expansion rate: 1.04.
(1 row)

-- 查询 local 分区索引单个分区信息
-- 建表语句：
gaussdb=# DROP TABLE IF EXISTS t;
DROP TABLE
gaussdb=# CREATE TABLE t(a int) WITH (storage_type=ustore) PARTITION BY range(a) (PARTITION
p1 VALUES LESS THAN(3000), PARTITION p2 VALUES LESS THAN(maxvalue));
CREATE TABLE
gaussdb=# CREATE INDEX ON t(a) LOCAL;
CREATE INDEX
gaussdb=# INSERT INTO t VALUES(generate_series(1, 10000));
INSERT 0 10000

-- 更新统计信息：
gaussdb=# VACUUM ANALYZE t;
VACUUM

-- 查询 oid：
gaussdb=# SELECT oid FROM pg_class WHERE relname='t_a_idx';
  oid
-------
 16814
(1 row)

gaussdb=# SELECT oid, relname, parttype FROM pg_partition;
  oid  | relname  | parttype
-------+----------+----------
```

```
 16811 | t        | r
 16812 | p1       | p
 16813 | p2       | p
 16815 | p1_a_idx | x
 16816 | p2_a_idx | x
(5 rows)

-- 估算膨胀率：
gaussdb=# SELECT * FROM gs_stat_freespace(16814, 16815, 'i', 0, false);
  oid  |                                                                                          info

-------
+-------------------------------------------------------------------------------------------------------------
-------------------------
-------------------------------------------------------------------------------------------------
 16815 | live_tuples: 10000, dead_tuples: 0, origin_avg_tuple_len 4, max_align_avg_tuple_len: 8, 
tuple_header_size: 16, rowptr_size: 40056, page_header_size: 24, nblocks: 14, approximate_table_size: 
281400 (Bytes), approximate expansion rate: 0.41
(1 row)

-- 采样信息：
gaussdb=# SELECT * FROM gs_stat_freespace(16814, 16815, 'i', 80, false);
  oid  |
                   info

-------
+-------------------------------------------------------------------------------------------------------------
-------------------------
-------------------------------------------------------------------------------------------------------------
-------------------------------
----------------------------------------
 16815 | Average page stat info: page_header_size: 24, tuple has storage: 215.93, normal tuple: 
215.93, dead tuple: 0.00, unused tuple: 0.50, redirected tuple: 0.00, td_count: 0.00, potential_freespace: 
0.00, actual_freespace: 2087.43. Freespace stat info: nblocks: 14, approximate_table_size: 91476 
(Bytes), approximate expansion rate: 1.25.
(1 row)

-- 查询 local 分区索引所有分区信息
-- 建表语句：
gaussdb=# DROP TABLE IF EXISTS t;
DROP TABLE
gaussdb=# CREATE TABLE t(a int) WITH (storage_type=ustore) PARTITION BY range(a) (PARTITION 
p1 VALUES LESS THAN(3000), PARTITION p2 VALUES LESS THAN(maxvalue));
CREATE TABLE
gaussdb=# CREATE INDEX ON t(a) LOCAL;
CREATE INDEX
gaussdb=# INSERT INTO t VALUES(generate_series(1, 10000));
INSERT 0 10000
更新统计信息：
gaussdb=# VACUUM ANALYZE t;
VACUUM
-- 查询 oid：
gaussdb=# SELECT oid FROM pg_class WHERE relname='t_a_idx';
  oid
-------
 16814
(1 row)

-- 估算膨胀率：
gaussdb=# SELECT * FROM gs_stat_freespace(16814, 0,  'i', 0, false);
  oid  |                                                                                          info

-------
+-------------------------------------------------------------------------------------------------------------
-------------------------
-------------------------------------------------------------------------------------------------
 16815 | live_tuples: 10000, dead_tuples: 0, origin_avg_tuple_len 4, max_align_avg_tuple_len: 8, 
tuple_header_size: 16, rowptr_size: 40056, page_header
_size: 24, nblocks: 14, approximate_table_size: 281400 (Bytes), approximate expansion rate: 0.41
```

```
 16816 | live_tuples: 10000, dead_tuples: 0, origin_avg_tuple_len 4, max_align_avg_tuple_len: 8,
tuple_header_size: 16, rowptr_size: 40116, page_header
_size: 24, nblocks: 29, approximate_table_size: 282900 (Bytes), approximate expansion rate: 0.84
(2 rows)

-- 采样信息：
gaussdb=# SELECT * FROM gs_stat_freespace(16814, 0,  'i', 80, false);
  oid  |
                    info

-------
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------
----------------------------------------
 16815 | Average page stat info: page_header_size: 24, tuple has storage: 215.93, normal tuple:
215.93, dead tuple: 0.00, unused tuple: 0.50, redirected tuple: 0.00, td_count: 0.00, potential_freespace:
0.00, actual_freespace: 2087.43. Freespace stat info: nblocks: 14, approximate_table_size: 91476
(Bytes), approximate expansion rate: 1.25.
 16816 | Average page stat info: page_header_size: 24, tuple has storage: 243.28, normal tuple:
243.28, dead tuple: 0.00, unused tuple: 0.24, redirected tuple: 0.00, td_count: 0.00, potential_freespace:
0.00, actual_freespace: 1323.03. Freespace stat info: nblocks: 29, approximate_table_size: 213237
(Bytes), approximate expansion rate: 1.11.
(2 rows)

-- 异常输入：
gaussdb=# SELECT * FROM gs_stat_freespace(16814, 0, 'k', 0, false);
ERROR:  Wrong relation_type input, type should be either 'r' or 'i'

gaussdb=# SELECT * FROM gs_stat_freespace(16814, 0, 'i', 600, false);
ERROR:  Wrong sample_ratio input, sample_ratio should be an integer between 0 and 100

gaussdb=# SELECT * FROM gs_stat_freespace(16814, 0, 'r', 60, false);
ERROR:  "t_a_idx" is not a ordinary relation or toast relation.
```

- gin_clean_pending_list(index regclass)

  描述：清理指定GIN/UGIN索引的“pending list”，将里面的条目批量移动到主索引页面中，返回清理的页面数量。如果pending list过长，可能需要多次执行该函数进行清理，一直清理到返回结果为零。如果索引未开启fastupdate选项，则不会发生清理，结果为0。

  返回值类型：bigint

- gs_walrcv_writer_stat(int operation)

  描述：用于统计备机接收WAL日志write与sync的次数频率与数据量，以及Xlog文件的信息。

  返回值类型：record

参数类型	参数名	类型	描述
输入参数	operation	int	● -1：表示关闭统计开关（默认状态为关闭）。 ● 0：表示打开统计开关。 ● 1：表示查询统计信息。 ● 2：表示重置统计信息。 关闭统计开关后重新打开时，会清理之前的统计信息。
输出参数	enable_stat	bool	统计开关是否打开。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	total_write_bytes	uint64	备机写入wal文件总量统计值（单位：B）。
输出参数	write_count	uint64	备机写入wal文件的次数。
输出参数	total_write_time	uint64	备机写入wal文件操作总时间统计（单位：us）。
输出参数	avg_write_time	uint64	备机写入wal文件每次调用write接口平均时间（单位：us）。
输出参数	max_write_time	uint64	备机写入wal文件调用write接口最大用时（单位：us）。
输出参数	avg_write_bytes	uint64	备机平均每次写入wal文件的数值（单位：B）。
输出参数	total_sync_time	uint64	备机wal文件落盘操作总时间统计（单位：us）。
输出参数	avg_sync_time	uint64	备机wal文件落盘每次调用sync接口平均时间（单位：us）。
输出参数	max_sync_time	uint64	备机写入wal文件调用sync接口最大用时（单位：us）。
输出参数	current_xlog_segno	uint64	备机当前正在写的Xlog段文件编号。
输出参数	newest_xlog_segno	uint64	备机最新创建的Xlog段文件编号。
输出参数	stat_start_time	timestamptz	本次统计数据开始时间。
输出参数	stat_end_time	timestamptz	本次统计数据结束时间。

示例：

```
gaussdb=# SELECT * FROM gs_walrcv_writer_stat(1);
-[ RECORD 1 ]------+------------------------------
enable_stat        | t
total_write_bytes  | 100754464
write_count        | 88426
total_write_time   | 3811972
avg_write_time     | 43
max_write_time     | 381739
avg_write_bytes    | 1139
total_sync_time    | 22280240
avg_sync_time      | 251
max_sync_time      | 655533
current_xlog_segno | 8154
newest_xlog_segno  | 8274
stat_start_time    | 2024-12-05 09:29:15.324159+08
stat_end_time      | 2024-12-05 09:29:45.92897+08
```


版权所有 © 华为云计算技术有限公司

- gstrace_reg_sql(scope text, unique_sql_id text)

  描述：添加跟踪的SQL语句，1个数据库实例最多支持1000个SQL语句的跟踪。执行成功后返回添加跟踪的SQL语句的结果是成功或失败。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。
unique_sql_id	必选。指定添加跟踪的SQL语句的unique sql id。	text类型。通过statement_history表查询到的unique_query_id。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT pg_catalog.gstrace_reg_sql('LOCAL', '3545025713');
  -[ RECORD 1 ]------+------------------------------
   (datanode,"ACTION:      [GSTRACE]->[TRACE_SQL]          +
   RESULT:                                                 +
   SUCCESSFUL - etrace unique_sql_id (3545025713) tracked")
  ```

- gstrace_reg_func(scope text, function_name text, so_index text, so_offset text, level text)

  描述：添加跟踪的函数，1个数据库实例最多支持1000个函数的跟踪。执行成功后返回添加跟踪的函数的结果是成功或失败。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。
function_name	必选。指定添加跟踪的函数的名称。	text类型。仅支持系统函数的名称。
so_index	可选。函数所在So的序列号。	text类型。通过pg_catalog.gstrace_show_mangled_func()函数查询到的系统函数的so_index。
so_offset	填入so_index时必选，否则不选。函数在So的偏移量。	text类型。通过pg_catalog.gstrace_show_mangled_func()函数查询到的系统函数的so_offset。
level	必选。函数下钻等级。	text类型。取值范围为'0'、'1'。'0'表示只跟踪函数本身，'1'表示跟踪函数和其子函数。

版权所有 © 华为云计算技术有限公司

返回值类型：text

示例1，入参不含So序列号和So偏移量：

```
gaussdb=# SELECT pg_catalog.gstrace_reg_func('LOCAL', 'pg_sleep', '0');
-------------------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[TRACE_FUNC]                         +
 RESULT:                                                                 +
 SUCCESSFUL - etrace function (_Z8pg_sleepP20FunctionCallInfoData) tracked")
```

示例2，入参包含So序列号和So偏移量：

```
gaussdb=# SELECT pg_catalog.gstrace_show_mangled_func('LOCAL', 'pg_sleep');
                      gstrace_show_mangled_func
-------------------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[SHOW_MANGLED_FUNC]                  +
 RESULT:                                                                 +
 SUCCESSFUL - function (pg_sleep)'s gangled func are these:              +
 function_name                          so_index so_name        so_offset+
 _Z8pg_sleepP20FunctionCallInfoData      0        gaussdb        59877694 +
 ")

gaussdb=# SELECT pg_catalog.gstrace_reg_func('LOCAL', 'pg_sleep', '0', '59877694', '0');
                        gstrace_reg_func
-------------------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[TRACE_FUNC]                         +
 RESULT:                                                                 +
 SUCCESSFUL - etrace function (_Z8pg_sleepP20FunctionCallInfoData) tracked")
```

- gstrace_unreg_sql(scope text, unique_sql_id text)

  描述：删除跟踪的SQL语句。执行成功后返回删除跟踪的SQL语句的结果是成功或失败。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。
unique_sql_id	必选。指定删除跟踪的SQL语句的unique sql id。	text类型。通过statement_history表查询到的unique_query_id。

  返回值类型：text

  示例：

```
gaussdb=# SELECT pg_catalog.gstrace_unreg_sql('LOCAL', '3545025713');
                     gstrace_unreg_sql
-------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[UNTRACE_SQL]              +
 RESULT:                                                   +
 SUCCESSFUL - etrace unique_sql_id (3545025713) delete tracked")
```

- gstrace_unreg_func(scope text, function_name text, so_index text, so_offset text)

  描述：删除跟踪的函数。执行成功后返回删除跟踪的函数的结果是成功或失败。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

版权所有 © 华为云计算技术有限公司

参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。
function_name	必选。指定删除跟踪的函数的名称。	text类型。仅支持系统函数的名称。
so_index	可选。函数所在So的序列号。	text类型。通过pg_catalog.gstrace_show_mangled_func()函数查询到的系统函数的so_index。
so_offset	填入so_index时必选，否则不选。函数在So的偏移量。	text类型。通过pg_catalog.gstrace_show_mangled_func()函数查询到的系统函数的so_offset。

返回值类型：text

示例1，入参不含So序列号和So偏移量：

```
gaussdb=# SELECT pg_catalog.gstrace_unreg_func('LOCAL', 'pg_sleep');
                            gstrace_unreg_func
-------------------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[UNTRACE_FUNC]                          +
 RESULT:                                                                    +
 SUCCESSFUL - etrace function (_Z8pg_sleepP20FunctionCallInfoData) untracked")
```

示例2，入参包含So序列号和So偏移量：

```
gaussdb=# SELECT pg_catalog.gstrace_show_mangled_func('LOCAL', 'pg_sleep');
                          gstrace_show_mangled_func
-------------------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[SHOW_MANGLED_FUNC]                     +
 RESULT:                                                                    +
 SUCCESSFUL - function (pg_sleep)'s gangled func are these:                 +
 function_name                          so_index so_name        so_offset+
 _Z8pg_sleepP20FunctionCallInfoData       0        gaussdb        59877694 +
 ")

gaussdb=# SELECT pg_catalog.gstrace_unreg_func('LOCAL', 'pg_sleep', '0', '59877694');
                            gstrace_unreg_func
-------------------------------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[UNTRACE_FUNC]                          +
 RESULT:                                                                    +
 SUCCESSFUL - etrace function (_Z8pg_sleepP20FunctionCallInfoData) untracked")
```

- gstrace_unreg_all_sql(scope text)

  描述：删除全部跟踪的SQL语句。执行成功后返回删除跟踪的SQL语句的结果是成功或失败。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

版权所有 © 华为云计算技术有限公司

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。

返回值类型：text

示例：

```
gaussdb=# SELECT pg_catalog.gstrace_unreg_all_sql('LOCAL');
               gstrace_unreg_all_sql
-----------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[UNTRACE_ALL_SQL]+
 RESULT:                                             +
 SUCCESSFUL - all traced SQL have been deleted!")
```

- gstrace_unreg_all_func(scope text)

  描述：删除全部跟踪的函数。执行成功后返回删除跟踪的函数的结果是成功或失败。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。

  返回值类型：text

  示例：

```
gaussdb=# SELECT pg_catalog.gstrace_unreg_all_func('LOCAL');
               gstrace_unreg_all_func
-----------------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[UNTRACE_ALL_FUNC]+
 RESULT:                                              +
 SUCCESSFUL - all tracked functions have been deleted!+
 ")
```

- gstrace_list_configured_sql(scope text)

  描述：查看正在跟踪的SQL语句。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。

  返回值类型：setof record

  返回字段说明：

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	字段说明
unique_sql_id	bigint	跟踪SQL语句的unique sql id。

示例：

```
gaussdb=# SELECT pg_catalog.gstrace_list_configured_sql('LOCAL');
            gstrace_list_configured_sql
-----------------------------------------------------
 (datanode,"ACTION:      [GSTRACE]->[LIST_TRACE_SQL]+
 RESULT:                                           +
 SUCCESSFUL - these unique_sql_id are tracking:    +
 unique_sql_id                                      +
 3545025713                                          +
 ")
```

- gstrace_list_configured_func(scope text)

  描述：查看正在跟踪的函数。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。

  返回值类型：setof record

  返回字段说明：

字段名称	字段类型	字段说明
function_name	text	正在跟踪的函数名称。
so_index	int	函数所在So的序列号，0：gaussdb bin，1：libgtmclient.so。
so_name	text	So的名称。
so_offset	int	函数在So的偏移量。
level	int	函数下钻等级。0表示只跟踪函数本身，1表示跟踪函数和其子函数。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT pg_catalog.gstrace_list_configured_func('LOCAL');
                          gstrace_list_configured_func
-----------------------------------------------------------------------------------------
(datanode,"ACTION:      [GSTRACE]->[LIST_TRACE_FUNC]                                +
RESULT:                                                                             +
SUCCESSFUL - these function are tracking:                                           +
function_name                   so_index so_name        so_offset       level+
_Z8pg_sleepP20FunctionCallInfoData      0       gaussdb         59877694        0    +
")
```

- gstrace_show_traced_func(scope text)

  描述：查看正在跟踪的函数的状态。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。

  返回值类型：setof record

  返回字段说明：

字段名称	字段类型	字段说明
function_name	text	正在跟踪的函数名称。
so_index	int	函数所在So的序列号，0：gaussdb bin，1：libgtmclient.so。
so_name	text	So的名称。
so_offset	int	函数在So的偏移量。
level	int	函数下钻等级。0表示只跟踪函数本身，1表示跟踪函数和其子函数。
module	text	申请跟踪此函数的模块。
if_traced	bool	函数汇编是否有跳转跟踪的指令。
if_hp_conflict	bool	是否因为和热补丁冲突而自动取消跟踪。
if_blacklist	bool	函数是否在跟踪函数黑名单中。
ref_count	int	函数被跟踪的引用次数。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT pg_catalog.gstrace_show_traced_func('LOCAL');
                                                       gstrace_show_traced_func
--------------------------------------------------------------------------------------------------------------------------------
---------------------------------------
 (datanode,"ACTION:      [GSTRACE]-
>[SHOW_TRACED_FUNC]                                                                                    +

RESULT:
        +
 SUCCESSFUL - tracked function info are
these:                                                                                   +
 function_name                         so_index so_name        so_offset       level   module  if_traced
if_hp_conflict  if_blacklist    ref_count+
 _Z8pg_sleepP20FunctionCallInfoData       0        gaussdb        59877694        0       1       true
false           false           1        +
 ")
```

- gstrace_show_mangled_func(scope text, function_name text)

  描述：查看指定函数mangle后的名称。调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。取值范围为'LOCAL、'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。
function_name	必选。指定添加跟踪的函数的名称。	text类型。仅支持系统函数的名称。

  返回值类型：setof record

  返回字段说明：

字段名称	字段类型	字段说明
function_name	text	正在跟踪的函数名称。
so_index	int	函数所在So的序列号，0：gaussdb bin，1：libgtmclient.so。
so_name	text	So的名称。
so_offset	int	函数在So的偏移量。

  示例：

```
gaussdb=# SELECT pg_catalog.gstrace_show_mangled_func('LOCAL',
'_Z8pg_sleepP20FunctionCallInfoData');
                          gstrace_show_mangled_func
```

```
--------------------------------------------------------------------------------
(datanode,"ACTION:      [GSTRACE]->[SHOW_MANGLED_FUNC]                          +
RESULT:                                                                         +
SUCCESSFUL - function (_Z8pg_sleepP20FunctionCallInfoData)'s gangled func are these:+
function_name                   so_index so_name        so_offset       +
_Z8pg_sleepP20FunctionCallInfoData      0       gaussdb         59877694        +
")
```

- gstrace_show_subfunc(scope text, function_name text, so_index text, so_offset text)

  描述：查看指定函数的全部子函数，调用该函数必须有MONADMIN权限或者SYSADMIN权限。

  参数：

入参	说明	取值范围
scope	必选。功能的范围。	text类型。分布式执行'LOCAL'或'GLOBAL'。'LOCAL'表示仅在当前节点生效，'GLOBAL'表示在所有CN/DN节点上生效。
function_name	必选。指定添加跟踪的函数的名称。	text类型。仅支持系统函数的名称。
so_index	可选。函数所在So的序列号。	text类型。通过pg_catalog.gstrace_show_mangled_func()函数查询到的系统函数的so_index。
so_offset	填入so_index时必选，否则不选。函数在So的偏移量。	text类型。通过pg_catalog.gstrace_show_mangled_func()函数查询到的系统函数的so_offset。

  返回值类型：setof record

  返回字段说明：

字段名称	字段类型	字段说明
function_name	text	正在跟踪的函数名称。
so_index	int	函数所在So的序列号，0：gaussdb bin，1：libgtmclient.so。
so_name	text	So的名称。
so_offset	int	函数在So的偏移量。

  示例：

```
gaussdb=# SELECT pg_catalog.gstrace_show_subfunc('LOCAL', '_Z8pg_sleepP20FunctionCallInfoData');
                                gstrace_show_subfunc
```


版权所有 © 华为云计算技术有限公司

```
-----------------------------------------------------------------------------------------------------
(datanode,"ACTION:      [GSTRACE]->[SHOW_SUBFUNC]                                                    +
RESULT:                                                                                              +
SUCCESSFUL - function (_Z8pg_sleepP20FunctionCallInfoData)'s sub func are these:                     +
function_name                                          so_index so_name        so_offset+
_Z14DatumGetFloat8m                                          0        gaussdb        68889422 +
_Z21get_current_timestampv                                   0        gaussdb        63347226 +
_ZL32optutil_gsstat_report_waitstatus23OptUtilExplainWaitStateb  0        gaussdb        59848101 +
_Z17ProcessInterruptsv                                       0        gaussdb        94358107 +
_Z9pg_usleepl                                                0        gaussdb        55249960 +
")
```

- gs_last_demote_spent_time()

  描述：获取主降备关键步骤耗时，包括线程退出及初始化、checkpoint、共享内存退出及初始化、双写初始化以及bufferpool初始化等。主降备信息查询需要在执行主降备操作的集群的主节点上进行，集群间以及集群内switchover主降备成功后可以查询到相关信息，信息存储于内存，重启后会重置。

  返回类型：setof record

  返回字段说明：

字段名称	字段类型	字段说明
key_step_name	text	主降备关键步骤： ● thread_exit：主降备线程退出。 ● checkpoint：主降备刷脏。 ● shmem_exit：主降备共享内存退出。 ● shmem_init：主降备共享内存初始化。 ● bufferpool_init：主降备buffer池初始化。 ● dw_init：主降备双写初始化。 ● thread_init：主降备线程初始化。
spent_time	int	主降备关键步骤耗时。

  示例：

```
gaussdb=# SELECT * FROM gs_last_demote_spent_time();
  key_step_name  | spent_time
-----------------+------------
 thread_exit     |    1.00299
 checkpoint      |    3.19896
 shmem_exit      |    .397594
 shmem_init      |    1.74218
 bufferpool_init |    1.24905
 dw_init         |    .038082
 thread_init     |    .400442
(7 rows)
```

- gs_tcache_memory_flush()

  描述：手动唤醒空闲业务线程和stream业务线程，并在对应线程上执行mallctl("thread.tcache.flush", NULL, NULL, NULL, 0);操作，主动清理jemalloc的tcache缓存。

  返回值类型：boolean。成功时则返回t（表示true），失败时则返回f（表示false）。

版权所有 © 华为云计算技术有限公司

说明

- 该函数仅在开启线程池的情况下生效。
- 仅具有SYSADMIN权限的用户可以使用该函数。
- 通过此函数可降低other memory，但仅消减jemalloc tcache，对于其他原因导致的other memory升高的场景无效（例如，由大量硬解析引起的内存碎片导致的other memory冲高场景，则无法消减）。

1.6.28 SPM 计划管理函数

注意

当前SPM计划管理特性在分布式下不支持，所以在分布式下调用下面的这些函数都会给出分布式下不支持的错误提示。

- GS_SPM_EVOLUTE_PLAN(sql_hash, plan_hash)

  描述：GS_SPM_EVOLUTE_PLAN属于DBE_SQL_UTIL schema，是SPM特性用于计划演进的接口函数。

  参数：参数介绍见表1-216。

  表 1-216 GS_SPM_EVOLUTE_PLAN 入参和返回值列表

参数	类型	描述	取值范围
sql_hash	IN bigint	SQL文本的hash值。	-
plan_hash	IN bigint	SQL计划outline文本的hash值。	-
evolute_status	OUT boolean	演进行为是否正常完成：t表示正常；异常会报错。	t/f

  返回值类型：boolean

- GS_SPM_SET_PLAN_STATUS(sql_hash, plan_hash, plan_status)

  描述：GS_SPM_SET_PLAN_STATUS属于DBE_SQL_UTIL schema，它是SPM特性用于修改baseline状态的接口函数，该接口生效时，会同步刷新gs_spm_baseline表中的invalid字段为false。

  参数：参数介绍见表1-217。

  表 1-217 GS_SPM_SET_PLAN_STATUS 入参和返回值列表

参数	类型	描述	取值范围
sql_hash	IN bigint	SQL文本的hash值。	-
plan_hash	IN bigint	SQL计划outline文本的hash值。	-

版权所有 © 华为云计算技术有限公司

参数	类型	描述	取值范围
plan_status	IN text	计划的状态。	• ACC：表示已经接受的计划。 • UNACC：表示未接受的计划。 • FIXED：一种特殊的ACC状态的计划，计划的匹配优先级高于ACC状态的计划。
execute_status	OUT boolean	修改计划状态行为是否正常完成：t表示正常；异常会报错。	t/f

返回值类型：boolean

- GS_SPM_DISPLAY_PLANS(sql_hash)

  描述：GS_SPM_DISPLAY_PLANS属于DBE_SQL_UTIL schema，它是SPM特性用于查看单条SQL所有baseline的接口函数。

  参数：参数介绍见**表1-218**。

  表 1-218 GS_SPM_DISPLAY_PLANS 入参和返回值列表

参数	类型	描述	取值范围
sql_hash	IN bigint	SQL唯一标识。	-
sql_hash	OUT bigint	SQL文本的hash值。	-
plan_hash	OUT bigint	SQL计划outline文本的hash值。	-
outline	OUT text	当前计划对应outline的所有Hint的组合字符串。	-
cost	OUT double	当前计划的代价。	-

版权所有 © 华为云计算技术有限公司

参数	类型	描述	取值范围
status	OUT text	当前计划的状态。	• ACC：表示已经接受的计划。 • UNACC：表示未接受的计划。 • FIXED：一种特殊的ACC状态的计划，计划的匹配优先级高于ACC状态的计划。
gplan	OUT boolean	当前计划是否是gplan。t表示是gplan；异常会报错。	t/f
namespace_oids	OUT text	当前search_path中的OID列表，其中OID满足以下特征： • 用户所拥有权限的Schema的OID • 存在的Schema的OID • 非本地临时表所在Schema的OID	-

返回值类型：bigint, text, double, text, boolean

- GS_SPM_RELOAD_PLAN(sql_hash，plan_hash)

  描述：GS_SPM_RELOAD_PLAN属于DBE_SQL_UTIL schema，它是SPM特性用于加载baseline系统表中某baseline到SPM global cache中的接口函数。

  参数：参数介绍见**表1-219**。

  表 1-219 GS_SPM_RELOAD_PLAN 入参和返回值列表

参数	类型	描述	取值范围
sql_hash	IN bigint	SQL文本的hash值。	-
plan_hash	IN bigint	SQL计划outline文本的hash值。	-
execute_status	OUT boolean	baseline的加载行为是否正常完成：t表示正常；异常会报错。	t/f

  返回值类型：boolean

- GS_SPM_VALIDATE_PLAN(sql_hash, plan_hash)

  描述：GS_SPM_VALIDATE_PLAN属于DBE_SQL_UTIL schema，它是SPM特性用于计划可用性验证的接口函数。

版权所有 © 华为云计算技术有限公司

参数：参数介绍见**表1-220**。

表 1-220 GS_SPM_VALIDATE_PLAN 入参和返回值列表

参数	类型	描	取值范围
sql_hash	IN bigint	SQL文本的hash值。	-
plan_hash	IN bigint	SQL计划outline文本的hash值。	-
execute_status	OUT boolean	被验证的计划是否可用： t：被验证的计划可用；f: 被验证的计划不可用	t/f

返回值类型：boolean

- GS_SPM_DELETE_PLAN(sql_hash, plan_hash)

  描述：GS_SPM_DELETE_PLAN属于DBE_SQL_UTIL schema，它是SPM特性用于计划删除baseline的接口函数。

  参数： 参数介绍见**表1-221**。

表 1-221 GS_SPM_DELETE_PLAN 入参和返回值列表

参数	类型	描述	取值范围
sql_hash	IN bigint	SQL文本的hash值。	-
plan_hash	IN bigint	SQL计划outline文本的hash值。	-
execute_status	OUT boolean	计划删除行为是否正常完成：t表示正常；异常会报错。	t/f

返回值类型：boolean

1.6.29 统计信息函数

统计信息函数根据访问对象分为两种类型：针对某个数据库进行访问的函数，以数据库中每个表或索引的OID作为参数，标识需要报告的数据库；针对某个服务器进行访问的函数，以一个服务器线程号为参数，其范围从1到当前活跃服务器的数目。

- pg_stat_get_db_conflict_tablespace(oid)

  描述：由于恢复与数据库中删除的表空间发生冲突而取消的查询数。

  返回值类型：bigint

- pg_stat_get_db_conflict_truncate(oid)

  描述：由于与数据库中在回放的truncate操作发生冲突而取消的查询数。

  返回值类型：bigint

- pg_stat_get_db_conflict_force_recycle(oid)

  描述：由于极致RTO备机读触发强制回收而取消的查询数。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

- pg_stat_get_db_conflict_standby_query_timeout(oid)
  描述：由于超过极致RTO备机读设置的查询时间上限而取消的查询数。
  返回值类型：bigint
- pg_stat_get_db_conflict_bufferpin(oid)
  描述：冲突的缓冲区数目。
  返回值类型：bigint
- pg_stat_get_db_conflict_snapshot(oid)
  描述：冲突的快照数目。
  返回值类型：bigint
- pg_stat_get_db_conflict_startup_deadlock(oid)
  描述：冲突的死锁数目。
  返回值类型：bigint
- pg_stat_get_db_conflict_all(oid)
  描述：由于数据库恢复冲突取消的查询数量（只在备用服务器发生的冲突）。
  返回值类型：bigint
- pg_control_group_config()
  描述：在当前节点上打印cgroup配置。该函数需要SYSADMIN权限的用户才能够执行。
  返回值类型：record
- pg_stat_get_db_stat_reset_time(oid)
  描述：上次重置数据库统计信息的时间。查询首次连接的数据库将返回数据库连接时的系统时间。当在数据库上调用pg_stat_reset以及针对其中的任何表或索引执行pg_stat_reset_single_table_counters时，重置时间都会更新。
  返回值类型：timestamptz
- pg_stat_get_function_total_time(oid)
  描述：该函数花费的总挂钟时间，以微秒为单位。包括花费在此函数调用其它函数上的时间。
  返回值类型：bigint
- pg_stat_get_xact_tuples_returned(oid)
  描述：当前事务中参数为表时通过顺序扫描读取的行数，或参数为索引时返回的索引条目数。
  返回值类型：bigint
- pg_stat_get_xact_numscans(oid)
  描述：当前事务中参数为表时执行的顺序扫描次数，或参数为索引时执行的索引扫描次数。
  返回值类型：bigint
- pg_stat_get_xact_blocks_fetched(oid)
  描述：当前事务中对表或索引的磁盘块获取请求数。
  返回值类型：bigint
- pg_stat_get_xact_blocks_hit(oid)
  描述：当前事务中对缓存中找到的表或索引的磁盘块获取请求数。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

- pg_stat_get_xact_function_calls(oid)

  描述：在当前事务中调用该函数的次数。

  返回值类型：bigint

- pg_stat_get_xact_function_self_time(oid)

  描述：在当前事务中仅花费在此函数上的时间。不包括花费在此函数内部调用其它函数上的时间。

  返回值类型：bigint

- pg_stat_get_xact_function_total_time(oid)

  描述：当前事务中该函数所花费的总挂钟时间（以微秒为单位）。包括花费在此函数内部调用其它函数上的时间。

  返回值类型：bigint

- pg_lock_status()

  描述：查询打开事务所持有的锁信息，所有用户均可执行该函数。

  返回值类型：返回字段可参考**PG_LOCKS**视图返回字段，该视图是通过查询本函数得到的结果。

- bucket_lock_status()

  描述：查询打开事务所持有的bucket锁信息，所有用户均可执行该函数。

  返回值类型：返回字段可参考**GS_BUCKET_LOCKS**视图返回字段中除node_name之外的字段。

- gs_bucket_lock_status()

  描述：查询所有节点中的打开事务所持有的bucket锁信息，所有用户均可执行该函数。

  返回值类型：返回字段可参考**GS_BUCKET_LOCKS**视图返回字段，该视图是通过查询本函数得到的结果。

- gs_lwlock_status()

  描述：查询数据库系统内所有轻量级锁信息，包括等锁和持锁信息，所有用户均可执行该函数。

  返回值类型：setof record

- pg_stat_get_wal_senders()

  描述：在主机端查询walsender信息。

  返回值类型：setof record

  返回字段说明如下：

表 1-222 返回字段说明

字段名称	字段类型	字段说明
pid	bigint	walsender的线程号。
sender_pid	integer	walsender的pid相对的轻量级线程号。
local_role	text	主节点类型。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	字段说明
peer_role	text	备节点类型。
peer_state	text	备节点状态。
state	text	walsender状态。
catchup_start	timestamp with time zone	catchup启动时间。
catchup_end	timestamp with time zone	catchup结束时间。
sender_sent_location	text	主节点发送位置。
sender_write_location	text	主节点落盘位置。
sender_flush_location	text	主节点flush磁盘位置。
sender_replay_location	text	主节点redo位置。
receiver_received_location	text	备节点接收位置。
receiver_write_location	text	备节点落盘位置。
receiver_flush_location	text	备节点flush磁盘位置。
receiver_replay_location	text	备节点redo磁盘位置。
sync_percent	text	同步百分比。
sync_state	text	同步状态。
sync_group	text	同步复制的所属分组。
sync_priority	text	同步复制的优先级。
sync_most_available	text	最大可用模式设置。
channel	text	walsender信道信息。

- get_paxos_replication_info()
  描述：查询Paxos模式下主机或备机的复制状态。
  返回值类型：setof record
  返回字段说明如下：

  表 1-223 返回字段说明

字段名称	字段类型	字段说明
paxos_write_location	text	已经写入DCF的Xlog位置。
paxos_commit_location	text	已经在DCF中达成一致的Xlog位置。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	字段说明
local_write_location	text	节点的落盘位置。
local_flush_location	text	节点的flush磁盘位置。
local_replay_location	text	节点的redo磁盘位置。
dcf_replication_info	text	节点的DCF模块信息。字符串为JSON格式，其中各字段含义分别如下： role：表示当前节点角色，角色可分为LEADER、FOLLOWER、LOGGER、PASSIVE、UNKNOW。 term：选举任期。 run_mode：DCF运行模式，当前0表示自动选举模式，1表示手动选举模式，2表示关闭选举模式。 work_mode：DCF工作模式。 hb_interval：DCF节点间心跳间隔时间，单位ms。 elc_timeout：DCF选举超时时间，单位ms。 applied_index：被应用到状态机的日志位置。 commit_index：已被大多数DCF节点保存的日志位置，此commit_index之前日志均已持久化。 first_index：DCF节点保存的首条日志位置，此位置会随着DN调用dcf_truncate而向后推进，之前的日志会被清理。 last_index：DCF节点保存的最后一条日志位置，此日志位置包含DCF节点存储在内存里但是没有持久化的日志。 cluster_min_apply_idx：集群最小已应用的日志位置。 leader_id：leader节点ID。 leader_ip：leader节点IP。 leader_port：leader节点端口，DCF内部使用 。 nodes：本集群其他节点信息，包含节点的id、ip、port、role等关键信息。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# select * from get_paxos_replication_info();
 paxos_write_location | paxos_commit_location | local_write_location | local_flush_location |
local_replay_location |

                                                                                dcf_replication_info



----------------------+----------------------+---------------------+---------------------+-----------------------
+-------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
------------------------------------------------------------------------
 38F/CD54F690         | 38F/CD54F690          | 38F/CD54F690         | 38F/CD546CF8         | 38F/
CD546CF8           | {"stream_id":1,"local_node_id":1,"role":"LEADER","
term":1215,"run_mode":1,"work_mode":0,"stg_mode":1,"hb_interval":1000,"elc_timeout":3000,"auto_el
c_pri_en":1,"elc_switch_thd":0,"group":0,"priority":3898464883456,"lead
er_group":0,"is_in_major":1,"applied_index":362830766,"commit_index":362830766,"first_index":35533
0879,"last_index":3916160068856,"last_term":1215,"dcf_last_index":3628
30766,"dcf_last_disk":362830754,"cluster_min_apply_idx":362830764,"cluster_mode":"primary_cluster",
"leader_id":1,"leader_ip":"*.*.*.*","leader_port":*,"nodes"
:
[{"node_id":1,"ip":"*.*.*.*","port":*,"role":"LEADER","next_index":362830766,"match_index":362830765,"
apply_index":362830765},{"node_id":2,"ip":"*.*.*.*
4","port":*,"role":"FOLLOWER","next_index":362830767,"match_index":362830766,"apply_index":36283
0765},{"node_id":3,"ip":"*.*.*.*","port":*,"role":"LOGGER"
,"next_index":362830767,"match_index":362830766,"apply_index":362830764}]}
(1 row)
```

- pgxc_get_senders_catchup_time()

  描述：在CN实例查询集群中是否存在处于日志追赶状态的备DN，以及追赶状态详情。

  返回值类型：setof record

- pg_stat_get_stream_replications()

  描述：查询主备复制状态。

  返回值类型：setof record

  返回值说明如下表所示。

表 1-224 返回值说明

返回参数	返回参数类型	返回参数说明
local_role	text	本地角色。
static_connections	integer	连接统计。
db_state	text	数据库状态。
detail_information	text	详细信息。

- pg_stat_get_db_numbackends(oid)
  描述：处理该数据库活跃的服务器线程数目。
  返回值类型：integer
- pg_stat_get_db_xact_commit(oid)
  描述：数据库中已提交事务的数量。
  返回值类型：bigint
- pg_stat_get_db_xact_rollback(oid)
  描述：数据库中回滚事务的数量。
  返回值类型：bigint
- pg_stat_get_db_blocks_fetched(oid)
  描述：数据库中磁盘块抓取请求的总数。
  返回值类型：bigint
- pg_stat_get_db_blocks_hit(oid)
  描述：数据库在缓冲区中找到的磁盘块抓取请求的总数。
  返回值类型：bigint
- pg_stat_get_db_tuples_returned(oid)
  描述：数据库返回的Tuple数。
  返回值类型：bigint
- pg_stat_get_db_tuples_fetched(oid)
  描述：数据库中获取的Tuple数。
  返回值类型：bigint
- pg_stat_get_db_tuples_inserted(oid)
  描述：数据库中插入的Tuple数。
  返回值类型：bigint
- pg_stat_get_db_tuples_updated(oid)
  描述：数据库中更新的Tuple数。
  返回值类型：bigint
- pg_stat_get_db_blk_read_time(oid)
  描述：通过数据库后端读取数据文件块花费的时间，以毫秒计算。
  返回值类型：double
- pg_stat_get_db_blk_write_time(oid)
  描述：通过数据库后端写入数据文件块花费的时间，以毫秒计算。
  返回值类型：double
- pg_stat_get_db_tuples_deleted(oid)
  描述：数据库中删除Tuple数。
  返回值类型：bigint
- pg_stat_get_db_conflict_lock(oid)
  描述：数据库中锁冲突的数量。
  返回值类型：bigint
- pg_stat_get_db_deadlocks(oid)

版权所有 © 华为云计算技术有限公司

描述：数据库中死锁的数量。

返回值类型：bigint

- pg_stat_get_numscans(oid)

  描述：如果参数是一个表，则顺序扫描读取的次数。如果参数是一个索引，则返回索引行的扫描次数。

  返回值类型：bigint

- pg_stat_get_role_name(oid)

  描述：根据用户oid获取用户名。仅SYSADMIN和MONADMIN用户可以访问。

  返回值类型：text

  示例：

  ```
  gaussdb=# SELECT pg_stat_get_role_name(10);
   pg_stat_get_role_name
  -----------------------
   aabbcc
  (1 row)
  ```

- pg_stat_get_tuples_returned(oid)

  描述：如果参数是一个表，则顺序扫描读取的活跃行数目。如果参数是一个索引，则返回的活跃索引行的数目。

  返回值类型：bigint

- pg_stat_get_tuples_fetched(oid)

  描述：如果参数是一个表，则位图扫描抓取的行数目。如果参数是一个索引，则用简单索引扫描在原表中抓取的行数目。

  返回值类型：bigint

- pg_stat_get_tuples_inserted(oid)

  描述：表中已插入的行数。

  返回值类型：bigint

- pg_stat_get_tuples_updated(oid)

  描述：在表中已更新行的数量。

  返回值类型：bigint

- pg_stat_get_tuples_deleted(oid)

  描述：从表中删除行的数量。

  返回值类型：bigint

- pg_stat_get_tuples_changed(oid)

  描述：该表上一次analyze或autoanalyze之后插入、更新、删除行的总数量。

  返回值类型：bigint

- pg_stat_get_tuples_hot_updated(oid)

  描述：表热更新的行数。

  返回值类型：bigint

- pg_stat_get_live_tuples(oid)

  描述：表活行数。

  返回值类型：bigint

- pg_stat_get_dead_tuples(oid)

  描述：表死行数。在Ustore表中仅代表不活跃行指针数量。

版权所有 © 华为云计算技术有限公司

返回值类型：bigint

- pg_stat_get_blocks_fetched(oid)
  描述：表或者索引的磁盘块抓取请求的数量。
  返回值类型：bigint
- pg_stat_get_blocks_hit(oid)
  描述：在缓冲区中找到的表或者索引的磁盘块请求数目。
  返回值类型：bigint
- pg_stat_get_xact_tuples_fetched(oid)
  描述：事务中扫描的tuple行数。
  返回值类型：bigint
- pg_stat_get_xact_tuples_inserted(oid)
  描述：表相关的活跃子事务中插入的tuple数。
  返回值类型：bigint
- pg_stat_get_xact_tuples_deleted(oid)
  描述：表相关的活跃子事务中删除的tuple数。
  返回值类型：bigint
- pg_stat_get_xact_tuples_hot_updated(oid)
  描述：表相关的活跃子事务中热更新的tuple数。
  返回值类型：bigint
- pg_stat_get_xact_tuples_updated(oid)
  描述：表相关的活跃子事务中更新的tuple数。
  返回值类型：bigint
- pg_stat_get_last_vacuum_time(oid)
  描述：用户在该表上最后一次手动启动清理或者autovacuum线程启动清理的时间。
  返回值类型：timestamptz
- pg_stat_get_last_autovacuum_time(oid)
  描述：autovacuum守护线程在该表上最后一次启动清理的时间。
  返回值类型：timestamptz
- pg_stat_get_vacuum_count(oid)
  描述：用户在该表上启动清理的次数。
  返回值类型：bigint
- pg_stat_get_autovacuum_count(oid)
  描述：autovacuum守护线程在该表上启动清理的次数。
  返回值类型：bigint
- pg_stat_get_last_analyze_time(oid)
  描述：用户在该表上最后一次手动启动分析或者autovacuum线程启动分析的时间。
  返回值类型：timestamptz
- pg_stat_get_last_autoanalyze_time(oid)
  描述：autovacuum守护线程在该表上最后一次启动分析的时间。

版权所有 © 华为云计算技术有限公司

返回值类型：timestamptz

- pg_stat_get_analyze_count(oid)

  描述：用户在该表上启动分析的次数。

  返回值类型：bigint

- pg_stat_get_autoanalyze_count(oid)

  描述：autovacuum守护线程在该表上启动分析的次数。

  返回值类型：bigint

- gs_stat_get_all_tabentries()

  描述：在GUC参数enable_gsstat_cache设置为on时，用户可使用gs_stat_get_all_tabentries()获取其他节点的PgStat统计信息。内部函数，不建议用户直接使用。

  返回值类型：setof record

  返回参数说明如**表1-225**所示。

表 1-225 返回参数说明

返回参数	返回参数类型	返回参数说明
node_name	name	节点的名称（pgxc_node中的node_name）。
nspname	name	Schema名称。
relname	name	表、索引名称。
tuples_returned	bigint	如果是表，则返回顺序扫描读取的行数目。如果是索引，则返回的索引行的数目。
tuples_fetched	bigint	如果是表，则用位图扫描抓取的行数目。如果是索引，则用简单索引扫描抓取的行数目。
tuples_inserted	bigint	插入表中的行的数量。
tuples_updated	bigint	在表中已更新行的数量。
tuples_deleted	bigint	从表中删除行的数量。
tuples_hot_updated	bigint	热更新的行数表。
n_live_tuples	bigint	活行数表。
n_dead_tuples	bigint	死行数表。在Ustore表中仅代表不活跃行指针数量。
blocks_fetched	bigint	表或者索引的磁盘块抓取请求的数量。

版权所有 © 华为云计算技术有限公司

返回参数	返回参数类型	返回参数说明
blocks_hit	bigint	在缓冲区中找到的表或者索引的磁盘块请求数目。
numscans	bigint	如果是表，则返回顺序扫描读取的行数目。如果是索引，则返回索引行的数目。
autovac_vacuum_timestamp	timestamp with time zone	autovacuum守护进程在该表上最后一次启动清理的时间。
autovac_vacuum_count	bigint	autovacuum守护进程在该表上启动清理的次数。
autovac_analyze_timestamp	timestamp with time zone	autovacuum守护进程在该表上最后一次启动分析的时间。
autovac_analyze_count	bigint	autovacuum守护进程在该表上启动分析的次数。
state_change_timestamp	timestamp with time zone	返回DBE_PERF.stat_all_tables视图中对各表各项监控指标字段更新的时间。
index_stat_change_timestamp	timestamp with time zone	返回DBE_PERF.stat_all_indexes视图中对各表各项监控指标字段更新的时间。
state_io_change_timestamp	timestamp with time zone	返回DBE_PERF.statio_all_indexes, DBE_PERF.statio_all_tables视图中对各表各项监控指标字段更新的时间。

- pg_total_autovac_tuples(bool)

  描述：返回total autovac相关的tuple记录，如nodename、nspname、relname以及各类tuple的IUD信息，入参为：是否查询relation信息。

  返回值类型：setof record

  返回参数说明如下。

版权所有 © 华为云计算技术有限公司

表 1-226 返回参数说明

返回参数	返回参数类型	返回参数说明
nodename	name	节点名称。
nspname	name	名称空间名称。
relname	name	表、索引、视图等对象名称。
partname	name	分区名称。
n_dead_tuples	bigint	表分区内的死行数。在Ustore表中仅代表不活跃行指针数量。
n_live_tuples	bigint	表分区内的活行数。
changes_since_analyze	bigint	analyze产生改变的数量。

- pg_total_gsi_autovac_tuples(bool)

  描述：返回total autovac相关的tuple记录，如nodename、nspname、relname以及GSI自上次analyze之后发生变化的记录数，入参为：是否查询relation信息。该函数用于辅助autovacuum全局二级索引，不建议用户使用。

  返回值类型：setof record

- pg_autovac_status(oid)

  描述：返回和autovac状态相关的参数信息，如nodename、nspname、relname、analyze、vacuum设置，analyze/vacuum阈值，analyze/vacuum tuple数等。仅SYSADMIN可以使用该函数。

  返回值类型：setof record

  返回值参数说明如下。

表 1-227 返回值参数说明

返回参数	返回参数类型	返回参数说明
nspname	text	名称空间名称。
relname	text	表、索引、视图等对象名称。
nodename	text	节点名称。
doanalyze	Boolean	是否执行analyze。
anltuples	bigint	analyze tuple数量。
anlthresh	bigint	analyze阈值。
dovacuum	Boolean	是否执行vacuum。
vactuples	bigint	vacuum tuple数量。

版权所有 © 华为云计算技术有限公司

返回参数	返回参数类型	返回参数说明
vacthresh	bigint	vacuum阈值。

- pg_autovac_timeout(oid)

  描述：返回某个表做autovac连续超时的次数，表信息非法或node信息异常返回NULL。

  返回值类型：bigint

- pg_autovac_coordinator(oid)

  描述：返回对某个表做autovac的coordinator名称，表信息非法或node信息异常返回NULL。

  返回值类型：text

- pg_stat_get_last_data_changed_time(oid)

  描述：对于在表上的修改insert/update/delete/truncate和在表的分区(partition/subpartition)上的修改exchange/truncate/drop，在该表上最后一次操作的时间，**PG_STAT_ALL_TABLES**视图last_data_changed列的数据是通过该函数求值，在表数量很大的场景中，通过视图获取表数据最后修改时间的性能较差，建议直接使用该函数获取表数据的最后修改时间。入参是表的oid。

  返回值类型：timestamptz

- pg_stat_set_last_data_changed_time(oid)

  描述：手动设置该表上最后一次insert/update/delete, exchange/truncate/drop partition操作的时间。

  返回值类型：void

- pg_stat_get_last_updated(oid, text)

  描述：返回DBE_PERF.stat_all_tables, DBE_PERF.stat_all_indexes, DBE_PERF.statio_all_indexes, DBE_PERF.statio_all_tables视图中对各表各项监控指标字段更新的时间。入参1为表的oid，入参2为text类型，取值为"stat_table", "stat_index", "stat_io"。

  返回值类型：timestamptz

- pg_backend_pid()

  描述：当前会话的服务器线程的线程ID。

  返回值类型：integer

- pg_stat_get_activity(integer)

  描述：返回一个关于带有特殊PID的后台线程的记录信息，当参数为NULL时，则返回每个活动的后台线程的记录。返回结果不包含connection_info列。初始用户、系统管理员和MONADMIN可以查看所有的数据，普通用户只能查询自己的结果。

  示例：

```
gaussdb=# SELECT pg_backend_pid();
 pg_backend_pid
-----------------
 139881386280704
(1 row)

-- 查询结果依赖于上方的返回值。
gaussdb=# SELECT * FROM pg_stat_get_activity(139881386280704);
 datid |       pid       | sessionid | usesysid | application_name | state  |
query                      | waiting |         xact_start          |         query_start         |
```


版权所有 © 华为云计算技术有限公司

```
    backend_start         |         state_change          | client_addr | client_hostname | client_port | enqueue
|     query_id      |   srespool   | global_sessionid | unique_sql_id | trace_id
-------+----------------+-----------+----------+-----------------+--------
+-------------------------------------------------+---------+-------------------------------
+--------------------------------+-----
--------------------------+--------------------------------+-------------+-----------------+-------------+---------
+------------------+-------------+-----------------+---------------+----------
 16545 | 139881386280704 |        69 |       10 | gsql            | active | select * from
pg_stat_get_activity(139881386280704); | f       | 2022-01-18 19:43:05.167718+08 | 2022-01-18
19:43:05.167718+08 | 2022
-01-18 19:42:33.513507+08 | 2022-01-18 19:43:05.16773+08 |             |                |         -1 |         |
72620543991624410 | default_pool | 1938253334#69#0  |    3751941862 |
(1 row)
```

返回值类型：setof record

返回参数说明如下。

表 1-228 返回参数说明

返回参数	返回参数类型	返回参数说明
datid	oid	用户会话在后台连接到的数据库OID。
pid	bigint	后台线程ID。
sessionid	bigint	会话ID。
usesysid	oid	登录该后台的用户OID。
application_name	text	连接到该后台的应用名。
state	text	该后台当前总体状态。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。
waiting	Boolean	如果后台当前正等待锁则为true。
xact_start	timestamp with time zone	启动当前事务的时间，如果没有事务是活跃的，则为null。 如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	开始当前活跃查询的时间，如果state的值不是active，则这个值是上一个查询的开始时间。

版权所有 © 华为云计算技术有限公司

返回参数	返回参数类型	返回参数说明
backend_start	timestamp with time zone	该过程开始的时间，即当客户端连接服务器时。
state_change	timestamp with time zone	上次状态改变的时间。
client_addr	inet	连接到该后台的客户端的IP地址。如果此字段是null，它表明通过服务器机器上UNIX套接字连接客户端或者这是内部线程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
enqueue	text	该字段暂不支持。
query_id	bigint	查询语句的ID。
srespool	name	资源池名字。
global_sessionid	text	全局会话id。
unique_sql_id	bigint	语句的unique sql id。
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。

- pg_stat_get_activity_with_conninfo(integer)

  描述：返回一个关于带有特殊PID的后台线程的记录信息，当参数为NULL时，则返回每个活动的后台线程的记录。初始用户、系统管理员和MONADMIN可以查看所有的数据，普通用户只能查询自己的结果。

  返回值类型：setof record

  返回值说明如下。

返回值	返回值类型	返回值说明
datid	oid	用户会话在后台连接到的数据库OID。
pid	bigint	后台线程ID。

版权所有 © 华为云计算技术有限公司

返回值	返回值类型	返回值说明
sessionid	bigint	会话ID。
usesysid	oid	登录该后台的用户OID。
application_name	text	连接到该后台的应用名。
state	text	该后台当前总体状态。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。
waiting	Boolean	如果后台当前正等待锁则为true。
xact_start	timestamp with time zone	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	开始当前活跃查询的时间，如果state的值不是active，则这个值是上一个查询的开始时间。
backend_start	timestamp with time zone	该过程开始的时间，即当客户端连接服务器时。
state_change	timestamp with time zone	上次状态改变的时间。
client_addr	inet	连接到该后台的客户端的IP地址。如果此字段是null，它表明通过服务器机器上UNIX套接字连接客户端或者这是内部线程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。

版权所有 © 华为云计算技术有限公司

返回值	返回值类型	返回值说明
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
enqueue	text	该字段暂不支持。
query_id	bigint	查询语句的ID。
connection_info	text	json格式字符串，记录当前连接数据库的驱动类型、驱动版本号、当前驱动的部署路径、线程属主用户等信息。
srespool	name	资源池名字。
global_sessionid	text	全局会话ID。
unique_sql_id	bigint	语句的unique sql id。
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。
top_xid	xid	事务的顶层事务ID。
current_xid	xid	事务的当前事务ID。
xlog_quantity	bigint	事务当前使用的Xlog量，单位为字节。
control_status	text	当进行异常SQL CPU资源管控时为under_cpu_throttle，否则为normal。

- gs_get_explain(integer)

  描述：返回一个关于指定PID的后台线程的运行态计划，PID不能为空。该功能生效的必要条件是GUC参数track_activities为on，支持语句的范围为可以explain且其计划不包含STREAM算子的SQL。详细说明如下：

  - 若GUC参数plan_collect_thresh=-1，则该函数的返回结果一直为空。
  - 若GUC参数plan_collect_thresh=0，若当前SQL执行时间大于等于GUC参数log_min_duration_statement指定值，且计划中所有算子处理的tuple数量和大于等于10000时，开始收集运行态计划；且后续每当所有算子处理tuple数量的和的增量超过10000时，进行一次收集。
  - 若GUC参数plan_collect_thresh>0，会按照该参数指定的阈值进行增量收集运行态计划。
  - 另外，该函数只能看到在CN生成的计划，若查看在DN生成的计划，需要参考函数gs_get_dn_explain。

  返回值类型：text，具体各个字段的类型和含义如下：

版权所有 © 华为云计算技术有限公司

返回值	返回值类型	返回值说明
运行态计划字符串	text	其中计划字符串中的A-rows是算子实时返回行数。

- gs_get_dn_explain(text, bigint)

  描述：返回一个关于指定node name（来自pgxc_node系统表的node_name字段）和global session id的DN的后台线程的运行态计划，global session id不能为空。该功能生效的必要条件是GUC参数track_activities为on，支持语句的范围为可以explain且其计划不包含STREAM算子的SQL。详细说明如下：

  - 同函数gs_get_explain，其中global session id是由pg_stat_activity字段global_sessionid计算而来。global_sessionid格式如下：${number1}:${number2}#{number3}。需要注意的是number1为有符号数，当number1为正数时，如global_sessionid样例为1938253334:1#0，那么gs_get_dn_explain函数的参数值应为${number2}*10000000000+${number1}；当number1为负数时，需要将其转换为32位无符号数，如global_sessionid样例为-1736975100:1#0，那么gs_get_dn_explain函数的参数值应为${number2}*10000000000+(${number1}+4294967296)。
  - 另外该函数需要ADMIN或MONADMIN权限才可执行。

  返回值类型：text，具体各个字段的类型和含义如下：

返回值	返回值类型	返回值说明
运行态计划字符串	text	其中计划字符串中的A-rows是算子实时返回行数。

- pg_stat_get_activity_ng(integer)

  描述：返回一个关于带有特殊PID的活跃后台线程记录信息，当参数为NULL时，则返回每个活跃的后台线程的记录。系统管理员和MONADMIN可以查看所有的数据，普通用户只能查询自己的结果。

  返回值类型：setof record

  函数返回字段说明如下：

名称	类型	描述
datid	oid	数据库oid。
pid	bigint	后端线程的ID。
sessionid	bigint	会话的id。
node_group	text	数据所属用户对应的node group。

- pg_stat_get_function_calls(oid)

  描述：函数已被调用次数。

  返回值类型：bigint

版权所有 © 华为云计算技术有限公司

- pg_stat_get_function_self_time(oid)

  描述：只有在此函数所花费的时间。此函数嵌套调用其它函数所花费的时间被排除在外。

  返回值类型：bigint

- pg_stat_get_backend_idset()

  描述：设置当前活动的服务器线程数（从1到活动服务器线程的数量）。

  返回值类型：setofinteger

- pg_stat_get_backend_pid(integer)

  描述：给定的服务器线程的线程ID。

  返回值类型：bigint

- pg_stat_get_backend_dbid(integer)

  描述：给定服务器线程的数据库ID。

  返回值类型：oid

- pg_stat_get_backend_userid(integer)

  描述：给定服务器线程的用户ID，本函数仅系统管理员可调用。

  返回值类型：oid

- pg_stat_get_backend_activity(integer)

  描述：给定服务器线程的当前活动查询，仅在调用者是系统管理员或被查询会话的用户，并且打开track_activities的时候才能获得结果。

  返回值类型：text

- pg_stat_get_backend_waiting(integer)

  描述：如果给定服务器线程在等待某个锁，并且调用者是系统管理员或被查询会话的用户，并且打开track_activities的时候才返回真。

  返回值类型：Boolean

- pg_stat_get_backend_activity_start(integer)

  描述：给定服务器线程当前正在执行的查询的起始时间，仅在调用者是系统管理员或被查询会话的用户，并且打开track_activities的时候才能获得结果。

  返回值类型：timestampwithtimezone

- pg_stat_get_backend_xact_start(integer)

  描述：给定服务器线程当前正在执行的事务的开始时间，但只有当前用户是系统管理员或被查询会话的用户，并且打开track_activities的时候才能获得结果。

  返回值类型：timestampwithtimezone

- pg_stat_get_backend_start(integer)

  描述：给定服务器线程启动的时间，如果当前用户不是系统管理员或被查询的后端的用户，则返回NULL。

  返回值类型：timestampwithtimezone

- pg_stat_get_backend_client_addr(integer)

  描述：连接到给定客户端后端的IP地址。如果是通过Unix域套接字连接的则返回NULL；如果当前用户不是系统管理员或被查询会话的用户，也返回NULL。

  返回值类型：inet

- pg_stat_get_backend_client_port(integer)

版权所有 © 华为云计算技术有限公司

描述：连接到给定客户端后端的TCP端口。如果是通过Unix域套接字连接的则返回-1；如果当前用户不是系统管理员或被查询会话的用户，也返回NULL。

返回值类型：integer

- pg_stat_get_bgwriter_timed_checkpoints()

  描述：后台写线程开启定时检查点的时间（因为checkpoint_timeout时间已经过期了）。

  返回值类型：bigint

- pg_stat_get_bgwriter_requested_checkpoints()

  描述：后台写线程开启基于后端请求的检查点的时间，因为已经超过了checkpoint_segments或因为已经执行了CHECKPOINT。

  返回值类型：bigint

- pg_stat_get_bgwriter_buf_written_checkpoints()

  描述：在检查点期间后台写线程写入的缓冲区数目。

  返回值类型：bigint

- pg_stat_get_bgwriter_buf_written_clean()

  描述：为日常清理脏块，后台写线程写入的缓冲区数目。

  返回值类型：bigint

- pg_stat_get_bgwriter_maxwritten_clean()

  描述：后台写线程停止清理扫描的时间，因为已经写入了更多的缓冲区（相比bgwriter_lru_maxpages参数声明的缓冲区数）。

  返回值类型：bigint

- pg_stat_get_buf_written_backend()

  描述：后端线程写入的缓冲区数，因为它们需要分配一个新的缓冲区。

  返回值类型：bigint

- pg_stat_get_buf_alloc()

  描述：分配的总缓冲区数。

  返回值类型：bigint

- pg_stat_clear_snapshot()

  描述：清理当前的统计快照。该函数仅SYSADMIN和MONADMIN可以执行。

  返回值类型：void

- pg_stat_reset()

  描述：为当前数据库重置统计计数器为0（需要系统管理员权限）。

  返回值类型：void

- gs_stat_reset()

  描述：将各节点上的为当前数据库重置统计计数器为0（需要系统管理员权限）。

  返回值类型：void

- pg_stat_reset_shared(text)

  描述：重置shared cluster每个节点当前数据统计计数器为0（需要系统管理员权限）。

  返回值类型：void

- pg_stat_reset_single_table_counters(oid)

 版权所有 © 华为云计算技术有限公司

描述：为当前数据库中的一个表或索引重置统计为0（需要系统管理员权限）。

返回值类型：void

- pg_stat_reset_single_function_counters(oid)

  描述：为当前数据库中的一个函数重置统计为0（需要系统管理员权限）。

  返回值类型：void

- pgxc_fenced_udf_process(integer)

  描述：查看UDF Master和Work线程数，仅SYSADMIN和MONADMIN用户有权限执行。入参为1时查看master线程数，入参为2时查看worker线程数，入参为3时终止所有worker线程。

  返回值类型：text

- fenced_udf_process(integer)

  描述：查看本地UDF Master和Work进程数。仅SYSADMIN和MONADMIN用户有权限终止所有worker进程。入参为1时查看master进程数，入参为2时查看worker进程数，入参为3时终止所有worker进程。

  返回值类型：text

- total_cpu()

  描述：获取当前节点使用的cpu时间，单位是jiffies。

  返回值类型：bigint

- total_memory()

  描述：获取当前节点使用的虚拟内存大小，单位KB。

  返回值类型：bigint

- pgxc_terminate_all_fenced_udf_process()

  描述：Kill所有的UDF Work线程，仅SYSADMIN和MONADMIN用户有权限执行。

  返回值类型：boolean

- gs_get_nodegroup_tablecount(name)

  描述：得到一个节点组中所有数据库包含的用户表数目。

  返回值类型：integer

- pgxc_max_datanode_size(name)

  描述：得到一个node_group的所有DN节点中数据库文件占用磁盘空间的最大值，单位为字节。

  返回值类型：bigint

- gs_check_tables_distribution()

  描述： 检查当前系统中用户表的分布是否存在不一致，如果返回空记录，表示不存在不一致。该函数应该在非扩缩容重分布时调用。

  返回值类型：record

- pg_stat_bad_block()

  描述：获取当前节点自启动后，读取出现Page的损坏信息。

  返回值类型：record

- pgxc_stat_bad_block()

  描述：获取集群所有节点自启动后，读取出现Page的损坏信息。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- pg_stat_bad_block_clear()

  描述：清理节点记录的读取出现的Page损坏信息（需要系统管理员权限）。

  返回值类型：void

- pgxc_stat_bad_block_clear

  描述：清理集群所有节点记录的读取出现的Page损坏信息（需要系统管理员权限）。

  返回值类型：void

- pgxc_log_comm_status(void)

  描述：当使用sctp（由于规格变更，当前版本已经不再支持本特性，请不要使用）通信模式时，PGXC系统视图将datanode的通信层状态输出到各个日志文件中。

  返回值类型：void

- gs_respool_exception_info(pool text)

  描述：查看某个资源池关联的查询规则信息。

  返回值类型：record

- gs_control_group_info(pool text)

  描述：查看资源池关联的控制组信息。该函数需要SYSADMIN权限的用户才能够执行。

  返回值类型：record

  返回信息如下：

属性	属性值	描述
name	class_a:workload_a1	class和workload名称。
class	class_a	Class控制组名称。
workload	workload_a1	Workload控制组名称。
type	DEFWD	控制组类型（Top、CLASS、BAKWD、DEFWD、TSWD）。
gid	87	控制组id。
shares	30	占父节点CPU资源的百分比。
limits	0	占父节点CPU核数的百分比。
rate	0	Timeshare中的分配比例。
cpucores	0-3	CPU核心数。

- gs_all_control_group_info()

  描述：查看数据库内所有的控制组信息。函数返回信息具体的字段参考GS_ALL_CONTROL_GROUP_INFO字段。

  返回值类型：record

- gs_get_control_group_info()

版权所有 © 华为云计算技术有限公司

描述：查看所有的控制组信息。函数返回信息具体的字段参考**GS_GET_CONTROL_GROUP_INFO**。该函数需要SYSADMIN权限的用户才能够执行。

返回值类型：record

- gs_get_session_sql_memory()

  描述：查看当前运行中语句的内存使用量。函数返回信息具体字段参考**DBE_PERF.SESSION_SQL_MEMORY**字段。

  返回值类型：record

- gs_get_global_session_sql_memory()

  描述：查看当前所有节点上运行中语句的内存使用量。函数返回信息具体字段参考**DBE_PERF.GLOBAL_SESSION_SQL_MEMORY**字段。

  返回值类型：record

- gs_blackbox_dump()

  描述：将系统中的黑匣子文件导出到“$GAUSSLOG/gs_blackbox/{nodename}”目录中。执行成功后返回在“$GAUSSLOG/gs_blackbox/{nodename}”中生成的黑匣子文件的名称。

  返回值类型：text

- gs_blackbox_show(blackBoxFile cstring)

  描述：解析黑匣子文件，并按照表格方式展示。

  参数：

参数名称	参数说明	取值范围
blackBoxFile	指定需要解析的黑匣子文件，不指定参数表示当前进程采集的黑匣子文件。	字符串，或者不指定，指定有效的黑匣子文件名字，则读取“$GAUSSLOG/gs_blackbox/nodename”目录下对应名称的文件。仅支持名称，不支持路径，如果在“$GAUSSLOG/gs_blackbox/nodename”下不存在对应名称的文件，则报错返回。

  返回值类型：setof record

  返回值说明如下：

返回值	返回值类型	返回值说明
type	text	黑匣子字段类型。
tid	bigint	线程号。
sample_time	timestamp	采集时间。

版权所有 © 华为云计算技术有限公司

返回值	返回值类型	返回值说明
value	text	黑匣子记录的内容。

查询结果举例如下：

type	tid	sample_time	value
ERRNUMBER	123456	2024-03-19 09:33:17.386	19460
BLACK_BOX_PLSQL_EXCEPTION	45678	2024-03-19 09:33:18.396	message: "Dropped rowtype entry for non-dropped column when make tuple." funcname: basic_make_tuple_from_row filename:pl_exec.cpp context:xxxx

- gs_blackbox_list()

  描述：列举当前实例黑匣子目录下的黑匣子文件。默认是“$GAUSSLOG/gs_blackbox/{nodename}”目录下。

  返回值类型：setof record

  返回值说明如下：

返回值	返回值类型	返回值说明
file	text	黑匣子文件名称。

- get_instr_workload_info(integer)

  描述：获取当前CN节点上事务量信息，事务时间信息。

  返回值类型：record

属性	类型	描述
user_oid	oid	用户ID。
commit_counter	bigint	前端事务commit数量。

版权所有 © 华为云计算技术有限公司

属性	类型	描述
rollback_counter	bigint	前端事务rollback数量。
resp_min	bigint	前端事务最小响应时间（单位：微秒）。
resp_max	bigint	前端事务最大响应时间（单位：微秒）。
resp_avg	bigint	前端事务平均响应时间（单位：微秒）。
resp_total	bigint	前端事务总响应时间（单位：微秒）。
bg_commit_counter	bigint	后端事务commit数量。
bg_rollback_counter	bigint	后端事务rollback数量。
bg_resp_min	bigint	后端事务最小响应时间（单位：微秒）。
bg_resp_max	bigint	后端事务最大响应时间（单位：微秒）。
bg_resp_avg	bigint	后端事务平均响应时间（单位：微秒）。
bg_resp_total	bigint	后端事务总响应时间（单位：微秒）。
dist_commit_counter	bigint	分布式事务commit数量。
dist_rollback_counter	bigint	分布式事务rollback数量。
dist_resp_min	bigint	分布式事务最小响应时间（单位：微秒）。
dist_resp_max	bigint	分布式事务最大响应时间（单位：微秒）。
dist_resp_avg	bigint	分布式事务平均响应时间（单位：微秒）。
dist_resp_total	bigint	分布式事务总响应时间（单位：微秒）。
dbid	oid	数据库标识。

说明

只统计主事务信息，子事务信息不会被统计在内。

版权所有 © 华为云计算技术有限公司

- pv_instance_time()

  描述：获取当前节点上各个关键阶段的时间消耗。

  返回值类型：record

Stat_name属性	属性值	描述
DB_TIME	1062385	所有线程端到端的时间消耗总和（单位：微秒）。
CPU_TIME	311777	所有线程CPU时间消耗总和（单位：微秒）。
EXECUTION_TIME	380037	消耗在执行器上的时间总和（单位：微秒）。
PARSE_TIME	6033	消耗在SQL解析上的时间总和（单位：微秒）。
PLAN_TIME	173356	消耗在执行计划生成上的时间总和（单位：微秒）。
REWRITE_TIME	2274	消耗在查询重写上的时间总和（单位：微秒）。
PL_EXECUTION_TIME	0	消耗在plpgsql执行上的时间总和（单位：微秒）。
PL_COMPILATION_TIME	557	消耗在SQL编译上的时间总和（单位：微秒）。
NET_SEND_TIME	1673	消耗在网络发送上的时间总和（单位：微秒）。
DATA_IO_TIME	426622	消耗在数据读写上的时间总和（单位：微秒）。

- DBE_PERF.get_global_instance_time()

  描述：提供整个集群各个关键阶段的时间消耗，仅在CN上支持查询。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- get_instr_unique_sql()

  描述：获取当前节点的执行语句（归一化SQL）信息，具有SYSADMIN和MONADMIN权限的用户可查询全部信息，其他用户只能查询自己执行的语句的信息。函数返回信息具体字段参考**DBE_PERF.STATEMENT**字段。

  返回值类型：record

- get_instr_wait_event(integer)

  描述：获取当前节点event等待的统计信息。

  返回值类型：record

- get_instr_user_login()

  描述：获取当前节点的用户登录或者退出登录次数信息，查询该函数必须具有SYSADMIN或者MONADMIN权限。

版权所有 © 华为云计算技术有限公司

返回值类型：record

- get_instr_rt_percentile(integer)

  描述：获取CCN节点SQL 响应时间P80、P95分布信息，集群统一的信息在CCN节点上，其他节点查询为0。

  返回值类型：record

- get_node_stat_reset_time()

  描述：获取当前节点的统计信息重置（重启，主备倒换，数据库删除）时间。

  返回值类型：record

- gs_paxos_stat_replication()

  描述：在主机端查询备机信息。

  返回值类型：setof record

  返回字段说明如下：

字段名称	字段类型	字段说明
local_role	text	发送日志节点的角色。
peer_role	text	接收日志节点的角色。
local_dcf_role	text	发送日志节点的DCF角色。
peer_dcf_role	text	接收日志节点的DCF角色。
peer_state	text	接收日志节点的状态。
sender_write_location	text	发送日志节点写到Xlog buffer的位置。
sender_commit_location	text	发送日志节点DCF日志达成一致性点。
sender_flush_location	text	发送日志节点写到Xlog disk的位置。
sender_replay_location	text	发送日志节点replay的位置。
receiver_write_location	text	接收日志节点写到Xlog buffer的位置。
receiver_commit_location	text	接收日志节点DCF日志达成一致性点。
receiver_flush_location	text	接收日志节点写到Xlog disk的位置。
receiver_replay_location	text	接收日志节点重演Xlog的位置。
sync_percent	text	同步百分比。
dcf_run_mode	int4	DCF同步模式。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	字段说明
channel	text	信道信息。

- get_paxos_replication_info()

  描述：查询Paxos模式下主机或备机的复制状态。

  返回值类型：setof record

  返回字段说明如下：

字段名称	字段类型	字段说明
paxos_write_location	text	已经写入DCF的Xlog位置。
paxos_commit_location	text	已经在DCF中达成一致的Xlog位置。
local_write_location	text	节点的落盘位置。
local_flush_location	text	节点的flush磁盘位置。
local_replay_location	text	节点的redo磁盘位置。
dcf_replication_info	text	节点的DCF模块信息。

- gs_wlm_get_user_info(int)

  描述：获取所有用户的相关信息，入参为int类型，可以为任意int值或NULL。该函数只有SYSADMIN权限的用户可以执行。

  返回值类型：record

- gs_wlm_readjust_user_space(oid)

  描述：修正指定用户的存储空间使用情况。普通用户仅可修正自己的使用情况，管理员用户可以修正所有用户的使用情况。当参数oid传入0时，表示需要修正所有用户的使用情况。

  返回值类型：record

- gs_wlm_readjust_user_space_through_username(text name)

  描述：修正指定用户的存储空间使用情况。该函数普通用户只能修正自己的使用情况，只有管理员用户可以修正所有用户的使用情况。当name指定为“0000”，表示需要修正所有用户的使用情况。

  返回值类型：record

- gs_wlm_readjust_user_space_with_reset_flag(text name, boolean isfirst)

  描述：修正指定用户的存储空间使用情况。入参isfirst为true表示从0开始统计，否则从上一次结果继续统计。该函数普通用户只能修正自己的使用情况，只有管理员用户可以修正所有用户的使用情况。当name指定为“0000”，表示需要修正所有用户的使用情况。

版权所有 © 华为云计算技术有限公司

返回值类型：record

- gs_io_wait_status()

  描述：返回当前节点I/O管控的实时统计信息。

  返回值类型：setof record

名称	类型	描述
node_name	text	节点名称。
device_name	text	节点挂载的数据磁盘名称。
read_per_second	float	读完成每秒次数。
write_per_second	float	写完成每秒次数。
write_ratio	float	写磁盘占总的I/O使用的比例。
io_util	float	每秒I/O所占CPU总时间的百分比。
total_io_util	integer	过去三次I/O所占CPU总时间的等级（取值为0~6）。
tick_count	integer	更新磁盘IO信息的周期，固定为1秒，每次读取数据前都会被清零。
io_wait_list_len	integer	I/O请求线程等待队列的大小，若为0，则表示当前没有I/O被管控。

- gs_get_shared_memctx_detail(text)

  描述：返回指定内存上下文上的内存申请的详细信息，包含每一处内存申请所在的文件、行号和大小（同一文件同一行大小会做累加）。只支持查询通过PG_SHARED_MEMORY_DETAIL视图（请参见**PG_SHARED_MEMORY_DETAIL**）查询出来的内存上下文，入参为内存上下文名称（即PG_SHARED_MEMORY_DETAIL视图返回结果的contextname列）。查询该函数必须具有SYSADMIN权限或者MONADMIN权限。

  返回值类型：setof record

名称	类型	描述
file	text	申请内存所在文件的文件名。
line	int8	申请内存所在文件的代码行号。
size	int8	申请的内存大小，同一文件同一行多次申请会做累加。

- gs_get_session_memctx_detail(text)

  描述：返回指定内存上下文上的内存申请的详细信息，包含每一处内存申请所在的文件、行号和大小（同一文件同一行大小会做累加）。仅在线程池模式下生效。且只支持查询通过PV_SESSION_MEMORY_CONTEXT视图（请参见**PV_SESSION_MEMORY_CONTEXT**）查询出来的内存上下文，入参为内存上下

版权所有 © 华为云计算技术有限公司

文名称（即PV_SESSION_MEMORY_CONTEXT视图返回结果的contextname列）。查询该函数必须具有SYSADMIN权限或者MONADMIN权限。

返回值类型：setof record

名称	类型	描述
file	text	申请内存所在文件的文件名。
line	int8	申请内存所在文件的代码行号。
size	int8	申请的内存大小，单位为byte，同一文件同一行多次申请会做累加。

说明

该函数仅在线程池模式下生效。

- gs_get_thread_memctx_detail(tid,text)

  描述：返回指定内存上下文上的内存申请的详细信息，包含每一处内存申请所在的文件、行号和大小（同一文件同一行大小会做累加）。只支持查询通过PV_THREAD_MEMORY_CONTEXT视图（请参见**PV_THREAD_MEMORY_CONTEXT**）查询出来的内存上下文，第一个入参为线程id（即PV_THREAD_MEMORY_CONTEXT视图返回数据的tid列），第二个参数为内存上下文名称（即PV_THREAD_MEMORY_CONTEXT视图返回数据的contextname列）。查询该函数必须具有SYSADMIN权限或者MONADMIN权限。

  返回值类型：setof record

名称	类型	描述
file	text	申请内存所在文件的文件名。
line	int8	申请内存所在文件的代码行号。
size	int8	申请的内存大小，单位为byte，同一文件同一行多次申请会做累加。

- gs_get_history_memory_detail(cstring)

  描述：查询历史内存快照信息，入参类型为cstring，取值为NULL或内存快照log文件名称：

  a. 若入参为NULL，则显示当前节点所有的内存快照log文件列表。
  b. 若入参为**a**查询到的列表中的内存快照log名称，则显示该log文件记录的内存快照详细信息。
  c. 若输入其他入参，则会提示入参错误或打开文件失败。

  查询该函数必须具有SYSADMIN权限或者MONADMIN权限。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memory_info	text	内存信息，如果函数入参为NULL，该列显示内存快照文件列表信息；入参为内存快照文件名称，则显示该文件的具体内容。

- gs_stat_get_hotkeys_info()

  说明

  若GUC参数enable_hotkeys_collection = off, gs_stat_get_hotkeys_info、global_stat_get_hotkeys_info函数和global_stat_hotkeys_info视图无法正常查询。不影响gs_stat_clean_hotkeys和global_stat_clean_hotkeys清理接口的正常使用。

  描述：获取当前节点上热点key的统计情况。

  返回值类型：record

```
gaussdb=# SELECT * FROM gs_stat_get_hotkeys_info() order by count, hash_value;
 database_name | schema_name |     table_name     | key_value | hash_value | count
---------------+-------------+--------------------+-----------+------------+-------
 regression    | public      | hotkey_single_col | {22}      | 1858004829 |     2
 regression    | public      | hotkey_single_col | {11}      | 2011968649 |     2
(2 rows)
```

名称	类型	描述
database_name	text	热点key所在database名称。
schema_name	text	热点key所在schema名称。
table_name	text	热点key所在table名称。
key_value	text	热点key的value。
hash_value	bigint	热点key在数据库中的哈希值，如果是List/Range分布表，该字段为0。
count	bigint	热点key被访问频次。

- gs_stat_clean_hotkeys()

  说明

  - 热点key检测是针对大并发大流量场景设计的特性，访问几次的场景，查询清理会存在一定误差。
  - 清理接口的设计上，只会清理LRU队列中的统计数据，而不会清理FIFO中的历史数据。因此如果清理完后，再访问一次FIFO中存在的历史键值，仍会被当做热点key处理。global_stat_clean_hotkeys同理。

  描述：清理当前节点上热点key的统计信息。

  返回值：boolean

```
gaussdb=# SELECT * FROM gs_stat_clean_hotkeys();
 gs_stat_clean_hotkeys
-----------------------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

- gs_stat_progress_vacuum(pid)

  描述：用于查看当前VACUUM的进度。仅统计Astore和Ustore的页式表的VACUUM进度，不支持统计hashbucket和段页式。

  返回值类型：record

  表 1-229 返回值参数说明

名称	类型	描述
pid	bigint	VACUUM线程的线程pid。
dataname	text	正在清理的数据库名称。
relid	oid	正在清理表的OID。
relname	text	正在清理的表名称。
phase	text	当前VACUUM所处的阶段。
detail	text	清理进度详情，包含总页面数、已扫描页面数、已清理页面数和已清理索引数。

  表 1-230 phase 取值详细说明

phase	字段类型	说明
invalid	text	VACUUM正在启动，此时将获取表锁。
initialize	text	VACUUM正在准备开始扫描堆表，该阶段预计非常短暂。
vacuuming main part	text	VACUUM正在清理分区主表（通常为清理分区主表的GPI），可通过detail列信息查看清理进度。
scanning heap	text	VACUUM正在扫描堆表，必要时会对每个页面进行碎片整理或执行冻结操作，可通过detail列信息查看清理进度。
vacuuming indexes	text	VACUUM正在清理索引。若表存在索引，该阶段至少发生一次，并可能在按需对索引进行多次清理后，才会返回继续扫描堆表，可通过detail列信息查看清理进度。
vacuuming heap	text	VACUUM正在对堆表进行清理（区别于扫描堆表），并且在每次索引清理后执行。若堆表未完全扫描，将在此阶段后返回继续扫描，可通过detail列信息查看清理进度。
vacuuming part cache	text	VACUUM正在清理已缓存分区的死元组，可通过detail列信息查看清理进度。
cleaning up indexes	text	VACUUM正在清理索引，该阶段在堆表完全扫描且所有索引与堆表清理完成后执行。

版权所有 © 华为云计算技术有限公司

phase	字段类型	说明
truncating heap	text	VACUUM正在截断堆表的尾部空页，并将空页返回操作系统。
vacuuming urq	text	VACUUM正在回收Ustore表的索引的页面。
unknown phase	text	表示当前处于未在上述阶段中定义的未知状态，通常不会出现。

```
--设置guc参数：
gaussdb=# SET vacuum_cost_delay=100;
SET
gaussdb=# CREATE TABLE t1 (a int, b int, c text) with (storage_type=astore)
gaussdb-# partition by range(a) (
gaussdb(# partition p1 values less than(10000),
gaussdb(# partition p2 values less than(20000),
gaussdb(# partition p3 values less than(30000),
gaussdb(# partition p4 values less than(40000),
gaussdb(# partition p5 values less than(50000),
gaussdb(# partition p6 values less than(60000),
gaussdb(# partition p7 values less than(70000),
gaussdb(# partition p8 values less than(80000),
gaussdb(# partition p9 values less than(90000),
gaussdb(# partition p10 values less than(maxvalue));
CREATE TABLE
gaussdb=# CREATE INDEX idx11 on t1(a) global;
CREATE INDEX
gaussdb=#
gaussdb=# CREATE INDEX idx12 on t1(b) global;
CREATE INDEX
gaussdb=#
gaussdb=# CREATE INDEX idx13 on t1(c) global;
CREATE INDEX
gaussdb=# INSERT INTO t1 VALUES (generate_Series(1,100000), generate_series(1,3), 'abcde');
INSERT 0 300000
gaussdb=# DELETE t1;
DELETE 300000
gaussdb=# VACUUM t1;
VACUUM
--新起一个会话连接当前节点，执行如下
gaussdb=#  SELECT pid,query FROM pg_stat_activity where query = 'vacuum t1;';
       pid       |   query
-----------------+------------
 140242746078976 | vacuum t1;
(1 row)
gaussdb=# SELECT * FROM gs_stat_progress_vacuum(140242746078976);
       pid       | dataname | relid | relname |      phase
|                                                                                             detail

-----------------+----------+-------+---------+----------------
+-----------------------------------------------------------------------------------------------------------------------
-------------------------------------------
------------------------------------------------------------------------------------------------------------------
 140242746078976 | postgres | 24962 | t1       | vacuuming heap | vacuum command "vacuum t1;"
starts at 2025-09-09 17:20:00.659862-04, vacuumed 0/11 partitions, current part "p10", current phase
already run 0s; heap: scanned 1
63/163 blocks, skipped 0 blocks, vacuumed 53/163 blocks; info: found 30003 dead tuples, already run
47s, delay 47s.
(1 row)

gaussdb=# DROP TABLE t1;
```

- global_stat_get_hotkeys_info()

  说明

  执行业务过程中，执行select * from global_stat_hotkeys_info minus select * from global_stat_get_hotkeys_info();因为存在时间差，可能出现不为0的情况。

  描述：获取整个集群中热点key的统计情况。

  返回值类型：record

```
gaussdb=# SELECT * FROM global_stat_get_hotkeys_info() order by count, hash_value;
 database_name | schema_name |     table_name      | key_value | hash_value | count
---------------+-------------+---------------------+-----------+------------+-------
 regression    | public      | hotkey_single_col | {22}      | 1858004829 |     2
 regression    | public      | hotkey_single_col | {11}      | 2011968649 |     2
(2 rows)
```

- global_stat_clean_hotkeys()

  描述：清理整个集群中热点key的统计信息。

  返回值：boolean

```
gaussdb=# SELECT * FROM global_stat_clean_hotkeys();
 global_stat_clean_hotkeys
-----------------------
 t
(1 row)
```

- global_comm_get_recv_stream()

  描述：获取所有DN节点上所有的通信库接收流状态。函数返回信息具体字段参考**PG_COMM_RECV_STREAM**字段。

  返回值类型：record

- global_comm_get_send_stream()

  描述：获取所有DN节点上所有的通信库发送流状态。函数返回信息具体字段参考**PG_COMM_SEND_STREAM**字段。

  返回值类型：record

- global_comm_get_status()

  描述：获取所有DN节点的通信库状态。函数返回信息具体字段参考**PG_COMM_STATUS**字段。

  返回值类型：record

- global_comm_client_info()

  描述：获取全局节点活跃的客户端连接信息。函数返回信息具体字段参考**COMM_CLIENT_INFO**字段。

  返回类型：record

- global_comm_get_client_info()

  描述：获取全局节点客户端连接信息。函数返回信息具体字段参考**COMM_CLIENT_INFO**字段。

  返回类型：record

- pgxc_stat_activity()

  描述：显示当前集群下所有CN的当前用户查询相关的信息，该函数仅具有SYSADMIN或者MONADMIN权限的用户可以执行，普通用户可查看本用户的相关信息。505.0.0版本新增三列信息，灰度升级观察期中执行会报错，需使用PG_STAT_ACTIVITY视图（请参见**PG_STAT_ACTIVITY**）直连目标节点进行查询。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

名称	类型	描述
coorname	text	当前集群下的CN名称。
datid	oid	用户会话在后台连接到的数据库OID。
datname	text	用户会话在后台连接到的数据库名称。
pid	bigint	后台线程ID。
sessionid	bigint	会话ID。
usesysid	oid	登录该后台的用户OID。
usename	text	登录该后台的用户名。
application_name	text	连接到该后台的应用名。
client_addr	inet	连接到该后台的客户端的IP地址。如果此字段是null，它表明通过服务器机器上UNIX套接字连接客户端或者这是内部线程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
backend_start	timestamp with time zone	该过程开始的时间，即当客户端连接服务器的时间。
xact_start	timestamp with time zone	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	开始当前活跃查询的时间，如果state的值不是active，则这个值是上一个查询的开始时间。
state_change	timestamp with time zone	上次状态改变的时间。
waiting	boolean	如果后台当前正等待锁则为true。
enqueue	text	语句当前排队状态。可能值是： ● waiting in queue：表示语句在排队中。 ● 空：表示语句正在运行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
state	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。 **说明** 只有系统管理员能查看到自己账户所对应的会话状态。其他账户的state信息为空。例如以judy用户连接数据库后，在PGXC_STAT_ACTIVITY视图中查看到的普通用户joe及初始用户omm的state信息为空： SELECT datname, usename, usesysid, state,pid FROM pgxc_stat_activity; datname | usename | usesysid | state | pid ----------+---------+----------+--------+----------------- testdb | omm | 10 | | 139968752121616 testdb | omm | 10 | | 139968903116560 db_tpcds | judy | 16398 | active | 139968391403280 testdb | omm | 10 | | 139968643069712 testdb | omm | 10 | | 139968680818448 testdb | joe | 16390 | | 139968563377936 (6 rows)
resource_pool	name	用户使用的资源池。
query_id	bigint	查询语句的ID。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。
global_sessionid	text	全局会话ID。
unique_sql_id	bigint	语句的unique sql id。
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
top_xid	xid	事务的顶层事务ID。
current_xid	xid	事务的当前事务ID。
xlog_quantity	bigint	事务当前使用的Xlog量，单位为字节。

- pgxc_stat_activity_with_conninfo()

  描述：显示当前集群下所有CN的当前用户查询相关的信息，返回值定义参考PGXC_STAT_ACTIVITY视图。该函数仅具有SYSADMIN或者MONADMIN权限的用户可以执行，普通用户只能查看本用户相关的信息。

  返回值类型：record

- pgxc_stat_all_tables()

  描述：显示各节点数据中每个表（包括TOAST表）的一行的统计信息，该函数仅具有SYSADMIN或者MONADMIN权限的用户可以执行。

  返回值类型：record

- pgxc_get_thread_wait_status()

  描述：查看集群各个节点上所有SQL语句产生的线程之间的调用层次关系，以及各个线程的阻塞等待状态。

  返回值类型：record

- pv_session_memory

  描述：统计Session级别的内存使用情况，包含执行作业在数据节点上GaussDB线程和Stream线程分配的所有内存。

  说明

  若GUC参数enable_memory_limit=off，该函数不能使用。

  返回值类型：record

  表 1-231 返回值说明

名称	类型	描述
sessid	text	线程启动时间+线程标识。
init_mem	integer	当前正在执行作业进入执行器前已分配的内存，单位MB。
used_mem	integer	当前正在执行作业已分配的内存，单位MB。
peak_mem	integer	当前正在执行作业已分配的内存峰值，单位MB。

- DBE_PERF.gs_stat_activity_timeout(int)

  描述：获取当前节点上执行时间超过超时阈值的查询作业信息。需要GUC参数track_activities设置为on才能正确返回结果。超时阈值的取值范围是0~2147483。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：setof record

版权所有 © 华为云计算技术有限公司

名称	类型	描述
database	name	用户会话连接的数据库名称。
pid	bigint	后台线程ID。
sessionid	bigint	会话ID。
usesysid	oid	登录该后台的用户OID。
application_name	text	连接到该后台的应用名。
query	text	该后台正在执行的查询。
xact_start	timestamptz	启动当前事务的时间。
query_start	timestamptz	开始当前查询的时间。
query_id	bigint	查询语句ID。

- DBE_PERF.global_stat_activity_timeout(int)

  描述：获取当前系统（所有CN）中执行时间超过超时阈值的查询作业信息。需要GUC参数track_activities设置为on才能正确返回结果。超时阈值的取值范围是0~2147483。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：setof record

名称	类型	描述
nodename	text	用户会话连接的coordinate node的名称。
database	name	用户会话连接的数据库名称。
pid	bigint	后台线程ID。
sessionid	bigint	会话ID。
usesysid	oid	登录该后台的用户OID。
application_name	text	连接到该后台的应用名。
query	text	该后台正在执行的查询。
xact_start	timestamptz	启动当前事务的时间。
query_start	timestamptz	开始当前查询的时间。
query_id	bigint	查询语句ID。

- DBE_PERF.get_global_active_session()

版权所有 © 华为云计算技术有限公司

描述：显示所有CN节点和主DN节点上的ACTIVE SESSION PROFILE内存中的样本的汇总。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。505.0版本新增event_start_time、current_xid、top_xid三个字段信息，由低于505.0版本升级至505.0或更高版本时，升级观察期无法查询。

返回值类型：record

- DBE_PERF.get_global_os_runtime()

  描述：显示当前操作系统运行的状态信息，仅在CN上支持查询。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_os_threads()

  描述：提供整个集群中所有正常节点下的线程状态信息，仅在CN上支持查询。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_os_threads()

  描述：提供整个集群中所有正常节点下的线程状态信息，仅在CN上支持查询。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_workload_sql_count()

  描述：提供整个集群中不同负载SELECT，UPDATE，INSERT，DELETE，DDL，DML，DCL计数信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_workload_sql_elapse_time()

  描述：提供整个集群中不同负载SELECT、UPDATE、INSERT和DELETE，响应时间信息（TOTAL、AVG、MIN、MAX）。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_workload_transaction()

  描述：获取集群内所有节点上的事务量信息，事务时间信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_session_stat()

  描述：获取集群内所有节点上的会话状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

  说明

  状态信息有如下14项，
  commit,rollback,sql,table_scan,blocks_fetched,physical_read_operation,shared_blocks_dirtied,local_blocks_dirtied,shared_blocks_read,local_blocks_read,blocks_read_time,blocks_write_time,sort_imemory,sort_idisk。

- DBE_PERF.get_global_session_time()

  描述：提供整个集群各节点各个关键阶段的时间消耗。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- DBE_PERF.get_global_session_memory()

  描述：汇聚各节点的Session级别的内存使用情况，包含执行作业在数据节点上gaussdb线程和Stream线程分配的所有内存，单位为MB。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_session_memory_detail()

  描述：汇聚各节点的线程的内存使用情况，以MemoryContext节点来统计。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_session_stat_activity()

  描述：汇聚集群内各节点上正在运行的线程相关的信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_thread_wait_status()

  描述：汇聚所有节点上工作线程（backend thread）以及辅助线程（auxiliary thread）的阻塞等待情况。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_operator_history_table()

  描述：汇聚当前用户所有CN上执行作业结束后的算子相关记录（持久化）。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record。

- DBE_PERF.get_global_operator_history()

  描述：汇聚当前用户所有CN上执行作业结束后的算子相关记录。集群创建后的默认情况下，查询该函数必须具有MONADMIN和SYSADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_operator_runtime()

  描述：汇聚当前用户所有CN上执行作业实时的算子相关记录。集群创建后的默认情况下，查询该函数必须具有MONADMIN和SYSADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statement_complex_history()

  描述：汇聚当前用户所有CN节点上复杂查询的历史记录。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record。

- DBE_PERF.get_global_statement_complex_history_table()

  描述：汇聚当前用户所有CN节点上复杂查询的历史记录（持久化）。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statement_complex_runtime()

  描述：汇聚当前用户所有CN节点上复杂查询的实时信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN和SYSADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_memory_node_detail()

版权所有 © 华为云计算技术有限公司

描述：汇聚某个数据库在所有节点上的内存使用情况。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

返回值类型：record

- DBE_PERF.get_global_shared_memory_detail()

  描述：汇聚所有节点已产生的共享内存上下文的使用信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_comm_delay()

  描述：汇聚所有DN节点的通信库时延状态。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_comm_recv_stream()

  描述：汇聚所有DN节点的通信库接收流状态。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_comm_send_stream()

  描述：汇聚所有DN节点的通信库发送流状态。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_comm_status()

  描述：汇聚所有DN节点的通信库状态。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_all_indexes()

  描述：汇集所有节点当前数据库中的索引信息及I/O统计量。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_local_toastname_and_toastindexname()

  描述：提供本地toast表的name和index和其关联表的对应关系。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_statio_all_indexes()

  描述：统计所有节点当前数据库中的每个索引行，显示特定索引的I/O统计。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_all_sequences()

  描述：提供命名空间中所有sequences的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_all_tables()

  描述：汇集各节点的数据库中每个表I/O的统计。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- DBE_PERF.get_summary_statio_all_tables()

  描述：统计集群内数据库中每个表I/O的统计。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_local_toast_relation()

  描述：提供本地toast表的name和其关联表的对应关系。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_sys_indexes()

  描述：汇聚各节点的命名空间中所有系统表索引的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_statio_sys_indexes()

  描述：统计各节点的命名空间中所有系统表索引的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_sys_sequences()

  描述：提供命名空间中所有系统表为sequences的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_sys_tables()

  描述：提供各节点的命名空间中所有系统表的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_statio_sys_tables()

  描述：集群内汇集命名空间中所有系统表的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_user_indexes()

  描述：各节点的命名空间中所有用户关系表索引的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_statio_user_indexes()

  描述：集群内汇集命名空间中所有用户关系表索引的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_user_sequences()

  描述：显示各节点的命名空间中所有用户的sequences的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_statio_user_tables()

  描述：显示各节点的命名空间中所有用户关系表的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

版权所有 © 华为云计算技术有限公司

返回值类型：record

- DBE_PERF.get_summary_statio_user_tables()

  描述：集群内汇聚命名空间中所有用户关系表的I/O状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_dn_stat_all_tables()

  描述：汇聚DN节点数据库中每个表的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_cn_stat_all_tables()

  描述：汇聚CN节点数据库中每个表的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_dn_stat_all_tables()

  描述：统计DN节点数据库中每个表的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_cn_stat_all_tables()

  描述：统计CN节点数据库中每个表的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_all_indexes()

  描述：汇聚所有节点数据库中每个索引的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_all_indexes()

  描述：统计所有节点数据库中每个索引的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_sys_tables()

  描述：汇聚各节点pg_catalog、information_schema模式的所有命名空间中系统表的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_sys_tables()

  描述：统计各节点pg_catalog、information_schema模式的所有命名空间中系统表的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_sys_indexes()

  描述：汇聚各节点pg_catalog、information_schema模式中所有系统表的索引状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- DBE_PERF.get_summary_stat_sys_indexes()

  描述：统计各节点pg_catalog、information_schema模式中所有系统表的索引状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_user_tables()

  描述：汇聚所有命名空间中用户自定义普通表的状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_user_tables()

  描述：统计所有命名空间中用户自定义普通表的状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_user_indexes()

  描述：汇聚所有数据库中用户自定义普通表的索引状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_user_indexes()

  描述：统计所有数据库中用户自定义普通表的索引状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_database()

  描述：汇聚所有节点数据库统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_database_conflicts()

  描述：统计所有节点数据库统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_xact_all_tables()

  描述：汇聚命名空间中所有普通表和toast表的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_xact_all_tables()

  描述：统计命名空间中所有普通表和toast表的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_xact_sys_tables()

  描述：汇聚所有节点命名空间中系统表的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_xact_sys_tables()

  描述：统计所有节点命名空间中系统表的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

版权所有 © 华为云计算技术有限公司

返回值类型：record

- DBE_PERF.get_global_stat_xact_user_tables()

  描述：汇聚所有节点命名空间中用户表的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_stat_xact_user_tables()

  描述：统计所有节点命名空间中用户表的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_user_functions()

  描述：汇聚所有节点命名空间中用户定义函数的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_xact_user_functions()

  描述：统计所有节点命名空间中用户定义函数的事务状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_stat_bad_block()

  描述：汇聚所有节点表、索引等文件的读取失败信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_file_redo_iostat()

  描述：统计所有节点表、索引等文件的读取失败信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_file_iostat()

  描述：汇聚所有节点数据文件IO的统计。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_locks()

  描述：汇聚所有节点的锁信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_replication_slots()

  描述：汇聚所有节点上逻辑复制信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.GET_GLOBAL_PARALLEL_DECODE_STATUS()

  描述：汇聚集群所有主DN节点上的复制槽的并行解码信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。该函数仅限CN上执行，DN上执行报错。返回值同视图**DBE_PERF.GLOBAL_PARALLEL_DECODE_STATUS**。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- DBE_PERF.GET_GLOBAL_PARALLEL_DECODE_THREAD_INFO()

  描述：汇聚集群所有主DN节点上的复制槽的并行解码线程信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。该函数仅限CN上执行，DN上执行报错。返回值同视图**DBE_PERF.GLOBAL_PARALLEL_DECODE_THREAD_INFO**。

  返回值类型：record

- DBE_PERF.get_global_bgwriter_stat()

  描述：汇聚所有节点后端写线程活动的统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_replication_stat()

  描述：汇聚各节点日志同步状态信息，如发起端发送日志位置，接收端接收日志位置等。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_pooler_status()

  描述：汇聚所有CN节点的pooler中的缓存连接状态。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_transactions_running_xacts()

  描述：汇聚各节点运行事务的信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_transactions_running_xacts()

  描述：统计各节点运行事务的信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_transactions_prepared_xacts()

  描述：汇聚各节点当前准备好进行两阶段提交的事务的信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_transactions_prepared_xacts()

  描述：统计各节点当前准备好进行两阶段提交的事务的信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_statement()

  描述：汇聚各节点历史执行语句状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。函数返回信息具体字段参考**DBE_PERF.SUMMARY_STATEMENT**字段。

  返回值类型：record

- DBE_PERF.get_global_statement_count()

  描述：汇聚各节点SELECT、UPDATE、INSERT、DELETE，响应时间信息（TOTAL、AVG、MIN、MAX）。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

- DBE_PERF.get_global_config_settings()

  描述：汇聚各节点GUC参数配置信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_wait_events()

  描述：汇聚各节点wait events状态信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_statement_responsetime_percentile()

  描述：获取集群SQL响应时间P80，P95分布信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_summary_user_login()

  描述：统计集群各节点用户登录或者退出登录次数信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.get_global_record_reset_time()

  描述：汇聚集群统计信息重置（重启，主备倒换，数据库删除）时间。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- DBE_PERF.standby_statement_history(only_slow[, time1, time2])

  描述：备机中用来查询FULL SQL记录的函数，主机通过表statement_history进行查询，备机通过此函数进行查询。查询该函数必须具有MONADMIN权限。函数返回信息具体字段参考**STATEMENT_HISTORY**字段。

  参数：参见**表1-232**。

  返回值类型：record

表 1-232 standby_statement_history 参数说明

参数	类型	描述
only_slow	boolean	是否仅查询慢SQL。 • true为是，相当于 select .. where is_slow_sql = true。 • false或NULL为查询全部SQL，相当于不对is_slow_sql进行过滤。
time1	timestamptz	可选输入，表示查询SQL的finish_time的最小时间。
time2	timestamptz	可选输入，表示查询SQL的finish_time的最大时间。若time1>=time2，则函数执行报错。

版权所有 © 华为云计算技术有限公司

说明

- 两个时间参数time1、time2的说明：表示查询的SQL的finish_time所属时间段，分别表示起始与终止时间，输入NULL或者不输入表示没有限制，功能等同于select .. where finish_time between time1 and time2。
- 备机上该功能的数据并非存在表里，不存在start_time列的索引，推荐使用参数对finish_time进行条件查找。
- 由于备机Full/Slow SQL采用异步下盘方式，所以用户SQL信息存储时刻可能有所滞后，建议客户查询此接口适当扩大时间查询范围。

- DBE_PERF.global_io_wait_info()

  描述：查询所有CN和DN节点上的IO管控的实时统计信息。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- pg_stat_get_mem_mbytes_reserved(tid)

  描述：统计资源管理相关变量值，仅用于定位问题使用。

  参数：线程id

  返回值类型：text

- pg_stat_get_file_stat()

  描述：通过对数据文件I/O的统计，反映数据的I/O性能，用以发现I/O操作异常等性能问题。

  返回值类型：record

- pg_stat_get_redo_stat()

  描述：用于统计会话线程日志回放情况。

  返回值类型：record

- pg_stat_get_status(int8)

  描述：可以检测当前实例中工作线程（backend thread）以及辅助线程（auxiliary thread）的阻塞等待情况。

  返回值类型：record

- get_local_rel_iostat()

  描述：查询当前节点的数据文件I/O状态累计值。

  返回值类型：record

- DBE_PERF.get_global_rel_iostat()

  描述：汇聚所有节点数据文件I/O的统计。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- pg_catalog.plancache_status(is_with_plan)

  描述：是否显示在当前节点上的全局计划缓存的状态信息，即plan_info列。入参is_with_plan为bool类型，默认值为false。当入参为true时，显示plan_info的信息。函数返回信息和**DBE_PERF.GLOBAL_PLANCACHE_STATUS**一致。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

说明

- 当pg_catalog.plancache_status的入参为true时，如果计划较多，可能产生一定的性能劣化，推荐使用pg_catalog.global_plancache_show_plan函数查询单条SQL的详细计划信息。
- 当对所查询计划依赖的表权限不足时，会在plan_info列显示”权限不足”的报错信息。

- DBE_PERF.global_plancache_status(is_with_plan)

  描述：是否显示在所有节点上的全局计划缓存的状态信息，即plan_info列。入参is_with_plan为bool类型，默认值为false。当入参为true时，显示plan_info的信息。函数返回信息见字段**DBE_PERF.GLOBAL_PLANCACHE_STATUS**。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

  说明

  - 当DBE_PERF.global_plancache_status的入参为true时，如果计划较多，可能产生一定的性能劣化，推荐使用pg_catalog.global_plancache_show_plan函数查询单条SQL的详细计划信息。
  - 当对所查询计划依赖的表权限不足时，会在plan_info列显示”权限不足”的报错信息。

- pg_catalog.gs_plancache_stat()

  描述：查看当前进程内全部session的参数化语句相关信息，函数返回信息和**DBE_PERF.GS_PLANCACHE_STAT**一致。查询该函数需要MONADMIN权限。

  返回值类型：record

- pg_catalog.gs_dist_plancache_stat()

  描述：查看包含备机的全部进程的参数化语句信息的系统函数，函数返回信息和**DBE_PERF.GS_DIST_PLANCACHE_STAT**一致。查询该函数需要MONADMIN权限。

  返回值类型：record

- DBE_PERF.global_threadpool_status()

  描述：显示在所有节点上的线程池中工作线程及会话的状态信息。函数返回信息见字段**DBE_PERF.GLOBAL_THREADPOOL_STATUS**。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  返回值类型：record

- comm_check_connection_status()

  描述：返回该CN和所有活跃节点（CN和主DN）的连接情况。该函数仅支持在CN上查询，普通用户可使用。

  参数：nan

  返回值类型：node_name text, remote_name text, remote_host text, remote_port integer, is_connected boolean, no_error_occur boolean。

- DBE_PERF.global_comm_check_connection_status()

  描述：返回所有CN和所有活跃节点（CN和主DN）的连接情况。该函数仅支持在CN上查询，权限控制继承DBE_PERF schema。集群创建后的默认情况下，查询该函数必须具有MONADMIN权限。

  参数：nan

  返回值类型：node_name text, remote_name text, remote_host text, remote_port integer, is_connected boolean, no_error_occur boolean

版权所有 © 华为云计算技术有限公司

- remote_candidate_stat()

  描述：显示本实例的候选buffer链中页面个数，buffer淘汰信息，包含normal buffer pool和segment buffer pool。

  返回值类型：record

  表 1-233 remote_candidate_stat 参数说明

名称	类型	描述
node_name	text	节点名称。
candidate_slots	integer	当前Normal Buffer Pool候选buffer链中页面个数。
get_buf_from_list	bigint	Normal Buffer Pool，buffer淘汰从候选buffer链中获取页面的次数。
get_buf_clock_sweep	bigint	Normal Buffer Pool，buffer淘汰从原淘汰方案中获取页面的次数。
seg_candidate_slots	integer	当前Segment Buffer Pool候选buffer链中页面个数。
seg_get_buf_from_list	bigint	Segment Buffer Pool，buffer淘汰从候选buffer链中获取页面的次数。
seg_get_buf_clock_sweep	bigint	Segment Buffer Pool，buffer淘汰从原淘汰方案中获取页面的次数。

- remote_ckpt_stat()

  描述：用于显示整个集群所有实例的检查点信息和各类日志刷页情况（本节点除外、DN上不可使用）。

  返回值类型：record

  表 1-234 remote_ckpt_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
ckpt_redo_point	text	当前实例的检查点。
ckpt_clog_flush_num	int8	从启动到当前时间clog刷盘页面数。
ckpt_csnlog_flush_num	int8	从启动到当前时间csnlog刷盘页面数。
ckpt_multixact_flush_num	int8	从启动到当前时间multixact刷盘页面数。
ckpt_predicate_flush_num	int8	从启动到当前时间predicate刷盘页面数。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
ckpt_twophase_flush_num	int8	从启动到当前时间twophase刷盘页面数。

- remote_double_write_stat()

  描述：显示整个集群所有实例的双写文件的情况(本节点除外、DN上不可使用)。

  返回值类型：record

  表 1-235 remote_double_write_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
curr_dwn	int8	当前双写文件的序列号。
curr_start_page	int8	当前双写文件恢复起始页面。
file_trunc_num	int8	当前双写文件复用的次数。
file_reset_num	int8	当前双写文件写满后发生重置的次数。
total_writes	int8	当前双写文件总的I/O次数。
low_threshold_writes	int8	低效率写双写文件的I/O次数（一次I/O刷页数量少于16页面）。
high_threshold_writes	int8	高效率写双写文件的I/O次数（一次I/O刷页数量多于一批，421个页面）。
total_pages	int8	当前刷页到双写文件区的总的页面个数。
low_threshold_pages	int8	低效率刷页的页面个数。
high_threshold_pages	int8	高效率刷页的页面个数。
file_id	int8	当前双写文件的id号。

- remote_single_flush_dw_stat()

  描述：显示整个集群所有实例的单页面淘汰双写文件的情况（本节点除外、DN上不可使用）。

  返回值类型：record

  表 1-236 remote_single_flush_dw_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
curr_dwn	integer	当前双写文件的序列号。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
curr_start_page	integer	当前双写文件start位置。
total_writes	bigint	当前双写文件总计写数据页面个数。
file_trunc_num	bigint	当前双写文件复用的次数。
file_reset_num	bigint	当前双写文件写满后发生重置的次数。

- remote_pagewriter_stat()
  描述：显示整个集群所有实例的刷页信息和检查点信息（本节点除外、DN上不可使用）。
  返回值类型：record

  表 1-237 remote_pagewriter_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
pgwr_actual_flush_total_num	int8	从启动到当前时间总计刷脏页数量。
pgwr_last_flush_num	int4	上一批刷脏页数量。
remain_dirty_page_num	int8	当前预计还剩余多少脏页。
queue_head_page_rec_lsn	text	当前实例的脏页队列第一个脏页的recovery_lsn。
queue_rec_lsn	text	当前实例的脏页队列的recovery_lsn。
current_xlog_insert_lsn	text	当前实例Xlog写入的位置。
ckpt_redo_point	text	当前实例的检查点。

- remote_recovery_status()
  描述：显示关于主机和备机的日志流控信息（本节点除外、DN上不可使用）。
  返回值类型：record

  表 1-238 remote_recovery_status 参数说明

参数	类型	描述
node_name	text	节点的名称，包含主机和备机。
standby_node_name	text	备机名称。
source_ip	text	主机的IP地址。
source_port	int4	主机的端口号。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
dest_ip	text	备机的IP地址。
dest_port	int4	备机的端口号。
current_rto	int8	备机当前的日志流控时间，单位秒。
target_rto	int8	备机通过GUC参数设置的预期流控时间，单位秒。
current_sleep_time	int8	为了达到这个预期主机所需要的睡眠时间，单位微秒。

- remote_rto_stat()
  描述：显示关于主机和备机的日志流控信息（本节点除外、DN上不可使用）。
  返回值类型：record

  表 1-239 remote_rto_stat 参数说明

参数	类型	描述
node_name	text	节点的名称，包含主机和备机。
rto_info	text	流控的信息，包含了备机当前的日志流控时间（单位：秒），备机通过GUC参数设置的预期流控时间（单位：秒），为了达到这个预期主机所需要的睡眠时间（单位：微秒）。

- remote_redo_stat()
  描述：显示整个集群所有实例的日志回放情况（本节点除外、DN上不可使用）。
  返回值类型：record

  表 1-240 remote_redo_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
redo_start_ptr	int8	当前实例日志回放的起始点。
redo_start_time	int8	当前实例日志回放的起始UTC时间。
redo_done_time	int8	当前实例日志回放的结束UTC时间。
curr_time	int8	当前实例的当前UTC时间。
min_recovery_point	int8	当前实例日志的最小一致性点位置。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
read_ptr	int8	当前实例日志的读取位置。
last_replayed_read_ptr	int8	当前实例的日志回放位置。
recovery_done_ptr	int8	当前实例启动完成时的回放位置。
read_xlog_io_counter	int8	当前实例读取回放日志的I/O次数计数。
read_xlog_io_total_dur	int8	当前实例读取回放日志的I/O总时延。
read_data_io_counter	int8	当前实例回放过程中读取数据页面的I/O次数计数。
read_data_io_total_dur	int8	当前实例回放过程中读取数据页面的I/O总时延。
write_data_io_counter	int8	当前实例回放过程中写数据页面的I/O次数计数。
write_data_io_total_dur	int8	当前实例回放过程中写数据页面的I/O总时延。
process_pending_counter	int8	当前实例回放过程中日志分发线程的同步次数计数。
process_pending_total_dur	int8	当前实例回放过程中日志分发线程的同步总时延。
apply_counter	int8	当前实例回放过程中回放线程的同步次数计数。
apply_total_dur	int8	当前实例回放过程中回放线程的同步总时延。
speed	int8	当前实例日志回放速率。
local_max_ptr	int8	当前实例启动成功后本地收到的回放日志的最大值。
primary_flush_ptr	int8	主机落盘日志的位置。
worker_info	text	当前实例回放线程信息，若没有开并行回放则该值为空。

- pgxc_gtm_snapshot_status()

  描述：用于查看当前GTM上事务信息，仅在GTM模式下支持本系统函数，GTM-LITE和GTM-FREE模式下不支持。

  返回值类型：record

  函数返回字段描述如下：

版权所有 © 华为云计算技术有限公司

表 1-241 pgxc_gtm_snapshot_status 返回参数说明

名称	类型	描述
xmin	xid	仍在运行的最小事务号。
xmax	xid	已完成的所有事务号中最大事务号的下一个事务号。
csn	integer	待提交事务的序列号。
oldestxmin	xid	当前最早的活跃事务在其取快照时，所有运行事务号最小的事务。
xcnt	integer	当前活跃的事务个数。
running_xids	text	当前活跃的事务号。

- pv_os_run_info()

  描述：显示当前操作系统运行的状态信息，具体字段信息参考**PV_OS_RUN_INFO**。

  参数：nan

  返回值类型：setof record

- pv_session_stat()

  描述：以会话线程或AutoVacuum线程为单位，统计会话状态信息，具体字段信息参考**PV_SESSION_STAT**。

  参数：nan

  返回值类型：setof record

- pv_session_time()

  描述：用于统计会话线程的运行时间信息，及各执行阶段所消耗时间，具体字段信息参考**PV_SESSION_TIME**。

  参数：nan

  返回值类型：setof record

- pg_stat_get_db_temp_bytes()

  描述：用于统计通过数据库查询写入临时文件的数据总量。计算所有临时文件，不论临时文件为何创建，也不管log_temp_files如何设置。

  参数：oid

  返回值类型：bigint

- pg_stat_get_db_temp_files()

  描述：通过数据库查询创建的临时文件数量。计算所有临时文件，不论临时文件为何创建（比如排序或者哈希），也不管log_temp_files如何设置。

  参数：oid

  返回值类型：bigint

- gs_prepared_statements()

  描述：显示所有会话所有可用的预备语句。该函数需要SYSADMIN权限的用户才能够执行。函数返回信息具体的字段和**GS_ALL_PREPARED_STATEMENTS**字段一致。

版权所有 © 华为云计算技术有限公司

返回值类型：record

- local_redo_time_count([int])

  描述：返回本节点回放流程的耗时统计（仅在备机上存在有效数据）。在并行回放模式下，返回各个回放线程的流程耗时统计。在极致RTO模式下，返回各回放线程的流程耗时统计和不同日志类型的流程耗时统计。

  参数：

参数名	类型	描述
operation	int	可选输入，若无输入则行为是查询统计信息。 ● -1：表示关闭统计开关（默认状态为关闭）。 ● 0：表示打开统计开关。 ● 1：表示查询统计信息。 ● 2：表示重置统计信息。 关闭统计开关后，重新打开时会清理之前的统计信息。

返回值如下：

表 1-242 local_redo_time_count 返回参数说明

字段名	描述
thread_name	线程名称。
step1_total	step1的总时间，每个线程对应的流程如下： ● 极致RTO： redo batch：从队列中获取一条日志。 redo manager：从队列中获取一条日志。 redo worker：从队列中获取一条日志。 trxn manager：从队列中读取一条日志。 trxn worker：从队列中读取一条日志。 read worker：从文件中读取一次Xlog page（整体）。 read page worker：从队列中获取一个日志。 startup：从队列中获取一个日志。 ● 并行回放： page redo：从队列中获取一条日志。 startup：读取一条日志。
step1_count	step1的统计次数。

版权所有 © 华为云计算技术有限公司

字段名	描述
step2_total	step2的总时间，每个线程对应的流程如下： ● 极致RTO： redo batch：处理日志（整体）。 redo manager：处理日志（整体）。 redo worker：处理日志（整体）。 trxn manager：处理日志（整体）。 trxn worker：处理日志（整体）。 read worker：读取Xlog page耗时。 read page worker：生成和发送lsn forwarder。 startup：check stop（是否回放到指定位置）。 ● 并行回放： startup：check stop（是否回放到指定位置）。
step2_count	step2的统计次数。
step3_total	step3的总时间，每个线程对应的流程如下： ● 极致RTO： redo batch：更新standbystate。 redo manager：数据日志处理。 redo worker：回放page页日志（整体）。 trxn manager：更新flush lsn。 trxn worker：回放日志处理。 read worker：推进Xlog segment。 read page worker：获取一个新的item。 startup：redo delay（延迟回放特性等待时间）。 ● 并行回放： page redo：更新standbystate。 startup：redo delay（延迟回放特性等待时间）。
step3_count	step3的统计次数。

版权所有 © 华为云计算技术有限公司

字段名	描述
step4_total	step4的总时间，每个线程对应的流程如下： ● 极致RTO： redo batch：解析Xlog。 redo manager：DDL处理。 redo worker：读取数据page页。 trxn manager：同步等待时间。 trxn worker：更新本线程lsn。 read page worker：将日志放入分发线程。 startup：分发（整体）。 ● 并行回放： page redo：undo 日志回放。 startup：分发（整体）。
step4_count	step4的统计次数。
step5_total	step5的总时间，每个线程对应的流程如下： ● 极致RTO： redo manager：database类日志处理。 redo worker：回放数据page页的日志。 trxn manager：分发给trxn worker。 trxn worker：强同步类型日志通知完成等待。 read page worker：更新本线程lsn。 startup：日志decode。 ● 并行回放： page redo：share trxn日志回放。 startup：日志回放。
step5_count	step5的统计次数。
step6_total	step6的总时间，每个线程对应的流程如下： ● 极致RTO： redo worker：回放非数据页page日志。 trxn manager：全局lsn更新。 ● 并行回放： page redo：synctrxn日志回放。 startup：强同步等待。
step6_count	step6的统计次数。

版权所有 © 华为云计算技术有限公司

字段名	描述
step7_total	step7的总时间，每个线程对应的流程如下： ● 极致RTO： redo manager：创建表空间。 redo worker：fsm更新。 ● 并行回放： page redo：single日志回放。
step7_count	step7的统计次数。
step8_total	step8的总时间，每个线程对应的流程如下： ● 极致RTO： redo worker：强同步等待。 redo manager：段页式类日志处理。 ● 并行回放: page redo：all workers do日志回放。
step8_count	step8的统计次数。
step9_total	step9的总时间，每个线程对应的流程如下： ● 极致RTO： redo manager：处理段页式UNDO日志。 ● 并行回放： page redo：multi workers do日志回放。
step9_count	step9的统计次数。

日志类型的流程耗时统计：（开关为关闭的情况下，会对回放的日志进行抽样统计）

字段名	流程描述
thread_name	采样的名称
step1_total	极致RTO: ● 数据类型日志：处理日志整体用时，包含step2。 ● 事务类型日志：回放事务日志。
step1_count	step1的统计次数。
step2_total	极致RTO: ● 数据类型日志：回放page页日志整体用时，包含step3、step4。step2和（step3+step4）的差值主要是执行备机读多版本页面的构建。 ● 事务类型日志：事务线程回放前同步等待时间。
step2_count	step2的统计次数。

版权所有 © 华为云计算技术有限公司

字段名	流程描述
step3_total	极致RTO: ● 数据类型日志：读取数据page页。
step3_count	step3的统计次数。
step4_total	极致RTO: ● 数据类型日志：回放数据page页的日志。
step4_count	step4的统计次数。
step5_total	极致RTO: ● 事务类型日志：在redo_worker线程回放的用时。
step5_count	step5的统计次数。
step6_total	极致RTO: ● 数据类型日志：更新fsm。
step6_count	step6的统计次数。
Step7_total	极致RTO: ● 事务类型日志：在redo_worker线程同步等待的用时。
Step7_count	Step7的统计次数。

- local_xlog_redo_statics()

  描述：返回本节点已经回放的各个类型的日志统计信息（仅在备机上有有效数据）。

  返回值如下：

  表 1-243 local_xlog_redo_statics 返回参数说明

字段名	描述
xlog_type	日志类型。
rmid	resource manager id。
info	Xlog operation。
num	日志个数。
extra	针对page回放日志和xact日志有有效值。 ● 如果是page页回放日志类型则表示从磁盘读取page的个数。 ● 如果是xact日志类型则表示删除文件的个数。

- remote_bgwriter_stat()

版权所有 © 华为云计算技术有限公司

描述：显示整个集群所有实例的bgwriter线程刷页信息，候选buffer链中页面个数，buffer淘汰信息（本节点除外、DN上不可使用）。

返回值类型：record

表 1-244 remote_bgwriter_stat 参数说明

参数	类型	描述
node_name	text	实例名称。
bgwr_actual_flush_total_num	bigint	从启动到当前时间bgwriter线程总计刷脏页数量。
bgwr_last_flush_num	integer	bgwriter线程上一批刷脏页数量。
candidate_slots	integer	当前候选buffer链中页面个数。
get_buffer_from_list	bigint	buffer淘汰从候选buffer链中获取页面的次数。
get_buf_clock_sweep	bigint	buffer淘汰从原淘汰方案中获取页面的次数。

示例：

remote_bgwriter_stat函数查询bgwriter线程刷页信息。

```
gaussdb=# SELECT * FROM remote_bgwriter_stat();
 node_name | bgwr_actual_flush_total_num | bgwr_last_flush_num | candidate_slots |
get_buffer_from_list | get_buf_clock_sweep
-----------+-----------------------------+---------------------+-----------------+----------------------
+----------------------
 datanode3 |                           0 |                   0 |          266232 |                 404 |                    0
 datanode2 |                           0 |                   0 |          266232 |                 424 |                    0
 datanode1 |                           0 |                   0 |          266232 |                 393 |                    0
(3 rows)
```

- gs_stack()

  描述：显示线程调用栈。查询该函数需要有SYSADMIN权限或者MONADMIN权限。

  参数：tid，线程id。tid是可选参数，指定tid参数时，函数返回tid对应线程调用栈；当不指定tid参数时，函数返回所有线程的调用栈。

  返回值：当指定tid时，返回值为text；当不指定tid时，返回值为setof record。

  示例：

  获取指定线程调用栈。

```
gaussdb=# SELECT * FROM gs_stack(139663481165568);
                          gs_stack
-------------------------------------------------------------------------
 __poll + 0x2d                                                          +
 WaitLatchOrSocket(Latch volatile*, int, int, long) + 0x29f       +
 WaitLatch(Latch volatile*, int, long) + 0x2e                    +
 JobScheduleMain() + 0x90f                                      +
 int GaussDbThreadMain<(knl_thread_role)9>(knl_thread_arg*) + 0x456+
 InternalThreadFunc(void*) + 0x2d                                +
 ThreadStarterFunc(void*) + 0xa4                                 +
 start_thread + 0xc5                                           +
```


版权所有 © 华为云计算技术有限公司

```
 clone + 0x6d                                            +
(1 row)
```

获取所有线程的调用栈。

```
gaussdb=# SELECT * FROM gs_stack();
-[ RECORD
1 ]----------------------------------------------------------------------------------------------------
tid   | 139670364324352
lwtid | 308
stack | __poll + 0x2d
      | CommWaitPollParam::caller(int (*)(pollfd*, unsigned long, int), unsigned long) + 0x34
      | int comm_socket_call<CommWaitPollParam, int (*)(pollfd*, unsigned long,
int)>(CommWaitPollParam*, int (*)(pollfd*, unsigned long
, int)) + 0x28
      | comm_poll(pollfd*, unsigned long, int) + 0xb1
      | ServerLoop() + 0x72b
      | PostmasterMain(int, char**) + 0x314e
      | main + 0x617
      | __libc_start_main + 0xf5
      | 0x55d38f8db3a7
[ RECORD 2 ]----------------------------------------------------------------------------------------------------
tid   | 139664851859200
lwtid | 520
stack | __poll + 0x2d
      | WaitLatchOrSocket(Latch volatile*, int, int, long) + 0x29f
      | SysLoggerMain(int) + 0xc86
      | int GaussDbThreadMain<(knl_thread_role)17>(knl_thread_arg*) + 0x45d
      | InternalThreadFunc(void*) + 0x2d
      | ThreadStarterFunc(void*) + 0xa4
      | start_thread + 0xc5
      | clone + 0x6d
```

说明

不支持多个gs_stack()并发。

- gs_perf_start(duration integer[, freq integer])

  描述：采集当前节点上各个线程的on-cpu火焰图的堆栈调用数据，函数返回火焰图落盘地址。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT/DBUSER模式下可用，在DISABLE模式下不可用。

  参数说明如下：

参数名称	描述	类型	范围
duration	采集堆栈时长（单位为秒，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	1~60
freq	采集堆栈频率（可选参数，单位为HZ，默认为100，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	10~1000

  返回值类型：text

  示例：

  设置频率为100HZ，采集10s的on-cpu堆栈信息。

```
gaussdb=# SELECT gs_perf_start(10,100);
                                gs_perf_start
-------------------------------------------------------------------------------
 Successfully collected perf flamegraph data, the result filepaths are :       +
```


版权所有 © 华为云计算技术有限公司

```
$GAUSSLOG/gs_flamegraph/cn_5001/flamegraph-manual-oncpu-2025-03-03_104812.html.gz;+
Record 886 cpu clock samples.                                                    +

(1 row)
```

须知

执行该函数时（包括后述gs_perf_start_offcpu()、gs_perf_start_all()、gs_perf_start_detail()）需要gs_perf_service正常运行。同时gs_perf_service在采集过程中，需要申请环形buffer，该buffer的大小受操作系统中/proc/sys/kernel/perf_event_mlock_kb控制。如果采集过程中出现“perf mmap failed”错误，可以在调整/proc/sys/kernel/perf_event_mlock_kb的大小后，再重新执行采集。

- gs_perf_start_offcpu(duration integer)

  描述：采集当前节点上各个线程的off-cpu火焰图的堆栈调用数据，函数返回火焰图落盘地址。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

  参数说明如下：

参数名称	描述	类型	范围
duration	采集堆栈时长（单位为毫秒，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	50~3000

  返回值类型：text

  示例：

  采集2000ms的off-cpu堆栈信息。

```
gaussdb=# SELECT gs_perf_start_offcpu(2000);
                                gs_perf_start_offcpu
-------------------------------------------------------------------------------------
Successfully collected perf flamegraph data, the result filepaths are :          +
$GAUSSLOG/gs_flamegraph/cn_5001/flamegraph-manual-offcpu-2025-03-03_104918.html.gz;+
Record 48 sched switch samples.                                                  +

(1 row)
```

- gs_perf_start_detail(duration integer[, freq integer])

  描述：采集当前节点所在机器上所有CPU核心的所有线程中on-cpu/off-cpu堆栈的调用栈详细信息，函数返回多核多线程原始数据压缩文件地址。每次运行该命令时，都会删除历史采集的多核多线程数据。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

  参数说明如下：

版权所有 © 华为云计算技术有限公司

参数名称	描述	类型	范围
duration	采集堆栈时长（单位为毫秒，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	50~3000
freq	采集堆栈频率（可选参数，单位为HZ，默认为100，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	10~1000

返回值类型：text

示例：

采集2000ms的多核多线程火焰图数据。

```
gaussdb=# SELECT gs_perf_start_detail(2000);
                                                                                                  gs_perf_start_detail
-----------------------------------------------------------------------------------------------------------------------
 Successfully collected perf detail data, the result filepath is :                                                    +
 $GAUSSLOG/gs_flamegraph/cn_5001/detail-2025-03-03_105043.lz4                                                           +
 Please save the file locally, otherwise the file will be deleted during the next data collection.+
 Record 1735211 samples.                                                                                              +

(1 row)
```

- gs_perf_start_all(duration integer[, freq integer])

  描述：采集当前节点所在机器上所有节点中on-cpu/off-cpu火焰图的堆栈调用数据，函数返回所有节点落盘火焰图文件地址。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER模式下只能采集oncpu火焰图数据，在DISABLE模式下不可用。

  参数说明如下：

参数名称	描述	类型	范围
duration	采集堆栈时长（单位为毫秒，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	50~3000
freq	采集堆栈频率（可选参数，单位为HZ，默认为100，如果输入浮点型，会对浮点型小数点后第一位进行四舍五入后取整处理）。	integer	10~1000

  返回值类型：text

  示例：

  设置频率为100HZ，采集1000ms当前节点所在机器上所有节点中oncpu/offcpu火焰图数据。

```
gaussdb=# SELECT gs_perf_start_all(1000,100);
                                                                                                  gs_perf_start_all
```


版权所有 © 华为云计算技术有限公司

```
-----------------------------------------------------------------------------------------------------------
 Successfully collected perf flamegraph data, the result filepaths are :                              +
 $GAUSSLOG/gs_flamegraph/{nodename}/flamegraph-manual-
oncpu-2025-03-03_105122.html.gz;                  +
 $GAUSSLOG/gs_flamegraph/{nodename}/flamegraph-manual-
offcpu-2025-03-03_105122.html.gz;                 +
 Please visit gs_flamegraph directory(all nodes of current machine) to get results for individual nodes.+
 Record 664 cpu clock samples.                                                                        +
 Record 370 sched switch samples.                                                                      +

(1 row)
```

- gs_perf_list()

  描述：查看当前节点上保存的on-cpu/off-cpu火焰图的采集数据，不会展示多核多线程火焰图记录。该函数在ROOT/DBUSER模式下可用，在DISABLE模式下不可用。

  参数：nan

  返回值类型：setof record

  函数返回字段说明：

名称	类型	描述
nodename	text	采集节点名称。
filename	text	采集落盘文件名称。
collect_type	text	采集类型。

  示例：

  查询当前节点上保存的火焰图生成记录。

```
gaussdb=# SELECT * FROM gs_perf_list() limit 10;
 nodename |                          filename                          | collect_type
----------+------------------------------------------------------------+--------------
 cn_5001  | flamegraph-auto-offcpu-2025-03-02_175022.html.gz       | auto
 cn_5001  | flamegraph-cpuspiking-offcpu-2025-03-02_140825.html.gz | cpuspiking
 cn_5001  | flamegraph-auto-offcpu-2025-03-03_063522.html.gz       | auto
 cn_5001  | flamegraph-auto-offcpu-2025-03-03_102523.html.gz       | auto
 cn_5001  | flamegraph-auto-offcpu-2025-03-02_173523.html.gz       | auto
 cn_5001  | flamegraph-manual-offcpu-2025-03-03_025410.html.gz     | manual
 cn_5001  | flamegraph-manual-oncpu-2025-03-03_003428.html.gz      | manual
 cn_5001  | flamegraph-manual-oncpu-2025-03-02_181653.html.gz      | manual
 cn_5001  | flamegraph-manual-oncpu-2025-03-03_031815.html.gz      | manual
 cn_5001  | flamegraph-manual-oncpu-2025-03-03_001013.html.gz      | manual
(10 rows)
```

- gs_perf_query([filename text])

  描述：查询历史采集on-cpu/off-cpu火焰图的堆栈调用数据。函数归并各个线程的函数调用栈并求和函数运行时间，返回聚集后的结果。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT/DBUSER模式下可用，在DISABLE模式下不可用。

  参数说明如下：

参数名称	描述	类型
filename	历史采集的文件的名称（可选参数，默认为最近一次手动采集的火焰图）。	text

版权所有 © 华为云计算技术有限公司

返回值类型：setof record

函数返回字段说明：

名称	类型	描述
backtrace	text	堆栈名称（带树状结构）。
period	bigint	堆栈的执行时间。
level	integer	堆栈所在堆栈调用树的层级。
sequence	integer	堆栈调用树排序后的顺序。
thread_name	text	该堆栈所在的线程名称。
overhead	float	该堆栈执行所占时间百分比。

示例：

查询采集到的堆栈信息。

```
gaussdb=# SELECT * FROM gs_perf_query() WHERE overhead > 2 AND level < 10;
                                  backtrace                                   |   period    | level | sequence |
thread_name  | overhead
------------------------------------------------------------------------------+-------------+-------+----------
+---------------+----------
 root                                                                         +| 74140000000 |     0 |        1 | root          |
100
                                                                             |             |       |          |               |
    worker                                                                     +| 69930000000 |     1 |        2 | worker
|    94.32
                                                                             |             |       |          |               |
    |── start_thread                                                           +| 67620000000 |     2 |        3 | worker
|    91.21
                                                                             |             |       |          |               |
    |    └── ThreadStarterFunc                                                  +| 67620000000 |     3 |        4 |
worker       |    91.21
                                                                             |             |       |          |               |
    |       └── internal_thread_func                                            +| 67620000000 |     4 |        5 |
worker       |    91.21
                                                                             |             |       |          |               |
    |          └── int gauss_db_thread_main                                     +| 67620000000 |     5 |        6 |
worker       |    91.21
                                                                             |             |       |          |               |
     |            └── backend_run                                              +| 67620000000 |     6 |        7 |
worker       |    91.21
                                                                             |             |       |          |               |
    |              |── PostgresMain                                           +| 67520000000 |     7 |        8 | worker
|    91.07
                                                                             |             |       |          |               |
    |              |   |── exec_simple_query                                   +| 64800000000 |     8 |        9 |
worker       |     87.4
                                                                             |             |       |          |               |
    |              |   |   |── OpFusion::opfusion_process                      +| 30130000000 |     9 |       10 |
worker       |    40.64
                                                                             |             |       |          |               |
    |              |   |   |── pg_analyze_and_rewrite                          +| 11290000000 |     9 |     1405 |
worker       |    15.23
                                                                             |             |       |          |               |
    |              |   |   |── pg_plan_queries                                 +|  9550000000 |     9 |     2660 |
worker       |    12.88
                                                                             |             |       |          |               |
    |              |   |   |── PortalRun                                      +|  4680000000 |     9 |     4310 | worker
|     6.31
                                                                             |             |       |          |               |
```


版权所有 © 华为云计算技术有限公司

```
    |                |   |   |── finish_xact_command                    +|  3120000000 |     9 |     4923 |
worker        |     4.21
                                                                         |           |      |         |            |
    |                |   |   |── pg_parse_query                       +|  1520000000 |     9 |     5262 |
worker        |     2.05
                                                                         |           |      |         |            |
    |                |   |   |── OpFusion::opfusion_factory           +|  1500000000 |     9 |     5374 |
worker        |     2.02
                                                                         |           |      |         |            |
    txnsnapworker                                                     +|  1640000000 |     1 |     6770 |
txnsnapworker |     2.21
                                                                         |           |      |         |            |
    |── start_thread                                                 +|  1620000000 |     2 |     6771 |
txnsnapworker |     2.19
                                                                         |           |      |         |            |
    |   └── ThreadStarterFunc                                         +|  1620000000 |     3 |     6772 |
txnsnapworker |     2.19
                                                                         |           |      |         |            |
    |       └── internal_thread_func                                  +|  1620000000 |     4 |     6773 |
txnsnapworker |     2.19
                                                                         |           |      |         |            |
    |           └── int gauss_db_thread_main                          +|  1620000000 |     5 |     6774 |
txnsnapworker |     2.19
                                                                         |           |      |         |            |
    |               └── txn_snap_cap_worker_main                      +|  1620000000 |     6 |     6775 |
txnsnapworker |     2.19
                                                                         |           |      |         |            |
    |                   └── PostgresInitializer::InitTxnSnapWorker    +|  1620000000 |     7 |     6776 |
txnsnapworker |     2.19
                                                                         |           |      |         |            |
    |                       |── PostgresInitializer::SetDatabase      +|  1600000000 |     8 |     6777 |
txnsnapworker |     2.16
                                                                         |           |      |         |            |
    |                       |   └── PostgresInitializer::SetDatabaseByName+|  1600000000 |     9 |     6778 |
txnsnapworker |     2.16
```

- gs_perf_query_general([filename text])

  描述：查询历史采集on-cpu/off-cpu火焰图的堆栈调用函数占比数据，由于计算精度原因，该占比数据总和可能会不等于1。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT/DBUSER模式下可用，在DISABLE模式下不可用。

  参数说明如下：

参数名称	描述	类型
filename	历史采集的文件的名称（可选参数，默认为最近一次手动采集的火焰图）。	text

  返回值类型：setof record

  函数返回字段说明：

名称	类型	描述
function_name	text	堆栈函数名称。
percentage	float	堆栈函数调用占比。

  示例：

  查询最近一次手动采集生成的火焰图函数调用占比。

```
gaussdb=# SELECT * FROM gs_perf_query_general();
         function_name          | percentage
-------------------------------+------------
 pthread_mutex_lock             |      28.57
 do_select                      |       7.17
 undopage::init_undo_zone_lock  |       7.14
 unknown                        |       7.14
 undo::recycle_main             |       7.14
 undo::is_valid_zone_id         |       7.14
 __vdso_gettimeofday            |       7.14
 hrtimer_start_range_ns         |       7.14
 undo::get_zone_id              |       7.14
 __pthread_mutex_unlock         |       7.14
 gs_stat_encap_status_info      |       7.14
(11 rows)
```

- gs_perf_query_detail([filename text])

  描述：查询历史采集on-cpu/off-cpu火焰图的堆栈函数调用占比及调用堆栈详细信息，由于计算精度原因，该占比数据总和可能会不等于1。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT/DBUSER模式下可用，在DISABLE模式下不可用。

  参数说明如下：

参数名称	描述	类型
filename	历史采集的文件的名称（可选参数，默认为最近一次手动采集的火焰图）。	text

  返回值类型：setof record

  函数返回字段说明：

名称	类型	描述
function_name	text	堆栈函数名称。
percentage	float	堆栈函数调用占比。
function_callstack	text	堆栈函数调用栈。
function_callstack_percentage	float	堆栈函数调用栈占比。

  示例：

  查询最近一次手动采集生成的火焰图调用占比详情。

```
testdb3=# SELECT * FROM gs_perf_query_detail();
      function_name       | percentage
|
         function_callstack
                                                                                                    |
function_callstack_percentage
--------------------------+------------
+-------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
--------------------------+------------------------------
 pthread_mutex_lock       |         30 | TrackStmtWorker->unknown->internal_thread_func->int
```

```
gauss_db_other_feature_thread_main->pthread_mutex_lock

|                        30
 stmt_flush_main          |         20 | TrackStmtWorker->unknown->internal_thread_func->int
gauss_db_other_feature_thread_main->stmt_flush_main

|                        20
 finish_task_switch       |         20 | AntiRecycler->unknown->internal_thread_func->int
gauss_db_storage_thread_main->anti_recycler_main->anti_tab_recycle_loop->__select ->[enter kernel]-
>system_call_fastpath->SyS_select->core_
sys_select->do_select->poll_schedule_timeout->schedule_hrtimeout_range-
>schedule_hrtimeout_range_clock->schedule->__schedule->finish_task_switch |                          10
 finish_task_switch       |         20 | vlogwriter->unknown->internal_thread_func->int
gauss_db_storage_thread_main->verifylogwriter_main->__select ->[enter kernel]-
>system_call_fastpath->SyS_select->core_sys_select->do_select->
poll_schedule_timeout->schedule_hrtimeout_range->schedule_hrtimeout_range_clock->schedule-
>__schedule->finish_task_switch                        |                          10
 system_call_after_swapgs |         10 | WALwriter->semctl ->[enter kernel]->system_call_after_swapgs

|                        10
 lwlock_release           |         10 | AntiRecycler->unknown->internal_thread_func->int
gauss_db_storage_thread_main->anti_recycler_main->anti_tab_recycle_loop->lwlock_release

|                        10
 __select                 |         10 | AntiRecycler->unknown->internal_thread_func->int
gauss_db_storage_thread_main->anti_recycler_main->anti_tab_recycle_loop->__select

|                        10
(7 rows)
```

- gs_perf_report()

  描述：当前版本采集逻辑默认落盘，不再需要手动指令进行落盘。作用为展示上一次手动查询oncpu的文件地址。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT/DBUSER模式下可用，在DISABLE模式不可用。

  参数：nan

  返回值类型：text

  示例：

  生成火焰图文件。

```
testdb3=# SELECT * FROM gs_perf_report();
               gs_perf_report
-------------------------------------------------
 flamegraph-manual-oncpu-2025-01-24_140232.html.gz
(1 row)
```

- gs_perf_clean()

  描述：当前版本采集逻辑默认落盘，采集后不再占用内存上下文，不再需要手动指令进行清理内存数据。当前版本提示函数废弃，后续版本直接移除函数。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT/DBUSER模式下可用，在DISABLE模式下不可用。

  参数：nan

  返回值类型：text

  示例：

  清理perf产生的数据。

```
gaussdb=# SELECT * FROM gs_perf_clean();
                                                gs_perf_clean
-------------------------------------------------------------------------------------------------------------------------------
-----------
 gs_perf_clean is a duplicated function. now you don't need clean perf memory. this function will be
removed in the next version
(1 row)
```


版权所有 © 华为云计算技术有限公司

- gs_perf_start_shaking_collect(scope text, unique_querys textarray, duration int, max_collect_times int, ringbuffer_size int)

  描述：用于开启性能抖动监控，并返回开启操作是否成功。开启性能抖动监控后会清理已存在的性能抖动火焰图文件。重复执行gs_perf_start_shaking_collect函数，若前一次性能抖动监控未结束，后一次仅会重设unique_querys参数和清理已存在的性能抖动火焰图文件，不会影响gs_perf_service的行为。当某SQL发生慢SQL且属于unique_querys入参时，则触发性能抖动火焰图生成，如果该慢SQL在CN上执行则采集所有节点的性能抖动火焰图，否则采集本节点的性能抖动火焰图。节点开启性能抖动监控失败后会尝试关闭该节点性能抖动监控。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

  表 1-245 gs_perf_start_shaking_collect 参数说明

参数名称	类型	描述	范围
scope	text	local代表开始本节点监控。all代表开启所有节点性能抖动监控，all仅支持在分布式CN上使用。必填，填NULL则为local。	local或all
unique_querys	textarray	被监控的unique sql id列表，慢sql属于这个列表的话是性能抖动。必填，填NULL则为空列表。	列表长度限制为0-32
duration	int	性能抖动最长连续监控时长，单位为分钟。必填，填NULL则为60。	5-6000
max_collect_times	int	性能抖动火焰图最大性能抖动快照保存次数，超过该值停止采集perf数据。必填，填NULL则为3。	1-10
ringbuffer_size	int	保存性能perf的环形缓冲区的大小，单位为MB。必填，填NULL则为8。	1-32

  返回值说明如下：

  表 1-246 gs_perf_start_shaking_collect 返回值说明

名称	类型	描述
nodename	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
message	text	返回值为“Successfully entered CPU_SHAKE mode.”，表示性能抖动监控开启成功。 返回值为“Current is in CPU_SHAKE mode.”，表示重复开启监控。 返回值为“Unexpected Result: start_shaking_collect error，try to close shaking collect success. Details：详细报错消息”，表示该节点开启失败，并成功关闭。 返回值为“Unexpected Result: start_shaking_collect error，try to close shaking collect failed. Details：详细报错消息”，表示该节点开启失败，并且关闭失败，常见于断连状态。

示例：

开启所有节点性能抖动监控。设置要监控的unique sql id 列表为{4124148646,2053082188}，监控时长为60分钟，最大性能抖动快照保存次数为4次，环形缓冲区大小为8MB。

```
gaussdb=# SELECT * FROM gs_perf_start_shaking_collect('all','{4124148646,2053082188}',60,4,8);
 nodename |                message
----------+---------------------------------------
 cn_5001  | Current is in CPU_SHAKE mode.
 cn_5002  | Current is in CPU_SHAKE mode.
 cn_5003  | Current is in CPU_SHAKE mode.
 dn_6001  | Successfully entered CPU_SHAKE mode.
 dn_6004  | Successfully entered CPU_SHAKE mode.
 dn_6007  | Successfully entered CPU_SHAKE mode.
 dn_6002  | Current is in CPU_SHAKE mode.
 dn_6003  | Current is in CPU_SHAKE mode.
 dn_6005  | Current is in CPU_SHAKE mode.
 dn_6006  | Current is in CPU_SHAKE mode.
 dn_6009  | Current is in CPU_SHAKE mode.
 dn_6008  | Current is in CPU_SHAKE mode.
(12 rows)
```


版权所有 © 华为云计算技术有限公司

说明

- 执行gs_perf_start_shaking_collect()、gs_perf_do_shaking_collect()、gs_perf_close_shaking_collect()、gs_perf_get_shaking_collect_status()需要gs_perf_service正常运行。同时gs_perf_service在采集过程中，需要申请环形buffer，该buffer的大小由cpu核心数和gs_perf_start_shaking_collect()的入参ringbuffer_size控制。
- 执行gs_perf_start_shaking_collect()、gs_perf_do_shaking_collect()、gs_perf_close_shaking_collect()、gs_perf_get_shaking_collect_status()命令过程中手动中止或者消息返回超时只会中断前端SQL，不会终止后台perf_service进程执行对应任务。在gs_perf_do_shaking_collect()执行过程中可以通过sudo kill -9 perf_service进程PID方式关闭perf_service进程，强制打断后台任务。
- 开启、关闭性能抖动监控以及保存性能抖动火焰图以物理机器为单位。因此，在某个节点上开启、关闭和保存时，会影响同一物理机器上其他节点的监控状态。例如，在某节点上开启性能抖动监控后，同一机器上其他节点也会进入监控状态并且不能再次开启监控。
- 当scope参数为all时，开启、关闭性能抖动监控以及保存性能抖动火焰图的实现方式为顺序遍历所有节点，在每个节点上执行对应的开启、关闭和保存操作。因此，同一机器上只有一个节点会返回成功消息，其他节点会返回重复执行的失败消息。
- 集群中某节点故障后，调用系统函数且scope为all，故障节点不会执行相应操作，正常节点会执行相应操作，互不影响。返回结果中故障节点node_name字段为node_host:node_port。
- 当发生被监控的SQL触发慢SQL并采集所有节点性能抖动数据时，会顺序遍历所有节点并采集节点所在机器性能抖动数据，遍历顺序不固定，最终保证每个物理机器都有一个节点采集了性能抖动数据。
- 升级部分分片之后，可以在升级上的分片上执行新增性能抖动火焰图函数，返回结果中升级上的分片正常返回。
- 采集进程gs_perf_service重启之后，性能抖动监控会被解除。如果希望再次监控，需要重新执行开启性能抖动监控的命令。
- 使用gs_perf_start_shaking_collect()函数启动性能抖动监控后，数据库需要约1秒的时间完成火焰图资源的初始化，随后采集性能抖动火焰图文件。

- gs_perf_do_shaking_collect(scope text)

  描述：用于执行性能抖动多核多线程火焰图落盘，即保存性能抖动快照，返回是否保存成功的结果。依赖于gs_perf_start_shaking_collect()接口，开启性能抖动监控才能执行性能抖动快照落盘。两次落盘最小时间间隔为1min。为防止后台采集进程hang住，当单节点执行性能抖动火焰图生成动作时，后台采集进程gs_perf_service超过10min未完成，或者gs_perf_service在执行性能抖动采集时异常退出，会返回超时消息。执行该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

  表 1-247 gs_perf_do_shaking_collect 参数说明

参数名称	类型	描述	范围
scope	text	local代表本节点的perf数据落盘，all代表开始所有节点的perf数据落盘，all仅支持在分布式CN上使用。必填，填NULL则为local。	all或local

  返回值说明如下：

版权所有 © 华为云计算技术有限公司

表 1-248 gs_perf_do_shaking_collect 返回值说明

名称	类型	描述
nodename	text	节点名称。
message	text	返回值为“Successfully collect perf detail data .....”，表示性能抖动快照保存成功。 返回值为“Current is not in CPU_SHAKE mode.”，表示未在监控状态，性能抖动快照保存失败。 返回值为“Skipping collection this time .....”，表示1min内重复保存。 返回值为“Unexpected Result: do_shaking_collect error. Details: 详细报错消息”，表示执行失败。

示例：

所有节点执行性能抖动快照落盘。返回保存成功、快照路径和落盘时间。

```
gaussdb=# SELECT * FROM gs_perf_do_shaking_collect('all');
 nodename |
message

----------
+----------------------------------------------------------------------------------------------------------------------------------
-------------
-------------------------------------------------------------
 cn_5001  | Skipping collection this time. Reason: The data has already been collected just now.
 cn_5002  | Skipping collection this time. Reason: The data has already been collected just now.
 cn_5003  | Skipping collection this time. Reason: The data has already been collected just now.
 dn_6001  | Successfully collect perf detail data. The file path is: $GAUSSLOG/gs_flamegraph/dn_6001/
shake_detail-2025-11-27_173849.lz4, collect time is 2025-11-27_173849
 dn_6004  | Successfully collect perf detail data. The file path is: $GAUSSLOG/gs_flamegraph/dn_6004/
shake_detail-2025-11-27_173850.lz4, collect time is 2025-11-27_173850
 dn_6007  | Skipping collection this time. Reason: The data has already been collected just now.
 dn_6003  | Successfully collect perf detail data. The file path is: $GAUSSLOG/gs_flamegraph/dn_6003/
shake_detail-2025-11-27_173850.lz4, collect time is 2025-11-27_173850
 dn_6002  | Skipping collection this time. Reason: The data has already been collected just now.
 dn_6005  | Skipping collection this time. Reason: The data has already been collected just now.
 dn_6006  | Skipping collection this time. Reason: The data has already been collected just now.
 dn_6009  | Skipping collection this time. Reason: The data has already been collected just now.
 dn_6008  | Skipping collection this time. Reason: The data has already been collected just now.
(12 rows)
```

- gs_perf_close_shaking_collect(scope text)

  描述：用于关闭性能抖动监控，并返回关闭操作是否成功。该函数依赖于gs_perf_start_shaking_collect()接口，必须先启动监控才能关闭，否则将返回未在监控中的提示信息。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

版权所有 © 华为云计算技术有限公司

表 1-249 gs_perf_close_shaking_collect 参数说明

参数名称	类型	描述	范围
scope	text	local代表结束本节点监控，all代表结束所有节点监控，all仅支持在分布式CN上使用。必填，填NULL则为local。	local或all

返回值说明如下：

表 1-250 gs_perf_close_shaking_collect 返回值说明

名称	类型	描述
nodename	text	节点名称。
message	text	返回值为“Successfully exit CPU_SHAKE mode.”，表示性能抖动监控关闭成功。 返回值为“Current is not in CPU_SHAKE mode.”，表示未在监控状态，关闭失败。 返回值为“Unexpected Result: close_shaking_collect error. Detail: 详细报错消息”，表示关闭失败。

示例：

```
gaussdb=# SELECT * FROM gs_perf_close_shaking_collect('all');
 nodename |              message
----------+-----------------------------------
 cn_5001  | Current is not in CPU_SHAKE mode.
 cn_5002  | Current is not in CPU_SHAKE mode.
 cn_5003  | Current is not in CPU_SHAKE mode.
 dn_6001  | Successfully exit CPU_SHAKE mode.
 dn_6004  | Current is not in CPU_SHAKE mode.
 dn_6007  | Current is not in CPU_SHAKE mode.
 dn_6002  | Current is not in CPU_SHAKE mode.
 dn_6003  | Current is not in CPU_SHAKE mode.
 dn_6005  | Successfully exit CPU_SHAKE mode.
 dn_6006  | Current is not in CPU_SHAKE mode.
 dn_6009  | Successfully exit CPU_SHAKE mode.
 dn_6008  | Current is not in CPU_SHAKE mode.
(12 rows)
```

- gs_perf_get_shaking_collect_result(scope text)

  描述：用于查看最近一次监控中的性能抖动火焰图采集历史文件。每次开启性能抖动监控时，会清空性能抖动快照采集历史，返回结果仅包括正常节点。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

版权所有 © 华为云计算技术有限公司

表 1-251 gs_perf_get_shaking_collect_result 参数说明

参数名称	类型	描述	范围
scope	text	local代表查询本节点性能抖动火焰图快照历史，all代表查询所有节点性能抖动火焰图快照历史，all仅支持在分布式CN上使用。必填，填NULL则为local。	local或all

返回值说明如下：

表 1-252 gs_perf_get_shaking_collect_result 返回值说明

名称	类型	描述
nodename	text	节点名称。
num	text	采集序号。
filename	text	性能抖动快照文件名。
shake_time	text	性能抖动快照落盘时间。

示例：

查看最近一次监控中的性能抖动火焰图采集历史。

```
gaussdb=# SELECT * FROM gs_perf_get_shaking_collect_result('all');
 nodename | num |               filename               |    shake_time
----------+-----+--------------------------------------+------------------
 dn_6001  | 1   | shake_detail-2025-11-28_160835.lz4 | 2025-11-28_160835
 dn_6004  | 1   | shake_detail-2025-11-28_160835.lz4 | 2025-11-28_160835
 dn_6003  | 1   | shake_detail-2025-11-28_160835.lz4 | 2025-11-28_160835
(3 rows)
```

- gs_perf_get_shaking_collect_status(scope text)

  描述：用于查看节点性能抖动监控状态。查询该函数需要有SYSADMIN权限或者MONADMIN权限。该函数在ROOT模式下可用，在DBUSER/DISABLE模式下不可用。

表 1-253 gs_perf_get_shaking_collect_status 参数说明

参数名称	类型	描述	范围
scope	text	local代表查询本节点所在机器的采集状态，all代表查询所有节点所在机器的采集状态，all仅用于分布式。必填，填NULL，如果执行节点为CN则为all，否则为local。	local或all

版权所有 © 华为云计算技术有限公司

表 1-254 gs_perf_get_shaking_collect_status 返回值说明

名称	类型	描述
nodename	text	节点名称。
status	text	该节点所在机器是否已开启性能抖动监控。 ● NOT IN SHAKE MODE：未在监控。 ● IN SHAKE MODE：已在监控中。 ● NO PERF SERVICE：采集进程未响应。 ● Unexpected Result: get_shaking_connect_status error. Detail: 详细报错消息：表示查询失败。

示例：

查看该节点所在机器是否已开启性能抖动监控。

```
gaussdb=# SELECT * FROM gs_perf_get_shaking_collect_status('all');
 nodename |    status
----------+---------------
 cn_5001  | IN SHAKE MODE
 cn_5002  | IN SHAKE MODE
 cn_5003  | IN SHAKE MODE
 dn_6001  | IN SHAKE MODE
 dn_6004  | IN SHAKE MODE
 dn_6007  | IN SHAKE MODE
 dn_6002  | IN SHAKE MODE
 dn_6003  | IN SHAKE MODE
 dn_6005  | IN SHAKE MODE
 dn_6006  | IN SHAKE MODE
 dn_6009  | IN SHAKE MODE
 dn_6008  | IN SHAKE MODE
(12 rows)
```

- gs_tpworker_execstmt_stat()

  描述：描述语句的运行时信息，SYSADMIN和MONADMIN用户执行则显示全部正在执行的语句的信息，普通用户查询只能查询自己执行的SQL语句的信息。

  返回值类型：setof record

名称	类型	描述
db_oid	oid	用户会话在后台连接到的数据库OID。
db_name	name	用户会话在后台连接到的数据库名称。
threadpool_worker	varchar	线程所属的numagroup和线程的ID，格式如下：numagroup_threadid。
thread_id	bigint	线程ID。
session_id	bigint	会话ID。
query_id	bigint	正在执行的SQL语句的ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
query_text	text	正在执行的SQL语句内容。
unique_sql_id	bigint	SQL语句生成的唯一id。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_app_name	text	客户端app的名字。
stmt_slow_time_threshold	int	单位毫秒，SQL语句被标记为慢SQL的预设超时时间。
stmt_start_time	timestamp with time zone	语句执行的开始时间。
stmt_elapse_time	bigint	距离查询开始执行时已经过去的时间。
stmt_control_status	varchar	当前语句状态： ● Waiting：等待状态，session接入未获得线程执行。 ● Running：当前语句正常执行。 ● Control：当前语句进入了资源管控阶段。
stmt_control_rule	text	当前语对应的慢SQL管控规则。
stmt_control_iostat	text	当前语句的iops值和最大iops上限。格式如下：curVal/maxVal。
stmt_control_memstat	text	当前字段预留，暂不支持。
stmt_control_cpustat	text	当前字段预留，暂不支持。
stmt_control_netstat	text	当前字段预留，暂不支持。

- gs_tpworker_execslot_stat()

  描述：描述线程的运行时信息，SYSADMIN和MONADMIN用户执行则显示全部线程的信息，普通用户查询只能查询自己执行的SQL语句所在的线程的信息。

  返回值类型：setof record

名称	类型	描述
numagroup	int	当前线程所属的numagroup。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
worker_id	int	当前线程的ID。
worker_bind_type	text	线程绑定方式numabind, cpubind, allbind，nobind。
worker_cpu_affinity	text	线程和CPU的亲和性，即线程可以调度的cpu核数范围。
worker_status	varchar	当前线程状态： ● Waiting：等待状态，session接入未获得线程执行。 ● Running：当前语句正常执行。 ● Control：当前语句进入了资源管控阶段。
served_query_id	bigint	正在执行的SQL语句的ID。
served_query_text	text	正在执行的SQL语句内容。

- gs_session_all_settings(sessionid bigint)

  描述：查询本节点上sessionid对应的session的全量GUC参数配置。需要SYSADMIN或者MONADMIN权限执行。

  入参说明：sessionid，会话ID。

  返回值类型：setof record

  函数返回字段说明如下：

名称	类型	描述
name	text	参数名称。
setting	text	参数当前值。
unit	text	参数的隐式单位。

  示例：

```
gaussdb=# CREATE USER testuser WITH PASSWORD '********' sysadmin;
gaussdb=# SET SESSION AUTHORIZATION testuser PASSWORD '********';
gaussdb=> SELECT sessionid FROM pg_stat_activity WHERE usename = 'testuser';
 sessionid
-----------
    788861
(1 row)

gaussdb=> SELECT * FROM gs_session_all_settings(788861) WHERE name = 'work_mem';
   name   | setting | unit
----------+---------+------
 work_mem | 131072  | kB
(1 row)

gaussdb=> RESET SESSION AUTHORIZATION;
gaussdb=# DROP USER testuser;
```


版权所有 © 华为云计算技术有限公司

- gs_session_all_settings()

  描述：查询本节点上所有session的全量GUC参数配置。需要SYSADMIN或者MONADMIN权限执行。

  返回值类型：setof record

名称	类型	描述
sessionid	bigint	会话的ID。
pid	bigint	后端线程的ID。
name	text	参数名称。
setting	text	参数当前值。
unit	text	参数的隐式单位。

  示例：

```
gaussdb=# SELECT * FROM gs_session_all_settings() WHERE name = 'work_mem';
    sessionid    |       pid       |   name   | setting | unit
-----------------+-----------------+----------+---------+------
 140550214145792 |           96974 | work_mem | 65536   | kB
 140550214145792 |           96971 | work_mem | 65536   | kB
 140549731735296 | 140549731735296 | work_mem | 65536   | kB
 140549764413184 | 140549764413184 | work_mem | 65536   | kB
(4 rows)
```

- gs_local_wal_preparse_statistics()

  描述：查询本节点上日志预解析线程最近一次启动，预解析日志的情况。需要SYSADMIN权限执行。日志预解析线程通常在备DN双机复制链路断开时，由CM组件发送启动命令启动。如果日志预解析线程已经被启动，可以通过调用该函数查询最近一次的预解析状态。如果线程尚未启动，该函数将返回默认值。在容灾场景下，灾备集群的DN和CN会自动启动日志预解析线程。需要注意的是，CN仅在灾备集群中会启动日志预解析线程。

  返回值类型：setof record

名称	类型	描述
preparser_term	text	最近一次预解析日志得到的最大term值。
preparser_start_time	timestamptz	最近一次预解析启动时间。
preparser_end_time	timestamptz	最近一次预解析结束时间。
preparser_start_location	text	最近一次预解析日志起始位置。
preparser_end_location	text	最近一次预解析日志结束位置。
preparser_total_bytes	int8	最近一次预解析日志量，单位：byte。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
preparser_speed	int8	最近一次预解析速度，单位：byte/ms。
is_valid	bool	最近一次预解析结果是否可以用于选主。

拉起过日志预解析线程后的查询结果示例：

```
gaussdb=# SELECT * FROM gs_local_wal_preparse_statistics();
 preparser_term |     preparser_start_time      |      preparser_end_time       | preparser_start_location |
preparser_end_location | preparser_total_bytes | preparser_speed | is_valid
----------------+-------------------------------+-------------------------------+-------------------------
+------------------------+-----------------------+-----------------+----------
 3107           | 2023-02-01 17:04:23.367946+08 | 2023-02-01 17:04:25.354434+08 | 00000003/
C3EEA660        | 00000004/0BE60738      |            1207394520 |      1207394520 | f
(1 row)
```

未拉起过日志预解析线程时的查询查询结果示例：

```
gaussdb=# SELECT * FROM gs_local_wal_preparse_statistics();
 preparser_term |  preparser_start_time  |   preparser_end_time   | preparser_start_location |
preparser_end_location | preparser_total_bytes | preparser_speed | is_valid
----------------+------------------------+------------------------+--------------------------
+------------------------+-----------------------+-----------------+----------
 0              | 2000-01-01 08:00:00+08 | 2000-01-01 08:00:00+08 | 00000000/00000000        |
00000000/00000000      |                     0 |               0 | f
(1 row)
```

- gs_hot_standby_space_info()

  描述：查询standby_read/base_page，standby_read/block_info_meta，standby_read/lsn_info_meta文件夹中的文件总数和总大小。

  返回值类型：setof record

名称	类型	描述
base_page_file_num	xid	base_page_file的总数量。
base_page_total_size	xid	base_page_file的总大小。
lsn_info_meta_file_num	xid	lsn_info_meta_file的总数量。
lsn_info_meta_total_size	xid	lsn_info_meta_file的总大小。
block_info_meta_file_num	xid	block_info_meta_file的总数量。
block_info_meta_total_size	xid	block_info_meta_file的总大小。

示例：

```
gaussdb=# SELECT * FROM gs_hot_standby_space_info();
 base_page_file_num | base_page_total_size | lsn_info_meta_file_num | lsn_info_meta_total_size |
block_info_meta_file_num | block_info_meta_total_size
--------------------+----------------------+------------------------+--------------------------
```


版权所有 © 华为云计算技术有限公司

```
+-------------------------+-------------------------
               6 |             163840 |                 6 |                3136 |                  16
|                147456
(1 row)
```

- exrto_file_read_stat()

  描述：查询备机读新增的base page file、lsn info meta file和block info meta file三种类型的文件磁盘访问次数和访问总时延。连接备DN或灾备集群CN查询，其他情况查询结果为0。

  返回值类型：setof record

名称	类型	描述
lsn_info_page_disk_read_counter	int8	lsn info meta file的磁盘访问次数。
lsn_info_page_disk_read_dur	int8	lsn info meta file的磁盘访问总时延。
blk_info_meta_disk_read_counter	int8	block info meta file的磁盘访问次数。
blk_info_meta_disk_read_dur	int8	block info meta file的磁盘访问总时延。
base_page_read_disk_counter	int8	base page file的磁盘访问次数。
base_page_read_disk_dur	int8	base page file的磁盘访问总时延。

  示例：

```
gaussdb=# SELECT * FROM exrto_file_read_stat();
 lsn_info_page_disk_read_counter | lsn_info_page_disk_read_dur | blk_info_meta_disk_read_counter |
blk_info_meta_disk_read_dur | base_page_read_disk_counter | base_page_read_disk_dur
---------------------------------+-----------------------------+---------------------------------
+-----------------------------+-----------------------------+-------------------------
                      14987 |                 92313 |                  23879 |                 129811
|                       0 |                   0
(1 row)
```

- gs_exrto_recycle_info()

  描述：查询资源回收位置，其中包括每个线程的回收lsn，全局回收的lsn，查询线程最旧的快照的lsn。连接备DN或灾备集群CN查询，其他情况查询结果为0。

  返回值类型：setof record

名称	类型	描述
page_redo_worker_thread_id	text	redo线程的回收lsn位置，其中thread_id为redo线程的线程id。
global_recycle_lsn	text	全局回收位置的lsn。
exrto_snapshot_oldest_lsn	text	查询线程的最旧的快照lsn。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT * FROM gs_exrto_recycle_info();
            thread_id             | recycle_lsn
----------------------------------+-------------
 page_redo_worker_140148895381248 | 0/7B4552E0
 page_redo_worker_140148872312576 | 0/7B4535B8
 global_recycle_lsn               | 0/7B4535B8
 exrto_snapshot_oldest_lsn        | 0/8488E6D0
(4 rows)
```

- gs_stat_get_db_conflict_all(oid)

  描述：查询发送回放冲突信号的数量。

  参数：输入参数为数据库的oid。

  返回值类型：setof record

名称	类型	描述
conflict_all	int8	发送回放冲突信号的总数量。
conflict_tablespace	int8	发送tablespace类型回放冲突信号的数量。
conflict_lock	int8	发送lock类型回放冲突信号的数量。
conflict_snapshot	int8	发送snapshot类型回放冲突信号的数量。
conflict_bufferpin	int8	发送bufferpin类型回放冲突信号的数量。
conflict_startup_deadlock	int8	发送startup_deadlock类型回放冲突信号的数量。
conflict_truncate	int8	发送truncate类型回放冲突信号的数量。
conflict_standby_query_timeout	int8	发送standby_query_timeout类型回放冲突信号的数量。
conflict_force_recycle	int8	发送force_recycle类型回放冲突信号的数量。

示例：

```
gaussdb=# SELECT * FROM gs_stat_get_db_conflict_all(12738);
 conflict_all | conflict_tablespace | conflict_lock | conflict_snapshot | conflict_bufferpin |
conflict_startup_deadlock | conflict_truncate | conflict_standby_query_timeout | conflict_force_recycle
--------------+---------------------+---------------+-------------------+--------------------
+---------------------------+-------------------+--------------------------------+------------------------
            0 |                 0 |             0 |                 0 |                 0 |                         0 |                0
|                         0 |                   0
(1 row)
```

- gs_redo_stat_info([int])

  描述：查询WAL日志回放的相关统计信息。包括回放线程的缓存命中率、回放DDL日志删除文件数量、回放线程读取数据页时产生物理读、回放线程读取WAL日志的等待事件信息。需要连接备DN查询。

  返回值类型：setof record

版权所有 © 华为云计算技术有限公司

参数类型	名称	类型	描述
输入参数	operation	int	可选输入，若无输入则行为是查询统计信息。 ● 1：表示查询统计信息。 ● 2：表示重置缓存命中率的统计信息。
输出参数	buffer_hit_rate	float8	回放线程的缓存命中率。
输出参数	ddl_unlink_nrels_count	int8	回放DDL日志删除文件数量。
输出参数	read_buffer_io_counter	int8	回放线程读取数据页时产生物理读的waitevent触发次数。
输出参数	read_buffer_io_total_dur	int8	回放线程读取数据页时产生物理读的waitevent总用时。
输出参数	read_buffer_io_avg_dur	int8	回放线程读取数据页时产生物理读的waitevent平均用时。
输出参数	read_buffer_io_min_dur	int8	回放线程读取数据页时产生物理读的waitevent最小用时。
输出参数	read_buffer_io_max_dur	int8	回放线程读取数据页时产生物理读的waitevent最大用时。
输出参数	read_wal_buf_counter	int8	回放线程读取WAL日志的waitevent触发次数。
输出参数	read_wal_buf_total_dur	int8	回放线程读取WAL日志的waitevent总用时。
输出参数	read_wal_buf_avg_dur	int8	回放线程读取WAL日志的waitevent平均用时。
输出参数	read_wal_buf_min_dur	int8	回放线程读取WAL日志的waitevent最小用时。
输出参数	read_wal_buf_max_dur	int8	回放线程读取WAL日志的waitevent最大用时。

示例：

```
gaussdb=# SELECT * FROM gs_redo_stat_info();
-[ RECORD 1 ]------------+----------
buffer_hit_rate          | 70.5707
ddl_unlink_nrels_count   | 3
read_buffer_io_counter   | 1732
read_buffer_io_total_dur | 2850806
read_buffer_io_avg_dur   | 1645
read_buffer_io_min_dur   | 3
read_buffer_io_max_dur   | 981639
read_wal_buf_counter     | 9779
read_wal_buf_total_dur   | 193612470
read_wal_buf_avg_dur     | 19798
```


版权所有 © 华为云计算技术有限公司

```
read_wal_buf_min_dur     | 3
read_wal_buf_max_dur     | 1914777
```

- gs_recovery_conflict_waitevent_info()

  描述：查询处理回放冲突的函数的waitevent相关信息。需要连接备DN或灾备集群CN查询。

  返回值类型：setof record

名称	类型	描述
conflict_lock_counter	int8	处理lock类型回放冲突的触发次数。
conflict_lock_total_dur	int8	处理lock类型回放冲突的总用时。
conflict_lock_avg_dur	int8	处理lock类型回放冲突的平均用时。
conflict_lock_min_dur	int8	处理lock类型回放冲突的最小用时。
conflict_lock_max_dur	int8	处理lock类型回放冲突的最大用时。
conflict_snapshot_counter	int8	处理snapshot类型回放冲突的触发次数。
conflict_snapshot_total_dur	int8	处理snapshot类型回放冲突的总用时。
conflict_snapshot_avg_dur	int8	处理snapshot类型回放冲突的平均用时。
conflict_snapshot_min_dur	int8	处理snapshot类型回放冲突的最小用时。
conflict_snapshot_max_dur	int8	处理snapshot类型回放冲突的最大用时。
conflict_tablespace_counter	int8	处理tablespace类型回放冲突的触发次数。
conflict_tablespace_total_dur	int8	处理tablespace类型回放冲突的总用时。
conflict_tablespace_avg_dur	int8	处理tablespace类型回放冲突的平均用时。
conflict_tablespace_min_dur	int8	处理tablespace类型回放冲突的最小用时。
conflict_tablespace_max_dur	int8	处理tablespace类型回放冲突的最大用时。
conflict_database_counter	int8	处理database类型回放冲突的触发次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
conflict_database_total_dur	int8	处理database类型回放冲突的总用时。
conflict_database_avg_dur	int8	处理database类型回放冲突的平均用时。
conflict_database_min_dur	int8	处理database类型回放冲突的最小用时。
conflict_database_max_dur	int8	处理database类型回放冲突的最大用时。
conflict_truncate_counter	int8	处理truncate类型回放冲突的触发次数。
conflict_truncate_total_dur	int8	处理truncate类型回放冲突的总用时。
conflict_truncate_avg_dur	int8	处理truncate类型回放冲突的平均用时。
conflict_truncate_min_dur	int8	处理truncate类型回放冲突的最小用时。
conflict_truncate_max_dur	int8	处理truncate类型回放冲突的最大用时。
conflict_standby_query_timeout_counter	int8	处理standby_query_timeout类型回放冲突的触发次数。
conflict_standby_query_timeout_total_dur	int8	处理standby_query_timeout类型回放冲突的总用时。
conflict_standby_query_timeout_avg_dur	int8	处理standby_query_timeout类型回放冲突的平均用时。
conflict_standby_query_timeout_min_dur	int8	处理standby_query_timeout类型回放冲突的最小用时。
conflict_standby_query_timeout_max_dur	int8	处理standby_query_timeout类型回放冲突的最大用时。
conflict_force_recycle_counter	int8	处理force_recycle类型回放冲突的触发次数。
conflict_force_recycle_total_dur	int8	处理force_recycle类型回放冲突的总用时。
conflict_force_recycle_avg_dur	int8	处理force_recycle类型回放冲突的平均用时。
conflict_force_recycle_min_dur	int8	处理force_recycle类型回放冲突的最小用时。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
conflict_force_recycle_max_dur	int8	处理force_recycle类型回放冲突的最大用时。
conflict_bufferpin_counter	int8	处理buffer pin类型回放冲突的触发次数。
conflict_bufferpin_total_dur	int8	处理buffer pin类型回放冲突的总用时。
conflict_bufferpin_avg_dur	int8	处理buffer pin类型回放冲突的平均用时。
conflict_bufferpin_min_dur	int8	处理buffer pin类型回放冲突的最小用时。
conflict_bufferpin_max_dur	int8	处理buffer pin类型回放冲突的最大用时。

示例：

```
gaussdb=# SELECT * FROM gs_recovery_conflict_waitevent_info();
-[ RECORD 1 ]----------------------------+------
conflict_lock_counter                    | 0
conflict_lock_total_dur                  | 0
conflict_lock_avg_dur                    | 0
conflict_lock_min_dur                    | 0
conflict_lock_max_dur                    | 0
conflict_snapshot_counter                | 0
conflict_snapshot_total_dur              | 0
conflict_snapshot_avg_dur                | 0
conflict_snapshot_min_dur                | 0
conflict_snapshot_max_dur                | 0
conflict_tablespace_counter              | 0
conflict_tablespace_total_dur            | 0
conflict_tablespace_avg_dur              | 0
conflict_tablespace_min_dur              | 0
conflict_tablespace_max_dur              | 0
conflict_database_counter                | 0
conflict_database_total_dur              | 0
conflict_database_avg_dur                | 0
conflict_database_min_dur                | 0
conflict_database_max_dur                | 0
conflict_truncate_counter                | 6
conflict_truncate_total_dur              | 35872
conflict_truncate_avg_dur                | 5978
conflict_truncate_min_dur                | 5130
conflict_truncate_max_dur                | 7459
conflict_standby_query_timeout_counter   | 0
conflict_standby_query_timeout_total_dur | 0
conflict_standby_query_timeout_avg_dur   | 0
conflict_standby_query_timeout_min_dur   | 0
conflict_standby_query_timeout_max_dur   | 0
conflict_force_recycle_counter           | 0
conflict_force_recycle_total_dur         | 0
conflict_force_recycle_avg_dur           | 0
conflict_force_recycle_min_dur           | 0
conflict_force_recycle_max_dur           | 0
conflict_bufferpin_counter               | 0
conflict_bufferpin_total_dur             | 0
conflict_bufferpin_avg_dur               | 0
```


版权所有 © 华为云计算技术有限公司

```
conflict_bufferpin_min_dur              | 0
conflict_bufferpin_max_dur              | 0
```

- gs_display_delay_ddl_info()

  描述：查看备机中延迟删除的文件信息。

  返回值类型：setof record

名称	类型	描述
type	INT4	删除操作的对象是表或数据库。
lsn	TEXT	标识特定日志文件记录在此日志文件中的位置。
tablespace	INT4	数据库中用于存储表和索引的物理空间。
database	INT4	该数据库的物理存储位置。
relation	INT4	数据库中的对象，可以是表、视图、索引的物理位置。
bucketid	INT4	指定关系对象所属的bucket。
opt	INT4	压缩表相关属性。
forknum	INT4	主体命名之后的后缀命名，通过主体命名和后缀命名，可以找到唯一的物理文件。

  示例：

```
gaussdb=# SELECT * FROM gs_display_delay_ddl_info();
 type | lsn | tablespace | database | relation | bucketid | opt | forknum
------+-----+------------+----------+----------+----------+-----+---------
(0 rows)
```

- gs_plsql_statements(sessionid bigint, pkg id oid, function id oid)

  描述：返回存储过程相关内存信息。

  返回值类型：setof record

  函数返回字段说明：

名称	类型	描述
pid	bigint	后台线程id。
sessionid	bigint	会话id。
global_sessionid	text	全局会话id。
databaseid	oid	数据库id。
pkg_id	oid	存储过程所属PACKAGE id，如果不属于package则为0。
func_id	oid	存储过程id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
func_signature	text	存储过程名称（name + fn_collation + search_path）。
plsql_compiled_totalsize	bigint	存储过程编译产物的总内存大小。
plsql_plan_totalsize	bigint	存储过程内计划的总内存大小。

示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_assign AS
    a int := 0;
BEGIN
    a = a + 1;
    RAISE INFO '%', a;
END;
/
gaussdb=# CALL proc_assign();
INFO:  1
 proc_assign
-------------

(1 row)

gaussdb=# SELECT * FROM pg_catalog.gs_plsql_statements(pg_current_sessid());
    pid    | sessionid | global_sessionid | databaseid | pkg_id | func_id |
func_signature
            | plsql_compiled_totalsize | plsql_plan_totalsize
-----------+-----------+------------------+------------+--------+---------
+-------------------------------------------------------------------------
------------+--------------------------+----------------------
 536868608 |      5052 | 0:0#0            |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |                    8192 |                65536
(1 row)

-- 需要过滤function id --
gaussdb=# SELECT * FROM pg_catalog.gs_plsql_statements(NULL, NULL, 16854);
    pid    | sessionid | global_sessionid | databaseid | pkg_id | func_id |
func_signature
            | plsql_compiled_totalsize | plsql_plan_totalsize
-----------+-----------+------------------+------------+--------+---------
+-------------------------------------------------------------------------
------------+--------------------------+----------------------
 536868608 |      5052 | 0:0#0            |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |                    8192 |                65536
(1 row)

gaussdb=# DROP PROCEDURE proc_assign;
```

- gs_global_plsql_statements(sessionid bigint, pkg id oid, function id oid)

  描述：返回存储过程相关内存信息。

  返回值类型：setof record

  函数返回字段说明：

名称	类型	描述
node_name	text	当前节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pid	bigint	后台线程id。
sessionid	bigint	会话id。
global_sessionid	text	全局会话id。
databaseid	oid	数据库id。
pkg_id	oid	存储过程所属PACKAGE id，如果不属于PACKAGE则为0。
func_id	oid	存储过程id。
func_signature	text	存储过程名称（name + fn_collation + search_path）。
plsql_compiled_totalsize	bigint	存储过程编译产物的总内存大小。
plsql_plan_totalsize	bigint	存储过程内计划的总内存大小。

示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_assign AS
    a int := 0;
BEGIN
    a = a + 1;
    RAISE INFO '%', a;
END;
/
gaussdb=# CALL proc_assign();
INFO:  1
 proc_assign
-------------

(1 row)

gaussdb=# SELECT * FROM pg_catalog.gs_global_plsql_statements(pg_current_sessid());
  node_name   |     pid     |    sessionid     | global_sessionid | databaseid | pkg_id | func_id
|
func_signature                                          | plsql_compiled_totalsize | plsql_plan_totalsize
--------------+-------------+------------------+------------------+------------+--------+---------
+-----------------------------------------------
-------------------------------------------------------------------+--------------------------+----------------------
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   16387 | name:
proc_assign(), fn_collation: 0, search_path: addUser: false, schemas: 2200
|                   24576 |                64512
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   13509 | name:
gs_global_plsql_statements(bigint,oid,oid), fn_collation: 0, search_path: addUser: false, schemas: 2200
|                  122880 |               216064
(2 rows)

gaussdb=# DROP PROCEDURE proc_assign;
```

- gs_plsql_statements_details(sessionid bigint, pkg id oid, function id oid)
  描述：返回存储过程计划缓存语句相关内存信息。

版权所有 © 华为云计算技术有限公司

说明

查询函数时，GaussDB会根据current_user的权限来决定是否允许访问。如果current_user是初始用户或系统管理员, 则可以查询所有会话信息。如果current_user是普通用户,则只能查看自己的会话信息(即session_user是当前用户的信息)。

返回值类型：setof record

函数返回字段说明：

名称	类型	描述
pid	bigint	后台线程id。
sessionid	bigint	会话id。
global_sessionid	text	全局会话id。
databaseid	oid	数据库id。
pkg_id	oid	存储过程所属PACKAGE id，如果不属于PACKAGE则为0。
func_id	oid	存储过程id。
func_signature	text	存储过程名称（name + fn_collation + search_path）。
lineno	bigint	语句在存储过程内行号。
query	text	计划缓存语句。
usedsize	bigint	计划缓存语句使用内存大小。
totalsize	bigint	计划缓存语句总内存大小。

示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_assign AS
    a int := 0;
BEGIN
    a = a + 1;
    RAISE INFO '%', a;
END;
/
gaussdb=# CALL proc_assign();
INFO:  1
 proc_assign
-------------


(1 row)

gaussdb=# SELECT * FROM pg_catalog.gs_plsql_statements_details(pg_current_sessid());
    pid     | sessionid | global_sessionid | databaseid | pkg_id | func_id |
func_signature
            | lineno |     query      | usedsize | totalsize
------------+-----------+------------------+------------+--------+---------
+-----------------------------------------------------------------------
-------------+--------+--------------+----------+-----------
 536868608 |      5251 | 0:0#0            |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |       3 | SELECT a + 1 |     15064 |      25600
```


版权所有 © 华为云计算技术有限公司

```
 536868608 |      5251 | 0:0#0           |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |      4 | SELECT a     |    12320 |     17408
 536868608 |      5251 | 0:0#0           |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |      1 | SELECT 0     |    12088 |     21504
(3 rows)

-- 需要过滤function id --
gaussdb=# SELECT * FROM pg_catalog.gs_plsql_statements(NULL, NULL, 16854);
    pid    | sessionid | global_sessionid | databaseid | pkg_id | func_id |
func_signature
           | lineno |    query     | usedsize | totalsize
-----------+-----------+------------------+------------+--------+---------
+-------------------------------------------------------------------------
-----------+--------+--------------+----------+-----------
 536868608 |      5251 | 0:0#0           |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |      3 | SELECT a + 1 |    15064 |     25600
 536868608 |      5251 | 0:0#0           |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |      4 | SELECT a     |    12320 |     17408
 536868608 |      5251 | 0:0#0           |      13108 |      0 |   16854 | name: proc_assign(), fn_collation: 0,
search_path: addUser: false, schemas: 2200 |      1 | SELECT 0     |    12088 |     21504
(3 rows)

gaussdb=# DROP PROCEDURE proc_assign;
```

- gs_global_plsql_statements_details(sessionid bigint, pkg id oid, function id oid)

  描述：返回存储过程计划缓存语句相关内存信息。

  说明

  查询函数时，GaussDB会根据current_user的权限来决定是否允许访问。如果current_user是初始用户或系统管理员, 则可以查询所有会话信息。如果current_user是普通用户,则只能查看自己的会话信息(即session_user是当前用户的信息)。

  返回值类型：setof record

  函数返回字段说明：

名称	类型	描述
node_name	text	当前节点名称。
pid	bigint	后台线程id。
sessionid	bigint	会话id。
global_sessionid	text	全局会话id。
databaseid	oid	数据库id。
pkg_id	oid	存储过程所属PACKAGE id，如果不属于PACKAGE则为0。
func_id	oid	存储过程id。
func_signature	text	存储过程名称（name + fn_collation + search_path）。
lineno	bigint	语句在存储过程内行号。
query	text	计划缓存语句。
usedsize	bigint	计划缓存语句使用内存大小。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
totalsize	bigint	计划缓存语句总内存大小。

示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_assign AS
    a int := 0;
BEGIN
    a = a + 1;
    RAISE INFO '%', a;
END;
/
gaussdb=# CALL proc_assign();
INFO:  1
 proc_assign
-------------

(1 row)

gaussdb=# SELECT * FROM pg_catalog.gs_global_plsql_statements_details(pg_current_sessid());
  node_name   |    pid     |    sessionid     | global_sessionid | databaseid | pkg_id | func_id
|
    func_signature                                                  | lineno
|
                                query                                                                   | use
dsize | totalsize
--------------+------------+-----------------+------------------+------------+--------+---------
+-----------------------------------------------
-----------------------------------------------------------------------+--------
+-------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
---------------------+----
------+-----------
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   16387 | name:
proc_assign(), fn_collation: 0, search_path: addUser: false, schemas: 2200
|      3 | SELECT a + 1
                                                                                                                    |
12160 |     18432
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   16387 | name:
proc_assign(), fn_collation: 0, search_path: addUser: false, schemas: 2200
|      4 | SELECT a
                                                                                                                    |
10152 |     14336
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   16387 | name:
proc_assign(), fn_collation: 0, search_path: addUser: false, schemas: 2200
|      1 | SELECT 0
                                                                                                                    |
 9832 |     14336
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   13510 | name:
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |      12 | SELECT COALESCE(sessionid::text, 'NULL')
                                                                                                                    |
13992 |     22528
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   13510 | name:
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |      13 | SELECT COALESCE(pkg_id::text, 'NULL')
                                                                                                                    |
13480 |     22528
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   13510 | name:
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |      14 | SELECT COALESCE(func_id::text, 'NULL')
                                                                                                                    |
13480 |     22528
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0            |      14023 |      0 |   13510 | name:
```

```
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |     15 | SELECT 'SELECT node_name FROM pg_catalog.pgxc_node WHERE node_type
in (''C'', ''D'') AND nodeis_active = true'
|
10528 |     14336
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0           |     14023 |     0 |   13510 | name:
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |     17 | SELECT 'EXECUTE DIRECT ON (' || pg_catalog.quote_ident(row_name
.node_name)|| ')''SELECT * from pg_catalog.gs_plsql_statements_details('|| sessionid_str ||', '|| pkg_id_str
||', '|| func_id_str || ');''' |
31456 |     47104
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0           |     14023 |     0 |   13510 | name:
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |     31 | SELECT (fet_active)
                                                                                                                |
10248 |     14336
 coordinator1 | 1171777280 | 139767997523712 | 0:0#0           |     14023 |     0 |   13510 | name:
gs_global_plsql_statements_details(bigint,oid,oid), fn_collation: 0, search_path: addUser: false,
schemas: 2200 |     31 | SELECT (query_str_nodes)
                                                                                                                |
10248 |     14336
(10 rows)

gaussdb=# DROP PROCEDURE proc_assign;
```

分区表统计信息函数

- gs_stat_get_partition_stats(oid)
  描述：获取特定分区的统计信息。
  返回值类型：record
- gs_stat_get_xact_partition_stats(oid)
  描述：获取特定分区的事务中统计信息。
  返回值类型：record
- gs_stat_get_all_partitions_stats()
  描述：获取所有分区的统计信息。
  返回值类型：setof record
- gs_stat_get_xact_all_partitions_stats()
  描述：获取所有分区的事务中统计信息。
  返回值类型：setof record
- gs_statio_get_all_partitions_stats()
  描述：获取所有分区的I/O统计信息。
  返回值类型：setof record
  上述五个函数示例：

⚠ 注意

运行时统计信息上报是异步的，且基于UDP协议，后台线程处理可能存在延迟和丢包，此处示例预期仅供参考。

事务外统计信息查询：

```
gaussdb=# CREATE TABLE part_tab1
gaussdb-# (
gaussdb(#     a int, b int
```


版权所有 © 华为云计算技术有限公司

```
gaussdb(# )
gaussdb-# PARTITION BY RANGE(b)
gaussdb-# (
gaussdb(#     PARTITION P1 VALUES LESS THAN(10),
gaussdb(#     PARTITION P2 VALUES LESS THAN(20),
gaussdb(#     PARTITION P3 VALUES LESS THAN(MAXVALUE)
gaussdb(# );
CREATE TABLE
gaussdb=# CREATE TABLE subpart_tab1
gaussdb-# (
gaussdb(#     month_code VARCHAR2 ( 30 ) NOT NULL ,
gaussdb(#     dept_code  VARCHAR2 ( 30 ) NOT NULL ,
gaussdb(#     user_no    VARCHAR2 ( 30 ) NOT NULL ,
gaussdb(#     sales_amt  int
gaussdb(# )
gaussdb-# PARTITION BY RANGE (month_code) SUBPARTITION BY RANGE (dept_code)
gaussdb-# (
gaussdb(#   PARTITION p_201901 VALUES LESS THAN( '201903' )
gaussdb(#   (
gaussdb(#     SUBPARTITION p_201901_a VALUES LESS THAN( '2' ),
gaussdb(#     SUBPARTITION p_201901_b VALUES LESS THAN( '3' )
gaussdb(#   ),
gaussdb(#   PARTITION p_201902 VALUES LESS THAN( '201904' )
gaussdb(#   (
gaussdb(#     SUBPARTITION p_201902_a VALUES LESS THAN( '2' ),
gaussdb(#     SUBPARTITION p_201902_b VALUES LESS THAN( '3' )
gaussdb(#   )
gaussdb(# );
CREATE TABLE
gaussdb=# CREATE INDEX index_part_tab1 ON part_tab1(b) LOCAL
gaussdb-# (
gaussdb(# PARTITION b_index1,
gaussdb(# PARTITION b_index2,
gaussdb(# PARTITION b_index3
gaussdb(# );
CREATE INDEX
gaussdb=# CREATE INDEX idx_user_no ON subpart_tab1(user_no) LOCAL;
CREATE INDEX
gaussdb=# INSERT INTO part_tab1 VALUES(1, 1);
INSERT 0 1
gaussdb=# INSERT INTO part_tab1 VALUES(1, 11);
INSERT 0 1
gaussdb=# INSERT INTO part_tab1 VALUES(1, 21);
INSERT 0 1
gaussdb=# UPDATE part_tab1 SET a = 2 WHERE b = 1;
UPDATE 1
gaussdb=# UPDATE part_tab1 SET a = 3 WHERE b = 11;
UPDATE 1
gaussdb=# UPDATE /*+ indexscan(part_tab1) */ part_tab1 SET a = 4 WHERE b = 21;
UPDATE 1
gaussdb=# DELETE FROM part_tab1;
DELETE 3
gaussdb=# ANALYZE part_tab1;
ANALYZE
gaussdb=# VACUUM part_tab1;
VACUUM
gaussdb=# INSERT INTO subpart_tab1 VALUES('201902', '1', '1', 1);
INSERT 0 1
gaussdb=# INSERT INTO subpart_tab1 VALUES('201902', '2', '2', 1);
INSERT 0 1
gaussdb=# INSERT INTO subpart_tab1 VALUES('201903', '1', '3', 1);
INSERT 0 1
gaussdb=# INSERT INTO subpart_tab1 VALUES('201903', '2', '4', 1);
INSERT 0 1
gaussdb=# UPDATE subpart_tab1 SET sales_amt = 2 WHERE user_no='1';
UPDATE 1
gaussdb=# UPDATE subpart_tab1 SET sales_amt = 3 WHERE user_no='2';
UPDATE 1
gaussdb=# UPDATE subpart_tab1 SET sales_amt = 4 WHERE user_no='3';
```


版权所有 © 华为云计算技术有限公司

```
UPDATE 1
gaussdb=# UPDATE /*+ indexscan(subpart_tab1) */ subpart_tab1 SET sales_amt = 5 WHERE
user_no='4';
UPDATE 1
gaussdb=# DELETE FROM subpart_tab1;
DELETE 4
gaussdb=# ANALYZE subpart_tab1;
ANALYZE
gaussdb=# VACUUM subpart_tab1;
VACUUM
gaussdb=# SELECT * FROM gs_stat_all_partitions;
 partition_oid | schemaname |    relname    | partition_name | sub_partition_name | seq_scan |
seq_tup_read | idx_scan | idx_tup_fetch | n_tup_ins | n_tup_upd | n_tup_del | n_tup_hot_upd | n_live_tup
|
 n_dead_tup |          last_vacuum          |     last_autovacuum      |         last_analyze          |
last_autoanalyze     | vacuum_count | autovacuum_count | analyze_count | autoanalyze_count
---------------+------------+--------------+----------------+--------------------+----------+--------------
+----------+---------------+-----------+-----------+-----------+---------------+------------+
------------+-------------------------------+-----------------------+-------------------------------
+-----------------------+--------------+------------------+---------------+-------------------
        16964 | public     | subpart_tab1 | p_201902       | p_201902_b         |        5 |           1 |        4
|            1 |         1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:45.293965+08 | 2000-01-01 08:00:00+08 | 2023-05-15
20:36:44.688861+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
        16963 | public     | subpart_tab1 | p_201902       | p_201902_a         |        5 |           1 |        4
|            0 |         1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:45.291022+08 | 2000-01-01 08:00:00+08 | 2023-05-15
20:36:44.688843+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
        16961 | public     | subpart_tab1 | p_201901       | p_201901_b         |        5 |           1 |        4
|            0 |         1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:45.288037+08 | 2000-01-01 08:00:00+08 | 2023-05-15
20:36:44.688829+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
        16960 | public     | subpart_tab1 | p_201901       | p_201901_a         |        5 |           1 |        4
|            0 |         1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:45.285311+08 | 2000-01-01 08:00:00+08 | 2023-05-15
20:36:44.688802+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
        16954 | public     | part_tab1    | p3             |                    |        2 |           1 |        1 |           1
|            1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:29.490636+08 | 2000-01-01 08:00:00+08 | 2023-05-15
20:36:28.540115+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
        16953 | public     | part_tab1    | p2             |                    |        4 |           1 |        1 |           0
|            1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:29.487914+08 | 2000-01-01 08:00:00+08 | 2023-05-15
20:36:28.540098+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
        16952 | public     | part_tab1    | p1             |                    |        5 |           1 |        1 |           0
|            1 |         1 |         1 |             1 |          0 |
         1 | 2023-05-15 20:36:29.48536+08  | 2000-01-01 08:00:00+08 | 2023-05-15 20:36:28.540071+08
| 2000-01-01 08:00:00+08 |            1 |               0 |            1 |               0
(7 rows)

gaussdb=# SELECT * FROM gs_statio_all_partitions;
 partition_oid | schemaname |    relname    | partition_name | sub_partition_name | heap_blks_read |
heap_blks_hit | idx_blks_read | idx_blks_hit | toast_blks_read | toast_blks_hit | tidx_blks_read | t
idx_blks_hit
---------------+------------+--------------+----------------+--------------------+----------------+--------------
+---------------+--------------+-----------------+----------------+----------------+--
-------------
        16964 | public     | subpart_tab1 | p_201902       | p_201902_b         |              4 |            8
|             2 |           21 |                 |                |                |

        16963 | public     | subpart_tab1 | p_201902       | p_201902_a         |              4 |            8
|             2 |           21 |                 |                |                |

        16961 | public     | subpart_tab1 | p_201901       | p_201901_b         |              4 |            8
|             2 |           21 |                 |                |                |

        16960 | public     | subpart_tab1 | p_201901       | p_201901_a         |              4 |            8
|             2 |           21 |                 |                |                |
```


版权所有 © 华为云计算技术有限公司

```
        16954 | public     | part_tab1    | p3           |                 |            4 |            8 |            2
|          15 |              |              |             |

        16953 | public     | part_tab1    | p2           |                 |            4 |            8 |            2
|          15 |              |              |             |

        16952 | public     | part_tab1    | p1           |                 |            4 |            8 |            2
|          15 |              |              |             |

(7 rows)

gaussdb=# SELECT * FROM gs_stat_get_partition_stats(16952);
 partition_oid | seq_scan | seq_tup_read | idx_scan | idx_tup_fetch | n_tup_ins | n_tup_upd | n_tup_del |
n_tup_hot_upd | n_live_tup | n_dead_tup |          last_vacuum          |     last_autovacuum
  |         last_analyze          |     last_autoanalyze     | vacuum_count | autovacuum_count |
analyze_count | autoanalyze_count |    last_data_changed     | heap_blks_read | heap_blks_hit |
idx_blks_re
ad | idx_blks_hit | tup_fetch | block_fetch
---------------+----------+--------------+----------+---------------+-----------+-----------+-----------
+---------------+------------+------------+-----------------------------+---------------------
--+-------------------------------+------------------------+--------------+------------------+---------------
+-------------------+------------------------+----------------+---------------+------------
---+--------------+-----------+-------------
        16952 |        5 |            1 |        1 |            0 |         1 |         1 |         1 |            1 |         0
|          1 | 2023-05-15 20:36:29.48536+08 | 2000-01-01 08:00:00+0
8 | 2023-05-15 20:36:28.540071+08 | 2000-01-01 08:00:00+08 |            1 |               0 |            1
|                0 | 2000-01-01 08:00:00+08 |              4 |             8 |
 2 |          21 |         0 |          12
(1 row)
```

事务内统计信息查询：

```
gaussdb=# BEGIN;
BEGIN
gaussdb=# INSERT INTO part_tab1 VALUES(1, 1);
INSERT 0 1
gaussdb=# INSERT INTO part_tab1 VALUES(1, 11);
INSERT 0 1
gaussdb=# INSERT INTO part_tab1 VALUES(1, 21);
INSERT 0 1
gaussdb=# UPDATE part_tab1 SET a = 2 WHERE b = 1;
UPDATE 1
gaussdb=# UPDATE part_tab1 SET a = 3 WHERE b = 11;
UPDATE 1
gaussdb=# UPDATE /*+ indexscan(part_tab1) */ part_tab1 SET a = 4 WHERE b = 21;
UPDATE 1
gaussdb=# DELETE FROM part_tab1;
DELETE 3
gaussdb=# INSERT INTO subpart_tab1 VALUES('201902', '1', '1', 1);
INSERT 0 1
gaussdb=# INSERT INTO subpart_tab1 VALUES('201902', '2', '2', 1);
INSERT 0 1
gaussdb=# INSERT INTO subpart_tab1 VALUES('201903', '1', '3', 1);
INSERT 0 1
gaussdb=# INSERT INTO subpart_tab1 VALUES('201903', '2', '4', 1);
INSERT 0 1
gaussdb=# UPDATE subpart_tab1 SET sales_amt = 2 WHERE user_no='1';
UPDATE 1
gaussdb=# UPDATE subpart_tab1 SET sales_amt = 3 WHERE user_no='2';
UPDATE 1
gaussdb=# UPDATE subpart_tab1 SET sales_amt = 4 WHERE user_no='3';
UPDATE 1
gaussdb=# UPDATE /*+ indexscan(subpart_tab1) */ subpart_tab1 SET sales_amt = 5 WHERE
user_no='4';
UPDATE 1
gaussdb=# DELETE FROM subpart_tab1;
DELETE 4
gaussdb=# SELECT * FROM gs_stat_xact_all_partitions;
 partition_oid | schemaname |    relname     | partition_name | sub_partition_name | seq_scan |
seq_tup_read | idx_scan | idx_tup_fetch | n_tup_ins | n_tup_upd | n_tup_del | n_tup_hot_upd
---------------+------------+--------------+----------------+--------------------+----------+--------------
```


版权所有 © 华为云计算技术有限公司

```
+----------+---------------+-----------+-----------+-----------+---------------
       16964 | public    | subpart_tab1 | p_201902      | p_201902_b        |      4 |          4 |       1
|           2 |         1 |         1 |         1 |           1
       16963 | public    | subpart_tab1 | p_201902      | p_201902_a        |      4 |          4 |       1
|           0 |         1 |         1 |         1 |           1
       16961 | public    | subpart_tab1 | p_201901      | p_201901_b        |      4 |          4 |       1
|           0 |         1 |         1 |         1 |           1
       16960 | public    | subpart_tab1 | p_201901      | p_201901_a        |      4 |          4 |       1
|           0 |         1 |         1 |         1 |           1
       16954 | public    | part_tab1    | p3            |                   |      1 |          1 |       1 |           2
|         1 |         1 |         1 |           1
       16953 | public    | part_tab1    | p2            |                   |      3 |          2 |       0 |           0
|         1 |         1 |         1 |           1
       16952 | public    | part_tab1    | p1            |                   |      4 |          2 |       0 |           0
|         1 |         1 |         1 |           1
(7 rows)

gaussdb=# SELECT * FROM gs_stat_get_xact_partition_stats(16952);
 partition_oid | seq_scan | seq_tup_read | idx_scan | idx_tup_fetch | n_tup_ins | n_tup_upd | n_tup_del |
n_tup_hot_upd | tup_fetch
---------------+----------+--------------+----------+---------------+-----------+-----------+-----------
+---------------+-----------
       16952 |        4 |           2 |        0 |             0 |         1 |         1 |         1 |             1 |         0
(1 row)

gaussdb=# DROP TABLE part_tab1;
gaussdb=# DROP TABLE subpart_tab1;
```

- gs_stat_get_partition_analyze_count(oid)

  描述：用户在该分区上启动分析的次数。

  返回值类型：bigint

- gs_stat_get_partition_autoanalyze_count(oid)

  描述：autovacuum守护线程在该分区上启动分析的次数。

  返回值类型：bigint

- gs_stat_get_partition_autovacuum_count(oid)

  描述：autovacuum守护线程在该分区上启动清理的次数。

  返回值类型：bigint

- gs_stat_get_partition_last_analyze_time(oid)

  描述：用户在该分区上最后一次手动启动分析或者autovacuum线程启动分析的时间。

  返回值类型：timestamptz

- gs_stat_get_partition_last_autoanalyze_time(oid)

  描述：autovacuum守护线程在该分区上最后一次启动分析的时间。

  返回值类型：timestamptz

- gs_stat_get_partition_last_autovacuum_time(oid)

  描述：autovacuum守护线程在该分区上最后一次启动清理的时间。

  返回值类型：timestamptz

- gs_stat_get_partition_last_data_changed_time(oid)

  描述：对于在分区上的修改insert/update/delete/truncate，在该表上最后一次操作的时间。当前暂不支持。

  返回值类型：timestamptz

- gs_stat_get_partition_last_vacuum_time(oid)

版权所有 © 华为云计算技术有限公司

描述：用户在该分区上最后一次手动启动清理或者autovacuum线程启动清理的时间。

返回值类型：timestamptz

- gs_stat_get_partition_numscans(oid)
  描述：分区顺序扫描读取的行数目。
  返回值类型：bigint
- gs_stat_get_partition_tuples_returned(oid)
  描述：分区顺序扫描读取的行数目。
  返回值类型：bigint
- gs_stat_get_partition_tuples_fetched(oid)
  描述：分区位图扫描抓取的行数目。
  返回值类型：bigint
- gs_stat_get_partition_vacuum_count(oid)
  描述：用户在该分区上启动清理的次数。
  返回值类型：bigint
- gs_stat_get_xact_partition_tuples_fetched(oid)
  描述：事务中扫描的tuple行数。
  返回值类型：bigint
- gs_stat_get_xact_partition_numscans(oid)
  描述：当前事务中分区执行的顺序扫描次数。
  返回值类型：bigint
- gs_stat_get_xact_partition_tuples_returned(oid)
  描述：当前事务中分区通过顺序扫描读取的行数。
  返回值类型：bigint
- gs_stat_get_partition_blocks_fetched(oid)
  描述：分区的磁盘块抓取请求的数量。
  返回值类型：bigint
- gs_stat_get_partition_blocks_hit(oid)
  描述：在缓冲区中找到的分区的磁盘块请求数目。
  返回值类型：bigint
- pg_stat_get_partition_tuples_inserted(oid)
  描述：已插入相应表分区中的行数。
  返回值类型：bigint
- pg_stat_get_partition_tuples_updated(oid)
  描述：在相应表分区中已更新行的数量。
  返回值类型：bigint
- pg_stat_get_partition_tuples_deleted(oid)
  描述：从相应表分区中删除行的数量。
  返回值类型：bigint
- pg_stat_get_partition_tuples_changed(oid)

版权所有 © 华为云计算技术有限公司

描述：该表分区上一次analyze或autoanalyze之后插入、更新、删除行的总数量。

返回值类型：bigint

- pg_stat_get_partition_live_tuples(oid)

  描述：分区表活行数。

  返回值类型：bigint

- pg_stat_get_partition_dead_tuples(oid)

  描述：分区表死行数。在Ustore表中仅代表不活跃行指针数量。

  返回值类型：bigint

- pg_stat_get_xact_partition_tuples_inserted(oid)

  描述：表分区相关的活跃子事务中插入的tuple数。

  返回值类型：bigint

- pg_stat_get_xact_partition_tuples_deleted(oid)

  描述：表分区相关的活跃子事务中删除的tuple数。

  返回值类型：bigint

- pg_stat_get_xact_partition_tuples_hot_updated(oid)

  描述：表分区相关的活跃子事务中热更新的tuple数。

  返回值类型：bigint

- pg_stat_get_xact_partition_tuples_updated(oid)

  描述：表分区相关的活跃子事务中更新的tuple数。

  返回值类型：bigint

- pg_stat_get_partition_tuples_hot_updated(oid)

  描述：返回给定分区id的分区热更新元组数的统计。

  参数：oid

  返回值类型：bigint

- gs_stat_partition_analyze_vacuum(oid)

  描述：缓存加速获取特定分区的vacuum/analyze相关统计信息。

  参数：partoid

  返回值类型：record

- gs_stat_get_parttabentries()

  描述：使用该函数可获取其他节点的分区表的vacuum/analyze相关统计信息。内部函数，不建议用户直接使用。

  返回值类型：record

- gs_wlm_respool_cpu_info()

  描述：描述资源池的CPU资源限制和使用信息。

  返回值类型：setof record

名称	类型	描述
respool_name	name	资源池名称。
control_group	name	cgroup名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cpu_affinity	name	CPU绑定core的数值。
cpu_usage	integer	资源池的CPU使用率。如果资源池没有进行CPU管控，则该项显示为0。

说明

对于CN和DN混合部署的场景，CN和DN共享同一份CPU资源，因此CN和DN的cpu_usage显示相同，对于CN和DN独立部署场景，会独立显示。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_RESPOOL_CPU_INFO();
 respool_name |     control_group      | cpu_affinity | cpu_usage
---------------+------------------------+--------------+-----------
 respool_cpu_2 | respool_cpu_2:Medium | 0-95         |         78
 default_pool  | DefaultClass:Medium  | 0-32         |         65
(2 rows)
```

- gs_wlm_respool_connection_info()

  描述：描述资源池的连接数资源限制和使用信息。

  返回值类型：setof record

名称	类型	描述
respool_name	name	资源池名称。
max_connections	name	资源池最大连接数。
curr_connections	integer	当前资源池已有连接数。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_RESPOOL_CONNECTION_INFO();
 respool_name  | max_connections | curr_connections
---------------+-----------------+------------------
 respool1      |              -1 |                0
 default_pool  |              -1 |                1
(2 rows)
```

- gs_wlm_respool_memory_info()

  描述：描述资源池的内存资源限制和使用信息。

  返回值类型：setof record

名称	类型	描述
respool_name	name	资源池名称。
max_dynamic_memory	integer	最大可使用的动态内存。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
current_dynamic_memory	integer	当前已经使用的动态内存。
max_shared_memory	integer	最大可使用的共享内存。
current_shared_memory	integer	当前已经使用的共享内存。
shared_memory_hits_percent	integer	该功能不可用，留待后续演进。字段保留，恒为0。

说明

在出现资源池动态内存使用超过最大值时，可能会出现查询GUC参数current_dynamic_memory返回的结果大于max_dynamic_memory的情况，这属于正常现象，实际并未申请内存。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_RESPOOL_MEMORY_INFO();
 respool_name   | max_dynamic_memory | current_dynamic_memory | max_shared_memory |
current_shared_memory | shared_memory_hits_percent
----------------+--------------------+------------------------+-------------------+-------------------------
+----------------------------
 default_pool   | -1                 | 3383kB                 | -1                | 3848kB                  |
0
 resource_pool_a | 30720kB           | 0kB                    | -1                | 0kB
|                         0
(2 rows)
```

- gs_wlm_respool_concurrency_info()

  描述：描述资源池中的并发数资源限制和使用信息。

  返回值类型：setof record

名称	类型	描述
respool_name	name	资源池名称。
max_concurrency	integer	资源池支持的最大并发数。
running_concurrency	integer	当前资源池正在执行的并发数。
waiting_concurrency	integer	当前资源池正在等待的并发数。如果资源池没有进行并发限制，则该项显示为0。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_RESPOOL_CONCURRENCY_INFO();
 respool_name   | max_concurrency | running_concurrency | waiting_concurrency
----------------+-----------------+---------------------+---------------------
 default_pool   |              -1 |                   1 |                   0
```


版权所有 © 华为云计算技术有限公司

```
 resource_pool_a |           -1 |                0 |                0
(2 rows)
```

- gs_wlm_respool_io_info()

  描述：描述资源池的I/O资源限制和使用信息。

  返回值类型：setof record

名称	类型	描述
respool_name	name	资源池名称。
io_limits	integer	每秒触发I/O的次数上限（单位：IOPS）。0表示不控制。
io_priority	text	I/O 资源管控的优先级等级。当I/O利用率高达90%时，作业的I/O操作按此优先级进行管控。None表示不控制。
current_iops	integer	当前I/O已经触发的次数（单位：IOPS）。 • 因该值与I/O统计算法有关，存在偶尔超过上限的情况，属于正常波动。 • 如果资源池没有进行I/O管控，则该项显示为0。

  说明

  在I/O管控过程中，I/O操作的计数单位由GUC参数io_control_unit定义。1个io_control_unit次为1次IOPS。

  示例：

```
gaussdb=# SELECT * FROM GS_WLM_RESPOOL_IO_INFO();
 respool_name   | io_limits | io_priority | current_iops
----------------+-----------+-------------+--------------
 default_pool   |         0 | None        |            0
 resource_pool_a |         0 | Low         |            0
(2 rows)
```

- gs_wlm_user_space_info()

  描述：描述用户的存储空间的使用信息。

  返回值类型：setof record

名称	类型	描述
user_name	name	用户名称。
max_permanent_space	bigint	用户可使用的最大永久存储空间，单位：B。
current_permanent_space	bigint	当前用户已使用的永久存储空间，单位：B。
max_temp_space	bigint	用户可使用的最大临时存储空间，单位：B。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
current_temp_space	bigint	当前用户已使用的临时存储空间，单位：B。
max_spill_space	bigint	用户可使用的最大算子落盘存储空间，单位：B。
current_spill_space	bigint	当前用户已使用的算子落盘存储空间，单位：B。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_USER_SPACE_INFO();
          user_name          | max_permanent_space | current_permanent_space | max_temp_space |
current_temp_space | max_spill_space | current_spill_space
-----------------------------+---------------------+-------------------------+----------------+--------------------
+-----------------+--------------------
 xy                          |                  -1 |                    2464 |             -1 |                  0 |              -1
|                 0
(1 rows)
```

- gs_wlm_session_io_info()

  描述：描述session的I/O的使用信息。

  返回值类型：setof record

名称	类型	描述。
session_id	integer	会话ID。
io_limits	integer	每秒触发I/O的次数上限（单位：IOPS）。0表示不控制。
io_priority	text	I/O 资源管控的优先级等级。当I/O利用率高达90%时，作业的I/O操作按此优先级进行管控。None表示不控制。
current_iops	integer	当前I/O已经触发的次数（单位：IOPS）。当前I/O统计值存在偶尔超过上限的情况，与I/O统计算法有关，属于正常波动。
wait_time	integer	当前I/O超过上限后等待的时间总和。

说明

在I/O管控过程中，I/O操作的计数单位由GUC参数io_control_unit定义。1个io_control_unit次为1次IOPS。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_SESSION_IO_INFO();
   session_id    | io_limits | io_priority | current_iops | wait_time
-----------------+-----------+-------------+--------------+-----------
 139976325986048 |        10 | None        |            0 |      2709
(1 row)
```

- gs_wlm_session_memory_info()

  描述：描述session的内存的使用信息。

版权所有 © 华为云计算技术有限公司

返回值类型：setof record

名称	类型	描述
session_id	integer	session_id。
sess_used_dynamic_memory	bigint	当前已经使用的动态内存，单位为B。
sess_max_dynamic_memory	bigint	最大可使用的动态内存，单位为B。

示例：

```
gaussdb=# SELECT * FROM GS_WLM_SESSION_MEMORY_INFO();
    sessid      | sess_used_dynamic_memory | sess_max_dynamic_memory
----------------+--------------------------+-------------------------
 139976325986048 |                  4326056 |                      -1
 139976402532096 |                  4452664 |                      -1
(2 rows)
```

- gs_emergency_operation_records()

  描述：展示当前节点对Unix Domain Socket运维通道的使用情况。

  返回值类型：setof record

名称	类型	描述
record_time	timestamptz	使用运维通道的时间。
action	text	使用运维通道执行的操作。
source	text	接入运维通道的来源，取值为gsql和cm。
detail	text	执行操作的具体结果。

- gs_memory_profiling(option integer)

  描述：生成GaussDB当前进程的jeprof报告。option表示是否开启内存采集功能，并根据实际输入返回结果为字符串类型的说明。

  参数：option，整型，操作选项。

  - 0：表示关闭内存采集，函数返回“memory profiling closed”说明。
  - 1：表示开启内存采集，函数返回“memory profiling opened”说明。
  - 2：表示在开启内存采集的情况下进行内存调用收集，并生成jeprof报告，报告位于mem_log目录下，报告的命名格式为jeprof-*YY-mm-dd_HHMMSS*，函数返回“dump succeeded”说明。
  - 其他：返回提示信息“Invalid input, please input 0(disable), 1(enable) or 2(dump).”。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

说明

- 仅具有SYSADMIN权限的用户可以使用该函数。
- 开启内存采集会影响性能，因此仅推荐在尝试分析内存问题（例如内存泄漏）的非性能敏感场景下开启功能。生成的jeprof报告可以通过如下命令生成PDF报告：
  jeprof --show_bytes *gaussdbbin路径* -pdf *生成的jeprof报告路径* > *期待输出文件*.pdf

- gs_jemalloc_info_details()

  描述：统计jemalloc的关键字段并生成jemalloc stat报告。stat报告位于mem_log目录下，命名为memory_stat_report-*YY-mm-dd_HHMMSS*.log。

  返回值类型：setof record

  表 1-255 gs_jemalloc_info_details 字段

名称	类型	描述
nodename	text	节点名称
memorytype	text	字段名称
memorymbytes	integer	对应字段下虚拟内存的大小（单位：MB）

说明

仅具有SYSADMIN权限的用户可以使用该函数。

1.6.30 触发器函数

- pg_get_triggerdef(oid)

  描述：获取触发器的定义信息。

  参数：待查触发器的OID。

  返回值类型：text

  示例:

```
-- 创建表tri_insert。
gaussdb=# CREATE TABLE tri_insert (a int, b int);
CREATE TABLE
-- 创建函数trigger_func。
gaussdb=# CREATE FUNCTION trigger_func() RETURNS trigger LANGUAGE plpgsql AS '
 BEGIN
 RAISE NOTICE ''trigger_func(%) called: action = %, when = %, level = %'', TG_ARGV[0], TG_OP,
TG_WHEN, TG_LEVEL;
 RETURN NULL;
 END;';
CREATE FUNCTION
-- 创建触发器before_ins_stmt_trig。
gaussdb=# CREATE TRIGGER before_ins_stmt_trig BEFORE INSERT ON tri_insert FOR EACH row
EXECUTE PROCEDURE trigger_func('before_ins_stmt');
CREATE TRIGGER
-- 创建触发器after_ins_when_trig。
gaussdb=# CREATE TRIGGER after_ins_when_trig AFTER INSERT ON tri_insert FOR EACH ROW WHEN
(new.a IS NOT NULL) EXECUTE PROCEDURE trigger_func('after_ins_when');
CREATE TRIGGER
-- 查看表tri_insert的触发器定义信息。
gaussdb=# SELECT pg_get_triggerdef(oid) FROM pg_trigger WHERE tgrelid = 'tri_insert'::regclass;
```


版权所有 © 华为云计算技术有限公司

```
pg_get_triggerdef
-------------------------------------------------------------------------------------------------------------------------------------------------------------
 CREATE TRIGGER after_ins_when_trig AFTER INSERT ON tri_insert FOR EACH ROW WHEN ((new.a IS NOT NULL)) EXECUTE PROCEDURE trigger_func('after_ins_when')
 CREATE TRIGGER before_ins_stmt_trig BEFORE INSERT ON tri_insert FOR EACH ROW EXECUTE PROCEDURE trigger_func('before_ins_stmt')
(2 rows)
```

- pg_get_triggerdef(oid, boolean)

  描述：获取触发器的定义信息。

  参数：待查触发器的OID及是否以pretty方式展示。

  说明

  仅在创建trigger时指定WHEN条件的情况下，布尔类型参数才生效。

  返回值类型：text

  示例:

```
-- 查看表tri_insert的触发器定义信息，以非pretty形式。
gaussdb=# SELECT pg_get_triggerdef(oid, false) FROM pg_trigger WHERE tgrelid = 'tri_insert'::regclass;

pg_get_triggerdef
-------------------------------------------------------------------------------------------------------------------------------------------------------------
 CREATE TRIGGER after_ins_when_trig AFTER INSERT ON tri_insert FOR EACH ROW WHEN ((new.a IS NOT NULL)) EXECUTE PROCEDURE trigger_func('after_ins_when')
 CREATE TRIGGER before_ins_stmt_trig BEFORE INSERT ON tri_insert FOR EACH ROW EXECUTE PROCEDURE trigger_func('before_ins_stmt')
(2 rows)

-- 查看表tri_insert的触发器定义信息，以pretty形式。
gaussdb=# SELECT pg_get_triggerdef(oid, true) FROM pg_trigger WHERE tgrelid = 'tri_insert'::regclass;

pg_get_triggerdef
-------------------------------------------------------------------------------------------------------------------------------------------------------------
 CREATE TRIGGER after_ins_when_trig AFTER INSERT ON tri_insert FOR EACH ROW WHEN (new.a IS NOT NULL) EXECUTE PROCEDURE trigger_func('after_ins_when')
 CREATE TRIGGER before_ins_stmt_trig BEFORE INSERT ON tri_insert FOR EACH ROW EXECUTE PROCEDURE trigger_func('before_ins_stmt')
(2 rows)

-- 清理表tri_insert。
gaussdb=# DROP TABLE tri_insert CASCADE;
DROP TABLE
-- 清理函数trigger_func。
gaussdb=# DROP FUNCTION trigger_func;
DROP FUNCTION
```

- suppress_redundant_updates_trigger()

  描述：用于抑制无数据变化的更新操作。在更新数据操作前调用该函数，若更新后的数据相对更新前没有产生变化，则该更新操作不生效。

  返回值类型：触发器

  说明

  该函数为内置的触发器执行函数，只能在触发器定义时调用，且仅在BEFORE UPDATE触发器中生效。

  示例：

```
-- 创建表tri_update。
gaussdb=# CREATE TABLE tri_update (a int, b int);
CREATE TABLE
-- 插入数据(1,1)。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO tri_update VALUES(1,1);
INSERT 0 1
-- 创建BEFORE UPDATE触发器，触发器函数为suppress_redundant_updates_trigger。
gaussdb=# CREATE TRIGGER before_update_trig BEFORE UPDATE ON tri_update FOR EACH ROW
EXECUTE PROCEDURE suppress_redundant_updates_trigger();
WARNING:  Trigger function with non-plpgsql type is not recommended.
DETAIL:  Non-plpgsql trigger function are not shippable by default.
HINT:  Unshippable trigger may lead to bad performance.
CREATE TRIGGER
-- 更新表中数据不变，依旧为(1,1),不执行更新操作。
gaussdb=# UPDATE tri_update SET b=1;
UPDATE 0
-- 清理表tri_update。
gaussdb=# DROP TABLE tri_update CASCADE;
DROP TABLE
```

1.6.31 HashFunc 函数

- bucketabstime(value，flag)

  描述：对abstime格式的数值value计算hash值并找到对应的hashbucket桶。

  参数：value为需要转换的数值，类型为abstime，flag为int类型表示数据分布方式，0表示hash分布。

  返回值类型：int32

  示例：

  ```
  gaussdb=# SELECT bucketabstime('2011-10-01 10:10:10.112',1);
   bucketabstime
  ---------------
          13954
  (1 row)
  ```

- bucketbool(value，flag)

  描述：对boolean格式的数值value计算hash值并找到对应的hashbucket桶。

  参数：value为需要转换的数值，类型为boolean，flag为int类型表示数据分布方式，0表示hash分布。

  返回值类型：int32

  示例：

  ```
  gaussdb=# SELECT bucketbool(true,1);
   bucketbool
  ------------
           1
  (1 row)
  gaussdb=# SELECT bucketbool(false,1);
   bucketbool
  ------------
           0
  (1 row)
  ```

- bucketbpchar(value, flag)

  描述：对bpchar格式的数值value计算hash值并找到对应的hashbucket桶。

  参数：value为需要转换的数值，类型为bpchar，flag为int类型表示数据分布方式，0表示hash分布。

  返回值类型：int32

  示例：

  ```
  gaussdb=# SELECT bucketbpchar('test',1);
   bucketbpchar
  --------------
          9761
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- bucketbytea(value，flag)

  描述：对bytea格式的数值value计算hash值并找到对应的hashbucket桶。

  参数：value为需要转换的数值，类型为bytea，flag为int类型表示数据分布方式，0表示hash分布。

  返回值类型：int32

  示例：

```
gaussdb=# SELECT bucketbytea('test',1);
 bucketbytea
-------------
        9761
(1 row)
```

- bucketcash(value，flag)

  描述：对money格式的数值value计算hash值并找到对应的hashbucket桶。

  参数：value为需要转换的数值，类型为money，flag为int类型表示数据分布方式，0表示hash分布。

  返回值类型：int32

  示例：

```
gaussdb=# SELECT bucketcash(10::money,1);
 bucketcash
------------
       8468
(1 row)
```

- getbucket(value，flag)

  描述：从分布列获取hashbucket桶。

  value为需要输入的数值，类型：

  “char”,abstime,bigint,boolean,bytea,character varying,character,date,double precision,int2vector,integer,interval,money,name,numeric,nvarchar2,oid,oidvector,raw,real,record,reltime,smalldatetime,smallint,text,time with time zone,time without time zone,timestamp with time zone,timestamp without time zone,tinyint,uuid。

  flag表示数据分布方式，类型：integer

  返回值类型：integer

  示例：

```
gaussdb=# SELECT getbucket(10,'H');
 getbucket
-----------
     14535
(1 row)

gaussdb=# SELECT getbucket(11,'H');
 getbucket
-----------
     13449
(1 row)

gaussdb=# SELECT getbucket(11,'R');
 getbucket
-----------
     13449
(1 row)

gaussdb=# SELECT getbucket(12,'R');
 getbucket
-----------
```


版权所有 © 华为云计算技术有限公司

```
    9412
(1 row)
```

- ora_hash(expression,[seed])

  描述：用于计算给定表达式的哈希值。expression:可输入的类型覆盖字符串，时间类型，数字类型，ROWID类型，根据expression进行计算哈希值。seed:可选参数，一个int8值，可以对同一个输入值返回不同的结果, 用于计算带随机数的hash值。

  返回类型：int8类型的哈希值。

  示例：

```
gaussdb=# SELECT ora_hash(123);
 ora_hash
------------
 4089882933
(1 row)
gaussdb=# SELECT ora_hash('123');
 ora_hash
------------
 2034089965
(1 row)
gaussdb=# SELECT ora_hash('sample');
 ora_hash
------------
 1573005290
(1 row)
gaussdb=# SELECT ora_hash(to_date('2012-1-2','yyyy-mm-dd'));
 ora_hash
------------
 1171473495
(1 row)
gaussdb=# SELECT ora_hash(123,234);
 ora_hash
------------
 -9089505052966355682
(1 row)
gaussdb=# SELECT ora_hash('123',234);
 ora_hash
------------
 5742589019960764616
(1 row)
gaussdb=# SELECT ora_hash('sample',234);
 ora_hash
------------
 -1747984408055821656
(1 row)
gaussdb=# SELECT ora_hash(to_date('2012-1-2','yyyy-mm-dd'),234);
 ora_hash
------------
 -3306025179710572679
(1 row)
gaussdb=# SELECT ora_hash('AAAAAAAAAP//AAAAAAAAAAAAAAAAA'::rowid);
  ora_hash
------------
 2241979125
(1 row)
```

  说明

  此函数在参数a_format_version值为10c和a_format_dev_version值为s2的情况下才能生效。

- hash_array(anyarray)

  描述：数组哈希，将数组的元素通过哈希函数得到结果，并返回合并结果。

  参数：数据类型为anyarray。

  返回值类型：integer

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT hash_array(ARRAY[[1,2,3],[1,2,3]]);
 hash_array
------------
 -382888479
(1 row)
```

- hash_numeric(numeric)

  描述：计算Numeric类型的数据的hash值。

  参数：Numeric类型的数据。

  返回值类型：integer

  示例：

  ```
  gaussdb=# SELECT hash_numeric(30);
   hash_numeric
  --------------
    -282860963
  (1 row)
  ```

- hash_range(anyrange)

  描述：计算range的哈希值。

  参数：anyrange类型的数据。

  返回值类型：integer

  示例：

  ```
  gaussdb=# SELECT hash_range(numrange(1.1,2.2));
   hash_range
  ------------
   683508754
  (1 row)
  ```

- hashbpchar(character)

  描述：计算bpchar的哈希值。

  参数：character类型的数据。

  返回值类型：integer

  示例：

  ```
  gaussdb=# SELECT hashbpchar('hello');
   hashbpchar
  -------------
   -1870292951
  (1 row)
  ```

- hashchar(char)

  描述：char和布尔数据转换为哈希值。

  参数：char类型的数据或者bool类型的数据。

  返回值类型：integer

  示例：

  ```
  gaussdb=# SELECT hashbpchar('hello');
   hashbpchar
  -------------
   -1870292951
  (1 row)

  gaussdb=# SELECT hashchar('true');
    hashchar
  ------------
   1686226652
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- hashenum(anyenum)
  描述：枚举类型转哈希值。
  参数：anyenum类型的数据。
  返回值类型：integer
  示例：

```
gaussdb=# CREATE TYPE b1 AS ENUM('good', 'bad', 'ugly');
CREATE TYPE
gaussdb=# call hashenum('good'::b1);
 hashenum
------------
 1821213359
(1 row)
gaussdb=# DROP TYPE b1;
```

- hashfloat4(real)
  描述：float4转哈希值。
  参数：real类型的数据。
  返回值类型：integer
  示例：

```
gaussdb=# SELECT hashfloat4(12.1234);
 hashfloat4
------------
 1398514061
(1 row)
```

- hashfloat8(double precision)
  描述：float8转哈希值。
  参数：double precision类型的数据。
  返回值类型：integer
  示例：

```
gaussdb=# SELECT hashfloat8(123456.1234);
 hashfloat8
------------
 1673665593
(1 row)
```

- hashinet(inet)
  描述：inet / cidr转哈希值。
  参数：inet类型的数据。
  返回值类型：integer
  示例：

```
gaussdb=# SELECT hashinet('127.0.0.1'::inet);
 hashinet
-------------
 -1435793109
(1 row)
```

- hashint1(tinyint)
  描述：INT1转哈希值。
  参数：tinyint类型的数据。
  返回值类型：uint32
  示例：

```
gaussdb=# SELECT hashint1(20);
 hashint1
```


版权所有 © 华为云计算技术有限公司

```
-------------
 -2014641093
(1 row)
```

- hashint2(smallint)
  描述：INT2转哈希值。
  参数：smallint类型的数据。
  返回值类型：uint32
  示例：

```
gaussdb=# SELECT hashint2(20000);
  hashint2
------------
 -863179081
(1 row)
```

- bucketchar()
  描述：计算入参的哈希值。
  参数：char, integer
  返回值类型：integer
- bucketdate()
  描述：计算入参的哈希值。
  参数：date, integer
  返回值类型：integer
- bucketfloat4()
  描述：计算入参的哈希值。
  参数：real, integer
  返回值类型：integer
- bucketfloat8()
  描述：计算入参的哈希值。
  参数：double precision, integer
  返回值类型：integer
- bucketint1()
  描述：计算入参的哈希值。
  参数：tinyint, integer
  返回值类型：integer
- bucketint2()
  描述：计算入参的哈希值。
  参数：smallint, integer
  返回值类型：integer
- bucketint2vector()
  描述：计算入参的哈希值。
  参数：int2vector, integer
  返回值类型：integer
- bucketint4()
  描述：计算入参的哈希值。

版权所有 © 华为云计算技术有限公司

参数：integer, integer
返回值类型：integer

- bucketint8()
  描述：计算入参的哈希值。
  参数：bigint, integer
  返回值类型：integer
- bucketinterval()
  描述：计算入参的哈希值。
  参数：interval, integer
  返回值类型：integer
- bucketname()
  描述：计算入参的哈希值。
  参数：name, integer
  返回值类型：integer
- bucketnumeric()
  描述：计算入参的哈希值。
  参数：numeric, integer
  返回值类型：integer
- bucketnvarchar2()
  描述：计算入参的哈希值。
  参数：nvarchar2, integer
  返回值类型：integer
- bucketoid()
  描述：计算入参的哈希值。
  参数：oid, integer
  返回值类型：integer
- bucketoidvector()
  描述：计算入参的哈希值。
  参数：oidvector, integer
  返回值类型：integer
- bucketraw()
  描述：计算入参的哈希值。
  参数：raw, integer
  返回值类型：integer
- bucketreltime()
  描述：计算入参的哈希值。
  参数：reltime, integer
  返回值类型：integer
- bucketsmalldatetime()
  描述：计算入参的哈希值。

版权所有 © 华为云计算技术有限公司

参数：smalldatetime, integer
返回值类型：integer

- buckettext()
  描述：计算入参的哈希值。
  参数：text, integer
  返回值类型：integer
- buckettime()
  描述：计算入参的哈希值。
  参数：time without time zone, integer
  返回值类型：integer
- buckettimestamp()
  描述：计算入参的哈希值。
  参数：timestamp without time zone, integer
  返回值类型：integer
- buckettimestamptz()
  描述：计算入参的哈希值。
  参数：timestamp with time zone, integer
  返回值类型：integer
- buckettimetz()
  描述：计算入参的哈希值。
  参数：time with time zone, integer
  返回值类型：integer
- bucketuuid()
  描述：计算入参的哈希值。
  参数：uuid, integer
  返回值类型：integer
- bucketvarchar()
  描述：计算入参的哈希值。
  参数：character varying, integer
  返回值类型：integer

1.6.32 提示信息函数

- report_application_error()
  描述：PL执行过程中，可以使用此函数来抛ERROR。
  返回值类型：void

表 1-256 report_application_error 参数说明

参数	类型	说明	是否必选
log	text	error消息的内容。	是

版权所有 © 华为云计算技术有限公司

参数	类型	说明	是否必选
code	int4	error消息对应的error code，范围为：-20999 ~ -20000。	否

示例

```
gaussdb=# CREATE OR REPLACE FUNCTION GET_RESULT_UNKNOWN(
gaussdb(#     IN context_id int, /*context_id*/
gaussdb(#     IN col_type text /*col_type*/
gaussdb(# )RETURNS INTEGER
gaussdb-# AS $$
gaussdb$# BEGIN
gaussdb$#     if col_type is NULL then
gaussdb$#         PG_CATALOG.REPORT_APPLICATION_ERROR('invalid input for the third parameter
col_type should not be null');
gaussdb$#     end if;
gaussdb$#     PG_CATALOG.REPORT_APPLICATION_ERROR('UnSupport data type for
column_value(context: '||context_id||', '||PG_CATALOG.QUOTE_LITERAL(col_type)||')');
gaussdb$#     return -1;
gaussdb$# END;
gaussdb$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
gaussdb=# CALL GET_RESULT_UNKNOWN(NULL, NULL);
ERROR:  invalid input for the third parameter col_type should not be null
CONTEXT:  SQL statement "CALL pg_catalog.report_application_error('invalid input for the third
parameter col_type should not be null')"
PL/pgSQL function get_result_unknown(integer,text) line 4 at PERFORM
```

1.6.33 故障注入系统函数

gs_fault_inject(int64, text, text, text, text, text)

描述：该函数不能调用，调用时会报WARNING信息："unsupported fault injection"，并不会对数据库产生任何影响和改变。

参数：int64注入故障类型（0：CLOG扩展页面；1：读取CLOG页面；2：强制死锁）。

- text第二个入参在第一入参为2的模式下若为“1”则死锁，其余不死锁；第二个入参在第一入参为0、1时，表示CLOG开始扩展或读取的起始页面号。
- text第三个入参在第一入参为0、1时，表示扩展或读取的页面个数。
- text第四到六入参为预留参数。

返回值类型：int64

1.6.34 重分布函数

以下函数为重分布期间gs_redis工具所用的系统函数，用户不要主动调用：

- pg_get_redis_rel_end_ctid(text, name, int, int)
- pg_get_redis_rel_start_ctid(text, name, int, int)
- pg_enable_redis_proc_cancelable()
- pg_disable_redis_proc_cancelable()
- pg_tupleid_get_blocknum(tid)
- pg_tupleid_get_offset(tid)

版权所有 © 华为云计算技术有限公司

- pg_tupleid_get_ctid_to_bigint(ctid)

1.6.35 分布列推荐函数

分布列推荐针对的是在分布式数据库下分布列以及分布方式的推荐，目的是在进行业务迁移或业务上线时，减少选择表分布列的人力成本。

- sqladvisor.init(char, boolean, boolean, boolean, int, int)

描述：初始化参数。

返回值类型：Boolean

表 1-257 init 参数说明

参数	类型	说明	是否必选
kind	char	推荐类型，目前支持分布列推荐 'D'。	是
isUseCost	boolean	是否使用优化器，有数据的情况选择是。	是
isUseCollect	boolean	是否从收集的负载中开始分析，默认值为false。	否
isConstraintPrimaryKey	boolean	是否保持主键约束，默认值为true。	否
sqlCount	int	收集sql数量，默认值为10000，范围：1 ~ 100000。	否
maxMemory	int	分布列推荐最大占用内存，默认值为1024，范围：1 ~ 10240，单位MB。	否

- sqladvisor.set_weight_params(real, real, real)

描述：设置启发式规则不同成分的权重，调用init的时候会设置一个默认参数，分析时可以不调用该函数。

返回值类型：Boolean

表 1-258 set_weight_params 参数说明。

参数	类型	说明	是否必选
joinWeight	real	join的权重，范围：0 ~ 1000	是
groupbyWeight	real	group by的权重，范围：0 ~ 1000	是
qualWeight	real	predicate的权重，范围：0 ~ 1000	是

版权所有 © 华为云计算技术有限公司

说明

此函数非必须调用，在执行init函数的时候，会预置一个默认权重分别是join: 1.0, group by: 0.1, predicate: 0.05。

- sqladvisor.set_cost_params(bigint, boolean, text)

描述：使用Whtif代价模式可以设置的参数。

返回值类型：Boolean

表 1-259 set_cost_params 参数说明

参数	类型	说明	是否必选
maxTime	bigint	推荐的最大时间，小于等于0默认为不限时，单位min。	是
isTotalSQL	boolean	是否使用全部SQL参与计算，true：全部SQL，false：会按照百分位过滤掉代价过大或者过小的SQL。	是
compressLevel	text	压缩程度表示推荐算法搜索空间的大小，'low', 'med', 'high'。	是

说明

- 此函数非必须调用，在执行init函数的时候，会预置参数maxTime：-1，isTotalSQL：true, compressLevel :'high'。
- 用户选择的压缩程度越低，时间越长，越有可能推出更好的结果。

- sqladvisor.assign_table_type(text)

描述：指定表为复制表。

参数：表名。

返回值类型：Boolean

说明

指定复制表需要在调用analyze_query和analyze_workload前使用。

- sqladvisor.analyze_query(text, int)

描述：导入需要推荐的SQL语句，并对语句的成分进行分析。

返回值类型：Boolean

表 1-260 analyze_query 参数说明

参数	类型	说明	是否必选
query	text	SQL语句。	是
frequency	int	该语句在负载中的频率，默认值为1，范围：1 ~ 2147483647。	否

版权所有 © 华为云计算技术有限公司

说明

- 如果参数query中存在如单引号（'）等特殊字符，可以使用单引号（'）进行转义。
- 半在线模式不支持该函数。

- sqladvisor.analyze_workload()

描述：分析在线收集的负载信息。

返回值类型：Boolean

- sqladvisor.get_analyzed_result(text)

描述：获取当前表提取出来的有益成分。

参数：text

返回值类型：record

函数返回字段说明如下：

名称	类型	描述
schema_name	text	模式名。
table_name	text	表名。
col_name	text	列名。
operator	text	算子类型。
count	int	统计该操作符的次数。

- sqladvisor.run()

描述：根据指定的模式和输入的SQL进行计算分析。

返回值类型：Boolean

- sqladvisor.get_distribution_key()

描述：获取推荐结果。

说明

分析结果保存在session中，session断连结果丢失。

返回值类型：record

函数返回字段说明如下：

版权所有 © 华为云计算技术有限公司

名称	类型	描述
db_name	text	数据库名。
schema_name	text	模式名。
table_name	text	表名。
distribution_type	text	推荐的分布类型。
distribution_key	text	推荐分布列。
start_time	timestamp	推荐开始时间。
end_time	timestamp	推荐结束时间。
cost_improve	text	推荐结果对于代价的提升。
comment	text	备注。

- sqladvisor.clean()

描述：清理session中推荐过程中的全部内存。

返回值类型：Boolean

- sqladvisor.start_collect_workload(int, int)

描述：开启在线收集负载。

返回值类型：Boolean

表 1-261 start_collect_workload 参数说明

参数	类型	说明	是否必选
sqlCount	int	在线收集负载最大sql数量，默认值为10000，范围：1 ~ 100000。	是
maxMemory	int	在线收集负载最大占用内存，默认值为1024，范围：1 ~ 10240，单位MB。	是

版权所有 © 华为云计算技术有限公司

须知

- 在线收集功能只能由系统管理员调用。
- 同一时间只能收集一个数据库的负载。
- 收集负载目前只支持普通SQL和存储过程中的DML、DQL语句。

- sqladvisor.end_collect_workload()

描述：关闭在线收集负载。

返回值类型：Boolean

须知

- 关闭在线收集功能只能由系统管理员调用。

- sqladvisor.clean_workload()

描述：清理负载中的内存。

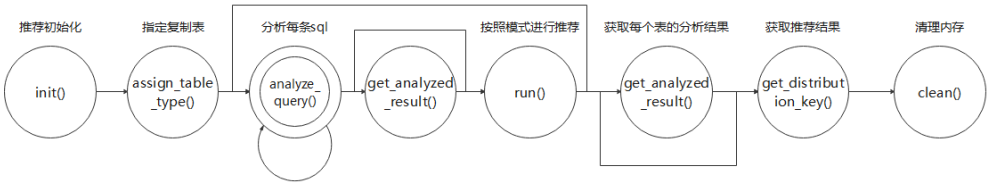
返回值类型：Boolean

须知

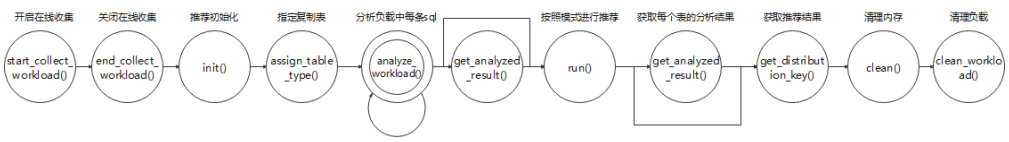
- 清理负载内存功能只能由系统管理员调用。
- 必须手动执行清理函数。

使用建议

- 启发式、WhatIf代价推荐模式调用状态机，具体流程如下图所示。



- 半在线推荐模式调用状态机，具体流程如下图所示。



1.6.36 内部函数

GaussDB中的下列函数仅用于内部逻辑功能实现，用户使用可能存在非预期结果，不建议使用。

- 选择率计算函数

版权所有 © 华为云计算技术有限公司

areajoinsel	areasel	arraycontjoinsel	arraycontsel	contjoinsel	contsel	eqjoinsel
eqsel	iclikejoinsel	iclikesel	icnlikejoinsel	icnlikesel	icregexeqjoinsel	icregexeqsel
icregexnejoinsel	icregexnesel	likejoinsel	likesel	neqjoinsel	neqsel	nlikejoinsel
nlikesel	positionjoinsel	positionsel	regexeqjoinsel	regexeqsel	regexnejoinsel	regexnesel
scalargtjoinsel	scalargtsel	scalarltjoinsel	scalarltsel	tsmatchjoinsel	tsmatchsel	-

- 统计信息收集函数

array_typanalyze	range_typanalyze	ts_typanalyze	local_rto_stat	remote_rto_stat	gs_plan_trace_delete	gs_plan_trace_watch_sqlid
gs_plan_trace_show_sqlids	standby_statement_history_internal	-	-	-	-	-

- 排序内部功能函数

bpchar_sortsupport	bytea_sortsupport	date_sortsupport	numeric_sortsupport	timestamp_sortsupport	-	-

- 内部类型处理函数

abstimerecv	euc_jis_2004_to_utf8	int2recv	line_recv	oidvectorrecv_extend	tidrecv	utf8_to_koi8u
anyarray_recv	euc_jp_to_mic	int2vectorrecv	lseg_recv	path_recv	time_recv	utf8_to_shift_jis_2004
array_recv	euc_jp_to_sjis	int4recv	macaddr_recv	pg_node_tree_recv	time_transform	utf8_to_sjis
ascii_to_mic	euc_jp_to_utf8	int8recv	mic_to_ascii	point_recv	timestamp_recv	utf8_to_uhc
ascii_to_utf8	euc_kr_to_mic	internal_out	mic_to_big5	poly_recv	timestamp_transform	utf8_to_win

版权所有 © 华为云计算技术有限公司

big5_to_euc_tw	euc_kr_to_utf8	interval_recv	mic_to_euc_cn	pound_nexttoken	timestamptz_recv	uuid_recv
big5_to_mic	euc_tw_to_big5	interval_transform	mic_to_euc_jp	prsd_nexttoken	timetz_recv	varbit_recv
big5_to_utf8	euc_tw_to_mic	iso_to_koi8r	mic_to_euc_kr	range_recv	tinterval recv	varbit_transform
bit_recv	euc_tw_to_utf8	iso_to_mic	mic_to_euc_tw	rawrecv	tsqueryrecv	varchar_transform
boolrecv	float4recv	iso_to_win1251	mic_to_iso	record_recv	tsvectorrecv	varcharrecv
box_recv	float8recv	iso_to_win866	mic_to_koi8r	regclassrecv	txid_snapshot_recv	void_recv
bpcharrecv	gb18030_to_utf8	iso8859_1_to_utf8	mic_to_latin1	regconfigrecv	uhc_to_utf8	win_to_utf8
btoidsortsupport	gbk_to_utf8	iso8859_to_utf8	mic_to_latin2	regdictionaryrecv	unknownrecv	win1250_to_latin2
byteearecv	-	johab_to_utf8	mic_to_latin3	regoperatorrecv	utf8_to_ascii	win1250_to_mic
byteawithoutorderwithequalcolrecv	gtsvector_compress	json_recv	mic_to_latin4	regoperrecv	utf8_to_big5	win1251_to_iso
cash_recv	gtsvector_consistent	koi8r_to_iso	mic_to_sjis	regprocedurerecv	utf8_to_euc_cn	win1251_to_koi8r
charrecv	gtsvector_decompress	koi8r_to_mic	mic_to_win1250	regprocrecv	utf8_to_euc_jis_2004	win1251_to_mic
cidr_recv	gtsvector_penalty	koi8r_to_utf8	mic_to_win1251	regtyperecv	utf8_to_euc_jp	win1251_to_win866
cidrecv	gtsvector_picksplit	koi8r_to_win1251	mic_to_win866	reltimerecv	utf8_to_euc_kr	win866_to_iso

版权所有 © 华为云计算技术有限公司

circle_recv	gtsvector_same	koi8r_to_win866	namerecv	shift_jis_2004_to_euc_jis_2004	utf8_to_euc_tw	win866_to_koi8r
cstring_recv	gtsvector_union	koi8u_to_utf8	ngram_nexttoken	shift_jis_2004_to_utf8	utf8_to_gb18030	win866_to_mic
date_recv	hll_recv	latin1_to_mic	numeric_recv	sjis_to_euc_jp	utf8_to_gbk	win866_to_win1251
domain_recv	hll_trans_recv	latin2_to_mic	numeric_transform	sjis_to_mic	utf8_to_iso8859	xidrecv
euc_cn_to_mic	-	latin2_to_win1250	nvarchar2recv	sjis_to_utf8	utf8_to_iso8859_1	xidrecv4
euc_cn_to_utf8	inet_recv	latin3_to_mic	oidrecv	smalldatetime_recv	utf8_to_johab	xml_recv
euc_jis_2004_to_shift_jis_2004	int1recv	latin4_to_mic	oidvectorrecv	textrecv	utf8_to_koi8r	-
numeric_bool	int2vectorin_extend	int2vectorout_extend	int2vectorrecv_extend	int2vectorsend_extend	int8_accum	large_seq_rollback_ntree
large_seq_upgrade_ntree	int16eq	int16ge	int16gt	int16in	int16le	int16lt
int16mi	int16mul	int16ne	int16out	int16pl	int16recv	int16send
int16_bool	i16toi1	anyset_in	anyset_out	btint2setcmp	btint4setcmp	btint8setcmp
btsetcmp	btsetint2cmp	btsetint4cmp	btsetint8cmp	btsetsortsupport	float4	float8
hashsetint	hashsettext	int2	int2seteq	int2setge	int2setgt	int2setle
int2setlt	int2setne	int4	int4seteq	int4setge	int4setgt	int4setle
int4setlt	int4setne	int8	int8seteq	int8setge	int8setgt	int8setle

版权所有 © 华为云计算技术有限公司

int8setlt	int8setne	set	set_in	set_out	set_recv	set_send
seteq	setge	setgt	setint2eq	setint2ge	setint2gt	setint2le
setint2lt	setint2ne	setint4eq	setint4ge	setint4gt	setint4le	setint4lt
setint4ne	setint8eq	setint8ge	setint8gt	setint8le	setint8lt	setint8ne
setle	setlt	setne	settexteq	settextge	settextgt	settextle
settextlt	settextne	settobpchar	settonumber	settonvarchar2	settotext	settovarchar
textseteq	textsetge	textsetgt	textsetle	textsetlt	textsetne	gb18030_2022_to_utf8
utf8_to_gb18030_2022	array_to_nesttable	array_to_indexby_int_table	nesttable_to_array	indexbytableint_to_array	array_to_nesttable	array_to_indexby_int_table
zhs16gbk_to_utf8	utf8_to_zhs16gbk	zhs16gbk_to_gb18030	gb18030_to_zhs16gbk	zhs16gbk_to_gb18030_2022	gb18030_2022_to_zhs16gbk	-

- 聚合操作内部函数

array_agg_finalfn	array_agg_transfn	bytea_string_agg_finalfn	bytea_string_agg_transfn	date_list_agg_noarg2_transfn	date_list_agg_transfn	float4_list_agg_noarg2_transfn
float4_list_agg_transfn	float8_list_agg_noarg2_transfn	float8_list_agg_transfn	int2_list_agg_noarg2_transfn	int2_list_agg_transfn	int4_list_agg_noarg2_transfn	int4_list_agg_transfn
int8_list_agg_noarg2_transfn	int8_list_agg_transfn	interval_list_agg_noarg2_transfn	interval_list_agg_transfn	list_agg_finalfn	list_agg_noarg2_transfn	list_agg_transfn
median	median_float8_finalfn	median_interval_finalfn	median_transfn	mode_final	numeric_list_agg_noarg2_transfn	numeric_list_agg_transfn

版权所有 © 华为云计算技术有限公司

ordered_set_transition	percentile_cont_float8_final	percentile_cont_interval_final	string_agg_finalfn	string_agg_transfn	timestamp_list_agg_noarg2_transfn	timestamp_list_agg_transfn
timestamptz_list_agg_noarg2_transfn	timestamptz_list_agg_transfn	checksumtext_agg_transfn	json_agg_transfn	json_agg_finalfn	json_object_agg_transfn	json_object_agg_finalfn
datea_list_agg_transfn	datea_list_agg_noarg2_transfn	int1_list_agg_noarg2_transfn	int1_list_agg_transfn	smalldatetime_list_agg_noarg2_transfn	smalldatetime_list_agg_transfn	rowid_list_agg_noarg2_transfn
rowid_list_agg_transfn	-	-	-	-	-	-

- 哈希内部功能函数

hashbeginscan	hashbuild	hashbuildempty	hashbulkdelete	hashcostestimate	hashendscan	hashgetbitmap
hashgettuple	hashinsert	hashmarkpos	hashmerge	hashrescan	hashrestrpos	hashvacuumcleanup
hashvarlena	jsonb_hash	datea_hash	rowid_hash	-	-	-

- Btree索引内部功能函数

cbtreebuild	cbtreecanreturn	cbtreecostestimate	cbtreegetbitmap	cbtreegettuple	btbeginscan	btbuild
btbuildempty	btbulkdelete	btcanreturn	btcostestimate	btendscan	btfloat4sortsupport	btfloat8sortsupport
btgetbitmap	btgettuple	btinsert	btint2sortsupport	btint4sortsupport	btint8sortsupport	btmarkpos
btmerge	btnamesortsupport	btrescan	btrestrpos	bttextsortsupport	btvacuumcleanup	cbtreeoptions
btvarstrequalimag	-	-	-	-	-	-

版权所有 © 华为云计算技术有限公司

- UGIN索引内部函数

ugininsert	uginbeginscan	ugingetbitmap	uginrescan	uginendscan	uginbuild	uginbuildempty
uginbulkdelete	uginvacuumcleanup	ugincostestimate	uginoptions	ugin_extract_value_trgm	ugin_extract_query_trgm	ugin_trgm_consistent
ugin_trgm_triconsistent	-	-	-	-	-	-

- GIN索引内部功能函数

gin_cmp_prefix	gin_extract_tsquery	gin_tsquery_consistent	gin_tsquery_triconsistent	ginarrayconsistent	ginarrayextract	ginarraytriconsistent
ginbeginscan	ginbuild	ginbuildempty	ginbulkdelete	gincostestimate	ginendscan	gingetbitmap
gininsert	ginmarkpos	ginmerge	ginqueryarrayextract	ginrescan	ginrestrpos	ginvacuumcleanup
cginbuild	cgingetbitmap	gin_consistent_jsonb_hash	gin_triconsistent_jsonb_hash	gin_extract_jsonb_hash	gin_extract_jsonb_query_hash	gin_triconsistent_jsonb
gin_compare_jsonb	gin_extract_jsonb	gin_extract_jsonb_query	gin_consistent_jsonb	-	-	-

- GiST索引内部功能函数

gist_box_compress	gist_box_consistent	gist_box_decompress	gist_box_penalty	gist_box_picksplit	gist_box_same	gist_box_union
gist_circle_compress	gist_circle_consistent	gist_point_compress	gist_point_consistent	gist_point_distance	gist_poly_compress	gist_poly_consistent
gistbeginscan	gistbuild	gistbuildempty	gistbulkdelete	gistcostestimate	gistendscan	gistgetbitmap
gistinsert	gistmarkpos	gistmerge	gistrescan	gistrestrpos	gistvacuumcleanup	range_gist_compress

版权所有 © 华为云计算技术有限公司

range_gist_decompress	range_gist_penalty	range_gist_picksplit	range_gist_same	range_gist_union	spg_kd_choose	spg_kd_config
spg_kd_picksplit	spg_quad_choose	spg_quad_config	spg_quad_inner_consistent	spg_quad_leaf_consistent	spg_quad_picksplit	spg_text_choose
spg_text_inner_consistent	spg_text_leaf_consistent	spg_text_picksplit	spgbeginscan	spgbuild	spgbuildempty	spgbulkdelete
spgcostestimate	spgendscan	spggetbitmap	spggettuple	spginsert	spgmarkpos	spgmerge
spgrestrpos	spgvacuumcleanup	-	-	-	-	-

- Psort索引内部函数

psortbuild	psortcanreturn	psortcostestimate	psortgetbitmap	psortgettuple	-	-

- Ubtree索引内部函数

ubtbeginscan	ubtbuild	ubtbuildempty	ubtbulkdelete	ubtcanreturn	ubtcostestimate	ubtendscan
ubtgetbitmap	ubtgettuple	ubtinsert	ubtmarkpos	ubtmerge	ubtoptions	ubtrescan
ubtrestrpos	ubtvacuumcleanup	-	-	-	-	-

- BM25索引内部函数（当前版本不支持）

gs_bm25beginscan	gs_bm25build	gs_bm25buildempty	gs_bm25bulkdelete	gs_bm25insert	gs_bm25costestimate	gs_bm25endscan
gs_bm25options	gs_bm25gettuple	gs_bm25vacuumcleanup	gs_bm25rescan	-	-	-

- plpgsql内部函数

版权所有 © 华为云计算技术有限公司

plpgsql_inline_handler	-	-	-	-	-	-

- 外表相关内部函数

dist_fdw_handler	roach_handler	streaming_fdw_handler	dist_fdw_validator	file_fdw_handler	file_fdw_validator	log_fdw_handler
gc_fdw_handler	gc_fdw_validator	dblink_fdw_handler	dblink_fdw_validator	-	-	-

- 数据倾斜优化相关内部函数

distributed_count	-	-	-	-	-	-

- 表统计信息相关内部函数

pgxc_get_stat_dirty_tables	pgxc_stat_dirty_tables	get_global_stat_all_tables	get_summary_stat_all_tables	-	-	-

- 远程读取数据函数
  - gs_read_block_from_remote 用于读取非段页式表文件的页面。默认只有初始化用户可以查看，其余用户需要赋权后才可以使用。
  - gs_read_segment_block_from_remote 用于读取段页式表文件的页面。默认只有初始化用户可以查看，其余用户需要赋权后才可以使用。
- 远程读取文件函数
  - gs_read_file_size_from_remote 用于读取指定文件的大小，gs_repair_file函数修复文件时，要先获取远端关于这个文件的大小，用于校验本地文件缺失的文件信息，然后将缺失的文件逐个修复。默认只有初始化用户可以查看，其余用户需要赋权后才可以使用。
  - gs_read_file_from_remote 用于读取指定的文件，gs_repair_file利用gs_read_file_size_from_remote函数获取文件大小后，依赖这个函数将远端文件逐段读取。默认只有初始化用户可以查看，其余用户需要赋权后才可以使用。
- 以备DN实例增量重建其他备或级联备DN实例辅助函数
  - gs_standby_incremental_filemap_create用于创建备DN增量重建临时filemap文件，用于存储当次增量重建需传输的数据路径与大小。只有初始化用户且application为gs_rewind时可以调用。
  - gs_standby_incremental_filemap_insert用于向指定临时filemap文件中插入文件信息，指定文件的路径、传输起始点、单次传输数据长度和rebuild标志位。只有初始化用户且application为gs_rewind时可以调用。

版权所有 © 华为云计算技术有限公司

  - gs_standby_incremental_filemap_execute用于获取指定临时filemap文件中存储的文件信息并删除指定filemap，用于备DN增量重建的数据传输。只有初始化用户且application为gs_rewind时可以调用。

- 段页式获取远程信息函数

remote_gs_seg_get_datafiles	remote_gs_seg_get_extents	remote_gs_seg_get_segment	remote_gs_seg_get_spc_extents	remote_gs_seg_spc_segments	remote_segment_space_info	-

- 账本数据库函数

get_dn_hist_relhash	-	-	-	-	-	-

- 其他函数

to_datea	sysdate_a	get_char_max_length	to_datea_internal	gs_get_part_relfilenode_id	gs_check_internal	-

- 视图相关引用函数

adm_hist_sqlstat_func	adm_hist_sqlstat_idlog_func	adm_hist_sqltext_func	-	-	-	-

- gs_txn_snapshot系统表维护函数

  gs_insert_delete_txn_snapshot用于分布式GTM-Lite模式下维护全局各节点gs_txn_snapshot系统表，只有系统管理员用户才能调用。

- xmltype类型相关函数

isschemavalid(不可用)	-	-	-	-	-	-

- 在线创建与删除索引内部函数

gs_mark_indisvalid	gs_mark_indisvalid_false	-	-	-	-	-

- pivot相关函数

版权所有 © 华为云计算技术有限公司

pivot_anyelement	pivot_bit	pivot_bool	pivot_bpchar	pivot_bytea	pivot_date	pivot_float4
pivot_float8	pivot_hll	pivot_int1	pivot_int2	pivot_int4	pivot_int8	pivot_interval
pivot_json	pivot_money	pivot_numeric	pivot_oid	pivot_smalldatetime	pivot_tdigestdata	pivot_text
pivot_tid	pivot_time	pivot_datea	pivot_timestamp	pivot_timestamptz	pivot_timetz	pivot_xml
pivot_char	pivot_varchar	pivot_nvarchar2	pivot_raw	pivot_clob	pivot_abstime	pivot_reltime
pivot_blob	pivot_rowid	pivot_varbit	-	-	-	-

- subtype相关函数

subtype_in	subtype_recv	-	-	-	-	-

- nesttable相关函数

nesttable_in	nesttable_out	nesttable_count	nesttable_delete	nesttable_eq	nesttable_except	nesttable_except_distinct
nesttable_exists	nesttable_extend	nesttable_first	nesttable_intersect	nesttable_intersect_distinct	nesttable_last	nesttable_limit
nesttable_ne	nesttable_next	nesttable_prior	nesttable_to_array	nesttable_trim	nesttable_union	nesttable_union_distinct

- indexbytable相关函数

indexbytable_count	indexbytable_in	indexbytable_out	indexbytableint_delete	indexbytableint_exists	indexbytableint_first	indexbytableint_last
indexbytableint_next	indexbytableint_prior	indexbytableint_to_array	indexbytablevarchar_delete	indexbytablevarchar_exists	indexbytablevarchar_first	indexbytablevarchar_last

版权所有 © 华为云计算技术有限公司

indexbytablevarchar_next	indexbytablevarchar_prior	-	-	-	-	-

- 消息队列相关函数

mq_clear	mq_get_queue	mq_query	mq_register_pull	mq_send_message	mq_sync	mq_sync_ccn
mq_transaction_processing	mq_unregister	-	-	-	-	-

- 排序计算相关函数

tdigest_in	tdigest_out	calculate_quantile_of	calculate_value_at	tdigest_merge	tdigest_merge_to_one	tdigest_mergep

- GsIVFFLAT索引内部函数

ivfflatbeginscan	ivfflatbuild	ivfflatbuildempty	ivfflatbulkdelete	ivfflatinsert	ivfflatcostestimate	ivfflatendscan
ivfflatoptions	ivfflatgettuple	ivfflatvacuumcleanup	ivfflatrescan	-	-	-

- GsDiskANN索引内部函数

diskannbeginscan	diskannbuild	diskannbuildempty	diskannbulkdelete	diskanninsert	diskanncostestimate	diskannendscan
diskannoptions	diskanngettuple	diskannvacuumcleanup	diskannrescan	-	-	-

1.6.37 敏感数据发现函数

- gs_sensitive_data_discovery(scan_target text, scan_classifier text)
  描述：扫描目标数据，返回统计的扫描结果。
  参数：

版权所有 © 华为云计算技术有限公司

scan_target：指定扫描对象，取值必须是schema、table或column的名称，但必须指定扫描对象的上级名称，比如扫描某列传入扫描对象为schema_name.table_name.column_name。

scan_classifier：指定使用的分类器，支持指定email、creditcard、phonenumber、chinesename、encryptedcontent5种分类器，多选可以用逗号分隔，或者使用all选中所有分类器。

返回值类型：record

示例请参见《开发指南》中“特性使用指导 > 敏感数据发现”章节。

- gs_sensitive_data_discovery_detail(scan_target text, scan_classifier text)

  描述：扫描目标数据，返回详细的扫描结果。

  scan_target：指定扫描对象，取值必须是schema、table或column的名称，但必须指定描对象的上级名称，比如扫描某列传入扫描对象为schema_name.table_name.column_name。

  scan_classifier：指定使用的分类器，支持email、creditcard、phonenumber、chinesename、encryptedcontent5种分类器，多选可以用逗号分隔，或者使用all选中所有分类器。

  返回值类型：record

  示例请参见《开发指南》中“特性使用指导 > 敏感数据发现”章节。

1.6.38 动态数据脱敏函数

- creditcardmasking(col text, letter char default 'x')

  描述：将col字符串后四位之前的数字使用letter替换。

  参数：待替换的字符串、替换字符。

  返回值类型：text

- basicemailmasking(col text, letter char default 'x')

  描述：将col字符串中第一个'@'之前的字符使用letter替换。

  参数：待替换的字符串、替换字符。

  返回值类型：text

- fullemailmasking(col text, letter char default 'x')

  描述：将col字符串中出现最后一个'.'之前的字符(除第一个'@'外)使用letter替换。

  参数：待替换的字符串、替换字符。

  返回值类型：text

- alldigitsmasking(col text, letter char default '0')

  描述：将col字符串中出现的数字使用letter替换。

  参数：待替换的字符串、替换字符。

  返回值类型：text

- shufflemasking(col text)

  描述：将col字符串中的字符乱序排列。

  参数：待替换的字符串、替换字符。

  返回值类型：text

- randommasking(col text)

  描述：将col字符串中的字符随机化。

版权所有 © 华为云计算技术有限公司

参数：待替换的字符串、替换字符。

返回值类型：text

- **可选:** regexpmasking(col text, reg text, replace_text text, pos INTEGER default 0, reg_len INTEGER default -1)

  描述：将col字符串使用正则表达式替换。

  参数：待替换的字符串、正则表达式、替换的起始位置、替换长度。

  返回值类型：text

1.6.39 层次递归查询函数

层次递归查询语句中可使用如下函数返回连接路径中的相关信息。

- sys_connect_by_path(col, separator)

  描述：仅在层次递归查询中适用，用于返回从根节点到当前行的连接路径。

  参数col为在路径中显示的列的名称，只支持类型为CHAR、VARCHAR、NVARCHAR2、TEXT、ROWID、INT1、INT2、INT4、INT8、FLOAT4、FLOAT8和NUMERIC的列，参数separator为路径节点之间的分隔符。当前该函数col输入不支持表达式输入，列内容与分隔符separator重复时可正常运行，不会报错。

  返回值类型：TEXT

  示例：

```
--建表导数。
gaussdb=# CREATE TABLE connect_by_table(id int,pid int,name text);
CREATE TABLE
gaussdb=# INSERT INTO connect_by_table VALUES(1,0,'a'),(2,1,'b'),(3,2,'c'),(4,1,'d');
INSERT 0 4
--查询。
gaussdb=# SELECT *, sys_connect_by_path(name, '-') FROM connect_by_table START WITH id = 1
CONNECT BY prior id = pid ORDER BY pid, id;
 id | pid | name | sys_connect_by_path
----+-----+------+---------------------
  1 |   0 | a    | -a
  2 |   1 | b    | -a-b
  4 |   1 | d    | -a-d
  3 |   2 | c    | -a-b-c
(4 rows)
--恢复环境。
gaussdb=# DROP TABLE connect_by_table;
DROP TABLE
```

- connect_by_root(col)

  描述：仅在层次递归查询中适用，用于返回当前行最顶层父行中某列的值。

  参数col为输出列的名称。仅支持如下类型的列：INT8、INT1、INT2、OID、INT4、BOOL、CHAR、NAME、FLOAT4、FLOAT8、ABSTIME、RELTIME、DATE、CASH、TIME、TIMESTAMP、TIMESTAMPTZ、SMALLDATETIME、UUID、INTERVAL、TIMETZ、INT2VECTOR、CLOB、NVARCHAR2、VARCHAR、TEXT、ROWID、VECTOR、BPCHAR、RAW、BYTEA、NUMERIC、XID、CID以及TID，且通过强制类型转换为上述类型的操作无法绕开白名单限制，如：connect_by_root(col::text)（其中col不是上述白名单中的类型）。

  返回值类型：所指定列col的数据类型。

  示例：

```
--建表导数。
gaussdb=# CREATE TABLE connect_by_table(id int,pid int,name text);
CREATE TABLE
gaussdb=# INSERT INTO connect_by_table VALUES(1,0,'a'),(2,1,'b'),(3,2,'c'),(4,1,'d');
INSERT 0 4
```


版权所有 © 华为云计算技术有限公司

```
--查询。
gaussdb=# SELECT *, connect_by_root(name) FROM connect_by_table START WITH id = 1 CONNECT
BY prior id = pid ORDER BY pid, id;
 id | pid | name | connect_by_root
----+-----+------+-----------------
  1 |   0 | a    | a
  2 |   1 | b    | a
  4 |   1 | d    | a
  3 |   2 | c    | a
(4 rows)
--恢复环境。
gaussdb=# DROP TABLE connect_by_table;
DROP TABLE
```

1.6.40 hotkey 特性函数

- gs_stat_get_hotkeys_info()

  描述：获取本地节点查询的热词信息。

  返回值类型：Tuple

  示例：

```
gaussdb=# SELECT * FROM  gs_stat_get_hotkeys_info() order by count, hash_value;
 database_name | schema_name |     table_name      | key_value | hash_value | count
---------------+-------------+---------------------+-----------+------------+-------
 regression    | public      | hotkey_single_col | {22}      | 1858004829 |     2
 regression    | public      | hotkey_single_col | {11}      | 2011968649 |     2
(2 rows)
```

- gs_stat_clean_hotkeys()

  描述：清理hotkey缓存，重置hotkey状态信息。

  返回值类型：bool，恒为true

  示例：

```
gaussdb=# SELECT * FROM  gs_stat_clean_hotkeys();
 gs_stat_clean_hotkeys
-----------------------
 t
(1 row)
```

1.6.41 Global SysCache 特性函数

- gs_gsc_table_detail(database_id default NULL, rel_id default NULL)

  描述：查看数据库里全局系统缓存的表元数据。调用该函数的用户需要具有SYSADMIN权限。

  参数：指定需要查看全局系统缓存的数据库和表，database_id默认值NULL或者-1表示所有的数据库，0表示共享表，其他数字表示指定数据库及共享表，rel_id表示指定表的oid，默认值NULL或者-1表示所有的表，其他值表示指定的表，database_id不存在会报错，rel_id 不存在查询结果为空。

  返回值类型：Tuple

```
gaussdb=# SELECT * FROM  gs_gsc_table_detail(-1) limit 1;
database_oid | database_name | reloid |          relname          | relnamespace | reltype | reloftype |
relowner | relam | relfilenode | reltablespace | relhasindex | relisshared | relkind | relnatts | relhasoids |
relhaspkey | parttype | tdhasuids | attnames  | extinfo
--------------+---------------+--------+---------------------------+--------------+---------+-----------+----------
+-------+-------------+---------------+-------------+-------------+---------+----------+------------+------------
+----------+-----------+-----------+---------
0 |               |   2676 | pg_authid_rolname_index |           11 |       0 |         0 |       10 |   403 |           0
|          1664 | f           | t           | i       |         1 | f           | f           | n       | f          | 'rolname' |
(1 row)
```


版权所有 © 华为云计算技术有限公司

- gs_gsc_catalog_detail(database_id default NULL, rel_id default NULL)

  描述：查看数据库里全局系统缓存的系统表行信息。调用该函数的用户需要具有SYSADMIN权限。

  参数：指定需要查看全局系统缓存的数据库和表，database_id 默认值NULL或者-1表示所有的数据库，0表示共享表，其他数字表示指定数据库及共享表，rel_id表示指定表的id，仅包含所有有系统缓存的系统表，默认值NULL或者-1表示所有的表，其他值表示指定的表，database_id不存在会报错，rel_id不存在结果为空。

  返回值类型：Tuple

  示例：

  ```
  --首先通过pg_database获取特定数据库的oid，查询语句一般为：SELECT oid, * FROM pg_database;
  --返回元组中通过datname列找到对应的oid列的值，然后执行如下查询，示例获取的oid为16574。
  gaussdb=# SELECT * FROM  gs_gsc_catalog_detail(16574, 1260);
   database_id | database_name | rel_id | rel_name  | cache_id |  self  |  ctid  | infomask | infomask2 |
  hash_value | refcount
  -------------+---------------+--------+-----------+----------+--------+--------+----------+-----------+------------
  +----------
          0 |               |   1260 | pg_authid |       10 | (0, 9) | (0, 9) |    10507 |        26 |  531311568 |
  10
          0 |               |   1260 | pg_authid |       11 | (0, 4) | (0, 4) |     2313 |        26 |  365368336 |         1
          0 |               |   1260 | pg_authid |       11 | (0, 9) | (0, 9) |    10507 |        26 | 3911517328 |
  10
          0 |               |   1260 | pg_authid |       11 | (0, 7) | (0, 7) |     2313 |        26 | 1317799983 |
  1
          0 |               |   1260 | pg_authid |       11 | (0, 5) | (0, 5) |     2313 |        26 | 3664347448 |
  1
          0 |               |   1260 | pg_authid |       11 | (0, 1) | (0, 1) |     2313 |        26 |  276477273 |         1
          0 |               |   1260 | pg_authid |       11 | (0, 3) | (0, 3) |     2313 |        26 | 2465837659 |
  1
          0 |               |   1260 | pg_authid |       11 | (0, 8) | (0, 8) |     2313 |        26 | 3205288035 |
  1
          0 |               |   1260 | pg_authid |       11 | (0, 6) | (0, 6) |     2313 |        26 |  131811687 |         1
          0 |               |   1260 | pg_authid |       11 | (0, 2) | (0, 2) |     2313 |        26 | 1226484587 |
  1
  (10 rows)
  ```

- gs_gsc_clean(database_id default NULL)

  描述：清理global syscache的缓存，需要注意，正在使用中的数据不会被清理。调用该函数的用户需要具有SYSADMIN权限。

  参数：指定需要清理全局系统缓存的数据库，默认值NULL或者-1表示强制清理所有的数据库全局系统缓存，0表示只淘汰共享表的全局系统缓存，其他数字表示淘汰指定数据库以及共享表的全局系统缓存，database_id不存在会报错。

  返回值类型：Boolean

  示例：

  ```
  gaussdb=# SELECT * FROM  gs_gsc_clean();
  gs_gsc_clean
  --------------
  t
  (1 row)
  ```

- gs_gsc_dbstat_info(database_id default NULL)

  描述：获取本地节点的GSC的内存统计信息，包括tuple、relation、partition的缓存查询，命中，加载、失效、占用空间信息，DB级别的淘汰信息，线程引用信息，内存占用信息。可以用于定位性能问题，例如当发现hits/searches数组远小于1时，可能是global_syscache_threshold设置太小，导致查询命中率下降。调用该函数的用户需要具有SYSADMIN权限。

版权所有 © 华为云计算技术有限公司

参数：指定需要查看的数据库全局系统缓存统计信息，NULL或者-1表示查看所有的数据库，0表示只查看共享表信息，其他数字表示查看指定的数据库和共享表的信息。不合法的输入值，database_id不存在会报错。

返回值类型：Tuple

示例：

```
gaussdb=# SELECT * FROM  gs_gsc_dbstat_info();
database_id | database_name | tup_searches | tup_hits | tup_miss | tup_count | tup_dead | tup_memory
| rel_searches | rel_hits | rel_mis
s | rel_count | rel_dead | rel_memory | part_searches | part_hits | part_miss | part_count | part_dead |
part_memory | total_memory | swa
pout_count | refcount
-------------+---------------+--------------+----------+----------+-----------+----------+------------
+--------------+----------+--------
--+-----------+----------+------------+---------------+-----------+-----------+------------+-----------
+-------------+--------------+----
-----------+----------
          0 |               |          300 |      235 |       31 |        22 |        2 |       9752 |          598 |      108
|       1
8 |        18 |        0 |      77720 |            0 |         0 |         0 |          0 |         0 |          0 |       752912 |
         0 |         0
      16574 | testdb      |         3368 |     2289 |      329 |       273 |        0 |      92593 |         1113 |
524 |       4
8 |        48 |        0 |     340456 |            0 |         0 |         0 |          0 |         0 |          0 |      4124792 |
         0 |        10
(2 rows)
```

1.6.42 数据损坏检测修复函数

- 修复主机备机文件页面的约束概述：

文件类型	文件页面级别	主备机	检测修复
普通行存表（Astore，Ustore），段页式表，跨页透明压缩表，不包括索引、压缩表	文件&页面	主机	手动检测手动修复。
undo（不包括 undo meta）	页面	主机	手动检测手动修复（不包括analyse verify）。
unlogged表的init fork文件	文件	主机	手动检测手动修复。
普通行存表（Astore，Ustore），ILM压缩表，跨页透明压缩表，段页式，Hashbucket表，索引（Btree，UBtree），undo（不包括undo meta，undo只支持CRC校验错误）	页面	备机	回放过程中自动检测自动修复。

版权所有 © 华为云计算技术有限公司

⚠ 注意

备机修复支持备集群的首备和级联备。

数据库内部使用临界区保护关键资源的访问的一致性，不允许在临界区内抛出错误。如果在临界区内，访问到损坏的页面，则会触发PANIC，不支持修复检测及自动修复。典型的执行过程中访问临界区的操作包括INSERT\DELETE\UPDATE\DROP等数据库写操作。

AutoVacuum和analyze执行期间，为保证读取页面的准确性，会临时关闭远程访问功能，可能导致备机修复主机失败。

- gs_verify_data_file(verify_segment bool)

  描述：校验当前实例当前库是否存在文件丢失的情况。校验只包括数据表主文件是否有中间段丢失的情况。默认参数是false，表示不校验段页式表数据文件。参数设置为true时仅校验段页式表文件，不支持hashbucket的文件校验。默认只有初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以查看，其余用户需要赋权后才可以使用。

  返回的结果：

  - 非段页式表：rel_oid和rel_name是对应文件的表oid和表名，miss_file_path表示丢失文件的相对路径。
  - 段页式表：因所有表存放在相同文件中，所以rel_oid和rel_name无法显示具体表的信息。对于段页式表，如果第一个文件损坏，不会检查出后面的.1 .2等文件。例如3、3.1、3.2损坏，只能检查出3损坏。当段页式文件不足5个时，使用函数检测时，未生成的文件也会校验出来，例如只有1和2文件，校验段页式时，也会检测出3，4，5文件。以下示例，第一个是校验非段页式表的示例，第二是校验段页式表的示例。

  参数说明：

  - verify_segment

    指定文件校验的范围。false校验非段页式表；true校验段页式表。

    取值范围：true和false，默认是false。

  返回值类型：record

  示例（仅当发现有异常时才会输出异常行，否则输出0行）：

  校验非段页式表：

  ```
  gaussdb=# SELECT * FROM  gs_verify_data_file();
  node_name         | rel_oid |  rel_name    |  miss_file_path
  ------------------+---------+--------------+------------------
  dn_6001_6002_6003 |   16554 |     test     | base/16552/24745
  ```

  校验段页式表：

  ```
  gaussdb=# SELECT * FROM  gs_verify_data_file(true);
      node_name      | rel_oid | rel_name | miss_file_path
  -------------------+---------+----------+----------------
   dn_6001_6002_6003 |       0 | none     | base/16552/2
  ```

- gs_repair_file(tableoid Oid, path text, timeout int)

  描述：根据传入的参数修复文件，仅支持有正常主备连接的主DN使用。只支持数据表主文件修复，参数依据gs_verify_data_file函数返回的oid和路径填写。段页式表tableoid赋值为0到4294967295的任意值（内部校验根据文件路径判断是否是段页式表文件，段页式表文件则不使用tableoid）。修复成功返回值为true，修复失败会显示具体失败原因。默认只有在主DN节点上，使用初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以查看，其余用户需要赋权后才可以使用。

版权所有 © 华为云计算技术有限公司

注意

1. 当DN实例上存在文件损坏时，进行升主会校验出错，报PANIC退出无法升主，为正常现象。
2. 当文件存在但是大小为0时，此时不会去修复该文件，若想要修复该文件，需要将为0的文件删除后再修复。
3. 删除文件需要等文件句柄（fd，file descriptor）自动关闭后再修复，人工操作可以执行重启进程，主备切换等命令。
4. 该修复函数要求主备机待修复的文件路径应保持一致，如果不一致，则修复失败。也就是说，如果使用RELATIVE LOCATION表空间，其路径因tblspcversiondir不同，导致路径不同，修复失败。

参数说明：

- tableoid

  要修复的文件对应的表oid，依据gs_verify_data_file函数返回的列表中rel_oid一列填写。

  取值范围： Oid，0 - 4294967295。注意：输入负值等数值都会被强制转成非负整数类型。

- path

  需要修复的文件路径，依据gs_verify_data_file函数返回的列表中miss_file_path一列填写。

  取值范围：字符串。

- timeout

  等待备DN回放的时长，修复文件需要等待备DN回放到当前主DN对应的位置，根据备DN回放所需时长设定。

  取值范围：60s - 3600s。

返回值类型：Boolean

示例（按照gs_verify_data_file的输出填写tableoid和path）：

页式存储：

```
gaussdb=# SELECT * FROM gs_repair_file(16554,'base/16552/24745',360);
gs_repair_file
----------------
t
```

段页式存储：

```
gaussdb=# SELECT * FROM gs_repair_file(16554,'base/16552/2',360);
gs_repair_file
----------------
t
```

- local_bad_block_info()

  描述：显示本实例页面损坏的情况。从磁盘读取页面，发现页面CRC校验失败时进行记录。默认只有初始用户、具有sysadmin属性的用户、具有监控管理员属性的用户以及在运维模式下具有运维管理员属性的用户、以及监控用户可以查看，其余用户需要赋权后才可以使用。file_path是损坏文件的相对路径，支持段页式。block_num是该文件损坏的具体页面号，页面号从0开始。check_time表示发现页面损坏的时间。repair_time表示修复页面的时间。

  返回值类型：record

  示例：（仅当有损坏记录时输出相关条目，否则输出0行。）

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM  local_bad_block_info();
node_name    | spc_node | db_node | rel_node| bucket_node | fork_num | block_num |    file_path     |
check_time            |   repair_time
-----------------+-------+--------+--------+-------------+----------+-----------+-----------------
+-------------------------+-------------------------------
dn_6001_6002_6003|  1663 |  16552 |  24745 |       -1    |    0    | 0        | base/16552/24745 |
2022-01-13 20:19:08.385004+08 | 2022-01-13 20:19:08.407314+08
```

- remote_bad_block_info()

  描述：CN上查询时，查询除本实例以外其他实例页面损坏的情况，记录数据和在其他实例上执行local_bad_block_info函数一致。DN上执行结果为空。默认只有初始用户、具有sysadmin属性的用户、具有监控管理员属性的用户以及在运维模式下具有运维管理员属性的用户、以及监控用户可以查看，其余用户需要赋权后才可以使用。

  返回值类型：record

- local_clear_bad_block_info()

  描述：清理local_bad_block_info中已修复页面的数据，也就是repair_time不为空的信息。默认只有初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户、以及监控用户可以查看，其余用户需要赋权后才可以使用。

  返回值类型：Boolean

  示例：

  ```
  gaussdb=# SELECT * FROM  local_clear_bad_block_info();
  result
  --------
  t
  ```

- remote_clear_bad_block_info()

  描述：CN上执行时，清理除本实例以外其他实例已修复页面的数据，也就是repair_time不为空的信息。DN上执行结果为空。默认只有初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户、以及监控用户可以查看，其余用户需要赋权后才可以使用。

  返回值类型：record

- gs_inconsistency_info(reset bool)

  描述：显示/重置数据库中存在的数据不一致相关信息，入参reset为true时进行重置，为false时则进行显示。默认只有初始用户、具有sysadmin属性的用户、具有监控管理员属性的用户、在运维模式下具有运维管理员属性的用户以及监控用户可以查看，其余用户需要赋权后才可以使用。其中，inconsistency_type是不一致的类型，node_name是节点名，spc_node是表空间ID，db_node是库节点ID，rel_node是表ID，bucket_node是hashbucket的ID，backend是临时表的ID，index_oid是索引的oid，tuple_blkno是元组所在堆表的页号，tuple_offset是元组所在堆表页的偏移，count是这一条不一致信息出现的次数，last_update_time是最后更新这条不一致信息的时间。

  返回值类型：record

  示例：（在入参reset为false，且当有不一致信息被校验出来时，才会输出相关条目；其他情况下输出0行。）

  ```
  gaussdb=# SELECT * FROM gs_inconsistency_info(false);
   inconsistency_type  | node_name | spc_node | db_node | rel_node | bucket_node | backend | index_oid
  | tuple_blkno | tuple_offset | count |       last_update_time
  --------------------+-----------+----------+---------+----------+-------------+---------+-----------+-------------
  +--------------+-------+-------------------------------
   INCONSISTENCY_INDEX | sgnode    |     1663 |   12902 |    16722 |          -1 |      -1 |     16728 |
  0 |            1 |     1 | 2024-11-20 16:18:56.85289-05
  ```


版权所有 © 华为云计算技术有限公司

- gs_verify_and_tryrepair_page (path text, blocknum Oid, verify_mem bool, is_segment bool)

  描述：校验本实例指定页面的情况。默认只有在主DN节点上，使用初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以查看，其余用户需要赋权后才可以使用。返回的结果信息，disk_page_res表示磁盘上页面的校验结果，mem_page_res表示内存中页面的校验结果，is_repair表示在校验的过程中是否触发修复功能，t表示已修复，f表示未修复。

  说明

  - 当DN实例上存在页面损坏时，进行升主会校验出错，报PANIC退出而无法升主，这属于正常现象。
  - 此函数触发的修复仅支持修复内存中的页面，需要在内存页面落盘后的物理页面修复后，才能正式生效。

  参数说明：

  - path

    损坏文件的路径，依据local_bad_block_info中file_path一列填写。如果要对存储类型为USTORE的表进行UNDO页面校验，请直接填写需要校验的UNDO页面路径。

    取值范围：字符串。

  - blocknum

    损坏文件的页号，依据local_bad_block_info中block_num一列填写。如果要对存储类型为USTORE的表进行UNDO页面校验，请直接填写需要校验的UNDO页面的块号。

    取值范围：Oid，0 - 4294967295。注意：输入负值等都会被强制转成非负整数类型。

  - verify_mem

    指定是否校验内存中的指定页面。设定为false时，只校验磁盘上的页面。设置为true时，校验内存中的页面和磁盘上的页面。如果发现磁盘上页面损坏，会将内存中的页面做一个基本信息校验刷盘，修复磁盘上页面。如果校验内存页面时发现页面不在内存中，会经内存接口读取磁盘上的页面。此过程中如果磁盘页面有问题，则会触发远程读自动修复功能。

    取值范围：bool，true和false。

  - is_segment

    是否是段页式表。根据local_bad_block_info中的bucket_node列值决定。如果bucket_node为-1时，表示不是段页式表，将is_segment设置为false；非-1的情况将is_segment设置为true。支持段页式存储。

    取值范围：bool，true和false。

  返回值类型：record

  示例（请依据local_bad_block_info的输出传参，否则报错）：

  页式存储：

```
gaussdb=# SELECT * FROM gs_verify_and_tryrepair_page('base/16552/24745',0,false,false);
node_name         |       path       |  blocknum  |         disk_page_res         | mem_page_res | is_repair
------------------+------------------+------------+-------------------------------+--------------+----------
dn_6001_6002_6003 | base/16552/24745 |     0      | page verification succeeded.|              | f
```

  段页式存储：

```
gaussdb=# SELECT * FROM gs_verify_and_tryrepair_page('base/14365/1',4494,false,true);
node_name         |     path     | blocknum |         disk_page_res         | mem_page_res | is_repair
------------------+--------------+----------+-------------------------------+--------------+-----------
dn_6001_6002_6003 | base/14365/1 |     4494 | page verification succeeded. |              | f
```


版权所有 © 华为云计算技术有限公司

- gs_repair_page(path text, blocknum oid, is_segment bool, timeout int)

  描述：修复本实例指定页面，仅支持有正常主备连接的主DN使用。默认只有在主DN节点上，使用初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以查看，其余用户需要赋权后才可以使用。

  注意：当DN实例上存在页面损坏时，进行升主会校验出错，报PANIC退出无法升主，为正常现象。

  参数说明

  - path

    损坏页面的路径。根据local_bad_block_info中file_path一列设置，或者是gs_verify_and_tryrepair_page函数中path一列设置。

    取值范围：字符串。

  - blocknum

    损坏页面的页面号。根据local_bad_block_info中block_num一列设置，或者是gs_verify_and_tryrepair_page函数中blocknum一列设置。

    取值范围：Oid，0 - 4294967295。注意：输入负值等数值都会被强制转成非负整数类型。

  - is_segment

    是否是段页式表。根据local_bad_block_info中的bucket_node列值决定，如果bucket_node为-1时，表示不是段页式表，将is_segment设置为false，非-1的情况将is_segment设置为true。支持段页式存储。

    取值范围：bool，true或者false。

  - timeout

    等待备DN回放的时长。修复页面需要等待备DN回放到当前主DN对应的位置，根据备DN回放所需时长设定。

    取值范围：60s - 3600s。

  返回值类型：bool

  示例（请根据local_bad_block_info的输出传参，否则报错）：

  页式存储：

  ```
  gaussdb=# SELECT * FROM gs_repair_page('base/16552/24745',0,false,60);
  result
  --------
  t
  ```

  段页式存储：

  ```
  gaussdb=# SELECT * FROM gs_repair_page('base/16552/1',4494,true,60);
  result
  --------
  t
  ```

- gs_seg_verify_datafile(IN tablespace_name name, IN database_name name, IN file_id integer, IN bucketnode integer, IN start_block_id bigint default 0, IN end_block_id bigint default UINT32MAX)

  描述：校验段页式1~5文件，main fork是否存在页面损坏，将校验结果记录到local_bad_block_info()函数中。只有具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户可以执行。

  参数：

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tablespace_name	name	表空间名称。
database_name	name	数据库名称。
file_id	integer	文件名称，1~5。
bucketnode	integer	表的bucketnode。 ● 0~1023表示hashbucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 当前仅支持hashbucket表与段页式普通表。
start_block_id	bigint	校验页面范围开始值，默认值：0。
end_block_id	bigint	校验页面范围结束值，默认值：4294967295。

返回值：无

示例：

```
gaussdb=# SELECT * FROM gs_seg_verify_datafile('seg_tblspc', 'postgres', 2, 1024);
WARNING:  page verification failed, calculated checksum 60994 but expected 11565, the block num is 4157
 gs_seg_verify_datafile
------------------------


(1 row)
```

- gs_edit_page_bypath(path text, blocknum int64, offset int, data text, data_size int, read_backup bool, storage_type text)

  描述：传入目标表文件的路径、块号、偏移量、修改的目标数据以及长度，将目标数据修改到页面对应字段中，其中，read_backup字段控制文件的读取方式，storage_type字段表示文件的存储方式（例如段页式存储），并返回修改后落盘的文件路径，为防止误修改操作，该函数不会直接对原页面进行修改，而是对复制页面进行修改，并将修改后的页面落盘到指定路径。必须是系统管理员或者运维模式下的运维管理员才能执行此函数，且仅适用于主节点。分布式环境下该函数需要在DN下执行。

  返回值类型：text。

版权所有 © 华为云计算技术有限公司

表 1-262 gs_edit_page_bypath 参数说明

参数类型	参数名	类型	描述
输入参数	path	text	待修改文件的物理文件路径，结合read_backup字段，既可以是数据库目录下文件的相对路径，也可以是备份等文件的绝对路径。若目标文件不存在或读取失败等，返回相应报错信息。 ● read_backup为false：path路径格式为tablespace name/database oid/表的relfilenode(物理文件名)。例如：base/16603/16394。 ● read_backup为true：path为合法路径，此时由于无法获取到输入文件的其他相关信息，所以需要用户保障输入数据的正确性。
输入参数	blocknum	bigint	指定修复页面的块号。 参数范围：0~MaxBlockNumber。 结合read_backup字段，读取指定物理/逻辑块号对应的页面，当指定块号超出范围等，返回相应报错信息。
输入参数	offset	int	修改字段的页内偏移。 参数范围：0<=offset<BLCKSZ。 当用户指定小于0或者大于等于BLCKSZ的值时，使用系统视图返回相应报错信息。
输入参数	data	text	修改的目标值类型。 类型： ● '0x'：表示十六进制。 ● '0b'：表示二进制。 ● '0s'：表示字符串。 其他：当data参数不为上述类型时仅限数据为十进制数字字符串。
输入参数	data_size	int	写入的数据长度，单位：字节。 参数范围：1 ~ 8。 当用户指定写入长度小于1或大于8字节，又或者offset+data_size > BLCKSZ时，使用系统视图返回相应报错信息。
输入参数	read_backup	bool	是否从备份目录中读取页面，当该字段为false时，将通过逻辑块号读取目标页面，否则基于物理块号读取页面。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输入参数	storage_type	text	文件的存储方式，可选参数： ● 'page'（页式）。 ● 'segment'（段页式）。 页式仅支持upage、ubtree数据页的编辑修改，段页式支持Astore数据页的编辑修改。不支持创建了表空间的表。由于无法获取到输入文件的其他相关信息，所以需要用户保障输入数据类型的正确性。
输出参数	output_msg	text	修改成功时，返回修改后文件落盘的绝对路径，修改后的文件存储于gs_log/dump目录下。若修改失败，则返回失败相关信息。

使用示例时请按参数说明传参并使用实际存在的物理路径。

示例1：在base/15808/25075表的0号页面偏移16字节处覆盖写入值为0X1FFF的数据。

```
gaussdb=# SELECT gs_edit_page_bypath('base/15808/25075',0,16,'0x1FFF', 2, false, 'page');
                        gs_edit_page_bypath
-------------------------------------------------------------------------
 /gs_log_dir/dump/1663_15808_25075_0.editpage
(1 rows)
```

示例2：当输入参数不符合规范时返回对应错误消息。

```
gaussdb=# SELECT gs_edit_page_bypath('base/15808/25075', 0,16,'@1231!', 8, false, 'page');
            gs_edit_page_bypath
-------------------------------------------
 Error: the parameter 'data' decode failed.
(1 row)
```

示例3：当需要写入的数据与原始值相同，返回告警信息。

```
gaussdb=# SELECT gs_edit_page_bypath('/gs_log_dir/dump/1663_15808_25075_0.editpage',
0,16,'0x1FFF', 2, true, 'page');
                   gs_edit_page_bypath
-----------------------------------------------------------
 Warning: source buffer is consistent with target buffer.
(1 row)
```

- gs_repair_page_bypath(src_path text, src_blkno int64, dest_path text, dest_blkno int64, storage_type text)

  描述：传入源文件路径以及页面号，将该页面覆盖写入到目标文件指定页面号上，支持通过主机修复备机页面，也支持基于备机修复主机页面，此外，该函数支持对坏块的初始化操作。分布式环境下该函数需要在DN下执行。

  a. 对目标页面进行覆盖写并同步备机，页式修改对象支持Uheap、Ubtree页面，后续支持Undo Record页面、Undo Slot页面、压缩表，以及Astore页面，段页式修改对象支持Astore页面，支持hashbucket。不支持系统表文件的修改，也不支持对数据区的修改。

  b. 功能支持将页面覆盖写到目标页面上。覆盖之前会将目标页面备份并落盘到指定目录，支持将备份页面重写回目标页面，在主机对普通表的修改会生成新的WAL日志并同步备机，在备机的修改不会记录WAL日志。

  c. 修复视图仅适用于主节点，或者开启备机读场景下的备节点。用户需要系统管理员或者运维模式下的运维管理员权限，所有修改均会记录数据库日志，并且，建议使用前开启系统函数的审计日志，便于记录审计信息。

版权所有 © 华为云计算技术有限公司

d. 在备机上调用修复视图时，仅支持从主机读取页面修复备机。

e. 修复时，源页面与目标页面的LSN必须一致，且页面CRC校验必须通过，否则修复将失败。在此过程中，可能触发其他修复方式。

f. 对目标页面进行校验。若目标页面为正确的页面，则不支持进行覆盖。当前校验支持Uheap、Ubtree页面。

返回值类型：text

⚠ 注意

调用本系统函数属于高危风险操作，请用户谨慎使用。

参数类型	参数名	类型	描述
输入参数	src_path	text	源文件的路径。支持的路径主要包含以下几类： ● 数据文件以及索引文件：gs_log/dump/1663_15808_25075_0.editpage。 ● 在主机端指定src_path为'standby'，即从备机端读取页面修复主机。 ● 在备机端可以指定src_path为'primary'，即从主机端读取页面修复备机，备机上仅支持‘primary’修复，并且使用前开启备机读。 ● 在主机端指定src_path为'init_block'，允许极端场景下跳过坏块。
输入参数	src_blkno	bigint	源页面的物理块号。 参数范围：0~MaxBlockNumber。
输入参数	dest_path	text	目标文件的相对路径。例如：base/15808/25075。
输入参数	dest_blkno	bigint	目标页面的逻辑块号。 参数范围：0~MaxBlockNumber。
输入参数	storage_type	text	目标文件的存储方式，可选参数： ● 'page'（页式）。 ● 'segment'（段页式）。
输出参数	output_msg	text	覆盖写成功时返回目标页面备份的路径，失败时返回报错信息。落盘文件名格式为relfilepath_blocknum_timestamp.repairpage。

版权所有 © 华为云计算技术有限公司

请按照实际情况根据上表传参并确认物理文件存在。传参有异常或修复失败时将报错。

示例1：输入指定路径下的文件，覆盖写入到目标文件中。

```
gaussdb=# SELECT * FROM gs_repair_page_bypath('gs_log/dump/1663_15991_16767_0.editpage', 0, 'base/15991/16767', 0, 'page');
                                    output_msg
-------------------------------------------------------------------------------------------------
/gs_log_dir/dump/1663_15991_16767_0_738039702421788.repairpage
(1 row)
```

示例2：从备机端读取页面修复主机。

```
gaussdb=# SELECT * FROM gs_repair_page_bypath('standby', 0, 'base/15990/16768', 0, 'page');
                                    output_msg
-------------------------------------------------------------------------------------------------
/gs_log_dir/dump/1663_15990_16768_0_738040397197907.repairpage
(1 row)
```

示例3：从主机端读取页面修复备机。

```
gaussdb=# SELECT * FROM gs_repair_page_bypath('primary', 0, 'base/15990/16768', 0, 'page');
                                    output_msg
-------------------------------------------------------------------------------------------------
 /gs_log_dir/dump/1663_15990_16768_0_738040506157799.repairpage
(1 row)
```

示例4：初始化目标页面，支持坏块跳过。

```
gaussdb=# SELECT * FROM gs_repair_page_bypath('init_block', 0, 'base/15990/16768', 0, 'page');
                                    output_msg
-------------------------------------------------------------------------------------------------
 /gs_log_dir/dump/1663_15990_16768_0_738040768010281.repairpage
(1 row)
```

- gs_edit_repair_page(path text, blocknum int64, area text, index int, name text, value uint64)

  描述：根据传入源文件路径及页面号，指定修改页面中的xid相关字段，并直接更新数据库中的页面。调用该函数修改页面会生成对应的Xlog。修改前，原始页面的源页面及解析页面将被dump至日志路径，同时函数返回dump文件的路径。

  该函数有如下约束：

  - 仅支持主DN调用。
  - 不支持修复压缩页面。
  - 不支持修复段页式页面。
  - 仅支持初始用户、具有sysadmin属性的用户以及在运维模式下具有运维管理员属性的用户调用。
  - 仅支持修复CRC校验正常的页面。

  返回值类型：text

  说明

  本系统函数作为应对xid损坏的兜底手段，属于高危风险操作，请用户谨慎使用，且不建议频繁调用。

版权所有 © 华为云计算技术有限公司

表 1-263 gs_edit_repair_page 参数说明

参数类型	参数名	参数类型	描述
输入参数	path	text	源文件的路径。表文件的相对路径可通过pg_relation_filepath(table_name text)查找；分区表的路径可查询pg_partition系统表并调用pg_partition_filepath(partition_oid)获得。例如：base/16603/16394。
输入参数	blocknum	bigint	源页面的物理块号，取值范围：0 ~ MaxBlockNumber。
输入参数	area	text	页面的目标区域，支持以下取值： ● header：页面头。 ● td：uheap页面的td或ted部分。 ● tuple：页面元组。
输入参数	index	integer	目标区域在同类区域中的下标，从0开始。其中header区域仅支持下标0。
输入参数	name	text	需要修复的目标字段。 ● header：pd_prune_xid、pd_xid_base、pd_multi_base ● td：xid ● tuple(uheap)：xid ● tuple(heap)：xmin、xmax ● tuple(ubtree)：xmin、xmax
输入参数	value	bigint	用于修复的目标值。
输出参数	output_msg	text	修复成功时返回dump文件路径；修复失败时返回错误信息。

请根据上表传参并确认物理文件存在。若参数异常或修复失败，函数将报错。

示例1：修改base/14816/16404文件第0个页面的header部分中的pd_prune_xid。

```
gaussdb=# select * from gs_edit_repair_page('base/14816/16404', 0, 'header', 0, 'pd_prune_xid', 0);
                                output_msg
---------------------------------------------------------------------------
/gs_log_dir/dump/1663_14816_16404_0_824366317606633.originpage+
/gs_log_dir/dump/1663_14816_16404_0.dump
(1 row)
```

示例2：修改base/14816/16404文件第3个页面的td部分中第0个ted的xid。

```
gaussdb=# select * from gs_edit_repair_page('base/14816/16404', 3, 'td', 0, 'xid', 196218);
                                output_msg
---------------------------------------------------------------------------
/gs_log_dir/dump/1663_14816_16404_3_824368882816559.originpage+
/gs_log_dir/dump/1663_14816_16404_3.dump
(1 row)
```


版权所有 © 华为云计算技术有限公司

示例3：修改base/14816/16387文件第5个页面的tuple部分中第16个tuple的xmax。

```
gaussdb=# select * from gs_edit_repair_page('base/14816/16387', 5, 'tuple', 16, 'xmax', 0);
                        output_msg
-----------------------------------------------------------------------
/gs_log_dir/dump/1663_14816_16387_5_824368882816559.originpage+
/gs_log_dir/dump/1663_14816_16387_5.dump
(1 row)
```

- gs_repair_undo_byzone(zone_id int)

  描述：传入待修复Undo Zone的zone_id，对目标Undo Zone的元信息进行修复，并返回修复结果的详细信息；如果没有进行修复则没有输出信息。

  返回值类型：record

  备注：当前函数仅支持在主节点进行调用，修复成功后会通过记录Xlog日志同步到备机，且调用者必须是系统管理员或者运维模式下的运维管理员，建议使用前开启系统函数的审计日志，便于记录审计信息。

⚠ 注意

调用本系统函数属于高危风险操作，请用户谨慎使用。

表 1-264 gs_repair_undo_byzone 参数说明

参数类型	参数名	类型	描述
输入参数	zone_id	int	Undo Zone编号： ● -1：修复所有Undo Zone的元信息。 ● 0~1048575：修复对应zone_id编号的Undo Zone的元信息。
输出参数	zone_id	int	Undo Zone编号。
输出参数	repair_detail	text	对应zone_id的Undo Zone元信息的修复结果，修复成功显示"rebuild undo meta succeed."；修复失败显示"rebuild undo meta failed."及其失败原因。

示例1：输入的zone_id对应的Undo Zone元信息没有损坏时，预期没有输出。

```
gaussdb=# SELECT * FROM gs_repair_undo_byzone(4);
 zone_id | repair_detail
---------+---------------
(0 rows)
```

示例2：输入的zone_id对应的Undo Zone元信息修复成功时，显示修复成功的信息。

```
gaussdb=# SELECT * FROM gs_repair_undo_byzone(78);
 zone_id | repair_detail
---------+---------------
      78 | rebuild undo meta succeed.
(1 row)
```

示例3：输入的zone_id对应的Undo Zone元信息修复失败时，显示修复失败的详细信息。

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM gs_repair_undo_byzone(0);
 zone_id | repair_detail
---------+---------------
       0 | rebuild undo meta failed. try lock undo zone_id failed.
(1 row)
```

说明

如果待修复的 Undo Zone 已损坏且 zone_id 已经被其他活跃线程占用时，调用该修复函数会将占用 zone_id 的线程结束并强制修复已损坏的 Undo Zone 元信息。

- gs_verify_urq(index_oid oid, partindex_oid oid, blocknum bigint, queue_type text)

  描述：校验索引回收队列（潜在队列/可用队列/单页面）的正确性。

  参数说明：如表1-265所示。

  返回值类型：record

表 1-265 gs_verify_urq 参数说明

参数类型	参数名	类型	描述
输入参数	index_oid	oid	UBTree索引oid： ● 普通索引：索引oid。 ● 全局索引：GPI oid。 ● local索引：主索引oid。
输入参数	partindex_oid	oid	UBTree分区索引oid： ● 普通索引：0。 ● 全局索引：0。 ● local索引：分区索引oid（一级/二级）。
输入参数	blocknum	bigint	页面号： ● 队列类型为single page时，校验单个页面blocknum的所有元组正确性。取值范围为[0，队列文件大小/8192）。 ● 队列类型为empty queue或free queue时，blocknum是一个无效值。
输入参数	queue_type	text	队列类型： ● empty queue：潜在队列。 ● free queue：可用队列。 ● single page：队列单页面。
输出参数	error_code	text	错误码。
输出参数	detail	text	具体报错及其他关键信息。

使用示例时请根据参数说明传参，使用实际存在的oid和blocknum，否则将报错。

版权所有 © 华为云计算技术有限公司

示例1：

```
gaussdb=# SELECT * FROM gs_verify_urq(16387, 0, 1, 'free queue');
 error_code | detail
------------+--------
(0 rows)
```

示例2：

```
gaussdb=# SELECT * FROMgs_verify_urq(16387, 0, 1, 'empty queue');
      error_code       |
detail
-----------------------
+-----------------------------------------------------------------------------------------------------------------------
 VERIFY_URQ_PAGE_ERROR | invalid urq meta: oid 16387, blkno 1, head_blkno = 1, tail_blkno = 3,
nblocks_upper = 4294967295, nblocks_lower = 1; urq_blocks = 6, index_blocks = 12
(1 row)
```

说明

该接口当前仅支持USTORE索引表。如果索引回收队列校验正常，则该视图不输出错误码和报错详细信息，否则输出错误码和报错详细信息，错误码包含"VERIFY_URQ_PAGE_ERROR"、"VERIFY_URQ_LINK_ERROR"、"VERIFY_URQ_HEAD_MISSED_ERROR"和"VERIFY_URQ_TAIL_MISSED_ERROR"，如出现以上错误码，请联系华为工程师辅助定位。

- gs_urq_dump_stat(index_oid oid, partindex_oid oid)

  描述：查询指定索引回收队列相关信息。

  显示信息：recentGlobalDataXmin和globalFrozenXid是回收队列判断索引页面是否可以被回收时使用的两个oldestxmin，next_xid为下一个最新的事务xid，urq_blocks为回收队列总页面数以及free queue（可用队列）、empty queue（潜在队列）有效页面里的相关信息。对于PCR版本的索引，输出的recentGlobalDataXmin会被替换成影响PCR索引页面回收的globalRecycleXid。

  参数说明：如表1-266所示。

  表 1-266 gs_urq_dump_stat 参数说明

参数类型	参数名	类型	描述
输入参数	index_oid	oid	UBTree索引oid： • 普通索引：索引oid。 • 全局索引：GPI oid。 • local索引：主索引oid。
输入参数	partindex_oid	oid	UBTree分区索引oid： • 普通索引：0。 • 全局索引：0。 • local索引：分区索引oid（一级/二级）。
输出参数	result	text	索引回收队列的详细统计信息。

  使用示例时请根据实际情况按参数说明传参，使用实际存在的oid，否则将报错。

  示例（RCR）：

```
gaussdb=# SELECT * FROM gs_urq_dump_stat(16387, 0);
                                                result
```


版权所有 © 华为云计算技术有限公司

```
-----------------------------------------------------------------------------------------------------------------------------------
 urq stat info: recentGlobalDataXmin = 213156, globalFrozenXid = 213156, next_xid = 214157,
urq_blocks = 6,                    +
   free  queue:    head page blkno = 0 min_xid = 211187 max_xid = 214157, tail page blkno = 0
min_xid = 211187 max_xid = 214157,+
               middle page min_xid = 1152921504606846975 max_xid = 0, valid_pages = 1, valid_items =
6, can_use_item = 3       +
   empty queue:    head page blkno = 1 min_xid = 212160 max_xid = 213160, tail page blkno = 3
min_xid = 213162 max_xid = 214156,+
               middle page min_xid = 1152921504606846975 max_xid = 0, valid_pages = 2, valid_items =
999, can_use_item = 498    +

(1 row)
```

示例（PCR）：

```
gaussdb=# SELECT gs_urq_dump_stat(17260,0);
                                                    gs_urq_dump_stat
-----------------------------------------------------------------------------------------------------------------------------------
-------------------------
 urq stat info: globalRecycleXid = 22113, globalFrozenXid = 22107, next_xid = 22116, urq_blocks =
6,                                 +
   free  queue:    head page blkno = 0 min_xid = 1152921504606846975 max_xid = 0, tail page blkno
= 0 min_xid = 1152921504606846975 max_xid = 0,+
               middle page min_xid = 1152921504606846975 max_xid = 0, valid_pages = 1, valid_items =
0, can_use_item = 0                +
   empty queue:    head page blkno = 1 min_xid = 1152921504606846975 max_xid = 0, tail page blkno
= 1 min_xid = 1152921504606846975 max_xid = 0,+
               middle page min_xid = 1152921504606846975 max_xid = 0, valid_pages = 1, valid_items =
0, can_use_item = 0                +

(1 row)
```

说明

该接口当前仅支持USTORE索引表。

- gs_repair_urq(index_oid oid, partindex_oid oid)

  描述：重建（有损）索引回收队列（潜在队列和可用队列）。删除当前索引的回收队列文件，重新创建一个空的回收队列文件。重建成功显示reinitial the recycle queue of index relation successfully。

  参数说明：如**表1-267**所示。

  备注：当前函数仅支持在主节点进行调用。

表 1-267 gs_repair_urq 参数说明

参数类型	参数名	类型	描述
输入参数	index_oid	oid	UBTree索引oid： • 普通索引：索引oid。 • 全局索引：GPI oid。 • local索引：主索引oid。
输入参数	partindex_oid	oid	UBTree分区索引oid： • 普通索引：0。 • 全局索引：0。 • local索引：分区索引oid（一级/二级）。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输出参数	result	text	重建成功显示reinitial the recycle queue of index relation successfully，否则为失败。

使用示例时请根据实际情况按参数说明传参，使用实际存在的oid，否则将报错。

示例：

```
gaussdb=# SELECT * FROM gs_repair_urq(16387, 0);
                                result
-----------------------------------------------------------------
 reinitial the recycle queue of index relation successfully.
(1 row)
```

说明

该接口当前仅支持USTORE索引表。

- gs_check_table(table_oid oid, partition_oid oid, index_oid oid, partindex_oid oid, start_blk bigint, end_blk bigint, duration text, need_repair bool, mode text)

  描述：校验堆表和索引的反向数据一致性。

  约束：支持Astore、Ustore存储引擎，不支持MOT；支持备节点进行校验，不支持logger节点；堆表支持普通行存表、ILM压缩表、透明加密表、Hashbucket表的校验，不支持临时表、全局临时表、unlogged表。索引支持B-tree、UB-tree索引的校验，兼容段页式存储。

  参数说明：详见**表1-268**。

  返回值类型：record

表 1-268 gs_check_table 参数说明

参数类型	参数名	类型	描述
输入参数	table_oid	oid	堆表oid： • 普通表：堆表 oid。 • 分区表：主表 oid。
输入参数	partition_oid	oid	分区oid： • 普通表：0。 • 分区表：分区 oid。 • 分区表扫描全分区：0。
输入参数	index_oid	oid	索引oid： • 普通索引：索引 oid。 • 全局索引：GPI oid。 • local索引：主索引oid。 • 扫描全部索引：0。

版权所有 © 华为云计算技术有限公司

参数类型	参数名	类型	描述
输入参数	partindex_oid	oid	分区索引oid： ● 普通索引：0。 ● 全局索引：0。 ● local索引：分区索引oid(一级/二级)。 ● local索引扫描全部分区：0。
输入参数	start_blk	bigint	校验开始页面号。和end_blk同时取值-1的场景下默认从头开始校验。分区表扫描全分区时，仅支持从头开始校验，start_blk和end_blk需同时取值为-1。
输入参数	end_blk	bigint	校验结束页面号。
输入参数	duration	text	校验时长，支持设置毫秒、秒、分钟、小时、天，例如“500ms”、“60s”，“10min”，“2h”，“1d”最长可设置时长为24h，可能存在毫秒级、秒级误差或分钟级误差，设置为0则默认校验完成结束。校验时长为实际执行时间和设置时间的最小值，执行超过设置时间会中止并记录当前校验位置，下次扫描支持从当前位置断点续作。
输入参数	need_repair	bool	是否需要自动修复不一致，即按照堆表来增加或删除索引元组，当前仅支持设置为false。预埋参数，本版本暂不支持自动修复能力。
输入参数	mode	text	校验模式预埋参数，取值范围包含{”increment”，”full”}。支持断点续做、增量（”increment”）校验，全量（”full”）校验等能力。
输出参数	error_code	int	错误码。
输出参数	output_filepath	text	解析后落盘相对路径。路径为gs_log/dump目录下，落盘文件名为dbverify_错误码_堆表oid_时间戳.log，例如： /$GAUSSLOG/gs_log/dump/dbverify_0_16384_123456.log

使用示例时请根据参数说明传参，使用实际存在且匹配的oid和blocknum，否则将报错。

示例1：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# select * from gs_check_table((select oid from pg_class where relname='t2'),0,0,0,-1,-1, '1h', false, 'full');
 verify_code |                                output_filepath
-------------+-------------------------------------------------------------------------------------
           0 | /$GAUSSLOG/gs_log/dump/dbverify_0_17279_766073191583799.log
(1 row)
```

示例2：

```
gaussdb=# select * from gs_check_table((select oid from pg_class where relname='t1'),0,(select oid from pg_class where relname='index_t1'),0,2,10, '2s', false, 'full');
 verify_code |                                output_filepath
-------------+-------------------------------------------------------------------------------------
           0 | /$GAUSSLOG/gs_log/dump/dbverify_0_17168_766073338594117.log
(1 row)
```

说明

调用本系统函数属于高危风险操作，请用户谨慎使用。

该接口当前支持Astore、Ustore表和B-tree、UB-tree索引的一致性校验。如果校验完成后未发现一致性问题，错误码为"0”；如出现非"0"错误码，请联系华为工程师辅助定位。

关于断点续做和增量校验：

断点续做和增量校验的相关信息不会进行持久化，数据库进程重启后之前的校验会丢失，重新校验时会从头开始进行全量校验。

校验模式为“increment”且上一次对该表和索引的校验未完成时会触发断点续做，从上次校验中断的位置开始校验。

校验模式为“increment”且上一次对该表和索引的校验已完成时会触发增量校验，校验增量修改页面。

校验模式为“full”时，从头开始进行全量校验。

关于IO流控：

默认设置io_priority为低优先级进行IO流控。

关于使用权限：

执行校验系统函数需要有初始用户、系统管理员、运维管理员或监控管理员权限，只能校验当前数据库的数据表或索引。

推荐使用场景：

校验推荐使用在业务低峰期，尽量保证校验期间大部分数据为静态数据，保证校验的完整性和准确性。

- gs_check_index(index_oid oid, partindex_oid oid, duration text, need_repair bool, mode text)

  描述：校验堆表和索引的正向数据一致性。

  约束：支持Astore、Ustore存储引擎，不支持MOT；支持备节点进行校验，不支持logger节点；堆表支持普通行存表、ILM压缩表、透明加密表、Hashbucket表的校验，不支持临时表、全局临时表、unlogged表。索引支持B-tree、UB-tree索引的校验，兼容段页式存储。

  参数说明：详见**表1-269**。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

表 1-269 gs_check_index 参数说明

参数类型	参数名	类型	描述
输入参数	index_oid	oid	索引oid： ● 普通索引：索引 oid。 ● 全局索引：GPI oid。 ● local索引：主索引oid。
输入参数	partindex_oid	oid	分区索引oid： ● 普通索引：0。 ● 全局索引：0。 ● local索引：分区索引oid(一级/二级)。 ● local索引扫描全部分区：0。
输入参数	duration	text	校验时长，支持设置毫秒、秒、分钟、小时、天，例如“500ms”“60s”，“10min”，“2h”，“1d”最长可设置时长为24h，可能存在毫秒级、秒级误差或分钟级误差，设置为0则默认校验完成结束。校验时长为实际执行时间和设置时间的最小值，执行超过设置时间会中止并记录当前校验位置，下次扫描支持从当前位置断点续作。
输入参数	need_repair	bool	是否需要自动修复不一致，即按照堆表来增加或删除索引元组，当前仅支持设置为false。预埋参数，本版本暂不支持自动修复能力。
输入参数	mode	text	校验模式预埋参数，取值范围包含{”increment”，”full”}。支持断点续做、增量（”increment”）校验，全量（”full”）校验等能力。
输出参数	error_code	int	错误码。
输出参数	output_filepath	text	解析后落盘相对路径。路径为gs_log/dump目录下，落盘文件名为dbverify_错误码_堆表oid_时间戳.log，例如： /$GAUSSLOG/gs_log/dump/dbverify_0_16384_123456.log

使用示例时请根据参数说明传参，使用实际存在且匹配的oid，否则将报错。

示例1：

```
gaussdb=# select * from gs_check_index((select oid from pg_class where relname='index_t1'),0, '1h', false, 'full');
```


版权所有 © 华为云计算技术有限公司

```
 verify_code |                                output_filepath
-------------+-------------------------------------------------------------------------------------
           0 | /$GAUSSLOG/gaussdb/single_node/gs_log/dump/dbverify_0_17302_766074761373434.log
(1 row)
```

示例2：

```
gaussdb=# select * from gs_check_index((select oid from pg_class where relname='index_range_2'),
(select oid from pg_partition where parentid=(select oid from pg_class where
relname='index_range_2')and relname='p1_a_idx'), '10s', false, 'full');
 verify_code |                                output_filepath
-------------+-------------------------------------------------------------------------------------
           0 | /$GAUSSLOG/gaussdb/single_node/gs_log/dump/dbverify_0_17306_766074834795344.log
(1 row)
```

说明

该接口当前支持Astore、Ustore表和B-tree、UB-tree索引的一致性校验。如果校验完成后未发现一致性问题，错误码为"0”；如出现非"0"错误码，请联系华为工程师辅助定位。

关于断点续做和增量校验：

断点续做和增量校验的相关信息不会进行持久化，数据库进程重启后之前的校验会丢失，重新校验时会从头开始进行全量校验。

校验模式为“increment”且上一次对该表和索引的校验未完成时会触发断点续做，从上次校验中断的位置开始校验。

校验模式为“increment”且上一次对该表和索引的校验已完成时会触发增量校验，校验增量修改页面。

校验模式为“full”时，从头开始进行全量校验。

关于IO流控：

默认设置io_priority为低优先级进行IO流控。

关于使用权限：

执行校验系统函数需要有初始用户、系统管理员、运维管理员或监控管理员权限，只能校验当前数据库的数据表或索引。

推荐使用场景：

校验推荐使用在业务低峰期，尽量保证校验期间大部分数据为静态数据，保证校验的完整性和准确性。

注意

调用本系统函数属于高危风险操作，请用户谨慎使用。

- gs_get_standby_bad_block_info()

  描述：显示备机上已经检测到但是还未修复的页面。默认只有在备DN节点上，使用初始用户、具有sysadmin权限的用户以及在运维模式下具有运维管理员权限的用户、以及监控用户可以查看，其余用户需要赋权后才可以使用。返回值invalid_type列共有4种类型：NOT_PRESENT（页面不存在）、NOT_INITIALIZED（页面初始化失败）、 LSN_CHECK_ERROR（LSN校验失败）、CRC_CHECK_ERROR（CRC校验失败）。

  返回值类型：record

  示例（若不存在已检测到但未修复的页面则输出0行）：

```
gaussdb=# SELECT * FROM gs_get_standby_bad_block_info();
 spc_node | db_node | rel_node | bucket_node | fork_num | block_num |  invalid_type   |
master_page_lsn
----------+---------+----------+-------------+----------+-----------+-----------------+-----------------
   1663   |   16552 |   24745  |      -1     |    0     |     0     | CRC_CHECK_ERROR | 0/B2009E8
(1 rows)
```


版权所有 © 华为云计算技术有限公司

1.6.43 Multixact 回收函数

- gs_vacuum_multixact()

  描述：回收pg_multixact目录下的offsets和members中的数据文件。

  返回值类型：Boolean

  - 成功回收文件时返回true。
  - 无可回收文件时返回false。

  该函数有如下约束：

  - 本函数仅支持在主DN上执行，或通过EXECUTE DIRECT方式在对应主DN执行回收。
  - 仅初始用户或具有sysadmin属性的用户可以调用。
  - 不支持在CN上执行手动回收。

  示例1：无可回收文件返回false。

```
gaussdb=# SELECT gs_vacuum_multixact();
 gs_vacuum_multixact
---------------------
 f
(1 row)
```

  示例2：回收文件成功返回true。

```
gaussdb=# SELECT gs_vacuum_multixact();
 gs_vacuum_multixact
---------------------
 t
(1 row)
```

  说明

  是否可回收的逻辑基于PG_DATABASE系统表的datminmxid计算，某些场景可能导致datminmxid不推进，如：当前库存在全局临时表。由于全局临时表的数据具有会话级隔离特性，VACUUM后台线程无法访问全局临时表的信息，导致该全局临时表的relminmxid字段无法自动推进，最终影响datminmxid无法推进。因此，需要在全局临时表的会话上对所有全局临时表执行“vacuum freeze 表名;”命令，以推进当前全局临时表的relminmxid。

1.6.44 XML 类型函数

说明

- 以下所有XML类型函数中，当GUC参数xmloption为content时，XML声明中encoding值可以取为ZHS16GBK；当GUC参数xmloption为document时，XML声明中encoding值不可以取为ZHS16GBK，若设置encoding = ZHS16GBK，则会发生报错。
- xml类型函数支持的最大XML嵌套层数不能超过256层。

- xmlparse ({ DOCUMENT | CONTENT } value [wellformed])

  描述：使用函数xmlparse，从字符数据产生XML类型的值。

  参数：数据类型为text。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT XMLPARSE (DOCUMENT '<?xml version="1.0"?><book><title>Manual</
title><chapter>...</chapter></book>');
                  xmlparse
-----------------------------------------------------------
 <book><title>Manual</title><chapter>...</chapter></book>
(1 row)
gaussdb=# SELECT XMLPARSE (CONTENT 'abc<foo>bar</foo><bar>foo</bar>');
```

 版权所有 © 华为云计算技术有限公司

```
        xmlparse
---------------------------------
 abc<foo>bar</foo><bar>foo</bar>
(1 row)
gaussdb=# SELECT XMLPARSE (CONTENT 'abc<foo>bar</foo' wellformed);
    xmlparse
------------------
 abc<foo>bar</foo
(1 row)
```

- xmlserialize({ DOCUMENT | CONTENT } value AS type)

  描述：使用函数xmlserialize，从XML产生一个字符串。

  参数：数据类型为xml。

  返回值类型：数据类型可以为character、character varying或text（或其中某个的变种）。

  示例：

```
gaussdb=# SELECT xmlserialize(CONTENT 'good' AS CHAR(10));
 xmlserialize
--------------
 good
(1 row)
gaussdb=# SELECT xmlserialize(DOCUMENT '<head>bad</head>' as text);
  xmlserialize
------------------
 <head>bad</head>
(1 row)
```

  说明

  当一个字符串值在没有通过xmlparse或xmlserialize的情况下，与xml类型进行转换时，具体选择DOCUMENT或CONTENT由“XML OPTION”会话配置参数决定，这个配置参数可以由标准命令来设置：

```
SET XML OPTION { DOCUMENT | CONTENT };
```

  或使用类似的语法来设置：

```
SET xmloption TO { DOCUMENT | CONTENT };
```

- xmlcomment(text)

  描述：创建一个XML值，并且它包含一个用指定文本作为内容的XML注释。该文本不包含“--”字符且结尾不存在“-”字符，符合XML注释的格式要求。当参数为空时，结果也为空。

  参数：数据类型为text。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT xmlcomment('hello');
  xmlcomment
--------------
 <!--hello-->
```

- xmlconcat(xml[, ...])

  描述：将由单个XML值组成的列表串接成一个单独的值，该值包含一个XML的内容片段。其中空值会被忽略，并且只有当所有参数都为空时结果才为空。在兼容ORA数据库模式下，设置a_format_version值为10c和a_format_dev_version值为s2，增加了校验输入片段是否为非良构XML文本。

  参数：数据类型为XML。

  返回值类型：XML

  示例1：

```
gaussdb=# SET xmloption=content;
SET
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT XMLCONCAT(('<?xml version="1.0" encoding="GB2312" standalone="no"?
><bar>foo</bar>'),('<?xml version="1.0" encoding="GB2312" standalone="no" ?><bar>foo</bar>')) ;
                          xmlconcat
---------------------------------------------------------------------
<?xml version="1.0" standalone="no"?><bar>foo</bar><bar>foo</bar>
(1 row)
gaussdb=# SELECT XMLCONCAT('abc>');
 xmlconcat
-----------
 abc>
(1 row)
```

示例2：兼容ORA数据库的语法示例。

```
gaussdb=# CREATE DATABASE gaussdb_ora DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c gaussdb_ora
gaussdb_ora=# SET a_format_version='10c';
SET
gaussdb_ora=# SET a_format_dev_version=s2;
SET
gaussdb_ora=# SET xmloption=content;
SET
gaussdb_ora=# SELECT XMLCONCAT(('<?xml version="1.0" encoding="GB2312" standalone="no"?
><bar>foo</bar>'),('<?xml version="1.0" encoding="GB2312" standalone="no" ?><bar>foo</bar>')) ;
                          xmlconcat
---------------------------------------------------------------------
 <?xml version="1.0" standalone="no"?><bar>foo</bar><bar>foo</bar>
(1 row)

gaussdb_ora=# SELECT XMLCONCAT('abc>');
ERROR:  invalid XML document
DETAIL:  line 1: Start tag expected, '<' not found
abc>
^
CONTEXT:  referenced column: xmlconcat
gaussdb_ora=# \c postgres
gaussdb=# DROP DATABASE gaussdb_ora;
DROP DATABASE
```

说明

在兼容ORA数据库模式下，设置a_format_version值为10c和a_format_dev_version值为s2时，若XML声明中encoding属性值为ZHS16GBK，则XMLCONCAT函数将会发生报错。

- xmlagg(xml [order_by_clause])

  描述：该函数是一个聚集函数，它将聚集函数调用的输入值串接起来，且支持跨行串接，order_by_clause详见**SELECT**。在兼容ORA数据库模式下，设置a_format_version值为10c和a_format_dev_version值为s2，数据库xmloption参数默认为content，当xmloption设置为document时，使用换行符串接多行XML。若XML声明中encoding属性值不为默认编码UTF-8时，聚集结果有XML声明。

  参数：XML

  返回值类型：XML

  示例1：

```
gaussdb=# CREATE TABLE xmltest (
           id int,
           data xml
          );
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'id' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
gaussdb=# INSERT INTO xmltest VALUES (1, '<value>one</value>');
INSERT 0 1
gaussdb=# INSERT INTO xmltest VALUES (2, '<value>two</value>');
INSERT 0 1
gaussdb=# SELECT xmlagg(data) FROM xmltest;
              xmlagg
```


版权所有 © 华为云计算技术有限公司

```
---------------------------------------
 <value>one</value><value>two</value>
(1 row)
```

示例2：为兼容ORA数据库的语法示例。

```
gaussdb=# SET xmloption=document;
SET
gaussdb=# SELECT xmlagg(data) FROM xmltest;
      xmlagg
--------------------
 <value>one</value>+
 <value>two</value>
(1 row)
gaussdb=# DELETE FROM XMLTEST;
DELETE 2
gaussdb=# INSERT INTO xmltest VALUES (1, '<?xml version="1.0" encoding="GBK"?><value>one</
value>');
INSERT 0 1
gaussdb=# INSERT INTO xmltest VALUES (2, '<?xml version="1.0" encoding="GBK"?><value>two</
value>');
INSERT 0 1
gaussdb=# SELECT xmlagg(data) FROM xmltest;
                      xmlagg
-------------------------------------------------------
 <?xml version="1.0" encoding="GBK"?><value>one</value>+
 <value>two</value>
(1 row)
gaussdb=# SELECT xmlagg(data order by id desc) FROM xmltest;
                      xmlagg
-------------------------------------------------------
 <?xml version="1.0" encoding="GBK"?><value>two</value>+
 <value>one</value>
(1 row)

gaussdb=# DROP TABLE xmltest;
```

说明

xmlagg函数的输入与返回类型均为XML类型。由于单个XML数据类型的大小上限为1GB，因此xmlagg不支持聚合结果超过1GB 的场景。

- xmlelement([ENTITYESCAPING | NOENTITYESCAPING] { [NAME] element_name | EVALNAME element_name } [, xmlattributes([ENTITYESCAPING | NOENTITYESCAPING] value [[AS] attname | AS EVALNAME attname] [, ...])] [, content [[AS] alias] [, ...]])

  描述：使用给定的名称、属性和内容产生一个XML元素。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT xmlelement(name foo);
 xmlelement
------------
 <foo/>

--在ORA兼容模式下。
gaussdb=# CREATE DATABASE gaussdb_ora DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c gaussdb_ora
gaussdb_ora=# SET a_format_version='10c';
SET
gaussdb_ora=# SET a_format_dev_version=s2;
SET
--xmlelement中默认不设置或者设置ENTITYESCAPING关键字时，xmlelement的内容中的保留字符将会被转义。
gaussdb_ora=# SELECT xmlelement("entityescaping<>", 'a$><&"b');
                      xmlelement
```


版权所有 © 华为云计算技术有限公司

```
-------------------------------------------------------------
 <entityescaping<>>a$&gt;&lt;&amp;&quot;b</entityescaping<>>
(1 row)

gaussdb_ora=# SELECT xmlelement(entityescaping "entityescaping<>", 'a$><&"b');
                      xmlelement
-------------------------------------------------------------
 <entityescaping<>>a$&gt;&lt;&amp;&quot;b</entityescaping<>>
(1 row)

--xmlelement中设置NOENTITYESCAPING关键字时，xmlelement的内容中的保留字符将不会被转义。
gaussdb_ora=# SELECT xmlelement(noentityescaping "entityescaping<>", 'a$><&"b');
                xmlelement
-------------------------------------------
 <entityescaping<>>a$><&"b</entityescaping<>>
(1 row)

--xmlelement中对内容使用[as] alias声明别名时，内容值类型必须为xml类型。
gaussdb_ora=# SELECT xmlelement("entityescaping<>", '<abc/>' b);
ERROR:  argument of XMLELEMENT must be type xml, not type unknown
LINE 1: SELECT xmlelement("entityescaping<>", '<abc/>' b);
                                              ^
CONTEXT:  referenced column: xmlelement

gaussdb_ora=# SELECT xmlelement("entityescaping<>", '<abc/>' as b);
ERROR:  argument of XMLELEMENT must be type xml, not type unknown
LINE 1: SELECT xmlelement("entityescaping<>", '<abc/>' as b);
                                              ^
CONTEXT:  referenced column: xmlelement

gaussdb_ora=# SELECT xmlelement("entityescaping<>", xml('<abc/>') b);
               xmlelement
-----------------------------------------
 <entityescaping<>><abc/></entityescaping<>>
(1 row)

gaussdb_ora=# SELECT xmlelement("entityescaping<>", xml('<abc/>') as b);
               xmlelement
-----------------------------------------
 <entityescaping<>><abc/></entityescaping<>>
(1 row)

--xmlattributes中默认不设置或者设置ENTITYESCAPING关键字时，xmlattributes的内容中的保留字符将会
被转义。
gaussdb_ora=# SELECT xmlelement("entityescaping<>", xmlattributes('entityescaping<>'
"entityescaping<>"));
                       xmlelement
-------------------------------------------------------------
 <entityescaping<> entityescaping<>="entityescaping&lt;&gt;"/>
(1 row)

gaussdb_ora=# SELECT xmlelement(name "entityescaping<>", xmlattributes(entityescaping
'entityescaping<>' "entityescaping<>"));
                       xmlelement
-------------------------------------------------------------
 <entityescaping<> entityescaping<>="entityescaping&lt;&gt;"/>
(1 row)

--xmlattributes中设置NOENTITYESCAPING关键字时，xmlattributes的内容中的保留字符将不会被转义。
gaussdb_ora=# SELECT xmlelement("entityescaping<>", xmlattributes(noentityescaping
'entityescaping<>' "entityescaping<>"));
                     xmlelement
-----------------------------------------------------
 <entityescaping<> entityescaping<>="entityescaping<>"/>
(1 row)

gaussdb_ora=# \c postgres
gaussdb=# DROP DATABASE gaussdb_ora;
DROP DATABASE
```


版权所有 © 华为云计算技术有限公司

说明

- xmlelement和xmlattributes的name字段赋NULL时，行为与O不一致。xmlelement的name字段赋NULL时，结果显示name信息为空，且不显示属性信息。xmlattributes的name字段赋NULL时，不显示属性信息。
- 设置如下两个参数后，xmlelement的内容转义规则为ORA兼容，未设置时xmlelement的内容转义规则为PG兼容。

```
SET a_format_version='10c';
SET a_format_dev_version=s2;
```

- xmlforest(content [AS name] [, ...])

  描述：使用给定名称和内容产生一个元素的XML序列。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT xmlforest('abc' AS foo, 123 AS bar);
          xmlforest
------------------------------
 <foo>abc</foo><bar>123</bar>
```

- xmlpi(name target [, content])

  描述：创建一个XML处理指令。若内容不为空，则内容不能包含字符序列。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT xmlpi(name php, 'echo "hello world";');
            xmlpi
-----------------------------
 <?php echo "hello world";?>
```

- xmlroot(xml, version text | no value [, standalone yes|no|no value])

  描述：修改一个XML值的根节点的属性。如果指定了一个版本，它会替换根节点的版本声明中的值；如果指定了一个独立设置，它会替换根节点的独立声明中的值。

  示例：

```
gaussdb=# SELECT xmlroot('<?xml version="1.1"?><content>abc</content>',version '1.0', standalone
yes);
                        xmlroot
----------------------------------------------------------------
 <?xml version="1.0" standalone="yes"?><content>abc</content>
(1 row)
```

- xmlexists(text passing [BY REF] xml [BY REF])

  描述：评估一个XPath 1.0表达式(第一个参数)，以传递的XML值作为其上下文项。 如果评估的结果是一个空节点集，该函数返回false；如果结果是任何其他值，则返回true。 如果任何参数为空，则函数返回null。 作为上下文项传递的非空值必须是一个XML文档，而不是内容片段或任何非XML值。

  参数：XML

  返回值类型：BOOLEAN

  示例：

```
gaussdb=# SELECT xmlexists('//town[text() = ''Toronto'']' PASSING BY REF '<towns><town>Toronto</
town><town>Ottawa</town></towns>');
 xmlexists
------------
 t
(1 row)
```

- xml_is_well_formed(text)

  描述：检查text是不是正确的XML类型格式、返回值为布尔类型。

  参数：text

版权所有 © 华为云计算技术有限公司

返回值类型：BOOLEAN

示例：

```
gaussdb=# SELECT xml_is_well_formed('<>');
 xml_is_well_formed
--------------------
 f
(1 row)
```

- xml_is_well_formed_document(text)

  描述：检查text是否为良构的XML document类型格式，返回值为布尔类型。

  参数：text

  返回值类型：BOOLEAN

  示例：

```
gaussdb=# SELECT xml_is_well_formed_document('<pg:foo xmlns:pg="http://postgresql.org/
stuff">bar</pg:foo>');
 xml_is_well_formed_document
-----------------------------
 t
(1 row)
```

- xml_is_well_formed_content(text)

  描述：检查text是否为良构的XML content类型格式，返回值为布尔类型。

  参数：text

  返回值类型：BOOLEAN

  示例：

```
gaussdb=# SELECT xml_is_well_formed_content('k');
 xml_is_well_formed_content
----------------------------
 t
(1 row)
```

- xpath(xpath, xml [, nsarray])

  描述：在XML类型的数据xml上，计算XPath 1.0表达式（如：xpath (a text value)）。它返回一个XML值的数组，该数组对应于该XPath表达式产生的节点集合。如果该XPath表达式返回一个标量值而不是一个节点集合，将会返回一个单一元素的数组。

  第二个参数必须是一个良构的XML文档。它必须有一个单一根节点元素。

  该函数可选的第三个参数是一个名字空间映射的数组。这个数组应该是一个二维text数组，其第二轴长度等于2（即应该是一个数组的数组，其中每一个都刚好由2个元素组成）。每个数组项的第一个元素是名字空间的名称（别名），第二个元素是名字空间的URI。并不要求在这个数组中提供的别名和在XML文档本身中使用的那些名字空间相同（即在XML文档中和在xpath函数环境中，别名都是本地的）。

  返回值类型：xml[]

  示例：

```
gaussdb=# SELECT xpath('/my:a/text()', '<my:a xmlns:my="http://example.com">test</
my:a>',ARRAY[ARRAY['my', 'http://example.com']]);
 xpath
--------
 {test}
(1 row)
```

- xpath_exists(xpath, xml [, nsarray])

  描述：该函数是xpath函数的一种特殊形式。这个函数不是返回满足XPath 1.0表达式的单一XML值，它返回一个布尔值表示查询是否被满足(它是否产生了空节点集以外的任何值)。这个函数等价于标准的XMLEXISTS谓词，不过它还提供了对一个名字空间映射参数的支持。

版权所有 © 华为云计算技术有限公司

返回值类型：BOOLEAN

示例：

```
gaussdb=# SELECT xpath_exists('/my:a/text()', '<my:a xmlns:my="http://example.com">test</
my:a>',ARRAY[ARRAY['my', 'http://example.com']]);
 xpath_exists
--------------
 t
(1 row)
```

说明

以下XML类型函数示例中，需进行前置数据准备，如下所示：

```
gaussdb=# CREATE SCHEMA testxmlschema;
CREATE SCHEMA
gaussdb=# CREATE TABLE testxmlschema.test1 (a int, b text);
CREATE TABLE
gaussdb=# INSERT INTO testxmlschema.test1 VALUES (1, 'one'), (2, 'two'), (-1, null);
INSERT 0 3
gaussdb=# CREATE DATABASE test;
CREATE DATABASE

--示例执行结束后，可使用如下命令删除上述前置数据。
gaussdb=# DROP DATABASE test;
DROP DATABASE
gaussdb=# DROP TABLE testxmlschema.test1;
DROP TABLE
gaussdb=# DROP SCHEMA testxmlschema;
DROP SCHEMA
```

- query_to_xml(query text, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将query查询的内容映射成XML模式文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT query_to_xml('SELECT * FROM testxmlschema.test1', false, false, '');
                        query_to_xml
-------------------------------------------------------------
 <table xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">+
                                                             +
 <row>                                                          +
   <a>1</a>                                                      +
   <b>one</b>                                                      +
 </row>                                                         +
                                                             +
 <row>                                                          +
   <a>2</a>                                                      +
   <b>two</b>                                                      +
 </row>                                                         +
                                                             +
 <row>                                                          +
   <a>-1</a>                                                     +
 </row>                                                         +
                                                             +
 </table>                                                       +
(1 row)
```

- query_to_xmlschema(query text, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将query查询的内容映射成XML文档和XML模式文档，并把两个文档连接在一起。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT query_to_xmlschema('SELECT * FROM testxmlschema.test1', false, false, '');
                                  query_to_xmlschema
-----------------------------------------------------------------------------------------------
```


版权所有 © 华为云计算技术有限公司

```
<xsd:schema                                                                 +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">                                  +
                                                                            +
<xsd:simpleType name="INTEGER">                                                    +
  <xsd:restriction base="xsd:int">                                                 +
    <xsd:maxInclusive value="2147483647"/>                                         +
    <xsd:minInclusive value="-2147483648"/>                                        +
  </xsd:restriction>                                                            +
</xsd:simpleType>                                                                 +
                                                                            +
<xsd:simpleType name="UDT.regression.pg_catalog.text">                                 +
  <xsd:restriction base="xsd:string">                                              +
  </xsd:restriction>                                                            +
</xsd:simpleType>                                                                 +
                                                                            +
<xsd:complexType name="RowType">                                                   +
  <xsd:sequence>                                                                +
    <xsd:element name="a" type="INTEGER" minOccurs="0"></xsd:element>                    +
    <xsd:element name="b" type="UDT.regression.pg_catalog.text" minOccurs="0"></xsd:element>+
  </xsd:sequence>                                                               +
</xsd:complexType>                                                                +
                                                                            +
<xsd:complexType name="TableType">                                                 +
  <xsd:sequence>                                                                +
    <xsd:element name="row" type="RowType" minOccurs="0" maxOccurs="unbounded"/>          +
  </xsd:sequence>                                                               +
</xsd:complexType>                                                                +
                                                                            +
<xsd:element name="table" type="TableType"/>                                       +
                                                                            +
</xsd:schema>
(1 row)
```

- query_to_xml_and_xmlschema(query text, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将query查询的内容映射成XML文档和XML模式文档，并把两个文档连接在一起。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT query_to_xml_and_xmlschema('SELECT * FROM testxmlschema.test1', true, true, '');
                                query_to_xml_and_xmlschema
-------------------------------------------------------------------------------------------
<xsd:schema                                                                 +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">                                  +
                                                                            +
<xsd:simpleType name="INTEGER">                                                    +
  <xsd:restriction base="xsd:int">                                                 +
    <xsd:maxInclusive value="2147483647"/>                                         +
    <xsd:minInclusive value="-2147483648"/>                                        +
  </xsd:restriction>                                                            +
</xsd:simpleType>                                                                 +
                                                                            +
<xsd:simpleType name="UDT.regression.pg_catalog.text">                                 +
  <xsd:restriction base="xsd:string">                                              +
  </xsd:restriction>                                                            +
</xsd:simpleType>                                                                 +
                                                                            +
<xsd:complexType name="RowType">                                                   +
  <xsd:sequence>                                                                +
    <xsd:element name="a" type="INTEGER" nillable="true"></xsd:element>                  +
    <xsd:element name="b" type="UDT.regression.pg_catalog.text" nillable="true"></xsd:element>+
  </xsd:sequence>                                                               +
</xsd:complexType>                                                                +
                                                                            +
<xsd:element name="row" type="RowType"/>                                           +
                                                                            +
</xsd:schema>                                                                     +
```


版权所有 © 华为云计算技术有限公司

```
                                                                +
<row xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">                          +
                                                                +
  <a>1</a>                                                        +
  <b>one</b>                                                       +
</row>                                                           +
                                                                +
<row xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">                          +
                                                                +
  <a>2</a>                                                        +
  <b>two</b>                                                       +
</row>                                                           +
                                                                +
<row xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">                          +
                                                                +
  <a>-1</a>                                                       +
  <b xsi:nil="true"/>                                              +
</row>                                                           +
                                                                +
(1 row)
```

- cursor_to_xml(cursor refcursor, count int, nulls boolean,tableforest boolean, targetns text)

  描述：该函数会将游标查询的内容映射成XML文档。

  返回值类型：XML

  示例：

```
gaussdb=# CURSOR xc WITH HOLD FOR SELECT * FROM testxmlschema.test1 ORDER BY 1, 2;
DECLARE CURSOR
gaussdb=# SELECT cursor_to_xml('xc'::refcursor, 5, false, true, '');
                    cursor_to_xml
-------------------------------------------------------------
 <row xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">+
                                                +
  <a>-1</a>                                        +
 </row>                                           +
                                                +
 <row xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">+
                                                +
  <a>1</a>                                         +
  <b>one</b>                                        +
 </row>                                           +
                                                +
 <row xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">+
                                                +
  <a>2</a>                                         +
  <b>two</b>                                        +
 </row>                                           +
                                                +


(1 row)
```

- cursor_to_xmlschema(cursor refcursor, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将游标查询的内容映射成XML模式文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT cursor_to_xmlschema('xc'::refcursor, true, false, '');
                                cursor_to_xmlschema
-----------------------------------------------------------------------------------------------
 <xsd:schema                                                                    +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">                                       +
                                                                +
 <xsd:simpleType name="INTEGER">                                          +
  <xsd:restriction base="xsd:int">                                      +
   <xsd:maxInclusive value="2147483647"/>                                  +
```


版权所有 © 华为云计算技术有限公司

```
  <xsd:minInclusive value="-2147483648"/>                                    +
 </xsd:restriction>                                                        +
</xsd:simpleType>                                                          +
                                                                          +
<xsd:simpleType name="UDT.regression.pg_catalog.text">                              +
 <xsd:restriction base="xsd:string">                                      +
 </xsd:restriction>                                                       +
</xsd:simpleType>                                                          +
                                                                          +
<xsd:complexType name="RowType">                                                +
 <xsd:sequence>                                                           +
  <xsd:element name="a" type="INTEGER" nillable="true"></xsd:element>              +
  <xsd:element name="b" type="UDT.regression.pg_catalog.text" nillable="true"></xsd:element>+
 </xsd:sequence>                                                          +
</xsd:complexType>                                                         +
                                                                          +
<xsd:complexType name="TableType">                                              +
 <xsd:sequence>                                                           +
  <xsd:element name="row" type="RowType" minOccurs="0" maxOccurs="unbounded"/>       +
 </xsd:sequence>                                                          +
</xsd:complexType>                                                         +
                                                                          +
<xsd:element name="table" type="TableType"/>                                    +
                                                                          +
</xsd:schema>
(1 row)
```

- schema_to_xml(schema name, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将整个模式的内容映射成XML文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT schema_to_xml('testxmlschema', false, true, '');
                        schema_to_xml
-----------------------------------------------------------------------
 <testxmlschema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">+
                                                                    +
 <test1>                                                            +
                                                                    +
  <a>1</a>                                                          +
  <b>one</b>                                                        +
 </test1>                                                           +
                                                                    +
 <test1>                                                            +
                                                                    +
  <a>2</a>                                                          +
  <b>two</b>                                                        +
 </test1>                                                           +
                                                                    +
 <test1>                                                            +
                                                                    +
  <a>-1</a>                                                         +
 </test1>                                                           +
                                                                    +
                                                                    +
 </testxmlschema>                                                         +
(1 row)
```

- schema_to_xmlschema(schema name, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将整个模式的内容映射成XML模式文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT schema_to_xmlschema('testxmlschema', false, true, '');
                                schema_to_xmlschema
```


版权所有 © 华为云计算技术有限公司

```
------------------------------------------------------------------------------------------------------------
 <xsd:schema                                                                         +
     xmlns:xsd="http://www.w3.org/2001/XMLSchema">                                   +
                                                                                     +
 <xsd:simpleType name="INTEGER">                                                     +
   <xsd:restriction base="xsd:int">                                                  +
     <xsd:maxInclusive value="2147483647"/>                                          +
     <xsd:minInclusive value="-2147483648"/>                                         +
   </xsd:restriction>                                                                +
 </xsd:simpleType>                                                                   +
                                                                                     +
 <xsd:simpleType name="UDT.t1.pg_catalog.text">                                      +
   <xsd:restriction base="xsd:string">                                               +
   </xsd:restriction>                                                                +
 </xsd:simpleType>                                                                   +
                                                                                     +
 <xsd:complexType name="SchemaType.t1.testxmlschema">                                +
   <xsd:sequence>                                                                    +
     <xsd:element name="test1" type="RowType.t1.testxmlschema.test1" minOccurs="0" 
maxOccurs="unbounded"/>+
   </xsd:sequence>                                                                   +
 </xsd:complexType>                                                                  +
                                                                                     +
 <xsd:element name="testxmlschema" type="SchemaType.t1.testxmlschema"/>
+
                                                                                     +
 </xsd:schema>
(1 row)
```

- schema_to_xml_and_xmlschema(schema name, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将整个模式的内容映射成XML文档和XML模式文档，并把两个文档连接在一起。

  返回值类型：XML

  示例：

```
 gaussdb=# SELECT schema_to_xml_and_xmlschema('testxmlschema', true, true, 'foo');
                                     schema_to_xml_and_xmlschema
------------------------------------------------------------------------------------------------------------
 <testxmlschema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="foo" 
xsi:schemaLocation="foo #">+
                                                                                     +
                                                                   +
 <xsd:schema                                                                         +
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"                                    +
     targetNamespace="foo"                                                           +
     elementFormDefault="qualified">                                                 +
                                                                                     +
 <xsd:simpleType name="INTEGER">                                                     +
   <xsd:restriction base="xsd:int">                                                  +
     <xsd:maxInclusive value="2147483647"/>                                          +
     <xsd:minInclusive value="-2147483648"/>                                         +
   </xsd:restriction>                                                                +
 </xsd:simpleType>                                                                   +
                                                                                     +
 <xsd:simpleType name="UDT.t1.pg_catalog.text">                                      +
   <xsd:restriction base="xsd:string">                                               +
   </xsd:restriction>                                                                +
 </xsd:simpleType>                                                                   +
                                                                                     +
 <xsd:complexType name="SchemaType.t1.testxmlschema">                                +
   <xsd:sequence>                                                                    +
     <xsd:element name="test1" type="RowType.t1.testxmlschema.test1" minOccurs="0" 
maxOccurs="unbounded"/>   +
   </xsd:sequence>                                                                   +
 </xsd:complexType>                                                                  +
                                                                                     +
 <xsd:element name="testxmlschema"
```

```
type="SchemaType.t1.testxmlschema"/>                                    +
                                                                        +
</xsd:schema>                                                             +
                                                                        +
<test1>                                                                   +
                                                                        +
  <a>1</a>                                                                +
  <b>one</b>                                                              +
</test1>                                                                  +
                                                                        +
<test1>                                                                   +
                                                                        +
  <a>2</a>                                                                +
  <b>two</b>                                                              +
</test1>                                                                  +
                                                                        +
<test1>                                                                   +
                                                                        +
  <a>-1</a>                                                               +
  <b xsi:nil="true"/>                                                     +
</test1>                                                                  +
                                                                        +
                                                                        +
</testxmlschema>                                                            +

(1 row)
```

- database_to_xml(nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将整个数据库的内容映射成XML文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT database_to_xml(true, true, 'test');
                            database_to_xml
-------------------------------------------------------------------------------
 <test xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="test">+
                                                                        +
 <dbe_x005F_xml>                                                          +
                                                                        +
 </dbe_x005F_xml>                                                         +
                                                                        +
 <dbe_x005F_xmldom>                                                         +
                                                                        +
 </dbe_x005F_xmldom>                                                        +
                                                                        +
 <dbe_x005F_xmlparser>                                                      +
                                                                        +
 </dbe_x005F_xmlparser>                                                     +
                                                                        +
 <public>                                                                 +
                                                                        +
 </public>                                                                +
                                                                        +
 </test>                                                                  +

(1 row)
```

- database_to_xmlschema(nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将整个数据库的内容映射成XML模式文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT database_to_xmlschema(true, true, 'test');
                              database_to_xmlschema
-------------------------------------------------------------------------------------
 <xsd:schema                                                                    +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"                                  +
    targetNamespace="test"                                                      +
```


版权所有 © 华为云计算技术有限公司

```
   elementFormDefault="qualified">                                                +
                                                                                  +
<xsd:complexType name="CatalogType.test">                                         +
  <xsd:all>                                                                       +
    <xsd:element name="dbe_x005F_xml" type="SchemaType.test.dbe_x005F_xml"/>      +
    <xsd:element name="dbe_x005F_xmldom" type="SchemaType.test.dbe_x005F_xmldom"/> +
    <xsd:element name="dbe_x005F_xmlparser" type="SchemaType.test.dbe_x005F_xmlparser"/>+
    <xsd:element name="public" type="SchemaType.test.public"/>                    +
  </xsd:all>                                                                      +
</xsd:complexType>                                                                +
                                                                                  +
<xsd:element name="test" type="CatalogType.test"/>                                +
                                                                                  +
</xsd:schema>
(1 row)
```

- database_to_xml_and_xmlschema(nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将整个数据库的内容映射成XML文档和XML模式文档，并把两个文档连接在一起。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT database_to_xml_and_xmlschema(true, true, 'test');
                                database_to_xml_and_xmlschema
-------------------------------------------------------------------------------------------------
 <test xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="test"
xsi:schemaLocation="test #">+
                                                                                  +
<xsd:schema                                                                       +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"                                  +
    targetNamespace="test"                                                        +
    elementFormDefault="qualified">                                               +
                                                                                  +
<xsd:complexType name="CatalogType.test">                                         +
  <xsd:all>                                                                       +
    <xsd:element name="dbe_x005F_xml" type="SchemaType.test.dbe_x005F_xml"/>      +
    <xsd:element name="dbe_x005F_xmldom"
type="SchemaType.test.dbe_x005F_xmldom"/>                                         +
    <xsd:element name="dbe_x005F_xmlparser"
type="SchemaType.test.dbe_x005F_xmlparser"/>                                      +
    <xsd:element name="public" type="SchemaType.test.public"/>                    +
  </xsd:all>                                                                      +
</xsd:complexType>                                                                +
                                                                                  +
<xsd:element name="test" type="CatalogType.test"/>                                +
                                                                                  +
</xsd:schema>                                                                     +
                                                                                  +
<dbe_x005F_xml>                                                                   +
                                                                                  +
</dbe_x005F_xml>                                                                  +
                                                                                  +
<dbe_x005F_xmldom>                                                                +
                                                                                  +
</dbe_x005F_xmldom>                                                               +
                                                                                  +
<dbe_x005F_xmlparser>                                                             +
                                                                                  +
</dbe_x005F_xmlparser>                                                            +
                                                                                  +
<public>                                                                          +
                                                                                  +
</public>                                                                         +
                                                                                  +
</test>                                                                           +

(1 row)
```

 版权所有 © 华为云计算技术有限公司

- table_to_xml(tbl regclass, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将关系表的内容映射成XML文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT table_to_xml('testxmlschema.test1', false, false, '');
                      table_to_xml
-------------------------------------------------------------
 <test1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">+
                                                             +
 <row>                                                       +
   <a>1</a>                                                   +
   <b>one</b>                                                  +
 </row>                                                      +
                                                             +
 <row>                                                       +
   <a>2</a>                                                   +
   <b>two</b>                                                  +
 </row>                                                      +
                                                             +
 <row>                                                       +
   <a>-1</a>                                                  +
 </row>                                                      +
                                                             +
 </test1>                                                    +

(1 row)
```

- table_to_xmlschema(tbl regclass, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将关系表的内容映射成XML模式文档。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT table_to_xmlschema('testxmlschema.test1', false, false, '');
                                        table_to_xmlschema
-----------------------------------------------------------------------------------------------------------------
 <xsd:schema                                                                                                    +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">                                                                +
                                                                                                                 +
 <xsd:simpleType name="INTEGER">                                                                                 +
   <xsd:restriction base="xsd:int">                                                                              +
     <xsd:maxInclusive value="2147483647"/>                                                                      +
     <xsd:minInclusive value="-2147483648"/>                                                                     +
   </xsd:restriction>                                                                                            +
 </xsd:simpleType>                                                                                               +
                                                                                                                 +
 <xsd:simpleType name="UDT.regression.pg_catalog.text">                                                          +
   <xsd:restriction base="xsd:string">                                                                           +
   </xsd:restriction>                                                                                            +
 </xsd:simpleType>                                                                                               +
                                                                                                                 +
 <xsd:complexType name="RowType.regression.testxmlschema.test1">
+
   <xsd:sequence>                                                                                                +
     <xsd:element name="a" type="INTEGER" minOccurs="0"></
xsd:element>                                                                                                     +
     <xsd:element name="b" type="UDT.regression.pg_catalog.text" minOccurs="0"></
xsd:element>                                                                                                     +
   </xsd:sequence>                                                                                               +
 </xsd:complexType>                                                                                              +
                                                                                                                 +
 <xsd:complexType name="TableType.regression.testxmlschema.test1">
+
   <xsd:sequence>                                                                                                +
     <xsd:element name="row" type="RowType.regression.testxmlschema.test1" minOccurs="0"
maxOccurs="unbounded"/>+
```


版权所有 © 华为云计算技术有限公司

```
  </xsd:sequence>                                                                +
 </xsd:complexType>                                                                +
                                                                                  +
 <xsd:element name="test1"
type="TableType.regression.testxmlschema.test1"/>                                 +
                                                                                  +
 </xsd:schema>
(1 row)
```

- table_to_xml_and_xmlschema(tbl regclass, nulls boolean, tableforest boolean, targetns text)

  描述：该函数会将关系表的内容映射成XML文档和XML模式文档，并把两个文档连接在一起。

  返回值类型：XML

  示例：

```
gaussdb=# SELECT table_to_xml_and_xmlschema('testxmlschema.test1', false, false, '');
                                    table_to_xml_and_xmlschema
-----------------------------------------------------------------------------------------------------------------
 <test1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="#">                +
                                                                                  +
 <xsd:schema                                                                       +
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">                                                 +
                                                                                  +
 <xsd:simpleType name="INTEGER">                                                                  +
  <xsd:restriction base="xsd:int">                                                               +
    <xsd:maxInclusive value="2147483647"/>                                                        +
    <xsd:minInclusive value="-2147483648"/>                                                       +
  </xsd:restriction>                                                              +
 </xsd:simpleType>                                                                +
                                                                                  +
 <xsd:simpleType name="UDT.regression.pg_catalog.text">                                                    +
  <xsd:restriction base="xsd:string">                                                            +
  </xsd:restriction>                                                              +
 </xsd:simpleType>                                                                +
                                                                                  +
 <xsd:complexType name="RowType.regression.testxmlschema.test1">
+
  <xsd:sequence>                                                                  +
    <xsd:element name="a" type="INTEGER" minOccurs="0"></
xsd:element>                                    +
    <xsd:element name="b" type="UDT.regression.pg_catalog.text" minOccurs="0"></
xsd:element>                  +
  </xsd:sequence>                                                                 +
 </xsd:complexType>                                                               +
                                                                                  +
 <xsd:complexType name="TableType.regression.testxmlschema.test1">
+
  <xsd:sequence>                                                                  +
    <xsd:element name="row" type="RowType.regression.testxmlschema.test1" minOccurs="0"
maxOccurs="unbounded"/>+
  </xsd:sequence>                                                                 +
 </xsd:complexType>                                                               +
                                                                                  +
 <xsd:element name="test1"
type="TableType.regression.testxmlschema.test1"/>                                 +
                                                                                  +
 </xsd:schema>                                                                    +
                                                                                  +
 <row>                                                                           +
  <a>1</a>                                                                        +
  <b>one</b>                                                                      +
 </row>                                                                          +
                                                                                  +
 <row>                                                                           +
  <a>2</a>                                                                        +
  <b>two</b>                                                                      +
```


版权所有 © 华为云计算技术有限公司

```
</row>                                                                  +
                                                                      +
<row>                                                                   +
  <a>-1</a>                                                              +
</row>                                                                  +
                                                                      +
</test1>                                                                +

(1 row)
```

说明

- xpath相关函数仅支持xpath()和xpath_exists()，由于其使用xpath语言查询XML文档，而这些函数都依赖于libxml2库，且这个库仅在Xpath1.0提供，所以对XPath的限制为1.0。
- 不支持xquery、xml extension、xslt功能。

- getclobval(xml)

  描述：将XML类型转化成CLOB类型。该函数在参数a_format_version值为10c和a_format_dev_version值为s4的情况下有效。

  参数：入参为XML类型。

  返回值类型：CLOB

  示例：

```
gaussdb=# CREATE DATABASE gaussdb_ora DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c gaussdb_ora
gaussdb_ora=# SET a_format_version='10c';
SET
gaussdb_ora=# SET a_format_dev_version='s4';
SET
gaussdb_ora=# declare
 xmldata xml;
 result clob;
begin
 xmldata := '<a>123</a>';
 result := getclobval(xmldata);
 RAISE NOTICE 'result is : %',result;
END;
/
NOTICE:  result is : <a>123</a>
gaussdb_ora=# SELECT getclobval(xmlparse(document '<a>123</a>'));
 getclobval
------------
 <a>123</a>
(1 row)

gaussdb_ora=# \c postgres
gaussdb=# DROP DATABASE gaussdb_ora;
DROP DATABASE
```

- getstringval(xml)

  描述：该函数将XML类型转化为字符串。此函数在参数a_format_version值为10c和a_format_dev_version值为s4的情况下有效。

  参数：入参为XML类型。

  返回值类型：VARCHAR2

  示例：

```
gaussdb=# CREATE DATABASE gaussdb_ora DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c gaussdb_ora
gaussdb_ora=# SET a_format_version='10c';
SET
gaussdb_ora=# SET a_format_dev_version='s4';
```


版权所有 © 华为云计算技术有限公司

```
SET
gaussdb_ora=# declare
 xmldata xml;
 result varchar2;
begin
 xmldata := '<a>123<b>456</b></a>';
 result := getstringval(xmldata);
 RAISE NOTICE 'result is : %',result;
END;
/
NOTICE:  result is : <a>123<b>456</b></a>
gaussdb_ora=# select getstringval(xmlparse(document '<a>123<b>456</b></a>'));
    getstringval
---------------------
 <a>123<b>456</b></a>
(1 row)

gaussdb_ora=# \c postgres
gaussdb=# DROP DATABASE gaussdb_ora;
DROP DATABASE
```

- xmlsequence(xml)

  描述：此函数的功能是将一个XML类型的参数转换为一个XMLTYPE类型的数组，每个数组元素都是XMLTYPE对象。这个函数的输入参数不能为空，也必须是一个有效的XML文档。如果输入参数不符合要求，函数会返回空值或者抛出异常。这个函数可以用于处理XML文档中的多个子节点，或者将XML文档分割为多个片段。

  参数：XML类型。

  返回值类型：xmltype类型的数组。

  示例：如果想要将这个文档转换为一个包含三个元素的数组，每个元素都是一个book节点，可以使用以下语句。

```
gaussdb=# SELECT xmlsequence(xml('<books><book><title>The Catcher in the Rye</
title><author>J.D. Salinger</author><year>1951</year></book><book><title>1984</
title><author>George Orwell</author><year>1949</year></book><book><title>The Hitchhiker''s
Guide to the Galaxy</title><author>Douglas Adams</author><year>1979</year></book></books>'));
                        xmlsequence
------------------------------------------------------------
{"<books>                                          +
    <book>                                        +
      <title>The Catcher in the Rye</title>            +
      <author>J.D. Salinger</author>                  +
      <year>1951</year>                              +
    </book>                                       +
    <book>                                        +
      <title>1984</title>                            +
      <author>George Orwell</author>                    +
      <year>1949</year>                              +
    </book>                                       +
    <book>                                        +
      <title>The Hitchhiker's Guide to the Galaxy</title>+
      <author>Douglas Adams</author>                    +
      <year>1979</year>                              +
    </book>                                       +
  </books>"}
(1 row)
```

  说明

  当XML文档中含有双引号时，单独查看xmlsequence函数，结果中会包含双引号的转义符，使用xmlsequence函数结果时不受影响。

- xmlcast(value_expression as datatype)

  描述：对于XML类型的输入数据，在提取文档中的文本内容后，将其转换为指定的目标数据类型。

 版权所有 © 华为云计算技术有限公司

参数：value_expression，表示XML类型的文档内容。

返回值类型：指定的目标数据类型。

示例：

```
-- XML类型数据转NUMBER类型。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
CREATE DATABASE
gaussdb=# \c ora_db
ora_db=# SELECT XMLCAST(XML('<number_value>123.45</number_value>') AS NUMBER) AS
number_value FROM dual;
 number_value
--------------
       123.45
(1 row)
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
DROP DATABASE
```

说明

- 仅在ORA兼容模式下支持使用该函数。
- 劣构XML文档仅支持value_expression表达式返回值为多根节点的XML。
- 不支持转换的目标数据类型包括BYTEAWITHOUTORDERWITHEQUALCOL、BYTEAWITHOUTORDERCOL、_BYTEAWITHOUTORDERWITHEQUALCOL、_BYTEAWITHOUTORDERCOL以及部分伪类型。
- 升级观察期不支持使用该函数。

1.6.45 XMLTYPE 类型函数

说明

xmltype类型函数支持的最大XML嵌套层数不能超过256层。

- createxml(varchar2[,varchar2,numeric,numeric])

  描述：varchar2类型入参，静态方法创建xmltype类型。

  参数：第一个参数要转换成xmltype的字符串（必传字段），第二个参数为用于使输入符合给定架构的可选架构URL（可选字段，默认为空，暂不生效），第三个参数为指示实例根据给定的XML架构有效的标志（可选字段，默认为0，暂不生效），第四个参数为是否为良构的标识（可选字段，默认为0，暂不生效）。

  返回值类型：xmltype

  示例：

```
gaussdb=# SELECT createxml('<a>123</a>');
 createxml
--------------
 <a>123</a>
(1 row)
```

 版权所有 © 华为云计算技术有限公司

说明

- 与Oracle数据库差异：在PL/SQL中，createxml允许入参为空串，并返回NULL。
- 字符串encoding只支持UTF-8、GBK、ZHS16GBK、LATIN1~LATIN10，version字段只支持1.x。
- createxml支持以xmltype.createxml()语法方式调用。

  示例：
  ```
  gaussdb=# SELECT xmltype.createxml('<a>123</a>');
   createxml
  ------------
   <a>123</a>
  (1 row)
  ```
- 本章内入参为xmltype的函数支持以xmltype().func()的方式调用，会将前一项返回的xmltype类型当作入参传入后一项的函数内，该语法支持多层嵌套（用户自定义函数入参为xmltype不支持该语法）。

  示例：
  ```
  gaussdb=# SELECT xmltype('<a>123<b>456</b></a>').extract('/a/b').getstringval();
   xmltypefunc
  -------------
   <b>456</b>
  (1 row)
  ```

  上述用例实际效果与函数嵌套一致。
  ```
  gaussdb=# SELECT getstringval(extractxml(xmltype('<a>123<b>456</b></a>'),'/a/b'));
   getstringval
  --------------
   <b>456</b>
  (1 row)
  ```
- 存储过程内支持xmltype类型的变量以a.func()方式调用函数，该语法支持一层嵌套。

  示例：
  ```
  gaussdb=# declare
          a xmltype;
          b varchar2;
  begin
          a:=xmltype('<a>123<b>456</b></a>');
          b:=a.getstringval();
          RAISE NOTICE 'xmltype_str is : %',b;
  end;
  /
  NOTICE:  xmltype_str is : <a>123<b>456</b></a>
  ```

- createxml(clob [,varchar2,numeric ,numeric])

  描述：clob类型入参，静态方法创建xmltype类型。

  参数：第一个参数要转换成xmltype的clob对象（必传字段），第二个参数为用于使输入符合给定架构的可选架构URL（可选字段，默认为空，暂不生效），第三个参数为实例根据给定的XML架构有效的标志（可选字段，默认为0，暂不生效），第四个参数为是否为良构的标识（可选字段，默认为0，暂不生效）。

  返回值类型：xmltype

  示例：
  ```
  gaussdb=# declare
   xmltype_clob clob;
   xmltype_obj xmltype;
   xmltype_str varchar2(1000);
  begin
   xmltype_clob := '<a>123</a>';
   xmltype_obj := createxml(xmltype_clob);
   xmltype_str := xmltype_obj.getstringval();
   RAISE NOTICE 'xmltype_str is : %',xmltype_str;
  end;
  /
  NOTICE:  xmltype_str is : <a>123</a>
  ```


版权所有 © 华为云计算技术有限公司

说明

clob类型参数入参最大支持1GB-1。

- createxml(blob,numeric[,varchar2,numeric,numeric])

  描述：blob类型入参，静态方法创建xmltype类型。

  参数：第一个参数要转换成xmltype的blob对象（必传字段），第二个参数为输入xml数据的字符集id（必传字段），第三个参数为用于使输入符合给定架构的可选架构URL（可选字段，默认为空，暂不生效），第四个参数为实例根据给定的XML架构有效的标志（可选字段，默认为0，暂不生效），第五个参数为是否为良构的标识（可选字段，默认为0，暂不生效）。

  返回值类型：xmltype

  示例：

```
gaussdb=# declare
 xmltype_blob blob;
 xmltype_obj xmltype;
 xmltype_str varchar2(1000);
begin
 xmltype_blob := xmltype('<a>123</a>').getblobval(7);
 xmltype_obj := createxml(xmltype_blob,7);
 xmltype_str := xmltype_obj.getstringval();
 RAISE NOTICE 'xmltype_str is : %',xmltype_str;
end;
/
NOTICE:  xmltype_str is : <?xml version="1.0" encoding="UTF8"?>
<a>123</a>
```

  说明

  - blob类型参数入参最大支持256MB-1。
  - 字符集id取值范围为1~43。

- getblobval(xmltype,numeric)

  描述：将xmltype类型转化成blob类型，支持xmltype().func()方式调用。

  参数：第一个参数为xmltype类型，第二个参数为要转换的目标字符集的字符集id。

  返回值类型：blob

  示例：

```
gaussdb=# SELECT getblobval(xmltype('<asd/>'),7);
                                   getblobval
--------------------------------------------------------------------------------

3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255544638223F3E0A3C6173
642F3E
(1 row)
```

  xmltype().func()方式：

```
gaussdb=# SELECT xmltype('<asd/>').getblobVal(7);
                                   xmltypefunc
--------------------------------------------------------------------------------

3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255544638223F3E0A3C6173
642F3E
(1 row)
```

  说明

  入参xmltype长度最大256MB-1。

- getclobval(xmltype)

  描述：将xmltype类型转化成clob类型，支持xmltype().func()方式调用。

版权所有 © 华为云计算技术有限公司

参数：xmltype类型。

返回值类型：clob

示例：

```
gaussdb=# SELECT getclobval(xmltype('<a>123</a>'));
 getclobval
--------------
 <a>123</a>
(1 row)
```

xmltype().func()方式：

```
gaussdb=# SELECT xmltype('<a>123</a>').getclobval();
 xmltypefunc
--------------
 <a>123</a>
(1 row)
```

- getnumberval(xmltype)

  描述：将xmltype类型转化成numeric类型，支持xmltype().func()方式调用。

  参数：入参为xmltype类型。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SELECT getnumberval(xmltype('<a>123</a>').extract('/a/text()'));
   getnumberval
  --------------
           123
  (1 row)
  ```

  xmltype().func()方式：

  ```
  gaussdb=# SELECT xmltype('<a>123</a>').extract('/a/text()').getnumberval();
   xmltypefunc
  --------------
           123
  (1 row)
  ```

- isfragment(xmltype)

  描述：返回该xmltype类型是片段（1）还是文档（0），支持xmltype().func()方式调用。

  参数：入参为xmltype类型。

  返回值类型：numeric

  示例：

  ```
  gaussdb=# SELECT isfragment(xmltype('<a>123</a>'));
   isfragment
  --------------
            0
  (1 row)
  ```

  xmltype().func()方式：

  ```
  gaussdb=# SELECT xmltype('<a>123</a>').isfragment();
   xmltypefunc
  --------------
            0
  (1 row)
  ```

- xmltype(varchar2[,varchar2,numeric,numeric])

  描述：varchar2类型创建xmltype类型。

  参数：第一个参数要转换成xmltype的字符串（必传字段），第二个参数为用于使输入符合给定架构的可选架构URL（可选字段，默认为空，暂不生效），第三个参数为指示实例根据给定的XML架构有效的标志（可选字段，默认为0，暂不生效），第四个参数为是否为良构的标识（可选字段，默认为0，暂不生效）。

  返回值类型：xmltype

 版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT xmltype('<a>123</a>');
 xmltype
--------------
 <a>123</a>
(1 row)
```

说明

- 与Oracle数据库差异：在PL/SQL中，xmltype允许入参为空串，并返回NULL。
- 字符串encoding只支持UTF-8、GBK、ZHS16GBK、LATIN1~LATIN10，version字段只支持1.x。

- xmltype(clob[,varchar2,numeric,numeric])

  描述：clob类型创建xmltype类型。

  参数：第一个参数要转换成xmltype的clob对象（必传字段），第二个参数为用于使输入符合给定架构的可选架构URL（可选字段，默认为空，暂不生效），第三个参数为实例根据给定的XML架构有效的标志（可选字段，默认为0，暂不生效），第四个参数为是否为良构的标识（可选字段，默认为0，暂不生效）。

  返回值类型：xmltype

  示例：

```
gaussdb=# declare
 xmltype_clob clob;
 xmltype_obj xmltype;
 xmltype_str varchar2(1000);
begin
 xmltype_clob := '<a>123</a>';
 xmltype_obj := xmltype(xmltype_clob);
 xmltype_str := xmltype_obj.getstringval();
 RAISE NOTICE 'xmltype_str is : %',xmltype_str;
end;
/
NOTICE:  xmltype_str is : <a>123</a>
```

  说明

  clob类型参数入参最大支持1GB-1。

- xmltype(blob, numeric [,varchar2,numeric ,numeric])

  描述：blob类型创建xmltype类型。

  参数：第一个参数要转换成xmltype的blob对象（必传字段），第二个参数为输入xml数据的字符集id，第三个参数为用于使输入符合给定架构的可选架构URL（可选字段，默认为空，暂不生效），第四个参数为实例根据给定的XML架构有效的标志（可选字段，默认为0，暂不生效），第五个参数为是否为良构的标识（可选字段，默认为0，暂不生效）。

  返回值类型：xmltype

  示例：

```
gaussdb=# declare
 xmltype_blob blob;
 xmltype_obj xmltype;
 xmltype_str varchar2(1000);
begin
 xmltype_blob := getblobval(createxml('<a>123</a>'),7);
 xmltype_obj := xmltype(xmltype_blob,7);
 xmltype_str := xmltype_obj.getstringval();
 RAISE NOTICE 'xmltype_str is : %',xmltype_str;
end;
/
NOTICE:  xmltype_str is : <?xml version="1.0" encoding="UTF8"?>
<a>123</a>
```

 版权所有 © 华为云计算技术有限公司

说明

- blob类型参数入参最大支持256MB-1。
- 字符集id取值范围为1~42。

- getstringval(xmltype)

  描述：此函数将xmltype转化为字符串。

  参数：需要转换的xmltype。

  返回值类型：varchar2

  getstringval函数有两种调用方式。

  示例1：

```
gaussdb=# SELECT getstringval('<a>123<b>456</b></a>');
    getstringval
----------------------
 <a>123<b>456</b></a>
(1 row)
```

  示例2：调用方式兼容ORA的语法。

```
gaussdb=# SELECT xmltype('<a>123<b>456</b></a>').getstringval();
    xmltypefunc
----------------------
 <a>123<b>456</b></a>
(1 row)
```

- getrootelement(xmltype)

  描述：此函数获取xmltype的根元素。

  参数：需要获取根元素的xmltype。

  返回值类型：varchar2

  getrootelement函数有两种调用方式。

  示例1：

```
gaussdb=# SELECT getrootelement('<a>123<b>456</b></a>');
 getrootelement
----------------
 a
(1 row)
```

  示例2：调用方式兼容ORA的语法。

```
gaussdb=# SELECT xmltype('<a>123<b>456</b></a>').getrootelement();
 xmltypefunc
-------------
 a
(1 row)
```

- getnamespace(xmltype)

  描述：此函数获取xmltype顶层元素的命名空间。

  参数：需要获取命名空间的xmltype。

  返回值类型：varchar2

  getnamespace函数有两种调用方式。

  示例1：

```
gaussdb=# SELECT getnamespace('<c:a xmlns:c="asd">123<d:b xmlns:d="qwe">456</d:b></c:a>');
 getnamespace
--------------
 asd
(1 row)
```

  示例2：调用方式兼容ORA的语法。

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT xmltype('<c:a xmlns:c="asd">123<d:b xmlns:d="qwe">456</d:b></
c:a>').getnamespace();
 xmltypefunc
-------------
 asd
(1 row)
```

- existsnode(xmltype, varchar2[, varchar2])

  描述：此函数根据xpath表达式判断在xmltype中是否存在该xml节点，如果存在返回1，否则返回0。

  参数：被查询的xmltype，查询的xpath节点路径，xpath路径的命名空间（在入参有命名空间时，xpath和命名空间都需要定义别名，如示例3）。

  返回值类型：numeric

  existsnode函数有两种调用方式。

  示例1：

  ```
  gaussdb=# SELECT existsnode('<a>123<b>456</b></a>','/a/b');
   existsnode
  ------------
            1
  (1 row)
  ```

  示例2：调用方式兼容ORA的语法。

  ```
  gaussdb=# SELECT xmltype('<a>123<b>456</b></a>').existsnode('/a/b');
   xmltypefunc
  -------------
             1
  (1 row)
  ```

  示例3：

  ```
  gaussdb=# SELECT existsnode('<a:b xmlns:a="asd">123<c>456</c></a:b>','/a:b/c','xmlns:a="asd"');
   existsnode
  ------------
            1
  (1 row)
  ```

  示例4：调用方式兼容ORA的语法。

  ```
  gaussdb=# SELECT xmltype('<a:b xmlns:a="asd">123<c>456</c></a:b>').existsnode('/a:b/
  c','xmlns:a="asd"');
   xmltypefunc
  -------------
             1
  (1 row)
  ```

- extractxml(xmltype, varchar2[, varchar2])

  描述：此函数根据xpath表达式判断在xmltype中是否存在该xml节点，如果存在返回包含该节点的xmltype，如果不存在返回NULL。可以将返回值插入xmltype类型的表中。

  参数：被查询的xmltype，查询的xpath节点路径，xpath路径的命名空间（在入参有命名空间时，xpath和命名空间都需要定义别名，如示例3）。

  返回值类型：xmltype

  extractxml函数有两种调用方式。

  示例1：

  ```
  gaussdb=# SELECT extractxml('<a>123<b>456</b></a>','/a/b');
   extractxml
  ------------
   <b>456</b>
  (1 row)
  ```

  示例2：调用方式兼容ORA的语法。

  ```
  gaussdb=# SELECT xmltype('<a>123<b>456</b></a>').extract('/a/b');
   xmltypefunc
  ```


版权所有 © 华为云计算技术有限公司

```
-------------
 <b>456</b>
(1 row)

gaussdb=# SELECT xmltype('<a>123<b>456</b></a>').extractxml('/a/b');
 xmltypefunc
-------------
 <b>456</b>
(1 row)
```

示例3：

```
gaussdb=# SELECT extractxml('<a:b xmlns:a="asd">123<c>456</c></a:b>','/a:b','xmlns:a="asd"');
              extractxml
----------------------------------------
 <a:b xmlns:a="asd">123<c>456</c></a:b>
(1 row)
```

示例4：调用方式兼容ORA的语法。

```
gaussdb=# SELECT xmltype('<a:b xmlns:a="asd">123<c>456</c></a:b>').extract('/a:b','xmlns:a="asd"');
             xmltypefunc
----------------------------------------
 <a:b xmlns:a="asd">123<c>456</c></a:b>
(1 row)

gaussdb=# SELECT xmltype('<a:b xmlns:a="asd">123<c>456</c></a:b>').extractxml('/
a:b','xmlns:a="asd"');
             xmltypefunc
----------------------------------------
 <a:b xmlns:a="asd">123<c>456</c></a:b>
(1 row)
```

- extractvalue(xmltype | xml, varchar2[, varchar2])

  描述：此函数根据xpath表达式（仅支持xpath1.0版本）在xml文本中提取该xpath表达式所对应的值。xpath表达式的结果必须为单个节点，且为文本节点、属性或元素。如果xpath表达式中含有元素表达式，则元素必须有一个文本节点作为子节点，函数返回该文本。如果结果是属性，函数返回的是属性的值。

  参数：xmltype | xml：被查询的xml文本；varchar2：需要查询的xpath表达式（xpath节点路径）；[, varchar2]：xpath节点路径的命名空间（可选，在入参有命名空间带别名时，xpath表达式和命名空间都需要定义别名,默认命名空间则不需要，如示例3）。

  返回值类型：varchar2

  extractvalue函数有两种调用方式：输入不带命名空间和输入带有命名空间。

  示例1：输入不带命名空间。

```
gaussdb=# set a_format_version='10c';
SET
gaussdb=# set a_format_dev_version='s5';
SET
gaussdb=# SELECT EXTRACTVALUE(xmltype('<book><title>Harry Potter</title><author>J.K. Rowling</
author></book>'), '/book/title') AS book_title;
  book_title
--------------
 Harry Potter
(1 row)
```

  示例2：输入带有命名空间。

```
gaussdb=# SELECT EXTRACTVALUE(xmltype('<ns:book xmlns:ns="http://
www.example.com"><ns:title>Harry Potter</ns:title><ns:author>J.K. Rowling</ns:author></ns:book>'),
'/ns:book/ns:title', 'xmlns:ns="http://www.example.com"') AS book_title;
  book_title
--------------
 Harry Potter
(1 row)
```

  示例3：输入带有多个命名空间。

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT EXTRACTVALUE(xmltype('<ns:book xmlns:ns="http://www.example.com"
xmlns:ff="http://www.ff.com"><ff:title>Harry Potter</ff:title><ns:author>J.K. Rowling</ns:author></
ns:book>'), '/ns:book/ff:title', 'xmlns:ns="http://www.example.com" xmlns:ff="http://www.ff.com"') AS
book_title;
  book_title
--------------
 Harry Potter
(1 row)
gaussdb=# SELECT EXTRACTVALUE(xmltype('<store><book xmlns="abc"><root xmlns="abcd">mike</
root><root>mikeab</root></book><root xmlns="abcd">mikedwsa</root></store>'),
'//root', 'xmlns="abc" xmlns:ns2="abc1" xmlns="abcd"') FROM dual;
 extractvalue
--------------
 mikeab
(1 row)
```

说明

- 当输入带有多个命名空间时，不同的命名空间之间可以用一个或多个空格（或换行符）隔开，但命名空间表达方式必须满足以下条件，带别名命名空间满足“xmlns:*名称*="*命名空间*"”，默认命名空间规则满足"xmlns='url'"。
- 该函数保持兼容之前的xmltype表达式函数，但需要保证xmltype文本返回节点值唯一。
- xpath表达式仅支持xpath1.0版本。
- 目前暂不支持xml文本中的命名空间url为空格，同时也不支持命名空间表达式中的命名空间url为空格。
- 在默认命名空间场景，在命名空间表达式中最先声明的默认命名空间就是当前xml文本的默认命名空间。
- 该函数仅在ORA模式下，且需要设置set a_format_version='10c'; set a_format_dev_version='s5';后才能使用。

- xmlsequence(xmltype)

  描述：此函数的功能是将一个XMLTYPE类型的参数转换为一个XMLTYPE类型的数组，每个数组元素都是XMLTYPE对象。这个函数的输入参数不能为空，也必须是一个有效的XML文档。如果输入参数不符合要求，函数会返回空值或者抛出异常。这个函数可以用于处理XML文档中的多个子节点，或者将XML文档分割为多个片段。

  参数：xmltype类型。

  返回值类型：xmltype类型的数组。

  示例1：如果想要将这个文档转换为一个包含三个元素的数组，每个元素都是一个book节点，可以使用以下语句：

```
gaussdb=# SELECT xmlsequence(xmltype('<books><book><title>The Catcher in the Rye</
title><author>J.D. Salinger</author><year>1951</year></book><book><title>1984</
title><author>George Orwell</author><year>1949</year></book><book><title>The Hitchhiker''s
Guide to the Galaxy</title><author>Douglas Adams</author><year>1979</year></book></books>'));
                                xmlsequence
------------------------------------------------------------
 {"<books>                                      +
    <book>                                      +
      <title>The Catcher in the Rye</title>     +
      <author>J.D. Salinger</author>            +
      <year>1951</year>                         +
    </book>                                     +
    <book>                                      +
      <title>1984</title>                       +
      <author>George Orwell</author>            +
      <year>1949</year>                         +
    </book>                                     +
    <book>                                      +
      <title>The Hitchhiker's Guide to the Galaxy</title>+
      <author>Douglas Adams</author>            +
      <year>1979</year>                         +
```

```
  </book>                                    +
 </books>"}
(1 row)
```

示例2：如果想要从这个数组中提取每本书的标题和作者，可以使用如下语句：

```
gaussdb=# SELECT unnest(xmlsequence(xmltype('<books><book><title>The Catcher in the Rye</
title><author>J.D. Salinger</author><year>1951</year></book><book><title>1984</
title><author>George Orwell</author><year>1949</year></book><book><title>The Hitchhiker''s
Guide to the Galaxy</title><author>Douglas Adams</author><year>1979</year></book></
books>').extract('//title/text()'))) AS title
 , unnest(xmlsequence(xmltype('<books><book><title>The Catcher in the Rye</title><author>J.D.
Salinger</author><year>1951</year></book><book><title>1984</title><author>George Orwell</
author><year>1949</year></book><book><title>The Hitchhiker''s Guide to the Galaxy</
title><author>Douglas Adams</author><year>1979</year></book></books>').extract('//author/
text()'))) AS author;
                          title                          |               author
---------------------------------------------------------------+----------------------------------------
 The Catcher in the Rye1984The Hitchhiker's Guide to the Galaxy | J.D. SalingerGeorge OrwellDouglas
Adams
(1 row)
```

示例3：如果想要将这个数组转换为一个JSON格式的字符串，可以使用以下语句：

```
gaussdb=# SELECT array_to_json(array_agg(row_to_json(t)))
FROM (
 SELECT unnest(xmlsequence(xmltype('<books><book><title>The Catcher in the Rye</
title><author>J.D. Salinger</author><year>1951</year></book><book><title>1984</
title><author>George Orwell</author><year>1949</year></book><book><title>The Hitchhiker''s
Guide to the Galaxy</title><author>Douglas Adams</author><year>1979</year></book></
books>').extract('//title/text()'))) AS title
  , unnest(xmlsequence(xmltype('<books><book><title>The Catcher in the Rye</title><author>J.D.
Salinger</author><year>1951</year></book><book><title>1984</title><author>George Orwell</
author><year>1949</year></book><book><title>The Hitchhiker''s Guide to the Galaxy</
title><author>Douglas Adams</author><year>1979</year></book></books>').extract('//author/
text()'))) AS author
) t;
                                array_to_json
-----------------------------------------------------------------------------------------------------------
 [{"title":"The Catcher in the Rye1984The Hitchhiker's Guide to the Galaxy","author":"J.D.
SalingerGeorge OrwellDouglas Adams"}]
(1 row)
```

说明

当输入xmlsequence函数的参数XML文档中含有双引号，单独查看xmlsequence函数时，结果中会包含双引号的转义符，使用xmlsequence函数结果时，不受影响。

- xmlcast(value_expression as datatype)

  描述：对于XMLTYPE类型的输入数据，在提取文档中的文本内容后，将其转换为指定的目标数据类型。

  参数：value_expression，表示XMLTYPE类型的文档内容。

  返回值类型：指定的目标数据类型。

  示例：

```
-- XMLTYPE数据转为VARCHAR2类型。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
CREATE DATABASE
gaussdb=# \c ora_db
ora_db=# SELECT XMLCAST(XMLTYPE('<varchar2_value>Hello, GaussDB!</varchar2_value>') AS
VARCHAR2(50)) AS varchar2_value FROM dual;
 varchar2_value
-----------------
 Hello, GaussDB!
(1 row)
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
DROP DATABASE
```


版权所有 © 华为云计算技术有限公司

说明

- 仅在ORA兼容模式下支持使用该函数。
- 劣构XML文档仅支持value_expression表达式返回值为多根节点的XML。
- 不支持转换的目标数据类型包括BYTEAWITHOUTORDERWITHEQUALCOL、BYTEAWITHOUTORDERCOL、_BYTEAWITHOUTORDERWITHEQUALCOL、_BYTEAWITHOUTORDERCOL以及部分伪类型。
- 升级观察期不支持使用该函数。

- xmlquery(sequence_string RETURNING SEQUENCE)

  描述：在SQL查询中联合使用XMLTABLE（即在SELECT查询中指定xmltable_clause）时，将指定的sequence_string生成序列。

  参数：sequence_string：形如“a to b”的字符串，表示生成从数字a到数字b的序列。

  返回值类型：XMLTYPE类型的数字序列。

  示例：

```
-- 生成序列。
gaussdb=# SELECT * FROM XMLTABLE ( '/' PASSING XMLQUERY ( '1 to 10' RETURNING SEQUENCE ) COLUMNS N NUMBER PATH '.' );
 n
----
  1
  2
  3
  4
  5
  6
  7
  8
  9
 10
(10 rows)
```

  说明

  - 仅在ORA兼容模式下支持使用该函数。
  - 仅支持形如“a to b”的字符串输入，且该场景仅支持在XMLTABLE中使用，XMLTABLE具体介绍请参见**指定查询源from_item中xmltable_clause**。
  - 生成序列的范围为[-9223372036854775808, 9223372036854775807]的int64数字，且生成序列个数最大为2147483647。
  - 升级观察期不支持使用该函数。

- xmlquery(xpath_string PASSING document_expression RETURNING CONTENT)

  描述：通过解析XPath表达式，可在XMLTYPE类型的文档内容中定位并返回指定的节点集合或原子值。

  参数：

参数	描述
xpath_string	用于搜索的XPath表达式。 • 仅支持XPath 1.0语法，不支持形如“namespace::*”的XPath表达式。 • 不支持XQuery表达式。

版权所有 © 华为云计算技术有限公司

参数	描述
document_expression	XMLTYPE类型的文档内容，目前仅支持单根数据。 XMLQUERY固定XML文本为根节点上下文空间开始寻找文本节点。

返回值类型：XMLTYPE类型的查询结果。

示例：

```
-- 计算XMLTYPE类型XPath表达式查询结果。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
CREATE DATABASE
gaussdb=# \c ora_db
ora_db=#
SELECT xmlquery( '/' PASSING XMLTYPE( '<PDRecord> <PDName>Leo</PDName> </PDRecord>')
RETURNING CONTENT);
                xmlquery
-------------------------------------------
 <PDRecord> <PDName>Leo</PDName> </PDRecord>+

(1 row)
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
DROP DATABASE
```

说明

- 仅在ORA兼容模式下支持使用该函数。
- 升级观察期不支持使用该函数。

1.6.46 数据透视函数

- tablefunc()

  描述：扩展接口，用于处理表数据，包括数据透视函数。仅系统管理员可以安装扩展。

  说明

  需要安装扩展，默认安装到public schema，建议安装到用户schema，create extension tablefunc [schema {user_schema}]。

- crosstab(source_sql text [, N int])

  描述：以source_sql的结果为源数据，产生一个数据透视表。

  返回值类型：setof record

  示例：

```
gaussdb=# CREATE extension tablefunc;
CREATE EXTENSION
gaussdb=# CREATE TABLE cross_test(group_id text, id int, var text);
CREATE TABLE
gaussdb=# SELECT * FROM cross_test;
 group_id | id | var
----------+----+-----
(0 rows)

gaussdb=# SELECT * FROM crosstab('SELECT group_id, var FROM cross_test order by 1, 2;') AS
c(group_ text, cat1 text, cat2 text, cat3 text);
 group_ | cat1 | cat2 | cat3
```


版权所有 © 华为云计算技术有限公司

```
--------+------+------+------
(0 rows)
```

说明

N是一个无效参数，不影响函数结果。

- crosstabN(source_sql text)

  描述：以source_sql的结果为源数据，产生一个 "N+1" 列的数据透视表。crosstabN是一组函数，包括crosstab2、crosstab3、crosstab4。

  返回值类型：setof tablefunc_crosstab_N。tablefunc_crosstab_N包括tablefunc_crosstab_2、tablefunc_crosstab_3、tablefunc_crosstab_4。

  示例：

```
--crossbatN(source_sql text)中的N为2时，产生一个3列的数据透视表。
gaussdb=# SELECT * FROM crosstab2('SELECT group_id, var FROM cross_test ORDER BY 1, 2;');
 row_name | category_1 | category_2
----------+------------+------------
(0 rows)
```

- crosstab(source_sql text, category_sql text)

  描述：以source_sql的结果为源数据，根据category_sql的结果分类，产生一个数据透视表。

  返回值类型：setof record

  示例：

```
gaussdb=# SELECT * FROM crosstab('SELECT group_id, var FROM cross_test order by 1, 2;', 'SELECT generate_series(1, 4)') AS c(group_ text, cat1 text, cat2 text, cat3 text, cat4 text);
 group_ | cat1 | cat2 | cat3 | cat4
--------+------+------+------+------
(0 rows)

gaussdb=# DROP TABLE cross_test;
gaussdb=# DROP EXTENSION tablefunc;
```

1.6.47 Global Plsql Cache 特性函数

- invalidate_plsql_object()，invalidate_plsql_object(schema, objname, objtype)

  描述：失效Global Plsql Cache全局缓存中的对象，仅在enable_global_plsqlcache = on时可用。当前版本分布式不支持该函数使用。调用该函数的用户需要具有SYSADMIN权限。

  参数：该函数为重载函数。当无入参时，将所有DATABASE内的所有全局缓存对象失效。

  当指定schema、objname、objtype三个参数时可将当前DATABASE内的指定全局缓存对象失效。其中：schema为对象所属的schema名称；objname为对象名称；objtype为对象类型，对象为package类型时值为“package”，对象为函数或存储过程时值为“function”。

- gs_plsql_memory_object_detail(db_oid, obj_oid, obj_type)

  描述：获取存储过程创建的package、function的主要有效内存占用情况。enable_global_plsqlcache = off时该函数显示当前session中的存储过程内存情况，跨session查询暂不支持。当前版本分布式不支持该函数使用。调用该函数的用户需要有SYSADMIN权限。

  参数：该函数需传入db_oid、obj_oid、obj_type三个参数，如传入参数不匹配，返回空行。具体参数说明见**表1-270**。其中db_oid、obj_oid传入0为默认查找cache中所有package、function的有效内存使用情况。

  返回值类型：Tuple

版权所有 © 华为云计算技术有限公司

表 1-270 gs_plsql_memory_object_detail 参数说明

参数列表	类型	描述	取值范围
db_oid	uint32	查询数据库oid为db_oid内存储的存储过程的主要有效内存占用情况。0默认为当前缓存中全部数据库实例。	0 - 2^32-1。
obj_oid	uint32	查询object的oid。0默认为当前缓存中全部pkg与function。	0 - 2 ^32-1。
obj_type	text	查询object的类型，标识查询目标object为package或function。	all：默认全部查询。 pkg：查询满足条件的package有效内存占用。 func：查询满足条件的function有效内存占用。 func_in_pkg：查询满足条件的package中的function的有效内存占用。

gs_plsql_memory_object_detail函数显示数据库内满足查询条件的有效内存占用情况说明如**表1-271**所示。

表 1-271 gs_plsql_memory_object_detail 返回值

名称	类型	描述
object_oid	uint32	查询内存对象的oid。
context_name	text	内存对象名。
item	text	查询内存对象项目名。
searchpath	text	内存对象访问编译产物的Schema及其他环境变量。
guc	uint64	创建对象时的环境参数，即behavior_compat_flags的值。
file	text	内存对象创建所在文件。
line	uint32	内存对象创建所在文件行数。
size	uint32	内存对象大小。
db_oid	uint32	查询内存对象所在数据库oid。

- dump_plsql_compiled_object(object_id, object_type)

  描述：打印存储过程的PACKAGE、FUNCTION编译产物信息。GUC参数enable_global_plsqlcache设置为off时，该函数只能查找当前session内的编译产

版权所有 © 华为云计算技术有限公司

物信息。暂不支持跨session、跨database查询。调用该函数需要有SYSADMIN权限。

参数：该函数需要传入object_oid和object_type两个参数，如果传入参数的数量或者类型不匹配，返回报错。具体参数说明如**表1-272**所示。

返回值类型：text

表 1-272 dump_plsql_compiled_object 参数说明

参数列表	类型	描述	取值范围
object_oid	uint32	查询object的oid。	0~2^32-1。
object_type	text	查询object的类型（不区分大小写）。	• "package"：查询object类型为package。 • "function"：查询object类型为function（查询效果与procedure等价）。 • "procedure"：查询object类型为procedure（查询效果与function等价）。

示例：

```
-- 如下用例构造当GUC参数enable_global_plsqlcache设置为off时，创建存储过程，调用
dump_plsql_compiled_object打印存储过程编译产物信息的情况。
gaussdb=# \x
Expanded display is on.
gaussdb=# CREATE OR REPLACE PROCEDURE test_proc(a out int, b out int) AS
gaussdb$# BEGIN
gaussdb$# a := 1;
gaussdb$# b := 1;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# SELECT test_proc();
-[ RECORD 1 ]----
test_proc | (1,1)


gaussdb=# SELECT oid FROM pg_proc WHERE proname = 'test_proc';
-[ RECORD 1 ]
oid | 16388


gaussdb=# SELECT * FROM dump_plsql_compiled_object(16388, 'function');
-[ RECORD 1 ]-------------
+-------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
dump_plsql_compiled_object | PL/pgSQL compiled function info:
                           | fn_oid: 16388
                           | pkg_oid: 0
                           | namespace_oid: 2200
                           | fn_owner: 10
                           | fn_input_collation: 0
                           | fn_xmin: 195809
                           | fn_signature: test_proc()
```

```
                        | fn_searchpath:{ 2200 , addCatalog: true, addTemp: true, addUser: false}
                        | fn_rettype: 2249
                        | fn_rettyplen: 4294967295
                        | fn_rettypioparam: 0
                        | glc_func_life: 2
                        | fn_retbyval: false
                        | fn_retistuple: true
                        | fn_retset: false
                        | fn_readonly: false
                        | is_private: false
                        | fn_is_trigger: false
                        | pre_parse_trig: false
                        | is_autonomous: false
                        | is_inline_handler: false
                        | is_plpgsql_func_with_outparam: false
                        | need_skip_process_autonm_pkg: false
                        | remembered_by_resowner: false
                        | out_param_varno: 2
                        | found_varno: 3
                        | fn_nallargs: 2
                        | sql_cursor_found_varno: 4
                        | sql_notfound_varno: 5
                        | sql_isopen_varno: 6
                        | sql_rowcount_varno: 7
                        | sql_bulk_exceptions_varno: 0
                        | sqlcode_varno: 8
                        | sqlstate_varno: 9
                        | sqlerrm_varno: 10
                        | retvarno: 0
                        | guc_stat: 0
                        | ref_count: 0
                        | resolve_option: 0
                        | ndatums: 11
                        | datums:{
                        | type: GSPLSQL_DTYPE_VAR, dno: 0 ispkg: false refname: $1 isImplicit: false
addNamespace: false pkg_oid: 0 varname: a isconst: 0,notnull: 0, isnull: true freeval: false
is_place_holder: false isbind: false is_cursor_var: false is_cursor_open: false is_diff_pkg_cursor: false
tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: int4, typoid: 23, ttype: 0, typlen: 4, typrelid: 0, typioparam:
23, collation: 0, typinput_oid: 42, atttypmod: -1, typbyval: true, typtype: b, typnamespace: pg_catalog,
collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0, cursorCompositeOid: 0,
subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 1 ispkg: false refname: $2 isImplicit: false
addNamespace: false pkg_oid: 0 varname: b isconst: 0,notnull: 0, isnull: true freeval: false
is_place_holder: false isbind: false is_cursor_var: false is_cursor_open: false is_diff_pkg_cursor: false
tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: int4, typoid: 23, ttype: 0, typlen: 4, typrelid: 0, typioparam:
23, collation: 0, typinput_oid: 42, atttypmod: -1, typbyval: true, typtype: b, typnamespace: pg_catalog,
collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0, cursorCompositeOid: 0,
subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_ROW, dno: 2 ispkg: false refname: (null) isImplicit: true
addNamespace: false pkg_oid: 0, varname: (null),
                        | TupleDesc: tdTableAmType: 0, tdisredistable: false, attrs:{ 0: 23 1: 23 }
                        |  tdtypeid: 2249, tdtypmod: -1, tdhasoid: false, tdrefcount: -1, tdhasuids: false,
                        | customErrorCode: 0, intoplaceholders: 0, recordVarTypOid: 0, hasExceptionInit:
false, is_diff_pkg_cursor: false
                        | default_val: nfields: 2, row fields:{ $1 = var 0 $2 = var 1 }
                        | type: GSPLSQL_DTYPE_VAR, dno: 3 ispkg: false refname: found isImplicit: true
addNamespace: true pkg_oid: 0 varname: (null) isconst: 0,notnull: 0, isnull: true freeval: false
is_place_holder: false isbind: false is_cursor_var: false is_cursor_open: false is_diff_pkg_cursor: false
tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: bool, typoid: 16, ttype: 0, typlen: 1, typrelid: 0,
typioparam: 16, collation: 0, typinput_oid: 1242, atttypmod: -1, typbyval: true, typtype: b,
typnamespace: pg_catalog, collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0,
cursorCompositeOid: 0, subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 4 ispkg: false refname:
```


版权所有 © 华为云计算技术有限公司

```
__gsdb_sql_cursor_attri_found__ isImplicit: true addNamespace: true pkg_oid: 0 varname: (null)
isconst: 0,notnull: 0, isnull: true freeval: false is_place_holder: false isbind: false is_cursor_var: false
is_cursor_open: false is_diff_pkg_cursor: false tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: bool, typoid: 16, ttype: 0, typlen: 1, typrelid: 0,
typioparam: 16, collation: 0, typinput_oid: 1242, atttypmod: -1, typbyval: true, typtype: b,
typnamespace: pg_catalog, collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0,
cursorCompositeOid: 0, subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 5 ispkg: false refname:
__gsdb_sql_cursor_attri_notfound__ isImplicit: true addNamespace: true pkg_oid: 0 varname: (null)
isconst: 0,notnull: 0, isnull: true freeval: false is_place_holder: false isbind: false is_cursor_var: false
is_cursor_open: false is_diff_pkg_cursor: false tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: bool, typoid: 16, ttype: 0, typlen: 1, typrelid: 0,
typioparam: 16, collation: 0, typinput_oid: 1242, atttypmod: -1, typbyval: true, typtype: b,
typnamespace: pg_catalog, collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0,
cursorCompositeOid: 0, subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 6 ispkg: false refname:
__gsdb_sql_cursor_attri_isopen__ isImplicit: true addNamespace: true pkg_oid: 0 varname: (null)
isconst: 0,notnull: 0, isnull: true freeval: false is_place_holder: false isbind: false is_cursor_var: false
is_cursor_open: false is_diff_pkg_cursor: false tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: bool, typoid: 16, ttype: 0, typlen: 1, typrelid: 0,
typioparam: 16, collation: 0, typinput_oid: 1242, atttypmod: -1, typbyval: true, typtype: b,
typnamespace: pg_catalog, collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0,
cursorCompositeOid: 0, subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 7 ispkg: false refname:
__gsdb_sql_cursor_attri_rowcount__ isImplicit: true addNamespace: true pkg_oid: 0 varname: (null)
isconst: 0,notnull: 0, isnull: true freeval: false is_place_holder: false isbind: false is_cursor_var: false
is_cursor_open: false is_diff_pkg_cursor: false tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: int4, typoid: 23, ttype: 0, typlen: 4, typrelid: 0, typioparam:
23, collation: 0, typinput_oid: 42, atttypmod: -1, typbyval: true, typtype: b, typnamespace: pg_catalog,
collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0, cursorCompositeOid: 0,
subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 8 ispkg: false refname: sqlcode isImplicit: true
addNamespace: true pkg_oid: 0 varname: (null) isconst: 0,notnull: 0, isnull: true freeval: false
is_place_holder: false isbind: false is_cursor_var: false is_cursor_open: false is_diff_pkg_cursor: false
tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: int4, typoid: 23, ttype: 0, typlen: 4, typrelid: 0, typioparam:
23, collation: 0, typinput_oid: 42, atttypmod: -1, typbyval: true, typtype: b, typnamespace: pg_catalog,
collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0, cursorCompositeOid: 0,
subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 9 ispkg: false refname: sqlstate isImplicit: true
addNamespace: true pkg_oid: 0 varname: (null) isconst: 0,notnull: 0, isnull: true freeval: false
is_place_holder: false isbind: false is_cursor_var: false is_cursor_open: false is_diff_pkg_cursor: false
tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: text, typoid: 25, ttype: 0, typlen: -1, typrelid: 0,
typioparam: 25, collation: 100, typinput_oid: 46, atttypmod: -1, typbyval: false, typtype: b,
typnamespace: pg_catalog, collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0,
cursorCompositeOid: 0, subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | type: GSPLSQL_DTYPE_VAR, dno: 10 ispkg: false refname: sqlerrm isImplicit: true
addNamespace: true pkg_oid: 0 varname: (null) isconst: 0,notnull: 0, isnull: true freeval: false
is_place_holder: false isbind: false is_cursor_var: false is_cursor_open: false is_diff_pkg_cursor: false
tableOfIndexType: 0 isIndexByTblOf: false
                        | GSPLSQLType: typname: text, typoid: 25, ttype: 0, typlen: -1, typrelid: 0,
typioparam: 25, collation: 100, typinput_oid: 46, atttypmod: -1, typbyval: false, typtype: b,
typnamespace: pg_catalog, collectionType: 0, tableofOid: 0, tableOfIndexType: 0, arrayTypOid: 0,
cursorCompositeOid: 0, subtypOid: 0, is_subtype: false, notnull: false, temptypmod: -1
                        | default_val: cursor_explicit_expr:
                        | }
                        | datum_need_free:{ 0: true 1: true 2: true 3: true 4: true 5: true 6: true 7: true 8:
true 9: true 10: true }
                        | namespace_name: public
                        | glc_status:{m_type: GLC_FUNCTION_OBJ, m_location: in_session,
m_glc_object_state: valid, m_refcount: 0}
                        | fn_nargs: 0
```


版权所有 © 华为云计算技术有限公司

```
| plan_total_mem_size: 30720
| dump count: 1
| ********************************************************************************
|
```

1.6.48 通用标识符函数

- sys_guid()

  描述：产生并返回一个全局唯一的标识符，由16个字节组成，生成的标识符有当前机器IP、生成时间戳和随机数，由以上标识符组成一个全局唯一的uuid。

  参数：无。

  返回值类型：raw

  示例：

```
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s5';
SET
gaussdb=# SELECT sys_guid();
             sys_guid
-----------------------------------
 9010675E560CB33C1BDAFA163E378F87
 (1 row)
```

- uuid()

  描述：返回一个根据RFC 4122、ISO/IEF 9834-8:2005以及相关标准定义的通用唯一标识符（UUID）。这个标识符是一个小写十六进制数字的字符串，由分字符分成几组，一组8位数字+三组4位数字+一组12位数字，总共32个数字代表128位。

  参数：无。

  返回值类型：varchar

  示例：

```
gaussdb=# SELECT uuid();
                 uuid
--------------------------------------
 dd8cbe92-1a25-013c-a514-e435c87e9182
(1 row)
```

- uuid_short()

  描述：返回一个在一定条件下具有唯一性的短通用标识符。这个标识符是一个64位无符号类型的整数。

  在满足下列条件时，返回的值是唯一的。

  - 当前集群下的服务节点数不能超过256个。
  - 不能在节点重新启动之间设置服务器主机的系统时间。
  - 在节点重新启动之间平均每秒调用uuid_short() 少于1600万次。

  参数：无。

  返回值类型：uint64

  示例：

```
gaussdb=# SELECT uuid_short();
    uuid_short
--------------------
 100440026956955649
(1 row)
```

  说明

  从505.0.0之前的版本升级至最新版本，升级未提交情况下不能使用uuid_short()函数。

版权所有 © 华为云计算技术有限公司

1.6.49 SQL 限流函数

- gs_add_workload_rule(rule_type, rule_name, databases, start_time, end_time, max_workload, option_val)

  描述：创建一条SQL限流规则。需要具有sysadmin权限的用户才可执行。只支持在CN节点上执行。

  参数：参数介绍请参见**表1-273**。

  返回值类型：int8

  表 1-273 gs_add_workload_rule 参数说明

参数名称	类型	描述	取值范围
rule_type	text	限流规则类型，不区分大小写。	“sqlid”：根据Unique SQL ID进行限流。 “select”、“insert”、“update”、“delete”、“merge”：根据查询类型和关键字进行限流。 “resource”：根据系统资源利用率进行CN进程级别的限流。 “client”：根据客户端信息，包括IP、ApplicationName进行限流。 **说明** 限流规则生效的优先级如下： client > resource > sqlid > select = insert = update = delete = merge
rule_name	name	限流规则名称，用于检索限流规则。	任意字符串，可以为NULL。
databases	name[]	限流规则生效的数据库名称数组，区分大小写。	数据库名列表，必须为已创建的数据库名。可以为NULL，表示所有数据库生效。多个数据库之间是“或”的关系。 目前只有指定rule_type为查询类型或client类型时，数据库列表才生效，因为Unique SQL ID本身是与库进行绑定的，其只属于某个库；客户端进行连接时会指定库；而根据资源利用率的限流规则是对整个CN进程生效的，即对所有库生效。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述	取值范围
start_time	timestamptz	限流规则生效的开始时间。	可以为NULL，表示从当前时间开始生效。 **说明** 建议在创建限流规则时，SQL中尽量避免使用 sysdate 作为限流规则的开始时间和结束时间，因为 sysdate 的精度仅到秒级，其秒后的小数部分会被四舍五入，可能导致限流规则在极短时间内不生效，推荐使用now()替换sysdate。
end_time	timestamptz	限流规则生效的结束时间。	可以为NULL或''，表示规则一直生效。
max_workload	int8	限流规则设置的最大并发数。	-
option_val	text[]	限流规则的补充信息。	与rule_type匹配，具体匹配关系如下： ● “sqlid”：要限流的Unique SQL ID，以及慢SQL管控规则，格式为'{id=1234, time_limit=100, max_execute_time=500, max_iops=1}'，其中id值为Unique SQL ID，为必选项，可通过dbe_perf.statement或者pg_stat_activity视图获取。其他选项非必选，其含义请参见《SQL调优指南》中“使用Plan Hint进行调优 > 设置慢SQL管控规则的Hint”章节。 ● “select”、“insert”、“update”、“delete”、“merge”：要限流的关键字序列，不区分大小写，可以为NULL。多个关键字之间是“且”的关系。 ● “resource”：要限流的资源阈值，形式为'{cpu-80, memory-70}'。需要注意的是，无论当前cpu和memory设置为多少，都当成“0”处理。即：无论实际的cpu和内存使用率是多少，都会触发此限流规则。 ● “client”：要限流的客户端信息，包括IP、ApplicationName，格式为'{ip=192.168.1.1, appname=app1}'，其中ip或appname至少指定其中之一。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=#  SELECT gs_add_workload_rule('sqlid', 'rule for one query', '{}', now(), NULL, 20,
'{id=32413214}');
 gs_add_workload_rule
----------------------
                    1
(1 row)

gaussdb=#  CREATE DATABASE db1;
gaussdb=#  CREATE DATABASE db2;
gaussdb=#  SELECT gs_add_workload_rule('select', 'rule for select', '{db1, db2}', NULL, NULL, 100,
'{tb1, tb2}');
 gs_add_workload_rule
----------------------
                    2
(1 row)

gaussdb=#  SELECT gs_add_workload_rule('resource', 'rule for resource', '{}', NULL, NULL, 20,
'{cpu-80}');
 gs_add_workload_rule
----------------------
                    3
(1 row)

gaussdb=# SELECT gs_add_workload_rule('client', 'rule for client', '{}', NULL, NULL, 20,
'{appname=app1}');
 gs_add_workload_rule
----------------------
                    4
(1 row)

gaussdb=#  SELECT gs_delete_workload_rule(rule_id) FROM gs_workload_rule; -- 删除全部限流规则
 gs_delete_workload_rule
-------------------------
 t
 t
 t
 t
(4 rows)

gaussdb=#  DROP DATABASE db1;
gaussdb=#  DROP DATABASE db2;
```

- gs_update_workload_rule(rule_id, rule_name, databases, start_time, end_time, max_workload, option_val)

  描述：更新一条SQL限流规则，需要重新设置全部参数，不支持只指定部分参数。需要具有sysadmin权限的用户才可执行。只支持在CN节点上执行。

  参数：参数介绍请参见**表1-274**。

  返回值类型：BOOLEAN

  表 1-274 gs_update_workload_rule 参数说明

参数名称	类型	描述	取值范围
rule_id	int8	要更新的限流规则ID。	-
rule_name	name	限流规则名称，用于检索限流规则。	任意字符串，可以为NULL。

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述	取值范围
databases	name[]	限流规则生效的数据库名称数组，区分大小写。	数据库名列表，必须为已创建的数据库名。可以为NULL，表示所有数据库生效。多个数据库之间是“或”的关系。 目前只有指定rule_type为查询类型或client类型时，数据库列表才生效，因为Unique SQL ID本身是与库进行绑定的，其只属于某个库；客户端进行连接时会指定库；而根据资源利用率的限流规则是对整个CN进程生效的，即对所有库生效。
start_time	timestamptz	限流规则生效的开始时间。	可以为NULL，表示从当前时间开始生效。 **说明** 建议在更新限流规则时，SQL中尽量避免使用 sysdate 作为限流规则的开始时间和结束时间，因为 sysdate 的精度仅到秒级，其秒后的小数部分会被四舍五入，可能导致限流规则在极短时间内不生效，推荐使用now()替换sysdate。
end_time	timestamptz	限流规则生效的结束时间。	可以为NULL，表示规则一直生效。
max_workload	int8	限流规则设置的最大并发数。	-

版权所有 © 华为云计算技术有限公司

参数名称	类型	描述	取值范围
option_val	text[]	限流规则的补充信息。	与rule_type匹配，具体匹配关系如下： ● “sqlid”：要限流的Unique SQL ID，以及慢SQL管控规则，格式为'{id=1234, time_limit=100, max_execute_time=500, max_iops=1}'，其中id指为Unique SQL ID，为必选项，可通过dbe_perf.statement或者pg_stat_activity视图获取。其他选项非必选，其含义请参见《SQL调优指南》中“使用Plan Hint进行调优 > 设置慢SQL管控规则的Hint”章节。 ● “select”、“insert”、“update”、“delete”、“merge”：要限流的关键字序列，不区分大小写，可以为NULL。多个关键字之间是“且”的关系。 ● “resource”：要限流的资源阈值，形式为'{cpu-80, memory-70}'。需要注意的是，当前cpu和memory无论设置为多少，都当成“0"处理。即：无论实际的cpu和内存使用率是多少，都会触发此限流规则。 ● “client”：要限流的客户端信息，包括IP、ApplicationName，格式为'{ip=192.168.1.1, appname=app1}'，其中ip或appname至少指定其中之一。

示例：

```
gaussdb=#  CREATE DATABASE db1;
gaussdb=#  SELECT gs_add_workload_rule('select', 'rule for select', '{db1}', NULL, NULL, 100, '{tb1,
tb2}');
 gs_add_workload_rule
----------------------
                    5
(1 row)

gaussdb=#  SELECT gs_update_workload_rule(5, 'rule for select 2', '{db1}', now(), NULL, 50, '{tb1}');
 gs_update_workload_rule
-------------------------
 t
(1 row)

gaussdb=#  SELECT gs_delete_workload_rule(rule_id) FROM gs_workload_rule; -- 删除全部限流规则
```


版权所有 © 华为云计算技术有限公司

```
 gs_delete_workload_rule
-------------------------
 t
(1 row)

gaussdb=#  DROP DATABASE db1;
```

- gs_delete_workload_rule(rule_id)

  描述：删除一条SQL限流规则。需要具有sysadmin权限的用户才可执行。只支持在CN节点上执行。

  参数：rule_id，要删除的限流规则ID，类型为int8。

  返回值类型：BOOLEAN

  示例：

```
gaussdb=#  SELECT gs_add_workload_rule('sqlid', 'rule for one query', '{}', NULL, NULL, 20,
'{id=32413214}');
 gs_add_workload_rule
----------------------
                    6
(1 row)

gaussdb=#  SELECT gs_delete_workload_rule(6);
 gs_delete_workload_rule
-------------------------
 t
(1 row)
```

- gs_get_workload_rule_stat(rule_id)

  描述：查询SQL限流规则拦截SQL的次数。需要具有sysadmin权限的用户才可执行。只支持在CN节点上执行。

  参数：rule_id，要查询的限流规则ID，类型为int8。可以指定rule_id为-1，此时表示查询所有的SQL限流规则。

  表3 返回值类型说明

名称	类型	描述
rule_id	int8	SQL限流规则的ID。
validate_count	int8	SQL限流规则拦截SQL的次数。

  示例：

```
gaussdb=#  SELECT gs_add_workload_rule('sqlid', 'rule for one query', '{}', now(), NULL, 20,
'{id=32413214}');
 gs_add_workload_rule
----------------------
                    7
(1 row)

gaussdb=#  SELECT gs_add_workload_rule('resource', 'rule for resource', '{}', NULL, NULL, 20,
'{cpu-80}');
 gs_add_workload_rule
----------------------
                    8
(1 row)

gaussdb=#  SELECT * FROM gs_get_workload_rule_stat(7);
 rule_id | validate_count
---------+----------------
       7 |              0
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=#  SELECT * FROM gs_get_workload_rule_stat(-1);
 rule_id | validate_count
---------+----------------
       7 |              0
       8 |              0
(2 rows)

gaussdb=#  SELECT gs_delete_workload_rule(rule_id) FROM gs_workload_rule;
 gs_delete_workload_rule
-------------------------
 t
 t
(2 rows)
```

- gs_refresh_workload_rule_cache()

  描述：刷新限流规则缓存，使其与限流规则系统表中的数据重新保持一致。不支持在事务中执行。需要具有sysadmin权限的用户才可执行。

  返回值类型：BOOLEAN

  说明

  当在事务中对限流规则执行增、删、改操作且该事务最终发生回滚时，由于限流规则缓存无法感知事务回滚，系统表将恢复至事务开始前的状态，而限流规则缓存仍会保留变更后的内容，从而导致系统表与限流规则缓存数据不一致，此时需及时调用本函数手动同步。

1.6.50 SQL 工具函数

用于管理SQL PATCH的工具，包括创建、删除、开启、禁用SQL PATCH等系统函数。普通用户只有usage权限，没有CREATE、ALTER、DROP、COMMENT等权限。

具体使用请参见《SQL调优指南》中“使用SQL PATCH进行调优”章节。

- DBE_SQL_UTIL.create_hint_sql_patch(patch_name, unique_sql_id, hint_string, description, enabled, query_string, is_sql_hash)

  描述：create_hint_sql_patch是用于在当前建立连接的CN上创建调优SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-275**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  说明

  - CN之间SQL PATCH不共享，要在各个CN上单独创建。
  - 开启负载均衡场景或者需要指定创建的CN的场景，推荐使用**DBE_SQL_UTIL.create_remote_hint_sql_patch**接口进行创建。
  - query_string传入EXECUTE语句时，系统会获取当前会话对应的PREPARE语句，并创建SQL PATCH。如果此时没有对应的PREPARE语句，则无效。

表 1-275 DBE_SQL_UTIL.create_hint_sql_patch 入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
unique_sql_id	IN bigint	查询全局唯一id。
hint_string	IN text	Hint文本。
description	IN text	PATCH的备注，默认值为NULL。
enabled	IN bool	PATCH是否生效，默认值为true。
query_string	IN text	SQL HASH对应的原SQL，默认值为NULL。
is_sql_hash	IN bool	PATCH是否是基于SQL HASH创建的，默认值为false。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.create_hint_sql_patch(patch_name, unique_sql_id, parent_unique_sql_id, hint_string, description, enabled)

  描述：create_hint_sql_patch是用于在当前建立连接的CN上创建调优SQL PATCH的接口函数，返回执行是否成功。本函数是原函数的重载函数，支持通过parent_unique_sql_id值限制hint patch的生效范围。

  参数：参数介绍请参见**表1-276**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  表 1-276 DBE_SQL_UTIL.create_hint_sql_patch 重载函数入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
unique_sql_id	IN bigint	查询全局唯一id。
parent_unique_sql_id	IN bigint	标识外层SQL语句的全局唯一id，表示SQL PATCH生效的范围。 • 为0时：表示SQL PATCH仅在外层SQL语句中生效。 • 为非0时：表示SQL PATCH仅在特定的存储过程中生效。
hint_string	IN text	Hint文本。
description	IN text	PATCH的备注，默认值为NULL。
enabled	IN bool	PATCH是否生效，默认值为true。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.create_remote_hint_sql_patch(node_name, patch_name, unique_sql_id, hint_string, description, enabled, query_string, is_sql_hash)

  描述：create_remote_hint_sql_patch是用于指定CN创建调优SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-277**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin、monadmin用户有权限调用。

  说明

  query_string传入EXECUTE语句时，系统会获取当前会话对应的PREPARE语句，并创建SQL PATCH。如果此时没有对应的PREPARE语句，则无效。

  表 1-277 DBE_SQL_UTIL.create_remote_hint_sql_patch 入参和返回值列表

参数	类型	描述
node_name	IN text	目标节点名，为'ALL'时表示给所有CN节点创建SQL PATCH。
patch_name	IN name	PATCH名称。
unique_sql_id	IN bigint	查询全局唯一id。
hint_string	IN text	Hint文本。
description	IN text	PATCH的备注，默认值为NULL。
enabled	IN bool	PATCH是否生效，默认值为true。
query_string	IN text	SQL HASH对应的原SQL，默认值为NULL。
is_sql_hash	IN bool	是否是基于SQL HASH创建的PATCH，默认值为false。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.create_abort_sql_patch(patch_name, unique_sql_id, description, enabled, query_string, is_sql_hash)

  描述：create_abort_sql_patch是用于在当前建立连接的CN上创建SQL异常阻断补丁的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-278**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

版权所有 © 华为云计算技术有限公司

说明

- CN之间SQL PATCH不共享，要在各个CN上单独创建。
- 开启负载均衡场景或者需要指定创建的CN的场景，推荐使用DBE_SQL_UTIL.create_remote_abort_sql_patch接口进行创建。
- query_string传入EXECUTE语句时，系统会获取当前会话对应的PREPARE语句，并创建SQL PATCH。如果此时没有对应的PREPARE语句，则无效。

表 1-278 DBE_SQL_UTIL.create_abort_sql_patch 入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
unique_sql_id	IN bigint	查询全局唯一id。
description	IN text	PATCH的备注，默认值为NULL。
enabled	IN bool	PATCH是否生效，默认值为true。
query_string	IN text	SQL HASH对应的原SQL，默认值为NULL。
is_sql_hash	IN bool	PATCH是否是基于SQL HASH创建的，默认值为false。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.create_abort_sql_patch(patch_name, unique_sql_id, parent_unique_sql_id, description, enabled)

  描述：create_abort_sql_patch是用于在当前建立连接的CN上创建SQL异常阻断补丁的接口函数，返回执行是否成功。本函数是原函数的重载函数，支持通过parent_unique_sql_id值限制SQL异常阻断补丁的生效范围。

  参数：参数介绍请参见**表1-279**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  表 1-279 DBE_SQL_UTIL.create_abort_sql_patch 重载函数入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
unique_sql_id	IN bigint	查询全局唯一id。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
parent_unique_sql_id	IN bigint	标识外层SQL语句的全局唯一id，表示SQL PATCH生效的范围。 • 为0时：表示SQL PATCH仅在外层SQL语句中生效。 • 为非0时：表示SQL PATCH仅在特定的存储过程中生效。
description	IN text	PATCH的备注，默认值为NULL。
enabled	IN bool	PATCH是否生效，默认值为true。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.create_remote_abort_sql_patch(node_name, patch_name, unique_sql_id, description, enabled, query_string, is_sql_hash)

  描述：create_remote_abort_sql_patch是用于指定CN创建SQL异常阻断补丁的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-280**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  说明

  query_string传入EXECUTE语句时，系统会获取当前会话对应的PREPARE语句，并创建SQL PATCH。如果此时没有对应的PREPARE语句，则无效。

  表 1-280 DBE_SQL_UTIL.create_remote_abort_sql_patch 入参和返回值列表

参数	类型	描述
node_name	IN text	目标节点名，为'ALL'时表示给所有CN节点创建SQL PATCH。
patch_name	IN name	PATCH名称。
unique_sql_id	IN bigint	查询全局唯一id。
description	IN text	PATCH的备注，默认值为NULL。
enabled	IN bool	PATCH是否生效，默认值为true。
query_string	IN text	SQL HASH对应的原SQL，默认值为NULL。
is_sql_hash	IN bool	是否是基于SQL HASH创建的PATCH，默认值为false。
result	OUT bool	执行是否成功。

版权所有 © 华为云计算技术有限公司

- DBE_SQL_UTIL.drop_sql_patch(patch_name)

  描述：drop_sql_patch是用于在当前建立连接的CN上删除SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-281**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  说明

  - CN之间SQL PATCH不共享，要在各个CN上单独执行。
  - 开启负载均衡场景或者需要指定创建的CN的场景，推荐使用**DBE_SQL_UTIL.drop_remote_sql_patch**接口进行删除操作。

  表 1-281 DBE_SQL_UTIL.drop_sql_patch 入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.drop_remote_sql_patch(node_name, patch_name)

  描述：drop_remote_sql_patch是用于指定CN删除SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-282**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  表 1-282 DBE_SQL_UTIL.drop_remote_sql_patch 入参和返回值列表

参数	类型	描述
node_name	IN text	目标节点名。
patch_name	IN name	PATCH名称。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.enable_sql_patch(patch_name)

  描述：enable_sql_patch是用于在当前建立连接的CN上开启SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-283**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

版权所有 © 华为云计算技术有限公司

说明

- CN之间SQL PATCH不共享，要在各个CN上单独执行。
- 开启负载均衡场景或者需要指定创建的CN的场景，推荐使用DBE_SQL_UTIL.enable_remote_sql_patch接口进行开启操作。

表 1-283 DBE_SQL_UTIL.enable_sql_patch 入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.enable_remote_sql_patch(node_name, patch_name)

  描述：enable_remote_sql_patch是用于指定CN开启SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见表1-284。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  表 1-284 DBE_SQL_UTIL.enable_remote_sql_patch 入参和返回值列表

参数	类型	描述
node_name	IN text	目标节点名。
patch_name	IN name	PATCH名称。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.disable_sql_patch(patch_name)

  描述：disable_sql_patch是用于在当前建立连接的CN上禁用SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见表1-285。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  说明

  - CN之间SQL PATCH不共享，要在各个CN上单独执行。
  - 开启负载均衡场景或者需要指定创建的CN的场景，推荐使用DBE_SQL_UTIL.disable_remote_sql_patch接口进行禁用操作。

版权所有 © 华为云计算技术有限公司

表 1-285 DBE_SQL_UTIL.disable_sql_patch 入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.disable_remote_sql_patch(node_name, patch_name)

  描述：disable_remote_sql_patch是用于指定CN禁用SQL PATCH的接口函数，返回执行是否成功。

  参数：参数介绍请参见**表1-286**。

  返回值类型：boolean

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  表 1-286 DBE_SQL_UTIL.disable_remote_sql_patch 入参和返回值列表

参数	类型	描述
node_name	IN text	目标节点名。
patch_name	IN name	PATCH名称。
result	OUT bool	执行是否成功。

- DBE_SQL_UTIL.show_sql_patch(patch_name, unique_sql_id, status, enabled, abort)

  描述：show_sql_patch是用于显示给定patch_name所对应的SQL PATCH的接口函数，返回运行结果。

  参数：参数介绍请参见**表1-287**。

  返回值类型：record

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  表 1-287 DBE_SQL_UTIL.show_sql_patch 入参和返回值列表

参数	类型	描述
patch_name	IN name	PATCH名称。
unique_sql_id	OUT bigint	查询全局唯一id。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
status	OUT char	表示该PATCH是基于SQL ID还是Unique SQL HASH创建的。 ● 'd'：表示Unique SQL ID。 ● 'h'：表示SQL HASH。
enabled	OUT bool	PATCH是否生效。
abort	OUT bool	是否是AbortHint。
hint_str	OUT text	Hint文本。

- DBE_SQL_UTIL.sqlpatch_query_hash(query_string, need_validate)

  描述：sqlpatch_query_hash是一个用于获取SQL对应SQL HASH的接口函数，返回传入SQL对应的HASH值。

  参数：参数介绍请参见**表1-288**。

  返回值类型：int8

  限制：仅初始用户、sysadmin、opradmin和monadmin用户有权限调用。

  说明

  在打开自动参数化，即cursor_sharing参数取值为force时，SQL HASH是基于参数化后的SQL文本计算的，使用sqlpatch_query_hash应该传入参数化后的SQL文本，并传入need_validate参数为false，得到自动参数化后的SQL HASH。例如：SELECT DBE_SQL_UTIL.sqlpatch_query_hash('SELECT * FROM t WHERE a = ?', false);

  表 1-288 DBE_SQL_UTIL.sqlpatch_query_hash 函数入参和返回值列表

参数	类型	描述
query_string	IN text	SQL文本。
need_validate	IN boolean	用于控制是否进行语法校验。
result	OUT bigint	传入SQL文本对应的HASH值。

- DBE_SQL_UTIL.gs_show_plan_by_hash(sql_hash, plan_hash)

  描述：当前版本中，若使用gs_show_plan_by_hash函数，会返回该函数不支持当前形态的提示信息。

  参数：参数介绍请参见**表1-289**。

  返回值类型：text

版权所有 © 华为云计算技术有限公司

表 1-289 DBE_SQL_UTIL.gs_show_plan_by_hash 入参和返回值列表

参数	类型	描述
sql_hash	IN bigint	SQL文本的hash值。
plan_hash	IN bigint	SQL计划的hash值。
gs_show_plan_by_hash	OUT text	返回结果为该函数不支持当前形态的提示信息。

1.6.51 多租数据库函数

下面涉及函数为多租数据库特性使用函数，分布式暂不支持该特性。以下函数不支持在分布式下使用。

- gs_reload_pdb_conf(name pdbname)
- gs_resplan_cgroup_info()
- gs_resplan_stat_info(view_type integer)
- gs_get_pdb_tablespace_location(pdb_id oid in, tablespace_oid oid in)
- get_instr_db_rt_percentile()
- get_instr_user_login_info()
- gs_resplan_shared_memory_info()

以下函数内部系统视图使用，不建议在分布式下使用。

- get_mtd_user()
- get_instr_sql_count_info()

1.6.52 全局临时表函数

- pg_get_gtt_relstats(relOid)

  描述：显示当前会话指定的全局临时表的基本信息。分布式全局临时表在CN上查询结果不具有参考价值，DN上查询结果为空，因此不建议在分布式上使用。

  参数：全局临时表的OID。

  返回值类型：record

  示例:

  ```
  gaussdb=# SELECT * FROM pg_get_gtt_relstats(24576);
   relfilenode | relpages | reltuples | relallvisible | relfrozenxid | relminmxid
  -------------+----------+-----------+---------------+--------------+------------
         24576 |        0 |         0 |             0 |        53969 |          0
  (1 row)
  ```

- pg_get_gtt_statistics(relOid, attnum, ''::text)

  描述：显示当前会话指定的全局临时表的单列统计信息。使用时，需先对表进行ANALYZE。

  参数：全局临时表的OID和属性attnum。

  返回值类型：record

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT * FROM  pg_get_gtt_statistics(24576, 1,''::text);
 starelid | starelkind | staattnum | stainherit | stanullfrac | stawidth | stadistinct | stakind1 | stakind2 |
stakind3 | stakind4 | stakind5 | staop1 | staop2 | staop3 | staop4 | staop5 | stanumbers1 | stanumbers2
| stanumbers3 | stanumbers
4 | stanumbers5
|
                                    stavalues1

          | stavalues2 | stavalues3 | stavalues4 | stavalues5 | stadndistinct | staextinfo

----------+------------+-----------+------------+-------------+----------+-------------+----------+----------
+----------+----------+----------+--------+--------+--------+--------+--------+-------------+-------------
+-------------+-----------
--+-------------
+-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
----------------------------------------+------------+------------+------------+------------+---------------
+-----------
-
   24576 | c          |         1 | f          |           0 |        4 |          -1 |        2 |        3 |        0 |        0 |
0 |     97 |     97 |      0 |      0 |      0 |           | {.495105}   |           |
  |            |
{1,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100,105,110,115,120,125,130,135,140,145,15
0,155,160,165,170,175,180,185,190,195,200,205,210,215,220,225,230,235,240,245,250,255,260,265,270,
275,280,285,290,295,300,
305,310,315,320,325,330,335,340,345,350,355,360,365,370,375,380,385,390,395,400,405,410,415,420,4
25,430,435,440,445,450,455,460,465,470,475,480,485,490,495,500} |            |           |           |
|             1 |
(1 row)
```

- pg_list_gtt_relfrozenxids()

  描述：显示各会话的冻结事务xid。pid=0的行，显示所有会话中最旧的冻结事务xid。分布式下推荐使用gs_global_list_gtt_relfrozenxids()。

  参数：无。

  返回值类型：record

  示例:

```
gaussdb=# SELECT * FROM pg_list_gtt_relfrozenxids();
       pid       | relfrozenxid
-----------------+--------------
 139988750038784 |        53969
               0 |        53969
(2 rows)
```

- pg_gtt_attached_pid(relOid)

  描述：显示正在使用指定全局临时表的所有线程pid和会话ID。当线程池开启时，如果正在使用全局临时表的会话为detach状态，pid会显示为0，sessionid为会话的ID。分布式下推荐使用gs_gtt_attached_pid(relOid)。

  参数：全局临时表的OID。

  返回值类型：record

  示例：

```
gaussdb=# SELECT * FROM pg_gtt_attached_pid(24576);
 relid |       pid       | sessionid
-------+-----------------+-----------
 24576 | 139988750038784 |         1
(1 row)
```

- gs_gtt_attached_pid(relOid)

  描述：DN上执行，显示正在使用指定分布式全局临时表的所有会话ID。不支持在CN上执行。

版权所有 © 华为云计算技术有限公司

参数：全局临时表的OID。

返回值类型：record

示例：

```
gaussdb=# SELECT * FROM gs_gtt_attached_pid(16384);
 relid |       pid       |   cn_name    | cn_timeline | cn_sequence | cn_session_id
-------+-----------------+--------------+-------------+-------------+---------------
 16384 | 140648560719616 | coordinator1 |           2 |           1 |             1
(1 row)
```

- gs_global_list_gtt_relfrozenxids()

  描述：CN上执行，用于显示各个DN的冻结事务XID信息。当前会话未关联GTT表时，pid为-1。不支持在DN上执行。

  参数：无

  返回值类型：record

  示例：

```
gaussdb=# SELECT * FROM gs_global_list_gtt_relfrozenxids();
 node_name |       pid       | relfrozenxid
-----------+-----------------+--------------
 datanode1 | 140648560719616 |        53959
 datanode1 |               0 |        53959
 datanode2 | 140539996403456 |        53959
 datanode2 |               0 |        53959
(4 rows)
```

- gs_local_gtt_clean_session(clean_type, clean_obj, clean_table)

  描述：DN上执行，用于清理与CN全局会话关联的全局临时表信息。不支持在CN上执行。

  参数：

名称	类型	描述
clean_type	枚举	● 0：清理所有小于当前timeLine的会话映射。 ● 1：清理指定的与该CN会话关联的会话映射。 ● 2：标记清理指定的表在所有会话上的数据（系统内部使用，用户不可执行）
clean_obj	string	构造方式： ● coorname_timeline_seq_sess（clean_type为1） ● coorname_timeline（clean_type为0） ● schema name（clean_type为2）
clean_table	string	clean_type为2时，设置为table name；clean_type为其他类型时，则指定为空值。

  返回值类型：void

  示例：

  首先在CN上使用gs_local_get_cn_session_key获取最新的cn_session_key。

```
gaussdb=# SELECT * FROM gs_local_get_cn_session_key();
   cn_name    | cn_timeline | cn_sequence | cn_session_id
--------------+-------------+-------------+---------------
 coordinator1 |           2 |           1 |             1
(1 row)
```

 版权所有 © 华为云计算技术有限公司

在需要手动清理的DN上使用gs_local_dump_gtt_sess_map获取DN上残留的cn_session_key。

```
gaussdb=# SELECT * FROM gs_local_dump_gtt_sess_map();
   cn_name    | cn_timeline | cn_sequence | cn_session_id | uniq_sess_id | refcnt | database_id |
current_database_id
--------------+-------------+-------------+---------------+--------------+--------+-------------
+---------------------
 coordinator1 |           1 |           1 |             1 |            1 |      0 |       13771 |             13771
(1 row)
```

清理满足如下条件的残留cn_session_key：

- timeline < CN timeline
- refcnt = 0

```
gaussdb=# SELECT * FROM gs_local_gtt_clean_session(0, 'coordinator1_2', '');
 gs_local_gtt_clean_session
----------------------------


(1 row)
```

- gs_global_dist_gtt_clean_remain_sessions()

  描述：CN上执行，清理全局临时表残留的所有会话信息。不支持在DN上执行。

  参数：无

  返回值类型：void

  示例：

```
gaussdb=# SELECT * FROM gs_global_dist_gtt_clean_remain_sessions();
 gs_global_dist_gtt_clean_remain_sessions
------------------------------------------


(1 row)
```

- gs_local_get_cn_session_key()

  描述：CN上执行，显示当前会话的session key。不支持在DN上执行。

  参数：无

  返回值类型：record

  示例：

```
gaussdb=# SELECT * FROM gs_local_get_cn_session_key();
   cn_name    | cn_timeline | cn_sequence | cn_session_id
--------------+-------------+-------------+---------------
 coordinator1 |           2 |           1 |             1
(1 row)
```

- gs_local_dump_gtt_sess_map()

  描述：DN上执行，显示DN上保存的分布式全局临时表会话唯一标识cn_session_key，该标识由如下字段组成：cn_name、cn_timeline、cn_sequence、cn_session_id、uniq_sess_id以及refcnt、database_id、current_database_id。不支持在CN上执行。

  参数：无

  返回值类型：record

  示例：

```
gaussdb=# SELECT * FROM gs_local_dump_gtt_sess_map();
   cn_name    | cn_timeline | cn_sequence | cn_session_id | uniq_sess_id | refcnt | database_id |
current_database_id
--------------+-------------+-------------+---------------+--------------+--------+-------------
+---------------------
 coordinator1 |           1 |           1 |             1 |            1 |      0 |       13771 |             13771
(1 row)
```


版权所有 © 华为云计算技术有限公司

- gs_global_gtt_clean_data(schema_name, table_name)

  描述：标记清理指定的全局临时表在所有会话上的数据文件。系统内部使用，用户不可执行。

  参数：

名称	类型	描述
schema_name	string	需要清理的schema名。
table_name	string	需要清理的表名。

  返回值类型：void

  示例：

  ```
  gaussdb=# SELECT gs_global_gtt_clean_data('public', 'gtt');
  ERROR:  The input parameter is invalid.
  CONTEXT:  referenced column: gs_local_gtt_clean_session
  SQL statement "EXECUTE DIRECT ON DATANODES 'SELECT pg_catalog.gs_local_gtt_clean_session(2, ''public'', ''gtt'')'"
  PL/pgSQL function gs_global_gtt_clean_data(text,text) line 6 at EXECUTE statement
  referenced column: gs_global_gtt_clean_data
  ```

- gs_gtt_local_clean_cn_remain_session_keys()

  描述：CN上执行，用于清理某个DN故障时其他DN上的全局临时表的残留数据，不支持在DN上执行。

  返回值类型：void

  ```
  gaussdb=# select * from gs_gtt_local_clean_cn_remain_session_keys();
   gs_gtt_local_clean_cn_remain_session_keys
  -------------------------------------------


  (1 row)
  ```

1.6.53 DATABASE LINK 函数

- close_database_link(dblink_name name)

  描述：通过DBLink名称主动关闭当前session下该DBLink的连接，不允许关闭存在活跃事务的dblink。

  表 1-290 close_database_link(dblink_name name)参数说明

名称	类型	输入/输出	描述
dblink_name	name	输入	DBLink的名称。
close_database_link	boolean	输出	关闭DATABASE LINK结果（t: 成功，f: 失败）。

  示例：

  ```
  gaussdb=# SELECT close_database_link('xxx');
   close_database_link
  ```


版权所有 © 华为云计算技术有限公司

```
---------------------
 t
(1 row)

gaussdb=# BEGIN;
BEGIN
gaussdb=# SELECT * FROM t1@dblink;
c1 | c2 | c3
----+----+----
1 | |
(1 row)
gaussdb=# INSERT INTO t1@dblink VALUES (2,0,0);
INSERT 0 1
gaussdb=# SELECT * FROM t1@dblink;
c1 | c2 | c3
----+----+----
1 | |
2 | 0 | 0
(2 rows)
gaussdb=# SELECT close_database_link('dblink');
ERROR: Dblink dblink is in use.
CONTEXT: referenced column: close_database_link
gaussdb=# END;
ROLLBACK
```

- gs_ora_diag(dblink_name name DEFAULT NULL::name)

  描述：该函数只能获取连接Oracle库的DBLink对应的信息。通过DBLink名称获取gs_ora_fdw版本、GaussDB版本、Oracle数据库客户端版本以及Oracle数据库服务端版本信息。当参数为NULL时，如果查询用户是管理员用户还会返回ORACLE_HOME、ORACLE_SID、TNS_ADMIN、TWO_TASK、 LDAP_ADMIN等信息，非管理员用户不显示。

  表 1-291 gs_ora_diag(dblink_name name DEFAULT NULL::name)参数说明

名称	类型	输入/输出	描述
dblink_name	name	输入	DBLink的名称。
gs_ora_diag	text	输出	该DBLink的diag信息。

  示例：

```
--管理员用户。
gaussdb=# SELECT gs_ora_diag('xxx');
                                gs_ora_diag
-------------------------------------------------------------------------------
 gs_ora_fdw 1.0.0, GaussDB 9.2.4, Oracle client 19.24.0.0.0, Oracle server 19.0.0.0.0
(1 row)
gaussdb=# SELECT gs_ora_diag();

gs_ora_diag
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 gs_ora_fdw 1.0.0, GaussDB 9.2.4, Oracle client 19.24.0.0.0, ORACLE_HOME=/xxx/instantclient_19_24,
TNS_ADMIN=/xxx/instantclient_19_24/network/admin
(1 row)

--普通用户。
gaussdb=# SELECT gs_ora_diag('xxx');
                                gs_ora_diag
-------------------------------------------------------------------------------
 gs_ora_fdw 1.0.0, GaussDB 9.2.4, Oracle client 19.24.0.0.0, Oracle server 19.0.0.0.0
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT gs_ora_diag();
                      gs_ora_diag
------------------------------------------------------------
 gs_ora_fdw 1.0.0, GaussDB 9.2.4, Oracle client 19.24.0.0.0
(1 row)
```

- gs_ora_get_handle_alloc(dblink_oid oid)

  描述：通过DBLink的OID，获取该DBLink通过OCI使用的内存大小，单位为字节。

  表 1-292 gs_ora_get_handle_alloc(dblink_oid oid)参数说明

名称	类型	输入/输出	描述
dblink_oid	oid	输入	指定的DBLink的OID。
gs_ora_get_handle_alloc	numeric	输出	DBLink通过OCI使用的内存大小，单位为字节。

  示例：

```
gaussdb=# SELECT gs_ora_get_handle_alloc(xxx);
 gs_ora_get_handle_alloc
-------------------------
                  151224
(1 row)
```

- gs_ora_get_handle_alloc_all()

  描述：获取每个节点的环境句柄，所有DBLink通过OCI使用的内存大小计算总和，单位为字节。

  表 1-293 gs_ora_get_handle_alloc_all()参数说明

名称	类型	输入/输出	描述
gs_ora_get_handle_alloc_all	numeric	输出	所有DBLink通过OCI使用的内存大小计算总和，单位为字节。

  示例：

```
gaussdb=# SELECT gs_ora_get_handle_alloc_all();
 gs_ora_get_handle_alloc_all
-----------------------------
                      429000
(1 row)
```

- gs_ora_connection_status(dblink_oid oid)

  描述：通过DBLink的OID，获取该DBLink在当前session下的连接状态。

  表 1-294 gs_ora_connection_status(dblink_oid oid)参数说明

名称	类型	输入/输出	描述
dblink_oid	oid	输入	指定的DBLink的OID。

版权所有 © 华为云计算技术有限公司

名称	类型	输入/输出	描述
gs_ora_connection_status	text	输出	该DBLink在当前session下的连接状态（CONNECTED： 连接， LOST: 未连接）。

示例：

```
gaussdb=# SELECT gs_ora_connection_status(xxx);
 gs_ora_connection_status
------------------------------
 CONNECTED
(1 row)
```

- gs_ora_fdw_handler()

  描述：外部数据封装器oracle_fdw的处理函数，函数功能为返回一个fdw_handler结构体，其中包含指向回调函数的指针。

  入参：无

  返回值类型：fdw_handler

- gs_ora_fdw_validator(option text[], dblink_oid oid)

  描述：外部数据封装器gs_ora_fdw的检查函数，用来检查提供给gs_ora_fdw的通用选项。如果选项或选项值无效或缺失，则报错。

  表 1-295 gs_ora_fdw_validator(option text[], dblink_oid oid)参数说明

名称	类型	输入/输出	描述
dblink_oid	oid	输入	指定的DBLink的OID。
option	text[]	输入	DBLink的OPTION选项。

  返回值类型：void

- is_dblink_in_use(dblink_oid oid)

  描述：通过DBLink的OID，获取该DBLink是否被使用。

  表 1-296 is_dblink_in_use(dblink_oid oid)参数说明

名称	类型	输入/输出	描述
dblink_oid	oid	输入	指定的DBLink的OID。
is_dblink_in_use	bool	输出	该DBLink的使用状态。

  示例：

```
gaussdb=# SELECT is_dblink_in_use(xxx);
 is_dblink_in_use
```


版权所有 © 华为云计算技术有限公司

```
-----------------------------
 f
(1 row)
```

- dblink_has_updatasent (dblink_oid oid)

  描述：判断当前事务中是否已使用OID对应的DATABASE LINK发送了DML语句且未提交。

  表 1-297 dblink_has_updatasent (dblink_oid oid)参数说明

名称	类型	输入/输出	描述
dblink_oid	oid	输入	指定的DBLink的OID。
dblink_has_updatasent	bool	输出	该DBLink是否已发送DML语句且处于未提交阶段。

  示例：

```
gaussdb=# SELECT dblink_has_updatasent(xxx);
 dblink_has_updatasent
-----------------------------
 f
(1 row)
```

- is_dblink_in_transaction(dblink_oid oid)

  描述：通过DBLink的OID，获取该DBLink是否正在事务中。

  表 1-298 is_dblink_in_transaction(dblink_oid oid)参数说明

名称	类型	输入/输出	描述
dblink_oid	oid	输入	指定的DBLink的OID。
is_dblink_in_transaction	bool	输出	该DBLink是否在事务中。

  示例：

```
gaussdb=# SELECT is_dblink_in_transaction(xxx);
 is_dblink_in_transaction
-----------------------------
 f
(1 row)
```

1.6.54 GSMonitor 系统资源统计函数

- gs_get_cpustat(start_time timestamptz, end_time timestamptz)

  描述：查询一段时间内CPU资源信息的接口，调用该函数必须具有MONADMIN权限或者SYSADMIN权限。

  参数说明：

版权所有 © 华为云计算技术有限公司

参数名称	参数说明	取值范围
start_time	必选。查询起始时间。	timestamptz类型。
end_time	必选。查询终止时间。	timestamptz类型。

返回值类型：setof record

返回字段说明：

字段名称	字段类型	字段说明
collection_time	timestamptz	采集时间。
cpu_name	text	名称。
%total	float8	总：用户+内核占CPU百分比。
%user	float8	用户态：用户空间占CPU百分比。
%system	float8	系统态：内核空间占CPU百分比。
%idle	float8	空闲CPU百分比。
%iowait	float8	等待IO的CPU百分比。
lavg_1	float8	1分钟负载（单位：个）。
lavg_5	float8	5分钟负载（单位：个）。
lavg_15	float8	15分钟负载（单位：个）。

- gs_get_procstat(start_time timestamptz, end_time timestamptz, sort_order text[, display_type text])

  描述：查询一段时间内进程相关信息的接口，调用该函数必须具有MONADMIN权限或者SYSADMIN权限。

  参数说明：

参数名称	参数说明	取值范围
start_time	必选。查询起始时间。	timestamptz类型。
end_time	必选。查询终止时间。	timestamptz类型。

版权所有 © 华为云计算技术有限公司

参数名称	参数说明	取值范围
sort_order	必选。排序规则。	text类型。取值范围为'IO_R'、'IO_W'、'CPU'、'MEM'（不区分大小写）。 ● 'IO_R'：按照IO读速率进行降序排序。 ● 'IO_W'：按照IO写速率进行降序排序。 ● 'CPU'：按照CPU利用率进行降序排序。 ● 'MEM'：按照内存使用大小进行降序排序。
display_type	可选。展示方式。	text类型。当前只支持'pretty'（不区分大小写），表示显示隶属关系。

返回值类型：setof record

返回字段说明：

字段名称	字段类型	字段说明
collection_time	timestamptz	采集时间。
pid	int8	进程名称。
tid	int8	线程名称。
state	text	状态。
ppid	int8	父进程ID。
priority	int8	优先级（动态）。
nice	int8	优先级（静态）。
num_threads	int8	线程数量。
utime	float8	用户态运行时间（单位：jiffies 时间片）。
stime	float8	核心态运行时间（单位：jiffies 时间片）。
ssize	int8	共享内存大小（单位：字节）。
vsize	int8	进程虚拟内存大小（单位：字节）。
io_r	float8	IO读速率（单位：次/s）。
io_w	float8	IO写速率（单位：次/s）。
io	float8	IO总速率（单位：次/s）。
rss	int8	进程使用的驻留集大小（单位：页）。
%cpu_usage	float8	CPU使用率。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	字段说明
comm	text	进程的命令名。

- gs_get_iostat(start_time timestamptz, end_time timestamptz[, display_type text])

  描述：查询一段时间内IO资源信息的接口，调用该函数必须具有MONADMIN权限或者SYSADMIN权限。

  参数说明：

参数名称	参数说明	取值范围
start_time	必选。查询起始时间。	timestamptz类型。
end_time	必选。查询终止时间。	timestamptz类型。
display_type	可选。展示方式。	text类型。当前只支持'pretty'（不区分大小写），表示显示隶属关系。

  返回值类型：setof record

  返回字段说明：

字段名称	字段类型	字段说明
collection_time	timestamptz	采集时间。
devname	text	设备名称。
rkB/s	float8	IO读速率（单位：kB/s）。
wkB/s	float8	IO写速率（单位：kB/s）。
r/s	float8	每秒读取磁盘的IO操作次数（单位：次/s）。
w/s	float8	每秒写入磁盘的IO操作次数（单位：次/s）。
rawait	float8	读取操作的平均等待时间（单位：毫秒）。
wawait	float8	写入操作的平均等待时间（单位：毫秒）。
svctm	float8	IO平均服务时间（单位：毫秒）。
avgqu-sz	int8	当前IO等待队列数。
%util	float8	磁盘使用率：设备花费在IO操作的时间百分比。

- gs_get_memstat(start_time timestamptz, end_time timestamptz)

  描述：查询一段时间内内存资源信息的接口，调用该函数必须具有MONADMIN权限或者SYSADMIN权限。

  参数说明：

版权所有 © 华为云计算技术有限公司

参数名称	参数说明	取值范围
start_time	必选。查询起始时间。	timestamptz类型。
end_time	必选。查询终止时间。	timestamptz类型。

返回值类型：setof record

返回字段说明：

字段名称	字段类型	字段说明
collection_time	timestamptz	采集时间。
mem_total	int8	内存总大小（单位：字节）。
mem_used	int8	已使用内存大小（单位：字节）。
mem_free	int8	空闲内存大小（单位：字节）。
mem_usage	float8	内存使用率。
buffer	int8	缓冲区占内存大小（单位：字节）。
cache	int8	缓存栈内存大小（单位：字节）。
mem_available	int8	可供启动新应用程序的内存大小（单位：字节）。
active	int8	活跃使用高速缓冲内存的大小（单位：字节）。
inactive	int8	不经常使用高速缓冲内存的大小（单位：字节）。
swap_total	int8	交换空间总大小（单位：字节）。
swap_free	int8	空闲交换空间大小（单位：字节）。

- gs_get_netstat(start_time timestamptz, end_time timestamptz)

  描述：查询一段时间内网络资源信息的接口，调用该函数必须具有MONADMIN权限或者SYSADMIN权限。

  参数说明：

参数名称	参数说明	取值范围
start_time	必选。查询起始时间。	timestamptz类型。
end_time	必选。查询终止时间。	timestamptz类型。

版权所有 © 华为云计算技术有限公司

返回值类型：setof record

返回字段说明：

字段名称	字段类型	字段说明
collection_time	timestamptz	采集时间。
interface	text	接口名称。
tag	text	标签。
rkb/s	float8	接收速率（Kbps）。
tkb/s	float8	发送速率（Kbps）。
kb/s	float8	总速率（Kbps）。
rp/s	float8	接收包速率（个/s）。
tp/s	float8	发送包速率（个/s）。
p/s	float8	数据包总速率（个/s）。
rerr/s	float8	接收包错误速率（个/s）。
terr/s	float8	发送包错误率（个/s）。
%err	float8	总错误率。

1.6.55 排序计算函数

排序计算函数提供基本的排序场景计算能力。

表 1-299 排序计算支持的函数一览

功能	函数
对给定的列按照时间序列排序并返回第一个值。	**first(value_array anyarray, time_array timearray)**
对给定的列按照时间序列排序并返回最后一个值。	**last(value_array anyarray, time_array timearray)**
对给定的列按照序列排序并返回百分位值。	**percentile_cont(percentile float)**
对给定的列按照从小到大的顺序排序并返回百分位值。	**value_of_percentile(column float，percentile float, compression float)**

版权所有 © 华为云计算技术有限公司

功能	函数
对给定的列按照从小到大的顺序排序，根据百分位值计算百分位。	percentile_of_value(column float, percentilevalue float, compression float)
对给定列排序后最小的K个元组。	top_k(array real[],k int)
对给定列排序后最大的K个元组。	bottom_k(array real[],k int)

first(value_array anyarray, time_array timearray)

描述：对给定的列按照时间序列排序并返回第一个值。

参数：参数说明如表1-300所示。

返回值类型：anyelement

表 1-300 first 函数参数说明

参数名	类型	描述
value_array	anyarray	查询列的数组。
time_array	timearray	时间值的数组。

说明

- 函数内部使用INTEGER类型作数组索引，不支持数据量超过INTEGER类型所支持的数值范围。

示例：

```
gaussdb=# SELECT
gaussdb-#     first(array_agg(value), array_agg(time))
gaussdb-# FROM
gaussdb-#     (VALUES ('2019-07-12 00:00:00'::timestamptz,'c','a'),
gaussdb(#             ('2019-07-12 00:01:00'::timestamptz,'b','a'),
gaussdb(#             ('2019-07-12 00:02:01'::timestamptz,'a','b')) v(time, value, tag)
gaussdb-# GROUP BY tag;
 first
-------
 c
 a
(2 rows)
```

last(value_array anyarray, time_array timearray)

描述：对给定的列按照时间序列排序并返回最后一个值。

参数：参数说明如表1-301所示。

返回值类型：anyelement

版权所有 © 华为云计算技术有限公司

表 1-301 last 函数参数说明

参数名	类型	描述
value_array	anyarray	查询列的数组。
time_array	timearray	时间值的数组。

说明

- 函数内部使用INTEGER类型作数组索引，不支持数据量超过INTEGER类型所支持的数值范围。

示例：

```
gaussdb=# SELECT
gaussdb-#     last(array_agg(value), array_agg(time))
gaussdb-# FROM
gaussdb-#     (VALUES ('2019-07-12 00:00:00'::timestamptz,'c','a'),
gaussdb(#             ('2019-07-12 00:01:00'::timestamptz,'b','a'),
gaussdb(#             ('2019-07-12 00:02:01'::timestamptz,'a','b')) v(time, value, tag)
gaussdb-# GROUP BY tag;
 last
------
 b
 a
(2 rows)
```

percentile_cont(percentile float)

描述：对给定的列按照序列排序并返回百分位值。需要与within group (order by) 结合使用，指定分位值的列，且该列类型需要为数值型或interval。

参数：参数说明如**表1-302**所示。

返回值类型：float或interval，取决于within group (order by)指定列的值为数值型或interval。

表 1-302 percentile_cont 函数参数说明

参数名	类型	描述
percentile	float	指定百分位，取值范围[0，1]，精度为浮点值，例如0.95为95分位值。

说明

- percentile为[0,1]之间的小数，精度为浮点值类型，不支持95%写法。

示例：

```
gaussdb=# SELECT
gaussdb-#     percentile_cont(0.5) WITHIN GROUP (ORDER BY value)
gaussdb-# FROM
gaussdb-#     (VALUES ('2019-07-12 00:00:01'::timestamptz,'red',1),
```


版权所有 © 华为云计算技术有限公司

```
gaussdb(#     ('2019-07-12 00:00:01'::timestamptz,'blue',2)) v(time,color,value);
 percentile_cont
-----------------
             1.5
(1 row)
```

value_of_percentile(column float，percentile float, compression float)

描述：对给定的列按照从小到大的顺序排序并返回百分位值。

参数：参数说明如**表1-303**所示。

返回值类型：float

表 1-303 value_of_percentile 函数参数说明

参数名	类型	描述
column	float	指定列名。
percentile	float	指定百分位，取值范围[0，1]，例如0.95为95分位值。
compression	float	指定压缩系数，取值范围[0，500]，系统默认为300，若用默认值，这个参数位置写0，否则按照取值范围自行填写，例如300则按照300对总体数据的数量进行压缩，数据越大，占用内存越大。

说明

- 需要指定列名。
- percentile为[0,1]之间的小数，精度为浮点值类型，不支持95%写法。
- compression为[0,500]之间的数，默认值为300，若输入值大于500或者小于0，则按照默认值处理。
- 建议在满足如下约束的场景下使用，在满足约束条件的前提下，大部分场景准确度大于75%：

  (1) 整体数据符合正态分布。

  (2) 每个DN上的数据分布均匀，也符合正态分布。

  (3) 数据量需要大于一万。

  (4) 只适合两端数据，对于极端数据百分位小于1%和大于99%的数据。

示例：

```
gaussdb=#DROP TABLE IF EXISTS partition_scan2;
gaussdb=#CREATE TABLE partition_scan2(a int);
gaussdb=#insert into partition_scan2 values(generate_series(1,100,1));
gaussdb=# SELECT value_of_percentile(a,0.8,0) from partition_scan2;
 value_of_percentile
---------------------
                80.5
(1 row)
```


版权所有 © 华为云计算技术有限公司

percentile_of_value(column float，percentilevalue float, compression float)

描述：对给定的列按照从小到大的顺序排序并返回百分位。

参数：参数说明如表1-304所示。

返回值类型：float

表 1-304 percentile_of_value 函数参数说明

参数名	类型	描述
column	float	指定列名。
percentilevalue	float	指定百分位值，根据百分位值计算百分位，精度为浮点值。
compression	float	指定压缩系数，取值范围[0，500]，精度为浮点值，系统默认为300，若用默认值，这个参数位置写0，否则按照取值范围自行填写，例如300则按照300对总体数据的数量进行压缩，数据越大，占用内存越大。

说明

- 需要指定列名。
- percentilevalue任意值，精度为浮点值类型。
- compression为[0,500]之间的数，默认值为300，若输入值大于500或者小于0，则按照默认值处理。
- 建议在满足如下约束的场景下使用，在满足约束条件的前提下，大部分场景准确度大于75%：
  (1) 整体数据符合正态分布。
  (2) 每个DN上的数据分布均匀，也符合正态分布。
  (3) 数据量需要大于一万。
  (4) 只适合两端数据，对于极端数据百分位小于1%和大于99%的数据。

示例：

```
gaussdb=#DROP TABLE IF EXISTS partition_scan2;
gaussdb=#CREATE TABLE partition_scan2(a int);
gaussdb=#insert into partition_scan2 values(generate_series(1,100,1));
gaussdb=# SELECT percentile_of_value(a,80,0) from partition_scan2;
 percentile_of_value
---------------------
                .795
(1 row)
```

top_k(array real[],k int)

描述：对给定的列按照从小到大排序返回最小的K个元组。

参数：参数说明如表1-305所示。

返回值类型：real[]

版权所有 © 华为云计算技术有限公司

表 1-305 top_k 函数参数说明

参数名	类型	描述
array	real[]	输入的数组。
k	int	指定最小的K个元组。

说明

- k应为非负数，若为负数，会返回空值。
- k应小于array大小，若大于当前数组长度，报错。
- 内部使用array_sort排序，仅支持一维数组。请参见**数组函数和操作符**章节中介绍。
- array不能为空，若为空函数直接返回空。
- 不支持对数据量超过int类型取值范围的数组使用。

示例：

```
gaussdb=# SELECT
gaussdb-#       top_k(array_agg(v), 0.5),v2
gaussdb-# FROM
gaussdb-#       (VALUES (1,'a'),(2,'a'),(3,'a'),(4,'a'),(5,'a'),(6,'a')) v(v,v2)
gaussdb-# GROUP BY v2;
 top_k | v2
-------+----
 {1}   | a
(1 row)
```

bottom_k(array real[],k int)

描述：对给定的列按照从小到大排序并返回最大的K个元组。

参数：参数说明如**表1-306**所示。

返回值类型：real[]

表 1-306 bottom_k 函数参数说明

参数名	类型	描述
k	int	指定最大的K个元组。

说明

- k应为非负数，若为负数，会返回空值。
- k应小于array大小，若大于当前数组长度，报错。
- 内部使用array_sort排序，仅支持一维数组。请参见**数组函数和操作符**章节中介绍。
- array不能为空，若为空函数直接返回空。
- 不支持对数据量超过int类型取值范围的数组使用。

示例：

```
gaussdb=# SELECT
gaussdb-#       bottom_k(array_agg(v), 0.5),v2
```


版权所有 © 华为云计算技术有限公司

```
gaussdb-# FROM
gaussdb-#      (VALUES (1,'a'),(2,'a'),(3,'a'),(4,'a'),(5,'a'),(6,'a')) v(v,v2)
gaussdb-# GROUP BY v2;
 bottom_k | v2
----------+----
 {6}      | a
(1 row)
```

1.6.56 自治事务特性函数

- gs_autonomous_transaction_detail()

  描述：查看当前CN节点上用户所属的自治事务会话及其父子事务关系。具备MONADMIN权限或者SYSADMIN权限的用户，可以查看当前CN节点所有自治事务会话及其父子事务关系。不允许在DN节点调用。

  参数：无。

  返回值类型：setof record

  返回值说明：如**表1-307**所示。

  表 1-307 gs_autonomous_transaction_detail 函数返回值说明

名称	类型	描述
autonomous_transaction_session_id	bigint	自治事务会话ID。
autonomous_transaction_thread_id	bigint	自治事务线程ID。
parent_autonomous_transaction_session_id	bigint	自治事务父会话ID。

  示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_test()  AS
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
pg_sleep(60);
END;
/
gaussdb=# CALL proc_test();

-- 需要新建一个会话进行查询 --
gaussdb=# SELECT * FROM pg_catalog.gs_autonomous_transaction_detail() AS
gaussdb-#      (autonomous_transaction_session_id bigint, autonomous_transaction_thread_id bigint,
parent_autonomous_transaction_session_id bigint);
 autonomous_transaction_session_id | autonomous_transaction_thread_id |
parent_autonomous_transaction_session_id
-----------------------------------+----------------------------------+-----------------------------------------
                  139778155607808 |                  139778155607808 |                    139777830549248
(1 row)
```

- gs_get_autonomous_transaction_count()

  描述：查询当前CN节点的自治事务计数。不允许在DN节点调用。

  参数：无。

  返回值类型：integer

版权所有 © 华为云计算技术有限公司

返回值说明：如表1-308所示。

表 1-308 gs_get_autonomous_transaction_count 函数返回值说明

名称	类型	描述
gs_get_autonomous_transaction_count	integer	当前自治事务计数。

示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_test()  AS
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
pg_sleep(60);
END;
/
gaussdb=# CALL proc_test();

-- 需要新建一个会话进行查询 --
gaussdb=# SELECT gs_get_autonomous_transaction_count();
 gs_get_autonomous_transaction_count
-------------------------------------
                                   1
(1 row)
```

- gs_reset_autonomous_transaction_count()

  描述：将当前CN节点的自治事务计数恢复至正确值，并返回当前正确的自治事务计数值。用户需拥有MONADMIN权限或者SYSADMIN权限才可调用该函数。不允许在DN节点调用。

  参数：无。

  返回值类型：integer

  返回值说明：如表1-309所示。

  表 1-309 gs_reset_autonomous_transaction_count 函数返回值说明

名称	类型	描述
gs_reset_autonomous_transaction_count	integer	恢复后正确的自治事务计数值。

  示例：

  ```
  gaussdb=# SELECT gs_reset_autonomous_transaction_count();
   gs_reset_autonomous_transaction_count
  ---------------------------------------
                                       0
  (1 row)

  gaussdb=#
  CREATE OR REPLACE PROCEDURE proc_test()  AS
  DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
  BEGIN
  pg_sleep(60);
  ```


版权所有 © 华为云计算技术有限公司

```
END;
/
gaussdb=# CALL proc_test();

-- 需要新建一个会话进行查询 --
gaussdb=# SELECT gs_reset_autonomous_transaction_count();
 gs_reset_autonomous_transaction_count
---------------------------------------
                                     1
(1 row)
```

- gs_global_autonomous_transaction_detail()

  描述：查看每个CN节点上用户所属的自治事务会话及其父子事务关系。具备MONADMIN权限或者SYSADMIN权限的用户，可以查看每个CN节点所有自治事务会话及其父子事务关系。不允许在DN节点调用。

  参数：无。

  返回值类型：setof record

  返回值说明：如**表1-310**所示。

  说明

  当GUC参数enable_nonsysadmin_execute_direct=off时，该函数仅允许具备MONADMIN权限或者SYSADMIN权限的用户调用。

  表 1-310 gs_global_autonomous_transaction_detail 函数返回值说明

名称	类型	描述
node_name	text	CN节点名称。
autonomous_transaction_session_id	bigint	自治事务会话ID。
autonomous_transaction_thread_id	bigint	自治事务线程ID。
parent_autonomous_transaction_session_id	bigint	自治事务父会话ID。

  示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_test()  AS
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
pg_sleep(60);
END;
/
gaussdb=# CALL proc_test();

-- 需要新建一个会话进行查询 --
gaussdb=# SELECT * FROM gs_global_autonomous_transaction_detail();
  node_name   | autonomous_transaction_session_id | autonomous_transaction_thread_id |
parent_autonomous_transaction_session_id
--------------+-----------------------------------+----------------------------------
+-----------------------------------------
 coordinator1 |                 140477506516736 |                 140477506516736 |
140477683726080
(1 row)
```


版权所有 © 华为云计算技术有限公司

- gs_global_get_autonomous_transaction_count()

  描述：查询每个CN节点当前的自治事务计数。不允许在DN节点调用。

  参数：无。

  返回值类型：setof record

  返回值说明：如表1-311所示。

  说明

  当GUC参数enable_nonsysadmin_execute_direct=off时，该函数仅允许具备MONADMIN权限或者SYSADMIN权限的用户调用。

  表 1-311 gs_global_get_autonomous_transaction_count 函数返回值说明

名称	类型	描述
node_name	text	CN节点名称。
gs_get_autonomous_transaction_count	integer	当前自治事务计数。

  示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_test()  AS
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
pg_sleep(60);
END;
/
gaussdb=# CALL proc_test();

-- 需要新建一个会话进行查询 --
gaussdb=# SELECT * FROM gs_global_get_autonomous_transaction_count();
  node_name   | gs_get_autonomous_transaction_count
--------------+-------------------------------------
 coordinator1 |                                   1
(1 row)
```

- gs_global_reset_autonomous_transaction_count([exec_node_name text])

  描述：将每个CN节点的自治事务计数恢复至正确值，并返回恢复后每个CN节点正确的自治事务计数值。若指定参数exec_node_name，则恢复指定节点的自治事务计数为正确值。调用该函数必须有MONADMIN权限或者SYSADMIN权限。不允许在DN节点调用。

  参数：exec_node_name，指定的CN节点名称，可选参数。

  返回值类型：setof record

  返回值说明：如表1-312所示。

  表 1-312 gs_global_reset_autonomous_transaction_count 函数返回值说明

名称	类型	描述
node_name	text	CN节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
gs_reset_autonomous_transaction_count	integer	恢复后正确的自治事务计数值。

示例：

```
gaussdb=#
CREATE OR REPLACE PROCEDURE proc_test()  AS
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
pg_sleep(60);
END;
/
gaussdb=# CALL proc_test();

-- 需要新建一个会话进行查询 --
gaussdb=# SELECT * FROM gs_global_reset_autonomous_transaction_count();
  node_name   | gs_reset_autonomous_transaction_count
--------------+---------------------------------------
 coordinator1 |                                     1
(1 row)

gaussdb=# SELECT * FROM gs_global_reset_autonomous_transaction_count('coordinator1');
  node_name   | gs_reset_autonomous_transaction_count
--------------+---------------------------------------
 coordinator1 |                                     1
(1 row)
```

1.6.57 DSL 规则管理函数

- dsl_add_rule()

  描述：新增一条DSL规则，该函数仅限初始用户调用。

  参数：参数说明如**表1-313**所示。

  返回值类型：text

  返回值说明：DSL规则NAME。

  表 1-313 dsl_add_rule 函数参数说明

入参名称	入参类型	是否必选	参数描述
rulename	name	Y	DSL规则名称。格式要求：字母开头，不支持中文；支持字母、数字、下划线、中划线组合；不支持其他特殊字符；长度不超过64个字节。
ruletext	text	Y	DSL规则文本内容。
priority	integer	Y	DSL规则优先级（取值范围：1-9999）,全局唯一， 数字越大优先级越低。

版权所有 © 华为云计算技术有限公司

入参名称	入参类型	是否必选	参数描述
enabled	boolean	N	是否开启规则，默认值为：FALSE。
ruledesc	name	N	DSL规则描述，可以为空值，长度不超过64个字节。
ruledbcompatibility	text	N	DSL规则在数据库的兼容模式。 支持类型：ORA、MYSQL、TD、PG、M、ANY； 分别表示兼容O、MY、TD、POSTGRES、M-Compatibility和全部兼容； 支持设置多个兼容模式，兼容模式间用 | 隔开，比如A|PG表示同时支持O和PG兼容； 默认值：ANY。
planmode	text	N	规则支持的计划类型，可选：Stream/RemoteQuery/ANY；默认值：ANY。

示例：

```
gaussdb=# SELECT dsl_add_rule(
'compute_const',
'Filter<> {args: $expr: *}
=>
Filter<> {args: (expr_compute $expr)
}',
3,
true);
 dsl_add_rule
---------------
 compute_const
(1 row)
```

- dsl_del_rule()

  描述：删除一条DSL规则，该函数仅限系统管理员用户及初始用户调用。

  参数：参数说明如**表1-314**所示。

  返回值类型：integer

  返回值说明：删除DSL规则的条数。

  表 1-314 dsl_del_rule 函数参数说明

入参名称	入参类型	入参描述
rulename	name	DSL规则NAME。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT dsl_del_rule('compute_const');
 dsl_del_rule
--------------
            1
(1 row)
```

- dsl_set_rule()

  描述：修改一条DSL规则（修改单个属性值），不允许修改规则本身的内容，仅系统管理员用户和初始用户使用。

  参数：参数说明如**表1-315**所示。

  返回值类型：integer

  返回值说明：修改DSL规则的条数。

  表 1-315 dsl_set_rule 函数参数说明

入参名称	入参类型	入参描述
rulename	name	DSL规则NAME。
attrname	text	属性名称，目前支持的属性有：enabled，priority，ruledesc，dbcompatibility，planmode。
attrval	text	属性值。

  示例：

```
gaussdb=# SELECT dsl_set_rule('compute_const', 'priority', '8');
 dsl_set_rule
--------------
            1
(1 row)

gaussdb=# \x
Expanded display is on.

gaussdb=# SELECT * FROM GS_DSL_REWRITERULE WHERE rulename = 'compute_const';
-[ RECORD 1 ]---+-----------------------------------
enabled         | t
externalrule    | t
costbased       | f
priority        | 8  --规则优先级被设置为8
dbcompatibility | 31
planmode        | 3
createtime      | xxxx-xx-xx 16:43:18.766029+08
lastupdatetime  | xxdsl_reload_rule函数参数说明xx-xx-xx 16:44:55.785637+08
rulename        | compute_const
rulehash        | 8576886722021209825
ruletext        | Filter<> {args: $expr: *}
                | =>
                | Filter<> {args: (expr_compute $expr)
                | }
ruledesc        |
```

- dsl_reload_rule()

  描述：加载DSL规则到缓存。

  参数：参数说明如**表1-316**所示。

  返回值类型：text

  返回值说明：success表示加载成功，failed表示加载失败。

版权所有 © 华为云计算技术有限公司

表 1-316 dsl_reload_rule 函数参数说明

入参名称	入参类型	入参描述
nodename	text	节点名称，为空表示在全部节点执行。

示例：

```
gaussdb=# SELECT dsl_reload_rule();
 dsl_reload_rule
-----------------
 success
(1 row)
```

1.6.58 ROWID 类型函数

- rowid_tableoid(rowid)

  描述：获取rowid的table_oid值。

  参数：rowid

  返回值类型：oid

  示例：

  ```
  gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
  CREATE DATABASE
  gaussdb=# \c ora_test
  Non-SSL connection (SSL connection is recommended when requiring high-security)
  You are now connected to database "ora" as user "$user".
  ora_test=# SELECT rowid_tableoid('AAAAAAAAAP//BBBBBAAAAAAAAAAAAC');
   rowid_tableoid
  ----------------
      1090785344
  (1 row)
  ora_test=# \c postgres
  Non-SSL connection (SSL connection is recommended when requiring high-security)
  You are now connected to database "postgres" as user "$user".
  gaussdb=# DROP DATABASE ora_test;
  DROP DATABASE
  ```

- rowid_sequence(rowid)

  描述：获取rowid的row_no值。

  参数：rowid

  返回值类型：int8

  示例：

  ```
  gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
  CREATE DATABASE
  gaussdb=# \c ora_test
  Non-SSL connection (SSL connection is recommended when requiring high-security)
  You are now connected to database "ora" as user "$user".
  ora_test=# SELECT rowid_sequence('AAAAAAAAAP//BBBBBAAAAAAAAAAAAC');
   rowid_sequence
  ----------------
               2
  (1 row)
  ora_test=# \c postgres
  Non-SSL connection (SSL connection is recommended when requiring high-security)
  You are now connected to database "postgres" as user "$user".
  gaussdb=# DROP DATABASE ora_test;
  DROP DATABASE
  ```


版权所有 © 华为云计算技术有限公司

- rowid_nodeid(rowid)

  描述：获取rowid的node_slice值。

  参数：rowid

  返回值类型：int2

  示例：

```
gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c ora_test
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora" as user "$user".
ora_test=# SELECT rowid_nodeid('AAAAAAAAAP//BBBBBAAAAAAAAAAAC');
 rowid_nodeid
--------------
            0
(1 row)
ora_test=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "$user".
gaussdb=# DROP DATABASE ora_test;
DROP DATABASE
```

- rowid_hashbucketid(rowid)

  描述：获取rowid的bucket_id值。

  参数：rowid

  返回值类型：int2

  示例：

```
gaussdb=# CREATE DATABASE ora_test DBCOMPATIBILITY='ORA';
CREATE DATABASE
gaussdb=# \c ora_test
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora" as user "$user".
ora_test=# SELECT rowid_hashbucketid('AAAAAAAAAP//BBBBBAAAAAAAAAAAC');
 rowid_hashbucketid
--------------------
                 -1
(1 row)
ora_test=# \c postgres
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "$user".
gaussdb=# DROP DATABASE ora_test;
DROP DATABASE
```

1.6.59 失效重编译函数

- transform_view_dep_source()

  描述：刷新**GS_DEP_SOURCE**系统表的source字段。即将视图定义语句中SELECT *的星号扩展；将视图创建时对象使用的schema补充写入；刷新视图依赖关系。该函数仅支持管理员用户调用。

  参数：无

  返回值类型：void

- gs_compile_schema(schema_name in varchar2 default null, compile_all in boolean default false, retry_times in int default 10)

  描述：重编译指定命名空间中的包、函数、存储过程和视图。

  参数：参数说明如**表1-317**所示。

  返回值类型：boolean，true表示重编译成功，false表示重编译失败。

版权所有 © 华为云计算技术有限公司

表 1-317 gs_compile_schema 参数说明

参数	类型	入参/出参	是否可以为空	描述
schema_name	VARCHAR2	IN	是	命名空间名称。
compile_all	BOOLEAN	IN	是	是否对全部对象进行重编译，包括当前处于有效状态的视图。 ● false：编译PG_OBJECT表中失效状态的视图。 ● true：编译PG_OBJECT表中所有的视图。
retry_times	INT	IN	是	重编译失败，重试次数。

示例：

```
-- 开启失效重编译参数。
gaussdb=# SET enable_view_invalidation='on';

-- 创建schema、表，视图依赖表，视图依赖视图。
gaussdb=# DROP SCHEMA IF EXISTS tesch2 CASCADE;
gaussdb=# CREATE SCHEMA tesch2;
gaussdb=# SET current_schema = tesch2;
gaussdb=# CREATE TABLE tab1 (c1 INT, c2 INT);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE VIEW v_tab1 AS SELECT * FROM tab1;
gaussdb=# CREATE VIEW v_tab2 AS SELECT * FROM v_tab1;
gaussdb=# CREATE VIEW v_tab3 AS SELECT * FROM tab1;

-- 删除表。
gaussdb=# DROP TABLE tab1;

-- 查询对象有效性。
gaussdb=# SELECT c.relname, o.valid FROM pg_catalog.pg_object o JOIN pg_catalog.pg_class c ON
c.oid = o.object_oid WHERE c.relname = 'v_tab1' or c.relname = 'v_tab2' or c.relname = 'v_tab3';
 relname | valid
---------+-------
 v_tab1  | f
 v_tab2  | f
 v_tab3  | f
(3 rows)

-- 恢复删除的表。
gaussdb=# CREATE TABLE tab1 (c1 INT, c2 INT);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

-- 重编译tesch2命名空间下视图对象。
gaussdb=# CALL pg_catalog.gs_compile_schema('tesch2', true, 1);
INFO:  successful
CONTEXT:  referenced column: gs_compile_schema
SQL statement "select pkg_util.gs_compile_schema(:schema_name, :compile_all, :retry_times);"
PL/pgSQL function gs_compile_schema(character varying,boolean,integer) line 24 at EXECUTE
statement
INFO:  The number of executions is 1
INFO:  successful
```


版权所有 © 华为云计算技术有限公司

```
 gs_compile_schema
-------------------
 t
(1 row)

-- 再次查询视图有效性。
gaussdb=# SELECT c.relname, o.valid FROM pg_catalog.pg_object o JOIN pg_catalog.pg_class c ON
c.oid = o.object_oid WHERE c.relname = 'v_tab1' or c.relname = 'v_tab2' or c.relname = 'v_tab3';
 relname | valid
---------+-------
 v_tab1  | t
 v_tab2  | t
 v_tab3  | t
(3 rows)
-- 删除视图。
gaussdb=# DROP VIEW v_tab3;
gaussdb=# DROP VIEW v_tab2;
gaussdb=# DROP VIEW v_tab1;
-- 删除表。
gaussdb=# DROP TABLE tab1;
gaussdb=# RESET current_schema;
-- 删除schema。
gaussdb=# DROP SCHEMA tesch2 CASCADE;
```

1.6.60 其他系统函数

其他系统函数包含两类，兼容PostgreSQL的函数和实现内部功能的函数。这些函数不推荐使用，若需使用，请联系华为技术支持工程师。

兼容 PostgreSQL 的函数和操作符

- 下述列表为GaussDB的内建函数和操作符兼容PostgreSQL，函数详细说明请参考PostgreSQL官方文档。

_pg_char_max_length	_pg_char_octet_length	_pg_datetime_precision	_pg_expandarray	_pg_index_position	_pg_interval_type	_pg_numeric_precision
_pg_numeric_precision_radix	_pg_numeric_scale	_pg_truetypid	_pg_truetypmod	q	abs	abstime
abstimeeq	abstimege	abstimegt	abstimein	abstimele	abstimelt	abstimene
abstimeout	abstimerecv	abstimesend	aclcontains	acldefault	aclexplode	aclinsert
aclitemeq	aclitemin	aclitemout	aclremove	acos	age	akeys
any_in	any_out	anyarray_in	anyarray_out	anyarray_recv	anyarray_send	anyelement_in
anyelement_out	anyenum_in	anyenum_out	anynonarray_in	anynonarray_out	anyrange_in	anyrange_out

版权所有 © 华为云计算技术有限公司

anytextcat	area	areajoinsel	areasel	array_agg	array_agg_finalfn	array_agg_transfn
array_append	array_cat	array_dims	array_eq	array_fill	array_ge	array_gt
array_in	array_larger	array_le	array_length	array_lower	array_lt	array_ndims
array_ne	array_out	array_prepend	array_recv	array_send	array_smaller	array_to_json
array_to_string	array_typanalyze	array_upper	arraycontained	arraycontains	arraycontjoinsel	arraycontsel
arrayoverlap	ascii	asin	atan	atan2	avals	avg
big5_to_euc_tw	big5_to_mic	big5_to_utf8	bit	bit_and	bit_in	bit_length
bit_or	bit_out	bit_recv	bit_send	bitand	bitcat	bitcmp
biteq	bitge	bitgt	bitle	bitlt	bitne	bitnot
bitor	bitshiftleft	bitshiftright	bittypmodin	bittypmodout	bitxor	bool
bool_and	bool_or	booland_statefunc	booleq	boolge	boolgt	boolin
boolle	boollt	boolne	boolor_statefunc	boolout	boolrecv	boolsend
box	box_above	box_above_eq	box_add	box_below	box_below_eq	box_center
box_contain	box_contain_pt	box_contained	box_distance	box_div	box_eq	box_ge
box_gt	box_in	box_intersect	box_le	box_left	box_lt	box_mul
box_out	box_overabove	box_overbelow	box_overlap	box_overleft	box_overright	box_recv
box_right	box_same	box_send	box_sub	bpchar	bpchar_larger	bpchar_pattern_ge

版权所有 © 华为云计算技术有限公司

bpchar_pattern_gt	bpchar_pattern_le	bpchar_pattern_lt	bpchar_smaller	bpchar_sortsupport	bpcharcmp	bpchareq
bpcharge	bpchargt	bpcharlclike	bpcharlcnlike	bpcharlcregexeq	bpcharlcregexne	bpcharin
bpcharle	bpcharlike	bpcharlt	bpcharne	bpcharnlike	bpcharout	bpcharrecv
bpcharregexeq	bpcharregexne	bpcharsend	bpchartypmodin	bpchartypmodout	broadcast	btabstimecmp
btarraycmp	btbeginscan	btboolcmp	btbpchar_pattern_cmp	btbuild	btbuildempty	btbulkdelete
btcanreturn	btcharcmp	btcostestimate	btendscan	btfloat48cmp	btfloat4cmp	btfloat4sortsupport
btfloat84cmp	btfloat8cmp	btfloat8sortsupport	btgetbitmap	btgettuple	btinsert	btint24cmp
btint28cmp	btint2cmp	btint2sortsupport	btint42cmp	btint48cmp	btint4cmp	btint4sortsupport
btint82cmp	btint84cmp	btint8cmp	btint8sortsupport	btmarkpos	btnamecmp	btnamesortsupport
btoidcmp	btoidsortsupport	btoidvectorcmp	btoptions	btrecordcmp	btreltimecmp	btrescan
btrestrpos	btrim	bttext_pattern_cmp	bttextcmp	bttextsortsupport	bttidcmp	bttintervalcmp
btvacuumcleanup	bytea_sortsupport	bytea_string_agg_finalfn	bytea_string_agg_transfn	byteacat	byteacmp	byteaeq
byteage	byteagt	byteain	byteale	bytealike	bytealt	byteane
byteanlike	byteaout	byteasecv	byteasend	cash_cmp	cash_div_cash	cash_div_flt4
cash_div_flt8	cash_div_int2	cash_div_int4	cash_div_int8	cash_eq	cash_ge	cash_gt

版权所有 © 华为云计算技术有限公司

cash_in	cash_le	cash_lt	cash_mi	cash_mul_flt4	cash_mul_flt8	cash_mul_int2
cash_mul_int4	cash_mul_int8	cash_ne	cash_out	cash_pl	cash_recv	cash_send
cashlarger	cashsmaller	cbrt	ceil	ceiling	center	char
char_length	character_length	chareq	charge	chargt	charin	charle
charlt	charne	charout	charrecv	charsend	chr	cideq
cidin	cidout	cidr	cidr_in	cidr_out	cidr_recv	cidr_send
cidrecv	cidsend	circle	circle_above	circle_add_pt	circle_below	circle_center
circle_contain	circle_contain_pt	circle_contained	circle_distance	circle_div_pt	circle_eq	circle_ge
circle_gt	circle_in	circle_le	circle_left	circle_lt	circle_mul_pt	circle_ne
circle_out	circle_overabove	circle_overbelow	circle_overlap	circle_overleft	circle_overright	circle_recv
circle_right	circle_same	circle_send	circle_sub_pt	clock_timestamp	close_lb	close_ls
close_lseg	close_pb	close_pl	close_ps	close_sb	close_sl	col_description
concat	concat_ws	contjoinsel	contsel	convert	convert_from	convert_to
corr	cos	cot	count	covar_pop	covar_samp	cstring_in
cstring_out	cstring_recv	cstring_send	cume_dist	current_database	current_query	current_schema
xpath_exists	current_setting	current_user	currtid	currtid2	currval	cursor_to_xml

版权所有 © 华为云计算技术有限公司

cursor_to_xmlschema	database_to_xml	database_to_xml_and_xmlschema	database_to_xmlschema	date	date_cmp	date_cmp_timestamp
date_cmp_timestamptz	date_eq	date_eq_timestamp	date_eq_timestamptz	date_ge	date_ge_timestamp	date_ge_timestamptz
date_gt	date_gt_timestamp	date_gt_timestamptz	date_in	date_larger	date_le	date_le_timestamp
date_le_timestamptz	date_lt	date_lt_timestamp	date_lt_timestamptz	date_mi	date_mi_interval	date_mii
date_ne	date_ne_timestamp	date_ne_timestamptz	date_out	date_pl_interval	date_pli	date_recv
date_send	date_smaller	date_sortsupport	daterange_canonical	daterange_subdiff	datetime_pl	datetimetz_pl
dcbrt	decode	defined	degrees	delete	dense_rank	dexp
diagonal	diameter	dispell_init	dispell_lexize	dist_cpoly	dist_lb	dist_pb
dist_pc	dist_pl	dist_ppath	dist_ps	dist_sb	dist_sl	div
dlog1	dlog10	domain_in	domain_recv	dpow	dround	dsimple_init
dsimple_lexize	dsnowball_init	dsnowball_lexize	dsqrt	dsynonym_init	dsynonym_lexize	dtrunc
each	enum_ne	enum_out	enum_range	enum_recv	enum_send	enum_smaller
eqjoinsel	eqsel	euc_cn_to_mic	euc_cn_to_utf8	euc_jis_2004_to_shift_jis_2004	euc_jis_2004_to_utf8	euc_jp_to_mic
euc_jp_to_sjis	euc_jp_to_utf8	euc_kr_to_mic	euc_kr_to_utf8	euc_tw_to_big5	euc_tw_to_mic	euc_tw_to_utf8

版权所有 © 华为云计算技术有限公司

every	exist	exists_all	exists_any	exp	factorial	family
fdw_handler_in	fdw_handler_out	fetchval	first_value	float4	float4_accum	float48div
float48eq	float48ge	float48gt	float48le	float48lt	float48mi	float48mul
float48ne	float48pl	float4abs	float4div	float4eq	float4ge	float4gt
float4in	float4larger	float4le	float4lt	float4mi	float4mul	float4ne
float4out	float4pl	float4recv	float4send	float4smaller	float4um	float4up
float8	float8_accum	float8_avg	float8_collect	float8_corr	float8_covar_pop	float8_covar_samp
float8_regr_accum	float8_regr_avgx	float8_regr_avgy	float8_regr_collect	float8_regr_intercept	float8_regr_r2	float8_regr_slope
float8_regr_sxx	float8_regr_sxy	float8_regr_syy	float8_stddev_pop	float8_stddev_samp	float8_var_pop	float8_var_samp
float84div	float84eq	float84ge	float84gt	float84le	float84lt	float84mi
float84mul	float84ne	float84pl	float8abs	float8div	float8eq	float8ge
float8gt	float8in	float8larger	float8le	float8lt	float8mi	float8mul
float8ne	float8out	float8pl	float8recv	float8send	float8smaller	float8um
float8up	floor	flt4_mul_cash	flt8_mul_cash	fmgr_c_validator	fmgr_internal_validator	fmgr_sql_validator
format	format_type	gb18030_to_utf8	gbk_to_utf8	generate_series	generate_subscripts	get_bit

版权所有 © 华为云计算技术有限公司

get_byte	get_current_ts_config	get_global_gs_asp	get_large_table_name	-	-	-
gtsquery_compress	gtsquery_consistent	gtsquery_decompress	gtsquery_penalty	gtsquery_picksplit	gtsquery_same	gtsquery_union
gtsvector_compress	gtsvector_consistent	gtsvector_decompress	gtsvector_penalty	gtsvector_picksplit	gtsvector_same	gtsvector_union
gtsvectorin	gtsvectorout	has_tablespace_privilege	has_type_privilege	hash_aclitem	hashbeginscan	hashbuild
hashbuildempty	hashbulkdelete	hashcostestimate	hashendscan	hashgetbitmap	hashgettuple	hashinsert
hashint2vector	hashint4	hashint8	hashmacaddr	hashmarkpos	hashname	hashoid
hashoidvector	hashoptions	hashrescan	hashrestrpos	hashtext	hashvacuumcleanup	hashvarlena
host	hostmask	iclikejoinsel	iclikesel	icnlikejoinsel	icnlikesel	icregexeqjoinsel
icregexeqsel	icregexnejoinsel	icregexnesel	inet_client_addr	inet_client_port	inet_in	inet_out
inet_recv	inet_send	inet_server_addr	inet_server_port	inetand	inetmi	inetmi_int8
inetnot	inetor	inetpl	initcap	int2_accum	int2_avg_accum	int2_mul_cash
int2_sum	int24div	int24eq	int24ge	int24gt	int24le	int24lt
int24mi	int24mul	int24ne	int24pl	int28div	int28eq	int28ge
int28gt	int28le	int28lt	int28mi	int28mul	int28ne	int28pl
int2abs	int2and	int2div	int2eq	int2ge	int2gt	int2in
int2larger	int2le	int2lt	int2mi	int2mod	int2mul	int2ne

版权所有 © 华为云计算技术有限公司

int2not	int2or	int2out	int2pl	int2recv	int2send	int2shl
int2shr	int2smaller	int2um	int2up	int2vectoreq	int2vectorin	int2vectorout
int2vectorrecv	int2vectorsend	int2xor	int4_accum	int4_avg_accum	int4_mul_cash	int4_sum
int42div	int42eq	int42ge	int42gt	int42le	int42lt	int42mi
int42mul	int42ne	int42pl	int48div	int48eq	int48ge	int48gt
int48le	int48lt	int48mi	int48mul	int48ne	int48pl	int4abs
int4and	int4div	int4eq	int4ge	int4gt	int4in	int4inc
int4larger	int4le	int4lt	int4mi	int4mod	int4mul	int4ne
int4not	int4or	int4out	int4pl	int4range	int4range_canonical	int4range_subdiff
int4recv	int4send	int4shl	int4shr	int4smaller	int4um	int4up
int4xor	int8	int8_avg	int8_avg_accum	int8_avg_collect	int8_mul_cash	int8_sum
int8_sum_to_int8	int8+1635:1668_accum	int82div	int82eq	int82ge	int82gt	int82le
int82lt	int82mi	int82mul	int82ne	int82pl	int84div	int84eq
int84ge	int84gt	int84le	int84lt	int84mi	int84mul	int84ne
int84pl	int8abs	int8and	int8div	int8eq	int8ge	int8gt
int8in	int8inc	int8inc_any	int8inc_float8_float8	int8larger	int8le	int8lt
int8mi	int8mod	int8mul	int8ne	int8not	int8or	int8out

版权所有 © 华为云计算技术有限公司

int8pl	int8pl_inet	int8range	int8range_canonical	int8range_subdiff	int8recv	int8send
int8shl	int8shr	int8smaller	int8um	int8up	int8xor	integer_pl_date
inter_lb	inter_sb	inter_sl	internal_in	internal_out	interval	interval_accum
interval_avg	interval_cmp	interval_collect	interval_div	interval_eq	interval_ge	interval_gt
interval_hash	interval_in	interval_larger	interval_le	interval_lt	interval_mi	interval_mul
interval_ne	interval_out	interval_pl	interval_pl_date	interval_pl_time	interval_pl_timestamp	interval_pl_timestamptz
interval_pl_timetz	interval_recv	interval_send	interval_smaller	interval_transform	interval_um	intervaltypmodin
intervaltypmodout	intinterval	isexists	ishorizontal	iso_to_koi8r	iso_to_mic	iso_to_win1251
iso_to_win866	iso8859_1_to_utf8	iso8859_to_utf8	isparallel	isperp	isvertical	johab_to_utf8
jsonb_in	jsonb_out	jsonb_recv	jsonb_send	-	-	-
json_in	json_out	json_recv	json_send	justify_days	justify_hours	justify_interval
koi8r_to_iso	koi8r_to_mic	koi8r_to_utf8	koi8r_to_win1251	koi8r_to_win866	koi8u_to_utf8	language_handler_in
language_handler_out	latin1_to_mic	latin2_to_mic	latin2_to_win1250	latin3_to_mic	latin4_to_mic	like_escape
likejoinsel	likesel	line	line_distance	line_eq	line_horizontal	line_in
line_interpt	line_intersect	line_out	line_parallel	line_perp	line_recv	line_send
line_vertical	ln	lo_close	lo_creat	lo_create	lo_export	lo_import

版权所有 © 华为云计算技术有限公司

lo_lseek	lo_open	lo_tell	lo_truncate	lo_unlink	log	loread
lower	lower_inc	lower_inf	lowrite	lpad	lseg	lseg_center
lseg_distance	lseg_eq	lseg_ge	lseg_gt	lseg_horizontal	lseg_in	lseg_interpt
lseg_intersect	lseg_le	lseg_length	lseg_lt	lseg_ne	lseg_out	lseg_parallel
lseg_perp	lseg_recv	lseg_send	lseg_vertical	ltrim	macaddr_and	macaddr_cmp
macaddr_eq	macaddr_ge	macaddr_gt	macaddr_in	macaddr_le	macaddr_lt	macaddr_ne
macaddr_not	macaddr_or	macaddr_out	macaddr_recv	macaddr_send	makeaclitem	masklen
max	md5 MD5加密算法安全性低，存在安全风险，建议使用更安全的加密算法。	mic_to_big5	mic_to_euc_cn	mic_to_euc_jp	mic_to_euc_kr	mic_to_euc_tw
mic_to_iso	mic_to_koi8r	mic_to_latin1	mic_to_latin2	mic_to_latin3	mic_to_latin4	mic_to_sjis
mic_to_win1250	mic_to_win1251	mic_to_win866	min	mktinterval	money	mul_d_interval
name	nameeq	namege	namegt	nameiclike	nameicnlike	nameicregexeq
nameicregexne	namein	namele	namelike	namelt	namene	namenlike
nameout	namerecv	nameregexeq	nameregexne	namesend	neqjoinsel	neqsel
network_cmp	network_eq	network_ge	network_gt	network_le	network_lt	network_ne
network_sub	network_subeq	network_sup	network_supeq	nlikejoinsel	nlikesel	numeric

版权所有 © 华为云计算技术有限公司

numeric_abs	numeric_accum	numeric_add	numeric_avg	numeric_avg_accum	numeric_avg_collect	numeric_cmp
numeric_collect	numeric_div	numeric_div_trunc	numeric_eq	numeric_exp	numeric_fac	numeric_ge
numeric_gt	numeric_in	numeric_inc	numeric_larger	numeric_le	numeric_ln	numeric_log
numeric_lt	numeric_mod	numeric_mul	numeric_ne	numeric_out	numeric_power	numeric_recv
numeric_send	numeric_smaller	numeric_sortsupport	numeric_sqrt	numeric_stddev_pop	numeric_stddev_samp	numeric_sub
numeric_transform	numeric_uminus	numeric_uplus	numeric_var_pop	numeric_var_samp	numerictypmodin	numerictypmodout
numrange_subdiff	oid	oideq	oidge	oidgt	oidin	oidlarger
oidle	oidlt	oidne	oidout	oidrecv	oidsend	oidsmaller
oidvectoreq	oidvectorge	oidvectorgt	oidvectorin	oidvectorle	oidvectorlt	oidvectorne
oidvectorout	oidvectorrecv	oidvectorsend	oidvectortypes	on_pb	on_pl	on_ppath
on_ps	on_sb	on_sl	opaque_in	opaque_out	ordered_set_transition	overlaps
overlay	path	path_add	path_add_pt	path_center	path_contain_pt	path_distance
path_div_pt	path_in	path_inter	path_length	path_mul_pt	path_n_eq	path_n_ge
path_n_gt	path_n_le	path_n_lt	path_npoints	path_out	path_recv	path_send

版权所有 © 华为云计算技术有限公司

path_sub_pt	percentile_cont	percentile_cont_float8_final	percentile_cont_interval_final	pg_char_to_encoding	pg_cursor	pg_encoding_max_length
pg_encoding_to_char	pg_extension_config_dump	-	-	pg_node_tree_in	pg_node_tree_out	pg_node_tree_recv
pg_node_tree_send	pg_prepared_statement	pg_prepared_xact	pg_notify	pg_stat_get_wal_receiver	pg_show_all_settings	pg_stat_get_bgwriter_stat_reset_time
pg_stat_get_buf_fsync_backend	pg_stat_get_checkpoint_sync_time	pg_stat_get_checkpoint_write_time	pg_stat_get_db_blk_read_time	pg_stat_get_db_blk_write_time	pg_stat_get_db_conflict_all	pg_stat_get_db_conflict_bufferpin
pg_stat_get_db_conflict_snapshot	pg_stat_get_db_conflict_startup_deadlock	pg_switch_xlog	xpath	pg_timezone_abbrevs	pg_timezone_names	pgxc_node_str
plpgsql_call_handler	plpgsql_inline_handler	plpgsql_validator	point_above	point_add	point_below	point_distance
point_div	point_eq	point_horiz	point_in	point_left	point_mul	point_ne
point_out	point_recv	point_right	point_send	point_sub	point_vert	poly_above
poly_below	poly_center	poly_contain	poly_contain_pt	poly_contained	poly_distance	poly_in
poly_left	poly_npoints	poly_out	poly_overabove	poly_overbelow	poly_overlap	poly_overleft
poly_overright	poly_recv	poly_right	poly_same	poly_send	polygon	position
positionjoinsel	positionsel	postgresql_fdw_validator	pow	power	prsd_end	prsd_headline
prsd_lextype	prsd_nexttoken	prsd_start	pt_contained_circle	pt_contained_poly	query_to_xml	query_to_xml_and_xmlschema

版权所有 © 华为云计算技术有限公司

query_to_xmlschema	quote_ident	quote_literal	quote_nullable	radians	radius	random
range_adjacent	range_after	range_before	range_cmp	range_contained_by	range_contains	range_contains_elem
range_ge	range_eq	range_gt	range_in	range_intersect	range_le	range_lt
range_minus	range_ne	range_out	range_overlaps	range_overleft	range_overright	range_recv
range_send	range_typanalyze	range_union	rank	record_eq	record_ge	record_gt
record_in	record_le	record_lt	record_ne	record_out	record_recv	record_send
regclass	regclassin	regclassout	regclassrecv	regclasssend	regconfigin	regconfigout
regconfigrecv	regconfigsend	regdictionaryin	regdictionaryout	regdictionaryrecv	regdictionarysend	regexeqjoinsel
regexeqsel	regexnejoinsel	regexnesel	regexp_matches	regexp_replace	regexp_split_to_array	regexp_split_to_table
regoperatorin	regoperatorout	regoperatorrecv	regoperatorsend	regoperin	regoperout	regoperrecv
regopersend	regprocedurein	regprocedureout	regprocedurerecv	regproceduresend	regprocin	regprocout
regprocrecv	regprocsend	regr_avgx	regr_avgy	regr_count	regr_intercept	regr_r2
regr_slope	regr_sxx	regr_sxy	regr_syy	regtypein	regtypeout	regtyperecv
regtypesend	reltime	reltimeeq	reltimege	reltimegt	reltimein	reltimele
reltimelt	reltimene	reltimeout	reltimerecv	reltimesend	repeat	replace

版权所有 © 华为云计算技术有限公司

reverse	RI_FKey_cascade_del	RI_FKey_cascade_upd	RI_FKey_check_ins	RI_FKey_check_upd	RI_FKey_noaction_del	RI_FKey_noaction_upd
RI_FKey_restrict_del	RI_FKey_restrict_upd	RI_FKey_setdefault_del	RI_FKey_setdefault_upd	RI_FKey_setnull_del	RI_FKey_setnull_upd	right
round	row_number	row_to_json	rpad	rtrim	scalargtjoinsel	scalargtsel
scalarltjoinsel	scalarltsel	schema_to_xml	schema_to_xml_and_xmlschema	schema_to_xmlschema	session_user	set_bit
set_byte	set_config	set_masklen	shift_jis_2004_to_euc_jis_2004	shift_jis_2004_to_utf8	sjis_to_euc_jp	sjis_to_mic
sjis_to_utf8	smgrin	smgrout	spg_kd_choose	spg_kd_config	spg_kd_inner_consistent	spg_kd_picksplit
spg_quad_choose	spg_quad_config	spg_quad_inner_consistent	spg_quad_leaf_consistent	spg_quad_picksplit	spg_text_choose	spg_text_config
spg_text_inner_consistent	spg_text_leaf_consistent	spg_text_picksplit	spgbeginscan	spgbuild	spgbuildempty	spgbulkdelete
spgcanreturn	spgcostestimate	spgendscan	spggetbitmap	spggettuple	spginsert	spgmarkpos
spgoptions	spgrescan	spgrestrpos	spgvacuumcleanup	stddev	stddev_pop	stddev_samp

版权所有 © 华为云计算技术有限公司

string_agg	string_agg_finalfn	string_agg_transfn	strip	sum	suppress_redundant_updates_trigger	table_to_xml
table_to_xml_and_xmlschema	table_to_xmlschema	tan	text	text_ge	text_gt	text_larger
text_le	text_lt	text_pattern_ge	text_pattern_gt	text_pattern_le	text_pattern_lt	text_smaller
textanycat	textcat	texteq	texticlike	texticnlike	texticregexeq	texticregexne
textin	textlike	textne	textnlike	textout	textrecv	textregexeq
textregexne	textsend	thesaurus_init	thesaurus_lexize	tideq	tidge	tidgt
tidin	tidlarger	tidle	tidlt	tidne	tidout	tidrecv
tidsend	tidsmaller	time	time_cmp	time_eq	time_ge	time_gt
time_hash	time_in	time_larger	time_le	time_lt	time_mi_interval	time_mi_time
time_ne	time_out	time_pl_interval	time_recv	time_send	time_smaller	time_transform
timedate_pl	timemi	timepl	timestamp	timestamp_cmp	timestamp_cmp_date	timestamp_cmp_timestamptz
timestamp_eq	timestamp_eq_date	timestamp_eq_timestamptz	timestamp_ge	timestamp_ge_date	timestamp_ge_timestamptz	timestamp_gt
timestamp_gt_date	timestamp_gt_timestamptz	timestamp_hash	timestamp_in	timestamp_larger	timestamp_le	timestamp_le_date

版权所有 © 华为云计算技术有限公司

timestamp_le_timestamptz	timestamp_lt	timestamp_lt_date	timestamp_lt_timestamptz	timestamp_mi	timestamp_mi_interval	timestamp_ne
timestamp_ne_date	timestamp_ne_timestamptz	timestamp_out	timestamp_pl_interval	timestamp_recv	timestamp_send	timestamp_smaller
timestamp_sortsupport	timestamp_transform	timestamptypmodin	timestamptypmodout	timestamptz	timestamptz_cmp	timestamptz_cmp_date
timestamptz_cmp_timestamp	timestamptz_eq	timestamptz_eq_date	timestamptz_eq_timestamp	timestamptz_ge	timestamptz_ge_date	timestamptz_ge_timestamp
timestamptz_gt	timestamptz_gt_date	timestamptz_gt_timestamp	timestamptz_in	timestamptz_larger	timestamptz_le	timestamptz_le_date
timestamptz_le_timestamp	timestamptz_lt	timestamptz_lt_date	timestamptz_lt_timestamp	timestamptz_mi	timestamptz_mi_interval	timestamptz_ne
timestamptz_ne_date	timestamptz_ne_timestamp	timestamptz_out	timestamptz_pl_interval	timestamptz_recv	timestamptz_send	timestamptz_smaller
timestamptztypmodin	timestamptztypmodout	timetypmodin	timetypmodout	timetz	timetz_cmp	timetz_eq
timetz_ge	timetz_gt	timetz_hash	timetz_in	timetz_larger	timetz_le	timetz_lt
timetz_mi_interval	timetz_ne	timetz_out	timetz_pl_interval	timetz_recv	timetz_send	timetz_smaller
timetzdate_pl	timetztypmodin	timetztypmodout	timezone（2069）	timezone（1159）	timezone（2037）	timezone (2070)

版权所有 © 华为云计算技术有限公司

timezone (1026)	timezone (2038)	tintervalct	tintervaleq	tintervalge	tintervalgt	tintervalin
tintervalle	tintervalleneq	tintervallenge	tintervallengt	tintervallenle	tintervallenlt	tintervallenne
tintervallt	tintervalne	tintervalout	tintervalov	tintervalrecv	tintervalsame	tintervalsend
tintervalstart	to_ascii（1845）	to_ascii（1847）	to_ascii（1846）	trigger_in	trigger_out	ts_match_qv
ts_match_tq	ts_match_tt	ts_match_vq	ts_rank	ts_rank_cd	ts_rewrite	ts_stat
ts_token_type	ts_typanalyze	tsmatchjoinsel	tsmatchsel	tsq_mcontained	tsq_mcontains	tsquery_and
tsquery_cmp	tsquery_eq	tsquery_ge	tsquery_gt	tsquery_le	tsquery_lt	tsquery_ne
tsquery_not	tsquery_or	tsqueryin	tsqueryout	tsqueryrecv	tsquerysend	tsrange
tsrange_subdiff	tstzrange	tstzrange_subdiff	tsvector_cmp	tsvector_concat	tsvector_eq	tsvector_ge
tsvector_gt	tsvector_le	tsvector_lt	tsvector_ne	tsvector_update_trigger	tsvector_update_trigger_column	tsvectorin
tsvectorout	tsvectorrecv	tsvectorsend	txid_current	txid_current_snapshot	txid_snapshot_in	txid_snapshot_out
txid_snapshot_recv	txid_snapshot_send	txid_snapshot_xip	txid_snapshot_xmax	txid_snapshot_xmin	txid_visible_in_snapshot	uhc_to_utf8
unique_key_recheck	unknownin	unknownout	unknownrecv	unknownsend	unnest	utf8_to_big5
utf8_to_euc_cn	utf8_to_euc_jis_2004	utf8_to_euc_jp	utf8_to_euc_kr	utf8_to_euc_tw	utf8_to_gb18030	utf8_to_gbk

版权所有 © 华为云计算技术有限公司

utf8_to_iso8859	utf8_to_iso8859_1	utf8_to_johab	utf8_to_koi8r	utf8_to_koi8u	utf8_to_shift_jis_2004	utf8_to_sjis
utf8_to_uhc	utf8_to_win	uuid_cmp	uuid_eq	uuid_ge	uuid_gt	uuid_hash
uuid_in	uuid_le	uuid_lt	uuid_ne	uuid_out	uuid_recv	uuid_send
var_pop	var_samp	varbit	varbit_in	varbit_out	varbit_recv	varbit_send
varbit_transform	varbitcmp	varbiteq	varbitge	varbitgt	varbitle	varbitlt
varbitne	varbittypmodin	varbittypmodout	varchar	varchar_transform	varcharin	varcharout
varcharrecv	varcharsend	varchartypmodin	varchartypmodout	variance	void_in	void_out
void_recv	void_send	win_to_utf8	win1250_to_latin2	win1250_to_mic	win1251_to_iso	win1251_to_koi8r
win1251_to_mic	win1251_to_win866	win866_to_iso	win866_to_koi8r	win866_to_mic	win866_to_win1251	xideq
xideqint4	xidin	xidout	xidrecv	xidsend	xml	xml_in
xml_is_well_formed	xml_is_well_formed_content	xml_is_well_formed_document	xml_out	xml_recv	xml_send	xmlagg
xmlcomment	xmlconcat2	xmlexists	xmlvalidate	-	year_in	year_out
year_recv	year_send	yeartypmodin	yeartypmodout	year_eq	year_ne	year_lt
year_le	year_gt	year_ge	year_cmp	year_hash	year_larger	year_smaller
year_mi	year_mi_int4	int4_mi_year	year_pl	year_pl_int4	int4_pl_year	int4_year
year_int4	date_year	numeric_year	text_year	time_year	timestamp_year	timestamptz_year

版权所有 © 华为云计算技术有限公司

bpcharlikebpchar	bpcharnlikebpchar	-	-	-	-	-

说明

升级模式下，不支持调用变长参数的系统函数，如concat。

实现内部功能的函数

以下函数用于GaussDB实现系统内部功能，不推荐使用，若需使用，请联系华为技术支持工程师。

- spread_collect()
  描述：该函数用于计算某段时间内最大和最小值的差值，用于聚合函数的数据收集过程。
  参数：s real[]、v real[]
  返回值类型：real[]
- spread_final()
  描述：该函数用于计算某段时间内最大和最小值的差值，用于聚合函数的数据最终处理过程。
  参数：s real[]
  返回值类型：real
- spread_internal()
  描述：该函数用于计算某段时间内最大和最小值的差值，用于聚合函数的数据中间过程。
  参数：s real[]、v real
  返回值类型：real[]
- xidin4()
  描述：输入4字节的xid。
  参数：cstring
  返回值类型：xid32
- gap_fill_internal()
  描述：返回参数列表中第一个非NULL的参数值。
  参数：s anyelement、v anyelement
  返回值类型：anyelement
- int1send()
  描述：将无符号一字节整数打包放入内部数据缓冲流。
  参数：tinyint
  返回值类型：bytea
- is_contain_namespace()
  描述：查找表名和namespace分割的位置，如果不存在namespace，返回0。
  参数：relationname name

版权所有 © 华为云计算技术有限公司

返回值类型：integer

- is_oid_in_group_members()
  描述：不支持
  参数：node_oid oid、group_members oidvector_extend
  返回值类型：boolean
- isubmit_on_nodes_internal()
  描述：不支持
  参数：job bigint、node_name name、database name、what text、next_date timestamp without time zone、job_interval text
  返回值类型：integer
- listagg()
  描述：list类型agg聚集函数。
  参数：smallint、text
  返回值类型：text
- log_fdw_validator()
  描述：验证函数。
  参数：text[]、oid
  返回值类型：void
- nvarchar2typmodin()
  描述：获取varchar的typmod信息。
  参数：cstring[]
  返回值类型：integer
- nvarchar2typmodout()
  描述：获取varchar的typmod信息，并构造字符串返回。
  参数：integer
  返回值类型：cstring
- pg_nodes_memmon()
  描述：不支持。
  参数：nan
  返回值类型：innernname text、innerusedmem bigint、innertopctxt bigint、nname text、usedmem text、sharedbuffercache text、topcontext text
- read_disable_conn_file()
  描述：读取禁止的连接文件。
  参数：nan
  返回值类型：disconn_mode text、disconn_host text、disconn_port text、local_host text、local_port text、redo_finished text
- regex_like_m()
  描述：正则匹配，判断字符串是否符合给定的正则表达式。
  参数：text、text
  返回值类型：boolean
- update_pgjob()

版权所有 © 华为云计算技术有限公司

描述：更新job。

参数：bigint、"char"、bigint、timestamp without time zone、timestamp without time zone、timestamp without time zone、timestamp without time zone、timestamp without time zone、smallint、text

返回值类型：void

- enum_cmp()

  描述：枚举类比较函数，用于判断两个枚举类是否相等，以及相对大小。

  参数：anyenum、anyenum

  返回值类型：integer

- enum_eq()

  描述：枚举类比较函数，用于实现=符号。

  参数：anyenum、anyenum

  返回值类型：boolean

- enum_first()

  描述：返回枚举类中的第一个元素。

  参数：anyenum

  返回值类型：anyenum

- enum_ge()

  描述：枚举类比较函数，用于实现>=符号。

  参数：anyenum、anyenum

  返回值类型：boolean

- enum_gt()

  描述：枚举类比较函数，用于实现>符号。

  参数：anyenum、anyenum

  返回值类型：boolean

- enum_in()

  描述：枚举类比较函数，用于判断元素是否在枚举类中。

  参数：cstring、oid

  返回值类型：anyenum

- enum_larger()

  描述：枚举类比较函数，用于实现>符号。

  参数：anyenum、anyenum

  返回值类型：anyenum

- enum_last()

  描述：返回枚举类中的最后一个元素。

  参数：anyenum

  返回值类型：anyenum

- enum_le()

  描述：枚举类比较函数，用于实现<=符号。

  参数：anyenum、anyenum

  返回值类型：boolean

版权所有 © 华为云计算技术有限公司

- enum_lt()
  描述：枚举类比较函数，用于实现<符号。
  参数：anyenum、anyenum
  返回值类型：boolean
- enum_smaller()
  描述：枚举类比较函数，用于实现<符号。
  参数：anyenum、anyenum
  返回值类型：boolean
- node_oid_name()
  描述：不支持
  参数：oid
  返回值类型：cstring
- pg_buffercache_pages()
  描述：从共享buffer缓存里读取数据。
  参数：nan
  返回值类型：bufferid integer、relfilenode oid、bucketid smallint、storage_type oid、reltablespace oid、reldatabase oid、relforknumber smallint、 relblocknumber bigint、isdirty boolean、usage_count smallint
- pg_check_xidlimit()
  描述：判断nextxid是否>= xidwarnlimit。
  参数：nan
  返回值类型：boolean
- pg_comm_delay()
  描述：展示单个DN的通信库时延状态。
  参数：nan
  返回值类型：text、text、integer、integer、integer、integer
- pg_comm_recv_stream()
  描述：展示单个DN上所有的通信库接收流状态。
  参数：nan
  返回值类型：text、bigint、text、bigint、integer、integer、integer、text、bigint、integer、integer、integer、bigint、bigint、bigint、bigint、bigint
- pg_comm_send_stream()
  描述：展示单个DN上所有的通信库发送流状态。
  参数：nan
  返回值类型：text、bigint、text、bigint、integer、integer、integer、text、bigint、integer、integer、integer、bigint、bigint、bigint、bigint、bigint
- pg_comm_status()
  描述： 展示单个DN的通信状态。
  参数：nan
  返回值类型：text、integer、integer、bigint、bigint、bigint、bigint、bigint、integer、integer、integer、integer、integer

版权所有 © 华为云计算技术有限公司

- pg_log_comm_status()
  描述：在dn上打印一些log。
  参数：nan
  返回值类型：boolean
- pg_parse_clog()
  描述：解析clog获取普通表的xid的status。
  参数：nan
  返回值类型：xid xid、status text
- pg_parse_clog(bucketid)
  描述：解析clog获取指定bucketid的xid的status。
  参数：bucketid，hashbucket表的物理bucketid，普通表是-1。
  返回值类型：xid xid、status text
- pg_pool_ping()
  描述：设置PoolerPing。
  参数：boolean
  返回值类型：SETOF boolean
- pg_pool_validate()
  描述：通过比较pgxc_node系统表中的字段，验证连接是否可用。
  参数：clear boolean、co_node_name cstring
  返回值类型：pid bigint、node_name text
- pg_resume_bkp_flag()
  描述：用于备份恢复获取delay Xlog标志。
  参数：slot_name name
  返回值类型：start_backup_flag boolean、to_delay boolean、ddl_delay_recycle_ptr text、rewind_time text

  说明

  调用该函数的用户需要具有SYSADMIN权限或OPRADMIN权限，运维管理员角色须打开operation_mode。
- gs_static_threadpool_ctrl_status()
  描述：查询线程池中静态池线程相关统计信息。CN线程池开启下分布式CN支持查询相关静态池信息，DN返回空行；线程池关闭下返回空行；参数static_thread_pool_num设置为0或小于线程池group数时静态线程池默认值为0。
  参数：nan
  返回值：text node_name、int group_id、text worker_info，返回字段如**表1-318**所示。

表 1-318 GS_STATIC_THREADPOOL_CTRL_STATUS 字段

名称	类型	描述
node_name	OUT text	实例名。
group_id	OUT int	线程池group组id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
worker_info	OUT text	描述当前group组线程池运行中的动态的统计信息。包括如下信息： default、default_s：线程数量。动态线程池默认值，静态线程池默认值。 expect、expect_s：线程数量。动态线程池预期值，静态线程池预期值。 actual：实际运行的线程数量，包括动态池线程和静态池线程。 static threads limit：当前group组配置的静态池线程数。 has static threads：当前group组是否创建静态池。默认为0，表示没有创建。 idle static threads：空闲的静态池线程个数。 wait session num：等待的会话数。

- gs_thrdstream_worker_info()

  描述：查询stream线程池相关统计信息。仅支持在以线程池模式运行的DN上查询。

  参数：nan

  返回值：text node_name、int group_id、text db_info、text stream_info。返回字段如**表1-319**所示。

  表 1-319 GS_THRDSTREAM_WORKER_INFO 字段

名称	类型	描述
node_name	OUT text	实例名。
group_id	OUT integer	线程池group组id。
db_info	OUT text	描述当前group组stream线程池运行中的动态的DATABASE统计信息。包括如下信息： • total stream reusing db：stream线程池当前使用或复用中的DATABASE数量。 • free session db：stream线程池当前暂未运行session的DATABASE数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stream_info	OUT text	描述当前group组stream线程池运行中的动态的stream线程统计信息。包括如下信息： • actual： 实际启动的stream线程数。 • idle：空闲的stream线程数。 • using：使用中的stream线程数。 • init session threads：初始化session的stream线程数。 • reuse session threads：复用session的stream线程数。

- pg_stat_get_pooler_status()
  描述：查询pooler中的缓存连接状态。
  参数：nan
  返回值类型：如**表1-320**所示。

  表 1-320 pg_stat_get_pooler_status 返回值字段

名称	类型	描述
database_name	OUT text	数据库名称。
user_name	OUT text	用户名。
tid	OUT bigint	非线程池逻辑下为连接CN的线程id，线程池逻辑下为连接CN的sessionid。
pgoptions	OUT text	数据库连接选项，详见《开发指南》中“应用程序开发教程 > 附录 > libpq > 连接参数说明”描述的options字段。
node_oid	OUT bigint	连接的实例节点OID。
in_use	OUT boolean	连接是否正被使用。 • t（true）：表示连接正在使用。 • f（false）：表示连接没有使用。
session_params	OUT text	由此连接下发的GUC session参数。
fdsock	OUT bigint	本端socket。
local_host	OUT text	本端IP。
local_port	OUT bigint	本端端口号。
remote_host	OUT text	对端IP。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
remote_port	OUT bigint	对端端口号。
remote_pid	OUT bigint	对端处于非线程池逻辑下为对端的线程id，对端处于线程池逻辑下为对端的sessionid。
used_count	OUT bigint	该连接的复用次数。
idx	OUT bigint	通信对端DN在本CN内的标识编号。
streamid	OUT bigint	通信流在物理连接中的标识编号。

- gs_stat_get_pooler_agent()

  描述：查询当前连接的CN上Pooler agent的信息，用于辅助连接池问题定位。访问该函数需要系统管理员权限。如果连接DN进行查询，将会返回空表。

  参数：nan

  返回值类型：如**表1-321**所示。

  表 1-321 gs_stat_get_pooler_agent()返回字段

名称	类型	描述
index	OUT bigint	当前agent在全局agent中的下标。
database_name	OUT text	当前会话的数据库名称。
user_name	OUT text	当前会话的用户名。
session_id	OUT bigint	• 非线程池逻辑下为连接CN的线程id。 • 线程池逻辑下为连接CN的sessionid。
dn_oids	OUT oid[]	记录的DN OID列表。
cn_oids	OUT oid[]	记录的CN OID列表。
agent_init	OUT timestamp with time zone	agent初始化的时间，agent在会话创建或pgxc表信息reload时进行初始化。

- get_node_modulo()

  描述：计算节点在group中的下标。

  参数：如**表1-322**所示。

版权所有 © 华为云计算技术有限公司

表 1-322 get_node_modulo 入参列表

参数	类型	描述	取值范围
node_oid	IN oid	节点OID。	[0, 2^32-1]
group_oid	IN oid	nodegroup的OID。	[0, 2^32-1]

返回值类型：uint4

- check_murmurhash_route_node()
  描述：计算key分布的节点是否正确。
  参数：如表1-323所示。

表 1-323 check_murmurhash_route_node 入参列表

参数	类型	描述	取值范围
key	IN text	表示hash键的字符串。	-
node_modulo	IN uint4	节点在nodegroup中的下标。	[0, 2^32-1]
node_num	IN uint4	nodegroup的节点数量。	[0, 2^32-1]

返回值类型：uint4

- gs_libcomm_fd_info()
  描述：查询libcomm通信库socket长连接信息。
  参数：nan
  返回值：如表1-324所示。

表 1-324 gs_libcomm_fd_info 返回值字段

名称	类型	描述
node_name	OUT text	DN实例名。
ip	OUT text	DN的IP。
ctrl_port	OUT bigint	控制通道端口。
data_port	OUT bigint	数据通道端口。
remote_name	OUT text	对端DN实例名。
remote_ip	OUT text	对端IP。
remote_port	OUT bigint	对端端口号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
local_ip	OUT text	本地IP。
local_port	OUT bigint	本地端口号。
fd	OUT bigint	socket连接。
type	OUT text	TCP连接通道类型，包括： • DATA_SEND_FD：数据通道发送端的socket。 • DATA_RECV_FD：数据通道接收端的socket。 • CTL_SEND_FD：控制通道发送端的socket。 • CTL_RECV_FD：控制通道接收端的socket。

说明

该函数只能在DN上使用。若函数返回空，表示当前DN实例未建立libcomm长连接。

- gs_libcomm_memory_info()
  描述：查询当前libcomm通信库内部内存使用的基础配置信息。
  参数：nan
  返回值：如表1-325所示。

表 1-325 gs_libcomm_memory_info 返回值字段

名称	类型	描述
node_name	OUT text	DN实例名。
current_used_memory	OUT bigint	libcomm通信库使用内存，单位为字节（byte）。
current_data_item_num	OUT bigint	libcomm无锁队列数据块个数。
init_mailbox_memory	OUT bigint	libcomm初始化mailbox使用的内存，单位为字节（byte）。
max_datanode	OUT bigint	libcomm通信库支持的最大DN个数。
max_stream	OUT bigint	libcomm单个TCP连接支持的最大数据流个数。
max_quota_size	OUT bigint	libcomm逻辑连接的buffer总大小，单位为字节（byte）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_usable_memory	OUT bigint	libcomm通信库缓存最大可使用内存，单位为字节（byte）。
max_memory_pool	OUT bigint	libcomm通信库在每个DN上可以使用的内存资源池总大小，单位为字节（byte）。

- gs_validate_ext_listen_ip()

  描述：连接DN实例执行，查询连接原扩展IP的无效的DN实例上的业务线程并支持清理。

  参数：如表1-326所示。

  返回值：bigint pid、text node_name

  注意：该函数仅用于当扩展IP被reload之后，原扩展IP上仍然存在旧的连接，该函数支持原IP连接所在线程被清理。执行该函数会导致扩展IP通信侦听被清理，运维操作请谨慎使用。当前仅支持连接DN执行，并由管理员下发。

  表 1-326 GS_VALIDATE_EXT_LISTEN_IP 字段

名称	类型	描述
clear	IN cstring	是否清理，on清理，off不清理。
validate_node_name	IN cstring	指定所要清理扩展IP连接所在DN实例名。
validate_ip	IN cstring	指定所要清理具体的扩展IP。
pid	OUT bigint	被清理的扩展IP连接所在业务线程ID。
node_name	OUT text	被清理的扩展IP连接所在实例名。

- gs_comm_listen_address_ext_info()

  描述：显示当前实例连接listen_address_ext配置扩展IP的DFX信息。

  参数：nan

  返回值类型：text node_name、text app、bigint tid、integer lwtid、bigint query_id、integer socket、text remote_ip、text remote_port、text local_ip、text local_port。返回字段如表1-327所示。

  表 1-327 GS_COMM_LISTEN_ADDRESS_EXT_INFO 字段

名称	类型	描述
node_name	OUT text	描述当前实例名。
app	OUT text	描述当前连接DN的客户端。
tid	OUT bigint	描述当前线程的线程号。
lwtid	OUT integer	描述当前线程的轻量级线程号。
query_id	OUT bigint	描述当前线程的查询ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
socket	OUT integer	描述当前物理连接的socket fd。
remote_ip	OUT text	描述当前连接对端IP。
remote_port	OUT text	描述当前连接对端port。
local_ip	OUT text	描述当前连接本端IP。
local_port	OUT text	描述当前连接本端port。

- gs_get_global_listen_address_ext_info

  描述：CN上提供查询全局扩展IP配置信息。

  参数：如表1-328所示。

  返回值类型：text node_name、text host、text port、text ext_listen_ip

  注意： 若函数入参为“all”，则函数依赖ext_ip_info静态配置文件，若文件异常需要依赖CM/OM工具修复，此时无法实时更新配置。用户可以通过入参为“primary”获取所有主DN相关配置信息。

  表 1-328 GS_GET_GLOBAL_LISTEN_ADDRESS_EXT_INFO 字段

名称	类型	描述
dn_mode	IN cstring	指定显示的DN范围，null默认查询所有。
node_name	OUT text	DN实例名。
host	OUT text	DN实例侦听IP。
port	OUT text	DN实例侦听port。
ext_listen_ip	OUT text	DN实例侦听扩展IP。

- gs_get_listen_address_ext_info()

  描述：提供查询当前实例扩展IP配置信息。

  参数：nan

  返回值类型：text node_name、text host、bigint port、text ext_listen_ip。返回字段如表1-329所示。

  表 1-329 GS_GET_LISTEN_ADDRESS_EXT_INFO

名称	类型	描述
node_name	OUT text	DN实例名。
host	OUT text	DN实例侦听IP。
port	OUT bigint	DN实例侦听port。
ext_listen_ip	OUT text	DN实例侦听扩展IP。

版权所有 © 华为云计算技术有限公司

- psortoptions()
  描述：返回psort属性。
  参数：text[]、boolean
  返回值类型：bytea
- remove_job_class_depend()
  描述：移除job依赖。
  参数：oid
  返回值类型：void
- xideq4()
  描述：对比两个xid是否相等。
  参数：xid32、xid32
  返回值类型：boolean
- xideqint8()
  描述：对比两个xid是否相等。
  参数：xid、bigint
  返回值类型：boolean
- xidlt()
  描述：返回xid1 < xid2是否成立。
  参数：xid、xid
  返回值类型：boolean
- xidlt4()
  描述：返回xid1 < xid2是否成立。
  参数：xid32、xid32
  返回值类型：boolean
- get_local_cont_query_stat()
  描述：获取本机节点的指定持续计算视图统计信息。
  参数：cq_id oid
  返回值类型：cq oid、w_in_rows int8、w_in_bytes int8、w_out_rows int8、w_out_bytes int8、w_pendings int8、w_errors int8、r_in_rows int8、r_in_bytes int8、r_out_rows int8、r_out_bytes int8、r_errors int8、c_in_rows int8、c_in_bytes int8、c_out_rows int8、c_out_bytes int8、c_pendings int8、c_errors int8
- get_local_cont_query_stats()
  描述：获取本机节点的所有持续计算视图统计信息。
  参数：nan
  返回值类型：cq oid、w_in_rows int8、w_in_bytes int8、w_out_rows int8、w_out_bytes int8、w_pendings int8、w_errors int8、r_in_rows int8、r_in_bytes int8、r_out_rows int8、r_out_bytes int8、r_errors int8、c_in_rows int8、c_in_bytes int8、c_out_rows int8、c_out_bytes int8、c_pendings int8、c_errors int8
- get_cont_query_stats()
  描述：获取各个DN节点的所有持续计算视图统计信息。

版权所有 © 华为云计算技术有限公司

参数：nan

返回值类型：node name、cq oid、w_in_rows int8、w_in_bytes int8、w_out_rows int8、w_out_bytes int8、w_pendings int8、w_errors int8、r_in_rows int8、r_in_bytes int8、r_out_rows int8、r_out_bytes int8、r_errors int8、c_in_rows int8、c_in_bytes int8、c_out_rows int8、c_out_bytes int8、c_pendings int8、c_errors int8

- reset_local_cont_query_stat()

  描述：复位本机节点的指定持续计算视图统计信息。

  参数：cq_id oid

  返回值类型：boolean

- reset_local_cont_query_stats()

  描述：复位本机节点的指定持续计算视图的关联统计信息。

  参数：cq_id oid

  返回值类型：boolean

- reset_cont_query_stats()

  描述：复位各个DN节点的STREAM对象对应的持续计算视图统计信息。

  参数：stream_id oid

  返回值类型：boolean

- check_cont_query_schema_changed()

  描述：判断指定持续计算视图的schema change状态。

  参数：cq_id oid

  返回值类型：boolean

- gs_get_standby_cluster_barrier_status()

  描述：查看备cn/dn的barrier日志回放情况，包括已接收到的最新barrier点、已接收到的最新barrier点的LSN，上一次回放的barrier点，回放的目标barrier点。

  参数：nan

  返回值类型：barrier_id text、barrier_lsn text、recovery_id text、target_id text

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_set_standby_cluster_target_barrier_id()

  描述：设置回放的目标barrier点。

  参数：barrier_id字符串

  返回值类型：target_id text

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- gs_query_standby_cluster_barrier_id_exist()

  描述：查询指定barrier点备机是否接收到。

  参数：barrier_id字符串

  返回值类型：boolean

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

- standby_read_status()

版权所有 © 华为云计算技术有限公司

描述：用于“GTM_FREE 分布式备机读”功能，获取集群内部所有DN的csn快照信息和主备差异时长，以及最后一次备机读查询所用的csn快照信息。

参数：nan

返回值类型：text、text、text、integer、bigint、bigint、bigint。返回字段如**表1-330**所示。

表 1-330 standby_read_status 字段

名称	类型	描述
name	text	DN节点名称。
type	text	DN节点类型。（主机为primary，备机为standby）
host	text	DN节点所使用的IP。
port	integer	DN节点所使用的PORT。
collected_csn	bigint	该DN的csn快照信息。（主机为0，备机为当前收集到的csn）
delay	bigint	主备DN之间的差异，单位为毫秒。
visited_csn	bigint	最后一次备机读查询使用的csn信息（仅最后一次查询执行的备机有值其余为0）

- gs_shutdown_cross_region_walsenders()

  描述：中断跨集群流式复制。

  参数：nan

  返回值类型：void

  备注：调用该函数的用户需要具有SYSADMIN权限或具有OPRADMIN权限，运维管理员角色须打开operation_mode。

  以下流函数虽存在但功能尚未支持，不建议使用：

  streaming_int8_avg_gather、streaming_numeric_avg_gather、streaming_float8_avg_gather、streaming_interval_avg_gather、streaming_int8_sum_gather、streaming_int2_int4_sum_gather。

- is_dblink_in_transaction()

  描述：判断当前事务中是否使用了oid对应的DATABASE LINK。

  参数：oid

  返回值类型：boolean

- dblink_has_updatasent()

  描述：判断当前事务中是否使用oid对应的DATABASE LINK发送了dml语句且未提交。

  参数：oid

  返回值类型：boolean

- get_last_xmin_by_oid()

  描述：通过表的oid获取该表所有字段中最大的xmin值。

  参数：oid

版权所有 © 华为云计算技术有限公司

返回值类型：xid

- get_relid_by_relname()
  描述：通过表的表名及relnamespace获取表的oid。
  参数：cstring、oid
  返回值类型：oid
- btint12cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int1、int2
  返回值类型：integer
- btint14cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int1、int4
  返回值类型：integer
- btint18cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int1、int8
  返回值类型：integer
- btint116cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int1、int16
  返回值类型：integer
- btint1numericcmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int1、numeric
  返回值类型：integer
- btint21cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int2、int1
  返回值类型：integer
- btint216cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int2、int16
  返回值类型：integer
- btint2numericcmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int2、numeric
  返回值类型：integer
- btint41cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int4、int1

版权所有 © 华为云计算技术有限公司

返回值类型：integer

- btint416cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int4、int16
  返回值类型：integer
- btint4numericcmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int4、numeric
  返回值类型：integer
- btint81cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int8、int1
  返回值类型：integer
- btint816cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int8、int16
  返回值类型：integer
- btint8numericcmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int8、numeric
  返回值类型：integer
- btint161cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int16、int1
  返回值类型：integer
- btint162cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int16、int2
  返回值类型：integer
- btint164cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int16、int4
  返回值类型：integer
- btint168cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int16、int8
  返回值类型：integer
- btnumericint1cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：numeric、int1

版权所有 © 华为云计算技术有限公司

返回值类型：integer

- btnumericint2cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：numeric、int2
  返回值类型：integer
- btnumericint4cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：numeric、int4
  返回值类型：integer
- btnumericint8cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：numeric、int8
  返回值类型：integer
- btint16cmp()
  描述：比较两个参数的大小关系，大于返回正数，等于返回0，小于返回负数。
  参数：int16、int16
  返回值类型：integer
- hashint16()
  描述：计算入参的hash值。
  参数：int16
  返回值类型：integer
- hashint1_numeric()
  描述：计算入参的hash值。
  参数：int1
  返回值类型：integer
- hashint2_numeric()
  描述：计算入参的hash值。
  参数：int2
  返回值类型：integer
- hashint4_numeric()
  描述：计算入参的hash值。
  参数：int4
  返回值类型：integer
- hashint8_numeric()
  描述：计算入参的hash值。
  参数：int8
  返回值类型：integer
- int12eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int1、int2

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int14eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int1、int4
  返回值类型：boolean
- int18eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int1、int8
  返回值类型：boolean
- int116eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int1、int16
  返回值类型：boolean
- int1numericeq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int1、numeric
  返回值类型：boolean
- int21eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int2、int1
  返回值类型：boolean
- int216eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int2、int16
  返回值类型：boolean
- int2numericeq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int2、numeric
  返回值类型：boolean
- int41eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int4、int1
  返回值类型：boolean
- int416eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int4、int16
  返回值类型：boolean
- int4numericeq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int4、numeric

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int81eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int8、int1
  返回值类型：boolean
- int816eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int8、int16
  返回值类型：boolean
- int8numericeq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int8、numeric
  返回值类型：boolean
- int161eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int16、int1
  返回值类型：boolean
- int162eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int16、int2
  返回值类型：boolean
- int164eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int16、int4
  返回值类型：boolean
- int168eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：int16、int8
  返回值类型：boolean
- numericint1eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：numeric、int1
  返回值类型：boolean
- numericint2eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：numeric、int2
  返回值类型：boolean
- numericint4eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：numeric、int4

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- numericint8eq()
  描述：比较两个参数是否相等，如果相等返回true，否则返回false。
  参数：numeric、int8
  返回值类型：boolean
- int12ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int1、int2
  返回值类型：boolean
- int14ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int1、int4
  返回值类型：boolean
- int18ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int1、int8
  返回值类型：boolean
- int116ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int1、int16
  返回值类型：boolean
- int1numericne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int1、numeric
  返回值类型：boolean
- int21ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int2、int1
  返回值类型：boolean
- int216ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int2、int16
  返回值类型：boolean
- int2numericne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int2、numeric
  返回值类型：boolean
- int41ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int4、int1

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int416ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int4、int16
  返回值类型：boolean
- int4numericne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int4、numeric
  返回值类型：boolean
- int81ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int8、int1
  返回值类型：boolean
- int816ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int8、int16
  返回值类型：boolean
- int8numericne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int8、numeric
  返回值类型：boolean
- int161ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int16、int1
  返回值类型：boolean
- int162ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int16、int2
  返回值类型：boolean
- int164ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int16、int4
  返回值类型：boolean
- int168ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：int16、int8
  返回值类型：boolean
- numericint1ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：numeric、int1

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- numericint2ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：numeric、int2
  返回值类型：boolean
- numericint4ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：numeric、int4
  返回值类型：boolean
- numericint8ne()
  描述：比较两个参数是否不相等，如果不相等返回true，否则返回false。
  参数：numeric、int8
  返回值类型：boolean
- int12lt()
  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。
  参数：int1、int2
  返回值类型：boolean
- int14lt()
  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。
  参数：int1、int4
  返回值类型：boolean
- int18lt()
  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。
  参数：int1、int8
  返回值类型：boolean
- int116lt()
  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。
  参数：int1、int16
  返回值类型：boolean
- int1numericlt()
  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。
  参数：int1、numeric
  返回值类型：boolean
- int21lt()
  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。
  参数：int2、int1

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int216lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int2、int16

  返回值类型：boolean

- int2numericlt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int2、numeric

  返回值类型：boolean

- int41lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int4、int1

  返回值类型：boolean

- int416lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int4、int16

  返回值类型：boolean

- int4numericlt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int4、numeric

  返回值类型：boolean

- int81lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int8、int1

  返回值类型：boolean

- int816lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int8、int16

  返回值类型：boolean

- int8numericlt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int8、numeric

  返回值类型：boolean

- int161lt()

版权所有 © 华为云计算技术有限公司

描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

参数：int16、int1

返回值类型：boolean

- int162lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int16、int2

  返回值类型：boolean

- int164lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int16、int4

  返回值类型：boolean

- int168lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：int16、int8

  返回值类型：boolean

- numericint1lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：numeric、int1

  返回值类型：boolean

- numericint2lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：numeric、int2

  返回值类型：boolean

- numericint4lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：numeric、int4

  返回值类型：boolean

- numericint8lt()

  描述：比较两个参数是否是小于关系，如果是小于关系返回true，否则返回false。

  参数：numeric、int8

  返回值类型：boolean

- int12gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int1、int2

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int14gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int1、int4

  返回值类型：boolean

- int18gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int1、int8

  返回值类型：boolean

- int116gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int1、int16

  返回值类型：boolean

- int1numericgt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int1、numeric

  返回值类型：boolean

- int21gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int2、int1

  返回值类型：boolean

- int216gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int2、int16

  返回值类型：boolean

- int2numericgt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int2、numeric

  返回值类型：boolean

- int41gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int4、int1

  返回值类型：boolean

- int416gt()

版权所有 © 华为云计算技术有限公司

描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

参数：int4、int16

返回值类型：boolean

- int4numericgt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int4、numeric

  返回值类型：boolean

- int81gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int8、int1

  返回值类型：boolean

- int816gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int8、int16

  返回值类型：boolean

- int8numericgt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int8、numeric

  返回值类型：boolean

- int161gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int16、int1

  返回值类型：boolean

- int162gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int16、int2

  返回值类型：boolean

- int164gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int16、int4

  返回值类型：boolean

- int168gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：int16、int8

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- numericint1gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：numeric、int1

  返回值类型：boolean

- numericint2gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：numeric、int2

  返回值类型：boolean

- numericint4gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：numeric、int4

  返回值类型：boolean

- numericint8gt()

  描述：比较两个参数是否是大于关系，如果是大于关系返回true，否则返回false。

  参数：numeric、int8

  返回值类型：boolean

- int12le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int1、int2

  返回值类型：boolean

- int14le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int1、int4

  返回值类型：boolean

- int18le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int1、int8

  返回值类型：boolean

- int116le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int1、int16

  返回值类型：boolean

- int1numericle()

版权所有 © 华为云计算技术有限公司

描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

参数：int1、numeric

返回值类型：boolean

- int21le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int2、int1

  返回值类型：boolean

- int216le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int2、int16

  返回值类型：boolean

- int2numericle()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int2、numeric

  返回值类型：boolean

- int41le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int4、int1

  返回值类型：boolean

- int416le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int4、int16

  返回值类型：boolean

- int4numericle()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int4、numeric

  返回值类型：boolean

- int81le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int8、int1

  返回值类型：boolean

- int816le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int8、int16

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int8numericle()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int8、numeric

  返回值类型：boolean

- int161le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int16、int1

  返回值类型：boolean

- int162le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int16、int2

  返回值类型：boolean

- int164le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int16、int4

  返回值类型：boolean

- int168le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：int16、int8

  返回值类型：boolean

- numericint1le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：numeric、int1

  返回值类型：boolean

- numericint2le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：numeric、int2

  返回值类型：boolean

- numericint4le()

  描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

  参数：numeric、int4

  返回值类型：boolean

- numericint8le()

版权所有 © 华为云计算技术有限公司

描述：比较两个参数是否是小于或等于关系，如果是小于或等于关系返回true，否则返回false。

参数：numeric、int8

返回值类型：boolean

- int12ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int1、int2

  返回值类型：boolean

- int14ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int1、int4

  返回值类型：boolean

- int18ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int1、int8

  返回值类型：boolean

- int116ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int1、int16

  返回值类型：boolean

- int1numericge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int1、numeric

  返回值类型：boolean

- int21ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int2、int1

  返回值类型：boolean

- int216ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int2、int16

  返回值类型：boolean

- int2numericge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int2、numeric

版权所有 © 华为云计算技术有限公司

返回值类型：boolean

- int41ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int4、int1

  返回值类型：boolean

- int416ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int4、int16

  返回值类型：boolean

- int4numericge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int4、numeric

  返回值类型：boolean

- int81ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int8、int1

  返回值类型：boolean

- int816ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int8、int16

  返回值类型：boolean

- int8numericge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int8、numeric

  返回值类型：boolean

- int161ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int16、int1

  返回值类型：boolean

- int162ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int16、int2

  返回值类型：boolean

- int164ge()

版权所有 © 华为云计算技术有限公司

描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

参数：int16、int4

返回值类型：boolean

- int168ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：int16、int8

  返回值类型：boolean

- numericint1ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：numeric、int1

  返回值类型：boolean

- numericint2ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：numeric、int2

  返回值类型：boolean

- numericint4ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：numeric、int4

  返回值类型：boolean

- numericint8ge()

  描述：比较两个参数是否是大于或等于关系，如果是大于或等于关系返回true，否则返回false。

  参数：numeric、int8

  返回值类型：boolean

- timestamp_in

  描述：将输入的cstring类型转为timestamp类型。

  参数：cstring

  返回值类型：timestamp

- timestamp_out

  描述：将timestamp类型转为cstring类型。

  参数：timestamp

  返回值类型：cstring

- datea_cmp

  描述：比较两个参数大小，若第一个参数大于第二个参数则返回1，若第一个参数小于第二个参数则返回-1，如果两者相等则返回0。

  参数：datea、datea

  返回值类型：int4

版权所有 © 华为云计算技术有限公司

- datea_eq
  描述：比较两个参数是否相等，如果相等则返回true，否则返回false。
  参数：datea、datea
  返回值类型：boolean
- datea_ge
  描述：如果第一个参数大于等于第二个参数则返回true，否则返回false。
  参数：datea、datea
  返回值类型：boolean
- datea_gt
  描述：如果第一个参数大于第二个参数则返回true，否则返回false。
  参数：datea、datea
  返回值类型：boolean
- datea_le
  描述：如果第一个参数小于等于第二个参数则返回true，否则返回false。
  参数：datea、datea
  返回值类型：boolean
- datea_lt
  描述：如果第一个参数小于第二个参数则返回true，否则返回false。
  参数：datea、datea
  返回值类型：boolean
- datea_neq
  描述：比较两个参数是否不相等，如果不相等则返回true，否则返回false。
  参数：datea、datea
  返回值类型：boolean
- datea_pl_interval
  描述：返回datea类型和interval类型相加得到的结果。
  参数：datea、interval
  返回值类型：datea
- interval_pl_datea
  描述：返回interval类型和datea类型相加得到的结果。
  参数：interval、datea
  返回值类型：datea
- datea_pl_float4
  描述：返回datea类型和float4类型相加得到的结果。
  参数：datea、float4
  返回值类型：datea
- float4_pl_datea
  描述：返回float4类型和datea类型相加得到的结果。
  参数：float4、datea
  返回值类型：datea

版权所有 © 华为云计算技术有限公司

- datea_pl_float8
  描述：返回datea类型和float8类型相加得到的结果。
  参数：datea、float8
  返回值类型：datea
- float8_pl_datea
  描述：返回float8类型和datea类型相加得到的结果。
  参数：float8、datea
  返回值类型：datea
- datea_pl_int8
  描述：返回datea类型和int8类型相加得到的结果。
  参数：datea、int8
  返回值类型：datea
- int8_pl_datea
  描述：返回int8类型和datea类型相加得到的结果。
  参数：int8、datea
  返回值类型：datea
- datea_pl_text
  描述：返回datea类型和text类型相加得到的结果。
  参数：datea、text
  返回值类型：datea
- text_pl_datea
  描述：返回text类型和datea类型相加得到的结果。
  参数：text、datea
  返回值类型：datea
- numeric_pl_datea
  描述：返回numeric类型和datea类型相加得到的结果。
  参数：numeric、datea
  返回值类型：datea
- datea_mi_interval
  描述：返回datea类型和interval类型相减得到的结果。
  参数：datea、interval
  返回值类型：datea
- datea_mi
  描述：返回两个datea类型相减得到的结果。
  参数：datea、datea
  返回值类型：numeric
- datea_mi_timestamp
  描述：返回datea类型和timestamp类型相减得到的结果。
  参数：datea、timestamp
  返回值类型：interval

版权所有 © 华为云计算技术有限公司

- timestamp_mi_datea
  描述：返回timestamp类型和datea类型相减得到的结果。
  参数：timestamp、datea
  返回值类型：interval
- datea_mi_float4
  描述：返回datea类型和float4类型相减得到的结果。
  参数：datea、float4
  返回值类型：datea
- datea_mi_float8
  描述：返回datea类型和float8类型相减得到的结果。
  参数：datea、float8
  返回值类型：datea
- datea_mi_int8
  描述：返回datea类型和int8类型相减得到的结果。
  参数：datea、int8
  返回值类型：datea
- datea_mi_numeric
  描述：返回datea类型和numeric类型相减得到的结果。
  参数：datea、numeric
  返回值类型：datea
- datea_mi_text
  描述：返回datea类型和text类型相减得到的结果。
  参数：datea、text
  返回值类型：datea
- datea_larger
  描述：返回两个datea类型中较大的值。
  参数：datea、datea
  返回值类型：datea
- datea_smaller
  描述：返回两个datea类型中较小的值。
  参数：datea、datea
  返回值类型：datea
- datea_in
  描述：将输入的cstring类型转为datea类型。
  参数：cstring
  返回值类型：datea
- datea_out
  描述：将datea类型转为cstring类型。
  参数：datea
  返回值类型：cstring

版权所有 © 华为云计算技术有限公司

- datea_recv
  描述：将二进制数据转换为datea类型。
  参数：internal
  返回值类型：datea
- datea_send
  描述：将datea类型转换为二进制数据。
  参数：datea
  返回值类型：bytea
- rowid_cmp(rowid, rowid)
  描述：比较两个参数大小，若第一个参数大于第二个参数则返回1，若第一个参数小于第二个参数则返回-1，如果两者相等则返回0。
  参数：rowid、rowid
  返回值类型：int4
- rowid_eq(rowid, rowid)
  描述：比较两个参数是否相等，如果相等则返回true，否则返回false。
  参数：rowid、rowid
  返回值类型：boolean
- rowid_ge(rowid, rowid)
  描述：如果第一个参数大于等于第二个参数则返回true，否则返回false。
  参数：rowid、rowid
  返回值类型：boolean
- rowid_gt(rowid, rowid)
  描述：如果第一个参数大于第二个参数则返回true，否则返回false。
  参数：rowid、rowid
  返回值类型：boolean
- rowid_le(rowid, rowid)
  描述：如果第一个参数小于等于第二个参数则返回true，否则返回false。
  参数：rowid、rowid
  返回值类型：boolean
- rowid_lt(rowid, rowid)
  描述：如果第一个参数小于第二个参数则返回true，否则返回false。
  参数：rowid、rowid
  返回值类型：boolean
- rowid_ne(rowid, rowid)
  描述：比较两个参数是否不相等，如果不相等则返回true，否则返回false。
  参数：rowid、rowid
  返回值类型：boolean
- rowid_larger(rowid, rowid)
  描述：返回两个rowid类型中较大的值。
  参数：rowid、rowid
  返回值类型：rowid

版权所有 © 华为云计算技术有限公司

- rowid_smaller(rowid, rowid)
  描述：返回两个rowid类型中较小的值。
  参数：rowid、rowid
  返回值类型：rowid
- rowid_in(cstring)
  描述：将输入的cstring类型转为rowid类型。
  参数：cstring
  返回值类型：rowid
- rowid_out(rowid)
  描述：将rowid类型转为cstring类型。
  参数：rowid
  返回值类型：cstring
- rowid_recv(internal)
  描述：将二进制数据转换为rowid类型。
  参数：internal
  返回值类型：rowid
- rowid_send(rowid)
  描述：将rowid类型转换为二进制数据。
  参数：rowid
  返回值类型：bytea
- gs_relation_is_updatable()
  描述：返回关系是否可插入、更新、删除。可更新返回4，可插入返回8，可删除返回16，同时满足的则返回对应值相加后得到的数值。
  参数：oid、boolean
  返回值类型：int32
- gs_column_is_updatable()
  描述：返回列是否可更新，可更新返回t，否则返回f。
  参数：oid、int16、boolean
  返回值类型：boolean
- hash_meet_partition(partition_num, hash_key_name1, hash_key_name2)
  描述：用于在线DDL交换分区中计算tuple位于指定的hash分区中。返回tuple所在分区的下标。
  参数：int32、text
  返回值类型：int32

参数说明	类型	描述	取值范围
partition_num	int32	分区表的总分区数。	-
hash_key_name	text	hash分区表的分区键名称。可变参数，个数与分区键个数一致。	-

版权所有 © 华为云计算技术有限公司

示例：

```
--创建表。
gaussdb=# CREATE TABLE t1(c1 INT,c2 INT) PARTITION BY HASH(c2)
(
    PARTITION P1,
    PARTITION P2,
    PARTITION P3
);

gaussdb=# INSERT INTO t1 VALUES(generate_series(1,10), generate_series(1,10));

gaussdb=# SELECT * FROM t1 WHERE hash_meet_partition(3, c2) = 0;
 c1 | c2
----+----
  1 |  1
  4 |  4
  8 |  8
  9 |  9
 10 | 10
(5 rows)

gaussdb=# SELECT * FROM t1 partition (p1);
 c1 | c2
----+----
  1 |  1
  4 |  4
  8 |  8
  9 |  9
 10 | 10
(5 rows)
```

- online_ddl_cleanup(relschema, relname)

  描述：用于在线DDL或在线Vacuum full执行过程中发生故障后，对临时工具进行清理恢复。

  参数：text、text

  返回值类型：text

参数	类型	描述	取值范围
relschema	text	表所在的模式。	-
relname	text	表的名称。	-

  说明

  - 备机上不支持online_ddl_cleanup。
  - 故障发生后，在下次执行在线DDL或在线VACUUM FULl时，会自动清理表上的临时状态。

  示例：

```
--创建表
gaussdb=# CREATE TABLE t1(c1 int, c2 int);
--执行在线DDL
gaussdb=# ALTER TABLE ONLINE t1 ALTER c2 type BIGINT;

--故障发生后进行清理恢复
gaussdb=# SELECT online_ddl_cleanup('public','t1');
 online_ddl_cleanup
--------------------
 cleanup success
(1 row)
```


版权所有 © 华为云计算技术有限公司

- gs_stat_progress_online_ddl(relschema, relname)
  描述：查询系统中正在执行的在线VACUUM FULL的状态和进度。
  参数：两个参数可以同时缺省、同时为NULL或同时不为NULL。
  返回类型：table类型。其中每一行代表一个DDL的一个阶段。

参数	类型	描述	取值范围
relschema	cstring	执行在线VACUUM FULL的Schema。	-
relname	cstring	执行在线VACUUM FULL的表名。	-

说明

1. 当前版本仅支持在线VACUUM FULL的进度监控，不支持其它在线DDL的进度监控。
2. 当relschema和relname参数非空时，可查询到该表残余的在线DDL信息。当relschema和relname参数均为空或者缺省时，可查询到所有正在执行的在线VACUUM FULL信息。
3. 在线DDL的临时表位于“online$$ddl$$[（表所在的模式+表名）的hash值]”schema下。
4. 若GUC参数delete_cctmp_table设置为on，在线DDL结束后将删除临时表的schema；若设置为off，则会将schema重命名为“online$$ddl$$[（表所在的模式+表名）的hash值]_[日期时间]”。
5. 若当前用户为初始用户或系统管理员，可查询所有正在执行的在线VACUUM FULL进度信息；若为普通用户，则仅能查看当前用户自己所执行任务的进度信息。
6. 进度监控信息为准实时：监控函数会实时返回当前的最新阶段，但对于每个阶段内部的进度信息不保证实时更新。

示例：

```
--会话1：执行在线Vacuum full
gaussdb=# VACUUM FULL t1 online;
--会话2： 执行进度监控函数
gaussdb=# \x -- 修改输出排版
gaussdb=# select * from gs_stat_progress_online_ddl('public', 't1') ;
-[ RECORD 1 ]------+--------------------------------------------------------------------------
pid                | 139717555775232
node_name          | coordinator2
phase_id           | 1
rel_schema         | public
rel_name           | t1
sql_command        | VACUUM FULL t1 online with (enable_cancel=true, parallel_threads=2);
started_time       | 2026-07-14 17:09:54.977
elapsed_time       | 10269
phase_name         | Initialization
phase_started_time | 2026-07-14 17:09:54.977
phase_elapsed_time | 716
lock_level         | 8
lock_try_times     | 1
lock_hold_time     | 705
phase_detail       |
-[ RECORD 2 ]------+--------------------------------------------------------------------------
pid                | 139717555775232
node_name          | coordinator2
phase_id           | 2
rel_schema         | public
rel_name           | t1
sql_command        | VACUUM FULL t1 online with (enable_cancel=true, parallel_threads=2);
started_time       | 2026-07-14 17:09:54.977
```


版权所有 © 华为云计算技术有限公司

```
elapsed_time       | 10269
phase_name         | Base data copy
phase_started_time | 2026-07-14 17:09:56.003
phase_elapsed_time | 2422
lock_level         | 4
lock_try_times     | 1
lock_hold_time     | 2479
phase_detail       | {
                   |     "total_pages":  4356,
                   |     "inserted_rows":        1000000,
                   |     "insert_speed": 447828.03403493058,
                   |     "parallel_threads":     2
                   | }
-[ RECORD 3 ]------+--------------------------------------------------------------------------
pid                | 139717555775232
node_name          | coordinator2
phase_id           | 3
rel_schema         | public
rel_name           | t1
sql_command        | VACUUM FULL t1 online with (enable_cancel=true, parallel_threads=2);
started_time       | 2026-07-14 17:09:54.977
elapsed_time       | 10269
phase_name         | Reindexing
phase_started_time | 2026-07-14 17:09:58.428
phase_elapsed_time | 4656
lock_level         | 4
lock_try_times     | 1
lock_hold_time     | 4432
phase_detail       | {
                   |     "total_indexes":        2,
                   |     "indexes_rebuilt":      2,
                   |     "parallel_threads":     2,
                   |     "parallel_workers":     1
                   | }
-[ RECORD 4 ]------+--------------------------------------------------------------------------
pid                | 139717555775232
node_name          | coordinator2
phase_id           | 4
rel_schema         | public
rel_name           | t1
sql_command        | VACUUM FULL t1 online with (enable_cancel=true, parallel_threads=2);
started_time       | 2026-07-14 17:09:54.977
elapsed_time       | 10269
phase_name         | Catchup 1
phase_started_time | 2026-07-14 17:10:04.518
phase_elapsed_time | 275
lock_level         | 4
lock_try_times     | 1
lock_hold_time     | 23
phase_detail       | {
                   |     "short_lock_level":     7,
                   |     "short_lock_try_times": 1,
                   |     "short_lock_hold_time": 252,
                   |     "total_delete_pages":   0,
                   |     "total_insert_pages":   0,
                   |     "deleted_rows": 0,
                   |     "delete_speed": 0,
                   |     "inserted_rows":        0,
                   |     "insert_speed": 0,
                   |     "last_catchup_threshold":       5000,
                   |     "max_catchup_times":    50
                   | }
-[ RECORD 5 ]------+--------------------------------------------------------------------------
pid                | 139717555775232
node_name          | coordinator2
phase_id           | 5
rel_schema         | public
rel_name           | t1
sql_command        | VACUUM FULL t1 online with (enable_cancel=true, parallel_threads=2);
```


版权所有 © 华为云计算技术有限公司

```
started_time       | 2026-07-14 17:09:54.977
elapsed_time       | 10269
phase_name         | Last catchup
phase_started_time | 2026-07-14 17:10:04.926
phase_elapsed_time | 299
lock_level         | 8
lock_try_times     | 1
lock_hold_time     | 74
phase_detail       | {
                   |     "total_delete_pages":   0,
                   |     "total_insert_pages":   0,
                   |     "deleted_rows": 0,
                   |     "delete_speed": 0,
                   |     "inserted_rows":        0,
                   |     "insert_speed": 0,
                   |     "last_catchup_threshold":       5000,
                   |     "max_catchup_times":    50
                   | }
```

各个字段的含义：

参数	描述
pid	进程ID。
node_name	执行的节点名称。
phase_id	阶段编号，顺序递增。
rel_schema	表所在的模式。
rel_name	表的名字。
sql_command	执行的SQL语句。
started_time	SQL语句开始执行的时间。
elapsed_time	截止查询时已执行时间。
phase_name	在线DDL执行阶段的名称，包括“Initialization”、“Base data copy”、“Reindexing”、“Catchup <N>”和“Last catchup”五类。
phase_started_time	该阶段开始的时间。
phase_elapsed_time	该阶段实际已经执行的时间。
lock_level	锁级别。
lock_try_times	拿锁的尝试次数。取值范围： ● 大于等于1：表示已经获取到锁。 ● 0：表示尚未获取到锁，或获取到锁但进度信息未实时更新。
lock_hold_time	拿锁的持续时间。
phase_detail	当前阶段的具体信息。JSON格式，各个阶段展示的信息可能不同。

- 初始化（Initialization）阶段phase_detail的信息为空。

版权所有 © 华为云计算技术有限公司

- 基础数据同步（Base data copy）阶段phase_detail包括以下字段：

参数	描述
total_pages	预计需要插入的page数量。
inserted_rows	实际插入的行数，在阶段结束后更新。
insert_speed	实际插入速率，在阶段结束后更新。
parallel_threads	本阶段待采用或实际采用的parallel_threads参数值。如果表中含有表达式索引，则无法并行建索引，parallel_threads始终为1。

- 重建索引（Reindexing）阶段phase_detail包括以下字段：

参数	描述
total_indexes	总共需要重建的索引。
indexes_rebuilt	当前已经重建的索引数量。
parallel_threads	本阶段待采用或实际采用的parallel_threads参数值。如果表中含有表达式索引，则无法并行建索引，parallel_threads始终为1。
parallel_workers	源表的parallel_workers选项，详细信息请参见**ALTER TABLE**章节中对parallel_workers的描述。

说明

在线DDL执行期间，会对生成的临时表额外创建2个索引（临时表位于“online$$ddl$$[（表所在的模式+表名）的hash值]”schema下）。因此，total_index的值等于用户创建的索引数量加2。

并行构建索引时，不能保证indexes_rebuilt的值实时更新。

- 追增（Catchup <N>）阶段phase_detail包括以下字段：

参数	描述
short_lock_level	追增阶段开始时短暂持有强锁的锁级别。
short_lock_try_times	追增阶段开始时尝试获取强锁的重试次数。
short_lock_hold_time	追增阶段开始时短暂持有强锁的持续时间。
total_delete_pages	预计需要删除的page数量。
total_insert_pages	预计需要插入的page数量。
deleted_rows	实际删除的行数，在阶段结束后更新。
delete_speed	实际删除的速率，在阶段结束后更新。

版权所有 © 华为云计算技术有限公司

参数	描述
inserted_rows	实际插入的行数，在阶段结束后更新。
insert_speed	实际插入的速率，在阶段结束后更新。
last_catchup_threshold	本阶段执行时，待采用或实际采用的last_catchup_threshold参数值。
max_catchup_times	本阶段执行时，待采用或实际采用的max_catchup_times参数值。

- 最后一轮追增（Last catchup）阶段phase_detail包括以下字段：

参数	描述
total_delete_pages	预计需要删除的page数量。
total_insert_pages	预计需要插入的page数量。
deleted_rows	实际删除的行数，在阶段结束后更新。
delete_speed	实际删除的速率，在阶段结束后更新。
inserted_rows	实际插入的行数，在阶段结束后更新。
insert_speed	实际插入的速率，在阶段结束后更新。
last_catchup_threshold	本阶段执行时，待采用或实际采用的last_catchup_threshold参数值。
max_catchup_times	本阶段执行时，待采用或实际采用的max_catchup_times参数值。
total_gpis	需要重建的全局分区索引（Global Partition Index，GPI）数量。仅在DDL为分区级VACUUM FULL时展示。
gpis_rebuilt	已重建的GPI数量。在当前版本中，该参数不会实时更新，仅在所有GPI重建完成后，其值一次性更新为total_GPIs。该字段仅当在线VACUUM FULL操作为分区级时展示。
parallel_workers	源表的parallel_workers选项，详细信息请参见**ALTER TABLE**章节中对parallel_workers的描述。

说明

重建索引阶段的parallel_workers由在线DDL参数parallel_threads和表的大小共同影响。但在最后一轮追增过程中，parallel_workers不受在线DDL的parallel_threads参数影响。

- gs_online_ddl_set_parameters(relschema, relname, params)
  描述：用于在ONLINE DDL或ONLINE VACUUM FULL执行过程中动态修改参数。
  参数：

版权所有 © 华为云计算技术有限公司

- relschema: 表的模式名，类型为cstring，不可为空。
- relname：表名，类型为cstring，不可为空。
- params: 可选参数，类型为变长text[]，表示要修改的参数键值对，格式为'key', 'value'，一次可更新多个参数。
  - 若提供params参数，则按键值对进行修改，最多支持7对键值对。其中，key必须为系统支持的参数，value必须为该参数支持范围内的有效值。
  - 若不提供params参数，则执行查询功能，返回该函数支持的所有参数值。
  - 若同一个参数被多次输入、参数值不在支持范围，或键值对数量超过7对，则报错并提示params参数错误。

说明

- 参数校验规则：仅处理白名单里的参数和对应合法值，否则将报错处理。
- 权限管控规则：该系统函数仅限系统管理员、初始用户和VACUUM FULL执行用户才可以修改和查询参数信息。

返回值类型：一行多列text，返回当前指定表上正在执行的ONLINE DDL的所有参数值。

示例：

```
gaussdb=# CREATE TABLE t1(c1 int, c2 int);

--会话1：执行在线Vacuum full
gaussdb=# VACUUM FULL t1 online;
--会话2:
gaussdb=# SELECT * from gs_online_ddl_set_parameters('public', 't1', 'last_catchup_threshold', '10000', 'max_catchup_times', '30','parallel_threads', '2','online_ddl_lockwait_timeout', '10', 'lockwait_interval', '10', 'trylock_threshold', '1', 'enable_cancel', 'true');
 parallel_threads | last_catchup_threshold | max_catchup_times | online_ddl_lockwait_timeout | lockwait_interval | trylock_threshold | enable_cancel
------------------+------------------------+-------------------+-----------------------------+-------------------+-------------------+---------------
 2                | 10000                  | 30                | 10                          | 10                | 1                 | true
(1 row)
```

具体支持的参数如下：

- parallel_threads：控制多线程并行，用于在资源充足的情况下追求更高性能。取值范围为1~32的整数，缺省时默认值为1。
  - 当用户开启资源管控（如io_limits > 0或io_priority = high/medium/low）时，parallel_threads将忽略用户/会话级别的资源管控，并提示当前资源管控无效，同时开启多线程加速。在线VACUUM FULL的多线程执行实际遵循初始用户的资源管控。
  - 该参数可与索引并行参数parallel_workers共同生效：parallel_workers控制VACUUM FULL期间索引构建的并行度，而parallel_threads控制VACUUM FULL期间追增阶段的并行度。
- max_catchup_times：控制最大追增轮次，用于防止VACUUM FULL执行时间过长。取值范围为1至INT_MAX之间的整数值，缺省时默认值为50。
  - 在线VACUUM FULL会记录数据重建过程中的IUD增量数据到临时表，并采用多轮次循环追增，将增量数据进行重建。为防止由于IUD并发数据量过大导致VACUUM FULL无法完成追增的情况，用户可以设置该参数以限制追增次数。

版权所有 © 华为云计算技术有限公司

  - last_catchup_threshold：触发最后一轮的追增时间上限。当完成一轮追增后，若该轮耗时小于该阈值，即可触发最后一轮追增和切换。取值范围为[1000,1200000]，单位为毫秒，默认值为5000。
  - online_ddl_lockwait_timeout：在线DDL的等锁超时时间。若等锁线程在该时间内未能获取锁，则报锁超时失败。触发CANCEL模式拿锁时，实际使用的锁超时时间为此参数设置的2倍。取值范围为[1,1200000]，单位为毫秒，默认值为10000。
  - lockwait_interval：等锁步长。即连续拿锁超时时的重试间隔。若当前轮次等锁超时，则等待该参数指定的时长后再进行下一轮尝试。取值范围为[0,300]，单位为秒，默认值为5。
  - trylock_threshold：等锁尝试次数上限，需与enable_cancel参数同时使用。当达到该上限后若仍因等锁超时而无法获取锁，则下一次尝试将采用CANCEL模式。若未开启enable_cancel，达到上限后取锁失败并报错。取值范围为[1,INT_MAX]，或-1（表示无上限），单位为次。默认值为-1。
  - enable_cancel：是否允许用户业务快速失败，即在达到等锁尝试次数上限，或在切表事务获取用户表8级锁时，通过快速中止用户业务来确保在线DDL成功获取锁。取值范围为[false,true]，默认值为false。

- anynesttable_out(anynesttable)

  描述：将anynesttable转换为cstring。

  参数：anynesttable

  返回类型：cstring

- nesttable_count(a nesttable)

  描述：返回参数nesttable中的元素数量。

  参数：nesttable

  返回类型：int

- nesttable_delete(a nesttable)

  描述：删除参数nesttable中的所有元素。

  参数：nesttable

  返回类型：nesttable

- nesttable_delete(a nesttable, index int)

  描述：若给定下标index在nesttable范围之内，则删除nesttable中给定下标为index的元素。

  参数：nesttable、int

  返回类型：nesttable

- nesttable_delete(a nesttable, lower int, upper int)

  描述：若给定下标lower和upper在nesttable范围之内，则删除nesttable中给定下标范围[lower, upper]内的元素。

  参数：nesttable、int、int

  返回类型：nesttable

- nesttable_eq(a nesttable, b nesttable)

  描述：比较两个nesttable是否相等。

  参数：nesttable、nesttable

  返回类型：bool

版权所有 © 华为云计算技术有限公司

- nesttable_except(a nesttable, b nesttable)
  描述：获取两个nesttable参数的差集。
  参数：nesttable、nesttable
  返回类型：nesttable
- nesttable_except_distinct(a nesttable, b nesttable)
  描述：获取两个nesttable参数去重后的差集。
  参数：nesttable、nesttable
  返回类型：nesttable
- nesttable_exists(a nesttable, index int)
  描述：查找参数nesttable中指定下标index是否存在元素。
  参数：nesttable、int
  返回类型：nesttable
- nesttable_extend(a nesttable, cnt int)
  描述：在nesttable变量末尾扩展cnt个元素空间。
  参数：nesttable、int
  返回类型：nesttable
- nesttable_extend(a nesttable, cnt int, index int)
  描述：将nesttable变量末尾扩展cnt个元素空间，扩展元素的值等于给定index下标元素的值。
  参数：nesttable、int、int
  返回类型：nesttable
- nesttable_first(a nesttable)
  描述：返回参数nesttable中第一个有效元素的下标，若没有则返回NULL。
  参数：nesttable
  返回类型：int
- nesttable_in(str cstring, oid Oid, typmod int)
  描述：将字符串转换为指定oid的nesttable。
  参数：cstring、Oid、int
  返回类型：nesttable
- nesttable_intersect(a nesttable, b nesttable)
  描述：获取两个nesttable参数的交集。
  参数：nesttable、nesttable
  返回类型：nesttable
- nesttable_intersect_distinct(a nesttable, b nesttable)
  描述：获取两个nesttable参数去重后的交集。
  参数：nesttable、nesttable
  返回类型：nesttable
- nesttable_last(a nesttable)
  描述：返回参数nesttable中最后一个有效元素的下标，若没有则返回NULL。
  参数：nesttable
  返回类型：int

版权所有 © 华为云计算技术有限公司

- nesttable_ne(a nesttable, b nesttable)
  描述：比较两个nesttable是否不相等。
  参数：nesttable、nesttable
  返回类型：bool
- nesttable_next(a nesttable, index int)
  描述：返回参数nesttable中指定下标index的后一个有效元素的下标，若没有则返回NULL。
  参数：nesttable、int
  返回类型：int
- nesttable_out(a nesttable)
  描述：将nesttable转换为cstring。
  参数：nesttable
  返回类型：cstring
- nesttable_prior(a nesttable, index int)
  描述：返回参数nesttable中指定下标index的前一个有效元素的下标，若没有则返回NULL。
  参数：nesttable、int
  返回类型：int
- nesttable_trim(a nesttable, cnt int)
  描述：删除nesttable尾部的cnt个元素并释放对应的存储空间。
  参数：nesttable、int
  返回类型：int
- nesttable_union(a nesttable, b nesttable)
  描述：获取两个nesttable参数的并集。
  参数：nesttable、nesttable
  返回类型：nesttable
- nesttable_union_distinct(a nesttable, b nesttable)
  描述：获取两个nesttable参数去重后的并集。
  参数：nesttable、nesttable
  返回类型：nesttable
- anyindexbytable_out(anyindexbytable)
  描述：将anyindexbytable转换为cstring。
  参数：anyindexbytable
  返回类型：cstring
- indexbytable_count(a indexbytable)
  描述：返回indexbytable参数中的元素数量。
  参数：indexbytable
  返回类型：int
- indexbytable_in(str cstring, oid Oid)
  描述：将字符串转换为指定oid的indexbytable。
  参数：cstring、Oid

版权所有 © 华为云计算技术有限公司

返回类型：indexbytable

- indexbytable_out(a indexbytable)
  描述：将indexbytable转换为cstring。
  参数：indexbytable
  返回类型：cstring
- indexbytableint_delete(a indexbytable)
  描述：删除参数indexbytable中的所有元素。
  参数：indexbytable
  返回类型：indexbytable
- indexbytableint_delete(a indexbytable, index int)
  描述：若给定下标index在indexbytable中，则删除indexbytable中给定下标为index的元素。
  参数：indexbytable、int
  返回类型：indexbytable
- indexbytableint_delete(a indexbytable, lower int, upper int)
  描述：删除indexbytable中给定下标范围[lower, upper]内的元素。
  参数：indexbytable、int、int
  返回类型：indexbytable
- indexbytableint_exists(a indexbytable, index int)
  描述：查找参数indexbytable中指定下标index是否存在元素。
  参数：indexbytable、int
  返回类型：indexbytable
- indexbytableint_first(a indexbytable)
  描述：返回参数indexbytable中第一个有效元素的下标，若没有则返回NULL。
  参数：indexbytable
  返回类型：int
- indexbytableint_last(a indexbytable)
  描述：返回参数indexbytable中最后一个有效元素的下标，若没有则返回NULL。
  参数：indexbytable
  返回类型：int
- indexbytableint_next(a indexbytable, index int)
  描述：返回参数indexbytable中指定下标index的后一个有效元素的下标，若没有则返回NULL。
  参数：indexbytable、int
  返回类型：int
- indexbytableint_prior(a indexbytable, index int)
  描述：返回参数indexbytable中指定下标index的前一个有效元素的下标，若没有则返回NULL。
  参数：indexbytable、int
  返回类型：int
- indexbytablevarchar_delete(a indexbytable)

版权所有 © 华为云计算技术有限公司

描述：删除参数indexbytable中的所有元素。

参数：indexbytable

返回类型：indexbytable

- indexbytablevarchar_delete(a indexbytable, index varchar)

  描述：若给定下标index在indexbytable中，则删除indexbytable中给定下标为index的元素。

  参数：indexbytable、varchar

  返回类型：indexbytable

- indexbytablevarchar_delete(a indexbytable, lower varchar, upper varchar)

  描述：删除indexbytable中给定下标范围[lower, upper]内的元素。

  参数：indexbytable、varchar、varchar

  返回类型：indexbytable

- indexbytablevarchar_exists(a indexbytable, index varchar)

  描述：查找参数indexbytable中指定下标index是否存在元素。

  参数：indexbytable、varchar

  返回类型：indexbytable

- indexbytablevarchar_first(a indexbytable)

  描述：返回参数indexbytable中第一个有效元素的下标，若没有则返回NULL。

  参数：indexbytable

  返回类型：varchar

- indexbytablevarchar_last(a indexbytable)

  描述：返回参数indexbytable中最后一个有效元素的下标，若没有则返回NULL。

  参数：indexbytable

  返回类型：varchar

- indexbytablevarchar_next(a indexbytable, index varchar)

  描述：返回参数indexbytable中指定下标index的后一个有效元素的下标，若没有则返回NULL。

  参数：indexbytable、varchar

  返回类型：varchar

- indexbytablevarchar_prior(a indexbytable, index varchar)

  描述：返回参数indexbytable中指定下标index的前一个有效元素的下标，若没有则返回NULL。

  参数：indexbytable、varchar

  返回类型：varchar

- gs_partitioned_temp_fix_filenode()

  描述：用于版本升级过程中，为旧版本创建的分区表和LOCAL索引本身创建物理文件，若创建成功则返回true，若原对象（旧版本创建的分区表和LOCAL索引本身）已存在物理文件，则返回false。该函数仅可以在升级期间由系统管理员执行。

  参数：regclass

  返回值类型：boolean

版权所有 © 华为云计算技术有限公司

- double_backslash(input)

  描述：用于LIKE匹配单个反斜杠字符逻辑中，双写字符串中的反斜杠字符，input可以为text、bpchar、bytea、raw、byteawithoutorderwithequalcol类型。

  参数：text | bpchar | bytea | raw | byteawithoutorderwithequalcol

  返回值类型：与输入参数类型相同。

1.6.61 废弃函数

GaussDB中下列函数在当前版本中已废弃：

- gs_wlm_get_session_info
- gs_wlm_get_user_session_info
- check_engine_status
- encode_plan_node
- model_train_opt
- gs_stat_get_wlm_plan_operator_info
- track_model_train_opt
- array_extend
- dbe_perf.global_slow_query_info
- dbe_perf.global_slow_query_info_bytime
- dbe_perf.global_slow_query_history
- pg_reload_conf
- pg_rotate_logfile
- gs_stat_ustore
- local_segment_space_info
- global_segment_space_info
- remote_segment_space_info
- pg_stat_segment_extent_usage
- pgxc_get_wlm_current_instance_info(text, int default null)
- pgxc_get_wlm_history_instance_info(text, TIMESTAMP, TIMESTAMP, int default null)
- GS_ALL_NODEGROUP_CONTROL_GROUP_INFO(text)
- create_wlm_operator_info(int flag)
- create_wlm_session_info(int flag)
- pg_stat_get_wlm_session_info(int flag)
- gs_wlm_get_resource_pool_info(int)
- gs_wlm_get_all_user_resource_info()
- gs_wlm_get_workload_records()
- gs_wlm_persistent_user_resource_info()
- gs_wlm_session_respool(bigint)
- pgxc_get_wlm_operator_history()

版权所有 © 华为云计算技术有限公司

- pgxc_get_wlm_operator_info()
- pgxc_get_wlm_operator_statistics()
- pgxc_get_wlm_session_history()
- pgxc_get_wlm_session_info()
- pgxc_get_wlm_session_info_bytime(tag text, begin timestamp, end timestamp, limit int)
- pgxc_get_wlm_session_statistics()
- pgxc_wlm_get_workload_records()
- DBE_PERF.get_wlm_controlgroup_ng_config()
- DBE_PERF.get_global_wlm_workload_runtime()
- gs_wlm_user_resource_info(name text)
- create_wlm_instance_statistics_info()
- pg_stat_get_session_wlmstat()
- pg_stat_get_wlm_instance_info()
- pg_stat_get_wlm_instance_info_with_cleanup()
- pg_stat_get_wlm_node_resource_info()
- pg_stat_get_wlm_operator_info()
- pg_stat_get_wlm_realtime_operator_info()
- pg_stat_get_wlm_realtime_session_info()
- pg_stat_get_wlm_session_info_internal()
- pg_stat_get_wlm_statistics()
- gs_redis_set_bucketxid(bigint)
- gs_check_logic_cluster_consistency()
- gs_upload_obs_file('slot_name', 'src_file', 'dest_file')
- gs_download_obs_file('slot_name', 'src_file', 'dest_file')
- gs_get_obs_file_context('file_name', 'slot_name')
- gs_set_obs_file_context('file_name', 'file_context','slot_name')
- gs_get_hadr_key_cn()
- gs_hadr_has_barrier_creator()
- gs_hadr_do_switchover()
- track_memory_context(context_list text)
- track_memory_context_detail()
- gs_stat_remain_segment_info()
- gs_local_stat_remain_segment_info()
- gs_free_remain_segment()
- gs_local_free_remain_segment()
- copy_error_log_add_custom_id
- populate_record
- cupointer_bigint

版权所有 © 华为云计算技术有限公司

- pg_autovac_coordinator
- mot_global_memory_detail
- mot_jit_detail
- mot_jit_profile
- mot_local_memory_detail
- mot_session_memory_detail
- gs_imcv_status()
- gs_imcv_flush(reloid oid)
- gs_htap_tmu_data(rel_oid oid, rowgroup_id integer)
- gs_htap_tmu_chunk_meta(rel_oid oid, rowgroup_id integer)
- gs_imcu_meta(rel_name)
- gs_htap_file_status(db_oid oid)
- gs_htap_scan_hit_status(reloid oid)
- gs_htap_common_debug(type integer, info text)
- gs_imcv_taskbuffer_view()
- gs_imcu_slot_status()
- prepare_statement_status（ ）
- DBE_PERF.global_prepare_statement_status()
- pg_replication_origin_create()
- pg_replication_origin_drop()
- pg_replication_origin_oid()
- pg_replication_origin_session_setup()
- pg_replication_origin_session_reset()
- pg_replication_origin_session_is_setup()
- pg_replication_origin_session_progress()
- pg_replication_origin_xact_setup()
- pg_replication_origin_xact_reset()
- pg_replication_origin_advance()
- pg_replication_origin_progress()
- pg_show_replication_origin_status()

1.6.62 向量数据函数与操作符

1.6.62.1 向量距离计算接口

l2_distance

功能说明：计算两个向量的欧式距离。

入参1的类型：floatvector

入参2的类型：floatvector

版权所有 © 华为云计算技术有限公司

出参类型：float8

代码示例：

```
gaussdb=# SELECT l2_distance(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT l2_distance('[1,2,3]',  '[5,-1,3.5]');
```

vector_l2_squared_distance

功能说明：获得两个向量的欧式距离的平方。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：float8

代码示例：

```
gaussdb=# SELECT vector_l2_squared_distance(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_l2_squared_distance('[1,2,3]',  '[5,-1,3.5]');
```

cosine_distance

功能说明：计算两个向量的余弦距离。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：float8

代码示例：

```
gaussdb=# select cosine_distance(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# select cosine_distance('[1,2,3]', '[5,-1,3.5]');
```

vector_spherical_distance

功能说明：计算两个归一化向量的球面距离（余弦夹角的弧度制表示）。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：float8

代码示例：

```
gaussdb=#  select vector_spherical_distance('[1,0,0,0]', '[0,0,0,1]');
```

说明

如果输入向量非归一化，无法获得正确的计算结果。

1.6.62.2 向量操作函数接口

向量操作函数实现的功能包括：向量大小比较、向量加法、向量减法、向量按位乘法等。

版权所有 © 华为云计算技术有限公司

inner_product

功能说明：计算两个向量的内积。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：float8

代码示例：

```
gaussdb=# SELECT inner_product(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT inner_product('[1,2,3]', '[5,-1,3.5]');
```

vector_negative_inner_product

功能说明：计算两个向量的负内积。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：float8

代码示例：

```
gaussdb=# SELECT vector_negative_inner_product(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_negative_inner_product('[1,2,3]', '[5,-1,3.5]');
```

vector_dims

功能说明：返回floatvector向量的维度值。

入参类型：floatvector

出参类型：int4

代码示例：

```
gaussdb=# SELECT vector_dims(floatvector('[1,2,3]'));
gaussdb=# SELECT vector_dims('[1,2,3]');
```

vector_norm

功能说明：返回向量的L2范数。

入参类型：floatvector

出参类型：float8

代码示例：

```
gaussdb=# SELECT vector_norm(floatvector('[1,2,3]'));
gaussdb=# SELECT vector_norm('[1,2,3]');
```

vector_add

功能说明：计算两个向量的和。

入参1的类型：floatvector

入参2的类型：floatvector

版权所有 © 华为云计算技术有限公司

出参类型：floatvector

代码示例：

```
gaussdb=# SELECT vector_add(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_add('[1,2,3]', '[5,-1,3.5]');
```

vector_sub

功能说明：计算两个向量的差。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：floatvector

代码示例：

```
gaussdb=# SELECT vector_sub(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_sub('[1,2,3]', '[5,-1,3.5]');
```

vector_lt

功能说明：比较两个向量的大小，向量1是否小于向量2。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT vector_lt(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_lt('[1,2,3]', '[5,-1,3.5]');
```

vector_le

功能说明：比较两个向量的大小，向量1是否小于等于向量2。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT vector_le(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_le('[1,2,3]', '[5,-1,3.5]');
```

vector_eq

功能说明：比较两个向量是否相等。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：BOOLEAN

代码示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT vector_eq(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_eq('[1,2,3]', '[5,-1,3.5]');
```

vector_ne

功能说明：比较两个向量是否不等。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT vector_ne(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_ne('[1,2,3]', '[5,-1,3.5]');
```

vector_ge

功能说明：比较两个向量的大小，向量1是否大于等于向量2。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT vector_ge(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_ge('[1,2,3]', '[5,-1,3.5]');
```

vector_gt

功能说明：比较两个向量的大小，向量1是否大于向量2。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT vector_gt(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_gt('[1,2,3]', '[5,-1,3.5]');
```

vector_cmp

功能说明：比较两个向量的大小。

入参1的类型：floatvector

入参2的类型：floatvector

出参类型：int4

代码示例：

```
gaussdb=# SELECT vector_cmp(floatvector('[1,2,3]'), floatvector('[5,-1,3.5]'));
gaussdb=# SELECT vector_cmp('[1,2,3]', '[5,-1,3.5]');
```


版权所有 © 华为云计算技术有限公司

vector_accum

功能说明：返回向量累加。

入参1的类型：anyarray

入参2的类型：floatvector

出参类型：anyarray

代码示例：

```
-- 系统函数，不推荐使用，若需使用，数组元素类型必须为float8类型。
gaussdb=# SELECT vector_accum(array[cast(3 as float8),1,2,3], floatvector('[5,-1,3.5]'));
```

vector_combine

功能说明：合并向量。

入参1的类型：anyarray

入参2的类型：anyarray

出参类型：anyarray

代码示例：

```
-- 系统函数，不推荐使用，若需使用，数组元素类型必须为float8类型。
gaussdb=# SELECT vector_combine(array[cast(1 as float8),2,3], array[cast(1 as float8),2,3]);
```

vector_avg

功能说明：平均向量。

入参类型：anyarray

出参类型：floatvector

代码示例：

```
-- 系统函数，不推荐使用，若需使用，数组元素类型必须为float8类型。
gaussdb=# SELECT vector_avg(array[cast(1 as float8),2,3]);
```

bool_vector_dims

功能说明：返回boolvector向量的维度值。

入参类型：boolvector

出参类型：int4

代码示例：

```
gaussdb=# SELECT bool_vector_dims(boolvector('[1,1,1]'));
```

bool_vector_cmp

功能说明：比较两个布尔向量的大小。

入参1的类型：boolvector

入参2的类型：boolvector

版权所有 © 华为云计算技术有限公司

出参类型：int4

代码示例：

```
gaussdb=# SELECT bool_vector_cmp('[1,0,0]','[1,1,0]');
gaussdb=# SELECT bool_vector_cmp('[1,1,0]','[1,1,0]');
```

bool_vector_eq

功能说明：比较两个布尔向量是否相等。

入参1的类型：boolvector

入参2的类型：boolvector

出参类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT bool_vector_eq(boolvector('[1,1,1]'), boolvector('[1,1,1]'));
gaussdb=# SELECT bool_vector_eq('[1,1,1]', '[1,1,1]');
```

1.6.62.3 向量函数和操作符

floatvector支持向量类型和数组类型之间的数据转换，同时支持特定格式的字符串转换成向量类型。

array<->floatvector：向量数据类型可以和相对应的数组类型进行自由转换；floatvector向量的成员数据类型为浮点型。

string ->floatvector：注意字符串格式，使用中括号（[]）或者花括号（{}）包含数组，元素间使用逗号（,）隔开。

floatvector

功能说明：

场景1：数组数据转换为向量数据。

入参类型：anyarray

出参类型：floatvector

代码示例：

```
gaussdb=# SELECT floatvector(ARRAY[1,2,9.3]);
```

场景2：floatvector转换，对维度进行检测。

入参1的类型：floatvector

入参2的类型：integer

出参类型：floatvector

代码示例：

```
gaussdb=# SELECT floatvector(floatvector('[1,2,9.3]'),3);
```

vector_to_array

功能说明：向量数据转换为数组数据。

版权所有 © 华为云计算技术有限公司

入参类型：floatvector

出参类型：real[]

代码示例：

```
gaussdb=# SELECT vector_to_array(floatvector('[1,2,3]'));
```

text_to_vector

功能说明：字符型数据转换为向量数据。

入参类型：cstring

出参类型：floatvector

代码示例：

```
gaussdb=# SELECT text_to_vector('[1,2,9.3]');
```

vector_in

功能说明：将文本格式的数据转换为内部floatvector类型。

入参1的类型：cstring

入参2的类型：oid

入参3的类型：int4

出参类型：floatvector

代码示例：

```
gaussdb=# SELECT vector_in('[1,2,3]',701,3);
```

vector_out

功能说明：将内部floatvector类型的向量数据转换为文本格式。

入参类型：floatvector

出参类型：cstring

代码示例：

```
gaussdb=# SELECT vector_out(floatvector('[1,2,3]'));
```

vector_send

功能说明：将内部floatvector类型的向量数据转换为bytea类型。

入参类型：floatvector

出参类型：bytea

代码示例：

```
gaussdb=# SELECT vector_send(floatvector('[1,2,3]'));
```


版权所有 © 华为云计算技术有限公司

vector_recv

功能说明：将二进制格式的数据转换为内部floatvector类型。

入参1的类型：integer

入参2的类型：oid

入参3的类型：int4

出参类型：floatvector

代码示例：

```
--系统函数，无法sql调用。
```

vector_typmod_in

功能说明：floatvector输入类型修改符函数。

入参类型：cstring[]

出参类型：integer

代码示例：

```
gaussdb=# SELECT vector_typmod_in( '{1}' );
```

说明

- 向量数据类型成员仅支持单精度。
- 向量间计算仅支持相同维度，如果维度不同将报错。
- floatvector支持向量加减操作，点乘操作由函数（inner_product）完成。
- 建表时向量类型必须要指定维度；当插入数据的维度与设置的数据维度不一致时，系统会报错。
- 需要注意如果数据表中已经插入数据，变换向量维度时需要表中数据符合维度设定，否则数据库报错；当向量属性已创建向量索引，属性的数据类型无法切换到其他数据类型，数据库会报错。

boolvector

场景1：

功能说明：数组数据转换为向量数据。

入参类型：anyarray

出参类型：boolvector

代码示例：

```
gaussdb=# SELECT boolvector(ARRAY[1,0,1]);
```

场景2：

功能说明：boolvector转换，对维度进行检测。

入参1的类型：boolvector

入参2的类型：integer

出参类型：boolvector

版权所有 © 华为云计算技术有限公司

代码示例：

```
gaussdb=# SELECT boolvector(boolvector('[1,0,1]'),3);
```

boolvector_to_array

功能说明：向量数据转换为数组数据。

入参类型：boolvector

出参类型：anyarray

代码示例：

```
gaussdb=# SELECT boolvector_to_array(boolvector('[1,0,1]'));
```

text_to_boolvector

功能说明：字符型数据转换为向量数据。

入参类型：cstring

出参类型：boolvector

代码示例：

```
gaussdb=# SELECT text_to_boolvector('[1,1,1]');
```

bool_vector_in

功能说明：将文本格式的数据转换为内部boolvector类型。

入参1的类型：cstring

入参2的类型：oid

入参3的类型：int4

出参类型：boolvector

代码示例：

```
gaussdb=# SELECT bool_vector_in('[1,1,1]',701,3);
```

bool_vector_out

功能说明：将内部boolvector类型的向量数据转换为文本格式。

入参类型：boolvector

出参类型：cstring

代码示例：

```
gaussdb=# SELECT bool_vector_out(boolvector('[1,1,1]'));
```

bool_vector_send

功能说明：将内部boolvector类型的向量数据转换为bytea类型。

入参类型：boolvector

版权所有 © 华为云计算技术有限公司

出参类型：bytea

代码示例：

```
gaussdb=# SELECT bool_vector_send(boolvector('[1,1,1]'));
```

bool_vector_recv

功能说明：将二进制格式的数据转换为内部boolvector类型。

入参1的类型：internal

入参2的类型：oid

入参3的类型：int4

出参类型：boolvector

代码示例：

```
--系统函数，无法sql调用。
```

bool_vector_typmod_in

功能说明：boolvector输入类型修改符函数。

入参类型：cstring[]

出参类型：integer

代码示例：

```
gaussdb=# SELECT bool_vector_typmod_in( '{1}' );
```

说明

- 向量数据类型不支持Null、Nan、Inf作为元素，当向量中含有NULL值，数据库会报错。
- 插入向量数据类型时，不支持NULL作为插入值，当插入NULL值作为向量数据时，数据库会报错。
- boolvector类型的元素可使用t(T)\f(F)、y(Y)\n(N)、1\0、true\false、yes\no、on\off等方式表达布尔型数据。
- boolvector仅支持等于操作，不支持大小比较。

gs_vector_index_options

功能说明：显示相关向量索引的超参取值。

入参类型：text

出参类型：text

代码示例：

```
--创建表。
gaussdb=# CREATE TABLE t1 (id int unique,repr floatvector(960));
插入数据：
gaussdb=# CREATE OR REPLACE FUNCTION float_random_array(dims int,range int) RETURNS float[] AS $
$BEGIN RETURN ARRAY(SELECT (random() * range)::int FROM generate_series(1, dims));END;$$
LANGUAGE plpgsql;
gaussdb=# INSERT INTO t1 SELECT i,floatvector(float_random_array(960,100)) FROM
generate_series(1,1000) as i;
创建索引:
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE INDEX test1v on t1 using gsdiskann (repr l2) with
(pq_nseg=120,pq_nclus=64,queue_size=120,num_parallels=30,enable_pq=true,using_clustering_for_parallel=f
alse);

gaussdb=# SELECT gs_vector_index_options('test1v');
```

gs_diskann_inspect

功能说明：查询索引健康状态，包括索引磁盘使用、PQ压缩准确度、索引图连接健康度、Vacuum状态等信息。

场景1：

入参类型：text

出参类型：record

代码示例：

```
-- 创建表
gaussdb=# CREATE TABLE t1 (id int unique,repr floatvector(960));
-- 插入数据
gaussdb=# CREATE OR REPLACE FUNCTION float_random_array(dims int,range int) RETURNS float[] AS $
$BEGIN RETURN ARRAY(SELECT (random() * range)::int FROM generate_series(1, dims));END;$$
LANGUAGE plpgsql;
gaussdb=# INSERT INTO t1 SELECT i,floatvector(float_random_array(960,100)) FROM
generate_series(1,1000) as i;
-- 创建向量索引
gaussdb=# CREATE INDEX test1v on t1 using gsdiskann (repr l2) with
(pq_nseg=120,pq_nclus=64,queue_size=120,num_parallels=30,enable_pq=true,using_clustering_for_parallel=f
alse);
-- 显示指定DN上索引状态
gaussdb=# EXECUTE DIRECT ON (datanode1) 'SELECT * FROM gs_diskann_inspect(''test1v'');';
-- 显示所有DN上索引状态
gaussdb=# SELECT * FROM gs_diskann_inspect('test1v');
```

场景2：

入参1的类型：text

入参2的类型：text

出参类型：record

代码示例：

```
-- 创建分区表
gaussdb=# CREATE TABLE t1 (id int unique,repr floatvector(960)) partition by hash(id) (partition p1,
partition p2, partition p3);
-- 插入数据
gaussdb=# CREATE OR REPLACE FUNCTION float_random_array(dims int,range int) RETURNS float[] AS $
$BEGIN RETURN ARRAY(SELECT (random() * range)::int FROM generate_series(1, dims));END;$$
LANGUAGE plpgsql;
gaussdb=# INSERT INTO t1 SELECT i,floatvector(float_random_array(960,100)) FROM
generate_series(1,1000) as i;
-- 创建分区向量索引
gaussdb=# CREATE INDEX test1v on t1 using gsdiskann (repr l2) local with
(pq_nseg=120,pq_nclus=64,queue_size=120,num_parallels=30,enable_pq=true,using_clustering_for_parallel=f
alse);
-- 查询分区名和分区索引名
SELECT oid,relname,parttype,parentid,boundaries FROM pg_partition WHERE parentid = 't1'::regclass::oid;
SELECT oid,relname,parttype,parentid,boundaries,indextblid FROM pg_partition WHERE parentid =
'test1v'::regclass::oid;
-- 显示指定DN上指定分区的索引状态
gaussdb=# EXECUTE DIRECT ON (datanode1) 'SELECT * FROM gs_diskann_inspect(''test1v'', ''p1_repr_idx'');';
-- 显示所有DN上指定分区的索引状态
gaussdb=# SELECT * FROM gs_diskann_inspect('test1v', 'p1_repr_idx');
```


版权所有 © 华为云计算技术有限公司

场景3：

入参1的类型：text

入参2的类型：text

入参3的类型：text

出参类型：record

代码示例：

```
-- 创建分区表
gaussdb=# CREATE TABLE t1 (id int unique,repr floatvector(960)) partition by hash(id) (partition p1,
partition p2, partition p3);
-- 插入数据
gaussdb=# CREATE OR REPLACE FUNCTION float_random_array(dims int,range int) RETURNS float[] AS $
$BEGIN RETURN ARRAY(SELECT (random() * range)::int FROM generate_series(1, dims));END;$$
LANGUAGE plpgsql;
gaussdb=# INSERT INTO t1 SELECT i,floatvector(float_random_array(960,100)) FROM
generate_series(1,1000) as i;
-- 创建分区向量索引
gaussdb=# CREATE INDEX test1v on t1 using gsdiskann (repr l2) local with
(pq_nseg=120,pq_nclus=64,queue_size=120,num_parallels=30,enable_pq=true,using_clustering_for_parallel=f
alse);
-- 查询分区名、分区索引名和DN名
SELECT oid,relname,parttype,parentid,boundaries FROM pg_partition WHERE parentid = 't1'::regclass::oid;
SELECT oid,relname,parttype,parentid,boundaries,indextblid FROM pg_partition WHERE parentid =
'test1v'::regclass::oid;
SELECT oid,node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
-- 显示指定DN上指定分区的索引状态
gaussdb=# SELECT * FROM gs_diskann_inspect('test1v', 'p1_repr_idx', 'datanode1');
```

gs_ivfflat_inspect

功能说明：查询Gslvfflat索引状态，包括固定页面号、索引磁盘使用、聚簇中向量数据分布等信息。需连接到DN节点或者使用EXECUTE DIRECT ON语法执行。

场景1：

入参类型：text

出参类型：record

代码示例：

```
-- 创建表
gaussdb=# CREATE TABLE t1 (id int unique,repr floatvector(960));
-- 插入数据
gaussdb=# CREATE OR REPLACE FUNCTION float_random_array(dims int,range int) RETURNS float[] AS $
$BEGIN RETURN ARRAY(SELECT (random() * range)::int FROM generate_series(1, dims));END;$$
LANGUAGE plpgsql;
gaussdb=# INSERT INTO t1 SELECT i,floatvector(float_random_array(960,100)) FROM
generate_series(1,1000) as i;
-- 创建向量索引
gaussdb=# SET maintenance_work_mem='128MB';
gaussdb=# CREATE INDEX test1v on t1 using gsivfflat (repr l2) with (ivf_nlist=10);
-- 显示指定DN上索引状态
gaussdb=# EXECUTE DIRECT ON (datanode1) 'SELECT * FROM gs_ivfflat_inspect(''test1v'');';
```

场景2：

入参1的类型：text

入参2的类型：text

出参类型：record

版权所有 © 华为云计算技术有限公司

代码示例：

```
-- 创建分区表
gaussdb=# CREATE TABLE t1 (id int unique,repr floatvector(960)) partition by hash(id) (partition p1,
partition p2, partition p3);
-- 插入数据
gaussdb=# CREATE OR REPLACE FUNCTION float_random_array(dims int,range int) RETURNS float[] AS $
$BEGIN RETURN ARRAY(SELECT (random() * range)::int FROM generate_series(1, dims));END;$$
LANGUAGE plpgsql;
gaussdb=# INSERT INTO t1 SELECT i,floatvector(float_random_array(960,100)) FROM
generate_series(1,1000) as i;
-- 创建分区向量索引
gaussdb=# SET maintenance_work_mem='128MB';
gaussdb=# CREATE INDEX test1v on t1 using gsivfflat (repr l2) LOCAL with (ivf_nlist=10);
-- 查询分区名和分区索引名
gaussdb=# SELECT oid,relname,parttype,parentid,boundaries FROM pg_partition WHERE parentid =
't1'::regclass::oid;
gaussdb=# SELECT oid,relname,parttype,parentid,boundaries,indextblid FROM pg_partition WHERE parentid
= 'test1v'::regclass::oid;
-- 显示指定DN上指定分区的索引状态
gaussdb=# EXECUTE DIRECT ON (datanode1) 'SELECT * FROM gs_ivfflat_inspect(''test1v'', ''p1_repr_idx'');';
```

针对向量计算的操作符，主要集中在衡量向量间相似度，以及向量的二元计算符：

<->

功能说明：计算两向量（floatvector）之间欧氏距离（L2）。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：double precision

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') <->  floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'<-> floatvector('[1,1,3,2]');
```

<+>

功能说明：计算两向量（floatvector）之间余弦距离（COSINE）。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：double precision

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') <+>  floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'<+> floatvector('[1,1,3,2]');
```

<#>

功能说明：计算两向量（boolvector）之间的汉明距离（Hamming distance）。

左参数类型：boolvector

右参数类型：boolvector

返回值类型：double precision

代码示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT boolvector('[1,0,1,0]') <#>  boolvector('[1,1,1,0]');
gaussdb=# SELECT '[1,0,1,0]'<#> boolvector('[1,1,1,0]');
```

+

功能说明：计算两个维度相同的向量按位相加。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：floatvector

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') + floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'+ floatvector('[1,1,3,2]');
```

-

功能说明：计算两个维度相同的向量按位相减。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：floatvector

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') - floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'- floatvector('[1,1,3,2]');
```

<

功能说明：比较两个维度相同的向量字典序的大小关系。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') < floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'< floatvector('[1,1,3,2]');
```

<=

功能说明：比较两个维度相同的向量字典序的大小关系。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') <= floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'<= floatvector('[1,1,3,2]');
```


版权所有 © 华为云计算技术有限公司

>

功能说明：比较两个维度相同的向量字典序的大小关系。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') > floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'> floatvector('[1,1,3,2]');
```

>=

功能说明：比较两个维度相同的向量字典序的大小关系。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') >= floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'>= floatvector('[1,1,3,2]');
```

=

场景1：

功能说明：判断两个维度相同的向量是否相等。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') = floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'= floatvector('[1,1,3,2]');
```

场景2：

功能说明：判断两个维度相同的布尔向量是否一致。

左参数类型：boolvector

右参数类型：boolvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT boolvector('[1,0,1,0]') =  boolvector('[1,1,1,0]');
gaussdb=# SELECT '[1,0,1,0]' =  boolvector('[1,1,1,0]');
```


版权所有 © 华为云计算技术有限公司

<>

功能说明：判断两个维度相同的向量是否不相等。

左参数类型：floatvector

右参数类型：floatvector

返回值类型：BOOLEAN

代码示例：

```
gaussdb=# SELECT floatvector('[1,1,3,2]') <> floatvector('[1,1,3,2]');
gaussdb=# SELECT '[1,2,3,2]'<> floatvector('[1,1,3,2]');
```

说明

若向量元素值过大，在计算距离时，会出现中间计算结果溢出单精度范围的情况，导致距离结果为NAN或INF。

1.6.62.4 相似文档排序召回检索函数和操作符

###

场景1：

功能说明：基于BM25算法族计算两个文本间的相似度，只对使用BM25索引的查询有效。

左参数类型：text

右参数类型：text

返回值类型：double precision

说明：该函数当前版本不可用。

场景2：

功能说明：基于BM25算法族计算两个分词文本数组间的相似度，只对使用BM25索引的查询有效。

左参数类型：text[]

右参数类型：text[]

返回值类型：double precision

说明：该函数当前版本不可用。

gs_ts_add_definition

功能说明：增量添加词典内容。

入参类型：regdictionary, "char", text[]

出参类型：BOOLEAN

说明：该函数当前版本不可用。

版权所有 © 华为云计算技术有限公司

gs_bm25_tokenize

功能说明：使用指定词典对输入文本进行分词操作，返回被分割的单词文本数组。

入参类型：text, cstring(默认值"pg_catalog.simple_cn_segmentation")

出参类型：text[]

说明：该函数当前版本不可用。

gs_bm25_score

功能说明：返回bm25文档相似分数，只在使用BM25索引检索时有效。

入参类型：ctid

出参类型：double precision

说明：该函数当前版本不可用。

gs_bm25_distance_text

功能说明：返回bm25文档相似分数，只在使用BM25索引检索时有效。

入参类型：text, text

出参类型：double precision

说明：该函数当前版本不可用。

gs_bm25_distance_textarr

功能说明：返回bm25文档相似分数，只在使用BM25索引检索时有效。

入参类型：text[], text[]

出参类型：double precision

说明：该函数当前版本不可用。

gs_bm25_docid_info

功能说明：返回指定BM25索引中，指定ctid的docid信息。

入参类型：oid, uint4, uint2

出参类型：record

说明：该函数当前版本不可用。

gs_bm25_document_info

功能说明：返回指定BM25索引中，指定docid的正排信息。

入参类型：oid, uint4

出参类型：record

说明：该函数当前版本不可用。

版权所有 © 华为云计算技术有限公司

gs_bm25_tokenhash_info

功能说明：返回指定BM25索引的哈希容器信息。

入参类型：oid

出参类型：record

说明：该函数当前版本不可用。

gs_bm25_token_info

功能说明：返回指定BM25索引中，指定文档在索引中的token信息。

入参类型：oid, text

出参类型：record

说明：该函数当前版本不可用。

gs_bm25_posting_info

功能说明：返回指定BM25索引中，指定tokenid的倒排信息。

入参类型：oid, text

出参类型：record

说明：该函数当前版本不可用。

1.7 表达式

1.7.1 简单表达式

逻辑表达式

逻辑表达式的操作符和运算规则，请参见**逻辑操作符**。

比较表达式

常用的比较操作符，请参见**比较操作符**。

除比较操作符外，还可以使用以下句式结构：

- BETWEEN操作符：
  a BETWEEN x AND y等效于a >= x AND a <= y
  a NOT BETWEEN x AND y等效于a < x OR a > y
- 检查一个值是不是null，可使用：
  expression IS NULL
  expression IS NOT NULL
  或者与之等价的句式结构，但不是标准的：
  expression ISNULL
  expression NOTNULL

版权所有 © 华为云计算技术有限公司

须知

- 不要写expression=NULL或expression<>(!=)NULL，因为NULL代表一个未知的值，不能通过该表达式判断两个未知值是否相等。
- XML类型数据仅支持比较表达式IS NULL、IS NOT NULL。

- is distinct from/is not distinct from
  - is distinct from

    A和B的数据类型、值不完全相同时为true。

    A和B的数据类型、值完全相同时为false。

    将空值视为相同。
  - is not distinct from

    A和B的数据类型、值不完全相同时为false。

    A和B的数据类型、值完全相同时为true。

    将空值视为相同。
- <=> 安全等于操作符

  在 '=' 比较的基础上增加NULL值的比较，在操作符左右值都不为NULL时与 '=' 结果相同。

  A和B的数据类型、值不完全相同时为false。

  A和B的数据类型、值完全相同时为true。

  将空值视为相同。

  说明

  - <=>操作符与 is not distinct from 用法完全相同。
  - 该操作符仅在数据库兼容MYSQL类型时（即sql_compatibility = 'MYSQL'）有效，其他类型不支持该操作符。

示例

```
gaussdb=# SELECT 2 BETWEEN 1 AND 3 AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 2 >= 1 AND 2 <= 3 AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 2 NOT BETWEEN 1 AND 3 AS RESULT;
 result
----------
 f
(1 row)

gaussdb=# SELECT 2 < 1 OR 2 > 3 AS RESULT;
 result
----------
 f
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT 2+2 IS NULL AS RESULT;
 result
----------
 f
(1 row)

gaussdb=# SELECT 2+2 IS NOT NULL AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 2+2 ISNULL AS RESULT;
 result
----------
 f
(1 row)

gaussdb=# SELECT 2+2 NOTNULL AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 2+2 IS DISTINCT FROM NULL AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 2+2 IS NOT DISTINCT FROM NULL AS RESULT;
 result
----------
 f
(1 row)

gaussdb=# SELECT 1 <=> 1 AS RESULT;
 result
--------
 t
(1 row)

gaussdb=# SELECT NULL <=> 1 AS RESULT;
 result
--------
 f
(1 row)

gaussdb=# SELECT NULL <=> NULL AS RESULT;
 result
--------
 t
(1 row)
```

1.7.2 条件表达式

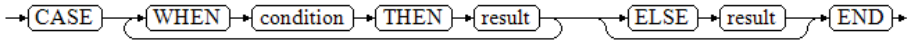
在执行SQL语句时，可通过条件表达式筛选出符合条件的数据。

条件表达式主要有以下几种：

- CASE
  CASE表达式是条件表达式，类似于其他编程语言中的CASE语句。
  CASE表达式的语法图如**图1-3**所示。

 版权所有 © 华为云计算技术有限公司

图 1-3 case::=



CASE子句可以用于合法的表达式中。condition是一个返回BOOLEAN数据类型的表达式：

- 如果结果为真，CASE表达式的结果就是符合该条件所对应的result。
- 如果结果为假，则以相同方式处理随后的WHEN或ELSE子句。
- 如果各WHEN condition都不为真，表达式的结果就是在ELSE子句执行的result。如果省略了ELSE子句且没有匹配的条件，结果为NULL。
- 支持对XML类型数据操作。
- GUC参数enable_case_intervaltonumeric设置为true时，支持将INTERVAL转换为NUMERIC而不报错。

示例：

```
gaussdb=# CREATE TABLE case_when_t1(CW_COL1 INT) DISTRIBUTE BY HASH (CW_COL1);

gaussdb=# INSERT INTO case_when_t1 VALUES (1), (2), (3);

gaussdb=# SELECT * FROM case_when_t1;
 cw_col1
---
 1
 2
 3
(3 rows)

gaussdb=# SELECT CW_COL1, CASE WHEN CW_COL1=1 THEN 'one' WHEN CW_COL1=2 THEN 'two'
ELSE 'other' END FROM case_when_t1 ORDER BY 1;
 cw_col1 | case
---------+-------
       1 | one
       2 | two
       3 | other
(3 rows)

gaussdb=# DROP TABLE case_when_t1;

-- GUC参数默认为false，此时报错。
gaussdb=# SHOW enable_case_intervaltonumeric;
 enable_case_intervaltonumeric
-------------------------------
 off
(1 row)

gaussdb=# SELECT CASE
gaussdb-#     WHEN 1=1 THEN
gaussdb-#         to_date('20240118','yyyymmdd')-to_date('20240116','yyyymmdd')
gaussdb-#     ELSE
gaussdb-#         1
gaussdb-# end;
ERROR:  CASE types integer and interval cannot be matched
LINE 3:         to_date('20240118','yyyymmdd')-to_date('20240116','y...
                ^
CONTEXT:  referenced column: case

-- 打开GUC参数，能正确返回结果。
gaussdb=# SET enable_case_intervaltonumeric=true;
SET
gaussdb=# SELECT CASE
gaussdb-#     WHEN 1=1 THEN
gaussdb-#         to_date('20240118','yyyymmdd')-to_date('20240116','yyyymmdd')
gaussdb-#     ELSE
```

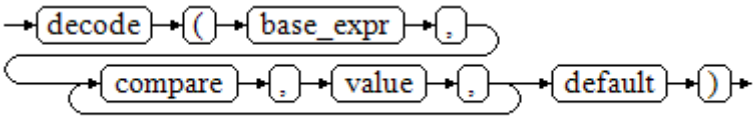

版权所有 © 华为云计算技术有限公司

```
gaussdb-#         1::int1
gaussdb-# END;
 case
------
    2
(1 row)
```

- DECODE

  DECODE的语法图如**图1-4**所示。

  图 1-4 decode::=

  将表达式base_expr与后面的每个compare(n) 进行比较，如果匹配返回相应的value(n)。如果没有发生匹配，则返回default。

  支持对XML类型数据操作。

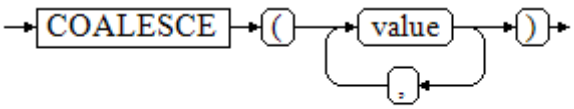
  示例请参见**条件表达式函数**。

```
gaussdb=# SELECT DECODE('A','A',1,'B',2,0);
 case
------
    1
(1 row)
```

- COALESCE

  COALESCE的语法图如**图1-5**所示。

  图 1-5 coalesce::=

  COALESCE返回它的第一个非NULL的参数值。如果参数都为NULL，则返回NULL。它常用于在显示数据时用缺省值替换NULL。和CASE表达式一样，COALESCE只计算用来判断结果的参数，即在第一个非空参数右边的参数不会被计算。

  支持对XML类型数据操作。

  示例：

```
gaussdb=# CREATE TABLE c_tabl(description varchar(10), short_description varchar(10), last_value
varchar(10))
DISTRIBUTE BY HASH (last_value);

gaussdb=# INSERT INTO c_tabl VALUES('abc', 'efg', '123');
gaussdb=# INSERT INTO c_tabl VALUES(NULL, 'efg', '123');

gaussdb=# INSERT INTO c_tabl VALUES(NULL, NULL, '123');

gaussdb=# SELECT description, short_description, last_value, COALESCE(description, short_description,
last_value) FROM c_tabl ORDER BY 1, 2, 3, 4;
 description | short_description | last_value | coalesce
-------------+-------------------+------------+----------
```


版权所有 © 华为云计算技术有限公司

```
 abc        | efg             | 123       | abc
            | efg             | 123       | efg
            |                 | 123       | 123
(3 rows)

gaussdb=# DROP TABLE c_tabl;
```

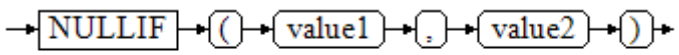
如果description不为NULL，则返回description的值，否则计算下一个参数short_description；如果short_description不为NULL，则返回short_description的值，否则计算下一个参数last_value；如果last_value不为NULL，则返回last_value的值，否则返回（none）。

```
gaussdb=# SELECT COALESCE(NULL,'Hello World');
   coalesce
---------------
 Hello World
(1 row)
```

- NULLIF

  NULLIF的语法图如**图1-6**所示。

  图 1-6 nullif::=

  只有当value1和value2相等时，NULLIF才返回NULL。否则它返回value1。支持对XML类型数据操作。

  示例：

```
gaussdb=# CREATE TABLE null_if_t1 (
    NI_VALUE1 VARCHAR(10),
    NI_VALUE2 VARCHAR(10)
)DISTRIBUTE BY HASH (NI_VALUE1);

gaussdb=# INSERT INTO null_if_t1 VALUES('abc', 'abc');
gaussdb=# INSERT INTO null_if_t1 VALUES('abc', 'efg');

gaussdb=# SELECT NI_VALUE1, NI_VALUE2, NULLIF(NI_VALUE1, NI_VALUE2) FROM null_if_t1 ORDER
BY 1, 2, 3;

 ni_value1 | ni_value2 | nullif
-----------+-----------+--------
 abc       | abc       |
 abc       | efg       | abc
(2 rows)
gaussdb=# DROP TABLE null_if_t1;
```

  如果value1等于value2则返回NULL，否则返回value1。

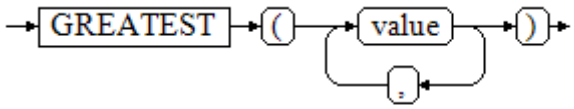
```
gaussdb=# SELECT NULLIF('Hello','Hello World');
 nullif
--------
 Hello
(1 row)
```

- GREATEST（最大值），LEAST（最小值）

  GREATEST的语法图如**图1-7**所示。

版权所有 © 华为云计算技术有限公司

图 1-7 greatest::=

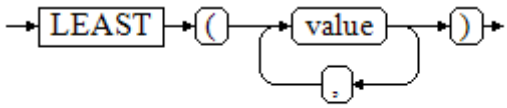


从一个任意数字表达式的列表里选取最大的数值。支持对XML类型数据操作。

```
gaussdb=# SELECT greatest(9000,155555,2.01);
 greatest
----------
   155555
(1 row)
```

LEAST的语法图如**图1-8**所示。

图 1-8 least::=



从一个任意数字表达式的列表里选取最小的数值。

以上的数字表达式必须都可以转换成一个普通的数据类型，该数据类型将是结果类型。

列表中的NULL值将被忽略。只有所有表达式的结果都是NULL的时候，结果才是NULL。

支持对XML类型数据操作。

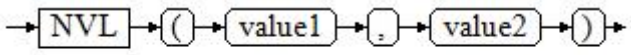
```
gaussdb=# SELECT least(9000,2);
 least
-------
     2
(1 row)
```

示例请参见**条件表达式函数**。

- NVL

  NVL的语法图如**图1-9**所示。

  图 1-9 nvl::=

  如果value1为NULL，则返回value2，如果value1非NULL，则返回value1。支持对XML类型数据操作。

  示例：

  ```
  gaussdb=# SELECT nvl(null,1);
  nvl
  -----
   1
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT nvl('Hello World',1);
     nvl
---------------
 Hello World
(1 row)
```

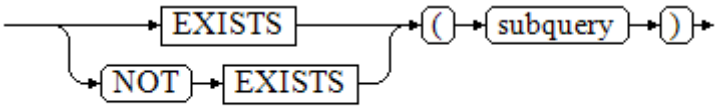
1.7.3 子查询表达式

子查询表达式主要有以下几种：

- EXISTS/NOT EXISTS

  EXISTS/NOT EXISTS的语法图如图1-10所示。

  图 1-10 EXISTS/NOT EXISTS::=

  EXISTS的参数是一个任意的SELECT语句，即子查询。系统对子查询进行运算以判断它是否返回行。如果它至少返回一行，则EXISTS结果就为“真”；如果子查询没有返回任何行， EXISTS的结果是“假”。

  这个子查询通常只是运行到能判断它是否可以生成至少一行为止，而不是等到全部结束。

  不支持对XML类型数据操作。

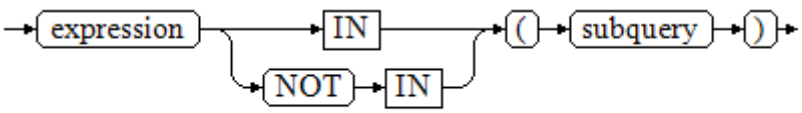
  示例：

  ```
  gaussdb=# CREATE TABLE exists_t1(a int, b int);
  gaussdb=# INSERT INTO exists_t1 VALUES(1, 2),(2, 3),(3, 4),(4, 5);

  gaussdb=# CREATE TABLE exists_t2(a int, c int);
  gaussdb=# INSERT INTO exists_t2 VALUES(3, 4),(4, 5),(5, 6),(6, 7);

  gaussdb=# SELECT * FROM exists_t1 t1 WHERE EXISTS (SELECT * FROM exists_t2 t2 WHERE t2.a =
  t1.a);
   a | b
  ---+---
   3 | 4
   4 | 5
  (2 rows)

  gaussdb=# DROP TABLE exists_t1, exists_t2;
  ```

- IN/NOT IN

  IN/NOT IN的语法如图1-11所示。

  图 1-11 IN/NOT IN::=

  右边是一个圆括号括起来的子查询，它必须只返回一个字段。左边表达式对子查询结果的每一行进行一次计算和比较。如果找到任何相等的子查询行，则IN结果

版权所有 © 华为云计算技术有限公司

为“真”。如果没有找到任何相等行，则结果为“假”（包括子查询没有返回任何行的情况）。

表达式或子查询行里的NULL遵照SQL处理布尔值和NULL组合时的规则。如果两个行对应的字段都相等且非空，则这两行相等；如果任意对应字段不等且非空，则这两行不等；否则结果是未知（NULL）。如果每一行的结果都是不等或NULL，并且至少有一个NULL，则IN的结果是NULL。

不支持对XML类型数据操作。

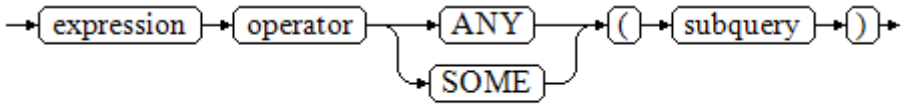
示例：

```
gaussdb=# CREATE TABLE in_t1(a int, b int);
gaussdb=# INSERT INTO in_t1 VALUES(1, 2),(2, 3),(3, 4),(4, 5);

gaussdb=# CREATE TABLE in_t2(a int, c int);
gaussdb=# INSERT INTO in_t2 VALUES(3, 4),(4, 5),(5, 6),(6, 7);

gaussdb=# SELECT * FROM in_t1 t1 WHERE t1.a IN (SELECT t2.a FROM in_t2 t2);
 a | b
---+---
 3 | 4
 4 | 5
(2 rows)

gaussdb=# DROP TABLE in_t1, in_t2;
```

- ANY/SOME

  ANY/SOME的语法图如图1-12所示。

  图 1-12 any/some::=

  右边是一个圆括号括起来的子查询，它必须只返回一个字段。左边表达式使用operator对子查询结果的每一行进行一次计算和比较，其结果必须是布尔值。如果至少获得一个真值，则ANY结果为“真”。如果全部获得假值，则结果是“假”（包括子查询没有返回任何行的情况）。SOME是ANY的同义词。IN与ANY可以等效替换。

  不支持对XML类型数据操作。

  示例：

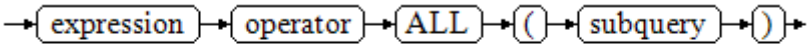
```
gaussdb=# CREATE TABLE any_t1(a int, b int);
gaussdb=# INSERT INTO any_t1 VALUES(1, 2),(2, 3),(3, 4),(4, 5);

gaussdb=# CREATE TABLE any_t2(a int, c int);
gaussdb=# INSERT INTO any_t2 VALUES(3, 4),(4, 5),(5, 6),(6, 7);

gaussdb=# SELECT * FROM any_t1 t1 WHERE t1.a < ANY(SELECT t2.a FROM any_t2 t2 where t2.a = 3
or t2.a = 4);
 a | b
---+---
 1 | 2
 2 | 3
 3 | 4
(3 rows)

gaussdb=# DROP TABLE any_t1, any_t2;
```

- ALL

版权所有 © 华为云计算技术有限公司

ALL的语法如图1-13所示。

图 1-13 all::=



右边是一个圆括号括起来的子查询，它必须只返回一个字段。左边表达式使用operator对子查询结果的每一行进行一次计算和比较，其结果必须是布尔值。如果全部获得真值，ALL结果为“真”（包括子查询没有返回任何行的情况）。如果至少获得一个假值，则结果是“假”。

不支持对XML类型数据操作。

示例：

```
gaussdb=# CREATE TABLE all_t1(a int, b int);
gaussdb=# INSERT INTO all_t1 VALUES(1, 2),(2, 3),(3, 4),(4, 5);

gaussdb=# CREATE TABLE all_t2(a int, c int);
gaussdb=# INSERT INTO all_t2 VALUES(3, 4),(4, 5),(5, 6),(6, 7);

gaussdb=# SELECT * FROM all_t1 t1 WHERE t1.a < ALL(SELECT t2.a FROM all_t2 t2 where t2.a = 3 or t2.a = 4);
 a | b
---+---
 1 | 2
 2 | 3
(2 rows)

gaussdb=# DROP TABLE all_t1, all_t2;
```

1.7.4 数组表达式

IN

expression **IN** *(value [, ...])*

右侧括号中的是一个表达式列表。左侧表达式的结果与表达式列表的内容进行比较。如果列表中的内容符合左侧表达式的结果，则IN的结果为true。如果没有相符的结果，则IN的结果为false。

示例如下：

```
gaussdb=# SELECT 8000+500 IN (10000, 9000) AS RESULT;
 result
----------
 f
(1 row)
```

说明

- 如果表达式结果为null，或者表达式列表不符合表达式的条件且右侧表达式列表返回结果至少一处为空，则IN的返回结果为null，而不是false。这样的处理方式和SQL返回空值的布尔组合规则是一致的。
- 不支持对XML类型数据操作。

NOT IN

expression **NOT IN** *(value [, ...])*

版权所有 © 华为云计算技术有限公司

右侧括号中的是一个表达式列表。左侧表达式的结果与表达式列表的内容进行比较。如果在列表中的内容没有符合左侧表达式结果的内容，则NOT IN的结果为true。如果有符合的内容，则NOT IN的结果为false。

示例如下：

```
gaussdb=# SELECT 8000+500 NOT IN (10000, 9000) AS RESULT;
 result
----------
 t
(1 row)
```

说明

- 如果查询语句返回结果为空，或者表达式列表不符合表达式的条件且右侧表达式列表返回结果至少一处为空，则NOT IN的返回结果为null，而不是false。这样的处理方式和SQL返回空值的布尔组合规则是一致的。
- 在所有情况下X NOT IN Y等价于NOT(X IN Y)。
- 不支持对XML类型数据操作。

ANY/SOME (array)

expression operator **ANY** *(array expression)*

expression operator **SOME** *(array expression)*

右侧括号中的是一个数组表达式，它必须产生一个数组值。左侧表达式的结果使用操作符对数组表达式的每一行结果都进行计算和比较，比较结果必须是布尔值。

示例如下：

```
gaussdb=# SELECT 8000+500 < SOME (array[10000,9000]) AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 8000+500 < ANY (array[10000,9000]) AS RESULT;
 result
----------
 t
(1 row)
```

说明

- 如果对比结果至少获取一个真值，则ANY的结果为true。
- 如果对比结果没有真值，则ANY的结果为false。
- 如果结果没有真值，并且数组表达式生成至少一个值为null，则ANY的值为NULL，而不是false。这样的处理方式和SQL返回空值的布尔组合规则是一致的。
- SOME是ANY的同义词。
- 不支持对XML类型数据操作。

ALL (array)

expression operator **ALL** *(array expression)*

右侧括号中的是一个数组表达式，它必须产生一个数组值。左侧表达式的结果使用操作符对数组表达式的每一行结果都进行计算和比较，比较结果必须是布尔值。

- 如果所有的比较结果都为真值（包括数组不含任何元素的情况），则ALL的结果为true。

版权所有 © 华为云计算技术有限公司

- 如果存在一个或多个比较结果为假值，则ALL的结果为false。
- 如果数组表达式产生一个NULL数组，则ALL的结果为NULL。如果左边表达式的值为NULL，则ALL的结果通常也为NULL(不严格的比较操作符可能得到不同的结果)。如果右边的数组表达式中包含null元素并且比较结果没有假值，则ALL的结果将是NULL(不严格的比较操作符可能得到不同的结果)，而不是“真”。这样的处理方式和SQL返回空值的布尔组合规则是一致的。
- 不支持对XML类型数据操作。

示例如下：

```
gaussdb=# SELECT 8000+500 < ALL (array[10000,9000]) AS RESULT;
 result
----------
 t
(1 row)
```

1.7.5 行表达式

语法如下：

```
row_constructor operator row_constructor
```

两边都是一个行构造器，两行值必须具有相同数目的字段，每一行都进行比较，行比较允许使用=，<>，<，<=，>=等操作符，或其中一个相似的语义符。

对于<，<=，>，>=的情况下，行中元素从左到右依次比较，直到遇到一对不相等的元素或者一对为空的元素。如果这对元素中存在至少一个null值，则比较结果是未知的（null），否则这对元素的比较结果为最终的结果。如果最终没有遇到不相等或者为空的元素，则认为这两行值相等，根据操作符含义判断最终结果。

不支持对XML类型数据操作。

示例：

```
gaussdb=# SELECT ROW(1,2,NULL) < ROW(1,3,0) AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT (4,5,6) > (3,2,1) AS result;
 result
--------
 t
(1 row)

gaussdb=# SELECT (4,1,1) > (3,2,1) AS result;
 result
--------
 t
(1 row)

gaussdb=# SELECT ('test','data') > ('data','data') AS result;
 result
--------
 t
(1 row)

gaussdb=# SELECT (4,1,1) > (3,2,null) AS result;
 result
--------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT (null,1,1) > (3,2,1) AS result;
 result
--------

(1 row)

gaussdb=# SELECT (null,5,6) > (null,5,6) AS result;
 result
--------

(1 row)

gaussdb=# SELECT (4,5,6) > (4,5,6) AS result;
 result
--------
 f
(1 row)

gaussdb=# SELECT (2,2,5) >= (2,2,3) AS result;
 result
--------
 t
(1 row)

gaussdb=# SELECT (2,2,1) <= (2,2,3) AS result;
 result
--------
 t
(1 row)
```

=，<>和别的操作符使用略有不同。如果两行值的所有字段都是非空并且符合操作符条件，则认为两行是符合操作符条件的；如果两行值的任意字段为非空并且不符合操作符条件，则认为两行是不符合操作符条件的；如果两行值的任意字段为空，则比较的结果是未知的（null）。

示例：

```
gaussdb=# SELECT (1,2,3) = (1,2,3) AS result;
 result
--------
 t
(1 row)

gaussdb=# SELECT (1,2,3) <> (2,2,3) AS result;
 result
--------
 t
(1 row)

gaussdb=# SELECT (2,2,3) <> (2,2,null) AS result;
 result
--------

(1 row)

gaussdb=# SELECT (null,5,6) <> (null,5,6) AS result;
 result
--------

(1 row)
```

1.7.6 时间间隔表达式

语法：INTERVAL EXPR UNIT

说明：EXPR表示数值，UNIT说明符用于解释数值的单位，如HOUR、DAY、WEEK等。关键字INTERVAL和说明符不区分大小写。

版权所有 © 华为云计算技术有限公司

时间间隔表达式中UNIT的取值范围如下表1-331所示，允许任何标点符号分隔EXPR格式。表1-331中显示的是建议的分隔符。

说明

INTERVAL表达式仅在sql_compatibility = 'MYSQL'，且参数b_format_version值为5.7、b_format_dev_version值为s1时，才支持上述功能。

表 1-331 时间间隔表达式 UNIT 取值范围

UNIT取值范围	预期EXPR格式
MICROSECOND	MICROSECONDS
SECOND	SECONDS
MINUTE	MINUTES
HOUR	HOURS
DAY	DAYS
WEEK	WEEKS
MONTH	MONTHS
QUARTER	QUARTERS
YEAR	YEARS
SECOND_MICROSECOND	'SECOND_MICROSECOND'
MINUTE_MICROSECOND	'MINUTES:SECONDS.MICROSECONDS'
MINUTE_SECOND	'MINUTES:SECONDS'
HOUR_MICROSECOND	'HOURS:MINUTES:SECONDS.MICROSECONDS'
HOUR_SECOND	'HOURS:MINUTES:SECONDS'
HOUR_MINUTE	'HOURS:MINUTES'
DAY_MICROSECOND	'DAYS HOURS:MINUTES:SECONDS.MICROSECONDS'
DAY_SECOND	'DAYS HOURS:MINUTES:SECONDS'
DAY_MINUTE	'DAYS HOURS:MINUTES'
DAY_HOUR	'DAYS HOURS'
YEAR_MONTH	'YEAR_MONTH'

示例：

```
gaussdb=# SELECT DATE_ADD('2018-05-01', INTERVAL 1 DAY);
  date_add
```


版权所有 © 华为云计算技术有限公司

```
------------
 2018-05-02
(1 row)

gaussdb=# SELECT DATE_SUB('2018-05-01', INTERVAL 1 YEAR);
  date_sub
------------
 2017-05-01
(1 row)

gaussdb=# SELECT DATE'2023-01-10' - INTERVAL 1 DAY;
      ?column?
---------------------
 2023-01-09 00:00:00
(1 row)

gaussdb=# SELECT DATE'2023-01-10' + INTERVAL 1 MONTH;
      ?column?
---------------------
 2023-02-10 00:00:00
(1 row)
```

1.8 伪列

ROWNUM是一个伪列，它返回一个数字，表示从查询中获取结果的行编号。第一行的ROWNUM为1，第二行的为2，后续行号按顺序递增，使用ROWNUM来限制查询返回的行数。

ROWNUM机制介绍：

- 赋值时机：ROWNUM是在查询结果集产生过程中动态分配的，是在查询获取到结果集之后再加上去的一个伪列。这意味着序号是在数据被WHERE子句过滤后，但在任何ORDER BY排序之前就确定了。
- 比较操作符使用：ROWNUM<=n（n大于1）是有效的，常用语限制返回的行数，例如获取前N条记录。如以下示例所示：

  ```
  gaussdb=# CREATE TABLE Students (name varchar(20), id int) with (STORAGE_TYPE = USTORE);
  gaussdb=# INSERT INTO Students VALUES ('Jack', 35);
  gaussdb=# INSERT INTO Students VALUES ('Leon', 15);
  gaussdb=# INSERT INTO Students VALUES ('James', 24);
  gaussdb=# INSERT INTO Students VALUES ('Taker', 81);
  gaussdb=# INSERT INTO Students VALUES ('Mary', 25);
  gaussdb=# INSERT INTO Students VALUES ('Rose', 64);
  gaussdb=# INSERT INTO Students VALUES ('Perl', 18);
  gaussdb=# INSERT INTO Students VALUES ('Under', 57);
  gaussdb=# INSERT INTO Students VALUES ('Angel', 101);
  gaussdb=# INSERT INTO Students VALUES ('Frank', 20);
  gaussdb=# INSERT INTO Students VALUES ('Charlie', 40);

  -- 输出表Students前10行数据。
  gaussdb=# SELECT * FROM Students WHERE rownum <= 10;
   name  | id
  -------+-----
   Jack  |  35
   Leon  |  15
   James |  24
   Taker |  81
   Mary  |  25
   Rose  |  64
   Perl  |  18
   Under |  57
   Angel | 101
   Frank |  20
  (10 rows)
  ```

- ROWNUM = 1是有效的，可以查询出第一条记录。

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM Students WHERE rownum = 1;
 name | id
------+----
 Jack | 35
(1 row)
```

ROWNUM = n（当n > 1）、ROWNUM > n（n >= 1）或ROWNUM >= n（n > 1）这些条件结果永远为FALSE，不会返回任何行。这是因为第一条取到的行分配ROWNUM=1，如果它不满足条件（例如ROWNUM > 1），则会被丢弃；接着处理下一条记录，它又成为了新的“第一条”记录，ROWNUM再次被赋值为1，如此循环，所有行都无法满足条件。

```
gaussdb=# SELECT * FROM Students WHERE rownum > 1;
 name | id
------+----
(0 rows)
```

ROWNUM的基本使用：

- 限制返回行数。

```
gaussdb=# SELECT * FROM Students WHERE rownum <= 2;
 name | id
------+----
 Jack | 35
 Leon | 15
(2 rows)
```

- 实现分页查询：由于不能直接使用ROWNUM > n，分页查询需要通过子查询为ROWNUM创建一个别名（例如rn），使其成为一个“固定”的列，然后在外层查询中基于这个别名进行范围筛选。

```
gaussdb=# SELECT *
FROM (SELECT a.*, ROWNUM AS rn
      FROM
      (SELECT * FROM Students ORDER BY id DESC) a -- 内层子查询先排序
      WHERE ROWNUM <= 10) -- 中层子查询限定上限并生成ROWNUM别名
WHERE rn >= 5;
  name   | id | rn
---------+----+----
 Charlie | 40 |  5
 Jack    | 35 |  6
 Mary    | 25 |  7
 James   | 24 |  8
 Frank   | 20 |  9
 Perl    | 18 | 10
(6 rows)
```

- 排序后的前N条数据：如果直接在同级查询中同时使用WHERE ROWNUM <= N和ORDER BY，可能会得到不正确的结果，应为ROWNUM在排序前就已分配。正确方法是先使用子查询进行排序，后分配ROWNUM。

```
gaussdb=# SELECT rownum, * FROM (SELECT * FROM Students ORDER BY 1) WHERE rownum <= 2;
 rownum |  name   | id
--------+---------+-----
      1 | Angel   | 101
      2 | Charlie |  40
(2 rows)
```

- 使用ROWNUM指定给表的一定范围的每一行分配值。

```
gaussdb=# SELECT * FROM Students;
  name   | id
---------+-----
 Jack    |  35
 Leon    |  15
 James   |  24
 Taker   |  81
 Mary    |  25
 Rose    |  64
 Perl    |  18
 Under   |  57
 Angel   | 101
```


版权所有 © 华为云计算技术有限公司

```
 Frank   |  20
 Charlie |  40
(11 rows)

gaussdb=# UPDATE Students SET id = id + 5 WHERE rownum < 4;
UPDATE 3
gaussdb=# SELECT * FROM Students;
  name   | id
---------+-----
 Jack    |  40
 Leon    |  20
 James   |  29
 Taker   |  81
 Mary    |  25
 Rose    |  64
 Perl    |  18
 Under   |  57
 Angel   | 101
 Frank   |  20
 Charlie |  40
(11 rows)

gaussdb=# DROP TABLE Students;
DROP TABLE
```

使用ROWNUM有一定的约束条件：

- ROWNUM不可作为别名，以免SQL语句出现歧义。
- 创建索引时不可使用ROWNUM。
- 创建表时默认值不可为ROWNUM。
- Where子句中不可使用ROWNUM的别名。
- 在插入数据时不可使用ROWNUM。
- 在无表查询中不可以使用ROWNUM。
- ROWNUM不能用于Limit子句。
- ROWNUM不能用于EXECUTE语句的参数。
- UPSERT语句不支持ROWNUM用做update子句更新。
- SELECT ... FOR UPDATE语句不支持ROWNUM用作投影列和WHERE条件。
- 若HAVING子句中含有ROWNUM（且不在聚合函数中）时，GROUP BY子句中必须含有ROWNUM（且不在聚合函数中），除非GROUP BY子句存在表达式，例如：SELECT a + a FROM t GROUP BY a + a HAVING rownum < 5。
- HAVING子句中如果存在ROWNUM条件则不允许HAVING子句下推至扫描节点：

```
gaussdb=# CREATE TABLE test (a int, b int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# INSERT INTO test SELECT generate_series, generate_series FROM generate_series(1, 10);
INSERT 0 10

--rownum条件不能下推至seqscan。
gaussdb=# EXPLAIN SELECT a,rownum FROM test group by a,rownum having rownum < 5;
                            QUERY PLAN
------------------------------------------------------------------------------
 HashAggregate  (cost=3.45..3.49 rows=3 width=4)
   Group By Key: a, ROWNUM
   Filter: ((ROWNUM) < 5::numeric)
   ->  Rownum  (cost=0.19..3.40 rows=10 width=4)
         ->  Streaming (type: GATHER)  (cost=0.19..3.40 rows=10 width=4)
               Node/s: All datanodes
               ->  Seq Scan on test  (cost=0.00..3.03 rows=10 width=4)
(7 rows)
```


版权所有 © 华为云计算技术有限公司

- 子查询中如果存在ROWNUM条件则不允许谓词下推至扫描节点：

```
-- b<5 不能下推至seqscan。
gaussdb=# EXPLAIN SELECT * FROM (SELECT * FROM test WHERE rownum < 5) WHERE b < 5;
                                QUERY PLAN
-----------------------------------------------------------------------------------
 Streaming (type: GATHER)  (cost=0.06..0.63 rows=2 width=8)
   Node/s: All datanodes
   ->  Subquery Scan on __unnamed_subquery__  (cost=0.00..0.50 rows=2 width=8)
         Filter: (__unnamed_subquery__.b < 5)
         ->  Rownum  (cost=0.00..0.45 rows=4 width=8)
               StopKey: (ROWNUM < 5::numeric)
               ->  Streaming(type: BROADCAST)  (cost=0.00..1.36 rows=12 width=8)
                     Spawn on: All datanodes
                     ->  Rownum  (cost=0.00..1.35 rows=4 width=8)
                           StopKey: (ROWNUM < 5::numeric)
                           ->  Seq Scan on test  (cost=0.00..3.03 rows=10 width=8)
(11 rows)

gaussdb=# DROP TABLE test;
DROP TABLE
```

说明

- ROWNUM查询结果将会由于CN接收DN数据的顺序不同导致查询结果不稳定。

```
gaussdb=# CREATE TABLE test(a int, b int);
gaussdb=# INSERT INTO test VALUES(generate_series(1,10),generate_series(1,10));

--CN先收到DN1数据再收到DN2数据。
gaussdb=# SELECT rownum,* FROM test;
 rownum | a  | b
--------+----+----
      1 |  1 |  1
      2 |  2 |  2
      3 |  5 |  5
      4 |  6 |  6
      5 |  8 |  8
      6 |  9 |  9
      7 |  3 |  3
      8 |  4 |  4
      9 |  7 |  7
     10 | 10 | 10
(10 rows)

--CN先收到DN2数据再收到DN1数据。
gaussdb=# SELECT rownum,* FROM test;
 rownum | a  | b
--------+----+----
      1 |  3 |  3
      2 |  4 |  4
      3 |  7 |  7
      4 | 10 | 10
      5 |  1 |  1
      6 |  2 |  2
      7 |  5 |  5
      8 |  6 |  6
      9 |  8 |  8
     10 |  9 |  9
(10 rows)
```

- 不推荐ROWNUM条件用于JOIN ON子句，GaussDB中ROWNUM条件用于JOIN ON子句时在LEFT JOIN、RIGHT JOIN、FULL JOIN场景下和MERGE INTO场景下与其他数据库行为不一致，直接进行业务迁移存在风险。

当父查询中有rownum限制条件同时子查询的投影列中有rownum时，该限制条件将下推至子查询。其中约束条件如下：

- 只有父查询rownum限制条件为“<”、“<=”、“=”，且子查询直接用rownum作为伪列时，才可以下推。

 版权所有 © 华为云计算技术有限公司

- 父查询中有多个对子查询中rownum的过滤条件并且满足下推要求时，按顺序仅下推第一个过滤条件。
- 当子查询中包括volatile函数、存储过程时不能下推。

1.9 类型转换

1.9.1 概述

背景信息

在SQL语言中，每个数据都与一个决定其行为和用法的数据类型相关。GaussDB提供一个可扩展的数据类型系统，该系统比其它SQL实现更具通用性和灵活性。因此，GaussDB中大多数类型转换是由通用规则来管理的，这种做法允许使用混合类型的表达式。

GaussDB扫描/分析器只将词法元素分解成五个基本种类：整数、浮点数、字符串、标识符和关键字。大多数非数字类型首先表现为字符串。SQL语言的定义允许将常量字符串声明为具体的类型。例如：

```
gaussdb=# SELECT text 'Origin' AS "label", point '(0,0)' AS "value";
 label  | value
--------+-------
 Origin | (0,0)
(1 row)
```

示例中有两个文本常量，类型分别为text和point。如果没有为字符串文本声明类型，则该文本首先被定义成一个unknown类型。

在GaussDB分析器里，有四种基本的SQL结构需要独立的类型转换规则：

- 函数调用

  多数SQL类型系统是建立在一套丰富的函数上的。函数调用可以有一个或多个参数。因为SQL允许函数重载，所以不能通过函数名直接找到要调用的函数，分析器必须根据函数提供的参数类型选择正确的函数。

- 操作符

  SQL允许在表达式上使用前缀或后缀（单目）操作符，也允许表达式内部使用双目操作符（两个参数）。像函数一样，操作符也可以被重载，因此操作符的选择也和函数一样取决于参数类型。

- 值存储

  INSERT和UPDATE语句将表达式结果存入表中。语句中的表达式类型必须和目标字段的类型一致或者可以转换为一致。

- UNION，CASE和相关构造

  因为联合SELECT语句中的所有查询结果必须在一列里显示出来，所以每个SELECT子句中的元素类型必须相互匹配并转换成一个统一类型。类似地，一个CASE构造的结果表达式必须转换成统一的类型，这样整个CASE表达式会有一个统一的输出类型。同样的要求也存在于ARRAY构造以及GREATEST和LEAST函数中。

系统表pg_cast存储了有关数据类型之间的转换关系以及如何执行这些转换的信息。详细信息请参见**PG_CAST**。

语义分析阶段会决定表达式的返回值类型并选择适当的转换行为。数据类型的基本类型分类，包括：Boolean、numeric、string、bitstring、datetime、timespan、

版权所有 © 华为云计算技术有限公司

geometric和network。每种类型都有一种或多种首选类型用于解决类型选择的问题。根据首选类型和可用的隐含转换，就可能保证有歧义的表达式（那些有多个候选解析方案的）得到有效的方式解决。

所有类型转换规则都是建立在下面几个基本原则上的：

- 隐含转换决不能有奇怪的或不可预见的输出。
- 如果一个查询不需要隐含的类型转换，分析器和执行器不应该进行更多的额外操作。任何一个类型匹配、格式清晰的查询不应该在分析器里耗费更多的时间，也不应该向查询中引入任何不必要的隐含类型转换调用。
- 如果一个查询在调用某个函数时需要进行隐式转换，当用户定义了一个有正确参数的函数后，解释器应该选择使用新函数。
- XML类型数据不支持隐式类型转换，包括字符串和XML类型之间的隐式转换。
- ORA兼容模式下，DATEA类型仅支持与text、bpchar、varchar、clob、nvarchar2、timestamp、timestamp with time zone进行类型转换。

1.9.2 操作符

操作符类型解析

1. 从系统表pg_operator中选出要考虑的操作符。如果可以找到一个参数类型以及参数个数都一致的操作符，那么这个操作符就是最终使用的操作符。如果找到了多个备选的操作符，将从中选择一个最合适的。
2. 寻找最优匹配。
   a. 抛弃那些输入类型不匹配并且也不能隐式转换成匹配的候选操作符。unknown文本在这种情况下可以转换成任何东西。如果只剩下一个候选项，则用之，否则继续下一步。
   b. 遍历所有候选操作符，保留那些输入类型匹配最准确的。此时，域类型被视为与其基本类型相同。如果没有一个操作符能被保留，则保留所有候选。如果只剩下一个候选项，则用之，否则继续下一步。
   c. 遍历所有候选操作符，保留那些需要类型转换时接受（属于输入数据类型的类型范畴的）首选类型位置最多的操作符。如果没有接受首选类型的操作符，则保留所有候选。如果只剩下一个候选项，则用之，否则继续下一步。
   d. 如果有任何输入参数是unknown类型，检查剩余的候选操作符对应参数位置的类型范畴。在每一个能够接受字符串类型范畴的位置使用string类型（这种对字符串的偏爱是合适的，因为unknown文本确实像字符串）。如果所有剩下的候选操作符都接受相同的类型范畴，则选择该类型范畴，否则抛出一个错误（因为在没有更多线索的条件下无法作出正确的选择）。现在抛弃不接受选定的类型范畴的候选操作符，如果任意候选操作符在某个给定的参数位置接受一个首选类型，则抛弃那些在该参数位置接受非首选类型的候选操作符。如果没有一个操作符能被保留，则保留所有候选。如果只剩下一个候选项，则用之，否则继续下一步。
   e. 如果同时有unknown和已知类型的参数，并且所有已知类型的参数都是相同的类型，那么假设unknown参数也是那种类型，并检查哪一个候选操作符在unknown参数位置接受该类型。如果只有一个操作符符合，那么使用它。否则产生一个错误。

版权所有 © 华为云计算技术有限公司

说明

在找到一个操作符后，如果输入的参数类型和操作符的参数类型不一致，可能会发生隐式类型转换，转换后可能发生不可预知的行为。如果隐式转换后行为有问题，可以通过显式类型转换规避此问题。例如，定长类型bpchar转换为变长类型text后，会消除字符串行尾空格，如果再和其它字符串比较时可能会发生错误行为。

示例

示例1：阶乘操作符类型解析。在系统表中里只有一个阶乘操作符（后缀!），它以bigint作为参数。扫描器给下面查询表达式的参数赋予bigint的初始类型：

```
gaussdb=# SELECT 40 ! AS "40 factorial";

                  40 factorial
-----------------------------------------------------
 815915283247897734345611269596115894272000000000
(1 row)
```

分析器对参数做类型转换，查询等效于：

```
gaussdb=# SELECT CAST(40 AS bigint) ! AS "40 factorial";
```

示例2：字符串连接操作符类型分析。一种字符串风格的语法既可以用于字符串也可以用于复杂的扩展类型。未声明类型的字符串将被所有可能的候选操作符匹配。有一个未声明的参数的例子：

```
gaussdb=# SELECT text 'abc' || 'def' AS "text and unknown";
 text and unknown
------------------
 abcdef
(1 row)
```

本例中分析器寻找两个参数都是text的操作符。确实有这样的操作符，两个参数都是text类型。

下面是连接两个未声明类型的值：

```
gaussdb=# SELECT 'abc' || 'def' AS "unspecified";
 unspecified
-------------
 abcdef
(1 row)
```

说明

因为查询中没有声明任何类型，所以本例中对类型没有任何初始提示。因此，分析器查找所有候选操作符，发现既存在接受字符串类型范畴的操作符也存在接受位串类型范畴的操作符。因为字符串类型范畴是首选，所以选择字符串类型范畴的首选类型text作为解析未知类型文本的声明类型。

示例3：绝对值和取反操作符类型分析。GaussDB操作符表里面有几条记录对应于前缀操作符@，它们都用于为各种数值类型实现绝对值操作。其中之一用于float8类型，它是数值类型范畴中的首选类型。因此，在面对unknown输入的时候，GaussDB会使用该类型：

```
gaussdb=# SELECT @ '-4.5' AS "abs";
 abs
-----
 4.5
(1 row)
```

系统在应用选定的操作符之前隐式的转换unknown类型的文字为float8类型。

版权所有 © 华为云计算技术有限公司

示例4：数组包含操作符类型分析。这里是解决一个操作符带有一个已知和一个未知类型输入的例子：

```
gaussdb=# SELECT array[1,2] <@ '{1,2,3}' as "is subset";
 is subset
-----------
 t
(1 row)
```

说明

GaussDB操作符表有几条记录对应于中缀操作符<@，但是只有两个可以在左侧接受一个整数数组的操作符是数组包含(anyarray <@ anyarray)和范围包含(anyelement <@ anyrange)的。因为没有多态的伪类型(参阅**伪类型**)是首选的，所以解析器不能解决这个基础上的歧义。然而最后一个解析规则告诉用户，假设未知类型的文字是和另外一个输入相同的类型，那就是整数数组。现在只有两个操作符中的一个可以匹配，所以选择数组包含。（如果用户选择了范围包含，用户将得到一个错误，因为字符串没有正确的格式成为范围的文字。）

1.9.3 函数

函数类型解析

- 从系统表pg_proc中选择所有可能被选到的函数。如果使用了一个不带模式修饰的函数名称，那么认为该函数是那些在当前搜索路径中的函数。如果给出一个带修饰的函数名，那么只考虑指定模式中的函数。

  如果搜索路径中找到了多个不同参数类型的函数。将从中选择一个合适的函数。
- 查找和输入参数类型完全匹配的函数。如果找到一个，则用之。如果输入的实参类型都是unknown类型，则不会找到匹配的函数。
- 如果未找到完全匹配，请查看该函数是否为一个特殊的类型转换函数。
- 寻找最优匹配。
  - 抛弃那些输入类型不匹配并且也不能隐式转换成匹配的候选函数。unknown文本在这种情况下可以转换成任何东西。如果只剩下一个候选项，则用之，否则继续下一步。
  - 遍历所有候选函数，保留那些输入类型匹配最准确的。此时，域被看作和它们的基本类型相同。如果没有一个函数能准确匹配，则保留所有候选。如果只剩下一个候选项，则用之，否则继续下一步。
  - 遍历所有候选函数，保留那些需要类型转换时接受首选类型位置最多的函数。如果没有接受首选类型的函数，则保留所有候选。如果只剩下一个候选项，则用之，否则继续下一步。
  - 如果有任何输入参数是unknown类型，检查剩余的候选函数对应参数位置的类型范畴。在每一个能够接受字符串类型范畴的位置使用string类型（这种对字符串的偏爱是合适的，因为unknown文本确实像字符串）。另外，如果所有剩下的候选函数都接受相同的类型范畴，则选择该类型范畴，否则抛出一个错误（因为在没有更多线索的条件下无法做出正确的选择）。现在抛弃不接受选定的类型范畴的候选函数，然后，如果任意候选函数在该范畴接受一个首选类型，则抛弃那些在该参数位置接受非首选类型的候选函数。如果没有一个候选符合这些测试则保留所有候选。如果只有一个候选函数符合，则使用它；否则，继续下一步。
  - 如果同时有unknown和已知类型的参数，并且所有已知类型的参数有相同的类型，假设unknown参数也是这种类型，检查哪一个候选函数可以在unknown参数位置接受这种类型。如果正好一个候选符合，那么使用它。否则，产生一个错误。

版权所有 © 华为云计算技术有限公司

示例

示例1：圆整函数参数类型解析。只有一个round函数有两个参数（第一个是numeric，第二个是integer）。所以下面的查询自动把第一个类型为integer的参数转换成numeric类型。

```
gaussdb=# SELECT round(4, 4);
 round
--------
 4.0000
(1 row)
```

实际上它被分析器转换成：

```
gaussdb=# SELECT round(CAST (4 AS numeric), 4);
```

因为带小数点的数值常量初始时被赋予numeric类型，因此下面的查询将不需要类型转换，并且可能会略微高效一些。

```
gaussdb=# SELECT round(4.0, 4);
```

示例2：子字符串函数类型解析。有好几个substr函数，其中一个接受text和integer类型。如果用一个未声明类型的字符串常量调用它，系统将选择接受string类型范畴的首选类型（也就是text类型）的候选函数。

```
gaussdb=# SELECT substr('1234', 3);
 substr
--------
    34
(1 row)
```

如果该字符串声明为varchar类型，就像从表中取出来的数据一样，分析器将试着将其转换成text类型。

```
gaussdb=# SELECT substr(varchar '1234', 3);
 substr
--------
    34
(1 row)
```

被分析器转换后实际上变成：

```
gaussdb=# SELECT substr(CAST (varchar '1234' AS text), 3);
```

说明

分析器从pg_cast表中了解到text和varchar是二进制兼容的，一个可以传递给接受另一个的函数而不需要做任何物理转换。因此，在这种情况下，实际上没有做任何类型转换。

如果以integer为参数调用函数，分析器将试图将其转换成text类型：

```
gaussdb=# SELECT substr(1234, 3);
substr
--------
 34
(1 row)
```

被分析器转换后实际上变成：

```
gaussdb=# SELECT substr(CAST (1234 AS text), 3);
 substr
--------
    34
(1 row)
```

 版权所有 © 华为云计算技术有限公司

1.9.4 值存储

值存储数据类型解析

1. 查找与目标字段精确匹配的项。
2. 尝试将表达式直接转换为目标类型：如果这两种类型之间存在已注册的转换函数，那么将直接调用该转换函数；如果表达式为未知类型（unknown、unspecified类型）文本，该文本字符串的内容将转换为目标类型。
3. 检查目标类型是否支持长度转换（长度转换指从某类型到其自身的转换）。如果在pg_cast表里面检索到对应的长度转换规则，那么在表达式存储到目标字段前，先对其应用该转换。这类转换函数需要接收一个integer类型的必选参数，用于标识目标字段的atttypmod值（即字段声明长度，atttypmod的解释因数据类型而异）；此外，这类转换函数还可以接收一个Boolean类型的可选参数，用于标识转换是显式的还是隐式的。转换函数需要负责实现与长度相关的语义逻辑，例如长度检查、内容截断等。

ORA 兼容模式常用类型支持的类型转换

表 1-332 常用类型转换支持

源类型	支持转换的目标类型
int1	int2, int4, int8, float4, float8, numeric, bpchar, varchar, nvarchar2, clob
int2	int1, int4, int8, float4, float8, numeric, bpchar, varchar, nvarchar2, clob
int4	int1, int2, int8, float4, float8, numeric, bpchar, varchar, nvarchar2, clob
int8	int1, int2, int4, float4, float8, numeric, bpchar, varchar, nvarchar2, clob
float4	int1, int2, int4, int8, float8, numeric, bpchar, varchar, nvarchar2, clob
float8	int1, int2, int4, int8, float4, numeric, bpchar, varchar, nvarchar2, clob
numeric	int1, int2, int4, int8, float4, float8, bpchar, varchar, nvarchar2, clob
date	timestamp, timestamptz, bpchar, varchar
datea	timestamp, timestamptz, bpchar, varchar, nvarchar2, clob
timestamp	datea, timestamptz, varchar, nvarchar2, clob
timestamptz	timestamp, datea, clob

版权所有 © 华为云计算技术有限公司

源类型	支持转换的目标类型
bpchar	int4, int8, float4, float8, numeric, date, datea, timestamp, varchar, nvarchar2, clob
varchar	int4, int8, float4, float8, numeric, date, datea, timestamp, bpchar, nvarchar2, clob, raw, blob
nvarchar2	int4, numeric, datea, timestamp, bpchar, varchar, clob
clob	int1, int2, int4, int8, float4, float8, numeric, datea, timestamp, bpchar, varchar, nvarchar2
raw	varchar, blob
blob	raw
xml	bpchar, varchar, nvarchar2, clob
xmltype	-

示例

character存储类型转换。对一个目标列定义为character(20)的语句，下面的语句显示存储值的长度正确：

```
gaussdb=# CREATE SCHEMA tpcds;
gaussdb=# CREATE TABLE tpcds.value_storage_t1 (
    VS_COL1 CHARACTER(20)
)DISTRIBUTE BY HASH (VS_COL1);
gaussdb=# INSERT INTO tpcds.value_storage_t1 VALUES('abcdef');
gaussdb=# SELECT VS_COL1, octet_length(VS_COL1) FROM tpcds.value_storage_t1;
       vs_col1        | octet_length
----------------------+--------------
 abcdef               |           20
(1 row)
)

gaussdb=# DROP TABLE tpcds.value_storage_t1;
gaussdb=# DROP SCHEMA tpcds;
```

说明

两个unknown文本缺省解析成text，这样就允许||操作符解析成text连接。然后操作符的text结果转换成bpchar("空白填充的字符型"，character类型内部名称)以匹配目标字段类型。从text到bpchar的转换是二进制兼容的，这样的转换是隐含的并且实际上不做任何函数调用。在系统表里找到长度转换函数bpchar(bpchar, integer, Boolean)并且应用于该操作符的结果和存储的字段长。这个类型相关的函数执行所需的长度检查和额外的空白填充。

1.9.5 UNION，CASE 和相关构造

SQL UNION构造必须把那些可能不太相似的类型匹配起来成为一个结果集。解析算法分别应用于联合查询的每个输出字段。INTERSECT和EXCEPT构造对不相同的类型使用和UNION相同的算法进行解析。CASE、ARRAY、VALUES、GREATEST和LEAST构造也使用同样的算法匹配它的部件表达式并且选择一个结果数据类型。

版权所有 © 华为云计算技术有限公司

UNION，CASE 和相关构造解析

- 如果所有输入都是相同的类型，并且不是unknown类型，那么解析成这种类型。
- 如果所有输入都是unknown类型则解析成text类型（字符串类型范畴的首选类型）。否则，忽略unknown输入。
- 如果输入不属于同一个类型范畴，则失败。（unknown类型除外）
- 如果输入类型是同一个类型范畴，则选择该类型范畴的首选类型。（例外：union操作会选择第一个分支的类型作为所选类型。）

  说明

  系统表pg_type中typcategory表示数据类型范畴，typispreferred表示是否是typcategory分类中的首选类型。
- 把所有输入转换为所选的类型（对于字符串保持原有长度）。如果从给定的输入到所选的类型没有隐式转换则失败。
- 若输入中含xml、xmltype、json、txid_snapshot、sys_refcursor或几何类型，则不能进行union。

以下为ORA兼容模式下常用类型UNION、CASE的推导规则，常用类型请参见**数据类型**。

说明

UNION和CASE部分类型推导与表达式的解析顺序相关。UNION类型推导按从左到右顺序处理每个子句的输出表达式；CASE类型推导从ELSE子句（默认结果表达式）开始，再从左到右处理各个WHEN子句的结果表达式。

表 1-333 数值类型

类型（纵向为类型1，横向为类型2）	int1	int2	int4	int8	float4	float8	numeric
int1	int1	int2	int4	int8	float4	float8	numeric
int2	int2	int2	int4	int8	float4	float8	numeric
int4	int4	int4	int4	int8	float4	float8	numeric
int8	int8	int8	int8	int8	float4	float8	numeric
float4	float4	float4	float4	float4	float4	float8	numeric
float8	float8	float8	float8	float8	float8	float8	float8
numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric

版权所有 © 华为云计算技术有限公司

表 1-334 日期时间类型

类型（纵向为类型1，横向为类型2）	datea	timestamp	timestamptz
datea	datea	timestamp	timestamptz
timestamp	timestamp	timestamp	timestamptz
timestamptz	timestamptz	timestamptz	timestamptz

表 1-335 字符类型

类型（纵向为类型1，横向为类型2）	bpchar	varchar	nvarchar2	clob
bpchar	bpchar	bpchar	bpchar	bpchar
varchar	varchar	varchar	varchar	varchar
nvarchar2	nvarchar2	nvarchar2	nvarchar2	nvarchar2
clob	clob	clob	clob	clob

表 1-336 二进制类型

类型（纵向为类型1，横向为类型2）	raw	blob
raw	raw	raw
blob	blob	blob

对于 case 和 coalesce，在 TD 兼容模式下的处理

- 如果所有输入都是相同的类型，并且不是unknown类型，那么解析成这种类型。
- 如果所有输入都是unknown类型则解析成text类型。
- 如果输入字符串（包括unknown，unknown当text来处理）和数字类型，那么解析成字符串类型，如果是其他不同的类型范畴，则报错。
- 如果输入类型是同一个类型范畴，则选择该类型的优先级较高的类型。
- 把所有输入转换为所选的类型。如果从给定的输入到所选的类型没有隐式转换则失败。

对于 case，在 ORA 兼容模式下的处理

decode(expr, search1, result1, search2, result2, ..., defresult)，在设置参数sql_beta_feature = a_style_coerce时，按ORA兼容模式下的处理，将整个表达式最终的返回值类型定为result1的数据类型，或者与result1同类型范畴的更高精度的数据类

版权所有 © 华为云计算技术有限公司

型。例如，numeric与int同属数值类型范畴，但numeric比int精度要高，具有更高优先级。对于CASE WHEN，ORA兼容性下与默认行为相同。

- 如果所有输入都是相同的类型，并且不是unknown类型，那么解析成这种类型。否则，进入后续步骤。
- 将result1的数据类型置为最终的返回值类型preferType，其所属类型范畴为preferCategory。
- 依次考虑result2、result3直至defresult的数据类型。如果其类型范畴也是preferCategory，即与result1具有相同的类型范畴，则判断其精度（优先级）是否高于preferType，如果精度高于preferType，则将preferType更新为更高精度的数据类型；如果其类型范畴不是preferCategory，则判断其数据类型是否可以隐式转换为preferType，不可以则报错。
- 将最终preferType记录的数据类型作为整个表达式最终的返回值类型；表达式的结果向此类型进行隐式转换。

注1：

为了兼容一种特殊情况，即表示了超大数字的字符类型向数值类型转换的情况，例如select decode(1, 2, 2, '53465465676465454657567678676')，大数超过了bigint、double等的表示范围。所以，当result1的类型范畴为数值类型，且不满足上述“所有输入都是相同的类型”条件时，将返回值的类型直接置为numeric，以兼容此种特殊情况。

注2：

数值类型的优先级排序：numeric>float8>float4>int8>int4>int2>int1

字符类型的优先级排序：text>varchar=nvarchar2>bpchar>char

日期类型的优先级排序：timestamptz>timestamp>datea（仅在ORA模式下生效）>smalldatetime>date>abstime>timetz>time

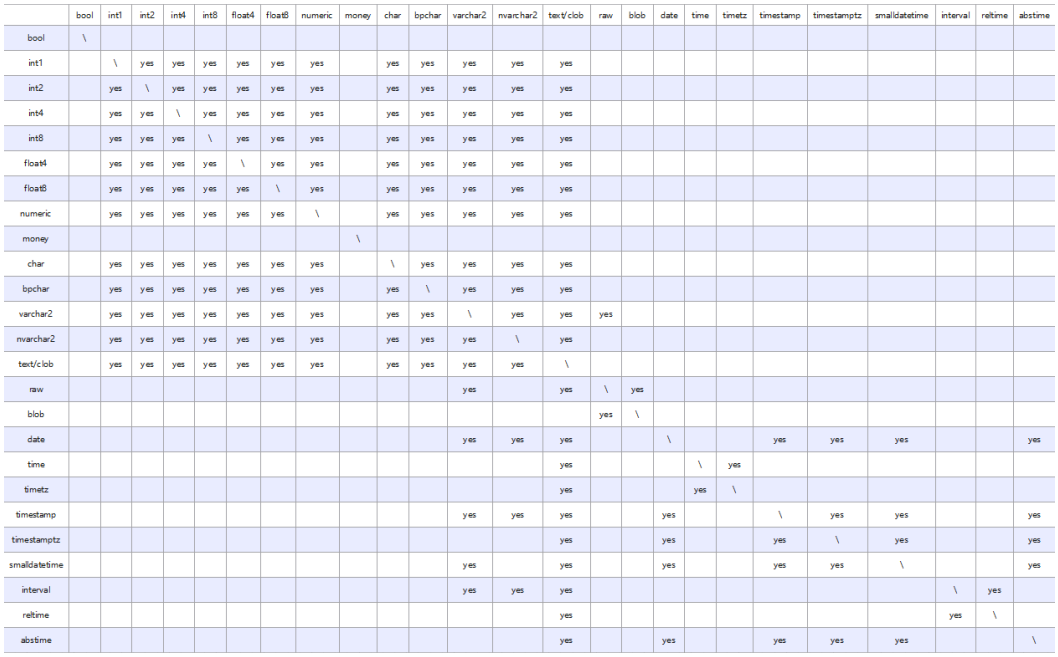
日期跨度类型的优先级排序：interval>tinterval>reltime

注3：

ORA兼容模式，当参数set sql_beta_feature的值设置为'a_style_coerce'时，所支持的隐式类型转换如**图1-14**所示，\代表不需要转换，yes表示支持，空白表示不支持。

版权所有 © 华为云计算技术有限公司

图 1-14 隐式转换支持情况



	bool	int1	int2	int4	int8	float4	float8	numeric	money	char	bpchar	varchar2	nvarchar2	text/clob	raw	blob	date	time	timetz	timestamp	timestamptz	smalldatetime	interval	reltime	abstime
bool	\																								
int1		\	yes	yes	yes	yes	yes	yes		yes	yes	yes	yes	yes											
int2		yes	\	yes	yes	yes	yes	yes		yes	yes	yes	yes	yes											
int4		yes	yes	\	yes	yes	yes	yes		yes	yes	yes	yes	yes											
int8		yes	yes	yes	\	yes	yes	yes		yes	yes	yes	yes	yes											
float4		yes	yes	yes	yes	\	yes	yes		yes	yes	yes	yes	yes											
float8		yes	yes	yes	yes	yes	\	yes		yes	yes	yes	yes	yes											
numeric		yes	yes	yes	yes	yes	yes	\		yes	yes	yes	yes	yes											
money									\																
char		yes	yes	yes	yes	yes	yes	yes		\	yes	yes	yes	yes											
bpchar		yes	yes	yes	yes	yes	yes	yes		yes	\	yes	yes	yes											
varchar2		yes	yes	yes	yes	yes	yes	yes		yes	yes	\	yes	yes	yes										
nvarchar2		yes	yes	yes	yes	yes	yes	yes		yes	yes	yes	\	yes											
text/clob		yes	yes	yes	yes	yes	yes	yes		yes	yes	yes	yes	\											
raw												yes		yes	\	yes									
blob															yes	\									
date												yes	yes	yes			\			yes	yes	yes			yes
time														yes				\	yes						
timetz														yes				yes	\						
timestamp												yes	yes	yes			yes			\	yes	yes			yes
timestamptz														yes			yes			yes	\	yes			yes
smalldatetime												yes		yes			yes			yes	yes	\			yes
interval												yes	yes	yes									\	yes	
reltime														yes									yes	\	
abstime														yes			yes			yes	yes	yes			\

示例

示例1：Union中的待定类型解析。unknown类型文本'b'将被解析成text类型。

```
gaussdb=# SELECT text 'a' AS "text" UNION SELECT 'b';
 text
------
 a
 b
(2 rows)
```

示例2：简单Union中的类型解析。文本1.2的类型为numeric，而且integer类型的1可以隐含地转换为numeric，因此使用这个类型。

```
gaussdb=# SELECT 1.2 AS "numeric" UNION SELECT 1;
 numeric
---------
       1
     1.2
(2 rows)
```

示例3：转置Union中的类型解析。类型real不能被隐含转换成integer，但是integer可以隐含转换成real，那么联合的结果类型将是real。

```
gaussdb=# SELECT 1 AS "real" UNION SELECT CAST('2.2' AS REAL);
 real
------
    1
  2.2
(2 rows)
```

示例4：TD模式下，coalesce参数输入int和varchar类型，那么解析成varchar类型。ORA模式下会报错。

```
--创建ORA兼容模式的数据库ora_1。
gaussdb=# CREATE DATABASE ora_1 dbcompatibility = 'ORA';

--切换数据库为ora_1。
gaussdb=# \c ora_1
```


版权所有 © 华为云计算技术有限公司

```
--创建表t1。
ora_1=# CREATE TABLE t1(a int, b varchar(10));

--查看coalesce参数输入int和varchar类型的查询语句的执行计划。
a_1=# EXPLAIN SELECT coalesce(a, b) FROM t1;
ERROR:  COALESCE types integer and character varying cannot be matched
LINE 1: EXPLAIN SELECT coalesce(a, b) FROM t1;
                               ^
CONTEXT:  referenced column: coalesce

--删除表。
ora_1=# DROP TABLE t1;

--切换数据库为testdb。
ora_1=# \c testdb

--在TD模式下，创建TD兼容模式的数据库td_1。
gaussdb=# CREATE DATABASE td_1 dbcompatibility = 'TD';

--切换数据库为td_1。
gaussdb=# \c td_1

--创建表t2。
td_1=# CREATE TABLE t2(a int, b varchar(10));

--查看coalesce参数输入int和varchar类型的查询语句的执行计划。
td_1=# EXPLAIN VERBOSE select coalesce(a, b) from t2;
                                QUERY PLAN
-------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Output: (COALESCE((t2.a)::character varying, t2.b))
   Node/s: All datanodes
   Remote query: SELECT COALESCE(a::character varying, b) AS "coalesce" FROM public.t2
(4 rows)

--删除表。
td_1=# DROP TABLE t2;

--切换数据库为testdb。
td_1=# \c testdb

--删除ORA和TD模式的数据库。
gaussdb=# DROP DATABASE ora_1;
gaussdb=# DROP DATABASE td_1;
```

示例5：ORA模式下，将整个表达式最终的返回值类型定为result1的数据类型，或者与result1同类型范畴的更高精度的数据类型。

```
--在ORA模式下，创建ORA兼容模式的数据库ora_1。
gaussdb=# CREATE DATABASE ora_1 dbcompatibility = 'ORA';

--切换数据库为ora_1。
gaussdb=# \c ora_1

--开启Decode兼容性参数。
ora_1=# SET sql_beta_feature='a_style_coerce';

--创建表t1。
ora_1=# CREATE TABLE t1(c_int int, c_float8 float8, c_char char(10), c_text text, c_date date);

--插入数据。
ora_1=# INSERT INTO t1 VALUES(1, 2, '3', '4', date '12-10-2010');

--result1类型为char，defresult类型为text，text精度更高，返回值的类型由char更新为text。
ora_1=# SELECT decode(1, 2, c_char, c_text) AS result, pg_typeof(result) FROM t1;
 result | pg_typeof
--------+-----------
 4      | text
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
--result1类型为int，属于数值类型范畴，返回值的类型置为numeric。
ora_1=# SELECT decode(1, 2, c_int, c_float8) AS result, pg_typeof(result) FROM t1;
 result | pg_typeof
--------+-----------
      2 | numeric
(1 row)

--不存在defresult数据类型向result1数据类型之间的隐式转换，报错处理。
ora_1=# SELECT decode(1, 2, c_int, c_date) FROM t1;
ERROR:  CASE types integer and timestamp without time zone cannot be matched
LINE 1: SELECT decode(1, 2, c_int, c_date) FROM t1;
                                   ^
CONTEXT:  referenced column: c_date

--关闭Decode兼容性参数。
ora_1=# SET sql_beta_feature='none';

--删除表。
ora_1=# DROP TABLE t1;
DROP TABLE

--切换数据库为testdb。
ora_1=# \c testdb

--删除ORA模式的数据库。
gaussdb=# DROP DATABASE ora_1;
DROP DATABASE
```

1.10 系统操作

GaussDB通过SQL语句执行不同的系统操作，比如：设置变量，显示执行计划和垃圾收集等操作。

设置变量

设置会话或事务中需要使用的各种参数，请参考**SET**。

显示执行计划

显示GaussDB为SQL语句规划的执行计划，请参考**EXPLAIN**。

事务日志检查点

预写式日志（WAL）缺省时在事务日志中每隔一段时间放置一个检查点。CHECKPOINT会立即进行检查，而不是等到下一次调度时的检查点。请参考**CHECKPOINT**。

垃圾收集

进行垃圾收集以及可选择的对数据库进行分析。请参考**VACUUM**。

收集统计信息

收集与数据库中表内容相关的统计信息。请参考**ANALYZE | ANALYSE**。

设置当前事务的约束检查模式

设置当前事务里的约束检查的特性。请参考**SET CONSTRAINTS**。

版权所有 © 华为云计算技术有限公司

1.11 事务控制

事务是用户定义的一个数据库操作序列，这些操作要么全做要么全不做，是一个不可分割的工作单位。

启动事务

GaussDB通过START TRANSACTION和BEGIN语法启动事务，请参考**START TRANSACTION**和**BEGIN**。

设置事务

GaussDB通过SET TRANSACTION或者SET LOCAL TRANSACTION语法设置事务，请参考**SET TRANSACTION**。

提交事务

GaussDB通过COMMIT或者END可完成提交事务的功能，即提交事务的所有操作，请参考**COMMIT | END**。

回滚事务

回滚是在事务运行的过程中发生了某种故障，事务不能继续执行，系统将事务中对数据库的所有已完成的操作全部撤销。请参考**ROLLBACK**。

说明

数据库中收到的一次执行请求（不在事务块中），如果含有多条语句，将会被打包成一个事务，如果其中有一个语句失败，那么整个请求都将会被回滚。

1.12 全文检索

1.12.1 介绍

1.12.1.1 全文检索概述

文本搜索操作符在数据库中已存在多年。GaussDB为文本数据类型提供~、~*、LIKE和ILIKE操作符；但它们缺乏现代信息系统所要求的许多必要文本属性，这些缺陷可以通过使用全文检索索引及词典进行解决。

文本检索缺乏信息系统所要求的必要属性：

- 没有语义支持，即使是英语。

  由于要识别派生词并不是那么容易，因此正则表达式也不能满足要求。例如：satisfies和satisfy，当使用正则表达式寻找satisfy时，并不会查询到包含satisfies的文档。用户可以使用OR搜索多种派生形式，但过程非常繁琐。并且有些词会有上千的派生词，因此容易出错。
- 没有对搜索结果的分类（排序）。当搜索出成千的文档时，查找效率很低。

版权所有 © 华为云计算技术有限公司

- 由于没有索引的支持，每一次的搜索需要遍历所有的文档，整体搜索比较缓慢。

使用全文检索可以对文档进行预处理，并且可以使后续的搜索更快速。预处理过程包括：

- 将文档解析成token。

  为每个文档标记不同类别的token是非常有必要的，例如：数字、文字、复合词、电子邮件地址，这样就可以做不同的处理。原则上token的类别依赖于具体的应用，但对于大多数的应用来说，可以使用一组预定义的token类。

- 将token转换为词素。

  词素像token一样是一个字符串，但它已经标准化处理，这样同一个词的不同形式是一样的。标准化通常包括：将大写字母折成小写字母、删除后缀（如英语中的s或者es）。这将允许通过搜索找到同一个词的不同形式，不需要繁琐地输入所有可能的变形样式。同时，这一步通常会删除停用词。这些停用词通常因为太常见而对搜索无用。（总之，token是文档文本的原片段，而词素被认为是有用的索引和搜索词。）GaussDB使用词典执行这一步，且提供了各种标准的词典。

- 保存搜索优化后的预处理文档。

  每个文档可以呈现为标准化词素的有序组合。伴随词素通常还需要存储词素位置信息以用于邻近排序。因此文档包含的查询词越密集其排序越高。

词典能够对token如何标准化做到细粒度控制。使用合适的词典，可以定义不被索引的停用词。

数据类型tsvector用于存储预处理文档，tsquery用于存储查询条件，详细请参见**文本检索类型**。为这些数据类型提供的函数和操作符请参见**文本检索函数和操作符**章节。其中最重要的是匹配运算符@@，将在**基本文本匹配**中介绍。

1.12.1.2 文档概念

文档是全文搜索系统的搜索单元，例如：杂志上的一篇文章或电子邮件消息。文本搜索引擎必须能够解析文档，而且可以存储父文档的关联词素（关键词）。这些关联词素用来搜索包含查询词的文档。

在GaussDB中，文档通常是一个数据库表中一行的文本字段，或者这些字段的可能组合（级联）。文档可能存储在多个表中或者需动态获取。即一个文档由被索引化的不同部分构成，因此无法存储为一个整体。比如：

```
gaussdb=# CREATE TABLE document_table (id int,person text,time timestamp,location text,thing
text,document text);
CREATE TABLE
gaussdb=# INSERT INTO document_table VALUES('1','Jack','2024-01-01 10:00:00','company','working','Jack
worked at the company on 2024-01-01 10:00:00.');
INSERT 0 1
gaussdb=# SELECT person || '-' || thing || '-' || location || '-' || time AS identify_serials FROM document_table
WHERE id =1;
           identify_serials
----------------------------------------
 Jack-working-company-2024-01-01 10:00:00
(1 row)
gaussdb=# DROP TABLE document_table;
```

另外一种可能是：文档在文件系统中作为简单的文本文件存储。在这种情况下，数据库可以用于存储全文索引并且执行搜索，同时可以使用一些唯一标识从文件系统中检索文档。然而，从数据库外部检索文件需要拥有系统管理员权限或者特殊函数支持。因此，还是将所有数据保存在数据库中比较方便。同时，将所有数据保存在数据库中可以方便地访问文档元数据以便于索引和显示。

为了实现文本搜索目的，必须将每个文档减少至预处理后的tsvector格式。搜索和相关性排序都是在tsvector形式的文档上执行的。原始文档只有在被选中要呈现给用户时才

版权所有 © 华为云计算技术有限公司

会被检索。因此，通常将tsvector说成文档，但是很显然其实它只是完整文档的一种紧凑表示。

1.12.1.3 基本文本匹配

GaussDB的全文检索基于匹配算子@@，当一个tsvector(document)匹配到一个tsquery(query)时，则返回true。其中，tsvector(document)和tsquery(query)两种数据类型可以任意排序。

```
gaussdb=# SELECT 'a fat cat sat on a mat and ate a fat rat'::tsvector @@ 'cat & rat'::tsquery AS RESULT;
 result
----------
 t
(1 row)

gaussdb=# SELECT 'fat & cow'::tsquery @@ 'a fat cat sat on a mat and ate a fat rat'::tsvector AS RESULT;
 result
----------
 f
(1 row)
```

正如上面例子表明，tsquery不仅是文本，且比tsvector包含的要多。tsquery包含已经标注化为词条的搜索词，同时可能是使用AND、OR、NOT或PHRASE操作符连接的多个术语。详细请参见**文本检索类型**。函数to_tsquery、plainto_tsquery和phraseto_tsquery对于将用户书写文本转换成适合的tsquery是非常有用的，比如将文本中的词标准化。类似地，to_tsvector用于解析和标准化文档字符串。因此，实际文本搜索匹配如下示例所示：

```
gaussdb=# SELECT to_tsvector('fat cats ate fat rats') @@ to_tsquery('fat & rat') AS RESULT;
result
----------
 t
(1 row)
```

需要注意的是，下面这种方式是不可行的：

```
gaussdb=# SELECT 'fat cats ate fat rats'::tsvector @@ to_tsquery('fat & rat')AS RESULT;
result
----------
 f
(1 row)
```

由于tsvector没有对rats进行标准化，所以rats不匹配rat。

@@操作符也支持text输入，允许一个文本字符串的显示转换为tsvector或者在简单情况下忽略tsquery。可用形式是：

```
tsvector @@ tsquery
tsquery  @@ tsvector
text @@ tsquery
text @@ text
```

已经看到了前面两种，形式text @@ tsquery等价于to_tsvector(text) @@ tsquery，而text @@ text等价于to_tsvector(text) @@ plainto_tsquery(text)。

1.12.1.4 配置

全文检索功能还可以做更多事情，例如：忽略索引某个词（停用词），处理同义词和使用复杂解析。这些功能通过文本搜索配置控制。GaussDB支持多语言的预定义的文本搜索配置。

在安装期间选择一个合适的文本搜索配置，并且在gaussdb.conf中相应的设置default_text_search_config。如果在为了整个集群使用同一个文本搜索配置项，可以

版权所有 © 华为云计算技术有限公司

使用gaussdb.conf中的值。如果需要在集群中使用不同文本搜索配置，可以使用ALTER DATABASE ... SET在任一数据库进行配置。用户也可以在每个会话中设置default_text_search_config。

每一个文本搜索函数都有一个可选的regconfig参数，因此要使用的配置可以被显式指定。只有当这个参数被忽略时，default_text_search_config才被使用。

1.12.2 表和索引

1.12.2.1 搜索表

在不使用索引的情况下也可以进行全文检索。

- 一个简单查询：将body字段中包含america的每一行打印出来。

```
gaussdb=# DROP SCHEMA IF EXISTS tsearch CASCADE;
gaussdb=# CREATE SCHEMA tsearch;
gaussdb=# CREATE TABLE tsearch.pgweb(id int, body text, title text, last_mod_date date)
with(storage_type = ASTORE);
gaussdb=# INSERT INTO tsearch.pgweb VALUES(1, 'America is a rock band, formed in England in
1970 by multi-instrumentalists Dewey Bunnell, Dan Peek, and Gerry Beckley.', 'America', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(2, 'England is a country that is part of the United
Kingdom. It shares land borders with Scotland to the north and Wales to the west.', 'England',
'2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(3, 'Australia, officially the Commonwealth of
Australia, is a country comprising the mainland of the Australian continent, the island of Tasmania,
and numerous smaller islands.', 'Australia', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(4, 'Japan is an island country in East Asia.', 'Japan',
'2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(5, 'Germany, officially the Federal Republic of
Germany, is a sovereign state and federal parliamentary republic in central-western Europe.',
'Germany', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(6, 'France, is a sovereign state comprising territory in
western Europe and several overseas regions and territories.', 'France', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(7, 'Italy officially the Italian Republic, is a unitary
parliamentary republic in Europe.', 'Italy', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(8, 'India, officially the Republic of India, is a country
in South Asia.', 'India', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(9, 'Brazil, officially the Federative Republic of Brazil,
is the largest country in both South America and Latin America.', 'Brazil', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(10, 'Canada is a country in the northern half of
North America.', 'Canada', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(11, 'Mexico, officially the United Mexican States, is a
federal republic in the southern part of North America.', 'Mexico', '2010-1-1');

gaussdb=# SELECT id, body, title FROM tsearch.pgweb WHERE to_tsvector('english', body) @@
to_tsquery('english', 'america');
 id |                                                 body                                                 |  title
----
+--------------------------------------------------------------------------------------------------------------------------------
----+---------
  1 | America is a rock band, formed in England in 1970 by multi-instrumentalists Dewey Bunnell, Dan
Peek, and Gerry Beckley. | America
  9 | Brazil, officially the Federative Republic of Brazil, is the largest country in both South America
and Latin America.   | Brazil
 10 | Canada is a country in the northern half of North
America.                                                | Canada
 11 | Mexico, officially the United Mexican States, is a federal republic in the southern part of North
America.             | Mexico
(4 rows)
```

像America这样的相关词也会被找到，因为这些词都被处理成了相同标准的词条。

上面的查询指定english配置来解析和规范化字符串。也可以省略此配置，通过default_text_search_config进行配置设置：

```
gaussdb=# SHOW default_text_search_config;
 default_text_search_config
```


版权所有 © 华为云计算技术有限公司

```
----------------------------
 pg_catalog.english
(1 row)

gaussdb=# SELECT id, body, title FROM tsearch.pgweb WHERE to_tsvector(body) @@
to_tsquery('america');
 id |                                                 body                                                 |  title
----
+----------------------------------------------------------------------------------------------------------------------------------
----+---------
  1 | America is a rock band, formed in England in 1970 by multi-instrumentalists Dewey Bunnell, Dan
Peek, and Gerry Beckley. | America
  9 | Brazil, officially the Federative Republic of Brazil, is the largest country in both South America
and Latin America.   | Brazil
 10 | Canada is a country in the northern half of North
America.                                             | Canada
 11 | Mexico, officially the United Mexican States, is a federal republic in the southern part of North
America.             | Mexico
(4 rows)
```

- 检索出在title或者body字段中包含north和america的最近10篇文档：

```
gaussdb=# SELECT title FROM tsearch.pgweb WHERE to_tsvector(title || ' ' || body) @@
to_tsquery('north & america') ORDER BY last_mod_date DESC LIMIT 10;
 title
--------
 Canada
 Mexico
(2 rows)

-- 删除表
gaussdb=# DROP TABLE tsearch.pgweb;
gaussdb=# DROP SCHEMA tsearch;
```

为了清晰，举例中没有调用coalesce函数在两个字段中查找包含NULL的行。

以上例子均在没有索引的情况下进行查询。对于大多数应用程序来说，这个方法很慢。因此除了偶尔的特定搜索，文本搜索在实际使用中通常需要创建索引。

1.12.2.2 创建索引

为了加速文本搜索，可以创建GIN索引。

```
gaussdb=# CREATE SCHEMA tsearch;
gaussdb=# CREATE TABLE tsearch.pgweb(id int, body text, title text, last_mod_date date) with(storage_type
= ASTORE);
gaussdb=# INSERT INTO tsearch.pgweb VALUES(1, 'America is a rock band, formed in England in 1970 by
multi-instrumentalists Dewey Bunnell, Dan Peek, and Gerry Beckley.', 'America', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(2, 'England is a country that is part of the United
Kingdom. It shares land borders with Scotland to the north and Wales to the west.', 'England', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(3, 'Australia, officially the Commonwealth of Australia, is a
country comprising the mainland of the Australian continent, the island of Tasmania, and numerous
smaller islands.', 'Australia', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(4, 'Japan is an island country in East Asia.', 'Japan',
'2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(5, 'Germany, officially the Federal Republic of Germany, is
a sovereign state and federal parliamentary republic in central-western Europe.', 'Germany', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(6, 'France, is a sovereign state comprising territory in
western Europe and several overseas regions and territories.', 'France', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(7, 'Italy officially the Italian Republic, is a unitary
parliamentary republic in Europe.', 'Italy', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(8, 'India, officially the Republic of India, is a country in
South Asia.', 'India', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(9, 'Brazil, officially the Federative Republic of Brazil, is the
largest country in both South America and Latin America.', 'Brazil', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(10, 'Canada is a country in the northern half of North
America.', 'Canada', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(11, 'Mexico, officially the United Mexican States, is a
federal republic in the southern part of North America.', 'Mexico', '2010-1-1');
gaussdb=# CREATE INDEX pgweb_idx_1 ON tsearch.pgweb USING gin(to_tsvector('english', body));
```


版权所有 © 华为云计算技术有限公司

to_tsvector()函数有两个版本。只输入一个参数的版本和输入两个参数的版本。只输入一个参数时，系统默认采用default_text_search_config所指定的默认配置。

请注意：创建索引时请使用to_tsvector的两参数版本。只有指定了文本搜索配置的全文检索函数才可以在索引表达式中正确被使用。这是因为索引的内容必须不受default_text_search_config的影响，否则索引内容可能不一致。由于default_text_search_config的值可以随时调整，从而导致不同条目生成的tsvector采用了不同的文本搜索配置，并且没有办法区分究竟使用了哪一个文本搜索配置。正确地转储和恢复这样的索引也是不可能的。

因为在上述创建索引中to_tsvector使用了两个参数，只有当查询时也使用了两个参数，且参数值与索引中相同时，才会使用该索引。即WHERE to_tsvector('english', body) @@ 'a & b' 可以使用索引，但WHERE to_tsvector(body) @@ 'a & b'不能使用索引。这确保只使用这样的索引——索引各条目是使用相同的分词器创建的。

索引中的分词器名称由另一列指定时可以建立更复杂的表达式索引。例如：

```
gaussdb=# ALTER TABLE tsearch.pgweb ADD COLUMN config_name regconfig;
gaussdb=# CREATE INDEX pgweb_idx_2 ON tsearch.pgweb USING gin(to_tsvector('ngram', body));
```

这允许在同一个索引中有混合配置，同时记录具体的配置被用于每一个索引项。例如，当如果文档集合中包含不同语言的文档时，这就可能会有用。同样，要使用索引的查询也必须匹配索引表达式，例如WHERE to_tsvector(config_name, body) @@ 'a & b'。

索引甚至可以连接列：

```
gaussdb=# CREATE INDEX pgweb_idx_3 ON tsearch.pgweb USING gin(to_tsvector('english', title || ' ' ||
body));
```

另一个方法是创建一个单独的tsvector列控制to_tsvector的输出。下面的例子是title和body的连接， 当其它是NULL的时候，使用coalesce确保一个字段仍然会被索引：

```
gaussdb=# ALTER TABLE tsearch.pgweb ADD COLUMN textsearchable_index_col tsvector;
gaussdb=# UPDATE tsearch.pgweb SET textsearchable_index_col = to_tsvector('english', coalesce(title,'') || ' '
|| coalesce(body,''));
```

然后为加速搜索创建一个GIN索引：

```
gaussdb=# CREATE INDEX textsearch_idx_4 ON tsearch.pgweb USING gin(textsearchable_index_col);
```

现在就可以执行一个快速全文搜索：

```
gaussdb=# SELECT title
FROM tsearch.pgweb
WHERE textsearchable_index_col @@ to_tsquery('north & america')
ORDER BY last_mod_date DESC
LIMIT 10;

 title
--------
 Canada
 Mexico
(2 rows)

-- 删除表
gaussdb=# DROP TABLE tsearch.pgweb;
gaussdb=# DROP SCHEMA tsearch;
```

相比于一个表达式索引，单独列方法的一个优势是：它没有必要在查询时明确指定分词器以便能使用索引。正如上面例子所示，查询可以依赖于default_text_search_config。另一个优势是搜索比较快速，因为它没有必要重新利用to_tsvector调用来验证索引匹配。表达式索引方法更容易建立，且它需要较少的磁盘空间，因为tsvector形式没有明确存储。

版权所有 © 华为云计算技术有限公司

说明

- GIN索引仅在ASTORE存储引擎下支持。
- 其他GIN索引的使用约束请参见**CREATE INDEX**章节对GIN索引的说明。

tsvector支持在USTORE存储引擎下创建UGIN索引。

```
-- 创建表。
gaussdb=# CREATE TABLE ustore_tbl (id int, content text, ts tsvector) WITH (storage_type = USTORE);

-- 创建索引。
gaussdb=# CREATE INDEX textsearch_ugin_idx1 ON ustore_tbl USING ugin(ts);
gaussdb=# CREATE INDEX textsearch_ugin_idx2 ON ustore_tbl USING ugin(to_tsvector('english', content));

-- 删除表。
gaussdb=# DROP TABLE ustore_tbl;
```

说明

- UGIN索引仅在USTORE存储引擎下支持，其他方面与GIN索引基本一致。
- 其他UGIN索引的使用约束请参见**CREATE INDEX**章节对UGIN索引的说明。

1.12.3 控制文本搜索

为了执行全文搜索，必须有个函数创建来自文档的tsvector和来自用户查询的tsquery。同时，为了以有效的顺序返回结果，需要一个函数对比文档的查询相关性。同时，能很好地显示结果也是很重要的。GaussDB支持所有这些函数。

1.12.3.1 解析文档

GaussDB中提供了to_tsvector函数把文档处理成tsvector数据类型。

```
to_tsvector([ config regconfig, ] document text) returns tsvector
```

to_tsvector将文本文档解析为token，再将token简化到词素，并返回一个tsvector。在tsvector中包含了各词素及词素在文档中的位置。文档根据默认的或指定的文本搜索分词器进行处理。这里有一个简单的例子：

```
gaussdb=# SELECT to_tsvector('english', 'a fat  cat sat on a mat - it ate a fat rats');
                  to_tsvector
-----------------------------------------------------
 'ate':9 'cat':3 'fat':2,11 'mat':7 'rat':12 'sat':4
```

通过以上例子可发现，在tsvector的结果中不包含词a、on或者it，且rats变成了rat，并忽略了标点符号“-”。

to_tsvector函数内部调用了一个解析器，将文档的文本分解成token并给每个token指定一个类型。对于每个token，有一系列词典可供查询。词典系列因token类型的不同而不同。识别token的第一本词典将产生一个或多个标准词素来表示token。例如：

- rats变成rat因为词典认为词rats是rat的复数形式。
- 有些词被作为停用词（请参见**停用词**），这样它们就会被忽略，因为它们出现得太过频繁以致于在搜索中没有用处。比如例子中的a、on和it。
- 如果没有词典识别token，那么它也被忽略。在这个例子中，符号“-”被忽略，因为词典没有给它分配token类型（空间符号），即空间符号永远不会被索引。

语法解析器、词典和要索引的token类型由选定的文本搜索配置决定。可以在同一个数据库中有多种不同的配置，并且有用于多种语言的预定义配置。在以上例子中，使用的是英语的默认english。

版权所有 © 华为云计算技术有限公司

setweight函数可以给tsvector的记录赋予权重，权重值是字母A、B、C、D之一，用于标记来自文档不同部分的记录，比如标题、正文。之后，便可以根据文档各部分的权重值对搜索结果排序。

因为to_tsvector(NULL)会返回空，当字段可能是空的时候，建议使用coalesce。以下是推荐的为结构化文档创建tsvector的方法：

```
gaussdb=# CREATE TABLE tsearch.tt (id int, title text, keyword text, abstract text, body text, ti tsvector);

gaussdb=# INSERT INTO tsearch.tt(id, title, keyword, abstract, body) VALUES (1, 'America', 'New York',
'America','America is a rock band, formed in England in 1970 by multi-instrumentalists Dewey Bunnell, Dan
Peek, and Gerry Beckley.');

gaussdb=# UPDATE tsearch.tt SET ti =
    setweight(to_tsvector(coalesce(title,'')), 'A')    ||
    setweight(to_tsvector(coalesce(keyword,'')), 'B')  ||
    setweight(to_tsvector(coalesce(abstract,'')), 'C') ||
    setweight(to_tsvector(coalesce(body,'')), 'D');
gaussdb=# DROP TABLE tsearch.tt;
```

上例使用setweight标记已完成的tsvector中的每个词的来源，并且使用tsvector连接操作符||合并标记过的tsvector值，**处理tsvector**一节详细介绍了这些操作。

1.12.3.2 解析查询

GaussDB提供了函数to_tsquery、plainto_tsquery和phraseto_tsquery将查询转换为tsquery数据类型，to_tsquery提供比plainto_tsquery、phraseto_tsquery更多的功能，但对其输入要求更严格。

```
to_tsquery([ config regconfig, ] querytext text) returns tsquery
```

to_tsquery从querytext中创建一个tsquery，querytext必须由运算符& (AND)、| (OR)、！(NOT)和<->/<n> (PHRASE)分隔的单个token组成。这些运算符可以用圆括号分组。to_tsquery输入必须遵循tsquery输入的通用规则，具体请参见**文本检索类型**。不同的是基本tsquery以token表面值作为输入，而to_tsquery使用指定或默认配置将每个token标准化成词素，并依据配置丢弃属于停用词的token。例如：

```
gaussdb=# SELECT to_tsquery('english', 'The & Fat & Rats');
   to_tsquery
---------------
 'fat' & 'rat'
(1 row)
```

与基本tsquery中的输入一样，weight(s)可以附加到每个词素来限制它只匹配那些有相同weight(s)的tsvector词素。例如：

```
gaussdb=# SELECT to_tsquery('english', 'Fat | Rats:AB');
    to_tsquery
------------------
 'fat' | 'rat':AB
(1 row)
```

*也可以附加到词素来指定前缀匹配：

```
gaussdb=# SELECT to_tsquery('supern:*A & star:A*B');
        to_tsquery
--------------------------
 'supern':*A & 'star':*AB
(1 row)
```

这样的词素将匹配tsquery中指定字符串和权重的项。

```
plainto_tsquery([ config regconfig, ] querytext text) returns tsquery
```


版权所有 © 华为云计算技术有限公司

plainto_tsquery将未格式化的文本querytext变换为tsquery。类似于to_tsvector，文本被解析并且标准化，然后在存在的词之间插入& (AND)布尔算子。

例如：

```
gaussdb=# SELECT plainto_tsquery('english', 'The Fat Rats');
 plainto_tsquery
-----------------
 'fat' & 'rat'
(1 row)
```

请注意：plainto_tsquery无法识别运算符、权重标签，或在其输入中的前缀匹配标签。

```
gaussdb=# SELECT plainto_tsquery('english', 'The Fat & Rats:C');
   plainto_tsquery
---------------------
 'fat' & 'rat' & 'c'
(1 row)
```

所有输入的标点符号作为空格符号丢弃。

与plainto_tsquery类似，phraseto_tsquery也可将未格式化的文本querytext变换为tsquery，连接词为<-> (PHRASE)。

1.12.3.3 排序查询结果

排序试图针对特定查询衡量文档的相关度，从而将众多的匹配文档中相关度最高的文档排在最前。GaussDB提供了两个预置的排序函数。函数考虑了词法、距离和结构信息；即考虑查询词在文档中出现的频率、紧密程度、以及出现的地方在文档中的重要性。然而，相关性的概念是模糊的，并且是跟应用强相关的。不同的应用程序可能需要额外的信息来排序，比如，文档的修改时间，内置的排序函数等。也可以开发自己的排序函数或者采用附加因素组合这些排序函数的结果来满足特定需求。

两个预置的排序函数：

```
ts_rank([ weights float4[], ] vector tsvector, query tsquery [, normalization integer ]) returns float4
```

ts_rank函数基于向量的匹配词位的频率来排名向量。

```
ts_rank_cd([ weights float4[], ] vector tsvector, query tsquery [, normalization integer ]) returns float4
```

ts_rank_cd函数为给定文档向量和查询计算覆盖密度排名，覆盖密度类似于ts_rank排名，不过它会考虑匹配词位相互之间的接近度。

该函数需要位置信息的输入。当匹配的token没有位置信息时，结果将会是零。

对于这两个函数，可选的weights参数提供给词加权重的能力，词的权重大小取决于所加的权值。权重阵列指定在排序时为每类词汇加多大的权重。

```
{D-weight, C-weight, B-weight, A-weight}
```

如果没有提供weights，则使用缺省值：{0.1, 0.2, 0.4, 1.0}。

通常的权重是用来标记文档特殊领域的词，如标题或最初的摘要，所以相对于文章主体中的词它们有着更高或更低的重要性。

由于较长的文档有更多的机会包含查询词，因此有必要考虑文档的大小。例如，包含有5个搜索词的一百字文档比包含有5个搜索词的一千字文档相关性更高。两个预置的排序函数都采用了一个整型的标准化选项来定义文档长度是否影响排序及如何影响。这个整型选项控制多个行为，所以它是一个屏蔽字：可以使用|指定一个或多个行为（例如，2|4）。

 版权所有 © 华为云计算技术有限公司

- 0（缺省）表示：跟长度大小没有关系。
- 1表示：排名（rank）除以(文档长度的对数+1)。
- 2表示：排名除以文档的长度。
- 4表示：排名除以两个扩展词间的调和平均距离。只能使用ts_rank_cd实现。
- 8表示：排名除以文档中单独词的数量。
- 16表示：排名除以单独词数量的对数+1。
- 32表示：排名除以排名本身+1。

当指定多个标志位时，会按照所列的顺序依次进行转换。

需要特别注意的是，排序函数不使用任何全局信息，所以不可能产生一个某些情况下需要的1%或100%的理想标准值。标准化选项32 (rank/(rank+1))可用于所有规模的从零到一之间的排序，这只是一个表面变化，它不会影响搜索结果的排序。

下面是一个关于仅选择排名前十的匹配示例：

```
gaussdb=# CREATE SCHEMA tsearch;
gaussdb=# CREATE TABLE tsearch.pgweb(id int, body text, title text, last_mod_date date) with(storage_type = ASTORE);
gaussdb=# INSERT INTO tsearch.pgweb VALUES(1, 'America is a rock band, formed in England in 1970 by multi-instrumentalists Dewey Bunnell, Dan Peek, and Gerry Beckley.', 'America', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(2, 'England is a country that is part of the United Kingdom. It shares land borders with Scotland to the north and Wales to the west.', 'England', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(3, 'Australia, officially the Commonwealth of Australia, is a country comprising the mainland of the Australian continent, the island of Tasmania, and numerous smaller islands.', 'Australia', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(4, 'Japan is an island country in East Asia.', 'Japan', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(5, 'Germany, officially the Federal Republic of Germany, is a sovereign state and federal parliamentary republic in central-western Europe.', 'Germany', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(6, 'France, is a sovereign state comprising territory in western Europe and several overseas regions and territories.', 'France', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(7, 'Italy officially the Italian Republic, is a unitary parliamentary republic in Europe.', 'Italy', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(8, 'India, officially the Republic of India, is a country in South Asia.', 'India', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(9, 'Brazil, officially the Federative Republic of Brazil, is the largest country in both South America and Latin America.', 'Brazil', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(10, 'Canada is a country in the northern half of North America.', 'Canada', '2010-1-1');
gaussdb=# INSERT INTO tsearch.pgweb VALUES(11, 'Mexico, officially the United Mexican States, is a federal republic in the southern part of North America.', 'Mexico', '2010-1-1');

gaussdb=# SELECT id, title, ts_rank_cd(to_tsvector(body), query) AS rank
FROM tsearch.pgweb, to_tsquery('america') query
WHERE query @@ to_tsvector(body)
ORDER BY rank DESC
LIMIT 10;
 id |  title  | rank
----+---------+------
 11 | Brazil  |  .2
  2 | America |  .1
 12 | Canada  |  .1
 13 | Mexico  |  .1
(4 rows)
```

使用标准化排序的相同例子：

```
gaussdb=# SELECT id, title, ts_rank_cd(to_tsvector(body), query, 32 /* rank/(rank+1) */ ) AS rank
FROM tsearch.pgweb, to_tsquery('america') query
WHERE  query @@ to_tsvector(body)
ORDER BY rank DESC
LIMIT 10;
 id |  title  |  rank
```


版权所有 © 华为云计算技术有限公司

```
----+---------+----------
 11 | Brazil  |  .166667
  2 | America | .0909091
 12 | Canada  | .0909091
 13 | Mexico  | .0909091
(4 rows)

-- 删除表
gaussdb=# DROP TABLE tsearch.pgweb;
gaussdb=# DROP SCHEMA tsearch;
```

排序要遍历每个匹配的tsvector，因此资源消耗多，可能会因为I/O限制导致排序慢。

1.12.3.4 高亮搜索结果

搜索结果的理想显示是：列出每篇文档中与搜索相关的部分，并标识为什么与查询相关。通常，搜索引擎显示文档片段时会对其中的搜索术语进行标记。GaussDB提供了函数ts_headline支持这部分功能。

```
ts_headline([ config regconfig, ] document text, query tsquery [, options text ]) returns text
```

ts_headline接受一个文档和一个查询，并且从该文档返回一个引用，其中来自查询的术语会被加亮。被用来解析该文档的配置可以用config指定；如果config被忽略，将会使用default_text_search_config配置。

options表示可选配置，由一个或多个option=value对组成，且必须用逗号分隔。options可以是下面的选项：

- StartSel，StopSel：分隔文档中出现的查询词，以区别于其他摘录词。当包含有空格或逗号时，必须用双引号将字符串引起来。内容最大长度为32767字节。
- MaxWords，MinWords：定义摘录的最长和最短值。
- ShortWord：在摘录的开始和结束会丢弃此长度或更短的词。默认值3会消除常见的英语冠词。
- HighlightAll：布尔标志。如果为真，整个文档将作为摘录。忽略前面三个参数的值。
- MaxFragments：要显示的文本摘录或片段的最大数量。默认值0表示选择非片段的摘录生成方法。 大于0的值表示选择基于片段的摘录生成。此方法查找带有尽可能多查询词的文本片段，并显示查询词周围的上下文片段。因此，查询词临近每个片段的中间，且查询词两边都有词。每个片段至多有MaxWords，并且长度为ShortWord或更短的词在每一个片段开始和结束被丢弃。如果在文档中没有找到所有的查询词，则文档中开头将显示MinWords单片段。
- FragmentDelimiter：当有一个以上的片段时，通过该字符串分隔这些片段。内容最大长度为32767字节。

不声明选项时，采用下面的缺省值：

```
StartSel=<b>, StopSel=</b>,
MaxWords=35, MinWords=15, ShortWord=3, HighlightAll=FALSE,
MaxFragments=0, FragmentDelimiter=" ... "
```

例如：

```
gaussdb=# SELECT ts_headline('english',
'The most common type of search
is to find all documents containing given query terms
and return them in order of their similarity to the
query.',
to_tsquery('english', 'query & similarity'));
                        ts_headline
------------------------------------------------------------
```


版权所有 © 华为云计算技术有限公司

```
 containing given <b>query</b> terms
 and return them in order of their <b>similarity</b> to the
 <b>query</b>.
(1 row)

gaussdb=# SELECT ts_headline('english',
'The most common type of search
is to find all documents containing given query terms
and return them in order of their similarity to the
query.',
to_tsquery('english', 'query & similarity'),
'StartSel = <, StopSel = >');
                      ts_headline
------------------------------------------------------------
 containing given <query> terms
 and return them in order of their <similarity> to the
 <query>.
(1 row)
```

ts_headline使用原始文档，而不是tsvector摘录，因此使用起来可能很慢，应慎重使用。

1.12.4 附加功能

1.12.4.1 处理 tsvector

GaussDB提供了用于操作tsvector类型的函数和操作符。

- tsvector || tsvector

  tsvector连接操作符返回一个新的tsvector类型，它综合了两个tsvector中词素和位置信息，并保留词素的位置信息和权重标签。右侧的tsvector的起始位置位于左侧tsvector的最后位置，因此，新生成的tsvector几乎等同于将两个原始文档字串连接后进行to_tsvector操作。（这个等价是不准确的，因为任何从左边tsvector中删除的停用词都不会影响结果，但是，在使用文本连接时，则会影响词素在右侧tsvector中的位置。）

  相较于对文本进行连接后再执行to_tsvector操作，使用tsvector类型进行连接操作的优势在于，可以对文档的不同部分使用不同配置进行解析。因为setweight函数会对给定的tsvector中的语素进行统一设置，如果想要对文档的不同部分设置不同的权重，需要在连接之前对文本进行解析和权重设置。

- setweight(vector tsvector, weight "char") returns tsvector

  setweight返回一个输入tsvector的副本，其中每一个位置都使用给定的权重做了标记。权值可以为A、B、C或D（D是tsvector副本的默认权重，并且不在输出中呈现）。当对tsvector进行连接操作时，这些权重标签将会被保留，文档不同部分以不同的权重进行排序。

  须知

  权重标签作用于位置，而不是词素。如果传入的tsvector已经被剥离了位置信息，那么setweight函数将无实际作用。

- length(vector tsvector) returns integer

  返回vector中的词素的数量。

- strip(vector tsvector) returns tsvector

版权所有 © 华为云计算技术有限公司

删除入参tsvector中所有词素的位置和权重信息，仅保留去重后的词素，返回新的tsvector。

1.12.4.2 处理查询

GaussDB提供了函数和操作符用来操作tsquery类型的查询。

- tsquery && tsquery

  返回两个给定查询tsquery的与结果。

- tsquery || tsquery

  返回两个给定查询tsquery的或结果。

- !! tsquery

  返回给定查询tsquery的非结果。

- tsquery <-> tsquery

  返回两个给定查询tsquery的词组结果。

- numnode(query tsquery) returns integer

  返回tsquery中的节点数目（词素加操作符），这个函数在检查查询是否有效（返回值大于0），或者只包含停用词（返回值等于0）时，是有用的。例如：

```
gaussdb=# SELECT numnode(plainto_tsquery('the any'));
NOTICE:  text-search query contains only stop words or doesn't contain lexemes, ignored
CONTEXT:  referenced column: numnode
 numnode
---------
       0

gaussdb=# SELECT numnode('foo & bar'::tsquery);
 numnode
---------
       3
```

- querytree(query tsquery) returns text

  返回可用于索引搜索的tsquery部分，该函数对于检测非索引查询是有用的（例如只包含停用词或否定项）。例如:

```
gaussdb=# SELECT querytree(to_tsquery('!defined'));
 querytree
-----------
 T
(1 row)
```

1.12.4.3 查询重写

ts_rewrite函数族可以从tsquery中搜索一个特定的目标子查询，并在该子查询每次出现的地方都替换为另一个子查询。 实际上这只是通过字串替换而得到的一个特定tsquery版本。目标子查询和替换查询组合起来可以被认为是一个重写规则。一组类似的重写规则可以为搜索提供强大的帮助。例如，可以使用同义词扩大搜索范围（例如，new york, big apple, nyc, gotham）或限制搜索范围。

- ts_rewrite (query tsquery, target tsquery, substitute tsquery) returns tsquery

  ts_rewrite的这种形式只适用于一个单一的重写规则：任何出现目标子查询的地方都被无条件替换。例如：

```
gaussdb=# SELECT ts_rewrite('a & b'::tsquery, 'a'::tsquery, 'c'::tsquery);
 ts_rewrite
------------
 'b' & 'c'
```


版权所有 © 华为云计算技术有限公司

- ts_rewrite (query tsquery, select text) returns tsquery

  ts_rewrite的这种形式接受一个起始查询和SQL查询命令。 这里的查询命令是文本字串形式，必须产生两个tsquery列。查询结果的每一行，第一个字段的值（目标子查询） 都会被第二个字段（替代子查询）替换。

  说明

  当多个规则需要重写时，重写顺序非常重要； 因此在实践中需要使用ORDER BY将源查询按照某些字段进行排序。

  例如：举一个现实生活中天文学上的例子。将使用表驱动的重写规则扩大supernovae的查询范围：

```
gaussdb=# CREATE TABLE tsearch.aliases (id int, t tsquery, s tsquery);

gaussdb=# INSERT INTO tsearch.aliases VALUES(1, to_tsquery('supernovae'), to_tsquery('supernovae|sn'));

gaussdb=# SELECT ts_rewrite(to_tsquery('supernovae & crab'), 'SELECT t, s FROM tsearch.aliases');
          ts_rewrite
---------------------------------
 'crab' & ( 'supernova' | 'sn' )
```

  可以通过更新表修改重写规则：

```
gaussdb=# UPDATE tsearch.aliases
SET s = to_tsquery('supernovae|sn & !nebulae')
WHERE t = to_tsquery('supernovae');

gaussdb=# SELECT ts_rewrite(to_tsquery('supernovae & crab'), 'SELECT t, s FROM tsearch.aliases');
                ts_rewrite
-------------------------------------------
 'crab' & ( 'supernova' | 'sn' & !'nebula' )
```

  需要重写的规则越多，重写操作就越慢。因为它要检查每一个可能匹配的规则。为了过滤明显的非候选规则，可以使用tsquery类型的操作符来实现。在下面的例子中， 只选择那些可能与原始查询匹配的规则：

```
gaussdb=# SELECT ts_rewrite('a & b'::tsquery, 'SELECT t,s FROM tsearch.aliases WHERE ''a & b''::tsquery @> t');
 ts_rewrite
------------
 'b' & 'a'
(1 row)
gaussdb=# DROP TABLE tsearch.aliases;
```

1.12.4.4 收集文献统计

函数ts_stat可用于检查配置和查找候选停用词。

```
ts_stat(sqlquery text, [ weights text, ]
      OUT word text, OUT ndoc integer,
      OUT nentry integer) returns setof record
```

sqlquery是一个包含SQL查询语句的文本，该SQL查询将返回一个tsvector。ts_stat执行SQL查询语句并返回一个包含tsvector中每一个不同的语素（词）的统计信息。返回信息包括：

- word text：词素。
- ndoc integer：词素在文档（tsvector）中的编号。
- nentry integer：词素出现的频率。

如果设置了权重条件，只有标记了对应权重的词素才会统计频率。例如，在一个文档集中检索使用频率最高的十个单词：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SET tsearch_options = 'debug_function';
gaussdb=# CREATE SCHEMA tpcds;
gaussdb=# CREATE TABLE tpcds.store_returns (
    sr_reason_sk      integer,
    sr_reason_id      character(16),
    sr_reason_desc    character(100),
    sr_customer_sk    integer);
gaussdb=# INSERT INTO tpcds.store_returns VALUES(1, 'test_001', 'test_desc_001', 5), (12, 'test_002',
'test_desc_002', 8), (23, 'test_003', 'test_desc_003', 9), (14, 'test_004', 'test_desc_004', 10), (45, 'test_005',
'test_desc_005', 12), (55, 'test_006', 'test_desc_006', 1), (43, 'test_007', 'test_desc_007', 12);
gaussdb=# SELECT * FROM ts_stat('SELECT to_tsvector(''english'', sr_reason_sk) FROM tpcds.store_returns
WHERE sr_customer_sk < 10') ORDER BY nentry DESC, ndoc DESC, word LIMIT 10;
 word | ndoc | nentry
------+------+--------
 1    |    1 |      1
 12   |    1 |      1
 23   |    1 |      1
 55   |    1 |      1
(4 rows)
```

同样的情况，但是只计算权重为A或者B的单词使用频率：

```
gaussdb=# SELECT * FROM ts_stat('SELECT to_tsvector(''english'', sr_reason_sk) FROM tpcds.store_returns
WHERE sr_customer_sk < 10', 'a') ORDER BY nentry DESC, ndoc DESC, word LIMIT 10;
 word | ndoc | nentry
------+------+--------
(0 rows)

-- 删除表
gaussdb=# DROP TABLE tpcds.store_returns;
gaussdb=# DROP SCHEMA tpcds;
gaussdb=# SET tsearch_options = '';
```

1.12.5 解析器

文本搜索解析器负责将原文档文本分解为多个token，并标识每个token的类型。这里的类型集由解析器本身定义。解析器并不修改文本，它只是确定合理的单词边界。由于此限制，人们更需要定制词典，而不是为每个应用程序定制解析器。

目前GaussDB提供了三个内置的解析器，分别为pg_catalog.default/pg_catalog.ngram/pg_catalog.pound，其中pg_catalog.default适用于英文分词场景，pg_catalog.ngram/pg_catalog.pound是为了支持中文全文检索功能新增的两种解析器，适用于中文及中英混合分词场景。

内置解析器pg_catalog.default，它能识别23种token类型，显示在**表1-337**中。

表 1-337 默认解析器的 token 类型

Token类型	描述	示例
asciiword	Word, all ASCII letters	elephant
word	Word, all letters	mañana
numword	Word, letters and digits	beta1
asciihword	Hyphenated word, all ASCII	up-to-date
hword	Hyphenated word, all letters	lógico-matemática
numhword	Hyphenated word, letters and digits	test-beta1

版权所有 © 华为云计算技术有限公司

Token类型	描述	示例
hword_asciipart	Hyphenated word part, all ASCII	test in the context test-beta1
hword_part	Hyphenated word part, all letters	lógico or matemática in the context lógico-matemática
hword_numpart	Hyphenated word part, letters and digits	beta1 in the context test-beta1
email	Email address	foo@example.com
protocol	Protocol head	http://
url	URL	example.com/stuff/index.html
host	Host	example.com
url_path	URL path	/stuff/index.html, in the context of a URL
file	File or path name	/usr/local/foo.txt, if not within a URL
sfloat	Scientific notation	-1.23E+56
float	Decimal notation	-1.234
int	Signed integer	-1234
uint	Unsigned integer	1234
version	Version number	8.3.0
tag	XML tag	<a href="dictionaries.html">
entity	XML entity	&
blank	Space symbols	(any whitespace or punctuation not otherwise recognized)

注意：对于解析器，一个“字母”的概念是由数据库的语言区域设置，即lc_ctype设置决定的。只包含基本ASCII字母的词被报告为一个单独的token类型，因为这类词有时需要被区分出来。大多数欧洲语言中，对token类型word和asciiword的处理方法是类似的。

email不支持某些由RFC 5322定义的有效电子邮件字符。可用于email用户名的非字母数字字符仅包含句号、破折号和下划线。

解析器可能对同一内容进行重叠token。例如，包含连字符的单词将作为一个整体进行报告，其组件也会分别被报告：

```
gaussdb=# SELECT alias, description, token FROM ts_debug('english','foo-bar-beta1');
     alias      |              description              |     token
----------------+---------------------------------------+---------------
 numhword       | Hyphenated word, letters and digits   | foo-bar-beta1
```


版权所有 © 华为云计算技术有限公司

```
hword_asciipart | Hyphenated word part, all ASCII         | foo
blank           | Space symbols                          | -
hword_asciipart | Hyphenated word part, all ASCII         | bar
blank           | Space symbols                          | -
hword_numpart   | Hyphenated word part, letters and digits | beta1
```

这种行为是有必要的，因为它支持搜索整个复合词和各组件。这里是另一个例子：

```
gaussdb=# SELECT alias, description, token FROM ts_debug('english','http://example.com/stuff/index.html');
  alias   |  description  |            token
----------+---------------+------------------------------
 protocol | Protocol head | http://
 url      | URL           | example.com/stuff/index.html
 host     | Host          | example.com
 url_path | URL path      | /stuff/index.html
```

N-gram是一种机械分词方法，适用于无语义中文分词场景。N-gram分词法可以保证分词的完备性，但是为了照顾所有可能，把很多不必要的词也加入到索引中，导致索引项增加。N-gram支持中文编码包括GBK、UTF-8。内置6种token类型，如**表1-338**所示。

表 1-338 token 类型

别名	描述
zh_words	chinese words
en_word	english word
numeric	numeric data
alnum	alnum string
grapsymbol	graphic symbol
multisymbol	multiple symbol

Pound是一种固定格式分词方法，适用于无语意但待解析文本以固定分隔符分割开来的中英文分词场景。支持中文编码包括GBK、UTF8，支持英文编码包括ASCII。内置6种token类型，如**表1-339**所示；支持5种分隔符，如**表1-340**所示，在用户不进行自定义设置的情况下分隔符默认为“#”。Pound限制单个token长度不能超过256个字符。

表 1-339 token 类型

别名	描述
zh_words	chinese words
en_word	english word
numeric	numeric data
alnum	alnum string
grapsymbol	graphic symbol
multisymbol	multiple symbol

版权所有 © 华为云计算技术有限公司

表 1-340 分隔符类型

分隔符	描述
@	Special character
#	Special character
$	Special character
%	Special character
/	Special character

1.12.6 词典

1.12.6.1 词典概述

词典用于定义停用词（stop words），即全文检索时不搜索哪些词。

词典还可以用于对同一词的不同形式进行规范化，这样同一个词的不同派生形式都可以进行匹配。规范化后的词称为词位（lexeme）。

除了提高检索质量外，词的规范化和删除停用词可以减少文档tsvector格式的大小，从而提高性能。词的规范化和删除停用词并不总是具有语言学意义，用户可以根据应用环境在词典定义文件中自定义规范化和删除规则。

一个词典是一个程序，接收标记（token）作为输入，并返回：

- 如果token在词典中已知，返回对应lexeme数组（注意，一个标记可能对应多个lexeme）。
- 一个lexeme。（该lexeme为设置了TSL_FILTER标志的lexeme，TSL_FILTER标志为过滤词典内部自动设置，用户不感知。）一个新token会代替输入token被传递给后继词典（当前词典可被称为过滤词典）。
- 如果token在词典中已知，但它是一个停用词，返回空数组。
- 如果词典不能识别输入的token，返回NULL。

GaussDB提供了多种语言的预定义字典，同时提供了以下预定义的词典模板，分别是Simple和Snowball，可用于创建自定义参数的新词典。

在使用全文检索时，建议用户：

- 可以在文本搜索配置中定义一个解析器，以及一组用于处理该解析器的输出标记词典。对于解析器返回的每个标记类型，可以在配置中指定不同的词典列表进行处理。当解析器输出一种类型的标记后，在对应列表的每个字典中会查阅该标记，直到某个词典识别它。如果它被识别为一个停用词， 或者没有任何词典识别，该token将被丢弃，即不被索引或检索到。通常情况下，第一个返回非空结果的词典决定了最终结果，后继词典将不会继续处理。但是一个过滤类型的词典可以依据规则替换输入token，然后将替换后的token传递给后继词典进行处理。
- 配置字典列表的一般规则是，第一个位置放置一个应用范围最小的、最具体化定义的词典，其次是更一般化定义的词典， 最后是一个普适定义的词典，比如Snowball词干词典或Simple词典。

版权所有 © 华为云计算技术有限公司

1.12.6.2 停用词

停用词是很常见的词，几乎出现在每一个文档中，并且没有区分值。因此，在全文搜索的语境下可忽视它们。停用词处理逻辑和词典类型相关。例如，Ispell词典会先对标记进行规范化，然后再查看停用词表，而Snowball词典会最先检查输入标记是否为停用词。

每个英文文本包含像a和the的单词，因此没必要将它们存储在索引中。停用词影响tsvector中的位置，同时位置也会影响相关度：

```
gaussdb=# SELECT to_tsvector('english','in the list of stop words');
        to_tsvector
----------------------------
 'list':3 'stop':5 'word':6
```

位置1、2、4是停用词，所以不显示。为包含和不包含停用词的文档计算出的排序是完全不同的：

```
gaussdb=# SELECT ts_rank_cd (to_tsvector('english','in the list of stop words'), to_tsquery('list & stop'));
 ts_rank_cd
------------
        .05

gaussdb=# SELECT ts_rank_cd (to_tsvector('english','list stop words'), to_tsquery('list & stop'));
 ts_rank_cd
------------
         .1
```

1.12.6.3 Simple 词典

Simple词典首先将输入标记转换为小写字母，然后检查停用词表。如果识别为停用词则返回空数组，即表示该标记会被丢弃。否则，输入标记的小写形式作为规范化后的lexeme返回。此外，Simple词典可通过设置参数Accept为false（默认值true），将非停用词报告为未识别，传递给后继词典继续处理。

注意事项

- 大多数词典的功能依赖于词典定义文件，词典定义文件名仅支持小写字母、数字、下划线组合。
- 临时模式pg_temp下不允许创建词典。
- 词典定义文件的字符集编码必须为UTF-8格式。实际应用时，如果与数据库的字符编码格式不一致，在读入词典定义文件时会进行编码转换。
- 通常情况下，每个session仅读取词典定义文件一次，当且仅当在第一次使用该词典时。需要修改词典文件时，可通过ALTER TEXT SEARCH DICTIONARY命令进行词典定义文件的更新和重新加载。

操作步骤

步骤1 创建Simple词典。

```
gaussdb=# CREATE TEXT SEARCH DICTIONARY public.simple_dict (
    TEMPLATE = pg_catalog.simple,
    STOPWORDS = english
);
```

其中，停用词表文件全名为english.stop。关于创建simple词典的语法和更多参数，请参见**CREATE TEXT SEARCH DICTIONARY**。

步骤2 使用Simple词典。

版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT ts_lexize('public.simple_dict','YeS');
 ts_lexize
-----------
 {yes}
(1 row)

gaussdb=# SELECT ts_lexize('public.simple_dict','The');
 ts_lexize
-----------
 {}
(1 row)
```

步骤3 设置参数ACCEPT=false，使Simple词典返回NULL，而不是返回非停用词的小写形式。

```
gaussdb=# ALTER TEXT SEARCH DICTIONARY public.simple_dict ( Accept = false );
ALTER TEXT SEARCH DICTIONARY
gaussdb=# SELECT ts_lexize('public.simple_dict','YeS');
 ts_lexize
-----------

(1 row)

gaussdb=# SELECT ts_lexize('public.simple_dict','The');
 ts_lexize
-----------
 {}
(1 row)

gaussdb=# DROP TEXT SEARCH DICTIONARY public.simple_dict;
```

----结束

1.12.6.4 Snowball 词典

Snowball词典模板支持词干分析词典，基于Martin Porter的Snowball项目，内置有许多语言的词干分析算法。GaussDB中预定义有多种语言的Snowball词典，可通过系统表PG_TS_DICT查看预定义的词干分析词典以及支持的语言词干分析算法。

无论是否可以简化，Snowball词典将标示所有输入为已识别，因此它应当被放置在词典列表的最后。把Snowball词典放在任何其他词典前面会导致后继词典失效，因为输入token不会通过Snowball词典进入到下一个词典。

关于Snowball词典的语法，请参见**CREATE TEXT SEARCH DICTIONARY**。

1.12.7 配置示例

文本搜索配置（Text Search Configuration），指定了将文档转换成tsvector过程中所必需的组件：

- 解析器，用于把文本分解成标记token。
- 词典列表，用于将每个token转换成词位lexeme。

每次to_tsvector或to_tsquery函数调用时，都需要指定一个文本搜索配置来指定具体的处理过程。GUC参数default_text_search_config指定了默认的文本搜索配置，当文本搜索函数中没有显式指定文本搜索配置参数时，将会使用该默认值进行处理。

GaussDB中预定义有一些可用的文本搜索配置，用户也可创建自定义的文本搜索配置。此外，为了便于管理文本搜索对象，还提供有多个gsql元命令，可以显示有关文本搜索对象的信息（详细请参见**元命令参考**）。

版权所有 © 华为云计算技术有限公司

操作步骤

步骤1 可以设置当前session使用的默认的文本搜索配置，当前全文检索配置支持：ngram、pound、simple、english分词配置。在当前session有效。

```
gaussdb=# set default_text_search_config = 'ngram';
SET
```

----结束

1.12.8 测试和调试文本搜索

须知

该功能为内部使用功能，不建议用户使用。

自定义文本搜索分词器的行为很容易变得混乱，本节中描述的函数用于对文本搜索对象进行测试。可以测试分词器整体，也可以单独测试解析器和词典。

1.12.8.1 分词器测试

函数ts_debug允许简单测试文本搜索分词器。

```
ts_debug([ config regconfig, ] document text,
        OUT alias text,
        OUT description text,
        OUT token text,
        OUT dictionaries regdictionary[],
        OUT dictionary regdictionary,
        OUT lexemes text[])
        returns setof record
```

ts_debug显示document的每个token信息，token是由解析器生成，由指定的词典进行处理。如果忽略对应参数，则使用config指定的分词器或者default_text_search_config指定的分词器。

ts_debug为文本解析器标识的每个token返回一行记录。记录中的列分别是：

- alias：text类型，token的别名。
- description：text类型，token的描述。
- token：text类型，token的文本内容。
- dictionaries：regdictionary数组类型，是分词器为token选定的词典。
- dictionary：regdictionary类型，用来识别token的词典。如果为空，则不做识别。
- lexemes：text数组类型，词典识别token时生成的词素。如果为空，则不生成词素。空数组（{}）意味着token将被识别成停用词。

一个简单的例子：

```
gaussdb=# SELECT * FROM ts_debug('english','a fat  cat sat on a mat - it ate a fat rats');
   alias   |   description   | token |  dictionaries  |  dictionary  | lexemes
-----------+-----------------+-------+----------------+--------------+---------
 asciiword | Word, all ASCII | a     | {english_stem} | english_stem | {}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | fat   | {english_stem} | english_stem | {fat}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | cat   | {english_stem} | english_stem | {cat}
```


版权所有 © 华为云计算技术有限公司

```
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | sat   | {english_stem} | english_stem | {sat}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | on    | {english_stem} | english_stem | {}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | a     | {english_stem} | english_stem | {}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | mat   | {english_stem} | english_stem | {mat}
 blank     | Space symbols   |       | {}             |              |
 blank     | Space symbols   | -     | {}             |              |
 asciiword | Word, all ASCII | it    | {english_stem} | english_stem | {}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | ate   | {english_stem} | english_stem | {ate}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | a     | {english_stem} | english_stem | {}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | fat   | {english_stem} | english_stem | {fat}
 blank     | Space symbols   |       | {}             |              |
 asciiword | Word, all ASCII | rats  | {english_stem} | english_stem | {rat}
(24 rows)
```

1.12.8.2 解析器测试

函数ts_parse可以直接测试文本搜索解析器。

```
ts_parse(parser_name text, document text,
        OUT tokid integer, OUT token text) returns setof record
```

ts_parse解析指定的document并返回一系列的记录，一条记录代表一个解析生成的token。每条记录包括标识token类型的tokid，及token文本。例如：

```
gaussdb=# SELECT * FROM ts_parse('default', '123 - a number');
 tokid | token
-------+--------
    22 | 123
    12 |
    12 | -
     1 | a
    12 |
     1 | number
(6 rows)
```

函数ts_token_type返回指定解析器的token类型及其描述信息。

```
ts_token_type(parser_name text, OUT tokid integer,
             OUT alias text, OUT description text) returns setof record
```

ts_token_type返回一个表，这个表描述了指定解析器可以识别的每种token类型。对于每个token类型，表中给出了整数类型的tokid--用于解析器标记对应的token类型；alias——命名分词器命令中的token类型及简单描述。比如：

```
gaussdb=# SELECT * FROM ts_token_type('default');
 tokid |      alias      |                description
-------+-----------------+-------------------------------------------
     1 | asciiword       | Word, all ASCII
     2 | word            | Word, all letters
     3 | numword         | Word, letters and digits
     4 | email           | Email address
     5 | url             | URL
     6 | host            | Host
     7 | sfloat          | Scientific notation
     8 | version         | Version number
     9 | hword_numpart   | Hyphenated word part, letters and digits
    10 | hword_part      | Hyphenated word part, all letters
    11 | hword_asciipart | Hyphenated word part, all ASCII
    12 | blank           | Space symbols
    13 | tag             | XML tag
    14 | protocol        | Protocol head
```


版权所有 © 华为云计算技术有限公司

```
   15 | numhword        | Hyphenated word, letters and digits
   16 | asciihword      | Hyphenated word, all ASCII
   17 | hword           | Hyphenated word, all letters
   18 | url_path        | URL path
   19 | file            | File or path name
   20 | float           | Decimal notation
   21 | int             | Signed integer
   22 | uint            | Unsigned integer
   23 | entity          | XML entity
(23 rows)
```

1.12.8.3 词典测试

函数ts_lexize用于进行词典测试。

ts_lexize(dict regdictionary, token text) returns text[]如果输入的token可以被词典识别，那么ts_lexize返回词素的数组；如果token可以被词典识别到它是一个停用词，则返回空数组；如果是一个不可识别的词则返回NULL。

示例如下：

```
gaussdb=# SELECT ts_lexize('english_stem', 'stars');
 ts_lexize
-----------
 {star}

gaussdb=# SELECT ts_lexize('english_stem', 'a');
 ts_lexize
-----------
 {}
```

须知

ts_lexize函数支持单一token，不支持文本。

1.12.9 限制约束

GaussDB的全文检索功能当前限制约束是：

- tsvector每个分词的字符长度必须小于2047字节。
- tsvector结构（分词+位置）的长度必须小于1MB。
- tsvector的位置值必须大于0，且小于等于16,383。大于16383的位置值将会被强制置为16383。
- tsvector每个分词在文档中位置数必须小于等于255，若超过将舍弃后面的位置信息。
- tsquery中的关键字及对应运算符最大支持到32768。
- 全文检索的准确度与文本搜索配置中的解析器和词典列表相关，中文场景建议使用ngram/pound作为文本搜索配置项，其他场景建议使用english/simple作为文本搜索配置项。
- 仅支持在**行表达式函数白名单**中的tsvector/tsquery相关函数为数据对象增加或修改ILM策略。
- Tokenweight词典功能为BM25索引内部使用，不建议用户在全文检索特性中使用，可能包含未知风险，请使用其他词典功能代替。
- 词组操作符（PHRASE）指定的距离必须大于等于0，且小于等于16384，否则报错。

版权所有 © 华为云计算技术有限公司

1.13 SQL 语法

1.13.1 SQL 语法格式说明

表 1-341 SQL 语法格式说明

格式	意义
[ ]	表示用“[ ]”括起来的部分是可选的。
...	表示前面的元素可重复出现。
[x | y | ...]	表示从两个或多个选项中选取一个或者不选。
{ x | y | ... }	表示从两个或多个选项中选取一个。
[x | y | ...] [...]	表示可选多个参数或者不选，如果选择多个参数，则参数之间用空格分隔。
[x | y | ...] [,...]	表示可选多个参数或者不选，如果选择多个参数，则参数之间用逗号分隔。
{ x | y | ... } [...]	表示可选多个参数，至少选一个，如果选择多个参数，则参数之间以空格分隔。
{ x | y | ... } [,...]	表示可选多个参数，至少选一个，如果选择多个参数，则参数之间用逗号分隔。

1.13.2 DCL 语法一览表

DCL（Data Control Language，数据控制语言），用于定义和控制用户对数据库对象（如表、视图、存储过程等）以及数据库实例本身的访问权限。

授权

GaussDB提供了针对数据对象和角色授权的语句，请参见**GRANT**。

收回权限

GaussDB提供了收回权限的语句，请参见**REVOKE**。

设置默认权限

GaussDB允许设置应用于将来创建的对象的权限，请参见**ALTER DEFAULT PRIVILEGES**。

修改属主

GaussDB提供了修改数据库对象属主的语句，请参见**REASSIGN OWNED**。

版权所有 © 华为云计算技术有限公司

定义角色

角色是数据库中用于管理权限的机制，从数据库安全的角度出发，应将各类管理和操作权限分配给不同的角色。所涉及的SQL语句，如表1-342所示。

表 1-342 角色定义相关 SQL

功能	相关SQL
创建角色	CREATE ROLE
修改角色属性	ALTER ROLE
删除角色	DROP ROLE

定义用户

在数据库管理中，用户是访问数据库系统的身份凭证。通过为用户授予权限，可以精确控制其对数据库对象以及数据库实例本身的访问权限，从而实现对数据库访问的精细化管理。所涉及的SQL语句，如表1-343所示。

表 1-343 用户定义相关 SQL

功能	相关SQL
创建用户	CREATE USER
修改用户属性	ALTER USER
删除用户	DROP USER

修改/显示/恢复运行参数

表 1-344 修改/显示/恢复运行参数相关 SQL

功能	相关SQL
修改运行时配置参数	SET
修改会话	ALTER SESSION
显示当前运行时参数的数值	SHOW
将指定的运行参数恢复为默认值	RESET

1.13.3 DDL 语法一览表

DDL（Data Definition Language，数据定义语言），用于定义和修改数据库对象的标准SQL语言部分，主要包括CREATE、ALTER和DROP等语句，用于创建、修改和删除数据库中的对象，例如：表、索引、视图、约束、数据类型定义等。

版权所有 © 华为云计算技术有限公司

> **说明**
>
> GaussDB不支持CN不完整时进行DDL操作。例如：集群中有1个CN故障时执行新建数据库、表等操作都会失败。

删除数据库角色所拥有的数据库对象

GaussDB提供了删除一个数据库角色所拥有的数据库对象的语句，请参见**DROP OWNED**。

定义客户端加密主密钥

客户端加密主密钥主要用于密态数据库特性中，进行加密列加密密钥（CEK）。客户端加密主密钥定义主要包括：创建客户端加密主密钥以及删除客户端加密主密钥。所涉及的SQL语句，如**表1-345**所示。

表 1-345 客户端加密主密钥定义相关 SQL

功能	相关SQL
创建客户端加密主密钥	**CREATE CLIENT MASTER KEY**
删除客户端加密主密钥	**DROP CLIENT MASTER KEY**

定义列加密密钥

列加密密钥主要用于密态数据库特性中，进行加密数据。列加密密钥定义主要包括：创建列加密密钥、轮转加密列加密密钥的客户端主密钥以及删除列加密密钥。所涉及的SQL语句，如**表1-346**所示。

表 1-346 列加密密钥定义相关 SQL

功能	相关SQL
创建列加密密钥	**CREATE COLUMN ENCRYPTION KEY**
修改列加密密钥指定的客户端主密钥	**ALTER COLUMN ENCRYPTION KEY**
删除列加密密钥	**DROP COLUMN ENCRYPTION KEY**

定义数据库

数据库是组织、存储和管理数据的仓库，而数据库定义主要包括：创建数据库、修改数据库属性以及删除数据库。所涉及的SQL语句，如**表1-347**所示。

表 1-347 数据库定义相关 SQL

功能	相关SQL
创建数据库	**CREATE DATABASE**

版权所有 © 华为云计算技术有限公司

功能	相关SQL
修改数据库属性	ALTER DATABASE
删除数据库	DROP DATABASE

定义模式

模式是一组数据库对象的集合，主要用于控制对数据库对象的访问。所涉及的SQL语句，如表1-348所示。

表 1-348 模式定义相关 SQL

功能	相关SQL
创建模式	CREATE SCHEMA
修改模式属性	ALTER SCHEMA
删除模式	DROP SCHEMA

定义表空间

表空间用于管理数据对象，与磁盘上的一个目录对应。所涉及的SQL语句，如表1-349所示。

表 1-349 表空间定义相关 SQL

功能	相关SQL
创建表空间	CREATE TABLESPACE
修改表空间属性	ALTER TABLESPACE
删除表空间	DROP TABLESPACE

定义表

表是数据库中的一种特殊数据结构，用于存储数据对象以及对象之间的关系。所涉及的SQL语句，如表1-350所示。

表 1-350 表定义相关 SQL

功能	相关SQL
创建表	CREATE TABLE
修改表属性	ALTER TABLE

版权所有 © 华为云计算技术有限公司

功能	相关SQL
修改一个或多个表名称	RENAME TABLE
删除表	DROP TABLE
根据查询结果创建表	CREATE TABLE AS

定义分区表

分区表是一种逻辑表，而数据由普通表进行存储，分区表主要用于提升查询性能。所涉及的SQL语句，如表1-351所示。

表 1-351 分区表定义相关 SQL

功能	相关SQL
创建分区表	CREATE TABLE PARTITION
修改分区表（包括增加/删除分区以及修改分区属性等）	ALTER TABLE PARTITION
删除分区表	DROP TABLE
根据查询结果创建分区表	CREATE TABLE PARTITION | SUBPARTITION AS

定义索引

索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息。所涉及的SQL语句，如表1-352所示。

表 1-352 索引定义相关 SQL

功能	相关SQL
创建索引	CREATE INDEX
修改索引属性	ALTER INDEX
删除索引	DROP INDEX
重建索引	REINDEX
在指定的表上创建全局二级索引	CREATE GLOBAL INDEX

定义存储过程

存储过程是一组为了完成特定功能的SQL语句集，经过编译后存储在数据库中，用户通过指定存储过程的名称以及指定具体参数（如果该存储过程带有参数）进行执行。所涉及的SQL语句，如表1-353所示。

版权所有 © 华为云计算技术有限公司

表 1-353 存储过程定义相关 SQL

功能	相关SQL
创建存储过程	CREATE PROCEDURE
删除存储过程	DROP PROCEDURE
修改自定义存储过程的属性或重编译存储过程	ALTER PROCEDURE

定义函数

在GaussDB中，函数和存储过程类似，也是一组SQL语句集，在使用时并无差别。所涉及的SQL语句，如表1-354所示。

表 1-354 函数定义相关 SQL

功能	相关SQL
创建函数	CREATE FUNCTION
修改函数属性或重编译	ALTER FUNCTION
删除函数	DROP FUNCTION

定义包

包由包头（package specification）和包体（package body）组成，用于分类管理存储过程和函数，类似于Java、C++等语言中的类。所涉及的SQL语句，如表1-355所示。

表 1-355 包定义相关 SQL

功能	相关SQL
创建包	CREATE PACKAGE
删除包	DROP PACKAGE
修改包属性	ALTER PACKAGE

说明

当前形态下，使用PACKAGE语法，需要在ORA兼容模式数据库中执行。

定义资源池

资源池是负载管理模块使用的系统表，主要用于指定资源管理相关的属性，如：控制组。所涉及的SQL语句，如表1-356所示。

版权所有 © 华为云计算技术有限公司

表 1-356 资源池定义相关 SQL

功能	相关SQL
创建资源池	CREATE RESOURCE POOL
修改资源池属性	ALTER RESOURCE POOL
删除资源池	DROP RESOURCE POOL

定义聚合函数

表 1-357 聚合函数定义相关 SQL

功能	相关SQL
创建一个新的聚合函数	CREATE AGGREGATE
修改聚合函数	ALTER AGGREGATE
删除聚合函数	DROP AGGREGATE

定义插件扩展

该特性为内部使用，只允许操作数据库内部的高级包或扩展。

表 1-358 插件扩展定义相关 SQL

功能	相关SQL
创建（安装）一个新的扩展	CREATE EXTENSION
从数据库中删除一个扩展	DROP EXTENSION

定义数据类型

表 1-359 数据类型定义相关 SQL

功能	相关SQL
创建一个新的数据类型	CREATE TYPE
修改数据类型	ALTER TYPE
删除数据类型	DROP TYPE

版权所有 © 华为云计算技术有限公司

定义 DATABASE LINK 对象

DATABASE LINK是一个可以操作远程数据库的对象，所涉及的SQL语句，如**表1-360**所示。

表 1-360 DATABASE LINK 对象相关 SQL

功能	相关SQL
创建一个新的DATABASE LINK对象	**CREATE DATABASE LINK**
修改DATABASE LINK对象	**ALTER DATABASE LINK**
删除DATABASE LINK对象	**DROP DATABASE LINK**

定义审计策略

表 1-361 审计策略定义相关 SQL

功能	相关SQL
创建统一审计策略	**CREATE AUDIT POLICY**
修改统一审计策略内容	**ALTER AUDIT POLICY**
删除一个审计策略	**DROP AUDIT POLICY**

修改指定节点的 nodeis_active 字段值

修改pgxc_node系统表中指定节点的nodeis_active字段值，请参见**ALTER COORDINATOR**。

定义目录对象

表 1-362 目录对象定义相关 SQL

功能	相关SQL
创建一个目录对象	**CREATE DIRECTORY**
修改对目录对象属性	**ALTER DIRECTORY**
删除指定的目录对象	**DROP DIRECTORY**

版权所有 © 华为云计算技术有限公司

定义外表

表 1-363 外表定义相关 SQL

功能	相关SQL
创建一个外表	CREATE FOREIGN TABLE
修改外表属性	ALTER FOREIGN TABLE
删除指定的外表	DROP FOREIGN TABLE

定义外部数据封装器

表 1-364 外部数据封装器相关 SQL

功能	相关SQL
创建外部数据封装器	CREATE FOREIGN DATA WRAPPER
修改外部数据封装器属性	ALTER FOREIGN DATA WRAPPER
删除外部数据封装器	DROP FOREIGN DATA WRAPPER

gs_global_config 系统表相关 SQL

表 1-365 gs_global_config 系统表相关 SQL

功能	相关SQL
新增、修改系统表的参数值	ALTER GLOBAL CONFIGURATION
删除系统表中的参数值	DROP GLOBAL CONFIGURATION
向系统表中插入一个或者多个弱口令	CREATE WEAK PASSWORD DICTIONARY
清空系统表中的所有弱口令	DROP WEAK PASSWORD DICTIONARY

定义用户组

表 1-366 用户组定义相关 SQL

功能	相关SQL
创建一个新用户组	CREATE GROUP
修改用户组的属性	ALTER GROUP

版权所有 © 华为云计算技术有限公司

功能	相关SQL
删除用户组	DROP GROUP

定义过程语言

表 1-367 过程语言定义相关 SQL

功能	相关SQL
创建一种新的过程语言	CREATE LANGUAGE
修改过程语言的属性	ALTER LANGUAGE
删除一种过程语言	DROP LANGUAGE

定义脱敏策略

表 1-368 脱敏策略定义相关 SQL

功能	相关SQL
创建脱敏策略	CREATE MASKING POLICY
修改脱敏策略内容	ALTER MASKING POLICY
删除脱敏策略	DROP MASKING POLICY

定义物化视图

表 1-369 物化视图定义相关 SQL

功能	相关SQL
创建一个全量物化视图	CREATE MATERIALIZED VIEW
创建一个增量物化视图	CREATE INCREMENTAL MATERIALIZED VIEW
修改现有物化视图的多个辅助属性	ALTER MATERIALIZED VIEW
强制删除数据库中已有的物化视图	DROP MATERIALIZED VIEW
对物化视图以全量刷新的方式进行刷新	REFRESH MATERIALIZED VIEW
对物化视图以增量刷新的方式进行刷新	REFRESH INCREMENTAL MATERIALIZED VIEW

版权所有 © 华为云计算技术有限公司

定义节点

表 1-370 节点定义相关 SQL

功能	相关SQL
创建一个新的节点	CREATE NODE
修改现有节点的属性	ALTER NODE
删除节点	DROP NODE

定义资源标签

表 1-371 资源标签定义相关 SQL

功能	相关SQL
创建资源标签	CREATE RESOURCE LABEL
修改资源标签属性	ALTER RESOURCE LABEL
删除资源标签	DROP RESOURCE LABEL

定义行访问控制策略

表 1-372 行访问控制策略定义相关 SQL

功能	相关SQL
对表创建行访问控制策略	CREATE ROW LEVEL SECURITY POLICY
对已存在的行访问控制策略进行修改	ALTER ROW LEVEL SECURITY POLICY
删除表上指定的行访问控制策略	DROP ROW LEVEL SECURITY POLICY

定义序列

表 1-373 序列定义相关 SQL

功能	相关SQL
向当前数据库增加一个新的序列	CREATE SEQUENCE
修改指定序列的参数	ALTER SEQUENCE
从当前数据库中删除序列	DROP SEQUENCE

版权所有 © 华为云计算技术有限公司

定义外部服务器

表 1-374 外部服务器定义相关 SQL

功能	相关SQL
创建一个新的外部服务器	**CREATE SERVER**
增加、修改和删除指定服务器的参数	**ALTER SERVER**
删除指定服务器	**DROP SERVER**

定义同义词对象

表 1-375 同义词对象定义相关 SQL

功能	相关SQL
创建一个同义词对象	**CREATE SYNONYM**
修改同义词对象的所有者	**ALTER SYNONYM**
删除指定的同义词对象	**DROP SYNONYM**

定义触发器

表 1-376 触发器定义相关 SQL

功能	相关SQL
创建一个触发器	**CREATE TRIGGER**
修改触发器属性	**ALTER TRIGGER**
删除触发器	**DROP TRIGGER**

定义视图

表 1-377 视图定义相关 SQL

功能	相关SQL
创建一个视图	**CREATE VIEW**
更改视图的各种辅助属性	**ALTER VIEW**
强制删除数据库中已有的视图	**DROP VIEW**

版权所有 © 华为云计算技术有限公司

收集统计信息

收集与数据库中普通表内容相关的统计信息，请参见**ANALYZE | ANALYSE**。

创建同步点

创建一个新集群节点间的同步点，请参见**CREATE BARRIER**。

创建编码转换

创建一种两个字符集编码之间的新转换关系，请参见**CREATE CONVERSION**。

定义模型

表 1-378 模型定义相关 SQL

功能	相关SQL
训练机器学习模型并保存模型	**CREATE MODEL**
删除一个已训练完成保存的模型对象	**DROP MODEL**

定义安全标签

表 1-379 安全标签定义相关 SQL

功能	相关SQL
创建安全标签	**CREATE SECURITY LABEL**
应用更新或取消安全标签	**SECURITY LABEL ON**
删除安全标签	**DROP SECURITY LABEL**

导入 DATABASE/TABLE

表 1-380 导入准备相关 SQL

功能	相关SQL
导入数据库的准备阶段	**IMPDP DATABASE CREATE**
导入数据库的执行阶段	**IMPDP RECOVER**
导入表的准备阶段	**IMPDP TABLE PREPARE**
导入表的执行阶段	**IMPDP TABLE**

版权所有 © 华为云计算技术有限公司

导出文件

表 1-381 导出文件相关 SQL

功能	相关SQL
导出数据库的全部物理文件	**EXPDP DATABASE**
导出与表相关的全部文件	**EXPDP TABLE**

全文检索

表 1-382 全文检索相关 SQL

功能	相关SQL
创建一个新的文本搜索配置	**CREATE TEXT SEARCH CONFIGURATION**
创建一个新的全文检索词典	**CREATE TEXT SEARCH DICTIONARY**
更改文本搜索配置的相关属性	**ALTER TEXT SEARCH CONFIGURATION**
修改全文检索词典的相关属性	**ALTER TEXT SEARCH DICTIONARY**
删除已有文本搜索配置	**DROP TEXT SEARCH CONFIGURATION**
删除全文检索词典	**DROP TEXT SEARCH DICTIONARY**

清理回收站

GaussDB提供了清理回收站的语句，请参见**PURGE**。

对表进行聚簇排序

GaussDB提供了根据指定索引对表进行聚簇排序的语句，请参见**CLUSTER**。

定义一个对象的注释

GaussDB提供了创建、修改指定对象的注释的语句，请参见**COMMENT**。

根据查询结果创建新表

GaussDB支持用于根据查询结果创建一个新表，并且将查询到的数据插入到新表的语句，请参见**SELECT INTO**。

版权所有 © 华为云计算技术有限公司

将表恢复到一个早期状态

GaussDB支持在人为操作后或应用程序错误时，将表恢复到一个早期状态的语句，请参见**TIMECAPSULE TABLE**。

清理表数据

GaussDB提供了快速地从表中删除所有行的语句，请参见**TRUNCATE**。

回收存储空间

GaussDB支持回收表或B-Tree索引中已经删除的行所占用的存储空间的语句，请参见**VACUUM**。

1.13.4 DML 语法一览表

DML（Data Manipulation Language，数据操作语言），用于对数据库表中的数据进行操作。如：插入、更新、查询以及删除等。

插入数据

表 1-383 插入数据相关 SQL

功能	相关SQL
向数据库表中添加一条或多条记录	**INSERT**
向多个表添加一条或多条记录	**INSERT ALL**

修改数据

修改数据库表中的一条或多条记录，请参见**UPDATE**。

修改或者插入数据

GaussDB提供了将目标表和源表中数据根据关联条件进行匹配的语句，若关联条件匹配时对目标表进行UPDATE操作，无法匹配时对目标表执行INSERT操作，请参见**MERGE INTO**。

查询数据

数据库查询语句（SELECT）是用于在数据库中检索适合条件信息的语句，请参见**SELECT**。

删除数据

GaussDB提供了删除表中指定条件的数据的语句，请参见**DELETE**。

复制数据

GaussDB提供了在表和文件之间复制数据的语句，请参见**COPY**。

版权所有 © 华为云计算技术有限公司

锁

表 1-384 锁相关 SQL

功能	相关SQL
多种锁模式（用于控制对表中数据的并发访问）	**LOCK**
bucket粒度的锁	**LOCK BUCKETS**

调用函数

GaussDB提供了多种用于调用函数的语句，其在语法结构上并无差别，请参见**CALL**和**CREATE FUNCTION**。

预备语句

表 1-385 预备语句相关 SQL

功能	相关SQL
创建一个预备语句	**PREPARE**
执行一个已提前准备好的预备语句	**EXECUTE**
删除指定的预备语句	**DEALLOCATE**

结束会话

用户与数据库之间建立的连接称为会话，GaussDB提供了结束会话的语句，请参见**ALTER SYSTEM KILL SESSION**。

执行匿名代码块

GaussDB提供了执行匿名代码块的语句，请参见**DO**。

在指定的节点上执行 SQL 语句

GaussDB提供了在指定的节点上执行SQL的语句，请参见**EXECUTE DIRECT**。

通过游标检索数据

GaussDB提供了通过已创建的游标进行检索数据的语句，请参见**FETCH**。

插入或替换数据

GaussDB提供了在表中插入或者替换新的数据的语句，请参见**REPLACE**。

版权所有 © 华为云计算技术有限公司

根据表达式计算值

GaussDB提供了根据指定的值表达式计算一个或一组行的值的语句，请参见**VALUES**。

导入数据

GaussDB提供了将文件中的数据导入到数据库指定表中的语句，请参见**LOAD DATA**。

定义游标

为了处理SQL语句，存储过程线程会分配一段内存区域用于保存上下文联系。游标是指向上下文区域的句柄或指针。借助游标，存储过程可以控制上下文区域的变化。所涉及的SQL语句，如**表1-386**所示。

表 1-386 游标定义相关 SQL

功能	相关SQL
创建游标	**CURSOR**
	DECLARE
移动游标	**MOVE**
关闭游标	**CLOSE**

显示 SQL 语句的执行计划

GaussDB提供了显示SQL语句的执行计划的语句，请参见**EXPLAIN**。

1.13.5 其他语法一览表

除了DCL、DDL和DML语法，GaussDB还提供了其他功能的语法。

关闭当前节点

GaussDB支持使用SHUTDOWN命令关闭当前数据库节点，请参见**SHUTDOWN**。

BUCKET 扩容相关 SQL

GaussDB支持用于扩容工具通知内核具体bucket已经完成搬迁的语句，请参见**MARK BUCKETS**。

清理数据库连接

GaussDB支持用于清理数据库连接的语句，请参见**CLEAN CONNECTION**。

存储执行计划

GaussDB提供了将查询执行的计划信息存储于PLAN_TABLE表中的语句，请参见**EXPLAIN PLAN**。

版权所有 © 华为云计算技术有限公司

预测

GaussDB提供了利用完成训练的模型进行推测任务的语句，请参见**PREDICT BY**。

事务相关 SQL

表 1-387 事务相关 SQL

功能	相关SQL
回滚当前事务并且撤销所有当前事务中所做的更改。	**ABORT**
	ROLLBACK
开始一个事务/启动事务。	**BEGIN**
	SET TRANSACTION
	START TRANSACTION
设置事务日志检查点。	**CHECKPOINT**
提交当前事务。	**COMMIT | END**
提交一个先前为两阶段提交准备好的事务。	**COMMIT PREPARED**
为当前事务做两阶段提交的准备。	**PREPARE TRANSACTION**
删除一个当前事务先前定义的保存点。	**RELEASE SAVEPOINT**
取消一个先前为两阶段提交准备好的事务。	**ROLLBACK PREPARED**
回滚到一个保存点。	**ROLLBACK TO SAVEPOINT**
在当前事务里建立一个新的保存点。	**SAVEPOINT**
设置当前事务检查行为的约束条件。	**SET CONSTRAINTS**

设置用户标识符

表 1-388 设置用户标识符相关 SQL

功能	相关SQL
设置当前会话的当前用户标识符	**SET ROLE**
把当前会话里的会话用户标识和当前用户标识都设置为指定的用户	**SET SESSION AUTHORIZATION**

版权所有 © 华为云计算技术有限公司

数据库升级相关 SQL

表 1-389 数据库升级相关 SQL

功能	相关SQL
生成系统对象的升级回滚脚本文件	GENERATED UPDATE SYSTEM OBJECT
将系统对象的版本信息刷新到系统表GS_SYSTEM_OBJECTS_VERSION中，用于升级时判断系统对象的变化情况	REFRESH SYSTEM OBJECT

1.13.6 Online DDL 及其操作

表对象DDL（数据库表对象的定义或者修改）操作根据其对数据的影响程度，可分为两大类：

即时变更类 DDL

定义与特点：

- 仅修改表的元数据，无需进行数据操作。
- 执行速度快，通常在毫秒级完成。
- 执行期间短暂持有排他锁，对并发业务影响小。

并发影响：

- 正常场景：当目标表上无长事务时，对业务影响较小。
- 异常场景：若存在长事务，可能阻塞DDL执行并影响后续业务。

当前支持的即时变更类DDL如**表1-390**所示。具体语法请参见**ALTER TABLE**。

版权所有 © 华为云计算技术有限公司

表 1-390 当前版本支持的即时变更类 DDL 介绍

分类	操作	耗时	备注
列操作	支持以下相关操作： ● 添加列：设置列默认值为表达式。 ● 删除列。 ● 重命名列。 ● 设置和删除列的默认值。 ● 扩展varchar大小（仅长度增加时）、 ● 修改numeric精度numeric(p,s)（仅p增大，s不变时）。 ● 修改ENUM列定义。	当仅修改元数据时，耗时与数据量无关。	● 即时变更类DDL无需结合ONLINE关键字，当为即时变更类DDL添加ONLINE关键字时，ONLINE关键字会忽略。 ● 即时变更类DDL执行期间会短暂持有排他锁。在没有并发长事务的场景下，DDL毫秒级完成，对并发事务影响较小。在DDL执行前，如果目标表上有并发长事务，DDL因等待排他锁而被阻塞，进一步会阻塞DDL之后的并发事务。
表操作	支持以下相关操作： ● 重命名表。 ● 指定字符集。 ● 删除表。	当仅修改元数据时，耗时与数据量无关。	
表空间操作	支持以下相关操作： ● 重命名表空间。 ● 修改表空间所有者。 ● 删除表空间。	当仅修改元数据时，耗时与数据量无关。	
约束操作	支持以下相关操作： ● 设置列约束为NULL。	当仅修改元数据时，耗时与数据量无关。	
分区操作	支持以下相关操作： ● 添加分区、删除分区、截断分区、行迁移开关、设置分区索引不可用。	当仅修改元数据时，耗时与数据量无关。	

数据操作类 DDL

定义与特点：

- 需要对实际数据进行重组或者检查操作。
- 执行时间与数据量成正比。
- 传统模式下执行期间表无法提供读写服务，可能导致长时间业务中断。

版权所有 © 华为云计算技术有限公司

在线 DDL：

定义与特点：

- 通过ONLINE关键字控制，数据操作类DDL通过基础数据重组（检查）和增量数据追增（检查）的方式进行在线变更。
- 不支持rowid表。

并发影响：

- 执行期间大部分时间用户可继续进行数据库的读写操作。
- 仅在前期和后期短暂持有排他锁。
- 显著降低对正常业务的影响。

当前支持的在线DDL的操作类型如**表1-391**所示。在线DDL具体参数设置和行为约束请参见**ALTER TABLE**、**VACUUM**。

表 1-391 当前版本支持的在线 DDL 介绍

分类	操作	耗时	备注
列操作	支持以下相关操作： • 修改列长度。 • 修改列精度。 • 修改列类型。	需要重整数据，耗时与数据量相关。	• 在线DDL需要结合ONLINE关键字实现。 • 在线DDL在初始化和最后元数据切换的阶段仍然会短暂持有排他锁，可能受目标表上的并发长事务的影响。在线DDL执行前或者元数据切换阶段，如果目标表上有并发长事务，DDL可能因等待排他锁而被阻塞，进一步会阻塞后续并发事务。 • 在线DDL不支持rowid表，详细语法请参见**ALTER TABLE**和**VACUUM**。
表操作	支持以下相关操作： 表级在线移动表空间。	需要移动数据，耗时与数据量相关。	
约束操作	支持以下相关操作： 添加非空、检查、唯一、主键约束。	需要查询数据，耗时与数据量相关。	
分区操作	支持以下相关操作： 切割/合并分区、交换分区、移动分区表空间	需要移动数据，耗时与数据量相关。	
回收存储空间	VACUUM FULL	需要重整数据，耗时与数据量相关。	
索引操作	支持创建索引CONCURRENTLY、重建索引CONCURRENTLY、删除索引CONCURRENTLY。	需要查询数据，耗时与数据量相关。	索引相关操作，请参见**CREATE INDEX**、**REINDEX**、**DROP INDEX**。

1.13.7 A

版权所有 © 华为云计算技术有限公司

1.13.7.1 ABORT

功能描述

回滚当前事务并且撤销所有当前事务中所做的更改。

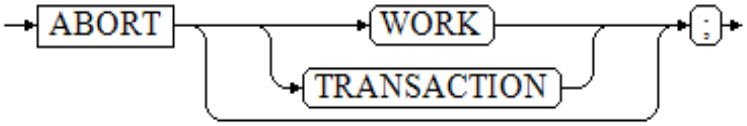
作用等同于**ROLLBACK**，早期SQL使用的是ABORT，现在推荐使用**ROLLBACK**。

注意事项

在事务外部执行ABORT语句不会影响事务的执行，但会返回一个NOTICE信息。

语法格式

```
ABORT [ WORK | TRANSACTION ] ;
```



参数说明

WORK | TRANSACTION

可选关键字，仅用于增强可读性，并无其他任何作用。

示例

```
--创建表customer_demographics_t1。
gaussdb=# CREATE TABLE customer_demographics_t1
(
    CD_DEMO_SK              INTEGER              NOT NULL,
    CD_GENDER               CHAR(1)                      ,
    CD_MARITAL_STATUS       CHAR(1)                      ,
    CD_EDUCATION_STATUS     CHAR(20)                     ,
    CD_PURCHASE_ESTIMATE    INTEGER                      ,
    CD_CREDIT_RATING        CHAR(10)                     ,
    CD_DEP_COUNT            INTEGER                      ,
    CD_DEP_EMPLOYED_COUNT   INTEGER                      ,
    CD_DEP_COLLEGE_COUNT    INTEGER
)
DISTRIBUTE BY HASH (CD_DEMO_SK);

--插入记录。
gaussdb=# INSERT INTO customer_demographics_t1 VALUES(1920801,'M', 'U', 'DOCTOR DEGREE', 200,
'GOOD', 1, 0,0);

--开启事务。
gaussdb=# START TRANSACTION;

--更新字段值。把cd_education_status字段值更新为Unknown。
gaussdb=# UPDATE customer_demographics_t1 SET cd_education_status= 'Unknown';

--终止事务，上面所执行的更新会被撤销掉。
gaussdb=# ABORT;

--查询数据。发现cd_education_status字段的值未被修改成Unknown。
gaussdb=# SELECT * FROM customer_demographics_t1 WHERE cd_demo_sk = 1920801;
cd_demo_sk | cd_gender | cd_marital_status | cd_education_status  | cd_purchase_estimate | cd_credit_rating
| cd_dep_count | cd_dep_employed_count | cd_dep_college_count
------------+-----------+-------------------+----------------------+----------------------+------------------
```


版权所有 © 华为云计算技术有限公司

```
+--------------+-----------------------+----------------------
   1920801 | M        | U              | DOCTOR DEGREE       |               200 | GOOD            |          1
|                 0 |                 0
(1 row)

--删除表。
gaussdb=# DROP TABLE customer_demographics_t1;
```

相关链接

SET TRANSACTION，**COMMIT | END**，**ROLLBACK**

1.13.7.2 ALTER AGGREGATE

功能描述

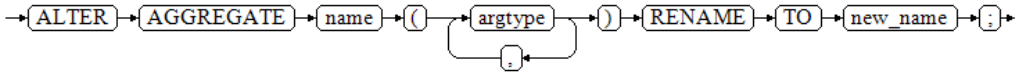
修改一个聚合函数的定义。包括名称、所有者和模式。

注意事项

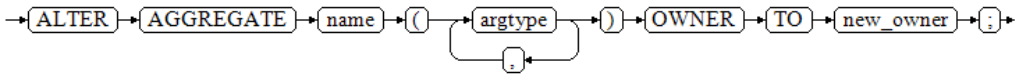
若要使用ALTER AGGREGATE语句，用户必须是该聚合函数的所有者；若要改变一个聚合函数的模式，必须在新模式上有CREATE权限；若要改变所有者，必须是新所有者的一个直接或间接成员，并且该角色（成员）必须在聚合函数的所属模式上有CREATE权限。以上约束限制了所有者通过删除和重建聚合函数执行非法操作，但具有SYSADMIN权限的用户，可以用任何方法任意更改聚合函数的所属关系。

语法格式

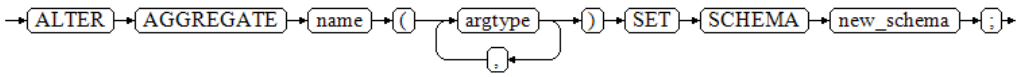
```
ALTER AGGREGATE name ( argtype [ , ... ] ) RENAME TO new_name;
```




```
ALTER AGGREGATE name ( argtype [ , ... ] ) OWNER TO new_owner;
```




```
ALTER AGGREGATE name ( argtype [ , ... ] ) SET SCHEMA new_schema;
```



参数说明

- **name**

  现有的聚合函数的名称，可以使用模式进行具体指定。

- **argtype**

  聚合函数操作的输入数据类型。若要引用一个零参数聚合函数，可以写入*代替输入数据类型列表。

- **new_name**

  聚合函数的新名称。

版权所有 © 华为云计算技术有限公司

- **new_owner**
  聚合函数的新所有者。
- **new_schema**
  聚合函数的新模式。

示例

- 修改聚合函数的名称。

```
--创建自定义函数。
gaussdb=# CREATE OR REPLACE FUNCTION int_add(int,int)
 RETURNS int AS $BODY$
DECLARE
BEGIN
 RETURN $1 + $2;
END;
$BODY$ language plpgsql;

--创建聚合函数。
gaussdb=# CREATE AGGREGATE myavg (int)
(
    sfunc = int_add,
    stype = int,
    initcond = '0'
);

--把一个接受int类型参数的聚合函数myavg重命名为 my_average。
gaussdb=# ALTER AGGREGATE myavg(int) RENAME TO my_average;
```

- 修改聚合函数的所有者。

```
--创建用户joe。
gaussdb=# CREATE USER joe PASSWORD '********';

--把一个接受int类型参数的聚合函数my_average的所有者改为joe。
gaussdb=# ALTER AGGREGATE my_average(int) OWNER TO joe;
```

- 修改聚合函数的模式。

```
--创建模式myschema。
gaussdb=# CREATE SCHEMA myschema;

--把一个接受int类型参数的聚合函数my_average移动到模式myschema中。
gaussdb=# ALTER AGGREGATE my_average(int) SET SCHEMA myschema;

--删除SCHEMA,用户及相关函数。
gaussdb=# DROP SCHEMA myschema CASCADE;
gaussdb=# DROP USER joe;
gaussdb=# DROP FUNCTION int_add(int,int);
```

相关链接

CREATE AGGREGATE，**DROP AGGREGATE**

兼容性

SQL标准里没有ALTER AGGREGATE语句。

1.13.7.3 ALTER ASYNC ENCRYPTION KEY ROTATION

功能描述

用于透明数据异步加密密钥轮转。

版权所有 © 华为云计算技术有限公司

注意事项

- 仅在enable_tde与enable_tde_async_encryption参数开启时，该语句可被执行。
- 只有具备SYSADMIN权限的用户有权限执行此语句。
- M-Compatibility数据库不支持该语句。
- 不支持记录统一审计日志。

语法格式

```
ALTER ASYNC ENCRYPTION KEY ROTATION;
```

示例

```
--确认透明加密异步加密开关打开。
gaussdb=# show enable_tde_async_encryption;
 enable_tde_async_encryption
-----------------------------
 on
(1 row)

--创建两张透明加密表t1、t2；执行\d元命令，可观察发现两张表使用同一个数据密钥（也即异步加密密钥）。
gaussdb=# CREATE TABLE t1 (c1 INT, c2 TEXT) WITH (enable_tde = on);
CREATE TABLE
gaussdb=# CREATE TABLE t2 (c1 INT, c2 TEXT) WITH (enable_tde = on);
CREATE TABLE
gaussdb=# \d

--轮转异步加密密钥。
gaussdb=# ALTER ASYNC ENCRYPTION KEY ROTATION;
ALTER TDE ROTATE ENCRYPTION KEY

--创建新的透明加密表t3；执行\d元命令，可观察发现表t3数据密钥与t1、t2不同，也即异步加密密钥发生了实际轮转。
gaussdb=# CREATE TABLE t3 (c1 INT, c2 TEXT) WITH (enable_tde = on);
CREATE TABLE
gaussdb=# \d

--删除加密表。
gaussdb=# DROP TABLE t1,t2,t3;
```

1.13.7.4 ALTER AUDIT POLICY

功能描述

修改统一审计策略。

注意事项

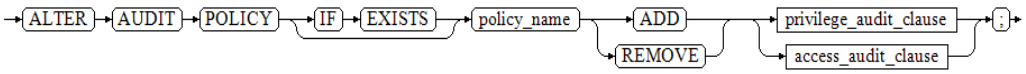
- 审计策略的维护有权限限制，只有POLADMIN、SYSADMIN或初始用户有权限进行此操作。
- 需要打开enable_security_policy开关统一审计策略才可以生效。

语法格式

添加/删除审计策略中的操作类型。

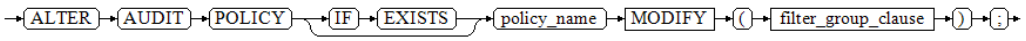
版权所有 © 华为云计算技术有限公司

```
ALTER AUDIT POLICY [ IF EXISTS ] policy_name { ADD | REMOVE } { privilege_audit_clause |
access_audit_clause };
```



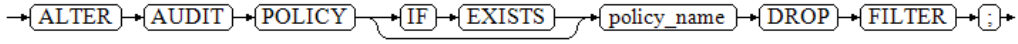
修改审计策略中的过滤条件。

```
ALTER AUDIT POLICY [ IF EXISTS ] policy_name MODIFY ( filter_group_clause );
```



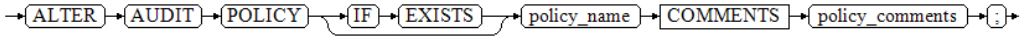
将审计策略中的过滤条件删除。

```
ALTER AUDIT POLICY [ IF EXISTS ] policy_name DROP FILTER;
```



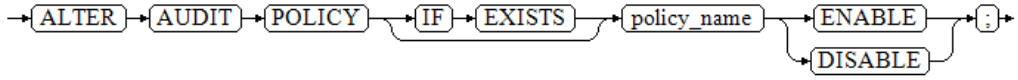
修改审计策略描述。

```
ALTER AUDIT POLICY [ IF EXISTS ] policy_name COMMENTS policy_comments;
```



打开或者关闭审计策略。

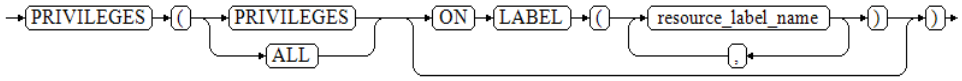
```
ALTER AUDIT POLICY [ IF EXISTS ] policy_name { ENABLE | DISABLE };
```



- privilege_audit_clause：

  审计策略中具体的PRIVILEGES操作类型及目标资源标签。

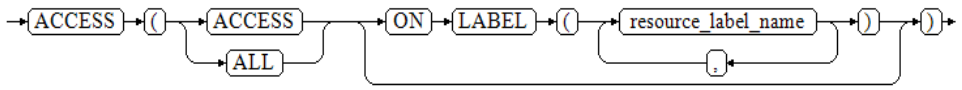
  ```
  PRIVILEGES ({ PRIVILEGES | ALL } [ ON LABEL ( resource_label_name [, ... ] ) ])
  ```

- access_audit_clause：

  审计策略中具体的ACCESS操作类型及目标资源标签。

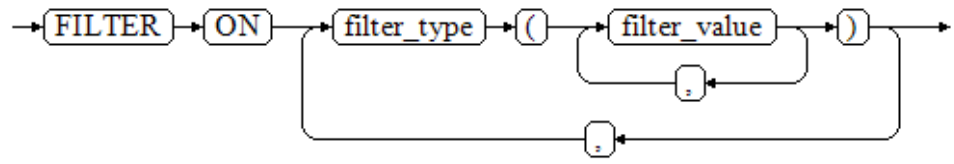
  ```
  ACCESS ({ ACCESS | ALL } [ ON LABEL ( resource_label_name [, ... ] ) ])
  ```

- filter_group_clause：

  审计策略中的过滤条件。

  ```
  FILTER ON { filter_type ( filter_value [, ... ] ) } [, ... ]
  ```

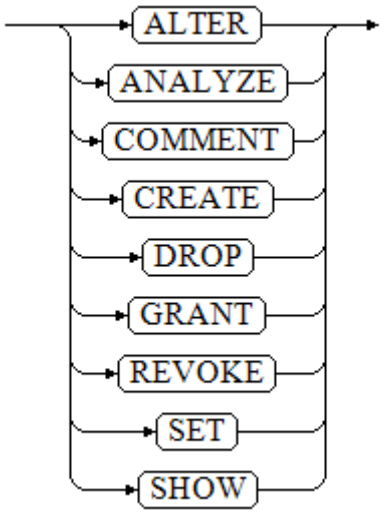
  

- PRIVILEGES：

  指的是针对数据库执行如下操作时进行审计。

版权所有 © 华为云计算技术有限公司

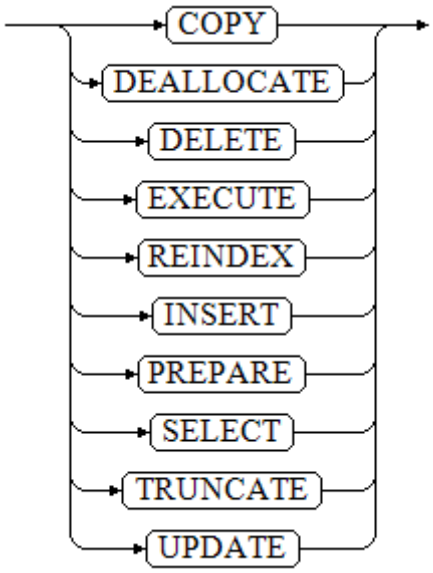
```
{ ( ALTER | ANALYZE | COMMENT | CREATE | DROP | GRANT | REVOKE | SET | SHOW ) }
```



- ACCESS：

  指的是针对数据库执行如下操作时进行审计。

```
{ ( COPY | DEALLOCATE | DELETE | EXECUTE | REINDEX | INSERT | PREPARE | SELECT | TRUNCATE | UPDATE ) }
```



参数说明

- **policy_name**

  审计策略名称，需满足唯一约束，不可重复。

  取值范围：字符串，要符合**标识符命名规范**。

- **resource_label_name**

  资源标签名称。

- **PRIVILEGES**

  当数据库执行指定的操作时进行审计，目前支持对ALTER、ANALYZE、COMMENT、CREATE、DROP、GRANT、REVOKE、SET、SHOW操作进行审计。

版权所有 © 华为云计算技术有限公司

- **ACCESS**

  当数据库执行指定的操作时进行审计，目前支持对COPY、DEALLOCATE、DELETE、EXECUTE、REINDEX、INSERT、PREPARE、SELECT、TRUNCATE、UPDATE。

- **ALL**

  PRIVILEGES或ACCESS中支持的所有关于数据库的操作的审计。当形式为{ PRIVILEGES | ALL }时，ALL表示所有PRIVILEGES操作；当形式为{ ACCESS | ALL }时，ALL表示所有ACCESS操作。

- **filter_type**

  指定审计策略的过滤信息，过滤类型包括：IP、ROLES、APP。

- **filter_value**

  具体过滤信息内容。

- **policy_comments**

  用于记录策略相关的描述信息。

- **ENABLE | DISABLE**

  可以打开或关闭统一审计策略。

示例

- 添加/删除审计策略中的操作类型。

```
--创建一个对数据库执行CREATE的审计策略adt1。
gaussdb=# CREATE AUDIT POLICY adt1 PRIVILEGES CREATE;

--添加adt1审计策略中的DROP。
gaussdb=# ALTER AUDIT POLICY adt1 ADD PRIVILEGES (DROP);

--删除adt1审计策略中的DROP。
gaussdb=# ALTER AUDIT POLICY adt1 REMOVE PRIVILEGES (DROP);
```

- 修改审计策略的注释信息。

```
--修改adt1审计策略的注释信息为adt1_comments。
gaussdb=#  ALTER AUDIT POLICY adt1 COMMENTS 'adt1_comments';
```

- 修改审计策略的过滤信息。

```
--创建bob_audit用户。
gaussdb=#  CREATE USER bob_audit PASSWORD '********';

--修改adt1审计策略的过滤用户为bob_audit。
gaussdb=#  ALTER AUDIT POLICY adt1 MODIFY (FILTER ON (ROLES(bob_audit)));

--删除bob_audit用户。
gaussdb=# DROP USER bob_audit;
```

- 删除审计策略的过滤条件。

```
--删除adt1审计策略的过滤条件。
gaussdb=#  ALTER AUDIT POLICY adt1  DROP FILTER;
```

- 关闭审计策略。

```
--关闭adt1审计策略。
gaussdb=#  ALTER AUDIT POLICY adt1  DISABLE;

--删除adt1审计策略。
gaussdb=# DROP AUDIT POLICY adt1;
```

相关链接

CREATE AUDIT POLICY，**DROP AUDIT POLICY**。

版权所有 © 华为云计算技术有限公司

1.13.7.5 ALTER COLUMN ENCRYPTION KEY

功能描述

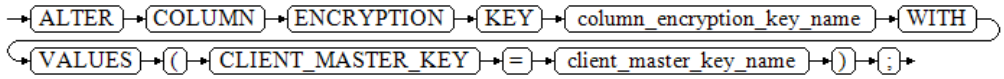
CMK密钥轮转，轮换加密COLUMN ENCRYPTION KEY的CLIENT MASTER KEY，对COLUMN ENCRYPTION KEY明文进行重加密。

注意事项

- 本语法属于全密态数据库特有语法，当连接数据库服务器时，需打开全密态数据库的开关，才能使用本语法。
- 本语法只能进行CMK轮转，对列加密密钥明文进行重加密，实际上列加密密钥明文不变，不能修改加密列数据的密文。

语法格式

```
ALTER COLUMN ENCRYPTION KEY column_encryption_key_name WITH VALUES ( CLIENT_MASTER_KEY = client_master_key_name );
```



参数说明

- **column_encryption_key_name**

  该参数作为密钥对象名，在同一命名空间下，需满足唯一约束。

  取值范围：字符串，要符合**标识符命名规范**。

- **client_master_key_name**

  指定用于重加密该CEK的新的CMK。

  取值范围：已存在的CMK对象名称，该CMK对象由CREATE CLIENT MASTER KEY语法创建。与密钥轮转前指定的客户端主密钥不为同一密钥。

须知

国密算法约束：由于SM2、SM3、SM4等算法属于中国国家密码标准算法，为规避法律风险，需配套使用。如果轮转CEK前使用的CMK是国密算法，则轮转CEK指定的CMK仍必须使用国密算法。

相关链接

CREATE COLUMN ENCRYPTION KEY，**DROP COLUMN ENCRYPTION KEY**

1.13.7.6 ALTER COORDINATOR

功能描述

修改协调节点CN的状态。可以在集群任意一个正常的CN上执行该语句，修改pgxc_node系统表中指定节点的nodeis_active字段值，并且还可以指定在哪些节点上修改系统表。

版权所有 © 华为云计算技术有限公司

注意事项

- ALTER COORDINATOR是修改系统表的语句，只有管理员用户和内部维护模式（例如CM集群管理员）可以执行。
- 该语句是CN剔除特性专用，需配合其他操作，不要单独使用，不建议用户自己执行。
- 该语句执行完成后，需要调用SELECT reload_active_coordinator()语句，更新发生了变化的系统表所属的节点连接池的信息。

语法格式

```
ALTER COORDINATOR nodename SET status
    WITH (other_nodename [, ... ]) (nodetype);
```

参数说明

- **nodename**

  CN节点名，对应在pgxc_node系统表的一行记录，指定后将修改记录中的nodeis_active字段值。

  取值范围：字符串，只支持CN节点名，并且需保证该节点名在pgxc_node系统表中存在对应的记录。

- **status**

  CN的状态，pgxc_node系统表中nodeis_active字段的更新值。

  取值范围：FALSE、TRUE

- **other_nodename**

  该SQL执行的节点范围，ALTER COORDINATOR执行时会自动下发到范围内的所有节点，需要包含当前执行节点。

  取值范围：字符串，只支持CN节点名，要保证该节点名在pgxc_node系统表中有对应的记录，并且节点状态正常，否则SQL执行失败。

- **nodetype**

  待修改节点的类型，pgxc_node系统表中node_type的值。

  取值范围："C"（协调节点）。

示例

- 剔除发生故障的CN。

```
--集群有3个CN，cn_5001、cn_5002、cn_5003，均处于正常工作状态。cn_5001发生故障且满足剔除时间要求后，需要将cn_5001从集群中剔除。
gaussdb=# ALTER COORDINATOR cn_5001 SET False WITH (cn_5002,cn_5003);

--刷新pgxc_node系统表。在cn_5002和cn_5003节点上执行SQL刷新pgxc_node系统表中cn_5001对应记录的nodeis_active为false。
gaussdb=# SELECT reload_active_coordinator();

--查看CN的状态。nodeis_active字段的值已变成f(false)。
gaussdb=#  SELECT nodeis_active,node_name FROM pgxc_node where node_name='cn_5001';
 nodeis_active | node_name
---------------+-----------
 f             | cn_5001
(1 row)
```

- 恢复已解除故障的CN。

```
--cn_5001故障解除后，在集群中加回cn_5001。
gaussdb=#  ALTER COORDINATOR cn_5001 SET True WITH (cn_5002,cn_5003);
```

 版权所有 © 华为云计算技术有限公司

```
--刷新pgxc_node系统表。在cn_5002和cn_5003节点上执行SQL刷新pgxc_node系统表中cn_5001对应记录的nodeis_active为true。
gaussdb=# SELECT reload_active_coordinator();

--查看CN的状态。nodeis_active字段的值已变成t(true)。
gaussdb=# SELECT nodeis_active,node_name FROM pgxc_node where node_name='cn_5001';
 nodeis_active | node_name
---------------+-----------
 t             | cn_5001
(1 row)
```

1.13.7.7 ALTER DATABASE

功能描述

修改数据库的属性，包括：数据库名称、所有者、连接数限制以及对象隔离属性等。

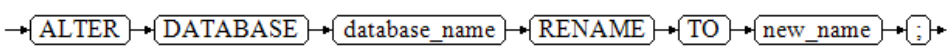
注意事项

- 仅数据库的所有者或者被授予了数据库ALTER权限的用户可以执行ALTER DATABASE命令，系统管理员默认拥有此权限。针对所要修改属性的不同，存在以下权限约束：
  - 修改数据库名称：必须拥有CREATEDB权限。
  - 修改数据库所有者：当前用户必须是该数据库的所有者或者系统管理员，必须拥有CREATEDB权限，且该用户是新所有者角色的成员。
  - 修改数据库默认表空间：该用户必须拥有新表空间的CREATE权限。进行该操作会从物理上将一个数据库原来缺省表空间上的表和索引移至新的表空间。注意不在缺省表空间的表和索引不受此影响。
- 不能直接对当前使用的数据库进行重命名，如果需要重新命名，请连接至其他数据库之后对原数据库进行重命名。
- 不允许将其他模式重命名为templatem，不允许将templatem模式重命名为其他名称，不允许将templatea模板重命名为其他名称。

语法格式

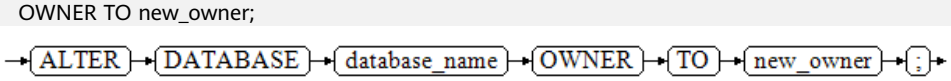
- 修改数据库名称。

```
ALTER DATABASE database_name
   RENAME TO new_name;
```



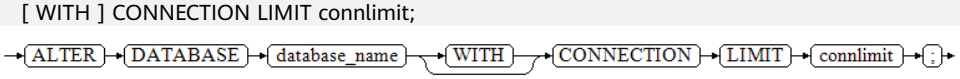
- 修改数据库所有者。

```
ALTER DATABASE database_name
   OWNER TO new_owner;
```



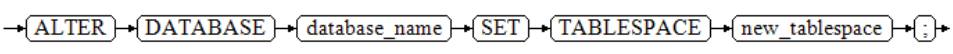
- 修改数据库的最大连接数。

```
ALTER DATABASE database_name
   [ WITH ] CONNECTION LIMIT connlimit;
```



- 修改数据库默认表空间。

```
ALTER DATABASE database_name
   SET TABLESPACE new_tablespace;
```



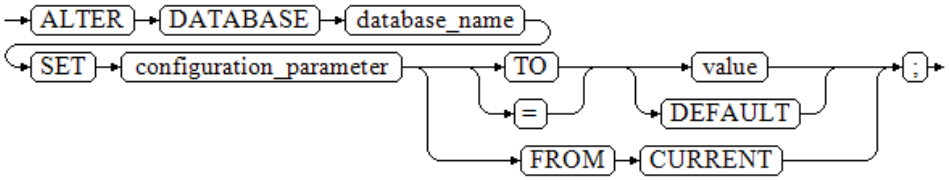
版权所有 © 华为云计算技术有限公司

须知

如果数据库中的某些表或对象已经创建在new_tablespace下，则无法将该数据库的默认表空间修改为new_tablespace，执行会报错。

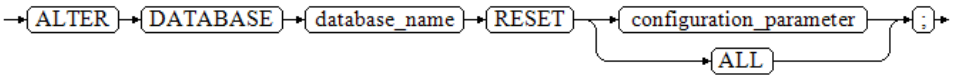
- 修改数据库指定会话参数值。当前版本不支持设置数据库级别参数。

```
ALTER DATABASE database_name
    SET configuration_parameter { { TO | = } { value | DEFAULT } | FROM CURRENT };
```



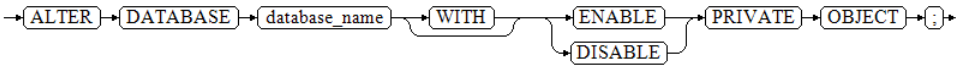
- 数据库配置参数重置。当前版本不支持设置数据库级别参数。

```
ALTER DATABASE database_name RESET
    { configuration_parameter | ALL };
```



- 修改数据库对象隔离属性。修改数据库的对象隔离属性时须连接至该数据库，否则无法更改。

```
ALTER DATABASE database_name [ WITH ] { ENABLE | DISABLE } PRIVATE OBJECT;
```

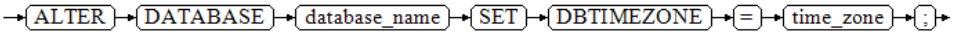


说明

新创建的数据库，对象隔离属性默认关闭。当开启该属性，数据库会为系统表PG_CLASS、PG_ATTRIBUTE、PG_PROC、PG_NAMESPACE、PGXC_SLICE、PG_DATABASE和PG_PARTITION自动添加行级访问控制策略，普通用户仅支持查看系统表中有权访问的对象（表、函数、视图、字段等）；管理员用户不受该属性的影响，当开启该属性后，管理员仍可查看全量的数据库对象。

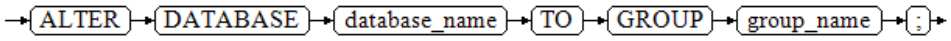
- 修改数据库时区。

```
ALTER DATABASE database_name SET DBTIMEZONE = time_zone;
```



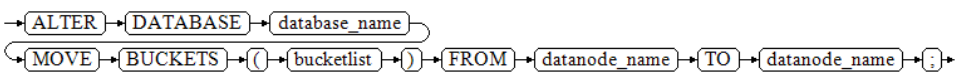
- 为创建hashbucket表绑定node group，确定bucketmap映射关系。

```
ALTER DATABASE database_name
    TO GROUP group_name ;
```



- 将指定的bucket从一个datanode迁移至另一个datanode。

```
ALTER DATABASE database_name
    MOVE BUCKETS(bucketlist) FROM datanode_name TO datanode_name;
```



说明

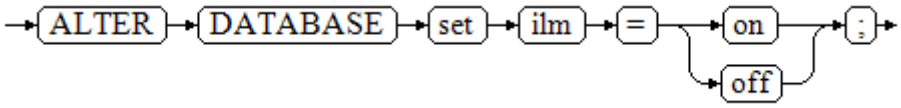
该语句仅支持在hashbucket扩容期间调用，是hashbucket库级重分布命令，例如：

```
ALTER DATABASE testdb MOVE BUCKETS (0，1，2，3) FROM datanode1 TO datanode3;
```


版权所有 © 华为云计算技术有限公司

- 开启或者关闭数据库ILM特性。

```
ALTER DATABASE set ilm = { on | off };
```



参数说明

- **database_name**

  需要修改属性的数据库名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **new_name**

  数据库的新名称。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  - 当数据库名称超过63个字符时，数据库会截断并保留前63个字符当作数据库名称。
  - 当数据库名称中包含大写字母时，数据库会自动转换为小写字母。
  - 如果需要创建包含大写字母的数据库名称，则需要使用双引号进行包裹。

- **new_owner**

  数据库的新所有者。

  取值范围：字符串，有效的用户名。

- **connlimit**

  数据库可以接收的最大并发连接数（管理员用户连接除外）。

  取值范围：[-1, 2^{31}-1]的整数。默认值为-1，表示不限制数据库的最大并发连接数，建议设置为1~50的整数。

- **new_tablespace**

  数据库新的默认表空间，该表空间为数据库中已经存在的表空间。

  取值范围：字符串，有效的表空间名。默认表空间为pg_default。

  说明

  修改数据库默认表空间，会将旧表空间中的所有表和索引转移到迁移至新表空间中，该操作不会影响其他非默认表空间中的表和索引。

- **time_zone**

  设置数据库的时区值，需要拥有数据库的操作权限。

  取值范围：字符串，当前支持设置的时区值为：

  - 系统支持的时区和其相应的缩写。
  - -15:59~+15:00。

  说明

  该参数仅在连接数据库为ORA兼容模式数据库且已设置a_format_version='10c'以及a_format_dev_version='s2'时支持指定。

- **group_name**

  hashbucket表绑定的node group名称，普通用户只支持绑定到installation node group上且不支持修改。

版权所有 © 华为云计算技术有限公司

- **bucketlist**
  重分布涉及的bucketlist
  取值范围：字符串，每个bucket的取值范围为[0, 1023]。
- **datanode_name**
  迁移bucket操作涉及的节点名称。
  取值范围：字符串，要符合**标识符命名规范**。
- **configuration_parameter**
  需要修改属性的数据库会话参数。

  须知

  当前版本不支持设置数据库级别参数。

  - **value**
    将指定的数据库会话参数值设置为给定的值。
    取值范围：字符串。
    - DEFAULT：在新的会话中使用系统的缺省设置。
    - OFF：关闭设置。
    - RESET：在新的会话中使用系统的缺省设置。
    - 用户指定的值：需要满足修改参数的取值限制。
  - **FROM CURRENT**
    取当前会话中的值设置为configuration_parameter的值。
- **RESET configuration_parameter**
  重置指定的数据库会话参数值。

  须知

  当前版本不支持重置数据库级别参数。
- **RESET ALL**
  重置全部的数据库会话参数值。

  须知

  当前版本不支持重置数据库级别参数。

示例

- 修改数据库名称

```
--创建数据库testdb。
gaussdb=# CREATE DATABASE testdb;
```


版权所有 © 华为云计算技术有限公司

```
--将testdb重命名为test_db1。
gaussdb=# ALTER DATABASE testdb RENAME TO test_db1;
```

- 修改数据库最大连接数

```
--修改test_db1最大连接数为100。
gaussdb=# ALTER DATABASE test_db1 WITH CONNECTION LIMIT 100;

--查看test_db1信息。
gaussdb=# SELECT datname,datconnlimit FROM pg_database WHERE datname = 'test_db1';
 datname  | datconnlimit
----------+--------------
 test_db1 |          100
(1 row)
```

- 修改数据库所有者

```
--创建用户scott。
gaussdb=# CREATE USER scott PASSWORD '********';

--将test_db1的所有者修改为scott。
gaussdb=# ALTER DATABASE test_db1 OWNER TO scott;

--查看test_db1信息。
gaussdb=# SELECT t1.datname, t2.usename
          FROM pg_database t1, gs_user t2
          WHERE t1.datname='test_db1' AND t1.datdba=t2.usesysid;
 datname  | usename
----------+---------
 test_db1 | scott
(1 row)
```

- 修改数据库默认表空间

```
--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace1/tbs_data1';

--修改test_db1默认表空间。
gaussdb=# ALTER DATABASE test_db1 SET TABLESPACE tbs_data1;

--查看test_db1信息。
gaussdb=# SELECT t1.datname AS database, t2.spcname AS tablespace
          FROM pg_database t1, pg_tablespace t2
          WHERE t1.dattablespace = t2.oid AND
               t1.datname = 'test_db1';
 database | tablespace
----------+------------
 test_db1 | tbs_data1
(1 row)
```

- 修改数据库时区

  前提条件：当前连接数据库为ORA兼容模式数据库且已设置a_format_version='10c'以及a_format_dev_version='s2'。

```
--创建ORA兼容性数据库，默认为PRC时区。
gaussdb=# CREATE DATABASE testdb2 DBCOMPATIBILITY = 'ORA';

--查看数据库testdb2时区。
gaussdb=# SELECT datname,datcompatibility,dattimezone FROM pg_database WHERE datname =
'testdb2';
 datname | datcompatibility | dattimezone
---------+------------------+-------------
 testdb2 | ORA              | PRC
(1 row)

--切换至数据库testdb2，修改数据库时区为'+08:00'。
gaussdb=# \c testdb2
testdb2=# SET a_format_version='10c';
testdb2=# SET a_format_dev_version='s2';
testdb2=# ALTER DATABASE testdb2 SET DBTIMEZONE = '+08:00';

--查看数据库时区。
testdb2=# SELECT datname,datcompatibility,dattimezone FROM pg_database WHERE datname =
'testdb2';
```


版权所有 © 华为云计算技术有限公司

```
 datname | datcompatibility | dattimezone
---------+------------------+-------------
 testdb2 | ORA              | +08:00
(1 row)
```

- 修改数据库对象隔离属性

```
--创建用户jack。
gaussdb=# CREATE USER jack PASSWORD '********';

--在test_db1创建表test_tbl1。
gaussdb=# \c test_db1
test_db1=# CREATE TABLE test_tbl1(c1 int,c2 int);

--切换至jack用户查看pg_tables。
test_db1=# SET ROLE jack PASSWORD '********';
test_db1=> SELECT tablename FROM pg_tables WHERE tablename = 'test_tbl1';
 tablename
-----------
 test_tbl1
(1 row)

--修改对象隔离属性。
test_db1=> SET ROLE scott PASSWORD '********';
test_db1=> ALTER DATABASE test_db1 ENABLE PRIVATE OBJECT;

--切换至jack用户查看pg_tables。
test_db1=> SET ROLE jack PASSWORD '********';

--由于隔离属性的原因，该查询只能查出0条数据。
test_db1=> SELECT tablename FROM pg_tables WHERE tablename = 'test_tbl1';
 tablename
-----------
(0 rows)

--切换至默认用户删除。
test_db1=> RESET ROLE;
test_db1=# DROP TABLE public.test_tbl1;

--切换至默认数据库，请根据实际情况修改数据库名字。
test_db1=# \c postgres
gaussdb=# DROP DATABASE test_db1;
gaussdb=# DROP DATABASE testdb2;
gaussdb=# DROP TABLESPACE tbs_data1;
gaussdb=# DROP USER jack;
gaussdb=# DROP USER scott;
```

相关链接

CREATE DATABASE，**DROP DATABASE**

1.13.7.8 ALTER DATABASE LINK

功能描述

修改DATABASE LINK对象。

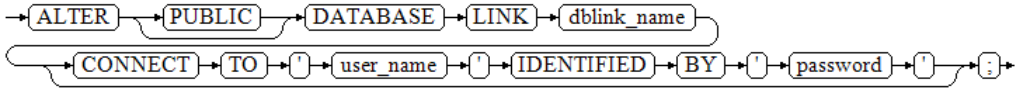
注意事项

- DATABASE LINK详细说明请见《开发指南》中“特性使用指导 > DATABASE LINK”章节。
- 当DATABASE LINK连接的数据库为GaussDB库时，目前仅支持修改DATABASE LINK对象的用户名和密码。

版权所有 © 华为云计算技术有限公司

语法格式

```
ALTER [PUBLIC] DATABASE LINK dblink_name
   [CONNECT TO 'user_name' IDENTIFIED BY 'password'];
```



参数说明

- **dblink_name**
  连接名称。
- **user_name**
  远程被连接数据库用户名。
- **password**
  远程被连接数据库用户密码。
- **PUBLIC**
  连接类型，未指定PUBLIC时，默认为PRIVATE。

示例

```
-- 创建一个兼容性为ORA的数据。
gaussdb=#  CREATE DATABASE ora_test_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE

-- 切换数据库。
gaussdb=# \c ora_test_db

-- 创建拥有系统管理员权限的用户。
ora_test_db=# CREATE USER local_user WITH SYSADMIN PASSWORD '********';
CREATE ROLE
ora_test_db=# SET ROLE local_user PASSWORD '********';
SET

-- 创建公共DBLink。
ora_test_db=> CREATE PUBLIC DATABASE LINK public_dblink CONNECT TO 'local_user' IDENTIFIED BY
'********' USING (host '***.***.***.***',port '*****',dbname 'remote_db');
CREATE DATABASE LINK

-- 创建普通用户。
ora_test_db=> CREATE USER remote_user PASSWORD '********';
CREATE ROLE

-- 修改DBLink对象信息。
ora_test_db=> ALTER PUBLIC DATABASE LINK public_dblink CONNECT TO 'remote_user' IDENTIFIED BY
'********';
ALTER DATABASE LINK

-- 删除公共DBLink。
ora_test_db=> DROP PUBLIC DATABASE LINK public_dblink;
DROP DATABASE LINK

-- 删除用户。
ora_test_db=> RESET ROLE;
RESET
ora_test_db=# DROP USER local_user;
DROP ROLE
ora_test_db=# DROP USER remote_user;
DROP ROLE

-- 切换回初始数据库，并删除测试数据库。请用真实的数据库名替换postgres。
```


版权所有 © 华为云计算技术有限公司

```
ora_test_db=# \c postgres
gaussdb=# DROP DATABASE ora_test_db;
DROP DATABASE
```

相关链接

CREATE DATABASE LINK，DROP DATABASE LINK

1.13.7.9 ALTER DEFAULT PRIVILEGES

功能描述

用于修改数据库中用户在特定对象上默认拥有的权限，不会影响到分配已有对象中的权限。

注意事项

目前只支持表（包括视图）、序列、函数，类型，密态数据库客户端主密钥和列加密密钥的权限更改。

语法格式

```
ALTER DEFAULT PRIVILEGES
    [ FOR { ROLE | USER } target_role [, ...] ]
    [ IN SCHEMA schema_name [, ...] ]
    abbreviated_grant_or_revoke;
```

- 其中abbreviated_grant_or_revoke子句用于指定对哪些对象进行授权或回收权限。

  ```
  {grant_on_tables_clause
    | grant_on_sequences_clause
    | grant_on_functions_clause
    | grant_on_types_clause
    | grant_on_client_master_keys_clause
    | grant_on_column_encryption_keys_clause
    | revoke_on_tables_clause
    | revoke_on_sequences_clause
    | revoke_on_functions_clause
    | revoke_on_types_clause
    | revoke_on_client_master_keys_clause
    | revoke_on_column_encryption_keys_clause
  }
  ```

- 其中grant_on_tables_clause子句用于对表授权。

  ```
  GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | ALTER | DROP |
  COMMENT | INDEX | VACUUM }
      [, ...] | ALL [ PRIVILEGES ] }
      ON TABLES
      TO { [ GROUP ] role_name | PUBLIC } [, ...]
      [ WITH GRANT OPTION ]
  ```

- 其中grant_on_sequences_clause子句用于对序列授权。

  ```
  GRANT { { SELECT | UPDATE | USAGE | ALTER | DROP | COMMENT }
      [, ...] | ALL [ PRIVILEGES ] }
      ON SEQUENCES
      TO { [ GROUP ] role_name | PUBLIC } [, ...]
      [ WITH GRANT OPTION ]
  ```

- 其中grant_on_functions_clause子句用于对函数授权。

  ```
  GRANT { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
      ON FUNCTIONS
      TO { [ GROUP ] role_name | PUBLIC } [, ...]
      [ WITH GRANT OPTION ]
  ```


版权所有 © 华为云计算技术有限公司

- 其中grant_on_types_clause子句用于对类型授权。
```
GRANT { { USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON TYPES
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ]
```
- 其中grant_on_client_master_keys_clause子句用于对客户端主密钥授权。
```
GRANT { { USAGE  | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON CLIENT_MASTER_KEYS
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ]
```
- 其中grant_on_column_encryption_keys_clause子句用于对列加密密钥授权。
```
GRANT { { USAGE | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON COLUMN_ENCRYPTION_KEYS
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ]
```
- 其中revoke_on_tables_clause子句用于回收表对象的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | ALTER | DROP | COMMENT |
INDEX | VACUUM }
    [, ...] | ALL [ PRIVILEGES ] }
    ON TABLES
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT | CASCADE CONSTRAINTS ]
```
- 其中revoke_on_sequences_clause子句用于回收序列的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | UPDATE | USAGE | ALTER | DROP | COMMENT }
    [, ...] | ALL [ PRIVILEGES ] }
    ON SEQUENCES
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT | CASCADE CONSTRAINTS ]
```
- 其中revoke_on_functions_clause子句用于回收函数的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON FUNCTIONS
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT | CASCADE CONSTRAINTS ]
```
- 其中revoke_on_types_clause子句用于回收类型的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON TYPES
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT | CASCADE CONSTRAINTS ]
```
- 其中revoke_on_client_master_keys_clause子句用于回收客户端主密钥的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { USAGE | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON CLIENT_MASTER_KEYS
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT | CASCADE CONSTRAINTS ]
```
- 其中revoke_on_column_encryption_keys_clause子句用于回收列加密密钥的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { USAGE | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON COLUMN_ENCRYPTION_KEYS
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT | CASCADE CONSTRAINTS ]
```

参数说明

- **target_role**

  已有角色的名称。如果未指定FOR ROLE | USER，则默认值为当前角色/用户。

版权所有 © 华为云计算技术有限公司

取值范围：已有角色的名称。

- **schema_name**
  已有模式的名称。
  取值范围：现有模式的名称。

  说明

  target_role必须拥有schema_name的CREATE权限。

- **role_name**
  被授予或者取消权限角色的名称。
  取值范围：已存在的角色名称。

须知

如果想删除一个被赋予了默认权限的角色，有必要恢复改变的缺省权限或者使用DROP OWNED BY来为角色脱离缺省的权限记录。

示例

```
--创建Schema。
gaussdb=# CREATE SCHEMA tpcds;

--将创建在模式tpcds里的所有表（和视图）的SELECT权限授予每一个用户。
gaussdb=# ALTER DEFAULT PRIVILEGES IN SCHEMA tpcds GRANT SELECT ON TABLES TO PUBLIC;

--创建用户普通用户jack。
gaussdb=# CREATE USER jack PASSWORD '******';

--将tpcds下的所有表的插入权限授予用户jack。
gaussdb=# ALTER DEFAULT PRIVILEGES IN SCHEMA tpcds GRANT INSERT ON TABLES TO jack;

--将tpcds下由jack创建的所有表的插入权限授予用户jake（假设数据库中已存在该用户）。
gaussdb=#  GRANT USAGE,CREATE ON SCHEMA tpcds TO jack;
gaussdb=# ALTER DEFAULT PRIVILEGES FOR ROLE jack IN SCHEMA tpcds GRANT INSERT ON TABLES TO
jake;

--撤销上述权限。
gaussdb=# ALTER DEFAULT PRIVILEGES IN SCHEMA tpcds REVOKE SELECT ON TABLES FROM PUBLIC;
gaussdb=# ALTER DEFAULT PRIVILEGES IN SCHEMA tpcds REVOKE INSERT ON TABLES FROM jack;

--删除用户jack。
gaussdb=# DROP USER jack CASCADE;

--删除Schema。
gaussdb=# DROP SCHEMA tpcds;
```

相关链接

GRANT，**REVOKE**

1.13.7.10 ALTER DIRECTORY

功能描述

修改一个目录对象的属性。

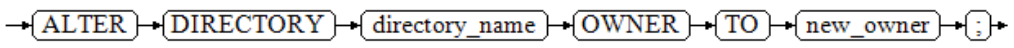
版权所有 © 华为云计算技术有限公司

注意事项

- 目前只支持修改目录对象的所有者。
- 当enable_access_server_directory=off时，只允许初始用户修改目录对象的所有者；当enable_access_server_directory=on时，具有SYSADMIN权限的用户和目录对象的所有者可以修改目录对象，且要求该用户是新所有者的成员。

语法格式

```
ALTER DIRECTORY directory_name
   OWNER TO new_owner;
```



参数说明

- **directory_name**

  需要进行修改属性的具体目录。

  取值范围：已经存在的目录名称。

- **new_owner**

  目录的新所有者。

示例

```
--创建目录。
gaussdb=# CREATE OR REPLACE DIRECTORY dir as '/tmp/';

--创建用户。
gaussdb=# CREATE USER jim PASSWORD '********';

--修改目录的所有者。
gaussdb=# ALTER DIRECTORY dir OWNER TO jim;

--删除目录对象。
gaussdb=# DROP DIRECTORY dir;

--删除用户。
gaussdb=# DROP USER jim;
```

相关链接

CREATE DIRECTORY，**DROP DIRECTORY**

1.13.7.11 ALTER FOREIGN DATA WRAPPER

功能描述

修改外部数据封装的属性。

注意事项

只有在内部初始化阶段或升级脚本中能够修改外部数据封装，其他场景下用户无法直接使用该语法。

版权所有 © 华为云计算技术有限公司

语法格式

设置外部数据封装属性

```
ALTER FOREIGN DATA WRAPPER name
[ HANDLER handler_function | NO HANDLER ]
[ VALIDATOR validator_function | NO VALIDATOR ]
[ OPTIONS ( { [ ADD | SET | DROP ] option ['value'] } [, ... ]) ];
```

参数说明

- **name**

  已有外部数据封装的名称。

- **HANDLER handler_function**

  为外部数据封装指定一个新的处理函数。

- **NO HANDLER**

  用于指定外部数据封装不再拥有处理函数。

  须知

  使用外部数据封装但没有handler的外表不能访问。

- **VALIDATOR validator_function**

  为外部数据封装指定一个新的验证函数。

  须知

  根据新的验证器，外部数据封装器或依赖的服务器、用户映射或外部表的已经存在的选项是有可能是无效的。用户在使用外部数据封装之前需要保证这些选项是正确的。不过，ALTER FOREIGN DATA WRAPPER 命令中指定的任何选项都将使用新的验证函数检查。

- **NO VALIDATOR**

  用于指定外部数据封装不再有验证函数。

- **OPTIONS ([ADD | SET | DROP] option ['value'] [, ...])**

  修改外部数据封装的选项，选项名需满足唯一约束。ADD、SET和DROP为指定执行的操作，如果未指定，默认为ADD。使用外部数据封装验证函数时，名称和取值也会被验证。

相关链接

CREATE FOREIGN DATA WRAPPER，DROP FOREIGN DATA WRAPPER

1.13.7.12 ALTER FOREIGN TABLE

功能描述

修改外表的属性。

版权所有 © 华为云计算技术有限公司

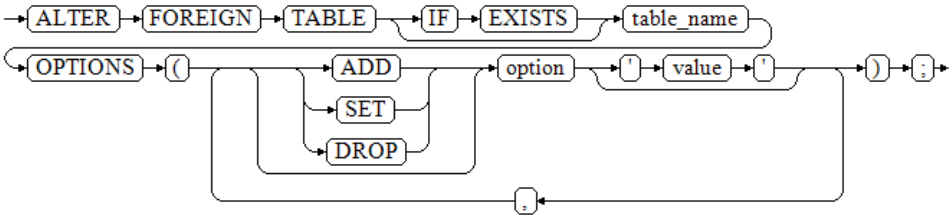
注意事项

OPTIONS中的敏感字段（如password、secret_access_key）在使用多层引号时，语义和不带引号的场景不同，因此不会被识别为敏感字段进行脱敏。

语法格式

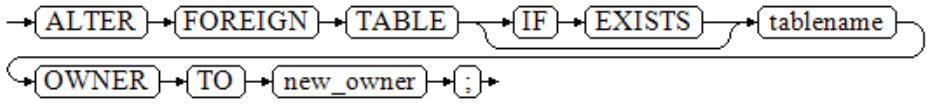
- 设置外表属性

```
ALTER FOREIGN TABLE [ IF EXISTS ]  table_name
    OPTIONS ( { [ ADD | SET | DROP ] option ['value']}[, ... ]);
```



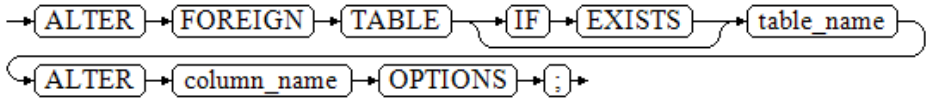
- 设置新的所有者

```
ALTER FOREIGN TABLE [ IF EXISTS ] tablename
    OWNER TO new_owner;
```



- 设置外表字段选项

```
ALTER FOREIGN TABLE [ IF EXISTS ] table_name
    ALTER column_name OPTIONS;
```



参数说明

- **table_name**

  需要修改的外表名称。

  取值范围：已存在的外表名。

- **OPTIONS ([ADD | SET | DROP] option ['value'] [, ...])**

  改变外表或外表字段的选项。ADD、SET和DROP为指定执行的操作。如果未指定，默认为SET。

  - option：需要修改的option名称。

    取值范围：请参见**CREATE FOREIGN TABLE**章节中的**OPTIONS ({ option_name ' value ' } [, ...])**。

  - value：option的新值。

- **new_owner**

  外表的新的所有者。

- **column_name**

  现有的列的名称。

  取值范围：字符串，要符合**标识符命名规范**。

版权所有 © 华为云计算技术有限公司

示例

```
--创建外部表。
gaussdb=# CREATE SCHEMA tpcds;
gaussdb=# CREATE FOREIGN TABLE tpcds.customer_ft
(
    c_customer_sk             integer               ,
    c_customer_id             char(16)              ,
    c_current_cdemo_sk        integer               ,
    c_current_hdemo_sk        integer               ,
    c_current_addr_sk         integer               ,
    c_first_shipto_date_sk    integer               ,
    c_first_sales_date_sk     integer               ,
    c_salutation              char(10)              ,
    c_first_name              char(20)              ,
    c_last_name               char(30)              ,
    c_preferred_cust_flag     char(1)               ,
    c_birth_day               integer               ,
    c_birth_month             integer               ,
    c_birth_year              integer               ,
    c_birth_country           varchar(20)           ,
    c_login                   char(13)              ,
    c_email_address           char(50)              ,
    c_last_review_date        char(10)
)
    SERVER gsmpp_server
    OPTIONS
(
    location 'gsfs://10.185.179.143:5000/customer1*.dat',
    FORMAT 'TEXT' ,
    DELIMITER '|',
    encoding 'utf8',
    mode 'Normal')
READ ONLY;

--修改外表属性，删除mode选项。
gaussdb=# ALTER FOREIGN TABLE tpcds.customer_ft options(drop mode);

--删除外部表。
gaussdb=# DROP FOREIGN TABLE tpcds.customer_ft;
gaussdb=# DROP SCHEMA tpcds CASCADE;
```

相关链接

CREATE FOREIGN TABLE，**DROP FOREIGN TABLE**

1.13.7.13 ALTER FUNCTION

功能描述

修改自定义函数的属性或重编译函数。

注意事项

- 只有函数的所有者或者被授予了函数ALTER权限的用户才能执行ALTER FUNCTION命令，系统管理员默认拥有该权限。针对所要修改属性的不同，还存在以下权限约束：
  - 如果函数中涉及对临时表相关的操作，则无法使用ALTER FUNCTION。
  - 修改函数的所有者或修改函数的模式，当前用户必须是该函数的所有者或者系统管理员，且该用户是新所有者角色的成员。
  - 只有系统管理员和初始化用户可以将函数的模式修改成public。

版权所有 © 华为云计算技术有限公司

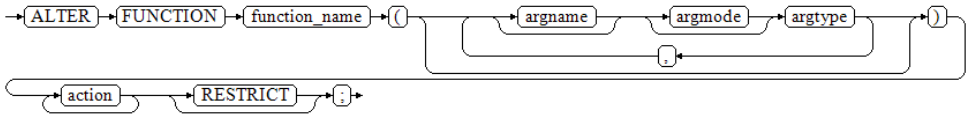
  - 修改函数的属性受GUC参数**using_danger_feature**的影响。

- 当打开三权分立时，对于定义者权限的函数，不允许任何角色修改函数的所有者。
- 当关闭三权分立时，对于定义者权限的函数，仅初始用户和系统管理员可以修改函数的所有者。
- 只有初始化用户才能修改函数的所有者为初始化用户。

语法格式

- 修改自定义函数的附加参数。

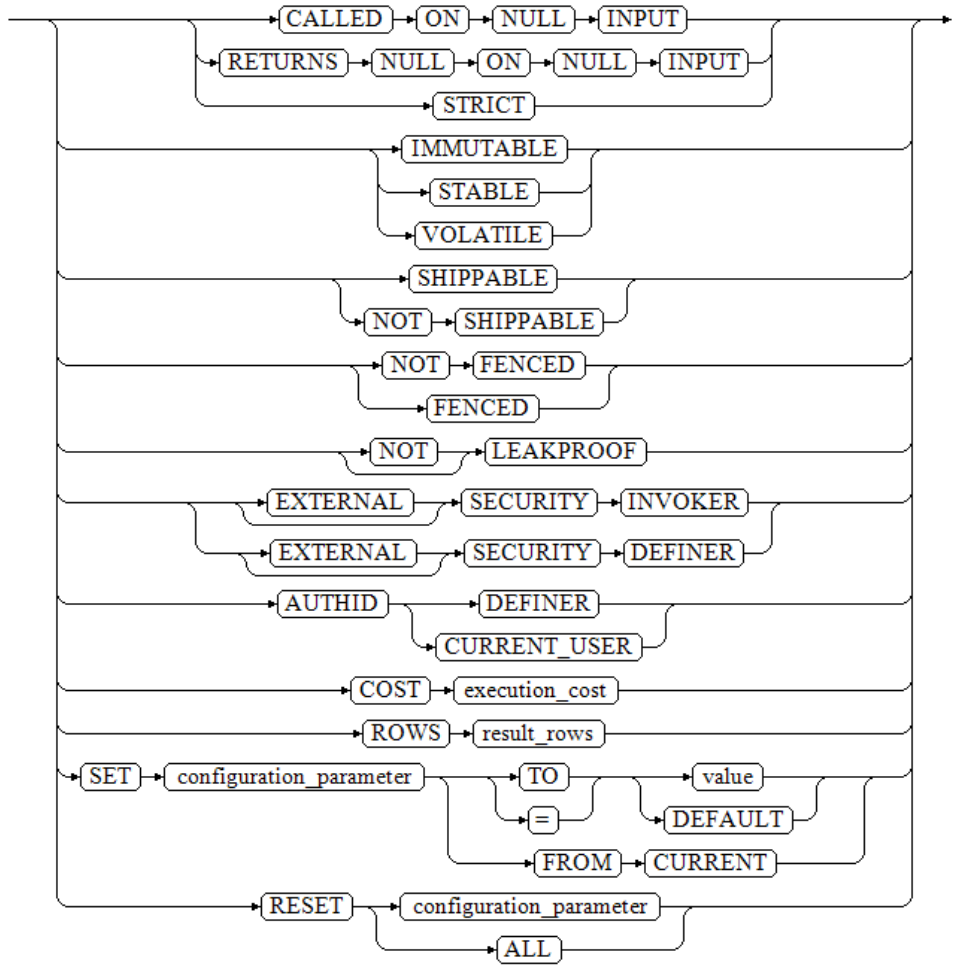
```
ALTER FUNCTION function_name ( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    action [ ... ] [ RESTRICT ];
```



其中附加参数action子句语法为：

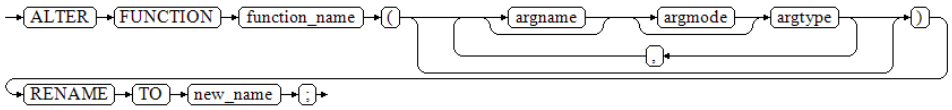
```
{{CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT}
 | {IMMUTABLE | STABLE | VOLATILE}
 | {SHIPPABLE | NOT SHIPPABLE}
 | {NOT FENCED | FENCED}
 | [ NOT ] LEAKPROOF
 | { [ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER }
 | AUTHID { DEFINER | CURRENT_USER }
 | COST execution_cost
 | ROWS result_rows
 | SET configuration_parameter { { TO | = } { value | DEFAULT }| FROM CURRENT}
 | RESET {configuration_parameter | ALL}}
```


版权所有 © 华为云计算技术有限公司



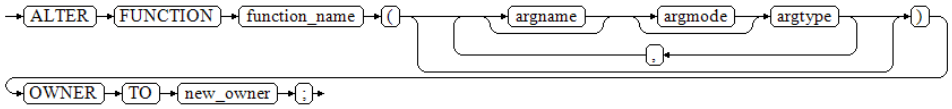
- 修改自定义函数的名称

```
ALTER FUNCTION function_name ( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    RENAME TO new_name;
```



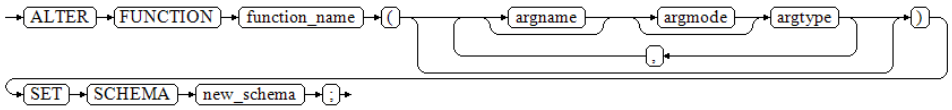
- 修改自定义函数的所有者

```
ALTER FUNCTION function_name ( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    OWNER TO new_owner;
```



- 修改自定义函数的模式

```
ALTER FUNCTION function_name ( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    SET SCHEMA new_schema;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **function_name**

  要修改的函数名称。

  取值范围：已存在的函数名。

- **argmode**

  标识该参数是输入或输出参数。

  取值范围：

  - IN：声明入参。
  - OUT：声明出参。
  - INOUT：声明出入参。
  - VARIADIC：声明数组类型的参数。

- **argname**

  参数名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **argtype**

  函数参数的类型。

  取值范围：有效的类型，请参见**数据类型**。

- **CALLED ON NULL INPUT**

  表明该函数的某些参数为NULL时，可以按照正常的方式调用。缺省时与指定此参数的作用相同。

- **RETURNS NULL ON NULL INPUT | STRICT**

  STRICT：用于指定当一个函数的某个参数为NULL时，则该函数返回NULL。如果声明了该参数，则如果存在NULL参数时不会执行该函数，而直接自动返回一个NULL结果。

  RETURNS NULL ON NULL INPUT和STRICT的功能相同。

- **IMMUTABLE**

  表示该函数在指定同样的参数值时总是返回同样的结果。

- **STABLE**

  表示该函数不能修改数据库，对相同参数值，在同一次表扫描里，该函数的返回值不变，但是返回值可能在不同SQL语句之间变化。

- **VOLATILE**

  表示该函数值可以在一次表扫描内改变，不会做任何优化。

- **SHIPPABLE | NOT SHIPPABLE**

  表示该函数是否可以下推到DN上执行。

  - 对于IMMUTABLE类型的函数，函数始终可以下推到DN上执行。
  - 对于STABLE或VOLATILE类型的函数，仅当函数的属性是SHIPPABLE的时候，函数可以下推到DN执行。

- **LEAKPROOF**

  表示该函数没有副作用，指出参数只包括返回值。LEAKPROOF只能由系统管理员设置。

- **EXTERNAL**

版权所有 © 华为云计算技术有限公司

（可选）目的是与SQL兼容，该关键字适合于所有函数，而不仅是外部函数

- **SECURITY INVOKER | AUTHID CURRENT_USER**
  表明该函数将以调用函数的用户的权限执行。缺省时与指定此参数的作用相同。
  SECURITY INVOKER和AUTHID CURRENT_USER的功能相同。
- **SECURITY DEFINER | AUTHID DEFINER**
  声明该函数将以创建函数的用户的权限执行。
  AUTHID DEFINER和SECURITY DEFINER的功能相同。
- **COST execution_cost**
  用于估计函数的执行成本。
  execution_cost以cpu_operator_cost为单位。
  取值范围：正数。
- **ROWS result_rows**
  估计函数返回的行数。用于函数返回的是一个集合。
  取值范围：正数，默认值是1000。
- **configuration_parameter**
  - **value**
    将指定的数据库会话参数值设置为具体定的值。
    取值范围：字符串。
    - DEFAULT：在新的会话中使用系统的缺省设置
    - OFF：关闭设置。
    - RESET：在新的会话中使用系统的缺省设置
    - 用户指定的值：需要满足修改参数的取值限制。
  - **FROM CURRENT**
    取当前会话中的值设置为configuration_parameter的值。
- **new_name**
  函数的新名称。要修改函数的所属模式，必须拥有新模式的CREATE权限。
  取值范围：字符串，要符合**标识符命名规范**。
- **new_owner**
  函数的新所有者。要修改函数的所有者，新所有者必须拥有该函数所属模式的CREATE权限。需要注意的是：仅初始化用户才可以将函数的所有者设置为初始化用户。
  取值范围：已存在的用户角色。
- **new_schema**
  函数的新模式。
  取值范围：已存在的模式。

示例

修改函数示例

```
--创建函数。
gaussdb=# CREATE OR REPLACE FUNCTION test_func(a int) RETURN int
```


版权所有 © 华为云计算技术有限公司

```
IS
    proc_var int;
BEGIN
    proc_var := a;
    return 1;
END;
/

--将函数test_func(a int)的名称修改为test_func_tk(a int)。
gaussdb=# ALTER FUNCTION test_func(a int) RENAME TO test_func_tk;

--创建jim用户。
gaussdb=# CREATE USER jim PASSWORD '********';

--新建模式test。
gaussdb=# CREATE SCHEMA test;

--将函数的所有者改为jim。
gaussdb=# ALTER FUNCTION test_func_tk(a int) OWNER TO jim;

--将函数模式改为test。
gaussdb=# ALTER FUNCTION test_func_tk(a int) SET SCHEMA test;

--删除函数。
gaussdb=# DROP FUNCTION test.test_func_tk(a int);

--删除jim用户。
gaussdb=# DROP USER jim;

--删除SCHEMA。
gaussdb=# DROP SCHEMA test;
```

相关链接

CREATE FUNCTION，**DROP FUNCTION**

1.13.7.14 ALTER GLOBAL CONFIGURATION

功能描述

新增、修改系统表gs_global_config的key-value值。如果修改的参数已经存在，则修改；如果不存在，则新增。

注意事项

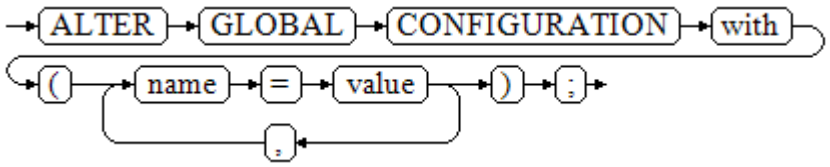
- 仅支持数据库初始用户运行此命令。
- 参数名称不能为weak_password、undostoragetype。
- 参数名称为扩容重分布预留参数时，需设置GUC参数enable_cluster_resize为on，或者使用扩容工具进行修改。涉及参数名称：lockwait_timeout、lockwait_interval、trylock_threshold、enable_cancel、last_catchup_threshold、catchup_times、write_error_mode、catchup_query_dop、parallel_catchup_threshold、parallel_reindex_jobs、redis_retry_times、redis_retry_interval、redis_bucket_expansion、redis_bucket_batchsize、redis_bucket_log_level、redis_bucket_lockhold_timeout、redis_bucket_timeout_penalty、redis_bucket_max_lockwait_timeout_retry、redis_bucket_max_lockhold_timeout_retry、redis_bucket_strategy_history_capacity、redis_bucket_strategy_recent_threshold、redis_bucket_workload_rule、redis_bucket_lsn_rule、redis_bucket_tx_bucket_min、

版权所有 © 华为云计算技术有限公司

redis_bucket_tx_bucket_max、redis_max_expected_tps、redis_bucket_sender_concurrency、redis_bucket_receiver_concurrency、redis_bucket_parse_redo_num、redis_bucket_workers_per_paser、redis_bucket_keep_bucketxlogs、redis_bucket_reindex_cbi_offline以及redis_bucket_reindex_parallel_workers。

语法格式

```
ALTER GLOBAL CONFIGURATION with ( { name = value } [, ...] );
```



参数说明

- **name**

  参数名称，text类型，不能为weak_password、undostoragetype，除此之外没有限制。

  说明

  参数名称为扩容重分布预留参数时，需设置GUC参数enable_cluster_resize为on，或者使用扩容工具进行修改。

- **value**

  参数值，text类型。

示例

```
--插入内容。
gaussdb=# ALTER GLOBAL CONFIGURATION with(redis_is_ok = true);

--查询。
gaussdb=# SELECT * FROM gs_global_config;
      name       | value
-----------------+-------
 buckets_len     | 16384
 undostoragetype | page
 redis_is_ok     | true
(3 rows)

--修改内容。
gaussdb=# ALTER GLOBAL CONFIGURATION with(redis_is_ok = false);

--查询。
gaussdb=# SELECT * FROM gs_global_config;
      name       | value
-----------------+-------
 buckets_len     | 16384
 undostoragetype | page
 redis_is_ok     | false
(3 rows)

--删除内容。
gaussdb=# DROP GLOBAL CONFIGURATION redis_is_ok;

--查询。
gaussdb=# SELECT * FROM gs_global_config;
      name       | value
```


版权所有 © 华为云计算技术有限公司

```
-----------------+-------
 buckets_len     | 16384
 undostoragetype | page
(2 rows)
```

相关链接

DROP GLOBAL CONFIGURATION

1.13.7.15 ALTER GROUP

功能描述

修改一个用户组的属性。

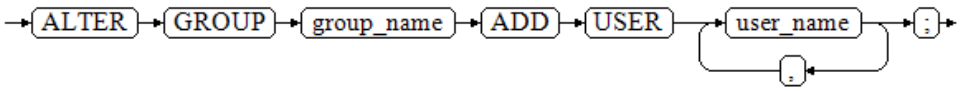
注意事项

- ALTER GROUP非SQL标准语法，不推荐用户使用。
- 其中两个子句（ADD USER，DROP USER）向用户组增加或删除用户（任何用户都可以是“用户”或者“用户组”）。这两个子句实际等效于将用户或者角色的权限授权或者回收给其他用户或角色，因此建议用GRANT或者REVOKE替代。
- RENAME TO子句修改用户组的名称，等效于ALTER ROLE重命名角色。

语法格式

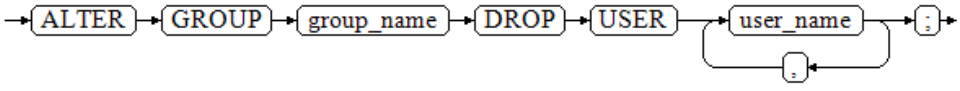
- 向用户组中添加用户

```
ALTER GROUP group_name
    ADD USER user_name [, ... ];
```



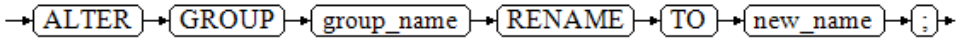
- 从用户组中删除用户

```
ALTER GROUP group_name
    DROP USER user_name [, ... ];
```



- 修改用户组的名称

```
ALTER GROUP group_name
    RENAME TO new_name;
```



参数说明

- **user_name**

  现有角色名称。

  取值范围：已存在的角色名，如果角色名称中包含大写字母则需要使用双引号进行包裹。

- **group_name**

  现有用户组名称。

版权所有 © 华为云计算技术有限公司

取值范围：已存在的用户组名称，如果用户组名称中包含大写字母则需要使用双引号进行包裹。

- **new_name**

  新角色名称。

  取值范围：字符串，要符合**标识符命名规范**，且最多为63个字符。若超过63个字符，数据库会截断并保留前63个字符当作角色名称。当角色名中包含大写字母时，数据库会自动转换为小写字母，如果需要创建包含大写字母的角色名，则需要使用双引号进行包裹。

示例

- 重命名用户组。

```
--创建用户test。
gaussdb=# CREATE ROLE test PASSWORD '********';

--修改用户名，等效于ALTER ROLE RENAME。
gaussdb=# ALTER GROUP test RENAME TO tu_a1;
```

- 向用户组中添加或者删除用户。

```
--创建用户tu_a2、tu_a3。
gaussdb=# CREATE ROLE tu_a2 PASSWORD '********';
gaussdb=# CREATE ROLE tu_a3 PASSWORD '********';

--向用户组tu_a1中添加用户tu_a2。
gaussdb=# ALTER GROUP tu_a1 ADD USER tu_a2;

--如上SQL等效于GRANT语句。
gaussdb=# GRANT tu_a1 TO tu_a3;

--查询。
gaussdb=# SELECT groname, grolist FROM pg_group WHERE groname = 'tu_a1';
 groname |    grolist
---------+---------------
 tu_a1   | {25590,25593}
(1 row)
gaussdb=# SELECT rolname, oid FROM gs_roles WHERE oid IN (25590,25593);
 rolname |  oid
---------+-------
 tu_a2   | 25590
 tu_a3   | 25593
(2 rows)

--删除。
gaussdb=# DROP ROLE tu_a1,tu_a2,tu_a3;
```

相关链接

CREATE GROUP，**DROP GROUP**，**ALTER ROLE**

1.13.7.16 ALTER INDEX

功能描述

ALTER INDEX用于修改现有索引的定义。

注意事项

- 索引的所有者、拥有索引所在表的INDEX权限的用户或者被授予了ALTER ANY INDEX权限的用户有权限执行此命令，当三权分立开关关闭时，系统管理员默认拥有此权限。

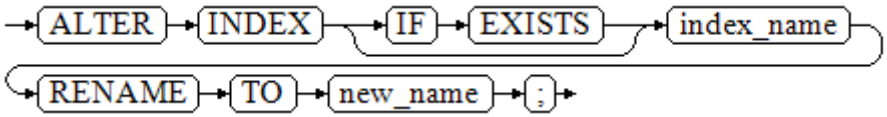
版权所有 © 华为云计算技术有限公司

- 请勿在同一基表上保持大量的不可见索引，否则可能会对INSERT、UPDATE、DELETE等DML操作的性能产生影响。
- rowid系统列索引仅支持修改索引所属表空间及重建索引或索引分区。

语法格式

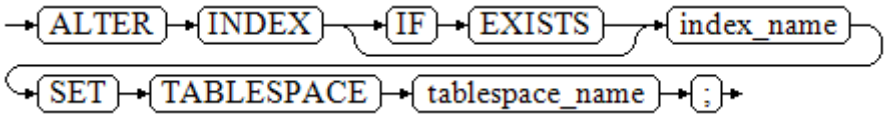
- 重命名表索引的名称。

```
ALTER INDEX [ IF EXISTS ] index_name
    RENAME TO new_name;
```



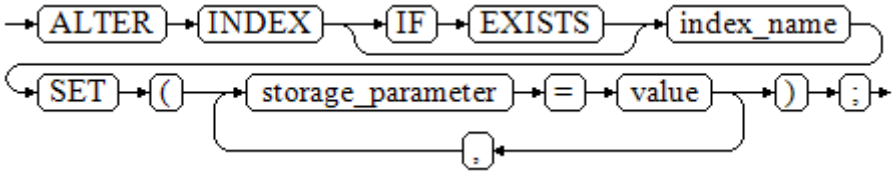
- 修改表索引的所属空间。

```
ALTER INDEX [ IF EXISTS ] index_name
    SET TABLESPACE tablespace_name;
```



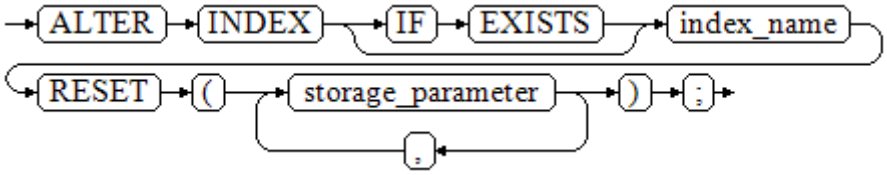
- 修改表索引的存储参数。

```
ALTER INDEX [ IF EXISTS ] index_name
    SET ( {storage_parameter = value} [, ... ] );
```



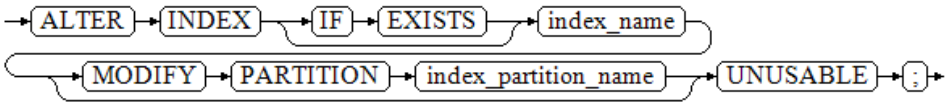
- 重置表索引的存储参数。

```
ALTER INDEX [ IF EXISTS ] index_name
    RESET ( storage_parameter [, ... ] );
```



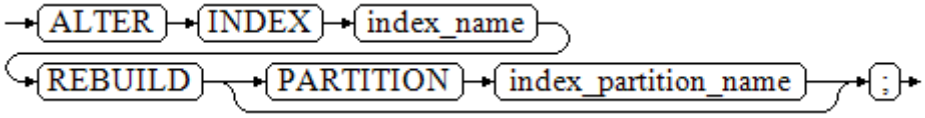
- 设置表索引或索引分区不可用。

```
ALTER INDEX [ IF EXISTS ] index_name
    [ MODIFY PARTITION index_partition_name ] UNUSABLE;
```



- 重建表索引或索引分区。

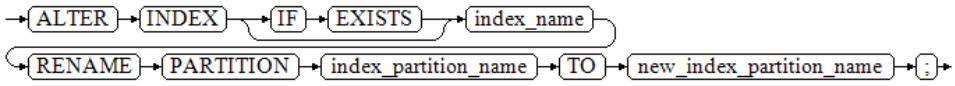
```
ALTER INDEX index_name
    REBUILD [ PARTITION index_partition_name ];
```



版权所有 © 华为云计算技术有限公司

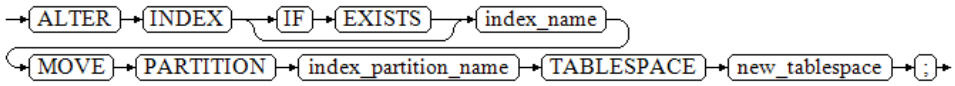
- 重命名索引分区。

```
ALTER INDEX [ IF EXISTS ] index_name
    RENAME PARTITION index_partition_name TO new_index_partition_name;
```



- 修改索引分区的所属表空间。

```
ALTER INDEX [ IF EXISTS ] index_name
    MOVE PARTITION index_partition_name TABLESPACE new_tablespace;
```



- 设置分布式全局二级索引状态。

```
ALTER INDEX [ IF EXISTS ] index_name GSIVALID;
```

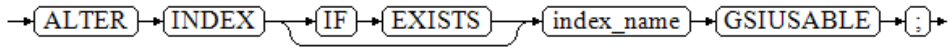
```
ALTER INDEX [ IF EXISTS ] index_name GSIREADY;
```

说明

此语法主要用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，修改全局二级索引状态，用户不建议使用。如使用，会返回错误提示信息。

- 设置分布式全局二级索引可用。

```
ALTER INDEX [ IF EXISTS ] index_name GSIUSABLE;
```

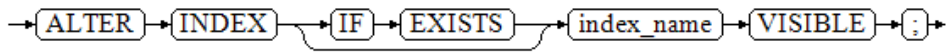


说明

此语法主要用于VACUUM FULL功能内部调用，修改全局二级索引状态，用户不建议使用，否则可能导致索引数据与表数据不一致。如需启用UNUSABLE状态的GSI，建议使用REINDEX INDEX语法重新创建GSI。

- 用于设置索引状态为可见。

```
ALTER INDEX [ IF EXISTS ] index_name VISIBLE;
```

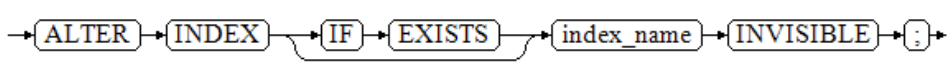


说明

- 此语法仅支持在ORA兼容性数据库下（即sql_compatibility = 'ORA'）设置，在其他数据库兼容模式下不支持。
- 当disable_keyword_options参数设置为“visible”时，VISIBLE关键字不支持使用。
- 升级未提交阶段，不支持使用此语法。

- 用于设置索引状态为不可见。

```
ALTER INDEX [ IF EXISTS ] index_name INVISIBLE;
```



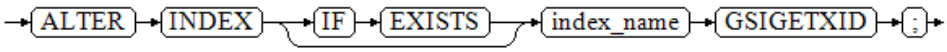
版权所有 © 华为云计算技术有限公司

说明

- 此语法仅支持在ORA兼容性数据库下（即sql_compatibility = 'ORA'）设置，在其他数据库兼容模式下不支持。
- 当disable_keyword_options参数设置为“invisible”时，INVISIBLE关键字不支持使用。
- 升级未提交阶段，不支持使用此语法。
- 备机读场景下，设置索引状态为不可见后，查询语句的执行计划可能会发生变化，备机查询性能可能会受到影响。

- 用于将下一个未分配的xid和csn记录在各节点内的临时用户表的reloption中，以备合并临时表时的剪枝。

```
ALTER INDEX [ IF EXISTS ] index_name GSIGETXID;
```



说明

此语法主要用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，用户不建议使用。如使用，会返回错误提示信息。

- 用于合并分布式全局二级索引与临时用户表。

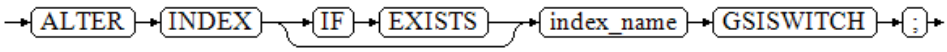
```
ALTER INDEX [ IF EXISTS ] index_name GSIMERGE;
```

说明

此语法主要用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，用户不建议使用。如使用，会返回错误提示信息。

- 用于在线重建分布式全局二级索引中切换新旧分布式全局二级索引。

```
ALTER INDEX [ IF EXISTS ] index_name GSISWITCH;
```



说明

此语法主要用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，用户不建议使用。如使用，会返回错误提示信息。

参数说明

- **index_name**

  要修改的索引名。

- **IF EXISTS**

  如果指定的索引不存在，则发出一个notice而不是error。

- **RENAME TO new_name**

  只改变索引的名称。对存储的数据没有影响。

  - **new_name**

    新的索引名。

    取值范围：字符串，要符合**标识符命名规范**。

- **SET TABLESPACE tablespace_name**

版权所有 © 华为云计算技术有限公司

改变索引的表空间为指定表空间，并且把索引相关的数据文件移动到新的表空间里。

- **tablespace_name**

  表空间的名称。

  取值范围：已存在的表空间。

- **SET ({storage_parameter = value} [, ...])**

  改变索引的一个或多个索引方法特定的存储参数。需要注意的是索引内容不会被这个命令立即修改，根据参数的不同，可能需要使用REINDEX重建索引来获得期望的效果。

  - **storage_parameter**

    索引方法特定的参数名。ACTIVE_PAGES表示索引的页面数量，可能比实际的物理文件页面少，可以用于优化器调优。目前只对ustore的分区表local索引生效，且会被vacuum、analyze更新（包括auto vacuum）。不建议用户手动设置该参数，该参数在分布式下无效。对GSI执行ALTER时，对INTERNAL_MASK、APPEND_MODE_INTERNAL、WAIT_CLEAN_GPI、PARTITION_DDL_FLAG、COLLATE、CROSSBUCKET和DEDUPLICATION参数不支持，行为同UBtree，支持FILLFACTOR、INDEXSPLIT参数，ACTIVE_PAGES和WAIT_CLEAN_CBI 不适用于GSI。

  - **value**

    索引方法特定的存储参数的新值。根据参数的不同，这可能是一个数字或单词。

- **RESET ({ storage_parameter } [, ...])**

  重置索引的一个或多个索引方法特定的存储参数为缺省值。与SET一样，可能需要使用REINDEX来完全更新索引。

- **[MODIFY PARTITION index_partition_name] UNUSABLE**

  用于设置表或者索引分区上的索引不可用。

  说明

  当指定的索引为唯一索引时，再次对表进行插入和更新操作时，会受到GUC参数enable_unique_checking_of_unusable_index的控制。

- **REBUILD [PARTITION index_partition_name]**

  用于重建表或者索引分区上的索引。重建索引时，若索引带有lpi_parallel_method选项，取值为PARTITION且表的parallel_workers选项大于0时，不支持对该索引并行重建；无该选项或选项取值为AUTO时，并行重建时会默认走页面级并行重建索引。具体请参见**CREATE INDEX**章节中的**LPI_PARALLEL_METHOD**。

- **RENAME PARTITION index_partition_name TO new_index_partition_name**

  用于重命名索引分区。

- **MOVE PARTITION index_partition_name TABLESPACE new_tablespace**

  用于修改索引分区的所属表空间。

- **new_index_partition_name**

  新索引分区名。

- **index_partition_name**

  索引分区名。

- **new_tablespace**

版权所有 © 华为云计算技术有限公司

新表空间。

- **GSIVALID** | GSIREADY

  用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，修改分布式全局二级索引状态。

- **GSIUSABLE**

  此语法主要用于VACUUM FULL功能内部调用，修改分布式全局二级索引状态。

- **VISIBLE**

  用于设置索引状态为可见。

- **INVISIBLE**

  用于设置索引状态为不可见。

- **GSIGETXID**

  用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，将下一个未分配的xid和csn记录在各节点内的临时用户表的reloption中，以备合并临时表时的剪枝。

- **GSIMERGE**

  用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，合并分布式全局二级索引与临时用户表。

- **GSISWITCH**

  用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，在线重建分布式全局二级索引中切换新旧分布式全局二级索引。

示例

- 重命名索引。

```
--创建test1表并为其创建索引。
gaussdb=# CREATE TABLE test1(col1 int, col2 int);
gaussdb=# CREATE INDEX aa ON test1(col1);

--将索引aa重命名为idx_test1_col1。
gaussdb=# ALTER INDEX aa RENAME TO idx_test1_col1;

--查询test1表上的索引信息。
gaussdb=# SELECT tablename,indexname,tablespace FROM pg_indexes WHERE tablename = 'test1';
 tablename |   indexname    | tablespace
-----------+----------------+------------
 test1     | idx_test1_col1 |
(1 row)
```

- 修改索引所属表空间。

```
--创建表空间tbs_index1。
gaussdb=# CREATE TABLESPACE tbs_index1 RELATIVE LOCATION 'tablespace1/tbs_index1';

--修改索引idx_test1_col1的所属表空间为tbs_index1。
gaussdb=# ALTER INDEX IF EXISTS idx_test1_col1 SET TABLESPACE tbs_index1;

--查询test1表上的索引信息。
gaussdb=# SELECT tablename,indexname,tablespace FROM pg_indexes WHERE tablename = 'test1';
 tablename |   indexname    | tablespace
-----------+----------------+------------
 test1     | idx_test1_col1 | tbs_index1
(1 row)
```

- 修改与重置索引存储参数。

```
--查看索引idx_test1_col1详细信息。
gaussdb=# \di idx_test1_col1                        List of relations
 Schema |      Name      | Type  | Owner | Table | Storage
```


版权所有 © 华为云计算技术有限公司

```
--------+----------------+-------+-------+-------+---------
 public | idx_test1_col1 | index | omm   | test1 |
(1 row)

--修改索引idx_test1_col1 的填充因子。
gaussdb=# ALTER INDEX IF EXISTS idx_test1_col1 SET (FILLFACTOR = 70);
--查看索引idx_test1_col1详细信息。
gaussdb=# \di idx_test1_col1
                      List of relations
 Schema |      Name      | Type  | Owner | Table |     Storage
--------+----------------+-------+-------+-------+-----------------
 public | idx_test1_col1 | index | omm   | test1 | {fillfactor=70}
(1 row)
--重置索引idx_test1_col1 的存储参数。
gaussdb=# ALTER INDEX IF EXISTS idx_test1_col1 RESET (FILLFACTOR);
--查看索引idx_test1_col1详细信息。
gaussdb=# \di idx_test1_col1
                  List of relations
 Schema |      Name      | Type  | Owner | Table | Storage
--------+----------------+-------+-------+-------+---------
 public | idx_test1_col1 | index | omm   | test1 |
(1 row)
```

- 修改索引可用性。

```
--设置索引idx_test1_col1不可用。
gaussdb=# ALTER INDEX IF EXISTS idx_test1_col1 UNUSABLE;
--查看索引idx_test1_col1的可用性。
gaussdb=# SELECT indisusable FROM pg_index WHERE indexrelid = 'idx_test1_col1'::regclass;
 indisusable
-------------
 f
(1 row)

--重建索引idx_test1_col1。
gaussdb=# ALTER INDEX idx_test1_col1 REBUILD;
--查看索引idx_test1_col1的可用性。
gaussdb=# SELECT indisusable FROM pg_index WHERE indexrelid = 'idx_test1_col1'::regclass;
 indisusable
-------------
 t
(1 row)

--删除。
gaussdb=# DROP INDEX idx_test1_col1;
gaussdb=# DROP TABLE test1;
gaussdb=# DROP TABLESPACE tbs_index1;
```

- 重命名索引分区。

```
--创建分区表test2。
gaussdb=# CREATE TABLE test2(col1 int, col2 int) PARTITION BY RANGE (col1)(
    PARTITION p1 VALUES LESS THAN (100),
    PARTITION p2 VALUES LESS THAN (200)
);
--创建分区索引。
gaussdb=# CREATE INDEX idx_test2_col1 ON test2(col1) LOCAL(
    PARTITION p1,
    PARTITION p2
);
--重命名索引分区。
gaussdb=# ALTER INDEX idx_test2_col1 RENAME PARTITION p1 TO p1_test2_idx;
gaussdb=# ALTER INDEX idx_test2_col1 RENAME PARTITION p2 TO p2_test2_idx;

--查询索引idx_test2_col1分区的名称。
gaussdb=# SELECT relname FROM pg_partition WHERE parentid = 'idx_test2_col1'::regclass;
   relname
--------------
 p1_test2_idx
 p2_test2_idx
(2 rows)
```


版权所有 © 华为云计算技术有限公司

- 修改索引分区的所属表空间。

```
--创建表空间tbs_index2与tbs_index3。
gaussdb=# CREATE TABLESPACE tbs_index2 RELATIVE LOCATION 'tablespace1/tbs_index2';
gaussdb=# CREATE TABLESPACE tbs_index3 RELATIVE LOCATION 'tablespace1/tbs_index3';

--修改索引idx_test2_col1分区的所属表空间。
gaussdb=# ALTER INDEX idx_test2_col1 MOVE PARTITION p1_test2_idx TABLESPACE tbs_index2;
gaussdb=# ALTER INDEX idx_test2_col1 MOVE PARTITION p2_test2_idx TABLESPACE tbs_index3;

--查询索引idx_test2_col1分区的所属表空间。
gaussdb=# SELECT t1.relname index_name,
       t2.spcname tablespace_name
FROM pg_partition t1, pg_tablespace t2
WHERE t1.parentid = 'idx_test2_col1'::regclass AND
      t1.reltablespace = t2.oid;
  index_name  | tablespace_name
--------------+-----------------
 p1_test2_idx | tbs_index2
 p2_test2_idx | tbs_index3
(2 rows)

--删除。
gaussdb=# DROP INDEX idx_test2_col1;
gaussdb=# DROP TABLE test2;
gaussdb=# DROP TABLESPACE tbs_index2;
gaussdb=# DROP TABLESPACE tbs_index3;
```

相关链接

CREATE INDEX，**DROP INDEX**，**REINDEX**

1.13.7.17 ALTER LANGUAGE

功能描述

修改一个过程语言的定义。

注意事项

当前形态暂不支持使用该语法。

1.13.7.18 ALTER MASKING POLICY

功能描述

修改脱敏策略。

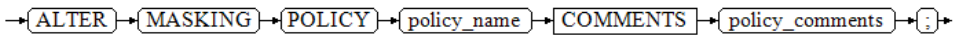
注意事项

- 只有poladmin，sysadmin或初始用户才能执行此操作。
- 需要开启安全策略开关，即设置GUC参数“enable_security_policy”=“on”，脱敏策略才可以生效。

语法格式

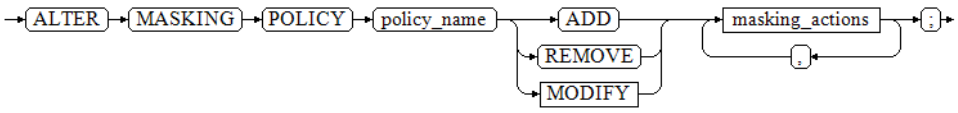
- 修改策略描述：

```
ALTER MASKING POLICY policy_name COMMENTS policy_comments;
```

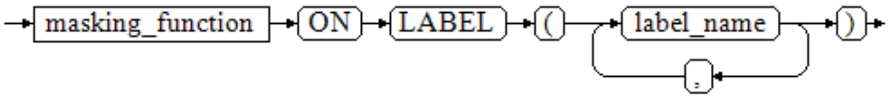


版权所有 © 华为云计算技术有限公司

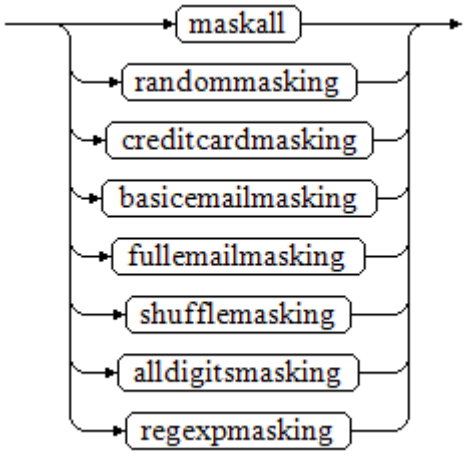
- 修改脱敏方式：
  ALTER MASKING POLICY policy_name { ADD | REMOVE | MODIFY } masking_actions[, ...];

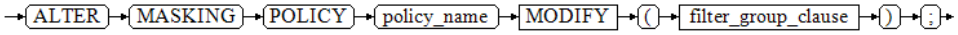
  - 其中masking_actions:
    masking_function ON LABEL(label_name[, ...])

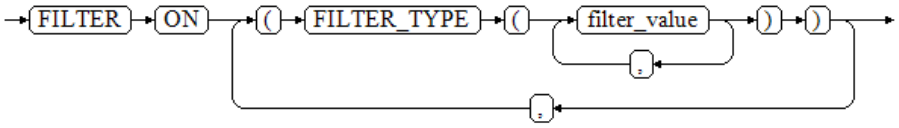
  - 其中masking_function：
    { maskall | randommasking | creditcardmasking | basicemailmasking | fullemailmasking | shufflemasking | alldigitsmasking | regexpmasking }

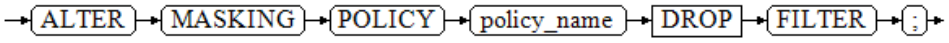
- 修改脱敏策略生效场景：
  ALTER MASKING POLICY policy_name MODIFY (filter_group_clause);

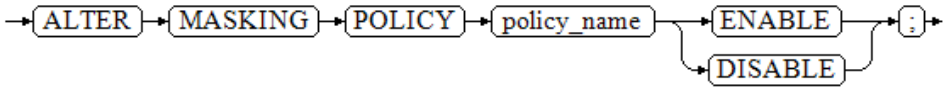
  - 其中filter_group_clause：
    FILTER ON { (FILTER_TYPE (filter_value [, ...])) [, ...] }

- 移除脱敏策略生效场景，使策略对所用场景生效：
  ALTER MASKING POLICY policy_name DROP FILTER;

- 修改脱敏策略开启/关闭：
  ALTER MASKING POLICY policy_name {ENABLE | DISABLE};

版权所有 © 华为云计算技术有限公司

参数说明

- **policy_name**
  脱敏策略名称，需要唯一，不可重复。
  取值范围：字符串，要符合**标识符命名规范**。
- **policy_comments**
  需要为脱敏策略添加或修改的描述信息。
- **masking_function**
  指的是预置的八种脱敏方式或者用户自定义的函数，支持模式。
  maskall不是预置函数，直接写入在代码中，不支持\df展示。
  预置的脱敏方式如下：
  ```
  { maskall | randommasking | creditcardmasking | basicemailmasking | fullemailmasking |
  shufflemasking | alldigitsmasking | regexpmasking }
  ```
- **label_name**
  资源标签名称。
- **FILTER_TYPE**
  指定脱敏策略的过滤信息，过滤类型包括：IP、ROLES、APP。
- **filter_value**
  指具体过滤信息内容，例如具体的IP，具体的APP名称，具体的用户名。
- **ENABLE|DISABLE**
  可以打开或关闭脱敏策略。若不指定ENABLE|DISABLE，语句默认为ENABLE。

示例

- 修改策略描述：
  ```
  --创建一个表tb_for_masking。
  gaussdb=# CREATE TABLE tb_for_masking(idx int, col1 text, col2 text, col3 text, col4 text, col5 text,
  col6 text, col7 text,col8 text);

  --向表tb_for_masking插入数据。
  gaussdb=# INSERT INTO tb_for_masking VALUES(1, '9876543210', 'usr321usr', 'abc@huawei.com',
  'abc@huawei.com', '1234-4567-7890-0123', 'abcdef 123456 ui 323 jsfd321 j3k2l3',
  '4880-9898-4545-2525', 'this is a llt case');

  --查看数据。
  gaussdb=# SELECT * FROM tb_for_masking;
   idx |    col1    |    col2    |      col3      |      col4      |       col5        |              col6
       |        col7        |       col8
  -----+------------+------------+----------------+----------------+-------------------+----------------------
  +------------------------------
  ------+--------------------+--------------------
     1 | 9876543210 | usr321usr | abc@huawei.com | abc@huawei.com | 1234-4567-7890-0123 | abcdef
  123456 ui 323 jsfd321 j
  3k2l3 | 4880-9898-4545-2525 | this is a llt case
  (1 row)

  --创建资源标签标记敏感列col1。
  gaussdb=# CREATE RESOURCE LABEL mask_lb1 ADD COLUMN(tb_for_masking.col1);

  --创建一个名为maskpol1的脱敏策略。
  gaussdb=# CREATE MASKING POLICY maskpol1 maskall ON LABEL(mask_lb1);

  --为脱敏策略maskpol1添加描述。
  gaussdb=# ALTER MASKING POLICY maskpol1 COMMENTS 'masking policy for tb_for_masking.col1';

  --查看脱敏策略maskpol1的描述。
  ```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM GS_MASKING_POLICY;
 polname  |             polcomments              |         modifydate         | polenabled
----------+--------------------------------------+----------------------------+------------
 maskpol1 | masking policy for tb_for_masking.col1 | 2023-11-07 16:38:31.607374 | t
(1 row)
```

- 修改脱敏策略：

```
--创建资源标签标记敏感列col2。
gaussdb=# CREATE RESOURCE LABEL mask_lb2 ADD COLUMN(tb_for_masking.col2);

--脱敏策略maskpol1新增randommasking的脱敏方式。
gaussdb=# ALTER MASKING POLICY maskpol1 ADD randommasking ON LABEL(mask_lb2);

--访问tb_for_masking表，col2列触发脱敏策略。
gaussdb=# SELECT col2 FROM tb_for_masking;
    col2
------------
 27e8da66cc
(1 row)

--脱敏策略maskpol1移除randommasking的脱敏方式。
gaussdb=# ALTER MASKING POLICY maskpol1 REMOVE randommasking ON LABEL(mask_lb2);

--访问表tb_for_masking，col2列的数据没有脱敏，说明脱敏方式randommasking失效。
gaussdb=# SELECT col2 FROM tb_for_masking;
   col2
-----------
 usr321usr
(1 row)

--脱敏策略maskpol1修改为randommasking的脱敏方式。
gaussdb=# ALTER MASKING POLICY maskpol1 MODIFY randommasking ON LABEL(mask_lb1);

--访问tb_for_masking表，col1列触发脱敏策略。
gaussdb=# SELECT col1 FROM tb_for_masking;
    col1
------------
 5a03debac1
(1 row)
```

- 修改脱敏策略生效场景：

```
--创建dev_mask和bob_mask用户。
gaussdb=# CREATE USER dev_mask PASSWORD '*********';
gaussdb=# CREATE USER bob_mask PASSWORD '*********';

--创建资源标签标记敏感列col8。
gaussdb=# CREATE RESOURCE LABEL mask_lb8 ADD COLUMN(tb_for_masking.col8);

--创建一个名为maskpol8的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol8 randommasking ON LABEL(mask_lb8) FILTER ON
ROLES(dev_mask, bob_mask), APP(gsql), IP('172.31.17.160', '127.0.0.0/24');

--修改脱敏策略maskpol8的过滤信息ROLES。
gaussdb=# ALTER MASKING POLICY maskpol1 MODIFY (FILTER ON ROLES(dev_mask));

--使用dev_mask用户查看tb_for_masking。
gaussdb=# GRANT ALL PRIVILEGES TO dev_mask;

--访问tb_for_masking表，col8列触发脱敏策略。
gaussdb=# SELECT col8 FROM tb_for_masking;
       col8
--------------------
 f134e06ef528013b46
(1 row)
```

- 移除脱敏策略生效场景，使策略对所用场景生效：

```
gaussdb=# ALTER MASKING POLICY maskpol1 DROP FILTER;
```

- 禁用脱敏策略：

```
--禁用脱敏策略maskpol1。
gaussdb=# ALTER MASKING POLICY maskpol1 DISABLE;
```

 版权所有 © 华为云计算技术有限公司

```
--查看脱敏策略maskpol1的状态，polenabled字段的值为f，说明该脱敏策略禁用成功。
gaussdb=# SELECT * FROM GS_MASKING_POLICY;
 polname  | polcomments |         modifydate         | polenabled
----------+-------------+----------------------------+------------
 maskpol1 |             | 2023-11-07 17:22:54.594111 | f
```

- 删除数据：

```
--删除脱敏策略。
gaussdb=# DROP MASKING POLICY maskpol1, maskpol8;

--删除资源标签。
gaussdb=# DROP RESOURCE LABEL mask_lb1, mask_lb2, mask_lb8;

--删除表tb_for_masking。
gaussdb=# DROP TABLE tb_for_masking;

--删除用户dev_mask和bob_mask。
gaussdb=# DROP USER dev_mask, bob_mask;
```

相关链接

CREATE MASKING POLICY，**DROP MASKING POLICY**

1.13.7.19 ALTER MATERIALIZED VIEW

功能描述

更改一个现有物化视图的多个辅助属性。

可用于ALTER MATERIALIZED VIEW的语句形式和动作是**ALTER TABLE**的一个子集，并且在用于物化视图时具有相同的含义。具体请参见**ALTER TABLE**。

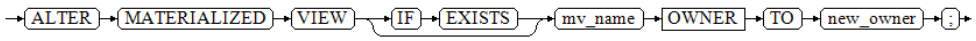
注意事项

- 只有物化视图的所有者可以执行ALTER MATERIALIZED VIEW命令，系统管理员默认拥有此权限。
- 不支持更改物化视图结构。

语法格式

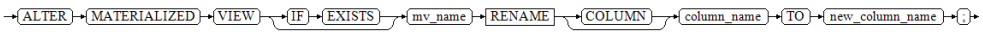
- 修改物化视图的所有者。

```
ALTER MATERIALIZED VIEW [ IF EXISTS ] mv_name
    OWNER TO new_owner;
```



- 重命名物化视图的列。

```
ALTER MATERIALIZED VIEW [ IF EXISTS ] mv_name
    RENAME [ COLUMN ] column_name TO new_column_name;
```



- 重命名物化视图。

```
ALTER MATERIALIZED VIEW [ IF EXISTS ] mv_name
    RENAME TO new_name;
```

版权所有 © 华为云计算技术有限公司

参数说明

- **mv_name**
  一个现有物化视图的名称，可以用模式修饰。
  取值范围：字符串，要符合**标识符命名规范**。
- **column_name**
  一个新的或者现有的列的名称。
  取值范围：字符串，要符合**标识符命名规范**。
- **new_column_name**
  一个现有列的新名称。
- **new_owner**
  该物化视图的新拥有者的用户名。
- **new_name**
  该物化视图的新名称。

示例

- 修改物化视图所有者。

```
--创建表。
gaussdb=# CREATE TABLE my_table (c1 int, c2 int) WITH(STORAGE_TYPE=ASTORE);

--创建全量物化视图。
gaussdb=# CREATE MATERIALIZED VIEW foo AS SELECT * FROM my_table;

--创建用户。
gaussdb=# CREATE USER test PASSWORD '********';

--修改全量物化视图的所有者。
gaussdb=# ALTER MATERIALIZED VIEW foo OWNER TO test;

--查看物化视图信息。
gaussdb=# \dm foo
                        List of relations
 Schema | Name |       Type        | Owner |             Storage
--------+------+-------------------+-------+---------------------------------
 public | foo  | materialized view | test  | {orientation=row,compression=no}
(1 row)
```

- 重命名物化视图的列。

```
--查询物化视图的列。
gaussdb=# \d foo;
Materialized view "public.foo"
 Column |  Type   | Modifiers
--------+---------+-----------
 c1     | integer |
 c2     | integer |
Rules:
    "_RETURN" AS
    ON SELECT TO foo DO INSTEAD  SELECT my_table.c1, my_table.c2
   FROM my_table
Replica Identity: NOTHING

--将物化视图foo的列c1修改为col1，c2修改为col2。
gaussdb=# ALTER MATERIALIZED VIEW foo RENAME c1 to col1;
gaussdb=# ALTER MATERIALIZED VIEW foo RENAME c2 to col2;

--通过SELECT查看该物化视图的列。
gaussdb=# SELECT * FROM foo WHERE 1=2;
 col1 | col2
------+------
(0 rows)
```


版权所有 © 华为云计算技术有限公司

- 重命名物化视图。

```
--将物化视图foo重命名为my_mview。
gaussdb=# ALTER MATERIALIZED VIEW foo RENAME TO my_mview;

--查询信息。
gaussdb=# \dm my_mview
                        List of relations
 Schema |   Name   |       Type        | Owner |            Storage
--------+----------+-------------------+-------+--------------------------------
 public | my_mview | materialized view | test  | {orientation=row,compression=no}
(1 row)

--删除。
gaussdb=# DROP MATERIALIZED VIEW my_mview;
gaussdb=# DROP TABLE my_table ;
gaussdb=# DROP USER test;
```

相关链接

CREATE INCREMENTAL MATERIALIZED VIEW，**CREATE MATERIALIZED VIEW**，**DROP MATERIALIZED VIEW**，**REFRESH INCREMENTAL MATERIALIZED VIEW**，**REFRESH MATERIALIZED VIEW**

1.13.7.20 ALTER NODE

功能描述

修改一个现有集群节点的定义。

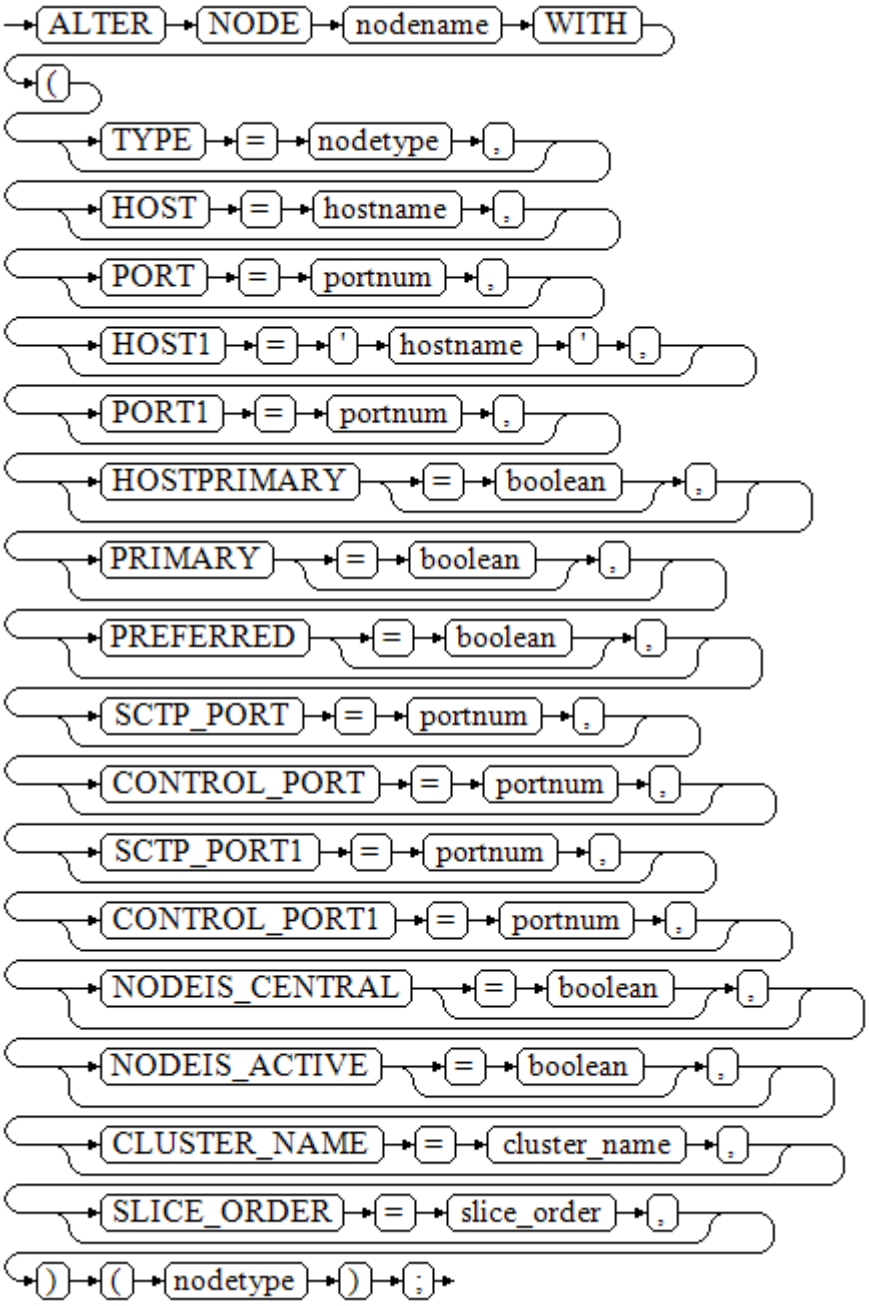
注意事项

ALTER NODE是集群管理工具封装的接口，用来实现集群管理，管理员用户才有权限使用该接口。该接口禁止用户直接使用，以免对集群状态造成影响，导致数据库不可用。

语法格式

```
ALTER NODE nodename WITH
  (
    [ TYPE = nodetype,]
    [ HOST = hostname,]
    [ PORT = portnum,]
    [ HOST1 = 'hostname',]
    [ PORT1 = portnum,]
    [ HOSTPRIMARY [ = boolean ],]
    [ PRIMARY [ = boolean ],]
    [ PREFERRED [ = boolean ],]
    [ SCTP_PORT = portnum,]
    [ CONTROL_PORT = portnum,]
    [ SCTP_PORT1 = portnum,]
    [ CONTROL_PORT1 = portnum, ]
    [ NODEIS_CENTRAL [ = boolean ],]
    [ NODEIS_ACTIVE [= boolean],]
    [ CLUSTER_NAME = cluster_name, ]
    [ SLICE_ORDER = slice_order, ]
  ) (nodetype);
```


版权所有 © 华为云计算技术有限公司



说明

PORT选项指定的端口号为节点间内部通信绑定的端口号，不同于外部客户端连接节点的端口号，可通过pgxc_node表查询。

参数说明

- **nodename**

  节点名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **TYPE = nodetype**

  指定节点的类型。

版权所有 © 华为云计算技术有限公司

取值范围：
  - 'coordinator'
  - 'datanode'
  - 'coordinator_standbycluster'
  - 'datanode_standbycluster_main'
  - 'datanode_standbycluster_other'
- **HOST = hostname**
  指定节点对应的主机名称或者IP地址。
- **PORT = portnum**
  指定节点绑定的主机端口号。
- **HOST1 = hostname**
  指定节点对应的备机名称或者IP地址。
- **PORT1 = portnum**
  指定节点绑定的备机端口号。
- **HOSTPRIMARY**
  声明主机名称或者IP地址是否为主。
  取值范围：
  - true
  - false（默认值）
- **PRIMARY = boolean**
  声明该节点是否为主节点。主节点允许做读写操作，否则只允许读操作。
  取值范围：
  - true
  - false（默认值）
- **PREFERRED = boolean**
  声明该节点是否为读操作的首选节点。
  取值范围：
  - true
  - false（默认值）
- **SCTP_PORT = portnum**
  主机TCP代理通信库使用的数据传输通道侦听端口，使用TCP协议侦听连接。
- **CONTROL_PORT = portnum**
  主机TCP代理通信库使用的控制传输通道侦听端口，使用TCP协议侦听连接。
- **SCTP_PORT1 = portnum**
  备机TCP代理通信库使用的数据传输通道侦听端口，使用TCP协议侦听连接。
- **CONTROL_PORT 1= portnum**
  备机TCP代理通信库使用的控制传输通道侦听端口，使用TCP协议侦听连接。
- **NODEIS_CENTRAL**
  表明当前节点是否为中心控制节点，只用于CN，对DN无效。
  - true

版权所有 © 华为云计算技术有限公司

  - false（默认值）
- **NODEIS_ACTIVE**

  表明当前节点是否是正常状态，用于标记CN是否被剔除，对DN无效。
  - true
  - false（默认值）
- **CLUSTER_NAME**

  集群名称。
- **SLICE_ORDER**

  节点归属分片的顺序序号。
- **nodetype**

  待修改节点的类型，pgxc_node系统表中node_type的值。

  取值范围："C"（协调节点）。

  说明

  多CN环境下，创建集群节点时只会在当前连接的CN下创建，不会同步至其他CN，需要在所有CN上执行，如果删除集群节点只需要链接一个CN即可。如果只在其中一个CN执行创建了集群节点，要删除需要定义当前cnnodename，并在当前CN下执行删除操作。

示例

```
--创建集群节点，在所有CN上执行。
gaussdb=# CREATE NODE datanode1 WITH(
   TYPE = datanode,
   PREFERRED = false
);

--查询集群DN初始状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 datanode1 | f
 datanode2 | f
(2 rows)

--将datanode1设为preferred DN。
gaussdb=# ALTER NODE datanode1 WITH(preferred = true);

--查询集群DN变更后状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 datanode1 | t
 datanode2 | f
(2 rows)

--删除集群节点，连接一个CN即可。
gaussdb=# DROP NODE datanode1;
```

相关链接

CREATE NODE，**DROP NODE**

版权所有 © 华为云计算技术有限公司

1.13.7.21 ALTER NODE GROUP

功能描述

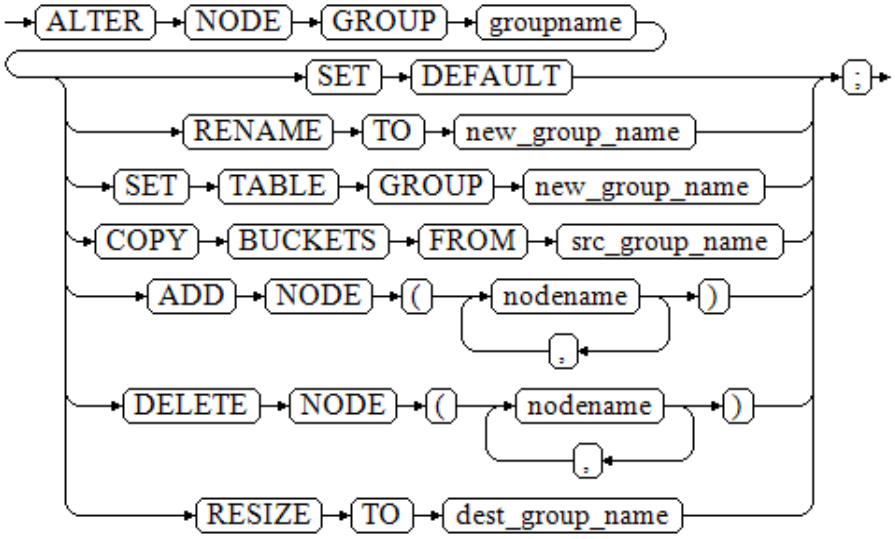
修改一个集群节点组的信息。

注意事项

- 只有系统管理员或者被授予了NODE GROUP的ALTER权限的用户可以修改NODE GROUP信息。
- 修改NODE GROUP操作都是系统内部操作，除了SET DEFAULT语法之外，其他操作都需要在维护模式下（调用SET xc_maintenance_mode=on;）。
- ALTER NODE GROUP语法仅仅应该在数据库内部使用，使用者不应该手动调用这些SQL语句，否则会导致数据库系统数据不一致。

语法格式

```
ALTER NODE GROUP groupname
{ SET DEFAULT
    | RENAME TO new_group_name
    | SET TABLE GROUP new_group_name
    | COPY BUCKETS FROM src_group_name
    | ADD NODE ( nodename [, ... ] )
    | DELETE NODE ( nodename [, ... ] )
    | RESIZE TO dest_group_name
};
```



参数说明

- **groupname**

  需要修改的集群节点组名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **SET DEFAULT**

  将系统中除groupname指定的集群节点组之外的其他集群节点组对象的in_redistribution字段设置为'y'。考虑到兼容以前版本，该语法仍然保留，且不需要设置维护模式。

版权所有 © 华为云计算技术有限公司

- **RENAME TO new_group_name**

  将指定的集群节点组的名称进行重命名（new_group_name）。

- **SET TABLE GROUP new_group_name**

  将所有CN节点的pgxc_class表中pgroup字段是group_name的记录修改为new_group_name。

- **COPY BUCKETS FROM src_group_name**

  在src_group_name表示的集群节点组中，将group_members字段和group_buckets字段的内容复制到groupname所表示的集群节点组中。

- **ADD NODE (nodename [, ...])**

  在groupname指定的集群节点组中增加指定的节点，这些新增节点在PGXC_NODE系统表中存在。该语句仅仅修改系统表，不会进行实际的节点添加和数据重分布，用户不应该直接调用该SQL语句。可以在PGXC_GROUP系统表中观察该语句的影响。

- **DELETE NODE (nodename [, ...])**

  在groupname指定的集群节点组中，将指定的节点移除，这些被移除的节点仍然存在于PGXC_NODE系统表中。该语句仅仅修改系统表，不会进行实际的节点移除和数据重分布，用户不应该直接调用该SQL语句。可以在PGXC_GROUP系统表中观察该语句的影响。

- **RESIZE TO dest_group_name**

  设置集群resize操作标志，将groupname指定的集群节点组设置为重分布的源集群节点组，并取消is_installation标志；同时将dest_group_name设置为重分布的目标集群节点组，并设置is_installation标志。

示例

```
--查询集群DN初始状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 dn_6001_6002_6003 | f
 dn_6004_6005_6006 | f
 dn_6007_6008_6009 | f
(3 rows)

--创建NODE GROUP，用上一步中查询到的真实节点名称替换dn_6001_6002_6003。
gaussdb=# CREATE NODE GROUP test_group WITH ( dn_6001_6002_6003 );

--修改创建出的NODE GROUP名称为test_group_new。需要在维护模式下（调用SET
xc_maintenance_mode=on;）。
gaussdb=# SET xc_maintenance_mode=on;
gaussdb=# ALTER NODE GROUP test_group RENAME TO test_group_new;

--查询NODE GROUP。
gaussdb=# SELECT group_name, group_members FROM pgxc_group;
   group_name   |   group_members
----------------+-------------------
 group_version1 | 16384 16388 16394
 test_group_new | 16384
(2 rows)

--删除NODE GROUP，且关闭维护模式。
gaussdb=# DROP NODE GROUP test_group_new;
gaussdb=# SET xc_maintenance_mode=off;
```

相关链接

CREATE NODE GROUP，DROP NODE GROUP

版权所有 © 华为云计算技术有限公司

1.13.7.22 ALTER PACKAGE

功能描述

修改包（PACKAGE）的属性或重编译包（PACKAGE）。

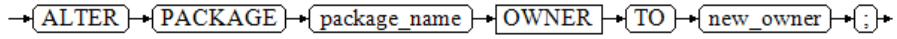
注意事项

- 目前仅支持ALTER PACKAGE OWNER功能，系统管理员默认拥有该权限，存在以下权限约束：当前用户必须是该PACKAGE的所有者或者系统管理员，且该用户是新所有者角色的成员。
- 当打开三权分立时，即使是系统管理员，也必须拥有用户组权限才能修改PACKAGE的所有者，DEFINER类型PACKAGE不允许修改所有者。
- 只有初始化用户才能修改PACKAGE的所有者为初始化用户。
- 当关闭三权分立时，仅系统管理员以上权限可以修改PACKAGE所有者。
- 不允许系统管理员将DEFINER类型的PACKAGE的所有者改为初始用户或运维管理员。

语法格式

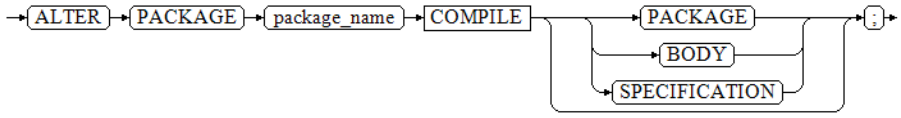
- 修改PACKAGE的所有者。

```
ALTER PACKAGE package_name OWNER TO new_owner;
```



- 重编译PACKAGE。

```
ALTER PACKAGE package_name COMPILE [PACKAGE | BODY | SPECIFICATION];
```



参数说明

- **package_name**

  要修改的PACKAGE名称。

  取值范围：已存在的PACKAGE名，仅支持修改单个PACKAGE名称。

- **new_owner**

  PACKAGE的新所有者。要修改PACKAGE的所有者，新所有者必须拥有该PACKAGE所属模式的CREATE权限。

  取值范围：已存在的用户角色。

示例

- 修改PACKAGE的所有者。

```
--创建数据库。
gaussdb=# CREATE DATABASE test DBCOMPATIBILITY 'ORA';
gaussdb=# \c test

--创建PACKAGE。
test=# CREATE OR REPLACE PACKAGE test_pkg AS
    pkg_var int := 1;
    PROCEDURE test_pkg_proc(var int);
```


版权所有 © 华为云计算技术有限公司

```
END test_pkg;
/

test=# CREATE OR REPLACE PACKAGE BODY test_pkg AS
    PROCEDURE test_pkg_proc(var int) AS
    BEGIN
        pkg_var := 1;
    END;
END test_pkg;
/

--创建用户。
test=# CREATE ROLE test PASSWORD '********';

--修改包的所有者。
test=# ALTER PACKAGE test_pkg OWNER TO test;

--查询test_pkg的所有者。
test=# SELECT t1.pkgname,t2.rolname FROM gs_package t1, gs_roles t2 WHERE t1.pkgname =
'test_pkg' AND t1.pkgowner = t2.oid;
 pkgname  | rolname
----------+---------
 test_pkg | test
(1 row)

--删除。
test=# DROP PACKAGE test_pkg;
test=# DROP ROLE test;
--切换至默认数据库（根据实际情况修改数据库名）。
test=# \c postgres
gaussdb=# DROP DATABASE test;
```

相关链接

CREATE PACKAGE，**DROP PACKAGE**

1.13.7.23 ALTER PROCEDURE

功能描述

修改自定义存储过程的属性或重编译存储过程。

注意事项

- 只有存储过程的所有者或者被授予了存储过程ALTER权限的用户才能执行ALTER PROCEDURE命令，系统管理员默认拥有该权限。针对所要修改属性的不同，还有以下权限约束：
  - 如果存储过程中涉及对临时表相关的操作，则无法使用ALTER PROCEDURE。
  - 修改存储过程的所有者或修改存储过程的模式，当前用户必须是该存储过程的所有者或者系统管理员，且该用户是新所有者角色的成员。
  - 只有系统管理员和初始化用户可以将存储过程的模式修改成public。
- 仅有初始用户或者创建该存储过程的用户可以修改存储过程为定义者权限的存储过程。
- 当打开三权分立时，对于定义者权限的存储过程，不允许任何角色修改存储过程的所有者。
- 当关闭三权分立时，对于定义者权限的存储过程，仅初始用户和系统管理员可以修改存储过程的所有者。

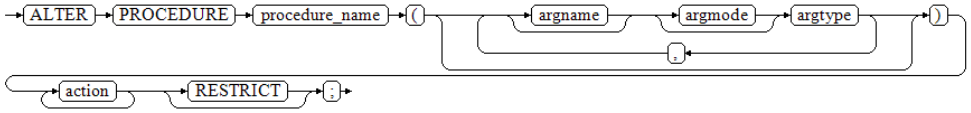
版权所有 © 华为云计算技术有限公司

- 只有初始用户才能修改存储过程的所有者为初始用户。

语法格式

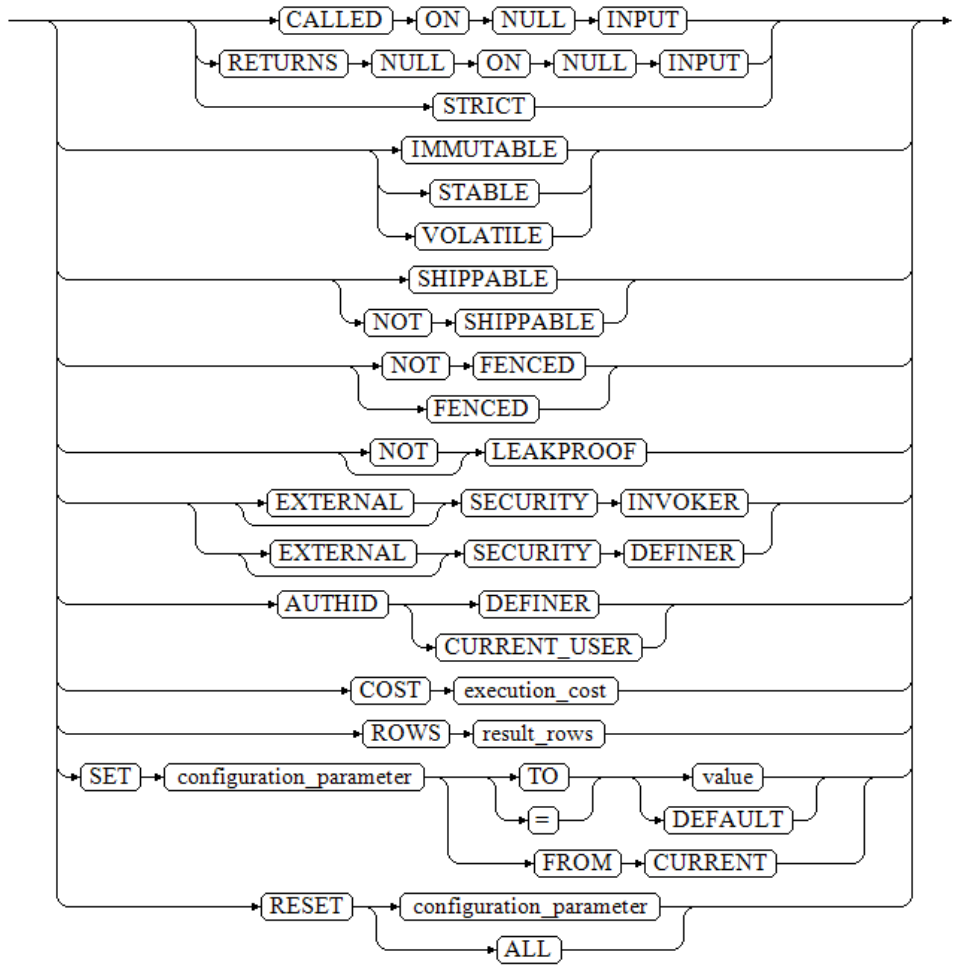
- 修改自定义存储过程的附加参数

```
ALTER PROCEDURE procedure_name ( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    action [ ... ] [ RESTRICT ];
```



其中，附加参数action子句语法为：

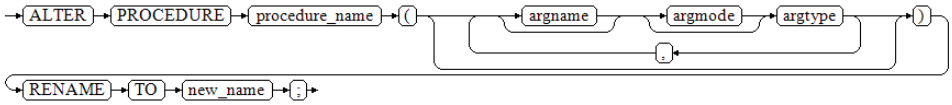
```
{{CALLED ON NULL INPUT  | RETURNS NULL ON NULL INPUT | STRICT}
 | {IMMUTABLE | STABLE | VOLATILE}
 | {SHIPPABLE | NOT SHIPPABLE}
 | {NOT FENCED | FENCED}
 | [ NOT ] LEAKPROOF
 | { [ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER }
 | AUTHID { DEFINER | CURRENT_USER }
 | COST execution_cost
 | ROWS result_rows
 | SET configuration_parameter { { TO | = } { value | DEFAULT }| FROM CURRENT}
 | RESET {configuration_parameter | ALL}}
```



- 修改自定义存储过程的名称

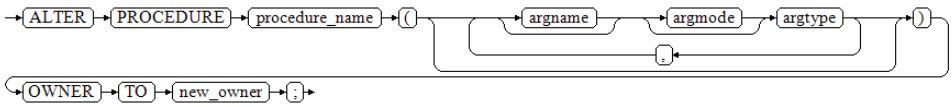
```
ALTER PROCEDURE procedure_name( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    RENAME TO new_name;
```


版权所有 © 华为云计算技术有限公司



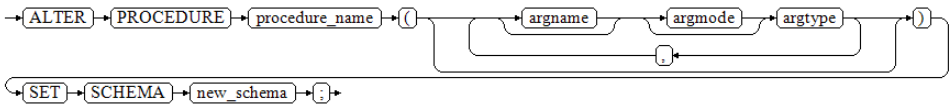
- 修改自定义存储过程的所有者

```
ALTER PROCEDURE procedure_name( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    OWNER TO new_owner;
```



- 修改自定义存储过程的模式

```
ALTER PROCEDURE procedure_name( [ { [ argname ] [ argmode ] argtype} [, ...] ] )
    SET SCHEMA new_schema;
```



参数说明

- **procedure_name**

  要修改的存储过程名称。

  取值范围：已存在的存储过程名。

- **argmode**

  用于标识该参数是输入参数或输出参数。

  取值范围：IN、OUT、INOUT或VARIADIC。

- **argname**

  参数名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **argtype**

  存储过程参数的类型。

- **CALLED ON NULL INPUT**

  表明该存储过程的某些参数为NULL时，可以按照正常的方式调用。缺省时与指定此参数的作用相同。

- **RETURNS NULL ON NULL INPUT | STRICT**

  STRICT：用于指定当一个存储过程的某个参数为NULL时，则该存储过程返回NULL。如果声明了该参数，则如果存在NULL参数时不会执行该存储过程，而直接自动返回一个NULL结果。

  RETURNS NULL ON NULL INPUT和STRICT的功能相同。

- **IMMUTABLE**

  表示该存储过程在给出同样的参数值时总是返回同样的结果。

- **STABLE**

  表示该存储过程不能修改数据库，对相同参数值，在同一次表扫描里，该存储过程的返回值不变，但是返回值可能在不同SQL语句之间变化。

- **VOLATILE**

  表示该存储过程值可以在一次表扫描内改变，不会做任何优化。

版权所有 © 华为云计算技术有限公司

- **SHIPPABLE | NOT SHIPPABLE、FENCED | NOT FENCED等**
  行为约束可选项。各参数的功能与**CREATE FUNCTION**类似，详细说明请参见**CREATE FUNCTION**。
- **LEAKPROOF**
  表示该存储过程没有副作用，指出参数只包括返回值。LEAKPROOF只能由系统管理员设置。
- **EXTERNAL**
  （可选）目的是和SQL兼容，这个特性适合于所有函数，而不仅是外部函数。
- **SECURITY INVOKER | AUTHID CURRENT_USER**
  表明该存储过程将以调用者的用户的权限执行。缺省时与指定此参数的作用相同。
  SECURITY INVOKER和AUTHID CURRENT_USER的功能相同。
- **SECURITY DEFINER | AUTHID DEFINER**
  声明该存储过程将以创建者的用户的权限执行。
  AUTHID DEFINER和SECURITY DEFINER的功能相同。
- **COST execution_cost**
  用于估计存储过程的执行成本。
  execution_cost以cpu_operator_cost为单位。
  取值范围：正数。
- **ROWS result_rows**
  估计存储过程返回的行数。用于存储过程返回的是一个集合。
  取值范围：正数，默认值是1000。
- **configuration_parameter**
  - **value**
    将指定的数据库会话参数值设置为具体值。
    取值范围：字符串。
    - DEFAULT：在新的会话中使用系统的缺省设置。
    - OFF：关闭设置。
    - RESET：在新的会话中使用系统的缺省设置。
    - 用户指定的值：需要满足修改参数的取值限制。
  - **DEFAULT**
    默认值。
  - **FROM CURRENT**
    取当前会话中的值设置为configuration_parameter的值。
- **ALL**
  所有运行时参数。
- **new_name**
  存储过程的新名称。要修改存储过程的所属模式，必须拥有新模式的CREATE权限。
  取值范围：字符串，要符合**标识符命名规范**。

版权所有 © 华为云计算技术有限公司

- **new_owner**

  存储过程的新所有者。要修改存储过程的所有者，新所有者必须拥有该存储过程所属模式的CREATE权限。

  取值范围：已存在的用户角色。

- **new_schema**

  存储过程的新模式。

  取值范围：已存在的模式。

- **RESTRICT**

  仅为了符合SQL标准，可被忽略。

示例

```
--创建存储过程。
gaussdb=# CREATE OR REPLACE PROCEDURE test_proc(a int)
IS
    proc_var int;
BEGIN
    proc_var := a;
END;
/

--修改存储过程的执行规则为IMMUTABLE,即参数不变时返回结果相同。
gaussdb=# ALTER PROCEDURE test_proc(a int) IMMUTABLE;

--新建用户。
gaussdb=# CREATE USER joe PASSWORD '********';

--修改存储过程的所有者。
gaussdb=# ALTER PROCEDURE test_proc(a int)  OWNER TO joe;

--新建模式。
gaussdb=# CREATE SCHEMA test;

--修改存储过程的所属模式。
gaussdb=# ALTER PROCEDURE  test_proc(a int)  SET SCHEMA test;

--删除存储过程,函数名需要指定所属模式。
gaussdb=# DROP PROCEDURE test.test_proc;

--删除新建的用户和模式。
gaussdb=# DROP USER joe;
gaussdb=# DROP SCHEMA test;
```

相关链接

CREATE PROCEDURE，**DROP PROCEDURE**

1.13.7.24 ALTER RESOURCE LABEL

功能描述

修改资源标签属性。

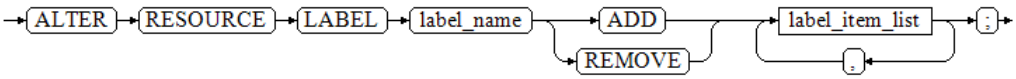
注意事项

只有poladmin、sysadmin或初始用户才能执行ALTER RESOURCE LABEL命令。

版权所有 © 华为云计算技术有限公司

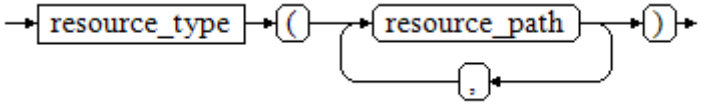
语法格式

```
ALTER RESOURCE LABEL label_name {ADD | REMOVE}
  label_item_list [, ...];
```



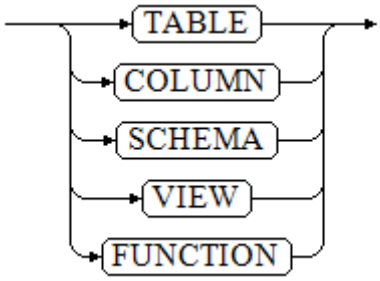
- label_item_list：

  ```
  resource_type (resource_path [, ...])
  ```

- resource_type：

  ```
  {TABLE | COLUMN | SCHEMA | VIEW | FUNCTION}
  ```

参数说明

- **label_name**

  资源标签名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **resource_type**

  指的是要标记的数据库资源类型。

- **resource_path**

  描述具体的数据库资源的路径。

示例

```
--创建基本表table_for_label。
gaussdb=# CREATE TABLE table_for_label(col1 int, col2 text);

--创建资源标签table_label。
gaussdb=# CREATE RESOURCE LABEL table_label ADD COLUMN(table_for_label.col1);

--将col2添加至资源标签table_label中。
gaussdb=# ALTER RESOURCE LABEL table_label ADD COLUMN(table_for_label.col2);

--将资源标签table_label中的一项移除。
gaussdb=# ALTER RESOURCE LABEL table_label REMOVE COLUMN(table_for_label.col1);

--删除资源标签table_label。
gaussdb=# DROP RESOURCE LABEL table_label;

--删除基本表table_for_label。
gaussdb=# DROP TABLE table_for_label;
```


版权所有 © 华为云计算技术有限公司

相关链接

CREATE RESOURCE LABEL，DROP RESOURCE LABEL。

1.13.7.25 ALTER RESOURCE POOL

功能描述

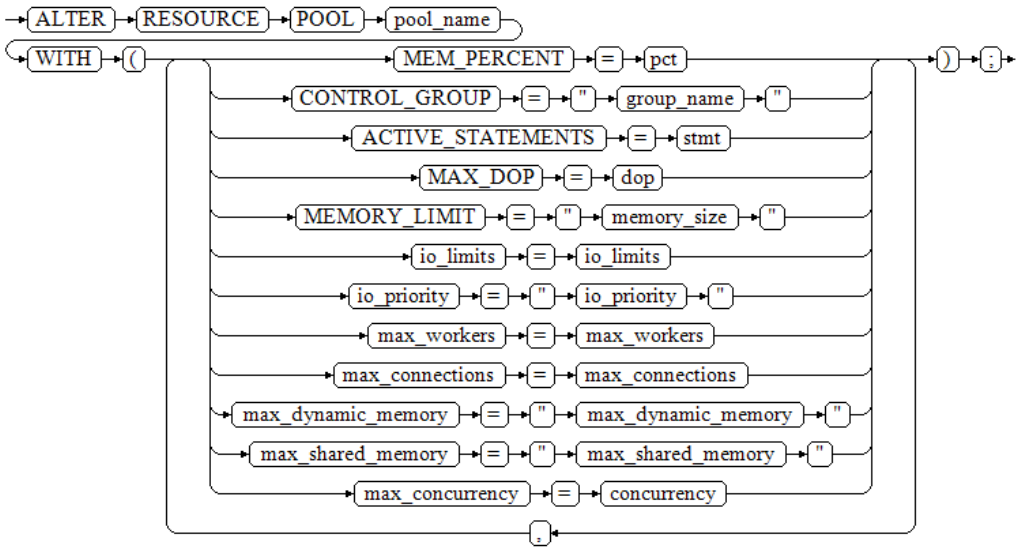
修改一个资源池，指定其他控制组。

注意事项

- 只有sysadmin或初始用户才能执行ALTER RESOURCE POOL命令。
- 升级期间不支持本语法。

语法格式

```
ALTER RESOURCE POOL pool_name
    WITH ({MEM_PERCENT = pct | CONTROL_GROUP = "group_name" | ACTIVE_STATEMENTS = stmt |
MAX_DOP = dop | MEMORY_LIMIT = "memory_size" | io_limits = io_limits | io_priority = "io_priority" |
max_worker = max_worker | max_connections = max_connections | max_dynamic_memory =
"max_dynamic_memory" | max_shared_memory = "max_shared_memory" | max_concurrency =
concurrency} [, ... ]);
```



参数说明

- **pool_name**

  资源池名称。资源池名称为已创建的资源池。

  取值范围：字符串，要符合标识符命名规范。

- **group_name**

  控制组名称。

  取值范围：已创建的控制组。

版权所有 © 华为云计算技术有限公司

说明

- 设置控制组名称时，语法可以使用双引号，也可以使用单引号。
- group_name对大小写敏感。
- 不指定group_name时，默认指定的字符串为"Medium"，代表指定DefaultClass控制组的"Medium" Timeshare控制组。
- 若数据库管理员指定自定义Class组下的Workload控制组，如control_group的字符串为："class1:workload1"；代表此资源池指定到class1控制组下的workload1控制组。也可同时指定Workload控制组的层次，如control_group的字符串为："class1:workload1:1"。
- 若数据库用户指定Timeshare控制组代表的字符串，即"Rush"、"High"、"Medium"或"Low"其中一种，如control_group的字符串为"High"；代表资源池指定到DefaultClass控制组下的"High" Timeshare控制组。
- 多租户场景下，组资源池关联的控制组为class级别，业务资源池关联Workload控制组。且不允许在各种资源池间相互切换。

- **stmt**

  资源池语句执行的最大并发数量。

  取值范围：数值型，-1~2147483647，其中，-1表示不限制并发数。

- **dop**

  资源池最大并发度，语句执行时能够创建的最多线程数量。

  取值范围：数值型，1~2147483647。

- **memory_size**

  资源池最大使用内存。

  取值范围：字符串，内容范围1KB~2047GB，单位大小写敏感。

- **mem_percent**

  资源池可用内存占全部内存或者组用户内存使用的比例。

  - 在多租户场景下，组用户和业务用户的mem_percent范围为1-100的整数，默认为20。
  - 在普通场景下，普通用户的mem_percent范围为0-100的整数，默认值为0。

  说明

  mem_percent和memory_limit同时指定时，只有mem_percent起作用。

- **io_limits**

  资源池每秒可触发I/O次数上限，以万次为单位计数。

  取值范围：数值型，0~2147483647，0表示不限制I/O次数。

- **io_priority**

  I/O利用率高达90%时，重消耗I/O作业进行I/O资源管控时关联的优先级等级。

  取值范围：枚举型，可选项为None、Low、Medium和High。不控制时，可设置为None，默认为None。

说明

io_limits和io_priority的设置都仅对复杂作业有效。包括批量导入（INSERT INTO SELECT, COPY FROM, CREATE TABLE AS等），单DN数据量大约超过500MB的复杂查询和VACUUM FULL等操作。

- **max_worker**

  只用于扩容的接口，表示扩容数据重分布时，表内插入并发度。

版权所有 © 华为云计算技术有限公司

- **max_connections**
  最大连接数，用于限制资源池可使用的最大连接数。

  说明

  所有资源池的最大连接数总和不能超过整个gaussdb进程设置的GUC参数max_connections指定的最大连接数。
- **max_dynamic_memory**
  最大动态内存值，用于限制资源池允许使用的最大动态内存值。
- **max_shared_memory**
  最大共享内存值，用于限制资源池允许使用的最大共享内存值。
- **max_concurrency**
  最大并发数，用于限制资源池允许使用的最大并发数。

示例

本示例假定用户已成功创建自定义的class1控制组及其下属的Low、wg1、wg2三个Workload控制组。若需创建控制组请联系管理员处理。

```
--创建一个资源池。
gaussdb=# CREATE RESOURCE POOL pool1;

--更新一个资源池，其控制组指定为"DefaultClass"组下属的"High" Timeshare Workload控制组。
gaussdb=# ALTER RESOURCE POOL pool1 WITH (CONTROL_GROUP="High");

--更新一个资源池，其控制组指定为"class1"组下属的"Low" Timeshare Workload控制组。
gaussdb=# ALTER RESOURCE POOL pool1 WITH (CONTROL_GROUP="class1:Low");

--更新一个资源池，其控制组指定为"class1"组下属的"wg1" Workload控制组。
gaussdb=# ALTER RESOURCE POOL pool1 WITH (CONTROL_GROUP="class1:wg1");

--更新一个资源池，其控制组指定为"class1"组下属的"wg2" Workload控制组。
gaussdb=# ALTER RESOURCE POOL pool1 WITH (CONTROL_GROUP="class1:wg2:3");

--删除资源池pool1。
gaussdb=# DROP RESOURCE POOL pool1;
```

相关链接

CREATE RESOURCE POOL，**DROP RESOURCE POOL**

1.13.7.26 ALTER ROLE

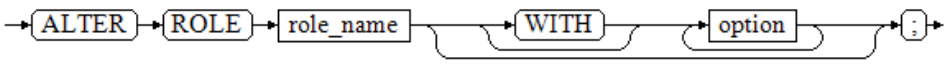
功能描述

修改角色属性。包括：修改角色的名称、锁定/解锁角色以及权限等。

语法格式

- 修改角色的权限

```
ALTER ROLE role_name [ [ WITH ] option [ ... ] ];
```

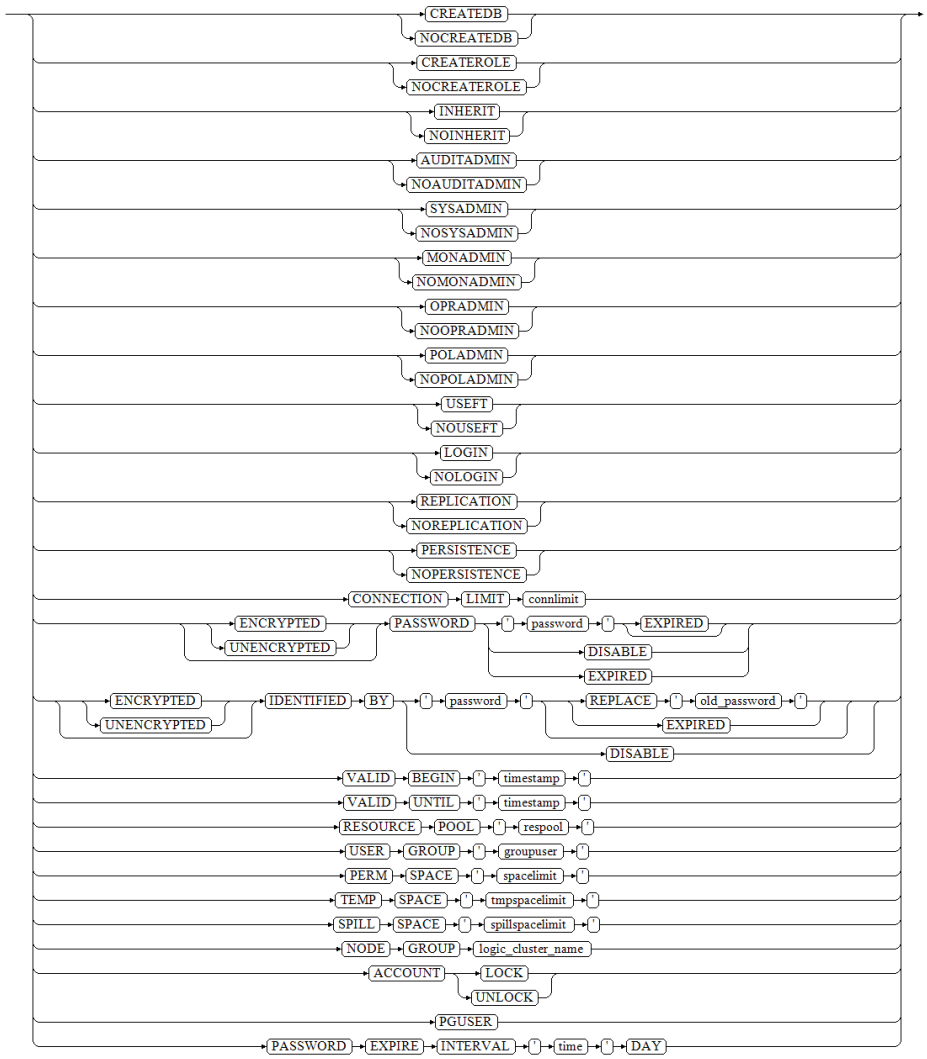


其中option子句为：

```
{
  {CREATEDB | NOCREATEDB}
```


版权所有 © 华为云计算技术有限公司

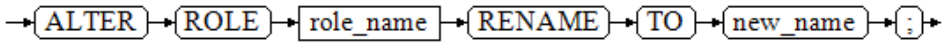
```
    | {CREATEROLE | NOCREATEROLE}
    | {INHERIT | NOINHERIT}
    | {AUDITADMIN | NOAUDITADMIN}
    | {SYSADMIN | NOSYSADMIN}
    | {MONADMIN | NOMONADMIN}
    | {OPRADMIN | NOOPRADMIN}
    | {POLADMIN | NOPOLADMIN}
    | {USEFT | NOUSEFT}
    | {LOGIN | NOLOGIN}
    | {REPLICATION | NOREPLICATION}
    | {PERSISTENCE | NOPERSISTENCE}
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD { 'password' [EXPIRED] | DISABLE | EXPIRED }
    | [ ENCRYPTED | UNENCRYPTED ] IDENTIFIED BY { 'password' [ REPLACE 'old_password' |
EXPIRED ] | DISABLE }
    | VALID BEGIN 'timestamp'
    | VALID UNTIL 'timestamp'
    | RESOURCE POOL 'respool'
    | USER GROUP 'groupuser'
    | PERM SPACE 'spacelimit'
    | TEMP SPACE 'tmpspacelimit'
    | SPILL SPACE 'spillspacelimit'
    | NODE GROUP logic_cluster_name
    | ACCOUNT { LOCK | UNLOCK }
    | PGUSER
    | PASSWORD EXPIRE INTERVAL 'time' DAY
}
```



版权所有 © 华为云计算技术有限公司

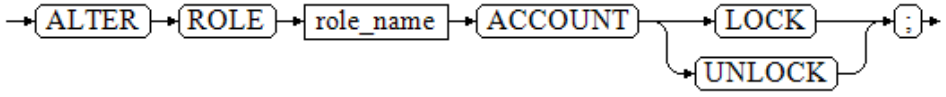
- 修改角色的名称

```
ALTER ROLE role_name
    RENAME TO new_name;
```



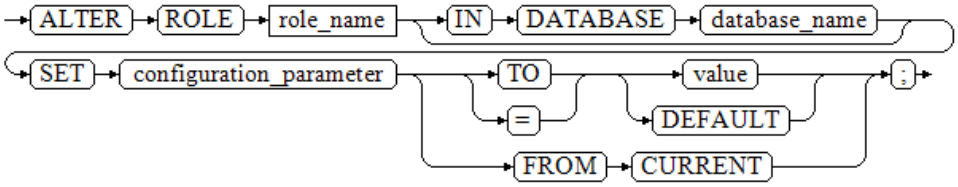
- 锁定或解锁角色

```
ALTER ROLE role_name
    ACCOUNT { LOCK | UNLOCK };
```



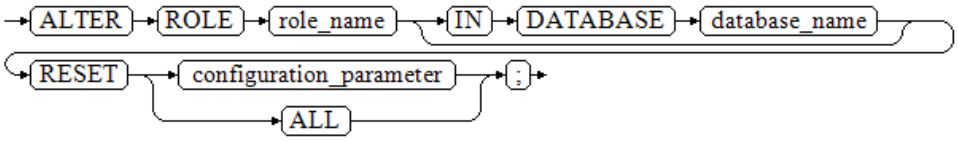
- 设置角色的配置参数

```
ALTER ROLE role_name [ IN DATABASE database_name ]
    SET configuration_parameter {{ TO | = } { value | DEFAULT } | FROM CURRENT};
```



- 重置角色的配置参数

```
ALTER ROLE role_name
    [ IN DATABASE database_name ] RESET {configuration_parameter|ALL};
```



参数说明

- **role_name**

  现有角色名。

  取值范围：已存在的角色名，如果角色名中包含大写字母则需要使用双引号进行包裹。

  须知

  当前不支持对重分布资源管控对象redisuser进行重命名（RENAME TO）操作，也不支持将其他角色的名称重命名（RENAME TO）为redisuser。

- **IN DATABASE database_name**

  修改角色在指定数据库上的参数。

- **SET configuration_parameter {{ TO | = } { value | DEFAULT } | FROM CURRENT}**

  设置角色的参数。

版权所有 © 华为云计算技术有限公司

须知

ALTER ROLE中修改的会话参数只针对指定的角色，当前版本不支持设置用户级别参数。

取值范围：

- configuration_parameter：取值请参见**SET**。
- value：取值请参见**SET**。
- DEFAULT：表示清除configuration_parameter参数的值，configuration_parameter参数的值将继承本角色新产生的SESSION的默认值。
- FROM CURRENT：取当前会话中的值设置为configuration_parameter参数的值。

- **RESET {configuration_parameter|ALL}**

  清除configuration_parameter参数的值。与SET configuration_parameter TO DEFAULT的效果相同。

须知

当前版本不支持重置用户级别参数。

取值范围：ALL表示清除所有参数的值。

- **ACCOUNT LOCK | ACCOUNT UNLOCK**
  - ACCOUNT LOCK：锁定角色，禁止角色通过密码认证登录数据库。
  - ACCOUNT UNLOCK：解锁角色，允许角色登录数据库。
- **PGUSER**

  当前版本不允许修改角色的PGUSER属性。
- **{PASSWORD|IDENTIFIED BY} 'password'**

  重置或修改用户密码。
  - 除了初始用户外，其他管理员或普通用户修改自己的密码需要输入正确的旧密码。
  - 只有初始用户、三权分立关闭时的系统管理员（sysadmin）或拥有创建用户（CREATEROLE）权限的用户才可以重置普通用户密码，无需输入旧密码。
  - 初始用户可以重置系统管理员的密码，系统管理员不允许重置其他系统管理员的密码。
  - 应当使用单引号或双引号将用户密码进行包裹。
- **EXPIRED**

  设置密码失效。
  - 只有初始用户、系统管理员（sysadmin）或拥有创建用户（CREATEROLE）权限的用户才可以设置用户密码失效，其中系统管理员只有在三权分立关闭时，才可以设置自己或其他系统管理员密码失效。
  - 不允许设置初始用户密码失效。
  - 密码失效的用户可以登录数据库但不能执行查询操作，只有修改密码或由管理员重置密码后才可以恢复正常查询操作。

版权所有 © 华为云计算技术有限公司

- 其他参数请参见**CREATE ROLE**章节的**参数说明**。

示例

```
--创建角色test_role。
gaussdb=# CREATE ROLE test_role PASSWORD '********';

--设置角色test_role可以登录数据库。
gaussdb=# ALTER ROLE test_role WITH LOGIN;

--创建数据库。
gaussdb=# CREATE DATABASE testdb1;

--锁定角色test_role。
gaussdb=# ALTER ROLE test_role ACCOUNT LOCK;

--给锁定锁定的角色解锁。
gaussdb=# ALTER ROLE test_role ACCOUNT UNLOCK;

--修改角色test_role密码。
gaussdb=# ALTER ROLE test_role PASSWORD '********';

--将角色test_role重命名为test_role2。
gaussdb=# ALTER ROLE test_role RENAME TO test_role2;

--修改角色test_role2为系统管理员。
gaussdb=# ALTER ROLE test_role2 SYSADMIN;

--删除。
gaussdb=# DROP ROLE test_role2;
gaussdb=# DROP DATABASE testdb1;
```

相关链接

CREATE ROLE，**DROP ROLE**，**SET ROLE**

1.13.7.27 ALTER ROW LEVEL SECURITY POLICY

功能描述

对已存在的行访问控制策略（包括：行访问控制策略的名称、行访问控制指定的用户、行访问控制的策略表达式）进行修改。

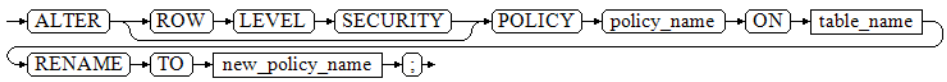
注意事项

表的所有者或管理员用户才能执行ALTER ROW LEVEL SECURITY POLICY命令。

语法格式

- 修改已存在行访问控制策略的名称

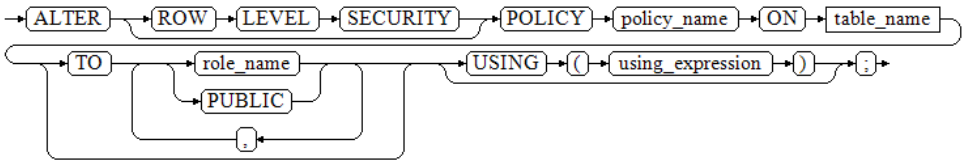
```
ALTER [ ROW LEVEL SECURITY ] POLICY policy_name ON table_name RENAME TO new_policy_name;
```



- 修改已存在行访问控制策略的指定用户、策略表达式

```
ALTER [ ROW LEVEL SECURITY ] POLICY policy_name ON table_name
    [ TO { role_name | PUBLIC } [, ...] ]
    [ USING ( using_expression ) ];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **policy_name**

  行访问控制策略名称。

- **table_name**

  行访问控制策略的表名。

- **new_policy_name**

  新的行访问控制策略名称。

- **role_name**

  行访问控制策略应用的数据库用户，可以指定多个用户，PUBLIC表示应用到所有用户。

- **using_expression**

  行访问控制策略，形式与WHERE子句中的布尔型表达式类似。

示例

```
--创建数据表all_data。
gaussdb=# CREATE TABLE all_data(id int, role varchar(100), data varchar(100));

--创建行访问控制策略，当前用户只能查看用户自身的数据。
gaussdb=# CREATE ROW LEVEL SECURITY POLICY all_data_rls ON all_data USING(role = CURRENT_USER);
gaussdb=# \d+ all_data
                        Table "public.all_data"
 Column |          Type          | Modifiers | Storage  | Stats target | Description
--------+------------------------+-----------+----------+--------------+-------------
 id     | integer                |           | plain    |              |
 role   | character varying(100) |           | extended |              |
 data   | character varying(100) |           | extended |              |
Row Level Security Policies:
    POLICY "all_data_rls" FOR ALL
      TO public
      USING (((role)::name = "current_user"()))
Has OIDs: no
Distribute By: HASH(id)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no, storage_type=USTORE, segment=off

--修改行访问控制all_data_rls的名称。
gaussdb=# ALTER ROW LEVEL SECURITY POLICY all_data_rls ON all_data RENAME TO all_data_new_rls;

--创建用户alice、bob。
gaussdb=# CREATE ROLE alice WITH PASSWORD '*********';
gaussdb=# CREATE ROLE bob WITH PASSWORD '*********';

--修改行访问控制策略影响的用户。
gaussdb=# ALTER ROW LEVEL SECURITY POLICY all_data_new_rls ON all_data TO alice, bob;
gaussdb=# \d+ all_data
                        Table "public.all_data"
 Column |          Type          | Modifiers | Storage  | Stats target | Description
--------+------------------------+-----------+----------+--------------+-------------
 id     | integer                |           | plain    |              |
 role   | character varying(100) |           | extended |              |
 data   | character varying(100) |           | extended |              |
```


版权所有 © 华为云计算技术有限公司

```
Row Level Security Policies:
    POLICY "all_data_new_rls" FOR ALL
      TO alice,bob
      USING (((role)::name = "current_user"()))
Has OIDs: no
Distribute By: HASH(id)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no, storage_type=USTORE, segment=off


--修改行访问控制策略表达式，只能查看列id大于100且用户自身的数据。
gaussdb=# ALTER ROW LEVEL SECURITY POLICY all_data_new_rls ON all_data USING (id > 100 AND role =
current_user);
gaussdb=# \d+ all_data
                            Table "public.all_data"
 Column |          Type          | Modifiers | Storage  | Stats target | Description
--------+------------------------+-----------+----------+--------------+-------------
 id     | integer                |           | plain    |              |
 role   | character varying(100) |           | extended |              |
 data   | character varying(100) |           | extended |              |
Row Level Security Policies:
    POLICY "all_data_new_rls" FOR ALL
      TO alice,bob
      USING (((id > 100) AND ((role)::name = "current_user"())))
Has OIDs: no
Distribute By: HASH(id)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no, storage_type=USTORE, segment=off

--删除访问控制策略。
gaussdb=# DROP ROW LEVEL SECURITY POLICY all_data_new_rls ON all_data;

--删除用户alice, bob。
gaussdb=# DROP ROLE alice, bob;

--删除数据表all_data。
gaussdb=# DROP TABLE all_data;
```

相关链接

CREATE ROW LEVEL SECURITY POLICY，**DROP ROW LEVEL SECURITY POLICY**

1.13.7.28 ALTER SCHEMA

功能描述

修改模式属性。包括：修改模式的名称、所有者以及模式的防篡改属性等。

注意事项

- 只有模式的所有者和被授予了模式ALTER权限的用户，在拥有当前数据库CREATE权限的情况下，才可以执行ALTER SCHEMA命令。三权分立开关关闭时，系统管理员默认拥有此权限。
- 修改模式的所有者，当前用户必须是该模式的所有者或者系统管理员，且该用户是新所有者角色的成员。
- 仅初始用户可以修改除PUBLIC以外的系统模式（如：pg_catalog、sys等）的所有者。修改系统自带模式的名称可能会导致部分功能不可用，甚至影响数据库正常运行，默认情况下不允许修改系统自带模式的名称。出于前向兼容性问题考虑，仅允许当系统在启动或升级过程中或参数allow_system_table_mods为on时可修改系统自带模式的名称。

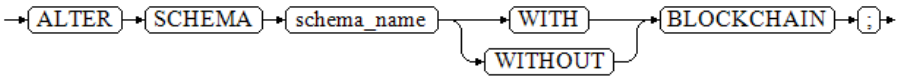
版权所有 © 华为云计算技术有限公司

- 除初始用户外，其他用户无法将模式的所有者修改为运维管理员。

语法格式

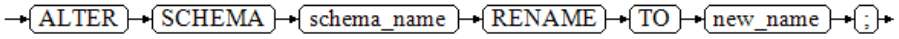
- 修改模式的防篡改属性

```
ALTER SCHEMA schema_name { WITH | WITHOUT } BLOCKCHAIN;
```



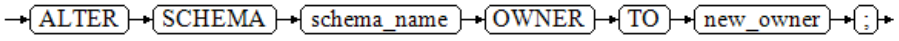
- 修改模式的名称

```
ALTER SCHEMA schema_name
  RENAME TO new_name;
```



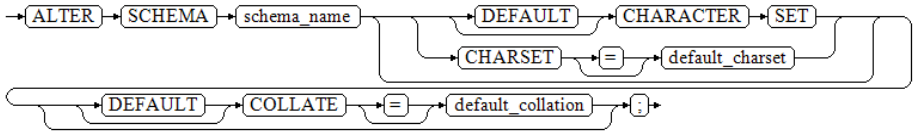
- 修改模式的所有者

```
ALTER SCHEMA schema_name
  OWNER TO new_owner;
```



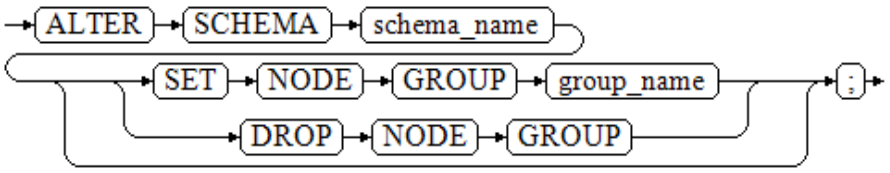
- 修改模式的默认字符集和字符序

```
ALTER SCHEMA schema_name
  [ [DEFAULT] CHARACTER SET | CHARSET [ = ] default_charset ] [ [DEFAULT] COLLATE [ = ]
default_collation ];
```



- 修改模式的单节点属性

```
ALTER SCHEMA schema_name
  [ SET NODE GROUP group_name | DROP NODE GROUP ];
```



参数说明

- **schema_name**

  现有模式的名称。

  取值范围：已存在的模式名。

- **RENAME TO new_name**

  修改模式的名称。new_name表示模式的新名称。

  取值范围：字符串，要符合**标识符命名规范**。

  模式新名称规则如下：

  - 新名称不能和当前数据库里其他的模式重名。
  - 新名称不能和当前数据库的初始用户重名。
  - 请勿使用“pg_”或“gs_role_”开头或“data_redis”命名新名称。

版权所有 © 华为云计算技术有限公司

- **OWNER TO new_owner**

  修改模式的所有者。new_owner表示模式的新所有者。

  取值范围：已存在的用户名或角色名。

  说明

  如果非系统管理员需要改变模式的所有者，则该用户还必须是新的所有角色的直接或间接成员，并且该成员必须在此数据库上有CREATE权限。

- **{ WITH | WITHOUT } BLOCKCHAIN**

  修改模式的防篡改属性。WITH BLOCKCHAIN表示防篡改模式。

  说明

  - 将普通模式修改为防篡改模式，需将GUC参数enable_ledger设置为on。该参数默认值为off，级别为SIGHUP。
  - 具有防篡改属性模式下的普通行存表均为防篡改历史表，不包括外表、临时表以及系统表。
  - 仅当模式下不包含任何表时，才可以对模式的防篡改属性进行修改或切换。
  - 不支持对临时表模式、toast表模式、dbe_perf模式或blockchain模式修改防篡改属性。

- **default_charset**

  修改模式的默认字符集。

  取值范围：支持设置的字符集请参见**表1-398**。

  须知

  - 仅在MYSQL模式数据库或M-Compatibility模式数据库下（即sql_compatibility = 'MYSQL'或sql_compatibility = 'M'）支持该语法。以上行为是在sql_compatibility = 'MYSQL'时的表现，sql_compatibility = 'M'的行为请参见《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > A > ALTER SCHEMA”的内容。
  - 未指定默认字符序时，模式的默认字符序为指定的字符集的默认字符序。

- **default_collate**

  修改模式的默认字符序。

  取值范围：支持设置的字符序请参见**表1-398**。

  须知

  - 仅在MYSQL模式数据库或M-Compatibility模式数据库下（即sql_compatibility = 'MYSQL'或sql_compatibility = 'M'）支持该语法。以上行为是在sql_compatibility = 'MYSQL'时的表现，sql_compatibility = 'M'的行为请参见《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > A > ALTER SCHEMA”的内容。
  - 未指定默认字符集时，模式的默认字符集为指定的字符序对应的字符集。

- **SET NODE GROUP group_name | DROP NODE GROUP**

版权所有 © 华为云计算技术有限公司

- SET NODE GROUP group_name：表示设置模式的单节点属性。group_name表示单节点组Node Group名称，仅支持设置系统预定的类型为's'的单节点组Node Group。
- DROP NODE GROUP：表示删除模式的单节点属性。

说明

在M-Compatibility模式数据库下（即sql_compatibility = 'M'）不支持该语法。

示例

- 修改模式的防篡改属性

```
--创建模式test_schema1。
gaussdb=# CREATE SCHEMA test_schema1;

--修改模式test_schema1为防篡改模式。
gaussdb=# ALTER SCHEMA test_schema1 WITH BLOCKCHAIN;

--查询模式信息，防篡改属性为true。
gaussdb=# \dn+ test_schema1
                          List of schemas
     Name     | Owner | Access privileges | Description | WithBlockChain
--------------+-------+-------------------+-------------+----------------
 test_schema1 | omm   |                   |             | t
(1 row)
```

- 修改模式名称

```
--将模式test_schema1重命名为test_sch1。
gaussdb=# ALTER SCHEMA test_schema1 RENAME TO test_sch1;

--查询模式信息。
gaussdb=# \dn+ test*
                         List of schemas
   Name    | Owner | Access privileges | Description | WithBlockChain
-----------+-------+-------------------+-------------+----------------
 test_sch1 | omm   |                   |             | t
(1 row)
```

- 修改模式所有者

```
--创建用户test_user。
gaussdb=# CREATE ROLE test_user PASSWORD '********';

--将模式test_sch1的所有者修改为test_user。
gaussdb=# ALTER SCHEMA test_sch1 OWNER TO test_user;

--查询模式信息。
gaussdb=# \dn+  test_sch1;
                          List of schemas
   Name    |   Owner   | Access privileges | Description | WithBlockChain
-----------+-----------+-------------------+-------------+----------------
 test_sch1 | test_user |                   |             | t
(1 row)

--删除新建的模式和用户。
gaussdb=# DROP SCHEMA test_sch1;
gaussdb=# DROP ROLE test_user;
```

- 修改默认字符集和默认字符序

须知

除binary字符集和字符序外，当前仅支持指定与数据库编码相同的字符集。

```
--创建并切换至测试数据库。
gaussdb=# CREATE DATABASE test1 WITH DBCOMPATIBILITY = 'MYSQL' ENCODING = 'UTF8'
```


版权所有 © 华为云计算技术有限公司

```
LC_COLLATE = 'zh_CN.utf8' LC_CTYPE = 'zh_CN.utf8';
gaussdb=# \c test1

--创建模式test_sch2。
test1=# CREATE SCHEMA test_sch2;
--修改其默认字符集为utf8mb4，默认字符序修改为utf8mb4_bin。
test1=# ALTER SCHEMA test_sch2 CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;

--删除新建的模式。
test1=# DROP SCHEMA test_sch2;

--切换至默认数据库（根据实际情况修改数据库名），并删除新建的数据库。
test1=# \c postgres
gaussdb=# DROP DATABASE test1;
```

- 修改模式的单节点属性。

```
--创建模式test_schema1。
gaussdb=# CREATE SCHEMA test_schema1;

--修改设置模式test_schema1单节点属性。
gaussdb=# ALTER SCHEMA test_schema1 SET NODE GROUP sys_single_group1;

--查询模式test_schema1单节点属性,pg_namespace系统表的nspgroupoid记录单节点属性对应node group的oid。
gaussdb=# SELECT * FROM pg_namespace WHERE nspname='test_schema1';
   nspname    | nspowner | nsptimeline | nspacl | in_redistribution | nspblockchain | nspcollation | nspgroupoid
--------------+----------+-------------+--------+-------------------+---------------+--------------+-------------
 test_schema1 |       10 |           0 |        | n                 | f             |              |       16389
(1 row)

--删除模式test_schema1单节点属性。
gaussdb=# ALTER SCHEMA test_schema1 DROP NODE GROUP;

--查询模式test_schema1单节点属性。
gaussdb=# SELECT * FROM pg_namespace WHERE nspname='test_schema1';
   nspname    | nspowner | nsptimeline | nspacl | in_redistribution | nspblockchain | nspcollation | nspgroupoid
--------------+----------+-------------+--------+-------------------+---------------+--------------+-------------
 test_schema1 |       10 |           0 |        | n                 | f             |              |
(1 row)
```

相关链接

CREATE SCHEMA，**DROP SCHEMA**

1.13.7.29 ALTER SEQUENCE

功能描述

修改指定序列的参数。

注意事项

- 序列的所有者、被授予了序列ALTER权限的用户或者被授予了ALTER ANY SEQUENCE权限的用户才能执行ALTER SEQUENCE命令，三权分立开关关闭时，系统管理员默认拥有该权限。但要修改序列的所有者，当前用户必须是该序列的所有者或者系统管理员，且该用户是新所有者角色的成员。
- 当前版本仅支持修改拥有者、归属列和最大值。若要修改其他参数，可以删除重建，并用setval函数恢复当前值。
- ALTER SEQUENCE MAXVALUE不支持在事务、函数和存储过程中使用。

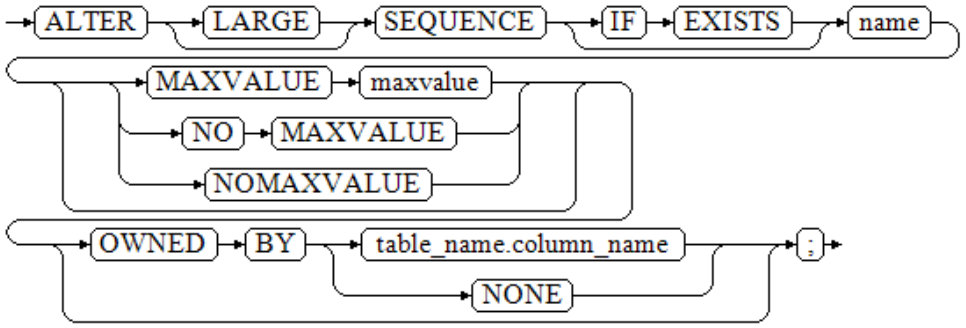
版权所有 © 华为云计算技术有限公司

- 修改序列的最大值后，会清空该序列在所有会话的缓存值。
- 如果创建序列时指定了LARGE标识，则在进行ALTER操作时，也需指定LARGE标识。
- ALTER SEQUENCE会阻塞nextval、setval、currval和lastval的调用。

语法格式

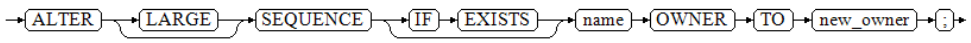
- 修改序列归属列和最大值

```
ALTER [LARGE] SEQUENCE [ IF EXISTS ] name
    [MAXVALUE maxvalue | NO MAXVALUE | NOMAXVALUE]
    [ OWNED BY { table_name.column_name | NONE } ] ;
```



- 修改序列的拥有者

```
ALTER [LARGE] SEQUENCE [ IF EXISTS ] name OWNER TO new_owner;
```



参数说明

- **LARGE**

  LARGE标识，如果创建序列时指定了LARGE标识，则在进行ALTER操作时，也需指定LARGE标识。

- **name**

  将要修改的序列名称。

- **IF EXISTS**

  当序列不存在时使用该选项，不会显示ERROR，而是返回一个NOTICE信息。

- **MAXVALUE maxvalue | NO MAXVALUE | NOMAXVALUE**

  执行序列的最大值。新修改的最大值必须大于当前gtm存储的最大值；如果没有指定，将保持原有的最大值。

  取值范围：(gtm_last_value, 2^{63}-1]，gtm_last_value表示序列中已经被申请的序列最大值，可以使用“\d 序列名”命令进行查询。若创建序列时指定了LARGE标识，取值范围应为(gtm_last_value, 2^{127}-1]。

- **OWNED BY { table_name.column_name | NONE }**

  将序列和一个表的指定字段进行关联，当进行删除该字段或其所在表的时候，会自动删除已关联的序列。

  - 如果序列已经和表有关联后，使用这个选项后新的关联关系会覆盖旧的关联。
  - 关联的表和序列的所有者必须是同一个用户，并且在同一个模式中。
  - 使用OWNED BY NONE将删除任何已经存在的关联。

版权所有 © 华为云计算技术有限公司

- **new_owner**

  序列新所有者的用户名。用户要修改序列的所有者，必须是新角色的直接或者间接成员，并且该角色必须有序列所在模式上的CREATE权限。

示例

```
--创建一个名为serial的递增序列，从101开始。
gaussdb=# CREATE SEQUENCE serial START 101;

--创建一个表,定义默认值。
gaussdb=# CREATE TABLE t1(c1 bigint default nextval('serial'));

--将序列serial的归属列变为t1.c1。
gaussdb=# ALTER SEQUENCE serial OWNED BY t1.c1;

--删除序列和表。
gaussdb=# DROP SEQUENCE serial CASCADE;
gaussdb=# DROP TABLE t1;

--创建一个序列seq_test。
gaussdb=# CREATE SEQUENCE seq_test;

--查询序列信息。
gaussdb=# \ds
              List of relations
 Schema |   Name   |   Type   | Owner | Storage
--------+----------+----------+-------+---------
 public | seq_test | sequence | omm   |
(1 row)

--创建用户u_test并修改序列所有者。
gaussdb=# CREATE USER u_test PASSWORD '********';
gaussdb=# ALTER SEQUENCE seq_test OWNER TO u_test;
gaussdb=# \ds
               List of relations
 Schema |   Name   |   Type   | Owner  | Storage
--------+----------+----------+--------+---------
 public | seq_test | sequence | u_test |
(1 row)

--删除用户和序列。
gaussdb=# DROP SEQUENCE seq_test;
gaussdb=# DROP USER u_test;
```

相关链接

CREATE SEQUENCE，**DROP SEQUENCE**

1.13.7.30 ALTER SERVER

功能描述

增加、修改和删除指定服务器的参数，可以从pg_foreign_server系统表中查询已有的服务器。

注意事项

- 只有SERVER的所有者或者被授予了SERVER的ALTER权限的用户才可以执行ALTER SERVER命令，系统管理员默认拥有该权限。但要修改SERVER的所有者，当前用户必须是该SERVER的所有者或者系统管理员，且该用户是新所有者角色的成员。
- OPTIONS中的敏感字段（如password、secret_access_key）在使用多层引号时，语义和不带引号的场景是不同的，因此不会被识别为敏感字段进行脱敏。

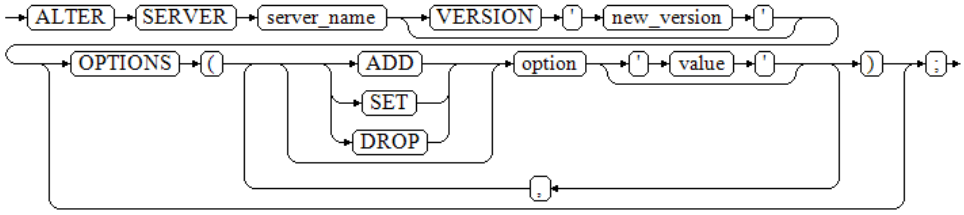
版权所有 © 华为云计算技术有限公司

语法格式

- 修改外部服务的参数

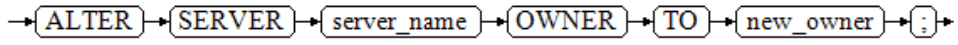
```
ALTER SERVER server_name [ VERSION 'new_version' ]
    [ OPTIONS ( {[ ADD | SET | DROP ] option ['value']} [, ... ] ) ];
```

在OPTIONS选项里，ADD、SET和DROP指定要执行的操作，未指定时默认为ADD操作。option和value为对应操作的参数。



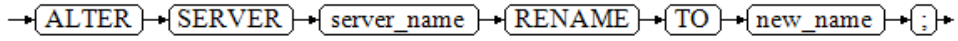
- 修改外部服务的所有者

```
ALTER SERVER server_name
    OWNER TO new_owner;
```



- 修改外部服务的名称

```
ALTER SERVER server_name
    RENAME TO new_name;
```



参数说明

- **server_name**

  需要修改的外部服务器的名称。

- **new_version**

  修改后，外部服务器的新版本名称。

- **OPTIONS**

  更改该服务器的选项，选项名需满足唯一约束。ADD、SET和DROP为指定执行的动作，如果未指定，默认为ADD。名称和值也会使用该服务器的外部数据封装器库进行验证。

  除了libpq支持的连接参数外，还额外提供以下参数：

  - **fdw_startup_cost**

    执行一个外表扫描时的启动耗时估算。该值通常包含建立连接、远端对请求的分析和生成计划的耗时。默认值为100，取值范围为大于0的实数。

  - **fdw_tuple_cost**

    在远端服务器上对每一个元组进行扫描时的额外消耗。该值通常表示数据在server间传输的额外消耗。默认值为0.01，取值范围为大于0的实数。

- **new_owner**

  修改后服务器的新拥有者。

  说明

  更改服务器的所有者，当前用户必须是外部服务器的所有者并且也是新的所有者角色的直接或者间接成员，并且该用户须对外部服务器的外部数据封装器有USAGE权限。

- **new_name**

版权所有 © 华为云计算技术有限公司

修改后服务器的新名称。

示例

```
--创建my_server。
gaussdb=# CREATE SERVER my_server FOREIGN DATA WRAPPER log_fdw;

--修改外部服务器的名称。
gaussdb=# ALTER SERVER my_server RENAME TO my_server_1;

--创建用户。
gaussdb=# CREATE USER jack PASSWORD '********';

--修改外部服务器的所有者。
gaussdb=# ALTER SERVER my_server_1 OWNER TO jack;

--增加外部服务器中dbname、dbuser信息。
gaussdb=# ALTER SERVER my_server_1 OPTIONS(ADD dbname 'test',dbuser' test_user');

--查询外部服务器信息。
gaussdb=# SELECT * FROM pg_foreign_server WHERE srvname='my_server_1';
   srvname   | srvowner | srvfdw | srvtype | srvversion | srvacl |            srvoptions
-------------+----------+--------+---------+------------+--------+----------------------------------
 my_server_1 |    22060 |  11826 |         |            |        | {dbname=test,"dbuser= test_user"}
(1 row)

--修改外部服务器中dbname名称为‘db_test’。
gaussdb=# ALTER SERVER my_server_1 OPTIONS (SET dbname 'db_test');

--查询外部服务器信息。
gaussdb=# SELECT srvname,srvoptions FROM pg_foreign_server WHERE srvname='my_server_1';
   srvname   |             srvoptions
-------------+-----------------------------------
 my_server_1 | {dbname=db_test,"dbuser= test_user"}
(1 row)

--删除外部服务器中dbname信息。
gaussdb=# ALTER SERVER my_server_1 OPTIONS (DROP dbname);

--查询外部服务器信息。
gaussdb=# SELECT srvname,srvoptions FROM pg_foreign_server WHERE srvname='my_server_1';
   srvname   |     srvoptions
-------------+--------------------
 my_server_1 | {"dbuser= test_user"}
(1 row)

--删除my_server_1。
gaussdb=# DROP SERVER my_server_1;

--删除用户。
gaussdb=# DROP USER jack;
```

相关链接

CREATE SERVER，**DROP SERVER**

1.13.7.31 ALTER SESSION

功能描述

定义或修改对当前会话有影响的条件或参数。修改后的会话参数会一直保持，直到断开当前会话。

版权所有 © 华为云计算技术有限公司

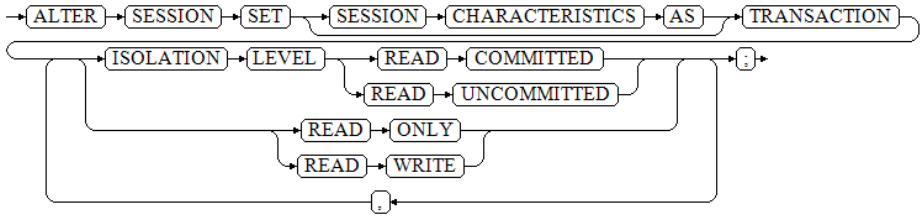
注意事项

- 如果执行SET TRANSACTION之前没有执行START TRANSACTION，则事务立即结束，命令无法显示效果。
- 可以用START TRANSACTION里面声明所需要的transaction_mode(s)的方法进行避免使用SET TRANSACTION。具体请参见**START TRANSACTION**。

语法格式

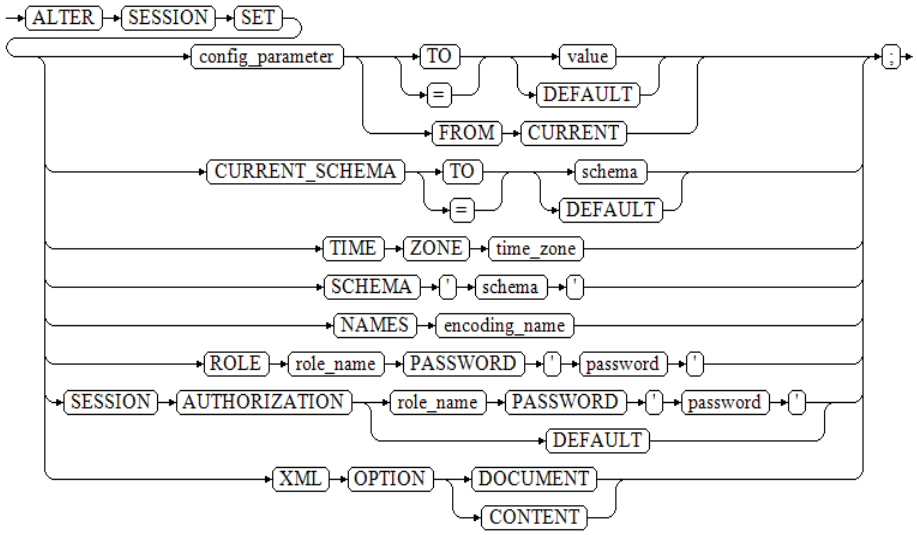
- 设置会话的事务参数

```
ALTER SESSION SET [ SESSION CHARACTERISTICS AS ] TRANSACTION
   { ISOLATION LEVEL { READ COMMITTED | READ UNCOMMITTED } | { READ ONLY  | READ
WRITE } } [, ...] ;
```



- 设置会话的其他运行时参数

```
ALTER SESSION SET
   {{config_parameter { { TO  | =  }  { value | DEFAULT } | FROM CURRENT }}
     | CURRENT_SCHEMA { TO | = } { schema | DEFAULT }
     | TIME ZONE time_zone
     | SCHEMA 'schema'
     | NAMES encoding_name
     | ROLE role_name PASSWORD 'password'
     | SESSION AUTHORIZATION { role_name PASSWORD 'password' | DEFAULT }
     | XML OPTION { DOCUMENT | CONTENT }
   } ;
```



参数说明

- **config_parameter**

  可设置的运行时参数的名称。可用的运行时参数可以使用SHOW ALL命令查看。

  - **value**

    config_parameter的新值，可以声明为字符串常量、标识符、数字，或者逗号分隔的列表。

版权所有 © 华为云计算技术有限公司

    - DEFAULT
    - OFF
    - RESET
    - 用户指定的值：需要满足修改参数的取值限制。
  - **DEFAULT**

    用于将config_parameter参数设置为其对应的默认值。
  - **FROM CURRENT**

    取当前会话中的值设置为configuration_parameter的值。
- **CURRENT_SCHEMA**

  用于指定当前的模式。

  取值范围：已存在模式名称。如果模式名不存在，会导致CURRENT_SCHEMA值为空。
- **TIME ZONE timezone**

  用于指定当前会话的本地时区。

  取值范围：有效的本地时区。该选项对应的运行时参数名称为TimeZone，DEFAULT缺省值为PRC。
- **SCHEMA schema**

  同CURRENT_SCHEMA。此处的schema为字符串。
- **NAMES encoding_name**

  用于设置客户端的字符编码。等价于SET client_encoding TO encoding_name。

  取值范围：有效的字符编码。该选项对应的运行时参数名称为client_encoding，默认编码为UTF8。
- **role_name**

  角色的名称。

  取值范围：字符串，要符合**标识符命名规范**。
- **password**

  角色的密码。

  取值范围：要求符合密码的命名规则。
- **SESSION AUTHORIZATION**

  当前会话的用户表识符。
- **XML OPTION { DOCUMENT | CONTENT }**

  用于设置XML的解析方式。

  取值范围：CONTENT（默认值）、DOCUMENT。
- **ISOLATION LEVEL**

  指定事务隔离级别，事务的隔离级别决定了该事务在存在并发事务时，可以读取到哪些数据（即数据的可见性）。

  取值范围：
  - READ COMMITTED（默认值）：读已提交隔离级别，只能读到已经提交的数据，而不会读到未提交的数据。

版权所有 © 华为云计算技术有限公司

- READ UNCOMMITTED：读未提交隔离级别，指定后的行为和READ COMMITTED行为一致。

示例

- 设置会话的事务参数。

  示例中关键字ALTER SESSION可以省略。

```
--开启事务，设置事务级别。
gaussdb=# START TRANSACTION;
gaussdb=# ALTER SESSION SET TRANSACTION READ ONLY;
gaussdb=# END;
```

- 设置会话的其他运行时参数。

  示例中关键字ALTER SESSION可以省略。

```
--创建模式ds。
gaussdb=# CREATE SCHEMA ds;

--设置模式搜索路径。
gaussdb=# SET SEARCH_PATH TO ds, public;

--设置日期时间风格为传统的POSTGRES风格（日在月前）。
gaussdb=# SET DATESTYLE TO postgres, dmy;

--设置当前会话的字符编码为UTF8。
gaussdb=# ALTER SESSION SET NAMES 'UTF8';

--设置时区为加州伯克利。
gaussdb=# SET TIME ZONE 'PST8PDT';

--设置时区为意大利。
gaussdb=# SET TIME ZONE 'Europe/Rome';

--设置当前模式。
gaussdb=# ALTER SESSION SET CURRENT_SCHEMA TO tpcds;

--设置XML OPTION为DOCUMENT。
gaussdb=# ALTER SESSION SET XML OPTION DOCUMENT;

--创建角色joe，并设置会话的角色为joe。
gaussdb=# CREATE ROLE joe WITH PASSWORD '********';
gaussdb=# ALTER SESSION SET SESSION AUTHORIZATION joe PASSWORD '********';

--切换到默认用户。
gaussdb=> ALTER SESSION SET SESSION AUTHORIZATION default;

--删除ds模式。
gaussdb=# DROP SCHEMA ds;

--删除joe。
gaussdb=# DROP ROLE joe;
```

相关链接

SET

1.13.7.32 ALTER SYNONYM

功能描述

修改同义词对象的属性。

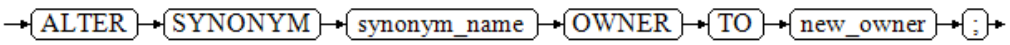
版权所有 © 华为云计算技术有限公司

注意事项

- 目前仅支持修改同义词对象的所有者。
- 只有系统管理员有权限修改同义词对象的所有者信息。当三权分立开关打开时，系统管理员默认没有权限修改同义词对象的所有者。
- 新所有者必须具有同义词对象所在模式的CREATE权限。
- 不支持修改PUBLIC同义词。

语法格式

```
ALTER SYNONYM synonym_name
    OWNER TO new_owner;
```



参数描述

- **synonym_name**

  待修改的同义词名称，可以使用模式进行具体指定。

  取值范围：字符串，要符合**标识符命名规范**。

- **new_owner**

  同义词对象的新所有者。

  取值范围：字符串，有效的用户名。

示例

```
--创建系统管理员用户。
gaussdb=# CREATE USER sysadmin WITH SYSADMIN PASSWORD '********';

--切换管理员用户。
gaussdb=#  \c - sysadmin

--创建同义词t1。
gaussdb=#  CREATE OR REPLACE SYNONYM t1 FOR ot.t1;

--创建新用户u1。
gaussdb=# CREATE USER u1 PASSWORD '********';

--给新用户赋权限。
gaussdb=# GRANT ALL ON SCHEMA sysadmin TO u1;

--修改同义词t1的owner为u1。
gaussdb=# ALTER SYNONYM t1 OWNER TO u1;

--删除同义词t1。
gaussdb=# DROP SYNONYM t1;

--收回用户u1权限。
gaussdb=# REVOKE ALL ON SCHEMA sysadmin FROM u1;

--删除用户u1。
gaussdb=# DROP USER u1;

--切换到初始用户init_user，请使用真实的初始用户名称替换init_user。
gaussdb=# \c - init_user

--删除用户sysadmin。
gaussdb=# DROP USER sysadmin;
```


版权所有 © 华为云计算技术有限公司

相关链接

CREATE SYNONYM，DROP SYNONYM

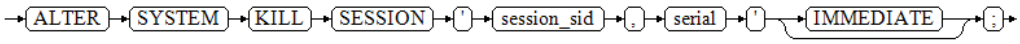
1.13.7.33 ALTER SYSTEM KILL SESSION

功能描述

ALTER SYSTEM KILL SESSION命令用于结束一个会话。

语法格式

```
ALTER SYSTEM KILL SESSION 'session_sid, serial' [ IMMEDIATE ];
```



参数说明

- **session_sid, serial**

  会话的SID和SERIAL（格式请参见示例）。

  可通过pg_stat_activity系统表配合查询当前活跃线程（可见示例），但执行ALTER SYSTEM KILL SESSION命令时线程可能已结束。

  取值范围：通过查看系统表dv_sessions可查看所有会话的SID和SERIAL。

- **IMMEDIATE**

  表明会话将在命令执行后立即结束。

示例

```
--开启两个会话，在第一个会话中建表并开启事务插入数据。
gaussdb=# CREATE TABLE tbl_test(id int);
gaussdb=# BEGIN;
gaussdb=# INSERT INTO tbl_test VALUES (1);

--在第二个会话中查询会话信息。state为“idle in transaction”表示事务等待提交。
gaussdb=# SELECT t1.datname,
       t1.usename,
       t1.pid,
       t2.serial#,
       t1.state
FROM pg_stat_activity t1,
     dv_sessions t2
WHERE t1.query LIKE 'INSERT INTO tbl_test%'
  AND t1.sessionid = t2.sid;
 datname  | usename |       pid       | serial# |        state
----------+---------+-----------------+---------+---------------------
 postgres | omm     | 139802072635136 |       0 | idle in transaction
(1 row)

--结束会话，不带IMMEDIATE参数，将会强制结束会话，该会话中的事务也会被强制结束。
gaussdb=# ALTER SYSTEM KILL SESSION '139802072635136,0';
 pg_terminate_backend
----------------------
 t
(1 row)

--重新连接，查询表tbl_test的数据会发现事务被强制结束后数据回滚。
gaussdb=# SELECT * FROM tbl_test;
 id
----
```


版权所有 © 华为云计算技术有限公司

```
(0 rows)

--删除表。
gaussdb=# DROP TABLE tbl_test;
```

1.13.7.34 ALTER TABLE

功能描述

修改表的属性。包括：修改表的定义、重命名表、重命名表中指定的列、重命名表的约束、设置表的所属模式、添加/更新多个列以及打开/关闭行访问控制开关等。

注意事项

- 基表为Hash分布时，若创建不包含基表分布键的主键索引或唯一索引，需要使用全局二级索引（指定BY GLOBAL INDEX字段）；若创建包含基表分布键的主键索引或唯一索引，需要使用普通索引（不指定BY GLOBAL INDEX字段）。单DN部署形式下，使用全局二级索引或者普通索引均可创建不包含基表分布键的主键索引或唯一索引；当基表为非Hash分布形式时，只能使用普通索引创建不包含基表分布键的主键索引或唯一索引，即索引键必须包含基表分布键。
- 表的所有者、被授予了表ALTER权限的用户或被授予ALTER ANY TABLE权限的用户有权限执行ALTER TABLE命令，系统管理员默认拥有此权限。但要修改表的所有者或者修改表的模式，当前用户必须是该表的所有者或者系统管理员，且在三权分立打开时，该用户必须是新所有者角色的成员。
- 不能修改分区表的表空间，但可以修改分区的表空间。
- 不支持修改存储参数ORIENTATION。
- SET SCHEMA操作不支持修改为系统内部模式，当前仅支持用户模式之间的修改。
- 不允许对表的分布列（distribute column）进行修改。
- 不支持对分区表的分区键字段改变和转换字符集。
- 不支持增加自增列，或者增加DEFAULT值中包含NEXTVAL()表达式的列。
- rowid和rowno系统列不支持重命名、删除、设置默认值、修改列定义。
- 不支持对外表、临时表开启行访问控制开关。
- 不支持对系统列添加约束。
- 通过约束名删除PRIMARY KEY约束时，不会删除NOT NULL约束，如果有需要，请手动删除NOT NULL约束。
- 使用JDBC时，支持通过PrepareStatement对DEFAULT值进行参数化设置。
- 如果用ADD COLUMN增加一个字段，表中已有的行的新增字段值均为DEFAULT子句指定的默认值。如果没有指定DEFAULT值，则已有行的新字段的值缺省为NULL。
- 当新增列的字段默认值为NULL时，不会触发全表更新。
- 新增列如果指定DEFAULT值，必须符合以下所有要求，否则会产生额外的全表更新开销，影响在线业务：
  - 新增列的DEFAULT值长度不超过128个字节。
  - 新增列的DEFAULT值不包含易变（volatile）函数，可通过PG_PROC系统表查看函数的provolatile属性是否为v进行判断是否存在volatile函数。
- 在为数据对象增加或者变更ILM策略时，如果追加了行表达式，仅支持白名单中列出的函数作为行表达式。具体白名单函数列表请参见**行表达式函数白名单**。

版权所有 © 华为云计算技术有限公司

说明

在ILM策略的ON（EXPR）行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。

- 表约束个数不能超过32767个。
- 执行在线DDL操作时，需预留足够的磁盘空间，即剩余空间是所操作的表（及其上索引等附属品）总和。
- 在线DDL不支持在事务内执行（通过JDBC驱动执行多条SQL时会隐式启动事务，禁止多条SQL一起执行），不支持在存储过程中执行。
- 如果用户取消在线DDL执行过程，首先进入残留清理流程，如果用户再次取消，则会结束清理流程导致残留。
- 在线DDL与用户业务之间存在资源争抢，不适合大业务背景下进行在线DDL。对于资源有限的场景，建议采用细粒度资源管控进行操作，且不建议使用并行加速功能。
- M-Compatibility模式数据库下不支持在线DDL操作。
- 以下情况中，在线DDL操作将不会生效，DDL操作默认以离线方式执行，如果指定ONLINE关键字，则会产生请不要指定ONLINE关键字的提示。
  - 使用MODIFY column_name data_type { { [COMMENT 'string'] [...] } | AUTO_INCREMENT }语法或MODIFY [COLUMN] column_name data_type [CHARACTER SET |CHARSET charset] [{[COLLATE collation] | [column_constraint]} [...]]语法在线修改列类型、精度或长度，子句包含COMMENT、AUTO_INCREMENT、CHARACTER SET、CHARSET、COLLATE等任一关键字或column_constraint。
  - 使用CHANGE [COLUMN] column_name new_column_name data_type[CHARACTER SET | CHARSET charset] [{[COLLATE collation] |[column_constraint]} [...]]语法在线修改列类型、精度或长度。
  - 在线将列类型修改或转换为SET数据类型。
  - 若ALTER TABLE语句包含当前在线DDL特性支持操作（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查约束、在线添加主键约束、在线添加唯一约束）以外的操作。
  - 在线DDL（包含在线修改列类型、精度、长度以及表级在线移动表空间）操作的表上包含GSI或语句中包含BY GLOBAL INDEX关键字。
- 对数据库级、索引级、二级分区表、段页式表、hash bucket表、临时表以及unlogged表进行在线DDL操作时（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查约束、在线添加主键约束、在线添加唯一约束），在线DDL操作将不会生效，DDL操作默认以离线方式执行，并提示当前DDL语句不支持在线执行。
- 长事务可能会阻塞在线DDL执行结果（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查、在线添加主键约束以及在线添加唯一约束），建议避免在存在长事务时进行在线DDL操作。
- 在线DDL（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查约束、在线添加主键约束以及在线添加唯一约束）本特性执行后期将阻塞DQL、DML操作。因此，对于存在超时报错机制的业务模型可能出现阻塞导致的请求失败，若出现此类情况，可以进行重试。
- 在线DDL执行期间，应尽量避免与其他DDL并发执行，否则可能会出现无响应以及报错等异常，请重新连接客户端进行重试。

版权所有 © 华为云计算技术有限公司

- 从不支持在线DDL操作的版本升级至支持在线DDL操作的版本，升级待观察期间不支持使用在线DDL操作（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查约束、在线添加主键约束以及在线添加唯一约束），DDL默认以离线方式执行，并提示升级期间不支持在线DDL操作。
- 在线DDL执行（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查约束、在线添加主键约束以及在线添加唯一约束）过程中会生成指定的Schema，Schema命名格式为online$$ddl$$[(*表所在的Schema 表名*)的hash值]。
  - Schema的属主默认为初始用户，不建议在线DDL执行中操作此Schema，可能会导致中止在线DDL并产生未知错误。不建议用户对该Schema下的表格进行查询，gs_dump导出会默认不导出该Schema。
  - 用户不可创建online$$为前缀的Schema，否则会产生报错提示。若已存在同名Schema冲突，在线DDL操作将不会生效，DDL默认以离线方式执行，并提示Schema名冲突导致当前DDL语句不可在线。
- 若在线DDL（包含在线修改列类型、精度、长度以及表级在线移动表空间）操作失败或者执行在线DDL（在线添加非空约束、在线添加检查、在线添加主键约束以及在线添加唯一约束），表上可能会存在明显存储空间膨胀，其中膨胀程度与DDL执行时长以及并发写操作数成正比，建议避免在存在长事务时进行在线DDL操作 。
- 若在线DDL操作包含添加非空约束、添加检查、添加主键约束或添加唯一约束操作，当出现违反约束的数据进行报错时，报错信息会存在区别于离线DDL的情况。
- 在线DDL操作涉及在线添加主键以及在线添加唯一约束时，若约束包含表达式，则默认以离线DDL的方式执行，若指定了ONLINE关键字，会提示当前DDL语句不可在线。
- 在线DDL（包含在线修改列类型、精度、长度、表级在线移动表空间、在线添加非空约束、在线添加检查、在线添加主键约束以及在线添加唯一约束）操作继承**CREATE TABLE LIKE**的使用约束。
- 避免同时执行在线DDL操作和扩缩容操作，否则可能会出现在线DDL或扩缩容执行失败的情况。

语法格式

- 修改表的定义。

```
ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
 { table_name [*] | ONLY table_name | ONLY ( table_name ) }
    action [, ... ];

ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
table_name
    ADD ( { column_name data_type [ COLLATE collation ] [ column_constraint [ ... ] ] } [, ... ] );

ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
table_name
    MODIFY ( { column_name data_type | column_name [ CONSTRAINT constraint_name ] NOT NULL
[ ENABLE ] | column_name [ CONSTRAINT constraint_name ] NULL } [, ... ] );

ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
table_name
    RENAME [TO | AS | = ] new_table_name;
ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
    { table_name [*] | ONLY table_name | ONLY ( table_name ) }
    RENAME [ COLUMN ] column_name TO new_column_name;

ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
    { table_name [*] | ONLY table_name | ONLY ( table_name ) }
```


版权所有 © 华为云计算技术有限公司

```
    RENAME CONSTRAINT constraint_name TO new_constraint_name;

ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
table_name
    SET SCHEMA new_schema;

ALTER TABLE  [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
table_name
    GSIWAITALL;
```

其中表操作action为：

```
{ distribute_clause
    [ DISTRIBUTE BY { REPLICATION | HASH ( column_name [,...] )
    | RANGE ( column_name [,...] ) { SLICE REFERENCES tablename | ( slice_less_than_item [,...] )
    | ( slice_start_end_item [,...] ) }
    | LIST ( column_name [,...] ) { SLICE REFERENCES tablename | ( slice_values_item [,...] ) }
    } ]
    [ TO { GROUP groupname | NODE ( nodename [,...] ) } ]
column_clause
    | ADD table_constraint [ NOT VALID ]
    | ADD table_constraint_using_index
    | VALIDATE CONSTRAINT constraint_name
    | DROP CONSTRAINT [ IF EXISTS ]  constraint_name [ RESTRICT | CASCADE ]
    | DROP PRIMARY KEY
    | CLUSTER ON index_name
    | SET WITHOUT CLUSTER
    | SET WITH ROWID
    | SET ( {storage_parameter = value} [, ... ] )
    | RESET ( storage_parameter [, ... ] )
    | OWNER TO new_owner
    | SET TABLESPACE new_tablespace
    | SET { COMPRESS | NOCOMPRESS }
    | ADD NODE ( nodename [, ... ] )
    | DELETE NODE ( nodename [, ... ] )
    | UPDATE SLICE LIKE table_name
    | DISABLE TRIGGER { trigger_name | ALL | USER }
    | ENABLE TRIGGER { trigger_name | ALL | USER }
    | ENABLE REPLICA TRIGGER trigger_name
    | ENABLE ALWAYS TRIGGER trigger_name
    | ENABLE ROW LEVEL SECURITY
    | DISABLE ROW LEVEL SECURITY
    | FORCE ROW LEVEL SECURITY
    | NO FORCE ROW LEVEL SECURITY
    | ENCRYPTION KEY ROTATION
    | REPLICA IDENTITY { DEFAULT | USING INDEX index_name | FULL | NOTHING }
    | AUTO_INCREMENT [ = ] value
    | COMMENT [ = ] 'string'
    | [ [ DEFAULT ] CHARACTER SET | CHARSET [ = ] default_charset ] [ [ DEFAULT ] COLLATE [ = ]
default_collation ]
    | ILM ADD POLICY ROW STORE COMPRESS ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day |
days | month | months | year | years } OF NO MODIFICATION [ ON ( EXPR )]
    | [ MODIFY { PARTITION partition_name | SUBPARTITION subpartition_name }] ILM { ENABLE |
DISABLE | DELETE } POLICY policy_name
    | [ MODIFY { PARTITION partition_name | SUBPARTITION subpartition_name }] ILM { ENABLE_ALL
| DISABLE_ALL | DELETE_ALL}
    | [ COLVIEW | COLVIEW NOCOLVIEW | NOCOLVIEW COLVIEW] [ ( { column_name [, ... ]} ) ]
[ PRIORITY { HIGH | LOW | NONE } ]
    | NOCOLVIEW }
```


版权所有 © 华为云计算技术有限公司

说明

- **ADD table_constraint [NOT VALID]**

  为表增加新的约束。

- **ADD table_constraint_using_index**

  根据已有唯一索引为表增加主键约束或唯一约束。目前不支持使用已有GSI索引为表进行ADD table_constraint_using_index操作。

- **VALIDATE CONSTRAINT constraint_name**

  用于对使用NOT VALID选项创建的检查类约束的验证，通过扫描全表以保证所有记录都符合约束条件。如果约束已标记为有效时，该语法将不会对约束行为进行改变。

- **DROP CONSTRAINT [IF EXISTS] constraint_name [RESTRICT | CASCADE]**

  删除表上的约束。

- **DROP PRIMARY KEY**

  删除表上的主键约束。该语法仅在MYSQL兼容模式以及ORA兼容模式下支持使用。

- **CLUSTER ON index_name**

  为CLUSTER聚簇操作选择默认索引。实际上并未对该表重新进行盘簇化处理。

- **SET WITHOUT CLUSTER**

  从表中删除最新使用的CLUSTER索引。该语法会影响未声明默认索引的CLUSTER聚簇操作。

- **SET WITH ROWID**

  为表添加rowid、rowno系统列，并设置表的存储选项hasrowid = on。

  - 该语法不支持用于系统表、本地临时表、全局临时表、Unlogged表、外表、hashbucket表、复制表。
  - 该语法不支持用于启用oid系统列或gs_tuple_uid系统列的表，即指定oids存储选项（或WITH OIDS子句）、hasuids存储选项的表。
  - 该语法不支持用于包含列名为rowid或者rowno的表。
  - 该语法仅在ORA兼容性下有效。
  - 对处于压缩状态的数据执行SET WITH ROWID操作后，该部分数据将被自动解压。

- **SET ({storage_parameter = value} [, ...])**

  修改表的一个或多个存储参数。

  当table_name为索引名时，ACTIVE_PAGES表示索引的页面数量，可能比实际的物理文件页面少，可以用于优化器调优。目前只对Ustore的分区表LOCAL索引生效，且会被VACUUM、ANALYZE更新（包括AUTO VACUUM）。不建议用户手动设置该参数，该参数在分布式下无效。

- **RESET (storage_parameter [, ...])**

  重置表的一个或多个存储参数。与SET一样，根据参数的不同可能需要重新修改表才能获得预期的效果。

- **OWNER TO new_owner**

  将表、序列或视图的属主修改为指定的用户。

  以下操作不被允许，若执行将会报错：

  1.系统管理员将表的owner更改为初始用户；

  2.三权分立模式下，系统管理员将表的owner更改为安全管理员或审计管理员。

- **SET TABLESPACE new_tablespace**

  将表空间修改为指定的表空间并将相关的数据文件移动到指定的表空间。表上的所有索引都不会被移动，索引的表空间可以通过ALTER INDEX语法的SET TABLESPACE选项来修改。

- **DISTRIBUTE BY { REPLICATION | HASH (column_name [, ...]) | RANGE (column_name [, ...]) { SLICE REFERENCES tablename | (slice_less_than_item**

版权所有 © 华为云计算技术有限公司

[, ...]) | (slice_start_end_item [, ...]) } | LIST (column_name [, ...]) { SLICE REFERENCES tablename | (slice_values_item [, ...]) }}

指定表如何在节点之间分布或者复制，具体请参见**CREATE TABLE**中关于DISTRIBUTE BY的介绍。

- **TO { GROUP groupname | NODE (nodename [, ...]) }**

  仅在扩展模式（GUC参数support_extended_features为on时）下可用。该模式谨慎打开，主要为内部扩容工具使用，不建议用户使用该模式。该命令仅修改表分布节点的逻辑映射关系，并未在DN节点上迁移表的元数据和数据。

- **ADD NODE (nodename [, ...])**

  该语法主要为内部扩容工具使用，不建议用户直接使用。

- **DELETE NODE (nodename [, ...])**

  该语法主要为内部缩容工具使用，不建议用户直接使用。

- **UPDATE SLICE LIKE table_name**

  该语法主要为内部扩缩容工具使用，不建议用户直接使用。

- **DISABLE TRIGGER { trigger_name | ALL | USER }**

  禁用指定触发器。trigger_name表示禁用指定的单个触发器，ALL表示禁用所有触发器，USER表示仅禁用用户触发器（此选项不包括内部生成的约束触发器，例如，可延迟唯一性和排除约束的约束触发器）。

  请谨慎使用此功能，一旦执行禁用触发器，原先期望的约束的完整性将无法保证。

- **ENABLE TRIGGER { trigger_name | ALL | USER }**

  启用指定触发器。trigger_name表示启用指定的单个触发器，ALL表示启用所有触发器，USER表示仅启用用户触发器。

- **ENABLE REPLICA TRIGGER trigger_name**

  触发器的触发机制受GUC配置参数session_replication_role的影响。在会话处于“replica”模式时，仅会触发配置为ENABLE REPLICA的触发器；在会话处于“origin”（默认值）或“local”模式时，仅会触发没有配置ENABLE REPLICA的触发器。

- **ENABLE ALWAYS TRIGGER trigger_name**

  指定为ENABLE ALWAYS TRIGGER的触发器，将不受当前复制模式session_replication_role的设置始终都会触发。

- **{ DISABLE | ENABLE } [REPLICA | ALWAYS] RULE**

  关闭或启动指定规则。

  - DISABLE [REPLICA | ALWAYS] RULE：关闭指定规则。
  - ENABLE REPLICA：表示仅在会话处于replica模式时启动。
  - ENABLE ALWAYS：指定为ENABLE ALWAYS TRIGGER的触发器，将不受当前复制模式session_replication_role的设置始终都会触发。

  由于涉及到视图的实现，ON SELECT规则不可禁用。配置属于表的重写规则时，已禁用的规则在系统中是可见的，仅在查询重写期间被禁用。规则触发机制同时受配置变量session_replication_role的影响。

- **{ DISABLE | ENABLE } ROW LEVEL SECURITY**

  开启或关闭表的行访问控制开关，相关开关状态会同步更新到各执行节点的PG_CLASS.RELOPTIONS 中，以保持系统表元数据一致。

  当开启行访问控制开关时，如果未在该数据表中定义相关行访问控制策略，数据表的行级访问将不受影响；如果关闭表的行访问控制开关，即使定义了行访问控制策略，数据表的行访问也不受影响。详细信息请参见**CREATE ROW LEVEL SECURITY POLICY**。

- **{NO FORCE|FORCE} ROW LEVEL SECURITY**

  强制开启或关闭表的行访问控制开关，相关开关状态会同步更新到各执行节点的PG_CLASS.RELOPTIONS 中，以保持系统表元数据一致。

  默认情况，表所有者不受行访问控制特性影响，但当强制开启表的行访问控制开关时，表的所有者（不包含系统管理员用户）将受到行访问控制特性影响，系统管理员不受行访问控制策略影响。

版权所有 © 华为云计算技术有限公司

- **ENCRYPTION KEY ROTATION**

  透明数据加密密钥轮转。

  - 仅当数据库开启透明加密功能且表的enable_tde选项为on时，才可以进行表的数据加密密钥轮转操作。
  - 执行透明数据加密密钥轮转操作后，系统会自动向KMS申请创建新的密钥，使用旧密钥加密的数据仍使用旧密钥进行解密，新写入的数据需使用新密钥进行解密。为保证加密数据安全，用户可根据加密表的新增数据量大小定期更新密钥。

- **REPLICA IDENTITY { DEFAULT | USING INDEX index_name | FULL | NOTHING }**

  在逻辑复制场景下，指定该表的UPDATE和DELETE操作中旧元组的记录级别。

  - DEFAULT：记录主键的列的旧值，没有主键则不记录。
  - USING INDEX：记录index_name覆盖的列（指定为NOT NULL的列）的旧值，仅记录唯一、全局以及最终的值。
  - FULL：记录该行中所有列的旧值。
  - NOTHING：不记录有关旧行的信息。

  在逻辑复制场景，解析该表的UPDATE和DELETE操作语句时，记录的信息组成解析出的旧元组。对于有主键表，该选项可设置为DEFAULT或FULL。对于无主键表，该选项需设置为FULL，否则解码时旧元组将解析为空。一般场景不建议设置为NOTHING，旧元组会始终解析为空，建议用户表定义主键或唯一约束，以保证回放的确定性，否则回放SQL语句时可能会导致多行修改影响数据正确性。针对Ustore表，设置为NOTHING时不生效，实际效果与设置为FULL行为一致；对于无主键表，设置为DEFAULT，记录该行所有列。

- **AUTO_INCREMENT [=] value**

  设置自动增长列下一次的自增值。

  - 该语法仅在参数sql_compatibility='MYSQL'时生效。
  - 设置的值value仅在大于当前自增计数器时才会生效，且设置的值value必须是小于2^{127}-1的非负数。

- **[[DEFAULT] CHARACTER SET | CHARSET [=] default_charset] [[DEFAULT] COLLATE [=] default_collation]**

  将表的字符集和字符序修改为默认字符集和默认字符序。修改不会影响表中当前已经存在的列。

- **ILM ADD POLICY ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW AFTER n { day | days | month | months | year | years } OF NO MODIFICATION [ON (EXPR)]**

  在表上追加指定的ILM策略。一个完整的ILM策略由ILM动作和ILM条件构成。ILM动作表示用于定义具体的数据压缩或移动行为，ILM条件表示用于定义触发ILM动作的数据压缩或移动行为。ILM条件为行级条件，即ILM条件作用于表中的每一行。EXPR仅支持表字段及白名单中列出的函数具体请参见**行表达式函数白名单**。在闪回点与当前点之间添加ILM策略，不影响闪回功能。

- **[MODIFY { PARTITION partition_name | SUBPARTITION subpartition_name }] ILM { ENABLE | DISABLE | DELETE } POLICY policy_name**

  修改表（分区或子分区）的单个ILM策略。policy_name为系统视图GS_ADM_ILMOBJECTS或GS_MY_ILMOBJECTS查询得到的POLICY_NAME信息。

- **[MODIFY { PARTITION partition_name | SUBPARTITION subpartition_name }] ILM { ENABLE_ALL | DISABLE_ALL | DELETE_ALL}**

  修改表（分区或子分区）的所有ILM策略。

– 其中列操作column_clause为：

```
ADD [ COLUMN ] [ IF NOT EXISTS ] column_name data_type [ CHARACTER SET | CHARSET charset ] [ COLLATE collation ] [ column_constraint [ ... ] ]
| MODIFY column_name data_type
| MODIFY column_name [ CONSTRAINT constraint_name ] NOT NULL [ ENABLE ]
| MODIFY column_name [ CONSTRAINT constraint_name ] NULL
| MODIFY column_name data_type { { [ COMMENT 'string' ] [ ... ] }
| DROP [ COLUMN ] [ IF EXISTS ] column_name [ RESTRICT | CASCADE ]
```


版权所有 © 华为云计算技术有限公司

```
| ALTER [ COLUMN ] column_name [ SET DATA ] TYPE data_type [ ENCRYPTED WITH
( COLUMN_ENCRYPTION_KEY = column_encryption_key, ENCRYPTION_TYPE =
encryption_type_value ) ] [ COLLATE collation ] [ USING expression ]
| ALTER [ COLUMN ] column_name { SET DEFAULT expression | DROP DEFAULT }
| ALTER [ COLUMN ] column_name { SET | DROP } NOT NULL
| ALTER [ COLUMN ] column_name SET STATISTICS [PERCENT] integer
| ADD STATISTICS (( column_1_name, column_2_name [, ...] ))
| DELETE STATISTICS (( column_1_name, column_2_name [, ...] ))
| DISABLE STATISTICS (( column_1_name, column_2_name [, ...] ))
| ENABLE STATISTICS (( column_1_name, column_2_name [, ...] ))
| ALTER [ COLUMN ] column_name SET ( {attribute_option = value} [, ... ] )
| ALTER [ COLUMN ] column_name RESET ( attribute_option [, ... ] )
| ALTER [ COLUMN ] column_name SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }
```


版权所有 © 华为云计算技术有限公司

说明

- **ADD [COLUMN] [IF NOT EXISTS] column_name data_type [CHARACTER SET | CHARSET [=] charset] [COLLATE collation] [column_constraint [...]]**

  向表中增加一个新的字段。用ADD COLUMN增加一个字段，所有表中现有行都初始化为该字段的缺省值（如果没有声明DEFAULT子句，值为NULL）。如果指定IF NOT EXISTS关键字，如果存在相同名称的列，返回NOTICE提示，告知列已存在。未指定IF NOT EXISTS关键字时，如果存在相同名称的列，返回ERROR报错。

- **ADD ({ column_name data_type } [, ...])**

  向表中增加多列。

- **MODIFY ({ column_name data_type | column_name [CONSTRAINT constraint_name] NOT NULL [ENABLE] | column_name [CONSTRAINT constraint_name] NULL } [, ...])**

  修改表已存在字段的数据类型。此命令会导致该字段的统计信息清空，建议在修改后重新收集该列的统计信息。

- **MODIFY column_name data_type { { [COMMENT 'string'] [...] }**
  - 修改表格已存在字段的定义，用新的定义替换字段的原定义，原字段上的索引、独立对象约束（例如：主键、唯一键、CHECK约束等）不会被删除。
  - 此语法只能在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）使用。
  - 此语法不支持外表、修改加密字段。
  - 不支持修改分区键字段的数据类型和排序规则，不支持修改规则引用的字段的数据类型和排序规则，不支持修改物化视图引用的字段的数据类型和排序规则。
  - 被修改数据类型或排序规则的字段如果被一个生成列引用，那么这个生成列的数据将会重新生成。
  - 被修改字段若被一些对象依赖（如：索引、独立对象约束、视图、触发器、行级访问控制策略等），修改字段过程中将会重建这些对象。若被修改后字段定义违反此类对象的约束，修改操作会失败，如：修改作为视图结果列的字段的数据类型。请修改字段前评估这类影响。
  - HASH、LIST、RANGE分布场景下不支持修改分布列数据类型和其注释。
  - 被修改字段若被一些对象调用（如：自定义函数、存储过程等），修改字段不会处理这些对象。修改字段后，这些对象有可能出现不可用的情况，请修改字段前评估这类影响。
  - 此子句与上一子句中“MODIFY column_name data_type”部分语法相同，语义功能不同，当GUC参数b_format_behavior_compat_options含有‘enable_modify_column’选项时，将按照此子句功能处理。
  - 此命令会导致该字段的统计信息清空，建议在修改后重新收集该列的统计信息。

- **DROP [COLUMN] [IF EXISTS] column_name [RESTRICT | CASCADE]**

  从表中删除一个字段，和这个字段相关的索引和表约束也会被自动删除。如果任何表之外的对象依赖于这个字段，必须声明CASCADE ，比如视图等。

  DROP COLUMN命令并不是物理上把字段删除，而只是简单地把它标记为对SQL操作不可见。随后对该表的插入和更新将在该字段存储一个NULL。因此，删除一个字段是很快的，但是它不会立即释放表在磁盘上的空间，因为被删除了的字段占据的空间还没有回收。这些空间将在执行VACUUM时而得到回收。

- **ALTER [COLUMN] column_name [SET DATA] TYPE data_type [ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = column_encryption_key, ENCRYPTION_TYPE = encryption_type_value)] [COLLATE collation] [USING expression]**

  改变表字段的数据类型。该字段涉及的索引和简单的表约束将被自动地转换为使用新的字段类型，方法是重新分析最初提供的表达式。

版权所有 © 华为云计算技术有限公司

当字段的原始数据类型和修改后的数据类型二进制兼容时，执行该语句不需要对整表进行重写，其他场景下会进行整表重写。原类型和目标类型是否二进制兼容可以在PG_CAST系统表中查看，如果castmethod为‘b’则二进制兼容。例如源表中数据类型是text类型，如果转为int类型则会触发表重写，转为clob类型则不会触发表重写。如果表重写被触发，该表上被删除的空间也将被立刻回收。

此命令会导致该字段的统计信息清空，建议在修改后重新收集该列的统计信息。

不支持修改加密字段。

- **ALTER [COLUMN] column_name { SET DEFAULT expression | DROP DEFAULT }**

  为一个字段设置或者删除缺省值。请注意缺省值只应用于随后的INSERT命令，它们不会修改表中已经存在的行。也可以为视图创建缺省，这个时候它们是在视图的ON INSERT规则应用之前插入到INSERT句中的。

- **ALTER [COLUMN] column_name { SET | DROP } NOT NULL**

  修改一个字段是否允许NULL值或者拒绝NULL值。如果表在字段中包含非NULL，则只能使用SET NOT NULL。

- **ALTER [COLUMN] column_name SET STATISTICS [PERCENT] integer**

  为随后的ANALYZE操作设置针对每个字段的统计收集目标。目标的范围可以在0到10000之内设置。设置为-1时表示重新恢复到使用系统缺省的统计目标。

- **{ADD | DELETE} STATISTICS ((column_1_name, column_2_name [, ...]))**

  用于添加和删除多列统计信息声明（不实际进行多列统计信息收集），以便在后续进行全表或全库analyze时进行多列统计信息收集。每组多列统计信息最多支持32列。不支持添加/删除多列统计信息声明的表：系统表、外表。

- **{DISABLE | ENABLE} STATISTICS ((column_1_name, column_2_name [, ...]))**

  用于设置多列统计信息的状态。如果多列统计信息被禁用，则相关的统计信息不被收集，优化器在生成计划的阶段也不使用该多列统计信息。

- **ALTER [COLUMN] column_name SET ({attribute_option = value} [, ...])**

  ALTER [COLUMN] column_name RESET (attribute_option [, ...])

  设置/重置属性选项。

  目前，属性选项只定义了n_distinct和n_distinct_inherited。n_distinct影响表本身的统计值，而n_distinct_inherited影响表及其继承子表的统计。目前，只支持SET/RESET n_distinct参数，禁止SET/RESET n_distinct_inherited参数。

- **ALTER [COLUMN] column_name SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }**

  为一个字段设置存储模式。这个设置控制这个字段是内联保存还是保存在一个附属的表里，以及数据是否要压缩。SET STORAGE本身并不改变表上的任何东西，只是设置将来的表操作时，建议使用的策略。

- 其中列约束column_constraint为：

```
[ CONSTRAINT constraint_name ]
   { NOT NULL |
     NULL |
     CHECK ( expression ) |
     DEFAULT default_expr |
     ON UPDATE update_expr |
     AUTO_INCREMENT |
     COMMENT 'string' |
     COLVIEW |
     NOCOLVIEW |
     UNIQUE [KEY] index_parameters |
     PRIMARY KEY index_parameters |
     ENCRYPTED WITH ( COLUMN_ENCRYPTION_KEY = column_encryption_key,
ENCRYPTION_TYPE = encryption_type_value ) |
     REFERENCES reftable [ ( refcolumn ) ] [ MATCH FULL | MATCH PARTIAL | MATCH
SIMPLE ]
       [ ON DELETE action ] [ ON UPDATE action ] }
   [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```


版权所有 © 华为云计算技术有限公司

    - 其中索引参数index_parameters为：
      ```
      [ WITH ( {storage_parameter = value} [, ... ] ) ]
        [ USING INDEX TABLESPACE tablespace_name ]
      ```
    - 其中update_expr为：
      ```
      { CURRENT_TIMESTAMP | LOCALTIMESTAMP | NOW() }
      ```
  - 其中根据已有唯一索引为表增加主键约束或唯一约束table_constraint_using_index为：
    ```
    [ CONSTRAINT constraint_name ]
      { UNIQUE | PRIMARY KEY } USING INDEX index_name
      [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
    ```
  - 其中表约束table_constraint为：
    ```
    [ CONSTRAINT [ constraint_name ] ]
      { CHECK ( expression ) |
        UNIQUE [ idx_name ] [ USING method ] ( { { column_name [ ( length ) ] | ( expression ) } [ ASC | DESC ] } [, ... ] ) index_parameters |
        PRIMARY KEY [ USING method ] ( { column_name [ ASC | DESC ] } [, ... ] ) index_parameters |
        FOREIGN KEY [ idx_name ] ( column_name [, ... ] ) REFERENCES reftable [ ( refcolumn [, ... ] ) ] }
      [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
      { [ COMMENT 'string' ] [ ... ] }
    ```
    其中索引参数index_parameters为：
    ```
    [ WITH ( {storage_parameter = value} [, ... ] ) ]
      [ USING INDEX TABLESPACE tablespace_name ][BY GLOBAL INDEX]
    ```

    说明

    当index_parameters指定BY GLOBAL INDEX时，将使用全局二级索引建立约束。

- 重命名表。对名称的修改不会影响所存储的数据。
  ```
  ALTER TABLE [ IF EXISTS ] table_name
    RENAME [ TO | AS | = ] new_table_name;
  ```

  说明

  在MYSQL模式库中的5.7版本下指定（即sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s2'）,会出现以下特殊现象：

  - 新表名对应字符串的开头为"#mysql50#"，且其后还有其他字符，"#mysql50#"将被忽略。
  - 如果新旧表名一致，不会报错。
- 重命名表中指定的列。
  ```
  ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name ) }
    RENAME [ COLUMN ] column_name TO new_column_name;
  ```
- 重命名表的约束。
  ```
  ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name ) }
    RENAME CONSTRAINT constraint_name TO new_constraint_name;
  ```
- 设置表的所属模式。
  ```
  ALTER TABLE [ IF EXISTS ] table_name
    SET SCHEMA new_schema;
  ```


版权所有 © 华为云计算技术有限公司

说明

- 这种形式把表移动到另外一个模式。相关的索引、约束都跟着移动。目前序列不支持改变schema。 若该表拥有序列，需要将序列删除，重建，或者取消拥有关系， 才能将表schema更改成功。
- 要修改一个表的模式，用户必须在新模式上拥有CREATE权限。要把该表添加为一个父表的新子表，用户必须同时又是父表的所有者。要修改所有者，用户还必须是新的所有角色的直接或间接成员，并且该成员必须在该表的模式上有CREATE权限。这些限制规定了该用户不能做出了重建和删除表之外的事情。系统管理员可以以任何方式修改任意表的所有权限。
- 除了RENAME和SET SCHEMA之外所有动作都可以捆绑在一个经过多次修改的列表中并行使用。比如，可以在一个命令里增加几个字段或修改几个字段的类型。对于大表，此种操作带来的效率提升更明显，原因在于只需要对该大表做一次处理。
- 增加一个CHECK或NOT NULL约束将会扫描该表，以保证现有的行符合约束要求。
- 用一个非空缺省值增加一个字段或者改变一个字段的现有类型会重写整个表。对于大表来说，这个操作可能会花很长时间，并且它还临时需要两倍的磁盘空间。

- 更新多个列。

```
ALTER TABLE [ IF EXISTS ] table_name
    MODIFY ( { column_name data_type | column_name [ CONSTRAINT constraint_name ] NOT NULL [ ENABLE ] | column_name [ CONSTRAINT constraint_name ] NULL } [, ...] );
```

- 等待执行该SQL时刻，表上所有的DML事务提交。

```
ALTER TABLE [ IF EXISTS ] table_name GSIWAITALL IN lock_mode MODE;
```

说明

此语法主要用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，同步全局二级索引在线创建过程中的锁等待，按给出的锁级别，在各节点获取与该锁级别互斥的事务列表，并等待这些事务结束。用户不建议使用。如使用，会返回错误提示信息。

参数说明

- **ONLINE | OFFLINE**

  该关键字用于控制是否开启在线DDL。

  用ONLINE | OFFLINE关键字：

  - ONLINE：DDL以在线方式执行。此时，若ALTER TABLE语句中包含当前DDL特性不支持的场景时，DDL以离线方式执行，并通过NOTICE提示忽略ONLINE关键字。
  - OFFLINE：DDL以离线方式执行。

版权所有 © 华为云计算技术有限公司

说明

- 在线DDL修改列类型、精度、长度或在线添加非空、检查、主键、唯一约束时，会检查数据是否满足数据类型转换、长度修改或约束条件。若检查结果为不满足，DDL操作则会报错。在线DDL可并发DML，可能会出现并发DML将不满足转换或违反约束的数据删除，但DDL仍然报错的情况。

  例如：在线DDL扫描表修改列类型时，若表上元组A的数据并不满足数据转换，在线DDL的事务扫描到元组A时，会立即报错。如果在同一时间使用并行事务DELETE把元组A删除，但在线DDL扫描检查的事务早于并行事务DELETE，因此不会认为元组A已被删除。
- 在线DDL错误故障处理：包含在线DDL修改列类型、精度、长度或在线添加非空、检查、主键、唯一约束时，系统会自动创建临时模式作为DDL实现工具并将表设置为追加（append_mode）状态。若在线DDL正常执行完成，则临时模式会自动清理和恢复。自动清理过程可能由于等锁超时等原因而失败，无法进行自动清理，需要手动清理恢复。若发生严重故障，如FATAL、PANIC、数据库故障和节点断连等故障时，创建的临时模式和表的追加状态可能残留，需要手动清理和恢复。包含如下情况：
  - 取消执行在线DDL：执行过程中取消在线DDL，系统会自动清理。用户取消执行后首先进入残留清理流程，此时如果用户再次取消，则会结束清理流程导致残留。
  - 操作报错：当DDL命令执行过程中出现需要报错场景，如类型转换无法执行或长度修改损伤数据等，系统会报错并自动清理。
  - 严重故障：
    1. 分布式一主无备部署节点宕机：分布式一主无备节点，CN或DN节点宕机，可通过**online_ddl_cleanup()**进行手动清理，或在下次执行在线DDL或在线VACUUM FULL时自动清理。
    2. 分布式一主多备部署节点宕机：
       - 执行DDL的CN节点断连宕机：可通过**online_ddl_cleanup()**进行手动清理，或在下次执行在线DDL或在线VACUUM FULL时自动清理。
       - 其他节点断连宕机等故障：分布式一主多备节点且具有HA主备切换时，非执行DDL的CN节点故障，等待集群完成主备切换集群恢复正常（在线DDL会等待60s），最多重试次数为3次，若三次内集群恢复正常则无需手动清理，若集群恢复失败则需用户待集群正常后通过**online_ddl_cleanup()**进行手动清理。根据集群异常场景可能存在重试无法进行的场景，需手动清理。
  - 对于在线添加主键和唯一约束：若在线创建索引期间发生异常情况（比如用户手动取消、唯一索引键值重复、资源不足、启动线程失败、锁超时等失败场景）导致在线创建索引失败。这种情况下可能会残留索引以及临时表，占用系统资源，具体请参见**CREATE INDEX**章节中**CONCURRENTLY**对异常情况处理。
- 不建议在线DDL执行中**online_ddl_cleanup()**清理操作，可能会导致中止在线DDL或产生未知错误。
- 在线修改列类型char类型扩展时，不会修改重写底层数据。在线DDL操作后，已有的数据底层存储仍保有旧的长度，DML带来的新增修改数据的底层存储则会服从扩展长度。

- **online_parameter**

  当指定ONLINE关键字时，用于指定在线DDL选项。若未指定ONLINE关键字或不满足在线DDL的条件，该参数不生效。

  取值范围：

  – parallel_threads：表示开启并行时的线程数。1-32之间的整数值，缺省时默认值为1。

    ▪ 当parallel_threads设置为大于1时，开启多线程，采用线程数为指定线程数。

版权所有 © 华为云计算技术有限公司

  - 当不设置parallel_threads或设置1，关闭多线程加速。

    当用户开启资源负载管理时，比如启用IO资源管理 (如io_limits>0或io_priority = high/medium/low)，parallel_threads将忽略用户/会话下的资源负载管理，并给出提示当前资源负载管理无效并开启多线程加速。在线DDL多线程执行过程实际遵循初始用户的资源负载管理。

  - 其中关于索引并行parallel_workers参数共同作用加速。parallel_workers参数控制（最大）索引内并行线程数，parallel_threads参数控制（最大）索引间并行数。
- max_catchup_times：表示在线DDL的最大追增轮次，防止在线DDL执行时间过长。1 - INT_MAX之间的整数值，缺省时默认值为50。

  - 在线DDL会记录数据重建过程中的DML增量数据到临时表，并采用多轮次循环追增，将增量数据进行重建。为防止由于DML并发数据量过大，在线DDL无法完成追增的情况，用户可以设置该参数，以限制追增次数。
- last_catchup_threshold：触发最后一轮追增的时间上限。当一轮追增完成后，若该轮耗时低于此阈值，则触发最后一轮追增及切换。取值范围：[1000,1200000]（单位：毫秒），默认值：5000。
- online_ddl_lockwait_timeout：在线DDL的等锁超时时间。若等锁线程在该时间内未能获取锁，则报锁超时失败。触发CANCEL模式获取锁时，实际使用的超时时间为该参数值的2倍。取值范围：[1,1200000]（单位：毫秒），默认值：10000。
- lockwait_interval：等锁步长，即连续等锁超时的重试间隔。若当前轮次等锁超时，等待该参数指定的时长后进入下一轮尝试。取值范围：[0,300]（单位：秒），默认值：5。
- trylock_threshold：等锁尝试次数上限，需与enable_cancel配合使用。达到该上限后若仍因等锁超时而无法获取锁，则下一次尝试将采用CANCEL模式。若未开启enable_cancel，达到上限后取锁失败并报错。取值范围：[1,INT_MAX]，或 -1（表示无上限），默认值：-1。
- enable_cancel：是否允许用户业务快速失败，即在达到等锁尝试次数上限，或在切表事务获取用户表8级锁时，通过快速中止用户业务来确保在线DDL成功获取锁。取值范围：[false,true]，默认值：false。

- **IF EXISTS**

  如果不存在相同名称的表，不会抛出一个错误，而会发出一个通知，告知表不存在。

- **table_name [*] | ONLY table_name | ONLY (table_name)**

  table_name是需要修改的表名。

  若声明了ONLY选项，则只有该表被更改。若未声明ONLY，该表及其所有子表都将会被更改。另外，可以在表名称后面显式地增加*选项来指定包括子表，即表示所有后代表都被扫描，这是默认行为。

- **constraint_name**
  - 在DROP CONSTRAINT操作中表示要删除的现有约束的名称。
  - 在ADD CONSTRAINT操作中表示新增的约束名称。

版权所有 © 华为云计算技术有限公司

须知

对于新增约束，在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）constraint_name为可选项，在其他模式数据库下，必须加上constraint_name。

- **index_name**

  索引名称。

  须知

  在ADD CONSTRAINT操作中：

  - index_name仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
  - 对于外键约束，constraint_name和index_name同时指定时，索引名为constraint_name。
  - 对于唯一键约束，constraint_name和index_name同时指定时，索引名为index_name。

- **USING method**

  指定创建索引的方法。

  取值范围请参见**CREATE INDEX**章节中的**USING method**。

  须知

  在ADD CONSTRAINT操作中：

  - USING method仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
  - 在MYSQL模式下，未指定USING method时，对于ASTORE的存储方式，默认索引方法为btree；对于USTORE的存储方式，默认索引方法为ubtree。
  - 当表的存储方式为USTORE时，SQL语句中约束指定为using btree，底层会自动将约束建立为using ubtree。

- **ASC | DESC**

  ASC表示指定按升序排序（默认）。DESC指定按降序排序。

  须知

  在ADD CONSTRAINT中，ASC | DESC只在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库不支持。

- **expression**

  创建一个基于该表的一个或多个字段的表达式索引约束，必须写在圆括号中。

版权所有 © 华为云计算技术有限公司

须知

UNIQUE约束中的表达式索引只在MYSQL模式数据库下支持（即sql_compatibility = 'MYSQL'），其他模式数据库不支持。

- **storage_parameter**

  表的存储参数的名称。

  在线扩容新增的选项：

  - append_mode(枚举类型)

    设置表上扩容方式为在线扩容，离线扩容，非扩容方式。在线扩容时允许对表进行部分的修改操作；离线扩容时，在扩容时不允许对表进行操作。

    正在扩容的表上需要新增数据，要以追加的方式写入，便于记录增量数据。

    - on：标记表为在线扩容模式，在线扩容时，设置后后续数据以追加方式写入。
    - off：关闭扩容模式，设置后表上数据按正常方式写入，并且在pg_class.reloptions中不显示在线扩容相关的选项。
    - read_only：标记表为离线扩容。离线扩容时，不允许对表进行操作。
    - end_catchup：最后一轮追增的写报错模式，写业务报错，读业务正常执行。

  - rel_cn_oid（OID类型）

    记录当前CN节点中表的OID，用于在DN节点上生成delete_delta表。

    当append_mode=on时，必须同时指定rel_cn_oid。

    这append_mode，rel_cn_oid两个选项只在在线扩容工具中使用，不建议用户使用。

  - exec_step（整型）

    记录断点续传的步骤，记录在临时表的relOptions中。

    取值范围：[1,4]

    只支持数据重分布工具使用。

  - create_time（长整型）

    记录断点续传时临时表创建时间，记录在临时表的relOptions中。

    只支持数据重分布工具使用。

  - redistribute_version(长整型)

    标记逻辑扩容重分布某张表的重分布成功次数，提供给备机读校验一致性使用，每次该表重分布完成后值加1。

    此选项只在扩容流程设置，不建议用户使用。

  - enable_update_distkey

    标记该表是否支持更新（UPDATE）分布列的操作。仅当表的该属性设置为on，且在约束条件下时，才支持更新（UPDATE）分布列的操作。否则，分布列不支持更新（UPDATE）操作。

    取值范围：on/off

    默认值：off

版权所有 © 华为云计算技术有限公司

说明

支持更新（UPDATE）分布列操作的约束如下：

- 仅当相应表属性enable_update_distkey设置为on时才支持更新分布列。
- 不支持将UPDATE语句下推DN执行，直接生成PGXC计划，不会根据更新分布列前后值而对计划做改变。
- 不支持带有行级UPDATE TRIGGER的表，否则会执行失败，报错进行提示。对行级INSERT/DELETE TRIGGER不生效，update statement级TRIGGER正常执行。
- 不支持并发更新同一行，先获取锁的执行，DN上后获取锁的按照GUC参数concurrent_dml_mode设置情况进行不同的行为（返回0或报错）。如果报错，则可能存在两种情况：（1）报错提示信息为update distribute column conflict。（2）当获取锁时间超过阈值时，报错提示信息为锁超时。
- 不支持带有全局二级索引（GSI）的表，否则会执行报错。
- 只支持HASH分布，不支持LIST/RANGE分布表，否则会执行报错。
- 不支持MERGE INTO和UPSERT更新分布列的行为，否则会执行报错。
- 不支持gtm_free，否则会执行报错。
- 不支持UPDATE RETURNING，否则会执行报错。
- 不支持带有关联表的语句，否则会执行报错。
- 不支持UPDATE + LIMIT，否则会执行报错。

创建索引新增一个选项：

- parallel_workers（int类型）

  表示创建索引时起的bgworker线程数量，例如2就表示将会起2个bgworker线程并发创建索引。

  取值范围：[0,32]，0表示关闭并行建索引。

  默认值：不设置该参数，表示未开启并行建索引功能。

复制表新增选项：

- primarynode（bool类型）

  默认值：off

  当primarynode=on时，将为复制表选择primary node，通常是pgxc_class表nodeoids字段记录的第一个节点。当复制表执行IUD操作时，将先下发到primarynode节点执行，收到结果后再下发到其它DN。

- logical_repl_node（字符串类型）

  分布式复制表逻辑解码时，向CN返回逻辑日志的DN节点名。对于复制表，如用户不指定，则默认为当前表所在node group的第一个节点。对该选项进行RESET操作时，会重置为当前表的第一个节点。

  取值范围：字符串。

  默认值：非复制表默认为空，复制表默认为第一个节点名。

透明数据加密选项：

- enable_tde（bool类型）

  是否将表设置为加密表。本参数仅支持行存表、段页式表、hashbucket表、临时表和unlogged表。设置enable_tde=on时，请确保已通过GUC参数enable_tde开启透明加密功能，并通过GUC参数tde_key_info设置访问密钥服务的信息。在《开发指南》中“特性使用指导 > 透明数据加密”章节可获取该参数的详细使用方法。

  取值范围：on/off。

  on：表示开启透明数据加密。

版权所有 © 华为云计算技术有限公司

off：表示关闭透明数据加密。

说明

- 从on切换为off后，插入或更新数据在写入旧页面时仍加密，写入切换后新生成的页面时会不加密。
- 从off切换为on后，插入或更新数据在写入旧页面时不加密，写入切换后新生成的页面时会自动加密。

即：加密开关切换后，生成的新数据页面加密状态与开关切换后状态保持一致，旧数据页面加密状态与开关切换前状态保持一致。建议用户在切换加密开关后，手动对表执行VACUUM FULL操作，使所有数据页面加密状态保持一致。

默认值：off

- encrypt_algo（string类型）

  指定加密表的加密算法。

  取值范围：字符串，有效值为：AES_128_CTR，SM4_CTR、AES_128_CTR_SHA256_HMAC、SM4_CTR_SM3_HMAC或AES_128_GCM。

  默认值：如果enable_tde=on，GUC参数tde_encrypt_config中子参数table_algorithm不为空，默认值与table_algorithm相同；如果table_algorithm为空，默认值为AES_128_CTR。如果enable_tde=off，默认值为空。

  说明

  - 类似于enable_tde，切换encrypt_algo后，插入或更新数据在写入旧页面时仍使用旧加密算法，写入切换后新生成的页面时则使用新加密算法。
  - 同样可通过手动对表执行VACUUM FULL操作，使所有数据页面均使用新算法加密。

- hasuids（bool类型）

  默认值：off

  参数开启：更新表元组时，为元组分配表级唯一标识id。

- statistic_granularity

  记录该表在分析统计信息时的默认partition_mode。partition_mode说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。此参数对非分区表设置无效。

  取值范围：请参见partition_mode取值范围。

  默认值：AUTO

ROWID特性新增选项：

- hasrowid

  指定表是否包含ROWID、ROWNO系统列。设置hasrowid = on时，该表会增加rowid系统列和rowno系统列，且会在rowno系统列上关联创建序列和索引。

  取值范围：on、off

  默认值：off

版权所有 © 华为云计算技术有限公司

说明

- 此参数只支持ORA兼容模式数据库（即sql_compatibility = 'ORA'）。
- 此参数不支持本地临时表、Unlogged表、全局临时表、物化视图、hashbucket表、复制表。
- 此参数不支持和hasuids参数同时指定。
- 开启此参数时，不支持定义普通列名为rowno、rowid。
- 不支持通过SET ({storage_parameter = value} [, ...])或RESET (storage_parameter [, ...])子句变更表的hasrowid存储选项为不同值。

在线DDL新增的选项：

- online_ddl_status（枚举类型）

  本参数用于更新目标表在线DDL执行状态。

  仅支持在线DDL内部调用，不建议用户使用。

  - on：开启在线DDL状态。
  - off：关闭在线DDL状态。
  - init：进行在线DDL临时对象初始化。
  - catchup：进行DDL操作的并发新增数据追增。
  - catchup_basedata: 进行在线ddl操作的基础数据同步。
  - scan_catchup：进行在线添加非空/检查约束操作的数据追增。
  - scan_catchup_basedata：进行在线添加非空/检查约束操作的基础数据检查。
  - vacuum_catchup：进行在线VACUUM FULL操作的数据追增。
  - vacuum_verbose_catchup：进行在线VACUUM FULL VERBOSE操作的数据追增。
  - last_catchup：进入在线DDL最后一轮追增的写报错模式。
  - switch：进行在线DDL最后一轮追增的临时表交换。
  - copy_switch：进行在线移动表空间操作最后一轮追增的临时表交换。
  - record_toast_oid：记录在线DDL操作目标表附带的TOAST表的OID。
  - abort：在线DDL进入abort状态。
  - flush_buffer：用于在线移动表空间操作中强制buffer落盘。
  - tmp_rel：用于标识该表为在线DDL过程创建的临时表。

- online_ddl_lock_tag（int类型）

  用于记录在线DDL目标表的唯一标识（“表所在的Schema+表名”的hash值），用于生成临时对象。

  仅支持在线DDL内部调用，不建议用户使用。

VACUUM新增的选项：

版权所有 © 华为云计算技术有限公司

  - vacuum_truncate
    表示是否开启表的VACUUM TRUNCATE功能。
    取值范围：on/off
    默认值：on

- **new_owner**
  表新拥有者的名称。
- **new_tablespace**
  表所属新的表空间名称。
- **column_name，column_1_name， column_2_name**
  现存的或新字段的名称。
- **data_type**
  新字段的类型，或者现存字段的新类型。
- **charset**
  指定表字段的字符集。单独指定时会将字段的字符序设置为指定的字符集的默认字符序。
  仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持该语法，其他模式数据库不支持。
- **collation**
  字段排序规则（字符序）名称。可选字段COLLATE指定了新字段的排序规则，如果省略，排序规则为新字段的默认类型。排序规则可以使用“select * from pg_collation”命令从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。
  对于MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）还支持utf8mb4_bin、utf8mb4_general_ci、utf8mb4_unicode_ci、binary、gbk_chinese_ci、gbk_bin、gb18030_chinese_ci、gb18030_bin字符序，部分说明请参见**表1-398**。

  说明

  - 仅字符类型支持指定字符集。指定为binary字符集或字符序实际是将字符类型转化为对应的二进制类型，若类型映射不存在则报错。当前仅有TEXT类型转化为BLOB的映射。
  - 除binary字符集和字符序外，当前仅支持指定与数据库编码相同的字符集。
  - 字段字符集或字符序未显式指定时，若指定了表的默认字符集或字符序，字段字符集和字符序将从表上继承。若表的默认字符集或字符序不存在，当b_format_behavior_compat_options包含‘default_collation’时，字段的字符集和字符序将继承当前数据库的字符集及其对应的默认字符序。
  - 当修改的字符集或字符序对应的字符集与当前字段字符集不同时，会将字段中的数据转换为指定的字符集进行编码。
- **USING expression**
  USING子句声明如何从旧的字段值里计算新的字段值；如果省略，缺省从旧类型向新类型的赋值转换。如果从旧数据类型到新类型没有隐含或者赋值的转换，则必须提供一个USING子句。

版权所有 © 华为云计算技术有限公司

说明

ALTER TYPE的USING选项实际上可以声明涉及该行旧值的任何表达式，即它可以引用除了正在被转换的字段之外其他的字段。这样，就可以用ALTER TYPE语法做非常普遍性的转换。因为这个灵活性，USING表达式并没有作用于该字段的缺省值（如果有的话），结果可能不是缺省表达式要求的常量表达式。这就意味着如果从旧类型到新类型没有隐含或者赋值转换的话，即使存在USING子句，ALTER TYPE也可能无法把缺省值转换成新的类型。在这种情况下，应该用DROP DEFAULT先删除缺省，执行ALTER TYPE，然后使用SET DEFAULT增加一个合适的新缺省值。类似的考虑也适用于涉及该字段的索引和约束。

- **NOT NULL | NULL**

  设置列是否允许空值。

- **ENABLE**

  表示启动该约束，缺省时默认启用。

- **integer**

  带符号的整数常值。当使用PERCENT时表示按照表数据的百分比收集统计信息，integer的取值范围为0-100。

- **attribute_option**

  属性选项。

- **PLAIN | EXTERNAL | EXTENDED | MAIN**

  字段存储模式。

  - PLAIN必须用于定长的数值（比如integer）并且是内联的、不压缩的。
  - MAIN用于内联、可压缩的数据。
  - EXTERNAL用于外部保存、不压缩的数据。使用EXTERNAL将令在text和bytea字段上的子字符串操作更快，但付出的代价是增加了存储空间。
  - EXTENDED用于外部的压缩数据，EXTENDED是大多数支持非PLAIN存储的数据的缺省。

- **CHECK (expression)**

  每次将要插入的新行或者将要被更新的行必须使表达式结果为真才能成功，否则会抛出一个异常并且不会修改数据库。

  声明为字段约束的检查约束应该只引用该字段的数值，而在表约束里出现的表达式可以引用多个字段。

  目前，CHECK表达式不能包含子查询也不能引用除当前行字段之外的变量。

- **DEFAULT default_expr**

  给字段指定缺省值。

  缺省表达式的数据类型必须和字段类型匹配。

  缺省表达式将被用于任何未声明该字段数值的插入操作。如果没有指定缺省值则缺省值为NULL 。

- **ON UPDATE update_expr**

  ON UPDATE子句为字段的一种属性约束。

  当对表中某元组执行UPDATE操作时，若更新字段的新值和表中旧值不相同，则表中该元组上具有该属性且不在更新字段内的字段值自动更新为当前时间戳；若更新字段的新值和表中旧值相同，则表中该元组上具有该属性且不在更新字段内的字段值不变，保持原有值；若具有该属性的字段在更新字段内，则对应这些字段值直接按指定更新的值更新。

版权所有 © 华为云计算技术有限公司

说明

- 该属性支持在MYSQL模式数据库中的5.7版本下指定（即sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s1'）和PG模式数据库模式中指定（升级观察期间不支持该语法）。
- 语法上update_expr支持CURRENT_TIMESTAMP、LOCALTIMESTAMP、NOW()三种关键字，也支持关键字带括号指定或不指定精度。三种关键字互为同义词，属性效果相同。例如：ON UPDATE CURRENT_TIMESTAMP()、ON UPDATE CURRENT_TIMESTAMP(5)、ON UPDATE LOCALTIMESTAMP()、ON UPDATE LOCALTIMESTAMP(6)等。不带括号或空括号时精度为0，其中NOW关键字不支持不带括号。PG兼容模式下不支持ON UPDATE CURRENT_TIMESTAMP()、ON UPDATE LOCALTIMESTAMP()和ON UPDATE NOW(5)。
- 该属性仅支持在如下类型的列上指定：timestamp、datetime、date、time without time zone、smalldatetime、abstime。
- CREATE TABLE AS语法不会继承该列属性。
- CREATE TABLE LIKE语法可通过INCLUDING UPDATE或EXCLUDING UPDATE来选择继承或排除该约束，PG兼容模式下只支持EXCLUDING UPDATE选项排除该约束。新增INCLUDING ILM选项复制旧表的ilm策略信息，结合INCLUDING PARTITION选项使用可以复制旧表上分区对象的策略信息。
- 该属性指定的精度和对应列上类型指定的精度可以不一致，通过该属性更新字段值后显示结果按最小精度显示。例如：ALTER TABLE t1 ADD col1 timestamp(6) ON UPDATE CURRENT_TIMESTAMP(3)；若UPDATE语法触发该属性生效，则本次更新后col1字段值小数位显示3位。
- 该属性和生成列约束不能同时指定同一列。
- 分区表中的分区键不支持指定该属性。
- 分布式场景中，分布列和主键不支持指定该属性。

- **AUTO_INCREMENT**

  指定列为自动增长列。

  具体请参见**CREATE TABLE**章节中的**AUTO_INCREMENT**。

- **COLUMN_ENCRYPTION_KEY = column_encryption_key**

  为ENCRYPTED WITH约束中列加密密钥的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **ENCRYPTION_TYPE = encryption_type_value**

  为ENCRYPTED WITH约束中的加密类型，encryption_type_value的值为[DETERMINISTIC | RANDOMIZED]。

- **COMMENT [=] 'string'**
  - COMMENT [=] 'string'子句表示给表添加注释。
  - 在column_constraint中的COMMENT 'string' 表示给列添加注释。
  - 在table_constraint中的COMMENT 'string' 表示给主键和唯一键对应的索引添加注释。

  具体请参见**CREATE TABLE**章节中的**COMMENT [=] 'string'**。

- **UNIQUE [KEY] index_parameters**

  UNIQUE约束表示表里的一个或多个字段的组合必须在全表范围内唯一。

  对于唯一约束，NULL被认为是互不相等的。

  UNIQUE KEY只能在在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持使用，与UNIQUE语义相同。

- **UNIQUE [index_name][USING method] ({ {column_name [(length)] | (expression) } [ASC | DESC] }[, ...]) index_parameters**

版权所有 © 华为云计算技术有限公司

UNIQUE约束表示表里的一个或多个字段的组合必须在全表范围内唯一。

column_name (length)是前缀键，具体请参见**CREATE INDEX**章节中的**column_name (length)**。

对于唯一约束，NULL被认为是互不相等的。

须知

- index_name仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
- 对于唯一键约束，constraint_name和index_name同时指定时，索引名为index_name。

- **PRIMARY KEY index_parameters**

  PRIMARY KEY [USING method] ({ column_name [ASC | DESC] } [, ...] ） index_parameters

  主键约束表明表中的一个或者一些字段只能包含唯一（不重复）的非NULL值。
- **USING method**

  指定创建索引的方法。

  取值范围请参见**CREATE INDEX**中的**参数说明**。

须知

- USING method仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
- 在MYSQL模式数据库下，未指定USING method时，对于ASTORE的存储方式，默认索引方法为btree；对于USTORE的存储方式，默认索引方法为ubtree。
- 对于默认建立全局二级索引的约束，由于全局二级索引底层使用ubtree存储，即使用户指定存储方式为btree，底层也会建立为ubtree。
- 当表的存储方式为USTORE时，SQL语句中约束指定为using btree，底层会自动将约束建立为using ubtree。

- **ASC | DESC**

  ASC表示指定按升序排序（默认）。DESC指定按降序排序。

须知

ASC | DESC只在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库不支持。

- **expression**

  创建一个基于该表的一个或多个字段的表达式索引约束，必须写在圆括号中。

版权所有 © 华为云计算技术有限公司

须知

UNIQUE约束中的表达式索引只在MYSQL模式数据库下支持（即sql_compatibility = 'MYSQL'），其他模式数据库不支持。

- **DEFERRABLE | NOT DEFERRABLE**

  这两个关键字用于设置约束是否可推迟。不可推迟的约束将在每条命令执行后立即检查。可推迟约束可以延迟到事务结尾通过SET CONSTRAINTS命令检查，缺省值为NOT DEFERRABLE。目前UNIQUE约束、主键约束、外键约束可以接受该子句，其他约束类型均不可推迟。

- **INITIALLY IMMEDIATE | INITIALLY DEFERRED**

  若约束可推迟，该子句声明检查约束的缺省时间。不可推迟的约束始终为INITIALLY IMMEDIATE且不可声明为INITIALLY DEFERRED。

  - 如果约束是INITIALLY IMMEDIATE（缺省），则在每条语句执行后立即检查。
  - 如果约束是INITIALLY DEFERRED ，则仅在事务结束时进行检查。

- **WITH ({storage_parameter = value} [, ...])**

  为表或索引指定一个可选的存储参数。

- **tablespace_name**

  索引所在表空间的名称。

- **NOCOMPRESS**

  如果指定关键字NOCOMPRESS则不会修改表的现有压缩特性。

- **new_table_name**

  修改后新的表名称。

- **new_column_name**

  表中指定列修改后新的列名称。

- **new_constraint_name**

  修改后表约束的新名称。

- **new_schema**

  修改后新的模式名称。

- **CASCADE**

  级联删除依赖于被依赖字段或者约束的对象（比如引用该字段的视图）。

- **RESTRICT**

  如果该列被其他字段或者约束引用，则拒绝删除该列。RESTRICT为CASCADE的缺省选项，如果未指定则为RESTRICT。语句示例如下：

  ```
  ALTER TABLE table_name [DROP [column] col_name [CASCADE | RESTRICT]];
  ```
  其中：table_name表示表名，col_name表示列名。

- **schema_name**

  表所在的模式名称。

- **IF NOT EXISTS**

  如果指定IF NOT EXISTS，如果存在相同名称的列，返回NOTICE提示，告知列已存在。未指定IF NOT EXISTS时，如果存在相同名称的列，返回ERROR报错。

- **[DEFAULT] CHARACTER SET | CHARSET [=] default_charset**

版权所有 © 华为云计算技术有限公司

修改表的默认字符集。单独指定时会将表的默认字符序设置为指定的字符集的默认字符序。

- **[DEFAULT] COLLATE [=] default_collation**

  修改表的默认字符序。单独指定时会将表的默认字符集设置为指定的字符序对应的字符集。

  仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持该语法。字符序请参见**表1-398**。

  说明

  表的字符集或字符序未显式指定时，若指定了模式的默认字符集或字符序，表字符集和字符序将从模式上继承。若模式的默认字符集或字符序不存在，当b_format_behavior_compat_options包含‘default_collation’时，表的字符集和字符序将继承当前数据库的字符集及其对应的默认字符序。

- **GSIWAITALL**

  用于CREATE GLOBAL INDEX CONCURRENTLY/REINDEX INDEX CONCURRENTLY功能内部调用，同步全局二级索引在线创建过程中的锁等待，按给出的锁级别，在各节点获取与该锁级别互斥的事务列表，并等待这些事务结束。

修改表示例

- **重命名表。**

```
gaussdb=# CREATE TABLE aa(c1 int, c2 int);
gaussdb=# ALTER TABLE IF EXISTS aa RENAME TO test_alt1;
```

- **修改表所属模式。**

```
--创建模式test_schema。
gaussdb=# CREATE SCHEMA test_schema;

--把表test_alt1的所属模式修改为test_schema。
gaussdb=# ALTER TABLE test_alt1 SET SCHEMA test_schema;

--查询表信息。
gaussdb=# SELECT schemaname,tablename FROM pg_tables WHERE tablename = 'test_alt1';
 schemaname  | tablename
-------------+-----------
 test_schema | test_alt1
(1 row)
```

- **修改表的所有者。**

```
--创建用户test_user。
gaussdb=# CREATE USER test_user PASSWORD '********';

--修改test_alt1表的所有者为test_user。
gaussdb=# ALTER TABLE IF EXISTS test_schema.test_alt1 OWNER TO test_user;

--查看。
gaussdb=# SELECT tablename, schemaname, tableowner FROM pg_tables WHERE tablename =
'test_alt1';
 tablename | schemaname  | tableowner
-----------+-------------+------------
 test_alt1 | test_schema | test_user
(1 row)
```

- **修改表的表空间。**

```
--创建表空间tbs_data1。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace1/tbs_data1';

--修改test_alt1表的空间为tbs_data1。
gaussdb=# ALTER TABLE test_schema.test_alt1 SET TABLESPACE tbs_data1;

--查看。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT tablename, tablespace FROM pg_tables WHERE tablename = 'test_alt1';
 tablename | tablespace
-----------+------------
 test_alt1 | tbs_data1
(1 row)

--创建表
gaussdb=# CREATE TABLE test(c1 int, c2 int);
--在线修改test表的空间为tbs_data1。
gaussdb=# ALTER TABLE ONLINE test SET TABLESPACE tbs_data1;

--查看。
gaussdb=# SELECT tablename, tablespace FROM pg_tables WHERE tablename = 'test';
 tablename | tablespace
-----------+------------
 test| tbs_data1
(1 row)

--删除。
gaussdb=# DROP TABLE test ;
gaussdb=# DROP TABLE test_schema.test_alt1;
gaussdb=# DROP TABLESPACE tbs_data1;
gaussdb=# DROP SCHEMA test_schema;
gaussdb=# DROP USER test_user;
```

修改列示例

- **修改列名。**

```
--建表。
gaussdb=# CREATE TABLE test_alt2(c1 INT,c2 INT);

--修改列名。
gaussdb=# ALTER TABLE test_alt2 RENAME c1 TO id;
gaussdb=# ALTER TABLE test_alt2 RENAME COLUMN c2 to areaid;

--查看。
gaussdb=# \d test_alt2
   Table "public.test_alt2"
 Column |  Type   | Modifiers
--------+---------+-----------
 id     | integer |
 areaid | integer |
```

- **增加列。**

```
--表test_alt2增加列。
gaussdb=# ALTER TABLE IF EXISTS test_alt2 ADD COLUMN name VARCHAR(20);

--查看。
gaussdb=# \d test_alt2
           Table "public.test_alt2"
  Column  |         Type          | Modifiers
----------+-----------------------+-----------
 id       | integer               |
 areaid   | integer               |
 name     | character varying(20) |
```

- **增加自增列。**

```
--表local_autoinc增加AUTO_INCREMENT自增列（根据实际情况修改DATANODE名字，SELECT
node_name FROM pgxc_node WHERE node_type = 'D';）。
gaussdb=# CREATE TABLE local_autoinc(col1 int)
DISTRIBUTE BY LIST(col1)(
    SLICE s1 VALUES (1) DATANODE datanode1,
    SLICE s2 VALUES (2) DATANODE datanode2
);
--数据分布到DN1上。
gaussdb=# INSERT INTO local_autoinc(col1) VALUES(1);

--添加一个本地自增列，每个DN从1开始自增。
gaussdb=# ALTER TABLE local_autoinc ADD COLUMN col int AUTO_INCREMENT;
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT col,col1 FROM local_autoinc ORDER BY 2,1;
 col | col1
-----+------
   1 |    1
(1 row)

--将所有DN的下一个自增值设为10。
gaussdb=# ALTER TABLE local_autoinc AUTO_INCREMENT = 10;

--数据分布到DN1上，NULL触发自增，自增值为10。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(NULL,1);

--数据分布到DN2上，0触发自增，自增值为10。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(0,2);
gaussdb=# SELECT col,col1 FROM local_autoinc ORDER BY 2,1;
 col | col1
-----+------
   1 |    1
  10 |    1
  10 |    2
(3 rows)
```

- **修改列的数据类型**。

```
--修改test_alt2表中name字段的类型。
gaussdb=# ALTER TABLE test_alt2 MODIFY name VARCHAR(50);


--查看。
gaussdb=# \d test_alt2
          Table "public.test_alt2"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 id     | integer               |
 areaid | integer               |
 name   | character varying(50) |

--在线修改test_alt2表中name字段的类型。
gaussdb=# ALTER TABLE ONLINE test_alt2 MODIFY name VARCHAR(60);


--查看。
gaussdb=# \d test_alt2
          Table "public.test_alt2"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 id     | integer               |
 areaid | integer               |
 name   | character varying(60) |
--修改test_alt2表中name字段的类型。
gaussdb=# ALTER TABLE test_alt2 ALTER COLUMN name TYPE VARCHAR(25);

--查看。
gaussdb=# \d test_alt2
          Table "public.test_alt2"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 id     | integer               |
 areaid | integer               |
 name   | character varying(25) |

--在线修改test_alt2表中name字段的类型。
gaussdb=# ALTER TABLE ONLINE test_alt2 ALTER COLUMN name TYPE VARCHAR(35);

--查看。
gaussdb=# \d test_alt2
          Table "public.test_alt2"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 id     | integer               |
```


版权所有 © 华为云计算技术有限公司

```
 areaid | integer               |
 name   | character varying(35) |
```

- **删除列**。

```
--删除test_alt2中areaid字段。
gaussdb=# ALTER TABLE test_alt2 DROP COLUMN areaid;

--查看。
gaussdb=# \d test_alt2
          Table "public.test_alt2"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 id     | integer                |
 name   | character varying(25) |
```

- **修改列的存储模式**。

```
--查看表详细信息。
gaussdb=# \d+ test_alt2
                          Table "public.test_alt2"
 Column |          Type          | Modifiers | Storage  | Stats target | Description
--------+------------------------+-----------+----------+--------------+-------------
 id     | integer                |           | plain    |              |
 name   | character varying(35) |           | extended |              |
Has OIDs: no
Distribute By: HASH(id)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no, storage_type=USTORE, segment=off

--修改test_alt2表中name字段的存储模式。
gaussdb=# ALTER TABLE test_alt2 ALTER COLUMN name SET STORAGE PLAIN;

--查看。
gaussdb=# \d+ test_alt2
                         Table "public.test_alt2"
 Column |          Type          | Modifiers | Storage | Stats target | Description
--------+------------------------+-----------+---------+--------------+-------------
 id     | integer                |           | plain   |              |
 name   | character varying(35) |           | plain   |              |
Has OIDs: no
Distribute By: HASH(id)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no, storage_type=USTORE, segment=off


--删除。
gaussdb=# DROP TABLE test_alt2;
```

修改约束示例

- **为列添加非空约束**。

```
--建表。
gaussdb=# CREATE TABLE test_alt3(pid INT, areaid CHAR(5), name VARCHAR(20));

--为pid添加非空约束。
gaussdb=# ALTER TABLE test_alt3 MODIFY pid NOT NULL;

--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                | not null
 areaid | character(5)           |
 name   | character varying(20) |

--取消列的非空约束。
gaussdb=# ALTER TABLE test_alt3 MODIFY pid NULL;

--在线为pid添加非空约束。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# ALTER TABLE ONLINE test_alt3 MODIFY pid NOT NULL;

--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                | not null
 areaid | character(5)           |
 name   | character varying(20) |
```

- **取消列的非空约束**。

```
gaussdb=# ALTER TABLE test_alt3 MODIFY pid NULL;
--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                |
 areaid | character(5)           |
 name   | character varying(20) |
```

- **修改字段默认值**。

```
--修改test_alt3表中id的默认值。
gaussdb=# ALTER TABLE test_alt3 ALTER COLUMN areaid SET DEFAULT '00000';

--查看。
gaussdb=# \d test_alt3
                Table "public.test_alt3"
 Column |          Type          |        Modifiers
--------+------------------------+--------------------------
 pid    | integer                |
 areaid | character(5)           | default '00000'::bpchar
 name   | character varying(20) |
--删除id的默认值。
gaussdb=# ALTER TABLE test_alt3 ALTER COLUMN areaid DROP DEFAULT;

--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                |
 areaid | character(5)           |
 name   | character varying(20) |
```

- **添加表级约束**。
  - 直接添加约束。

```
--给表添加主键约束。
gaussdb=# ALTER TABLE test_alt3 ADD CONSTRAINT pk_test3_pid PRIMARY KEY (pid);

--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                | not null
 areaid | character(5)           |
 name   | character varying(20) |
Indexes:
    "pk_test3_pid" PRIMARY KEY, ubtree (pid) WITH (storage_type=USTORE) TABLESPACE
pg_default

--删除表
gaussdb=# DROP TABLE IF EXISTS test_alt3;

--建表。
gaussdb=# CREATE TABLE test_alt3(pid INT, areaid CHAR(5), name VARCHAR(20));

--在线给表添加主键约束。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# ALTER TABLE ONLINE test_alt3 ADD CONSTRAINT pk_test3_pid PRIMARY KEY (pid);

--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                | not null
 areaid | character(5)           |
 name   | character varying(20) |
Indexes:
    "pk_test3_pid" PRIMARY KEY, ubtree (pid) WITH (storage_type=USTORE) TABLESPACE
pg_default
```

- 先创建索引然后再添加约束。

```
--建表。
gaussdb=# CREATE TABLE test_alt4(c1 INT, c2 INT);

--建索引。
gaussdb=# CREATE UNIQUE INDEX pk_test4_c1 ON test_alt4(c1);

--添加约束时关联已经创建的索引。
gaussdb=# ALTER TABLE test_alt4 ADD CONSTRAINT pk_test4_c1 PRIMARY KEY USING INDEX
pk_test4_c1;

--查看。
gaussdb=# \d test_alt4
   Table "public.test_alt4"
 Column |  Type   | Modifiers
--------+---------+-----------
 c1     | integer | not null
 c2     | integer |
Indexes:
    "pk_test4_c1" PRIMARY KEY, ubtree (c1) WITH (storage_type=USTORE) TABLESPACE
pg_default

--删除。
gaussdb=# DROP TABLE test_alt4;
```

- **删除表级约束**。

```
--删除约束。
gaussdb=# ALTER TABLE test_alt3 DROP CONSTRAINT IF EXISTS pk_test3_pid;

--查看。
gaussdb=# \d test_alt3
          Table "public.test_alt3"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pid    | integer                | not null
 areaid | character(5)           |
 name   | character varying(20) |

--删除。
gaussdb=# DROP TABLE test_alt3;

--删除主键约束（该语法仅在MYSQL兼容模式以及ORA兼容模式下支持使用）。
gaussdb=# CREATE TABLE test_drop_primary_key(c1 INT PRIMARY KEY);
gaussdb=# \d test_drop_primary_key
Table "public.test_drop_primary_key"
 Column |  Type   | Modifiers
--------+---------+-----------
 c1     | integer | not null
Indexes:
    "test_drop_primary_key_pkey" PRIMARY KEY, ubtree (c1) WITH (storage_type=USTORE)
TABLESPACE pg_default


gaussdb=# ALTER TABLE test_drop_primary_key DROP PRIMARY KEY;
gaussdb=# \d test_drop_primary_key
Table "public.test_drop_primary_key"
```


版权所有 © 华为云计算技术有限公司

```
 Column |  Type   | Modifiers
--------+---------+-----------
 c1     | integer | not null
```

- **重命名约束。**

```
--创建表tbl_test1,指定主键约束。
gaussdb=# CREATE TABLE tbl_test1(
    id int,
    name varchar(50),
    CONSTRAINT aaa PRIMARY KEY (id)
);

--重命名约束。
gaussdb=# ALTER TABLE tbl_test1 RENAME CONSTRAINT aaa TO pk_test1_id;

--查看信息。
gaussdb=# \d tbl_test1;
          Table "public.tbl_test1"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 id     | integer               | not null
 name   | character varying(50) |
Indexes:
    "pk_test1_id" PRIMARY KEY, ubtree (id) WITH (storage_type=USTORE) TABLESPACE pg_default


--删除表。
gaussdb=# DROP TABLE tbl_test1;
```

相关链接

CREATE TABLE，**DROP TABLE**

1.13.7.35 ALTER TABLE PARTITION

功能描述

修改表分区，包括增加/删除分区、切割/合并分区、清空分区、移动分区表空间、交换分区、重命名分区，以及修改分区属性等。

注意事项

- 只有分区表的所有者或者被授予了分区表ALTER权限的用户有权限执行ALTER TABLE PARTITION命令，当三权分立开关关闭时，系统管理员默认拥有此权限。
- 添加分区的表空间不能是PG_GLOBAL。
- 添加分区的名称不能与该分区表已有分区的名称相同。
- 添加分区的分区键值要和分区表的分区键类型一致。
- 若添加RANGE分区，添加分区键值要大于分区表中最后一个范围分区的上边界。
- 若添加LIST分区，添加分区键值不能与现有分区键值重复。
- 不支持添加哈希分区。
- 如果目标分区表中已有分区数达到了最大值1048575，则不能继续添加分区。
- 当分区表只有一个分区时，不能删除该分区。
- 选择分区使用PARTITION FOR()，括号里指定值个数应该与定义分区时使用的列个数相同，并且一一对应。
- Value分区表不支持Alter Partition操作。

版权所有 © 华为云计算技术有限公司

- 哈希分区表不支持切割分区，不支持合并分区，不支持添加/删除分区。
- 删除、切割、合并、清空、交换分区的操作会使GLOBAL索引失效，可以声明UPDATE GLOBAL INDEX子句同步更新索引。
- 如果删除、切割、合并、清空、交换分区操作不声明UPDATE GLOBAL INDEX子句，并发的DML业务有可能因为索引不可用而报错。
- 删除、切割、合并、清空、交换分区的操作会使分区表上的全局二级索引失效，对于交换分区，同时也会失效普通表上的所有全局二级索引，可以声明UPDATE DISTRIBUTED GLOBAL INDEX子句同步更新索引。
- 如果删除、切割、合并、清空、交换分区操作不声明UPDATE DISTRIBUTED GLOBAL INDEX子句，并发的DML业务有可能因为索引不可用而报错。
- 若设置参数enable_gpi_auto_update为on，即使不声明UPDATE GLOBAL INDEX子句，也会自动更新Global索引。
- 若设置参数enable_drop_truncate_partition_keep_gpi为on，删除、清空分区的操作会保持Global索引有效。此时无论DROP/TRUNCATE PARTITION语句是否带有UPDATE GLOBAL INDEX子句，都会保持Global索引有效。此优先级高于enable_gpi_auto_update参数和用户显式指定的UPDATE GLOBAL INDEX。
- 在为数据对象增加或者变更ILM策略的时候，如果追加了行级表达式，需要注意行表达式目前只支持白名单中列出的函数。具体白名单函数列表参考**行表达式函数白名单**。

  说明

  在ILM策略的ON(EXPR)行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。
- 执行在线DDL进行切割/合并分区、交换分区、移动分区表空间操作时，需预留足够的磁盘空间，即剩余空间是所操作的表或分区（及其上索引等附属品）总和。
- 在线DDL不支持在事务内执行（通过JDBC驱动执行多条SQL时会隐式启动事务，禁止多条SQL一起执行），不支持在存储过程中执行。
- 如果用户取消在线DDL执行过程，首先进入残留清理流程，如果用户再次取消，则会结束清理流程导致残留。
- 在线DDL与用户业务之间存在资源争抢，不适合大业务背景下做在线DDL。对于资源有限的场景，建议采用细粒度资源管控进行操作，且不建议使用并行加速功能。
- M-Compatibility模式数据库下不支持在线DDL操作。
- 以下情况中，在线DDL操作将不会生效，DDL操作默认以离线方式执行，如果指定ONLINE关键字，则会产生忽略ONLINE关键字的提示。
  - 如在线修改表分区语句包含当前在线DDL特性支持操作（在线切割/合并分区、交换分区、移动分区表空间）以外的操作。
  - 在线修改表分区（在线切割/合并分区、交换分区、移动分区表空间）操作的表上包含GSI且需同步更新GSI索引时。
  - 在线修改表分区语句包含多子句操作时（比如语句同时包含在线切割分区和交换分区）。
- 对数据库级、索引级、二级分区表、段页式表、hash bucket表、临时表、unlogged表进行在线DDL操作时（包含在线切割/合并分区、交换分区、移动分区表空间），在线DDL操作将不会生效，DDL操作默认以离线方式执行，并提示当前DDL语句不支持在线执行。

版权所有 © 华为云计算技术有限公司

- 长事务可能会阻塞在线DDL执行结果（包含在线切割/合并分区、交换分区、移动分区表空间），建议避免在存在长事务时进行在线DDL操作 。
- 在线DDL修改表分区（包含在线切割/合并分区、交换分区、移动分区表空间）执行后期将阻塞DQL和DML操作。因此，对于存在超时报错机制的业务模型可能出现阻塞导致的请求失败，使用本特性前需要谨慎评估业务模型对等锁的容忍程度，以确认是否能够进行本特性操作。
- 在线DDL修改表分区执行期间，应尽量避免与其他DDL并发执行，否则可能会出现无响应以及报错等异常，请重新连接客户端进行重试。
- 从不支持在线DDL修改表分区操作的版本升级至支持在线DDL修改表分区的版本，升级待观察期间不支持使用在线DDL修改表分区操作（包含在线切割/合并分区、交换分区、移动分区表空间），DDL默认以离线方式执行，并提示升级期间不支持在线DDL操作。
- 在线DDL修改表分区执行（包含在线切割/合并分区、交换分区、移动分区表空间）过程中会生成指定的Schema，Schema命名格式为online$$ddl$$[(表所在的Schema 表名)的hash值]。
  - Schema的属主默认为初始用户，在线DDL执行中不建议操作此Schema，可能会导致中止在线DDL并产生未知错误。
  - 用户不可创建online$$为前缀的Schema，否则会产生报错提示。若已存在同名Schema冲突，在线DDL操作将不会生效，DDL默认以离线方式执行，并提示Schema名冲突导致当前DDL语句不可在线。
- 若在线DDL（包含在线切割/合并分区、移动分区表空间）操作失败时，表上可能会存在明显存储空间膨胀。在线分区交换操作无论失败与否，表上可能会有明显存储空间膨胀。其膨胀程度与DDL执行时长以及并发写操作数成正比，建议避免在存在长事务时进行在线DDL操作。
- 在线DDL（包含在线切割/合并分区、交换分区、移动分区表空间）操作继承**CREATE TABLE**的使用约束。
- 在线DDL合并分区操作时，若待合并分区的数据在操作前已被压缩，可能存在在线DDL合并操作完成后数据被解压的情况。
- 在线DDL（包含在线切割/合并分区、交换分区、移动分区表空间）若涉及自动更新Global索引时，在线DDL执行后期可能存在长时间阻塞并发查询、DML业务。

语法格式

修改分区表分区包括修改表分区主语法、修改表分区名称的语法和重置分区ID的语法。

- 修改表分区主语法。

```
ALTER TABLE [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ] [ IF EXISTS ]
{ table_name  [*] | ONLY table_name | ONLY ( table_name  )}
    action [, ... ];
```

其中action统指如下分区维护子语法。当存在多个分区维护子句时，保证了分区的连续性，无论这些子句的排序如何，GaussDB总会先执行DROP PARTITION再执行ADD PARTITION操作，最后顺序执行其它分区维护操作。

```
{ move_clause  |
    exchange_clause  |
    row_clause  |
    merge_clause  |
    modify_clause  |
    split_clause  |
    add_clause  |
    drop_clause  |
    truncate_clause  |
```


版权所有 © 华为云计算技术有限公司

```
  ilm_clause |
  set_partitioning_clause }
```

- move_clause子语法用于移动分区到新的表空间。

  ```
  MOVE PARTITION { partition_name | FOR ( partition_value [, ...] ) } TABLESPACE tablespacename
  ```

- exchange_clause子语法用于把普通表的数据迁移到指定的分区。

  ```
  EXCHANGE PARTITION { ( partition_name ) | partition_name | FOR ( partition_value [, ...] ) }
     WITH TABLE {[ ONLY ] ordinary_table_name | ordinary_table_name * | ONLY
  ( ordinary_table_name )}
     [ { WITH | WITHOUT } VALIDATION ] [ VERBOSE ] [ UPDATE GLOBAL INDEX ] [ UPDATE
  DISTRIBUTED GLOBAL INDEX | NO UPDATE DISTRIBUTED GLOBAL INDEX ]
  ```

  进行交换的普通表和分区必须满足如下条件：

  - 普通表和分区的列数相同，对应列的信息严格一致，包括：列名、列的数据类型、列约束、列的Collation信息、列的存储参数、列的压缩信息等。
  - 普通表和分区的表压缩信息严格一致。
  - 普通表和分区的分布列信息严格一致。
  - 普通表索引和分区Local索引个数相同，且对应索引的信息严格一致。
  - 普通表和分区的表约束个数相同，且对应表约束的信息严格一致。
  - 普通表和分区表不能被其他表外键约束关联。
  - 普通表不可以是临时表，分区表只能是范围分区表，列表分区表，哈希分区表或间隔分区表。
  - 在内置安全策略开关开启的情况下，普通表和分区表上不可以有动态数据脱敏，行访问控制约束。

版权所有 © 华为云计算技术有限公司

须知

- 完成交换后，普通表和分区的数据被置换，同时普通表和分区的表空间信息被置换。此时，普通表和分区的统计信息变得不可靠，需要对普通表和分区重新执行analyze。
- 由于非分区键不能建立本地唯一索引，只能建立全局唯一索引，所以如果普通表含有唯一索引时，可能会导致无法交换数据。

  如果需要进行数据交换操作，可以通过创建中间表的方式，先将分区数据插入到中间表，truncate分区，普通表数据插入分区表，drop普通表，重命名中间表的方式完成数据交换操作。
- 对于普通表和分区表都是Ustore的场景，如果普通表的Ubtree索引类型（RCR或者PCR，默认为RCR）和分区表本地的Ubtree索引类型（RCR或PCR，默认为RCR）不一致，会导致无法完成数据交换的操作。
- 如果在普通表/分区表上进行了DROP COLUMN操作，被删除的列依然物理存在，则需要保证普通表和分区的被删除列严格对齐才能交换成功。
- EXCHANGE PARTITION { (partition_name) | partition_name | FOR (partition_value [, ...]) }操作在MYSQL模式数据库（即sql_compatibility = 'MYSQL'）下可用，其他模式下仅EXCHANGE PARTITION { (partition_name) | FOR (partition_value [, ...]) }可用。当partition_name为一级分区名时，进行交换的是一级分区和普通表。
- 当执行交换分区操作的普通表和分区包含rowid列且处于压缩状态时，进行交换分区后，普通表和分区的数据将被自动解压。

– row_clause子语法用于设置分区表的行迁移开关。

```
{ ENABLE | DISABLE } ROW MOVEMENT
```

– merge_clause子语法用于把多个分区合并成一个分区。一个命令中合并的源分区上限为300。

```
MERGE PARTITIONS { partition_name } [, ...] INTO PARTITION partition_name
   [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]
[ TABLESPACE tablespacename ] [ UPDATE GLOBAL INDEX ] [ UPDATE DISTRIBUTED GLOBAL INDEX | NO UPDATE DISTRIBUTED GLOBAL INDEX ]
```

须知

- 对于范围分区/间隔分区，MERGE分区要求源分区的范围连续递增，且MERGE后的分区名可以与最后一个源分区名相同；对于列表分区，则源分区无顺序要求，且MERGE后的分区名可以与任一源分区名相同。如果MERGE后的分区名与源分区名相同，视为同一个分区。
- 未打开guc参数enable_ilm的情况下，如果使用merge_clause子语法把多个带有ilm policy的分区合并成一个分区，新分区不继承ilm policy。
- USTORE存储引擎表不支持在事务块/存储过程中执行ALTER TABLE MERGE PARTITIONS的操作。

– modify_clause子语法用于设置分区索引是否可用。

```
MODIFY PARTITION partition_name { UNUSABLE LOCAL INDEXES | REBUILD UNUSABLE LOCAL INDEXES | COLVIEW [ PRIORITY { HIGH | LOW | NONE } ] | NOCOLVIEW }
```

– split_clause子语法用于把一个分区切割成多个分区。

版权所有 © 华为云计算技术有限公司

```
SPLIT PARTITION { partition_name | FOR ( partition_value [, ...] ) } { split_point_clause | no_split_point_clause } [ UPDATE GLOBAL INDEX ] [ UPDATE DISTRIBUTED GLOBAL INDEX | NO UPDATE DISTRIBUTED GLOBAL INDEX ]
```

须知

- SPLIT后的分区名可以与源分区名相同，将视为不同的分区。
- 未打开guc参数enable_ilm的情况下，如果使用split_clause子语法把一个带有ilm policy的分区分割成多个分区，新分区不继承ilm policy。

- 范围分区表指定切割点split_point_clause的语法为：

```
AT ( partition_value ) INTO ( PARTITION partition_name [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] , PARTITION partition_name [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] )
```

  须知

  切割点的大小要位于正在被切割的分区的分区键范围内，指定切割点的方式只能把一个分区切割成两个新分区。

- 范围分区表不指定切割点no_split_point_clause的语法为：

```
INTO { ( partition_less_than_item [, ...] ) | ( partition_start_end_item [, ...] ) }
```

  须知

  - 不指定切割点的方式，partition_less_than_item指定的第一个新分区的分区键要大于正在被切割的分区的前一个分区（如果存在）的分区键，partition_less_than_item指定的最后一个分区的分区键要等于正在被切割的分区的分区键。
  - 不指定切割点的方式，partition_start_end_item指定的第一个新分区的起始点（如果存在）必须等于正在被切割的分区的前一个分区（如果存在）的分区键，partition_start_end_item指定的最后一个分区的终止点（如果存在）必须等于正在被切割的分区的分区键。
  - partition_less_than_item支持的分区键个数最多为16，而partition_start_end_item仅支持1个分区键，其支持的数据类型参见**CREATE TABLE PARTITION**章节的**PARTITION BY RANGE(partition_key)**。
  - 在同一语句中partition_less_than_item和partition_start_end_item两者不可同时使用；不同split语句之间没有限制。

- 分区项partition_less_than_item的语法为如下，其中最后一个分区可以不写分区范围定义，即VALUES LESS THAN (partition_value)部分，默认继承源分区范围定义的上界值。

```
PARTITION partition_name VALUES LESS THAN {( { partition_value | MAXVALUE } [, ...] ) | MAXVALUE }
```


版权所有 © 华为云计算技术有限公司

```
  [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ]
```

须知

RANGE分区时支持MAXVALUE关键字不带括号，只能支持MYSQL模式使用，不支持在二级分区的子分区中使用，不支持在分区字段为多列的场景使用。

- 分区项partition_start_end_item的语法为，其约束参见**CREATE TABLE PARTITION**章节的**START END语法使用限制**。

```
PARTITION partition_name {
      {START(partition_value) END (partition_value) EVERY (interval_value)} |
      {START(partition_value) END ({partition_value | MAXVALUE})} |
      {START(partition_value)} |
      {END({partition_value | MAXVALUE})}
} [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [TABLESPACE tablespace_name]
```

- 列表分区表指定切割点split_point_clause的语法如下：

```
VALUES ( partition_value_list ) INTO ( PARTITION partition_name [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] , PARTITION partition_name [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] )
```

须知

切割点必须是源分区的一个非空真子集，指定切割点的方式只能把一个分区切割成两个新分区。

- 列表分区表不指定切割点no_split_point_clause的语法如下：

```
INTO ( PARTITION partition_name VALUES (partition_value_list) [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ][, ...] )
```

须知

- 其中最后一个分区不能写分区范围定义，即VALUES (partition_value_list)部分，其范围等于源分区去掉其他子分区后的剩余集合。
- 不指定切割点的方式，每一个新分区都必须是源分区的一个非空真子集，且互不交叉。

– add_clause子语法用于为指定的分区表添加一个或多个分区。

```
ADD {{partition_less_than_item | partition_start_end_item} |
   PARTITION({partition_less_than_item | partition_start_end_item | partition_list_item})}
```


版权所有 © 华为云计算技术有限公司

须知

- PARTITION({partition_less_than_item | partition_start_end_item | partition_list_item})语法只支持MYSQL模式下使用。
- 不支持ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition2,…);语法添加多分区。仅支持原有添加多分区语法：ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2), …。

分区项partition_list_item的语法为：

```
PARTITION partition_name VALUES [IN] (list_values_clause)
   [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]
[ TABLESPACE tablespacename ]
```

须知

- partition_list_item支持最多16个分区键，其支持的数据类型参见**CREATE TABLE PARTITION**章节的**PARTITION BY LIST [COLUMNS] (partition_key)**。
- 哈希分区表不支持添加分区。
- IN需要在MYSQL模式下使用，不支持在二级分区的子分区中使用。

– drop_clause子语法用于删除分区表中的指定分区。

```
DROP PARTITION  { partition_name | FOR (  partition_value [, ...] )  } [ UPDATE GLOBAL INDEX ] [ UPDATE DISTRIBUTED GLOBAL INDEX | NO UPDATE DISTRIBUTED GLOBAL INDEX ]
```

须知

- 哈希分区表不支持删除分区。
- 当分区表只有一个分区时，不能删除该分区。

– truncate_clause子语法用于清空分区表中的指定分区。

```
TRUNCATE PARTITION  { partition_name | FOR (  partition_value [, ...] )  } [ UPDATE GLOBAL INDEX ] [ UPDATE DISTRIBUTED GLOBAL INDEX | NO UPDATE DISTRIBUTED GLOBAL INDEX ]
```

– ilm_clause子语法用于为分区添加ILM策略，为数据生命周期管理-OLTP表压缩特性支持语法。

```
MODIFY PARTITION partition_name ILM ADD POLICY ROW STORE COMPRESS ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR ) ]
```

– set_partitioning_clause子语法用于分区设置存储参数。仅支持在线DDL内部调用，不建议用户使用。

```
SET ({storage_parameter = value} )
```

- 修改表分区名称的语法。

```
ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name  )}
   RENAME PARTITION { partition_name | FOR ( partition_value [, ...] ) } TO partition_new_name;
```

- 重置分区ID的语法。

```
ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name  )} RESET PARTITION;
```


版权所有 © 华为云计算技术有限公司

参数说明

- **ONLINE | OFFLINE**

  该关键字用于控制是否开启在线DDL。

  使用ONLINE | OFFLINE关键字：

  - ONLINE：DDL以在线方式执行。此时，若ALTER TABLE语句中包含当前DDL特性不支持的场景时，DDL以离线方式执行，并通过NOTICE提示忽略ONLINE关键字。
  - OFFLINE：DDL以离线方式执行。

  说明

  - 在线DDL修改在线交换分区时，会检查数据是否在对应分区的范围内。若检查结果为不在范围内，DDL操作则会报错。在线DDL可并发DML，可能会出现并发DML将不在对应分区范围的数据删除，但DDL仍然报错的情况。

    例如：在线DDL扫描表检查元组A是否在对应分区的范围内时，若表上元组A并不在对应分区的范围，在线DDL的事务扫描到元组A时，会立即报错。如果在同一时间使用并行事务DELETE把元组A删除。但在线DDL扫描检查的事务早于并行事务DELETE，因此不会认为元组A已被删除。
  - 在线DDL错误故障处理：包含在线DDL修改列类型、精度、长度或在线添加非空、检查、主键、唯一约束时，系统会自动创建临时模式作为DDL实现工具并将表设置为追加（append_mode）状态。若在线DDL正常执行完成，则临时模式会自动清理和恢复。自动清理过程可能由于等锁超时等原因而失败，无法进行自动清理，需要手动清理恢复。若发生严重故障，如FATAL、PANIC、数据库故障和节点断连等故障时，创建的临时模式和表的追加状态可能残留，需要手动清理和恢复。包含如下情况：
    - 取消执行在线DDL：执行过程中取消在线DDL，系统会自动清理。用户取消执行后首先进入残留清理流程，此时如果用户再次取消，则会结束清理流程导致残留。
    - 操作报错：当DDL命令执行过程中出现需要报错场景，如类型转换无法执行或长度修改损伤数据等，系统会报错并自动清理。
    - 严重故障：用户待集群正常后，可通过**online_ddl_cleanup()**进行手动清理，或在下次执行在线DDL或在线VACUUM FULL时自动清理。
  - 不建议在线DDL执行中**online_ddl_cleanup()**清理操作，可能会导致中止在线DDL或产生未知错误。

- **online_parameter**

  当指定ONLINE关键字时，用于指定在线DDL选项。若未指定ONLINE关键字或不满足在线DDL的条件，该参数不生效。

  取值范围：

  - parallel_threads：表示开启并行时的线程数。取值范围：1-32之间的整数值，缺省时默认值为1。

    - 当parallel_threads设置为大于1时，开启多线程，采用线程数为指定线程数。
    - 当不设置parallel_threads或设置1，关闭多线程加速。

      当用户开启资源负载管理时，比如启用IO资源管理 (如io_limits>0或io_priority = high/medium/low)，parallel_threads将忽略用户/会话下的资源负载管理，并给出提示当前资源负载管理无效并开启多线程加速。在线DDL多线程执行过程实际遵循初始用户的资源负载管理。
    - 其中关于索引并行parallel_workers参数和parallel_workers参数共同作用加速时，parallel_workers参数控制（最大）索引内并行线程数，parallel_threads参数控制（最大）索引间并行数。

版权所有 © 华为云计算技术有限公司

  - max_catchup_times：表示在线DDL的最大追增轮次，防止在线DDL执行时间过长。1-50之间的整数值，缺省时默认值为50。

    - 在线DDL会记录数据重建过程中的DML增量数据到临时表，并采用多轮次循环追增，将增量数据进行重建。为防止由于DML并发数据量过大，在线DDL无法完成追增的情况，用户可以设置该参数，以限制追增次数。

- **table_name**

  分区表名。

  取值范围：已存在的分区表名。

- **partition_name**

  分区名。

  取值范围：已存在的分区名。

- **tablespacename**

  指定分区要移动到哪一个表空间。

  取值范围：已存在的表空间名。

- **partition_value**

  分区键值。

  通过PARTITION FOR（partition_value [, ...])子句指定的这一组值，可以唯一确定一个分区。

  取值范围：需要进行操作的分区的分区键的取值范围。

- **UNUSABLE LOCAL INDEXES**

  设置该分区上的所有索引不可用。

- **REBUILD UNUSABLE LOCAL INDEXES**

  重建该分区上的所有索引。

- **{ ENABLE | DISABLE } ROW MOVEMENT**

  行迁移开关。

  如果进行UPDATE操作时，更新了元组在分区键上的值，造成了该元组所在分区发生变化，就会根据该开关给出报错信息，或者进行元组在分区间的转移。

  取值范围：

  - ENABLE：打开行迁移开关。
  - DISABLE：关闭行迁移开关。

  默认是关闭状态。

- **ordinary_table_name**

  进行迁移的普通表的名称。

  取值范围：已存在的普通表名。

- **{ WITH | WITHOUT } VALIDATION**

  在进行数据迁移时，是否检查普通表中的数据满足指定分区的分区键范围。

  取值范围：

  - WITH：对于普通表中的数据要检查是否满足分区的分区键范围，如果有数据不满足，则报错。
  - WITHOUT：对于普通表中的数据不检查是否满足分区的分区键范围。

  默认是WITH状态。

版权所有 © 华为云计算技术有限公司

由于检查比较耗时，特别是当数据量很大的情况下。所以在保证当前普通表中的数据满足分区的分区键范围时，可以加上WITHOUT来指明不进行检查。

- **VERBOSE**

  在VALIDATION是WITH状态时，如果检查出普通表有不满足要交换分区的分区键范围的数据，那么把这些数据插入到正确的分区，如果路由不到任何分区，再报错。

  须知

  只有在VALIDATION是WITH状态时，才可以指定VERBOSE。

- **partition_new_name**

  分区的新名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **UPDATE GLOBAL INDEX**

  如果使用该参数，则会更新分区表上的所有全局索引，以确保使用全局索引可以查询出正确的数据。

  如果不使用该参数，则分区表上的所有全局索引将会失效。

- **UPDATE DISTRIBUTED GLOBAL INDEX**

  如果使用该参数，则会更新分区表上的所有可用的全局二级索引，使得全局二级索引和基表的数据一致。对于EXCHANGE PARTITION，也会更新普通表上的所有可用全局二级索引。

  如果不使用该参数，则会使分区表上的所有全局二级索引失效。对于EXCHANGE PARTITION，也会使普通表上的所有全局二级索引失效。

- **NO UPDATE DISTRIBUTED GLOBAL INDEX**

  如果使用该参数，则会使分区表上的所有全局二级索引失效。对于EXCHANGE PARTITION，也会使普通表上的所有全局二级索引失效。

- **storage_parameter**

  表的存储参数的名称。

  仅支持在线DDL内部调用，不建议用户使用：

  - online_ddl_status（枚举类型）

    本参数用于更新目标表在线DDL执行状态。

    仅支持在线DDL内部调用，不建议用户使用。

    - on：开启在线DDL状态。
    - off：关闭在线DDL状态。
    - init：进行在线DDL临时对象初始化。
    - catchup：进行在线DDL操作的并发新增数据追增同步。
    - catchup_basedata: 进行在线DDL操作的基础数据同步。
    - scan_catchup：进行在线添加非空/检查约束操作的数据追增。
    - scan_catchup_basedata：进行在线添加非空/检查约束操作的基础数据检查。

版权所有 © 华为云计算技术有限公司

  - vacuum_catchup：进行在线VACUUM FULL操作的数据追增。
  - vacuum_verbose_catchup：进行在线VACUUM FULL VERBOSE操作的数据追增。
  - last_catchup：进入在线DDL最后一轮追增的写报错模式。
  - switch：进行在线DDL最后一轮追增的临时表交换。
  - copy_switch：进行在线移动表空间操作最后一轮追增的临时表交换。
  - record_toast_oid：记录在线DDL操作目标表附带的TOAST表的OID。
  - abort：在线DDL进入abort状态。
  - flush_buffer：用于在线移动表空间操作中强制buffer落盘。
  - tmp_rel：用于标识该表为在线DDL过程创建的临时表。
  - validate_constr：用于对NOT VALID选项创建的检查类约束进行约束验证
  - vacuum_switch: 进行在线vacuum full最后一轮追增的临时表交换。
- online_ddl_lock_tag（int类型）

  用于记录在线DDL目标表的唯一标识（“表所在的Schema+表名”的hash值），用于生成临时对象。

  仅支持在线DDL内部调用，不建议用户使用。

示例

- 修改表分区名称。

```
--创建前置分区表。
gaussdb=# CREATE TABLE test_p1 (col1 INT, col2 INT) PARTITION BY RANGE (col1)
(
    PARTITION p1 VALUES LESS THAN (10),
    PARTITION p2 VALUES LESS THAN (20),
    PARTITION p3 VALUES LESS THAN (MAXVALUE)
);

--修改分区名称。
gaussdb=# ALTER TABLE test_p1 RENAME PARTITION p3 TO pmax;

--查询分区信息。
gaussdb=# SELECT relname, boundaries, oid FROM pg_partition WHERE parentid='test_p1'::regclass
AND parttype <> 'r';
 relname | boundaries |  oid
---------+------------+-------
 p1      | {10}       | 17066
 p2      | {20}       | 17067
 pmax    | {NULL}     | 17068
(3 rows)
```

- 移动分区表空间。

```
--创建分区。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace1/tbs_data1';

--移动分区表空间。
gaussdb=# ALTER TABLE test_p1 MOVE PARTITION P1 TABLESPACE tbs_data1;

--查看分区表空间。
gaussdb=# SELECT relname, spcname FROM pg_partition t1, pg_tablespace t2 WHERE
```


版权所有 © 华为云计算技术有限公司

```
T1.reltablespace=t2.oid and t1.parentid='test_p1'::regclass;
 relname |  spcname
---------+-----------
 p1      | tbs_data1
(1 row)

--在线移动分区表空间。
gaussdb=# ALTER TABLE test_p1 MOVE PARTITION P1 TABLESPACE pg_default;
gaussdb=# ALTER TABLE ONLINE test_p1 MOVE PARTITION P1 TABLESPACE tbs_data1;
ALTER TABLE
gaussdb=# SELECT relname, spcname FROM pg_partition t1, pg_tablespace t2 WHERE
T1.reltablespace=t2.oid and t1.parentid='t1'::regclass;
 relname | spcname
---------+---------
 p1      | tbs_data1
(1 row)
```

- 分区交换。

```
--创建普通表，插入数据。
gaussdb=# CREATE TABLE test_ep1(col1 INT,col2 INT);
gaussdb=# INSERT INTO test_ep1 VALUES (GENERATE_SERIES(1,30), 1000);

--迁移普通表数据到指定分区。
gaussdb=# ALTER TABLE test_p1 EXCHANGE PARTITION (p1) WITH TABLE test_ep1 VERBOSE;

--查询。
gaussdb=# SELECT COUNT(*) FROM test_p1 PARTITION (p1);
 count
-------
     9
(1 row)

--清除表数据。
gaussdb=# TRUNCATE TABLE test_ep1;
TRUNCATE TABLE
--插入数据。
gaussdb=# INSERT INTO test_ep1 VALUES (GENERATE_SERIES(1,30), 1000);
INSERT 0 30
--在线迁移普通表数据到指定分区。
gaussdb=# ALTER TABLE ONLINE test_p1 EXCHANGE PARTITION (p1) WITH TABLE test_ep1 VERBOSE;
ALTER TABLE
--查询。
gaussdb=# SELECT COUNT(*) FROM test_p1 PARTITION (p1);
 count
-------
     9
(1 row)

--删除表test_ep1。
gaussdb=# DROP TABLE test_ep1;
```

- 分区合并。

```
--将test_p1表中p2,pmax分区合并到pmax中。
gaussdb=# ALTER TABLE test_p1 MERGE PARTITIONS p2,pmax INTO PARTITION pmax;

--查看分区。
gaussdb=# SELECT relname, boundaries, oid FROM pg_partition WHERE parentid='test_p1'::regclass
AND parttype <> 'r' order by 1;
 relname | boundaries |  oid
---------+------------+-------
 p1      | {10}       | 17066
 pmax    | {NULL}     | 17070
(2 rows)

--删除表。
gaussdb=# DROP TABLE test_p1;
DROP TABLE
--创建前置分区表。
gaussdb=# CREATE TABLE test_p1 (col1 INT, col2 INT) PARTITION BY RANGE (col1)
(
```


版权所有 © 华为云计算技术有限公司

```
    PARTITION p1 VALUES LESS THAN (10),
    PARTITION p2 VALUES LESS THAN (20),
    PARTITION p3 VALUES LESS THAN (MAXVALUE)
);
--在线将test_p1表中p2,pmax分区合并到pmax中。
gaussdb=# ALTER TABLE ONLINE test_p1 MERGE PARTITIONS p2,p3 INTO PARTITION pmax;
ALTER TABLE
gaussdb=# SELECT relname, boundaries FROM pg_partition WHERE parentid='test_p1'::regclass AND
parttype <> 'r' order by 1;
 relname | boundaries
---------+------------
 p1      | {10}
 pmax    | {NULL}
(2 rows)

--删除表和表空间。
gaussdb=# DROP TABLE test_p1;
gaussdb=# DROP TABLESPACE tbs_data1;
```

- 切割分区。

```
--建表。
gaussdb=# CREATE TABLE test_r1 (col1 INT,col2 INT) PARTITION BY RANGE (col1)(
    PARTITION p1   VALUES LESS THAN (10),
    PARTITION pmax VALUES LESS THAN (MAXVALUE)
);

--切割分区。
gaussdb=# ALTER TABLE test_r1 SPLIT PARTITION pmax AT (20) INTO (PARTITION p2, PARTITION
pmax);
gaussdb=# ALTER TABLE test_r1 SPLIT PARTITION pmax INTO (
    PARTITION p3   VALUES LESS THAN (30),
    PARTITION pmax VALUES LESS THAN (MAXVALUE)
);

--查询。
gaussdb=# SELECT relname, boundaries, oid FROM pg_partition WHERE parentid='test_r1'::regclass
AND parttype <> 'r' order by 1;
 relname | boundaries |  oid
---------+------------+-------
 p1      | {10}       | 17088
 p2      | {20}       | 17090
 p3      | {30}       | 17092
 pmax    | {NULL}     | 17093
(4 rows)

--删除表。
gaussdb=# DROP TABLE test_r1;
DROP TABLE
--建表。
gaussdb=# CREATE TABLE test_r1 (col1 INT,col2 INT) PARTITION BY RANGE (col1)(
    PARTITION p1   VALUES LESS THAN (10),
    PARTITION pmax VALUES LESS THAN (MAXVALUE)
);
--在线切割分区。
gaussdb=# ALTER TABLE ONLINE test_r1 SPLIT PARTITION pmax AT (20) INTO (PARTITION p2,
PARTITION pmax);
ALTER TABLE
gaussdb=# ALTER TABLE ONLINE test_r1 SPLIT PARTITION pmax INTO (
    PARTITION p3   VALUES LESS THAN (30),
    PARTITION pmax VALUES LESS THAN (MAXVALUE)
);
ALTER TABLE
gaussdb=#  SELECT relname, boundaries FROM pg_partition WHERE parentid='test_r1'::regclass AND
parttype <> 'r' order by 1;
 relname | boundaries
---------+------------
 p1      | {10}
 p2      | {20}
 p3      | {30}
 pmax    | {NULL}
```


版权所有 © 华为云计算技术有限公司

```
(4 rows)

--删除表test_r1。
gaussdb=# DROP TABLE test_r1;
--建表。
gaussdb=# CREATE TABLE test_r2(col1 INT, col2 INT) PARTITION BY RANGE (col1)(
    PARTITION p1   START(1) END(10),
    PARTITION p2   START(10) END(20),
    PARTITION pmax START(20) END(MAXVALUE)
);

--切割分区。
gaussdb=# ALTER TABLE test_r2 SPLIT PARTITION pmax INTO (
    PARTITION p3 START(20) END(30),
    PARTITION pmax START(30) END (MAXVALUE)
);

--查看。
gaussdb=# SELECT relname, boundaries, oid FROM pg_partition WHERE parentid='test_r2'::regclass
AND parttype <> 'r' order by 1;
 relname | boundaries |  oid
---------+------------+-------
 p1_0    | {1}        | 17112
 p1_1    | {10}       | 17113
 p2      | {20}       | 17114
 p3      | {30}       | 17116
 pmax    | {NULL}     | 17117
(5 rows)

--删除表。
gaussdb=# DROP TABLE test_r1;
DROP TABLE
--建表。
gaussdb=# CREATE TABLE test_r2(col1 INT, col2 INT) PARTITION BY RANGE (col1)(
    PARTITION p1   START(1) END(10),
    PARTITION p2   START(10) END(20),
    PARTITION pmax START(20) END(MAXVALUE)
);
--在线切割分区。
gaussdb=# ALTER TABLE ONLINE test_r2 SPLIT PARTITION pmax INTO (
    PARTITION p3 START(20) END(30),
    PARTITION pmax START(30) END (MAXVALUE)
);
ALTER TABLE
--查看。
gaussdb=# SELECT relname, boundaries FROM pg_partition WHERE parentid='test_r2'::regclass AND
parttype <> 'r' order by 1;
 relname | boundaries
---------+------------
 p1_0    | {1}
 p1_1    | {10}
 p2      | {20}
 p3      | {30}
 pmax    | {NULL}
(5 rows)

--删除表test_r2。
gaussdb=# DROP TABLE test_r2;
--建表。
gaussdb=# CREATE TABLE test_l1(col1 INT, col2 INT) PARTITION BY LIST(col1)(
    PARTITION p1   VALUES (10,20),
    PARTITION p2 VALUES (30,40)
);

--切割分区。
gaussdb=# ALTER TABLE test_l1 SPLIT PARTITION p1 VALUES (10) INTO (PARTITION p1_1, PARTITION
p1_2);
gaussdb=# ALTER TABLE test_l1 SPLIT PARTITION p2 INTO (PARTITION p3_1 VALUES(30), PARTITION
p3_2);
```


版权所有 © 华为云计算技术有限公司

```
--查看。
gaussdb=# SELECT relname, boundaries, oid FROM pg_partition WHERE parentid='test_l1'::regclass
AND parttype <> 'r' order by 1;
 relname | boundaries |  oid
---------+------------+-------
 p1_1    | {10}       | 17132
 p1_2    | {20}       | 17133
 p3_1    | {30}       | 17134
 p3_2    | {40}       | 17135
(4 rows)

--删除表test_l1。
gaussdb=# DROP TABLE test_l1;
```

- 添加分区。

```
--建表。
gaussdb=# CREATE TABLE test_p2 (col1 INT, col2 INT) PARTITION BY RANGE (col1)(
    PARTITION p1 VALUES LESS THAN (10),
    PARTITION p2 VALUES LESS THAN (20)
);

--添加分区。
gaussdb=# ALTER TABLE test_p2 ADD PARTITION p3 VALUES LESS THAN (30);

--删除表test_p2。
gaussdb=# DROP TABLE test_p2;
--建表。
gaussdb=# CREATE TABLE test_p3 (col1 INT, col2 INT) PARTITION BY LIST(col1)(
    PARTITION p1 VALUES (1),
    PARTITION p2 VALUES (2)
);

--添加分区。
gaussdb=# ALTER TABLE test_p3 ADD PARTITION p3 VALUES (3);

--删除表test_p3。
gaussdb=# DROP TABLE test_p3;
```

- 删除分区。

```
--建表。
gaussdb=# CREATE TABLE test_p4 (col1 INT, col2 INT) PARTITION BY LIST(col1)(PARTITION p1
VALUES (1),PARTITION p2 VALUES (2));

--删除test_p4表的p2分区。
gaussdb=# ALTER TABLE test_p4 DROP PARTITION p2;

--查看。
gaussdb=# SELECT relname, boundaries, oid FROM pg_partition WHERE parentid='test_p4'::regclass;
 relname | boundaries |  oid
---------+------------+-------
 test_p4 |            | 17187
 p1      | {1}        | 17188
(2 rows)

--删除表test_p4。
gaussdb=# DROP TABLE test_p4;

--指定partition value删除分区。
--建表。
gaussdb=# CREATE TABLE test_p4 (col1 INT, col2 INT) PARTITION BY RANGE(col1)(PARTITION p1
VALUES LESS THAN(1),PARTITION p2 VALUES LESS THAN (2));

--删除test_p4表的分区键为1时所处的分区。
gaussdb=# ALTER TABLE test_p4 DROP PARTITION FOR (1);

--查看。
gaussdb=# SELECT relname, boundaries FROM pg_partition WHERE parentid='test_p4'::regclass order
by 1 desc;
 relname | boundaries
```


版权所有 © 华为云计算技术有限公司

```
---------+------------
 test_p4 |
 p1      | {1}
(2 rows)

--删除表test_p4。
gaussdb=# DROP TABLE test_p4;
```

- 清空分区。

```
--建表。
gaussdb=# CREATE TABLE test_p5 (col1 INT, col2 INT) PARTITION BY RANGE (col1)(
    PARTITION p1 VALUES LESS THAN (5),
    PARTITION p2 VALUES LESS THAN (10)
);

--插入数据。
gaussdb=# INSERT INTO test_p5 VALUES (GENERATE_SERIES(1,9), 100);

--查看p2分区数据。
gaussdb=# SELECT * FROM test_p5 PARTITION (p2);
 col1 | col2
------+------
    5 |  100
    6 |  100
    7 |  100
    8 |  100
    9 |  100
(5 rows)

--清空p2分区的数据。
gaussdb=# ALTER TABLE test_p5 TRUNCATE PARTITION p2;

--查看p2分区数据。
gaussdb=# SELECT * FROM test_p5 PARTITION (p2);
 col1 | col2
------+------
(0 rows)

--删除表test_p5。
gaussdb=# DROP TABLE test_p5;
```

相关链接

CREATE TABLE PARTITION，**DROP TABLE**

1.13.7.36 ALTER TABLE SUBPARTITION

功能描述

修改二级分区表分区，包括增删分区、清空分区、切割/合并分区、移动分区表空间、交换分区、重命名分区，以及修改分区属性等。

注意事项

- 添加分区的表空间不能是PG_GLOBAL。
- 添加分区的名称不能与该分区表已有一级分区和二级分区的名称相同。
- 添加分区的分区键值要和分区表的分区键的类型一致。
- 若添加RANGE分区，添加分区键值要大于分区表中最后一个范围分区的上边界。若需要在有MAXVALUE分区的表上新增分区，建议使用SPLIT语法。
- 若添加LIST分区，添加分区键值不能与现有分区键值重复。若需要在有DEFAULT分区的表上新增分区，建议使用SPLIT语法。

版权所有 © 华为云计算技术有限公司

- 不支持添加HASH分区。只有一种情况例外，二级分区表的二级分区方式为HASH且一级分区方式不是HASH，此时支持新增一级分区并创建对应的二级分区。
- 如果目标分区表中已有分区数达到了最大值1048575，则不能继续添加分区。
- 当分区表只有一个一级分区或二级分区时，不能删除该分区。
- 不支持删除HASH分区。
- 选择分区使用PARTITION FOR()或SUBPARTITION FOR()，括号里指定值个数应该与定义分区时使用的列个数相同，并且一一对应。
- 切割分区只能对二级分区（叶子节点）进行切割，被切割分区只能是RANGE、LIST分区策略，不支持切割 HASH分区策略。
- 合并分区只能对二级分区（叶子节点）进行合并，且源分区必须属于同一个一级分区。
- 只有分区表的所有者或者被授予了分区表ALTER权限的用户可以执行ALTER TABLE PARTITION命令，系统管理员默认拥有此权限。
- 删除、切割、清空、交换分区的操作会使Global索引失效，可以声明UPDATE GLOBAL INDEX子句同步更新索引。
- 如果删除、切割、清空、交换分区操作不声明UPDATE GLOBAL INDEX子句，并发的DML业务有可能因为索引不可用而报错。
- 若设置参数enable_gpi_auto_update为on，即使不声明UPDATE GLOBAL INDEX子句，也会自动更新Global索引。
- 若设置参数enable_drop_truncate_partition_keep_gpi为on，删除、清空分区的操作会保持Global索引有效。此时无论DROP/TRUNCATE PARTITION语句是否带有UPDATE GLOBAL INDEX子句，都会保持Global索引有效。此优先级高于enable_gpi_auto_update参数和用户显式指定的UPDATE GLOBAL INDEX。
- 在为数据对象增加或者变更ILM策略的时候，如果追加了行级表达式，需要注意行表达式目前只支持白名单中列出的函数。具体白名单函数列表请参见**行表达式函数白名单**。

  说明

  在ILM策略的ON（EXPR）行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。

语法格式

修改二级分区表分区包括修改表分区主语法、修改表分区名称的语法、重置分区ID的语法。

- 修改表分区主语法。

```
ALTER TABLE [ IF EXISTS ] { table_name  [*] | ONLY table_name | ONLY ( table_name  )}
    action [, ... ];
```

其中action统指如下分区维护子语法。当存在多个分区维护子句时，保证了分区的连续性，无论这些子句的排序如何，GaussDB总会先执行DROP PARTITION再执行ADD PARTITION操作，最后顺序执行其它分区维护操作。

```
{ move_clause |
    exchange_clause |
    row_clause |
    merge_clause |
    modify_clause |
    add_clause    |
    drop_clause   |
    split_clause |
```


版权所有 © 华为云计算技术有限公司

```
truncate_clause |
ilm_clause }
```

- move_clause子语法用于移动分区到新的表空间。

```
MOVE SUBPARTITION { subpartition_name | FOR ( subpartition_value [, ...] ) } TABLESPACE
tablespacename
```

- exchange_clause子语法用于把普通表的数据迁移到指定的分区。

```
EXCHANGE SUBPARTITION { ( subpartition_name ) | FOR ( subpartition_value [, ...] ) }
    WITH TABLE {[ ONLY ] ordinary_table_name | ordinary_table_name * | ONLY
( ordinary_table_name )}
    [ { WITH | WITHOUT } VALIDATION ] [ VERBOSE ] [ UPDATE GLOBAL INDEX ]
```

进行交换的普通表和分区必须满足如下条件：

  - 普通表和分区的列数目相同，对应列的信息严格一致，包括：列名、列的数据类型、列约束、列的Collation信息、列的存储参数、列的压缩信息等。
  - 普通表和分区的表压缩信息严格一致。
  - 普通表索引和分区Local索引个数相同，且对应索引的信息严格一致。
  - 普通表和分区的表约束个数相同，且对应表约束的信息严格一致。
  - 普通表不可以是临时表，分区表只能是二级分区表。
  - 普通表和分区表上不可以有动态数据脱敏，行访问控制约束。

须知

- 完成交换后，普通表和分区的数据被置换，同时普通表和分区的表空间信息被置换。此时，普通表和分区的统计信息变得不可靠，需要对普通表和分区重新执行analyze。
- 由于非分区键不能建立本地唯一索引，只能建立全局唯一索引，所以如果普通表含有唯一索引时，可能会导致不能交换数据。

  如果需要进行数据交换操作可以通过创建中间表的方式，先将分区数据插入到中间表，truncate分区，普通表数据插入分区表，drop普通表，重命名中间表的方式完成数据交换操作。
- 如果在普通表/分区表上进行了drop column操作，被删除的列依然物理存在，所以需要保证普通表和分区的被删除列也严格对齐才能交换成功。
- 当执行交换分区操作的普通表和分区包含rowid列且处于压缩状态时，进行交换分区后，普通表和分区的数据将被自动解压。

- row_clause子语法用于设置分区表的行迁移开关。

```
{ ENABLE | DISABLE } ROW MOVEMENT
```

- merge_clause子语法用于把多个分区合并成一个分区。一个命令中合并的源分区上限为300。

```
MERGE SUBPARTITIONS { subpartition_name } [, ...] INTO SUBPARTITION subpartition_name
    [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n
{ day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM |
HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]
[ TABLESPACE tablespacename ] [ UPDATE GLOBAL INDEX ]
```


版权所有 © 华为云计算技术有限公司

须知

- 对于范围分区，MERGE分区要求源分区的范围连续递增，且MERGE后的分区名可以与最后一个源分区名相同；对于列表分区，则源分区无顺序要求，且MERGE后的分区名可以与任一源分区名相同。如果MERGE后的分区名与源分区名相同，视为同一个分区。
- 未打开guc参数enable_ilm的情况下，如果使用merge_clause子语法把多个带有ilm policy的分区合并成一个分区，新分区不继承ilm policy。
- USTORE存储引擎表不支持在事务块/存储过程中执行ALTER TABLE MERGE SUBPARTITIONS的操作。

- modify_clause子语法用于设置分区索引是否可用。语法可以作用在一级分区上。

```
MODIFY PARTITION partition_name { UNUSABLE LOCAL INDEXES | REBUILD UNUSABLE LOCAL INDEXES }
```

  也可以作用在二级分区上。

```
MODIFY SUBPARTITION subpartition_name { UNUSABLE LOCAL INDEXES | REBUILD UNUSABLE LOCAL INDEXES }
```

- add_clause子语法用于为指定的分区表添加一个或多个分区。语法可以作用在一级分区上。

```
ADD {partition_less_than_item | partition_list_item } [ ( subpartition_definition_list ) ]
```

  也可以作用在二级分区上。

```
MODIFY PARTITION partition_name ADD subpartition_definition
```

  其中，分区项partition_less_than_item为RANGE分区定义语法，具体语法如下。

```
PARTITION partition_name VALUES LESS THAN ( partition_value | MAXVALUE )[ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ]
```

  分区项partition_list_item为LIST分区定义语法，具体语法如下。

```
PARTITION partition_name VALUES ( partition_value [, ...] | DEFAULT ) [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ]
```

  subpartition_definition_list为1到多个二级分区subpartition_definition对象，subpartition_definition具体语法如下。

```
SUBPARTITION subpartition_name [ VALUES LESS THAN ( partition_value | MAXVALUE ) | VALUES ( partition_value [, ...] | DEFAULT )] [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]  [ TABLESPACE tablespace ]
```

须知

若一级分区为HASH分区，不支持以ADD形式新增一级分区；若二级分区为HASH分区，不支持以MODIFY形式新增二级分区。

- drop_clause子语法用于删除分区表中的指定分区。语法可以作用在一级分区上。

```
DROP PARTITION  { partition_name | FOR (  partition_value )  } [ UPDATE GLOBAL INDEX ]
```

  也可以作用在二级分区上。

版权所有 © 华为云计算技术有限公司

```
DROP SUBPARTITION { subpartition_name | FOR ( partition_value, subpartition_value ) }
[ UPDATE GLOBAL INDEX ]
```

须知

- 若一级分区为HASH分区，不支持删除一级分区；若二级分区为HASH分区，不支持删除二级分区。
- 不支持删除唯一子分区。

– split_clause子语法用于把一个分区切割成多个分区。

```
SPLIT SUBPARTITION { subpartition_name| FOR ( subpartition_value [, ...] ) } { split_point_clause
| no_split_point_clause } [ UPDATE GLOBAL INDEX ]
```

须知

- SPLIT后的分区名可以与源分区名相同，但视为不同的分区。
- 未打开guc参数enable_ilm的情况下，如果使用split_clause子语法把一个带有ilm policy的分区分割成多个分区，新分区不继承ilm policy。

■ 范围分区指定切割点split_point_clause的语法为：

```
AT ( subpartition_value ) INTO ( SUBPARTITION subpartition_name [ ILM ADD POLICY
ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month |
year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT
AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE
tablespacename ] , SUBPARTITION subpartition_name [ ILM ADD POLICY ROW STORE
COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO
MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour |
day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] )
```

须知

切割点的大小要位于正在被切割的分区的分区键范围内，指定切割点的方式只能把一个分区切割成两个新分区。

■ 范围分区不指定切割点no_split_point_clause 的语法如下：

```
INTO ( SUBPARTITION subpartition_name VALUES LESS THAN (subpartition_value) [ ILM
ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n
{ day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM |
HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]
[ TABLESPACE tablespacename ][, ...] )
```

须知

- 第一个新分区的分区范围定义要大于正在被切割的分区的前一个分区（如果存在的话）的分区范围定义。
- 其中最后一个分区不能写分区范围定义，即VALUES LESS THAN (subpartition_value)部分，默认继承源分区范围定义的上界值。
- 新分区必须满足分区范围定义递增的约束。

版权所有 © 华为云计算技术有限公司

- 列表范围分区指定切割点split_point_clause的语法如下：

```
VALUES ( subpartition_value_list ) INTO ( SUBPARTITION subpartition_name [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] , SUBPARTITION subpartition_name [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ] )
```

须知

切割点必须是源分区的一个非空真子集，指定切割点的方式只能把一个分区切割成两个新分区。

- 列表分区表不指定切割点no_split_point_clause的语法如下：

```
INTO ( SUBPARTITION subpartition_name VALUES (subpartition_value_list) [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ TABLESPACE tablespacename ][, ...] )
```

须知

- 其中最后一个分区不能写分区范围定义，即VALUES (subpartition_value_list)部分，其范围等于源分区去掉其他子分区后的剩余集合。
- 不指定切割点的方式，每一个新分区都必须是源分区的一个非空真子集，且互不交叉。

– truncate_clause子语法用于清空分区表中的指定分区。语法可以作用在一级分区上。

```
TRUNCATE PARTITION { partition_name | FOR ( partition_value [, ...] ) } [ UPDATE GLOBAL INDEX ]
```

也可以作用在二级分区上。

```
TRUNCATE SUBPARTITION { subpartition_name | FOR ( subpartition_value [, ...] ) } [ UPDATE GLOBAL INDEX ]
```

– ilm_clause子语法用于为分区添加ILM策略，为数据生命周期管理-OLTP表压缩特性支持语法。语法可以作用在一级分区上。

```
MODIFY PARTITION partition_name ILM ADD POLICY ROW STORE COMPRESS ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR ) ]
```

也可以作用在二级分区上

```
MODIFY SUBPARTITION subpartition_name ILM ADD POLICY ROW STORE COMPRESS ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR ) ]
```

- 修改表分区名称的语法。可以修改分区表的一级分区。

```
ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name )}
    RENAME PARTITION { partition_name | FOR ( partition_value [, ...] ) } TO partition_new_name;
```

也可以修改分区表的二级分区。

```
ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name )}
    RENAME SUBPARTITION { subpartition_name | FOR ( subpartition_value [, ...] ) } TO subpartition_new_name;
```


版权所有 © 华为云计算技术有限公司

- 重置分区ID的语法。
```
ALTER TABLE [ IF EXISTS ] { table_name [*] | ONLY table_name | ONLY ( table_name )} RESET PARTITION;
```

参数说明

- **table_name**

  分区表名。

  取值范围：已存在的分区表名。

- **subpartition_name**

  二级分区名。

  取值范围：已存在的二级分区名。

- **tablespacename**

  指定分区要移动到哪一个表空间。

  取值范围：已存在的表空间名。

- **partition_value**

  一级分区键值。

  通过PARTITION FOR (partition_value [, ...])子句指定的这一组值，可以唯一确定一个一级分区。

  取值范围：需要进行操作的一级分区的分区键的取值范围。

- **subpartition_value**

  一级分区键值和二级分区键值。

  通过SUBPARTITION FOR (subpartition_value [, ...])子句指定的这一组值，可以唯一确定一个二级分区。

  取值范围：对于需要进行操作的二级分区，需要同时有其一级分区分区键和二级分区分区键的取值范围。

- **UNUSABLE LOCAL INDEXES**

  设置该分区上的所有索引不可用。

- **REBUILD UNUSABLE LOCAL INDEXES**

  重建该分区上的所有索引。

- **{ ENABLE | DISABLE } ROW MOVEMENT**

  行迁移开关。

  如果进行UPDATE操作时，更新了元组在分区键上的值，造成了该元组所在分区发生变化，就会根据该开关给出报错信息，或者进行元组在分区间的转移。

  取值范围：

  - ENABLE：打开行迁移开关。
  - DISABLE：关闭行迁移开关。

  默认是关闭状态。

- **ordinary_table_name**

  进行迁移的普通表的名称。

  取值范围：已存在的普通表名。

- **{ WITH | WITHOUT } VALIDATION**

  在进行数据迁移时，是否检查普通表中的数据满足指定分区的分区键范围。

版权所有 © 华为云计算技术有限公司

取值范围：

- WITH：对于普通表中的数据要检查是否满足分区的分区键范围，如果有数据不满足，则报错。
- WITHOUT：对于普通表中的数据不检查是否满足分区的分区键范围。

默认是WITH状态。

由于检查比较耗时，特别是当数据量很大的情况。所以在保证当前普通表中的数据满足分区的分区键范围时，可以加上WITHOUT来指明不进行检查。

- **VERBOSE**

  在VALIDATION是WITH状态时，如果检查出普通表有不满足要交换分区的分区键范围的数据，那么把这些数据插入到正确的分区，如果路由不到任何分区，再报错。

  须知

  只有在VALIDATION是WITH状态时，才可以指定VERBOSE。

- **partition_new_name**

  分区的新名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **subpartition_new_name**

  二级分区的新名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **UPDATE GLOBAL INDEX**

  如果使用该参数，则会更新分区表上的所有全局索引，以确保使用全局索引可以查询出正确的数据；如果不使用该参数，则分区表上的所有全局索引将会失效。

示例

- **重命名分区**。

```
--创建二级分区表tbl_rge_lst_test。
gaussdb=# CREATE TABLE tbl_lst_reg_test(
    area_id char(5),
    sdate char(8),
    eid char(5),
    sales_amt int
) PARTITION BY LIST(area_id) SUBPARTITION BY RANGE(sdate)(
    PARTITION p_1001 VALUES ('1001')(
        SUBPARTITION p_1001_201901 VALUES LESS THAN ('20190201'),
        SUBPARTITION p_1001_201902 VALUES LESS THAN ('20190301'),
        SUBPARTITION p_1001_201903 VALUES LESS THAN ('20190401')
    ),
    PARTITION p_1002 VALUES ('1002')(
        SUBPARTITION p_1002_201901 VALUES LESS THAN ('20190201'),
        SUBPARTITION p_1002_201902 VALUES LESS THAN ('20190301'),
        SUBPARTITION p_100w VALUES LESS THAN ('20190401')
    )
);

--将子分区p_100w名称修改为p_1002_201903。
gaussdb=# ALTER TABLE tbl_lst_reg_test RENAME SUBPARTITION p_100w TO p_1002_201903;

--查询。
gaussdb=# SELECT table_name,partition_name,subpartition_name FROM db_tab_subpartitions
WHERE table_name = 'tbl_lst_reg_test' AND partition_name = 'p_1002';
```


版权所有 © 华为云计算技术有限公司

```
   table_name    | partition_name | subpartition_name
-----------------+----------------+-------------------
 tbl_lst_reg_test | p_1002         | p_1002_201901
 tbl_lst_reg_test | p_1002         | p_1002_201902
 tbl_lst_reg_test | p_1002         | p_1002_201903
(3 rows)
```

- **移动分区表空间**。

```
--创建表空间tbs_data1。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace/tbs_data1';

--移动二级分区p_1002_201901至表空间tbs_data1。
gaussdb=# ALTER TABLE tbl_lst_reg_test MOVE SUBPARTITION p_1002_201901 TABLESPACE
tbs_data1;

--移动二级分区p_1002_201902至表空间tbs_data1。
gaussdb=# ALTER TABLE tbl_lst_reg_test MOVE SUBPARTITION FOR ('1002','20190325') TABLESPACE
tbs_data1;

--查询修改后的二级分区的表空间。
gaussdb=# SELECT subpartition_name,tablespace_name FROM db_tab_subpartitions WHERE
subpartition_name IN ('p_1002_201901','p_1002_201903');
 subpartition_name | tablespace_name
-------------------+-----------------
 p_1002_201901     | tbs_data1
 p_1002_201903     | tbs_data1
(2 rows)
```

- **分区交换**。

```
--创建普通表并插入数据。
gaussdb=# CREATE TABLE tbl_test(
    area_id char(5),
    sdate char(8),
    eid char(5),
    sales_amt int
);

gaussdb=# INSERT INTO tbl_test VALUES ('1002','20190326','00001',9000);
gaussdb=# INSERT INTO tbl_test VALUES ('1002','20190326','00002',7500);
gaussdb=# INSERT INTO tbl_test VALUES ('1002','20190326','00003',6000);

--在分区表中插入数据。
gaussdb=# INSERT INTO tbl_lst_reg_test VALUES ('1002','20190301','00001',126);

--执行分区交换。
gaussdb=# ALTER TABLE tbl_lst_reg_test EXCHANGE SUBPARTITION (p_1002_201903) WITH TABLE
tbl_test;

--查询数据发现表和分区的数据发生了交换。
gaussdb=# SELECT * FROM tbl_lst_reg_test;
 area_id |  sdate   |  eid  | sales_amt
---------+----------+-------+-----------
 1002    | 20190326 | 00001 |      9000
 1002    | 20190326 | 00002 |      7500
 1002    | 20190326 | 00003 |      6000
(3 rows)

gaussdb=# SELECT * FROM tbl_test;
 area_id |  sdate   |  eid  | sales_amt
---------+----------+-------+-----------
 1002    | 20190301 | 00001 |       126
(1 row)

--查看分区表空间也发生了交换。
gaussdb=# SELECT subpartition_name,tablespace_name FROM db_tab_subpartitions WHERE
subpartition_name = 'p_1002_201903';
 subpartition_name |  tablespace_name
-------------------+--------------------
 p_1002_201903     | DEFAULT TABLESPACE
(1 row)
```


版权所有 © 华为云计算技术有限公司

- **分区合并**。

```
--分区合并。
gaussdb=# ALTER TABLE tbl_lst_reg_test MERGE SUBPARTITIONS
p_1002_201901,p_1002_201902,p_1002_201903 INTO SUBPARTITION p_1002_20191;

--查询二级分区信息。
gaussdb=# SELECT table_name,partition_name,subpartition_name FROM db_tab_subpartitions
WHERE table_name = 'tbl_lst_reg_test';
    table_name    | partition_name | subpartition_name
------------------+----------------+-------------------
 tbl_lst_reg_test | p_1001         | p_1001_201901
 tbl_lst_reg_test | p_1001         | p_1001_201902
 tbl_lst_reg_test | p_1001         | p_1001_201903
 tbl_lst_reg_test | p_1002         | p_1002_20191
(4 rows)
```

- **添加分区**。

```
--添加一个一级分区。
gaussdb=# ALTER TABLE tbl_lst_reg_test ADD PARTITION p_1003 VALUES('1003') (SUBPARTITION
p_1003_201901 VALUES LESS THAN ('20190201'));

--在特定的一级分区中添加二级分区。
gaussdb=# ALTER TABLE tbl_lst_reg_test MODIFY PARTITION p_1003 ADD SUBPARTITION
p_1003_201902 VALUES LESS THAN ('20190301');

--查询分区信息。
gaussdb=# SELECT table_name,partition_name,subpartition_name FROM db_tab_subpartitions
WHERE table_name = 'tbl_lst_reg_test';
    table_name    | partition_name | subpartition_name
------------------+----------------+-------------------
 tbl_lst_reg_test | p_1001         | p_1001_201901
 tbl_lst_reg_test | p_1001         | p_1001_201902
 tbl_lst_reg_test | p_1001         | p_1001_201903
 tbl_lst_reg_test | p_1002         | p_1002_20191
 tbl_lst_reg_test | p_1003         | p_1003_201901
 tbl_lst_reg_test | p_1003         | p_1003_201902
(6 rows)
```

- **删除分区**。

```
--删除二级分区p_1003_201902。
gaussdb=# ALTER TABLE tbl_lst_reg_test DROP SUBPARTITION p_1003_201902;

--查看。
gaussdb=# SELECT table_name,partition_name,subpartition_name FROM db_tab_subpartitions
WHERE table_name = 'tbl_lst_reg_test' AND partition_name = 'p_1003';
    table_name    | partition_name | subpartition_name
------------------+----------------+-------------------
 tbl_lst_reg_test | p_1003         | p_1003_201901
(1 row)

--删除一级分区p_1003。
gaussdb=# ALTER TABLE tbl_lst_reg_test DROP PARTITION p_1003;

--查看。
gaussdb=# SELECT table_name,partition_name,subpartition_name FROM db_tab_subpartitions
WHERE table_name = 'tbl_lst_reg_test';
    table_name    | partition_name | subpartition_name
------------------+----------------+-------------------
 tbl_lst_reg_test | p_1001         | p_1001_201901
 tbl_lst_reg_test | p_1001         | p_1001_201902
 tbl_lst_reg_test | p_1001         | p_1001_201903
 tbl_lst_reg_test | p_1002         | p_1002_20191
(4 rows)
```

- **切割分区**。

```
--指定切割点切割分区。
gaussdb=# ALTER TABLE tbl_lst_reg_test SPLIT SUBPARTITION p_1002_20191
    AT ('20190201') INTO (SUBPARTITION p_1002_201901,SUBPARTITION p_1002_20191) UPDATE
GLOBAL INDEX;
```


版权所有 © 华为云计算技术有限公司

```
--不指定切割点切割分区。
gaussdb=# ALTER TABLE tbl_lst_reg_test SPLIT SUBPARTITION p_1002_20191 INTO (
    SUBPARTITION p_1002_201902 VALUES LESS THAN ('20190301'),
    SUBPARTITION p_1002_201903
) UPDATE GLOBAL INDEX;

--查询分区信息。
gaussdb=# SELECT table_name,partition_name,subpartition_name,high_value FROM
db_tab_subpartitions;
    table_name    | partition_name | subpartition_name | high_value
------------------+----------------+-------------------+------------
 tbl_lst_reg_test | p_1001         | p_1001_201901     | 20190201
 tbl_lst_reg_test | p_1001         | p_1001_201902     | 20190301
 tbl_lst_reg_test | p_1001         | p_1001_201903     | 20190401
 tbl_lst_reg_test | p_1002         | p_1002_201901     | 20190201
 tbl_lst_reg_test | p_1002         | p_1002_201902     | 20190301
 tbl_lst_reg_test | p_1002         | p_1002_201903     | 20190401
(6 rows)
```

- **清空分区数据**。

```
--清空一级分区。
gaussdb=# ALTER TABLE tbl_lst_reg_test TRUNCATE PARTITION p_1001 UPDATE GLOBAL INDEX;

--清空二级分区。
gaussdb=# ALTER TABLE tbl_lst_reg_test TRUNCATE SUBPARTITION p_1002_201903 UPDATE GLOBAL
INDEX;

--删除表。
gaussdb=# DROP TABLE tbl_lst_reg_test;
gaussdb=# DROP TABLE tbl_test;

--删除表空间。
gaussdb=# DROP TABLESPACE tbs_data1;
```

相关链接

CREATE TABLE SUBPARTITION

1.13.7.37 ALTER TABLESPACE

功能描述

修改表空间的属性。

注意事项

- 只有表空间的所有者或者被赋予了表空间ALTER权限的用户有权限执行ALTER TABLESPACE命令，系统管理员默认拥有此权限。但要修改表空间的所有者，当前用户必须是该表空间的所有者或系统管理员，且该用户是new_owner角色的成员。
- 对行存表的ALTER TABLESPACE操作不支持在事务块中执行。
- 修改系统表空间所有者时，仅允许初始用户将所有者修改为自己，其他情况均不允许修改。
- 要修改表空间的所有者A为B，则A必须是B的直接或者间接成员。

  说明

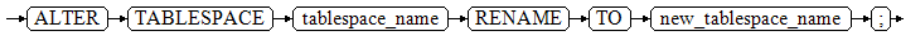
  如果new_owner与old_owner一致，此处不再校验当前执行操作的用户是否具有修改权限，而直接显示ALTER成功。

版权所有 © 华为云计算技术有限公司

语法格式

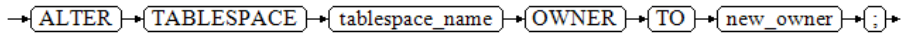
- 重命名表空间的语法。

```
ALTER TABLESPACE tablespace_name
    RENAME TO new_tablespace_name;
```



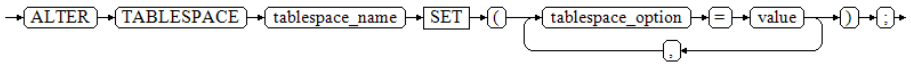
- 设置表空间所有者的语法。

```
ALTER TABLESPACE tablespace_name
    OWNER TO new_owner;
```



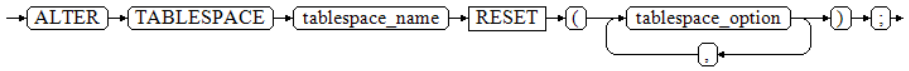
- 设置表空间属性的语法。

```
ALTER TABLESPACE tablespace_name
    SET ( { tablespace_option = value } [, ... ] );
```



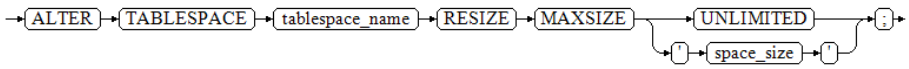
- 重置表空间属性的语法。

```
ALTER TABLESPACE tablespace_name
    RESET ( { tablespace_option } [, ... ] );
```



- 设置表空间限额的语法。

```
ALTER TABLESPACE tablespace_name
    RESIZE MAXSIZE { UNLIMITED | 'space_size'};
```



参数说明

- **tablespace_name**

  要修改的表空间。

  取值范围：已存在的表空间名。

- **new_tablespace_name**

  表空间的新名称，新名称不能以"PG_"开头。

  取值范围：字符串，符合**标识符命名规范**。

- **new_owner**

  表空间的新所有者。

  取值范围：已存在的用户名。

- **tablespace_option**

  设置或者重置表空间的参数。

  取值范围：

  - seq_page_cost：设置优化器计算一次顺序获取磁盘页面的开销。缺省为1.0。
  - random_page_cost：设置优化器计算一次非顺序获取磁盘页面的开销。缺省为4.0。

版权所有 © 华为云计算技术有限公司

说明

- random_page_cost是相对于seq_page_cost的取值，等于或者小于seq_page_cost时毫无意义。
- 默认值为4.0的前提条件是，优化器采用索引来扫描表数据，并且表数据在cache中命中率可以90%左右。
- 如果表数据空间要比物理内存小，那么减小该值到一个适当水平；相反地，如果表数据在cache中命中率要低于90%，那么适当增大该值。
- 如果采用了类似于SSD的随机访问代价较小的存储器，可以适当减小该值，以反映真正的随机扫描代价。

value的取值范围：浮点类型的正数。

- **RESIZE MAXSIZE**

  重新设置表空间限额的数值。

  取值范围：

  - UNLIMITED，该表空间不设置限额。
  - 由space_size来确定，其格式请参见**CREATE TABLESPACE**。

  说明

  - 若调整后的限额值比当前表空间实际使用的值要小，调整操作可以执行成功，后续用户需要将该表空间的使用值降低到新限额值之下，才能继续往该表空间中写入数据。

示例

- **重命名表空间。**

```
--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace1/tbs_data1';

--重命名表空间。
gaussdb=# ALTER TABLESPACE tbs_data1 RENAME TO tbs_data2;

--查询。
gaussdb=# \db tbs_data2
            List of tablespaces
   Name    | Owner |       Location
-----------+-------+-----------------------
 tbs_data2 | omm   | tablespace1/tbs_data1
```

- **设置表空间所有者。**

  示例：

```
--创建用户。
gaussdb=# CREATE USER test PASSWORD '********';

--修改表空间所有者。
gaussdb=# ALTER TABLESPACE tbs_data2 OWNER TO test;

--查看。
gaussdb=# \db tbs_data2
            List of tablespaces
   Name    | Owner |       Location
-----------+-------+-----------------------
 tbs_data2 | test  | tablespace1/tbs_data1
(1 row)
```

- **设置表空间属性。**

  示例：

```
--修改seq_page_cost的值。
gaussdb=# ALTER TABLESPACE tbs_data2 SET (seq_page_cost = 10);
```


版权所有 © 华为云计算技术有限公司

```
--查看。
gaussdb=# SELECT * FROM pg_tablespace WHERE spcname = 'tbs_data2';
  spcname  | spcowner | spcacl |     spcoptions      | spcmaxsize | relative
-----------+----------+--------+---------------------+------------+----------
 tbs_data2 |    16778 |        | {seq_page_cost=10} |            | t
(1 row)
```

- **重置表空间属性。**

```
--将seq_page_cost参数的值重置为缺省值。
gaussdb=# ALTER TABLESPACE tbs_data2 RESET (seq_page_cost);

--查看。
gaussdb=# SELECT * FROM pg_tablespace WHERE spcname = 'tbs_data2';
  spcname  | spcowner | spcacl | spcoptions | spcmaxsize | relative
-----------+----------+--------+------------+------------+----------
 tbs_data2 |    16778 |        |            |            | t
(1 row)
```

- **设置表空间限额。**

  示例：

```
--设置表空间最大使用的空间。
gaussdb=# ALTER TABLESPACE tbs_data2 RESIZE MAXSIZE '10G';

--查看。
gaussdb=# SELECT * FROM pg_tablespace WHERE spcname = 'tbs_data2';
  spcname  | spcowner | spcacl | spcoptions | spcmaxsize | relative
-----------+----------+--------+------------+------------+----------
 tbs_data2 |    16778 |        |            | 10485760 K | t
(1 row)

--删除表空间。
gaussdb=# DROP TABLESPACE tbs_data2;

--删除用户。
gaussdb=# DROP USER test;
```

相关链接

CREATE TABLESPACE，DROP TABLESPACE

1.13.7.38 ALTER TEXT SEARCH CONFIGURATION

须知

该功能为内部使用功能，不建议用户使用。

功能描述

ALTER TEXT SEARCH CONFIGURATION语句用于更改文本搜索配置的定义。用户可以将映射从字符串类型调整为字典，或者改变配置的名称、所有者，或者修改搜索配置的配置参数。

注意事项

- 当一个搜索配置已经被引用（如被用来创建索引），则不允许用户修改此文本搜索配置。
- 要使用ALTER TEXT SEARCH CONFIGURATION，用户必须是配置的所有者。

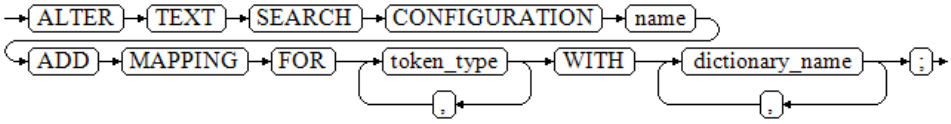
版权所有 © 华为云计算技术有限公司

- ADD MAPPING FOR选项为文本搜索配置增加字符串类型映射；如果ADD MAPPING FOR后面任何一个字符串类型的映射已经存在于此文本搜索配置中，那么系统将会报错。
- ALTER MAPPING FOR选项会首先清除已有的字符串类型映射，然后添加指定的字符串类型映射。
- ALTER MAPPING REPLACE ... WITH ... 与ALTER MAPPING FOR ... REPLACE ... WITH ...选项会直接使用new_dictionary替换old_dictionary。需要注意的是，只有pg_ts_config_map系统表中存在maptokentype与old_dictionary对应关系的元组时，才能更新成功，否则不会成功，也不会有任何提示信息返回。
- DROP MAPPING FOR选项会删除当前文本搜索配置中指定的字符串类型映射。如果没有指定IF EXISTS选项，当DROP MAPPING FOR选项指定的字符串类型映射在文本搜索配置中不存在时，数据库会报错。
- 在使用全文检索功能时，文本检索函数的返回结果会受到文本搜索配置的直接影响，当对文本搜索配置进行修改后，由于已缓存的查询计划不会立即更新，再次调用函数进行查询时可能会导致函数返回结果未同步更新，请进行清空计划缓存操作。如，相关函数存在于参数化绑定执行（PBE）场景或启用自动参数化等含有计划缓存的场景。

语法格式

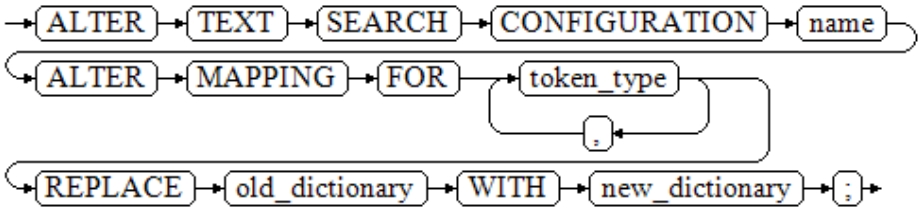
- 增加文本搜索配置字符串类型映射语法。

```
ALTER TEXT SEARCH CONFIGURATION name
    ADD MAPPING FOR { token_type } [, ... ] WITH { dictionary_name } [, ... ];
```



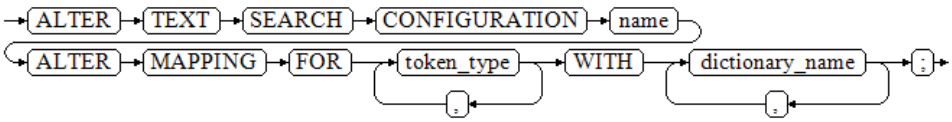
- 修改文本搜索配置字典语法。

```
ALTER TEXT SEARCH CONFIGURATION name
    ALTER MAPPING FOR { token_type } [, ... ] REPLACE old_dictionary WITH new_dictionary;
```



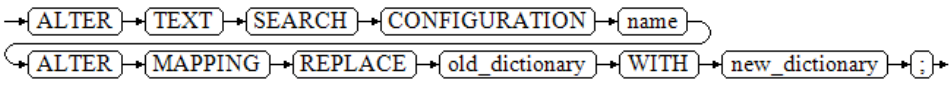
- 修改文本搜索配置字符串类型语法。

```
ALTER TEXT SEARCH CONFIGURATION name
    ALTER MAPPING FOR { token_type } [, ... ] WITH { dictionary_name } [, ... ];
```



- 更改文本搜索配置字典语法。

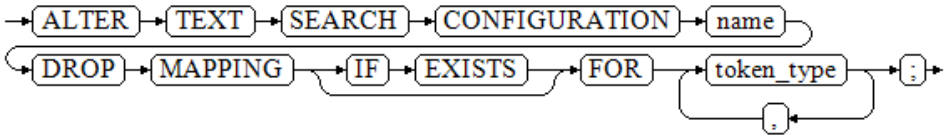
```
ALTER TEXT SEARCH CONFIGURATION name
    ALTER MAPPING REPLACE old_dictionary WITH new_dictionary;
```



- 删除文本搜索配置字符串类型映射语法。

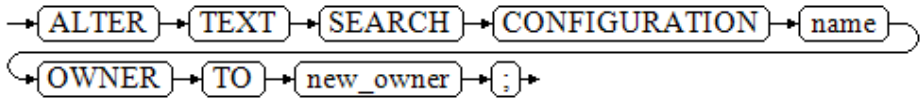
版权所有 © 华为云计算技术有限公司

```
ALTER TEXT SEARCH CONFIGURATION name
    DROP MAPPING [ IF EXISTS ] FOR { token_type } [, ... ];
```



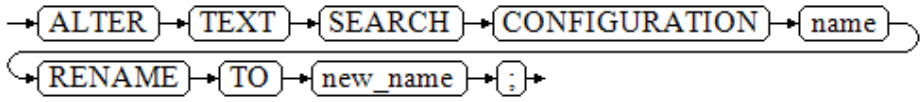
- 重命名文本搜索配置所有者语法。

```
ALTER TEXT SEARCH CONFIGURATION name OWNER TO new_owner;
```



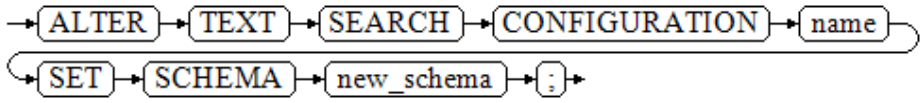
- 重命名文本搜索配置名称语法。

```
ALTER TEXT SEARCH CONFIGURATION name RENAME TO new_name;
```



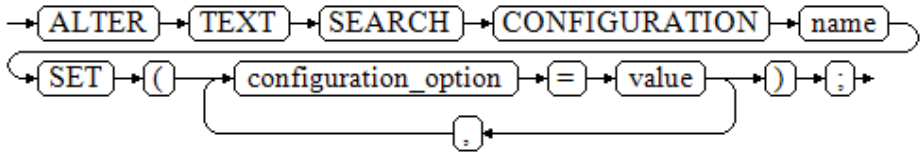
- 重命名文本搜索配置命名空间语法。

```
ALTER TEXT SEARCH CONFIGURATION name SET SCHEMA new_schema;
```



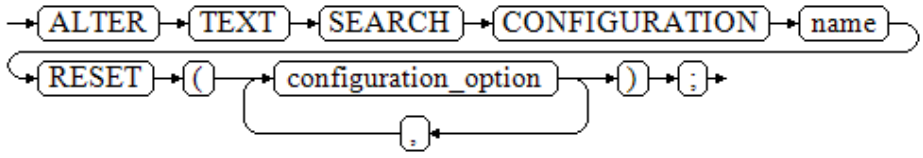
- 修改文本搜索配置属性语法。

```
ALTER TEXT SEARCH CONFIGURATION name SET ( { configuration_option = value } [, ...] );
```



- 重置文本搜索配置属性语法。

```
ALTER TEXT SEARCH CONFIGURATION name RESET ( {configuration_option} [, ...] );
```



参数说明

- **name**

  已有文本搜索配置的名称（可以有模式修饰）。

- **token_type**

  与配置的语法解析器关联的字符串类型的名称。

- **dictionary_name**

  文本搜索字典名称。 如果有多个字典，则它们会按指定的顺序搜索。

- **old_dictionary**

  映射中拟被替换的文本搜索字典名称。

版权所有 © 华为云计算技术有限公司

- **new_dictionary**
  替换old_dictionary的文本搜索字典的名称。
- **new_owner**
  文本搜索配置的新所有者。
- **new_name**
  文本搜索配置的新名称。
- **new_schema**
  文本搜索配置的新模式名。
- **configuration_option**
  文本搜索配置项。
- **value**
  文本搜索配置项的值。

示例

```
--创建文本搜索配置english_1。
gaussdb=# CREATE TEXT SEARCH CONFIGURATION english_1 (parser=default);
CREATE TEXT SEARCH CONFIGURATION

--创建文本搜索配置english_2。
gaussdb=# CREATE TEXT SEARCH CONFIGURATION english_2 (parser=default);
CREATE TEXT SEARCH CONFIGURATION

--增加文本搜索配置字符串类型映射语法。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION english_1 ADD MAPPING FOR word WITH
simple,english_stem;
ALTER TEXT SEARCH CONFIGURATION

gaussdb=# ALTER TEXT SEARCH CONFIGURATION english_1 ADD MAPPING FOR email WITH english_stem,
french_stem;
ALTER TEXT SEARCH CONFIGURATION

--查询文本搜索配置相关信息。
gaussdb=# SELECT b.cfgname,a.maptokentype,a.mapseqno,a.mapdict,c.dictname FROM pg_ts_config_map
a,pg_ts_config b, pg_ts_dict c WHERE a.mapcfg=b.oid AND a.mapdict=c.oid AND b.cfgname='english_1'
ORDER BY 1,2,3,4,5;
  cfgname  | maptokentype | mapseqno | mapdict |    dictname
-----------+--------------+----------+---------+--------------
 english_1 |            2 |        1 |    3765 | simple
 english_1 |            2 |        2 |   12960 | english_stem
 english_1 |            4 |        1 |   12960 | english_stem
 english_1 |            4 |        2 |   12964 | french_stem
(4 rows)

--修改文本搜索配置字符串类型映射语法。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION english_1 ALTER MAPPING REPLACE french_stem with
german_stem;
ALTER TEXT SEARCH CONFIGURATION

--查询文本搜索配置相关信息。
gaussdb=# SELECT b.cfgname,a.maptokentype,a.mapseqno,a.mapdict,c.dictname FROM pg_ts_config_map
a,pg_ts_config b, pg_ts_dict c WHERE a.mapcfg=b.oid AND a.mapdict=c.oid AND b.cfgname='english_1'
ORDER BY 1,2,3,4,5;
  cfgname  | maptokentype | mapseqno | mapdict |    dictname
-----------+--------------+----------+---------+--------------
 english_1 |            2 |        1 |    3765 | simple
 english_1 |            2 |        2 |   12960 | english_stem
 english_1 |            4 |        1 |   12960 | english_stem
 english_1 |            4 |        2 |   12966 | german_stem
(4 rows)
```


版权所有 © 华为云计算技术有限公司

```
--将文本搜索配置名称english_2重命名为english_3。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION english_2 RENAME TO english_3;

--创建用户joe。
gaussdb=# CREATE USER joe IDENTIFIED BY '********';

--重命名文本搜索配置所有者为joe。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION english_3 OWNER TO joe;

--重命名文本搜索配置命名空间。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION english_3 set schema joe;

--删除Schema。
gaussdb=# DROP SCHEMA IF EXISTS joe CASCADE;

--删除用户joe
gaussdb=# DROP ROLE IF EXISTS joe;

--删除文本搜索配置。
gaussdb=# DROP TEXT SEARCH CONFIGURATION english_1;
```

1.13.7.39 ALTER TEXT SEARCH DICTIONARY

须知

该功能为内部使用功能，不建议用户使用。

功能描述

ALTER TEXT SEARCH DICTIONARY语句用于修改全文检索词典的相关定义，包括参数、名称、所有者和模式等。

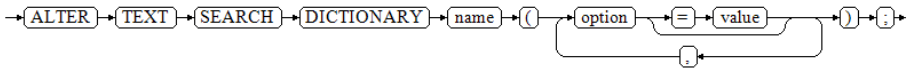
注意事项

- 预定义词典不支持ALTER操作。
- 只有词典的所有者可以执行ALTER操作，当三权分立开关关闭时，系统管理员默认拥有此权限。
- 创建或修改词典之后，任何对于filepath路径下用户自定义的词典定义文件的修改，将不会影响到数据库中的词典。如果需要在数据库中使用这些修改，需使用ALTER TEXT SEARCH DICTIONARY语句更新对应词典的定义文件。
- 在使用全文检索功能时，文本检索函数的返回结果会受到全文检索词典配置的直接影响，当对全文检索词典配置进行修改后，由于已缓存的查询计划不会立即更新，再次调用函数进行查询时可能会导致函数返回结果未同步更新，请进行清空计划缓存操作。如，相关函数存在于参数化绑定执行（PBE）场景或启用自动参数化等含有计划缓存的场景。

语法格式

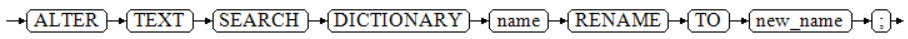
- 修改词典定义。

```
ALTER TEXT SEARCH DICTIONARY name (
    { option [ = value ] } [, ... ]
);
```

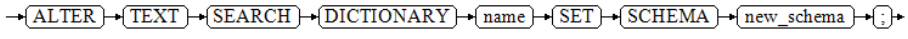


版权所有 © 华为云计算技术有限公司

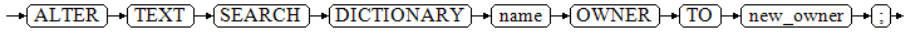
- 重命名词典。
  ALTER TEXT SEARCH DICTIONARY name RENAME TO new_name;

- 设置词典的所属模式。
  ALTER TEXT SEARCH DICTIONARY name SET SCHEMA new_schema;

- 修改词典的所有者。
  ALTER TEXT SEARCH DICTIONARY name OWNER TO new_owner;

参数说明

- **name**

  已存在的词典名（可指定模式名，否则默认在当前模式下）。

  取值范围：已存在的词典名。

- **option**

  要修改的参数名。与template对应，不同的词典类型具有不同的参数列表，且与指定顺序无关。详细介绍请参见**CREATE TEXT SEARCH DICTIONARY**章节的**option**。

  说明

  - 不支持修改词典的TEMPLATE参数值。
  - 不支持仅修改FILEPATH参数而不修改对应的词典定义文件参数。
  - 词典定义文件的文件名仅支持小写字母、数据、下划线混合。

- **value**

  要修改的参数值。如果省略等号（=）和value，则表示删除该option的先前设置，使用默认值。

  取值范围：对应option定义。

- **new_name**

  词典的新名称。

  取值范围：符合标识符命名规范的字符串，且最大长度不超过63个字符。

- **new_owner**

  词典新的所有者。

  取值范围：已存在的用户。

- **new_schema**

  词典的新模式。

  取值范围：已存在的模式。

示例

```
--创建字典my_dict。
gaussdb=# CREATE TEXT SEARCH DICTIONARY my_dict ( TEMPLATE = Simple );

--创建字典my_dict1。
gaussdb=# CREATE TEXT SEARCH DICTIONARY my_dict1 ( TEMPLATE = Simple );
```


版权所有 © 华为云计算技术有限公司

```
--重命名词典my_dict1为my_dict2。
gaussdb=# ALTER TEXT SEARCH DICTIONARY my_dict1 RENAME TO my_dict2;

--更改Simple类型字典，将非停用词设置为已识别，其他参数保持不变。
gaussdb=# ALTER TEXT SEARCH DICTIONARY my_dict ( Accept = true );

--更改Simple类型字典，重置Accept参数。
gaussdb=# ALTER TEXT SEARCH DICTIONARY my_dict ( Accept );

--更新词典定义，不实际更改任何内容。
gaussdb=# ALTER TEXT SEARCH DICTIONARY my_dict ( dummy );

--删除字典。
gaussdb=# DROP TEXT SEARCH DICTIONARY my_dict;
gaussdb=# DROP TEXT SEARCH DICTIONARY my_dict2;
```

1.13.7.40 ALTER TRIGGER

功能描述

ALTER TRIGGER语句用于修改触发器名称及所有者。

说明

目前只支持修改触发器的名称及所有者。

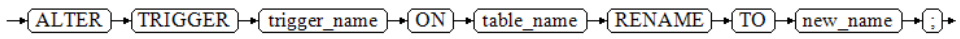
注意事项

- 触发器所在表的所有者或者被授予了ALTER ANY TRIGGER权限的用户可以执行ALTER TRIGGER操作，系统管理员默认拥有此权限。
- 若要修改触发器owner，当前用户必须是该触发器的所有者或者系统管理员，且当前用户必须是新owner所属角色的直接或者间接成员。新owner必须有触发器所在模式上的CREATE权限。三权分立开关关闭时，系统管理员默认拥有该权限。当三权分立开关打开时，系统管理员默认没有该权限。
- 只有初始化用户才能修改触发器的owner为初始化用户。

语法格式

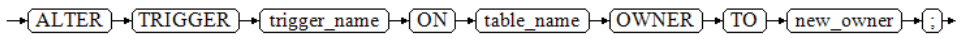
- 修改触发器名称。

```
ALTER TRIGGER trigger_name ON table_name RENAME TO new_name;
```



- 修改触发器owner。

```
ALTER TRIGGER trigger_name ON table_name OWNER TO new_owner;
```



参数说明

- **trigger_name**

  要修改的触发器名称。

  取值范围：已存在的触发器。

- **table_name**

  要修改的触发器所在的表名称。

  取值范围：已存在的含触发器的表。

版权所有 © 华为云计算技术有限公司

- **new_name**
  修改后的新名称，不能与所在表上其他触发器同名。
  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。
- **new_owner**
  修改后的触发器新owner名称。
  取值范围：数据库中存在的用户。

示例

```
--创建源表及触发表。
gaussdb=# CREATE TABLE test_trigger_src_tbl(id1 INT, id2 INT, id3 INT);
gaussdb=# CREATE TABLE test_trigger_des_tbl(id1 INT, id2 INT, id3 INT);

--创建DELETE触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION TRI_DELETE_FUNC() RETURNS TRIGGER AS
        $$
        DECLARE
        BEGIN
                DELETE FROM test_trigger_des_tbl WHERE id1=OLD.id1;
                RETURN OLD;
        END
        $$ LANGUAGE plpgsql;

--创建DELETE触发器。
gaussdb=# CREATE TRIGGER delete_trigger BEFORE DELETE ON test_trigger_src_tbl FOR EACH ROW
EXECUTE PROCEDURE tri_delete_func();

--修改触发器的名称。
gaussdb=# ALTER TRIGGER delete_trigger ON test_trigger_src_tbl RENAME TO delete_trigger_renamed;

--修改触发器owner为jack（需保证jack为数据库中已创建的用户）。
gaussdb=# ALTER TRIGGER delete_trigger ON test_trigger_src_tbl OWNER TO jack;

--删除触发器。
gaussdb=# DROP TRIGGER delete_trigger_renamed ON test_trigger_src_tbl;

--删除函数。
gaussdb=# DROP FUNCTION tri_delete_func;

--删除源表及触发表。
gaussdb=# DROP TABLE test_trigger_src_tbl;
gaussdb=# DROP TABLE test_trigger_des_tbl;
```

相关链接

CREATE TRIGGER，**DROP TRIGGER**，**ALTER TABLE**

1.13.7.41 ALTER TYPE

功能描述

修改一个类型的定义。

注意事项

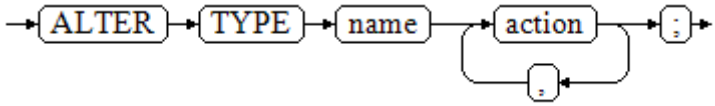
类型的所有者、被授予了类型ALTER权限的用户、或者被授予了ALTER ANY TYPE权限的用户可以执行ALTER TYPE命令，三权分立开关关闭时，系统管理员默认拥有此权限。若要修改类型的所有者或者修改类型的模式，当前用户必须是该类型的所有者或者系统管理员，且该用户是新所有者角色的成员。

版权所有 © 华为云计算技术有限公司

语法格式

- 修改类型。

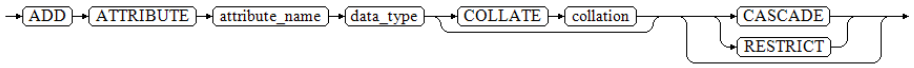
  ```
  ALTER TYPE name action [, ... ];
  ```

  其中action对应的子句如下:

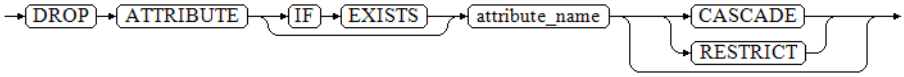
  - 给复合类型增加新的属性。

    ```
    ADD ATTRIBUTE attribute_name data_type [ COLLATE collation ] [ CASCADE | RESTRICT ]
    ```

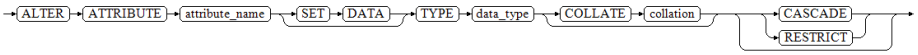
  - 从复合类型中删除一个属性。

    ```
    DROP ATTRIBUTE [ IF EXISTS ] attribute_name [ CASCADE | RESTRICT ]
    ```

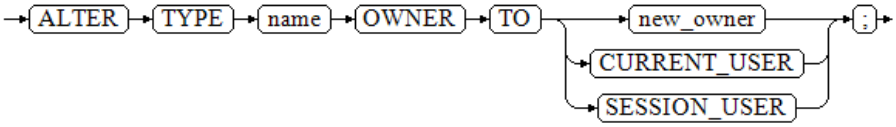
  - 改变一种复合类型中某个属性的类型。

    ```
    ALTER ATTRIBUTE attribute_name [ SET DATA ] TYPE data_type [ COLLATE collation ]
    [ CASCADE | RESTRICT ]
    ```

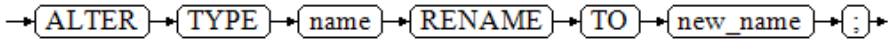
- 改变类型的所有者。

  ```
  ALTER TYPE name OWNER TO { new_owner | CURRENT_USER | SESSION_USER };
  ```

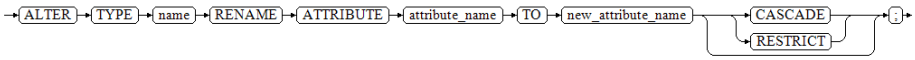
- 改变类型的名称。

  ```
  ALTER TYPE name RENAME TO new_name;
  ```

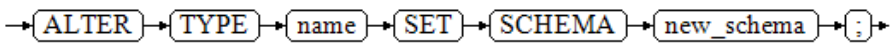
- 改变一个复合类型中一个属性的名称。

  ```
  ALTER TYPE name RENAME ATTRIBUTE attribute_name TO new_attribute_name [ CASCADE |
  RESTRICT ];
  ```

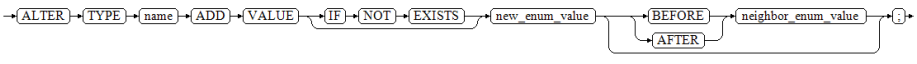
- 将类型移至一个新的模式中。

  ```
  ALTER TYPE name SET SCHEMA new_schema;
  ```

- 为枚举类型增加一个新值。

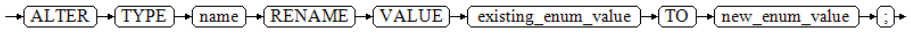
  ```
  ALTER TYPE name ADD VALUE [ IF NOT EXISTS ] new_enum_value [ { BEFORE | AFTER }
  neighbor_enum_value ];
  ```

- 重命名枚举类型的一个标签值。

  ```
  ALTER TYPE name RENAME VALUE existing_enum_value TO new_enum_value;
  ```


版权所有 © 华为云计算技术有限公司



参数说明

- **name**
  一个需要修改的现有类型的名称(可以有模式修饰) 。
- **new_name**
  该类型的新名称。
- **new_owner**
  新所有者的用户名 。
- **new_schema**
  该类型的新模式 。
- **attribute_name**
  拟增加、更改或删除的属性的名称。
- **new_attribute_name**
  拟改名的属性的新名称。
- **data_type**
  拟新增属性的数据类型，或是拟更改的属性的新类型名。
- **new_enum_value**
  枚举类型新增加的标签值，是一个非空的长度不超过63个字节的字符串。
- **neighbor_enum_value**
  一个已有枚举标签值，新值应该增加在紧接着该枚举值之前或者之后的位置上。
- **existing_enum_value**
  现有的要重命名的枚举值，是一个非空的长度不超过63个字节的字符串
- **CASCADE**
  自动级联更新需更新类型以及相关联的记录和继承它们的子表。
- **RESTRICT**
  如果需联动更新类型是已更新类型的关联记录，则拒绝更新，这是缺省选项。

须知

- ADD ATTRIBUTE、DROP ATTRIBUTE和ALTER ATTRIBUTE选项可以组合成一个列表同时处理多个属性更改。 例如，在一条命令中同时增加几个属性或是更改几个属性的类型。
- 要修改一个类型的模式，必须在新模式上拥有CREATE权限。要修改所有者，必须是新的所有角色的直接或间接成员，并且该成员必须在此类型的模式上有CREATE权限（这些限制了修改所有者不会做任何通过删除和重建类型不能做的事情。但是，三权分立关闭时的系统管理员可以以任何方式修改任意类型的所有权）。要增加一个属性或是修改一个属性的类型，也必须有该类型的USAGE权限。

- **CURRENT_USER**
  当前用户。

版权所有 © 华为云计算技术有限公司

- **SESSION_USER**
  当前系统用户。
- **COLLATE collation**
  COLLATE子句为该列（必须是一种可排序数据类型）赋予一个排序规则。 如果没有指定，将使用该列数据类型的默认排序规则。

示例

- 对复合类型的修改。

```
--创建复合类型。
gaussdb=# CREATE TYPE typ_stu AS (name varchar(10),age int);

--创建表并插入数据，其中info字段数据类型为typ_stu。
gaussdb=# CREATE TABLE tbl_test (id int PRIMARY KEY,info typ_stu);
gaussdb=# INSERT INTO tbl_test VALUES (1,('Jim',16));

--查看tbl_test数据。
gaussdb=# SELECT * FROM tbl_test;
 id |   info
----+----------
  1 | (Jim,16)
(1 row)

--为复合类型增加一个新的属性。
gaussdb=# ALTER TYPE typ_stu ADD ATTRIBUTE year int CASCADE;

--再次查看tbl_test数据，info多了一个属性。
gaussdb=# SELECT * FROM tbl_test;
 id |   info
----+-----------
  1 | (Jim,16,)
(1 row)

--从复合类型删除一个属性。
gaussdb=# ALTER TYPE typ_stu DROP ATTRIBUTE year;
```

- 修改类型所有者。

```
--创建用户test。
gaussdb=# CREATE ROLE test PASSWORD '********';

--修改typ_stu的所有者为test。
gaussdb=# ALTER TYPE typ_stu OWNER TO test;

--查询typ_stu拥有者。
gaussdb=# SELECT t1.typname, t2.rolname AS owner
FROM pg_type t1, gs_roles t2
WHERE t1.typname = 'typ_stu' AND
      t1.typowner = t2.oid;
 typname | owner
---------+--------
 typ_stu | test
(1 row)
```

- 修改类型或是一个复合类型中的一个属性的名称。

```
--将typ_stu的其中一个属性age重命名为age1。
gaussdb=# ALTER TYPE typ_stu RENAME ATTRIBUTE age to age1;

--通过如下SQL查询tbl_test表中年龄为16的数据。
gaussdb=# SELECT id,(info).name,(info).age1 FROM tbl_test WHERE (info).age1 = 16;
 id | name | age1
----+------+------
  1 | Jim  |   16
(1 row)
```

- 将类型移至一个新的模式中。

```
--创建模式sctest。
gaussdb=# CREATE SCHEMA sctest;
```


版权所有 © 华为云计算技术有限公司

```
--将typ_stu移至模式sc_test中。
gaussdb=# ALTER TYPE typ_stu SET SCHEMA sctest;

--查询typ_stu的所属模式。
gaussdb=# \dT sctest.*
         List of data types
 Schema |      Name      | Description
--------+----------------+-------------
 sctest | sctest.typ_stu |
(1 row)

--删除表。
gaussdb=# DROP TABLE tbl_test;
gaussdb=# DROP TYPE sctest.typ_stu;
gaussdb=# DROP ROLE test;
gaussdb=# DROP SCHEMA sctest;
```

- 为枚举类型增加一个新值。

```
--创建一个枚举类型typ_bugstatus。
gaussdb=# CREATE TYPE typ_bugstatus AS ENUM ('create', 'modify', 'closed');

--创建表tbl_test1。
gaussdb=# CREATE TABLE tbl_test1 (id serial, bugstat typ_bugstatus);

--插入数据时bugstat字段必须是枚举类型中的值，否则报错。
gaussdb=# INSERT INTO tbl_test1 (bugstat) VALUES ('closed');
gaussdb=# INSERT INTO tbl_test1 (bugstat) VALUES ('deleted');
ERROR:  invalid input value for enum typ_bugstatus: "deleted"
LINE 1: INSERT INTO tbl_test1 (bugstat) VALUES ('deleted');
                                                ^
CONTEXT:  referenced column: bugstat

--查看tbl_test1数据。
gaussdb=# SELECT * FROM tbl_test1;
 id | bugstat
----+---------
  1 | closed
(1 row)

--为枚举类型增加一个标签值。
gaussdb=# ALTER TYPE typ_bugstatus ADD VALUE IF NOT EXISTS 'regress' BEFORE 'closed';

--查询。
gaussdb=# \dT+ typ_bugstatus
                                  List of data types
 Schema |     Name      | Internal name | Size | Elements | Access privileges | Description
--------+---------------+---------------+------+----------+-------------------+-------------
 public | typ_bugstatus | typ_bugstatus | 4    | create  +|                   |
        |               |               |      | modify  +|                   |
        |               |               |      | regress +|                   |
        |               |               |      | closed   |                   |
(1 row)
```

- 重命名枚举类型的一个标签值。

```
--将类型typ_bugstatus中closed修改为close。
gaussdb=# ALTER TYPE typ_bugstatus RENAME VALUE 'closed' TO 'close';

--查看表tbl_test1数据,其中closed的数据也都改成了close。
gaussdb=# SELECT * FROM tbl_test1;
 id | bugstat
----+---------
  1 | close
(1 row)

--查询。
gaussdb=# \dT+ typ_bugstatus;
                                  List of data types
 Schema |     Name      | Internal name | Size | Elements | Access privileges | Description
--------+---------------+---------------+------+----------+-------------------+-------------
```


版权所有 © 华为云计算技术有限公司

```
 public | typ_bugstatus | typ_bugstatus | 4    | create  +|                |
        |              |               |      | modify  +|                |
        |              |               |      | regress +|                |
        |              |               |      | close    |                |
(1 row)

--删除。
gaussdb=# DROP TABLE tbl_test1;
gaussdb=# DROP TYPE typ_bugstatus;
```

相关链接

CREATE TYPE，**DROP TYPE**

1.13.7.42 ALTER USER

功能描述

修改数据库用户的属性。包括数据库用户名称以及权限等。

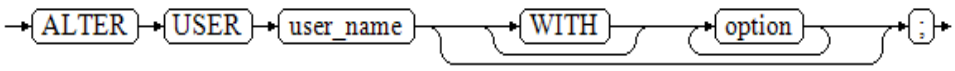
注意事项

ALTER USER中修改的会话参数只针对指定的用户，且在下一次会话中有效。

语法格式

- 修改用户的权限等信息。

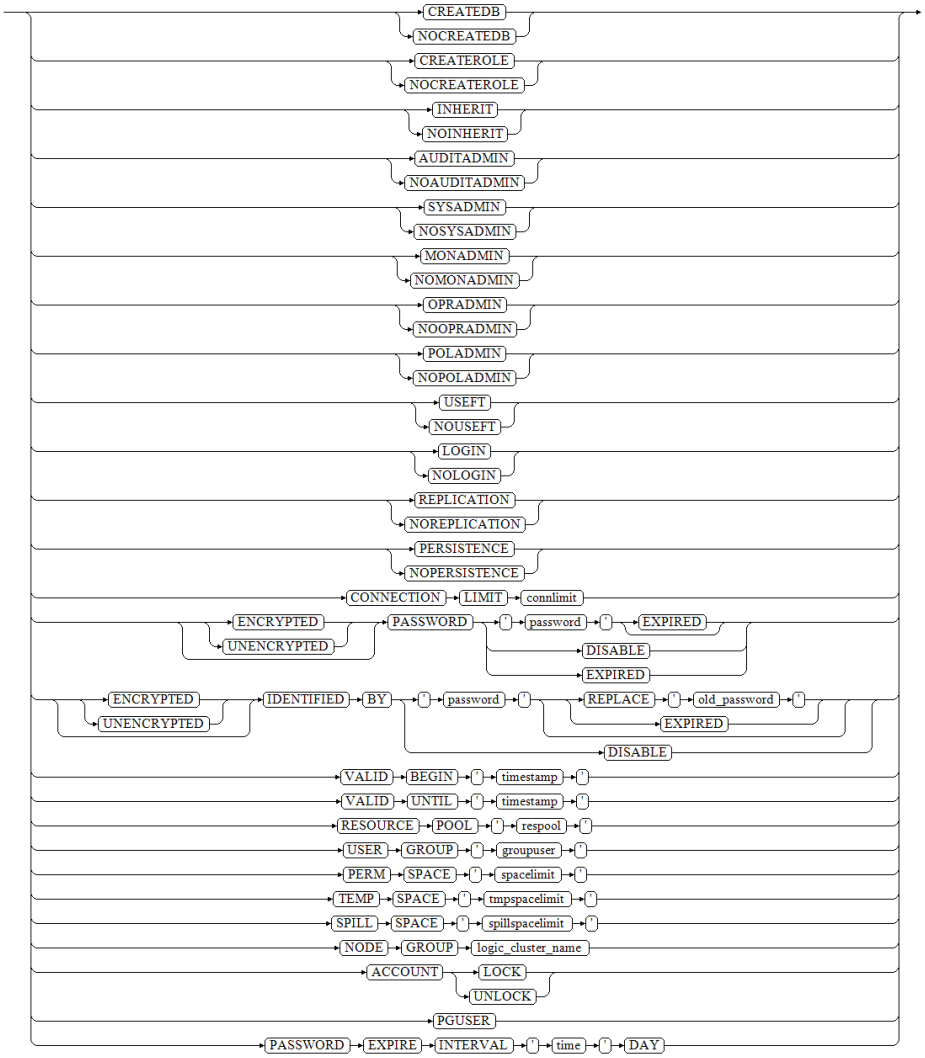
```
ALTER USER user_name [ [ WITH ] option [ ... ] ];
```



其中option子句为：

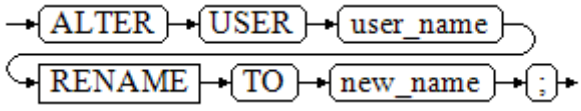
```
{
   { CREATEDB | NOCREATEDB }
   | { CREATEROLE | NOCREATEROLE }
   | { INHERIT | NOINHERIT }
   | { AUDITADMIN | NOAUDITADMIN }
   | { SYSADMIN | NOSYSADMIN }
   | { MONADMIN | NOMONADMIN }
   | { OPRADMIN | NOOPRADMIN }
   | { POLADMIN | NOPOLADMIN }
   | { USEFT | NOUSEFT }
   | { LOGIN | NOLOGIN }
   | { REPLICATION | NOREPLICATION }
   | { PERSISTENCE | NOPERSISTENCE }
   | CONNECTION LIMIT connlimit
   | [ ENCRYPTED | UNENCRYPTED ] PASSWORD { 'password' [EXPIRED] | DISABLE | EXPIRED }
   | [ ENCRYPTED | UNENCRYPTED ] IDENTIFIED BY { 'password' [ REPLACE 'old_password' |
EXPIRED ] | DISABLE }
   | VALID BEGIN 'timestamp'
   | VALID UNTIL 'timestamp'
   | RESOURCE POOL 'respool'
   | USER GROUP 'groupuser'
   | PERM SPACE 'spacelimit'
   | TEMP SPACE 'tmpspacelimit'
   | SPILL SPACE 'spillspacelimit'
   | NODE GROUP logic_cluster_name
   | ACCOUNT { LOCK | UNLOCK }
   | PGUSER
   | PASSWORD EXPIRE INTERVAL 'time' DAY
}
```


版权所有 © 华为云计算技术有限公司



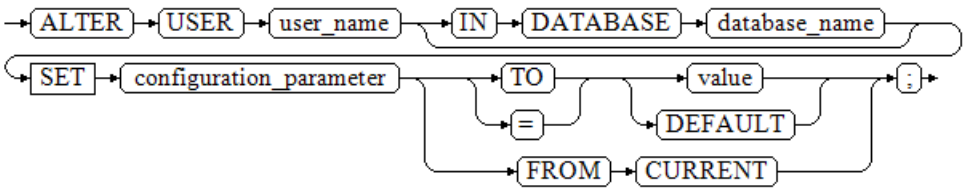
- 修改用户名。

```
ALTER USER user_name
    RENAME TO new_name;
```



- 修改与用户关联的指定会话参数值。当前版本不支持设置用户级别参数。

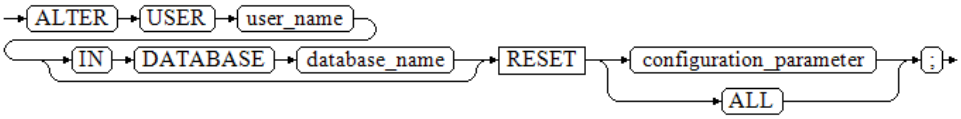
```
ALTER USER user_name [ IN DATABASE database_name ]
    SET configuration_parameter {{ TO | = } { value | DEFAULT }|FROM CURRENT};
```



- 重置与用户关联的指定会话参数值。当前版本不支持设置用户级别参数。

```
ALTER USER user_name
    [ IN DATABASE database_name ] RESET {configuration_parameter|ALL};
```


版权所有 © 华为云计算技术有限公司



参数说明

- **user_name**

  现有用户名。

  取值范围：已存在的用户名，如果用户名中包含大写字母则需要使用双引号进行声明。

  须知

  当前不支持对重分布资源管控对象redisuser进行RENAME TO操作，也不支持将其他用户名进行RENAME TO redisuser操作。

- **new_password**

  新密码。

  取值范围：字符串。

  密码规则如下：

  - 不能与当前密码相同。
  - 密码默认不少于8个字符。
  - 不能与用户名及用户名倒序相同。
  - 至少包含大写字母（A~Z）、小写字母（a~z）、数字（0~9）以及非字母数字字符（限定为~!@#$%^&*()-_=+\|[{}];:,<.>/?）四类字符中的三类字符。当密码中包含其他字符（不属于上述四种字符范围内）时语句执行报错。
  - 用户密码需要使用单引号或双引号进行声明。

- **old_password**

  旧密码。

- **ACCOUNT { LOCK | UNLOCK }**
  - ACCOUNT LOCK：锁定账户，禁止通过密码认证登录数据库。
  - ACCOUNT UNLOCK：解锁账户，允许登录数据库。

- **PGUSER**

  当前版本不允许修改用户的PGUSER属性。

- 其他参数请参见**ALTER ROLE**章节的**参数说明**以及**ALTER ROLE**章节的**参数说明**。

示例

```
--创建用户jim，登录密码为********。
gaussdb=# CREATE USER jim PASSWORD '********';

--创建数据库。
gaussdb=# CREATE DATABASE testdb1;

--修改用户jim的登录密码。
gaussdb=# ALTER USER jim IDENTIFIED BY '**********' REPLACE '********';

--锁定jim账户。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# ALTER USER jim ACCOUNT LOCK;

--解锁jim账户。
gaussdb=# ALTER USER jim ACCOUNT UNLOCK;

--修改用户名。
gaussdb=# ALTER USER jim RENAME TO lisa;

--删除用户。
gaussdb=# DROP USER lisa CASCADE;
```

相关链接

CREATE ROLE，**CREATE USER**，**DROP USER**

1.13.7.43 ALTER VIEW

功能描述

ALTER VIEW更改视图的各种辅助属性（如果用户是更改视图的查询定义，要使用CREATE OR REPLACE VIEW）或重编译视图。

注意事项

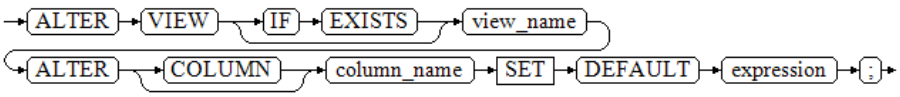
只有视图的所有者或者被授予了视图ALTER权限的用户才可以执行ALTER VIEW命令，三权分立开关关闭时，系统管理员默认拥有该权限。针对所要修改属性的不同，对其还有以下权限约束：

- 修改视图的模式，当前用户必须是视图的所有者或者系统管理员，且要有新模式的CREATE权限。三权分立开关打开时，系统管理员不能修改视图模式。
- 修改视图的所有者，当前用户必须是视图的所有者或者系统管理员，且该用户必须是新所有者角色的成员，并且此角色必须有视图所在模式的CREATE权限。三权分立开关打开时，系统管理员不能修改视图的所有者。

语法格式

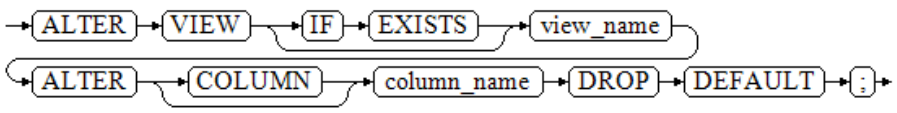
- 设置视图列的默认值。

```
ALTER VIEW [ IF EXISTS ] view_name
    ALTER [ COLUMN ] column_name SET DEFAULT expression;
```



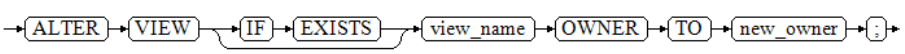
- 取消列视图列的默认值。

```
ALTER VIEW [ IF EXISTS ] view_name
    ALTER [ COLUMN ] column_name DROP DEFAULT;
```



- 修改视图的所有者。

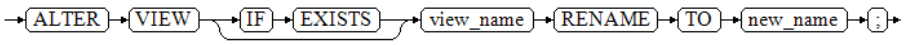
```
ALTER VIEW [ IF EXISTS ] view_name
    OWNER TO new_owner;
```



- 重命名视图。

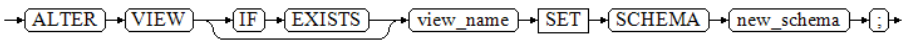
版权所有 © 华为云计算技术有限公司

```
ALTER VIEW [ IF EXISTS ] view_name
    RENAME TO new_name;
```



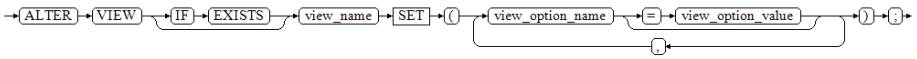
- 设置视图的所属模式。

```
ALTER VIEW [ IF EXISTS ] view_name
    SET SCHEMA new_schema;
```



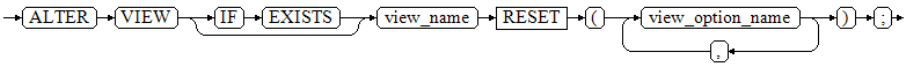
- 设置视图的选项。

```
ALTER VIEW [ IF EXISTS ] view_name
    SET ( { view_option_name [ = view_option_value ] } [, ... ] );
```



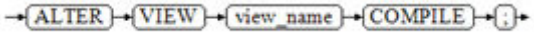
- 重置视图的选项。

```
ALTER VIEW [ IF EXISTS ] view_name
    RESET ( view_option_name [, ... ] );
```



- 重编译视图。

```
ALTER VIEW view_name COMPILE;
```



参数说明

- **IF EXISTS**

  使用这个选项，如果视图不存在时不会产生错误，仅会有一个提示信息。
- **view_name**

  视图名称，可以用模式修饰。

  取值范围：字符串，已经存在的视图名。
- **column_name**

  字段名称。

  取值范围：字符串，已经存在的视图的字段名。
- **SET/DROP DEFAULT**

  设置或删除一个列的缺省值，该参数暂无实际意义。
- **new_owner**

  视图新所有者的用户名称。
- **new_name**

  视图的新名称。
- **new_schema**

  视图的新模式。
- **view_option_name [= view_option_value]**

  该子句为视图指定一个可选的参数。
  - security_barrier：设置视图是否提供行级安全。取值范围为Boolean类型。缺省值为true。

版权所有 © 华为云计算技术有限公司

  - check_option：控制更新视图的行为。支持参数值为CASCADED或LOCAL。该参数取值不可缺省。

- **expression**

  常量、函数或SQL表达式。

- COMPILE

  使用这个选项，对视图进行手动重编译。

  说明

  视图失效重编译请参见《开发指南》中“特性使用指导 > 失效重编译”章节。

示例

- 重命名视图。

```
--创建test_tbl表。
gaussdb=# CREATE TABLE test_tb1(col1 INT,col2 INT);

--创建视图。
gaussdb=# CREATE VIEW abc AS SELECT * FROM test_tb1;

--设置视图列的默认值为100。
gaussdb=# ALTER VIEW IF EXISTS  abc ALTER  COLUMN  col1 SET DEFAULT 100;

--取消视图列的默认值。
gaussdb=# ALTER VIEW IF EXISTS  abc ALTER  COLUMN  col1 DROP DEFAULT;

--重命名视图。
gaussdb=# ALTER VIEW IF EXISTS abc RENAME TO test_v1;

--查看视图。
gaussdb=# \dv
            List of relations
 Schema |  Name   | Type | Owner | Storage
--------+---------+------+-------+---------
 public | test_v1 | view | omm   |
(1 row)
```

- 修改视图所有者。

```
--创建用户。
gaussdb=# CREATE ROLE role_test PASSWORD '********';

--修改视图所有者。
gaussdb=# ALTER VIEW IF EXISTS test_v1 OWNER TO role_test;

--查看视图信息。
gaussdb=# \dv
              List of relations
 Schema |  Name   | Type |   Owner   | Storage
--------+---------+------+-----------+---------
 public | test_v1 | view | role_test |
(1 row)
```

- 设置视图所属模式。

```
--创建模式。
gaussdb=# CREATE SCHEMA tcpds;

--修改视图所属模式。
gaussdb=# ALTER VIEW test_v1 SET SCHEMA tcpds;

--查看视图信息。
gaussdb=# \dv tcpds.test_v1;
              List of relations
 Schema |  Name   | Type |   Owner   | Storage
--------+---------+------+-----------+---------
 tcpds  | test_v1 | view | role_test |
(1 row)
```


版权所有 © 华为云计算技术有限公司

- 设置与重置视图选项。

```
--修改视图选项。
gaussdb=# ALTER VIEW tcpds.test_v1 SET (security_barrier = TRUE);
ALTER VIEW

--查看。
gaussdb=# \dv tcpds.test_v1
                    List of relations
 Schema |  Name   | Type |   Owner   |         Storage
--------+---------+------+-----------+-------------------------
 tcpds  | test_v1 | view | role_test | {security_barrier=true}
(1 row)

--修改check_option选项。
gaussdb=# ALTER VIEW tcpds.test_v1 SET (check_option = 'LOCAL');

--查看。
gaussdb=# \dv tcpds.test_v1
                              List of relations
 Schema |  Name   | Type |   Owner   |                  Storage
--------+---------+------+-----------+-------------------------------------------
 tcpds  | test_v1 | view | role_test | {security_barrier=true,check_option=LOCAL}
(1 row)


--重置视图选项。
gaussdb=# ALTER VIEW tcpds.test_v1 RESET (security_barrier);
ALTER VIEW
gaussdb=# ALTER VIEW tcpds.test_v1 RESET (check_option);
ALTER VIEW

--查看。
gaussdb=# \dv tcpds.test_v1
               List of relations
 Schema |  Name   | Type |   Owner   | Storage
--------+---------+------+-----------+---------
 tcpds  | test_v1 | view | role_test |
(1 row)

--删除视图test_v1。
gaussdb=# DROP VIEW tcpds.test_v1;
DROP VIEW

--删除表test_tb1。
gaussdb=# DROP TABLE test_tb1;
DROP TABLE

--删除用户。
gaussdb=# DROP ROLE role_test;
DROP ROLE
--删除schema。
gaussdb=# DROP SCHEMA tcpds;
DROP SCHEMA
```

- 重编译视图。

```
--开启视图失效重编译参数。
gaussdb=# SET enable_view_invalidation=on;

--创建表。
gaussdb=# CREATE TABLE test_tb1(col1 INT, col2 INT);

--创建视图。
gaussdb=# CREATE VIEW test_v1 AS SELECT col1, col2 FROM test_tb1;

--删除表test_tb1。
gaussdb=# DROP TABLE test_tb1;
DROP TABLE

--查询视图有效性。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT c.relname, o.valid FROM pg_catalog.pg_object o JOIN pg_catalog.pg_class c ON
c.oid = o.object_oid WHERE c.relname = 'test_v1';
 relname | valid
---------+-------
 test_v1 | f
(1 row)

--重建删除的表test_tb1。
gaussdb=# CREATE TABLE test_tb1(col1 INT, col2 INT);

--重编译视图。
gaussdb=# ALTER VIEW test_v1 COMPILE;
ALTER VIEW

--再次查询视图有效性。
gaussdb=# SELECT c.relname, o.valid FROM pg_catalog.pg_object o JOIN pg_catalog.pg_class c ON
c.oid = o.object_oid WHERE c.relname = 'test_v1';
 relname | valid
---------+-------
 test_v1 | t
(1 row)

--删除视图test_v1。
gaussdb=# DROP VIEW test_v1;
DROP VIEW

--删除表test_tb1。
gaussdb=# DROP TABLE test_tb1;
DROP TABLE
```

相关链接

CREATE VIEW，**DROP VIEW**

1.13.7.44 ANALYZE | ANALYSE

功能描述

- 用于收集与数据库中普通表内容相关的统计信息，统计结果存储在系统表PG_STATISTIC、PG_STATISTIC_EXT下，执行ANALYZE命令后，可在上述系统表中查询收集到的统计信息，也可以通过系统视图PG_STATS、PG_EXT_STATS查询信息。执行计划生成器会使用这些统计数据，以确定最有效的执行计划。
- 如果没有指定参数，ANALYZE会分析当前数据库中的每个表和分区表。同时也可以通过指定table_name、column_name和partition_name参数把分析限定在特定的表、列或分区表中。
- ANALYZE|ANALYSE VERIFY用于校验数据库目标表相关文件，具体支持校验类型见下表。

页面类型\校验类型	页面物理损坏	页面逻辑校验	元组校验
UHEAP	支持	支持	COMPLETE模式支持
HEAP	支持	-	COMPLETE模式支持
FSM	支持	-	-
VM	-	-	-

版权所有 © 华为云计算技术有限公司

页面类型\校验类型	页面物理损坏	页面逻辑校验	元组校验
UBTREE	支持	支持	-
UBTREE.FSM(URQ)	支持	支持	-
BTREE	支持	-	-
BTREE.FSM	支持	-	-

- ANALYZE|ANALYSE VERIFY执行完成所需时间与当前操作对象（数据库目标表或数据库）的大小以及当前机器的性能有关。
- 每次收集的统计信息，都会存入统计信息历史表（**GS_STATISTIC_HISTORY**、**GS_STATISTIC_EXT_HISTORY**、**GS_TABLESTATS_HISTORY**）中，历史表存放的数量和统计信息的保留时间由GUC参数stats_history_record_limit和stats_history_retention_time控制。

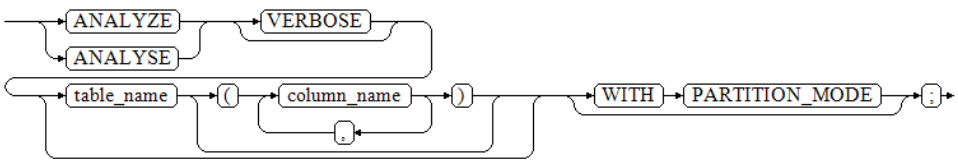
注意事项

- ANALYZE非临时表不支持在一个匿名块、事务块、函数或存储过程内被执行。ANALYZE临时表支持存储过程中被执行，不支持统计信息回滚操作。
- ANALYZE VERIFY如果不涉及远程读场景，远程读参数则不会生效。对于检测出关键系统表页面损坏的错误，将直接报错不再继续检测。
- 如果ANALYZE不指定参数，则默认处理当前用户拥有相应权限的表。如果参数指定了一个表参数，ANALYZE只处理指定的表。
- 要对一个表进行ANALYZE操作，用户必须是表的所有者或者被授予了指定表VACUUM权限，三权分立开关关闭时，默认系统管理员有该权限。数据库的所有者允许对数据库中除了共享目录以外的所有表进行ANALYZE操作（该限制意味着只有系统管理员才能真正对一个数据库进行ANALYZE操作）。ANALYZE会跳过那些用户没有权限的表。
- ANALYZE不收集无法做比较或等值运算的列，例如：CURSOR类型。
- 如果拟分析的表成了一个空表，ANALYZE不会记录该表的统计信息，而原来已有统计信息则会保留。

语法格式

- 收集表的统计信息。

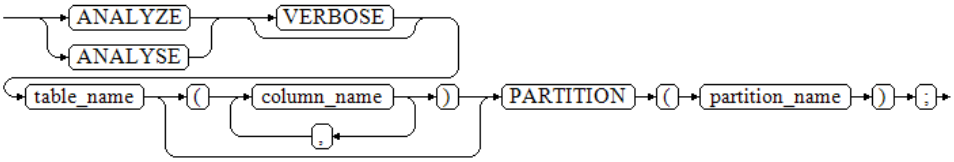
```
{ ANALYZE | ANALYSE } [ VERBOSE ]
    [ table_name [ ( column_name [, ...] ) ] ] [ WITH PARTITION_MODE ];
```



- 收集分区表的分区统计信息。

```
{ ANALYZE | ANALYSE } [ VERBOSE ]
    table_name [ ( column_name [, ...] ) ] PARTITION ( partition_name ) ;
```


版权所有 © 华为云计算技术有限公司

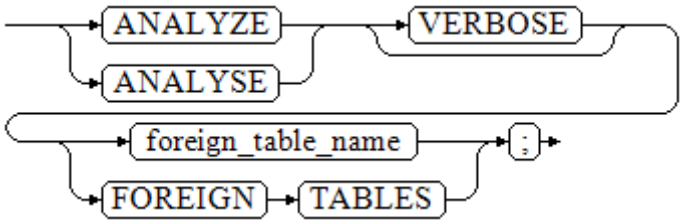


说明

- 使用关键字PARTITION，partition_name必须为一级分区名字。
- 分布式不支持二级分区。

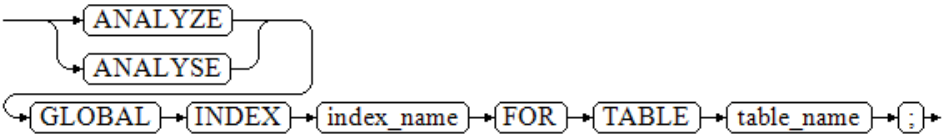
- 收集外表的统计信息。

```
{ ANALYZE | ANALYSE } [ VERBOSE ]
    { foreign_table_name | FOREIGN TABLES };
```



- 收集全局二级索引的统计信息。

```
{ ANALYZE | ANALYSE } GLOBAL INDEX index_name FOR TABLE table_name;
```

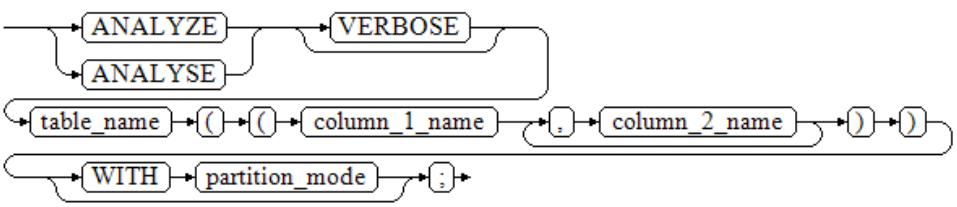


说明

- 在对全局二级索引执行ANALYZE之前需要先对基表执行一次ANALYZE，以保证全局二级索引统计信息的准确性。
- 在对全局二级索引进行REINDEX/REBUILD类型操作之后，需要先对基表进行一次ANALYZE，然后才能对全局二级索引进行ANALYZE，以保证全局二级索引统计信息的准确性。
- 当前对全局二级索引进行统计信息收集必须指定index_name和table_name，且index_name必须是全局二级索引名。
- 全局二级索引统计信息仅包括pg_class系统表中的统计信息（reltuple和relpages）。
- 当ANALYZE执行时间窗内遇到并发的autoanalyze时，较低有可能会产生报错Duplicate key value violates unique constraint "gs_tablestats_history_relid_kind_time_index"，可以重新手动执行ANALYZE进行规避。

- 手动收集多列统计信息。

```
{ ANALYZE | ANALYSE } [ VERBOSE ]
    table_name (( column_1_name, column_2_name [, ...] )) [ WITH partition_mode ];
```



版权所有 © 华为云计算技术有限公司

说明

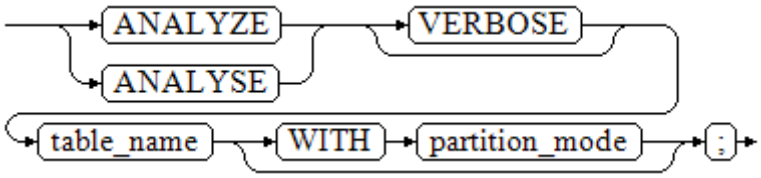
- 如果关闭GUC参数enable_functional_dependency，每组多列统计信息最多支持32列。
- 不支持收集多列统计信息的表：系统表。

- 自动收集多列统计信息。

  打开auto_statistic_ext_columns参数后执行ANALYZE，自动根据该表的索引前缀创建多列统计信息，多列统计信息的列数不超过设置的auto_statistic_ext_columns值。

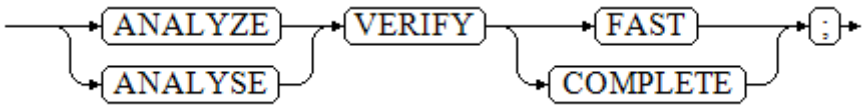
  例：表t存在索引(a,b,c,d)，设置auto_statistic_ext_columns参数为4，则analyze t将创建关于(a,b)、(a,b,c)、(a,b,c,d)的多列统计信息。

  ```
  { ANALYZE | ANALYSE } [ VERBOSE ] table_name [ WITH partition_mode ];
  ```

- 检测当前库的数据文件。

  ```
  { ANALYZE | ANALYSE } VERIFY { FAST | COMPLETE };
  ```

  说明

  - 由于该操作会直接从磁盘上读取数据，如果此时对正在校验的表有并发的DML操作，即并发修改磁盘文件，会导致获取的数据不准确，进而导致校验过程中出现误报的情况，因此建议在没有数据库业务时或者业务低峰期使用。
  - 为保障执行效率，建议执行此操作前，先执行checkpoint。
  - 支持对全库进行操作，由于涉及的表较多，建议以重定向保存结果。

    ```
    gsql -d database -p port -f sqlfile> sqllog.txt  2>&1
    ```

  - 只有对外可见的表对外提示NOTICE，内部表的检测会包含在它所依赖的外部表，不对外显示。
  - 对于全库操作，当出现损坏则直接报错，不再继续执行。
  - 只支持对当前库中的普通表、普通索引进行校验，不支持对本地临时表、UNLOGGED表进行校验，在实际过程中会自动跳过对不支持的表类型的校验。
  - 支持在备机执行。当库中表或数据较多时，执行时间较长，由于备机上有recovery业务，analyze verify业务可能被自动取消，并屏显相关报错信息。

- 检测表和索引的数据文件。

  ```
  { ANALYZE | ANALYSE } VERIFY { FAST  | COMPLETE } { table_name  [ CASCADE ] | index_name };
  ```

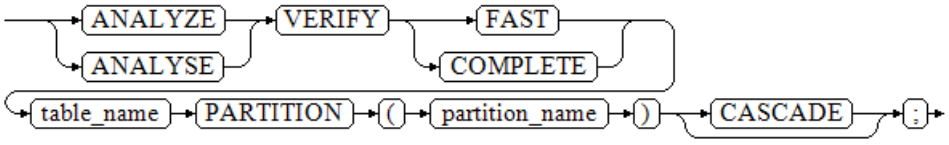

版权所有 © 华为云计算技术有限公司

说明

- 支持对普通表的操作和索引表的操作，但不支持对索引表index使用CASCADE操作。原因是由于CASCADE模式用于处理主表的所有索引表，当单独对索引表进行检测时，无需使用CASCADE模式。
- 对于主表的检测会同步检测主表的内部表，例如toast表等。
- 当提示索引表损坏时，建议使用reindex命令进行重建索引操作。
- 不支持对本地临时表、UNLOGGED表进行校验。
- 支持在备机执行。

- 检测表分区的数据文件。

```
{ ANALYZE | ANALYSE } VERIFY { FAST | COMPLETE } table_name PARTITION (partition_name)
[ CASCADE ];
```



说明

- 支持对表的单独分区进行检测操作，但不支持对索引表index使用CASCADE操作。
- 不支持指定二级分区。
- 支持在备机执行。

参数说明

- **VERBOSE**

  启用显示进度信息。

  说明

  如果指定了VERBOSE，ANALYZE发出进度信息，表明目前正在处理的表。各种有关表的统计信息也会打印出来。

- **table_name**

  需要分析的特定表的表名（可能会带模式名），如果省略，将对数据库中的所有表（非外部表）进行分析。

  取值范围：已有的表名。

- **column_name，column_1_name，column_2_name**

  需要分析特定列的列名，默认为所有列。

  取值范围：已有的列名。

- **partition_name**

  如果table为分区表，在关键字PARTITION后面指定分区名partition_name表示分析该分区表的统计信息。

  取值范围：表的某一个分区名。

- **foreign_table_name**

  需要分析的特定表的表名（可能会带模式名）。

  取值范围：已有的表名。

- **FOREIGN TABLES**

  分析所有当前用户权限下的外表。

版权所有 © 华为云计算技术有限公司

- **index_name**

  需要分析的特定索引表的表名（可能会带模式名）。

  取值范围：已有的表名。

- **FAST|COMPLETE**

  对于行存表，FAST模式下主要对于行存表的CRC和page header进行校验，如果校验失败则会告警；而COMPLETE模式下，则主要对行存表的指针、tuple进行解析校验。

- **CASCADE**

  CASCADE模式下会对当前表的所有索引进行检测处理。

- **PARTITION_MODE**

  PARTITION_MODE适用于分区表级联收集统计信息，可选项和含义如**表1-392**所示。不适用于非分区表。

  表 1-392 PARTITION_MODE 选项说明

PARTITION_MODE选项	含义
ALL	收集整表、一级分区的统计信息。
GLOBAL	收集整表的统计信息。
PARTITION	收集一级分区的统计信息。
GLOBAL AND PARTITION	收集整表、一级分区的统计信息。
ALL COMPLETE	收集整表、一级分区的统计信息。
AUTO	缺省值，以statistic_granularity或default_statistic_granularity中配置的参数为准，其中表级参数statistic_granularity的优先级高于全局参数default_statistic_granularity。statistic_granularity介绍请参见**statistic_granularity**。

版权所有 © 华为云计算技术有限公司

说明

- ALL和ALL COMPLETE的区别：ALL模式下对整表进行一次采样，ALL COMPLETE模式下对每个分区进行一次采样。对于分区来说，ALL COMPLETE模式比ALL模式计算统计信息更准确，同时也消耗更多资源。
- 分区统计信息会在cplan裁剪到单分区时使用。建议结合统计信息的收集时间与业务实际需求，选择合适的分区表统计信息收集策略。
- 在大规模导数后，建议使用ALL COMPLETE模式手动收集分区级的统计信息。
- 从505.0.0之前版本升级到505.0.0及更高版本时，在升级观察期期间，PARTITION_MODE不生效，其行为与升级前版本保持一致。
- 当对整个数据库进行ANALYZE操作时，不支持指定PARTITION_MODE，即不支持ANALYZE WITH GLOBAL等语法。
- 当对整个数据库进行ANALYZE操作时，支持设置default_statistic_granularity为GLOBAL或ALL，此时ANALYZE行为和default_statistic_granularity的定义相同；当default_statistic_granularity设置为其他值时，ANALYZE行为退化为GLOBAL行为。
- 不支持对二级分区的统计信息的收集。
- ALL模式下，如果分区表上的partition_statistics_min_rows不为0，且GUC参数**enable_partition_statistics_min_rows**打开时，会保证分区的最小采样行数为partition_statistics_min_rows，但是整表总采样行数上限会受到GUC参数**maintenance_work_mem**控制。另外，在不同分区数据量存在严重倾斜的状态下，在ALL模式下执行ANALYZE后，建议补充一次针对整表的GLOBAL模式的ANALYZE，保证整表的统计信息可更准确的反映数据的倾斜特征。

示例

- 收集表的统计信息。

```
--创建customer_info表。
gaussdb=# CREATE TABLE customer_info(
wr_returned_date_sk INTEGER ,
wr_returned_time_sk INTEGER ,
wr_item_sk INTEGER NOT NULL
,wr_refunded_customer_sk INTEGER ) DISTRIBUTE BY HASH (wr_item_sk);

--使用ANALYZE语句更新统计信息。
gaussdb=# ANALYZE customer_info;

--使用ANALYZE VERBOSE语句更新统计信息，并输出customer_info表的相关信息。
gaussdb=# ANALYZE VERBOSE customer_info;
INFO: analyzing "public.customer_info"(cn_5002 pid=53078)

--使用ANALYZE VERBOSE输出customer_info表的wr_returned_time_sk列信息。
gaussdb=# ANALYZE VERBOSE customer_info(wr_returned_time_sk);
INFO:  analyzing "public.customer_info"(cn_5002 pid=53078)
```

- 收集分区表的统计信息。

```
--创建分区表。
gaussdb=# CREATE TABLE customer_par(
wr_returned_date_sk INTEGER ,
wr_returned_time_sk INTEGER ,
wr_item_sk INTEGER NOT NULL,
wr_returned_customer_sk INTEGER)
DISTRIBUTE BY HASH (wr_item_sk)PARTITION BY RANGE(wr_returned_date_sk)(PARTITION P1
VALUES LESS THAN(2452275),PARTITION P2 VALUES LESS THAN(2452640),PARTITION P3 VALUES
LESS THAN(2453000),PARTITION P4 VALUES LESS THAN(MAXVALUE))ENABLE ROW MOVEMENT;

--使用ANALYZE语句更新统计信息。
gaussdb=# ANALYZE customer_par;

--使用ANALYZE VERBOSE输出customer_par表的相关信息。
gaussdb=# ANALYZE VERBOSE customer_par;
INFO:  analyzing "public.customer_par"(cn_5001 pid=446617)

--使用ANALYZE VERBOSE输出P1一级分区表的相关信息。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# ANALYZE VERBOSE customer_par PARTITION(P1);
INFO:  analyzing "public.customer_par"(cn_5001 pid=446617)
```

- 手动收集多列统计信息。

```
--手动收集wr_returned_date_sk、wr_returned_time_sk这两列的统计信息。
gaussdb=# ANALYZE VERBOSE customer_info (wr_returned_date_sk,wr_returned_time_sk);
INFO:  analyzing "public.customer_par"(cn_5001 pid=446617)
```

- 自动收集多列统计信息。

```
--给customer_info表创建索引。
gaussdb=# CREATE INDEX customer_index ON customer_info USING
btree(wr_returned_date_sk,wr_returned_time_sk,wr_item_sk,wr_refunded_customer_sk);

--设置auto_statistic_ext_columns参数为4。
gaussdb=# set auto_statistic_ext_columns=4;

--自动收集多列统计信息。
gaussdb=# ANALYZE VERBOSE customer_info;
INFO:  analyzing "public.customer_info"(cn_5001 pid=446617)
```

- 检测当前库的数据文件。

```
gaussdb=# ANALYZE VERIFY FAST;
```

- 检测表和索引的数据文件。

```
--检查customer_info表。
gaussdb=# ANALYZE VERIFY FAST customer_info;

--检查customer_index索引。
gaussdb=# ANALYZE VERIFY FAST customer_index;
```

- 检测表分区的数据文件。

```
--检查customer_par分区表中的P1分区。
gaussdb=# ANALYZE VERIFY FAST customer_par PARTITION (P1);
```

- 删除数据。

```
--删除索引customer_index。
gaussdb=# DROP INDEX customer_index;

--删除表customer_info。
gaussdb=# DROP TABLE customer_info;

--删除分区表customer_par。
gaussdb=# DROP TABLE customer_par;
```

1.13.7.45 AUTOHINT

功能描述

启用AUTOHINT用于指定SQL探索Hint组合，并输出能提升当前SQL执行速度的推荐Hint组合。

注意事项

- 仅支持SELECT语句，并且需要使用者拥有super权限或管理员权限。
- 支持的多种选项中，除ANALYZE选项外，使用其他选项时需将其置于括号内，例如：AUTOHINT (VERBOSE) statement。

语法格式

- 使用AUTOHINT为执行SQL进行Hint推荐，支持多种选项，对选项顺序无要求。

```
AUTOHINT [ (  option  [, ...] ) ] statement;
```

其中选项option子句的语法为。

```
{
   ANALYZE [ boolean ] |
   VERBOSE [ boolean ] |
```


版权所有 © 华为云计算技术有限公司

```
  SQLPATCH [ boolean ] |
  TEST [ boolean ] |
  DEBUG [ boolean ]
}
```

参数说明

- **statement**

  指定进行Hint推荐的SQL语句（仅支持SELECT语句）。

  须知

  若SQL语句已经自带Hint，AUTOHINT将会无视已有的Hint进行推荐。例如，对于语句SELECT /*+IndexScan(t1)*/* FROM t1, t2 where t1.id = t2.id and t1.a > 20，使用AUTOHINT进行推荐后，得到的推荐结果是基于语句SELECT * FROM t1, t2 where t1.id = t2.id and t1.a > 20 的推荐结果，而非在已有的 IndexScan(t1) 的基础上进行的进一步推荐。

- **ANALYZE boolean**

  在进行HINT推荐前首先进行HINT探索，若关闭则根据历史HINT探索记录直接对当前SQL进行HINT推荐。

  取值范围：

  - TRUE（缺省值）：在推荐前进行HINT探索。
  - FALSE：不进行HINT探索。

- **VERBOSE boolean**

  显示HINT探索与推荐过程的额外信息。

  取值范围：

  - TRUE（缺省值）：显示额外信息。
  - FALSE：不显示。

- **SQLPATCH boolean**

  推荐HINT后输出相应的SQLPATCH指令。

  取值范围：

  - TRUE（缺省值）：输出SQLPATCH指令。
  - FALSE：不输出。

- **TEST boolean**

  推荐HINT后，分别执行初始查询和应用HINT后的查询来验证推荐结果，并输出实际计划执行时间。

  取值范围：

  - TRUE（缺省值）：进行推荐结果验证并输出验证结果。
  - FALSE：不进行验证。

- **DEBUG boolean**

  输出相对于VERBOSE选项更多的细节信息，包括每轮探索时的初始计划树及执行时间，并打印每轮探索结束后得到的探索子图，以及最终合并后总的探索图。

  取值范围：

  - TRUE（缺省值）：输出细节信息。

版权所有 © 华为云计算技术有限公司

- FALSE：不输出。

示例 1

使用AUTOHINT为慢SQL语句推荐HINT。以JOB数据集中的语句为例。

```
--导入JOB数据集，使用ANALYZE选项为02a查询进行HINT探索推荐。
gaussdb=# AUTOHINT ANALYZE SELECT MIN(t.title) AS movie_title
FROM company_name AS cn,
    keyword AS k,
    movie_companies AS mc,
    movie_keyword AS mk,
    title AS t
WHERE cn.country_code ='[de]'
  AND k.keyword ='character-name-in-title'
  AND cn.id = mc.company_id
  AND mc.movie_id = t.id
  AND t.id = mk.movie_id
  AND mk.keyword_id = k.id
  AND mc.movie_id = mk.movie_id;
--得到HINT推荐结果与预期提升。
gaussdb=#
                    AUTOHINT
------------------------------------------------------------
 Recommended hintset:
   MergeJoin(@sel$1 k@sel$1 mk@sel$1 mc@sel$1 cn@sel$1)

 Expected boost:
   planning:     0%
   execution:    50%
   overall:      50%
(7 rows)
```

1.13.7.46 AUTOHINT DROP MODEL

功能描述

删除指定SQL的历史HINT探索结果。

注意事项

仅支持SELECT语句，并且需要使用者拥有super权限或管理员权限。

语法格式

删除指定SQL的历史HINT探索结果。

```
AUTOHINT DROP MODEL statement;
```

参数说明

statement

指定删除探索记录的SQL语句（仅支持SELECT语句）。

示例

使用AUTOHINT DROP MODEL删除指定SQL的HINT历史探索结果。

```
gaussdb=# AUTOHINT DROP MODEL SELECT MIN(t.title) AS movie_title
FROM company_name AS cn,
    keyword AS k,
```

 版权所有 © 华为云计算技术有限公司

```
    movie_companies AS mc,
    movie_keyword AS mk,
    title AS t
WHERE cn.country_code ='[de]'
  AND k.keyword ='character-name-in-title'
  AND cn.id = mc.company_id
  AND mc.movie_id = t.id
  AND t.id = mk.movie_id
  AND mk.keyword_id = k.id
  AND mc.movie_id = mk.movie_id;
--删除成功。
AUTOHINT: 1 MODEL REMOVED
--若无该SQL的历史探索结果，删除失败。
AUTOHINT: 0 MODELS REMOVE
```

1.13.7.47 AUTOHINT PURGE

功能描述

删除AUTOHINT的历史HINT探索结果。

注意事项

需要使用者拥有super权限或管理员权限。

语法格式

删除AUTOHINT的历史HINT探索结果。

```
AUTOHINT PURGE;
```

示例

使用AUTOHINT PURGE删除AUTOHINT的所有历史探索结果。

```
gaussdb=# AUTOHINT PURGE;
--删除成功。
AUTOHINT: 12 MODELS REMOVED
--若无任何的历史探索结果，删除失败。
AUTOHINT: 0 MODELS REMOVE
```

1.13.8 B

1.13.8.1 BEGIN

功能描述

BEGIN可以用于开始一个匿名块，也可以用于开始一个事务。

匿名块是能够动态地创建和执行过程代码的结构，而不需要以持久化的方式将代码作为数据库对象储存在数据库中。

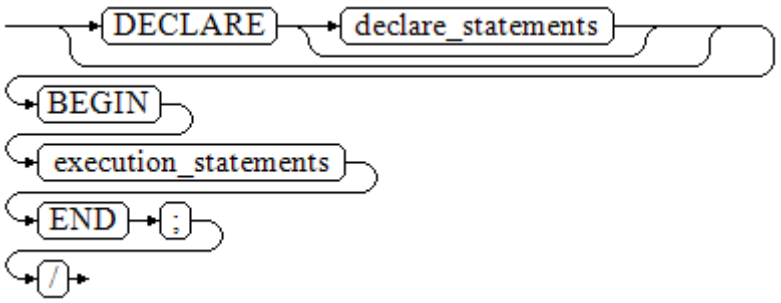
语法格式

- 开启匿名块

  ```
  [DECLARE [declare_statements]]
  BEGIN
  execution_statements
  ```

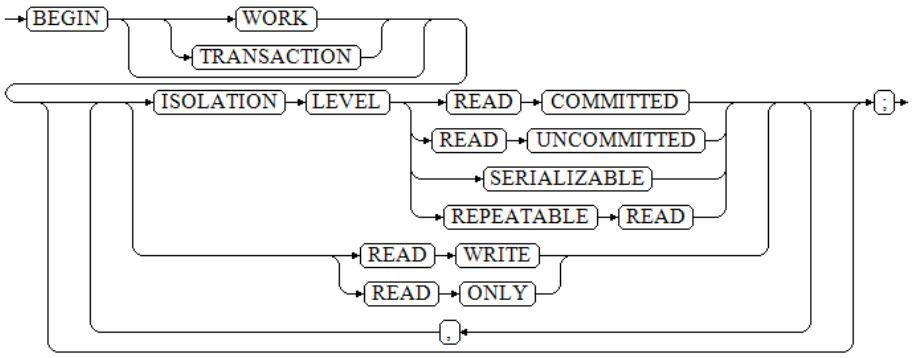

版权所有 © 华为云计算技术有限公司

```
END;
/
```



- 开启事务

```
BEGIN [ WORK | TRANSACTION ]
 [
  {
    ISOLATION LEVEL { READ COMMITTED | READ UNCOMMITTED | SERIALIZABLE | REPEATABLE
READ }
    | { READ WRITE | READ ONLY }
  } [, ...]
 ];
```



参数说明

- **declare_statements**

  声明变量，包括变量名和变量类型，如“sales_cnt int”。

- **execution_statements**

  匿名块中需要执行的语句。

  取值范围：DML操作（SELECT、INSERT、DELETE、UPDATE）或系统表中已注册的函数名称。

- **WORK | TRANSACTION**

  BEGIN语法格式中的可选关键字，没有实际作用。

- **ISOLATION LEVEL**

  指定事务隔离级别，用于控制并发事务之间的数据可见性。

  说明

  在事务中第一个数据修改语句（INSERT、DELETE、UPDATE、FETCH或COPY）执行之后，事务隔离级别无法再次设置。

  取值范围：

  - READ COMMITTED：读已提交隔离级别，仅能读取已提交的数据，不会读取到未提交的数据。未指定隔离级别时，默认为READ COMMITTED。

版权所有 © 华为云计算技术有限公司

  - READ UNCOMMITTED：读未提交隔离级别，指定后的行为和READ COMMITTED行为一致。
  - REPEATABLE READ：可重复读隔离级别，仅可看到事务开始之前提交的数据，不能看到未提交的数据，以及在事务执行期间由其它并发事务提交的修改。
  - SERIALIZABLE：目前功能上不支持此隔离级别，设置该隔离级别时，等价于REPEATABLE READ。
- **READ WRITE | READ ONLY**

  指定事务访问模式（读/写模式或者只读模式）。

示例

- 开启事务

```
--建表并插入数据。
gaussdb=# CREATE TABLE tbl_test1(col1 int, col2 int);
gaussdb=# INSERT INTO tbl_test1 VALUES (1,1), (2,2), (3,3);

--以默认方式启动事务。
gaussdb=# BEGIN;
INSERT INTO tbl_test1 VALUES (4,4);
END;

--以隔离级别为REPEATABLE READ方式启动事务。
gaussdb=# BEGIN ISOLATION LEVEL REPEATABLE READ;
SELECT * FROM tbl_test1;
END;

--删除表。
gaussdb=# DROP TABLE tbl_test1;
```

- 匿名块

```
--使用匿名块输出字符串。
gaussdb=# BEGIN
dbe_output.print_line('Hello');
END;
/
```

相关链接

START TRANSACTION

1.13.9 C

1.13.9.1 CALL

功能描述

使用CALL命令可以调用已定义的函数和存储过程。

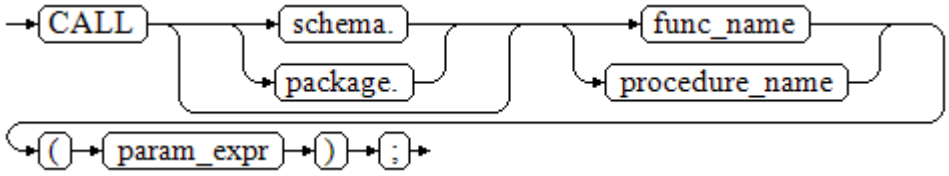
注意事项

- 函数或存储过程的所有者、被授予了函数或存储过程EXECUTE权限的用户或被授予EXECUTE ANY FUNCTION权限的用户，可以执行CALL命令，当三权分立开关关闭时，系统管理员默认拥有此权限。
- 在database为当前数据库或pg_catalog时，支持使用三段式database.schema.name调用函数或存储过程，此时调用的均是当前数据库的对象。

版权所有 © 华为云计算技术有限公司

语法格式

```
CALL [ schema.|package.] { func_name | procedure_name } ( param_expr );
```



参数说明

- **schema.**

  函数或存储过程所在的模式名称。

- **package.**

  函数或存储过程所在的PACKAGE名称。

- **func_name**

  要调用的函数的名称。

  取值范围：已存在的函数名称。

  说明

  - 支持使用DATABASE LINK方式对远端函数或存储过程进行操作，具体使用方式及约束请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。
  - 要调用的函数可以是独立函数，也可以是指定PACKAGE或指定模式中的成员。
  - 创建和系统函数重名的函数时，调用时需要指定函数所在的具体Schema。

- **procedure_name**

  要调用的存储过程的名称。

  取值范围：字符串，要符合**标识符命名规范**。

  说明

  - 支持使用DATABASE LINK方式对远端函数或存储过程进行操作，具体使用方式及约束请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。
  - 要调用的存储过程可以是独立过程，也可以是指定PACKAGE或指定模式中的成员。

- **param_expr**

  参数列表。

  可以使用命名标记法的方式传参，即使用“:=”符号或者“=>”符号将参数名和参数值隔开，这种方法可以对参数进行任意顺序的排列。

  若按照直接传入参数值的方式传参，即参数列表中仅出现参数值，则参数值的排列顺序必须和函数或存储过程定义时的相同。

  取值范围：已存在的函数参数名称或存储过程参数名称。

  说明

  - 参数可以包含入参（参数名和类型之间指定“IN”关键字）和出参（参数名和类型之间指定“OUT”关键字），使用CALL命令调用函数或存储过程时，对于非重载的函数，参数列表必须包含出参，出参可以传入一个变量或者任一常量。对于重载的PACKAGE函数，参数列表里可以忽略出参，忽略出参时可能会导致函数无法匹配；包含出参时，出参只能是常量。
  - 仅支持在ORA兼容性数据库中使用PACKAGE。

版权所有 © 华为云计算技术有限公司

示例

```
--创建一个函数func_add_sql，计算两个整数的和，并返回结果。
gaussdb=# CREATE FUNCTION func_add_sql(num1 integer, num2 integer) RETURN integer
AS
BEGIN
RETURN num1 + num2;
END;
/

--按参数值传参。
gaussdb=# CALL func_add_sql(1, 3);

--使用命名标记法传参。
gaussdb=# CALL func_add_sql(num1 => 1,num2 => 3);
gaussdb=# CALL func_add_sql(num2 := 2, num1 := 3);

--删除函数。
gaussdb=# DROP FUNCTION func_add_sql;

--创建包含出参的函数。
gaussdb=# CREATE FUNCTION func_increment_sql(num1 IN integer, num2 IN integer, res OUT integer)
RETURN integer
AS
BEGIN
res := num1 + num2;
END;
/

--出参传入常量。
gaussdb=# CALL func_increment_sql(1,2,1);

--出参传入变量。
gaussdb=# DECLARE
res int;
BEGIN
func_increment_sql(1, 2, res);
dbe_output.print_line(res);
END;
/

--删除函数。
gaussdb=# DROP FUNCTION func_increment_sql;

--创建重载的函数。
gaussdb=# CREATE OR REPLACE PROCEDURE package_func_overload(col int, col2 out int) PACKAGE
AS
DECLARE
    col_type text;
BEGIN
    col := 122;
    dbe_output.print_line('two out parameters ' || col2);
END;
/

gaussdb=# CREATE OR REPLACE PROCEDURE package_func_overload(col int, col2 out varchar) PACKAGE
AS
DECLARE
    col_type text;
BEGIN
    col2 := '122';
    dbe_output.print_line('two varchar parameters ' || col2);
END;
/

--函数调用。
gaussdb=# CALL package_func_overload(1, 'test');
gaussdb=# CALL package_func_overload(1, 1);
```


版权所有 © 华为云计算技术有限公司

```
--删除函数。
gaussdb=# DROP FUNCTION package_func_overload;
```

相关链接

CREATE FUNCTION，CREATE PROCEDURE

1.13.9.2 CHECKPOINT

功能描述

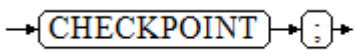
设置事务日志检查点。检查点（CHECKPOINT）为一个事务日志中的点，所有数据文件都在该点被更新以反映日志中的信息，所有数据文件都将被刷新到磁盘。预写式日志（WAL）默认情况下在事务日志中每隔一段时间放置一个检查点，可以使用gs_guc命令设置相关运行时参数（checkpoint_segments, checkpoint_timeout和incremental_checkpoint_timeout）进行调整该原子化检查点的间隔。

注意事项

- 只有系统管理员和运维管理员可以调用CHECKPOINT命令。
- CHECKPOINT会立即进行检查，而不是等到下一次调度时的检查点。

语法格式

```
CHECKPOINT;
```



示例

```
--设置检查点。
gaussdb=# CHECKPOINT;
```

1.13.9.3 CLEAN CONNECTION

功能描述

清理CN节点到其他指定数据库节点（CN/DN）的网络连接。允许清理CN节点中指定数据库、指定用户的相关空闲/无效连接、强制清理活跃的连接。

注意事项

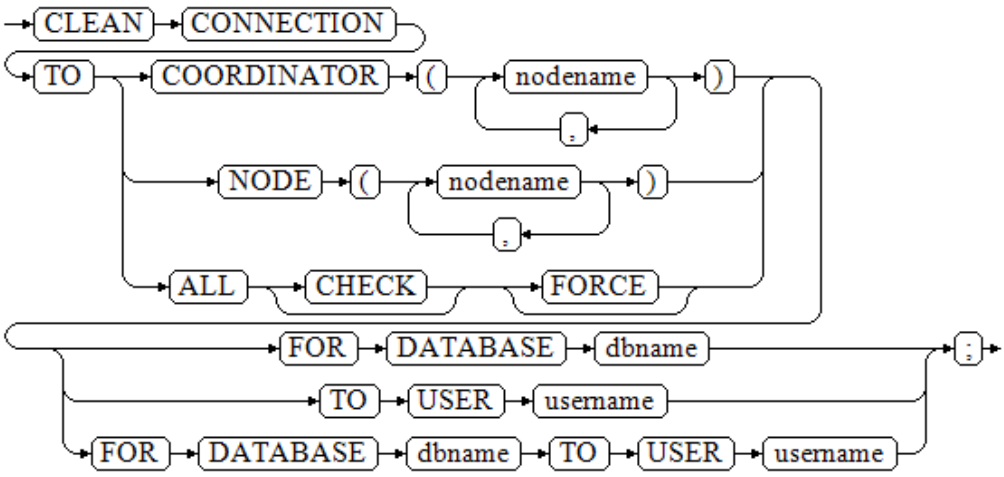
- 建议仅在数据库出现网络连接异常时执行CLEAN CONNECTION命令，可以通过查询pg_stat_get_pooler_status()函数查看缓存的连接，检验清理的效果。
- 未指定FORCE模式时，存在以下结果：
  - CLEAN CONNECTION操作仅清理数据库节点（CN/DN）之间的连接，不影响客户端连接。
  - CLEAN CONNECTION操作仅清理CN中已缓存的空闲/无效的连接，正在使用的正常连接不被清理。
- 指定FORCE模式时，存在以下结果：
  - CLEAN CONNECTION操作可同时强制清理正在使用的正常连接以及客户端连接。

版权所有 © 华为云计算技术有限公司

- CLEAN CONNECTION操作会清理DN中指定数据库、指定用户的连接。若DN存在无法清理的残留连接，建议运维通过pg_terminate_session函数进行清理该无效会话。

语法格式

```
CLEAN CONNECTION
   TO { COORDINATOR ( nodename [, ... ] ) | NODE ( nodename [, ... ] )| ALL [ CHECK ] [ FORCE ] }
   { FOR DATABASE dbname | TO USER username | FOR DATABASE dbname TO USER username };
```



参数说明

- **CHECK**

  主要用于DROP DATABASE操作之前的连接访问检查，若发现存在其他会话连接，则报错并终止删除操作。仅当节点列表为TO ALL时可以指定该参数，指定该参数后，会在清理连接前检查数据库是否被其他会话连接访问。

- **FORCE**

  清理模式，指定该参数后，当前CN中所有与指定数据库和指定用户相关的线程都将会收到SIGTERM信号，相应的会话被强制关闭、事务会被中止以及网络连接会被清理；DN中所有与指定数据库和指定用户相关的线程都将会收到SIGTERM信号，并且会被强制关闭。

  仅在节点列表指定为TO ALL时可以指定该参数。

- **COORDINATOR (nodename [, ...] }) | NODE (nodename [, ...]) | ALL**

  节点列表，删除CN节点与指定节点的空闲/无效连接。存在以下场景：

  - COORDINATOR：删除CN到指定CN节点上的空闲/无效连接。
  - NODE：删除CN到指定DN节点上的空闲/无效连接。
  - ALL：删除CN到所有节点上的空闲/无效连接，包括CN和DN。

  取值范围：已存在的节点名。

- **dbname**

  删除当前CN节点中指定数据库相关的连接。如果未指定，则默认删除所有数据库相关的连接。

  取值范围：已存在数据库名称。

- **username**

  删除当前CN节点中指定用户相关连接。如果未指定，则默认删除所有用户相关的连接。

版权所有 © 华为云计算技术有限公司

取值范围：已存在的用户。

示例

```
--创建数据库test_clean_connection。
gaussdb=# CREATE DATABASE test_clean_connection;

--创建jack用户。
gaussdb=# CREATE USER jack PASSWORD '********';

--在另一个会话用jack用户登录该数据库之后，通过视图查询到该连接信息。
gaussdb=# SELECT datname,usename,application_name,waiting,state
          FROM pg_stat_activity
          WHERE datname = 'test_clean_connection';
        datname         | usename | application_name | waiting | state
------------------------+---------+------------------+---------+-------
 test_clean_connection | jack    | gsql             | f       | idle
(1 row)

--此时直接删除数据库test_clean_connection会有如下报错：
gaussdb=# DROP DATABASE test_clean_connection;
ERROR:  Database "test_clean_connection" is being accessed by other users. You can stop all connections by command: "clean connection to all force for database XXXX;" or wait for the sessions to end by querying view: "pg_stat_activity".
DETAIL:  There is 1 other session using the database.

--删除登录数据库数据库test_clean_connection的当前CN节点与所有节点的连接。
--如果不使用FORCE参数是无法删除stat状态为其他的状态的连接。
gaussdb=# CLEAN CONNECTION TO ALL FORCE FOR DATABASE test_clean_connection;

--查询登录数据库test_clean_connection的连接。
gaussdb=# SELECT datname,usename,application_name,waiting,state
          FROM pg_stat_activity
          WHERE datname = 'test_clean_connection';
 datname | usename | application_name | waiting | state
---------+---------+------------------+---------+-------
(0 rows)

--删除数据库test_clean_connection。
gaussdb=# DROP DATABASE test_clean_connection;

--删除用户jack。
gaussdb=# DROP USER jack;
```

1.13.9.4 CLOSE

功能描述

释放与指定游标相关联的所有资源。

注意事项

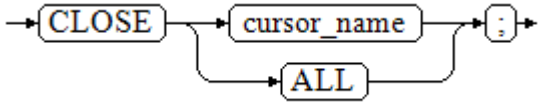
- 一个不再使用的游标应当及时关闭，以释放其占用的数据库资源，避免资源泄露。
- 不允许对一个已关闭的游标再进行任何操作。
- 游标的关闭行为与其可保持性属性密切相关，具体如下：
  - 不可保持游标（WITHOUT HOLD）：当创建该游标的事务以COMMIT或ROLLBACK结束时，不可保持游标将自动隐式关闭。
  - 可保持游标（WITH HOLD）：
    - 若事务以ROLLBACK结束，可保持游标将被隐式关闭。

 版权所有 © 华为云计算技术有限公司

    - 若事务成功提交（COMMIT），可保持游标仍保持打开状态，继续有效，直到执行显式的CLOSE操作或客户端连接终止。
- GaussDB没有明确打开游标的OPEN语句，因为一个游标在使用CURSOR命令定义时就已经打开，可以通过查询系统视图PG_CURSORS获取所有可用的游标。

语法格式

```
CLOSE { cursor_name | ALL } ;
```



参数说明

- **cursor_name**

  指定待关闭或释放的游标名称。
- **ALL**

  关闭或释放所有已打开的游标。

示例

具体示例请参见**示例**。

相关链接

FETCH，**MOVE**

1.13.9.5 CLUSTER

功能描述

使用指定索引对指定表执行聚簇排序操作，该索引必须已存在于目标表内。

- GaussDB在执行表聚簇排序操作后，该表将基于索引信息进行物理存储。CLUSTER是一次性操作，当该表被更新之后，更改的内容不会被重新聚簇，即系统不会试图按照索引顺序对新的存储内容及更新记录进行重新聚簇。
- GaussDB在执行表聚簇排序操作后，会记录该表所使用的聚簇索引。后续执行CLUSTER *table_name*;命令时，数据库将自动基于先前记录的聚簇索引进行重新聚簇。用户可通过ALTER TABLE *table_name* CLUSTER ON *index_name*;指定后续聚簇排序操作的聚簇索引，或通过ALTER TABLE *table_name* SET WITHOUT CLUSTER;清除该表的聚簇索引设置。
- 无参数CLUSTER命令会对当前用户所拥有的数据库中所有已聚簇表进行重新聚簇。若系统管理员调用该命令，则对所有已聚簇表重新聚簇排序。
- GaussDB在执行表聚簇排序操作时，会对该表请求ACCESS EXCLUSIVE锁，从而避免在CLUSTER操作完成前对该表执行其他的操作（包括读写）。

注意事项

- 只有行存B-Tree索引支持CLUSTER聚簇排序操作。

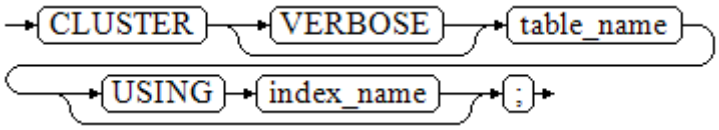
版权所有 © 华为云计算技术有限公司

- 当表数据访问模式为随机读取时，数据存储顺序无显著影响；若存在频繁按特定索引访问的场景，且索引对数据具有分组效应，使用CLUSTER可提升查询性能。
- 当查询条件涉及索引范围扫描或索引键对应多行数据时，CLUSTER操作可有效提升查询性能。若索引已标识出首条匹配记录所在的存储页，其余匹配行通常位于同一页内，从而减少磁盘I/O并加速查询。
- GaussDB在执行表聚簇排序操作时，会先创建一个按照索引顺序建立的表的临时备份，同时也建立表上的每个索引的临时备份。因此，聚簇过程中需要保证磁盘上有足够的剩余空间（至少为表大小与全部索引大小之和）。
- 优化器依赖表的排序统计信息进行查询规划。对表执行聚簇排序操作后，需运行ANALYZE操作以更新优化器的排序统计信息。若未及时更新，优化器可能基于过时信息生成非最优的查询执行计划。
- 不允许在事务中执行CLUSTER聚簇排序操作。
- 如果没有将GUC参数xc_maintenance_mode设置为on，那么CLUSTER聚簇排序操作将忽略所有的系统表。
- 对于全局二级索引（GSI）：
  - 在单个分区上执行CLUSTER命令或通过CLUSTER [VERBOSE];对已聚簇表重新进行聚簇后，查询语句若访问全局二级索引将报错，需执行REINDEX INDEX重建全局二级索引以恢复查询。
  - 对带有全局二级索引（GSI）的表执行CLUSTER操作后，需要重新收集GSI的统计信息，即调用ANALYZE GLOBAL INDEX gsi_name FOR TABLE table_name语句。具体操作请参见**ANALYZE | ANALYSE**。
- 通过表达式索引进行聚簇排序操作时，需拥有此表达式索引的重建索引的权限。

语法格式

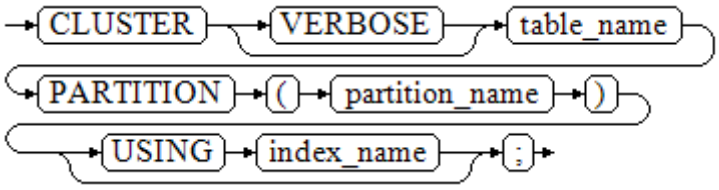
- 对表进行聚簇排序操作

```
CLUSTER [ VERBOSE ] table_name [ USING index_name ];
```



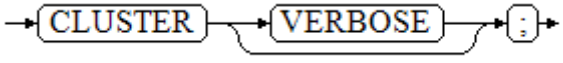
- 对分区进行聚簇排序操作

```
CLUSTER [ VERBOSE ] table_name PARTITION ( partition_name ) [ USING index_name ];
```



- 对已聚簇表重新进行聚簇排序操作

```
CLUSTER [ VERBOSE ];
```



参数说明

- **VERBOSE**

  可选。用于控制是否输出操作的详细进度信息。

版权所有 © 华为云计算技术有限公司

- **table_name**

  需要进行聚簇排序操作的指定表名称。

  取值范围：已存在的表名称。

- **USING index_name**

  对表/分区进行聚簇排序时，指定的索引名称。

  取值范围：已存在的索引名称。

  说明

  首次对表进行聚簇排序操作时，必须指定该选项；当再次对表进行聚簇排序操作时，可省略指定index_name，数据库将自动基于先前记录的聚簇索引进行重新聚簇。

- **partition_name**

  对分区进行聚簇排序时，指定的分区名称。

  取值范围：已存在的分区名称。

示例

- 对表进行聚簇排序操作：

```
--建表并向表中插入数据。
gaussdb=# CREATE TABLE test_c1(id int, name varchar(20));
gaussdb=# CREATE INDEX idx_test_c1_id ON test_c1(id);
gaussdb=# INSERT INTO test_c1 VALUES (3,'Joe'),(1,'Jack'),(2,'Scott');

--查看。
gaussdb=# SELECT * FROM test_c1;
 id | name
----+-------
  3 | Joe
  1 | Jack
  2 | Scott
(3 rows)

--聚簇排序。
gaussdb=# CLUSTER test_c1 USING idx_test_c1_id;

--查看。
gaussdb=# SELECT * FROM test_c1;
 id | name
----+-------
  1 | Jack
  2 | Scott
  3 | Joe
(3 rows)

--删除。
gaussdb=# DROP TABLE test_c1;
```

- 对已聚簇的表重新进行聚簇排序操作：

```
--建表。
gaussdb=# CREATE TABLE test(col1 int,CONSTRAINT pk_test PRIMARY KEY (col1));

--第一次聚簇排序不带USING关键字报错。
gaussdb=# CLUSTER test;
ERROR:  there is no previously clustered index for table "test"

--聚簇排序。
gaussdb=# CLUSTER test USING pk_test;

--插入数据。
gaussdb=# INSERT INTO test VALUES (1),(99),(10),(8);

--对已聚簇的表重新进行聚簇排序。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CLUSTER VERBOSE test;
INFO:  clustering "public.test" using index scan on "pk_test"(dn_6001 pid=3672)
INFO:  "test": found 0 removable, 4 nonremovable row versions in 1 pages(dn_6001 pid=3672)
DETAIL:  0 dead row versions cannot be removed yet.
CPU 0.00s/0.00u sec elapsed 0.01 sec.
CLUSTER

--删除。
gaussdb=# DROP TABLE test;
```

- 对分区进行聚簇排序操作：

```
--建表并向表中插入数据。
gaussdb=# CREATE TABLE test_c2(id int, info varchar(4)) PARTITION BY RANGE (id)(
    PARTITION p1 VALUES LESS THAN (11),
    PARTITION p2 VALUES LESS THAN (21)
);
gaussdb=# CREATE INDEX idx_test_c2_id1 ON test_c2(id);
gaussdb=# INSERT INTO test_c2 VALUES (6,'ABBB'),(2,'ABAB'),(9,'AAAA');
gaussdb=# INSERT INTO test_c2 VALUES (11,'AAAB'),(19,'BBBA'),(16,'BABA');

--查看。
gaussdb=# SELECT * FROM test_c2;
 id | info
----+------
  6 | ABBB
  2 | ABAB
  9 | AAAA
 11 | AAAB
 19 | BBBA
 16 | BABA
(6 rows)

--对分区p2进行聚簇排序。
gaussdb=# CLUSTER test_c2 PARTITION (p2) USING idx_test_c2_id1;

--查看。
gaussdb=# SELECT * FROM test_c2;
 id | info
----+------
  6 | ABBB
  2 | ABAB
  9 | AAAA
 11 | AAAB
 16 | BABA
 19 | BBBA
(6 rows)

--删除。
gaussdb=# DROP TABLE test_c2;
```

1.13.9.6 COMMENT

功能描述

创建或修改一个对象的注释。

注意事项

- 每个对象仅存储一条注释信息。修改注释时，对同一对象重新执行COMMENT命令；删除注释时，将注释文本设为NULL。对象被删除时，其关联注释将自动清除。
- 注释浏览不具备安全保护机制。连接到数据库的用户均可查看该数据库中所有对象的注释。共享对象（如数据库、角色、表空间）的注释为全局存储，连接到任意数据库的用户均可查询。因此，请勿在注释中存储与安全相关的敏感信息。

版权所有 © 华为云计算技术有限公司

- 对于大多数对象，只有对象的所有者或者被授予了对象COMMENT权限的用户可以执行该命令，系统管理员默认拥有该权限。
- 角色没有所有者，所以COMMENT ON ROLE命令仅可以由系统管理员对系统管理员角色执行，拥有CREATE ROLE权限的角色也可以为非系统管理员角色设置注释。系统管理员可以对所有对象进行注释。

语法格式

```
COMMENT ON
{
  AGGREGATE agg_name (agg_type [, ...] ) |
  CAST (source_type AS target_type) |
  COLLATION object_name |
  COLUMN { table_name.column_name | view_name.column_name } |
  CONSTRAINT constraint_name ON table_name |
  CONVERSION object_name |
  DATABASE object_name |
  EXTENSION object_name |
  FOREIGN DATA WRAPPER object_name |
  FOREIGN TABLE object_name |
  FUNCTION function_name ( [ {[ argname ] [ argmode ] argtype} [, ...] ] ) |
  INDEX object_name |
   OPERATOR operator_name (left_type, right_type) |
  OPERATOR CLASS object_name USING index_method |
  OPERATOR FAMILY object_name USING index_method |
  [ PROCEDURAL ] LANGUAGE object_name |
  ROLE object_name |
  SCHEMA object_name |
  SERVER object_name |
  TABLE object_name |
  TABLESPACE object_name |
  TEXT SEARCH CONFIGURATION object_name |
  TEXT SEARCH DICTIONARY object_name |
  TEXT SEARCH PARSER object_name |
  TEXT SEARCH TEMPLATE object_name |
  TYPE object_name |
  VIEW object_name |
  TRIGGER trigger_name ON table_name
}
   IS 'text';
```

参数说明

- **agg_name**
  聚集函数的名称
- **agg_type**
  聚集函数参数的类型。
- **source_type**
  类型转换的源数据类型。
- **target_type**
  类型转换的目标数据类型。
- **object_name**
  对象名。
- **table_name.column_name | view_name.column_name**
  列名称。前缀可加表名称或者视图名称。
- **constraint_name**
  表约束的名称。

版权所有 © 华为云计算技术有限公司

- **table_name**

  表的名称。

- **function_name**

  函数名称。

- **argmode,argname,argtype**

  函数参数的模式、名称以及类型。

- **operator_name**

  操作符名称。

- **left_type,right_type**

  操作参数的数据类型，可以使用模式进行具体指定。当前置或者后置操作符不存在时，可以增加NONE选项。

- **trigger_name**

  触发器名称。

- **text**

  注释文本内容。

示例

```
--建表。
gaussdb=# CREATE TABLE emp(
    empno varchar(7),
    ename varchar(50),
    job varchar(50),
    mgr varchar(7),
    deptno int
);

--表添加注释。
gaussdb=# COMMENT ON TABLE emp IS '部门表';

--字段添加注释。
gaussdb=# COMMENT ON COLUMN emp.empno  IS  '员工编号';
gaussdb=# COMMENT ON COLUMN emp.ename  IS  '员工姓名';
gaussdb=# COMMENT ON COLUMN emp.job    IS  '职务';
gaussdb=# COMMENT ON COLUMN emp.mgr    IS  '上司编号';
gaussdb=# COMMENT ON COLUMN emp.deptno IS  '部门编号';

--查看表的注释。
gaussdb=# \d+
 Schema |    Name     |   Type   | Owner |    Size    |             Storage              | Description
--------+-------------+----------+-------+------------+----------------------------------+-------------
 public | emp         | table    | omm   | 0 bytes    | {orientation=row,compression=no} | 部门表

--查看字段注释。
gaussdb=# \d+ emp
                            Table "public.emp"
 Column |         Type          | Modifiers | Storage  | Stats target | Description
--------+-----------------------+-----------+----------+--------------+-------------
 empno  | character varying(7)  |           | extended |              | 员工编号
 ename  | character varying(50) |           | extended |              | 员工姓名
 job    | character varying(50) |           | extended |              | 职务
 mgr    | character varying(7)  |           | extended |              | 上司编号
 deptno | integer               |           | plain    |              | 部门编号
Has OIDs: no
Distribute By: HASH(empno)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no
```


版权所有 © 华为云计算技术有限公司

```
--删除表emp。
gaussdb=# DROP TABLE emp;
```

1.13.9.7 COMMIT | END

功能描述

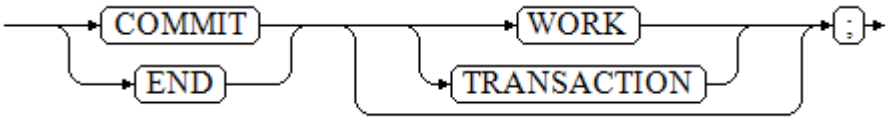
通过该命令可完成提交事务的功能，即提交事务的所有操作。

注意事项

执行COMMIT时，命令执行者必须是该事务的创建者或系统管理员，且创建和提交操作允许在不同会话中执行。

语法格式

```
{ COMMIT | END } [ WORK | TRANSACTION ] ;
```



参数说明

- **COMMIT | END**
  提交当前事务，使当前事务的更改对其他事务可见。
- **WORK | TRANSACTION**
  可选关键字，仅语法兼容无实际含义。

示例

```
--创建Schema。
gaussdb=# CREATE SCHEMA tpcds;

--创建表。
gaussdb=# CREATE TABLE tpcds.customer_demographics_t2
(
   CD_DEMO_SK               INTEGER               NOT NULL,
   CD_GENDER                CHAR(1)                       ,
   CD_MARITAL_STATUS        CHAR(1)                       ,
   CD_EDUCATION_STATUS      CHAR(20)                      ,
   CD_PURCHASE_ESTIMATE     INTEGER                       ,
   CD_CREDIT_RATING         CHAR(10)                      ,
   CD_DEP_COUNT             INTEGER                       ,
   CD_DEP_EMPLOYED_COUNT    INTEGER                       ,
   CD_DEP_COLLEGE_COUNT     INTEGER
) DISTRIBUTE BY HASH (CD_DEMO_SK);

--开启事务。
gaussdb=# START TRANSACTION;

--插入数据。
gaussdb=# INSERT INTO tpcds.customer_demographics_t2 VALUES(1,'M', 'U', 'DOCTOR DEGREE', 1200,
'GOOD', 1, 0, 0);
gaussdb=# INSERT INTO tpcds.customer_demographics_t2 VALUES(2,'F', 'U', 'MASTER DEGREE', 300, 'BAD',
1, 0, 0);

--提交事务，让所有更改永久化。
gaussdb=# COMMIT;
```


版权所有 © 华为云计算技术有限公司

```
--查询数据。
gaussdb=# SELECT * FROM tpcds.customer_demographics_t2;

--删除表tpcds.customer_demographics_t2。
gaussdb=# DROP TABLE tpcds.customer_demographics_t2;

--删除Schema。
gaussdb=# DROP SCHEMA tpcds;
```

相关链接

ROLLBACK

1.13.9.8 COMMIT PREPARED

功能描述

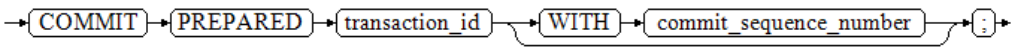
提交一个预先为两阶段提交准备的事务。

注意事项

- 该功能为内部使用功能，不建议用户使用。
- 该功能仅在维护模式（GUC参数xc_maintenance_mode为on时）下可用，该模式仅供维护人员排查问题使用，一般用户不应使用该模式。
- 命令执行者必须是该事务的创建者或系统管理员，且创建和提交操作允许在不同会话中执行。
- 事务功能由数据库自动维护，不应显式使用事务功能。

语法格式

```
COMMIT PREPARED transaction_id [ WITH commit_sequence_number ];
```



参数说明

- **transaction_id**
  待提交事务的标识符。不能和任何当前预备事务已经使用了的标识符同名。
- **commit_sequence_number**
  待提交事务的序列号，为一个64位递增的无符号数。

示例

```
--开始。
gaussdb=# SET XC_MAINTENANCE_MODE = ON;
SET

gaussdb=# BEGIN;

--创建表。
gaussdb=# CREATE TABLE item1(id int);

--准备标识符为的trans_test的事务。
gaussdb=# PREPARE TRANSACTION 'trans_test';
```


版权所有 © 华为云计算技术有限公司

```
--查询表item1，这里应该访问不到。
gaussdb=# SELECT * FROM item1;
ERROR:  Relation "item1" does not exist on test.
LINE 1: SELECT * FROM item1;

--提交标识符为的trans_test的事务。
gaussdb=# COMMIT PREPARED 'trans_test';

--查询表item1，这里应该可以访问。
gaussdb=# SELECT * FROM item1;
 a
---
(0 rows)

--删除表。
gaussdb=# DROP TABLE item1;
```

相关链接

PREPARE TRANSACTION，**ROLLBACK PREPARED**

1.13.9.9 COPY

功能描述

通过COPY命令实现在表和文件之间复制数据。

COPY FROM从一个文件复制数据到一个表，COPY TO把一个表的数据复制到一个文件。

注意事项

- 当参数enable_copy_server_files关闭时，只允许初始用户执行COPY FROM FILENAME或COPY TO FILENAME命令，当参数enable_copy_server_files打开时，允许具有SYSADMIN权限的用户或继承了内置角色gs_role_copy_files权限的用户执行，但默认禁止对数据库配置文件、密钥文件、证书文件、环境变量分离文件和审计日志执行COPY FROM FILENAME或COPY TO FILENAME，以防止用户越权查看或修改敏感文件。
- 数据库管理员可以通过GUC参数safe_data_path设置所有用户可以进行导入导出路径的白名单，白名单默认生效；将此参数设置为空时，不对使用的路径进行拦截，即白名单失效，请谨慎进行设置。该参数会对COPY使用路径中的相对路径进行报错处理。
- COPY只能用于表，不能用于视图。
- COPY TO需要读取的表的SELECT权限，COPY FROM需要插入的表的INSERT权限。
- 如果声明了一个字段列表，COPY将只在文件和表之间复制已声明字段的数据。如果表中有任何不在字段列表里的字段，COPY FROM将为那些字段插入缺省值。
- 如果声明了数据源文件，服务器必须可以访问该文件；如果指定了STDIN，数据将在客户前端和服务器之间流动，输入时，表的列与列之间使用TAB键分隔，在新的一行中以反斜杠和句点（\.）表示输入结束。
- 如果数据文件的任意行包含比预期多或者少的字段，COPY FROM将抛出一个错误。
- 数据的结束可以用一个只包含反斜杠和句点（\.）的行表示。如果从文件中读取数据，数据结束的标记是不必要的；如果在客户端应用之间复制数据，必须要有结束标记。

版权所有 © 华为云计算技术有限公司

- COPY FROM中\N为空字符串，如果要输入实际数据值\N ，使用\\N。
- COPY FROM 支持通过列表达式对数据做预处理，但是列表达式中不支持子查询这类能力。
- COPY FROM在遇到数据格式错误时会回滚事务，但没有足够的错误信息，不方便用户从大量的原始数据中定位错误数据。
- COPY FROM/TO适合低并发，本地小数据量导入导出。
- COPY使用二进制格式时，不支持分布式下转码。
- COPY是服务端命令，执行环境和数据库服务端进程保持一致；\COPY是客户端元命令，执行环境和客户端gsql保持一致。需要注意的是，当在沙箱环境中使用数据库和gsql时，COPY命令和\COPY命令都使用沙箱内的路径；当在沙箱环境中使用数据库，在沙箱外使用gsql时，COPY命令使用沙箱内的路径，\COPY命令则使用沙箱外的路径。
- 在对建有全局二级索引的基表执行COPY数据导入时，需要开启enable_stream_operator参数，以达到最优数据导入性能。
- 在COPY TO导出的过程中，如果关闭GUC参数support_zero_character，表内字段数据存在'\0'字符，则字段数据在导出时会发生截断，字段中只有'\0'之前的数据会被导出。打开参数后，'\0'字符也会被导出，性能损耗与'\0'字符数量正相关。
- pgxc_copy_error_log表的模式从public变更为pg_catalog后，原public模式下的表功能失效。由于public下的表可能与用户表同名，因此需要用户进行识别，确认不是用户表后，将数据迁移到pg_catalog模式下相应的表里之后，删除原public模式下对应的表。由于pg_catalog模式下存在同名表，GSQL元命令\d(+)优先匹配pg_catalog.pgxc_copy_error_log，而因为是系统表的缘故不显示，因此列表中找不到pgxc_copy_error_log，但是实际public模式下存在这张表。可以直接通过public.pgxc_copy_error_log引用。
- 在导出float4/float8类型的数据时，推荐将extra_float_digits设为3，避免有效位数丢失造成导入导出前后数据不一致。
- 部分场景可以通过设置GUC参数batch_insert_index_types="rcr_ubtree"提升批量插入索引性能，具体介绍请参见**INSERT**章节中的**批量插入索引性能优化**。
- 出于安全考虑，禁止向pg_authid、pg_auth_history、gs_global_config和gs_workload_rule系统表中通过COPY FROM导入数据。
- 当参数 enable_security_policy 被打开时，若已设置了脱敏策略，那么系统将会对数据执行脱敏操作。
- 在进行数据导入和导出时，需要注意数据中是否存在转义符的情况。如果数据文件由用户自行准备，用户需要评估导入的方式（希望进行转义导入还是原样导入）。如果数据文件是通过COPY命令导出的，导入时的转义选项应与导出时保持一致。
- 使用COPY命令导入或导出数据时，非转码场景下单行数据元组大小不得超过1GB-1B，转码场景下不得超过256MB-1B。
- COPY命令在导入数据过程中涉及错误表的使用，相关注意事项和使用方法详见对应章节。系统视图包括pg_catalog.pgxc_copy_error_log；系统表包括pg_catalog.gs_copy_error_log。
- 在COPY FROM过程中，数据插入由DN完成，约束类型异常由DN上报给CN。由于DN只能看到自身数据，所以DN上报给CN的报错行号与数据文件中的行号是不对应的。
- 使用\copy命令时，通过detail_info信息作为查询条件查询审计信息时要忽略大小写，SQL语句会被转换，审计中展示的是COPY关键字。

版权所有 © 华为云计算技术有限公司

- 不支持在COPY命令的列表达式中使用返回集合的函数（Set-Returning Functions）。
- 若非法字符入库，COPY导出后无法保证原样导入。

语法格式

- 从一个文件复制数据到一个表。

```
COPY [BINARY] table_name [ ( column_name [, ...] ) ]
    [ WITH OIDS ]
    FROM { 'filename' | STDIN }
    [ LOAD_DISCARD 'discard_file_name' ]
    [ LOAD_BAD 'bad_file_name' ]
    [ USEEOF ]
    [ [ USING ] DELIMITERS 'delimiters' ]
    [ WITHOUT ESCAPING ]
    [ LOG ERRORS | LOG ERRORS DATA ]
    [ REJECT LIMIT 'limit' ]
    [ WITH ]
    [ copy_option [ ...] | ( option [, ...] ) ];
```

- 把一个表的数据复制到一个文件。

```
COPY table_name [ ( column_name [, ...] ) ]
    [ WITH OIDS ]
    TO { 'filename' | STDOUT }
    [ [ USING ] DELIMITERS 'delimiters' ]
    [ WITHOUT ESCAPING ]
    [ WITH ]
    [ copy_option [ ...] | ( option [, ...] ) ];

COPY query {(SELECT) | (VALUES)}
    TO { 'filename' | STDOUT }
    [ WITHOUT ESCAPING ]
    [ WITH ]
    [ copy_option [ ...] | ( option [, ...] ) ];
```

说明

- COPY TO语法形式约束如下：

  (query)与[USING] DELIMITERS不兼容，即若COPY TO的数据来自于一个query的查询结果，那么COPY TO语法不能再指定[USING] DELIMITERS语法子句。
- 对于FIXED FORMATTER语法后面跟随的copy_option是以空格进行分隔的。
- copy_option是指COPY原生的参数形式，而option是兼容外表导入的参数形式。

其中可选参数copy_option子句语法为：

```
{ OIDS
| DELIMITER [ AS ] 'delimiter_string'
| NULL [ AS ] 'null_string'
| HEADER
| FILEHEADER 'header_file_string'
| FREEZE
| FORCE NOT NULL column_name [, ...]
| FORCE QUOTE { column_name [, ...] | * }
| BINARY
| CSV
| FIXED
| QUOTE [ AS ] 'quote_character'
| ESCAPE [ AS ] 'escape_character'
| EOL 'newline_character'
| ENCODING 'encoding_name'
| IGNORE_EXTRA_DATA
| FILL_MISSING_FIELDS [ { 'one' | boolean } ]
| COMPATIBLE_ILLEGAL_CHARS
| DATE_FORMAT 'date_format_string'
| TIME_FORMAT 'time_format_string'
| TIMESTAMP_FORMAT 'timestamp_format_string'
| DATEA_FORMAT 'datea_format_string'
```


版权所有 © 华为云计算技术有限公司

```
| SMALLDATETIME_FORMAT 'smalldatetime_format_string'
| COPY_CUSTOM_ID 'custom_id_string'
| TABLE_COMPRESS_CLAUSE( ROW STORE COMPRESS ADVANCED [ MEDIUM | HIGH ] ROW [ON
(EXPR)])
| FORMATTER ( [ column_name( offset, length ) ] [, ...] )
| TRANSFORM ( [ column_name [ data_type ] [ AS transform_expr ] ] [, ...] )
| POSITION_KEEP_TRAILING_SPACE }
```

其中可选参数option子句语法为：

```
{ FORMAT 'format_name'
| OIDS [ boolean ]
| DELIMITER 'delimiter_character'
| NULL 'null_string'
| HEADER [ boolean ]
| USEEOF [ boolean ]
| FILEHEADER 'header_file_string'
| FREEZE [ boolean ]
| QUOTE 'quote_character'
| ESCAPE 'escape_character'
| EOL 'newline_character'
| NOESCAPING [ boolean ]
| FORCE_QUOTE { ( column_name [, ...] ) | * }
| FORCE_NOT_NULL ( column_name [, ...] )
| ENCODING 'encoding_name'
| IGNORE_EXTRA_DATA [ boolean ]
| FILL_MISSING_FIELDS [ { 'one' | boolean } ]
| COMPATIBLE_ILLEGAL_CHARS [ boolean ]
| DATE_FORMAT 'date_format_string'
| TIME_FORMAT 'time_format_string'
| TIMESTAMP_FORMAT 'timestamp_format_string'
| DATEA_FORMAT 'datea_format_string'
| SMALLDATETIME_FORMAT 'smalldatetime_format_string'
| COPY_CUSTOM_ID 'custom_id_string'
| POSITION_KEEP_TRAILING_SPACE }
```

参数说明

- **query**

  其结果将被复制。

  取值范围：仅支持一个SELECT或VALUES命令，命令结尾不需要分号。

- **table_name**

  表的名称（可以有模式修饰）。

  取值范围：已存在的表名。

- **column_name**

  可选的待复制字段列表。

  取值范围：如果没有声明字段列表，将使用所有字段。

- **STDIN**

  声明输入是来自标准输入。输入时，表的列与列之间使用TAB键分隔，在新的一行中以反斜杠和句点（\.）表示输入结束。

  说明

  在并行导入模式下，若同时通过标准输入（STDIN）导入数据，客户端会以多线程竞争锁的方式进行数据发送。这可能导致部分线程事务数据提交失败，造成数据仅部分导入，且后续无法对失败数据进行重试导入。因此，不建议在表内有数据的情况下，使用标准输入的方式进行并行导入操作。

- **STDOUT**

  声明输出打印到标准输出。

版权所有 © 华为云计算技术有限公司

- **LOAD_DISCARD 'discard_file_name'**

  内部预留接口，不推荐用户直接使用。

- **LOAD_BAD 'bad_file_name'**

  内部预留接口，不推荐用户直接使用。

- **USEEOF**

  当设置参数时，不对导入数据中的\.做报错处理。在对单列表进行导入时，在stdin模式下单独输入"\."，会被视作结束符而不是数据本身。

  取值范围：true/on，false/off。

  缺省值：false

- **[USING] DELIMITERS 'delimiters'**

  在文件中分隔各个字段的字符串，分隔符最大长度不超过10个字节。

  取值范围：

  - 文本模式不允许包含\.abcdefghijklmnopqrstuvwxyz0123456789中的任何一个字符，csv格式无此限制。
  - 分隔符不允许设置为'\r'、'\n'或自定义换行符。
  - 分隔符不能包含null字符串（具体请参见**NULL null_string**），也不能被null字符串包含。
  - 分隔符不能包含\0字符。
  - 分隔符不能为空字符。
  - 在指定COMPATIBLE_ILLEGAL_CHARS（具体请参见**COMPATIBLE_ILLEGAL_CHARS**）时，分隔符不允许设置为' '或'?'，以免和容错转换后的字符冲突。
  - 在CSV模式下，分隔符不能包含QUOTE字符（具体请参见**QUOTE [AS] 'quote_character'**）。
  - 支持多字符分隔符，但分隔符不能超过10个字节。

  缺省值：在文本模式下，缺省是水平制表符，在CSV模式下是一个逗号。

  说明

  - 出于历史原因，DELIMITER和DELIMITERS都可以指定分隔符，但是DELIMITERS后面可以直接接括号语法，DELIMITER不可以直接接括号，会产生报错。
  - 仅支持在TEXT或CSV模式下指定分隔符。
  - 数据文件中单行数据长度需<1GB，如果分隔符较长且数据列较多的情况下，会影响导出有效数据的长度。
  - 分隔符可以使用多字符和不可见字符。多字符例如'$^&'；不可见字符例如0x07，0x08，0x1b等。
  - TEXT格式下导出时，当字段数据中存在与分隔符相同的数据，为保证导出的数据内容能够被正常划分识别，会在冲突的数据前添加'\'进行转义。
  - 使用多字节分隔符进行导出和导入时，不推荐用户将中文字符或其他包含多个字节的字符用作分隔符；这种场景下，当字段内存在与分隔符相同的数据时，可能会出现导出后再导入报错的问题。
  - 使用多字节分隔符时，建议使用ASCII码范围内的单字节字符，拼接组合成为多字节分隔符使用。

- **WITHOUT ESCAPING**

  在TEXT格式中，不对'\'和后面的字符进行转义。

  取值范围：仅支持TEXT格式。

版权所有 © 华为云计算技术有限公司

说明

对于转义符相关选项，需要注意：

- 如果数据文件为用户自己准备的，需要用户自行进行确认其希望的导入行为（转义导入或原样导入）。
- 如果是内核COPY语法导出的数据文件，在导出和导入时的转义选项应该保持一致。
- TEXT格式导出时，默认场景下（不使用WITHOUT ESCAPING），对于字段中的'\b'、'\f'、'\n'、'\r'、'\t'以及'\v'字符，会对其通过'\'+对应字母的形式导出成两个字节。

- **LOG ERRORS**

  若指定，则开启对于COPY FROM语句中数据类型错误的容错机制。

  取值范围：仅支持导入（即COPY FROM）时指定。

  说明

  此容错选项的使用限制如下：

  - 此容错机制仅捕捉COPY FROM过程中数据库主节点上数据解析过程中相关的数据类型错误（DATA_EXCEPTION）。
  - COPY已有的容错选项（如IGNORE_EXTRA_DATA）开启时，对应类型的错误会按照已有的方式处理而不会报出异常，因此错误表也不会有相应数据。
  - 此容错选项默认不支持对约束冲突进行容错。
  - 从低版本滚动升级到该版本时，在所有节点升级完成之前禁止使用COPY容错导入能力。
  - 三权分立下，需要先通过如下语句将错误表创建到用户的同名Schema下，才能使用容错机制，具体请参见**GS_COPY_ERROR_LOG**。

    ```
    CREATE TABLE load_user.pgxc_copy_error_log (relname varchar(64), begintime timestamptz, filename varchar, lineno int8, rawrecord text, detail text);
    CREATE INDEX copy_error_log_relname_idx ON load_user.pgxc_copy_error_log(relname);
    ```

  - 建议将backtrace_min_messages参数值设置到error级别以上，否则会对性能有影响。

- **LOG ERRORS DATA**

  LOG ERRORS DATA和LOG ERRORS的区别:

  a. LOG ERRORS DATA会填充容错表的rawrecord字段。

  b. 只有sysadmin权限的用户才能使用LOG ERRORS DATA参数选项。

  说明

  - 使用“LOG ERRORS DATA”时，若错误内容过于复杂可能存在写入容错表失败的风险，导致任务失败。
  - 对于以某种编码无法读起来的错误，对应ERRCODE_CHARACTER_NOT_IN_REPERTOIRE和ERRCODE_UNTRANSLATABLE_CHARACTER两种错误码，不记录rawrecord字段。
  - 批插场景下，主键约束的错误数据不记录rawrecord字段。
  - 如果使用BINARY格式进行导入，若存在错误数据，容错表中将不记录对应的rawrecord字段。
  - 从低版本滚动升级到该版本时，在所有节点升级完成之前禁止使用COPY容错导入能力。

- **REJECT LIMIT 'limit'**

  与LOG ERRORS选项共同使用，对COPY FROM的容错机制设置数值上限，一旦此COPY FROM语句错误数据超过选项指定条数，则会按照原有机制报错。

  取值范围：正整数[1, 2147483647]，'unlimited'（无最大值限制）

  缺省值：若未指定LOG ERRORS，则会报错；若指定LOG ERRORS，则默认为0。

版权所有 © 华为云计算技术有限公司

说明

如上述LOG ERRORS中描述的容错机制，REJECT LIMIT的计数也是按照执行COPY FROM的数据库主节点上遇到的解析错误数量计算，而不是数据库节点的错误数量。

- **copy_option**

  用于指定COPY原生的各类参数。

  - OIDS

    对于包含OID的表（通常是系统表）指定是否对OID隐藏列进行导入导出。

    取值范围：true/on，false/off。

    缺省值：false

    说明

    - 如果要导出的表没有OID，则直接报错。
    - 导出的OID默认是第一个字段，如果指定HEADER选项，在导出的HEADER行不会有关于OID的字段描述。

  - DELIMITER [AS] 'delimiter_string'

    指定数据文件行数据的字段分隔符，取值与**[USING] DELIMITERS 'delimiters'**相同。

  - NULL [AS] 'null_string'

    用来指定数据文件中空值的表示。

    取值范围：

    - NULL值不能包含'\r'、'\n'或自定义换行符。
    - NULL值最大为100个字符。
    - NULL不能包含分隔符（具体请参见**[USING] DELIMITERS 'delimiters'**），也不能被分隔符包含。
    - 在CSV模式下，NULL不能包含QUOTE字符（具体请参见**QUOTE [AS] 'quote_character'**）。
    - 在指定COMPATIBLE_ILLEGAL_CHARS（具体请参见**COMPATIBLE_ILLEGAL_CHARS**）时，NULL不允许设置为' '或'?'，以免和容错转换后的字符冲突。
    - NULL值不能包含'\0'字符。

    缺省值：

    - CSV格式下默认值是一个没有引号的空字符串。
    - 在TEXT格式下默认值是\N。

    说明

    - 仅支持在TEXT或CSV模式下指定NULL参数。
    - 在使用COPY FROM的时候，任何匹配这个字符串的字符串将被存储为NULL值，所以应该确保指定的字符串和COPY TO相同。

  - HEADER

    导出数据文件是否包含标题行以及标识导入时数据流中第一行是否是标题行。标题行一般用来描述表中每个字段的信息。

版权所有 © 华为云计算技术有限公司

说明

- 仅支持在CSV或者FIXED模式下指定该参数，其中FIXED格式的文件中HEADER行的内容也需要满足列长的定义。
- 带上该选项时表示导入时数据文本第一行会被识别为标题行，会忽略此行。导出时数据文件中第一行为标题行。
- 不带该选项时表示导入时数据文本第一行会被识别为数据，进行导入。导出时数据文件中不会存在标题行。

– FILEHEADER 'header_file_string'

导出数据时用于定义标题行的文件，一般用来描述每一列的数据信息。

须知

- 仅在开启HEADER功能时有效。
- fileheader指定的是绝对路径。
- 该文件只能包含一行标题信息，并以换行符结尾，多余的行将被丢弃（标题信息不能包含换行符）。
- 该文件包括换行符在内长度不超过1M。

– FREEZE

将COPY加载的数据行设置为已经被frozen，就像这些数据行执行过VACUUM FREEZE。

这是一个初始数据加载的性能选项。仅当以下三个条件同时满足时，数据行会被frozen：

- 在同一事务中create或truncate这张表之后执行COPY。
- 当前事务中没有打开的游标。
- 当前事务中没有原有的快照。

说明

COPY完成后，所有其他会话将会立刻看到这些数据。但是这违反了MVCC可见性的一般原则，用户应当了解这样会导致潜在的风险。

– FORCE NOT NULL column_name [, ...]

在CSV COPY FROM模式下，指定的字段不为空。若输入为空，则将视为长度为0的字符串。

取值范围：已存在的字段。

– FORCE QUOTE { column_name [, ...] | * }

在CSV COPY TO模式下，强制在每个声明的字段周围对所有非NULL值都使用封闭符（具体请参见**QUOTE [AS] 'quote_character'**）包围。*代表所有字段，NULL输出不会被封闭符包围。

取值范围：已存在的字段。

– BINARY

使用二进制格式存储和读取，而不是以文本的方式。

版权所有 © 华为云计算技术有限公司

说明

- 在二进制模式下，不能声明DELIMITER、NULL、CSV、EOL选项。
- 指定BINARY类型后，不能再通过option或copy_option指定CSV、FIXED、TEXT等类型。
- 如果设置了GUC参数support_binary_copy_version='header_encoding'，则通过COPY TO导出的二进制文件头中包含数据库的服务端编码信息，COPY FROM导入二进制文件时对编码信息的一致性进行校验，文件中编码信息须与服务端编码保持一致。
- 如果设置了GUC参数copy_special_character_version='no_error'，则会在导入的过程中屏蔽非法字符编码的校验，将非法编码字符按原样导入，查询时以乱码显示。要求数据库服务端编码与客户端编码保持一致。在了解后果的情况下，请谨慎选择开启此参数。
- 二进制模式下copy_special_character_version='no_error'，仅对TEXT、CHAR、VARCHAR、NVARCHAR2、CLOB类型的字段生效。
- 不支持导出到STDOUT或者从STDIN导入。

部分不支持二进制模式导入导出的数据类型如下表所示：

smgr	aclitem	gtsvector	any	trigger
language_handler	internal	opaque	anyelement	anynesttable
anyindexbytable	anynonarray	anyenum	fdw_handler	anyrange
hll_hashval	hash16	hash32	anyset	-

部分不支持二进制模式导入导出的选项如下表所示：

DELIMITER	NULL	EOL	CSV	FIXED
SMALLDATETIME_FORMAT	DATE_FORMAT	TIME_FORMAT	TIMESTAMP_FORMAT	DATEA_FORMAT

- CSV

  打开逗号分隔变量（CSV）模式。指定CSV类型后，不能再通过option或copy_option指定BINARY、FIXED、TEXT等类型。

  说明

  CSV格式最容易出错的是QUOTE包裹符，通常使用的QUOTE是双引号。如果数据字段中存在包裹符，必须做转义处理，否则COPY无法解析错误的数据格式。

- FIXED

  打开字段固定长度模式。在字段固定长度模式下，不能声明DELIMITER，NULL，CSV选项。指定FIXED类型后，不能再通过option或copy_option指定BINARY、CSV、TEXT等类型。定长模式下无法正确处理数据列中的换行符。必须指定FORMATTER。

版权所有 © 华为云计算技术有限公司

说明

- 定长格式定义如下：
  - 每条记录的每个字段长度相同。
  - 长度不足的字段以空格填充，数字类型字段左对齐，定长字符字段左对齐，变长字符字段右对齐。
  - 字段和字段之间没有分隔符。
- 使用定长格式的导入和导出时，不推荐用户进行编码转换操作，因为编码转换会导致字符的字节长度发生变化，导致定长格式的字段截取和解析错误，需要手动调整定长格式下指定的起始位置和终止位置，或使用COPY其他的文件格式。
- 使用FIXED时，需要使用表的全部字段，不支持指定部分字段。

– FORMATTER（ [column_name（ offset, length ）] [, ...] ）

只能用于在固定长度模式中，定义每一个字段在数据文件中的位置。按照column(offset,length)格式定义每一列在数据文件中的位置。

取值范围：

- offset取值不能小于0，以字节为单位。
- length取值不能小于0，以字节为单位。

所有列的总长度和不能大于1GB。

所有列不能相互重叠。

文件中没有出现的列默认以空值代替。

– QUOTE [AS] 'quote_character'

指定CSV格式文件下的封闭符。

取值范围：

- 单字节字符，不能包含'\0'、'\r'、'\n'或行结束符。建议使用不可见字符作为quote，例如0x07，0x08，0x1b等。
- 不能包含分隔符（具体请参见**[USING] DELIMITERS 'delimiters'**），也不能被分隔符包含。
- 不能包含NULL（具体请参见**NULL [AS] 'null_string'**），也不能被NULL包含。
- 在指定COMPATIBLE_ILLEGAL_CHARS（具体请参见**COMPATIBLE_ILLEGAL_CHARS**）时，NULL不允许设置为' '或'?'，以免和容错转换后的字符冲突。

缺省值：双引号 ""。

– ESCAPE [AS] 'escape_character'

指定CSV格式下的逃逸字符。

取值范围：

- 单字节字符，不能包含'\0'。建议使用不可见字符作为ESCAPE，例如0x07，0x08，0x1b等。
- 不能包含分隔符（具体请参见**[USING] DELIMITERS 'delimiters'**），也不能被分隔符包含。
- 不能包含NULL（具体请参见**NULL [AS] 'null_string'**），也不能被NULL包含。

版权所有 © 华为云计算技术有限公司

- 在指定COMPATIBLE_ILLEGAL_CHARS（具体请参见**COMPATIBLE_ILLEGAL_CHARS**）时，NULL不允许设置为' '或'?'，以免和容错转换后的字符冲突。
- 当设置封闭符（具体请参见**QUOTE [AS] 'quote_character'**）而未设置ESCAPE时，ESCAPE与QUOTE字符一致。

缺省值：双引号 '"'。

- EOL 'newline_character'

  指定导入导出数据文件行结束符（换行符）样式。

  取值范围：常见的行结束符，如\r、\n、\r\n（设成0x0D、0x0A、0x0D0A效果是相同的），其他字符或字符串，如$、#。

  - 支持多字符行结束符，但不能超过10个字节。
  - 行结束符符不能被分隔符包含。
  - 行结束符符不能被NULL字符串包含。
  - 非CSV模式下，行结束符不能包含.abcdefghijklmnopqrstuvwxyz0123456789中的字符。
  - 行结束符不能包含'\0'字符。
  - 不支持在指定自定义换行符的场景下进行并行导入。

  说明

  EOL参数只能用于TEXT格式的导入导出，不支持CSV格式和FIXED格式导入。为了兼容原有EOL参数，仍然支持导出CSV格式和FIXED格式时指定EOL参数为0x0A或0x0D0A。

- ENCODING 'encoding_name'

  指定文件编码格式名称。

  取值范围：有效的编码格式，常用的编码格式名称有utf8、gb18030以及gbk。

  缺省值：当前客户端编码格式。

  说明

  - 指定encoding参数后，导入时会将数据文件或标准输入中的数据按照参数指定的编码进行识别，并根据服务端编码的取值判断是否需要进行编码转换。
  - BINARY格式下，encoding参数不生效，服务端编码和客户端编码间的转换依赖server_encoding及client_encoding的设置。
  - encoding指定的字符集，应该和文件的编码格式保持一致，否则可能导致报错或者导入数据乱码。

- IGNORE_EXTRA_DATA

  指定当数据源文件比外表定义列数多时，忽略行尾多出来的列。该参数只在数据导入过程中使用。

  若不使用该参数，在数据源文件比外表定义列数多的情况下，会显示如下错误信息。

  ```
  extra data after last expected column
  ```

- POSITION_KEEP_TRAILING_SPACE

版权所有 © 华为云计算技术有限公司

控制定长格式导入过程中，是否移除字段值尾部的空格。未配置POSITION_KEEP_TRAILING_SPACE时：默认行为为移除尾部空格；配置POSITION_KEEP_TRAILING_SPACE后，将保留字段值尾部的空格。该参数只对字符类型生效，对于数值、blob等类型不生效。

说明

在导入数据的过程中，系统会先从数据文件中读取原始数据，随后依次进行列表达式计算和数据类型转换等处理。

trim尾部空格的行为仅发生在数据读取阶段，该参数仅在此阶段生效。一旦数据读取完成，后续的列表达式处理、类型转换等操作若涉及尾部空格修剪，则不受该参数控制。

例如，当控制文件中将某列定义为char类型，而表中对应列实际为varchar类型时，从char到varchar的转换会自动修剪尾部空格，这一行为不受上述读取参数影响。若希望在此类类型转换场景下保留尾部空格，可通过设置behavior_compat_options参数包含char_coerce_compat选项来实现。

- COMPATIBLE_ILLEGAL_CHARS

  导入导出非法字符容错参数。

  取值范围：true/on，false/off。

  - 参数为true/on，则导入时遇到非法字符进行容错处理，非法字符转换后入库，不报错，不中断导入。导出时将非法字符转换后进行导出，不报错，不中断导出。
  - 参数为false/off，导入时遇到非法字符进行报错，中断导入。导出时遇到非法字符或编码不存在的字符进行报错，中断导出。

  缺省值：false/off

版权所有 © 华为云计算技术有限公司

说明

导入非法字符容错规则如下：

1. 对于'\0'，容错后转换为空格。当support_zero_character打开时则会按照原有模式进行导入。
2. 对于其他非法字符，容错后转换为guc参数convert_illegal_char_mode设置的字符，默认为'?'。
3. 若compatible_illegal_chars为true/on标识，导入时对于非法字符进行容错处理，若NULL、DELIMITER、QUOTE、ESCAPE设置为空格或问号，则会通过如"illegal chars conversion may confuse COPY escape 0x20"等报错信息提示用户修改可能引起混淆的参数以避免导入错误。
4. compatible_illegal_chars为true/on时，不支持对binary格式的数据进行导入。会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。
5. 如果设置了guc参数copy_special_character_version='no_error'，则会在导入的过程中屏蔽非法字符编码的校验，将非法编码字符按原样导入，查询时以乱码显示。在了解后果的情况下，请谨慎选择开启此参数。
6. copy_special_character_version='no_error'时优先级会高于compatible_illegal_chars=true，即非法编码字符会按原样导入。
7. 对于设置了copy_special_character_version='no_error'或compatible_illegal_chars=true的场景，每识别到一行包含非法编码的数据，都会向数据库运行日志中写入一条记录，写日志的行为会占用少量磁盘带宽。如果想要消除此部分的影响，可以配置guc参数enable_log_copy_illegal_chars为off，即不向数据库运行日志中写入记录。除有性能需要，否则不推荐修改此配置。

导出非法字符容错规则如下：

导出时分为需要转码的场景（数据库服务端编码与客户端编码不一致）和不需要转码的场景（数据库服务端编码与客户端编码保持一致），不同场景下导出的表现不同。可以通过show server_encoding;语句查询数据库服务端编码。通过show client_encoding;语句查询数据库客户端编码。

- 需要转码的场景：

1. 对于'\0'，容错后转换为空格。当support_zero_character打开时则会按照原有模式进行导出。
2. 对于其他非法字符，容错后转换为guc参数convert_illegal_char_mode设置的字符，默认为'?'。
3. 若compatible_illegal_chars为true/on标识导出时对于非法字符进行容错处理，若NULL、DELIMITER、QUOTE、ESCAPE设置为空格或问号，则会通过如"illegal chars conversion may confuse COPY escape 0x20"等报错信息提示用户修改可能引起混淆的参数，以避免导出错误。
4. compatible_illegal_chars为true/on时，不支持对binary格式的数据进行导出。
5. 对于设置了compatible_illegal_chars=true的场景，每识别到一行包含非法编码的数据，都会向数据库运行日志中写入一条记录，写日志的行为会占用少量磁盘带宽。如果想要消除此部分的影响，可以配置guc参数enable_log_copy_illegal_chars为off，即不向数据库运行日志中写入记录。除有性能需要，否则不推荐修改此配置。

- 不需要转码的场景：

1. 数据会按照数据库内原样进行导出，即使数据中存在非法字符并设置compatible_illegal_chars=true。
2. compatible_illegal_chars为true/on时，不支持对binary格式的数据进行导出。

– FILL_MISSING_FIELDS [{ 'one' | boolean }]

当数据加载时，指定当数据行的最后一个或多个字段缺失时的处理方式。当仅指定FILL_MISSING_FIELDS而未显示指定选项值，或者指定的选项值为one/true/on，则最后一个字段缺失时用null补齐（多个字段缺失时会报错）。

版权所有 © 华为云计算技术有限公司

取值范围：one，true/on，false/off。

缺省值：false/off。

- DATE_FORMAT 'date_format_string'

  导入时对于DATE类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法DATE格式。具体请参见**时间和日期处理函数和操作符**。

- TIME_FORMAT 'time_format_string'

  导入时对于TIME类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法TIME格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

- TIMESTAMP_FORMAT 'timestamp_format_string'

  导入时对于TIMESTAMP类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法TIMESTAMP格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

- DATEA_FORMAT 'datea_format_string'

  导入时对于DATEA类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法DATEA格式。具体请参见**时间和日期处理函数和操作符**。

- SMALLDATETIME_FORMAT 'smalldatetime_format_string'

  导入时对于SMALLDATETIME类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法SMALLDATETIME格式。具体请参见**时间和日期处理函数和操作符**。

- TABLE_COMPRESS_CLAUSE(ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW [ON (EXPR)])

  数据导入时，对导入数据进行压缩。EXPR为表达式，满足表达式的数据会被压缩，不满足表达式的数据不被压缩。此参数仅对COPY FROM导入有效。

  说明

  - 仅适用于Astore表。
  - TOAST数据不支持压缩。
  - 升级观察期不支持该功能。
  - 使用该功能时，对于导入数据的存储使用新数据页。
  - 该功能与容错机制（LOG ERRORS或LOG ERRORS DATA）共用时，如果错误数据频繁出现，可能会导致压缩无收益。
  - 不支持单行导入时压缩（例如：导入分区表并且开启了违法约束不回滚功能；表上带触发器等）。
  - 不支持与逻辑解码同时使用，如果开启逻辑解码，则COPY导入压缩会直接报错。

版权所有 © 华为云计算技术有限公司

- TRANSFORM ({ column_name [data_type] [AS transform_expr] } [, ...])

  指定表中各个列的转换表达式；其中data_type指定该列在表达式参数中的数据类型；transform_expr为目标表达式，返回与表中目标列数据类型一致的结果值，表达式可参考**表达式**。

  说明

  COPY FROM不支持对分布列指定转换表达式。

- COPY_CUSTOM_ID 'custom_id_string'

  数据导入时，输入用户指定的自定义字符串，用作本次copy导入的标识。

  说明

  - 非三权分立下，pg_catalog.pgxc_copy_error_log中默认包含custom_id字段，不使用copy_custom_id会向字段内插入NULL。
  - 三权分立下，如果当前连接用户的同名schema下存在pgxc_copy_error_log表，表上存在custom_id字段，并使用了LOG ERRORS或LOG ERRORS DATA语法，且容错数据量小于reject limit指定的值，则会将此参数输入的字符串记录到pgxc_copy_error_log表的custom_id字段上，一次导入可能会对应0条或多条数据。
  - 只对导入过程生效，导出的过程不会记录错误表以及日志表。
  - 用户自定义输入的字符串长度不能超过64字节。
  - 当错误表和日志表上不存在custom_id字段时，又传入了copy_custom_id参数，会报错提示用户通过函数或者ALTER语句在表上增加字段。
  - 错误表和日志表上的custom_id字段不保证唯一性，字段内容可以重复。
  - 本参数仅支持在ORA兼容模式下使用。

- **option**

  用于指定兼容外表的各类参数。

  - FORMAT 'format_name'

    数据源文件的格式。

    取值范围：CSV、TEXT、BINARY。

    - CSV格式文件：可以有效处理数据列中的换行符，但对一些特殊字符处理有欠缺。等价于copy_option中的**CSV**参数。
    - TEXT格式文件：可以有效处理一些特殊字符，但无法正确处理数据列中的换行符。
    - BINARY格式文件：使得所有的数据被存储/读作二进制格式而不是文本。 这比TEXT和CSV格式的要快一些，但是一个BINARY格式文件可移植性比较差。等价于copy_option中的**BINARY**参数。

    缺省值：TEXT

  - OIDS [boolean]

    对于包含OID的表（通常是系统表）指定是否对OID隐藏列进行导入导出。

    取值范围：true/on，false/off。

    缺省值：false

  - DELIMITER 'delimiter_character'

    指定数据文件行数据的字段分隔符，取值与**[USING] DELIMITERS 'delimiters'**一致。

版权所有 © 华为云计算技术有限公司

- NULL 'null_string'

  用来指定数据文件中空值的表示。取值与**NULL null_string**一致。

- HEADER [boolean]

  指定导出数据文件是否包含标题行以及标识导入时数据流中第一行是否是标题行。标题行一般用来描述表中每个字段的信息。

  取值范围：true/on，false/off。

  - 参数为true/on，则导入时数据文本第一行会被识别为标题行，会忽略此行。导出时数据文件中第一行为标题行。
  - 参数为false/off，则导入时数据文本第一行会被识别为数据，进行导入。导出时数据文件中不会存在标题行。

  缺省值：false

  说明

  仅支持在CSV或者FIXED模式下指定该参数，其中FIXED格式的文件中HEADER行的内容也需要满足列长的定义。

- FILEHEADER 'header_file_string'

  导出数据时用于定义标题行的文件，一般用来描述每一列的数据信息。等价于copy_option中的**FILEHEADER 'header_file_string'**参数。

- FREEZE [boolean]

  导出数据时用于定义标题行的文件，一般用来描述每一列的数据信息。

  取值范围：true/on，false/off。取值true或on时，等价于copy_option中的**FREEZE**参数。

  缺省值：false

- NOESCAPING [boolean]

  在TEXT格式中，不对'\'和后面的字符进行转义。

  取值范围：true/on，false/off。取值true或on时，等价于copy_option中的**WITHOUT ESCAPING**参数。

  缺省值：false

- SKIP int_number

  指定数据导入时，跳过数据文件的前 int_number 行。

  取值范围：整数[0, 2147483647]。

- USEEOF [boolean]

  当设置参数或取值为true/on时，不对导入数据中的”\.”做报错处理。在对单列表进行导入时，在stdin模式下单独输入"\."，会被视作结束符而不是数据本身。

  取值范围：true/on，false/off。取值true或on时，等价于**USEEOF**参数。

  缺省值：false

- QUOTE 'quote_character'

  指定CSV格式文件下的引号字符。等价于copy_option中的**QUOTE [AS] 'quote_character'**参数。

- ESCAPE 'escape_character'

  指定CSV格式文件下的逃逸字符。等价于copy_option中的**ESCAPE [AS] 'escape_character'**参数。

版权所有 © 华为云计算技术有限公司

- EOL 'newline_character'

  指定导入导出数据文件换行符样式。取值与**EOL 'newline_character'**一致。

- FORCE_QUOTE { (column_name [, ...]) | * }

  在CSV COPY TO模式下，强制在每个声明的字段周围对所有非NULL值都使用QUOTE字符（具体请参见**QUOTE [AS] 'quote_character'**）包围。

  取值范围：已存在的字段。

  说明

  - *代表所有字段。
  - NULL输出不会被引号包围，若原数据与NULL选项内容一致时，则会将原数据进行QUOTE。

- FORCE_NOT_NULL (column_name [, ...])

  在CSV COPY FROM模式下，指定的字段在数据写入时若发现为NULL，则写入NULL（具体请参见**NULL [AS] 'null_string'**）指定的内容来代替。即该列不允许直接写入NULL。

  取值范围：已存在的字段。

- ENCODING 'encoding_name'

  指定数据文件的编码格式名称，缺省为当前客户端编码格式。encoding指定的字符集，应该和文件的编码格式保持一致，否则可能导致报错或者导入数据乱码。

- IGNORE_EXTRA_DATA [boolean]

  若数据源文件比外表定义列数多，是否会忽略对多出的列。该参数只在数据导入过程中使用。

  取值范围：true/on、false/off。

  - 参数为true/on，若数据源文件比外表定义列数多，则忽略行尾多出来的列。
  - 参数为false/off，若数据源文件比外表定义列数多，会显示如下错误信息。

    ```
    extra data after last expected column
    ```

  缺省值：false。

  须知

  如果行尾换行符丢失，使两行变成一行时，设置此参数为true将导致后一行数据被忽略掉。

- POSITION_KEEP_TRAILING_SPACE

  控制定长格式导入过程中，是否移除字段值尾部的空格。等价于copy_option中的**POSITION_KEEP_TRAILING_SPACE**参数。

- COMPATIBLE_ILLEGAL_CHARS [boolean]

  导入导出非法字符容错参数。

  取值范围：true/on，false/off。

  - 参数为true/on，则导入时遇到非法字符进行容错处理，非法字符转换后入库，不报错，不中断导入。导出时将非法字符转换后进行导出，不报错，不中断导出。

版权所有 © 华为云计算技术有限公司

- 参数为false/off，导入时遇到非法字符进行报错，中断导入。导出时遇到非法字符或编码不存在的字符进行报错，中断导出。

缺省值：false/off

说明

导入非法字符容错规则如下：

1. 对于'\0'，容错后转换为空格。当support_zero_character打开时则会按照原有模式进行导入。
2. 对于其他非法字符，容错后转换为guc参数convert_illegal_char_mode设置的字符，默认为'?'。
3. 若compatible_illegal_chars为true/on标识导入时对于非法字符进行容错处理，若NULL、DELIMITER、QUOTE、ESCAPE设置为空格或问号，则会通过如"illegal chars conversion may confuse COPY escape 0x20"等报错信息提示用户修改可能引起混淆的参数以避免导入错误。
4. compatible_illegal_chars为true/on时，不支持对binary格式的数据进行导入，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。
5. 如果设置了guc参数copy_special_character_version='no_error'，则会在导入的过程中屏蔽非法字符编码的校验，将非法编码字符按原样导入，查询时以乱码显示。在了解后果的情况下，请谨慎选择开启此参数。
6. copy_special_character_version='no_error'时优先级会高于compatible_illegal_chars=true，即非法编码字符会按原样导入。
7. 对于设置了copy_special_character_version='no_error'或compatible_illegal_chars=true的场景，每识别到一行包含非法编码的数据，都会向数据库运行日志中写入一条记录，写日志的行为会占用少量磁盘带宽。如果想要消除此部分的影响，可以配置guc参数enable_log_copy_illegal_chars为off，即不向数据库运行日志中写入记录。除有性能需要，否则不推荐修改此配置。

导出非法字符容错规则如下：

导出时分为需要转码的场景（数据库服务端编码与客户端编码不一致）和不需要转码的场景（数据库服务端编码与客户端编码保持一致），不同场景下导出的表现不同。可以通过show server_encoding;语句查询数据库服务端编码。通过show client_encoding;语句查询数据库客户端编码。

- 需要转码的场景：

1. 对于'\0'，容错后转换为空格。当support_zero_character打开时则会按照原有模式进行导出。
2. 对于其他非法字符，容错后转换为guc参数convert_illegal_char_mode设置的字符，默认为'?'。
3. 若compatible_illegal_chars为true/on标识导出时对于非法字符进行容错处理，若NULL、DELIMITER、QUOTE、ESCAPE设置为空格或问号，则会通过如"illegal chars conversion may confuse COPY escape 0x20"等报错信息提示用户修改可能引起混淆的参数，以避免导出错误。
4. compatible_illegal_chars为true/on时，不支持对binary格式的数据进行导出。
5. 对于设置了compatible_illegal_chars=true的场景，每识别到一行包含非法编码的数据，都会向数据库运行日志中写入一条记录，写日志的行为会占用少量磁盘带宽。如果想要消除此部分的影响，可以配置guc参数enable_log_copy_illegal_chars为off，即不向数据库运行日志中写入记录。除有性能需要，否则不推荐修改此配置。

- 不需要转码的场景：

1. 数据会按照数据库内原样进行导出，即使数据中存在非法字符并设置compatible_illegal_chars=true。
2. compatible_illegal_chars为true/on时，不支持对binary格式的数据进行导出。

– FILL_MISSING_FIELDS [{ 'one' | boolean }]

版权所有 © 华为云计算技术有限公司

当数据加载时，若数据源文件中一行的最后一个字段缺失的处理方式。

取值范围：one，true/on，false/off。

缺省值：false/off

- COPY_CUSTOM_ID 'custom_id_string'

  数据导入时，输入用户指定的自定义字符串，用作本次copy导入的标识。等价于copy_option中的**COPY_CUSTOM_ID 'custom_id_string'**参数。

- DATE_FORMAT 'date_format_string'

  导入对于DATE类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法DATE格式。具体请参见**时间和日期处理函数和操作符**。

- TIME_FORMAT 'time_format_string'

  导入对于TIME类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法TIME格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

- TIMESTAMP_FORMAT 'timestamp_format_string'

  导入对于TIMESTAMP类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法TIMESTAMP格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

- SMALLDATETIME_FORMAT 'smalldatetime_format_string'

  导入对于SMALLDATETIME类型指定格式。此参数不支持BINARY格式，会报“cannot specify bulkload compatibility options in BINARY mode”错误信息。此参数仅对COPY FROM导入有效。

  取值范围：合法SMALLDATETIME格式。具体请参见**时间和日期处理函数和操作符**。

- DATEA_FORMAT 'datea_format_string'

  导入对于DATEA类型（仅在ORA模式下支持）指定格式。此参数不支持BINARY格式，会报错“cannot specify bulkload compatibility options in BINARY mode”。此参数仅对COPY FROM导入有效。

  取值范围：合法DATEAFORMAT格式。具体请参见**时间和日期处理函数和操作符**。

版权所有 © 华为云计算技术有限公司

说明

COPY FROM能够识别的特殊反斜杠序列如下所示。

- **\b**：反斜杠（ASCII 8）。
- **\f**：换页（ASCII 12）。
- **\n**：换行符（ASCII 10）。
- **\r**：回车符（ASCII 13）。
- **\t**：水平制表符（ASCII 9）。
- **\v**：垂直制表符（ASCII 11）。
- **\digits**：反斜杠后面跟着一到三个八进制数，表示ASCII值为该数的字符。
- **\xdigits**：反斜杠x后面跟着一个或两个十六进制位声明指定数值编码的字符。

权限控制示例

```
gaussdb=> copy t1 from '/home/xy/t1.csv';
ERROR:  COPY to or from a file is prohibited for security concerns
HINT:  Anyone can COPY to stdout or from stdin. gsql's \copy command also works for anyone.
gaussdb=> grant gs_role_copy_files to xxx;
```

此错误为非初始用户没有使用COPY的权限示例，解决方式为打开GUC参数enable_copy_server_files，则管理员可以使用COPY功能，普通用户需要在此基础上加入gs_role_copy_files群组。

示例

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA tpcds;

--创建tpcds.ship_mode表。
gaussdb=# CREATE TABLE tpcds.ship_mode
(
    SM_SHIP_MODE_SK           INTEGER               NOT NULL,
    SM_SHIP_MODE_ID           CHAR(16)              NOT NULL,
    SM_TYPE                   CHAR(30)                      ,
    SM_CODE                   CHAR(10)                      ,
    SM_CARRIER                CHAR(20)                      ,
    SM_CONTRACT               CHAR(20)
)
;

--向tpcds.ship_mode表插入一条数据。
gaussdb=# INSERT INTO tpcds.ship_mode VALUES (1,'a','b','c','d','e');

--将tpcds.ship_mode中的数据复制到/home/omm/ds_ship_mode.dat文件中。
gaussdb=# COPY tpcds.ship_mode TO '/home/omm/ds_ship_mode.dat';

--将tpcds.ship_mode 输出到STDOUT。
gaussdb=# COPY tpcds.ship_mode TO STDOUT;

--将tpcds.ship_mode 的数据输出到STDOUT，使用参数如下：分隔符为','(delimiter ',')，编码格式为UTF8(encoding 'utf8')。
gaussdb=# COPY tpcds.ship_mode TO STDOUT WITH (delimiter ',', encoding 'utf8');

--将tpcds.ship_mode 的数据输出到STDOUT，使用参数如下：导入格式为CSV（format 'CSV'），引号包围SM_SHIP_MODE_SK字段的导出内容(force_quote(SM_SHIP_MODE_SK))。
gaussdb=# COPY tpcds.ship_mode TO STDOUT WITH (format 'CSV', force_quote(SM_SHIP_MODE_SK));

--创建tpcds.ship_mode_t1表。
gaussdb=# CREATE TABLE tpcds.ship_mode_t1
(
    SM_SHIP_MODE_SK           INTEGER               NOT NULL,
    SM_SHIP_MODE_ID           CHAR(16)              NOT NULL,
```


版权所有 © 华为云计算技术有限公司

```
    SM_TYPE               CHAR(30)                ,
    SM_CODE               CHAR(10)                ,
    SM_CARRIER            CHAR(20)                ,
    SM_CONTRACT           CHAR(20)
)
;

--从STDIN复制数据到表tpcds.ship_mode_t1。
gaussdb=# COPY tpcds.ship_mode_t1 FROM STDIN;

--输入一行数据示例
--gaussdb=# COPY tpcds.ship_mode_t1 FROM STDIN;
--Enter data to be copied followed by a newline.
--End with a backslash and a period on a line by itself.
-->> 1      a    a    a    a    a
-->> \.
-- 备注：数据与数据之间输入tab

--从/home/omm/ds_ship_mode.dat文件复制数据到表tpcds.ship_mode_t1。
gaussdb=# COPY tpcds.ship_mode_t1 FROM '/home/omm/ds_ship_mode.dat';

--从/home/omm/ds_ship_mode.dat文件复制数据到表tpcds.ship_mode_t1，应用TRANSFORM表达式转换，取SM_TYPE列左边10个字符插入到表中。
gaussdb=# COPY tpcds.ship_mode_t1 FROM '/home/omm/ds_ship_mode.dat' TRANSFORM (SM_TYPE AS LEFT(SM_TYPE, 10));

--从/home/omm/ds_ship_mode.dat文件复制数据到表tpcds.ship_mode_t1，使用参数如下：导入格式为TEXT（format 'text'），分隔符为'\t'（delimiter E'\t'），忽略多余列（ignore_extra_data 'true'），不指定转义（noescaping 'true'）。
gaussdb=# COPY tpcds.ship_mode_t1 FROM '/home/omm/ds_ship_mode.dat' WITH(format 'text', delimiter E'\t', ignore_extra_data 'true', noescaping 'true');

--从/home/omm/ds_ship_mode_fixed.dat文件复制数据到表tpcds.ship_mode_t1，使用参数如下：导入格式为FIXED（FIXED），指定定长格式（FORMATTER(SM_SHIP_MODE_SK(0, 2), SM_SHIP_MODE_ID(2,16), SM_TYPE(18,30), SM_CODE(50,10), SM_CARRIER(61,20), SM_CONTRACT(82,20)) ），忽略多余列（ignore_extra_data），有数据头（header）。
gaussdb=# COPY tpcds.ship_mode TO '/home/omm/ds_ship_mode_fixed.dat' FIXED FORMATTER(SM_SHIP_MODE_SK(0, 2), SM_SHIP_MODE_ID(2,16), SM_TYPE(18,30), SM_CODE(50,10), SM_CARRIER(61,20), SM_CONTRACT(82,20)) header;
gaussdb=# COPY tpcds.ship_mode_t1 FROM '/home/omm/ds_ship_mode_fixed.dat' FIXED FORMATTER(SM_SHIP_MODE_SK(0, 2), SM_SHIP_MODE_ID(2,16), SM_TYPE(18,30), SM_CODE(50,10), SM_CARRIER(61,20), SM_CONTRACT(82,20)) header ignore_extra_data;

--删除表和SCHEMA。
gaussdb=# DROP TABLE tpcds.ship_mode;
gaussdb=# DROP TABLE tpcds.ship_mode_t1;
gaussdb=# DROP SCHEMA tpcds;
```

1.13.9.10 CREATE AGGREGATE

功能描述

创建一个新的聚集函数。

语法格式

```
CREATE AGGREGATE name ( input_data_type [ , ... ] ) (
    SFUNC = sfunc,
    -- SFUNC1 = sfunc,
    STYPE = state_data_type
    -- STYPE1 = state_data_type,
    [ , FINALFUNC = ffunc ]
    [ , INITCOND = initial_condition ]
    -- [ , INITCOND1 = initial_condition ]
    [ , SORTOP = sort_operator ]
    [ , CFUNC = collection_func ]
    [ , INITCOLLECT = initial_collection_condition ]
```

 版权所有 © 华为云计算技术有限公司

```
    [ , IFUNC = init_func ]
    [ , SHIPPABLE = is_shippable ]
);
```

原语法/较早的语法格式：

```
CREATE AGGREGATE name (
    BASETYPE = base_type,
    SFUNC = sfunc,
    -- SFUNC1 = sfunc,
    STYPE = state_data_type
    -- STYPE1 = state_data_type,
    [ , FINALFUNC = ffunc ]
    [ , INITCOND = initial_condition ]
    -- [ , INITCOND1 = initial_condition ]
    [ , SORTOP = sort_operator ]
    [ , CFUNC = collection_func ]
    [ , INITCOLLECT = initial_collection_condition ]
    [ , IFUNC = init_func ]
);
```

参数说明

- **name**

  要创建的聚集函数名，可以使用模式进行具体指定。

- **input_data_type**

  该聚集函数需要处理的输入数据类型。

  需要创建一个无入参的聚集函数，可以使用*代替输入数据类型列表，count(*)即为零参数的聚集函数的一个实例。

- **base_type**

  直接通过指定BASETYPE关键字的方式定义聚集函数需要处理的输入数据类型，而并非直接将输入类型写在聚集函数名称后面（即通过指定**input_data_type**的方式）。

  当通过该方式进行定义聚集函数需要处理的输入数据类型时，聚集函数应为具有一个输入参数的聚集函数，当创建的聚集函数为一个零参数聚集函数，请将该关键字指定为ANY（而并非*）。

- **sfunc**

  将在每一个输入行上调用的状态转换函数的名称。对于有N个参数的聚集函数，sfunc必须有N+1个参数，其中的第一个参数类型为state_data_type，其余的参数类型匹配已声明的输入数据类型。函数必须返回一个state_data_type类型的值。这个函数接受当前状态值和当前输入数据，并返回下个状态值。

  Oracle数据库的转换函数默认行为是STRICT，即会跳过NULL输入值。GaussDB需要用户自行定义转换函数STRICT属性。

- **state_data_type**

  聚集的状态值的数据类型。

- **ffunc**

  在转换完所有输入行后调用的最终处理函数，计算聚集的结果。此函数必须接受一个类型为state_data_type的参数。聚集的输出数据类型被定义为此函数的返回类型。如果没有声明ffunc，则使用聚集结果的状态值作为聚集的结果，且输出类型为state_data_type。

- **initial_condition**

  状态值的初始设置（值）。initial_condition必须为一个state_data_type类型可以接受的文本常量值。如果没有声明，状态值的初始设置（值）为null。

版权所有 © 华为云计算技术有限公司

- **sort_operator**

  用于MIN或MAX类型聚集的排序操作符。仅表示一个操作符名，可以使用模式进行具体指定。

- **collection_func**

  collection_func本质上是组合不同DATANODE产生的状态转换结果。如果没有最终函数，collection_func产生的结果即为聚集的结果。

  如果需要collection_func跳过null值，请将collection_func标记为STRICT。

  collection_func必须有两个类型为state_data_type的入参，第一个入参state_data_type将传递给下一次collection_func执行的第一个参数。如果is_shippable为false（即聚集函数不支持下推行为），则不会执行collection_func。

- **initial_collection_condition**

  collection_func状态值的初始设置（值）。initial_collection_condition必须是一个state_data_type类型可以接受的文本常量值。如果没有声明，状态值的初始设置（值）为null。

- **init_func**

  sfunc状态值的初始设置函数。不能返回null值，必须返回state_data_type类型并且没有入参。如果同时设置init_func和initial_condition，init_func的返回值将会优先作为初始状态值。

- **is_shippable**

  标记该聚集函数是否能支持下推执行，仅可以取值true或者false。目前该参数不影响系统内置聚集函数的下推行为，仅影响用户自定义聚集函数的行为，默认值为false。仅当存在collection_func且is_shippable为true时，用户自定义聚集函数可以下推。

  说明

  如果聚集函数支持下推执行，那么聚集函数不能引用PACKAGE的内容（包括变量、函数等），否则可能会在执行中报错。如sfunc、cfunc、finalfunc以及ifunc不能是PACKAGE函数，或者sfunc、finalfunc以及ifunc内部也不能引用PACKAGE变量、调用PACKAGE函数等。

示例

```
--创建自定义函数。
gaussdb=# CREATE OR REPLACE FUNCTION int_add(int,int)
 returns int as $BODY$
declare
begin
 return $1 + $2;
end;
$BODY$ language plpgsql;

--创建聚集函数。
gaussdb=# CREATE AGGREGATE sum_add(int)
(
    sfunc = int_add,

    stype = int,

    initcond = '0'

);

--创建测试表和添加数据。
gaussdb=# CREATE TABLE test_sum(a int,b int,c int);
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO test_sum VALUES(1,2),(2,3),(3,4),(4,5);

--执行聚集函数。
gaussdb=# SELECT sum_add(a) FROM test_sum;
  sum_add
-----------
  10

--删除聚集函数。
gaussdb=# DROP AGGREGATE sum_add(int);

--删除自定义函数。
gaussdb=# DROP FUNCTION int_add(int,int);

--删除测试表。
gaussdb=# DROP TABLE test_sum;
```

相关链接

ALTER AGGREGATE，**DROP AGGREGATE**

1.13.9.11 CREATE AUDIT POLICY

功能描述

创建审计策略。

注意事项

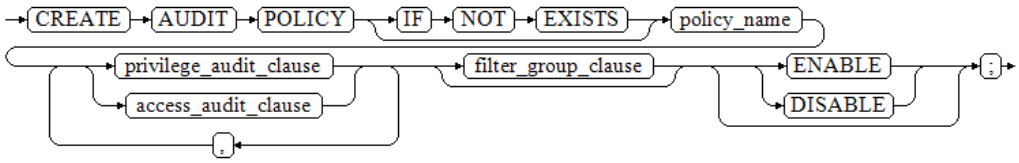
- 审计策略的创建与维护存在权限限制，只有POLADMIN、SYSADMIN或初始用户有权限执行此命令。
- 在创建审计策略之前，需要确保已经开启安全策略开关，即设置GUC参数enable_security_policy为on后，审计策略才会生效。
- 系统管理员或安全策略管理员可以访问GS_AUDITING_POLICY、GS_AUDITING_POLICY_ACCESS、GS_AUDITING_POLICY_PRIVILEGES和GS_AUDITING_POLICY_FILTERS系统表，查询已创建的审计策略。

须知

在使用DATABASE LINK功能的场景下，客户端发起的DATABASE LINK请求，实际的发送方是服务端，发送端IP地址等相关的属性将是服务端的值。具体的使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

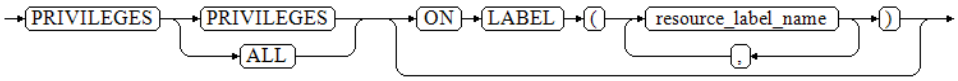
语法格式

```
CREATE AUDIT POLICY [ IF NOT EXISTS ] policy_name { { privilege_audit_clause | access_audit_clause }
[, ... ] [ filter_group_clause ] [ ENABLE | DISABLE ] };
```

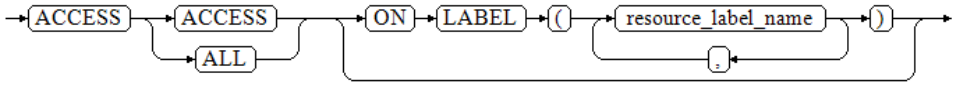


- privilege_audit_clause：
  ```
  PRIVILEGES { PRIVILEGES | ALL } [ ON LABEL ( resource_label_name [, ... ] ) ]
  ```

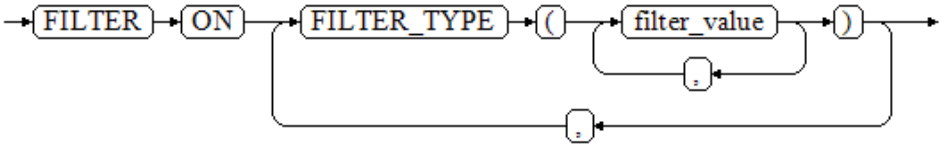

版权所有 © 华为云计算技术有限公司



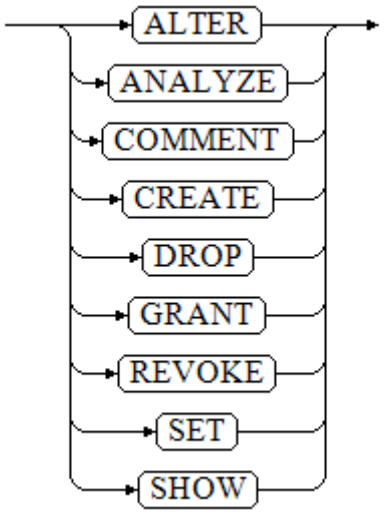
- access_audit_clause：
  ACCESS { ACCESS | ALL } [ON LABEL (resource_label_name [, ...])]



- filter_group_clause：
  FILTER ON { FILTER_TYPE (filter_value [, ...]) } [, ...]

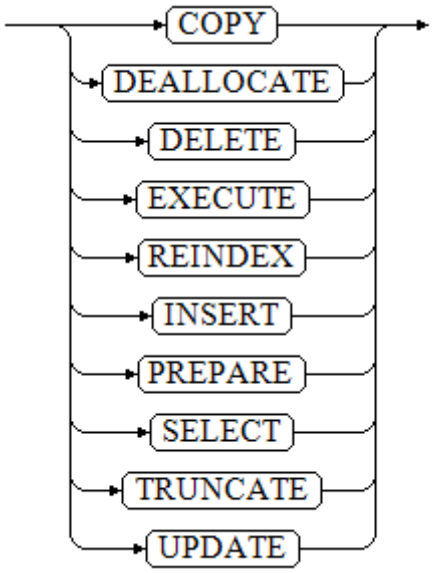


- PRIVILEGES：
  { ALTER | ANALYZE | COMMENT | CREATE | DROP | GRANT | REVOKE | SET | SHOW }

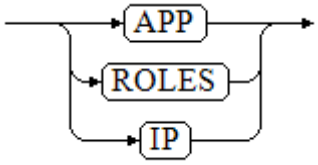


- ACCESS：
  { COPY | DEALLOCATE | DELETE | EXECUTE | REINDEX | INSERT | PREPARE | SELECT | TRUNCATE | UPDATE }

版权所有 © 华为云计算技术有限公司



- FILTER_TYPE：
  { APP | ROLES | IP }



参数说明

- **IF NOT EXISTS**

  可以使用该关键字进行检查指定的审计策略是否存在，以避免重复创建审计策略。

- **policy_name**

  审计策略的名称，应具有唯一性，避免与现有策略发生冲突。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  - 审计策略名称若超过63个字符，数据库会截断并保留前63个字符当作审计策略名称。
  - 当审计策略名称中包含大写字母时，数据库会自动转换为小写字母，如果需要创建包含大写字母的审计策略名称则需要使用双引号进行包裹。
  - 审计策略名称需要为小写字母（a~z）、大写字母（A~Z）、下划线（_）、数字（0~9）或美元符号（$），且必须以字母或下划线开头。

- **resource_label_name**

  资源标签名称。

- **PRIVILEGES**

  针对数据库执行如下操作时进行审计，目前支持CREATE、ALTER、DROP、ANALYZE、COMMENT、GRANT、REVOKE、SET以及SHOW操作。

  当取值为ANALYZE时，进行ANALYZE和VACUUM操作都会被审计。

- **ACCESS**

版权所有 © 华为云计算技术有限公司

针对数据库执行如下操作时进行审计，目前支持SELECT、COPY、DEALLOCATE、DELETE、EXECUTE、INSERT、PREPARE、REINDEX、TRUNCATE以及UPDATE操作。

- **ALL**

  PRIVILEGES或ACCESS中支持的所有对数据库的操作。当形式为{ PRIVILEGES | ALL }时，ALL指所有PRIVILEGES操作；当形式为{ ACCESS | ALL }时，ALL指所有ACCESS操作。

- **FILTER_TYPE**

  描述审计策略过滤的条件类型，包括APP、ROLES或IP。

- **filter_value**

  具体审计策略过滤信息内容。

- **ENABLE | DISABLE**

  打开或关闭统一审计策略。若不指定，默认为ENABLE。

示例

- 创建一个对数据库执行CREATE的审计策略。

```
--创建adt1策略。
gaussdb=# CREATE AUDIT POLICY adt1 PRIVILEGES CREATE;

--查看adt1策略。
gaussdb=# SELECT * FROM GS_AUDITING_POLICY;
 polname | polcomments |          modifydate          | polenabled
---------+-------------+------------------------------+------------
 adt1    |             | 2023-11-06 16:41:40.947417 | t

--查看审计日志的存放位置。
gaussdb=# SHOW audit_directory;

--删除审计策略adt1。
gaussdb=# DROP AUDIT POLICY adt1;
```

- 创建一个审计策略，仅审计用户dev_audit进行CREATE操作。

```
--创建dev_audit用户。
gaussdb=# CREATE USER dev_audit PASSWORD '********';

--创建一个表tb_for_audit。
gaussdb=# CREATE TABLE tb_for_audit(col1 text, col2 text, col3 text);

--创建基于tb_for_audit表的adt_lb0资源标签。
gaussdb=# CREATE RESOURCE LABEL adt_lb0 add TABLE(public.tb_for_audit);

--创建针对adt_lb0资源进行create操作的adt2审计策略。
gaussdb=# CREATE AUDIT POLICY adt2 PRIVILEGES CREATE ON LABEL(adt_lb0) FILTER ON
ROLES(dev_audit);

--删除审计策略adt2。
gaussdb=# DROP AUDIT POLICY adt2;

--删除表tb_for_audit。
gaussdb=# DROP TABLE tb_for_audit;

--删除dev_audit用户。
gaussdb=# DROP USER dev_audit;
```

- 创建一个仅审计记录用户dev_audit，客户端工具为gsql，IP地址为'10.20.30.40'和'127.0.0.0/24'，在执行针对adt_lb0资源进行的SELECT、INSERT、DELETE操作数据库的审计策略。

```
--创建dev_audit用户。
gaussdb=# CREATE USER dev_audit PASSWORD '********';
```


版权所有 © 华为云计算技术有限公司

```
--创建审计策略adt3。
gaussdb=# CREATE AUDIT POLICY adt3 ACCESS SELECT ON LABEL(adt_lb0), INSERT ON
LABEL(adt_lb0), DELETE FILTER ON ROLES(dev_audit), APP(gsql), IP('10.20.30.40', '127.0.0.0/24');

--删除审计策略adt3。
gaussdb=# DROP AUDIT POLICY adt3;

--删除dev_audit用户。
gaussdb=# DROP USER dev_audit;
```

相关链接

ALTER AUDIT POLICY，**DROP AUDIT POLICY**

1.13.9.12 CREATE BARRIER

功能描述

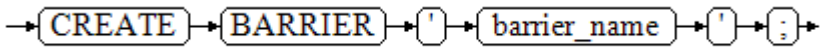
创建一个新集群节点间的同步点，该同步点可用于进行数据恢复。

注意事项

- 该功能为内部使用功能，不建议用户直接使用。
- 该功能通常只用于备份恢复操作，因此只允许在以下场景下可以执行：
  - 数据库初始化用户可执行。
  - 在CN节点上开启备份恢复模式，即GUC参数operation_mode为on的前提下，且具备OPRADMIN权限的用户可执行。

语法格式

```
CREATE BARRIER 'barrier_name';
```



参数说明

barrier_name

同步点的名称。

取值范围：字符串，要符合**标识符命名规范**。

示例

```
--指定barrier名称。
gaussdb=# CREATE BARRIER 'barrier1';
```

1.13.9.13 CREATE CLIENT MASTER KEY

功能描述

创建主密钥对象（CMK）。

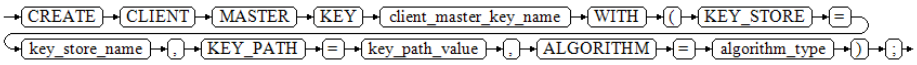
版权所有 © 华为云计算技术有限公司

注意事项

- 该语法属于全密态数据库特有语法。
- 在使用该语法前，请参见《开发指南》中“特性使用指导 > 设置密态等值查询”章节，在数据库驱动侧，为外部密钥管理者设置环境变量。
- 连接数据库时，在数据库驱动侧，需开启密态等值查询特性连接参数，才可执行CREATE CLIENT MASTER KEY命令。密态等值查询特性使用多级加密模型，主密钥加密列密钥，列密钥加密数据。
- 主密钥由外部密钥管理者提供，该语法仅处理密钥来源、密钥ID等信息，已支持的外部密钥管理者包括：
  - 华为云密钥管理服务huawei_kms。
  - 用户密码派生密钥或直接提供密钥user_token。
  - 基于鲲鹏TEE的内生密钥管理和三未信安SJJ1994加密机sdf_kms。

语法格式

```
CREATE CLIENT MASTER KEY client_master_key_name WITH (KEY_STORE = key_store_name, KEY_PATH = key_path_value, ALGORITHM = algorithm_type);
```



参数说明

- **client_master_key_name**

  新创建的主密钥对象名，同一命名空间下，CMK名需满足命名唯一性约束。

  取值范围：字符串，需符合**标识符命名规范**。

- **KEY_STORE**

  外部密钥管理者。

  取值范围：如**表1-393**所示。
  - 当KEY_STORE为user_token时，无需用户提供KEY_PATH参数。
  - 当KEY_STORE为sdf_kms时，无需用户提供ALGORITHM参数，默认支持SGD_SM4_ECB。

- **KEY_PATH**

  由外部密钥管理者管理的密钥，不同密钥管理者格式不同。

  取值范围：字符串，如**表1-393**所示。KEY_PATH内容使用单引号或双引号进行包裹，如果字符串长度超过64，则只能使用单引号包裹。

- **ALGORITHM**

  密钥采用的加密算法。

  取值范围：如**表1-393**所示。

版权所有 © 华为云计算技术有限公司

表 1-393 针对不同密钥管理者的参数值

KEY_STORE	KEY_PATH	ALGORITHM
huawei_kms	格式：‘{KmsApiUrl}/{密钥ID}' 参考：'https://kms.{项目}.myhuaweicloud.com/v1.0/{项目ID}/kms/{密钥ID}' 示例：'https://kms.cn-north-4.myhuaweicloud.com/v1.0/00000000000000000000000000000000/kms/00000000-0000-0000-0000-000000000000'	AES_256 SM4
user_token	不需用户提供KEY_PATH。	当前支持的加密算法为： ● AES_256_CBC ● AES_256_GCM ● SM4 ● SM4_HMAC_SM3 ● SM4_CTR_HMAC_SM3
sdf_kms	格式：'{密钥加密密钥（KEK）的ID}' 注意：该参数可调用iTrustee提供的密钥管理工具生成，或者由三未信安SJJ1994加密机中密钥管理工具生成。	不需用户提供ALGORITHM

示例

- user_token场景：

```
--解压GaussDB-Kernel_数据库版本号_操作系统版本号_64bit_Gsql.tar.gz安装包，找到脚本gsql_env.sh。

--通过脚本，自动配置环境变量。
source gsql_env.sh

--连接数据库，使用特权账号，创建新用户例如alice。
gsql -p 端口号 -d postgres -r
gaussdb=# CREATE USER alice PASSWORD '*******';
gaussdb=# \q

--连接数据库，务必使用"-C"参数。
gsql -p 端口号 -d postgres -U alice -r -C

--设置用户密码或派生密钥，gsql可以通过password=stdin或key_token=stdin进行交互式输入。
--设置用户密码，密码长度不小于8，包含大、小写字母、数字、符号的其中3种。
gaussdb=> \key_info keyType=user_token,password=*********
--或者直接通过对接满足安全强度的密钥，此处密钥为16进制编码。
gaussdb=> \key_info keyType=user_token,key_token=********************************

--创建主密钥。
gaussdb=> CREATE CLIENT MASTER KEY alice_cmk WITH ( KEY_STORE = user_token , ALGORITHM = AES_256_GCM );
```


版权所有 © 华为云计算技术有限公司

```
--删除主密钥。
gaussdb=> DROP CLIENT MASTER KEY alice_cmk;
gaussdb=> \q

--连接数据库，使用特权账号，删除用户alice。
gsql -p 端口号 -d postgres -r
gaussdb=# DROP USER alice;
```

相关链接

DROP CLIENT MASTER KEY

1.13.9.14 CREATE COLUMN ENCRYPTION KEY

功能描述

创建一个列加密密钥对象（CEK），该密钥可用于加密表中的指定列。

注意事项

- 该语法属于全密态数据库特有语法。
- 当使用gsql连接数据库服务器时，需使用“-C”参数，打开全密态数据库的开关，才能使用CREATE COLUMN ENCRYPTION KEY命令。
- 在定义表中列字段时，可指定CEK对象，用于加密该列。

语法格式

```
CREATE COLUMN ENCRYPTION KEY column_encryption_key_name WITH VALUES(CLIENT_MASTER_KEY = client_master_key_name, ALGORITHM = algorithm_type [, ENCRYPTED_VALUE = encrypted_value]);
```

参数说明

- **column_encryption_key_name**

  新创建的列密钥对象名。在同一命名空间下，CEK名需满足命名唯一性约束。由该语法创建的列密钥对象可用于列级加密。

  取值范围：字符串，要符合**标识符命名规范**。

- **CLIENT_MASTER_KEY**

  客户端主密钥，用于加密指定的列密钥对象。

  取值范围：CMK对象名，该CMK对象由**CREATE CLIENT MASTER KEY**语法创建。

- **ALGORITHM**

  密钥采用的加密算法。

  取值范围：AEAD_AES_256_CBC_HMAC_SHA256、AEAD_AES_128_CBC_HMAC_SHA256、AEAD_AES_256_CTR_HMAC_SHA256、AES_256_GCM、SM4_SM3、SM4_HMAC_SM3或SM4_CTR_HMAC_SM3。

  - 不同加密算法的数据膨胀率不同，具体为AEAD_AES_256_CTR_HMAC_SHA256 = SM4_CTR_HMAC_SM3 < AES_256_GCM < AEAD_AES_256_CBC_HMAC_SHA256 = AEAD_AES_128_CBC_HMAC_SHA256 = SM4_SM3 = SM4_HMAC_SM3，推荐使用AEAD_AES_256_CTR_HMAC_SHA256、SM4_CTR_HMAC_SM3或AES_256_GCM加密算法。

版权所有 © 华为云计算技术有限公司

- **ENCRYPTED_VALUE**

  可选，用户自定义的密钥，密钥长度范围为28个 ~ 256个字符。

  - 28个字符派生的密钥，其安全强度符合AES-128。
  - 如需使用AES-256，密钥长度应为39个字符。
  - 如果不指定，将自动生成256个字符的密钥。
  - 当使用Huawei KMS生成的主密钥对列密钥进行加密时，通过ENCRYPTED_VALUE字段传入的密钥长度须为16字节的整数倍。

须知

国密算法约束：由于SM2、SM3、SM4等算法属于中国国家密码标准算法，为规避法律风险，需配套使用。如果创建主密钥时指定SM4、SM4_HMAC_SM3、SM4_CTR_HMAC_SM3算法进行加密列密钥，则创建列密钥时必须指定SM4_SM3、SM4_HMAC_SM3、SM4_CTR_HMAC_SM3算法进行加密数据。使用sdf_kms密钥管理时，不受该约束限制。

相关链接

ALTER COLUMN ENCRYPTION KEY，DROP COLUMN ENCRYPTION KEY

1.13.9.15 CREATE CONVERSION

功能描述

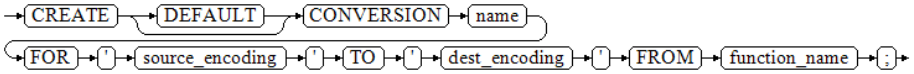
创建新的编码转换规则，将数据从一种字符编码（源编码）转换为另一种字符编码（目标编码）。

注意事项

- 该功能为内部使用功能，用户无法直接使用。编码转换规则仅支持系统内部创建。

语法格式

```
CREATE [ DEFAULT ] CONVERSION name
    FOR 'source_encoding' TO 'dest_encoding' FROM function_name;
```



参数说明

- **DEFAULT**

  将该转换设置为默认转换。客户端与服务器之间默认执行源编码到目标编码之间的转换。

  同一模式中，每个编码对仅存在一个默认转换。

- **name**

  转换规则的名称。同一个模式中，转换规则名称必须唯一。

版权所有 © 华为云计算技术有限公司

- **source_encoding**

  源字符编码格式。

- **dest_encoding**

  目标字符编码格式。

- **function_name**

  执行实际编码转换的函数。

  该函数必须具有以下格式：

```
conv_proc(
    integer, -- 源编码ID
    integer, -- 目标编码ID
    cstring, -- 源字符串（空值终止的C字符串）
    internal,-- 目标（用一个空值终止的C字符串填充）
    integer  -- 源字符串长度
) RETURNS void;
```

1.13.9.16 CREATE DATABASE

功能描述

创建新数据库。默认情况下，新数据库将通过复制标准系统数据库template0模板的方式进行创建。

注意事项

- 仅拥有CREATEDB权限的用户可以执行该命令，系统管理员默认拥有此权限。
- 不能在事务块中执行该命令。
- 创建数据库时，禁止使用templatem作为数据库名称。
- 在创建数据库时，若未指定DBCOMPATIBILITY关键字和TEMPLATE关键字时，默认兼容的数据库的类型以GUC参数**distribute_default_compatibility**的取值为准。
- 在创建兼容模式的数据库时，当创建的数据库为非ORA兼容模式、非M兼容模式以及非MYSQL兼容模式时，以下系统对象中的部分信息存在差异，具体如**表1-394**所示。

表 1-394 系统对象差异

系统对象类型	系统对象名称	差异
系统视图	pg_catalog.my_tab_modifications	创建数据库时，视图中timestamp字段为timestamp(0) without time zone类型。 视图升级后，该字段类型会变成date类型。
	pg_catalog.db_tab_modifications	
	pg_catalog.v$open_cursor	创建数据库时，视图中last_sql_active_time字段为timestamp(0) without time zone类型。 视图升级后，该字段类型会变成date类型。

版权所有 © 华为云计算技术有限公司

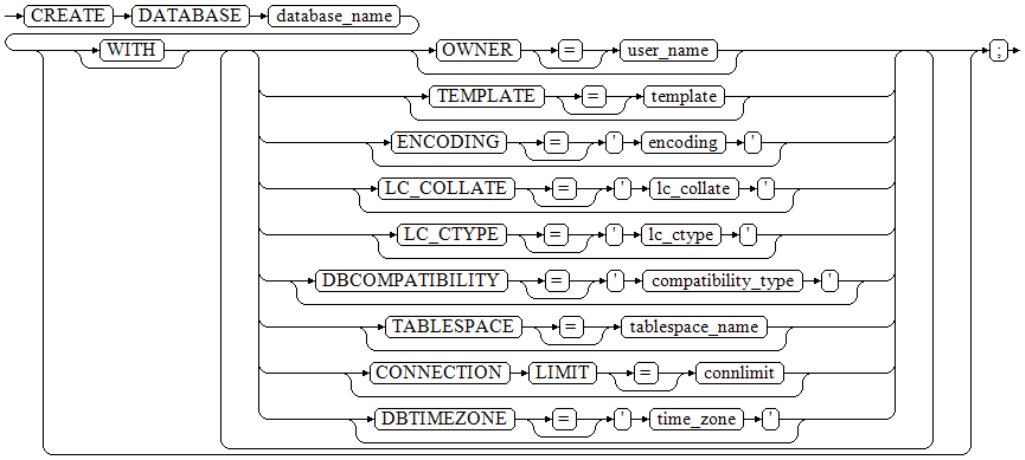
系统对象类型	系统对象名称	差异
	pg_catalog.my_audit_trail	创建数据库时，视图中timestamp、logoff_time字段为timestamp(0) without time zone类型。 视图升级后，该字段类型会变成date类型。
	sys.v_session sys.gv_session	创建数据库时，视图中logon_time、prev_exec_start字段为timestamp(0) without time zone类型。 视图升级后，该字段类型会变成date类型。
	sys.db_indexes sys.adm_indexes sys.adm_tab_cols sys.db_tab_columns sys.adm_tab_columns sys.db_tab_col_statistics sys.adm_tab_col_statistics sys.db_part_col_statistics sys.adm_part_col_statistics sys.db_subpart_col_statistics sys.adm_subpart_col_statistics	创建数据库时，视图中last_analyzed字段为timestamp(0) without time zone类型。 视图升级后，该字段类型会变成date类型。
高级包函数	dbe_scheduler.set_attribute	创建数据库时，缺少dbe_scheduler.set_attribute(name text, attribute text, value date)重载函数。 packages插件升级后，存在dbe_scheduler.set_attribute(name text, attribute text, value date)重载函数。

语法格式

```
CREATE DATABASE database_name
   [ [ WITH ] { [ OWNER [=] user_name ] |
            [ TEMPLATE [=] template ] |
            [ ENCODING [=] 'encoding' ] |
```


版权所有 © 华为云计算技术有限公司

```
        [ LC_COLLATE [=] 'lc_collate' ] |
        [ LC_CTYPE [=] 'lc_ctype' ] |
        [ DBCOMPATIBILITY [=] 'compatibility_type' ] |
        [ TABLESPACE [=] tablespace_name ] |
        [ CONNECTION LIMIT [=] connlimit ] |
        [ DBTIMEZONE [=] 'time_zone' ] }[...] ];
```



参数说明

- **database_name**

  数据库名称。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  - 当数据库名称超过63个字符，数据库会截断并保留前63个字符当作数据库名称。
  - 当数据库名称中包含大写字母时，数据库会自动转换为小写字母，如果需要创建包含大写字母的数据库名称则需要使用双引号进行包裹。

- **OWNER [=] user_name**

  可选，数据库所有者。

  取值范围：已存在的用户名，未指定时，新数据库的所有者为当前用户。

- **TEMPLATE [=] template**

  可选，创建数据库时，根据指定的标准系统数据库模板进行创建。

  GaussDB采用从模板数据库复制的方式进行创建新的数据库。GaussDB包含模板数据库（template0、template1、templatea和templatem）以及默认的用户数据库（postgres）。

  取值范围：template0、templatem或者templatea。

- **ENCODING [=] 'encoding'**

  可选，指定数据库使用的字符编码。

  常用取值：GBK、UTF8、Latin1或者GB18030等，具体支持的字符编码如**表1-395**所示。

  - 不指定字符编码时，M-Compatibility模式数据库默认使用UTF8编码，其他兼容模式数据库默认使用模板数据库的编码。
  - 模板数据库template0、template1和templatea的默认编码与操作系统环境相关。
  - 模板数据库template1不允许修改字符编码，若要变更编码，请使用模板数据库template0或模板数据库templatea创建数据库。

版权所有 © 华为云计算技术有限公司

表 1-395 GaussDB 支持的字符编码

名称	描述	语言	是否服务器端?	ICU（International Components for Unicode）?	字节/字符	别名
BIG5	Big Five	繁体中文	否	否	1-2	WIN950、Windows950
EUC_CN	扩展UNIX编码-中国	简体中文	是	是	1-3	-
EUC_JP	扩展UNIX编码-日本	日文	是	是	1-3	-
EUC_JIS_2004	扩展UNIX编码-日本, JIS X 0213	日文	是	否	1-3	-
EUC_KR	扩展UNIX编码-韩国	韩文	是	是	1-3	-
EUC_TW	扩展UNIX编码-中国台湾	繁体中文	是	是	1-3	-
GB18030	国家标准	中文	是	否	1-4	-
GB18030_2022	国家标准	中文	是	否	1-4	-
GBK	扩展国家标准	简体中文	是	否	1-2	WIN936、Windows936
ISO_8859_5	ISO 8859-5, ECMA 113	拉丁语/西里尔语	是	是	1	-

版权所有 © 华为云计算技术有限公司

名称	描述	语言	是否服务器端?	ICU（International Components for Unicode）?	字节/字符	别名
ISO_8859_6	ISO 8859-6, ECMA 114	拉丁语/阿拉伯语	是	是	1	-
ISO_8859_7	ISO 8859-7, ECMA 118	拉丁语/希腊语	是	是	1	-
ISO_8859_8	ISO 8859-8, ECMA 121	拉丁语/希伯来语	是	是	1	-
JOHAB	JOHAB	韩语	否	否	1-3	-
KOI8R	KOI8-R	西里尔语（俄语）	是	是	1	KOI8
KOI8U	KOI8-U	西里尔语（乌克兰语）	是	是	1	-
LATIN1	ISO 8859-1, ECMA 94	西欧	是	是	1	ISO88591
LATIN2	ISO 8859-2, ECMA 94	中欧	是	是	1	ISO88592
LATIN3	ISO 8859-3, ECMA 94	南欧	是	是	1	ISO88593
LATIN4	ISO 8859-4, ECMA 94	北欧	是	是	1	ISO88594

版权所有 © 华为云计算技术有限公司

名称	描述	语言	是否服务器端?	ICU（International Components for Unicode）?	字节/字符	别名
LATIN5	ISO 8859-9, ECMA 128	土耳其语	是	是	1	ISO88599
LATIN6	ISO 8859-10, ECMA 144	日耳曼语	是	是	1	ISO885910
LATIN7	ISO 8859-13	波罗的海	是	是	1	ISO885913
LATIN8	ISO 8859-14	凯尔特语	是	是	1	ISO885914
LATIN9	ISO 8859-15	带欧罗巴和口音的LATIN1	是	是	1	ISO885915
LATIN10	ISO 8859-16, ASRO SR 14111	罗马尼亚语	是	否	1	ISO885916
MULE_INTERNAL	Mule内部编码	多语种编辑器	是	否	1-4	-
SJIS	Shift JIS	日语	否	否	1-2	Mskanji、ShiftJIS、WIN932、Windows932
SHIFT_JIS_2004	Shift JIS, JIS X 0213	日语	否	否	1-2	-
SQL_ASCII	未指定（见文本）	*任意*	是	否	1	-

版权所有 © 华为云计算技术有限公司

名称	描述	语言	是否服务器端?	ICU（International Components for Unicode）?	字节/字符	别名
UHC	统一韩语编码	韩语	否	否	1-2	WIN949, Windows949
UTF8	Unicode, 8-bit	*所有*	是	是	1-4	Unicode
WIN866	Windows CP866	西里尔语	是	是	1	ALT
WIN874	Windows CP874	泰语	是	否	1	-
WIN1250	Windows CP1250	中欧	是	是	1	-
WIN1251	Windows CP1251	西里尔语	是	是	1	WIN
WIN1252	Windows CP1252	西欧	是	是	1	-
WIN1253	Windows CP1253	希腊语	是	是	1	-
WIN1254	Windows CP1254	土耳其语	是	是	1	-
WIN1255	Windows CP1255	希伯来语	是	是	1	-
WIN1256	Windows CP1256	阿拉伯语	是	是	1	-
WIN1257	Windows CP1257	波罗的海	是	是	1	-

版权所有 © 华为云计算技术有限公司

名称	描述	语言	是否服务器端?	ICU（International Components for Unicode）?	字节/字符	别名
WIN1258	Windows CP1258	越南语	是	是	1	ABC, TCVN, TCVN5712, VSCII
ZHS16GBK	扩展国家标准	简体中文	是	否	1-2	-

须知

- 并非所有的客户端API都支持**表1-395**中列出的字符编码。
- SQL_ASCII编码格式的设置与其他编码格式的设置表现存在差异。如果服务器字符编码格式为SQL_ASCII时，服务器会将字节值0~127根据ASCII标准解析，而字节值128~255则当作无法解析的字符。通常如果使用任何非ASCII数据，请勿将字符编码格式设置为SQL_ASCII，如果设置为SQL_ASCII，数据库将无法转换或者校验非ASCII字符。
- 新指定的数据库字符编码必须与所选择的本地环境中（LC_COLLATE和LC_CTYPE）的设置兼容。
- 当指定的字符编码为GBK或ZHS16GBK时，部分中文生僻字无法直接作为对象名。主要因为GBK第二个字节的编码范围在0x40~0x7E之间时，字节编码与ASCII字符@、A~Z、[、\、]、^、_、`、a~z、{、|以及}重叠。其中，@、[、\、]、^、_、'、{、|以及}为数据库中的操作符，直接作为对象名时，会产生语法报错。
- 若客户端编码和服务器端编码不一致时，数据库需要支持编码格式转换，支持的所有的编码格式转换可通过系统表PG_CONVERSION进行查看。若数据库无法进行编码格式转换，则建议客户端编码与服务器端编码保持一致，客户端编码可通过GUC参数client_encoding修改。
- 当指定数据库字符集编码与客户端编码同时为GB18030_2022时，须确保客户端操作系统支持的GB18030字符集为2022版本，否则由于GB18030字符集各版本间存在不完全兼容可能导致数据的不一致性。同时，当涉及历史数据切换为GB18030_2022数据库时，应当遵循切换数据库流程进行数据迁移操作。
- 当指定的字符编码为ZHS16GBK时，支持欧元符号（€）的输入，其编码值为0x80。

- **LC_COLLATE [=] 'lc_collate'**

  可选，指定新数据库使用的字符集。

  取值范围：操作系统支持的字符集，默认使用模板数据库的字符集。

版权所有 © 华为云计算技术有限公司

须知

- 该参数的使用会影响到对字符串的排序顺序，如执行ORDER BY操作的结果排序或在文本列上使用索引的排序。
- 该参数对M-Compatibility模式数据库不生效。
- lc_collate和lc_ctype参数的取值范围，与本地环境支持的字符集相关，如在Linux操作系统上，可通过locale -a命令获取操作系统支持的字符集列表，当指定LC_COLLATE和LC_CTYPE关键字时，从获取的字符集列表中选择用户需要的字符集和字符分类。
- 如果需要设置字符编码为GB18030_2022时，lc_collate参数和lc_ctype参数的取值范围与GB18030字符编码保持一致。
- 如果需要设置字符编码为ZHS16GBK时，lc_collate参数和lc_ctype参数的取值范围需与GBK字符编码保持一致。

- **LC_CTYPE [=] 'lc_ctype'**

  可选，指定新数据库使用的字符分类。

  取值范围：操作系统支持的字符分类，默认使用模板数据库的字符分类。

须知

- 该参数的使用会影响到字符的分类，如大写、小写和数字。
- 该参数对M-Compatibility模式数据库不生效。
- lc_collate和lc_ctype参数的取值范围，与本地环境支持的字符集相关，如在Linux操作系统上，可通过locale -a命令获取操作系统支持的字符集列表，当指定LC_COLLATE和LC_CTYPE关键字时，从获取的字符集列表中选择用户需要的字符集和字符分类。
- 如果需要设置字符编码为GB18030_2022时，lc_collate参数和lc_ctype参数的取值范围与GB18030字符编码保持一致。
- 如果需要设置字符编码为ZHS16GBK时，lc_collate参数和lc_ctype参数的取值范围需与GBK字符编码保持一致。

- **DBCOMPATIBILITY [=] 'compatibility_type'**

  可选，指定兼容的数据库的类型。未指定该选项时，默认兼容模式受TEMPLATE关键字以及GUC参数distribute_default_compatibility的设置影响。

  取值范围：MYSQL、TD、ORA、PG和M。分别表示：MySQL兼容性MYSQL模式、TD（Teradata）、Oracle、PostgreSQL和MySQL兼容性M-Compatibility模式。

版权所有 © 华为云计算技术有限公司

须知

- ORA兼容模式数据库下，将空字符串作为NULL处理。当GUC参数mapping_date_to_datea关闭时，数据类型DATE会被替换为TIMESTAMP(0) WITHOUT TIME ZONE，当GUC参数mapping_date_to_datea打开时，数据类型DATE会被替换为DATEA。
- 将字符串类型转换成整数类型时，如果输入不合法，MYSQL兼容模式数据库会将输入转换为0，而其他兼容模式数据库下则会报错。
- PG兼容模式数据库下，CHAR和VARCHAR以字符为计数单位，其他兼容模式数据库下以字节为计数单位。例如，UTF-8字符编码，CHAR(3)在PG兼容性下能存放3个中文字符，而在其他兼容模式数据库下只能存放1个中文字符。
- 执行语句报错时，部分报错信息中显示“A-format”或“B-format”，表示“ORA-format”和“MYSQL-format”。
- M-Compatibility模式数据库下，具体的数据库特性请参见《M-Compatibility开发指南》。

- **TABLESPACE [=] tablespace_name**

  可选，指定数据库对应的表空间。

  取值范围：已存在表空间名。

- **CONNECTION LIMIT [=] connlimit**

  可选，数据库可以接受的并发连接数。

  取值范围：[-1, 2^{31}-1]的整数。默认值为-1，表示没有限制。

须知

- 系统管理员不受此参数的限制。
- 每个节点的connlimit单独统计，集群整体的连接数 = connlimit * 当前正常CN节点个数。

- **DBTIMEZONE [=] 'time_zone'**

  指定新数据库的时区。该参数会影响新数据库的时区，默认设置为PRC时区。

  取值范围：

  - 操作系统支持的时区名称和缩写。
  - -15:59~+15:00。

  说明

  - 该参数仅在ORA兼容模式数据库下且已设置a_format_version='10c'以及a_format_dev_version='s2'时支持使用。
  - 有关字符编码的一些限制：
    - 若区域设置为C或POSIX，则允许所有的编码类型；对于其他的区域设置，字符编码必须和区域设置相同。
    - 字符编码和区域设置必须匹配模板数据库（除将template0作为模板），当数据库中包含不匹配指定编码的数据或者包含排序顺序受LC_COLLATE和LC_CTYPE影响的索引，复制数据时可能会存在新数据库中的索引失效的场景。

版权所有 © 华为云计算技术有限公司

示例

```
--创建jim用户。
gaussdb=# CREATE USER jim PASSWORD '********';

--创建一个GBK编码的数据库testdb1。
gaussdb=# CREATE DATABASE testdb1 ENCODING 'GBK' template = template0;

--查看数据库testdb1信息。
gaussdb=# SELECT datname,pg_encoding_to_char(encoding) FROM pg_database WHERE datname =
'testdb1';
 datname | pg_encoding_to_char
---------+---------------------
 testdb1 | GBK
(1 row)

--创建兼容ORA格式的数据库testdb2，并指定所有者为jim。
gaussdb=# CREATE DATABASE testdb2 OWNER jim DBCOMPATIBILITY = 'ORA';

--查看testdb2信息。
gaussdb=# SELECT t1.datname,t2.usename,t1.datcompatibility
          FROM pg_database t1,gs_user t2
          WHERE t1.datname = 'testdb2' AND t1.datdba=t2.usesysid;
 datname | usename | datcompatibility
---------+---------+------------------
 testdb2 | jim     | ORA
(1 row)
gaussdb=# CREATE DATABASE testdb_tmp OWNER jim template=templatea;

--查看testdb2信息。
gaussdb=# SELECT t1.datname,t2.usename,t1.datcompatibility
                   FROM pg_database t1,gs_user t2
                   WHERE t1.datname = 'testdb_tmp' AND t1.datdba=t2.usesysid;
  datname   | usename | datcompatibility
------------+---------+------------------
 testdb_tmp | jim     | ORA
(1 row)

--创建可以接受1000并发连接数的数据库。
gaussdb=# CREATE DATABASE testdb4 CONNECTION LIMIT=1000;

--查看testdb4信息。
gaussdb=# SELECT datname,datconnlimit  FROM pg_database where datname = 'testdb4';
 datname | datconnlimit
---------+--------------
 testdb4 |         1000
(1 row)

--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_test RELATIVE LOCATION 'test_tablespace/tbs_test';

--创建数据库指定默认表空间为tbs_test。
gaussdb=# CREATE DATABASE testdb5 TABLESPACE = tbs_test;

--查看testdb5信息。
gaussdb=# SELECT db.datname,ts.spcname FROM pg_database db JOIN pg_tablespace ts ON
db.dattablespace = ts.oid WHERE db.datname = 'testdb5';
 datname | spcname
---------+----------
 testdb5 | tbs_test
(1 row)

--切换至ORA兼容性数据库testdb2，并设置会话参数。
gaussdb=# \c testdb2
testdb2=# SET a_format_version='10c';
testdb2=# SET a_format_dev_version='s2';

--创建兼容ORA格式的数据库并指定时区。
testdb2=# CREATE DATABASE testdb3 DBCOMPATIBILITY 'ORA' DBTIMEZONE='+08:00';
```


版权所有 © 华为云计算技术有限公司

```
--查看testdb3信息。
testdb2=# SELECT datname,datcompatibility,dattimezone FROM pg_database WHERE datname = 'testdb3';
 datname | datcompatibility | dattimezone
---------+------------------+-------------
 testdb3 | ORA              | +08:00
(1 row)

--切换至初始数据库，请根据实际情况修改数据库名。
testdb2=# \c postgres

--删除数据库。
gaussdb=# DROP DATABASE testdb1;
gaussdb=# DROP DATABASE testdb2;
gaussdb=# DROP DATABASE testdb3;
gaussdb=# DROP DATABASE testdb4;
gaussdb=# DROP DATABASE testdb5;
gaussdb=# DROP DATABASE testdb_tmp;

--删除用户。
gaussdb=# DROP USER jim;

--删除表空间。
gaussdb=# DROP TABLESPACE tbs_test;
```

相关链接

ALTER DATABASE，**DROP DATABASE**

1.13.9.17 CREATE DATABASE LINK

功能描述

创建DATABASE LINK对象。

注意事项

- DATABASE LINK详细说明请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。
- DATABASE LINK特性只能在ORA兼容模式数据库下使用。
- 禁止通过使用DATABASE LINK的方式连接初始用户，禁止初始用户创建、修改和删除DATABASE LINK对象。
- 创建DATABASE LINK对象时，不会对其是否可连接成功进行验证，如果创建过程中缺乏相关关键字，可能会在使用时报错。
- 升级未提交情况下无法创建使用DATABASE LINK。
- 非透明多写特性下，使用本机IP及对应端口可连接当前实例数据库。由于CN和DN可能分布在不同服务器中，建议host参数不要使用127.0.0.1和localhost，可能会出现连接失败的情况。

语法格式

```
CREATE [ PUBLIC ] DATABASE LINK dblink
  [ CONNECT TO { CURRENT_USER | 'user' IDENTIFIED BY 'password' } ] [ USING ( {option 'value'} [, ...]) ];
```

其中option为：

```
{ HOST | PORT | DBNAME | FETCH_SIZE }
```


版权所有 © 华为云计算技术有限公司

参数说明

- **PUBLIC**

  指定创建对所有用户可见的公共数据库连接。如果未指定，则为私有的数据库连接，仅对当前用户可用。

- **dblink**

  创建的DATABASE LINK的名称。

- **user**

  创建的DATABASE LINK连接远程数据库使用的用户名。

- **password**

  创建的DATABASE LINK连接远程数据库使用的用户对应的密码。

- **CURRENT_USER**

  使用当前数据库初始用户名和空密码连接。

  说明

  当前版本指定CURRENT_USER选项后，可正常创建DATABASE LINK，但该连接无法正常使用。

- **USING ({option 'value'} [, ...])**

  可选。USING可选择指定要连接的数据库的IP地址、端口号、远程数据库名称等参数，支持的options包括：

  - host：指定连接的数据库IP地址，不指定时默认为空。可以指定多个IP地址，以逗号（,）进行分隔，当前不支持密态数据库和SSL设置和证书认证。
  - port：指定连接的端口号，不指定时默认为5432。
  - dbname：指定连接的数据库名称，不指定时默认为连接远程使用的用户名。
  - fetch_size：从远程每次获取数据量大小，取值范围为[0, 2147483647]，默认值为100。

示例

```
-- 创建一个兼容性为ORA的数据库。
gaussdb=# CREATE DATABASE ora_test_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE

-- 切换数据库。
gaussdb=# \c ora_test_db

-- 创建拥有系统管理员权限的用户。
ora_test_db=# CREATE USER local_user WITH SYSADMIN PASSWORD '********';
CREATE ROLE
ora_test_db=# SET ROLE local_user PASSWORD '********';
SET

-- 创建私有DBLink，host也可以是IPv6地址。
ora_test_db=> CREATE DATABASE LINK private_dblink CONNECT TO 'remote_user' IDENTIFIED BY '********'
USING (host '***.***.***.***',port '*****',dbname 'remote_db');
CREATE DATABASE LINK

-- 删除私有DBLink。
ora_test_db=> DROP DATABASE LINK private_dblink;
DROP DATABASE LINK

-- 创建公共DBLink，host也可以是IPv6地址。
ora_test_db=> CREATE PUBLIC DATABASE LINK public_dblink CONNECT TO 'remote_user' IDENTIFIED BY
'********' USING (host '***.***.***.***',port '*****',dbname 'remote_db');
CREATE DATABASE LINK
```


版权所有 © 华为云计算技术有限公司

```
-- 删除公共DBLink。
ora_test_db=> DROP PUBLIC DATABASE LINK public_dblink;
DROP DATABASE LINK

-- 删除创建出的用户。
ora_test_db=> RESET ROLE;
RESET
ora_test_db=# DROP USER local_user;
DROP ROLE

-- 切换回初始数据库，并删除测试数据库。请用真实的数据库名替换postgres。
ora_test_db=# \c postgres
gaussdb=# DROP DATABASE ora_test_db;
DROP DATABASE
```

相关链接

ALTER DATABASE LINK，**DROP DATABASE LINK**

1.13.9.18 CREATE DIRECTORY

功能描述

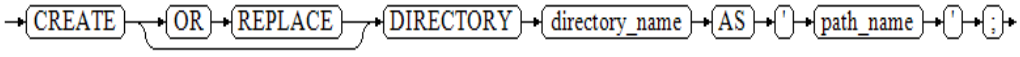
创建一个目录对象，该目录对象可以定义服务器文件系统上目录的别名，用于存放用户使用的数据文件，用户可以通过DBE_FILE高级包进行读写这些文件。用户可在PG_DIRECTORY表中查看系统已存在目录对象。

注意事项

- 目录对象的默认所有者为创建目录对象的用户，创建用户默认拥有此目录对象的READ和WRITE操作权限。当enable_access_server_directory=off时，只允许初始用户创建目录对象。当enable_access_server_directory=on时，具有SYSADMIN权限的用户和继承了内置角色gs_role_directory_create权限的用户可以创建目录对象。
- 对于用户自行部署的数据库场景，由用户自主完成权限体系的初始配置及管理，当进行目录权限授权时，必须确保目标目录满足以下安全条件：通过安全基线核查、具备访问控制审计机制以及完成敏感数据分类分级等安全条件要求。权限配置应遵循最小权限原则，并建立权限审批追溯机制等。
- 目录对象可向指定用户授予READ和WRITE操作权限，用于dbe_file的权限控制。请合理创建目录对象，若目录对象创建于操作系统的关键目录（如根目录）下，当对目录文件中的配置进行修改时，可能会引发非预期的安全风险问题。

语法格式

```
CREATE [OR REPLACE] DIRECTORY directory_name
   AS 'path_name';
```



参数说明

- **directory_name**

  创建的目录对象的名称。

  取值范围：字符串，要符合**标识符命名规范**。

版权所有 © 华为云计算技术有限公司

说明

创建目录对象时会进行以下合法性校验：

- 检查添加路径是否为操作系统实际存在路径，如不存在会提示用户使用风险。
- 校验数据库初始化用户（如omm）对于添加路径的权限（即操作系统目录权限，读/写/执行 - R/W/X），如果权限不足，会提示用户使用风险。

- **path_name**

  服务器文件系统上的目录路径。

  取值范围：有效的操作系统路径。

  说明

  - 在集群环境下，用户指定的目录路径需要保证各节点上路径的一致性，否则在不同节点上执行会产生无法匹配路径的问题。
  - 禁止创建以下目录路径：
    - 路径含特殊字符。
    - 路径是相对路径。

示例

```
--创建目录对象。
gaussdb=# CREATE OR REPLACE DIRECTORY dir AS '/tmp/';

--查看已创建的目录对象。
gaussdb=# select * from pg_directory;
 dirname | owner | dirpath | diracl
---------+-------+---------+--------
 dir     |    10 | /tmp    |
(1 row)

--删除目录对象。
gaussdb=# DROP DIRECTORY dir;
```

相关链接

ALTER DIRECTORY，**DROP DIRECTORY**

1.13.9.19 CREATE EXTENSION

须知

扩展功能为内部使用功能，目前仅允许创建（安装）数据库内部的高级包或扩展。当前仅支持创建（安装）postgis、postgis_sfcgal、postgis_raster、postgis_topology、yukon_geomodel、yukon_geogridcoder、pkg_bpchar_opc以及tablefunc。

功能描述

安装一个扩展。

注意事项

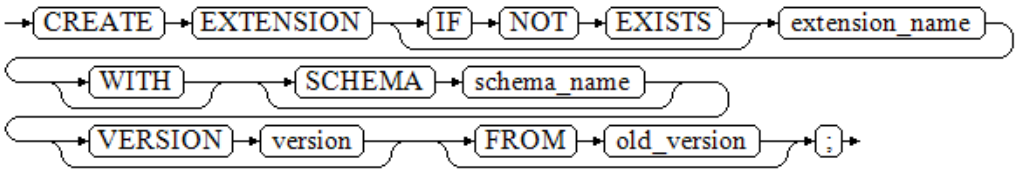
- 在执行该命令将扩展安装到数据库之前，必须先完成该扩展所依赖文件（即扩展本身）的安装。

版权所有 © 华为云计算技术有限公司

- 同一数据库中不得存在同名的扩展。若需避免因名称冲突导致的扩展安装失败场景，应在安装扩展前确认目标扩展名未被占用。
- 若数据库中存在与待安装的扩展同名的数据库对象（PACKAGE、同义词、操作符、目录、函数、存储过程、视图以及表）时，将会导致安装失败。
- 安装扩展即执行其对应的扩展脚本文件。该脚本将创建一个新的SQL实体，包括但不限于：函数、数据类型、操作符和索引支持的方法。
- 安装扩展需要拥有与创建扩展的组件对象相同的权限。对于大多数扩展，这意味着需要拥有数据库所有者的权限，对于后续的权限检查和该扩展脚本所创建的实体，运行CREATE EXTENSION命令的角色将成为扩展的所有者。
- 数据库默认禁止直接创建扩展，原因在于：扩展可能会引发非预期的错误，且在数据库升级后可能面临兼容性的问题。如果需要创建扩展，需要设置enable_extension为true才能创建。
- 创建扩展时，如果GUC参数enable_object_special_character为off，且扩展的脚本文件中使用@extschema@占位符，则扩展控制文件中schema参数的值不能包含以下任一特殊字符：[、"、$、'、\以及]。

语法格式

```
CREATE EXTENSION [ IF NOT EXISTS ] extension_name
[ WITH ] [ SCHEMA schema_name ]
[ VERSION version ]
[ FROM old_version ];
```



参数说明

- **IF NOT EXISTS**

  创建扩展时，用于判断是否存在同名的扩展。

  如果指定IF NOT EXISTS关键字，创建扩展前会在当前数据库中查找是否已存在同名的扩展，若存在同名扩展，则不会创建并返回NOTICE提示；未指定IF NOT EXISTS关键字时，若存在同名扩展，则直接返回ERROR告警。

- **extension_name**

  待安装的扩展名称，数据库中将使用文件SHAREDIR/extension/extension_name.control中的详细信息创建扩展。

- **schema_name**

  创建扩展过程中，指定扩展脚本文件所创建的函数、数据类型等对象所属的具体模式，指定的模式必须存在。如果创建扩展时脚本文件所创建的对象所属模式（即该选项）以及扩展的控制文件均未指定具体模式，则使用默认模式进行创建。

  说明

  扩展所依赖文件（即扩展本身）不属于任何模式，其名称在数据库范围内唯一，不受模式约束；但对于扩展创建的对象（函数、数据类型等），其属于某个具体的模式。

- **version**

版权所有 © 华为云计算技术有限公司

指定安装扩展的版本，可以作为一个标识符或者字符串，默认的版本在扩展的控制文件中指定。

- **old_version**

  当且仅当需要升级安装已有模块中不存在的内容时，才需要指定FROM old_version。通过CREATE EXTENSION FROM old_version命令运行安装脚本，将新的内容安装到已有模块的扩展中，而不是创建新的对象。特别说明，需要指定Schema为包含这些已存在对象的对应模式。

示例

在当前数据库安装扩展。例如安装security_plugin：

```
--安装扩展之前需要设置enable_extension为true才能够创建。
gaussdb=# SET enable_extension = true;

--安装扩展。
gaussdb=# CREATE EXTENSION IF NOT EXISTS security_plugin;

--删除扩展。
gaussdb=# DROP EXTENSION security_plugin;
```

1.13.9.20 CREATE FOREIGN DATA WRAPPER

功能描述

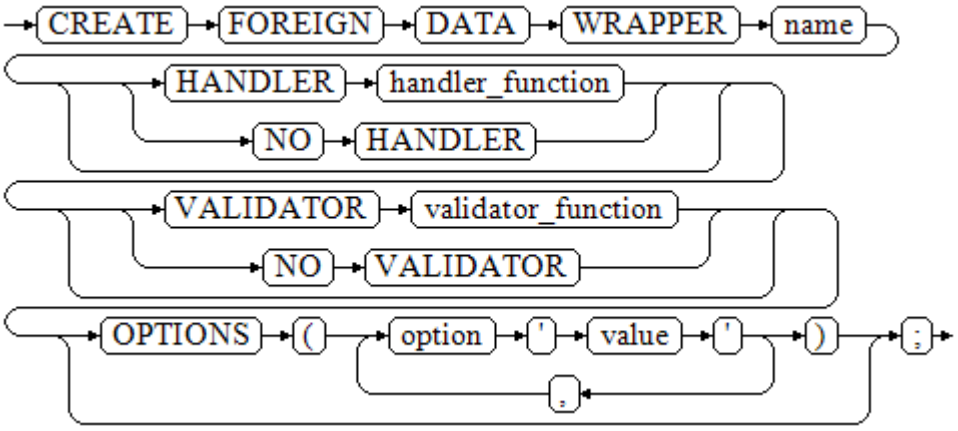
创建一个新的外部数据封装器。

注意事项

只有在内部初始化阶段或升级脚本中可以创建外部数据封装器，其他场景下用户无法直接使用该语法。

语法格式

```
CREATE FOREIGN DATA WRAPPER name
[ HANDLER handler_function | NO HANDLER ]
[ VALIDATOR validator_function | NO VALIDATOR ]
[ OPTIONS ( { option 'value' } [, ... ] ) ];
```



版权所有 © 华为云计算技术有限公司

参数说明

- **name**

  创建的外部数据封装器的名称。

  外部数据封装器的名称必须在数据库中唯一。

- **HANDLER handler_function**

  handler_function指已经注册了的函数的名称，用于为外部表检索执行的处理器函数。处理器函数handler_function必须没有参数，并且其返回类型必须为fdw_handler。

  可以不使用处理器函数进行创建外部数据封装器，但是使用这种数据封装器的外部表只能被声明，无法被访问。

- **NO HANDLER**

  在创建数据封装器时，不设置（使用）处理器函数，指定NO HANDLER创建的数据封装器的外部表只能被声明，无法被访问。

- **VALIDATOR validator_function**

  validator_function：表示已经注册了的函数的名字，用来检查提供给外部数据封装器的通用验证选项，还有使用该外部数据封装器的外部服务器、用户映射和外部表的验证选项。如果没有验证器选项（函数）或声明了NO VALIDATOR，那么在创建时将不检查选项（外部数据封装器可能在运行时忽略或拒绝无效的选项说明）。

  验证器选项（函数）validator_function必须接受两个参数：

  - 一个类型为text[]，包含存储在系统目录中的选项的数组。
  - 一个类型为oid，包含这些选项的系统目录的OID。

- **NO VALIDATOR**

  在创建时暂不设置验证器函数。

- **OPTIONS (option 'value' [, ...])**

  外部数据封装器声明选项。允许指定OPTIONS (option 'value' [, ...])是特定于每一个外部数据封装器的，并且是通过外部数据封装器中验证器函数进行验证的。选项名options必须符合唯一约束。

相关链接

ALTER FOREIGN DATA WRAPPER，**DROP FOREIGN DATA WRAPPER**

1.13.9.21 CREATE FOREIGN TABLE

功能描述

在GaussDB中，创建外表（CREATE FOREIGN TABLE）是一种方便的机制，用于访问外部数据源并对其进行统一管理和查询。通过创建外表，用户可以将存储在文件系统、外部表、视图等外部数据源中的数据集成到GaussDB数据库中，并像操作普通表一样进行查询、连接和并行操作等。在当前数据库创建一个GDS外表，用于数据并行导入导出。GDS外表分为只读外表和只写外表，分别用于数据并行导入和并行导出，缺省为只读外表。

注意事项

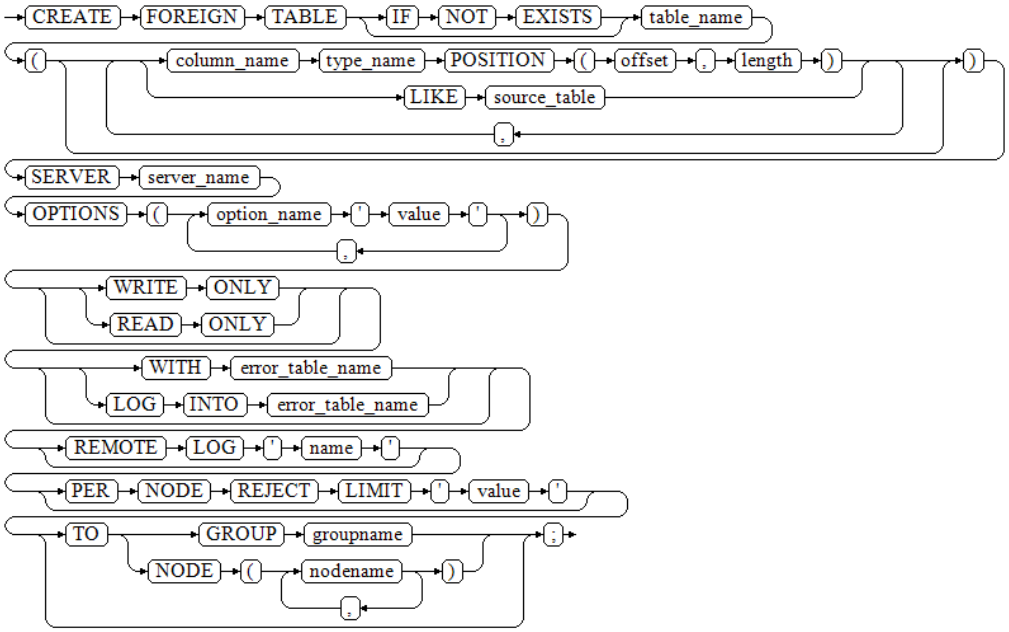
- 进行创建外表的用户将成为其所有者。

版权所有 © 华为云计算技术有限公司

- GDS外表不需要显式指定分布方式，默认支持ROUNDROBIN分布方式。
- 对于GDS外表指定任何约束（列约束、表约束等）均不生效。
- OPTIONS中的敏感字段（如password、secret_access_key）在使用多层引号时，语义和不带引号的场景是不同的，因此不会被识别为敏感字段进行脱敏。
- 在对建有全局二级索引的基表执行GDS数据导入时，需要开启enable_stream_operator参数，以达到最优数据导入性能。
- 使用GDS导入/导出数据时，数据中不能存在空字符（\0），否则会发生截断。

语法格式

```
CREATE FOREIGN TABLE [ IF NOT EXISTS ] table_name
   ( [ { column_name type_name  POSITION ( offset, length ) | LIKE source_table } [, ...] ] )
   SERVER server_name
   OPTIONS ( { option_name 'value' } [, ...] )
   [ WRITE ONLY | READ ONLY  ]
   [ WITH error_table_name | LOG INTO error_table_name]
   [REMOTE LOG 'name']
   [PER NODE REJECT LIMIT 'value']
   [ TO { GROUP groupname | NODE ( nodename [, ... ] ) } ];
```



参数概览

创建外表语法提供了多个参数，常用参数分类如**表1-396**所示。

版权所有 © 华为云计算技术有限公司

表 1-396 外表参数说明

必选/可选参数	具体参数	
必选参数	• table_name • column_name • type_name • POSITION(offset,length) • LIKEsource_table • SERVER server_name • OPTIONS可选参数	
可选参数	外表的数据源位置参数	location
	数据格式参数	• format • header（仅支持CSV、FIXED格式） • fileheader（仅支持CSV、FIXED格式） • out_filename_prefix • delimiter • quote（仅支持CSV格式） • escape（仅支持CSV格式） • null • noescaping（仅支持TEXT格式） • encoding • eol
	容错性参数	• fill_missing_fields • ignore_extra_data • reject_limit • compatible_illegal_chars • WITH error_table_name • LOG INTO error_table_name • REMOTE LOG 'name' • PER NODE REJECT LIMIT 'value'

参数说明

- **IF NOT EXISTS**

  用于判断是否存在同名的表。如果数据库中已经存在一个同名的表，不会直接报错，而是会产生一个提示，用于告知表关系已存在

- **table_name**

  外表的表名。

版权所有 © 华为云计算技术有限公司

取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

  外表中的字段名。

  取值范围：字符串，要符合**标识符命名规范**。

- **type_name**

  字段的数据类型。

- **POSITION (offset, length)**

  在固定长度模式中，定义每一个字段在数据文件中的位置。offset为该字段在数据源文件中的起始位置，length为该字段的长度。

  取值范围：offset取值不能小于0字节，单位为字节。每条记录的长度不能大于1GB，文件中没有出现的列默认以空值代替。

- **LIKE source_table**

  声明一个外表，新的外表自动从这个外表中继承所有字段名及其数据类型。

- **SERVER server_name**

  外表的server名称。对于GDS外表，其server是初始数据库默认创建的，即gsmpp_server。

  说明

  server对应的外部数据封装器名称不能为file_fdw。

- **OPTIONS ({ option_name ' value ' } [, ...])**

  用于指定外表数据的各类参数。

  - location

    外表的数据源位置，目前支持URL方式和本地文件方式的描述。多个URL和多个本地文件都使用或符号（|）进行分隔。

    说明

    - 对于使用GDS从远端服务器并行导入时的只读外部表（默认为只读）的URL末尾必须指定文件的匹配模式或者文件名。例如：gsfs:// 192.168.0.90:5000/*或者file:///data/data.txt或者gsfs:// 192.168.0.90:5000/* | gsfs:// 192.168.0.91:5000/*。
    - 对于使用GDS并行导出到远端服务器时的可写外部表，URL不需要指定文件名。当数据源位置为本地时，例如：file:///data/，外部表数据源位置只可指定一个，并且每个节点上都要预先创建好对应目录。当导出数据文件存储位置为远端URL时，例如：gsfs:// 192.168.0.90:5000/，数据源位置可指定多个；如果导出数据文件存储位置数量小于等于数据节点数量时，使用此外部表执行导出任务，系统会遍历节点并逐个分配导出数据文件存储位置；一轮分配结束后，系统将从头开始进行再次分配，因此，节点数量越多，分配会越均匀；若导出数据存储位置数量大于数据节点数量时，执行导出任务，数据将被平均分配给此位置列表中从前端开始等于数据节点数量的数据源位置下，剩余数据源位置仍会创建数据文件，但文件中不会有任何数据。
    - 对于使用GDS从远端服务器并行导入时的只读外表，URL个数应小于DN个数，且不能使用多个location相同的URL。
    - 当使用gsfss协议，即当URL为“gsfss://”开头，当执行加密导入导出时，并发不能超过10。
    - 如使用SSL加密传输时，发生证书替换时，需重启GDS。

  - format

    外表中数据源文件的格式。

    取值范围：CSV、TEXT、FIXED，默认值为TEXT。

版权所有 © 华为云计算技术有限公司

- CSV：对一些转义序列按照普通字符串进行处理，可以有效处理数据列中的换行符。
- TEXT：可以有效处理一些转义序列，无法正确处理数据列中的换行符。
- FIXED：适用于每条数据的数据列都比较固定的数据，长度不足的列会添加空格进行补齐。

说明

- 转义序列指的是反斜线（\）开头的字符串，包括：\b（退格）、\f（换页）、\n（换行）、\r（回车）、\t（横向制表）、\v（纵向制表）、\数字（八进制编码）、\x数字（十六进制编码）。TEXT格式可以按照本身含义进行处理，其他格式只能按照普通字符串进行处理。
- 当为定长格式（FIXED）时，必须为每一列指定POSITION。FIXED定义如下：
  - 每条记录的每个字段长度相同。
  - 长度不足的字段以空格填充，数字类型字段左对齐，字符字段右对齐。
  - 字段和字段之间没有分隔符。

– header

指定导出数据文件是否包含标题行，标题行一般用于描述表中每个字段的信息。header仅能用于CSV或FIXED格式的文件中。

- 在导入数据时：如果header选项为true/on，则数据文本第一行会被识别为标题行，忽略此行；如果header为false/off，数据文件中第一行都会被识别为数据。
- 在导出数据时：如果header选项为true/on，则需要指定**fileheader**。fileheader用于指定导出头文件的格式；如果header为false/off，则导出数据文件不包含标题行。

取值范围：true/on，false/off。默认值为false/off。

– fileheader

指定导出数据要包含的标题行定义的文件，文件一般只包含一行用于描述每一列数据信息的字符串。

例如：在包含商品信息的数据前加标题行，定义文件如下：The information of products.\n

说明

- 标题行定义文件仅在header为true/on的情况下有效，且需要提前写好备用。
- 在REMOTE导出模式下，定义文件必须放在GDS的工作目录（即启动GDS时指定的-d路径）下。
- 在LOCAL导出模式下，定义文件必须放在各个节点的相同路径，且fileheader指定的是绝对路径。
- 定义文件只能包含一行标题信息，并以换行符结尾，多余的行将被丢弃（标题信息不能包含换行符）。
- 定义文件包括换行符在内长度不超过1M。

– out_filename_prefix

指定只写（WRITE ONLY）外表导出时，GDS端生成导出数据文件的文件名前缀。

版权所有 © 华为云计算技术有限公司

说明

- 指定文件名前缀需合法，符合GDS部署物理环境使用的文件系统的约束，否则出现文件创建失败：
  - 指定的导出文件名前缀中不能含有非法字符（/、?、*、:'、|、\\、<、>、@、#、$、&、(、)、+以及-），其中非法字符包括但不限于以上列举，允许的字符范围为小写字母（a~z）、大写字母（A~Z）、数字（0~9）和下划线（_）。
  - 指定的导出文件名前缀中不可以为Windows和linux预留的特性字段，其中包括但不限于con、aux、nul、prn、com0、com1、com2、com3、com4、com5、com6、com7、com8、com9、lpt0、lpt1、lpt2、lpt3、lpt4、lpt5、lpt6、lpt7、lpt8或lpt9。
  - 指定的导出文件名前缀，在与gds –d目录和".dat.x"拼接为绝对路径后必须符合GDS所在的部署文件系统的文件名长度要求。
  - 指定的导出文件名前缀，需要可以被数据文件的最终接收方正确解析识别（包括但不限于GDS再次导入库中），对于造成文件名解析问题的指定选项，需要用户自己识别。
- 在多文件同时导出的高并发导出场景下，请确认并发的导出任务不要使用同一个文件名前缀设定，否则从操作系统/文件系统层面可能会出现导出文件的覆盖和丢失。

– delimiter

  指定数据文件行数据的字段分隔符，不指定则使用默认分隔符。TEXT格式默认分隔符为水平制表符（tab），CSV格式默认分隔符为逗号（,），FIXED格式没有默认分隔符。

  说明

  - 分隔符不能是\r和\n。
  - 分隔符不能和null参数相同，CSV格式数据的分隔符不能和quote参数相同。
  - TEXT格式数据的分隔符不能包含：反斜线（\）、点（.）、字母（a~z）、数字（0~9）。
  - 数据文件中单行数据长度需<1GB，如果分隔符较长且数据列较多的情况下，会影响导出有效数据的长度。
  - 分隔符推荐使用多字符和不可见字符（例如0x07、0x08、0x1b等）。

  取值范围：支持多字符分隔符，但分隔符不能超过10个字节。

– quote

  用于设置将CSV格式数据源文件中，某类字符识别为引号字符。默认值为双引号。

  说明

  - quote参数不能和分隔符、null参数相同。
  - quote参数只能是单字节的字符。
  - 推荐不可见字符（例如0x07，0x08，0x1b等）作为quote。

– escape

  用于设置将CSV格式的数据源文件中，某类字符识别为逃逸字符。逃逸字符只能指定为单字节字符。默认值为双引号。

  当escape与quote值相同时，会被替换为空字符（\0）。

– null

  用于指定数据源文件中空值的表示。

版权所有 © 华为云计算技术有限公司

说明

- null值不能是\r和\n，最大为100个字符。
- null值不能和分隔符、quote参数相同。

取值范围：

- 在TEXT格式：默认值为\n。
- CSV格式：默认值为一个没有引号的空字符串。

– noescaping

TEXT格式下，不对反斜线（\）和后面的字符进行转义。

说明

noescaping参数仅在TEXT格式下有效。

取值范围：true/on，false/off。默认值为false/off。

– encoding

指定数据源文件的编码格式名称，即需要以何编码格式对数据源文件进行解析和校验以及指定输出文件为何种编码格式。默认值为当前数据库的默认客户端编码格式，即client_encoding。

取值范围：请参见**CREATE DATABASE**章节的**ENCODING [=] 'encoding'**。

说明

- 导入外表时，建议指定encoding为文件的编码格式，或根据文件的字符集在导入前对client_encoding进行设置，否则可能会导致冗余解析、校验错误以及其导致的导入报错回滚，甚至非法数据入库。
- 导出外表时，建议同时指定此选项，以避免导出采用默认字符集设置时与预期不符。
- 在创建外表未指定encoding选项时，会在客户端提示对应的WARNING信息。
- 目前GDS导入外表不支持解析带有多种字符集编码格式混合的文件，GDS导出外表不支持写出带有多种字符集编码格式混合的文件。

– fill_missing_fields

在进行数据导入时，用于指定数据源文件中一行数据的最后一个字段缺失的处理方式。

取值范围：true/on，false/off。默认值为false/off。

- 当参数设置为true/on，进行数据导入时，若数据源文件中一行数据的最后一个字段缺失，则将最后一个字段的值设置为NULL，不报错。
- 当参数设置为false/off，如果最后一个字段缺失则会显示如下错误信息：

```
missing data for column "tt"
```

– ignore_extra_data

若数据源文件比外表定义列数多，对源文件中多出的列是否进行忽略操作。该参数只在数据导入过程中使用。

取值范围：true/on，false/off。默认值为false/off。

- 当参数设置为true/on，若数据源文件比外表定义列数多，则忽略行尾多余列。

版权所有 © 华为云计算技术有限公司

- 当参数设置为false/off，若数据源文件比外表定义列数多，则显示如下错误信息：
  ```
  extra data after last expected column
  ```

  说明

  如果行尾的换行符丢失导致两行数据合并为一行时，设置此参数为true/on时，后一行的数据将会被忽略。

– reject_limit

  指定本次数据导入允许出现的数据格式错误个数，当导入过程中出现的数据格式错误未达到限定值时，本次数据导入仍可成功执行。

  取值范围：整数类型值、unlimited（无限制）。默认值为0，表示存在错误信息则立即返回。指定正整数类型值时，需要添加单引号。

  说明

  - 此语法建议使用PER NODE REJECT LIMIT 'value'进行替代。
  - 数据格式错误是指缺少或多出字段值、数据类型错误或者编码错误。对于非数据格式错误，一旦发生就将导致整个数据导入失败。

– mode

  指定数据导入过程中，数据导入的策略。

  取值范围：

  - Normal（默认值）：支持所有文件格式（包括CSV、TEXT、FIXED），数据导入需要在数据服务器上启动Gauss data service协助完成。
  - Shared：支持TEXT文件格式，此时导入无需Gauss data service的协助，但是需要将用户数据通过NFS或添加到所有节点的相同路径下。
  - Private：用于用户数据已经存放到DN的本地目录下，并且位于相同的路径。

  说明

  - 对于Shared策略和Private策略下的LOCAL类型的外表，创建外表时会受GUC参数enable_copy_server_files以及GUC参数safe_data_path的控制。
  - 当GUC参数enable_copy_server_files关闭时，只允许初始用户创建Shared mode或Private mode的LOCAL类型外表；当GUC参数enable_copy_server_files打开时，允许具有SYSADMIN权限的用户或继承了内置角色gs_role_copy_files权限的用户执行，以防止用户越权查看或修改敏感文件。
  - 数据库管理员可以通过GUC参数safe_data_path设置所有用户可以导入导出路径的白名单，使用文件的路径必须为safe_data_path路径的子路径，将此GUC参数设置为空时候，不对用户使用的路径进行拦截。

– eol

  指定导入导出数据文件换行符样式。

  取值范围：支持多字符换行符，但换行符不能超过10个字节。常见的换行符，如\r、\n、\r\n（0x0D、0x0A、0x0D0A效果是相同的），其他字符或字符串，如$、#。

版权所有 © 华为云计算技术有限公司

说明

- eol参数只能用于TEXT格式的导入导出，不支持CSV格式以及FIXED格式的导入。为了兼容原有eol参数，支持导出CSV格式和FIXED格式时指定eol参数为0x0D或0x0D0A。
- eol参数不能和分隔符、null参数相同。
- eol参数不能包含：点（.）、字母（a~z）、数字（0~9）。

- fix

  指定每一行定长格式数据的长度，单位为字节。此参数仅对只读（READ ONLY）外表有效。

  取值范围：>= POSITION指定的总长度（总长度即为表定义最后一个字段的offset与length的和）并且不超过1GB。

- out_fix_alignment

  指定定长导出中，指定BYTEAOID、CHAROID、NAMEOID、TEXTOID、BPCHAROID、VARCHAROID、NVARCHAR2OID以及CSTRINGOID对应类型所在列的对齐方式。

  取值范围：align_left、align_right，默认值为align_right。

  说明

  在导入需要严格遵循十六进制或八进制格式的BYTE数据时，需确保数据左对齐且无前导空格。因此，在导出文件需要重新以GDS外表入库且数据长度小于外表formatter指定长度的情况下进行导出操作时，仍需确保数据左对齐，否则会在入库的过程中报错。

- date_format

  导入对于DATE类型指定格式。此语法仅对只读（READ ONLY）外表有效。

  取值范围：合法DATE格式，具体请参见**时间和日期处理函数和操作符**。

  说明

  对于指定为ORA兼容模式的数据库：

  - 当GUC参数mapping_date_to_datea设置为off时，创建DATE类型时，数据库按照TIMESTAMP(0) WITHOUT TIME ZONE类型进行实现。在进行导入操作时，若需指定格式，请参见**timestamp_format**。
  - 当GUC参数mapping_date_to_datea设置为on时，创建DATE类型时，数据库按照DATEA类型进行实现。在进行导入操作时，若需指定格式，请参见**datea_format**。

- time_format

  导入对于TIME类型指定格式。此语法仅对只读（READ ONLY）外表有效。

  取值范围：合法TIME格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

- timestamp_format

  导入对于TIMESTAMP类型指定格式。此语法仅对只读（READ ONLY）外表有效。

  取值范围：合法TIMESTAMP格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

- datea_format

  导入对于DATEA类型指定格式。此语法仅对只读（READ ONLY）外表有效。

  取值范围：合法DATEA格式，不支持时区。具体请参见**时间和日期处理函数和操作符**。

版权所有 © 华为云计算技术有限公司

- smalldatetime_format

  导入对于SMALLDATETIME类型指定格式。此语法仅对只读（READ ONLY）外表有效。

  取值范围：合法SMALLDATETIME格式。具体请参见**时间和日期处理函数和操作符**。

- compatible_illegal_chars

  导入非法字符容错参数。此语法仅对只读（READ ONLY）外表有效。

  取值范围：true/on，false/off。默认值为false/off。

  - 当参数设置为true/on，则导入时存在非法字符，进行容错处理，非法字符转换后入库且不报错，不中断导入。
  - 当参数设置为false/off，导入时存在非法字符则会产生报错，中断导入。

  说明

  导入非法字符容错规则如下：

  - 对于非法字符，容错后转换为GUC参数convert_illegal_char_mode设置的字符，默认为问号（？）。
  - 如果参数compatible_illegal_chars设置为true/on进行导入时，对于非法字符进行容错处理；如果参数null、参数delimiter、参数quote、参数escape设置为空格或问号，则会通过如“illegal chars conversion may confuse COPY escape 0x20”等报错信息提示用户修改可能引起混淆的参数，以避免导入错误。
  - 如果设置了GUC参数copy_special_character_version='no_error'，则会在导入和查询的过程中屏蔽非法字符编码的校验，将非法编码字符按原始样式导入，查询时以乱码显示，请谨慎选择开启此参数。
  - GUC参数copy_special_character_version设置为'no_error'的优先级高于参数compatible_illegal_chars设置为true/on。

- **READ ONLY**

  外表只读，该参数仅在数据导入时使用。

- **WRITE ONLY**

  外表只写。该参数仅在数据导出时使用。

- **WITH error_table_name**

  数据导入过程中出现的数据格式错误信息将被写入error_table_name指定的错误信息表中，可以在并行导入结束后查询此错误信息表并获取详细的错误信息。此参数仅在设置了reject_limit参数时有效。

  取值范围：字符串，要符合**标识符命名规范**。

  说明

  如果需要兼容PG开源接口，此语法建议使用LOG INTO代替。

- **LOG INTO error_table_name**

  数据导入过程中出现的数据格式错误信息将被写入error_table_name指定的错误信息表中，可以在并行导入结束后查询此错误信息表并获取详细的错误信息。此参数仅在设置了PER NODE REJECT LIMIT 'value'参数时有效。

  取值范围：字符串，要符合**标识符命名规范**。

- **REMOTE LOG 'name'**

  数据导入过程中出现的数据格式错误信息将被写入到GDS端以文件方式保存。name为错误数据文件的文件名前缀。

版权所有 © 华为云计算技术有限公司

- **PER NODE REJECT LIMIT 'value'**

  指定本次数据导入过程中每个DN实例上允许出现的数据格式错误的数量，如果有一个DN实例上的错误数量大于设定值，本次导入失败，报错退出。

  取值范围：整数类型值，unlimited（无限），默认值为0，有错误信息立即返回。

  说明

  - 此语法指定的是单个节点的错误容忍度。
  - 数据格式错误是指缺少或者多出字段值、数据类型错误或者编码错误。对于非数据格式错误，一旦发生就将导致整个数据扫描失败。

- **TO { GROUP groupname | NODE (nodename [, ...]) }**

  当前不支持使用该语法。仅为内部扩容工具使用，一般用户不应使用。

示例

```
--建立外表，用来以TEXT格式导入GDS服务器192.168.0.90和192.168.0.91上的数据，导入过程错误信息将记录到err_HR_staffS中。
gaussdb=# CREATE FOREIGN TABLE foreign_HR_staffS
(
  staff_ID       NUMBER(6) ,
  FIRST_NAME     VARCHAR2(20),
  LAST_NAME      VARCHAR2(25),
  EMAIL          VARCHAR2(25),
  PHONE_NUMBER   VARCHAR2(20),
  HIRE_DATE      DATE,
  employment_ID  VARCHAR2(10),
  SALARY         NUMBER(8,2),
  COMMISSION_PCT NUMBER(2,2),
  MANAGER_ID     NUMBER(6),
  section_ID  NUMBER(4)
) SERVER gsmpp_server OPTIONS (location 'gsfs://192.168.0.90:5000/* | gsfs://192.168.0.91:5000/*', format 'TEXT', delimiter E'\x20',  null '') WITH err_HR_staffS;

--建立外表，用来以TEXT格式导入GDS服务器192.168.0.90和192.168.0.91上的数据，导入过程错误信息将记录到err_HR_staffS中。本次数据导入允许出现的数据格式错误个数为2。
gaussdb=# CREATE FOREIGN TABLE foreign_HR_staffS_ft3
(
  staff_ID       NUMBER(6) ,
  FIRST_NAME     VARCHAR2(20),
  LAST_NAME      VARCHAR2(25),
  EMAIL          VARCHAR2(25),
  PHONE_NUMBER   VARCHAR2(20),
  HIRE_DATE      DATE,
  employment_ID  VARCHAR2(10),
  SALARY         NUMBER(8,2),
  COMMISSION_PCT NUMBER(2,2),
  MANAGER_ID     NUMBER(6),
  section_ID  NUMBER(4)
) SERVER gsmpp_server OPTIONS (location 'gsfs://192.168.0.90:5000/* | gsfs://192.168.0.91:5000/*', format 'TEXT', delimiter E'\x20',  null '',reject_limit '2') WITH err_HR_staffS_ft3;

--建立外表，用来以CSV格式导入input_data目录下存放在各个节点名文件下的所有文件。
gaussdb=# CREATE FOREIGN TABLE foreign_HR_staffS_ft1
(
  staff_ID       NUMBER(6) ,
  FIRST_NAME     VARCHAR2(20),
  LAST_NAME      VARCHAR2(25),
  EMAIL          VARCHAR2(25),
  PHONE_NUMBER   VARCHAR2(20),
  HIRE_DATE      DATE,
  employment_ID  VARCHAR2(10),
  SALARY         NUMBER(8,2),
  COMMISSION_PCT NUMBER(2,2),
```


版权所有 © 华为云计算技术有限公司

```
  MANAGER_ID      NUMBER(6),
  section_ID      NUMBER(4)
) SERVER gsmpp_server OPTIONS (location 'file:///input_data/*', format 'csv', mode 'private', delimiter ',')
WITH err_HR_staffS_ft1;

--建立外表，用来以CSV格式导出数据到output_data目录下。
gaussdb=# CREATE FOREIGN TABLE foreign_HR_staffS_ft2
(
  staff_ID        NUMBER(6) ,
  FIRST_NAME      VARCHAR2(20),
  LAST_NAME       VARCHAR2(25),
  EMAIL           VARCHAR2(25),
  PHONE_NUMBER    VARCHAR2(20),
  HIRE_DATE       DATE,
  employment_ID  VARCHAR2(10),
  SALARY          NUMBER(8,2),
  COMMISSION_PCT NUMBER(2,2),
  MANAGER_ID      NUMBER(6),
  section_ID  NUMBER(4)
) SERVER gsmpp_server OPTIONS (location 'file:///output_data/', format 'csv', delimiter '|', header 'on')
WRITE ONLY;

--删除外表。
gaussdb=# DROP FOREIGN TABLE foreign_HR_staffS;
gaussdb=# DROP FOREIGN TABLE foreign_HR_staffS_ft1;
gaussdb=# DROP FOREIGN TABLE foreign_HR_staffS_ft2;
gaussdb=# DROP FOREIGN TABLE foreign_HR_staffS_ft3;
```

相关链接

DROP FOREIGN TABLE

优化建议

- delimiter
  - 分隔符不能是\r和\n，分隔符不能和null参数相同，CSV格式数据的分隔符不能和quote参数相同。
  - 数据文件中单行数据长度需<1GB，如果分隔符较长且数据列较多的情况下，会影响导出有效数据的长度。
  - 分隔符推荐使用多字符和不可见字符（例如0x07、0x08、0x1b等）。
- quote
  - quote参数只能是单字节的字符，quote参数不能和分隔符、null参数相同。
  - 推荐不可见字符（例如0x07，0x08，0x1b等）作为quote。
- mode Normal

  支持所有文件格式（包括CSV、TEXT、FIXED），数据导入需要在数据服务器上启动Gauss data service协助完成。
- mode Shared

  支持TEXT文集格式，此时导入无需Gauss data service的协助，但是需要将用户数据通过NFS或挂载到所有节点的相同路径下。
- mode Private

  适用于用户数据已经存放到DN的本地目录下，并且位于相同的路径。

版权所有 © 华为云计算技术有限公司

1.13.9.22 CREATE FUNCTION

功能描述

创建一个函数。

注意事项

- 权限说明：
  - 新创建的函数默认会为PUBLIC角色授予执行权限（具体请参见**GRANT**）。用户默认继承PUBLIC角色权限，因此其他用户也会拥有函数的执行权限并可以查看函数的定义，另外执行函数时还需要具备函数所在Schema的USAGE权限。用户在创建函数时可以选择回收PUBLIC的默认执行权限，然后根据实际需求将执行权限授予给相应用户，为了避免出现新函数能被所有人访问的时间窗口，应在一个事务中创建函数并且设置函数执行权限。开启数据库对象隔离属性后，普通用户只能查看有权限执行的函数定义。
  - 被授予CREATE ANY FUNCTION权限的用户，可以在用户模式下进行创建以及替换函数操作。
  - 非初始用户无权对初始用户的函数进行REPLACE操作。
  - 如果将定义者权限的函数创建到其他用户Schema下，则会以其他用户的权限执行该函数，存在越权风险，请谨慎使用。
  - 函数默认为SECURITY INVOKER权限，如果想将默认行为改为SECURITY DEFINER权限，需要将GUC参数behavior_compat_options设置为plsql_security_definer。
- 兼容性行为说明：
  - ORA兼容模式的数据库，创建兼容Oracle风格数据库的函数；PG兼容模式的数据库，创建兼容PostgreSQL风格数据库的函数。不建议混合创建。
  - 兼容Oracle风格数据库的函数支持参数注释的查看与导出/导入，并且支持介于IS/AS与plsql_body之间的注释的查看与导出/导入。
- 函数创建行为说明：
  - 不能在avg函数外面嵌套其他agg函数或者其他系统函数。
  - 不能创建与存储过程拥有相同名称和参数列表的函数。
  - 不支持使用函数自身作为入参默认值。
  - 如果函数内部使用SET语句设置current_schema或search_path，该设置仅在函数执行期间对函数内部生效，函数执行结束后会自动还原为调用前的值，即函数内设置的search_path和current_schema在函数执行前后保持一致。
  - 不检查函数内返回值的类型。如果参数或返回值带有精度，不会进行精度检测。
  - 在普通集群模式下，暂不支持将返回值、参数以及变量设置为建在非系统默认安装Node Group的表，sql function内部语句暂不支持对建在非系统默认安装Node Group的表操作。
- 函数重载行为说明：
  - 兼容PostgreSQL风格的函数或者带有PACKAGE属性的函数支持函数重载。在进行指定REPLACE操作时，如果参数个数、类型或返回值发生变化，不会替换原有函数，而是会建立新的函数。
  - 函数如果支持重载，需要添加PACKAGE关键字。重载的函数在调用时，变量需要明确具体的类型。

版权所有 © 华为云计算技术有限公司

  - 不能创建仅形参名字不同（函数名和参数列表类型都一样）的重载函数。
  - 不支持函数重载特性的函数包括：仅返回类型不同的函数、仅默认值不同的函数、以及形参仅在自定义ref cursor与sys_refcursor之间不同的函数。

- 函数调用行为说明：
  - 在函数内部使用未声明的变量，函数被调用时会报错。在函数内部调用其他无参数的函数时，可直接使用函数名进行调用（即可以省略括号）。
  - SELECT调用可以指定不同参数来进行同名函数调用。语法CALL不支持调用未指定PACKAGE属性的同名函数。
  - 函数复杂调用（如：func(x).a），函数调用返回复合类型，支持跨Schema调用，不支持通过database.schema.package.func(x).b的方式调用。
  - 在函数的RETURN语句中，返回复合类型的构造器的调用时，实际返回类型与定义返回类型不一致时，可以隐式转换为定义返回类型时则对结果进行类型转换，支持跨Schema调用（如：RETURN schema.record），不支持跨DATABASE调用（如：RETURN package.schema.record）；在函数的RETURN语句中，返回FUNCTION的调用时，不支持在运算操作的表达式中带OUT参数的函数（如：RETURN func(c out) + 1）。
  - 带OUT模式参数的函数不能在SQL语句中被调用（包括SELECT INTO语法），且不支持嵌套调用。
- GUC行为控制说明：
  - 函数中存在通过GUC参数控制特性的语法、函数等，如果在会话内更改相关GUC参数，可能存在与预期结果不符，修改参数后，调用函数可能会维持修改前的行为，请谨慎进行变更GUC参数设置的操作。
  - 如果函数参数中带有出参且需要出参生效时，必须打开GUC参数behavior_compat_options（即SET behavior_compat_options = 'proc_outparam_override';），当进行调用函数（SELECT、CALL）时，出参位置必须提供实际参数进行调用，否则函数调用失败。
  - 在函数内部调用其他带有出参的函数，如果在赋值表达式中调用时，需要打开GUC参数（即SET behavior_compat_options = 'proc_outparam_override';），并提前定义与出参类型相同的变量，并将变量作为出参调用带有出参的其他函数，出参才能生效。否则，被调函数的出参会被忽略。
  - 在打开GUC参数（即SET behavior_compat_options = 'proc_outparam_override';）后，如果函数返回值为setof类型时，OUT出参不会生效。如果函数返回值为TABLE类型时，调用函数（SELECT、CALL）时，必须提供TABLE类型实际参数进行调用，否则函数调用失败。
  - 当GUC参数behavior_compat_options中不包含proc_outparam_override时，被匿名块或存储过程直接调用的函数的OUT、INOUT出参不能使用复合类型，并且RETURN值会被当作OUT出参的第一个值导致调用失败。如需正确使用OUT、INOUT出参，需打开GUC参数（即SET behavior_compat_options = 'proc_outparam_override';）。
  - 调用带OUT出参的存储过程，设置GUC参数（即SET behavior_compat_options = 'proc_outparam_transfer_length';）后可以传递参数长度。规格限制如下：
    - 支持的基本类型包括：CHAR(n)、CHARACTER(n)、NCHAR(n)、VARCHAR(n)、VARYING(n)、VARCHAR2(n)以及NVARCHAR2(n)。
    - OUT出参不生效的情况下（比如perform）不需要传递长度。

版权所有 © 华为云计算技术有限公司

  - 不支持精度传递的基本类型包括：NUMERIC、DECIMAL、NUMBER、FLOAT、DEC、INTEGER、TIME、TIMESTAMP、INTERVAL、TIME WITH TIME ZONE、TIMESTAMP WITH TIME ZONE、TIME WITHOUT TIME ZONE以及TIMESTAMP WITHOUT TIME ZONE。
  - GUC参数behavior_compat_options是否设置为proc_outparam_override时，都支持传递参数长度。
  - 如果需要传递集合类型的元素长度和被集合类型嵌套的数组类型的元素长度，则需要在GUC参数behavior_compat_options里开启tableof_elem_constraints选项。

- 其他行为说明：

  在表达式中使用OUT参数作为出参时，如下情况OUT出参不会生效，例如：使用EXECUTE IMMEDIATE sqlv USING func语法执行函数、使用SELECT func INTO语法执行函数、使用INSERT、UPDATE等DML语句执行函数、使用SELECT WHERE a=func()的方式、带OUT出参的函数作为入参时（如：fun(func(OUT b), a)）且OUT出参b未生效、使用带OUT出参的函数为变量默认值赋值等。

语法格式

- 兼容PostgreSQL风格的创建自定义函数语法。

```
CREATE [ OR REPLACE ] FUNCTION function_name
   ( [ { argname [ argmode ] argtype [ { DEFAULT | := | = } expression ]} [, ...] ] )
   [ RETURNS rettype [ DETERMINISTIC ] | RETURNS TABLE ( { column_name column_type }
[, ...] )]
   LANGUAGE lang_name
   [
     {IMMUTABLE | STABLE | VOLATILE }
      | {SHIPPABLE | NOT SHIPPABLE}
      | WINDOW
      | [ NOT ] LEAKPROOF
      | {CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }
      | {[ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER | AUTHID DEFINER |
AUTHID CURRENT_USER}
      | {fenced | not fenced}
      | {PACKAGE}
      | COST execution_cost
      | ROWS result_rows
      | SET configuration_parameter { {TO | =} value | FROM CURRENT }
   ][...]
   {
      AS 'definition'
      | AS 'obj_file', 'link_symbol'
   };
```

- 兼容Oracle风格的创建自定义函数的语法。

```
CREATE [ OR REPLACE ] FUNCTION function_name
   ( [ { argname [ argmode ] argtype [ { DEFAULT | := | = } expression ] } [, ...] ] )
   RETURN rettype [ DETERMINISTIC ]
   [
      {IMMUTABLE | STABLE | VOLATILE }
      | {SHIPPABLE | NOT SHIPPABLE}
      | {PACKAGE}
      | {FENCED | NOT FENCED}
      | [ NOT ] LEAKPROOF
      | {CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }
      | {[ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER | AUTHID DEFINER |
AUTHID CURRENT_USER}
      | COST execution_cost
      | ROWS result_rows
      | SET configuration_parameter { {TO | =} value | FROM CURRENT }
      | LANGUAGE lang_name
```


版权所有 © 华为云计算技术有限公司

```
  ][...]
  {
    IS | AS
  } plsql_body
/
```

参数说明

- **function_name**

  创建的函数名称，可以使用具体的模式进行指定。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  - 当函数名称超过63个字符时，数据库会截断并保留前63个字符当作函数名称。
  - 创建函数时，函数定义中，对于表对象的操作建议采用显式指定模式，否则可能会导致函数执行异常。
  - 建议不要创建和系统函数重名的函数，否则调用时需要指定函数的Schema。

- **argname**

  函数参数的名称。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  当函数参数的名称超过63个字符时，数据库会截断并保留前63个字符当作函数参数的名称。

- **argmode**

  函数参数的模式。

  取值范围：IN、OUT、INOUT或VARIADIC。默认值为IN。

  说明

  - VARIADIC用于声明数组类型的参数。
  - 只有OUT模式的参数后面能跟VARIADIC，并且OUT和INOUT模式的参数不能用在RETURNS TABLE的函数定义中。

- **argtype**

  函数参数的类型。可以使用%ROWTYPE间接引用表的类型，或者使用%TYPE间接引用表或复合类型中某一列的类型。

  具体请参见**定义变量**。

- **expression**

  参数的默认表达式。

版权所有 © 华为云计算技术有限公司

说明

- 在参数a_format_version值为10c且a_format_dev_version值为s2的情况下，函数参数为INOUT模式时不支持默认表达式。
- 推荐使用方式：将所有默认值参数定义在所有非默认值参数后。
- 调用带有默认参数的函数时，入参从左向右排入函数，如果有非默认参数的入参存在缺失时则会报错。
- 打开proc_uncheck_default_param参数，调用带有默认参数的函数时，入参从左向右排入函数，允许缺省默认参数的入参，如果有非默认参数的入参存在缺失时，则会用错位的默认值填充该参数。
- 在参数a_format_version值为10c且a_format_dev_version值为s1的情况下，关闭proc_outparam_override参数，函数参数同时包括OUT出参和DEFAULT时，默认值不可缺省。

- **rettype**

  函数返回值的数据类型。与**argtype**相同，可以使用%ROWTYPE或者%TYPE间接引用类型。

  如果存在OUT或INOUT参数，可以省略RETURNS子句。如果没有省略，则该子句必须与输出参数所表示的结果类型一致，如果存在多个输出参数，则为RECORD，否则与单个输出参数的类型相同。SETOF修饰词表示该函数将返回一个集合，而不是单独一项。

- **column_name**

  字段名称。

- **column_type**

  字段类型。

  说明

  当返回类型为RETURNS TABLE时，会将每个column都当作OUT参数，在打开GUC参数（即SET behavior_compat_options = 'proc_outparam_override';）时，使用SELECT、CALL调用函数必须根据对应column提供实参进行调用，否则报错函数不存在。

- **definition**

  一个定义函数的字符串常量，含义取决于语言。可以为一个内部函数名称、一个指向某个目标文件的路径、一个SQL查询或者一个过程语言文本。

- **DETERMINISTIC**

  SQL语法兼容接口，当前版本不支持使用。

- **LANGUAGE lang_name**

  用以实现函数的语言的名称。可以为SQL、C、internal或者是用户定义的过程语言名称。为了保证兼容性，该名称可以用单引号（包围）。若采用单引号，则引号内必须为大写。

  由于兼容性问题，Oracle风格的语法无论指定任何语言，最终创建的语言都为plpgsql。

  说明

  - internal函数在定义时，如果AS指定为内部系统函数，则新创建函数的参数类型、参数个数以及返回值类型需要与内部系统函数保持一致，且需要有执行此内部系统函数的权限。
  - internal函数只支持拥有SYSADMIN权限的用户创建。

- **WINDOW**

版权所有 © 华为云计算技术有限公司

表示该函数是窗口函数，通常只用于C语言编写的函数。替换函数定义时不能改变WINDOW属性。

须知

自定义窗口函数只支持LANGUAGE是internal，并且引用的内部函数必须是窗口函数。

- **IMMUTABLE**

  表示该函数在给出同样的参数值时总是返回同样的结果。

- **STABLE**

  表示该函数不能修改数据库。对相同参数值，在同一次表扫描里，该函数的返回值不变，但是返回值可能在不同SQL语句之间变化。

- **VOLATILE**

  表示该函数值可以在一次表扫描内改变，因此不会做任何优化。

- **SHIPPABLE | NOT SHIPPABLE**

  表示该函数是否可以下推到DN上执行。

  - 对于IMMUTABLE类型的函数，函数始终可以下推到DN上执行。
  - 对于STABLE或者VOLATILE类型的函数，仅当函数的属性是SHIPPABLE的时候，函数可以下推到DN执行。

须知

- 如果未指定IMMUTABLE | STABLE | VOLATILE关键字时，默认指定为VOLATILE。
- 如果未指定SHIPPABLE | NOT SHIPPABLE关键字时，默认指定为NOT SHIPPABLE。
- 对于指定了SHIPPABLE或IMMUTABLE的函数或者存储过程，其不能包含EXCEPTION或调用含有EXCEPTION的函数或者存储过程。
- 创建函数时，如果指定为IMMUTABLE和SHIPPABLE类型，请勿在函数中进行INSERT、UPDATE、DELETE、MERGE和DDL操作，否则执行结果可能产生错误。由于以上操作应该由CN判断对应的执行节点，如果在声明为IMMUTABLE和SHIPPABLE类型的函数中推执行了DDL，可能会导致各节点数据库对象不一致。可通过在CN上创建VOLATILE PL/SQL函数，函数定义中使用EXECUTE语句动态执行用于修复系统对象的DDL，再使用EXECUTE DIRECT ON语法在指定的DN上执行修复函数调用的方式进行解决该问题。
- 如果函数的结果依赖对数据表的扫描，或者函数有对数据表修改数据、DDL的操作，请勿将函数属性设值为IMMUTABLE或者SHIPPABLE，否则可能造成数据库执行异常。

- **PACKAGE**

  表示该函数是否支持重载。

  - 不允许package函数和非package函数重载或者替换。
  - package函数不支持VARIADIC类型的参数。

版权所有 © 华为云计算技术有限公司

  - 不允许修改函数的package属性。
- **LEAKPROOF**

  表示该函数的参数只包括返回值。仅系统管理员可以设置LEAKPROOF。
- **CALLED ON NULL INPUT**

  表示该函数的某些参数是NULL时，可以按照正常的方式进行调用。该参数可以省略。
- **RETURNS NULL ON NULL INPUT | STRICT**

  STRICT用于指定如果函数的某个参数为NULL时，此函数则一直返回NULL。如果声明了STRICT，当函数中存在NULL值时，该函数不会被执行，自动返回一个NULL结果。

  RETURNS NULL ON NULL INPUT和STRICT的功能相同。
- **EXTERNAL**

  可选，目的是与SQL进行兼容，EXTERNAL适合于所有函数，而不仅是外部函数。
- **SECURITY INVOKER | AUTHID CURRENT_USER**

  表示该函数将以调用者（调用函数的用户）的权限执行。该参数可以省略。

  SECURITY INVOKER和AUTHID CURRENT_USER的功能相同。
- **SECURITY DEFINER | AUTHID DEFINER**

  表示该函数将以创建者（创建函数的用户）的权限执行。

  AUTHID DEFINER和SECURITY DEFINER的功能相同。
- **FENCED | NOT FENCED**

  该参数用于声明函数是在保护模式还是非保护模式下执行。如果函数声明为NOT FENCED模式，则函数的执行在CN或者DN进程中进行。如果函数声明为FENCED模式，则函数在新fork的进程执行，这样函数的异常不会影响CN或者DN进程。FENCED以及NOT FENCED模式的选择方式如下：
  - 正在开发或者调试的函数使用FENCED模式。开发测试完成，使用NOT FENCED模式执行，减少fork进程以及通信的开销。
  - 复杂的操作系统操作（例如：打开文件，信号处理，线程处理等操作），使用FENCED模式，否则可能影响GaussDB数据库的执行。
  - 用户自定义PL/SQL函数，如果不指定该参数，默认为NOT FENCED，且不支持指定为FENCED执行模式。
- **COST execution_cost**

  用于估计函数的执行成本。execution_cost以cpu_operator_cost为单位。

  取值范围：大于等于0的数值。
- **ROWS result_rows**

  估计函数返回的行数。仅用于函数返回集合的场景。

  取值范围：大于等于0的数值，默认值为1000。
- **configuration_parameter**

  配置参数。
  - **value**

    将指定的数据库会话参数值设置为给定的值。

    取值范围：字符串
    - DEFAULT：新的会话中使用系统的默认设置。

版权所有 © 华为云计算技术有限公司

  - OFF：关闭设置。
  - RESET：新的会话中使用系统的默认设置
  - 用户自定义的默认值。
- **FROM CURRENT**

  将当前会话中的值设置为configuration_parameter的值。

- **plsql_body**

  PL/SQL存储过程体。

  创建一个存储过程时，plsql_body支持以“END;”或“END function_name;”两种形式结尾。以“END function_name;”结尾时需遵循的约束请参见**CREATE PROCEDURE**章节中的**说明**。

  说明

  当在函数体中进行创建用户、修改密码或加解密等涉及密码或密钥相关操作时，系统表及日志中会记录密码或密钥的明文信息。为防止敏感信息泄露，不建议用户在函数体中进行涉及密码或密钥等敏感信息的相关操作。

示例

```
--创建一个兼容性为ORA的数据库。
gaussdb=# CREATE DATABASE ora_compatible_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE

--切换数据库。
gaussdb=# \c ora_compatible_db

--定义函数为SQL查询。
ora_compatible_db=# CREATE FUNCTION func_add_sql(integer, integer) RETURNS integer
    AS 'select $1 + $2;'
    LANGUAGE SQL
    IMMUTABLE
    RETURNS NULL ON NULL INPUT;

--利用参数名用plpgsql自增一个整数。
ora_compatible_db=# CREATE OR REPLACE FUNCTION func_increment_plsql(i integer) RETURNS integer AS
$$
        BEGIN
            RETURN i + 1;
        END;
    $$ LANGUAGE plpgsql;

--返回RECORD类型。
ora_compatible_db=# CREATE OR REPLACE FUNCTION compute(i int, out result_1 bigint, out result_2
bigint)
    RETURNS SETOF RECORD
    AS $$
    BEGIN
        result_1 = i + 1;
        result_2 = i * 10;
    RETURN next;
    END;
    $$ LANGUAGE plpgsql;

--返回一个包含多个输出参数的记录。
ora_compatible_db=# CREATE FUNCTION func_dup_sql(in int, out f1 int, out f2 text)
    AS $$ SELECT $1, CAST($1 AS text) || ' is text' $$
    LANGUAGE SQL;

--调用func_dup_sql函数。
ora_compatible_db=# SELECT * FROM func_dup_sql(42);
 f1 |     f2
```


版权所有 © 华为云计算技术有限公司

```
----+------------
 42 | 42 is text
(1 row)

--计算两个整数的和，并返回结果。如果输入为null，则返回null。
ora_compatible_db=# CREATE FUNCTION func_add_sql2(num1 integer, num2 integer) RETURN integer
    AS
    BEGIN
        RETURN num1 + num2;
    END;
    /

--创建package属性的重载函数。
ora_compatible_db=# CREATE OR REPLACE FUNCTION package_func_overload(col int, col2  int)
    RETURN integer PACKAGE
    AS
    DECLARE
        col_type text;
    BEGIN
         col := 122;
         dbe_output.print_line('two int parameters ' || col2);
         RETURN 0;
    END;
    /

--创建package_func_overload函数第二个形参参数类型不一样的重载函数。
ora_compatible_db=# CREATE OR REPLACE FUNCTION package_func_overload(col int, col2 smallint)
    RETURN integer PACKAGE
    AS
    DECLARE
        col_type text;
    BEGIN
         col := 122;
         dbe_output.print_line('two smallint parameters ' || col2);
         RETURN 0;
    END;
    /

--删除函数。
ora_compatible_db=# DROP FUNCTION func_add_sql;
ora_compatible_db=# DROP FUNCTION func_increment_plsql;
ora_compatible_db=# DROP FUNCTION compute;
ora_compatible_db=# DROP FUNCTION func_dup_sql;
ora_compatible_db=# DROP FUNCTION package_func_overload(col int, col2  int);
ora_compatible_db=# DROP FUNCTION package_func_overload(col int, col2 smallint);
ora_compatible_db=# DROP FUNCTION func_add_sql2;



--不打开参数set behavior_compat_options = 'proc_outparam_override'时，被匿名块或存储过程直接调用的函数的OUT、INOUT出参不能使用复合类型，并且RETURN值会被当做OUT出参的第一个值，导致调用失败。
ora_compatible_db=# CREATE TYPE rec AS(c1 int, c2 int);
ora_compatible_db=# CREATE OR REPLACE FUNCTION func(a in out rec, b in out int) RETURN int
    AS
    BEGIN
      a.c1:=100;
      a.c2:=200;
      b:=300;
      RETURN 1;
    END;
    /
ora_compatible_db=# DECLARE
        r rec;
        b int;
    BEGIN
        func(r,b);  --不支持
    END;
    /
ERROR:  A non-composite value cannot be assigned to a row variable.
```


版权所有 © 华为云计算技术有限公司

```
CONTEXT:  PL/pgSQL function inline_code_block line 4 at SQL statement
ora_compatible_db=# DROP FUNCTION func;
ora_compatible_db=# DROP TYPE rec;

--以下示例只有当数据库兼容模式为ORA时可以执行。
ora_compatible_db=# CREATE OR REPLACE PACKAGE pkg_type AS
        type table_of_index_int is table of integer index by integer; --创建integer类型
        type table_of_index_int01 is table of table_of_index_int index by integer; --创建嵌套integer类型
        type table_of_index_var is table of integer index by varchar(5); --创建varchar类型
        type table_of_index_var01 is table of table_of_index_var index by varchar(5); --创建嵌套varchar类型
    END pkg_type;
    /

--创建一个返回table of integer index by integer类型结果的函数。
ora_compatible_db=# CREATE OR REPLACE FUNCTION func_001(a in out pkg_type.table_of_index_int, b in
out pkg_type.table_of_index_var) --#add in & inout #defult value
    RETURN pkg_type.table_of_index_int
    AS
        table_of_index_int_val pkg_type.table_of_index_int;
        table_of_index_var_val pkg_type.table_of_index_var;
    BEGIN
        for i in 1..2 loop
            table_of_index_int_val(i) := i;
            a(i) := i;
            table_of_index_var_val(i) := i;
            b(i) := i;
        end loop;
        raise info '%',table_of_index_int_val;
        raise info '%',table_of_index_var_val;
        raise info '%',a;
        raise info '%',b;
    RETURN table_of_index_int_val;
    END;
    /
ora_compatible_db=# DECLARE
        table_of_index_int_val pkg_type.table_of_index_int;
        table_of_index_var_val pkg_type.table_of_index_var;
    BEGIN
        func_001(table_of_index_int_val,table_of_index_var_val);
    END;
    /

--创建一个含有IN/OUT类型参数的函数。
ora_compatible_db=# CREATE OR REPLACE FUNCTION func_001(a in out date, b in out date) --#add in &
inout #defult value
    RETURN integer
    AS
    BEGIN
        raise info '%', a;
        raise info '%', b;
        RETURN 1;
    END;
    /
ora_compatible_db=# DECLARE
        date1 date := '2022-02-02';
        date2 date := '2022-02-02';
    BEGIN
        func_001(date1, date2);
    END;
    /

--创建一个含有IN/OUT类型参数的函数。
ora_compatible_db=# CREATE OR REPLACE FUNCTION func_001(a in out INT, b in out date) --#add in &
inout #default value
    RETURN INT
    AS
    BEGIN
        raise info '%', a;
        raise info '%', b;
```


版权所有 © 华为云计算技术有限公司

```
    RETURN a;
    END;
    /
ora_compatible_db=# DECLARE
        date1 int := 1;
        date2 date := '2022-02-02';
    BEGIN
        func_001(date1, date2);
    END;
    /
INFO:  1
CONTEXT:  PL/pgSQL function inline_code_block line 5 at SQL statement
INFO:  2022-02-02 00:00:00
CONTEXT:  PL/pgSQL function inline_code_block line 5 at SQL statement
ANONYMOUS BLOCK EXECUTE

--删除函数。
ora_compatible_db=# DROP FUNCTION func_001;

--删除package。
ora_compatible_db=# DROP PACKAGE pkg_type;

--切换回初始数据库，并删除测试数据库。请用真实的数据库名替换postgres。
ora_compatible_db=# \c postgees
gaussdb=# DROP DATABASE ora_compatible_db;
```

相关链接

ALTER FUNCTION，**DROP FUNCTION**

1.13.9.23 CREATE GLOBAL INDEX

功能描述

在指定的表上创建全局二级索引（Global Secondary Index，简称GSI）。

全局二级索引允许用户定义与基表分布不一致的索引，从而实现基表非分布列查询的单节点计划和基表非分布列上的unique/主键约束。

说明

建议在GSI的基表创建普通索引提升DML操作执行效率。另外，在基表上创建GSI后将影响DML操作的性能，带来劣化。

功能约束

- 仅支持GTM-LITE模式。
- GSI支持表达式索引，但存在以下约束：
  - 同基表约束，不支持分布键包含表达式（且无法创建索引列仅包含表达式的GSI，因为此时分布键必定为表达式）。
  - 同普通索引约束，不支持CONTAINING列中包含表达式。
  - 若表上存在以"expr"为前缀的列名，不支持创建带有表达式的GSI。
- 创建GSI：
  - 不支持创建与基表分布一致的GSI。
  - 不支持基表为复制表、List分布/Range分布等，对于GSI本身不支持Hash分布以外的分布。
  - 当基表为分区非HASHBUCKET表时，GSI最多支持27列；当基表为HASHBUCKET非分区表时，GSI最多支持27列；当基表为HASHBUCKET分区

版权所有 © 华为云计算技术有限公司

表时，GSI最多支持26列；当基表为非分区非HASHBUCKET表最多支持28列。

- 在线创建重建GSI：

  不支持表达式索引和部分索引CREATE GSI CONCURRENTLY。

注意事项

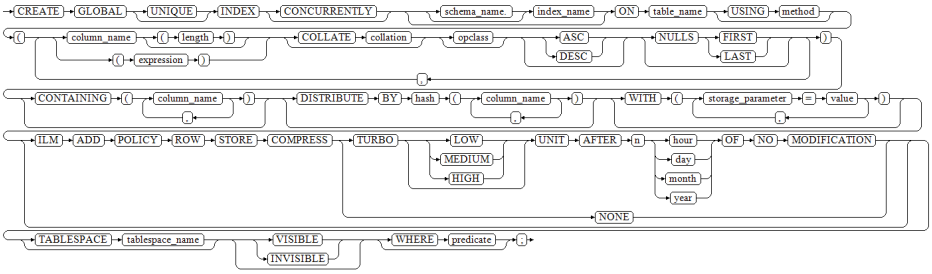
- 对于创建GSI、重建GSI以及涉及重建GSI的操作：比如分区表分区操作（包括DROP、TRUNCATE、MERGE、SPLIT、EXCHANGE PARTITION）指定UPDATE DISTRIBUTED GLOBAL INDEX、ALTER TABLE涉及重建数据的操作、HASHBUCKET表ALTER SET TABLESPACE操作、MOVE PARTITION操作，建议开启STREAM模式以达到最优性能。（其中，STREAM模式指设置enable_stream_operator参数为on，并设置create_gsi_opt参数置为build）
- 如果在执行VACUUM FULL、CLUSTER或者REINDEX操作时中断，表上的GSI可能会变为UNUSABLE状态，此时查询语句执行GSI会报错，建议执行REINDEX INDEX重建GSI。
- 在对建有GSI的基表执行COPY、GDS数据导入时，需要开启enable_stream_operator参数，以达到最优数据导入性能。
- 回表支持STREAM和非STREAM分布式计划。其中STREAM回表计划，继承STREAM相关约束。考虑到STREAM通信时延，当选择率过低或者谓词命中行数较少时，性能非最优，不建议使用GSI STREAM回表计划。非STREAM分布式回表计划，需使用gsitable hint，适合在谓词命中行数较少的场景使用。当STREAM回表计划性能非最优时，也可考虑使用非STREAM分布式回表计划。具体案例参见《SQL调优指南》中“实际调优案例 > 案例：使用全局二级索引加速查询”章节。
- 对于INSERT INTO SELECT批量插入、UPDATE批量更新、DELETE批量删除、UPSERT批量场景，建议打开enable_stream_operator，执行STREAM计划（当基表为HASHBUCKET表，以及防篡改表时，不会执行STREAM计划，仍然采用回到CN的计划），如果关闭enable_stream_operator，执行计划采用回到CN的方式，性能较差。
- 对于建有GSI的基表的DELETE、UPDATE、UPSERT语句，若为STREAM计划，同UPDATE约束，不支持并发更新同一行。
- GSI支持UPSERT，约束和普通UPSERT保持一致，具体请参见**参数说明**。
- 当前会使GSI失效的操作：REINDEX数据库级、CLUSTER数据库级/分区级、ALTER TABLE PARTITION（DROP、TRUNCATE、MERGE、SPLIT、EXCHANGE PARTITION未指定UPDATE DISTRIBUTED GLOBAL INDEX将失效分区表上的所有GSI，其中，EXCHANGE PARTITION未指定UPDATE DISTRIBUTED GLOBAL INDEX将同步失效普通表上的所有GSI）。
- 不支持对基表列名或者ctid、xc_node_hash、xmin、xmax、tableoid（当基表为分区表时）、tablebucketid（当基表为HASHBUCKET表时）增加_new$$、_NEW$$后与自身列名重复的基表创建GSI。
- 不支持REINDEX INDEX并发重建GSI索引，会存在阻塞的情况。
- GSI回表为跨数据节点回表，当优化器优先生成下推计划时，无法生成GSI回表计划。

语法格式

```
CREATE GLOBAL [ UNIQUE ] INDEX [ CONCURRENTLY ] [ [schema_name.]index_name ] ON table_name
[ USING method ]
    ({ { column_name [ ( length ) ] | ( expression ) } [ COLLATE collation ] [ opclass ] [ ASC | DESC ]
```


版权所有 © 华为云计算技术有限公司

```
[ NULLS { FIRST | LAST } ] }[, ...] )
    [CONTAINING (column_name [ ,... ])] [DISTRIBUTE BY hash(column_name [ ,... ])][ WITH
( {storage_parameter = value} [, ... ] )] [ ILM ADD POLICY ROW STORE COMPRESS { TURBO [ LOW |
MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION| NONE}]
    [ TABLESPACE tablespace_name ] [ VISIBLE | INVISIBLE ] [ WHERE predicate];
```



参数说明

- **UNIQUE**

  创建唯一性索引，每次添加数据时检测表中是否存在重复值。如果插入或更新的值会引起重复的记录时，将报错。

- **CONCURRENTLY**

  以不阻塞DML的方式创建索引（加ShareUpdateExclusiveLock锁）。创建索引时，一般会阻塞其他语句对该索引所依赖表的访问。指定此关键字，可以实现创建过程中不阻塞DML。

  - 普通CREATE GLOBAL INDEX命令可以在事务内执行，但是CREATE GLOBAL INDEX CONCURRENTLY不可以在事务内执行。
  - 索引在线创建过程中可能会发生构建失败。构建失败的可能情形包括用户指令终止在线创建命令、修改该索引的定义、修改基表定义且影响了该索引的定义等。对于非严重错误，索引在线构建失败时有报错提示并尝试自动清理该索引。对于严重错误（如FATAL/PANIC/数据库故障等场景）或清理失败时，会留下一个“不可用”的索引。该索引会被查询忽略，但其仍消耗IUD开销。因此推荐的恢复方法是通过DROP INDEX IF EXISTS语法删除该索引并尝试再次在线创建索引，或通过REINDEX/REBUILD语法重建索引。
  - CREATE GLOBAL INDEX CONCURRENTLY执行期间，应尽量避免与其他DDL并发执行，存在一定概率的死锁、报错等情况，可以重新连接客户端进行重试。
  - 其他注意事项请参见**CREATE INDEX**章节中的**CONCURRENTLY**。

- **schema_name**

  模式的名称。

  取值范围：已存在模式名。

- **index_name**

  创建的索引名，不能包含模式名，索引的模式与表相同。

  取值范围：字符串，要符合**标识符命名规范**。

- **table_name**

  需要为其创建索引的表的名称，可以使用模式进行具体指定。

  取值范围：已存在的表名。

- **USING method**

  指定创建索引的方法。

  取值范围：ubtree，提供多版本B-Tree索引，索引页面上包含事务信息。

版权所有 © 华为云计算技术有限公司

- **column_name**

  表中需要创建索引的列的名称（字段名）。

  如果索引方式支持多字段索引，可以声明多个字段，对于非分区基表最多可以声明28个字段，对于分区基表最多可以声明27个字段。

- **column_name (length)**

  支持前缀键，具体请参见**CREATE INDEX**章节中的**column_name (length)**。

  前缀索引作为一种表达式索引，在GSI中的约束与表达式键一致。

- **expression**

  创建一个基于该表的一个或多个字段的表达式索引，通常必须写在圆括号中。如果表达式有函数调用的形式，括号可以省略。表达式索引可用于获取对基本数据的某种变形的快速访问。比如，一个在upper(col)上的函数索引将允许WHERE upper(col) = 'JIM'子句使用索引。在创建表达式索引时，如果表达式中包含IS NULL子句，则该索引为无效的索引。此时，建议用户尝试创建一个部分索引。

- **COLLATE collation**

  指定列的排序规则（该列必须为可排列的数据类型）。如果未指定，则使用默认的排序规则。可以使用“SELECT * FROM pg_collation;”命令从pg_collation系统表中查询排序规则，默认的排序规则为查询结果中以default开始的行。

- **opclass**

  操作符类的名称。对于索引的每一列可以指定一个操作符类，操作符类标识了索引那一列的使用的操作符。

- **ASC**

  指定按升序排序（默认）。

- **DESC**

  指定按降序排序。

- **NULLS FIRST**

  指定空值在排序中排在非空值之前，当指定DESC排序时，本选项为默认的。

- **NULLS LAST**

  指定空值在排序中排在非空值之后，未指定DESC排序时，本选项为默认的。

- **CONTAINING**

  全局二级索引可以包含索引键以外的基表属性。

- **CONTAINING (column_name [, ...])**

  全局二级索引包含的索引键以外的基表属性。

- **DISTRIBUTE BY**

  指定全局二级索引分布键，其分布键需要与基表不同，且只能使用哈希分布方式。

- **DISTRIBUTE BY hash(column_name [,...])**

  全局二级索引分布键。

  需要被包含在索引键中。

- **WITH ({storage_parameter = value} [, ...])**

  指定GSI的存储参数。

  取值范围：

  - FILLFACTOR

版权所有 © 华为云计算技术有限公司

一个索引的填充因子（fillfactor）是一个介于10和100之间的百分数。

取值范围：10~100

- STORAGE_TYPE

  指定GSI的存储类型。

  取值范围：USTORE

- INDEXSPLIT

  控制选择UBtree合适分裂点的方式，与Btree相同。

  取值范围：DEFAULT、INSERTPT

- enable_tde

  指定该索引为加密索引。数据库会自动将加密索引中的数据先加密再存储。使用该参数前，请确保已通过GUC参数enable_tde开启透明加密功能，并通过GUC参数tde_key_info设置访问密钥服务的信息，以及基表也设置了enable_tde属性。在《开发指南》中“特性使用指导 > 透明数据加密”章节可获取该参数的详细使用方法。本参数仅支持btree，ubtree索引，不支持hash等其他索引。

  取值范围：on/off。设置enable_tde=on时，key_type、tde_cmk_id、dek_cipher参数由数据库自动生成。

  默认值：off

- encrypt_algo

  指定加密索引的加密算法，需与enable_tde结合使用。

  取值范围：字符串，有效值为：AES_128_CTR，SM4_CTR、AES_128_CTR_SHA256_HMAC、SM4_CTR_SM3_HMAC或AES_128_GCM。

  默认值：如果enable_tde=on，GUC参数tde_encrypt_config中子参数table_algorithm不为空，默认值与table_algorithm相同；如果table_algorithm为空，默认值为AES_128_CTR。如果enable_tde=off，默认值为空。

- dek_cipher

  数据密钥的密文。该参数不支持手动设置或更改。用户为索引设置enable_tde参数后，索引会自动复制基表的dek_cipher参数。

  取值范围：字符串

  默认值：空

- key_type

  主密钥的类型。该参数不支持手动设置或更改。用户为索引设置enable_tde参数后，索引会自动复制基表的key_type参数。

  默认值：空

- cmk_id

  主密钥的ID。该参数不支持手动设置或更改。用户为索引设置enable_tde参数后，索引会自动复制基表的cmk_id参数。

  取值范围：字符串

  默认值：空

- opt_with_pkey

  指定当前创建的全局二级索引为主键优化索引。该参数仅支持创建时指定，不支持手动设置或更改。

  取值范围：on/off。

版权所有 © 华为云计算技术有限公司

默认值：off

说明

- 允许通过opt_with_pkey选项创建主键优化的全局二级索引（下称主键优化GSI）。当表发生更新时，如果更新列不涉及该主键优化GSI的列，则不更新该主键优化GSI，从而提升更新性能。
- 创建/重建主键优化GSI时，要求表上存在生效（USABLE）状态的非GSI主键索引。
- 集群升级未提交时，不允许创建主键优化GSI。
- 无法创建带有主键优化的表达式索引GSI和部分索引GSI。
- 当目标表上存在主键优化GSI时，主键无法直接删除，只能进行级联删除（CASCADE）。
- 当目标表上存在主键优化GSI时，失效（UNUSABLE）主键索引，会将所有主键优化GSI级联失效。
- 主键优化状态（opt_with_pkey）仅支持在创建时指定，无法更改。

– new_partbucket_gsi

指定当前在分区hashbucket表上创建的全局二级索引是否使用新的数据结构。该参数在创建全局二级索引时内部自动指定。

取值范围：on

默认值：on

说明

- 在首次升级至支持new_partbucket_gsi选项的版本的升级观察期，new_partbucket_gsi选项不支持指定。
- 当尝试在非全局二级索引或非分区hashbucket表上的全局二级索引上指定new_partbucket_gsi选项时，将报错。
- REINDEX不会影响new_partbucket_gsi选项的值。
- new_partbucket_gsi选项的值不支持使用ALTER语法进行更改。

- **[ILM ADD POLICY ROW STORE COMPRESS { TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]**

  – 创建GSI索引时，可以调用ILM ADD POLICY添加跨页透明压缩策略。比如：
  ```
  CREATE GLOBAL INDEX gsi_idx on t1(a) ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT AFTER 3 HOUR OF NO MODIFICATION;
  ```
  表示创建gsi_idx索引并添加跨页透明压缩策略：策略级别为high，冷热管理时间间隔为3小时。

  其中，AFTER n { hour | day | month | year } OF NO MODIFICATION表示n小时/天/月/年没有修改的压缩组。

  – 创建索引时，可以调用ILM ADD POLICY ROW STORE COMPRESS NONE显式指定索引不添加跨页透明压缩策略。

- **TABLESPACE tablespace_name**

  指定索引的表空间，如果没有声明则使用默认的表空间。

  取值范围：已存在的表空间名。

- **VISIBLE**

  设置索引状态为可见，此选项为默认的。

版权所有 © 华为云计算技术有限公司

说明

- VISIBLE仅支持在ORA模式数据库下（即sql_compatibility = 'ORA'）设置，在其他数据库兼容模式下不支持。
- 当disable_keyword_options参数设置为“visible”时，此关键字不支持使用。
- 升级未提交阶段，不支持使用此关键字。

- **INVISIBLE**

  设置索引状态为不可见。

  说明

  - INVISIBLE仅支持在ORA模式数据库下（即sql_compatibility = 'ORA'）设置，在其他数据库兼容模式下不支持。
  - 当disable_keyword_options参数设置为“invisible”时，此关键字不支持使用。
  - 升级未提交阶段，不支持使用此关键字。

- **WHERE predicate**

  创建一个部分索引。部分索引是一个只包含表的一部分记录的索引，通常是该表中比其他部分数据更有用的部分。例如，有一个表，表里包含已记账和未记账的定单，未记账的定单只占表的一小部分而且这部分是最常用的，此时就可以通过只在未记账部分创建一个索引来改善性能。另外一个可能的用途是使用带有UNIQUE的WHERE强制一个表的某个子集的唯一性。

  取值范围：predicate表达式只能引用表的字段，它可以使用所有字段，而不仅是被索引的字段。目前，子查询和聚集表达式不能出现在WHERE子句里。不建议使用int等数值类型作为predicate，因为int等数值类型可以隐式转换为BOOLEAN值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

示例

```
--创建包含3列的基表test1。
gaussdb=# CREATE TABLE test1(c1 int, c2 int, c3 int);

--在test表的c2列上创建GSI，并包含c3列，基于c2列哈希分布。
gaussdb=# CREATE GLOBAL INDEX idx_gsi_1 ON test1(c2) CONTAINING(c3) DISTRIBUTE BY HASH(c2);

--创建包含3列的基表test2。
gaussdb=# CREATE TABLE test2(c1 int, c2 int, c3 int);

--在test2表的c2列上创建GSI，并包含c3列，默认基于c2列哈希分布。
gaussdb=# CREATE GLOBAL INDEX idx_gsi_2 ON test2(c2) CONTAINING(c3);

--创建包含3列的基表test3。
gaussdb=# CREATE TABLE test3(c1 int, c2 int, c3 int);

--在test3表的c2列上创建UNIQUE形式的GSI，默认基于c2列哈希分布。
gaussdb=# CREATE GLOBAL UNIQUE INDEX idx_gsi_3 ON test3(c2) DISTRIBUTE BY HASH(c2);

--删除索引。
gaussdb=# DROP INDEX idx_gsi_1;
gaussdb=# DROP INDEX idx_gsi_2;
gaussdb=# DROP INDEX idx_gsi_3;

--删除表。
gaussdb=# DROP TABLE test1;
gaussdb=# DROP TABLE test2;
gaussdb=# DROP TABLE test3;
```

相关链接

ALTER INDEX，**CREATE INDEX**，**DROP INDEX**

版权所有 © 华为云计算技术有限公司

1.13.9.24 CREATE GROUP

功能描述

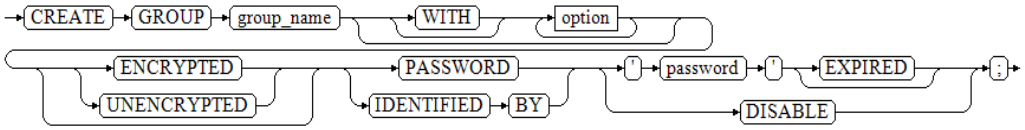
创建一个新用户组。

注意事项

CREATE GROUP为**CREATE ROLE**的别名，非SQL标准语法，不推荐使用，建议用户直接使用**CREATE ROLE**。

语法格式

```
CREATE GROUP group_name [ [ WITH ] option [ ... ] ]
    [ ENCRYPTED | UNENCRYPTED ] { PASSWORD | IDENTIFIED BY } { 'password' [ EXPIRED ] | DISABLE };
```



其中可选项option子句语法为：

```
{{SYSADMIN | NOSYSADMIN}
    | {MONADMIN | NOMONADMIN}
    | {OPRADMIN | NOOPRADMIN}
    | {AUDITADMIN | NOAUDITADMIN}
    | {CREATEDB | NOCREATEDB}
    | {USEFT | NOUSEFT}
    | {CREATEROLE | NOCREATEROLE}
    | {INHERIT | NOINHERIT}
    | {LOGIN | NOLOGIN}
    | {REPLICATION | NOREPLICATION}
    | {PERSISTENCE | NOPERSISTENCE}
    | CONNECTION LIMIT connlimit
    | VALID BEGIN 'timestamp'
    | VALID UNTIL 'timestamp'
    | RESOURCE POOL 'respool'
    | USER GROUP 'groupuser'
    | PERM SPACE 'spacelimit'
    | TEMP SPACE 'tmpspacelimit'
    | SPILL SPACE 'spillspacelimit'
    | NODE GROUP logic_group_name
    | IN ROLE role_name [, ...]
    | IN GROUP role_name [, ...]
    | ROLE role_name [, ...]
    | ADMIN role_name [, ...]
    | USER role_name [, ...]
    | SYSID uid
    | DEFAULT TABLESPACE tablespace_name
    | PROFILE DEFAULT
    | PROFILE profile_name
    | PGUSER}
```

参数说明

具体说明请参见**CREATE ROLE**章节中的**参数说明**。

示例

```
--创建用户组，与CREATE ROLE效果一样。
gaussdb=# CREATE GROUP test_group WITH PASSWORD "********";
```


版权所有 © 华为云计算技术有限公司

```
--使用CREATE ROLE创建角色，默认不能登录数据库。
--可使用ALTER ROLE role_name WITH LOGIN语句让用户可以登录数据库。
gaussdb=# CREATE ROLE test_role WITH PASSWORD "********";

--使用CREATE USER创建用户，自动创建同名模式，有登录的权限。
gaussdb=# CREATE USER test_user WITH PASSWORD "********";

--查看用户信息。
gaussdb=# \du test*
            List of roles
 Role name  |  Attributes  | Member of
------------+--------------+-----------
 test_group | Cannot login | {}
 test_role  | Cannot login | {}
 test_user  |              | {}

--查询CREATE USER命令自动创建的模式。
gaussdb=# \dn test*
    List of schemas
   Name    |   Owner
-----------+-----------
 test_user | test_user
(1 row)

--删除。
gaussdb=# DROP ROLE test_role;
gaussdb=# DROP GROUP test_group;
gaussdb=# DROP USER test_user;
```

相关链接

ALTER GROUP，**DROP GROUP**，**CREATE ROLE**

1.13.9.25 CREATE INCREMENTAL MATERIALIZED VIEW

功能描述

创建一个增量物化视图。可以使用REFRESH MATERIALIZED VIEW（全量刷新）和REFRESH INCREMENTAL MATERIALIZED VIEW（增量刷新）进行物化视图的数据刷新。

CREATE INCREMENTAL MATERIALIZED VIEW语句与CREATE TABLE AS语句功能相似，但其会持久化存储初始化查询语句，从而支持后续的数据刷新操作。物化视图具有与普通表相同的持久化属性，但不同于临时表，系统不支持创建临时类型的物化视图。

说明

具体的适用场景请参见《开发指南》中“基础概念 > 物化视图”章节的物化视图简介部分。

注意事项

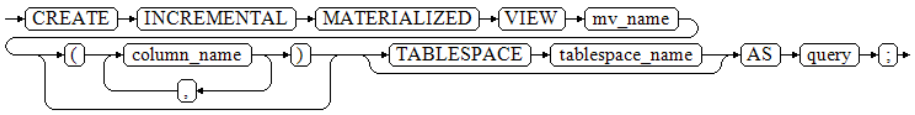
- 不支持在DATABASE LINK表、临时表或全局临时表上创建增量物化视图。
- 增量物化视图仅支持简单过滤查询和基表UNION ALL查询。
- 创建增量物化视图不支持WITH子句、GROUP BY子句、ORDER BY子句、LIMIT子句、WINDOW子句、DISTINCT算子、AGG算子，不支持除UNION ALL外的子查询。
- 创建增量物化视图不可指定分布列。

版权所有 © 华为云计算技术有限公司

- 创建增量物化视图后，基表中的绝大多数DDL操作不再支持。
- 不支持对增量物化视图进行IUD操作。
- 创建增量物化视图后，当基表数据发生变化时，需要使用刷新（REFRESH）命令保持物化视图与基表同步。
- Ustore引擎不支持物化视图的创建和使用。

语法格式

```
CREATE INCREMENTAL MATERIALIZED VIEW mv_name
    [ (column_name [, ...] ) ]
    [ TABLESPACE tablespace_name ]
    AS query;
```



参数说明

- **mv_name**

  创建的物化视图的名称，可以使用模式进行具体指定。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

  新物化视图中的一个列名（字段）。物化视图支持指定列，指定列需要和后面的查询语句结果的列数量保持一致；如果没有指定列名，将从查询的输出列名中获取列名。

  取值范围：字符串，要符合**标识符命名规范**。

- **TABLESPACE tablespace_name**

  可选。指定新建物化视图所属的表空间。如果没有指定，将使用默认表空间进行创建。

- **AS query**

  一个SELECT或者TABLE命令。这个查询将在一个安全受限的操作中运行。

示例

```
--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace1/tbs_data1';

--创建一个普通表。
gaussdb=# CREATE TABLE my_table (c1 int, c2 int);

--创建增量物化视图。
gaussdb=# CREATE INCREMENTAL MATERIALIZED VIEW my_imv (col1,col2) TABLESPACE tbs_data1 AS
SELECT * FROM my_table;

--基表写入数据。
gaussdb=# INSERT INTO my_table VALUES(1,1);

--查询增量物化视图数据。
gaussdb=# SELECT * FROM my_imv;
 col1 | col2
------+------
(0 rows)

--对增量物化视图my_imv进行增量刷新。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# REFRESH INCREMENTAL MATERIALIZED VIEW my_imv;

--查询增量物化视图数据。
gaussdb=# SELECT * FROM my_imv;
 col1 | col2
------+------
    1 |    1
(1 row)

--删除增量物化视图。
gaussdb=# DROP MATERIALIZED VIEW my_imv;

--删除普通表my_table。
gaussdb=# DROP TABLE my_table;

--删除表空间。
gaussdb=# DROP TABLESPACE tbs_data1;
```

相关链接

ALTER MATERIALIZED VIEW，**CREATE MATERIALIZED VIEW**，**CREATE TABLE**，**DROP MATERIALIZED VIEW**，**REFRESH INCREMENTAL MATERIALIZED VIEW**，**REFRESH MATERIALIZED VIEW**

1.13.9.26 CREATE INDEX

功能描述

在指定的表上创建索引。

索引可以提高数据库查询性能，但是不恰当的使用将导致数据库性能下降。建议仅在匹配如下某条原则时创建索引：

- 经常执行查询的字段。
- 在连接条件上创建索引，对于存在多字段连接的查询，建议在这些字段上建立组合索引。例如：SELECT * FROM t1 JOIN t2 ON t1.a=t2.a AND t1.b=t2.b;，可以在t1表上的a，b字段上建立组合索引。
- WHERE子句的过滤条件字段上（尤其是范围条件）。
- 在经常出现在ORDER BY、GROUP BY和DISTINCT后的字段。
- 分区表上创建索引会根据如下规则进行判断：
  - 创建索引时指定了GLOBAL/LOCAL关键字，则创建对应类型的索引。
  - 创建索引时指定了分区名，则创建LOCAL索引。
  - 对于UNIQUE索引，包含全部分区键时，创建LOCAL索引，否则创建GLOBAL索引。
  - 默认创建GLOBAL索引。

注意事项

- 基表为Hash分布时，若创建不包含基表分布键的主键或唯一索引，需要使用全局二级索引（CREATE GLOBAL INDEX），若创建包含基表分布键的主键或唯一索引，需要使用普通索引（CREATE INDEX），单DN部署形式下，使用全局二级索引或者普通索引均可创建成功；当基表为除Hash分布以外的其他分布形式时，主键或唯一索引只能使用普通索引（CREATE INDEX），即索引键必须包含基表分布键。

版权所有 © 华为云计算技术有限公司

- 索引自身也占用存储空间、消耗计算资源，创建过多的索引将对数据库性能造成负面影响（尤其影响数据导入的性能，建议在数据导入后再建索引）。
- 索引定义里的所有函数和操作符都必须是immutable类型，即其结果必须只能依赖于自身的输入参数，而不受任何外部的影响（如另外一个表的内容或者当前时间）。要在一个索引上或WHERE中使用用户定义函数，请将其标记为immutable类型函数。
- 分区表索引分为LOCAL索引与GLOBAL索引，LOCAL索引与某个具体分区绑定，而GLOBAL索引则对应整个分区表。
- 在同一列集合上，不能同时建立LOCAL索引和GLOBAL索引。
- 被授予CREATE ANY INDEX权限的用户，可以在PUBLIC模式和用户模式下创建索引。
- 如果基表是Hash分布/Range分布/List分布，创建唯一索引时必须包含基表的分布键且不能含有表达式。
- 如果表达式索引中调用的是用户自定义函数，则按照表Owner权限执行表达式索引函数。
- 所有用户都可以在自己的表中创建表达式索引。非初始用户不能在初始用户的表上创建表达式索引，初始用户可以对所有表创建表达式索引。三权分立关闭时，系统管理员可以在非初始用户表上创建表达式索引。三权分立开的时候，系统管理员可以在普通用户表上创建表达式索引，不可以在初始用户表、审计管理员表以及安全管理员表上创建表达式索引。
- 不支持XML类型数据作为普通索引、UNIQUE索引、GLOBAL索引、LOCAL索引、部分索引。
- 在线创建索引的类型只支持B-Tree索引和UB-Tree索引。索引创建形式只支持非分区表普通索引及分区表GLOBAL索引、LOCAL索引，不支持PCR UB-tree索引、二级分区与GSI。在线并行创建索引只支持Astore及Ustore的普通索引、GLOBAL索引、LOCAL索引。
- UGIN索引目前仅支持Ustore存储引擎，仅支持单列索引，其他场景说明请参见**Using ugin语法说明**。
- GIN、GiST索引仅支持Astore存储引擎，其他场景说明请参见语法说明。
- CREATE INDEX创建索引可能会改变表的访问方式从而导致查询执行计划改变。
- 由于B-Tree索引有结构性约束，需要保证单个页面有至少三个索引元组，因此有最长索引长度的约束。B-Tree和UB-Tree索引单个索引元组最长长度为2696字节，但是由于Ustore的UB-Tree索引中有事务信息，所以其单个索引元组实际的最长长度为2688字节。用户对超过这个长度的字段创建索引会报错，同样地，从Astore表迁移到Ustore表后对应的索引也存在创建失败的风险。
- 不支持创建包含rowid，rowno系统列的索引。
- 当创建索引的SQL语句过长（长度大于等于5250字符）时，会上报一条WARNING，建议缩短创建索引语句的长度。

  说明

  创建索引的SQL语句过长时，可能会在备份恢复索引时发生失败。由于备份恢复创建索引语句由gs_dump导出，gs_dump在导出索引定义时会对函数名、条件、列名进行添加引号、括号和列的类型的操作，原SQL语句会变得更长，进而导致该语句对应的pg_index元组不能存到一个页面上，造成创建索引失败。
- 在创建组合索引时，需根据查询条件和组合索引最左匹配的原则创建。

版权所有 © 华为云计算技术有限公司

说明

- 组合索引最左匹配原则：如果查询条件包含了组合索引的一列或者多列，那么组合索引的最左侧开始的连续列需要与查询条件匹配。
- 当查询条件为WHERE a = ?, b = ?,c = ?, d = ?或者WHERE a = ?, b = ?,c = ?等时，则为最佳的索引匹配；当查询条件为WHERE b = ?,c = ?, d = ?或者WHERE c = ?, d = ?等时，在代价计算后，可能也会执行索引idx_test_abcd，但在这种情况执行索引时，会扫描索引的全部页面，导致SQL性能并不理想。若出现类似场景，建议根据最左匹配原则，创建适合此查询条件的组合索引。

```
--创建表test。
gaussdb=# CREATE TABLE test(a int, b int, c int, d int, e int, f text);
创建组合索引。
gaussdb=# CREATE INDEX idx_test_abcd ON test(a,b,c,d);
```

语法格式

- 在表上创建索引。

```
CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ] [ IF NOT EXISTS ] [ [schema_name.] index_name ]
ON table_name [ USING method ]
    ({ { column_name [ ( length ) ] | ( expression ) } [ COLLATE collation ] [ opclass ] [ ASC | DESC ]
[ NULLS { FIRST | LAST } ] }[, ...] )
    [ INCLUDE ( column_name [, ...] ) ]
    [ WITH ( {storage_parameter = value} [, ... ] ) ]
    [ ILM ADD POLICY ROW STORE COMPRESS
 { TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO
MODIFICATION
 | NONE}]
    [ TABLESPACE tablespace_name ]
    [ COMMENT 'string' ] [ ... ]
    [ VISIBLE | INVISIBLE ]
    [ WHERE predicate ];
```

- 在分区表上创建索引

```
CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ] [ IF NOT EXISTS ] [ [schema_name.] index_name ]
ON table_name [ USING method ]
    ( { { column_name [ ( length ) ] | ( expression ) } [ COLLATE collation ] [ opclass ] [ ASC | DESC ]
[ NULLS LAST ] } [, ...] )
    [ LOCAL
        [ ( PARTITION index_partition_name [ FOR { partition_name | ( partition_value [, ...] ) } ]
[ TABLESPACE index_partition_tablespace ]
                [ ( [SUBPARTITION index_subpartition_name] [ FOR { partition_name | ( partition_value
[, ...] ) } ]  [ TABLESPACE index_partition_tablespace ]
                    [, ...] ) ]
            [, ...] ) ]
        | GLOBAL ]
    [ INCLUDE ( column_name [, ...] ) ]
    [ WITH ( { storage_parameter = value } [, ...] ) ]
    [ ILM ADD POLICY ROW STORE COMPRESS
 { TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO
MODIFICATION
 | NONE}]
    [ TABLESPACE tablespace_name ]
    [ COMMENT 'string' ] [ ... ]
    [ VISIBLE | INVISIBLE ];
```

参数说明

- **UNIQUE**

  创建唯一性索引，每次添加数据时检测表中是否有重复值。如果插入或更新的值会引起重复的记录时，将导致一个错误。

  目前只有行存表B-Tree及UB-Tree索引支持唯一索引。

- **CONCURRENTLY**

版权所有 © 华为云计算技术有限公司

以不阻塞DML的方式创建索引（加ShareUpdateExclusiveLock锁）。创建索引时，一般会阻塞其他语句对该索引所依赖表的访问。指定此关键字，可以实现创建过程中不阻塞DML。

- 此选项只能指定一个索引的名称。
- 普通CREATE INDEX命令可以在事务内执行。但是CREATE INDEX CONCURRENTLY不可以在事务内执行，也不可以在存储过程中执行。
- 对于临时表，支持使用CONCURRENTLY关键字创建索引，但是实际创建过程中，采用的是阻塞式的创建方式，因为没有其他会话会并发访问临时表，并且阻塞式创建成本更低。

版权所有 © 华为云计算技术有限公司

说明

- 创建索引时指定此关键字，Astore需要执行先后两次对该表的全表扫描来完成build，第一次扫描的时候创建索引，不阻塞读写操作；第二次扫描的时候合并更新第一次扫描到目前为止发生的变更。Ustore只需全表扫描一次，在全表扫描过程中并发DML产生的数据会被插入到以“index_oid_cctmp”命名的临时表中，扫描结束后将临时表中的数据合并到主索引中并删除临时表，完成索引创建。可使用GUC参数delete_cctmp_table控制在线创建索引结束后是否删除临时表：该参数默认为on，即默认删除临时表；设置为off时，保留临时表。
- Astore由于需要执行两次对表的扫描和build，而且必须等待现有的所有可能对该表执行修改的事务结束。这意味着该索引的创建比正常耗时更长，同时因此带来的CPU和I/O消耗对其他业务也会造成影响。Ustore虽只需全表扫描一次来完成索引创建，但上述消耗同样存在。
- 在线创建索引过程中可能会发生异常（例如用户手动取消、唯一索引键值重复、资源不足、启动线程失败、锁超时等场景），导致在线创建索引失败，这种情况可能会留下not valid状态的索引，为避免占用系统资源，会自动将其清理（清理失败索引前，需要先等该表上的事务结束，如果存在长事务，可能会出现一直等待的情况，用户取消该线程会残留失败索引。特别地，如果用户取消了在线创建索引线程，若此时在线创建索引的三个阶段均已结束，则不会进入失败索引清理流程，因此会残留下失败索引；如果此时在线创建索引的三个阶段还未结束，则会首先进入失败索引清理流程，如果用户再次取消，则会结束失败索引清理流程，残留下失败索引）。自动清理索引存在一个阶段需要等待数据库其他事务结束，因此可能出现锁超时、死锁等问题，导致清理索引失败。但对于严重错误，比如FATAL/PANIC/数据库故障等场景，需要用户手动清理索引及临时表。在线创建索引失败且自动残留清理失效的情况下，会残留下not valid的索引，这些残留索引可能是ready或者not ready状态（取决于在线创建索引是在哪一个阶段失败的，比如在第一阶段失败索引是not ready状态，在第三阶段失败索引是ready状态）。如果残留索引是ready状态，DML仍会维护此残留索引，且维护此索引过程中可能会产生其他错误（比如索引列大小超过最大值，违反唯一索引约束等）。为了避免残留索引占用系统资源和产生用户预期外报错，需要用户尽快手动将其删除。分布式中只有DN节点会自动清理失败索引（如果DN节点发生异常时，在线创建索引的三个阶段已经执行完成，此时也不会自动清理失败索引），coordinator node节点的失败索引需要手动清理。严重错误场景会导致分布式部分节点中无法匹配失败索引的元信息，DROP INDEX语句无法删除这些索引，请使用DROP INDEX IF EXISTS进行删除。
- Astore由于在第二次扫描之后，索引构建必须等待任何持有早于第二次扫描拿的快照的事务终止，而且建索引时加的ShareUpdateExclusiveLock锁（4级）会和大于等于4级的锁冲突，在创建这类索引时，容易引发卡住（hang）或者死锁问题。例如：
  - 两个会话对同一个表创建CONCURRENTLY索引，会引起死锁问题。
  - 两个会话中，一个会话对表创建CONCURRENTLY索引，另一个会话进行DROP TABLE，会引起死锁问题。
  - 三个会话中，会话1先对表a添加锁并不提交，会话2接着对表b创建CONCURRENTLY索引，会话3再次对表a执行写入操作，在会话1事务提交之前，会话2会一直被阻塞。
  - 创建CONCURRENTLY索引与同一个表的TRUNCATE操作并发，会引起死锁问题。
  - 将事务隔离级别设置成可重复读（默认为读已提交），创建两个会话，会话1进行对表a执行写入操作且不提交，会话2对表b创建CONCURRENTLY索引，在会话1事务提交之前，会话2会一直被阻塞。
- 索引构建过程中或者构建失败的情况下，需要确认索引进度或状态，可以通过查询函数gs_get_index_status('schema_name', 'index_name')进行确认当前所有节点上索引的状态，其中入参为schema_name和index_name，分别用来指定索引的模式名称和索引名称，返回值为node_name，indisready和indisvalid，分别表示节点名称、索引在该节点上是否可插入，以及索引在该节点上是否可用，只有当所有节点indisready和indisvalid均为true的情况下，索引才是“可用的”，否则请等待索引创建完成，或者构建失败情况下，删除索引重新创建。
- 在I/O、CPU、网络带宽等资源不受限的情况下，在线创建索引对业务性能的劣化一般可以控制在10%以内，但在特殊场景下劣化可能会超过此数值。这是因为在线创建索引本身是一种消耗I/O、CPU、网络带宽等资源较多的长事务，需要比离线创建索引消耗更多的资源。在线创建索引事务持续时间越长，对业务性能的影响越大。在线创建索引

版权所有 © 华为云计算技术有限公司

时间与基表数据量、并发DML产生的数据量正相关，在I/O、CPU、网络带宽等资源不受限的情况下，在线创建索引时间大约是离线创建索引的2~6倍，但当并发事务量较大（>10000TPS）或存在资源争抢的情况时，可能会超过此数值。如果在线创建索引期间存在长事务，还要加上长事务运行时间。在Astore及Ustore模式下，可以使用并行创建索引来缩短创建索引时间；在线并行创建索引性能随着并行工作线程数量增加而提升到一定值后稳定，相比串行创建索引性能一般可提升30%左右。建议在业务低谷期进行在线创建索引，以避免对业务造成较大影响。同时避免在线创建索引期间执行长事务。虽然在线创建索引在一定程度上提供了业务不中断的能力，但仍然需要谨慎实施。

- 在线创建唯一索引（CREATE UNIQUE INDEX CONCURRENTLY）时，会扫表检查表上数据是否满足唯一性，如果在线创建索引扫表过程中表上数据不满足唯一性，在线创建索引就会报错退出。注意在线创建索引过程中，表上数据是否满足唯一性可能会发生变化，是否失败取决于扫表过程中是否扫到了重复的数据。考虑以下情况：（1）扫表过程中插入了元组A和元组B，且它们的索引列相同，在线创建索引扫表时同时扫到了A和B，则可能会报违反唯一性，但是如果后续B被删除了，则可能会出现在线创建索引过程中报错违反唯一性，但在线创建索引失败退出后查找表上数据发现满足唯一性的情况。（2）在线创建索引过程中，先插入元组A，后删除A，再插入A，使用SnapshotNow扫表，则A可能会被扫表两次，可能会违反唯一性约束。在线创建索引过程中Astore使用SnapshotMVCC扫表，Ustore使用SnapshotNow扫表。

- **IF NOT EXISTS**
  - 指定IF NOT EXISTS关键字时，创建索引前会在当前Schema中查找是否已存在名字相同的relation。若存在同名relation，则不会新建，返回NOTICE提示。
  - 未指定IF NOT EXISTS关键字时，若Schema中存在同名relation，返回ERROR告警。
- **schema_name**

  模式的名称。

  取值范围：已存在模式名。
- **index_name**

  要创建的索引名，不能包含模式名，索引的模式与表相同。

  取值范围：字符串，要符合**标识符命名规范**。
- **table_name**

  需要创建索引的表的名称，可以使用模式进行具体指定。

  取值范围：已存在的表名。
- **USING method**

  指定创建索引的方法。

  取值范围：
  - btree：B-Tree索引使用一种类似于B+Tree的结构存储数据的键值，通过这种结构能够快速的查找索引。B-Tree适合支持比较查询以及范围查询。
  - ubtree：仅供Ustore表使用的多版本B-Tree索引，索引页面上包含事务信息，并能自主回收页面。UB-Tree索引默认开启INSERTPT功能。
  - ugin：通用倒排索引，一种仅供Ustore表使用的GIN索引。

版权所有 © 华为云计算技术有限公司

说明

- UGIN索引当前支持字符类型的opclass：ugin_trgm_ops。
  - ugin_trgm_ops不是默认opclass。
  - ugin_trgm_ops支持text、varchar、nvarchar2、clob数据类型的索引键，以及这四种类型的别名数据类型，不支持char(n)类型（bpchar）。
  - 指定了ugin_trgm_ops的索引键，其字符序不允许为M-Compatibility模式数据库的字符序（如：utf8mb4_bin、utf8mb4_general_ci、utf8mb4_unicode_ci、gbk_chinese_ci、gbk_bin、gb18030_chinese_ci、gb18030_bin等），否则创建索引时将报错。
- UGIN索引当前支持文本检索类型tsvector的opclass：ugin_tsvector_ops。
  - ugin_tsvector_ops是默认操作符类。
  - ugin_tsvector_ops支持tsvector数据类型的索引键，支持@@/@@@操作符进行索引扫描。
- UGIN索引当前支持数组类型的opclass：_bool_ops、_bpchar_ops、_char_ops、_float4_ops、_float8_ops、_int2_ops、_int4_ops、_int8_ops、_interval_ops、_numeric_ops、_oid_ops、_text_ops、_time_ops、_timetz_ops、_timestamp_ops、_timestamptz_ops、_varchar_ops。
  - 上述数组类型的opclass都是默认操作符类。
  - 支持下列数组类型创建UGIN索引：bool[]、char[]、"char"[]、float4[]、float8[]、int2[]、int4[]、int8[]、interval[]、numeric[]、oid[]、text[]、time[]、timestamp[]、timestamptz[]、timetz[]、varchar[]。
  - 支持下列操作符：“&&”、“@>”、“<@”、“=”。
- 只支持单列索引，索引键不支持ASC/DESC语法，不支持NULLS FIRST/LAST语法，不支持INCLUDE语法。
- 只支持Bitmap Index Scan执行计划扫描。
- 只支持GUC参数gin_fuzzy_search_limit为默认0值，若非0值，UGIN索引扫描时将报错。
- 不支持hashbucket表、hashbucket扩容以及不支持段页式。
- 只支持普通表索引和LOCAL索引，不支持GLOBAL索引。不支持作为unlog表和临时表的索引。
- 不支持在线重建、在线创建、在线删除、在线VACUUM FULL等操作（在线相关都不支持），不支持并行创建。
- 不支持闪回查询使用UGIN索引。
- 不支持作为主键、唯一键、虚拟索引，不支持透明数据加密和全密态数据库。
- 只支持LIKE模糊查询使用UGIN索引，其他情况不支持使用UGIN索引。
- 不支持在sql_compatibility = 'M'的数据库中创建和使用。
- UGIN索引支持所有字符集类型的字符串，但需确保数据库的字符集设置（server_encoding、LC_CTYPE、LC_COLLATE）与数据定义一致。这是因为分词算法中，一个字符的概念由字符集决定，而字母由LC_CTYPE决定，数据库的字符集设置不匹配会导致分词异常。

– gin：通用倒排索引，可以处理包含多个键的值（例如数组）。

版权所有 © 华为云计算技术有限公司

说明

- GIN索引支持的数据类型及索引操作符如下：
  - 数组类型：abstime[]、bit[]、bool[]、bpchar[]、bytea[]、char[]、cidr[]、date[]、float4[]、float8[]、inet[]、int2[]、int4[]、int8[]、interval[]、macaddr[]、money[]、name[]、numeric[]、oid[]、oidvector[]、reltime[]、text[]、time[]、timestamp[]、timestamptz[]、timetz[]、tinterval[]、varbit[]以及varchar[]。

    可索引的操作符：=、&&、@>以及<@。
  - 文本搜索类型：tsvector。

    可索引的操作符：@@和@@@。
  - JSON/JSONB类型：jsonb。

    对于JSONB类型，提供两个可索引的操作符类（jsonb_ops和jsonb_hash_ops），默认的可索引的操作符类（jsonb_ops）支持使用@>、?、?&和?|操作符查询，非默认的可索引的操作符类（jsonb_hash_ops）只支持索引@>操作符。
- 索引键不支持ASC/DESC语法，不支持NULLS FIRST/LAST语法，不支持INCLUDE语法。
- 只支持Bitmap Index Scan执行计划扫描。
- 支持普通表、本地临时表索引和分区表的LOCAL索引，不支持GLOBAL索引，不支持作为全局临时表和unlogged表的索引。
- 不支持hashbucket表和hashbucket扩容。
- 不支持在线重建、在线创建、在线删除、在线VACUUM FULL等操作，不支持并行创建。
- 不支持作为主键，不支持作为唯一键，不支持作为虚拟索引，不支持透明加密和全密态加密。
- 不支持索引压缩。
- array、jsonb、tsvector数据类型不支持收集统计信息，因此，在这些类型上创建的GIN索引的代价估算存在误差。

– gist：适用于几何和地理等多维数据类型和集合数据类型。目前支持的数据类型包括box、point、poly、circle、tsvector、tsquery和range。

版权所有 © 华为云计算技术有限公司

说明

- GiST索引支持的数据类型及索引操作符如下：
  - 矩形box：可索引的操作符包括<<、&<、&&、&>、>>、~=、@>、<@、&<|、<<|、|>>、|&>、~以及@。
  - 圆circle：可索引的操作符包括<<、&<、&&、&>、>>、~=、@>、<@、&<|、<<|、|>>以及|&>。
  - 点point：可索引的操作符包括<<、>>、~=、<@、<^、>^以及<->。
  - 多边形polygon：可索引的操作符包括<<、&<、&>、>>、~=、<<|、&<|、|&>以及|>>。
  - 范围类型range，包括int4range、int8range、numrange、tsrange、tstzrange、daterange、自定义range类型：可索引的操作符包括=、&&、@>、<@、<<、>>、@<、&>以及-|-。
  - 文本搜索类型tsquery：可索引的操作符包括<@和@>。
  - 文本搜索类型tsvector：可索引的操作符包括@@。
- GiST支持的数据类型都不支持收集统计信息，因此在这些类型上创建的GiST索引的代价估算存在误差。
- 仅支持Astore存储引擎，不支持Ustore、hashbucket、透明加密、全密态加密、物化视图、并行创建索引场景，不支持在线重建、在线创建、在线删除、在线VACUUM FULL场景。
- 支持Bitmap Index Scan和Index Scan执行计划扫描。
- 支持普通表、本地临时表索引和分区表的分区索引，不支持作为全局临时表索引，不支持作为unlogged表索引。
- 不支持ASC/DESC语法，不支持NULLS FIRST/LAST语法，不支持INCLUDE语法。
- 不支持作为主键，不支持作为唯一键，不支持作为虚拟索引。
- gs_check_table/gs_check_index不支持GiST索引。
- daterange在ORA兼容模式下不支持@>(daterange, date)。
- 不支持索引压缩。

说明

- btree与ubtree是与表的存储类型ASTORE/USTORE强相关，在创建索引时指定索引类型与主表不对应时会自动进行转换。
- 行存表（ASTORE存储引擎）支持的索引类型为btree（行存表默认值）、gin以及gist。行存表（USTORE存储引擎）支持的索引类型为ubtree以及ugin。
- gin和ugin索引使用建议：
  - 索引不建议在高并发场景下修改，因为同步融合增量数据会出现性能波动。
  - 索引不建议高频修改，其空间回收和统计信息更新依赖于VACUUM操作，如果在两次VACUUM之间进行频繁增删，可能会导致空间膨胀和执行计划不准确。
  - 增量缓存区扫描速度不如B-tree扫描，如果扫描性能不佳，建议用户手动清理增量缓存区。
  - 索引只适合大数据量中查询返回少量数据的情况，不适合用于返回大量数据的情况。

- **column_name**

  表中需要创建索引的列的名称（字段名）。

  如果索引方式支持多字段索引，可以声明多个字段。全局索引最多可以声明31个字段，其他索引最多可以声明32个字段。

- **column_name (length)**

  创建一个基于该表一个字段的前缀键索引，column_name为前缀键的字段名，length为前缀长度。

版权所有 © 华为云计算技术有限公司

前缀键将取指定字段数据的前缀作为索引键值，可以减少索引占用的存储空间。含有前缀键字段的部分过滤条件和连接条件可以使用索引。

说明

- 前缀键支持的索引方法包含B-Tree索引（btree）和UB-Tree索引（ubtree）。
- 前缀键的字段的数据类型必须是二进制类型或字符类型（不包括特殊字符类型）。
- 前缀长度length必须是不超过2676的正整数，并且不能超过字段的最大长度。对于二进制类型，前缀长度以字节数为单位。对于非二进制字符类型，前缀长度以字符数为单位。前缀键值的实际长度受内部页面限制，若字段中含有多字节字符、或者一个索引上有多个键，索引行长度可能会超限，导致报错，设定较长的前缀长度时请考虑此情况。
- CREATE INDEX语法中，不支持以下关键字作为前缀键的字段名称：COALESCE、EXTRACT、GREATEST、LEAST、LNNVL、NULLIF、NVL、NVL2、OVERLAY、POSITION、REGEXP_LIKE、SUBSTRING、TIMESTAMPDIFF、TREAT、TRIM、XMLCONCAT、XMLELEMENT、XMLEXISTS、XMLFOREST、XMLPARSE、XMLPI、XMLROOT、XMLSERIALIZE。
- 前缀键属于一种特殊的表达式键，部分未说明的约束和限制，与表达式键一致，请参考表达式索引的说明。
- 前缀索引作为一种表达式索引，与表达式索引一致，不支持作为唯一索引和主键。

- **expression**

  创建一个基于该表的一个或多个字段的表达式索引，通常必须写在括号中。如果表达式有函数调用的形式，可以省略括号。

  表达式索引可用于获取对基本数据的某种变形的快速访问。比如，一个在upper(col)上的函数索引将允许WHERE upper(col) = 'JIM'子句使用索引。

  在创建表达式索引时，如果表达式中包含IS NULL子句，则这种索引是无效的。此时，建议用户尝试创建一个部分索引。

- **COLLATE collation**

  指定列的排序规则（该列必须是可排列的数据类型）。如果没有指定，则使用默认的排序规则。排序规则可以使用“SELECT * FROM pg_collation;”命令从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。

  说明

  MYSQL兼容模式（即sql_compatibility = 'MYSQL'）下，当使用MYSQL兼容字符序时，collation需保持索引字符序和列字符序一致，否则创建索引失败并报错。

- **opclass**

  操作符类的名称。对于索引的每一列可以指定一个操作符类，操作符类标识了索引列使用的操作符。例如：一个B-Tree索引在一个四字节整数上可以使用int4_ops；这个操作符类包括四字节整数的比较函数。实际上对于列上的数据类型默认的操作符类是足够用的。操作符类主要用于一些有多种排序的数据。例如：用户想按照绝对值或者实数部分排序一个复数。能通过定义两个操作符类然后在建立索引时选择合适的类。另外，如果包含字符串类型（varchar、varchar2、text等）的索引的COLLATE的值不是C或者POSIX，但希望索引能够支持前缀匹配，则需要指定varchar_pattern_ops选项。

- **ASC**

  指定按升序排序（默认）。

- **DESC**

  指定按降序排序。

- **NULLS FIRST**

  指定空值在排序中排在非空值之前，当指定DESC排序时，本选项为默认的。

版权所有 © 华为云计算技术有限公司

- **NULLS LAST**

  指定空值在排序中排在非空值之后，未指定DESC排序时，本选项为默认的。

- **LOCAL**

  指定创建的分区索引为LOCAL索引。

- **GLOBAL**

  指定创建的分区索引为GLOBAL索引，当不指定LOCAL、GLOBAL关键字时，默认创建GLOBAL索引。

- **INCLUDE（column_name [, ...]）**

  可选的INCLUDE子句指定将一些非键列（non-key columns）包含在索引中。非键列不能用于作为索引扫描的加速搜索条件，同时在检查索引的唯一性约束时会忽略非键列。

  仅索引扫描（Index Only Scan）可以直接返回非键列中的内容，而不必去访问索引所对应的堆表。

  将非键列添加为INCLUDE列需要谨慎操作，尤其是对于宽列。如果索引元组超过索引类型允许的最大大小，数据将插入失败。需要注意的是，任何情况下为索引添加非键列都会增加索引的空间占用，从而可能减慢搜索速度。

  目前只有UB-Tree索引访问方式支持该特性。非键列会被保存在与堆元组对应的索引叶子元组中，不会包含在索引上层页面的元组中。

- **PARTITION子句**

  ```
  PARTITION index_partition_name [ FOR { partition_name | ( partition_value [, ...] ) } ] [ TABLESPACE index_partition_tablespace ]
  ```

  如果分区子句指定了FOR partition_name或FOR partition_value，此时创建的分区索引带有分类索引属性；如果未指定FOR partition_name或FOR partition_value，分区子句个数必须和目标表分区个数一一对应。

  说明

  - 分类索引是分区表LOCAL索引的一种，即只在单个分区上创建索引分区的LOCAL索引，与普通LOCAL索引相比，普通LOCAL索引需要在所有表分区上创建对应的索引分区。
  - 仅支持在单个分区上创建分类索引，包括一级分区表的单个一级分区，以及二级分区表的单个一级分区和单个二级分区。
  - 分类索引仅支持B-Tree索引和UB-Tree索引。
  - 分类索引不支持UNIQUE属性。
  - 当前不支持GLOBAL索引。
  - 创建的分类索引会带有sparsely_partitioned=true选项，在二级分区表的一级分区上创建分类索引会带有level1_sparsely_partitioned=true选项，sparsely_partitioned=true选项和level1_sparsely_partitioned=true选项不可以通过CREATE TABLE或ALTER TABLE语句将非分类索引显式指定为分类索引。如果创建的索引为分类索引，即使声明sparsely_partitioned=false或level1_sparsely_partitioned=false也会被忽略。
  - 从505.0.0之前版本升级到505.0.0及更高版本时，在升级观察期期间，不支持创建分类索引。
  - 从506.1.0之前版本升级到506.1.0及更高版本时，在升级观察期期间，不支持为二级分区表的一级分区创建分类索引。

  - **PARTITION index_partition_name**

    索引分区的名称。

    取值范围：字符串，要符合**标识符命名规范**。

  - **FOR partition_name**

    指定的目标分区名。如果指定的分区不存在则报错。

版权所有 © 华为云计算技术有限公司

- **FOR (partition_value [, ...])**

  指定的分区键值。分区键值需要放入分区键值FOR列表中，如果存在多列分区键，则需要指定多个partition_value。

- **TABLESPACE index_partition_tablespace**

  索引分区的表空间。

  取值范围：如果没有声明，将使用分区表索引的表空间index_tablespace。

- **WITH ({storage_parameter = value} [, ...])**

  指定索引方法的存储参数。

  取值范围：

  说明

  - Psort、UGIN和GIN之外的索引都支持FILLFACTOR参数。
  - 只有UB-Tree索引支持INDEXSPLIT参数。
  - 只有非分区表的B-Tree索引支持DEDUPLICATION参数。
  - 只有UB-Tree索引支持INDEX_TXNTYPE参数。
  - UGIN和GIN索引还支持gin_pending_list_limit、fastupdate、pending_pages、total_pages、entry_pages、data_pages和entries参数。
  - GiST索引还支持buffering参数。

  - STORAGE_TYPE

    表示索引所在的表的存储引擎类型，当索引指定的存储引擎类型与索引类型冲突时，会自动修改为正确的存储类型。仅支持B-Tree索引、UB-Tree索引、GIN索引和UGIN索引。该参数设置成功后就不再支持修改。

    取值范围：

    - USTORE，表示索引所在的表为Inplace-Update存储引擎。
    - ASTORE，表示索引所在的表为Append-Only存储引擎。

    默认值：Ustore表创建的索引默认为USTORE，Astore表创建的索引默认为ASTORE。

  - FILLFACTOR

    索引的填充因子（fillfactor），一个介于10和100之间的百分数。对于大并发插入且键值范围比较密集的场景，插入时同一个索引页面的竞争比较大，选择较小的填充因子更加合适。

    取值范围：10~100。

  - FASTUPDATE

    指定GIN索引/UGIN索引是否使用快速更新。UGIN索引暂时不支持关闭增量缓存区（pending list区），所以该参数仅支持设置为on，设置为off会报错。

    取值范围：on或off。

    默认值：on。

  - GIN_PENDING_LIST_LIMIT

    设置GIN索引/UGIN索引缓存区（pending list）容量的上限。当pending list的容量大于设置值时，会把pending list中的数据批量移动到索引数据结构中以进行清理。

    取值范围：64~2147483647，单位kB。

    默认值：取决于GUC参数gin_pending_list_limit的值（默认为4MB）。

版权所有 © 华为云计算技术有限公司

- CROSSBUCKET

  索引是否使用跨hashbucket索引。

  取值范围：on或off。

  默认值：off。

- ACTIVE_PAGES

  表示索引的页面数量，可能比实际的物理文件页面少，可以用于优化器调优。目前只对Ustore的分区表LOCAL索引生效，且会被VACUUM、ANALYZE更新（包括AUTO VACUUM）。不建议用户手动设置该参数，该参数在当前形态下不生效。

- DEDUPLICATION

  索引参数，设置索引是否对键值重复的元组进行去重压缩。在重复键值的索引较多时，开启参数可以有效降低索引占用空间。对主键索引和唯一索引不生效。非唯一索引且索引键值重复度很低或者唯一的场景，开启参数会使索引插入性能小幅度劣化。暂不支持分区表的LOCAL/GLOBAL索引。

  取值范围：布尔值，取决于GUC参数中enable_default_index_deduplication的值（默认为off）。

- INDEXSPLIT

  UB-Tree索引选择采取的分裂策略。

  取值范围：

  - INSERTPT：与B-Tree相同的分裂策略。
  - DEFAULT：可以在某些场景下显著降低索引空间占用。

  默认值：INSERTPT。

  设置建议：针对同一时间并发插入大量索引key值相同的数据的场景，使用INSERTPT分裂算法可能导致索引膨胀，建议修改分裂策略为DEFAULT。

  举例：

  ```
  CREATE INDEX idx ON tbl(col) WITH (indexsplit=default);
  ```

- INDEX_TXNTYPE

  UB-Tree索引类型，仅UB-Tree索引支持该参数，当该参数的取值为PCR时，可支持通过UB-Tree进行闪回查询。PCR版本UB-Tree索引当前不支持在线创建索引、全局二级索引、极致RTO回放和备机读的功能。当不指定该参数时，具体创建的索引类型通过GUC参数index_txntype进行控制。INDEX_TXNTYPE不支持ALTER INDEX INDEX_NAME SET (INDEX_TXNTYPE=PCR或RCR)进行修改。

  类型：字符串（不区分大小写）。

  取值范围：RCR或PCR。

  默认值：RCR。

  举例：

  ```
  CREATE UNIQUE INDEX t2_b_pkey ON t(b) WITH(index_txntype='pcr');
  ```

- STAT_STATE

  用于标识该索引的统计信息是否被锁定，如果被锁定，该索引的统计信息无法更新。该参数在当前形态下无效。

  取值范围：locked或unlock。

  默认值：unlock。

版权所有 © 华为云计算技术有限公司

- enable_tde

  指定该索引是否为加密索引，若是加密索引，数据库会自动将加密索引中的数据先加密再存储。

  使用该参数前，请确保已通过GUC参数enable_tde开启透明加密功能，并通过GUC参数tde_key_info设置访问密钥服务的信息，以及基表也设置了enable_tde属性。该参数的详细使用方法请参见《开发指南》中“特性使用指导 > 透明数据加密”章节。本参数仅支持B-Tree索引和UB-Tree索引，不支持其他索引。

  取值范围：on或off。设置enable_tde=on时，key_type、tde_cmk_id、dek_cipher参数由数据库自动生成。

  默认值：off。

- encrypt_algo

  索引的加密算法。用户为索引设置enable_tde参数后，索引会自动复制基表的encrypt_algo参数。该参数不支持手动设置或更改。

  取值范围：字符串；有效值为AES_128_CTR、SM4_CTR、AES_128_CTR_SHA256_HMAC、SM4_CTR_SM3_HMAC或AES_128_GCM。

  默认值：空。

- dek_cipher

  数据密钥的密文。该参数不支持手动设置或更改。用户为索引设置enable_tde参数后，索引会自动复制基表的dek_cipher参数。

  取值范围：字符串。

  默认值：空。

- key_type

  主密钥的类型。该参数不支持手动设置或更改。用户为索引设置enable_tde参数后，索引会自动复制基表的key_type参数。

  默认值：空。

- cmk_id

  主密钥的ID。该参数不支持手动设置或更改。用户为索引设置enable_tde参数后，索引会自动复制基表的cmk_id参数。

  取值范围：字符串。

  默认值：空。

- LPI_PARALLEL_METHOD

  索引参数，设置分区表LOCAL索引并行创建的方式。

  类型：字符串。

  取值范围：

  - PAGE：页面级并行创建索引，开启多个子线程执行数据的扫描和排序，每个子线程一次处理一个数据页面，扫描排序后，在主线程串行合并排序结果并将元组插入到索引中。
  - PARTITION：分区级并行创建索引，开启多个子线程，每个子线程负责一个分区中数据的扫描、排序、索引插入。
  - AUTO：根据分区表统计信息来预估页面级和分区级两种并行创建索引方式的代价，并选择代价较小的并行创建方式（统计信息可能会和实际数据情况有误差而导致计算结果不准确）。

版权所有 © 华为云计算技术有限公司

默认值：PAGE。

设置建议：当分区表数据在各分区分布均匀时，建议指定该参数为PARTITION。该参数仅支持Astore分区表的BTREE LOCAL索引，不支持分区表GLOBAL索引、非分区表索引、段页式表索引、hashbucket表索引以及在线创建索引。

举例：

```
CREATE INDEX idx ON tbl(col) WITH (lpi_parallel_method = 'partition');
```

- pending_pages、total_pages、entry_pages、data_pages、entries

  分别表示UGIN索引和GIN索引的快速缓存区的页面数、索引总页面数、分词树页面数、posting tree的页面数及分词总个数。

  说明

  - 以上参数属于UGIN索引和GIN索引独有的页面统计信息，通过ANALYZE命令更新，不允许用户通过CREATE INDEX/ALTER INDEX等方式手动设置。
  - CREATE TABLE LIKE INCLUDING {index | all} 不会继承索引的统计信息。
  - gs_dump不导出独有的索引信息。
  - pg_get_indexdef函数不获取索引独有的索引信息。
  - \d+ table_name不会显示独有的索引信息。
  - 事务级别的临时表不支持收集GIN索引的页面统计信息。

- buffering

  GiST索引是否使用缓冲构建技术来构建索引。

  取值范围：

  - off：缓冲被禁用。
  - on：缓冲被启用。
  - auto：缓冲最初被禁用，但当索引大小达到effective_cache_size时，会立即启用。

  默认值：auto。默认行为适用于多数情况，但如果输入数据已排序，关闭缓冲特性可以加速构建过程。

- wait_clean_cbi

  CBI索引中是否含有扩容bucket搬迁产生的残留元组。扩容重分布后会对含有残留元组的CBI索引设置（wait_clean_cbi=y），含有残留元组的CBI索引通过VACUUM或REINDEX等方式清理结束后自动修改为（wait_clean_cbi=n）。该参数只在扩容工具中使用，不支持手动设置或更改。

  取值范围：y或n。

  默认值：n。

- **[ILM ADD POLICY ROW STORE COMPRESS { TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]**

  - 创建索引时，可以调用ILM ADD POLICY添加跨页透明压缩策略。比如：

    ```
    CREATE INDEX i1 on t1(a) ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT AFTER 3 HOUR OF NO MODIFICATION;
    ```

    表示创建i1索引并添加跨页透明压缩策略：策略级别为high，冷热管理时间间隔为3小时。

  - AFTER n { hour | day | month | year } OF NO MODIFICATION：表示n小时/天/月/年没有修改的压缩组。

版权所有 © 华为云计算技术有限公司

  - 创建索引时，可以调用ILM ADD POLICY ROW STORE COMPRESS NONE显式指定索引不添加跨页透明压缩策略。
  - 显式指定索引压缩策略时，以指定策略为准（未指定压缩级别时默认取MEDIUM），未显式指定索引压缩策略时，索引是否压缩和压缩级别与表保持一致。

- **TABLESPACE tablespace_name**

  指定索引的表空间，如果没有指定，则使用默认的表空间。

  取值范围：已存在的表空间名。

- **WHERE predicate**

  创建一个部分索引。

  部分索引是一个只包含表的一部分记录的索引，通常是该表中比其他部分数据更有用的部分。

  取值范围：predicate表达式只能引用表的字段，可以使用所有字段，而不仅是被索引的字段。目前，子查询和聚集表达式不能出现在WHERE子句里。不建议使用INT等数值类型作为predicate，因为INT等数值类型可以隐式转换为布尔值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

  对于分区表索引，当创建索引带GLOBAL关键字，或者最终创建的索引类型为GLOBAL索引时，不支持带WHERE子句创建索引。

- **COMMENT 'string'**

  为索引添加注释。

  说明

  - 仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）有效。
  - 索引级注释支持的最大长度为1024字符。

- **VISIBLE**

  设置索引状态为可见，此选项为默认的。

  说明

  - VISIBLE仅支持在ORA模式数据库下（即sql_compatibility = 'ORA'）设置，在其他数据库兼容模式下不支持。
  - 当disable_keyword_options参数设置为visible时，此关键字不支持使用。
  - 升级未提交阶段，不支持使用此关键字。

- **INVISIBLE**

  设置索引状态为不可见。

  说明

  - INVISIBLE仅支持在ORA模式数据库下（即sql_compatibility = 'ORA'）设置，在其他数据库兼容模式下不支持。
  - 当disable_keyword_options参数设置为invisible时，此关键字不支持使用。
  - 升级未提交阶段，不支持使用此关键字。

示例

- 普通索引

```
-- 创建tbl_test1表。
gaussdb=# CREATE TABLE tbl_test1(
    id int,           --用户id
    name varchar(50), --用户姓名
```


版权所有 © 华为云计算技术有限公司

```
    postcode char(6)  --邮编
);

-- 创建表空间tbs_index1。
gaussdb=# CREATE TABLESPACE tbs_index1 RELATIVE LOCATION 'test_tablespace/tbs_index1';

-- 为表tbl_test1创建索引idx_test1指定表空间。
gaussdb=# CREATE INDEX idx_test1 ON tbl_test1(name) TABLESPACE tbs_index1;

-- 查询索引idx_test1信息。
gaussdb=# SELECT indexname,tablename,tablespace FROM pg_indexes WHERE indexname =
'idx_test1';
 indexname | tablename | tablespace
-----------+-----------+------------
 idx_test1 | tbl_test1 | tbs_index1
(1 row)

-- 删除索引。
gaussdb=# DROP INDEX idx_test1;

-- 删除表空间。
gaussdb=# DROP TABLESPACE tbs_index1;
```

- 唯一索引

```
-- 为表tbl_test1创建唯一索引idx_test2。
gaussdb=# CREATE UNIQUE INDEX idx_test2 ON tbl_test1(id);

-- 查询索引信息。
gaussdb=# \d tbl_test1
          Table "public.tbl_test1"
  Column  |         Type          | Modifiers
----------+-----------------------+-----------
 id       | integer               |
 name     | character varying(50) |
 postcode | character(6)          |
Indexes:
    "idx_test2" UNIQUE, btree (id) TABLESPACE pg_default

-- 删除索引。
gaussdb=# DROP INDEX idx_test2;
```

- 表达式索引

```
-- 为表tbl_test1创建一个表达式索引。
gaussdb=# CREATE INDEX idx_test3 ON tbl_test1(substr(postcode,2));

-- 查询索引信息。
gaussdb=# \d tbl_test1
          Table "public.tbl_test1"
  Column  |         Type          | Modifiers
----------+-----------------------+-----------
 id       | integer               |
 name     | character varying(50) |
 postcode | character(7)          |
Indexes:
    "idx_test3" btree (substr(postcode::text, 2)) TABLESPACE pg_default

-- 删除索引。
gaussdb=# DROP INDEX idx_test3;
```

- 部分索引

```
-- 为表tbl_test1中id不为为空的数据建立索引。
gaussdb=# CREATE INDEX idx_test4 ON tbl_test1(id) WHERE id IS NOT NULL;

-- 删除索引。
gaussdb=# DROP INDEX idx_test4;

-- 删除表。
gaussdb=# DROP TABLE tbl_test1;
```

- 分区索引

版权所有 © 华为云计算技术有限公司

```
-- 建表。
gaussdb=# CREATE TABLE student(id int, name varchar(20)) PARTITION BY RANGE (id) (
    PARTITION p1 VALUES LESS THAN (200),
    PARTITION pmax VALUES LESS THAN (MAXVALUE)
);

-- 创建LOCAL分区索引不指定索引分区的名称。
gaussdb=# CREATE INDEX idx_student1 ON student(id) LOCAL;

-- 查看索引分区信息，LOCAL索引分区数和表的分区数一致。
gaussdb=# SELECT relname FROM pg_partition WHERE parentid = 'idx_student1'::regclass;
   relname
-------------
 p1_id_idx
 pmax_id_idx
(2 rows)

-- 删除LOCAL分区索引。
gaussdb=# DROP INDEX idx_student1;

-- 创建GLOBAL索引。
gaussdb=# CREATE INDEX idx_student2 ON student(id) GLOBAL;

-- 查看索引分区信息，GLOBAL索引分区数和表的分区数不一致。
gaussdb=# SELECT relname FROM pg_partition WHERE parentid = 'idx_student2'::regclass;
 relname
---------
(0 rows)

-- 删除GLOBAL索引。
gaussdb=# DROP INDEX idx_student2;

-- 创建LOCAL表达式索引，不指定索引分区的名称。
gaussdb=# CREATE INDEX idx_student3 ON student(abs(id)) LOCAL;

-- 查看索引分区信息，LOCAL索引分区数和表的分区数一致。
gaussdb=# SELECT relname FROM pg_partition WHERE parentid = 'idx_student3'::regclass;
   relname
-------------
 p1_abs_idx
 pmax_abs_idx
(2 rows)

-- 删除LOCAL分区表达式索引。
gaussdb=# DROP INDEX idx_student3;

-- 创建GLOBAL表达式索引。
gaussdb=# CREATE INDEX idx_student4 ON student(abs(id)) GLOBAL;

-- 查看索引分区信息，GLOBAL表达式索引分区数和表的分区数不一致。
gaussdb=# SELECT relname FROM pg_partition WHERE parentid = 'idx_student4'::regclass;
 relname
---------
(0 rows)

-- 删除GLOBAL表达式索引。
gaussdb=# DROP INDEX idx_student4;

-- 删除表。
gaussdb=# DROP TABLE student;
```

- UGIN索引

```
-- 建表。
gaussdb=# CREATE TABLE test_trgm(t varchar(100)) WITH (STORAGE_TYPE=USTORE);

-- 插入数据。
gaussdb=# INSERT INTO test_trgm VALUES('qwertyu0001');
gaussdb=# INSERT INTO test_trgm VALUES('qwertyu0099');
gaussdb=# INSERT INTO test_trgm VALUES('2024新年快乐7days');
```


版权所有 © 华为云计算技术有限公司

```
-- 创建UGIN索引。
gaussdb=# CREATE INDEX trgm_idx ON test_trgm USING UGIN (t ugin_trgm_ops);

-- LIKE查询语句。
gaussdb=# SELECT * FROM test_trgm WHERE t LIKE '%24新%';

-- 删除表。
gaussdb=# DROP TABLE test_trgm CASCADE;

-- 数组类型创建UGIN索引。
-- 建表。
gaussdb=# CREATE TABLE test_array(a int, b bool[]) WITH (STORAGE_TYPE=USTORE);

-- 插入数据。
gaussdb=# INSERT INTO test_array VALUES(1, ARRAY[true, false]);

-- 创建UGIN索引。
gaussdb=# CREATE INDEX ugin_idx_array_bool ON test_array USING UGIN (b _bool_ops);

-- 查询数据。
gaussdb=# SELECT b FROM test_array WHERE b = ARRAY[true, false];
   b
-------
 {t,f}
(1 row)

-- 删除表。
gaussdb=# DROP TABLE test_array CASCADE;
```

- GIN索引

```
-- 建表。
gaussdb=# CREATE TABLE test_gin (id INT, info INT[]) WITH(STORAGE_TYPE = ASTORE);

-- 创建索引。
gaussdb=# CREATE INDEX test_gin_idx ON test_gin USING GIN(info);

-- 插入数据。
gaussdb=# INSERT INTO test_gin SELECT g, ARRAY[1, g % 5, g] FROM generate_series(1, 20000) g;

-- 查询语句。
gaussdb=# SELECT * FROM test_gin WHERE info @> '{2}' AND info @> '{22}' ORDER BY id, info;

-- 删除表。
gaussdb=# DROP TABLE test_gin CASCADE;
```

- GiST索引

```
-- 建表。
gaussdb=# CREATE TABLE gist_box_tbl(f1 point, f2 int)WITH(storage_type = astore);

-- 创建索引。
gaussdb=# CREATE INDEX gist_box_index ON gist_box_tbl USING gist(f1);

-- 插入数据。
gaussdb=# INSERT INTO gist_box_tbl VALUES(point(1,2),1);
gaussdb=# INSERT INTO gist_box_tbl VALUES(point(3,4),3);

-- 查询语句。
gaussdb=# SELECT count(*) FROM gist_box_tbl WHERE f1 ~= point(1, 2);

-- 删除表。
gaussdb=# DROP TABLE gist_box_tbl CASCADE;
```

相关链接

ALTER INDEX，**DROP INDEX**

版权所有 © 华为云计算技术有限公司

1.13.9.27 CREATE LANGUAGE

功能描述

定义一种新的过程语言。

注意事项

当前形态暂不支持使用该语法。

1.13.9.28 CREATE MASKING POLICY

功能描述

创建脱敏策略。

注意事项

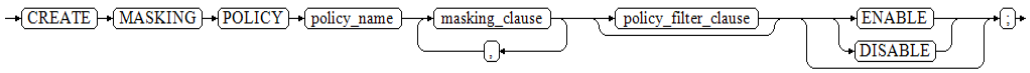
- 仅POLADMIN、SYSADMIN以及初始用户可以执行此操作。
- 需要开启安全策略开关（GUC参数enable_security_policy设置为on），脱敏策略才可以生效。

说明

在使用DATABASE LINK功能的场景下，客户端发起的DATABASE LINK请求，实际的发送方是服务端，发送端IP地址等相关的属性将是服务端的值。具体的使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

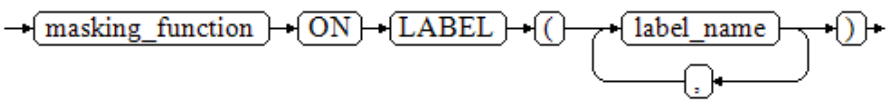
语法格式

```
CREATE MASKING POLICY policy_name masking_clause[, ...] [ policy_filter_clause ] [ENABLE | DISABLE];
```



- masking_clause：

  ```
  masking_function ON LABEL(label_name[, ...])
  ```

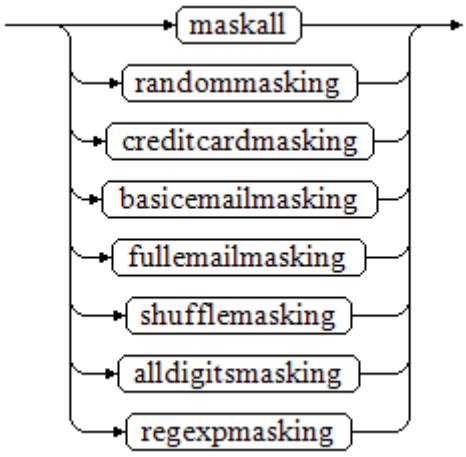
- masking_function：

  maskall不是预置函数，不支持\df展示。

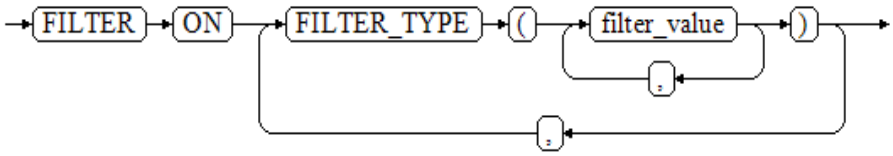
  预置的脱敏方式如下：

  ```
  { maskall | randommasking | creditcardmasking | basicemailmasking | fullemailmasking |
  shufflemasking | alldigitsmasking | regexpmasking }
  ```

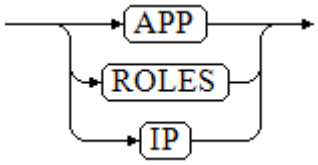

版权所有 © 华为云计算技术有限公司



- policy_filter_clause:
  FILTER ON { FILTER_TYPE (filter_value [, ...]) } [, ...]



- FILTER_TYPE:
  { APP | ROLES | IP }



参数说明

- **policy_name**

  脱敏策略名称，需要符合唯一约束。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  - 脱敏策略名称若超过63个字符，数据库会截断并保留前63个字符当做脱敏策略名称。
  - 当脱敏策略名称中包含大写字母时，数据库会自动转换为小写字母，如果需要创建包含大写字母的脱敏策略名称，则需要使用双引号进行包裹。
  - 脱敏策略名称需要为小写字母（a~z）、大写字母（A~Z）、下划线（_）、数字（0~9）或美元符号（$），且必须以字母或下划线开头。

- **label_name**

  资源标签名称。

- **masking_clause**

  对指定的数据库资源（label_name）进行脱敏操作的脱敏函数标识，支持通过schema.function的方式指定脱敏函数。

- **policy_filter**

  脱敏策略生效的用户标识，若为空表示对所有用户生效。

版权所有 © 华为云计算技术有限公司

- **FILTER_TYPE**
  描述策略过滤的条件类型，包括IP、APP以及ROLES。
- **filter_value**
  具体过滤信息内容，例如具体的IP、具体的APP名称以及具体的用户名。
- **ENABLE | DISABLE**
  是否开启脱敏策略。若不指定ENABLE或DISABLE，默认为ENABLE。

示例

- 创建使用maskall的脱敏策略（将字符串类型的所有值脱敏为x，其他类型的值显示为默认值）。

```
--创建一个表tb_for_masking。
gaussdb=# CREATE TABLE tb_for_masking(idx int, col1 text, col2 text, col3 text, col4 text, col5 text,
col6 text, col7 text,col8 text);

--向表tb_for_masking中插入数据。
gaussdb=# INSERT INTO tb_for_masking VALUES(1, '9876543210', 'usr321usr', 'abc@huawei.com',
'abc@huawei.com', '1234-4567-7890-0123', 'abcdef 123456 ui 323 jsfd321 j3k2l3',
'4880-9898-4545-2525', 'this is a llt case');

--查看数据。
gaussdb=# SELECT * FROM tb_for_masking;
 idx |    col1    |   col2    |      col3      |      col4      |       col5         |              col6
     |        col7         |        col8
-----+------------+-----------+----------------+----------------+--------------------
+-------------------------------
------+---------------------+--------------------
   1 | 9876543210 | usr321usr | abc@huawei.com | abc@huawei.com | 1234-4567-7890-0123 | abcdef
123456 ui 323 jsfd321 j
3k2l3 | 4880-9898-4545-2525 | this is a llt case
(1 row)

--创建资源标签标记敏感列col1。
gaussdb=# CREATE RESOURCE LABEL mask_lb1 ADD COLUMN(tb_for_masking.col1);

--创建一个名为maskpol1的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol1 maskall ON LABEL(mask_lb1);

--访问tb_for_masking表，col1列触发脱敏策略。
gaussdb=# SELECT col1 FROM tb_for_masking;
    col1
------------
 xxxxxxxxxx
(1 row)
```

- 创建使用randommasking的脱敏策略（将字符串类型的值脱敏为随机数字，并且每次访问结果随机）。

```
--创建资源标签标记敏感列col2。
gaussdb=# CREATE RESOURCE LABEL mask_lb2 ADD COLUMN(tb_for_masking.col2);

--创建一个名为maskpol2的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol2 randommasking ON LABEL(mask_lb2);

--访问tb_for_masking表，col2列触发脱敏策略。
gaussdb=# SELECT col2 FROM tb_for_masking;
   col2
-----------
 0e8612d9a
(1 row)
```

- 创建使用basicemailmasking的脱敏策略（将字符串类型邮箱格式值@符号之前的所有数据内容设为x）。

```
--创建资源标签标记敏感列col3。
gaussdb=# CREATE RESOURCE LABEL mask_lb3 ADD COLUMN(tb_for_masking.col3);
```

 版权所有 © 华为云计算技术有限公司

```
--创建一个名为maskpol3的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol3 basicemailmasking ON LABEL(mask_lb3);

--访问tb_for_masking表，col3列触发脱敏策略。
gaussdb=# SELECT col3 FROM tb_for_masking;
     col3
----------------
 xxx@huawei.com
(1 row)
```

- 创建使用fullemailmasking的脱敏策略（将字符串类型邮箱格式值仅保留@符号和邮箱dot结尾，其余全部设为x）。

```
--创建资源标签标记敏感列col4。
gaussdb=# CREATE RESOURCE LABEL mask_lb4 ADD COLUMN(tb_for_masking.col4);

--创建一个名为maskpol4的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol4 fullemailmasking ON LABEL(mask_lb4);

--访问tb_for_masking表，col4列触发脱敏策略。
gaussdb=# SELECT col4 FROM tb_for_masking;
    col4
----------------
 xxx@xxxxxx.com
(1 row)
```

- 创建使用creditcardmasking的脱敏策略（将字符串类型的值保留连接符号-和末尾4位数字，其余全部设为x）。

```
--创建资源标签标记敏感列col5。
gaussdb=# CREATE RESOURCE LABEL mask_lb5 ADD COLUMN(tb_for_masking.col5);

--创建一个名为maskpol5的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol5 creditcardmasking ON LABEL(mask_lb5);

--访问tb_for_masking表，col5列触发脱敏策略。
gaussdb=# SELECT col5 FROM tb_for_masking;
       col5
---------------------
 xxxx-xxxx-xxxx-0123
(1 row)
```

- 创建使用shufflemasking的脱敏策略（将字符串类型的值交换位置）。

```
--创建资源标签标记敏感列col6。
gaussdb=# CREATE RESOURCE LABEL mask_lb6 ADD COLUMN(tb_for_masking.col6);

--创建一个名为maskpol6的脱敏策略。
gaussdb=#  CREATE MASKING POLICY maskpol6 shufflemasking ON LABEL(mask_lb6);

--访问tb_for_masking表，col6列触发脱敏策略。
gaussdb=# SELECT col6 FROM tb_for_masking;
              col6
-------------------------------------
 2 b6jusfd54c3312 13d23lk3jf3  2eai
(1 row)
```

- 创建使用regexpmasking的脱敏策略（将字符串类型的值进行正则表达式脱敏）。

```
--创建资源标签标记敏感列col7。
gaussdb=# CREATE RESOURCE LABEL mask_lb7 ADD COLUMN(tb_for_masking.col7);

--创建一个名为maskpol7的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol7 regexpmasking('[\d+]','*',2, 9) ON LABEL(mask_lb7);

--访问tb_for_masking表，col7列触发脱敏策略。
gaussdb=# SELECT col7 FROM tb_for_masking;
       col7
--------------------
 48**-****-*545-2525
(1 row)
```

 版权所有 © 华为云计算技术有限公司

- 创建仅对用户dev_mask和bob_mask、客户端工具为gsql、IP地址为'172.31.17.160', '127.0.0.0/24'场景下生效的脱敏策略。

```
--创建dev_mask和bob_mask用户。
gaussdb=# CREATE USER dev_mask PASSWORD '*********';
gaussdb=# CREATE USER bob_mask PASSWORD '*********';

--创建资源标签标记敏感列col8。
gaussdb=# CREATE RESOURCE LABEL mask_lb8 ADD COLUMN(tb_for_masking.col8);

--创建一个名为maskpol8的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol8 randommasking ON LABEL(mask_lb8) FILTER ON
ROLES(dev_mask, bob_mask), APP(gsql), IP('172.31.17.160', '127.0.0.0/24');

--访问tb_for_masking表，col8列触发脱敏策略。
gaussdb=# SELECT * FROM tb_for_masking;

--使用gsql工具,IP地址为'172.31.17.160',用户dev_mask查看tb_for_masking。
gaussdb=# GRANT ALL PRIVILEGES TO dev_mask;

--使用maskpol8脱敏，结果随机，每次不同。
gaussdb=# SELECT col8 FROM tb_for_masking;
       col8
--------------------
 cf32a9aa427f219ab0
(1 row)

gaussdb=# SELECT col8 FROM tb_for_masking;
       col8
--------------------
 13efa056dda1e1a474
(1 row)
```

- 删除数据。

```
--删除脱敏策略。
gaussdb=# DROP MASKING POLICY maskpol1, maskpol2, maskpol3, maskpol4, maskpol5, maskpol6,
maskpol7, maskpol8;

--删除资源标签。
gaussdb=# DROP RESOURCE LABEL mask_lb1, mask_lb2, mask_lb3, mask_lb4, mask_lb5, mask_lb6,
mask_lb7, mask_lb8;

--删除表tb_for_masking。
gaussdb=# DROP TABLE tb_for_masking;

--删除用户dev_mask和bob_mask。
gaussdb=# DROP USER dev_mask, bob_mask;
```

相关链接

ALTER MASKING POLICY，DROP MASKING POLICY

1.13.9.29 CREATE MATERIALIZED VIEW

创建一个全量物化视图，可以使用REFRESH MATERIALIZED VIEW（全量刷新）进行物化视图的数据刷新。

CREATE MATERIALIZED VIEW语句与CREATE TABLE AS语句功能相似，但其会持久化存储初始化查询语句，从而支持后续的数据刷新操作。物化视图具有与普通表相同的持久化属性，但不同于临时表，系统不支持创建临时类型的物化视图。

说明

具体的适用场景请参见《开发指南》中“基础概念 > 物化视图”章节的物化视图简介部分。

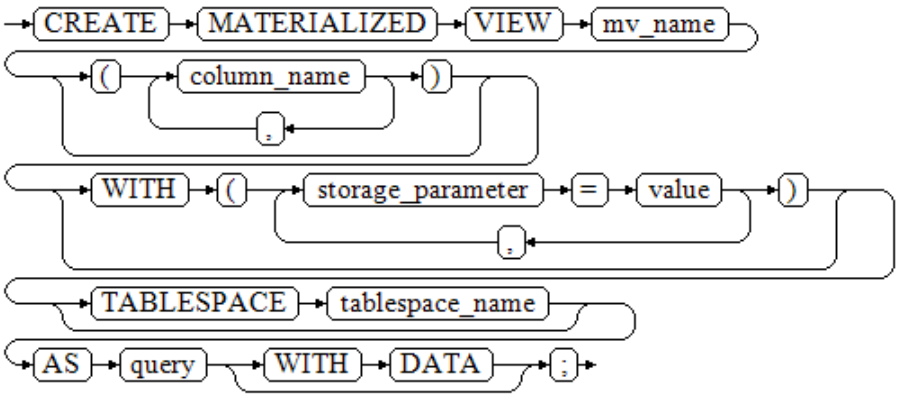
版权所有 © 华为云计算技术有限公司

注意事项

- 不支持在临时表或全局临时表上创建全量物化视图。
- 全量物化视图不支持nodegroup。
- 创建全量物化视图后，基表中的部分DDL操作不再支持。
- 不支持对全量物化视图进行IUD操作。
- 全量物化视图创建后，当基表数据发生变化时，需要使用刷新（REFRESH）命令保持物化视图与基表同步。
- USTORE存储引擎不支持物化视图的创建和使用。
- 段页式不支持物化视图的创建和使用。

语法格式

```
CREATE MATERIALIZED VIEW mv_name
   [ (column_name [, ...] ) ]
   [ WITH ( {storage_parameter = value} [, ... ] ) ]
   [ TABLESPACE tablespace_name ]
   AS query
   [ WITH DATA ];
```



参数说明

- **mv_name**

  创建的物化视图的名称，可以使用模式进行具体指定。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

  新物化视图中的一个列名（字段）。物化视图支持指定列，指定列需要和后面的查询语句结果的列数量保持一致；如果没有提供列名，将从查询的输出列名中获取列名。

  取值范围：字符串，要符合**标识符命名规范**。

- **WITH (storage_parameter [= value] [, ...])**

  为表或索引指定一个可选的存储参数，具体请参见**CREATE TABLE**。

- **TABLESPACE tablespace_name**

  可选，指定新建物化视图所属表空间。如果未指定，将使用默认表空间进行创建。

- **AS query**

版权所有 © 华为云计算技术有限公司

一个SELECT、TABLE或者VALUES命令。这个查询将在一个安全受限的操作中运行。

示例

```
--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_data1 RELATIVE LOCATION 'tablespace1/tbs_data1';

--创建一个普通表。
gaussdb=# CREATE TABLE my_table (c1 int, c2 int);

--创建全量物化视图。
gaussdb=# CREATE MATERIALIZED VIEW my_mv TABLESPACE tbs_data1 AS SELECT * FROM my_table;

--向基表中写入数据。
gaussdb=# INSERT INTO my_table VALUES(1,1),(2,2);

--查询全量物化视图。
gaussdb=# SELECT * FROM my_mv;
 c1 | c2
----+----
(0 rows)

--对全量物化视图my_mv进行全量刷新。
gaussdb=# REFRESH MATERIALIZED VIEW my_mv;

--查询全量物化视图。
gaussdb=# SELECT * FROM my_mv;
 c1 | c2
----+----
  1 |  1
  2 |  2
(2 rows)

--删除全量物化视图。
gaussdb=# DROP MATERIALIZED VIEW my_mv;

--删除普通表my_table。
gaussdb=# DROP TABLE my_table;

--删除表空间。
gaussdb=# DROP TABLESPACE tbs_data1;
```

相关链接

ALTER MATERIALIZED VIEW，**CREATE INCREMENTAL MATERIALIZED VIEW**，**CREATE TABLE**，**DROP MATERIALIZED VIEW**，**REFRESH INCREMENTAL MATERIALIZED VIEW**，**REFRESH MATERIALIZED VIEW**

1.13.9.30 CREATE MODEL

功能描述

用于训练机器学习模型并保存模型。

注意事项

当前形态暂不支持该语法。

版权所有 © 华为云计算技术有限公司

1.13.9.31 CREATE NODE

功能描述

创建一个新的集群节点。

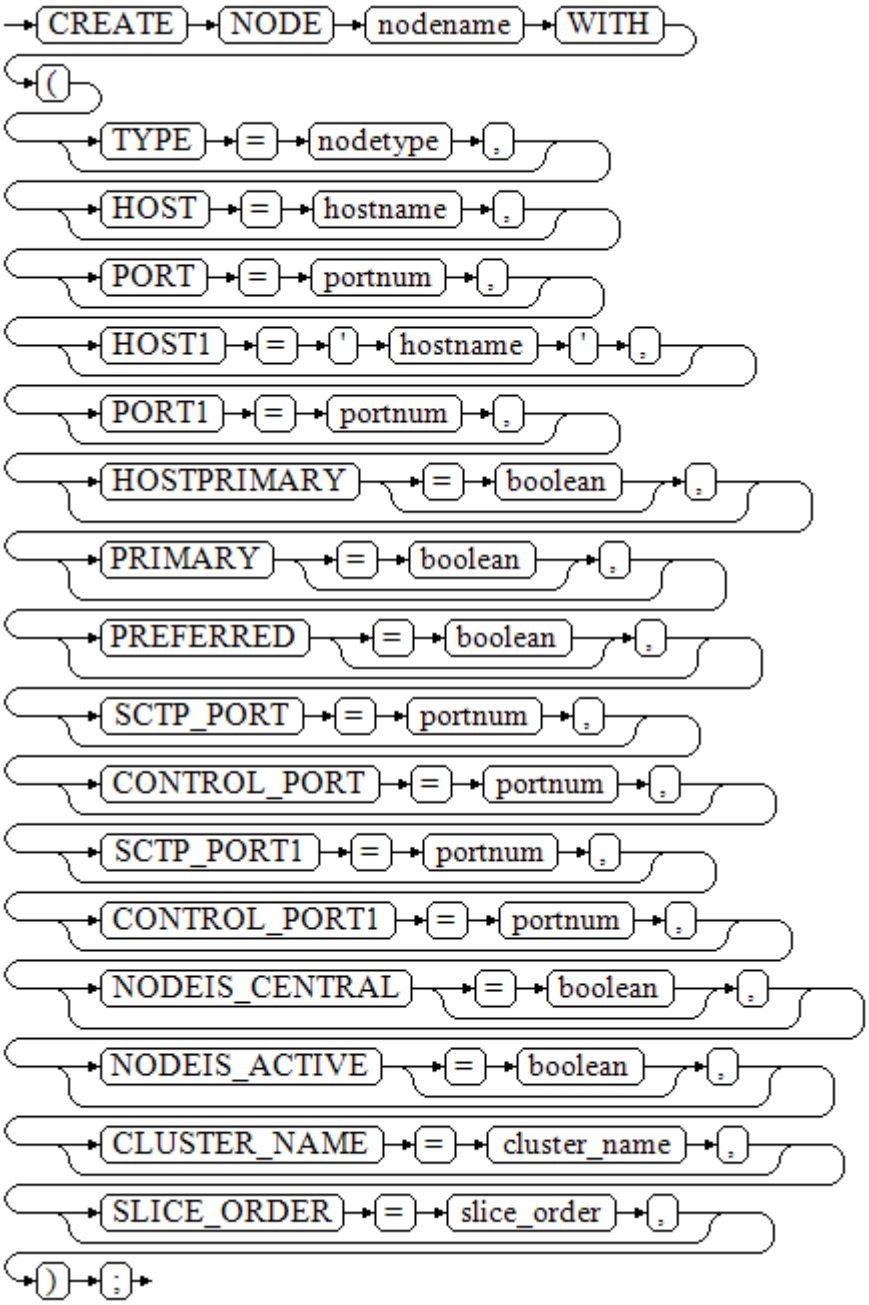
注意事项

- CREATE NODE是集群管理工具封装的接口，用来实现集群管理，管理员用户才有权限使用该接口。
- 该接口禁止用户直接使用，以免对集群状态造成影响，导致数据库不可用。

语法格式

```
CREATE NODE nodename WITH
 (
   [ TYPE = nodetype,]
   [ HOST = hostname,]
   [ PORT = portnum,]
   [ HOST1 = 'hostname',]
   [ PORT1 = portnum,]
   [ HOSTPRIMARY [ = boolean ],]
   [ PRIMARY [ = boolean ],]
   [ PREFERRED [ = boolean ],]
   [ SCTP_PORT = portnum,]
   [ CONTROL_PORT = portnum,]
   [ SCTP_PORT1 = portnum,]
   [ CONTROL_PORT1 = portnum, ]
   [ NODEIS_CENTRAL [= boolean], ]
   [ NODEIS_ACTIVE [= boolean],]
   [ CLUSTER_NAME = cluster_name, ]
   [ SLICE_ORDER = slice_order, ]
 );
```


版权所有 © 华为云计算技术有限公司



说明

PORT选项指定的端口号为节点间内部通信绑定的端口号，不同于外部客户端连接节点的端口号，可通过pgxc_node表查询。

参数说明

- **nodename**

  节点名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **TYPE = nodetype**

  指定节点的类型。

版权所有 © 华为云计算技术有限公司

取值范围：

- 'coordinator'
- 'datanode'
- 'coordinator_standbycluster'
- 'datanode_standbycluster_main'
- 'datanode_standbycluster_other'

- **HOST = hostname**
  指定节点对应的主机名称或者IP地址。
- **PORT = portnum**
  指定节点绑定的主机端口号。
- **HOST1 = hostname**
  指定节点对应的备机名称或者IP地址。
- **PORT1 = portnum**
  指定节点绑定的备机端口号。
- **HOSTPRIMARY**
  声明主机名称或者IP地址是否为主。
  取值范围：
  - true
  - false（默认值）
- **PRIMARY = boolean**
  声明该节点是否为主节点。主节点允许进行读写操作，否则只允许读操作。
  取值范围：
  - true
  - false（默认值）
- **PREFERRED = boolean**
  声明该节点是否为读操作的首选节点。
  取值范围：
  - true
  - false（默认值）
- **SCTP_PORT = portnum**
  主机TCP代理通信库使用的数据传输通道侦听端口，使用TCP协议侦听连接。
- **CONTROL_PORT = portnum**
  主机TCP代理通信库使用的控制传输通道侦听端口，使用TCP协议侦听连接。
- **SCTP_PORT1 = portnum**
  备机TCP代理通信库使用的数据传输通道侦听端口，使用TCP协议侦听连接。
- **CONTROL_PORT 1= portnum**
  备机TCP代理通信库使用的控制传输通道侦听端口，使用TCP协议侦听连接。
- **NODEIS_CENTRAL**
  表明当前节点是否为中心控制节点，只用于CN，对DN无效。

版权所有 © 华为云计算技术有限公司

  - true
  - false（默认值）
- **NODEIS_ACTIVE**

  表明当前节点是否是正常状态，用于标记CN是否被剔除，对DN无效。
  - true
  - false（默认值）
- **CLUSTER_NAME**

  集群名称。
- **SLICE_ORDER**

  节点归属分片的顺序序号。

  说明

  多CN环境下，创建集群节点时只会在当前连接的CN下创建，不会同步至其他CN，需要在所有CN上执行，如果删除集群节点只需要连接一个CN即可。如果只在其中一个CN执行创建了集群节点，要删除需要定义当前cnnodename，并在当前CN下执行删除操作。请参见**DROP NODE**。

示例

```
--创建集群节点，在所有CN上执行。
gaussdb=# CREATE NODE datanode1 WITH(
   TYPE = datanode,
   PREFERRED = false
);
gaussdb=# CREATE NODE datanode2 WITH(
   TYPE = datanode,
   PREFERRED = false
);

--查询集群DN初始状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 datanode1 | f
 datanode2 | f
(2 rows)

--将datanode1设为preferred DN。
gaussdb=# ALTER NODE datanode1 WITH(preferred = true);
--查询集群DN变更后状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 datanode1 | t
 datanode2 | f
(2 rows)

--删除集群节点，连接一个CN即可。
gaussdb=# DROP NODE datanode1;
gaussdb=# DROP NODE datanode2;
```

相关链接

ALTER NODE，**DROP NODE**

版权所有 © 华为云计算技术有限公司

1.13.9.32 CREATE NODE GROUP

功能描述

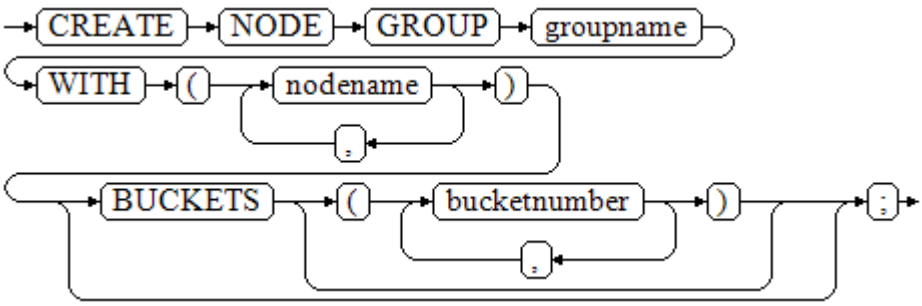
创建一个新的集群节点组。

注意事项

- CREATE NODE GROUP是集群管理工具封装的接口，用来实现集群管理。
- 该接口仅对管理员用户开放使用。

语法格式

```
CREATE NODE GROUP groupname
   WITH ( nodename [, ... ] )
   [ BUCKETS [ ( bucketnumber [, ... ] ) ] ];
```



参数说明

- **groupname**
  节点组名称。
  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  节点组命名虽然支持ASCII字符集上所有字符，但是仍建议用户按照**标识符命名规范**进行命名。

- **nodename**
  节点名称。
  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。
- **BUCKETS [(bucketnumber [, ...])]**
  集群管理工具的内部用法，仅支持在扩容期间调用。

示例

```
--查询集群DN初始状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 dn_6001_6002_6003 | f
 dn_6004_6005_6006 | f
 dn_6007_6008_6009 | f
(3 rows)

--创建节点组，用上一步中查询到的真实节点名称替换dn_6001_6002_6003。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE NODE GROUP test_group WITH ( dn_6001_6002_6003 );

--查询节点组。
gaussdb=# SELECT group_name, group_members FROM pgxc_group;
   group_name   |   group_members
----------------+-------------------
 group_version1 | 16384 16388 16394
 test_group     | 16384
(2 rows)

--删除节点组。
gaussdb=# DROP NODE GROUP test_group;
```

相关链接

ALTER NODE GROUP，**DROP NODE GROUP**

1.13.9.33 CREATE PACKAGE

功能描述

创建一个新的PACKAGE。

注意事项

- 仅支持在ORA兼容模式数据库中使用PACKAGE。
- 在包头（PACKAGE SPECIFICATION）中声明过的函数或者存储过程，必须在包体（PACKAGE BODY）明确定义。
- 创建存储过程时，仅对CREATE的存储过程或PACKAGE本身加写锁；对于执行过程中的编译以及执行会对函数和函数依赖的PACKAGE加读锁。
- 在实例化中，无法调用含有COMMIT或ROLLBACK的存储过程。
- 不能在触发器中调用PACKAGE函数。
- 不能在外部SQL中直接使用PACKAGE中的变量。
- 不允许在PACKAGE外部调用PACKAGE的私有变量和存储过程。
- 存储过程中不支持的用法，PACKAGE中也不支持使用。例如：在函数（function）中不允许调用COMMIT/ROLLBACK，则PACKAGE的函数（function）中同样无法调用COMMIT/ROLLBACK。
- 不支持Schema与PACKAGE同名。
- 只支持兼容Oracle数据库的存储过程和函数定义。
- 不支持PACKAGE内有同名变量，包括包内同名参数。
- PACKAGE的全局变量为session级，不同session之间PACKAGE的变量不共享。
- PACKAGE中调用自治事务的函数，不允许使用PACKAGE中的游标变量，以及不允许递归的使用PACKAGE中游标变量的函数。
- PACKAGE中不支持声明REF CURSOR变量。
- PACKAGE默认为SECURITY INVOKER权限，如果想将默认行为改为SECURITY DEFINER权限，需要设置GUC参数behavior_compat_options='plsql_security_definer'。
- 被授予CREATE ANY PACKAGE权限的用户，可以在PUBLIC模式和用户模式下创建PACKAGE。

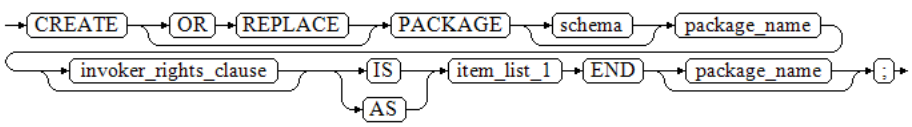
版权所有 © 华为云计算技术有限公司

- 如果需要创建带有特殊字符的PACKAGE名，特殊字符中不能含有空格，且需设置GUC参数behavior_compat_options="skip_insert_gs_source"，否则可能引起报错。
- 创建PACKAGE函数时，其参数默认值支持含有PACKAGE内变量，不支持跨PACKAGE变量。
- PACKAGE中函数复杂调用，如：func(x).a，函数调用返回复合类型，支持跨Schema调用，不支持通过database.schema.package.func(x).b的方式调用。
- 在创建PACKAGE内的存储过程时，如果存储过程名称为schema.func或package.func形式，只会获取func的名称，不会对Schema声明以及PACKAGE声明进行获取，如果需要禁止这种行为，可以设置GUC参数behavior_compat_options='forbid_package_function_with_prefix'。
- 不支持view%RowType作为函数参数。
- 不支持存储过程/函数中的INSERT VALUES使用复合类型record。
- 当前形态不支持SAVE EXCEPTIONS（调用的存过中使用）。
- 表达式不支持COMMIT/ROLLBACK。
- 对于无参数的函数，不支持省略括号。
- 自定义类型变量不支持下推。
- 在创建PACKAGE内的存储过程时，如果存储过程的入参包含伪类型（如：anyelement），不支持调用时传入变长类型的参数。
- PACKAGE内创建的游标或者函数带参数游标，可以跨PACKAGE进行类型引用（%RowType）和打开使用，存在以下限制：
  - 暂不支持通过database.schema.package.cursor的方式调用游标。
  - 不支持将游标定义为函数参数。
  - 当引用游标时，不支持cursor%RowType。
- 在PACKAGE SPECIFICATION或PACKAGE BODY SPECIFICATION中声明自治事务PRAGMA AUTONOMOUS_TRANSACTION语法，可成功创建PACKAGE，但自治事务不生效。
- 非初始用户无权对初始用户的PACKAGE进行REPLACE操作。
- 创建PACKAGE时，仅对包头（PACKAGE SPECIFICATION）、包体（PACKAGE BODY）内变量默认值进行有效性检查（检查默认值依赖函数是否存在、是否有效），不执行相关语句，包头、包体变量初始化及包体实例化部分（PACKAGE BODY ANONYMOUS BLOCK）仅在第一次调用时执行一次。
- 在PACKAGE创建、重编译及相关嵌套编译过程中，PACKAGE实例化（包头、包体变量初始化及包体实例化部分）均不会被执行。

语法格式

- CREATE PACKAGE SPECIFICATION（创建包头）语法格式。

```
CREATE [ OR REPLACE ] PACKAGE [ schema ] package_name
  [ invoker_rights_clause ] { IS | AS } item_list_1 END [ package_name ];
```



  - invoker_rights_clause：可以使用AUTHID DEFINER或AUTHID CURRENT_USER进行声明包头的调用者权限，其中，AUTHID DEFINER表示创建者权限，AUTHID CURRENT_USER表示调用者权限。包头内的变量、函

版权所有 © 华为云计算技术有限公司

数或存储过程按照声明时指定的权限（AUTHID DEFINER或AUTHID CURRENT_USER）执行，而包头内的游标则以调用者权限（AUTHID CURRENT_USER）执行。

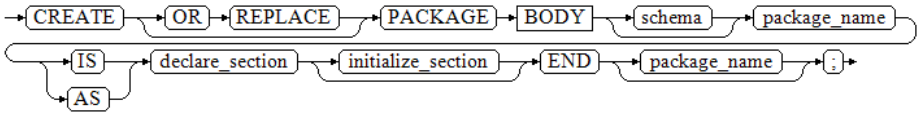
- item_list_1：可以是声明的变量或者存储过程以及函数。
- END后的package_name可省略（gs_dump生成的SQL文件，PACKAGE定义会自动补充END后名称），若未指定该package_name，则CREATE PACKAGE SPECIFICATION（创建包头）语法格式的结尾必须添加结束符（/）。

说明

PACKAGE SPECIFICATION声明了包内的公有变量、函数以及异常等，可以被外部函数或者存储过程调用。在PACKAGE SPECIFICATION中只能声明存储过程以及函数，不能定义存储过程或者函数。

- CREATE PACKAGE BODY（创建包体）语法格式。

```
CREATE [ OR REPLACE ] PACKAGE BODY [ schema ] package_name
  { IS | AS } declare_section [ initialize_section ] END [ package_name ];
```



END后的package_name可省略（gs_dump生成的SQL文件，PACKAGE定义会自动补充END后名称），若未指定该package_name，则CREATE PACKAGE BODY（创建包体）语法格式的结尾必须添加结束符（/）。

说明

- PACKAGE BODY定义了包的私有变量以及函数等。如果变量或者函数没有在PACKAGE SPECIFICATION中声明过，那么这个变量或者函数则为私有变量或者私有函数。
- PACKAGE BODY也可以声明实例化部分，用于初始化PACKAGE。

示例

- CREATE PACKAGE示例

```
--创建数据库。
gaussdb=# CREATE DATABASE test DBCOMPATIBILITY 'ORA';
gaussdb=# \c test

--创建测试表tbl_test。
test=# CREATE TABLE tbl_test(uid varchar(6) PRIMARY KEY, area_id varchar(5), level int);

--创建包头。
test=# CREATE OR REPLACE PACKAGE pkg_test AS
   --公有存储过程可以被外部调用。
   PROCEDURE proc_ist_tbl_test(v_uid in varchar, v_area_id varchar, v_level int);
   PROCEDURE proc_del_tbl_test(v_uid in varchar);
   PROCEDURE proc_upd_tbl_test(v_uid in varchar, v_area_id varchar, v_level int);
END pkg_test;
/

--创建包体。
test=# CREATE OR REPLACE PACKAGE BODY pkg_test AS
   PROCEDURE proc_ist_tbl_test(v_uid in varchar, v_area_id varchar, v_level int) AS
   BEGIN
      INSERT INTO tbl_test VALUES (v_uid, v_area_id, v_level);
   END;
   PROCEDURE proc_del_tbl_test(v_uid in varchar) AS
   BEGIN
      DELETE FROM tbl_test WHERE uid = v_uid;
   END;
```


版权所有 © 华为云计算技术有限公司

```
    PROCEDURE proc_upd_tbl_test(v_uid in varchar, v_area_id varchar, v_level int) AS
    BEGIN
        UPDATE tbl_test SET area_id = v_area_id, level = v_level WHERE uid = v_uid;
    END;
    var4 int := 10;
    --实例化开始。
BEGIN
    var4 := 10;
    dbe_output.print_line(var4);
END pkg_test;
/
```

- 调用PACKAGE示例

```
--使用CALL调用PACKAGE的存储过程。
test=# CALL pkg_test.proc_ist_tbl_test('0A00B1','01001',24);
test=# SELECT * FROM tbl_test;
  uid   | area_id | level
--------+---------+---------
 0A00B1 | 01001   |      24
(1 row)

--使用SELECT调用PACKAGE的存储过程。
test=# SELECT pkg_test.proc_upd_tbl_test('0A00B1','01001','26');
test=# SELECT * FROM tbl_test;
  uid   | area_id | level
--------+---------+-------
 0A00B1 | 01001   |    26
(1 row)

--匿名块调用PACKAGE的存储过程。
test=# BEGIN
pkg_test.proc_del_tbl_test('0A00B1');
END;
/
test=# SELECT * FROM tbl_test;
 uid | area_id | level
-----+---------+-------
(0 rows)

--删除。
test=# DROP TABLE tbl_test;
test=# DROP PACKAGE pkg_test;
--切换至默认数据库并删除test数据库（请根据实际情况修改数据库名）。
test=# \c postgres
gaussdb=# DROP DATABASE test;
```

相关链接

ALTER PACKAGE，**DROP PACKAGE**

1.13.9.34 CREATE PROCEDURE

功能描述

创建一个新的存储过程。

注意事项

- 创建存储过程时，如果参数或返回值带有精度，创建操作中不进行精度检测。
- 创建存储过程时，存储过程定义中对表对象的操作建议明确指定模式，否则可能会导致存储过程执行异常。
- 创建存储过程时，仅对CREATE的存储过程或PACKAGE本身加写锁；对于执行过程中编译以及执行会对函数和函数依赖的PACKAGE均加读锁。

版权所有 © 华为云计算技术有限公司

- 创建存储过程时，存储过程内部通过SET语句设置current_schema和search_path无效。执行函数后的search_path和current_schema与执行函数前的search_path和current_schema保持一致。
- 通过SELECT以及CALL调用函数时，必须在出参位置提供实参进行调用，实参不会发生作用。
- 存储过程指定PACKAGE属性时支持重载。
- 不能创建仅形参名字不同（存储过程名和参数列表类型都一样）的重载存储过程。
- 不能创建与函数拥有相同名称和参数列表的存储过程。
- 不支持仅默认值不同的存储过程重载。
- 存储过程仅IN、OUT、INOUT这三种类型不同的参数，当打开GUC参数behavior_compat_options（即SET behavior_compat_options='proc_outparam_override';）后，不允许重载。关闭GUC参数后，支持重载。
- ORA兼容模式的数据库，建立兼容Oracle数据库风格的函数；PG兼容模式的数据库，建立兼容PostgreSQL数据库风格的函数。不建议混合创建。
- 函数如果支持重载，需要指定PACKAGE关键字。
- 重载的存储过程在调用时变量需要明确具体的类型。
- 在存储过程内部使用未声明的变量，存储过程被调用时会报错。
- 在创建存储过程时，不能在avg函数外面嵌套其他agg函数，或者其他系统函数。
- 函数定义时，如果指定为IMMUTABLE和SHIPPABLE类型，应尽量避免函数中存在INSERT、UPDATE、DELETE、MERGE和DDL操作，由于以上操作需要由CN判断对应的执行节点，可能存在执行结果错误的问题。
- 存储过程中不支持需要进行RETURN集合的操作。
- 在存储过程内部调用其他无参数的存储过程时，可以省略括号，直接使用存储过程名进行调用。
- 在存储过程内部调用其他有出参的函数，如果在赋值表达式中调用时，需要打开GUC参数behavior_compat_options（即SET behavior_compat_options = 'proc_outparam_override';），并定义与出参类型相同的变量，并将变量作为出参调用带有出参的其他函数，出参才能生效。否则，被调函数的出参会被忽略。
- 在表达式中使用OUT参数作为出参时，如下情况OUT出参不会生效，例如：使用EXECUTE IMMEDIATE sqlv USING func语法执行函数、使用SELECT func INTO语法执行函数、使用INSERT以及UPDATE等DML语句执行函数、使用SELECT WHERE a=func()的方式执行、以及带OUT出参的函数作为入参时（如：fun(func(OUT b), a)）且OUT出参b未生效、使用带OUT出参的函数为变量默认值赋值等。
- 存储过程支持参数注释的查看与导出/导入。
- 存储过程支持介于IS/AS与plsql_body之间的注释的查看与导出/导入。
- 被授予CREATE ANY FUNCTION权限的用户，可以在用户模式下创建/替换存储过程。
- OUT/INOUT参数必须传入变量，不能传入常量。
- 存储过程默认为SECURITY INVOKER权限，如果想将默认行为改为SECURITY DEFINER权限，需要设置GUC参数behavior_compat_options='plsql_security_definer'。

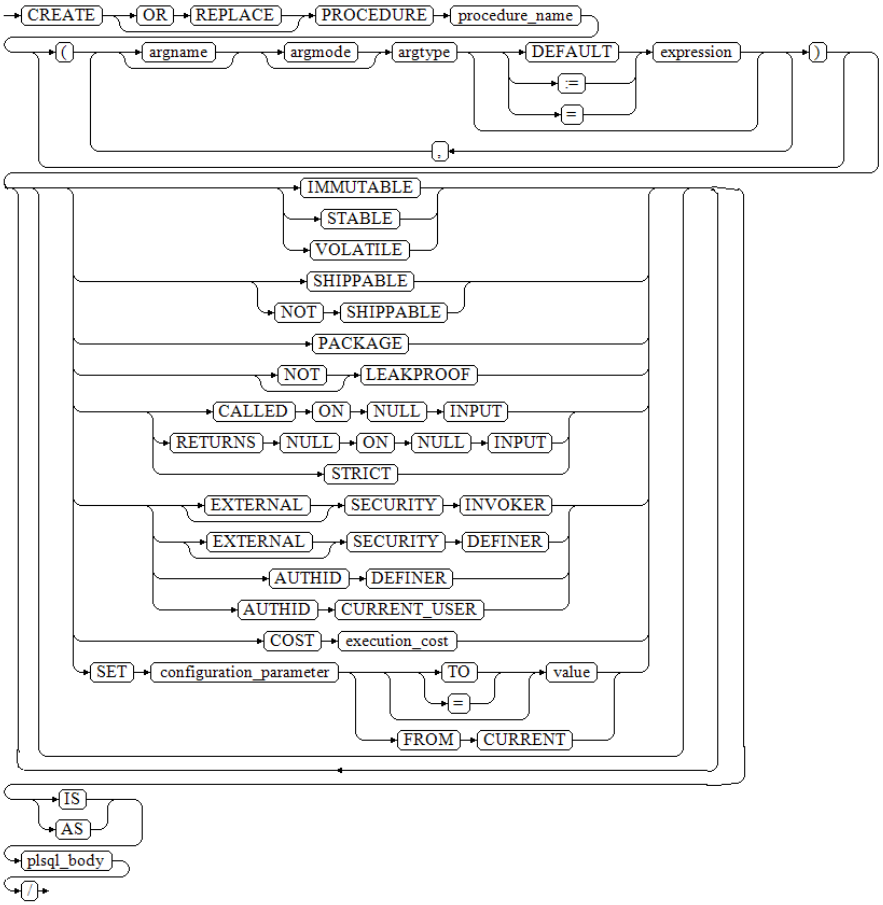
版权所有 © 华为云计算技术有限公司

- 如果将定义者权限的存储过程创建到其他用户Schema下，则会以其他用户的权限执行该存储过程，存在越权风险，请谨慎使用。
- 在存储过程内部函数复杂调用（如：func(x).a），函数调用返回复合类型，支持跨Schema调用，不支持通过database.schema.package.func(x).b的方式调用。
- 调用带OUT出参的存储过程，设置GUC参数behavior_compat_options = 'proc_outparam_transfer_length'后可以传递参数长度。规格限制如下：
  - 支持的基本类型包括：CHAR(n)、CHARACTER(n)、NCHAR(n)、VARCHAR(n)、VARYING(n)、VARCHAR2(n)以及NVARCHAR2(n)。
  - OUT出参不生效的情况下（比如PERFORM）不需要传递长度。
  - 不支持精度传递的基本类型包括：NUMERIC、DECIMAL、NUMBER、FLOAT、DEC、INTEGER、TIME、TIMESTAMP、INTERVAL、TIME WITH TIME ZONE、TIMESTAMP WITH TIME ZONE、TIME WITHOUT TIME ZONE以及TIMESTAMP WITHOUT TIME ZONE。
  - 无论GUC参数behavior_compat_options是否设置为proc_outparam_override，均支持传递参数长度。
  - 要传递集合类型的元素长度和被集合类型嵌套的数组类型的元素长度，需要在GUC参数behavior_compat_options里同时开启tableof_elem_constraints选项。
- 函数中存在通过GUC参数控制特性的语法以及函数等，如果在会话内更改相关GUC参数，修改参数后，调用函数可能会维持修改前的行为，请谨慎进行变更GUC参数配置的操作。
- 非初始用户无权对初始用户的存储过程进行REPLACE操作。
- 当在存储过程体中进行创建用户、修改密码或加解密等涉及密码或密钥相关操作时，系统表及日志中会记录密码或密钥的明文信息。为防止敏感信息泄露，不建议用户在存储过程体中进行涉及密码或密钥等敏感信息的相关操作。

语法格式

```
CREATE [ OR REPLACE ] PROCEDURE procedure_name
    [ ( {[ argname ] [ argmode ] argtype [ { DEFAULT | := | = } expression ]}[,...]) ]
    [
      { IMMUTABLE | STABLE | VOLATILE }
      | { SHIPPABLE | NOT SHIPPABLE }
      | {PACKAGE}
      | [ NOT ] LEAKPROOF
      | { CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }
      | {[ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER | AUTHID DEFINER | AUTHID
CURRENT_USER}
      | COST execution_cost
      | SET configuration_parameter { [ TO | = ] value | FROM CURRENT }
    ][ ... ]
 { IS | AS }
plsql_body
/
```


版权所有 © 华为云计算技术有限公司



参数说明

- **OR REPLACE**

  当存在同名的存储过程时，替换原来的定义。

- **procedure_name**

  创建的存储过程名称，可以使用模式进行具体指定。

  取值范围：字符串，要符合**标识符命名规范**。

  说明

  建议不要创建和系统函数重名的函数，否则调用时需要指定函数的Schema。

- **argname**

  参数的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **argmode**

  参数的模式。

  取值范围：IN、OUT、INOUT或VARIADIC，默认值是IN。只有OUT模式的参数后面可以添加VARIADIC，并且OUT和INOUT模式的参数不能用在RETURNS TABLE的过程定义中。

版权所有 © 华为云计算技术有限公司

说明

VARIADIC用于声明数组类型的参数。

- **argtype**

  参数的数据类型。可以使用%RowType间接引用表的类型，或者使用%Type间接引用表或复合类型中某一列的类型。

  取值范围：可用的数据类型。

  说明

  argname和argmode的顺序没有严格要求，推荐按照argname、argmode、argtype的顺序使用。

- **expression**

  参数的默认表达式。

  说明

  - 在参数a_format_version值为10c和a_format_dev_version值为s2的情况下，参数为INOUT模式时不支持默认表达式。
  - 推荐使用方式：将所有默认值参数定义在所有非默认值参数后。
  - 调用带有默认参数的函数时，入参从左向右排入函数，如果存在非默认参数的入参缺失则会报错。
  - 打开proc_uncheck_default_param参数，调用带有默认参数的函数时，入参从左向右排入函数，允许缺省默认参数的入参，如果存在非默认参数的入参缺失，则会用错位的默认值填充该参数。
  - 在参数a_format_version值为10c、a_format_dev_version值为s1和关闭proc_outparam_override，函数参数同时包括OUT出参和DEFAULT时，默认值不可省略。

- **IMMUTABLE | STABLE**

  行为约束选项。

  参数的功能与CREATE FUNCTION类似，具体说明请参见**CREATE FUNCTION**

- **plsql_body**

  PL/SQL存储过程体。

  创建一个存储过程时，plsql_body支持以“END;”或“END procedure_name;”两种形式结尾。

  说明

  以“END procedure_name;”结尾时遵循以下约束：

  - 仅ORA兼容模式（数据库级别）下支持使用。
  - 仅支持创建兼容Oracle风格数据库的语法。
  - 仅支持END后设置存储过程名称的场景。
  - 创建的存储过程在DBE_PLDEVELOPER.gs_source、my_source、db_source、adm_source视图中显示与存储过程创建时定义相同（创建时“END+名称”则显示“END+名称”）；在pg_proc视图以及\sf查看时，最外层END后在任何情况下都不跟名称，内部则按照创建时定义（创建时“END+名称”则显示“END+名称”）显示。
  - gs_dump生成的SQL文件，存储过程无法保留最外层END后名称。
  - 以IF/LOOP命名存储过程或嵌套子程序时，不支持以END IF/LOOP结束存储过程。

版权所有 © 华为云计算技术有限公司

示例

- 创建一个存储过程

```
--创建存储过程，返回传入参数的和。
gaussdb=# CREATE OR REPLACE PROCEDURE proc_add(i int, j int)
AS
BEGIN
    dbe_output.print_line('result is：'|| i+j);
END;
/

--使用CALL命令调用存储过程。
gaussdb=# CALL proc_add(16,17);

--使用程序块调用存储过程。
gaussdb=# BEGIN
   proc_add(16,17);
END;
/

--删除。
gaussdb=# DROP PROCEDURE proc_add;
```

- ORA兼容模式数据库下以END结尾创建一个存储过程

```
--创建存储过程，打印传入参数的和。
gaussdb=# CREATE OR REPLACE PROCEDURE proc_add(i int, j int)
AS
BEGIN
    dbe_output.print_line('result is：'|| i+j);
END proc_add;
/

--使用CALL命令调用存储过程。
gaussdb=# CALL proc_add(16,17);

--使用程序块调用存储过程。
gaussdb=# BEGIN
   proc_add(16,17);
END;
/

--删除。
gaussdb=# DROP PROCEDURE proc_add;
```

- 创建一个参数模型为VARIADIC的存储过程

```
--创建存储过程pro_variadic。
gaussdb=# CREATE OR REPLACE PROCEDURE pro_variadic (var1 VARCHAR2(10) DEFAULT 'hello!',var4
VARIADIC int4[])
AS
BEGIN
   dbe_output.print_line(var1);
   for i in 1..array_length(var4,1) loop
       dbe_output.print_line(var4[i]);
   end loop;
END;
/

--执行此存储过程。
gaussdb=# SELECT pro_variadic(var1=>'hello', VARIADIC var4=> array[3,5,11,2]);

--删除。
gaussdb=# DROP PROCEDURE pro_variadic;
```

- 参数模型IN与OUT
  - IN表示该参数是传入的参数（缺省项）。
  - OUT表示该参数是传出的参数。
  - IN OUT表示该参数既要传入也要传出。

 版权所有 © 华为云计算技术有限公司

```
--创建存储过程proc_add1，num1和num2为传入的参数，num3为传出的参数。
gaussdb=# CREATE PROCEDURE proc_add1 (num1 in int, num2 in int, num3 out int)
AS
BEGIN
    num3 := num1 + num2;
END;
/

--使用程序块调用该存储过程，使用变量c来接收该存储过程传出的参数。
gaussdb=# DECLARE
    a int := 20;
    b int := 32;
    c int := 0;
BEGIN
    proc_add1(a,b,c);
    dbe_output.put_line(c);
END;
/

--删除。
gaussdb=# DROP PROCEDURE proc_add1;
```

相关链接

ALTER PROCEDURE，**DROP PROCEDURE**

1.13.9.35 CREATE RESOURCE LABEL

功能描述

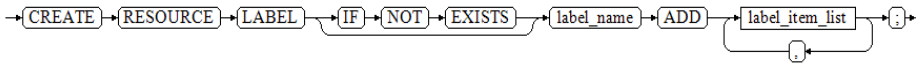
创建资源标签。

注意事项

仅POLADMIN、SYSADMIN或初始用户可以执行此操作。

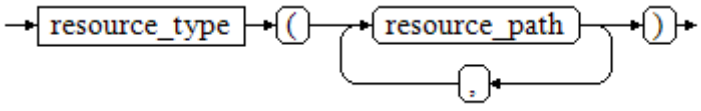
语法格式

```
CREATE RESOURCE LABEL [IF NOT EXISTS] label_name ADD label_item_list[, ...];
```



- label_item_list：

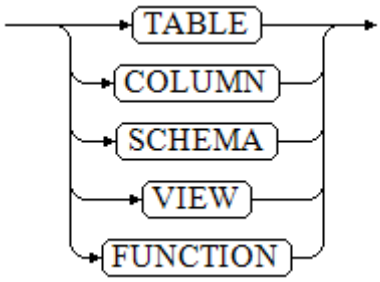
  ```
  resource_type(resource_path[, ...])
  ```

- resource_type：

  ```
  { TABLE | COLUMN | SCHEMA | VIEW | FUNCTION }
  ```


版权所有 © 华为云计算技术有限公司



参数说明

- **IF NOT EXISTS**

  判断是否存在同名的资源标签。如果已经存在相同名称的资源标签，不会产生错误，而是产生一条提示，用于告知此资源标签已存在。

- **label_name**

  资源标签名称，需要满足唯一约束。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  - 资源标签名称若超过63个字符，数据库会截断并保留前63个字符当作资源标签名称。
  - 当资源标签名称中包含大写字母时数据库会自动转换为小写字母，如果需要创建包含大写字母的资源标签名称则需要使用双引号进行包裹。
  - 资源标签名称需要为小写字母（a~z）、大写字母（A~Z）、下划线（_）、数字（0~9）或美元符号（$），且必须以字母或下划线开头。

- **resource_type**

  需要标记的数据库资源的类型。

  取值范围：表（TABLE）、列（COLUMN）、模式（Schema）、视图（VIEW）以及函数（FUNCTION）。

- **resource_path**

  描述具体的数据库资源的路径。

示例

```
--创建一个表tb_for_label。
gaussdb=# CREATE TABLE tb_for_label(col1 text, col2 text, col3 text);

--基于表创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS table_label add TABLE(public.tb_for_label);

--再次创建已存在的表资源标签，对比加参数IF NOT EXISTS与不加IF NOT EXISTS参数的区别。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS table_label add TABLE(public.tb_for_label);
NOTICE:  table_label label already defined, skipping
CREATE RESOURCE LABEL
gaussdb=# CREATE RESOURCE LABEL table_label add TABLE(public.tb_for_label);
ERROR:  table_label label already defined

--基于列创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS column_label add COLUMN(public.tb_for_label.col1);

--创建一个模式schema_for_label。
gaussdb=# CREATE SCHEMA schema_for_label;

--基于模式创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS schema_label add SCHEMA(schema_for_label);
```


版权所有 © 华为云计算技术有限公司

```
--创建一个视图view_for_label。
gaussdb=# CREATE VIEW view_for_label AS SELECT 1;

--基于视图创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS view_label add VIEW(view_for_label);

--创建一个函数func_for_label。
gaussdb=# CREATE FUNCTION func_for_label RETURNS TEXT AS $$ SELECT col1 FROM tb_for_label; $$
LANGUAGE SQL;

--基于函数创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS func_label add FUNCTION(func_for_label);

--删除表资源标签table_label。
gaussdb=# DROP RESOURCE LABEL IF EXISTS table_label;

--删除列资源资源标签column_label。
gaussdb=# DROP RESOURCE LABEL IF EXISTS column_label;

--删除函数资源标签func_for_label。
gaussdb=# DROP FUNCTION func_for_label;

--删除视图资源标签view_for_label。
gaussdb=# DROP VIEW view_for_label;

--删除模式资源标签schema_for_label。
gaussdb=# DROP SCHEMA schema_for_label;

--删除表tb_for_label。
gaussdb=# DROP TABLE tb_for_label;
```

相关链接

ALTER RESOURCE LABEL，**DROP RESOURCE LABEL**

1.13.9.36 CREATE RESOURCE POOL

功能描述

用于创建一个资源池，并指定此资源池相关联的控制组。

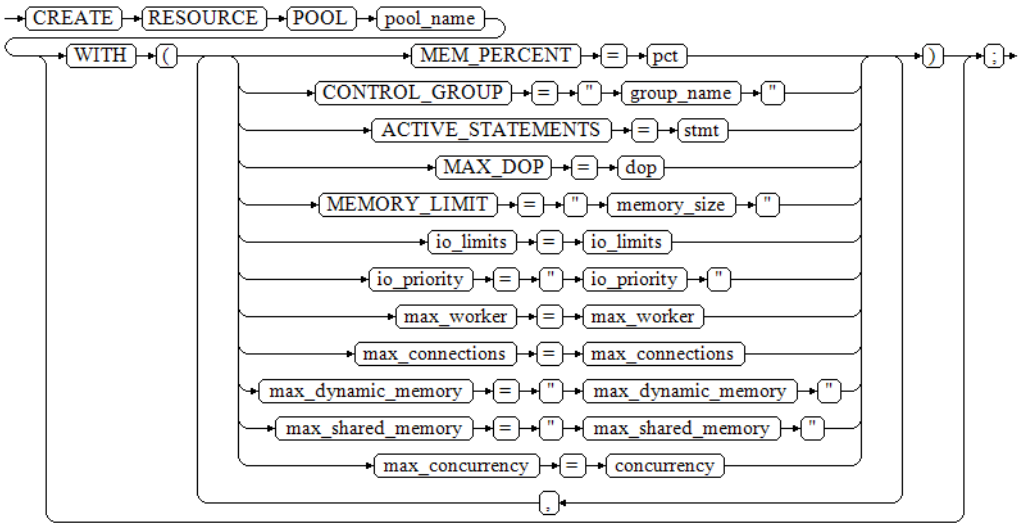
注意事项

- 仅SYSADMIN或初始用户可以执行此操作。
- 升级期间不支持使用该语法。

语法格式

```
CREATE RESOURCE POOL pool_name
    [WITH ({MEM_PERCENT=pct | CONTROL_GROUP="group_name" | ACTIVE_STATEMENTS=stmt |
MAX_DOP = dop | MEMORY_LIMIT="memory_size" | io_limits=io_limits | io_priority="io_priority" |
max_worker=max_worker | max_connections=max_connections |
max_dynamic_memory="max_dynamic_memory" | max_shared_memory="max_shared_memory" |
max_concurrency=concurrency}[, ... ])];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **pool_name**

  资源池名称。不能和当前数据库中已存在的资源池重名，需满足唯一约束。

  取值范围：字符串，要符合**标识符命名规范**。

- **group_name**

  控制组名称。

  取值范围：字符串，要符合如下说明中的规则：

  说明

  - 设置控制组名称时，语法可以使用双引号，也可以使用单引号。
  - 控制组名称对大小写敏感。
  - 不指定group_name时，默认指定的字符串为 "Medium"，代表指定DefaultClass控制组的"Medium" Timeshare控制组。
  - 若数据库管理员指定自定义Class组下的Workload控制组，如control_group的字符串为："class1:workload1"；代表此资源池指定到class1控制组下的workload1控制组。也可同时指定Workload控制组的层次，如control_group的字符串为："class1:workload1:1"。
  - 若数据库用户指定Timeshare控制组代表的字符串，即"Rush"、"High"、"Medium"或"Low"其中一种，如control_group的字符串为"High"；代表资源池指定到DefaultClass控制组下的"High" Timeshare控制组。
  - 多租户场景下，组资源池关联的控制组为Class级别，业务资源池关联Workload控制组。且不允许在各种资源池间相互切换。

- **stmt**

  资源池语句执行的最大并发数量。

  取值范围：整数值型，-1~2147483647。-1表示不限制，0表示禁止任何语句执行。

- **dop**

  资源池最大并发度，语句执行时能够创建的最多线程数量。

  取值范围：正整数类型，1~2147483647。

- **memory_size**

  资源池最大使用内存。

版权所有 © 华为云计算技术有限公司

取值范围：字符串，内容范围1KB~2047GB。

- **mem_percent**

  资源池可用内存占全部内存或者组用户内存使用的比例。

  - 在多租户场景下，组用户和业务用户的mem_percent范围1~100，默认为20。
  - 在普通场景下，普通用户的mem_percent范围为0~100，默认值为0。

  说明

  mem_percent和memory_limit同时指定时，仅mem_percent生效。

- **io_limits**

  资源池每秒可触发I/O次数上限。以万次为单位计数。

- **io_priority**

  I/O利用率高达50%时，重消耗I/O作业进行I/O资源管控时关联的优先级等级。分为Low、Medium和High。

  不控制时可设置为None。默认为None。

  说明

  io_limits和io_priority的设置都仅对复杂作业有效。包括批量导入（INSERT INTO SELECT、COPY FROM、CREATE TABLE AS等）、单DN数据量大约超过500MB的复杂查询和VACUUM FULL等操作。

- **max_worker**

  只用于扩容的接口，表示扩容数据重分布时，表内插入并发度。

- **max_connections**

  最大连接数，用于限制资源池可使用的最大连接数。

  说明

  所有资源池的最大连接数总和不能超过整个数据库进程设置的GUC参数max_connections指定的最大连接数。

- **max_dynamic_memory**

  最大动态内存值，用于限制资源池允许使用的最大动态内存值。

- **max_shared_memory**

  最大共享内存值，用于限制资源池允许使用的最大共享内存值。

- **max_concurrency**

  最大并发数，用于限制资源池允许使用的最大并发数。

示例

本示例假定用户已预先成功创建控制组。若需创建控制组，请联系管理员处理。

```
-- 创建控制组（命令行执行）。
gs_ssh -c "gs_cgroup -c -S class1 -s 40"
gs_ssh -c "gs_cgroup -c -S class1 -G wg1 -g 20"
gs_ssh -c "gs_cgroup -c -S class1 -G wg2 -g 20"

-- 使用gsql以管理员身份登录数据库。
-- 创建一个默认资源池，其控制组为"DefaultClass"组下属的"Medium" Timeshare Workload控制组。
gaussdb=# CREATE RESOURCE POOL pool1;

-- 创建一个资源池，其控制组指定为"DefaultClass"组下属的"High" Timeshare Workload控制组。
```

 版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE RESOURCE POOL pool2 WITH (CONTROL_GROUP="High");

-- 创建一个资源池，其控制组指定为"class1"组下属的"Low" Timeshare Workload控制组。
gaussdb=# CREATE RESOURCE POOL pool3 WITH (CONTROL_GROUP="class1:Low");

-- 创建一个资源池，其控制组指定为"class1"组下属的"wg1" Workload控制组。
gaussdb=# CREATE RESOURCE POOL pool4 WITH (CONTROL_GROUP="class1:wg1");

-- 创建一个资源池，其控制组指定为"class1"组下属的"wg2" Workload控制组。
gaussdb=# CREATE RESOURCE POOL pool5 WITH (CONTROL_GROUP="class1:wg2:3");

-- 删除资源池。
gaussdb=# DROP RESOURCE POOL pool1;
gaussdb=# DROP RESOURCE POOL pool2;
gaussdb=# DROP RESOURCE POOL pool3;
gaussdb=# DROP RESOURCE POOL pool4;
gaussdb=# DROP RESOURCE POOL pool5;

-- 删除控制组（命令行执行）。
gs_ssh -c "gs_cgroup -d -S class1"
```

相关链接

ALTER RESOURCE POOL，**DROP RESOURCE POOL**

1.13.9.37 CREATE ROLE

功能描述

创建角色。

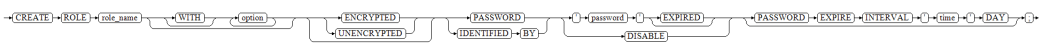
角色是拥有GaussDB数据库对象和权限的实体，在不同的环境中角色可以表示用户、组或者同时存在。

注意事项

- 在数据库中添加的新角色默认无LOGIN属性，仅拥有LOGIN属性的角色可以登录数据库。
- 创建角色的用户必须具备CREATE ROLE的权限或者是系统管理员。

语法格式

```
CREATE ROLE role_name [ [ WITH ] option [ ... ] ] [ ENCRYPTED | UNENCRYPTED ] { PASSWORD |
IDENTIFIED BY } { 'password' [EXPIRED] | DISABLE }[ PASSWORD EXPIRE INTERVAL 'time' DAY ];
```



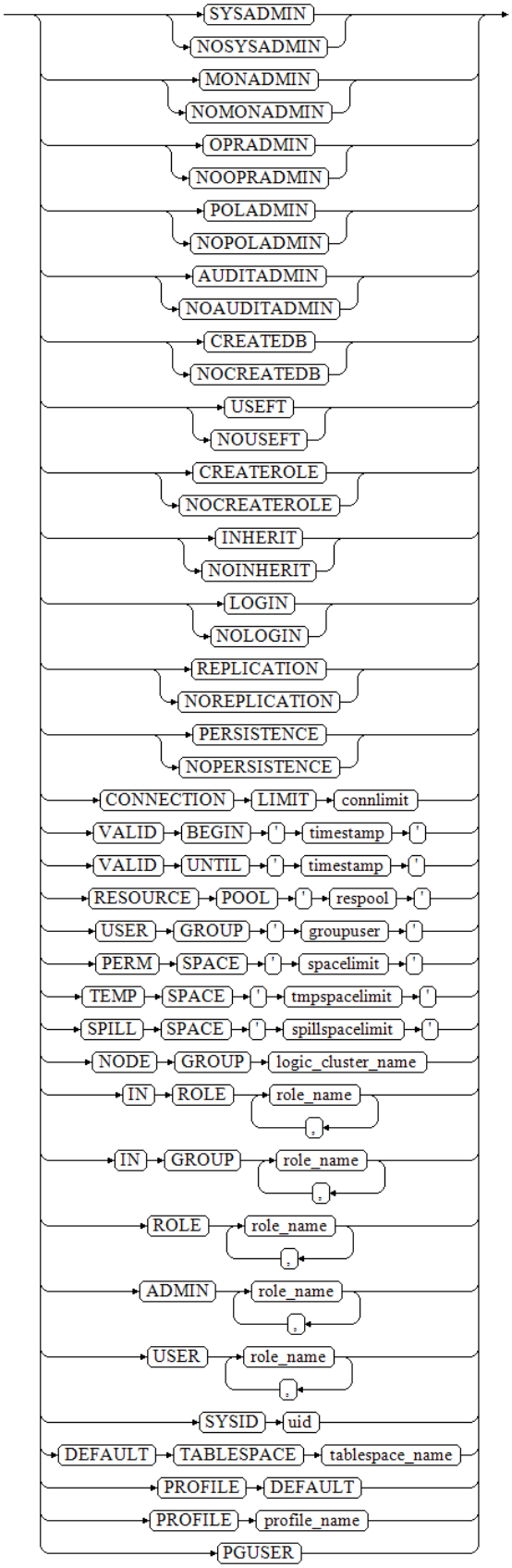
其中角色信息设置子句option语法为：

```
{
   {SYSADMIN | NOSYSADMIN}
   | {MONADMIN | NOMONADMIN}
   | {OPRADMIN | NOOPRADMIN}
   | {POLADMIN | NOPOLADMIN}
   | {AUDITADMIN | NOAUDITADMIN}
   | {CREATEDB | NOCREATEDB}
   | {USEFT | NOUSEFT}
   | {CREATEROLE | NOCREATEROLE}
   | {INHERIT | NOINHERIT}
   | {LOGIN | NOLOGIN}
   | {REPLICATION | NOREPLICATION}
   | {PERSISTENCE | NOPERSISTENCE}
   | CONNECTION LIMIT connlimit
```


版权所有 © 华为云计算技术有限公司

```
    | VALID BEGIN 'timestamp'
    | VALID UNTIL 'timestamp'
    | RESOURCE POOL 'respool'
    | USER GROUP 'groupuser'
    | PERM SPACE 'spacelimit'
    | TEMP SPACE 'tmpspacelimit'
    | SPILL SPACE 'spillspacelimit'
    | NODE GROUP logic_cluster_name
    | IN ROLE role_name [, ...]
    | IN GROUP role_name [, ...]
    | ROLE role_name [, ...]
    | ADMIN role_name [, ...]
    | USER role_name [, ...]
    | SYSID uid
    | DEFAULT TABLESPACE tablespace_name
    | PROFILE DEFAULT
    | PROFILE profile_name
    | PGUSER
}
```


版权所有 © 华为云计算技术有限公司



版权所有 © 华为云计算技术有限公司

参数说明

- **role_name**

  角色的名称。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  说明

  - 角色名称若超过63个字符，数据库会截断并保留前63个字符当作角色名称。
  - 当角色名称中包含大写字母时，数据库会自动转换为小写字母，如果需要创建包含大写字母的角色名称则需要使用双引号进行包裹。
  - 角色名称需要为小写字母（a~z）、大写字母（A~Z）、下划线（_）、数字（0~9）或美元符号（$），且必须以字母或下划线开头。
  - redisuser为重分布资源管控对象，当前不支持创建role_name为redisuser的角色。

- **password**

  角色的登录密码。

  取值范围：不为空的字符串。

  密码规则如下：

  - 密码默认不少于8个字符。
  - 不能与角色名及角色名倒序相同。
  - 至少包含大写字母（A~Z）、小写字母（a~z）、数字（0~9）和非字母数字字符（限定为~、!、@、#、$、%、^、&、*、(、)、-、_、=、+、\、|、[、{、}、]、;、:、,、<、.、>、/或?）四类字符中的三类字符。
  - 密码也可以使用符合格式要求的密文字符串，主要用于用户数据导入场景，不推荐用户直接使用。数据库不会对密文密码复杂度进行校验，直接使用密文密码可能存在数据安全风险，如果用户直接使用密文密码，需清楚密文密码对应的明文以及保证明文密码复杂度。
  - 创建角色时，用户密码需要使用单引号或双引号进行包裹。

- **EXPIRED**

  在创建角色时可指定EXPIRED，即创建密码失效角色。

  指定EXPIRED后，该角色不允许执行简单查询和扩展查询，仅当修改自身密码后才可正常执行语句。

- **PASSWORD EXPIRE INTERVAL 'time' DAY**

  在创建角色时，可以选择设置密码的有效天数，当密码的使用时间超过设置的有效天数后将禁止角色登录，需要对密码进行修改后才能登录。

  time为0~999的整数，当未指定有效天数时，默认为0，即密码长期有效。

- **DISABLE**

  默认情况下，角色可以更改自身密码，当指定DISABLE后，角色无法对自身密码进行修改，即禁用角色的密码。

  当指定DISABLE后，角色自身密码将从系统中删除，角色无法通过自身密码连接数据库，仅能通过外部认证进行连接操作，如kerberos认证。

  仅管理员才能启用或禁用密码，普通用户不能禁用初始用户的密码。启用密码时，请执行ALTER USER并指定密码。

- **ENCRYPTED | UNENCRYPTED**

  控制密码存储在系统表里的内容是否加密。

  按照产品安全要求，密码必须加密存储，UNENCRYPTED在GaussDB中禁止使用。

版权所有 © 华为云计算技术有限公司

由于系统无法对指定的加密口令字符串进行解密，所以当口令字符串已经是用SHA256加密的格式时，无论是否声明了ENCRYPTED或UNENCRYPTED，口令会继续按照该格式进行存储，实现dump/restore操作时重新加载加密的口令。

- **SYSADMIN | NOSYSADMIN**

  定义角色是否为系统管理员，默认为NOSYSADMIN。指定SYSADMIN后，角色拥有系统最高权限。

  - 三权分立关闭时，具有SYSADMIN属性的角色拥有创建具有SYSADMIN、REPLICATION、CREATEROLE、AUDITADMIN、MONADMIN、POLADMIN和CREATEDB属性的用户以及创建普通用户的权限。
  - 三权分立打开时，具有SYSADMIN属性的用户无权创建用户。

- **MONADMIN | NOMONADMIN**

  定义角色是否是监控管理员，默认为NOMONADMIN。

- **OPRADMIN | NOOPRADMIN**

  定义角色是否是运维管理员，默认为NOOPRADMIN。

- **POLADMIN | NOPOLADMIN**

  定义角色是否是安全策略管理员，默认为NOPOLADMIN。

- **AUDITADMIN | NOAUDITADMIN**

  定义角色是否有审计管理属性，默认为NOAUDITADMIN。

- **CREATEDB | NOCREATEDB**

  定义角色是否可以创建数据库，默认为NOCREATEDB。

- **USEFT | NOUSEFT**

  该参数为保留参数，暂未启用。

- **CREATEROLE | NOCREATEROLE**

  定义角色是否可以创建新角色，即该角色可以执行CREATE ROLE和CREATE USER操作，默认为NOCREATEROLE。指定CREATEROLE后，角色还可以修改和删除其他角色。

  - 三权分立关闭时，具有CREATEROLE属性的用户拥有创建具有CREATEROLE、AUDITADMIN、MONADMIN、POLADMIN和CREATEDB属性的用户以及创建普通用户的权限。
  - 三权分立打开时，具有CREATEROLE属性的用户拥有创建具有CREATEROLE、MONADMIN、POLADMIN和CREATEDB属性的用户以及创建普通用户的权限。

- **INHERIT | NOINHERIT**

  定义角色是否继承其所在组的角色的权限。不推荐使用。

- **LOGIN | NOLOGIN**

  定义角色是否可以登录数据库，具有LOGIN属性的角色才可以登录数据库，默认为NOLOGIN。

  一个拥有LOGIN属性的角色可以认为是一个用户。

- **REPLICATION | NOREPLICATION**

  定义角色是否允许流复制或设置系统为备份模式。REPLICATION属性是特定的角色，仅用于复制。

  默认为NOREPLICATION。

- **PERSISTENCE | NOPERSISTENCE**

版权所有 © 华为云计算技术有限公司

定义永久用户。仅允许初始用户创建、修改和删除具有PERSISTENCE属性的永久用户。

- **CONNECTION LIMIT connlimit**

  声明该角色可以使用的并发连接数量。

  取值范围：[-1, 2^{31}-1]的整数。默认值为-1，表示没有限制。

  须知

  - 系统管理员不受此参数的限制。
  - 每个CN的connlimit单独统计，集群整体的连接数= connlimit * 当前正常CN节点个数。

- **VALID BEGIN 'timestamp'**

  设置角色生效的时间，如果未指定生效时间，表示角色无有效开始时间限制。

  timestamp为生效时间，格式为'YYYY-MM-DD HH:mm:ss'。

- **VALID UNTIL 'timestamp'**

  设置角色失效的时间戳。如果未指定失效时间，表示角色无有效结束时间限制。

  timestamp为失效时间，格式为'YYYY-MM-DD HH:mm:ss'。

- **RESOURCE POOL 'respool'**

  设置角色使用的资源池名称，该名称属于系统表pg_resource_pool。

- **USER GROUP 'groupuser'**

  创建一个user的子用户。

- **PERM SPACE 'SPACELIMIT'**

  设置角色使用空间的大小。

- **TEMP SPACE 'tmpspacelimit'**

  设置角色临时表存储空间限额。

- **SPILL SPACE 'spillspacelimit'**

  设置角色算子落盘空间限额。

- **IN ROLE role_name**

  新角色立即拥有IN ROLE子句中列出的一个或多个现有角色拥有的权限。不推荐使用。

- **IN GROUP role_name**

  IN GROUP是IN ROLE过去时的语法结构。不推荐使用。

- **ROLE role_name**

  ROLE子句列出一个或多个现有的角色，它们将自动添加为这个新角色的成员，拥有新角色所有的权限。

- **ADMIN role_name**

  ADMIN子句类似ROLE子句，不同的是ADMIN后的角色可以把新角色的权限赋给其他角色。

- **USER role_name**

  USER子句是ROLE子句过时的拼法。

版权所有 © 华为云计算技术有限公司

- **SYSID uid**
  SYSID子句将被忽略，无实际意义。
- **DEFAULT TABLESPACE tablespace_name**
  DEFAULT TABLESPACE子句将被忽略，无实际意义。
- **PROFILE profile_name**
  PROFILE子句将被忽略，无实际意义。
- **PGUSER**
  当前版本该属性没有实际意义，仅作为语法的前向兼容而保留。

示例

- CREATE ROLE与CREATE USER的区别

```
--使用CREATE ROLE创建角色test_role。
gaussdb=# CREATE ROLE test_role PASSWORD '********';
--使用CREATE USER创建用户test_user。
gaussdb=# CREATE USER test_user PASSWORD '********';

--查看信息，CREATE ROLE创建的角色默认禁止登录数据库。
gaussdb=# \du test*
           List of roles
 Role name |  Attributes  | Member of
-----------+--------------+-----------
 test_role | Cannot login | {}
 test_user |              | {}

--使用如下SQL可以使test_role角色正常登录数据库。
gaussdb=# ALTER ROLE test_role WITH LOGIN;

gaussdb=# \du test*
           List of roles
 Role name | Attributes | Member of
-----------+------------+-----------
 test_role |            | {}
 test_user |            | {}

--查看SCHEMA信息，CREATE USER创建用户时，会自动创建同名SCHEMA。
gaussdb=# \dn test*
    List of schemas
   Name    |   Owner
-----------+-----------
 test_user | test_user
(1 row)

--删除。
gaussdb=# DROP ROLE test_role;
gaussdb=# DROP USER test_user;
```

- 创建权限类角色、三权分立

```
--创建具有OPRADMIN权限的用户lisa和具有SYSADMIN权限的用户bob。
gaussdb=# CREATE ROLE lisa WITH OPRADMIN PASSWORD '*********';
gaussdb=# CREATE ROLE bob WITH SYSADMIN PASSWORD '*********';

--查看权限。
gaussdb=# \du lisa;
                    List of roles
 Role name |         Attributes          | Member of
-----------+-----------------------------+-----------
 lisa      | Cannot login, Operatoradmin | {}

gaussdb=# \du bob
                 List of roles
 Role name |       Attributes       | Member of
-----------+------------------------+-----------
 bob       | Cannot login, Sysadmin | {}
```


版权所有 © 华为云计算技术有限公司

```
--创建具有CREATEDB权限的角色role11。
gaussdb=# CREATE ROLE role11 CREATEDB PASSWORD '********';
CREATE ROLE

--查看权限。
gaussdb=# \du role11
                List of roles
 Role name |       Attributes        | Member of
-----------+-------------------------+-----------
 role11    | Create DB, Cannot login | {}

--使用ADMIN子句创建新角色role12。
gaussdb=# CREATE ROLE role12 WITH  CREATEDB  ADMIN role11  PASSWORD '********';
CREATE ROLE
--查看权限。
gaussdb=# \du role11
                List of roles
 Role name |       Attributes        | Member of
-----------+-------------------------+-----------
 role11    | Create DB, Cannot login | {role12}

gaussdb=# \du role12
                List of roles
 Role name |       Attributes        | Member of
-----------+-------------------------+-----------
 role12    | Create DB, Cannot login | {}

--删除角色。
gaussdb=# DROP ROLE role11;
gaussdb=# DROP ROLE role12;
gaussdb=# DROP ROLE lisa;
gaussdb=# DROP ROLE bob;

--三权分立配置：设置参数enableSeparationOfDuty为on，则会开启三权分立配置，限制系统管理员的权限。不允许系统管理员创建或更改用户配置。

--查看三权分立是否打开。
gaussdb=# SHOW enableSeparationOfDuty;
 enableSeparationOfDuty
------------------------
 off
(1 row)
--执行以下语句修改参数。
gs_guc set -Z coordinator -Z datanode -N all -I all -c "enableSeparationOfDuty = on"

--设置成功后，重启数据库生效。
gs_om -t stop
gs_om -t start

--再次查看三权分立。
gaussdb=# SHOW enableSeparationOfDuty;
 enableSeparationOfDuty
------------------------
 on
(1 row)
```

- 创建密码失效角色

```
--创建密码失效角色test_role2。
gaussdb=# CREATE ROLE test_role2 PASSWORD '********' EXPIRED;
gaussdb=# ALTER ROLE test_role2 WITH LOGIN;

--test_role2登录到数据库后无法进行任何操作，只有根据提示修改密码后才可以进行操作。
gaussdb=# SET ROLE test_role2 PASSWORD '********';
gaussdb=> \d
ERROR:  Please use "ALTER ROLE user_name IDENTIFIED BY 'password' REPLACE 'old password';" to modify the expired password of user test_role2 before operation!

--修改test_role2密码。
gaussdb=> ALTER ROLE test_role2 IDENTIFIED BY '********' REPLACE '********';
```


版权所有 © 华为云计算技术有限公司

```
--删除。
gaussdb=> RESET ROLE;
gaussdb=# DROP ROLE test_role2;
```

- 创建角色指定生效与失效日期

```
--创建一个角色，从2015年1月1日开始生效，到2026年1月1日失效。
gaussdb=# CREATE ROLE test_role3 WITH LOGIN PASSWORD '********' VALID BEGIN '2015-01-01'
VALID UNTIL '2026-01-01';
--删除。
gaussdb=# DROP ROLE test_role3;
```

- 创建角色设置密码有效期

```
--创建一个角色，有效期为30天。
gaussdb=# CREATE ROLE role4 PASSWORD '********' PASSWORD EXPIRE INTERVAL 30 DAY;
```

相关链接

SET ROLE，**ALTER ROLE**，**DROP ROLE**，**GRANT**，**REVOKE**

1.13.9.38 CREATE ROW LEVEL SECURITY POLICY

功能描述

对表创建行访问控制策略。

注意事项

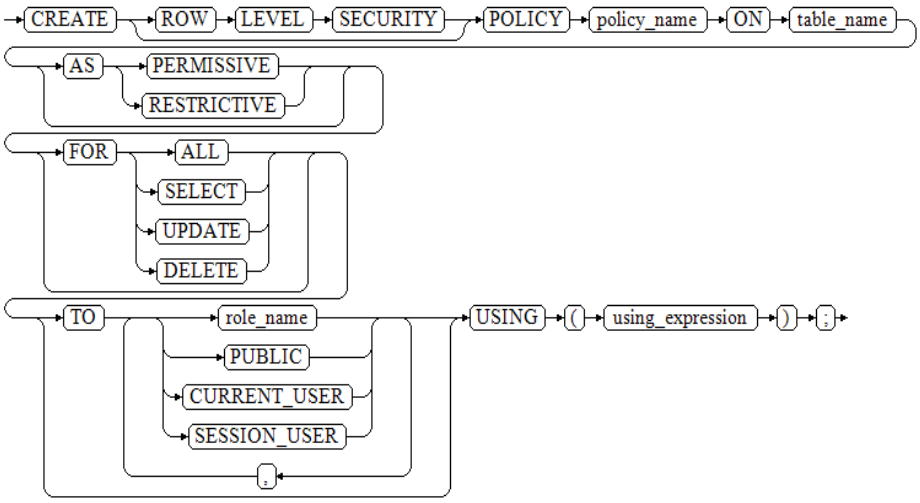
- 支持对行存表、行存分区表、复制表、unlogged表以及hash表创建行访问控制策略。
- 不支持外表或临时表创建行访问控制策略。
- 不支持对视图创建行访问控制策略。
- 同一张表上可以创建多个行访问控制策略，一张表最多创建100个行访问控制策略。
- 系统管理员不受行访问控制影响，可以查看表的全量数据。
- 通过SQL语句、视图、函数以及存储过程查询包含行访问控制策略的表，都会受影响。
- 不支持对添加了行级访问控制策略的表字段进行修改数据类型操作。
- 当对表创建了行访问控制策略，只有打开该表的行访问控制开关（ALTER TABLE ... ENABLE ROW LEVEL SECURITY），策略才能生效，否则不生效。
- 当前行访问控制影响数据表的读取操作（SELECT、UPDATE或DELETE），暂不影响数据表的写入操作（INSERT以及MERGE INTO）。
- 表所有者或系统管理员可以在USING子句中创建表达式，在客户端执行数据表读取操作时，数据库后台在查询重写阶段会将满足条件的表达式拼接并应用到执行计划中。针对数据表的每一条元组，当USING表达式返回true时，元组对当前用户可见，当USING表达式返回false或null时，元组对当前用户不可见。
- 行访问控制策略名称仅针对于指定的表，同一个数据表上不能有同名的行访问控制策略；对不同的数据表，可以存在同名的行访问控制策略。

语法格式

```
CREATE [ ROW LEVEL SECURITY ] POLICY policy_name ON table_name
    [ AS { PERMISSIVE | RESTRICTIVE } ]
    [ FOR { ALL | SELECT | UPDATE | DELETE } ]
```


版权所有 © 华为云计算技术有限公司

```
[ TO { role_name | PUBLIC | CURRENT_USER | SESSION_USER } [, ...] ]
USING ( using_expression );
```



参数说明

- **policy_name**

  行访问控制策略名称，同一个数据表上行访问控制策略名称不能相同。

- **table_name**

  行访问控制策略的表名。

- **PERMISSIVE | RESTRICTIVE**

  - PERMISSIVE：指定行访问控制策略为宽容性策略，宽容性策略的条件用OR表达式拼接。
  - RESTRICTIVE：指定行访问控制策略为限制性策略，限制性策略的条件用AND表达式拼接。

  具体的拼接方式如下：

  ```
  (using_expression_permissive_1 OR using_expression_permissive_2 ...) AND
  (using_expression_restrictive_1 AND using_expression_restrictive_2 ...)
  ```

  未指定时默认为PERMISSIVE。

- **command**

  当前行访问控制影响的SQL操作，可指定操作包括ALL、SELECT、UPDATE以及DELETE。未指定时默认为ALL。

  - ALL：影响SELECT、UPDATE以及DELETE类的所有操作。
  - SELECT：SELECT类操作受行访问控制的影响，只能查看到满足条件（using_expression返回值为true）的元组数据。受影响的操作包括SELECT、SELECT FOR UPDATE/SHARE、UPDATE ... RETURNING以及DELETE ... RETURNING。
  - UPDATE：UPDATE类操作受行访问控制的影响，只能更新满足条件（using_expression返回值为true）的元组数据。受影响的操作包括UPDATE、UPDATE ... RETURNING以及SELECT ... FOR UPDATE/SHARE。
  - DELETE：DELETE类操作受行访问控制的影响，只能删除满足条件（using_expression返回值为true）的元组数据。受影响的操作包括DELETE以及DELETE ... RETURNING。

  行访问控制策略与适配的SQL语法关系如**表1-397**所示。

版权所有 © 华为云计算技术有限公司

表 1-397 ROW LEVEL SECURITY 策略与适配 SQL 语法关系

Command	SELECT/ALL policy	UPDATE/ALL policy	DELETE/ALL policy
SELECT	Existing row	No	No
SELECT FOR UPDATE/SHARE	Existing row	Existing row	No
UPDATE	No	Existing row	No
UPDATE RETURNING	Existing row	Existing row	No
DELETE	No	No	Existing row
DELETE RETURNING	Existing row	No	Existing row

- **role_name | PUBLIC | CURRENT_USER | SESSION_USER**

  行访问控制影响的数据库用户/角色。行访问控制策略可以应用到指定的用户/角色，也可应用到全部用户（PUBLIC）；若未指定受影响的用户，默认为PUBLIC。

  - role_name：指定的数据库用户。
  - CURRENT_USER：当前执行环境的用户名；
  - SESSION_USER：会话用户名；
  - PUBLIC：影响所有数据库用户，可以指定多个受影响的数据库用户。

  须知

  系统管理员不受行访问控制特性影响。

- **using_expression**

  行访问控制的表达式（返回boolean值）。

  说明

  - 条件表达式中不能包含agg函数和窗口函数。
  - 在查询重写阶段，如果数据表的行访问控制开关打开，满足条件的表达式会添加到计划树中。
  - 针对数据表的每条元组，会进行表达式计算，只有表达式返回值为true时，行数据对用户才可见（SELECT、UPDATE、DELETE）；当表达式返回false时，该元组对当前用户不可见，用户无法通过SELECT语句查看此元组，无法通过UPDATE语句更新此元组，无法通过DELETE语句删除此元组。

示例

```
--创建用户alice。
gaussdb=# CREATE USER alice PASSWORD '*********';

--创建用户bob。
gaussdb=# CREATE USER bob PASSWORD '*********';

--创建数据表all_data。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE TABLE public.all_data(id int, role varchar(100), data varchar(100));

--向数据表插入数据。
gaussdb=# INSERT INTO all_data VALUES(1, 'alice', 'alice data');
gaussdb=# INSERT INTO all_data VALUES(2, 'bob', 'bob data');
gaussdb=# INSERT INTO all_data VALUES(3, 'peter', 'peter data');

--将表all_data的读取权限赋予alice和bob用户。
gaussdb=# GRANT SELECT ON all_data TO alice, bob;

--打开行访问控制策略开关。
gaussdb=# ALTER TABLE all_data ENABLE ROW LEVEL SECURITY;

--创建行访问控制策略，当前用户只能查看用户自身的数据。
gaussdb=# CREATE ROW LEVEL SECURITY POLICY all_data_rls ON all_data USING(role = CURRENT_USER);

--查看表all_data相关信息。
gaussdb=# \d+ all_data
                        Table "public.all_data"
 Column |          Type          | Modifiers | Storage  | Stats target | Description
--------+------------------------+-----------+----------+--------------+-------------
 id     | integer                |           | plain    |              |
 role   | character varying(100) |           | extended |              |
 data   | character varying(100) |           | extended |              |
Row Level Security Policies:
    POLICY "all_data_rls" FOR ALL
      TO public
      USING (((role)::name = "current_user"()))
Has OIDs: no
Distribute By: HASH(id)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no, storage_type=USTORE, segment=off, enable_rowsecurity=true

--当前用户执行SELECT操作。
gaussdb=# SELECT * FROM all_data;
 id | role  |    data
----+-------+------------
  1 | alice | alice data
  2 | bob   | bob data
  3 | peter | peter data
(3 rows)

gaussdb=# EXPLAIN(COSTS OFF) SELECT * FROM all_data;
 id |          operation
----+-----------------------------
  1 | ->  Streaming (type: GATHER)
  2 |    ->  Seq Scan on all_data
(2 rows)

--切换至alice用户执行SELECT操作。
gaussdb=# SET ROLE alice PASSWORD '********';
gaussdb=> SELECT * FROM all_data;
 id | role  |    data
----+-------+------------
  1 | alice | alice data
(1 row)

gaussdb=> EXPLAIN(COSTS OFF) SELECT * FROM all_data;
 id |          operation
----+-----------------------------
  1 | ->  Streaming (type: GATHER)
  2 |    ->  Seq Scan on all_data
(2 rows)

        Predicate Information (identified by plan id)
-------------------------------------------------------------------
   2 --Seq Scan on all_data
        Filter: ((role)::name = "current_user"())
 Notice: This query is influenced by row level security feature
```


版权所有 © 华为云计算技术有限公司

```
(3 rows)

--插入数据，因未授权插入权限，则会报错。
gaussdb=> INSERT INTO all_data VALUES(4, 'test', 'test data');
ERROR: The operation permission on relation all_data is denied.
DETAIL: N/A.

--将表all_data的插入权限赋予alice用户。
gaussdb=> RESET ROLE;
gaussdb=# GRANT INSERT ON all_data TO alice;

--切换至alice用户并插入数据。
gaussdb=# SET ROLE alice PASSWORD '********';
gaussdb=> INSERT INTO all_data VALUES(4, 'alice', 'test data');
INSERT 0 1

--查询表all_data。
gaussdb=> SELECT * FROM all_data;
 id | role  |    data
----+-------+------------
  1 | alice | alice data
  4 | alice | test data
(2 rows)

--删除行访问控制策略。
gaussdb=> RESET ROLE;
gaussdb=# DROP ROW LEVEL SECURITY POLICY all_data_rls ON all_data;

--删除数据表all_data。
gaussdb=# DROP TABLE public.all_data;

--删除用户alice, bob。
gaussdb=# DROP USER alice, bob;
```

相关链接

DROP ROW LEVEL SECURITY POLICY，**ALTER ROW LEVEL SECURITY POLICY**

1.13.9.39 CREATE SCHEMA

功能描述

创建模式。

注意事项

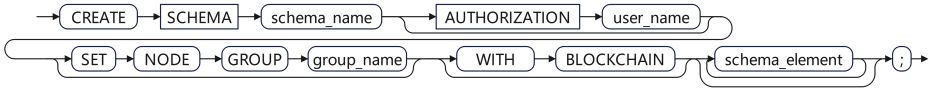
- 数据库中拥有CREATE权限的用户可以创建模式。
- 系统管理员在普通用户同名的Schema下创建的对象，其所有者为Schema的同名用户（非系统管理员）。
- 访问命名对象时可以使用模式名作为前缀进行具体访问，如果无模式名前缀，则访问当前模式下的命名对象。创建命名对象时也可用模式名作为前缀进行具体指定。
- CREATE SCHEMA可以包括在新模式中创建对象的子命令，这种子命令和先进行创建模式再执行子命令的方式没有区别。如果使用了AUTHORIZATION子句，则所有创建的对象都将被该用户所拥有。

语法格式

- 根据指定模式名称创建模式

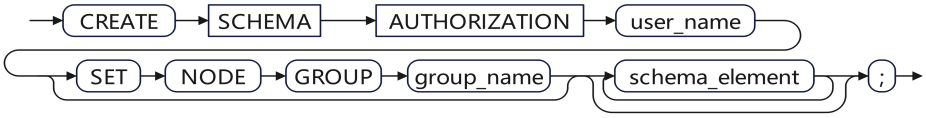
版权所有 © 华为云计算技术有限公司

```
CREATE SCHEMA schema_name
   [ AUTHORIZATION user_name ] [SET NODE GROUP group_name] [WITH BLOCKCHAIN]
[ schema_element [ ... ] ];
```



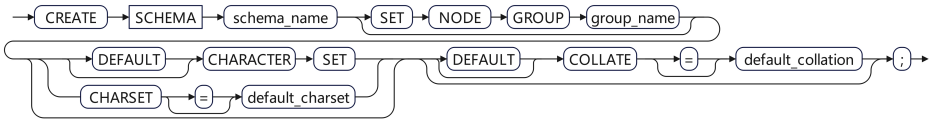
- 根据用户名创建模式

```
CREATE SCHEMA AUTHORIZATION user_name [SET NODE GROUP group_name] [ schema_element
[ ... ] ];
```



- 创建模式并指定默认字符集和字符序

```
CREATE SCHEMA schema_name [SET NODE GROUP group_name]
   [ [DEFAULT] CHARACTER SET | CHARSET [ = ] default_charset ] [ [DEFAULT] COLLATE [ = ]
default_collation ];
```



参数说明

- **schema_name**

  模式名称。

  取值范围：字符串，要符合**标识符命名规范**。

  模式新名称规则如下：

  - 新名称不能和当前数据库里其他的模式重名。
  - 新名称不能和当前数据库的初始用户重名。
  - 请勿使用“pg_”或“gs_role_”开头或“data_redis”命名新名称。

- **AUTHORIZATION user_name**

  指定模式的所有者。当不指定schema_name时（即**根据用户名创建模式**的方式），将user_name作为模式名，此时user_name只能是角色名。

  取值范围：已存在的用户名/角色名。

- **SET NODE GROUP group_name**

  设置模式的单节点属性。group_name表示单节点组Node Group名称，仅支持设置系统预定的类型为's'的单节点组Node Group，其名称前缀为“sys_single_group”。

  说明

  - 在M-Compatibility模式数据库下（即sql_compatibility = 'M'）不支持该语法。
  - CREATE SCHEMA语法包含CREATE TABLE子句的复合SQL语句不支持单节点表相关功能，既不支持单节点表创建，也不支持模式设置单节点表属性。

- **WITH BLOCKCHAIN**

  指定模式的防篡改属性。

版权所有 © 华为云计算技术有限公司

说明

- 将普通模式修改为防篡改模式，需将GUC参数enable_ledger设置为on。该参数默认值为off，级别为SIGHUP。
- 防篡改模式下的行存普通用户表将自动扩展为防篡改用户表。

- **schema_element**

  在模式里创建对象的SQL语句。

  当前仅支持CREATE TABLE、CREATE VIEW、CREATE INDEX、CREATE TABLE PARTITION、CREATE SEQUENCE、CREATE TRIGGER以及GRANT子句。

  说明

  - schema_element所创建的对象都被AUTHORIZATION user_name子句指定的user_name所拥有。
  - 如果当前搜索路径内的模式存在同名对象时，需要明确指定引用对象所在的模式。可以通过命令SHOW search_path查看当前搜索路径上的模式。

- **default_charset**

  指定模式的默认字符集。未指定该选项时，模式的默认字符集为指定的字符序COLLATE的默认字符集。

  取值范围：支持设置的字符集请参见**表1-398**。

  说明

  仅在MYSQL模式数据库或M-Compatibility模式数据库下（即sql_compatibility = 'MYSQL'或sql_compatibility = 'M'）支持该语法。以上行为是在sql_compatibility = 'MYSQL'时的表现，sql_compatibility = 'M'的行为请参见《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > C > CREATE SCHEMA”的内容。

- **default_collation**

  指定模式的默认字符序。未指定该选项时，模式的默认字符序为指定的字符集CHARSET的默认字符序。

  取值范围：支持设置的字符集请参见**表1-398**。

  说明

  仅在MYSQL模式数据库或M-Compatibility模式数据库下（即sql_compatibility = 'MYSQL'或sql_compatibility = 'M'）支持该语法。以上行为是在sql_compatibility = 'MYSQL'时的表现，sql_compatibility = 'M'的行为请参见《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > C > CREATE SCHEMA”的内容。

示例

```
--创建一个角色role1。
gaussdb=# CREATE ROLE role1 IDENTIFIED BY '********';
CREATE ROLE

--为用户role1创建一个同名Schema，schema_element子命令创建的表films和winners的拥有者为role1。
gaussdb=# CREATE SCHEMA AUTHORIZATION role1
    CREATE TABLE films (title text, release date, awards text[])
    CREATE VIEW winners AS SELECT title, release FROM films WHERE awards IS NOT NULL;
CREATE SCHEMA

--创建新角色role2。
gaussdb=# CREATE ROLE role2 IDENTIFIED BY '********';
CREATE ROLE

--根据名称创建schema1并指定所有者role2。
gaussdb=# CREATE SCHEMA schema1 AUTHORIZATION role2;
```


版权所有 © 华为云计算技术有限公司

```
CREATE SCHEMA

--根据名称创建schema2并显示设置单节点表属性。
gaussdb=# CREATE SCHEMA schema2 SET NODE GROUP sys_single_group1;
CREATE SCHEMA

--修改防篡改属性。
--防篡改属性修改前，需要将GUC参数enable_ledger设置为on，enable_ledger默认为off，需要通过gs_guc进行设置。
gs_guc reload -Z coordinator -Z datanode -N all -I all -c "enable_ledger=on";

--设置完成后，查看enable_ledger状态是否为on。
gaussdb=# SHOW enable_ledger;
 enable_ledger
---------------
 on
(1 row)

--创建Schema并指定防篡改属性。
gaussdb=# CREATE SCHEMA ledgernsp WITH BLOCKCHAIN;
CREATE SCHEMA

--删除Schema。
gaussdb=# DROP SCHEMA role1 CASCADE;
gaussdb=# DROP SCHEMA schema1 CASCADE;
gaussdb=# DROP SCHEMA schema2 CASCADE;
gaussdb=# DROP SCHEMA ledgernsp CASCADE;

--删除用户。
gaussdb=# DROP USER role1 CASCADE;
gaussdb=# DROP USER role2 CASCADE;
```

相关链接

ALTER SCHEMA，**DROP SCHEMA**

1.13.9.40 CREATE SECURITY LABEL

功能描述

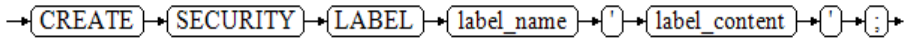
在当前数据库中为指定的安全策略创建新的安全标签。

注意事项

初始用户、具有SYSADMIN权限的用户或者继承了内置角色gs_role_seclabel权限的用户可以进行创建安全标签操作。

语法格式

```
CREATE SECURITY LABEL label_name 'label_content';
```



参数说明

- **label_name**

  安全标签名称，需在数据库内符合唯一约束。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

版权所有 © 华为云计算技术有限公司

说明

- 安全标签名称若超过63个字符，数据库会截断并保留前63个字符当作安全标签名称。
- 当安全标签名称中包含大写字母时数据库会自动转换为小写字母，如果需要创建包含大写字母的安全标签名称则需要使用双引号进行包裹。
- 安全标签名称需要为小写字母（a~z）、大写字母（A~Z）、下划线（_）、数字（0~9）或美元符号（$），且必须以字母或下划线开头。

- **label_content**

  安全标签内容，要求如下：

  安全标签由等级和范围两部分组成，两者中间用冒号分隔。形如“等级类别:范围类别”，其中等级类别有且仅由一个等级组成，范围类别可由多个范围组成，但至少需要有一个范围，例如“L1:G2,G41,G6-G27”。

  - 等级分类中有1024个等级，命名为Li，其中1≤i≤1024，等级满足偏序关系（若i ≤ j，则Li ≤ Lj）。例如：等级L1小于等级L3。
  - 范围分类中有1024个范围，命名为Gi，其中1≤i≤1024，范围之间无法比较大小，但可以进行集合运算，多个范围之间用逗号分隔，连字符表示区间。例如：{G2-G5}表示{G2,G3,G4,G5}，集合{G1}是集合{G1,G6}的子集。
  - 等级和范围的首字母L和G均为大写；L和G之后至少要有一个数字字符，且第一位非零，不允许出现其他非数字字符；{Gxxx-Gyyy}形式中数字yyy必须大于等于xxx。
  - 不符合要求的等级和范围均为非法输入，系统会报错。

```
--例子
gaussdb=# CREATE SECURITY LABEL sec_label3 'L3:';  //标签内容范围类别至少要有一个才行。
ERROR:  in label text "L3:", there at least have one level and one group
```

示例

```
--创建安全标签sec_label。
gaussdb=# CREATE SECURITY LABEL sec_label 'L1:G4';

--创建安全标签sec_label，安全标签内容为'L1:G2,G4'。
gaussdb=# CREATE SECURITY LABEL sec_label 'L1:G2,G4';
ERROR:  security label "sec_label" already exists
--创建安全标签sec_label1，安全标签内容为'L1:G2,G4'。
gaussdb=# CREATE SECURITY LABEL sec_label1 'L1:G2,G4';

--创建安全标签sec_label2，安全标签内容为'L3:G1-G5'。
gaussdb=# CREATE SECURITY LABEL sec_label2 'L3:G1-G5';

--查看系统中已创建好的安全标签。
gaussdb=# SELECT * FROM gs_security_label;
 label_name | label_content
------------+---------------
 sec_label  | L1:G4
 sec_label1 | L1:G2,G4
 sec_label2 | L3:G1-G5
(3 rows)

--删除已存在的安全标签sec_label、sec_label1、sec_label2。
gaussdb=# DROP SECURITY LABEL sec_label;
gaussdb=# DROP SECURITY LABEL sec_label1;
gaussdb=# DROP SECURITY LABEL sec_label2;

--再次查看系统中已创建好的安全标签。
gaussdb=# SELECT * FROM gs_security_label;
 label_name | label_content
------------+---------------
(0 rows)
```


版权所有 © 华为云计算技术有限公司

相关链接

DROP SECURITY LABEL，SECURITY LABEL ON

1.13.9.41 CREATE SEQUENCE

功能描述

向当前数据库增加一个新的序列。序列是指一个存放等差数列的特殊表，该表不存储常规数据，通常用于为行或者表生成唯一的标识符。

注意事项

- 仅当用户被授予CREATE ANY SEQUENCE权限，才允许在PUBLIC模式和用户模式下进行创建序列操作。创建序列的用户将成为序列的所有者。
- 创建序列后，使用序列的nextval()函数以及generate_series(1,N)函数对表插入数据时，需保证nextval()函数的可调用次数大于等于N+1次，否则会因generate_series()内部调用N+1次nextval()而导致报错。
- 序列不支持与rowid、rowno系统列关联。

语法格式

```
CREATE [LARGE] SEQUENCE [ IF NOT EXISTS ] name [ INCREMENT [ BY ] increment ]
    [ MINVALUE minvalue | NO MINVALUE | NOMINVALUE ] [ MAXVALUE maxvalue | NO MAXVALUE | NOMAXVALUE]
    [ START [ WITH ] start ] [ CACHE cache | NOCACHE ] [ [ NO ] CYCLE | NOCYCLE ] [ RESTART [ WITH start ] ]
    [ OWNED BY { table_name.column_name | NONE } ];
```

参数说明

- **LARGE**

  如果指定LARGE关键字，会创建int128位的序列；未指定LARGE关键字，则默认创建int64的序列。

  SEQUENCE默认最大值为2^{63}-1，如果使用了LARGE标识，最大值可以支持到2^{127}-1。

- **IF NOT EXISTS**

  如果指定IF NOT EXISTS关键字，创建序列前会在当前Schema中查找是否存在同名序列，若存在则不会新建，返回NOTICE提示。

  未指定IF NOT EXISTS关键字时，若Schema中存在同名序列，返回ERROR告警。

- **name**

  创建的序列名称。可以使用模式进行具体指定，指定模式名，则在指定的模式中创建该序列，否则会在当前模式中进行创建。同一模式下，序列名需满足唯一约束，不能和同一个模式中的其他序列、表、索引、视图或外表的名称相同。

  取值范围：字符串，要符合标识符命名规范且最大长度不超过63个字符。

- **increment**

  可选，指定序列的步长。

  - 正数：将生成一个递增的序列。
  - 负数：将生成一个递减的序列。

  默认值为1。

版权所有 © 华为云计算技术有限公司

说明

在MYSQL兼容模式下，步长为浮点数时会自动转为整型。其他模式下，该参数不支持输入浮点数。

- **MINVALUE minvalue | NO MINVALUE| NOMINVALUE**

  可选，执行序列的最小值。NOMINVALUE等价于NO MINVALUE。

  如果没有声明minvalue或者声明了NO MINVALUE时：

  - 对于未声明LARGE关键字的序列，递增序列的默认值为1，递减序列的默认值为$-2^{63}+1$。
  - 对于声明了LARGE关键字的序列，递增序列的默认值为1，递减序列的默认值为-2^{127}。

- **MAXVALUE maxvalue | NO MAXVALUE| NOMAXVALUE**

  可选，执行序列的最大值。NOMAXVALUE等价于NO MAXVALUE。

  如果没有声明maxvalue或者声明了NO MAXVALUE：

  - 对于未声明LARGE关键字的序列，递增序列的默认值为$2^{63}-1$，递减序列的默认值为-1。
  - 对于声明了LARGE关键字的序列，递增序列的默认值为$2^{127}-1$，递减序列的默认值为-1。

  说明

  序列与表字段绑定时，若序列值增长超过字段类型最大值，将导致插入报错。因此，字段类型必须能容纳序列的最大可能值，避免业务异常。

- **START [WITH] start**

  可选。指定序列的起始值。

  默认值：对于递增序列为minvalue，递减序列为maxvalue。

- **CACHE cache | NOCACHE**

  可选。为了快速访问，而在内存中预先存储序列号的个数。NOCACHE等价于不指定cache值。

  默认值为1，表示一次只能生成一个值，即没有缓存。

  说明

  - 不建议同时定义cache和maxvalue/minvalue。定义cache后无法保证序列的连续性，可能会产生空洞，造成序列号段浪费。如果对并发性能有明确的要求，请同时设置GUC参数session_sequence_cache。
  - cache指定了单CN/DN一次向GTM中申请的值；session_sequence_cache指定的是单个会话一次向CN/DN申请缓存的值，会话结束后会自动丢弃。
  - cache越大，序列首次执行nextval()的耗时会越长。

- **[NO] CYCLE | NOCYCLE**

  可选，使序列达到maxvalue或者minvalue后可循环并继续循环。

  如果声明了NO CYCLE，则在序列达到其最大值后任何对nextval()的调用都会返回一个错误。

  - NOCYCLE的作用等价于NO CYCLE。当未指定循环循环选项时，默认值为NO CYCLE。
  - 若指定序列为CYCLE时，则不能保证序列的唯一性。

- **RESTART [WITH start]**

版权所有 © 华为云计算技术有限公司

可选。重新指定序列的起始值。

当未指定具体的start时，对于递增序列该选项的默认值为minvalue，对于递减序列该选项的默认值为maxvalue。

- **OWNED BY**

  可选。将序列和一个表的指定字段进行关联。在删除该字段或其所在表的时候，将自动删除已关联的序列。关联的表和序列的所有者必须是同一个用户，并且在同一个模式中。需要注意的是，通过指定OWNED BY，仅仅是建立了表的对应列和序列之间关联关系，并不会在插入数据时在该列上产生自增序列。

  默认值为OWNED BY NONE，表示不存在关联。

须知

通过OWNED BY创建的SEQUENCE不建议用于其他表，如果希望多个表共享序列，该序列不应该是只从属于特定的表。

示例

- 创建一个名为seq1的递增序列，从101开始，步长为10。

```
gaussdb=# CREATE SEQUENCE seq1
    START 101
    INCREMENT 10;

--从序列中选出下一个数字：
gaussdb=# SELECT nextval('seq1');
 nextval
---------
     101
(1 row)
gaussdb=# SELECT nextval('seq1');
 nextval
---------
     111

--删除序列。
gaussdb=# DROP SEQUENCE seq1;
```

- 表自增列的一种实现。

```
--创建表。
gaussdb=# CREATE TABLE test1(id int PRIMARY KEY, name varchar(20));

--创建与表关联的序列。
gaussdb=# CREATE SEQUENCE test_seq2
    START 1
    NO CYCLE
    OWNED BY test1.id;

--设置字段的默认值。
gaussdb=# ALTER TABLE test1 ALTER COLUMN id SET DEFAULT nextval('test_seq2'::regclass);

-- 插入数据。
gaussdb=# INSERT INTO test1 (name) values ('Joe'),('Scott'),('Ben');

--查询。
gaussdb=# SELECT * FROM test1;
 id | name
----+-------
  3 | Ben
  1 | Joe
  2 | Scott
(3 rows)
```


版权所有 © 华为云计算技术有限公司

```
--删除序列和表。
gaussdb=# DROP SEQUENCE test_seq2 CASCADE;
gaussdb=# DROP TABLE test1;
```

- 序列的循环。

```
--创建一个递增序列seq_test1，最大值为3，且可以循环。
gaussdb=# CREATE SEQUENCE seq_test1
    START 1
    MAXVALUE 3
    CYCLE;

--建表，并插数据。
gaussdb=# CREATE TABLE tbl_test1(
    c1 int DEFAULT nextval('seq_test1'),
    c2 varchar
);
gaussdb=# INSERT INTO tbl_test1 (c2) values ('test');
gaussdb=# INSERT INTO tbl_test1 (c2) values ('test');
gaussdb=# INSERT INTO tbl_test1 (c2) values ('test');
gaussdb=# INSERT INTO tbl_test1 (c2) values ('test');
gaussdb=# INSERT INTO tbl_test1 (c2) values ('test');

--查询表数据，当递增序列seq_test的值达到最大值后会重新从最小值开始。
gaussdb=# SELECT ctid,* FROM tbl_test1 ORDER BY 1,2;
 ctid  | c1 |  c2
-------+----+------
 (0,1) |  1 | test
 (0,1) |  3 | test
 (0,2) |  2 | test
 (0,3) |  1 | test
 (0,4) |  2 | test
(5 rows)

--删除表和序列。
gaussdb=# DROP TABLE tbl_test1;
gaussdb=# DROP SEQUENCE seq_test1;
```

相关链接

DROP SEQUENCE，**ALTER SEQUENCE**

1.13.9.42 CREATE SERVER

功能描述

创建一个外部服务器。外部服务器是存储其他同构集群信息的载体。

注意事项

- 默认只有系统管理员才可以创建外部服务器，否则需要对所使用的FOREIGN DATA WRAPPER授权才可以创建，授权语法为：

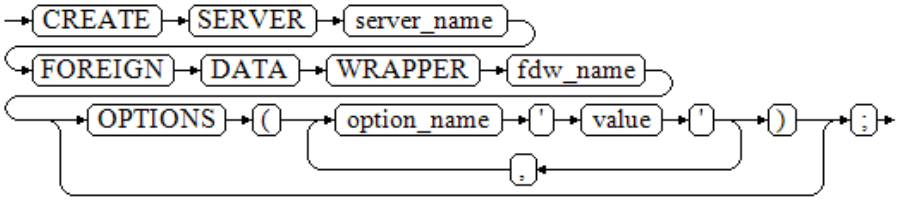
  ```
  GRANT USAGE ON FOREIGN DATA WRAPPER fdw_name TO username
  ```

  其中fdw_name为FOREIGN DATA WRAPPER的名称，username为创建SERVER的用户名。
- OPTIONS中的敏感字段（如password）在使用多层引号时，语义和不带引号的场景是不同的，因此不会被识别为敏感字段进行脱敏。

语法格式

```
CREATE SERVER server_name
    FOREIGN DATA WRAPPER fdw_name
    [ OPTIONS ( { option_name ' value ' } [, ...] ) ];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **server_name**

  创建的外部服务器名称。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

- **FOREIGN DATA WRAPPER fdw_name**

  指定外部数据封装器的名称。

  取值范围：fdw_name是数据库初始化时系统创建的数据封装器，对于其他同构集群，fdw_name为gc_fdw。还可以创建dist_fdw、file_fdw、log_fdw。其中：log_fdw仅为语法兼容，可以创建外表，无实际使用意义；dist_fdw用于gds导数，由于有内置gsmpp_server，因此不需要手动用dist_fdw创建server。

- **OPTIONS ({ option_name ' value ' } [, ...])**

  用于指定外部服务器的各类参数。

  - address

    外部服务器地址。

  - dbname

    外部服务器DB名称。

  - username

    外部服务器用户名。

  - password

    外部服务器密码。

示例

创建一个外部服务器，其中file_fdw为数据库中存在的FOREIGN DATA WRAPPER。

```
--创建外部服务器。
gaussdb=# CREATE SERVER my_server FOREIGN DATA WRAPPER file_fdw;

--删除外部服务器。
gaussdb=# DROP SERVER my_server;
```

建立另外一个同构集群的外部服务器，其中gc_fdw为数据库中存在的foreign data wrapper。

```
--创建外部服务器。
gaussdb=# CREATE SERVER server_remote FOREIGN DATA WRAPPER GC_FDW OPTIONS
    (address '10.146.187.231:8000,10.180.157.130:8000' ,
  dbname 'test',
  username 'test',
  password '********'
);

--删除外部服务器。
gaussdb=# DROP SERVER server_remote;
```


版权所有 © 华为云计算技术有限公司

相关链接

ALTER SERVER，**DROP SERVER**

1.13.9.43 CREATE SYNONYM

功能描述

创建一个同义词对象。同义词是数据库对象的别名，用于记录与其他数据库对象名间的映射关系，用户可以使用同义词访问关联的数据库对象。使用同义词时，用户需要具有关联对象的相应权限，当前支持通过同义词访问的数据库对象包括表、视图、类型、PACKAGE、函数和存储过程、序列和其他同义词对象。

注意事项

- 仅被授予了CREATE ANY SYNONYM权限的用户，可以在用户模式下创建同义词。
- 支持使用同义词的DML语句包括：SELECT、INSERT、UPDATE、DELETE、EXPLAIN以及CALL。
- 不支持通过同义词执行DDL语句（CREATE、DROP、ALTER）进行操作其关联的对象。
- 禁止在存储过程或函数内部进行创建同义词操作。同义词须在外部预先创建，存储过程内通过引用系统表pg_synonym中已存在的同义词对象进行使用。
- 为防止敏感信息泄露，禁止对包含密码等敏感信息的函数（如：加解密类函数gs_encrypt、gs_decrypt、gs_encrypt_bytea、gs_decrypt_bytea等）进行创建同义词操作以及禁止通过同义词调用此类函数。
- 不建议对临时表创建同义词。如需创建，须指定目标临时表的模式名，否则无法正常使用该同义词，并须在当前会话结束前执行DROP SYNONYM。
- 不支持对包含加密列的密态表以及基于密态表的视图、函数或存储过程进行创建同义词操作。
- 不支持通过\d、\df、\sf等元命令方式访问同义词关联对象的信息。
- 同义词支持嵌套：使用同义词时，若遇同义词指向另一同义词，系统将逐层追踪，直至找到最终关联对象。
- 使用同义词时，同义词搜索解析规则说明如下：
  - 设置search_path后，且未指定同义词Schema时，解析顺序如下：

    存储过程和函数：优先按照名称检索pg_proc表，若没有同名函数时，检索同义词，最后按照search_path进行检索。

    其他对象（表、视图等）：优先按search_path进行检索；同一Schema下，对象本身优先级高于同义词。
  - SQL语句给定Schema搜索对象schema1.object1：

    首先搜索schema1.object1对象是否存在。当该对象不存在时，搜索同义词schema1.object1是否存在。若schema1.object1同义词存在，搜索同义词关联的对象schema2.object2。只有对象schema2.object2和同义词schema2.object2都不存在且schema2是用户名时，会额外搜索PUBLIC同义词object2。
  - SQL语句不给定Schema搜索对象object1：

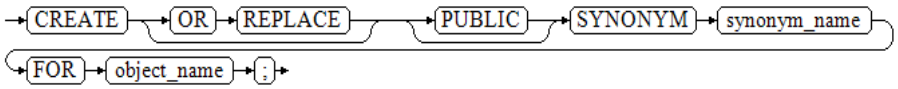
    首先遍历search_path，查找指定Schema下的对象schema.object1，当对象schema.object1不存在时，则搜索同义词schema.object1。只有当同义词

版权所有 © 华为云计算技术有限公司

schema.object1不存在且搜索的Schema为用户名时，会额外搜索PUBLIC同义词object1。

语法格式

```
CREATE [ OR REPLACE ] [PUBLIC] SYNONYM synonym_name
  FOR object_name;
```



参数说明

- **OR REPLACE**

  可选。如果同义词已存在，则重新定义。

- **PUBLIC**

  可选参数。创建PUBLIC同义词。

  说明

  - 同一数据库下，PUBLIC同义词不能重名。
  - 当数据库从不支持PUBLIC同义词的版本升级到支持PUBLIC同义词的版本时，升级提交前不支持创建、删除PUBLIC同义词。
  - 所有用户都可以访问PUBLIC同义词。除初始用户、系统管理员的用户外，创建和删除PUBLIC同义词需要具有对应CREATE PUBLIC SYNONYM和DROP PUBLIC SYNONYM的权限。
  - PUBLIC同义词在系统表pg_synonym中的synnamespace和synowner对应为0，在系统视图ADM_SYNONYMS和DB_SYNONYMS的owner为PUBLIC，schema_name为NULL。
  - 未指定同义词Schema情况下，首先检索同名对象，判断同名对象是否存在；然后检索同义词，按照search_path检索；最后检索PUBLIC同义词。指定同义词Schema情况下，不检索PUBLIC同义词。

- **synonym_name**

  创建的同义词名称，可以使用模式进行具体指定。

  取值范围：字符串，要符合**标识符命名规范**。

  说明

  若指定模式，则在指定模式中创建同义词，否则在当前模式下创建同义词。当同义词的模式为用户所在模式时，该同义词的所有者为模式的所有者；其他场景下，同义词的所有者默认为同义词的创建者。

- **object_name**

  关联的对象名称，可以使用模式进行具体指定。

  取值范围：字符串，要符合**标识符命名规范**。

版权所有 © 华为云计算技术有限公司

说明

- object_name可以是不存在的对象名称。
- object_name可以是PACKAGE对象，但不可关联PACKAGE内函数。可通过PACKAGE同义词访问PACKAGE内函数和变量。
- object_name可以是使用DATABASE LINK方式访问远程对象。DATABASE LINK的具体使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。
- 删除原对象后，与之关联的同义词不会被级联删除，继续访问该同义词将会报错：若原对象为表，提示已失效；若原对象为函数、存储过程或PACKAGE等，提示对象不存在。

示例

```
--创建模式ot。
gaussdb=# CREATE SCHEMA ot;

--创建表ot.test_tbl1。
gaussdb=# CREATE TABLE ot.test_tbl1(c1 INT, c2 INT);
gaussdb=# INSERT INTO ot.test_tbl1 values(1,1);

--查看当前search_path。
gaussdb=# SHOW search_path;
  search_path
----------------
 "$user",public
(1 row)

--因为当前search_path中不包含ot，并且当前用户不是ot，所以直接查看表名会报错。
gaussdb=# SELECT * FROM test_tbl1;
ERROR:  relation "test_tbl1" does not exist
LINE 1: SELECT * FROM test_tbl1;
                      ^
--创建同义词。
gaussdb=# CREATE OR REPLACE SYNONYM test_tbl1 FOR ot.test_tbl1;

--使用同义词。
gaussdb=# SELECT * FROM test_tbl1;
 c1 | c2
----+----
  1 |  1
(1 row)
gaussdb=# INSERT INTO test_tbl1 VALUES (2,2);

--查询已经创建的同义词。
gaussdb=# SELECT synname,synobjschema,synobjname FROM pg_synonym WHERE synname = 'test_tbl1';
  synname  | synobjschema | synobjname
-----------+--------------+------------
 test_tbl1 | ot           | test_tbl1
(1 row)

--删除。
gaussdb=# DROP SYNONYM test_tbl1;
gaussdb=# DROP TABLE ot.test_tbl1;
gaussdb=# DROP SCHEMA ot CASCADE;
```

相关链接

ALTER SYNONYM，**DROP SYNONYM**

版权所有 © 华为云计算技术有限公司

1.13.9.44 CREATE TABLE

功能描述

在当前数据库中创建一个新的空白表，该表由命令执行者所有。当前默认创建行存表。

注意事项

- 对于HASH分布表，当对非分布键添加主键约束或唯一约束，将默认建立全局二级索引，其中全局二级索引的约束请参见**功能约束**。对于非HASH分布表，主键约束和唯一约束必须包含分布键，否则会报错。
- 分布方式默认取值为HASH(column_name)，column_name包括：表的主键列、表的唯一约束列（如果存在）或表中数据类型支持作为分布列的第一列，优先级别为主键列>唯一约束列>表中数据类型支持作为分布列的第一列。当同时存在主键列以及多个唯一约束列且未指定表的分布方式时，按照优先级别选取表的主键列后，其余唯一约束列将默认建立全局二级索引。

  须知

  MYSQL兼容模式下（即sql_compatibility = 'MYSQL'），若使用字符类型的分布列，不建议使用不区分大小写的字符序（如：utf8mb4_general_ci、utf8mb4_unicode_ci、utf8mb4_0900_ai_ci），容易造成数据倾斜，无法平均分布，推荐使用utf8mb4_bin字符序。

- 仅当表属性enable_update_distkey设置为on时，满足以下约束条件支持更新（UPDATE）分布列的操作，否则分布列不支持更新（UPDATE）操作。

  说明

  - 仅当相应表属性enable_update_distkey设置为on或时才支持更新分布列。
  - 不支持将UPDATE语句下推DN执行，直接生成PGXC计划，不会根据更新分布列前后值而对计划做改变。
  - 不支持带有行级UPDATE TRIGGER的表，否则会执行失败，报错进行提示。
  - 对行级INSERT/DELETE TRIGGER不生效，STATEMENT级UPDATE TRIGGER正常执行。
  - 不支持并发更新同一行，先获取锁的执行，DN上后获取锁的行为按照GUC参数concurrent_dml_mode设置情况存在不同的行为（返回0或报错）。如果报错，则可能存以下情况：
    - 报错提示信息为update distribute column conflict。
    - 报错提示信息为锁超时，即当获取锁时间超过阈值。
  - 不支持带有全局二级索引（GSI）的表，否则会执行报错。
  - 只支持HASH分布，不支持LIST分布或RANGE分布，否则会执行报错。
  - 不支持MERGE INTO和UPSERT更新分布列的行为，否则会执行报错。
  - 不支持gtm_free，否则会执行报错。
  - 不支持UPDATE RETURNING，否则会执行报错。
  - 不支持JOIN，否则会执行报错。
  - 不支持UPDATE LIMIT，否则会执行报错。
- 如果在建表过程中数据库系统发生故障，系统恢复后可能无法自动清除已创建的磁盘文件和大小为0的磁盘文件。

版权所有 © 华为云计算技术有限公司

- 使用JDBC时，支持通过PreparedStatement对DEFAULT值进行参数化设置。
- 行存表的表级约束不支持外键。
- 不支持在rowid/rowno系统列上创建外键约束。
- 被授予CREATE ANY TABLE权限的用户，可以在public模式和用户模式下创建表。如果需要创建包含serial类型列的表，还需要CREATE ANY SEQUENCE创建序列的权限。
- XML类型数据不能作为主键和外键。
- 在为数据对象增加或者变更ILM策略时，如果追加了行表达式，仅支持白名单中列出的函数作为行表达式。具体请参见**行表达式函数白名单**。

须知

- 如果数据库无限创建表，可能会对CN造成以下影响：
  - 资源耗尽：每个表都会占用一定的磁盘空间，无限创建表会导致大量的内存和磁盘空间被占用，可能会导致CN的资源耗尽，从而导致系统崩溃或变得不稳定。
    - 性能下降：无限创建表会导致大量的I/O操作和CPU计算，数据库的元数据信息将会变得十分庞大，可能会导致CN的性能下降，包括插入、查询、更新和删除等操作，从而导致系统响应变慢或无法满足业务需求。
    - 安全问题：过多的表会导致数据库的管理和维护变得困难，无限创建表可能会导致数据泄露或数据丢失等安全问题，数据库的稳定性会降低从而给企业带来不可估量的损失。

  因此，对于数据库，应该合理规划表的数量和大小，避免无限创建表，从而保证系统的稳定性、可靠性和安全性。
- 在ILM策略的ON（EXPR）行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。

- 表约束个数不能超过32767个。

语法格式

创建表。

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [ IF NOT EXISTS ] table_name
    { ( { column_name data_type [ CHARACTER SET | CHARSET charset ] [ COLLATE collation ]
[ column_constraint [ ... ] ]
        | table_constraint
        | LIKE source_table [ like_option [...] ] }
        [, ... ] )
    | LIKE source_table }
    [ table_option [ ,... ] [ ... ] ]
    [ htap_option ]
    [ WITH ( {storage_parameter = value} [, ... ] ) ]
    [ WITHOUT OIDS ]
    [ ON COMMIT { PRESERVE ROWS | DELETE ROWS } ]
    [ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | days | month | months | year | years } OF NO
MODIFICATION [ ON ( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour| hours | day | days | month | months | year |
years } OF NO MODIFICATION
 | NONE}]
```


版权所有 © 华为云计算技术有限公司

```
  [ TABLESPACE tablespace_name ]
  [ DISTRIBUTE BY { REPLICATION | HASH ( column_name [, ...] )
  | MURMURHASH ( expression )
  | RANGE ( column_name [, ...] ) { SLICE REFERENCES tablename | ( slice_less_than_item [, ...] )
  | ( slice_start_end_item [, ...] ) }
  | LIST ( column_name [, ...] ) { SLICE REFERENCES tablename | ( slice_values_item [, ...] ) }
  } ]
  [ TO { GROUP groupname | NODE ( nodename [, ... ] ) } ];
```

- 其中table_option为：

  ```
  { COMMENT [ = ] 'string' |
    AUTO_INCREMENT [ = ] value |
    [ DEFAULT ] CHARACTER SET | CHARSET [ = ] default_charset |
    [ DEFAULT ] COLLATE [ = ] default_collation |
    ENGINE [ = ] { InnoDB | 'InnoDB' | "InnoDB" } }
  ```

- 其中htap_option为：

  ```
  { COLVIEW [ PRIORITY { HIGH | LOW | NONE } ] |
    NOCOLVIEW }
  ```

- 其中列约束column_constraint为：

  ```
  [ CONSTRAINT constraint_name ]
  { NOT NULL |
    NULL |
    CHECK ( expression ) |
    DEFAULT default_expr |
    ON UPDATE update_expr |
    AUTO_INCREMENT |
    COMMENT 'string' |
    COLVIEW |
    NOCOLVIEW |
    UNIQUE [KEY] index_parameters |
    PRIMARY KEY index_parameters |
    ENCRYPTED WITH ( [DATATYPE_CL = client_logic_type,]COLUMN_ENCRYPTION_KEY =
  column_encryption_key, ENCRYPTION_TYPE = encryption_type_value ) |
    REFERENCES reftable [ ( refcolumn ) ] [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]
      [ ON DELETE action ] [ ON UPDATE action ] }
  [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
  ```

- 其中表约束table_constraint为：

  ```
  [ CONSTRAINT [ constraint_name ] ]
  { CHECK ( expression ) |
    UNIQUE [ index_name ] [ USING method ] ( { { column_name [ ( length ) ] | ( expression ) } [ ASC |
  DESC ] }[, ... ] ) index_parameters |
    PRIMARY KEY [ USING method ] ( { column_name [ ASC | DESC ] } [, ... ] ) index_parameters
  [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
  { [ COMMENT 'string' ] [ ... ] }
  ```

- 其中like选项like_option为：

  ```
  { INCLUDING | EXCLUDING } { DEFAULTS | CONSTRAINTS | INDEXES | STORAGE | COMMENTS |
  PARTITION | RELOPTIONS | DISTRIBUTION | UPDATE | ILM | ALL }
  ```

- 其中索引参数index_parameters为：

  ```
  [ WITH ( {storage_parameter = value} [, ... ] ) ]
  [ USING INDEX TABLESPACE tablespace_name ]
  ```

- RANGE分布规则如下：
  - slice_less_than_item为：

    ```
    SLICE name VALUES LESS THAN ({ expression | MAXVALUE } [, ...]) [ DATANODE
    datanode_name | ( datanode_name_list [, ... ] ) ]
    ```

  - slice_start_end_item为：

    ```
    SLICE name {
    { START ( expression ) END ( expression ) EVERY ( expression ) } |
    { START ( literal ) END ( { literal | MAXVALUE } ) } |
    { START ( literal ) } |
    { END ( { literal | MAXVALUE } ) }
    }
    ```

- 其中LIST分布规则如下：

  slice_values_item为：

版权所有 © 华为云计算技术有限公司

```
[ ( SLICE name VALUES (expression [, ... ]) [DATANODE datanode_name | ( datanode_name_list )]
 [, ... ] ) |
 ( SLICE name VALUES (DEFAULT) [DATANODE datanode_name] | ( datanode_name_list ) )
]
```

- 其中update_expr为：

```
{ CURRENT_TIMESTAMP | LOCALTIMESTAMP | NOW() }
```

参数说明

- **UNLOGGED**

  如果指定此关键字，则创建的表为非日志表。在非日志表中写入的数据不会被写入到预写日志中，这样就会比普通表快很多。但是非日志表在冲突、执行操作系统重启、数据库重启、主备切换、切断电源操作或异常关机后会被全部清空，会造成数据丢失的风险。非日志表中的内容也不会被复制到备服务器中。在非日志表中创建的索引也不会被自动记录。

  使用场景：非日志表不能保证数据的安全性，用户应该在确保数据已经做好备份的前提下使用，例如系统升级时进行数据的备份。

  故障处理：当异常关机等操作导致非日志表上的索引发生数据丢失时，用户应该对发生错误的索引进行重建。

- **GLOBAL | LOCAL**

  创建临时表时可以在TEMP或TEMPORARY前指定GLOBAL或LOCAL关键字。如果指定GLOBAL关键字，GaussDB会创建全局临时表，否则创建本地临时表。

- **TEMPORARY | TEMP**

  如果指定TEMP或TEMPORARY关键字，则创建的表为临时表。临时表分为全局临时表和本地临时表两种类型。创建临时表时如果指定GLOBAL关键字则为全局临时表，否则为本地临时表。

  全局临时表的元数据对所有会话可见，会话结束后元数据继续存在。会话与会话之间的用户数据、索引和统计信息相互隔离，每个会话只能看到或更改自己提交的数据。全局临时表有两种模式：一种基于会话级别（ON COMMIT PRESERVE ROWS），当会话结束时自动清空用户数据；一种基于事务级别（ON COMMIT DELETE ROWS），当执行COMMIT或ROLLBACK时自动清空用户数据。建表时如果没有指定ON COMMIT选项，则缺省为会话级别。与本地临时表不能指定schema不同，全局临时表建表时可以指定schema。

  本地临时表只在当前会话可见，本会话结束后自动删除。因此，在除当前会话连接的CN以外的其他CN故障时，仍然可以在当前会话上创建和使用临时表。由于临时表只在当前会话创建，对于涉及对临时表操作的DDL语句，会产生DDL失败的报错。因此，建议DDL语句中不要对临时表进行操作。TEMP和TEMPORARY等价。

版权所有 © 华为云计算技术有限公司

须知

- 本地临时表通过每个会话中以pg_temp开头的Schema来保证只对当前会话可见，因此，用户在日常操作中请勿手动删除以pg_temp，pg_toast_temp开头的Schema。
- 如果建表时不指定TEMPORARY | TEMP关键字，而指定表的Schema为当前会话的pg_temp开头的Schema，则该表会被创建为临时表。
- ALTER/DROP全局临时表和索引时，如果其它会话正在使用，则禁止操作（ALTER INDEX index_name REBUILD除外）。
- 全局临时表的DDL只会影响当前会话的用户数据和索引。例如TRUNCATE、REINDEX、ANALYZE只对当前会话有效。
- 全局临时表功能可以通过设置GUC参数max_active_global_temporary_table控制是否启用。如果max_active_global_temporary_table=0，则关闭全局临时表功能。
- 临时表只对当前会话可见，因此不支持与\parallel on（并行执行）同时使用。
- 临时表不支持DN故障或者主备切换。
- 全局临时表不响应自动清理，在长连接场景使用时尽量使用ON COMMIT DELETE ROWS的全局临时表，或定期手动执行VACUUM，否则可能导致clog日志不回收（可以通过设置vacuum_gtt_defer_check_age，对长时间未清理全局临时表的情况告警）。
- 全局临时表不支持以下场景：
  - 不支持创建全局临时SEQUENCE，各个会话的全局临时表使用共享的SEQUENCE，只能保证唯一性，不保证连续性。
  - 不支持创建全局临时视图。
  - 不支持创建分区表。
  - 不支持创建hashbucket表。
  - 不支持扩展统计信息。
  - 不支持ON COMMIT DROP属性。
  - 不支持GSI索引。
  - 不支持‘M’兼容性。
  - 不支持分布式备机读。
  - 不支持stream计划。
  - 不支持扩缩容，扩缩容回滚存在数据丢失风险，扩缩容结束数据清空。
  - 在DN主备切换、DN节点故障、CN故障的情况下，存在数据丢失的风险（全局临时表无Xlog，不做主备数据同步）。

- **IF NOT EXISTS**

  如果已经存在相同名称的表，不会产生错误，而会发出通知，告知该表已存在。

- **[ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [MEDIUM | HIGH] ROW AFTER n { day | days | month | months | year | years } OF NO MODIFICATION [ON (EXPR)] | TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | hours | day | days | month | months | year | years } OF NO MODIFICATION | NONE}]**

  创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW给行存添加高级压缩中/高级别策略。比如：

版权所有 © 华为云计算技术有限公司

```
CREATE TABLE t1 (a int) ILM ADD POLICY ROW STORE COMPRESS ADVANCED ROW AFTER 3 DAY OF NO MODIFICATION ON (a != 0);
```

表示创建t1表并添加高级压缩策略：三天未修改并且a != 0的行。

- AFTER n { day | month | year } OF NO MODIFICATION ： 表示n天/月/年没有修改的行，高级压缩不支持hour。
- ON（EXPR）：行级表达式，用于判断行的冷热。
- EXPR仅支持白名单函数，具体请参见**行表达式函数白名单**。

  说明

  - 在ILM策略的ON(EXPR)行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转换大写。
  - 压缩级别是可选项，不指定时默认使用MEDIUM级别。

创建新表时，可以调用ILM ADD POLICY给行存添加跨页透明压缩策略。比如：

```
CREATE TABLE t1 (a int) ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT AFTER 3 HOUR OF NO MODIFICATION;
```

表示创建t1表并添加跨页透明压缩策略：策略级别为high，冷热管理时间间隔为3小时。

- AFTER n { hour | day | month | year } OF NO MODIFICATION ： 表示n小时/天/月/年没有修改的压缩组。
- 压缩级别不指定时默认为MEDIUM。

- **table_name**

  要创建的表名。

  须知

  物化视图的一些处理逻辑会通过表名的前缀来识别是不是物化视图日志表和物化视图关联表。因此，用户不要创建表名以mlog_或matviewmap_为前缀的表，否则会影响此表的一些功能。

- **column_name**

  新表中要创建的字段名。

- **data_type**

  字段的数据类型。

- **CHARACTER SET | CHARSET charset**

  指定表字段的字符集。未单独指定COLLATE时，会将字段的字符序设置为指定的字符集的默认字符序。

- **COLLATE collation**

  指定列的字符序（数据类型需为支持排列的类型）。未单独指定时，则使用默认字符序。

  可通过SELECT * FROM pg_collation;命令查看pg_collation系统表中显示的字符序信息，默认的字符序为查询结果中以default开始的行。

版权所有 © 华为云计算技术有限公司

说明

- 仅字符类型支持指定字符集。
- 指定为binary字符集或字符序时，实际是将字符类型转化为对应的二进制类型，若类型映射不存在会产生报错，当前仅支持TEXT类型转化为BLOB的映射。
- 除指定为binary字符集或字符序外，当前仅支持指定与数据库编码相同的字符集。
- 未显式指定字段字符集或字符序时，若指定了表的默认字符集或字符序，字段字符集和字符序将从表上继承。若表的默认字符集或字符序不存在，当b_format_behavior_compat_options包含'default_collation'时，字段的字符集和字符序将继承当前数据库的字符集及其对应的默认字符序。
- 对于MYSQL兼容模式下（即sql_compatibility = 'MYSQL'），还支持utf8mb4_bin、utf8mb4_general_ci、utf8mb4_unicode_ci、binary、gbk_chinese_ci、gbk_bin、gb18030_chinese_ci、gb18030_bin、utf8mb4_0900_ai_ci、utf8_general_ci以及utf8_bin字符序，具体介绍如**表1-398**所示。

表 1-398 MYSQL 兼容模式下（即 sql_compatibility = 'MYSQL'）支持的字符集和字符序介绍

字符序名称	对应的字符集	描述
utf8mb4_general_ci	utf8mb4（即utf8）	使用通用排序规则，不区分大小写。
utf8mb4_unicode_ci	utf8mb4（即utf8）	使用通用排序规则，不区分大小写。
utf8mb4_bin	utf8mb4（即utf8）	使用二进制排序规则，区分大小写。
binary	binary	使用二进制排序规则。
gbk_chinese_ci	gbk	使用中文语言排序规则。
gbk_bin	gbk	使用二进制排序规则，区分大小写。
gb18030_chinese_ci	gb18030	使用中文语言排序规则。
gb18030_bin	gb18030	使用二进制排序规则，区分大小写。
utf8mb4_0900_ai_ci	utf8mb4	使用UCA (Unicode collation algorithm) 算法规则，不区分大小写。
utf8_general_ci	utf8	使用通用排序规则，不区分大小写。
utf8_bin	utf8	使用二进制排序规则，区分大小写。

- **LIKE source_table [like_option ...]**

  LIKE子句声明一个表，新表自动从这个表中继承所有字段名及其数据类型和非空约束，以及声明为serial的缺省表达式。

  新表与源表之间在创建动作完毕之后是相互独立的。在源表做的任何修改都不会被继承到也不会新表中，并且在扫描源表的时候，不包含新表的数据。

版权所有 © 华为云计算技术有限公司

被继承的列和约束并不使用相同的名称进行融合。如果明确地指定了相同的名称或者在另外一个LIKE子句中，将会报错。

- 源表上除serial外的字段缺省表达式只有在指定INCLUDING DEFAULTS时，才会被继承到新表中。默认情况下，不包含缺省表达式，即新表中的所有字段的缺省值都是NULL。
- 如果指定了INCLUDING UPDATE，则源表列的ON UPDATE CURRENT_TIMESTAMP属性会被继承到新表列中。默认不复制该属性。
- 源表上的CHECK约束仅在指定INCLUDING CONSTRAINTS时，会被继承到新表中，而其他类型的约束将不会被继承到新表中。非空约束一直可以继承到新表中。此规则同时适用于表约束和列约束。
- 如果指定了INCLUDING INDEXES，则源表上的索引也将在新表上创建，默认情况下，不建立索引。新表上的索引所在表空间和源表上的索引所在表空间保持一致。
- 如果指定了INCLUDING STORAGE，则源表列的STORAGE设置会被继承到新表中，默认情况下，不包含STORAGE设置。
- 如果指定了INCLUDING COMMENTS，则源表列、约束和索引的注释会复制到新表中。默认情况下，不对源表的注释进行继承。
- 如果指定了INCLUDING PARTITION，则源表的分区定义会被继承到新表中，同时新表将不能再使用PARTITION BY子句。默认情况下，不对源表的分区定义进行继承。
- 如果指定了INCLUDING RELOPTIONS，则源表的存储参数（即源表的WITH子句）将会被继承到新表中。默认情况下，不对源表的存储参数进行继承。
- 如果指定了INCLUDING DISTRIBUTION，则源表的分布信息会被继承到新表中，包括分布类型和分布列，同时新表将不能再使用DISTRIBUTE BY子句。默认情况下，不对源表的分布信息进行继承。
- 如果指定了INCLUDING ILM，则源表的ILM策略信息会被继承到新表中，如果需要同时复制源表上的分区对象的ILM策略信息，需要同时指定INCLUDING PARTITION。
- INCLUDING ALL包含了INCLUDING DEFAULTS、INCLUDING UPDATE、INCLUDING CONSTRAINTS、INCLUDING INDEXES、INCLUDING STORAGE、INCLUDING COMMENTS、INCLUDING PARTITION、INCLUDING RELOPTIONS、INCLUDING DISTRIBUTION和INCLUDING ILM的内容。

版权所有 © 华为云计算技术有限公司

须知

- 如果指定了INCLUDING RELOPTIONS，且源表的存储参数（即源表的WITH子句）中不包含storage_type，则新表的存储引擎类型将由GUC参数enable_default_ustore_table的取值决定。
- 如果源表包含serial、bigserial、smallserial类型，或者源表字段的默认值是SEQUENCE，且SEQUENCE属于源表（通过CREATE SEQUENCE ... OWNED BY创建），这些SEQUENCE不会关联到新表中，新表中会重新创建SEQUENCE。如果用户希望源表和新表共享SEQUENCE，需要首先创建一个共享的SEQUENCE（避免使用OWNED BY），并配置为源表字段默认值，这样创建的新表会和源表共享该SEQUENCE。
- 不建议将其他表私有的SEQUENCE配置为源表字段的默认值，尤其是其他表只分布在特定的NodeGroup上，这可能导致CREATE TABLE ... LIKE执行失败。另外，如果源表配置其他表私有的SEQUENCE，当该表删除时SEQUENCE也会连带删除，这样源表的SEQUENCE将不可用。如果用户希望多个表共享SEQUENCE，建议创建共享的SEQUENCE。
- 如果源表为rowid表，指定了INCLUDING ALL/INCLUDING RELOPTIONS后，新表也为rowid表；仅指定INCLUDING INDEXES时，新表为非rowid表，且不会复制源表的rowno索引。
- CREATE TABLE table_name LIKE source_table;语法仅在MYSQL模式数据库（即sql_compatibility = 'MYSQL'）下，且参数b_format_version值为5.7、b_format_dev_version值为s2时支持。
- 在MYSQL模式数据库下，且参数b_format_version值为5.7、b_format_dev_version值为s2时，不支持指定INCLUDING和EXCLUDING选项，默认情况下与指定INCLUDING ALL的行为一致。
- 若source_table为系统表，则不支持指定INCLUDING ALL或INCLUDING DISTRIBUTION。

- **AUTO_INCREMENT [=] value**

  这个子句为自动增长列指定一个初始值，value必须为正数，不得超过2^{127}-1。

须知

该子句仅在参数sql_compatibility='MYSQL'时有效。

- **COMMENT [=] 'string'**
  - COMMENT [=] 'string'子句表示给表添加注释。
  - 在column_constraint中的COMMENT 'string'表示给列添加注释。
  - 在table_constraint中的COMMENT 'string'表示给主键和唯一键对应的索引添加注释。

版权所有 © 华为云计算技术有限公司

须知

- 该子句仅在MYSQL模式数据库（即sql_compatibility = 'MYSQL'）下有效。
- 表级注释支持的最大字符串长度为2048字符，列级和索引级注释支持的最大长度为1024字符。
- table_constraint中的COMMENT仅支持主键和唯一键，其他约束不支持。

- **ENGINE**

  MYSQL兼容模式下支持，仅语法适配，且只支持设置InnoDB，无实际效果。

须知

不支持CREATE TABLE table_name LIKE source_table;语法中使用ENGINE语法。

- **COLVIEW | NOCOLVIEW**

  指定表、列、分区是否创建IMCV表元信息。其中创建IMCV表需要依赖于enable_htap参数打开。

  不支持与LIKE source_table语法同时使用。

- **PRIORITY**

  指定数据加载至IMCV表的优先级。分为：默认优先级、低优先级以及高优先级。

- **WITH ({ storage_parameter = value } [, ...])**

  表示为表或为索引指定一个可选的存储参数。当WITH子句用于表时，还可以包含OIDS=FALSE，表示不分配OID。

须知

- WITH子句不支持OIDS=TRUE语法。
- 使用任意精度类型Numeric定义列时，建议指定精度p以及刻度s。在不指定精度p和刻度s时，按输入格式直接输出。

参数的详细描述如下所示：

  - FILLFACTOR

    一个表的填充因子（fillfactor）是一个介于10和100之间的百分数。100（完全填充）是默认值。在Ustore存储引擎下，该值的默认值为92。如果指定了较小的填充因子，INSERT操作仅按照填充因子指定的百分率填充表页。每个页上的剩余空间将用于在该页上更新行，这就使得UPDATE有机会在同一页上放置同一条记录的新版本，这比把新版本放置在其他页上更有效。对于一个从不更新的表将填充因子设为100是最佳选择，但是对于频繁更新的表，选择较小的填充因子则更加合适。

    取值范围：10~100

  - ORIENTATION

    指定表数据的存储方式，该参数设置成功后就不再支持修改。

    取值范围：

版权所有 © 华为云计算技术有限公司

- ROW，表示表的数据将以行式存储。

  行存储适合于OLTP业务，此类型的表上交互事务比较多，一次交互会涉及表中的多个列，用行存查询效率较高。

默认值：若指定表空间为普通表空间，默认值为ROW。

– STORAGE_TYPE

指定存储引擎类型，该参数设置成功后就不再支持修改。

取值范围：

- USTORE，表示表支持Inplace-Update存储引擎。特别需要注意，使用UStore表，必须要开启track_counts和track_activities参数，否则会引起空间膨胀。
- ASTORE，表示表支持Append-Only存储引擎。

默认值：不指定时，由参数enable_default_ustore_table决定存储引擎方式，默认是Inplace-Update存储。

– TOAST.TOAST_STORAGE_TYPE

指定Toast表的存储类型。当GUC参数“enable_enhance_toast_table=on”时，Toast表的存储类型支持通过Vacuum Full操作将Toast升级为Enhanced Toast结构。

取值范围：

- ENHANCED_TOAST，指定Toast表的存储类型为Enhanced Toast。
- TOAST，指定Toast表的存储类型为Toast。

默认值：

建表时不指定TOAST.TOAST_STORAGE_TYPE，则Toast表的存储类型依赖于GUC参数enable_enhance_toast_table。若该参数值为on，则Toast表的存储类型为ENHANCED_TOAST；若该参数值为off，则Toast表的存储类型为TOAST。

– INIT_TD

创建UStore表时，指定初始化的TD个数，该参数可以通过ALTER TABLE进行修改。特别需要注意，该参数会影响数据页面存放的单个元组的最大大小，具体换算方法为MAX_TUPLE_SIZE = BLCKSZ - INIT_TD * TD_SIZE，例如用户将INIT_TD数量从4修改为8，单个元组最大大小会减小4 * INIT_TD大小。

取值范围：2~128，默认值为4。

– COMPRESSLEVEL

指定表数据同一压缩级别下的不同压缩水平，它决定了同一压缩级别下表数据的压缩比以及压缩时间。对同一压缩级别进行了更加详细的划分，为用户选择压缩比和压缩时间提供了更多的空间。总体来讲，此值越大，表示同一压缩级别下压缩比越大，压缩时间越长；反之亦然。

取值范围：0~3

默认值：0

– segment

使用段页式的方式存储。本参数仅支持行存表。不支持1~5号物理文件非法删除破坏场景的防护，不支持在升级观察期创建。

取值范围：on/off

默认值：off

版权所有 © 华为云计算技术有限公司

- hashbucket

  创建hash bucket存储。本参数仅支持行存表。

  取值范围：on/off

  默认值：off

  须知

  - 创建hashbucket表时当前数据库必须绑定到集群默认的node group上。数据库绑定node group请参见**ALTER DATABASE**。
  - 当前版本hashbucket表相关DDL操作性能受限，不建议频繁对hashbucket表进行DDL操作。
  - hashbucket表绑定段页式存储，即hashbucket=on隐含segment=on。
  - 小表（小于：普通表8M，分区表8M * 分区数量）不建议创建成hashbucket表，顺序扫描性能较普通表劣化2倍。

- enable_tde

  指定该表为加密表。数据库会自动将加密表中的数据先加密再存储。使用该参数前，请确保已通过GUC参数enable_tde开启透明加密功能，并通过GUC参数tde_key_info设置访问密钥服务的信息，在《开发指南》中“特性使用指导 > 透明数据加密”章节可获取该参数的详细使用方法。本参数仅支持行存表、段页式表、hashbucket表、临时表和unlogged表。

  取值范围：on/off。设置enable_tde=on时，key_type、cmk_id、dek_cipher参数由数据库自动生成，用户无法手动指定或更改。

  默认值：off

- encrypt_algo

  指定加密表的加密算法，需与enable_tde结合使用。

  取值范围：字符串，有效值为：AES_128_CTR、SM4_CTR、AES_128_CTR_SHA256_HMAC、SM4_CTR_SM3_HMAC或AES_128_GCM。

  默认值：

  - 如果enable_tde=on：
    - GUC参数tde_encrypt_config中的子参数table_algorithm不为空，默认值与table_algorithm相同。
    - GUC参数tde_encrypt_config中的子参数table_algorithm为空，默认值为AES_128_CTR。
  - 如果enable_tde=off，默认值为空。

- parallel_workers

  表示创建索引时，启动的bgworker线程数量，例如2就表示将会起2个bgworker线程并发创建索引。

  取值范围：[0,32]，int类型，0表示关闭并行建索引。

  默认值：不设置该参数，表示不开启并行建索引功能。

- dek_cipher

  数据密钥的密文。该存储参数不支持用户手动设置或更改。当用户为表设置enable_tde参数后，数据库自动生成数据密钥。

版权所有 © 华为云计算技术有限公司

取值范围：字符串

默认值：空

- key_type

  主密钥的类型。该存储参数不支持用户手动设置或更改。当用户为表设置enable_tde参数后，数据库自动从GUC参数tde_key_info中获取主密钥的类型。

  取值范围：字符串

  默认值：空

- cmk_id

  主密钥的ID。该存储参数不支持用户手动设置或更改。当用户为表设置enable_tde参数后，数据库自动从GUC参数tde_key_info中获取主密钥的ID。

  取值范围：字符串

  默认值：空

- hasuids

  参数开启：更新表元组时，为元组分配表级唯一标识id。

  取值范围：on/off

  默认值：off

- logical_repl_node

  复制表逻辑解码时，向CN返回逻辑日志的DN节点名。对于复制表，如用户不指定，则默认为当前表所在node group的第一个节点。对该选项进行RESET操作时，会重置为当前表的第一个节点。

  取值范围：字符串

  默认值：非复制表默认为空，复制表默认为第一个节点名。

- skip_logical_deparse

  当数据库进行处理逻辑解码时，跳过对该表的处理。不建议用户使用，若使用不当，会造成逻辑解码缺失内容。

  取值范围：on/off

  默认值：off

- STAT_STATE

  该表的统计信息是否被锁定的标识符，如果被锁定了，该表的统计信息无法更新。当前形态下该参数不生效。

  取值范围：locked/unlock

  默认值：unlock

- statistic_granularity

  记录该表在分析统计信息时的默认partition_mode。partition_mode说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。此参数对非分区表设置无效。

  取值范围：请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。

  默认值：AUTO

- autovacuum_enabled

  表示是否开启表的自动清理功能。

  取值范围：on/off

  默认值：on

版权所有 © 华为云计算技术有限公司

- autovacuum_vacuum_threshold

  开启自动清理功能，指定在该表中触发VACUUM所需的更新或删除的最小元组数（仅对Astore表生效）。

  取值范围：0~2147483647

  默认值：缺省时与GUC参数autovacuum_vacuum_threshold一致。

- autovacuum_analyze_threshold

  自动清理功能中，指定在该表中触发ANALYZE所需的插入、更新或删除的最小元组数。

  取值范围：0~2147483647

  默认值：缺省时与GUC参数autovacuum_analyze_threshold一致。

- autovacuum_vacuum_scale_factor

  自动清理功能中，指定在该表中触发VACUUM所需的插入、更新或删除元组的比例（仅对Astore表生效）。

  取值范围：0.0~100.0

  默认值：缺省时与GUC参数autovacuum_vacuum_scale_factor一致。

- autovacuum_analyze_scale_factor

  自动清理功能中，指定在该表中触发ANALYZE所需的插入、更新或删除元组的比例。

  取值范围：0.0~100.0

  默认值：缺省时与GUC参数autovacuum_analyze_scale_factor一致。

- autovacuum_freeze_min_age

  自动清理功能中，在该表参数指定了一个行版本的最小年龄，超过这个年龄的行才会被冻结。

  取值范围：0~576460752303423487

  默认值：缺省时与GUC参数vacuum_freeze_min_age一致。

- autovacuum_freeze_max_age

  自动清理功能中，该表pg_class.relfrozenxid字段在超过多少个事务后，就会强制执行VACUUM操作。即使自动清理被禁用，系统也会启动AUTOVACUUM进程。清理操作还允许从pg_clog/子目录中删除旧文件（仅对Astore表生效）。

  取值范围：100000~576460752303423487

  默认值：缺省时与GUC参数autovacuum_freeze_max_age一致。

- autovacuum_freeze_table_age

  自动清理功能中，该表被标记为不需要自动清理时，将保持不变的时间。（仅对Astore表生效）。

  取值范围：0~576460752303423487

  默认值：缺省时与GUC参数vacuum_freeze_table_age一致。

- vacuum_truncate

  表示是否开启表的VACUUM TRUNCATE功能，在VACUUM时截断末尾空页。

  取值范围：on/off

  默认值：on

版权所有 © 华为云计算技术有限公司

说明

- 该参数仅对Astore生效，对hashbucket表和segment表不生效。
- 从不支持该参数版本升级至支持该参数版本时，不支持在升级观察期创建、修改该参数。
- 支持通过toast.vacuum_truncate参数单独控制TOAST表是否截断末尾空页。

- skip_logical_deparse

  在内核处理逻辑解码时跳过对该表的处理，用于内核处理过程，不建议用户使用。若使用不当，会造成逻辑解码缺失内容。

  取值范围：on/off

  默认值：off

- enable_update_distkey

  标记该表是否支持更新（UPDATE）分布列的操作。仅当表的该属性设置为on，且在约束条件下时，才支持更新（UPDATE）分布列的操作。否则，分布列不支持更新（UPDATE）操作。

  取值范围：on/off

  默认值：off

  说明

  支持更新（UPDATE）分布列操作的约束请参见**注意事项**。

- hasrowid

  指定该表是否为rowid表。设置hasrowid = on时，该表会增加rowid系统列和rowno系统列，且会在rowno系统列上关联创建序列和索引。

  取值范围：on、off

  默认值：off

  说明

  - 此参数只支持ORA兼容模式数据库（即sql_compatibility = 'ORA'）。
  - 此参数不支持本地临时表、Unlogged表、全局临时表、物化视图、hashbucket表、复制表。
  - 此参数不支持和hasuids参数、oids参数同时指定。
  - 开启此参数时，不支持定义普通列名为rowno、rowid。
  - rowid表数据量不能超过9223372036854775807。
  - rowid表在GUC参数opt_behavior_knob开启ROWID_OPT选项时，支持ROWID系统列谓词条件改写、order by语句改写、max/min优化、分布剪枝和分区剪枝。详情请参见**opt_behavior_knob**内容。

- partition_statistics_min_rows

  对分区表执行ANALYZE命令时，用于确定分区级最小采样行数的参数值。

  取值范围：[0, 10000]

  默认值：50

版权所有 © 华为云计算技术有限公司

说明

- 此参数仅对分区表有效。
- 此参数仅在ANALYZE命令的PARTITION_MODE = ALL时，且GUC参数enable_partition_statistics_min_rows启用时生效，PARTITION_MODE说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。
- 上限约束：最终的分区表整表采样行数不超过GUC参数maintenance_work_mem可容纳的行数。

- **WITHOUT OIDS**

  等价于WITH（OIDS=FALSE）的语法，表示不分配OID。

- **ON COMMIT { PRESERVE ROWS | DELETE ROWS }**

  ON COMMIT选项决定在事务中执行创建临时表操作，当事务提交时，此临时表的后续操作。当前支持PRESERVE ROWS和DELETE ROWS选项。

  - PRESERVE ROWS（缺省值）：提交时不对临时表做任何操作，临时表及其表数据保持不变。
  - DELETE ROWS：提交时删除临时表中数据。

- **TABLESPACE tablespace_name**

  创建新表时指定此关键字，表示新表将要在指定表空间tablespace_name内创建。如果没有声明，将使用默认表空间。

- **DISTRIBUTE BY**

  指定表如何在节点之间分布或者复制。

  取值范围：

  - REPLICATION：表的每一行存在所有数据节点（DN）中，即每个数据节点都有完整的表数据。
  - HASH（column_name）：对指定的列进行Hash操作，通过映射，将数据分布到对应DN。
  - MURMURHASH（diskey_expr）：对指定的列进行Hash操作，Hash算法指定为MURMURHASH算法，通过映射，将数据分布到对应DN上。diskey_expr可以是单个列名，也可以是**lpad_s(string text, length int [, fill text])**函数修饰的列名，使用该函数进行修饰时，只有第一个参数可以为列名，形如lpad_s(a,10,'0')。lpad_s作为分布列表达式时不能省略第三个参数。
  - RANGE(column_name) 对指定列按照范围进行映射，把数据分布到对应DN。
  - LIST(column_name) 对指定列按照具体值进行映射，把数据分布到对应DN。

版权所有 © 华为云计算技术有限公司

说明

- 对于HASH分布，分布键最多个数与列最多个数一致，最多支持1600个；对于MURMURHASH分布，分布键只支持一列（一个表达式或一个表中的列）；对于RANGE（VALUE LESS THAN）分布和LIST分布，分布键最多支持四列；对于RANGE（START END）分布，分布键只支持一列。
- 只有MURMURHASH分布支持分布列表达式，分布列表达式中只支持**lpad_s(string text, length int [, fill text])**函数。
- 对于MURMURHASH分布，有以下约束：
  - 不支持stream计划，支持FQS计划和PGXC计划。
  - 不支持扩缩容、hashbucket表、GSI及物化视图。
  - 两个HASH表做关联，在分布键上有等值关联条件时，目前暂不支持FQS计划。
  - MURMURHASH分布表必须指定nodegroup（自定义的nodegroup）且该nodegroup为非installation的nodegroup建表。
  - 不支持通过CREATE TABLE AS方式建表。
  - MURMURHASH分布表的分布键必须含有NOT NULL约束。
- MURMURHASH分布是为了将采用MURMURHASH算法的遗留系统迁移时，保证数据分布不变而使用的分布算法。暂不支持扩容和stream计划，在选用分布算法时，建议优先选择HASH分布算法。
- 对于从句是VALUE LESS THAN语法格式的RANGE分布策略，分布规则如下：
  - 从插入值的第一列开始比较。
  - 如果插入值的第一列小于待插入的分片的当前列的边界值，则直接插入。
  - 如果插入值的第一列等于待插入的分片的当前列 的边界值，则比较插入值的下一列与待插入的分片的下一列的边界值，如果小于，则直接插入。如果相等，继续比较插入值的下一列与待插入的分片的下一列的边界值，直至小于并插入。
  - 如果插入值的所有列大于待插入的分片的所有列的边界值，则比较下一分片。
- 如果RANGE分布表的分片对应多个DN，会计算分布键的hash值，之后在这些DN中用hash值对DN个数取模，重新映射一个DN。详情见示例。
- 如果LIST分布表的分片对应多个DN，对于default分片，会计算分布键的hash值，之后在这些DN中用hash值对DN个数取模，重新映射一个DN。对于非default分片，会使用Round Robin方式，把values列表中的值和DN进行映射。详情见示例。
- RANGE/LIST分布表只支持扩容不支持缩容，slice扩容规则请联系管理员。
- 对于复制表，存在以下下推充要条件：
  - 窗口函数场景：根据不同的窗口函数类型，有不同的下推充要条件：
    - 对于rank系列函数以及ratio_to_report函数：无限制条件，均支持下推。
    - 因为对于相同的PARTITION BY和ORDER BY值，窗口函数计算结果相同，并不依赖数据的相对顺序。
    - 对于row_number函数以及ntile函数，满足以下任一条件即可下推：
      - 投影列中仅包含窗口函数投影列。
      - 投影列中仅包含OVER子句中PARTITION BY的字段或ORDER BY的字段（无其他字段）+窗口函数投影列。
      - 当OVER子句仅存在PARTITION BY或ORDER BY时，PARTITION BY或ORDER BY对应列为主键。
      - 当OVER子句同时存在PARTITION BY和ORDER BY时，PARTITION BY和ORDER BY对应列均为主键。
    - 因为当窗口函数PARTITION BY或ORDER BY的列值相同但其他列值不同时，若其他列出现在窗口函数投影列中，可能导致不同数据节点（DN）上数据顺序不一致，引发窗口函数结果差异和数据不一致问题。
    - 对于lag函数以及lead函数：offset参数为0时支持下推。

版权所有 © 华为云计算技术有限公司

- 因为当offset参数为0时，相当于查询某一列，不依赖数据相对顺序；如果offset大于0时，由于相对顺序不同问题，会导致具有相同值的不同元组获取到NULL值。
- 对于first_value函数、last_value函数以及nth_value函数，满足条件之一即可下推：
  - 参数表达式中的字段必须是OVER子句中PARTITION BY或ORDER BY的字段。
  - 当OVER子句仅存在PARTITION BY或ORDER BY时，PARTITION BY或ORDER BY对应列为主键。
  - 当OVER子句同时存在PARTITION BY和ORDER BY时，PARTITION BY和ORDER BY对应列均为主键。
- 对复制表的查询包含系统列及volatile函数时不进行下推。
- 拼接聚集函数场景：下推充要条件为拼接列等于ORDER BY列或者ORDER BY的字段为主键。
- UPDATE/DELETE语句使用WHERE CURRENT OF cursor_name语法时，不进行下推。

– 对于HASH分布，column_name的数据类型如下：
  - INTEGER TYPES：TINYINT、SMALLINT、INT、BIGINT、NUMERIC/DECIMAL
  - CHARACTER TYPES：CHAR、BPCHAR、VARCHAR、VARCHAR2、NVARCHAR2、TEXT
  - DATE/TIME TYPES：DATE、DATEA（仅在ORA兼容性生效）、TIME、TIMETZ、TIMESTAMP、TIMESTAMPTZ、INTERVAL、SMALLDATETIME
– 对于MURMURHASH分布，diskey_expr中的列名的数据类型如下：
  - INTEGER TYPES：TINYINT、SMALLINT、INT、BIGINT
  - CHARACTER TYPES：CHAR、BPCHAR、VARCHAR、VARCHAR2、NVARCHAR2
– 对于RANGE（VALUES LESS THAN）/LIST分布，column_name的数据类型如下：
  - INTEGER TYPES：TINYINT、SMALLINT、INT、BIGINT、NUMERIC/DECIMAL
  - CHARACTER TYPES：CHAR、BPCHAR、VARCHAR、VARCHAR2、NVARCHAR2、TEXT
  - DATE/TIME TYPES：DATE、DATEA（仅在ORA兼容性生效）、TIMESTAMP、TIMESTAMPTZ
– 对于RANGE（START END）分布，column_name的数据类型如下：
  - INTEGER TYPES：TINYINT、SMALLINT、INT、BIGINT、NUMERIC/DECIMAL
  - DATE/TIME TYPES：DATE、DATEA（仅在ORA兼容性生效）、TIMESTAMP、TIMESTAMPTZ

版权所有 © 华为云计算技术有限公司

说明

在建表时，选择分布列和分区键可对SQL查询性能产生重大影响。因此，需要根据一定策略选择合适的分布列和分区键。

- 选择合适的分布列：

  对于采用散列（HASH）方式的数据分布表，一个合适的分布列应将一个表内的数据，均匀分散存储在多个DN内，避免出现数据倾斜现象（即多个DN内数据分布不均）。请按照如下原则判定合适的分布列：

  1. 判断是否已发生数据倾斜现象。

     连接数据库，执行如下语句，查看各DN内元组数目。其中tablename为待分析的表名。

     ```
     gaussdb=# SELECT a.count,b.node_name FROM (SELECT count(*) AS count,xc_node_id FROM tablename GROUP BY xc_node_id) a, pgxc_node b WHERE a.xc_node_id=b.node_id ORDER BY a.count DESC;
     ```

     如果各DN内元组数目相差较大（如相差数倍、数十倍），则表明已发生数据倾斜现象，请按照2调整分布列。

  2. 重新选择分布列并重新建表。由于当前不支持通过ALTER TABLE语句调整分布列，因此调整分布列时需要重新建表。

     选择原则如下：

     分布列的列值应比较离散，以便数据能够均匀分布到各个DN。例如，考虑选择表的主键为分布列，如在人员信息表中选择身份证号码为分布列。

     在满足上面原则的情况下，考虑选择查询中的连接条件为分布列，以便JOIN任务能够下推到DN中执行，且减少DN之间的通信数据量。

- 选择合适的分区键：

  数据分区功能，可根据表的一列或者多列，将要插入表的记录分为若干个范围（这些范围在不同的分区里没有重叠），然后为每个范围创建一个分区，用来存储相应的数据。

  调整分区键，使每次查询结果尽可能存储在相同或者最少的分区内（称为“分区剪枝”），通过获取连续I/O大幅度提升查询性能。实际业务中，经常将时间作为查询对象的过滤条件。因此，可考虑选择时间列为分区键，键值范围可根据总数据量、单次查询数据量调整。

- RANGE/LIST分布：

  当没有为RANGE/LIST分布表的分片显示指定DN时，数据库内部为分片分配DN是采用roundrobin的算法。另外，在使用RANGE/LIST分布的场景中，考虑到后续扩容的需要，建议用户在建表时定义尽可能多的分片数，因为如果定义的分片数小于扩容前的DN节点数，数据重分布时则无法落入新的DN节点。需要特别注意的是，由于是由用户自行设计分片规则，在某些极端情况下，扩容也可能无法解决存储空间不足的问题。

- **TO { GROUP groupname | NODE (nodename [, ...]) }**

  TO GROUP指定创建表所在的Node Group。

  TO NODE主要供内部扩容工具使用，一般用户不应该使用。

- **CONSTRAINT [constraint_name]**

  列约束或表约束的名称。可选的约束子句用于声明约束，新行或者更新的行必须满足这些约束才能成功插入或更新。

  定义约束有两种方法：

  - 列约束：作为一个列定义的一部分，仅影响该列。不支持为AUTO_INCREMENT | COMMENT 'string' | ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = column_encryption_key, ENCRYPTION_TYPE = encryption_type_value)子句添加约束名。
  - 表约束：不和某个列绑在一起，可以作用于多个列。

版权所有 © 华为云计算技术有限公司

须知

表约束在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）constraint_name为可选项，在其他模式数据库下，必须加上constraint_name。

- **NOT NULL**

  字段值不允许为NULL。

- **NULL**

  字段值允许为NULL ，这是缺省值。

  这个子句只是为和非标准SQL数据库兼容。不建议使用。

- **CHECK (expression)**

  CHECK约束声明一个布尔表达式，每次插入或者更新的行必须使表达式结果为真或未知才能成功，否则会抛出一个异常并且不会修改数据库。

  声明为字段约束的检查约束应该只引用该字段的数值，而在表约束里出现的表达式可以引用多个字段。

  说明

  - expression表达式中，如果存在“<>NULL”或“！=NULL”，这种写法是无效的，需要修改为“IS NOT NULL”。
  - 不支持在系统列上添加CHECK约束。系统列请参见《开发指南》中“基础概念 > 系统列”章节。

- **DEFAULT default_expr**

  DEFAULT子句给字段指定缺省值。该数值可以是任何不含变量的表达式（不允许使用子查询和对本表中的其他字段的交叉引用）。缺省表达式的数据类型必须和字段类型匹配。

  缺省表达式将被用于任何未声明该字段数值的插入操作。如果没有指定缺省值则缺省值为NULL 。

- **ON UPDATE update_expr**

  ON UPDATE子句为字段的一种属性约束。

  当对表中某元组执行UPDATE操作时，

  若更新字段的新值和表中旧值不相同，则表中该元组上具有该属性且不在更新字段内的字段值，将被自动更新为当前时间戳。若更新字段的新值和表中旧值相同，则表中该元组上具有该属性且不在更新字段内的字段值不变，将保持原有值；若具有该属性的字段在更新字段内，则对应这些字段值直接按指定更新的值更新。

版权所有 © 华为云计算技术有限公司

说明

- 该属性支持在MYSQL模式数据库中的5.7版本下指定（即sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s1'）和PG模式数据库模式中指定（升级观察期间不支持该语法）。
- 语法上update_expr支持CURRENT_TIMESTAMP、LOCALTIMESTAMP、NOW()三种关键字，也支持关键字带括号指定或不指定精度。三种关键字互为同义词，属性效果相同。例如：ON UPDATE CURRENT_TIMESTAMP()、ON UPDATE CURRENT_TIMESTAMP(5)、ON UPDATE LOCALTIMESTAMP()、ON UPDATE LOCALTIMESTAMP(6)等。不带括号或空括号时精度为0，其中NOW关键字不支持不带括号。PG兼容模式下不支持ON UPDATE CURRENT_TIMESTAMP()、ON UPDATE LOCALTIMESTAMP()和ON UPDATE NOW(5)。
- 该属性仅支持在如下类型的列上指定：timestamp、datetime、date、time without time zone、smalldatetime、abstime。
- CREATE TABLE AS语法不会继承该列属性。
- CREATE TABLE LIKE语法可通过INCLUDING UPDATE或EXCLUDING UPDATE来选择继承或排除该约束，PG兼容模式下只支持EXCLUDING UPDATE选项排除该约束。新增INCLUDING ILM选项复制旧表的ilm策略信息，结合INCLUDING PARTITION选项使用可以复制旧表上分区对象的策略信息。
- 该属性指定的精度和对应列上类型指定的精度可以不一致，通过该属性更新字段值后显示结果按最小精度显示。例如：CREATE TABLE t1 (col1 timestamp(6) ON UPDATE CURRENT_TIMESTAMP(3)); 若UPDATE语法触发该属性生效，则本次更新后col1字段值小数位显示3位。
- 该属性和生成列约束不能同时指定同一列。
- 分区表中的分区键不支持指定该属性。
- 分布式场景中，分布列和主键不支持指定该属性。

- **AUTO_INCREMENT**

  该关键字将字段指定为自动增长列。

  目前自动增长列默认为本地自动增长列，在各个DN上独立增长，互不影响，自增值全局不唯一。

  若在插入时不指定此列的值（或指定此列的值为0、NULL或DEFAULT），此列的值将由自增计数器自动增长得到。

  若插入或更新此列为一个大于当前自增计数器的值，执行成功后，自增计数器将刷新为此值。

  自增初始值由AUTO_INCREMENT [=] value子句设置，若不设置，默认为1。

版权所有 © 华为云计算技术有限公司

说明

- 仅在参数sql_compatibility='MYSQL'时可以指定自动增长列。
- 自动增长列数据类型只能为整数类型、4字节或8字节浮点类型、布尔类型。
  当自增值已经达到字段数据类型的最大值时，继续自增将产生错误。
- 每个表只能有一个自动增长列。
- 本地自动增长列建议为索引（非全局二级索引）的第一个字段，否则含有自动增长列的表进行某些操作时会产生错误，例如：ALTER TABLE EXCHANGE PARTITION。
- 自动增长列不能指定DEFAULT缺省值。
- CHECK约束的表达式中不能含有自动增长列。
- 可以指定自动增长列允许NULL，若不指定，默认自动增长列含有NOT NULL约束。
- 含有本地自动增长列的表创建时，会在每个DN上创建一个依赖于此列的本地序列作为自增计数器，不允许通过序列相关功能修改或删除此序列，可以查看序列的值。请勿使其他序列依赖或关联此自动增长列。
- 自增计数器自增和刷新操作不会回滚。
  - 数据插入到表之前，0/NULL会触发自增。数据插入或更新到表之后，会更新自增计数器。如果在自增之后出现了报错，导致数据没有插入或更新到表中，此时自增计数器不会回滚。后续插入语句基于自增计数器触发自增，会出现表中自动增长列的值不连续的情况。
  - 批量插入或导入预留自增缓存值也有可能产生自动增长列的值不连续的情况，详见auto_increment_cache参数说明。
- 自动增长列不支持作为分布列。
- 本地临时表暂不支持自动增长列。
- 复制表（DISTRIBUTE BY REPLICATION）暂不支持自动增长列。

- **UNIQUE [KEY] index_parameters**

  UNIQUE约束表示表里的一个字段或多个字段的组合必须在全表范围内唯一。

  对于唯一约束，NULL被认为是互不相等的。

  UNIQUE KEY仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持使用，与UNIQUE语义相同。

- **UNIQUE [index_name][USING method] ({ {column_name | (expression) } [ASC | DESC] }[, ...]) index_parameters**

  UNIQUE约束表示表里的一个字段或多个字段的组合必须在全表范围内唯一。

  对于唯一约束，NULL被认为是互不相等的。

  index_name为索引名。

  须知

  - index_name仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
  - 对于唯一键约束，constraint_name和index_name同时指定时，索引名为index_name。

- **PRIMARY KEY index_parameters**

  PRIMARY KEY [USING method] ({ column_name [ASC | DESC] } [, ...]) index_parameters

  主键约束声明表中的一个或者多个字段只能包含唯一的非NULL值。

版权所有 © 华为云计算技术有限公司

一个表只能声明一个主键。

- **REFERENCES**

  当前版本暂不支持REFERENCES子句。

- **USING method**

  指定创建索引的方法。

  取值范围请参见**CREATE INDEX**章节中的**USING method**

须知

- USING method仅在MYSQL兼容模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
- 在MYSQL模式下，未指定USING method时，对于ASTORE的存储方式，默认索引方法为btree；对于USTORE的存储方式，默认索引方法为ubtree。
- 对于默认建立全局二级索引的约束，由于全局二级索引底层使用ubtree存储，即使用户指定存储方式为btree，底层也会建立为ubtree。
- 当表的存储方式为USTORE时，SQL语句中约束指定为using btree，底层会自动将约束建立为using ubtree。

- **ASC | DESC**

  ASC表示指定按升序排序（默认）。DESC指定按降序排序。

须知

ASC | DESC只在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库不支持。

- **expression**

  创建一个基于该表的一个或多个字段的表达式索引约束，必须写在圆括号中。

须知

UNIQUE约束中的表达式索引只在MYSQL模式数据库下支持（即sql_compatibility = 'MYSQL'），其他模式数据库不支持。

- **DEFERRABLE | NOT DEFERRABLE**

  这两个关键字用于设置约束是否可推迟。不可推迟的约束将在每条命令执行后立即检查。可推迟约束可以延迟到事务结尾通过SET CONSTRAINTS命令检查，缺省值为NOT DEFERRABLE。目前UNIQUE约束和主键约束可以接受该子句，其他约束类型均不可推迟。

- **INITIALLY IMMEDIATE | INITIALLY DEFERRED**

  若约束可推迟，该子句声明检查约束的缺省时间。不可推迟的约束始终为INITIALLY IMMEDIATE且不可声明为INITIALLY DEFERRED。

  - 如果约束是INITIALLY IMMEDIATE（缺省），则在每条语句执行后立即检查。

版权所有 © 华为云计算技术有限公司

– 如果约束是INITIALLY DEFERRED，则仅在事务结束时进行检查。

约束检查的时间可以用SET CONSTRAINTS命令修改。

- **USING INDEX TABLESPACE tablespace_name**

  为UNIQUE约束或PRIMARY KEY约束相关的索引声明一个表空间。如果没有提供这个子句，这个索引将在default_tablespace中创建，如果default_tablespace为空，将使用数据库的缺省表空间。

- DATATYPE_CL = client_logic_type

  为可选参数，用于指定加密列的类型。

  取值范围：BYTEAWITHOUTORDER或者BYTEAWITHOUTORDERWITHEQUAL。

- **COLUMN_ENCRYPTION_KEY = column_encryption_key**

  为ENCRYPTED WITH约束中列加密密钥的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **ENCRYPTION_TYPE = encryption_type_value**

  为ENCRYPTED WITH约束中的加密类型，encryption_type_value的值为[DETERMINISTIC | RANDOMIZED]。

- **[DEFAULT] CHARACTER SET | CHARSET [=] default_charset**

  指定表的默认字符集，单独指定时会将表的默认字符序设置为指定的字符集的默认字符序。

- **[DEFAULT] COLLATE [=] default_collation**

  指定表的默认字符序，单独指定时会将表的默认字符集设置为指定的字符序对应的字符集。

  仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持该语法。字符序参见**表1-398**。

  说明

  表的字符集或字符序未显式指定时，若指定了模式的默认字符集或字符序，表字符集和字符序将从模式上继承。若模式的默认字符集或字符序不存在，当b_format_behavior_compat_options包含'default_collation'时，表的字符集和字符序将继承当前数据库的字符集及其对应的默认字符序。

建表示例

- **创建普通表**

```
gaussdb=# CREATE TABLE tbl_test1(
    id       int,
    name     varchar(50),
    province varchar (60), --省
    country  varchar (60) DEFAULT 'China' --国籍
);

gaussdb=# DROP TABLE tbl_test1;
```

- **WITH子句为表或者索引增加存储参数**

```
--建表指定填充因子。
gaussdb=# CREATE TABLE tbl_test2(
    id       int,
    name     varchar(50),
    province varchar (60), --省
    country  varchar (60) DEFAULT 'China' --国籍
) WITH (FILLFACTOR = 70);

--建表指定存储引擎。
gaussdb=# CREATE TABLE tbl_test3(
```


版权所有 © 华为云计算技术有限公司

```
    id       int,
    name     varchar(50),
    province varchar (60), --省
    country  varchar (60) DEFAULT 'China' --国籍
) WITH (STORAGE_TYPE = ASTORE);

--删除。
gaussdb=# DROP TABLE tbl_test2;
gaussdb=# DROP TABLE tbl_test3;
```

- **加密表**

  使用该参数前，请确保已通过GUC参数enable_tde开启透明加密功能，并通过GUC参数tde_key_info设置访问密钥服务的信息，在《开发指南》中“特性使用指导 > 透明数据加密”章节可获取该参数的详细使用方法。

```
gaussdb=# CREATE TABLE  t1 (c1 TEXT) WITH (enable_tde=on);

--删除表。
gaussdb=# DROP TABLE t1;
```

- **临时表**

```
--创建临时表，并指定提交事务时删除该临时表数据。
gaussdb=# CREATE TEMPORARY TABLE test_t2(
    id       CHAR(7),
    name     VARCHAR(20),
    province VARCHAR(60),                       --省
    country   VARCHAR(30) DEFAULT 'China'       --国籍
) ON COMMIT DELETE ROWS;
gaussdb=# DROP TABLE test_t2;
```

- **建表时指定字符集字符序**

```
--创建前置数据库。
gaussdb=# CREATE DATABASE testdb1 ENCODING = 'UTF8';
gaussdb=# \c testdb1

--创建t1表，设置t1的默认字符集为utf8mb4，默认字符序为utf8mb4_bin，设置c1字段为表的默认字符集字符序，设置c2字段的字符集为utf8mb4，字符序为utf8mb4_unicode_ci。
testdb1=# CREATE TABLE t1(c1 text, c2 text charset utf8mb4 collate utf8mb4_unicode_ci) charset utf8mb4 collate utf8mb4_bin;

--删除。
testdb1=# DROP TABLE t1;
testdb1=# \c postgres
gaussdb=# DROP DATABASE testdb1;
```

- **IF NOT EXISTS关键字**

  使用该关键字，表不存在时报NOTICE；如不用该关键字，则报ERROR。两种情况下表都不会创建成功。

```
gaussdb=# CREATE TABLE test_t3(id INT);
--创建一个已经存在同名的表test_t3。
gaussdb=# CREATE TABLE test_t3(id INT);
ERROR:  Relation "test_t3" already exists in schema "public".
DETAIL:  Creating new table with existing name in the same schema.

--使用IF NOT EXISTS关键字。
gaussdb=# CREATE TABLE IF NOT EXISTS test_t3(id INT);
NOTICE:  Relation test_t3 already exists, skipping.
CREATE TABLE

--删除表。
gaussdb=# DROP TABLE test_t3;
```

- **建表时指定表空间**

```
--创建表空间。
gaussdb=# CREATE TABLESPACE ds_tbs1 RELATIVE LOCATION 'tablespace/tablespace_1';

--创建表时，指定表空间。
gaussdb=# CREATE TABLE test(id CHAR(7), name VARCHAR(20)) TABLESPACE ds_tbs1;
```


版权所有 © 华为云计算技术有限公司

```
--删除表和表空间。
gaussdb=# DROP TABLE test;
gaussdb=# DROP TABLESPACE ds_tbs1;
```

- **建表时指定AUTO_INCREMENT自增列**

```
--本地自增列不作为分布列，每个DN从10开始自增（根据实际情况修改DATANODE名字，SELECT node_name FROM pgxc_node WHERE node_type = 'D';）。
gaussdb=# CREATE TABLE local_autoinc(col int AUTO_INCREMENT, col1 int) AUTO_INCREMENT = 10 DISTRIBUTE BY LIST(col1)(
    SLICE s1 VALUES (1) DATANODE datanode1,
    SLICE s2 VALUES (2) DATANODE datanode2
);

--建议自增列作为索引首列，创建一个索引。
gaussdb=# CREATE INDEX local_autoinc_ai ON local_autoinc(col);

--数据分布到DN1上，NULL触发自增，自增值为10。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(NULL,1);

--数据分布到DN2上，0触发自增，自增值为10。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(1 - 1,2);

--数据分布到DN1上，100不触发自增，插入成功后，自增计数更新为100。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(100,1);

--数据分布到DN1上，0触发自增，自增值为101。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(0,1);

--数据分布到DN2上，DEFAULT触发自增，自增值为11。
gaussdb=# INSERT INTO local_autoinc(col, col1) VALUES(DEFAULT,2);

gaussdb=# SELECT col,col1 FROM local_autoinc ORDER BY 2,1;
 col | col1
-----+------
  10 |    1
 100 |    1
 101 |    1
  10 |    2
  11 |    2
(5 rows)

--删除。
gaussdb=# DROP TABLE local_autoinc;
```

- **CREATE TABLE ... LIKE建表**

```
--创建源表t1。
gaussdb=# CREATE TABLE t1(col INT);
CREATE TABLE

gaussdb=# \d t1
      Table "public.t1"
 Column |  Type   | Modifiers
--------+---------+-----------
 col    | integer |

--创建目标表t2。
gaussdb=# CREATE TABLE t2(LIKE t1);
CREATE TABLE

gaussdb=# \d t2
      Table "public.t2"
 Column |  Type   | Modifiers
--------+---------+-----------
 col    | integer |

--删除。
gaussdb=# DROP TABLE t1,t2;
```


版权所有 © 华为云计算技术有限公司

建表添加约束示例

- **非空约束**

  非空约束的字段，如果在添加数据时没有指定值，就会报错。可以为表中多个字段添加非空约束。

```
--建表并给id字段添加非空约束。
gaussdb=# CREATE TABLE test_t4(
    id       CHAR(7) NOT NULL,
    name     VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'China'       --国籍
);

--插入数据时,如果不指定id的值或者id为NULL，会触发非空约束,导致插入失败。
gaussdb=# INSERT INTO test_t4 (name,province) VALUES ('scott','Shanghai');
ERROR: The null value in column "id" violates the not-null constraint.
DETAIL: Failing row contains (null, scott, Shanghai, China).


--删除表。
gaussdb=# DROP TABLE test_t4;
```

- **唯一约束**

  关键字UNIQUE给字段添加一个唯一约束，插入数据时该字段如有重复则触发约束，多个NULL不算重复，添加唯一约束时，会自动增加一个唯一索引。可以为表中多个字段添加唯一约束。

```
--建表添加唯一约束。
gaussdb=# CREATE TABLE test_t5(
    id       CHAR(7) UNIQUE USING INDEX TABLESPACE pg_default,
    name     VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'China'       --国籍
);

--也可以用如下写法，为唯一约束命名，以及为多个字段添加约束。
gaussdb=# CREATE TABLE test_t6(
    id       CHAR(7),
    name     VARCHAR(20),
    province VARCHAR(60),                     --省
    country   VARCHAR(30) DEFAULT 'China',       --国籍
    CONSTRAINT unq_test_id UNIQUE (id,name)
);
--插入id重复的数据，触发约束，导致插入失败。
gaussdb=# INSERT INTO test_t5(id) VALUES('0000010');
INSERT 0 1
gaussdb=# INSERT INTO test_t5(id) VALUES('0000010');
ERROR:  Duplicate key value violates unique constraint "test_t5_id_key".
DETAIL:  Key (id)=(0000010) already exists.
--多次插入id是NULL的数据不会触发约束。
gaussdb=# INSERT INTO test_t5(id) VALUES (NULL);
INSERT 0 1
gaussdb=# INSERT INTO test_t5(id) VALUES (NULL);
INSERT 0 1

--删除表。
gaussdb=# DROP TABLE test_t5;
gaussdb=# DROP TABLE test_t6;
```

- **主键约束**

  关键字PRIMARY KEY给字段添加唯一约束，要求字段唯一且不为空。添加主键约束时自动为该表创建唯一索引，也会为该字段自动增加一个非空约束。

  每个表里面只能定义一个主键约束，不能定义多个。

```
--建表添加主键约束。
gaussdb=# CREATE TABLE test_t6(
    id       CHAR(7) PRIMARY KEY,
```


版权所有 © 华为云计算技术有限公司

```
    name     VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'China'        --国籍
);
gaussdb=# INSERT INTO test_t6 (id,name,province) VALUES ('0000001','july','Beijing');

--也可以用如下写法，人工为唯一约束命名，以及为多个字段添加约束。
gaussdb=# CREATE TABLE test_t7(
    id       CHAR(7),
    name     VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'China',       --国籍
    CONSTRAINT pk_test_t6_id PRIMARY KEY (id,name)
);
--插入id为NULL的数据，触发约束。
gaussdb=# INSERT INTO test_t6 (id,name,province) VALUES (NULL,'july','Beijing');
ERROR:  The null value in column "id" violates the not-null constraint.
DETAIL:  Failing row contains (null, july, Beijing, China).

--插入id重复的数据，触发约束。
gaussdb=# INSERT INTO test_t6 (id,name,province) VALUES ('0000001','ben','Shanghai');
ERROR:  Duplicate key value violates unique constraint "test_t6_pkey"
DETAIL:  Key (id)=(0000001) already exists.
--删除表。
gaussdb=# DROP TABLE test_t6;
gaussdb=# DROP TABLE test_t7;
```

- **检查约束**

  关键字CHECK为字段添加检查约束，在检查约束中必须引用表中的一个或多个字段，并且表达式返回结果必须是一个布尔值。在表达式中不能包含子查询。对同一个字段可以同时定义检查约束和非空约束。

```
--建表，添加检查约束。
gaussdb=# CREATE TABLE test_t8 (
    id       CHAR(7),
    name     VARCHAR(20),
    age      INT CHECK(age > 0 AND age < 150)
);

--也可以使用如下SQL，人工为检查约束命名以及为一个或者多个字段添加检查约束。
gaussdb=# CREATE TABLE test_t9 (
    id       CHAR(7),
    name     VARCHAR(20),
    age      INT,
    CONSTRAINT chek_test_t8_age CHECK(age > 0 AND age < 150)
);
--插入不符合表达式的值，会触发检查约束导致插入失败。
gaussdb=# INSERT INTO test_t8 (id,name,age) VALUES ('0000007','scott',200);
ERROR:  New row for relation test_t8 violates check constraint test_t8_age_check.
DETAIL:  N/A
--删除表。
gaussdb=# DROP TABLE test_t8;
gaussdb=# DROP TABLE test_t9;
```

表数据分布示例

- **REPLICATION**

```
gaussdb=# CREATE TABLE test_replication(
    id       CHAR(7),
    name     VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'China'        --国籍
)DISTRIBUTE BY REPLICATION;

--查询表信息。
gaussdb=# \d+ test_replication
                                Table "public.test_replication"
 Column |         Type          |          Modifiers          | Storage  | Stats target | Description
```


版权所有 © 华为云计算技术有限公司

```
----------+------------------------+-----------------------------------+----------+--------------+-------------
 id       | character(7)           |                                   | extended |              |
 name     | character varying(20)  |                                   | extended |              |
 province | character varying(60)  |                                   | extended |              |
 country  | character varying(30) | default 'China'::character varying | extended |              |
Has OIDs: no
Distribute By: REPLICATION
Location Nodes: ALL DATANODES
Options: orientation=row, logical_repl_node=-1, compression=no, storage_type=USTORE, segment=off

--删除。
gaussdb=# DROP TABLE test_replication;
```

- **HASH**

```
--定义一个表，使用HASH分布。
gaussdb=# CREATE TABLE test_hash(
    id       CHAR(7),
    name     VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'China'        --国籍
)DISTRIBUTE BY HASH(id);

--插入数据。
gaussdb=# INSERT INTO test_hash VALUES ('0000001', 'Bob',   'Shanghai', 'China'),
                          ('0000002', 'Jack',  'Beijing',  'China'),
                          ('0000003', 'Scott', 'Beijing',  'China');
--查看数据分布情况。
gaussdb=# SELECT a.count,b.node_name
FROM (SELECT COUNT(*) AS count, xc_node_id FROM test_hash GROUP BY xc_node_id) a,
     pgxc_node b
WHERE a.xc_node_id=b.node_id ORDER BY a.count DESC;

count |     node_name
-------+--------------------
     2 | dn_6001_6002_6003
     1 | dn_6004_6005_6006
(2 rows)
--删除表。
gaussdb=# DROP TABLE test_hash;
```

- **MURMURHASH**

```
--测试环境包含1个CN和6个DN。
--创建NODE GROUP，NODE GROUP中的DN名字可以通过语句SELECT node_name FROM PGXC_NODE
WHERE node_type = 'D';查询，查询的结果按需替换CREATE NODE GROUP语句中WITH后的DN名。
gaussdb=# CREATE NODE GROUP NG1 WITH(datanode1, datanode2, datanode3, datanode4,
datanode5, datanode6);
--定义一个表，使用MURMURHASH分布。
gaussdb=# CREATE TABLE test_murmurhash1 (a int NOT NULL, b int) DISTRIBUTE BY
MURMURHASH(a) TO GROUP NG1;
gaussdb=# CREATE TABLE test_murmurhash2 (a int NOT NULL, b int) DISTRIBUTE BY
MURMURHASH(lpad_s(a,10,'0')) TO GROUP NG1;
--插入数据。
gaussdb=# INSERT INTO test_murmurhash1 VALUES(0,1);
gaussdb=# INSERT INTO test_murmurhash2 VALUES(1,2);
--查询数据。
gaussdb=# SELECT * FROM test_murmurhash1;
 a | b
---+---
 0 | 1
(1 row)
gaussdb=# SELECT * FROM test_murmurhash2;
 a | b
---+---
 1 | 2
(1 row)
--删除表。
gaussdb=# DROP TABLE test_murmurhash1;
gaussdb=# DROP TABLE test_murmurhash2;
```


版权所有 © 华为云计算技术有限公司

```
--删除NODE GROUP。
gaussdb=# DROP NODE GROUP NG1;
```

- **RANGE**

```
--定义一个表，使用RANGE分布（需要根据实际情况修改DN名字，查询DN节点名可以通过语句SELECT node_name FROM PGXC_NODE WHERE node_type = 'D';查询）。
gaussdb=# CREATE TABLE test_range(
    id       INT,
    name     VARCHAR(20),
    province VARCHAR(60),                       --省
    country   VARCHAR(30) DEFAULT 'China'       --国籍
)DISTRIBUTE BY RANGE(id)(
    SLICE s1 VALUES LESS THAN (100) DATANODE dn_6001_6002_6003,
    SLICE s2 VALUES LESS THAN (200) DATANODE dn_6004_6005_6006,
    SLICE s3 VALUES LESS THAN (MAXVALUE) DATANODE dn_6007_6008_6009
);

--插入数据。
gaussdb=# INSERT INTO test_range VALUES (52,  'Bob',    'Beijing',   'China');
gaussdb=# INSERT INTO test_range VALUES (100, 'Ben',    'Shanghai',  'China');
gaussdb=# INSERT INTO test_range VALUES (150, 'Scott',  'Guangzhou', 'China');
gaussdb=# INSERT INTO test_range VALUES (300, 'Jordan', 'Beijing',   'China');
--查看数据分布情况。
gaussdb=# SELECT a.count,b.node_name
FROM (SELECT COUNT(*) AS count, xc_node_id FROM test_range GROUP BY xc_node_id) a,
     pgxc_node b
WHERE a.xc_node_id=b.node_id ORDER BY a.count DESC;
 count |     node_name
-------+--------------------
     2 | dn_6004_6005_6006
     1 | dn_6001_6002_6003
     1 | dn_6007_6008_6009
(3 rows)
--查询各DN上存储的数据。
gaussdb=# SELECT b.node_name, a.*
FROM (SELECT *, xc_node_id FROM test_range) a,
     pgxc_node b
WHERE a.xc_node_id=b.node_id order by node_name;
     node_name     | id |  name  | province  | country | xc_node_id
-------------------+-----+--------+-----------+---------+-------------
 dn_6001_6002_6003 |  52 | Bob    | Beijing   | China   | -1072999043
 dn_6004_6005_6006 | 100 | Ben    | Shanghai  | China   |  -564789568
 dn_6004_6005_6006 | 150 | Scott  | Guangzhou | China   |  -564789568
 dn_6007_6008_6009 | 300 | Jordan | Beijing   | China   |  1532339558
(4 rows)
--删除表。
gaussdb=# DROP TABLE test_range;
```

- **LIST**

```
--定义一个表，使用LIST分布（需要根据实际情况修改DN名字，查询DN节点名可以通过语句SELECT node_name FROM PGXC_NODE WHERE node_type = 'D';查询）。
gaussdb=# CREATE TABLE test_list(
    id       INT,
    name     VARCHAR(20),
    country   VARCHAR(30) DEFAULT 'China'       --国籍
)DISTRIBUTE BY LIST(country)(
    SLICE s1 VALUES ('China') DATANODE dn_6001_6002_6003,
    SLICE s2 VALUES ('USA')   DATANODE dn_6004_6005_6006,
    SLICE s3 VALUES (DEFAULT) DATANODE dn_6007_6008_6009
);

--插入数据。
gaussdb=# INSERT INTO test_list VALUES (1,'Scott','China');
gaussdb=# INSERT INTO test_list VALUES (2,'Henry','USA');
gaussdb=# INSERT INTO test_list VALUES (3,'Michael','France');
gaussdb=# INSERT INTO test_list VALUES (4,'Jack','UK');
--查询各DN上存储的数据。
gaussdb=# SELECT b.node_name, a.*
FROM (SELECT *, xc_node_id FROM test_list) a,
     pgxc_node b
```


版权所有 © 华为云计算技术有限公司

```
WHERE a.xc_node_id=b.node_id order by node_name;
     node_name      | id |  name   | country | xc_node_id
--------------------+----+---------+---------+-------------
 dn_6001_6002_6003 |  1 | Scott   | China   | -1072999043
 dn_6004_6005_6006 |  2 | Henry   | USA     |  -564789568
 dn_6007_6008_6009 |  3 | Michael | France  |  1532339558
 dn_6007_6008_6009 |  4 | Jack    | UK      |  1532339558
(4 rows)
--删除表。
gaussdb=# DROP TABLE test_list;
```

相关链接

ALTER TABLE，**DROP TABLE**，**CREATE TABLESPACE**

优化建议

- UNLOGGED
  - UNLOGGED表和表上的索引因为数据写入时不通过WAL日志机制，写入速度远高于普通表。因此，可以用于缓冲存储复杂查询的中间结果集，增强复杂查询的性能。
  - UNLOGGED表无主备机制，在系统故障或异常断点等情况下，会有数据丢失风险，因此，不可用来存储基础数据。
- TEMPORARY | TEMP
  - 临时表只在当前会话可见，会话结束后会自动删除。
  - 除了当前CN外，其他CN对于该临时表不可见。
- LIKE
  - 新表自动从这个表中继承所有字段名及其数据类型和非空约束，新表与源表之间在创建动作完毕之后是完全无关的。
- LIKE INCLUDING DEFAULTS
  - 源表上的字段缺省表达式仅在指定INCLUDING DEFAULTS时，才会复制到新表中。缺省是不包含缺省表达式的，即新表中的所有字段的缺省值都是NULL。
- LIKE INCLUDING CONSTRAINTS
  - 源表上的CHECK约束仅在指定INCLUDING CONSTRAINTS时，才会复制到新表中，其他类型的约束不会复制到新表中。非空约束总是复制到新表中。此规则同时适用于表约束和列约束。
- LIKE INCLUDING INDEXES
  - 如果指定了INCLUDING INDEXES，则源表上的索引也将在新表上创建，默认情况下，不建立索引。新表上的索引所在表空间和源表上的索引所在表空间保持一致。
- LIKE INCLUDING STORAGE
  - 如果指定了INCLUDING STORAGE，则复制列的STORAGE设置会复制到新表中，默认情况下，不包含STORAGE设置。
- LIKE INCLUDING COMMENTS
  - 如果指定了INCLUDING COMMENTS，则源表列、约束和索引的注释会复制到新表中。默认情况下，不复制源表的注释。
- LIKE INCLUDING PARTITION

版权所有 © 华为云计算技术有限公司

  - 如果指定了INCLUDING PARTITION，则源表的分区定义会复制到新表中，同时新表将不能再使用PARTITION BY子句。默认情况下，不复制源表的分区定义。
- LIKE INCLUDING RELOPTIONS
  - 如果指定了INCLUDING RELOPTIONS，则源表的存储参数（即源表的WITH子句）会复制到新表中。默认情况下，不复制源表的存储参数。
- LIKE INCLUDING DISTRIBUTION
  - 如果指定了INCLUDING DISTRIBUTION，则源表的分布信息会复制到新表中，包括分布类型和分布列，同时新表将不能再使用DISTRIBUTE BY子句。默认情况下，不复制源表的分布信息。
- LIKE INCLUDING ALL
  - INCLUDING ALL包含了INCLUDING DEFAULTS、INCLUDING CONSTRAINTS、INCLUDING INDEXES、INCLUDING STORAGE、INCLUDING COMMENTS、INCLUDING PARTITION、INCLUDING RELOPTIONS、INCLUDING DISTRIBUTION和INCLUDING ILM的内容。
- ORIENTATION ROW
  - 创建行存表，行存储适合于OLTP业务，此类型的表上交互事务比较多，一次交互会涉及表中的多个列，用行存查询效率较高。
- DISTRIBUTE BY
  - 事实表或者数据量较大的维度表建议创建为分布表。对指定的列进行Hash操作，通过映射，把数据分布到指定DN。语法为：DISTRIBUTE BY HASH(column_name)。
  - 数据量较小的维度表建议创建为复制表。表的每条记录存在所有数据节点（DN）中，即每个数据节点都有完整的表数据。语法为：DISTRIBUTE BY REPLICATION。

1.13.9.45 CREATE TABLESPACE

功能描述

在数据库中创建一个新的表空间。

注意事项

- 系统管理员或者继承了内置角色gs_role_tablespace权限的用户可以创建表空间。
- 不允许在一个事务块内部执行CREATE TABLESPACE操作。
- 当表空间创建过程中内部目录/文件操作成功但后续步骤失败时，系统会产生残留的目录/文件，重新创建表空间时，需要用户手动清理表空间指定的目录下残留的内容。如果在创建过程中涉及到数据目录下的表空间软连接残留，需要先将软连接的残留文件删除，再重新执行OM相关操作。
- CREATE TABLESPACE不支持两阶段事务，如果部分节点执行失败，不支持回滚。
- 创建表空间时，系统会按顺序将创建指令分发到各节点，如果远端节点执行失败则会提示该报错信息的来源：CONTEXT: Error message received from nodes: *节点名称*。
- 在公有云场景下一般不建议用户使用自定义的表空间。

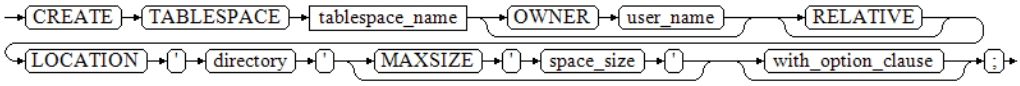
  原因：用户自定义表空间通常配合主存（即默认表空间所在的存储设备，如磁盘）以外的其他存储介质使用，以隔离不同业务可以使用的I/O资源，而在公有云

版权所有 © 华为云计算技术有限公司

场景下，存储设备都是采用标准化的配置，无其他可用的存储介质，自定义表空间使用不当不利于系统长稳运行以及影响整体性能，因此建议使用默认表空间即可。

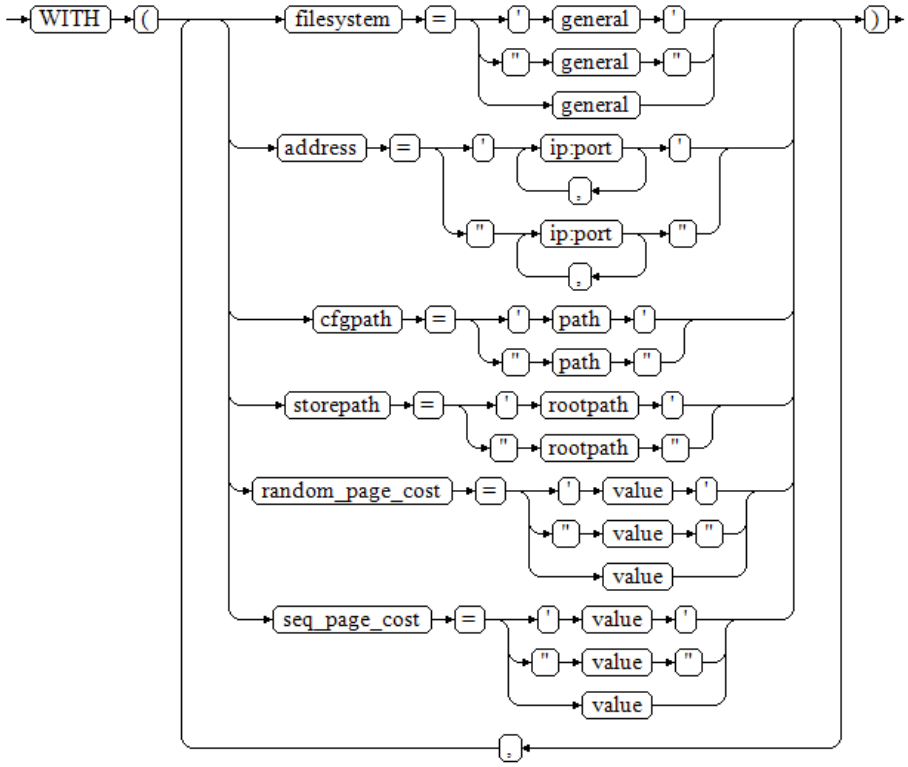
语法格式

```
CREATE TABLESPACE tablespace_name
    [ OWNER user_name ] [ RELATIVE ] LOCATION 'directory' [ MAXSIZE 'space_size' ]
    [with_option_clause];
```



其中普通表空间的with_option_clause为：

```
WITH ({filesystem= { ' general ' | " general " | general } | address = { ' ip:port [, ... ] ' | " ip:port [, ... ] " } | cfgpath = { ' path ' | " path " } | storepath = { ' rootpath ' | " rootpath " } | random_page_cost = { ' value ' | " value " | value } | seq_page_cost = { ' value ' | " value " | value }}[, ... ])
```



参数说明

- **tablespace_name**
  要创建的表空间名称。
  取值范围：字符串，要符合**标识符命名规范**。

  说明

  表空间名称不能和数据库中的其他表空间重名，且不能以“pg”作为表空间名称的前缀。

- **OWNER user_name**
  表空间的所有者。默认情况下，新表空间的所有者为当前用户。
  取值范围：字符串，已存在的用户。

版权所有 © 华为云计算技术有限公司

说明

只有系统管理员可以进行创建表空间，可以通过OWNER子句将表空间的所有权赋予给其他非系统管理员。

- **RELATIVE**
  - 若指定该参数，表示使用相对路径，表空间的目录LOCATION是相对于各个CN/DN数据目录下的。

    目录层次：CN和DN的数据目录/pg_location/相对路径。相对路径最多指定两层。
  - 若没有指定该参数，表示使用绝对表空间路径，表空间的目录LOCATION需要使用绝对路径。
- **LOCATION directory**

  用于表空间的目录。当创建绝对表空间路径时（即没有指定RELATIVE参数），对于目录存在如下要求：
  - 系统用户必须对该目录拥有读写权限，并且目录为空。如果该目录不存在，将由系统自动创建。
  - 目录必须是绝对路径，目录中不得含有特殊字符（如$、>等）。
  - 目录不允许指定在数据库数据目录下。
  - 目录需为本地路径。
  - 建议首选使用Ext4格式文件的目录。

  取值范围：字符串，有效的目录。
- **MAXSIZE 'space_size'**

  表空间在单个DN上的最大值。

  取值范围：当前支持的单位包括KB、MB、GB、TB以及PB，解析后的数值以KB为单位且范围不能够超过64比特表示的有符号整数，即1KB~9007199254740991KB。
- **address**

  IP地址以及端口号。
- **cfgpath**

  配置文件地址。
- **storepath**

  根路径地址。
- **filesystem**

  表空间使用的文件系统。

  取值范围：
  - general：普通文件系统。
  - hdfs：Hadoop分布式文件系统。当前版本不支持。

  默认值：general。
- **random_page_cost**

  随机读取page所需要的时间和资源。

  取值范围：0~1.79769e+308。

  默认值：使用GUC参数random_page_cost的值。

版权所有 © 华为云计算技术有限公司

- **seq_page_cost**
  指定顺序读取page所需要的时间和资源。
  取值范围：0~1.79769e+308。
  默认值：使用GUC参数seq_page_cost的值。

示例

```
--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_location1 RELATIVE LOCATION 'test_tablespace/test_tablespace_1';

--创建表空间指定最大值。
gaussdb=# CREATE TABLESPACE tbs_location2 RELATIVE LOCATION 'test_tablespace/test_tablespace_2'
MAXSIZE '10G';

--查询表空间信息。
gaussdb=# SELECT * FROM pg_tablespace WHERE spcname = 'tbs_location2';
    spcname    | spcowner | spcacl | spcoptions | spcmaxsize | relative
---------------+----------+--------+------------+------------+----------
 tbs_location2 |       10 |        |            | 10485760 K | t
(1 row)

--创建用户joe。
gaussdb=# CREATE ROLE joe IDENTIFIED BY '********';

--创建表空间，且所有者指定为用户joe。
gaussdb=# CREATE TABLESPACE tbs_location3 OWNER joe RELATIVE LOCATION 'test_tablespace/
test_tablespace_3';

--创建表空间指定绝对路径（请根据实际情况修改路径）。
gaussdb=# CREATE TABLESPACE tbs_location4 LOCATION '/tmp/tbs_location4';

--删除表空间及用户。
gaussdb=# DROP TABLESPACE tbs_location1;
gaussdb=# DROP TABLESPACE tbs_location2;
gaussdb=# DROP TABLESPACE tbs_location3;
gaussdb=# DROP TABLESPACE tbs_location4;
gaussdb=# DROP ROLE joe;
```

相关链接

CREATE DATABASE，**CREATE TABLE**，**CREATE INDEX**，**DROP TABLESPACE**，**ALTER TABLESPACE**

1.13.9.46 CREATE TABLE AS

功能描述

根据查询结果进行创建表。

CREATE TABLE AS创建一个表并且用来自SELECT命令的结果填充该表。该表的字段和SELECT输出字段的名称及数据类型相关。用户可以通过指定字段名称列表进行覆盖SELECT输出字段的名称。

CREATE TABLE AS和创建视图部分逻辑相似，CREATE TABLE AS会创建一个新表并且只将SELECT命令的第一次结果写入新表中。这个表之后的数据将不会根据源表变化而改变；而对于视图，只要视图被查询，那么视图的定义SELECT语句将会被重新计算。

注意事项

- 通过此方式创建的新表只继承SELECT输出字段的名称及数据类型。

版权所有 © 华为云计算技术有限公司

- 分区表不能采用CREATE TABLE AS进行创建。
- 如果在建表过程中数据库系统发生故障，系统恢复后可能无法自动清除之前已创建的、大小非0的磁盘文件。此种情况出现概率小，不影响数据库系统的正常运行。
- 在为数据对象增加或者变更ILM策略时，如果追加了行级表达式，目前仅支持白名单中列出的函数作为行级表达式，具体白名单函数列表请参见**行表达式函数白名单**。

  说明

  在ILM策略的ON（EXPR）行级表达式支持的函数中，存在部分函数的输出可能会受兼容性参数的影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。

语法格式

```
CREATE [ [ GLOBAL | LOCAL ] [ TEMPORARY | TEMP ] | UNLOGGED ] TABLE [ IF NOT EXISTS ] table_name
   [ (column_name [, ...] ) ]
   [ { ENGINE [ = ] { InnoDB | 'InnoDB' | "InnoDB" } } [ ,... ] [ ... ] ]
   [ WITH ( {storage_parameter = value} [, ... ] ) ]
   [ ON COMMIT { PRESERVE ROWS | DELETE ROWS } ]
   [ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION
 | NONE}]
   [ TABLESPACE tablespace_name ]
   [ DISTRIBUTE BY { REPLICATION | { [HASH ] ( column_name ) } } ]
   [ TO { GROUP groupname | NODE ( nodename [, ... ] ) } ]
   AS query
   [ WITH [ NO ] DATA ];
```

各个字段详细的信息请参见**语法格式**。

参数说明

- **UNLOGGED**

  指定该关键字，创建的表为非日志表。在非日志表中写入的数据不会被写入到预写日志中，相较于普通表速度更快。但是非日志表在冲突、执行操作系统重启、数据库重启、主备切换、切断电源操作或异常关机后会被全部清空，会造成数据丢失的风险。非日志表中的内容也不会被复制到备服务器中。在非日志表中创建的索引也不会被自动记录。

  - 使用场景：非日志表不能保证数据的安全性，用户应该在确保数据已经做好备份的前提下使用，例如：系统升级时进行数据的备份。
  - 故障处理：当异常关机等操作导致非日志表上的索引发生数据丢失时，用户应该对发生错误的索引进行重建。

- **GLOBAL | LOCAL**

  创建临时表时可以在TEMP或TEMPORARY前指定GLOBAL或LOCAL关键字。如果指定GLOBAL关键字，GaussDB会创建全局临时表，否则创建本地临时表。

- **TEMPORARY | TEMP**

  指定TEMP或TEMPORARY关键字，创建的表为临时表。临时表分为全局临时表（GLOBAL）和本地临时表（LOCAL）。创建临时表时如果指定GLOBAL关键字则为全局临时表，否则为本地临时表。

  - 全局临时表（GLOBAL）的元数据对所有会话可见，会话结束后元数据继续存在。会话与会话之间的用户数据、索引和统计信息相互隔离，每个会话只能看到或更改自己提交的数据。全局临时表有两种模式：

版权所有 © 华为云计算技术有限公司

  - 一种基于会话级别（ON COMMIT PRESERVE ROWS），当会话结束时自动清空用户数据。
  - 一种基于事务级别（ON COMMIT DELETE ROWS），当执行COMMIT或ROLLBACK时自动清空用户数据。

  建表时如果没有指定ON COMMIT选项，默认为会话级别。与本地临时表不同，全局临时表建表时可以指定非pg_temp_开头的Schema。
- 本地临时表（LOCAL）只在当前会话可见，本会话结束后会自动删除。因此，在除当前会话连接的CN以外的其他CN故障时，仍然可以在当前会话上创建和使用临时表。由于临时表只在当前会话创建，对于涉及对临时表操作的DDL语句，会产生DDL失败的报错。因此，建议DDL语句中不要对临时表进行操作。

须知

- 本地临时表（LOCAL）通过每个会话中以pg_temp开头的Schema进行保证只对当前会话可见，因此，不建议用户在日常操作中手动删除以pg_temp、pg_toast_temp开头的Schema。
- 如果建表时不指定TEMPORARY | TEMP关键字，而指定表的Schema为当前会话以pg_temp开头的Schema，则该表会被创建为临时表。
- 临时表只对当前会话可见，因此不支持与\parallel on（并行执行）同时使用。
- 临时表不支持DN故障或者主备切换。
- 如果存在其他会话正在使用全局临时表和索引时，则禁止对全局临时表和索引进行ALTER或DROP操作。
- 全局临时表的DDL只会影响当前会话的用户数据和索引，例如TRUNCATE、REINDEX、ANALYZE只对当前会话有效。

- **IF NOT EXISTS**

  判断是否存在同名的表。如果指定IF NOT EXISTS关键字，创建表前会在当前Schema中查找是否已有名字相同的依赖关系relation，若存在同名的relation，则不会新建并返回NOTICE提示。未指定IF NOT EXISTS关键字时，若Schema中存在同名relation，则返回ERROR告警。
- **table_name**

  创建的表名。

  取值范围：字符串，要符合**标识符命名规范**。
- **column_name**

  可选。新表中的字段名。如果没有指定字段名，那么新表的字段名和SELECT语句输入的字段名一致。

  取值范围：字符串，要符合**标识符命名规范**。
- **ENGINE**

  MYSQL模式下支持，仅语法适配，且只支持设置InnoDB，并无实际效果。
- **WITH (storage_parameter [= value] [, ...])**

  为表或索引指定一个可选的存储参数。参数的详细说明如下所示。

版权所有 © 华为云计算技术有限公司

说明

支持透明加密相关存储参数：enable_tde、encrypt_algo、dek_cipher、key_type、cmk_id。

详细参数说明请参见**CREATE TABLE**章节。

- FILLFACTOR

  表的填充因子（fillfactor），一个介于10和100之间的百分数。未指定时默认值为100（完全填充）。在Ustore存储引擎下，该值的默认值为92。如果指定了较小的填充因子，INSERT操作仅按照填充因子指定的百分率填充表页，每个页上的剩余空间将用于在该页上更新行。较小的填充因子对于UPDATE操作，可以在同一页上有机会放置同一条记录的新版本，相较于将新版本的数据放置在其他页上更为有效。对于一个从不更新的表，建议将填充因子设为100，对于频繁更新的表，建议选择较小的填充因子。该参数仅对行存表有效。

  取值范围：10~100

- ORIENTATION

  取值范围：ROW，表的数据将以行式存储。

  默认值：ROW

- autovacuum_enabled

  表示是否开启表的自动清理功能。

  取值范围：on/off

  默认值：on

- autovacuum_vacuum_threshold

  开启自动清理功能时，指定在该表中触发VACUUM所需的更新或删除的最小元组数（仅对Astore表生效）。

  取值范围：0~2147483647

  默认值：未指定时与GUC参数autovacuum_vacuum_threshold一致。

- autovacuum_analyze_threshold

  开启自动清理功能时，指定在该表中触发ANALYZE所需的插入、更新或删除的最小元组数。

  取值范围：0~2147483647

  默认值：未指定时与GUC参数autovacuum_analyze_threshold一致。

- autovacuum_vacuum_scale_factor

  开启自动清理功能时，指定在该表中触发VACUUM所需的插入、更新或删除元组的比例（仅对Astore表生效）。

  取值范围：0.0~100.0

  默认值：未指定时与GUC参数autovacuum_vacuum_scale_factor一致。

- autovacuum_analyze_scale_factor

  开启自动清理功能时，指定在该表中触发ANALYZE所需的插入、更新或删除元组的比例。

  取值范围：0.0~100.0

  默认值：未指定时与GUC参数autovacuum_analyze_scale_factor一致。

- autovacuum_freeze_min_age

  开启自动清理功能时，在该表参数指定了一个行版本的最小范围，超过这个范围的行会被冻结。

版权所有 © 华为云计算技术有限公司

取值范围：0~576460752303423487

默认值：未指定时与GUC参数vacuum_freeze_min_age一致。

- autovacuum_freeze_max_age

  开启自动清理功能时，表pg_class.relfrozenxid字段在超过指定个事务后，就会强制执行VACUUM操作，即使自动清理被禁用，系统也会启动AUTOVACUUM进程。清理操作还允许从pg_clog/子目录中删除旧文件（仅对Astore表生效）。

  取值范围：100000~576460752303423487

  默认值：未指定时与GUC参数autovacuum_freeze_max_age一致。

- autovacuum_freeze_table_age

  开启自动清理功能时，该表被标记为不需要自动清理时，将保持不变的时间（仅对Astore表生效）。

  取值范围：0~576460752303423487

  默认值：未指定时与GUC参数vacuum_freeze_table_age一致。

- vacuum_truncate

  表示是否开启表的VACUUM TRUNCATE功能，在VACUUM时截断末尾空页。

  取值范围：on/off

  默认值：on

  说明

  - 该参数仅对Astore生效，对hashbucket表和segment表不生效。
  - 从不支持该参数版本升级至支持该参数版本时，不支持在升级观察期创建、修改该参数。
  - 支持通过toast.vacuum_truncate参数单独控制TOAST表是否截断末尾空页。

- hashbucket

  创建hash bucket存储。该参数仅支持行存表和行存Range表。

  取值范围：on/off

  默认值：off

须知

当前版本hashbucket表相关DDL操作性能受限，不建议频繁对hashbucket表进行DDL操作。

- hasrowid

  指定表是否包含rowid、rowno系统列。设置hasrowid = on时，该表会增加rowid系统列和rowno系统列，且会在rowno系统列上关联创建序列和索引。

  取值范围：on、off

  默认值：off

版权所有 © 华为云计算技术有限公司

说明

- 此参数只支持ORA兼容模式数据库（即sql_compatibility = 'ORA'）。
- 此参数不支持本地临时表、Unlogged表、全局临时表。
- 此参数不支持和hasuids、oids参数同时指定。
- 开启此参数时，不支持定义普通列名为rowno、rowid。

– partition_statistics_min_rows

对分区表执行ANALYZE命令时，用于确定分区级最小采样行数的参数值。

取值范围：[0, 10000]

默认值：50

说明

- 此参数仅对分区表有效。
- 此参数仅在ANALYZE命令的PARTITION_MODE = ALL时，且GUC参数enable_partition_statistics_min_rows启用时生效，PARTITION_MODE说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。
- 上限约束：最终的分区表整表采样行数不超过GUC参数maintenance_work_mem可容纳的行数。

- **ON COMMIT { PRESERVE ROWS | DELETE ROWS }**

  用于表示在事务中执行创建临时表操作且当事务提交时，对该临时表进行的其他操作。当前仅支持PRESERVE ROWS和DELETE ROWS选项。

  – PRESERVE ROWS（缺省值）：提交时不对临时表执行任何操作，临时表及其表数据保持不变。

  – DELETE ROWS：提交时删除临时表中数据。

- **ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [MEDIUM | HIGH] ROW AFTER n { day | month | year } OF NO MODIFICATION [ON (EXPR)] | TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}**

  – 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW为行存添加高级压缩中/高级别策略。

    ▪ AFTER n { day | month | year } OF NO MODIFICATION：表示n天/月/年没有修改的行，高级压缩不支持hour。

    ▪ ON（EXPR）：行级表达式，用于判断行的冷热。

    说明

    - 在ILM策略的ON（EXPR）行级表达式支持的函数中，部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转换大写。
    - 压缩级别是可选项，不指定时默认使用MEDIUM级别。

  – 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT为行存添加跨页透明压缩策略，子分区继承分区的策略。

    AFTER n { hour | day | month | year } OF NO MODIFICATION：表示n小时/天/月/年没有修改的行。

    压缩级别不指定时默认为MEDIUM。

  – 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS NONE显式指定行存表不添加跨页透明压缩策略。

版权所有 © 华为云计算技术有限公司

- **TABLESPACE tablespace_name**

  创建新表时指定此关键字，表示新表将要在指定表空间tablespace_name内创建。如果没有声明，将使用默认表空间。

- **DISTRIBUTE BY**

  详细信息请参见**CREATE TABLE**章节中的**DISTRIBUTE BY**。

- **TO { GROUP groupname | NODE (nodename [, ...]) }**

  指定创建表所在的Node Group。

  该语法仅供内部扩容工具使用，不建议用户直接使用。

- **AS query**

  一个SELECT VALUES命令或者一个运行预备好的SELECT或VALUES查询的EXECUTE命令。

须知

在需要进行数据类型转换时，特别是使用CREATE TABLE AS SELECT语句时，建议采用显式类型转换，避免因隐式类型转换可能导致的意外结果。

- **[WITH [NO] DATA]**

  创建表时，是否将查询到的数据进行插入。默认插入，当指定NO时，则表示不插入数据。

示例

- 不指定字段名时新表字段和SELECT查询结果一致。

```
-- 创建test1表并向表中插入2条记录。
gaussdb=# CREATE TABLE test1(col1 int PRIMARY KEY,col2 varchar(10));
gaussdb=# INSERT INTO test1 VALUES (1,'col1'),(101,'col101');
-- 查询表中col1<100的数据。
gaussdb=# SELECT * FROM test1 WHERE col1 < 100;
 col1 | col2
------+------
    1 | col1
(1 row)

-- 创建test2表并向表中插入上面查询的数据。
gaussdb=# CREATE TABLE test2 AS SELECT * FROM test1 WHERE col1 < 100;

-- 查询test2表表结构。
gaussdb=# \d test2;
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 col1   | integer               |
 col2   | character varying(10) |
```

- 为新表指定字段名。

```
-- 使用test1复制一个新表test3并指定字段名。
gaussdb=# CREATE TABLE test3(c1,c2) AS SELECT * FROM test1;

-- 查询test3表结构。
gaussdb=# \d test3
           Table "public.test3"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 c1     | integer               |
 c2     | character varying(10) |

-- 删除。
gaussdb=# DROP TABLE test1,test2,test3;
```


版权所有 © 华为云计算技术有限公司

- 创建开启ILM策略的表。

```
--创建ORA兼容的数据库。
gaussdb=# CREATE DATABASE ilmtabledb WITH dbcompatibility = 'ORA';
gaussdb=# \c ilmtabledb

--开启数据库ILM特性。
ilmtabledb=#  ALTER DATABASE SET ILM = on;

--创建一个开启ILM策略的表ilm_table并插入old_table的数值。
ilmtabledb=# CREATE TABLE old_table (a int);
ilmtabledb=# CREATE TABLE ilm_table
  ILM ADD POLICY ROW STORE COMPRESS ADVANCED
  ROW AFTER 3 MONTHS OF NO MODIFICATION
  AS (SELECT * FROM old_table);

--删除。
ilmtabledb=# DROP TABLE old_table,ilm_table;
ilmtabledb=# \c postgres
gaussdb=# DROP DATABASE ilmtabledb;
```

相关链接

CREATE TABLE，**SELECT**

1.13.9.47 CREATE TABLE PARTITION

功能描述

创建分区表。分区表是把逻辑上的一张表根据某种方案分成几张物理块进行存储，这张逻辑上的表称之为分区表，物理块称之为分区。分区表是一张逻辑表，不存储数据，数据实际是存储在分区上的。

常见的分区方案有范围分区（Range Partitioning）、间隔分区（Interval Partitioning）、哈希分区（Hash Partitioning）、列表分区（List Partitioning）以及数值分区（Value Partitioning）等。目前行存表支持范围分区、哈希分区以及列表分区。

- 范围分区：

  根据表的一列或者多列，将要插入表的记录分为若干个范围，这些范围在不同的分区里没有重叠。为每个范围创建一个分区，用来存储相应的数据。

  范围分区策略：记录插入分区的方式。根据分区键值将记录映射到已创建的某个分区上，如果可以映射到已创建的某一分区上，则把记录插入到对应的分区上，否则给出报错和提示信息。这是最常用的分区策略。

- 哈希分区

  根据表的一列，为每个分区指定模数和余数，将要插入表的记录划分到对应的分区中，每个分区所持有的行都需要满足条件（分区键的值除以为其指定的模数将产生为其指定的余数）。

  哈希分区策略：根据分区键值将记录映射到已创建的某个分区上，如果可以映射到已创建的某一分区上，则把记录插入到对应的分区上，否则返回报错和提示信息。

- 列表分区

  列表分区是根据表的一列，将要插入表的记录通过每一个分区中出现的键值划分到对应的分区中，这些键值在不同的分区里没有重叠。为每组键值创建一个分区，用来存储相应的数据。

版权所有 © 华为云计算技术有限公司

列表分区策略：根据分区键值将记录映射到已创建的某个分区上，如果可以映射到已创建的某一分区上，则把记录插入到对应的分区上，否则给出报错和提示信息。

创建分区的优势：

- 查询性能得以提升。特别是对于表中访问率较高的行位于一个单独分区或少数几个分区内，对于创建分区后，可以减少数据的搜索空间，提高数据访问效率。
- 当查询或更新一个分区的大部分记录时，仅对单独分区进行连续扫描，而并非对整个数据表进行访问，性能得以提升。
- 如果需要大量加载或者删除的记录位于单独的分区上，可以直接读取或删除该分区的数据。同时可以避免由于大量DELETE导致的VACUUM超载（哈希分区不支持删除分区）。

注意事项

- 唯一约束和主键约束的约束键包含所有分区键将为约束创建LOCAL索引，否则创建GLOBAL索引。
- 目前哈希分区仅支持单列构建分区键，暂不支持多列构建分区键。
- 对于分区表PARTITION FOR (values)语法，values只能是常量。
- 对于分区表PARTITION FOR (values)语法，values在需要数据类型转换时，建议使用强制类型转换，以防隐式类型转换结果与预期不符。
- 单个表的最大分区数为1048575（包括一级分区和二级分区）。在默认配置下，建议单个表的分区数不超过400个。具体可参考《开发指南》中“特性使用指导 > 分区表 > 分区表介绍 > 分区基本使用 > 创建分区表”章节。
- 当分区数太多导致内存不足时，会间接导致性能急剧下降。
- 不支持XML类型数据作为分区键、二级分区键。
- 不支持在rowid/rowno系统列上创建外键约束。
- 对于分区表进行UPDATE/DELETE时，如果生成的计划不是FQS或Stream计划，语句执行效率会比较差。建议排查语句，消除不可下推因素，从而生成FQS或Stream计划。
- 在为数据对象增加或者变更ILM策略的时候，如果追加了行级表达式，需要注意行表达式目前只支持白名单中列出的函数。具体白名单函数列表请参见**行表达式函数白名单**。

  说明

  在ILM策略的ON（EXPR）行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。

语法格式

```
CREATE TABLE [ IF NOT EXISTS ] partition_table_name
{ ( [
   { column_name data_type [ CHARACTER SET | CHARSET charset ] [ COLLATE collation ]
[ column_constraint [ ... ] ]
   | table_constraint
   | LIKE source_table [ like_option [...] ] }
   [, ... ]
] )
   | LIKE source_table }
   [ table_option [ ,... ] [ ... ] ]
   [ htap_option ]
```


版权所有 © 华为云计算技术有限公司

```
  [ WITH ( {storage_parameter = value} [, ... ] ) ]
  [ ILM ADD POLICY ROW STORE COMPRESS
{ ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
( EXPR )]
| TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION
| NONE}]
  [ TABLESPACE tablespace_name ]
  [ DISTRIBUTE BY { REPLICATION | HASH ( column_name [, ...] )
  | MURMURHASH ( diskey_expr )
  | RANGE ( column_name [, ...] ) { SLICE REFERENCES tablename | ( slice_less_than_item [, ...] )
  | ( slice_start_end_item [, ...] ) }
  | LIST ( column_name [, ...] ) { SLICE REFERENCES tablename | ( slice_values_item [, ...] ) }
  } ]
  [ TO { GROUP groupname | NODE ( nodename [, ... ] ) } ]
  PARTITION BY {
     {RANGE [COLUMNS] (partition_key) [ PARTITIONS integer ] ( partition_less_than_item [, ... ] )} |
     {RANGE [COLUMNS] (partition_key) [ PARTITIONS integer ] ( partition_start_end_item [, ... ] )} |
     {LIST [COLUMNS] (partition_key) [ PARTITIONS integer ] ( PARTITION partition_name VALUES [IN]
(list_values) [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n
{ day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT
AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [ htap_option ][TABLESPACE [=]
tablespace_name][, ... ])} |
     { HASH (partition_key) [ PARTITIONS integer ] ( PARTITION partition_name [ ILM ADD POLICY ROW
STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO
MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month |
year } OF NO MODIFICATION | NONE}] [ htap_option ][TABLESPACE [=] tablespace_name][, ... ])}
  } [ { ENABLE | DISABLE } ROW MOVEMENT ];
```

- 其中table_option为：

  ```
  { COMMENT [ = ] 'string' |
    AUTO_INCREMENT [ = ] value |
    [ DEFAULT ] CHARACTER SET | CHARSET [ = ] default_charset |
    [ DEFAULT ] COLLATE [ = ] default_collation }
  ```

- 其中htap_option为：

  ```
  { COLVIEW [ PRIORITY { HIGH | LOW | NONE } ] |
    NOCOLVIEW }
  ```

- 列约束column_constraint：

  ```
  [ CONSTRAINT constraint_name ]
  { NOT NULL |
    NULL |
    CHECK ( expression ) |
    AUTO_INCREMENT |
    COMMENT 'string' |
    DEFAULT default_expr |
    ON UPDATE update_expr |
    UNIQUE [KEY] [ index_parameters ] |
    PRIMARY KEY [ index_parameters] }
  [ DEFERRABLE | NOT DEFERRABLE ][ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
  ```

- 表约束table_constraint：

  ```
  [ CONSTRAINT constraint_name ]
  { CHECK ( expression ) |
    UNIQUE [ index_name ] [ USING method ] ( { column_name [ ASC | DESC ] } [, ... ] )
  [ index_parameters ] |
    PRIMARY KEY ( column_name [, ... ] ) [ index_parameters]}
  [ DEFERRABLE | NOT DEFERRABLE ][ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
  { [ COMMENT 'string' ] [ ... ] }
  ```

- like选项like_option：

  ```
  { INCLUDING | EXCLUDING } { DEFAULTS | CONSTRAINTS | INDEXES | STORAGE | COMMENTS |
  RELOPTIONS | DISTRIBUTION | UPDATE | ILM | ALL }
  ```

- 索引存储参数index_parameters：

  ```
  [ WITH ( {storage_parameter = value} [, ... ] ) ]
  [ USING INDEX TABLESPACE tablespace_name ]
  ```

- partition_less_than_item：

  ```
  PARTITION partition_name VALUES LESS THAN { ( { partition_value | MAXVALUE }[, ... ] ) |
  MAXVALUE } [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW
  ```


版权所有 © 华为云计算技术有限公司

```
AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [TABLESPACE tablespace_name]
```

- partition_start_end_item：

```
PARTITION partition_name {
      {START(partition_value) END (partition_value) EVERY (interval_value)} |
      {START(partition_value) END ({partition_value | MAXVALUE})} |
      {START(partition_value)} |
      {END({partition_value | MAXVALUE})}
} [ ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON ( EXPR )] | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}] [TABLESPACE tablespace_name]
```

- 其中update_expr为：

```
{ CURRENT_TIMESTAMP | LOCALTIMESTAMP | NOW() }
```

参数说明

- **IF NOT EXISTS**

  判断是否存在相同名称的表。

  如果指定IF NOT EXISTS关键字，当已经存在相同名称的表时，在创建新表的时候不抛出错误，而是发出一个notice，告知表已存在。

- **partition_table_name**

  分区表的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

  新表中要创建的字段名。

  取值范围：字符串，要符合**标识符命名规范**。

- **data_type**

  字段的数据类型。

- **COLLATE collation**

  COLLATE子句指定列的排序规则（该列必须是可排列的数据类型）。如果没有指定，则使用默认的排序规则。排序规则可以使用“SELECT * FROM pg_collation;”命令从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。

- **CONSTRAINT constraint_name**

  列约束或表约束的名称。可选的约束子句用于声明约束，新行或者更新的行必须满足这些约束才能成功插入或更新。排序规则可以使用“SELECT * FROM pg_collation”命令从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。

  定义约束有两种方法：

  - 列约束：作为一个列定义的一部分，仅影响该列。不支持为AUTO_INCREMENT | COMMENT 'string' | ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = column_encryption_key, ENCRYPTION_TYPE = encryption_type_value)子句添加约束名。
  - 表约束：不和某个列绑在一起，可以作用于多个列。

- **LIKE source_table [like_option ...]**

  LIKE子句声明一个表，新表自动从这个表里面继承所有字段名及其数据类型和非空约束。

版权所有 © 华为云计算技术有限公司

新表与源表之间在创建动作完毕之后相互独立。在源表做的任何修改都不会继承到新表中，并且也不可能在扫描源表的时候包含新表的数据。

- 字段缺省表达式只有在声明了INCLUDING DEFAULTS之后才会包含进来。默认情况下，不包含缺省表达式，即新表中所有字段的缺省值都是NULL。
- 非空约束将一直继承到新表中，CHECK约束则仅在指定了INCLUDING CONSTRAINTS的时候才会被继承，而其他类型的约束则永远也不会被继承。此规则同时适用于表约束和列约束。

被复制的列和约束并不使用相同的名称进行融合。如果明确的指定了相同的名称或者在另外一个LIKE子句中，将会报错。

- 如果指定了INCLUDING UPDATE，则源表列的ON UPDATE CURRENT_TIMESTAMP属性会继承到新表列中。默认情况下，不继承该属性。
- 如果指定了INCLUDING INDEXES，则源表上的索引也将在新表上创建，默认情况下，不建立索引。新表上的索引所在表空间和源表上的索引所在表空间保持一致。
- 如果指定了INCLUDING STORAGE，则源表列的STORAGE设置也将被继承，默认情况下，不包含STORAGE设置。
- 如果指定了INCLUDING COMMENTS，则源表列、约束和索引的注释也会被继承。默认情况下，不对源表的注释进行继承。
- 如果指定了INCLUDING RELOPTIONS，则源表的存储参数（即源表的WITH子句）也将被继承至新表。默认情况下，不对源表的存储参数进行继承。
- 如果指定了INCLUDING DISTRIBUTION，则新表将继承源表的分布信息，包括分布类型和分布列，同时新表将不能再使用DISTRIBUTE BY子句。默认情况下，不对源表的分布信息进行继承。
- 如果指定了INCLUDING ILM，则源表的ILM策略信息会被继承到新表中，如果需要同时复制源表上的分区对象的ILM策略信息，需要同时指定INCLUDING PARTITION。
- INCLUDING ALL是INCLUDING DEFAULTS、INCLUDING UPDATE、INCLUDING CONSTRAINTS、INCLUDING INDEXES、INCLUDING STORAGE、INCLUDING COMMENTS、INCLUDING RELOPTIONS、INCLUDING DISTRIBUTION和INCLUDING ILM的简写形式。

须知

- 如果源表为ROWID表，指定了INCLUDING ALL/INCLUDING RELOPTIONS后，新表也为ROWID表；仅指定INCLUDING INDEXES时，新表为非ROWID表，且不会复制源表的ROWNO索引。
- CREATE TABLE table_name LIKE source_table;语法仅在MYSQL模式数据库（即sql_compatibility = 'MYSQL'）下，且参数b_format_version值为5.7、b_format_dev_version值为s2时支持。
- 在MYSQL模式数据库下，且参数b_format_version值为5.7、b_format_dev_version值为s2时，不支持指定INCLUDING和EXCLUDING选项，缺省等同于指定INCLUDING ALL。

- **AUTO_INCREMENT [=] value**

  这个子句为自动增长列指定一个初始值，value必须为正数，不得超过2^{127}-1。

版权所有 © 华为云计算技术有限公司

须知

该子句仅在参数sql_compatibility='MYSQL'时有效。

- **COMMENT [=] 'string'**
  - COMMENT [=] 'string'子句表示给表添加注释。
  - 在column_constraint中的COMMENT 'string'表示给列添加注释。
  - 在table_constraint中的COMMENT 'string'表示给主键和唯一键对应的索引添加注释。

  具体请参见**CREATE TABLE**章节中的**COMMENT [=] 'string'**。
- **CHARACTER SET | CHARSET charset**

  指定表字段的字符集。单独指定时会将字段的字符序设置为指定的字符集的默认字符序。
- **COLLATE collation**

  COLLATE子句指定列的排序规则（该列必须是可排列的数据类型）。如果没有指定，则使用默认的排序规则。排序规则可以使用“SELECT * FROM pg_collation”命令从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。对于MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）还支持utf8mb4_bin、utf8mb4_general_ci、utf8mb4_unicode_ci、binary、gbk_chinese_ci、gbk_bin、gb18030_chinese_ci、gb18030_bin字符序。
- **WITH (storage_parameter [= value] [, ...])**

  这个子句为表或索引指定一个可选的存储参数。参数的详细描述如下所示：

  说明

  支持透明加密相关存储参数：enable_tde、encrypt_algo、dek_cipher、key_type、cmk_id。

  详细参数说明请参见**CREATE TABLE**章节。

  - FILLFACTOR

    一个表的填充因子（fillfactor）是一个介于10~100之间的数字。100（完全填充）是默认值。如果指定了较小的填充因子，INSERT操作仅按照填充因子指定的百分率填充表页。每个页上的剩余空间将用于在该页上更新行，这就使得UPDATE有机会在同一页上放置同一条记录的新版本，这比把新版本放置在其他页上更有效。对于一个从不更新的表将填充因子设为100是最佳选择，但是对于频繁更新的表，选择较小的填充因子则更加合适。

    取值范围：10~100
  - ORIENTATION

    决定了表的数据的存储方式。

    取值范围：ROW（缺省值）：表的数据将以行式存储。

    须知

    ORIENTATION不支持修改。
  - STORAGE_TYPE

    指定存储引擎类型，该参数设置成功后就不再支持修改。

版权所有 © 华为云计算技术有限公司

取值范围：

- USTORE，表示表支持Inplace-Update存储引擎。特别需要注意，使用USTORE表，必须要开启track_counts和track_activities参数，否则会引起空间膨胀。
- ASTORE，表示表支持Append-Only存储引擎。

默认值：

不指定时，由参数enable_default_ustore_table决定存储引擎方式，默认是Inplace-Update存储。

- statistic_granularity

  记录该表在分析统计信息时的默认partition_mode，partition_mode说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**，此参数对非分区表设置无效。

  取值范围：请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。

  默认值：AUTO。

- autovacuum_enabled

  自动清理功能是否对该表启用。

  取值范围：on/off

  默认值：on

- autovacuum_vacuum_threshold

  自动清理功能中，指定在该表中触发VACUUM所需的更新或删除的最小元组数（仅对Astore表生效）。

  取值范围：0~2147483647

  默认值：缺省时与GUC参数autovacuum_vacuum_threshold一致。

- autovacuum_analyze_threshold

  自动清理功能中，指定在该表中触发ANALYZE所需的插入、更新或删除的最小元组数。

  取值范围：0~2147483647

  默认值：缺省时与GUC参数autovacuum_analyze_threshold一致。

- autovacuum_vacuum_scale_factor

  自动清理功能中，指定在该表中触发VACUUM所需的插入、更新或删除元组的比例（仅对Astore表生效）。

  取值范围：0.0~100.0

  默认值：缺省时与GUC参数autovacuum_vacuum_scale_factor一致。

- autovacuum_analyze_scale_factor

  自动清理功能中，指定在该表中触发ANALYZE所需的插入、更新或删除元组的比例。

  取值范围：0.0~100.0

  默认值：缺省时与GUC参数autovacuum_analyze_scale_factor一致。

- autovacuum_freeze_min_age

  自动清理功能中，指定在该表参数指定了一个行版本的最小年龄，超过这个年龄的行才会被冻结。

  取值范围：0~576460752303423487

版权所有 © 华为云计算技术有限公司

默认值：缺省时与GUC参数vacuum_freeze_min_age一致。

- autovacuum_freeze_max_age

  自动清理功能中，该表pg_class.relfrozenxid字段在超过多少个事务后，就会强制执行VACUUM操作。即使自动清理被禁用，系统也会启动AUTOVACUUM进程。清理操作还允许从pg_clog/子目录中删除旧文件（仅对ASTORE表生效）。

  取值范围：100000~576460752303423487

  默认值：缺省时与GUC参数autovacuum_freeze_max_age一致。

- autovacuum_freeze_table_age

  自动清理功能中，该表被标记为不需要自动清理时，将保持不变的时间。（仅对Astore表生效）。

  取值范围：0~576460752303423487

  默认值：缺省时与GUC参数vacuum_freeze_table_age一致。

- vacuum_truncate

  表示是否开启表的VACUUM TRUNCATE功能，在VACUUM时截断末尾空页。

  取值范围：on/off

  默认值：on

  说明

  - 该参数仅对Astore生效，对hashbucket表和segment表不生效。
  - 从不支持该参数版本升级至支持该参数版本时，不支持在升级观察期创建、修改该参数。
  - 支持通过toast.vacuum_truncate参数单独控制TOAST表是否截断末尾空页。

- hashbucket

  创建hash bucket存储。本参数仅支持行存表和行存range表。

  取值范围：on/off

  默认值：off

须知

当前版本hashbucket表相关DDL操作性能受限，不建议频繁对hashbucket表进行DDL操作。

- hasrowid

  指定该表是否为rowid表。设置hasrowid = on时，该表会增加rowid系统列和rowno系统列，且会在rowno系统列上关联创建序列和索引。

  取值范围：on、off

  默认值：off

版权所有 © 华为云计算技术有限公司

说明

- 此参数只支持ORA兼容模式数据库（即sql_compatibility = 'ORA'）。
- 此参数不支持本地临时表、Unlogged表、全局临时表、物化视图、hashbucket表、复制表。
- 此参数不支持和hasuids、oids参数同时指定。
- 开启此参数时，不支持定义普通列名为rowno、rowid。
- rowid表数据量不能超过9223372036854775807。

– partition_statistics_min_rows

对分区表执行ANALYZE命令时，用于确定分区级最小采样行数的参数值。

取值范围：[0, 10000]

默认值：50

说明

- 此参数仅对分区表有效。
- 此参数仅在ANALYZE命令的PARTITION_MODE = ALL时，且GUC参数enable_partition_statistics_min_rows启用时生效，PARTITION_MODE说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。
- 上限约束：最终的分区表整表采样行数不超过GUC参数maintenance_work_mem可容纳的行数。

- **[ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [MEDIUM | HIGH] ROW AFTER n { day | month | year } OF NO MODIFICATION [ON (EXPR)] | TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]**

– 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW给行存添加高级压缩中/高级别策略，分区可以单独指定策略，未指定策略时，分区策略继承表的策略。

  ▪ AFTER n { day | month | year } OF NO MODIFICATION ： 表示n天/月/年没有修改的行，高级压缩不支持hour。

  ▪ ON（EXPR）：行级表达式，用于判断行的冷热。

说明

- 在ILM策略的ON（EXPR）行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转换大写。
- 压缩级别是可选项，不指定时默认使用MEDIUM级别。

– 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT给行存添加跨页透明压缩策略，子分区继承分区的策略。

其中，AFTER n { hour | day | month | year } OF NO MODIFICATION表示n小时/天/月/年没有修改的行。

压缩级别不指定时默认为MEDIUM。

– 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS NONE显式指定行存表不添加跨页透明压缩策略。

- **TABLESPACE tablespace_name**

指定新表将要在tablespace_name表空间内创建。如果没有声明，将使用默认表空间。

版权所有 © 华为云计算技术有限公司

- **DISTRIBUTE BY**

  指定表如何在节点之间分布或者复制。

  取值范围及详细信息请参见**CREATE TABLE**章节中的**•DISTRIBUTE BY**。

- **TO { GROUP groupname | NODE (nodename [, ...]) }**

  TO GROUP指定创建表所在的Node Group用。TO NODE主要供内部扩容工具使用，一般用户不应该使用。

- **PARTITION BY RANGE [COLUMNS] (partition_key)**

  创建范围分区。partition_key为分区键的名称。

  说明

  COLUMNS关键字只能在sql_compatibility='MYSQL'时使用，PARTITION BY RANGE COLUMNS语义同与PARTITION BY RANGE等价。

  - 对于从句是VALUES LESS THAN的语法格式：

    范围分区策略的分区键最多支持16列。

    分区键支持的数据类型为：

    - INTEGER TYPES：TINYINT、SMALLINT、INTEGER、BIGINT、DECIMAL、NUMERIC
    - FLOAT TYPES：REAL、FLOAT4、FLOAT8、DOUBLE PRECISION
    - CHARACTER TYPES：CHARACTER VARYING、VARCHAR、CHARACTER、CHAR、BPCHAR、TEXT、NVARCHAR2、NAME
    - DATE/TIME TYPES：TIMESTAMP[(p)] [WITHOUT TIME ZONE]、TIMESTAMP[(p)] [WITH TIME ZONE]、YEAR、DATE、DATEA（仅在ORA模式下生效）

  - 对于从句是START END的语法格式：

    范围分区策略的分区键仅支持1列。

    分区键支持的数据类型为：

    - INTEGER TYPES：TINYINT、SMALLINT、INTEGER、BIGINT、DECIMAL、NUMERIC
    - FLOAT TYPES：REAL、FLOAT4、FLOAT8、DOUBLE PRECISION
    - DATE/TIME TYPES：TIMESTAMP[(p)] [WITHOUT TIME ZONE]、DATE、DATEA（仅在ORA模式下生效）

- **PARTITION partition_name VALUES LESS THAN {({ partition_value | MAXVALUE } [,...]) | MAXVALUE }**

  指定各分区的信息。partition_name为范围分区的名称。partition_value为范围分区的上边界，取值依赖于partition_key的类型。MAXVALUE表示分区的上边界，它通常用于设置最后一个范围分区的上边界。

版权所有 © 华为云计算技术有限公司

须知

- 每个分区都需要指定一个上边界。
- 分区上边界的类型应当和分区键的类型一致。
- 分区列表是按照分区上边界升序排列的，值较小的分区位于值较大的分区之前。
- 不在括号内的MAXVALUE只能在sql_compatibility='MYSQL'时使用，并且只能有一个分区键。

- **PARTITION partition_name {START (partition_value) END (partition_value) EVERY (interval_value)} | {START (partition_value) END (partition_value|MAXVALUE)} | {START(partition_value)} | {END (partition_value | MAXVALUE)}**

  指定各分区的信息，各参数意义如下：

  - partition_name：范围分区的名称或名称前缀，除以下情形外（假定其中的partition_name是p1），均为分区的名称。
    - 若该定义是START+END+EVERY从句，则语义上定义的分区的名称依次为p1_1, p1_2, ...。例如对于定义“PARTITION p1 START(1) END(4) EVERY(1)”，则生成的分区是：[1, 2), [2, 3) 和 [3, 4)，名称依次为p1_1, p1_2和p1_3，即此处的p1是名称前缀。
    - 若该定义是第一个分区定义，且该定义有START值，则范围（MINVALUE, START）将自动作为第一个实际分区，其名称为p1_0，然后该定义语义描述的分区名称依次为p1_1, p1_2, ...。例如对于完整定义“PARTITION p1 START(1), PARTITION p2 START(2)”，则生成的分区是：(MINVALUE, 1), [1, 2) 和 [2, MAXVALUE)，其名称依次为p1_0, p1_1和p2，即此处p1是名称前缀，p2是分区名称。这里MINVALUE表示最小值。
  - partition_value：范围分区的端点值（起始或终点），取值依赖于partition_key的类型，不可是MAXVALUE。
  - interval_value：对[START，END) 表示的范围进行切分，interval_value是指定切分后每个分区的宽度，不可是MAXVALUE；如果（END-START）值不能整除以EVERY值，则仅最后一个分区的宽度小于EVERY值。
  - MAXVALUE：表示最大值，它通常用于设置最后一个范围分区的上边界。

版权所有 © 华为云计算技术有限公司

须知

1. 在创建分区表若第一个分区定义含START值，则范围（MINVALUE，START）将自动作为实际的第一个分区。
2. START END语法需要遵循以下限制：
   - 每个partition_start_end_item中的START值（如果有的话，下同）必须小于其END值；
   - 相邻的两个partition_start_end_item，第一个的END值必须等于第二个的START值；
   - 每个partition_start_end_item中的EVERY值必须是正向递增的，且必须小于（END-START）值；
   - 每个分区包含起始值，不包含终点值，即形如：[起始值，终点值)，起始值是MINVALUE时则不包含；
   - 一个partition_start_end_item创建的每个分区所属的TABLESPACE一样；
   - partition_name作为分区名称前缀时，其长度不要超过57字节，超过时自动截断；
   - 在创建、修改分区表时请注意分区表的分区总数不可超过最大限制（1048575）；
3. 在创建分区表时START END与LESS THAN语法不可混合使用。
4. 即使创建分区表时使用START END语法，备份（gs_dump）出的SQL语句也是VALUES LESS THAN语法格式。

- **PARTITION BY LIST [COLUMNS] (partition_key)**

  创建列表分区。partition_key为分区键的名称。

  COLUMNS关键字只能在sql_compatibility='MYSQL'时使用，“PARTITION BY LIST COLUMNS”语义同“PARTITION BY LIST”。

  - 对于partition_key，列表分区策略的分区键最多支持16列。
  - 对于从句是VALUES [IN] (list_values)的语法格式，list_values中包含了对应分区存在的键值，每个分区的键值数量不超过64个。
  - 从句"VALUES IN"只能在sql_compatibility='MYSQL'时使用，语义同"VALUES"。

  分区键支持的数据类型为：

  - INTEGER TYPES：TINYINT、SMALLINT、INTEGER、BIGINT、NUMERIC
  - CHARACTER TYPES：VARCHAR、CHAR、BPCHAR、NVARCHAR2
  - DATE/TIME TYPES：TIMESTAMP[(p)] [WITHOUT TIME ZONE]、TIMESTAMP[(p)] [WITH TIME ZONE]、YEAR、DATE、DATEA（仅在ORA模式下生效）

  分区个数不能超过1048575个。

- **PARTITION BY HASH(partition_key)**

  创建哈希分区。partition_key为分区键的名称。

  对于partition_key，哈希分区策略的分区键仅支持1列。

  分区键支持的数据类型为：

  - INTEGER TYPES：TINYINT、SMALLINT、INTEGER、BIGINT、NUMERIC
  - CHARACTER TYPES：VARCHAR、CHAR、BPCHAR、TEXT、NVARCHAR2

版权所有 © 华为云计算技术有限公司

- DATE/TIME TYPES：TIME、TIMESTAMP[(p)] [WITHOUT TIME ZONE]、TIMESTAMP[(p)] [WITH TIME ZONE]、YEAR、DATE、DATEA（仅在ORA模式下生效）

分区个数不能超过1048575个。

- **PARTITIONS integer**

  指定分区个数。

  integer为分区数，必须为大于0的整数，且不得大于1048575。

  - 当在RANGE和LIST分区后指定此子句时，必须显式定义每个分区，且定义分区的数量必须与integer值相等。只能在sql_compatibility='MYSQL'时在RANGE和LIST分区后指定此子句。
  - 当在HASH和KEY分区后指定此子句时，若不列出各个分区定义，将自动生成integer个分区，自动生成的分区名为“p+数字”，数字依次为0到integer-1，分区的表空间默认为此表的表空间；也可以显式列出每个分区定义，此时定义分区的数量必须与integer值相等。若既不列出分区定义，也不指定分区数量，将创建唯一一个分区。

- **{ ENABLE | DISABLE } ROW MOVEMENT**

  行迁移开关。

  如果进行UPDATE操作时，更新了元组在分区键上的值，造成了该元组所在分区发生变化，就会根据该开关给出报错信息，或者进行元组在分区间的转移。

  取值范围：

  - ENABLE：行迁移开关打开。
  - DISABLE（缺省值）：行迁移开关关闭。

  在打开行迁移开关情况下，并发UPDATE、DELETE操作可能会报错，原因如下：

  UPDATE和DELETE操作对于旧数据都是标记为已删除。在打开行迁移开关情况下，如果更新分区键时，导致了跨分区更新。内核会把旧分区中旧数据标记为已删除，在新分区中新增加一条数据，无法通过旧数据找到新数据。

  在UPDATE和UPDATE并发、DELETE和DELETE并发、UPDATE和DELETE并发三个并发场景下，如果并发操作同一行数据时，数据跨分区和非跨分区结果有不同的行为。

  a. 对于数据非跨分区结果，第一个操作执行完后，第二个操作不会报错。

     如果第一个操作是UPDATE，第二个操作能成功找到最新的数据，之后对新数据操作。

     如果第一个操作是DELETE，第二个操作看到当前数据已经被删除而且找不到最新数据，就终止操作。

  b. 对于数据跨分区结果，第一个操作执行完后，第二个操作会报错。

     如果第一个操作是UPDATE，由于新数据在新分区中，第二个操作不能成功找到最新的数据，就无法操作，之后会报错。

     如果第一个操作是DELETE，第二个操作看到当前数据已经被删除而且找不到最新数据，但无法判断删除旧数据的操作是UPDATE还是DELETE。如果是UPDATE，报错处理。如果是DELETE，终止操作。为了保持数据的正确性，只能报错处理。

  如果是UPDATE和UPDATE并发，UPDATE和DELETE并发场景，需要串行执行才能解决问题，如果是DELETE和DELETE并发，关闭行迁移开关可以解决问题。

- **NOT NULL**

  字段值不允许为NULL。ENABLE用于语法兼容，可省略。

 版权所有 © 华为云计算技术有限公司

- **NULL**

  字段值允许NULL ，这是缺省。

  这个子句只是为和非标准SQL数据库兼容。不建议使用。

- **CHECK (condition) [NO INHERIT]**

  CHECK约束声明一个布尔表达式，每次要插入的新行或者要更新的行的新值必须使表达式结果为真或未知才能成功，否则会抛出一个异常并且不会修改数据库。

  声明为字段约束的检查约束应该只引用该字段的数值，而在表约束里出现的表达式可以引用多个字段。

  用NO INHERIT标记的约束将不会传递到子表中去。

  ENABLE用于语法兼容，可省略。

- **DEFAULT default_expr**

  DEFAULT子句给字段指定缺省值。该数值可以是任何不含变量的表达式(不允许使用子查询和对本表中的其他字段的交叉引用)。缺省表达式的数据类型必须和字段类型匹配。

  缺省表达式将被用于任何未声明该字段数值的插入操作。如果没有指定缺省值则缺省值为NULL 。

- **ON UPDATE update_expr**

  ON UPDATE子句为字段的一种属性约束。

  当对表中某元组执行UPDATE操作时，若更新字段的新值和表中旧值不相同，则表中该元组上具有该属性且不在更新字段内的字段值自动更新为当前时间戳；若更新字段的新值和表中旧值相同，则表中该元组上具有该属性且不在更新字段内的字段值不变，保持原有值；若具有该属性的字段在更新字段内，则对应这些字段值直接按指定更新的值更新。

  说明

  - 该属性仅支持在MYSQL模式数据库中的5.7版本下指定（即sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s1'）。
  - 语法上update_expr支持CURRENT_TIMESTAMP 、LOCALTIMESTAMP 、NOW()三种关键字，也支持关键字带括号指定或不指定精度。例如：ON UPDATE CURRENT_TIMESTAMP()、ON UPDATE CURRENT_TIMESTAMP(5)、ON UPDATE LOCALTIMESTAMP()、ON UPDATE LOCALTIMESTAMP(6)等。不带括号或空括号时精度为0，其中NOW关键字不支持不带括号。三种关键字互为同义词，属性效果相同。
  - 该属性仅支持在如下类型的列上指定: timestamp、datetime、date、time without time zone、smalldatetime、abstime。
  - CREATE TABLE AS语法不会继承该列属性。
  - CREATE TABLE LIKE语法可通过INCLUDING UPDATE或EXCLUDING UPDATE来选择继承或排除该约束。新增INCLUDING ILM选项复制旧表的ilm策略信息，结合INCLUDING PARTITION选项使用可以复制旧表上分区对象的策略信息。
  - 该属性指定的精度和对应列上类型指定的精度可以不一致，通过该属性更新字段值后显示结果按最小精度显示。例如：CREATE TABLE t1 (col1 timestamp(6) ON UPDATE CURRENT_TIMESTAMP(3)); 若UPDATE语法触发该属性生效，则本次更新后col1字段值小数位显示3位。
  - 该属性和生成列约束不能同时指定同一列。
  - 分区表中的分区键不支持指定该属性。
  - 分布式场景中，分布列和主键不支持指定该属性。

- **AUTO_INCREMENT**

  指定列为自动增长列。

  具体请参见**CREATE TABLE**章节中的**AUTO_INCREMENT**。

版权所有 © 华为云计算技术有限公司

- **UNIQUE [KEY] index_parameters**
  UNIQUE (column_name [, ...]) index_parameters
  UNIQUE约束表示表里的一个字段或多个字段的组合必须在全表范围内唯一。
  对于唯一约束，NULL被认为是互不相等的。
  UNIQUE KEY只能在sql_compatibility='MYSQL'时使用，与UNIQUE语义相同。
- **PRIMARY KEY index_parameters**
  PRIMARY KEY (column_name [, ...]) index_parameters
  主键约束声明表中的一个或者多个字段只能包含唯一的非NULL值。
  一个表只能声明一个主键。
- **DEFERRABLE | NOT DEFERRABLE**
  这两个关键字用于设置约束是否可推迟。不可推迟的约束将在每条命令执行后立即检查。可推迟约束可以延迟到事务结尾通过SET CONSTRAINTS命令检查，缺省值为NOT DEFERRABLE。目前UNIQUE约束和主键约束可以接受该子句，其他约束类型均不可推迟。
- **INITIALLY IMMEDIATE | INITIALLY DEFERRED**
  若约束可推迟，该子句声明检查约束的缺省时间。不可推迟的约束始终为INITIALLY IMMEDIATE且不可声明为INITIALLY DEFERRED。
  - 如果约束是INITIALLY IMMEDIATE（缺省），则在每条语句执行后立即检查。
  - 如果约束是INITIALLY DEFERRED ，则仅在事务结束时进行检查。

  约束检查的时间可以用SET CONSTRAINTS命令修改。
- **USING INDEX TABLESPACE tablespace_name**
  为UNIQUE或PRIMARY KEY约束相关的索引声明一个表空间。如果没有提供这个子句，这个索引将在default_tablespace中创建，如果default_tablespace为空，将使用数据库的缺省表空间。

范围分区

- VALUES LESS THAN

```
--创建表空间。
CREATE TABLESPACE tbs_test_range1_p1 RELATIVE LOCATION 'tbs_test_range1/tablespace_1';
CREATE TABLESPACE tbs_test_range1_p2 RELATIVE LOCATION 'tbs_test_range1/tablespace_2';
CREATE TABLESPACE tbs_test_range1_p3 RELATIVE LOCATION 'tbs_test_range1/tablespace_3';
CREATE TABLESPACE tbs_test_range1_p4 RELATIVE LOCATION 'tbs_test_range1/tablespace_4';

--创建分区表test_range1。
CREATE TABLE test_range1(
    id INT,
    info VARCHAR(20)
) PARTITION BY RANGE (id) (
    PARTITION p1 VALUES LESS THAN (200) TABLESPACE tbs_test_range1_p1,
    PARTITION p2 VALUES LESS THAN (400) TABLESPACE tbs_test_range1_p2,
    PARTITION p3 VALUES LESS THAN (600) TABLESPACE tbs_test_range1_p3,
    PARTITION pmax VALUES LESS THAN (MAXVALUE) TABLESPACE tbs_test_range1_p4
);

--插入1000条数据
INSERT INTO test_range1 VALUES(GENERATE_SERIES(1,1000),'abcd');

--查看p1分区的行数199条，[1,200)。
SELECT COUNT(*) FROM test_range1 PARTITION (p1);
 count
-------
   199
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

--查看p2分区的行数200条，[200,400)。
SELECT COUNT(*) FROM test_range1 PARTITION (p2);
 count
-------
   200
(1 row)

--查看分区信息。
SELECT a.relname, a.boundaries, b.spcname
FROM pg_partition a, pg_tablespace b
WHERE a.reltablespace = b.oid AND a.parentid = 'test_range1'::regclass;
 relname | boundaries |       spcname
---------+------------+---------------------
 p1      | {200}      | tbs_test_range1_p1
 p2      | {400}      | tbs_test_range1_p2
 p3      | {600}      | tbs_test_range1_p3
 pmax    | {NULL}     | tbs_test_range1_p4
(4 rows)

--删除
DROP TABLE test_range1;
DROP TABLESPACE tbs_test_range1_p1;
DROP TABLESPACE tbs_test_range1_p2;
DROP TABLESPACE tbs_test_range1_p3;
DROP TABLESPACE tbs_test_range1_p4;
```

- START END

```
--创建分区表。
CREATE TABLE test_range2(
    id INT,
    info VARCHAR(20)
) PARTITION BY RANGE (id) (
    PARTITION p1 START(1) END(600) EVERY(200),
    PARTITION p2 START(600) END(800),
    PARTITION pmax START(800) END(MAXVALUE)
);

--查看分区信息。
SELECT relname, boundaries FROM pg_partition WHERE parentid = 'test_range2'::regclass AND
parttype = 'p' ORDER BY 1;
 relname | boundaries
---------+------------
 p1_0    | {1}
 p1_1    | {201}
 p1_2    | {401}
 p1_3    | {600}
 p2      | {800}
 pmax    | {NULL}
(6 rows)

--删除。
DROP TABLE test_range2;
```

列表分区

```
--创建列表分区表。
CREATE TABLE test_list ( NAME VARCHAR ( 50 ), area VARCHAR ( 50 ) )
PARTITION BY LIST (area) (
    PARTITION p1 VALUES ('Beijing'),
    PARTITION p2 VALUES ('Shanghai'),
    PARTITION p3 VALUES ('Guangzhou'),
    PARTITION p4 VALUES ('Shenzhen'),
    PARTITION pdefault VALUES (DEFAULT)
);

--插入数据。
INSERT INTO test_list VALUES ('bob', 'Shanghai'),('scott', 'Sichuan');
```


版权所有 © 华为云计算技术有限公司

```
--查询分区数据。
SELECT * FROM test_list PARTITION (p2);
 name |   area
------+----------
 bob  | Shanghai
(1 row)
SELECT * FROM test_list PARTITION (pdefault);
 name  |  area
-------+---------
 scott | Sichuan
(1 row)

--删除。
DROP TABLE test_list;
```

哈希分区

```
--创建哈希分区表，指定分区数。
CREATE TABLE test_hash1(c1 int) PARTITION BY HASH(c1) PARTITIONS 3;

--创建哈希分区表，并指定分区名。
CREATE TABLE test_hash2(c1 int) PARTITION BY HASH(C1)(
    PARTITION pa,
    PARTITION pb,
    PARTITION pc
);

--查看分区信息。
SELECT b.relname AS table_name,
       a.relname AS partition_name
FROM pg_partition a,
     pg_class b
WHERE b.relname LIKE 'test_hash%'
  AND a.parttype = 'p'
  AND a.parentid = b.oid;
 table_name | partition_name
------------+----------------
 test_hash1 | p2
 test_hash1 | p1
 test_hash1 | p0
 test_hash2 | pc
 test_hash2 | pb
 test_hash2 | pa
(6 rows)

--删除。
DROP TABLE test_hash1,test_hash2;
```

相关链接

ALTER TABLE PARTITION，**DROP TABLE**

1.13.9.48 CREATE TABLE PARTITION | SUBPARTITION AS

功能描述

根据查询结果创建分区表。

CREATE TABLE PARTITION | SUBPARTITION AS创建一个分区表并且用来自SELECT命令的结果填充该分区表，数据将会按照指定的分区策略进行分区。该分区表的字段和SELECT输出字段的名称及数据类型相关。用户可以通过指定字段名称列表对SELECT输出字段的名称进行覆盖替换。

版权所有 © 华为云计算技术有限公司

语法格式

- 使用CREATE TABLE PARTITION AS创建一级分区表，并填充子查询数据：

```
CREATE TABLE [ IF NOT EXISTS ] partition_table_name
    [ (column_name [, ...] ) ]
    [ { ENGINE [ = ] { InnoDB | 'InnoDB' | "InnoDB" } } [ ,... ] [ ... ] ]
    [ WITH ( {storage_parameter = value} [, ... ] ) ]
    [ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
 ( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO
MODIFICATION
 | NONE}]
    [ TABLESPACE tablespace_name ]
    [ DISTRIBUTE BY { REPLICATION | { [HASH ] ( column_name ) } } ]
    [ TO { GROUP groupname | NODE ( nodename [, ... ] ) } ]
    PARTITION BY partition_clause partition_definition_list
    AS query
    [ WITH [ NO ] DATA ];
```

- 使用CREATE TABLE SUBPARTITION AS创建二级分区表，并填充子查询数据：

```
CREATE TABLE [ IF NOT EXISTS ] partition_table_name
    [ (column_name [, ...] ) ]
    [ { ENGINE [ = ] { InnoDB | 'InnoDB' | "InnoDB" } } [ ,... ] [ ... ] ]
    [ WITH ( {storage_parameter = value} [, ... ] ) ]
    [ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
 ( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO
MODIFICATION
 | NONE}]
    [ TABLESPACE tablespace_name ]
     PARTITION BY partition_clause
     SUBPARTITION BY subpartition_clause
     partition_subpartition_definition_list
    AS query
    [ WITH [ NO ] DATA ];
```

各个字段详细信息请参见**语法格式**。

参数说明

- **IF NOT EXISTS**

  判断是否存在相同名称的relation。

  如果指定IF NOT EXISTS关键字，创建表前会在当前Schema中查找是否已有名字相同的relation，若存在同名的relation，则不会新建并返回NOTICE提示。未指定IF NOT EXISTS关键字时，若Schema中存在同名relation，返回ERROR告警。

- **partition_table_name**

  要创建的分区表名。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

  可选。新表中要创建的字段名。如果没有指定字段名，那么新表的字段名和SELECT语句输入的字段名一致。

  取值范围：字符串，要符合**标识符命名规范**。

- **ENGINE**

  MYSQL模式下支持，仅语法适配，且只支持设置InnoDB，无实际效果。

- **WITH (storage_parameter [= value] [, ...])**

  为表或索引指定一个可选的存储参数。参数的详细说明如下所示：

版权所有 © 华为云计算技术有限公司

说明

支持透明加密相关存储参数：enable_tde、encrypt_algo、dek_cipher、key_type、cmk_id。

详细参数说明请参见**CREATE TABLE**章节。

- FILLFACTOR

  表的填充因子（fillfactor），一个介于10和100之间的百分数。未指定时默认值为100（完全填充）。在Ustore存储引擎下，该值的默认值为92。如果指定了较小的填充因子，INSERT操作仅按照填充因子指定的百分率填充表页，每个页上的剩余空间将用于在该页上更新行。较小的填充因子对于UPDATE操作，可以在同一页上有机会放置同一条记录的新版本，相较于将新版本的数据放置在其他页上更为有效。对于一个从不更新的表，建议将填充因子设为100，对于频繁更新的表，建议选择较小的填充因子。该参数仅对行存表有效。

  取值范围：10~100

- ORIENTATION

  取值范围：ROW，表的数据将以行式存储。

  默认值：ROW

- autovacuum_enabled

  表示是否开启表的自动清理功能。

  取值范围：on/off

  默认值：on

- autovacuum_vacuum_threshold

  开启自动清理功能时，指定在该表中触发VACUUM所需的更新或删除的最小元组数（仅对Astore表生效）。

  取值范围：0~2147483647

  默认值：未指定时与GUC参数autovacuum_vacuum_threshold一致。

- autovacuum_analyze_threshold

  开启自动清理功能时，指定在该表中触发ANALYZE所需的插入、更新或删除的最小元组数。

  取值范围：0~2147483647

  默认值：未指定时与GUC参数autovacuum_analyze_threshold一致。

- autovacuum_vacuum_scale_factor

  开启自动清理功能时，指定在该表中触发VACUUM所需的插入、更新或删除元组的比例（仅对Astore表生效）。

  取值范围：0.0~100.0

  默认值：未指定时与GUC参数autovacuum_vacuum_scale_factor一致。

- autovacuum_analyze_scale_factor

  开启自动清理功能时，指定在该表中触发ANALYZE所需的插入、更新或删除元组的比例。

  取值范围：0.0~100.0

  默认值：未指定时与GUC参数autovacuum_analyze_scale_factor一致。

- autovacuum_freeze_min_age

  开启自动清理功能时，在该表参数指定了一个行版本的最小范围，超过这个范围的行会被冻结。

版权所有 © 华为云计算技术有限公司

取值范围：0~576460752303423487

默认值：未指定时与GUC参数vacuum_freeze_min_age一致。

- autovacuum_freeze_max_age

  开启自动清理功能时，表pg_class.relfrozenxid字段在超过指定个事务后，就会强制执行VACUUM操作，即使自动清理被禁用，系统也会启动AUTOVACUUM进程。清理操作还允许从pg_clog/子目录中删除旧文件（仅对Astore表生效）。

  取值范围：100000~576460752303423487

  默认值：未指定时与GUC参数autovacuum_freeze_max_age一致。

- autovacuum_freeze_table_age

  开启自动清理功能时，该表被标记为不需要自动清理时，将保持不变的时间（仅对Astore表生效）。

  取值范围：0~576460752303423487

  默认值：未指定时与GUC参数vacuum_freeze_table_age一致。

- vacuum_truncate

  表示是否开启表的VACUUM TRUNCATE功能，在VACUUM时截断末尾空页。

  取值范围：on/off

  默认值：on

  说明

  - 该参数仅对Astore生效，对hashbucket表和segment表不生效。
  - 从不支持该参数版本升级至支持该参数版本时，不支持在升级观察期创建、修改该参数。
  - 支持通过toast.vacuum_truncate参数单独控制TOAST表是否截断末尾空页。

- hashbucket

  创建hash bucket存储。本参数仅支持行存表和行存Range表。

  取值范围：on/off

  默认值：off

须知

当前版本hashbucket表相关DDL操作性能受限，不建议频繁对hashbucket表进行DDL操作。

- hasrowid

  指定该表是否为rowid表。设置hasrowid = on时，该表会增加rowid系统列和rowno系统列，且会在rowno系统列上关联创建序列和索引。

  取值范围：on、off

  默认值：off

版权所有 © 华为云计算技术有限公司

说明

- 此参数只支持ORA兼容模式数据库（即sql_compatibility = 'ORA'）。
- 此参数不支持本地临时表、Unlogged表、全局临时表、物化视图、hashbucket表、复制表。
- 此参数不支持和hasuids参数、oids参数同时指定。
- 开启此参数时，不支持定义普通列名为rowno、rowid。
- rowid表数据量不能超过9223372036854775807。

– partition_statistics_min_rows

对分区表执行ANALYZE命令时，用于确定分区级最小采样行数的参数值。

取值范围：[0, 10000]

默认值：50

说明

- 此参数仅对分区表有效。
- 此参数仅在ANALYZE命令的PARTITION_MODE = ALL时，且GUC参数enable_partition_statistics_min_rows启用时生效，PARTITION_MODE说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。
- 上限约束：最终的分区表整表采样行数不超过GUC参数maintenance_work_mem可容纳的行数。

- **ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [MEDIUM | HIGH] ROW AFTER n { day | month | year } OF NO MODIFICATION [ON (EXPR)] | TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}**

  – 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW为行存添加高级压缩中/高级别策略。

    ▪ AFTER n { day | month | year } OF NO MODIFICATION：表示n天/月/年没有修改的行，高级压缩不支持hour。

    ▪ ON（EXPR）：行级表达式，用于判断行的冷热。

    说明

    - 在ILM策略的ON（EXPR）行级表达式支持的函数中，部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转换大写。
    - 压缩级别是可选项，不指定时默认使用MEDIUM级别。

  – 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT为行存添加跨页透明压缩策略，子分区继承分区的策略。

    AFTER n { hour | day | month | year } OF NO MODIFICATION：表示n小时/天/月/年没有修改的行。

    压缩级别不指定时默认为MEDIUM。

  – 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS NONE显式指定行存表不添加跨页透明压缩策略。

- **TABLESPACE tablespace_name**

  创建新表时指定此关键字，表示新表将要在指定的表空间tablespace_name内创建。如果没有声明，将使用默认表空间。

- **DISTRIBUTE BY**

版权所有 © 华为云计算技术有限公司

详细信息请参见**CREATE TABLE**章节中的**DISTRIBUTE BY**。

- **TO { GROUP groupname | NODE (nodename [, ...]) }**
  指定创建表所在的Node Group。
  该语法仅供内部扩容工具使用，不建议用户直接使用。
- **PARTITION BY**
  partition_clause与partition_definition_list的各字段详细信息请参见**参数说明**。
- **AS query**
  一个SELECT VALUES命令或者一个运行预备好的SELECT或VALUES查询的EXECUTE命令。
- **[WITH [NO] DATA]**
  创建分区表时，是否将查询到的数据进行插入。默认插入，当指定NO时，则表示不插入数据。

示例

- 创建一级分区表，并使用子查询数据填充分区表。

```
-- 创建一个源表t1，并插入3条数据。
gaussdb=# CREATE TABLE t1(a int, b int) DISTRIBUTE BY HASH(a);
gaussdb=# INSERT INTO t1 VALUES(8, 2);
gaussdb=# INSERT INTO t1 VALUES(18, 3);
gaussdb=# INSERT INTO t1 VALUES(28, 4);

-- 使用CREATE TABLE PARTITION AS创建一级分区表，并使用子查询数据填充分区表。
gaussdb=# CREATE TABLE t1_part_dup DISTRIBUTE BY HASH(a) PARTITION BY RANGE(a)
(
    PARTITION p1 VALUES LESS THAN(10),
    PARTITION p2 VALUES LESS THAN(20),
    PARTITION p3 VALUES LESS THAN(MAXVALUE)
) AS SELECT * FROM t1;

gaussdb=# \d+ t1_part_dup;
          Table "create_table_as_partition_by.t1_part_dup"
 Column |  Type   | Modifiers | Storage | Stats target | Description
--------+---------+-----------+---------+--------------+-------------
 a      | integer |           | plain   |              |
 b      | integer |           | plain   |              |
Partition By RANGE(a)
Number of partitions: 3 (View pg_partition to check each partition range.)
Has OIDs: no
Distribute By: HASH(a)
Location Nodes: ALL DATANODES
Options: orientation=row, compression=no

gaussdb=# SELECT a, b FROM t1_part_dup PARTITION(p1) ORDER BY a;
 a | b
---+---
 8 | 2
(1 row)

gaussdb=# SELECT a, b FROM t1_part_dup PARTITION(p2) ORDER BY a;
 a  | b
----+---
 18 | 3
(1 row)

gaussdb=# SELECT a, b FROM t1_part_dup PARTITION(p3) ORDER BY a;
 a  | b
----+---
 28 | 4
(1 row)

gaussdb=# DROP TABLE t1, t1_part_dup;
```


版权所有 © 华为云计算技术有限公司

- 创建二级分区表，并使用子查询数据填充分区表。

```
-- 创建一个源表t1，并插入6条数据。
gaussdb=# CREATE TABLE t1(a int, b int, c int, d varchar, e varchar) PARTITION BY RANGE(a)
SUBPARTITION BY RANGE(b)
(
    PARTITION p1 VALUES LESS THAN(200)
    (SUBPARTITION s1 VALUES LESS THAN (500),
     SUBPARTITION s2 VALUES LESS THAN (MAXVALUE)
    ),
    PARTITION p2 VALUES LESS THAN(500)
    (SUBPARTITION s3 VALUES LESS THAN (500),
     SUBPARTITION s4 VALUES LESS THAN (MAXVALUE)
    ),
    PARTITION p3 VALUES LESS THAN(MAXVALUE)
    (SUBPARTITION s5 VALUES LESS THAN (500),
     SUBPARTITION s6 VALUES LESS THAN (MAXVALUE)
    )
);

gaussdb=# INSERT INTO t1 VALUES (100, 300, 1, 'aA');
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1, 'bB');
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1, 'cC');
gaussdb=# INSERT INTO t1 VALUES (350, 560, 1, 'dD');
gaussdb=# INSERT INTO t1 VALUES (550, 100, 1, 'eE');
gaussdb=# INSERT INTO t1 VALUES (650, 600, 1, 'fF');

-- 使用CREATE TABLE SUBPARTITION AS创建二级分区表，并使用子查询数据填充分区表。
gaussdb=# CREATE TABLE t1_part_dup PARTITION BY RANGE(a) SUBPARTITION BY RANGE(b)
(
    PARTITION p1 VALUES LESS THAN(200)
    (SUBPARTITION s1 VALUES LESS THAN (500),
     SUBPARTITION s2 VALUES LESS THAN (MAXVALUE)
    ),
    PARTITION p2 VALUES LESS THAN(500)
    (SUBPARTITION s3 VALUES LESS THAN (500),
     SUBPARTITION s4 VALUES LESS THAN (MAXVALUE)
    ),
    PARTITION p3 VALUES LESS THAN(MAXVALUE)
    (SUBPARTITION s5 VALUES LESS THAN (500),
     SUBPARTITION s6 VALUES LESS THAN (MAXVALUE)
    )
) AS SELECT * FROM t1;

gaussdb=# \d+ t1_part_dup;
                Table "create_table_as_partition_by.t1_part_dup"
 Column |       Type        | Modifiers | Storage  | Stats target | Description
--------+-------------------+-----------+----------+--------------+-------------
 a      | integer           |           | plain    |              |
 b      | integer           |           | plain    |              |
 c      | integer           |           | plain    |              |
 d      | character varying |           | extended |              |
 e      | character varying |           | extended |              |
Partition By RANGE(a) Subpartition By RANGE(b)
Number of partitions: 3 (View pg_partition to check each partition range.)
Number of subpartitions: 6 (View pg_partition to check each subpartition range.)
Has OIDs: no
Options: orientation=row, compression=no

gaussdb=# SELECT a, b, c, d, e FROM t1_part_dup ORDER BY a;
  a  |  b  | c | d  | e
-----+-----+---+----+---
 100 | 300 | 1 | aA |
 150 | 550 | 1 | bB |
 250 | 150 | 1 | cC |
 350 | 560 | 1 | dD |
 550 | 100 | 1 | eE |
 650 | 600 | 1 | fF |
(6 rows)
```

 版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT a, b, c, d, e FROM t1_part_dup PARTITION(p1) ORDER BY a;
 a  |  b  | c | d  | e
-----+-----+---+----+---
 100 | 300 | 1 | aA |
 150 | 550 | 1 | bB |
(2 rows)

gaussdb=# SELECT a, b, c, d, e FROM t1_part_dup PARTITION(s1) ORDER BY a;
 a  |  b  | c | d  | e
-----+-----+---+----+---
 100 | 300 | 1 | aA |
(1 row)

gaussdb=# DROP TABLE t1, t1_part_dup;
```

相关链接

CREATE TABLE AS，**CREATE TABLE PARTITION**，**CREATE TABLE SUBPARTITION**

1.13.9.49 CREATE TABLE SUBPARTITION

功能描述

创建二级分区表。分区表是把逻辑上的一张表根据某种方案分成几张物理块进行存储，这张逻辑上的表称之为分区表，物理块称之为分区。分区表是一张逻辑表，不存储数据，数据实际是存储在分区上的。对于二级分区表，顶层节点表和一级分区都是逻辑表，不存储数据，只有二级分区（叶子节点）存储数据。

二级分区表的分区方案是由两个一级分区的分区方案组合而来的，一级分区的分区方案请参见**CREATE TABLE PARTITION**章节。

常见的二级分区表组合方案有Range-Range分区、Range-List分区、Range-Hash分区、List-Range分区、List-List分区、List-Hash分区、Hash-Range分区、Hash-List分区、Hash-Hash分区等。目前二级分区仅支持行存表。

注意事项

- 二级分区表有两个分区键，对于Range/List分区键只能支持16列，对于Hash分区键只能支持1列。
- 唯一约束和主键约束的约束键包含所有分区键将为约束创建LOCAL索引，否则创建GLOBAL索引。如果指定创建LOCAL唯一索引，必须包含所有分区键。
- 创建二级分区表时，如果在其一级分区下不显式指定二级分区，会自动创建一个同范围的二级分区。
- 单个表的最大分区数为1048575（包括一级分区和二级分区）。在默认配置下，建议单个表的分区数不超过400个。具体可参考《开发指南》中“特性使用指导 > 分区表 > 分区表介绍 > 分区基本使用 > 创建分区表”章节。
- 二级分区表只支持行存，不支持hashbucket。
- 不支持在rowid/rowno系统列上创建外键约束。
- 不支持cluster。
- 在进行指定分区的查询时，如SELECT * FROM tablename PARTITION/SUBPARTITION (partitionname)。请注意，关键字PARTITION和SUBPARTITION必须准确无误，如果拼写错误，查询不会报错，但这时查询会变为使用别名进行查询，而非指定分区的查询。
- 不支持密态数据库、账本数据库和行级访问控制。

版权所有 © 华为云计算技术有限公司

- 对于二级分区表PARTITION/SUBPARTITION FOR (VALUES)语法，VALUES只能是常量。
- 对于二级分区表PARTITION/SUBPARTITION FOR (VALUES)语法，VALUES在需要数据类型转换时，建议使用强制类型转换，以防隐式类型转换结果与预期不符。
- 创建二级分区表时，RANGE策略的分区定义是VALUES LESS THAN (val1_1)，LIST策略的分区定义是VALUES (val1[, …])，HASH策略的分区无VALUES定义。
- 在为数据对象增加或者变更ILM策略的时候，如果追加了行级表达式，需要注意行表达式目前只支持白名单中列出的函数。具体白名单函数列表请参见**行表达式函数白名单**。

  说明

  在ILM策略的ON(EXPR)行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转大写。

语法格式

```
CREATE TABLE [ IF NOT EXISTS ] subpartition_table_name
{(
{ column_name data_type [ CHARACTER SET | CHARSET charset ] [ COLLATE collation ]
[ column_constraint [ ... ] ]
   | table_constraint
   | LIKE source_table [ like_option [...] ] }[, ... ]
)
   | LIKE source_table }
   [ table_option [ ,... ] [ ... ] ]
   [ WITH ( {storage_parameter = value} [, ... ] ) ]
   [ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION
 | NONE}][ htap_option ]
   [ TABLESPACE tablespace_name ]
   PARTITION BY {RANGE [ COLUMNS ] | LIST [ COLUMNS ] | HASH | KEY} (partition_keys) [ PARTITIONS
integer ]
   SUBPARTITION BY {RANGE | LIST | HASH | KEY} (subpartition_keys) [ SUBPARTITIONS integer ]
   (
    PARTITION partition_name1 [ VALUES LESS THAN { (val1 | MAXVALUE) | MAXVALUE } | VALUES [ IN ]
(val1[, ...]) ]
[ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION
 | NONE}][ htap_option ]
[ TABLESPACE [=] tablespace ]
    [(
     { SUBPARTITION subpartition_name1 [ VALUES LESS THAN (val1_1) | VALUES (val1_1[, ...])]
[ ILM ADD POLICY ROW STORE COMPRESS
 { ADVANCED [ MEDIUM | HIGH ] ROW AFTER n { day | month | year } OF NO MODIFICATION [ ON
( EXPR )]
 | TURBO [ LOW | MEDIUM | HIGH ] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION
 | NONE}][ htap_option ]
[ TABLESPACE [=] tablespace ] } [, ...]
)][, ...]
)[ { ENABLE | DISABLE } ROW MOVEMENT ];
```

- 其中table_option为：

  ```
  { COMMENT [ = ] 'string' |
    AUTO_INCREMENT [ = ] value |
    [ DEFAULT ] CHARACTER SET | CHARSET [ = ] default_charset |
    [ DEFAULT ] COLLATE [ = ] default_collation |
    ENGINE [ = ] { InnoDB | 'InnoDB' | "InnoDB" } }
  ```


版权所有 © 华为云计算技术有限公司

- 其中htap_option为：
```
{ COLVIEW [ PRIORITY { HIGH | LOW | NONE } ] |
  NOCOLVIEW }
```
- 列约束column_constraint：
```
[ CONSTRAINT constraint_name ]
{ NOT NULL |
  NULL |
  CHECK ( expression ) |
  DEFAULT default_expr |
  ON UPDATE update_expr |
  AUTO_INCREMENT |
  COMMENT 'string' |
  UNIQUE [KEY] index_parameters |
  PRIMARY KEY index_parameters |
  REFERENCES reftable [ ( refcolumn ) ] [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]
    [ ON DELETE action ] [ ON UPDATE action ] }
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```
- 表约束table_constraint：
```
[ CONSTRAINT [ constraint_name ] ]
{ CHECK ( expression ) |
  UNIQUE [ index_name ] [ USING method ] ( { column_name [ ASC | DESC ] } [, ... ] )
index_parameters |
  PRIMARY KEY [ USING method ] ( { column_name [ ASC | DESC ] } [, ... ] ) index_parameters |
  FOREIGN KEY [ index_name ] ( column_name [, ... ] ) REFERENCES reftable [ ( refcolumn [, ... ] ) ]
    [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ] [ ON DELETE action ] [ ON UPDATE action ] }
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
{ [ COMMENT 'string' ] [ ... ] }
```
- like选项like_option：
```
{ INCLUDING | EXCLUDING } { DEFAULTS | CONSTRAINTS | INDEXES | STORAGE | COMMENTS |
RELOPTIONS | UPDATE | ALL }
```
- 索引存储参数index_parameters：
```
[ WITH ( {storage_parameter = value} [, ... ] ) ]
[ USING INDEX TABLESPACE tablespace_name ]
```
- 其中update_expr为：
```
{ CURRENT_TIMESTAMP | LOCALTIMESTAMP | NOW() }
```

参数说明

- **IF NOT EXISTS**

  如果已经存在相同名称的表，不会抛出一个错误，而会发出一个通知，告知表关系已存在。

- **subpartition_table_name**

  二级分区表的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

  新表中要创建的字段名。

  取值范围：字符串，要符合**标识符命名规范**。

- **data_type**

  字段的数据类型。

- **COLLATE collation**

  COLLATE子句指定列的排序规则（该列必须是可排列的数据类型）。如果没有指定，则使用默认的排序规则。排序规则可以使用“SELECT * FROM pg_collation;”语句从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。

版权所有 © 华为云计算技术有限公司

- **CONSTRAINT constraint_name**

  列约束或表约束的名称。可选的约束子句用于声明约束，新行或者更新的行必须满足这些约束才能成功插入或更新。

  定义约束有两种方法：

  - 列约束：作为列定义的一部分，仅影响该列。
  - 表约束：作用于多个列。

  须知

  在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）constraint_name为可选项，在其他模式数据库下，必须加上constraint_name。

- **index_name**

  索引名。

  须知

  - index_name仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
  - 对于唯一键约束，constraint_name和index_name同时指定时，索引名为index_name。

- **USING method**

  指定创建索引的方法。

  取值范围请参见**CREATE INDEX**章节中的**USING method**。

  须知

  - USING method仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库下不支持。
  - 在MYSQL模式下，未指定USING method时，对于ASTORE的存储方式，默认索引方法为btree；对于USTORE的存储方式，默认索引方法为ubtree。

- **ASC | DESC**

  ASC表示指定按升序排序（默认）。DESC指定按降序排序。

  须知

  ASC|DESC只在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持，其他模式数据库不支持。

- **LIKE source_table [like_option ...]**

  LIKE子句用于声明一个表，新表将自动从这个表里面继承所有字段名及其数据类型、以及非空约束。

版权所有 © 华为云计算技术有限公司

新表与源表之间在创建动作完毕之后是完全独立的。在源表做的任何修改都不会传播到新表中，同时，在扫描源表的时候也不会包含新表的数据。

- 字段缺省表达式只有在声明了INCLUDING DEFAULTS之后才会包含进来。缺省是不包含缺省表达式的，即新表中所有字段的缺省值都是NULL。
- 如果指定了INCLUDING UPDATE，则源表列的ON UPDATE CURRENT_TIMESTAMP属性会复制到新表列中。默认不复制该属性。
- 非空约束将总是复制到新表中，CHECK约束则仅在指定了INCLUDING CONSTRAINTS的时候才复制，而其他类型的约束则永远也不会被复制。此规则同时适用于表约束和列约束。

被复制的列和约束并不会使用相同的名称进行融合。如果明确的指定了相同的名称或者在另外一个LIKE子句中，将会报错。

- 如果指定了INCLUDING INDEXES，则源表上的索引也将在新表上创建，默认不建立索引。新表上的索引所在的表空间和源表上的索引所在的表空间保持一致。
- 如果指定了INCLUDING STORAGE，则源表列的STORAGE设置也将被复制，默认情况下不包含STORAGE设置。
- 如果指定了INCLUDING COMMENTS，则源表列、约束和索引的注释也会被复制过来。默认情况下，不复制源表的注释。
- 如果指定了INCLUDING RELOPTIONS，则源表的存储参数（即源表的WITH子句）也将复制至新表。默认情况下，不复制源表的存储参数。
- 如果指定了INCLUDING IDENTITY，则源表的identity功能会复制到新表中，并创建一个与源表SEQUENCE参数相同的SEQUENCE。默认情况下，不复制源表的identity功能。
- 如果指定了INCLUDING ILM，则源表的ILM策略信息会被复制到新表中，如果需要同时复制源表上的分区对象的ILM策略信息，需要同时指定INCLUDING PARTITION。
- INCLUDING ALL包含了INCLUDING DEFAULTS、INCLUDING UPDATE、INCLUDING CONSTRAINTS、INCLUDING INDEXES、INCLUDING STORAGE、INCLUDING COMMENTS、INCLUDING PARTITION、INCLUDING RELOPTIONS、INCLUDING IDENTITY和INCLUDING ILM的内容。

- **AUTO_INCREMENT [=] value**

  这个子句为自动增长列指定一个初始值，value必须为正整数，不得超过2^{127}-1。

  须知

  该子句仅在参数sql_compatibility='MYSQL'时有效。

- **COMMENT [=] 'string'**
  - COMMENT [=] 'string'子句表示给表添加注释。
  - 在column_constraint中的COMMENT 'string'表示给列添加注释。
  - 在table_constraint中的COMMENT 'string'表示给主键和唯一键对应的索引添加注释。

  具体请参见**CREATE TABLE**章节中的**COMMENT [=] 'string'**。

- **WITH (storage_parameter [= value] [, ...])**

  这个子句为表或索引指定一个可选的存储参数。参数的详细描述如下所示：

版权所有 © 华为云计算技术有限公司

说明

支持透明加密相关存储参数：enable_tde、encrypt_algo、dek_cipher、key_type、cmk_id。

详细参数说明请参见**CREATE TABLE**章节。

- FILLFACTOR

  一个表的填充因子（fillfactor）是一个介于10和100之间的百分数。在Ustore存储引擎下，该值的默认值为92，在Astore存储引擎下默认值为100（完全填充）。如果指定了较小的填充因子，INSERT操作仅按照填充因子指定的百分率填充表页。每个页上的剩余空间将用于在该页上更新行，这就使得UPDATE有机会在同一页上放置同一条记录的新版本，这比把新版本放置在其他页上更有效。对于一个从不更新的表将填充因子设为100是最佳选择，但是对于频繁更新的表，选择较小的填充因子则更加合适。

  取值范围：10~100

- ORIENTATION

  决定了表的数据的存储方式。

  取值范围：

  - ROW（缺省值）：表的数据将以行式存储。

    须知

    ORIENTATION不支持修改。

- STORAGE_TYPE

  指定存储引擎类型，该参数设置成功后就不再支持修改。

  取值范围：

  - USTORE，表示表支持Inplace-Update存储引擎。

    须知

    使用USTORE表，需开启track_counts和track_activities参数，否则会引起空间膨胀。

  - ASTORE，表示表支持Append-Only存储引擎。

  默认值：

  不指定表时，默认是Inplace-Update存储。

- segment

  使用段页式的方式存储。本参数仅支持行存表。不支持临时表、unlog表。

  取值范围：on/off

  默认值：off

- statistic_granularity

  记录该表在分析统计信息时的默认partition_mode，partition_mode说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**，此参数对非分区表设置无效。

版权所有 © 华为云计算技术有限公司

取值范围：请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。

默认值：AUTO。

- hasrowid

  指定该表是否为rowid表。设置hasrowid = on时，该表会增加rowid系统列和rowno系统列，且会在rowno系统列上关联创建序列和索引。

  取值范围：on、off

  默认值：off

  说明

  - 此参数只支持ORA兼容模式数据库（即sql_compatibility = 'ORA'）。
  - 此参数不支持本地临时表、Unlogged表、全局临时表、物化视图、hashbucket表、复制表。
  - 此参数不支持和hasuids参数、oids参数同时指定。
  - 开启此参数时，不支持定义普通列名为rowno、rowid。
  - rowid表数据量不能超过9223372036854775807。

- vacuum_truncate

  表示是否开启表的VACUUM TRUNCATE功能，在VACUUM时截断末尾空页。

  取值范围：on/off

  默认值：on

  说明

  - 该参数仅对Astore生效，对hashbucket表和segment表不生效。
  - 从不支持该参数版本升级至支持该参数版本时，不支持在升级观察期创建、修改该参数。
  - 支持通过toast.vacuum_truncate参数单独控制TOAST表是否截断末尾空页。

- partition_statistics_min_rows

  对分区表执行ANALYZE命令时，用于确定分区级最小采样行数的参数值。

  取值范围：[0, 10000]

  默认值：50

  说明

  - 此参数仅对分区表有效。
  - 此参数仅在ANALYZE命令的PARTITION_MODE = ALL时，且GUC参数enable_partition_statistics_min_rows启用时生效，PARTITION_MODE说明请参见**ANALYZE | ANALYSE**章节中的**PARTITION_MODE**。
  - 上限约束：最终的分区表整表采样行数不超过GUC参数maintenance_work_mem可容纳的行数。

- **[ILM ADD POLICY ROW STORE COMPRESS { ADVANCED [MEDIUM | HIGH] ROW AFTER n { day | month | year } OF NO MODIFICATION [ON (EXPR)] | TURBO [LOW | MEDIUM | HIGH] UNIT AFTER n { hour | day | month | year } OF NO MODIFICATION | NONE}]**

  - 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS ADVANCED [MEDIUM | HIGH] ROW给行存添加高级压缩中/高级别策略，子分区可以单独指定策略，子分区未指定策略时，如果分区已指定策略，子分区继承分区的策略，如果分区未指定策略，子分区策略将继承表的策略。

版权所有 © 华为云计算技术有限公司

- AFTER n { day | month | year } OF NO MODIFICATION ： 表示n天/月/年没有修改的行，高级压缩不支持hour。
- ON（EXPR）：行级表达式，用于判断行的冷热。

说明

- 在ILM策略的ON（EXPR）行级表达式支持的函数中，有部分函数的输出可能会受兼容性参数影响。例如，upper函数在MYSQL兼容模式下设置b_format_version='5.7'和b_format_dev_version='s2'后，将无法转换大写。
- 压缩级别是可选项，不指定时默认使用MEDIUM级别。

– 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS TURBO HIGH UNIT给行存添加跨页透明压缩策略，子分区继承分区的策略。

其中，AFTER n { hour | day | month | year } OF NO MODIFICATION表示n小时/天/月/年没有修改的行。

压缩级别不指定时默认为MEDIUM。

– 创建新表时，可以调用ILM ADD POLICY ROW STORE COMPRESS NONE显式指定行存表不添加跨页透明压缩策略。

- **TABLESPACE tablespace_name**

  指定新表将要在tablespace_name表空间内创建。如果没有声明，将使用默认表空间。

- **PARTITION BY {RANGE [COLUMNS] | LIST [COLUMNS] | HASH | KEY} (partition_keys)**
  – 对于partition_keys，对于Range/List分区键只能支持16列，对于Hash分区键只能支持1列。
  – 分区键支持的数据类型和一级分区表约束保持一致。
  – COLUMNS关键字只能在sql_compatibility='MYSQL'时使用，只能加在RANGE或LIST之后，“RANGE COLUMNS” 语义同 “RANGE”，“LIST COLUMNS” 语义同 “LIST”。
  – KEY关键字只能在sql_compatibility='MYSQL'时使用，KEY与HASH同义。

- **SUBPARTITION BY {RANGE | LIST | HASH | KEY} (subpartition_keys)**
  – 对于subpartition_keys，对于Range/List分区键只能支持16列，对于Hash分区键只能支持1列。
  – 分区键支持的数据类型和一级分区表约束保持一致。
  – KEY关键字只能在sql_compatibility='MYSQL'时使用，KEY与HASH同义。

- **PARTITIONS integer**

  指定分区个数。

  integer为分区数，必须为大于0的整数，且不得大于1048575。

  – 当在RANGE和LIST分区后指定此子句时，必须显式定义每个分区，且定义分区的数量必须与integer值相等。只能在sql_compatibility='MYSQL'时在RANGE和LIST分区后指定此子句。
  – 当在HASH和KEY分区后指定此子句时，若不列出各个分区定义，将自动生成integer个分区，自动生成的分区名为“p+数字”，数字依次为0到integer-1，分区的表空间默认为此表的表空间；也可以显式列出每个分区定义，此时定义分区的数量必须与integer值相等。若既不列出分区定义，也不指定分区数量，将创建唯一一个分区。

版权所有 © 华为云计算技术有限公司

- **SUBPARTITIONS integer**

  指定二级分区数量。

  integer为二级分区个数，必须为大于0的整数，且不得大于1048575。

  – 只能在HASH和KEY二级分区后指定此子句。

    - 若不列出各个二级分区定义，将在每个一级分区内自动生成integer个二级分区，自动生成的二级分区名为“一级分区名+sp+数字”，数字依次为0到integer-1，分区的表空间默认为此表的表空间。
    - 也可以列出每个二级分区定义，此时二级分区的数量必须与integer值相等。
    - 若既不列出每个二级分区定义，也不指定二级分区数量，将创建唯一一个二级分区。

- **{ ENABLE | DISABLE } ROW MOVEMENT**

  行迁移开关。

  如果进行UPDATE操作时，更新了元组在分区键上的值，造成了该元组所在分区发生变化，就会根据该开关给出报错信息，或者进行元组在分区间的转移。

  取值范围：

  – ENABLE：行迁移开关打开。

  – DISABLE（缺省值）：行迁移开关关闭。

  在打开行迁移开关情况下，并发UPDATE、DELETE操作可能会报错，原因如下：

  UPDATE和DELETE操作对于旧数据都是标记为已删除。在打开行迁移开关情况下，如果更新分区键时，导致了跨分区更新，内核会把旧分区中旧数据标记为已删除，在新分区中新增加一条数据，无法通过旧数据找到新数据。

  在以下三个并发场景下，UPDATE和UPDATE并发、DELETE和DELETE并发和UPDATE和DELETE并发，如果并发操作同一行数据时，数据跨分区和非跨分区结果有不同的行为。

  a. 对于数据非跨分区结果，第一个操作执行完后，第二个操作不会报错。

     - 如果第一个操作是UPDATE，第二个操作能成功找到最新的数据，之后对新数据操作。
     - 如果第一个操作是DELETE，第二个操作看到当前数据已经被删除而且找不到最新数据，就终止操作。

  b. 对于数据跨分区结果，第一个操作执行完后，第二个操作会报错。

     - 如果第一个操作是UPDATE，由于新数据在新分区中，第二个操作不能成功找到最新的数据，就无法操作，之后会报错。
     - 如果第一个操作是DELETE，第二个操作当前数据已经被删除而且找不到最新数据，但无法判断删除旧数据的操作是UPDATE还是DELETE。如果是UPDATE，报错处理。如果是DELETE，终止操作。为了保持数据的正确性，只能报错处理。

  如果是UPDATE和UPDATE并发，UPDATE和DELETE并发场景，需要串行执行才能解决问题，如果是DELETE和DELETE并发，关闭行迁移开关可以解决问题。

- **NOT NULL**

  字段值不允许为NULL。ENABLE用于语法兼容，可省略。

 版权所有 © 华为云计算技术有限公司

- **NULL**

  字段值允许NULL ，缺省值。

  该子句只是为和非标准SQL数据库兼容。不建议使用。

- **CHECK (condition) [NO INHERIT]**

  CHECK约束声明一个布尔表达式，每次要插入的新行或者要更新的行的新值必须使表达式结果为真或未知才能成功，否则会抛出一个异常并且不会修改数据库。

  声明为字段约束的检查约束应该只引用该字段的数值，而在表约束里出现的表达式可以引用多个字段。

  用NO INHERIT标记的约束将不会传递到子表中去。

  ENABLE用于语法兼容，可省略。

- **DEFAULT default_expr**

  DEFAULT子句给字段指定缺省值。该数值可以是任何不含变量的表达式(不允许使用子查询和对本表中的其他字段的交叉引用)。缺省表达式的数据类型必须和字段类型匹配。

  缺省表达式将被用于任何未声明该字段数值的插入操作。如果没有指定缺省值则缺省值为NULL 。

- **ON UPDATE update_expr**

  ON UPDATE子句为字段的一种属性约束。

  当对表中某元组执行UPDATE操作时，若更新字段的新值和表中旧值不相同，则表中该元组上具有该属性且不在更新字段内的字段值自动更新为当前时间戳；若更新字段的新值和表中旧值相同，则表中该元组上具有该属性且不在更新字段内的字段值不变，保持原有值；若具有该属性的字段在更新字段内，则对应这些字段值直接按指定更新的值更新。

  说明

  - 该属性仅支持在MYSQL模式库中的5.7版本下指定（即sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s1'）。
  - 语法上update_expr支持CURRENT_TIMESTAMP 、LOCALTIMESTAMP 、NOW()三种关键字，也支持关键字带括号指定或不指定精度。例如：ON UPDATE CURRENT_TIMESTAMP()、ON UPDATE CURRENT_TIMESTAMP(5)、ON UPDATE LOCALTIMESTAMP()、ON UPDATE LOCALTIMESTAMP(6)等。不带括号或空括号时精度为0，其中NOW关键字不支持不带括号。三种关键字互为同义词，属性效果相同。
  - 该属性仅支持在如下类型的列上指定: timestamp、datetime、date、time without time zone、smalldatetime、abstime。
  - CREATE TABLE AS语法不会继承该列属性。
  - CREATE TABLE LIKE语法可通过INCLUDING UPDATE或EXCLUDING UPDATE来选择继承或排除该约束。新增INCLUDING ILM选项复制旧表的ilm策略信息，结合INCLUDING PARTITION选项使用可以复制旧表上分区对象的策略信息。
  - 该属性指定的精度和对应列上类型指定的精度可以不一致，通过该属性更新字段值后显示结果按最小精度显示。例如：CREATE TABLE t1 (col1 timestamp(6) ON UPDATE CURRENT_TIMESTAMP(3)); 若UPDATE语法触发该属性生效，则本次更新后col1字段值小数位显示3位。
  - 该属性和生成列约束不能同时指定同一列。
  - 分区表中的分区键不支持指定该属性。

- **AUTO_INCREMENT**

  指定列为自动增长列。

  请参见**CREATE TABLE**章节中的**AUTO_INCREMENT**。

版权所有 © 华为云计算技术有限公司

- **UNIQUE [KEY] index_parameters**
  UNIQUE (column_name [, ...]) index_parameters
  UNIQUE约束表示表里的一个字段或多个字段的组合必须在全表范围内唯一。
  对于唯一约束，NULL被认为是互不相等的。
  UNIQUE KEY只能在sql_compatibility='MYSQL'时使用，与UNIQUE语义相同。
- **PRIMARY KEY index_parameters**
  PRIMARY KEY (column_name [, ...]) index_parameters
  主键约束声明表中的一个或者多个字段只能包含唯一的非NULL值。
  一个表只能声明一个主键。
- **DEFERRABLE | NOT DEFERRABLE**
  这两个关键字用于设置约束是否可推迟。不可推迟的约束将在每条命令执行后立即检查。可推迟约束可以延迟到事务结尾通过SET CONSTRAINTS命令检查，缺省值为NOT DEFERRABLE。目前UNIQUE约束、主键约束、外键约束可以接受该子句，其他约束类型均不可推迟。
- **INITIALLY IMMEDIATE | INITIALLY DEFERRED**
  若约束可推迟，该子句声明检查约束的缺省时间。不可推迟的约束始终为INITIALLY IMMEDIATE且不可声明为INITIALLY DEFERRED。
  - 如果约束是INITIALLY IMMEDIATE（缺省），则在每条语句执行后立即检查。
  - 如果约束是INITIALLY DEFERRED ，则仅在事务结束时进行检查。

  约束检查的时间可以用SET CONSTRAINTS命令修改。
- **USING INDEX TABLESPACE tablespace_name**
  为UNIQUE或PRIMARY KEY约束相关的索引声明一个表空间。如果没有提供这个子句，这个索引将在default_tablespace中创建，如果default_tablespace为空，将使用数据库的缺省表空间。

示例

- **创建二级分区表示例**

```
--创建二级分区表tbl_list_list，一级分区和二级分区类型都是LIST。
gaussdb=# CREATE TABLE tbl_list_list(
    sal_year    varchar(4)  NOT NULL,
    area_id     char(5)     NOT NULL,
    emp_id      char(5)     NOT NULL,
    sales_amt   int
) PARTITION BY LIST (sal_year) SUBPARTITION BY LIST(area_id)(
    PARTITION P_2019 VALUES ('2019')(
        SUBPARTITION p_2019_01001 VALUES ('01001'),
        SUBPARTITION p_2019_01002 VALUES ('01002'),
        SUBPARTITION p_2019_01003 VALUES ('01003')
    ),
    PARTITION p_2020 VALUES ('2020')(
        SUBPARTITION p_2020_01001 VALUES ('01001'),
        SUBPARTITION p_2020_01002 VALUES ('01002'),
        SUBPARTITION p_2020_01003 VALUES ('01003')
    )
);

--创建二级分区表tbl_range_list,其一级分区为RANGE类型，二级分区为list类型。
gaussdb=# CREATE TABLE tbl_range_list(
    sal_date    varchar(6)  NOT NULL,
    area_id     char(5)     NOT NULL,
    emp_id      char(5)     NOT NULL,
    sales_amt   int
```


版权所有 © 华为云计算技术有限公司

```
) PARTITION BY RANGE (sal_date) SUBPARTITION BY LIST(area_id)(
   PARTITION p_201901 VALUES LESS THAN (201902)(
      SUBPARTITION p_201901_01001 VALUES ('01001'),
      SUBPARTITION p_201901_01002 VALUES ('01002'),
      SUBPARTITION p_201901_01003 VALUES ('01003')
   ),
   PARTITION p_201902 VALUES LESS THAN (201903)(
      SUBPARTITION p_201902_01001 VALUES ('01001'),
      SUBPARTITION p_201902_01002 VALUES ('01002'),
      SUBPARTITION p_201902_01003 VALUES ('01003')
   )
);

--创建多列分区键的二级分区表multi_range_list,其一级分区为RANGE类型，二级分区为list类型。
gaussdb=# CREATE TABLE multi_range_list(c1 int, c2 int, c3 int)
PARTITION BY RANGE (c1,c2) SUBPARTITION BY LIST (c2, c3)
(
   PARTITION P_RANGE1 VALUES LESS THAN (10,10)
   (
      SUBPARTITION P_RANGE1_LIST1 VALUES ((1,1),(2,2)),
      SUBPARTITION P_RANGE1_LIST2 VALUES ((5,5),(6,6)),
      SUBPARTITION P_RANGE1_LIST3 VALUES ((8,8),(9,9))
   ),
   PARTITION P_RANGE2 VALUES LESS THAN (20,20)
   (
      SUBPARTITION P_RANGE2_LIST1 VALUES ((11,11),(12,12)),
      SUBPARTITION P_RANGE2_LIST2 VALUES ((15,15),(16,16)),
      SUBPARTITION P_RANGE2_LIST3 VALUES ((18,18),(19,19))
   ),
   PARTITION P_RANGE3 VALUES LESS THAN (30,30)
   (
      SUBPARTITION P_RANGE3_LIST1 VALUES ((21,21),(22,22)),
      SUBPARTITION P_RANGE3_LIST2 VALUES ((25,25),(26,26)),
      SUBPARTITION P_RANGE3_LIST3 VALUES ((28,28),(29,29))
   )
);
```

- **对二级分区表进行DML指定分区操作**
  - INSERT

    ```
    --指定一级分区插入数据。
    gaussdb=# INSERT INTO tbl_range_list PARTITION(p_201901) VALUES('201901', '01001', '0001',
    75000 );

    --实际分区和指定分区不一致，报错。
    gaussdb=# INSERT INTO tbl_range_list PARTITION(p_201902) VALUES('201901', '01001', '0002',
    6000);
    ERROR:  inserted partition key does not map to the table partition
    DETAIL:  N/A.

    --指定二级分区插入数据。
    gaussdb=# INSERT INTO tbl_range_list SUBPARTITION(p_201902_01001) VALUES('201902',
    '01001', '0002', 8000);
    ```

  - SELECT

    ```
    --指定分区查询数据。
    gaussdb=# SELECT * FROM tbl_range_list PARTITION(p_201902);
     sal_date | area_id | emp_id | sales_amt
    ----------+---------+--------+-----------
     201902   | 01001   | 0002   |      8000
    (1 row)

    gaussdb=# SELECT * FROM tbl_range_list SUBPARTITION(p_201901_01001);
     sal_date | area_id | emp_id | sales_amt
    ----------+---------+--------+-----------
     201901   | 01001   | 0001   |     75000
    (1 row)
    ```

  - UPDATE

    ```
    --指定分区更新数据。
    gaussdb=# UPDATE tbl_range_list PARTITION(p_201901) SET sales_amt = 7000;
    ```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * FROM tbl_range_list;
 sal_date | area_id | emp_id | sales_amt
----------+---------+--------+-----------
 201901   | 01001   | 0001   |      7000
 201902   | 01001   | 0002   |      8000
(2 rows)

gaussdb=# UPDATE tbl_range_list SUBPARTITION FOR('201902','01001') SET sales_amt = 6000;

gaussdb=# SELECT * FROM tbl_range_list;
 sal_date | area_id | emp_id | sales_amt
----------+---------+--------+-----------
 201901   | 01001   | 0001   |      7000
 201902   | 01001   | 0002   |      6000
(2 rows)
```

- DELETE

```
--指定分区删除数据。
gaussdb=# DELETE FROM tbl_range_list PARTITION (p_201901);
DELETE 1

gaussdb=# DELETE FROM tbl_range_list SUBPARTITION (p_201902_01001);
DELETE 1

gaussdb=# DELETE FROM tbl_range_list SUBPARTITION FOR ('201901','01002');
DELETE 0

--参数sql_compatibility='MYSQL'时，可指定多分区删除数据。
gaussdb=# CREATE DATABASE db DBCOMPATIBILITY 'MYSQL';
gaussdb=# \c db
db=# CREATE TABLE range_list
(
    month_code VARCHAR2 ( 30 ) NOT NULL ,
    dept_code  VARCHAR2 ( 30 ) NOT NULL ,
    user_no    VARCHAR2 ( 30 ) NOT NULL ,
    sales_amt  int
)
PARTITION BY RANGE (month_code) SUBPARTITION BY LIST (dept_code)
(
  PARTITION p_201901 VALUES LESS THAN( '201903' )
  (
    SUBPARTITION p_201901_a VALUES ('1'),
    SUBPARTITION p_201901_b VALUES ('2')
  ),
  PARTITION p_201902 VALUES LESS THAN( '201910' )
  (
    SUBPARTITION p_201902_a VALUES ('1'),
    SUBPARTITION p_201902_b VALUES ('2')
  )
);

db=# DELETE FROM range_list AS t PARTITION (p_201901_a, p_201901);
DELETE 0

--删除数据库(根据实际情况替换数据库名)。
db=# \c postgres
gaussdb=# DROP DATABASE db;

--删除表。
gaussdb=# DROP TABLE tbl_list_list;
gaussdb=# DROP TABLE tbl_range_list;
gaussdb=# DROP TABLE multi_range_list;
```

相关链接

ALTER TABLE SUBPARTITION

版权所有 © 华为云计算技术有限公司

1.13.9.50 CREATE TEXT SEARCH CONFIGURATION

功能描述

创建自定义的文本搜索配置。一个文本搜索配置指定一个文本搜索解析器以及对应的词典列表。

须知

该功能为内部使用功能，不建议用户直接使用。

注意事项

- 如果仅声明PARSER，那么新创建的文本搜索配置没有从字符串类型到词典的映射， 须调用ALTER TEXT SEARCH CONFIGURATION命令创建词典映射使配置生效。如果声明了COPY选项，会自动复制指定的文本搜索配置的解析器、映射、配置选项等信息。
- 若指定模式名称，文本搜索配置会在指定的模式中创建，否则会在当前模式中创建。
- 创建自定义文本搜索配置的用户将自动成为其所有者。
- PARSER和COPY选项互相排斥，因为当一个现有的文本搜索配置被复制，其文本搜索解析器也会被复制。

语法格式

```
CREATE TEXT SEARCH CONFIGURATION name
    ( {PARSER = parser_name | COPY = source_config} )
    [ WITH ( {configuration_option = value} [, ...] )];
```

参数说明

- **name**

  创建的文本搜索配置的名称。可以使用模式进行具体指定。

- **parser_name**

  用于该文本搜索配置的文本搜索解析器的名称。

- **source_config**

  要复制的现有文本搜索配置的名称。

- **configuration_option**

  文本搜索配置的配置参数。

  取值范围：目前支持default、ngram两种类型的解析器，具体配置参数configuration_option如表1-399所示。

  表 1-399 不同类型解析器对应的配置参数

解析器	配置参数	参数描述	取值范围
default	-	-	-

版权所有 © 华为云计算技术有限公司

解析器	配置参数	参数描述	取值范围
ngram	gram_size	分词长度。	正整数，1~4 默认值：2
	punctuation_ignore	是否忽略标点符号。	• true（默认值）：忽略标点符号。 • false：不忽略标点符号。
	grapsymbol_ignore	是否忽略图形化字符。	• true：忽略图形化字符。 • false（默认值）：不忽略图形化字符。

示例

```
--创建文本搜索配置ngram2。
gaussdb=# CREATE TEXT SEARCH CONFIGURATION ngram2 (PARSER=ngram) WITH (GRAM_SIZE = 2,
GRAPSYMBOL_IGNORE = false);

--创建文本搜索配置ngram3。
gaussdb=# CREATE TEXT SEARCH CONFIGURATION ngram3 (COPY=ngram2) WITH (GRAM_SIZE = 2,
GRAPSYMBOL_IGNORE = false);

--添加类型映射。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION ngram2 ADD MAPPING FOR multisymbol WITH simple;

--创建用户joe。
gaussdb=# CREATE USER joe IDENTIFIED BY '********';

--修改文本搜索配置的所有者。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION ngram2 OWNER TO joe;

--修改文本搜索配置的schema。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION ngram2 SET SCHEMA joe;

--重命名文本搜索配置。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION joe.ngram2 RENAME TO ngram_2;

--删除类型映射。
gaussdb=# ALTER TEXT SEARCH CONFIGURATION joe.ngram_2 DROP MAPPING IF EXISTS FOR
multisymbol;

--删除文本搜索配置。
gaussdb=# DROP TEXT SEARCH CONFIGURATION joe.ngram_2;
gaussdb=# DROP TEXT SEARCH CONFIGURATION ngram3;

--删除Schema及用户joe。
gaussdb=# DROP SCHEMA IF EXISTS joe CASCADE;
gaussdb=# DROP ROLE IF EXISTS joe;
```

1.13.9.51 CREATE TEXT SEARCH DICTIONARY

功能描述

创建一个新的全文检索词典。词典是一种指定在全文检索时识别特定词并处理的方法。

版权所有 © 华为云计算技术有限公司

须知

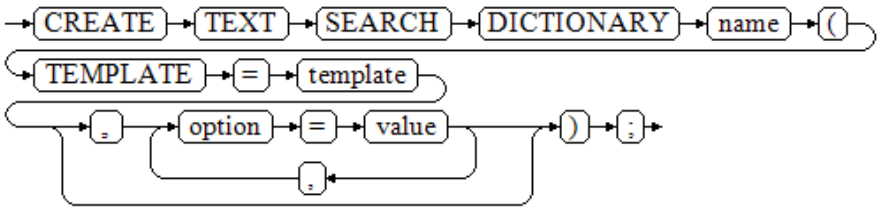
该功能为内部使用功能，不建议用户直接使用。

注意事项

- 具有SYSADMIN权限的用户可以执行创建词典操作，创建该词典的用户自动成为其所有者。
- 词典的创建依赖于预定义模板（在系统表pg_ts_template中定义），当前支持创建Simple、Ispell、Synonym、Thesaurus、Snowball和Tokenweight词典，不同类型的词典可以完成不同的任务。
- 临时模式（pg_temp）下不允许创建词典。
- 创建或修改词典操作之后，当再次对用户新创建的词典的FILEPATH文件进行修改时，需使用ALTER语句更新对应词典的FILEPATH文件。

语法格式

```
CREATE TEXT SEARCH DICTIONARY name (
    TEMPLATE = template
    [, { option = value } [, ... ]]
);
```



参数说明

- **name**

  要创建的词典的名称，可以使用模式进行具体指定，未指定模式时，将在当前模式中创建。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

- **template**

  创建词典使用的模板。

  取值范围：系统表pg_ts_template中预定义的模板，包括：

  - Simple：简单词典/停用词词典，该词典模板将token转换为小写形式，并且根据停用词文件对其进行检查。如果该token在停用词文件中可以匹配，则返回一个空数组，并对该token进行忽略；否则，将该token按照小写形式进行标准化处理。Simple词典中也可以将非停用词设置为未识别，即允许非停用词传递给词典列表中的下一个词典。
  - Synonym：同义词词典，该词典模板用于将单词替换为同义词。
  - Thesaurus：分类词典（可简写为TZ），分类词典是一个词的集合，包括词与短语之间的联系，即广义词、狭义词、首选词、非首选词以及相关词等。分类词典可以使用一个首选词替换所有非首选词并且可选择性地保留原始术语用于索引。
  - Ispell：Ispell词典模板为支持多语言词法的词典，可以将一个词的不同语言学的形式标准化成相同的词位。例如，一个英语Ispell词典可以匹配搜索词bank

版权所有 © 华为云计算技术有限公司

的词尾变化和词形变化，例如：banking、banked、banks、banks'和bank's。

- Snowball：Snowball词典对许多语言提供词干分析算法，每一个词干分析算法可以识别不同语言中的拼写规则，可根据指定的LANGUAGE参数选择具体语言的词干解析器。Snowball词典中，可以选择指定一个停用词文件STOPWORDS，对需要被消除的词进行过滤输出。
- Tokenweight：词权重干预词典，用于调整自然语言处理或信息检索中特定词汇权重的机制，通过人工干预优化算法对关键词的重要性进行判断。

- **option**

  参数名。与template对应，不同的词典模板具有不同的参数列表：

  - Simple词典对应的option：
    - STOPWORDS

      停用词词典定义文件名，默认后缀名为stop。

      文件格式为一组word列表，每行定义一个停用词。Simple词典处理时，停用词词典文件中的空行和空格会被忽略，并将停用词词组转换为小写形式。

    - ACCEPT

      是否将非停用词设置为已识别。默认值为true。

      当Simple词典设置参数ACCEPT=true时，将不会传递任何token给后继词典，此时建议将Simple词典放置在词典列表的最后。反之，当Simple词典设置参数ACCEPT=false时，会将token传递给后继词典，如果Simple词典放置在词典列表的最后，ACCEPT不生效，此时不建议将该Simple词典放置在词典列表的最后。

    - FILEPATH

      Simple词典定义文件所在目录。目录可以指定为本地目录，格式为“file://absolute_path”。默认值为预定义词典文件所在目录。FILEPATH参数必须和STOPWORDS参数同时指定，不允许单独指定。
  - Synonym词典对应的option：
    - SYNONYM

      同义词词典定义文件名，默认后缀名为syn。

      文件格式为一组同义词列表，每行格式为“token synonym”，token和其对应的synonym以空格相连。

    - CASESENSITIVE

      设置是否大小写敏感，默认值为false，此时同义词词典文件中的token和synonym均会转为小写形式。如果设置为true，则不会进行小写转换。

    - FILEPATH

      Synonym词典定义文件所在目录。目录可以指定为本地目录，格式为“file://absolute_path”。默认值为预定义词典文件所在目录。
  - Thesaurus词典对应的option：
    - DICTFILE

      分类词典定义文件名，默认后缀名为ths。

      文件格式为一组同义词列表，每行格式为“sample words : indexed words”，格式中间的冒号（:）作为短语和其替换词间的分隔符。

版权所有 © 华为云计算技术有限公司

Thesaurus词典处理时，如果有多个匹配的sample words，将选择最长匹配输出。

- DICTIONARY

  用于词规范化的子词典名，必须指定且仅能定义一个。该词典必须是已经存在的，在检查短语匹配之前使用，用于识别和规范输入文本。

  如果子词典无法识别输入词，将会报错，需移除该词或更新子词典使其可以识别。对于替换词（indexed words），可在indexed words的开头添加一个星号（*）用于跳过子词典对该indexed words的识别，但对于所有的词（sample words），必须可以被子词典识别。

  如果Thesaurus词典定义文件DICTFILE的sample words中含有子词典中定义的停用词，需要用问号（?）替代停用词。假设a和the是子词典中所定义的停用词，如下：

  ```
  ? one ? two : swsw
  ```

  上述同义词组定义会匹配"a one the two"以及"the one a two"，这两个短语均会被swsw替代输出。

- FILEPATH

  Thesaurus词典定义文件所在目录。目录可以指定为本地目录，格式为“file://absolute_path”。默认值为预定义词典文件所在目录。

– Ispell词典对应的option：

- DICTFILE

  词典定义文件名，默认后缀名为dict。

- AFFFILE

  词缀文件名，默认后缀名为affix。

- STOPWORDS

  停用词文件名，默认后缀名为stop，文件格式要求与Simple词典的停用词文件相同。

- FILEPATH

  Ispell词典定义文件所在目录。可以指定为本地目录，格式为“file://absolute_path”。默认值为预定义词典文件所在目录。

– Snowball词典对应的option：

- LANGUAGE

  语言名，标识使用哪种语言的词干分析算法。算法按照对应语言中的拼写规则，缩减输入词的常见变体形式为一个基础词或词干。

- STOPWORDS

  停用词表文件名，默认后缀名为stop，文件格式要求与Simple词典的停用词文件相同。

- FILEPATH

  Snowball词典定义文件所在目录。可以指定为本地目录，格式为“file://absolute_path”。默认值为预定义词典文件所在目录。FILEPATH参数必须和STOPWORDS参数同时指定，不允许单独指定。

– Tokenweight词典对应的option：

版权所有 © 华为云计算技术有限公司

- TOKENFILE

  词权重定义文件名，默认后缀名为tokw。

  文件格式为一组词权重与同义词相似度的列表，每行可以表示词权重或者同义词相似度，词典处理时，文件中空行或者开头为“#”的注释以及每行开头和结尾的空格会被忽略。

  - 词权重格式：<token words> [: <token weight>]。token weight可以为小数或者整数，默认值为3，中间的冒号（:）作为分隔符，词权重越高，分词时越容易将该字符串分成一个词。
  - 同义词相似度格式：<token words 1> :: <token words 2> [: <similarity>]。表示token words 2是token words 1的同义词，其中，similarity可以为0~1之间的数值表示其相似度，默认值为0.8。同义词相似度只表示单向同义，双向同义需用词位置相反的两行定义表示。

- STOPFILE

  停用词表文件名，默认后缀名为stop，文件格式要求与Simple词典的停用词文件相同。

- FILEPATH

  Tokenweight词典定义文件所在目录。可以指定为本地目录，格式为“file://absolute_path”。默认值为预定义词典文件所在目录。FILEPATH参数必须和STOPFILE或者TOKENFILE参数同时指定，不允许单独指定。

说明

词典定义文件的文件名仅支持小写字母、数字以及下划线混合。

- **value**

  参数值。当参数值为字符串类型时，必须使用单引号进行包裹（标示符和数字同样可以使用单引号进行包裹）。

示例

文本搜索配置（Text Search Configuration），指定了将文档转换成tsvector过程中所必需的组件：

- 解析器，用于把文本分解成标记token。
- 词典列表，用于将每个token转换成词位lexeme。

每次to_tsvector或to_tsquery函数调用时，都需要指定一个文本搜索配置来指定具体的处理过程。GUC参数default_text_search_config指定了默认的文本搜索配置，当文本搜索函数中没有显式指定文本搜索配置参数时，将会使用该默认值进行处理。

GaussDB中预定义有一些可用的文本搜索配置，用户也可创建自定义的文本搜索配置。此外，为了便于管理文本搜索对象，还提供有多个gsql元命令，可以显示有关文本搜索对象的信息（详细请参见**元命令参考**）。

操作步骤：

步骤1 创建一个文本搜索配置ts_conf，复制预定义的文本搜索配置english。

```
gaussdb=# CREATE TEXT SEARCH CONFIGURATION ts_conf ( COPY = pg_catalog.english );
CREATE TEXT SEARCH CONFIGURATION
```


版权所有 © 华为云计算技术有限公司

步骤2 创建Synonym词典。

假设同义词词典定义文件gs_dict.syn内容如下：

```
gaussdb    gs
gauss      gs
```

执行如下语句创建Synonym词典：

```
gaussdb=# CREATE TEXT SEARCH DICTIONARY gs_dict (
    TEMPLATE = synonym,
    SYNONYMS = gs_dict,
    FILEPATH = 'file:///home/dicts'
);
```

步骤3 创建一个Ispell词典english_ispell（词典定义文件来自开源词典）。

```
gaussdb=# CREATE TEXT SEARCH DICTIONARY english_ispell (
   TEMPLATE = ispell,
   DictFile = english,
   AffFile = english,
   StopWords = english,
   FILEPATH = 'file:///home/dicts'
);
```

步骤4 设置文本搜索配置ts_conf，修改某些类型的token对应的词典列表。

```
gaussdb=# ALTER TEXT SEARCH CONFIGURATION ts_conf
   ALTER MAPPING FOR asciiword, asciihword, hword_asciipart,
                     word, hword, hword_part
   WITH gs_dict, english_ispell, english_stem;
```

步骤5 在文本搜索配置中，选择设置不索引或搜索某些token类型。

```
gaussdb=# ALTER TEXT SEARCH CONFIGURATION ts_conf
   DROP MAPPING FOR email, url, url_path, sfloat, float;
```

步骤6 使用文本检索调测函数ts_debug()对所创建的词典配置ts_conf进行测试。

```
gaussdb=# SELECT * FROM ts_debug('ts_conf', '
GaussDB, the highly scalable, SQL compliant, open source object-relational
database management system, is now undergoing beta testing of the next
version of our software.
');
```

步骤7 可以设置当前session使用ts_conf作为默认的文本搜索配置。此设置仅在当前session有效。

```
gaussdb=# \dF+ ts_conf
     Text search configuration "public.ts_conf"
Parser: "pg_catalog.default"
      Token      |            Dictionaries
-----------------+------------------------------------
 asciihword      | gs_dict,english_ispell,english_stem
 asciiword       | gs_dict,english_ispell,english_stem
 file            | simple
 host            | simple
 hword           | gs_dict,english_ispell,english_stem
 hword_asciipart | gs_dict,english_ispell,english_stem
 hword_numpart   | simple
 hword_part      | gs_dict,english_ispell,english_stem
 int             | simple
 numhword        | simple
 numword         | simple
 uint            | simple
 version         | simple
 word            | gs_dict,english_ispell,english_stem

gaussdb=# SET default_text_search_config = 'public.ts_conf';
SET
gaussdb=# SHOW default_text_search_config;
 default_text_search_config
----------------------------
```


版权所有 © 华为云计算技术有限公司

```
 public.ts_conf
(1 row)
```

----结束

1.13.9.52 CREATE TRIGGER

功能描述

CREATE TRIGGER语句用于创建一个触发器。触发器是一种特殊类型的存储过程，触发器主要用于复杂的业务规则和要求，有助于引用的完整性，触发器将与指定的表或视图关联，并在特定条件下执行指定的函数，这样便于在添加、更新或删除表中的行时保留表中已定义的关系。

注意事项

- 当前仅支持在普通表和视图上创建触发器，不支持在临时表、unlogged表等类型表上创建触发器。
- 如果为同一事件定义了多个相同类型的触发器，则按触发器的名称字母顺序触发它们。
- 触发器常用于多表间数据关联同步场景，对SQL执行性能影响较大，不建议在大数据量同步及对性能要求高的场景中使用。
- 当触发器满足如下条件时，触发语句能和触发器一起下推到DN执行并提升触发器执行性能：
  - GUC参数enable_trigger_shipping和enable_fast_query_shipping开启。
  - 源表触发器使用的触发器函数为plpgsql类型（推荐类型）。
  - 源表与触发表分布键的类型、数量完全相同，均为行存表，且所属相同的NODEGROUP。
  - 原INSERT/UPDATE/DELETE语句条件中包含所有分布键与NEW/OLD等值比较表达式。
  - 原INSERT/UPDATE/DELETE语句在没有触发器的情况下就能query shipping。
  - 源表上只有BEFORE INSERT FOR EACH ROW、AFTER INSERT FOR EACH ROW、BEFORE UPDATE FOR EACH ROW、AFTER UPDATE FOR EACH ROW、BEFORE DELETE FOR EACH ROW、AFTER DELETE FOR EACH ROW六类触发器，且所有触发器都可下推。
- 执行触发器语句时是用触发器创建者的身份进行权限判断的。
- 执行创建触发器操作的用户需要拥有指定表的TRIGGER权限或被授予了CREATE ANY TRIGGER权限。
- BEFORE触发的行级触发器函数可以返回一个NULL值，表示忽略对该行的操作，之后的触发器将不再执行，并且不会对该行产生INSERT/UPDATE/DELETE动作。

  AFTER触发器函数返回值无影响。
- BEFORE DELETE的情况下，触发器函数返回值NEW等于NULL。BEFORE INSERT的情况下，触发器函数返回值OLD等于NULL。BEFORE UPDATE的情况下，触发器函数返回值只有显示为NULL才是NULL值。
- 对于event为INSERT/UPDATE的触发器函数，正常返回值是NEW。如果返回一个非NULL的行，BEFORE触发器将修改插入或者更新的行，AFTER触发器不受影响。对于event为DELETE的触发器函数，正常返回值是OLD。

版权所有 © 华为云计算技术有限公司

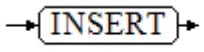
- INSTEAD OF触发器只能作用于视图，其触发器函数同样可以返回NULL值，表示随后的触发器将不再执行。
- 不支持使用package函数作为触发器函数。
- 当存在重载函数时，不支持使用触发器内部隐式变量（如：old、new）作为重载函数入参。
- 分布式PGXC计划下，如果BEFORE ROW触发器修改目标表的同一行，会导致触发器的修改不生效。
- MERGE INTO语句中不支持触发触发器。
- REPLACE INTO语句中不支持触发行级（ROW）触发器；对于语句级（STATEMENT）触发器，仅支持触发INSERT STATEMENT触发器，不支持触发DELETE STATEMENT触发器。
- INSERT ... ON DUPLICATE KEY UPDATE | ON CONFLICT语句中支持触发语句级（STATEMENT）触发器，不支持触发行级（ROW）触发器。
- 在触发器函数或触发器匿名块函数中，cursor%RowType被视为复合类型（即record），不支持解析游标的具体类型。只有当使用该类型定义变量并赋值时，才会获取相应的列信息。

语法格式

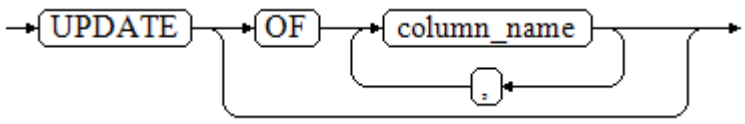
```
CREATE [OR REPLACE] [ CONSTRAINT ] TRIGGER name { BEFORE | AFTER | INSTEAD OF } { event [ OR
event ] [ ... ] }
   ON table_name
   [ FROM referenced_table_name ]
   [ NOT DEFERRABLE [ INITIALLY IMMEDIATE ] | [ DEFERRABLE ] [ INITIALLY IMMEDIATE | INITIALLY
DEFERRED ] ]
   [ FOR [ EACH ] { ROW | STATEMENT } ]
   [ WHEN ( condition ) ]
   [ EXECUTE PROCEDURE function_name ( arguments ) | AnonyBlock ];
```

其中event包含以下几种：

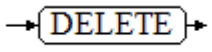
```
INSERT
```



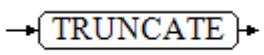

```
UPDATE [ OF column_name [, ... ] ]
```




```
DELETE
```




```
TRUNCATE
```



参数说明

- **OR REPLACE**

版权所有 © 华为云计算技术有限公司

可选项，指定此参数将在TRIGGER已存在的前提下修改已有的TRIGGER。

无法通过OR REPLACE语法创建或修改CONSTRAINT TRIGGER和INTERNAL TRIGGER。CONSTRAINT TRIGGER是指通过CREATE CONSTRAINT TRIGGER创建的TRIGGER，INTERNAL TRIGGER是指某些SQL语句隐式创建的TRIGGER（如给表加上外键约束会隐式创建4个trigger，这4个TRIGGER即为INTERNAL TRIGGER）。

- **CONSTRAINT**

  可选项，指定此参数将创建约束触发器，即触发器作为约束来使用。除了可以使用SET CONSTRAINTS调整触发器触发的时间之外，这与常规触发器相同。 约束触发器必须是AFTER ROW触发器。

- **name**

  触发器名称，该名称不能限定模式，因为触发器自动继承其所在表的模式，且同一个表的触发器不能重名。 对于约束触发器，使用**SET CONSTRAINTS**修改触发器行为时也使用此名称。

  取值范围：符合**标识符命名规范**的字符串，且最大长度不超过63个字符。

- **BEFORE**

  触发器函数是在触发事件发生前执行。

- **AFTER**

  触发器函数是在触发事件发生后执行，约束触发器只能指定为AFTER。

- **INSTEAD OF**

  触发器函数直接替代触发事件。

- **event**

  启动触发器的事件，取值范围包括：INSERT、UPDATE、DELETE或TRUNCATE，也可以通过OR同时指定多个触发事件。

  对于UPDATE事件类型，可以使用下面语法指定列：

  ```
  UPDATE OF column_name1 [, column_name2 ... ]
  ```

  表示只有这些列作为UPDATE语句的目标列时，才会启动触发器，但是INSTEAD OF UPDATE类型不支持指定列信息。如果UPDATE OF指定的列包含生成列，当生成列依赖的列是UPDATE语句的目标列时，也会启动触发器。

- **table_name**

  需要创建触发器的表名称。

  取值范围：数据库中已经存在的表名称。

- **referenced_table_name**

  约束引用的另一个表的名称。 只能为约束触发器指定，常见于外键约束。由于当前不支持外键，因此不建议使用。

  取值范围：数据库中已经存在的表名称。

- **DEFERRABLE | NOT DEFERRABLE**

  约束触发器的启动时机，仅作用于约束触发器。这两个关键字设置该约束是否可推迟。

  详细介绍请参见**CREATE TABLE**。

- **INITIALLY IMMEDIATE | INITIALLY DEFERRED**

  如果约束是可推迟的，则这个子句声明检查约束的缺省时间，仅作用于约束触发器。

  详细介绍请参见**CREATE TABLE**。

版权所有 © 华为云计算技术有限公司

- **FOR EACH ROW | FOR EACH STATEMENT**

  触发器的触发频率。

  – FOR EACH ROW是指该触发器是受触发事件影响的每一行触发一次。

  – FOR EACH STATEMENT是指该触发器是每个SQL语句只触发一次。

  未指定时默认值为FOR EACH STATEMENT。约束触发器只能指定为FOR EACH ROW。

- **condition**

  决定是否实际执行触发器函数的条件表达式。当指定WHEN时，只有在条件返回true时才会调用该函数。

  在FOR EACH ROW触发器中，WHEN条件可以通过分别写入OLD.column_name或NEW.column_name来引用旧行或新行值的列。INSERT触发器不能引用OLD，DELETE触发器不能引用NEW。

  INSTEAD OF触发器不支持WHEN条件。

  WHEN表达式不能包含子查询。

  对于约束触发器，WHEN条件的评估不会延迟，而是在执行更新操作后立即发生。 如果条件返回值不为true，则触发器不会排队等待延迟执行。

- **function_name**

  用户定义的函数，必须声明为不带参数并返回类型为触发器，在触发器触发时执行。

- **arguments**

  执行触发器时要提供给函数的可选的以逗号分隔的参数列表。参数是文字字符串常量，简单的名称和数字常量也可以写在这里，但它们都将被转换为字符串。 请检查触发器函数的实现语言的描述，以了解如何在函数内访问这些参数。

  说明

  关于触发器种类：

  - INSTEAD OF的触发器必须标记为FOR EACH ROW，并且只能在视图上定义。
  - BEFORE和AFTER触发器作用在视图上时，只能标记为FOR EACH STATEMENT。
  - TRUNCATE类型触发器仅限FOR EACH STATEMENT。

表 1-400 表和视图上支持的触发器种类：

触发时机	触发事件	行级	语句级
BEFORE	INSERT/UPDATE/DELETE	表	表和视图
	TRUNCATE	不支持	表
AFTER	INSERT/UPDATE/DELETE	表	表和视图
	TRUNCATE	不支持	表
INSTEAD OF	INSERT/UPDATE/DELETE	视图	不支持
	TRUNCATE	不支持	不支持

版权所有 © 华为云计算技术有限公司

表 1-401 plpgsql 类型触发器函数特殊变量：

变量名	变量含义
NEW	INSERT及UPDATE操作涉及tuple信息中的新值，对DELETE为空，并且对于statement级别的触发器此变量也为空。
OLD	UPDATE及DELETE操作涉及tuple信息中的旧值，对INSERT为空，并且对于statement级别的触发器此变量也为空。
TG_NAME	触发器名称。
TG_WHEN	触发器触发时机（BEFORE/AFTER/INSTEAD OF）。
TG_LEVEL	触发频率（ROW/STATEMENT）。
TG_OP	触发操作（INSERT/UPDATE/DELETE/TRUNCATE）。
TG_RELID	触发器所在表OID。
TG_RELNAME	触发器所在表名（现用TG_TABLE_NAME替代，与TG_TABLE_NAME实现功能相同，建议使用TG_TABLE_NAME）。
TG_TABLE_NAME	触发器所在表名。
TG_TABLE_SCHEMA	触发器所在表的SCHEMA信息。
TG_NARGS	触发器函数参数个数。
TG_ARGV[]	触发器函数参数列表。

示例

```
--创建源表及触发表。
gaussdb=# CREATE TABLE test_trigger_src_tbl(id1 INT, id2 INT, id3 INT);
gaussdb=# CREATE TABLE test_trigger_des_tbl(id1 INT, id2 INT, id3 INT);
```

INSERT触发器的使用。

```
--创建insert触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION tri_insert_func() RETURNS TRIGGER AS
        $$
        DECLARE
        BEGIN
                INSERT INTO test_trigger_des_tbl VALUES(NEW.id1, NEW.id2, NEW.id3);
                RETURN NEW;
        END
        $$ LANGUAGE plpgsql;

--创建INSERT触发器。
gaussdb=# CREATE TRIGGER insert_trigger
        BEFORE INSERT ON test_trigger_src_tbl
        FOR EACH ROW
```


版权所有 © 华为云计算技术有限公司

```
        EXECUTE PROCEDURE tri_insert_func();

--执行INSERT触发事件并检查触发结果。
gaussdb=# INSERT INTO test_trigger_src_tbl VALUES(100,200,300);

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 300
(1 row)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //查看触发操作是否生效。
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 300
(1 row)
```

UPDATE触发器的使用。

```
--创建UPDATE触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION tri_update_func() RETURNS TRIGGER AS
$$
DECLARE
BEGIN
UPDATE test_trigger_des_tbl SET id3 = NEW.id3 WHERE id1=OLD.id1;
RETURN OLD;
END
$$ LANGUAGE plpgsql;

--创建UPDATE触发器。
gaussdb=# CREATE TRIGGER update_trigger
        AFTER UPDATE ON test_trigger_src_tbl
        FOR EACH ROW
        EXECUTE PROCEDURE tri_update_func();

--执行UPDATE触发事件并检查触发结果。
gaussdb=# UPDATE test_trigger_src_tbl SET id3=400 WHERE id1=100;

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 400
(1 row)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //查看触发操作是否生效。
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 400
(1 row)
```

DELETE触发器的使用。

```
--创建DELETE触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION TRI_DELETE_FUNC() RETURNS TRIGGER AS
        $$
        DECLARE
        BEGIN
            DELETE FROM test_trigger_des_tbl WHERE id1=OLD.id1;
            RETURN OLD;
        END
        $$ LANGUAGE plpgsql;

--创建DELETE触发器。
gaussdb=# CREATE TRIGGER delete_trigger BEFORE DELETE ON test_trigger_src_tbl FOR EACH ROW
EXECUTE PROCEDURE tri_delete_func();

--执行DELETE触发事件并检查触发结果。
gaussdb=# DELETE FROM test_trigger_src_tbl WHERE id1=100;

gaussdb=# SELECT * FROM test_trigger_src_tbl;
```


版权所有 © 华为云计算技术有限公司

```
 id1 | id2 | id3
-----+-----+-----
(0 rows)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //查看触发操作是否生效。
 id1 | id2 | id3
-----+-----+-----
(0 rows)
```

修改触发器名称。

```
--修改触发器的名称。
gaussdb=# ALTER TRIGGER delete_trigger ON test_trigger_src_tbl RENAME TO delete_trigger_renamed;
```

禁用触发器。

```
--禁用insert_trigger触发器。
gaussdb=# ALTER TABLE test_trigger_src_tbl DISABLE TRIGGER insert_trigger;

gaussdb=# INSERT INTO test_trigger_src_tbl VALUES(100,200,300);

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 300
(1 row)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //可以看到触发器没有生效。
 id1 | id2 | id3
-----+-----+-----
(0 rows)

--禁用当前表上所有触发器。
gaussdb=# ALTER TABLE test_trigger_src_tbl DISABLE TRIGGER ALL;
```

删除触发器。

```
gaussdb=# DROP TRIGGER insert_trigger ON test_trigger_src_tbl;

gaussdb=# DROP TRIGGER update_trigger ON test_trigger_src_tbl;

gaussdb=# DROP TRIGGER delete_trigger_renamed ON test_trigger_src_tbl;
```

删除函数。

```
gaussdb=# DROP FUNCTION tri_insert_func;

gaussdb=# DROP FUNCTION tri_update_func;

gaussdb=# DROP FUNCTION tri_delete_func;
--删除源表及触发表。
gaussdb=# DROP TABLE test_trigger_src_tbl;
gaussdb=# DROP TABLE test_trigger_des_tbl;
```

相关链接

ALTER TRIGGER，**DROP TRIGGER**，**ALTER TABLE**

1.13.9.53 CREATE TYPE

功能描述

在当前数据库中定义一种新的数据类型。

当前仅支持创建复合类型、基本类型、shell类型、枚举类型和集合类型。具体介绍如下：

版权所有 © 华为云计算技术有限公司

- 复合类型

  复合类型由一个属性名和数据类型的列表指定。如果属性的数据类型是可排序的，也可以指定该属性的排序规则。复合类型本质上和表的行类型相同，但是如果只想定义一种类型，使用CREATE TYPE避免了创建一个实际的表。单独的复合类型也是很有用的，例如可以作为函数的参数或者返回类型。

  为了能够创建复合类型，必须拥有在其所有属性类型上的USAGE特权。

- 基本类型

  用户可以自定义一种新的基本类型（标量类型）。

- shell类型

  shell类型是一种用于后面要定义的类型的占位符，通过发出一个不带除类型名之外其他参数的CREATE TYPE命令可以创建这种类型。在创建基本类型时，需要shell类型作为一种向前引用。

- 枚举类型

  由若干个标签构成的列表，每一个标签值都是一个非空字符串，且字符串长度必须不超过64个字节。

- 集合类型

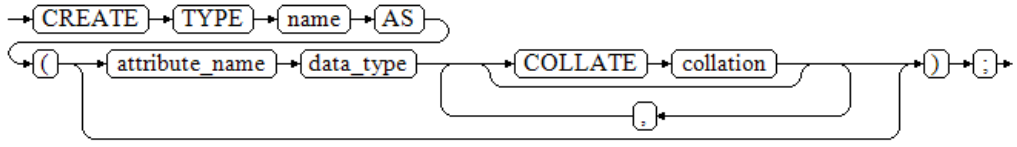
  类似数组，但是没有长度限制，主要在存储过程中使用。

注意事项

- 如果指定了具体模式名，那么新创建的数据类型将被创建在指定的模式中，否则将被创建在当前模式中。
- 新创建的类型名称不能与同一个模式中已有的类型或者表重名（由于表具有相关的数据类型属性，新创建的类型名称不能与同一个模式中已有表的名称相同）。
- 创建数据类型的用户将成为该数据类型的所有者。
- 被授予CREATE ANY TYPE权限的用户，可以在PUBLIC模式和用户模式下创建类型。
- 类型只适用于行存表。
- 当用户使用关联函数进行创建非系统自带类型时，须了解新创建类型的定义及其所关联的函数。如果操作不当，可能会因所关联的函数而导致权限被滥用的风险。

语法格式

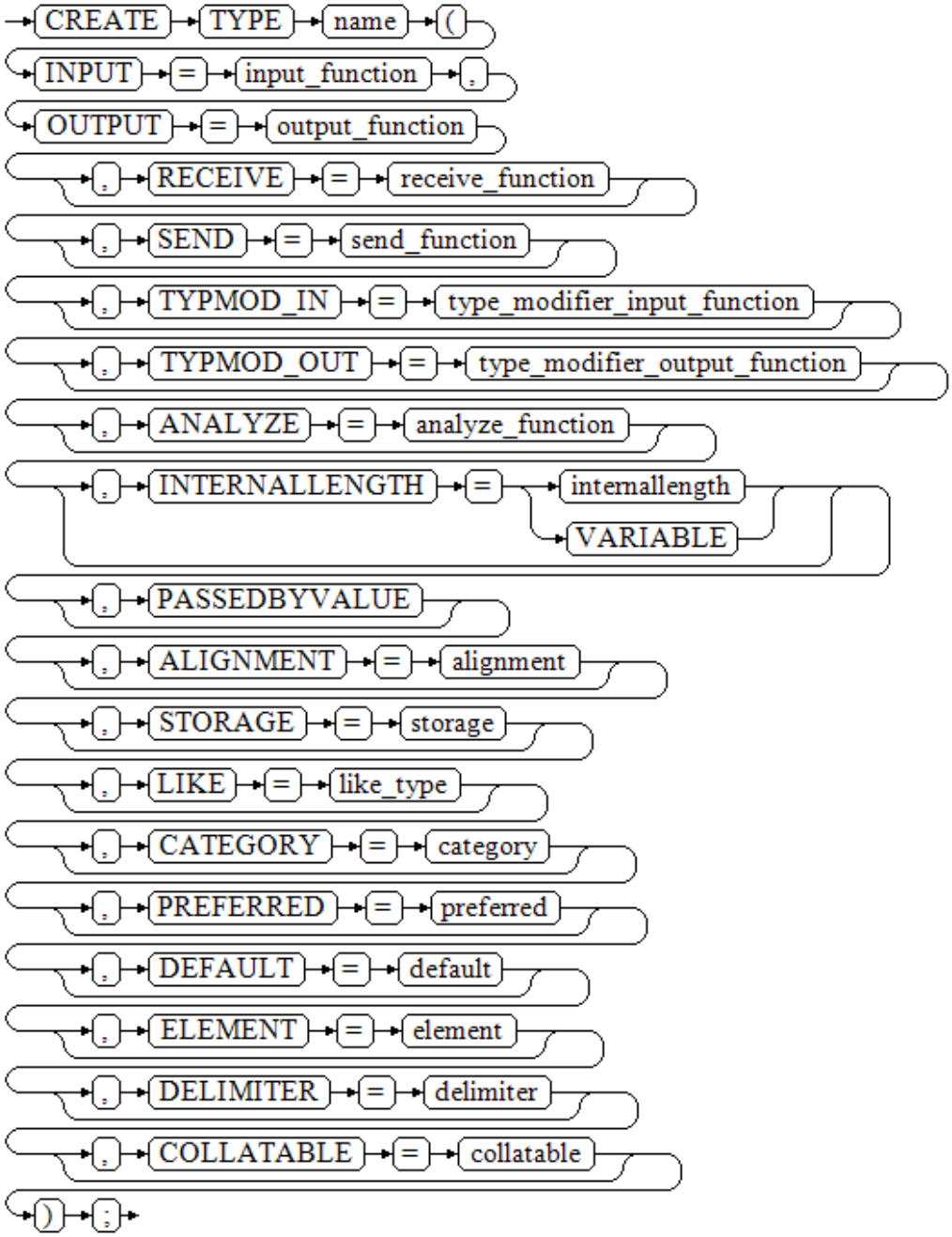
```
CREATE TYPE name AS
    ( [ attribute_name data_type [ COLLATE collation ] [, ... ] ] );
```



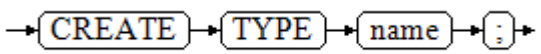

```
CREATE TYPE name (
    INPUT = input_function,
    OUTPUT = output_function
    [ , RECEIVE = receive_function ]
    [ , SEND = send_function ]
    [ , TYPMOD_IN = type_modifier_input_function ]
    [ , TYPMOD_OUT = type_modifier_output_function ]
    [ , ANALYZE = analyze_function ]
    [ , INTERNALLENGTH = { internallength | VARIABLE } ]
```


版权所有 © 华为云计算技术有限公司

```
    [ , PASSEDBYVALUE ]
    [ , ALIGNMENT = alignment ]
    [ , STORAGE = storage ]
    [ , LIKE = like_type ]
    [ , CATEGORY = category ]
    [ , PREFERRED = preferred ]
    [ , DEFAULT = default ]
    [ , ELEMENT = element ]
    [ , DELIMITER = delimiter ]
    [ , COLLATABLE = collatable ]
);
```

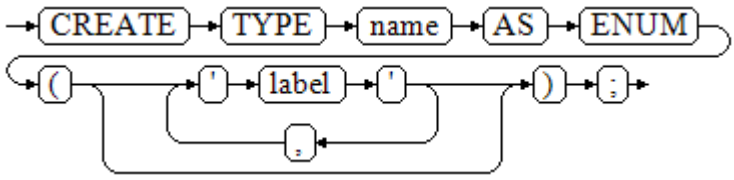



```
CREATE TYPE name;
```

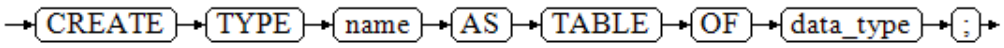


版权所有 © 华为云计算技术有限公司

```
CREATE TYPE name AS ENUM
    ( [ {'label'} [, ... ] ] );
```




```
CREATE TYPE name AS TABLE OF data_type;
```



参数说明

- 复合类型
  - **name**

    要创建的类型的名称，可以使用模式进行具体指定。
  - **attribute_name**

    复合类型的一个属性（列）的名称。
  - **data_type**

    要成为复合类型的一个列的现有数据类型的名称。可以使用%RowType间接引用表的类型，或者使用%Type间接引用表或复合类型中某一列的类型。
  - **collation**

    要关联到复合类型的一列的现有排序规则的名称。排序规则可以使用“SELECT * FROM pg_collation;”命令从pg_collation系统表中查询，默认的排序规则为查询结果中以default开始的行。

    说明

    复合类型构造器赋值时支持使用=>，约束如下：
    - 复合类型构造器赋值时使用=>功能，仅在兼容ORA模式数据库下（即sql_compatibility='ORA'）支持。
    - 复合类型构造器赋值时使用=>功能，仅在入参赋值连续使用=>到结尾时支持，如composite_name(elename1 => val1, elename2 => val2, elename3 => val3)或composite_name(val1, elename2 => val2, elename3 => val3)；不支持非连续使用=>赋值或者使用=>赋值时未到最后一个入参，如composite_name(elename1 => val1, elename2 => val2, val3)或composite_name(val1, elename2 => val2, val3)。
- 基本类型：自定义基本类型时，参数可以以任意顺序出现，input_function和output_function为必选参数，其他为可选参数。
  - **input_function**

    将数据从类型的外部文本形式转换为内部形式的函数名。

    输入函数可以被声明为有一个CSTRING类型的参数，或者有三个分别为CSTRING类型、OID类型、INTEGER类型的参数。

    - CSTRING类型参数是以C字符串存在的输入文本。
    - OID类型参数是该类型自身的OID（对于数组类型则是其元素类型的OID）。

版权所有 © 华为云计算技术有限公司

- INTEGER类型参数是目标列的typmod（未知情况下传递-1）。

输入函数必须返回一个该数据类型本身的值。通常，一个输入函数应该被声明为STRICT，否则，在读到一个NULL输入值时，调用输入函数时，第一个参数会是NULL，导致该函数必须仍然返回NULL，除非调用函数发生了错误。

说明

- 输入和输出函数能被声明为具有新类型的结果或参数是因为：必须在创建新类型之前创建这两个函数。而新类型应该首先被定义为一种shell type，它是一种占位符类型，除了名称和拥有者之外它没有其他属性。这可以通过不带额外参数的命令CREATE TYPE name做到。然后用C写的I/O函数可以被定义为引用这种shell type。最后，用带有完整定义的CREATE TYPE把该shell type替换为一个完全的、合法的类型定义，之后新类型就可以正常使用了。
- 输入和输出函数若为INTERNAL类型且指定为内部系统函数，则其输入函数和输出函数的参数类型需保持一致，且新类型的INTERNALLENGTH和PASSEDBYVALUE需要与输入函数和输出函数的参数类型保持一致。

– **output_function**

将数据从类型的内部形式转换为外部文本形式的函数名。

输出函数必须被声明为有一个新数据类型的参数。输出函数必须返回类型CSTRING。对于NULL值不会调用输出函数。

– **receive_function**

可选。将数据从类型的外部二进制形式转换成内部形式的函数名。

如果没有该函数，该类型不能参与到二进制输入中。二进制表达转换成内部形式代价更低，然而却更容易移植（例如，标准的整数数据类型使用网络字节序作为外部二进制表达，而内部表达是机器本地的字节序）。receive_function应该执行足够的检查以确保该值是有效的。

接收函数可以被声明为有一个INTERNAL类型的参数，或者有三个分别为INTERNAL类型、OID类型、INTEGER类型的参数。

- INTERNAL类型参数是一个指向StringInfo缓冲区的指针，其中保存着接收到的字节串。
- OID类型参数和INTEGER类型参数和文本输入函数的相同。

接收函数必须返回一个该数据类型本身的值。通常，一个接收函数应该被声明为STRICT，否则，在读到一个NULL输入值时调用接收函数时，第一个参数会是NULL，导致该函数必须仍然返回NULL，除非接收函数发生了错误。

– **send_function**

可选。将数据从类型的内部形式转换为外部二进制形式的函数名。

如果没有该函数，该类型将不能参与到二进制输出中。发送函数必须被声明为有一个新数据类型的参数。发送函数必须返回类型BYTEA。对于NULL值不会调用发送函数。

– **type_modifier_input_function**

可选。将类型的修饰符数组转换为内部形式的函数名。

– **type_modifier_output_function**

可选。将类型的修饰符的内部形式转换为外部文本形式的函数名。

版权所有 © 华为云计算技术有限公司

说明

如果该类型支持修饰符（附加在类型声明上的可选约束，例如，char(5)或numeric(30,2)），则需要可选的type_modifier_input_function以及type_modifier_output_function。GaussDB允许用户定义的类型有一个或者多个简单常量或者标识符作为修饰符。不过，为了存储在系统目录中，该信息必须能被打包到一个非负整数值中。所声明的修饰符会被以CSTRING数组的形式传递给type_modifier_input_function。type_modifier_input_function必须检查该值的合法性（如果值错误就抛出一个错误），如果值正确，要返回一个非负integer值，该值将被存储在“typmod”列中。如果类型没有type_modifier_input_function则类型修饰符将被拒绝。type_modifier_output_function把内部的整数typmod值转换回正确的形式用于用户显示。type_modifier_output_function必须返回一个CSTRING值，该值就是追加到类型名称后的字符串。例如，numeric的函数可能会返回(30,2)。如果默认的显示格式就是只把存储的typmod整数值放在圆括号内，则允许省略type_modifier_output_function。

- **analyze_function**

  可选。为该数据类型执行统计分析的函数名的可选参数。

  默认情况下，如果该类型有一个默认的B-Tree操作符类，ANALYZE将尝试用类型的“equals”和“less-than”操作符来收集统计信息。这种行为对于非标量类型并不合适，因此可以通过指定一个自定义分析函数来覆盖这种行为。分析函数必须被声明为有一个类型为INTERNAL的参数，并且返回一个布尔值结果。

- **internallength**

  可选。一个数字常量，用于指定新类型的内部表达的字节长度。默认为变长。

  虽然只有I/O函数和其他为该类型创建的函数才知道新类型的内部表达的细节，但是内部表达的一些属性必须被向GaussDB声明。其中最重要的是INTERNALLENGTH。基本数据类型可以是定长的（这种情况下internallength是一个正整数）或者是变长的（把INTERNALLENGTH设置为VARIABLE，在内部通过把typlen设置为-1表示）。所有变长类型的内部表达都必须以一个4字节整数开始，internallength定义了总长度。

- **PASSEDBYVALUE**

  可选。表示这种数据类型的值需要被传值而不是传引用。传值的类型必须是定长的，并且其内部表达不能超过Datum类型（某些机器上是4字节，其他机器上是8字节）的尺寸。

- **alignment**

  可选。该参数指定数据类型的存储对齐需求。如果被指定，必须是CHAR、INT2、INT4或者DOUBLE。默认是INT4。

  允许的值等同于以1、2、4或8字节边界对齐。要注意变长类型的alignment参数必须至少为4，因为需要包含一个INT4作为其第一个组成部分。

- **storage**

  可选。该数据类型的存储策略。

  如果被指定，必须是plain、external、extended或者main。默认是plain。

  - plain指定该类型的数据将总是被存储在线内并且不会被压缩。（对定长类型只允许plain）
  - extended指定系统将首先尝试压缩一个长的数据值，并且将在数据仍然太长的情况下把值移出主表行。
  - external允许值被移出主表，但是系统将不会尝试对其进行压缩。

版权所有 © 华为云计算技术有限公司

- main允许压缩，但是不鼓励把值移出主表（如果没有其他办法让行的大小变得合适，具有这种存储策略的数据项仍将被移出主表，但比起extended以及external项，这种存储策略的数据项会被优先考虑保留在主表中）。

除plain之外所有的storage值都暗示该数据类型的函数能处理被TOAST过的值。指定的值仅仅是决定一种可TOAST数据类型的列的默认TOAST存储策略，用户可以使用ALTER TABLE SET STORAGE为列选取其他策略。

– **like_type**

可选。与新类型具有相同表达的现有数据类型的名称。会从这个类型中复制internallength、passedbyvalue、alignment以及storage的值（除非在这个CREATE TYPE命令的其他地方用显式说明覆盖）。

当新类型的低层实现是以一种现有的类型为参考时，用这种方式指定表达特别有用。

– **category**

可选。这种类型的分类码（一个ASCII 字符）。默认是“用户定义类型”的'U'。为了创建自定义分类，也可以选择其他ASCII字符。

– **preferred**

可选。如果这种类型是其类型分类中的优先类型则为true，否则为false。默认为false。在一个现有类型分类中创建一种新的优先类型要非常谨慎，因为这可能会导致很大的改变。

说明

category和preferred参数可以被用来帮助控制在混淆的情况下应用哪一种隐式造型。每一种数据类型都属于一个用单个ASCII字符命名的分类，并且每一种类型可以是其所属分类中的“首选”。当有助于解决重载函数或操作符时，解析器将优先造型到首选类型（但是只能从同类的其他类型造型）。对于没有隐式转换到或来自任意其他类型的类型，让这些设置保持默认即可。不过，对于有隐式转换的相关类型的组，把它们都标记为属于同一个类别并且选择一种或两种“最常用”的类型作为该类别的首选通常是很有用的。在把一种用户定义的类型增加到一个现有的内建类别（例如，数字或者字符串类型）中时，category参数特别有用。不过，也可以创建新的全部是用户定义类型的类别。对这样的类别，可选择除大写字母之外的任何ASCII字符。

– **default**

可选。数据类型的默认值。如果被省略，默认值是空。

如果用户希望该数据类型的列被默认为某种非空值，可以指定一个默认值。默认值可以用DEFAULT关键词指定（这样一个默认值可以被附加到一个特定列的显式DEFAULT子句覆盖）。

– **element**

可选。被创建的类型是一个数组，element指定了数组元素的类型。例如，要定义一个4字节整数的数组（INT4），应指定ELEMENT = INT4。

– **delimiter**

可选。指定这种类型组成的数组中分隔值的定界符。

可以把delimiter设置为一个特定字符，默认的定界符是逗号（,）。注意定界符是与数组元素类型相关的，而不是数组类型本身相关。

– **collatable**

可选。如果这个类型的操作可以使用排序规则信息，则为true。默认为false。

版权所有 © 华为云计算技术有限公司

如果collatable为true，这种类型的列定义和表达式可能通过使用COLLATE子句携带有排序规则信息。在该类型上操作的函数的实现负责真正利用这些信息，仅把类型标记为可排序的并不会让它们自动地去使用这类信息。

- **label**

  可选。与枚举类型的一个值相关的文本标签，其值为长度不超过63个字符的非空字符串。

说明

在创建用户定义类型的时候，GaussDB会自动创建一个与之关联的数组类型，其名称由该元素类型的名称前缀一个下划线组成。

示例

```
--创建一种复合类型，建表并插入数据以及查询。
gaussdb=# CREATE TYPE compfoo AS (f1 int, f2 text);
gaussdb=# CREATE TABLE t1_compfoo(a int, b compfoo);
gaussdb=# CREATE TABLE t2_compfoo(a int, b compfoo);
gaussdb=# INSERT INTO t1_compfoo values(1,(1,'demo'));
gaussdb=# INSERT INTO t2_compfoo SELECT * FROM t1_compfoo;
gaussdb=# SELECT (b).f1 FROM t1_compfoo;
gaussdb=# SELECT * FROM t1_compfoo t1 JOIN t2_compfoo t2 ON (t1.b).f1=(t1.b).f1;

--重命名数据类型。
gaussdb=# ALTER TYPE compfoo RENAME TO compfoo1;

--要改变一个用户定义类型compfoo1的所有者为usr1。
gaussdb=# CREATE USER usr1 PASSWORD '********';
gaussdb=# ALTER TYPE compfoo1 OWNER TO usr1;

--把用户定义类型compfoo1的模式改变为usr1。
gaussdb=# ALTER TYPE compfoo1 SET SCHEMA usr1;

--给一个数据类型增加一个新的属性。
gaussdb=# ALTER TYPE usr1.compfoo1 ADD ATTRIBUTE f3 int;

--删除compfoo1类型。
gaussdb=# DROP TYPE usr1.compfoo1 CASCADE;

--删除相关表和用户。
gaussdb=# DROP TABLE t1_compfoo;
gaussdb=# DROP TABLE t2_compfoo;
gaussdb=# DROP SCHEMA usr1;
gaussdb=# DROP USER usr1;

--创建一个枚举类型。
gaussdb=# CREATE TYPE bugstatus AS ENUM ('create', 'modify', 'closed');

--添加一个标签值。
gaussdb=# ALTER TYPE bugstatus ADD VALUE IF NOT EXISTS 'regress' BEFORE 'closed';

--重命名一个标签值。
gaussdb=# ALTER TYPE bugstatus RENAME VALUE 'create' TO 'new';

--创建一个集合类型。
gaussdb=# CREATE TYPE bugstatus_table AS TABLE OF bugstatus;

--删除集合类型。
gaussdb=# DROP TYPE bugstatus_table;
--删除枚举类型。
gaussdb=# DROP TYPE bugstatus CASCADE;
--创建ORA兼容模式数据库，并切换到此数据库。
gaussdb=#  CREATE DATABASE ora_compatible_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE
gaussdb=#  \c ora_compatible_db;
Non-SSL connection (SSL connection is recommended when requiring high-security)
```


版权所有 © 华为云计算技术有限公司

```
You are now connected to database "ora_compatible_db" as user "omm".
--创建复合类型。
ora_compatible_db=# CREATE TYPE person_type AS  (name VARCHAR2(50), age NUMBER, gender
VARCHAR2(10));
CREATE TYPE
ora_compatible_db=# CREATE TYPE address_type AS  (street VARCHAR2(50), city VARCHAR2(50), zip_code
VARCHAR2(10));
CREATE TYPE
ora_compatible_db=# CREATE TYPE customer_type AS  (id NUMBER, person_info person_type, address_info
address_type);
CREATE TYPE
--复合类型构造器赋值时全部入参使用=>，执行成功。
ora_compatible_db=# DECLARE
  v_customer customer_type;
 BEGIN
  v_customer := customer_type(
    id => 123,
    person_info => person_type(name => 'John', age => 30, gender => 'Male'),
    address_info => address_type(street => '123 Main St', city => 'Anytown', zip_code => '12345')
  );
 END;
 /
ANONYMOUS BLOCK EXECUTE
--复合类型构造器赋值时，入参赋值连续使用=>到最后一个入参，执行成功。
ora_compatible_db=# DECLARE
  v_person person_type;
 BEGIN
  v_person := person_type('John', age => 30, gender => 'Male');
 END;
 /
ANONYMOUS BLOCK EXECUTE
--复合类型构造器赋值时未连续使用=>到最后一个入参，提示错误。
ora_compatible_db=# DECLARE
  v_person person_type;
 BEGIN
  v_person := person_type(name => 'John', age => 30, 'Male');
 END;
 /
ERROR:  Positional parameters cannot follow named parameters
LINE 1: SELECT person_type(name => 'John', age => 30, 'Male')
                                                      ^
QUERY:  SELECT person_type(name => 'John', age => 30, 'Male')
CONTEXT:  referenced column: person_type
PL/pgSQL function inline_code_block line 4 at assignment
--删除复合类型。
ora_compatible_db=# DROP TYPE customer_type;
DROP TYPE
ora_compatible_db=# DROP TYPE address_type;
DROP TYPE
ora_compatible_db=# DROP TYPE person_type;
DROP TYPE
--删除创建的数据库，并切换回原数据库。
ora_compatible_db=# \c postgres;
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "omm".
gaussdb=# DROP DATABASE ora_compatible_db;
DROP DATABASE
```

相关链接

ALTER TYPE，**DROP TYPE**

1.13.9.54 CREATE USER

功能描述

创建一个带有指定密码的新用户。

版权所有 © 华为云计算技术有限公司

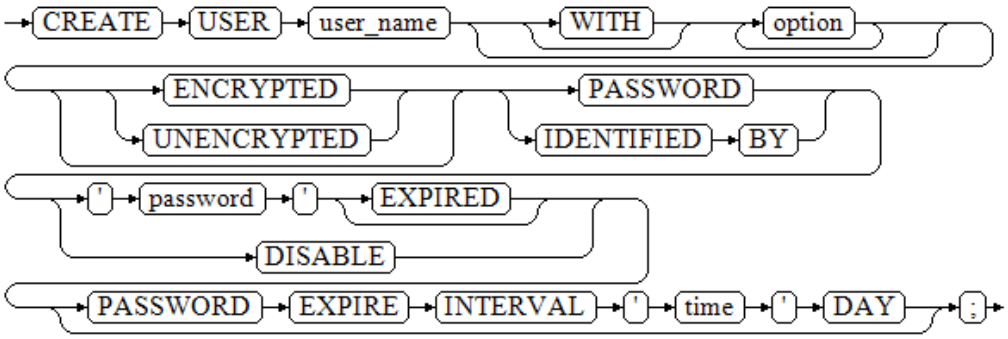
用户是GaussDB数据库认证的基本元素，可以通过正确的用户名和密码登录GaussDB数据库，并对不同的用户授予不同的权限，以便不同的用户进行不同的操作。

注意事项

- 通过CREATE USER创建的用户，默认具有LOGIN权限。
- 通过CREATE USER创建用户的同时，系统会在执行该命令的数据库中，为该用户创建一个同名的Schema。
- 系统管理员在普通用户同名Schema下创建的对象，所有者为Schema的同名用户（非系统管理员）。

语法格式

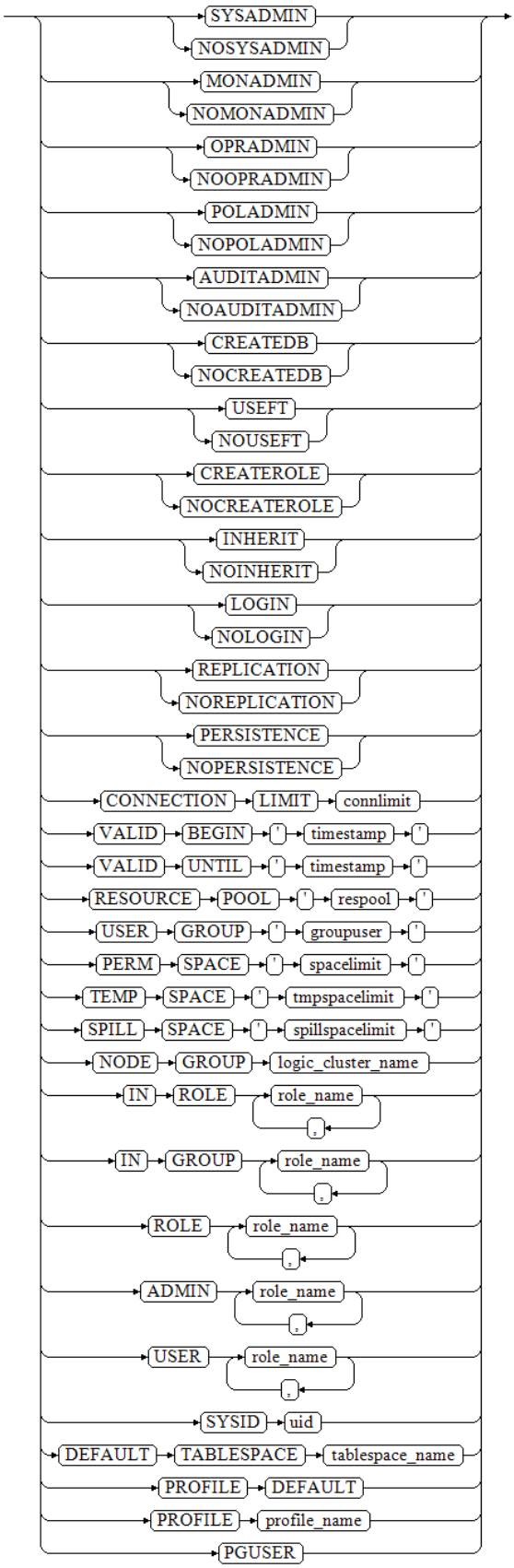
```
CREATE USER user_name [ [ WITH ] option [ ... ] ] [ ENCRYPTED | UNENCRYPTED ] { PASSWORD |
IDENTIFIED BY } { 'password' [EXPIRED] | DISABLE }[ PASSWORD EXPIRE INTERVAL 'time' DAY ];
```



其中option子句用于设置权限及属性等信息。

```
{
   {SYSADMIN | NOSYSADMIN}
   | {MONADMIN | NOMONADMIN}
   | {OPRADMIN | NOOPRADMIN}
   | {POLADMIN | NOPOLADMIN}
   | {AUDITADMIN | NOAUDITADMIN}
   | {CREATEDB | NOCREATEDB}
   | {USEFT | NOUSEFT}
   | {CREATEROLE | NOCREATEROLE}
   | {INHERIT | NOINHERIT}
   | {LOGIN | NOLOGIN}
   | {REPLICATION | NOREPLICATION}
   | {PERSISTENCE | NOPERSISTENCE}
   | CONNECTION LIMIT connlimit
   | VALID BEGIN 'timestamp'
   | VALID UNTIL 'timestamp'
   | RESOURCE POOL 'respool'
   | USER GROUP 'groupuser'
   | PERM SPACE 'spacelimit'
   | TEMP SPACE 'tmpspacelimit'
   | SPILL SPACE 'spillspacelimit'
   | NODE GROUP logic_cluster_name
   | IN ROLE role_name [, ...]
   | IN GROUP role_name [, ...]
   | ROLE role_name [, ...]
   | ADMIN role_name [, ...]
   | USER role_name [, ...]
   | SYSID uid
   | DEFAULT TABLESPACE tablespace_name
   | PROFILE DEFAULT
   | PROFILE profile_name
   | PGUSER
}
```


版权所有 © 华为云计算技术有限公司



版权所有 © 华为云计算技术有限公司

参数说明

- **user_name**

  需要创建的用户名称。

  取值范围：字符串，要符合**标识符命名规范**且最大长度不超过63个字符。

  当用户名称超过63个字符时，数据库会截断并保留前63个字符当作用户名称。

  当用户名称中包含大写字母时，数据库会自动转换为小写字母，如果需要创建包含大写字母的用户名称则需要使用双引号进行声明。

  须知

  redisuser为重分布资源管控对象，当前不支持创建user_name为redisuser的用户。

- **password**

  登录密码。

  取值范围：字符串。

  密码规则如下：

  - 密码默认不少于8个字符。
  - 不能与用户名及用户名倒序相同。
  - 至少包含大写字母（A~Z）、小写字母（a~z）、数字（0~9）以及非字母数字字符（限定为~!@#$%^&*()-_=+\|[{}];:,<.>/?）四类字符中的三类字符。当密码中包含其他字符（不属于上述四种字符范围内）时语句执行报错。
  - 密码也可以使用符合格式要求的密文字符串，主要用于用户数据导入场景，不推荐用户直接使用。数据库不会对密文密码复杂度进行校验，直接使用密文密码可能存在数据安全风险，如果用户直接使用密文密码，需清楚密文密码对应的明文以及保证明文密码复杂度。
  - 创建用户时，用户密码需要使用单引号或双引号进行声明。

- 其他参数请参见**CREATE ROLE**。

示例

```
--创建用户jim，登录密码为********。
gaussdb=# CREATE USER jim PASSWORD '********';

--创建用户kim，登录密码为********。
gaussdb=# CREATE USER kim IDENTIFIED BY '********';

--创建用户tom。
gaussdb=# CREATE USER TOM PASSWORD '********';

--创建用户TOM。
gaussdb=# CREATE USER "TOM" PASSWORD '********';

--如果需要创建具有“创建数据库”权限的用户，则需要加CREATEDB关键字。
gaussdb=# CREATE USER dim CREATEDB PASSWORD '********';

--查看用户dim的权限。
gaussdb=#  \du dim
          List of roles
 Role name | Attributes | Member of
-----------+------------+-----------
 dim       | Create DB  | {}
（可以看到用户dim具有CREATEDB的权限）
```


版权所有 © 华为云计算技术有限公司

```
--修改jim用户密码。
gaussdb=# ALTER USER jim IDENTIFIED BY '**********' REPLACE '********';

--为用户jim追加CREATEROLE权限。
gaussdb=# ALTER USER jim CREATEROLE;

--查看用户jim追加的CREATEROLE权限。
gaussdb=# \du jim
           List of roles
 Role name | Attributes  | Member of
-----------+-------------+-----------
 jim       | Create role | {}

--锁定jim账户。
gaussdb=# ALTER USER jim ACCOUNT LOCK;

--解锁jim账户。
gaussdb=# ALTER USER jim ACCOUNT UNLOCK;

--修改用户密码。
gaussdb=# ALTER USER dim WITH PASSWORD '********';

--修改用户名。
gaussdb=# ALTER USER dim RENAME TO lisa;

--创建具有OPRADMIN权限的用户user1和具有SYSADMIN权限的用户user2。
gaussdb=# CREATE USER user1 WITH OPRADMIN PASSWORD '********';
gaussdb=# CREATE USER user2 WITH SYSADMIN PASSWORD '********';

--查看权限。
gaussdb=# \du user1
                  List of roles
 Role name |         Attributes          | Member of
-----------+-----------------------------+-----------
 user1     |        Operatoradmin        | {}

gaussdb=# \du user2
               List of roles
 Role name |       Attributes       | Member of
-----------+------------------------+-----------
 user2     |       Sysadmin         | {}

--创建具有CREATEDB权限的角色user3。
gaussdb=# CREATE USER user3 CREATEDB PASSWORD '********';
CREATE ROLE

--查看权限。
gaussdb=# \du user3
                List of roles
 Role name |       Attributes        | Member of
-----------+-------------------------+-----------
 user3     |       Create DB         | {}

--使用ADMIN子句创建新角色user4。
gaussdb=# CREATE USER user4 WITH  CREATEDB  ADMIN user3  PASSWORD '********';
CREATE ROLE
--查看权限。
gaussdb=# \du user3
                List of roles
 Role name |       Attributes        | Member of
-----------+-------------------------+-----------
 user3     |       Create DB         | {user4}

gaussdb=# \du user4
                List of roles
 Role name |       Attributes        | Member of
-----------+-------------------------+-----------
 user4     |       Create DB         | {}
```


版权所有 © 华为云计算技术有限公司

```
--删除角色。
gaussdb=# DROP USER user1;
gaussdb=# DROP USER user2;
gaussdb=# DROP USER user3;
gaussdb=# DROP USER user4;

--三权分立配置：设置参数enableSeparationOfDuty为on，则会开启三权分立配置，限制系统管理员的权限。不允许系统管理员创建或更改用户配置。

--查看三权分立是否打开。
gaussdb=# SHOW enableSeparationOfDuty;
 enableSeparationOfDuty
------------------------
 off
(1 row)
--切换到数据库omm用户后，执行以下语句修改参数。
gs_guc set -Z coordinator -Z datanode -N all -I all -c "enableSeparationOfDuty = on"

--设置成功后，重启数据库生效。
gs_om -t stop
gs_om -t start

--再次查看三权分立。
gaussdb=# SHOW enableSeparationOfDuty;
 enableSeparationOfDuty
------------------------
 on
(1 row)

--删除用户。
gaussdb=# DROP USER kim CASCADE;
gaussdb=# DROP USER jim CASCADE;
gaussdb=# DROP USER lisa CASCADE;
gaussdb=# DROP USER TOM CASCADE;
gaussdb=# DROP USER "TOM" CASCADE;

--设置用户密码有效期。
gaussdb=# CREATE USER user1 PASSWORD '********' PASSWORD EXPIRE INTERVAL 10 DAY;
```

相关链接

ALTER USER，**CREATE ROLE**，**DROP USER**

1.13.9.55 CREATE VIEW

功能描述

创建一个视图。视图与基本表不同，是一个虚拟的表。数据库中仅存放视图的定义，而不存放视图对应的数据，这些数据仍存放在原来的基本表中。若基本表中的数据发生变化，从视图中查询出的数据也随之改变。从这个意义上讲，视图就像一个窗口，透过它可以看到数据库中用户感兴趣的数据及变化。

- 将经常使用的数据定义为视图，可以将复杂的查询SQL语句进行封装。简化操作。
- 安全性，用户只能查询视图定义的数据。隐藏基表字段，保护数据库的数据结构。
- 简化用户权限的管理，只授予用户使用视图的权限。

注意事项

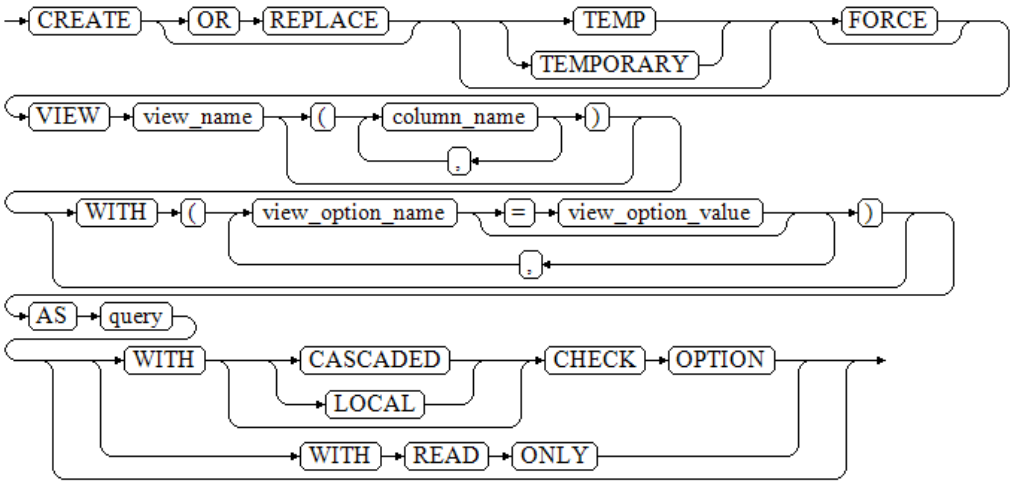
被授予CREATE ANY TABLE权限的用户，可以在public模式和用户模式下创建视图。

版权所有 © 华为云计算技术有限公司

非初始用户不允许对SYS、DBE_PERF以及INFORMATION_SCHEMA模式下的系统视图执行CREATE OR REPLACE操作。

语法格式

```
CREATE [ OR REPLACE ] [ TEMP | TEMPORARY ] [ FORCE ] VIEW view_name [ ( column_name [, ...] ) ]
    [ WITH ( {view_option_name [= view_option_value]} [, ... ] ) ]
    AS query
    [ WITH [ CASCADED | LOCAL ] CHECK OPTION  | WITH READ ONLY ];
```



说明

- 创建视图时使用WITH(security_barrier)可以创建一个相对安全的视图，避免攻击者利用低成本函数的RAISE语句打印出基表数据。
- 对嵌套视图中被依赖视图做修改列名、列类型以及删除列操作，不论是否开启失效参数enable_view_invalidation，均会导致依赖视图失效。

参数说明

- **OR REPLACE**

  可选。如果视图已存在，则重新定义。

- **TEMP | TEMPORARY**

  可选。创建一个临时视图。在当前会话结束时会自动删除掉视图。如果视图引用的任何表是临时表，视图将被创建为临时视图（不管SQL中有没有指定TEMP|TEMPORARY）。

  说明

  - 临时视图不支持自动重编译。
  - 视图失效重编译请参见《开发指南》中“特性使用指导 > 失效重编译”章节。

- **FORCE**

  可选。指定FORCE后，创建视图时如果视图依赖的对象不存在或缺少查询权限，该视图仍能够创建成功。

- **view_name**

  要创建的视图名称。可以用模式修饰。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_name**

版权所有 © 华为云计算技术有限公司

可选的名称列表，用作视图的字段名。如果没有给出，字段名取自查询中的字段名。

取值范围：字符串，要符合**标识符命名规范**。

- **view_option_name [= view_option_value]**

  该子句为视图指定一个可选的参数。

  目前view_option_name支持的参数仅有security_barrier和check_option。

  security_barrier：当VIEW视图提供行级安全时，应使用该参数。取值范围：Boolean类型（true、false）。

  check_option：控制更新视图的行为。取值范围：CASCADED、LOCAL。

- **query**

  为视图提供行和列的SELECT或VALUES语句。

须知

若query包含指定分区表分区的子句，创建视图会将所指定分区的OID直接写入到系统表中。如果使用导致指定分区的OID发生变更的分区DDL语法，如DROP/SPLIT/MERGE该分区，则会导致视图失效（参见《开发指南》中“特性使用指导 > 失效重编译”章节）。

- **WITH [CASCADED | LOCAL] CHECK OPTION**

  控制更新视图的行为，对视图的INSERT和UPDATE，要检查确保新行满足视图定义的条件，即新行可以通过视图看到。如果没有通过检查，则拒绝修改。如果没有添加该选项，则允许通过对视图的INSERT和UPDATE来创建该视图不可见的行。对视图的DELETE，如果要删除数据的基表在多表连接视图中重复，且重复的基表在视图中不都是键保留表，则指定WITH CHECK OPTION时不允许删除数据。WITH CHECK OPTION选项可以指定为CASCADED或LOCAL。

  CASCADED：检查该视图和所有底层视图定义的条件。如果仅声明了CHECK OPTION，没有声明LOCAL和CASCADED，默认是CASCADED。

  LOCAL：只检查视图本身直接定义的条件。若底层视图也定义了CHECK OPTION，则会检查底层视图定义的条件。

- **WITH READ ONLY**

  定义只读视图，无法对该视图进行插入、更新和删除操作。

版权所有 © 华为云计算技术有限公司

说明

- 对视图的更新、插入和删除的约束涉及到的一些概念，解释如下：
  - 连接视图：多张表JOIN创建的视图。
  - 保留键表：对多表连接视图进行插入、更新、删除受到键保留表的限制。在多表视图中，若源表的每一行与视图中的每一行一一对应， 而不存在源表中一行数据在JOIN连接后在视图中对应多行数据的情况，则源表为保留键表。
  - 当前仅支持通过JOIN条件进行判断目标表是否为保留键表，不支持等价类的方式（即通过JOIN条件和WHERE条件联合）判断目标表是否为保留键表。
  - 顶层与底层关系：视图可能有多层嵌套，如一个视图由一个或多个视图或子查询构成。将当前DML直接操作的视图称为顶层，将构成视图的表、视图、及WITH子句中的表、视图等称为对应的底层关系。
  - 可更新列：不是系统列或whole-row reference，直接引用基表中的用户列的列可更新。对于视图或表中的列是否为可更新列，可以通过查询information_schema.columns的is_updatable字段辅助确认。
  - 可更新视图：可以对视图做插入、更新、删除操作的，称为可更新视图。可更新视图不包含DISTINCT、GROUP BY、HAVING、LIMIT、OFFSET、UNPIVOT、FOR UPDATE、FETCH、START WITH CONNECT BY子句 ，不包含集合运算（UNION、INTERSECT、EXCEPT），并且不包含聚集函数、窗口函数、返回集合函数（array_agg、json_agg、generate_series等）、闪回。WITH CHECK OPTION选项只能定义在可更新的视图上。对于视图是否为可更新视图，可以通过查询information_schema.views中的is_updatable和is_insertable_into字段，或查询information_schema.tables.is_insertable_into字段辅助确认。其中，information_schema.views中的is_updatable返回是否可对视图做更新和删除操作，is_insertable_into返回是否可对视图插入数据，information_schema.tables.is_insertable_into返回是否可对关系插入数据。
- 如果一个可更新的视图拥有INSTEAD OF触发器或INSTEAD规则，则使用CHECK OPTION不会检查该视图上的条件。
- 如果一个可更新视图的某一底层视图拥有INSTEAD OF触发器，且该可更新视图定义了CASCADED类型的CHECK OPTION选项，那么非ORA兼容模式下会递归检查到带有INSTEAD OF触发器的这一底层视图的条件为止，并且不会检查带有触发器的底层视图上的条件。ORA兼容模式下则所有视图的CHECK OPTION选项都失效不进行检查。
- 如果一个可更新的视图或者任何它的底层关系存在INSTEAD规则，导致INSERT或UPDATE重写，那么使用CHECK OPTION将不会检查所有视图上的条件。
- 如果上层视图定义了CASCADED的CHECK OPTION，底层视图定义了LOCAL CHECK OPTION，底层视图的LOCAL CHECK OPTION会被上层视图的CASCADED CHECK OPTION覆盖。
- 如果指定了CHECK OPTION，无法对多表连接视图或多表连接子查询中的连接列进行插入、更新操作。
- 如果指定了CHECK OPTION，若多表连接视图或多表连接子查询中出现重复基表，且重复的基表不都是保留键表，则无法对该视图或子查询进行删除操作。

示例

- 普通视图

```
--创建test_tb1表，并向表中插入100条数据。
gaussdb=# CREATE TABLE test_tb1(col1 int, col2 int);
gaussdb=# INSERT INTO test_tb1 VALUES (generate_series(1,100),generate_series(1,100));
--创建一个col1小于3的视图。
gaussdb=# CREATE VIEW test_v1 AS SELECT * FROM test_tb1 WHERE col1 < 3;
--查看视图。
gaussdb=# SELECT * FROM test_v1;
 col1 | col2
------+------
    1 |    1
    2 |    2
(2 rows)
```


版权所有 © 华为云计算技术有限公司

```
--删除表和视图。
gaussdb=# DROP VIEW test_v1;
gaussdb=# DROP TABLE test_tb1;
```

- 只读视图

```
--创建test_tb表。
gaussdb=# CREATE TABLE test_tb(c1 int, c2 int);

--创建只读视图test_v。
gaussdb=# CREATE  VIEW test_v AS SELECT * FROM test_tb WITH READ ONLY;

--向视图中写入数据来判断视图是否是只读视图。
gaussdb=# INSERT INTO test_v VALUES (1, 1);
ERROR:  DML operation cannot be performed on a read-only view.

--删除表和视图。
gaussdb=# DROP VIEW test_v;
gaussdb=# DROP TABLE test_tb;
```

- REPLACE的应用。

```
--创建a，b表。
gaussdb=# CREATE TABLE a(c1 int, c2 int);
gaussdb=# CREATE TABLE b(c1 int, c2 int, c3 int);

--向a、b分别插入数据。
gaussdb=# INSERT INTO a VALUES (1, 1);
gaussdb=# INSERT INTO b VALUES (1, 1, 1);

--查看。
gaussdb=# SELECT * FROM a;
 c1 | c2
----+----
  1 |  1
(1 row)

gaussdb=# SELECT * FROM b;
 c1 | c2 | c3
----+----+----
  1 |  1 |  1
(1 row)

--创建视图a_view，这里视图不存在，相当于新建视图。
gaussdb=# CREATE OR REPLACE VIEW a_view AS SELECT * FROM a;

--查看视图a_view。
gaussdb=# SELECT * FROM a_view;
 c1 | c2
----+----
  1 |  1
(1 row)

--创建视图a_view，这里视图已存在，相当于重新定义视图
gaussdb=# CREATE OR REPLACE VIEW a_view AS SELECT * FROM b;

--再次查看视图a_view。
gaussdb=# SELECT * FROM a_view;
 c1 | c2 | c3
----+----+----
  1 |  1 |  1
(1 row)

--重建视图a_view，并删除c1列
gaussdb=# CREATE OR REPLACE VIEW a_view AS SELECT c2, c3 FROM b;

--再次查看视图a_view。
gaussdb=# SELECT * FROM a_view;
 c2 | c3
----+----
  1 |  1
(1 row)
```

 版权所有 © 华为云计算技术有限公司

```
--重建视图a_view，并修改列c2类型
gaussdb=# CREATE OR REPLACE VIEW a_view AS SELECT c2::TEXT, c3 FROM b;

--再次查看视图a_view。
gaussdb=# SELECT * FROM a_view;
 c2 | c3
----+----
 1  |  1
(1 row)

--重建视图a_view，并重命名c2列
gaussdb=# CREATE OR REPLACE VIEW a_view AS SELECT c2 AS c2_new, c3 FROM b;

--再次查看视图a_view。
gaussdb=# SELECT * FROM a_view;
 c2_new | c3
--------+----
      1 |  1
(1 row)

--删除视图和表。
gaussdb=# DROP VIEW a_view;
gaussdb=# DROP TABLE a;
gaussdb=# DROP TABLE b;
```

- 视图一次性入库

```
-- 创建视图，依赖的基表test_tb1当前不存在。
gaussdb=# CREATE FORCE VIEW test_v1 AS SELECT * FROM test_tb1 WHERE col1 < 3;
WARNING:  View "test_v1" created with compilation errors.
DETAIL:  Relation "test_tb1" does not exist on datanode.
WARNING:  View "test_v1" created with compilation errors.
DETAIL:  Relation "test_tb1" does not exist on datanode.
CREATE VIEW
--视图可以入库，状态为无效。
gaussdb=# SELECT c.relname, o.valid FROM pg_object o JOIN pg_class c ON c.oid = o.object_oid
WHERE c.relname = 'test_v1';
 relname | valid
---------+-------
 test_v1 | f
(1 row)

-- 创建test_tb1表，并重编译视图test_v1成功。
gaussdb=# CREATE TABLE test_tb1(col1 int, col2 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# ALTER VIEW test_v1 COMPILE;
ALTER VIEW
gaussdb=# SELECT c.relname, o.valid FROM pg_object o JOIN pg_class c ON c.oid = o.object_oid
WHERE c.relname = 'test_v1';
 relname | valid
---------+-------
 test_v1 | t
(1 row)

-- 查询视图。
gaussdb=# SELECT * FROM test_v1;
 col1 | col2
------+------
(0 rows)

-- 删除表和视图。
gaussdb=# DROP VIEW test_v1;
DROP VIEW
gaussdb=# DROP TABLE test_tb1;
DROP TABLE
```

- 临时视图

版权所有 © 华为云计算技术有限公司

```
--创建表和临时视图。
gaussdb=# CREATE TABLE test_tb2(c1 int, c2 int);
gaussdb=# CREATE TEMP VIEW test_v2 AS SELECT * FROM test_tb2;
--查看表和视图信息（临时表所属模式不是public，而是以pg_temp开头的模式）。
gaussdb=# \dv
                    List of relations
        Schema          |  Name   | Type | Owner | Storage
--------------------------+---------+------+-------+---------
 pg_temp_cn_5001_1_2_1742 | test_v2 | view | Mike  |
(1 row)

--结束当前会话重新登录后，再次查看临时视图已经被删除。
gaussdb=# \dv
No relations found.

--删除视图和表。
gaussdb=# DROP TABLE test_tb2;
```

- 对视图进行插入、更新、删除

```
--创建单表视图
gaussdb=# CREATE TABLE t_view_iud1 (x int, y int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# INSERT INTO t_view_iud1 VALUES (11, 11);
INSERT 0 1
gaussdb=# CREATE VIEW vt AS SELECT * FROM t_view_iud1;
CREATE VIEW
gaussdb=# CREATE VIEW vt_wco AS SELECT * FROM t_view_iud1 WHERE x > 5 WITH CHECK OPTION;
CREATE VIEW

--通过information_schema.columns查询视图中的列是否可更新
gaussdb=# SELECT table_schema, table_name, column_name, is_updatable FROM
information_schema.columns WHERE table_schema = current_schema AND table_name = 'vt';
 table_schema | table_name | column_name | is_updatable
--------------+------------+-------------+--------------
 public       | vt         | y           | YES
 public       | vt         | x           | YES
(2 rows)

--通过对视图插入、更新数据，如果视图创建时指定了WITH CHECK OPTION， 更新视图前会对数据做相
应检查
gaussdb=# INSERT INTO vt VALUES (1, 1);
INSERT 0 1
gaussdb=# INSERT INTO vt_wco VALUES (1, 1);
ERROR:  New row violates WITH CHECK OPTION for view "vt_wco".
DETAIL:  Failing row contains (1, 1).
gaussdb=# UPDATE vt SET y = 121 WHERE y = 1;
UPDATE 1
gaussdb=# UPDATE vt_wco SET y = 6 WHERE y = 11;
UPDATE 1
gaussdb=# DELETE FROM vt WHERE y =11;
DELETE 0

--创建多层嵌套视图，底层关系中指定了WITH CHECK OPTION，对顶层视图做DML操作时也会检查相应
条件
gaussdb=# CREATE VIEW vvtt AS SELECT * FROM vt_wco;
CREATE VIEW
gaussdb=# INSERT INTO vvtt VALUES (1, 1), (2, 2);
ERROR:  New row violates WITH CHECK OPTION for view "vt_wco".
DETAIL:  Failing row contains (1, 1).

--通过information_schema.tables和information_schema.views查询视图是否可插入、更新
gaussdb=# SELECT table_schema, table_name, is_insertable_into
        FROM information_schema.tables
        WHERE table_schema = current_schema AND table_name = 'vvtt';
 table_schema | table_name | is_insertable_into
--------------+------------+--------------------
 public       | vvtt       | YES
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT table_name, is_updatable, check_option
          FROM information_schema.views
          WHERE table_schema = current_schema AND table_name = 'vvtt';
 table_name | is_updatable | check_option
------------+--------------+--------------
 vvtt       | YES          | NONE
(1 row)

gaussdb=# SELECT table_name, is_updatable, check_option FROM information_schema.views WHERE
table_schema = current_schema AND table_name = 'vt_wco';
 table_name | is_updatable | check_option
------------+--------------+--------------
 vt_wco     | YES          | CASCADED
(1 row)

--删除视图和表
gaussdb=# DROP VIEW vvtt, vt, vt_wco CASCADE;
DROP VIEW
gaussdb=# DROP TABLE t_view_iud1;
DROP TABLE
```

- 不支持等价类的方式（即通过JOIN条件和WHERE条件联合）判断目标表是否为保留键表。

```
-- 创建测试表。
gaussdb=# CREATE TABLE A(col1 int, col2 int, col3 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'col1' as the distribution column by
default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE B(col1 int, col2 int, col3 int unique);
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "b_col3_key" for table "b"
CREATE TABLE
gaussdb=# CREATE UNIQUE INDEX index_1 ON A(col1, col2);
CREATE INDEX

-- 执行更新语句，因为通过A.col1 = B.col1无法判断出表A为保留键表，更新失败。
gaussdb=# UPDATE (
gaussdb(#     SELECT A.col1, A.col2, A.col3
gaussdb(#     FROM A INNER JOIN B ON A.col1 = B.col1
gaussdb(#     WHERE B.col3 = 3
gaussdb(# )
gaussdb-# SET col1 = 1, col2 = 2, col3 = 3;
ERROR:  Failed to modify a column which maps to a non key-preserved table.

-- 删除测试表。
gaussdb=# DROP TABLE A;
DROP TABLE
gaussdb=# DROP TABLE B;
DROP TABLE
```

相关链接

ALTER VIEW，**DROP VIEW**

1.13.9.56 CREATE WEAK PASSWORD DICTIONARY

功能描述

创建弱口令字典。弱口令字典默认为空，用户可以通过此语法向系统表gs_global_config中新增一条或多条弱口令。

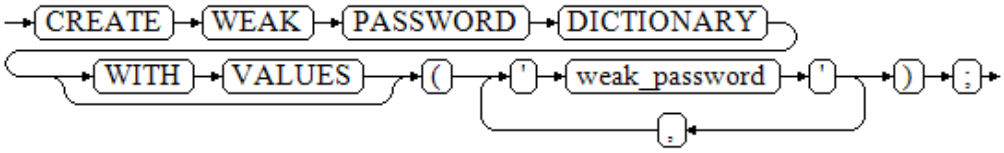
版权所有 © 华为云计算技术有限公司

注意事项

- 仅初始用户、系统管理员和安全管理员可以执行。
- 弱口令字典中的口令存放在gs_global_config系统表中。
- 弱口令字典默认为空，用户通过本语法可以向系统表gs_global_config中新增一条或多条弱口令。
- 当用户尝试通过本语法插入gs_global_config表中已存在的弱口令时，会只在表中保留一条该弱口令。

语法格式

```
CREATE WEAK PASSWORD DICTIONARY
    [WITH VALUES] ( {'weak_password'} [, ...] );
```



参数说明

weak_password

弱口令。

密码取值范围：字符串。

示例

```
--向gs_global_config系统表中插入单个弱口令。
gaussdb=# CREATE WEAK PASSWORD DICTIONARY WITH VALUES ('********');

--查看gs_global_config系统表现有的弱口令。
gaussdb=# SELECT * FROM gs_global_config WHERE NAME LIKE 'weak_password';
      name       |    value
-----------------+--------------
 weak_password   | ********
(1 rows)

--向gs_global_config系统表中插入多个弱口令。
gaussdb=# CREATE WEAK PASSWORD DICTIONARY WITH VALUES ('********'),('********');

--再次查看gs_global_config系统表弱口令（弱密码的*不代表密码内容）。
gaussdb=# SELECT * FROM gs_global_config WHERE NAME LIKE 'weak_password';
      name       |     value
-----------------+----------------
 weak_password   | ********
 weak_password   | ********
 weak_password   | ********
(3 rows)

--清空gs_global_config系统表中所有弱口令。
gaussdb=# DROP WEAK PASSWORD DICTIONARY;

--查看现有弱口令。
gaussdb=# SELECT * FROM gs_global_config WHERE NAME LIKE 'weak_password';
 name | value
------+-------
(0 rows)
```


版权所有 © 华为云计算技术有限公司

相关链接

DROP WEAK PASSWORD DICTIONARY

1.13.9.57 CURSOR

功能描述

用于创建一个游标，从一个查询里检索出指定的几行数据。

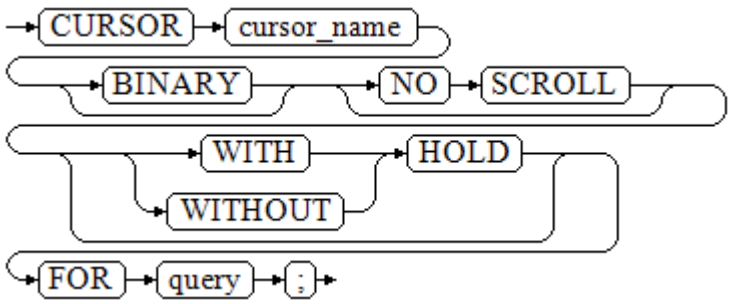
为了处理SQL语句，存储过程线程分配一段内存区域来保存上下文联系。游标是指向上下文区域的句柄或指针。借助游标，存储过程可以控制上下文区域的变化。

注意事项

- 游标命令只能在事务块中进行使用。
- 通常游标和SELECT一样返回文本格式。因为数据在系统内部是用二进制格式存储的，系统必须对数据做一定转换以生成文本格式。一旦数据是以文本形式返回，客户端应用需要把它们转换成二进制进行操作。使用FETCH语句，游标可以返回文本或二进制格式。
- 谨慎使用二进制游标。文本格式一般比对应的二进制格式占用的存储空间大。二进制游标返回内部二进制形态的数据，可能更易于操作。如果想以文本方式显示数据，则以文本方式检索会为用户节约很多客户端的工作。比如，如果查询从某个整数列返回1，在缺省的游标里将获得一个字符串1，但在二进制游标里将得到一个4字节的包含该数值内部形式的数值（大端顺序）。
- 静态游标支持定义参数，参数支持默认值。但是参数默认值不能应用PACKAGE外的变量，如果需要使用，可以在包内定义一个变量，再将包外变量赋值给该变量，使用该变量作为默认值。
- 游标的定义会产生查询对象，PACKAGE或存储过程暂不支持建立对查询对象的依赖关系。

语法格式

```
CURSOR cursor_name
    [ BINARY ]  [ NO SCROLL ]  [ { WITH | WITHOUT } HOLD ]
    FOR query;
```



参数说明

- **cursor_name**

  将要创建的游标名。

  取值范围：字符串，要符合**标识符命名规范**。

版权所有 © 华为云计算技术有限公司

- **BINARY**

  指定游标以二进制格式返回数据，而不是文本格式返回数据。

- **NO SCROLL**

  声明游标检索数据行的方式。

  - NO SCROLL：声明该游标不能用于以倒序的方式检索数据行。
  - 未指定：根据执行计划的不同，自动判断该游标是否可以用于以倒序的方式检索数据行。

- **WITH HOLD | WITHOUT HOLD**

  当创建游标的事务结束后，游标是否能继续使用。未指定WITH HOLD或WITHOUT HOLD时，默认行为是WITHOUT HOLD。

  - WITH HOLD：在创建游标的事务结束后，该游标仍可继续使用。
  - WITHOUT HOLD：在创建游标的事务之外不能再继续使用该游标，游标将在事务结束时被自动关闭。

  跨节点事务不支持WITH HOLD（例如在多Coordinator部署集群中所创建的含有DDL的事务属于跨节点事务）。

- **query**

  使用SELECT或VALUES子句指定游标返回的行。

  取值范围：SELECT或VALUES子句。

示例

示例1：

```
--建表并插入数据。
gaussdb=# CREATE TABLE tbl_test(c1 int);
gaussdb=# INSERT INTO tbl_test VALUES (generate_series(1,20));

--建立一个名为cursor1的游标。
gaussdb=# BEGIN;
gaussdb=# CURSOR cursor1 FOR SELECT * FROM tbl_test ORDER BY 1;

--使用FETCH命令检索3行数据。
gaussdb=# FETCH FORWARD 3 FROM cursor1;
 c1
----
  1
  2
  3
(3 rows)

--使用MOVE命令使游标向后移动2行，不返回结果。
gaussdb=# MOVE FORWARD 2 FROM cursor1;
MOVE 2

--使用FETCH命令检索两行数据。
gaussdb=# FETCH FORWARD 2 FROM cursor1;
 c1
----
  6
  7
(2 rows)

--关闭游标并结束事务。
gaussdb=# CLOSE cursor1;
gaussdb=# END;
```

示例2：带WITH HOLD属性的游标：

版权所有 © 华为云计算技术有限公司

```
--建立一个名为cursor2的WITH HOLD游标。
gaussdb=# BEGIN;
gaussdb=# CURSOR cursor2 WITH HOLD FOR SELECT * FROM tbl_test ORDER BY 1;

--使用FETCH命令检索3行数据。
gaussdb=# FETCH FORWARD 3 FROM cursor2;
 c1
----
  1
  2
  3
(3 rows)

--和普通的游标不一样，结束事务后，带WITH HOLD属性的游标依然可以使用。
gaussdb=# END;
gaussdb=# FETCH FORWARD 2 FROM cursor2;
 c1
----
  4
  5
(2 rows)

--关闭游标。
gaussdb=# CLOSE cursor2;

--删除。
gaussdb=# DROP TABLE tbl_test;
```

相关链接

FETCH，**CLOSE**，**MOVE**

1.13.10 D

1.13.10.1 DEALLOCATE

功能描述

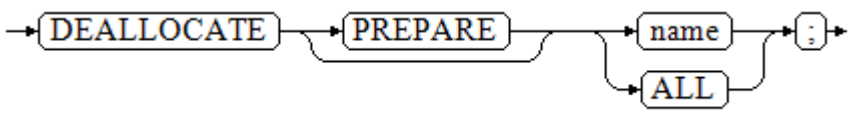
从数据库中删除一个预备语句。

注意事项

使用DEALLOCATE可以手动释放通过PREPARE创建的预处理语句。如果未显式释放，预处理语句会在会话结束时自动清除。

语法格式

```
DEALLOCATE [ PREPARE ] { name | ALL };
```



参数说明

- **PREPARE**
  可选，仅用于语法兼容性，实际执行时会被忽略。

版权所有 © 华为云计算技术有限公司

- **name**

  将要删除的预备语句。

- **ALL**

  删除所有预备语句。

示例

```
--查看现有预备语句。
gaussdb=# SELECT name, statement, parameter_types FROM pg_prepared_statements;
 name | statement | parameter_types
------+-----------+-----------------
(0 rows)

--创建q1,q2,q3，q4四个预备语句。
gaussdb=# PREPARE q1 AS SELECT 1 AS a;
PREPARE
gaussdb=# PREPARE q2 AS SELECT 1 AS a;
PREPARE
gaussdb=# PREPARE q3 AS SELECT 1 AS a;
PREPARE
gaussdb=# PREPARE q4 AS SELECT 1 AS a;
PREPARE

--再次查看现有预备语句。
gaussdb=# SELECT name, statement, parameter_types FROM pg_prepared_statements;
 name |          statement          | parameter_types
------+-----------------------------+-----------------
 q1   | PREPARE q1 AS SELECT 1 AS a; | {}
 q4   | PREPARE q4 AS SELECT 1 AS a; | {}
 q3   | PREPARE q3 AS SELECT 1 AS a; | {}
 q2   | PREPARE q2 AS SELECT 1 AS a; | {}
(4 rows)

--删除q4预备语句并查看剩余预备语句。
gaussdb=# DEALLOCATE q4;
gaussdb=# SELECT name, statement, parameter_types FROM pg_prepared_statements;
 name |          statement          | parameter_types
------+-----------------------------+-----------------
 q1   | PREPARE q1 AS SELECT 1 AS a; | {}
 q3   | PREPARE q3 AS SELECT 1 AS a; | {}
 q2   | PREPARE q2 AS SELECT 1 AS a; | {}
(3 rows )

--删除所有预备语句并查看剩余预备语句。
gaussdb=# DEALLOCATE ALL;
DEALLOCATE ALL
gaussdb=# SELECT name, statement, parameter_types FROM pg_prepared_statements;
 name | statement | parameter_types
------+-----------+-----------------
(0 rows)
```

1.13.10.2 DECLARE

功能描述

DECLARE既可以定义一个游标，用于在一个大的查询里面检索少数几行数据，也可以用于声明匿名块（但在声明匿名块时，需放置在匿名块的语句之前）。

本节主要描述定义为游标的用法，定义为匿名块的用法请参见**BEGIN**章节。

为了处理SQL语句，存储过程线程分配一段内存区域进行保存上下文联系，游标是指向上下文区域的句柄或指针。借助游标，存储过程可以控制上下文区域的变化。

版权所有 © 华为云计算技术有限公司

通常游标和SELECT一样返回文本格式。因为数据在系统内部是用二进制格式存储的，系统必须对数据做一定转换以生成文本格式。一旦数据是以文本形式返回，客户端应用需要把文本格式转换成二进制进行操作。使用FETCH语句，游标可以返回文本格式或二进制格式。

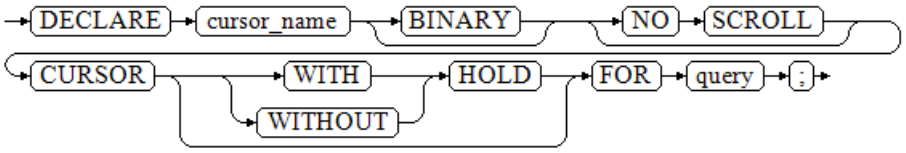
注意事项

- 游标命令只能在事务块里使用。
- 谨慎使用二进制游标。文本格式一般比对应的二进制格式占用的存储空间大。二进制游标返回内部二进制形态的数据，可能更易于操作。如果想以文本方式显示数据，则以文本方式检索会为用户节约很多客户端的工作。比如，如果查询从某个整数列返回1，在缺省的游标里将获得一个字符串1，但在二进制游标里将得到一个4字节的包含该数值内部形式的数值（大端顺序）。

语法格式

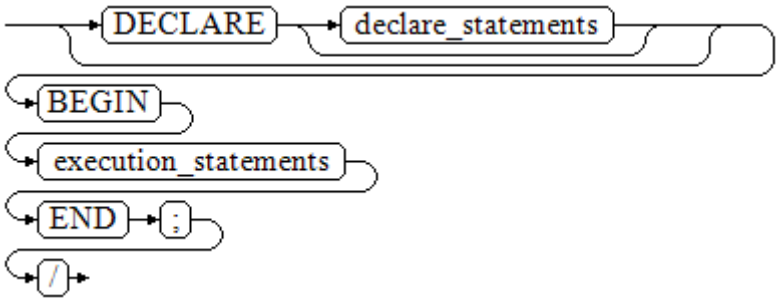
- 定义游标

```
DECLARE cursor_name [ BINARY ] [ NO SCROLL ]
    CURSOR [ { WITH | WITHOUT } HOLD ] FOR query;
```



- 开启匿名块

```
[DECLARE [declare_statements]]
BEGIN
execution_statements
END;
/
```



参数说明

- **cursor_name**

  将要创建的游标名。

  取值范围：字符串，要符合**标识符命名规范**。

- **BINARY**

  指定游标以二进制格式返回数据，而不是文本格式返回数据。

- **NO SCROLL**

  声明游标检索数据行的方式。

  - NO SCROLL：声明该游标不能用于以倒序的方式检索数据行。
  - 未指定：根据执行计划的不同，自动判断该游标是否可以用于以倒序的方式检索数据行。

版权所有 © 华为云计算技术有限公司

- **WITH HOLD | WITHOUT HOLD**

  当创建游标的事务结束后，游标是否能继续使用。未指定WITH HOLD或WITHOUT HOLD时，默认行为是WITHOUT HOLD。

  - WITH HOLD：在创建游标的事务结束后，该游标仍可继续使用。
  - WITHOUT HOLD：在创建游标的事务之外不能再继续使用该游标，游标将在事务结束时被自动关闭。

  须知

  声明为WITH HOLD的游标，在事务结束时，会缓存游标所有数据，若游标数据量较大，此过程耗时可能较长。

- **query**

  使用SELECT或VALUES子句指定游标返回的行。

  取值范围：SELECT或VALUES子句。

- **declare_statements**

  声明变量，包括变量名和变量类型，如“sales_cnt int”。

- **execution_statements**

  匿名块中要执行的语句。

  取值范围：已存在的函数名称。

示例

- 定义游标

```
--创建表test，并向表中插入10条数据。
gaussdb=# CREATE TABLE test(c1 int, c2 int);
gaussdb=# INSERT INTO test VALUES (generate_series(1,10),generate_series(1,10));

--开始一个事务，建立一个名为cursor1的游标。
gaussdb=# START TRANSACTION;
gaussdb=# DECLARE cursor1 CURSOR FOR SELECT * FROM test ORDER BY 1;

--指定游标从关联位置开始检索3行数据。
gaussdb=# FETCH FORWARD 3 FROM cursor1;
 c1 | c2
----+----
  1 |  1
  2 |  2
  3 |  3
(3 rows)

--关闭游标并提交事务。
gaussdb=# CLOSE cursor1;
gaussdb=# END;

--删除表。
gaussdb=# DROP TABLE test;
```

- 开启匿名块

```
--声明变量并输出。
gaussdb=# DECLARE num INTEGER := 10;
BEGIN
    dbe_output.print_line(num);
END;
/
```


版权所有 © 华为云计算技术有限公司

相关链接

BEGIN，**FETCH**

1.13.10.3 DELETE

功能描述

从指定的表里删除满足WHERE子句的行。如果WHERE子句不存在，将删除表中所有行，结果只保留表结构。

注意事项

- 表的所有者、被授予表DELETE权限的用户或被授予DELETE ANY TABLE权限的用户有权删除表中数据，当三权分立开关关闭时，系统管理员默认拥有此权限。同时在进行DELETE操作时，用户也必须拥有USING子句所引用的表以及condition上读取的表的SELECT权限。
- 对于行存复制表，仅支持在含有主键约束的场景下或者执行计划能下推的场景下的DELETE操作。
- 对于子查询是Stream计划的DELETE语句，不支持删除的行数据同时进行UPDATE操作。

语法格式

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
DELETE [/*+ plan_hint */] [FROM] [ ONLY ] {table_name [ * ] subquery [ [ AS ] alias ] | view_name [ [ AS ] alias ]}
    [ USING using_list ]
    [ WHERE condition | WHERE CURRENT OF cursor_name ]
    [ ORDER BY {expression [  ASC | DESC | USING operator ] } ]
    [ LIMIT row_count ]
    [ RETURNING { * | { output_expr [ [ AS ] output_name ] } [, ...] } ];
```

- with_query的格式为：

  ```
  with_query_name [ ( column_name [, ...] ) ] AS [ [ NOT ] MATERIALIZED ]
  ( {select | values | insert | update | delete} )
  ```

- subquery的格式为：

  ```
  [ WITH [ RECURSIVE ] with_query [, ...] ]
  SELECT [/*+ plan_hint */] [ ALL ]
  { * | {expression [ [ AS ] output_name ]} [, ...] }
  [ into_option ]
  [ FROM from_item [, ...] ]
  [ WHERE condition ]
  [ [ START WITH condition ] CONNECT BY [NOCYCLE] condition [ ORDER SIBLINGS BY expression ] ]
  [ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS
  { FIRST | LAST } ]} [, ...] ]
  [ OFFSET start [ ROW | ROWS ] ]
  [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]
  [ into_option ];
  ```

  其中，指定子查询源from_item可以为：

  ```
  {[ ONLY ] {table_name | view_name} [ * ] [ partition_clause ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  |( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
  |with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  |from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]}
  ```

参数说明

- **WITH [RECURSIVE] with_query [, ...]**

版权所有 © 华为云计算技术有限公司

声明一个或多个可以在主查询中通过名称引用的子查询，相当于临时表。如果指定了RECURSIVE，那么允许SELECT子查询通过名称引用其本身。

- with_query_name：指定子查询生成的结果集名称，在查询中可使用该名称访问子查询的结果集。
- column_name：指定子查询结果集中显示的列名。
- 每个子查询可以是SELECT、VALUES、INSERT、UPDATE或DELETE语句。
- 用户可以使用MATERIALIZED或NOT MATERIALIZED对CTE进行修饰。
  - 如果声明为MATERIALIZED，WITH查询将被物化，生成一个子查询结果集的复制，在引用处直接查询该复制，因此WITH子查询无法和主干SELECT语句进行联合优化（如：谓词下推、等价类传递等），对于此类场景可以使用NOT MATERIALIZED进行修饰，如果WITH子查询语义上可以作为子查询内联执行，则可以进行上述优化。
  - 如果用户没有显式指定物化属性则遵守以下规则：如果CTE只在所属主干语句中被引用一次，且语义上支持内联执行，则会被改写为子查询内联执行，否则以CTE Scan的方式物化执行。

- **plan_hint**

  以/*+ *plan_hint* */的形式在DELETE关键字后，用于对DELETE对应的语句块生成的计划进行Hint调优，具体用法请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。每条语句中只有第一个/*+ *plan_hint* */注释块会作为Hint生效，里面可以写多条Hint。

- **ONLY**

  如果指定ONLY则只有该表被删除；如果未声明，则该表和对应的所有子表都将被删除。

- **table_name**

  目标表的名称，可以使用模式进行具体指定。

  取值范围：已存在的表名。

  说明

  支持使用DATABASE LINK方式对远程表进行操作，使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

- **subquery**

  删除目标对象可以是子查询，在对子查询中的数据进行删除时，会将子查询当成一个临时视图，支持在子查询后面加CHECK OPTION选项。

  如果子查询中只有一张表，则对该表做删除；如果子查询中有多张表或有嵌套关系，则通过判断是否有保留键表确定是否可删除。关于保留键表和WITH CHECK OPTION请参见**CREATE VIEW**。

- **view_name**

  目标视图的名称。

版权所有 © 华为云计算技术有限公司

说明

对视图和子查询的删除，存在如下约束：

- 只有直接引用基表用户列的列可进行DELETE操作。
- 子查询或视图必须至少包含一个可更新列，关于可更新列请参见**CREATE VIEW**。
- 不支持在顶层包含DISTINCT、GROUP BY、HAVING、LIMIT、OFFSET子句的视图和子查询。
- 不支持在顶层包含集合运算（UNION、INTERSECT、EXCEPT、MINUS）的视图和子查询。
- 不支持目标列表中包含聚集函数、窗口函数、返回集合函数（array_agg、json_agg、generate_series等）的视图和子查询。
- 不支持仅带有BEFORE/AFTER触发器，没有INSTEAD OF触发器或INSTEAD规则的视图。
- 视图和子查询中支持的表类型包括普通表、临时表、全局临时表、分区表、二级分区表、Ustore表、Astore表。
- 连接视图或子查询只能对视图或子查询中的保留键表做删除操作，如果只存在一张保留键表，则删除该表数据，如果存在多张保留键表，仅删除FROM后的第一张保留键表的数据。
- 如果连接视图或子查询中指定了CHECK OPTION选项，且基表重复，重复的基表与视图或子查询中保留键表属性不一致，则无法从连接视图或子查询中删除行。关于保留键表请参见**CREATE VIEW**。
- 不支持对系统视图进行DELETE操作。
- 不支持多表删除功能。

- **alias**

  目标表的别名。

  取值范围：字符串，要符合**标识符命名规范**。

- **using_list**

  using子句。

- **condition**

  一个返回布尔值的表达式，用于判断需要被删除的行。建议不要使用INT等数值类型作为condition，因为INT等数值类型可以隐式转换为布尔值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

- **WHERE CURRENT OF cursor_name**

  当游标指向表的某一行时，可以使用此语法删除游标当前指向的行。使用限制及约束请参见**UPDATE**。

- **ORDER BY**

  具体信息请参见**SELECT**。

- **LIMIT**

  具体信息请参见**SELECT**。

  说明

  当需要进行DELETE操作的表为复制表时，不支持指定使用LIMIT关键字。

- **output_expr**

  DELETE命令删除行之后计算输出结果的表达式，该表达式可以使用表的任意字段，可以使用*返回被删除行的所有字段。

- **output_name**

版权所有 © 华为云计算技术有限公司

一个字段的输出名称。

取值范围：字符串，要符合**标识符命名规范**。

示例

- 删除部分数据

```
--建表。
gaussdb=# CREATE TABLE test_t1(col1 INT,col2 INT);
gaussdb=# INSERT INTO test_t1 VALUES (1, 1), (2, 2), (3, 3), (4, 4), (4, 6);

--删除表中部分记录。
gaussdb=# DELETE FROM test_t1 WHERE col1 = 4;

--查询。
gaussdb=# SELECT * FROM test_t1;
 col1 | col2
------+------
    1 |    1
    2 |    2
    3 |    3
(3 rows)
```

- 删除所有数据

```
--删除所有的数据。
gaussdb=# DELETE FROM test_t1;

--查询。
gaussdb=# SELECT * FROM test_t1;
 col1 | col2
------+------
(0 rows)

--删除表。
gaussdb=# DROP TABLE test_t1;
```

- WITH [RECURSIVE] with_query [, ...]

```
--创建学生表。
gaussdb=# CREATE TABLE student(id INT,name varchar(50));

--创建成绩表。
gaussdb=# CREATE TABLE grade(id INT,score CHAR);

gaussdb=# INSERT INTO student VALUES (1, 'tom'), (2, 'jerry'), (3, 'david');
gaussdb=# INSERT INTO grade VALUES (1, 'A'), (2, 'B'), (3, 'b');

--在学生表中删除id=2的数据同时删除该学生在成绩表中的数据。
gaussdb=# WITH del_stu AS(DELETE FROM student WHERE id = 2 RETURNING id)
    DELETE FROM grade WHERE id = (SELECT id FROM del_stu);

--查询数据。
gaussdb=# SELECT * FROM student;
 id | name
----+-------
  1 | tom
  3 | david
(2 rows)

gaussdb=# SELECT * FROM grade;
 id | score
----+-------
  1 | A
  3 | b
(2 rows)

--删除表。
gaussdb=# DROP TABLE grade;
gaussdb=# DROP TABLE student;
```


版权所有 © 华为云计算技术有限公司

- 删除子查询

```
--创建Schema。
gaussdb=# CREATE SCHEMA del_subqry;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = 'del_subqry';
SET

--创建表并插入数据。
gaussdb=# CREATE TABLE t1 (x1 int, y1 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE t2 (x2 int PRIMARY KEY, y2 int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t2_pkey" for table "t2"
CREATE TABLE
gaussdb=# CREATE TABLE tdata (x INT PRIMARY KEY, y INT);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "tdata_pkey" for table "tdata"
CREATE TABLE
gaussdb=# CREATE TABLE tinfo (z INT PRIMARY KEY, comm VARCHAR2(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "tinfo_pkey" for table "tinfo"
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4
gaussdb=# INSERT INTO t2 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4
gaussdb=# INSERT INTO tdata VALUES (1, 1), (2, 2), (3, 3);
INSERT 0 3
gaussdb=# INSERT INTO tinfo VALUES (1,'one'), (2, 'two'), (3, 'three'), (5, 'wrong three');
INSERT 0 4

--通过子查询删除t1中的数据。
gaussdb=# DELETE FROM (SELECT * FROM t1) where y1 = 3;
DELETE 1

--子查询带READ ONLY，无法删除数据。
gaussdb=# DELETE FROM (SELECT * FROM t1 WITH READ ONLY) WHERE y1 = 1;
ERROR:  DML operation cannot be performed on a read-only view.

--对多表连接的子查询删除。
gaussdb=# SELECT * FROM t1, t2 WHERE x1 = x2;
 x1 | y1 | x2 | y2
----+----+----+----
  1 |  1 |  1 |  1
  2 |  2 |  2 |  2
  5 |  5 |  5 |  5
(3 rows)

gaussdb=# DELETE FROM (SELECT * FROM t1, t2 WHERE x1 = x2) WHERE y2 = 5;
DELETE 1
gaussdb=# SELECT * FROM t1, t2 WHERE x1 = x2;
 x1 | y1 | x2 | y2
----+----+----+----
  1 |  1 |  1 |  1
  2 |  2 |  2 |  2
(2 rows)

--子查询带CHECK OPTION，tdata表重复，其中td1不是保留键表，td2是保留键表。
gaussdb=# DELETE FROM (SELECT td1.x x1, td1.y y1, td2.x x2, td2.y y2 FROM tdata td1, tdata td2,
tinfo WHERE td2.y=tinfo.z AND td1.x=td2.y WITH CHECK OPTION) WHERE y1 = 2;
ERROR:  Failed to delete from view with no key-preserved table.

--不带CHECK OPTION，创建同样结构的子查询，删除成功。
gaussdb=# DELETE FROM (SELECT td1.x x1, td1.y y1, td2.x x2, td2.y y2 FROM tdata td1, tdata td2,
tinfo WHERE td2.y=tinfo.z AND td1.x=td2.y) WHERE y1 = 2;
DELETE 1
--删除Schema。
gaussdb=# RESET CURRENT_SCHEMA;
RESET
gaussdb=# DROP SCHEMA del_subqry CASCADE;
```


版权所有 © 华为云计算技术有限公司

```
NOTICE:  drop cascades to 5 other objects
DETAIL:  drop cascades to table del_subqry.t1
drop cascades to table del_subqry.t2
drop cascades to table del_subqry.tdata
drop cascades to table del_subqry.tinfo
drop cascades to view del_subqry.vv_wco
DROP SCHEMA
```

- 删除视图

```
--创建Schema。
gaussdb=# CREATE SCHEMA del_view;
CREATE SCHEMA。
gaussdb=# SET CURRENT_SCHEMA = 'del_view';
SET

--创建表并插入数据。
gaussdb=# CREATE TABLE t1 (x1 int, y1 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE t2 (x2 int PRIMARY KEY, y2 int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t2_pkey" for table "t2"
CREATE TABLE
gaussdb=# CREATE TABLE tdata (x INT PRIMARY KEY, y INT);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "tdata_pkey" for table "tdata"
CREATE TABLE
gaussdb=# CREATE TABLE tinfo (z INT PRIMARY KEY, comm VARCHAR2(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "tinfo_pkey" for table "tinfo"
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4
gaussdb=# INSERT INTO t2 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4
gaussdb=# INSERT INTO tdata VALUES (1, 1), (2, 2), (3, 3);
INSERT 0 3
gaussdb=# INSERT INTO tinfo VALUES (1,'one'), (2, 'two'), (3, 'three'), (5, 'wrong three');
INSERT 0 4

--创建单表视图。
gaussdb=# CREATE VIEW v_del1 AS SELECT * FROM t1;
CREATE VIEW
gaussdb=# CREATE VIEW v_del_read AS SELECT * FROM t1 WITH READ ONLY;
CREATE VIEW

--通过视图删除t1中的数据。
gaussdb=# DELETE FROM v_del1 where y1 = 3;
DELETE 1

--视图带READ ONLY，无法删除数据。
gaussdb=# DELETE FROM v_del_read WHERE y1 = 1;
ERROR:  DML operation cannot be performed on a read-only view.

--创建多表连接视图。
gaussdb=# CREATE VIEW vvt1t2 AS SELECT * FROM t1, t2 WHERE x1 = x2;
CREATE VIEW
gaussdb=# CREATE VIEW vv_dup AS SELECT td1.x x1, td1.y y1, td2.x x2, td2.y y2 FROM tdata td1,
tdata td2, tinfo WHERE td2.y=tinfo.z AND td1.x=td2.y;
CREATE VIEW
gaussdb=# CREATE VIEW vv_dup_wco AS SELECT td1.x x1, td1.y y1, td2.x x2, td2.y y2 FROM tdata td1,
tdata td2, tinfo WHERE td2.y=tinfo.z AND td1.x=td2.y WITH CHECK OPTION;
CREATE VIEW

--对多表连接的视图做删除操作。
gaussdb=# SELECT * FROM vvt1t2;
 x1 | y1 | x2 | y2
----+----+----+----
  1 |  1 |  1 |  1
  2 |  2 |  2 |  2
  5 |  5 |  5 |  5
(3 rows)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DELETE FROM vvt1t2 WHERE y2 = 5;
DELETE 1

gaussdb=# SELECT * FROM vvt1t2;
 x1 | y1 | x2 | y2
----+----+----+----
  1 |  1 |  1 |  1
  2 |  2 |  2 |  2
(2 rows)

--视图带CHECK OPTION，tdata表重复，其中td1不是保留键表，td2是保留键表。
gaussdb=# DELETE FROM vv_dup_wco WHERE y1 = 2;
ERROR:  Failed to delete from view with no key-preserved table.

--不带CHECK OPTION，创建同样结构的视图，删除成功。
gaussdb=# DELETE FROM vv_dup WHERE y1 = 2;
DELETE 1

--删除Schema。
gaussdb=# RESET CURRENT_SCHEMA;
RESET
gaussdb=# DROP SCHEMA del_view CASCADE;
NOTICE:  drop cascades to 9 other objects
DETAIL:  drop cascades to table del_view.t1
drop cascades to table del_view.t2
drop cascades to table del_view.tdata
drop cascades to table del_view.tinfo
drop cascades to view del_view.v_del1
drop cascades to view del_view.v_del_read
drop cascades to view del_view.vvt1t2
drop cascades to view del_view.vv_dup
drop cascades to view del_view.vv_dup_wco
DROP SCHEMA
```

优化建议

如果需要删除表中的所有记录，建议使用**TRUNCATE**。

1.13.10.4 DO

功能描述

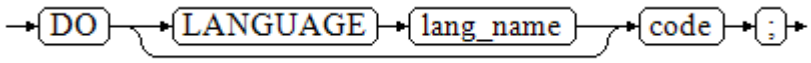
执行匿名代码块。代码块被看作是没有参数的一段函数体，返回值类型是VOID。其解析和执行同时进行。

注意事项

- 程序语言在使用之前，必须通过命令CREATE LANGUAGE安装到当前的数据库中。plpgsql是默认的安装语言，其他语言安装时必须指定。
- 如果语言是不受信任的，用户必须有使用程序语言的USAGE权限，或者是系统管理员权限。

语法格式

```
DO [ LANGUAGE lang_name ] code;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **lang_name**

  用于解析代码的程序语言的名称。未指定时，默认的语言为plpgsql。

- **code**

  可以被执行的程序语言代码，必须指定为字符串。

示例

```
--创建用户webuser。
gaussdb=# CREATE USER webuser PASSWORD '********';

--授予用户webuser对模式tpcds下视图的所有操作权限。
gaussdb=# DO $$DECLARE r record;
BEGIN
    FOR r IN SELECT c.relname,n.nspname FROM pg_class c,pg_namespace n
            WHERE c.relnamespace = n.oid AND n.nspname = 'tpcds' AND relkind IN ('r','v')
    LOOP
        EXECUTE 'GRANT ALL ON ' || quote_ident(r.table_schema) || '.' || quote_ident(r.table_name) || ' TO
webuser';
    END LOOP;
END$$;


--删除用户webuser。
gaussdb=# DROP USER webuser CASCADE;
```

1.13.10.5 DROP AGGREGATE

功能描述

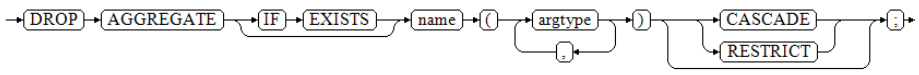
从数据库中删除一个聚集函数。

注意事项

仅聚集函数的所有者可以对该聚集函数进行DROP AGGREGATE操作。

语法格式

```
DROP AGGREGATE [ IF EXISTS ] name ( argtype [ , ... ] ) [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的聚集函数是否存在。

  - 如果指定的聚集函数存在，则删除成功。
  - 如果指定的聚集函数不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的聚集函数不存在。

- **name**

  需要删除的聚集函数，可以使用模式进行具体指定。

- **argtype**

版权所有 © 华为云计算技术有限公司

聚集函数操作的输入数据类型。当指定的聚集函数为零参数类型时，请用*代替输入数据类型列表。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该聚集函数的对象，默认为RESTRICT。

  - CASCADE：级联删除依赖于该聚集函数的对象。
  - RESTRICT：如果存在依赖于该聚集函数的对象，则拒绝删除该聚集函数。

示例

```
--创建自定义函数。
gaussdb=# CREATE OR REPLACE FUNCTION int_add(int,int)
 RETURNS int AS $BODY$
DECLARE
BEGIN
 RETURN $1 + $2;
END;
$BODY$ language plpgsql;

--创建聚集函数。
gaussdb=# CREATE AGGREGATE myavg (int)
(
    sfunc = int_add,
    stype = int,
    initcond = '0'
);

--将int类型的聚集函数myavg删除。
gaussdb=# DROP AGGREGATE myavg(int);

--删除自定义函数。
gaussdb=# DROP FUNCTION int_add(int,int);
```

相关链接

ALTER AGGREGATE，**CREATE AGGREGATE**

兼容性

SQL标准中没有DROP AGGREGATE的语句。

1.13.10.6 DROP AUDIT POLICY

功能描述

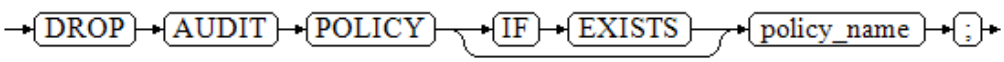
从数据库中删除一个审计策略。

注意事项

仅POLADMIN、SYSADMIN或初始用户可以执行该操作。

语法格式

```
DROP AUDIT POLICY [IF EXISTS] policy_name;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **policy_name**

  审计策略名称，需满足唯一约束，不可重复。

  取值范围：字符串，要符合**标识符命名规范**。

- **IF EXISTS**

  判断指定的审计策略是否存在。

  - 如果指定的审计策略存在，则删除成功。
  - 如果指定的审计策略不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的审计策略不存在。

示例

```
--创建adt1策略。
gaussdb=# CREATE AUDIT POLICY adt1 PRIVILEGES CREATE;
CREATE AUDIT POLICY

--删除审计策略adt1。
gaussdb=# DROP AUDIT POLICY adt1;
DROP AUDIT POLICY

--删除一个不存在的审计策略adt0,提示删除失败，该审计策略不存在。
gaussdb=# DROP AUDIT POLICY adt0;
ERROR:  adt0 policy does not exist, drop failed
```

相关链接

CREATE AUDIT POLICY，**ALTER AUDIT POLICY**

1.13.10.7 DROP CLIENT MASTER KEY

功能描述

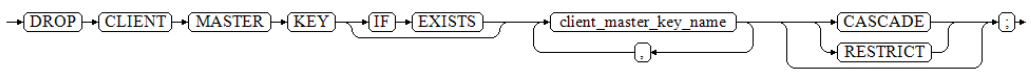
删除客户端加密主密钥（CMK）。

注意事项

- 仅客户端加密主密钥所有者或者被授予了DROP权限的用户可以执行该操作，系统管理员默认拥有此权限。
- 该命令只能删除数据库中系统表记录的元数据信息，不能真正删除CMK密钥文件，需要通过KeyTool工具才能删除CMK密钥文件。

语法格式

```
DROP CLIENT MASTER KEY [ IF EXISTS ] client_master_key_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的客户端加密主密钥是否存在。

版权所有 © 华为云计算技术有限公司

  - 如果指定的客户端加密主密钥存在，则删除成功。
  - 如果指定的客户端加密主密钥不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的客户端加密主密钥不存在。
- **client_master_key_name**

  要删除的客户端加密主密钥名称。

  取值范围：字符串，已存在的客户端加密主密钥名称。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于客户端加密主密钥的对象，默认为RESTRICT。
  - CASCADE：级联删除依赖于该客户端加密主密钥的对象。
  - RESTRICT：如果存在依赖于该客户端加密主密钥的对象，则拒绝删除该客户端加密主密钥。

相关链接

CREATE CLIENT MASTER KEY

1.13.10.8 DROP COLUMN ENCRYPTION KEY

功能描述

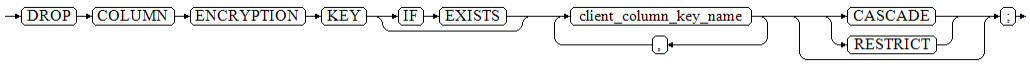
删除一个列加密密钥（CEK）。

注意事项

仅列加密密钥所有者或者被授予了DROP权限的用户可以执行该操作，系统管理员默认拥有此权限。

语法格式

```
DROP COLUMN ENCRYPTION KEY [ IF EXISTS ] client_column_key_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的列加密密钥是否存在。
  - 如果指定的列加密密钥存在，则删除成功。
  - 如果指定的列加密密钥不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的列加密密钥不存在。
- **client_column_key_name**

  需要删除的列加密密钥名称。

  取值范围：字符串，已存在的列加密密钥名称。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于列加密密钥的对象。

  对于密态特性来说，级联删除加密列属于危险操作，实际上，无法删除依赖于列加密密钥的加密列。

版权所有 © 华为云计算技术有限公司

相关链接

ALTER COLUMN ENCRYPTION KEY，CREATE COLUMN ENCRYPTION KEY

1.13.10.9 DROP DATABASE

功能描述

删除一个数据库。

注意事项

- 仅数据库所有者或者被授予了数据库DROP权限的用户可以执行该操作，系统管理员默认拥有此权限。
- 无法对系统默认安装的模板数据库和用户数据库（template0、template1、templatea、templatem和postgres）执行删除操作。如果需要查看当前服务中存在的数据库，可以使用gsql的\l命令进行查看。
- 当需要删除的数据库存在用户连接时，将无法成功删除数据库。可以通过视图DV_SESSIONS查看当前数据库的连接状态。
- 不能在事务块中执行该操作。
- 确定删除数据库前，需要执行“CLEAN CONNECTION TO ALL FORCE FOR DATABASE XXXX;”命令，用于强制停止当前已有的用户连接及后台线程，防止因为存在后台线程未完全退出而导致删除操作失败问题。

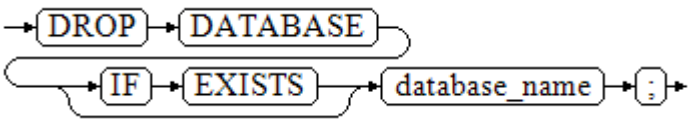
  须知

  DROP DATABASE一旦执行将无法撤销，请谨慎使用。强制停止后台线程可能导致当前数据库数据一致性问题，“CLEAN CONNECTION TO ALL FORCE FOR DATABASE XXXX;”仅在确定删除数据库阶段执行。

- 如果执行DROP DATABASE失败，事务回滚，需要再次执行一次DROP DATABASE IF EXISTS操作。

语法格式

```
DROP DATABASE [ IF EXISTS ] database_name;
```



参数说明

- **IF EXISTS**

  判断指定的数据库是否存在，使用该参数可以避免出现报错。
  - 如果指定的数据库存在，则成功删除。
  - 如果指定的数据库不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的数据库不存在。
- **database_name**

版权所有 © 华为云计算技术有限公司

指定删除的数据库名称。

取值范围：字符串，已存在的数据库名称。

示例

```
--创建数据库。
gaussdb=# CREATE DATABASE testdb1;

--删除数据库。
gaussdb=# DROP DATABASE testdb1;
```

相关链接

CREATE DATABASE

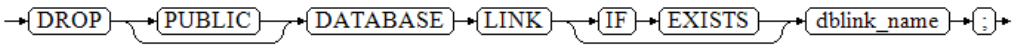
1.13.10.10 DROP DATABASE LINK

功能描述

删除一个DATABASE LINK对象。

语法格式

```
DROP [ PUBLIC ] DATABASE LINK [ IF EXISTS ] dblink_name;
```



参数说明

- **dblink_name**

  连接对象的名称。

- **IF EXISTS**

  判断指定的DATABASE LINK是否存在。

  - 如果指定的DATABASE LINK存在，则删除成功。
  - 如果指定的DATABASE LINK不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的DATABASE LINK不存在。

- **PUBLIC**

  连接类型，未指定PUBLIC时，默认为PRIVATE。

示例

```
-- 创建一个兼容性为ORA的数据库。
gaussdb=# CREATE DATABASE ora_test_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE

-- 切换数据库。
gaussdb=# \c ora_test_db

-- 创建拥有系统管理员权限的用户。
ora_test_db=# CREATE USER local_user WITH SYSADMIN PASSWORD '********';
CREATE ROLE
ora_test_db=# SET ROLE local_user PASSWORD '********';
SET

-- 创建私有DBLink，host也可以是IPv6地址。
```


版权所有 © 华为云计算技术有限公司

```
ora_test_db=> CREATE DATABASE LINK private_dblink CONNECT TO 'remote_user' IDENTIFIED BY '********'
USING (host '***.***.***.***',port '*****',dbname 'remote_db');
CREATE DATABASE LINK

-- 删除私有DBLink。
ora_test_db=> DROP DATABASE LINK private_dblink;
DROP DATABASE LINK

-- 创建公共DBLink，host也可以是IPv6地址。
ora_test_db=> CREATE PUBLIC DATABASE LINK public_dblink CONNECT TO 'remote_user' IDENTIFIED BY
'********' USING (host '***.***.***.***',port '*****',dbname 'remote_db');
CREATE DATABASE LINK

-- 删除公共DBLink。
ora_test_db=> DROP PUBLIC DATABASE LINK public_dblink;
DROP DATABASE LINK

-- 删除创建出的用户。
ora_test_db=> RESET ROLE;
RESET
ora_test_db=# DROP USER local_user;
DROP ROLE

-- 切换回初始数据库，并删除测试数据库。请用真实的数据库名替换postgres。
ora_test_db=# \c postgres
gaussdb=# DROP DATABASE ora_test_db;
DROP DATABASE
```

相关链接

ALTER DATABASE LINK，**CREATE DATABASE LINK**

1.13.10.11 DROP DIRECTORY

功能描述

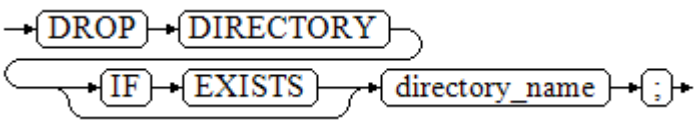
从数据库中删除一个目录对象。

注意事项

- 当enable_access_server_directory=off时，仅允许初始用户可以执行该操作。
- 当enable_access_server_directory=on时，具有SYSADMIN权限的用户、DIRECTORY对象的所有者、被授予了该目录对象的DROP权限的用户或者继承了内置角色gs_role_directory_drop权限的用户可以执行该操作。

语法格式

```
DROP DIRECTORY [ IF EXISTS ] directory_name;
```



参数说明

- **directory_name**
  目录名称。
  取值范围：已经存在的目录名。

版权所有 © 华为云计算技术有限公司

- **IF EXISTS**

  判断指定的目录是否存在。

  - 如果指定的目录存在，则删除成功。
  - 如果指定的目录不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的目录不存在。

示例

```
--创建目录对象。
gaussdb=# CREATE OR REPLACE DIRECTORY  dir  as '/tmp/';

--删除目录对象。
gaussdb=# DROP DIRECTORY dir;
```

相关链接

CREATE DIRECTORY，**ALTER DIRECTORY**

1.13.10.12 DROP EXTENSION

须知

扩展功能为内部使用功能，只允许删除数据库内部的高级包或扩展。当前仅支持postgis、postgis_sfcgal、postgis_raster、postgis_topology、yukon_geomodel、yukon_geogridcoder、pkg_bpchar_opc和tablefunc。

功能描述

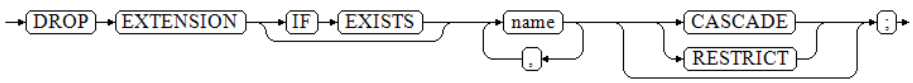
从数据库中删除一个扩展。

注意事项

- 仅扩展的所有者或初始用户才可以执行该操作。
- 在删除扩展的过程中，构成扩展的组件也会被级联删除。

语法格式

```
DROP EXTENSION [ IF EXISTS ] name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的扩展是否存在。

  - 如果指定的扩展存在，则删除成功。
  - 如果指定的扩展不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的扩展不存在。

- **name**

版权所有 © 华为云计算技术有限公司

已安装的扩展模块的名称。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该扩展的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖于该扩展的对象。
  - RESTRICT：如果存在依赖于扩展的对象，则不允许删除此扩展（除非所有的成员对象和其他扩展对象在一条DROP命令中同时删除）。

相关链接

CREATE EXTENSION

1.13.10.13 DROP FOREIGN DATA WRAPPER

功能描述

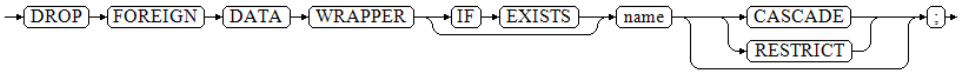
从数据库中删除一个外部数据封装器。

注意事项

- 只有在GUC参数support_extended_features设置为on时，才能成功执行该操作。
- 只有在内部初始化阶段或升级脚本中可以删除外部数据封装器，其他场景下用户无法直接使用该语法。

语法格式

```
DROP FOREIGN DATA WRAPPER [ IF EXISTS ] name [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的外部数据封装器是否存在。

  - 如果指定的外部数据封装器存在，则删除成功。
  - 如果指定的外部数据封装器不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的外部数据封装器不存在。

- **name**

  已存在的外部数据封装器的名称。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该外部数据封装器的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖外部数据封装器的对象（如服务器）。
  - RESTRICT：如果有依赖于外部数据封装器的对象，则不允许删除外部数据封装器。

示例

```
--创建一个外部数据封装dbi。
gaussdb=# CREATE FOREIGN DATA WRAPPER dbi OPTIONS (bar 'true');
```


版权所有 © 华为云计算技术有限公司

```
--删除外部数据封装器dbi。（前置条件：需确认GUC参数support_extended_features=on）
gaussdb=# DROP FOREIGN DATA WRAPPER dbi;
```

相关链接

CREATE FOREIGN DATA WRAPPER，ALTER FOREIGN DATA WRAPPER

1.13.10.14 DROP FOREIGN TABLE

功能描述

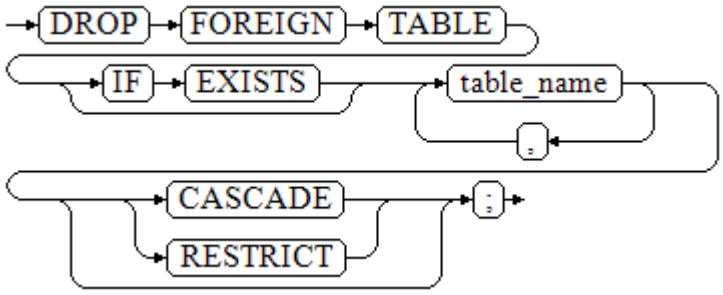
从数据库中删除一个外表。

注意事项

该操作会强制删除指定的表，删除表后，依赖该表的索引会被级联删除，使用到该表的函数和存储过程将无法成功执行。

语法格式

```
DROP FOREIGN TABLE [ IF EXISTS ] table_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的外表是否存在。
  - 如果指定的外表存在，则删除成功。
  - 如果指定的外表不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的外表不存在。
- **table_name**

  需要删除的表的名称。

  取值范围：已存在的表名。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该外表的对象。未指定时，默认为RESTRICT。
  - CASCADE：级联删除依赖于表的对象（如视图）。
  - RESTRICT：如果存在依赖对象，则拒绝删除该表。

示例

请参见CREATE FOREIGN TABLE章节的示例。

版权所有 © 华为云计算技术有限公司

相关链接

CREATE FOREIGN TABLE

1.13.10.15 DROP FUNCTION

功能描述

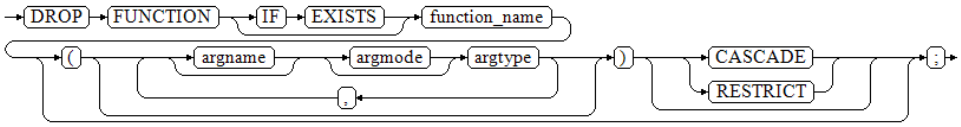
从数据库中删除一个已存在的函数。

注意事项

- 如果函数中涉及对临时表的相关操作，则无法使用该操作进行删除。
- 仅函数的所有者或者被授予了函数DROP权限的用户可以执行该操作，系统管理员默认拥有该权限。
- 仅初始用户可以对初始用户的函数进行删除操作。

语法格式

```
DROP FUNCTION [ IF EXISTS ] function_name
[ ( [ { [ argname ] [ argmode ] argtype} [, ...] ] ) [ CASCADE | RESTRICT ] ];
```



参数说明

- **IF EXISTS**

  判断指定的函数是否存在。
  - 如果指定的函数存在，则删除成功。
  - 如果指定的函数不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的函数不存在。
- **function_name**

  要删除的函数名称。

  取值范围：已存在的函数名。
- **argmode**

  函数参数的模式。
- **argname**

  函数参数的名称。
- **argtype**

  函数参数的类型。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该函数的对象。未指定时，默认为RESTRICT。
  - CASCADE：级联删除依赖于函数的对象。
  - RESTRICT：如果存在依赖于函数的对象，则拒绝删除该函数。

版权所有 © 华为云计算技术有限公司

示例

- 删除函数可省略参数列表

```
--创建函数。
gaussdb=# CREATE FUNCTION func_test(varchar) RETURN VARCHAR AS
BEGIN
    RETURN $1||_'test';
END;
/

--删除函数。
gaussdb=# DROP FUNCTION func_test;
```

- 删除存在同名的函数

  如果存在同名函数，删除时需加上参数列表，否则报错。

```
--创建函数。
gaussdb=# CREATE FUNCTION func_add(int) RETURNS int AS $$
BEGIN
    RETURN $1+10;
END;
$$ LANGUAGE PLPGSQL;

--重载函数func_add。
gaussdb=# CREATE FUNCTION func_add(int,int) RETURNS int AS $$
BEGIN
    RETURN $1+$2;
END;
$$ LANGUAGE PLPGSQL;

--删除函数。
gaussdb=# DROP FUNCTION func_add(int);
gaussdb=# DROP FUNCTION func_add(int,int);
```

相关链接

ALTER FUNCTION，**CREATE FUNCTION**

1.13.10.16 DROP GLOBAL CONFIGURATION

功能描述

删除系统表gs_global_config中的参数值。

注意事项

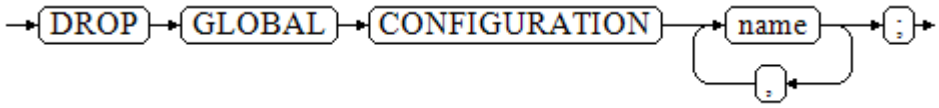
- 仅支持数据库初始用户运行此命令。
- 参数名称不能为weak_password和undostoragetype。
- 参数名称为扩容重分布预留参数时，需设置GUC参数enable_cluster_resize为on。涉及参数名称：lockwait_timeout、lockwait_interval、trylock_threshold、enable_cancel、last_catchup_threshold、catchup_times、write_error_mode、catchup_query_dop、parallel_catchup_threshold、parallel_reindex_jobs、redis_retry_times、redis_retry_interval、redis_bucket_expansion、redis_bucket_batchsize、redis_bucket_log_level、redis_bucket_lockhold_timeout、redis_bucket_timeout_penalty、redis_bucket_max_lockwait_timeout_retry、redis_bucket_max_lockhold_timeout_retry、redis_bucket_strategy_history_capacity、redis_bucket_strategy_recent_threshold、redis_bucket_workload_rule、redis_bucket_lsn_rule、redis_bucket_tx_bucket_min、

版权所有 © 华为云计算技术有限公司

redis_bucket_tx_bucket_max、redis_max_expected_tps、redis_bucket_sender_concurrency、redis_bucket_receiver_concurrency、redis_bucket_parse_redo_num、redis_bucket_workers_per_paser、redis_bucket_keep_bucketxlogs、redis_bucket_reindex_cbi_offline以及redis_bucket_reindex_parallel_workers。

语法格式

```
DROP GLOBAL CONFIGURATION name [, ...];
```



参数说明

name

参数名称必须是gs_global_config中已经存在的参数，删除不存在的参数将报错。参数名称为扩容重分布预留参数时，需设置GUC参数enable_cluster_resize为on。

示例

请参见**ALTER GLOBAL CONFIGURATION**章节的**示例**。

相关链接

ALTER GLOBAL CONFIGURATION

1.13.10.17 DROP GROUP

功能描述

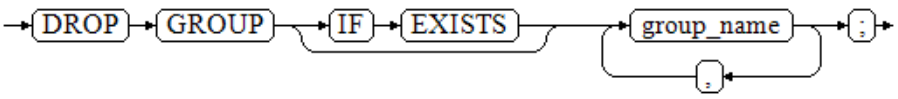
删除一个用户组。DROP GROUP与DROP ROLE为同义词。

注意事项

仅对有CREATE ROLE权限的用户开放，CREATE ROLE权限通过管理员用户赋予。

语法格式

```
DROP GROUP [ IF EXISTS ] group_name [, ...];
```



参数说明

- **IF EXISTS**

  判断指定的用户组是否存在。

  - 如果指定的用户组存在，则删除成功。

版权所有 © 华为云计算技术有限公司

  - 如果指定的用户组不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的用户组不存在。

- **group_name**

  要删除的用户组名称。

  取值范围：已存在的用户组/角色。

示例

请参见**CREATE GROUP**章节的**示例**。

相关链接

CREATE GROUP，**ALTER GROUP**，**DROP ROLE**

1.13.10.18 DROP INDEX

功能描述

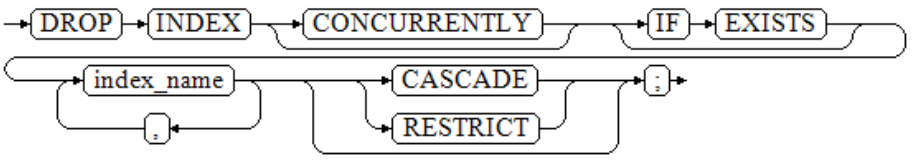
删除一个索引。

注意事项

索引的所有者、索引所在模式的所有者、拥有索引所在表的INDEX权限的用户或者被授予了DROP ANY INDEX权限的用户可以执行该操作，三权分立关闭时的系统管理员默认拥有此权限。

语法格式

```
DROP INDEX [ CONCURRENTLY ] [ IF EXISTS ]
    index_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **CONCURRENTLY**

  以不阻塞DML的方式在线删除索引（加四级锁）。删除索引时，一般会阻塞其他语句对该索引所依赖表的访问。加此关键字，可在删除过程中避免阻塞。

版权所有 © 华为云计算技术有限公司

说明

- 删除索引指定CONCURRENTLY选项时，只能指定一个索引的名称， 并且不支持CASCADE选项。
- 普通DROP INDEX命令可以在事务内执行，但是DROP INDEX CONCURRENTLY不可以在事务内执行。
- 对于临时表，支持使用CONCURRENTLY关键字删除索引，但是实际删除过程中，采用的是阻塞式的删除方式，因为没有其他会话会并发访问临时表，并且阻塞式删除成本更低。
- 与在线创建索引类似，在线删除索引也存在阶段需要等待索引对应表上相关的其他事务结束，因此可能出现锁超时问题，导致在线删除索引失败。
- 与在线创建索引类似，由于需要等待表上相关的其他事务结束，在线删除索引可能会和其他DDL出现死锁问题，导致在线删除索引失败。具体原因是：在线删除索引持有四级锁，等待并发DDL事务结束；并发DDL持有事务锁，等待在线删除索引释放四级锁。
- 当发生锁超时、死锁、数据库异常等故障或者用户取消操作时，可能造成在线删除索引失败，残留非法状态的索引。残留索引在CN、DN的状态可能不一致（比如CN为not valid状态，DN为valid状态；或者部分CN为valid状态，部分CN为not valid状态，DN为valid状态）或者残留索引在部分节点存在而其他节点不存在，但不会造成不一致问题。为避免占用系统资源，需尽快将其清理，可以继续使用DROP INDEX CONCURRENTLY或者DROP INDEX将非法索引删除，如果使用DROP INDEX CONCURRENTLY删除索引持续失败，可以使用DROP INDEX删除索引。
- 在线删除索引时间与并发业务量、并发业务运行时间有关。无并发业务时，在线删除索引为秒级别；有并发业务时，需要额外等待某时刻正在运行的所有并发业务结束，因此在线删除索引时间要额外加上这些并发业务运行时间。
- 分布式在线删除索引只支持btree和ubtree索引，不支持GSI。
- 在执行DROP INDEX CONCURRENTLY期间，涉及该索引的Stream计划可能引发报错。建议在执行该操作前，通过SQL Patch或为相关SQL添加Hint的方式，禁用Stream计划，使其执行pgxc计划。
- 生成WDR快照期间如果有在线删除索引操作，可能会出现Could not open relation with OID %u.报错导致WDR快照生成失败，此现象为预期内行为，待索引删除完成后重新生成WDR快照即可。

- **IF EXISTS**

  判断指定的索引是否存在。

  - 如果指定的索引存在，则删除成功。
  - 如果指定的索引不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定索引不存在。

  在线创建、删除索引失败后，可能会导致部分节点中无法匹配到失败索引的元信息，DROP INDEX语句无法删除该索引，需要用DROP INDEX IF EXISTS进行删除。

- **index_name**

  要删除的索引名。

  取值范围：已存在的索引。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该索引的对象。未指定时，默认为RESTRICT。

  - CASCADE：允许级联删除依赖于该索引的对象。
  - RESTRICT：存在依赖于此索引的对象时，该索引无法被删除。

示例

```
--创建表。
gaussdb=# CREATE TABLE test1_index (id INT, name VARCHAR(20));
```


版权所有 © 华为云计算技术有限公司

```
--创建索引。
gaussdb=# CREATE INDEX idx_test1 ON test1_index(id);

--删除索引。
gaussdb=# DROP INDEX IF EXISTS idx_test1 CASCADE;

--在线创建索引。
gaussdb=# CREATE INDEX CONCURRENTLY idx_test2 ON test1_index(name);

--在线删除索引。
gaussdb=# DROP INDEX CONCURRENTLY idx_test2;

--删除表。
gaussdb=# DROP TABLE test1_index;
```

相关链接

ALTER INDEX，**CREATE INDEX**

1.13.10.19 DROP LANGUAGE

功能描述

删除一个过程语言。

注意事项

当前形态暂不支持使用该语法。

1.13.10.20 DROP MASKING POLICY

功能描述

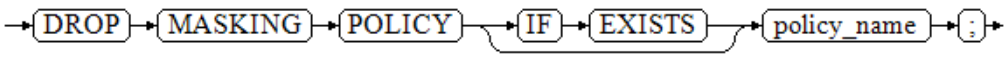
删除一种脱敏策略。

注意事项

仅POLADMIN，SYSADMIN或初始用户可以执行该操作。

语法格式

```
DROP MASKING POLICY [ IF EXISTS ] policy_name;
```



参数说明

policy_name

脱敏策略名称，不可重复。

取值范围：字符串，已存在的策略名称。

示例

```
--创建一个表tb_for_masking。
gaussdb=# CREATE TABLE tb_for_masking(idx int, col1 text, col2 text, col3 text, col4 text, col5 text, col6
```


版权所有 © 华为云计算技术有限公司

```
text, col7 text,col8 text);

--给表tb_for_masking插入数据。
gaussdb=# INSERT INTO tb_for_masking VALUES(1, '9876543210', 'usr321usr', 'abc@huawei.com',
'abc@huawei.com', '1234-4567-7890-0123', 'abcdef 123456 ui 323 jsfd321 j3k2l3', '4880-9898-4545-2525',
'this is a llt case');

--创建资源标签标记敏感列col1。
gaussdb=# CREATE RESOURCE LABEL mask_lb1 ADD COLUMN(tb_for_masking.col1);

--创建资源标签标记敏感列col2。
gaussdb=# CREATE RESOURCE LABEL mask_lb2 ADD COLUMN(tb_for_masking.col2);

--创建资源标签标记敏感列col3。
gaussdb=# CREATE RESOURCE LABEL mask_lb3 ADD COLUMN(tb_for_masking.col3);

--创建一个名为maskpol1的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol1 maskall ON LABEL(mask_lb1);

--创建一个名为maskpol2的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol2 randommasking ON LABEL(mask_lb2);

--创建一个名为maskpol3的脱敏策略。
gaussdb=# CREATE MASKING POLICY maskpol3 basicemailmasking ON LABEL(mask_lb3);

--删除一个脱敏策略。
gaussdb=# DROP MASKING POLICY IF EXISTS maskpol1;

--删除一组脱敏策略。
gaussdb=# DROP MASKING POLICY IF EXISTS maskpol2, maskpol3;

--删除资源标签。
gaussdb=# DROP RESOURCE LABEL mask_lb1;
gaussdb=# DROP RESOURCE LABEL mask_lb2;
gaussdb=# DROP RESOURCE LABEL mask_lb3;

--删除表。
gaussdb=# DROP TABLE tb_for_masking;
```

相关链接

ALTER MASKING POLICY，**CREATE MASKING POLICY**

1.13.10.21 DROP MATERIALIZED VIEW

功能描述

删除数据库中已有的物化视图。

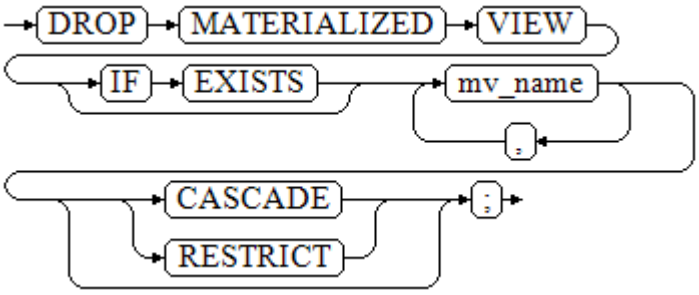
注意事项

物化视图的所有者、物化视图所在模式的所有者、被授予了物化视图DROP权限的用户或拥有DROP ANY TABLE权限的用户可以执行该操作，系统管理员默认拥有此权限。

语法格式

```
DROP MATERIALIZED VIEW [ IF EXISTS ] mv_name [, ...] [ CASCADE | RESTRICT ];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **IF EXISTS**

  判断指定的物化视图是否存在。

  - 如果指定的物化视图存在，则删除成功。
  - 如果指定的物化视图不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的物化视图不存在。

- **mv_name**

  要删除的物化视图名称。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该物化视图的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖此物化视图的对象。
  - RESTRICT：如果存在依赖此物化视图的对象，则拒绝删除此物化视图。

示例

```
--创建表。
gaussdb=# CREATE TABLE my_table (c1 int, c2 int)
WITH(STORAGE_TYPE=ASTORE);

--创建名为my_mv的物化视图。
gaussdb=# CREATE MATERIALIZED VIEW my_mv AS SELECT * FROM my_table;

--删除名为my_mv的物化视图。
gaussdb=# DROP MATERIALIZED VIEW my_mv;

--删除表。
gaussdb=# DROP TABLE my_table;
```

相关链接

ALTER MATERIALIZED VIEW，**CREATE INCREMENTAL MATERIALIZED VIEW**，**CREATE MATERIALIZED VIEW**，**CREATE TABLE**，**REFRESH INCREMENTAL MATERIALIZED VIEW**，**REFRESH MATERIALIZED VIEW**

1.13.10.22 DROP MODEL

功能描述

删除一个已训练完成保存的模型对象。

版权所有 © 华为云计算技术有限公司

注意事项

当前形态暂不支持使用该语法。

1.13.10.23 DROP NODE

功能描述

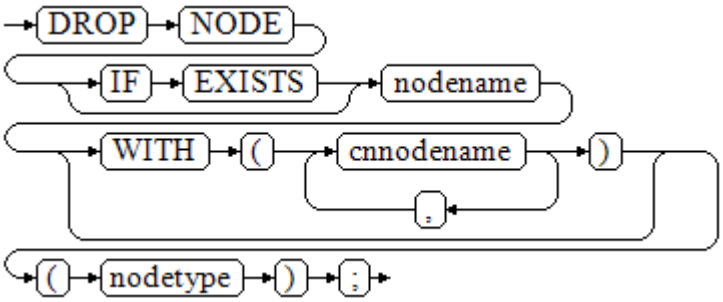
删除集群节点。

注意事项

DROP NODE是集群管理工具封装的接口，用来实现集群管理，管理员用户才有权限使用该接口。该接口禁止用户直接使用，以免对集群状态造成影响，导致数据库不可用。

语法格式

```
DROP NODE [ IF EXISTS ] nodename [WITH ( cnnodename [, ... ] )] (nodetype);
```



参数说明

- **IF EXISTS**

  如果指定的节点不存在，则发出一个notice而不是抛出一个错误。

- **nodename**

  要删除的节点名。

  取值范围：已存在的节点nodename。

- **cnnodename**

  CN名称。如果定义了该参数，则删除当前连接CN下的集群节点以及所定义CN下的集群节点。

  取值范围：已存在的CN的nodename。

- **nodetype**

  待修改节点的类型，pgxc_node系统表中node_type的值。

  取值范围："C"（协调节点）。

说明

多CN环境下，创建集群节点时只会在当前连接的CN下创建，不会同步至其他CN，需要在所有CN上执行创建操作，如果删除集群节点只需要链接一个CN即可。如果只在其中一个CN执行创建了集群节点，要删除需要定义当前cnnodename，并在此CN下执行删除操作。

版权所有 © 华为云计算技术有限公司

示例

```
--创建集群节点，在所有CN上执行。
gaussdb=# CREATE NODE datanode1 WITH(
   TYPE = datanode,
   PREFERRED = false
);

--删除集群节点，连接一个CN即可。
gaussdb=# DROP NODE datanode1;
```

相关链接

CREATE NODE，**ALTER NODE**

1.13.10.24 DROP NODE GROUP

功能描述

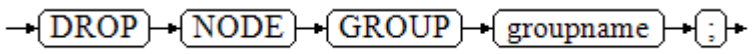
删除节点组。

注意事项

- DROP NODE GROUP是集群管理工具封装的接口，用于实现集群管理。
- 仅系统管理员或者被授予了节点组DROP权限的用户可以执行该操作。

语法格式

```
DROP NODE GROUP groupname;
```



参数说明

groupname

需要删除的节点组名。

取值范围：已存在的节点组。

示例

```
--查询集群DN初始状态。
gaussdb=# SELECT node_name, nodeis_preferred FROM pgxc_node WHERE node_type = 'D' ORDER BY 1;
 node_name | nodeis_preferred
-----------+------------------
 dn_6001_6002_6003 | f
 dn_6004_6005_6006 | f
 dn_6007_6008_6009 | f
(3 rows)

--创建NODE GROUP，用上一步中查询到的真实节点名称替换dn_6001_6002_6003。
gaussdb=# CREATE NODE GROUP test_group WITH ( dn_6001_6002_6003 );

--删除NODE GROUP。
gaussdb=# DROP NODE GROUP test_group;
```


版权所有 © 华为云计算技术有限公司

相关链接

ALTER NODE GROUP，CREATE NODE GROUP

1.13.10.25 DROP OWNED

功能描述

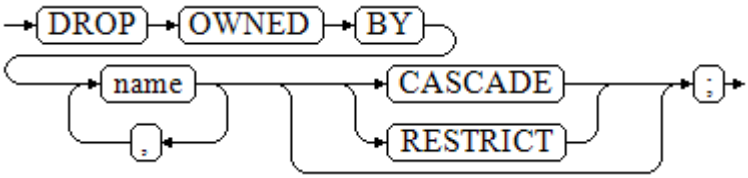
删除一个数据库角色所拥有的数据库对象。

注意事项

- 所有该角色在当前数据库里和共享对象（数据库，表空间）上的所有对象上的权限都将被撤销。
- DROP OWNED常常视为移除一个或者多个角色做准备，DROP OWNED只影响当前数据库中的对象，通常需要在包含将被移除角色所拥有的对象所在的每一个数据库中都执行该命令。
- 使用CASCADE选项可能导致该操作递归删除其他用户所拥有的对象，请谨慎使用。
- 角色所拥有的数据库及表空间将不会被移除。
- 角色所拥有的私有DATABASE LINK连接，需要添加CASCADE才可以删除。

语法格式

```
DROP OWNED BY name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **name**

  指定的角色名。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该角色的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖于被删除角色对象的对象。
  - RESTRICT：拒绝删除存在依赖对象的角色对象。

示例

```
--创建jim用户。
gaussdb=# CREATE USER jim PASSWORD '********';

--撤销jim在当前数据库里和共享对象（数据库，表空间）上的所有对象上的权限。
gaussdb=# DROP OWNED BY jim;

--删除jim用户。
gaussdb=# DROP USER jim;
```


版权所有 © 华为云计算技术有限公司

相关链接

REASSIGN OWNED

1.13.10.26 DROP PACKAGE

功能描述

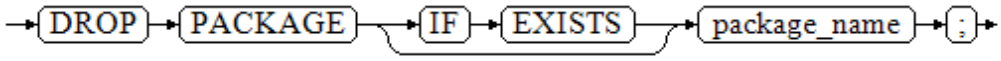
删除已存在的PACKAGE或PACKAGE BODY。

注意事项

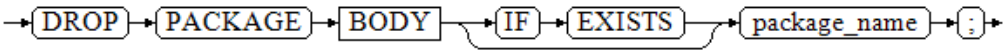
- 删除PACKAGE BODY后，PACKAGE内的存储过程及函数会同时失效。
- 仅初始用户有权对初始用户的PACKAGE进行DROP操作。

语法格式

```
DROP PACKAGE [ IF EXISTS ] package_name;
```




```
DROP PACKAGE BODY [ IF EXISTS ] package_name;
```



参数说明

- **IF EXISTS**

  判断指定的PACKAGE或PACKAGE BODY是否存在。
  - 如果指定的PACKAGE或PACKAGE BODY存在，则删除成功。
  - 如果指定的PACKAGE或PACKAGE BODY不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的PACKAGE或PACKAGE BODY不存在。
- **package_name**

  要删除的PACKAGE名称。

  取值范围：已存在的PACKAGE名。

示例

```
--创建ORA兼容性数据库。
gaussdb=# CREATE DATABASE ora_compat_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE
gaussdb=# \c ora_compat_db

--创建PACKAGE。
ora_compat_db=# CREATE OR REPLACE PACKAGE emp_bonus IS
var1 INT:=1;--公有变量
var2 INT:=2;
PROCEDURE testpro1(var3 INT);--公有存储过程，可以被外部调用
END emp_bonus;
/

--删除PACKAGE。
ora_compat_db=# DROP PACKAGE emp_bonus;
```


版权所有 © 华为云计算技术有限公司

```
--切换回初始数据库，并删除ORA兼容性数据库。请用真实的数据库名替换postgres。
ora_compat_db=# \c postgres
gaussdb=# DROP DATABASE ora_compat_db;
```

相关链接

ALTER PACKAGE，CREATE PACKAGE

1.13.10.27 DROP PROCEDURE

功能描述

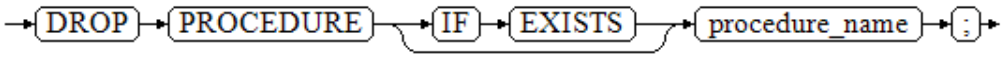
删除已存在的存储过程。

注意事项

仅初始用户有权对初始用户的存储过程进行DROP操作。

语法格式

```
DROP PROCEDURE [ IF EXISTS ] procedure_name;
```



参数说明

- **IF EXISTS**

  判断指定的存储过程是否存在。

  - 如果指定的存储过程存在，则删除成功。
  - 如果指定的存储过程不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的存储过程不存在。

- **procedure_name**

  要删除的存储过程名称。

  取值范围：已存在的存储过程名。

示例

请参见CREATE PROCEDURE章节的示例。

相关链接

ALTER PROCEDURE，CREATE PROCEDURE

1.13.10.28 DROP RESOURCE LABEL

功能描述

删除一个资源标签。

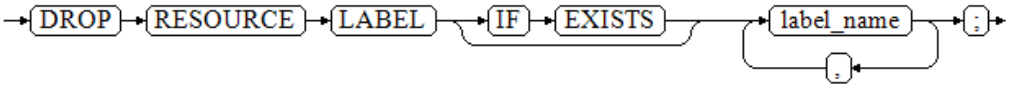
版权所有 © 华为云计算技术有限公司

注意事项

仅POLADMIN、SYSADMIN或初始用户可以执行该操作。

语法格式

```
DROP RESOURCE LABEL [IF EXISTS] label_name [, ...];
```



参数说明

label_name

需要删除资源标签的名称。

取值范围：字符串，要符合**标识符命名规范**。

示例

```
--创建一个表tb_for_label。
gaussdb=# CREATE TABLE tb_for_label(col1 text, col2 text, col3 text);

--基于表创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS table_label add TABLE(public.tb_for_label);

--再次创建已存在的表资源标签，对比加参数IF NOT EXISTS与不加IF NOT EXISTS参数的区别。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS table_label add TABLE(public.tb_for_label);
NOTICE:  table_label label already defined, skipping
CREATE RESOURCE LABEL
gaussdb=# CREATE RESOURCE LABEL table_label add TABLE(public.tb_for_label);
ERROR:  table_label label already defined

--基于列创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS column_label add COLUMN(public.tb_for_label.col1);

--创建一个模式schema_for_label。
gaussdb=# CREATE SCHEMA schema_for_label;

--基于模式创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS schema_label add SCHEMA(schema_for_label);

--创建一个视图view_for_label。
gaussdb=# CREATE VIEW view_for_label AS SELECT 1;

--基于视图创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS view_label add VIEW(view_for_label);

--创建一个函数func_for_label。
gaussdb=# CREATE FUNCTION func_for_label RETURNS TEXT AS $$ SELECT col1 FROM tb_for_label; $$
LANGUAGE SQL;

--基于函数创建资源标签。
gaussdb=# CREATE RESOURCE LABEL IF NOT EXISTS func_label add FUNCTION(func_for_label);

--删除表资源标签table_label。
gaussdb=# DROP RESOURCE LABEL IF EXISTS table_label;

--删除列资源资源标签column_label。
gaussdb=# DROP RESOURCE LABEL IF EXISTS column_label;

--删除函数资源标签func_for_label。
gaussdb=# DROP FUNCTION func_for_label;
```


版权所有 © 华为云计算技术有限公司

```
--删除视图资源标签view_for_label。
gaussdb=# DROP VIEW view_for_label;

--删除模式资源标签schema_for_label。
gaussdb=# DROP SCHEMA schema_for_label;

--删除表tb_for_label。
gaussdb=# DROP TABLE tb_for_label;
```

相关链接

ALTER RESOURCE LABEL，**CREATE RESOURCE LABEL**

1.13.10.29 DROP RESOURCE POOL

功能描述

删除一个资源池。

说明

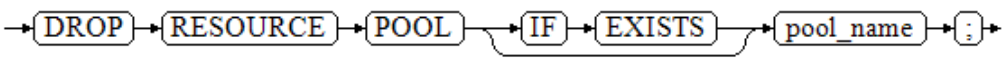
如果某个角色已关联到该资源池，无法删除。

注意事项

- 仅SYSADMIN或初始用户可以执行该操作。
- 升级期间不支持该语法。

语法格式

```
DROP RESOURCE POOL [ IF EXISTS ] pool_name;
```



参数说明

- **IF EXISTS**

  判断指定的资源池是否存在。
  - 如果指定的资源池存在，则删除成功。
  - 如果指定的资源池不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的资源池不存在。

- **pool_name**

  已创建过的资源池名称。

  取值范围：字符串，要符合**标识符命名规范**。

说明

多租户场景下，如果删除组资源池，其业务资源池都将被删除。只有不关联用户时，资源池才能被删除。

示例

请参见**CREATE RESOURCE POOL**章节的**示例**。

版权所有 © 华为云计算技术有限公司

相关链接

ALTER RESOURCE POOL，CREATE RESOURCE POOL

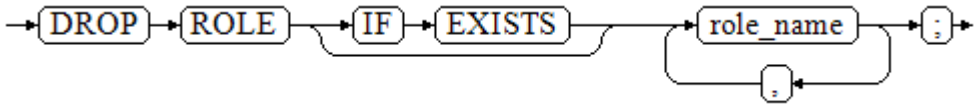
1.13.10.30 DROP ROLE

功能描述

删除一个角色。

语法格式

```
DROP ROLE [ IF EXISTS ] role_name [, ...];
```



参数说明

- **IF EXISTS**

  判断指定的角色是否存在。

  - 如果指定的角色存在，则删除成功。
  - 如果指定的角色不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的角色不存在。

- **role_name**

  需要删除的角色名称。

  取值范围：已存在的角色。

示例

```
--创建角色role11。
gaussdb=# CREATE ROLE role11  PASSWORD '********';
CREATE ROLE

--删除role11角色。
gaussdb=# DROP ROLE IF EXISTS role11;
DROP ROLE

--删除不存在的角色role12。
gaussdb=# DROP ROLE IF EXISTS role12;
NOTICE:  role "role12" does not exist, skipping
DROP ROLE
```

相关链接

CREATE ROLE，ALTER ROLE，SET ROLE

1.13.10.31 DROP ROW LEVEL SECURITY POLICY

功能描述

删除表上某个行访问控制策略。

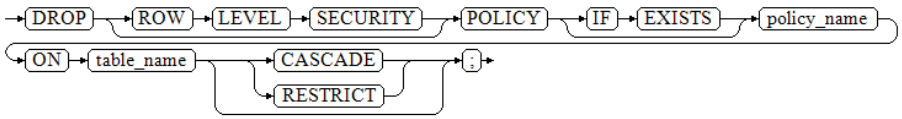
版权所有 © 华为云计算技术有限公司

注意事项

仅表的所有者或者管理员用户才能删除表的行访问控制策略。

语法格式

```
DROP [ ROW LEVEL SECURITY ] POLICY [ IF EXISTS ] policy_name ON table_name [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的行访问控制策略是否存在。

  - 如果指定的行访问控制策略存在，则删除成功。
  - 如果指定的行访问控制策略不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的行访问控制策略不存在。

- **policy_name**

  要删除的行访问控制策略的名称。

- **table_name**

  行访问控制策略所在的数据表名。

- **CASCADE | RESTRICT**

  仅适配此语法，无对象依赖于该行访问控制策略，CASCADE和RESTRICT效果相同。

示例

```
--创建数据表all_data。
gaussdb=# CREATE TABLE all_data(id int, role varchar(100), data varchar(100));

--创建行访问控制策略。
gaussdb=# CREATE ROW LEVEL SECURITY POLICY all_data_rls ON all_data USING(role = CURRENT_USER);

--删除行访问控制策略。
gaussdb=# DROP ROW LEVEL SECURITY POLICY all_data_rls ON all_data;

--删除数据表all_data。
gaussdb=# DROP TABLE all_data;
```

相关链接

ALTER ROW LEVEL SECURITY POLICY，**CREATE ROW LEVEL SECURITY POLICY**

1.13.10.32 DROP SCHEMA

功能描述

从数据库中删除指定模式。

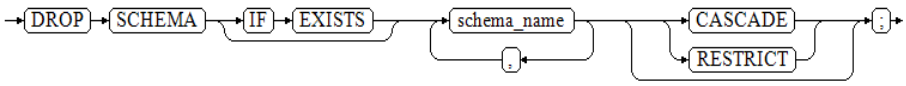
版权所有 © 华为云计算技术有限公司

注意事项

- 仅模式的所有者或者被授予了模式DROP权限的用户可以执行DROP SCHEMA命令，当三权分立开关关闭时，系统管理员默认拥有此权限。
- allow_system_table_mods关闭时，禁止删除DBE_PLDEVELOPER。
- 仅初始用户可以对系统模式进行DROP操作，其中Public Schema不支持删除。

语法格式

```
DROP SCHEMA [ IF EXISTS ] schema_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的模式是否存在，使用该参数可以避免出现报错。
  - 如果指定的模式存在，直接进行DROP操作。
  - 如果指定的模式不存在，则产生NOTICE，不会产生错误。
- **schema_name**

  模式的名称。

  取值范围：已存在模式名。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该模式的对象。未指定时，默认为RESTRICT。
  - CASCADE：自动删除包含在模式中的对象。
  - RESTRICT：如果模式包含任何对象，则删除失败。

须知

- 请勿随意对pg_temp或pg_toast_temp开头的模式进行DROP操作，该模式为系统内部使用，删除后可能导致无法预知的结果。
- 无法直接对当前模式进行DROP操作，如果需要删除当前模式，请切换至其他模式后再指定删除当前模式。

示例

```
--创建角色role2。
gaussdb=# CREATE ROLE role2 IDENTIFIED BY '********';

--根据名称创建schema。
gaussdb=# CREATE SCHEMA schema1 AUTHORIZATION role2;
CREATE SCHEMA

--删除schema。
gaussdb=# DROP SCHEMA schema1 CASCADE;

--删除用户。
gaussdb=# DROP USER role2 CASCADE;
```


版权所有 © 华为云计算技术有限公司

相关链接

ALTER SCHEMA，CREATE SCHEMA

1.13.10.33 DROP SECURITY LABEL

功能描述

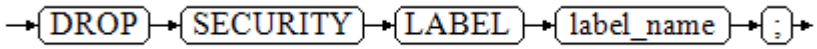
删除当前数据库中的安全标签。

注意事项

初始用户、具有SYSADMIN权限的用户或者继承了内置角色gs_role_seclabel权限的用户有权限删除安全标签。

语法格式

```
DROP SECURITY LABEL label_name;
```



参数说明

label_name

安全标签名称，需要删除的安全标签必须在数据库中存在。

示例

```
--创建安全标签sec_label。
gaussdb=# CREATE SECURITY LABEL sec_label 'L1:G4';

--删除不存在的安全标签sec_label2。
gaussdb=# DROP SECURITY LABEL sec_label2;
ERROR:  security label "sec_label2" does not exist

--删除已存在的安全标签sec_label。
gaussdb=# DROP SECURITY LABEL sec_label;
```

相关链接

CREATE SECURITY LABEL，SECURITY LABEL ON

1.13.10.34 DROP SEQUENCE

功能描述

从当前数据库里删除序列。

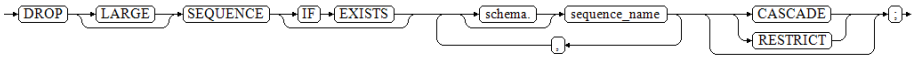
注意事项

序列的所有者、序列所在模式的所有者、被授予了序列DROP权限的用户或者被授予了DROP ANY SEQUENCE权限的用户可以执行该操作，当三权分立开关关闭时，系统管理员默认拥有该权限。

版权所有 © 华为云计算技术有限公司

语法格式

```
DROP [LARGE] SEQUENCE [ IF EXISTS ] { [schema.] sequence_name } [ , ... ] [ CASCADE | RESTRICT ];
```



参数说明

- **LARGE**

  LARGE标识。如果创建序列时使用了LARGE标识，进行删除时也需要指定LARGE标识。

- **IF EXISTS**

  判断指定的序列是否存在。

  - 如果指定的序列存在，则删除成功。
  - 如果指定的序列不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的序列不存在。

- **sequence_name**

  序列名称。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该序列的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖序列的对象。
  - RESTRICT：如果存在任何依赖的对象，则拒绝删除序列。

示例

```
--创建一个名为serial的递增序列，从101开始。
gaussdb=# CREATE SEQUENCE serial START 101;

--删除序列。
gaussdb=# DROP SEQUENCE serial;
```

相关链接

ALTER SEQUENCE，**DROP SEQUENCE**

1.13.10.35 DROP SERVER

功能描述

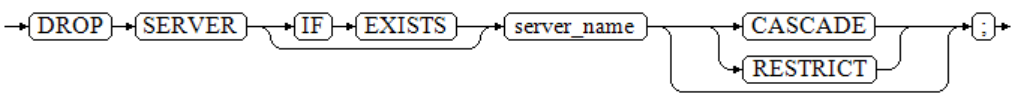
删除一个已有的数据服务器。

注意事项

只有SERVER的所有者或者被授予了SERVER的DROP权限的用户可以执行该操作，系统管理员默认拥有该权限。

语法格式

```
DROP SERVER [ IF EXISTS ] server_name [ CASCADE | RESTRICT ];
```



版权所有 © 华为云计算技术有限公司

参数描述

- **IF EXISTS**

  判断指定的数据服务器是否存在。

  - 如果指定的数据服务器存在，则删除成功。
  - 如果指定的数据服务器不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的数据服务器不存在。

- **server_name**

  服务器名称。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该数据服务器的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖于该数据服务器的对象。
  - RESTRICT：如果存在依赖对象，则拒绝删除该数据服务器。

示例

```
--创建SERVER。
gaussdb=# CREATE SERVER my_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER

--删除SERVER。
gaussdb=# DROP SERVER my_server;
DROP SERVER
```

相关链接

ALTER SERVER，**CREATE SERVER**

1.13.10.36 DROP SYNONYM

功能描述

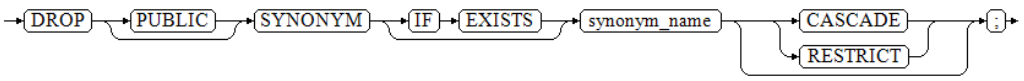
删除一个同义词对象。

注意事项

仅同义词的所有者或者被授予了DROP ANY SYNONYM权限的用户可以执行该操作，系统管理员默认拥有此权限。

语法格式

```
DROP [PUBLIC] SYNONYM [ IF EXISTS ] synonym_name [ CASCADE | RESTRICT ];
```



参数描述

- **PUBLIC**

  可选。删除PUBLIC同义词时使用。

- **IF EXISTS**

  判断指定的同义词是否存在。

版权所有 © 华为云计算技术有限公司

  - 如果指定的同义词存在，则删除成功。
  - 如果指定的同义词不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的同义词不存在。
- **synonym_name**

  同义词名字，可以使用模式进行具体指定。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该同义词的对象。未指定时，默认为RESTRICT。
  - CASCADE：级联删除依赖同义词的对象（如视图）。
  - RESTRICT：如果存在依赖对象，则拒绝删除同义词。

示例

```
--创建同义词。
gaussdb=# CREATE OR REPLACE SYNONYM test_tbl1 FOR test_tbl1;
--删除同义词。
gaussdb=# DROP SYNONYM test_tbl1;
```

相关链接

ALTER SYNONYM，**CREATE SYNONYM**

1.13.10.37 DROP TABLE

功能描述

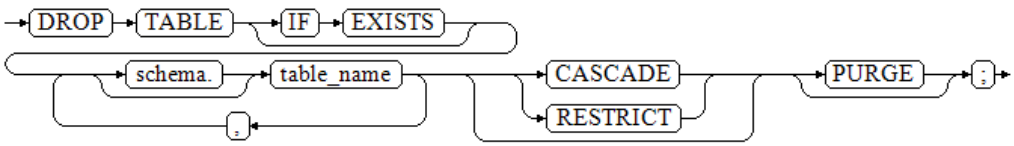
删除指定的表。

注意事项

- 对表进行删除后，依赖该表的索引会被删除，而使用到该表的函数和存储过程将无法执行。删除分区表，会同时删除分区表中的所有分区。
- 仅表的所有者、表所在模式的所有者、被授予了表的DROP权限的用户或被授予DROP ANY TABLE权限的用户可以执行该操作，三权分立关闭时的系统管理员默认拥有该权限。

语法格式

```
DROP TABLE [ IF EXISTS ]
  { [schema.]table_name } [, ...] [ CASCADE | RESTRICT ] [ PURGE ];
```



参数说明

- **IF EXISTS**

  判断指定的表是否存在。
  - 如果指定的表存在，则删除成功。
  - 如果指定的表不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的表不存在。

版权所有 © 华为云计算技术有限公司

- **schema**

  模式名称。可选，用于对表进行指定限制。

- **table_name**

  表名称。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该表的对象。未指定时，默认为RESTRICT。

  - CASCADE：表示允许级联删除依赖于该表的对象（如视图、触发器、索引；注：关联的表对象无法被级联删除）。
  - RESTRICT：表示有依赖于该表的对象存在时，该表无法被删除。

- **PURGE**

  该参数表示即使开启回收站功能，使用DROP TABLE删除表时，也会直接物理删除表，而不是将其放入回收站中。

示例

```
--创建test表。
gaussdb=# CREATE TABLE test(c1 int, c2 int);
--删除test表。
gaussdb=# DROP TABLE IF EXISTS test;

--创建test1表。
gaussdb=# CREATE TABLE test1(c1 int, c2 int);
--创建v_test1视图。
gaussdb=# CREATE VIEW v_test1 AS SELECT * FROM test1 WHERE c1 < 20;
--删除表报错。
gaussdb=# DROP TABLE test1;
ERROR:  table test1 cannot be dropped because other objects depend on it.
DETAIL:  view v_test1 depends on table test1
HINT:  Use DROP ... CASCADE to drop the dependent objects.

--使用CASCADE参数删除test1表，会级联删除掉视图。
gaussdb=# DROP TABLE test1 CASCADE;
NOTICE:  drop cascades to view v_test1
DROP TABLE
```

相关链接

ALTER TABLE，**CREATE TABLE**

1.13.10.38 DROP TABLESPACE

功能描述

删除一个表空间。

注意事项

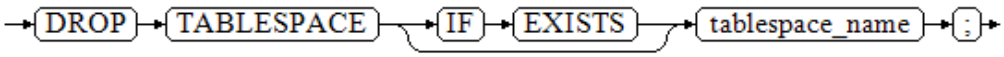
- 仅表空间所有者或者被授予了表空间DROP权限的用户可以执行该操作，系统管理员默认拥有此权限。
- 在删除一个表空间之前，表空间里面不能有任何数据库对象，否则会报错。
- 删除表空间后不支持回滚，因此，不能在事务块内部执行删除表空间操作。
- 执行删除表空间操作时，如果存在执行\db查询操作的其他会话，可能会由于表空间事务的原因导致查询失败，请重新执行\db查询操作。

版权所有 © 华为云计算技术有限公司

- 如果执行DROP TABLESPACE失败，需要再次执行一次DROP TABLESPACE IF EXISTS。

语法格式

```
DROP TABLESPACE [ IF EXISTS ] tablespace_name;
```



参数说明

- **IF EXISTS**

  判断指定的表空间是否存在。

  - 如果指定的表空间存在，则删除成功。
  - 如果指定的表空间不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的表空间不存在。

- **tablespace_name**

  表空间的名称。

  取值范围：已存在的表空间的名称。

示例

```
--创建表空间。
gaussdb=# CREATE TABLESPACE tbs_location1 RELATIVE LOCATION 'test_tablespace/test_tablespace_1';

--删除表空间。
gaussdb=# DROP TABLESPACE tbs_location1;
```

相关链接

ALTER TABLESPACE，**CREATE TABLESPACE**

1.13.10.39 DROP TEXT SEARCH CONFIGURATION

须知

该功能为内部使用功能，不建议用户使用。

功能描述

删除一个已有文本搜索配置。

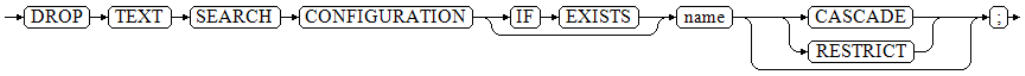
注意事项

仅文本搜索配置的所有者可以执行该操作。

语法格式

```
DROP TEXT SEARCH CONFIGURATION [ IF EXISTS ] name [ CASCADE | RESTRICT ];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **IF EXISTS**

  判断指定的文本搜索配置是否存在。

  - 如果指定的文本搜索配置存在，则删除成功。
  - 如果指定的文本搜索配置不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的文本搜索配置不存在。

- **name**

  要删除的文本搜索配置名称，可以使用模式进行具体指定。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该文本搜索配置的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖文本搜索配置的对象。
  - RESTRICT：若存在依赖于文本搜索配置的对象，则拒绝删除该文本搜索配置。

示例

请参见**CREATE TEXT SEARCH CONFIGURATION**章节的**示例**。

1.13.10.40 DROP TEXT SEARCH DICTIONARY

须知

该功能为内部使用功能，不建议用户使用。

功能描述

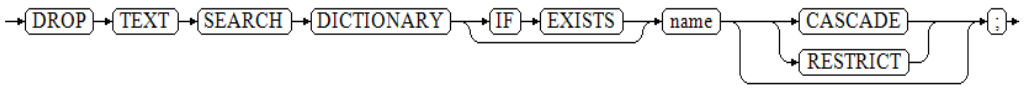
删除一个已有的全文检索词典。

注意事项

- 预定义词典模板不支持删除操作。
- 只有词典的所有者可以执行该操作，系统管理员默认拥有此权限。
- 谨慎执行显式指定CASCADE的删除操作，该操作将级联删除使用该词典的文本搜索配置（TEXT SEARCH CONFIGURATION）。

语法格式

```
DROP TEXT SEARCH DICTIONARY [ IF EXISTS ] name [ CASCADE | RESTRICT ];
```



版权所有 © 华为云计算技术有限公司

参数说明

- **IF EXISTS**

  判断指定的全文检索词典是否存在。

  - 如果指定的全文检索词典存在，则删除成功。
  - 如果指定的全文检索词典不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的全文检索词典不存在。

- **name**

  要删除的词典名称（可指定模式名，否则默认在当前模式下）。

  取值范围：已存在的词典名。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该词典的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖于该词典的对象，并依次删除依赖于这些对象的所有对象。
  - RESTRICT：如果存在依赖词典的对象，则拒绝删除该词典。

示例

```
--创建词典english。
gaussdb=# CREATE TEXT SEARCH DICTIONARY english (
    TEMPLATE = simple
);

--删除词典english。
gaussdb=# DROP TEXT SEARCH DICTIONARY english;
```

相关链接

CREATE TEXT SEARCH DICTIONARY

1.13.10.41 DROP TRIGGER

功能描述

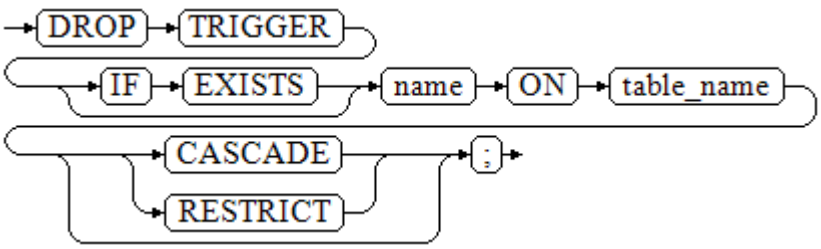
删除一个触发器。

注意事项

触发器的所有者或者被授予了DROP ANY TRIGGER权限的用户可以执行该操作，系统管理员默认拥有此权限。

语法格式

```
DROP TRIGGER [ IF EXISTS ] name ON table_name [ CASCADE | RESTRICT ];
```



版权所有 © 华为云计算技术有限公司

参数说明

- **IF EXISTS**

  判断指定的触发器是否存在。

  - 如果指定的触发器存在，则删除成功。
  - 如果指定的触发器不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的触发器不存在。

- **trigger_name**

  要删除的触发器名称。

  取值范围：已存在的触发器。

- **table_name**

  要删除的触发器所在的表名称。

  取值范围：已存在的含触发器的表。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该触发器的对象。未指定时，默认为RESTRICT。

  - CASCADE：级联删除依赖此触发器的对象。
  - RESTRICT：如果有依赖对象存在，则拒绝删除此触发器。

示例

```
--创建源表及触发表。
gaussdb=# CREATE TABLE test_trigger_src_tbl(id1 INT, id2 INT, id3 INT);
gaussdb=# CREATE TABLE test_trigger_des_tbl(id1 INT, id2 INT, id3 INT);
```

INSERT触发器的使用。

```
--创建insert触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION tri_insert_func() RETURNS TRIGGER AS
        $$
        DECLARE
        BEGIN
                INSERT INTO test_trigger_des_tbl VALUES(NEW.id1, NEW.id2, NEW.id3);
                RETURN NEW;
        END
        $$ LANGUAGE plpgsql;

--创建INSERT触发器。
gaussdb=# CREATE TRIGGER insert_trigger
        BEFORE INSERT ON test_trigger_src_tbl
        FOR EACH ROW
        EXECUTE PROCEDURE tri_insert_func();

--执行INSERT触发事件并检查触发结果。
gaussdb=# INSERT INTO test_trigger_src_tbl VALUES(100,200,300);

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 300
(1 row)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //查看触发操作是否生效。
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 300
(1 row)
```

UPDATE触发器的使用。

版权所有 © 华为云计算技术有限公司

```
--创建UPDATE触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION tri_update_func() RETURNS TRIGGER AS
$$
DECLARE
BEGIN
UPDATE test_trigger_des_tbl SET id3 = NEW.id3 WHERE id1=OLD.id1;
RETURN OLD;
END
$$ LANGUAGE plpgsql;

--创建UPDATE触发器。
gaussdb=# CREATE TRIGGER update_trigger
        AFTER UPDATE ON test_trigger_src_tbl
        FOR EACH ROW
        EXECUTE PROCEDURE tri_update_func();

--执行UPDATE触发事件并检查触发结果。
gaussdb=# UPDATE test_trigger_src_tbl SET id3=400 WHERE id1=100;

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 400
(1 row)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //查看触发操作是否生效。
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 400
(1 row)
```

DELETE触发器的使用。

```
--创建DELETE触发器函数。
gaussdb=# CREATE OR REPLACE FUNCTION TRI_DELETE_FUNC() RETURNS TRIGGER AS
        $$
        DECLARE
        BEGIN
                DELETE FROM test_trigger_des_tbl WHERE id1=OLD.id1;
                RETURN OLD;
        END
        $$ LANGUAGE plpgsql;

--创建DELETE触发器。
gaussdb=# CREATE TRIGGER delete_trigger BEFORE DELETE ON test_trigger_src_tbl FOR EACH ROW
EXECUTE PROCEDURE tri_delete_func();

--执行DELETE触发事件并检查触发结果。
gaussdb=# DELETE FROM test_trigger_src_tbl WHERE id1=100;

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
(0 rows)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //查看触发操作是否生效。
 id1 | id2 | id3
-----+-----+-----
(0 rows)
```

修改触发器名称。

```
--修改触发器的名称。
gaussdb=# ALTER TRIGGER delete_trigger ON test_trigger_src_tbl RENAME TO delete_trigger_renamed;
```

禁用触发器。

```
--禁用insert_trigger触发器。
gaussdb=# ALTER TABLE test_trigger_src_tbl DISABLE TRIGGER insert_trigger;
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO test_trigger_src_tbl VALUES(100,200,300);

gaussdb=# SELECT * FROM test_trigger_src_tbl;
 id1 | id2 | id3
-----+-----+-----
 100 | 200 | 300
(1 row)

gaussdb=# SELECT * FROM test_trigger_des_tbl; //可以看到触发器没有生效。
 id1 | id2 | id3
-----+-----+-----
(0 rows)

--禁用当前表上所有触发器。
gaussdb=# ALTER TABLE test_trigger_src_tbl DISABLE TRIGGER ALL;
```

删除触发器。

```
gaussdb=# DROP TRIGGER insert_trigger ON test_trigger_src_tbl;

gaussdb=# DROP TRIGGER update_trigger ON test_trigger_src_tbl;

gaussdb=# DROP TRIGGER delete_trigger_renamed ON test_trigger_src_tbl;
```

删除函数。

```
gaussdb=# DROP FUNCTION tri_insert_func;

gaussdb=# DROP FUNCTION tri_update_func;

gaussdb=# DROP FUNCTION tri_delete_func;
--删除源表及触发表。
gaussdb=# DROP TABLE test_trigger_src_tbl;
gaussdb=# DROP TABLE test_trigger_des_tbl;
```

相关链接

CREATE TRIGGER，**ALTER TRIGGER**，**ALTER TABLE**

1.13.10.42 DROP TYPE

功能描述

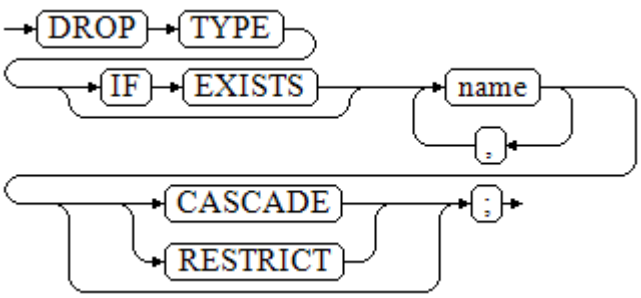
删除一个用户定义的数据类型。

注意事项

仅类型的所有者、被授予了类型DROP权限的用户或者被授予了DROP ANY TYPE权限的用户可以执行该操作，当三权分立开关关闭时，系统管理员默认拥有此权限。

语法格式

```
DROP TYPE [ IF EXISTS ] name [, ...] [ CASCADE | RESTRICT ];
```



版权所有 © 华为云计算技术有限公司

参数说明

- **IF EXISTS**

  判断指定的类型是否存在。

  - 如果指定的类型存在，则删除成功。
  - 如果指定的类型不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的类型不存在。

- **name**

  要删除的类型名，可以使用模式进行具体指定。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该类型的对象。未指定时，默认为RESTRICT。

  - **CASCADE**：级联删除依赖该类型的对象（如字段、函数、操作符等）。
  - **RESTRICT**：如果存在依赖对象，则拒绝删除该类型。

示例

请参见**CREATE TYPE**章节的**示例**。

相关链接

CREATE TYPE，**ALTER TYPE**

1.13.10.43 DROP USER

功能描述

删除指定的用户。

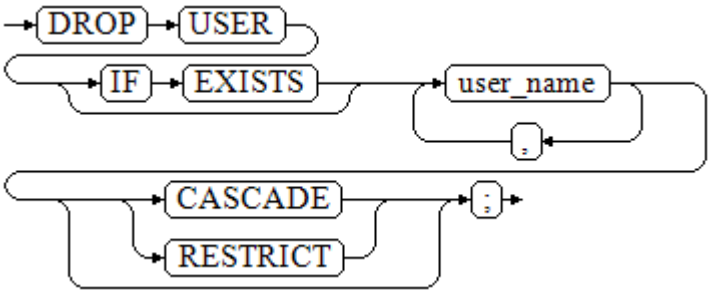
注意事项

- 执行DROP USER操作需要具备创建用户的权限才能执行，当执行删除用户的指令成功后，同名的Schema也会同时删除。
- 当显式指定CASCADE级联删除依赖用户的对象时，如果级联对象处于锁定状态，此级联对象无法被删除，直到对象被解锁或锁定级联对象的线程终止之后再删除对象。
- 在数据库中删除用户时，如果依赖用户的对象在其他数据库中或者依赖用户的对象是其他数据库，请先手动删除其他数据库中的依赖对象或直接删除依赖数据库，再删除用户，即DROP USER不支持跨数据库进行级联删除。
- 跨数据库删除用户前，需先删除该用户所在数据库中由该用户创建的同名Schema。
- 在删除用户时，需要先删除该用户拥有的所有对象并且收回该用户在其他对象上的权限，或者通过指定CASCADE级联删除该用户拥有的对象和被授予的权限。
- 在多租户场景下，删除组用户时，业务用户也会同时被删除，如果指定CASCADE级联删除组用户时，删除业务用户时也需同时显式指定CASCADE。
- 如果用户下存在创建GDS外表时指定的错误表，则无法通过DROP USER指定CASCADE关键字直接删除用户。

版权所有 © 华为云计算技术有限公司

语法格式

```
DROP USER [ IF EXISTS ] user_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的用户是否存在。
  - 如果指定的用户存在，则成功删除。
  - 如果指定的用户不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的用户不存在。

- **user_name**

  需要删除的用户名。

  取值范围：数据库中已存在的用户名。

- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该用户的对象。未指定时，默认为RESTRICT。
  - CASCADE：级联删除依赖用户的对象，并收回授予该用户的权限。
  - RESTRICT：如果用户存在依赖的对象或被授予了其他对象的权限，则拒绝删除该用户。

  说明

  在GaussDB中，配置文件gaussdb.conf中的配置参数enable_kill_query影响级联删除用户对象的行为：
  - 当参数enable_kill_query为on，且使用CASCADE模式删除用户时，会自动结束锁定用户级联对象的线程，并直接删除用户。
  - 当参数enable_kill_query为off，且使用CASCADE模式删除用户时，会等待锁定级联对象的线程结束之后再删除用户。

示例

```
--创建用户jim，登录密码为********。
gaussdb=# CREATE USER jim PASSWORD '********';

--删除不存在的tom用户。
gaussdb=# DROP USER IF EXISTS tom;
NOTICE:  role "tom" does not exist, skipping
DROP ROLE

--删除用户。
gaussdb=# DROP USER jim CASCADE;
```

相关链接

ALTER USER，**CREATE USER**

版权所有 © 华为云计算技术有限公司

1.13.10.44 DROP VIEW

功能描述

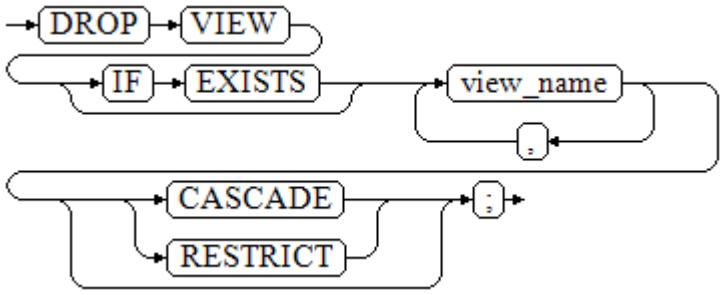
删除数据库中的视图。

注意事项

仅视图的所有者、视图所在模式的所有者、被授予了视图DROP权限的用户或拥有DROP ANY TABLE权限的用户可以执行该操作，三权分立关闭时的系统管理员默认拥有此权限。

语法格式

```
DROP VIEW [ IF EXISTS ] view_name [, ...] [ CASCADE | RESTRICT ];
```



参数说明

- **IF EXISTS**

  判断指定的视图是否存在。
  - 如果指定的视图存在，则删除成功。
  - 如果指定的视图不存在，不会进行删除操作，仅产生一个NOTICE提示，用于告知指定的视图不存在。
- **view_name**

  要删除的视图名称。

  取值范围：已存在的视图。
- **CASCADE | RESTRICT**

  可选，是否级联删除依赖于该视图的对象。未指定时，默认为RESTRICT。
  - CASCADE：级联删除依赖此视图的对象（如其他视图）。
  - RESTRICT：如果有依赖对象存在，则拒绝删除此视图。

示例

```
--创建test_tb1表，并向表中插入100条数据。
gaussdb=# CREATE TABLE test_tb1(col1 int, col2 int);
gaussdb=# INSERT INTO test_tb1 VALUES (generate_series(1,100),generate_series(1,100));

--创建一个col1小于3的视图。
gaussdb=# CREATE VIEW test_v1 AS SELECT * FROM test_tb1 WHERE col1 < 3;

--查看视图。
gaussdb=# SELECT * FROM test_v1;
 col1 | col2
```


版权所有 © 华为云计算技术有限公司

```
------+------
   1 |    1
   2 |    2
(2 rows)

--删除表和视图。
gaussdb=# DROP VIEW test_v1;
gaussdb=# DROP TABLE test_tb1;
```

相关链接

ALTER VIEW，**CREATE VIEW**

1.13.10.45 DROP WEAK PASSWORD DICTIONARY

功能描述

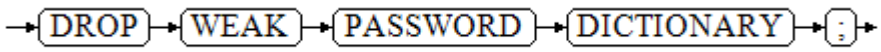
清空弱口令字典。

注意事项

- 仅初始用户、系统管理员和安全管理员可以执行该操作。
- 用户可以通过该语句清空系统表gs_global_config中的全部弱口令。

语法格式

```
DROP WEAK PASSWORD DICTIONARY;
```



示例

```
--向gs_global_config系统表中插入单个弱口令。
gaussdb=# CREATE WEAK PASSWORD DICTIONARY WITH VALUES ('********');

--查看gs_global_config系统表现有的弱口令。
gaussdb=# SELECT * FROM gs_global_config WHERE NAME LIKE 'weak_password';
      name       |    value
-----------------+--------------
 weak_password   | ********
(1 rows)

--清空gs_global_config系统表中所有弱口令。
gaussdb=# DROP WEAK PASSWORD DICTIONARY;

--查看现有弱口令。
gaussdb=# SELECT * FROM gs_global_config WHERE NAME LIKE 'weak_password';
 name | value
------+-------
(0 rows)
```

相关链接

CREATE WEAK PASSWORD DICTIONARY

1.13.11 E

版权所有 © 华为云计算技术有限公司

1.13.11.1 EXECUTE

功能描述

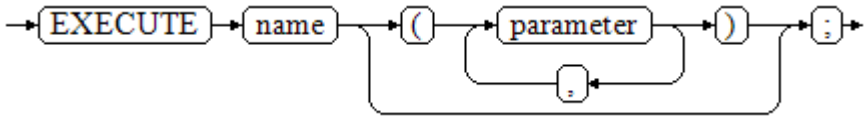
执行一个准备好的预备语句。因为预备语句只在会话的生命期里存在，所以预备语句必须是在当前会话中更早用PREPARE语句创建的。

注意事项

- 如果创建预备语句时，PREPARE语句声明了一些参数，那么传递给EXECUTE语句的必须是一个兼容的参数集，否则会出现错误。如果未在预备语句声明中定义参数，则后续EXECUTE语句中传入的任何合法参数绑定都将被系统静默忽略。
- 如果预备语句所依赖的表结构发生变化时（如：添加、删除或修改列定义），缓存计划中期望的结果结构可能与实际的表结构已存在不匹配情况，当后续再执行该预备语句时，将导致非预期结果。

语法格式

```
EXECUTE name [ ( parameter [, ...] ) ];
```



参数说明

- **name**

  要执行的预备语句的名称。

- **parameter**

  给预备语句的参数的具体数值。它必须是一个和生成与创建这个预备语句时指定参数的数据类型相兼容的值的表达式，不支持ROWNUM作为参数。

示例

```
--创建表reason。
gaussdb=# CREATE TABLE reason (
    CD_DEMO_SK          int NOT NULL,
    CD_GENDER           varchar(10),
    CD_MARITAL_STATUS   varchar(10)
);

--为一个INSERT语句创建一个预备语句然后执行它。
gaussdb=# PREPARE insert_reason(int,varchar(10),varchar(10)) AS INSERT INTO reason VALUES($1,$2,$3);
gaussdb=# EXECUTE insert_reason(52, 'AAAAAAAADD', 'reason 52');

--查询数据。
gaussdb=# SELECT * FROM reason;
 cd_demo_sk | cd_gender  | cd_marital_status
------------+------------+-------------------
         52 | AAAAAAAADD | reason 52
(1 row)

--删除表reason。
gaussdb=# DROP TABLE reason;
```


版权所有 © 华为云计算技术有限公司

相关链接

PREPARE，**DEALLOCATE**

1.13.11.2 EXECUTE DIRECT

功能描述

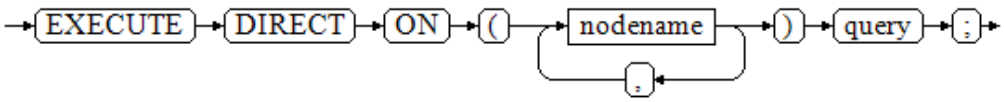
在指定的节点上执行SQL语句。一般情况下，SQL语句的执行是由集群负载自动分配到合适的节点上，EXECUTE DIRECT主要用于数据库维护和测试。

注意事项

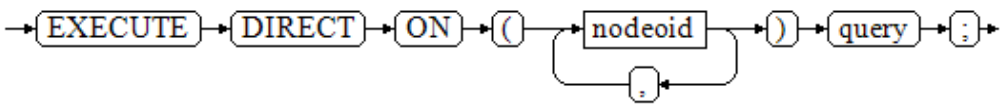
- 当enable_nonsysadmin_execute_direct=off时，只有系统管理员和监控管理员才能执行EXECUTE DIRECT。
- 当enable_udf_execute_direct=off时，禁止EXECUTE DIRECT执行用户自定义函数、高级包DBE_SQL、PKG_SERVICE、DBE_XMLGE中的函数和DBE_UTILITY.EXEC_DDL_STATEMENT。
- 为了各个节点上数据的一致性，SQL语句仅支持SELECT，不允许执行事务语句、DDL、DML。
- 使用此类型语句在指定的DN执行stddev聚集计算时，返回结果集是以三元数组形式返回，如{3, 8, 30}，表示count结果为3，sum结果为8，平方和为30。使用此类型语句在指定的DN执行AVG聚集计算时，返回结果集以二元组形式返回，如{4，2}，表示count结果为4，sum结果为2。
- 当指定多个节点时，不支持agg函数，当query中包含agg函数时，会返回“EXECUTE DIRECT on multinode not support agg functions.”
- 由于CN节点不存储用户表数据，不允许指定CN节点执行用户表上的SELECT查询。
- 不允许执行嵌套的EXECUTE DIRECT语句，即执行的SQL语句不能同样是EXECUTE DIRECT语句，此时可直接执行最内层EXECUTE DIRECT语句代替。
- agg函数查询结果与直接在CN上查询不一致，会返回多个信息，不支持array_avg函数。
- 参数为nodeoid或者nodeoid list时，仅支持分布式下参数enable_direct_standby_datanodes成功设置时才生效。实现到指定DN节点查询，不支持无效或重复nodeoid，以及不支持nodeoid和nodename混合使用。

语法格式

```
EXECUTE DIRECT ON ( nodename [, ... ] ) query ;
```

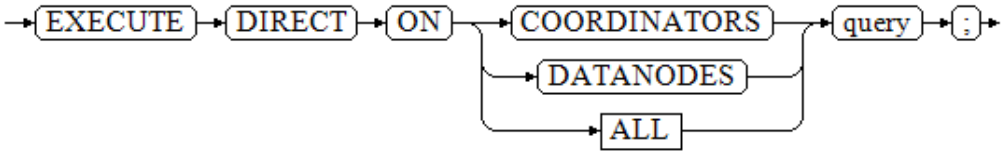



```
EXECUTE DIRECT ON ( nodeoid [, ... ] ) query;
```




```
EXECUTE DIRECT ON { COORDINATORS | DATANODES | ALL } query;
```


版权所有 © 华为云计算技术有限公司



参数说明

- **nodename**

  节点名称。

  取值范围：已存在的节点。

- **nodeoid**

  节点oid。

  取值范围：已存在的DN节点oid，查看**PGXC_NODE**系统表获取。

- **query**

  要执行查询语句。

- **COORDINATORS**

  在所有coordinator执行查询语句

- **DATANODES**

  在所有datanode执行查询语句

- **ALL**

  在所有coordinator和datanode执行查询语句

示例

```
--查询当前集群的节点名称。
gaussdb=# SELECT node_name,oid FROM pgxc_node WHERE node_type IN ('C','D') ORDER BY 1;
     node_name     |  oid
-------------------+-------
 cn_5001           | 15517
 cn_5002           | 16387
 cn_5003           | 16391
 dn_6001_6002_6003 | 16384
 dn_6004_6005_6006 | 16388
 dn_6007_6008_6009 | 16392
(6 rows)


--创建以range分布的表，并插入数据。
gaussdb=# CREATE TABLE tbl_range(
  id       INT,
  name     VARCHAR(20),
  province VARCHAR(60),                       --省
  country   VARCHAR(30) DEFAULT 'China'       --国籍
)DISTRIBUTE BY RANGE(id)(
  SLICE s1 VALUES LESS THAN (100) DATANODE dn_6001_6002_6003,
  SLICE s2 VALUES LESS THAN (200) DATANODE dn_6004_6005_6006,
  SLICE s3 VALUES LESS THAN (MAXVALUE) DATANODE dn_6007_6008_6009
);
gaussdb=# INSERT INTO tbl_range VALUES (52,'Bob','Beijing','China'),
                                       (100,'Ben','Shanghai','China'),
                                       (150,'Scott','Guangzhou','China'),
                                       (300,'Jordan','Beijing','China');

--查询该表分布在dn_6007_6008_6009节点上的数据。
gaussdb=# EXECUTE DIRECT ON(dn_6007_6008_6009) 'SELECT * FROM tbl_range';
```


版权所有 © 华为云计算技术有限公司

```
 id  |  name  | province | country
-----+--------+----------+---------
 300 | Jordan | Beijing  | China
(1 row)

--删除表tbl_range。
gaussdb=# DROP TABLE tbl_range;
```

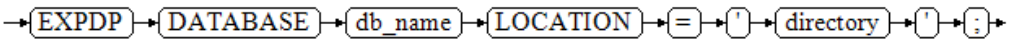
1.13.11.3 EXPDP DATABASE

功能描述

导出DATABASE的全部物理文件。

语法格式

```
EXPDP DATABASE db_name LOCATION = 'directory';
```



参数说明

- **db_name**
  导出的库名。
- **directory**
  导出文件的存储目录。

示例

```
--EXPDP DATABASE语法用于细粒度备份恢复，由备份恢复工具调用，用户直接调用会报错，错误信息：
ERROR: Only auxdb is allowed to execute expdp cmd.
gaussdb=# EXPDP DATABASE test LOCATION = '/data1/expdp/database';
```

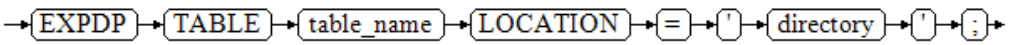
1.13.11.4 EXPDP TABLE

功能描述

导出与表相关的索引、sequence、分区、toast、toast index等全部文件。

语法格式

```
EXPDP TABLE table_name LOCATION = 'directory';
```



参数说明

- **table_name**
  导出的表名。
- **directory**
  导出文件的存储目录。

版权所有 © 华为云计算技术有限公司

示例

```
--EXPDP TABLE语法用于细粒度备份恢复，由备份恢复工具调用，用户直接调用会报错，错误信息：ERROR:
Only auxdb is allowed to execute expdp cmd.
gaussdb=# EXPDP TABLE test_t LOCATION = '/data1/expdp/table0';
```

1.13.11.5 EXPLAIN

功能描述

显示SQL语句的执行计划。主要包括：

- 显示SQL语句所引用的表的扫描方式，如简单的顺序扫描或索引扫描等。
- 如果SQL语句中引用了多个表，执行计划还会显示多表之间的JOIN方式。

注意事项

- 执行计划的最主要指标是SQL语句的预计执行开销，这是计划生成器估算执行该语句所需时间的度量。
- 如果指定了ANALYZE选项，该SQL语句将被执行，并根据实际运行结果显示统计数据。统计数据用于作为判断计划生成器的估计值是否接近实际值的依据，主要包括每个计划节点的时间总开销（以毫秒为单位）和实际返回的总行数。
- 在指定ANALYZE选项时，语句会被执行。如果用户想使用EXPLAIN分析INSERT，UPDATE，DELETE，CREATE TABLE AS或EXECUTE语句，而不想改动数据（执行这些语句会影响数据），请使用如下方法。

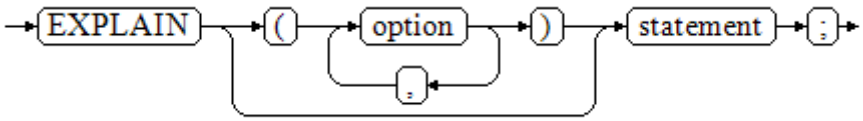
  ```
  START TRANSACTION;
  EXPLAIN ANALYZE ...;
  ROLLBACK;
  ```

- 当使用EXPLAIN查看执行计划时，审计日志记录的审计类型由分析的具体SQL决定。

语法格式

- 显示SQL语句的执行计划，支持多种选项，对选项顺序无要求。

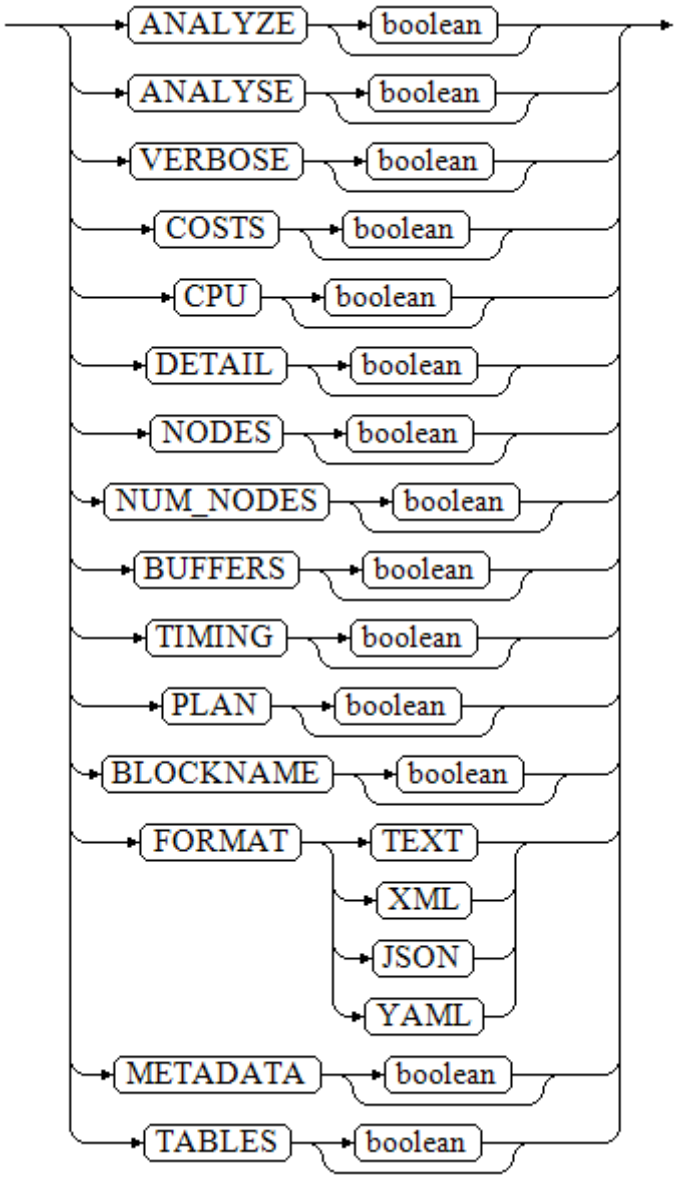
  ```
  EXPLAIN [ ( option [, ...] ) ] statement;
  ```

  其中选项option子句的语法为。

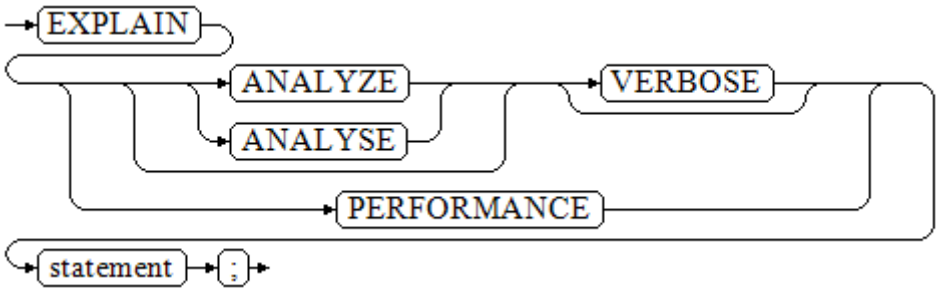
  ```
  {
    ANALYZE [ boolean ] |
    ANALYSE [ boolean ] |
    VERBOSE [ boolean ] |
    COSTS [ boolean ] |
    CPU [ boolean ] |
    DETAIL [ boolean ] |
    NODES [ boolean ] |
    NUM_NODES [ boolean ] |
    BUFFERS [ boolean ] |
    TIMING [ boolean ] |
    PLAN [ boolean ] |
    BLOCKNAME [ boolean ] |
    FORMAT { TEXT | XML | JSON | YAML } |
    METADATA [ boolean ] |
    TABLES [ boolean ]
  }
  ```


版权所有 © 华为云计算技术有限公司



- 显示SQL语句的执行计划，且要按顺序给出选项。

```
EXPLAIN { [ ANALYZE | ANALYSE ] [ VERBOSE ] | PERFORMANCE } statement;
```



参数说明

- **statement**

  指定要分析的SQL语句。

版权所有 © 华为云计算技术有限公司

- **ANALYZE boolean | ANALYSE boolean**

  显示实际运行时间和其他统计数据。当两个参数同时使用时，在option中排在后面的一个生效。

  取值范围：

  - TRUE（缺省值）：显示实际运行时间和其他统计数据。
  - FALSE：不显示。

- **VERBOSE boolean**

  显示有关计划的额外信息。

  取值范围：

  - TRUE（缺省值）：显示额外信息。
  - FALSE：不显示。

- **COSTS boolean**

  包括每个规划节点的估计总成本，以及估计的行数和每行的宽度。

  取值范围：

  - TRUE（缺省值）：显示估计总成本和宽度。
  - FALSE：不显示。

- **CPU boolean**

  打印CPU的使用情况的信息。需要结合ANALYZE或ANALYSE选项一起使用。

  取值范围：

  - TRUE（缺省值）：显示CPU的使用情况。
  - FALSE：不显示。

- **DETAIL boolean**

  打印DN上的信息。需要结合ANALYZE或ANALYSE选项一起使用。

  取值范围：

  - TRUE（缺省值）：打印DN的信息。
  - FALSE：不打印。

- **NODES boolean**

  打印query执行的节点信息。

  取值范围：

  - TRUE（缺省值）：打印执行的节点的信息。
  - FALSE：不打印。

- **NUM_NODES boolean**

  打印执行中的节点的个数信息。

  取值范围：

  - TRUE（缺省值）：打印DN个数的信息。
  - FALSE：不打印。

- **BUFFERS boolean**

  包括缓冲区的使用情况的信息。需要结合ANALYZE或ANALYSE选项一起使用。

  取值范围：

  - TRUE：显示缓冲区的使用情况。

版权所有 © 华为云计算技术有限公司

  - FALSE（缺省值）：不显示。

- **TIMING boolean**

  包括实际的启动时间和花费在输出节点上的时间信息。需要结合ANALYZE或ANALYSE选项一起使用。

  取值范围：

  - TRUE（缺省值）：显示启动时间和花费在输出节点上的时间信息。
  - FALSE：不显示。

- **PLAN boolean**

  是否将执行计划存储在plan_table中。当该选项开启时，会将执行计划存储在plan_table中，不打印到当前屏幕，因此该选项为on时，不能与其他选项同时使用。

  取值范围：

  - TRUE（缺省值）：将执行计划存储在plan_table中，不打印到当前屏幕。执行成功返回EXPLAIN SUCCESS。
  - FALSE：不存储执行计划，将执行计划打印到当前屏幕。

- **BLOCKNAME boolean**

  是否显示计划的每个操作所处于的查询块。当该选项开启时，会将每个操作所处于的查询块的名字输出在Query Block列上，方便用户获取查询块名字，并使用Hint修改执行计划：

  - TRUE（缺省值）：显示计划时，将每个操作所处于的查询块的名字输出在新增列Query Block列上。该选项需要在pretty模式下使用。请参见《SQL调优指南》中“使用Plan Hint进行调优 > 指定Hint所处的查询块Queryblock”章节。
  - FALSE：不对计划显示产生影响。

- **FORMAT**

  指定输出格式。

  取值范围：TEXT，XML，JSON和YAML。

  默认值：TEXT。

- **PERFORMANCE**

  使用此选项时，即打印执行中的所有相关信息。下述为部分信息描述：

  - ex c/r：代表平均每行使用cpu周期数，等于（ex cyc）/（ex row）。
  - ex row：执行行数。
  - ex cyc：代表使用的cpu周期数。
  - inc cyc：代表包含子节点使用的总cpu周期数。
  - shared hit：代表算子的share buffer命中情况。
  - loops：算子循环执行次数。
  - total_calls：生成元素总数。
  - remote query poll time stream gather：算子用于侦听各DN数据到达CN的网络poll时间。
  - deserialize time：反序列化所需时间。
  - estimated time：估计时间。
  - Network Poll Time：分布式stream网络通信时，表示libcomm接收侧等待数据耗时。

版权所有 © 华为云计算技术有限公司

  - Stream Send time：分布式stream网络通信时，表示libcomm或libpq发送数据耗时。
  - OS Kernel Send time：分布式stream网络通信时，表示操作系统层发送数据耗时，大于0才会显示。
  - Wait Quota time：分布式stream网络通信时，表示libcomm等待对端发送quota流控大小耗时，大于0才会显示。
  - Data Serialize time：分布式stream网络通信时，表示数据序列化时间 。
  - Data Copy time：分布式stream网络通信时，表示数据复制时间，大于0才会显示。

- **METADATA boolean**

  是否显示语句元数据内容。当不指定boolean时，默认为TRUE。该部分内容仅在语句使用stream类型计划时展示，下述为输出信息描述：

  - 概要对象定义，涵盖对象包括语句中所引用的表、索引、视图、函数、操作符。
  - 概要统计信息，分为表级和列级两部分，其公共字段包括：
    - nspname：模式名，对应pg_namespace.nspname。
    - relname：表名，对应表或索引的pg_class.relname。
    - partname：分区名，对应分区的pg_partition.relname。
    - subpartname：子分区名，对应子分区的pg_partition.relname。

    其他表级统计信息的字段包括：

    - reltuples：统计的表总行数，对应pg_class/pg_partition.reltuples。
    - relpages：统计的表总页数，对应pg_class/pg_partition.relpages。
    - analyze_time：该表或索引上次触发ANANLYZE的时间戳，对应pg_stat_get_last_analyze_time/gs_stat_get_partition_last_analyze_time的返回值。
    - rows_changed：自上次执行ANALYZE后该表或索引的变更行数，对应pg_stat_get_tuples_changed/pg_stat_get_partition_tuples_changed的返回值。
    - rows_inserted：自上次执行ANALYZE后该表或索引的插入行数，该值为空。

    其他列级统计信息（包括单列与多列）的字段包括：

    - colnames：列名，单列对应pg_stats.attname，多列对应pg_ext_stats.attname中列号所代表的列名集合。
    - nullfrac：空值率，对应pg_stats/pg_ext_stats.nullfrac。
    - avgwidth：平均行宽，对应pg_stats/pg_ext_stats.avgwidth。
    - distinct：不同值的数量，对应pg_stats/pg_ext_stats.n_distinct。
    - dndistinct：DN上不同值的数量，对应pg_stats/pg_ext_stats.n_dndistinct。

版权所有 © 华为云计算技术有限公司

  - mcvs_top5：高频值，对应pg_stats/pg_ext_stats.most_common_vals，只取频率最高的5个。
  - mcfs_top5：高频值的频率，与mcvs_top5一一对应，对应pg_stats/pg_ext_stats.most_common_freqs。
  - hist_qrtl：直方图桶值，单列对应pg_stats.histogram_bounds，只取四分位边界上的5个值，多列为空。
  - correlation：相关性，单列对应pg_stats.correlation，多列为0。
- 在Query Others中，包含以下内容：
  - current time：当前时间戳，用于与上述analyze_time进行对比。
  - use_hints: Yes, <string>：如果语句有Hint生效，则打印该提示，并列出生效的Hint。

取值范围：

- TRUE（默认值）：显示语句元信息，该选项需要在pretty模式下使用。
- FALSE：不显示。

- **TABLES boolean**

  是否显示SQL语句所依赖的表对象信息，包括表与物化视图。当不指定boolean时，默认为TRUE。该部分内容仅在语句使用stream类型计划时展示。

  - TRUE（默认值）：显示语句所依赖的表对象信息，该选项需要在pretty模式下使用。
  - FALSE：不显示。

示例 1

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA tpcds;

--创建表tpcds.customer_address。
gaussdb=# CREATE TABLE tpcds.customer_address
(
ca_address_sk         INTEGER           NOT NULL,
ca_address_id         CHARACTER(16)     NOT NULL
);

--向表中插入多条记录。
gaussdb=# INSERT INTO tpcds.customer_address VALUES (5000, 'AAAAAAAABAAAAAAA'),(10000,
'AAAAAAAACAAAAAAA');

--创建一个表tpcds.customer_address_p1。
gaussdb=# CREATE TABLE tpcds.customer_address_p1 AS TABLE tpcds.customer_address;

--修改explain_perf_mode为normal。
gaussdb=# SET explain_perf_mode=normal;

--显示表简单查询的执行计划。
gaussdb=# EXPLAIN SELECT * FROM tpcds.customer_address_p1;
QUERY PLAN
--------------------------------------------------
Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
Node/s: All datanodes
(2 rows)

--使用ANALYZE选项,在输出中增加运行时间统计。
gaussdb=# EXPLAIN ANALYZE SELECT * FROM tpcds.customer_address_p1;
                                QUERY PLAN
```


版权所有 © 华为云计算技术有限公司

```
-----------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0) (actual time=1.754..3.218 rows=2 loops=1)
   Node/s: All datanodes
 Total runtime: 3.272 ms
(3 rows)

--使用ANALYZE选项和CPU选项，输出CPU的使用信息。
gaussdb=# EXPLAIN (ANALYZE,CPU)SELECT * FROM tpcds.customer_address_p1;
                                    QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0) (actual time=1.996..2.214 rows=2 loops=1)
   Node/s: All datanodes
   (CPU: ex c/r=25694795469106248, ex row=2, ex cyc=51389590938212496, inc cyc=51389590938212496)
 Total runtime: 2.251 ms
(4 rows)

--以JSON格式输出的执行计划（explain_perf_mode为normal时）。
gaussdb=# EXPLAIN(FORMAT JSON) SELECT * FROM tpcds.customer_address_p1;
              QUERY PLAN
--------------------------------------
 [                              +
   {                            +
     "Plan": {                    +
       "Node Type": "Data Node Scan",+
       "Startup Cost": 0.00,        +
       "Total Cost": 0.00,          +
       "Plan Rows": 0,              +
       "Plan Width": 0,             +
       "Node/s": "All datanodes"    +
     }                            +
   }                            +
 ]
(1 row)


--以YAML格式输出的执行计划（explain_perf_mode为normal时）。
gaussdb=# EXPLAIN(FORMAT YAML) SELECT * FROM tpcds.customer_address_p1 WHERE
ca_address_sk=10000;
            QUERY PLAN
---------------------------------
 - Plan:                       +
     Node Type: "Data Node Scan"+
     Startup Cost: 0.00         +
     Total Cost: 0.00           +
     Plan Rows: 0               +
     Plan Width: 0              +
     Node/s: "dn_6005_6006"
(1 row)

--禁止开销估计的执行计划。
gaussdb=# EXPLAIN(COSTS FALSE)SELECT * FROM tpcds.customer_address_p1 WHERE
ca_address_sk=10000;
       QUERY PLAN
------------------------
 Data Node Scan
   Node/s: dn_6005_6006
(2 rows)

--带有聚集函数查询的执行计划。
gaussdb=# EXPLAIN SELECT SUM(ca_address_sk) FROM tpcds.customer_address_p1 WHERE
ca_address_sk<10000;
                                QUERY PLAN
---------------------------------------------------------------------------------
 Aggregate  (cost=18.19..14.32 rows=1 width=4)
   ->  Streaming (type: GATHER)  (cost=18.19..14.32 rows=3 width=4)
         Node/s: All datanodes
         ->  Aggregate  (cost=14.19..14.20 rows=3 width=4)
               ->  Seq Scan on customer_address_p1  (cost=0.00..14.18 rows=10 width=4)
                     Filter: (ca_address_sk < 10000)
```


版权所有 © 华为云计算技术有限公司

```
(6 rows)

--删除表tpcds.customer_address_p1。
gaussdb=# DROP TABLE tpcds.customer_address_p1;

--删除表tpcds.customer_address。
gaussdb=# DROP TABLE tpcds.customer_address;

--删除SCHEMA。
gaussdb=# DROP SCHEMA tpcds CASCADE;
```

示例 2

```
# 展示SQL语句元信息。
# 使用gsql连接数据库。
[omm@localhost ~]$ gsql -d postgres -p 8000 -r

-- 设置explain_perf_mode。
gaussdb=# SET explain_perf_mode = pretty;

-- 关闭多列统计信息贝叶斯网络，以收集普通的多列MCV统计信息。
gaussdb=# SET enable_ai_stats = off;

-- 关闭自动参数化，避免语句跳过解析。
gaussdb=# SET cursor_sharing = off;

-- 创建SCHEMA，并默认使用该SCHEMA。
gaussdb=# CREATE SCHEMA em;
gaussdb=# SET search_path = em;

-- 创建表em.t1。
gaussdb=# CREATE TABLE em.t1 (
    c1 integer
) DISTRIBUTE BY HASH (c1) PARTITION BY RANGE (c1) (
    PARTITION p1 VALUES LESS THAN (1000),
    PARTITION p2 VALUES LESS THAN (MAXVALUE)
);

-- 在em.t1上增加索引。
gaussdb=# CREATE INDEX em.idx_t1_c1 ON em.t1(c1) LOCAL;

-- 插入多条记录。
gaussdb=# INSERT INTO em.t1 SELECT (s % 2000)::int FROM pg_catalog.generate_series(1, 100000) s;

-- ANALYZE em.t1。
gaussdb=# ANALYZE em.t1 WITH ALL COMPLETE;

-- 创建表em.t2。
gaussdb=# CREATE TABLE em.t2 (c1 integer, c2 integer) WITH (storage_type=ASTORE) DISTRIBUTE BY
HASH (c1);

-- 插入多条记录。
gaussdb=# INSERT INTO em.t2 SELECT (s % 6)::int, (s % 3)::int FROM pg_catalog.generate_series(1, 24) s;

-- ANALYZE em.t2。
gaussdb=# ANALYZE em.t2;

-- ANALYZE em.t2的c1列和c2列。
gaussdb=# ANALYZE em.t2((c1,c2));

-- 创建物化视图em.mv。
gaussdb=# CREATE MATERIALIZED VIEW em.mv AS SELECT c1 FROM t2 WHERE c1 = 2 AND c2 = 2;
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.

-- ANALYZE物化视图。
gaussdb=# ANALYZE em.mv;

-- 使用METADATA选项，输出语句涉及的元信息。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# EXPLAIN (METADATA ON) SELECT t.c1, v.c1 FROM em.t1 t, em.mv v WHERE t.c1 < 500 AND
v.c1=2;
 id |                          operation                          | E-rows | E-width | E-costs
----+-------------------------------------------------------------+--------+---------+---------
  1 | ->  Streaming (type: GATHER)                                | 100584 |       8 | 4764.48
  2 |    ->  Nested Loop (3,4)                                    | 100584 |       8 | 573.48
  3 |       ->  Partitioned Index Only Scan using idx_t1_c1 on t1 t |  25146 |       4 | 150.24
  4 |       ->  Materialize                                       |     12 |       4 | 4.15
  5 |          ->  Streaming(type: BROADCAST)                     |     12 |       4 | 4.13
  6 |             ->  Seq Scan on mv v                            |      4 |       4 | 3.05
(6 rows)

      Predicate Information (identified by plan id)
-----------------------------------------------------------
   3 --Partitioned Index Only Scan using idx_t1_c1 on t1 t
         Index Cond: (c1 < 500)
         Selected Partitions:  1
   6 --Seq Scan on mv v
         Filter: (c1 = 2)
(5 rows)

                                                                                                            Object
Definition

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------
 CREATE TABLE em.t1 (c1 integer) WITH (orientation=row, compression=no, storage_type=USTORE,
segment=off) DISTRIBUTE BY HASH(c1) TO GROUP group1 PARTITION BY RANGE (c1) (PARTITION p1
VALUES LESS THAN (1000) TABLESPACE pg_default) /*Number of partitions
: 2*/;
 CREATE INDEX em.idx_t1_c1 ON em.t1 USING ubtree (c1) LOCAL WITH (storage_type=USTORE)
TABLESPACE pg_default;
 CREATE TABLE em.t2 (c1 integer, c2 integer) WITH (orientation=row, storage_type=astore,
compression=no) DISTRIBUTE BY HASH(c1) TO GROUP group1;
 CREATE MATERIALIZED VIEW em.mv AS SELECT t2.c1 FROM t2 WHERE ((t2.c1 = 2) AND (t2.c2 = 2));
(4 rows)

 nspname |  relname  | partname  | subpartname | reltuples | relpages |         analyze_time          |
rows_changed | rows_inserted
---------+-----------+-----------+-------------+-----------+----------+-------------------------------+--------------
+---------------
 em      | t1        |           |             | 100000    | 384      | 2026-07-06 11:12:08.316871+08 | 0            |
 em      | t1        | p1        |             | 50000     | 192      | 2026-07-06 11:12:08.61814+08  | 0            |
 em      | idx_t1_c1 |           |             | 100000    | 58       |                               | 0            |
 em      | idx_t1_c1 | p1_c1_idx |             | 100000    | 58       |                               | 0            |
 em      | t2        |           |             | 48        | 3        | 2026-07-06 11:15:38.374133+08 | 0            |
 em      | mv        |           |             | 4         | 3        | 2026-07-06 11:15:45.16283+08  | 0            |
(6 rows)

 nspname | relname | partname | subpartname | colnames | nullfrac | avgwidth | distinct | dndistinct |
mcvs_top5         |                      mcfs_top5                      |       hist_qrtl        | correlation
---------+---------+----------+-------------+----------+----------+----------+----------+------------
+--------------------------+-----------------------------------------------------+------------------------+-------------
 em      | t1      |          |             | c1       | 0        | 4        | 2000     | 667        | {358,132,188,267,613}     |
{.001,.000833333,.000833333,.000833333,.000833333} | {0,513,1016,1516,1999} | .715956
 em      | t1      | p1       |             | c1       | 0        | 4        | 1000     | 667        | {594,996,7,454,591}       |
{.00133333,.00133333,.0013,.0013,.0013}            | {0,250,497,749,999}    | .0195734
 em      | t2      |          |             | c1       | 0        | 4        | -.125    | -.125      | {0,1,2,3,4}               |
{.166667,.166667,.166667,.166667,.166667}          |                        | .462875
 em      | t2      |          |             | c2       | 0        | 4        | 3        | 1          | {0,1,2}                   |
{.333333,.333333,.333333}                          |                        | .865393
 em      | t2      |          |             | {c1,c2}  | 0        | 8        | -.125    | -.125      | {{0,1,2,3,4},{0,1,2,0,1}} |
{.166667,.166667,.166667,.166667,.166667}          |                        | 0
 em      | mv      |          |             | c1       | 0        | 4        | -.25     | 1          | {2}                       |
{1}                                                |                        | 1
(6 rows)

         ====== Query Others =====
```


版权所有 © 华为云计算技术有限公司

```
--------------------------------------------
 current time: 2026-07-06 11:15:52.968669+08
(1 row)

-- 使用TABLES选项，打印SQL语句所依赖的表对象信息列表。
gaussdb=# EXPLAIN (TABLES) SELECT t.c1, v.c1 FROM em.t1 t, em.mv v WHERE t.c1 < 500 AND v.c1=2;
 id |                        operation                        | E-rows | E-width | E-costs
----+---------------------------------------------------------+--------+---------+---------
  1 | ->  Streaming (type: GATHER)                            | 100584 |       8 | 4764.48
  2 |    ->  Nested Loop (3,4)                                | 100584 |       8 | 573.48
  3 |       ->  Partitioned Index Only Scan using idx_t1_c1 on t1 t |  25146 |       4 | 150.24
  4 |       ->  Materialize                                   |     12 |       4 | 4.15
  5 |          ->  Streaming(type: BROADCAST)                 |     12 |       4 | 4.13
  6 |             ->  Seq Scan on mv v                        |      4 |       4 | 3.05
(6 rows)

      Predicate Information (identified by plan id)
-------------------------------------------------------------
   3 --Partitioned Index Only Scan using idx_t1_c1 on t1 t
         Index Cond: (c1 < 500)
         Selected Partitions:  1
   6 --Seq Scan on mv v
         Filter: (c1 = 2)
(5 rows)

           Tables Referenced
-------------------------------------------
 ('em', 't1'), ('em', 't2'), ('em', 'mv')
(1 row)

-- 清理创建对象。
gaussdb=# DROP SCHEMA em CASCADE;
gaussdb=# \q
```

相关链接

ANALYZE | ANALYSE

1.13.11.6 EXPLAIN PLAN

功能描述

通过EXPLAIN PLAN命令可以将查询执行的计划信息存储于PLAN_TABLE表中。与EXPLAIN命令不同的是，EXPLAIN PLAN仅将计划信息进行存储，而不会打印到屏幕。

注意事项

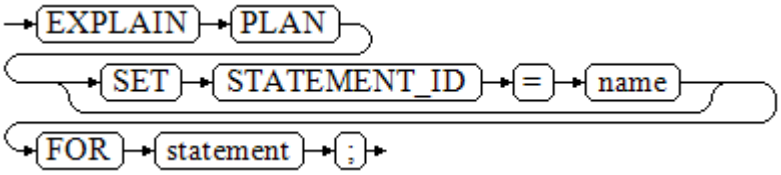
- EXPLAIN PLAN不支持在DN上执行。
- 对于执行错误的SQL语句无法进行计划信息的收集。
- PLAN_TABLE中的数据是session级生命周期并且session隔离和用户隔离，用户只能看到当前session、当前用户的数据。
- PLAN_TABLE无法与GDS外表进行关联查询。
- 对于不能下推的查询，执行EXPLAIN PLAN后PLAN_TABLE中的object仅收集到REMOTE_QUERY或CTE等信息。详见**示例 2**。
- EXPLAIN中的PLAN选项表示需要将计划信息存储于PLAN_TABLE中，存储成功将返回“EXPLAIN SUCCESS”。
- 用户可通过STATEMENT_ID对查询设置标签，输入的标签信息也将存储于PLAN_TABLE中。

版权所有 © 华为云计算技术有限公司

- 执行EXPLAIN PLAN 后会将计划信息自动存储于PLAN_TABLE中，不支持对PLAN_TABLE进行INSERT、UPDATE、ANALYZE等操作。PLAN_TABLE详细介绍见**PLAN_TABLE**。

语法格式

```
EXPLAIN PLAN [ SET STATEMENT_ID = name ] FOR statement ;
```



参数说明

- **name**

  查询标签。

  取值范围：字符串

  说明

  用户在执行EXPLAIN PLAN时，如果没有设置STATEMENT_ID，则默认为空值。同时，用户可输入的STATEMENT_ID最大长度为30个字节，超过长度将会报错。

- **statement**

  指定要分析的SQL语句。

示例 1

```
--创建表foo1,foo2。
gaussdb=# CREATE TABLE foo1(f1 int, f2 text, f3 text[]);
gaussdb=# CREATE TABLE foo2(f1 int, f2 text, f3 text[]);

--执行EXPLAIN PLAN。
gaussdb=# EXPLAIN PLAN SET STATEMENT_ID = 'TPCH-Q4' FOR SELECT f1, count(*) FROM foo1 WHERE f1
> 1 AND f1 < 3 AND EXISTS (SELECT * FROM foo2) GROUP BY f1;

--查询PLAN_TABLE。
gaussdb=# SELECT * FROM plan_table;

--清理PLAN_TABLE表中的数据，删除表foo1，foo2。
gaussdb=# DELETE FROM plan_table WHERE STATEMENT_ID = 'TPCH-Q4';
gaussdb=# DROP TABLE foo1;
gaussdb=# DROP TABLE foo2;
```

示例 2

```
--关闭enable_stream_recursive，使得with recursive关联查询不能下推。
gaussdb=# SET enable_stream_recursive = off;

--创建表chinamap。
gaussdb=# CREATE TABLE chinamap
(
  id integer,
  pid integer,
  name text
) DISTRIBUTE BY hash(id);

--plan_table收集的计划。
gaussdb=# EXPLAIN PLAN SET statement_id = 'cte can not be push down'
```


版权所有 © 华为云计算技术有限公司

```
FOR
WITH RECURSIVE rq AS
(
  SELECT id, name FROM chinamap WHERE id = 11
  UNION ALL
  SELECT origin.id, rq.name || ' > ' || origin.name
  FROM rq JOIN chinamap origin ON origin.pid = rq.id
)
SELECT id, name FROM rq ORDER BY 1;

--查询PLAN_TABLE。
gaussdb=# SELECT * FROM plan_table;

--删除表chinamap。
gaussdb=# DROP TABLE chinamap;
```

1.13.11.7 EXPLAIN AUTOHINT

功能描述

EXPLAIN AUTOHINT用于在正常执行EXPLAIN命令的同时将查询执行的计划信息与执行结果存储于AUTOHINT内存上下文中，以便进行AUTOHINT特性的HINT探索与推荐。

注意事项

- 该语句只允许在AUTOHINT任务进行时由内核进行自动调用，用户自行调用将无任何效果或会触发AUTOHINT的报错阻拦。
- AUTOHINT内存上下文中存储的数据是session级生命周期并且session隔离和用户隔离，当前session下调用的AUTOHINT特性功能用户只能访问并调用当前session、当前用户的数据。

语法格式

```
EXPLAIN AUTOHINT cache_id exec_type argument
FOR statement ;
```

参数说明

- **statement**

  当前AUTOHINT任务需要进行的计划与执行结果存储的SQL语句。（仅支持SELECT语句）

- **cache_id**

  当前AUTOHINT任务的任务ID，int型。

- **exec_type**

  当前执行该语句所存储的结果类型。

  取值范围：

  - PLAN：存储计划信息。
  - EXECUTE：存储执行结果信息。

- **argument**

  当前执行该语句所需的执行参数，json格式文本。

  若exec_type参数为PLAN类型，则为当前语句所应用的HINT。

  若exec_type参数为EXECUTE类型，则为当前语句对应执行计划的HASH值。

 版权所有 © 华为云计算技术有限公司

示例 1

```
--AUTOHINT执行语句为对应SQL存储计划信息。
gaussdb=# EXPLAIN AUTOHINT 2 PLAN
          '{"hints":[{"type": 0, "name": "enable_seqscan", "value": "true"},
           {"type": 0, "name": "enable_mergejoin", "value": "false"}]}' FOR
              SELECT * FROM t1, t2 WHERE t1.id = t2.id and t1.a > 100 and t2.b < 200;
--用户直接执行无效。
 QUERY PLAN
------------
(0 rows)
--用户直接执行被报错阻拦。
gaussdb=# ERROR:  Can not get exploration cache for cache id: 2.
```

示例 2

```
--AUTOHINT执行语句为对应SQL存储执行信息。
gaussdb=# EXPLAIN AUTOHINT 2 EXECUTE '{"topology": 740798663, "cards": [18936500, 10592, 9308,
9308]}' FOR
              SELECT * FROM t1, t2 WHERE t1.id = t2.id and t1.a > 100 and t2.b < 200;
--用户直接执行无效。
 QUERY PLAN
------------
(0 rows)
--用户直接执行被报错阻拦。
gaussdb=# ERROR:  Can not get exploration cache for cache id: 2.
```

1.13.12 F

1.13.12.1 FETCH

功能描述

FETCH通过已创建的游标来检索数据。

每个游标都有一个供FETCH使用的关联位置。游标的关联位置可以在查询结果的第一行之前，或者在结果中的任意行，或者在结果的最后一行之后：

- 游标刚创建完之后，关联位置在第一行之前。
- 在抓取了一些移动行之后，关联位置在检索到的最后一行上。
- 如果FETCH抓取完了所有可用行，它就停在最后一行后面，或者在反向抓取的情况下是停在第一行前面。
- FETCH ALL或FETCH BACKWARD ALL总是把游标的关联位置放在最后一行或者在第一行前面。

注意事项

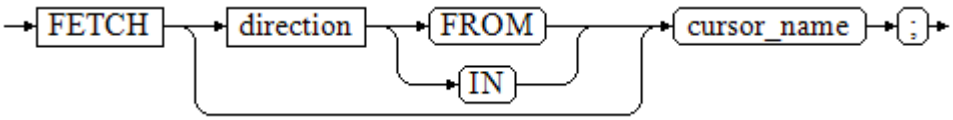
- 如果游标定义了NO SCROLL，则不允许使用例如FETCH BACKWARD之类的反向抓取。
- NEXT、PRIOR、FIRST、LAST和ABSOLUTE和RELATIVE形式在恰当地移动游标之后抓取一条记录。如果后面没有数据行，就返回一个空的结果，此时游标就会停在查询结果的最后一行之后（向后查询时）或者第一行之前（向前查询时）。
- FORWARD和BACKWARD形式在向前或者向后移动的过程中抓取指定的行数，然后把游标定位在最后返回的行上；如果count大于可用的行数，则会把游标定位在所有行之后（向后查询时）或者之前（向前查询时）。

 版权所有 © 华为云计算技术有限公司

- RELATIVE 0、FORWARD 0和BACKWARD 0都要求在不移动游标的前提下抓取当前行，也就是重新抓取最近刚抓取过的行。除非游标定位在第一行之前或者最后一行之后，否则这个动作都应该成功。而当游标定位在第一行之前或者最后一行之后，不返回任何行。
- 当FETCH的游标上涉及非系统表时，不支持BACKWARD、PRIOR、FIRST等涉及反向获取操作。

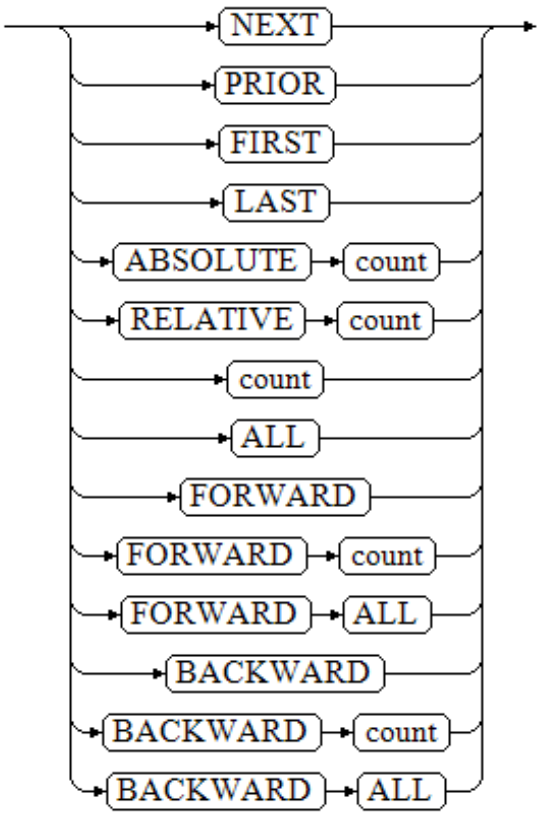
语法格式

```
FETCH [ direction { FROM | IN } ] cursor_name;
```



其中direction子句为可选参数。

```
{ NEXT
  | PRIOR
  | FIRST
  | LAST
  | ABSOLUTE count
  | RELATIVE count
  | count
  | ALL
  | FORWARD
  | FORWARD count
  | FORWARD ALL
  | BACKWARD
  | BACKWARD count
  | BACKWARD ALL }
```



版权所有 © 华为云计算技术有限公司

参数说明

- **direction**

  定义抓取数据的方向。

  取值范围：

  - NEXT（缺省值）

    从当前关联位置开始，抓取下一行。
  - PRIOR

    从当前关联位置开始，抓取上一行。
  - FIRST

    抓取查询的第一行（和ABSOLUTE 1相同）。
  - LAST

    抓取查询的最后一行（和ABSOLUTE -1相同）。
  - ABSOLUTE count

    抓取查询中第count行。

    ABSOLUTE抓取不会比用相对位移移动到需要的数据行更快，因为下层的实现必须遍历所有中间的行。

    count取值范围：有符号的整数

    - count为正数，就从查询结果的第一行开始，抓取第count行。当count小于当前游标位置时，涉及到rewind操作，暂不支持。
    - count为负数或0，涉及到反向扫描操作，暂不支持。
  - RELATIVE count

    从当前关联位置开始，抓取随后或前面的第count行。

    取值范围：有符号的整数

    - count为正数就抓取当前关联位置之后的第count行。
    - count为负数或0，涉及到反向扫描操作，暂不支持。
    - 如果当前行没有数据的话，RELATIVE 0返回空。
  - count

    抓取随后的count行（和FORWARD count一样）。
  - ALL

    从当前关联位置开始，抓取所有剩余的行（和FORWARD ALL一样）。
  - FORWARD

    抓取下一行（和NEXT一样）。
  - FORWARD count

    与RELATIVE count的效果相同，从当前关联位置开始，抓取随后或前面的第count行。
  - FORWARD ALL

    从当前关联位置开始，抓取所有剩余行。
  - BACKWARD

    从当前关联位置开始，抓取前面一行(和PRIOR一样) 。

 版权所有 © 华为云计算技术有限公司

  - BACKWARD count
    从当前关联位置开始，抓取前面的count行（向后扫描）。
    取值范围：有符号的整数
    - count为正数就抓取当前关联位置之前的第count行。
    - count为负数就抓取当前关联位置之后的第abs（count）行。
    - 如果有数据的话，BACKWARD 0重新抓取当前行。
  - BACKWARD ALL
    从当前关联位置开始，抓取所有前面的行（向后扫描）。
- **{ FROM | IN } cursor_name**
  使用关键字FROM或IN指定游标名称。
  取值范围：已创建的游标的名称。

示例

```
--创建表test，并向表中插入20条数据。
gaussdb=# CREATE TABLE test(c1 int, c2 int);
gaussdb=# INSERT INTO test VALUES (generate_series(1,20),generate_series(1,20));

--开始一个事务，建立一个名为cursor1的游标。
gaussdb=# START TRANSACTION;
gaussdb=# CURSOR cursor1 FOR SELECT * FROM test ORDER BY 1;

--指定游标从关联位置开始检索3行数据。
gaussdb=# FETCH FORWARD 3 FROM cursor1;
 c1 | c2
----+----
  1 |  1
  2 |  2
  3 |  3
(3 rows)

--关闭游标并提交事务。
gaussdb=# CLOSE cursor1;
gaussdb=# END;

--删除表。
gaussdb=# DROP TABLE test;
```

相关链接

CLOSE，**MOVE**，**CURSOR**

1.13.13 G

1.13.13.1 GENERATED UPDATE SYSTEM OBJECT

功能描述

生成系统对象的升级回滚脚本文件，在升级回滚时由OM调用。内部语法，不建议用户使用。

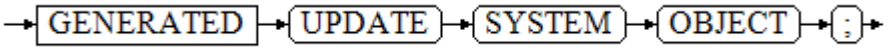
版权所有 © 华为云计算技术有限公司

注意事项

- 此语句仅支持升级中通过OM调用，即在GUC参数upgrade_mode不为0且GUC参数application_name设置为"OM"时此语句可被执行。
- 只有初始用户有权限执行此语句。

语法格式

```
GENERATED UPDATE SYSTEM OBJECT;
```



1.13.13.2 GRANT

功能描述

对角色和用户进行授权操作。

使用GRANT命令进行用户授权包括以下场景：

- **将系统权限授权给角色或用户**

  系统权限又称为用户属性，包括SYSADMIN、CREATEDB、CREATEROLE、AUDITADMIN、MONADMIN、OPRADMIN、POLADMIN、INHERIT、REPLICATION和LOGIN等。

  系统权限一般通过CREATE/ALTER ROLE语法来指定。其中，SYSADMIN权限可以通过GRANT/REVOKE ALL PRIVILEGE授予或撤销。但系统权限无法通过ROLE和USER的权限被继承，也无法授予PUBLIC。

- **将数据库对象授权给角色或用户**

  将数据库对象（表、视图、指定字段、数据库、函数、模式、表空间等）的相关权限授予特定角色或用户。

  GRANT命令将数据库对象的特定权限授予一个或多个角色，这些权限会追加到已有的权限上。

  关键字PUBLIC表示该权限要赋予所有角色，包括以后创建的用户。PUBLIC可以看做是一个隐含定义好的组，它总是包括所有角色。任何角色或用户都将拥有通过GRANT直接赋予的权限和所属的权限，再加上PUBLIC的权限。

  如果声明了WITH GRANT OPTION，则被授权的用户也可以将此权限赋予他人，否则就不能授权给他人。这个选项不能赋予PUBLIC，这是GaussDB特有的属性。

  GaussDB会将某些类型的对象上的权限授予PUBLIC。默认情况下，对表、表字段、序列、外部数据源、外部服务器、模式或表空间对象的权限不会授予PUBLIC，而以下这些对象的权限会授予PUBLIC：数据库的CONNECT权限和CREATE TEMP TABLE权限、函数的EXECUTE特权、语言和数据类型的USAGE特权。对象拥有者可以撤销默认授予PUBLIC的权限并专门授予权限给其他用户。为了更安全，建议在同一个事务中创建对象并设置权限，这样其他用户就没有时间窗口使用该对象。这些初始的默认权限可以使用ALTER DEFAULT PRIVILEGES命令修改。

  对象的所有者缺省具有该对象上的所有权限，出于安全考虑所有者可以舍弃部分权限，但ALTER、DROP、COMMENT、INDEX、VACUUM以及对象的可再授予权限属于所有者固有的权限，隐式拥有。

- **将角色或用户的权限授权给其他角色或用户**

版权所有 © 华为云计算技术有限公司

将一个角色或用户的权限授予一个或多个其他角色或用户。在这种情况下，每个角色或用户都可视为拥有一个或多个数据库权限的集合。

当声明了WITH ADMIN OPTION，被授权的用户可以将该权限再次授予其他角色或用户，以及撤销所有由该角色或用户继承到的权限。当授权的角色或用户发生变更或被撤销时，所有继承该角色或用户权限的用户拥有的权限都会随之发生变更。

三权分立关闭时，系统管理员可以赋予或者撤销任何非永久用户和运维管理员角色的权限。安全管理员可以赋予或者撤销任何非系统管理员、内置角色、永久用户和运维管理员角色的权限。

- **将ANY权限授予给角色或用户**

  将ANY权限授予特定的角色和用户，ANY权限的取值范围参见语法格式。当声明了WITH ADMIN OPTION，被授权的用户可以将该ANY权限再次授予其他角色/用户，或从其他角色/用户处回收该ANY权限。ANY权限可以通过角色被继承，但不能赋予PUBLIC。初始用户和三权分立关闭时的系统管理员用户可以给任何角色/用户授予或撤销ANY权限。

  目前支持以下ANY权限：CREATE ANY TABLE、ALTER ANY TABLE、DROP ANY TABLE、SELECT ANY TABLE、INSERT ANY TABLE、UPDATE ANY TABLE、DELETE ANY TABLE、TRUNCATE ANY TABLE、CREATE ANY SEQUENCE、CREATE ANY INDEX、CREATE ANY FUNCTION、EXECUTE ANY FUNCTION、CREATE ANY PACKAGE、EXECUTE ANY PACKAGE、CREATE ANY TYPE、ALTER ANY TYPE、DROP ANY TYPE、ALTER ANY SEQUENCE、DROP ANY SEQUENCE、SELECT ANY SEQUENCE、ALTER ANY INDEX、DROP ANY INDEX、CREATE ANY SYNONYM、DROP ANY SYNONYM、CREATE ANY TRIGGER、ALTER ANY TRIGGER、DROP ANY TRIGGER。详细的ANY权限范围描述请参见**表1-402**。

注意事项

- 不允许将ANY权限授予PUBLIC，也不允许从PUBLIC回收ANY权限。
- ANY权限属于数据库内的权限，只对授予该权限的数据库内的对象有效，例如SELECT ANY TABLE只允许用户查看当前数据库内的所有用户表数据，对其他数据库内的用户表无查看权限。
- ANY权限与原有的权限相互无影响。
- 如果用户被授予CREATE ANY TABLE权限，在同名模式下创建表的所有者是该同名模式的所有者，用户对表进行其他操作时，需要授予相应的操作权限。与此类似的还有CREATE ANY FUNCTION、CREATE ANY PACKAGE、CREATE ANY TYPE、CREATE ANY SEQUENCE和CREATE ANY INDEX，在同名模式下创建的对象的所有者是同名模式的所有者；而对于CREATE ANY TRIGGER和CREATE ANY SYNONYM，在同名模式下创建的对象的所有者为创建者。
- 需要谨慎授予用户CREATE ANY FUNCTION或CREATE ANY PACKAGE的权限，以免其他用户利用DEFINER类型的函数或PACKAGE进行权限提升。
- 通过GRANT授予用户使用表的权限时，如果用户使用不当，可能会通过ALTER语法在表的默认值、约束增加表达式、创建索引在索引上增加表达式等操作导致权限被利用的风险。
- 通过GRANT授予用户使用TRIGGER的权限时，如果用户使用不当，可能通过WHEN条件创建表达式，当触发器被触发时，存在权限被利用的风险。
- 给用户赋权时，需要特别注意定义者函数/PACKAGE，定义者函数/PACKAGE会使用函数/PACKAGE的owner权限执行，若赋权不当（包括GRANT ROLE TO ROLE），则存在权限被利用风险。

版权所有 © 华为云计算技术有限公司

- 不要把对象的权限赋予给过多的用户，可以根据业务需求使用角色或PUBLIC。

语法格式

- 将表或视图的访问权限赋予指定的用户或角色。

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER | ALTER | DROP
| COMMENT | INDEX | VACUUM } [, ...]
    | ALL [ PRIVILEGES ] }
  ON { [ TABLE ] table_name [, ...]
    | ALL TABLES IN SCHEMA schema_name [, ...] }
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将表中字段的访问权限赋予指定的用户或角色。

```
GRANT { {{ SELECT | INSERT | UPDATE | REFERENCES | COMMENT } ( column_name [, ...] )} [, ...]
    | ALL [ PRIVILEGES ] ( column_name [, ...] ) }
  ON [ TABLE ] table_name [, ...]
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

  说明

  如果拥有表的访问权限，则默认拥有表中所有字段的访问权限。如果要仅赋予表中某个字段的访问权限，需要先撤销所属表的访问权限。

- 将序列的访问权限赋予指定的用户或角色。

```
GRANT { { SELECT | UPDATE | USAGE | ALTER | DROP | COMMENT } [, ...]
    | ALL [ PRIVILEGES ] }
  ON { [ SEQUENCE ] sequence_name [, ...]
    | ALL SEQUENCES IN SCHEMA schema_name [, ...] }
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将数据库的访问权限赋予指定的用户或角色。

```
GRANT { { CREATE | CONNECT | TEMPORARY | TEMP | ALTER | DROP | COMMENT } [, ...]
    | ALL [ PRIVILEGES ] }
  ON DATABASE database_name [, ...]
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将客户端加密主密钥CMK的访问权限赋予指定的用户或角色。

```
GRANT { { USAGE | DROP } [, ...] | ALL [ PRIVILEGES ] }
  ON CLIENT_MASTER_KEY client_master_key
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将列加密密钥CEK的访问权限赋予指定的用户或角色。

```
GRANT { { USAGE | DROP } [, ...] | ALL [ PRIVILEGES ] }
  ON COLUMN_ENCRYPTION_KEY column_encryption_key
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将外部数据源的访问权限赋予给指定的用户或角色。

```
GRANT { USAGE | ALL [ PRIVILEGES ] }
  ON FOREIGN DATA WRAPPER fdw_name [, ...]
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将外部服务器的访问权限赋予给指定的用户或角色。

```
GRANT { { USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
  ON FOREIGN SERVER server_name [, ...]
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```

- 将函数的访问权限赋予给指定的用户或角色。

```
GRANT { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
  ON { FUNCTION {function_name ( [ {[ argmode ] [ arg_name ] arg_type} [, ...] ] )} [, ...]
    | ALL FUNCTIONS IN SCHEMA schema_name [, ...] }
  TO { [ GROUP ] role_name | PUBLIC } [, ...]
  [ WITH GRANT OPTION ];
```


版权所有 © 华为云计算技术有限公司

- 将存储过程的访问权限赋予给指定的用户或角色。

```
GRANT { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON PROCEDURE {proc_name ( [ {[ argmode ] [ arg_name ] arg_type} [, ...] ] )} [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ];
```

- 将过程语言的访问权限赋予给指定的用户或角色。

```
GRANT { USAGE | ALL [ PRIVILEGES ] }
    ON LANGUAGE lang_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ];
```

说明

本版本只有C函数支持所有用户创建，而Java和Internal只支持拥有sysadmin权限的用户创建。

- 拥有sysadmin权限的用户通过grant语法来将创建C函数的权限授权给普通用户时，不支持GRANT USAGE ON LANGUAGE c TO PUBLIC，只支持授权给特定用户。
- 拥有sysadmin权限的用户通过grant语法来将创建C函数的权限授权给普通用户时，不支持WITH GRANT OPTION。

- 将子集群的访问权限赋予指定的用户或角色。

```
GRANT { { CREATE | USAGE | COMPUTE | ALTER | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON NODE GROUP group_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ];
```

说明

将子集群的create权限赋予指定用户或角色时，会默认把usage和compute权限赋予指定用户或角色。

- 将模式的访问权限赋予指定的用户或角色。

```
GRANT { { CREATE | USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ];
```

说明

将模式中的表或者视图对象授权给其他用户时，需要将表或视图所属的模式的USAGE权限同时授予该用户，若没有该权限，则只能看到这些对象的名称，并不能实际进行对象访问。

- 将表空间的访问权限赋予指定的用户或角色。

```
GRANT { { CREATE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON TABLESPACE tablespace_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ];
```

- 将类型的访问权限赋予指定的用户或角色。

```
GRANT { { USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON TYPE type_name [, ...]
    TO { [ GROUP ] role_name | PUBLIC } [, ...]
    [ WITH GRANT OPTION ];
```

说明

本版本暂时不支持赋予类型的访问权限。

- 将DIRECTORY对象的权限赋予指定的角色。

```
GRANT { { READ | WRITE | ALTER | DROP } [, ...] | ALL [PRIVILEGES] }
  ON DIRECTORY directory_name [, ...]
  TO { [GROUP] role_name | PUBLIC } [, ...]
  [WITH GRANT OPTION];
```

- 将PACKAGE对象的权限赋予指定的角色。

版权所有 © 华为云计算技术有限公司

```
GRANT { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [PRIVILEGES] }
  ON PACKAGE package_name [, ...]
  TO { [GROUP] role_name | PUBLIC } [, ...]
  [WITH GRANT OPTION];
```

- 将角色的权限赋予其他用户或角色的语法。

```
GRANT role_name [, ...]
  TO role_name [, ...]
  [ WITH ADMIN OPTION ];
```

说明

通过GRANT role1 to role2的方式将角色role1授权给role2时，role2不会继承role1的系统权限，但由于role2拥有了role1的所有对象权限，role1的系统权限存在被利用的风险。谨慎将拥有系统权限的角色授权其他用户或角色。

- 将SYSADMIN权限赋予指定的角色。

```
GRANT ALL { PRIVILEGES | PRIVILEGE }
  TO role_name;
```

- 将ANY权限赋予其他用户或角色的语法。

```
GRANT { CREATE ANY TABLE | ALTER ANY TABLE | DROP ANY TABLE | SELECT ANY TABLE |
INSERT ANY TABLE | UPDATE ANY TABLE | DELETE ANY TABLE | TRUNCATE ANY TABLE | CREATE ANY
SEQUENCE |
CREATE ANY INDEX | CREATE ANY FUNCTION | EXECUTE ANY FUNCTION | CREATE ANY PACKAGE |
EXECUTE ANY PACKAGE | CREATE ANY TYPE | ALTER ANY TYPE | DROP ANY TYPE |
ALTER ANY SEQUENCE | DROP ANY SEQUENCE | SELECT ANY SEQUENCE | ALTER ANY INDEX |
DROP ANY INDEX | CREATE ANY SYNONYM | DROP ANY SYNONYM | CREATE ANY TRIGGER | ALTER
ANY TRIGGER |
  DROP ANY TRIGGER } [, ...]
  TO [ GROUP ] role_name [, ...]
  [ WITH ADMIN OPTION ];
```

- 将DATABASE LINK对象权限赋予指定用户。

```
GRANT { CREATE | ALTER | DROP } [PUBLIC] DATABASE LINK TO role_name;
```

说明

- PUBLIC：指定公共以创建对所有用户可见的公共数据库链接。如果省略此子句，则数据库链接是私有的，只作为兼容接口使用，无实际意义。远程数据库上可访问的数据取决于数据库链接在连接到远程数据库时使用的标识。
- 当赋予用户创建DATABASE LINK对象权限时，相当于许可用户使用服务端DATABASE的IP对远端进行访问。若不希望有此效果，应不要使用GRANT对用户赋权。
- 现在除DATABASE LINK的直接赋权语句外，还可以通过权限继承和赋予管理员用户获取到DATABASE LINK的相关权限。
- DATABASE LINK详细说明请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

- 将创建PUBLIC同义词的权限赋予指定用户。

```
GRANT { CREATE | DROP } PUBLIC SYNONYM TO role_name [ WITH ADMIN OPTION ];
```

通过内置角色（gs_role_public_synonym_create、gs_role_public_synonym_drop）的方式也可以实现赋予用户创建、删除PUBLIC同义词的权限。

  - 赋予用户创建PUBLIC同义词的权限：

    ```
    GRANT gs_role_public_synonym_create TO role_name;
    ```

  - 赋予用户删除PUBLIC同义词的权限：

    ```
    GRANT gs_role_public_synonym_drop TO role_name;
    ```

参数说明

GRANT的权限分类如下所示。

版权所有 © 华为云计算技术有限公司

- **SELECT**

  允许对指定的表、视图、序列执行SELECT语句。UPDATE或DELETE时也需要对应字段上的SELECT权限。

- **INSERT**

  允许对指定的表执行INSERT语句。

- **UPDATE**

  允许对声明的表中任意字段执行UPDATE语句，UPDATE命令也需要SELECT权限来查询出哪些行需要更新。SELECT… FOR UPDATE和SELECT… FOR SHARE除了需要SELECT权限外，还需要UPDATE权限。

- **DELETE**

  允许执行DELETE语句删除指定表中的数据。DELETE命令也需要SELECT权限来查询出哪些行需要删除。

- **TRUNCATE**

  允许执行TRUNCATE语句删除指定表中的所有记录。

- **REFERENCES**

  创建一个外键约束，必须拥有参考表和被参考表的REFERENCES权限，分布式场景暂不支持。

- **TRIGGER**

  允许在指定的表上创建触发器。

- **CREATE**
  - 对于数据库，允许在数据库里创建新的模式。
  - 对于模式，允许在模式中创建新的对象。如果要重命名一个对象，用户除了必须是该对象的所有者外，还必须拥有该对象所在模式的CREATE权限。
  - 对于表空间，允许在表空间中创建表，允许在创建数据库和模式的时候把该表空间指定为缺省表空间。
  - 对于子集群，允许在子集群中创建表对象。

- **CONNECT**

  允许用户连接到指定的数据库。

- **TEMPORARY**

  允许用户在数据库中创建临时表。

- **READ**

  允许用户读指定目录。

- **WRITE**

  允许用户写指定目录。

- **EXECUTE**

  允许使用指定的函数，以及利用这些函数实现的操作符。

- **USAGE**
  - 对于过程语言，允许用户在创建函数的时候指定过程语言。
  - 对于模式，USAGE允许访问包含在指定模式中的对象，若没有该权限，则只能看到这些对象的名称。
  - 对于序列，USAGE允许使用nextval函数。
  - 对于子集群， 对包含在指定模式中的对象有访问权限时，USAGE允许访问指定子集群下的表对象。

版权所有 © 华为云计算技术有限公司

  - 对于密钥对象，USAGE是使用密钥的权限。

- **COMPUTE**

  针对计算子集群，允许用户在具有compute权限的计算子集群上进行弹性计算。

- **ALTER**

  允许用户修改指定对象的属性，但不包括修改对象的所有者和修改对象所在的模式。

- **DROP**

  允许用户删除指定的对象。

- **COMMENT**

  允许用户定义或修改指定对象的注释。

- **INDEX**

  允许用户在指定表上创建索引，并管理指定表上的索引，还允许用户对指定表执行REINDEX和CLUSTER操作。

- **VACUUM**

  允许用户对指定的表执行ANALYZE和VACUUM操作。

- **ALL PRIVILEGES**

  一次性给指定用户/角色赋予所有可赋予的权限。只有系统管理员有权执行GRANT ALL PRIVILEGES。

GRANT的参数说明如下所示。

- **role_name**

  已存在用户名称。

- **table_name**

  已存在表名称。

- **column_name**

  已存在字段名称。

- **schema_name**

  已存在模式名称。

- **database_name**

  已存在数据库名称。

- **function_name**

  已存在函数名称。

- **sequence_name**

  已存在序列名称。

- **fdw_name**

  已存在外部数据包名称。

- **lang_name**

  已存在语言名称。

- **type_name**

  已存在类型名称。

- **group_name**

版权所有 © 华为云计算技术有限公司

已存在的子集群名称。

- **argmode**

  参数模式。

  取值范围：字符串，要符合**标识符命名规范**。

- **arg_name**

  参数名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **arg_type**

  参数类型。

  取值范围：字符串，要符合**标识符命名规范**。

- **tablespace_name**

  表空间名称。

- **client_master_key**

  客户端加密主密钥的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **column_encryption_key**

  列加密密钥的名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **directory_name**

  目录名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **WITH GRANT OPTION**

  如果声明了WITH GRANT OPTION，则被授权的用户也可以将此权限赋予他人，否则就不能授权给他人。这个选项不能赋予PUBLIC。

非对象所有者给其他用户授予对象权限时，命令按照以下规则执行：

- 如果用户没有该对象上指定的权限，命令立即失败。
- 如果用户有该对象上的部分权限，则GRANT命令只授予他有授权选项的权限。
- 如果用户没有可用的授权选项，GRANT ALL PRIVILEGES形式将发出一个警告信息，其他命令形式将发出在命令中提到的且没有授权选项的相关警告信息。

说明

三权分立关闭时的数据库系统管理员可以访问所有对象，而不会受对象的权限设置影响。除了必要的情况外，建议不要总是以系统管理员身份进行操作。

- **WITH ADMIN OPTION**

  对于角色，当声明了WITH ADMIN OPTION，被授权的用户可以将该角色再授予其他角色/用户，或从其他角色/用户回收该角色。

  对于ANY权限，当声明了WITH ADMIN OPTION，被授权的用户可以将该ANY权限再授予其他角色/用户，或从其他角色/用户回收该ANY权限。

版权所有 © 华为云计算技术有限公司

表 1-402 ANY 权限列表

ANY权限名称	描述
CREATE ANY TABLE	用户能够在public模式和用户模式下创建表或视图。如果想要创建serial类型列的表，还需要授予创建序列的权限。
ALTER ANY TABLE	用户拥有对public模式和用户模式下表或视图的ALTER权限。如果想要修改表的唯一索引为表增加主键约束或唯一约束，还需要授予该表的索引权限。
DROP ANY TABLE	用户拥有对public模式和用户模式下表或视图的DROP权限。
SELECT ANY TABLE	用户拥有对public模式和用户模式下表或视图的SELECT权限，仍然受行级访问控制限制。
UPDATE ANY TABLE	用户拥有对public模式和用户模式下表或视图的UPDATE权限，仍然受行级访问控制限制。
INSERT ANY TABLE	用户拥有对public模式和用户模式下表或视图的INSERT权限。
DELETE ANY TABLE	用户拥有对public模式和用户模式下表或视图的DELETE权限，仍然受行级访问控制限制。
TRUNCATE ANY TABLE	用户拥有对public模式和用户模式下表或视图的TRUNCATE权限。
CREATE ANY FUNCTION	用户能够在用户模式下创建函数或存储过程。
EXECUTE ANY FUNCTION	用户拥有在public模式和用户模式下函数或存储过程的EXECUTE权限。
CREATE ANY PACKAGE	用户能够在public模式和用户模式下创建PACKAGE。
EXECUTE ANY PACKAGE	用户拥有在public模式和用户模式下PACKAGE的EXECUTE权限。
CREATE ANY TYPE	用户能够在public模式和用户模式下创建类型。
CREATE ANY SEQUENCE	用户能够在public模式和用户模式下创建序列。
CREATE ANY INDEX	用户能够在public模式和用户模式下创建索引。如果在某表空间创建分区表索引，需要授予用户该表空间的创建权限。
ALTER ANY TYPE	用户拥有对public模式和用户模式下类型的ALTER权限，但不包括修改类型的所有者或者修改类型的模式。
DROP ANY TYPE	用户拥有对public模式和用户模式下类型的DROP权限。
ALTER ANY SEQUENCE	用户拥有对public模式和用户模式下序列的ALTER权限，但不包括修改序列的所有者。

版权所有 © 华为云计算技术有限公司

ANY权限名称	描述
DROP ANY SEQUENCE	用户拥有对public模式和用户模式下序列的DROP权限。
SELECT ANY SEQUENCE	用户拥有对public模式和用户模式下序列的SELECT、USAGE和UPDATE权限。
ALTER ANY INDEX	用户拥有对public模式和用户模式下索引的ALTER权限。如果要重命名索引，还需要索引所在模式下创建对象的权限。如果涉及表空间的操作，还需要对应表空间的相应操作权限。如果设置索引不可用（UNUSABLE），还需要DROP ANY INDEX权限。
DROP ANY INDEX	用户拥有对public模式和用户模式下索引的DROP权限。
CREATE ANY TRIGGER	用户能够在public模式和用户模式下创建触发器。
ALTER ANY TRIGGER	用户拥有对public模式和用户模式下触发器的ALTER权限。
DROP ANY TRIGGER	用户拥有对public模式和用户模式下触发器的DROP权限。
CREATE ANY SYNONYM	用户能够在用户模式下创建同义词。
DROP ANY SYNONYM	用户拥有对public模式和用户模式下同义词的DROP权限。

说明

用户被授予任何一种ANY权限后，用户对public模式和用户模式具有USAGE权限，对表8-1中除public之外的系统模式没有USAGE权限。

示例

示例1：将系统权限授权给用户或者角色。

创建名为joe的用户，并将sysadmin权限授权给他。

```
gaussdb=# CREATE USER joe PASSWORD 'xxxxxxxxxx';
gaussdb=# GRANT ALL PRIVILEGES TO joe;
```

授权成功后，用户joe会拥有sysadmin的所有权限。

示例2：将对象权限授权给用户或者角色。

1. 撤销joe用户的sysadmin权限，然后将模式tpcds的使用权限和表tpcds.reason的所有权限授权给用户joe。

```
gaussdb=# CREATE SCHEMA tpcds;
gaussdb=# CREATE TABLE tpcds.reason (
r_reason_sk         INTEGER       NOT NULL,
r_reason_id         CHAR(16)      NOT NULL,
r_reason_desc       VARCHAR(20)
);
gaussdb=# REVOKE ALL PRIVILEGES FROM joe;
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# GRANT USAGE ON SCHEMA tpcds TO joe;
gaussdb=# GRANT ALL PRIVILEGES ON tpcds.reason TO joe;
```

授权成功后，joe用户就拥有了tpcds.reason表的所有权限，包括增删改查等权限。

2. 将tpcds.reason表中r_reason_sk、r_reason_id、r_reason_desc列的查询权限，r_reason_desc的更新权限授权给joe。

```
gaussdb=# GRANT select (r_reason_sk,r_reason_id,r_reason_desc),update (r_reason_desc) ON tpcds.reason TO joe;
```

授权成功后，用户joe对tpcds.reason表中r_reason_sk，r_reason_id, r_reason_desc的查询权限会立即生效。如果joe用户需要拥有将这些权限授权给其他用户的权限，可以通过以下语法对joe用户进行授权。

```
gaussdb=# GRANT select (r_reason_sk, r_reason_id) ON tpcds.reason TO joe WITH GRANT OPTION;
```

将数据库testdb的连接权限授权给用户joe，并给予其在testdb中创建schema的权限，而且允许joe将此权限授权给其他用户。

```
gaussdb=# CREATE DATABASE testdb;
gaussdb=# GRANT create,connect on database testdb TO joe WITH GRANT OPTION;
```

创建角色tpcds_manager，将模式tpcds的访问权限授权给角色tpcds_manager，并授予该角色在tpcds下创建对象的权限，不允许该角色中的用户将权限授权给其他人。

```
gaussdb=# CREATE ROLE tpcds_manager PASSWORD 'xxxxxxxxxx';
gaussdb=# GRANT USAGE,CREATE ON SCHEMA tpcds TO tpcds_manager;
```

将表空间tpcds_tbspc的所有权限授权给用户joe，但用户joe无法将权限继续授予其他用户。

```
gaussdb=# CREATE TABLESPACE tpcds_tbspc RELATIVE LOCATION 'tablespace/tablespace_1';
gaussdb=# GRANT ALL ON TABLESPACE tpcds_tbspc TO joe;
```

将创建tpcds.reason表的PUBLIC同义词权限赋予给joe，并通过内置角色（gs_role_public_synonym_create）的方式实现。

```
gaussdb=# CREATE OR REPLACE SYNONYM t1 FOR tpcds.reason;
gaussdb=# GRANT gs_role_public_synonym_create TO joe;
```

3. 在模式tpcds下创建函数fun1，并将函数fun1的ALTER权限赋予给joe。

```
gaussdb=# CREATE or replace FUNCTION tpcds.fun1() RETURN boolean AS
BEGIN
SELECT current_user;
RETURN true;
END;
/
gaussdb=# GRANT ALTER ON FUNCTION tpcds.fun1() TO joe;
```

示例3：将用户或者角色的权限授权给其他用户或角色。

1. 创建角色manager，将joe的权限授权给manager，并允许该角色将权限授权给其他人。

```
gaussdb=# CREATE ROLE manager PASSWORD 'xxxxxxxxxxx';
gaussdb=# GRANT joe TO manager WITH ADMIN OPTION;
```

2. 创建用户senior_manager，将用户manager的权限授权给该用户。

```
gaussdb=# CREATE ROLE senior_manager PASSWORD 'xxxxxxxxxxx';
gaussdb=# GRANT manager TO senior_manager;
```

3. 撤销权限，并清理用户。

```
gaussdb=# REVOKE joe FROM manager;
gaussdb=# REVOKE manager FROM senior_manager;
gaussdb=# DROP USER manager;
gaussdb=# DROP DATABASE testdb;
```

示例4：将CMK或者CEK的权限授权给其他用户或角色。

1. 连接密态数据库。

版权所有 © 华为云计算技术有限公司

```
--使用参数-C打开密态数据库开关
gsql -p 57101 postgres -r -C
gaussdb=#  CREATE CLIENT MASTER KEY MyCMK1 WITH ( KEY_STORE = xxx , KEY_PATH = xxx , ALGORITHM = AES_256_CBC);
CREATE CLIENT MASTER KEY
gaussdb=# CREATE COLUMN ENCRYPTION KEY MyCEK1 WITH VALUES (CLIENT_MASTER_KEY = MyCMK1, ALGORITHM = AEAD_AES_256_CBC_HMAC_SHA256);
CREATE COLUMN ENCRYPTION KEY
```

2. 创建角色newuser，将密钥的权限授权给newuser。

```
gaussdb=# CREATE USER newuser PASSWORD 'xxxxxxxxx';
CREATE ROLE
gaussdb=# GRANT ALL ON SCHEMA public TO newuser;
GRANT
gaussdb=# GRANT USAGE ON COLUMN_ENCRYPTION_KEY MyCEK1 to newuser;
GRANT
gaussdb=# GRANT USAGE ON CLIENT_MASTER_KEY MyCMK1 to newuser;
GRANT
```

3. 设置该用户连接数据库,使用该CEK创建加密表。

```
gaussdb=# SET ROLE newuser PASSWORD 'xxxxxxxxx';
gaussdb=>  CREATE TABLE acltest1 (x int, x2 varchar(50) ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = MyCEK1, ENCRYPTION_TYPE = DETERMINISTIC));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=> SELECT has_cek_privilege('newuser', 'MyCEK1', 'USAGE');
 has_cek_privilege
-------------------
 t
(1 row)
```

4. 撤销权限，并清理用户。

```
gaussdb=> RESET ROLE;
gaussdb=# REVOKE USAGE ON COLUMN_ENCRYPTION_KEY MyCEK1 FROM newuser;
gaussdb=# REVOKE USAGE ON CLIENT_MASTER_KEY MyCMK1 FROM newuser;
gaussdb=# DROP TABLE newuser.acltest1;
gaussdb=# DROP COLUMN ENCRYPTION KEY MyCEK1;
gaussdb=# DROP CLIENT MASTER KEY MyCMK1;
gaussdb=# DROP SCHEMA IF EXISTS newuser CASCADE;
gaussdb=# REVOKE ALL ON SCHEMA public FROM newuser;
gaussdb=# DROP ROLE IF EXISTS newuser;
```

示例5：撤销上述授予的权限，并清理角色和用户。

```
gaussdb=# REVOKE gs_role_public_synonym_create FROM joe;
gaussdb=# REVOKE ALTER ON FUNCTION tpcds.fun1() FROM joe;
gaussdb=# REVOKE ALL PRIVILEGES ON tpcds.reason FROM joe;
gaussdb=# REVOKE ALL PRIVILEGES ON SCHEMA tpcds FROM joe;
gaussdb=# REVOKE ALL ON TABLESPACE tpcds_tbspc FROM joe;
gaussdb=# DROP TABLESPACE tpcds_tbspc;
gaussdb=# REVOKE USAGE,CREATE ON SCHEMA tpcds FROM tpcds_manager;
gaussdb=# DROP ROLE tpcds_manager;
gaussdb=# DROP ROLE senior_manager;
gaussdb=# DROP USER joe CASCADE;
gaussdb=# DROP TABLE tpcds.reason;
gaussdb=# DROP SYNONYM t1;
gaussdb=# DROP FUNCTION tpcds.fun1();
gaussdb=# DROP SCHEMA tpcds CASCADE;
```

相关链接

REVOKE，**ALTER DEFAULT PRIVILEGES**

1.13.14 I

版权所有 © 华为云计算技术有限公司

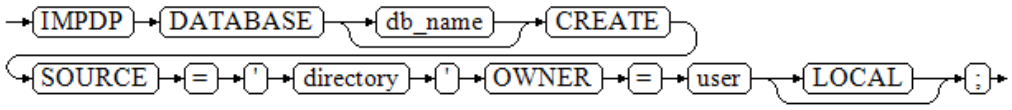
1.13.14.1 IMPDP DATABASE CREATE

功能描述

导入DATABASE的准备阶段。

语法格式

```
IMPDP DATABASE [db_name] CREATE SOURCE = 'directory' OWNER = user [LOCAL];
```



参数说明

- **db_name**
  导入后的新库名，如不指定则导入后保持原库名。
- **directory**
  导入的database的数据源目录。
- **user**
  导入后库的属主。
- **LOCAL**
  指定该字段表示导入到原集群，如不指定表示导入到新集群。

示例

```
--IMPDP DATABASE CREATE语法用于细粒度备份恢复，由备份恢复工具调用，用户直接调用可能会提示目录不存在等报错，不推荐用户直接调用该SQL。
gaussdb=# IMPDP DATABASE test CREATE SOURCE = '/data1/impdp/database' OWNER = admin;
```

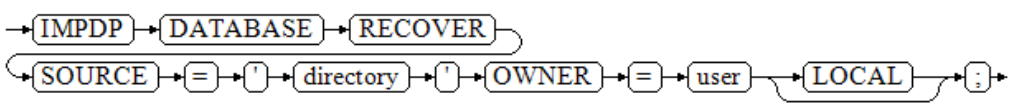
1.13.14.2 IMPDP RECOVER

功能描述

导入DATABASE的执行阶段。

语法格式

```
IMPDP DATABASE RECOVER SOURCE = 'directory' OWNER = user [LOCAL];
```



参数说明

- **directory**
  导入的database的数据源目录。
- **user**
  导入后库的属主。

版权所有 © 华为云计算技术有限公司

- **LOCAL**
  指定该字段表示导入到原集群，如不指定表示导入到新集群。

示例

```
--IMPDP DATABASE RECOVER语法用于细粒度备份恢复，由备份恢复工具调用，用户直接调用可能会提示目录不存在等报错，不推荐用户直接调用该SQL。
gaussdb=# IMPDP DATABASE RECOVER SOURCE = '/data1/impdp/database' OWNER=admin;
```

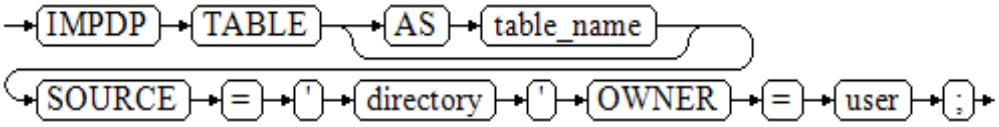
1.13.14.3 IMPDP TABLE

功能描述

导入表的执行阶段。

语法格式

```
IMPDP TABLE [AS table_name] SOURCE = 'directory' OWNER = user;
```



参数说明

- **table_name**
  导入后的新表名，如不指定则导入后保持原表名。
- **directory**
  导入的表的数据源目录。
- **user**
  导入后表的属主。

示例

```
--IMPDP TABLE语法用于细粒度备份恢复，由备份恢复工具调用，用户直接调用可能会提示目录不存在等报错，不推荐用户直接调用该SQL。
gaussdb=# IMPDP TABLE SOURCE = '/data1/impdp/table0' OWNER=admin;
```

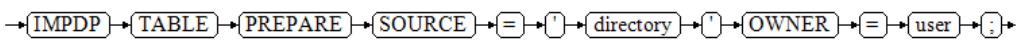
1.13.14.4 IMPDP TABLE PREPARE

功能描述

导入表的准备阶段。

语法格式

```
IMPDP TABLE PREPARE SOURCE = 'directory' OWNER = user;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **directory**
  导入的表的数据源目录。
- **user**
  导入后表的属主。

示例

```
--IMPDP TABLE PREPARE语法用于细粒度备份恢复，由备份恢复工具调用，用户直接调用可能会提示目录不存在等报错，不推荐用户直接调用该SQL。
gaussdb=# IMPDP TABLE PREPARE SOURCE = '/data1/impdp/table0' OWNER=admin;
```

1.13.14.5 INSERT

功能描述

向表中插入一行或多行数据。

注意事项

- 表的所有者、拥有表INSERT权限的用户或拥有INSERT ANY TABLE权限的用户，可向表中插入数据，三权分立关闭时的系统管理员默认拥有此权限。
- 如果使用RETURNING子句，用户必须要有该表的SELECT权限。
- 如果使用ON DUPLICATE KEY UPDATE，用户必须要有该表的INSERT、UPDATE权限，UPDATE子句中列的SELECT权限。
- 如果使用query子句插入来自查询里的数据行，用户还需要拥有在查询里使用的表的SELECT权限。
- 如果使用query子句插入来自查询动态数据脱敏列的数据，插入的结果即为脱敏后的值，无法被还原。
- 当连接到TD兼容的数据库时，td_compatible_truncation参数设置为on时，将启用超长字符串自动截断功能，在后续的INSERT语句中（不包含外表的场景下），对目标表中CHAR和VARCHAR类型的列上插入超长字符串时，系统会自动按照目标表中相应列定义的最大长度对超长字符串进行截断。

  说明

  如果向字符集为字节类型编码（SQL_ASCII，LATIN1等）的数据库中插入多字节字符数据（如汉字等），且字符数据跨越截断位置，这种情况下，按照字节长度自动截断，自动截断后会在尾部产生非预期结果。如果用户有对于截断结果正确性的要求，建议用户采用UTF8等能够按照字符截断的输入字符集作为数据库的编码集。

语法格式

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
INSERT [/*+ plan_hint */] [ IGNORE ] INTO table_name [ { [alias_name] [ ( column_name [, ...] ) ] } |
{ [partition_clause] [ AS alias ] [ ( column_name [, ...] ) ] } ]
   { DEFAULT VALUES
   | { VALUES | VALUE } {( { expression | DEFAULT } [, ...] ) }[, ...]
   | query }
   [ { ON DUPLICATE KEY UPDATE { NOTHING | { column_name = { expression | DEFAULT } } [, ...]
[ WHERE condition ] } }
   | { [ ON CONFLICT [ conflict_target ] conflict_action ] } ]
   [ RETURNING {* | {output_expression [ [ AS ] output_name ] }[, ...]} ];
```

对子查询和视图插入：

 版权所有 © 华为云计算技术有限公司

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
INSERT [/*+ plan_hint */] INTO {subquery | view_name} [ AS alias ] [ ( column_name [, ...] ) ]
   { DEFAULT VALUES
   | { VALUES | VALUE } {( { expression | DEFAULT } [, ...] ) }[, ...]
   | query }
   [ RETURNING {* | {output_expression [ [ AS ] output_name ] }[, ...]} ];
```

- subquery的具体信息为：

  ```
  [ WITH [ RECURSIVE ] with_query [, ...] ]
  SELECT [/*+ plan_hint */] [ ALL ]
  { * | {expression [ [ AS ] output_name ]} [, ...] }
  [ into_option ]
  [ FROM from_item [, ...] ]
  [ WHERE condition ]
  [ [ START WITH condition ] CONNECT BY [NOCYCLE] condition [ ORDER SIBLINGS BY expression ] ]
  [ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS
  { FIRST | LAST } ]} [, ...] ]
  [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]
  [ into_option ];
  ```

- 其中指定子查询源from_item为：

  ```
  {[ ONLY ] {table_name | view_name} [ * ] [ partition_clause ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  |( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
  |with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  |from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]}
  ```

- 其中with_query的具体信息为：

  ```
  with_query_name [ ( column_name [, ...] ) ] AS [ [ NOT ] MATERIALIZED ]
  ( {SELECT | VALUES | INSERT | UPDATE | DELETE} )
  ```

- partition_clause的具体信息为：

  ```
  PARTITION { ( partition_name ) | FOR ( partition_value [, ...] ) }
  ```

- conflict_target的具体信息为：

  ```
  ( { index_column_name | ( index_expression ) } [, ...] ) [ WHERE index_predicate ]
  ```

- conflict_action的具体信息为：

  ```
  DO NOTHING |
  DO UPDATE SET { column_name = { expression | DEFAULT } | ( column_name [, ...] ) = ( { expression |
  DEFAULT } [, ...] ) } [, ...] [ WHERE condition ]
  ```

参数说明

- **WITH [RECURSIVE] with_query [, ...]**

  用于声明一个或多个可以在主查询中通过名称引用的子查询，相当于临时表。

  如果声明了RECURSIVE，那么允许SELECT子查询通过名称引用。

  其中with_query中：

  – with_query_name指定子查询生成的结果集名称，在查询中可使用该名称访问子查询的结果集。

  – column_name指定子查询结果集中显示的列名。

  – 每个子查询可以是SELECT，VALUES，INSERT，UPDATE或DELETE语句。

  须知

  当子查询为附带RETURNING语句的DML时，INSERT语句的回显记录数由子查询决定。假设存在表T1，执行以下语句：

  ```
  WITH CTE AS (INSERT INTO T1 VALUES(1,2) RETURNING *) INSERT INTO T1 SELECT * FROM CTE;
  ```

  上述语句的回显记录数由以下部分决定，而不是由整条语句决定，即回显为“INSERT 0 1”，而不是“INSERT 0 2”。

  ```
  INSERT INTO T1 VALUES(1,2) RETURNING *
  ```


版权所有 © 华为云计算技术有限公司

– 用户可以使用MATERIALIZED / NOT MATERIALIZED对CTE进行修饰。
  - 如果声明为MATERIALIZED，WITH查询将被物化，生成一个子查询结果集的复制，在引用处直接查询该复制，因此WITH子查询无法和主干SELECT语句进行联合优化（如谓词下推、等价类传递等），对于此类场景可以使用NOT MATERIALIZED进行修饰，如果WITH查询语义上可以作为子查询内联执行，则可以进行上述优化。
  - 如果用户没有显示声明物化属性则遵守以下规则：如果CTE只在所属主干语句中被引用一次，且语义上支持内联执行，则会被改写为子查询内联执行，否则以CTE Scan的方式物化执行。

说明

INSERT ON DUPLICATE KEY UPDATE不支持WITH及WITH RECURSIVE子句。

- **plan_hint子句**

  以/*+ */的形式在INSERT关键字后，用于对INSERT对应的语句块生成的计划进行hint调优，详细用法请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。每条语句中只有第一个/*+ plan_hint */注释块会作为hint生效，里面可以写多条hint。

- **IGNORE**

  INSERT语句使用IGNORE关键字时，可将某些ERROR级别的错误降级为WARNING级抛出，并根据不同的错误场景将无效值调整为最接近的值插入。GaussDB支持错误降级的错误码如下：
  - 控制破坏NOT NULL约束。
  - 唯一键冲突。
  - 插入的值没有找到对应的分区。
  - 指定分区插入时，插入的数据与指定分区不匹配。
  - 子查询返回多行。
  - 数据过长。
  - 时间函数溢出。
  - 被0除。
  - 值不正确。

版权所有 © 华为云计算技术有限公司

说明

1. INSERT IGNORE只在MYSQL模式（即sql_compatibility = 'MYSQL'），b_format_version = '5.7'和b_format_dev_version = 's1'场景下支持。
2. INSERT IGNORE不支持密态表、外表。
3. INSERT IGNORE不支持延迟生效的主键约束或唯一键约束。
4. INSERT IGNORE支持NULL值破坏NOT NULL约束的数据类型对应的oid为：TIMESTAMPOID、TIMESTAMPTZOID、TIMEOID、TIMETZOID、RELTIMEOID、INTERVALOID、TINTERVALOOID、SMALLDATETIMEOID、DATEOID、NAMEOID、POINTOID、PATHOID、POLYGONOID、CIRCLEOID、LSEGOID、BOXOID、JSONOID、JSONBOID、XMLOID、XMLTYPEOID、VARBITOID、NUMERICOID、CIDROID、INETOID、MACADDROID、NUMRANGEOID、INT8RANGEOID、INT4RANGEOID、TSRANGEOID、TSTZRANGEOID、DATERANGEOID、ABSTIMEOID、BOOLOID、INT8OID、INT4OID、INT2OID、INT1OID、FLOAT4OID、FLOAT8OID、CASHOID、UINT1OID、UINT2OID、UINT4OID、UINT8OID、BPCHAROID、VARCHAROID、NVARCHAR2OID、CHAROID、BYTEAOID、RAWOID、BLOBOID、CLOBOID、TEXTOID、YEAROID，各数据类型对应的零值参见**表1-403**。
5. 默认零值参与计算时，需要谨慎使用IGNORE忽略NOT NULL约束。
6. INSERT IGNORE不支持包含全局二级索引的表。
7. 如果错误是在CN上产生，则会显示降级的WARNING信息，如果错误是在DN上产生，则不会显示降级后的WARNING信息。

表 1-403 数据类型默认零值

数据类型的OID	默认零值
INT8OID、INT4OID、INT2OID、INT1OID、UINT1OID、UINT2OID、UINT4OID、UINT8OID、FLOAT4OID、FLOAT8OID、NUMERICOID	0或0.00（小数点后0的个数由参数指定）
BPCHAROID、VARCHAROID、CHAROID、BYTEAOID、RAWOID、BLOBOID、NVARCHAR2OID、CLOBOID、TEXTOID、VARBITOID、NAMEOID	空字符串
NUMRANGEOID、INT8RANGEOID、INT4RANGEOID、TSRANGEOID、TSTZRANGEOID、DATERANGEOID	empty
TIMEOID	time '00:00:00'
TIMETZOID	timetz '00:00:00'
INTERVALOID	interval '00:00:00'
TINTERVALOOID	tinterval(abstime '1970-01-01 00:00:00', abstime '1970-01-01 00:00:00')
SMALLDATETIMEOID	smalldatetime '1970-01-01 00:00:00'
ABSTIMEOID	abstime '1970-01-01 00:00:00'

版权所有 © 华为云计算技术有限公司

数据类型的OID	默认零值
RELTIMEOID	reltime '00:00:00'
TIMESTAMPOID、TIMESTAMPTZOID	1970-01-01 00:00:00
DATEOID	1970-01-01
YEAROID	0000
POINTOID	(0,0)
PATHOID、POLYGONOID	((0,0))
CIRCLEOID	<(0,0),0>
LSEGOID	[(0,0),(0,0)]
BOXOID	(0,0),(0,0)
JSONOID、JSONBOID、XMLOID	'null'
XMLTYPEOID	'<null/>'
CIDROID	0.0.0.0/32
INETOID	0.0.0.0
MACADDROID	00:00:00:00:00:00
BOOLOID	f
CASHOID	$0.00

- **table_name**

  要插入数据的目标表名。

  取值范围：已存在的表名。

  说明

  支持使用DATABASE LINK方式对远程表进行操作，使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

- **subquery**

  插入对象可以是子查询，在对子查询进行插入时，会将子查询当成一个临时视图，支持在子查询后面加CHECK OPTION选项。

  如果子查询中只有一张表，则对该表插入数据；如果子查询中有多张表或有嵌套关系，则通过判断是否有保留键表确定是否可插入。关于保留键表和WITH CHECK OPTION请参见**CREATE VIEW**。

- **view_name**

  要插入的目标视图。

版权所有 © 华为云计算技术有限公司

说明

对视图和子查询的插入，有如下约束：

- 只有直接引用基表用户列的列可进行INSERT。
- 子查询或视图必须至少包含一个可更新列，关于可更新列请参见**CREATE VIEW**。
- 不支持在顶层包含DISTINCT、GROUP BY、HAVING、LIMIT、OFFSET子句的视图和子查询。
- 不支持在顶层包含集合运算（UNION、INTERSECT、EXCEPT、MINUS）的视图和子查询。
- 不支持目标列表中包含聚集函数、窗口函数、返回集合函数（array_agg、json_agg、generate_series等）的视图和子查询。
- 不支持视图仅带有BEFORE/AFTER触发器，没有INSTEAD OF触发器或INSTEAD规则。
- 不支持ON DUPLICATE KEY UPDATE功能。
- 视图和子查询中支持的表类型包括普通表、临时表、全局临时表、分区表、二级分区表、ustore表、astore表。
- 对多表连接视图或子查询，一次只能对一张基表做插入。
- INSERT连接视图或子查询时显式指定的插入列或隐式指定（即创建视图或子查询中指定的列）不能引用非保留键表的列，如果使用了WITH CHECK OPTION子句则连接视图中或子查询中的连接列无法进行插入操作。关于保留键表请参见**CREATE VIEW**。
- 不支持对系统视图进行INSERT操作。

- **alias_name**

  INSERT语句不包含AS alias语法时的表别名。

  说明

  1. INSERT不包含AS alias语法时的表别名不能为关键字（如SELECT、VALUE）或者表达式，别名要求符合标识符命名规则。
  2. INSERT的分区表不包含AS alias语法时的表别名不支持指定分区插入操作。

- **partition_clause**

  指定分区插入操作。

  关键字详请参见**SELECT**。

  如果VALUE子句的值和指定分区不一致，结果会提示异常。

- **column_name**

  目标表中的字段名称：

  - 字段名可以有子字段名或者数组下标修饰。
  - 没有在字段列表中出现的每个字段，将由系统默认值，或者声明时的默认值填充，若都没有则用NULL填充。例如，向一个复合类型中的某些字段插入数据的话，其他字段将是NULL。
  - 目标字段（column_name）可以按顺序排列。如果没有列出任何字段，则默认全部字段，且顺序为表声明时的顺序。
  - 如果VALUE子句和query中只提供了N个字段，则目标字段为前N个字段。
  - VALUE子句和query提供的值在表中从左到右关联到对应列。
  - 支持insert表别名.列名的用法。

  取值范围：已存在的字段名。

- **expression**

  赋予对应column的一个有效表达式或值：

版权所有 © 华为云计算技术有限公司

  - 如果是INSERT ON DUPLICATE KEY UPDATE语句下，expression可以为VALUES(column_name)或EXCLUDED.column_name用来表示引用冲突行对应的column_name字段的值。其中VALUES(column_name)支持嵌套在复合表达式中，例如：VALUES(column_name)+1、VALUES(column_name)+VALUES(column_name)、function_name(VALUES(column_name))等。

    说明

    - 上述VALUES(column_name)特性仅支持在ON DUPLICATE KEY UPDATE子句中使用。
    - 上述VALUES(column_name)特性不支持在IN或NOT IN表达式中使用。
  - 向表中字段插入单引号 " ' "时需要使用单引号自身进行转义。
  - 如果插入行的表达式不是正确的数据类型，系统试图进行类型转换，若转换不成功，则插入数据失败，系统返回错误信息。

- **DEFAULT**

  对应字段名的缺省值。如果没有缺省值，则为NULL。

- **query**

  一个查询语句（SELECT语句），将查询结果作为插入的数据。

- **RETURNING**

  返回实际插入的行，RETURNING列表的语法与SELECT的输出列表一致。

- **output_expression**

  INSERT命令在每一行都被插入之后用于计算输出结果的表达式。

  取值范围：该表达式可以使用table的任意字段。可以使用*返回被插入行的所有字段。

- **output_name**

  字段的输出名称。

  取值范围：字符串，要符合**标识符命名规范**。

- **ON DUPLICATE KEY UPDATE**

  对于带有唯一约束（UNIQUE INDEX或PRIMARY KEY）的表，如果插入数据违反唯一约束，则对冲突行执行UPDATE子句完成更新。如果UPDATE子句为NOTHING，则不做任何操作。可通过"EXCLUDED." 或者 "VALUES()" 来选择源数据相应的列。

  对于不带唯一约束的表，则仅执行插入。

  - INSERT ... ON DUPLICATE KEY UPDATE语句中支持触发语句级（STATEMENT）触发器，不支持触发行级（ROW）触发器。触发器由执行的语句决定：
    - 执行UPDATE NOTHING：触发BEFORE INSERT、AFTER INSERT触发器。
    - 执行UPDATE：触发BEFORE INSERT、AFTER INSERT、BEFORE UPDATE、AFTER UPDATE触发器。
  - 不支持延迟生效（DEFERRABLE）的唯一约束或主键。
  - 如果表中存在多个唯一约束，如果所插入数据违反多个唯一约束，对于检测到冲突的第一行进行更新，其他冲突行不更新（检查顺序与索引维护具有强相关性，一般先创建的索引先进行冲突检查）。
  - 主键、唯一索引列不允许UPDATE。

版权所有 © 华为云计算技术有限公司

  - 不支持外表。
  - UPDATE的WHERE子句和expression中不支持包含子链接。

- **ON CONFLICT [conflict_target] conflict_action**

  在进行插入操作期间存在唯一约束或排他性约束冲突时，指定是否继续进行插入操作的行为。未发生冲突时，执行普通INSERT流程。当发生冲突时，ON CONFLICT后指定为DO NOTHING时，放弃当前插入操作；ON CONFLICT后指定为DO UPDATE SET时，执行更新操作。

  - **conflict_target**
    - index_column_name：索引列的列名。
    - index_expression：与index_column_name类似，为基于该表的一个或多个字段的表达式索引，可用于获取对基本数据的某种变形的快速访问。
    - index_predicate：索引的一个条件表达式，用于描述部分唯一索引的范围。
  - **conflict_action**
    - expression：要赋给对应列的表达式或值。
    - DEFAULT：相应列将被填充为默认值。

  说明

  对INSERT ... ON CONFLICT语法的使用，存在如下约束：

  - 仅支持PG兼容性。
  - 不支持[COLLATE collation] [opclass]语法。
  - 不支持ON CONSTRAINT constraint_name语法。
  - 当执行DO UPDATE SET时，用户需要有UPDATE操作权限。
  - 不支持在conflict_target子句和UPDATE SET子句中嵌套子链接，不支持在UPDATE SET子句中嵌套子查询。
  - 不支持列格式表、外表、STREAM对象以及CONTVIEW对象。
  - 不支持更新分布键。
  - 分布式下，当执行计划为stream、light proxy或fqs时，不支持在同一语句中一次插入多条唯一键相同的数据，当执行计划为PGXC时，支持在同一语句中一次插入多条唯一键相同的数据。
  - 不支持使用GSI全局二级索引场景。
  - 不支持DBLINK。
  - 不支持插入子查询。
  - 支持触发语句级（STATEMENT）触发器，不支持触发行级（ROW）触发器。

示例

- **插入一条数据**

  示例：

```
--建表。
gaussdb=# CREATE TABLE test_t1(col1 INT,col2 VARCHAR);

--插入数据。
gaussdb=# INSERT INTO test_t1 (col1, col2) VALUES (1,'AB');
```


版权所有 © 华为云计算技术有限公司

```
--只给表中部分列添加值。
gaussdb=# INSERT INTO test_t1 (col1) VALUES (2);

--VALUES关键字左边没有括号，右边括号里面必须严格按照表结构的顺序给所有的字段添加值。
gaussdb=# INSERT INTO test_t1 VALUES (3,'AC');

--查询表。
gaussdb=# SELECT * FROM test_t1;
 col1 | col2
------+------
    1 | AB
    2 |
    3 | AC
(3 rows)
--删除。
gaussdb=# DROP TABLE test_t1;
```

- **插入多条数据**

  示例：

```
--建表。
gaussdb=# CREATE TABLE test_t2(col1 INT,col2 VARCHAR);
gaussdb=# CREATE TABLE test_t3(col1 INT,col2 VARCHAR);

--插入多条数据。
gaussdb=# INSERT INTO test_t2 (col1, col2) VALUES (10,'AA'),(20,'BB'),(30,'CC');

--查询。
gaussdb=# SELECT * FROM test_t2;
 col1 | col2
------+------
   10 | AA
   20 | BB
   30 | CC
(3 rows)

--把test_t2中的数据插入到test_t3中。
gaussdb=# INSERT INTO test_t3 SELECT * FROM test_t2;

--查询。
gaussdb=# SELECT * FROM test_t3;
 col1 | col2
------+------
   10 | AA
   20 | BB
   30 | CC
(3 rows)

--删除。
gaussdb=# DROP TABLE test_t2;
DROP TABLE test_t3;
```

- **ON DUPLICATE KEY UPDATE**

  示例：

```
--建表。
gaussdb=# CREATE TABLE test_t4 (id INT PRIMARY KEY, info VARCHAR(10));
gaussdb=# INSERT INTO test_t4 VALUES (1, 'AA'), (2,'BB'), (3, 'CC');

--使用ON DUPLICATE KEY UPDATE关键字。
gaussdb=# INSERT INTO test_t4 VALUES (3, 'DD'), (4, 'EE') ON DUPLICATE KEY UPDATE info =
VALUES(info);

--查询。
gaussdb=# SELECT * FROM test_t4;
 id | info
----+------
  1 | AA
  2 | BB
  4 | EE
  3 | DD
```


版权所有 © 华为云计算技术有限公司

```
--删除。
gaussdb=# DROP TABLE test_t4;
```

- **ON CONFLICT**

  示例：

```
-- 创建PG兼容模式数据库以及创建数据表。
gaussdb=# CREATE DATABASE pg_db dbcompatibility = 'PG';
gaussdb=# \c pg_db
pg_db=# CREATE TABLE insert_onconflict_tb1(id INT, a INT PRIMARY KEY, b INT, c VARCHAR2(20));
pg_db=# INSERT INTO insert_onconflict_tb1 VALUES (1, 1, 2, 'aaa'), (2, 2, 0, 'zzz');

-- 查询表信息。
pg_db=# SELECT * FROM insert_onconflict_tb1;
 id | a | b |  c
----+---+---+-----
  1 | 1 | 2 | aaa
  2 | 2 | 0 | zzz
(2 rows)

-- 使用ON CONFLICT关键字。
pg_db=# INSERT INTO insert_onconflict_tb1 VALUES (1, 1, 5, 'ddd'), (2, 2, 1, 'yyy') ON CONFLICT (a)
DO UPDATE SET b=excluded.b, c=excluded.c;
pg_db=# INSERT INTO insert_onconflict_tb1 VALUES (1, 1, 4, 'ccc'), (2, 2, 3, 'zzz') ON CONFLICT (a)
DO NOTHING;

-- 查询表信息。
pg_db=# SELECT * FROM insert_onconflict_tb1;
 id | a | b |  c
----+---+---+-----
  1 | 1 | 5 | ddd
  2 | 2 | 1 | yyy
(2 rows)

-- 删除数据表并删除PG兼容模式数据库。
pg_db=# DROP TABLE insert_onconflict_tb1;
pg_db=# \c postgres
gaussdb=# DROP DATABASE pg_db;
```

- **INSERT IGNORE**

  示例1：破坏NOT NULL约束

```
--建表。
gaussdb=# CREATE TABLE test_t5(f1 INT NOT NULL);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t5 VALUES(NULL);
WARNING:  The null value in column "f1" violates the not-null constraint
DETAIL:  Failing row contains (null).
INSERT 0 1

--查询表。
gaussdb=# SELECT * FROM test_t5;
 f1
----
  0
(1 row)

--删除表。
gaussdb=# DROP TABLE test_t5;
DROP TABLE
```

  示例2：唯一键冲突

```
--建表。
gaussdb=# CREATE TABLE test_t6(f1 INT PRIMARY KEY);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "test_t6_pkey" for table "test_t6"
```


版权所有 © 华为云计算技术有限公司

```
CREATE TABLE

--插入数据。
gaussdb=# INSERT INTO test_t6 VALUES(1);
INSERT 0 1

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t6 VALUES(1);
INSERT 0 0

--查询表。
gaussdb=# SELECT * FROM test_t6;
 f1
----
  1
(1 row)

--删除表。
gaussdb=# DROP TABLE test_t6;
DROP TABLE
```

示例3：插入的值没有找到对应的分区

```
--建表。
gaussdb=# CREATE TABLE test_t7(f1 INT, f2 INT) PARTITION BY LIST(f1) (PARTITION p0 VALUES(1, 4,
7), PARTITION p1 VALUES (2, 5, 8));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t7 VALUES(3, 5);
INSERT 0 0

--查询表。
gaussdb=# SELECT * FROM test_t7;
 f1 | f2
----+----
(0 rows)

--删除表。
gaussdb=# DROP TABLE test_t7;
DROP TABLE
```

示例4：指定分区插入时，插入的数据与指定的分区不匹配

```
--建表。
gaussdb=# CREATE TABLE test_t8(f1 INT NOT NULL, f2 TEXT, f3 INT) PARTITION BY RANGE(f1)
(PARTITION p0 VALUES LESS THAN(5), PARTITION p1 VALUES LESS THAN(10), PARTITION p2
VALUES LESS THAN(15), PARTITION p3 VALUES LESS THAN(MAXVALUE));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t8 PARTITION(p2) VALUES(20, 'Jan', 1);
INSERT 0 0

--查询表。
gaussdb=# SELECT * FROM test_t8;
 f1 | f2 | f3
----+----+----
(0 rows)

--删除表。
gaussdb=# DROP TABLE test_t8;
DROP TABLE
```

示例5：子查询返回多行

```
--建表。
gaussdb=# CREATE TABLE test_t9(f1 INT, f2 INT);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f1' as the distribution column by default.
```


版权所有 © 华为云计算技术有限公司

```
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--插入数据。
gaussdb=# INSERT INTO test_t9 VALUES(1, 1), (2, 2), (3, 3);
INSERT 0 3

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t9 VALUES((SELECT f1 FROM test_t9), 0);
WARNING:  more than one row returned by a subquery used as an expression
CONTEXT:  referenced column: f1
INSERT 0 1

--查询表。
gaussdb=# SELECT * FROM test_t9 WHERE f2 = 0;
 f1 | f2
----+----
    |  0
(1 row)

--删除表。
gaussdb=# DROP TABLE test_t9;
DROP TABLE
```

示例6：数据过长

```
--建表。
gaussdb=# CREATE TABLE test_t10(f1 VARCHAR(5));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t10 VALUES('aaaaaaaaa');
WARNING:  value too long for type character varying(5)
CONTEXT:  referenced column: f1
INSERT 0 1

--查询表。
gaussdb=# SELECT * FROM test_t10;
  f1
-------
 aaaaa
(1 row)

--删除表。
gaussdb=# DROP TABLE test_t10;
DROP TABLE
```

示例7：时间函数溢出

```
--建表。
gaussdb=# CREATE TABLE test_t11(f1 DATETIME);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t11 VALUES(date_sub('2000-01-01', INTERVAL 2001 YEAR));
WARNING:  Datetime function: datetime field overflow
CONTEXT:  referenced column: f1
INSERT 0 1

--查询表。
gaussdb=# SELECT * FROM test_t11;
 f1
----

(1 row)

--删除表。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# DROP TABLE test_t11;
DROP TABLE
```

示例8：被0除

```
--建表。
gaussdb=# CREATE TABLE test_t12(f1 INT);
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t12 VALUES(1/0);
WARNING:  division by zero
CONTEXT:  referenced column: f1
INSERT 0 1

--查询表。
gaussdb=# SELECT * FROM test_t12;
 f1
----

(1 row)

--删除表。
gaussdb=# DROP TABLE test_t12;
DROP TABLE
```

示例9：值不正确

```
--建表。
gaussdb=# CREATE TABLE test_t13(f0 INT, f1 FLOAT);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'f0' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用IGNORE关键字。
gaussdb=# INSERT IGNORE INTO test_t13 VALUES(1, '1.11aaa');
WARNING:  invalid input syntax for type real: "1.11aaa"
LINE 1: INSERT IGNORE INTO test_t13 VALUES(1, '1.11aaa');
                                              ^
CONTEXT:  referenced column: f1
INSERT 0 1

--查询表。
gaussdb=# SELECT * FROM  test_t13;
 f0 |  f1
----+------
  1 | 1.11
(1 row)

--删除表。
gaussdb=# DROP TABLE test_t13;
DROP TABLE
```

示例10：使用表别名插入一条数据

```
--建表。
gaussdb=# create table tb1 (va int , vb int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'va' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--使用表别名
gaussdb=# insert into tb1 as tt(tt.va, tt.vb) values (1,2);
INSERT 0 1
gaussdb=# insert into tb1 tt(tt.va, tt.vb) values (3,4);
INSERT 0 1

--查询表。
gaussdb=# select * from tb1;
 va | vb
----+----
  1 |  2
  3 |  4
```


版权所有 © 华为云计算技术有限公司

```
(2 rows)

--删除表。
gaussdb=# DROP TABLE tb1;
DROP TABLE
```

- **WITH [RECURSIVE] with_query [, ...]**

  示例：

```
--成绩表。
gaussdb=# CREATE TABLE grade (
    sid INT,
    course VARCHAR(20),
    score FLOAT
);
--学生表。
gaussdb=# CREATE TABLE student(
    sid INT PRIMARY KEY,
    class INT,
    name  VARCHAR(50),
    gender INT CHECK (gender = 0 or gender = 1)
);

--插入数据。
gaussdb=# WITH student_sid AS ( INSERT INTO student ( sid, CLASS, NAME, gender ) VALUES ( 1, 1,
'scott', 1 ) RETURNING sid )
    INSERT INTO grade ( sid, course, score )
    VALUE ( ( SELECT sid FROM student_sid ), '数学', '96' ),
   ( ( SELECT sid FROM student_sid ), '语文', '82' ),
   ( ( SELECT sid FROM student_sid ), '英语', '86' );

--查询数据。
gaussdb=# SELECT * FROM student;
 sid | class | name  | gender
-----+-------+-------+-----
   1 |     1 | scott |   1
(1 row)

gaussdb=# SELECT * FROM grade;
 sid | course  | score
-----+---------+-------
   1 | 数学    |    96
   1 | 语文    |    82
   1 | 英语    |    86
(3 rows)

--删除。
gaussdb=# DROP TABLE student;
gaussdb=# DROP TABLE grade;
```

- **向视图或子查询中插入**

  示例1：插入子查询

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA ins_subqry;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = 'ins_subqry';
SET

--创建表。
gaussdb=# CREATE TABLE t1 (x1 int, y1 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE t2 (x2 int PRIMARY KEY, y2 int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t2_pkey" for table "t2"
CREATE TABLE

--通过子查询向t1插入。
gaussdb=# INSERT INTO (SELECT * FROM t1) VALUES (1, 1);
```


版权所有 © 华为云计算技术有限公司

```
INSERT 0 1
gaussdb=# INSERT INTO (SELECT * FROM t1 WHERE y1 < 3) VALUES (5, 5);
INSERT 0 1

--插入子查询带CHECK OPTION。
gaussdb=# INSERT INTO (SELECT * FROM t1 WHERE y1 < 3 WITH CHECK OPTION) VALUES (5, 5);
ERROR:  New row violates WITH CHECK OPTION for view "__unnamed_subquery__"
DETAIL:  Failing row contains (5, 5).

--插入子查询带READONLY。
gaussdb=# INSERT INTO (SELECT * FROM t1 WITH READ ONLY) VALUES (5, 5);
ERROR:  DML operation cannot be performed on a read-only view.

--插入多表连接的子查询。
gaussdb=# INSERT INTO (SELECT * FROM t1, t2 WHERE x1 = x2) (x1, y1) VALUES (2, 2);
INSERT 0 1

--删除SCHEMA。
gaussdb=# RESET CURRENT_SCHEMA;
RESET
gaussdb=# DROP SCHEMA ins_subqry CASCADE;
NOTICE:  drop cascades to 2 other objects
DETAIL:  drop cascades to table ins_subqry.t1
drop cascades to table ins_subqry.t2
DROP SCHEMA
```

示例2： 插入视图

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA ins_view;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = 'ins_view';
SET

--创建表。
gaussdb=# CREATE TABLE t1 (x1 int, y1 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE t2 (x2 int PRIMARY KEY, y2 int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t2_pkey" for table "t2"
CREATE TABLE

--创建单表视图。
gaussdb=# CREATE VIEW v_ins1 AS SELECT * FROM t1;
CREATE VIEW
gaussdb=# CREATE VIEW v_ins2 AS SELECT * FROM t1 WHERE y1 < 3;
CREATE VIEW
gaussdb=# CREATE VIEW v_ins2_wco AS SELECT * FROM t1 WHERE y1 < 3 WITH CHECK OPTION;
CREATE VIEW
gaussdb=# CREATE VIEW v_ins_read AS SELECT * FROM t1 WITH READ ONLY;
CREATE VIEW

--通过视图对t1插入。
gaussdb=# INSERT INTO v_ins1 VALUES (1, 1);
INSERT 0 1
gaussdb=# INSERT INTO v_ins2 VALUES (5, 5);
INSERT 0 1
gaussdb=# INSERT INTO v_ins2_wco VALUES (5, 5);
ERROR:  new row violates WITH CHECK OPTION for view "v_ins2_wco"
DETAIL:  Failing row contains (5, 5).

gaussdb=# INSERT INTO v_ins_read VALUES (5, 5);
ERROR:  DML operation cannot be performed on a read-only view.

--创建多表视图。
gaussdb=# CREATE VIEW vv_ins AS SELECT * FROM t1, t2 WHERE x1 = x2;
CREATE VIEW

--通过视图对t1插入。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO vv_ins (x1, y1) VALUES (2, 2);
INSERT 0 1

--删除SCHEMA。
gaussdb=# RESET CURRENT_SCHEMA;
RESET
gaussdb=# DROP SCHEMA ins_view CASCADE;
NOTICE:  drop cascades to 7 other objects
DETAIL:  drop cascades to table ins_view.t1
drop cascades to table ins_view.t2
drop cascades to view ins_view.v_ins1
drop cascades to view ins_view.v_ins2
drop cascades to view ins_view.v_ins2_wco
drop cascades to view ins_view.v_ins_read
drop cascades to view ins_view.vv_ins
DROP SCHEMA
```

优化建议

- VALUES

  通过INSERT语句批量插入数据时，建议将多条记录合并入一条语句中执行插入，以提高数据加载性能。

  例如：

  ```
  INSERT INTO sections VALUES (30, 'Administration', 31, 1900),(40, 'Development', 35, 2000), (50, 'Development' , 60 , 2001);
  ```

  如果INSERT多VALUES语句中VALUES的值分布在一个DN上，GaussDB可以把语句下推到对应DN执行。目前只支持VALUES中值为常量，简单表达式和可下推函数（pg_proc中字段provolatile为'i'）。如果表中列带有DEFAULT值，只支持DEFAULT值为常量，简单表达式。单VALUES不能下推单DN的语句，多VALUES同样不支持下推。

- 批量插入索引性能优化

  当以下任何一个场景满足所有的约束条件时，建议打开GUC参数batch_insert_index_types并设置为"rcr_ubtree"以提升批量插入索引性能：

  场景1：使用JDBC连接数据库，通过java的批量插入数据接口executeBatch进行批量插入索引；

  场景2：执行语句insert into t1 select * from t2，其中t1和t2表schema相同，t1表包含索引，此语句支持在t1表上批量插入索引；此场景必须要开启stream计划（enable_fast_query_shipping参数被设置成off），并在批量插入表数据的前提下才会批量插入索引（explain verbose命令可以查看是否对表数据做批量插入，Streaming代表是批量插入数据）。满足这些前提后，打开批量插入索引开关就会执行批量插入索引；

  场景3：执行语句COPY FROM（示例：COPY t1 FROM 'file.csv' delimiter ',' csv header），其中t1和file.csv文件中的schema相同，t1表包含索引，此语句支持在t1表上批量插入索引。

  约束1：插入的索引类型为UBTree RCR索引；

  约束2：插入数据不递增（插入数据递增会利用缓存，即最后一页，这种情况下批量插入索引无性能提升）；

  约束3：enable_stream_operator参数被设置成on；

  约束4：不是SMP的批量索引插入。

版权所有 © 华为云计算技术有限公司

说明

此特性在以下三个场景会影响性能：

1. 如果批量插入数据是递增的（违反约束2），关闭批量插入功能后，性能较好。这种场景下开启批量插入功能会导致性能出现轻微劣化。

2. 在执行持续的批量插入时，索引会逐渐变大，进而导致批量插入性能有所下降。

3. 批量插入数据过于分散导致缓存页面命中率过低，或者运行内存过小的场景下，开启批量插入功能会导致性能出现较大幅度劣化。

如果性能劣化幅度较大，建议设置GUC参数batch_insert_index_types为""，关闭该功能，性能劣化问题即可恢复。

如果需要技术支持工程师定位原因，可以设置GUC参数logging_module='on(BIINDEX)'，打开批量插入索引的日志打印，执行几分钟批量插入直到日志中出现BIINDEX字符串为止。

1.13.14.6 INSERT ALL

功能描述

向多个表中插入一条或多条数据。

注意事项

- 只有拥有表INSERT权限的用户，才可以向表中插入数据。用户被授予INSERT ANY TABLE权限，即用户对除系统模式之外的任何模式具有USAGE权限，并且拥有这些模式下表的INSERT权限。
- 如果使用QUERY子句插入来自查询里的数据行，用户还需要拥有在查询里使用的表的SELECT权限。
- 生成列不能被直接写入。在INSERT命令中不能为生成列指定值，但可以指定关键字DEFAULT。
- 仅支持在ORA兼容模式下执行该语法。

语法格式

```
INSERT [/*+ plan_hint */] [ ALL | FIRST ]
   { [ WHEN condition THEN ]
   { INTO table_name
   [ { [alias_name] [ ( column_name [, ...] ) ] } | { [partition_clause] [ AS alias ]
   [ ( column_name [, ...] ) ] } ]
   [ VALUES {( { expression | DEFAULT } [, ...] ) } ] }[, ...] }[, ...]
   { subquery };
```

其中subquery的具体信息为：

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [/*+ plan_hint */] [ ALL ]
{ * | {expression [ [ AS ] output_name ]} [, ...] }
[ into_option ]
[ FROM from_item [, ...] ]
[ WHERE condition ]
[ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS { FIRST | LAST } ]} [, ...] ]
[ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]
[ into_option ];
```

- 其中with_query的具体信息为：

  ```
  with_query_name [ ( column_name [, ...] ) ] AS [ [ NOT ] MATERIALIZED ]
  ( {select | values | insert | update | delete} )
  ```


版权所有 © 华为云计算技术有限公司

- 其中指定子查询源from_item为：

```
{[ ONLY ] {table_name | view_name} [ * ] [ partition_clause ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
|( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
|with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
|from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]}
```

其中partition_clause的具体信息为：

```
PARTITION { ( partition_name ) | FOR ( partition_value [, ...] ) } |
SUBPARTITION { ( subpartition_name ) | FOR ( subpartition_value [, ...] ) }
```

参数说明

- **plan_hint子句**

  以/*+ */的形式在INSERT关键字后，目前仅适配了语法格式，并未实现具体功能。

- **ALL/FIRST**

  如果不带条件子句（WHEN condition THEN），此关键字不可省略，且只能为ALL。如果含有条件子句，此关键字可以省略，默认为ALL，此时ALL和FIRST的含义分别为：

  - ALL：遍历所有匹配条件的分支，只要满足条件就会将数据插入对应的表。
  - FIRST：查询到第一个满足条件的分支后，将数据插入该分支对应的表，不再继续检查剩余分支。

- **WHEN condition THEN**

  插入表时的条件判断，只有满足条件才会进行插入操作，condition中可以引用子查询（SUBQUERY）中的列。

  condition中不支持子查询。

- **table_name**

  要插入数据的目标表名。

  取值范围：已存在的表名。

- **alias_name**

  INSERT语句不包含AS alias语法时的表别名。

  说明

  1. INTO子句不包含AS alias语法时的表别名不能为关键字（如SELECT、VALUE）或者表达式，表别名要求符合**标识符命名规范**。
  2. INTO子句不包含AS alias语法时的表别名不支持INTO table_name alias_name(alias_name.col1,...,alias_name.coln) VALUES(xxx);的语法格式。
  3. INTO子句的分区表不包含AS alias语法时的表别名不支持指定分区插入操作。

- **partition_clause**

  指定分区插入操作。

```
PARTITION { ( partition_name ) | FOR ( partition_value [, ...] ) } |
SUBPARTITION { ( subpartition_name ) | FOR ( subpartition_value [, ...] ) }
```

  关键字具体信息请参见**SELECT**。

  如果VALUE子句的值和指定分区不一致，结果会提示异常。

- **column_name**

  目标表中的字段名称。

  - 字段名可以有子字段名或者数组下标修饰。
  - 没有在字段列表中出现的每个字段，将由系统默认值或者声明时的默认值进行填充，若无默认值或未声明时则使用NULL填充。例如，向一个复合类型中的某些字段插入数据，其他字段使用NULL进行填充。

版权所有 © 华为云计算技术有限公司

  - 目标字段（column_name）按顺序排列。如果没有列出任何字段，则默认全部字段，且顺序为表声明时的顺序。
  - 如果VALUE子句和QUERY中只提供了N个字段，则目标字段为前N个字段。
  - VALUE子句和QUERY提供的值在表中从左到右关联到对应列。

  取值范围：已存在的字段名。

- **VALUES**

  不可跟多行。

  不支持子查询。

- **expression**

  赋予对应column的一个有效表达式或值。
  - 向表中字段插入单引号（'）时需要使用单引号对自身进行转义。
  - 如果插入行的表达式不是正确的数据类型，系统将自动进行类型转换，若转换失败，则插入数据失败，系统返回错误信息。
  - 不支持聚集函数。
  - 不支持子查询。

- **DEFAULT**

  对应字段名的缺省值。如果没有缺省值，则为NULL。

- **subquery**

  一个查询语句（SELECT语句），将查询结果作为插入的数据，不可省略，如果没有子查询可以使用SELECT * FROM DUAL。

  如果子查询含有表别名，不可在CONDITION子句和INTO子句中引用该别名。

  具体示例请参见**SELECT**。

示例

- **不带条件的多表插入**

  示例：

```
--创建ORA兼容数据库。
gaussdb=# CREATE DATABASE insertall_test_db DBCOMPATIBILITY 'ORA';

--切换至ORA兼容数据库。
gaussdb=# \c insertall_test_db;

--建表。
insertall_test_db=# CREATE TABLE insert_all_test0(c1 INT, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test1(c1 INT, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test2(c1 INT, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test3(c1 INT, c2 VARCHAR2(20));

--构造数据。
insertall_test_db=# INSERT INTO insert_all_test0 VALUES(1,'a'),(2,'b'),(3,'c'),(4,'d'),(5,'e');

--向表中插入数据。
insertall_test_db=# INSERT ALL
INTO insert_all_test1
INTO insert_all_test1 VALUES(1,'a')
INTO insert_all_test1(c1,c2)
INTO insert_all_test1(c1) VALUES(1)
INTO insert_all_test1(c2) VALUES('aaa')
INTO insert_all_test2(c2) VALUES(c2)
INTO insert_all_test2(c2) VALUES(c2||'*')
INTO insert_all_test3(c1,c2) VALUES(1,'a')
INTO insert_all_test3(c1,c2) VALUES(1,c2)
INTO insert_all_test3(c1,c2) VALUES(c1,c2)
```


版权所有 © 华为云计算技术有限公司

```
INTO insert_all_test3(c2,c1) VALUES('***',666)
SELECT * FROM insert_all_test0;

--查询数据。
insertall_test_db=# SELECT * FROM insert_all_test1 ORDER BY c1,c2;
 c1 | c2
----+-----
  1 | a
  1 | a
  1 | a
  1 | a
  1 | a
  1 | a
  1 | a
  1 |
  1 |
  1 |
  1 |
  1 |
  2 | b
  2 | b
  3 | c
  3 | c
  4 | d
  4 | d
  5 | e
  5 | e
    | aaa
    | aaa
    | aaa
    | aaa
    | aaa
(25 rows)

insertall_test_db=# SELECT * FROM insert_all_test2 ORDER BY c1,c2;
 c1 | c2
----+----
    | a
    | a*
    | b
    | b*
    | c
    | c*
    | d
    | d*
    | e
    | e*
(10 rows)

insertall_test_db=# SELECT * FROM insert_all_test3 ORDER BY c1,c2;
 c1  | c2
-----+-----
   1 | a
   1 | a
   1 | a
   1 | a
   1 | a
   1 | a
   1 | a
   1 | b
   1 | c
   1 | d
   1 | e
   2 | b
   3 | c
   4 | d
   5 | e
 666 | ***
 666 | ***
```


版权所有 © 华为云计算技术有限公司

```
 666 | ***
 666 | ***
 666 | ***
(20 rows)

--删除表。
insertall_test_db=# DROP TABLE insert_all_test0;
insertall_test_db=# DROP TABLE insert_all_test1;
insertall_test_db=# DROP TABLE insert_all_test2;
insertall_test_db=# DROP TABLE insert_all_test3;

insertall_test_db=# \c postgres;
gaussdb=# DROP DATABASE insertall_test_db;
```

- **带条件的多表插入**

  示例1：INSERT FIRST

```
--创建ORA兼容数据库。
gaussdb=# CREATE DATABASE insertall_test_db DBCOMPATIBILITY 'ORA';

--切换至ORA兼容数据库。
gaussdb=# \c insertall_test_db;

--建表。
insertall_test_db=# CREATE TABLE insert_all_test0(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test1(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test2(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test3(c1 int, c2 VARCHAR2(20));

--构造数据。
insertall_test_db=# INSERT INTO insert_all_test0 VALUES(1,'a'),(2,'b'),(3,'c'),(4,'d'),(5,'e');
--向表中插入数据。
insertall_test_db=# INSERT FIRST
WHEN c1 < 4 THEN
INTO insert_all_test1
WHEN c1 < 3 THEN
INTO insert_all_test2
ELSE
INTO insert_all_test3
SELECT * FROM insert_all_test0;

--查询数据。
insertall_test_db=# SELECT * FROM insert_all_test1 ORDER BY c1,c2;
 c1 | c2
----+----
  1 | a
  2 | b
  3 | c
(3 rows)

insertall_test_db=# SELECT * FROM insert_all_test2 ORDER BY c1,c2;
 c1 | c2
----+----
(0 rows)

insertall_test_db=# SELECT * FROM insert_all_test3 ORDER BY c1,c2;
 c1 | c2
----+----
  4 | d
  5 | e
(2 rows)

--删除表。
insertall_test_db=# DROP TABLE insert_all_test0;
insertall_test_db=# DROP TABLE insert_all_test1;
insertall_test_db=# DROP TABLE insert_all_test2;
insertall_test_db=# DROP TABLE insert_all_test3;

insertall_test_db=# \c postgres;
gaussdb=# DROP DATABASE insertall_test_db;
```


版权所有 © 华为云计算技术有限公司

示例2：INSERT ALL

```
--创建ORA兼容数据库。
gaussdb=# CREATE DATABASE insertall_test_db DBCOMPATIBILITY 'ORA';

--切换至ORA兼容数据库。
gaussdb=# \c insertall_test_db;

--建表。
insertall_test_db=# CREATE TABLE insert_all_test0(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test1(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test2(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test3(c1 int, c2 VARCHAR2(20));

--构造数据。
insertall_test_db=# INSERT INTO insert_all_test0 VALUES(1,'a'),(2,'b'),(3,'c'),(4,'d'),(5,'e');

--向表中插入数据，带ALL。
insertall_test_db=# INSERT ALL
WHEN c1 < 4 THEN
INTO insert_all_test1
WHEN c1 < 3 THEN
INTO insert_all_test2
ELSE
INTO insert_all_test3
SELECT * FROM insert_all_test0;

--查询数据。
insertall_test_db=# SELECT * FROM insert_all_test1 ORDER BY c1,c2;
 c1 | c2
----+----
  1 | a
  2 | b
  3 | c
(3 rows)

insertall_test_db=# SELECT * FROM insert_all_test2 ORDER BY c1,c2;
 c1 | c2
----+----
  1 | a
  2 | b
(2 rows)

insertall_test_db=# SELECT * FROM insert_all_test3 ORDER BY c1,c2;
 c1 | c2
----+----
  4 | d
  5 | e
(2 rows)

--删除数据。
insertall_test_db=# TRUNCATE TABLE insert_all_test1;
insertall_test_db=# TRUNCATE TABLE insert_all_test2;
insertall_test_db=# TRUNCATE TABLE insert_all_test3;

--向表中插入数据，省略ALL。
insertall_test_db=# insert
WHEN c1 < 4 THEN
INTO insert_all_test1
WHEN c1 < 3 THEN
INTO insert_all_test2
ELSE
INTO insert_all_test3
SELECT * FROM insert_all_test0;

--查询数据。
insertall_test_db=# SELECT * FROM insert_all_test1 ORDER BY c1,c2;
 c1 | c2
----+----
  1 | a
```


版权所有 © 华为云计算技术有限公司

```
 2 | b
 3 | c
(3 rows)

insertall_test_db=# SELECT * FROM insert_all_test2 ORDER BY c1,c2;
 c1 | c2
----+----
 1 | a
 2 | b
(2 rows)

insertall_test_db=# SELECT * FROM insert_all_test3 ORDER BY c1,c2;
 c1 | c2
----+----
 4 | d
 5 | e
(2 rows)

--删除表。
insertall_test_db=# DROP TABLE insert_all_test0;
insertall_test_db=# DROP TABLE insert_all_test1;
insertall_test_db=# DROP TABLE insert_all_test2;
insertall_test_db=# DROP TABLE insert_all_test3;

insertall_test_db=# \c postgres;
gaussdb=# DROP DATABASE insertall_test_db;
```

- **支持PLSQL中使用INSERT ALL**

  示例：

```
--创建ORA兼容数据库。
gaussdb=# CREATE DATABASE insertall_test_db DBCOMPATIBILITY 'ORA';

--切换至ORA兼容数据库。
gaussdb=# \c insertall_test_db;

--建表。
insertall_test_db=# CREATE TABLE insert_all_test0(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test1(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test2(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test3(c1 int, c2 VARCHAR2(20));

--构造数据。
insertall_test_db=# INSERT INTO insert_all_test0 VALUES(1,'a'),(2,'b'),(3,'c'),(4,'d'),(5,'e');

--PLSQL，向表中插入数据。
insertall_test_db=# DECLARE
var1 INT := 666;
var2 VARCHAR2(20) := '***';
BEGIN
INSERT ALL
INTO insert_all_test1 VALUES(var1,var2)
INTO insert_all_test1(c1) VALUES(var1)
INTO insert_all_test2(c2) VALUES(var2||'???')
INTO insert_all_test3(c1,c2) VALUES(c1,c2)
SELECT * FROM insert_all_test0;
END;
/

--查询数据。
insertall_test_db=# SELECT * FROM insert_all_test1 ORDER BY c1,c2;
 c1  | c2
-----+-----
 666 | ***
 666 | ***
 666 | ***
 666 | ***
 666 | ***
 666 |
 666 |
```


版权所有 © 华为云计算技术有限公司

```
 666 |
 666 |
 666 |
(10 rows)

insertall_test_db=# SELECT * FROM insert_all_test2 ORDER BY c1,c2;
 c1 |   c2
----+--------
    | ***???
    | ***???
    | ***???
    | ***???
    | ***???
(5 rows)

insertall_test_db=# SELECT * FROM insert_all_test3 order by c1,c2;
 c1 | c2
----+----
  1 | a
  2 | b
  3 | c
  4 | d
  5 | e
(5 rows)

--删除表。
insertall_test_db=# DROP TABLE insert_all_test0;
insertall_test_db=# DROP TABLE insert_all_test1;
insertall_test_db=# DROP TABLE insert_all_test2;
insertall_test_db=# DROP TABLE insert_all_test3;

insertall_test_db=# \c postgres;
gaussdb=# DROP DATABASE insertall_test_db;
```

- **异常情况**

  示例：

```
--创建ORA兼容数据库。
gaussdb=# CREATE DATABASE insertall_test_db DBCOMPATIBILITY 'ORA';

--切换至ORA兼容数据库。
gaussdb=# \c insertall_test_db;

--建表。
insertall_test_db=# CREATE TABLE insert_all_test0(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test1(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test2(c1 int, c2 VARCHAR2(20));
insertall_test_db=# CREATE TABLE insert_all_test3(c1 int, c2 VARCHAR2(20));

--构造数据。
insertall_test_db=# INSERT INTO insert_all_test0 VALUES(1,'a'),(2,'b'),(3,'c'),(4,'d'),(5,'e');

--VALUES内不支持聚集函数。
insertall_test_db=# INSERT ALL INTO insert_all_test1 VALUES(max(c1), 'a') SELECT * FROM
insert_all_test0;
ERROR:  An aggregate function cannot be used in VALUES.
LINE 1: INSERT ALL INTO insert_all_test1 VALUES(max(c1), 'a') SELECT

--VALUES后不能添加多行。
insertall_test_db=# INSERT ALL INTO insert_all_test1 VALUES(1,'a'),(2,'b') SELECT * FROM
insert_all_test0;
ERROR:  syntax error at or near ","
LINE 1: INSERT ALL INTO insert_all_test1 VALUES(1,'a'),(2,'b') SELEC...

--WHEN条件中的列在子查询中不存在。
insertall_test_db=# INSERT ALL WHEN c1>1 THEN INTO insert_all_test1 VALUES(1,'a') SELECT c2
FROM insert_all_test0;
ERROR:  Column "c1" does not exist.
LINE 1: INSERT ALL WHEN c1>1 THEN INTO insert_all_test1 VALUES(1,'a'...
                               ^
```


版权所有 © 华为云计算技术有限公司

```
HINT:  There is a column named "c1" in table "insert_all_test1", but it cannot be referenced from this part of the query.

--不支持为子查询表设置别名。
insertall_test_db=# INSERT ALL INTO insert_all_test1 VALUES(t1.c1, 'a') SELECT * FROM insert_all_test0 t1;
ERROR:  Missing FROM-clause entry for table "t1"
LINE 2: INSERT ALL INTO insert_all_test1 VALUES(t1.c1, 'a')

--不支持插入数据到视图。
insertall_test_db=# CREATE VIEW v1 as SELECT * FROM insert_all_test1;
insertall_test_db=# INSERT ALL INTO v1 SELECT * FROM insert_all_test0;
ERROR:  Not allowed to insert into view.

--子查询不存在。
insertall_test_db=# INSERT ALL INTO insert_all_test1 VALUES(1, 'a');
ERROR:  syntax error at or near ";"
LINE 2: INSERT ALL INTO insert_all_test1 VALUES(1, 'a');

--删除表。
insertall_test_db=# DROP VIEW v1;
insertall_test_db=# DROP TABLE insert_all_test0;
insertall_test_db=# DROP TABLE insert_all_test1 CASCADE;
insertall_test_db=# DROP TABLE insert_all_test2;
insertall_test_db=# DROP TABLE insert_all_test3;

insertall_test_db=# \c postgres;
gaussdb=# DROP DATABASE insertall_test_db;
```

1.13.15 L

1.13.15.1 LOAD DATA

功能描述

将文件中的数据导入到数据库指定表中。

注意事项

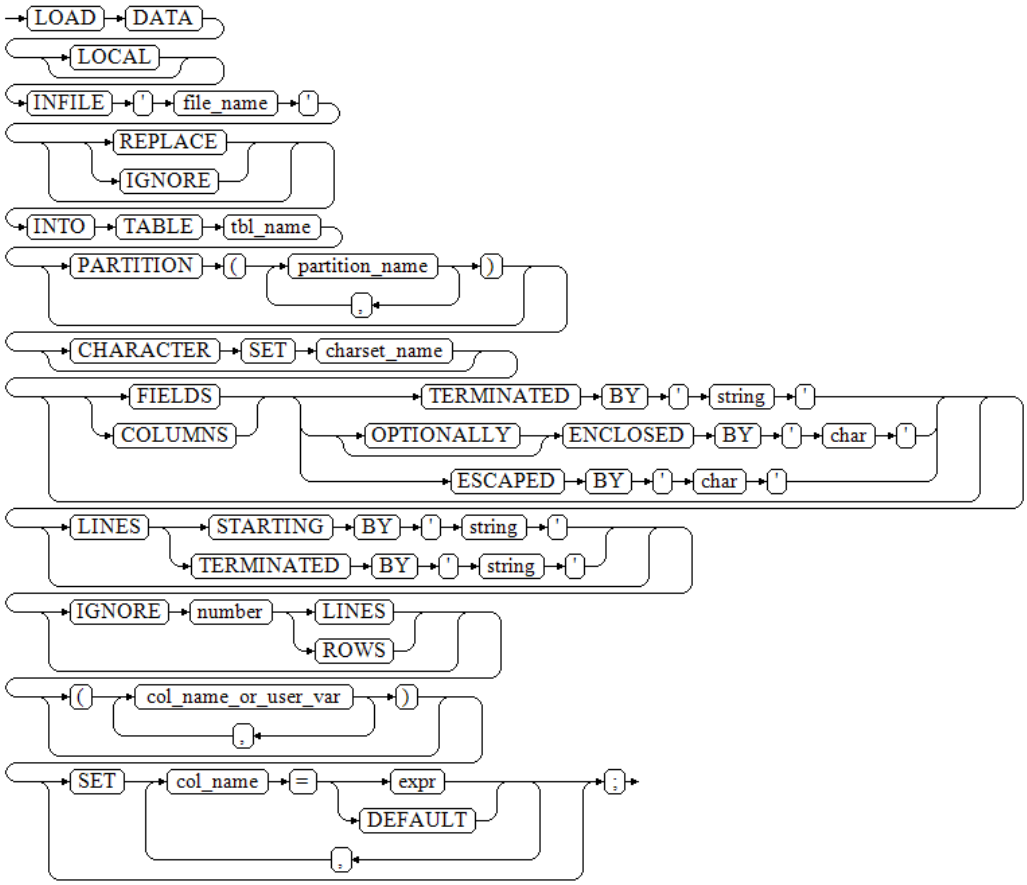
- 仅在MYSQL兼容模式（sql_compatibility = 'MYSQL'）下且开启b_format_version='5.7'和b_format_dev_version='s2'参数后可正常使用该语法。
- 当在其他兼容模式下或未设置兼容性参数（b_format_version='5.7'和b_format_dev_version='s2'）时会返回若干条语句，其主要用于gs_loader工具内部使用。
- **原生语法**仅供gs_loader工具内部调用，不建议用户直接使用，建议用户按**推荐语法**进行使用。
- LOAD DATA语法权限相关GUC参数与COPY FROM语法**注意事项**一致。
- LOAD DATA语法需要拥有表的INSERT和DELETE权限。
- 执行LOAD DATA语法写入表中的数据若无法转换为表中数据类型格式时，将导致导入失败。
- LOAD DATA只能用于表，不能用于视图。

推荐语法

```
LOAD DATA [LOCAL] INFILE 'file_name'
    [REPLACE | IGNORE]
    INTO TABLE tbl_name
    [PARTITION (partition_name [, ...])]
```


版权所有 © 华为云计算技术有限公司

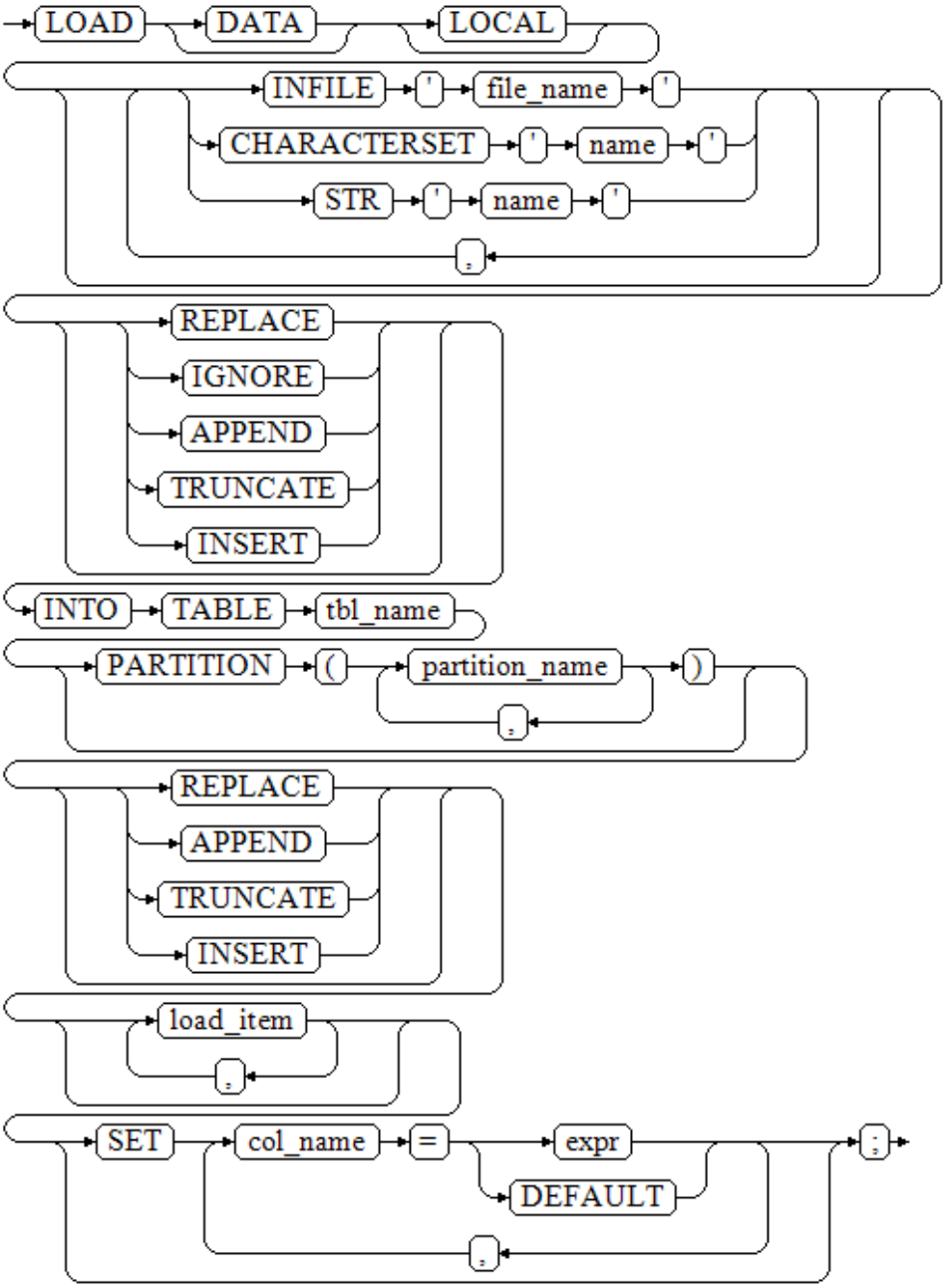
```
    [CHARACTER SET charset_name]
    [{FIELDS | COLUMNS} {TERMINATED BY 'string' | [OPTIONALLY] ENCLOSED BY 'char' | ESCAPED BY
'char'}]
    [LINES {STARTING BY 'string' | TERMINATED BY 'string'}]
    [IGNORE number {LINES | ROWS}]
    [(col_name_or_user_var [, ...])]
    [SET {col_name={expr | DEFAULT}} [, ...]];
```



原生语法

```
LOAD [DATA] [LOCAL]
    [{INFILE 'file_name' | CHARACTERSET 'name' | STR 'name' } [,...]]
    [REPLACE | IGNORE | APPEND | TRUNCATE | INSERT ]
    INTO TABLE tbl_name
    [PARTITION (partition_name [, ...])]
 [REPLACE | APPEND | TRUNCATE | INSERT ]
    [load_item [,...]]
    [SET {col_name={expr | DEFAULT}} [,...]];
```

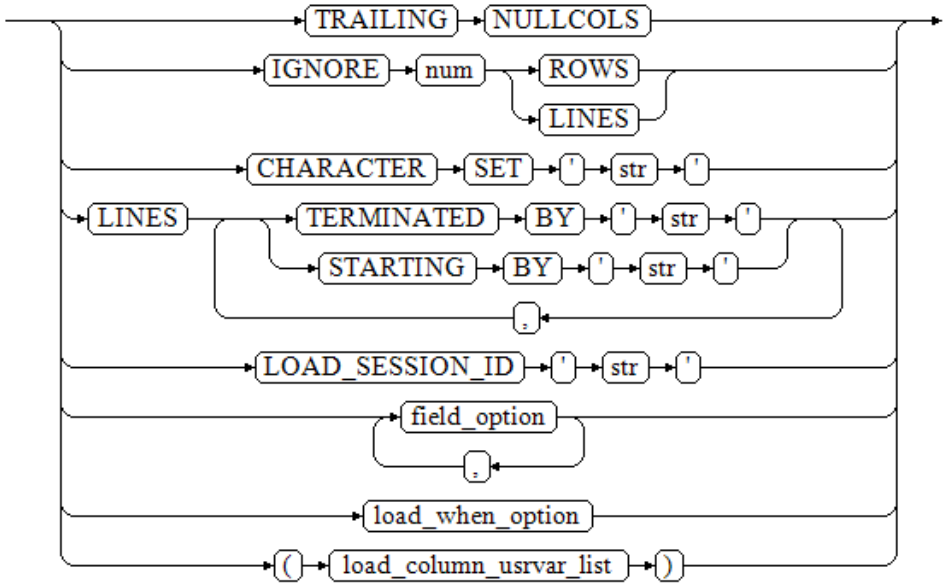

版权所有 © 华为云计算技术有限公司



- 其中load_item为导入文件的具体选项，其语法格式为：

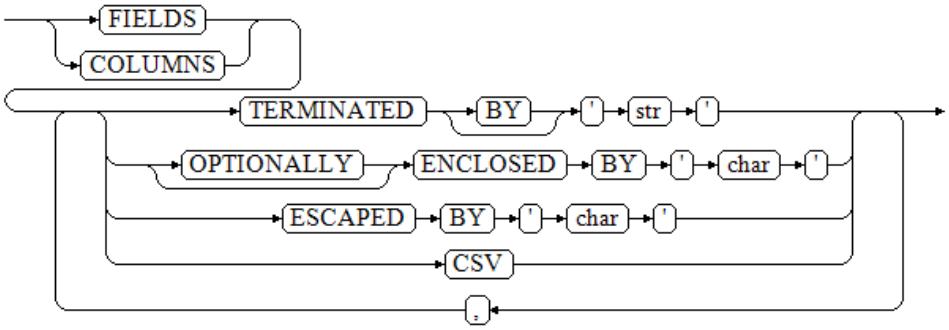
```
{
  TRAILING NULLCOLS | IGNORE num {ROWS | LINES} | CHARACTER SET 'str'|
  LINES {{TERMINATED BY 'str' | STARTING BY 'str'} [, ...]} |
  LOAD_SESSION_ID 'str' |
  {field_option [,...]} |
  load_when_option |
  ( load_column_usrvar_list )
}
```


版权所有 © 华为云计算技术有限公司



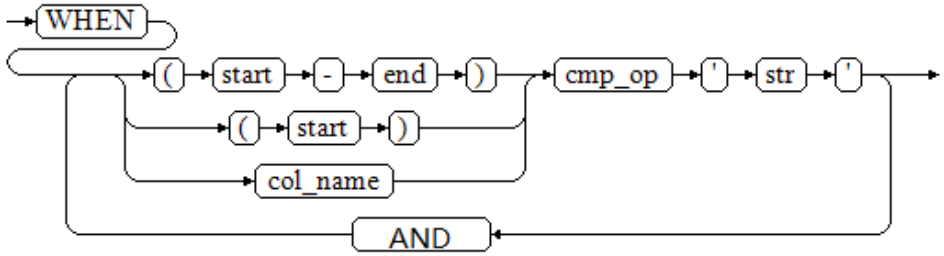
- 其中filed_option为导入文件的格式相关选项，其语法格式为：

```
{
 {FIELDS | COLUMNS}
 {{TERMINATED [BY] 'str' | [OPTIONALLY] ENCLOSED BY 'char' | ESCAPED BY 'char' | CSV } [, ...]}
}
```



- 其中load_when_option为导入文件时使用的具体过滤选项，其语法格式为：

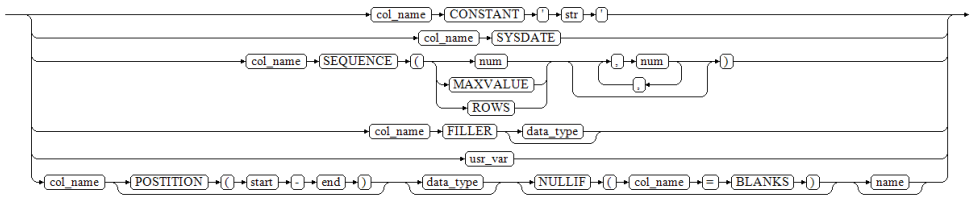
```
{
 WHEN
 {{{( start - end ) | (start) | col_name } cmp_op 'str' }[AND ...]}
}
```



- 其中load_column_usrvar_list为在导入数据时，对于列进行赋值处理、生成序列值、过滤等操作的语法选项，其语法格式为：

```
{
 col_name CONSTANT 'str' | col_name SYSDATE |
 col_name SEQUENCE ( {num | MAXVALUE | ROWS} [ {, num} [, ...]] ) |
 col_name FILLER [data_type] | usr_var |
 col_name [ POSTITION ( start - end ) ] [ data_type ] [ NULLIF ( col_name = BLANKS )] [ name ]
}
```


版权所有 © 华为云计算技术有限公司



说明

当开启**disable_unsupported_copy_load_options**时，若使用**原生语法**选项将会报错。

参数说明

- **LOCAL**

  指定导入文件的位置。

  不指定LOCAL时，若file_name为相对路径，则默认导入路径为数据目录。

  若指定LOCAL参数则需要指定file_name为绝对路径，当指定为相对路径时默认导入路径为数据库二进制所在路径，即$GAUSSHOME/bin/。

- **REPLACE | IGNORE**

  控制导入文件数据冲突及错误处理的行为选项。

  - 指定REPLACE选项

    当导入数据与表中原有数据冲突时，替换冲突行数据；当文件中字段数小于指定表的字段数时，宽松模式下告警，严格模式下报错。

  - 指定IGNORE选项

    当导入数据与表中原有数据冲突时，跳过冲突行数据；当文件中字段数小于指定表的字段数时，按默认值填充表中未指定的字段，并产生告警。

  - 不指定REPLACE与IGNORE选项

    - 不指定LOCAL选项时，当导入数据与表中原有数据冲突时报错；当文件中字段数小于指定表的字段数时，宽松模式下告警，严格模式下报错。
    - 指定LOCAL选项时，行为与指定IGNORE选项一致。

- **PARTITION**

  当导入表为分区表时，此参数用来指定分区。若数据与指定分区范围不一致，则报错。

- **CHARACTER SET**

  指定数据文件的编码格式名称，缺省为当前客户端编码格式。

- **FIELDS | COLUMNS**

  - **TERMINATED BY**

    指定两列之间分隔符，缺省为“\t”。

    说明

    指定换行符不能与分隔符相同。

  - **[OPTIONALLY] ENCLOSED BY**

    指定引号字符，缺省为''。

    OPTIONALLY参数为可选参数，无实际作用。

    引号符仅支持单字符，不支持字符串。

版权所有 © 华为云计算技术有限公司

  - **ESCAPED BY**

    指定转义符，缺省为“\”。

    说明

    - 转义字符仅支持单字符，不支持字符串。
    - 如果转义符非空，当处理到转义符时，会判断转义符及其后的字符序列是否满足指定的转义序列。如果满足指定的转义序列，则将转义符及其后的字符转义成指定字符。如果不满足指定的转义序列，则会忽略该转义符并继续导入数据。支持的转义序列受GUC参数**load_data_behavior_compat_options**是否配置new_escape_string选项控制。

- **LINES**
  - **STARTING BY**

    指定导入数据文件起始字段样式。
  - **TERMINATED BY**

    指定导入数据文件换行符样式。

    说明

    GUC参数**load_data_behavior_compat_options**的配置选项new_escape_string，控制是否对指定的行终结符做转义处理。
- **IGNORE**

  指定数据导入时，跳过数据文件的前number行。
- **col_name_or_user_var**

  可选的待复制字段列表。

  取值范围：如果没有声明字段列表，将使用所有字段。

  说明

  - 指定列参数不支持重复指定列。
  - LOAD DATA语法指定列时，col_name_or_user_var仅支持指定为表中存在列，不支持指定为用户变量。
- **SET**

  指定列值，可以指定为表达式或DEFAULT。

  说明

  - 表达式中不支持列名。
  - 若表达式结果类型与被赋值列对应类型之间不存在隐式转换函数则报错。

示例

```
-- 环境准备。
gaussdb=# CREATE DATABASE test_load WITH DBCOMPATIBILITY = 'MYSQL';
gaussdb=# \c test_load
test_load=# SET b_format_version='5.7';
test_load=# SET b_format_dev_version='s2';

-- 准备待导入的文件。
test_load=# CREATE TABLE load_data_tmp1(a int, b int);
test_load=# INSERT INTO load_data_tmp1 VALUES(1,1),(2,2),(3,3);
test_load=# \copy load_data_tmp1 to '/home/omm/load1.csv';
test_load=# CREATE TABLE load_data_tmp2(a int, b int);
test_load=# INSERT INTO load_data_tmp2 VALUES(1,2);
test_load=# \copy load_data_tmp2 to '/home/omm/load2.csv';
```


版权所有 © 华为云计算技术有限公司

```
test_load=# CREATE TABLE load_data_tmp3(a int, b int);
test_load=# INSERT INTO load_data_tmp3 VALUES(4,4),(5,5);
test_load=# \copy load_data_tmp3 to '/home/omm/load3.csv';
test_load=# CREATE TABLE load_data_tmp4(a char(50));
test_load=# INSERT INTO load_data_tmp4 VALUES('"load test quote"'), ('\'load test single_quote\'');
test_load=# \copy load_data_tmp4 to '/home/omm/load4.csv';

-- 创建表。
test_load=# CREATE TABLE load_data_tbl1(load_col1 INT UNIQUE, load_col2 INT, load_col3 CHAR(10));

-- 向表中插入一条数据。
test_load=# INSERT INTO load_data_tbl1 VALUES(0,0,'load0');

-- 从文件/home/omm/load1.csv中复制数据到load_data_tbl表,指定列名，设置.load_col3列值统一为"load"。
test_load=# LOAD DATA INFILE '/home/omm/load1.csv' INTO TABLE load_data_tbl1(load_col1, load_col2)
SET load_col3 = 'load';
--后面导入数据load_col3列值均为'load'
test_load=# SELECT * FROM load_data_tbl1 ORDER BY load_col1;
load_col1 | load_col2 | load_col3
-----------+-----------+-----------
        0 |         0 | load0
        1 |         1 | load
        2 |         2 | load
        3 |         3 | load
(4 rows)

-- 从文件/home/omm/load2.csv中复制数据到load_data_tbl表，指定IGNORE忽略冲突。
test_load=# LOAD DATA INFILE '/home/omm/load2.csv' IGNORE INTO TABLE load_data_tbl1;
-- 表load_data_tbl1中数据不变，冲突数据跳过。
test_load=# SELECT * FROM load_data_tbl1 ORDER BY load_col1;
load_col1 | load_col2 | load_col3
-----------+-----------+-----------
        0 |         0 | load0
        1 |         1 | load
        2 |         2 | load
        3 |         3 | load
(4 rows)

-- 创建分区表。
test_load=# CREATE TABLE load_data_tbl2
(
    load_col_col1 INT,
    load_col_col2 INT
) PARTITION BY RANGE (load_col_col2)
(
    PARTITION load_p1 VALUES LESS THAN(3),
    PARTITION load_p2 VALUES LESS THAN(9),
    PARTITION load_p3 VALUES LESS THAN(MAXVALUE)
);

-- 从文件/home/omm/load3.csv中复制数据到load_data_tbl2表，指定PARTITION。
test_load=# LOAD DATA INFILE '/home/omm/load3.csv' INTO TABLE load_data_tbl2 PARTITION (load_p2);
--数据导入到load_data_tbl2表中指定分区
test_load=# SELECT * FROM load_data_tbl2;
 load_col_col1 | load_col_col2
---------------+---------------
             4 |             4
             5 |             5
(2 rows)

-- 创建表。
test_load=# CREATE TABLE load_data_tbl3(load_col_col1 CHAR(30));

-- 从文件/home/omm/load4.csv中复制数据到load_data_tbl3表，指定FIELDS ENCLOSED BY。
test_load=# LOAD DATA INFILE '/home/omm/load4.csv' INTO TABLE load_data_tbl3 FIELDS ENCLOSED BY
'"';
-- 数据"load test quote"双引号被去掉，'load test single_quote'单引号保留。
test_load=# SELECT * from load_data_tbl3;
        load_col_col1
```


版权所有 © 华为云计算技术有限公司

```
--------------------------------
 load test quote
 'load test single_quote'
(2 rows)
-- 删除表。
test_load=# DROP TABLE load_data_tmp1;
test_load=# DROP TABLE load_data_tmp2;
test_load=# DROP TABLE load_data_tmp3;
test_load=# DROP TABLE load_data_tmp4;
test_load=# DROP TABLE load_data_tbl1;
test_load=# DROP TABLE load_data_tbl2;
test_load=# DROP TABLE load_data_tbl3;
```

1.13.15.2 LOCK

功能描述

LOCK TABLE命令用于获取表级锁。

GaussDB在为一个引用了表的命令自动请求锁时，尽可能选择最小限制的锁模式。如果用户需要一种更为严格的锁模式，可以使用LOCK命令。例如，一个应用是在Read Committed隔离级别上运行事务，并且它需要保证表中的数据在事务的运行过程中不被修改。为实现这个目的，可以在查询之前对表使用SHARE锁模式进行锁定。这样可以防止数据被并发修改，从而保证后续的查询可以读到已提交的持久化的数据。因为SHARE锁模式与任何写操作需要的ROW EXCLUSIVE模式冲突，并且LOCK TABLE name IN SHARE MODE语句将等到所有当前持有ROW EXCLUSIVE模式锁的事务提交或回滚后才能执行。因此，一旦获得该锁，就不会存在未提交的写操作，并且其他操作也只能等到该锁释放之后才能开始。

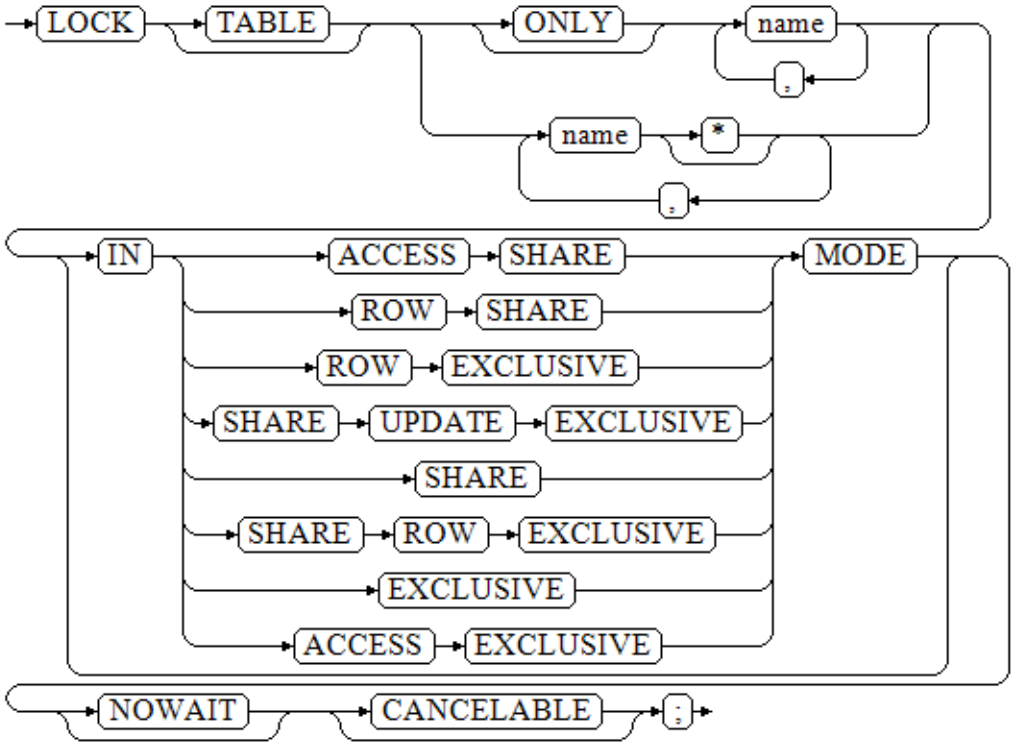
注意事项

- LOCK TABLE只能在一个事务块的内部有用，因为锁在事务结束时就会被释放。出现在任意事务块外面的LOCK TABLE都会报错。
- 如果没有声明锁模式，缺省为最严格的模式ACCESS EXCLUSIVE。
- LOCK TABLE ... IN ACCESS SHARE MODE需要在目标表上有SELECT权限。所有其他形式的LOCK需要UPDATE或DELETE权限。
- 没有UNLOCK TABLE命令，锁总是在事务结束时释放。
- LOCK TABLE只处理表级的锁，因此那些带“ROW”字样的锁模式都是有歧义的。这些模式名称通常可理解为用户试图在一个被锁定的表中获取行级的锁。同样，ROW EXCLUSIVE模式也是一个可共享的表级锁。注意，只要是涉及到LOCK TABLE，所有锁模式都有相同的语意，区别仅在于规则中锁与锁之间是否冲突，规则请参见**表1-404**。
- 如果没有打开xc_maintenance_mode参数，那么对系统表申请ACCESS EXCLUSIVE级别锁将报错。
- 自动CANCEL业务接口只允许重分布工具使用。

语法格式

```
LOCK [ TABLE ] {[ ONLY ] name [, ...]| {name [ * ]} [, ...]}
    [ IN {ACCESS SHARE | ROW SHARE | ROW EXCLUSIVE | SHARE UPDATE EXCLUSIVE | SHARE | SHARE ROW EXCLUSIVE | EXCLUSIVE | ACCESS EXCLUSIVE} MODE ]
    [ NOWAIT ][CANCELABLE];
```


版权所有 © 华为云计算技术有限公司



参数说明

表 1-404 冲突的锁模式

请求的锁模式/当前锁模式	ACCESS SHARE	ROW SHARE	ROW EXCLUSIVE	SHARE UPDATE EXCLUSIVE	SHARE	SHARE ROW EXCLUSIVE	EXCLUSIVE	ACCESS EXCLUSIVE
ACCESS SHARE	-	-	-	-	-	-	-	X
ROW SHARE	-	-	-	-	-	-	X	X
ROW EXCLUSIVE	-	-	-	-	X	X	X	X
SHARE UPDATE EXCLUSIVE	-	-	-	X	X	X	X	X
SHARE	-	-	X	X	-	X	X	X

版权所有 © 华为云计算技术有限公司

请求的锁模式/当前锁模式	ACCESS SHARE	ROW SHARE	ROW EXCLUSIVE	SHARE UPDATE EXCLUSIVE	SHARE	SHARE ROW EXCLUSIVE	EXCLUSIVE	ACCESS EXCLUSIVE
SHARE ROW EXCLUSIVE	-	-	X	X	X	X	X	X
EXCLUSIVE	-	X	X	X	X	X	X	X
ACCESS EXCLUSIVE	X	X	X	X	X	X	X	X

LOCK的参数说明如下所示：

- **name**

  要锁定的表的名称，可以有模式修饰。

  LOCK TABLE命令中声明的表的顺序就是上锁的顺序。

  取值范围：已存在的表名。

  说明

  支持使用DATABASE LINK方式对远端表进行操作，使用方式详情请见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

- **ONLY**

  如果指定ONLY，只有该表被锁定。如果没有声明，该表和他的所有子表将都被锁定。

- **ACCESS SHARE**

  ACCESS SHARE锁只允许对表进行读取，而禁止对表进行修改。所有对表进行读取而不修改的SQL语句都会自动请求这种锁。例如，SELECT命令会自动在被引用的表上请求一个ACCESS SHARE锁。

- **ROW SHARE**

  ROW SHARE锁允许对表进行并发读取，禁止对表进行其他操作。

  SELECT FOR UPDATE和SELECT FOR SHARE命令会自动在目标表上请求ROW SHARE锁（且所有被引用但不是FOR SHARE/FOR UPDATE的其他表上，还会自动加上ACCESS SHARE锁）。

  对于分区表，SELECT FOR SHARE操作还会在DN上获取partition对象的ROW EXCLUSIVE锁进行并发控制。

- **ROW EXCLUSIVE**

  ROW EXCLUSIVE锁与ROW SHARE锁相同，允许并发读取表，但是禁止修改表中数据。UPDATE、DELETE、INSERT命令会自动在目标表上请求这个锁（且所有被

版权所有 © 华为云计算技术有限公司

引用的其他表上还会自动加上的ACCESS SHARE锁）。通常情况下，所有会修改表数据的命令都会请求表的ROW EXCLUSIVE锁。

- **SHARE UPDATE EXCLUSIVE**

  这个模式保护一个表的模式不被并发修改，以及禁止在目标表上执行垃圾回收命令（VACUUM）。

  VACUUM（不带FULL选项）、ANALYZE、CREATE INDEX CONCURRENTLY命令会自动请求这样的锁。

- **SHARE**

  SHARE锁允许并发的查询，但是禁止对表进行修改。

  CREATE INDEX（不带CONCURRENTLY）语句会自动请求这种锁。

- **SHARE ROW EXCLUSIVE**

  SHARE ROW EXCLUSIVE锁禁止对表进行任何的并发修改，而且是独占锁，因此一个会话中只能获取一次。

  任何SQL语句都不会自动请求这个锁模式。

- **EXCLUSIVE**

  EXCLUSIVE锁允许对目标表进行并发查询，但是禁止任何其他操作。

  这个模式只允许并发加ACCESS SHARE锁，也就是说，只有对表的读动作可以和持有这个锁模式的事务并发执行。

  任何SQL语句都不会在用户表上自动请求这个锁模式。然而在某些操作的时候，会在某些系统表上请求它。

- **ACCESS EXCLUSIVE**

  这个模式保证其所有者（事务）是可以访问该表的唯一事务。

  ALTER TABLE、DROP TABLE、TRUNCATE、REINDEX、CLUSTER、VACUUM FULL命令会自动请求这种锁。

  在LOCK TABLE命令没有明确声明需要的锁模式时，它是缺省锁模式。

- **NOWAIT**

  声明LOCK TABLE不去等待任何冲突的锁释放，如果无法立即获取该锁，该命令退出并且发出一个错误信息。

  在不指定NOWAIT的情况下获取表级锁时，如果有其他互斥锁存在的话，则等待其他锁的释放。

- **CANCELABLE**

  通过指定该参数允许等锁线程给持锁线程和等锁线程发送CANCEL信号。

  只允许重分布工具使用，其它用户使用将报错。

示例

- SHARE ROW EXCLUSIVE

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA tpcds;

--创建表tpcds.reason。
gaussdb=# CREATE TABLE tpcds.reason (
r_reason_sk         INTEGER       NOT NULL,
r_reason_id         CHAR(16)      NOT NULL,
r_reason_desc       INTEGER
);

--向表中插入多条记录。
gaussdb=# INSERT INTO tpcds.reason VALUES (1, 'AAAAAAAABAAAAAAA', '18'),(5,
```


版权所有 © 华为云计算技术有限公司

```
'AAAAAAAACAAAAAAA', '362'),(7, 'AAAAAAAADAAAAAAA', '585');

--创建一个新表reason_t1。
gaussdb=# CREATE TABLE tpcds.reason_t1 AS TABLE tpcds.reason;

--开启事务。
gaussdb=# START TRANSACTION;

--使用SHARE ROW EXCLUSIVE MODE锁模式锁表。
gaussdb=# LOCK TABLE tpcds.reason_t1 IN SHARE ROW EXCLUSIVE MODE;

--在另一个终端执行DELETE操作会发现阻塞。
gaussdb=# DELETE FROM tpcds.reason_t1 WHERE r_reason_sk = 7;

--结束事务，释放锁。
gaussdb=# COMMIT;
```

- ROW EXCLUSIVE

```
--开启事务。
gaussdb=# START TRANSACTION;

--执行update语句后，在被引用的表上请求一个ROW EXCLUSIVE锁。
gaussdb=# UPDATE tpcds.reason_t1 SET r_reason_desc=180 WHERE r_reason_sk=1;

--在另外一个终端alter table语句，出现阻塞。
gaussdb=# ALTER TABLE tpcds.reason_t1 ADD r_reason_asc int;

--结束事务，释放锁。
gaussdb=# COMMIT;
```

- ACCESS EXCLUSIVE

```
--开启事务。
gaussdb=# START TRANSACTION;

--执行TRUNCATE语句后，在被引用的表上请求一个ACCESS SHARE锁。
gaussdb=# TRUNCATE tpcds.reason_t1;

--在另一个终端执行select语句，出现阻塞。
gaussdb=# SELECT * FROM tpcds.reason_t1;

--结束事务，释放锁。
gaussdb=# COMMIT;

--删除表tpcds.reason_t1。
gaussdb=# DROP TABLE tpcds.reason_t1;

--删除表tpcds.reason。
gaussdb=# DROP TABLE tpcds.reason;

--删除SCHEMA。
gaussdb=# DROP SCHEMA tpcds CASCADE;
```

1.13.15.3 LOCK BUCKETS

功能描述

bucket粒度的锁。

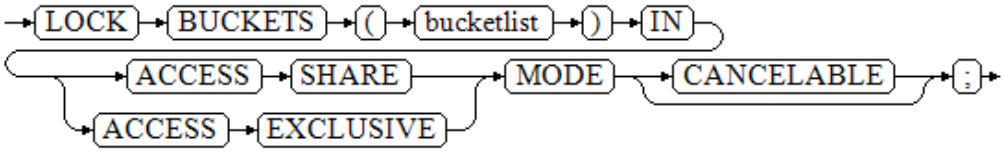
注意事项

- 仅支持在hashbucket扩容期间调用。
- LOCK BUCKETS只能在一个事务块的内部有用，因为锁在事务结束时就会被释放。出现在任意事务块外面的LOCK BUCKETS都会报错。

版权所有 © 华为云计算技术有限公司

语法格式

```
LOCK BUCKETS (bucketlist) IN {ACCESS SHARE|ACCESS EXCLUSIVE} MODE [CANCELABLE];
```



参数说明

LOCK BUCKETS的参数说明如下所示：

- **bucketlist**
  要锁定的bucket的列表。
  取值范围：对于每个bucket，取值范围为[0,1023]。
- **ACCESS SHARE**
  ACCESS锁只允许对bucket进行读取，禁止对bucket进行修改。
- **ACCESS EXCLUSIVE**
  这个模式保证其所有者（事务）是可以访问该bucket的唯一事务。
- **CANCELABLE**
  通过指定该参数允许等锁线程给持锁线程和等锁线程发送CANCEL信号。

示例

```
gaussdb=# START TRANSACTION;

gaussdb=# LOCK BUCKETS (0,1,2,3) IN ACCESS EXCLUSIVE MODE;

gaussdb=# COMMIT;
```

1.13.16 M

1.13.16.1 MARK BUCKETS

功能描述

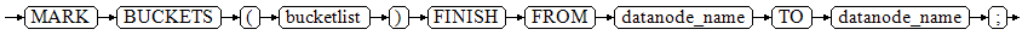
用于扩容工具通知内核哪些bucket已经完成搬迁。

注意事项

仅支持在hashbucket扩容期间调用。只支持管理员权限用户调用。

语法格式

```
MARK BUCKETS (bucketlist) FINISH FROM datanode_name TO datanode_name;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **bucketlist**

  已完成搬迁的bucket列表。

  取值范围：对于每个bucket，取值范围为[0,1023]。

- **datanode_name**

  搬迁bucket涉及的节点名称。

  取值范围：字符串，要符合**标识符命名规范**。

示例

```
gaussdb=# MARK BUCKETS (0,1,2,3) FINISH FROM datanode1 TO datanode3;
```

1.13.16.2 MERGE INTO

功能描述

通过MERGE INTO语句，将目标表和源表中的数据针对关联条件进行匹配，若关联条件匹配时对目标表进行UPDATE，无法匹配时对目标表执行INSERT。此语法可以很方便地用来合并执行UPDATE和INSERT，避免多次执行。

注意事项

- 进行MERGE INTO操作的用户需要同时拥有目标表的UPDATE和INSERT权限，以及源表的SELECT权限。
- 不支持重分布过程中MERGE INTO。
- 如果MERGE INTO操作的源表包含被动态数据脱敏的数据列，则执行过程中向目标表中插入或更新的结果即为脱敏后的值，无法被还原。
- MERGE INTO的STREAM计划不支持并发DELETE和UPDATE场景，其他类型的计划支持。
- MERGE INTO语句中不支持触发触发器。

语法格式

```
MERGE [/*+ plan_hint */] INTO table_name [ [ AS ] alias ]
USING { { table_name | view_name } | subquery } [ [ AS ] alias ]
ON ( condition )
[
  WHEN MATCHED THEN
  UPDATE SET { column_name = { expression  | DEFAULT } |
         ( column_name [, ...] ) = ( { expression  | DEFAULT } [, ...] ) } [, ...]
  [ WHERE condition ]
]
[
  WHEN NOT MATCHED THEN
  INSERT { DEFAULT VALUES |
  [ ( column_name [, ...] ) ] VALUES ( { expression  | DEFAULT } [, ...] ) [, ...] [ WHERE condition ] }
];
```

参数说明

- **plan_hint子句**

  以/*+ */的形式在MERGE关键字后，用于对MERGE对应的语句块生成的计划进行hint调优，详细用法请参见《SQL调优指南》中“使用Plan Hint进行调优”章

版权所有 © 华为云计算技术有限公司

节。每条语句中只有第一个/*+ plan_hint */注释块会作为hint生效，里面可以写多条hint。

- **INTO子句**

  指定正在更新或插入的目标表。目标表为复制表时，暂不支持目标表中某列默认值为volatile函数（如自增列），enable_stream_operator=off时目标表需要包含主键或带有unique not null。

  - **table_name**

    目标表的表名。

  - **alias**

    目标表的别名。

    取值范围：字符串，要符合**标识符命名规范**。

- **USING子句**

  指定源表，源表可以为表、视图或子查询。目标表为复制表时，暂不支持USING子句中包含非复制表。

  - **table_name | view_name**

    源表、视图的名称。

  - **subquery**

    一个子查询， 用于提供要合并到目标表中的行。

- **ON子句**

  condition

  关联条件，用于指定目标表和源表的关联条件。不支持更新关联条件中的字段。

- **WHEN MATCHED子句**

  当源表和目标表中数据针对关联条件可以匹配上时，选择WHEN MATCHED子句进行UPDATE操作。具体语法请参见**UPDATE**。

  不支持更新分布列。不支持更新系统表、系统列。

- **WHEN NOT MATCHED子句**

  当源表和目标表中数据针对关联条件无法匹配时，选择WHEN NOT MATCHED子句进行INSERT操作。具体语法请参见**INSERT**。

  不支持INSERT子句中包含多个VALUES。

  WHEN MATCHED和WHEN NOT MATCHED子句顺序可以交换，可以缺省其中一个，但不能同时缺省，不支持同时指定两个WHEN MATCHED或WHEN NOT MATCHED子句。

- **DEFAULT**

  用对应字段的缺省值填充该字段。

  如果没有缺省值，则为NULL。

- **WHERE condition**

  UPDATE子句和INSERT子句的条件，只有在条件满足时才进行更新操作，可缺省。不支持WHERE条件中引用系统列。不建议使用int等数值类型作为condition，因为int等数值类型可以隐式转换为bool值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

示例

```
-- 创建目标表products和源表newproducts，并插入数据
gaussdb=# CREATE TABLE products
```


版权所有 © 华为云计算技术有限公司

```
(
product_id INTEGER,
product_name VARCHAR2(60),
category VARCHAR2(60)
);

gaussdb=# INSERT INTO products VALUES (1501, 'vivitar 35mm', 'electrncs');
gaussdb=# INSERT INTO products VALUES (1502, 'olympus is50', 'electrncs');
gaussdb=# INSERT INTO products VALUES (1600, 'play gym', 'toys');
gaussdb=# INSERT INTO products VALUES (1601, 'lamaze', 'toys');
gaussdb=# INSERT INTO products VALUES (1666, 'harry potter', 'dvd');

gaussdb=# CREATE TABLE newproducts
(
product_id INTEGER,
product_name VARCHAR2(60),
category VARCHAR2(60)
);

gaussdb=# INSERT INTO newproducts VALUES (1502, 'olympus camera', 'electrncs');
gaussdb=# INSERT INTO newproducts VALUES (1601, 'lamaze', 'toys');
gaussdb=# INSERT INTO newproducts VALUES (1666, 'harry potter', 'toys');
gaussdb=# INSERT INTO newproducts VALUES (1700, 'wait interface', 'books');

-- 进行MERGE INTO操作
gaussdb=# MERGE INTO products p
USING newproducts np
ON (p.product_id = np.product_id)
WHEN MATCHED THEN
  UPDATE SET p.product_name = np.product_name, p.category = np.category WHERE p.product_name !=
'play gym'
WHEN NOT MATCHED THEN
  INSERT VALUES (np.product_id, np.product_name, np.category) WHERE np.category = 'books';
MERGE 4

-- 查询更新后的结果
gaussdb=# SELECT * FROM products ORDER BY product_id;
 product_id |  product_name  | category
------------+----------------+-----------
       1501 | vivitar 35mm   | electrncs
       1502 | olympus camera | electrncs
       1600 | play gym       | toys
       1601 | lamaze         | toys
       1666 | harry potter   | toys
       1700 | wait interface | books
(6 rows)

-- 删除表
gaussdb=# DROP TABLE products;
gaussdb=# DROP TABLE newproducts;
```

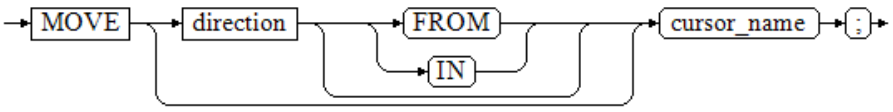
1.13.16.3 MOVE

功能描述

MOVE在不检索数据的情况下重新定位一个游标。MOVE的作用类似于**FETCH**命令，区别是MOVE只重定位游标而不返回行。

语法格式

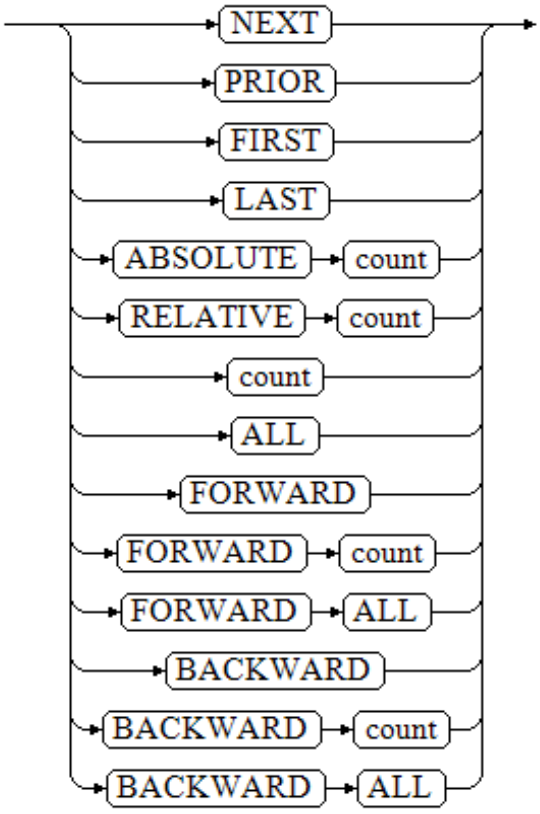
```
MOVE [ direction [ FROM | IN ] ] cursor_name;
```



版权所有 © 华为云计算技术有限公司

其中direction子句为可选参数。

```
{ NEXT
  | PRIOR
  | FIRST
  | LAST
  | ABSOLUTE count
  | RELATIVE count
  | count
  | ALL
  | FORWARD
  | FORWARD count
  | FORWARD ALL
  | BACKWARD
  | BACKWARD count
  | BACKWARD ALL }
```



参数说明

MOVE命令的参数与FETCH的相同，详细请参见**FETCH**章节中的**参数说明**。

说明

成功完成时，MOVE命令将返回一个“MOVE count”的标签，count是一个使用相同参数的FETCH命令会返回的行数（可能为零）。

示例

```
--建表并插入数据。
gaussdb=# CREATE TABLE tbl_test(c1 int);
gaussdb=# INSERT INTO tbl_test VALUES (generate_series(1,20));

--建立一个名为cursor1的游标。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# BEGIN;
gaussdb=# CURSOR cursor1 FOR SELECT * FROM tbl_test ORDER BY 1;

--使用MOVE命令使游标向后移动5行，不返回结果。
gaussdb=# MOVE FORWARD 5 FROM cursor1;
MOVE 5

--使用FETCH命令检索两行数据。
gaussdb=# FETCH FORWARD 2 FROM cursor1;
 c1
----
  6
  7
(2 rows)

--关闭游标并结束事务。
gaussdb=# CLOSE cursor1;
gaussdb=# END;

--删除。
gaussdb=# DROP TABLE tbl_test;
```

相关链接

CLOSE，**FETCH**

1.13.17 P

1.13.17.1 PREDICT BY

功能描述

利用完成训练的模型进行推测任务。

注意事项

当前形态暂不支持使用该语法。

1.13.17.2 PREPARE

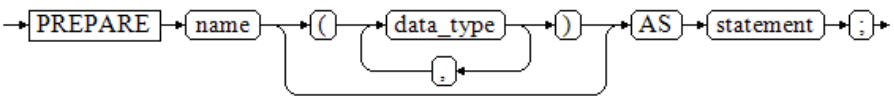
功能描述

创建一个预备语句。

预备语句是服务端的对象，可以用于优化性能。在执行PREPARE语句的时候，指定的查询被解析、分析、重写。当随后发出EXECUTE语句的时候，预备语句被规划和执行。这种设计避免了重复解析、分析工作。PREPARE语句创建后在整个数据库会话期间一直存在，一旦创建成功，即便是在事务块中创建，事务回滚，PREPARE也不会删除。只能通过显式调用**DEALLOCATE**进行删除，会话结束时，PREPARE也会自动删除。

语法格式

```
PREPARE name [ ( data_type [, ...] ) ] AS statement;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **name**

  预备语句的名称。必须在该会话中是唯一的。

- **data_type**

  参数的数据类型。

- **statement**

  SELECT、INSERT、UPDATE、DELETE、MERGE INTO或VALUES语句之一。

说明

PREPARE语句的解析过程中，所有参数默认均被解析为unknown数据类型。若解析涉及操作符或函数的选择，遵循文档中**操作符**和**函数**中对unknown类型参数的处理规范。

示例

请参见EXECUTE的**示例**。

相关链接

DEALLOCATE，**EXECUTE**

1.13.17.3 PREPARE TRANSACTION

功能描述

为当前事务做两阶段提交的准备。

在命令之后，事务就不再和当前会话关联了；它的状态完全保存在磁盘上，它被提交成功的可能性非常高，即使是在请求提交之前数据库发生了崩溃也如此。

一旦准备好了，一个事务就可以在稍后用**COMMIT PREPARED**或**ROLLBACK PREPARED**命令分别进行提交或者回滚。这些命令可以从任何会话中发出，而不光是最初执行事务的会话。

从发出命令的会话的角度来看，PREPARE TRANSACTION不同于ROLLBACK：在执行它之后，就不再有活跃的当前事务了，并且预备事务的效果无法见到 (在事务提交的时候其效果会再次可见)。

如果PREPARE TRANSACTION因为某些原因失败，那么它就会变成一个ROLLBACK，当前事务被取消。

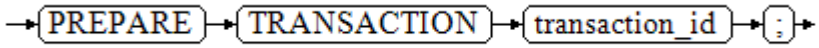
注意事项

- 事务功能由数据库自动维护，不应显式使用事务功能。
- 分布式当前不支持客户调用自定义PREPARE TRANSACTION操作，保留该语法仅用于维护模式（GUC参数xc_maintenance_mode为on时）下CN本地二阶段清理使用，不保证全局事务能力。
- 在运行PREPARE TRANSACTION命令时，必须在gaussdb.conf配置文件中增大max_prepared_transactions的数值。建议至少将其设置为等于max_connections，这样每个会话都可以有一个等待中的预备事务。

版权所有 © 华为云计算技术有限公司

语法格式

```
PREPARE TRANSACTION transaction_id;
```



参数说明

transaction_id

待提交事务的标识符，用于后面在COMMIT PREPARED或ROLLBACK PREPARED的时候标识这个事务。它不能和任何当前预备事务已经使用了的标识符同名。

取值范围：标识符必须以字符串文本的方式书写，并且必须小于200字节长。

示例

```
--维护模式下开始一个事务。
gaussdb=# SET XC_MAINTENANCE_MODE = ON;
SET

gaussdb=# BEGIN;
BEGIN

--准备标识符为的trans_test的事务。
gaussdb=# PREPARE TRANSACTION 'trans_test';
PREPARE TRANSACTION

--取消标识符为的trans_test的事务。
gaussdb=# ROLLBACK PREPARED 'trans_test';
ROLLBACK PREPARED
```

相关链接

COMMIT PREPARED，**ROLLBACK PREPARED**

1.13.17.4 PURGE

功能描述

使用PURGE语句可以实现如下功能：

- 从回收站中清理表或索引，并释放对象相关的全部空间。
- 清理回收站。

注意事项

- 清除（PURGE）操作支持：表（PURGE TABLE）、索引（PURGE INDEX）、回收站（PURGE RECYCLEBIN）。
- 执行PURGE操作的权限要求如下：
  - PURGE TABLE：用户必须是表的所有者，且用户必须拥有表所在模式的USAGE权限，当三权分立开关关闭时，系统管理员默认拥有此权限。
  - PURGE INDEX：用户必须是索引的所有者，用户必须拥有索引所在模式的USAGE权限，当三权分立开关关闭时，系统管理员默认拥有此权限。
  - PURGE RECYCLEBIN：具有系统管理员权限或SYSDBA权限的用户可以执行该操作，普通用户无此权限。普通用户只能清理回收站中当前用户拥有的对

版权所有 © 华为云计算技术有限公司

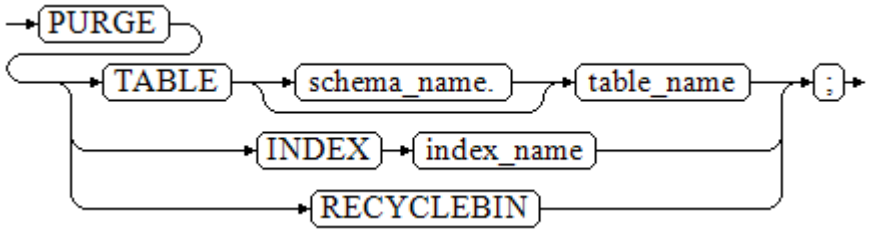
象，且用户必须拥有对象所在模式的USAGE权限，当三权分立开关关闭时，系统管理员默认可以清理回收站所有对象。

前提条件

- 开启enable_recyclebin参数，启用回收站，参数使用请联系管理员处理。
- recyclebin_retention_time参数用于设置回收站对象保留时间，超过该时间的回收站对象将被自动清理，参数使用请联系管理员处理。

语法格式

```
PURGE { TABLE [schema_name.]table_name
      | INDEX index_name
      | RECYCLEBIN
    };
```



参数说明

- **schema_name**

  模式名。
- **TABLE [schema_name.] table_name**

  清空回收站中指定的表，可用模式名修饰。
- **INDEX index_name**

  清空回收站中指定的索引。
- **RECYCLEBIN**

  清空回收站中的所有对象。

示例

```
--创建表reason_t1。
gaussdb=# CREATE TABLE reason_t1(
    r_reason_sk    integer,
    r_reason_id    character(16),
    r_reason_desc  character(100)
) WITH(STORAGE_TYPE = ustore);

--创建表reason_t2。
gaussdb=# CREATE TABLE reason_t2(
    r_reason_sk    integer,
    r_reason_id    character(16),
    r_reason_desc  character(100)
) WITH(STORAGE_TYPE = ustore);

-- 对表reason_t1和reason_t2添加索引。
gaussdb=#  CREATE INDEX idx_t1 on reason_t1(r_reason_id);
gaussdb=#  CREATE INDEX idx_t2 on reason_t2(r_reason_id);
gaussdb=#  DROP TABLE reason_t1;
gaussdb=#  DROP TABLE reason_t2;
```


版权所有 © 华为云计算技术有限公司

```
--查看回收站。
gaussdb=#  SELECT rcyname,rcyoriginname,rcytablespace FROM GS_RECYCLEBIN;
          rcyname           | rcyoriginname | rcytablespace
----------------------------+---------------+---------------
 BIN$31C94EB4207$8001$0==$0 | reason_t1     |             0
 BIN$31C94EB420D$8001$0==$0 | idx_t1        |             0
 BIN$31C94EB420A$8004$0==$0 | reason_t2     |             0
 BIN$31C94EB420E$8004$0==$0 | idx_t2        |             0
(4 rows)

--PURGE清除表。
gaussdb=#  PURGE TABLE reason_t1;
gaussdb=#  SELECT rcyname,rcyoriginname,rcytablespace FROM GS_RECYCLEBIN;
          rcyname           | rcyoriginname | rcytablespace
----------------------------+---------------+---------------
 BIN$31C94EB420A$8004$0==$0 | reason_t2     |             0
 BIN$31C94EB420E$8004$0==$0 | idx_t2        |             0
(2 rows)

--PURG清除索引。
gaussdb=# PURGE INDEX idx_t2;
gaussdb=# SELECT rcyname,rcyoriginname,rcytablespace FROM GS_RECYCLEBIN;
          rcyname           | rcyoriginname | rcytablespace
----------------------------+---------------+---------------
 BIN$31C94EB420A$8004$0==$0 | reason_t2     |             0
(1 row)

--PURGE清除回收站所有对象。
gaussdb=#  PURGE recyclebin;
gaussdb=#  SELECT rcyname,rcyoriginname,rcytablespace FROM GS_RECYCLEBIN;
         rcyname          | rcyoriginname | rcytablespace
--------------------------+---------------+---------------
(0 rows)
```

1.13.18 R

1.13.18.1 REASSIGN OWNED

功能描述

修改数据库对象的所有者。

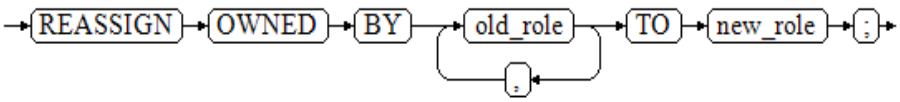
REASSIGN OWNED用于将旧角色的数据库对象所有者改为新角色。

注意事项

- REASSIGN OWNED常用于在删除角色之前的准备工作。
- 执行REASSIGN OWNED需要有原角色和目标角色上的权限。

语法格式

```
REASSIGN OWNED BY old_role [, ...] TO new_role;
```



参数说明

- **old_role**

版权所有 © 华为云计算技术有限公司

旧属主的角色名。

- **new_role**

  将要成为这些对象属主的新角色的名称。注意：只有初始用户才可以通过REASSIGN OWNED语法将属主修改为初始化用户。

示例

```
--创建test_jim和test_tom用户。
gaussdb=# CREATE USER test_jim PASSWORD '********';
gaussdb=# CREATE USER test_tom PASSWORD '********';

--查看自动创建的模式的拥有者为同名的用户。
gaussdb=# \dn test*
   List of schemas
   Name   |  Owner
----------+----------
 test_jim | test_jim
 test_tom | test_tom
(2 rows)

--将所有test_jim拥有的数据库对象的属主更改为test_tom。
gaussdb=# REASSIGN OWNED BY test_jim TO test_tom;

--查看schema的信息，test_jim模式的属主更换为test_tom。
gaussdb=# \dn test*
   List of schemas
   Name   |  Owner
----------+----------
 test_jim | test_tom
 test_tom | test_tom
(2 rows)

--删除test_jim和test_tom用户。
gaussdb=# DROP USER test_jim, test_tom CASCADE;
```

相关链接

DROP OWNED

1.13.18.2 REFRESH INCREMENTAL MATERIALIZED VIEW

功能描述

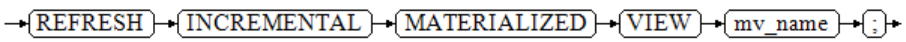
REFRESH INCREMENTAL MATERIALIZED VIEW会以增量刷新的方式对物化视图进行刷新。

注意事项

- 增量刷新仅支持增量物化视图。
- 刷新物化视图需要当前用户拥有基表的SELECT权限。

语法格式

```
REFRESH INCREMENTAL MATERIALIZED VIEW mv_name;
```



版权所有 © 华为云计算技术有限公司

参数说明

- **mv_name**
  要刷新的物化视图的名称。

示例

```
--创建一个普通表。
gaussdb=# CREATE TABLE my_table (c1 int, c2 int);

--创建增量物化视图。
gaussdb=# CREATE INCREMENTAL MATERIALIZED VIEW my_imv AS SELECT * FROM my_table;

--基表写入数据。
gaussdb=# INSERT INTO my_table VALUES(1,1),(2,2);

--对增量物化视图my_imv进行增量刷新。
gaussdb=# REFRESH INCREMENTAL MATERIALIZED VIEW my_imv;

--查看增量物化视图。
gaussdb=# SELECT * FROM my_imv;
 c1 | c2
----+----
  1 |  1
  2 |  2
(2 rows)

--删除增量物化视图。
gaussdb=# DROP MATERIALIZED VIEW my_imv;

--删除表my_table。
gaussdb=# DROP TABLE my_table;
```

相关链接

ALTER MATERIALIZED VIEW，**CREATE INCREMENTAL MATERIALIZED VIEW**，**CREATE MATERIALIZED VIEW**，**CREATE TABLE**，**DROP MATERIALIZED VIEW**，**REFRESH MATERIALIZED VIEW**

1.13.18.3 REFRESH MATERIALIZED VIEW

功能描述

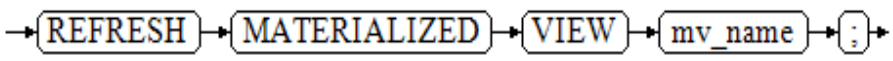
REFRESH MATERIALIZED VIEW会以全量刷新的方式对物化视图进行刷新。

注意事项

- 全量刷新既可以对全量物化视图执行，也可以对增量物化视图执行。
- 刷新物化视图需要当前用户拥有基表的SELECT权限。

语法格式

```
REFRESH MATERIALIZED VIEW mv_name;
```



参数说明

mv_name

版权所有 © 华为云计算技术有限公司

要刷新的物化视图的名称。

示例

```
--创建一个普通表。
gaussdb=# CREATE TABLE my_table (c1 int, c2 int);

--创建全量物化视图。
gaussdb=# CREATE MATERIALIZED VIEW my_mv AS SELECT * FROM my_table;

--创建增量物化视图。
gaussdb=# CREATE INCREMENTAL MATERIALIZED VIEW my_imv AS SELECT * FROM my_table;

--基表写入数据。
gaussdb=# INSERT INTO my_table VALUES(1,1),(2,2);

--对全量物化视图my_mv进行全量刷新。
gaussdb=# REFRESH MATERIALIZED VIEW my_mv;

--对增量物化视图my_imv进行全量刷新。
gaussdb=# REFRESH MATERIALIZED VIEW my_imv;

--删除增量物化视图。
gaussdb=# DROP MATERIALIZED VIEW my_imv;

--删除全量物化视图。
gaussdb=# DROP MATERIALIZED VIEW my_mv;

--删除表my_table。
gaussdb=# DROP TABLE my_table;
```

相关链接

ALTER MATERIALIZED VIEW，**CREATE INCREMENTAL MATERIALIZED VIEW**，**CREATE MATERIALIZED VIEW**，**CREATE TABLE**，**DROP MATERIALIZED VIEW**，**REFRESH INCREMENTAL MATERIALIZED VIEW**

1.13.18.4 REFRESH SYSTEM OBJECT

功能描述

将系统对象的版本信息刷新到系统表**GS_SYSTEM_OBJECTS_VERSION**中，用于升级时判断系统对象的变化情况。

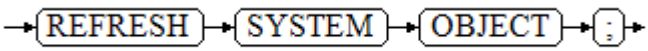
从507.0之前的版本升级时调用该语法。内部语法，不建议用户使用。

注意事项

- 此语句仅支持升级阶段调用，即在GUC参数upgrade_mode不为0时此语句可被执行。
- 只有初始用户有权限执行此语句。

语法格式

```
REFRESH SYSTEM OBJECT;
```



版权所有 © 华为云计算技术有限公司

1.13.18.5 REINDEX

功能描述

为表中的数据重建索引。

在以下几种情况下需要使用REINDEX重建索引：

- 索引崩溃，并且不再包含有效的数据。
- 索引变得“臃肿”，包含大量的空页或接近空页。
- 为索引更改了存储参数（例如填充因子），并且希望这个更改完全生效。
- 索引失效后，索引的统计信息不会更新，建议使用REINDEX重建索引。

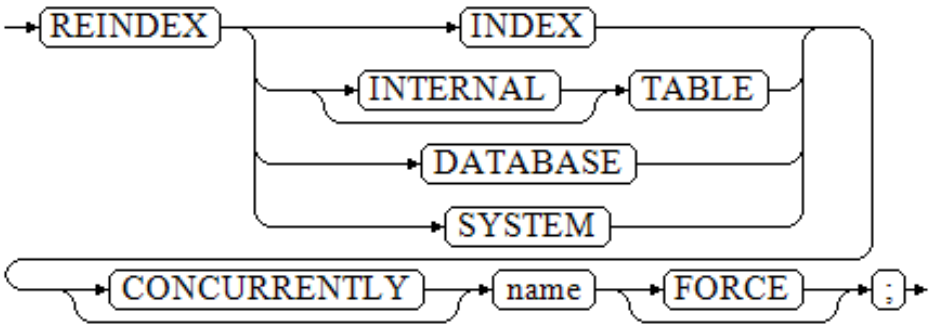
注意事项

- REINDEX DATABASE和SYSTEM这种形式的重建索引不能在事务块中执行。
- 对于全局二级索引，当前仅支持REINDEX INDEX和REINDEX TABLE。
- 执行REINDEX操作时，若索引lpi_parallel_method选项设置为PARTITION且表parallel_workers选项设置大于0，则并行重建索引按分区级并行方式进行；未设置lpi_parallel_method选项或该选项取值为AUTO时，默认按页面级并行方式进行。
- REINDEX CONCURRENTLY在线重建索引导致表上索引顺序改变时，可能会导致查询计划改变。
- REINDEX [CONCURRENTLY]当前不支持修改ubtree索引的RCR和PCR类型，即REINDEX后的索引类型与创建索引时的类型保持一致。
- REINDEX CONCURRENTLY在线重建索引操作，不支持对基于rowid系统列的索引进行重建。

语法格式

- 重建普通索引。

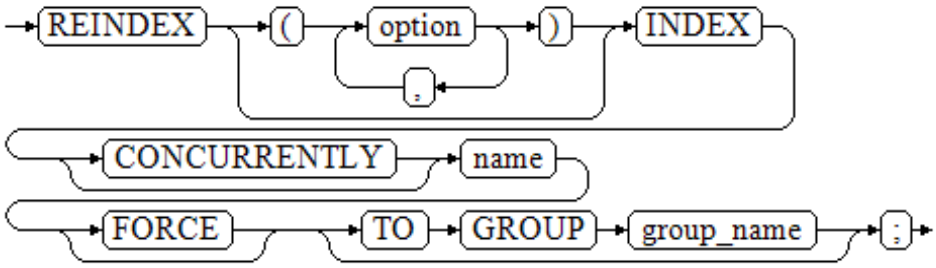
```
REINDEX { INDEX | [INTERNAL] TABLE | DATABASE | SYSTEM } [ CONCURRENTLY ] name [ FORCE ];
```



- 重建索引并进行类型转换。

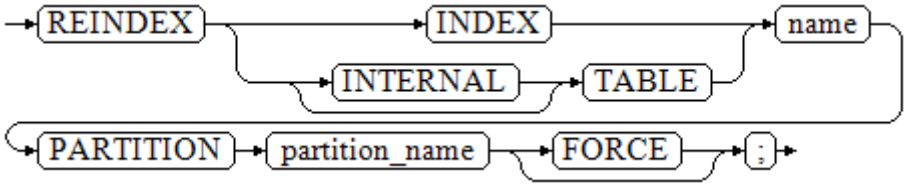
```
REINDEX [ ( option [, ...] ) ] { INDEX } [ CONCURRENTLY ] name [ FORCE ] [TO GROUP group_name];
```


版权所有 © 华为云计算技术有限公司



- 重建索引分区。

```
REINDEX { INDEX | [INTERNAL] TABLE } name
    PARTITION partition_name [ FORCE ];
```



参数说明

- **INDEX**

  重新建立指定的索引。

- **TABLE**

  重新建立指定表的所有索引，如果表有从属的“TOAST”表，则这个表也会重建索引。如果表上有索引已经被alter unusable失效，则这个索引无法被重新创建。当指定CONCURRENTLY选项时，暂不支持重建从属“TOAST”表上的索引。

- **DATABASE**

  重建当前数据库里的所有索引。当指定CONCURRENTLY选项时，暂不支持重建数据库中表的从属“TOAST”表上的索引。

- **SYSTEM**

  在当前数据库上重建所有系统表上的索引。不会处理在用户表上的索引。

- **option**

  当前支持CROSSBUCKET、TABLEOIDTRANS两种option，对于CROSSBUCKET，其值只能为ON或OFF，对于TABLEOIDTRANS，不允许指定值。

  a. CROSSBUCKET用来控制hash bucket表的索引是否在REINDEX过程中转换为跨bucket索引（cross-bucket-index，CBI）或bucket本地索引（local-bucket-index，LBI）索引。此项转换仅支持分布式hashbucket表的索引，不支持GSI索引，当前版本仅标记了扩容流程的Session可以使用在线索引类型转换。

  b. TABLEOIDTRANS用来控制hashbucket表上的GSI索引是否在重建的同时进行内部数据格式转换。

- **CONCURRENTLY**

  以不阻塞DML的方式重建索引（加ShareUpdateExclusiveLock锁）。重建索引时，一般会阻塞其他语句对该索引所依赖表的访问。指定此关键字，可以实现重建过程中不阻塞DML。不支持在线重建系统表上的索引。不支持REINDEX INTERNAL TABLE CONCURRENTLY和REINDEX SYSTEM CONCURRENTLY，不支持REINDEX INVALID INDEX CONCURRENTLY。当执行REINDEX DATABASE CONCURRENTLY时，在线重建当前数据库中用户表上的所有索引（不会处理系统

版权所有 © 华为云计算技术有限公司

表上的索引）。REINDEX CONCURRENTLY不可以在事务内执行。在线重建索引只支持B-tree索引和UB-tree索引，只支持普通索引、GLOBAL索引、LOCAL索引，不支持PCR ubtree索引，不支持二级分区。GSI支持通过REINDEX INDEX CONCURRENTLY在线重建，但不支持在线重建表达式GSI、部分GSI。GSI在线重建期间，应尽量避免与其他DDL并发执行，可能会出现死锁、报错等，可以重新连接客户端进行重试。在线并行重建索引只支持Astore及Ustore的普通索引、GLOBAL索引、LOCAL索引、GSI，其他继承当前版本在线重建索引规格约束。如果在线重建索引失败，对于用户手动取消、唯一索引键值重复、资源不足、启动线程失败、锁超时等场景，为避免占用资源，系统会自动清理新索引，自动清理索引存在一个阶段需要等待数据库其他事务结束，因此可能出现锁超时、死锁等问题，导致清理索引失败。在系统无法自动清理失败新索引的情况下（比如数据库宕机、FATAL、PANIC），需要尽快手动清除（使用DROP INDEX语句）非法新索引及（使用DROP TABLE语句）临时表，以防占用更多资源。一般来说，非法的新索引的后缀名为_ccnew，对于GSI（全局二级索引），非法的新索引前缀为ccgsi_tmp_index_。分布式中只有data node节点会自动清理失败索引，coordinator node节点的失败索引需要手动清理。严重错误场景会导致分布式部分节点中找不到失败索引的元信息，DROP INDEX语句无法删除这些索引，需要用DROP INDEX IF EXISTS删除。REINDEX INDEX CONCURRENTLY对表加4级会话锁，且其前几个阶段与CREATE INDEX CONCURRENTLY相似，因此也可能产生卡住或死锁的问题，具体场景与CREATE INDEX CONCURRENTLY相似（比如两个会话同时对同一个索引或表进行REINDEX CONCURRENTLY操作，会引发死锁问题），具体请参见**CREATE INDEX**章节中的**CONCURRENTLY**。

说明

重建索引时指定此关键字，Astore需要执行先后两次对全表的扫描来完成build，第一次扫描时创建新索引，不阻塞读写操作，第二次扫描时合并更新第一次扫描到目前为止发生的变更；Ustore需完成一次全表扫描，在扫描过程中并发DML产生的数据会被插入到以“index_oid_cctmp”命名的临时表中，扫描结束后合并临时表到以“_ccnew{n}”为后缀名（对于全局二级索引，以“ccgsi_tmp_index_”为前缀名）的新索引中并删除临时表，交换新旧索引，旧索引标记为死亡，启用新索引，重建索引完成。

生成WDR快照期间如果有在线重建索引操作，可能会出现Could not open relation with OID %u.报错导致WDR快照生成失败，此现象为预期内行为，待索引重建完成后重新生成WDR快照即可。

- **name**

  需要重建索引的索引、表、数据库的名称。表和索引可以有模式修饰。

  说明

  REINDEX DATABASE和SYSTEM只能重建当前数据库的索引，所以name必须和当前数据库名称相同。

- **FORCE**

  仅为语法兼容，并无实际含义。

- **TO GROUP group_name**

  将索引重建到指定的名称为group_name的node group上，仅在扩容流程中使用。

- **partition_name**

  需要重建索引的索引名称或者表名称。

  取值范围：

  - 如果前面是REINDEX INDEX，则这里应该指定索引的名称。
  - 如果前面是REINDEX TABLE，则这里应该指定表的名称。

版权所有 © 华为云计算技术有限公司

须知

- REINDEX DATABASE和SYSTEM这种形式的重建索引不能在事务块中执行。
- REINDEX、REINDEX CONCURRENTLY不支持单独操作toast表或toast索引。
- REINDEX CONCURRENTLY执行期间，涉及索引的stream计划可能出现报错。在执行REINDEX CONCURRENTLY前，可通过SQL patch或针对SQL添加hint关闭stream，走pgxc计划。

示例

```
--创建表tbl_test，并插入数据。
gaussdb=# CREATE TABLE tbl_test(c1 int,c2 varchar) WITH (storage_type = ustore);
gaussdb=# INSERT INTO tbl_test VALUES (1, 'AAAAAAA'),(5, 'AAAAAAB'),(10, 'AAAAAAC');

--创建索引,并查看索引大小
gaussdb=# CREATE INDEX idx_test_c1 ON tbl_test(c1);
gaussdb=# SELECT pg_size_pretty(pg_total_relation_size('idx_test_c1')) AS size;
  size
--------
 176 kB
(1 row)

--插入一万条数据，然后删除数据，发现索引变大。
gaussdb=# INSERT INTO tbl_test VALUES (generate_series(1,10000),'test');
gaussdb=# DELETE FROM tbl_test WHERE c2 = 'test';
gaussdb=# SELECT pg_size_pretty(pg_total_relation_size('idx_test_c1')) AS size;
  size
--------
 504 kB
(1 row)

--重建一个单独索引之后，查看索引信息索引大小变回初始大小。
gaussdb=# REINDEX INDEX idx_test_c1;
gaussdb=# SELECT pg_size_pretty(pg_total_relation_size('idx_test_c1')) AS size;
  size
--------
 176 kB
(1 row)

--在线重建一个单独索引。
gaussdb=# REINDEX INDEX CONCURRENTLY idx_test_c1;

--重建表tbl_test上的所有索引。
gaussdb=# REINDEX TABLE tbl_test;

--在线重建表tbl_test上的所有索引。
gaussdb=# REINDEX TABLE CONCURRENTLY tbl_test;

--删除tbl_test表。
gaussdb=# DROP TABLE tbl_test;
```

优化建议

- INTERNAL TABLE
  此种情况大多用于故障恢复，不建议进行并发操作。
- DATABASE
  不建议在事务中REINDEX DATABASE。
- SYSTEM
  不建议在事务中REINDEX系统表。

版权所有 © 华为云计算技术有限公司

1.13.18.6 RELEASE SAVEPOINT

功能描述

RELEASE SAVEPOINT删除一个当前事务先前定义的保存点。

把一个保存点删除就令其无法作为回滚点使用，除此之外它没有其它用户可见的行为。它并不能撤销在保存点建立起来之后执行的命令的影响，要撤销那些命令可以使用ROLLBACK TO SAVEPOINT 。当不再需要的时候删除一个保存点可以令系统在事务结束之前提前回收一些资源。

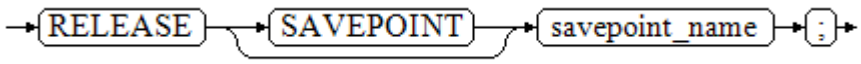
RELEASE SAVEPOINT也删除所有在指定的保存点建立之后的所有保存点。

注意事项

- 不能RELEASE一个没有定义的保存点，语法上会报错。
- 如果事务在回滚状态，则不能释放保存点。
- 如果多个保存点拥有同样的名称，只有最近定义的才被释放。

语法格式

```
RELEASE [ SAVEPOINT ] savepoint_name;
```



参数说明

savepoint_name

要删除的保存点的名称

示例

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA tpcds;

--创建一个新表。
gaussdb=# CREATE TABLE tpcds.table1(a int);

--开启事务。
gaussdb=# START TRANSACTION;

--插入数据。
gaussdb=# INSERT INTO tpcds.table1 VALUES (3);

--建立保存点。
gaussdb=# SAVEPOINT my_savepoint;

--插入数据。
gaussdb=# INSERT INTO tpcds.table1 VALUES (4);

--删除保存点。
gaussdb=# RELEASE SAVEPOINT my_savepoint;

--查询表的内容，会同时看到3和4。
gaussdb=# SELECT * FROM tpcds.table1;
 a
---
 3
```


版权所有 © 华为云计算技术有限公司

```
 4
(2 rows)

--提交事务。
gaussdb=# COMMIT;

--再次查询表的内容，会同时看到3和4。
gaussdb=# SELECT * FROM tpcds.table1;
 a
---
 3
 4
(2 rows)

--删除表。
gaussdb=# DROP TABLE tpcds.table1;

--删除SCHEMA。
gaussdb=# DROP SCHEMA tpcds CASCADE;
```

相关链接

SAVEPOINT，**ROLLBACK TO SAVEPOINT**

1.13.18.7 RENAME TABLE

功能描述

同一条语句中，修改单个表或多个表的名称，对名称的修改不会影响所存储的数据。

注意事项

- 当对单个表修改表名称时，与ALTER TABLE中RENAME语法等价。
- 当对多个表修改表名称时，与同时执行多次ALTER TABLE中RENAME语法等价；但不支持多组表中，同时存在本地临时表和非本地临时表修改表名称场景。

语法格式

修改表的名称。

```
RENAME { TABLE | TABLES } { table_name TO new_table_name } [,...];
```

说明

在MYSQL模式库中的5.7版本下（即sql_compatibility = 'MYSQL'、b_format_version='5.7'、b_format_dev_version='s2'）执行该命令，会出现以下特殊现象：

- 新表名对应字符串以"#mysql50#"开始，且其后还有其他字符，"#mysql50#"将被忽略。
- 如果新旧表名一致，不会报错。

参数说明

- **TABLE | TABLES**

  TABLE和TABLES可以互相替换使用，与语句中操作表的个数无关。
- **table_name TO new_table_name**

  table_name为需要修改的表名。

  new_table_name为修改后的新表名。

版权所有 © 华为云计算技术有限公司

TO为中间连接词。

修改表示例

- 单表修改名称

```
gaussdb=# CREATE TABLE aa(c1 int, c2 int);
gaussdb=# RENAME TABLE aa TO test_alt1;
gaussdb=# DROP TABLE test_alt1;
```

- 多个表修改名称

```
gaussdb=# CREATE TABLE aa(c1 int, c2 int);
gaussdb=# CREATE TABLE bb(c1 int, c2 int);
gaussdb=# RENAME TABLE aa TO test_alt1, bb TO test_alt2;
gaussdb=# DROP TABLE test_alt1,test_alt2;
```

相关链接

ALTER TABLE

1.13.18.8 REPLACE

功能描述

在表中插入或者替换新的数据。当插入的数据与原有数据存在主键或唯一键冲突时，执行REPLACE语句会先删除原有数据，再插入新的数据。

REPLACE语句有如下三种形式：

- 值替换插入。即通过VALUES或VALUE构造一行记录并插入到表中。
- 查询替换插入。通过SELECT子句返回的结果集构造一行或多行记录插入到表中。
- 设定指定字段值。与值插入类似，对于没有指定的列取其默认值。

注意事项

- 执行该语句的用户需要有表的DELETE和INSERT权限。
- 主键或唯一键不冲突的情况下，可直接插入；对于主键或唯一键冲突的情况，先删除原有数据，再插入新的数据。
- REPLACE操作返回格式为REPLACE 0 X，X表示删除和插入的操作数。
- REPLACE...SELECT形式，select_list列数必须与待插入的字段数保持一致。
- REPLACE...SET语法中：
  - 若col_name有默认值，则SET col_name = col_name + 1等同于SET col_name = col_name的默认值 + 1；
  - 若col_name无默认值且没有NOT NULL约束时，则SET col_name = col_name + 1等同于col_name = NULL；
  - 若col_name无默认值但存在NOT NULL约束时，支持有默认值的数据类型为：timestamp、timestamp with time zone、time、time with time zone、interval、tinterval、smalldatetime、date、datea（仅在ORA兼容性生效）、uuid、name、point、polygon、circle、lseg、box、json、jsonb、xml、varbit、numeric、cidr、inet、macaddr、numrange、int8range、int4range、tsrange、tstzrange、daterange、hash16、hash32、bool、bytea、char、bigint、int、smallint、tinyint、text、raw、blob、clob、float4、float8、abstime、reltime、bpchar、varchar、nvarchar2、money、uint1、uint2、uint4、uint8、enum。默认值如下表所示：

版权所有 © 华为云计算技术有限公司

数据类型	col_name默认值
int8、int4、int2、int1、float8、float4、numeric、uint1、uint2、uint4、uint8	0，如果numeric指定小数位，则显示小数位。如NUMERIC(10, 3)：0.000
text、clob、bpchar、varchar、char、nvarchar2、name、blob、raw、varbit	空字符串
numrange、int8range、int4range、tsrange、tstzrange、daterange	empty
bytea	\x
money	$0.00
json、jsonb、xml	'null'
macaddr	00:00:00:00:00:00
inet	0.0.0.0
cidr	0.0.0.0/32
point	(0,0)
lseg	[(0,0),(0,0)]
box	(0,0),(0,0)
path	((0,0))
polygon	((0,0))
circle	<(0,0),0>
uuid	00000000-0000-0000-0000-000000000000
hash16	0000000000000000
hash32	00000000000000000000000000000000
bool	f
abstime	abstime '1970-01-01 00:00:00'
reltime	reltime '00:00:00'
interval	interval '00:00:00'
tinterval	tinterval(abstime '1970-01-01 00:00:00', abstime '1970-01-01 00:00:00')
timestamp	timestamp '1970-01-01 00:00:00'

版权所有 © 华为云计算技术有限公司

数据类型	col_name默认值
timestamp with time zone	timestamptz '1970-01-01 00:00:00'
date	date '1970-01-01'
datea	datea '01-JAN-70'
time	time '00:00:00'
time with time zone	timetz '00:00:00'
smalldatetime	smalldatetime '1970-01-01 00:00:00'

说明

- 枚举类型（ENUM）默认值为第一个元素，如果没有第一个元素，则返回NULL。
- ORA模式数据库下（即sql_compatibility = 'ORA'），text、clob、blob、raw、bytea、varchar、nvarchar2、bpchar、char、name、byteawithoutorderwithqualcol、byteawithoutordercol类型的空串等价于NULL，在列有NOT NULL约束时，使用引用列插入表时会报错。
- MYSQL模式数据库下（即sql_compatibility = 'MYSQL'），如果b_format_version = '5.7'，b_format_dev_version = 's1'，sql_mode不包含'strict_trans_tables、only_full_group_by、no_zero_in_date、no_zero_date、error_for_division_by_zero'，timestamp和datetime在NOT NULL约束下默认值为0000-00-00 00:00:00，DATE在NOT NULL约束下默认值为0000-00-00。
- uint1、uint2、uint4、uint8数据类型仅在MYSQL模式数据库下（即sql_compatibility = 'MYSQL'）支持。
- 函数表达式的默认值，如果在解析时可计算出结果，则默认值为计算出的常量，否则为NULL。
- 除以上场景，默认值为NULL。
- datea类型以字面量形式输入时，输入字符串需符合nls_date_format格式，否则可能会报错或产生错误数据。

- REPLACE ... SET语法中，后面设置的col_name依赖于前面col_name的值，如果前面没有设置，则取默认值。如SET f1 = f1 + 1, f2 = f1场景下，f1等于f1的默认值（假设为1）+1，f2等于f1计算后的值2。
- REPLACE INTO语句中不支持触发行级（ROW）触发器；对于语句级（STATEMENT）触发器，仅支持触发INSERT STATEMENT触发器，不支持触发DELETE STATEMENT触发器。
- 不支持延迟生效（DEFERRABLE）的唯一约束或主键。
- 如果表中存在多个唯一约束，如果所插入数据违反多个唯一约束， 则会删除所有违反约束的数据，插入新的数据。此场景需要注意，可能会误删不需要删除的数据，该场景需要慎重操作。
- 如果插入多行，这些行均与表中同一行数据存在唯一约束冲突，则按照执行顺序，依次进行REPLACE操作。
- SET col_name = col_name + 1，col_name的长度不支持超过1。如SET col_name = table_name.col_name + 1中，table_name.col_name的长度为2，形如B.A，C.B.A，D.C.B.A格式都不支持。
- 包含全局二级索引的表不支持REPLACE语法。

版权所有 © 华为云计算技术有限公司

- 浮点型比较时，需要注意浮点型可能存在精度丢失。
- 如果表上创建了行访问控制策略，则不支持REPLACE INTO。
- 不支持外表。
- 不支持密态表。
- 不支持内存表。

语法格式

- 值替换插入。
```
REPLACE [ INTO ] table_name
    [ PARTITION ( partition_name [, ... ] ) ]
    [ ( col_name [, ... ] ) ]
    { VALUES | VALUE } ( value [, ... ] ) [, ... ];
```
其中value格式为：
```
{ expression | DEFAULT }
```
- 查询替换插入。
```
REPLACE [ INTO ] table_name
    [ PARTITION ( partition_name [, ... ] ) ]
    [ ( col_name [, ... ] ) ]
    query;
```
- 设置指定字段值。
```
REPLACE [INTO] table_name
    [ PARTITION ( partition_name [, ... ] ) ]
    SET col_name = value [, ... ];
```
其中value格式为：
```
{ expression | DEFAULT }
```

参数说明

- **table_name**

  要插入数据的目标表名。

  取值范围：已存在的表名。

- **col_name**

  目标表中的字段名。

  - 字段名可以有子字段名或者数组下标修饰。
  - 没有在字段列表中出现的每个字段，将由系统默认值，或者声明时的默认值填充，若都没有则用NULL填充。例如，向一个复合类型中的某些字段插入数据，那么其他字段将是NULL。
  - 目标字段（col_name）可以按顺序排列。如果没有列出任何字段，则默认全部字段，且顺序为表声明时的顺序。
  - 如果VALUE子句和QUERY中只提供了N个字段，则目标字段为前N个字段。
  - VALUE子句和QUERY提供的值在表中从左到右关联到对应列。

  取值范围：已存在的字段名。

- **PARTITION (partition_name [, ...])**

  指定分区插入操作。其中partition_name为分区名。

  如果VALUE子句的值和指定分区不一致，结果会提示异常。

- **value**

  待插入的值，value格式为：

版权所有 © 华为云计算技术有限公司

```
{ expression | DEFAULT }
```

  - expression：表示赋予对应列一个有效表达式或值。
    - 向表中字段插入单引号 " ' "时需要使用单引号自身进行转义。
    - 如果插入行的表达式不是正确的数据类型，系统试图进行类型转换，若转换不成功，则插入数据失败，系统返回错误信息。
  - DEFAULT：表示对应字段名的缺省值。如果没有缺省值，则为NULL。

- **query**

  一个查询语句（SELECT语句），将查询结果作为插入的数据。

示例

```
--创建一个新表test。
gaussdb=# CREATE TABLE test(f1 int primary key, f2 int, f3 int);

--插入3条数据。
gaussdb=# INSERT INTO test VALUES(1, 1, 1), (2, 2, 2), (3, 3, 3);
INSERT 0 3

--值替换插入数据。
gaussdb=# REPLACE INTO test VALUES(1, 11, 11);
REPLACE 0 2

--查询值替换插入的结果。
gaussdb=# SELECT * FROM test WHERE f1 = 1;
 f1 | f2 | f3
----+----+----
  1 | 11 | 11
(1 row)

--查询替换插入数据。
gaussdb=# REPLACE INTO test SELECT 2, 22, 22;
REPLACE 0 2

--查询查询替换插入的结果。
gaussdb=# SELECT * FROM test WHERE f1 = 2;
 f1 | f2 | f3
----+----+----
  2 | 22 | 22
(1 row)

--设置指定字段替换插入数据。
gaussdb=# REPLACE INTO test SET f1 = f1 + 3, f2 = f1 * 10 + 3, f3 = f2;
REPLACE 0 2

--查询设置指定字段替换插入数据的结果。
gaussdb=# SELECT * FROM test WHERE f1 = 3;
 f1 | f2 | f3
----+----+----
  3 | 33 | 33
(1 row)

--删除表格test。
gaussdb=# DROP TABLE test;
```

1.13.18.9 RESET

功能描述

RESET将指定的运行时参数恢复为缺省值。这些参数的缺省值是指gaussdb.conf配置文件中所描述的参数缺省值。

版权所有 © 华为云计算技术有限公司

RESET命令与如下命令的作用相同：

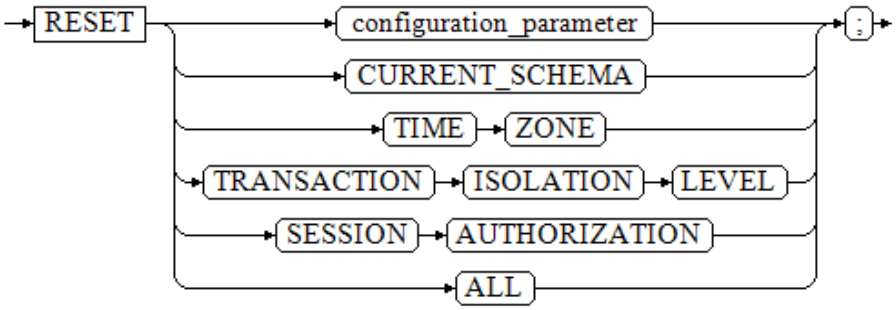
```
SET configuration_parameter TO DEFAULT;
```

注意事项

RESET的事务性行为和SET相同，它的影响将会被事务回滚撤销。

语法格式

```
RESET {configuration_parameter | CURRENT_SCHEMA | TIME ZONE | TRANSACTION ISOLATION LEVEL |
SESSION AUTHORIZATION | ALL };
```



参数说明

- **configuration_parameter**
  运行时参数的名称。
  取值范围：可以使用SHOW ALL命令查看运行时参数。

  说明

  部分通过SHOW ALL查看的参数不能通过SET设置，如max_datanodes。
- **CURRENT_SCHEMA**
  当前模式。
- **TIME ZONE**
  时区。
- **TRANSACTION ISOLATION LEVEL**
  事务的隔离级别。
- **SESSION AUTHORIZATION**
  当前会话的用户标识符。
- **ALL**
  所有运行时参数。

示例

```
--设置时区为意大利。
gaussdb=# SET timezone TO 'Europe/Rome';
--查看当前时区。
gaussdb=# SHOW timezone;
  TimeZone
-------------
 Europe/Rome
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
--将时区设置为缺省值。
gaussdb=# RESET timezone;
--查看当前时区。
gaussdb=# SHOW timezone;
 TimeZone
----------
 PRC
(1 row)

--如上SQL等效于如下两条SQL，将时区设置为缺省值。
gaussdb=# SET timezone TO DEFAULT;
gaussdb=# ALTER SESSION SET timezone to DEFAULT;
```

相关链接

SET，**SHOW**

1.13.18.10 REVOKE

功能描述

REVOKE用于撤销一个或多个角色的权限。

注意事项

非对象所有者试图在对象上REVOKE权限，命令按照以下规则执行：

- 如果授权用户没有该对象上的权限，则命令立即失败。
- 如果授权用户有部分权限，则只撤销那些有授权选项的权限。
- 如果授权用户没有授权选项，REVOKE ALL PRIVILEGES形式将发出一个错误信息，而对于其他形式的命令而言，如果是命令中指定名称的权限没有相应的授权选项，该命令将发出一个警告。

语法格式

- 回收指定表或视图上权限。

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | ALTER | DROP | COMMENT |
INDEX | VACUUM }[, ...]
    | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
        | ALL TABLES IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```

- 回收表上指定字段权限。

```
REVOKE [ GRANT OPTION FOR ]
    { {{ SELECT | INSERT | UPDATE | REFERENCES | COMMENT } ( column_name [, ...] )}[, ...]
    | ALL [ PRIVILEGES ] ( column_name [, ...] ) }
    ON [ TABLE ] table_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```

- 回收指定序列上权限。

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | UPDATE | ALTER | DROP | COMMENT }[, ...]
    | ALL [ PRIVILEGES ] }
    ON { [ SEQUENCE ] sequence_name [, ...]
        | ALL SEQUENCES IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```


版权所有 © 华为云计算技术有限公司

- 回收指定数据库上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { CREATE | CONNECT | TEMPORARY | TEMP | ALTER | DROP | COMMENT } [, ...]
    | ALL [ PRIVILEGES ] }
    ON DATABASE database_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定客户端加密主密钥上的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { USAGE | DROP } [, ...] | ALL [PRIVILEGES] }
    ON CLIENT_MASTER_KEYS client_master_keys_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定列加密密钥上的权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { USAGE | DROP } [, ...] | ALL [PRIVILEGES]}
    ON COLUMN_ENCRYPTION_KEYS column_encryption_keys_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定目录上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { READ | WRITE | ALTER | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON DIRECTORY directory_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定外部数据源上权限。
```
REVOKE [ GRANT OPTION FOR ]
  { USAGE | ALL [ PRIVILEGES ] }
    ON FOREIGN DATA WRAPPER fdw_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定外部服务器上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON FOREIGN SERVER server_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定函数上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON { FUNCTION {function_name ( [ {[ argmode ] [ arg_name ] arg_type} [, ...] ] )} [, ...]
       | ALL FUNCTIONS IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定存储过程上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON { PROCEDURE {proc_name ( [ {[ argmode ] [ arg_name ] arg_type} [, ...] ] )} [, ...]
       | ALL PROCEDURE IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定过程语言上权限。
```
REVOKE [ GRANT OPTION FOR ]
  { USAGE | ALL [ PRIVILEGES ] }
    ON LANGUAGE lang_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定模式上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { CREATE | USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```


版权所有 © 华为云计算技术有限公司

- 回收指定表空间上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { CREATE  | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON TABLESPACE tablespace_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定类型上权限。
```
REVOKE [ GRANT OPTION FOR ]
   { { USAGE | ALTER | DROP | COMMENT } [, ...] | ALL [ PRIVILEGES ] }
    ON TYPE type_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收指定子集群上权限。
```
REVOKE [ GRANT OPTION FOR ]
    { { CREATE | USAGE | COMPUTE  | ALTER | DROP } [, ...] | ALL [ PRIVILEGES ] }
    ON NODE GROUP group_name [, ...]
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ];
```

  说明

  回收子集群的create权限时，会默认回收usage和compute权限。

- 回收package对象的权限。
```
REVOKE [ GRANT OPTION FOR ]
   { { EXECUTE | ALTER | DROP | COMMENT } [, ...] | ALL [PRIVILEGES] }
   ON PACKAGE package_name [, ...]
   FROM {[GROUP] role_name | PUBLIC} [, ...]
   [ CASCADE | RESTRICT ];
```
- 按角色回收角色上的权限。
```
REVOKE [ ADMIN OPTION FOR ]
    role_name [, ...] FROM role_name [, ...]
    [ CASCADE | RESTRICT ];
```
- 回收角色上的sysadmin权限。
```
REVOKE ALL { PRIVILEGES | PRIVILEGE } FROM role_name;
```
- 回收ANY权限。
```
REVOKE [ ADMIN OPTION FOR ]
  { CREATE ANY TABLE | ALTER ANY TABLE | DROP ANY TABLE | SELECT ANY TABLE | INSERT ANY
TABLE | UPDATE ANY TABLE |
  DELETE ANY TABLE | TRUNCATE ANY TABLE | CREATE ANY SEQUENCE | CREATE ANY INDEX |
CREATE ANY FUNCTION | EXECUTE ANY FUNCTION |
  CREATE ANY PACKAGE | EXECUTE ANY PACKAGE | CREATE ANY TYPE | ALTER ANY TYPE | DROP ANY
TYPE | ALTER ANY SEQUENCE | DROP ANY SEQUENCE |
  SELECT ANY SEQUENCE | ALTER ANY INDEX | DROP ANY INDEX | CREATE ANY SYNONYM | DROP
ANY SYNONYM | CREATE ANY TRIGGER | ALTER ANY TRIGGER |
  DROP ANY TRIGGER
 } [, ...]
  FROM [ GROUP ] role_name [, ...];
```
- 回收DATABASE LINK对象权限。
```
REVOKE { CREATE | ALTER | DROP } [PUBLIC] DATABASE LINK FROM role_name;
```
- 回收PUBLIC同义词的权限。
```
REVOKE { CREATE | DROP } PUBLIC SYNONYM FROM role_name;
```

  通过内置角色（gs_role_public_synonym_create、gs_role_public_synonym_drop）的方式也可以实现回收PUBLIC同义词的权限。

  – 回收创建PUBLIC同义词权限：
    ```
    REVOKE gs_role_public_synonym_create FROM role_name;
    ```
  – 回收删除PUBLIC同义词权限：
    ```
    REVOKE gs_role_public_synonym_drop FROM role_name;
    ```


版权所有 © 华为云计算技术有限公司

参数说明

关键字PUBLIC表示一个隐式定义的拥有所有角色的组。

权限类别和参数说明，请参见**参数说明**。

任何特定角色拥有的特权包括直接授予该角色的特权、从该角色作为其成员的角色中得到的权限以及授予给PUBLIC的权限。因此，从PUBLIC收回SELECT特权并不一定会意味着所有角色都会失去在该对象上的SELECT特权，那些直接被授予的或者通过另一个角色被授予的角色仍然会拥有它。类似地，从一个用户收回SELECT后，如果PUBLIC仍有SELECT权限，该用户还是可以使用SELECT。

指定GRANT OPTION FOR时，只撤销对该权限授权的权力，而不撤销该权限本身。

如用户A拥有某个表的UPDATE权限，及WITH GRANT OPTION选项，同时A把这个权限赋予了用户B，则用户B持有的权限称为依赖性权限。当用户A持有的权限或者授权选项被撤销时，依赖性权限仍然存在，但如果声明了CASCADE，则所有依赖性权限都被撤销。

一个用户只能撤销由它自己直接赋予的权限。例如，如果用户A被指定授权（WITH ADMIN OPTION）选项，且把一个权限赋予了用户B，然后用户B又赋予了用户C，则用户A不能直接将C的权限撤销。但是，用户A可以撤销用户B的授权选项，并且使用CASCADE。这样，用户C的权限就会自动被撤销。另外一个例子：如果A和B都赋予了C同样的权限，则A可以撤销他自己的授权选项，但是不能撤销B的，因此C仍然拥有该权限。

如果执行REVOKE的角色持有的权限是通过多层成员关系获得的，则具体是哪一个包含的角色执行的该命令是不确定的。在这种场合下，建议的方法是使用SET ROLE成为特定角色，然后执行REVOKE，否则可能导致删除了不想删除的权限，或者是任何权限都没有删除。

示例

- 从用户joe收回角色jerry的权限。

```
--创建角色jerry。
gaussdb=# CREATE role jerry PASSWORD 'xxxxxxxxxx';

--将create any table权限授权给jerry,并允许该角色将权限授权给其他人。
gaussdb=# GRANT create any table TO jerry with admin option;

--创建用户joe，将角色jerry的权限授权给该用户。
gaussdb=#  CREATE user joe PASSWORD 'xxxxxxxxxxx';
gaussdb=#  GRANT jerry TO joe;

--从用户joe收回角色jerry的权限。
gaussdb=#  REVOKE jerry FROM joe;
```

- 从用户tom收回系统权限。

```
--创建用户tom。
gaussdb=# CREATE USER tom PASSWORD 'xxxxxxxxxx';

--给tom用户授予系统权限。
gaussdb=# GRANT ALL PRIVILEGES TO tom;

--从用户tom收回系统权限。
gaussdb=# REVOKE ALL PRIVILEGES FROM tom;
```

- 从用户joe收回对模式tpcds下表reason的SELECT权限。

```
--创建tpcds模式。
gaussdb=# CREATE SCHEMA tpcds;

--在tpcds模式下创建reason表。
gaussdb=# CREATE TABLE tpcds.reason (
```


版权所有 © 华为云计算技术有限公司

```
r_reason_sk         INTEGER      NOT NULL,
r_reason_id         CHAR(16)     NOT NULL,
r_reason_desc       VARCHAR(20)
);

--将tpcds.reason表查询权限授权给joe。
gaussdb=# GRANT select ON tpcds.reason TO joe;

--从用户joe收回对模式tpcds下表reason的SELECT权限。
gaussdb=# REVOKE SELECT ON TABLE tpcds.reason FROM joe;
```

- 从用户joe收回对模式tpcds下函数fun1()的ALTER权限。

```
--创建fun1()函数。
gaussdb=# CREATE or replace FUNCTION tpcds.fun1() RETURN boolean AS
BEGIN
SELECT current_user;
RETURN true;
END;
/
--给joe用户授予对fun1()的ALTER权限。
gaussdb=# GRANT ALTER ON FUNCTION tpcds.fun1() TO joe;

--从用户joe收回对模式tpcds下函数fun1的ALTER权限。
gaussdb=#  REVOKE ALTER ON FUNCTION tpcds.fun1() FROM joe;
```

- 从用户joe收回对数据库testdb的CONNECT权限。

```
--创建数据库testdb。
gaussdb=# CREATE DATABASE testdb;

--给joe用户授予连接testdb的权限。
gaussdb=# GRANT connect on database testdb TO joe WITH GRANT OPTION;

--从用户joe收回对数据库testdb的CONNECT权限。
gaussdb=# REVOKE CONNECT ON database testdb FROM joe;
```

- 清除数据。

```
gaussdb=# DROP TABLE tpcds.reason;
gaussdb=# DROP FUNCTION tpcds.fun1();
gaussdb=# DROP SCHEMA tpcds CASCADE;
gaussdb=# DROP USER joe;
gaussdb=# DROP USER tom;
gaussdb=# REVOKE create any table FROM jerry;
gaussdb=# DROP ROLE jerry;
gaussdb=# DROP DATABASE testdb;
```

相关链接

GRANT

1.13.18.11 ROLLBACK

功能描述

回滚当前事务并取消当前事务中的所有更新。

在事务运行的过程中发生了某种故障，事务不能继续执行，系统将事务中对数据库的所有已完成的操作全部撤销，数据库状态回到事务开始时。

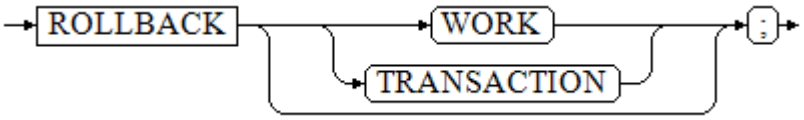
注意事项

如果不在一个事务内部发出ROLLBACK就不会有问题，但是将抛出一个NOTICE信息。

语法格式

```
ROLLBACK [ WORK | TRANSACTION ];
```


版权所有 © 华为云计算技术有限公司



参数说明

WORK | TRANSACTION

可选关键字。除了增加可读性，没有任何其他作用。

示例

```
--创建表。
gaussdb=# CREATE TABLE test (id int, name text);

--开启一个事务。
gaussdb=# START TRANSACTION;

--修改表结构。
gaussdb=# ALTER TABLE test ADD COLUMN score int;

--查看表结构。
gaussdb=# \d test;
     Table "public.test"
 Column |  Type   | Modifiers
--------+---------+-----------
 id     | integer |
 name   | text    |
 score  | integer |


--回滚。
gaussdb=# ROLLBACK;

--表结构恢复初始状态。
gaussdb=# \d test;
     Table "public.test"
 Column |  Type   | Modifiers
--------+---------+-----------
 id     | integer |
 name   | text    |


--删除test表。
gaussdb=# DROP TABLE test;
```

相关链接

COMMIT | END

1.13.18.12 ROLLBACK PREPARED

功能描述

取消一个先前为两阶段提交准备好的事务。

注意事项

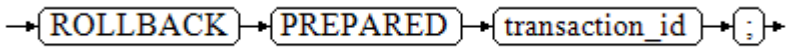
- 该功能仅在维护模式（GUC参数xc_maintenance_mode为on时）下可用。该模式谨慎打开，一般供维护人员排查问题使用，一般用户不应使用该模式。

版权所有 © 华为云计算技术有限公司

- 要想回滚一个预备事务，必须是最初发起事务的用户，或者是系统管理员。
- 事务功能由数据库自动维护，不应显式使用事务功能。

语法格式

```
ROLLBACK PREPARED transaction_id;
```



说明

分布式环境对外部不支持ROLLBACK PREPARED语法。

参数说明

transaction_id

待提交事务的标识符。它不能和任何当前预备事务已经使用了的标识符同名。

示例

```
--维护模式下开始一个事务。
gaussdb=# SET XC_MAINTENANCE_MODE = ON;
SET

gaussdb=# BEGIN;
BEGIN

--准备标识符为的trans_test的事务。
gaussdb=# PREPARE TRANSACTION 'trans_test';
PREPARE TRANSACTION

--取消标识符为的trans_test的事务。
gaussdb=# ROLLBACK PREPARED 'trans_test';
ROLLBACK PREPARED
```

相关链接

COMMIT PREPARED，**PREPARE TRANSACTION**。

1.13.18.13 ROLLBACK TO SAVEPOINT

功能描述

ROLLBACK TO SAVEPOINT用于回滚到指定保存点，该保存点之后的操作不生效。

回滚所有指定保存点建立之后执行的命令。保存点仍然有效，并且需要时可以再次回滚到该点。

注意事项

- 不能回滚到一个未定义的保存点，语法上会报错。
- 在保存点方面，游标有一些非事务性的行为。任何在保存点里打开的游标都会在回滚掉这个保存点之后关闭。如果一个前面打开了的游标在保存点里面，并且游标被一个FETCH命令影响，而这个保存点稍后回滚了，那么这个游标的位置仍然在FETCH让它指向的位置（也就是FETCH不会被回滚）。关闭一个游标的行为也

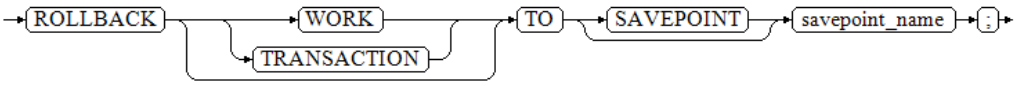
版权所有 © 华为云计算技术有限公司

不会被回滚给撤消掉。如果一个游标的操作导致事务回滚，那么这个游标就会置于不可执行状态，所以，尽管一个事务可以用ROLLBACK TO SAVEPOINT重新恢复，但是游标不能再使用了。

- 使用ROLLBACK TO SAVEPOINT回滚到一个保存点。使用RELEASE SAVEPOINT删除一个保存点，但是保留该保存点建立后执行的命令的结果。

语法格式

```
ROLLBACK [ WORK | TRANSACTION ] TO [ SAVEPOINT ] savepoint_name;
```



参数说明

- **WORK | TRANSACTION**

  可选关键字。除了增加可读性，没有任何其他作用。

- **savepoint_name**

  回滚截至的保存点。

示例

```
--开启事务。
gaussdb=# START TRANSACTION;

--设置保存点my_savepoint。
gaussdb=# SAVEPOINT my_savepoint;

--回滚至保存点my_savepoint。
gaussdb=# ROLLBACK TO SAVEPOINT my_savepoint;

--游标位置不受保存点回滚的影响。
gaussdb=# DECLARE foo CURSOR FOR SELECT 1 UNION SELECT 2;

--设置保存点foo。
gaussdb=# SAVEPOINT foo;

--读取一条数据，为第一条数据。
gaussdb=# FETCH 1 FROM foo;
 ?column?
----------
        1
--回滚至保存点foo。
gaussdb=# ROLLBACK TO SAVEPOINT foo;

--读取一条数据，为第二条数据。
gaussdb=# FETCH 1 FROM foo;
 ?column?
----------
        2
--释放保存点。
gaussdb=# RELEASE SAVEPOINT my_savepoint;

--结束事务。
gaussdb=# COMMIT;
```

相关链接

SAVEPOINT，**RELEASE SAVEPOINT**

版权所有 © 华为云计算技术有限公司

1.13.19 S

1.13.19.1 SAVEPOINT

功能描述

SAVEPOINT用于在当前事务里建立一个新的保存点。

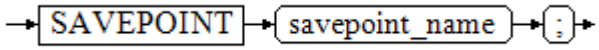
保存点是事务中的一个特殊记号，它允许将那些在它建立后执行的命令全部回滚，把事务的状态恢复到保存点所在的时刻。

注意事项

- 使用ROLLBACK TO SAVEPOINT回滚到一个保存点。使用RELEASE SAVEPOINT删除一个保存点，但是保留该保存点建立后执行的命令的效果。
- 保存点只能在一个事务块里面建立。在一个事务里面可以定义多个保存点。
- 由于节点故障或者通信故障引起的分布式节点线程或进程退出导致的报错，以及由于COPY FROM操作中源数据与目标表的表结构不一致导致的报错，均不能正常回滚到保存点之前，而是整个事务回滚。
- SQL标准要求，使用SAVEPOINT建立一个同名保存点时，需要自动删除前面同名保存点。在GaussDB数据库里，将保留旧的保存点，但是在回滚或者释放的时候，只使用最近的保存点。释放了新的保存点将导致旧的再次成为ROLLBACK TO SAVEPOINT和RELEASE SAVEPOINT可以访问的保存点。除此之外，SAVEPOINT是完全符合SQL标准的。

语法格式

```
SAVEPOINT savepoint_name;
```



参数说明

savepoint_name

新建保存点的名称。

须知

使用SAVEPOINT时，建议及时RELEASE SAVEPOINT，避免子事务的嵌套个数过大，建议嵌套个数不要超过10000，嵌套数过多时，可能引发当前事务性能劣化。

示例

```
--创建一个新表。
gaussdb=# CREATE TABLE table1(a int);

--开启事务。
gaussdb=# START TRANSACTION;

--插入数据。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO table1 VALUES (1);

--建立保存点。
gaussdb=# SAVEPOINT my_savepoint;

--插入数据。
gaussdb=# INSERT INTO table1 VALUES (2);

--回滚保存点。
gaussdb=# ROLLBACK TO SAVEPOINT my_savepoint;

--插入数据。
gaussdb=# INSERT INTO table1 VALUES (3);

--提交事务。
gaussdb=# COMMIT;

--查询表的内容，会同时看到1和3,不能看到2，因为2被回滚。
gaussdb=# SELECT * FROM table1;

--删除表。
gaussdb=# DROP TABLE table1;
```

相关链接

RELEASE SAVEPOINT，**ROLLBACK TO SAVEPOINT**

1.13.19.2 SECURITY LABEL ON

功能描述

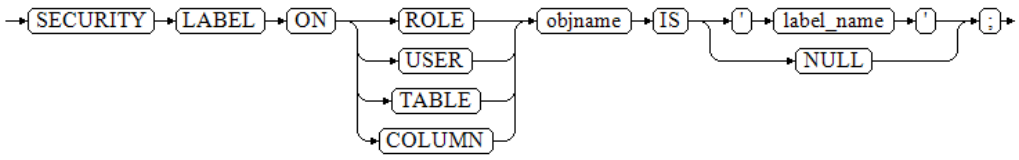
应用更新或取消安全标签。

注意事项

初始用户，具有SYSADMIN权限的用户或者继承了内置角色gs_role_seclabel权限的用户有权限应用更新或取消安全标签。

语法格式

```
SECURITY LABEL ON { ROLE | USER | TABLE | COLUMN } objname IS {'label_name' | NULL};
```



参数说明

- **objname**
  - 对于ROLE和USER，objname表示角色/用户名。
  - 对于TABLE，objname表示表名，名称前可加上模式名。
  - 对于COLUMN，objname表示表名.列名，名称前可加上模式名。
- **label_name**
  安全标签名。

版权所有 © 华为云计算技术有限公司

- **NULL**
  表示取消安全标签。

示例

```
--创建安全标签。
gaussdb=# CREATE SECURITY LABEL sec_label 'L1:G4';

--创建表。
gaussdb=# CREATE TABLE tbl(c1 int, c2 int);

--创建用户。
gaussdb=# CREATE USER bob WITH PASSWORD '********';

--给用户应用安全标签。
gaussdb=# SECURITY LABEL ON ROLE bob IS 'sec_label';

--给表应用安全标签。
gaussdb=# SECURITY LABEL ON TABLE tbl IS 'sec_label';

--给表的列应用安全标签。
gaussdb=# SECURITY LABEL ON COLUMN tbl.c1 IS 'sec_label';

--给用户取消安全标签。
gaussdb=# SECURITY LABEL ON ROLE bob IS NULL;
--给表取消安全标签。
gaussdb=# SECURITY LABEL ON TABLE tbl IS NULL;

--给表的列取消安全标签。
gaussdb=# SECURITY LABEL ON COLUMN tbl.c1 IS NULL;

--删除表tb1。
gaussdb=# DROP TABLE tbl;

--删除用户bob。
gaussdb=# DROP USER bob;

--删除已存在的安全标签sec_label。
gaussdb=# DROP SECURITY LABEL sec_label;
```

相关链接

CREATE SECURITY LABEL，**DROP SECURITY LABEL**

1.13.19.3 SELECT

功能描述

从表或视图中查询数据。SELECT操作就像叠加在数据库表上的过滤器，通过指定查询的条件信息，可以从数据库表中过滤出用户需要的数据。

注意事项

- 表的所有者、拥有表SELECT权限的用户或拥有SELECT ANY TABLE权限的用户，可以对表或视图进行SELECT操作查看数据，当三权分立开关关闭时，系统管理员默认拥有此权限。
- SELECT支持普通表之间的JOIN，不支持普通表和GDS外表的JOIN，即SELECT语句中不能同时出现普通表和GDS外表。
- 用户进行SELECT操作，必须拥有SELECT语句中使用的字段的SELECT权限。

版权所有 © 华为云计算技术有限公司

- 仅同时拥有对表的SELECT权限和UPDATE权限的用户，可以对表进行FOR UPDATE或FOR SHARE操作。
- 指定分区语句目前不支持走全局索引扫描。

语法格式

查询数据：

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [/*+ plan_hint */] [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
   { * | {expression [ [ AS ] output_name ]} [, ...] }
   [ FROM from_item [, ...] ]
   [ WHERE condition ]
   [ [ START WITH condition ] CONNECT BY [NOCYCLE]
   condition [ ORDER SIBLINGS BY expression ] ]
   [ GROUP BY grouping_element [, ...] ]
   [ HAVING condition [, ...] ]
   [ WINDOW {window_name AS ( window_definition )} [, ...] ]
   [ { UNION | INTERSECT | EXCEPT | MINUS } [ ALL | DISTINCT ] select ]
   [ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS { FIRST |
LAST } ]} [, ...] ]
   [ LIMIT { [offset,] count | ALL } ]
   [ OFFSET start [ ROW | ROWS ] ]
   [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY [OFFSET offset_count]]
   [ {FOR { UPDATE | SHARE } [ OF table_name [, ...] ] [ NOWAIT | WAIT n | SKIP LOCKED ]} [...] ];
```

须知

- 缓存SELECT语句计划的场景下，WHERE IN候选子集建议条件个数不超过100，防止引发动态内存过高问题：
  - WHERE IN候选子集过大时，生成计划的内存占用会增大。
  - 当拼接SQL构造的WHERE IN子集不同，缓存计划的SQL模板无法复用。会生成大量不同的计划且计划无法共享，占用大量内存。
- condition和expression中可以使用targetlist（查询返回列表）中表达式的别名：
  - 只能同一层引用。
  - 只能引用targetlist中的别名。
  - 只能是后面的表达式引用前面的表达式。
  - 不能包含volatile函数。
  - 不能包含Window function函数。
  - 不支持在JOIN ON条件中引用别名。
  - targetlist中有多个要应用的别名则报错。

- 其中子查询with_query为：

```
with_query_name [ ( column_name [, ...] ) ]
    AS [ [ NOT ] MATERIALIZED ] ( {select | values | insert | update | delete} )
```

- 其中指定查询源from_item为：

```
{[ ONLY ] table_name [ * ] [ partition_clause ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
[ TABLESAMPLE sampling_method ( argument [, ...] ) [ REPEATABLE ( seed ) ] ]
[ TIMECAPSULE {TIMESTAMP | CSN} expression ]
|( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
|with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
|function_name ( [ argument [, ...] ] ) [ AS ] alias [ ( column_alias [, ...] | column_definition [, ...] ) ]
|function_name ( [ argument [, ...] ] ) AS ( column_definition [, ...] )
|xmltable_clause
```


版权所有 © 华为云计算技术有限公司

```
|from_item unpivot_clause
|from_item pivot_clause
|from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]}
```

- 指定查询源from_item中指定分区partition_clause为：

  ```
  {PARTITION { ( { partition_name | subpartition_name } [, ...] ) | FOR (  partition_value [, ...] ) }
  | SUBPARTITION { ( subpartition_name ) | FOR (  subpartition_value [, ...] )}}
  ```

  说明

  - 指定分区只适合分区表。
  - PARTITION指定多个分区名时，可以存在相同的分区名，最终的分区范围取多个分区的并集。

- 指定查询源from_item中xmltable_clause为：

  ```
  XMLTABLE(
  xmlnamespaces_clause
  row_expression
  passing_clause
  columns_clause
  )
  ```

  - xmlnamespaces_clause为：

    ```
    {
       XMLNAMESPACES(
       {string AS identifier
       |
       DEFAULT string }
       [, { string AS identifier  |  DEFAULT string } ][...]
       ),
    }
    ```

  - passing_clause为：

    ```
    [
       PASSING [BY { REF | VALUE }] document_expression [BY { REF | VALUE }]
    ]
    ```

  - columns_clause为：

    ```
    [
       COLUMNS
       name { type [PATH column_expression] [DEFAULT default_expression] [NOT NULL |
    NULL] | FOR ORDINALITY }
       [, ...]
    ]
    ```

- 其中GROUP BY子句为：

  ```
  {()
  |expression
  |( expression [, ...] )
  |ROLLUP ( { expression | ( expression [, ...] ) } [, ...] )
  |CUBE ( { expression | ( expression [, ...] ) } [, ...] )
  |GROUPING SETS ( grouping_element [, ...] )}
  ```

- 其中设置排序方式nlssort_expression_clause为：

  ```
  NLSSORT ( column_name, ' NLS_SORT = { SCHINESE_PINYIN_M | generic_m_ci } ' )
  ```

- 简化版查询语法，功能相当于SELECT * FROM table_name。

  ```
  TABLE { ONLY {(table_name)| table_name} | table_name [ * ]};
  ```

参数说明

- **WITH [RECURSIVE] with_query [, ...]**

  用于声明一个或多个可以在主查询中通过名称引用的子查询，相当于临时表。这种子查询语句结构称为CTE（Common Table Expression）结构，应用CTE结构时，执行计划中将存在CTE SCAN的内容。如果声明了RECURSIVE，SELECT子查询可以通过名称引用其自身。

版权所有 © 华为云计算技术有限公司

其中with_query的详细格式为：

```
with_query_name [ ( column_name [, ...] ) ] AS [ [ NOT ] MATERIALIZED ] ( {select | values | insert | update | delete} )
```

- with_query_name：指定子查询生成的结果集名称，在查询中可使用该名称访问子查询的结果集。
- column_name：指定子查询结果集中显示的列名。
- 每个子查询可以是SELECT、VALUES、INSERT、UPDATE或DELETE语句。
- 应用CTE结构，需要声明RECURSIVE，且RECURSIVE需要在WITH后进行声明。当存在多个CTE的情况下，只需要在第一个CTE处声明RECURSIVE。
- 用户可以使用[NOT] MATERIALIZED对CTE进行修饰。对于Stream计划，目前仅支持内联执行方式，此时该语法不生效。
  - 如果声明为MATERIALIZED，WITH查询将被物化，生成一个子查询结果集的复制，在引用处直接查询该复制，因此WITH子查询无法和主干SELECT语句进行联合优化（如谓词下推、等价类传递等）。当使用NOT MATERIALIZED进行修饰时，如果WITH查询语义上可以作为子查询内联执行，则可以和主干SELECT语句进行联合优化（如谓词下推、等价类传递等）。
  - 如果用户没有显式声明物化属性则遵守以下规则：如果CTE只在所属SELECT主干中被引用一次，且语义上支持内联执行，则会被改写为子查询内联执行，否则以CTE SCAN的方式物化执行。

- **plan_hint子句**

  用于对SELECT对应语句块生成的计划进行hint调优，以/*+ */的形式在SELECT关键字后进行声明，具体的hint调优用法请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。每条语句中只有第一个/*+ plan_hint */注释块会作为hint生效，注释块中可以写多条hint。

- **ALL**

  可选，返回所有符合SELECT查询条件的行，为SELECT查询的默认行为。

- **DISTINCT [ON (expression [, ...])]**

  从SELECT的结果集中删除所有重复的行，确保结果集中的每行都是唯一的。

  ON (expression [, ...])：只保留符合expression运算的结果集合中的第一行数据。

  须知

  - 由于DISTINCT ON仅保留符合expression的行的集合的第一行，并未对符合expression的行的集合进行排序，所以指定DISTINCT ON后，其结果并非固定，当需要返回唯一且符合预期的数据行时，建议同时指定ORDER BY进行数据筛选。
  - DISTINCT ON表达式中不支持非整数常量排序。

- **SELECT列表**

  指定查询表中列名，可以是部分列或者是全部（使用通配符*表示）。

  通过使用子句AS output_name可以对输出字段进行添加别名，别名通常用于输出字段的显示。

  列名可以用以下形式进行表达：

版权所有 © 华为云计算技术有限公司

- 手动输入列名，多个列之间用逗号分隔。
- 列名可以用FROM子句计算出来的结果/字段表示。
- 支持关键字NAME、VALUE和TYPE作为列别名。

- **FROM子句**

  为SELECT查询限定一个或多个数据源表。

  FROM子句中涉及如下的元素：

  - table_name

    表名或视图名，名称前可加模式名进行限定，如：schema_name.table_name。

    说明

    支持使用DATABASE LINK方式对远程表以及同义词进行操作，具体的使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

  - alias

    给表或复杂的表引用添加一个临时的表别名，方便被其他的查询进行引用。

    别名用于方便引用或者在自连接中消除歧义。如果提供了别名，可直接引用别名（代替表的实际名称）。

    说明

    JOIN操作产生的拼接表joined_table添加别名时，如果joined_table被括号指定（即(joined_table)），请勿使用非保留关键字UNPIVOT和PIVOT作为表别名。

  - TABLESAMPLE sampling_method（argument [, ...]）[REPEATABLE（seed）]

    表示使用指定的采样方法sampling_method检索表table_name中行的子集。可选择的采样方法sampling_method包括：

    - SYSTEM(percent)

      按数据块随机采样。其中，percent的取值范围为[0, 100]，表示每个数据块被选中的概率百分比。

    - BERNOULLI(percent)

      按行独立随机采样。其中，percent的取范围为[0, 100]，表示每一行被选中的概率百分比。

    - HYBRID(p1,p2)

      混合随机采样。包括两个步骤：首先以概率p1对数据块进行随机采样，其次对已采样的数据内容以概率p2进行按行独立随机采样。其中，p1和p2的取值范围均为[0, 100]。

    REPEATABLE子句：可选，采样方法sampling_method的补充条件，seed为采样方法中产生随机数的种子值，可以为任意非空常量值。当表结构/数据未发生更改且进行多次采样操作时，如果未指定REPEATABLE（seed），系统将基于自动生成的种子值为每一次查询获取一个新的随机采样，获取的采样结果并非固定；指定REPEATABLE（seed）后，获取的采样结果唯一固定。

  - TIMECAPSULE { TIMESTAMP | CSN } expression

    闪回查询的标识，查询指定CSN点或指定时间点表的内容。

    - TIMECAPSULE TIMESTAMP

版权所有 © 华为云计算技术有限公司

根据date日期，闪回查找指定时间点的结果集。date日期必须为一个过去有效的时间戳。

- TIMECAPSULE CSN

  根据表的CSN闪回查询指定CSN点的结果集。其中CSN可从gs_txn_snapshot记录的snpcsn获取。

- expression

  常量、函数或SQL表达式。

  说明

  - 系统表、DFS表、全局临时表、本地临时表、UNLOGGED表、视图、序列表、共享表以及继承表目前不支持闪回查询。
  - 当对表进行表结构修改操作或物理存储操作后，无法对表再次进行闪回查询，即在闪回点和当前点之间如果执行过修改表结构或影响物理存储的语句（TRUNCATE、DDL、DCL或VACUUM FULL），进行闪回查询会显示操作失败。执行过DDL的表进行闪回操作报错“ERROR：The table definition of %s has been changed.”；涉及表结构改变等操作的DDL执行闪回操作报错“ERROR: recycle object %s desired does not exist.”。
  - 闪回查询支持PCR类型的UBTree索引进行索引扫描；如果没有创建PCR类型的UBTree索引，闪回查询只能通过seqScan进行全表扫描。
  - 因闪回点过旧（闪回版本被回收等导致无法获取闪回点版本），导致闪回失败报错“Restore point too old.”。
  - 通过TIMESTAMP指定闪回点，闪回数据和实际时间点最多偏差为3秒。
  - 当对表执行TRUNCATE后，再次进行闪回查询或闪回表操作时，通过TIMESTAMP进行的闪回操作会报错（Snapshot too old）；通过CSN进行的闪回操作会找不到数据或者报错（Snapshot too old）。
  - GTM-Free场景下，各节点使用本地CSN且没有全局统一CSN号，暂不支持使用CSN方式进行闪回查询操作。

- column_alias

  列别名。

- PARTITION

  查询分区表的某个分区的数据。

- partition_name

  分区名。

- partition_value

  指定的分区键值。在创建分区表时，如果指定了多个分区键，可以通过PARTITION FOR子句指定的这一组分区键的值，唯一确定一个分区。

- subquery

  子查询，创建一个临时表保存子查询的输出。

- with_query_name

  可以通过WITH查询的名称对其进行引用。

- function_name

  函数名称。

- join_type

  - [INNER] JOIN

    内连接（等值连接）。只返回连接表中与连接条件完全匹配的行数据库。

版权所有 © 华为云计算技术有限公司

- LEFT [OUTER] JOIN

  左（外）关联。将两个表进行拼接，以左边的表作为主表，根据查询条件，主表数据全部展示，从表数据根据连接条件是否匹配进行展示，无法匹配的信息数据用NULL填充。

- RIGHT [OUTER] JOIN

  右（外）关联。将两个表进行拼接，以右边的表作为主表，根据查询条件，主表数据全部展示，从表数据根据连接条件是否匹配进行展示，无法匹配的信息数据用NULL填充。

  可以将右（外）关联转换成左（外）关联，只要把左边和右边的输入互换位置即可。

- FULL [OUTER] JOIN

  返回所有内连接的结果行，以及返回连接表中与连接条件不匹配的所有的行。

- CROSS JOIN

  CROSS JOIN等效于INNER JOIN ON (TRUE)，即没有被条件删除的行。这种连接类型只是符号上的方便，与简单的FROM和WHERE的效果相同。

  说明

  INNER JOIN和OUTER JOIN必须声明一个连接条件（即ON join_condition或USING (join_column [， ...])），CROSS JOIN中不需要声明。

  其中CROSS JOIN和INNER JOIN生成一个简单的笛卡尔积，和在FROM的顶层列出两个项的结果相同。

– ON join_condition

  连接条件，用于限定连接中的连接对象的匹配规则，如ON left_table.a = right_table.a。不建议使用int等数值类型作为join_condition，因为int等数值类型可以隐式转换为bool值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

– USING(join_column[，...])

  ON left_table.a = right_table.a AND left_table.b = right_table.b ... 的简写。要求对应的列必须同名。

– NATURAL

  NATURAL是具有相同名称的两个表的所有列的USING列表的简写。

– from item

  用于连接的查询源对象的名称。

– xmltable_clause

  xmltable_clause仅支持在ORA兼容模式和PG兼容模式下使用。
  xmltable_clause基于指定的XMLTYPE类型（ORA兼容模式下）或XML类型（PG兼容模式下）的数据产生一个虚拟表。

  - xmlnamespaces_clause

    可选，结构由XMLNAMESPACES以及以英文逗号分隔的XML命名空间声明的子句组成（以英文逗号分隔的XML命名空间声明的子句位于XMLNAMESPACES之后）。其中string为英文单引号作为边界符的命名空间全称，identifier为英文双引号作为边界符的命名空间别名且别名可

版权所有 © 华为云计算技术有限公司

以在row_expression及columns_clause中使用。当前版本不支持使用DEFAULT声明默认的命名空间。

- row_expression

  row_expression为英文单引号作为边界符的XPath 1.0表达式。将passing_clause中的document_expression作为其上下文，获得一组XML节点。XML节点经过columns_clause处理生成虚拟表的每一行。

  在ORA兼容模式下，row_expression支持传入以英文单引号（'）作为边界符、逗号（,）作为分隔符的数字或者XML文本组成的序列。

- passing_clause

  passing_clause以PASSING开始，其中的document_expression为传入XMLTABLE中被处理的XMLTYPE类型（ORA兼容模式下）或XML类型（PG兼容模式下）的数据，目前仅支持单个数据。在ORA兼容模式下，使用BY VALUE或BY REF都会报错；在PG兼容模式下，虽然可以识别接收第一处和第二处的BY VALUE或BY REF，但并未做功能上的处理。

- columns_clause

  可选，用于指明将在虚拟表中生成的列所需的相关信息，如**表1-405**所示。

  表 1-405 COLUMNS 信息说明

信息	说明
name	列的名称。
type	列的类型。
PATH	可选，column_expression为XPath 1.0的表达式，由row_expression得到的节点集中的某个节点作为其上下文，经其处理得到该节点生成的行数据中对应列的值所需的数据，在将其转换为type类型的结果时，存在隐式转换。若没有指定PATH部分，则name会被当作column_expression。
DEFAULT	可选，default_expression为表达式，如果通过column_expression处理后得到的是NULL值，会将计算default_expression得到的结果用于替换NULL值，default_expression仅在需要的时候才进行计算，若表达式固定，则在需要的时候只会计算一次。
NOT NULL | NULL	可选，在PG兼容模式下，用于限定该列是否可为NULL值，若列值与指定的选项不符则会报错。
FOR ORDINALITY	说明该列为行号列，从1开始为生成的行填充行号，最多只能有一个列被标记为FOR ORDINALITY。

在ORA兼容模式下，columns_clause可省略，省略columns_clause后，默认生成COLUMNS column_value XMLTYPE PATH '.'用于进行后续处理。

PG兼容模式下，columns_clause不可省略。

版权所有 © 华为云计算技术有限公司

说明

目前xmltable_clause存在以下约束：

- 涉及的XPath表达式均为XPath 1.0。
- 目前不支持在row_expression使用形如“namespace::*”的XPath表达式。
- 目前不支持在column_expression使用形如“..”的XPath表达式。
- 通过column_expression得到的数据在转换为type类型的数据时：在ORA兼容模式下，当数据过长且超过类型的typmod会截断处理；目前仅对CHAR、CHARACTER、NCHAR、BPCHAR、VARCHAR、CHARACTER VARYING、VARCHAR2以及NVARCHAR2数据类型（包括附带(n)形式）的typmod大于0时，过长的数据会进行截断处理。在PG兼容模式下，当数据过长且超过类型的typmod时会报错处理。
- 目前支持传入的XMLTYPE类型（ORA兼容模式下）或XML类型（PG兼容模式下）数据的最大值为1GB。
- 从不支持XMLTABLE的版本升级到支持XMLTABLE的版本时，在升级观察期期间，不可使用XMLTABLE语法。
- 仅在ORA兼容模式下支持XPath中存在“*:nodename”，表示选取nodename节点时忽略其命名空间，其中nodename为节点名称。
- 非保留关键字XMLTABLE不能作为Functions as Table Sources类型函数的函数名称。
- 在ORA兼容模式下，使用数字和XML文本组成的序列输入场景下，存在以下约束：
  - 不支持和passing_clause、columns_clause联合使用。
  - 序列中以逗号分隔的元素个数范围为[1, 1024]。
  - 不支持表达式作为序列元素。
  - XML文本内容中的>、"以及&，不会被转义为>、"和&。
  - XML文本节点名称仅支持与大于符号（>）之间存在一个或多个空格，并且输出结果不会删除节点名称与大于符号（>）之间的空格。
  - 升级观察期不支持该功能。
- xmltable嵌套xmlquery的returning sequence场景下，column_clause中的xpath仅支持'.'。
- xpath中'.'后不能跟下标表达式，例如：'.[6]'；或者谓词表达式，例如：'.[.>5]'。
- 在输入非良构的XMLTYPE类型（ORA兼容模式下）数据时，在非良构部分之后的用于节点间控制书写格式的空格和回车字符当作文本节点解析，如node2节点的结束标签与node3节点的开始标签之间的空格和回车即会被当作文本节点解析。

  输入情况：

  ```
  <root>
  <node1>node1</node1>
  malform
  <node2>node2</node2>
  <node3>node3</node3>
  </root>
  ```

  实际情况：

  ```
  <root><node1>node1</node1>
  malform
  <node2>node2</node2>
  <node3>node3</node3>
  </root>
  ```

示例：

```
-- 创建ORA兼容数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA' ENCODING = 'utf8';
gaussdb=# \c ora_db
-- 支持数字和xml文本组成的序列输入。
```


版权所有 © 华为云计算技术有限公司

```
ora_db=# SELECT * FROM XMLTABLE('1, 2, <a>12</a>');
 column_value
--------------
 1
 2
 <a>12</a>
(3 rows)

ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

- unpivot_clause

  将列转换为行。

  ```
  UNPIVOT [ {INCLUDE | EXCLUDE} NULLS ]
    (
       unpivot_col_clause
       unpivot_for_clause
       unpivot_in_clause
    )
  ```

  - {INCLUDE | EXCLUDE} NULLS

    可选，该子句用于控制转置后的结果是否包含存在NULL值的行。未指定该子句，unpivot操作默认会从结果集中过滤存在NULL值的行。

    INCLUDE NULLS：结果集中保留包含存在NULL值的行。

    EXCLUDE NULLS：从结果集中过滤存在NULL值的行。

  - unpivot_col_clause为：

    ```
    unpivot_col_element
    ```

    unpivot_col_element指定了输出的列名，用于保存待转置列的列值。

  - unpivot_col_element为：

    ```
    { column | ( column [, column]... ) }
    ```

  - unpivot_for_clause为：

    ```
    FOR { unpivot_col_element }
    ```

    unpivot_col_element：用于指定输出的列名，用于保存待转置列的列名或别名。

  - unpivot_in_clause为：

    ```
    IN ( unpivot_in_element [,unpivot_in_element...] )
    ```

    unpivot_in_clause：待转置列，列名和列值将保存在unpivot_col_element中。

    unpivot_in_element为：

    ```
    { unpivot_col_element }[  AS { unpivot_alias_element } ]
    ```

    unpivot_col_element：指定的待转置列，若采用(column [, column]...)形式指定待转置列，(column [, column]...)中所有的column列名将通过下划线“_”进行拼接，并保存在输出列中。如，IN ((col1, col2))将会生成列名“col1_col2”，并保存在unpivot_for_clause指定的输出列中。AS关键字可为待转置列指定别名，当指定别名后，输出列中将保存别名而不再保存待转置列的列名。

  - unpivot_alias_element为：

    ```
    { alias | ( alias [, alias]... )}
    ```

    alias：为指定的别名。

版权所有 © 华为云计算技术有限公司

说明

目前unpivot_clause存在如下约束：

- 仅支持在ORA兼容模式数据库下使用。
- unpivot_clause子句内不支持与hint配合使用。
- 对于unpivot_col_clause，其unpivot_col_element指定的输出列数目需与unpivot_in_clause中unpivot_col_element的列数目相同。
- 对于unpivot_for_clause，其unpivot_col_element指定的输出列数目需与unpivot_in_clause中unpivot_alias_element的别名数目相同。
- 对于unpivot_in_clause，别名必须为常量，或者可以转换为常量的表达式。
- 对于unpivot_in_clause，常量表达式支持的函数仅支持为不可变（IMMUTABLE）函数。
- 对于unpivot_in_clause的所有unpivot_col_element，如果unpivot_col_element相同位置的column类型存在差异，unpivot会尝试进行类型转换，以将这些转置列的列值转换为公共类型。同理。对于unpivot_alias_element，如果unpivot_alias_element相同位置的alias类型存在差异，unpivot也会进行类似的类型转换。

– pivot_clause

将行转换为列。

```
PIVOT [ XML ]
( aggregate_function ( expr ) [[AS] alias ]
    [, aggregate_function ( expr ) [[AS] alias ] ]...
    pivot_for_clause
    pivot_in_clause
)
```

- aggregate_function (expr) [[AS] alias]

  aggregate_function针对给定的表达式进行聚合计算，计算结果将保存在pivot_in_clause指定的输出列中。

  [AS] alias：为aggregate_function指定别名，别名将以“_别名”格式记录在pivot_in_clause指定的输出列名后。

- pivot_for_clause为：

  ```
  FOR { column
      | ( column [, column]... )
      }
  ```

  pivot_for_clause：待转置行。

  column：表示待转置行的列。

- pivot_in_clause为：

  ```
  IN ( { { { expr
           | ( expr [, expr]... )
           } [ [AS] alias]
       }...
     }
  )
  ```

  pivot_in_clause指定了输出结果的列名，列名可由一个expr或多个expr构成。当列名由多个expr构成时，expr按顺序通过下划线“_”进行连接，即“expr1, expr2”对应的输出列名为“expr1_expr2”。除了生成输出列名外，expr还决定着聚合函数触发时机，当待转置行的行值与expr的值相同时，pivot将进行聚合函数aggregate_function的计算，并将计算结果保存在列名由expr构成的输出列中。

版权所有 © 华为云计算技术有限公司

说明

目前pivot_clause存在如下约束：

- 仅支持在ORA兼容模式下使用。
- pivot_clause子句内不支持与hint配合使用。
- 当指定多于一个aggregate_function时，最多允许一个aggregate_function没有别名，其余aggregate_function均需指定别名。
- XML只支持语法不支持功能。
- pivot_in_clause中的expr可以是常量，或者是可以转换为常量的表达式。若expr不是一元表达式，则需为expr指定别名。
- 对于pivot_in_clause中的expr，常量表达式支持的函数只支持不可变（IMMUTABLE）函数。
- 对于pivot_in_clause中的expr，当通过AS为其指定别名时，非保留关键字可作为别名使用；其他情况下，非保留关键字不能作为别名使用。
- 输出列的列名长度限制为63，超出将报错。

- **WHERE子句**

  WHERE子句构成一个行选择表达式，用来缩小SELECT查询的范围。

  condition：返回值为布尔型的任意表达式，仅满足该条件的行才会被显示。请勿使用int等数值类型作为condition，因为int等数值类型可以隐式转换为bool值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

  WHERE子句中可以通过指定"(+)"操作符的方法将表的连接关系转换为外连接。该操作并非SQL的标准语法，在做平台迁移的时候可能面临语法兼容性的问题，不建议用户使用这种用法。同时使用"(+)"存在以下限制：

  - "(+)"只能出现在WHERE子句中。
  - 如果FROM子句中已经对表连接关系进行指定，那么无法继续在WHERE子句中使用"(+)"。
  - "(+)"只能作用在表或者视图的列上，不能作用在表达式上。
  - 如果表存在多个连接条件，那么必须在所有的连接条件中指定"(+)"，否则"(+)"将不会生效，表连接会转化成内连接，并且不显示任何提示信息。
  - "(+)"作用的连接条件中的表不能跨查询或者子查询。如果"(+)"作用的表，不在当前查询或者子查询的FROM子句中，则会报错。如果"(+)"作用的对端的表不存在，则不报错，同时连接关系会转化为内连接。
  - 除同一单表的外连接表达式外，"(+)"作用的表达式不可直接通过OR操作符连接。
  - 如果"(+)"作用的列是和一个常量的比较关系，那么这个表达式会成为JOIN条件的一部分。
  - 当一张表被同时作为多张表的外表时，若其对应的内表之间没有连接关系，则内表之间先做笛卡尔积，再和外表做外连接。
  - "(+)"只能出现在比较表达式、BETWEEN AND/NOT BETWEEN AND表达式、NOT表达式、ANY/SOME表达式、ALL表达式、IN/NOT IN表达式、NULLIF表达式、IS DISTINCT FROM/IS NOT DISTINCT FROM表达式、IS NULL/IS NOT NULL表达式或IS OF表达式中。"(+)"不能出现在其他类型表达式中，并且这些表达式中不允许出现通过AND和OR连接的表达式。
  - "(+)"只能转化为左外连接或者右外连接，不能转化为全连接，即不能在一个表达式的两个表上同时指定"(+)"。
  - 在ORA兼容模式下，"(+)"操作符括号中可以包含空格，例如："(+)"、"(+)"。

版权所有 © 华为云计算技术有限公司

- "(+)"操作符不支持自动参数化处理。

版权所有 © 华为云计算技术有限公司

须知

- 对于WHERE子句的LIKE操作符，当LIKE中要查询特殊字符“%”、“_”、“\”时，需要使用“\”来进行转义。在ORA兼容模式下，可以在GUC参数standard_conforming_strings为on时，设置behavior_compat_options参数的取值为ignore_like_default_escape，以支持将反斜杠字符“\”作为普通字符匹配。

```
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db;
ora_db=# SET standard_conforming_strings = on;
ora_db=# SET behavior_compat_options = ignore_like_default_escape;
ora_db=# SELECT 'aa\a' LIKE 'aa\a';
 ?column?
----------
 t
(1 row)
ora_db=# SELECT 'aaa' LIKE 'aa\a';
 ?column?
----------
 f
(1 row)
ora_db=# SET behavior_compat_options = '';
ora_db=# SELECT 'aa\a' LIKE 'aa\a';
 ?column?
----------
 f
(1 row)
ora_db=# SELECT 'aaa' LIKE 'aa\a';
 ?column?
----------
 t
(1 row)
ora_db=# \c postgres;
gaussdb=# DROP DATABASE ora_db;
```

- WHERE子句后面添加IN子句且当IN后面不添加括号时，支持IN后面只添加一个元素，约束仅支持在ORA兼容模式数据库下使用，并且此时WHERE子句中仅支持存在这一个条件。该元素有如下限制：
  - 常量：如字符串、整数、浮点数等。
  - 列名。
  - CASE WHEN子句。
  - 函数。
- "(+)"操作符支持在提取公共表达式后执行合法性校验。公共表达式定义为完全相同的表达式结构，包括表达式内的顺序、表达式内的占位符序号等。

```
SELECT * FROM a,a2 WHERE a.id(+) IS NULL AND a.id(+) = a2.id OR a.id(+) = a2.id AND a.id(+) = 1;
```

  提取公共表达式a.id(+) = a2.id后，可转化为：

```
SELECT * FROM a,a2 WHERE a.id(+) = a2.id AND (a.id(+) IS NULL OR a.id(+) = 1);
```

  转化后的表达式满足OR操作符两侧均为同一单表的外连接表达式，因此校验成功。

```
SELECT * FROM a,a2 WHERE a.id(+) IS NULL AND a2.id = a.id(+) OR a.id(+) = a2.id AND a.id(+) = 1;
```

  由于a2.id = a.id(+)与a.id(+) = a2.id的表达式顺序不同，无法提取公共表达式，因此校验失败。

  在PREPARE语句或通过游标动态执行查询语句中，使用占位符绑定参数时，需满足公共表达式中的占位符序号完全相同。

```
PREPARE p1 AS SELECT * FROM a,a2 WHERE a.id IS NULL AND a.id(+) = $1 OR a.id(+) = $2 AND a.id = 1;
```


版权所有 © 华为云计算技术有限公司

由于a.id(+) = $1和a.id(+) = $2的占位符序号不同（$1≠$2），不符合公共表达式定义，因此校验失败。

- **START WITH子句**

  START WITH子句通常与CONNECT BY子句同时使用，数据进行层次递归遍历查询，START WITH代表递归的初始条件。若省略该子句，单独使用CONNECT BY子句，则表示以表中的所有行作为初始集合。该功能请参见**CONNECT BY子句**子句。

- **CONNECT BY子句**

  CONNECT BY代表递归连接条件，和START WITH子句同时使用，实现数据遍历递归的功能。

  CONNECT BY条件中可以对列指定PRIOR关键字，表示以列为递归键进行递归。若在递归连接条件前指定NOCYCLE，则表示当出现循环记录时停止递归。

须知

含START WITH condition CONNECT BY子句的SELECT语句不支持使用FOR SHARE或FOR UPDATE锁。

START WITH语句的执行流程是：

a. 由START WITH区域的条件选择初始的数据集。示例中先将('A', 1, 0)选择出来，再将初始的数据集设为工作集。

b. 当工作集不为空，将工作集的数据作为输入，查询下一轮的数据，过滤条件由CONNECT BY区域指定。其中，PRIOR关键字表示当前记录，示例中prior id = fatherid表示当前记录的id是下一条记录的fatherid。

c. 将**上一步**中筛选出来的数据集作为工作集，并重复操作**上一步**。

```
gaussdb=#  CREATE TABLE test(name varchar, id int, fatherid int);
gaussdb=#  INSERT INTO test VALUES('A', 1, 0), ('B', 2, 1),('C',3,1),('D',4,1),('E',5,2);
gaussdb=#  SELECT * FROM test START WITH id = 1 CONNECT BY prior id = fatherid ORDER SIBLINGS BY id DESC;
 name | id | fatherid
------+----+----------
 A    | 1 |        0
 D    | 4 |        1
 C    | 3 |        1
 B    | 2 |        1
 E    | 5 |        2
(5 rows)
```

当前START WITH默认行为是宽度优先搜索，可以通过和**伪列**配合，实现深度优先搜索。如：

```
gaussdb=# SELECT sys_connect_by_path(name,'-') as path, *, LEVEL FROM test START WITH id = 1 CONNECT BY fatherid=prior id ORDER BY path;
  path   | name | id | fatherid | level
---------+------+----+----------+-------
 -A      | A    | 1 |        0 |     1
 -A-B    | B    | 2 |        1 |     2
 -A-B-E  | E    | 5 |        2 |     3
 -A-C    | C    | 3 |        1 |     2
 -A-D    | D    | 4 |        1 |     2
gaussdb=# DROP TABLE test;
```

如果在数据集中存在循环，数据库会提供循环检测。默认行为当检查到循环时，会直接报错，不返回任何数据；若指定NOCYCLE关键字，查询可以正常执行，当出现循环记录时停止递归并退出。

版权所有 © 华为云计算技术有限公司

- 伪列：数据库为每一条选出来的数据添加伪列，方便用户了解数据在递归或者树状结构中的位置。
  - LEVEL：节点的层级，根节点层级为1。
  - CONNECT_BY_ISCYCLE ： 用于返回当前节点是否与父行/子行成环。
  - CONNECT_BY_ISLEAF：是否为叶子节点。当某个节点不存在任何子节点时，其即为叶子节点，CONNECT_BY_ISLEAF会被置为1；否则会被置为0。

  举例：（假设已存在表T1，数据结构如**表1-406**所示。）

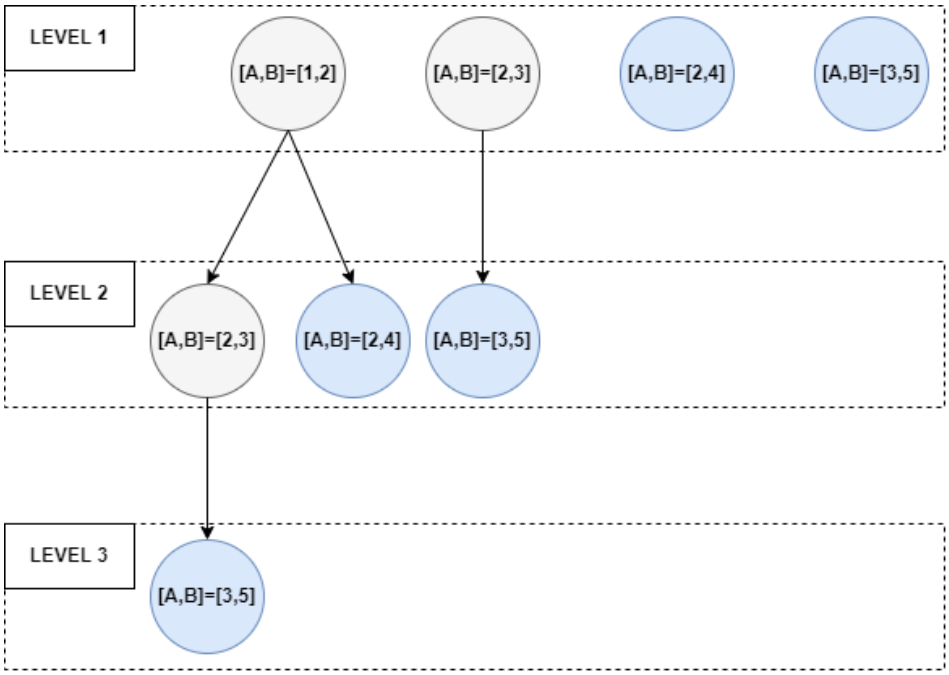
  ```
  SELECT * FROM T1 CONNECT BY PRIOR B=A AND LEVEL<=3;
  ```

  表 1-406 T1 数据结构

A	B
1	2
2	3
2	4
3	5

  结果的树型结构如**图1-15**所示，其中蓝色的节点为叶子节点。

  图 1-15 执行逻辑结构图

- 查询函数：除了伪列之外，数据库还提供了查询函数（具体请参见**层次递归查询函数**）。

版权所有 © 华为云计算技术有限公司

  - sys_connect_by_path(col, separator)：返回从根节点到当前行的连接路径。col为路径中显示的列的名称，separator为连接符。
  - connect_by_root(col)：显示该节点最顶级的节点，col为输出列的名称。
- 除了伪列和查询函数外，在CONNECT BY子句之外还可使用PRIOR关键字修饰列。在SELECT查询列、DISTINCT子句、WHERE子句、GROUP BY子句、HAVING子句、WINDOW子句、ORDER BY子句使用，可显示父行列值。
- 在层次查询过程中，严格按照深度优先搜索的顺序进行。如果在START WITH或CONNECT中使用rownum作为过滤条件，对于每条尝试被返回的记录，rownum会先加1，之后按照rownum相关条件判断；对于不满足的记录，会被丢弃且rownum会减1。

版权所有 © 华为云计算技术有限公司

须知

- PRIOR关键字不能在START WITH语句中指定。
- 除targetlist外，"prior(单列)"会被解析为"prior 单列"，不感知用户自定义的名为prior的函数。
- 只能对表中的列指定PRIOR，不支持对表达式、伪列及类型转换指定PRIOR关键字。
- CONNECT BY语句中，PRIOR修饰的列不可以和level/rownum等伪列在同一个条件里（如(PRIOR a = level)），但可在不同条件里（如(PRIOR a = b) and (level = 1)），不同条件是指CONNECT BY语句最上层的AND连接起来的条件。
- START WITH/CONNECT BY语句中禁止将伪列用于子连接，即类似于"rownum = (子查询)"或"rownum in (子查询)"。
- CONNECT BY子句中不建议使用ROWNUM，如需使用请充分测试，避免结果和预期不一致。
- 在WITH AS定义的CTE上调用START WITH/CONNECT BY时，如果存在多个CTE，需要保证每一个CTE的定义不依赖于其他CTE。
- 如果数据中不存在环路，但是报错runs into cycle，需要考虑增大max_recursive_times。
- connect_by_isleaf、connect_by_iscycle和level类型均为int类型。
- connect_by_isleaf、connect_by_iscycle和level在层次查询下，会被解析为伪列，而不会解析为投影列中别名名称和伪列相同的列。
- 不建议在定义带有层次查询的视图时引用其他视图。
- START WITH调优建议：
  - 根据CONNECT BY中的条件，建立对应的索引，来提高START WITH语句的性能。
  - 根据explain performance或者WDR报告中的计划识别瓶颈点，如果发现Recursive Union的递归部分的算子（内层计划）为Hash JOIN，但是Hash表是针对临时表tmp_result构建或者计划中显示hash表发生物化（batch大于1），可能是work_mem过小导致无法对外层数据表建立Hash表。可以通过调大work_mem参数来提高性能。

    GaussDB会对小数据量的表有优化，把表的结果缓存在 hash 表中来提高性能，此时不需要索引。但是如果数据量超过 work_mem 的限制，该优化会失效，此时可采用建立索引的方式尝试优化。
- START WITH语句中使用空或NULL表示递归初始条件为空，返回空值结果。
- CONNECT BY语句中使用空表示递归条件为空，返回所有符合START WITH子句条件的结果。
- PRIOR修饰的列仅支持如下类型：INT8、INT1、INT2、OID、INT4、BOOL、CHAR、NAME、FLOAT4、FLOAT8、ABSTIME、RELTIME、DATE、DATEA（仅在ORA兼容性下生效）、CASH、TIME、TIMESTAMP、TIMESTAMPTZ、SMALLDATETIME、UUID、INTERVAL、TIMETZ、INT2VECTOR、CLOB、NVARCHAR2、VARCHAR、TEXT、VECTOR、BPCHAR、RAW、BYTEA、NUMERIC、XID、CID以及TID。通过强制类型转换为上述类型的操作无法避开白名单限制，如：PRIOR col::text（其中col不是上述白名单中的类型）。

版权所有 © 华为云计算技术有限公司

- 在CONNECT BY子句之外使用PRIOR关键字修饰列时，需结合GUC参数enable_startwith_prior_enhance值，显示语句行为。关闭参数时使用PRIOR关键字修饰列等同于不添加PRIOR关键字，会显示当前行列值。打开参数时在支持子句（SELECT查询列、DISTINCT子句、WHERE子句、GROUP BY子句、HAVING子句、WINDOW子句、ORDER BY子句）中使用PRIOR关键字修饰列，可显示父行列值。除此外，以下场景语句会报错：
  - 在支持子句（SELECT查询列、DISTINCT子句、WHERE子句、GROUP BY子句、HAVING子句、WINDOW子句、ORDER BY子句）中，PRIOR修饰的列为伪列LEVEL、CONNECT_BY_ISCYCLE、CONNECT_BY_ISLEAF。
  - 在带有CONNECT BY子句的SELECT语句中，ON条件、ORDER SIBLINGS BY子句中使用PRIOR修饰列。
  - 在不带CONNECT BY子句的语句中，使用PRIOR关键字修饰列。
- 当在同一查询语句中多次使用PRIOR修饰不同列时，只有当所有被修饰列的值均相同时，才认为是重复数据，即存在环路。
- level/connect_by_isleaf/connect_by_iscycle列作为层次递归查询函数sys_connect_by_path/connect_by_root入参时，会被解析为伪列并报错。

- **ORDER SIBLINGS BY子句**

  START WITH语句输出时，不同层的数据会依次返回。但是在每一层内部，数据库会自动选择最优的执行路径，所以无法保证实现的顺序。如果用户对最终输出顺序有需求，可以用ORDER SIBLINGS BY子句，用法和ORDER BY子句一样，用于在递归过程中每层内部的排序。

须知

ORDER SIBLINGS BY后的表达式仅支持对普通列、列名偏移量以及对列名的窗口函数调用的方式进行排序，不支持对列名调用START WITH相关系统函数和使用START WITH相关伪列等方式。

- **GROUP BY子句**

  将查询结果按某一列或多列的值分组，值相等的为一组。

  - CUBE ({ expression | (expression [, ...]) } [, ...])

    CUBE是自动对GROUP BY子句中列出的字段进行分组汇总，结果集将包含维度列中各值的所有可能组合，以及与这些维度值组合相匹配的基础行中的聚合值。它会为每个分组返回一行汇总信息， 用户可以使用CUBE来产生交叉表值。如在CUBE子句中给出三个表达式（n = 3），运算结果为$2^n = 2^3 = 8$组。 以n个表达式的值分组的行称为常规行，其余的行称为超级聚集行。

  - GROUPING SETS (grouping_element [, ...])

    GROUPING SETS是GROUP BY子句的高级扩展功能，其允许在单个查询中显式定义多个GROUP BY选项。与CUBE或ROLLUP不同，GROUPING SETS通过精准指定所需的分组集，数据库不需要执行完整CUBE或ROLLUP生成的聚合集合，从而显著提升查询效率。

版权所有 © 华为云计算技术有限公司

说明

使用GROUPING SETS子句时，如果SQL查询语句中含有DISTINCT ON、ORDER BY或多表等值连接（JOIN）等逻辑语句，可能会出现报错。建议将GROUPING SETS子句的计算逻辑封装到内层子查询中，使分组集运算优先独立执行。

```
SELECT DISTINCT ON (a) a, b FROM t1 GROUP BY GROUPING SETS ((a, b), (b)) ORDER BY a, b DESC;
```

可改写为：

```
SELECT DISTINCT ON (a) a, b FROM (SELECT a, b FROM t1 GROUP BY GROUPING SETS ((a, b), (b))) AS sub_t ORDER BY a, b DESC;
```

须知

- 若sql_mode中包含ONLY_FULL_GROUP_BY选项，存下以下情况：
  - SELECT列表中非聚集函数列与GROUP BY字段一致时，不会报错。
  - SELECT列表中非聚集函数列与GROUP BY字段不一致时，存在以下情况：
    - GROUP BY列表中包含主键或唯一非空键时，SQL语句不会报错。
    - SELECT列表中非聚集函数列，在GROUP BY列表或WHERE列表中包含，且当SELECT列表中非聚集函数列在WHERE列表中包含时，WHERE子句中的列需要是等于某一常量的形式，不会报错。
    - 对于同时含有JOIN子句和GROUP BY子句时，存在以下情况：
      - LEFT JOIN

        语法中的ON子句为两列等值判断时，GROUP BY子句中包含左连接表列，不会报错。

        如：将表t1与表t2进行LEFT JOIN操作，若ON子句的条件为t1.col1=t2.col2 ...时，GROUP BY子句只需包含t1.col1。
      - RIGHT JOIN

        语法中的ON子句为两列等值判断时，GROUP BY子句中包含右连接表列，不会报错。

        如：将表t1与表t2进行RIGHT JOIN操作，若ON子句的条件为t1.col1=t2.col2 ...时，GROUP BY子句只需包含t2.col2。
      - INNER JOIN/CROSS JOIN/STRAIGHT_JOIN

        语法中的ON子句为两列等值判断时，GROUP BY子句中包含任一列即可，不会报错。
- 若sql_mode中不包含ONLY_FULL_GROUP_BY选项,无上述约束，以上报错场景均不报错。
  - SELECT同时存在非聚集函数列和聚集函数时，需要与GROUP BY子句同时使用，否则报错。
  - 若非聚集函数列不在GROUP BY列中，且分组后的非聚集函数列包含多条数据，由于执行计划不同，DN数据分布不同，查询出的非聚集函数列数据为分组后的数据中的其中一条。
- GROUP BY子句中不支持非整数常量排序。

- **HAVING子句**

  与GROUP BY子句配合使用，实现选择特殊的组。HAVING子句将组的一些属性与一个常数值比较，只有满足HAVING子句中的逻辑表达式的组才会被展示。

版权所有 © 华为云计算技术有限公司

- **WINDOW子句**

  ```
  WINDOW window_name AS ( window_definition ) [,  ...]
  ```

  - window_name：被窗口定义window_definition所引用的名称。
  - window_definition：

    ```
    [ existing_window_name ]
    [ PARTITION BY expression [, ...] ]
    [ ORDER BY expression [ ASC | DESC | USING operator ] [ NULLS { FIRST | LAST } ] [, ...] ]
    [ frame_clause ]
    ```

    frame_clause：对窗口函数定义一个窗口框架window frame，窗口函数（并非所有）依赖于该框架。

    window frame：表示当前查询行的一组相关行。

    其中frame_clause的形式如下：

    ```
    [ RANGE | ROWS ] frame_start
    [ RANGE | ROWS ] BETWEEN frame_start AND frame_end
    ```

    frame_start和frame_end形式如下：

    ```
    UNBOUNDED PRECEDING
    VALUE PRECEDING
    CURRENT ROW
    VALUE FOLLOWING
    UNBOUNDED FOLLOWING
    ```

    说明

    针对使用到VALUE场景存在如下约束：

    - 被聚集函数/窗口函数操作的表列仅支持INT1、INT2、INT4、INT8、FLOAT4、FLOAT8、NUMERIC、INTERVAL、TIME、TIMETZ、TIMESTAMP、TIMESTAMPTZ、DATEA（仅在ORA兼容性生效）和DATE类型。
    - VALUE类型仅支持INT1、INT2、INT4、INT8、INT16、FLOAT4、FLOAT8、NUMERIC、INTERVAL和字符串（仅支持可以隐式转换为数字的字符串）类型。
    - VALUE的值仅支持为非负数。
    - VALUE当前只支持常量作为上下边界，不支持变量与表达式作为上下边。

- **UNION子句**

  ```
  select_statement UNION [ALL] select_statement
  ```

  - select_statement：不包含ORDER BY、LIMIT或FOR UPDATE子句的SELECT语句。
  - UNION操作符：计算多个select_statement返回行集合的并集。
  - ALL：可选，未指定时，UNION结果不包含重复的行。

  UNION子句有如下约束条件：

  - 多个SELECT语句中列的数量必须相同，每列的数据类型以及顺序必须相同。
  - 同一个SELECT语句中的多个UNION操作符优先级为从左向右依次执行，如需要改变优先级，可使用括号指定顺序。
  - FOR UPDATE不能在UNION的结果或输入中声明。
  - UNION子表达式中可以包含用括号包围的ORDER BY子句或用括号包围的LIMIT子句。

版权所有 © 华为云计算技术有限公司

须知

- 拼接UNION ALL子句时，推荐子句个数<100，超出时需保证实例内存足够，以避免内存不足。
- 在与NULL进行集合操作时，存在以下约束：
  - 只支持NULL本身是投影列或者NULL是FROM语句子查询中的投影列。
  - CTE场景与递归查询场景中，不支持复杂的子查询（DISTINCT语句、GROUP BY语句、ORDER BY语句以及PARTITION BY语句）。

- **INTERSECT子句**

```
select_statement INTERSECT select_statement
```

  - select_statement：不包含FOR UPDATE子句的SELECT语句。
  - INTERSECT操作符：计算多个select_statement返回行集合的交集，并对重复的行进行去重。

  INTERSECT子句存在以下约束条件：

  - 同一个SELECT语句中的多个INTERSECT操作符优先级为从左向右依次执行，如需要改变优先级，可使用括号指定顺序。
  - 当对多个SELECT语句的执行结果进行UNION和INTERSECT操作，INTERSECT优先级高于UNION。

- **EXCEPT子句**

```
select_statement EXCEPT [ ALL ] select_statement
```

  - select_statement：不包含FOR UPDATE子句的SELECT表达式。
  - EXCEPT操作符：计算存在于左边select_statement的输出而不存在于右边select_statement的输出行。同一个SELECT语句中的多个EXCEPT操作符优先级为从左向右依次执行，如需要改变优先级，可使用括号指定顺序。
  - ALL：可选，未指定时，EXCEPT的结果不包含任何重复的行；指定ALL时，一个在左边表中有m个重复而在右边表中有n个重复的行将在结果中出现max(m-n,0)次。

  目前，不支持对EXCEPT的结果或对任何EXCEPT的输入声明FOR UPDATE子句。

- **MINUS子句**

  与EXCEPT子句具有相同的功能和用法。

- **ORDER BY子句**

  对SELECT语句检索得到的数据进行升序或降序排序。ORDER BY子句中不支持非整数常量排序。

  - ORDER BY表达式中如果包含多列，则根据以下行为进行排序：
    - 从左向右依次执行，首先根据最左边的列进行排序，如果值相同，则根据下一个表达式进行比较。
    - 如果对于所有声明的表达式都相同，则按随机顺序返回。
    - 在与DISTINCT关键字一起使用的情况下，ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。
    - 在与GROUP BY子句一起使用的情况下，ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。
  - 对于不与GROUP BY子句一起使用，并且SELECT语句所检索的结果集的列中包含聚合函数的情况：

版权所有 © 华为云计算技术有限公司

- 如果检索的结果集的列中不包含聚合函数，则忽略ORDER BY子句。
- 如果检索的结果集的列中同时包含聚合函数，则仅保留对聚合函数列的排序，忽略其他排序。

– 在ORA兼容模式数据库中，当指定DISTINCT关键字时，支持ORDER BY子句中所引用的列或表达式必须在DISTINCT列表中出现：

- ORDER BY子句支持的表达式范围，需要遵守ORDER BY自身的约束。

  说明

  1. ORDER BY表达式和常量运算，仅支持常量和ORDER BY整体表达式运算，不保证支持常量和非整体表达式的运算：
     ```
     -- 支持。
     -- 建议常量表达式和整体表达式运算时，对整体表达式加括号。
     SELECT DISTINCT c1*c2+c2*c1 FROM t1 ORDER BY c2*c1+c1*c2+1;
     SELECT DISTINCT c1*c2+c2*c1 FROM t1 ORDER BY (c2*c1+c1*c2)+1;
     SELECT DISTINCT c1*c2+c2*c1 FROM t1 ORDER BY 1+(c2*c1+c1*c2);

     -- 不支持。
     -- 常量和非整体表达式运算，常量的位置会影响表达式的结构校验逻辑，不保证正常运算。
     SELECT DISTINCT c1*c2+c2*c1 FROM t1 ORDER BY 1+c2*c1+c1*c2;
     SELECT DISTINCT c1*c2+c2*c1 FROM t1 ORDER BY c2*c1+1+c1*c2;
     ```
  2. 如果ORDER BY和DISTINCT表达式含有多次强制类型转换，数据库会对强制类型转换的表达式进行解析转换。若ORDER BY解析后的表达式在DISTINCT解析后的列表中出现，则校验通过，否则不通过。若DISTINCT表达式和ORDER BY不一致，不保证解析后的结果一致：
     ```
     -- 不支持。
     -- DISTINCT表达式和ORDER BY原始表达式不一致，且解析后的结果也不一致。
     SELECT DISTINCT c1::int::char::int,c2::integer::int3::int4 FROM t1 ORDER BY c2::char::int + c1;

     -- 支持。
     -- DISTINCT表达式和ORDER BY原始表达式不一致，但解析后的结果一致。
     SELECT DISTINCT c1::int::char::int,c2::integer::int3::int4 FROM t1 ORDER BY c2::char::int + c1::char::int;
     SELECT DISTINCT c1::int::int::char::int::int,c2::integer::int3::int4 FROM t1 ORDER BY c2::char::int + c1::int::char::int;
     SELECT DISTINCT c1::int::int::char::char::char::int::int,c2::integer::int3::int4 FROM t1 ORDER BY c2::char::int + c1::int::char::int;
     ```
  3. 不支持函数内部的加法和乘法的交换场景：
     ```
     SELECT DISTINCT BIT_LENGTH(c1+c2) FROM t1 ORDER BY BIT_LENGTH(c2+c1);
     ```
  4. 当DISTINCT表达式和ORDER BY表达式都包含函数别名，仅支持函数属性为STABLE和IMMUTABLE的别名：
     ```
     -- 不支持。
     -- 函数属性为VOLATILE。
     SELECT DISTINCT DBE_RANDOM.GET_VALUE(c1+c2,100) AS c0 FROM t1 ORDER BY c0+1;

     -- 支持。
     -- 函数属性为STABLE。
     SELECT DISTINCT add_numbers(c1,c2) AS c0 FROM t1 ORDER BY c0+1;
     ```
  5. 若ORDER BY表达式后面只有常数，常数含义为DISTINCT表达式的位置，需要满足ORDER BY本身的约束：
     ```
     -- 支持。
     -- 1指的是DISTINCT列表中第一个位置的表达式。
     SELECT DISTINCT c1,c2 FROM test_table ORDER BY 1;

     -- 不支持。
     -- 2指的是DISTINCT列表中第二个位置的表达式，此处DISTINCT列表中仅有一个表达式，会报错“位置2的表达式不存在”。
     SELECT DISTINCT 1 FROM test_table ORDER BY 2;
     ```


版权所有 © 华为云计算技术有限公司

- 针对加法和乘法运算，仅支持加法和乘法左右表达式一致的交换场景，不支持加法和乘法运算含有不一致左右表达式的交换场景。

说明

如果要支持中文拼音排序，需要在初始化数据库时指定编码格式为UTF-8、GB18030、GB18030_2022、GBK或ZHS16GBK。命令如下:

```
initdb –E UTF8 –D ../data –locale=zh_CN.UTF-8
initdb -E GB18030 -D ../data -locale=zh_CN.GB18030
initdb -E GB18030_2022 -D ../data -locale=zh_CN.GB18030
initdb –E GBK –D ../data –locale=zh_CN.GBK
initdb –E ZHS16GBK –D ../data –locale=zh_CN.GBK
```

- **LIMIT子句**

  LIMIT子句由两个独立的子句组成：

  LIMIT { count | ALL }：限制返回行数，count为指定行数，LIMIT ALL的效果和省略LIMIT子句一样。

  OFFSET start count：声明返回的最大行数。count为指定行数，start为指定开始返回行之前忽略的行数。如果同时指定，则会在start行之后才开始计算count个返回行。

  LIMIT子句不支持ROWNUM作为count或者offset。

- **OFFSET子句**

  SQL：2008开始提出一种不同的语法：

  ```
  OFFSET start { ROW | ROWS }
  ```

  start：指定开始返回行之前忽略的行数。

- **FETCH { FIRST | NEXT } [count] { ROW | ROWS } ONLY [OFFSET offset_count]**

  如果不指定count，默认值为1，FETCH子句限定返回查询结果从第一行开始的总行数。

- **FOR UPDATE | SHARE子句**

  – FOR UPDATE：对SELECT检索出的行进行添加排他锁，用于避免检索出的行数据在当前事务结束前被其他事务修改或者删除，即其他的UPDATE、DELETE、SELECT FOR UPDATE操作将被阻塞，直到当前事务结束。

  – FOR SHARE：对SELECT检索出的行进行添加共享锁，用于避免检索出的行数据在当前事务结束前被其他事务修改或者删除，即其他的UPDATE、DELETE、SELECT操作将被阻塞（不阻塞SELECT FOR SHARE），直到当前事务结束。

  – [OF table_name [, ...]]：添加锁的对象。如果明确地指定表名称，则只对指定的表进行添加锁，其他在SELECT中使用的表将不会被锁定；否则将锁定该命令中所有使用的表。如果应用于一个视图或者子查询，同样将锁定所有该视图或子查询中使用到的指定的表。

  为了避免操作等待其他事务提交，可通过以下选项进行设置：

  – NOWAIT：如果被选择的行不能立即添加锁，执行SELECT FOR UPDATE NOWAIT将会立即报错。

  – WAIT n：如果被选择的行不能立即添加锁，等待n秒（其中，n为int类型，取值范围：0 <= n <= 2147483），n秒内获取锁则正常执行，否则报错。

  – SKIP LOCKED：对表添加锁时跳过已经加锁的行，SKIP LOCKED只能跳过行锁，对不同事务中锁与锁不互相阻塞的场景不会跳过，如SELECT FOR SHARE - SELECT FOR SHARE SKIP LOCKED。

版权所有 © 华为云计算技术有限公司

须知

- 多个FOR UPDATE和FOR SHARE子句可以用于为不同的表指定不同的锁定模式。
- 如果一个表中同时出现（或隐含同时出现）在FOR UPDATE和FOR SHARE子句中，则按照FOR UPDATE处理。类似的，同理，如果影响一个表的任意子句中出现了NOWAIT，该表将按照NOWAIT处理。
- 对于FOR UPDATE或FOR SHARE操作，执行计划不能下推的SQL直接返回报错信息；对于执行计划可以下推的，下推到DN执行。
- 对投影列或者WHERE条件中存在rownum的查询不支持FOR UPDATE或FOR SHARE。
- 对于子查询是stream计划的FOR UPDATE或FOR SHARE操作，不支持加锁的同一行被并发更新。
- 对于ORDER BY FOR UPDATE或ORDER BY FOR SHARE操作，stream计划的sort算子和lock算子执行顺序与其他计划有所差别。stream计划是先执行lock再执行sort，其他计划是先执行sort再执行lock。原因是stream计划如果数据不在当前DN，需要重分布数据，回到数据的原始DN去加锁。由于重分布后数据会变得无序，所以最后还要加sort算子。如果先sort后lock，这样会导致原始有序的数据重新变得无序，这时sort就无意义了，可以消除此sort算子。最终计划的执行顺序就由原始的sort -> lock -> sort变为了lock -> sort。
- 对于FOR UPDATE或FOR SHARE操作并发场景，通常使用添加ORDER BY对数据进行排序的方式来避免无响应等异常，在分布式内这种做法不合理，因为DN的加锁顺序无法通过ORDER BY进行确定，另外指定ORDER BY会造成性能开销，所以不建议加ORDER BY解决该问题。
- DATABASE LINK功能不支持SKIP LOCKED语法。
- 分布式场景下不支持对多表FOR UPDATE/SHARE。
- 层次查询语句不支持与该子句同时使用。

- **NLS_SORT**

  按照指定的方式排序。

  取值范围：

  - SCHINESE_PINYIN_M，按照中文拼音排序。
  - generic_m_ci，不区分大小写排序（可选，仅支持纯英文不区分大小写排序）。

须知

目前仅支持中文拼音格式排序和不区分大小写排序。在创建数据库时，需指定编码格式为UTF8、GB18030、GB18030_2022、GBK或ZHS16GBK；如果指定为其他编码，如SQL_ASCII，则可能报错或排序无效。

- **PARTITION子句**

  查询分区表中相应分区的数据。

1.13.19.3.1 简单查询

简单查询指从一个或多个表或视图中检索一个或多个列数据的操作。

版权所有 © 华为云计算技术有限公司

示例：

```
--建表并插入数据。
gaussdb=# CREATE TABLE student(
    sid INT PRIMARY KEY,
    class INT,
    name  VARCHAR(50),
    gender INT CHECK(gender = 0 OR gender = 1) -- 性别：1为男，0为女。
);
gaussdb=# INSERT INTO student (sid, class, name, gender) VALUES (1, 1, 'michael', 0);
gaussdb=# INSERT INTO student (sid, class, name, gender) VALUES (2, 2, 'bob', 1);
gaussdb=# INSERT INTO student (sid, class, name, gender) VALUES (3, 2, 'gary', 0);

--查询部分列。
gaussdb=# SELECT sid, name FROM student;
 sid |  name
-----+---------
   1 | michael
   2 | bob
   3 | gary
(3 rows)

--查询所有列。
gaussdb=# SELECT * FROM student;
 sid | class |  name   | gender
-----+-------+---------+-----
   1 |     1 | michael |   0
   2 |     2 | bob     |   1
   3 |     2 | gary    |   0
(3 rows)

--给列取别名。
gaussdb=# SELECT sid student_id, name FROM student;
 student_id |  name
------------+---------
          1 | michael
          2 | bob
          3 | gary
(3 rows)

--删除表。
gaussdb=# DROP TABLE student;
```

1.13.19.3.2 条件查询

条件查询指查询出符合条件的数据。

示例：

```
--建表并插入数据。
gaussdb=# CREATE TABLE test_grade(
    id INT,
    name VARCHAR(20),
    score FLOAT       -- score表示成绩。
);
gaussdb=# INSERT INTO test_grade VALUES (1,'Scott',90),(2,'Jack',87.5),(3,'Ben',48);

--查询出成绩不及格的学生信息。
gaussdb=# SELECT * FROM test_grade WHERE score < 60;
 id | name | score
----+------+-------
  3 | Ben  |    48
(1 row)

--删除表。
gaussdb=# DROP TABLE test_grade;
```


版权所有 © 华为云计算技术有限公司

1.13.19.3.3 分组查询

分组查询通常用于配合聚合函数，查询分类统计的信息。常见的聚合函数有总行数count()、求和sum()、平均值avg()、最小值min()、最大值max()。一般和GROUP BY联合使用。

```
--建表并插入数据。
gaussdb=# CREATE TABLE student (id INT,name VARCHAR(20),class INT);
gaussdb=# INSERT INTO student VALUES (1, 'Scott', 1), (2, 'Ben', 1);
gaussdb=# INSERT INTO student VALUES (3, 'Jack', 2),(4, 'Anna', 2),(5, 'Judy', 2);
gaussdb=# INSERT INTO student VALUES (6, 'Sally', 3), (7, 'Jordan', 3);

--使用聚合函数COUNT()统计每个班级有多少人。
gaussdb=# SELECT class, count(*) AS cnt FROM student GROUP BY class;
 class | cnt
-------+-----
     2 |   3
     1 |   2
     3 |   2
(3 rows)

--使用HAVING子句过滤人数少于3个的班级数据。
gaussdb=# SELECT class, COUNT(*) AS num FROM student GROUP BY class HAVING COUNT(*) < 3;
 class | num
-------+-----
     1 |   2
     3 |   2
(2 rows)

--删除表。
gaussdb=# DROP TABLE student;
```

sql_mode中包含ONLY_FULL_GROUP_BY选项，用于校验GROUP BY 列表与SELECT列表。

```
--建表并插入数据。
gaussdb=# CREATE TABLE test (
          id int  primary key NOT NULL,
          id_2 int unique NOT NULL,
          s1 varchar(10),
          s2 int,
          s3 varchar(12)
        );
gaussdb=# INSERT INTO test value(1,101,'a',1000,'aa');
gaussdb=# INSERT INTO test value(2,102,'b',2000,'aaa');
gaussdb=# INSERT INTO test value(3,103,'c',3000,'aaaa');
gaussdb=# INSERT INTO test value(4,104,'b',4000,'aaa');
gaussdb=# INSERT INTO test value(5,105,'c',5000,'aaaaa');
gaussdb=# CREATE TABLE tb1(a int, b int, c int);
gaussdb=# INSERT INTO tb1 VALUES(1, 4, 11);
gaussdb=# INSERT INTO tb1 VALUES(2, 5, 12);
gaussdb=# INSERT INTO tb1 VALUES(2, 6, 13);
gaussdb=# INSERT INTO tb1 VALUES(2, 2, 13);
gaussdb=# CREATE TABLE tb2(a int, b int);
gaussdb=# INSERT INTO tb2 VALUES(2, 7);
gaussdb=# INSERT INTO tb2 VALUES(2, 8);
gaussdb=# INSERT INTO tb2 VALUES(3, 9);

--设置sql_mode。
gaussdb=# SET sql_mode =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_
BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';

--SELECT列表中非聚集函数列与GROUP BY字段一致时，不会报错。
gaussdb=# SELECT s1, s3 FROM test GROUP BY s1,s3 ORDER BY s1,s3;
 s1 |  s3
----+-------
 a  | aa
 b  | aaa
```

 版权所有 © 华为云计算技术有限公司

```
 c | aaaa
 c | aaaaa
(4 rows)

gaussdb=# SELECT s1, s3 FROM test GROUP BY s3 ORDER BY s3;
ERROR:  Column "test.s1" must be used in the GROUP BY clause or aggregate function.
LINE 1: SELECT s1, s3 FROM test GROUP BY s3 ORDER BY s3;
               ^
--SELECT列表中非聚集函数列与GROUP BY字段不一致时，分两种情况：
--情况一：GROUP BY列表包含主键或唯一非空键时，SQL语句不会报错。
gaussdb=# SELECT s1, MAX(LENGTH(s3)) FROM test GROUP BY id ORDER BY id;
 s1 | max
----+-----
 a |   2
 b |   3
 c |   4
 b |   3
 c |   5
(5 rows)

--情况二：SELECT列表中非聚集函数列，都出现GROUP BY列表或WHERE列表中，且WHERE子句中的列需要等于某一常量。
gaussdb=# SELECT s1, SUM(s2) FROM test WHERE s1 = 'b' GROUP BY s1;
 s1 | sum
----+------
 b  | 6000
(1 row)

--设置sql_mode,取消ONLY_FULL_GROUP_BY选项。
gaussdb=# SET SQL_MODE =
'STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';

--SELECT列表中非聚集函数列与GROUP BY字段一致时，不会报错。
gaussdb=# SELECT s1, s3 FROM test GROUP BY s1,s3 ORDER BY s1,s3;
 s1 |  s3
----+-------
 a  | aa
 b  | aaa
 c  | aaaa
 c  | aaaaa
(4 rows)

gaussdb=# SELECT s1, s3 FROM test GROUP BY s3 ORDER BY s3;
 s1 |  s3
----+-------
 a  | aa
 b  | aaa
 c  | aaaa
 c  | aaaaa
(4 rows)

--SELECT列表中非聚集函数列与GROUP BY字段不一致时，分两种情况：
--情况一：GROUP BY列表包含主键或唯一非空键时，SQL语句不会报错。
gaussdb=# SELECT s1, MAX(LENGTH(s3)) FROM test GROUP BY id ORDER BY id;
 s1 | max
----+-----
 a |   2
 b |   3
 c |   4
 b |   3
 c |   5
(5 rows)

--情况二：SELECT列表中非聚集函数列，都出现GROUP BY列表或WHERE列表中，且WHERE子句中的列需要等于某一常量。
gaussdb=# SELECT s1, SUM(s2) FROM test WHERE s1 = 'b' GROUP BY s1;
 s1 | sum
----+------
```


版权所有 © 华为云计算技术有限公司

```
 b  | 6000
(1 row)

--删除。
gaussdb=# DROP TABLE test, tb1, tb2;
```

1.13.19.3.4 分页查询

语法如下：

```
SELECT query_list FROM table_name [ LIMIT { [offset,] count | ALL } ]
SELECT query_list FROM table_name [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY [OFFSET
offset_count]]
```

- offset：表示从第几行向后开始。
- count：表示向后查询几条数据。
- ALL：表示向后查询所有的数据。
- FIRST：表示从结果集开头查询数据，可配合OFFSET从offset_count偏移量处开始查询。
- NEXT：配合OFFSET，表示从offset_count偏移量处开始查询，未指定OFFSET时，从结果集开头查询数据。
- offset_count：偏移量，表示在指定偏移量后查询数据。

```
-- 建表并插入100条数据。
gaussdb=# CREATE TABLE testl(id int PRIMARY KEY, flag varchar(10));
gaussdb=# INSERT INTO testl (id, flag) VALUES (generate_series(1,100),'flag'||generate_series(1,100));

-- 查询前5条数据。
gaussdb=# SELECT * FROM testl ORDER BY 1  LIMIT 5;
 id | flag
----+-------
  1 | flag1
  2 | flag2
  3 | flag3
  4 | flag4
  5 | flag5
(5 rows)

gaussdb=# SELECT * FROM testl ORDER BY 1 FETCH FIRST 5 ROWS ONLY;
 id | flag
----+-------
  1 | flag1
  2 | flag2
  3 | flag3
  4 | flag4
  5 | flag5
(5 rows)

gaussdb=# SELECT * FROM testl ORDER BY 1 FETCH FIRST 2+3 ROWS ONLY;
 id | flag
----+-------
  1 | flag1
  2 | flag2
  3 | flag3
  4 | flag4
  5 | flag5
(5 rows)

gaussdb=# SELECT * FROM testl ORDER BY 1 FETCH FIRST 4.5 ROWS ONLY;
 id | flag
----+-------
  1 | flag1
  2 | flag2
  3 | flag3
  4 | flag4
```


版权所有 © 华为云计算技术有限公司

```
 5 | flag5
(5 rows)

-- 从第20条向后查询4条数据。
gaussdb=# SELECT * FROM testl ORDER BY 1 LIMIT 20,4;
 id |  flag
----+--------
 21 | flag21
 22 | flag22
 23 | flag23
 24 | flag24
(4 rows)

gaussdb=# SELECT * FROM testl ORDER BY 1 FETCH NEXT 4 ROWS ONLY OFFSET 20;
 id |  flag
----+--------
 21 | flag21
 22 | flag22
 23 | flag23
 24 | flag24
(4 rows)

-- 从第96条向后查询出所有数据。
gaussdb=# SELECT * FROM testl ORDER BY 1 LIMIT 96,ALL;
 id  |  flag
-----+---------
  97 | flag97
  98 | flag98
  99 | flag99
 100 | flag100
(4 rows)

-- 删除表。
gaussdb=# DROP TABLE testl;
```

1.13.19.3.5 分区查询

查询指定分区的数据。

示例：

```
-- 创建范围分区表。
gaussdb=# CREATE TABLE test_range1(
    id INT,
    info VARCHAR(20)
) PARTITION BY RANGE (id) (
    PARTITION p1 VALUES LESS THAN (200),
    PARTITION p2 VALUES LESS THAN (400),
    PARTITION p3 VALUES LESS THAN (600),
    PARTITION p4 VALUES LESS THAN (800),
    PARTITION pmax VALUES LESS THAN (MAXVALUE)
);

-- 插入1000数据。
gaussdb=# INSERT INTO test_range1 VALUES(GENERATE_SERIES(1,1000),'abcd');

-- 查询p1分区有多少条数据。
gaussdb=# SELECT COUNT(*) FROM test_range1 PARTITION (p1);
 count
-------
   199
(1 row)

-- 删除表。
gaussdb=# DROP TABLE test_range1;
```

1.13.19.3.6 连接查询

连接查询也可称为跨表查询，需要关联多个表进行查询。

 版权所有 © 华为云计算技术有限公司

```
-- 建表并插入数据。
gaussdb=# CREATE TABLE emp(
    id int,         -- 员工编号。
    name varchar,   -- 员工姓名。
    deptno int      -- 所属部门编号。
);

gaussdb=# CREATE TABLE dept(
    deptno int,     -- 所属部门编号。
    depname varchar -- 部门名称。
);

gaussdb=# INSERT INTO emp VALUES (1, 'Joe', 10), (2, 'Scott', 20), (3, 'Ben', 999);   -- Ben还没有确认是哪一个部门所以部门编号是999。
gaussdb=# INSERT INTO dept VALUES (10, 'hr'), (20, 'it'), (30, 'sal');                -- sal部门没有员工。
```

- 内连接（INNER JOIN）

```
gaussdb=# SELECT t1.id,t1.name,t2.depname FROM emp t1 JOIN dept t2 ON t1.deptno = t2.deptno;
 id | name  | depname
----+-------+---------
  1 | Joe   | hr
  2 | Scott | it
(2 rows)
```

- 左连接（LEFT JOIN）

```
gaussdb=# SELECT t1.id,t1.name,t2.depname FROM emp t1 LEFT JOIN dept t2 ON t1.deptno =
t2.deptno;
 id | name  | depname
----+-------+---------
  1 | Joe   | hr
  2 | Scott | it
  3 | Ben   |
(3 rows)
```

- 右连接（RIGHT JOIN）

```
gaussdb=# SELECT t1.id,t1.name,t2.depname FROM emp t1 RIGHT JOIN dept t2 ON t1.deptno =
t2.deptno;
 id | name  | depname
----+-------+---------
  1 | Joe   | hr
  2 | Scott | it
    |       | sal
(3 rows)
```

- 全连接（FULL JOIN）

```
gaussdb=# SELECT t1.id,t1.name,t2.depname FROM emp t1 FULL JOIN dept t2 ON t1.deptno =
t2.deptno;
 id | name  | depname
----+-------+---------
  1 | Joe   | hr
  2 | Scott | it
    |       | sal
  3 | Ben   |
(4 rows)

-- 删除表。
gaussdb=# DROP TABLE emp,dept;
```

- 多表连接

```
gaussdb=# CREATE TABLE t1(a int, b int, c int);
gaussdb=# CREATE TABLE t2(a int, b int, c int);
gaussdb=# CREATE TABLE t3(a int, b int, c int);

gaussdb=# INSERT INTO t1 VALUES(1,generate_series(1,10),generate_series(1,20));
gaussdb=# INSERT INTO t2 VALUES(generate_series(1,5),generate_series(1,10),generate_series(1,20));
gaussdb=# INSERT INTO t3
VALUES(generate_series(6,10),generate_series(1,20),generate_series(1,10));

gaussdb=# SELECT * FROM t1 LEFT JOIN t2 ON t1.a =t2.a INNER JOIN t3 ON t1.b = t3.b AND t2.c =
t3.c;
 a | b | c  | a | b | c | a | b | c
```

 版权所有 © 华为云计算技术有限公司

```
---+---+----+---+---+---+---+---+---
 1 | 1 |  1 | 1 | 1 | 1 | 6 | 1 | 1
 1 | 6 |  6 | 1 | 6 | 6 | 6 | 6 | 6
 1 | 1 | 11 | 1 | 1 | 1 | 6 | 1 | 1
 1 | 6 | 16 | 1 | 6 | 6 | 6 | 6 | 6
(4 rows)

gaussdb=# DROP TABLE t1,t2,t3;
```

1.13.19.3.7 子查询

一个查询可以嵌套在另一个查询中，其结果作为另一个查询的数据来源或判断条件。其中外层查询也叫父查询，内层查询也叫子查询。

- 按照子查询的返回记录数分类可分为单行子查询，多行子查询。
- 按照子查询是否被执行多次分类可分为关联子查询，非关联子查询。

单行子查询

单行子查询操作符有>=、>、<=、<、<>。

```
-- 创建学生表并插入数据。
gaussdb=# CREATE TABLE student(
    sid VARCHAR(5),     -- 学号
    grade INT,          -- 年级
    name VARCHAR(20),   -- 姓名
    height INT          -- 身高
);
gaussdb=# INSERT INTO student VALUES ('00001',1,'Scott',135),('00002',1,'Jack',95),('00003',1,'Ben',100);
gaussdb=# INSERT INTO student VALUES ('00004',2,'Henry',115),('00005',2,'Jordan',130),
('00006',2,'Bob',126);
gaussdb=# INSERT INTO student VALUES ('00007',3,'Bella',128),('00008',3,'Alicia',136);
-- 创建老师表并插入数据。
gaussdb=# CREATE TABLE teacher (
    name VARCHAR(20),    -- 教师姓名
    grade INT            -- 班级
);
-- 插入数据
gaussdb=# INSERT INTO teacher VALUES ('Bill',1),('Sally',2),('Luke',3);

-- 查询出身高比Bella高的学生。
gaussdb=# SELECT * FROM student
WHERE height > (SELECT height FROM student WHERE name = 'Bella');
  sid  | grade |  name  | height
-------+-------+--------+--------
 00001 |     1 | Scott  |    135
 00005 |     2 | Jordan |    130
 00008 |     3 | Alicia |    136
(3 rows)
```

多行子查询

多行子查询操作符：

- in：等于列表中的任意一个。
- any：需要和单行比较符一起使用，和子查询返回的任意值比较。
- all：需要和单行比较符一起使用，和子查询返回的所有值比较。
- some：any的别名，作用相同。

示例：查询出Sally和Luke的学生

```
gaussdb=# SELECT * FROM student t1 WHERE t1.grade IN (
    SELECT grade FROM teacher WHERE name = 'Sally' OR name = 'Luke'
```


版权所有 © 华为云计算技术有限公司

```
);
 sid  | grade |  name  | height
-------+-------+--------+--------
 00004 |     2 | Henry  |    115
 00005 |     2 | Jordan |    130
 00006 |     2 | Bob    |    126
 00007 |     3 | Bella  |    128
 00008 |     3 | Alicia |    136
(5 rows)
```

示例：查询出2年级比3年级任意一个人都高的学生。

```
gaussdb=# SELECT * FROM student
WHERE grade = 2 AND
    height > ANY (SELECT height FROM student WHERE grade = 3);
 sid  | grade |  name  | height
-------+-------+--------+--------
 00005 |     2 | Jordan |    130
(1 row)

-- 查询结果相当于：
gaussdb=# SELECT * FROM student
WHERE grade = 2 AND
    height > (SELECT MIN(height) FROM student WHERE grade = 3);
```

示例：查询出1年级比2年级所有人都高的学生。

```
gaussdb=# SELECT * FROM student
WHERE grade = 1 AND
    height > ALL (SELECT height FROM student WHERE grade = 2);
 sid  | grade | name  | height
-------+-------+-------+--------
 00001 |     1 | Scott |    135
(1 row)

-- 查询结果相当于：
gaussdb=# SELECT * FROM student
WHERE grade = 1 AND
    height > (SELECT MAX(height) FROM student WHERE grade = 2);
```

关联子查询

特点：子查询不能单独运行，是和父查询相关的。先执行父查询，再执行子查询。每执行一次父查询，子查询都要重新计算一次。

示例：查询身高大于本班级平均身高的学生。

```
gaussdb=# SELECT * FROM student out
WHERE height > (SELECT AVG(height) FROM student
              WHERE grade = out.grade);
 sid  | grade |  name  | height
-------+-------+--------+--------
 00001 |     1 | Scott  |    135
 00005 |     2 | Jordan |    130
 00006 |     2 | Bob    |    126
 00008 |     3 | Alicia |    136
(4 rows)
```

非关联子查询

特点：子查询先将值查询出来，再返回给外层查询。

示例：查询身高大于本班级平均身高的学生。

```
gaussdb=# SELECT t1.*
FROM student t1,
    (SELECT grade, AVG(height) avg_hei FROM student GROUP BY grade) t2
```


版权所有 © 华为云计算技术有限公司

```
WHERE t1.grade = t2.grade AND
      t1.height > t2.avg_hei;
 sid  | grade |  name  | height
-------+-------+--------+--------
 00001 |     1 | Scott  |    135
 00005 |     2 | Jordan |    130
 00006 |     2 | Bob    |    126
 00008 |     3 | Alicia |    136
(4 rows)

-- 删除表。
gaussdb=# DROP TABLE student;
gaussdb=# DROP TABLE teacher;
```

1.13.19.3.8 层次查询

语法格式

```
SELECT { { column | LEVEL | CONNECT_BY_ISLEAF | CONNECT_BY_ISCYCLE | SYS_CONNECT_BY_PATH(col,
separator) | CONNECT_BY_ROOT(col) } [, ...] }
 FROM table
START WITH condition
CONNECT BY [NOCYCLE] [PRIOR column1 = column2|column1 = prior column2]
[ORDER SIBLINGS BY expression] [ORDER BY expression]
```

参数说明

- LEVEL：伪列, 用于返回查询的层次。
- CONNECT_BY_ISLEAF: 伪列，用于返回当前节点是否为叶子节点。
- CONNECT_BY_ISCYCLE ： 伪列，用于返回当前节点是否与父行/子行成环。
- NOCYCLE：指定该选项时，遇到循环记录时停止递归。未指定该选项时，遇到循环记录时报错。
- SYS_CONNECT_BY_PATH(col, separator)：用于层次查询的函数，返回从根节点到当前行的连接路径。参数col为路径中显示的列的名称，separator为连接符。
- CONNECT_BY_ROOT(col)：用于层次查询的函数，显示该节点最顶级的节点，col为输出列的名称。
- START WITH：用于指定层次关系，即查询的根行。
- CONNECT BY：用来指定父行和子行的关系。
- PRIOR：用于指定哪一个是父级别。
- ORDER SIBLINGS BY：指定同一层级之间的排列顺序。

示例

```
-- 建表并插入数据。
gaussdb=# CREATE TABLE area (id INT,name VARCHAR(25),parent_id INT);
gaussdb=# INSERT INTO area VALUES (1,'中国',NULL);
gaussdb=# INSERT INTO area VALUES (2,'北京市',1);
gaussdb=# INSERT INTO area VALUES (3,'朝阳区',2);
gaussdb=# INSERT INTO area VALUES (4,'陕西省',1);
gaussdb=# INSERT INTO area VALUES (5,'西安市',4);
gaussdb=# INSERT INTO area VALUES (6,'雁塔区',5);
gaussdb=# INSERT INTO area VALUES (7,'未央区',5);

-- 层次查询。
gaussdb=# SELECT level,
    name,
    connect_by_isleaf,
    connect_by_iscycle,
    connect_by_root(name),
```


版权所有 © 华为云计算技术有限公司

```
    sys_connect_by_path(name, '->')
FROM area START WITH (id = 1) CONNECT BY PRIOR id = parent_id;
 level |  name  | connect_by_isleaf | connect_by_iscycle | connect_by_root |      sys_connect_by_path
-------+--------+-------------------+--------------------+-----------------+-------------------------------
     1 | 中国   |                 0 |                  0 | 中国            | ->中国
     2 | 陕西省 |                 0 |                  0 | 中国            | ->中国->陕西省
     3 | 西安市 |                 0 |                  0 | 中国            | ->中国->陕西省->西安市
     4 | 未央区 |                 1 |                  0 | 中国            | ->中国->陕西省->西安市->未央区
     4 | 雁塔区 |                 1 |                  0 | 中国            | ->中国->陕西省->西安市->雁塔区
     2 | 北京市 |                 0 |                  0 | 中国            | ->中国->北京市
     3 | 朝阳区 |                 1 |                  0 | 中国            | ->中国->北京市->朝阳区
(7 rows)

-- 使用WHERE过滤掉一个节点（仅北京市）。
gaussdb=# SELECT level, name FROM area
    WHERE id <> 2
    START WITH (id = 1)
    CONNECT BY PRIOR id = parent_id;
 level |  name
-------+--------
     1 | 中国
     2 | 陕西省
     3 | 西安市
     4 | 未央区
     4 | 雁塔区
     3 | 朝阳区
(6 rows)

-- 层次查询遇到循环记录
gaussdb=# INSERT INTO area VALUES (8,'江苏省',10);
gaussdb=# INSERT INTO area VALUES (9,'南京市',8);
gaussdb=# INSERT INTO area VALUES (10,'雨花台区',9);
gaussdb=# SELECT level,
    name,
    connect_by_isleaf,
    connect_by_iscycle,
    connect_by_root(name),
    sys_connect_by_path(name, '->')
FROM area START WITH (id = 8) CONNECT BY NOCYCLE PRIOR id = parent_id;
 level |   name   | connect_by_isleaf | connect_by_iscycle | connect_by_root |    sys_connect_by_path
-------+----------+-------------------+--------------------+-----------------+---------------------------
     1 | 江苏省   |                 0 |                  0 | 江苏省          | ->江苏省
     2 | 南京市   |                 0 |                  0 | 江苏省          | ->江苏省->南京市
     3 | 雨花台区 |                 1 |                  1 | 江苏省          | ->江苏省->南京市->雨花台区
(3 rows)

gaussdb=# DELETE FROM area WHERE id >= 8;

-- 在CONNECT BY PRIOR子句中添加条件，过滤掉一个分支（北京市及其子地区）。
gaussdb=# SELECT level, name FROM area
    START WITH (id = 1)
    CONNECT BY PRIOR id = parent_id AND id <> 2;
 level |  name
-------+--------
     1 | 中国
     2 | 陕西省
     3 | 西安市
     4 | 未央区
     4 | 雁塔区
(5 rows)

-- 在SELECT查询列中添加PRIOR修饰id列，显示父行元组。
gaussdb=# SELECT level, PRIOR id as pid, parent_id, name FROM area
    START WITH (id = 1)
    CONNECT BY PRIOR id = parent_id;
 level | pid | parent_id |  name
-------+-----+-----------+--------
     1 |     |           | 中国
     2 |   1 |         1 | 陕西省
```


版权所有 © 华为云计算技术有限公司

```
   3 |  4 |       4 | 西安市
   4 |  5 |       5 | 未央区
   4 |  5 |       5 | 雁塔区
   2 |  1 |       1 | 北京市
   3 |  2 |       2 | 朝阳区
(7 rows)

-- 删除表。
gaussdb=# DROP TABLE area;
```

1.13.19.3.9 复合查询

包含复合运算符的查询，即复合查询。所有的复合查询都具有相同的优先级，参加集合操作的各查询结果的列数、表达式的数量都必须一致，类型必须兼容。

常见的集合运算有：

- UNION：两个查询结果集的并集，对结果进行去重。
- UNION ALL：两个查询的并集，只将两个查询的结果合并。
- INTERSECT：两个查询的交集（只查询重复的数据）。
- MINUS：查询用于获得两个查询结果集的差集，只会显示第一个结果集中存在，但第二个结果集中不存在的数据，并且会以第一列结果进行排序。

示例

```
-- 建表并插入数。
gaussdb=# CREATE TABLE test1(c11 INT, c12 VARCHAR);
gaussdb=# INSERT INTO test1 VALUES (1,'a'),(2,'b'),(4,'d');

gaussdb=# CREATE TABLE test2(c21 INT, c22 VARCHAR);
gaussdb=# INSERT INTO test2 VALUES (1,'a'),(3,'c');
```

- UNION

```
gaussdb=# SELECT * FROM test1 UNION SELECT * FROM test2;
 c11 | c12
-----+-----
   1 | a
   4 | d
   2 | b
   3 | c
(4 rows)
```

- UNION ALL

```
gaussdb=# SELECT * FROM test1 UNION ALL SELECT * FROM test2;
 c11 | c12
-----+-----
   1 | a
   2 | b
   4 | d
   1 | a
   3 | c
(5 rows)
```

- INTERSECT

```
gaussdb=# SELECT * FROM test1 INTERSECT SELECT * FROM test2;
 c11 | c12
-----+-----
   1 | a
(1 row)
```

- MINUS

```
gaussdb=# SELECT * FROM test1 MINUS SELECT * FROM test2;
 c11 | c12
-----+-----
   4 | d
```


版权所有 © 华为云计算技术有限公司

```
  2 | b
(2 rows)
```

```
-- 删除表。
gaussdb=# DROP TABLE test1,test2;
```

1.13.19.3.10 行转列与列转行

- **行转列**

```
-- 建表并插入数据。
gaussdb=# CREATE TABLE test_p2(id INT, class VARCHAR(20), score INT);
gaussdb=# INSERT INTO test_p2 VALUES (1,'math',64), (1,'english',78);
gaussdb=# INSERT INTO test_p2 VALUES (2,'math',98), (2,'english',82);
gaussdb=# INSERT INTO test_p2 VALUES (3,'math',68), (3,'english',59);
gaussdb=# SELECT * FROM test_p2;
 id |  class  | score
----+---------+-------
  3 | math    |    68
  3 | english |    59
  1 | math    |    64
  1 | english |    78
  2 | math    |    98
  2 | english |    82
(6 rows)

-- 行转列。
gaussdb=# SELECT * FROM test_p2 PIVOT(MAX(score) FOR class IN('math','english'));
 id | 'math' | 'english'
----+--------+-----------
  3 |     68 |        59
  1 |     64 |        78
  2 |     98 |        82
(3 rows)

-- 删除表。
gaussdb=# DROP TABLE test_p2;
```

- **列转行**

```
-- 建表并插入数据。
gaussdb=# CREATE TABLE test_p1(id INT, math INT, english INT);
gaussdb=# INSERT INTO test_p1 VALUES (1,84,78), (2,98,82), (3,68,59);
gaussdb=# SELECT * FROM test_p1;
 id | math | english
----+------+---------
  3 |   68 |      59
  1 |   84 |      78
  2 |   98 |      82
(3 rows)

-- 列转行。
gaussdb=# SELECT * FROM test_p1 UNPIVOT(score FOR class IN(math, english));
 id |  class  | score
----+---------+-------
  3 | MATH    |    68
  3 | ENGLISH |    59
  1 | MATH    |    84
  1 | ENGLISH |    78
  2 | MATH    |    98
  2 | ENGLISH |    82
(6 rows)

-- 删除表。
gaussdb=# DROP TABLE test_p1;
```

1.13.19.3.11 窗口函数查询

窗口函数对数据集中的相关行集执行计算，返回一个结果集。与聚集函数所完成的计算相比，窗口函数并不会使多行被聚集成一个单独的输出行。

版权所有 © 华为云计算技术有限公司

示例：

```
-- 建表并插入数据。
gaussdb=# CREATE TABLE fruit_sale (
  "statistical_date" date,
  "product" varchar(255),
  "year" varchar(5),
  "sales_quantity" numeric(8),
  "amount" numeric(8)
);

gaussdb=# INSERT INTO fruit_sale VALUES ('2024-01-01', '西瓜', '2024', 1721, 253541);
gaussdb=# INSERT INTO fruit_sale VALUES ('2024-01-01', '苹果', '2024', 5559, 269419);
gaussdb=# INSERT INTO fruit_sale VALUES ('2024-02-01', '西瓜', '2024', 4711, 129644);

-- 统计所有水果的销售量。
gaussdb=# SELECT *,SUM(sales_quantity) OVER (PARTITION by null) total_qty FROM fruit_sale;
 statistical_date | product | year | sales_quantity | amount | total_qty
------------------+---------+------+----------------+--------+-----------
 2024-01-01       | 西瓜    | 2024 |           1721 | 253541 |     11991
 2024-01-01       | 苹果    | 2024 |           5559 | 269419 |     11991
 2024-02-01       | 西瓜    | 2024 |           4711 | 129644 |     11991
(3 rows)

-- 删除表。
gaussdb=# DROP TABLE fruit_sale;
```

1.13.19.3.12 简化版查询

简化查询，可以不用SELECT关键字查询表的数据。

示例：

```
-- 创建表。
gaussdb=# CREATE TABLE t1(c1 int, c2 int, c3 int);

-- 插入数据。
gaussdb=# INSERT INTO t1 VALUES (1,2,3);
 c1 | c2 | c3
----+----+----
  1 |  2 |  3
(1 row)

-- 查询表的数据。
gaussdb=# TABLE t1;
 c1 | c2 | c3
----+----+----
  1 |  2 |  3
(1 row)

-- 删除表。
gaussdb=# DROP TABLE t1;
```

1.13.19.3.13 Hint 查询

使用Hint可以修改SQL的执行计划。

示例：

```
-- 创建表。
gaussdb=# CREATE TABLE t1(c1 int, c2 int, c3 int);
gaussdb=# CREATE TABLE t2(c1 int, c2 int, c3 int);
gaussdb=# CREATE INDEX it1 ON t1(c1,c2);

-- Hint指定扫描方式。
gaussdb=# EXPLAIN SELECT /*+ indexscan(t1 it1)*/* from t1,t2 WHERE t1.c1=t2.c2;
 id |                 operation                  | E-rows | E-width | E-costs
----+--------------------------------------------+--------+---------+---------
```

 版权所有 © 华为云计算技术有限公司

```
 1 | ->  Streaming (type: GATHER)               |     30 |      24 | 61.51
 2 |    ->  Hash Join (3,4)                     |     30 |      24 | 60.26
 3 |       ->  Index Scan using it1 on t1       |     30 |      12 | 44.46
 4 |       ->  Hash                             |     29 |      12 | 15.49
 5 |          ->  Streaming(type: REDISTRIBUTE) |     30 |      12 | 15.49
 6 |             ->  Seq Scan on t2             |     30 |      12 | 14.14
(6 rows)

 Predicate Information (identified by plan id)
-----------------------------------------------
   2 --Hash Join (3,4)
         Hash Cond: (t1.c1 = t2.c2)
(2 rows)

-- Hint指定行数。
gaussdb=# EXPLAIN SELECT /*+ rows(t1 t2 #5)*/* FROM t1,t2;
 id |              operation               | E-rows | E-width | E-costs
----+--------------------------------------+--------+---------+---------
 1 | ->  Streaming (type: GATHER)         |      5 |      24 | 37.60
 2 |    ->  Nested Loop (3,5)             |      5 |      24 | 37.35
 3 |       ->  Streaming(type: BROADCAST) |     90 |      12 | 17.93
 4 |          ->  Seq Scan on t1          |     30 |      12 | 14.14
 5 |       ->  Materialize                |     30 |      12 | 14.21
 6 |          ->  Seq Scan on t2          |     30 |      12 | 14.14
(6 rows)

-- 删除表。
gaussdb=# DROP TABLE t1;
gaussdb=# DROP TABLE t2;
```

1.13.19.4 SELECT INTO

功能描述

SELECT INTO用于根据查询结果创建一个新表，并且将查询到的数据插入到新表中。和普通的SELECT不同，SELECT INTO数据并不返回给客户端。新表的字段具有和SELECT的输出字段相同的名称和数据类型。

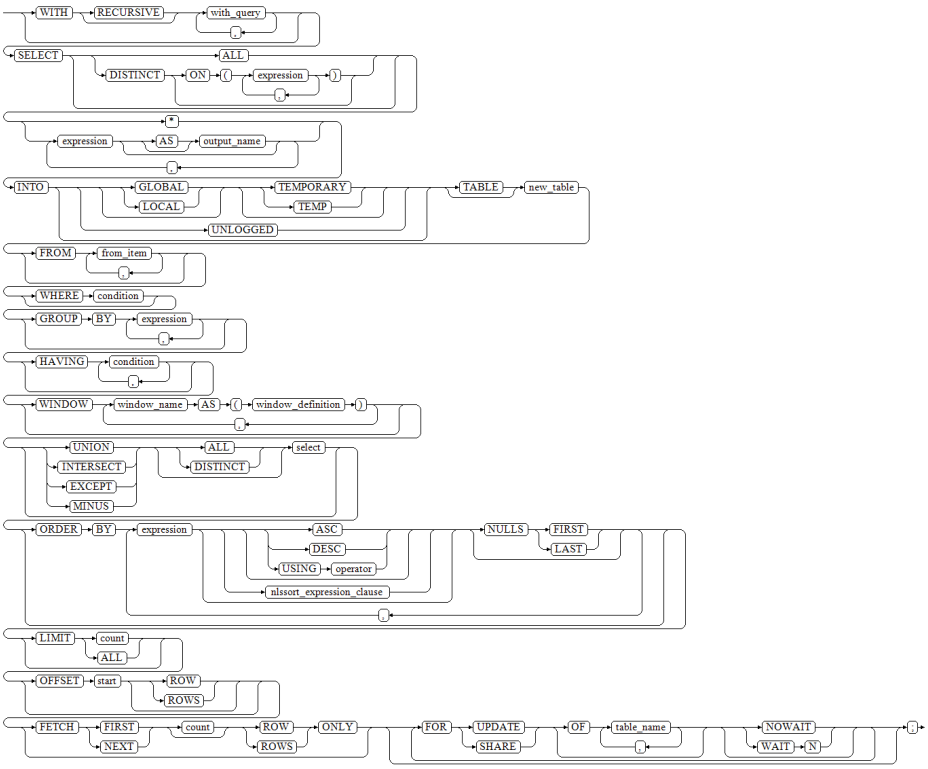
注意事项

CREATE TABLE AS的作用和SELECT INTO类似，且提供了SELECT INTO所提供功能的超集。建议使用CREATE TABLE AS语法替代SELECT INTO，因为SELECT INTO不能在存储过程中使用。

语法格式

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
    { * | {expression [ [ AS ] output_name ]} [, ...] }
    INTO [ [ GLOBAL | LOCAL ] [ TEMPORARY | TEMP ] | UNLOGGED ] [ TABLE ] new_table
    [ FROM from_item [, ...] ]
    [ WHERE condition ]
    [ GROUP BY expression [, ...] ]
    [ HAVING condition [, ...] ]
    [ WINDOW {window_name AS ( window_definition )} [, ...] ]
    [ { UNION | INTERSECT | EXCEPT | MINUS } [ ALL | DISTINCT ] select ]
    [ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS { FIRST |
LAST } ]} [, ...] ]
    [ LIMIT { count | ALL } ]
    [ OFFSET start [ ROW | ROWS ] ]
    [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]
    [ {FOR { UPDATE | SHARE } [ OF table_name [, ...] ] [ NOWAIT | WAIT N]} [...] ];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **new_table**

  new_table指定新建表的名称。

- **UNLOGGED**

  指定表为非日志表。非日志表中写入的数据不会被写入到预写日志中，比普通表快很多。但是，非日志表在冲突、执行操作系统重启、数据库重启、主备切换、切断电源操作或异常关机后会被全部清空，会造成数据丢失的风险，非日志表中的内容也不会被复制到备用服务器中，在该类表中创建的索引也不会被自动记录。

  - 使用场景：非日志表不能保证数据的安全性，用户应该在确保数据已经做好备份的前提下使用，例如系统升级时进行数据的备份。
  - 故障处理：当异常关机等操作导致非日志表上的索引发生数据丢失时，用户应该对发生错误的索引进行重建。

说明

SELECT INTO的其他参数可参见 **参数说明**。

示例

```
--建表并插入数据。
gaussdb=# CREATE TABLE tbl_person (
  id     integer,
  name   varchar(20),
  gender varchar(5) CHECK(gender = '男' or gender = '女')
);
gaussdb=# INSERT INTO tbl_person VALUES (1, 'Bob', '男'),(2, 'Anne', '女'),(3, 'Jack', '男'),(4, 'Danny', '男'),(5, 'Alice', '女'),(6, 'Susan', '女');

--将person表中所有男生的信息加入到新表中
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT * INTO tbl_man FROM tbl_person WHERE gender = '男';

--查询tbl_man数据。
gaussdb=# SELECT * FROM tbl_man;
 id | name  | gender
----+-------+-----
  1 | Bob   | 男
  3 | Jack  | 男
  4 | Danny | 男
(3 rows)

--删除表。
gaussdb=# DROP TABLE tbl_person, tbl_man;
```

相关链接

SELECT

1.13.19.5 SET

功能描述

用于修改运行时配置参数。

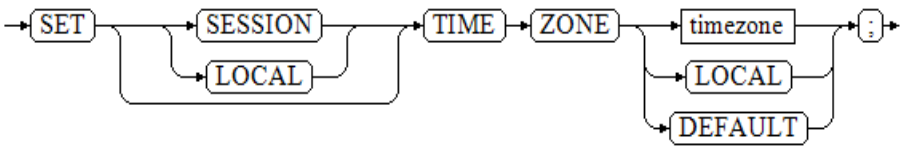
注意事项

大多数运行时参数都可以用SET在运行时设置，但有些则在服务运行过程中或会话开始之后不能修改。

语法格式

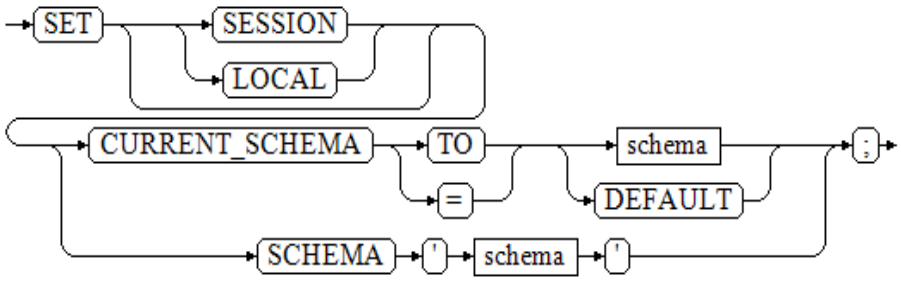
- 设置所处的时区。

```
SET [ SESSION | LOCAL ] TIME ZONE { timezone | LOCAL | DEFAULT };
```



- 设置所属的模式。

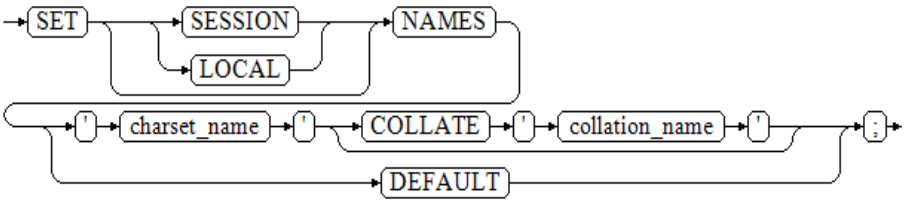
```
SET [ SESSION | LOCAL ]
   {CURRENT_SCHEMA { TO | = } { schema | DEFAULT }
   | SCHEMA 'schema'};
```



- 设置客户端编码集。

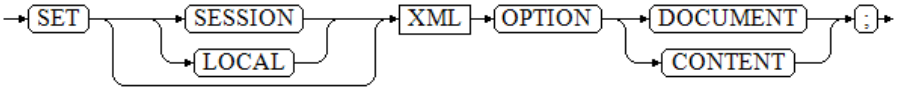
```
SET [ SESSION | LOCAL ] NAMES {'charset_name' [COLLATE 'collation_name'] | DEFAULT};
```


版权所有 © 华为云计算技术有限公司



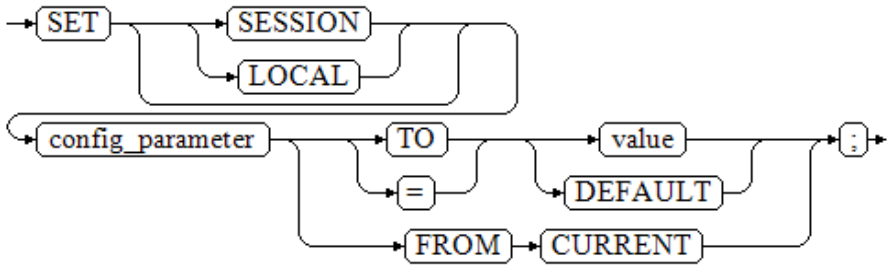
- 设置XML的解析方式。

```
SET [ SESSION | LOCAL ] XML OPTION { DOCUMENT | CONTENT };
```



- 设置其他运行时参数。

```
SET [ SESSION | LOCAL ]
  {config_parameter { { TO | = } { value | DEFAULT }
                    | FROM CURRENT }};
```



参数说明

- **SESSION**

  声明的参数只对当前会话起作用。如果SESSION和LOCAL都没出现，则SESSION为缺省值。

  如果在事务中执行了此命令，命令的产生影响将在事务回滚之后消失。如果该事务已提交，影响将持续到会话的结束，除非被另外一个SET命令重置参数。

- **LOCAL**

  声明的参数只在当前事务中有效。在COMMIT或ROLLBACK之后，会话级别的设置将再次生效。

  不论事务是否提交，此命令的影响只持续到当前事务结束。一个特例是：在一个事务里面，既有SET命令，又有SET LOCAL命令，且SET LOCAL在SET后面，则在事务结束之前，SET LOCAL命令会起作用，但事务提交之后，则是SET命令会生效。

- **TIME ZONE timezone**

  用于指定当前会话的本地时区。

  取值范围：有效的本地时区。该选项对应的运行时参数名称为TimeZone，DEFAULT缺省值为PRC。

  说明

  SET TIME ZONE不支持类似"-12:30"的时、分时区写法，可以使用英文时区、小数写法，如"EST"、"America/New_York"、"-12.5"。

- **CURRENT_SCHEMA schema**

  CURRENT_SCHEMA用于指定当前的模式。

版权所有 © 华为云计算技术有限公司

取值范围：已存在模式名称。

- **SCHEMA schema**

  同CURRENT_SCHEMA。此处的schema是个字符串。

  例如：set schema 'public';

- **NAMES {'charset_name' [COLLATE 'collation_name'] | DEFAULT};**
  - 在sql_compatibility = 'MYSQL'模式下，并设置b_format_version='5.7'、b_format_dev_version='s2'时，支持指定COLLATE子句。

    描述：用于设置客户端字符编码，常量字符串的字符集、字符序，及返回结果的字符集。

    等价为：

    ```
    SET client_encoding = charset_name;
    SET character_set_connection = charset_name;
    SET collation_connection = collation_name;
    SET character_set_results = charset_name;
    ```

    取值范围：MYSQL兼容模式下支持的字符集、字符序。暂不支持指定charset_name与数据库字符集不同。
  - 其他场景下不支持指定COLLATE子句。

    描述：用于设置客户端的字符编码。

    等价为：

    ```
    SET client_encoding TO charset_name;
    ```

    取值范围：有效的字符编码。默认编码为UTF8。

- **XML OPTION option**

  用于设置XML的解析方式。

  取值范围：CONTENT（缺省）、DOCUMENT

- **config_parameter**

  可设置的运行时参数的名称。可用的运行时参数可以使用SHOW ALL命令查看。设置规则请参见**设置参数**。

  说明

  部分通过SHOW ALL查看的参数不能通过SET设置。如max_datanodes。

- **value**

  config_parameter的新值。可以声明为字符串常量、标识符、数字，或者逗号分隔的列表。DEFAULT用于把这些参数设置为它们的缺省值。

- **FROM CURRENT**

  取当前会话中的值设置为config_parameter的值。

示例

```
--设置所处的时区。
gaussdb=# SET TIME ZONE 'PST8PDT';

--设置模式搜索路径。
gaussdb=# SET search_path TO tpcds, public;

--设置客户端编码集。
gaussdb=# SET client_encoding TO utf8;

--设置XML的解析方式。
gaussdb=# SET XML OPTION DOCUMENT;
```


版权所有 © 华为云计算技术有限公司

```
--把日期时间风格设置为传统的 POSTGRES 风格(日在月前)。
gaussdb=# SET datestyle TO postgres;
```

相关链接

RESET，**SHOW**

1.13.19.6 SET CONSTRAINTS

功能描述

SET CONSTRAINTS设置当前事务检查行为的约束条件。

IMMEDIATE约束是在每条语句后面进行检查。DEFERRED约束一直到事务提交时才检查。每个约束都有自己的模式。

从创建约束条件开始，一个约束总是设定为DEFERRABLE INITIALLY DEFERRED、DEFERRABLE INITIALLY IMMEDIATE和NOT DEFERRABLE三个特性之一。第三种总是IMMEDIATE，并且不会受SET CONSTRAINTS影响。前两种以指定的方式启动每个事务，但是其行为可以在事务里用SET CONSTRAINTS改变。

带着一个约束名列表的SET CONSTRAINTS改变这些约束的模式（都必须是可推迟的）。如果有多个约束匹配某个名称，则所有都会被影响。SET CONSTRAINTS ALL改变所有可推迟约束的模式。

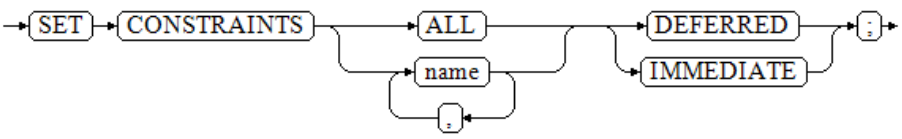
当SET CONSTRAINTS把一个约束从DEFERRED改成IMMEDIATE的时候，任何将在事务结束准备进行的数据修改都将在SET CONSTRAINTS的时候执行检查。如果违反了任何约束，SET CONSTRAINTS都会失败（并且不会修改约束模式）。因此，SET CONSTRAINTS可以用于强制在事务中某一点进行约束检查。检查和唯一约束总是不可推迟的。

注意事项

SET CONSTRAINTS只在当前事务里设置约束的行为。因此，如果用户在事务块之外（START TRANSACTION/COMMIT对）执行这个命令，它将没有任何作用。

语法格式

```
SET CONSTRAINTS { ALL | { name  } [, ...] } { DEFERRED | IMMEDIATE };
```



参数说明

- **name**
  约束名。
  取值范围：已存在的约束名。可以在系统表pg_constraint中查到。
- **ALL**
  所有约束。
- **DEFERRED**

版权所有 © 华为云计算技术有限公司

约束一直到事务提交时才检查。

- **IMMEDIATE**
  约束在每条语句后进行检查。

示例

```
--设置所有约束在事务提交时检查。
gaussdb=# SET CONSTRAINTS ALL DEFERRED;

--设置所有约束在每条语句后进行检查。
gaussdb=# SET CONSTRAINTS ALL IMMEDIATE;
```

1.13.19.7 SET ROLE

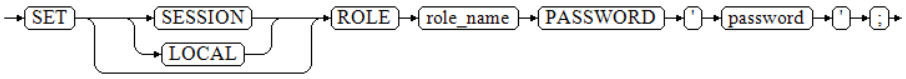
功能描述

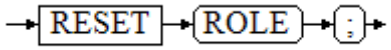
设置当前会话的当前用户标识符。

注意事项

- 当前会话的用户必须是指定的rolename角色的成员，当三权分立关闭时，系统管理员可以选择任何角色。
- 使用这条命令，它可能会增加一个用户的权限，也可能会限制一个用户的权限。如果会话用户的角色有INHERITS属性，则它自动拥有它能SET ROLE变成的角色的所有权限；在这种情况下，SET ROLE实际上是删除了所有直接赋予会话用户的权限，以及它的所属角色的权限，只剩下指定角色的权限。另一方面，如果会话用户的角色有NOINHERITS属性，SET ROLE删除直接赋予会话用户的权限，而获取指定角色的权限。
- 当用户被配置为通过LDAP方式进行登录认证时，无法通过本操作设置用户标识符为该用户。

语法格式

- 设置当前会话的当前用户标识符。
  ```
  SET [ SESSION | LOCAL ] ROLE role_name PASSWORD 'password';
  ```
  

- 重置当前用户标识为当前会话用户标识符。
  ```
  RESET ROLE;
  ```
  

参数说明

- **SESSION**
  声明这个命令只对当前会话起作用，此参数为缺省值。
- **LOCAL**
  声明该命令只在当前事务中有效。
- **role_name**
  角色名。

版权所有 © 华为云计算技术有限公司

取值范围：字符串，数据库中已存在的用户名。

- **password**

  角色的密码。要求符合密码的命名规则。

  说明

  - 使用密文密码限制如下：
    - 管理员用户不能使用密文密码切换到其他管理员用户，只能向权限更低用户切换。
  - 使用密文密码通常用于gs_dump、gs_dumpall导出场景，其他场景不建议直接使用密文密码。

- **RESET ROLE**

  用于重置当前用户标识。

示例

```
--查询当前会话用户，当前用户。
gaussdb=# SELECT SESSION_USER, CURRENT_USER;
 session_user | current_user
--------------+--------------
 omm          | omm
(1 row)

--创建角色paul。
gaussdb=# CREATE ROLE paul PASSWORD '********';

--设置当前用户为paul。
gaussdb=# SET ROLE paul PASSWORD '********';

--查看当前会话用户，当前用户。
gaussdb=> SELECT SESSION_USER, CURRENT_USER;
 session_user | current_user
--------------+--------------
 omm          | paul
(1 row)

--重置当前用户。
gaussdb=> RESET ROLE;

--删除用户。
gaussdb=# DROP USER paul;
```

相关参考

SET SESSION AUTHORIZATION

1.13.19.8 SET SESSION AUTHORIZATION

功能描述

把当前会话里的会话用户标识和当前用户标识都设置为指定的用户。

注意事项

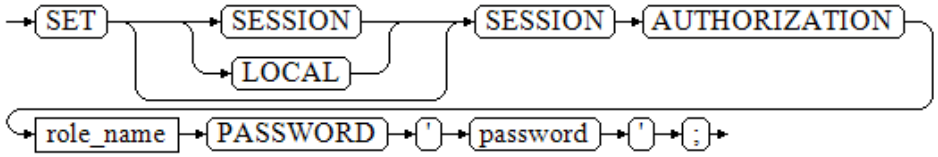
- 只有在初始会话用户有系统管理员权限的时候，会话用户标识符才能改变。否则，只有在指定了被认证的用户名的情况下，系统才接受该命令。
- 当用户被配置为通过LDAP方式进行登录认证时，无法通过本操作设置用户标识符为该用户。

版权所有 © 华为云计算技术有限公司

语法格式

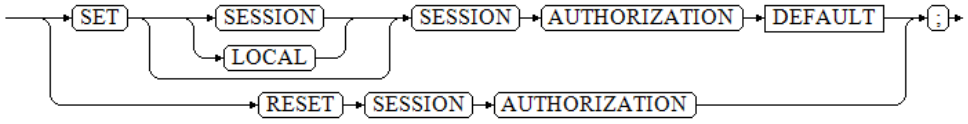
- 为当前会话设置会话用户标识符和当前用户标识符。

```
SET [ SESSION | LOCAL ] SESSION AUTHORIZATION role_name PASSWORD 'password';
```



- 重置会话和当前用户标识符为初始认证的用户名。

```
{SET [ SESSION | LOCAL ] SESSION AUTHORIZATION DEFAULT
   | RESET SESSION AUTHORIZATION};
```



参数说明

- **SESSION**

  声明这个命令只对当前会话起作用。

- **LOCAL**

  声明该命令只在当前事务中有效。

- **role_name**

  用户名。

  取值范围：字符串，数据库中已经存在的用户名。

- **password**

  角色的密码。要求符合密码的命名规则。

  说明

  使用密文密码限制如下：

  - 管理员用户不能使用密文密码切换到其他管理员用户，只能向权限更低用户切换；

  使用密文密码通常用于gs_dump、gs_dumpall导出场景，其他场景不建议直接使用密文密码。

- **DEFAULT**

  重置会话和当前用户标识符为初始认证的用户名。

示例

```
--查询会话用户，当前用户。
gaussdb=# SELECT SESSION_USER, CURRENT_USER;
 session_user | current_user
--------------+--------------
 omm          | omm
(1 row)

--创建角色paul。
gaussdb=# CREATE ROLE paul PASSWORD '********';

--设置当前用户和会话用户为paul，并查询会话用户，当前用户。
gaussdb=# SET SESSION AUTHORIZATION paul PASSWORD '********';
gaussdb=> SELECT SESSION_USER, CURRENT_USER;
 session_user | current_user
```


版权所有 © 华为云计算技术有限公司

```
--------------+--------------
 paul         | paul
(1 row)

--重置会话用户，当前用户。
gaussdb=> RESET SESSION AUTHORIZATION;
gaussdb=# SELECT SESSION_USER, CURRENT_USER;
 session_user | current_user
--------------+--------------
 omm          | omm
(1 row)

--删除用户。
gaussdb=# DROP USER paul;
```

相关参考

SET ROLE

1.13.19.9 SET TRANSACTION

功能描述

为事务设置特性。事务特性包括事务隔离级别、事务访问模式(读/写或者只读)。可以设置当前事务的特性（LOCAL)，也可以设置会话的默认事务特性（SESSION）。

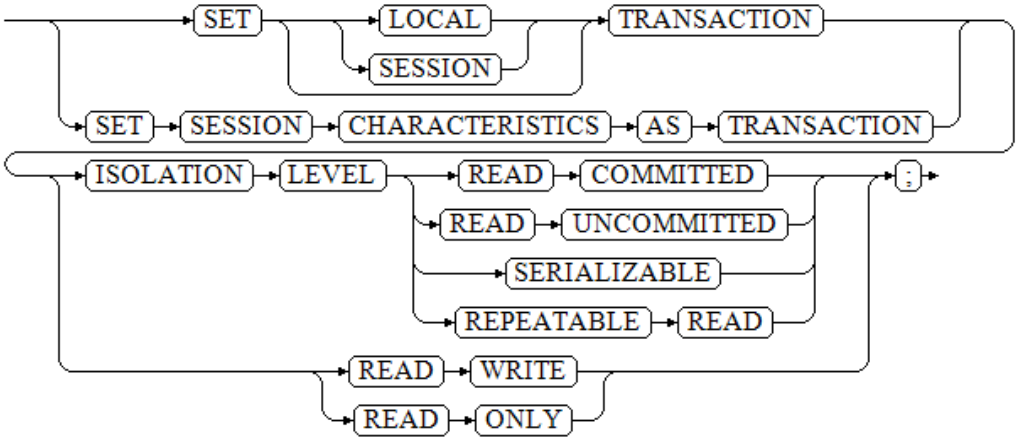
注意事项

设置当前事务特性需要在事务中执行（即执行SET TRANSACTION之前需要执行START TRANSACTION或者BEGIN），否则设置不生效。

语法格式

设置事务的隔离级别、读写模式。

```
{ SET [ LOCAL | SESSION ] TRANSACTION | SET SESSION CHARACTERISTICS AS TRANSACTION }
  { ISOLATION LEVEL { READ COMMITTED | READ UNCOMMITTED | SERIALIZABLE | REPEATABLE READ }
  | { READ WRITE | READ ONLY }
  };
```



参数说明

- **LOCAL**
  声明该命令只在当前事务中有效。

版权所有 © 华为云计算技术有限公司

- **SESSION**

  声明这个命令只对当前会话起作用。

- **SESSION CHARACTERISTICS**

  声明这个命令只对当前会话起作用。

- **ISOLATION LEVEL**

  指定事务隔离级别，该参数决定当一个事务中存在其他并发运行事务时能够看到什么数据。

  说明

  - 在事务中第一个数据修改语句（INSERT、DELETE、UPDATE、FETCH或COPY）执行之后，当前事务的隔离级别就不能再次设置。

  取值范围：

  - READ COMMITTED：读已提交隔离级别，只能读到已经提交的数据，而不会读到未提交的数据。这是缺省值。
  - READ UNCOMMITTED：读未提交隔离级别，指定后的行为和READ COMMITTED行为一致。
  - REPEATABLE READ：可重复读隔离级别，仅仅能看到事务开始之前提交的数据，不能看到未提交的数据，以及在事务执行期间由其它并发事务提交的修改。
  - SERIALIZABLE：GaussDB目前功能上不支持此隔离级别，等价于REPEATABLE READ。

- **READ WRITE | READ ONLY**

  指定事务访问模式（读/写或者只读）。

示例

```
--开启一个事务。
gaussdb=# START TRANSACTION;

--设置事务的隔离级别为READ COMMITTED，访问模式为READ ONLY。
gaussdb=# SET LOCAL TRANSACTION ISOLATION LEVEL READ COMMITTED READ ONLY;

--查看当前事务的隔离级别。
gaussdb=# SHOW transaction_isolation;
 transaction_isolation
-----------------------
 read committed
(1 row)

--提交事务。

gaussdb=# COMMIT;
```

1.13.19.10 SHOW

功能描述

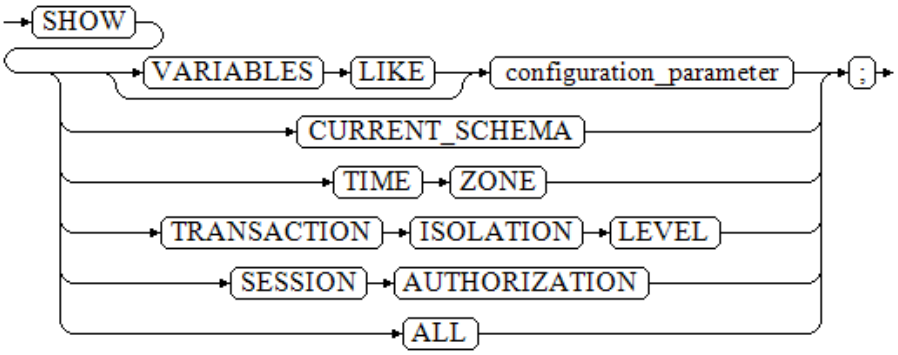
SHOW将显示当前运行时参数的数值。

语法格式

```
SHOW
 {
   [VARIABLES LIKE] configuration_parameter |
```


版权所有 © 华为云计算技术有限公司

```
  CURRENT_SCHEMA |
  TIME ZONE |
  TRANSACTION ISOLATION LEVEL |
  SESSION AUTHORIZATION |
  ALL
};
```



参数说明

- **configuration_parameter**
  运行时参数的名称。
  取值范围：可以使用SHOW ALL命令查看运行时参数。

  说明

  部分通过SHOW ALL查看的参数不能通过SET设置，如max_datanodes。
- **CURRENT_SCHEMA**
  当前模式。
- **TIME ZONE**
  时区。
- **TRANSACTION ISOLATION LEVEL**
  事务的隔离级别。
- **SESSION AUTHORIZATION**
  当前会话的用户标识符。
- **ALL**
  所有运行时参数。

示例

```
--显示timezone参数值。
gaussdb=# SHOW timezone;
TimeZone
----------
 PRC
(1 row)

--显示当前模式。
gaussdb=# SHOW CURRENT_SCHEMA;
 current_schema
----------------
 "$user",public
(1 row)

--显示事务的隔离级别。
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SHOW TRANSACTION ISOLATION LEVEL;
 transaction_isolation
-----------------------
 read committed
(1 row)

--显示当前会话的用户标识符。
gaussdb=# SHOW SESSION AUTHORIZATION;
 session_authorization
-----------------------
 omm1
(1 row)

--显示所有参数。
gaussdb=# SHOW ALL;

--显示参数名中包含”var”的所有参数。
gaussdb=# SHOW VARIABLES LIKE var;
```

相关链接

SET，**RESET**

1.13.19.11 SHUTDOWN

功能描述

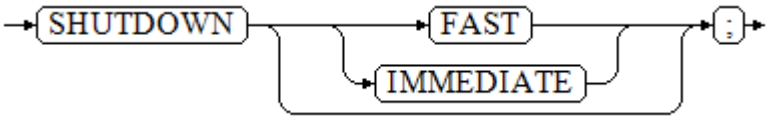
SHUTDOWN将关闭当前连接的数据库节点。

注意事项

- 仅拥有管理员权限的用户可以运行此命令。
- 分布式数据库不支持SHUTDOWN命令。

语法格式

```
SHUTDOWN [FAST | IMMEDIATE];
```



参数说明

FAST | IMMEDIATE

FAST表示不等待客户端中断连接，将所有活跃事务回滚并且强制断开客户端，然后关闭数据库节点。

IMMEDIATE表示强行关闭，在下次重新启动的时候将导致故障恢复。

若不指定关闭模式，默认为FAST。

示例

```
--关闭当前数据库节点。
gaussdb=# SHUTDOWN;
```


版权所有 © 华为云计算技术有限公司

```
--使用fast模式关闭当前数据库节点。
gaussdb=# SHUTDOWN FAST;
```

1.13.19.12 START TRANSACTION

功能描述

通过START TRANSACTION启动事务。如果声明了隔离级别、读写模式，那么新事务就使用这些特性，类似执行了**SET TRANSACTION**。

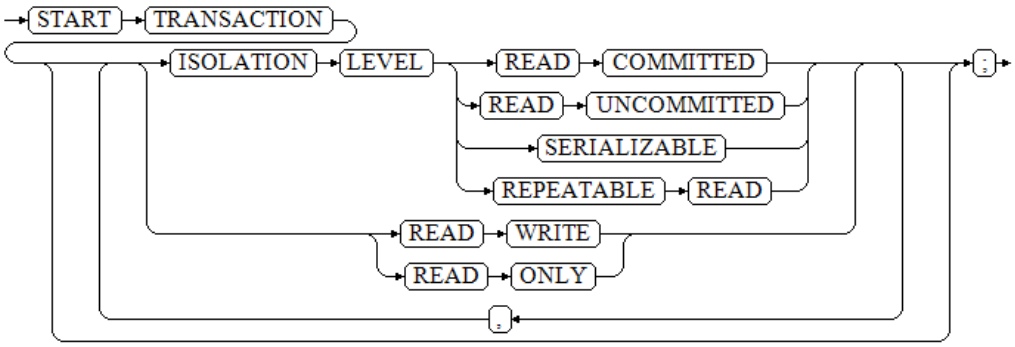
注意事项

使用START TRANSACTION时，自动提交保持禁用状态，直到COMMIT或结束事务ROLLBACK，自动提交模式才将恢复为初始状态。

语法格式

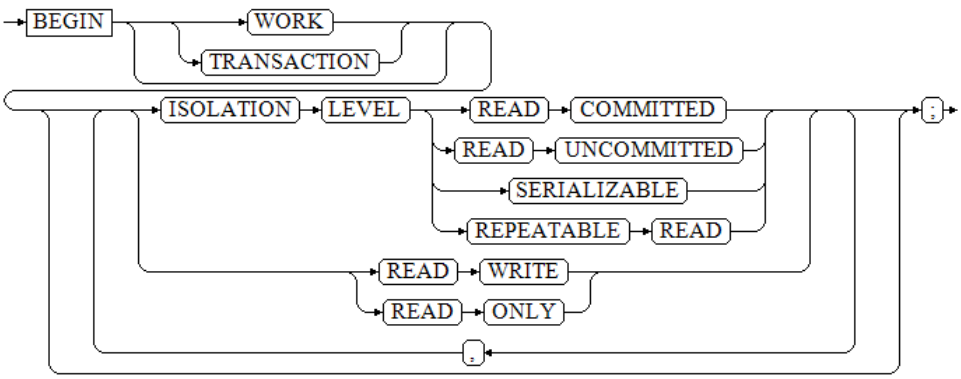
格式一：START TRANSACTION格式

```
START TRANSACTION
 [
  {
    ISOLATION LEVEL { READ COMMITTED | READ UNCOMMITTED | SERIALIZABLE | REPEATABLE READ }
    | { READ WRITE | READ ONLY }
  } [, ...]
 ];
```



格式二：BEGIN格式

```
BEGIN [ WORK | TRANSACTION ]
 [
  {
    ISOLATION LEVEL { READ COMMITTED | READ UNCOMMITTED | SERIALIZABLE | REPEATABLE READ }
    | { READ WRITE | READ ONLY }
  } [, ...]
 ];
```



版权所有 © 华为云计算技术有限公司

参数说明

- **WORK | TRANSACTION**

  BEGIN格式中的可选关键字，没有实际作用。

- **ISOLATION LEVEL**

  指定事务隔离级别，它决定当一个事务中存在其他并发运行事务时它能够看到什么数据。

  说明

  在事务中第一个数据修改语句（INSERT、DELETE、UPDATE、FETCH、COPY）执行之后，事务隔离级别就不能再次设置。

  取值范围：

  - READ COMMITTED：读已提交隔离级别，只能读到已经提交的数据，而不会读到未提交的数据。这是缺省值。
  - READ UNCOMMITTED：读未提交隔离级别，指定后的行为和READ COMMITTED行为一致。
  - REPEATABLE READ：可重复读隔离级别，仅仅看到事务开始之前提交的数据，它不能看到未提交的数据，以及在事务执行期间由其它并发事务提交的修改。
  - SERIALIZABLE：GaussDB目前功能上不支持此隔离级别，设置该隔离级别时，等价于REPEATABLE READ。

- **READ WRITE | READ ONLY**

  指定事务访问模式（读/写或者只读）。

示例

```
--创建SCHEMA。
gaussdb=# CREATE SCHEMA tpcds;

--创建表tpcds.reason。
gaussdb=# CREATE TABLE tpcds.reason (c1 int, c2 int);

--以默认方式启动事务。
gaussdb=# START TRANSACTION;
gaussdb=# SELECT * FROM tpcds.reason;
gaussdb=# END;

--以默认方式启动事务。
gaussdb=# BEGIN;
gaussdb=# SELECT * FROM tpcds.reason;
gaussdb=# END;

--以隔离级别为READ COMMITTED，读/写方式启动事务。
gaussdb=# START TRANSACTION ISOLATION LEVEL READ COMMITTED READ WRITE;
gaussdb=# SELECT * FROM tpcds.reason;
gaussdb=# COMMIT;

--删除表。
gaussdb=# DROP TABLE tpcds.reason;

--删除SCHEMA。
gaussdb=# DROP SCHEMA tpcds CASCADE;
```

相关链接

COMMIT | END，**ROLLBACK**，**SET TRANSACTION**

版权所有 © 华为云计算技术有限公司

1.13.20 T

1.13.20.1 TIMECAPSULE TABLE

功能描述

在人为操作或应用程序错误时，使用TIMECAPSULE TABLE语句恢复可将表恢复到一个早期状态。

表可以闪回到过去的时间点，这依赖于系统中保存的旧版本数据。此外GaussDB数据库不能恢复到通过DDL操作改变了表结构的早期状态。

注意事项

- TIMECAPSULE TABLE语句的用法主要分为两大类：闪回旧版本数据和从回收站中闪回。
  - TO TIMESTAMP和TO CSN能够将表闪回到过去的某个版本，当前仅支持Ustore存储引擎。
  - 回收站记录了DROP和TRUNCATE的对象数据。TO BEFORE DROP和TO BEFORE TRUNCATE就是从回收站中闪回，当前支持Ustore以及Astore存储引擎。
- 不支持闪回表的对象类型：系统表、DFS表、全局临时表、本地临时表、UNLOGGED表、序列表、密态表。
- 不支持含有自定义类型表的闪回。
- 开启闪回后，回收站里的表可以进行表级备份，无法进行表级恢复。
- 闪回点和当前点之间，执行过修改表结构或影响物理存储的语句（DDL、DCL、VACUUM FULL），则闪回失败。
- 执行闪回删除需要用户具有如下权限：用户必须具有垃圾对象所在Schema的CREATE和USAGE权限，并且用户必须是Schema的所有者或者是垃圾对象的所有者。

  执行闪回TRUNCATE需要用户具有如下权限：用户必须具有垃圾对象所在SCHEMA的CREATE和USAGE权限，并且用户必须是Schema的所有者或者是垃圾对象的所有者，另外用户必须具有垃圾对象的TRUNCATE权限。
- 不适用闪回DROP/TRUNCATE功能的场景或表：
  - 回收站关闭场景：enable_recyclebin = off。
  - 系统处于维护态（xc_maintenance_mode = on）。
  - 从不支持的基线版本升级到支持的版本且处于升级观察期时。
  - 多对象删除场景：DROP/TRUNCATE TABLE命令同时指定多个对象。
  - 系统表、DFS表、全局临时表、本地临时表、UNLOGGED表、序列表、密态表。
  - 回收站对象被清理后无法闪回DROP/TRUNCATE，recyclebin_retention_time参数用于设置回收站对象保留时间。
  - 不支持扩缩容场景：扩容重分布时会强制将回收站中的数据清空，扩容期间，DROP的对象不会放入回收站。
  - 回收站对象禁止DML、DCL、DDL等写操作，不支持DQL查询操作。

 版权所有 © 华为云计算技术有限公司

  - TRUNCATE表和闪回TRUNCATE操作之间，执行过修改表结构或影响物理文件的语句（DDL、DCL、VACUUM FULL、增加/删除/切割/合成等分区操作），闪回失败。
  - 整表删除或截断时，分区会随着整表放入回收站，单个删除的分区不支持放入回收站，避免破坏数据一致性。
  - 如果有在线索引残留（存在未创建的节点，在该节点上搜索不到该索引），DROP表时会报错，需要清理残留数据才能DROP成功，将对象放入回收站。
  - 如果表依赖的对象为外部对象（如表列为复合类型、自定义类型等），则采用物理删除，不将表放入回收站。

- DROP闪回约束

  可以指定原始用户指定的表的名称，或对象删除时数据库分配的系统生成名称。

  - 回收站中系统生成的对象名称是唯一的。因此，如果指定系统生成名称，那么数据库检索指定的对象。使用“SELECT * FROM gs_recyclebin;”语句查看回收站中的内容。
  - 如果指定了用户指定的名称，且回收站中包含多个该名称的对象，那么数据库检索回收站中最近移动的对象。如果需要检索更早版本的表，请按以下步骤执行：
    - 指定需要检索的表的系统生成名称。
    - 执行TIMECAPSULE TABLE ... TO BEFORE DROP语句，直到找到要检索的表。
  - 恢复DROP表时，只恢复基表名，其他子对象名均保持回收站对象名。用户可根据需要，执行DDL命令手工调整子对象名。
  - 如果表存在缺省值引用序列和自定义函数，那么闪回DROP表成功但不会恢复缺省值。
  - 如果表存在视图引用，DROP表时需要级联删除视图，那么闪回DROP表成功但不会恢复视图。
  - 回收站对象不支持DML、DCL、DDL等写操作，不支持DQL查询操作（后续支持）。
  - recyclebin_retention_time配置参数用于设置回收站对象保留时间，超过该时间的回收站对象将被自动清理。

- TRUNCATE闪回约束
  - TRUNCATE闪回后，统计信息无变化，仍显示为0，可以在业务低峰期（降低性能影响）时候手动ANALYZE来修正统计信息。
  - RENAME TO仅支持DROP闪回操作为检索表指定新名称，不支持TRUNCATE闪回。
  - TRUNCATE闪回不能跨越影响表结构或物理存储的语句，否则会报错。即闪回点和当前点之间，如果执行过修改表结构或影响物理存储的语句（DDL、DCL、VACUUM FULL、增加/删除/切割/合成等分区操作），则闪回失败。执行过DDL的表进行闪回操作报错：“ERROR：The table definition of %s has been modified.”。涉及namespace、表名改变等操作的DDL执行闪回操作报错：“ERROR: recycle object %s desired does not exist.”。

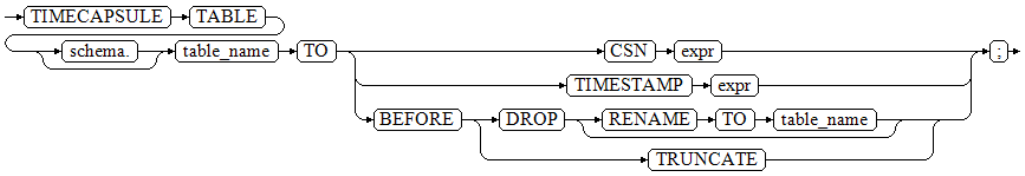
- vacuum更新库级frozenXid时会参考回收站中rcyfrozenxid64的值，若回收站中存在长时间未清理的表，可能会导致frozenXid无法推进，影响clog的回收（clog文件占存储空间较小，实际场景影响较小），可通过清理回收站中不再需要闪回的表来推进frozenXid。

版权所有 © 华为云计算技术有限公司

- 从503.0之前的版本升级到最新版本时，若升级前已开启回收站，需确保回收站中不存在分区表对象后再进行升级操作，否则升级将失败。
- 从503.0之前的版本升级到最新版本时，对于分区表，无论是在升级未提交状态还是升级已提交状态，执行闪回表或闪回查询操作均会报错，提示表结构发生变更。

语法格式

```
TIMECAPSULE TABLE
   [schema.]table_name TO { CSN expr | TIMESTAMP expr | BEFORE { DROP [RENAME TO table_name] |
TRUNCATE } };
```



参数说明

- **schema**

  指定模式包含的表。如果缺省，则为当前模式。
- **table_name**

  指定表名。
- **TO CSN**

  指定要返回表的时间点对应的事务提交序列号（CSN）。expr必须计算一个数字，代表有效的CSN。

  说明：GTM-Free场景各节点使用本地CSN，没有全局统一CSN号，暂不支持使用TO CSN方式进行闪回操作。
- **TO TIMESTAMP**

  指定要返回表的时间点对应的时间戳。expr必须计算一个过去有效的时间戳（使用TO_TIMESTAMP函数将字符串转换为时间类型）。表将被闪回到指定时间戳大约3秒内的时间点。

  说明：闪回点过旧时，因旧版本被回收导致无法获取旧版本，会导致闪回失败并报错：Restore point too old。
- **TO BEFORE DROP**

  使用这个子句检索回收站中已删除的表及其子对象。
- **RENAME TO**

  为从回收站中检索的表指定一个新名称。
- **TO BEFORE TRUNCATE**

  闪回到TRUNCATE之前。

示例

- **闪回到某个时间点**

  需要设置undo_retention_time参数，用于设置旧版本undo的保留时间。

  参数使用请联系管理员处理。

```
--建表并插入数据。
gaussdb=# CREATE TABLE tbl_test(c1 int, c2 int) with(storage_type = ustore) DISTRIBUTE BY
```


版权所有 © 华为云计算技术有限公司

```
REPLICATION;
gaussdb=# INSERT INTO tbl_test VALUES (1,1),(2,2),(3,3);

--查询当前时间和全局所有节点上的next_csn。
gaussdb=# SELECT now();
              now
-------------------------------
 2023-11-27 17:06:34.840698+08
(1 row)

gaussdb=# SELECT int8in(xidout(next_csn)) FROM gs_get_next_xid_csn();
 int8in
--------
  25391
  25391
  25391
  25391
  25391
  25391
(6 rows)

--修改数据。
gaussdb=# UPDATE tbl_test SET c1=111, c2=222 WHERE c1=1;

--查询数据，并将数据闪回至UPDATE之前。
gaussdb=# SELECT * FROM tbl_test;
 c1  | c2
-----+-----
 111 | 222
   2 |   2
   3 |   3
(3 rows)
gaussdb=# TIMECAPSULE table tbl_test TO TIMESTAMP to_timestamp('2023-11-27
17:06:34.840698','YYYY-MM-DD HH24:MI:SS.FF');

--也可使用如下SQL。
gaussdb=# TIMECAPSULE table tbl_test TO CSN 25391;

gaussdb=# SELECT * FROM tbl_test;
 c1 | c2
----+----
  2 |  2
  3 |  3
  1 |  1
(3 rows)

--删除。
gaussdb=# DROP TABLE tbl_test PURGE;
```

- **从回收站中闪回数据**

  前提条件：

  - 开启enable_recyclebin参数，启用回收站，参数使用请联系管理员处理。
  - recyclebin_retention_time参数用于设置回收站对象保留时间，超过该时间的回收站对象将被自动清理，参数使用请联系管理员处理。

```
--建表并插入数据。
gaussdb=# CREATE TABLE tbl_test1(c1 int, c2 varchar(10))with(storage_type = ustore);
gaussdb=# INSERT INTO tbl_test1 VALUES (1,'AAA'),(2,'BBB');

--删除表。
gaussdb=# DROP TABLE tbl_test1;

--闪回至表删除之前。
gaussdb=# TIMECAPSULE TABLE tbl_test1 TO BEFORE DROP;

--查询数据。
gaussdb=# SELECT * FROM tbl_test1;
 c1 | c2
```


版权所有 © 华为云计算技术有限公司

```
----+-----
 1 | AAA
 2 | BBB
(2 rows)

--删除表（添加PURGE参数,级联删除回收站该表数据）。
gaussdb=# DROP TABLE tbl_test1 PURGE;
```

1.13.20.2 TRUNCATE

功能描述

清理表数据，TRUNCATE快速地从表中删除所有行。

它和在目标表上进行无条件的DELETE有同样的效果，但由于TRUNCATE不做表扫描，因而快得多。在大表上操作效果更明显。

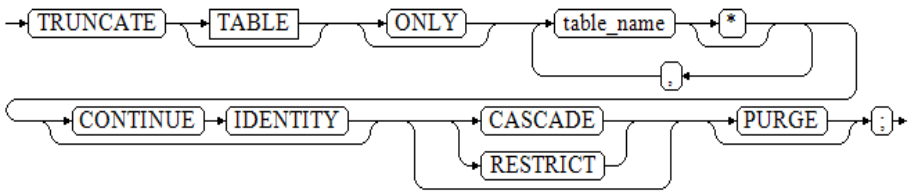
注意事项

- 表的所有者、被授予了表的TRUNCATE权限的用户或被授予TRUNCATE ANY TABLE权限的用户，有权TRUNCATE指定表，三权分立关闭时的系统管理员默认拥有该权限。
- TRUNCATE TABLE在功能上与不带WHERE子句DELETE语句相同：二者均删除表中的全部行。
- TRUNCATE TABLE比DELETE速度快且使用系统和事务日志资源少：
  - DELETE语句每次删除一行，并在事务日志中为所删除每行记录一项。
  - TRUNCATE TABLE通过释放存储表数据所用数据页来删除数据，并且只在事务日志中记录页的释放。
- TRUNCATE、DELETE和DROP三者的差异如下：
  - TRUNCATE TABLE：删除内容，释放空间，但不删除定义。
  - DELETE TABLE：删除内容，不删除定义，不释放空间。
  - DROP TABLE：删除内容和定义，释放空间。

语法格式

- 清理表数据。

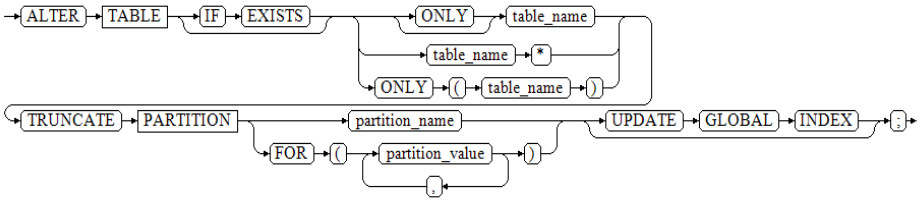
```
TRUNCATE [ TABLE ] [ ONLY ] {table_name [ * ]} [, ... ]
    [ CONTINUE IDENTITY ] [ CASCADE | RESTRICT ] [ PURGE ];
```



- 清理表分区的数据。

```
ALTER TABLE [ IF EXISTS  ] { [ ONLY  ] table_name
                           | table_name *
                           | ONLY ( table_name )  }
    TRUNCATE PARTITION { partition_name
                       | FOR (  partition_value  [, ...] )  } [ UPDATE GLOBAL INDEX ];
```


版权所有 © 华为云计算技术有限公司



参数说明

- **ONLY**

  如果声明ONLY，只有指定的表会被清空。如果没有声明ONLY，这个表以及其所有子表（若有）会被清空。

- **table_name**

  目标表的名称（可以有模式修饰）。

  取值范围：已存在的表名。

- **CONTINUE IDENTITY**

  不改变序列的值。这是缺省值。

- **CASCADE | RESTRICT**
  - CASCADE：级联清空所有由于CASCADE而被添加到组中的表。
  - RESTRICT（缺省值）：如果其他表在该表上有外键（分布式场景暂不支持）引用则拒绝清空。

- **PURGE**

  默认将表数据放入回收站中，PURGE直接清理。

- **partition_name**

  目标分区表的分区名。

  取值范围：已存在的分区名。

- **partition_value**

  指定的分区键值。

  通过PARTITION FOR子句指定的这一组值，可以唯一确定一个分区。

  取值范围：需要进行删除数据分区的分区键的取值范围。

  须知

  使用PARTITION FOR子句时，partition_value所在的整个分区会被清空。

- **UPDATE GLOBAL INDEX**

  如果使用该参数，则会更新分区表上的所有全局索引，以确保使用全局索引可以查询出正确的数据。

  如果不使用该参数，并且GUC参数enable_gpi_auto_update和enable_drop_truncate_partition_keep_gpi取值都为off时，则分区表上的所有全局索引将会失效。

示例

- 清理表数据

版权所有 © 华为云计算技术有限公司

```
--创建表reason。
gaussdb=# CREATE TABLE reason (r_reason_sk int,r_reason_id varchar(16),r_reason_desc
varchar(100));

--向表中插入多条记录。
gaussdb=# INSERT INTO reason values(1,'AAAAAAAABAAAAAAA','reason 1'),
(5,'AAAAAAAABAAAAAAA','reason 2'),
                                (15,'AAAAAAAABAAAAAAA','reason 3'), (25,'AAAAAAAABAAAAAAA','reason
4'),
                                (35,'AAAAAAAABAAAAAAA','reason 5'), (45,'AAAAAAAACAAAAAAA','reason
6');

--查看表的信息，大小约为16kB
gaussdb=# \d+
                                 List of relations
 Schema |  Name  | Type  | Owner | Size  |             Storage              | Description
--------+--------+-------+-------+-------+----------------------------------+-------------
 public | reason | table | omm   | 16 kB | {orientation=row,compression=no} |
(1 row)

--使用DELETE语句不带WHERE条件，清空表的数据，并查看表的大小。
gaussdb=# DELETE FROM reason;
gaussdb=# \d+
                                 List of relations
 Schema |  Name  | Type  | Owner | Size  |             Storage              | Description
--------+--------+-------+-------+-------+----------------------------------+-------------
 public | reason | table | omm   | 16 kB | {orientation=row,compression=no} |
(1 row)

--使用TRUNCATE清空表reason，并查看表的大小
gaussdb=# TRUNCATE TABLE reason;
gaussdb=# \d+
                                 List of relations
 Schema |  Name  | Type  | Owner |  Size   |             Storage              | Description
--------+--------+-------+-------+---------+----------------------------------+-------------
 public | reason | table | omm   | 0 bytes | {orientation=row,compression=no} |
(1 row)

--删除表。
gaussdb=# DROP TABLE reason;
```

- 清理分区表数据。

```
--创建分区表。
gaussdb=# CREATE TABLE reason_p(
    r_reason_sk integer,
    r_reason_id character(16),
    r_reason_desc character(100)
)PARTITION BY RANGE (r_reason_sk)(
    partition p_05_before values less than (05),
    partition p_15 values less than (15),
    partition p_25 values less than (25),
    partition p_35 values less than (35),
    partition p_45_after values less than (MAXVALUE)
);

--插入数据。
gaussdb=# INSERT INTO reason_p values(1,'AAAAAAAABAAAAAAA','reason 1'),
(5,'AAAAAAAABAAAAAAA','reason 2'),
                                 (15,'AAAAAAAABAAAAAAA','reason 3'), (25,'AAAAAAAABAAAAAAA','reason
4'),
                                 (35,'AAAAAAAABAAAAAAA','reason 5'), (45,'AAAAAAAACAAAAAAA','reason
6');

--清空分区p_05_before。
gaussdb=# ALTER TABLE reason_p TRUNCATE PARTITION p_05_before UPDATE GLOBAL INDEX;

--清空分区表。
gaussdb=# TRUNCATE TABLE reason_p;
```


版权所有 © 华为云计算技术有限公司

```
--删除表reason_p。
gaussdb=# DROP TABLE reason_p;
```

1.13.21 U

1.13.21.1 UPDATE

功能描述

更新表中的数据，UPDATE对满足条件的所有行中指定的字段值进行修改；WHERE子句表示声明条件；SET子句中指定的字段将会被修改，没有出现的字段则保持原有的字段值。

注意事项

- 表的所有者、拥有表UPDATE权限的用户或拥有UPDATE ANY TABLE权限的用户，皆有权限更新表中的数据，当三权分立开关关闭时，系统管理员默认拥有此权限。
- UPDATE..... LIMIT row_count 仅支持执行计划下推的部分场景（不支持复制表），前置条件要求过滤条件包含等值分布列且过滤条件相对简单，避免出现强制类型转换的情况。如果执行失败，请简化过滤条件。
- 对expression或condition条件里涉及到的任何表要有SELECT权限。
- 仅当表属性enable_update_distkey设置为on，且在约束条件下时，支持更新（UPDATE）分布列的操作。否则，分布列不支持更新（UPDATE）操作。

  说明

  支持更新（UPDATE）分布列操作的约束如下：
  - 仅当相应表属性enable_update_distkey设置为on才支持更新分布列；
  - 不支持将UPDATE语句下推DN执行，直接生成PGXC计划，不会根据更新分布列前后值而对计划做改变；
  - 不支持带有行级UPDATE TRIGGER的表，否则会执行失败，报错进行提示。对行级INSERT/DELETE TRIGGER不生效，update statement级TRIGGER正常执行；
  - 不支持并发更新同一行，先获取锁的执行，DN上后获取锁的按照GUC参数concurrent_dml_mode设置情况进行不同的行为（返回0或报错）。如果报错，则可能存在两种情况：（1）报错提示信息为update distribute column conflict；（2）当获取锁时间超过阈值时，报错提示信息为锁超时。
  - 不支持带有全局二级索引（GSI）的表，否则会执行报错。
  - 只支持HASH分布，不支持LIST/RANGE分布表，否则会执行报错。
  - 不支持MERGE INTO和UPSERT更新分布列的行为，否则会执行报错。
  - 不支持gtm_free，否则会执行报错。
  - 不支持UPDATE RETURNING，否则会执行报错。
  - 不支持带有关联表的语句，否则会执行报错。
  - 不支持UPDATE + LIMIT，否则会执行报错。
  - 由于ORA兼容和MYSQL兼容模式下不支持DELETE指定多分区，UPDATE分布列反解析DELETE语句时不会带有多分区，数据一致性由条件tableoid保证。
- 对于行存复制表，仅支持两种场景下的UPDATE操作：
  - 有主键约束的场景。
  - 执行计划能下推的场景。

 版权所有 © 华为云计算技术有限公司

- 对于子查询是STREAM计划的UPDATE语句，不支持并发更新同一行。
- 不支持用户通过UPDATE系统表的方式对数据库字符编码进行修改，该操作会导致存在存量数据或其他部分操作异常的情况。如需更改数据库的字符集编码，应当遵循切换数据库流程，进行相关的数据迁移操作。
- 高并发场景下，对超宽表执行大规模UPDATE时，若存在秒级频繁DDL，可能导致UPDATE耗时升至小时级。分钟级DDL对性能无明显影响。

语法格式

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
UPDATE [/*+ plan_hint */] [ ONLY ] {table_name [ partition_clause ] | subquery | view_name} [ * ] [ [ AS ] alias ]
SET {column_name = { expression | DEFAULT }
    |( column_name [, ...] ) = {( { expression | DEFAULT } [, ...] ) |sub_query }}[, ...]
    [ FROM from_list] [ WHERE condition | WHERE CURRENT OF cursor_name ]
    [ ORDER BY {expression [ [ ASC | DESC | USING operator ]
[ LIMIT { count } ]
    [ RETURNING {*
                | {output_expression [ [ AS ] output_name ]} [, ...] }];

where sub_query can be:
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
{ * | {expression [ [ AS ] output_name ]} [, ...] }
[ FROM from_item [, ...] ]
[ WHERE condition  ]
[ GROUP BY grouping_element [, ...] ]
[ HAVING condition [, ...] ]
[ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS { FIRST | LAST } ]} [, ...] ]
[ LIMIT { [offset,] count | ALL } ]
```

- 其中子查询with_query为：
  ```
  with_query_name [ ( column_name [, ...] ) ] AS [ [ NOT ] MATERIALIZED ] ( {select | values | insert | update | delete} )
  ```
- subquery为：
  ```
  [ WITH [ RECURSIVE ] with_query [, ...] ]
  SELECT [/*+ plan_hint */] [ ALL ]
  { * | {expression [ [ AS ] output_name ]} [, ...] }
  [ into_option ]
  [ FROM from_item [, ...] ]
  [ WHERE condition ]
  [ [ START WITH condition ] CONNECT BY [NOCYCLE] condition [ ORDER SIBLINGS BY expression ] ]
  [ ORDER BY {expression [ [ ASC | DESC | USING operator ] | nlssort_expression_clause ] [ NULLS { FIRST | LAST } ]} [, ...] ]
  [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]
  [ into_option ];
  ```
- 其中指定子查询源from_item为：
  ```
  {[ ONLY ] {table_name | view_name} [ * ] [ partition_clause ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  |( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
  |with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  |from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]}
  ```
- partition_clause为：
  ```
  PARTITION { ( partition_name [, ...] ) | FOR ( partition_value [, ...] ) }
  ```

参数说明

- **WITH [RECURSIVE] with_query [, ...]**

  用于声明一个或多个可以在主查询中通过名称引用的子查询，相当于临时表。这种子查询语句结构称为CTE（Common Table Expression）结构，应用这种结构时，执行计划中将存在CTE SCAN的内容。

  如果声明了RECURSIVE，那么允许SELECT子查询通过名称引用它自己。

版权所有 © 华为云计算技术有限公司

其中with_query中：

- with_query_name指定子查询生成的结果集名称，在查询中可使用该名称访问子查询的结果集。
- column_name指定子查询结果集中显示的列名。
- 每个子查询可以是SELECT、VALUES、INSERT、UPDATE或DELETE语句。
- 用户可以使用MATERIALIZED或NOT MATERIALIZED对CTE进行修饰。
  - 如果声明为MATERIALIZED，WITH查询将被物化，生成一个子查询结果集的复制，在引用处直接查询该复制，因此WITH子查询无法和主干SELECT语句进行联合优化（如谓词下推、等价类传递等），对于此类场景可以使用NOT MATERIALIZED进行修饰，如果WITH查询语义上可以作为子查询内联执行，则可以进行上述优化。
  - 如果用户没有显示声明物化属性则遵守以下规则：如果CTE只在所属SELECT主干中被引用一次，且语义上支持内联执行，则会被改写为子查询内联执行，否则以CTE Scan的方式物化执行。

- **plan_hint**

  以/*+ */的形式在UPDATE关键字后，用于对UPDATE对应的语句块生成的计划进行hint调优，详细用法请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。每条语句中只有第一个/*+ plan_hint */注释块会作为hint生效，里面可以写多条hint。

- **table_name**

  要更新的表名，可以使用模式修饰。如果在表名前指定了ONLY，只会更新表中匹配的行。如果未指定，任何从该表继承到的表中的匹配行也会被更新。

  取值范围：已存在的表名称。

  说明

  支持使用DATABASE LINK方式对远程表进行操作，使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

- **subquery**

  要更新的子查询，在对子查询进行更新时，会将子查询当成一个临时视图，支持在子查询后面加CHECK OPTION选项。

  如果子查询中只有一张表，则对该表更新数据；如果子查询中有多张表或有嵌套关系，则通过判断是否有保留键表确定是否可更新。关于保留键表和WITH CHECK OPTION请参见**CREATE VIEW**。

- **view_name**

  要更新的目标视图。

版权所有 © 华为云计算技术有限公司

说明

对视图和子查询的更新，有如下约束：

- 只有直接引用基表用户列的列可进行UPDATE操作。
- 子查询或视图必须至少包含一个可更新列，关于可更新列请参见**CREATE VIEW**。
- 不支持在顶层包含DISTINCT、GROUP BY、HAVING、LIMIT、OFFSET子句的视图和子查询。
- 不支持在顶层包含集合运算（UNION、INTERSECT、EXCEPT、MINUS）的视图和子查询。
- 不支持目标列表中包含聚集函数、窗口函数、返回集合函数（array_agg、json_agg、generate_series等）的视图和子查询。
- 不支持仅带有BEFORE/AFTER触发器，没有INSTEAD OF触发器或INSTEAD规则的视图。
- 视图和子查询中支持的表类型包括普通表、临时表、全局临时表、分区表、二级分区表、ustore表、astore表。
- 多表连接视图或连接子查询中一次只能更新一张基表。
- 连接视图或子查询只能更新保留键表，如果指定了CHECK OPTION选项，则无法对连接列做更新操作。关于保留键表请参见**CREATE VIEW**。
- 不支持更新系统视图。

- **partition_clause**

  指定分区更新操作。

  说明

  PARTITION指定多个分区名时，可以存在相同的分区名，最终分区范围取其并集。

- **alias**

  目标表的别名。

  取值范围：字符串，符合**标识符命名规范**。

- **column_name**

  要修改的字段名。

  支持使用目标表的别名加字段名来引用这个字段。例如：UPDATE foo AS f SET f.col_name = 'namecol'。

  取值范围：已存在的字段名。

- **expression**

  赋给字段的值或表达式。

- **DEFAULT**

  用对应字段的缺省值填充该字段。

  如果没有缺省值，则为NULL。

- **sub_query**

  子查询。

  使用同一数据库里其他表的信息来更新一个表可以使用子查询的方法。其中SELECT子句具体介绍请参考**SELECT**。

  在UPDATE单列时，支持使用ORDER BY子句与LIMIT子句；而在UPDATE多列时，则不支持使用ORDER BY子句与LIMIT子句。

版权所有 © 华为云计算技术有限公司

说明

- 当使用子查询更新多个列时，如果关闭GUC参数enable_shared_set_update开关，则优化器会为每个被更新列分别生成一个subplan；开启GUC参数enable_shared_set_update开关时，多个列可以共享使用一个subplan以提升查询效率，减少查询优化和执行开销。该GUC参数设置请联系管理员。
- 使用子查询更新多个列时，不推荐在子查询的targetlist中使用'*'，请明确列名，否则可能导致更新列不明确。
- 当SET子句中，子查询的WHERE语句使用“+”操作符时，需在兼容性参数behavior_compat_options中增加配置项enable_ora_joinop_in_updatestmt才能生效。

- **from_list**

  一个表的表达式列表，允许在WHERE条件里使用其他表的字段。与在一个SELECT语句的FROM子句里声明表列表类似。

  须知

  目标表不能出现在from_list里，除非在使用一个自连接（此时它必须以from_list的别名出现）。

- **condition**

  一个返回Boolean类型结果的表达式。只有这个表达式返回true的行才会被更新。不建议使用int等数值类型作为condition，因为int等数值类型可以隐式转换为Boolean值（非0值隐式转换为true，0转换为false），可能导致非预期的结果。

- **WHERE CURRENT OF cursor_name**

  当cursor指向表的某一行时，可以使用此语法更新cursor当前指向的行。

  cursor_name：指定游标的名称。

  须知

  - MYSQL兼容模式的数据库不支持使用此语法。
  - 此语法仅支持普通表，不支持分区表，不支持Hashbucket表。
  - 仅支持在存储过程中使用。
  - 不支持与其他WHERE条件组合使用。
  - 不支持与WITH、USING、ORDER BY、FROM组合使用。
  - CURSOR对应的SELECT语句必须声明为FOR UPDATE。
  - CURSOR对应的SELECT语句仅支持单表，不支持LIMIT/OFFSET，不支持带有子查询、子链接。
  - 存储过程中声明为FOR UPDATE的CURSOR，在COMMIT/ROLLBACK后，将无法再次使用。
  - 若CURSOR指向的行已经不存在，在ORA兼容性模式下将报错指定的行不存在（仅UPDATE时报错，DELETE不报错），其他兼容模式下不报错。
  - 复制表使用此语法时，不支持下推，必须有主键。

- **ORDER BY**

  关键字详见**SELECT**章节介绍。

版权所有 © 华为云计算技术有限公司

- **LIMIT**
  关键字详见**SELECT**章节介绍。
- **RETURNING output_expression**
  在所有需要更新的行都被更新之后，UPDATE命令用于计算返回值的表达式。
  取值范围：使用任何TABLE以及FROM中列出的表的字段。*表示返回所有字段。
- **output_name**
  字段的返回名称。

示例

- 修改表中所有数据。

```
--创建tbl_test1表并插入数据。
gaussdb=# CREATE TABLE tbl_test1(id int, info varchar(10));
gaussdb=# INSERT INTO tbl_test1 VALUES (1, 'A'), (2, 'B');

--查询。
gaussdb=# SELECT * FROM tbl_test1;
 id | info
----+------
  1 | A
  2 | B
(2 rows)

--修改tbl_test1表中所有数据的info列。
gaussdb=# UPDATE tbl_test1 SET info = 'aa';

--查询tbl_test1表。
gaussdb=# SELECT * FROM tbl_test1;
 id | info
----+------
  1 | aa
  2 | aa
(2 rows)
```

- 修改表中部分数据。

```
--修改tbl_test1表中id=2的数据。
gaussdb=# UPDATE tbl_test1 SET info = 'bb' WHERE id = 2;

--查询tbl_test1表。
gaussdb=# SELECT * FROM tbl_test1;
 id | info
----+------
  1 | aa
  2 | bb
(2 rows)
```

- 修改数据，并返回修改后的数据。

```
--修改tbl_test1表中id=1的数据，并指定返回info列。
gaussdb=# UPDATE tbl_test1 SET info = 'ABC' WHERE id = 1 RETURNING info;
 info
------
 ABC
(1 row)

UPDATE 1

--删除tbl_test1表。
gaussdb=# DROP TABLE tbl_test1;
```

- 使用子查询，在修改数据的同时，根据已有数据，插入新的数据。

```
--建表。
gaussdb=# CREATE TABLE test_grade (
    sid int,                --学号
    name varchar(50),       --姓名
    score char,             --成绩
```


版权所有 © 华为云计算技术有限公司

```
    examtime date,          --考试时间
    last_exam boolean       --是否是最后一次考试
);

--插入数据。
gaussdb=# INSERT INTO test_grade VALUES (1,'Scott','A','2008-07-08',1),(2,'Ben','D','2008-07-08',1),
(3,'Jack','D','2008-07-08',1);

--查询。
gaussdb=# SELECT * FROM test_grade;
 sid | name  | score |  examtime  | last_exam
-----+-------+-------+------------+-----------
   3 | Jack  | D     | 2008-07-08 | t
   1 | Scott | A     | 2008-07-08 | t
   2 | Ben   | D     | 2008-07-08 | t
(3 rows)

--2008-08-25 Ben参加了补考,成绩为B，正常步骤需要先修改last_exam为否,然后插入2008-08-25这一天的成绩。
gaussdb=# WITH old_exa AS ( UPDATE test_grade SET last_exam = 0 WHERE sid = 2 AND examtime = '2008-07-08' RETURNING sid, name )
        INSERT INTO test_grade VALUES ( ( SELECT sid FROM old_exa ), (SELECT name FROM old_exa), 'B', '2008-08-25', 1 );

--查询。
gaussdb=# SELECT * FROM test_grade;
 sid | name  | score |  examtime  | last_exam
-----+-------+-------+------------+-----------
   3 | Jack  | D     | 2008-07-08 | t
   1 | Scott | A     | 2008-07-08 | t
   2 | Ben   | D     | 2008-07-08 | f
   2 | Ben   | B     | 2008-08-25 | t
(4 rows)

--删除。
gaussdb=# DROP TABLE test_grade;
```

- **更新视图或子查询**

  示例1：更新子查询

```
--创建SCHEMA
gaussdb=# CREATE SCHEMA upd_subqry;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = 'upd_subqry';
SET

--创建表并插入数据
gaussdb=# CREATE TABLE t1 (x1 int, y1 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE t2 (x2 int PRIMARY KEY, y2 int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t2_pkey" for table "t2"
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4
gaussdb=# INSERT INTO t2 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4

--通过子查询更新t1
gaussdb=# UPDATE (SELECT * FROM t1) SET y1 = 13 where y1 = 3;
UPDATE 1
gaussdb=# UPDATE (SELECT * FROM t1 WHERE y1 < 2) SET y1 = 12 WHERE y1 = 2;
UPDATE 0

--插入子查询带READ ONLY
gaussdb=# UPDATE (SELECT * FROM t1 WITH READ ONLY) SET y1 = 1 WHERE y1 = 11;
ERROR:  DML operation cannot be performed on a read-only view.

--插入多表连接的子查询
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# UPDATE (SELECT * FROM t1, t2 WHERE x1 = x2) SET y1 = 11 WHERE y2 = 1;
UPDATE 1

--插入带CHECK OPTION的多表连接子查询，连接列x1, x2不可更新
gaussdb=# UPDATE (SELECT * FROM t1, t2 WHERE x1 = x2 WITH CHECK OPTION) SET y1 = 1 WHERE
y2 = 1;
UPDATE 1
gaussdb=# UPDATE (SELECT * FROM t1, t2 WHERE x1 = x2 WITH CHECK OPTION) SET x1 = 6 WHERE
y2 = 5;
ERROR:  This operation is not allowed on virtual column.

--删除SCHEMA
gaussdb=# RESET CURRENT_SCHEMA;
RESET
gaussdb=# DROP SCHEMA upd_subqry CASCADE;
NOTICE:  drop cascades to 2 other objects
DETAIL:  drop cascades to table upd_subqry.t1
drop cascades to table upd_subqry.t2
DROP SCHEMA
```

示例2： 更新视图

```
--创建SCHEMA
gaussdb=# CREATE SCHEMA upd_view;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = 'upd_view';
SET

--创建表并插入数据
gaussdb=# CREATE TABLE t1 (x1 int, y1 int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'x1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE TABLE t2 (x2 int PRIMARY KEY, y2 int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t2_pkey" for table "t2"
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4
gaussdb=# INSERT INTO t2 VALUES (1, 1), (2, 2), (3, 3), (5, 5);
INSERT 0 4

--创建单表视图
gaussdb=# CREATE VIEW v_upd1 AS SELECT * FROM t1;
CREATE VIEW
gaussdb=# CREATE VIEW v_upd_read AS SELECT * FROM t1 WITH READ ONLY;
CREATE VIEW

--通过视图对t1更新
gaussdb=# UPDATE v_upd1 SET y1 = 13 where y1 = 3;
UPDATE 1
gaussdb=# UPDATE v_upd_read SET y1 = 1 WHERE y1 = 11;
ERROR:  DML operation cannot be performed on a read-only view.

--创建多表视图
gaussdb=# CREATE VIEW vv_upd AS SELECT * FROM t1, t2 WHERE x1 = x2;
CREATE VIEW
gaussdb=# CREATE VIEW vv_upd_wco AS SELECT * FROM t1, t2 WHERE x1 = x2 WITH CHECK OPTION;
CREATE VIEW

--通过连接视图对t1更新
gaussdb=# UPDATE vv_upd SET y1 = 1 WHERE y2 = 1;
UPDATE 1
gaussdb=# UPDATE vv_upd_wco SET x1 = 6 WHERE y2 = 5;
ERROR:  This operation is not allowed on virtual column.

--删除SCHEMA
gaussdb=# RESET CURRENT_SCHEMA;
RESET
gaussdb=# DROP SCHEMA upd_view CASCADE;
NOTICE:  drop cascades to 6 other objects
DETAIL:  drop cascades to table upd_view.t1
```


版权所有 © 华为云计算技术有限公司

```
drop cascades to table upd_view.t2
drop cascades to view upd_view.v_upd1
drop cascades to view upd_view.v_upd_read
drop cascades to view upd_view.vv_upd
drop cascades to view upd_view.vv_upd_wco
DROP SCHEMA
ERROR:  virtual column not allowed here
```

1.13.21.1.1 伪更新分布列

功能描述

执行UPDATE语句更新分布列时，GaussDB会检查等值的分布列，并通过查询重写机制消除未实际参与更新的分布列信息，从而避免未实际更新的分布列删除或重建操作导致的SELECT FOR UPDATE语法可见性报错。该功能由GUC参数enable_remove_redundant_update_col控制，参数取值为on时开启该功能。

该功能开启后实际效果如下所示：

原始用户语句：

```
UPDATE TABLE SET distkey=value,amount=10000 WHERE distkey=value;
```

查询重写后语句：

```
UPDATE TABLE SET amount=10000 WHERE distkey=value;
```

说明

该功能开启前，伪更新分布列会有如下报错（不会有额外报错信息提示）：

```
ERROR:  Distributed key cannot be updated in the current version.
```

约束

- 仅支持ORA兼容性。
- 仅支持HASH分布的表。
- 伪更新分布列语法约束：
  - 仅支持UPDATE语法，不支持MERGE INTO、REPLACE INTO、UPSERT语法。
  - 仅支持在查询重写阶段能计算出实际值的场景。包含以下场景：
    - 普通const常量：数字、字符串等值传递数据。
    - 参数类型：PBE报文中的常量值传递（数字、字符串等）。
    - 函数类型：immutable函数（stable、volatile函数不支持）。
    - 操作符类型：入参为常量的一般操作符（+-*/等）。
    - 支持部分bool表达式类型：or、and、not传参为常量的场景。
    - 显式类型转换：常量显式类型转换场景（'1.23'::numeric）。
    - 条件表达式类型：case when参数为常量场景。nvl/coalesce参数为常量场景。
    - NULL类型：NULL值场景。

版权所有 © 华为云计算技术有限公司

  - 仅支持等值表达式，不支持大于、小于、between、in等区间范围。
  - 不支持子查询场景。
  - 不支持UPDATE和CTE的组合场景。
  - 不支持更新视图。
- 伪更新分布列可消除的场景，支持消除后不更新分布列场景，可不遵循**说明**中罗列的不支持场景。
- 伪更新分布列会强行选择cplan，计划选择的优先级如下：
  /*+ use_gplan*/ > 伪更新分布列 > **plan_cache_mode**。

示例

- 示例1：nvarchar2类型支持伪更新分布列

```
-- 创建测试用数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db

-- 创建hash分布表。
ora_db=# CREATE TABLE test_hash(
    id        nvarchar2(10),
    name      VARCHAR(20),
    province VARCHAR(60),                    --省
    country   VARCHAR(30) DEFAULT 'Country1'        --国籍
)DISTRIBUTE BY HASH(id);
ora_db=# INSERT INTO test_hash VALUES ('0000001', 'Bob',   'City1', 'Country1'),
    ('0000002', 'Jack',  'City2',  'Country1'),
    ('0000003', 'Scott', 'City2',  'Country1');

-- 原场景报错。
ora_db=# UPDATE test_hash SET id = '0000001', name = 'bob1' WHERE id = '0000001';
ERROR:  Distributed key cannot be updated in the current version.

-- 开启功能后不报错。
ora_db=# SET enable_remove_redundant_update_col=on;
ora_db=# UPDATE test_hash SET id = '0000001', name = 'bob1' WHERE id = '0000001';
UPDATE 1

-- 计划展示（set中投影列id = '0000001'，被成功消减）。
ora_db=# EXPLAIN VERBOSE UPDATE test_hash SET id = '0000001', name = 'bob1' WHERE id =
'0000001';
                                    QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Data Node Scan on "__REMOTE_FQS_QUERY__"  (cost=0.00..0.00 rows=0 width=0)
   Node/s: datanode1
   Remote query: UPDATE public.test_hash SET name = 'bob1'::character varying WHERE id::text =
'0000001'::text
(3 rows)

-- 删除测试用数据库，请用实际初始数据库名替换postgres。
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

- 示例2：多分布键支持伪更新分布列

```
-- 创建测试用数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db

-- 开启功能。
ora_db=# SET enable_remove_redundant_update_col=on;

-- 创建hash分布表。
ora_db=# CREATE TABLE test_hash_multi_distcol(
    id        int,
    name      VARCHAR(20),
    province VARCHAR(60),                    --省
```


版权所有 © 华为云计算技术有限公司

```
    country   VARCHAR(30) DEFAULT 'Country1'        --国籍
)DISTRIBUTE BY HASH(id,name);
ora_db=# INSERT INTO test_hash_multi_distcol VALUES (1, 'Bob',   'City1', 'Country1'),
    (2, 'Jack',  'City1',  'Country1'),
    (3, 'Scott', 'City2',  'Country1');

-- 开启功能后不报错。
ora_db=# UPDATE test_hash_multi_distcol SET id = 1,name='Bob', province = 'City1' WHERE id = 1
and name='Bob';
UPDATE 1

-- 计划展示。
ora_db=# EXPLAIN VERBOSE UPDATE test_hash_multi_distcol SET id = 1,name='Bob', province =
'City1' WHERE id = 1 AND name='Bob';
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
-----------------------
 Data Node Scan on "__REMOTE_FQS_QUERY__"  (cost=0.00..0.00 rows=0 width=0)
   Node/s: datanode3
   Remote query: UPDATE public.test_hash_multi_distcol SET province = 'City1'::character varying
WHERE id = 1 AND name::text = 'Bob'::text
(3 rows)

-- 删除测试用数据库，请用实际初始数据库名替换postgres。
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

- 示例3：PBE场景支持伪更新分布列

```
-- 创建测试用数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db

-- 开启功能。
ora_db=# SET enable_remove_redundant_update_col=on;

-- 创建hash分布表。
ora_db=# CREATE TABLE test_hash_int(
    id       int,
    name     VARCHAR(20),
    province VARCHAR(60),                       --省
    country   VARCHAR(30) DEFAULT 'Country1'        --国籍
)DISTRIBUTE BY HASH(id);
ora_db=# INSERT INTO test_hash_int VALUES (1, 'Bob',   'City1', 'Country1'),
    (2, 'Jack',  'City2',  'Country1'),
    (3, 'Scott', 'City2',  'Country1');

ora_db=# SELECT a.count,b.node_name
    FROM (SELECT COUNT(*) AS count, xc_node_id FROM test_hash_int GROUP BY xc_node_id) a,
    pgxc_node b
WHERE a.xc_node_id=b.node_id ORDER BY a.count DESC;
 count | node_name
-------+-----------
     2 | datanode2
     1 | datanode1
(2 rows)

-- 开启功能后不报错。
ora_db=# PREPARE p1 AS UPDATE test_hash_int SET id = $1, name = $2 WHERE id = $3;
PREPARE
ora_db=# EXECUTE p1(1,'bob1',1);
UPDATE 1

-- 删除测试用数据库，请用实际初始数据库名替换postgres。
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

- 示例4：timestamp和text隐式类型转换不支持

```
-- 创建测试用数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db
```


版权所有 © 华为云计算技术有限公司

```
-- 开启功能。
ora_db=# SET enable_remove_redundant_update_col = on;

-- 创建hash分布表。
ora_db=# CREATE TABLE distcol_update_timestamp(c1 timestamp(6),c2 int,c3 text) DISTRIBUTE BY
hash(c1);
ora_db=# INSERT INTO distcol_update_timestamp VALUES ('2026-03-05
09:25:19.411077+08',1,'gaussdb'),
    ('2026-03-05 09:25:19.411077+08',2,'gaussdb'),
    ('2026-03-05 09:25:20.411077+08',3,'gaussdb'),
    ('2026-03-05 09:25:21.411077+08',4,'gaussdb'),
    ('2026-03-05 09:25:22.411077+08',5,'gaussdb'),
    ('2026-03-06 09:25:22.411077+08',6,'gaussdb'),
    ('2026-03-07 09:25:22',7,'gaussdb');
ora_db=# CREATE INDEX ON distcol_update_timestamp(c1);

ora_db=# EXPLAIN VERBOSE UPDATE distcol_update_timestamp SET c1 = '2026-03-05
09:25:20.411077+08',c2 = -3
WHERE c1 = '2026-03-05 09:25:20.411077+08'::varchar2;
                                                                QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------
 Data Node Scan on "__REMOTE_FQS_QUERY__"  (cost=0.00..0.00 rows=0 width=0)
   Node/s: datanode2
   Remote query: UPDATE public.distcol_update_timestamp SET c2 = (-3) WHERE c1 = '2026-03-05
09:25:20.411077+08'::character varying::timestamp without time zone
(3 rows)

ora_db=# EXPLAIN VERBOSE UPDATE distcol_update_timestamp SET c1 = '2026-03-05
09:25:20.411077+08',c2 = -3
WHERE c1 = '2026-03-05 09:25:20.411077+08';
                                                         QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
--------------------
 Data Node Scan on "__REMOTE_FQS_QUERY__"  (cost=0.00..0.00 rows=0 width=0)
   Node/s: datanode2
   Remote query: UPDATE public.distcol_update_timestamp SET c2 = (-3) WHERE c1 = '2026-03-05
09:25:20.411077'::timestamp without time zone
(3 rows)

ora_db=# EXPLAIN VERBOSE UPDATE distcol_update_timestamp SET c1 = '2026-03-05
09:25:20.411077+08',c2 = -3
WHERE c1 = '2026-03-05 09:25:20.411077+08'::text;
ERROR:  Distributed key cannot be updated in the current version.

-- 两个隐式类型转换函数，一个是stable的一个是immutable的，可以通过provolatile列看到text隐式类型
转换到timestamp为s类型，不支持。
ora_db=# SELECT * FROM pg_proc WHERE proname IN ('varchar_timestamp','text_timestamp');
      proname      | pronamespace | proowner | prolang | procost | prorows | provariadic | protransform |
proisagg | proiswindow | prosecdef | proleakproof | proisstrict | proretset | provolatile |
pronargs | pronargdefaults | prorettype | proargtypes | proallargtypes | proargmodes | proargnames |
proargdefaults |       prosrc       | probin | proconfig | proacl | prodefaultargpos | fencedmode
| proshippable | propackage | prokind | proargsrc | propackageid | proisprivate | proargtypesext |
prodefaultargposext | allargtypes | allargtypesext
------------------+--------------+----------+---------+---------+---------+-------------+--------------+----------
+-------------+-----------+--------------+-------------+-----------+-------------+-
---------+-----------------+------------+-------------+----------------+-------------+-------------+----------------
+-------------------+--------+-----------+--------+------------------+------------
+--------------+------------+---------+-----------+--------------+--------------+----------------
+---------------------+------------+----------------
 text_timestamp    |          11 |       10 |      12 |       1 |       0 |          0 | -          | f        | f          |
f         | f           | t          | f        | s          |
       1 |            0 |       1114 | 25         |            |           |           |            | text_timestamp
|       |          |       |              | f
|           | f         | f       |          |          0 | f          |             |               | 25         |
 varchar_timestamp |          11 |       10 |      12 |       1 |       0 |          0 | -          | f        | f          |
f         | f           | t          | f        | i          |
       1 |            0 |       1114 | 1043       |            |           |           |            |
```


版权所有 © 华为云计算技术有限公司

```
varchar_timestamp |       |          |       |              | f
|           | f         | f      |          |          0 | f          |              |                 | 1043       |
(2 rows)

-- 删除测试用数据库，请用实际初始数据库名替换postgres。
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

说明

示例中强制转换varchar2类型的操作，受GUC参数modify_function_property_json影响：

- 将GUC参数取值设置为空，或设置GUC参数中oid为4186的配置项中，property_name为provolatile的对象，对应的property_value从s改为i，并重启数据库，正常执行返回结果。
- 不修改modify_function_property_json参数，会报错，符合预期，原因同强转text一样，函数易变性为stable，报错如下：

```
ora_db=# EXPLAIN VERBOSE UPDATE distcol_update_timestamp SET c1 = '2026-03-05
09:25:20.411077+08',c2 = -3 WHERE c1 = '2026-03-05 09:25:20.411077+08'::varchar2;
ERROR:  Distributed key cannot be updated in the current version.
```

- 示例5：NVL等系统函数和隐式类型转换不支持

```
-- 创建测试用数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db

-- 开启功能。
ora_db=# SET enable_remove_redundant_update_col=on;

-- 创建hash分布表。
ora_db=# CREATE TABLE distcol_update_timestamp(c1 timestamp(6),c2 int,c3 text) DISTRIBUTE BY
HASH(c1);
ora_db=# INSERT INTO distcol_update_timestamp VALUES ('2026-03-05
09:25:19.411077+08',1,'gaussdb'),
    ('2026-03-05 09:25:19.411077+08',2,'gaussdb'),
    ('2026-03-05 09:25:20.411077+08',3,'gaussdb'),
    ('2026-03-05 09:25:21.411077+08',4,'gaussdb'),
    ('2026-03-05 09:25:22.411077+08',5,'gaussdb'),
    ('2026-03-06 09:25:22.411077+08',6,'gaussdb'),
    ('2026-03-07 09:25:22',7,'gaussdb');
ora_db=# CREATE INDEX ON distcol_update_timestamp(c1);

-- 更新语句。
ora_db=# UPDATE distcol_update_timestamp SET c1 = '2026-03-05 09:25:20.411077+08',c2 = -3
WHERE c1 = nvl('2026-03-05 09:25:20.411077+08','2026-03-05 09:25:21.411077+08');
ERROR:  Distributed key cannot be updated in the current version.

-- 删除测试用数据库，请用实际初始数据库名替换postgres。
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

- **示例6：**bigint、int类型执行伪分布列更新set a = null,b = 1 where a = null报错不支持

```
-- 创建测试用数据库。
gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
gaussdb=# \c ora_db

-- 创建hash分布表。
ora_db=# CREATE TABLE distcol_update_bigint(c1 bigint,c2 int,c3 text) DISTRIBUTE BY hash(c1);
ora_db=# INSERT INTO distcol_update_bigint VALUES(null,9,'gaussdb');

-- 设置参数，更新语句。
ora_db=# SET enable_remove_redundant_update_col=on;
ora_db=# SET a_format_version = '10c';
ora_db=# SET a_format_dev_version = 's6';
ora_db=# UPDATE distcol_update_bigint SET c1 = null,c3 = 'update4' WHERE c1 = null;
ERROR:  Distributed key cannot be updated in the current version.

-- 删除测试用数据库，请用实际初始数据库名替换postgres。
```


版权所有 © 华为云计算技术有限公司

```
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

1.13.22 V

1.13.22.1 VACUUM

功能描述

VACUUM回收表或B-Tree索引中已经删除的行所占据的存储空间。在一般的数据库操作里，那些已经DELETE的行并没有从它们所属的表中物理删除；在完成VACUUM之前它们仍然存在。因此有必要周期地运行VACUUM，特别是在经常更新的表上。

注意事项

- 如果没有参数，VACUUM处理当前数据库里用户拥有相应权限的每个表。如果参数指定了一个表，VACUUM只处理指定的表。
- 要对一个表进行VACUUM操作，通常用户必须是表的所有者或者被授予了指定表VACUUM权限的用户，三权分立开关关闭时，默认系统管理员有该权限。数据库的所有者允许对数据库中除了共享目录以外的所有表进行VACUUM操作（该限制意味着只有系统管理员才能真正对一个数据库进行VACUUM操作）。VACUUM命令会跳过那些用户没有权限的表进行垃圾回收操作。
- VACUUM不能在事务块内执行。
- 建议生产数据库经常清理（至少每晚一次），以保证不断地删除失效的行。尤其是在增删了大量记录之后，对受影响的表执行VACUUM ANALYZE命令是一个很好的习惯。这样将更新系统目录为最近的更改，并且允许查询优化器在规划用户查询时有更好的选择。
- 不建议日常使用FULL选项，但是可以在特殊情况下使用。例如在用户删除了一个表的大部分行之后，希望从物理上缩小该表以减少磁盘空间占用。VACUUM FULL通常要比单纯的VACUUM收缩更多的表尺寸。如果执行此命令后所占用物理空间无变化（未减少），请确认是否有其他活跃事务（删除数据事务开始之前开始的事务，并在VACUUM FULL执行前未结束）存在，如果有等其他活跃事务退出进行重试。
- VACUUM FULL通过重建表的方式将表内空闲空间归还给表空间，重建过程需要额外申请表中有效数据相当的存储空间，存储空间不足时执行VACUUM FULL可能会因为空间不足导致集群变为只读模式。对于非段页式表，VACUUM FULL执行结束后，原表所占物理文件会被删除，原表所占的物理文件的空间会归还给操作系统；对于段页式表，VACUUM FULL执行结束后，原表所占的物理空间，会被归还给段页式数据文件，不会归还给操作系统。
- VACUUM会导致I/O流量的大幅增加，这可能会影响其他活动会话的性能。因此，有时候会建议使用基于开销的VACUUM延迟特性。
- 如果指定了VERBOSE选项，VACUUM将打印处理过程中的信息，以表明当前正在处理的表。各种有关当前表的统计信息也会打印出来。
- 当含有带括号的选项列表时，选项可以以任何顺序写入。如果没有括号，则选项必须按语法显示的顺序给出。
- VACUUM和VACUUM FULL时，会根据参数vacuum_defer_cleanup_age延迟清理行存表记录，即不会立即清理刚刚删除的元组。
- VACUUM ANALYZE先执行一个VACUUM操作，然后给每个选定的表执行一个ANALYZE。对于日常维护脚本而言，这是一个很方便的组合。

版权所有 © 华为云计算技术有限公司

- 简单的VACUUM（不带FULL选项）只是简单地回收空间并且令其可以再次使用。这种形式的命令可以和对表的普通读写并发操作，因为没有请求排他锁。
  VACUUM FULL执行更广泛的处理，包括跨块移动行，以便把表压缩到最少的磁盘块数目里。这种形式要慢许多并且在处理的时候需要在表上施加一个排他锁。
- 如果没有打开xc_maintenance_mode参数，那么VACUUM FULL会跳过所有系统表。
- VACUUM HDFSDIRECTORY为预留语法。
- 执行DELETE后立即执行VACUUM FULL命令不会回收空间。执行DELETE后再执行1000个非SELECT事务，或者等待1s后再执行1个事务，之后再执行VACUUM FULL命令空间才会回收。
- VACUUM FULL期间会对表加排他锁，不建议在业务高峰期运行VACUUM FULL，可能导致等锁时间过长或者死锁。
- 为保证性能和统计信息的准确性，避免VACUUM ANALYZE、AUTOANALYZE、手动ANALYZE等涉及ANALYZE的命令同时执行或执行过于频繁。
- Ustore手动VACUUM与Astore手动VACUUM行为一致，会对堆表、索引等进行加锁清理；而Ustore的AUTOVACUUM仅做分区表GPI清理、堆表FSM更新以及索引页面回收。
- VACUUM FULL分区表时，会遍历分区进行清理，并在分区清理后重建GPI，因此当分区较多时，建议先删除GPI，在VACUUM FULL执行完成后重新创建索引，以此降低VACUUM FULL的执行时间。对于在线VACUUM FULL，对象为分区表时，会在进行分区清理的同时重建GPI；仅当对象为单分区时，才会在分区清理后再重建GPI。故在线VACUUM FULL无需先删除GPI。
- 在线VACUUM FULL不支持对于数据库级、索引级、段页式表、hash bucket表、临时表、unlogged表对象执行在线VACUUM FULL。如对象涉及以上不支持的类型，ONLINE关键字将不会生效，VACUUM FULL默认以离线方式执行，并提示当前VACUUM FULL语句不支持在线执行。
- 执行在线VACUUM FULL操作时，需预留足够的磁盘空间，即剩余空间是所操作的表（及其上索引等附属品）总和。
- 如果用户取消在线VACUUM FULL执行过程，首先进入残留清理流程，如果用户再次取消，则会结束清理流程导致残留。
- 在线VACUUM FULL与用户业务之间存在资源争抢，不适合大业务背景下做在线VACUUM FULL。对于资源有限的场景，建议采用细粒度资源管控进行操作，且不建议使用并行加速功能。
- M-Compatibility模式下不支持在线VACUUM FULL操作。
- 在线VACUUM FULL不支持FREEZE关键字，若同时指定FREEZE关键字与ONLINE关键字，VACUUM FULL默认以离线方式执行，并提示当前VACUUM FULL语句不支持在线执行。
- 在线VACUUM FULL不支持在事务内执行（通过JDBC驱动执行多条SQL时会隐式启动事务，禁止多条SQL一起执行），不支持在存储过程中执行。
- 若在线VACUUM FULL操作失败，表上可能会有明显存储空间膨胀。其中膨胀程度与在线VACUUM FULL执行时长以及并发写操作量成正比。长事务可能会阻塞在线VACUUM FULL，建议避免在长事务存在时进行在线VACUUM FULL。
- 在线VACUUM FULL执行后期将阻塞DQL、DML操作。
- 在线VACUUM FULL执行期间，应尽量避免与其他DDL并发执行，否则可能会出现死锁、报错等；若出现此类情况，可以重新连接客户端进行重试。
- 在线VACUUM FULL过程中会生成名为online$$ddl$$[hash值]的模式。

版权所有 © 华为云计算技术有限公司

  - 模式的属主默认为初始用户，不建议在线VACUUM FULL执行中操作此模式，可能会导致中止在线VACUUM FULL并产生未知错误。不建议用户对该模式下的表格进行查询。gs_dump导出会默认不导出此Schema。
  - 用户不可创建online$$为前缀的模式，会有报错提示。若已存在同名模式冲突，在线VACUUM FULL将不会生效，在线VACUUM FULL默认以离线方式执行，并提示当前VACUUM FULL语句是因为模式名冲突不可在线。
- 若对象表上存在全局二级索引（GSI），ONLINE关键字将不会生效，VACUUM FULL默认以离线方式执行，并提示当前VACUUM FULL语句不支持在线执行。
- 避免同时执行在线VACUUM FULL操作与扩缩容操作，否则可能出现在线VACUUM FULL操作或扩缩容执行失败的情况。
- 除VACUUM ANALYZE语句，其他VACUUM操作，在原有行为不变更的情况下，不会在统计信息中上报分区表主表的relpages/reltuples字段信息。
- 在PG_PARTITION系统表的reloptions字段中，wait_clean_gpi表示是否需要等待清理全局分区索引，如果存在wait_clean_gpi=y的待清理元组，则该元组无法直接被清理，需要等待清理完GPI后才能被清理。

语法格式

- 回收空间并更新统计信息，对关键字顺序无要求。
```
VACUUM [ ( { FULL | FREEZE | VERBOSE | { ANALYZE | ANALYSE } } [, ... ] ) ]
    [ table_name [ (column_name [, ...] ) ] [ PARTITION ( partition_name ) | SUBPARTITION
( subpartition_name ) ] ]
    [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ];
```
- 仅回收空间，不更新统计信息。
```
VACUUM [ FULL ] [ FREEZE ] [ VERBOSE ] [ table_name [ PARTITION ( partition_name ) |
SUBPARTITION ( subpartition_name ) ] ]
    [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ];
```
- 回收空间并更新统计信息，且对关键字顺序有要求。
```
VACUUM [ FULL ] [ FREEZE ] [ VERBOSE ] { ANALYZE | ANALYSE } [ VERBOSE ]
    [ table_name [ ( column_name [, ...] ) ] ] [ PARTITION ( partition_name ) ]
    [ OFFLINE | [ ONLINE [ WITH ( { online_parameter = value } [, ... ] ) ] ] ];
```
- 仅回收指定bucket的空间，更新pg_hashbucket系统表的bucketxid列，不更新统计信息。仅支持在扩容期间调用，仅支持管理员权限用户调用。
```
VACUUM FREEZE BUCKETS (bucketid [, ...]);
```

参数说明

- **FULL**

  选择“FULL”清理，这样可以恢复更多的空间，但是需要耗时更多，并且在表上施加了排他锁。

  说明

  使用FULL参数会导致统计信息丢失，如果需要收集统计信息，请在VACUUM FULL语句中加上ANALYZE关键字。
- **FREEZE**

  指定FREEZE相当于执行VACUUM时将vacuum_freeze_min_age参数设为0。
- **VERBOSE**

  为每个表打印一份详细的清理工作报告。
- **ANALYZE | ANALYSE**

  更新用于优化器的统计信息，以决定执行查询的最有效方法。

 版权所有 © 华为云计算技术有限公司

- **table_name**

  要清理的表的名称（可以有模式修饰）。

  取值范围：要清理的表的名称。缺省时为当前数据库中的所有表。

- **column_name**

  要分析的具体的字段名称，需要配合analyze选项使用。

  取值范围：要分析的具体的字段名称。缺省时为所有字段。

  说明

  由于VACUUM ANALYZE语句的机制是依次执行VACUUM和ANALYZE，因此当column_name错误时，会存在VACUUM执行成功但ANALYZE执行失败的情况；对于分区表，则会出现对某个分区VACUUM执行成功之后ANALYZE执行失败的情况。

- **partition_name**

  要清理的表的分区名称。缺省时为所有分区。

- **ONLINE | OFFLINE**

  以在线或者离线方式执行VACUUM。主要用于在线执行VACUUM FULL，使VACUUM FULL期间不会长时间阻塞读写业务。

  使用ONLINE | OFFLINE关键字：

  - ONLINE：以在线方式执行。此时若VACUUM FULL语句包含当前特性不支持的场景（如FREEZE关键字等），则以离线方式执行，并通过NOTICE提示忽略ONLINE关键字。
  - OFFLINE：以离线方式执行。

  取值范围：关键字OFFLINE或ONLINE，分别对应离线方式执行和在线方式执行。缺省时为OFFLINE。

  说明

  在线VACUUM FULL执行失败时，如用户指令终止命令、锁超时、数据库故障等，系统会对残留对象进行自动清理。自动清理过程可能由于等锁超时等原因而失败，无法进行自动清理，需要手动清理恢复。对于严重故障，如FATAL、PANIC、数据库故障等场景时，可能会残留临时模式对象，影响后续在线VACUUM FULL和在线DDL的执行，此时可选择使用**online_ddl_cleanup()**函数进行手动执行清理，或者选择在下次执行在线VACUUM FULL或在线DDL时自动清理。若无故障残留，不建议使用online_ddl_cleanup()函数，否则可能导致在线VACUUM FULL中止并产生未知错误。

- **online_parameter**

  当指定ONLINE关键字时，用于指定在线VACUUM FULL选项。若未指定ONLINE关键字或不满足在线VACUUM FULL的条件，该参数不生效。

  取值范围：

  - parallel_threads：控制多线程并行，用于在资源充足的情况下追求更高性能。1-32之间的整数值，缺省时默认值为1。
    - 当用户开启资源管控时，如IO资源管控 (io_limits>0或io_priority = high/medium/low)，parallel_threads将忽略用户/会话下的资源管控，并给出提示当前资源管控无效并开启多线程加速。在线VACUUM FULL多线程执行实际遵循初始用户的资源管控。
    - 可以与索引并行参数parallel_workers共同生效。parallel_workers控制VACUUM FULL期间索引构建的并行度，parallel_threads参数控制VACUUM FULL期间追增的并行度。
  - max_catchup_times：控制最大追增轮次，用于防止VACUUM FULL执行时间过长。1 - INT_MAX之间的整数值，缺省时默认值为50。

版权所有 © 华为云计算技术有限公司

    - 在线VACUUM FULL会记录数据重建过程中的IUD增量数据到临时表，并采用多轮次循环追增，将增量数据进行重建。为防止由于IUD并发数据量过大，VACUUM FULL无法完成追增的情况，用户可以设置该参数，以限制追增次数。
  - last_catchup_threshold：触发最后一轮追增的时间上限。当一轮追增完成后，若该轮耗时低于此阈值，则触发最后一轮追增及切换。取值范围：[1000,1200000]（单位：毫秒），默认值：5000。
  - online_ddl_lockwait_timeout：在线DDL的等锁超时时间。若等锁线程在该时间内未能获取锁，则报锁超时失败。触发CANCEL模式获取锁时，实际使用的超时时间为该参数值的2倍。取值范围：[1,1200000]（单位：毫秒），默认值：10000。
  - lockwait_interval：等锁步长，即连续等锁超时的重试间隔。若当前轮次等锁超时，等待该参数指定的时长后进入下一轮尝试。取值范围：[0,300]（单位：秒），默认值：5。
  - trylock_threshold：等锁尝试次数上限，需与enable_cancel配合使用。达到该上限后若仍因等锁超时而无法获取锁，则下一次尝试将采用CANCEL模式。若未开启enable_cancel，达到上限后取锁失败并报错。取值范围：[1,INT_MAX]，或 -1（表示无上限），默认值：-1。
  - enable_cancel：是否允许用户业务快速失败，即在达到等锁尝试次数上限，或在切表事务获取用户表8级锁时，通过快速中止用户业务来确保在线DDL成功获取锁。取值范围：[false,true]，默认值：false。

示例

- VACUUM

```
--创建表tbl_test，并插入数据。
gaussdb=# CREATE TABLE tbl_test(c1 int);
gaussdb=# INSERT INTO tbl_test VALUES (1);

--查看数据，和数据的ctid。
gaussdb=# SELECT ctid,* FROM tbl_test;
 ctid  | c1
-------+----
 (0,1) |  1
(1 row)

--删除该数据。
gaussdb=# DELETE FROM tbl_test;

--重新插入一条数据，发现使用了一个新的ctid。
gaussdb=# INSERT INTO tbl_test VALUES (2);
gaussdb=# SELECT ctid,* FROM tbl_test;
 ctid  | c1
-------+----
 (0,2) |  2
(1 row)

--使用VACUUM命令之后，再插入数据，发现复用了旧的空间。
gaussdb=# VACUUM ANALYZE tbl_test;
gaussdb=# INSERT INTO tbl_test VALUES (3);
gaussdb=# SELECT ctid,* FROM tbl_test;
 ctid  | c1
-------+----
 (0,1) |  3
 (0,2) |  2
(2 rows)

--删除表。
gaussdb=# DROP TABLE tbl_test;
```


版权所有 © 华为云计算技术有限公司

- VACUUM FULL

```
--建表。
gaussdb=# CREATE TABLE tbl_test2(c1 int);

--插入10万条数据并查看表的大小。
gaussdb=# INSERT INTO tbl_test2 VALUES (generate_series(1,100000));
gaussdb=# SELECT 'tbl_test2' AS tablename, pg_size_pretty(pg_relation_size('tbl_test2')) AS size;
 tablename |  size
-----------+---------
 tbl_test2 | 3048 kB
(1 row)

--删除数据并查看表大小。
gaussdb=# DELETE FROM tbl_test2;
gaussdb=# SELECT 'tbl_test2' AS tablename, pg_size_pretty(pg_relation_size('tbl_test2')) AS size;
 tablename |  size
-----------+---------
 tbl_test2 | 3048 kB
(1 row)

--执行SELECT获取当前事务的ID。
gaussdb=# SELECT TXID_CURRENT();
txid_current
-------------
       55196

--使用VACUUM FULL回收空间并同时更新统计信息,并查看表的大小。
gaussdb=# VACUUM FULL ANALYZE tbl_test2;
gaussdb=# SELECT 'tbl_test2' AS tablename, pg_size_pretty(pg_relation_size('tbl_test2')) AS size;
 tablename |  size
-----------+---------
 tbl_test2 | 0 bytes
(1 row)

--使用在线VACUUM FULL回收空间并同时更新统计信息。
gaussdb=# VACUUM FULL ANALYZE tbl_test2 ONLINE;
VACUUM

--使用在线VACUUM FULL回收空间、更新统计信息，并设置online_parameter。
gaussdb=# VACUUM FULL ANALYZE tbl_test2 ONLINE with(parallel_threads=10,
max_catchup_times=20, last_catchup_threshold=3000, online_ddl_lockwait_timeout=10,
lockwait_interval=10, trylock_threshold=1, enable_cancel=true);
VACUUM

--删除。
gaussdb=# DROP TABLE tbl_test2;
```

优化建议

- vacuum
  - VACUUM不能在事务块内执行。
  - 建议生产数据库经常清理（至少每晚一次），以保证不断地删除失效的行。尤其是在增删了大量记录后，对相关表执行VACUUM ANALYZE命令。
  - 不建议日常使用FULL选项，但是可以在特殊情况下使用。例如，一个例子就是在用户删除了一个表的大部分行之后，希望从物理上缩小该表以减少磁盘空间占用。

1.13.22.2 VALUES

功能描述

根据给定的值表达式计算一个或一组行的值。它通常用于在一个较大的命令内生成一个“常数表”。

版权所有 © 华为云计算技术有限公司

注意事项

- 应当避免使用VALUES返回数量非常大的结果行，否则可能会出现内存耗尽或者性能低下的情况。出现在INSERT中的VALUES是一个特殊情况，因为目标字段类型可以从INSERT的目标表获知，并不需要通过扫描VALUES列表来推测，所以在此情况下可以处理非常大的结果行。
- 如果指定了多行，那么每一行都必须拥有相同的元素个数。
- 在使用IN表达式进行多列条件查询时，IN LIST和IN VALUES LIST虽然功能相似，但其内部处理机制存在本质差异。为确保查询性能和数据准确性，在替换时，需特别关注以下关键区别：
  - 逻辑结构差异：IN LIST会将多列条件解析为多条OR条件的组合，例如：(a, b) IN ((1, 2), (2, 3))会被转化为(a = 1 AND b = 2) OR (a = 2 AND b = 3)，当IN后的列表长度极大时，生成的OR条件链将异常冗长，查询优化器需遍历整条条件链，可能导致查询性能显著下降。
  - 查询优化机制：IN VALUES LIST会将VALUES子句视为派生表，将IN表达式解析为ANY类型的子查询，并通过查询改写阶段将其转换为JOIN操作。该机制有效避免了处理长OR条件链的性能开销。
  - 类型处理差异：IN LIST对每组数据单独进行类型检查和隐式转换，而IN VALUES LIST则先为VALUES子句中各列统一推导公共数据类型，再进行类型检查和转换。这种统一类型处理方式更为严格，要求VALUES子句中同一列的数据类型保持一致，确保查询结果的准确性。

语法格式

```
VALUES {( expression [, ...] )} [, ...]
    [ ORDER BY { sort_expression [ ASC | DESC | USING operator ] } [, ...] ]
    [ LIMIT { count | ALL } ]
    [ OFFSET start [ ROW | ROWS ] ]
    [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ];
```

参数说明

- **expression**

  用于计算或插入结果表指定地点的常量或者表达式。

  在一个出现在INSERT顶层的VALUES列表中，expression可以被DEFAULT替换以表示插入目的字段的缺省值。除此以外，当VALUES出现在其他场合的时候是不能使用DEFAULT的。

- **sort_expression**

  一个表示如何排序结果行的表达式或者整数常量。

- **ASC**

  指定按照升序排列。

- **DESC**

  指定按照降序排列。

- **operator**

  一个排序操作符。

- **count**

  返回的最大行数。

- **OFFSET start [ROW | ROWS]**

  声明返回的最大行数，而start声明开始返回行之前忽略的行数。

版权所有 © 华为云计算技术有限公司

- **FETCH { FIRST | NEXT } [count] { ROW | ROWS } ONLY**
  FETCH子句限定返回查询结果从第一行开始的总行数，count的缺省值为1。

示例

```
--建表。
gaussdb=# CREATE TABLE test_t2(col1 INT,col2 VARCHAR);
gaussdb=# CREATE TABLE test_t3(col1 INT,col2 VARCHAR);

--插入多条数据。
gaussdb=# INSERT INTO test_t2 (col1, col2) VALUES (10,'AA'),(20,'BB'),(30,'CC');

--分页查询，返回第一页，每页显示1行数据。
gaussdb=# SELECT * FROM test_t2 ORDER BY col1 ASC LIMIT 1 OFFSET 0;
col1 | col2
------+------
   10 | AA
(1 row)

--把test_t2中的数据插入到test_t3中。
gaussdb=# INSERT INTO test_t3 SELECT * FROM test_t2;

--结合FETCH查询,返回第一页，每页显示1行数据。
gaussdb=# SELECT * FROM test_t3 ORDER BY col1 ASC OFFSET 0 FETCH FIRST 1 ROWS ONLY;
 col1 | col2
------+------
   10 | AA
(1 row)

--删除表test_t2、test_t3。
gaussdb=# DROP TABLE test_t2;
gaussdb=# DROP TABLE test_t3;
```

1.14 附录

1.14.1 扩展函数

下表列举了GaussDB中支持的扩展函数，不作为商用特性交付，仅供参考。

分类	函数名称	描述
触发器函数	pg_get_triggerdef(trigger_oid)	为触发器获取CREATE [CONSTRAINT] TRIGGER命令
	pg_get_triggerdef(trigger_oid, pretty_bool)	为触发器获取CREATE [CONSTRAINT] TRIGGER命令

1.14.2 美元引用的字符串常量

如果在字符串序列中包含有'(单引号)，那么应当将'(单引号)加倍为''(两个单引号)否则sql语句很可能无法执行。

如果字符串中包含很多单引号或者反斜杠，那么理解字符串的内容可能就会变得很苦涩，并且容易出错，因为单引号都要加倍。

为了让这种场合下的查询更具可读性，允许另外一种称作“美元符界定”的字符串常量书写办法。一个通过美元符界定声明的字符串常量由一个美元符号($)、零个或多个

版权所有 © 华为云计算技术有限公司

字符组成的"记号"、另一个美元符号、组成字符串常量的任意字符序列、一个美元符号、与前面相同的记号、一个美元符号组成的。

```
gaussdb=# SELECT $$it's an example$$;
   ?column?
-----------------
 it's an example
(1 row)
```

1.14.3 行表达式函数白名单

表 1-407 为数据对象增加或修改策略 ILM 所支持的行表达式函数白名单

func_oid_value	func_name
38	int2in
39	int2out
42	int4in
43	int4out
46	textin
47	textout
56	boollt
57	boolgt
60	booleq
61	chareq
63	int2eq
64	int2lt
65	int4eq
66	int4lt
67	texteq
70	charne
72	charle
73	chargt
74	charge
77	int4
78	char
84	boolne
111	numeric_fac

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
141	int4mul
144	int4ne
145	int2ne
146	int2gt
147	int4gt
148	int2le
149	int4le
150	int4ge
151	int2ge
152	int2mul
153	int2div
154	int4div
155	int2mod
156	int4mod
157	textne
158	int24eq
159	int42eq
160	int24lt
161	int42lt
162	int24gt
163	int42gt
164	int24ne
165	int42ne
166	int24le
167	int42le
168	int24ge
169	int42ge
170	int24mul
171	int42mul
172	int24div

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
173	int42div
176	int2pl
177	int4pl
178	int24pl
179	int42pl
180	int2mi
181	int4mi
182	int24mi
183	int42mi
200	float4in
201	float4out
202	float4mul
203	float4div
204	float4pl
205	float4mi
206	float4um
207	float4abs
209	float4larger
211	float4smaller
212	int4um
213	int2um
214	float8in
215	float8out
216	float8mul
217	float8div
218	float8pl
219	float8mi
220	float8um
221	float8abs
223	float8larger

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
224	float8smaller
228	dround
229	dtrunc
235	float8
236	float4
237	int2
238	int2
244	timepl
245	timemi
248	intinterval
249	tintervalrel
251	abstimeeq
252	abstimene
253	abstimelt
254	abstimegt
255	abstimele
256	abstimege
257	reltimeeq
258	reltimene
259	reltimelt
260	reltimegt
261	reltimele
262	reltimege
263	tintervalsame
264	tintervalct
265	tintervalov
266	tintervalleneq
267	tintervallenne
268	tintervallenlt
269	tintervallengt

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
270	tintervallenle
271	tintervallenge
273	tintervalend
275	isfinite
279	float48mul
280	float48div
281	float48pl
282	float48mi
283	float84mul
284	float84div
285	float84pl
286	float84mi
287	float4eq
288	float4ne
289	float4lt
290	float4le
291	float4gt
292	float4ge
293	float8eq
294	float8ne
295	float8lt
296	float8le
297	float8gt
298	float8ge
299	float48eq
300	float48ne
301	float48lt
302	float48le
303	float48gt
304	float48ge

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
305	float84eq
306	float84ne
307	float84lt
308	float84le
309	float84gt
310	float84ge
311	float8
312	float4
313	int4
314	int2
316	float8
317	int4
318	float4
319	int4
350	btint2cmp
351	btint4cmp
352	year_int1_mysql
354	btfloat4cmp
355	btfloat8cmp
357	btabstimecmp
358	btcharcmp
360	bttextcmp
377	cash_cmp
380	btreltimecmp
381	bttintervalcmp
385	regexp_count
386	regexp_count
387	regexp_count
400	hashtext
432	hash_numeric

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
449	hashint2
450	hashint4
451	hashfloat4
452	hashfloat8
454	hashchar
458	text_larger
459	text_smaller
460	int8in
461	int8out
462	int8um
463	int8pl
464	int8mi
465	int8mul
466	int8div
467	int8eq
468	int8ne
469	int8lt
470	int8gt
471	int8le
472	int8ge
474	int84eq
475	int84ne
476	int84lt
477	int84gt
478	int84le
479	int84ge
480	int4
481	int8
482	float8
483	int8

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
630	regexp_instr
631	regexp_instr
632	regexp_instr
633	regexp_instr
634	regexp_instr
652	float4
653	ftoi8
654	hashint1_numeric
665	hashint2_numeric
667	hashint16
668	bpchar
669	varchar
676	mktinterval
682	hashint4_numeric
714	int2
720	octet_length
721	get_byte
722	set_byte
723	get_bit
724	set_bit
740	text_lt
741	text_le
742	text_gt
743	text_ge
754	int8
755	hashint8_numeric
766	int4inc
768	int4larger
769	int4smaller
770	int2larger

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
771	int2smaller
784	tintervaleq
785	tintervalne
786	tintervallt
787	tintervalgt
788	tintervalle
789	tintervalge
792	btint12cmp
793	btint14cmp
794	btint18cmp
795	btint116cmp
796	btint1numericcmp
797	btint21cmp
798	btint216cmp
799	btint2numericcmp
800	btint41cmp
801	btint416cmp
802	btint4numericcmp
803	btint81cmp
804	btint816cmp
805	btint8numericcmp
837	int82pl
838	int82mi
839	int82mul
840	int82div
841	int28pl
842	btint8cmp
846	cash_mul_flt4
847	cash_div_flt4
848	flt4_mul_cash

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
849	position
852	int48eq
853	int48ne
854	int48lt
855	int48gt
856	int48le
857	int48ge
860	bpchar
862	int4_mul_cash
863	int2_mul_cash
864	cash_mul_int4
865	cash_div_int4
866	cash_mul_int2
867	cash_div_int2
868	strpos
870	lower
871	upper
877	substr
883	substr
888	cash_eq
889	cash_ne
890	cash_lt
891	cash_le
892	cash_gt
893	cash_ge
894	cash_pl
895	cash_mi
896	cash_mul_flt8
897	cash_div_flt8
898	cashlarger

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
899	cashsmaller
919	flt8_mul_cash
935	cash_words
936	substring
937	substring
940	mod
941	mod
942	int28mi
943	int28mul
944	char
945	int8mod
947	mod
948	int28div
949	hashint8
1026	timezone
1044	bpcharin
1045	bpcharout
1046	varcharin
1047	varcharout
1048	bpchareq
1049	bpcharlt
1050	bpcharle
1051	bpchargt
1052	bpcharge
1053	bpcharne
1063	bpchar_larger
1064	bpchar_smaller
1078	bpcharcmp
1080	hashbpchar
1102	time_lt

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1103	time_le
1104	time_gt
1105	time_ge
1106	time_ne
1107	time_cmp
1116	regexp_replace
1117	regexp_replace
1144	time_out
1145	time_eq
1152	timestamptz_eq
1153	timestamptz_ne
1154	timestamptz_lt
1155	timestamptz_le
1156	timestamptz_ge
1157	timestamptz_gt
1158	to_timestamp
1159	timezone
1162	interval_eq
1163	interval_ne
1164	interval_lt
1165	interval_le
1166	interval_ge
1167	interval_gt
1168	interval_um
1169	interval_pl
1170	interval_mi
1172	date_part
1173	timestamptz
1177	interval
1180	abstime

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1188	timestamptz_mi
1194	reltime
1195	timestamptz_smaller
1196	timestamptz_larger
1197	interval_smaller
1198	interval_larger
1199	age
1200	interval
1218	date_trunc
1219	int8inc
1230	int8abs
1236	int8larger
1237	int8smaller
1238	texticregexeq
1239	texticregexne
1246	charlt
1251	int4abs
1253	int2abs
1254	textregexeq
1256	textregexne
1271	overlaps
1273	date_part
1274	int84pl
1275	int84mi
1276	int84mul
1277	int84div
1278	int48pl
1279	int48mi
1280	int48mul
1281	int48div

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1282	quote_ident
1283	quote_literal
1289	quote_nullable
1299	now
1304	overlaps
1308	overlaps
1309	overlaps
1310	overlaps
1311	overlaps
1314	timestamptz_cmp
1315	interval_cmp
1316	time
1326	interval_div
1337	interval
1342	round
1343	trunc
1352	timetz_eq
1353	timetz_ne
1354	timetz_lt
1355	timetz_le
1356	timetz_ge
1357	timetz_gt
1358	timetz_cmp
1359	timestamptz
1360	space
1370	interval
1373	isfinite
1374	octet_length
1375	octet_length
1377	time_larger

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1378	time_smaller
1379	timetz_larger
1380	timetz_smaller
1384	date_part
1385	date_part
1389	isfinite
1390	isfinite
1394	abs
1395	abs
1396	abs
1397	abs
1398	abs
1419	time
1481	tinterval
1564	bit_in
1565	bit_out
1581	biteq
1582	bitne
1592	bitge
1593	bitgt
1594	bitle
1595	bitlt
1596	bitcmp
1608	degrees
1618	interval_mul
1620	ascii
1621	chr
1622	repeat
1623	similar_escape
1624	mul_d_interval

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1633	texticlike
1634	texticnlike
1656	bpcharicregexeq
1657	bpcharicregexne
1658	bpcharregexeq
1659	bpcharregexne
1660	bpchariclike
1661	bpcharicnlike
1666	varbiteq
1667	varbitne
1668	varbitge
1669	varbitgt
1670	varbitle
1671	varbitlt
1672	varbitcmp
1673	bitand
1674	bitor
1675	bitxor
1676	bitnot
1677	bitshiftleft
1678	bitshiftright
1679	bitcat
1682	octet_length
1683	bit
1684	int4
1685	bit
1687	varbit
1688	time_hash
1690	time_mi_time
1691	boolle

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1692	boolge
1693	btboolcmp
1696	timetz_hash
1697	interval_hash
1698	position
1701	numeric_in
1702	numeric_out
1703	numeric
1704	numeric_abs
1705	abs
1706	sign
1707	round
1709	trunc
1710	trunc
1711	ceil
1712	floor
1718	numeric_eq
1719	numeric_ne
1720	numeric_gt
1721	numeric_ge
1722	numeric_lt
1723	numeric_le
1724	numeric_add
1725	numeric_sub
1726	numeric_mul
1727	numeric_div
1728	mod
1729	numeric_mod
1740	numeric
1742	numeric

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1743	numeric
1744	int4
1745	float4
1746	float8
1747	time_pl_interval
1748	time_mi_interval
1749	timetz_pl_interval
1750	timetz_mi_interval
1752	trunc
1753	trunc
1764	numeric_inc
1766	numeric_smaller
1767	numeric_larger
1769	numeric_cmp
1771	numeric_uminus
1779	numeric_int8
1781	numeric
1782	numeric
1783	int2
1810	bit_length
1811	bit_length
1812	bit_length
1840	int2_sum
1841	int4_sum
1842	int8_sum
1845	to_ascii
1846	to_ascii
1848	interval_pl_time
1850	int28eq
1851	int28ne

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1852	int28lt
1853	int28gt
1854	int28le
1855	int28ge
1856	int82eq
1857	int82ne
1858	int82lt
1859	int82gt
1860	int82le
1861	int82ge
1874	btint161cmp
1875	btint162cmp
1876	btint164cmp
1877	btint168cmp
1878	btnumericint1cmp
1879	btnumericint2cmp
1880	btnumericint4cmp
1881	btnumericint8cmp
1882	btint16cmp
1892	int2and
1893	int2or
1894	int2xor
1895	int2not
1896	int2shl
1897	int2shr
1898	int4and
1899	int4or
1900	int4xor
1901	int4not
1902	int4shl

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
1903	int4shr
1904	int8and
1905	int8or
1906	int8xor
1907	int8not
1908	int8shl
1909	int8shr
1910	int8up
1911	int2up
1912	int4up
1913	float4up
1914	float8up
1915	numeric_uplus
1946	encode
1961	timestamp
1967	timestamptz
1968	time
1969	timetz
1973	div
1980	numeric_div_trunc
2012	substring
2013	substring
2014	position
2020	date_trunc
2021	date_part
2024	timestamp
2025	timestamp
2029	date
2031	timestamp_mi
2032	timestamp_pl_interval

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
2033	timestamp_mi_interval
2035	timestamp_smaller
2036	timestamp_larger
2038	timezone
2039	timestamp_hash
2041	overlaps
2042	overlaps
2043	overlaps
2044	overlaps
2045	timestamp_cmp
2046	time
2048	isfinite
2052	timestamp_eq
2053	timestamp_ne
2054	timestamp_lt
2055	timestamp_le
2056	timestamp_ge
2057	timestamp_gt
2058	age
2069	timezone
2070	timezone
2073	substring
2075	bit
2076	int8
2089	to_hex
2090	to_hex
2160	text_pattern_lt
2161	text_pattern_le
2163	text_pattern_ge
2164	text_pattern_gt

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
2166	bttext_pattern_cmp
2167	ceiling
2174	bpchar_pattern_lt
2175	bpchar_pattern_le
2177	bpchar_pattern_ge
2178	bpchar_pattern_gt
2180	btbpchar_pattern_cmp
2188	btint48cmp
2189	btint84cmp
2190	btint24cmp
2191	btint42cmp
2192	btint28cmp
2193	btint82cmp
2194	btfloat48cmp
2195	btfloat84cmp
2308	ceil
2309	floor
2310	sign
2320	ceiling
2338	date_timestamp_lt_mysql
2339	date_le_timestamp
2340	date_eq_timestamp
2341	date_gt_timestamp
2342	date_ge_timestamp
2343	date_ne_timestamp
2364	timestamp_date_lt_mysql
2365	timestamp_le_date
2366	timestamp_eq_date
2367	timestamp_gt_date
2368	timestamp_ge_date

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
2369	timestamp_ne_date
2424	float4recv
2425	float4send
2426	float8recv
2456	bit_recv
2457	bit_send
2460	numeric_recv
2461	numeric_send
2470	time_recv
2515	booland_statefunc
2516	boolor_statefunc
2522	timestamp_eq_timestamptz
2529	timestamptz_eq_timestamp
2547	interval_pl_timetz
2548	interval_pl_timestamp
2557	bool
2558	int4
2765	regexp_split_to_table
2766	regexp_split_to_table
2805	int8inc_float8_float8
2906	timestamptypmodout
2908	timestamptztypmodout
2910	timetypmodout
2912	timetztypmodout
2918	numerictypmodout
2919	bittypmodin
2920	bittypmodout
2971	text
2996	int8_sum_to_int8
3032	get_bit

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
3033	set_bit
3062	reverse
3167	instr
3168	instr
3169	instr
3170	multiply
3171	multiply
3175	lengthb
3176	lengthb
3177	int8_bool
3178	bool_int8
3180	int2_bool
3181	bool_int2
3182	substring_inner
3183	substring_inner
3192	int4_bpchar
3226	timestamp_diff
3227	timestamp_diff
3343	int8_mul_cash
3344	cash_mul_int8
3345	cash_div_int8
3360	addtime
3361	subtime
3616	tsvector_lt
3617	tsvector_le
3618	tsvector_eq
3619	tsvector_ne
3620	tsvector_ge
3621	tsvector_gt
3625	tsvector_concat

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
3634	ts_match_vq
3635	ts_match_qv
3662	tsquery_lt
3663	tsquery_le
3664	tsquery_eq
3665	tsquery_ne
3666	tsquery_ge
3667	tsquery_gt
3669	tsquery_and
3670	tsquery_or
3671	tsquery_not
3691	tsq_mcontains
3692	tsq_mcontained
3822	cash_div_cash
3922	int4range_subdiff
3923	int8range_subdiff
3924	numrange_subdiff
3925	daterange_subdiff
3929	tsrange_subdiff
3930	tstzrange_subdiff
3951	pg_systimestamp
4035	rawin
4036	rawout
4065	int1_varchar
4067	int1_bpchar
4068	int2_bpchar
4069	int8_bpchar
4070	float4_bpchar
4071	float8_bpchar
4072	numeric_bpchar

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
4073	text_timestamp
4098	group_concat_finalfn_mysql
4099	group_concat_transfn_mysql
4159	date_text
4160	date_bpchar
4161	date_varchar
4162	varchar_date
4163	bpchar_date
4164	text_date
4165	int1_text
4166	int2_text
4167	int4_text
4168	int8_text
4169	float4_text
4170	float8_text
4171	numeric_text
4172	bpchar_numeric
4173	varchar_numeric
4178	timestamp_text
4179	timestamp_varchar
4180	int2_varchar
4181	int4_varchar
4182	int8_varchar
4184	float4_varchar
4185	float8_varchar
4186	varchar_timestamp
4187	bpchar_timestamp
4194	text_numeric
4471	strcmp
4490	texticregexeq_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
4491	texticregexne_mysql
4609	btequalimage
4835	datea_cmp
4915	timestamptz_datea
4916	timestamp_datea
4918	datea_timestamp
5046	int2_int8_equal_mysql
5047	int8_int4_equal_mysql
5048	int4_int8_equal_mysql
5049	int2_int2_equal_mysql
5050	int4_int4_equal_mysql
5051	int2_int4_equal_mysql
5052	int4_int2_equal_mysql
5053	float4equal_mysql
5054	float48equal_mysql
5055	float84equal_mysql
5056	bpchar_equal_mysql
5080	float4_bit_mysql
5081	float8_bit_mysql
5082	numeric_bit_mysql
5083	cast
5084	cast
5107	double_bool_m
5108	datetime_bool_m
5109	timestamp_bool_m
5110	date_bool_m
5111	year_bool_m
5112	time_bool_m
5113	mediumint_bool_m
5114	uint1_bool_m

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
5115	uint2_bool_m
5116	uint3_bool_m
5117	uint4_bool_m
5118	uint8_bool_m
5119	text_bool_m
5120	blob_bool_m
5121	binary_bool_m
5519	int1cmp
5520	hashint1
5521	int1_numeric
5522	numeric_int1
5523	i1toi2
5524	i2toi1
5525	i1toi4
5526	i4toi1
5527	i1toi8
5528	i8toi1
5529	i1tof4
5530	f4toi1
5531	i1tof8
5532	f8toi1
5541	int1in
5542	int1out
5543	int1recv
5544	int1send
5561	real_bool_m
5577	bit_in_for_copy
5580	smalldatetime_eq
5581	smalldatetime_ne
5582	smalldatetime_lt

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
5583	smalldatetime_le
5584	smalldatetime_ge
5585	smalldatetime_gt
5586	smalldatetime_cmp
5587	smalldatetime_hash
5588	bit_bool_m
5809	b_db_last_day
5810	b_db_last_day
5811	b_db_last_day
5816	b_db_last_day
5844	bool_bit_mysql
5940	datea_eq
5945	datea_ge
5950	datea_gt
5955	datea_le
5960	datea_lt
5965	datea_neq
5970	datea_pl_interval
5971	interval_pl_datea
5972	datea_mi_interval
5973	datea_mi
5974	datea_mi_timestamp
5975	timestamp_mi_datea
6041	year_in_mysql
6042	year_out_mysql
6043	year_recv_mysql
6044	year_send_mysql
6045	yeartypmodin_mysql
6046	yeartypmodout_mysql
6069	date_year_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6070	time_year_mysql
6071	timestamp_year_mysql
6118	mediumtext_equal_mysql
6135	now_b
6230	float4_row_text_mysql
6231	float4_row_blob_mysql
6282	datea_larger
6283	datea_smaller
6294	datea_hash
6296	trunc
6308	date_part
6407	int16
6408	int2
6409	int16
6410	int4
6411	int16
6412	int8
6413	int16
6414	float8
6415	int16
6416	float4
6419	int16
6420	int16_bool
6421	int16
6422	numeric
6423	int16eq
6424	int16ne
6425	int16lt
6426	int16le
6427	int16gt

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6428	int16ge
6429	int16pl
6430	int16mi
6431	int16mul
6432	int16div
6433	numeric
6434	numeric_bool
6435	uint_int3_gt_mysql
6436	uint_int3_le_mysql
6437	int3_uint_eq_mysql
6438	int21gt
6439	int21le
6440	int21ge
6441	int216eq
6442	int216ne
6443	int216lt
6444	int216gt
6445	int216le
6446	int216ge
6447	int2numericeq
6448	int2numericne
6449	int2numericlt
6450	int2numericgt
6451	int2numericle
6452	int2numericge
6453	int41eq
6454	int41ne
6455	int41lt
6456	int41gt
6457	int41le

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6458	int41ge
6459	int416eq
6460	int416ne
6461	int416lt
6462	int416gt
6463	int416le
6464	int416ge
6465	int4numericeq
6466	int4numericne
6467	int4numericlt
6468	int4numericgt
6469	int4numericle
6470	int4numericge
6471	int81eq
6472	int81ne
6473	int81lt
6474	int81gt
6475	int81le
6476	int81ge
6477	int816eq
6478	int816ne
6479	int816lt
6480	int816gt
6481	int816le
6482	int816ge
6483	int8numericeq
6484	int8numericne
6485	int8numericlt
6486	int8numericgt
6487	int8numericle

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6488	int8numericge
6539	int21eq
6540	int21ne
6551	tinytext_send_mysql
6577	varbinary_equal_mysql
6578	b_timestampdiff
6579	b_timestampdiff
6582	b_timestampdiff
6583	b_timestampdiff
6584	b_timestampdiff
6585	b_timestampdiff
6586	b_timestampdiff
6587	b_timestampdiff
6588	b_timestampdiff
6589	b_timestampdiff
6590	b_timestampdiff
6591	b_timestampdiff
6592	b_timestampdiff
6593	b_timestampdiff
6594	b_timestampdiff
6595	b_timestampdiff
6599	blobtypmodout
6601	texttypmodout
6603	i1typmodout_mysql
6605	i2typmodout_mysql
6607	i4typmodout_mysql
6609	i8typmodout_mysql
6610	plus_any
6611	minus_any
6612	mul_any

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6613	div_any
6614	mod_any
6615	neg_any
6616	logical_oper_bool
6617	xor_bool
6618	bit_or_any
6619	bit_and_any
6620	bit_xor_any
6621	bit_neg_any
6622	bit_shift_left_any
6623	bit_shift_right_any
6624	like_any
6625	nlike_any
6626	int_div_any
6627	tinytext_equal_mysql
6628	binary_equal_mysql
6629	tinyblob_equal_mysql
6630	cast_to_int8
6631	cast_to_float8
6632	cast_to_numeric
6633	cast_to_cstring
6634	cast_to_datetime
6635	int21lt
6636	cast_to_binary
6637	coalesce
6640	makedate
6641	maketime
6643	float4_typmod_out_mysql
6644	float4_mysql
6645	float4_int3_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6646	float4_uint1_mysql
6647	float4_uint2_mysql
6648	float4_uint3_mysql
6649	float4_uint4_mysql
6650	float4_uint8_mysql
6652	float8_typmod_out_mysql
6653	float8_mysql
6654	float8_int3_mysql
6655	float8_uint1_mysql
6656	float8_uint2_mysql
6657	float8_uint3_mysql
6658	float8_uint4_mysql
6659	float8_uint8_mysql
6661	float8_timestamp_mysql
6662	float8_datetime_mysql
6663	float8_date_mysql
6664	float8_time_mysql
6665	text_binary_mysql
6671	text_datetime_mysql
6672	text_int8_mysql
6673	text_int4_mysql
6674	text_int2_mysql
6675	text_int1_mysql
6676	text_uint8_mysql
6677	text_uint4_mysql
6678	text_uint2_mysql
6679	text_uint1_mysql
6680	text_varchar_mysql
6681	text_text_mysql
6682	text_tinytext_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6683	text_mediumtext_mysql
6684	text_longtext_mysql
6685	text_tinyblob_mysql
6686	text_blob_mysql
6687	text_mediumblob_mysql
6688	text_longblob_mysql
6689	text_varbinary_mysql
6690	text_year_mysql
6691	text_time_mysql
6692	text_float4_mysql
6693	text_float8_mysql
6694	text_mediumint_mysql
6699	numeric_blob_mysql
6704	numeric_uint1_mysql
6705	numeric_uint2_mysql
6706	numeric_uint4_mysql
6707	numeric_uint8_mysql
6708	numeric_mediumint_mysql
6709	numeric_year_mysql
6710	numeric_timestamp_mysql
6711	numeric_datetime_mysql
6712	numeric_time_mysql
6713	numeric_date_mysql
6714	i8toi3_mysql
6715	i8toui1_mysql
6716	i8toui2_mysql
6717	i8toui4_mysql
6718	i8toui8_mysql
6719	int4_year_mysql
6720	i8tobinary_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6721	i8tovarbinary_mysql
6725	timestamptz_datetime_mysql
6727	i8toblob_mysql
6728	convert_tz
6729	int8_time_mysql
6730	int8_date_mysql
6731	int8_datetime_mysql
6732	int8_timestamp_mysql
6733	bit_tinyint_mysql
6734	bit_smallint_mysql
6735	bit_mediumint_mysql
6736	bit_uint1_mysql
6737	bit_uint2_mysql
6738	bit_uint4_mysql
6739	bit_uint8_mysql
6740	bit_year_mysql
6741	bit_float_mysql
6742	bit_double_mysql
6743	bit_numeric_mysql
6746	bit_text_mysql
6748	bit_char_mysql
6749	bit_varchar_mysql
6750	bit_binary_mysql
6751	bit_varbinary_mysql
6754	bit_blob_mysql
6756	bit_date_mysql
6757	bit_datetime_mysql
6758	bit_time_mysql
6759	bit_timestamp_mysql
6760	timestamp_int8_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6761	timestamp_float4_mysql
6762	timestamp_float8_mysql
6763	timestamp_numeric_mysql
6764	timestamp_datetime_mysql
6765	timestamp_uint8_mysql
6766	timestamp_bpchar_mysql
6767	datetime_float4_mysql
6768	datetime_float8_mysql
6769	datetime_numeric_mysql
6770	datetime_time_mysql
6771	datetime_timestamp_mysql
6772	datetime_date_mysql
6773	date_float4_mysql
6774	date_float8_mysql
6775	date_numeric_mysql
6776	date_datetime_mysql
6777	date_date_mysql
6778	date_time_mysql
6779	time_float4_mysql
6780	time_float8_mysql
6781	time_numeric_mysql
6782	time_date_mysql
6783	time_datetime_mysql
6784	time_timestamp_mysql
6785	datetime_int8_mysql
6786	date_int8_mysql
6787	time_int8_mysql
6788	time_uint4_mysql
6789	time_uint8_mysql
6790	time_int4_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6791	datetime_uint8_mysql
6792	date_uint8_mysql
6793	timestamp_binary_mysql
6794	timestamp_varbinary_mysql
6795	datetime_bpchar_mysql
6796	datetime_varchar_mysql
6797	datetime_binary_mysql
6798	datetime_varbinary_mysql
6799	datetime_text_mysql
6800	date_binary_mysql
6801	date_varbinary_mysql
6802	time_bpchar_mysql
6803	time_varchar_mysql
6804	time_binary_mysql
6805	time_varbinary_mysql
6806	time_text_mysql
6810	binary_send_mysql
6811	binary_recv_mysql
6812	binary_typmodin_mysql
6813	binary_typmodout_mysql
6814	int12eq
6815	numericint1eq
6816	binary_float_mysql
6817	binary_int4_mysql
6818	mediumint_equal_mysql
6822	binary_datetime_mysql
6823	binary_char_mysql
6824	binary_text_mysql
6825	binary_numeric_mysql
6826	binary_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
6827	blob_recv_mysql
6828	blob_send_mysql
6845	longblob_recv_mysql
6846	longblob_send_mysql
6853	int168ge
6859	longtext_recv_mysql
6860	longtext_send_mysql
6871	mediumblob_recv_mysql
6872	mediumblob_send_mysql
6939	rowid_in
6940	rowid_out
6942	rowid_send
6945	mediumtext_recv_mysql
6946	mediumtext_send_mysql
6954	rowid_recv
6956	chartorowid
6959	tinyblob_recv_mysql
6960	tinyblob_send_mysql
6967	chartorowid
6969	datetime_timestamp_equal_mysql
6973	datetime_date_equal_mysql
6983	mediumblob_equal_mysql
6985	year_equal_mysql
6987	int1_equal_mysql
6993	bit_equal_mysql
6995	timestamp_datetime_equal_mysql
6999	timestamp_equal_mysql
7030	tinytext_recv_mysql
7039	year_bit_mysql
7044	uint1_send_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7045	uint1_recv_mysql
7050	uint2_send_mysql
7051	uint2_recv_mysql
7056	uint4_send_mysql
7057	uint4_recv_mysql
7062	uint8_send_mysql
7063	uint8_recv_mysql
7076	date_timestamp_equal_mysql
7093	date_datetime_equal_mysql
7095	str_to_date
7098	date_equal_mysql
7104	timestamp_date_equal_mysql
7121	longblob_equal_mysql
7136	blob_equal_mysql
7161	bool_char_mysql
7162	bool_binary_mysql
7163	year_timestamp_mysql
7164	year_time_mysql
7165	int8_int3_cmp_mysql
7168	int3_int8_cmp_mysql
7176	int4_int3_cmp_mysql
7181	varbinary_send_mysql
7182	varbinary_recv_mysql
7183	varbinary_typmodin_mysql
7184	varbinary_typmodout_mysql
7185	chartorowid
7186	varbinary_float_mysql
7187	varbinary_int_mysql
7188	varbinary_datetime_mysql
7189	varbinary_char_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7190	varbinary_text_mysql
7193	datetime_recv_mysql
7194	datetime_send_mysql
7195	datetimetypmodin_mysql
7196	datetimetypmodout_mysql
7198	datetime_tinyblob_mysql
7199	datetime_mediumblob_mysql
7200	datetime_blob_mysql
7201	datetime_longblob_mysql
7202	datetime_tinytext_mysql
7203	datetime_mediumtext_mysql
7204	datetime_longtext_mysql
7206	datetime_pl_interval
7207	interval_pl_datetime
7208	datetime_mi
7209	datetime_mi_interval
7212	time_tinyblob_mysql
7213	time_mediumblob_mysql
7214	time_blob_mysql
7215	time_longblob_mysql
7216	time_tinytext_mysql
7217	time_mediumtext_mysql
7218	time_longtext_mysql
7219	varchar_time_mysql
7220	bpchar_time_mysql
7222	timestamp_tinyblob_mysql
7223	timestamp_mediumblob_mysql
7224	timestamp_blob_mysql
7225	timestamp_longblob_mysql
7226	timestamp_tinytext_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7227	timestamp_mediumtext_mysql
7228	timestamp_longtext_mysql
7229	date_tinyblob_mysql
7230	date_mediumblob_mysql
7231	date_blob_mysql
7232	int3_int4_cmp_mysql
7235	date_longblob_mysql
7236	date_tinytext_mysql
7237	date_mediumtext_mysql
7238	date_longtext_mysql
7245	year_uint2_mysql
7246	year_uint4_mysql
7247	year_uint8_mysql
7248	year_int2_mysql
7249	year_int4_mysql
7250	year_int8_mysql
7251	year_float4_mysql
7252	year_float8_mysql
7253	year_numeric_mysql
7254	year_binary_mysql
7255	year_varbinary_mysql
7256	chartorowid
7257	rowidtobpchar
7258	year_blob_mysql
7259	rowidtochar
7260	rowidtonchar
7261	rowidtotext
7262	year_text_mysql
7264	year_varchar_mysql
7265	year_bpchar_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7266	float4_year_mysql
7267	float8_year_mysql
7268	varchar_year_mysql
7269	bpchar_year_mysql
7270	rowid_nodeid
7275	varbinary_uint8_mysql
7276	binary_uint8_mysql
7277	day
7278	month
7279	year
7280	dayofyear
7281	weekofyear
7282	quarter
7283	hour
7284	minute
7285	second
7286	microsecond
7287	to_days
7288	datediff
7289	lpad
7290	rpad
7291	left
7292	right
7293	acos
7294	asin
7295	length
7296	bit_length
7297	char_length
7298	character_length
7299	is_true

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7300	is_false
7301	is_not_true
7302	is_not_false
7303	cos
7304	week
7305	week
7306	yearweek
7307	yearweek
7308	rowid_hashbucketid
7309	degrees
7310	exp
7311	floor
7312	greatest
7313	least
7314	dayofmonth
7315	last_day
7316	weekday
7317	from_unixtime
7318	from_unixtime
7319	timediff
7320	ceiling
7321	sec_to_time
7322	rowid_tableoid
7323	repeat
7324	sin
7325	rowid_sequence
7327	dayname
7328	monthname
7329	date_format
7330	ascii

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7331	sqrt
7332	sign
7333	hex
7334	unhex
7335	substr
7336	substr
7337	truncate
7338	lower
7339	lcase
7340	upper
7341	ucase
7342	atan
7343	atan2
7344	tan
7348	replace
7349	sha
7350	sha1
7351	sha2
7352	pow
7353	power
7354	ltrim
7355	rtrim
7356	trim
7357	trim
7358	period_add
7359	period_diff
7360	get_format
7364	dayofweek
7365	find_in_set
7366	time_format

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7367	substring_index
7368	varbinary_mysql
7369	bit_or
7370	bit_and
7371	bit_xor
7372	year_uint1_mysql
7373	year_int3_mysql
7374	ui1toi3_mysql
7375	ui2toi3_mysql
7376	ui4toi3_mysql
7377	ui8toi3_mysql
7378	bit_func_char_mysql
7379	bit_func_varchar_mysql
7380	bit_func_binary_mysql
7381	bit_func_varbinary_mysql
7390	uint8_bit_mysql
7392	to_seconds
7393	time_to_sec
7394	extract
7395	abs
7396	datetime_scale_mysql
7397	timestampdiff
7398	concat_ws
7400	reverse
7401	eq_datetime
7402	eq_binary
7403	ne_datetime
7404	ne_binary
7405	equal_int
7406	equal_numeric

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7407	equal_float
7408	equal_string
7409	equal_datetime
7410	equal_time
7411	equal_binary
7412	ge_datetime
7413	ge_binary
7414	gt_datetime
7415	gt_binary
7416	le_datetime
7417	le_binary
7418	lt_datetime
7419	lt_binary
7420	text_bit_mysql
7421	blob_bit_mysql
7422	bool_int1_mysql
7423	bool_int2_mysql
7424	bool_mediumint_mysql
7425	bool_int4_mysql
7426	bool_int8_mysql
7427	bool_uint1_mysql
7428	int2_int3_cmp_mysql
7430	bool_uint2_mysql
7431	bool_uint4_mysql
7432	bool_uint8_mysql
7433	bool_real_mysql
7434	bool_double_mysql
7435	bool_numeric_mysql
7436	booltext_mysql
7437	uint_cmp_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7438	uint_sortsupport_mysql
7439	hash_uint8_mysql
7440	bool_bool_equal_mysql
7441	int8_int2_equal_mysql
7442	int3_int2_cmp_mysql
7443	year_cmp_mysql
7444	year_sortsupport
7445	datetime_cmp_mysql
7446	datetime_sortsupport
7447	datetime_cmp_date
7448	datetime_cmp_timestamp
7449	datetime_cmp_time
7450	date_cmp_datetime
7451	timestamp_cmp_datetime
7452	time_cmp_datetime
7453	datetime_cmp_timestamp
7454	datetime_cmp_date
7455	datetime_cmp_time
7456	date_cmp_datetime
7457	time_cmp_datetime
7458	timestamp_cmp_datetime
7459	sum
7460	sum_any
7461	avg
7462	avg_any
7463	avg_final
7464	max
7465	max_any
7466	min
7467	min_any

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7468	sum_final
7469	avg_collect
7470	int1_int16_mysql
7471	int3_int16_mysql
7472	uint1_int16_mysql
7473	uint2_int16_mysql
7474	uint3_int16_mysql
7475	uint4_int16_mysql
7476	uint8_int16_mysql
7477	int16_int1_mysql
7478	int16_int3_mysql
7479	int16_uint1_mysql
7480	int16_uint2_mysql
7481	int16_uint3_mysql
7482	int16_uint4_mysql
7483	int16_uint8_mysql
7484	mediumint_eq_mysql
7485	mediumint_ne_mysql
7486	mediumint_lt_mysql
7487	mediumint_le_mysql
7488	mediumint_gt_mysql
7489	mediumint_ge_mysql
7501	date_int1_mysql
7502	date_int2_mysql
7503	date_int3_mysql
7504	date_int4_mysql
7505	datetime_int1_mysql
7506	datetime_int2_mysql
7507	datetime_int3_mysql
7508	datetime_int4_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7509	timestamp_int1_mysql
7510	timestamp_int2_mysql
7511	timestamp_int3_mysql
7512	timestamp_int4_mysql
7514	binary_eq_mysql
7515	varbinary_eq_mysql
7516	tinyblob_eq_mysql
7517	mediumblob_eq_mysql
7518	longblob_eq_mysql
7519	binary_ne_mysql
7520	varbinary_ne_mysql
7521	tinyblob_ne_mysql
7522	mediumblob_ne_mysql
7523	longblob_ne_mysql
7524	binary_lt_mysql
7525	varbinary_lt_mysql
7526	tinyblob_lt_mysql
7527	mediumblob_lt_mysql
7528	longblob_lt_mysql
7529	binary_le_mysql
7530	varbinary_le_mysql
7531	tinyblob_le_mysql
7532	mediumblob_le_mysql
7533	longblob_le_mysql
7534	binary_gt_mysql
7535	varbinary_gt_mysql
7536	tinyblob_gt_mysql
7537	mediumblob_gt_mysql
7538	longblob_gt_mysql
7539	binary_ge_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7540	varbinary_ge_mysql
7541	tinyblob_ge_mysql
7542	mediumblob_ge_mysql
7543	longblob_ge_mysql
7544	tinytext_eq_mysql
7545	mediumtext_eq_mysql
7546	longtext_eq_mysql
7547	tinytext_ne_mysql
7548	mediumtext_ne_mysql
7549	longtext_ne_mysql
7550	tinytext_lt_mysql
7551	mediumtext_lt_mysql
7552	longtext_lt_mysql
7553	tinytext_le_mysql
7554	mediumtext_le_mysql
7555	longtext_le_mysql
7556	tinytext_gt_mysql
7557	mediumtext_gt_mysql
7558	longtext_gt_mysql
7559	tinytext_ge_mysql
7560	mediumtext_ge_mysql
7561	longtext_ge_mysql
7562	date_literal_mysql
7563	time_literal_mysql
7564	datetime_literal_mysql
7568	year_eq_mysql
7569	year_ne_mysql
7570	year_lt_mysql
7571	year_le_mysql
7572	year_gt_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7573	year_ge_mysql
7574	time
7575	date
7576	timestamp
7577	timestamp
7582	uint3_send_mysql
7583	uint3_recv_mysql
7600	i8toui3_mysql
7605	numeric_uint3_mysql
7606	bit_uint3_mysql
7607	bool_uint3_mysql
7608	year_uint3_mysql
7609	text_uint3_mysql
7610	time_bit_mysql
7611	date_bit_mysql
7612	datetime_bit_mysql
7613	timestamp_bit_mysql
7614	date_uint1_mysql
7615	date_uint2_mysql
7616	date_uint3_mysql
7617	date_uint4_mysql
7618	datetime_uint1_mysql
7619	datetime_uint2_mysql
7620	datetime_uint3_mysql
7621	datetime_uint4_mysql
7622	timestamp_uint1_mysql
7623	timestamp_uint2_mysql
7624	timestamp_uint3_mysql
7625	timestamp_uint4_mysql
7635	ln

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7636	log2
7637	log10
7638	log
7639	log
7640	mysql_compare_int
7641	mysql_compare_numeric
7642	mysql_compare_float
7643	mysql_compare_string
7644	mysql_compare_datetime
7645	mysql_compare_time
7646	mysql_compare_binary
7653	int2_int3_eq_mysql
7654	int2_int3_ne_mysql
7655	int2_int3_equal_mysql
7656	int2_int3_gt_mysql
7657	int2_int3_lt_mysql
7658	int2_int3_ge_mysql
7659	int2_int3_le_mysql
7660	int3_int2_eq_mysql
7661	int3_int2_ne_mysql
7662	int3_int2_equal_mysql
7663	int3_int2_gt_mysql
7664	int3_int2_lt_mysql
7665	int3_int2_ge_mysql
7666	int3_int2_le_mysql
7667	int3_int4_eq_mysql
7668	int3_int4_ne_mysql
7669	int3_int4_equal_mysql
7670	int3_int4_gt_mysql
7671	int3_int4_lt_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7672	int3_int4_ge_mysql
7673	int3_int4_le_mysql
7674	int3_int8_eq_mysql
7675	int3_int8_ne_mysql
7676	int3_int8_equal_mysql
7677	int3_int8_gt_mysql
7678	int3_int8_lt_mysql
7679	int3_int8_ge_mysql
7680	int3_int8_le_mysql
7681	int4_int3_eq_mysql
7682	int4_int3_ne_mysql
7683	int4_int3_equal_mysql
7684	int4_int3_gt_mysql
7685	int4_int3_lt_mysql
7686	int4_int3_ge_mysql
7687	int4_int3_le_mysql
7688	int8_int3_eq_mysql
7689	int8_int3_ne_mysql
7690	int8_int3_equal_mysql
7691	int8_int3_gt_mysql
7692	int8_int3_lt_mysql
7693	int8_int3_ge_mysql
7694	int8_int3_le_mysql
7695	binary_int8_mysql
7696	blob_ne_mysql
7697	blob_lt_mysql
7698	blob_le_mysql
7699	blob_gt_mysql
7700	blob_ge_mysql
7701	blob_eq_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7717	rowid_eq
7718	rowid_ne
7719	rowid_lt
7720	rowid_le
7721	rowid_gt
7722	rowid_ge
7723	rowid_cmp
7724	rowid_hash
7726	if
7727	ifnull
7728	nullif
7729	rowid_larger
7730	rowid_smaller
7737	aes_encrypt
7738	aes_encrypt
7739	aes_encrypt
7740	aes_encrypt
7741	aes_encrypt
7742	aes_decrypt
7743	aes_decrypt
7744	aes_decrypt
7745	aes_decrypt
7746	aes_decrypt
7747	numericint2le
7748	numericint2ge
7749	numericint4eq
7750	numericint4ne
7751	numericint4lt
7752	numericint4gt
7753	numericint4le

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7754	numericint4ge
7755	numericint8eq
7756	numericint8ne
7757	numericint8lt
7758	numericint8gt
7759	numericint8le
7760	numericint8ge
7761	int161eq
7762	int161ne
7763	int161lt
7766	rowid_list_agg_noarg2_transfn
7767	rowid_list_agg_transfn
7768	pivot_
7779	year_datetime_mysql
7788	year_date_mysql
7790	uint_int3_eq_mysql
7791	uint_int3_ne_mysql
7792	uint_int3_equal_mysql
7793	uint_int3_ge_mysql
7797	int3_uint_ne_mysql
7818	datetime_year_mysql
7821	attname_eq_mysql
7822	date_func_int2_mysql
7823	date_func_int3_mysql
7824	date_func_int4_mysql
7825	date_func_int8_mysql
7826	date_func_uint1_mysql
7827	date_func_uint2_mysql
7828	date_func_uint3_mysql
7829	date_func_uint4_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7830	date_func_uint8_mysql
7831	date_func_bit_mysql
7832	date_row_int1_mysql
7833	date_row_int2_mysql
7834	date_row_int3_mysql
7835	date_row_int4_mysql
7836	date_row_int8_mysql
7837	date_row_uint1_mysql
7838	date_row_uint2_mysql
7839	date_row_uint3_mysql
7840	date_row_uint4_mysql
7841	date_row_uint8_mysql
7842	date_row_float4_mysql
7843	date_row_float8_mysql
7844	date_row_numeric_mysql
7845	date_row_bit_mysql
7846	time_func_bit_mysql
7847	time_row_int1_mysql
7848	time_row_int2_mysql
7849	time_row_int3_mysql
7850	time_row_int4_mysql
7851	time_row_int8_mysql
7852	time_row_uint1_mysql
7853	time_row_uint2_mysql
7854	time_row_uint3_mysql
7855	time_row_uint4_mysql
7856	time_row_uint8_mysql
7857	time_row_float4_mysql
7858	time_row_float8_mysql
7859	time_row_numeric_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7860	time_row_bit_mysql
7861	datetime_func_bit_mysql
7862	datetime_row_int1_mysql
7863	datetime_row_int2_mysql
7864	datetime_row_int3_mysql
7865	datetime_row_int4_mysql
7866	datetime_row_int8_mysql
7867	datetime_row_uint1_mysql
7868	datetime_row_uint2_mysql
7869	datetime_row_uint3_mysql
7870	datetime_row_uint4_mysql
7871	datetime_row_uint8_mysql
7872	datetime_row_float4_mysql
7873	datetime_row_float8_mysql
7874	datetime_row_numeric_mysql
7875	datetime_row_bit_mysql
7876	timestamp_row_int1_mysql
7877	timestamp_row_int2_mysql
7878	timestamp_row_int3_mysql
7879	timestamp_row_int4_mysql
7880	timestamp_row_int8_mysql
7897	timestamp_row_uint1_mysql
7898	timestamp_row_uint2_mysql
7899	timestamp_row_uint3_mysql
7900	timestamp_row_uint4_mysql
7901	timestamp_row_uint8_mysql
7902	timestamp_row_float4_mysql
7903	timestamp_row_float8_mysql
7904	timestamp_row_numeric_mysql
7905	timestamp_row_bit_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
7906	date_func_year_mysql
7907	date_row_year_mysql
7908	date_row_time_mysql
7909	time_func_year_mysql
7910	time_row_year_mysql
7911	time_row_date_mysql
7912	time_row_datetime_mysql
7913	time_row_timestamp_mysql
7914	datetime_func_year_mysql
7915	datetime_row_year_mysql
7916	datetime_row_date_mysql
7917	datetime_row_time_mysql
7918	timestamp_row_year_mysql
7919	timestamp_row_date_mysql
7920	timestamp_row_time_mysql
7921	date_func_int1_mysql
8117	uint_int3_lt_mysql
8234	binarycmp_mysql
8300	bpchar_bit_mysql
8523	uint_int8_eq_mysql
8524	uint_int8_ne_mysql
8525	uint_int8_equal_mysql
8526	uint_int8_ge_mysql
8527	uint_int8_gt_mysql
8528	uint_int8_le_mysql
8529	uint_int8_lt_mysql
8530	int8_uint_eq_mysql
8531	int8_uint_ne_mysql
8532	int8_uint_equal_mysql
8533	int8_uint_ge_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
8534	int8_uint_gt_mysql
8535	int8_uint_le_mysql
8536	int8_uint_lt_mysql
8537	uint_int4_eq_mysql
8538	uint_int4_ne_mysql
8539	uint_int4_equal_mysql
8540	uint_int4_ge_mysql
8541	uint_int4_gt_mysql
8542	uint_int4_le_mysql
8543	uint_int4_lt_mysql
8544	int4_uint_eq_mysql
8545	int4_uint_ne_mysql
8546	int4_uint_equal_mysql
8547	int4_uint_ge_mysql
8548	int4_uint_gt_mysql
8549	int4_uint_le_mysql
8550	int4_uint_lt_mysql
8627	uint_int2_eq_mysql
8628	uint_int2_ne_mysql
8629	uint_int2_equal_mysql
8630	uint_int2_ge_mysql
8631	uint_int2_gt_mysql
8632	uint_int2_le_mysql
8633	uint_int2_lt_mysql
8634	int2_uint_eq_mysql
8635	int2_uint_ne_mysql
8636	int2_uint_equal_mysql
8637	int2_uint_ge_mysql
8638	int2_uint_gt_mysql
8639	int2_uint_le_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
8640	int2_uint_lt_mysql
8705	uint_uint_eq_mysql
8706	uint_uint_ne_mysql
8707	uint_uint_equal_mysql
8708	uint_uint_ge_mysql
8709	uint_uint_gt_mysql
8710	uint_uint_le_mysql
8711	uint_uint_lt_mysql
8746	int3_uint_equal_mysql
8747	int3_uint_ge_mysql
8748	int3_uint_gt_mysql
8749	int3_uint_le_mysql
8750	int3_uint_lt_mysql
8751	int161gt
8752	int161le
8753	int161ge
8754	int162eq
8755	int162ne
8756	int162lt
8757	int162gt
8758	int162le
8759	int162ge
8760	int164eq
8761	int164ne
8762	int164lt
8763	int164gt
8764	int164le
8765	int164ge
8766	int168eq
8767	int168ne

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
8768	int168lt
8769	int168gt
8770	int168le
8815	concat
8983	uint_int2_cmp_mysql
8984	uint_int3_cmp_mysql
8985	uint_int4_cmp_mysql
8986	uint_int8_cmp_mysql
8987	int2_uint_cmp_mysql
8988	int3_uint_cmp_mysql
8989	int4_uint_cmp_mysql
8990	int8_uint_cmp_mysql
9000	date_add
9011	smalldatetime_smaller
9012	smalldatetime_larger
9362	hash_meet_partition
9548	mediumblobtypmodout
9555	mediumtexttypmodout
9557	tinytexttypmodout
9558	int12ne
9559	int12lt
9560	int12gt
9561	int12le
9562	int12ge
9563	int14eq
9564	int14ne
9566	int14lt
9567	int14gt
9568	int14le
9569	int14ge

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
9573	int18eq
9574	int18ne
9575	int18lt
9576	int18gt
9584	int18le
9585	int18ge
9586	int116eq
9587	int116ne
9588	int116lt
9589	int116gt
9590	int116le
9591	int116ge
9592	int1numericeq
9593	int1numericne
9594	int1numericlt
9595	int1numericgt
9596	int1numericle
9597	int1numericge
9624	numericint1ne
9625	numericint1lt
9626	numericint1gt
9627	numericint1le
9628	numericint1ge
9629	numericint2eq
9630	numericint2ne
9631	numericint2lt
9632	numericint2gt
9749	time_uint1_mysql
9750	time_int1_mysql
9751	time_uint2_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
9752	time_int2_mysql
9753	time_uint3_mysql
9754	time_int3_mysql
9807	datea_pl_text
9808	datea_mi_text
9809	text_pl_datea
9810	datea_pl_int8
9811	datea_mi_int8
9812	int8_pl_datea
9813	datea_pl_float4
9814	datea_mi_float4
9815	float4_pl_datea
9816	datea_pl_float8
9817	datea_mi_float8
9818	float8_pl_datea
9819	datea_pl_numeric
9820	datea_mi_numeric
9821	numeric_pl_datea
9887	float4_blob_mysql
9888	float4_varbinary_mysql
9889	float4_binary_mysql
9890	float8_blob_mysql
9891	float8_varbinary_mysql
9892	float8_binary_mysql
9910	substring_index
9934	datetime_date_eq_mysql
9935	date_datetime_eq_mysql
9936	datetime_timestamp_eq_mysql
9937	timestamp_datetime_eq_mysql
9938	datetime_date_gt_mysql

版权所有 © 华为云计算技术有限公司

func_oid_value	func_name
9939	date_datetime_gt_mysql
9940	datetime_timestamp_gt_mysql
9941	timestamp_datetime_gt_mysql
9942	datetime_date_lt_mysql
9943	date_datetime_lt_mysql
9944	datetime_timestamp_lt_mysql
9945	timestamp_datetime_lt_mysql
9946	datetime_date_le_mysql
9947	date_datetime_le_mysql
9948	datetime_timestamp_le_mysql
9949	timestamp_datetime_le_mysql
9950	datetime_date_ge_mysql
9951	date_datetime_ge_mysql
9952	datetime_timestamp_ge_mysql
9953	timestamp_datetime_ge_mysql
9954	datetime_date_ne_mysql
9955	date_datetime_ne_mysql
9956	datetime_timestamp_ne_mysql
9957	timestamp_datetime_ne_mysql
9964	tinyblobtypmodout

版权所有 © 华为云计算技术有限公司

2 存储过程

2.1 存储过程

商业规则和业务逻辑可以通过程序存储在GaussDB中，这个程序就是存储过程。

存储过程是SQL、PL/SQL和Java语句的组合。通过存储过程，执行商业规则的代码可以从应用程序层迁移至到数据库层，从而实现代码复用，即代码只需存储一次即可被多个程序调用。

存储过程的创建及调用方法，请参见**CREATE PROCEDURE**。

《开发指南》中“用户自定义函数 > PL/SQL语言函数”中所述的PL/SQL语言创建的函数与存储过程的应用方法相同。下面各节中，除非特别声明，否则内容均适用于存储过程和PL/SQL语言函数。

2.2 数据类型

数据类型是一组值的集合以及定义在这个值集上的一组操作。GaussDB数据库是由表的集合组成的，而各表中的列定义了该表，每一列都属于一种数据类型，GaussDB会根据数据类型提供对应的函数进行操作，例如对数值型数据进行加、减、乘、除等操作。

XML类型数据支持作为存储过程的入参、出参和自定义变量和返回值。支持自治事务的存储过程。

2.3 数据类型转换

数据库系统中，部分数据类型支持隐式类型转换（例如赋值、函数调用的参数等），而部分数据类型禁止隐式数据类型转换（例如INT和复合类型）。可尝试使用GaussDB提供的类型转换函数进行数据类型强转。

GaussDB数据库常见的隐式类型转换如**表2-1**所示。

须知

GaussDB支持的DATE的效限范围是：公元前4713年到公元294276年。

版权所有 © 华为云计算技术有限公司

表 2-1 隐式类型转换表

原始数据类型	目标数据类型	备注
CHAR	VARCHAR2	-
CHAR	NUMBER	原数据必须由数字组成。
CHAR	DATE	原数据不能超出合法日期范围。
CHAR	RAW	-
CHAR	CLOB	-
VARCHAR2	CHAR	-
VARCHAR2	NUMBER	原数据必须由数字组成。
VARCHAR2	DATE	原数据不能超出合法日期范围。
VARCHAR2	CLOB	-
NUMBER	CHAR	-
NUMBER	VARCHAR2	-
DATE	CHAR	-
DATE	VARCHAR2	-
RAW	CHAR	-
RAW	VARCHAR2	-
CLOB	CHAR	-
CLOB	VARCHAR2	-
CLOB	NUMBER	原数据必须由数字组成。
INT4	CHAR	-
DATEA	TEXT	-
DATEA	VARCHAR	-
DATEA	CHAR	-
DATEA	CLOB	-
DATEA	NVARCHAR2	-
DATEA	TIMESTAMP WITH TIME ZONE	-
DATEA	TIMESTAMP WITHOUT TIME ZONE	-
TEXT	DATEA	原数据不能超出合法日期范围，且符合nls_date_format格式。

版权所有 © 华为云计算技术有限公司

原始数据类型	目标数据类型	备注
VARCHAR	DATEA	原数据不能超出合法日期范围，且符合nls_date_format格式。
CHAR	DATEA	原数据不能超出合法日期范围，且符合nls_date_format格式。
CLOB	DATEA	原数据不能超出合法日期范围，且符合nls_date_format格式。
NVARCHAR2	DATEA	原数据不能超出合法日期范围，且符合nls_date_format格式。

2.4 数组、集合和 record

2.4.1 数组

2.4.1.1 数组类型的使用

在使用数组之前，需要自定义一个数组类型。

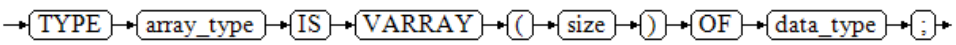
语法

在存储过程中紧跟AS关键字后面定义数组类型。定义方法为：

```
TYPE array_type IS VARRAY(size) OF data_type;
```

语法图如图2-1所示。

图 2-1 数组类型定义语法



其中：

- array_type：要定义的数组类型名。
- size：取值为正整数，表示可以容纳的成员的最大数量。
- data_type：要创建的数组中成员的类型。

注意事项

- GaussDB不仅支持通过索引语法（圆括号）访问数组元素，还提供了多种专用函数（如EXTEND、COUNT、FIRST、LAST、PRIOR、EXISTS、TRIM、NEXT和DELETE）以实现灵活的数组操作与管理。
- 在GaussDB中，数组会自动增长，访问越界会返回一个NULL，不会报错。
- 在存储过程中定义的数组类型，其作用域仅在该存储过程中。
- size仅用于语法兼容，无实际意义。

版权所有 © 华为云计算技术有限公司

- data_type也可以为存储过程中定义的record类型（匿名块不支持）、集合类型，但不可以为存储过程中定义的数组类型、游标类型。
- data_type为集合类型时，不支持使用多维数组。
- array类型的构造器仅支持在ORA兼容模式下使用。
- array类型的构造器不支持作为函数或存储过程参数的默认值。
- 当数组的元素是集合类型并且数组的data_type为VARCHAR、NUMERIC等可以定义长度和精度的类型时，要校验该数组的元素长度或者将元素转换成对应的精度，需要设置GUC参数behavior_compat_options启用tableof_elem_constraints选项。
- 设置enable_recordtype_check_strict参数值为on后，成员是record类型，且record类型有列具有NOT NULL属性或DEFAULT属性，在存储过程或PACKAGE编译时会报错。
- 存储过程中如果有DML语句（SELECT、UPDATE、INSERT和DELETE），DML语句只能使用中括号来访问数组元素，这样可以和函数表达式区分开。
- 不支持NOT NULL语法。
- 不支持在DML语句中访问varray类型变量的元素。
- 自定义数组类型不支持下推，下推场景使用变量承接数组元素 。
- 通过下标访问数组成员时，解析器优先将数组类型变量视为表列名解析，如果表列名有歧义，例如两张表有同名字段，则会引发错误。

示例

```
-- 演示在存储过程中对数组进行操作。
gaussdb=# CREATE OR REPLACE PROCEDURE array_proc AS
DECLARE
    TYPE ARRAY_INTEGER IS VARRAY(1024) OF INTEGER; -- 定义数组类型。
    ARRINT ARRAY_INTEGER := ARRAY_INTEGER(); -- 声明数组类型的变量。
BEGIN
    ARRINT.EXTEND(10);
    FOR I IN 1..10 LOOP
        ARRINT(I) := I;
    END LOOP;
    DBE_OUTPUT.PRINT_LINE(ARRINT.COUNT);
    DBE_OUTPUT.PRINT_LINE(ARRINT(1));
    DBE_OUTPUT.PRINT_LINE(ARRINT(10));
    DBE_OUTPUT.PRINT_LINE(ARRINT(ARRINT.FIRST));
    DBE_OUTPUT.PRINT_LINE(ARRINT(ARRINT.LAST));
    DBE_OUTPUT.PRINT_LINE(ARRINT(ARRINT.NEXT(ARRINT.FIRST)));
    DBE_OUTPUT.PRINT_LINE(ARRINT(ARRINT.PRIOR(ARRINT.LAST)));
    ARRINT.TRIM();
    IF ARRINT.EXISTS(10) THEN
       DBE_OUTPUT.PRINT_LINE('Exist 10th element');
    ELSE
       DBE_OUTPUT.PRINT_LINE('Not exist 10th element');
    END IF;
    DBE_OUTPUT.PRINT_LINE(ARRINT.COUNT);
    DBE_OUTPUT.PRINT_LINE(ARRINT(ARRINT.FIRST));
    DBE_OUTPUT.PRINT_LINE(ARRINT(ARRINT.LAST));
    ARRINT.DELETE();
END;
/

-- 调用该存储过程。
gaussdb=# CALL array_proc();
10
1
10
1
```


版权所有 © 华为云计算技术有限公司

```
10
2
9
Not exist 10th element
9
1
9
 array_proc
------------

(1 row)


-- 删除存储过程。
gaussdb=# DROP PROCEDURE array_proc;
```

2.4.1.2 数组支持的函数

说明

- 以下示例为ORA兼容模式下的示例，函数定义说明里[]中的内容代表可选项，如count[()]表示可以写成count或count()。
- 暂不支持数组未初始化报错（An uninitialized collection is referenced）、数组下标越界报错（Subscript beyond count）、数组下标超限制报错（Subscript outside of limit）。
- 在内层表达式中，不支持通过嵌套的方式调用数组类型函数。
- extend(count)、delete(idx)和trim(n)函数的参数只支持单个常量或变量，不支持其他形式的参数，如函数调用、表达式等形式。
- 当类型嵌套且数组类型作为内部类型时，无参的extend、delete、trim函数只支持extend、delete、trim形式调用，不支持extend()、delete()、trim()形式调用。

- extend[(count)]

  参数：count为int4类型。

  返回值：无返回值。

  功能描述：在varray变量末尾拓展1个或count个元素，元素默认扩展为NULL。

  示例1：extend，扩展1个元素，默认扩展为NULL。

  ```
  gaussdb=# DECLARE
  gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
  gaussdb-#     arrint array_integer := array_integer();
  gaussdb-# BEGIN
  gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
  gaussdb$#     arrint.EXTEND;
  gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
  gaussdb$# END;
  gaussdb$# /
  0
  1
  ANONYMOUS BLOCK EXECUTE
  ```

  示例2：extend(count)，扩展count个元素，默认扩展为NULL。

  ```
  gaussdb=# DECLARE
  gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
  gaussdb-#     arrint array_integer := array_integer();
  gaussdb-# BEGIN
  gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
  gaussdb$#     arrint.EXTEND(3);
  gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
  gaussdb$# END;
  gaussdb$# /
  0
  3
  ANONYMOUS BLOCK EXECUTE
  ```


版权所有 © 华为云计算技术有限公司

- extend(count, idx)

  参数：idx和count为INT4类型。

  返回值：无返回值。

  功能描述：在varray变量末尾拓展count个元素，并且扩展元素的值等于给定idx下标元素的值。

  示例：extend(count, idx)，扩展第idx位置的元素。

```
gaussdb=# SET a_format_version='10c';
SET
gaussdb=# SET a_format_dev_version='s1';
SET
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer := array_integer(1,2);
gaussdb-# BEGIN
gaussdb$#     arrint.EXTEND(2, 1);
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     FOR i IN 1..arrint.COUNT LOOP
gaussdb$#         DBE_OUTPUT.PRINT_LINE(arrint(i));
gaussdb$#     END LOOP;
gaussdb$# END;
gaussdb$# /
4
1
2
1
1
ANONYMOUS BLOCK EXECUTE
```

- count[()]

  参数：无。

  返回值：INT4类型。

  功能描述：返回数组中的元素个数。

  示例：查看已初始化的数组元素个数。

```
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer;
gaussdb-#     len INT;
gaussdb-# BEGIN
gaussdb$#     arrint := array_integer(1, 2, 3);
gaussdb$#     len := arrint.COUNT();
gaussdb$#     DBE_OUTPUT.PRINT_LINE(len);
gaussdb$# END;
gaussdb$# /
3
ANONYMOUS BLOCK EXECUTE
```

- trim[(n)]

  参数：n为INT4类型。

  返回值：无返回值。

  功能描述：无参数时，删除数组末尾一个元素；给定输入参数n时，删除数组末尾指定数量元素。

  示例：删除数组中的n个元素。

```
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer := array_integer(1,2,3);
gaussdb-# BEGIN
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     arrint.TRIM(1);
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
```


版权所有 © 华为云计算技术有限公司

```
gaussdb$# END;
gaussdb$# /
3
2
ANONYMOUS BLOCK EXECUTE
-- n大于数组元素个数, 清空数组元素
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer := array_integer(1,2,3);
gaussdb-# BEGIN
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     arrint.TRIM(4);
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$# END;
gaussdb$# /
3
0
ANONYMOUS BLOCK EXECUTE
```

- delete[()]

  参数：无参数。

  返回值：无返回值。

  功能描述：清空数组中的元素。

  示例：清空数组中的元素。

```
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer := array_integer(1,2,3);
gaussdb-# BEGIN
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     arrint.DELETE();
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$# END;
gaussdb$# /
3
0
ANONYMOUS BLOCK EXECUTE，数组元素均为NULL
-- 数组未初始化
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer;
gaussdb-# BEGIN
gaussdb$#     arrint.DELETE();
gaussdb$# END;
gaussdb$# /
ANONYMOUS BLOCK EXECUTE
```

- delete(idx)

  参数：INT4类型。

  返回值：无返回值。

  功能描述：给定下标idx在数组范围内，则删除数组中给定下标idx的元素。

  示例1：给定idx下标在数组范围内。

```
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer := array_integer(1, 2, 3);
gaussdb-# BEGIN
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     arrint.DELETE(2);
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     FOR i IN 1..arrint.COUNT LOOP
gaussdb$#         DBE_OUTPUT.PRINT_LINE(arrint(i));
gaussdb$#     END LOOP;
gaussdb$# END;
gaussdb$# /
3
```


版权所有 © 华为云计算技术有限公司

```
2
1
3
ANONYMOUS BLOCK EXECUTE
```

示例2：给定idx下标在数组范围外。

```
gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer := array_integer(1, 2, 3);
gaussdb-# BEGIN
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     arrint.DELETE(2);
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     FOR i IN 1..arrint.COUNT LOOP
gaussdb$#         DBE_OUTPUT.PRINT_LINE(arrint(i));
gaussdb$#     END LOOP;
gaussdb$# END;
gaussdb$# /
3
INFO:  {1,2,3}
ANONYMOUS BLOCK EXECUTE

gaussdb=# DECLARE
gaussdb-#     TYPE array_integer IS VARRAY(10) OF INTEGER;
gaussdb-#     arrint array_integer:= array_integer(1, 2, 3);
gaussdb-#BEGIN
gaussdb$#     DBE_OUTPUT.PRINT_LINE(arrint.COUNT);
gaussdb$#     arrint.DELETE(11);
gaussdb$#     RAISE INFO '%', arrint;
gaussdb$# END;
gaussdb$# /
3
INFO:  {1,2,3}
ANONYMOUS BLOCK EXECUTE
```

- first[()]

  参数：无。

  返回值：INT4类型。

  功能描述：返回数组中第一个元素的下标，若没有第一个元素，则返回NULL。

  示例：

```
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc');
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'first is %',v.FIRST();
gaussdb$# END;
gaussdb$# /
INFO:  first is 1
ANONYMOUS BLOCK EXECUTE
-- 数组未初始化返回NULL
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr;
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'first is %',v.FIRST;
gaussdb$# END;
gaussdb$# /
INFO:  first is <NULL>
ANONYMOUS BLOCK EXECUTE
```

- last[()]

  参数：无。

  返回值：INT4类型。

  功能描述：返回数组中最后一个元素的下标。若没有最后一个元素，则返回NULL。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc');
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'last is %',v.LAST();
gaussdb$# END;
gaussdb$# /
INFO:  last is 2
ANONYMOUS BLOCK EXECUTE
-- 数组未初始化返回NULL
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr;
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'last is %',v.LAST;
gaussdb$# END;
gaussdb$# /
INFO:  last is <NULL>
ANONYMOUS BLOCK EXECUTE
```

- prior(idx)

  参数：int4类型。

  返回值：INT4类型。

  功能描述：返回数组中指定下标的前一个元素下标，若无法找到对应元素下标则返回NULL。

  示例：

```
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc');
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'prior is %',v.PRIOR(2);
gaussdb$# END;
gaussdb$# /
INFO:  prior is 1
ANONYMOUS BLOCK EXECUTE
-- 数组未初始化返回NULL
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr;
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'prior is %',v.PRIOR(2);
gaussdb$# END;
gaussdb$# /
INFO:  prior is <NULL>
ANONYMOUS BLOCK EXECUTE
-- 下标越界，大于数组范围
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc','qwe');
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'prior is %',v.PRIOR(10);
gaussdb$# END;
gaussdb$# /
INFO:  prior is 3
ANONYMOUS BLOCK EXECUTE
```

- next(idx)

  参数：int4类型。

  返回值：INT4类型。

  功能描述：返回数组中指定下标的后一个元素下标，若无法找到对应元素下标则返回NULL。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc');
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'next is %',v.NEXT(1);
gaussdb$# END;
gaussdb$# /
INFO:  next is 2
ANONYMOUS BLOCK EXECUTE
-- 数组未初始化返回NULL
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr;
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'next  is %',v.NEXT(1);
gaussdb$# END;
gaussdb$# /
INFO:  next  is <NULL>
ANONYMOUS BLOCK EXECUTE
-- 下标越界，大于数组范围
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc','qwe');
gaussdb-# BEGIN
gaussdb$#     RAISE INFO 'next  is %',v.NEXT(10);
gaussdb$# END;
gaussdb$# /
INFO:  next  is <NULL>
ANONYMOUS BLOCK EXECUTE
```

- exists(idx)

  参数：int4类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：查找指定位置是否存在元素。

  示例：

```
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr := varr('asd','zxc');
gaussdb-#     flag BOOL;
gaussdb-# BEGIN
gaussdb$#     flag := v.EXISTS(1);
gaussdb$#     RAISE INFO '%',flag;
gaussdb$#     flag := v.EXISTS(3);
gaussdb$#     RAISE INFO '%',flag;
gaussdb$#     flag := v.EXISTS(7);
gaussdb$#     RAISE INFO '%',flag;
gaussdb$# END;
gaussdb$# /
INFO:  t
INFO:  f
INFO:  f
ANONYMOUS BLOCK EXECUTE
-- 数组未初始化返回false
gaussdb=# DECLARE
gaussdb-#     TYPE varr IS VARRAY(10) OF VARCHAR(3);
gaussdb-#     v varr;
gaussdb-#     flag BOOL;
gaussdb-# BEGIN
gaussdb$#     flag := v.EXISTS(1);
gaussdb$#     RAISE INFO '%',flag;
gaussdb$#     flag := v.EXISTS(3);
gaussdb$#     RAISE INFO '%',flag;
gaussdb$#     flag := v.EXISTS(7);
gaussdb$#     RAISE INFO '%',flag;
gaussdb$# END;
gaussdb$# /
INFO:  f
```


版权所有 © 华为云计算技术有限公司

```
INFO:  f
INFO:  f
ANONYMOUS BLOCK EXECUTE
```

2.4.2 集合

2.4.2.1 集合类型的使用

在使用集合之前，需要自定义一个集合类型。

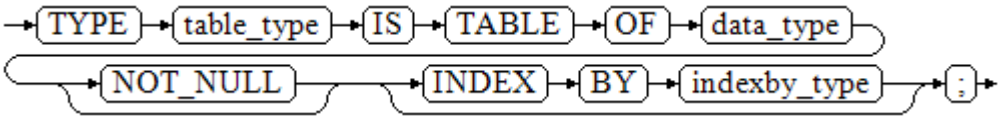
语法

在存储过程中紧跟AS关键字后面定义集合类型。定义方法如下。

```
TYPE table_type IS TABLE OF data_type [ [NOT_NULL] [INDEX BY indexby_type]];
```

语法图如图2-2所示。

图 2-2 集合类型定义语法



其中：

- table_type：要定义的集合类型名。
- data_type：要创建的集合中成员的类型。
- indexby_type：创建集合索引的类型。

无索引的集合类型

- 类型定义

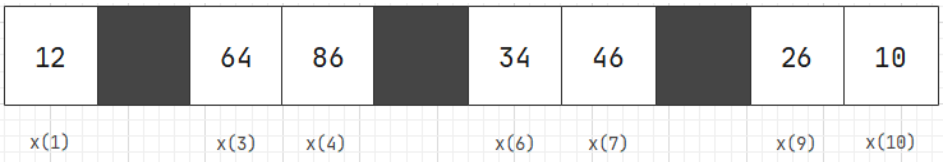
  定义集合类型后，使用table_type作为类型名声明变量。

- 变量声明和初始化

  在变量声明时或者声明后使用类型构造器对变量进行初始化。如未初始化，变量var_name的值为NULL。

  该集合类型将下标和对应成员值以变长数组的方式存储指定数据类型的成员，用户可以通过extend函数扩展存储空间，通过trim函数释放存储空间。以存储空间为10，成员类型为int的集合变量x为例，存储方式如图2-3所示。

  图 2-3 存储方式

  其中成员 x(2)，x(5)，x(8)三个成员是无效的，但是存储空间仍然保留，后续可以继续赋值，而不需要重新分配空间。

版权所有 © 华为云计算技术有限公司

无索引的集合类型作为函数返回值或者自治事务入参时，会删除无效成员的存储空间，所有成员的下标会相应发生改变。

- 变量赋值

  变量声明和初始化后，可通过下标访问集合成员，对成员进行赋值。下标的范围为[1, upper]，upper的值为当前空间的大小。如访问被删除的成员，会返回no data found的错误信息。

- 变量取值

  变量声明和初始化后，可通过下标访问集合成员，获取成员变量的值。下标的范围为[1, upper]，upper的值为当前空间的大小。如访问被删除的成员，会返回no data found的错误信息。

使用约束

- GaussDB支持使用圆括号来访问集合元素，且还支持一些特有的函数，如extend、count、first、last、prior、next以及delete来访问集合的内容。集合函数支持multiset union、intersect、except all和distinct函数。
- 仅支持在兼容ORA模式下（参数sql_compatibility值为ORA）创建集合类型。
- 在GaussDB中，无索引的集合不会自动增长，访问下标越界时会报错。
- 支持在schema、匿名块、存储过程、自定义函数和PACKAGE中定义的集合类型，其作用域各不相同。
- NOT NULL只支持语法不支持功能。
- 当data_type为varchar、numeric等可以定义长度和精度的类型时，要校验集合的元素长度或者将元素转换成对应的精度，需要开启tableof_elem_constraints参数（设置behavior_compat_options参数值为tableof_elem_constraints）。
- data_type为数组类型时，数组类型的元素长度校验或精度转换也受参数tableof_elem_constraints是否开启影响。
- 通过数组类型转换成的集合类型的值不支持对元素长度校验或精度转换。
- data_type可以为基础数据类型、或存储过程内定义的record类型、集合类型和数组类型，不支持ref cursor类型。
- 不同的集合类型的变量不能相互赋值。即使成员类型相同，但集合类型名称不同，也是不同的集合类型。如TYPE t1 IS TABLE OF int;和TYPE t2 IS TABLE OF int;定义的两个集合类型，t1和t2是不同的集合类型，以其定义的变量不支持相互赋值（作为成员类型时该约束不保证生效，赋值逻辑受父类型影响）。
- 只支持集合的等值（=）与非等值（<>或!=）比较，不支持其他关系运算和算数运算操作。
- 集合类型与NULL比较时，请使用 IS [NOT] NULL，使用 = 操作符与NULL比较的结果不准确。
- 支持集合类型变量作为函数的参数和返回值，此时要求参数或者返回值的类型是在schema中定义的集合类型。
- 无索引的集合作为函数入参时，可以传入对应子元素类型相同的数组类型作为入参，不支持多维数组，且要求数组下标从1开始（此功能为陈旧设计，不建议使用。可执行“SET behavior_compat_options = 'disable_rewrite_nesttable';”禁用）。
- 不支持对XML类型数据操作。
- 集合类型以及嵌套集合的类型不支持作为表中的一列来创建表。
- 集合类型的构造器不支持浮点数以及表达式作为下标。

版权所有 © 华为云计算技术有限公司

- 在匿名块中定义的集合类型，匿名块执行ROLLBACK或发生EXCEPTION后，集合类型将无法继续使用。
- 设置GUC参数enable_recordtype_check_strict为on后，若成员是record类型，且record类型有列具有NOT NULL属性或DEFAULT属性，在存储过程或PACKAGE编译时会报错。
- 不支持在DML语句中访问非Schema中定义的集合类型的元素。
- 自定义集合类型不支持下推，下推场景使用变量承接集合元素。
- 通过下标访问集合成员时，解析器优先将集合类型变量视为表列名解析，如果表列名有歧义，例如两张表有同名字段，则会引发错误。

带索引的集合类型

该集合类型将下标和对应成员值以键值对的方式存储在HASH表中，对该类型变量的所有操作实际就是对HASH表的操作。用户无需自行扩展或释放存储空间，仅需通过赋值或delete方式进行存储和删除成员。集合相关操作、说明如下：

1. 类型定义

   索引集合类型定义需同时指定成员类型data_type和下标类型indexby_type，其中下标类型仅支持INTEGER和VARCHAR。

2. 变量声明和初始化

   索引集合类型声明后存在3种初始化场景：未初始化、初始化为空、初始化指定下标和成员值。其中未初始化和初始化为空场景对变量的效果一致。未初始化或初始化为空后变量不为NULL，后续都可以对变量直接进行赋值。初始化指定下标和成员值场景会将指定的下标和成员值以键值对的形式保存到变量中。

3. 变量赋值

   索引集合类型变量赋值分为2种：成员赋值和整体赋值。成员赋值可通过指定下标方式对某个成员赋值，若该成员不存在则等价于插入成员后赋值，若存在则刷新该成员值。整体赋值则会将被赋值变量中原有成员都清空后重新保存新的成员值。整体赋值场景不能给变量赋NULL值，否则报错。

4. 变量取值

   通过指定下标方式可获取变量中对应下标的成员值，若通过下标找不到该成员则会返回no data found的错误信息。

使用约束

- 仅支持在兼容ORA模式下（参数sql_compatibility值为ORA）创建带索引集合类型。
- 支持在匿名块、存储过程、自定义函数、package中定义带索引集合类型，其作用域各不相同。不支持在schema中定义带索引集合类型。
- NOT NULL只支持语法不支持功能。
- 当data_type为VARCHAR、NUMERIC等可以定义长度和精度的类型时，要校验集合的元素长度或者将元素转换成对应的精度，需要开启tableof_elem_constraints参数（设置behavior_compat_options参数值为tableof_elem_constraints）。
- data_type为数组类型时，数组类型的元素长度校验或精度转换也受参数tableof_elem_constraints是否开启影响。
- 通过数组类型转换成的集合类型的值不支持对元素长度校验或精度转换。
- data_type可以为基础数据类型，或存储过程内定义的record类型、集合类型和数组类型，不支持ref cursor类型。

版权所有 © 华为云计算技术有限公司

- indexby_type仅支持INTEGER和VARCHAR。
- indexby_type为VARCHAR时，开启参数tableof_elem_constraints后在对带索引集合类型赋值时会校验index值的长度，校验行为不受char_coerce_compat参数是否开启影响，index长度大于定义长度则报错；不开启参数tableof_elem_constraints则不会对索引值进行长度校验。
- 未初始化的带索引集合类型变量非NULL。
- 带索引集合类型变量不能赋NULL值，否则报错。
- 带索引集合类型变量作为入参不能赋NULL值或''。
- 不同的带索引集合类型的变量不能相互赋值。即使成员类型和下标类型相同，但集合类型名称不同，也是不同的集合类型。如 TYPE t1 IS TABLE OF int index by int; 和 TYPE t2 IS TABLE OF int index by int; 定义的两个集合类型，t1和t2是不同的集合类型，以其定义的变量不支持相互赋值（作为成员类型时该约束不保证生效，赋值逻辑受父类型影响）。
- 带索引集合类型不支持关系运算和算数运算操作。
- SELECT ... bulk collect INTO 方式赋值带索引集合类型变量时，只支持下标为INTEGER类型的集合类型，下标为VARCHAR类型集合不支持。
- 支持带索引集合类型变量作为函数的参数和返回值，此时要求参数或者返回值的类型是在package中定义的集合类型。
- 带索引的集合作为函数入参时，可以传入对应子元素类型相同的数组类型作为入参，不支持多维数组，不支持索引类型为VARCHAR（过时的方法，不建议使用该功能。可执行“SET behavior_compat_options = 'disable_rewrite_nesttable';”禁用）。
- 类型构造器目前仅支持集合类型，其参数个数的上限与用户自定义函数参数个数上限相同。对于带索引的集合类型，构造器在使用时索引的值仅支持为常量。
- 不支持对XML类型数据操作。
- 集合类型以及嵌套集合的类型不支持作为表中的一列来创建表。
- 集合类型的构造器不支持浮点数以及表达式作为下标。
- 在匿名块中定义的集合类型，匿名块执行ROLLBACK或发生EXCEPTION后，集合类型将无法继续使用。
- 设置GUC参数enable_recordtype_check_strict为on后，若成员是record类型，且record类型有列具有NOT NULL属性或DEFAULT属性，在存储过程或PACKAGE编译时会报错。
- 不支持在DML语句中访问非Schema中定义的集合类型的元素。
- 自定义集合类型不支持下推，下推场景使用变量承接集合元素。
- 通过下标访问集合成员时，解析器优先将集合类型变量视为表列名解析，如果表列名有歧义，例如两张表有同名字段，则会引发错误。

示例

示例1：无索引的集合类型。

```
--演示在存储过程中对集合进行操作。
gaussdb=# CREATE OR REPLACE PROCEDURE table_proc AS
DECLARE
    TYPE TABLE_INTEGER IS TABLE OF INTEGER;--定义集合类型
    TABLEINT TABLE_INTEGER := TABLE_INTEGER();  --声明集合类型的变量
BEGIN
    TABLEINT.extend(10);
    FOR I IN 1..10 LOOP
```

 版权所有 © 华为云计算技术有限公司

```
        TABLEINT(I) := I;
    END LOOP;
    DBE_OUTPUT.PRINT_LINE(TABLEINT.COUNT);
    DBE_OUTPUT.PRINT_LINE(TABLEINT(1));
    DBE_OUTPUT.PRINT_LINE(TABLEINT(10));
END;
/
CREATE PROCEDURE
--调用该存储过程。
gaussdb=# CALL table_proc();
10
1
10
 table_proc
------------

(1 row)

--删除存储过程。
gaussdb=# DROP PROCEDURE table_proc;
DROP PROCEDURE
--演示在存储过程中对嵌套集合进行操作。
gaussdb=# CREATE OR REPLACE PROCEDURE nest_table_proc AS
DECLARE
    TYPE TABLE_INTEGER IS TABLE OF INTEGER;--定义集合类型
    TYPE NEST_TABLE_INTEGER IS TABLE OF TABLE_INTEGER;--定义集合类型
    NEST_TABLE_VAR NEST_TABLE_INTEGER := NEST_TABLE_INTEGER(); --声明嵌套集合类型的变量
BEGIN
    NEST_TABLE_VAR.extend(10);
    FOR I IN 1..10 LOOP
        NEST_TABLE_VAR(I) := TABLE_INTEGER();
        NEST_TABLE_VAR(I).extend(10);
        NEST_TABLE_VAR(I)(I) := I;
    END LOOP;
    DBE_OUTPUT.PRINT_LINE(NEST_TABLE_VAR.COUNT);
    DBE_OUTPUT.PRINT_LINE(NEST_TABLE_VAR(1)(1));
    DBE_OUTPUT.PRINT_LINE(NEST_TABLE_VAR(10)(10));
END;
/
CREATE PROCEDURE
--调用该存储过程。
gaussdb=# CALL nest_table_proc();
10
1
10
 nest_table_proc
-----------------

(1 row)

--删除存储过程。
gaussdb=# DROP PROCEDURE nest_table_proc;
DROP PROCEDURE
```

示例2：带索引的集合类型。

```
--演示在存储过程中对带索引集合进行操作。
gaussdb=# CREATE OR REPLACE PROCEDURE index_table_proc AS
DECLARE
    TYPE TABLE_INTEGER IS TABLE OF INTEGER INDEX BY INTEGER; --定义集合类型
    TYPE TABLE_VARCHAR IS TABLE OF INTEGER INDEX BY VARCHAR; --定义集合类型
    TABLEINT_01 TABLE_INTEGER;                              --声明集合类型变量,未初始化
    TABLEINT_02 TABLE_INTEGER := TABLE_INTEGER();           --声明集合类型变量,初始化为空
    TABLEINT_03 TABLE_INTEGER := TABLE_INTEGER(2=>3,3=>4);   --声明集合类型变量,初始化指定值
    RES INTEGER;
BEGIN
    FOR I IN 1..10 LOOP
        TABLEINT_01(I) := I;     --成员赋值
        TABLEINT_02(I) := I + 1; --成员赋值
    END LOOP;
```


版权所有 © 华为云计算技术有限公司

```
    TABLEINT_01 := TABLEINT_02;  --整体赋值
    RES := TABLEINT_03(2);       --取值
    DBE_OUTPUT.PRINT_LINE(RES);
    DBE_OUTPUT.PRINT_LINE(TABLEINT_01(1));
    DBE_OUTPUT.PRINT_LINE(TABLEINT_01(10));
END;
/
CREATE PROCEDURE
--调用该存储过程。
gaussdb=# CALL index_table_proc();
3
2
11
 index_table_proc
------------------

(1 row)

--删除存储过程。
gaussdb=# DROP PROCEDURE index_table_proc;
DROP PROCEDURE
--演示在存储过程中对嵌套集合进行操作。
gaussdb=# CREATE OR REPLACE PROCEDURE nest_table_proc AS
DECLARE
    TYPE TABLE_INTEGER IS TABLE OF INTEGER INDEX BY INTEGER;          --定义集合类型
    TYPE NEST_TABLE_INTEGER IS TABLE OF TABLE_INTEGER INDEX BY INTEGER;--定义集合类型
    NEST_TABLE_VAR NEST_TABLE_INTEGER;                                --声明嵌套集合类型的变量
BEGIN
    FOR I IN 1..10 LOOP
        NEST_TABLE_VAR(I)(I) := I;
    END LOOP;
    DBE_OUTPUT.PRINT_LINE(NEST_TABLE_VAR.COUNT);
    DBE_OUTPUT.PRINT_LINE(NEST_TABLE_VAR(1)(1));
    DBE_OUTPUT.PRINT_LINE(NEST_TABLE_VAR(10)(10));
END;
/
CREATE PROCEDURE
--调用该存储过程。
gaussdb=# CALL nest_table_proc();
10
1
10
 nest_table_proc
-----------------

(1 row)

--删除存储过程。
gaussdb=# DROP PROCEDURE nest_table_proc;
DROP PROCEDURE
```

2.4.2.2 集合支持的函数

集合操作符

- =

  参数：nesttable类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：判断两个集合是否相等。

  示例：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT
 a nest := nest(1,2);
 b nest := nest(1,2);
 flag BOOL;
```


版权所有 © 华为云计算技术有限公司

```
 BEGIN
    flag := a = b;
    RAISE INFO '%', flag;
END;
 /
INFO:  t
ANONYMOUS BLOCK EXECUTE
```

- <>

  参数：nesttable类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：判断两个集合是否不相等。

  示例：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2);
  b nest := nest(1,2);
  flag BOOL;
 BEGIN
   flag := a <> b;
   RAISE INFO '%', flag;
END;
/
INFO:  f
ANONYMOUS BLOCK EXECUTE
```

- IS [NOT] NULL

  参数：nesttable类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：IS NULL判断一个集合是否为NULL，IS NOT NULL判断一个集合是否为非NULL。

  示例：

```
gaussdb=# DECLARE
   TYPE nest IS TABLE OF INT;
   a nest;
   b nest := NULL;
   c nest := nest(1,2);
   flag BOOL;
BEGIN
    flag := a IS NULL;
    RAISE INFO '%', flag;
    flag := b IS NULL;
    RAISE INFO '%', flag;
    flag := c IS NULL;
    RAISE INFO '%', flag;
 END;
 /
INFO:  t
INFO:  t
INFO:  f
ANONYMOUS BLOCK EXECUTE

gaussdb=# DECLARE
   TYPE nest IS TABLE OF INT;
   a nest;
   b nest := NULL;
   c nest := nest(1,2);
   flag BOOL;
BEGIN
    flag := a IS NOT NULL;
    RAISE INFO '%', flag;
    flag := b IS NOT NULL;
    RAISE INFO '%', flag;
```


版权所有 © 华为云计算技术有限公司

```
  flag := c IS NOT NULL;
  RAISE INFO '%', flag;
END;
/
INFO:  f
INFO:  f
INFO:  t
ANONYMOUS BLOCK EXECUTE
```

- ^=

  参数：nesttable类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：判断两个集合是否不相等。

  示例：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2);
  b nest := nest(1,2);
  flag BOOL;
BEGIN
  flag := a ^= b;
  RAISE INFO '%', flag;
END;
/
INFO:  f
ANONYMOUS BLOCK EXECUTE
```

- [NOT] IN

  参数：nesttable类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：判断一个集合是否在集合列表中。

  示例：

```
gaussdb=# DECLARE
  TYPE tab IS TABLE OF VARCHAR(10);
  tmp1 tab := tab('a', 'b');
  tmp2 tab := tab('a', 'b');
  flag BOOL;
BEGIN
  flag := tmp2 in (tmp1);
  RAISE INFO '%', flag;
END;
/
INFO:  t
ANONYMOUS BLOCK EXECUTE
```

集合 MULTISET 函数

- MULTISET UNION [ALL | DISTINCT]

  参数：nesttable类型。

  返回值：nesttable类型。

  功能描述：两个集合变量的并集，ALL表示不去除重复元素，DISTINCT表示去除重复元素，ALL为默认值。

  示例1，求两个集合变量的并集，且不去除重复元素，即MULTISET UNION ALL：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2);
  b nest := nest(2,3);
BEGIN
  a := a MULTISET UNION ALL b;
```


版权所有 © 华为云计算技术有限公司

```
  RAISE INFO '%', a;
 end;
 /
INFO:  {1,2,2,3}
ANONYMOUS BLOCK EXECUTE
```

示例2，求两个集合变量的并集，且去除重复元素，即MULTISET UNION DISTINCT：

```
gaussdb=# DECLARE
    TYPE nest IS TABLE OF INT;
    a nest := nest(1,2);
    b nest := nest(2,3);
 BEGIN
    a := a MULTISET UNION DISTINCT b;
    RAISE INFO '%', a;
 END;
 /
INFO:  {1,2,3}
ANONYMOUS BLOCK EXECUTE
```

- MULTISET EXCEPT [ALL | DISTINCT]

  参数：nesttable类型。

  返回值：nesttable类型。

  功能描述：两个集合变量的差集。如A MULTISET EXCEPT B：ALL表示去除A中A具有的元素且B也具有的元素并返回，其中去除动作计算元素个数。例如，对于特定元素，如果A中出现了m次，B中出现了n次，当 m > n时，则去除A中m - n个该元素，当m <= n时，则去除A中m个该元素；DISTINCT表示去除A中A具有的元素且B也具有的元素并返回，其中去除动作不计算元素个数。对于特定元素，如果在A中出现且又在B中出现了，则去除A中所有该元素。ALL为默认值。

  示例1，求两个集合变量的差集，去除A中A具有的元素且B也具有的元素并返回，其中去除动作计算元素个数。即MULTISET EXCEPT ALL：

  ```
  gaussdb=# DECLARE
    TYPE nest IS TABLE OF INT;
    a nest := nest(1,2,2);
    b nest := nest(2,3);
  BEGIN
    a := a MULTISET EXCEPT ALL b;
    RAISE INFO '%', a;
  END;
  /
  INFO:  {1,2}
  ANONYMOUS BLOCK EXECUTE
  ```

  示例2，求两个集合变量的差集，去除A中A具有的元素且B也具有的元素并返回，其中去除动作不计算元素个数。即MULTISET EXCEPT DISTINCT：

  ```
  gaussdb=# DECLARE
    TYPE nest IS TABLE OF INT;
    a nest := nest(1,2,2);
    b nest := nest(2,3);
  BEGIN
    a := a MULTISET EXCEPT DISTINCT b;
    RAISE INFO '%', a;
  END;
  /
  INFO:  {1}
  ANONYMOUS BLOCK EXECUTE
  ```

- MULTISET INTERSECT [ALL | DISTINCT]

  参数：nesttable类型。

  返回值：nesttable类型。

版权所有 © 华为云计算技术有限公司

功能描述：两个集合变量的交集。如 A MULTISET INTERSECT B：ALL表示保留所有重复元素（包括多次重复的元素）；DISTINCT表示返回唯一值交集，去除重复元素。ALL为默认值。

示例1，求两个集合变量的交集，不去除重复元素，即MULTISET INTERSECT ALL：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2,2);
  b nest := nest(2,2,3);
 BEGIN
  a := a MULTISET INTERSECT ALL b;
  RAISE INFO '%', a;
 END;
 /
INFO:  {2,2}
ANONYMOUS BLOCK EXECUTE
```

示例2，求两个集合变量的交集，去除重复元素，即MULTISET INTERSECT DISTINCT：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2,2);
  b nest := nest(2,2,3);
 BEGIN
  a := a MULTISET INTERSECT DISTINCT b;
  RAISE INFO '%', a;
 END;
 /
INFO:  {2}
ANONYMOUS BLOCK EXECUTE
```

集合类型函数

说明

- 以下函数定义说明里[]中的内容代表可选项，如count[()]表示可以写成count或count()。
- 在内层表达式中，不支持通过嵌套的方式调用集合类型函数。
- 当类型嵌套且集合类型作为内部类型时，无参的extend、delete、trim函数只支持extend、delete、trim形式调用，不支持extend()、delete()、trim()形式调用。

- exists(idx)

  参数：idx为INT4类型或VARCHAR类型。

  返回值：TRUE或FALSE，BOOLEAN类型。

  功能描述：查找指定位置是否存在有效元素。

  示例：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF VARCHAR2;
  a nest := nest('happy','?');
  flag BOOL;
 BEGIN
  flag := a.EXISTS(1);
  RAISE INFO '%', flag;
  flag := a.EXISTS(10);
  RAISE INFO '%', flag;
 END;
 /
INFO:  t
INFO:  f
ANONYMOUS BLOCK EXECUTE
gaussdb=# DECLARE
  TYPE nest IS TABLE OF VARCHAR2 INDEX BY VARCHAR2;
```

 版权所有 © 华为云计算技术有限公司

```
  a nest;
  flag BOOL;
 BEGIN
  a('1') := 'Be';
  a('2') := 'happy';
  a('3') := '.';
  flag := a.EXISTS('1');
  RAISE INFO '%', flag;
  flag := a.EXISTS('ddd');
  RAISE INFO '%', flag;
 END;
 /
INFO:  t
INFO:  f
ANONYMOUS BLOCK EXECUTE
```

- extend[(count[, idx])]

  参数：idx和count为INT4类型，没有参数传入时扩展1个元素值为NULL。

  返回值：无返回值。

  功能描述：extend函数仅支持nesttable类型变量使用。在nesttable变量末尾拓展1个或count个元素。存在idx下标元素时，复制count个idx下标元素到变量末尾。

  示例1，nesttable类型变量扩展1个元素且所扩展的元素值为NULL：

```
gaussdb=# DECLARE
   TYPE nest IS TABLE OF INT;
   a nest := nest(1);
 BEGIN
   RAISE INFO '%', a;
   a.EXTEND;
   RAISE INFO '%', a;
   a.EXTEND;
   RAISE INFO '%', a;
 END;
 /
INFO:  {1}
INFO:  {1,NULL}
INFO:  {1,NULL,NULL}
ANONYMOUS BLOCK EXECUTE
```

  示例2，nesttable类型变量扩展n个元素且所扩展的元素值为NULL：

```
gaussdb=# DECLARE
   TYPE nest IS TABLE OF INT;
   a nest := nest(1);
 BEGIN
   RAISE INFO '%', a;
   a.EXTEND(2);
   RAISE INFO '%', a;
 END;
 /
INFO:  {1}
INFO:  {1,NULL,NULL}
ANONYMOUS BLOCK EXECUTE
```

  示例3，nesttable类型变量扩展n个元素且所扩展的元素值为指定下标元素的值：

```
gaussdb=# DECLARE
   TYPE nest IS TABLE OF INT;
   a nest := nest(9);
 BEGIN
   RAISE INFO '%', a;
   a.EXTEND(2,1);
   RAISE INFO '%', a;
 END;
 /
INFO:  {9}
INFO:  {9,9,9}
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

- delete[(idx1[, idx2])]

  参数：idx1和idx2为INT4类型或VARCHAR2类型。

  返回值：无返回值。

  功能描述：无参数时，无索引集合nesttable类型变量删除集合类型的所有元素和空间，后续使用需要重新扩展空间，带索引集合indexbytable类型变量删除所有元素内容；一个参数删除指定位置元素（不删除空间）；两个参数则删除下标区间内的所有元素（不删除空间）。

  约束：对于无索引集合nesttable类型来说，删除元素而没删除空间，会使该空间处于无效状态，可继续赋值。作为函数返回值或自治事务入参时，会删除无效成员的存储空间，所有成员的下标会相应发生改变。

  示例1，无索引集合nesttable类型变量删除集合类型的所有元素和空间：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2,3,4,5);
 BEGIN
  RAISE INFO '%', a;
  a.DELETE();
  RAISE INFO '%', a;
 END;
 /
INFO:  {1,2,3,4,5}
INFO:  {}
ANONYMOUS BLOCK EXECUTE
```

  示例2，无索引集合nesttable类型变量删除指定位置元素：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2,3,4,5);
 BEGIN
  RAISE INFO '%', a;
  a.DELETE(3);
  RAISE INFO '%', a;
  a(3) := 3;
  RAISE INFO '%', a;
 END;
 /
INFO:  {1,2,3,4,5}
INFO:  {1,2,4,5}
INFO:  {1,2,3,4,5}
ANONYMOUS BLOCK EXECUTE
```

  示例3，无索引集合nesttable类型变量删除指定区间元素：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  a nest := nest(1,2,3,4,5);
 BEGIN
  RAISE INFO '%', a;
  a.DELETE(2,4);
  RAISE INFO '%', a(1);
  RAISE INFO '%', a(5);
  RAISE INFO '%', a;
 END;
 /
INFO:  {1,2,3,4,5}
INFO:  1
INFO:  5
INFO:  {1,5}
ANONYMOUS BLOCK EXECUTE
```

  示例4，带索引集合indexbytable类型变量删除集合类型的所有元素和空间：

```
gaussdb=# DECLARE
  TYPE t1 IS TABLE OF INT INDEX BY VARCHAR;
  v t1 := t1('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4);
 BEGIN
```


版权所有 © 华为云计算技术有限公司

```
  v.DELETE();
  RAISE INFO '%', v.COUNT();
 END;
 /
INFO:  0
ANONYMOUS BLOCK EXECUTE
```

示例5，带索引集合indexbytable类型变量删除指定位置元素：

```
gaussdb=# declare
  type t1 is table of int index by varchar;
  v t1 := t1('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4);
 begin
  raise info '%', v('a');
  v.delete('a');
  raise info '%', v('a');
 end;
 /
INFO:  1
ERROR:  no data found
CONTEXT:  PL/pgSQL function inline_code_block line 6 at RAISE
```

示例6，带索引集合indexbytable类型变量删除指定区间元素：

```
gaussdb=# declare
  type t1 is table of int index by varchar;
  v t1 := t1('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4);
 begin
  raise info '%', v('b');
  v.delete('a', 'c');
  raise info '%', v('b');
 end;
 /
INFO:  2
ERROR:  no data found
CONTEXT:  PL/pgSQL function inline_code_block line 6 at RAISE
```

- trim[(n)]

  参数：n为INT4类型。

  返回值：无返回值。

  功能描述：trim函数仅支持nesttable类型变量使用。无参数时，删除末尾一个元素空间，输入参数合法时，删除末尾指定数量元素空间。

  示例：

```
gaussdb=# DECLARE
  TYPE nest IS TABLE OF INT;
  aa nest := nest(11,22,33,44,55);
 BEGIN
  RAISE INFO 'aa:%' ,aa;
  aa.TRIM;  -- 无参数
  RAISE INFO 'aa:%' ,aa;
  aa.TRIM();  -- 无参数
  RAISE INFO 'aa:%' ,aa;
  aa.TRIM(2);  -- 参数合法
  RAISE INFO 'aa:%' ,aa;
  aa.TRIM(2);  -- 集合元素空间不足2, 参数不合法报错
 END;
 /
INFO:  aa:{11,22,33,44,55}
INFO:  aa:{11,22,33,44}
INFO:  aa:{11,22,33}
INFO:  aa:{11}
ERROR:  Subscript beyond count
CONTEXT:  PL/pgSQL function inline_code_block line 11 at assignment
```

- count[()]

  参数：无。

  返回值：INT类型。

版权所有 © 华为云计算技术有限公司

功能描述：返回集合中存在有效元素的个数。

示例1，无索引集合类型变量使用count函数：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT;
 aa nest:=nest(11,22,33,44,55);
 BEGIN
 RAISE INFO 'count:%' ,aa.COUNT;
 aa.DELETE(3);  -- 删除一个元素，下标为3的元素无效
 RAISE INFO 'count:%' ,aa.COUNT();
 END;
 /
INFO:  count:5
INFO:  count:4
ANONYMOUS BLOCK EXECUTE
```

示例2，带索引集合类型变量使用count函数：

```
gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY INT;
 aa t1;
 BEGIN
 aa(1) := 111;
 aa(2) := 222;
 aa(3) := 333;
 RAISE INFO 'count:%' ,aa.COUNT();
 END;
 /
INFO:  count:3
ANONYMOUS BLOCK EXECUTE

gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY INT;
 aa t1;
 BEGIN
 aa(1) := 111;
 aa(2) := 222;
 aa(3) := 333;
 RAISE INFO 'count:%' ,aa.COUNT();
 END;
 /
INFO:  count:3
ANONYMOUS BLOCK EXECUTE
```

- first[()]

  参数：无。

  返回值：INT类型或VARCHAR类型。

  功能描述：返回集合中第一个有效元素的下标。

  示例1，无索引集合类型变量使用first函数：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT;
 aa nest:=nest(11,22,33,44,55);
 BEGIN
 RAISE INFO 'first:%' ,aa.FIRST();
 aa.DELETE(1);
 RAISE INFO 'first:%' ,aa.FIRST;  -- 下标1的元素无效，第一个有效元素下标为2
 END;
 /
INFO:  first:1
INFO:  first:2
ANONYMOUS BLOCK EXECUTE
```

  示例2，带索引集合类型变量使用first函数：

```
gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY INT;
 aa t1;
 BEGIN
```


版权所有 © 华为云计算技术有限公司

```
 aa(3) := 111;
 aa(2) := 222;
 aa(1) := 333;
 RAISE INFO 'first:%' ,aa.FIRST;
 END;
 /
INFO:  first:1
ANONYMOUS BLOCK EXECUTE

gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY VARCHAR;
 aa t1;
 BEGIN
 aa('aaa') := 111;
 aa('bbb') := 222;
 aa('ccc') := 333;
 RAISE INFO 'first:%' ,aa.FIRST;
 END;
 /
INFO:  first:aaa
ANONYMOUS BLOCK EXECUTE
```

- last[()]

  参数：无。

  返回值：INT类型或VARCHAR类型。

  功能描述：返回集合中最后一个有效元素的下标。

  示例1，无索引集合类型变量使用last函数：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT;
 aa nest:=nest(11,22,33,44,55);
 BEGIN
 RAISE INFO 'last:%' ,aa.LAST;
 aa.DELETE(5);
 RAISE INFO 'last:%' ,aa.LAST();  -- 下标5的元素无效，最后一个有效元素下标为4
 END;
 /
INFO:  last:5
INFO:  last:4
ANONYMOUS BLOCK EXECUTE
```

  示例2，带索引集合类型变量使用last函数：

```
gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY VARCHAR;
 aa t1;
 BEGIN
 aa(3) := 111;
 aa(2) := 222;
 aa(1) := 333;
 RAISE INFO 'last:%' ,aa.LAST();
 END;
 /
INFO:  last:3
ANONYMOUS BLOCK EXECUTE

gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY VARCHAR;
 aa t1;
 BEGIN
 aa(3) := 111;
 aa(2) := 222;
 aa(1) := 333;
 RAISE INFO 'last:%' ,aa.LAST();
 END;
 /
INFO:  last:ccc
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

- prior(idx)

  参数：idx为int类型或varchar类型。

  返回值：int类型或varchar类型。

  功能描述：返回集合中给定下标的前一个有效元素下标，下标不存在返回集合中最后一个元素。

  示例1，无索引集合类型变量使用prior函数：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT;
 aa nest:=nest(11,22,33,44,55);
 BEGIN
 RAISE INFO 'prior:%' ,aa.PRIOR(3);
 END;
 /
INFO:  prior:2
ANONYMOUS BLOCK EXECUTE
```

  示例2，带索引集合类型变量使用prior函数：

```
gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY VARCHAR;
 aa t1;
 BEGIN
 aa('ccc') := 111;
 aa('bbb') := 222;
 aa('aaa') := 333;
 RAISE INFO 'prior:%' ,aa.PRIOR('bbb');
 RAISE INFO 'prior:%' ,aa.PRIOR('ddd');
 END;
 /
INFO:  prior:aaa
INFO:  prior:ccc
ANONYMOUS BLOCK EXECUTE
```

- next(idx)

  参数：idx为INT类型或VARCHAR类型，下标不存在返回集合中第一个元素。

  返回值：INT类型或VARCHAR类型。

  功能描述：返回集合中给定下标的后一个有效元素下标。

  示例1，无索引集合类型变量使用next函数：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT;
 aa nest:=nest(11,22,33,44,55);
 BEGIN
 RAISE INFO 'next:%' ,aa.NEXT(3);
 END;
 /
INFO:  next:4
ANONYMOUS BLOCK EXECUTE
```

  示例2，带索引集合类型变量使用next函数：

```
gaussdb=# DECLARE
 TYPE t1 IS TABLE OF INT INDEX BY INT;
 aa t1;
BEGIN
 aa(3) := 111;
 aa(2) := 222;
 aa(1) := 333;
 RAISE INFO 'next:%' ,aa.NEXT(2);
 RAISE INFO 'next:%' ,aa.NEXT(-999);
 END;
 /
INFO:  next:3
INFO:  next:1
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

- limit

  参数：无。

  返回值：NULL。

  功能描述：仅用于nesttable类型变量，且返回NULL值。

  示例：

```
gaussdb=# DECLARE
 TYPE nest IS TABLE OF INT;
 aa nest:=nest(11,22,33,44,55);
 BEGIN
 RAISE INFO 'limit:%' ,aa.LIMIT;
 END;
 /
INFO:  limit:<NULL>
ANONYMOUS BLOCK EXECUTE
```

集合相关函数

- unnest_table(anynesttable)或unnest(anynesttable)

  参数：任意无索引集合类型。

  描述：返回给定nesttable中所有有效元素的结果集，会返回多行数据。

  返回类型：setof anyelement。

  示例：

```
-- 当集合的元素类型为record类型且有元素为NULL时，该元素不会被返回，而是会返回一个非NULL的record值，该record的所有列为NULL。
gaussdb=# CREATE OR REPLACE p1()
gaussdb-# AS
gaussdb$#     TYPE t1 IS TABLE OF INT;
gaussdb$#     v2 t1 := t1(NULL, 2, 3, 4, NULL);
gaussdb$#     tmp INT;
gaussdb$#     CURSOR c1 IS SELECT * FROM UNNEST_TABLE(v2);
gaussdb$# BEGIN
gaussdb$# OPEN c1;
gaussdb$# FOR i IN 1 .. v2.COUNT LOOP
gaussdb$#     FETCH c1 INTO tmp;
gaussdb$#     IF tmp IS NULL THEN
gaussdb$#         DBE_OUTPUT.PRINT_LINE(i || ': is null');
gaussdb$#     ELSE
gaussdb$#         DBE_OUTPUT.PRINT_LINE(i || ': ' || tmp);
gaussdb$#     END IF;
gaussdb$# END LOOP;
gaussdb$# CLOSE c1;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# CALL p1();
1: is null
2: 2
3: 3
4: 4
5: is null
 p1
----

(1 row)
gaussdb=# DROP PROCEDURE IS EXISTS p1();
DROP PROCEDURE

-- nesttable嵌套record类型示例
gaussdb=# CREATE OR REPLACE PROCEDURE p1() IS
gaussdb$#   TYPE rec IS RECORD(c1 INT, c2 INT);
gaussdb$#   TYPE t1 IS TABLE OF rec;
gaussdb$#   v t1 := t1(rec(1, 1), rec(2, NULL), rec(NULL, NULL), NULL);
```


版权所有 © 华为云计算技术有限公司

```
gaussdb$#   v2 t1 := t1();
gaussdb$#   CURSOR cur IS SELECT * FROM UNNEST(v);
gaussdb$# BEGIN
gaussdb$#   v2.EXTEND(v.COUNT);
gaussdb$#   OPEN cur;
gaussdb$#   FOR i IN 1 .. v.COUNT LOOP
gaussdb$#     FETCH cur INTO v2(i);
gaussdb$#     RAISE INFO '%', v2(i);
gaussdb$#   END LOOP;
gaussdb$#   CLOSE cur;
gaussdb$# END
gaussdb$# /
CREATE PROCEDURE
gaussdb=# CALL p1();
INFO:  (1,1)
INFO:  (2,)
INFO:  (,)
INFO:  (,)
 p1
----

(1 row)
gaussdb=# DROP PROCEDURE IF EXISTS p1();
DROP PROCEDURE
```

- unnest_table(anyindexbytable)或unnest(anyindexbytable)

  参数：任意带索引集合类型。

  描述：返回给定indexbytable中所有元素根据索引排序后的结果集，会返回多行数据。

  返回类型：setof anyelement

  约束：只支持INDEX BY INT类型。

  示例：

```
-- 当集合的元素类型为record类型且有元素为NULL时，该元素不会被返回，而是会返回一个非NULL的record值，该record的所有列为NULL。
gaussdb=# CREATE OR REPLACE PROCEDURE p1()
gaussdb-# AS
gaussdb$#     TYPE t1 IS TABLE OF INT INDEX BY INT;
gaussdb$#     v2 t1 := t1(1=>1, -10=>(-10), 6=>6, 4=>NULL);
gaussdb$#     tmp INT;
gaussdb$#     CURSOR c1 IS SELECT * FROM UNNEST_TABLE(v2);
gaussdb$# BEGIN
gaussdb$# OPEN c1;
gaussdb$# FOR i IN 1 .. v2.COUNT LOOP
gaussdb$#     FETCH c1 INTO tmp;
gaussdb$#     IF tmp IS NULL THEN
gaussdb$#         DBE_OUTPUT.PRINT_LINE(i || ': is null');
gaussdb$#     ELSE
gaussdb$#         DBE_OUTPUT.PRINT_LINE(i || ': ' || tmp);
gaussdb$#     END IF;
gaussdb$# END LOOP;
gaussdb$# CLOSE c1;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# CALL p1();
1: -10
2: 1
3: is null
4: 6
 p1
----

(1 row)
gaussdb=# DROP PROCEDURE IF EXISTS p1();
DROP PROCEDURE
```


版权所有 © 华为云计算技术有限公司

```
-- index by table嵌套record类型示例
gaussdb=# CREATE OR REPLACE PROCEDURE p1() IS
gaussdb$#   TYPE rec IS RECORD(c1 INT, c2 INT);
gaussdb$#   TYPE t1 IS TABLE OF rec INDEX BY INT;
gaussdb$#   v t1 := t1(1 => rec(1, 1), 2 => rec(2, NULL), 3 => rec(NULL, NULL), 4 => NULL);
gaussdb$#   v2 t1 := t1();
gaussdb$#   CURSOR cur IS SELECT * FROM UNNEST(v);
gaussdb$# BEGIN
gaussdb$#   OPEN cur;
gaussdb$#   FOR i IN 1 .. v.COUNT LOOP
gaussdb$#     FETCH cur INTO v2(i);
gaussdb$#     RAISE INFO '%', v2(i);
gaussdb$#   END LOOP;
gaussdb$#   CLOSE cur;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# CALL p1();
INFO:  (1,1)
INFO:  (2,)
INFO:  (,)
INFO:  (,)
 p1
----

(1 row)
gaussdb=# drop procedure if exists p1();
DROP PROCEDURE
```

2.4.3 record

record 类型的变量

定义一个record类型，然后使用该类型来声明一个变量，该变量即为record变量。

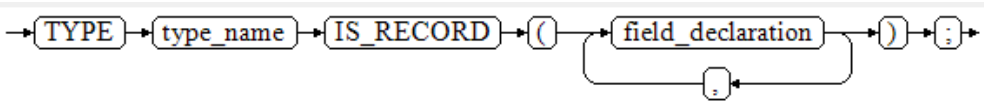
语法

定义record类型的语法为：

```
TYPE type_name IS_RECORD (field_declaration [,...] );
```

语法图如图2-4所示。

图 2-4 定义 record 类型的语法

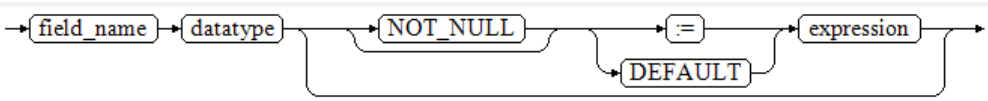


其中field_declaration为：

```
field_name datatype [ [NOT_NULL] { := | DEFAULT } expression]
```

field_declaration语法图如图2-5所示。

图 2-5 field_declaration



版权所有 © 华为云计算技术有限公司

对以上语法格式的解释如下：

- type_name：声明的类型名称。
- field_name：record类型中的成员名称。
- datatype：record类型中成员的类型。
- expression：设置默认值的表达式。

使用约束

在GaussDB中：

- record类型的变量的赋值支持：
  - 在函数或存储过程的声明阶段，声明一个record类型，并且可以在该类型中定义成员变量。
  - 一个record变量到另一个record变量的赋值。
  - SELECT INTO和FETCH向一个record类型的变量中赋值。
  - 将一个NULL值赋值给一个record变量。
- record类型构造器赋值时支持使用=>，约束如下：
  - record类型构造器赋值时使用=>功能，仅在兼容ORA模式数据库下（即sql_compatibility='ORA'）支持。
  - 支持在package内部定义record类型。
  - record类型构造器赋值时使用=>功能，仅在入参赋值连续使用=>到结尾时支持，如record_name(elename1 => val1, elename2 => val2, elename3 => val3)或record_name(val1, elename2 => val2, elename3 => val3)；不支持非连续使用=>赋值或者使用=>赋值时未到最后一个入参，如record_name(elename1 => val1, elename2 => val2, val3)或record_name(val1, elename2 => val2, val3)。
- 不支持INSERT和UPDATE语句使用record变量进行插入数据和更新数据。
- 不支持record类型的构造器作为函数或存储过程参数的默认值。
- 如果成员有复合类型，在声明阶段不支持指定默认值，该行为同声明阶段的变量一样。
- 如果成员有record类型，则内层record类型的默认值不支持传递至外层record类型中。
- 设置GUC参数enable_recordtype_check_strict为on后，成员是record类型，且record类型有列具有NOT NULL属性或DEFAULT属性，在存储过程或PACKAGE编译时会报错。
- 设置GUC参数enable_recordtype_check_strict为on后，当PACKAGE里有record类型，且record类型有列具有NOT NULL属性或DEFAULT属性会在创建或编译时报错。
- 设置GUC参数enable_recordtype_check_strict为on后，存储过程里的record类型NOT NULL功能生效。
- 设置GUC参数behavior_compat_options启用proc_outparam_override选项后：
  - 带有out出参且返回record的函数，支持在外部SQL使用SELECT、CALL调用，在存储过程内支持使用PERFORM、表达式调用。
  - 在函数直接返回未定义的record类型时，至少需要带有一个out参数；如果返回的是已经定义的record类型，则可以不带out参数。详见**示例**。

版权所有 © 华为云计算技术有限公司

  - 带out出参返回复合类型或record的函数，需要保持函数的预期返回类型与实际返回类型一致。
- 在INSERT INTO table_name VALUES record_var语法中，record_var仅支持record类型变量和单层数组嵌套的record类型变量，不支持多层数组嵌套的record类型变量。
- PL/SQL中静态DDL语句不支持使用record类型变量和成员。
- record变量整体赋值时，按照列的声明顺序依次进行赋值，若某个列在赋值时出现异常，已赋值的列不会回滚。

示例

```
-- 创建一个表，并插入一些数据。
gaussdb=# CREATE TABLE emp_rec(
gaussdb(#     empno NUMERIC(4,0) NOT NULL,
gaussdb(#     ename VARCHAR(10)
gaussdb(# );
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'empno' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# INSERT INTO emp_rec VALUES(111, 'aaa'), (222, 'bbb'), (333, 'ccc');
INSERT 0 3
-- 表定义如下。
gaussdb=# \d emp_rec
          Table "public.emp_rec"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 empno  | numeric(4,0)          | not null
 ename  | character varying(10) |

-- 演示在函数中对record进行操作。
gaussdb=# CREATE OR REPLACE FUNCTION regress_record(p_w VARCHAR2) RETURNS VARCHAR2 AS $$
gaussdb$# DECLARE
gaussdb$#     -- 声明一个record类型。
gaussdb$#     TYPE rec_type IS RECORD (name  VARCHAR2(100), epno INT);
gaussdb$#     employer rec_type;
gaussdb$#     -- 使用%type声明record类型。
gaussdb$#     TYPE rec_type1 IS RECORD (name  emp_rec.ename%TYPE, epno INT :=10);
gaussdb$#     employer1 rec_type1;
gaussdb$#     -- 声明带有默认值的record类型。
gaussdb$#     TYPE rec_type2 IS RECORD (
gaussdb$#         name VARCHAR2 NOT NULL := 'SCOTT',
gaussdb$#         epno INT NOT NULL :=10
gaussdb$#     );
gaussdb$#     employer2 rec_type2;
gaussdb$#     CURSOR C1 IS  SELECT ename,empno FROM emp_rec ORDER BY 1 LIMIT 1;
gaussdb$# BEGIN
gaussdb$#     -- 对一个record类型的变量的成员赋值。
gaussdb$#     employer.name := 'WARD';
gaussdb$#     employer.epno = 18;
gaussdb$#     RAISE INFO 'employer name: % , epno:%', employer.name, employer.epno;
gaussdb$#
gaussdb$#     -- 将一个record类型的变量赋值给另一个变量。
gaussdb$#     employer1 := employer;
gaussdb$#     RAISE INFO 'employer1 name: % , epno: %',employer1.name, employer1.epno;
gaussdb$#
gaussdb$#     -- 将一个record类型变量赋值为NULL。
gaussdb$#     employer1 := NULL;
gaussdb$#     RAISE INFO 'employer1 name: % , epno: %',employer1.name, employer1.epno;
gaussdb$#
gaussdb$#     -- 获取record变量的默认值。
gaussdb$#     RAISE INFO 'employer2 name: % ,epno: %', employer2.name, employer2.epno;
gaussdb$#
gaussdb$#     -- 在for循环中使用record变量。
gaussdb$#     FOR employer IN SELECT ename,empno FROM emp_rec ORDER BY 1 LIMIT 1 LOOP
gaussdb$#         RAISE INFO 'employer name: % , epno: %', employer.name, employer.epno;
```


版权所有 © 华为云计算技术有限公司

```
gaussdb$#     END LOOP;
gaussdb$#
gaussdb$#     -- 在select into中使用record变量。
gaussdb$#     SELECT ename,empno  INTO employer2 FROM emp_rec ORDER BY 1 LIMIT 1;
gaussdb$#     RAISE INFO 'employer name: % , epno: %', employer2.name, employer2.epno;
gaussdb$#
gaussdb$#     -- 在cursor中使用record变量。
gaussdb$#     OPEN C1;
gaussdb$#     FETCH C1 INTO employer2;
gaussdb$#     RAISE INFO 'employer name: % , epno: %', employer2.name, employer2.epno;
gaussdb$#     CLOSE C1;
gaussdb$#     RETURN employer.name;
gaussdb$# END; $$
gaussdb-# LANGUAGE plpgsql;
CREATE FUNCTION

-- 调用该函数。
gaussdb=# CALL regress_record('abc');
INFO:  employer name: WARD , epno:18
INFO:  employer1 name: WARD , epno: 18
INFO:  employer1 name: <NULL> , epno: <NULL>
INFO:  employer2 name: SCOTT ,epno: 10
INFO:  employer name: aaa , epno: 111
INFO:  employer name: aaa , epno: 111
INFO:  employer name: aaa , epno: 111
 regress_record
----------------
 aaa
(1 row)
-- 删除函数。
gaussdb=# DROP FUNCTION regress_record;
DROP FUNCTION
-- 删除表。
gaussdb=# DROP TABLE emp_rec;
DROP TABLE

-- 打开兼容性参数proc_outparam_override时，返回已定义的record类型，函数可以不需out参数。
gaussdb=# CREATE TYPE rec_type IS (c1 INT, c2 INT);
CREATE TYPE
gaussdb=# SET behavior_compat_options = 'proc_outparam_override';
SET
gaussdb=# CREATE OR REPLACE FUNCTION func(a in INT) RETURN rec_type IS
gaussdb$#     r rec_type;
gaussdb$# BEGIN
gaussdb$#     r.c1:=1;
gaussdb$#     r.c2:=1;
gaussdb$#     RETURN r;
gaussdb$# END;
gaussdb$# /
CREATE FUNCTION
gaussdb=# CALL func(0);
 c1 | c2
----+----
  1 |  1
(1 row)
gaussdb=# DROP FUNCTION func;
DROP FUNCTION
gaussdb=# DROP TYPE rec_type;
DROP TYPE
-- 打开兼容性参数proc_outparam_override时，函数直接返回未定义的record类型时，至少需要带有一个out参数。
gaussdb=# SET behavior_compat_options = 'proc_outparam_override';
SET
gaussdb=# CREATE OR REPLACE FUNCTION func(a OUT INT) RETURN RECORD IS
gaussdb$#     TYPE rc IS RECORD(c1 INT, c2 INT);
gaussdb$#     r rc;
gaussdb$# BEGIN
gaussdb$#     r.c1 := 1;
gaussdb$#     r.c2 := 1;
```


版权所有 © 华为云计算技术有限公司

```
gaussdb$#     a := 999;
gaussdb$#     RETURN r;
gaussdb$# END;
gaussdb$# /
CREATE FUNCTION
gaussdb=# CALL func(1);
 func  |  a
-------+-----
 (1,1) | 999
(1 row)
gaussdb=# DROP FUNCTION func;
DROP FUNCTION

-- 创建ORA兼容模式数据库，并切换到此数据库。
gaussdb=#  CREATE DATABASE ora_compatible_db DBCOMPATIBILITY 'ORA';
CREATE DATABASE
gaussdb=#  \c ora_compatible_db;
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ora_compatible_db" as user "omm".
-- record在pkg内部定义，record构造器全部入参赋值使用=>，执行成功。
ora_compatible_db=#
CREATE OR REPLACE PACKAGE pa_test1
AS
 PROCEDURE p_display1(a NUMBER, b  VARCHAR);
 TYPE t1 IS RECORD(va INT,vb INT);
 v1 t1;
END pa_test1;
/
ora_compatible_db=#
CREATE OR REPLACE PACKAGE BODY pa_test1
AS
  PROCEDURE p_display1(a NUMBER, b  VARCHAR) AS
 BEGIN
  v1 := t1(va => 1, vb => 2);
  DBE_OUTPUT.PUT_LINE('pkg test1');
 END;
END pa_test1;
/
ora_compatible_db=# CALL pa_test1.p_display1(b => 'm', a => 1);
 p_display1
------------

(1 row)
-- record在pkg内部定义，record构造器入参赋值未连续使用=>，提示错误。
ora_compatible_db=#
CREATE OR REPLACE PACKAGE pa_test1
AS
 PROCEDURE p_display1(a NUMBER, b  VARCHAR);
 TYPE t1 IS RECORD(va INT,vb INT);
 v1 t1;
END pa_test1;
/
CREATE OR REPLACE PACKAGE BODY pa_test1
AS
  PROCEDURE p_display1(a NUMBER, b  VARCHAR) AS
 BEGIN
  v1 := t1(va => 1, 2);
  DBE_OUTPUT.PUT_LINE('pkg test1');
 END;
END pa_test1;
/
ora_compatible_db=# CALL pa_test1.p_display1(b => 'm', a => 1);
ERROR:  positional argument cannot follow named argument
LINE 1: SELECT "17347.t1"(va => 1, 2)
                          ^
QUERY:  SELECT "17347.t1"(va => 1, 2)
CONTEXT:  referenced column: 17347.t1
PL/pgSQL function p_display1(numeric,character varying) line 2 at assignment
-- 切换回原数据库，并删除创建的数据库。
```


版权所有 © 华为云计算技术有限公司

```
ora_compatible_db=# \c postgres;
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "omm".
gaussdb=# DROP DATABASE ora_compatible_db;
DROP DATABASE
```

2.5 声明语法

2.5.1 基本结构

结构

PL/SQL块中可以包含子块，子块可以位于PL/SQL中任何部分。PL/SQL块的结构如下：

- 声明部分：声明PL/SQL用到的变量、类型、游标以及局部存储过程和函数。
  ```
  DECLARE
  ```

  说明

  不涉及变量声明时声明部分可以没有。

  - 对匿名块来说，没有变量声明部分时，可以省去DECLARE关键字。
  - 对存储过程来说，不使用DECLARE关键字时， AS关键字等同于DECLARE。即使没有显式的变量声明，AS关键字也必须保留在语法结构中。
- 执行部分：过程及SQL语句，程序的主要部分。必选。
  ```
  BEGIN
  ```
- 执行异常部分：错误处理。可选。
  ```
  EXCEPTION
  ```
- 结束。必选。
  ```
  END;
  /
  ```

须知

禁止在PL/SQL块中使用连续的Tab，连续的Tab可能会造成在使用gsql工具带“-r”参数执行PL/SQL块时出现异常。

分类

PL/SQL块可以分为以下几类：

- 匿名块：动态构造，只能执行一次。语法请参见图2-6。
- 子程序：存储在数据库中的存储过程、函数和操作符及高级包等。子程序创建完成后，可以在其他程序中被调用。

2.5.2 匿名块

匿名块（Anonymous Block）一般用于不频繁执行的脚本或不重复进行的活动。它们在一个会话中执行，并不被存储。

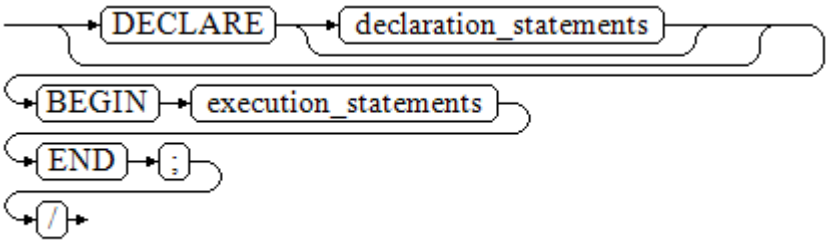
版权所有 © 华为云计算技术有限公司

语法

```
[ DECLARE [ declaration_statements ] ]
BEGIN execution_statements
END ;
/
```

匿名块（Anonymous Block）语法图如图2-6所示。

图 2-6 anonymous_block::=



- 匿名块程序实施部分，以BEGIN语句开始，以END语句停顿，以一个分号结束。输入“/”按回车执行它。

  须知

  最后的结束符“/”必须独占一行，不能直接跟在END后面。

- 声明部分包括变量定义、类型、游标定义等。
- 最简单的匿名块不执行任何命令。但一定要在任意实施块里至少有一个语句，甚至是一个NULL语句。

示例

下面列举了基本的匿名块程序：

```
--空语句块
gaussdb=# BEGIN
    NULL;
END;
/

--将信息打印到控制台：
gaussdb=# BEGIN
    dbe_output.print_line('hello world!');
END;
/
hello world!
ANONYMOUS BLOCK EXECUTE
--将变量内容打印到控制台：
gaussdb=# DECLARE
    my_var VARCHAR2(30);
BEGIN
    my_var :='world';
    dbe_output.print_line('hello'||my_var);
END;
/
helloworld
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

匿名块使用 JDBC 驱动绑定参数

须知

- 该功能仅支持ORA兼容性数据库，并且仅支持Oracle风格的匿名块。
- 在全密态加密场景中：
  - 当JDBC连接配置enable_ce=1时不支持匿名块绑定参数。
- 当匿名块内定义的变量（包括显示定义的变量或类型）与绑定参数变量（$1、$2...）命名冲突时会报错。
- 声明为自治事务的匿名块不支持绑定参数。
- 匿名块绑定参数个数上限为32767，绑定出参个数上限为1664。

前置条件：已连接数据库并且连接对象为conn。

匿名块使用JDBC驱动绑定参数的步骤如下：

步骤1 调用Connection的prepareCall方法创建调用语句对象cstmt。

```
CallableStatement cstmt = conn.prepareCall("DECLARE var int; BEGIN var := ?; ? := 50; END;");
```

步骤2 调用CallableStatement的setInt方法设置参数。

```
cstmt.setInt(1, 20);
```

步骤3 调用CallableStatement的registerOutParameter方法注册输出参数。

```
cstmt.registerOutParameter(2, Types.INTEGER);  //注册out类型的参数，类型为整型。
```

步骤4 调用CallableStatement的execute方法执行匿名块语句。

```
cstmt.execute();
```

步骤5 调用CallableStatement的getInt方法获取out输出参数。

```
int out = cstmt.getInt(2);
```

步骤6 调用CallableStatement的close方法关闭调用语句对象cstmt。

```
cstmt.close();
```

----结束

说明

- 507.0.0版本的JDBC连接历史版本的数据库（503.1.0及之后的版本），不支持匿名块使用JDBC驱动绑定参数。历史版本的JDBC（503.1.0及之后的版本）连接507.0.0版本的数据库，支持匿名块使用JDBC驱动绑定入参，不支持匿名块使用JDBC驱动绑定出参。
- 从507.0.0之前版本升级到507.0.0及更高版本时，在升级观察期期间：使用升级后的JDBC连接数据库，支持匿名块使用JDBC驱动绑定参数；使用升级前的JDBC（503.1.0及之后的版本）连接507.0.0版本的数据库，支持匿名块使用JDBC驱动绑定入参，不支持匿名块使用JDBC驱动绑定出参。

2.5.3 子程序

存储在数据库中的存储过程、函数、操作符以及高级包等对象。

在数据库上建立好后子程序后，即可在其他程序中调用。

版权所有 © 华为云计算技术有限公司

2.5.3.1 独立的子程序

在Schema内创建的子程序，在Schema内创建的存储过程、函数、包等，可称为独立的子程序。

详细使用请参考：

- **CREATE PACKAGE**
- **CREATE FUNCTION**
- **CREATE PROCEDURE**

2.5.3.2 包内的子程序

在包体内创建的子程序，在包内声明创建的存储过程及函数。

详细使用请参考：**CREATE PACKAGE**。

2.5.3.3 嵌套的子程序

在PL/SQL块内创建的子程序，支持在匿名块、存储过程、函数、包内的存储过程及函数中，声明并创建子存储过程或子函数。

注意事项

- 在ORA兼容性数据库下使用。
- 最大嵌套层数限制通过GUC参数max_subpro_nested_layers控制（默认值为3，取值范围0~100）。若嵌套子程序中含有匿名块，匿名块不计入层数，但匿名块内嵌套子程序计入总层数。
- 嵌套子程序不支持重载、不支持使用SETOF。
- 嵌套子程序内不支持定义为自治事务，但可调用含有自治事务的存储过程或函数。
- 子函数（FUNCTION）不支持直接调用且必须包含返回值，子存储过程（PROCEDURE）不支持在表达式中调用。
- 嵌套子程序不支持PERFORM调用，动态语句中不能包含嵌套子程序。
- 当前嵌套子程序的修饰符支持如下，其余修饰符暂不支持。
  ```
  {IMMUTABLE | STABLE | VOLATILE }
  {CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }
  ```
- 仅支持一个限定符引用嵌套子程序或嵌套子程序的变量。
- 当子函数（FUNCTION）返回值类型为函数自定义的record类型时，无法通过subfunc().col的方式访问子函数返回值的列属性，执行时会报错。
- 嵌套子程序的声明必须位于声明部分的最后（在其他变量、游标、类型等声明完成之后再声明嵌套子程序）。
- 嵌套子程序只能在声明的函数或存储过程内部调用，外部不可使用。
- 嵌套子程序不支持debugger打断点，但支持step单步调试。
- 其余注意事项同存储过程及函数一致。

语法格式

- 创建子存储过程语法格式：
  ```
  PROCEDURE procedure_name [ (parameters) ]
    [{IMMUTABLE | STABLE | VOLATILE }
  ```


版权所有 © 华为云计算技术有限公司

```
    | {CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }]
  { IS | AS }
  [ declarations ]
BEGIN
  plsql_body
END;
```

- 创建子函数语法：

```
FUNCTION function_name  [ (parameters) ] RETURN rettype
  [{IMMUTABLE  | STABLE  | VOLATILE }
   | {CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }]
  { IS | AS }
  [ declarations ]
BEGIN
  plsql_body
END;
```

在declarations部分可再定义下层的嵌套子程序。

- 示例：

```
-- 创建一个存储过程
CREATE OR REPLACE PROCEDURE proc_test() AS
  -- 声明并定义一个子存储过程
  PROCEDURE proc_sub() IS
  BEGIN
    dbe_output.put_line('this is subprogram');
  END;
BEGIN
  dbe_output.put_line('this is a procedure');
  -- 执行块内调用子存储过程
  proc_sub();
END;
/
-- 外部调用存储过程
BEGIN
  proc_test;
END;
/
-- 输出结果
this is a procedure
this is subprogram
ANONYMOUS BLOCK EXECUTE
```

嵌套子程序的声明及定义规则

- 嵌套子程序不可重复声明，不可重复定义，不支持重载。
- 嵌套子程序的标识符不得与变量名，关键字同名。
- 支持先声明后定义，所有先声明的子程序必须在后续声明块中找到其定义。

嵌套子程序的调用规则

- 嵌套子程序可调用自身实现递归调用。
- 嵌套子程序可调用上层作用域声明的子程序。
- 嵌套子程序可调用本地声明的下层子程序，但不可访问下层子程序内部的嵌套子程序。
- 嵌套子程序可调用相同作用域中先于自身声明的子程序。

示例：

```
-- 调用自身
CREATE OR REPLACE PROCEDURE proc_test(var1 int) AS
  PROCEDURE proc_sub(var2 int) IS
  BEGIN
    dbe_output.put_line('var = ' || var2);
    IF var2 > 1 THEN
```


版权所有 © 华为云计算技术有限公司

```
            proc_sub(var2 - 1);
        END IF;
    END;
BEGIN
    proc_sub(var1);
END;
/
BEGIN
    proc_test(3);
END;
/
--输出结果
var = 3
var = 2
var = 1

--调用上层子程序
CREATE OR REPLACE PROCEDURE proc_test(var1 int) AS
    PROCEDURE procsub_1(var2 int) IS
    BEGIN
 proc_test(var2 - 1);
    END;
BEGIN
    dbe_output.put_line('proc_test var1 = ' || var1);
 IF var1 > 1 THEN
       procsub_1(var1);
 END IF;
END;
/
BEGIN
    proc_test(3);
END;
/
-- 输出结果
proc_test var1 = 3
proc_test var1 = 2
proc_test var1 = 1

-- 调用本地声明的下层子程序
CREATE OR REPLACE PROCEDURE proc_test() AS
    PROCEDURE proc_sub1 IS
        procedure proc_sub2 IS
        BEGIN
            dbe_output.put_line('--this is subprogram2 begin');
            dbe_output.put_line('--this is subprogram2 end');
        END;
    BEGIN
 dbe_output.put_line('this is subprogram1 begin');
 proc_sub2();
 dbe_output.put_line('this is subprogram1 end');
    END;
BEGIN
    dbe_output.put_line('this is a procedure begin');
    proc_sub1();
    dbe_output.put_line('this is a procedure end');
END;
/
BEGIN
    proc_test;
END;
/
-- 输出结果
this is a procedure begin
this is subprogram1 begin
--this is subprogram2 begin
--this is subprogram2 end
this is subprogram1 end
this is a procedure end
```


版权所有 © 华为云计算技术有限公司

嵌套子程序的变量

变量类型包括：基础类型，record类型、table of类型、cursor类型、varray类型等其它PL/SQL支持的变量类型。

- 可访问的变量：
  - 自身声明的变量。
  - 上层子程序声明的变量。
- 变量的访问规则：
  - 如果变量不带限定符，首先将在本程序内查找变量。如果变量名不存在，则向上层查找变量，以此类推。
  - 如果变量带有限定符，将在限定符的区域查找。目前仅支持一个限定符的使用。

2.6 基本语句

在PL/SQL编程中，需要定义变量、进行赋值、调用存储过程等基础操作。本节将介绍PL/SQL中的基本语句，包括定义变量、赋值语句、调用语句以及返回语句。

说明

避免在存储过程中调用包含密码的SQL语句，因为存储在数据库中的存储过程文本可能被其他具有访问权限的用户查看，从而导致密码信息泄露。若存储过程中包含其他敏感信息，建议配置存储过程的访问权限，以确保敏感数据的安全性。

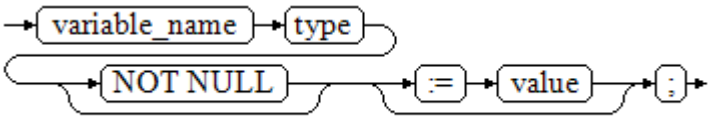
2.6.1 定义变量

介绍PL/SQL中变量的声明，以及该变量在代码中的作用域。

变量声明

变量声明语法如图2-7所示。

图 2-7 declare_variable::=



对以上语法格式的解释如下：

- variable_name，为变量名。
- type，为变量类型。
- value，是该变量的初始值（如果不给定初始值，则初始为NULL）。value也可以是表达式。

示例

```
gaussdb=# DECLARE
    emp_id  INTEGER := 7788;  --定义变量并赋值
BEGIN
```


版权所有 © 华为云计算技术有限公司

```
    emp_id := 5*7784;  --变量赋值
END;
/
```

变量类型除了支持基本类型，还可使用%TYPE和%ROWTYPE去声明一些与其他表字段或表结构本身相关的变量。

%TYPE属性

%TYPE主要用于声明某个与其他变量类型（例如，表中某列的类型）相同的变量。假如定义一个my_name变量，它的变量类型与employee表的firstname列类型相同，可以通过如下定义：

```
my_name employee.firstname%TYPE
--示例
DROP TABLE IF EXISTS employee;
CREATE TABLE employee(firstname varchar,secondname varchar);
DECLARE
    my_name employee.firstname%TYPE;
BEGIN
    my_name = 'abc';
    DBE_OUTPUT.PRINT_LINE(my_name);
END;
/
```

这样定义可以带来两个优势，首先，无需预先知道employee表的firstname列的数据类型。其次，当firstname列的数据类型发生变化时，也无需修改my_name的类型定义。

%ROWTYPE属性

%ROWTYPE属性主要用于声明一组数据的类型，用于存储表中的一行数据，或从游标匹配的结果中获取。若需要一组数据，其字段名称与字段类型均与employee表相同。可以通过如下定义：

```
my_employee employee%ROWTYPE
DROP TABLE IF EXISTS employee;
CREATE TABLE employee(firstname varchar,secondname varchar);
DECLARE
    my_employee employee%ROWTYPE;
BEGIN
    my_employee.firstname := 'ab1';
    my_employee.secondname := 'ab2';
    DBE_OUTPUT.PRINT_LINE(my_employee.firstname);
    DBE_OUTPUT.PRINT_LINE(my_employee.secondname);
END;
/
```


版权所有 © 华为云计算技术有限公司

说明

- 多个CN的环境下，存储过程中无法声明临时表的%ROWTYPE及%TYPE属性。因为临时表仅在当前session有效，在编译阶段其他CN无法看到当前CN的临时表。故多个CN的环境下，会提示该临时表不存在。
- %TYPE不支持引用非跨PACKAGE的record类型的变量、不支持引用cursor变量的某列类型等。
- %ROWTYPE不支持引用复合类型或RECORD类型变量的类型。
- 不支持非PACKAGE的record变量%TYPE作为变量类型或者出入参类型。
- 不支持view%ROWTYPE、SCHEMA.view%ROWTYPE作为出入参类型。
- 不支持PACKAGE.record变量%TYPE、SCHEMA.PACKAGE.record变量%TYPE作为出入参类型。
- 不支持表/视图.column.column%TYPE、SCHEMA.表/视图.column.column%TYPE嵌套1层及以上，作为变量类型和或者出入参类型。
- 不支持record变量.column.column%TYPE、PACKAGE.record变量.column.column%TYPE嵌套1层及以上的record的某列类型，作为变量类型和或者出入参类型。
- 不支持PACKAGE.cursor变量%ROWTYPE、SCHEMA.PACKAGE.cursor变量%ROWTYPE等更多级的cursor变量作为参数类型或者出入参类型。
- 在匿名块中，游标的%ROWTYPE（即cursor_name%ROWTYPE）不包含列信息，在作为集合元素的基类型使用时，可能会出现异常，例如BULK COLLECT INTO。

变量作用域

变量的作用域定义了变量在代码块中的可访问性及可用性。变量仅在其作用域范围内有效。

- 变量必须在DECLARE部分声明，即必须建立BEGIN-……END块结构。该结构强制要求变量必须先声明后使用，即变量在过程中具有不同的作用域与生存期。
- 同一变量在不同作用域内可重复定义，内层作用域的定义会覆盖外层作用域的定义。
- 外部块中定义的变量可在嵌套块中使用，但嵌套块的变量无法被外部块访问。

示例

```
gaussdb=# DECLARE
    emp_id  INTEGER :=7788;  --定义变量并赋值
    outer_var  INTEGER :=6688;  --定义变量并赋值
BEGIN
    DECLARE
        emp_id INTEGER :=7799;  --定义变量并赋值
        inner_var  INTEGER :=6688;  --定义变量并赋值
    BEGIN
        dbe_output.print_line('inner emp_id ='||emp_id);  --显示值为7799
        dbe_output.print_line('outer_var ='||outer_var);  --引用外部块的变量
    END;
    dbe_output.print_line('outer emp_id ='||emp_id);  --显示值为7788
END;
/
```

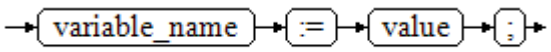
2.6.2 赋值语句

变量语法

给变量赋值的语法如图2-8所示。

版权所有 © 华为云计算技术有限公司

图 2-8 assignment_value::=



对以上语法格式的解释如下：

- variable_name，为变量名。
- value，可以是值或表达式。值value的类型需与变量variable_name的类型兼容才能正确赋值。

变量赋值示例

```
DECLARE
    emp_id  INTEGER := 7788;--赋值
BEGIN
    emp_id := 5;--赋值
    DBE_OUTPUT.PRINT_LINE(emp_id);
    emp_id := 5*7784;
    DBE_OUTPUT.PRINT_LINE(emp_id);
END;
/
--结果如下：
5
38920
ANONYMOUS BLOCK EXECUTE
```

说明

在打开GUC参数（即SET behavior_compat_options = 'proc_outparam_override';）后，赋值语句的值value不支持返回类型为复合类型或数组类型的重载函数以引用字段的形式调用（例如func(x).a或func(x)[1]）。

INTO/BULK COLLECT INTO

将存储过程内语句返回的值存储到变量内，BULK COLLECT INTO支持将部分或全部返回值临时存储至数组中。

语法格式

```
SELECT select_expressions INTO [STRICT] target FROM ...
SELECT INTO [STRICT] target [FROM ..]
EXECUTE [IMMEDIATE] select_expressions BULK COLLECT INTO target ...
```

对以上语法格式的解释如下：

- select_expressions：查询的SQL语句。通过基础SQL命令加INTO子句可以将单行或多列的结果赋值给一个变量（记录、行类型和标量变量列表）。
- target ： 目标变量。可以是一个记录变量、一个行变量或一个有逗号分隔的简单变量和记录/行域列表。
- STRICT：可选。在开启参数set behavior_compat_options = 'select_into_return_null'的前提下（默认未开启），若指定该选项，则该查询结果必须返回一行结果集，且该行不为空，否则会报错。报错信息可能是NO_DATA_FOUND（没有行）、TOO_MANY_ROWS（多于一行）或QUERY_RETURNED_NO_ROWS（没有数据返回）。若不指定该选项，则没有该限定，且支持返回空结果集。

版权所有 © 华为云计算技术有限公司

说明

- BULK COLLECT INTO只支持批量赋值给数组或集合类型，合理使用LIMIT子句避免操作过多数据导致性能下降。
- INTO/BULK COLLECT INTO只支持4层以下Record类型直接嵌套。
- 返回空结果集需要数据库初始化使用PG兼容参数，配置GUC参数behavior_compat_options = 'select_into_return_null' 为开启。配置GUC参数behavior_compat_options = ''则关闭。
- 对于数组变量，小括号()将优先识别为下标，因此对于带括号的表达式，不支持写在数组变量后面。如对于SELECT (1+3) INTO va(5)，不支持写为SELECT INTO va(5) (1+3)或SELECT INTO va[5] (1+3)。
- BULK COLLECT INTO 只支持在ORA兼容性数据库下使用。
- IMMEDIATE关键字仅用作语法兼容，无实际意义。
- 存储过程内SELECT INTO数据放入指定的输出变量中，当输出变量的个数大于实际查询结果集合的数据列时，会给多出的变量赋空值，不会产生报错；当输出变量的个数小于实际游标查询结果集合的数据列时，多余的列将被静默丢弃，不会产生报错。

示例

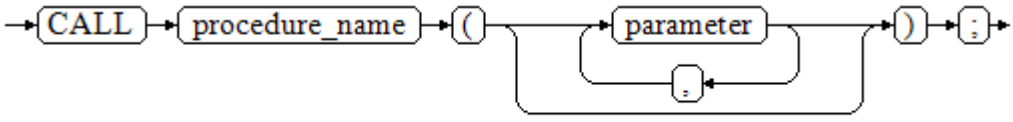
```
gaussdb=# DROP TABLE IF EXISTS customers;
gaussdb=# CREATE TABLE customers(id int,name varchar);
gaussdb=# INSERT INTO customers VALUES(1,'ab');
gaussdb=# DECLARE
    my_id integer;
BEGIN
    select id into my_id from customers limit 1; -- 赋值
END;
/
ANONYMOUS BLOCK EXECUTE
gaussdb=# DECLARE
    type id_list is varray(6) of customers.id%type;
    id_arr id_list;
BEGIN
    select id bulk collect into id_arr from customers order by id DESC limit 20; -- 批量赋值
END;
/
ANONYMOUS BLOCK EXECUTE
gaussdb=#  DECLARE
    TYPE id_list IS varray(6) OF customers.id%type;
    id_arr id_list;
    sql_qry varchar2(150);
BEGIN
    sql_qry := 'SELECT id FROM customers ORDER BY id DESC LIMIT 20';
    EXECUTE IMMEDIATE sql_qry BULK COLLECT INTO id_arr; -- 批量赋值
END;
/
ANONYMOUS BLOCK EXECUTE
```

2.6.3 调用语句

语法

调用语句的语法请参见图2-9。

图 2-9 call_clause::=



版权所有 © 华为云计算技术有限公司

对以上语法格式的解释如下：

- procedure_name，为存储过程名称。
- parameter，为存储过程的参数，可以没有或有多个参数。

示例

```
--建表
gaussdb=# CREATE SCHEMA hr;
gaussdb=# SET CURRENT_SCHEMA = hr;
gaussdb=# CREATE TABLE staffs
(
  section_id INTEGER,
  salary INTEGER
);
gaussdb=# INSERT INTO staffs VALUES (30, 10);
gaussdb=# INSERT INTO staffs VALUES (30, 20);

--创建存储过程proc_staffs
gaussdb=# CREATE OR REPLACE PROCEDURE proc_staffs
(
section     NUMBER(6),
salary_sum out NUMBER(8,2),
staffs_count out INTEGER
)
IS
BEGIN
SELECT sum(salary), count(*) INTO salary_sum, staffs_count FROM hr.staffs where section_id = section;
END;
/
CREATE PROCEDURE
--创建存储过程proc_return.
gaussdb=# CREATE OR REPLACE PROCEDURE proc_return
AS
v_num NUMBER(8,2);
v_sum INTEGER;
BEGIN
proc_staffs(30, v_sum, v_num);  --调用语句
dbe_output.print_line(v_sum||'#'||v_num);
RETURN;  --返回语句
END;
/
CREATE PROCEDURE
--调用存储过程proc_return.
gaussdb=# CALL proc_return();
30#2.00
 proc_return
-------------

(1 row)

--清除存储过程
gaussdb=# DROP PROCEDURE proc_staffs;
gaussdb=# DROP PROCEDURE proc_return;

--创建函数func_return.
gaussdb=# CREATE OR REPLACE FUNCTION func_return returns void
language plpgsql
AS $$
DECLARE
v_num INTEGER := 1;
BEGIN
dbe_output.print_line(v_num);
RETURN;  --返回语句
END $$;
CREATE FUNCTION

-- 调用函数func_return
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL func_return();
1
 func_return
-------------

(1 row)

-- 清除函数
gaussdb=# DROP FUNCTION func_return;

-- 清除当前数据库模式
gaussdb=# DROP SCHEMA hr CASCADE;
```

2.7 动态语句

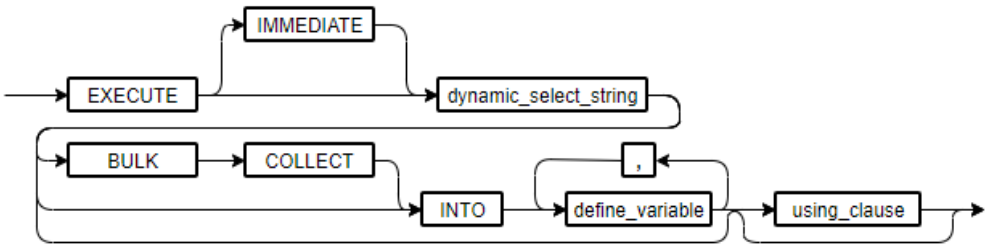
2.7.1 执行动态查询语句

本节介绍动态查询语句及其相关执行示例。GaussDB支持使用EXECUTE IMMEDIATE、OPEN FOR两种方式实现动态查询。前者通过动态执行SELECT语句，后者结合游标使用。当需要将查询结果保存在数据集中用于提取时，可使用OPEN FOR实现动态查询。

EXECUTE IMMEDIATE

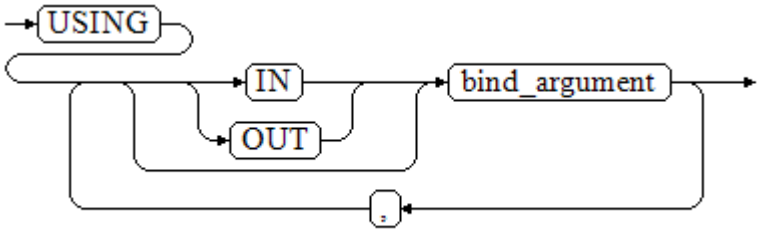
语法图请参见图2-10。

图 2-10 EXECUTE IMMEDIATE dynamic_select_clause::=



using_clause子句的语法图参见图2-11。

图 2-11 using_clause::=



对以上语法格式的解释如下：

版权所有 © 华为云计算技术有限公司

- define_variable：用于指定存放查询结果的变量。
- USING IN bind_argument：用于指定存放传递给动态SQL值的变量，即在dynamic_select_string中存在占位符时使用。
- USING OUT bind_argument：用于指定存放动态SQL返回值的变量。

须知

- 查询语句中，into和out不能同时存在。
- 占位符命名以“:”开始，后面可跟数字、字符或字符串（不能使用带引号的数字、字符或字符串），与USING子句的bind_argument一一对应。
- bind_argument只能是值、变量或表达式，不能是表名、列名、数据类型等数据库对象，即不支持使用bind_argument为动态SQL语句传递模式对象。如果存储过程需要通过声明参数传递数据库对象来构造动态SQL语句（常见于执行DDL语句时），建议采用连接运算符“||”拼接dynamic_select_clause。
- 动态PL/SQL块允许出现重复的占位符，即相同占位符只能与USING子句的一个bind_argument按位置对应。当设置GUC参数behavior_compat_options值为dynamic_sql_compat时，会按照占位符的顺序依次匹配USING子句的bind_argument，重复的占位符不再识别为同一个占位符。
- IMMEDIATE关键字仅用作语法兼容，无实际意义。

示例

```
gaussdb=# DROP SCHEMA IF EXISTS hr CASCADE;
gaussdb=# CREATE SCHEMA hr;
gaussdb=# SET CURRENT_SCHEMA = hr;
gaussdb=# CREATE TABLE staffs
(
  staff_id NUMBER,
  first_name VARCHAR2,
  salary NUMBER
);
gaussdb=# INSERT INTO staffs VALUES (200, 'mike', 5800);
gaussdb=# INSERT INTO staffs VALUES (201, 'lily', 3000);
gaussdb=# INSERT INTO staffs VALUES (202, 'john', 4400);

--从动态语句检索值（INTO 子句）：
gaussdb=# DECLARE
   staff_count  VARCHAR2(20);
BEGIN
   EXECUTE IMMEDIATE 'select count(*) from hr.staffs'
      INTO staff_count;
   dbe_output.print_line(staff_count);
END;
/
3
ANONYMOUS BLOCK EXECUTE
--传递并检索值（INTO子句用在USING子句前）：
gaussdb=# CREATE OR REPLACE PROCEDURE dynamic_proc
AS
   staff_id     NUMBER(6) := 200;
   first_name   VARCHAR2(20);
   salary       NUMBER(8,2);
BEGIN
   EXECUTE IMMEDIATE 'select first_name, salary from hr.staffs where staff_id = :1'
      INTO first_name, salary
      USING IN staff_id;
   dbe_output.print_line(first_name || ' ' || salary);
END;
/
```


版权所有 © 华为云计算技术有限公司

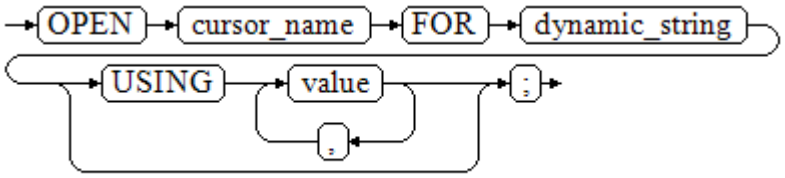
```
CREATE PROCEDURE
--调用存储过程
gaussdb=# CALL dynamic_proc();
mike 5800.00
 dynamic_proc
--------------

(1 row)

--删除存储过程
gaussdb=# DROP PROCEDURE dynamic_proc;

-- 清除当前数据库模式
gaussdb=# DROP SCHEMA hr CASCADE;
```

OPEN FOR

动态查询语句可以使用OPEN FOR打开动态游标来执行。

语法参见**图2-12**。

图 2-12 open_for::=



参数说明：

- cursor_name：要打开的游标名。
- dynamic_string：动态查询语句。
- USING value：在dynamic_string中存在占位符时使用。

游标的使用请参考**游标**。

示例

```
gaussdb=# CREATE SCHEMA hr;
gaussdb=# SET CURRENT_SCHEMA = hr;
gaussdb=# DROP TABLE IF EXISTS staffs;
gaussdb=# CREATE TABLE staffs
(
  section_id NUMBER,
  first_name VARCHAR2,
  phone_number VARCHAR2,
  salary NUMBER
  );
gaussdb=# INSERT INTO staffs VALUES (30, 'mike', '13567829252', 5800);
gaussdb=# INSERT INTO staffs VALUES (40, 'john', '17896354637', 4000);

gaussdb=# DECLARE
    name          VARCHAR2(20);
    phone_number  VARCHAR2(20);
    salary        NUMBER(8,2);
    sqlstr        VARCHAR2(1024);

    TYPE app_ref_cur_type IS REF CURSOR;  --定义游标类型
    my_cur app_ref_cur_type;  --定义游标变量

BEGIN
```


版权所有 © 华为云计算技术有限公司

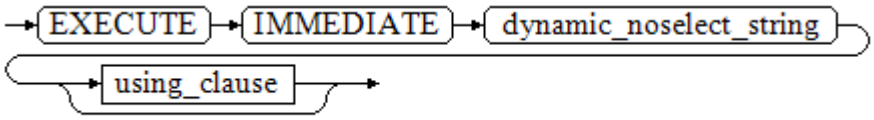
```
    sqlstr := 'select first_name,phone_number,salary from hr.staffs
        where section_id = :1';
    OPEN my_cur FOR sqlstr USING '30';  --打开游标, using是可选的
    FETCH my_cur INTO name, phone_number, salary; --获取数据
    WHILE my_cur%FOUND LOOP
        dbe_output.print_line(name||'#'||phone_number||'#'||salary);
        FETCH my_cur INTO name, phone_number, salary;
    END LOOP;
    CLOSE my_cur;   --关闭游标
END;
/
mike#13567829252#5800.00
ANONYMOUS BLOCK EXECUTE
gaussdb=# DROP TABLE staffs;
```

2.7.2 执行动态非查询语句

语法

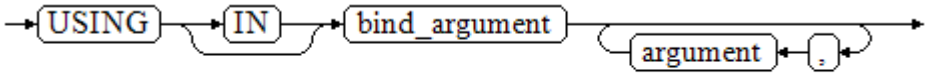
语法请参见图2-13。

图 2-13 noselect::=



using_clause子句的语法参见图2-14。

图 2-14 using_clause::=



对以上语法格式的解释如下：

USING IN bind_argument：用于指定存放传递给动态SQL值的变量，在dynamic_noselect_string中存在占位符时使用，即动态SQL语句执行时，bind_argument将替换相对应的占位符。

须知

- bind_argument只能是值、变量或表达式，不能是表名、列名、数据类型等数据库对象。如果存储过程需要通过声明参数传递数据库对象来构造动态SQL语句（常见于执行DDL语句时），建议采用连接运算符“||”拼接dynamic_select_clause。
- 动态语句允许出现重复的占位符，相同占位符只能与唯一一个bind_argument按位置一一对应。当设置GUC参数behavior_compat_options值为dynamic_sql_compat时，会按照占位符的顺序依次匹配USING子句的bind_argument，重复的占位符不再识别为同一个占位符（占位符名不能使用带引号的数字、字符或字符串）。

版权所有 © 华为云计算技术有限公司

示例

```
--创建表
gaussdb=# CREATE TABLE sections_t1
(
   section       NUMBER(4) ,
   section_name  VARCHAR2(30),
   manager_id    NUMBER(6),
   place_id      NUMBER(4)
)
DISTRIBUTE BY hash(manager_id);

--声明变量
gaussdb=# DECLARE
   section       NUMBER(4) := 280;
   section_name  VARCHAR2(30) := 'Info support';
   manager_id    NUMBER(6) := 103;
   place_id      NUMBER(4) := 1400;
   new_colname   VARCHAR2(10) := 'sec_name';
BEGIN
--执行查询
    EXECUTE IMMEDIATE 'insert into sections_t1 values(:1, :2, :3, :4)'
       USING section, section_name, manager_id,place_id;
--执行查询（重复占位符）
    EXECUTE IMMEDIATE 'insert into sections_t1 values(:1, :2, :3, :1)'
       USING section, section_name, manager_id;
--执行ALTER语句（建议采用“||”拼接数据库对象构造DDL语句）
    EXECUTE IMMEDIATE 'alter table sections_t1 rename section_name to ' || new_colname;
END;
/
ANONYMOUS BLOCK EXECUTE
--查询数据
gaussdb=# SELECT * FROM sections_t1;
 section |   sec_name   | manager_id | place_id
---------+--------------+------------+----------
     280 | Info support |        103 |     1400
     280 | Info support |        103 |      280
(2 rows)

--删除表
gaussdb=# DROP TABLE sections_t1;
```

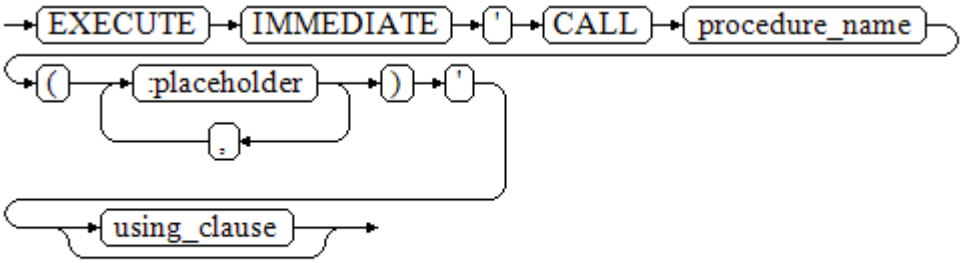
2.7.3 动态调用存储过程

动态调用存储过程时，必须使用匿名语句块将存储过程或语句块包裹。通过EXECUTE IMMEDIATE…USING语句配合IN、OUT实现参数输入输出。

语法

语法请参见图2-15。

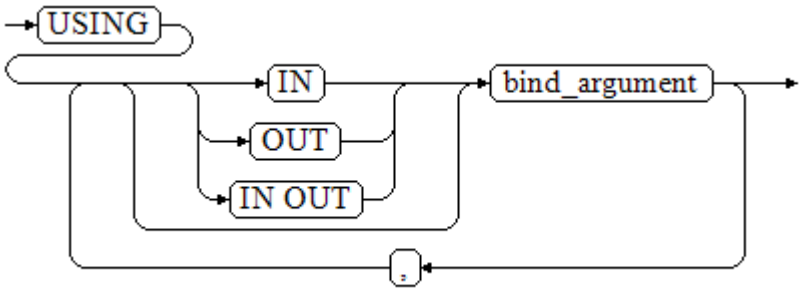
图 2-15 call_procedure::=



版权所有 © 华为云计算技术有限公司

using_clause子句的语法参见图2-16。

图 2-16 using_clause::=



对以上语法格式的解释如下：

- CALL procedure_name：调用存储过程。
- [:placeholder1，:placeholder2，…]：存储过程参数占位符列表，占位符名不能使用带引号的数字、字符或字符串。占位符个数与参数个数相同。
- USING [IN|OUT|IN OUT] bind_argument：用于指定存放传递给存储过程参数值的变量。bind_argument前的修饰符与对应参数的修饰符一致。

不支持调用存储过程时使用=>跳过参数进行调用。

示例

```
--创建存储过程proc_add。
gaussdb=# CREATE OR REPLACE PROCEDURE proc_add
(
    param1    in   INTEGER,
    param2    out  INTEGER,
    param3    in   INTEGER
)
AS
BEGIN
   param2:= param1 + param3;
END;
/

gaussdb=# DECLARE
    input1 INTEGER:=1;
    input2 INTEGER:=2;
    statement  VARCHAR2(200);
    param2     INTEGER;
BEGIN
   --声明调用语句
    statement := 'call proc_add(:col_1, :col_2, :col_3)';
    --执行语句
    EXECUTE IMMEDIATE statement
        USING IN input1, OUT param2, IN input2;
    dbe_output.print_line('result is: '||to_char(param2));
END;
/
result is: 3
ANONYMOUS BLOCK EXECUTE
--删除存储过程
gaussdb=# DROP PROCEDURE proc_add;
```


版权所有 © 华为云计算技术有限公司

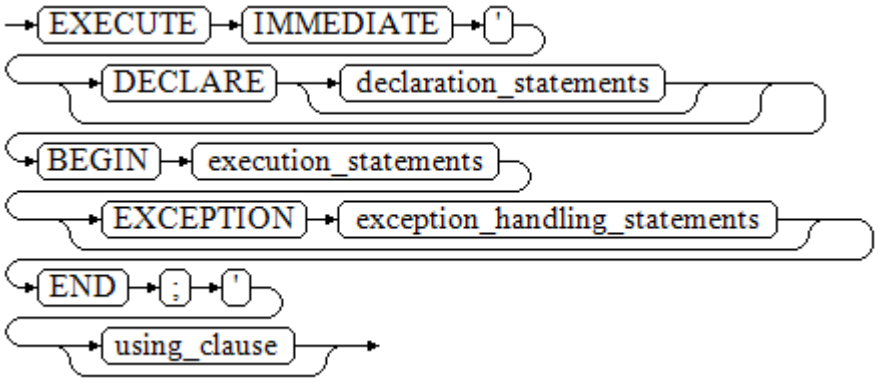
2.7.4 动态调用匿名块

动态调用匿名块是指在动态语句中执行匿名块。通过EXECUTE IMMEDIATE…USING语句配合IN、OUT实现参数输入输出。

语法

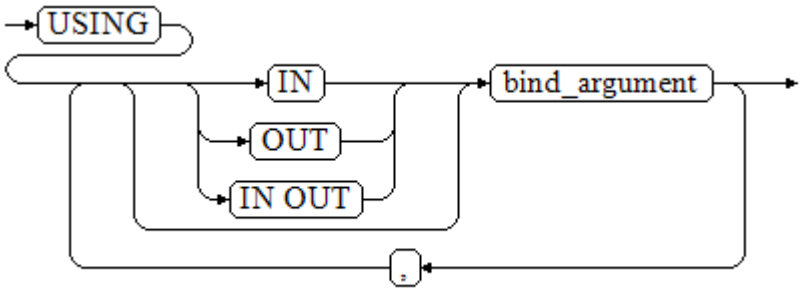
动态调用匿名块语法如图2-17所示。

图 2-17 call_anonymous_block::=



using_clause子句的语法如图2-18所示。

图 2-18 using_clause::=



对以上语法格式的解释如下：

- 匿名块程序的实施部分以BEGIN语句开始，以END语句结束，以分号终止。
- USING [IN|OUT|IN OUT] bind_argument，用于指定存放传递给存储过程参数值的变量。bind_argument前的修饰符与对应参数的修饰符一致。
- 匿名块中的输入输出参数使用占位符标识，要求占位符数量与参数数量一致，且占位符对应参数的顺序须与USING子句中参数的顺序保持一致。

版权所有 © 华为云计算技术有限公司

须知

- 仅支持匿名块中调用SQL语句和存储过程时绑定参数，其余绑定参数场景皆不支持。例如：匿名块中使用表达式以及cursor等、匿名块中嵌套调用动态语句。
- 不支持匿名块中SELECT INTO语句调用含有出参的FUNCTION/PROCEDURE时，绑定出参。
- 不支持匿名块绑定复合类型的出参。
- 不支持同一条语句同时使用匿名块内声明的变量和绑定参数。
- 不支持绑定参数时使用PERFORM关键字调用存储过程。
- 调用存储过程时，仅支持绑定参数直接作为出入参，不支持表达式形式（如“1+:va”）作为出入参，并且占位符名不能使用带引号的数字、字符或字符串。
- IMMEDIATE关键字仅用作语法兼容，无实际意义。
- 绑定入参类型为refcursor时，存储过程内的修改与入参相互隔离。
- 打开dynamic_sql_check参数时，占位符个数与参数个数一致时使用同名占位符作为匿名块参数会报错，需修改为不同名参数，详见**示例**。

示例

```
gaussdb=# DROP SCHEMA IF EXISTS hr CASCADE;
gaussdb=# CREATE SCHEMA hr;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = hr;
SET
gaussdb=# CREATE TABLE staffs
(
  staff_id NUMBER,
  first_name VARCHAR2,
  salary NUMBER
);
CREATE TABLE
gaussdb=# INSERT INTO staffs VALUES (200, 'mike', 5800);
INSERT 0 1
gaussdb=# INSERT INTO staffs VALUES (201, 'lily', 3000);
INSERT 0 1
gaussdb=# INSERT INTO staffs VALUES (202, 'john', 4400);
INSERT 0 1

--创建重载函数
gaussdb=# CREATE OR REPLACE PACKAGE pkg1
IS
   PROCEDURE plus(var1 in int, var2 int, var3 out int);
   PROCEDURE plus(var1 in out int);
END pkg1;
/
CREATE PACKAGE
gaussdb=# CREATE OR REPLACE PACKAGE BODY pkg1 IS
   PROCEDURE plus(var1 in int, var2 int, var3 out int)
   AS
   BEGIN
      var3 = var1 + var2 + 1;
   END;
   PROCEDURE plus(var1 in out int)
   AS
   BEGIN
      var1 = var1 + 1;
   END;
END pkg1;
/
CREATE PACKAGE BODY
```


版权所有 © 华为云计算技术有限公司

```
--创建存储过程dynamic_proc
gaussdb=# CREATE OR REPLACE PROCEDURE dynamic_proc
AS
   staff_id     NUMBER(6) := 200;
   first_name   VARCHAR2(20);
   salary       NUMBER(8,2);
BEGIN
    --执行匿名块
    EXECUTE IMMEDIATE 'begin select first_name, salary into :first_name, :salary from hr.staffs where
staff_id= :dno; end;'
       USING OUT first_name, OUT salary, IN staff_id;
   dbe_output.print_line(first_name|| ' ' || salary);
END;
/
CREATE PROCEDURE
--创建存储过程调用重载函数
gaussdb=# CREATE OR REPLACE PROCEDURE dynamic_proc1
AS
     v_sql  VARCHAR2(200);
     var1   NUMBER(6) := 1;
     var2   NUMBER(6) := 2;
     var3   NUMBER(6);
BEGIN
     v_sql := 'begin pkg1.plus(:1, :2, :3); end;';
     EXECUTE IMMEDIATE v_sql USING var1, var2, out var3;
     dbe_output.print_line('var3: ' || var3);
END;
/
CREATE PROCEDURE

--调用存储过程
gaussdb=# CALL dynamic_proc();
mike 5800.00
 dynamic_proc
--------------

(1 row)
gaussdb=# CALL dynamic_proc1();
var3: 4
 dynamic_proc1
---------------

(1 row)
--删除存储过程
gaussdb=# DROP PROCEDURE dynamic_proc;
DROP PROCEDURE
gaussdb=# DROP PROCEDURE dynamic_proc1;
DROP PROCEDURE
--开启dynamic_sql_check时报错示例
gaussdb=# SET behavior_compat_options = 'dynamic_sql_check';
SET
gaussdb=# CREATE OR REPLACE PROCEDURE test_proc_exception001(a out integer, b inout integer, c
integer)
as
BEGIN
 a := 1;
 begin b := 1/0; end;
EXCEPTION
 WHEN others THEN
  b := 2;
END;
/
CREATE PROCEDURE
gaussdb=#
DECLARE
 a integer := 1;
 c integer;
BEGIN
```


版权所有 © 华为云计算技术有限公司

```
  execute immediate 'begin test_proc_exception001(:1,:2,:1); end;' using in out a, out c, a;
END;
/
ERROR:  The numbers of parameters do not match in Dynamic SQL, expected number of parameters : 3 , actual number of SQL parameters : 2.
CONTEXT:  PL/pgSQL function inline_code_block line 4 at EXECUTE statement
--修改同名占位符
gaussdb=#
DECLARE
 a integer := 1;
 c integer;
BEGIN
  execute immediate 'begin test_proc_exception001(:1,:2,:3); end;' using in out a, out c, a;
END;
/
ANONYMOUS BLOCK EXECUTE
gaussdb=# DROP PROCEDURE test_proc_exception001;
DROP PROCEDURE
```

2.8 控制语句

2.8.1 返回语句

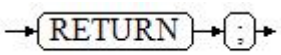
GaussDB提供两种方式返回数据：RETURN或RETURN NEXT及RETURN QUERY。其中，RETURN NEXT和RETURN QUERY只适用于函数，不适用存储过程。

2.8.1.1 RETURN

语法

返回语句的语法请参见图2-19。

图 2-19 return_clause::=



对以上语法的解释如下：

用于将存储过程或函数返回结果赋值给调用者。

示例

请参见调用语句的示例。

2.8.1.2 RETURN NEXT 及 RETURN QUERY

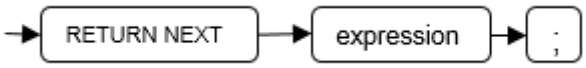
语法

创建函数时需要指定返回值SETOF datatype。

- return_next_clause::=
  return_next_clause::=语句如图2-20所示。

版权所有 © 华为云计算技术有限公司

图 2-20 return_next_clause::=



- return_query_clause::=

  return_query_clause::=语句如图2-21所示。

  图 2-21 return_query_clause::=

当需要函数返回一个集合时，使用RETURN NEXT或RETURN QUERY向结果集追加结果，然后继续执行函数下一条语句。随着后续RETURN NEXT或RETURN QUERY的执行，结果集将累积多个结果。函数执行完成后一次性返回所有结果。

RETURN NEXT可用于标量和复合数据类型。

RETURN QUERY有一种变体RETURN QUERY EXECUTE，后面还可以增加动态查询，通过USING向查询插入参数。

示例

```
gaussdb=# DROP TABLE t1;
gaussdb=# CREATE TABLE t1(a int);
gaussdb=# INSERT INTO t1 VALUES(1),(10);

--RETURN NEXT
gaussdb=# CREATE OR REPLACE FUNCTION fun_for_return_next() RETURNS SETOF t1 AS $$
DECLARE
  r t1%ROWTYPE;
BEGIN
  FOR r IN select * from t1
  LOOP
    RETURN NEXT r;
  END LOOP;
  RETURN;
END;
$$ LANGUAGE plpgsql;
gaussdb=# call fun_for_return_next();
 a
---
 1
 10
(2 rows)
-- RETURN QUERY
gaussdb=# CREATE OR REPLACE FUNCTION fun_for_return_query() RETURNS SETOF t1 AS $$
DECLARE
  r t1%ROWTYPE;
BEGIN
  RETURN QUERY select * from t1;
END;
$$
language plpgsql;
gaussdb=# call fun_for_return_query();
 a
---
 1
 10
(2 rows)
```


版权所有 © 华为云计算技术有限公司

2.8.2 条件语句

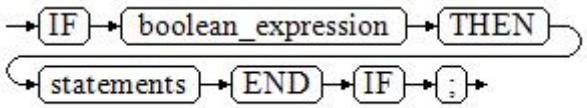
条件语句的主要作用判断参数或者语句是否满足已给定的条件，根据判定结果执行相应的操作。

GaussDB有五种形式的IF：

- IF_THEN

  IF_THEN::=语句如图2-22所示。

  图 2-22 IF_THEN::=

  IF_THEN语句是IF的最简单形式。如果条件为真，statements将被执行。否则，将忽略它们的结果使该IF_THEN语句执行结束。

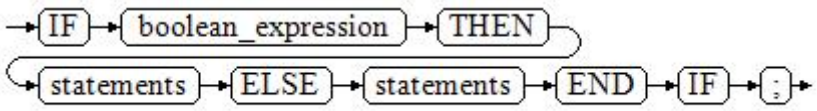
  示例

  ```
  gaussdb=# DECLARE
      v_user_id integer default 1;
  BEGIN
  IF v_user_id <> 0 THEN
      raise info 'v_user_id is NOT 0';
  END IF;
  END;
  /
  INFO:  v_user_id is NOT 0
  ```

- IF_THEN_ELSE

  IF_THEN_ELSE::=语句如图2-23所示。

  图 2-23 IF_THEN_ELSE::=

  IF_THEN_ELSE语句增加了ELSE的分支，可以声明在条件为假的时候执行的语句。

  示例

  ```
  gaussdb=# DECLARE
      v_user_id integer default 0;
  BEGIN
      IF v_user_id <> 0 THEN
          raise info 'v_user_id is NOT 0';
      ELSE
          raise info 'v_user_id is 0';
      END IF;
  END;
  /
  INFO:  v_user_id is 0
  ```

- IF_THEN_ELSE IF

  IF语句可以嵌套，嵌套方式如下：

版权所有 © 华为云计算技术有限公司

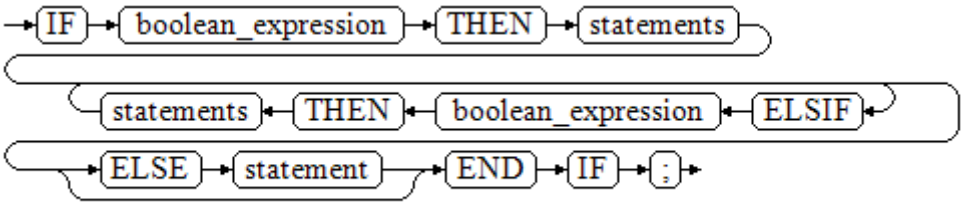
```
gaussdb=# DECLARE
    v_user_id integer default 1;
BEGIN
    IF v_user_id = 0 THEN
        raise info 'v_user_id is  0';
    ELSE
        IF v_user_id > 0 THEN
            raise info 'v_user_id > 0';
        END IF;
    END IF;
END;
/
INFO:  v_user_id > 0
```

这种形式实际上就是在一个IF语句的ELSE部分嵌套了另一个IF语句。因此需要一个END IF语句给每个嵌套的IF，另外还需要一个END IF语句结束父IF-ELSE。如果有多个选项，可使用下面的形式。

- IF_THEN_ELSIF_ELSE

  IF_THEN_ELSIF_ELSE::=语句如图2-24所示。

  图 2-24 IF_THEN_ELSIF_ELSE::=

  示例

```
gaussdb=# DECLARE
    v_user_id integer default NULL;
BEGIN
    IF v_user_id = 0 THEN
        raise info 'v_user_id is  0';
    ELSIF v_user_id > 0 THEN
        raise info 'v_user_id > 0';
    ELSIF v_user_id < 0 THEN
        raise info 'v_user_id < 0';
    ELSE
        raise info 'v_user_id is NULL';
    END IF;
END;
/
INFO:  v_user_id is NULL
```

- IF_THEN_ELSEIF_ELSE

  ELSEIF是ELSIF的别名。

  综合示例

```
gaussdb=# CREATE OR REPLACE PROCEDURE proc_control_structure(i in integer)
AS
    BEGIN
        IF i > 0 THEN
            raise info 'i:% is greater than 0. ',i;
        ELSIF i < 0 THEN
            raise info 'i:% is smaller than 0. ',i;
        ELSE
            raise info 'i:% is equal to 0. ',i;
        END IF;
        RETURN;
    END;
```


版权所有 © 华为云计算技术有限公司

```
/
CREATE PROCEDURE

gaussdb=# CALL proc_control_structure(3);
INFO:  i:3 is greater than 0.
 proc_control_structure
------------------------

(1 row)

--删除存储过程
gaussdb=# DROP PROCEDURE proc_control_structure;
DROP PROCEDURE
```

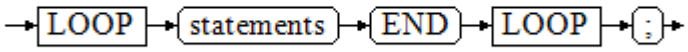
2.8.3 循环语句

简单 LOOP 语句

语法图

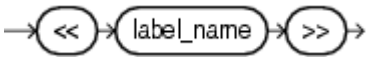
loop::=语句如图2-25所示。

图 2-25 loop::=



label declaration ::=语句如图2-26所示。

图 2-26 label declaration ::=



示例

```
gaussdb=# CREATE OR REPLACE PROCEDURE proc_loop(i in integer, count out integer)
AS
    BEGIN
        count:=0;
        LOOP
        IF count > i THEN
            raise info 'count is %. ', count;
            EXIT;
        ELSE
            count:=count+1;
        END IF;
        END LOOP;
    END;
/
CREATE PROCEDURE

gaussdb=# CALL proc_loop(10,5);
INFO:  count is 11.
 count
-------
    11
(1 row)
```


版权所有 © 华为云计算技术有限公司

须知

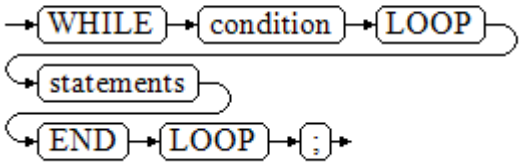
该循环必须要结合EXIT使用，否则将陷入死循环。

WHILE_LOOP 语句

语法图

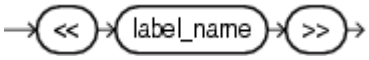
while_loop::=语句如图2-27所示。

图 2-27 while_loop::=



label declaration ::=语句如图2-28所示。

图 2-28 label declaration ::=



只要条件表达式为真，WHILE语句就会不停地在一系列语句上进行循环，在每次进入循环体的时候进行条件判断。

示例

```
gaussdb=# CREATE TABLE integertable(c1 integer) DISTRIBUTE BY hash(c1);
CREATE TABLE
gaussdb=# CREATE OR REPLACE PROCEDURE proc_while_loop(maxval in integer)
AS
    DECLARE
    i int :=1;
    BEGIN
        WHILE i < maxval LOOP
            INSERT INTO integertable VALUES(i);
            i:=i+1;
        END LOOP;
    END;
/
CREATE PROCEDURE

--调用存储过程
gaussdb=# CALL proc_while_loop(10);
 proc_while_loop
-----------------

(1 row)

--删除存储过程和表
gaussdb=# DROP PROCEDURE proc_while_loop;
DROP PROCEDURE
```


版权所有 © 华为云计算技术有限公司

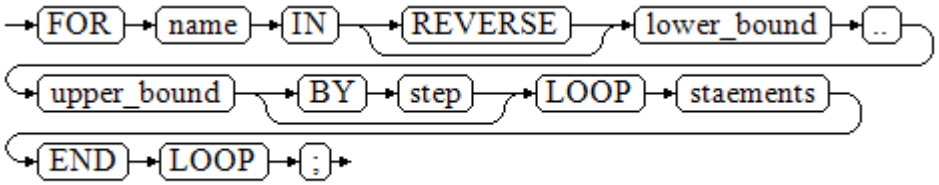
```
gaussdb=# DROP TABLE integertable;
DROP TABLE
```

FOR_LOOP（integer 变量）语句

语法图

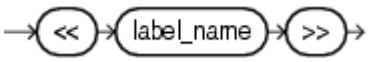
for_loop::=语句如图2-29所示。

图 2-29 for_loop::=



label declaration ::=语句如图2-30所示。

图 2-30 label declaration ::=



说明

- 变量name会自动定义为integer类型并且只在此循环里存在。变量name介于lower_bound和upper_bound之间。
- 当使用REVERSE关键字时，lower_bound必须大于等于upper_bound，否则循环体不会被执行。
- 使用FORALL + BULK COLLECT INTO时，INTO的变量仅返回单条DML语句执行结果。

示例

```
--从0到5进行循环
gaussdb=# CREATE OR REPLACE PROCEDURE proc_for_loop()
AS
    BEGIN
    FOR I IN 0..5 LOOP
        DBE_OUTPUT.PRINT_LINE('It is '||to_char(I) || ' time;') ;
    END LOOP;
END;
/
CREATE PROCEDURE

--调用存储过程
gaussdb=# CALL proc_for_loop();
It is 0 time;
It is 1 time;
It is 2 time;
It is 3 time;
It is 4 time;
It is 5 time;
 proc_for_loop
---------------

(1 row)

--删除存储过程
```


版权所有 © 华为云计算技术有限公司

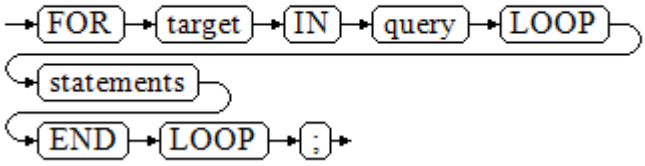
```
gaussdb=# DROP PROCEDURE proc_for_loop;
DROP PROCEDURE
```

FOR_LOOP 查询语句

语法图

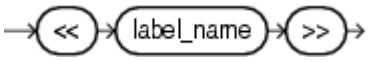
for_loop_query::=语句如图2-31所示。

图 2-31 for_loop_query::=



label declaration ::=语句如图2-32所示。

图 2-32 label declaration ::=



说明

- 变量target会自动定义，类型和query的查询结果的类型一致，并且只在此循环中有效。target的取值就是query的查询结果。
- query可以使用EXECUTE增加动态查询，通过USING向查询插入参数。详见示例中的动态查询相关内容。

示例

```
--循环输出查询结果。
gaussdb=# CREATE OR REPLACE PROCEDURE proc_for_loop_query()
AS
    record VARCHAR2(50);
BEGIN
    FOR record IN SELECT spcname FROM pg_tablespace LOOP
    dbe_output.print_line(record);
    END LOOP;
END;
/
CREATE PROCEDURE

--调用存储过程
gaussdb=# CALL proc_for_loop_query();
pg_default
pg_global
 proc_for_loop_query
---------------------

(1 row)

--删除存储过程
gaussdb=# DROP PROCEDURE proc_for_loop_query;
DROP PROCEDURE
```


版权所有 © 华为云计算技术有限公司

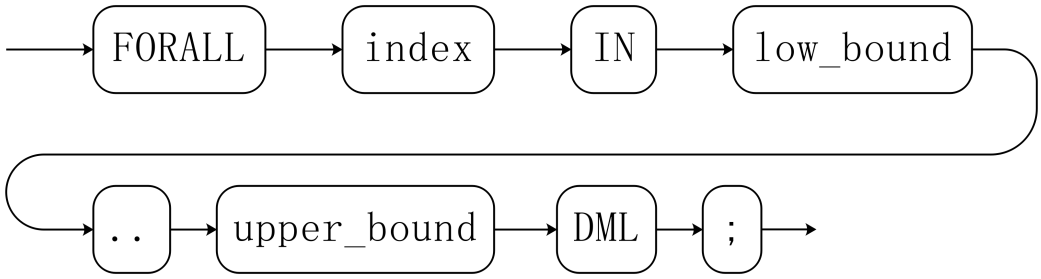
```
--动态查询
gaussdb=# CREATE TABLE t1(id int);
gaussdb=# INSERT INTO t1 VALUES(1);
gaussdb=#
DECLARE
a int := 1;
item t1%rowtype;
BEGIN
    FOR item IN EXECUTE 'SELECT * FROM t1 WHERE id = :1' USING a LOOP
        RAISE INFO '%', item;
    END LOOP;
END;
/
INFO:  (1)
ANONYMOUS BLOCK EXECUTE
gaussdb=# DROP TABLE t1;
```

FORALL 批量查询语句

语法图

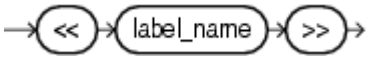
forall::=语句如图2-33所示。

图 2-33 forall::=



label declaration ::语句如图2-34所示。

图 2-34 label declaration ::=



说明

- 变量index会自动定义为integer类型并且只在此循环里存在。index的取值介于low_bound和upper_bound之间。
- FORALL循环体中只能包含DML语句。

示例

```
gaussdb=# CREATE TABLE TEST_t1 (
  title NUMBER(6),
  did VARCHAR2(20),
  data_period VARCHAR2(25),
  kind VARCHAR2(25),
  interval VARCHAR2(20),
  time DATE,
  isModified VARCHAR2(10)
)
DISTRIBUTE BY hash(did);
CREATE TABLE
```


版权所有 © 华为云计算技术有限公司

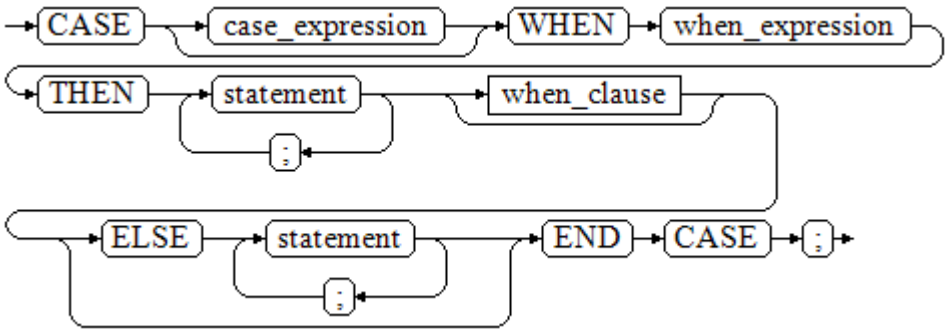
```
gaussdb=# INSERT INTO TEST_t1 VALUES( 8, 'Donald', 'OConnell', 'DOCONNEL', '650.507.9833',
to_date('21-06-1999', 'dd-mm-yyyy'), 'SH_CLERK' );
INSERT 0 1
gaussdb=# CREATE OR REPLACE PROCEDURE proc_forall()
AS
BEGIN
    FORALL i IN 100..120
        update TEST_t1 set title = title + 100*i;
END;
/
CREATE PROCEDURE

--调用存储过程
gaussdb=# CALL proc_forall();
 proc_forall
-------------

(1 row)

--查询存储过程调用结果
gaussdb=# SELECT * FROM TEST_t1;
 title  |  did   | data_period |   kind   |   interval   |        time         | ismodified
--------+--------+-------------+----------+--------------+---------------------+------------
 231008 | Donald | OConnell    | DOCONNEL | 650.507.9833 | 1999-06-21 00:00:00 | SH_CLERK
(1 row)

--删除存储过程和表
gaussdb=# DROP PROCEDURE proc_forall;
DROP PROCEDURE

gaussdb=# DROP TABLE TEST_t1;
DROP TABLE
```

2.8.4 分支语句

语法

分支语句的语法请参见图2-35。

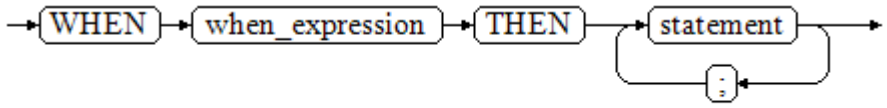
图 2-35 case_when::=



when_clause子句的语法图参见图2-36。

版权所有 © 华为云计算技术有限公司

图 2-36 when_clause::=



参数说明：

- case_expression：变量或表达式。
- when_expression：常量或者条件表达式。
- statement：执行语句。

示例

```
gaussdb=# CREATE OR REPLACE PROCEDURE proc_case_branch(pi_result in integer, pi_return out integer)
AS
   BEGIN
      CASE pi_result
         WHEN 1 THEN
            pi_return := 111;
         WHEN 2 THEN
            pi_return := 222;
         WHEN 3 THEN
            pi_return := 333;
         WHEN 6 THEN
            pi_return := 444;
         WHEN 7 THEN
            pi_return := 555;
         WHEN 8 THEN
            pi_return := 666;
         WHEN 9 THEN
            pi_return := 777;
         WHEN 10 THEN
            pi_return := 888;
         ELSE
            pi_return := 999;
      END CASE;
      raise info 'pi_return : %',pi_return ;
END;
/
CREATE PROCEDURE

gaussdb=# CALL proc_case_branch(3,0);
INFO:  pi_return : 333
 pi_return
-----------
       333
(1 row)

--删除存储过程
gaussdb=# DROP PROCEDURE proc_case_branch;
DROP PROCEDURE
```

2.8.5 空语句

在PL/SQL程序中，可以用NULL语句来说明“不用做任何事情”，相当于一个占位符，可以使某些语句变得有意义，提高程序的可读性。

语法

空语句的用法如下：

版权所有 © 华为云计算技术有限公司

```
DECLARE
    ...
BEGIN
    ...
    IF v_num IS NULL THEN
        NULL; -- 不需要处理任何数据。
    END IF;
END;
/
```

参数说明：

- v_num：变量或表达式。

示例

```
gaussdb=# DECLARE
    v_num integer default NULL;
BEGIN
    IF v_num IS NOT NULL THEN
        raise info 'v_num is NULL';
    ELSE
        NULL; -- 不需要处理任何数据。
    END IF;
END;
/
ANONYMOUS BLOCK EXECUTE
```

2.8.6 错误捕获语句

缺省时，当PL/SQL函数执行过程中发生错误时退出函数执行，并且周围的事务也会回滚。可以用一个带有EXCEPTION子句的BEGIN块捕获错误并且从中恢复。其语法是正常的BEGIN块语法的一个扩展：

```
[<<label>>]
[DECLARE
    declarations]
BEGIN
    statements
EXCEPTION
    WHEN condition [OR condition ...] THEN
        handler_statements
    [WHEN condition [OR condition ...] THEN
        handler_statements
    ...]
END;
```

如果没有发生错误，这种形式的包围块只是简单地执行所有语句，然后转到END之后的下一个语句。但是如果在执行的语句内部发生了一个错误，则这个语句将会回滚，然后转到EXCEPTION列表，寻找匹配错误的第一个条件。若找到匹配，则执行对应的handler_statements，然后转到END之后的下一个语句。如果没有找到匹配，则会向事务的外层报告错误，和没有EXCEPTION子句一样。错误码可以捕获同一类的其他错误码。

也就是说该错误可以被一个包围块用EXCEPTION捕获，如果没有包围块，则进行退出函数处理。

condition的名称可以是SQL标准错误码编号说明的任意值。特殊的条件名OTHERS匹配除了QUERY_CANCELED之外的所有错误类型。

如果在选中的handler_statements里发生了新错误，则不能被这个EXCEPTION子句捕获，而是向事务的外层报告错误。一个外层的EXCEPTION子句可以捕获它。

如果一个错误被EXCEPTION捕获，PL/SQL函数的局部变量保持错误发生时的原值，但是所有该块中想写入数据库中的状态都回滚。

 版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CREATE TABLE mytab(id INT,firstname VARCHAR(20),lastname VARCHAR(20)) DISTRIBUTE BY
hash(id);
CREATE TABLE

gaussdb=# INSERT INTO mytab(firstname, lastname) VALUES('Tom', 'Jones');
INSERT 0 1

gaussdb=# CREATE FUNCTION fun_exp() RETURNS INT
AS $$
DECLARE
    x INT :=0;
    y INT;
BEGIN
    UPDATE mytab SET firstname = 'Joe' WHERE lastname = 'Jones';
    x := x + 1;
    y := x / 0;
EXCEPTION
    WHEN division_by_zero THEN
        RAISE NOTICE 'caught division_by_zero';
        RETURN x;
END;$$
LANGUAGE plpgsql;
CREATE FUNCTION

gaussdb=# call fun_exp();
NOTICE:  caught division_by_zero
 fun_exp
---------
       1
(1 row)

gaussdb=# select * from mytab;
 id | firstname | lastname
----+-----------+----------
    | Tom       | Jones
(1 row)

gaussdb=# DROP FUNCTION fun_exp();
DROP FUNCTION

gaussdb=# DROP TABLE mytab;
DROP TABLE
```

当控制到达给y赋值的地方时，会有一个division_by_zero错误失败。这个错误将被EXCEPTION子句捕获。而在RETURN语句里返回的数值将是x的增量值。

说明

- 进入和退出一个包含EXCEPTION子句的块要比不包含的块开销大的多。因此，不必要的时候不要使用EXCEPTION。
- 在下列场景中，无法捕获处理异常，整个存储过程回滚：1. 节点故障，2. 网络故障引起的存储过程参与节点线程退出以及COPY FROM操作中源数据与目标表的表结构不一致造成的异常，3. EXCEPTION清理子事务时的异常（对于包含EXCEPTION语句的存储过程，执行时会开启一个隐式的子事务，正常执行完毕statements时，会自动清理此子事务，清理过程中可能会由于内存管控发生异常）。

示例： UPDATE/INSERT异常

这个例子根据使用异常处理器执行恰当的UPDATE或INSERT 。

```
gaussdb=# CREATE TABLE db (a INT, b TEXT);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# CREATE FUNCTION merge_db(key INT, data TEXT) RETURNS VOID AS
```


版权所有 © 华为云计算技术有限公司

```
$$
BEGIN
    LOOP

--第一次尝试更新key
        UPDATE db SET b = data WHERE a = key;
        IF found THEN
            RETURN;
        END IF;
--不存在，所以尝试插入key，如果其他人同时插入相同的key，可能得到唯一key失败。
        BEGIN
            INSERT INTO db(a,b) VALUES (key, data);
            RETURN;
        EXCEPTION WHEN unique_violation THEN
        --什么也不做，并且循环尝试再次更新。
        END;
     END LOOP;
END;
$$
LANGUAGE plpgsql;
CREATE FUNCTION

gaussdb=# SELECT merge_db(1, 'david');
 merge_db
----------

(1 row)

gaussdb=# SELECT merge_db(1, 'dennis');
 merge_db
----------

(1 row)

--删除FUNCTION和TABLE
gaussdb=# DROP FUNCTION merge_db;
DROP FUNCTION

gaussdb=# DROP TABLE db ;
DROP TABLE
```

ORA兼容模式下，在GUC参数exec_behavior_knob开启'on(PLSQL_RETURN_EXCEPTION)'选项时，返回语句类型转换的错误会被EXCEPTION子句捕获；关闭该选项时，返回语句类型转换的错误不会被EXCEPTION子句捕获。详细可见**exec_behavior_knob**。

```
gaussdb=# CREATE OR REPLACE FUNCTION func RETURN INT AS
BEGIN
    RETURN 'abc';
    EXCEPTION WHEN OTHERS THEN
    RETURN 123;
END;
/
CREATE FUNCTION
gaussdb=# set exec_behavior_knob='on(PLSQL_RETURN_EXCEPTION)';
SET
gaussdb=# SELECT func();
 func
------
  123
(1 row)
gaussdb=# set exec_behavior_knob='oFF(PLSQL_RETURN_EXCEPTION)';
SET
gaussdb=# SELECT func();
ERROR:  invalid input syntax for integer: "abc"
CONTEXT:  PL/pgSQL function func() while casting return value to function's return type
referenced column: func
gaussdb=# DROP FUNCTION IF EXISTS func;
DROP FUNCTION
```


版权所有 © 华为云计算技术有限公司

2.8.6.1 获取有关错误的信息

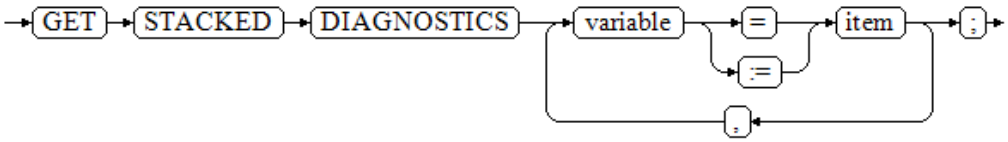
可以通过使用GET STACKED DIAGNOSTICS语句，查询当前PL/SQL函数执行过程中的异常信息。

语法

```
GET STACKED DIAGNOSTICS {{variable { = | := } item }[ , ... ]};
```

语法图如图2-37所示。

图 2-37 获取有关错误信息语法



其中，item表示异常专用的诊断项，目前可用的诊断项包括：

- PG_EXCEPTION_DETAIL：详细错误信息。
- PG_EXCEPTION_HINT：错误提示。
- PG_EXCEPTION_CONTEXT：异常的调用栈信息。
- MESSAGE_TEXT：主要错误信息。
- RETURNED_SQLSTATE：异常的SQLSTATE错误代码。

示例

```
gaussdb=# CREATE TABLE diag_table(int_col int, str_col varchar2);
CREATE TABLE
gaussdb=# CREATE OR REPLACE PROCEDURE get_diag_proc() AS
DECLARE
    sql_state text;
    message_text text;
    pg_exception_detail text;
    pg_exception_hint text;
    pg_exception_context text;
BEGIN
    UPDATE diag_table set int_col = 'aaa';
    EXCEPTION WHEN others THEN
        GET STACKED DIAGNOSTICS sql_state = returned_sqlstate;
        GET STACKED DIAGNOSTICS message_text = message_text;
        GET STACKED DIAGNOSTICS pg_exception_detail = pg_exception_detail;
        GET STACKED DIAGNOSTICS pg_exception_hint = pg_exception_hint;
        GET STACKED DIAGNOSTICS pg_exception_context = pg_exception_context;
        raise info 'sql_state: %, message_text: %', sql_state, message_text;
END;
/
CREATE PROCEDURE
gaussdb=# CALL get_diag_proc();
INFO:  sql_state: 22P02, message_text: invalid input syntax for integer: "aaa"
 get_diag_proc
---------------

(1 row)

gaussdb=# DROP TABLE diag_table;
DROP TABLE
gaussdb=# DROP PROCEDURE get_diag_proc();
DROP PROCEDURE
```


版权所有 © 华为云计算技术有限公司

2.8.6.2 获取当前的执行信息

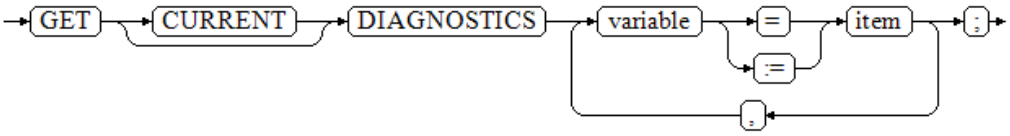
可以通过使用GET [CURRENT] DIAGNOSTICS语句，查询当前PL/SQL函数执行过程中的状态信息。

语法

```
GET [ CURRENT ] DIAGNOSTICS {{variable { = | := } item }[ , ... ]};
```

语法图如图2-38所示。

图 2-38 获取当前执行信息语法



其中，item表示执行状态的诊断项，目前可用的诊断项包括：

- ROW_COUNT：上一个SQL语句处理的行数量。
- RESULT_OID：上一个插入语句的Oid。

示例

```
gaussdb=# CREATE TABLE diag_table(int_col int, str_col varchar2);
CREATE TABLE
gaussdb=# CREATE OR REPLACE PROCEDURE get_diag_proc() AS
DECLARE
    nrows INT;
    result_oid oid;
BEGIN
    INSERT INTO diag_table values(1, 'aaa');
    GET CURRENT DIAGNOSTICS nrows = row_count;
    GET CURRENT DIAGNOSTICS result_oid = result_oid;
    raise info 'row_count: %, result_oid: %', nrows, result_oid;
END;
/
CREATE PROCEDURE
gaussdb=# CALL get_diag_proc();
INFO:  row_count: 1, result_oid: 0
 get_diag_proc
---------------

(1 row)

gaussdb=# DROP TABLE diag_table;
DROP TABLE
gaussdb=# DROP PROCEDURE get_diag_proc();
DROP PROCEDURE
```

2.8.7 GOTO 语句

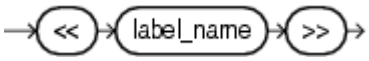
GOTO语句可以实现从GOTO位置到目标语句的无条件跳转。GOTO语句会改变当前的执行逻辑，因此应该慎重使用，或者也可以使用EXCEPTION处理特殊场景。当执行GOTO语句时，目标Label必须是唯一的。

语法

label declaration ::=语句如图2-39所示。

版权所有 © 华为云计算技术有限公司

图 2-39 label declaration ::



goto statement ::=语句如图2-40所示。

图 2-40 goto statement ::

示例

```
gaussdb=# CREATE OR REPLACE PROCEDURE GOTO_test()
AS
DECLARE
   v1  int;
BEGIN
   v1  := 0;
      LOOP
      EXIT WHEN v1 > 100;
            v1 := v1 + 2;
            if v1 > 25 THEN
                  GOTO pos1;
            END IF;
      END LOOP;
<<pos1>>
v1 := v1 + 10;
raise info 'v1 is %. ', v1;
END;
/
CREATE PROCEDURE
gaussdb=# call GOTO_test();
INFO:  v1 is 36.
 goto_test
-----------

(1 row)
```

限制场景

GOTO使用有以下限制场景

- 不支持有多个相同的GOTO labels目标场景，无论是否在同一个block中。
  ```
  BEGIN
    GOTO pos1;
    <<pos1>>
    SELECT * FROM ...
    <<pos1>>
    UPDATE t1 SET ...
  END;
  ```
- 不支持GOTO跳转到IF语句，CASE语句，LOOP语句中。
  ```
  BEGIN
     GOTO pos1;
     IF valid THEN
       <<pos1>>
       SELECT * FROM ...
     END IF;
   END;
  ```
- 不支持GOTO语句从一个IF子句跳转到另一个IF子句，或从一个CASE语句的WHEN子句跳转到另一个WHEN子句。
  ```
  BEGIN
    IF valid THEN
  ```


版权所有 © 华为云计算技术有限公司

```
    GOTO pos1;
    SELECT * FROM ...
  ELSE
    <<pos1>>
    UPDATE t1 SET ...
  END IF;
END;
```

- 不支持从外部块跳转到内部的BEGIN-END块。

```
BEGIN
  GOTO pos1;
  BEGIN
    <<pos1>>
    UPDATE t1 SET ...
  END;
END;
```

- 不支持从异常处理部分跳转到当前的BEGIN-END块。但可以跳转到上层BEGIN-END块。

```
BEGIN
  <<pos1>>
  UPDATE t1 SET ...
  EXCEPTION
    WHEN condition THEN
      GOTO pos1;
END;
```

- 如果从GOTO到一个不包含执行语句的位置，需要添加NULL语句。

```
DECLARE
  done  BOOLEAN;
BEGIN
  FOR i IN 1..50 LOOP
    IF done THEN
      GOTO end_loop;
    END IF;
    <<end_loop>>  -- 除非后面有可执行语句，否则不允许GOTO到这里
    NULL; -- 这里添加NULL语句避免错误
  END LOOP;  -- 如果没有上一句NULL会报错
END;
/
```

2.9 事务语句

存储过程本身会自动处于一个事务中。调用最外围存储过程开始时会自动开启一个事务，同时在调用结束时自动提交或者中间异常时回滚。除了系统自动的事务控制外，也可以使用COMMIT/ROLLBACK来控制存储过程中的事务。在存储过程中调用COMMIT/ROLLBACK命令，将提交/回滚当前事务并自动开启一个新的事务，后续的所有操作都会运行在此新的事务中。

保存点SAVEPOINT是事务中的一个特殊记号，它允许将那些在它建立后执行的命令全部回滚，把事务的状态恢复到保存点所在的时刻。存储过程中允许使用保存点来进行事务管理，当前支持保存点的创建、回滚和释放操作。需要特别注意，存储过程中使用回滚保存点只是回退当前事务的修改，而不会改变存储过程的执行流程，也不会回退存储过程中的局部变量值等。

语法

```
定义保存点
    SAVEPOINT savepoint_name;
回滚保存点
    ROLLBACK TO [SAVEPOINT] savepoint_name;
释放保存点
    RELEASE [SAVEPOINT] savepoint_name;
```


版权所有 © 华为云计算技术有限公司

使用场景

支持调用的上下文环境：

- 支持在PL/SQL的存储过程/函数内使用COMMIT/ROLLBACK/SAVEPOINT。
- 支持含有EXCEPTION的存储过程/函数使用COMMIT/ROLLBACK/SAVEPOINT。
- 支持在存储过程的EXCEPTION语句内使用COMMIT/ROLLBACK/SAVEPOINT。
- 支持在事务块里调用含有COMMIT/ROLLBACK/SAVEPOINT的存储过程，即通过BEGIN/START/END等开启控制的外部事务。
- 支持在子事务中调用含SAVEPOINT的存储过程并使用外部定义的SAVEPOINT，回退事务状态到存储过程外定义的SAVEPOINT位置。
- 支持在存储过程外部可见存储过程内部定义的SAVEPOINT，即存储过程外可以将事务修改回滚到存储过程中定义SAVEPOINT的位置。
- 支持多数PL/SQL的上下文和语句内调用COMMIT/ROLLBACK/SAVEPOINT，包括常用的IF/FOR/CURSOR LOOP/WHILE。

支持提交/回滚的内容：

- 支持DDL在COMMIT/ROLLBACK后的提交/回滚。
- 支持DML的COMMIT/ROLLBACK后的提交。
- 支持存储过程内GUC参数的回滚提交。

使用限制

不支持调用的上下文环境：

- 不支持除PL/SQL的其他存储过程中调用COMMIT/ROLLBACK/SAVEPOINT，例如PL/PYTHON等。
- 不支持事务块中调用了SAVEPOINT后，调用含有COMMIT/ROLLBACK的存储过程。
- 不支持TRIGGER中调用含有COMMIT/ROLLBACK/SAVEPOINT语句的存储过程。
- 不支持EXECUTE语句中调用COMMIT/ROLLBACK/SAVEPOINT语句。
- 不支持在CURSOR语句中打开一个含有COMMIT/ROLLBACK/SAVEPOINT的存储过程。
- 不支持带有IMMUTABLE以及SHIPPABLE的存储过程调用COMMIT/ROLLBACK/SAVEPOINT，或调用带有COMMIT/ROLLBACK/SAVEPOINT语句的存储过程。
- 不支持SQL中调用含有COMMIT/ROLLBACK/SAVEPOINT语句的存储过程，除了SELECT PROC以及CALL PROC。
- 存储过程头带有GUC参数设置的不允许调用COMMIT/ROLLBACK/SAVEPOINT语句。
- 不支持CURSOR/EXECUTE语句，以及各类表达式内调用COMMIT/ROLLBACK/SAVEPOINT。
- 不支持存储过程返回值与表达式计算中调用含有COMMIT/ROLLBACK/SAVEPOINT的存储过程。
- 不支持存储过程中释放存储过程外部定义的保存点。
- 存储过程事务和其中的自治事务是两个独立的事务，不能互相使用对方事务中定义的保存点。

版权所有 © 华为云计算技术有限公司

- 不支持高级包通过DBE_SQL调用含有COMMIT/ROLLBACK/SAVEPOINT语句的存储过程。

不支持提交回滚的内容：

- 不支持存储过程内声明变量以及传入变量的提交/回滚。
- 不支持存储过程内必须重启生效的GUC参数的提交/回滚。

说明

在存储过程、函数或匿名块内执行COMMIT或ROLLBACK语句时，PG_STAT相关系统对象中记录的统计信息不会立即刷新，当最外层事务执行完成时，统计信息才会更新。因此，通过存储过程、函数或匿名块执行大批量数据更新操作时，相关统计信息可能更新不及时。

示例

- 示例1：支持在PL/SQL的存储过程内使用COMMIT/ROLLBACK，后续示例依赖此用例。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);

gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE()
AS
BEGIN
    FOR i IN 0..20 LOOP
        INSERT INTO EXAMPLE1(COL1) VALUES (i);
        IF i % 2 = 0 THEN
            COMMIT;
        ELSE
            ROLLBACK;
        END IF;
    END LOOP;
END;
/
gaussdb=# call TRANSACTION_EXAMPLE();
 transaction_example
---------------------


(1 row)
```

- 示例2：

  支持含有EXCEPTION的存储过程使用COMMIT/ROLLBACK。

  支持在存储过程的EXCEPTION语句内使用COMMIT/ROLLBACK。

  支持DDL在COMMIT/ROLLBACK后的提交/回滚。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# CREATE OR REPLACE PROCEDURE TEST_COMMIT_INSERT_EXCEPTION_ROLLBACK()
AS
BEGIN
 DROP TABLE IF EXISTS TEST_COMMIT;
 CREATE TABLE TEST_COMMIT(A INT, B INT);
 INSERT INTO TEST_COMMIT SELECT 1, 1;
 COMMIT;
       CREATE TABLE TEST_ROLLBACK(A INT, B INT);
 RAISE EXCEPTION 'RAISE EXCEPTION AFTER COMMIT';
EXCEPTION
    WHEN OTHERS THEN
 INSERT INTO TEST_COMMIT SELECT 2, 2;
 ROLLBACK;
END;
/
gaussdb=# call TEST_COMMIT_INSERT_EXCEPTION_ROLLBACK();
NOTICE:  table "test_commit" does not exist, skipping
CONTEXT:  SQL statement "DROP TABLE IF EXISTS TEST_COMMIT"
```


版权所有 © 华为云计算技术有限公司

```
PL/pgSQL function test_commit_insert_exception_rollback() line 3 at SQL statement
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE TEST_COMMIT(A INT, B INT)"
PL/pgSQL function test_commit_insert_exception_rollback() line 4 at SQL statement
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE TEST_ROLLBACK(A INT, B INT)"
PL/pgSQL function test_commit_insert_exception_rollback() line 7 at SQL statement
 test_commit_insert_exception_rollback
---------------------------------------

(1 row)
```

- 示例3：支持在事务块里调用含有COMMIT/ROLLBACK的存储过程，即通过BEGIN/START/END等开启控制的外部事务。

```
gaussdb=# BEGIN;
    -- TEST_COMMIT_INSERT_EXCEPTION_ROLLBACK定义见示例2
    CALL TEST_COMMIT_INSERT_EXCEPTION_ROLLBACK();
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE TEST_COMMIT(A INT, B INT)"
PL/pgSQL function test_commit_insert_exception_rollback() line 4 at SQL statement
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE TEST_ROLLBACK(A INT, B INT)"
PL/pgSQL function test_commit_insert_exception_rollback() line 7 at SQL statement
 test_commit_insert_exception_rollback
---------------------------------------

(1 row)

gaussdb=# END;
COMMIT
```

- 示例4：支持多数PL/SQL的上下文和语句内调用COMMIT/ROLLBACK，包括常用的IF/FOR/CURSOR LOOP/WHILE。

```
gaussdb=# CREATE OR REPLACE PROCEDURE TEST_COMMIT2()
IS
BEGIN
    DROP TABLE IF EXISTS TEST_COMMIT;
    CREATE TABLE TEST_COMMIT(A INT);
    FOR I IN REVERSE 3..0 LOOP
 INSERT INTO TEST_COMMIT SELECT I;
 COMMIT;
    END LOOP;
    FOR I IN REVERSE 2..4 LOOP
 UPDATE TEST_COMMIT SET A=I;
 COMMIT;
    END LOOP;
EXCEPTION
WHEN OTHERS THEN
 INSERT INTO TEST_COMMIT SELECT 4;
    COMMIT;
END;
/
gaussdb=# call TEST_COMMIT2();
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE TEST_COMMIT(A INT)"
PL/pgSQL function test_commit2() line 4 at SQL statement
 test_commit2
--------------

(1 row)
```

- 示例5：支持存储过程内GUC参数的回滚提交。

```
gaussdb=# SHOW explain_perf_mode;
 explain_perf_mode
-------------------
```


版权所有 © 华为云计算技术有限公司

```
 normal
(1 row)

gaussdb=# SHOW enable_force_vector_engine;
 enable_force_vector_engine
----------------------------
 off
(1 row)

gaussdb=# CREATE OR REPLACE PROCEDURE GUC_ROLLBACK()
AS
BEGIN
    SET enable_force_vector_engine = on;
    COMMIT;
    SET explain_perf_mode TO pretty;
    ROLLBACK;
END;
/

gaussdb=# call GUC_ROLLBACK();
 guc_rollback
--------------

(1 row)

gaussdb=# SHOW explain_perf_mode;
 explain_perf_mode
-------------------
 normal
(1 row)

gaussdb=# SHOW enable_force_vector_engine;
 enable_force_vector_engine
----------------------------
 on
(1 row)

gaussdb=# SET enable_force_vector_engine = off;
```

- 示例6：支持在PL/SQL的存储过程内使用保存点回退事务部分修改。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# CREATE OR REPLACE PROCEDURE STP_SAVEPOINT_EXAMPLE1()
AS
BEGIN
    INSERT INTO EXAMPLE1 VALUES(1);
    SAVEPOINT s1;
    INSERT INTO EXAMPLE1 VALUES(2);
    ROLLBACK TO s1;  -- 回退插入记录2
    INSERT INTO EXAMPLE1 VALUES(3);
END;
/
gaussdb=# call STP_SAVEPOINT_EXAMPLE1();
 stp_savepoint_example1
------------------------

(1 row)
```

- 示例7：支持在PL/SQL的存储过程中使用保存点回退到存储过程外部定义的保存点。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# CREATE OR REPLACE PROCEDURE STP_SAVEPOINT_EXAMPLE2()
AS
BEGIN
    INSERT INTO EXAMPLE1 VALUES(2);
    ROLLBACK TO s1;  -- 回退插入记录2
    INSERT INTO EXAMPLE1 VALUES(3);
END;
/
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# BEGIN;
INSERT INTO EXAMPLE1 VALUES(1);
SAVEPOINT s1;
gaussdb=# CALL STP_SAVEPOINT_EXAMPLE2();
 stp_savepoint_example2
------------------------

(1 row)

gaussdb=# SELECT * FROM EXAMPLE1;
 col1
------
    0
    4
   10
   14
   16
   18
   20
    3
    3
    3
    2
    6
    8
   12
    1
    1
(16 rows)

gaussdb=# COMMIT;
COMMIT
```

- 示例8：支持在存储过程外部回退到在PL/SQL存储过程内部定义的保存点。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# CREATE OR REPLACE PROCEDURE STP_SAVEPOINT_EXAMPLE3()
AS
BEGIN
    INSERT INTO EXAMPLE1 VALUES(1);
    SAVEPOINT s1;
    INSERT INTO EXAMPLE1 VALUES(2);
END;
/

gaussdb=# BEGIN;
gaussdb=# INSERT INTO EXAMPLE1 VALUES(3);
gaussdb=# CALL STP_SAVEPOINT_EXAMPLE3();
 stp_savepoint_example3
------------------------

(1 row)

gaussdb=# ROLLBACK TO SAVEPOINT s1; --回退存储过程中插入记录2
ROLLBACK
gaussdb=# SELECT * FROM EXAMPLE1;
 col1
------
    0
    4
   10
   14
   16
   18
   20
    3
    3
    3
    3
```


版权所有 © 华为云计算技术有限公司

```
   2
   6
   8
  12
   1
   1
   1
(18 rows)

gaussdb=# COMMIT;
COMMIT
```

- 示例9：支持函数(Function)中调用commit/rollback语句。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# CREATE OR REPLACE FUNCTION FUNCTION_EXAMPLE1() RETURN INT
AS
EXP INT;
BEGIN
    FOR i IN 0..20 LOOP
        INSERT INTO EXAMPLE1(col1) VALUES (i);
        IF i % 2 = 0 THEN
            COMMIT;
        ELSE
            ROLLBACK;
        END IF;
    END LOOP;
    SELECT COUNT(*) FROM EXAMPLE1 INTO EXP;
    RETURN EXP;
END;
/
gaussdb=# call FUNCTION_EXAMPLE1();
 function_example1
-------------------
                29
(1 row)
```

- 示例10：不允许Trigger的存储过程包含COMMIT/ROLLBACK语句、或调用带有COMMIT/ROLLBACK语句的存储过程。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# INSERT INTO EXAMPLE1(col1) VALUES (1);
gaussdb=# CREATE OR REPLACE FUNCTION FUNCTION_TRI_EXAMPLE2() RETURN TRIGGER
AS
EXP INT;
BEGIN
    FOR i IN 0..20 LOOP
        INSERT INTO EXAMPLE1(col1) VALUES (i);
        IF i % 2 = 0 THEN
            COMMIT;
        ELSE
            ROLLBACK;
        END IF;
    END LOOP;
    SELECT COUNT(*) FROM EXAMPLE1 INTO EXP;
END;
/

gaussdb=# CREATE TRIGGER TRIGGER_EXAMPLE AFTER DELETE ON EXAMPLE1
FOR EACH ROW EXECUTE PROCEDURE FUNCTION_TRI_EXAMPLE2();
CREATE TRIGGER
gaussdb=# DELETE FROM EXAMPLE1;
ERROR:  Can not commit/rollback if it's atomic is true: can not use commit rollback in Complex SQL
CONTEXT:  PL/pgSQL function function_tri_example2() line 7 at COMMIT
```

- 示例11：不支持带有IMMUTABLE以及SHIPPABLE的存储过程调用commitrollback，或调用带有commit/rollback语句的存储过程。

```
gaussdb=# DROP TABLE IF EXISTS EXAMPLE1;
gaussdb=# CREATE TABLE EXAMPLE1(COL1 INT);
gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE1()
```


版权所有 © 华为云计算技术有限公司

```
IMMUTABLE
AS
EXP INT;
BEGIN
    FOR i IN 0..20 LOOP
        SELECT COUNT(*) FROM EXAMPLE1 INTO EXP;
        IF i % 2 = 0 THEN
            COMMIT;
        ELSE
            ROLLBACK;
        END IF;
    END LOOP;
END;
/
gaussdb=# CALL TRANSACTION_EXAMPLE1();
ERROR:  Can not commit/rollback if it's atomic is true: commit/rollback/savepoint is not allowed in a non-volatile function
CONTEXT:  PL/pgSQL function transaction_example1() line 7 at COMMIT
```

- 示例12：不支持存储过程中任何变量的提交，包括存储过程内声明的变量或者传入的参数。

```
gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE2(EXP_OUT OUT INT)
AS
EXP INT;
BEGIN
    EXP_OUT := 0;
    COMMIT;
    DBE_OUTPUT.PRINT_LINE('EXP IS:'||EXP);
    EXP_OUT := 1;
    ROLLBACK;
    DBE_OUTPUT.PRINT_LINE('EXP IS:'||EXP);
END;
/
gaussdb=# CALL TRANSACTION_EXAMPLE2(100);
EXP IS:
EXP IS:
 exp_out
---------
       1
(1 row)
```

- 示例13：不支持出现在SQL中的调用（除了SELECT PROCEDURE）。

```
gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE3()
AS
BEGIN
    FOR i IN 0..20 LOOP
        INSERT INTO EXAMPLE1 (col1) VALUES (i);
        IF i % 2 = 0 THEN
            EXECUTE IMMEDIATE 'COMMIT';
        ELSE
            EXECUTE IMMEDIATE 'ROLLBACK';
        END IF;
    END LOOP;
END;
/
gaussdb=# CALL TRANSACTION_EXAMPLE3();
ERROR:  cannot call transaction statements in EXECUTE IMMEDIATE statement.
CONTEXT:  PL/pgSQL function transaction_example3() line 6 at EXECUTE statement
```

- 示例14：存储过程头带有GUC参数设置的不允许调用COMMIT/ROLLBACK语句。

```
gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE4()
SET ARRAY_NULLS TO "ON"
AS
BEGIN
    FOR i IN 0..20 LOOP
        INSERT INTO EXAMPLE1 (col1) VALUES (i);
        IF i % 2 = 0 THEN
            COMMIT;
        ELSE
            ROLLBACK;
```


版权所有 © 华为云计算技术有限公司

```
        END IF;
    END LOOP;
END;
/
gaussdb=# CALL TRANSACTION_EXAMPLE4();
ERROR:  Can not commit/rollback if it's atomic is true: transaction statement in store procedure with GUC setting in option clause is not supported
CONTEXT:  PL/pgSQL function transaction_example4() line 6 at COMMIT
```

- 示例15：游标OPEN的对象不允许为带有COMMIT/ROLLBACK语句的存储过程。

```
gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE5(INTIN IN INT, INTOUT OUT INT)
AS
BEGIN
INTOUT := INTIN + 1;
COMMIT;
END;
/

gaussdb=# CREATE OR REPLACE PROCEDURE TRANSACTION_EXAMPLE6()
AS
CURSOR CURSOR1(EXPIN INT)
IS SELECT TRANSACTION_EXAMPLE5(EXPIN);
INTEXP INT;
BEGIN
    FOR i IN 0..20 LOOP
        OPEN CURSOR1(i);
        FETCH CURSOR1 INTO INTEXP;
        INSERT INTO EXAMPLE1(COL1) VALUES (INTEXP);
        IF i % 2 = 0 THEN
            COMMIT;
        ELSE
            ROLLBACK;
        END IF;
        CLOSE CURSOR1;
    END LOOP;
END;
/
gaussdb=# CALL TRANSACTION_EXAMPLE6();
ERROR:  Can not commit/rollback if it's atomic is true: transaction statement in store procedure used as cursor is not supported
CONTEXT:  PL/pgSQL function transaction_example5(integer) line 4 at COMMIT
referenced column: transaction_example5
PL/pgSQL function transaction_example6() line 8 at FETCH
```

- 示例16：不支持CURSOR/EXECUTE语句，以及各类表达式内调用COMMIT/ROLLBACK。

```
gaussdb=# CREATE OR REPLACE PROCEDURE exec_func1()
AS
BEGIN
    CREATE TABLE TEST_exec(A INT);
COMMIT;
END;
/
gaussdb=# CREATE OR REPLACE PROCEDURE exec_func2()
AS
BEGIN
EXECUTE exec_func1();
COMMIT;
END;
/
gaussdb=# CALL exec_func2();
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE TEST_exec(A INT)"
PL/pgSQL function exec_func1() line 3 at SQL statement
PL/pgSQL function exec_func2() line 3 at EXECUTE statement
ERROR:  Can not commit/rollback if it's atomic is true: transaction statement in store procedure used as a expression is not supported
```


版权所有 © 华为云计算技术有限公司

```
CONTEXT:  PL/pgSQL function exec_func1() line 4 at COMMIT
PL/pgSQL function exec_func2() line 3 at EXECUTE statement
```

- 示例17：不支持存储过程返回值与表达式计算。

```
gaussdb=# CREATE OR REPLACE PROCEDURE exec_func3(RET_NUM OUT INT)
AS
BEGIN
    RET_NUM := 1+1;
COMMIT;
END;
/
gaussdb=# CREATE OR REPLACE PROCEDURE exec_func4(ADD_NUM IN INT)
AS
SUM_NUM INT;
BEGIN
SUM_NUM := ADD_NUM + exec_func3();
COMMIT;
END;
/
gaussdb=# CALL exec_func4(1);
ERROR:  Can not commit/rollback if it's atomic is true: transaction statement in store procedure used as a expression is not supported
CONTEXT:  PL/pgSQL function exec_func3() line 4 at COMMIT
PL/pgSQL function exec_func4(integer) line 4 at assignment
```

- 示例18：不支持存储过程中释放存储过程外部定义的保存点。

```
gaussdb=# CREATE OR REPLACE PROCEDURE STP_SAVEPOINT_EXAMPLE3()
AS
BEGIN
    INSERT INTO EXAMPLE1 VALUES(2);
    RELEASE SAVEPOINT s1;  -- 释放存储过程外部定义的保存点
    INSERT INTO EXAMPLE1 VALUES(3);
END;
/

gaussdb=# BEGIN;
INSERT INTO EXAMPLE1 VALUES(1);
SAVEPOINT s1;
CALL STP_SAVEPOINT_EXAMPLE3();
COMMIT;
ERROR:  cannot release outer savepoint
CONTEXT:  PL/pgSQL function stp_savepoint_example3() line 4 at RELEASE SAVEPOINT
```

2.10 其他语句

2.10.1 锁操作

GaussDB提供了多种锁模式用于控制对表中数据的并发访问。

GaussDB提供的多种锁模式可以用在MVCC（多版本并发控制）无法给出期望行为的场合。

大多数GaussDB命令自动施加恰当的锁，以保证被引用的表在命令执行时不会以一种不兼容的方式被删除或者修改。比如，在存在其他并发操作的时候，ALTER TABLE是不能在同一个表上执行的。

2.10.2 游标操作

GaussDB中游标（cursor）是系统为用户开设的一个数据缓冲区，存放着SQL语句的执行结果。每个游标区都有一个名称。用户可以用SQL语句逐一从游标中获取记录，并赋给主变量，交由主语言进一步处理。

游标的操作主要有游标的定义、打开、获取和关闭。

版权所有 © 华为云计算技术有限公司

完整的游标操作示例可参考**显式游标**。

2.11 游标

2.11.1 游标概述

为了处理SQL语句，存储过程进程分配一段内存区域来保存上下文联系。游标是指向上下文区域的句柄或指针。借助游标，存储过程可以控制上下文区域的变化。

须知

- 当存储过程使用returns返回游标时，如果使用JDBC调用该存储过程，返回的游标将不可用，使用out出参不受影响。
- 在分布式下，当SQL语句中调用的存储过程参数为refcursor类型或返回值为refcursor类型时，不支持下推。
- 存储过程内commit/rollback时，显式游标为保证在commit/rollback后仍可用，会缓存游标所有数据，若游标数据量较大，此过程耗时可能较长。
- 存储过程内FETCH游标内的数据放入指定的输出变量中，当输出变量的个数大于实际游标查询结果集合的数据列时，会给多出的变量赋空值，不会产生报错；当输出变量的个数小于实际游标查询结果集合的数据列时，多余的列将被静默丢弃，不会产生报错。
- 开启GUC参数**forbid_hold_cursor_with_forupdate**时，若游标带有FOR UPDATE/FOR SHARE特性且在游标循环或for loop查询语句中使用，在执行事务提交/回滚操作后的下次循环访问游标时报错。
- 在开启GUC参数behavior_compat_options的plstmt_implicit_savepoint选项时，若游标执行下推到DN的SQL报错，会导致事务状态异常，无法被EXCEPTION捕获。

游标的使用分为显式游标和隐式游标。对于不同的SQL语句，游标的使用情况不同，详细信息请参见**表2-2**。

表 2-2 游标使用情况

SQL语句	游标
非查询语句	隐式的
结果是单行的查询语句	隐式的或显式的
结果是多行的查询语句	显式的

2.11.2 显式游标

显式游标主要用于对查询语句的处理，尤其是在查询结果为多条记录的情况下。

处理步骤

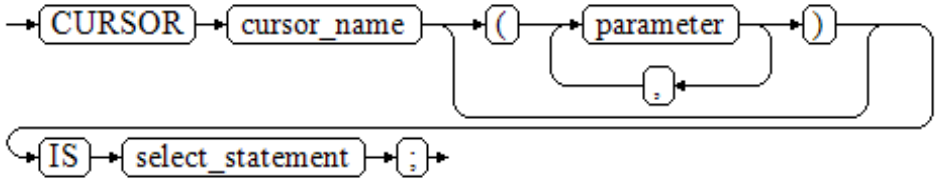
显式游标处理需六个PL/SQL步骤：

版权所有 © 华为云计算技术有限公司

步骤1 定义静态游标，即定义一个游标名，以及与其相对应的SELECT语句。

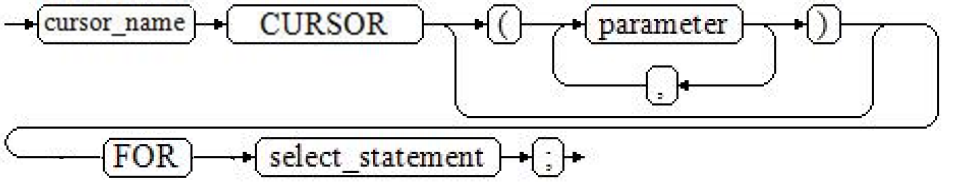
定义静态游标的语法图，如图2-41或图2-42所示。

图 2-41 static_cursor_define::=



或

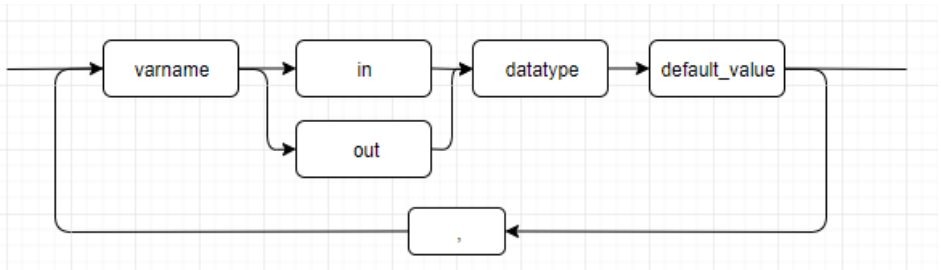
图 2-42 static_cursor_define::=



参数说明：

- cursor_name：定义的游标名。
- parameter：游标参数，只能为输入参数，参数支持定义默认值，以:=、=、default定义。其格式如图2-43所示。

  图 2-43 parameter 格式使用参考图

- select_statement：查询语句。

说明

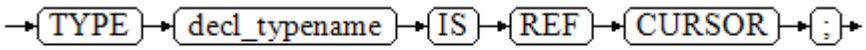
- 根据执行计划的不同，系统会自动判断该游标是否可以用于以倒序的方式检索数据行。
- 语法上支持parameter为输出参数，但其行为与输入参数保持一致。
- 在exception内部会自动关闭。
- PACKAGE内创建的带参数游标，游标参数可以作为变量使用。

定义动态游标，指ref游标，可以通过一组静态的SQL语句动态的打开游标。首先定义ref游标类型，然后定义该游标类型的游标变量，在打开游标时通过OPEN FOR动态绑定SELECT语句。

版权所有 © 华为云计算技术有限公司

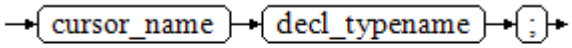
定义动态游标的语法图，如图2-44和图2-45所示。

图 2-44 cursor_typename::=



GaussDB支持sys_refcursor动态游标类型，函数或存储过程可以通过sys_refcursor参数传入或传出游标结果集合，函数也可以通过返回sys_refcursor来返回游标结果集合。

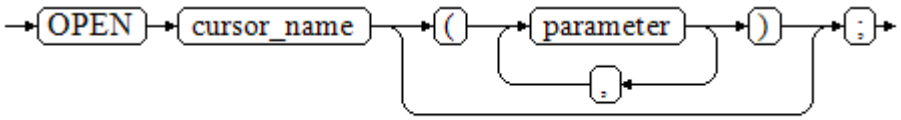
图 2-45 dynamic_cursor_define::=



步骤2 打开静态游标：就是执行游标所对应的SELECT语句，将其查询结果放入工作区，并且指针指向工作区的首部，标识游标结果集合。如果游标查询语句中带有FOR UPDATE选项，OPEN语句还将锁定数据库表中游标结果集合对应的数据行。

打开静态游标的语法图，如图2-46所示。

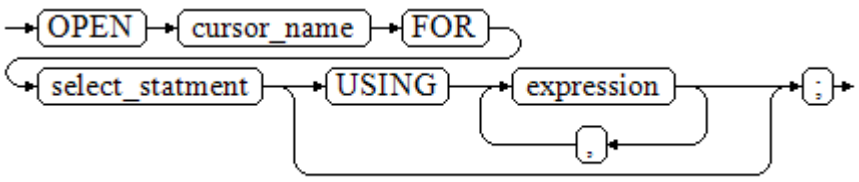
图 2-46 open_static_cursor::=



打开动态游标：可以通过OPEN FOR语句打开动态游标，动态绑定SQL语句。

打开动态游标的语法图，如图2-47所示。

图 2-47 open_dynamic_cursor::=



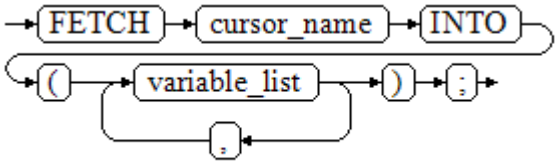
PL/SQL程序不能用OPEN语句重复打开一个游标。

步骤3 提取游标数据：检索结果集合中的数据行，放入指定的输出变量中。

提取游标数据的语法图，如图2-48所示。

版权所有 © 华为云计算技术有限公司

图 2-48 fetch_cursor::=



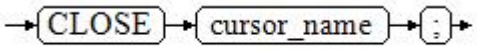
步骤4 对该记录进行处理。

步骤5 继续处理，直到活动集合中没有记录。

步骤6 关闭游标：当提取和处理完游标结果集合数据后，应及时关闭游标，以释放该游标所占用的系统资源，并使该游标的工作区变成无效，不能再使用FETCH语句获取其中数据。关闭后的游标可以使用OPEN语句重新打开。

关闭游标的语法图，如图2-49所示。

图 2-49 close_cursor::=



----结束

属性

游标的属性用于控制程序流程或者了解程序的状态。当运行DML语句时，PLSQL打开一个内建游标并处理结果，游标是维护查询结果的内存中的一个区域，游标在运行DML语句时打开，完成后关闭。显式游标的属性为：

- %FOUND布尔型属性：当最近一次读记录时成功返回，则值为TRUE。
- %NOTFOUND布尔型属性：当最近一次读记录时失败返回，则值为TRUE。
- %ISOPEN布尔型属性：当游标已打开时返回TRUE。
- %ROWCOUNT数值型属性：返回已从游标中读取的记录数。

示例

前置DDL、DML，本节后续示例依赖此用例。

```
gaussdb=# drop schema if exists hr cascade;
gaussdb=# create schema hr;
gaussdb=# set current_schema = hr;
gaussdb=# drop table if exists sections;
gaussdb=# drop table if exists staffs;
gaussdb=# drop table if exists department;
--创建部门表
gaussdb=# create table sections(
    section_name varchar(100),
    place_id     int,
    section_id   int
);
gaussdb=# insert into sections values ('hr',1,1);

--创建员工表
gaussdb=# create table staffs(
    staff_id number(6),
    salary number(8,2),
```


版权所有 © 华为云计算技术有限公司

```
    section_id int,
    first_name varchar(20)
);
gaussdb=# insert into staffs values (1,100,1,'Tom');

--创建部门表
gaussdb=# create table department(
    section_id int
);
--游标参数的传递方法。
gaussdb=# CREATE OR REPLACE PROCEDURE cursor_proc1()
AS
DECLARE
    DEPT_NAME VARCHAR(100);
    DEPT_LOC NUMBER(4);
    --定义游标
    CURSOR C1 IS
        SELECT section_name, place_id FROM hr.sections WHERE section_id <= 50;
    CURSOR C2(sect_id INTEGER) IS
        SELECT section_name, place_id FROM hr.sections WHERE section_id <= sect_id;
    TYPE CURSOR_TYPE IS REF CURSOR;
    C3 CURSOR_TYPE;
    SQL_STR VARCHAR(100);
BEGIN
    OPEN C1;--打开游标
    LOOP
        --通过游标取值
        FETCH C1 INTO DEPT_NAME, DEPT_LOC;
        EXIT WHEN C1%NOTFOUND;
        DBE_OUTPUT.PRINT_LINE(DEPT_NAME||'---'||DEPT_LOC);
    END LOOP;
    CLOSE C1;--关闭游标

    OPEN C2(10);
    LOOP
        FETCH C2 INTO DEPT_NAME, DEPT_LOC;
        EXIT WHEN C2%NOTFOUND;
        DBE_OUTPUT.PRINT_LINE(DEPT_NAME||'---'||DEPT_LOC);
    END LOOP;
    CLOSE C2;

    SQL_STR := 'SELECT section_name, place_id FROM hr.sections WHERE section_id <= :DEPT_NO;';
    OPEN C3 FOR SQL_STR USING 50;
    LOOP
        FETCH C3 INTO DEPT_NAME, DEPT_LOC;
        EXIT WHEN C3%NOTFOUND;
        DBE_OUTPUT.PRINT_LINE(DEPT_NAME||'---'||DEPT_LOC);
    END LOOP;
    CLOSE C3;
END;
/
gaussdb=# CALL cursor_proc1();

hr---1
hr---1
hr---1
 cursor_proc1
--------------

(1 row)


gaussdb=# DROP PROCEDURE cursor_proc1;
--给工资低于3000的员工增加工资500。
gaussdb=# CREATE TABLE hr.staffs_t1 AS TABLE hr.staffs;

gaussdb=# CREATE OR REPLACE PROCEDURE cursor_proc2()
AS
DECLARE
```


版权所有 © 华为云计算技术有限公司

```
  V_EMPNO  NUMBER(6);
  V_SAL    NUMBER(8,2);
  CURSOR C IS SELECT staff_id, salary FROM hr.staffs_t1;
BEGIN
  OPEN C;
  LOOP
    FETCH C INTO V_EMPNO, V_SAL;
    EXIT WHEN C%NOTFOUND;
    IF V_SAL<=3000 THEN
         UPDATE hr.staffs_t1 SET salary =salary + 500 WHERE staff_id = V_EMPNO;
    END IF;
  END LOOP;
  CLOSE C;
END;
/

gaussdb=# CALL cursor_proc2();
 cursor_proc2
--------------

(1 row)

--删除存储过程
gaussdb=# DROP PROCEDURE cursor_proc2;
gaussdb=# DROP TABLE hr.staffs_t1;
--SYS_REFCURSOR类型作为函数参数
gaussdb=# CREATE OR REPLACE PROCEDURE proc_sys_ref(O OUT SYS_REFCURSOR)
IS
C1 SYS_REFCURSOR;
BEGIN
OPEN C1 FOR SELECT section_id FROM HR.sections ORDER BY section_id;
O := C1;
END;
/

gaussdb=# DECLARE
C1 SYS_REFCURSOR;
TEMP NUMBER(4);
BEGIN
proc_sys_ref(C1);
LOOP
  FETCH C1 INTO TEMP;
  DBE_OUTPUT.PRINT_LINE(C1%ROWCOUNT);
  EXIT WHEN C1%NOTFOUND;
END LOOP;
END;
/
1
1
ANONYMOUS BLOCK EXECUTE
--删除存储过程
gaussdb=# DROP PROCEDURE proc_sys_ref;
```

2.11.3 隐式游标

对于非查询语句，如修改、删除操作，则由系统自动地为这些操作设置游标并创建其工作区，这些由系统隐含创建的游标称为隐式游标，隐式游标的名称为SQL，这是由系统定义的。

简介

对于隐式游标的操作，如定义、打开、取值及关闭操作，都由系统自动地完成，无需用户进行处理。用户只能通过隐式游标的相关属性，来完成相应的操作。在隐式游标的工作区中，所存放的数据是最新处理的一条SQL语句所包含的数据，与用户自定义的显式游标无关。

版权所有 © 华为云计算技术有限公司

格式调用为： SQL%

说明

- INSERT、UPDATE、DELETE和SELECT语句中不必明确定义游标。
- 隐式游标属性不受commit/rollback操作影响。
- 如果存储过程内的DML语句触发异常，隐式游标属性值不受该DML语句执行的影响，并保留执行该DML语句之前的值。

属性

隐式游标属性为：

- SQL%FOUND布尔型属性：当最近一次读记录时成功返回，则值为TRUE。
- SQL%NOTFOUND布尔型属性：当最近一次读记录时失败返回，则值为TRUE。
- SQL%ROWCOUNT数值型属性：返回已从游标中读取的记录数。
- SQL%ISOPEN布尔型属性：取值总是FALSE。SQL语句执行完毕立即关闭隐式游标。

示例

```
--删除员工表hr.staffs中某部门的所有员工，如果该部门中已没有员工，则在部门表hr.department中删除该部门。

gaussdb=# CREATE OR REPLACE PROCEDURE proc_cursor3()
AS
   DECLARE
   V_DEPTNO NUMBER(4) := 100;
   BEGIN
      DELETE FROM hr.staffs WHERE section_id = V_DEPTNO;
      --根据游标状态做进一步处理
      IF SQL%NOTFOUND THEN
      DELETE FROM hr.department WHERE section_id = V_DEPTNO;
      END IF;
   END;
/

gaussdb=# CALL proc_cursor3();
 proc_cursor3
--------------

(1 row)

--删除存储过程和临时表
gaussdb=# DROP PROCEDURE proc_cursor3;
```

2.11.4 游标循环

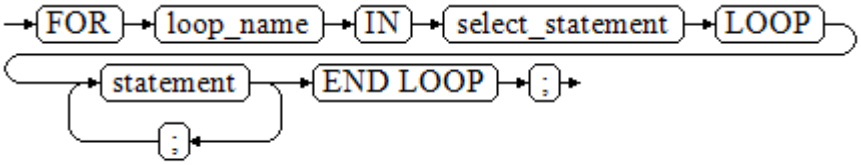
游标在WHILE语句、LOOP语句中的使用称为游标循环，一般这种循环都需要使用OPEN、FETCH和CLOSE语句。下面要介绍的一种循环不需要这些操作，可以简化游标循环的操作，这种循环方式适用于静态游标的循环，不用执行静态游标的四个步骤。

语法

FOR AS循环的语法如图2-50所示。

版权所有 © 华为云计算技术有限公司

图 2-50 FOR_AS_loop::=



注意事项

- 不能在该循环语句中对查询的表进行更新操作。
- 变量loop_name会自动定义且只在此循环中有效，类型和select_statement的查询结果类型一致。loop_name的取值就是select_statement的查询结果。
- 变量loop_name在编译过程中不会解析具体的类型，如果有需要解析具体类型的场景（比如loop_name作为重载函数或者存储过程的出入参）会编译报错。

示例

```
gaussdb=# DROP SCHEMA IF EXISTS hr CASCADE;
gaussdb=# CREATE SCHEMA hr;
gaussdb=# SET current_schema = hr;
--创建员工表
gaussdb=# CREATE TABLE staffs(
    staff_id number(6),
    salary number(8,2),
    section_id int,
    first_name varchar(20)
);
gaussdb=# INSERT INTO staffs VALUES (1,100,1,'Tom');
gaussdb=# BEGIN
FOR ROW_TRANS IN
        SELECT first_name FROM hr.staffs
    LOOP
        DBE_OUTPUT.PRINT_LINE (ROW_TRANS.first_name );
    END LOOP;
END;
/
--执行结果
gaussdb=# Tom
ANONYMOUS BLOCK EXECUTE
```

2.12 高级包

高级包现有两套接口，第一套为基础接口，第二套是为了提高易用性做了二次封装的接口，推荐使用第二套接口。

2.12.1 基础接口

2.12.1.1 PKG_SERVICE

PKG_SERVICE支持的接口请参见**表2-3**。

版权所有 © 华为云计算技术有限公司

表 2-3 PKG_SERVICE

接口名称	描述
PKG_SERVICE.SQL_IS_CONTEXT_ACTIVE	确认该CONTEXT是否已注册。
PKG_SERVICE.SQL_CLEAN_ALL_CONTEXTS	取消所有注册的CONTEXT。
PKG_SERVICE.SQL_REGISTER_CONTEXT	注册一个CONTEXT。
PKG_SERVICE.SQL_UNREGISTER_CONTEXT	取消注册该CONTEXT。
PKG_SERVICE.SQL_SET_SQL	向CONTEXT设置一条SQL语句，目前只支持SELECT。
PKG_SERVICE.SQL_RUN	在一个CONTEXT上执行设置的SQL语句。
PKG_SERVICE.SQL_NEXT_ROW	读取该CONTEXT中的下一行数据。
PKG_SERVICE.SQL_GET_VALUE	读取该CONTEXT中动态定义的列值。
PKG_SERVICE.SQL_SET_RESULT_TYPE	根据类型OID动态定义该CONTEXT的一个列。
PKG_SERVICE.JOB_CANCEL	通过任务id来删除定时任务。
PKG_SERVICE.JOB_FINISH	禁用或者启用定时任务。
PKG_SERVICE.JOB_SUBMIT	提交一个定时任务。作业号由系统自动生成或由用户指定。
PKG_SERVICE.JOB_UPDATE	修改定时任务的属性，包括任务内容、下次执行时间、执行间隔。
PKG_SERVICE.SUBMIT_ON_NODES	提交一个任务到所有节点，作业号由系统自动生成。
PKG_SERVICE.ISUBMIT_ON_NODES	提交一个任务到所有节点，作业号由用户指定。
PKG_SERVICE.SQL_GET_ARRAY_RESULT	获取该CONTEXT中返回的数组值。
PKG_SERVICE.SQL_GET_VARIABLE_RESULT	获取该CONTEXT中返回的列值。

说明

使用EXECUTE DIRECT ON语法执行PKG_SERVICE包中的函数时，如果GUC参数enable_udf_execute_direct的值为on，则可以执行；如果其值为off，则不可以执行。

- PKG_SERVICE.SQL_IS_CONTEXT_ACTIVE

  该函数用于确认一个CONTEXT是否已注册。该函数传入待查找的CONTEXT id，如果该CONTEXT存在返回true，否则返回false。

版权所有 © 华为云计算技术有限公司

PKG_SERVICE.SQL_IS_CONTEXT_ACTIVE原型为：

```
PKG_SERVICE.SQL_IS_CONTEXT_ACTIVE(
 context_id     IN INTEGER
)
RETURN BOOLEAN;
```

表 2-4 PKG_SERVICE.SQL_IS_CONTEXT_ACTIVE 接口参数说明

参数名称	描述
context_id	待查找的CONTEXT id号。

- PKG_SERVICE.SQL_CLEAN_ALL_CONTEXTS

  该函数用于取消所有CONTEXT。

  PKG_SERVICE.SQL_CLEAN_ALL_CONTEXTS原型为：

  ```
  PKG_SERVICE.SQL_CLEAN_ALL_CONTEXTS(
  )
  RETURN VOID;
  ```

- PKG_SERVICE.SQL_REGISTER_CONTEXT

  该函数用于打开一个CONTEXT。打开CONTEXT是后续对该CONTEXT进行各项操作的前提。该函数无需传入任何参数，内部会自动递增生成CONTEXT id，并将其作为返回值赋给一个INTEGER类型的变量。

  PKG_SERVICE.SQL_REGISTER_CONTEXT原型为：

  ```
  DBE_SQL.REGISTER_CONTEXT(
  )
  RETURN INTEGER;
  ```

- PKG_SERVICE.SQL_UNREGISTER_CONTEXT

  该函数用于关闭一个CONTEXT。关闭CONTEXT是该CONTEXT中各项操作的结束。如果在存储过程结束时未调用该函数，CONTEXT占用的内存将不会被释放，因此关闭CONTEXT非常重要。由于异常情况可能导致存储过程中途退出，从而导致CONTEXT未能关闭，建议存储过程中有异常处理，将该接口包含在内。

  PKG_SERVICE.SQL_UNREGISTER_CONTEXT原型为：

  ```
  PKG_SERVICE.SQL_UNREGISTER_CONTEXT(
   context_id     IN INTEGER
  )
  RETURN INTEGER;
  ```

  表 2-5 PKG_SERVICE.SQL_UNREGISTER_CONTEXT 接口参数说明

参数名称	描述
context_id	计划关闭的CONTEXT id号。

- PKG_SERVICE.SQL_SET_SQL

  该函数用于解析指定游标的查询语句，被传入的查询语句会立即执行。目前仅支持SELECT查询语句的解析，且语句参数仅可通过text类型传递，长度不大于1G。

  PKG_SERVICE.SQL_SET_SQL原型为：

  ```
  PKG_SERVICE.SQL_SET_SQL(
  context_id     IN INTEGER,
  query_string   IN TEXT,
  language_flag  IN INTEGER
  )
  RETURN BOOLEAN;
  ```


版权所有 © 华为云计算技术有限公司

表 2-6 PKG_SERVICE.SQL_SET_SQL 接口参数说明

参数名称	描述
context_id	执行查询语句解析的CONTEXT id。
query_string	执行的查询语句。
language_flag	版本语言号，指定不同版本的行为。 ● 1为非兼容版本。 ● 2为ORA兼容模式版本。

- PKG_SERVICE.SQL_RUN

  该函数用于执行一个给定的CONTEXT。该函数接收一个CONTEXT id，运行后获得的数据用于后续操作。目前仅支持SELECT查询语句的执行。

  PKG_SERVICE.SQL_RUN原型为：

```
PKG_SERVICE.SQL_RUN(
context_id     IN INTEGER,
)
RETURN INTEGER;
```

表 2-7 PKG_SERVICE.SQL_RUN 接口参数说明

参数名称	描述
context_id	执行查询语句解析的CONTEXT id。

- PKG_SERVICE.SQL_NEXT_ROW

  该函数返回执行SQL实际返回的数据行数，每一次调用该接口都会读取下一行数据，直到数据读取完毕获取不到新行为止。

  PKG_SERVICE.SQL_NEXT_ROW原型为：

```
PKG_SERVICE.SQL_NEXT_ROW(
context_id     IN INTEGER,
)
RETURN INTEGER;
```

表 2-8 PKG_SERVICE.SQL_NEXT_ROW 接口参数说明

参数名称	描述
context_id	执行的CONTEXT id。

- PKG_SERVICE.SQL_GET_VALUE

  该函数用于返回指定CONTEXT中在给定位置的CONTEXT元素值。该接口访问的是由PKG_SERVICE.SQL_NEXT_ROW获取的数据。

  PKG_SERVICE.SQL_GET_VALUE原型为：

```
PKG_SERVICE.SQL_GET_VALUE(
context_id          IN    INTEGER,
pos                 IN    INTEGER,
col_type            IN    ANYELEMENT
)
RETURN ANYELEMENT;
```


版权所有 © 华为云计算技术有限公司

表 2-9 PKG_SERVICE.SQL_GET_VALUE 接口参数说明

参数名称	描述
context_id	执行的CONTEXT id。
pos	动态定义列在查询中的位置。
col_type	任意类型变量，定义列的返回值类型。

- PKG_SERVICE.SQL_SET_RESULT_TYPE

  该函数用于定义从指定CONTEXT返回的列。该接口只能应用于SELECT定义的CONTEXT中。定义的列通过查询列表的相对位置来标识，
  PKG_SERVICE.SQL_SET_RESULT_TYPE原型为：

```
PKG_SERVICE.SQL_SET_RESULT_TYPE(
context_id      IN INTEGER,
pos             IN INTEGER,
coltype_oid     IN ANYELEMENT,
maxsize         IN INTEGER
)
RETURN INTEGER;
```

表 2-10 PKG_SERVICE.SQL_SET_RESULT_TYPE 接口参数说明

参数名称	描述
context_id	执行的CONTEXT id。
pos	动态定义列在查询中的位置。
coltype_oid	任意类型的变量，可根据变量类型得到对应类型OID。
maxsize	定义的列的长度。

- PKG_SERVICE.JOB_CANCEL

  该函数用于删除指定的定时任务。

  PKG_SERVICE.JOB_CANCEL原型为：

```
PKG_SERVICE.JOB_CANCEL(
id  IN  INTEGER);
```

表 2-11 PKG_SERVICE.JOB_CANCEL 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	INTEGER	IN	否	将要删除的定时任务的id。

- PKG_SERVICE.JOB_FINISH

  该函数用于禁用或者启用定时任务。

  PKG_SERVICE.JOB_FINISH原型为：

版权所有 © 华为云计算技术有限公司

```
PKG_SERVICE.JOB_FINISH(
id          IN   INTEGER,
broken      IN   BOOLEAN,
next_time   IN   TIMESTAMP  DEFAULT  sysdate);
```

表 2-12 PKG_SERVICE.JOB_FINISH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	INTEGER	IN	否	将要禁用或启用的定时任务的id。
broken	BOOLEAN	IN	否	状态标志位，true代表禁用，false代表启用。根据true或false值更新当前JOB的状态；如果为空值，则不改变原有JOB的状态。
next_time	TIMESTAMP	IN	是	下次运行时间，默认为当前系统时间。该参数可以省略，为默认值。 ● 如果参数broken状态为true，则更新该参数为'4000-1-1'。 ● 如果参数broken状态为false，参数next_time不为空值时，更新指定JOB的next_time值；next_time为空值时，不更新next_time值。

- PKG_SERVICE.JOB_SUBMIT

  该函数用于提交一个系统提供的定时任务。

  PKG_SERVICE.JOB_SUBMIT原型为：

```
PKG_SERVICE.JOB_SUBMIT(
id           IN   BIGINT,
content      IN   TEXT,
next_time    IN   TIMESTAMP DEFAULT sysdate,
interval_time IN   TEXT  DEFAULT 'null',
job          OUT  INTEGER);
```

  说明

  - 当创建一个定时任务（JOB）时，系统默认将当前数据库和用户名与当前创建的定时任务绑定起来。
  - 该接口函数可以通过CALL语句或SELECT语句调用，如果通过SELECT语句调用，可以不填写出参。如果在存储过程中，则需要通过PERFORM语句调用该接口函数。
  - 如果提交的SQL语句任务使用到非PUBLIC的schema，应该指定表或者函数的schema，或者在SQL语句前添加SET current_schema = xxx;语句。

版权所有 © 华为云计算技术有限公司

表 2-13 PKG_SERVICE.JOB_SUBMIT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	BIGINT	IN	否	定时任务id。如果传入id为NULL，则内部会生成作业id。
content	TEXT	IN	否	要执行的SQL语句。支持一个或多个DML、匿名块、调用存储过程的语句或三种类型混合的场景。
next_time	TIMESTAMP	IN	否	下次作业运行时间。默认值为当前系统时间（sysdate）。如果是过去时间，在提交作业时表示立即执行。
interval_time	TEXT	IN	是	用来计算下次作业运行时间的时间表达式，可以是interval表达式，也可以是sysdate加上一个NUMERIC值（例如：sysdate+1.0/24）。如果为空值或字符串"null"，表示只执行一次，执行后，JOB状态STATUS变成'd'，表示不再执行。
job	INTEGER	OUT	否	作业号。范围为1～3000000。当使用SELECT语句调用pkg_service.job_submit时，该参数可以省略。

- PKG_SERVICE.JOB_UPDATE

  该函数用于修改定时任务的属性，包括任务内容、下次执行时间和执行间隔。

  PKG_SERVICE.JOB_UPDATE原型为：

```
PKG_SERVICE.JOB_UPDATE(
id            IN   BIGINT,
next_time     IN   TIMESTAMP,
interval_time IN   TEXT,
content       IN   TEXT);
```

表 2-14 PKG_SERVICE.JOB_UPDATE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	INTEGER	IN	否	指定的定时任务id。
next_time	TIMESTAMP	IN	是	下次运行时间。如果该参数为空值，则不更新指定JOB的next_time值，否则更新指定JOB的next_time值。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
interval_time	TEXT	IN	是	用来计算下次作业运行时间的时间表达式。 ● 如果该参数为空值，不更新指定job的interval_time值。 ● 如果该参数不为空值，会校验interval_time是否为有效的时间类型或interval类型，若是则更新指定JOB的interval_time值。 ● 如果为字符串"null"，表示只执行一次，执行后，JOB状态STATUS变成'd' ，表示不再执行。
content	TEXT	IN	是	执行的存储过程名或者SQL语句块。如果该参数为空值，则不更新指定JOB的content值，否则更新指定JOB的content值。

示例：

```
CREATE TABLE test_table(a int);
CREATE TABLE

CREATE OR REPLACE PROCEDURE test_job(a in int) IS
BEGIN
INSERT INTO test_table VALUES(a);
COMMIT;
END;
/
CREATE PROCEDURE

--PKG_SERVICE.JOB_SUBMIT
SELECT PKG_SERVICE.JOB_SUBMIT(NULL, 'call test_job(1);', to_date('20180101','yyyymmdd'),'sysdate
+1');
 job_submit
------------
      28269
(1 row)

SELECT PKG_SERVICE.JOB_SUBMIT(NULL, 'call test_job(1);', to_date('20180101','yyyymmdd'),'sysdate
+1.0/24');
 job_submit
------------
       1506
(1 row)

CALL PKG_SERVICE.JOB_SUBMIT(NULL, 'INSERT INTO test_table VALUES(1);  call test_job(1); call
test_job(1);', add_months(to_date('201701','yyyymm'),1), 'date_trunc(''day'',SYSDATE) + 1
+(8*60+30.0)/(24*60)' ,:jobid);
  job
-------
 14131
(1 row)

SELECT PKG_SERVICE.JOB_SUBMIT (101, 'insert_msg_statistic1;', sysdate, 'sysdate+3.0/24');
 job_submit
------------
       101
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
--PKG_SERVICE.JOB_UPDATE
CALL PKG_SERVICE.JOB_UPDATE(101, sysdate, 'sysdate + 1.0/1440', 'call test_job(1);');
 job_update
------------

(1 row)

CALL PKG_SERVICE.JOB_UPDATE(101, sysdate, 'sysdate + 1.0/1440', 'insert into test_table values(1);');
 job_update
------------

(1 row)

--PKG_SERVICE.JOB_FINISH
CALL PKG_SERVICE.JOB_FINISH(101,true);
 job_finish
------------

(1 row)

--PKG_SERVICE.JOB_CANCEL
CALL PKG_SERVICE.JOB_CANCEL(101);
 job_cancel
------------

(1 row)

DROP TABLE test_table;
DROP TABLE
```

- PKG_SERVICE.SUBMIT_ON_NODES

  该函数用于创建一个所有CN/DN上的定时任务。仅sysadmin/monitor admin有权限调用该函数。

  PKG_SERVICE.SUBMIT_ON_NODES原型为：

```
PKG_SERVICE.SUBMIT_ON_NODES(
node_name     IN   NAME,
database      IN   NAME,
what          IN   TEXT,
next_date     IN   TIMESTAMP WITHOUT TIME ZONE,
job_interval IN   TEXT,
job           OUT  INTEGER);
```

表 2-15 PKG_SERVICE.SUBMIT_ON_NODES 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
node_name	TEXT	IN	否	指定作业的执行节点，当前仅支持值为'ALL_NODE'（在所有节点执行）与'CCN'（在central coordinator执行）。
database	TEXT	IN	否	集群作业所使用的database，节点类型为'ALL_NODE'时仅支持值为'postgres'。
what	TEXT	IN	否	要执行的SQL语句。支持一个或多个DML、匿名块、调用存储过程的语句或三种类型混合的场景。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
nextdate	TIMESTAMP	IN	否	下次作业运行时间。默认值为当前系统时间（sysdate）。如果是过去时间，在提交作业时表示立即执行。
job_interval	TEXT	IN	否	用来计算下次作业运行时间的时间表达式，可以是interval表达式，也可以是sysdate加上一个NUMERIC值（例如：sysdate+1.0/24）。如果为空值或字符串"null"，表示只执行一次，执行后，JOB状态STATUS变成'd'，表示不再执行。
job	INTEGER	OUT	否	作业号。范围为1～3000000。当使用SELECT调用DBMS.SUBMIT_ON_NODES时，该参数可以省略。

示例：

```
SELECT pkg_service.submit_on_nodes('ALL_NODE', 'postgres', 'select
capture_view_to_json(''dbe_perf.statement'', 0);', sysdate, 'interval ''60 second''');
 submit_on_nodes
-----------------
           12068
(1 row)

SELECT pkg_service.submit_on_nodes('CCN', 'postgres', 'select
capture_view_to_json(''dbe_perf.statement'', 0);', sysdate, 'interval ''60 second''');
 submit_on_nodes
-----------------
            9027
(1 row)
```

- PKG_SERVICE.ISUBMIT_ON_NODES

  ISUBMIT_ON_NODES与SUBMIT_ON_NODES语法功能相同，但其第一个参数是入参，即指定的作业号，SUBMIT最后一个参数是出参，表示系统自动生成的作业号。仅sysadmin/monitor admin有权限调用该函数。

- PKG_SERVICE.SQL_GET_ARRAY_RESULT

  该函数用于返回绑定的数组类型的OUT参数的值，可以用来获取存储过程中的OUT参数。

  PKG_SERVICE.SQL_GET_ARRAY_RESULT原型为：

```
PKG_SERVICE.SQL_GET_ARRAY_RESULT(
   context_id in int,
   pos in VARCHAR2,
   column_value inout anyarray,
   result_type in anyelement
);
```

表 2-16 PKG_SERVICE.SQL_GET_ARRAY_RESULT 接口参数说明

参数名称	描述
context_id	待查找的CONTEXT id号。

版权所有 © 华为云计算技术有限公司

参数名称	描述
pos	绑定的参数名。
column_value	返回值。
result_type	返回值类型。

- PKG_SERVICE.SQL_GET_VARIABLE_RESULT

  该函数用于返回绑定的非数组类型的OUT参数的值，可以用来获取存储过程中的OUT参数。

  PKG_SERVICE.SQL_GET_VARIABLE_RESULT原型为：

```
PKG_SERVICE.SQL_GET_VARIABLE_RESULT(
    context_id in int,
    pos in VARCHAR2,
    result_type in anyelement
)
RETURNS anyelement;
```

表 2-17 PKG_SERVICE.SQL_GET_VARIABLE_RESULT 接口参数说明

参数名称	描述
context_id	待查找的CONTEXT id号。
pos	绑定的参数名。
result_type	返回值类型。

2.12.1.2 PKG_UTIL

PKG_UTIL支持的所有接口请参见**表2-18**：

说明

PKG_UTIL包中的LOB修改接口（如LOB_WRITE_HUGE、LOB_COPY_HUGE、LOB_CONVERTTOBLOB_HUGE、LOADBLOBFROMFILE等）仅更新内存中的LOB变量，不会自动更新其关联的物理表数据。即使目标行已通过SELECT ... FOR UPDATE锁定，LOB数据变更也不会自动持久化到数据库表中。如需保存修改结果，必须显式执行UPDATE语句。

表 2-18 PKG_UTIL

接口名称	描述
PKG_UTIL.LOB_READ	读取lob对象的一部分。
PKG_UTIL.LOB_WRITE	将源对象按照指定格式写入到目标对象。
PKG_UTIL.LOB_APPEND	将lob源对象追加到目标lob对象。
PKG_UTIL.LOB_COMPARE	根据指定长度比较两个lob对象。

版权所有 © 华为云计算技术有限公司

接口名称	描述
PKG_UTIL.LOB_MATCH	返回一个字符串在LOB中第N次出现的位置。
PKG_UTIL.LOB_RESET	该函数将BLOB中一段数据set为value值，返回处理后的BLOB以及实际处理的长度。
PKG_UTIL.LOB_GET_LENGTH	该函数获取并返回指定的LOB类型对象的长度。
PKG_UTIL.LOB_READ_HUGE	根据指定的长度及起始位置偏移读取LOB内容的一部分,并返回读取到的LOB和长度。
PKG_UTIL.LOB_WRITEAPPEND_HUGE	该函数将源blob/clob对象读取指定长度内容，并追加到目标blob/clob对象, 并返回目标对象。
PKG_UTIL.LOB_APPEND_HUGE	该函数将源blob/clob对象追加到目标blob/clob对象, 并返回目标对象。
PKG_UTIL.READ_BFILE_TO_BLOB	该函数将源BFILE文件读取成目标BLOB对象, 并返回目标对象。
PKG_UTIL.LOB_COPY_HUGE	该函数将源blob/clob对象，从指定偏移读取指定长度内容，写入到目标blob/clob对象的指定偏移位置, 并返回目标对象。
PKG_UTIL.BLOB_RESET	该函数将BLOB中一段数据set为value值，返回处理后的BLOB以及实际处理的长度。
PKG_UTIL.CLOB_RESET	该函数将一段数据set为空格，返回处理后的CLOB以及实际处理的长度。
PKG_UTIL.LOADBLOBFROMFILE	将源BFILE对象，从指定偏移读取指定长度内容，写入到目标BLOB对象的指定偏移位置, 并返回目标对象，读取位置，写入位置。
PKG_UTIL.LOADCLOBFROMFILE	将源BFILE对象，从指定偏移读取指定长度内容，写入到目标CLOB对象的指定偏移位置, 并返回目标对象，读取位置，写入位置。
PKG_UTIL.LOB_CONVERTTOBLOB_HUGE	将src_clob从指定偏移位置读取指定长度内容转成BLOB，并写入dest_lob的指定位置，amount为要转换的长度。
PKG_UTIL.LOB_CONVERTTOCLOB_HUGE	将src_clob从指定偏移位置读取指定长度内容转成CLOB，并写入dest_lob的指定位置，amount为要转换的长度。

版权所有 © 华为云计算技术有限公司

接口名称	描述
PKG_UTIL.BFILE_GET_LENGTH	该函数获取并返回指定的BFILE文件的长度。
PKG_UTIL.BFILE_OPEN	该函数打开BFILE文件，返回文件描述符。
PKG_UTIL.BFILE_CLOSE	该函数关闭打开的BFILE文件。
PKG_UTIL.LOB_WRITE_HUGE	将源对象从起始位置读取len长度内容，写入目标LOB对象的指定偏移位置，覆盖该位置原内容, 并返回目标LOB对象。
PKG_UTIL.IO_PRINT	将字符串打印输出。
PKG_UTIL.RAW_GET_LENGTH	获取raw的长度。
PKG_UTIL.RAW_CAST_FROM_VARCHAR2	将varchar2转化为raw。
PKG_UTIL.RAW_CAST_FROM_BINARY_INTEGER	将binary integer转化为raw。
PKG_UTIL.RAW_CAST_TO_BINARY_INTEGER	将raw转化为binary integer。
PKG_UTIL.RANDOM_SET_SEED	设置随机种子。
PKG_UTIL.RANDOM_GET_VALUE	返回随机值。
PKG_UTIL.FILE_SET_DIRNAME	设置当前操作的目录。
PKG_UTIL.FILE_OPEN	根据指定文件名和设置的目录打开一个文件。
PKG_UTIL.FILE_SET_MAX_LINE_SIZE	设置写入文件一行的最大长度。
PKG_UTIL.FILE_IS_CLOSE	检测一个文件句柄是否关闭。
PKG_UTIL.FILE_READ	从一个打开的文件句柄中读取指定长度的数据。
PKG_UTIL.FILE_READLINE	从一个打开的文件句柄中读取一行数据。
PKG_UTIL.FILE_WRITE	将buffer中的数据写入到文件中。
PKG_UTIL.FILE_WRITELINE	将buffer写入文件，并追加换行符。
PKG_UTIL.FILE_NEWLINE	新起一行。
PKG_UTIL.FILE_READ_RAW	从一个打开的文件句柄中读取指定长度的二进制数据。
PKG_UTIL.FILE_WRITE_RAW	将二进制数据写入到文件中。
PKG_UTIL.FILE_FLUSH	将一个文件句柄中的数据写入到物理文件中。

版权所有 © 华为云计算技术有限公司

接口名称	描述
PKG_UTIL.FILE_CLOSE	关闭一个打开的文件句柄。
PKG_UTIL.FILE_REMOVE	删除一个物理文件，操作需要有对应权限。
PKG_UTIL.FILE_RENAME	对于磁盘上的文件进行重命名，类似UNIX的mv。
PKG_UTIL.FILE_SIZE	返回文件大小。
PKG_UTIL.FILE_BLOCK_SIZE	返回文件含有的块数量。
PKG_UTIL.FILE_EXISTS	判断文件是否存在。
PKG_UTIL.FILE_GETPOS	返回文件的偏移量，单位字节。
PKG_UTIL.FILE_SEEK	设置文件位置为指定偏移。
PKG_UTIL.FILE_CLOSE_ALL	关闭一个会话中打开的所有文件句柄。
PKG_UTIL.EXCEPTION_REPORT_ERROR	抛出一个异常。
PKG_UTIL.APP_READ_CLIENT_INFO	读取client_info信息。
PKG_UTIL.APP_SET_CLIENT_INFO	设置client_info信息。
PKG_UTIL.LOB_CONVERTTOBLOB	CLOB类型转换成BLOB类型。
PKG_UTIL.LOB_CONVERTTOCLOB	BLOB类型转换成CLOB类型。
PKG_UTIL.LOB_RAWTOTEXT	将输入的BLOB对象转换为十六进制文本输出，奇数位自动前置补零。输入必须由合法的十六进制数字组成（0-9, A-F, a-f），如果包含任何非法字符会报错。
PKG_UTIL.LOB_TEXTTORAW	将输入的CLOB文本对象转换为RAW输出，奇数位自动前置补零。输入的文本必须由合法的十六进制数字组成（0-9, A-F, a-f），如果包含任何非法字符会报错。
PKG_UTIL.MATCH_EDIT_DISTANCE_SIMILARITY	计算两个字符串的差异。
PKG_UTIL.RAW_CAST_TO_VARCHAR2	raw类型转成VARCHAR2类型。
PKG_UTIL.SESSION_CLEAR_CONTEXT	清空session_context中的属性值。
PKG_UTIL.SESSION_SEARCH_CONTEXT	查找一个属性值。
PKG_UTIL.SESSION_SET_CONTEXT	设置一个属性值。
PKG_UTIL.UTILITY_FORMAT_CALL_STACK	查看存储过程的调用堆栈。

版权所有 © 华为云计算技术有限公司

接口名称	描述
PKG_UTIL.UTILITY_FORMAT_ERROR_BACKTRACE	查看存储过程的错误堆栈。
PKG_UTIL.UTILITY_FORMAT_ERROR_STACK	查看存储过程的报错信息。
PKG_UTIL.UTILITY_GET_TIME	查看系统Unix时间戳。
PKG_UTIL.UTILITY_COMPILE_SCHEMA	重编译指定Schema、PACKAGE、函数和存储过程。当编译到的PL/SQL对象遇到报错时，将直接返回，不再继续编译。不推荐使用该接口，推荐使用pkg_util.gs_compile_schema。
PKG_UTIL.GS_COMPILE_SCHEMA	重编译指定Schema、PACKAGE、函数和存储过程。当编译遇到PL/SQL对象报错时，异常将会被捕获，继续编译其它对象，直到全部对象编译成功或者到达尝试次数后停止。通过jdbc执行该高级包，sqlstate会打印00000错误码，00000错误码代表成功完成。
PKG_UTIL.APP_SET_MODULE	设置module的值。
PKG_UTIL.APP_READ_MODULE	读取module的值。
PKG_UTIL.APP_SET_ACTION	设置action的值。
PKG_UTIL.APP_READ_ACTION	读取action的值。
PKG_UTIL.MODIFY_PACKAGE_STATE	用于修改当前会话的PL/SQL的状态。

- PKG_UTIL.LOB_READ

  该函数LOB_READ读取一个对象，并返回指定部分。

  PKG_UTIL.LOB_READ函数原型为：

```
PKG_UTIL.LOB_READ(
lob       IN   ANYELEMENT,
len       IN   INT,
start     IN   INT,
mode      IN   INT
)
RETURN ANYELEMENT
```

  表 2-19 PKG_UTIL.LOB_READ 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob	CLOB / BLOB	IN	否	CLOB或者BLOB类型数据。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
len	INT	IN	否	返回结果长度。
start	INT	IN	否	相较于第一个字符的偏移量。
mode	INT	IN	否	判断读取操作的类型， 0： read; 1 : trim; 2 : substr。

- PKG_UTIL.LOB_WRITE

  该函数LOB_WRITE将源对象按照指定的参数写入目标对象，并返回目标对象。

  PKG_UTIL.LOB_WRITE函数原型为：

```
PKG_UTIL.LOB_WRITE(
dest_lob    INOUT   BLOB,
src_lob     IN      RAW
len         IN      INT,
start_pos   IN      BIGINT
)
RETURN BLOB;
PKG_UTIL.LOB_WRITE(
dest_lob    INOUT   CLOB,
src_lob     IN      VARCHAR2
len         IN      INT,
start_pos   IN      BIGINT
)
RETURN CLOB;
```

  表 2-20 PKG_UTIL.LOB_WRITE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	CLOB / BLOB	INOUT	否	写入的目标对象。
src_lob	CLOB / BLOB	IN	否	被写入的源对象。
len	INT	IN	否	源对象的写入长度。
start_pos	BIGINT	IN	否	目标对象的写入起始位置。

- PKG_UTIL.LOB_APPEND

  该函数LOB_APPEND将源BLOB/CLOB对象追加到目标BLOB/CLOB对象, 并返回目标对象。

  PKG_UTIL.LOB_APPEND函数原型为：

```
PKG_UTIL.LOB_APPEND(
dest_lob    INOUT   BLOB,
src_lob     IN      BLOB,
len         IN      INT DEFAULT NULL
```


版权所有 © 华为云计算技术有限公司

```
)
RETURN BLOB;

PKG_UTIL.LOB_APPEND(
dest_lob    INOUT   CLOB,
src_lob     IN      CLOB,
len         IN      INT DEFAULT NULL
)
RETURN CLOB;
```

表 2-21 PKG_UTIL.LOB_APPEND 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB/CLOB	INOUT	否	写入的目标BLOB/CLOB对象。
src_lob	BLOB/CLOB	IN	否	被写入的源BLOB/CLOB对象。
len	INT	IN	是	src中读取并append到dest上的长度，默认NULL，将src全部append上去。

- PKG_UTIL.LOB_COMPARE

  该函数LOB_COMPARE按照指定的起始位置、个数比较对象是否相同，lob1大则返回1，lob2大返回-1，相等返回0。

  PKG_UTIL.LOB_COMPARE函数原型为：

```
PKG_UTIL.LOB_COMPARE(
lob1        IN   ANYELEMENT,
lob2        IN   ANYELEMENT,
len         IN   INT DEFAULT 1073741771,
start_pos1      IN   INT DEFAULT 1,
start_pos2      IN   INT DEFAULT 1
)
RETURN INTEGER;
```

表 2-22 PKG_UTIL.LOB_COMPARE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob1	CLOB/BLOB	IN	否	待比较的字符串。
lob2	CLOB/BLOB	IN	否	待比较的字符串。
len	INT	IN	否	比较的长度。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
start_pos1	INT	IN	否	lob1起始偏移量。
start_pos2	INT	IN	否	lob2起始偏移量。

- PKG_UTIL.LOB_MATCH

  该函数LOB_MATCH返回pattern出现在lob对象中第match_nth次的位置。

  PKG_UTIL.LOB_MATCH函数原型为：

```
PKG_UTIL.LOB_MATCH(
lob          IN   ANYELEMENT,
pattern      IN   ANYELEMENT,
start        IN   INT,
match_nth    IN   INT DEFAULT 1
)
RETURN INTEGER;
```

表 2-23 PKG_UTIL.LOB_MATCH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob	CLOB/BLOB	IN	否	待比较的字符串。
pattern	CLOB/BLOB	IN	否	待匹配的pattern。
start	INT	IN	否	lob的起始比较位置。
match_nth	INT	IN	否	第几次匹配到。

- PKG_UTIL.LOB_RESET

  将BLOB中一段数据set为value值，返回处理后的BLOB以及实际处理的长度。

  PKG_UTIL.LOB_RESET函数原型为：

```
PKG_UTIL.LOB_RESET(
lob          INOUT   BLOB,
len          INOUT   INT,
start        IN   INT DEFAULT 1,
value         IN   INT DEFAULT 0
)
RETURN RECORD;
```


版权所有 © 华为云计算技术有限公司

表 2-24 PKG_UTIL.LOB_RESET 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob	BLOB	INOUT	否	待重置的字符串。 **说明** 该函数不对lob参数的内容进行重置，处理后的结果由函数返回。
len	INT	INOUT	否	重置的长度。
start	INT	IN	是	重置的起始位置。默认值‘1’。
value	INT	IN	是	设置的字符。默认值‘0’。

- PKG_UTIL.LOB_GET_LENGTH

  该函数获取并返回指定的LOB类型对象的长度。

  PKG_UTIL.LOB_GET_LENGTH函数原型为：

```
PKG_UTIL.LOB_GET_LENGTH(
   lob IN BLOB)
RETURN BIGINT;

PKG_UTIL.LOB_GET_LENGTH(
   lob IN CLOB)
RETURN BIGINT;
```

表 2-25 PKG_UTIL.LOB_GET_LENGTH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob	BLOB/CLOB	IN	否	指定的LOB类型对象。

- PKG_UTIL.LOB_READ_HUGE

  根据指定的长度及起始位置偏移读取LOB内容的一部分，并返回读取到的LOB和长度。

  PKG_UTIL.LOB_READ_HUGE函数原型为：

```
PKG_UTIL.LOB_READ_HUGE(
   lob       IN CLOB,
   len       IN BIGINT,
   start_pos IN BIGINT,
   mode      IN INTEGER)
RETURN RECORD;

PKG_UTIL.LOB_READ_HUGE(
   lob       IN BLOB,
   len       IN BIGINT,
   start_pos IN BIGINT,
   mode      IN INTEGER)
RETURN RECORD;
```


版权所有 © 华为云计算技术有限公司

```
PKG_UTIL.LOB_READ_HUGE(
   fd        IN INTEGER,
   len       IN BIGINT,
   start_pos IN BIGINT,
   mode      IN INTEGER)
RETURN RECORD;
```

表 2-26 PKG_UTIL.LOB_READ_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob/fd	BLOB/CLOB/INTEGER	IN	否	指定的LOB类型对象/BFILE文件的文件描述符。
len	BIGINT	IN	否	读取长度。
start_pos	BIGINT	IN	否	读取起始偏移位置。
mode	INTEGER	IN	否	read模式（0代表read、1代表trim、2代表substr）。

- PKG_UTIL.LOB_WRITEAPPEND_HUGE

  该函数LOB_WRITEAPPEND_HUGE将源BLOB/CLOB对象读取指定长度内容，并追加到目标BLOB/CLOB对象, 并返回目标对象。

  PKG_UTIL.LOB_WRITEAPPEND_HUGE函数原型为：

```
PKG_UTIL.LOB_WRITEAPPEND_HUGE(
   dest_lob INOUT CLOB,
   len      IN    INTEGER,
   src_lob  IN    VARCHAR2
)RETURN CLOB;

PKG_UTIL.LOB_WRITEAPPEND_HUGE(
   dest_lob INOUT BLOB,
   len      IN    INTEGER,
   src_lob  IN    RAW
)RETURN BLOB;
```


版权所有 © 华为云计算技术有限公司

表 2-27 PKG_UTIL.LOB_WRITEAPPEND_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB/CLOB	INOUT	否	写入的目标BLOB/CLOB对象。
len	INTEGER	IN	是	写入源对象的长度，为NULL则默认写入源对象全部。
src_lob	VARCHAR2/RAW	IN	否	被写入的源BLOB/CLOB对象。

- PKG_UTIL.LOB_APPEND_HUGE

  该函数LOB_APPEND_HUGE将源BLOB/CLOB对象追加到目标BLOB/CLOB对象, 并返回目标对象。

  PKG_UTIL.LOB_APPEND_HUGE函数原型为：

```
PKG_UTIL.LOB_APPEND_HUGE(
   dest_lob INOUT BLOB,
   src_lob  IN    BLOB)
RETURN BLOB;
PKG_UTIL.LOB_APPEND_HUGE(
   dest_lob INOUT CLOB,
   src_lob  IN    CLOB)
RETURN CLOB;
```

表 2-28 PKG_UTIL.LOB_APPEND_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB/CLOB	INOUT	否	写入的目标BLOB/CLOB对象。
src_lob	BLOB/CLOB	IN	否	被写入的源BLOB/CLOB对象。

- PKG_UTIL.READ_BFILE_TO_BLOB

  该函数READ_BFILE_TO_BLOB将源BFILE文件读取成目标BLOB对象, 并返回目标对象。

  PKG_UTIL.READ_BFILE_TO_BLOB函数原型为：

版权所有 © 华为云计算技术有限公司

```
PKG_UTIL.READ_BFILE_TO_BLOB(
   fd IN INTEGER
)RETURN BLOB;
```

表 2-29 PKG_UTIL.READ_BFILE_TO_BLOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
fd	INTEGER	IN	否	读取的源BFILE文件。

- PKG_UTIL.LOB_COPY_HUGE

  该函数LOB_COPY_HUGE将源BLOB/CLOB对象，从指定偏移读取指定长度内容，写入到目标BLOB/CLOB对象的指定偏移位置, 并返回目标对象。

  PKG_UTIL.LOB_COPY_HUGE函数原型为：

```
PKG_UTIL.LOB_COPY_HUGE(
   lob_obj    INOUT BLOB,
   source_obj  IN    BLOB,
   amount      IN    BIGINT,
   dest_offset IN    BIGINT DEFAULT 1,
   src_offset  IN    BIGINT DEFAULT 1
)RETURN BLOB;

PKG_UTIL.LOB_COPY_HUGE(
   lob_obj    INOUT CLOB,
   source_obj  IN    CLOB,
   amount      IN    BIGINT,
   dest_offset IN    BIGINT DEFAULT 1,
   src_offset  IN    BIGINT DEFAULT 1
)RETURN CLOB;
```

表 2-30 PKG_UTIL.LOB_COPY_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob_obj	BLOB/CLOB	INOUT	否	目标BLOB/CLOB对象。
source_obj	BLOB/CLOB	IN	否	源BLOB/CLOB对象。
amount	BIGINT	IN	否	复制的长度（BLOB以字节为单位，CLOB以字符为单位）。
dest_offset	BIGINT	IN	否	目标LOB的载入偏移位置。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
src_offset	BIGINT	IN	否	源LOB的读取偏移位置。

- PKG_UTIL.BLOB_RESET

  该函数将BLOB中一段数据set为value值，返回处理后的BLOB以及实际处理的长度。

  PKG_UTIL.BLOB_RESET函数原型为：

```
PKG_UTIL.BLOB_RESET(
   lob       INOUT BLOB,
   len       INOUT BIGINT,
   start_pos IN    BIGINT DEFAULT 1,
   value     IN    INTEGER DEFAULT 0
)RETURN RECORD;
```

  表 2-31 PKG_UTIL.BLOB_RESET 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob	BLOB	INOUT	否	待重置的LOB。 **说明** 该函数不对lob参数的内容进行重置，处理后的结果由函数返回。
len	BIGINT	INOUT	否	重置的长度，单位字节。
start	BIGINT	IN	是	重置的起始位置。默认值‘1’。
value	INTEGER	IN	是	设置的字符。默认值‘0’。

- PKG_UTIL.CLOB_RESET

  该函数将一段数据set为空格。

  PKG_UTIL.CLOB_RESET函数原型为：

```
PKG_UTIL.CLOB_RESET(
   lob       INOUT CLOB,
   len       INOUT BIGINT,
   start_pos IN    BIGINT DEFAULT 1
)RETURN RECORD;
```


版权所有 © 华为云计算技术有限公司

表 2-32 PKG_UTIL.CLOB_RESET 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
lob	CLOB	INOUT	否	待重置的LOB。
len	BIGINT	INOUT	否	重置的长度，单位字符。
start	BIGINT	IN	是	重置的起始位置，默认为1。

- PKG_UTIL.LOADBLOBFROMFILE

  该函数LOADBLOBFROMFILE将源BFILE对象，从指定偏移读取指定长度内容，写入到目标BLOB对象的指定偏移位置, 并返回目标对象、读取位置和写入位置。

  PKG_UTIL.LOADBLOBFROMFILE函数原型为：

```
PKG_UTIL.LOADBLOBFROMFILE(
   dest_lob     INOUT  BLOB,
   fd           IN     INTEGER,
   amount       IN     BIGINT,
   dest_offset  INOUT  BIGINT,
   file_offset  INOUT  BIGINT
)RETURN RECORD;
```

表 2-33 PKG_UTIL.LOADBLOBFROMFILE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB	INOUT	否	IN参数为目标BLOB对象，OUT参数为写入后的目标BLOB对象。
fd	INTEGER	IN	否	源BFILE对象的文件描述符。
amount	BIGINT	IN	否	复制的长度（BLOB以字节为单位，CLOB以字符为单位）。
dest_offset	BIGINT	INOUT	否	IN参数为目标LOB的写入偏移位置，OUT参数为实际写入位置。
src_offset	BIGINT	INOUT	否	IN参数为源BFILE的读取偏移位置，OUT参数为实际读取位置。

- PKG_UTIL.LOADCLOBFROMFILE

  该函数LOADCLOBFROMFILE将源BFILE对象，从指定偏移读取指定长度内容，写入到目标CLOB对象的指定偏移位置, 并返回目标对象，读取位置，写入位置。

  PKG_UTIL.LOADCLOBFROMFILE函数原型为：

```
PKG_UTIL.LOADCLOBFROMFILE(
   dest_lob     INOUT  CLOB,
```


版权所有 © 华为云计算技术有限公司

```
  fd          IN     INTEGER,
  amount      IN     BIGINT,
  dest_offset  INOUT  BIGINT,
  file_offset  INOUT  BIGINT
)RETURN RECORD;
```

表 2-34 PKG_UTIL.LOADCLOBFROMFILE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	CLOB	INOUT	否	IN参数为目标CLOB对象，OUT参数为写入后的目标CLOB对象。
fd	INTEGER	IN	否	源BFILE对象的文件描述符。
amount	BIGINT	IN	否	复制的长度（CLOB以字符为单位）。
dest_offset	BIGINT	INOUT	否	IN参数为目标LOB的写入偏移位置，OUT参数为实际写入位置。
src_offset	BIGINT	INOUT	否	IN参数为源BFILE的读取偏移位置，OUT参数为实际读取位置。

- PKG_UTIL.LOB_CONVERTTOBLOB_HUGE

  将src_clob从指定偏移位置读取指定长度内容转成BLOB，并写入dest_lob的指定位置，amount为要转换的长度。

  PKG_UTIL.LOB_CONVERTTOBLOB_HUGE函数原型为：

```
PKG_UTIL.LOB_CONVERTTOBLOB_HUGE(
  dest_lob    INOUT  BLOB,
  src_clob    IN     CLOB,
  amount      IN     BIGINT,
  dest_offset INOUT  BIGINT,
  src_offset  INOUT  BIGINT)
)RETURN RECORD;
```

表 2-35 PKG_UTIL.LOB_CONVERTTOBLOB_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB	INOUT	否	目标LOB。
src_clob	CLOB	IN	否	要转换的CLOB。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
amount	BIGINT	IN	否	转换的长度，字符为单位。
dest_offset	BIGINT	INOUT	否	IN参数为目标LOB的写入起始位置，OUT参数为实际的写入位置。
src_offset	BIGINT	INOUT	否	IN参数为源CLOB的读取起始位置，OUT参数为实际的读取起始位置。

- PKG_UTIL.LOB_CONVERTTOCLOB_HUGE

  将src_clob从指定偏移位置读取指定长度内容转成CLOB，并写入dest_lob的指定位置，amount为要转换的长度。

  PKG_UTIL.LOB_CONVERTTOCLOB_HUGE函数原型为：

```
PKG_UTIL.LOB_CONVERTTOCLOB_HUGE(
    dest_lob    INOUT  CLOB,
    src_blob    IN     BLOB,
    amount      IN     BIGINT,
    dest_offset INOUT  BIGINT,
    src_offset  INOUT  BIGINT)
)RETURN RECORD;
```

表 2-36 PKG_UTIL.LOB_CONVERTTOCLOB_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	CLOB	INOUT	否	目标LOB。
src_blob	BLOB	IN	否	要转换的BLOB。
amount	BIGINT	IN	否	转换的长度，字节为单位。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
dest_offset	BIGINT	INOUT	否	IN参数为目标LOB的写入起始位置，OUT参数为实际的写入位置。
src_offset	BIGINT	INOUT	否	IN参数为源CLOB的读取起始位置，OUT参数为实际的读取起始位置。

- PKG_UTIL.BFILE_GET_LENGTH
  该函数获取并返回指定的BFILE文件的长度。
  PKG_UTIL.BFILE_GET_LENGTH函数原型为：

```
PKG_UTIL.BFILE_GET_LENGTH(
    fd INTEGER
)RETURN BIGINT;
```

表 2-37 PKG_UTIL.LOB_GET_LENGTH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
fd	INTEGER	IN	否	指定的BFILE文件的文件描述符。

- PKG_UTIL.BFILE_OPEN
  该函数打开BFILE文件，返回文件描述符。
  PKG_UTIL.BFILE_OPEN函数原型为：

```
PKG_UTIL.BFILE_OPEN(
    file_name TEXT,
    open_mode TEXT)
RETURN INTEGER;
```

表 2-38 PKG_UTIL.BFILE_OPEN 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file_name	TEXT	IN	否	指定的BFILE文件的文件name。
open_mode	TEXT	IN	否	打开方式（只支持‘r’，read功能）

版权所有 © 华为云计算技术有限公司

- PKG_UTIL.BFILE_CLOSE

  该函数关闭打开的BFILE文件。

  PKG_UTIL.BFILE_CLOSE函数原型为：

```
PKG_UTIL.BFILE_CLOSE(
   fd INTEGER)
RETURN BOOL;
```

表 2-39 PKG_UTIL.BFILE_CLOSE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
fd	INTEGER	IN	否	指定的BFILE文件的文件描述符。

- PKG_UTIL.LOB_WRITE_HUGE

  将源对象从起始位置读取len长度内容，写入目标LOB对象的指定偏移位置，覆盖该位置原内容, 并返回目标LOB对象。

  PKG_UTIL.LOB_WRITE_HUGE函数原型为：

```
PKG_UTIL.LOB_WRITE_HUGE(
   dest_lob  INOUT BLOB,
   len       IN INTEGER,
   start_pos IN BIGINT,
   src_lob   IN RAW
)RETURN BLOB;

PKG_UTIL.LOB_WRITE_HUGE(
   dest_lob  INOUT CLOB,
   len       IN INTEGER,
   start_pos IN BIGINT,
   src_lob   IN VARCHAR2
)RETURN CLOB;
```

表 2-40 PKG_UTIL.LOB_WRITE_HUGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB/CLOB	INOUT	否	IN参数为待写入的目标LOB，OUT参数为写入内容后的LOB。
len	INTEGER	IN	否	待写入的长度（BLOB以字节为单位，CLOB以字符为单位）。
start_pos	BIGINT	IN	否	在dest_lob中写入的偏移位置。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
src_lob	RAW/VARCHAR2	IN	否	源LOB对象。

- PKG_UTIL.IO_PRINT
  该函数IO_PRINT将一段字符串打印输出。
  PKG_UTIL.IO_PRINT函数原型为：

```
PKG_UTIL.IO_PRINT(
format       IN   TEXT,
is_one_line  IN   BOOLEAN
)
RETURN VOID;
```

表 2-41 PKG_UTIL.IO_PRINT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
format	TEXT	IN	否	待打印输出的字符串。
is_one_line	BOOLEAN	IN	否	是否输出为一行。

- PKG_UTIL.RAW_GET_LENGTH
  该函数RAW_GET_LENGTH获取raw的长度。
  PKG_UTIL.RAW_GET_LENGTH函数原型为：

```
PKG_UTIL.RAW_GET_LENGTH(
value       IN   RAW
)
RETURN INTEGER;
```

表 2-42 PKG_UTIL.RAW_GET_LENGTH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
raw	RAW	IN	否	待获取长度的对象。

- PKG_UTIL.RAW_CAST_FROM_VARCHAR2
  该函数RAW_CAST_FROM_VARCHAR2将VARCHAR2转化为RAW。
  PKG_UTIL.RAW_CAST_FROM_VARCHAR2函数原型为：

版权所有 © 华为云计算技术有限公司

```
PKG_UTIL.RAW_CAST_FROM_VARCHAR2(
str       IN   VARCHAR2
)
RETURN RAW;
```

表 2-43 PKG_UTIL.RAW_CAST_FROM_VARCHAR2 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
str	VARCHAR2	IN	否	需要转化的源数据。

- PKG_UTIL.RAW_CAST_FROM_BINARY_INTEGER

  该函数RAW_CAST_FROM_BINARY_INTEGER将BIGINT转化为RAW。

  PKG_UTIL.RAW_CAST_FROM_BINARY_INTEGER函数原型为：

```
PKG_UTIL.RAW_CAST_FROM_BINARY_INTEGER(
value       IN   BIGINT,
endianess   IN   INTEGER
)
RETURN RAW;
```

表 2-44 PKG_UTIL.RAW_CAST_FROM_BINARY_INTEGER 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
value	BIGINT	IN	否	需要转化的源数据。
endianess	INTEGER	IN	否	表示字典序的整型值，现支持1或2或3（1代表BIG_ENDIAN，2代表LITTLE_ENDIAN，3代表MACHINE_ENDIAN）。

- PKG_UTIL.RAW_CAST_TO_BINARY_INTEGER

  该函数RAW_CAST_TO_BINARY_INTEGER将RAW转化为BINARY_INTEGER。

  PKG_UTIL.RAW_CAST_TO_BINARY_INTEGER函数原型为：

```
PKG_UTIL.RAW_CAST_TO_BINARY_INTEGER(
value       IN   RAW,
endianess   IN   INTEGER
)
RETURN INTEGER;
```

表 2-45 PKG_UTIL.RAW_CAST_TO_BINARY_INTEGER 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
value	RAW	IN	否	需要转化的源数据。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
endianess	INTEGER	IN	否	表示字典序的整型值，现支持1或2或3（1代表BIG_ENDIAN，2代表LITTLE_ENDIAN，3代表MACHINE_ENDIAN）。

- PKG_UTIL.RANDOM_SET_SEED

  该函数RANDOM_SET_SEED设置随机数种子。

  PKG_UTIL.RANDOM_SET_SEED函数原型为：

```
PKG_UTIL.RANDOM_SET_SEED(
seed        IN   INT
)
RETURN INTEGER;
```

表 2-46 PKG_UTIL.RANDOM_SET_SEED 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
seed	INT	IN	否	随机数种子。

- PKG_UTIL.RANDOM_GET_VALUE

  该函数RANDOM_GET_VALUE返回0~1区间的一个随机数，其有效数字为15位。

  PKG_UTIL.RANDOM_GET_VALUE函数原型为：

```
PKG_UTIL.RANDOM_GET_VALUE(
)
RETURN NUMERIC;
```

  说明

  建议不要在安全场景下使用PKG_UTIL.RANDOM_SET_SEED和PKG_UTIL.RANDOM_GET_VALUE函数。

- PKG_UTIL.FILE_SET_DIRNAME

  设置当前操作的目录，基本上所有涉及单个目录的操作，都需要调用此方法先设置操作的目录。

  PKG_UTIL.FILE_SET_DIRNAME函数原型为：

```
PKG_UTIL.FILE_SET_DIRNAME(
dir  IN  TEXT
)
RETURN BOOL
```


版权所有 © 华为云计算技术有限公司

表 2-47 PKG_UTIL.FILE_SET_DIRNAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dir	TEXT	IN	否	文件的目录位置，这个字符串是一个目录对象名。 **说明** 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误，下面的涉及location作为参数的函数也是同样的情况。

- PKG_UTIL.FILE_OPEN

  该函数用来打开一个文件，最多可以同时打开50个文件。并且该函数返回INTEGER类型的一个句柄。

  PKG_UTIL.FILE_OPEN函数原型为：

```
PKG_UTIL.FILE_OPEN(
file_name    IN  TEXT,
open_mode    IN  INTEGER)
```

表 2-48 PKG_UTIL.FILE_OPEN 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file_name	TEXT	IN	否	文件名，包含扩展（文件类型），不包括路径名。如果文件名中包含路径，在OPEN中会被忽略，在UNIX系统中，文件名不能以/.结尾。
open_mode	INTEGER	IN	否	指定文件的打开模式，包含r：read text，w: write text和a: append text。 **说明** 对于写操作，会检测文件类型，如果写入elf文件，将会报错并退出。

- PKG_UTIL.FILE_SET_MAX_LINE_SIZE

  设置写入文件一行的最大长度。

  PKG_UTIL.FILE_SET_MAX_LINE_SIZE函数原型为：

```
PKG_UTIL.FILE_SET_MAX_LINE_SIZE(
max_line_size IN INTEGER)
RETURN BOOL
```


版权所有 © 华为云计算技术有限公司

表 2-49 PKG_UTIL.FILE_SET_MAX_LINE_SIZE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
max_line_size	INTEGER	IN	是	每行最大字符数，包含换行符（最小值是1，最大值是32767）。如果没有指定，会指定一个默认值1024。

- PKG_UTIL.FILE_IS_CLOSE

  检测一个文件句柄是否关闭。

  PKG_UTIL.FILE_IS_CLOSE函数原型为：

```
PKG_UTIL.FILE_IS_CLOSE(
file IN INTEGER
)
RETURN BOOL
```

表 2-50 PKG_UTIL.FILE_IS_CLOSE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。

- PKG_UTIL.FILE_READ

  根据指定的长度从一个打开的文件句柄中读取出数据。

  PKG_UTIL.FILE_READ函数原型为：

```
PKG_UTIL.FILE_READ(
file     IN   INTEGER,
buffer   OUT  TEXT,
len      IN   BIGINT DEFAULT 1024)
```

表 2-51 PKG_UTIL.FILE_READ 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	通过调用OPEN打开的文件句柄，文件必须以读的模式打开，否则会抛出INVALID_OPERATION的异常。
buffer	TEXT	OUT	否	用于接收数据的BUFFER。
len	BIGINT	IN	否	从文件中读取的字节数。

版权所有 © 华为云计算技术有限公司

- PKG_UTIL.FILE_READLINE

  根据指定的长度从一个打开的文件句柄中读取出一行数据。

  PKG_UTIL.FILE_READLINE函数原型为：

```
PKG_UTIL.FILE_READLINE(
file    IN  INTEGER,
buffer  OUT TEXT,
len     IN  INTEGER DEFAULT NULL)
```

  表 2-52 PKG_UTIL.FILE_READLINE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	通过调用OPEN打开的文件句柄，文件必须以读的模式打开，否则会抛出INVALID_OPERATION的异常。
buffer	TEXT	OUT	否	用于接收数据的BUFFER。
len	INTEGER	IN	是	从文件中读取的字节数，默认是NULL。如果是默认NULL，会使用max_line_size来指定大小。

- PKG_UTIL.FILE_WRITE

  将BUFFER中指定的数据写入到文件中。

  PKG_UTIL.FILE_WRITE函数原型为：

```
PKG_UTIL.FILE_WRITE(
file IN INTEGER,
buffer IN TEXT
)
RETURN BOOL
```

  表 2-53 PKG_UTIL.FILE_WRITE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。
buffer	TEXT	IN	否	要写入文件的文本数据，BUFFER的最大值是32767个字节。如果没有指定值，默认是1024个字节，没有刷新到文件之前，一系列的PUT操作的BUFFER总和不能超过32767个字节。 **说明** 对于写操作，会检测文件类型，如果写入elf文件，将会报错并退出。

版权所有 © 华为云计算技术有限公司

- PKG_UTIL.FILE_NEWLINE
  向一个打开的文件中写入一个行终结符。行终结符和平台相关。
  PKG_UTIL.FILE_NEWLINE函数原型为：

```
PKG_UTIL.FILE_NEWLINE(
file IN INTEGER
)
RETURN BOOL
```

表 2-54 PKG_UTIL.FILE_NEWLINE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。

- PKG_UTIL.FILE_WRITELINE
  向一个打开的文件中写入一行。
  PKG_UTIL.FILE_WRITELINE函数原型为：

```
PKG_UTIL.FILE_WRITELINE(
file IN INTEGER,
buffer IN TEXT
)
RETURN BOOL
```

表 2-55 PKG_UTIL.FILE_WRITELINE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。
buffer	TEXT	IN	否	要写入的内容。

- PKG_UTIL.FILE_READ_RAW
  从一个打开的文件句柄中读取指定长度的二进制数据，返回读取的二进制数据，返回类型为raw。
  PKG_UTIL.FILE_READ_RAW函数原型为：

```
PKG_UTIL.FILE_READ_RAW(
file      IN INTEGER,
length    IN INTEGER DEFAULT NULL
)
RETURN RAW
```


版权所有 © 华为云计算技术有限公司

表 2-56 PKG_UTIL.FILE_READ_RAW 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。
length	INTEGER	IN	是	要读取的长度，默认为NULL。默认情况下读取文件中所有数据，最大为1G。

- PKG_UTIL.FILE_WRITE_RAW

  向一个打开的文件中写入传入二进制对象RAW。插入成功返回true。

  PKG_UTIL.FILE_WRITE_RAW函数原型为：

```
PKG_UTIL.FILE_WRITE_RAW(
file IN INTEGER,
r    IN RAW
)
RETURN BOOL
```

表 2-57 PKG_UTIL.FILE_WRITE_RAW 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。
r	RAW	IN	否	准备传入文件的数据。 **说明** 对于写操作，会检测文件类型，如果写入elf文件，将会报错并退出。

- PKG_UTIL.FILE_FLUSH

  一个文件句柄中的数据要写入到物理文件中，缓冲区中的数据必须要有一个行终结符。当文件必须在打开时读取，刷新非常有用。例如，调试信息可以刷新到文件中，以便立即读取。

  PKG_UTIL.FILE_FLUSH函数原型为：

```
PKG_UTIL.FILE_FLUSH(
file IN INTEGER
)
RETURN VOID
```


版权所有 © 华为云计算技术有限公司

表 2-58 PKG_UTIL.FILE_FLUSH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。

- PKG_UTIL.FILE_CLOSE
  关闭一个打开的文件句柄。
  PKG_UTIL.FILE_CLOSE函数原型为：

```
PKG_UTIL.FILE_CLOSE(
file IN INTEGER
)
RETURN BOOL
```

表 2-59 PKG_UTIL.FILE_CLOSE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。

- PKG_UTIL.FILE_REMOVE
  删除一个磁盘文件，操作的时候需要有充分的权限。
  PKG_UTIL.FILE_REMOVE函数原型为：

```
PKG_UTIL.FILE_REMOVE(
file_name IN TEXT
)
RETURN VOID
```

表 2-60 PKG_UTIL.FILE_REMOVE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file_name	TEXT	IN	否	要删除的文件名。

- PKG_UTIL.FILE_RENAME
  对于磁盘上的文件进行重命名，类似Unix的mv。
  PKG_UTIL.FILE_RENAME函数原型为：

版权所有 © 华为云计算技术有限公司

```
PKG_UTIL.FILE_RENAME(
src_dir IN TEXT,
src_file_name IN TEXT,
dest_dir IN TEXT,
dest_file_name IN TEXT,
overwrite BOOLEAN DEFAULT false)
```

表 2-61 PKG_UTIL.FILE_RENAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
src_dir	TEXT	IN	否	源文件目录（大小写敏感）。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开guc参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
src_file_name	TEXT	IN	否	源文件名。
dest_dir	TEXT	IN	否	目标文件目录（大小写敏感）。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开guc参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
dest_file_name	TEXT	IN	否	目标文件名。
overwrite	BOOLEAN	IN	是	默认是false，如果目的目录下存在一个同名的文件，不会进行重写。

- PKG_UTIL.FILE_SIZE

  返回指定的文件大小。

  PKG_UTIL.FILE_SIZE函数原型为：

  ```
  bigint PKG_UTIL.FILE_SIZE(
  file_name IN TEXT
  )return BIGINT
  ```


版权所有 © 华为云计算技术有限公司

表 2-62 PKG_UTIL.FILE_SIZE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file_name	TEXT	IN	否	文件名。

- PKG_UTIL.FILE_BLOCK_SIZE

  返回指定的文件含有的块数量。

  PKG_UTIL.FILE_BLOCK_SIZE函数原型为：

  ```
  bigint PKG_UTIL.FILE_BLOCK_SIZE(
  file_name IN TEXT
  )return BIGINT
  ```

  表 2-63 PKG_UTIL.FILE_BLOCK_SIZE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file_name	TEXT	IN	否	文件名。

- PKG_UTIL.FILE_EXISTS

  判断指定的文件是否存在。

  PKG_UTIL.FILE_EXISTS函数原型为：

  ```
  PKG_UTIL.FILE_EXISTS(
  file_name IN TEXT
  )
  RETURN BOOL
  ```

  表 2-64 PKG_UTIL.FILE_EXISTS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file_name	TEXT	IN	否	文件名。

- PKG_UTIL.FILE_GETPOS

  返回文件的偏移量，单位字节。

  PKG_UTIL.FILE_GETPOS函数原型为：

  ```
  PKG_UTIL.FILE_GETPOS(
  file IN INTEGER
  )
  RETURN BIGINT
  ```


版权所有 © 华为云计算技术有限公司

表 2-65 PKG_UTIL.FILE_GETPOS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。

- PKG_UTIL.FILE_SEEK

  根据用户指定的字节数向前或者向后调整文件指针的位置。

  PKG_UTIL.FILE_SEEK函数原型为：

  ```
  void PKG_UTIL.FILE_SEEK(
  file IN INTEGER,
  start IN BIGINT
  )
  RETURN VOID
  ```

  表 2-66 PKG_UTIL.FILE_SEEK 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER	IN	否	一个打开的文件句柄。
start	BIGINT	IN	否	文件偏移，字节。

- PKG_UTIL.FILE_CLOSE_ALL

  关闭一个会话中打开的所有的文件句柄。该函数无入参、出参。

  PKG_UTIL.FILE_CLOSE_ALL函数原型为：

  ```
  PKG_UTIL.FILE_CLOSE_ALL(
  )
  RETURN VOID
  ```

- PKG_UTIL.EXCEPTION_REPORT_ERROR

  抛出一个异常。

  PKG_UTIL.EXCEPTION_REPORT_ERROR函数原型为：

  ```
  PKG_UTIL.EXCEPTION_REPORT_ERROR(
  code INTEGER,
  log TEXT,
  flag BOOLEAN DEFAULT FALSE
  )
  RETURN INTEGER
  ```


版权所有 © 华为云计算技术有限公司

表 2-67 PKG_UTIL.EXCEPTION_REPORT_ERROR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
code	INTEGER	IN	否	运行异常所打印的错误码。
log	TEXT	IN	否	运行异常所打印的日志提示信息。
flag	BOOLEAN	IN	是	保留字段，默认为FALSE。

- PKG_UTIL.APP_READ_CLIENT_INFO
  读取client_info信息。
  PKG_UTIL.APP_READ_CLIENT_INFO函数原型为：

```
PKG_UTIL.APP_READ_CLIENT_INFO(
OUT buffer TEXT
)return TEXT
```

表 2-68 PKG_UTIL.APP_READ_CLIENT_INFO 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
buffer	TEXT	OUT	否	返回的client_info信息

- PKG_UTIL.APP_SET_CLIENT_INFO
  设置client_info信息。
  PKG_UTIL.APP_SET_CLIENT_INFO函数原型为：

```
PKG_UTIL.APP_SET_CLIENT_INFO(
str TEXT
)
```

表 2-69 PKG_UTIL.APP_SET_CLIENT_INFO 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
str	TEXT	IN	否	要设置的client_info信息

- PKG_UTIL.LOB_CONVERTTOBLOB
  将clob转成blob，amount为要转换的长度。
  PKG_UTIL.LOB_CONVERTTOBLOB函数原型为：

版权所有 © 华为云计算技术有限公司

```
PKG_UTIL.LOB_CONVERTTOBLOB(
dest_lob BLOB,
src_clob CLOB,
amount INTEGER,
dest_offset INTEGER,
src_offset INTEGER
)return RAW
```

表 2-70 PKG_UTIL.LOB_CONVERTTOBLOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	BLOB	IN	否	目标LOB。
src_clob	CLOB	IN	否	要转换的CLOB。
amount	INTEGER	IN	否	转换的长度。长度小于1或者大于32767则报错。
dest_offset	INTEGER	IN	否	目标LOB的起始位置。偏移量小于1或者大于1073741312则报错。
src_offset	INTEGER	IN	否	源CLOB的起始位置。偏移量小于1或者大于1073741312则报错。

- PKG_UTIL.LOB_CONVERTTOCLOB

  将BLOB转成CLOB，amount为要转换的长度。

  PKG_UTIL.LOB_CONVERTTOCLOB函数原型为：

```
PKG_UTIL.LOB_CONVERTTOCLOB(
dest_lob CLOB,
src_blob BLOB,
amount INTEGER,
dest_offset INTEGER,
src_offset INTEGER
)return TEXT
```

表 2-71 PKG_UTIL.LOB_CONVERTTOCLOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dest_lob	CLOB	IN	否	目标LOB。
src_blob	BLOB	IN	否	要转换的BLOB。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
amount	INTEGER	IN	否	转换的长度。长度小于1或者大于32767则报错。
dest_offset	INTEGER	IN	否	目标lob的起始位置。偏移量小于1或者大于1073741312则报错。
src_offset	INTEGER	IN	否	源clob的起始位置。偏移量小于1或者大于1073741312则报错。

- PKG_UTIL.LOB_TEXTTORAW

  将输入的CLOB文本对象转换为RAW输出，奇数位自动前置补零。输入的文本必须由合法的十六进制数字组成（0-9, A-F, a-f），如果包含任何非法字符会报错。

  PKG_UTIL.LOB_TEXTTORAW函数原型为：

  ```
  PKG_UTIL.LOB_TEXTTORAW(
  src_lob CLOB
  )
  RETURN RAW
  ```

  表 2-72 PKG_UTIL.LOB_TEXTTORAW 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
src_lob	CLOB	IN	否	要转换的LOB数据。

- PKG_UTIL.LOB_RAWTOTEXT

  将输入的BLOB对象转换为十六进制文本输出，奇数位自动前置补零。输入必须由合法的十六进制数字组成（0-9, A-F, a-f），如果包含任何非法字符会报错。

  PKG_UTIL.LOB_RAWTOTEXT函数原型为：

  ```
  PKG_UTIL.LOB_RAWTOTEXT(
  src_lob IN BLOB
  )
  RETURN TEXT
  ```


版权所有 © 华为云计算技术有限公司

表 2-73 PKG_UTIL.LOB_RAWTOTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
src_lob	BLOB	IN	否	要转换的LOB数据。

- PKG_UTIL.MATCH_EDIT_DISTANCE_SIMILARITY
  计算两个字符串的差别。
  PKG_UTIL.MATCH_EDIT_DISTANCE_SIMILARITY函数原型为：

```
PKG_UTIL.MATCH_EDIT_DISTANCE_SIMILARITY(
str1 TEXT,
str2 TEXT
)
RETURN INTEGER
```

表 2-74 PKG_UTIL.MATCH_EDIT_DISTANCE_SIMILARITY 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
str1	TEXT	IN	否	第一个字符串。
str2	TEXT	IN	否	第二个字符串。

- PKG_UTIL.RAW_CAST_TO_VARCHAR2
  raw类型转成varchar2。
  PKG_UTIL.RAW_CAST_TO_VARCHAR2函数原型为：

```
PKG_UTIL.RAW_CAST_TO_VARCHAR2(
str RAW
)
RETURN VARCHAR2
```

表 2-75 PKG_UTIL.RAW_CAST_TO_VARCHAR2 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
str	RAW	IN	否	十六进制字符串。

- PKG_UTIL.SESSION_CLEAR_CONTEXT
  清除session_context信息。

版权所有 © 华为云计算技术有限公司

PKG_UTIL.SESSION_CLEAR_CONTEXT函数原型为：

```
PKG_UTIL.SESSION_CLEAR_CONTEXT(
namespace TEXT,
client_identifier TEXT,
attribute TEXT
)
RETURN INTEGER
```

表 2-76 PKG_UTIL.SESSION_CLEAR_CONTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
namespace	TEXT	IN	否	属性的命名空间。
client_identifier	TEXT	IN	是	client_identifier，一般与namespace即可，当为null时，默认修改所有namespace。
attribute	TEXT	IN	否	要清除的属性值。

- PKG_UTIL.SESSION_SEARCH_CONTEXT

  查找属性值。

  PKG_UTIL.SESSION_SEARCH_CONTEXT函数原型为：

  ```
  PKG_UTIL.SESSION_SEARCH_CONTEXT(
  namespace TEXT,
  attribute TEXT
  )
  RETURN INTEGER
  ```

  表 2-77 PKG_UTIL.SESSION_SEARCH_CONTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
namespace	TEXT	IN	否	属性的命名空间。
attribute	TEXT	IN	否	要查找的属性值。

- PKG_UTIL.SESSION_SET_CONTEXT

  设置属性值。

  PKG_UTIL.SESSION_SET_CONTEXT函数原型为：

  ```
  PKG_UTIL.SESSION_SET_CONTEXT(
  namespace TEXT,
  attribute TEXT,
  value TEXT
  )
  RETURN INTEGER
  ```


版权所有 © 华为云计算技术有限公司

表 2-78 PKG_UTIL.SESSION_SET_CONTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
namespace	TEXT	IN	否	属性的命名空间。
attribute	TEXT	IN	否	要设置的属性。
value	TEXT	IN	否	属性对应的值。

- PKG_UTIL.UTILITY_GET_TIME

  打印unix时间戳。

  PKG_UTIL.UTILITY_GET_TIME函数原型为：

  ```
  PKG_UTIL.UTILITY_GET_TIME()
  RETURN BIGINT
  ```

- PKG_UTIL.UTILITY_FORMAT_ERROR_BACKTRACE

  查看存储过程的错误堆栈。

  PKG_UTIL.UTILITY_FORMAT_ERROR_BACKTRACE函数原型为：

  ```
  PKG_UTIL.UTILITY_FORMAT_ERROR_BACKTRACE()
  RETURN TEXT
  ```

- PKG_UTIL.UTILITY_FORMAT_ERROR_STACK

  查看存储过程的报错信息。

  PKG_UTIL.UTILITY_FORMAT_ERROR_STACK函数原型为：

  ```
  PKG_UTIL.UTILITY_FORMAT_ERROR_STACK()
  RETURN TEXT
  ```

- PKG_UTIL.UTILITY_FORMAT_CALL_STACK

  查看存储过程调用堆栈。

  PKG_UTIL.UTILITY_FORMAT_CALL_STACK函数原型为：

  ```
  PKG_UTIL.UTILITY_FORMAT_CALL_STACK()
  RETURN TEXT
  ```

- PKG_UTIL.UTILITY_COMPILE_SCHEMA

  重编译指定schema中的包函数和存储过程。

  PKG_UTIL.UTILITY_COMPILE_SCHEMA函数原型为：

  ```
  PKG_UTIL.UTILITY_COMPILE_SCHEMA (
  schema IN VARCHAR2,
  compile_all IN BOOLEAN DEFAULT TRUE,
  reuse_settings IN BOOLEAN DEFAULT FALSE
  )
  RETURNS VOID
  ```

- PKG_UTIL.GS_COMPILE_SCHEMA

  重编译指定schema中的包函数和存储过程。

  PKG_UTIL.GS_COMPILE_SCHEMA存储过程原型为：

  ```
  pkg_util.GS_COMPILE_SCHEMA (
  schema_name IN VARCHAR2 DEFAULT NULL,
  compile_all IN BOOLEAN DEFAULT FALSE,
  ```


版权所有 © 华为云计算技术有限公司

```
retry_times IN INT DEFAULT 10
)
```

表 2-79 PKG_UTIL.GS_COMPILE_SCHEMA 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
schema_name	VARCHAR2	IN	是	命名空间的名称。
compile_all	BOOLEAN	IN	是	编译所有。 ● false：编译pg_object表中状态为false的包、函数、存储过程。 ● true：编译pg_object表中所有的包、函数、存储过程。 ● retry_times：重试次数。

- PKG_UTIL.APP_SET_MODULE

  设置module的值。

  PKG_UTIL.APP_SET_MODULE存储过程原型为：

  ```
  PKG_UTIL.APP_SET_MODULE (
      str TEXT)
  returns VOID
  ```

  表 2-80 PKG_UTIL.APP_SET_MODULE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
text	TEXT	IN	否	要设置module的值。

  示例：

  ```
  CALL PKG_UTIL.APP_SET_MODULE('set module');
   app_set_module
  ----------------

  (1 row)
  ```

- PKG_UTIL.APP_READ_MODULE

  读取module的值。

  PKG_UTIL.APP_READ_MODULE存储过程原型为：

  ```
  PKG_UTIL.APP_READ_MODULE(
  OUT buffer TEXT
  )
  ```


版权所有 © 华为云计算技术有限公司

表 2-81 PKG_UTIL.APP_READ_MODULE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
buffer	TEXT	OUT	否	返回的module的值。

示例：

```
CALL PKG_UTIL.APP_SET_MODULE('set module');
 app_set_module
----------------

(1 row)

DECLARE
      module VARCHAR2(64);
BEGINPKG_UTIL.APP_READ_MODULE(module);
      dbe_output.print_line(module);
END;
/
set module
ANONYMOUS BLOCK EXECUTE
```

- PKG_UTIL.APP_SET_ACTION

  设置action的值。

  PKG_UTIL.APP_SET_ACTION存储过程原型为：

  ```
  PKG_UTIL.APP_SET_ACTION (
     str TEXT)
  returns VOID
  ```

  表 2-82 PKG_UTIL.APP_SET_ACTION 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
text	TEXT	IN	否	要设置action的值。

  示例：

  ```
  CALL PKG_UTIL.APP_SET_ACTION('set action');
   app_set_action
  ----------------

  (1 row)
  ```

- PKG_UTIL.APP_READ_ACTION

  读取action的值。

  PKG_UTIL.APP_READ_ACTION存储过程原型为：

  ```
  PKG_UTIL.APP_READ_ACTION(
  OUT buffer TEXT
  )
  ```


版权所有 © 华为云计算技术有限公司

表 2-83 PKG_UTIL.APP_READ_ACTION 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
buffer	TEXT	OUT	否	返回的action的值。

示例：

```
CALL PKG_UTIL.APP_SET_ACTION('set action');
 app_set_action
----------------

(1 row)

DECLARE
       action VARCHAR2(64);
BEGINPKG_UTIL.APP_READ_ACTION(action);
      dbe_output.print_line(action);
END;
/
set action
ANONYMOUS BLOCK EXECUTE
```

- PKG_UTIL.MODIFY_PACKAGE_STATE

  用于修改当前会话的PL/SQL的状态。

  PKG_UTIL.MODIFY_PACKAGE_STATE存储过程原型为：

  ```
  PKG_UTIL.MODIFY_PACKAGE_STATE (
     flags INT)
  returns VOID
  ```

  表 2-84 PKG_UTIL.MODIFY_PACKAGE_STATE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
flags	INT	IN	否	对PL/SQL采取的操作的位标志。 当flags=1时，在执行完最顶层的PL/SQL后，会释放会话中目前运行的PL/SQL的会话缓存。清除任何包全局的当前值并关闭缓存的游标。在后续使用时，PL/SQL将重新实例化，并重新初始化包全局。 不支持其他位标志。

  示例：

  ```
  CALL PKG_UTIL.MODIFY_PACKAGE_STATE(1);
   modify_package_state
  ----------------------

  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

2.12.1.3 DBE_XML

DBE_XML支持的所有接口请参见**表2-85**：

表 2-85 DBE_XML 接口参数说明

接口名称	描述
DBE_XML.XML_FREE_PARSER	释放PARSER。
DBE_XML.XML_PARSER_GET_DOC	获取解析的document节点。
DBE_XML.XML_GET_VALIDATION_MODE	获取validate属性。
DBE_XML.XML_NEW_PARSER	新建PARSER实例。
DBE_XML.XML_PARSE_BUFFER	解析VARCHAR字符串。
DBE_XML.XML_PARSE_CLOB	解析CLOB字符串。
DBE_XML.XML_SET_VALIDATION_MODE	设置validate属性。
DBE_XML.XML_DOM_APPEND_CHILD	将newchild node添加到parent(n)节点最后面,并返回新添加的Node节点。
DBE_XML.XML_DOM_CREATE_ELEMENT	返回创建指定名称的DOMELEMENT对象。
DBE_XML.XML_DOM_CREATE_ELEMENT_NS	返回创建指定名称和命名空间的DOMELEMENT对象。
DBE_XML.XML_DOM_CREATE_TEXT_NODE	创建并返回DOMTEXT对象。
DBE_XML.XML_DOM_FREE_DOCUMENT	将指定的xmldom类型对象释放。
DBE_XML.XML_DOM_FREE_ELEMENT	将指定的xmldom类型对象释放。
DBE_XML.XML_DOM_FREE_NODE	释放DOMNODE节点。
DBE_XML.XML_DOM_FREE_NODELIST	释放DOMNODELIST节点。
DBE_XML.XML_DOM_GET_ATTRIBUTE	获取指定的xmldom类型对象的属性。
DBE_XML.XML_DOM_GET_ATTRIBUTES	将DOMNode节点属性值作为map返回。
DBE_XML.XML_DOM_GET_CHILD_NODES	将节点下的若干子节点转换成节点列表。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME	获取指定的xmldom类型对象指定子节点组成的列表。
DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME_NS	获取指定的xmldom类型对象指定命名空间指定子节点组成的列表。
DBE_XML.XML_DOM_GET_DOCUMENT_ELEMENT	返回指定DOCUMENT的首个子节点。
DBE_XML.XML_DOM_GET_FIRST_CHILD	返回node节点的第一个子节点。
DBE_XML.XML_DOM_GET_LAST_CHILD	返回node节点的最后一个子节点。
DBE_XML.XML_DOM_GET_LENGTH	根据类型节点中内容返回节点数。
DBE_XML.XML_DOM_GET_LOCALNAME	返回给定对象的本地名称。
DBE_XML.XML_DOM_GET_NAMED_ITEM	检索由名称指定的节点。
DBE_XML.XML_DOM_GET_NAMED_ITEM_NS	检索由名称和命名空间指定的节点。
DBE_XML.XML_DOM_GET_NEXT_SIBLING	返回该节点的下一个节点。
DBE_XML.XML_DOM_GET_NODE_NAME	返回节点的名称。
DBE_XML.XML_DOM_GET_NODE_TYPE	返回节点的类型。
DBE_XML.XML_DOM_GET_NODE_VALUE	返回NODE节点的值。
DBE_XML.XML_DOM_GET_PARENT_NODE	返回给定NODE节点的父节点。
DBE_XML.XML_DOM_GET_TAGNAME	获取指定的xmldom类型对象的标签名。
DBE_XML.XML_DOM_HAS_CHILD_NODES	检查DOMNODE对象是否拥有任一子节点。
DBE_XML.XML_DOM_IMPORT_NODE	该函数将节点复制到另一节点中，并将复制后的节点挂载到指定document中。
DBE_XML.XML_DOM_IS_NULL	判断给定对象是否为NULL。
DBE_XML.XML_DOM_ITEM	根据索引返回list或map中与索引对应的元素。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_XML.XML_DOM_MAKE_ELEMENT	返回转换后的DOMELEMENT对象。
DBE_XML.XML_DOM_MAKENODE	将给定对象强制转换为DOMNODE类型。
DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_EMPTY	返回新的DOMDOCUMENT对象。
DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_CLOB	返回从指定的CLOB类型创建的新DOMDOCUMENT实例对象。
DBE_XML.XML_DOM_NEW_DOCUMENT_XMLTYPE	返回从指定的XMLType类型创建的新DOMDOCUMENT实例对象。
DBE_XML.XML_DOM_SET_ATTRIBUTE	设置指定的xmldom类型对象的属性。
DBE_XML.XML_DOM_SET_CHARSET	设置DOM设置DOMDOCUMENT的CHARSET字符集。
DBE_XML.XML_DOM_SET_DOCTYPE	设置DOMDOCUMENT的外部DTD。
DBE_XML.XML_DOM_SET_NODE_VALUE	此函数用于向DOMNODE对象中设置节点的值。
DBE_XML.XML_DOM_WRITE_TO_BUFFER_DOC	将给定的DOMDOCUMENT类型对象写入缓冲区。
DBE_XML.XML_DOM_WRITE_TO_BUFFER_NODE	将给定的DOMNODE类型对象写入缓冲区。
DBE_XML.XML_DOM_WRITE_TO_CLOB_DOC	将给定的DOMDOCUMENT类型对象写入Clob。
DBE_XML.XML_DOM_WRITE_TO_CLOB_NODE	将给定的DOMNODE类型对象写入Clob。
DBE_XML.XML_DOM_WRITE_TO_FILE_DOC	使用数据库字符集将XML节点写入指定文件。
DBE_XML.XML_DOM_WRITE_TO_FILE_NODE	使用数据库字符集将XML节点写入指定文件。
DBE_XML.XML_DOM_GET_SESSION_TREE_NUM	显示当前session中所有类型的dom树的数量。
DBE_XML.XML_DOM_GET_DOC_TREES_INFO	显示document类型的dom树的内存占用、节点数量等统计信息。
DBE_XML.XML_DOM_GET_DETAIL_DOC_TREE_INFO	显示特定的document变量的各类型节点数量。

版权所有 © 华为云计算技术有限公司

- DBE_XML.XML_FREE_PARSER
  释放给定的PARSER对象。
  DBE_XML.XML_FREE_PARSER的存储过程原型为：

```
DBE_XML.XML_FREE_PARSER(
id IN RAW(13))
returns VOID;
```

表 2-86 DBE_XML.XML_FREE_PARSER 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的parser类型对象。

- DBE_XML.XML_PARSER_GET_DOC
  XML_PARSER_GET_DOC返回PARSER构建的DOM树文档的根节点。
  DBE_XML.XML_PARSER_GET_DOC的函数原型为：

```
DBE_XML.XML_PARSER_GET_DOC(
id IN RAW(13))
returns RAW(13);
```

表 2-87 DBE_XML.XML_PARSER_GET_DOC 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的parser类型对象。

说明

- DBE_XML.XML_PARSER_GET_DOC函数传空，返回NULL。
- DBE_XML.XML_PARSER_GET_DOC函数传入的parser还没有解析文档，返回NULL。

- DBE_XML.XML_GET_VALIDATION_MODE
  获取给定Parser的解析验证模式。如果DTD验证开启返回TRUE，否则返回FALSE。
  DBE_XML.XML_GET_VALIDATION_MODE的函数原型为：

```
DBE_XML.XML_GET_VALIDATION_MODE(
id RAW(13))
returns BOOL;
```

表 2-88 DBE_XML.XML_GET_VALIDATION_MODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的parser类型对象。

- DBE_XML.XML_NEW_PARSER
  新建Parser对象，返回一个新的解析器实例。

版权所有 © 华为云计算技术有限公司

DBE_XML.XML_NEW_PARSER的函数原型为：

```
DBE_XML.XML_NEW_PARSER()
RETURNS RAW(13);
```

- DBE_XML.XML_PARSE_BUFFER

  XML_PARSE_BUFFER解析存储在字符串中的XML文档。

  DBE_XML.XML_PARSE_BUFFER的存储过程原型为：

  ```
  DBE_XML.XML_PARSE_BUFFER(
  id RAW(13),
  xmlstr VARCHAR2)
  RETURNS VOID;
  ```

  表 2-89 DBE_XML.XML_PARSE_BUFFER 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的parser类型对象。
xmlstr	VARCHAR2	IN	否	存储XML文档的字符串。

  说明

  - xml_parse_buffer函数能够解析的字符串最大长度为32767，超过最大长度解析报错。
  - 与ORA数据库差异：字符串encoding只支持UTF-8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。
  - 与ORA数据库DTD校验差异：
    - !ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而ORA数据库不报错。
    - <!ENTITY baidu "www.baidu.com">..... &Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而ORA数据库不报错。
  - 与ORA数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而ORA数据库会报错。
  - 与ORA数据库xml预定义实体解析差异：'"会被解析转义为字符’”，而ORA数据库中预定义实体统一都没有转义为字符。

- DBE_XML.XML_PARSE_CLOB

  XML_PARSE_CLOB解析存储在Clob中的XML文档。

  DBE_XML.XML_PARSE_CLOB的存储过程原型为：

  ```
  DBE_XML.XML_PARSE_CLOB(
  id IN RAW(13),
  doc IN CLOB)
  returns VOID;
  ```


版权所有 © 华为云计算技术有限公司

表 2-90 DBE_XML.XML_PARSE_CLOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的parser类型对象。
doc	CLOB	IN	否	存储XML文档的字符串。

说明

- xml_parse_clob不支持解析大于1GB的CLOB。
- 与ORA数据库差异：字符串encoding只支持UTF-8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。
- 与ORA数据库DTD校验差异：
  - !ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而ORA数据库不报错。
  - <!ENTITY baidu "www.baidu.com">..... &Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而ORA数据库不报错。
- 与ORA数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而ORA数据库会报错。
- 与ORA数据库XML预定义实体解析差异：'"会被解析转义为字符’”，而ORA数据库中预定义实体统一都没有为字符。

- DBE_XML.XML_SET_VALIDATION_MODE

  设置给定Parser的解析验证模式。

  DBE_XML.XML_SET_VALIDATION_MODE的存储过程原型为：

```
DBE_XML.XML_SET_VALIDATION_MODE(
id RAW(13),
validate BOOLEAN)
returns VOID;
```

表 2-91 DBE_XML.XML_SET_VALIDATION_MODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的parser类型对象。
validate	BOOLEAN	IN	是	要设置的模式： ● TRUE：开启DTD验证。 ● FALSE：不开启验证。

版权所有 © 华为云计算技术有限公司

说明

- DBE_XML.XML_SET_VALIDATION_MODE函数validate传入为空，不改变parser的解析验证模式。
- parser初始化默认为开启DTD验证模式。

- DBE_XML.XML_DOM_APPEND_CHILD

  将newchild node添加到parent(n)节点最后面,并返回新添加的Node节点。

  DBE_XML.XML_DOM_APPEND_CHILD的存储过程原型为：

```
DBE_XML.XML_DOM_APPEND_CHILD(
    parentId IN RAW(13),
    childId IN RAW(13)
)
RETURNS RAW(13);
```

表 2-92 DBE_XML.XML_DOM_APPEND_CHILD 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
parentId	RAW(13)	IN	否	指定的xmldom类型对象。
childId	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_CREATE_ELEMENT

  返回创建指定名称的DOMELEMENT对象。

  DBE_XML.XML_DOM_CREATE_ELEMENT的函数原型为：

```
DBE_XML.XML_DOM_CREATE_ELEMENT(
    id IN RAW(13),
    tagname IN VARCHAR2
)
RETURNS RAW(13);
```

表 2-93 DBE_XML.XML_DOM_CREATE_ELEMENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
tagname	VARCHAR2	IN	否	新建的DOMELEMENT名称。

- DBE_XML.XML_DOM_CREATE_ELEMENT_NS

  返回创建指定名称和命名空间的DOMELEMENT对象。

  DBE_XML.XML_DOM_CREATE_ELEMENT_NS的函数原型为：

```
DBE_XML.XML_DOM_CREATE_ELEMENT_NS(
    id IN RAW(13),
```


版权所有 © 华为云计算技术有限公司

```
    tagname IN VARCHAR2,
    ns IN VARCHAR2
)
RETURNS RAW(13);
```

表 2-94 DBE_XML.XML_DOM_CREATE_ELEMENT_NS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
tagname	VARCHAR2	IN	否	新建的DOMELEMENT名称。
ns	VARCHAR2	IN	否	命名空间。

- DBE_XML.XML_DOM_CREATE_TEXT_NODE

  创建并返回DOMTEXT对象。

  DBE_XML.XML_DOM_CREATE_TEXT_NODE的函数原型为：

```
DBE_XML.XML_DOM_CREATE_TEXT_NODE(
    id IN RAW(13),
    data IN VARCHAR2
)
RETURNS RAW(13);
```

表 2-95 DBE_XML.XML_DOM_CREATE_TEXT_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
data	VARCHAR2	IN	否	新建的DOMTEXT节点内容。

- DBE_XML.XML_DOM_FREE_DOCUMENT

  将指定的xmldom类型对象释放。

  DBE_XML.XML_DOM_FREE_DOCUMENT的存储过程原型为：

```
DBE_XML.XML_DOM_FREE_DOCUMENT(
    id RAW(13)
)
RETURNS VOID;
```

表 2-96 DBE_XML.XML_DOM_FREE_DOCUMENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

版权所有 © 华为云计算技术有限公司

- DBE_XML.XML_DOM_FREE_ELEMENT
  将指定的xmldom类型对象释放
  DBE_XML.XML_DOM_FREE_ELEMENT的存储过程原型为：

```
DBE_XML.XML_DOM_FREE_ELEMENT (
    id RAW(13)
)
RETURNS VOID;
```

表 2-97 DBE_XML.XML_DOM_FREE_ELEMENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_FREE_NODE
  释放DOMNODE节点。
  DBE_XML.XML_DOM_FREE_NODE的函数原型为：

```
DBE_XML.XML_DOM_FREE_NODE (
    id RAW(13)
)
RETURNS VOID;
```

表 2-98 DBE_XML.XML_DOM_FREE_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_FREE_NODELIST
  释放DOMNODELIST节点
  DBE_XML.XML_DOM_FREE_NODELIST的存储过程原型为：

```
DBE_XML.XML_DOM_FREE_NODELIST(
    id IN RAW(13)

)
RETURNS VOID
```

表 2-99 DBE_XML.XML_DOM_FREE_NODELIST 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_ATTRIBUTE

版权所有 © 华为云计算技术有限公司

获取指定的xmldom类型对象的属性

DBE_XML.XML_DOM_GET_ATTRIBUTE的存储过程原型为：

```
DBE_XML.XML_DOM_GET_ATTRIBUTE (
   docid   IN  RAW(13),
   name    IN  VARCHAR2
)
RETURNS VARCHAR2;
```

表 2-100 DBE_XML.XML_DOM_GET_ATTRIBUTE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
docid	RAW(13)	IN	否	指定的xmldom类型对象。
name	VARCHAR2	IN	否	字符串。

- DBE_XML.XML_DOM_GET_ATTRIBUTES

  将DOMNode节点属性值作为map返回。

  DBE_XML.XML_DOM_GET_ATTRIBUTES的函数原型为：

  ```
  DBE_XML.XML_DOM_GET_ATTRIBUTES (
     id RAW(13)
  )
  RETURNS RAW(13);
  ```

  表 2-101 DBE_XML.XML_DOM_GET_ATTRIBUTES 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_CHILD_NODES

  将节点下的若干子节点转换成节点列表

  DBE_XML.XML_DOM_GET_CHILD_NODES的函数原型为：

  ```
  DBE_XML.XML_DOM_GET_CHILD_NODES(
     id IN RAW(13)
  )
  RETURNS RAW(13);
  ```

  表 2-102 DBE_XML.XML_DOM_GET_CHILD_NODES 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME

  获取指定的xmldom类型对象指定子节点组成的列表

版权所有 © 华为云计算技术有限公司

DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME的存储过程原型为：

```
DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME (
   docid   IN  RAW(13),
   name    IN  VARCHAR2
)
RETURNS RAW(13);
```

表 2-103 DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
docid	RAW(13)	IN	否	指定的xmldom类型对象。
name	VARCHAR2	IN	否	字符串。

- DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME_NS

  获取指定的xmldom类型对象指定命名空间指定子节点组成的列表

  DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME_NS的存储过程原型为：

  ```
  DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME_NS (
     docid   IN  RAW(13),
     name    IN  VARCHAR2,
     ns      IN  VARCHAR2
  )
  RETURNS RAW(13);
  ```

  表 2-104 DBE_XML.XML_DOM_GET_CHILDREN_BY_TAGNAME_NS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
docid	RAW(13)	IN	否	指定的xmldom类型对象。
name	VARCHAR2	IN	否	字符串。
ns	VARCHAR2	IN	是	字符串。

- DBE_XML.XML_DOM_GET_DOCUMENT_ELEMENT

  返回指定DOCUMENT的首个子节点。

  DBE_XML.XML_DOM_GET_DOCUMENT_ELEMENT的存储过程原型为：

  ```
  DBE_XML.XML_DOM_GET_DOCUMENT_ELEMENT(
     id  RAW(13)
  )
  RETURNS RAW(13);
  ```


版权所有 © 华为云计算技术有限公司

表 2-105 DBE_XML.XML_DOM_GET_DOCUMENT_ELEMENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_FIRST_CHILD
  返回node节点的第一个子节点。
  DBE_XML.XML_DOM_GET_FIRST_CHILD的函数原型为：

```
DBE_XML.XML_DOM_GET_FIRST_CHILD(
    id IN RAW(13)
)
RETURNS RAW(13);
```

表 2-106 DBE_XML.XML_DOM_GET_FIRST_CHILD 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_LAST_CHILD
  返回node节点的最后一个子节点。
  DBE_XML.XML_DOM_GET_LAST_CHILD的函数原型为：

```
DBE_XML.XML_DOM_GET_LAST_CHILD(
    id IN RAW(13)
)
RETURNS RAW(13);
```

表 2-107 DBE_XML.XML_DOM_GET_LAST_CHILD 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_LENGTH
  根据类型节点中内容返回节点数。
  DBE_XML.XML_DOM_GET_LENGTH的存储过程原型为：

```
DBE_XML.XML_DOM_GET_LENGTH(
    id RAW(13)
)
RETURNS VOID;
```


版权所有 © 华为云计算技术有限公司

表 2-108 DBE_XML.XML_DOM_GET_LENGTH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_LOCALNAME
  返回给定对象的本地名称
  DBE_XML.XML_DOM_GET_LOCALNAME的存储过程原型为：

```
DBE_XML.XML_DOM_GET_LOCALNAME (
    id  RAW(13)
)
RETURNS VARCHAR2;
```

表 2-109 DBE_XML.XML_DOM_GET_LOCALNAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_NAMED_ITEM
  检索由名称指定的节点
  DBE_XML.XML_DOM_GET_NAMED_ITEM的函数原型为：

```
DBE_XML.XML_DOM_GET_NAMED_ITEM(
    id IN RAW(13),
    nodeName IN VARCHAR2
)
RETURNS RAW(13);
```

表 2-110 DBE_XML.XML_DOM_GET_NAMED_ITEM 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
nodeName	VARCHAR2	IN	否	要检索的元素的名称。

- DBE_XML.XML_DOM_GET_NAMED_ITEM_NS
  检索由名称和命名空间指定的节点
  DBE_XML.XML_DOM_GET_NAMED_ITEM_NS的函数原型为：

```
DBE_XML.XML_DOM_GET_NAMED_ITEM_NS(
    id  RAW(13),
    nodeName IN VARCHAR2,
    ns IN VARCHAR2
)
RETURNS RAW(13);
```


版权所有 © 华为云计算技术有限公司

表 2-111 DBE_XML.XML_DOM_GET_NAMED_ITEM_NS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
nodeName	VARCHAR2	IN	否	要检索的元素的名称。
ns	VARCHAR2	IN	是	命名空间。

- DBE_XML.XML_DOM_GET_NEXT_SIBLING

  返回该节点的下一个节点。

  DBE_XML.XML_DOM_GET_NEXT_SIBLING的函数原型为：

```
DBE_XML.XML_DOM_GET_NEXT_SIBLING(
    id IN RAW(13)
)
RETURNS RAW(13);
```

表 2-112 DBE_XML.XML_DOM_GET_NEXT_SIBLING 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_NODE_NAME

  返回节点的名称。

  DBE_XML.XML_DOM_GET_NODE_NAME的函数原型为：

```
DBE_XML.XML_DOM_GET_NODE_NAME(
    id IN RAW(13)
)
RETURNS VARCHAR2;
```

表 2-113 DBE_XML.XML_DOM_GET_NODE_NAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_NODE_TYPE

  返回节点的类型。

  DBE_XML.XML_DOM_GET_NODE_TYPE的函数原型为：

```
DBE_XML.XML_DOM_GET_NODE_TYPE(
    id IN RAW(13)
)
RETURNS INTEGER;
```


版权所有 © 华为云计算技术有限公司

表 2-114 DBE_XML.XML_DOM_GET_NODE_TYPE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_NODE_VALUE
  返回NODE节点的值。
  DBE_XML.XML_DOM_GET_NODE_VALUE的存储过程原型为：

```
DBE_XML.XML_DOM_GET_NODE_VALUE(
id IN RAW(13))
RETURNS VARCHAR2;
```

表 2-115 DBE_XML.XML_DOM_GET_NODE_VALUE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_PARENT_NODE
  返回给定NODE节点的父节点。
  DBE_XML.XML_DOM_GET_PARENT_NODE的存储过程原型为：

```
DBE_XML.XML_DOM_GET_PARENT_NODE(
id IN RAW(13))
RETURNS RAW(13);
```

表 2-116 DBE_XML.XML_DOM_GET_PARENT_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_GET_TAGNAME
  获取指定的xmldom类型对象的标签名
  DBE_XML.XML_DOM_GET_TAGNAME的存储过程原型为：

```
DBE_XML.XML_DOM_GET_TAGNAME (
   docid RAW(13)
)
RETURNS VARCHAR2;
```


版权所有 © 华为云计算技术有限公司

表 2-117 DBE_XML.XML_DOM_GET_TAGNAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
docid	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_HAS_CHILD_NODES

  检查DOMNODE对象是否拥有任一子节点。

  DBE_XML.XML_DOM_HAS_CHILD_NODES的存储过程原型为：

```
DBE_XML.XML_DOM_HAS_CHILD_NODES(
id IN RAW(13))
RETURNS BOOLEAN
```

表 2-118 DBE_XML.XML_DOM_HAS_CHILD_NODES 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_IMPORT_NODE

  该函数将节点复制到另一节点中，并将复制后的节点挂载到指定document中。若被复制节点的类型不属于xmldom的constants所规定的12种类型，则直接抛出类型不支持异常。

  DBE_XML.XML_DOM_IMPORT_NODE的函数原型为：

```
DBE_XML.XML_DOM_IMPORT_NODE(
   doc_id IN RAW(13),
   node_id IN RAW(13),
   deep IN BOOLEAN
)
RETURNS RAW(13);
```

表 2-119 DBE_XML.XML_DOM_IMPORT_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
doc_id	RAW(13)	IN	否	节点挂载的文档。
node_id	RAW(13)	IN	否	将要导入的节点。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
deep	BOOLEAN	IN	否	设置递归导入： ● 如果为TRUE，则导入该节点及其所有子节点。 ● 如果为FALSE，则只导入节点本身。

- DBE_XML.XML_DOM_IS_NULL
  判断给定对象是否为NULL，如果是则返回True，否则返回false。
  DBE_XML.XML_DOM_IS_NULL的函数原型为：

```
DBE_XML.XML_DOM_IS_NULL (
   id  RAW(13)
)
RETURNS BOOLEAN;
```

表 2-120 DBE_XML.XML_DOM_IS_NULL 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_ITEM
  根据索引返回list或map中与索引对应的元素。
  DBE_XML.XML_DOM_ITEM的函数原型为：

```
DBE_XML.XML_DOM_ITEM (
   id IN RAW(13),
   index IN INTEGER
)
RETURNS RAW(13);
```

表 2-121 DBE_XML.XML_DOM_ITEM 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
index	INTEGER	IN	否	要检索的元素的索引。

- DBE_XML.XML_DOM_MAKE_ELEMENT
  返回转换后的DOMELEMENT对象。

版权所有 © 华为云计算技术有限公司

DBE_XML.XML_DOM_MAKE_ELEMENT的存储过程原型为：

```
DBE_XML.XML_DOM_MAKE_ELEMENT(
id IN RAW(13))
RETURNS RAW(13)
```

表 2-122 DBE_XML.XML_DOM_MAKE_ELEMENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

- DBE_XML.XML_DOM_MAKENODE
  将给定对象强制转换为DOMNODE类型。
  DBE_XML.XML_DOM_MAKENODE的存储过程原型为：

```
DBE_XML.XML_DOM_MAKENODE(
  id    RAW(13)
)
RETURNS DOMNODE;
```

表 2-123 DBE_XML.XML_DOM_MAKENODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_EMPTY
  返回新的DOMDOCUMENT对象。
  DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_EMPTY的函数原型为：

```
DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_EMPTY()
RETURNS RAW(13);
```

- DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_CLOB
  返回从指定的CLOB类型创建的新DOMDOCUMENT实例对象。
  DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_CLOB的函数原型为：

```
DBE_XML.XML_DOM_NEW_DOM_DOCUMENT_CLOB(
   content IN  CLOB
)
RETURNS RAW(13);
```

表 2-124 xml_dom_new_dom_document_clob 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
content	CLOB	IN	否	指定的CLOB类型。

- DBE_XML.XML_DOM_NEW_DOCUMENT_XMLTYPE
  返回从指定的XMLType类型创建的新DOMDOCUMENT实例对象。

版权所有 © 华为云计算技术有限公司

DBE_XML.XML_DOM_NEW_DOCUMENT_XMLTYPE的函数原型为：

```
DBE_XML.XML_DOM_NEW_DOCUMENT_XMLTYPE(
    content IN  CLOB
)
RETURNS RAW(13);
```

表 2-125 DBE_XML.XML_DOM_NEW_DOCUMENT_XMLTYPE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
content	CLOB	IN	否	指定的CLOB类型。

- DBE_XML.XML_DOM_SET_ATTRIBUTE

  设置指定的xmldom类型对象的属性。

  DBE_XML.XML_DOM_SET_ATTRIBUTE的存储过程原型为：

  ```
  DBE_XML.XML_DOM_SET_ATTRIBUTE(
     docid   IN  RAW(13),
     name    IN  VARCHAR2,
     value   IN  VARCHAR2
  )
  RETURNS void;
  ```

  表 2-126 DBE_XML.XML_DOM_SET_ATTRIBUTE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
docid	RAW(13)	IN	否	指定的xmldom类型对象。
name	VARCHAR2	IN	否	字符串。
value	VARCHAR2	IN	否	字符串。

- DBE_XML.XML_DOM_SET_CHARSET

  设置DOMDOCUMENT的CHARSET字符集。

  DBE_XML.XML_DOM_SET_CHARSET的函数原型为：

  ```
  DBE_XML.XML_DOM_SET_CHARSET(
     id      IN RAW(13),
     charset IN VARCHAR2
  )
  RETURNS void;
  ```

  表 2-127 DBE_XML.XML_DOM_SET_CHARSET 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
charset	VARCHAR2	IN	否	字符集。

版权所有 © 华为云计算技术有限公司

- DBE_XML.XML_DOM_SET_DOCTYPE
  设置DOMDOCUMENT的外部DTD。
  DBE_XML.XML_DOM_SET_DOCTYPE的函数原型为：

```
DBE_XML.XML_DOM_SET_DOCTYPE(
    id          IN  RAW(13),
    dtd_name    IN  VARCHAR2,
    system_id   IN  VARCHAR2,
    public_id   IN  VARCHAR2
)
RETURNS void;
```

表 2-128 DBE_XML.XML_DOM_SET_DOCTYPE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
dtd_name	VARCHAR2	IN	否	需要初始化doctype的名称。
system_id	VARCHAR2	IN	否	需要初始化doctype的system ID。
public_id	VARCHAR2	IN	否	需要初始化doctype的public ID。

- DBE_XML.XML_DOM_SET_NODE_VALUE
  此函数用于向DOMNODE对象中设置节点的值。
  DBE_XML.XML_DOM_SET_NODE_VALUE的存储过程原型为：

```
DBE_XML.XML_DOM_SET_NODE_VALUE(
id IN RAW(13),
node_value IN VARCHAR2)
RETURNS VOID
```

表 2-129 DBE_XML.XML_DOM_SET_NODE_VALUE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
node_value	VARCHAR2	IN	否	向DOMNODE对象中设置的字符串。

- DBE_XML.XML_DOM_WRITE_TO_BUFFER_DOC
  将给定的DOMDOCUMENT类型对象写入缓冲区。

版权所有 © 华为云计算技术有限公司

DBE_XML.XML_DOM_WRITE_TO_BUFFER_DOC的存储过程原型为：

```
DBE_XML.XML_DOM_WRITE_TO_BUFFER_DOC(
id IN RAW(13))
RETURNS VARCHAR2;
```

表 2-130 DBE_XML.XML_DOM_WRITE_TO_BUFFER_DOC 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_WRITE_TO_BUFFER_NODE
  将给定的DOMNODE类型对象写入缓冲区。
  DBE_XML.XML_DOM_WRITE_TO_BUFFER_NODE的存储过程原型为：

```
DBE_XML.XML_DOM_WRITE_TO_BUFFER_NODE(
id IN RAW(13))
RETURNS VARCHAR2;
```

表 2-131 DBE_XML.XML_DOM_WRITE_TO_BUFFER_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_WRITE_TO_CLOB_DOC
  将给定的DOMDOCUMENT类型对象写入Clob。
  DBE_XML.XML_DOM_WRITE_TO_CLOB_DOC的存储过程原型为：

```
DBE_XML.XML_DOM_WRITE_TO_CLOB_DOC(
    id IN RAW(13)
)
RETURNS VARCHAR2;
```

表 2-132 DBE_XML.XML_DOM_WRITE_TO_CLOB_DOC 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_WRITE_TO_CLOB_NODE
  将给定的DOMNODE类型对象写入Clob。
  DBE_XML.XML_DOM_WRITE_TO_CLOB_NODE的存储过程原型为：

```
DBE_XML.XML_DOM_WRITE_TO_CLOB_NODE(
    id          IN  RAW(13)
)
RETURNS CLOB;
```


版权所有 © 华为云计算技术有限公司

表 2-133 DBE_XML.XML_DOM_WRITE_TO_CLOB_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。

- DBE_XML.XML_DOM_WRITE_TO_FILE_DOC
  使用数据库字符集将XML节点写入指定文件。
  DBE_XML.XML_DOM_WRITE_TO_FILE_DOC的存储过程原型为：

```
DBE_XML.XML_DOM_WRITE_TO_FILE_DOC(
id IN  RAW(13),
file_dir IN  VARCHAR2)
RETURNS VOID

DBE_XML.XML_DOM_WRITE_TO_FILE_DOC(
id IN  RAW(13),
file_dir IN  VARCHAR2,
charset  IN  VARCHAR2)
RETURNS VOID PACKAGE
```

表 2-134 DBE_XML.XML_DOM_WRITE_TO_FILE_DOC 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	是	指定的xmldom类型对象。
file_dir	VARCHAR2	IN	否	要写入的文件。
charset	VARCHAR2	IN	否	指定字符集。

- DBE_XML.XML_DOM_WRITE_TO_FILE_NODE
  使用数据库字符集将XML节点写入指定文件。
  DBE_XML.XML_DOM_WRITE_TO_FILE_NODE的存储过程原型为：

```
DBE_XML.XML_DOM_WRITE_TO_FILE_NODE(
id IN  RAW(13),
filename IN  VARCHAR2)
RETURNS VOID
```

表 2-135 DBE_XML.XML_DOM_WRITE_TO_FILE_NODE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。
filename	VARCHAR2	IN	否	指定文件地址。

- DBE_XML.XML_DOM_GET_SESSION_TREE_NUM
  查询当前session中所有类型的dom树数量。

版权所有 © 华为云计算技术有限公司

DBE_XML.XML_DOM_GET_SESSION_TREE_NUM的函数原型为：

```
DBE_XML.XML_DOM_GET_SESSION_TREE_NUM()
RETURNS INTEGER
```

- DBE_XML.XML_DOM_GET_DOC_TREES_INFO

  查询当前session中Document类型的dom树信息，如内存占用等。

  DBE_XML.XML_DOM_GET_DOC_TREES_INFO的函数原型为：

  ```
  DBE_XML.XML_DOM_GET_DOC_TREES_INFO()
  RETURNS VARCHAR2
  ```

- DBE_XML.XML_DOM_GET_DETAIL_DOC_TREE_INFO

  查询传入的document内的各类型子节点的数量。

  DBE_XML.XML_DOM_GET_DETAIL_DOC_TREE_INFO的函数原型为：

  ```
  dbe_xml.xml_dom_get_detail_doc_tree_info(
  id IN  RAW(13))
  RETURNS VARCHAR2
  ```

表 2-136 DBE_XML.XML_DOM_GET_DETAIL_DOC_TREE_INFO 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	RAW(13)	IN	否	指定的xmldom类型对象。

2.12.2 二次封装接口

2.12.2.1 DBE_APPLICATION_INFO

接口介绍

高级功能包DBE_APPLICATION_INFO支持的接口请参见**表2-137**。
DBE_APPLICATION_INFO作用范围是当前session。

表 2-137 DBE_APPLICATION_INFO

接口名称	描述
DBE_APPLICATION_INFO.SET_CLIENT_INFO	写入客户端信息。
DBE_APPLICATION_INFO.READ_CLIENT_INFO	读取客户端信息。
DBE_APPLICATION_INFO.SET_MODULE	将当前正在运行的模块的名称设置为新模块。设置module和action。
DBE_APPLICATION_INFO.READ_MODULE	读取当前会话的模块和操作字段的值。
DBE_APPLICATION_INFO.SET_ACTION	设置当前模块中当前操作的名称，设置action的值。

版权所有 © 华为云计算技术有限公司

- DBE_APPLICATION_INFO.SET_CLIENT_INFO

  写入客户端信息。

  DBE_APPLICATION_INFO.SET_CLIENT_INFO原型为：

```
DBE_APPLICATION_INFO.SET_CLIENT_INFO(
    str text
)returns void;
```

  表 2-138 DBE_APPLICATION_INFO.SET_CLIENT_INFO 接口参数说明

参数	描述
str	写入的客户端信息，最长为64字节，超过64字节将被截断。

- DBE_APPLICATION_INFO.READ_CLIENT_INFO

  读取客户端信息。

  DBE_APPLICATION_INFO.READ_CLIENT_INFO原型为：

```
DBE_APPLICATION_INFO.READ_CLIENT_INFO(
 OUT client_info text);
```

  表 2-139 DBE_APPLICATION_INFO.READ_CLIENT_INFO 接口参数说明

参数	描述
client_info	客户端信息。

- DBE_APPLICATION_INFO.SET_MODULE

  将当前正在运行的模块的名称设置为新模块。

  DBE_APPLICATION_INFO.SET_MODULE原型为：

```
DBE_APPLICATION_INFO.SET_MODULE(
    IN module_name text,
    IN action_name text
);
```

  表 2-140 DBE_APPLICATION_INFO.SET_MODULE 接口参数说明

参数	描述
module_name	当前正在运行的模块的名称。最长为64字节，超过64字节将被截断。
action_name	当前模块中当前操作的名称。最长为64字节，超过64字节将被截断。

  示例：

```
CALL dbe_application_info.set_module('module_name','action_name');
 set_module
------------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
```

- DBE_APPLICATION_INFO.READ_MODULE

  读取当前会话的模块和操作字段的值。

  DBE_APPLICATION_INFO.READ_MODULE原型为：

```
DBE_APPLICATION_INFO.READ_MODULE(
    OUT module_name text,
    OUT action_name text
);
```

表 2-141 DBE_APPLICATION_INFO.READ_MODULE 接口参数说明

参数	描述
module_name	当前正在运行的模块的名称。
action_name	当前模块中当前操作的名称。

示例：

```
DECLARE
 module varchar2(64);
 action varchar2(64);
BEGIN
 dbe_application_info.read_module(module,action);
 dbe_output.print_line(module);
 dbe_output.print_line(action);
END;
/
module_name
action_name
ANONYMOUS BLOCK EXECUTE
```

- DBE_APPLICATION_INFO.SET_ACTION

  设置当前模块中当前操作的名称。

  DBE_APPLICATION_INFO.SET_ACTION原型为：

```
DBE_APPLICATION_INFO.SET_ACTION(
      IN action_name text
);
```

表 2-142 DBE_APPLICATION_INFO.SET_ACTION 接口参数说明

参数	描述
action_name	当前模块中当前操作的名称。最长为64字节，超过64字节将被截断。

示例：

```
CALL dbe_application_info.set_action('action_name');
 set_action
------------

(1 row)
```


版权所有 © 华为云计算技术有限公司

2.12.2.2 DBE_COMPRESSION

接口介绍

根据输入的参数，评估指定数据对象的采样压缩率或者获取指定行数据的压缩类型。DBE_COMPRESSION支持的接口请参见**表2-143**。

表 2-143 DBE_COMPRESSION

接口名称	描述
DBE_COMPRESSION.GET_COMPRESSION_RATIO	根据输入参数评估指定数据对象的采样压缩率。
DBE_COMPRESSION.GET_COMPRESSION_TYPE	根据输入参数获取指定行数据的压缩类型。
DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO	根据输入参数评估指定数据对象的真实压缩率。
DBE_COMPRESSION.GET_HOLE_RATIO	根据输入参数评估指定跨页透明压缩数据对象的采样空洞率。
DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS	输出turbo_compress_shared_buffers的使用情况。
DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE	根据输入参数获取指定索引页的压缩类型。

说明

GET_COMPRESSION_TYPE接口只支持分布式数据库中数据节点（DN），其他节点暂不支持。

- DBE_COMPRESSION.GET_COMPRESSION_RATIO

  根据输入参数评估指定数据对象的采样压缩率。

  DBE_COMPRESSION.GET_COMPRESSION_RATIO原型为：

```
DBE_COMPRESSION.GET_COMPRESSION_RATIO(
SCRATCHTBSNAME IN VARCHAR2,
OWNNAME        IN VARCHAR2,
OBJNAME        IN VARCHAR2,
SUBOBJNAME     IN VARCHAR2,
COMPTYPE       IN NUMBER,
BLKCNT_CMP     OUT INTEGER,
BLKCNT_UNCMP   OUT INTEGER,
ROW_CMP        OUT INTEGER,
ROW_UNCMP      OUT INTEGER,
CMP_RATIO      OUT NUMBER,
COMPTYPE_STR   OUT VARCHAR2,
SAMPLE_RATIO   IN NUMBER DEFAULT 20,
OBJTYPE        IN INTEGER DEFAULT 1);
```


版权所有 © 华为云计算技术有限公司

说明

- 对于跨页透明压缩表，一级分区表必须传入一级分区名称，二级分区表必须传入二级分区名称，否则会报错提示“no data found.”。对于高级压缩表，分区表必须传入分区名称（二级分区表也可传入一级分区名称），否则会报错提示“no data found.”。
- 不支持在高级压缩表上评估跨页透明压缩的压缩率，直接使用会报错提示“The ilm turbo policy is not supported here.”。

表 2-144 DBE_COMPRESSION.GET_COMPRESSION_RATIO 接口参数说明

参数	描述
SCRATCHTBSNAME	数据对象所属表空间。
OWNNAME	数据对象所有者（所属模式）。
OBJNAME	数据对象名称。
SUBOBJNAME	数据子对象名称。
COMPTYPE	压缩类型，支持： ● 1：未压缩。 ● 2：高级压缩。 ● 3：跨页透明压缩low。 ● 4：跨页透明压缩medium。 ● 5：跨页透明压缩high。 ● 6：高级压缩high。
BLKCNT_CMP	样本被压缩后占用的块数。
BLKCNT_UNCMP	样本未压缩占用的块数。
ROW_CMP	样本被压缩后单个块内可容纳的行数，跨页透明压缩显示为0。
ROW_UNCMP	样本未被压缩时单个数据块可容纳的行数，跨页透明压缩显示为0。
CMP_RATIO	压缩比，blkcnt_uncmp除以blkcnt_cmp。
COMPTYPE_STR	描述压缩类型的字符串。
SAMPLE_RATIO	采样比例，输入为0-100的整数或浮点数，对应为百分之N的采样比例。默认为20，即对20%的行数进行采样。
OBJTYPE	对象类型，支持： ● 1：表对象。

版权所有 © 华为云计算技术有限公司

- DBE_COMPRESSION.GET_COMPRESSION_TYPE

  根据输入参数获取指定行数据的压缩状态（返回1表示未压缩，2表示压缩），该接口属于运维类接口，不做可见性判断，即传入的ctid为已删除的行时，该接口依然会返回当前行在页面上最新的状态。跨页透明压缩不支持此高级包，默认返回1，无实际意义。

  DBE_COMPRESSION.GET_COMPRESSION_TYPE原型为：

```
DBE_COMPRESSION.GET_COMPRESSION_TYPE(
OWNNAME    IN VARCHAR2,
TABNAME    IN VARCHAR2,
CTID       IN TEXT,
SUBOBJNAME IN VARCHAR2 DEFAULT NULL,
BUCKETID   IN pg_catalog.int2 DEFAULT NULL))
```

  表 2-145 DBE_COMPRESSION.GET_COMPRESSION_TYPE 接口参数说明

参数	描述
OWNNAME	数据对象所有者（所属模式）。
TABNAME	数据对象名称。
CTID	目标行ctid。
SUBOBJNAME	数据子对象名称。
BUCKETID	目标行tablebucketid。

- DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO

  根据输入参数评估指定数据对象的真实压缩率，仅支持跨页透明压缩表。

  DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO原型为：

```
DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO(
OWNER_NAME       IN VARCHAR2,
OBJNAME          IN VARCHAR2,
FILEPATH         OUT VARCHAR2,
IS_COMPRESS      OUT BOOL,
FILE_COUNT       OUT INTEGER,
LOGIC_SIZE       OUT BIGINT,
PHYSIC_SIZE      OUT BIGINT,
COMPRESS_RATIO   OUT VARCHAR2);
```

  表 2-146 DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO 接口参数说明

参数	描述
OWNER_NAME	数据对象所有者（所属模式）。
OBJNAME	数据对象名称。
FILEPATH	数据对象路径。
IS_COMPRESS	压缩类型。

版权所有 © 华为云计算技术有限公司

参数	描述
FILE_COUNT	文件个数，包含数据文件和cmap文件。
LOGIC_SIZE	逻辑大小。
PHYSIC_SIZE	物理大小。
COMPRESS_RATIO	压缩比。 小表场景下，压缩比可能出现小于1的现象，属于正常现象。

说明

小表场景下，压缩比可能出现小于1的现象，属于正常现象。

示例：建表table_test，插入一条数据，使用高级包进行查询，压缩比显示小于1。

```
gaussdb=# CREATE TABLE table_test(a int) ILM ADD POLICY ROW STORE COMPRESS TURBO
HIGH UNIT AFTER 1 days OF NO MODIFICATION;
gaussdb=# INSERT INTO table_test VALUES(1);
gaussdb=# CALL DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO('public', 'table_test',
NULL,NULL,NULL,NULL,NULL,NULL);
```

- DBE_COMPRESSION.GET_HOLE_RATIO

  根据输入参数评估指定跨页透明压缩数据对象的采样空洞率，仅支持跨页透明压缩表。

  DBE_COMPRESSION.GET_HOLE_RATIO原型为：

```
DBE_COMPRESSION.GET_HOLE_RATIO(
OWNER_NAME       IN VARCHAR2,
OBJNAME          IN VARCHAR2,
FILE_COUNT       OUT INTEGER,
PHYSIC_SIZE      OUT BIGINT,
HOLE_SIZE        OUT BIGINT,
HOLE_RATIO       OUT VARCHAR2,
SAMPLE_RATIO     IN NUMBER DEFAULT 20);
```

表 2-147 DBE_COMPRESSION.GET_HOLE_RATIO 接口参数说明

参数	描述
OWNER_NAME	数据对象所有者（所属模式）。
OBJNAME	数据对象名称。
FILE_COUNT	数据对象的文件个数，只包含数据文件
PHYSIC_SIZE	物理大小。
HOLE_SIZE	空洞大小
HOLE_RATIO	空洞率

版权所有 © 华为云计算技术有限公司

参数	描述
SAMPLE_RATIO	采样率。

- DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS

  输出turbo_compress_shared_buffers的使用情况。

  DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS原型为：

```
DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS(
CAPACITY        OUT BIGINT,
MAIN_LEN        OUT BIGINT,
FREE_LEN        OUT BIGINT,
BUFFER_ACCESS   OUT BIGINT,
BUFFER_NOREADS  OUT BIGINT,
DISK_READS      OUT BIGINT,
RECYCLE_TIMES   OUT BIGINT,
RECYCLE_STEP    OUT BIGINT,
RECYCLE_WAITS   OUT BIGINT,
HIT_RATIO       OUT VARCHAR2);
```

  表 2-148 DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS 接口参数说明

参数	描述
CAPACITY	turbo_compress_shared_buffers的容量。
MAIN_LEN	lrulist中main_list的长度。
FREE_LEN	lrulist中free_list的长度。
BUFFER_ACCESS	逻辑读的次数。
BUFFER_NOREADS	新扩展出来的block个数。
DISK_READS	物理读的次数。
RECYCLE_TIMES	回收次数。
RECYCLE_STEP	回收步长。
RECYCLE_WAITS	alloc buf ctrl时的等待次数。
HIT_RATIO	在内存中的命中率。

- DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE

  查询索引页面是否已经被压缩。此高级包仅能查询到高级压缩的状态，查询不到跨页透明压缩的状态。

  DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE原型为：

```
DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE(
IDXOID          IN OID,
```


版权所有 © 华为云计算技术有限公司

```
IDXTYPE        IN TEXT,
BUCKET_ID       IN INT2 DEFAULT NULL,
BLOCK_ID       IN INT8);
```

表 2-149 DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE 接口参数说明

参数	描述
IDXOID	索引oid。
IDXTYPE	索引类型，支持普通索引（'i'）、全局索引（'l'）、分区索引（'x'）。
BUCKET_ID	如果索引的bucket存储，传入对应的bucket id，默认为NULL。
BLOCK_ID	待查询的索引block id，如果输入是小数，会四舍五入取整。

2.12.2.3 DBE_DESCRIBE

数据类型介绍

高级包DBE_DESCRIBE内置了两个数据类型，这两个数据类型使用自定义类型创建，用于DESCRIBE_PROCEDURE接口的返回值。

- DBE_DESCRIBE.NUMBER_TABLE

  该类型是NUMBER的TABLE类型，通过TABLE OF语法实现。

  DBE_DESCRIBE.NUMBER_TABLE原型为：

  ```
  CREATE TYPE DBE_DESCRIBE.NUMBER_TABLE AS TABLE OF NUMBER INDEX BY INTEGER;
  ```

- DBE_DESCRIBE.VARCHAR2_TABLE

  该类型是VARCHAR2的TABLE类型，通过TABLE OF语法实现。

  DBE_DESCRIBE.VARCHAR2_TABLE原型为：

  ```
  CREATE TYPE DBE_DESCRIBE.VARCHAR2_TABLE AS TABLE OF VARCHAR2(30) INDEX BY INTEGER;
  ```

接口介绍

高级包DBE_DESCRIBE提供了接口DBE_DESCRIBE.DESCRIBE_PROCEDURE，接口描述请参见**表2-150**。

表 2-150 DBE_DESCRIBE

接口名称	描述
DBE_DESCRIBE.DESCRIBE_PROCEDURE	用来显示存储过程或函数的参数信息。

- DBE_DESCRIBE.DESCRIBE_PROCEDURE

  存储过程DESCRIBE_PROCEDURE用来显示存储过程或函数的参数信息，如参数名、参数模式、参数位置等，将函数或存储过程的参数信息以列表的方式返回。

  DBE_DESCRIBE.DESCRIBE_PROCEDURE原型：

版权所有 © 华为云计算技术有限公司

```
DBE_DESCRIBE.DESCRIBE_PROCEDURE(
  object_name               IN  VARCHAR2,
  reserved1                 IN  VARCHAR2,
  reserved2                 IN  VARCHAR2,
  overload                  OUT DBE_DESCRIBE.NUMBER_TABLE,
  dataposition              OUT DBE_DESCRIBE.NUMBER_TABLE,
  datalevel                 OUT DBE_DESCRIBE.NUMBER_TABLE,
  argument_name             OUT DBE_DESCRIBE.VARCHAR2_TABLE,
  datatype                  OUT DBE_DESCRIBE.NUMBER_TABLE,
  default_value             OUT DBE_DESCRIBE.NUMBER_TABLE,
  in_out                    OUT DBE_DESCRIBE.NUMBER_TABLE,
  datalength                OUT DBE_DESCRIBE.NUMBER_TABLE,
  dataprecision             OUT DBE_DESCRIBE.NUMBER_TABLE,
  scale                     OUT DBE_DESCRIBE.NUMBER_TABLE,
  radix                     OUT DBE_DESCRIBE.NUMBER_TABLE,
  spare                     OUT DBE_DESCRIBE.NUMBER_TABLE,
  include_string_constraints OUT BOOLEAN
);
```

表 2-151 DBE_DESCRIBE.DESCRIBE_PROCEDURE 接口参数说明

参数	类型	是否允许为空	描述
object_name	varchar2	否	存储过程的名称。此参数的语法格式为[[schema.]package.]function[@dblink]，其中： ● schema：可选，模式名称。 ● package：可选，包名称。 ● function：不能为空，函数或存储过程名称。 ● dblink：可选，远程连接名称。
reserved1	varchar2	是	预留参数。
reserved2	varchar2	是	预留参数。
overload	number_table	是	分配给存储过程/函数的唯一编号。 如果存储过程/函数存在重载，则overload为该存储过程/函数的每个重载，以整个存储过程/函数为粒度，对overload从1开始递增编号。如果存储过程/函数不存在重载，则overload置0。
dataposition	number_table	是	表示指定参数在参数列表中的位置。
datalevel	number_table	是	置0。
argument_name	varchar2_table	是	与指定的存储过程相关联的参数名称。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
datatype	number_table	是	被指定参数的数据类型OID。
default_value	number_table	是	如果被指定的参数有一个默认值，则值为1；否则值为0。
in_out	number_table	是	指定参数的模式。 其中参数模式有： ● 0：IN。 ● 1：OUT。 ● 2：IN OUT。
datalength	number_table	是	暂不支持，默认置0。
dataprecision	number_table	是	暂不支持，默认置0。
scale	number_table	是	暂不支持，默认置0。
radix	number_table	是	若为数值类型（如NUMBER、INTEGER等），返回10，否则置0；数值类型请参阅**数值类型**。
spare	number_table	是	预留参数，默认置0。
include_string_constraints	boolean	是	预留参数，默认置FALSE。

版权所有 © 华为云计算技术有限公司

说明

- datatype参数与ORA模式数据库的数据类型存在差异。GaussDB返回数据类型的OID，ORA模式数据库返回ORA模式数据库内部的数据类型编号。
- 参数include_string_constraints作为预留参数，本身的值不会发生变化，也不会影响其它参数的返回值。
- 使用CREATE TYPE操作创建的数据类型，由于这些数据类型的OID是不确定的，所以请勿将这些OID用于固定判断等。
- 对于dataposition参数，如果指定的是存储过程，则返回值从1开始；如果指定的是函数，则返回值从0开始，其中0代表函数返回值的位置序号。
- 对于argument_name参数，如果指定的是函数，那么返回值的第一个位置为空，该位置表示被描述函数的返回值的名称（即空的名称）。
- 不可直接指定包Package，否则会报错。
- 对于没有执行权限的存储过程/函数/包，会当作不存在的实体并报错处理。
- 入参reserved1和reserved2不参与内部处理，输入任何字符串都不会对返回结果有影响。
- 该高级包不可指定通过DBLINK获取的存储过程/函数。
- 推荐在被指定的存储过程/函数前增加Schema前缀。若省略了Schema前缀，则该高级包会使用当前会话的Schema来查找其所属实体（若当前会话Schema中无该所属实体，则会遵循behavior_compat_options中配置项bind_procedure_searchpath的机制来操作）。
- 若使用%type操作从表字段中取得数据类型，则不会保留类型的约束（带约束的数据类型如NUMBER(3)、VARCHAR2(10)等）。

示例：

```
-- 创建存储过程封装该高级包，用于打印返回值。
gaussdb=# CREATE PROCEDURE PRINT_DESCRIBE (obj_name IN VARCHAR2)
AS
  a_overload      DBE_DESCRIBE.NUMBER_TABLE;
  a_position      DBE_DESCRIBE.NUMBER_TABLE;
  a_level         DBE_DESCRIBE.NUMBER_TABLE;
  a_arg_name      DBE_DESCRIBE.VARCHAR2_TABLE;
  a_dty           DBE_DESCRIBE.NUMBER_TABLE;
  a_def_val       DBE_DESCRIBE.NUMBER_TABLE;
  a_mode          DBE_DESCRIBE.NUMBER_TABLE;
  a_length        DBE_DESCRIBE.NUMBER_TABLE;
  a_precision     DBE_DESCRIBE.NUMBER_TABLE;
  a_scale         DBE_DESCRIBE.NUMBER_TABLE;
  a_radix         DBE_DESCRIBE.NUMBER_TABLE;
  a_spare         DBE_DESCRIBE.NUMBER_TABLE;
  a_include_string_constraints BOOLEAN;
BEGIN
  DBE_DESCRIBE.DESCRIBE_PROCEDURE(
    obj_name,
    null,
    null,
    a_overload,
    a_position,
    a_level,
    a_arg_name,
    a_dty,
    a_def_val,
    a_mode,
    a_length,
    a_precision,
    a_scale,
    a_radix,
    a_spare,
    a_include_string_constraints
  );
  dbe_output.print('overload                ' || chr(9));
  for indx in 1 .. a_overload.count
```


版权所有 © 华为云计算技术有限公司

```
loop
  dbe_output.print(a_overload(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('dataposition             ' || chr(9));
for indx in 1 .. a_position.count
loop
  dbe_output.print(a_position(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('datalevel                ' || chr(9));
for indx in 1 .. a_level.count
loop
  dbe_output.print(a_level(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('argument_name            ' || chr(9));
for indx in 1 .. a_arg_name.count
loop
  dbe_output.print(a_arg_name(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('default_value            ' || chr(9));
for indx in 1 .. a_def_val.count
loop
  dbe_output.print(a_def_val(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('in_out                   ' || chr(9));
for indx in 1 .. a_mode.count
loop
  dbe_output.print(a_mode(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('length                   ' || chr(9));
for indx in 1 .. a_length.count
loop
  dbe_output.print(a_length(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('precision                ' || chr(9));
for indx in 1 .. a_precision.count
loop
  dbe_output.print(a_precision(indx) || chr(9));
end loop;

dbe_output.print_line(' ');

dbe_output.print('scale                    ' || chr(9));
for indx in 1 .. a_scale.count
loop
  dbe_output.print(a_scale(indx) || chr(9));
end loop;

dbe_output.print_line(' ');
```


版权所有 © 华为云计算技术有限公司

```
  dbe_output.print('radix                    ' || chr(9));
  for indx in 1 .. a_radix.count
  loop
    dbe_output.print(a_radix(indx) || chr(9));
  end loop;

  dbe_output.print_line(' ');
END;
/
CREATE PROCEDURE

-- 创建含有三个重载的函数。
gaussdb=# CREATE OR REPLACE FUNCTION TEST_FUNC_OVERLOAD (
  param_a IN NUMBER,
  param_b IN VARCHAR2,
  param_c OUT TEXT
)
RETURN VARCHAR2 package
AS
BEGIN
  dbe_output.print_line('This procedure/function test num param.');
  RETURN 'This procedure/function test num param.';
END;
/
CREATE FUNCTION

gaussdb=# CREATE OR REPLACE FUNCTION TEST_FUNC_OVERLOAD (
  param_a IN NUMBER DEFAULT 20,
  param_b VARCHAR2 DEFAULT 'n',
  param_c IN TEXT, param_d OUT DATE,
  param_e INOUT RAW
)
RETURN VARCHAR2 package
AS
BEGIN
  dbe_output.print_line('This procedure/function test num param.');
  RETURN 'This procedure/function test num param.';
END;
/
CREATE FUNCTION

gaussdb=# CREATE OR REPLACE FUNCTION TEST_FUNC_OVERLOAD (
  param_a IN NUMBER DEFAULT 20,
  param_b VARCHAR2 DEFAULT 'n',
  param_c IN TEXT, param_d IN DATE,
  param_e OUT RAW, param_f INOUT INTEGER
)
RETURN VARCHAR2 package
AS
BEGIN
  dbe_output.print_line('This procedure/function test num param.');
  RETURN 'This procedure/function test num param.';
END;
/
CREATE FUNCTION

-- 调用上述封装。
gaussdb=# BEGIN PRINT_DESCRIBE('TEST_FUNC_OVERLOAD'); END;
/

-- 执行结果。
overload                 1    1    1    1    2    2    2    2    2    2    3    3    3    3    3    3    3
dataposition              0    1    2    3    0    1    2    3    4    5    0    1    2    3    4    5    6
datalevel                0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
argument_name                    param_a    param_b    param_c        param_a    param_b    param_c
param_d    param_e       param_a    param_b    param_c    param_d    param_e    param_f
default_value             0    0    0    0    0    1    1    0    0    0    0    1    1    0    0    0    0
in_out                   1    0    0    1    1    0    0    0    1    2    1    0    0    0    0    1    2
```


版权所有 © 华为云计算技术有限公司

```
length                  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
precision               0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
scale                   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
radix                   0   10  0   0   0   10  0   0   0   0   0   10  0   0   0   0   0   10
ANONYMOUS BLOCK EXECUTE

-- 清理。
gaussdb=# DROP FUNCTION TEST_FUNC_OVERLOAD (
  param_a IN NUMBER,
  param_b IN VARCHAR2,
  param_c OUT TEXT
);
DROP FUNCTION

gaussdb=# DROP FUNCTION TEST_FUNC_OVERLOAD (param_a IN NUMBER,
  param_b VARCHAR2,
  param_c IN TEXT,
  param_d OUT DATE,
  param_e INOUT RAW
);
DROP FUNCTION

gaussdb=# DROP FUNCTION TEST_FUNC_OVERLOAD (param_a IN NUMBER,
  param_b VARCHAR2,
  param_c IN TEXT,
  param_d IN DATE,
  param_e OUT RAW,
  param_f INOUT INTEGER
);
DROP FUNCTION

gaussdb=# DROP PROCEDURE PRINT_DESCRIBE (obj_name IN VARCHAR2);
DROP PROCEDURE
```

2.12.2.4 DBE_FILE

DBE_FILE包为存储过程提供了读取和写入操作系统文本文件的功能。

注意事项

- DBE_FILE要求通过DBE_FILE.FOPEN打开的文件必须使用数据库字符集编码，若打开的文件未按预期字符集编码，在使用DBE_FILE.READ_LINE读取时会发生编码校验错误；DBE_FILE要求通过DBE_FILE.FOPEN_NCHAR打开的文件必须使用UTF8字符集编码，若打开的文件未按预期的字符集编码，在使用DBE_FILE.READ_LINE_NCHAR读取时会发生编码校验错误。
- 使用DBE_OUTPUT.PUT_LINE打印DBE_FILE.READ_LINE_NCHAR接口结果时，需确保UTF8字符集编码能够转换为当前数据库字符集编码，满足该条件后可正常输出。DBE_OUTPUT.PRINT_LINE不支持该功能。
- DBE_FILE要求客户端字符集编码必须与数据库字符集编码保持一致。
- 当数据库字符集为ASCII编码，客户端字符集为支持中文的编码，且在客户端调用DBE_FILE.WRITE_NCHAR或DBE_FILE.WRITE_LINE_NCHAR接口写入中文内容时，若输入内容为UTF8编码格式，可能无法保证写入的内容按UTF8格式编码。这可能会导致后续使用DBE_FILE.READ_LINE_NCHAR时报错。

数据类型介绍

DBE_FILE.FILE_TYPE

DBE_FILE.FILE_TYPE类型定义了DBE_FILE包中文件的表示方式，该类型包含的字段为DBE_FILE包的私有字段，禁止直接修改DBE_FILE.FILE_TYPE类型对象的字段值。

版权所有 © 华为云计算技术有限公司

DBE_FILE.FILE_TYPE原型为：

```
CREATE TYPE DBE_FILE.FILE_TYPE AS(
    id INTEGER,
    datatype INTEGER,
    byte_mode BOOLEAN
);
```

表 2-152 DBE_FILE.FILE_TYPE 字段说明

参数	描述
id	文件句柄。
datatype	表明文件是CHAR文件、NCHAR文件或二进制文件。目前支持CHAR文件和NCHAR文件。CHAR文件返回1，NCHAR文件返回2。
byte_mode	表明文件是以二进制模式（TRUE）或文本模式（FALSE）打开。

接口介绍

高级功能包DBE_FILE支持的接口请参见表2-153。

表 2-153 DBE_FILE

接口名称	描述
DBE_FILE.OPEN/ DBE_FILE.FOPEN	根据指定的目录和文件名打开文件，返回对应的文件句柄或封装了句柄的DBE_FILE.FILE_TYPE类型对象。
DBE_FILE.IS_CLOSE	检测指定文件句柄是否关闭。
DBE_FILE.IS_OPEN	检测指定文件句柄是否打开。
DBE_FILE.READ_LINE	从一个打开的文件句柄中读取一行指定长度的数据。
DBE_FILE.WRITE	将数据写入到一个打开文件的缓冲区中。
DBE_FILE.NEW_LINE	将一个或多个行结束符写入到一个打开文件的缓冲区中。
DBE_FILE.WRITE_LINE	将数据写入到一个打开文件的缓冲区中，并自动追加一个行结束符。
DBE_FILE.FORMAT_WRITE	将数据按指定格式写入到一个打开文件的缓冲区中。
DBE_FILE.GET_RAW	从一个打开的文件中读取指定字节数的RAW类型数据。
DBE_FILE.PUT_RAW	将RAW类型数据写入到一个打开文件的缓冲区中。
DBE_FILE.FLUSH	将缓存区中的数据写入到物理文件中。
DBE_FILE.CLOSE	关闭一个打开的文件句柄。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_FILE.CLOSE_ALL	关闭一个会话中所有已打开的文件句柄。
DBE_FILE.REMOVE	根据指定目录和文件名删除一个磁盘文件，操作需具备充分权限。
DBE_FILE.RENAME	重命名一个磁盘文件，功能类似于UNIX的mv命令。
DBE_FILE.COPY	复制一个连续的区域内容到一个新创建的文件中，若未指定start_line和end_line则复制整个文件。
DBE_FILE.GET_ATTR	读取并返回磁盘文件的属性。
DBE_FILE.SEEK	根据用户指定的字节数向前或者向后调整文件指针的位置。
DBE_FILE.GET_POS	以字节为单位返回文件当前的偏移量。
DBE_FILE.FOPEN_NCHAR	以NCHAR模式根据指定的目录和文件名打开一个文件。
DBE_FILE.WRITE_NCHAR	将NVARCHAR2类型的数据写入到一个打开的NCHAR模式文件缓冲区中。
DBE_FILE.WRITE_LINE_NCHAR	将NVARCHAR2类型的数据写入到一个打开的NCHAR模式文件缓冲区中，并自动追加一个行结束符。
DBE_FILE.FORMAT_WRITE_NCHAR	将NVARCHAR2类型的数据按指定格式写入到一个打开的NCHAR模式文件缓冲区中，允许格式化的DBE_FILE.WRITE_NCHAR接口。
DBE_FILE.READ_LINE_NCHAR	从一个打开的NCHAR模式文件中读取一行指定长度的数据。

- DBE_FILE.OPEN/DBE_FILE.FOPEN

  函数DBE_FILE.OPEN用于打开一个文件，可以指定文件每行的最大字节数，同一个会话内最多可以同时打开50个文件。该函数返回一个INTEGER类型的文件句柄。函数DBE_FILE.FOPEN功能和DBE_FILE.OPEN类似，但返回一个DBE_FILE.FILE_TYPE类型的对象。

  DBE_FILE.OPEN和DBE_FILE.FOPEN原型为：

```
DBE_FILE.OPEN (
dir             IN    TEXT,
file_name       IN    TEXT,
open_mode       IN    TEXT,
max_line_size   IN    INTEGER DEFAULT 1024)
RETURN INTEGER;

DBE_FILE.FOPEN(
    dir           IN TEXT,
    file_name     IN TEXT,
    open_mode     IN TEXT,
    max_line_size IN INTEGER DEFAULT 1024)
RETURN DBE_FILE.FILE_TYPE;
```


版权所有 © 华为云计算技术有限公司

表 2-154 DBE_FILE.OPEN/DBE_FILE.FOPEN 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dir	TEXT	IN	否	文件的目录位置，该字符串表示一个目录对象名。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户仅能通过高级包读写safe_data_path指定文件路径下的文件。
file_name	TEXT	IN	否	文件名，包含扩展（文件类型），不包含路径名。如果文件名中包含路径，在OPEN中会被忽略，在Unix系统中，文件名不能以/.结尾。
open_mode	TEXT	IN	否	指定文件的打开模式，包含： • r：read text。 • w：write text。 • a：append text。 • rb：read byte。 • wb：write byte。 • ab：append byte。 **说明** 对于写操作，则检测写入文件类型，如果为elf类型文件，则报错退出。
max_line_size	INTEGER	IN	是	每行的最大字节数，包含换行符在内，最小值是1，最大值是32767。如果没有指定或指定该参数为空，则使用默认值1024。

- DBE_FILE.IS_CLOSE

  函数DBE_FILE.IS_CLOSE用于检测指定的文件句柄是否已经关闭，返回布尔值。异常情况为INVALID_FILEHANDLE。

  DBE_FILE.IS_CLOSE原型为：

```
DBE_FILE.IS_CLOSE (
    file IN INTEGER)
RETURN BOOLEAN;

DBE_FILE.IS_CLOSE(
    file IN DBE_FILE.FILE_TYPE)
RETURN BOOLEAN;
```


版权所有 © 华为云计算技术有限公司

表 2-155 DBE_FILE.IS_CLOSE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	是	待检测的文件句柄或DBE_FILE.FILE_TYPE类型的对象，为空时DBE_FILE.IS_CLOSE接口返回空。

- DBE_FILE.IS_OPEN

  函数DBE_FILE.IS_OPEN用于检测指定的文件句柄是否已经打开，返回布尔值。异常情况为INVALID_FILEHANDLE。

  DBE_FILE.IS_OPEN原型为：

```
DBE_FILE.IS_OPEN(
    file IN INTEGER)
RETURN BOOLEAN;

DBE_FILE.IS_OPEN(
    file IN DBE_FILE.FILE_TYPE)
RETURN BOOLEAN;
```

表 2-156 DBE_FILE.IS_OPEN 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	是	待检测的文件句柄或DBE_FILE.FILE_TYPE类型的对象，为空时DBE_FILE.IS_OPEN接口返回false。

- DBE_FILE.READ_LINE

  函数DBE_FILE.READ_LINE从一个打开的文件读取数据，并把读取的结果存放到buffer中。读取的时候会读取到行尾，但不包含行终结符，或者读取到文件末尾，或者读取到len参数指定的大小。读取的长度不能超过OPEN的时候指定的max_line_size。

  DBE_FILE.READ_LINE原型为：

```
DBE_FILE.READ_LINE(
    file   IN  INTEGER,
    buffer OUT TEXT,
    len    IN  INTEGER DEFAULT NULL)
RETURN TEXT;

DBE_FILE.READ_LINE(
    file   IN  DBE_FILE.FILE_TYPE,
    buffer OUT TEXT,
    len    IN  INTEGER DEFAULT NULL)
RETURN TEXT;
```


版权所有 © 华为云计算技术有限公司

表 2-157 DBE_FILE.READ_LINE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象，文件必须以读模式打开，否则会抛出INVALID_OPERATION的异常。
buffer	TEXT	OUT	否	接收数据的buffer。
len	INTEGER	IN	是	从文件中读取的字节数，默认是NULL。如果是默认NULL，会使用max_linesize来指定大小。

- DBE_FILE.WRITE

  函数DBE_FILE.WRITE用于向文件对应的缓冲区中写入buffer中的数据，文件必须以写模式打开，这个操作不会写入行终结符。

  DBE_FILE.WRITE原型为：

```
DBE_FILE.WRITE(
    file   IN INTEGER,
    buffer IN TEXT)
RETURN BOOLEAN;

DBE_FILE.WRITE(
    file   IN DBE_FILE.FILE_TYPE,
    buffer IN TEXT)
RETURN VOID;
```

表 2-158 DBE_FILE.WRITE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象，要写入的文件必须以写模式打开，这个操作不会写入行终结符。
buffer	TEXT	IN	是	写入文件的文本数据。每行的累计写入长度不能大于或等于OPEN或FOPEN时指定或默认的max_line_size，否则会在刷新到文件时报错，该参数为空时接口会直接返回。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。

版权所有 © 华为云计算技术有限公司

- DBE_FILE.NEW_LINE

  函数DBE_FILE.WRITE_LINE用于向文件对应的缓冲区中写入一个或者多个行终结符，行终结符和平台相关。

  DBE_FILE.NEW_LINE原型为：

```
DBE_FILE.NEW_LINE(
   file      IN INTEGER,
   line_nums IN INTEGER DEFAULT 1)
RETURN BOOLEAN;

DBE_FILE.NEW_LINE(
   file      IN DBE_FILE.FILE_TYPE,
   line_nums IN INTEGER DEFAULT 1)
RETURN VOID;
```

  表 2-159 DBE_FILE.NEW_LINE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象，要写入的文件必须以写模式打开，这个操作不会写入行终结符。
line_nums	INTEGER	IN	是	写入到文件中的行终结符的数量，默认值为1，指定为空时不写入行终结符。

- DBE_FILE.WRITE_LINE

  函数DBE_FILE.WRITE_LINE用于向文件对应的缓冲区中写入buffer中的数据，文件必须以写模式打开，这个操作会自动追加行终结符。

  DBE_FILE.WRITE_LINE原型为：

```
DBE_FILE.WRITE_LINE(
   file   IN INTEGER,
   buffer IN TEXT,
   flush  IN BOOLEAN DEFAULT FALSE)
RETURN BOOLEAN;

DBE_FILE.WRITE_LINE(
   file   IN DBE_FILE.FILE_TYPE,
   buffer IN TEXT,
   flush  IN BOOLEAN DEFAULT FALSE)
RETURN VOID;
```


版权所有 © 华为云计算技术有限公司

表 2-160 DBE_FILE.WRITE_LINE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。
buffer	TEXT	IN	是	要写入文件的文本数据，每行的长度（包含换行符）不能大于OPEN或FOPEN时指定或默认的max_line_size，否则会在刷新到文件时报错。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。
flush	BOOLEAN	IN	是	在WRITE_LINE后是否要将文件对应缓冲区中的数据刷到磁盘，默认值或者该参数为空时为false。

- DBE_FILE.FORMAT_WRITE

  函数DBE_FILE.FORMAT_WRITE将格式化数据写入到一个打开文件对应的缓冲区中，是允许格式化的DBE_FILE.WRITE接口。

  DBE_FILE.FORMAT_WRITE原型为：

```
DBE_FILE.FORMAT_WRITE(
   file   IN INTEGER,
   format IN TEXT,
   arg1   IN TEXT DEFAULT NULL,
   . . .
   arg6   IN TEXT DEFAULT NULL)
RETURN BOOLEAN;

DBE_FILE.FORMAT_WRITE(
   file   IN DBE_FILE.FILE_TYPE,
   format IN TEXT,
   arg1   IN TEXT DEFAULT NULL,
   . . .
   arg6   IN TEXT DEFAULT NULL)
RETURN VOID;
```

表 2-161 DBE_FILE.FORMAT_WRITE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
format	TEXT	IN	是	格式化的字符串，包含文本和格式符\n和%s。若指定为空时，则不写入任何数据。
[arg1. . .arg6]	TEXT	IN	是	1到6个可选的参数串，参数和格式化字符的位置是一一对应的，如果存在格式化字符而没有提供参数或者参数为空，会使用空串来替代%s。

- DBE_FILE.GET_RAW

  函数DBE_FILE.GET_RAW从一个打开的文件读取RAW类型数据，并把读取的结果存放到buffer中，从r中返回。

  DBE_FILE.GET_RAW原型为：

```
DBE_FILE.GET_RAW(
   file   IN  INTEGER,
   r      OUT RAW,
   length IN  INTEGER DEFAULT NULL)
RETURN RAW;

DBE_FILE.GET_RAW(
   file   IN  DBE_FILE.FILE_TYPE,
   r      OUT RAW,
   length IN  INTEGER DEFAULT NULL)
RETURN RAW;
```

  表 2-162 DBE_FILE.GET_RAW 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。
r	RAW	OUT	否	接收RAW类型数据的buffer。
length	INTEGER	IN	是	从文件中读取的字节数，默认值为NULL，如果是NULL，会使用RAW类型最大长度来指定大小。

- DBE_FILE.PUT_RAW

  函数DBE_FILE.PUT_RAW用于向文件对应的缓冲区中写入RAW类型数据。

  DBE_FILE.PUT_RAW原型为：

```
DBE_FILE.PUT_RAW (
   file  IN INTEGER,
   r     IN RAW,
   flush IN BOOLEAN DEFAULT FALSE)
```


版权所有 © 华为云计算技术有限公司

```
RETURN BOOLEAN;

DBE_FILE.PUT_RAW (
   file  IN DBE_FILE.FILE_TYPE,
   r     IN RAW,
   flush IN BOOLEAN DEFAULT FALSE)
RETURN VOID;
```

表 2-163 DBE_FILE.PUT_RAW 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。
r	RAW	IN	是	写入文件的RAW类型数据。指定为NULL时，不写入任何数据。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。
flush	BOOLEAN	IN	是	在PUT_RAW后是否要刷到磁盘，不指定或指定为空时采用FALSE。

- DBE_FILE.FLUSH

  函数DBE_FILE.FLUSH将缓冲区中的数据写入到物理文件中，缓存中的数据必须要有一个行终结符。该函数可以将缓冲区的数据及时写入到对应的物理文件中。

  DBE_FILE.FLUSH原型为：

  ```
  DBE_FILE.FLUSH(
     file IN INTEGER)
  RETURN VOID;

  DBE_FILE.FLUSH(
     file IN DBE_FILE.FILE_TYPE)
  RETURN VOID;
  ```

  表 2-164 DBE_FILE.FLUSH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。

- DBE_FILE.CLOSE

版权所有 © 华为云计算技术有限公司

函数DBE_FILE.CLOSE用于关闭一个打开的文件句柄，当调用这个函数的时候，如果还有等待写入的缓存的数据，可能会收到异常信息，正常关闭返回TRUE。

DBE_FILE.CLOSE原型为：

```
DBE_FILE.CLOSE(
    file IN INTEGER)
RETURN BOOLEAN;

DBE_FILE.CLOSE(
    file IN DBE_FILE.FILE_TYPE)
RETURN BOOLEAN;
```

表 2-165 DBE_FILE.CLOSE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。

- DBE_FILE.CLOSE_ALL

  函数DBE_FILE.CLOSE_ALL关闭一个会话中打开的所有的文件句柄，可用于紧急的清理操作。

  DBE_FILE.CLOSE_ALL原型为：

  ```
  DBE_FILE.CLOSE_ALL()
  RETURN VOID;
  ```

- DBE_FILE.REMOVE

  函数DBE_FILE.REMOVE删除一个磁盘文件。拥有相应权限的用户才可执行此函数。

  DBE_FILE.REMOVE原型为：

  ```
  DBE_FILE.REMOVE(
      dir       IN TEXT,
      file_name IN TEXT)
  RETURN VOID;
  ```


版权所有 © 华为云计算技术有限公司

表 2-166 DBE_FILE.REMOVE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dir	TEXT	IN	否	文件所在的目录位置，该字符串是一个目录对象名。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
file_name	TEXT	IN	否	文件名。

- DBE_FILE.RENAME

  函数DBE_FILE.RENAME重命名一个磁盘文件，类似Unix的mv指令。

  DBE_FILE.RENAME原型为：

```
DBE_FILE.RENAME(
    dir          IN TEXT,
    src_file_name  IN TEXT,
    dest_dir       IN TEXT,
    dest_file_name IN TEXT,
    overwrite      IN BOOLEAN DEFAULT FALSE)
RETURN VOID;
```

表 2-167 DBE_FILE.RENAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dir	TEXT	IN	否	源文件的目录位置（大小写敏感）。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
src_file_name	TEXT	IN	否	要进行命名的源文件。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
dest_dir	TEXT	IN	否	目的目录位置（大小写敏感）。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
dest_file_name	TEXT	IN	否	新的文件名。
overwrite	BOOLEAN	IN	是	是否重写，参数指定为空或者不指定时表示不重写。在不重写的情况下，如果目的目录下已存在同名文件会报错。

- DBE_FILE.COPY

  函数DBE_FILE.COPY复制一个连续区域的内容到一个新创建的文件中，如果忽略了start_line和end_line会复制整个文件。

  DBE_FILE.COPY原型为：

```
DBE_FILE.COPY (
src_dir           IN     TEXT,
src_file_name     IN     TEXT,
dest_dir          IN     TEXT,
dest_file_name    IN     TEXT,
start_line        IN     INTEGER DEFAULT 1,
end_line          IN     INTEGER DEFAULT NULL)
RETURN VOID;
```

表 2-168 DBE_FILE.COPY 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
src_dir	TEXT	IN	否	源文件所在的目录。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
src_file_name	TEXT	IN	否	要复制的源文件名。
dest_dir	TEXT	IN	否	目的文件所在的目录。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
dest_file_name	TEXT	IN	否	目的文件名。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。
start_line	INTEGER	IN	否	复制开始的行号，默认是1。
end_line	INTEGER	IN	是	复制结束的行号，默认是NULL，如果是NULL，则指定到文件尾。

- DBE_FILE.GET_ATTR

  函数DBE_FILE.GET_ATTR读取并返回一个磁盘文件的属性。

  DBE_FILE.GET_ATTR原型为：

```
DBE_FILE.GET_ATTR(
    location    IN  TEXT,
    filename    IN  TEXT,
    fexists     OUT BOOLEAN,
    file_length OUT BIGINT,
    block_size  OUT INTEGER)
RETURN RECORD;
```


版权所有 © 华为云计算技术有限公司

表 2-169 DBE_FILE.GET_ATTR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
location	TEXT	IN	否	文件所在的目录。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包操作safe_data_path指定文件路径下的文件。
filename	TEXT	IN	否	文件名。
fexists	BOOLEAN	OUT	否	文件是否存在。
file_length	BIGINT	OUT	否	文件的字节长度，如果文件不存在返回NULL。
block_size	INTEGER	OUT	否	文件系统的块大小（单位字节），如果文件不存在返回NULL。

- DBE_FILE.SEEK

  函数DBE_FILE.SEEK根据用户指定的字节数向前或者向后调整文件指针的位置。

  DBE_FILE.SEEK原型为：

```
DBE_FILE.SEEK(
    file          IN INTEGER,
    absolute_start IN BIGINT DEFAULT NULL,
    relative_start IN BIGINT DEFAULT NULL)
RETURN VOID;

DBE_FILE.SEEK(
    file          IN DBE_FILE.FILE_TYPE,
    absolute_start IN INTEGER DEFAULT NULL,
    relative_start IN INTEGER DEFAULT NULL)
RETURN VOID;
```


版权所有 © 华为云计算技术有限公司

表 2-170 DBE_FILE.SEEK 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。
absolute_start	BIGINT或INTEGER	IN	是	文件偏移的绝对位置，这个默认值为NULL。
relative_start	BIGINT或INTEGER	IN	是	文件偏移的相对位置。如果这个值是正数，向前偏移；如果是负数，向后偏移；默认值为NULL。如果和absolute_start参数同时指定，以absolute_start参数为准。

- DBE_FILE.GET_POS

  函数DBE_FILE.GET_POS以字节为单位返回文件当前的偏移量。

  DBE_FILE.GET_POS原型为：

```
DBE_FILE.GET_POS(
    file IN INTEGER)
RETURN BIGINT;

DBE_FILE.GET_POS(
    file IN DBE_FILE.FILE_TYPE)
RETURN BIGINT;
```

表 2-171 DBE_FILE.GET_POS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	INTEGER或DBE_FILE.FILE_TYPE	IN	否	通过OPEN打开的文件句柄或者FOPEN打开的DBE_FILE.FILE_TYPE类型的对象。

- DBE_FILE.FOPEN_NCHAR

  函数DBE_FILE.FOPEN_NCHAR用来打开一个文件，可以指定文件每行的最大字节数，一个会话内最多可以同时打开50个文件。该函数返回一个封装了文件句柄的DBE_FILE.FILE_TYPE类型对象。该函数以国家字符集模式打开文件以进行输入或输出。

  DBE_FILE.FOPEN_NCHAR原型为：

```
DBE_FILE.FOPEN_NCHAR(
    dir           IN TEXT,
    file_name     IN TEXT,
    open_mode     IN TEXT,
    max_line_size IN INTEGER DEFAULT 1024)
RETURN DBE_FILE.FILE_TYPE;
```


版权所有 © 华为云计算技术有限公司

表 2-172 DBE_FILE.FOPEN_NCHAR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
dir	TEXT	IN	否	文件的目录位置，该字符串是一个目录对象名。 **说明** • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包读写safe_data_path指定文件路径下的文件。
file_name	TEXT	IN	否	文件名，包含扩展（文件类型），不包括路径名。如果文件名中包含路径，在FOPEN_NCHAR中会被忽略，在Unix系统中，文件名不能以/.结尾。
open_mode	TEXT	IN	否	指定文件的打开模式，包含： • r：read text。 • w：write text。 • a：append text。 • rb：read byte。 • wb：write byte。 • ab：append byte。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。
max_line_size	INTEGER	IN	是	每行的最大字节数，包含换行符（最小值是1，最大值是32767）。如果没有指定或指定该参数为空，会使用默认值1024。

- DBE_FILE.WRITE_NCHAR

  函数DBE_FILE.WRITE_NCHAR用于向文件的缓冲区中写入buffer中的数据，文件必须以国家字符集模式和写模式打开，这个操作不会写入行终结符。文本字符串将以UTF8字符集格式写入。

  DBE_FILE.WRITE_NCHAR原型为：

```
DBE_FILE.WRITE_NCHAR(
    file   IN DBE_FILE.FILE_TYPE,
    buffer IN NVARCHAR2)
RETURN VOID;
```


版权所有 © 华为云计算技术有限公司

表 2-173 DBE_FILE.WRITE_NCHAR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	DBE_FILE.FILE_TYPE	IN	否	通过FOPEN_NCHAR打开的DBE_FILE.FILE_TYPE类型的对象，要写入的文件必须以写模式打开，这个操作不会写入行终结符。
buffer	VARCHAR2	IN	是	写入文件的文本数据。每行的累计写入长度不能大于或等于FOPEN_NCHAR时指定或默认的max_line_size，否则会在刷新到文件时报错。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。

- DBE_FILE.WRITE_LINE_NCHAR

  函数DBE_FILE.WRITE_LINE_NCHAR用于向文件的缓冲区中写入buffer中的数据，文件必须以国家字符集模式和写模式打开，这个操作会自动追加行终结符。文本字符串将以UTF8字符集格式写入。

  DBE_FILE.WRITE_LINE_NCHAR原型为：

```
DBE_FILE.WRITE_LINE_NCHAR(
   file   IN DBE_FILE.FILE_TYPE,
   buffer IN NVARCHAR2)
RETURN VOID;
```

表 2-174 DBE_FILE.WRITE_LINE_NCHAR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	DBE_FILE.FILE_TYPE	IN	否	通过FOPEN_NCHAR打开的DBE_FILE.FILE_TYPE类型的对象。
buffer	VARCHAR2	IN	是	要写入文件的文本数据，每行的长度（包含换行符）不能大于FOPEN_NCHAR时指定或默认的max_line_size，否则会在刷新到文件时报错。 **说明** 对于写操作，会检测写入文件类型，如果为elf类型文件，会报错退出。

- DBE_FILE.FORMAT_WRITE_NCHAR

  函数DBE_FILE.FORMAT_WRITE_NCHAR将格式化数据写入到一个打开的文件的缓冲区中，是允许格式化的DBE_FILE.WRITE_NCHAR接口。

  DBE_FILE.FORMAT_WRITE_NCHAR原型为：

```
DBE_FILE.FORMAT_WRITE_NCHAR(
   file   IN DBE_FILE.FILE_TYPE,
```


版权所有 © 华为云计算技术有限公司

```
   format IN NVARCHAR2,
   arg1   IN NVARCHAR2 DEFAULT NULL,
   . . .
   arg5   IN NVARCHAR2 DEFAULT NULL)
RETURN VOID;
```

表 2-175 DBE_FILE.FORMAT_WRITE_NCHAR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	DBE_FILE.FILE_TYPE	IN	否	通过FOPEN_NCHAR打开的DBE_FILE.FILE_TYPE类型的对象。
format	VARCHAR2	IN	是	格式化的字符串，包含文本和格式符\n和%s。
[arg1. . .arg5]	VARCHAR2	IN	是	1到5个可选的参数串，参数和格式化字符的位置是一一对应的，如果存在格式化字符而没有提供参数，会使用空串来替代%s。

- DBE_FILE.READ_LINE_NCHAR

  函数DBE_FILE.READ_LINE_NCHAR从一个打开的文件读取数据，并把读取的结果存放到buffer中。读取的时候会读取到行尾，但不包含行终结符，或者读取到文件末尾，或者读取到len参数指定的大小。读取的长度不能超过FOPEN_NCHAR的时候指定的max_line_size。

  DBE_FILE.READ_LINE_NCHAR原型为：

```
DBE_FILE.READ_LINE_NCHAR(
   file   IN  DBE_FILE.FILE_TYPE,
   buffer OUT NVARCHAR2,
   len    IN  INTEGER DEFAULT NULL)
RETURN NVARCHAR2;
```

表 2-176 DBE_FILE.READ_LINE_NCHAR 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
file	DBE_FILE.FILE_TYPE	IN	否	通过FOPEN_NCHAR打开的DBE_FILE.FILE_TYPE类型的对象，文件必须以读模式打开，否则会抛出INVALID_OPERATION的异常。
buffer	VARCHAR2	OUT	否	接收数据的buffer。
len	INTEGER	IN	是	从文件中读取的字节数，默认值为NULL。如果是NULL，会使用max_line_size来指定大小。

版权所有 © 华为云计算技术有限公司

示例

```
-- 系统管理员向PG_DIRECTORY系统表中加入目录/tmp/。
CREATE OR REPLACE DIRECTORY dir AS '/tmp/';
-- 执行结果。
CREATE DIRECTORY
-- 使用DBE_FILE高级包。
DECLARE
    f INTEGER;
    buffer VARCHAR2;
    raw_buffer RAW;

    f1 DBE_FILE.FILE_TYPE;
    f2 DBE_FILE.FILE_TYPE;

    fexists BOOLEAN;
    file_length BIGINT;
    block_size INTEGER;
    pos BIGINT;

    nvarchar_buffer nvarchar2;
    f_nchar DBE_FILE.FILE_TYPE;
BEGIN
    -- 打开文件。
    f := DBE_FILE.OPEN('dir', 'sample.txt', 'w');

    IF DBE_FILE.IS_OPEN(f) = true THEN
        DBE_OUTPUT.PRINT_LINE('file opened');
    END IF;

    -- 关闭文件。
    DBE_FILE.CLOSE(f);

    IF DBE_FILE.IS_CLOSE(f) = true THEN
        DBE_OUTPUT.PRINT_LINE('file closed');
    END IF;

    f := DBE_FILE.OPEN('dir', 'sample.txt', 'w');

    -- 写文件。
    DBE_FILE.WRITE(f, 'A');
    DBE_FILE.NEW_LINE(f);
    DBE_FILE.WRITE(f, 'B');
    DBE_FILE.WRITE(f, 'C');
    DBE_FILE.NEW_LINE(f, 2);
    DBE_FILE.WRITE_LINE(f, 'ABC');
    DBE_FILE.FORMAT_WRITE(f, '[1 -> %s, 2 -> %s]\n', 'GaussDB', 'DBE_FILE');
    DBE_FILE.PUT_RAW(f, '414243');
    DBE_FILE.NEW_LINE(f);
    DBE_FILE.CLOSE(f);

    -- 新建sample_copy.txt并复制sample.txt的内容。
    DBE_FILE.COPY('dir', 'sample.txt', 'dir', 'sample_copy.txt');

    -- 以读模式打开文件。
    f := DBE_FILE.OPEN('dir', 'sample_copy.txt', 'r');
    -- 读文件。
    DBE_FILE.READ_LINE(f, buffer); -- A
    DBE_OUTPUT.PRINT_LINE(buffer);
    DBE_FILE.READ_LINE(f, buffer); -- BC
    DBE_OUTPUT.PRINT_LINE(buffer);
    DBE_FILE.READ_LINE(f, buffer);
    DBE_FILE.READ_LINE(f, buffer); -- ABC
    DBE_OUTPUT.PRINT_LINE(buffer);
    DBE_FILE.READ_LINE(f, buffer); -- [1 -> GaussDB, 2 -> DBE_FILE]
    DBE_OUTPUT.PRINT_LINE(buffer);
    DBE_FILE.READ_LINE(f, buffer); -- RAW 414243 --> ABC
    DBE_OUTPUT.PRINT_LINE(buffer);

    -- 关闭文件。
```


版权所有 © 华为云计算技术有限公司

```
    DBE_FILE.CLOSE(f);

    f1 := DBE_FILE.FOPEN('dir', 'sample1.txt', 'w');
    f2 := DBE_FILE.FOPEN('dir', 'sample2.txt', 'w');
    DBE_FILE.CLOSE_ALL();

    IF DBE_FILE.IS_CLOSE(f1) = true and DBE_FILE.IS_CLOSE(f2) = true THEN
        DBE_OUTPUT.PRINT_LINE('f1 and f2 all closed');
    END IF;

    -- 删除文件。
    DBE_FILE.REMOVE('dir', 'sample1.txt');
    DBE_FILE.REMOVE('dir', 'sample2.txt');
    DBE_FILE.REMOVE('dir', 'sample_copy.txt');

    -- 打开文件，清理sample.txt里的数据。
    f := DBE_FILE.OPEN('dir', 'sample.txt', 'w');
    DBE_FILE.WRITE_LINE(f, 'ABC');
    DBE_FILE.CLOSE(f);

    f := DBE_FILE.OPEN('dir', 'sample.txt', 'r');
    -- GET_RAW。
    DBE_FILE.GET_RAW(f, raw_buffer); -- 4142430A 0A是换行符。
    DBE_OUTPUT.PRINT_LINE(raw_buffer);
    DBE_FILE.CLOSE(f);

    -- 获取文件属性。
    DBE_FILE.GET_ATTR('dir', 'sample.txt', fexists, file_length, block_size);

    IF fexists = true THEN
        DBE_OUTPUT.PRINT_LINE('file length: ' || file_length);
     END IF;

    -- 修改文件名。
    DBE_FILE.RENAME('dir', 'sample.txt', 'dir', 'sample_rename.txt', true);
    f1 := DBE_FILE.FOPEN('dir', 'sample_rename.txt', 'r');
    DBE_FILE.SEEK(f1, 1, null);
    pos := DBE_FILE.GET_POS(f1);
    DBE_OUTPUT.PRINT_LINE('position is: ' || pos);
    DBE_FILE.READ_LINE(f1, buffer); -- BC
    DBE_OUTPUT.PRINT_LINE(buffer);
    DBE_FILE.CLOSE(f1);

    -- FLUSH。
    f1 := DBE_FILE.FOPEN('dir', 'sample_rename.txt', 'w');
    DBE_FILE.WRITE_LINE(f1, 'ABCEFG');
    DBE_FILE.FLUSH(f1);

    f2 := DBE_FILE.FOPEN('dir', 'sample_rename.txt', 'r');
    DBE_FILE.READ_LINE(f2, buffer); -- ABCEFG
    DBE_OUTPUT.PRINT_LINE(buffer);
    DBE_FILE.CLOSE(f1);
    DBE_FILE.CLOSE(f2);
    DBE_FILE.REMOVE('dir', 'sample_rename.txt');

    -- NCHAR函数。
    f_nchar := DBE_FILE.FOPEN_NCHAR('dir', 'sample_nchar.txt', 'w');
    DBE_FILE.WRITE_NCHAR(f_nchar, 'ABCDE');
    DBE_FILE.WRITE_LINE_NCHAR(f_nchar, 'ABCDE');
    DBE_FILE.FORMAT_WRITE_NCHAR(f_nchar, '%s, %s', 'hello', 'world');
    DBE_FILE.CLOSE(f_nchar);
    f_nchar := DBE_FILE.FOPEN_NCHAR('dir', 'sample_nchar.txt', 'r');
    DBE_FILE.READ_LINE_NCHAR(f_nchar, nvarchar_buffer); -- ABCDEABCDE
    DBE_OUTPUT.PRINT_LINE(nvarchar_buffer);
    DBE_FILE.READ_LINE_NCHAR(f_nchar, nvarchar_buffer); -- hello, world
    DBE_OUTPUT.PRINT_LINE(nvarchar_buffer);
    DBE_FILE.CLOSE(f_nchar);
    DBE_FILE.REMOVE('dir', 'sample_nchar.txt');
END;
```


版权所有 © 华为云计算技术有限公司

```
/

-- 执行结果。
file opened
file closed
A
BC
ABC
[1 -> GaussDB, 2 -> DBE_FILE]
ABC
f1 and f2 all closed
4142430A
file length: 4
position is: 1
BC
ABCEFG
ABCDEABCDE
hello, world
ANONYMOUS BLOCK EXECUTE
```

2.12.2.5 DBE_HEAT_MAP

接口介绍

DBE_HEAT_MAP根据输入的参数，返回目标数据块中行的最后修改时间等信息，用于直观浏览每一行被判定为冷、热行的依据。该接口属于运维类接口，不做可见性判断，即传入的ctid为已删除的行时，该接口依然会返回当前行在页面上最新的状态。DBE_HEAT_MAP支持的接口请参见**表2-177**。跨页透明压缩不支持此高级包，跨页透明压缩表查询结果中行的最后修改时间为空。

表 2-177 DBE_HEAT_MAP

接口名称	描述
DBE_HEAT_MAP.ROW_HEAT_MAP	根据对象所属Schema、数据对象名称、数据对象分区名及ctid获取行的最后修改时间等信息。

- DBE_HEAT_MAP.ROW_HEAT_MAP

  根据对象所属Schema、数据对象名称、数据对象分区名及ctid获取行的最后修改时间等信息。

  DBE_HEAT_MAP.ROW_HEAT_MAP原型为：

  ```
  BE_HEAT_MAP.ROW_HEAT_MAP(
  OWNER           IN VARCHAR2,
  SEGMENT_NAME    IN VARCHAR2,
  PARTITION_NAME  IN VARCHAR2     DEFAULT NULL,
  CTID            IN TEXT,
  V_DEBUG         IN BOOL         DEFAULT FALSE,
  BUCKETID        IN pg_catalog.int2 DEFAULT NULL))
  ```

  表 2-178 DBE_HEAT_MAP.ROW_HEAT_MAP 接口参数说明

参数	描述
OWNER	数据对象所属schema。

版权所有 © 华为云计算技术有限公司

参数	描述
SEGMENT_NAME	数据对象名称。
PARTITION_NAME	数据对象分区名，可选参数，默认为NULL。
CTID	目标行的ctid，即block_id或row_id。
V_DEBUG	debug调试，增加日志打印。
BUCKETID	目标行的tablebucketid。

表 2-179 DBE_HEAT_MAP.ROW_HEAT_MAP 输出参数

参数	描述
OWNER	数据对象的所有者。
SEGMENT_NAME	数据对象名称。
PARTITION_NAME	数据对象分区名称，可选参数。
TABLESPACE_NAME	数据所属的表空间名称。
FILE_ID	行所属的绝对文件id。
RELATIVE_FNO	行所属的相对文件id（GaussDB中无此逻辑，因此取值同上）。
CTID	行的ctid，即block_id或row_id。
WRITETIME	行的最后修改时间。

说明

DBE_HEAT_MAP.ROW_HEAT_MAP 接口只支持分布式数据库中数据节点（DN），其他节点暂不支持。

示例：

```
gaussdb=# ALTER DATABASE set ilm = on;
gaussdb=# CREATE SCHEMA HEAT_MAP_DATA;
gaussdb=# SET current_schema=HEAT_MAP_DATA;

gaussdb=# CREATE TABLESPACE example1 RELATIVE LOCATION 'tablespace1';
gaussdb=# CREATE TABLE HEAT_MAP_DATA.heat_map_table(id INT, value TEXT) TABLESPACE
example1;
gaussdb=# INSERT INTO HEAT_MAP_DATA.heat_map_table VALUES (1, 'test_data_row_1');

gaussdb=# SELECT * FROM DBE_HEAT_MAP.ROW_HEAT_MAP(
    owner          =>  'heat_map_data',
    segment_name   =>  'heat_map_table',
    partition_name  => NULL,
```


版权所有 © 华为云计算技术有限公司

```
   ctid          =>  '(0,1)');
    owner     |  segment_name  | partition_name | tablespace_name | file_id | relative_fno | ctid  |
writetime
---------------+----------------+----------------+-----------------+---------+--------------+-------+-----------
 heat_map_data | heat_map_table |                | example1        |   17291 |        17291 | (0,1) |
(1 row)
```

2.12.2.6 DBE_ILM

接口介绍

服务于ILM策略实施，实现ADO Task的手动触发与停用接口。DBE_ILM支持的接口请参见**表2-180**。

表 2-180 DBE_ILM

接口名称	描述
DBE_ILM.EXECUTE_ILM	根据参数执行对指定的数据和ILM策略进行评估，评估通过则会生成对应的压缩Job。
DBE_ILM.STOP_ILM	根据参数停止正在执行的压缩Job。

- DBE_ILM.EXECUTE_ILM

  根据参数执行对指定的数据和ILM策略进行评估，评估通过则会生成对应的压缩Job。

  DBE_ILM.EXECUTE_ILM原型为：

```
DBE_ILM.EXECUTE_ILM(
OWNER          IN VARCHAR2,
OBJECT_NAME    IN VARCHAR2,
TASK_ID        IN OUT NUMBER,
SUBOBJECT_NAME IN VARCHAR2 DEFAULT NULL,
POLICY_NAME    IN VARCHAR2 DEFAULT 'ALL POLICIES',
EXECUTION_MODE IN NUMBER DEFAULT 2,
TURBO_FORCE    IN BOOLEAN DEFAULT FALSE,
TURBO_BLKSTART IN INTEGER DEFAULT -1,
TURBO_BLKEND   IN INTEGER DEFAULT -1)
```

表 2-181 DBE_ILM.EXECUTE_ILM 接口参数说明

参数	描述
OWNER	对象所属Schema。
OBJECT_NAME	对象名称。
TASK_ID	输出生成ADO task的描述符id。
SUBOBJECT_NAME	子对象名称。

版权所有 © 华为云计算技术有限公司

参数	描述
POLICY_NAME	策略名称，通过查询GS_ADM_ILMOBJECTS视图可知，默认DBE_ILM.ILM_ALL_POLICIES代表该对象上所有策略。DBE_ILM.ILM_ALL_POLICIES默认值为'ALL POLICIES'，不支持小写。
EXECUTION_MOD	执行模式，当前版本不支持该参数，仅做预留。
TURBO_FORCE	是否开启忽略冷热管理的强制压缩，仅对跨页透明压缩生效。TRUE表示开启，FALSE表示关闭。
TURBO_BLKSTART	强制压缩开始的block number，仅对跨页透明压缩生效。
TURBO_BLKEND	强制压缩结束的block number，仅对跨页透明压缩生效。

说明

- DBE_ILM.EXECUTE_ILM依据当前数据库名及表名，生成类似“数据库名_表名_分区名_数字”格式的JOBNAME命名规则，然后通过DBE_SCHEDULER.Create_job创建对应名称的任务；当表名包含特殊字符时，调用该接口将报错：Invalid program name。
- 当DBE_ILM.EXECUTE_ILM与DBE_ILM.STOP_ILM并发时，有较低概率会导致任务FAILED，gs_adm_ilmresults视图中comments字段内容显示“tuple concurrently updated”。
- 业务进行压缩操作时（无论是自动调度还是手动调度），应尽可能避免对压缩中的相关表进行DDL操作。特别是在一段长事务中执行DDL，可能会引发死锁或锁等待问题。若确有必要进行DDL操作，建议使用DBE_ILM.STOP_ILM接口将相关表的压缩任务停止，确认停止后再进行DDL操作。
- 执行DBE_ILM.EXECUTE_ILM需要拥有CREATE JOB权限，可通过DBE_SCHEDULER.GRANT_USER_AUTHORIZATION进行赋权，非SYSADMIN用户执行需要表的owner权限。
- 若要压缩二级分区，SUBOBJECT_NAME字段不可传入其一级分区名称，需传入具体的二级分区名称（或者传入NULL，即压缩所有子二级分区）。
- 事务内调用此接口，需要等待事务结束、或者COMMIT后，压缩动作才能生效。多个不同事务内调用DBE_ILM.EXECUTE_ILM接口，先执行的事务若一直不提交，后执行的事务可能存在锁超时风险。
- TURBO_FORCE参数的不同取值，对gs_adm_ilmresults视图有不同影响：
  - 当指定为FALSE时：由于gs_adm_ilmresults视图中的statistics字段统计内容本身对跨页透明压缩无意义，因此跨页透明压缩对象在该字段中显示为默认值。
  - 当指定为TRUE时：当前压缩任务独立于ILM原有的任务划分机制运行，不对该对象进行statistics统计，因此gs_adm_ilmresults视图中的statistics字段显示为空。

- DBE_ILM.STOP_ILM

  根据参数停止正在执行的ILM策略。

  DBE_ILM.STOP_ILM原型为：

```
DBE_ILM.STOP_ILM (
TASK_ID               IN NUMBER DEFAULT -1,
P_DROP_RUNNING_JOBS     IN BOOLEAN DEFAULT FALSE,
P_JOBNAME               IN VARCHAR2 DEFAULT NULL);
```


版权所有 © 华为云计算技术有限公司

表 2-182 DBE_ILM.STOP_ILM 接口参数说明

参数	描述
TASK_ID	指定待停止ADO task的描述符id。
P_DROP_RUNNING_JOBS	是否停止正在执行中的任务，TRUE为强制停止，FALSE为不停止正在执行的任务。
P_JOBNAME	标识待停止的特定JobName，通过GS_MY_ILMEVALUATIONDETAILS视图可以查询。

说明

当并发量较大时，执行DBE_ILM.STOP_ILM可能会提示资源繁忙，稍后重试即可。提示内容为“Resources are busy, please try again later.”。

示例

```
gaussdb=# CREATE DATABASE ilmtabledb with dbcompatibility = 'ORA';
gaussdb=# \c ilmtabledb
gaussdb=# ALTER DATABASE set ilm = on;
gaussdb=# CREATE Schema ILM_DATA;
gaussdb=# SET current_schema=ILM_DATA;
gaussdb=# CREATE SEQUENCE ILM_DATA.ORDER_TABLE_SE_ORDER_ID MINVALUE 1;
gaussdb=# CREATE OR REPLACE PROCEDURE ILM_DATA.ORDER_TABLE_CREATE_DATA(NUM INTEGER) IS
BEGIN
    FOR X IN 1..NUM
        LOOP
            INSERT INTO ORDER_TABLE VALUES(ORDER_TABLE_SE_ORDER_ID.nextval, '零食大礼包A', NOW());
        END LOOP;
    COMMIT;
END;
/
gaussdb=# CREATE TABLE ILM_DATA.ORDER_TABLE (ORDER_ID INT, GOODS_NAME TEXT, CREATE_TIME
TIMESTAMP) WITH (STORAGE_TYPE=ASTORE) ILM ADD POLICY ROW STORE COMPRESS ADVANCED ROW
AFTER 1 DAYS OF NO MODIFICATION;
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'order_id' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
BEGIN
    ILM_DATA.ORDER_TABLE_CREATE_DATA(100);
    PERFORM PG_SLEEP(2);
END;
/
DECLARE
    v_taskid number;
BEGIN
    DBE_ILM.EXECUTE_ILM(OWNER        => 'ilm_data',
                        OBJECT_NAME    => 'order_table',
                        TASK_ID        => v_taskid,
                        SUBOBJECT_NAME => NULL,
                        POLICY_NAME    => 'ALL POLICIES',
                        EXECUTION_MODE => 2);
    RAISE INFO 'Task ID is:%', v_taskid;
END;
/
INFO:  Task ID is:1
SELECT * FROM pg_sleep(3);
 pg_sleep
----------

(1 row)

gaussdb=# EXECUTE DIRECT ON DATANODES 'SELECT A.DBNAME, A.JOB_STATUS, A.ENABLE,
```


版权所有 © 华为云计算技术有限公司

```
A.FAILURE_MSG FROM PG_JOB A WHERE A.DBNAME = ''ilmtabledb'' AND A.JOB_NAME LIKE ''ilmjob$_%''
ORDER BY A.JOB_NAME DESC LIMIT 1';
  dbname   | job_status | enable | failure_msg
-----------+------------+--------+-------------
 dbeilmdb | s          | f      |
 dbeilmdb | s          | f      |
 dbeilmdb | s          | f      |
 dbeilmdb | s          | f      |
 dbeilmdb | s          | f      |
 dbeilmdb | s          | f      |
(6 rows)

gaussdb=# CALL DBE_ILM.STOP_ILM(-1, true, NULL);
 stop_ilm
----------

(1 row)
```

2.12.2.7 DBE_ILM_ADMIN

接口介绍

服务于ILM策略实施，实现ADO的后台调度以及各个限流参数的控制。
DBE_ILM_ADMIN支持的接口请参见**表2-183**。

表 2-183 DBE_ILM_ADMIN

接口名称	描述
DBE_ILM_ADMIN.CUSTOMIZE_ILM	根据输入参数定制ILM策略属性。
DBE_ILM_ADMIN.DISABLE_ILM	关闭后台调度。
DBE_ILM_ADMIN.ENABLE_ILM	开启后台调度。
DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY	创建库级压缩策略。
DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY	删除库级压缩策略。

说明

当并发量较大时，执行DBE_ILM_ADMIN.DISABLE_ILM或DBE_ILM_ADMIN.ENABLE_ILM可能会提示资源繁忙，稍后重试即可。提示内容为“Resources are busy, please try again later.”。

- DBE_ILM_ADMIN.CUSTOMIZE_ILM

  根据输入参数定制ILM策略属性。

  DBE_ILM_ADMIN.CUSTOMIZE_ILM原型为：

  ```
  DBE_ILM_ADMIN.CUSTOMIZE_ILM(
  PARAM    IN    NUMBER,
  VAL      IN    NUMBER);
  ```


版权所有 © 华为云计算技术有限公司

表 2-184 DBE_ILM_ADMIN.CUSTOMIZE_ILM 接口参数说明

参数	描述
PARAM	参数序号。
VAL	参数取值。

说明

由于兼容性影响，在MYSQL或M-Compatibility兼容模式下，若调用DBE_ILM_ADMIN.CUSTOMIZE_ILM()时传入的VAL值为非数字的字符，例如DBE_ILM_ADMIN.CUSTOMIZE_ILM(13, '*')，会默认将传入的VAL值赋值为0。

表 2-185 DBE_ILM_ADMIN.CUSTOMIZE_ILM 接口参数范围

参数序号	参数值	描述
1	EXECUTION_INTERVAL	ADO Task的执行频率，单位分钟，默认值15。取值范围为大于等于1小于等于2147483647的整数或浮点数，作用时向下取整。
2	RETENTION_TIME	ADO相关历史的保留时长，单位天，默认值30。取值范围为大于等于1小于等于2147483647的整数或浮点数，作用时向下取整。
7	ENABLE	后台调度的状态。该参数不支持使用DBE_ILM_ADMIN.CUSTOMIZE_ILM修改，应使用DBE_ILM_ADMIN.DISABLE_ILM和DBE_ILM_ADMIN.ENABLE_ILM修改。
11	POLICY_TIME	控制ADO的条件单位是天或者秒，秒仅用来做测试用。取值为： ● 0：ILM_POLICY_IN_DAYS（默认值）。 ● 1：ILM_POLICY_IN_SECONDS。
12	ABS_JOBLIMIT	控制一次ADO Task最多生成ADO Job的数量。取值范围为大于等于0小于等于2147483647的整数或浮点数，作用时向下取整。
13	JOB_SIZELIMIT	控制单个ADO Job可以处理的最大字节数，单位兆。取值范围为大于等于0小于等于2147483647的整数或浮点数，作用时向下取整。当JOB_SIZELIMIT为0时，手动执行压缩任务会报warning且不做压缩处理，自动调度后台任务也不做压缩处理。
14	WIND_DURATION	维护窗口持续时长，单位分钟，默认值240（4小时）。取值范围为大于等于0小于1440（24小时）的整数。

版权所有 © 华为云计算技术有限公司

参数序号	参数值	描述
15	BLOCK_LIMITS	控制实例级的行存压缩速率上限，默认值40，单位是block/ms（表示每毫秒最多压缩多少个block）。取值范围是0到10000（0表示不限制）。跨页透明压缩不支持此参数。
16	ENABLE_META_COMPRESSION	是否开启header压缩，默认值0。取值范围为0（关闭）和1（开启）。跨页透明压缩不支持此参数。 **说明** 设置此参数为1时，对于单行数据较短的表，压缩率会有一定提升，但是访问压缩行的性能会有较大幅度的下降。若数据库多是单行数据较长的表，不建议开启此参数。
17	SAMPLE_MIN	常量编码和等值编码采样步长最小值，默认值10。取值范围[1, 100]，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
18	SAMPLE_MAX	常量编码和等值编码采样步长最大值，默认值10。取值范围[1, 100]，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
19	CONST_PRIO	常量编码优先级，默认值40。取值范围[0, 100]，100表示关闭常量编码，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
20	CONST_THRESHOLD	常量编码阈值，默认值90。取值范围[1, 100]，表示一列常量值的占比超过该阈值时进行常量编码，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
21	EQVALUE_PRIO	等值编码优先级，默认值60。取值范围[0, 100]，100表示关闭等值编码，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
22	EQVALUE_THRESHOLD	等值编码阈值，默认值80。取值范围[1, 100]，表示两列数据的等值比例超过该阈值时进行等值编码，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
23	ENABLE_DELTA_ENCODE_SWITCH	差值编码开关，默认值1。取值范围[0, 1]，0表示关闭，1表示开启，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
24	LZ4_COMPRESSION_LEVEL	lz4压缩等级，默认值0。取值范围[0, 16]，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。
25	ENABLE_LZ4_PARTIAL_DECOMPRESSION	部分解压开关，默认值1。取值范围[0, 1]，0表示关闭，1表示开启，支持小数输入，小数会自动向下取整。跨页透明压缩不支持此参数。

版权所有 © 华为云计算技术有限公司

参数序号	参数值	描述
26	INDEXJOB_SIZELIMIT	控制索引压缩时单个ADO Job可以处理的最大字节数，单位兆，默认为1024。取值范围为大于等于0小于等于2147483647的整数或浮点数，作用时向下取整。当INDEXJOB_SIZELIMIT为0时，手动执行压缩任务会报warning且不做压缩处理，自动调度后台任务也不做压缩处理。
27	ENABLE_INDEX_COMPRESS	索引压缩使能开关，0表示关闭，1表示打开，默认为0。

示例：

```
gaussdb=# CALL DBE_ILM_ADMIN.CUSTOMIZE_ILM(1, 15);
 customize_ilm
---------------


(1 row)
gaussdb=# SELECT * FROM gs_adm_ilmparameters;
               name               | value
----------------------------------+-------
 EXECUTION_INTERVAL               |    15
 RETENTION_TIME                   |    30
 ENABLED                          |     1
 POLICY_TIME                      |     0
 ABS_JOBLIMIT                     |    10
 JOB_SIZELIMIT                    |  1024
 WIND_DURATION                    |   240
 BLOCK_LIMITS                     |    40
 ENABLE_META_COMPRESSION          |     0
 SAMPLE_MIN                       |    10
 SAMPLE_MAX                       |    10
 CONST_PRIO                       |    40
 CONST_THRESHOLD                  |    90
 EQVALUE_PRIO                     |    60
 EQVALUE_THRESHOLD                |    80
 ENABLE_DELTA_ENCODE_SWITCH       |     1
 LZ4_COMPRESSION_LEVEL            |     0
 ENABLE_LZ4_PARTIAL_DECOMPRESSION |     1
 INDEXJOB_SIZELIMIT               |  1024
 ENABLE_INDEX_COMPRESS            |     0
(20 rows)
```

- DBE_ILM_ADMIN.DISABLE_ILM

  关闭后台调度。

  DBE_ILM_ADMIN.DISABLE_ILM原型为：

  ```
  CALL DBE_ILM_ADMIN.DISABLE_ILM();
  ```

- DBE_ILM_ADMIN.ENABLE_ILM

  开启后台调度。

  DBE_ILM_ADMIN.ENABLE_ILM原型为：

  ```
  CALL DBE_ILM_ADMIN.ENABLE_ILM();
  ```

  说明

  后台调度生效需要先在数据库运维平台将GUC参数enable_ilm设置为on。

- DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY

版权所有 © 华为云计算技术有限公司

基于输入参数创建库级压缩策略，该策略将执行以下操作：

- 自动覆盖存量表：为当前数据库中所有未绑定压缩策略的用户表附加此策略。
- 新表默认继承：后续新建表若未显式声明压缩策略，将自动继承该库级策略定义。

DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY原型为：

```
DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY(
DAYS IN INT,
COMPRESSION_LEVEL IN TEXT DEFAULT 'MEDIUM');
```

表 2-186 DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY 接口参数说明

参数	描述
DAYS	压缩策略天数。
COMPRESSION_LEVEL	压缩策略级别。仅支持'HIGH' 和 'MEDIUM'，默认值为 'MEDIUM'。

- DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY

  删除库级压缩策略及所有表上继承于库级压缩策略的压缩策略。

说明

- 库级压缩策略仅支持高级压缩。
- 库级策略不支持表达式功能。
- 在使用gs_dump导出库级策略定义时，需满足以下条件方可生效：必须执行全库导出操作且用户具备系统管理员权限。若上述条件未同时满足，则相关策略将以表、分区或子分区级别的独立策略形式导出，而非作为库级策略处理。
- 多个不同事务内若调用DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY、DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY、DBE_ILM.EXECUTE_ILM等接口并发执行，先执行的事务若一直不提交，后执行的事务可能存在锁超时风险。

库级策略示例：

- 创建一个策略时间为10天压缩策略:

```
gaussdb=# CALL DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY(10);
 create_ilm_db_policy
----------------------


(1 row)
```

- 删除库级策略

```
gaussdb=# CALL DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY();
 delete_ilm_db_policy
----------------------


(1 row)
```

- 创建一个策略时间为10天的high级别压缩策略:

```
gaussdb=# CALL DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY(10, 'high');
 create_ilm_db_policy
----------------------


(1 row)
```


版权所有 © 华为云计算技术有限公司

2.12.2.8 DBE_LICENSE

说明

只有具有系统管理员或运维管理员权限的用户可以调用该高级包下的接口。

接口介绍

高级功能包DBE_LICENSE主要用于激活、查看、注销License。主要支持的接口参见表DBE_LICENSE。

表 2-187 DBE_LICENSE

接口名称	描述
DBE_LICENSE.ACTIVATE	激活License。
DBE_LICENSE.DEACTIVATE	注销当前License。
DBE_LICENSE.GET_ESN	查询节点ESN信息。
DBE_LICENSE.GET_LICENSE_INFO	查询当前节点License状态信息。
DBE_LICENSE.GET_ALL_NODE_LICENSE_ACTIVATE	查询数据库集群的License激活状态。

- DBE_LICENSE.ACTIVATE

  函数ACTIVATE根据License类型和对应文件激活License。

  当前支持令牌文件或ESDP License文件两种License进行激活。

  DBE_LICENSE.ACTIVATE原型为：

```
DBE_LICENSE.ACTIVATE(
    license_type IN VARCHAR2,
    token_file IN VARCHAR2)
RETURN BOOLEAN;

DBE_LICENSE.ACTIVATE(
    license_type IN VARCHAR2,
    license_file IN VARCHAR2,
    key_file IN VARCHAR2)
RETURN BOOLEAN;
```

  表 2-188 DBE_LICENSE.ACTIVATE 接口参数说明

参数	描述
license_type	待激活License类型，可选值为OnCloud或OffCloud。 • OnCloud：表示使用令牌文件激活License。 • OffCloud：表示使用ESDP License文件激活License。
token_file	令牌文件名称。需遵循以下要求： • 名称不能为license.json、license.xml； • 名称中不能包含'/'或'\\'字符。

版权所有 © 华为云计算技术有限公司

参数	描述
license_file	ESDP格式License文件名称。
key_file	公钥文件名称。

- DBE_LICENSE.DEACTIVATE

  函数DEACTIVATE用于注销当前节点的License。

  当传入参数为false时，函数在License处于已注销状态下会抛出错误；当传入参数为true时，函数在此状态下将返回true。

  参数未传入时，默认值为false。

  须知

  - 执行注销操作后，原先的License将失效，系统将重新为当前数据库节点生成新的ESN信息，且License处于未激活状态。为了保护您的权益不受影响，在执行该操作前，请务必谨慎考虑。
  - 使用ESDP License激活License时，注销节点的License会在数据目录下的gs_esn文件夹中生成撤销码文件RevoTicket.txt。您可用该文件去ESDP平台刷新License的ESN值用于再次激活节点License。

  DBE_LICENSE.DEACTIVATE原型为：

```
DBE_LICENSE.DEACTIVATE(
    idempotent    IN BOOLEAN DEFAULT FALSE)
RETURN BOOLEAN;
```

- DBE_LICENSE.GET_ESN

  函数GET_ESN查询当前数据库节点的ESN信息，ESN是数据库节点的唯一标识符，数据库内部会尽可能保证每个节点ESN的唯一性。

  DBE_LICENSE.GET_ESN原型为：

```
DBE_LICENSE.GET_ESN()
RETURN VARCHAR2;
```

- DBE_LICENSE.GET_LICENSE_INFO

  函数GET_LICENSE_INFO查询节点license信息。

  DBE_LICENSE.GET_LICENSE_INFO原型为：

```
DBE_LICENSE.GET_LICENSE_INFO()
```

  返回值类型：setof record

  示例：

```
gaussdb=# SELECT * FROM  dbe_license.get_license_info();
      create_time       |     activate_time      |      expire_time       |  type   | license_version | customer |
state   |       lsn       |                                  feature_list
                                | dn_capacity | cn_capacity
------------------------+------------------------+------------------------+---------+-----------------+----------
+-----------+-----------------+-----------------------------------------------------------
-----------------------------------+-------------+-------------
 2024-10-20 14:30:45+08 | 2026-01-22 12:02:17+08 | 2026-01-22 12:03:45+08 | OnCloud |               1 |
Test     | Activated | lsntesttesttest | enable_ilm,enable_tde,enable_ledger,enable
_security_policy |           1 |           1
(1 row)
```


版权所有 © 华为云计算技术有限公司

表 2-189 DBE_LICENSE.GET_LICENSE_INFO 接口返回值字段说明

名称	类型	描述
create_time	timestamptz(0)	License创建时间。
activate_time	timestamptz(0)	License激活时间。 ● 已激活的正式License，激活时间为调用接口激活的时间。 ● 试用的License，激活时间为进入试用模式的时间。
expire_time	timestamptz(0)	License到期时间。 ● 已激活的License，到期时间为令牌文件或ESDP License文件到期时间。 ● 试用的License，到期时间为进入试用模式时的第 90 天。
type	text	License的部署类型： ● 带管控部署：OnCloud。 ● 无管控部署：OffCloud。 ● 其他：Illegal deploy mode
license_version	bigint	License的版本。 ● 带管控部署时，用于区分签名算法及License结构等。 ● 无管控部署时为0。
customer	text	License的购买方。
state	text	License的状态： ● Activated：表示License已激活。 ● Deactivated：表示License已注销。 ● Expired：表示License已过期。 ● Trial：表示License为试用状态。 ● Error：表示License异常。
lsn	text	License的识别编号。
feature_list	text	License包含的高阶特性列表，由逗号分割的特性GUC参数名称组成。如“ enable_tde,enable_ledger”。
dn_capacity	bigint	预留字段，无实际含义。
cn_capacity	bigint	预留字段，无实际含义。

- DBE_LICENSE.GET_ALL_NODE_LICENSE_ACTIVATE

版权所有 © 华为云计算技术有限公司

函数GET_ALL_NODE_LICENSE_ACTIVATE查询当前数据库集群的License激活状态。当集群所有CN和主DN节点License均处于已激活或试用状态时，返回true；当任意CN或主DN节点的状态不为已激活或试用状态时，返回false。

DBE_LICENSE.GET_ALL_NODE_LICENSE_ACTIVATE原型为：

```
DBE_LICENSE.GET_ALL_NODE_LICENSE_ACTIVATE()
RETURN BOOLEAN;
```

2.12.2.9 DBE_LOB

接口介绍

高级功能包DBE_LOB支持的接口参见表2-190。

说明

- ORA数据库中空格的实际字节内容为00，GaussDB中空格对应字节内容为ASCII码值（32）。
- 分布式环境中，CLOB类型对象（下文简称CLOB）、BLOB类型对象（下文简称BLOB）以及BFILE文件（下文简称BFILE）最大支持1GB。
- LOBMAXSIZE最大支持1073741771字节。
- DBE_LOB包中的LOB修改接口（如LOB_WRITE、LOB_COPY、CONVERTTOBLOB、LOB_CONVERTTOBLOB、LOADBLOBFROMBFILE等）仅更新内存中的LOB变量，不会自动更新其关联的物理表数据。即使目标行已通过SELECT ... FOR UPDATE锁定，LOB数据变更也不会自动持久化到数据库表中。如需保存修改结果，必须显式执行UPDATE语句。

表 2-190 DBE_LOB

接口名称	描述
DBE_LOB.GET_LENGTH	获取并返回指定的LOB的长度（不支持大于2GB）。
DBE_LOB.LOB_GET_LENGTH	获取并返回指定的LOB/BFILE的长度。
DBE_LOB.OPEN	打开一个LOB，并返回一个LOB的描述符。
DBE_LOB.READ	根据指定的长度及起始位置偏移读取LOB内容的一部分到BUFFER缓冲区。
DBE_LOB.LOB_READ	根据指定的长度及起始位置偏移读取LOB内容的一部分到BUFFER缓冲区（支持BFILE读取）。
DBE_LOB.WRITE	根据指定长度及起始位置偏移将BUFFER中内容写入到LOB中。
DBE_LOB.WRITE_APPEND	根据指定长度将BUFFER中内容写入到LOB的尾部。
DBE_LOB.LOB_WRITE_APPEND	根据指定长度将BUFFER中内容写入到LOB的尾部。
DBE_LOB.COPY	根据指定长度及起始位置偏移将LOB内容写入到另一个LOB中。
DBE_LOB.LOB_COPY	根据指定长度及起始位置偏移将LOB内容写入到另一个LOB中。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_LOB.ERASE	根据指定长度及起始位置偏移删除LOB中的内容（不支持大于1GB）。
DBE_LOB.LOB_ERASE	根据指定长度及起始位置偏移删除LOB中的内容。
DBE_LOB.CLOSE	关闭已经打开的LOB描述符。
DBE_LOB.MATCH	返回一个字符串在LOB中第N次出现的位置。
DBE_LOB.COMPARE	比较两个LOB或者两个LOB的某一部分（支持BFILE比较）。
DBE_LOB.SUBSTR	用于读取一个LOB的子串，返回读取到的子串。
DBE_LOB.LOB_SUBSTR	用于读取一个LOB或者BFILE的子串，返回读取到的子串。
DBE_LOB.STRIP	用于截断指定长度的LOB，执行完会将LOB的长度设置为参数指定的长度。
DBE_LOB.LOB_STRIP	用于截断指定长度的LOB，执行完会将LOB的长度设置为参数指定的长度。
DBE_LOB.CREATE_TEMPORARY	创建一个临时的BLOB或者CLOB。
DBE_LOB.APPEND	将源LOB的内容拼接到目的LOB中。
DBE_LOB.LOB_APPEND	将源LOB的内容拼接到目的LOB中。
DBE_LOB.FREETEMPORARY	删除一个临时的BLOB或者CLOB。
DBE_LOB.FILEOPEN	打开一个数据库BFILE，并返回文件描述符。
DBE_LOB.FILECLOSE	关闭由FILEOPEN打开的BFILE。
DBE_LOB.BFILEOPEN	打开一个数据库BFILE。
DBE_LOB.BFILECLOSE	关闭一个由BFILEOPEN打开的BFILE。
DBE_LOB.LOADFROMFILE	读取指定位置和长度的数据库BFILE到指定位置的BLOB中。
DBE_LOB.LOADFROMBFILE	读取指定位置和长度的数据库BFILE到指定位置的LOB中。
DBE_LOB.LOADBLOBFROMFILE	读取指定位置和长度的数据库外部文件到指定位置的BLOB中（不支持大于1GB）。
DBE_LOB.LOADBLOBFROMBFILE	读取指定位置和长度的数据库BFILE到指定位置的BLOB中。
DBE_LOB.LOADCLOBFROMFILE	读取指定位置和长度的数据库外部文件到指定位置的CLOB中（不支持大于1GB）。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_LOB.LOADCLOBFROMBFILE	读取指定位置和长度的数据库BFILE到指定位置的CLOB中。
DBE_LOB.CONVERTTOBLOB	将CLOB转换为BLOB（不支持大于1GB）。
DBE_LOB.CONVERTTOCLOB	将BLOB转换为CLOB（不支持大于1GB）。
DBE_LOB.LOB_CONVERTTOBLOB	将CLOB转换为BLOB。
DBE_LOB.LOB_CONVERTTOCLOB	将BLOB转换为CLOB。
DBE_LOB.GETCHUNKSIZE	获取数据库中CHUNK结构中用于存储LOB数据的最大SIZE。
DBE_LOB.LOB_WRITE	将源对象从起始位置读取指定长度内容，写入目标LOB的指定偏移位置，覆盖该位置原内容, 并返回目标LOB。
DBE_LOB.BFILENAME	根据目录和文件名构造返回DBE_LOB.BFILE。

- DBE_LOB.GET_LENGTH

  函数GET_LENGTH获取并返回指定的LOB类型对象的长度，最大支持2GB。

  DBE_LOB.GET_LENGTH原型为：

```
DBE_LOB.GET_LENGTH (
    blob_obj IN BLOB)
RETURN INTEGER;

DBE_LOB.GET_LENGTH (
    clob_obj IN CLOB)
RETURN INTEGER;
```

  表 2-191 DBE_LOB.GET_LENGTH 接口参数说明

参数	描述
blob_obj/ clob_obj	待获取长度的BLOB类型对象/CLOB类型对象。

- DBE_LOB.LOB_GET_LENGTH

  函数LOB_GET_LENGTH获取并返回指定的LOB类型对象/BFILE文件的长度，最大支持32TB。

  DBE_LOB.LOB_GET_LENGTH原型为：

```
DBE_LOB.LOB_GET_LENGTH (
    blob_obj IN BLOB)
RETURN BIGINT;

DBE_LOB.LOB_GET_LENGTH (
    clob_obj IN CLOB)
RETURN BIGINT;
```


版权所有 © 华为云计算技术有限公司

```
DBE_LOB.LOB_GET_LENGTH (
   bfile IN DBE_LOB.BFILE)
RETURN BIGINT;
```

表 2-192 DBE_LOB.LOB_GET_LENGTH 接口参数说明

参数	描述
blob_obj/ clob_obj/bfile	待获取长度的BLOB类型对象/CLOB类型对象/BFILE文件。

- DBE_LOB.OPEN

  存储过程打开一个LOB类型对象，并返回一个LOB描述符，该过程无实际意义，仅用于兼容。

  DBE_LOB.OPEN原型为：

```
DBE_LOB.OPEN (
   lob INOUT BLOB);

DBE_LOB.OPEN (
   lob INOUT CLOB);

DBE_LOB.OPEN (
   bfile     INOUT DBE_LOB.BFILE,
   open_mode IN    TEXT DEFAULT 'null');
```

表 2-193 DBE_LOB.OPEN 接口参数说明

参数	描述
lob/bfile	被打开的BLOB类型对象/CLOB类型对象/BFILE文件。
open_mode	操作模式，现支持[R,W,A,RB,WB,AB]。

- DBE_LOB.READ

  存储过程READ根据指定长度及起始位置偏移读取LOB类型对象的一部分到out_put缓冲区。

  DBE_LOB.READ原型为：

```
DBE_LOB.READ (
   blob_obj IN  BLOB,
   amount   IN  INTEGER,
   off_set  IN  INTEGER,
   out_put  OUT RAW);

DBE_LOB.READ (
   clob_obj IN  CLOB,
   amount   IN  INTEGER,
   off_set  IN  INTEGER,
   out_put  OUT VARCHAR2);
```

表 2-194 DBE_LOB.READ 接口参数说明

参数	说明
blob_obj/ clob_obj	待读入的BLOB/CLOB类型对象。

版权所有 © 华为云计算技术有限公司

参数	说明
amount	读入长度。 **说明** 如果读入长度小于1，或大于32767，则报错。
off_set	指定从参数LOB类型对象开始读取偏移（即相对LOB类型对象起始位置的字节数）的位置。如果偏移量小于1或者大于LOB类型对象长度，则报错。初始位置为1。
out_put	读取参数LOB类型对象后存放的目标缓冲区。

- DBE_LOB.LOB_READ

  存储过程LOB_READ根据指定长度及起始位置偏移读取LOB类型对象/BFILE文件的一部分到out_put缓冲区。

  DBE_LOB.LOB_READ原型为：

```
DBE_LOB.LOB_READ(
   blob_obj IN    BLOB,
   amount   INOUT BIGINT,
   off_set  IN    BIGINT,
   out_put  OUT   RAW);

DBE_LOB.LOB_READ(
   clob_obj IN    CLOB,
   amount   INOUT BIGINT,
   off_set  IN    BIGINT,
   out_put  OUT   VARCHAR2);

DBE_LOB.LOB_READ(
   bfile   IN    DBE_LOB.BFILE,
   amount  INOUT BIGINT,
   off_set IN    BIGINT,
   out_put OUT   RAW);
```

  表 2-195 DBE_LOB.LOB_READ 接口参数说明

参数	说明
blob_obj/ clob_obj/bfile	待读入的BLOB类型对象/CLOB类型对象/BFILE文件（支持大于1GB）。
amount	IN参数为读入长度，OUT参数为实际读取的长度。 **说明** 如果读入长度小于1，或大于32767，则报错。
off_set	指定从参数LOB类型对象/BFILE文件开始读取偏移（即相对LOB内容起始位置的字节数）的位置。如果偏移量小于1或者大于LOB类型对象/BFILE文件长度，则报错。初始位置为1。
out_put	读取参数LOB类型对象/BFILE文件后存放的目标缓冲区。

- DBE_LOB.WRITE

  存储过程WRITE根据指定长度及起始位置将source中内容写入到LOB类型对象中。

  DBE_LOB.WRITE原型为：

```
DBE_LOB.WRITE (
   blob_obj INOUT BLOB,
```


版权所有 © 华为云计算技术有限公司

```
  amount   IN    INTEGER,
  off_set  IN    INTEGER,
  source   IN    RAW);

DBE_LOB.WRITE (
  clob_obj INOUT CLOB,
  amount   IN    INTEGER,
  off_set  IN    INTEGER,
  source   IN    VARCHAR2);
```

表 2-196 DBE_LOB.WRITE 接口参数说明

参数	说明
blob_obj/ clob_obj	待写入的BLOB类型对象/CLOB类型对象。
amount	写入长度，最大支持32767字节。 **说明** 如果写入长度小于1或写入长度大于待写入的内容长度，则报错。
off_set	指定从BLOB类型对象/CLOB类型对象开始写入偏移（即相对LOB内容起始位置的字节数）的位置。 **说明** 如果偏移量小于1或者大于LOBMAXSIZE时，则报错。初始位置是1，最大值为LOB类型对象最大长度。
source	待写入的内容。

- DBE_LOB.WRITE_APPEND

  存储过程WRITE_APPEND根据指定长度将source_obj中内容写入到LOB类型对象的尾部。

  DBE_LOB.WRITE_APPEND原型为：

```
DBE_LOB.WRITE_APPEND (
  blob_obj   INOUT BLOB,
  amount     IN    INTEGER,
  source_obj IN    RAW);

DBE_LOB.WRITE_APPEND (
  clob_obj   INOUT CLOB,
  amount     IN    INTEGER,
  source_obj IN    VARCHAR2);
```

表 2-197 DBE_LOB.WRITE_APPEND 接口参数说明

参数	说明
blob_obj/ clob_obj	待写入的指定BLOB类型对象/CLOB类型对象。
amount	写入长度，最大支持32767字节。 **说明** 如果写入长度小于1或写入长度大于待写入的内容长度，则报错。
source_obj	待写入的内容。

- DBE_LOB.LOB_WRITE_APPEND

版权所有 © 华为云计算技术有限公司

存储过程LOB_WRITE_APPEND根据指定长度将source_obj中内容写入到LOB类型对象的尾部。

DBE_LOB.LOB_WRITE_APPEND原型为：

```
DBE_LOB.LOB_WRITE_APPEND(
   blob_obj    INOUT BLOB,
   amount      IN    INTEGER,
   source_obj  IN    RAW);

DBE_LOB.LOB_WRITE_APPEND (
   clob_obj    INOUT CLOB,
   amount      IN    INTEGER,
   source_obj  IN    VARCHAR2);
```

表 2-198 DBE_LOB.LOB_WRITE_APPEND 接口参数说明

参数	说明
blob_obj/ clob_obj	待写入的指定BLOB类型对象/CLOB类型对象。
amount	写入长度，最大支持32767字节。 **说明** 如果写入长度小于1或写入长度大于待写入的内容长度，则报错。
source_obj	待写入的内容。

- DBE_LOB.COPY

  存储过程COPY根据指定长度及起始位置偏移将LOB类型对象内容复制到另一个LOB类型对象中。

  DBE_LOB.COPY原型为：

  ```
  DBE_LOB.COPY (
     dest_lob   INOUT BLOB,
     src_lob    IN    BLOB,
     len        IN    INTEGER,
     dest_start IN    INTEGER DEFAULT 1,
     src_start  IN    INTEGER DEFAULT 1);
  ```

  表 2-199 DBE_LOB.COPY 接口参数说明

参数	说明
dest_lob	待拷入的LOB类型对象。
src_lob	待拷出的LOB类型对象。
len	复制长度。
dest_start	指定从dest_lob内容开始拷入偏移（即相对LOB内容起始位置的字节数）的位置。
src_start	指定从src_lob内容开始拷出偏移（即相对LOB内容起始位置的字节数）的位置。

- DBE_LOB.LOB_COPY

  存储过程COPY根据指定长度及起始位置偏移将LOB类型对象内容复制到另一个LOB类型对象中。

版权所有 © 华为云计算技术有限公司

DBE_LOB.LOB_COPY原型为：

```
DBE_LOB.LOB_COPY(
   blob_obj    INOUT BLOB,
   source_obj  IN    BLOB,
   amount      IN    BIGINT,
   dest_offset IN    BIGINT DEFAULT 1,
   src_offset  IN    BIGINT DEFAULT 1);

DBE_LOB.LOB_COPY(
   clob_obj    INOUT CLOB,
   source_obj  IN    CLOB,
   amount      IN    BIGINT,
   dest_offset IN    BIGINT DEFAULT 1,
   src_offset  IN    BIGINT DEFAULT 1);
```

表 2-200 DBE_LOB.LOB_COPY 接口参数说明

参数	说明
blob_obj/ clob_obj	待拷入的LOB类型对象。
source_obj	待拷出的LOB类型对象。
amount	复制长度。 **说明** 如果拷入长度小于1或拷入长度大于LOBMAXSIZE，则报错。
dest_offset	指定从blob_obj/clob_obj内容开始拷入偏移（即相对LOB类型对象起始位置的字节数/字符数，BLOB对象以字节为单位，CLOB对象以字符为单位）的位置。 **说明** 如果偏移量小于1或者大于LOBMAXSIZE，则报错。
src_offset	指定从source_obj内容开始拷出偏移（即相对LOB类型对象起始位置的字节数/字符数，BLOB对象以字节为单位，CLOB对象以字符为单位）的位置。 **说明** 如果偏移量小于1则报错。

- DBE_LOB.ERASE

  存储过程ERASE根据指定长度及起始位置偏移删除blob_obj中的内容（不支持大于1GB），blob_obj中删除部分的字节填充为0。

  DBE_LOB.ERASE原型为：

  ```
  DBE_LOB.ERASE (
     blob_obj INOUT BLOB,
     amount   INOUT INTEGER,
     off_set  IN    INTEGER DEFAULT 1);
  ```

  表 2-201 DBE_LOB.ERASE 接口参数说明

参数	说明
blob_obj	IN参数为待删除内容的LOB类型对象，OUT参数为删除指定部分后的LOB类型对象，传空报错。

版权所有 © 华为云计算技术有限公司

参数	说明
amount	IN参数为待删除的长度（BLOB类型对象以字节为单位），OUT参数为实际删除的长度。 **说明** 如果删除长度小于1或传空，则报错。
off_set	指定从LOB类型对象开始删除偏移（即相对BLOB内容起始位置的字节数，不支持大于1GB）的位置。 **说明** 如果偏移量小于1或偏移量传空，则报错。

- DBE_LOB.LOB_ERASE

  存储过程LOB_ERASE根据指定长度及起始位置偏移删除LOB类型对象中的内容，BLOB类型对象中删除部分的字节填充为0，CLOB类型对象中删除部分的字符填充为空格，支持LOB类型对象大于1GB，最大支持32TB。

  DBE_LOB.LOB_ERASE原型为：

```
DBE_LOB.LOB_ERASE (
   blob_obj INOUT BLOB,
   amount   INOUT BIGINT,
   off_set  IN    BIGINT DEFAULT 1);

DBE_LOB.LOB_ERASE (
   clob_obj INOUT CLOB,
   amount   INOUT BIGINT,
   off_set  IN    BIGINT DEFAULT 1);
```

  表 2-202 DBE_LOB.LOB_ERASE 接口参数说明

参数	说明
blob_obj/ clob_obj	IN参数为待删除内容的LOB类型对象，OUT参数为删除指定部分后的LOB类型对象，传空报错。
amount	IN参数为待删除的长度（BLOB类型对象以字节为单位，CLOB类型对象以字符为单位），OUT参数为实际删除的长度。 **说明** 如果删除长度小于1或传空，则报错。
off_set	指定从LOB类型对象开始删除偏移（即相对BLOB内容起始位置的字节数/相对CLOB内容起始位置的字符数）的位置。 **说明** 如果偏移量小于1或偏移量传空，则报错。

- DBE_LOB.CLOSE

  存储过程CLOSE关闭已经打开的LOB类型对象描述符。

  DBE_LOB.CLOSE原型为：

```
DBE_LOB.CLOSE(
   lob IN BLOB);

DBE_LOB.CLOSE (
   lob IN CLOB);
```


版权所有 © 华为云计算技术有限公司

```
DBE_LOB.CLOSE (
   file IN INTEGER);
```

表 2-203 DBE_LOB.CLOSE 接口参数说明

参数	说明
lob/file	待关闭的BLOB类型对象/CLOB类型对象/BFILE文件类型对象。

- DBE_LOB.MATCH

  该函数返回字符串在LOB类型对象或者BFILE文件中第N次出现的位置，如果输入的是一些无效值会返回NULL值。支持LOB类型对象或者BFILE文件大于1GB，最大支持32TB。

  DBE_LOB.MATCH原型为：

```
DBE_LOB.MATCH(
   blob_obj    IN BLOB,
   blob_obj2   IN RAW,
   beg_index   IN BIGINT DEFAULT 1,
   occur_index IN BIGINT DEFAULT 1)
RETURN BIGINT;

DBE_LOB.MATCH(
   clob_obj    IN CLOB,
   clob_obj2   IN VARCHAR2,
   beg_index   IN BIGINT DEFAULT 1,
   occur_index IN BIGINT DEFAULT 1)
RETURN BIGINT;

DBE_LOB.MATCH(
   bfile       IN DBE_LOB.BFILE,
   blob_obj2     IN RAW,
   beg_index   IN BIGINT DEFAULT 1,
   occur_index IN BIGINT DEFAULT 1)
RETURN BIGINT;
```

表 2-204 DBE_LOB.MATCH 接口参数说明

参数	说明
blob_obj/ clob_obj/ bfile	要查找的BLOB类型对象/CLOB类型对象描述符，或者BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开），传空返回NULL。
blob_obj2/ clob_obj2	要匹配的模式，对于BLOB类型对象/BFILE文件是由一组RAW类型的数据组成，对于CLOB类型对象是由一组VARCHAR2类型的数据组成，传空返回NULL。
beg_index	对于BLOB类型对象/BFILE文件是以字节为单位的绝对偏移量，对于CLOB类型对象是以字符为单位的偏移量，模式匹配的起始位置是1。 **说明** 有效范围为1~LOBMAXSIZE，超过返回NULL。

版权所有 © 华为云计算技术有限公司

参数	说明
occur_index	模式匹配的次数，最小值为1。 **说明** 若大于模式串在LOB类型对象中最大能匹配上的次数，则返回0，若不在范围1~LOBMAXSIZE，则返回NULL。

- DBE_LOB.COMPARE

  该函数比较部分或者全部LOB类型对象或BFILE文件。

  - 如果比较的结果相等返回0，否则返回非零的值。
  - 如果第一个LOB类型对象比第二个小，返回-1；如果第一个LOB类型对象比第二个大，返回1。
  - 如果len、start1、start2这几个参数有无效参数返回NULL，有效偏移量范围是1~LOBMAXSIZE。
  - 如果start_pos1、start_pos2同时超过LOB类型对象/BFILE文件长度，则返回0。

  DBE_LOB.COMPARE原型为：

```
DBE_LOB.COMPARE (
   lob1       IN BLOB,
   lob2       IN BLOB,
   len        IN BIGINT DEFAULT 1073741312,
   start_pos1 IN BIGINT DEFAULT 1,
   start_pos2 IN BIGINT DEFAULT 1)
RETURN INTEGER;

DBE_LOB.COMPARE (
   lob1       IN CLOB,
   lob2       IN CLOB,
   len        IN BIGINT DEFAULT 1073741312,
   start_pos1 IN BIGINT DEFAULT 1,
   start_pos2 IN BIGINT DEFAULT 1)
RETURN INTEGER;

DBE_LOB.COMPARE (
   file1      IN DBE_LOB.BFILE,
   file2      IN DBE_LOB.BFILE,
   len        IN BIGINT DEFAULT 1073741312,
   start_pos1 IN BIGINT DEFAULT 1,
   start_pos2 IN BIGINT DEFAULT 1)
RETURN INTEGER;
```

表 2-205 DBE_LOB.COMPARE 接口参数说明

参数	说明
lob1/file1	第一个要比较的BLOB类型对象/CLOB类型对象/BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开）。
lob2/file2	第二个要比较的BLOB类型对象/CLOB类型对象/BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开）。
len	要比较的字符数或者字节数，默认值为1073741312。
start_pos1	第一个LOB类型对象描述符偏移量，初始位置是1，最大值为LOB类型对象最大长度。

版权所有 © 华为云计算技术有限公司

参数	说明
start_pos2	第二个LOB类型对象描述符的偏移量，初始位置是1，最大值为LOB类型对象最大长度。

- DBE_LOB.SUBSTR

  该函数用于读取一个LOB类型对象的子串，返回读取的子串。

  DBE_LOB.SUBSTR原型为：

```
DBE_LOB.SUBSTR(
   lob_loc IN BLOB,
   amount  IN INTEGER DEFAULT 32767,
   off_set IN INTEGER  DEFAULT 1)
RETURN RAW;

DBE_LOB.SUBSTR(
   lob_loc IN CLOB,
   amount  IN INTEGER DEFAULT 32767,
   off_set IN INTEGER  DEFAULT 1)
RETURN VARCHAR2;
```

  表 2-206 DBE_LOB.SUBSTR 接口参数说明

参数	说明
lob_loc	将要读取子串的LOB类型对象描述符，对于BLOB类型对象的返回值是读取的RAW类型，对于CLOB类型对象的返回值是VARCHAR2类型。
amount	要读取的字节数或者字符数量。 **说明** 范围为1~32767，超出返回NULL。
off_set	从开始位置偏移的字符数或者字节数量。 **说明** 范围为1~LOBMAXSIZE，超出返回NULL。

- DBE_LOB.LOB_SUBSTR

  该函数用于读取一个LOB类型对象或者BFILE文件的子串，返回读取的子串，LOB类型对象或BFILE文件最大支持32TB。

  DBE_LOB.LOB_SUBSTR原型为：

```
DBE_LOB.LOB_SUBSTR(
   lob_loc IN BLOB,
   amount  IN INTEGER DEFAULT 32767,
   off_set IN BIGINT  DEFAULT 1)
RETURN RAW;

DBE_LOB.LOB_SUBSTR(
   lob_loc IN CLOB,
   amount  IN INTEGER DEFAULT 32767,
   off_set IN BIGINT  DEFAULT 1)
RETURN VARCHAR2;

DBE_LOB.LOB_SUBSTR(
   bfile   IN DBE_LOB.BFILE,
   amount  IN INTEGER DEFAULT 32767,
   off_set IN BIGINT  DEFAULT 1)
RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-207 DBE_LOB.LOB_SUBSTR 接口参数说明

参数	说明
lob_loc/ bfile	将要读取子串的LOB类型对象描述符或BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开），对于BLOB类型对象/BFILE文件的返回值是读取的RAW类型，对于CLOB类型对象的返回值是VARCHAR2类型。
amount	要读取的字节数或者字符数量。 **说明** 范围为1~32767，超出返回NULL。
off_set	从开始位置偏移的字符数或者字节数量。 **说明** 范围为1~LOBMAXSIZE，超出返回NULL。

- DBE_LOB.STRIP

  STRIP用于截断指定长度的LOB类型对象，执行完STRIP会将LOB类型对象的长度设置为newlen参数指定的长度。

  DBE_LOB.STRIP原型为：

```
DBE_LOB.STRIP(
   lob_loc INOUT BLOB,
   newlen  IN INTEGER);

DBE_LOB.STRIP(
   lob_loc INOUT CLOB,
   newlen  IN INTEGER);
```

表 2-208 DBE_LOB.STRIP 接口参数说明

参数	说明
lob_loc	IN参数为待读入的指定LOB类型对象，OUT参数为截断后的对象，传空报错。
newlen	截断后LOB类型对象的新长度，对于BLOB类型对象是字节数，对于CLOB类型对象是字符数。

- DBE_LOB.LOB_STRIP

  LOB_STRIP用于截断指定长度的LOB类型对象，执行完LOB_STRIP会将LOB类型对象的长度设置为newlen参数指定的长度。支持LOB类型对象大于1GB，最大支持32TB。

  DBE_LOB.LOB_STRIP原型为：

```
DBE_LOB.LOB_STRIP(
   lob_loc INOUT BLOB,
   newlen  IN    BIGINT);

DBE_LOB.LOB_STRIP(
   lob_loc INOUT CLOB,
   newlen  IN    BIGINT);
```


版权所有 © 华为云计算技术有限公司

表 2-209 DBE_LOB.LOB_STRIP 接口参数说明

参数	说明
lob_loc	IN参数为待读入的指定LOB类型对象，OUT参数为截断后的对象，传空报错。
newlen	截断后LOB类型对象的新长度，对于BLOB类型对象是字节数，对于CLOB类型对象是字符数。 **说明** 小于1返回NULL，大于LOB类型对象的长度，报错。

- DBE_LOB.CREATE_TEMPORARY

  CREATE_TEMPORARY创建一个临时的BLOB类型对象或者CLOB类型对象，CREATE_TEMPORARY仅用于语法上的兼容，并无实际意义。

  DBE_LOB.CREATE_TEMPORARY原型为：

```
DBE_LOB.CREATE_TEMPORARY (
   lob_loc INOUT BLOB,
   cache   IN    BOOLEAN,
   dur     IN    INTEGER DEFAULT 10);

DBE_LOB.CREATE_TEMPORARY (
   lob_loc INOUT CLOB,
   cache   IN    BOOLEAN,
   dur     IN    INTEGER DEFAULT 10);
```

表 2-210 DBE_LOB.CREATE_TEMPORARY 接口参数说明

参数	说明
lob_loc	LOB类型对象描述符。
cache	仅用于语法上的兼容。
dur	仅用于语法上的兼容。

- DBE_LOB.APPEND

  存储过程APPEND将source_obj拼接在目标LOB类型对象之后。

  DBE_LOB.APPEND原型为：

```
DBE_LOB.APPEND (
   blob_obj INOUT BLOB,
   source_obj  IN    BLOB);

DBE_LOB.APPEND (
   clob_obj INOUT CLOB,
   source_obj  IN    CLOB);
```

表 2-211 DBE_LOB.APPEND 接口参数说明

参数	说明
blob_obj/ clob_obj	要写入的BLOB类型对象/CLOB类型对象。
source_obj	读取的BLOB类型对象/CLOB类型对象。

版权所有 © 华为云计算技术有限公司

- DBE_LOB.LOB_APPEND

  存储过程LOB_APPEND将source_obj拼接在目标LOB类型对象之后。

  DBE_LOB.LOB_APPEND原型为：

```
DBE_LOB.LOB_APPEND(
   blob_obj    INOUT BLOB,
   source_obj IN     BLOB);

DBE_LOB.LOB_APPEND(
   clob_obj    INOUT CLOB,
   source_obj IN     CLOB);
```

  表 2-212 DBE_LOB.LOB_APPEND 接口参数说明

参数	说明
blob_obj/ clob_obj	要写入的BLOB类型对象/CLOB类型对象。
source_obj	读取的BLOB类型对象/CLOB类型对象。

- DBE_LOB.FREETEMPORARY

  存储过程FREETEMPORARY用于释放由CREATE_TEMPORARY创建的LOB类型对象。

  DBE_LOB.FREETEMPORARY原型为：

```
DBE_LOB.FREETEMPORARY (
   blob INOUT BLOB);

DBE_LOB.FREETEMPORARY (
   clob INOUT CLOB);
```

  表 2-213 DBE_LOB.FREETEMPORARY 接口参数说明

参数	说明
blob/clob	要释放的BLOB类型对象/CLOB类型对象。

- DBE_LOB.FILEOPEN

  FILEOPEN用于打开数据库外部BFILE类型，并返回该文件对应的文件描述符（fd），一个会话最多支持打开10个BFILE文件。

  BFILE类型定义为：

```
DBE_LOB.BFILE (
   directory TEXT,
   filename  TEXT,
   fd        INTEGER);
```

  DBE_LOB.FILEOPEN原型为：

```
DBE_LOB.FILEOPEN (
   bfile     IN DBE_LOB.BFILE,
   open_mode IN TEXT)
RETURN INTEGER;
```


版权所有 © 华为云计算技术有限公司

表 2-214 DBE_LOB.FILEOPEN 接口参数说明

参数	说明
bfile	要打开的数据库外部文件（BFILE类型包含了文件路径和文件名、文件描述符（fd））。 **说明** 该变量中包含文件目录的位置directory，文件名filename。 • 文件目录的位置，需要添加到系统表**PG_DIRECTORY**中，如果传入的路径和**PG_DIRECTORY**中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包读写safe_data_path指定文件路径下的文件。 • 文件名，包含扩展（文件类型），不包括路径名。如果文件名中包含路径，在OPEN中会被忽略，在Unix系统中，文件名不能以/.结尾。
open_mode	文件打开模式，只支持read模式（r），其他模式报错。

- DBE_LOB.FILECLOSE

  该函数用于关闭数据外部BFILE文件。

  DBE_LOB.FILECLOSE原型为：

```
DBE_LOB.FILECLOSE (
    file IN INTEGER);
```

表 2-215 DBE_LOB.FILECLOSE 接口参数说明

参数	说明
file	要关闭的数据库外部文件（由FILEOPEN返回的文件描述符）。

- DBE_LOB.BFILEOPEN

  BFILEOPEN用于打开数据库外部BFILE文件，一个会话最多支持打开10个BFILE文件。

  DBE_LOB.BFILEOPEN原型为：

```
DBE_LOB.BFILEOPEN (
    bfile     INOUT DBE_LOB.BFILE,
    open_mode IN    TEXT DEFAULT 'R');
```


版权所有 © 华为云计算技术有限公司

表 2-216 DBE_LOB.BFILEOPEN 接口参数说明

参数	说明
bfile	INOUT参数为打开的数据库BFILE文件。 **说明** 该变量中包含文件目录的位置directory，文件名filename。 • 文件目录的位置，需要添加到系统表PG_DIRECTORY中，如果传入的路径和PG_DIRECTORY中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包读写safe_data_path指定文件路径下的文件。 • 文件名，包含扩展（文件类型），不包括路径名。如果文件名中包含路径，在OPEN中会被忽略，在Unix系统中，文件名不能以/.结尾。
open_mode	文件打开模式，只支持read模式（r），其他模式报错。

示例：

```
-- 获取bfile文件的子串（文件内容为ABCD）。
DECLARE
bfile dbe_lob.bfile;
BEGIN
bfile = DBE_LOB.BFILENAME(dir_name, file_name); -- 获取对应bfile文件对象。
DBE_LOB.bfileopen(bfile, 'r'); -- 打开bfile文件。
RAISE NOTICE 'res:%', DBE_LOB.lob_substr(bfile, 10, 1); -- 获取子串，并打印。
DBE_LOB.bfileclose(bfile);-- 关闭bfile文件。
END;
/
NOTICE:  res:41
ANONYMOUS BLOCK EXECUTE
```

- DBE_LOB.BFILECLOSE

  这个存储过程用于关闭数据库外部BFILE文件。

  DBE_LOB.BFILECLOSE原型为：

```
DBE_LOB.BFILECLOSE (
    bfile INOUT DBE_LOB.BFILE);
```

表 2-217 DBE_LOB.BFILECLOSE 接口参数说明

参数	说明
bfile	INOUT参数为关闭的数据库BFILE文件。

- DBE_LOB.LOADFROMFILE

  该函数用于将BFILE外部文件读取到BLOB类型对象中，并以RAW类型返回该对象。

  DBE_LOB.LOADFROMFILE原型为：

```
DBE_LOB.LOADFROMFILE (
    dest_lob     IN BLOB,
    src_file     IN INTEGER,
    amount       IN INTEGER,
    dest_offset IN INTEGER,
    src_offset   IN INTEGER)
RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-218 DBE_LOB.LOADFROMFILE 接口参数说明

参数	说明
dest_lob	目标BLOB类型对象，BFILE文件将读取到这个文件中的指定偏移位置。
src_bfile	需要读取的源BFILE文件。
amount	读取BFILE文件内容的长度。 **说明** 长度小于1或者大于32767则报错。
dest_offset	BLOB类型对象的偏移长度。 **说明** 偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	BFILE文件的偏移长度。 **说明** ● 偏移量小于1或者大于LOBMAXSIZE则报错。 ● amount + src_offset 大于 src_bfile的长度 + 1 则报错。

- DBE_LOB.LOADFROMBFILE

  LOADFROMBFILE用于将BFILE外部文件读取到LOB类型对象中。

  DBE_LOB.LOADFROMBFILE原型为：

```
DBE_LOB.LOADFROMBFILE (
   dest_lob    INOUT BLOB,
   src_file    IN    DBE_LOB.BFILE,
   amount      IN    BIGINT,
   dest_offset IN    BIGINT DEFAULT 1,
   src_offset  IN    BIGINT DEFAULT 1)
RETURN BLOB;

DBE_LOB.LOADFROMBFILE (
   dest_lob    INOUT CLOB,
   src_file    IN    DBE_LOB.BFILE,
   amount      IN    BIGINT,
   dest_offset IN    BIGINT DEFAULT 1,
   src_offset  IN    BIGINT DEFAULT 1)
RETURN CLOB;
```

表 2-219 DBE_LOB.LOADFROMBFILE 接口参数说明

参数	说明
dest_lob	IN参数为目标LOB类型对象，BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开）将读取到这个对象中，OUT参数为完成读取后返回的LOB类型对象，大小支持超过1GB，最大支持32TB。
src_file	需要读取的源BFILE文件，BFILE文件的大小支持超过1GB，最大支持32TB。
amount	读取BFILE文件内容和写入LOB类型对象的长度。 **说明** 长度小于1或者大于LOBMAXSIZE则报错。

版权所有 © 华为云计算技术有限公司

参数	说明
dest_offset	LOB类型对象的偏移长度。 **说明** 偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	BFILE文件的偏移长度。 **说明** ● 偏移量小于1或者大于LOBMAXSIZE则报错。 ● amount + src_offset 大于 src_bfile的长度 + 1 则报错。

- DBE_LOB.LOADBLOBFROMFILE

  该函数用于将BFILE外部文件读取到BLOB类型对象中，并以RAW类型返回该对象。

  DBE_LOB.LOADBLOBFROMFILE原型为：

```
DBE_LOB.LOADBLOBFROMFILE (
    dest_lob     IN BLOB,
    src_file     IN INTEGER,
    amount       IN INTEGER,
    dest_offset IN INTEGER,
    src_offset   IN INTEGER)
RETURN RAW;
```

  表 2-220 DBE_LOB.LOADBLOBFROMFILE 接口参数说明

参数	说明
dest_lob	目标BLOB类型对象，BFILE文件将读取到这个对象中。
src_file	需要读取的源BFILE文件。
amount	BLOB类型对象的长度，超过这个阈值的文件将不会保存到BLOB类型对象中。 **说明** 长度小于1或者大于32767则报错。
dest_offset	BLOB类型对象的偏移长度。 **说明** 偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	BFILE文件的偏移长度。 **说明** ● 偏移量小于1或者大于LOBMAXSIZE则报错。 ● amount + src_offset 大于 src_bfile的长度 + 1 则报错。

- DBE_LOB.LOADBLOBFROMBFILE

  LOADBLOBFROMBFILE用于将BFILE外部文件读取到BLOB类型对象中。

  DBE_LOB.LOADBLOBFROMBFILE原型为：

```
DBE_LOB.LOADBLOBFROMFILE (
    dest_lob     INOUT BLOB,
    src_file     IN     DBE_LOB.BFILE,
    amount       IN     BIGINT,
    dest_offset INOUT BIGINT,
    src_offset   INOUT BIGINT)
```


版权所有 © 华为云计算技术有限公司

表 2-221 DBE_LOB.LOADBLOBFROMBFILE 接口参数说明

参数	说明
dest_lob	IN参数为目标BLOB类型对象，BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开）将读取到这个对象中，OUT参数为完成读取后返回的BLOB类型对象。大小支持超过1GB，最大支持32TB。
src_file	需要读取的源BFILE文件，BFILE文件的大小支持超过1GB，最大支持32TB。
amount	BLOB类型对象的长度，超过这个阈值的文件将不会保存到BLOB类型对象中。 **说明** 长度小于1或者大于LOBMAXSIZE则报错。
dest_offset	BLOB类型对象的偏移长度。 **说明** 偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	BFILE文件的偏移长度。 **说明** • 偏移量小于1或者大于LOBMAXSIZE则报错。 • amount + src_offset 大于 src_bfile的长度 + 1 则报错。

- DBE_LOB.LOADCLOBFROMFILE

  该函数用于将BFILE外部文件读取到CLOB类型对象中，并以RAW类型返回该对象。

  DBE_LOB.LOADCLOBFROMFILE原型为：

```
DBE_LOB.LOADCLOBFROMFILE (
    dest_lob     IN CLOB,
    src_file     IN INTEGER,
    amount       IN INTEGER,
    dest_offset IN INTEGER,
    src_offset   IN INTEGER)
RETURN RAW;
```

表 2-222 DBE_LOB.LOADCLOBFROMFILE 接口参数说明

参数	说明
dest_lob	目标CLOB类型对象，BFILE文件将读取到这个文件中。
src_file	需要读取的源BFILE文件。
amount	CLOB类型对象的长度。 **说明** 长度小于1或者大于32767则报错。
dest_offset	CLOB类型对象的偏移长度。 **说明** 偏移量小于1或者大于LOBMAXSIZE则报错。

版权所有 © 华为云计算技术有限公司

参数	说明
src_offset	BFILE文件的偏移长度。 **说明** ● 偏移量小于1或者大于LOBMAXSIZE则报错。 ● amount + src_offset 大于 src_bfile的长度 + 1 则报错。

- DBE_LOB.LOADCLOBFROMBFILE

  LOADCLOBFROMBFILE用于将BFILE外部文件读取到CLOB类型对象中。

  DBE_LOB.LOADCLOBFROMBFILE原型为：

```
DBE_LOB.LOADCLOBFROMBFILE (
   dest_lob    INOUT CLOB,
   src_file    IN    DBE_LOB.BFILE,
   amount      IN    BIGINT,
   dest_offset INOUT BIGINT,
   src_offset  INOUT BIGINT)
```

  表 2-223 DBE_LOB.LOADCLOBFROMBFILE 接口参数说明

参数	说明
dest_lob	IN参数为目标CLOB类型对象，BFILE文件（必须先通过DBE_LOB.BFILEOPEN打开）将读取到这个对象中，OUT参数为完成读取后返回的CLOB类型对象。大小支持超过1GB，最大支持32TB。
src_file	需要读取的源BFILE文件。BFILE文件的大小支持超过1GB，最大支持32TB。
amount	CLOB类型对象的长度，超过这个阈值的文件将不会保存到CLOB类型对象中。 **说明** 长度小于1或者大于LOBMAXSIZE则报错。
dest_offset	CLOB类型对象的偏移长度。 **说明** 偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	BFILE文件的偏移长度。 **说明** ● 偏移量小于1或者大于LOBMAXSIZE则报错。 ● amount + src_offset 大于 src_bfile的长度 + 1 则报错。

- DBE_LOB.CONVERTTOBLOB

  该函数将CLOB类型对象转换成BLOB类型对象，不支持大于1GB的场景。

  DBE_LOB.CONVERTTOBLOB原型为：

```
DBE_LOB.CONVERTTOBLOB(
   dest_blob   IN BLOB,
   src_clob    IN CLOB,
   amount      IN INTEGER DEFAULT 32767,
   dest_offset IN INTEGER DEFAULT 1,
   src_offset  IN INTEGER DEFAULT 1)
RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-224 DBE_LOB.CONVERTTOBLOB 接口参数说明

参数	说明
dest_blob	目标BLOB类型对象，CLOB类型对象转换后的文件。
src_clob	需要读取的源CLOB类型对象。
amount	CLOB类型对象的长度，超过这个阈值的文件将不会保存到BLOB类型对象中。长度小于1或者大于LOBMAXSIZE则报错。
dest_offset	BLOB类型对象的偏移长度，dest_offset=1将从文件起始位置开始载入，以此类推。
src_offset	CLOB类型对象的偏移长度，src_offset=1将从文件起始位置开始读取，以此类推。

- DBE_LOB.LOB_CONVERTTOBLOB

  该函数将CLOB类型对象转换成BLOB类型对象，支持LOB类型对象大于1GB。

  DBE_LOB.LOB_CONVERTTOBLOB原型为：

```
DBE_LOB.LOB_CONVERTTOBLOB(
    dest_blob   INOUT BLOB,
    src_clob    IN    CLOB,
    amount      IN    BIGINT,
    dest_offset INOUT BIGINT,
    src_offset  INOUT BIGINT)
```

表 2-225 DBE_LOB.LOB_CONVERTTOBLOB 接口参数说明

参数	说明
dest_blob	目标BLOB类型对象，CLOB类型对象转换后的文件。
src_clob	需要读取的源CLOB类型对象。
amount	CLOB类型对象的长度，超过这个阈值的文件将不会保存到BLOB类型对象中。长度小于1或者大于LOBMAXSIZE则报错。
dest_offset	BLOB类型对象的偏移长度，dest_offset=1将从文件起始位置开始载入，以此类推。偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	CLOB类型对象的偏移长度，src_offset=1将从文件起始位置开始读取，以此类推。偏移量小于1或者大于LOBMAXSIZE则报错。

- DBE_LOB.CONVERTTOCLOB

  该函数将BLOB类型对象转换成CLOB类型对象，不支持大于1GB的场景。

  DBE_LOB.CONVERTTOCLOB原型为：

```
DBE_LOB.CONVERTTOCLOB(
    dest_clob   IN CLOB,
    src_blob    IN BLOB,
    amount      IN INTEGER DEFAULT 32767,
    dest_offset IN INTEGER DEFAULT 1,
```


版权所有 © 华为云计算技术有限公司

```
  src_offset  IN INTEGER DEFAULT 1)
RETURN text;
```

表 2-226 DBE_LOB.CONVERTTOCLOB 接口参数说明

参数	说明
dest_clob	目标CLOB类型对象，BLOB类型对象转换后的文件。
src_blob	需要读取的源BLOB类型对象。
amount	BLOB类型对象的长度，超过这个阈值的文件将不会保存到CLOB类型对象中。
dest_offset	CLOB类型对象的偏移长度，dest_offset=1将从文件起始位置开始载入，以此类推。
src_offset	BLOB类型对象的偏移长度，src_offset=1将从文件起始位置开始读取，以此类推。

- DBE_LOB.LOB_CONVERTTOCLOB

  该函数将BLOB类型对象转换成CLOB类型对象，支持LOB类型对象大于1G。

  DBE_LOB.LOB_CONVERTTOCLOB原型为：

```
DBE_LOB.LOB_CONVERTTOCLOB(
  dest_clob    INOUT CLOB,
  src_blob    IN    BLOB,
  amount      IN    BIGINT,
  dest_offset INOUT BIGINT,
  src_offset  INOUT BIGINT)
```

表 2-227 DBE_LOB.LOB_CONVERTTOCLOB 接口参数说明

参数	说明
dest_clob	目标CLOB类型对象，BLOB类型对象转换后的文件。
src_blob	需要读取的源BLOB类型对象。
amount	BLOB类型对象的长度，超过这个阈值的文件将不会保存到CLOB类型对象中。
dest_offset	CLOB类型对象的偏移长度，dest_offset=1将从文件起始位置开始载入，以此类推。偏移量小于1或者大于LOBMAXSIZE则报错。
src_offset	BLOB类型对象的偏移长度，src_offset=1将从文件起始位置开始读取，以此类推。偏移量小于1或者大于LOBMAXSIZE则报错。

- DBE_LOB.GETCHUNKSIZE

  在数据库存储LOB类型对象时，内部使用toast存储，本函数返回TOAST_MAX_CHUNK_SIZE。

  DBE_LOB.GETCHUNKSIZE的原型为：

```
DBE_LOB.GETCHUNKSIZE(
  lob_loc IN CLOB
```


版权所有 © 华为云计算技术有限公司

```
)RETURN INTEGER

DBE_LOB.GETCHUNKSIZE(
    lob_loc IN BLOB
)RETURN INTEGER
```

表 2-228 DBE_LOB.GETCHUNKSIZE 接口参数说明

参数	说明
lob_loc	目标CLOB类型对象/BLOB类型对象。

- DBE_LOB.LOB_WRITE

  将源对象从起始位置读取指定长度内容，写入目标LOB类型对象的指定偏移位置，覆盖该位置原内容, 并返回目标LOB类型对象。

  DBE_LOB.LOB_WRITE原型为：

```
DBE_LOB.LOB_WRITE(
    clob_obj INOUT CLOB,
    amount   IN    INTEGER,
    off_set  IN    BIGINT,
    source   IN    VARCHAR2
)
RETURN CLOB;

DBE_LOB.LOB_WRITE(
    blob_obj INOUT BLOB,
    amount   IN    INTEGER,
    off_set  IN    BIGINT,
    source   IN    RAW
)
RETURN BLOB;
```

表 2-229 DBE_LOB.LOB_WRITE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
blob_obj / clob_obj	BLOB/ CLOB	INOUT	否	IN参数为待写入的目标LOB类型对象，OUT参数为写入内容后的LOB类型对象。
amount	INTEGER	IN	否	待写入的长度（BLOB类型对象以字节为单位，CLOB类型对象以字符为单位）。
off_set	BIGINT	IN	否	在blob_obj/clob_obj中写入的偏移位置。
source	RAW/ VARCHAR2	IN	否	源对象。

- DBE_LOB.BFILENAME

版权所有 © 华为云计算技术有限公司

该函数用于根据目录和文件名构造BFILE文件。

DBE_LOB.BFILENAME原型为：

```
DBE_LOB.BFILENAME(
    directory IN TEXT,
    filename  IN TEXT)
RETURN DBE_LOB.BFILE;
```

表 2-230 DBE_LOB.BFILENAME 接口参数说明

参数	说明
directory	文件目录。 **说明** 文件目录的位置，需要添加到系统表**PG_DIRECTORY**中，如果传入的路径和**PG_DIRECTORY**中的路径不匹配，会报路径不存在的错误。 • 在打开GUC参数safe_data_path时，用户只能通过高级包读写safe_data_path指定文件路径下的文件。 • 文件名，包含扩展（文件类型），不包括路径名。如果文件名中包含路径，在OPEN中会被忽略，在Unix系统中，文件名不能以/.结尾。
filename	文件名。

示例

```
-- 获取字符串的长度。
SELECT DBE_LOB.GET_LENGTH('12345678');
 get_length
------------
          8
(1 row)

-- DBE_LOB.READ接口示例。
DECLARE
myraw  RAW(100);
amount INTEGER :=2;
buffer INTEGER :=1;
begin
DBE_LOB.READ('123456789012345',amount,buffer,myraw);
dbe_output.print_line(myraw);
end;
/
0123
ANONYMOUS BLOCK EXECUTE

CREATE TABLE blob_Table (t1 blob) DISTRIBUTE BY REPLICATION;
CREATE TABLE blob_Table_bak (t2 blob) DISTRIBUTE BY REPLICATION;
INSERT INTO blob_Table VALUES('abcdef');
INSERT INTO blob_Table_bak VALUES('22222');

-- DBE_LOB多接口示例。
DECLARE
str varchar2(100) := 'abcdef';
source raw(100);
dest blob;
copyto blob;
amount int;
PSV_SQL varchar2(100);
PSV_SQL1 varchar2(100);
a int :=1;
len int;
```


版权所有 © 华为云计算技术有限公司

```
BEGIN
source := dbe_raw.cast_from_varchar2_to_raw(str);
amount := dbe_raw.get_length(source);

PSV_SQL :='select * from blob_Table for update';
PSV_SQL1 := 'select * from blob_Table_bak for update';

EXECUTE IMMEDIATE PSV_SQL into dest;
EXECUTE IMMEDIATE PSV_SQL1 into copyto;

DBE_LOB.WRITE(dest, amount, 1, source);
DBE_LOB.WRITE_APPEND(dest, amount, source);

DBE_LOB.ERASE(dest, a, 1);
DBE_OUTPUT.PRINT_LINE(a);
DBE_LOB.COPY(copyto, dest, amount, 10, 1);
perform DBE_LOB.CLOSE(dest);
RETURN;
END;
/
1
ANONYMOUS BLOCK EXECUTE

-- 删除表。
DROP TABLE blob_Table;
DROP TABLE blob_Table_bak;
```

2.12.2.10 DBE_MATCH

接口介绍

高级功能包DBE_MATCH支持的接口请参见**表2-231**。

表 2-231 DBE_MATCH

接口名称	描述
DBE_MATCH.EDIT_DISTANCE_SIMILARITY	比较两个字符串的差距（删除、新增、变换的最小步骤），并归一化到0~100（100表示完全一致，0表示完全不一致）。

- DBE_MATCH.EDIT_DISTANCE_SIMILARITY

  比较两个字符串的差距（删除、新增和变换的最小步骤），并归一化到0~100（100表示完全一致，0表示完全不一致）。

  DBE_MATCH.EDIT_DISTANCE_SIMILARITY原型为：

  ```
  DBE_MATCH.EDIT_DISTANCE_SIMILARITY(
     str1 IN text,
     str2 IN text
  )returns integer ;
  ```

  表 2-232 DBE_MATCH.EDIT_DISTANCE_SIMILARITY 接口参数说明

参数	描述
str1	第一个字符串，如果为null，直接输出0。
str2	第二个字符串，如果为null，直接输出0。

版权所有 © 华为云计算技术有限公司

2.12.2.11 DBE_OBFUSCATION_TOOLKIT

接口介绍

须知

由于DES算法、DES3算法和MD5哈希算法存在算法安全风险，请谨慎使用DBE_OBFUSCATION_TOOLKIT进行加解密操作。

DBE_OBFUSCATION_TOOLKIT高级功能包是GaussDB提供的兼容性功能包，主要用于支持DES、DES3、MD5哈希算法的接口，以保持与原有业务逻辑的兼容。DBE_OBFUSCATION_TOOLKIT支持的接口请参见**表2-233**。

表 2-233 DBE_OBFUSCATION_TOOLKIT

接口名称	描述
DBE_OBFUSCATION_TOOLKIT.DESGETKEY	生成DES算法的密钥。
DBE_OBFUSCATION_TOOLKIT.DESENCRYPT	使用DES算法对数据进行加密。
DBE_OBFUSCATION_TOOLKIT.DESDECRYPT	使用DES算法对数据进行解密。
DBE_OBFUSCATION_TOOLKIT.DES3GETKEY	生成DES3算法的密钥。
DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT	使用DES3算法对数据进行加密。
DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT	使用DES3算法对数据进行解密。
DBE_OBFUSCATION_TOOLKIT.MD5	计算数据的MD5哈希值。

- DBE_OBFUSCATION_TOOLKIT.DESGETKEY

  功能：生成DES算法的密钥。

  DBE_OBFUSCATION_TOOLKIT.DESGETKEY原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.DESGETKEY(
seed_string IN VARCHAR2
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.DESGETKEY(
seed IN RAW
) RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-234 DBE_OBFUSCATION_TOOLKIT.DESGETKEY 接口参数说明

参数	类型	是否可以为空	描述
seed_string	VARCHAR2	否	用于生成密钥的种子值，其长度至少为80字节。
seed	RAW	否	

示例：

```
-- 创建ORA兼容模式的数据库
gaussdb=# CREATE DATABASE ora_db dbcompatibility = 'ORA' encoding = 'utf8';
gaussdb=# \c ora_db
-- 生成随机密钥
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.desgetkey(seed_string =>
'0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000')));
WARNING:  The DES algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 A7F0B24CF66DAECF
(1 row)

ora_db=# SELECT rawtohex(dbe_obfuscation_toolkit.desgetkey(seed =>
dbe_raw.cast_from_varchar2_to_raw('00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000')));
WARNING:  The DES algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 2A5984CA3AB67344
(1 row)
```

说明

DBE_OBFUSCATION_TOOLKIT.DESGETKEY生成的密钥为随机密钥。

- DBE_OBFUSCATION_TOOLKIT.DESENCRYPT

  功能：使用DES算法对数据进行加密。

  DBE_OBFUSCATION_TOOLKIT.DESENCRYPT原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.DESENCRYPT(
input_string IN VARCHAR2,
key_string   IN VARCHAR2
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.DESENCRYPT(
input IN RAW,
key   IN RAW
) RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-235 DBE_OBFUSCATION_TOOLKIT.DESENCRYPT 接口参数说明

参数	类型	是否可以为空	描述
input_string	VARCHAR2	否	待加密的数据，其长度必须为8字节的倍数。
input	RAW	否	
key_string	VARCHAR2	否	加密密钥，其长度至少为8字节，超过8字节的数据不影响加密结果。
key	RAW	否	

示例：

```
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.desencrypt(input_string =>
'12345678', key_string => '12345678')));
WARNING:  The DES algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 96D0028878D58C89
(1 row)

ora_db=# SELECT rawtohex(dbe_obfuscation_toolkit.desencrypt(input =>
dbe_raw.cast_from_varchar2_to_raw('12345678'), key =>
dbe_raw.cast_from_varchar2_to_raw('12345678')));
WARNING:  The DES algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 96D0028878D58C89
(1 row)
```

说明

由于该函数执行过程需传入密钥，出于安全考虑，gsql工具不会将包含该函数名称的SQL记录至执行历史。因此，无法在gsql中通过上下翻页功能查找该函数的执行历史。

- DBE_OBFUSCATION_TOOLKIT.DESDECRYPT

  功能：使用DES算法对数据进行解密。

  DBE_OBFUSCATION_TOOLKIT.DESDECRYPT原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.DESDECRYPT(
input_string IN VARCHAR2,
key_string   IN VARCHAR2
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.DESDECRYPT(
input IN RAW,
key   IN RAW
) RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-236 DBE_OBFUSCATION_TOOLKIT.DESDECRYPT 接口参数说明

参数	类型	是否可以为空	描述
input_string	VARCHAR2	否	待解密的数据，其长度必须为8字节的倍数。
input	RAW	否	
key_string	VARCHAR2	否	解密密钥，其长度至少为8字节，超过8字节的数据不影响解密结果。
key	RAW	否	

示例：

```
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.desdecrypt(input_string =>
'12345678', key_string => '12345678')));
WARNING:  The DES algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 F5E8E9EB81F28B73
(1 row)

ora_db=# SELECT rawtohex(dbe_obfuscation_toolkit.desdecrypt(input =>
dbe_raw.cast_from_varchar2_to_raw('12345678'), key =>
dbe_raw.cast_from_varchar2_to_raw('12345678')));
WARNING:  The DES algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 F5E8E9EB81F28B73
(1 row)
```

说明

由于该函数执行过程需传入密钥，出于安全考虑，gsql工具不会将包含该函数名称的SQL记录至执行历史。因此，无法在gsql中通过上下翻页功能查找该函数的执行历史。

- DBE_OBFUSCATION_TOOLKIT.DES3GETKEY

  功能：生成DES3算法的密钥。

  DBE_OBFUSCATION_TOOLKIT.DES3GETKEY原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.DES3GETKEY(
which       IN INTEGER DEFAULT 0,
seed_string IN VARCHAR2
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.DES3GETKEY(
which IN INTEGER DEFAULT 0,
seed  IN RAW
) RETURN RAW;
```


版权所有 © 华为云计算技术有限公司

表 2-237 DBE_OBFUSCATION_TOOLKIT.DES3GETKEY 接口参数说明

参数	类型	是否可以为空	描述
which	INTEGER	否	密钥模式。 取值范围：0或1 ● 0：返回16字节的密钥（即两个密钥）。 ● 1：返回24字节的密钥（即三个密钥）。 默认值：0
seed_string	VARCHAR2	否	用于生成密钥的种子值，其长度至少为80字节。
seed	RAW	否	

示例：

```
-- 生成随机密钥
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.des3getkey(seed_string =>
'00000000000000000000000000000000000000000000000000000000000000000000000000000000')));
WARNING:  The DES3 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
             rawtohex
----------------------------------
 9A64E470BC16D4969E782F3F5A8491A6
(1 row)

ora_db=# SELECT rawtohex(dbe_obfuscation_toolkit.des3getkey(which => 1, seed =>
dbe_raw.cast_from_varchar2_to_raw('00000000000000000000000000000000000000000000000000000
000000000000000000000000000')));
WARNING:  The DES3 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
                     rawtohex
--------------------------------------------------
 7497F5D61F23E751D3D337EE6B88B257350FF8B9732BC288
(1 row)
```

说明

DBE_OBFUSCATION_TOOLKIT.DES3GETKEY生成的密钥为随机密钥。

- DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT

  功能：使用DES3算法对数据进行加密。

  DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT(
input_string IN VARCHAR2,
key_string   IN VARCHAR2,
which        IN INTEGER DEFAIULT 0,
iv_string    IN VARCHAR2 DEFAULT NULL
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT(
input IN RAW,
key   IN RAW,
which IN INTEGER DEFAIULT 0,
iv    IN RAW DEFAULT NULL
) RETURN RAW;
```

表 2-238 DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT 接口参数说明

参数	类型	是否可以为空	描述
input_string	VARCHAR2	否	待加密的数据，其长度必须为8字节的倍数。
input	RAW	否	
key_string	VARCHAR2	否	加密密钥，其长度依赖which的取值。 ● 当which取值为0时：加密密钥的长度至少为16字节，超过16字节的数据不影响加密结果。 ● 当which取值为1时：加密密钥的长度至少为24字节，超过24字节的数据不影响加密结果。
key	RAW	否	
which	INTEGER	否	密钥模式。 取值范围：0或1 ● 0：按照16字节的密钥模式（即两个密钥）进行加密。 ● 1：按照24字节的密钥模式（即三个密钥）进行加密。 默认值：0
iv_string	VARCHAR2	是	初始化向量，当该参数取值不为NULL时，其长度必须为8字节的倍数，超过8字节的数据不影响加密结果。 默认值：NULL
iv	RAW	是	

示例：

```
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.des3encrypt(input_string =>
'12345678', key_string => '1234567823456789', iv_string => '12345678')));
WARNING:  The DES3 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 EFDD1DFBCEE3B0CA
(1 row)

ora_db=# SELECT rawtohex(dbe_obfuscation_toolkit.des3encrypt(input =>
dbe_raw.cast_from_varchar2_to_raw('12345678'), key =>
dbe_raw.cast_from_varchar2_to_raw('1234567812345678876543210'), which => 1));
WARNING:  The DES3 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 1A1546D3DEAEB8BE
(1 row)
```

说明

由于该函数执行过程需传入密钥，出于安全考虑，gsql工具不会将包含该函数名称的SQL记录至执行历史。因此，无法在gsql中通过上下翻页功能查找该函数的执行历史。

版权所有 © 华为云计算技术有限公司

- DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT

  功能：使用DES3算法对数据进行解密。

  DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT(
input_string IN VARCHAR2,
key_string   IN VARCHAR2,
which        IN INTEGER DEFAIULT 0,
iv_string    IN VARCHAR2 DEFAULT NULL
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT(
input IN RAW,
key   IN RAW,
which IN INTEGER DEFAIULT 0,
iv    IN RAW DEFAULT NULL
) RETURN RAW;
```

  表 2-239 DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT 接口参数说明

参数	类型	是否可以为空	描述
input_string	VARCHAR2	否	待解密的数据，其长度必须为8字节的倍数。
input	RAW	否	
key_string	VARCHAR2	否	解密密钥，其长度依赖which的取值。 ● 当which取值为0时：解密密钥的长度至少为16字节，超过16字节的数据不影响解密结果。 ● 当which取值为1时：解密密钥的长度至少为24字节，超过24字节的数据不影响解密结果。
key	RAW	否	
which	INTEGER	否	密钥模式。 取值范围：0或1 ● 0：按照16字节的密钥模式（即两个密钥）进行解密。 ● 1：按照24字节的密钥模式（即三个密钥）进行解密。 默认值：0
iv_string	VARCHAR2	是	初始化向量，当该参数取值不为NULL时，其长度必须为8字节的倍数，超过8字节的数据不影响解密结果。 默认值：NULL
iv	RAW	是	

  示例：

```
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.des3decrypt(input_string =>
```


版权所有 © 华为云计算技术有限公司

```
'12345678', key_string => '1234567823456789', iv_string => '12345678')));
WARNING:  The DES3 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 8C2ED115CCA35582
(1 row)

ora_db=# SELECT rawtohex(dbe_obfuscation_toolkit.des3decrypt(input =>
dbe_raw.cast_from_varchar2_to_raw('12345678'), key =>
dbe_raw.cast_from_varchar2_to_raw('123456781234567887654321 0'), which => 1, iv =>
dbe_raw.cast_from_varchar2_to_raw('12345678')));
WARNING:  The DES3 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
    rawtohex
------------------
 A6FF1AA4A875B3E8
(1 row)
```

说明

由于该函数执行过程需传入密钥，出于安全考虑，gsql工具不会将包含该函数名称的SQL记录至执行历史。因此，无法在gsql中通过上下翻页功能查找该函数的执行历史。

- DBE_OBFUSCATION_TOOLKIT.MD5

  功能：计算数据的MD5哈希值。

  DBE_OBFUSCATION_TOOLKIT.MD5原型可以分为2种：

```
--原型1：入参为VARCHAR2类型
DBE_OBFUSCATION_TOOLKIT.MD5(
input_string IN VARCHAR2
) RETURN VARCHAR2;

--原型2：入参为RAW类型
DBE_OBFUSCATION_TOOLKIT.MD5(
input IN RAW
) RETURN RAW;
```

表 2-240 DBE_OBFUSCATION_TOOLKIT.MD5 接口参数说明

参数	类型	是否可以为空	描述
input_string	VARCHAR2	否	待进行哈希操作的数据。
input	RAW	否	

示例：

```
ora_db=# SELECT
rawtohex(dbe_raw.cast_from_varchar2_to_raw(dbe_obfuscation_toolkit.md5(input_string =>
'1234567890')));
WARNING:  The MD5 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
            rawtohex
----------------------------------
 E807F1FCF82D132F9BB018CA6738A19F
(1 row)

ora_db=# SELECT
rawtohex(dbe_obfuscation_toolkit.md5(dbe_raw.cast_from_varchar2_to_raw('1234567890')));
WARNING:  The MD5 algorithm is an insecure algorithm, please use it with caution.
CONTEXT:  referenced column: rawtohex
```


版权所有 © 华为云计算技术有限公司

```
              rawtohex
----------------------------------
 E807F1FCF82D132F9BB018CA6738A19F
(1 row)

-- 删除ora_db数据库
ora_db=# \c postgres
gaussdb=# DROP DATABASE ora_db;
```

2.12.2.12 DBE_OUTPUT

说明

当使用DBE_OUTPUT.PUT_LINE打印DBE_FILE.READ_LINE_NCHAR接口得到的结果时，需要确保UTF8字符集编码能够转换成当前数据库字符集编码，满足上述条件后可以正常输出结果。DBE_OUTPUT.PRINT_LINE不支持该功能。

接口介绍

高级功能包DBE_OUTPUT支持的接口请参见**表2-241**。

表 2-241 DBE_OUTPUT

接口名称	描述
DBE_OUTPUT.PRINT_LINE	输出指定的文本，并添加换行符。
DBE_OUTPUT.PRINT	输出指定的文本，不添加换行符。
DBE_OUTPUT.SET_BUFFER_SIZE	设置输出缓冲区的大小，如果不指定则缓冲区最大能容纳20000字节，如果指定小于等于2000字节，则缓冲区允许容纳2000字节。
DBE_OUTPUT.DISABLE	禁用PUT、PUT_LINE、NEW_LINE、GET_LINE、GET_LINES调用，并清空输出缓冲区。
DBE_OUTPUT.ENABLE	开启缓冲区，允许对PUT、PUT_LINE、NEW_LINE、GET_LINE和GET_LINES的调用，设置缓冲区大小。
DBE_OUTPUT.GET_LINE	从缓冲区中以换行符作为分界获取一行数据，获取的数据将不会输出到客户端。
DBE_OUTPUT.GET_LINES	以VARCHAR数组的形式获取缓冲区的指定行数的字符串，被取出的内容将会在缓冲区中清除，不会输出到客户端。
DBE_OUTPUT.NEW_LINE	放置一行在缓冲区末尾，放置行尾标记，空出新的一行。
DBE_OUTPUT.PUT	将输入字符串放入到缓冲区，末尾不加换行符，在匿名块执行结束时会将以换行符结尾的行输出显示。
DBE_OUTPUT.PUT_LINE	将输入字符串放入到缓冲区，并末尾添加换行符，在匿名块执行结束时会将以换行符结尾的行输出显示。

版权所有 © 华为云计算技术有限公司

- DBE_OUTPUT.PRINT_LINE

  存储过程PRINT_LINE输出指定的文本，并添加换行符。

  DBE_OUTPUT.PRINT_LINE原型为：

  ```
  DBE_OUTPUT.PRINT_LINE (
  format IN VARCHAR2);
  ```

  表 2-242 DBE_OUTPUT.PRINT_LINE 接口参数说明

参数	描述
format	输出的文本。

- DBE_OUTPUT.PRINT

  存储过程PRINT输出指定的文本，不添加换行符。

  DBE_OUTPUT.PRINT原型为：

  ```
  DBE_OUTPUT.PRINT (
  format IN VARCHAR2);
  ```

  表 2-243 DBE_OUTPUT.PRINT 接口参数说明

参数	描述
format	输出的文本。

- DBE_OUTPUT.SET_BUFFER_SIZE

  存储过程SET_BUFFER_SIZE设置输出缓冲区的大小，如果不指定的话缓冲区最大只能容纳20000字节。

  DBE_OUTPUT.SET_BUFFER_SIZE原型为：

  ```
  DBE_OUTPUT.SET_BUFFER_SIZE (
  size IN INTEGER default 20000);
  ```

  表 2-244 DBE_OUTPUT.SET_BUFFER_SIZE 接口参数说明

参数	描述
size	设置输出缓冲区的大小。取值范围为2000 ~ 1000000字节，若传入值小于2000则自动调整为2000，若大于1000000则自动调整为1000000。

- DBE_OUTPUT.DISABLE

  存储过程DISABLE禁用PUT、PUT_LINE、NEW_LINE、GET_LINE、GET_LINES调用，并清空输出缓冲区。

  DBE_OUTPUT.DISABLE原型为：

  ```
  DBE_OUTPUT.DISABLE;
  ```

- DBE_OUTPUT.ENABLE

  存储过程ENABLE开启缓冲区，允许对PUT、PUT_LINE、NEW_LINE、GET_LINE和GET_LINES的调用，并设置缓冲区大小，如果未指定缓冲区大小则默认容纳20000字节。

版权所有 © 华为云计算技术有限公司

DBE_OUTPUT.ENABLE原型为：

```
DBE_OUTPUT.ENABLE (
    buffer_size IN INTEGER DEFAULT 20000
);
```

表 2-245 DBE_OUTPUT.ENABLE 接口参数说明

参数	描述
buffer_size	缓冲区大小的上限，以字节为单位。将buffer_size设置为NULL表示使用默认值（20000字节）。

- DBE_OUTPUT.GET_LINE

  存储过程GET_LINE从缓冲区中以换行符作为分界获取一行数据，获取的数据将不会输出到客户端。

  DBE_OUTPUT.GET_LINE原型为：

  ```
  DBE_OUTPUT.GET_LINE (
      line   OUT VARCHAR2,
      status OUT INTEGER
  );
  ```

  表 2-246 DBE_OUTPUT.GET_LINE 接口参数说明

参数	描述
line	表示获取到的字符串。
status	表示调用状态是否正常，如果获取到字符串则为0，否则为1。

- DBE_OUTPUT.GET_LINES

  存储过程GET_LINES以VARCHAR数组的形式获取缓冲区的指定行数的字符串，被取出的内容将会在缓冲区中清除，不会输出到客户端。

  DBE_OUTPUT.GET_LINES原型为：

  ```
  DBE_OUTPUT.GET_LINES (
      lines    OUT   VARCHAR[],
      numlines IN OUT INTEGER
  );
  ```

  表 2-247 DBE_OUTPUT.GET_LINES 接口参数说明

参数	描述
lines	输出从缓冲区读取的多行字符串数组。
numlines	输入值为需要从缓冲区检索的行数。输出值为实际检索到的行数，取值范围为大于0且不能为NULL，如果输出值小于输入值，则表明缓冲区中存在的行数小于输入的行数。

- DBE_OUTPUT.NEW_LINE

  存储过程NEW_LINE放置一行在缓冲区末尾，放置行尾标记，空出新的一行。

  DBE_OUTPUT.NEW_LINE原型为：

  ```
  DBE_OUTPUT.NEW_LINE;
  ```


版权所有 © 华为云计算技术有限公司

- DBE_OUTPUT.PUT

  存储过程PUT将输入字符串放入到缓冲区，末尾不加换行符，在匿名块执行结束时会将以换行符结尾的行输出显示。

  DBE_OUTPUT.PUT原型为：

  ```
  DBE_OUTPUT.PUT (
      item IN VARCHAR2);
  ```

  表 2-248 DBE_OUTPUT.PUT 接口参数说明

参数	描述
item	待放在缓存区的字符或字符串item。单行最大长度限制为32767字节。

- DBE_OUTPUT.PUT_LINE

  存储过程PUT_LINE将输入字符串放入到缓冲区，并且末尾添加换行符，在匿名块执行结束时会将以换行符结尾的行输出显示。

  DBE_OUTPUT.PUT_LINE原型为：

  ```
  DBE_OUTPUT.PUT_LINE (
      item IN VARCHAR2);
  ```

  表 2-249 DBE_OUTPUT.PUT_LINE 接口参数说明

参数	描述
item	待放在缓存区的字符或字符串。单行最大长度限制为32767字节。

示例

```
BEGIN
    DBE_OUTPUT.SET_BUFFER_SIZE(50);
    DBE_OUTPUT.PRINT('hello, ');
    DBE_OUTPUT.PRINT_LINE('database!'); -- 输出hello, database!。
END;
/
-- 预期结果为：
hello, database!
ANONYMOUS BLOCK EXECUTE

-- 测试disable禁用put、put_line、new_line、get_line、get_lines调用,测试put_line不输出。
BEGIN
 dbe_output.disable();
        dbe_output.put_line('1');
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE

-- 测试enable启用put、put_line、new_line、get_line、get_lines调用，测试put_line输出1。
BEGIN
 dbe_output.enable();
        dbe_output.put_line('1');
END;
/
-- 预期结果为：
1
```


版权所有 © 华为云计算技术有限公司

```
ANONYMOUS BLOCK EXECUTE

-- 测试put，输入字符串a放入到缓冲区，末尾不加换行符，a不输出。
BEGIN
 dbe_output.enable();
 dbe_output.put('a');
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE

-- 测试new_line,添加换行，输出a。
BEGIN
 dbe_output.enable();
 dbe_output.put('a');
        dbe_output.new_line;
END;
/
-- 预期结果为：
a
ANONYMOUS BLOCK EXECUTE

-- 测试get_line获取缓冲区数据保存到变量，使用put_line输出。
DECLARE
 line VARCHAR(32672);
 status INTEGER := 0;
BEGIN
 dbe_output.put_line('hello');
 dbe_output.get_line(line, status);
 dbe_output.put_line('-----------');
 dbe_output.put_line(line);
 dbe_output.put_line(status);
END;
/
-- 预期结果为：
-----------
hello
0
ANONYMOUS BLOCK EXECUTE

-- 测试get_lines获取缓冲区多行内容，使用put_line输出。
DECLARE
    lines varchar[];
    linenum integer;
BEGIN
    dbe_output.put_line('line 1');
    dbe_output.put_line('line 2');
    dbe_output.put_line('line 3');
    linenum := 100;
    dbe_output.get_lines(lines, linenum);
    dbe_output.put_line('num: ' || linenum);
    FOR i IN 1 .. linenum LOOP
        dbe_output.put_line(lines[i]);
    END LOOP;
END;
/
-- 预期结果为：
num: 3
line 1
line 2
line 3
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

说明

当服务端的字符编码类型server_encoding不为UTF8时，如果数据中存在字符编码属于合法的UTF8编码，DBE_OUTPUT.PUT_LINE函数和DBE_OUTPUT.PUT函数在处理该字符编码时会将其按照UTF8编码逻辑进行处理，将其先转换成server_encoding的编码格式后再进行后续操作，这样可能会导致函数返回值不符合预期或发生报错。当开启GUC参数enable_convert_illegal_char后（enable_convert_illegal_char值为on），DBE_OUTPUT.PUT_LINE函数和DBE_OUTPUT.PUT函数在处理此类编码的表现与参数关闭时保持一致，特殊字符不会以占位符形式输出，因此不建议用户对含有特殊字符的数据使用DBE_OUTPUT.PUT_LINE函数和DBE_OUTPUT.PUT函数。

2.12.2.13 DBE_RANDOM

接口介绍

高级功能包DBE_RANDOM支持的接口请参见**表2-250**。

表 2-250 DBE_RANDOM 接口参数说明

接口名称	描述
DBE_RANDOM.SET_SEED	设置一个随机数的种子。
DBE_RANDOM.GET_VALUE	生成一个大小介于指定的low及high之间的随机数。

- DBE_RANDOM.SET_SEED
  存储过程SEED用于设置一个随机数的种子。
  DBE_RANDOM.SET_SEED原型为：

```
DBE_RANDOM.SET_SEED (seed  IN  INTEGER);
```

表 2-251 DBE_RANDOM.SET_SEED 接口参数说明

参数	描述
seed	用于产生一个随机数的种子。

- DBE_RANDOM.GET_VALUE
  函数GET_VALUE生成一个大小介于指定的low及high之间的随机数。
  DBE_RANDOM.GET_VALUE原型为：

```
DBE_RANDOM.GET_VALUE(
min  IN  NUMBER default 0,
max  IN  NUMBER default 1)
RETURN NUMBER;
```

表 2-252 DBE_RANDOM.GET_VALUE 接口参数说明

参数	描述
min	指定随机数大小的下边界，生成的随机数大于或等于min。

版权所有 © 华为云计算技术有限公司

参数	描述
max	指定随机数大小的上边界，生成的随机数小于max。

说明

- DBE_RANDOM.GET_VALUE要求参数类型是NUMERIC，对于左右边界的大小没有要求。
- DBE_RANDOM实现的是伪随机，所以若使用的初值（种子）不变，那么伪随机数的数序也不变，使用时需要注意，建议不要在安全场景下使用DBE_RANDOM。
- 生成的随机数有效数字为15位。

示例：

```
-- 产生0到1之间的随机数。
SELECT DBE_RANDOM.GET_VALUE(0,1);
    get_value
------------------
 .917468812743886
(1 row)

-- 对于指定范围内的整数，要加入参数min和max，并从结果中截取较小的数（最大值不能被作为可能的值）。所以对于0到99之间的整数，使用下面的代码。
SELECT TRUNC(DBE_RANDOM.GET_VALUE(0,100));
 trunc
-------
    26
(1 row)
```

2.12.2.14 DBE_RAW

DBE_RAW包提供了用于操作RAW类型的接口。

说明

RAW类型的外部表现形式是十六进制，内部存储形式是二进制。例如一个RAW类型的数据11001011的表现形式为‘CB’，即在实际的类型转换中输入的是‘CB’。

权限介绍

DBE_RAW模式属于系统目录，DBE_RAW模式除了系统管理员用户在初始化数据库时拥有创建权限，模式及模式下的所有接口，所有用户仅具有使用权限。

接口介绍

高级功能包DBE_RAW支持的接口请参见表2-253。

表 2-253 DBE_RAW

接口名称	描述
DBE_RAW.CAST_FROM_BINARY_INTEGER_TO_RAW	将INTEGER类型值转换为二进制表示形式（即RAW类型）。
DBE_RAW.CAST_FROM_RAW_TO_BINARY_INTEGER	将RAW类型值转换为INTEGER类型。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_RAW.GET_LENGTH	获取RAW类型对象的长度。
DBE_RAW.CAST_FROM_VARCHAR2_TO_RAW	将VARCHAR2类型值转化为RAW类型。
DBE_RAW.CAST_TO_VARCHAR2	将RAW类型值转换成VARCHAR2类型。
DBE_RAW.BIT_OR	返回RAW类型值按位或计算后的值。
DBE_RAW.SUBSTR	返回RAW类型值取子串后的值。
DBE_RAW.BIT_AND	返回RAW类型按位与计算后的结果。
DBE_RAW.BIT_COMPLEMENT	返回RAW类型按位取反计算后的结果。
DBE_RAW.BIT_XOR	返回RAW类型按位异或计算后的结果。
DBE_RAW.CAST_FROM_BINARY_DOUBLE_TO_RAW	将BINARY_DOUBLE类型值转换为RAW类型。
DBE_RAW.CAST_FROM_RAW_TO_BINARY_DOUBLE	将RAW类型值转换为BINARY_DOUBLE类型。
DBE_RAW.CAST_FROM_RAW_TO_BINARY_FLOAT	将RAW类型值转换为FLOAT4类型。
DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW	将FLOAT4类型值转换为RAW类型。
DBE_RAW.CAST_FROM_RAW_TO_NUMBER	将RAW类型值转换为NUMERIC类型。
DBE_RAW.CAST_FROM_NUMBER_TO_RAW	将NUMERIC类型值转换为RAW类型。
DBE_RAW.CAST_FROM_RAW_TO_NVARCHAR2	将RAW类型值转换为NVARCHAR2类型。
DBE_RAW.COMPARE	返回两个RAW类型值首个不相同的位置。
DBE_RAW.CONCAT	将最多12个RAW类型值连接到一个新的RAW类型值中，并返回。
DBE_RAW.CONVERT	将一个RAW类型值从原编码方式from_charset转换为目标编码方式to_charset。
DBE_RAW.COPIES	将一个RAW类型值复制n次，串联在一起，并返回连接后的结果。
DBE_RAW.OVERLAY	用一个RAW类型数据对另一个RAW类型数据进行覆盖，可以指定起始覆盖位置和覆盖长度。
DBE_RAW.REVERSE	将RAW类型数据按字节进行翻转。
DBE_RAW.TRANSLATE	转换或舍弃RAW类型值的指定字节。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_RAW.TRANSLITERATE	转换RAW类型值的指定字节。
DBE_RAW.XRANGE	从一个字节到另一个字节的所有字节拼接后返回。

- DBE_RAW.CAST_FROM_BINARY_INTEGER_TO_RAW

  CAST_FROM_BINARY_INTEGER_TO_RAW将INTEGER类型值转换为RAW类型。

  DBE_RAW.CAST_FROM_BINARY_INTEGER_TO_RAW原型为：

  ```
  DBE_RAW.CAST_FROM_BINARY_INTEGER_TO_RAW (
  value          IN  BIGINT,
  endianess      IN  INTEGER DEFAULT 1)
  RETURN RAW;
  ```

  表 2-254 DBE_RAW.CAST_FROM_BINARY_INTEGER_TO_RAW 接口参数说明

参数	描述
value	待转成RAW类型的INTEGER类型值。 **说明** DBE_RAW.CAST_FROM_BINARY_INTEGER_TO_RAW预期处理INTEGER类型值，为了兼容性考虑，该参数被定义为BIGINT类型，BIGINT类型范围比INTEGER类型范围更大，不影响接口的使用。
endianess	表示字节序的整型值1或2。 • 1代表BIG_ENDIAN。默认值为BIG_ENDIAN。 • 2代表LITTLE-ENDIAN。

- DBE_RAW.CAST_FROM_RAW_TO_BINARY_INTEGER

  CAST_FROM_RAW_TO_BINARY_INTEGER将RAW类型值转换为INTEGER类型。

  DBE_RAW.CAST_FROM_RAW_TO_BINARY_INTEGER原型为：

  ```
  DBE_RAW.CAST_FROM_RAW_TO_BINARY_INTEGER (
  value          IN  RAW,
  endianess      IN  INTEGER DEFAULT 1)
  RETURN INTEGER;
  ```

  表 2-255 DBE_RAW.CAST_FROM_RAW_TO_BINARY_INTEGER 接口参数说明

参数	描述
value	待转成INTEGER类型的RAW类型值。

版权所有 © 华为云计算技术有限公司

参数	描述
endianess	字节序，取值范围：整型值1、2或3。 ● 1表示BIG_ENDIAN。默认值为BIG_ENDIAN。 ● 2表示LITTLE_ENDIAN。 ● 3表示MACHINE_ENDIAN。 如果机器默认为BIG_ENDIAN，则函数执行结果MACHINE_ENDIAN和BIG_ENDIAN一致，如果机器默认为LITTLE_ENDIAN，则函数执行结果MACHINE_ENDIAN和LITTLE_ENDIAN一致。

- DBE_RAW.GET_LENGTH

  GET_LENGTH返回RAW类型值的长度。

  DBE_RAW.GET_LENGTH原型为：

  ```
  DBE_RAW.GET_LENGTH(
  value     IN     RAW)
  RETURN INTEGER;
  ```

  表 2-256 DBE_RAW.GET_LENGTH 接口参数说明

参数	描述
value	RAW类型对象。

- DBE_RAW.CAST_FROM_VARCHAR2_TO_RAW

  CAST_FROM_VARCHAR2_TO_RAW将VARCHAR2类型值转换成RAW类型。

  DBE_RAW.CAST_FROM_VARCHAR2_TO_RAW原型为：

  ```
  DBE_RAW.CAST_TO_RAW(
  str     IN     VARCHAR2)
  RETURN RAW;
  ```

  表 2-257 DBE_RAW.CAST_FROM_VARCHAR2_TO_RAW 接口参数说明

参数	描述
str	待转成RAW类型的VARCHAR2类型值。

- DBE_RAW.CAST_TO_VARCHAR2

  CAST_TO_VARCHAR2将RAW类型值转换成VARCHAR2类型。

  DBE_RAW.CAST_TO_VARCHAR2原型为：

  ```
  DBE_RAW.CAST_TO_VARCHAR2(
  str     IN     RAW)
  RETURN VARCHAR2;
  ```


版权所有 © 华为云计算技术有限公司

表 2-258 DBE_RAW.CAST_TO_VARCHAR2 接口参数说明

参数	描述
str	待转成RAW类型的VARCHAR2类型值。

- DBE_RAW.BIT_OR

  BIT_OR返回两个RAW类型值按位或计算后的值。

  DBE_RAW.BIT_OR原型为：

```
DBE_RAW.BIT_OR(
    str1 IN TEXT,
    str2 IN TEXT)
RETURN TEXT;
```

表 2-259 DBE_RAW.BIT_OR 接口参数说明

参数	描述
str1	按位或的第一个字符串。 **说明** 由于历史原因，该参数被定义为TEXT类型，但是DBE_RAW.BIT_OR接口预期并处理RAW类型值，这里定义为TEXT类型不影响对RAW类型值的处理。
str2	按位或的第二个字符串。 **说明** 由于历史原因，该参数被定义为TEXT类型，但是DBE_RAW.BIT_OR接口预期并处理RAW类型值，这里定义为TEXT类型不影响对RAW类型值的处理。

- DBE_RAW.SUBSTR

  SUBSTR将RAW类型值按起始位off_set和长度amount截取。

  DBE_RAW.SUBSTR原型为：

```
DBE_RAW.SUBSTR(
    IN lob_loc BLOB,
    IN off_set INTEGER DEFAULT 1,
    IN amount INTEGER DEFAULT 32767)
RETURN RAW;
```

表 2-260 DBE_RAW.SUBSTR 接口参数说明

参数	描述
lob_loc	待截取的RAW类型值。 **说明** 由于历史原因，该参数被定义为BLOB类型，但是DBE_RAW.SUBSTR接口预期并处理RAW类型值，这里定义为BLOB类型不影响对RAW类型值的处理。
off_set	子串的起始位置，默认值1。
amount	子串的长度，默认值32767。

版权所有 © 华为云计算技术有限公司

- DBE_RAW.BIT_AND

  BIT_AND求两个RAW类型值按位与的结果。

  DBE_RAW.BIT_AND原型为：

```
DBE_RAW.BIT_AND(
   r1 IN RAW,
   r2 IN RAW)
RETURN RAW;
```

  表 2-261 DBE_RAW.BIT_AND 接口参数说明

参数	描述
r1	与r2进行按位与的RAW类型值，最大长度不超过32767。
r2	与r1进行按位与的RAW类型值，最大长度不超过32767。

- DBE_RAW.BIT_COMPLEMENT

  BIT_COMPLEMENT求一个RAW类型值按位取反的结果。

  DBE_RAW.BIT_COMPLEMENT原型为：

```
DBE_RAW.BIT_COMPLEMENT(
   r IN RAW)
RETURN RAW;
```

  表 2-262 DBE_RAW.BIT_COMPLEMENT 接口参数说明

参数	描述
r	进行按位取反的RAW类型值，最大长度不超过32767。

- DBE_RAW.BIT_XOR

  BIT_XOR求两个RAW类型值按位异或的结果。

  DBE_RAW.BIT_XOR原型为：

```
DBE_RAW.BIT_XOR(
   r1 IN RAW,
   r2 IN RAW)
RETURN RAW;
```

  表 2-263 DBE_RAW.BIT_XOR 接口参数说明

参数	描述
r1	与r2进行按位异或的RAW类型值，最大长度不超过32767。
r2	与r1进行按位异或的RAW类型值，最大长度不超过32767。

- DBE_RAW.CAST_FROM_BINARY_DOUBLE_TO_RAW

  CAST_FROM_BINARY_DOUBLE_TO_RAW将BINARY_DOUBLE类型值转换为RAW类型。

  DBE_RAW.CAST_FROM_BINARY_DOUBLE_TO_RAW原型为：

```
DBE_RAW.CAST_FROM_BINARY_DOUBLE_TO_RAW (
   n        IN BINARY_DOUBLE,
```


版权所有 © 华为云计算技术有限公司

```
  endianess IN INTEGER DEFAULT 1)
RETURN RAW;
```

表 2-264 DBE_RAW.CAST_FROM_BINARY_DOUBLE_TO_RAW 接口参数说明

参数	描述
n	需要进行转换的BINARY_DOUBLE类型值。
endianess	字节序，取值范围：整型值：1、2或3。 ● 1表示BIG_ENDIAN。默认值为BIG_ENDIAN。 ● 2表示LITTLE_ENDIAN。 ● 3表示MACHINE_ENDIAN。 如果机器默认为BIG_ENDIAN，则函数执行结果MACHINE_ENDIAN和BIG_ENDIAN一致，如果机器默认为LITTLE_ENDIAN，则函数执行结果MACHINE_ENDIAN和LITTLE_ENDIAN一致。

- DBE_RAW.CAST_FROM_RAW_TO_BINARY_DOUBLE

  CAST_FROM_RAW_TO_BINARY_DOUBLE将RAW类型值转换为BINARY_DOUBLE类型。

  DBE_RAW.CAST_FROM_RAW_TO_BINARY_DOUBLE原型为：

  ```
  DBE_RAW.CAST_FROM_RAW_TO_BINARY_DOUBLE(
    r         IN RAW,
    endianess IN INTEGER DEFAULT 1)
  RETURN BINARY_DOUBLE;
  ```

  表 2-265 DBE_RAW.CAST_FROM_RAW_TO_BINARY_DOUBLE 接口参数说明

参数	描述
r	需要进行转换的RAW类型值，最短长度不低于8，最大长度不超过32767。
endianess	字节序，取值范围：整型值1、2或3。 ● 1表示BIG_ENDIAN。默认值为BIG_ENDIAN。 ● 2表示LITTLE_ENDIAN。 ● 3表示MACHINE_ENDIAN。 如果机器默认为BIG_ENDIAN，则函数执行结果MACHINE_ENDIAN和BIG_ENDIAN一致，如果机器默认为LITTLE_ENDIAN，则函数执行结果MACHINE_ENDIAN和LITTLE_ENDIAN一致。

- DBE_RAW.CAST_FROM_RAW_TO_BINARY_FLOAT

  CAST_FROM_RAW_TO_BINARY_FLOAT将RAW类型值转换为FLOAT4类型。

  DBE_RAW.CAST_FROM_RAW_TO_BINARY_FLOAT原型为：

  ```
  DBE_RAW.CAST_FROM_RAW_TO_BINARY_FLOAT(
    r         IN RAW,
    endianess IN INTEGER DEFAULT 1)
  RETURN FLOAT4;
  ```


版权所有 © 华为云计算技术有限公司

表 2-266 DBE_RAW.CAST_FROM_RAW_TO_BINARY_FLOAT 接口参数说明

参数	描述
r	需要进行转换的RAW类型值，最短长度不低于4，最大长度不超过32767。
endianess	字节序，取值范围：整型值1、2或3。 ● 1表示BIG_ENDIAN。默认值为BIG_ENDIAN。 ● 2表示LITTLE_ENDIAN。 ● 3表示MACHINE_ENDIAN。 如果机器默认为BIG_ENDIAN，则函数执行结果MACHINE_ENDIAN和BIG_ENDIAN一致，如果机器默认为LITTLE_ENDIAN，则函数执行结果MACHINE_ENDIAN和LITTLE_ENDIAN一致。

- DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW

  CAST_FROM_BINARY_FLOAT_TO_RAW将FLOAT4类型值转换为RAW类型。

  说明

  DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW在将FLOAT4类型值转换为RAW类型之前，会先对输入值n进行范围和精度截断处理，具体规则如下：
  - 当n大于FLOAT4的最大值3.40282E+38时，按3.40282E+38处理。
  - 当n小于FLOAT4的最小值-3.40282E+38时，按-3.40282E+38处理。
  - 当n满足-1.4013E-45 < n < 1.4013E-45时，按0处理。

  最终的转换结果基于截断后的值生成。

  DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW原型为：

```
DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW(
   n         IN FLOAT4,
   endianess IN INTEGER DEFAULT 1)
RETURN RAW;
```

表 2-267 DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW 接口参数说明

参数	描述
n	需要进行转换的FLOAT4类型值。
endianess	字节序，取值范围：整型值1、2或3。 ● 1表示BIG_ENDIAN。默认值为BIG_ENDIAN。 ● 2表示LITTLE_ENDIAN。 ● 3表示MACHINE_ENDIAN。 如果机器默认为BIG_ENDIAN，则函数执行结果MACHINE_ENDIAN和BIG_ENDIAN一致，如果机器默认为LITTLE_ENDIAN，则函数执行结果MACHINE_ENDIAN和LITTLE_ENDIAN一致。

- DBE_RAW.CAST_FROM_RAW_TO_NUMBER

  CAST_FROM_RAW_TO_NUMBER将RAW类型值转换为NUMERIC类型。

版权所有 © 华为云计算技术有限公司

说明

由于ORA数据库与GaussDB的NUMBER数据类型底层实现不同，而RAW类型为数据的底层实现的二进制流的十六进制表示，故该函数在ORA数据库与GaussDB的表现不同。用户不能将ORA数据库中的一个NUMBER类型数据对应的RAW类型在GaussDB中得到相同的NUMBER类型数据，在GaussDB中的表现可参考**示例**。

DBE_RAW.CAST_FROM_RAW_TO_NUMBER原型为：

```
DBE_RAW.CAST_FROM_RAW_TO_NUMBER(
    r IN RAW)
RETURN NUMERIC;
```

表 2-268 DBE_RAW.CAST_FROM_RAW_TO_NUMBER 接口参数说明

参数	描述
r	需要进行转换的RAW类型值，最短长度不低于4，最大长度不超过32767。

- DBE_RAW.CAST_FROM_NUMBER_TO_RAW

  CAST_FROM_NUMBER_TO_RAW将NUMERIC类型值转换为RAW类型。

  说明

  由于ORA数据库与GaussDB的NUMBER数据类型底层实现不同，而RAW类型为数据的底层实现的二进制流的十六进制表示，故该函数在ORA数据库与GaussDB的表现不同。用户不能将ORA数据库的NUMBER类型数据的RAW类型表示在GaussDB数据库中还原成ORA原NUMBER类型数值，该函数在GaussDB中的表现可参考**示例**。

  DBE_RAW.CAST_FROM_NUMBER_TO_RAW原型为：

  ```
  DBE_RAW.CAST_FROM_NUMBER_TO_RAW(
      value IN NUMERIC)
  RETURN RAW;
  ```

  表 2-269 DBE_RAW.CAST_FROM_NUMBER_TO_RAW 接口参数说明

参数	描述
value	需要进行转换的NUMERIC类型。

- DBE_RAW.CAST_FROM_RAW_TO_NVARCHAR2

  CAST_FROM_RAW_TO_NVARCHAR2将RAW类型值转换为NVARCHAR2类型。

  DBE_RAW.CAST_FROM_RAW_TO_NVARCHAR2原型为：

  ```
  DBE_RAW.CAST_FROM_RAW_TO_NVARCHAR2(
      r IN RAW)
  RETURN NVARCHAR2;
  ```

  表 2-270 DBE_RAW.CAST_FROM_RAW_TO_NVARCHAR2 接口参数说明

参数	描述
r	需要进行转换的RAW类型值，最大长度不超过32767。

- DBE_RAW.COMPARE

版权所有 © 华为云计算技术有限公司

COMPARE返回两个RAW类型值首个不相同的位置。

DBE_RAW.COMPARE原型为：

```
DBE_RAW.COMPARE(
    r1  IN RAW,
    r2  IN RAW,
    pad IN RAW DEFAULT NULL)
RETURN INTEGER;
```

表 2-271 DBE_RAW.COMPARE 接口参数说明

参数	描述
r1	要比较的第一个数据，可以为NULL或长度为0，最大长度不超过32767。
r2	要比较的第二个数据，可以为NULL或长度为0，最大长度不超过32767。
pad	取pad的首字节，用于扩展r1或r2中较短的一个，最大长度不超过32767。当该参数为NULL、长度为0或缺省时，扩展值为0x00。

- DBE_RAW.CONCAT

  CONCAT将最多12个RAW类型值连接到一个新的RAW类型值中，并返回。如果连接后的长度超过32767，则会报错。

  DBE_RAW.CONCAT原型为：

```
DBE_RAW.CONCAT(
    r1  IN RAW DEFAULT NULL,
    r2  IN RAW DEFAULT NULL,
    r3  IN RAW DEFAULT NULL,
    r4  IN RAW DEFAULT NULL,
    r5  IN RAW DEFAULT NULL,
    r6  IN RAW DEFAULT NULL,
    r7  IN RAW DEFAULT NULL,
    r8  IN RAW DEFAULT NULL,
    r9  IN RAW DEFAULT NULL,
    r10 IN RAW DEFAULT NULL,
    r11 IN RAW DEFAULT NULL,
    r12 IN RAW DEFAULT NULL)
RETURN RAW;
```

表 2-272 DBE_RAW.CONCAT 接口参数说明

参数	描述
r1...r12	需要进行连接的RAW类型值。

- DBE_RAW.CONVERT

  CONVERT将一个RAW类型值从原编码方式from_charset转换为目标编码方式to_charset。

  如果原编码格式到目标编码格式的转化规则不存在，则参数r不进行任何转换直接返回，如GBK和LATIN1之间的转换规则是不存在的，具体转换规则可以通过查看系统表**PG_CONVERSION**获得。

  DBE_RAW.CONVERT原型为：

```
DBE_RAW.CONVERT(
    r           IN RAW,
```


版权所有 © 华为云计算技术有限公司

```
    to_charset   IN NAME,
    from_charset IN NAME)
RETURN RAW;
```

表 2-273 DBE_RAW.CONVERT 接口参数说明

参数	描述
r	需要进行转换的RAW类型值，最大长度不超过32767。
to_charset	目标编码字符集名称。
from_charset	原编码字符集名称，在该编码下，r必须是合法的。

- DBE_RAW.COPIES

  COPIES将一个RAW类型值复制n次，串联在一起，并返回连接后的结果。如果复制后的长度超过32767，则会报错。

  DBE_RAW.COPIES原型为：

```
DBE_RAW.COPIES(
    r IN RAW,
    n IN NUMERIC)
RETURN RAW;
```

表 2-274 DBE_RAW.COPIES 接口参数说明

参数	描述
r	需要进行复制的RAW类型值。
n	复制的次数（必须为正数，如果是小数会进行四舍五入）。

- DBE_RAW.OVERLAY

  OVERLAY用一个RAW类型值对另一个RAW类型值进行覆盖，可以指定起始覆盖位置和覆盖长度。

  DBE_RAW.OVERLAY原型为：

```
DBE_RAW.OVERLAY(
    overlay_str IN RAW,
    target      IN RAW,
    pos         IN BINARY_INTEGER DEFAULT 1,
    len         IN BINARY_INTEGER DEFAULT NULL,
    pad         IN RAW            DEFAULT NULL)
RETURN RAW;
```

表 2-275 DBE_RAW.OVERLAY 接口参数说明

参数	描述
overlay_str	用于覆盖的字节，不能为NULL。
target	被覆盖的字节串，RAW类型，长度可超过32767字节，但受pos和len参数限制，覆盖操作只能发生在其前32767个字节内，不能为NULL。

版权所有 © 华为云计算技术有限公司

参数	描述
pos	从第几个字节开始覆盖，第一个字节位置为1，限制pos>=1且len+(pos-1)<=32767，缺省值为1。
len	要覆盖的长度，限制len>=0且len+(pos-1)<=32767，缺省值为覆盖字节（overlay_str）的长度。
pad	填充字节，只有第一个字节有效，缺省值为NULL，为NULL时默认为0x00。

- DBE_RAW.REVERSE

  REVERSE将RAW类型值按字节进行翻转。

  DBE_RAW.REVERSE原型为：

```
DBE_RAW.REVERSE(
    r IN RAW
)
RETURN RAW;
```

  表 2-276 DBE_RAW.REVERSE 接口参数说明

参数	描述
r	需要进行翻转的RAW类型值，最大长度不超过32767字节，如果为NULL则返回NULL。

- DBE_RAW.TRANSLATE

  TRANSLATE将转换或舍弃RAW类型值的指定字节。

  DBE_RAW.TRANSLATE原型为：

```
DBE_RAW.TRANSLATE(
    r        IN RAW,
    from_set IN RAW,
    to_set   IN RAW)
RETURN RAW;
```

  表 2-277 DBE_RAW.TRANSLATE 接口参数说明

参数	描述
r	要转换的原字节串，RAW类型，最大长度不超过32767字节，不能为NULL。
from_set	原字节串中要转换的字节码，RAW类型，不能为NULL。原字节串中存在于from_set中的字节都将被替换成to_set中对应位置的字节，假如from_set中有多个相同的字节，只会替换成第一个该字节对应的字节。比如：r[x]=from_set[n]，那么r[x]将被替换成to_set[n]；如果from_set[n]对应的to_set[n]不存在（也就是to_set字节数不超过n）时，r[x]将被舍弃。
to_set	对应from_ set字节转换为的字节码，RAW类型，不能为NULL。

- DBE_RAW.TRANSLITERATE

版权所有 © 华为云计算技术有限公司

TRANSLITERATE将RAW类型值转换为NUMERIC类型。

DBE_RAW.TRANSLITERATE原型为：

```
DBE_RAW.TRANSLITERATE(
   r        IN RAW,
   to_set   IN RAW DEFAULT NULL,
   from_set IN RAW DEFAULT NULL,
   pad      IN RAW DEFAULT NULL)
RETURN NUMERIC;
```

表 2-278 DBE_RAW.TRANSLITERATE 接口参数说明

参数	描述
r	要转换的字节串，RAW类型，最大长度不超过32767字节，不能为NULL。
to_set	对应from_ set字节转换为的字节码，RAW类型，缺省值为NULL。如果为NULL，则r中的所有存在于from_set中的字节都将被转换成pad。
from_set	字节串r中要转换的字节码，RAW类型，缺省值为NULL。如果from_set为NULL，则字节串r的所有字节将被转换成由pad填充的等长数据。否则字节串r中存在于from_set中的字节都将被替换成to_set中对应位置的字节，比如r[x]=from_set[n]，那么r[x]将被转换成to_set[n]，如果to_set[n]不存在，r[x]将被转换成pad。
pad	填充字节，只有第一个字节有效，缺省值为NULL，为NULL时默认为0x00。

- DBE_RAW.XRANGE

  XRANGE返回包含以指定字节码开始和结束的连续一字节编码的RAW类型值。

  DBE_RAW.XRANGE原型为：

```
DBE_RAW.XRANGE(
   start_byte IN RAW,
   end_byte   IN RAW)
RETURN RAW;
```

表 2-279 DBE_RAW.XRANGE 接口参数说明

参数	描述
start_byte	开始字节，只有第一个字节有效，如果为NULL则默认为0x00。
end_byte	结束字节，只有第一个字节有效，如果为NULL则会默认为0xFF。如果end_byte<start_byte，函数会触发循环回绕机制：从起始字节start_byte开始，一直递增到十六进制最大值FF，然后从00继续递增，直到达到结束字节end_byte为止。

示例

```
-- 在存储过程中操作RAW数据。
CREATE OR REPLACE PROCEDURE proc_raw
AS
```


版权所有 © 华为云计算技术有限公司

```
str varchar2(100) := 'abcdef';
source raw(100);
amount integer;
BEGIN
source := dbe_raw.cast_from_varchar2_to_raw(str); -- 类型转换。
amount := dbe_raw.get_length(source); -- 获取长度。
dbe_output.print_line(amount);
END;
/
CREATE PROCEDURE

-- 调用存储过程。
CALL proc_raw();
6
 proc_raw
----------

(1 row)
-- 删除存储过程。
DROP PROCEDURE proc_raw;
DROP PROCEDURE

DECLARE
v_raw RAW;
v_double BINARY_DOUBLE;
v_float FLOAT4;
v_numeric NUMERIC;
v_nvarchar2 NVARCHAR2;
BEGIN
-- RAW类型按位与。
SELECT DBE_RAW.BIT_AND('AFF', 'FF0B') INTO v_raw; -- 0A0B
-- RAW类型按位取反。
SELECT DBE_RAW.BIT_COMPLEMENT('0AFF') INTO v_raw; -- F500
-- RAW类型按位异或。
SELECT DBE_RAW.BIT_XOR('AFF', 'FF0B') INTO v_raw; -- F5F4
-- BINARY_DOUBLE类型值转RAW类型。
SELECT DBE_RAW.CAST_FROM_BINARY_DOUBLE_TO_RAW(1.0001,1) INTO v_raw; -- 3FF00068DB8BAC71
-- RAW类型值转BINARY_DOUBLE类型。
SELECT DBE_RAW.CAST_FROM_RAW_TO_BINARY_DOUBLE('3FF00068DB8BAC71',1) INTO v_double; --
1.0001
-- RAW类型转FLOAT4类型。
SELECT DBE_RAW.CAST_FROM_RAW_TO_BINARY_FLOAT('40200000',1) INTO v_float; -- 2.5
-- FLOAT4类型转RAW类型。
SELECT DBE_RAW.CAST_FROM_BINARY_FLOAT_TO_RAW('2.5',1) INTO v_raw; -- 40200000
-- RAW类型转NUMERIC类型。
SELECT DBE_RAW.CAST_FROM_RAW_TO_NUMBER('808002008813') INTO v_numeric; -- 2.5
-- NUMERIC类型转RAW类型。
SELECT DBE_RAW.CAST_FROM_NUMBER_TO_RAW('2.5') INTO v_raw; -- 808002008813
-- RAW类型转NVARCHAR2类型。
SELECT DBE_RAW.CAST_FROM_RAW_TO_NVARCHAR2('12345678') INTO v_nvarchar2; -- \x124Vx
-- RAW类型COMPARE。
SELECT DBE_RAW.COMPARE('ABCD','AB') INTO v_numeric; -- 2
-- RAW类型CONCAT。
SELECT DBE_RAW.CONCAT('ABCD','AB') INTO v_raw; -- ABCDAB
-- RAW类型CONVERT。
SELECT DBE_RAW.CONVERT('E695B0', 'GBK','UTF8') INTO v_raw; -- CAFD
-- RAW类型COPIES。
SELECT DBE_RAW.COPIES('ABCD',2) INTO v_raw; -- ABCDABCD
-- RAW类型指定位置和长度进行覆盖。
SELECT DBE_RAW.OVERLAY('abcef', '12345678123456', 2, 5, '9966') INTO v_raw; -- 120ABCEF999956
-- RAW类型按字节翻转。
SELECT DBE_RAW.REVERSE('12345678') INTO v_raw; -- 78563412
-- RAW类型字节转换（无填充码）。
SELECT DBE_RAW.TRANSLATE('1122112233', '1133','55') INTO v_raw; -- 55225522
-- RAW类型字节转换（有填充码）。
SELECT DBE_RAW.TRANSLITERATE('1122112233', '55','1133','FFEE') INTO v_raw; -- 55225522FF
-- RAW类型两个字节间的所有字节。
SELECT DBE_RAW.XRANGE('00','03') INTO v_raw; -- 00010203
SELECT DBE_RAW.XRANGE('FA'::raw, '01'::raw); -- FAFBFCFDFEFF0001
```


版权所有 © 华为云计算技术有限公司

```
END;
/
ANONYMOUS BLOCK EXECUTE
```

2.12.2.15 DBE_SCHEDULER

接口介绍

高级功能包DBE_SCHEDULER支持通过调度（schedule）和程序（program）更加灵活地创建定时任务。支持的接口请参见表2-280。

说明

- DBE_SCHEDULER支持CN节点间同步定时任务，不支持不同CN节点间创建同名任务。
- 由于505.0.0之前版本未实现同步功能，所有同步接口都只针对505.0.0及之后版本的数据进行同步操作。
- 升级前会对旧版本CN是否存在同名job进行检查，如果旧版本CN间同名数据库下存在名称相同、且job创建节点不相同的job，报错提示用户删除job后尝试重新升级。

表 2-280 DBE_SCHEDULER

接口名称	描述
DBE_SCHEDULER.CREATE_JOB	创建定时任务。
DBE_SCHEDULER.DROP_JOB	删除定时任务。
DBE_SCHEDULER.DROP_SINGLE_JOB	删除单个定时任务。
DBE_SCHEDULER.SET_ATTRIBUTE	设置对象属性。
DBE_SCHEDULER.RUN_JOB	运行定时任务。
DBE_SCHEDULER.RUN_BACKEND_JOB	后台运行定时任务。
DBE_SCHEDULER.RUN_FOREGROUND_JOB	前台运行定时任务。
DBE_SCHEDULER.STOP_JOB	停止定时任务。
DBE_SCHEDULER.STOP_SINGLE_JOB	停止单个定时任务。
DBE_SCHEDULER.GENERATE_JOB_NAME	生成定时任务名。
DBE_SCHEDULER.CREATE_PROGRAM	创建程序。
DBE_SCHEDULER.DEFINE_PROGRAM_ARGUMENT	定义程序参数。
DBE_SCHEDULER.DROP_PROGRAM	删除程序。
DBE_SCHEDULER.DROP_SINGLE_PROGRAM	删除单个程序。
DBE_SCHEDULER.SET_JOB_ARGUMENT_VALUE	设置定时任务参数值。
DBE_SCHEDULER.CREATE_SCHEDULE	创建调度。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_SCHEDULER.DROP_SCHEDULE	删除调度。
DBE_SCHEDULER.DROP_SINGLE_SCHEDULE	删除单个调度。
DBE_SCHEDULER.CREATE_JOB_CLASS	创建定时任务类。
DBE_SCHEDULER.DROP_JOB_CLASS	删除定时任务类。
DBE_SCHEDULER.DROP_SINGLE_JOB_CLASS	删除单个定时任务类。
DBE_SCHEDULER.GRANT_USER_AUTHORIZATION	赋予用户特殊权限。
DBE_SCHEDULER.REVOKE_USER_AUTHORIZATION	撤销用户特殊权限。
DBE_SCHEDULER.CREATE_CREDENTIAL	创建证书。
DBE_SCHEDULER.DROP_CREDENTIAL	销毁证书。
DBE_SCHEDULER.ENABLE	启用对象。
DBE_SCHEDULER.ENABLE_SINGLE	启用单个对象。
DBE_SCHEDULER.DISABLE	停用对象。
DBE_SCHEDULER.DISABLE_SINGLE	停用单个对象。
DBE_SCHEDULER.EVAL_CALENDAR_STRING	分析调度任务周期。
DBE_SCHEDULER.EVALUATE_CALENDAR_STRING	分析调度任务周期。

- DBE_SCHEDULER.CREATE_JOB
  创建一个定时任务，同步任务到所有CN节点。
  DBE_SCHEDULER.CREATE_JOB原型可以分为4种：

```
-- 内联调度和程序的定时任务
DBE_SCHEDULER.CREATE_JOB(
job_name TEXT,
job_type TEXT,
job_action TEXT,
number_of_arguments INTEGER             DEFAULT 0,
start_date TIMESTAMP WITH TIME ZONE     DEFAULT NULL,
repeat_interval TEXT                    DEFAULT NULL,
end_date TIMESTAMP WITH TIME ZONE       DEFAULT NULL,
job_class TEXT            DEFAULT 'DEFAULT_JOB_CLASS',
enabled BOOLEAN                         DEFAULT FALSE,
auto_drop BOOLEAN                       DEFAULT TRUE,
comments TEXT                           DEFAULT NULL,
credential_name TEXT                    DEFAULT NULL,
destination_name TEXT                   DEFAULT NULL
)

-- 引用创建好的调度和程序的定时任务
```


版权所有 © 华为云计算技术有限公司

```
DBE_SCHEDULER.CREATE_JOB(
job_name TEXT,
program_name TEXT,
schedule_name TEXT,
job_class TEXT            DEFAULT 'DEFAULT_JOB_CLASS',
enabled BOOLEAN                DEFAULT FALSE,
auto_drop BOOLEAN               DEFAULT TRUE,
comments TEXT                  DEFAULT NULL,
job_style TEXT                DEFAULT 'REGULAR',
credential_name TEXT            DEFAULT NULL,
destination_name TEXT           DEFAULT NULL
)

-- 引用创建好的程序，内联调度的定时任务
DBE_SCHEDULER.CREATE_JOB(
job_name text,
program_name TEXT,
start_date TIMESTAMP WITH TIME ZONE     DEFAULT NULL,
repeat_interval TEXT              DEFAULT NULL,
end_date TIMESTAMP WITH TIME ZONE       DEFAULT NULL,
job_class TEXT            DEFAULT 'DEFAULT_JOB_CLASS',
enabled BOOLEAN                DEFAULT FALSE,
auto_drop BOOLEAN               DEFAULT TRUE,
comments TEXT                  DEFAULT NULL,
job_style TEXT                DEFAULT 'REGULAR',
credential_name TEXT            DEFAULT NULL,
destination_name TEXT           DEFAULT NULL
)

-- 引用创建好的调度，内联程序的定时任务
DBE_SCHEDULER.CREATE_JOB(
job_name TEXT,
schedule_name TEXT,
job_type TEXT,
job_action TEXT,
number_of_arguments INTEGER             DEFAULT 0,
job_class TEXT            DEFAULT 'DEFAULT_JOB_CLASS',
enabled BOOLEAN                DEFAULT FALSE,
auto_drop BOOLEAN               DEFAULT TRUE,
comments TEXT                  DEFAULT NULL,
credential_name TEXT            DEFAULT NULL,
destination_name TEXT           DEFAULT NULL
)
```

说明

- 利用DBE_SCHEDULER创建的定时任务不会与DBE_TASK中的定时任务相冲突。
- DBE_SCHEDULER创建的定时任务会生成对应的job_id，但是在使用过程中这个id并没有实际意义。
- 对于CREATE类型接口，不做入参类型合法性校验，创建成功不代表会执行成功，可通过系统表**PG_JOB**查询当前任务的执行状态。
- 普通用户因缺乏定时任务权限，首次调用该函数时会失败，需通过DBE_SCHEDULER.GRANT_USER_AUTHORIZATION授予定时任务权限。
- 当job_type取值为'PLSQL_BLOCK'时，不接受程序参数，number_of_arguments不允许传入大于0的值，否则创建定时任务会失败。
- 若start_date大于end_date，任务可创建成功且无报错，但无法触发调度。

版权所有 © 华为云计算技术有限公司

表 2-281 DBE_SCHEDULER.CREATE_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务名称。非ilm特性调度场景下，无法创建前缀为"ilmadowind$_"和"ilmjob$_"的job。
job_type	TEXT	IN	否	定时任务内联程序类型，可用类型为： ● 'PLSQL_BLOCK'：匿名存储过程块。 ● 'STORED_PROCEDURE'：保存的存储过程。 ● 'EXTERNAL_SCRIPT'：外部脚本。
job_action	TEXT	IN	否	定时任务内联程序执行内容。
number_of_arguments	INTEGER	IN	否	定时任务内联程序参数个数。取值范围为0 ~ 255。
program_name	TEXT	IN	否	定时任务引用程序名称。
start_date	TIMESTAMP WITH TIME ZONE	IN	是	定时任务内联调度起始时间。
repeat_interval	TEXT	IN	是	定时任务内联调度任务周期。
end_date	TIMESTAMP WITH TIME ZONE	IN	是	定时任务内联调度失效时间。
schedule_name	TEXT	IN	否	定时任务引用调度名称。
job_class	TEXT	IN	否	定时任务类名。
enabled	BOOLEAN	IN	否	定时任务启用状态。
auto_drop	BOOLEAN	IN	否	定时任务自动删除。
comments	TEXT	IN	是	备注。
job_style	TEXT	IN	否	定时任务行为模式，仅支持'REGULAR'。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
credential_name	TEXT	IN	是	定时任务证书名。
destination_name	TEXT	IN	是	定时任务目标名，当前只支持值'CCN'（不区分大小写），指定destination_name为CCN的任务只在central coordinator执行。否则按原有逻辑处理。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'select
pg_sleep(1);', 3, false, 'test');
 create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule1', NULL, 'sysdate', NULL, 'test');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_job(job_name=>'job1', program_name=>'program1',
schedule_name=>'schedule1');
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule1');
 drop_schedule
---------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

说明

创建'EXTERNAL_SCRIPT'类型的定时任务需要管理员赋予相关的权限和证书，且需要数据库启动用户对该外部脚本有读取权限才可以正常生效。

- DBE_SCHEDULER.DROP_JOB

  删除定时任务。仅任务被同步时同步操作。

  DBE_SCHEDULER.DROP_JOB函数原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_SCHEDULER.drop_job(
job_name text,
force boolean                    default false,
defer boolean                    default false,
commit_semantics text      default 'STOP_ON_FIRST_ERROR'
)
```

说明

DBE_SCHEDULER.DROP_JOB可以指定一个或多个任务，也可以指定任务类进行定时任务删除。

表 2-282 DBE_SCHEDULER.DROP_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务或定时任务类名称，可以指定一个或多个，当指定多个定时任务时需要利用逗号隔开。
force	BOOLEAN	IN	否	删除定时任务行为标记位： ● true：尝试先停止当前定时任务，再进行删除。 ● false：如果定时任务正在运行会删除失败。
defer	BOOLEAN	IN	否	删除定时任务行为标记位： ● true：允许定时任务完成后再进行删除。 ● false：不允许定时任务继续执行，尝试进行删除。
commit_semantics	TEXT	IN	否	提交规则： ● 'STOP_ON_FIRST_ERROR'：在第一个报错之前的删除操作会提交。 ● 'TRANSACTIONAL'：事务级提交，报错前的删除操作会回滚。 ● 'ABSORB_ERRORS'：尝试越过报错，将成功的删除操作提交。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 3, false, 'test');
 create_program
----------------


(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule1', NULL, 'sysdate', NULL, 'test');
 create_schedule
-----------------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_job(job_name=>'job1', program_name=>'program1',
schedule_name=>'schedule1');
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule1');
 drop_schedule
---------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

说明

commit_semantic中的'TRANSACTIONAL'选项必须在force为false的情况下才会生效。

- DBE_SCHEDULER.DROP_SINGLE_JOB

  删除一个定时任务。仅任务被同步时同步操作。

  DBE_SCHEDULER.DROP_SINGLE_JOB原型为：

```
DBE_SCHEDULER.drop_single_job(
job_name text,
force boolean                   default false,
defer boolean                   default false
)
```

表 2-283 DBE_SCHEDULER.DROP_SINGLE_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务或定时任务类名称。
force	BOOLEAN	IN	否	删除定时任务行为标记位： ● true：尝试先停止当前定时任务，再进行删除。 ● false：如果定时任务正在运行会删除失败。
defer	BOOLEAN	IN	否	删除定时任务行为标记位： ● true：允许定时任务完成后再进行删除。 ● false：不允许定时任务继续执行，尝试进行删除。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 0, false, 'test');
 create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_job('job1', 'program1', '2021-07-20', 'interval ''3 minute''',
'2121-07-20', 'DEFAULT_JOB_CLASS', false, false,'test', 'style', NULL, NULL);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_single_job('job1', false, false);
 drop_single_job
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

- DBE_SCHEDULER.SET_ATTRIBUTE

  修改定时任务属性。仅任务被同步时同步操作。

  DBE_SCHEDULER.SET_ATTRIBUTE 5种原型为：

```
DBE_SCHEDULER.set_attribute(
name                text,
attribute           text,
value               boolean
)

DBE_SCHEDULER.set_attribute(
name                text,
attribute           text,
value               datea
)

DBE_SCHEDULER.set_attribute(
name                text,
attribute           text,
value               timestamp
)

DBE_SCHEDULER.set_attribute(
name                text,
attribute           text,
value               timestamp with time zone
)

DBE_SCHEDULER.set_attribute(
name text,
attribute text,
value text,
value2 text                     default NULL
)
```


版权所有 © 华为云计算技术有限公司

说明

- name在这里可以指定任何DBE_SCHEDULER内部的对象。
- 必须将mapping_date_to_datea设置为on才能正常使用datea作为入参的DBE_SCHEDULER.set_attribute函数。

表 2-284 DBE_SCHEDULER.SET_ATTRIBUTE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
name	TEXT	IN	否	对象名。
attribute	TEXT	IN	否	属性名。
value	BOOLEAN/DATE/TIMESTAMP/TIMESTAMP WITH TIME ZONE/TEXT	IN	否	属性值，可选属性如下： ● 定时任务相关：job_type、job_action、number_of_arguments、start_date、repeat_interval、end_date、job_class、enabled、auto_drop、comments、credential_name、destination_name、program_name、schedule_name和job_style。 ● 程序相关：program_action、program_type、number_of_arguments和comments。 ● 调度相关：start_date、repeat_interval、end_date和comments。
value2	TEXT	IN	是	额外属性值。保留参数位，目前尚不支持拥有额外属性值的目标属性。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 3, false, 'test');
 create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.set_attribute('program1', 'number_of_arguments', 0);
 set_attribute
---------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.set_attribute('program1', 'program_type', 'STORED_PROCEDURE');
 set_attribute
---------------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

说明

- 不建议使用DBE_SCHEDULER.SET_ATTRIBUTE来将参数置空。
- 对象名不能通过DBE_SCHEDULER.SET_ATTRIBUTE来更改。
- 内联对象不能通过DBE_SCHEDULER.SET_ATTRIBUTE来更改。

- DBE_SCHEDULER.RUN_JOB

  运行定时任务。

  DBE_SCHEDULER.RUN_JOB原型为：

```
DBE_SCHEDULER.run_job(
job_name text,
use_current_session boolean             default true
)
```

说明

DBE_SCHEDULER.RUN_JOB主要用于立即运行定时作业，独立于定时任务本身的调度，甚至可以同时运行。

表 2-285 DBE_SCHEDULER.RUN_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务名称，可以指定一个或多个，当指定多个定时任务时需要利用逗号隔开。
use_current_session	BOOLEAN	IN	否	运行定时任务标志位： • true：使用当前会话运行，主要用于查看定时任务是否可以正常运行。 • false：后台拉起定时任务，运行结果会打印到日志中。

示例：

```
gaussdb=# SELECT DBE_SCHEDULER.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.run_job('job1', false);
 run_job
---------

(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

说明

use_current_session目前只作用于job_type为'EXTERNAL_SCRIPT'的定时任务上。

- DBE_SCHEDULER.RUN_BACKEND_JOB

  后台运行定时任务。

  DBE_SCHEDULER.RUN_BACKEND_JOB原型为：

```
DBE_SCHEDULER.run_backend_job(
job_name text
)
```

表 2-286 DBE_SCHEDULER.RUN_BACKEND_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	text	IN	否	定时任务名称。

示例：

```
gaussdb=# SELECT DBE_SCHEDULER.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.run_backend_job('job1');
 run_backend_job
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

- DBE_SCHEDULER.RUN_FOREGROUND_JOB

  当前会话运行定时任务。仅支持运行external类型任务。

  返回值：text

  DBE_SCHEDULER.RUN_FOREGROUND_JOB原型为：

```
DBE_SCHEDULER.run_foreground_job(
job_name text
)return text
```

表 2-287 DBE_SCHEDULER.RUN_FOREGROUND_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务名称。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CREATE USER test1 IDENTIFIED BY '*********';
NOTICE:  The encrypted password contains MD5 ciphertext, which is not secure.
CREATE ROLE
gaussdb=# SELECT DBE_SCHEDULER.create_credential('cre_1', 'test1', '*********');
 create_credential
-------------------

(1 row)

gaussdb=# SELECT DBE_SCHEDULER.create_job(job_name=>'job1', job_type=>'EXTERNAL_SCRIPT',
job_action=>'/usr/bin/pwd', enabled=>true, auto_drop=>false, credential_name => 'cre_1');
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.run_foreground_job('job1');
  run_foreground_job
---------------------------------
 Host key verification failed.\r+

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_credential('cre_1', false);
 drop_credential
-----------------

(1 row)

gaussdb=# DROP USER test1;
DROP ROLE
```

- DBE_SCHEDULER.STOP_JOB

  终止定时任务。

  DBE_SCHEDULER.STOP_JOB原型为：

```
DBE_SCHEDULER.stop_job(
job_name text,
force boolean                    default false,
commit_semantics text                default 'STOP_ON_FIRST_ERROR'
)
```

表 2-288 DBE_SCHEDULER.STOP_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务或定时任务类名称，可以指定一个或多个，当指定多个定时任务时需要利用逗号隔开。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
force	BOOLEAN	IN	否	删除定时任务行为标记位： ● true：调度器会发送终止信号立即结束任务线程。 ● false：调度器会尝试利用打断信号终止定时任务线程。
commit_semantics	TEXT	IN	否	提交规则： ● 'STOP_ON_FIRST_ERROR'：在第一个报错之前的打断操作会提交。 ● 'ABSORB_ERRORS'：尝试越过报错，将成功的打断操作提交。

示例：

```
gaussdb=# SELECT dbe_scheduler.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.stop_job('job1', true, 'STOP_ON_FIRST_ERROR');
 stop_job
----------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

- DBE_SCHEDULER.STOP_SINGLE_JOB

  终止单个定时任务。

  DBE_SCHEDULER.STOP_SINGLE_JOB原型为：

```
DBE_SCHEDULER.stop_single_job(
job_name text,
force boolean                    default false
)
```

表 2-289 DBE_SCHEDULER.STOP_SINGLE_JOB 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务或定时任务类名称

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
force	BOOLEAN	IN	否	删除定时任务行为标记位： ● true：调度器会发送终止信号立即结束任务线程。 ● false：调度器会尝试利用打断信号终止定时任务线程。

示例：

```
gaussdb=# SELECT dbe_scheduler.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.stop_single_job('job1', true);
 stop_single_job
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

- DBE_SCHEDULER.GENERATE_JOB_NAME

  生成定时任务名称。

  DBE_SCHEDULER.GENERATE_JOB_NAME原型为：

  ```
  DBE_SCHEDULER.generate_job_name(
  prefix text                         default 'JOB$_'
  )return text
  ```

  表 2-290 DBE_SCHEDULER.GENERATE_JOB_NAME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
prefix	TEXT	IN	否	生成名称的前缀，默认为'JOB$_'，反复执行生成的定时任务名为： job$_1, job$_2, job$_3 ...。

  示例：

  ```
  gaussdb=# CALL DBE_SCHEDULER.generate_job_name();
   generate_job_name
  -------------------
  ```


版权所有 © 华为云计算技术有限公司

```
 JOB$_1
(1 row)

gaussdb=# CALL DBE_SCHEDULER.generate_job_name();
 generate_job_name
-------------------
 JOB$_2
(1 row)

gaussdb=# CALL DBE_SCHEDULER.generate_job_name('job');
 generate_job_name
-------------------
 job3
(1 row)

gaussdb=# CALL DBE_SCHEDULER.generate_job_name('job');
 generate_job_name
-------------------
 job4
(1 row)
```

说明

首次执行DBE_SCHEDULER.GENERATE_JOB_NAME会在public下创建一个临时序列用于保存当前名称的序号。由于普通用户没有在public下的CREATE权限，因此如果普通用户在当前数据库下第一次调用该函数，会失败，需要授权该普通用户在public下的CREATE权限，或者使用有该权限的用户调用该接口以创建临时序列。

- DBE_SCHEDULER.CREATE_PROGRAM

  创建程序，同步到所有CN节点 。

  DBE_SCHEDULER.CREATE_PROGRAM原型为：

```
DBE_SCHEDULER.create_program(
program_name text,
program_type text,
program_action text,
number_of_arguments integer          default 0,
enabled boolean                    default false,
comments text                      default NULL
)
```

表 2-291 DBE_SCHEDULER.CREATE_PROGRAM 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
program_name	TEXT	IN	否	程序名称。
program_type	TEXT	IN	否	程序类型，可用类型为： ● 'PLSQL_BLOCK'：匿名存储过程块。 ● 'STORED_PROCEDURE'：保存的存储过程。 ● 'EXTERNAL_SCRIPT'：外部脚本。
program_action	TEXT	IN	否	程序操作。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
number_of_arguments	INTEGER	IN	否	程序采用的参数数量。取值范围为0 ~ 255。
enabled	BOOLEAN	IN	否	程序启用状态。
comments	TEXT	IN	是	备注。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 3, false, 'test');
 create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

- DBE_SCHEDULER.DEFINE_PROGRAM_ARGUMENT

  定义程序参数。仅程序被同步时同步操作。修正带默认值default_value字段的接口默认转换成小写行为，对字符大小写做出区分。

  DBE_SCHEDULER.DEFINE_PROGRAM_ARGUMENT原型为：

```
DBE_SCHEDULER.define_program_argument(
program_name text,
argument_position integer,
argument_name text                  default NULL,
argument_type text,
out_argument boolean                 default false
)

-- 带有默认值 --
DBE_SCHEDULER.define_program_argument(
program_name text,
argument_position integer,
argument_name text                  default NULL,
argument_type text,
default_value text,
out_argument boolean                 default false
)
```

表 2-292 DBE_SCHEDULER.DEFINE_PROGRAM_ARGUMENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
program_name	TEXT	IN	否	程序名称。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
argument_position	INTEGER	IN	否	参数位置。取值范围1 ~ number_of_arguments。若program的number_of_arguments为0，则调用此接口会报错。
argument_name	TEXT	IN	否	参数名称。
argument_type	TEXT	IN	否	参数类型。
default_value	TEXT	IN	否	默认值。
out_argument	BOOLEAN	IN	否	预留参数。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 2, false, 'test');
create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.define_program_argument('program1', 1, 'pa1', 'type1', false);
 define_program_argument
-------------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.define_program_argument('program1', 1, 'pa1', 'type1', 'value1',
false);
 define_program_argument
-------------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

- DBE_SCHEDULER.DROP_PROGRAM

  删除程序。仅程序被同步时同步操作。

  DBE_SCHEDULER.DROP_PROGRAM原型为：

```
DBE_SCHEDULER.drop_program(
program_name text,
force boolean                    default false
)
```


版权所有 © 华为云计算技术有限公司

表 2-293 DBE_SCHEDULER.DROP_PROGRAM 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
program_name	TEXT	IN	否	程序名称。
force	BOOLEAN	IN	否	删除程序行为标记位： ● true：在删除程序之前，将禁用应用该程序的所有作业。 ● false：该程序不能被任何作业引用，否则会发送错误。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 2, false, 'test');
create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

- DBE_SCHEDULER.DROP_SINGLE_PROGRAM

  删除单个程序。仅程序被同步时同步操作。

  DBE_SCHEDULER.DROP_SINGLE_PROGRAM原型为：

```
DBE_SCHEDULER.drop_single_program(
program_name text,
force boolean                    default false
)
```

表 2-294 DBE_SCHEDULER.DROP_SINGLE_PROGRAM 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
program_name	TEXT	IN	否	程序名称。
force	BOOLEAN	IN	否	删除程序行为标记位： ● true：在删除程序之前，将禁用应用该程序的所有作业。 ● false：该程序不能被任何作业引用，否则会发送错误。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'STORED_PROCEDURE', 'SELECT
pg_sleep(1);', 2, false, 'test');
create_program
----------------


(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_single_program('program1', false);
 drop_single_program
---------------------


(1 row)
```

- DBE_SCHEDULER.SET_JOB_ARGUMENT_VALUE

  设置定时任务参数值。argument_value字段支持赋空入参。仅任务被同步时同步操作。

  DBE_SCHEDULER.SET_JOB_ARGUMENT_VALUE原型为：

```
DBE_SCHEDULER.set_job_argument_value(
job_name text,
argument_position integer,
argument_value text
)

DBE_SCHEDULER.set_job_argument_value(
job_name text,
argument_name text,
argument_value text
)
```

表 2-295 DBE_SCHEDULER.SET_JOB_ARGUMENT_VALUE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_name	TEXT	IN	否	定时任务名称。
argument_position	INTEGER	IN	否	参数位置。取值范围1 ~ number_of_arguments。若关联的program的number_of_arguments为0，则调用此接口会报错。
argument_name	TEXT	IN	是	参数名称。
argument_value	TEXT	IN	是	参数值。

示例：

```
gaussdb=# CALL dbe_scheduler.create_job('job1','EXTERNAL_SCRIPT','BEGIN INSERT INTO test1
VALUES(12); end;',2,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# CALL DBE_SCHEDULER.set_job_argument_value('job1', 1, 'value1');
 set_job_argument_value
------------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

- DBE_SCHEDULER.CREATE_SCHEDULE

  创建调度。同步到所有CN节点。

  DBE_SCHEDULER.CREATE_SCHEDULE原型为：

```
DBE_SCHEDULER.create_schedule(
schedule_name text,
start_date timestamp with time zone      default NULL,
repeat_interval text,
end_date timestamp with time zone        default NULL,
comments text                            default NULL
)
```

表 2-296 DBE_SCHEDULER.CREATE_SCHEDULE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
schedule_name	TEXT	IN	否	调度名称。
start_date	TIMESTAMP WITH TIME ZONE	IN	是	调度开始时间。
repeat_interval	TEXT	IN	否	调度重复频率。
end_date	TIMESTAMP WITH TIME ZONE	IN	是	调度结束时间。
comments	TEXT	IN	是	备注。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule1', sysdate, 'sysdate + 3 / (24 * 60 * 60)',
null, 'test1');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule2', sysdate, 'FREQ=DAILY; BYHOUR=6;',
null, 'test1');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule3', sysdate, 'FREQ=DAILY; BYHOUR=6;');
 create_schedule
```


版权所有 © 华为云计算技术有限公司

```
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule1');
 drop_single_schedule
----------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule2', false);
 drop_single_schedule
----------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule3', true);
 drop_single_schedule
----------------------

(1 row)
```

- DBE_SCHEDULER.DROP_SCHEDULE

  删除调度。仅调度被同步时同步操作。

  DBE_SCHEDULER.DROP_SCHEDULE原型为：

```
DBE_SCHEDULER.drop_schedule(
schedule_name text,
force boolean                    default false
)
```

表 2-297 DBE_SCHEDULER.DROP_SCHEDULE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
schedule_name	TEXT	IN	否	调度名称。
force	BOOLEAN	IN	否	删除调度行为标记位： ● true：在删除调度之前，将禁用使用此调度的任何作业或窗口。 ● false：调度不能被任何作业或窗口引用，否则会发生错误。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule1', sysdate, 'sysdate + 3 / (24 * 60 * 60)',
null, 'test1');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule2', sysdate, 'FREQ=DAILY; BYHOUR=6;',
null, 'test1');
 create_schedule
-----------------

(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule3', sysdate, 'FREQ=DAILY; BYHOUR=6;');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule1');
 drop_single_schedule
----------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule2', false);
 drop_single_schedule
----------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_schedule('schedule3', true);
 drop_single_schedule
----------------------

(1 row)
```

- DBE_SCHEDULER.DROP_SINGLE_SCHEDULE

  删除单个调度。仅调度被同步时同步操作。

  DBE_SCHEDULER.DROP_SINGLE_SCHEDULE原型为：

```
DBE_SCHEDULER.drop_single_schedule(
schedule_name text,
force boolean                   default false
)
```

表 2-298 DBE_SCHEDULER.DROP_SINGLE_SCHEDULE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
schedule_name	TEXT	IN	否	调度名称。
force	BOOLEAN	IN	否	删除调度行为标记位： ● true：在删除调度之前，将禁用使用此调度的任何作业或窗口。 ● false：调度不能被任何作业或窗口引用，否则会发生错误。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule1', sysdate, 'sysdate + 3 / (24 * 60 * 60)',
null, 'test1');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule2', sysdate, 'FREQ=DAILY; BYHOUR=6;',
null, 'test1');
 create_schedule
```


版权所有 © 华为云计算技术有限公司

```
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_schedule('schedule3', sysdate, 'FREQ=DAILY; BYHOUR=6;');
 create_schedule
-----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_single_schedule('schedule1');
 drop_single_schedule
----------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_single_schedule('schedule2', false);
 drop_single_schedule
----------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_single_schedule('schedule3', true);
 drop_single_schedule
----------------------

(1 row)
```

- DBE_SCHEDULER.CREATE_JOB_CLASS

  创建定时任务类。同步到所有CN节点。

  DBE_SCHEDULER.CREATE_JOB_CLASS原型为：

```
DBE_SCHEDULER.create_job_class(
job_class_name text,
resource_consumer_group text            default NULL,
service text                          default NULL,
logging_level integer                   default 0,
log_history integer                   default NULL,
comments text                          default NULL
)
```

表 2-299 DBE_SCHEDULER.CREATE_JOB_CLASS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_class_name	TEXT	IN	否	定时任务类名称。
resource_consumer_group	TEXT	IN	是	定时任务类内联资源消费组。
service	TEXT	IN	是	定时任务类内联数据库服务。
logging_level	INTEGER	IN	否	定时任务类记录信息个数。
log_history	INTEGER	IN	是	定时任务类记录信息保留天数。
comments	TEXT	IN	是	备注。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_job_class(job_class_name => 'jc1', resource_consumer_group
=> '123');
 create_job_class
------------------


(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job_class('jc1', false);
 drop_job_class
----------------


(1 row)
```

- DBE_SCHEDULER.DROP_JOB_CLASS

  删除定时任务类。仅定时任务类被同步时同步操作。

  DBE_SCHEDULER.DROP_JOB_CLASS原型为：

  ```
  DBE_SCHEDULER.drop_job_class(
  job_class_name text,
  force boolean                    default false
  )
  ```

  表 2-300 DBE_SCHEDULER.DROP_JOB_CLASS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_class_name	TEXT	IN	否	定时任务类名称
force	BOOLEAN	IN	否	删除定时任务类行为标记位： ● true：该类的作业将被禁用，并且其他类将设置为默认类，只有在成功的情况下，才会删除该类。 ● false：被删除的类不得被任何作业引用，否则会发生错误。

  示例：

  ```
  gaussdb=# CALL DBE_SCHEDULER.create_job_class(job_class_name => 'jc1', resource_consumer_group
  => '123');
   create_job_class
  ------------------


  (1 row)

  gaussdb=# CALL DBE_SCHEDULER.drop_job_class('jc1', false);
   drop_job_class
  ----------------


  (1 row)
  ```

- DBE_SCHEDULER.DROP_SINGLE_JOB_CLASS

  删除单个定时任务类。仅定时任务类被同步时同步操作。

  DBE_SCHEDULER.DROP_SINGLE_JOB_CLASS原型为：

  ```
  DBE_SCHEDULER.drop_single_job_class(
  job_class_name text,
  ```


版权所有 © 华为云计算技术有限公司

```
force boolean                    default false
)
```

表 2-301 DBE_SCHEDULER.DROP_SINGLE_JOB_CLASS 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job_class_name	TEXT	IN	否	定时任务类名称。
force	BOOLEAN	IN	否	删除定时任务类行为标记位： ● true：该类的作业将被禁用，并且其他类将设置为默认类，只有在成功的情况下，才会删除该类。 ● false：被删除的类不得被任何作业引用，否则会发生错误。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_job_class(job_class_name => 'jc1', resource_consumer_group
=> '123');
 create_job_class
------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_single_job_class('jc1', false);
 drop_single_job_class
-----------------------

(1 row)
```

- DBE_SCHEDULER.GRANT_USER_AUTHORIZATION

  为数据库用户提供定时任务权限。调用该函数的用户需要具有SYSADMIN权限。同步到所有CN节点。

  DBE_SCHEDULER.GRANT_USER_AUTHORIZATION原型为：

```
DBE_SCHEDULER.grant_user_authorization(
username              text,
privilege             text
)
```

表 2-302 DBE_SCHEDULER.GRANT_USER_AUTHORIZATION 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
username	TEXT	IN	否	数据库用户名称。
privilege	TEXT	IN	否	定时任务权限。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CREATE USER user1 PASSWORD '1*s*****';
NOTICE:  The encrypted password contains MD5 ciphertext, which is not secure.
CREATE ROLE
gaussdb=# CALL DBE_SCHEDULER.grant_user_authorization('user1', 'create job');
 grant_user_authorization
--------------------------


(1 row)
gaussdb=# DROP USER user1;
DROP ROLE
```

- DBE_SCHEDULER.REVOKE_USER_AUTHORIZATION

  撤销数据库用户的定时任务权限。调用该函数的用户需要具有SYSADMIN权限。仅赋权同步时同步操作。

  DBE_SCHEDULER.REVOKE_USER_AUTHORIZATION原型为：

  ```
  DBE_SCHEDULER.revoke_user_authorization(
  username              text,
  privilege             text
  )
  ```

  表 2-303 DBE_SCHEDULER.REVOKE_USER_AUTHORIZATION 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
username	TEXT	IN	否	数据库用户名称。
privilege	TEXT	IN	否	定时任务权限。

  示例：

  ```
  gaussdb=# CREATE USER user1 PASSWORD '1*s*****';
  NOTICE:  The encrypted password contains MD5 ciphertext, which is not secure.
  CREATE ROLE
  gaussdb=# CALL DBE_SCHEDULER.grant_user_authorization('user1', 'create job');
   grant_user_authorization
  --------------------------


  (1 row)

  gaussdb=# CALL DBE_SCHEDULER.revoke_user_authorization('user1', 'create job');
   revoke_user_authorization
  ---------------------------


  (1 row)
  gaussdb=# DROP USER user1;
  DROP ROLE
  ```

- DBE_SCHEDULER.CREATE_CREDENTIAL

  创建授权证书。调用该函数的用户需要具有SYSADMIN权限。同步到所有CN节点。

  DBE_SCHEDULER.CREATE_CREDENTIAL原型为：

  ```
  DBE_SCHEDULER.create_credential(
  credential_name         text,
  username                text,
  password                text            default NULL,
  database_role           text            default NULL,
  windows_domain          text            default NULL,
  comments                text            default NULL
  )
  ```


版权所有 © 华为云计算技术有限公司

表 2-304 DBE_SCHEDULER.CREATE_CREDENTIAL 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
credential_name	TEXT	IN	否	授权证书名称。
username	TEXT	IN	否	数据库用户名称。
password	TEXT	IN	是	用户密码。
database_role	TEXT	IN	是	数据库系统权限。
windows_domain	TEXT	IN	是	Windows用户所属域。
comments	TEXT	IN	是	备注。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_credential('cre_1', 'user1', '');
 create_credential
-------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_credential('cre_1', false);
 drop_credential
-----------------

(1 row)
```

说明

DBE_SCHEDULER.CREATE_CREDENTIAL的password字段请传NULL或者'******'，该参数仅做兼容性处理，不代表实际含义。禁止使用安装用户对应的os用户名创建证书。

- DBE_SCHEDULER.DROP_CREDENTIAL

  销毁授权证书。调用该函数的用户需要具有SYSADMIN权限。仅证书被同步时同步操作。

  DBE_SCHEDULER.DROP_CREDENTIAL原型为：

```
DBE_SCHEDULER.drop_credential(
credential_name         text,
force boolean default false
)
```

表 2-305 DBE_SCHEDULER.DROP_CREDENTIAL 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
credential_name	TEXT	IN	否	授权证书名称。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
force	BOOLEAN	IN	否	删除授权证书行为标记位： ● true：无论是否有作业引用该证书，都会被删除。 ● false：任何作业都无法引用该证书，否则会发生错误。

示例：

```
gaussdb=# CALL DBE_SCHEDULER.create_credential('cre_1', 'user1', '');
 create_credential
-------------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_credential('cre_1', false);
 drop_credential
-----------------

(1 row)
```

- DBE_SCHEDULER.ENABLE

  启用对象。同步到所有CN节点。

  DBE_SCHEDULER.ENABLE原型为：

  ```
  DBE_SCHEDULER.enable(
  name text,
  commit_semantics text                 default 'STOP_ON_FIRST_ERROR'
  )
  ```

  表 2-306 DBE_SCHEDULER.ENABLE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
name	TEXT	IN	否	对象名称，可以指定一个或多个，当指定多个程序时需利用逗号隔开。
commit_semantics	TEXT	IN	否	提交规则。支持以下类型： ● 'STOP_ON_FIRST_ERROR'：在第一个报错之前的启用操作会提交。 ● 'TRANSACTIONAL'：事务级提交，报错前的启用操作会回滚。 ● 'ABSORB_ERRORS'：尝试越过报错，将成功的启用操作提交。

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL dbe_scheduler.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'stored_procedure', 'INSERT INTO
tb_job_test(key) VALUES(null);', 0, false, '');
 create_program
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.enable('job1');
 enable
--------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.enable('program1', 'STOP_ON_FIRST_ERROR');
 enable
--------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

- DBE_SCHEDULER.ENABLE_SINGLE

  启用单个对象。同步到所有CN节点。

  DBE_SCHEDULER.ENABLE_SINGLE原型为：

```
DBE_SCHEDULER.enable_single(
name text
)
```

表 2-307 DBE_SCHEDULER.ENABLE_SINGLE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
name	TEXT	IN	否	对象名称。

示例：

```
gaussdb=# CALL dbe_scheduler.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL DBE_SCHEDULER.enable_single('job1');
 enable_single
---------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

- DBE_SCHEDULER.DISABLE

  禁用多个对象，name为逗号分隔的字符串，每个逗号分隔的字符串为一个对象。仅启用操作同步时同步操作。

  DBE_SCHEDULER.DISABLE原型为：

```
DBE_SCHEDULER.disable(
name text,
force boolean                   default false,
commit_semantics text               default 'STOP_ON_FIRST_ERROR'
)
```

表 2-308 DBE_SCHEDULER.DISABLE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
name	TEXT	IN	否	对象名称。
force	BOOLEAN	IN	否	禁用对象行为标记位： ● true：无论是否有其他对象依赖于该对象，也会被禁用。 ● false：任何对象都无法依赖于该对象，否则会发生错误。
commit_semantics	TEXT	IN	否	提交规则。支持以下类型： ● 'STOP_ON_FIRST_ERROR'：在第一个报错之前的禁用操作会提交。 ● 'TRANSACTIONAL'：事务级提交，报错前的禁用操作会回滚。 ● 'ABSORB_ERRORS'：尝试越过报错，将成功的禁用操作提交。

示例：

```
gaussdb=# CALL dbe_scheduler.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.create_program('program1', 'stored_procedure', 'INSERT INTO
tb_job_test(key) VALUES(null);', 0, false, '');
 create_program
```


版权所有 © 华为云计算技术有限公司

```
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.disable('job1');
 disable
---------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.disable('program1', false, 'STOP_ON_FIRST_ERROR');
 disable
---------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_program('program1', false);
 drop_program
--------------

(1 row)
```

- DBE_SCHEDULER.DISABLE_SINGLE

  禁用单个对象。仅启用操作同步时同步操作。

  DBE_SCHEDULER.DISABLE_SINGLE原型为：

```
DBE_SCHEDULER.disable_single(
name text,
force boolean                    default false
)
```

表 2-309 DBE_SCHEDULER.DISABLE_SINGLE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
name	TEXT	IN	否	对象名称。
force	BOOLEAN	IN	否	禁用对象行为标记位： ● true：无论是否有其他对象依赖于该对象，也会被禁用。 ● false：任何对象都无法依赖于该对象，否则会发生错误。

示例：

```
gaussdb=# CALL dbe_scheduler.create_job('job1','PLSQL_BLOCK','BEGIN INSERT INTO test1
VALUES(12); end;',0,null,null,null,'DEFAULT_JOB_CLASS',false,false,null,null,null);
 create_job
------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.disable_single('job1', false);
```


版权所有 © 华为云计算技术有限公司

```
 disable_single
----------------

(1 row)

gaussdb=# CALL DBE_SCHEDULER.drop_job('job1', true, false, 'STOP_ON_FIRST_ERROR');
 drop_job
----------

(1 row)
```

- DBE_SCHEDULER.EVAL_CALENDAR_STRING

  分析调度任务周期。

  返回值类型：timestamp with time zone

  DBE_SCHEDULER.EVAL_CALENDAR_STRING原型为：

```
DBE_SCHEDULER.evaluate_calendar_string(
IN calendar_string text,
IN start_date timestamp with time zone,
IN return_date_after timestamp with time zone
)return timestamp with time zone
```

  表 2-310 DBE_SCHEDULER.EVAL_CALENDAR_STRING 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
calendar_string	TEXT	IN	否	定时任务日期字符串。
start_date	TIMESTAMP WITH TIME ZONE	IN	否	定时任务开始时间。
return_date_after	TIMESTAMP WITH TIME ZONE	IN	否	定时任务返回日期，用于筛选出晚于该时间的第一个匹配调度时刻。

  示例：

```
gaussdb=# CALL DBE_SCHEDULER.eval_calendar_string('FREQ=DAILY; BYHOUR=6;', sysdate, sysdate);
  eval_calendar_string
------------------------
 2023-09-15 06:47:24+08
(1 row)
```

- DBE_SCHEDULER.EVALUATE_CALENDAR_STRING

  分析调度任务周期。

  返回值类型：timestamp with time zone

  DBE_SCHEDULER.EVALUATE_CALENDAR_STRING原型为：

```
DBE_SCHEDULER.evaluate_calendar_string(
IN calendar_string text,
IN start_date timestamp with time zone,
IN return_date_after timestamp with time zone,
OUT next_run_date timestamp with time zone
)return timestamp with time zone
```


版权所有 © 华为云计算技术有限公司

表 2-311 DBE_SCHEDULER.EVALUATE_CALENDAR_STRING 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
calendar_string	TEXT	IN	否	定时任务日期字符串。
start_date	TIMESTAMP WITH TIME ZONE	IN	否	定时任务开始时间。
return_date_after	TIMESTAMP WITH TIME ZONE	IN	否	定时任务返回日期，用于筛选出晚于该时间的第一个匹配调度时刻。
next_run_date	TIMESTAMP WITH TIME ZONE	OUT	否	定时任务返回下一个日期。

示例：

```
gaussdb=# CREATE OR REPLACE PROCEDURE pr1(calendar_str text) as
DECLARE
    start_date        timestamp with time zone;
    return_date_after timestamp with time zone;
    next_run_date     timestamp with time zone;
BEGIN
    start_date := '2003-2-1 10:30:00.111111+8'::timestamp with time zone;
    return_date_after := start_date;
    DBE_SCHEDULER.evaluate_calendar_string(
        calendar_str,
        start_date, return_date_after, next_run_date);
    DBE_OUTPUT.PRINT_LINE('next_run_date: ' || next_run_date);
    return_date_after := next_run_date;
END;
/
CREATE PROCEDURE

gaussdb=# CALL pr1('FREQ=hourly;INTERVAL=2;BYHOUR=6,10;BYMINUTE=0;BYSECOND=0');
next_run_date: 2003-02-02 06:00:00+08
 pr1
-----

(1 row)
```

2.12.2.16 DBE_SESSION

接口介绍

高级功能包DBE_SESSION支持的接口请参见**表2-312**。DBE_SESSION作用范围是会话级别。

表 2-312 DBE_SESSION

接口名称	描述
DBE_SESSION.SET_CONTEXT	设置指定context下，某一属性（attribute）的值（value）。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_SESSION.CLEAR_CONTEXT	清除指定context下，某一属性（attribute）的值（value）。
DBE_SESSION.SEARCH_CONTEXT	查找指定context下，某一属性（attribute）的值（value）。
DBE_SESSION.MODIFY_PACKAGE_STATE	用于修改当前会话的PL/SQL的状态。

- DBE_SESSION.SET_CONTEXT

  向指定namespace(context)下，设置某一属性（attribute）的值（value）。

  DBE_SESSION.SET_CONTEXT原型为：

```
DBE_SESSION.SET_CONTEXT(
    namespace text,
    attribute text,
    value text
)returns void;
```

表 2-313 DBE_SESSION.SET_CONTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
namespace	TEXT	IN	否	需要设置的context名称，当context不存在时，新建context，最长支持128个字节，超长时将会截断。
attribute	TEXT	IN	否	属性名称，最长支持1024个字节，超长时将会截断。
value	TEXT	IN	否	要设置的值的名称，最长支持1024个字节，超长时将会截断。

- DBE_SESSION.CLEAR_CONTEXT

  清除指定namespace(context)下，某一属性（attribute）的值(value）。

  DBE_SESSION.CLEAR_CONTEXT原型为：

```
DBE_SESSION.CLEAR_CONTEXT (
    namespace text,
    client_identifier text default null,
    attribute text default null
)returns void ;
```

表 2-314 DBE_SESSION.CLEAR_CONTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
namespace	TEXT	IN	否	用户指定的context，最长支持128个字节，超长时将会截断。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
client_identifier	TEXT	IN	是	客户端认证，默认值null，通常情况用户无需手动设置。
attribute	TEXT	IN	是	要清除的属性，默认值null，表示清除指定context的所有的属性，最长支持1024个字节，超长时将会截断。 **说明** 为了保证前向兼容，参数值为字符串'null'时也会清除指定context的所有属性。

- DBE_SESSION.SEARCH_CONTEXT

  查找指定namespace(context)下，某一属性（attribute）的值（value）。

  DBE_SESSION.SEARCH_CONTEXT原型为：

```
DBE_SESSION.SEARCH_CONTEXT (
    namespace text,
    attribute text
)returns text;
```

  表 2-315 DBE_SESSION.SEARCH_CONTEXT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
namespace	TEXT	IN	否	用户指定的context，最长支持128个字节，超长时将会截断。
attribute	TEXT	IN	否	要查找的属性，最长支持1024个字节，超长时将会截断。

  示例：

```
DECLARE
    a text;
BEGIN
    DBE_SESSION.set_context('test', 'gaussdb', 'one');   --设置名为test的context下属性为gaussdb的值为one
    a := DBE_SESSION.search_context('test', 'gaussdb');
    DBE_OUTPUT.PRINT_LINE(a);
    DBE_SESSION.clear_context('test', 'test','gaussdb');
END;
/
-- 预期结果为：
one
ANONYMOUS BLOCK EXECUTE
```

- DBE_SESSION.MODIFY_PACKAGE_STATE

  用于修改当前会话的PL/SQL的状态。

  DBE_SESSION.MODIFY_PACKAGE_STATE原型为：

```
DBE_SESSION.MODIFY_PACKAGE_STATE (
    flags int);
```


版权所有 © 华为云计算技术有限公司

表 2-316 DBE_SESSION.MODIFY_PACKAGE_STATE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
flags	INT	IN	否	对PL/SQL采取的操作的位标志。 当flags=1时，执行完最顶层的PL/SQL代码后，会释放会话中当前运行的PL/SQL会话缓存。清除所有包的全局变量当前值，并关闭缓存的游标。在后续使用时，PL/SQL将重新实例化，并重新初始化包的全局变量。 不支持其他位标志。

示例：

```
CALL dbe_session.modify_package_state(1);
 modify_package_state
----------------------

(1 row)
```

2.12.2.17 DBE_SQL

数据类型介绍

- DBE_SQL.DESC_REC

  该类型是复合类型，用来存储DBE_SQL.SQL_DESCRIBE_COLUMNS接口中的描述信息。

  DBE_SQL.DESC_REC类型的原型为：

  ```
  CREATE TYPE DBE_SQL.DESC_REC AS (
      col_type            int,
      col_max_len          int,
      col_name            VARCHAR2(32),
      col_name_len         int,
      col_schema_name      VARCHAR2(32),
      col_schema_name_len int,
      col_precision        int,
      col_scale           int,
      col_charsetid        int,
      col_charsetform      int,
      col_null_ok          BOOLEAN
  );
  ```

- DBE_SQL.DESC_TAB

  该类型是DESC_REC的TABLE类型，通过TABLE OF语法实现。

  DBE_SQL.DESC_TAB类型的原型为：

  ```
  CREATE TYPE DBE_SQL.DESC_TAB AS TABLE OF DBE_SQL.DESC_REC;
  ```

- DBE_SQL.DATE_TABLE

  该类型是DATE的TABLE类型，通过TABLE OF语法实现。

  DBE_SQL.DATE_TABLE类型的原型为：

  ```
  CREATE TYPE DBE_SQL.DATE_TABLE AS TABLE OF DATE;
  ```


版权所有 © 华为云计算技术有限公司

说明

- 当高级包创建时，若GUC参数mapping_date_to_datea的值为on，则会将DBE_SQL.DATE_TABLE映射为DATEA的TABLE类型。
- 使用EXECUTE DIRECT ON语法执行DBE_SQL包中的函数时，如果GUC参数enable_udf_execute_direct的值为on，则可以执行；如果其值为off，则不可以执行。

- DBE_SQL.NUMBER_TABLE

  该类型是NUMBER的TABLE类型，通过TABLE OF语法实现。

  DBE_SQL.NUMBER_TABLE类型的原型为：

  ```
  CREATE TYPE DBE_SQL.NUMBER_TABLE AS TABLE OF NUMBER;
  ```

- DBE_SQL.VARCHAR2_TABLE

  该类型是VARCHAR2的TABLE类型，通过TABLE OF语法实现。

  DBE_SQL.VARCHAR2_TABLE类型的原型为：

  ```
  CREATE TYPE DBE_SQL.VARCHAR2_TABLE AS TABLE OF VARCHAR2(32767);
  ```

- DBE_SQL.BLOB_TABLE

  该类型是BLOB的TABLE类型，通过TABLE OF语法实现。

  DBE_SQL.BLOB_TABLE类型的原型为：

  ```
  CREATE TYPE DBE_SQL.BLOB_TABLE AS TABLE OF BLOB;
  ```

接口介绍

高级功能包DBE_SQL支持的接口请参见**表2-317**。

表 2-317 DBE_SQL

接口名称	描述
DBE_SQL.REGISTER_CONTEXT	打开一个游标。
DBE_SQL.SQL_UNREGISTER_CONTEXT	关闭一个已打开的游标。
DBE_SQL.SQL_SET_SQL	向游标传递一组SQL语句或匿名块。
DBE_SQL.SQL_RUN	执行给定游标中的SQL语句或匿名块。
DBE_SQL.NEXT_ROW	读取游标一行数据。
DBE_SQL.SET_RESULT_TYPE	动态定义一个列。
DBE_SQL.SET_RESULT_TYPE_CHAR	动态定义一个CHAR类型的列。
DBE_SQL.SET_RESULT_TYPE_INT	动态定义一个INT类型的列。
DBE_SQL.SET_RESULT_TYPE_LONG	动态定义一个LONG类型的列。
DBE_SQL.SET_RESULT_TYPE_RAW	动态定义一个RAW类型的列。
DBE_SQL.SET_RESULT_TYPE_TEXT	动态定义一个TEXT类型的列。
DBE_SQL.SET_RESULT_TYPE_UNKNOWN	动态定义一个未知列（类型不识别时使用此接口）。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_SQL.GET_RESULT	读取一个已动态定义的列值。
DBE_SQL.GET_RESULT_CHAR	读取一个已动态定义的列值（指定CHAR类型）。
DBE_SQL.GET_RESULT_INT	读取一个已动态定义的列值（指定INT类型）。
DBE_SQL.GET_RESULT_LONG	读取一个已动态定义的列值（指定LONG类型）。
DBE_SQL.GET_RESULT_RAW	读取一个已动态定义的列值（指定RAW类型）。
DBE_SQL.GET_RESULT_TEXT	读取一个已动态定义的列值（指定TEXT类型）。
DBE_SQL.GET_RESULT_UNKNOWN	读取一个已动态定义的列值（类型不识别时使用此接口）。
DBE_SQL.DBE_SQL_GET_RESULT_CHAR	读取一个已动态定义的列值（指定CHAR类型）。
DBE_SQL.DBE_SQL_GET_RESULT_LONG	读取一个已动态定义的列值（指定LONG类型）。
DBE_SQL.DBE_SQL_GET_RESULT_LONG2	读取一个已动态定义的列值（指定LONG类型）。
DBE_SQL.DBE_SQL_GET_RESULT_RAW	读取一个已动态定义的列值（指定RAW类型）。
DBE_SQL.IS_ACTIVE	检查游标是否已打开。
DBE_SQL.LAST_ROW_COUNT	返回获取行数的累积计数。
DBE_SQL.RUN_AND_NEXT	在游标上执行一组动态定义操作后，读取游标数据。
DBE_SQL.SQL_BIND_VARIABLE	根据语句中的变量，绑定一个值到该变量。
DBE_SQL.SQL_BIND_ARRAY	根据语句中的变量，绑定一组值到该变量。
DBE_SQL.SET_RESULT_TYPE_INTS	动态定义一个INT数组类型的列。
DBE_SQL.SET_RESULT_TYPE_TEXTS	动态定义一个TEXT数组类型的列。
DBE_SQL.SET_RESULT_TYPE_RAWS	动态定义一个RAW数组类型的列。
DBE_SQL.SET_RESULT_TYPE_BYTEAS	动态定义一个BYTEA数组类型的列。
DBE_SQL.SET_RESULT_TYPE_CHARS	动态定义一个CHAR数组类型的列。
DBE_SQL.SET_RESULTS_TYPE	动态定义一个数组类型的列。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_SQL.GET_RESULTS_INT	读取一个已动态定义的列值（指定INT数组类型）。
DBE_SQL.GET_RESULTS_TEXT	读取一个已动态定义的列值（指定TEXT数组类型）。
DBE_SQL.GET_RESULTS_RAW	读取一个已动态定义的列值（指定RAW数组类型）。
DBE_SQL.GET_RESULTS_BYTEA	读取一个已动态定义的列值（指定BYTEA数组类型）。
DBE_SQL.GET_RESULTS_CHAR	读取一个已动态定义的列值（指定CHAR数组类型）。
DBE_SQL.GET_RESULTS	读取一个已动态定义的列值。
DBE_SQL.DESCRIBE_COLUMNS	描述游标读取的列信息。
DBE_SQL.SQL_DESCRIBE_COLUMNS	描述游标读取的列信息。
DBE_SQL.BIND_VARIABLE	绑定参数接口。
DBE_SQL.SQL_SET_RESULTS_TYPE_C	动态定义一个数组类型的列。
DBE_SQL.SQL_GET_VALUES_C	读取一个已动态定义的列值。
DBE_SQL.GET_VARIABLE_RESULT	读取一个SQL语句执行后的返回值。
DBE_SQL.GET_VARIABLE_RESULT_CHAR	读取一个SQL语句执行后的返回值（指定CHAR类型）。
DBE_SQL.GET_VARIABLE_RESULT_RAW	读取一个SQL语句执行后的返回值（指定RAW类型）。
DBE_SQL.GET_VARIABLE_RESULT_TEXT	读取一个SQL语句执行后的返回值（指定TEXT类型）。
DBE_SQL.GET_VARIABLE_RESULT_INT	读取一个SQL语句执行后的返回值（指定INT类型）。
DBE_SQL.GET_ARRAY_RESULT_TEXT	读取一个SQL语句执行后的返回值（指定TEXT数组类型）。
DBE_SQL.GET_ARRAY_RESULT_RAW	读取一个SQL语句执行后的返回值（指定RAW数组类型）。
DBE_SQL.GET_ARRAY_RESULT_CHAR	读取一个SQL语句执行后的返回值（指定CHAR数组类型）。
DBE_SQL.GET_ARRAY_RESULT_INT	读取一个SQL语句执行后的返回值（指定INT数组类型）。

版权所有 © 华为云计算技术有限公司

说明

- 建议使用DBE_SQL.SET_RESULT_TYPE及DBE_SQL.GET_RESULT定义参数列。
- 当结果集大小超过GUC参数work_mem的设置值时，将触发结果集的临时磁盘存储，但临时磁盘存储最大阈值不超过512MB。

- DBE_SQL.REGISTER_CONTEXT

  该函数用来打开一个游标，是后续DBE_SQL各项操作的前提。该函数不传入任何参数，内部自动递增生成游标ID，并作为返回值返回给integer定义的变量。

  说明

  DBE_SQL打开的游标是会话级的变量，不支持跨会话调用打开的游标（如自治事务），如果调用跨会话的游标，那么行为不可预知。

  DBE_SQL.REGISTER_CONTEXT原型为：

```
DBE_SQL.REGISTER_CONTEXT(
)
RETURN INTEGER;
```

- DBE_SQL.SQL_UNREGISTER_CONTEXT

  该函数用来关闭一个游标，是DBE_SQL各项操作的结束步骤。如果在存储过程结束时未调用该函数，游标占用的内存将不会被释放。因此，关闭游标至关重要。由于异常情况可能导致存储过程提前退出，从而未能关闭游标，建议在存储过程的异常处理中包含此接口。

  DBE_SQL.SQL_UNREGISTER_CONTEXT原型为：

```
DBE_SQL.SQL_UNREGISTER_CONTEXT(
    context_id    IN INT
)
RETURN INTEGER;
```

表 2-318 DBE_SQL.SQL_UNREGISTER_CONTEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	将要关闭的游标id号。

- DBE_SQL.SQL_SET_SQL

  该函数用来解析给定游标的SQL语句或匿名块。目前语句参数仅可通过text类型传递，长度不大于1GB。

  DBE_SQL.SQL_SET_SQL原型为：

```
DBE_SQL.SQL_SET_SQL(
   context_id    IN INT,
   query_string  IN TEXT,
   language_flag IN INT
)
RETURN BOOLEAN;
```


版权所有 © 华为云计算技术有限公司

表 2-319 DBE_SQL.SQL_SET_SQL 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行查询语句解析的游标id。
query_string	TEXT	IN	否	执行的查询语句。
language_flag	INT	IN	否	版本语言号。指定SQL语句不同版本的行为，1为非兼容版本，2为兼容ORA版本。

- DBE_SQL.SQL_RUN

  该函数用于执行指定的游标。它接收一个游标id，并执行给定游标中的SQL语句或匿名块。

  DBE_SQL.SQL_RUN原型为：

```
DBE_SQL.SQL_RUN(
    context_id     IN INT,
)
RETURN INTEGER;
```

表 2-320 DBE_SQL.SQL_RUN 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行查询语句解析的游标id。

- DBE_SQL.NEXT_ROW

  该函数返回符合查询条件的数据行数，每次调用该函数时，都会获取一个新的行数的集合，直到数据读取完毕获取不到新行为止。

  DBE_SQL.NEXT_ROW原型为：

```
DBE_SQL.NEXT_ROW(
    context_id     IN INT,
)
RETURN INTEGER;
```

表 2-321 DBE_SQL.NEXT_ROW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。

- DBE_SQL.SET_RESULT_TYPE

  该函数用来定义从指定游标返回的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_SQL.SET_RESULT_TYPE(
    context_id IN INT,
    pos        IN INT,
    column_ref IN ANYELEMENT,
    maxsize    IN INT default 1024
)
RETURN INTEGER;
```

表 2-322 DBE_SQL.SET_RESULT_TYPE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	查询列在返回结果中的相对位置，起始为1。
column_ref	ANYELEMENT	IN	否	任意类型的变量，可根据变量类型选择适当的接口动态定义列。
maxsize	INT	IN	是	定义列返回类型长度。

- DBE_SQL.SET_RESULT_TYPE_CHAR

  该函数用来定义从指定游标返回的CHAR类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_CHAR原型为：

```
DBE_SQL.SET_RESULT_TYPE_CHAR(
    context_id  IN INT,
    pos         IN INT,
    column_ref  IN TEXT,
    column_size IN INT
)
RETURN INTEGER;
```

表 2-323 DBE_SQL.SET_RESULT_TYPE_CHAR 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	TEXT	IN	否	需要定义的某类型的参数变量。
column_size	INT	IN	否	动态定义列长度。

- DBE_SQL.SET_RESULT_TYPE_INT

版权所有 © 华为云计算技术有限公司

该函数用来定义从指定游标返回的INT类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

DBE_SQL.SET_RESULT_TYPE_INT原型为：

```
DBE_SQL.SET_RESULT_TYPE_INT(
    context_id IN INT,
    pos        IN INT
)
RETURN INTEGER;
```

表 2-324 DBE_SQL.SET_RESULT_TYPE_INT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

- DBE_SQL.SET_RESULT_TYPE_LONG

  该函数用来定义从指定游标返回的长列类型（非数据类型long）的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。长列的大小限制为1G。

  DBE_SQL.SET_RESULT_TYPE_LONG原型为：

  ```
  DBE_SQL.SET_RESULT_TYPE_LONG(
      context_id IN INT,
      pos        IN INT
  )
  RETURN INTEGER;
  ```

  表 2-325 DBE_SQL.SET_RESULT_TYPE_LONG 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

- DBE_SQL.SET_RESULT_TYPE_RAW

  该函数用来定义从指定游标返回的RAW类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_RAW原型为：

  ```
  DBE_SQL.SET_RESULT_TYPE_RAW(
      context_id  IN INT,
      pos         IN INT,
      column_ref  IN BYTEA,
      column_size IN INT
  )
  RETURN INTEGER;
  ```


版权所有 © 华为云计算技术有限公司

表 2-326 DBE_SQL.SET_RESULT_TYPE_RAW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	BYTEA	IN	否	RAW类型的参数变量。
column_size	INT	IN	否	列的长度。

- DBE_SQL.SET_RESULT_TYPE_TEXT

  该函数用来定义从指定游标返回的TEXT类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_TEXT原型为：

```
DBE_SQL.SET_RESULT_TYPE_TEXT(
    context_id IN INT,
    pos        IN INT,
    maxsize    IN INT
)
RETURN INTEGER;
```

表 2-327 DBE_SQL.SET_RESULT_TYPE_TEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	IN	IN	否	执行的游标id。
pos	IN	IN	否	动态定义列在查询中的位置。
maxsize	IN	IN	否	定义的TEXT类型的最大长度。

- DBE_SQL.SET_RESULT_TYPE_UNKNOWN

  该函数用来处理从指定游标返回的未知数据类型的列，该接口仅在数据类型无法识别时用于报错并退出。

  DBE_SQL.SET_RESULT_TYPE_UNKNOWN原型为：

```
DBE_SQL.SET_RESULT_TYPE_UNKNOWN(
    context_id IN INT,
    pos        IN INT,
    col_type   IN TEXT
)
RETURN INTEGER;
```


版权所有 © 华为云计算技术有限公司

表 2-328 DBE_SQL.SET_RESULT_TYPE_UNKNOWN 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
col_type	TEXT	IN	否	动态定义的参数。

- DBE_SQL.GET_RESULT

  该函数用来返回指定游标在给定位置的元素值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULT原型为：

```
DBE_SQL.GET_RESULT(
    context_id    IN INT,
    pos           IN INT,
    column_value INOUT ANYELEMENT
);
```

表 2-329 DBE_SQL.GET_RESULT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	查询列在返回结果中的相对位置，起始为1。
column_value	ANYELEMENT	INOUT	否	定义的列的返回值。

- DBE_SQL.GET_RESULT_CHAR

  该函数用来返回指定游标在给定位置的CHAR类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULT_CHAR原型为：

```
DBE_SQL.GET_RESULT_CHAR(
    context_id    IN    INT,
    pos           IN    INT,
    tr           INOUT CHAR,
    err           INOUT NUMERIC DEFAULT 0,
    actual_length INOUT INT DEFAULT 1024
);
```

表 2-330 DBE_SQL.GET_RESULT_CHAR 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
pos	INT	IN	否	动态定义列在查询中的位置。
tr	CHAR	INOUT	否	返回值。
err	NUMERIC	INOUT	是	错误号。传出参数，须传入变量做参数（默认值为0）。目前未实现，固定传出-1。
actual_length	INT	INOUT	是	返回值的实际长度，默认值为1024。

- DBE_SQL.GET_RESULT_INT

  该函数用来返回指定游标在给定位置的INT类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULT_INT原型为：

```
DBE_SQL.GET_RESULT_INT(
    context_id IN    INT,
    pos        IN    INT
)
RETURN INTEGER;
```

  表 2-331 DBE_SQL.GET_RESULT_INT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

- DBE_SQL.GET_RESULT_LONG

  该函数用来返回指定游标在给定位置的长列（非long/bigint整型）类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULT_LONG原型为：

```
DBE_SQL.GET_RESULT_LONG(
    context_id IN    INT,
    pos        IN    INT,
    lgth       IN    INT,
    off_set    IN    INT,
    vl         INOUT TEXT,
    vl_length  INOUT INT
);
```


版权所有 © 华为云计算技术有限公司

表 2-332 DBE_SQL.GET_RESULT_LONG 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
lgth	INT	IN	否	返回值的长度。
off_set	INT	IN	否	返回值的起始位置。
vl	TEXT	INOUT	否	返回值。
vl_length	INT	INOUT	否	实际返回值的长度。

- DBE_SQL.GET_RESULT_RAW

  该函数用来返回指定游标在给定位置的RAW类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULT_RAW原型为：

```
DBE_SQL.GET_RESULT_RAW(
    context_id     IN     INT,
    pos            IN     INT,
    tr             INOUT BYTEA,
    err            INOUT NUMERIC DEFAULT 0,
    actual_length INOUT INT DEFAULT 1024
);
```

表 2-333 DBE_SQL.GET_RESULT_RAW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
tr	BYTEA	INOUT	否	返回的列值。
err	NUMERIC	INOUT	否	错误号。传出参数，须传入变量做参数（默认值为0）。目前未实现，固定传出-1。
actual_length	INT	INOUT	否	返回值的实际长度，不能长于此值，否则截断，默认值为1024。

- DBE_SQL.GET_RESULT_TEXT

  该函数用来返回指定游标在给定位置的TEXT类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULT_TEXT原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_SQL.GET_RESULT_TEXT(
    context_id IN INT,
    pos        IN INT
)
RETURN TEXT;
```

表 2-334 DBE_SQL.GET_RESULT_TEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

- DBE_SQL.GET_RESULT_UNKNOWN

  该函数用来返回指定游标在给定位置的未知类型的值。该接口为类型不支持时的报错处理接口。

  DBE_SQL.GET_RESULT_UNKNOWN原型为：

```
DBE_SQL.GET_RESULT_UNKNOWN(
    context_id IN INT,
    pos        IN INT,
    col_type   IN TEXT
)
RETURN INTEGER;
```

表 2-335 DBE_SQL.GET_RESULT_UNKNOWN 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
col_type	TEXT	IN	否	返回的参数类型。

- DBE_SQL.DBE_SQL_GET_RESULT_CHAR

  该函数用来返回指定游标在给定位置的CHAR类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。和DBE_SQL.GET_RESULT_CHAR的区别是，该函数不设置返回值长度，返回整个字符串。

  DBE_SQL.DBE_SQL_GET_RESULT_CHAR原型为：

```
DBE_SQL.DBE_SQL_GET_RESULT_CHAR(
    context_id    IN INT,
    pos           IN INT
)
RETURN CHARACTER;
```


版权所有 © 华为云计算技术有限公司

表 2-336 DBE_SQL.DBE_SQL_GET_RESULT_CHAR 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

- DBE_SQL.DBE_SQL_GET_RESULT_LONG

  该函数用来返回指定游标在给定位置的长列（非long/bigint整型）类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。和DBE_SQL.GET_RESULT_LONG的区别在于，该函数不设置返回值长度，返回整个BIGINT值。

  DBE_SQL.DBE_SQL_GET_RESULT_LONG原型为：

```
DBE_SQL.DBE_SQL_GET_RESULT_LONG(
    context_id    IN INT,
    pos           IN INT
)
RETURN BIGINT;
```

表 2-337 DBE_SQL.DBE_SQL_GET_RESULT_LONG 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

说明

由于该函数的返回类型和DBE_SQL.SET_RESULT_TYPE_LONG、DBE_SQL.GET_RESULT_LONG不一致，请使用DBE_SQL.DBE_SQL_GET_RESULT_LONG2代替。

- DBE_SQL.DBE_SQL_GET_RESULT_LONG2

  该函数用来返回指定游标在给定位置的长列（非long/bigint整型）类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。和DBE_SQL.GET_RESULT_LONG的区别在于，该函数不设置返回值长度，返回整列的值。

  DBE_SQL.DBE_SQL_GET_RESULT_LONG2原型为：

```
DBE_SQL.DBE_SQL_GET_RESULT_LONG2(
    context_id IN INTEGER,
    pos        IN INTEGER
)
RETURN TEXT;
```


版权所有 © 华为云计算技术有限公司

表 2-338 DBE_SQL.DBE_SQL_GET_RESULT_LONG2 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INTEGER	IN	否	执行的游标id。
pos	INTEGER	IN	否	动态定义列在查询中的位置。

- DBE_SQL.DBE_SQL_GET_RESULT_RAW

  该函数用来返回指定游标在给定位置的RAW类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  和函数DBE_SQL.GET_RESULT_RAW的区别在于，该函数不设置返回值长度，返回整个字符串。

  DBE_SQL.DBE_SQL_GET_RESULT_RAW原型为：

```
DBE_SQL.DBE_SQL_GET_RESULT_RAW(
    context_id    IN    INT,
    pos           IN    INT
)
RETURN RAW;
```

表 2-339 DBE_SQL.GET_RESULT_RAW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。

- DBE_SQL.IS_ACTIVE

  该函数用来返回游标的当前状态。游标处于打开、解析、执行、定义时为true；关闭后为false；未知时报错；其余默认为关闭。

  DBE_SQL.IS_ACTIVE原型为：

```
DBE_SQL.IS_ACTIVE(
    context_id    IN    INT
)
RETURN BOOLEAN;
```

表 2-340 DBE_SQL.IS_ACTIVE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。

- DBE_SQL.LAST_ROW_COUNT

  该函数用于返回最近一次NEXT_ROW执行后获取的数据行数的累积计数。

  DBE_SQL.LAST_ROW_COUNT原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_SQL.LAST_ROW_COUNT(
)
RETURN INTEGER;
```

- DBE_SQL.RUN_AND_NEXT

  该函数的功能等同于在调用SQL_RUN后接着调用NEXT_ROW。

  DBE_SQL.RUN_AND_NEXT原型为：

```
DBE_SQL.RUN_AND_NEXT(
    context_id IN INT
)
RETURNS INTEGER;
```

  表 2-341 DBE_SQL.RUN_AND_NEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	执行查询语句解析的游标id。

- DBE_SQL.SQL_BIND_VARIABLE

  该函数用于将一个参数绑定到SQL语句，当执行该SQL语句时，将根据该绑定的值来执行。

  DBE_SQL.SQL_BIND_VARIABLE原型为：

```
DBE_SQL.SQL_BIND_VARIABLE(
    context_id      IN int,
    query_string    IN text,
    language_flag   IN anyelement,
    out_value_size IN int default null
)
RETURNS void;
```

  表 2-342 DBE_SQL.SQL_BIND_VARIABLE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
query_string	TEXT	IN	否	绑定的变量名。
language_flag	ANYELEMENT	IN	否	绑定的值。
out_value_size	INT	IN	是	返回值的大小，默认值为NULL。

- DBE_SQL.SQL_BIND_ARRAY

  该函数用于将一组参数绑定到SQL语句，当执行该SQL语句时，将根据该绑定的数组来执行。

  DBE_SQL.SQL_BIND_ARRAY函数的原型为：

```
DBE_SQL.SQL_BIND_ARRAY(
    context_id    IN int,
    query_string IN text,
    value         IN anyarray
```


版权所有 © 华为云计算技术有限公司

```
)
RETURNS void;
DBE_SQL.SQL_BIND_ARRAY(
    context_id   IN int,
    query_string IN text,
    value        IN anyarray,
    lower_index  IN int,
    higher_index IN int
)
RETURNS void;
```

表 2-343 DBE_SQL.SQL_BIND_ARRAY 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
query_string	TEXT	IN	否	绑定的变量名。
value	ANYARRAY	IN	否	绑定的数组。
lower_index	INT	IN	否	绑定数组的最小下标。
higher_index	INT	IN	否	绑定数组的最大下标。

- DBE_SQL.SET_RESULT_TYPE_INTS

  该函数用于定义从指定游标返回的INT数组类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_INTS原型为：

```
DBE_SQL.SET_RESULT_TYPE_INTS(
    context_id IN int,
    pos        IN int,
    column_ref IN anyarray,
    cnt        IN int,
    lower_bnd  IN int
)
RETURNS integer;
```

表 2-344 DBE_SQL.SET_RESULT_TYPE_INTS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。

- DBE_SQL.SET_RESULT_TYPE_TEXTS

  该函数用来定义从指定游标返回的TEXT数组类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_TEXTS原型为：

```
DBE_SQL.SET_RESULT_TYPE_TEXTS(
    context_id IN int,
    pos        IN int,
    column_ref IN anyarray,
    cnt        IN int,
    lower_bnd  IN int,
    maxsize    IN int
)
RETURNS VOID;
```

  表 2-345 DBE_SQL.SET_RESULT_TYPE_TEXTS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。
maxsize	INT	IN	否	定义的TEXT类型的最大长度。

- DBE_SQL.SET_RESULT_TYPE_RAWS

  该函数用来定义从指定游标返回的RAW数组类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_RAWS原型为：

```
DBE_SQL.set_result_type_raws(
    context_id  IN int,
    pos         IN int,
    column_ref  IN anyarray,
    cnt         IN int,
    lower_bnd   IN int,
    column_size IN int
)
RETURNS integer;
```


版权所有 © 华为云计算技术有限公司

表 2-346 DBE_SQL.SET_RESULT_TYPE_RAWS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。
column_size	INT	IN	否	列的长度。

- DBE_SQL.SET_RESULT_TYPE_BYTEAS

  该函数用来定义从指定游标返回的BYTEA数组类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_BYTEAS原型为：

```
DBE_SQL.set_result_type_byteas(
    context_id  IN int,
    pos         IN int,
    column_ref  IN anyarray,
    cnt         IN int,
    lower_bnd   IN int,
    column_size IN int
)
RETURNS integer;
```

表 2-347 DBE_SQL.SET_RESULT_TYPE_BYTEAS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。
column_size	INT	IN	否	列的长度。

版权所有 © 华为云计算技术有限公司

- DBE_SQL.SET_RESULT_TYPE_CHARS

  该函数用来定义从指定游标返回的CHAR数组类型的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULT_TYPE_CHARS原型为：

```
DBE_SQL.SET_RESULT_TYPE_CHARS(
    context_id  IN int,
    pos         IN int,
    column_ref  IN anyarray,
    cnt         IN int,
    lower_bnd   IN int,
    column_size IN int
)
RETURNS integer;
```

  表 2-348 DBE_SQL.SET_RESULT_TYPE_CHARS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。
column_size	INT	IN	否	列的长度。

- DBE_SQL.SET_RESULTS_TYPE

  该函数用来定义从指定游标返回的列。该接口只能应用于SELECT定义的游标。定义的列通过查询列表中的相对位置来标识，传入变量的数据类型决定了该列被定义的类型。

  DBE_SQL.SET_RESULTS_TYPE原型为：

```
DBE_SQL.SET_RESULTS_TYPE(
    context_id IN int,
    pos        IN int,
    column_ref IN anyarray,
    cnt        IN int,
    lower_bnd  IN int,
    maxsize    IN int DEFAULT 1024
) returns void;
```


版权所有 © 华为云计算技术有限公司

表 2-349 DBE_SQL.SET_RESULTS_TYPE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。
maxsize	INT	IN	是	定义的类型的最大长度。

- DBE_SQL.GET_RESULTS_INT

  该函数用来返回指定游标在给定位置的INT数组类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULTS_INT原型为：

```
DBE_SQL.GET_RESULTS_INT(
    context_id   IN    int,
    pos          IN    int,
    column_value INOUT anyarray
);
```

表 2-350 DBE_SQL.GET_RESULTS_INT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_RESULTS_TEXT

  该函数用来返回指定游标在给定位置的TEXT数组类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULTS_TEXT原型为：

```
DBE_SQL.GET_RESULTS_TEXT(
    context_id   IN    int,
    pos          IN    int,
    column_value INOUT anyarray
);
```


版权所有 © 华为云计算技术有限公司

表 2-351 DBE_SQL.GET_RESULTS_TEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_RESULTS_RAW

  该函数用来返回指定游标在给定位置的RAW数组类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULTS_RAW原型为：

```
DBE_SQL.GET_RESULTS_RAW(
    context_id   IN    int,
    pos          IN    int,
    column_value INOUT anyarray
);
```

表 2-352 DBE_SQL.GET_RESULTS_RAW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_RESULTS_BYTEA

  该函数用来返回指定游标在给定位置的BYTEA数组类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULTS_BYTEA原型为：

```
DBE_SQL.GET_RESULTS_BYTEA(
    context_id   IN    int,
    pos          IN    int,
    column_value INOUT anyarray
);
```

表 2-353 DBE_SQL.GET_RESULTS_BYTEA 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
pos	INT	IN	否	动态定义列在查询中的位置。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_RESULTS_CHAR

  该函数用来返回指定游标在给定位置的CHAR数组类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  DBE_SQL.GET_RESULTS_CHAR原型为：

```
DBE_SQL.GET_RESULTS_CHAR(
    context_id   IN    int,
    pos          IN    int,
    column_value INOUT anyarray
);
```

  表 2-354 DBE_SQL.GET_RESULTS_CHAR 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_RESULTS

  该函数用来返回指定游标在给定位置的数组类型的值。该接口访问的是由DBE_SQL.NEXT_ROW获取的数据。

  说明

  由于DBE_SQL.GET_RESULTS的底层机制是通过数组实现的，当使用不同的数组获取同一列的返回值时，内部索引的不连续性会导致数组中填充NULL值，以确保数组索引的连续性。这将导致返回结果数组的长度和ORA数据库中的不一致。

  DBE_SQL.GET_RESULTS原型为：

```
DBE_SQL.GET_RESULTS(
    context_id   IN    int,
    pos          IN    int,
    column_value INOUT anyarray
);
```

  表 2-355 DBE_SQL.GET_RESULTS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
pos	INT	IN	否	动态定义列在查询中的位置。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.SQL_DESCRIBE_COLUMNS

  该函数用来描述列信息。该接口只能应用于SELECT定义的游标。

  DBE_SQL.SQL_DESCRIBE_COLUMNS原型为：

```
DBE_SQL.SQL_DESCRIBE_COLUMNS(
    context_id IN     int,
    col_cnt    INOUT int,
    desc_t     INOUT dbe_sql.desc_tab
)RETURNS record;
```

  表 2-356 DBE_SQL.SQL_DESCRIBE_COLUMNS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
col_cnt	INT	INOUT	否	返回的列的数量。
desc_t	DBE_SQL.DESC_TAB	INOUT	否	返回的列的描述信息。

- DBE_SQL.DESCRIBE_COLUMNS

  该函数用来描述列信息。该接口为兼容接口，只能应用于SELECT定义的游标。

  DBE_SQL.DESCRIBE_COLUMNS原型为：

```
DBE_SQL.DESCRIBE_COLUMNS(
    context_id IN  int,
    col_cnt    OUT int,
    desc_t     OUT dbe_sql.desc_tab
)
```

  表 2-357 DBE_SQL.DESCRIBE_COLUMNS 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
col_cnt	INT	OUT	否	返回的列的数量。
desc_t	DBE_SQL.DESC_TAB	OUT	否	返回的列的描述信息。

- DBE_SQL.BIND_VARIABLE

版权所有 © 华为云计算技术有限公司

该函数是绑定参数接口。建议使用DBE_SQL.SQL_BIND_VARIABLE。

- DBE_SQL.SQL_SET_RESULTS_TYPE_C

  该函数动态定义一个数组类型的列。不建议用户使用。

  DBE_SQL.SQL_SET_RESULTS_TYPE_C原型为：

```
DBE_SQL.sql_set_results_type_c(
    context_id IN int,
    pos        IN int,
    column_ref IN anyarray,
    cnt        IN int,
    lower_bnd  IN int,
    col_type   IN anyelement,
    maxsize    IN int
)return integer;
```

表 2-358 DBE_SQL.SQL_SET_RESULTS_TYPE_C 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	INT	IN	否	动态定义列在查询中的位置。
column_ref	ANYARRAY	IN	否	标记返回的数组类型。
cnt	INT	IN	否	标记一次获取多少个值。
lower_bnd	INT	IN	否	标记返回数组时的开始下标。
col_type	ANYELEMENT	IN	否	标记返回的数组类型对应的变量类型。
maxsize	INT	IN	否	定义的类型的最大长度。

- DBE_SQL.SQL_GET_VALUES_C

  该函数读取一个已动态定义的列值。不建议用户使用。

  DBE_SQL.SQL_GET_VALUES_C原型为：

```
DBE_SQL.sql_get_values_c(
    context_id   IN    int,
    pos          IN    int,
    results_type INOUT anyarray,
    result_type  IN    anyelement
)return anyarray;
```

表 2-359 DBE_SQL.SQL_GET_VALUES_C 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
pos	INT	IN	否	参数位置信息。
results_type	ANYARRAY	INOUT	否	获取的结果。
result_type	ANYELEMENT	IN	否	获取的结果类型。

- DBE_SQL.GET_VARIABLE_RESULT

  该函数用来返回绑定的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_VARIABLE_RESULT原型为：

```
DBE_SQL.get_variable_result(
    context_id   IN    int,
    pos          IN    VARCHAR2,
    column_value INOUT anyelement
);
```

  表 2-360 DBE_SQL.GET_VARIABLE_RESULT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。
column_value	ANYELEMENT	INOUT	否	返回值。

- DBE_SQL.GET_VARIABLE_RESULT_CHAR

  该函数用于返回绑定的CHAR类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_VARIABLE_RESULT_CHAR原型为：

```
DBE_SQL.get_variable_result_char(
    context_id IN int,
    pos        IN VARCHAR2
)
RETURNS char
```

  表 2-361 DBE_SQL.GET_VARIABLE_RESULT_CHAR 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。

版权所有 © 华为云计算技术有限公司

- DBE_SQL.GET_VARIABLE_RESULT_RAW

  该函数用于返回绑定的RAW类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_VARIABLE_RESULT_RAW原型为：

```
DBE_SQL.get_variable_result_raw(
    context_id IN     int,
    pos        IN     VARCHAR2,
    value      INOUT anyelement
);
```

  表 2-362 DBE_SQL.GET_VARIABLE_RESULT_RAW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。
value	ANYELEMENT	INOUT	否	返回值。

- DBE_SQL.GET_VARIABLE_RESULT_TEXT

  该函数用于返回绑定的TEXT类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_VARIABLE_RESULT_TEXT原型为：

```
CREATE OR REPLACE FUNCTION DBE_SQL.get_variable_result_text(
    context_id IN int,
    pos        IN VARCHAR2
)
RETURNS text
```

  表 2-363 DBE_SQL.GET_VARIABLE_RESULT_TEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。

- DBE_SQL.GET_VARIABLE_RESULT_INT

  该函数用于返回绑定的INT类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_VARIABLE_RESULT_INT原型为：

```
DBE_SQL.get_variable_result_int(
    context_id IN     int,
    pos        IN     VARCHAR2,
    value      INOUT anyelement
);
```


版权所有 © 华为云计算技术有限公司

表 2-364 DBE_SQL.GET_VARIABLE_RESULT_INT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。
value	ANYELEMENT	INOUT	否	返回值。

- DBE_SQL.GET_ARRAY_RESULT_TEXT

  该函数用于返回绑定的TEXT数组类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_ARRAY_RESULT_TEXT原型为：

```
DBE_SQL.get_array_result_text(
    context_id   IN    int,
    pos          IN    VARCHAR2,
    column_value INOUT anyarray
);
```

表 2-365 DBE_SQL.GET_ARRAY_RESULT_TEXT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_ARRAY_RESULT_RAW

  该函数用于返回绑定的RAW数组类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_ARRAY_RESULT_RAW原型为：

```
DBE_SQL.get_array_result_raw(
    context_id   IN    int,
    pos          IN    VARCHAR2,
    column_value INOUT anyarray
);
```


版权所有 © 华为云计算技术有限公司

表 2-366 DBE_SQL.GET_ARRAY_RESULT_RAW 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_ARRAY_RESULT_CHAR

  该函数用于返回绑定的CHAR数组类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_ARRAY_RESULT_CHAR原型为：

```
DBE_SQL.get_array_result_char(
    context_id   IN    int,
    pos          IN    VARCHAR2,
    column_value INOUT anyarray
);
```

表 2-367 DBE_SQL.GET_ARRAY_RESULT_CHAR 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。
pos	VARCHAR2	IN	否	绑定的参数名。
column_value	ANYARRAY	INOUT	否	返回值。

- DBE_SQL.GET_ARRAY_RESULT_INT

  该函数用于返回绑定的INT数组类型的OUT参数的值，可以用于获取存储过程中的OUT参数。

  DBE_SQL.GET_ARRAY_RESULT_INT原型为：

```
DBE_SQL.get_array_result_int(
    context_id   IN    int,
    pos          IN    VARCHAR2,
    column_value INOUT anyarray
);
```

表 2-368 DBE_SQL.GET_ARRAY_RESULT_INT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
context_id	INT	IN	否	被查询的游标id。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
pos	VARCHAR2	IN	否	绑定的参数名。
column_value	ANYARRAY	INOUT	否	返回值。

示例

```
-- 示例1
-- 创建表并插入数据
gaussdb=# CREATE TABLE test_desc_cols(
    id NUMBER,
    name VARCHAR2(50)
);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'id' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# INSERT INTO test_desc_cols(id, name) VALUES (1, 'xiaoming');
INSERT 0 1
gaussdb=# INSERT INTO test_desc_cols(id, name) VALUES (2, 'xiaohong');
INSERT 0 1
gaussdb=# INSERT INTO test_desc_cols(id, name) VALUES (3, 'xiaolan');
INSERT 0 1

gaussdb=# DECLARE
context_id  INTEGER;
col_cnt     INTEGER;
v_id int;
v_name varchar2;
execute_ret  INTEGER;
BEGIN
-- 打开游标
context_id := DBE_SQL.REGISTER_CONTEXT();
-- 编译游标
DBE_SQL.SQL_SET_SQL(context_id, 'SELECT * FROM test_desc_cols', 2);
-- 设置列返回值的类型
DBE_SQL.SET_RESULT_TYPE(context_id, 1, v_id);
DBE_SQL.SET_RESULT_TYPE(context_id, 2, v_name);
execute_ret := DBE_SQL.SQL_RUN(context_id);
loop
exit when (DBE_SQL.NEXT_ROW(context_id) <= 0);
--获取值
DBE_SQL.GET_RESULT(context_id, 1, v_id);
DBE_SQL.GET_RESULT(context_id, 2, v_name);
--输出结果
dbe_output.print_line('id:'|| v_id || ' name:' || v_name);
end loop;
DBE_SQL.SQL_UNREGISTER_CONTEXT(context_id);
END;
/
id:1 name:xiaoming
id:2 name:xiaohong
id:3 name:xiaolan
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop table if exists test_desc_cols;
DROP TABLE

-- 示例2
gaussdb=# CREATE OR REPLACE PROCEDURE test_square(n NUMBER, square OUT NUMBER) IS
BEGIN
```


版权所有 © 华为云计算技术有限公司

```
  square := n * n;
END;
/
CREATE PROCEDURE

gaussdb=# DECLARE
cur NUMBER;
query varchar(2000);
ret integer;
n NUMBER;
square Integer;
BEGIN
  n := 2;
  cur := DBE_SQL.REGISTER_CONTEXT();
  query := 'BEGIN test_square(:n_bnd, :square_bnd); END;';
  DBE_SQL.SQL_SET_SQL(cur, query, 2);
  DBE_SQL.SQL_BIND_VARIABLE(cur, 'n_bnd', n);
  DBE_SQL.SQL_BIND_VARIABLE(cur, 'square_bnd', square);
  ret := DBE_SQL.SQL_RUN(cur);
  DBE_SQL.GET_VARIABLE_RESULT(cur, 'square_bnd', square);
  DBE_OUTPUT.PRINT_LINE('square = ' || square);
  DBE_SQL.SQL_UNREGISTER_CONTEXT(cur);
END;
/
square = 4
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop PROCEDURE test_square;
DROP PROCEDURE

-- 示例3
-- DESCRIBE_COLUMNS、RUN_AND_NEXT和LAST_ROW_COUNT接口示例
-- 创建表并插入数据
gaussdb=# CREATE TABLE test_desc_cols(
    id NUMBER,
    name VARCHAR2(50)
);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'id' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# INSERT INTO test_desc_cols(id, name) VALUES (1, 'xiaoming');
INSERT 0 1
gaussdb=# INSERT INTO test_desc_cols(id, name) VALUES (2, 'xiaohong');
INSERT 0 1
gaussdb=# INSERT INTO test_desc_cols(id, name) VALUES (3, 'xiaolan');
INSERT 0 1
-- 创建列描述信息打印存储过程
gaussdb=# CREATE OR REPLACE PROCEDURE print_rec(
    rec in DBE_SQL.DESC_REC
)package AS
BEGIN
    raise INFO 'col_type            =    %', rec.col_type;
    raise INFO 'col_name            =    %', rec.col_name;
    raise INFO 'col_name_len        =    %', rec.col_name_len;
END;
/
CREATE PROCEDURE

-- 进行功能验证
gaussdb=# DECLARE
context_id  INTEGER;
col_cnt     INTEGER;
rec_tab     DBE_SQL.DESC_TAB;
execute_ret  INTEGER;
nextrow_ret INTEGER;
last_row_count INTEGER;
BEGIN
-- 打开游标
```


版权所有 © 华为云计算技术有限公司

```
context_id := DBE_SQL.REGISTER_CONTEXT();
-- 编译游标
DBE_SQL.SQL_SET_SQL(context_id, 'SELECT * FROM test_desc_cols', 2);
-- 进行列描述和信息打印
DBE_SQL.DESCRIBE_COLUMNS(context_id, col_cnt, rec_tab);
FOR var IN 1..col_cnt LOOP
    print_rec(rec_tab(var));
END LOOP;
-- 执行并获取一行数据
execute_ret := DBE_SQL.RUN_AND_NEXT(context_id);
-- 获取一行数据
nextrow_ret := DBE_SQL.NEXT_ROW(context_id);
-- 得到目前已经获取的数据行数
last_row_count := DBE_SQL.LAST_ROW_COUNT();
DBE_OUTPUT.PRINT_LINE('last_row_count =  '|| last_row_count);
DBE_SQL.SQL_UNREGISTER_CONTEXT(context_id);
END;
/
INFO:  col_type           =    1700
CONTEXT:  SQL statement "CALL print_rec(rec_tab[var])"
PL/pgSQL function inline_code_block line 16 at PERFORM
INFO:  col_name           =    id
CONTEXT:  SQL statement "CALL print_rec(rec_tab[var])"
PL/pgSQL function inline_code_block line 16 at PERFORM
INFO:  col_name_len       =    2
CONTEXT:  SQL statement "CALL print_rec(rec_tab[var])"
PL/pgSQL function inline_code_block line 16 at PERFORM
INFO:  col_type           =    1043
CONTEXT:  SQL statement "CALL print_rec(rec_tab[var])"
PL/pgSQL function inline_code_block line 16 at PERFORM
INFO:  col_name           =    name
CONTEXT:  SQL statement "CALL print_rec(rec_tab[var])"
PL/pgSQL function inline_code_block line 16 at PERFORM
INFO:  col_name_len       =    4
CONTEXT:  SQL statement "CALL print_rec(rec_tab[var])"
PL/pgSQL function inline_code_block line 16 at PERFORM
last_row_count =  2
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop table test_desc_cols;
DROP TABLE

-- 示例4
gaussdb=# DROP TABLE if exists dbe_sql_tab;
NOTICE:  table "dbe_sql_tab" does not exist, skipping
DROP TABLE
gaussdb=# create table dbe_sql_tab(a char(30), b int, c text, d bytea, e text, f bool);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# insert into dbe_sql_tab values('aaa', 10, 'abcdefghijklmn', 'a', 'abcdefghijklmn', true);
INSERT 0 1

gaussdb=# DECLARE
cursorid int;
execute_ret int;
query varchar(2000);
err numeric;
v_char char(30);
v_int int;
v_long text;
v_long_len int;
v_bytea bytea;
v_text text;
BEGIN
query := 'select * from dbe_sql_tab';
cursorid := DBE_SQL.register_context();
DBE_SQL.sql_set_sql(cursorid, query, 1);
```


版权所有 © 华为云计算技术有限公司

```
DBE_SQL.SET_RESULT_TYPE_CHAR(cursorid, 1, v_char, 30);
DBE_SQL.SET_RESULT_TYPE_INT(cursorid, 2);
DBE_SQL.SET_RESULT_TYPE_LONG(cursorid, 3);
DBE_SQL.SET_RESULT_TYPE_RAW(cursorid, 4, v_bytea, 8);
DBE_SQL.SET_RESULT_TYPE_TEXT(cursorid, 5, 1024);
execute_ret := DBE_SQL.sql_run(cursorid);
loop
exit when (DBE_SQL.next_row(cursorid) <= 0);
DBE_SQL.GET_RESULT_CHAR(cursorid, 1, v_char);
v_int := DBE_SQL.GET_RESULT_INT(cursorid, 2);
DBE_SQL.GET_RESULT_LONG(cursorid, 3, 3, 3, v_long, v_long_len);
DBE_SQL.GET_RESULT_RAW(cursorid, 4, v_bytea);
v_text := DBE_SQL.GET_RESULT_TEXT(cursorid, 5);
dbe_output.print_line('a:'|| v_char);
dbe_output.print_line('b:'|| v_int);
dbe_output.print_line('c:'|| v_long);
raise info 'd:%', v_bytea;
dbe_output.print_line('e:'|| v_text);
end loop;
DBE_SQL.sql_unregister_context(cursorid);
END;
/
a:aaa
b:10
c:cde
INFO:  d:\x61
e:abcdefghijklmn
ANONYMOUS BLOCK EXECUTE

gaussdb=# DECLARE
cursorid int;
execute_ret int;
query varchar(2000);
BEGIN
query := 'select * from dbe_sql_tab';
cursorid := DBE_SQL.register_context();
DBE_SQL.sql_set_sql(cursorid, query, 1);
DBE_SQL.SET_RESULT_TYPE_UNKNOWN(cursorid, 7, 'boolean');
execute_ret := DBE_SQL.sql_run(cursorid);
loop
exit when (DBE_SQL.next_row(cursorid) <= 0);
DBE_SQL.GET_RESULT_UNKNOWN(cursorid, 7, 'boolean');
end loop;
DBE_SQL.sql_unregister_context(cursorid);
END;
/
ERROR:  UnSupport data type for set_result_type(context: 8, pos: 7, 'boolean')
CONTEXT:  SQL statement "CALL pg_catalog.report_application_error('UnSupport data type for
set_result_type(context: '||context_id||', pos: '||pos||', '||PG_CATALOG.QUOTE_LITERAL(col_type)||')')"
PL/pgSQL function dbe_sql.set_result_type_unknown(integer,integer,text) line 8 at PERFORM
SQL statement "CALL dbe_sql.set_result_type_unknown(cursorid,7,'boolean')"
PL/pgSQL function inline_code_block line 10 at PERFORM

-- 清理环境
gaussdb=# drop table dbe_sql_tab;
DROP TABLE

-- 示例5
gaussdb=# drop table if exists dbe_sql_tab;
NOTICE:  table "dbe_sql_tab" does not exist, skipping
DROP TABLE
gaussdb=# create table dbe_sql_tab(a char(30), b raw);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# insert into dbe_sql_tab values('aaa', HEXTORAW('DEADBEEF'));
INSERT 0 1

gaussdb=# DECLARE
```

 版权所有 © 华为云计算技术有限公司

```
cursorid int;
execute_ret int;
query varchar(2000);
v_char char(30);
v_raw bytea;
BEGIN
query := 'select * from dbe_sql_tab';
cursorid := DBE_SQL.register_context();
DBE_SQL.sql_set_sql(cursorid, query, 2);
DBE_SQL.SET_RESULT_TYPE(cursorid, 1, v_char);
DBE_SQL.SET_RESULT_TYPE_RAW(cursorid, 2, v_raw, 1024);
execute_ret := DBE_SQL.sql_run(cursorid);
loop
exit when (DBE_SQL.next_row(cursorid) <= 0);
v_char := DBE_SQL.DBE_SQL_GET_RESULT_CHAR(cursorid, 1);
v_raw := DBE_SQL.DBE_SQL_GET_RESULT_RAW(cursorid, 2);
dbe_output.print_line('a:'|| v_char);
raise info 'b:%', v_raw;
end loop;
DBE_SQL.sql_unregister_context(cursorid);
END;
/
a:aaa
INFO:  b:\x4445414442454546
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop table dbe_sql_tab;
DROP TABLE

-- 示例6
gaussdb=# DECLARE
cursorid int;
execute_ret int;
is_open boolean;
BEGIN
cursorid := DBE_SQL.register_context();
is_open := DBE_SQL.IS_ACTIVE(cursorid);
dbe_output.print_line('is_open:' ||is_open);
DBE_SQL.sql_unregister_context(cursorid);
is_open := DBE_SQL.IS_ACTIVE(cursorid);
dbe_output.print_line('is_open:' ||is_open);
END;
/
is_open:true
is_open:false
ANONYMOUS BLOCK EXECUTE

-- 示例7
gaussdb=# create table tbl(a integer ,b varchar(100));
OTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

gaussdb=# DECLARE
c integer;
v1 integer[];
v2 varchar2[];
query varchar(2000);
ret integer;
begin
c := dbe_sql.register_context();
query := 'insert into tbl(a,b) values(:v_1, :v_2);';
dbe_sql.sql_set_sql(c, query, 2);
v1(1) := 1;
v1(2) := 2;
v2(1) := '1';
v2(2) := '2';
dbe_sql.sql_bind_array(c, 'v_1', v1);
```


版权所有 © 华为云计算技术有限公司

```
dbe_sql.sql_bind_array(c, 'v_2', v2);
ret := dbe_sql.sql_run(c);
dbe_sql.sql_unregister_context(c);
end;
/
ANONYMOUS BLOCK EXECUTE

gaussdb=# select * from tbl order by a;
 a | b
---+---
 1 | 1
 2 | 2
(2 rows)

-- 清理环境
gaussdb=# drop table tbl;
DROP TABLE

-- 示例8
-- 前置条件
gaussdb=# drop table if exists dbe_sql_tab;
NOTICE:  table "dbe_sql_tab" does not exist, skipping
DROP TABLE
gaussdb=# create table dbe_sql_tab(a int, b text, c raw, d text, e char, f int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# insert into dbe_sql_tab values(1, '9', '5', '13', 'a', 1);
INSERT 0 1
gaussdb=# insert into dbe_sql_tab values(2, '9', '6', '14', 'b', 2);
INSERT 0 1
gaussdb=# insert into dbe_sql_tab values(3, '7', '7', '15', 'c', 3);
INSERT 0 1
gaussdb=# insert into dbe_sql_tab values(4, '6', '8', '16', 'd', 4);
INSERT 0 1

gaussdb=# DECLARE
query varchar(2000);
context_id int;
execute_ret int;
v_id int;
v_ints int[];
v_texts text[];
v_raws raw[];
v_byteas bytea[];
v_chars character[];
v_type int[];
BEGIN
query := ' select * from dbe_sql_tab order by 1';
context_id := dbe_sql.register_context();
dbe_sql.sql_set_sql(context_id, query, 1);
DBE_SQL.SET_RESULT_TYPE_INTS(context_id, 1, v_ints, 3, 1);
DBE_SQL.SET_RESULT_TYPE_TEXTS(context_id, 2, v_texts, 3, 1, 1024);
DBE_SQL.SET_RESULT_TYPE_RAWS(context_id, 3, v_raws, 3, 1, 1024);
DBE_SQL.SET_RESULT_TYPE_BYTEAS(context_id, 4, v_byteas, 3, 1, 1024);
DBE_SQL.SET_RESULT_TYPE_CHARS(context_id, 5, v_chars, 3, 1, 1024);
DBE_SQL.SET_RESULTS_TYPE(context_id, 6, v_type, 3, 1);
execute_ret := dbe_sql.sql_run(context_id);
loop
v_id := dbe_sql.next_row(context_id);
v_ints := DBE_SQL.GET_RESULTS_INT(context_id, 1, v_ints);
v_texts := DBE_SQL.GET_RESULTS_TEXT(context_id, 2, v_texts);
v_raws := DBE_SQL.GET_RESULTS_RAW(context_id, 3, v_raws);
v_byteas := DBE_SQL.GET_RESULTS_BYTEA(context_id, 4, v_byteas);
v_chars := DBE_SQL.GET_RESULTS_CHAR(context_id, 5, v_chars);
v_type := DBE_SQL.GET_RESULTS(context_id, 6, v_type);
exit when(v_id != 3);
end loop;
FOR i IN v_ints.FIRST .. v_ints.LAST  LOOP
```


版权所有 © 华为云计算技术有限公司

```
    dbe_output.print_line('int' || i || ' = ' || v_ints[i]);
END LOOP;
FOR i IN v_texts.FIRST .. v_texts.LAST  LOOP
    dbe_output.print_line('text' || i || ' = ' || v_texts[i]);
END LOOP;
FOR i IN v_raws.FIRST .. v_raws.LAST  LOOP
    dbe_output.print_line('raw' || i || ' = ' || v_raws[i]);
END LOOP;
FOR i IN v_byteas.FIRST .. v_byteas.LAST  LOOP
    dbe_output.print_line('bytea' || i || ' = ' || v_byteas[i]);
END LOOP;
FOR i IN v_chars.FIRST .. v_chars.LAST  LOOP
    dbe_output.print_line('char' || i || ' = ' || v_chars[i]);
END LOOP;
FOR i IN v_type.FIRST .. v_type.LAST  LOOP
    dbe_output.print_line('type' || i || ' = ' || v_type[i]);
END LOOP;
dbe_sql.sql_unregister_context(context_id);
END;
/
int1 = 1
int2 = 2
int3 = 3
int4 = 4
text1 = 9
text2 = 9
text3 = 7
text4 = 6
raw1 = 05
raw2 = 06
raw3 = 07
raw4 = 08
bytea1 = \x3133
bytea2 = \x3134
bytea3 = \x3135
bytea4 = \x3136
char1 = a
char2 = b
char3 = c
char4 = d
type1 = 1
type2 = 2
type3 = 3
type4 = 4
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop table if exists dbe_sql_tab;
DROP TABLE

-- 示例9
-- 前置条件
gaussdb=# drop table if exists dbe_sql_tab;
NOTICE:  table "dbe_sql_tab" does not exist, skipping
DROP TABLE
gaussdb=# create table dbe_sql_tab(a int ,b int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# insert into dbe_sql_tab values(1,3);
INSERT 0 1

gaussdb=# DECLARE
context_id int;
type re_rssc is record (col_num int, desc_col dbe_sql.desc_tab);
employer re_rssc;
res re_rssc;
d int;
dd dbe_sql.desc_tab;
```


版权所有 © 华为云计算技术有限公司

```
query varchar(2000);
BEGIN
query := 'select * from dbe_sql_tab';
--打开游标
context_id := dbe_sql.register_context();
--编译游标
dbe_sql.sql_set_sql(context_id, query, 1);
--执行
res := dbe_sql.sql_describe_columns(context_id, d,dd);
--输出结果
dbe_output.print_line('col_num:' || res.col_num);
dbe_output.print_line('col_type:' || res.desc_col[1].col_type);
dbe_output.print_line('col_max_len:' || res.desc_col[1].col_max_len);
dbe_output.print_line('col_name:' || res.desc_col[1].col_name);
dbe_output.print_line('col_name_len:' || res.desc_col[1].col_name_len);
dbe_output.print_line('col_schema_name:' || res.desc_col[1].col_schema_name);
dbe_output.print_line('col_schema_name_len:' || res.desc_col[1].col_schema_name_len);
dbe_output.print_line('col_precision:' || res.desc_col[1].col_precision);
dbe_output.print_line('col_scale:' || res.desc_col[1].col_scale);
dbe_output.print_line('col_charsetid:' || res.desc_col[1].col_charsetid);
dbe_output.print_line('col_charsetform:' || res.desc_col[1].col_charsetform);
dbe_output.print_line('col_null_ok:' || res.desc_col[1].col_null_ok);
--关闭游标
dbe_sql.sql_unregister_context(context_id);
END;
/
col_num:2
col_type:23
col_max_len:4
col_name:a
col_name_len:1
col_schema_name:
col_schema_name_len:0
col_precision:0
col_scale:0
col_charsetid:0
col_charsetform:0
col_null_ok:true
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop table if exists dbe_sql_tab;
DROP TABLE

-- 示例10
gaussdb=# drop table if exists dbe_sql_tab;
NOTICE:  table "dbe_sql_tab" does not exist, skipping
DROP TABLE
gaussdb=# create table dbe_sql_tab(a int);
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE
gaussdb=# insert into dbe_sql_tab values(1);
INSERT 0 1
gaussdb=# insert into dbe_sql_tab values(2);
INSERT 0 1
gaussdb=# insert into dbe_sql_tab values(3);
INSERT 0 1

gaussdb=# DECLARE
query varchar(2000);
context_id int;
execute_ret int;
v_id int;
v_ints int[];
i1 integer;
BEGIN
query := 'select * from dbe_sql_tab order by 1';
context_id := dbe_sql.register_context();
```


版权所有 © 华为云计算技术有限公司

```
dbe_sql.sql_set_sql(context_id, query, 1);
DBE_SQL.SQL_SET_RESULTS_TYPE_C(context_id, 1, v_ints, 3, 1, i1, 0);
execute_ret := dbe_sql.sql_run(context_id);
loop
v_id := dbe_sql.next_row(context_id);
v_ints := DBE_SQL.SQL_GET_VALUES_C(context_id, 1, v_ints, i1);
exit when(v_id != 3);
end loop;
FOR i IN v_ints.FIRST .. v_ints.LAST  LOOP
        dbe_output.print_line('int' || i || ' = ' || v_ints[i]);
END LOOP;
dbe_sql.sql_unregister_context(context_id);
END;
/
int1 = 1
int2 = 2
int3 = 3
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop table if exists dbe_sql_tab;
DROP TABLE

-- 示例11
gaussdb=# CREATE OR REPLACE PROCEDURE test_proc(out_int out Integer, out_char out char, out_raw out
raw, out_text out text) IS
BEGIN
  out_int := 1;
  out_char := 'a';
  out_raw := 'b';
  out_text := 'c';
END;
/
CREATE PROCEDURE

gaussdb=# DECLARE
cur NUMBER;
query varchar(2000);
ret integer;
v_int Integer;
v_char char;
v_raw raw;
v_text text;
BEGIN
  cur := DBE_SQL.REGISTER_CONTEXT();
  query := 'BEGIN test_proc(:v_int, :v_char, :v_raw, :v_text); END;';
  DBE_SQL.SQL_SET_SQL(cur, query, 2);
  DBE_SQL.SQL_BIND_VARIABLE(cur, 'v_int', v_int);
  DBE_SQL.SQL_BIND_VARIABLE(cur, 'v_char', v_char, 1024);
  DBE_SQL.SQL_BIND_VARIABLE(cur, 'v_raw', v_raw, 1024);
  DBE_SQL.SQL_BIND_VARIABLE(cur, 'v_text', v_text, 1024);
  ret := DBE_SQL.SQL_RUN(cur);
  DBE_SQL.GET_VARIABLE_RESULT_INT(cur, 'v_int', v_int);
  v_char := DBE_SQL.GET_VARIABLE_RESULT_CHAR(cur, 'v_char');
  DBE_SQL.GET_VARIABLE_RESULT_RAW(cur, 'v_raw', v_raw);
  v_text := DBE_SQL.GET_VARIABLE_RESULT_TEXT(cur, 'v_text');
  DBE_OUTPUT.PRINT_LINE('v_int = ' || v_int);
  DBE_OUTPUT.PRINT_LINE('v_char = ' || v_char);
  DBE_OUTPUT.PRINT_LINE('v_raw = ' || v_raw);
  DBE_OUTPUT.PRINT_LINE('v_text = ' || v_text);
  DBE_SQL.SQL_UNREGISTER_CONTEXT(cur);
END;
/
v_int = 1
v_char = a
v_raw = 0B
v_text = c
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

```
-- 清理环境
gaussdb=# drop procedure test_proc;
DROP PROCEDURE

-- 示例12
gaussdb=# CREATE OR REPLACE PROCEDURE test_proc(out_int out Integer[], out_char out char[], out_raw
out raw[], out_text out text[]) IS
BEGIN
  out_int(0) := 1;
  out_char(0) := 'a';
  out_raw(0) := 'b';
  out_text(0) := 'c';
END;
/
CREATE PROCEDURE

gaussdb=# DECLARE
cur NUMBER;
query varchar(2000);
ret integer;
v_int Integer[];
v_char char[];
v_raw raw[];
v_text text[];
BEGIN
  cur := DBE_SQL.REGISTER_CONTEXT();
  query := 'call test_proc(:v_int, :v_char, :v_raw, :v_text);';
  DBE_SQL.SQL_SET_SQL(cur, query, 1);
  DBE_SQL.SQL_BIND_ARRAY(cur, 'v_int', v_int);
  DBE_SQL.SQL_BIND_ARRAY(cur, 'v_char', v_char);
  DBE_SQL.SQL_BIND_ARRAY(cur, 'v_raw', v_raw);
  DBE_SQL.SQL_BIND_ARRAY(cur, 'v_text', v_text);
  ret := DBE_SQL.SQL_RUN(cur);
  DBE_SQL.GET_ARRAY_RESULT_INT(cur, 'v_int', v_int);
  DBE_SQL.GET_ARRAY_RESULT_CHAR(cur, 'v_char', v_char);
  DBE_SQL.GET_ARRAY_RESULT_RAW(cur, 'v_raw', v_raw);
  DBE_SQL.GET_ARRAY_RESULT_TEXT(cur, 'v_text', v_text);
  DBE_OUTPUT.PRINT_LINE('v_int = ' || v_int(0));
  DBE_OUTPUT.PRINT_LINE('v_char = ' || v_char(0));
  DBE_OUTPUT.PRINT_LINE('v_raw = ' || v_raw(0));
  DBE_OUTPUT.PRINT_LINE('v_text = ' || v_text(0));
  DBE_SQL.SQL_UNREGISTER_CONTEXT(cur);
END;
/
v_int = 1
v_char = a
v_raw = 0B
v_text = c
ANONYMOUS BLOCK EXECUTE

-- 清理环境
gaussdb=# drop PROCEDURE test_proc;
DROP PROCEDURE
```

2.12.2.18 DBE_STATS

高级包DBE_STATS提供了统计信息管理的功能，包括锁定、解锁、导出、导入、设置、回退历史统计信息等。DBE_STATS支持的接口请参见**表2-369**。

版权所有 © 华为云计算技术有限公司

说明

- DBE_STATS不支持临时表。
- DBE_STATS统计信息管理规则同现有的统计信息管理规则一致，仅支持内核已有统计信息相关功能。
- DBE_STATS和多版本统计信息管理不支持M兼容模式。
- DBE_STATS不支持删除统计信息接口。
- 多版本统计信息管理功能要求相关GUC参数所有节点一致：
- 要求历史统计信息保留时间stats_history_retention_time、历史统计信息保留条数上限stats_history_record_limit所有节点设置一致。如果不同节点GUC参数值不一致，会导致不同节点同一对象的历史统计信息不一致，回退时可能在某个节点查找不到指定时间戳的历史统计信息，进而导致回退失败。
- 扩容节点后使用回退接口，指定扩容前的时间戳回退时，新增加节点复用了CN的元数据，历史统计信息也属于元数据之一，后续按照该节点的历史统计信息进行统计信息回退。
- 统计信息的修改（导入、回退、设置）会导致计划缓存失效，每个缓存的执行计划会记录其依赖的统计信息版本，如果当前统计信息版本与缓存计划记录的版本不一致，则判定缓存计划失效，需重新生成。

接口介绍

表 2-369 DBE_STATS

接口名称	描述	权限要求
DBE_STATS.LOCK_TABLE_STATS	锁定表级统计信息。	使用该接口时，用户需要对指定的表拥有与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。
DBE_STATS.LOCK_PARTITION_STATS	锁定分区级统计信息。	
DBE_STATS.LOCK_COLUMN_STATS	锁定列级统计信息。	
DBE_STATS.LOCK_SCHEMA_STATS	锁定指定Schema下所有表相关的统计信息。	• 调用该接口时，会锁定该Schema下所有当前用户有ANALYZE权限的表，对于没有权限的表直接跳过。 • 调用该接口需拥有指定Schema的USAGE权限。
DBE_STATS.UNLOCK_TABLE_STATS	解锁表级统计信息。	使用该接口时，用户需要对指定的表拥有与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。
DBE_STATS.UNLOCK_PARTITION_STATS	解锁分区级统计信息。	
DBE_STATS.UNLOCK_COLUMN_STATS	解锁列级统计信息。	

版权所有 © 华为云计算技术有限公司

接口名称	描述	权限要求
DBE_STATS.UNLOCK_SCHEMA_STATS	解锁指定Schema下所有表相关的统计信息。	• 调用该接口时，会解锁该Schema下所有当前用户有ANALYZE权限的表，对于没有权限的表直接跳过。 • 调用该接口需拥有指定Schema的USAGE权限。
DBE_STATS.RESTORE_TABLE_STATS	回退表级统计信息到指定时间点。	使用该接口时，用户需要对指定的表拥有与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。
DBE_STATS.RESTORE_PARTITION_STATS	回退分区级统计信息到指定时间点。	
DBE_STATS.RESTORE_COLUMN_STATS	回退列级统计信息到指定时间点。	
DBE_STATS.RESTORE_SCHEMA_STATS	回退指定Schema下所有表相关的统计信息到指定时间点。	• 调用该接口时，会回退该Schema下所有当前用户有ANALYZE权限的表，对于没有权限的表直接跳过。 • 调用该接口需拥有指定Schema的USAGE权限。
DBE_STATS.PURGE_STATS	清除指定时间节点前的所有历史统计信息。	只有初始用户可调用该接口。
DBE_STATS.GET_STATS_HISTORY_RETENTION	获取历史统计信息的保留时间。	-
DBE_STATS.GET_STATS_HISTORY_AVAILABILITY	获取最早可用的历史统计信息的时间。	-
DBE_STATS.CREATE_STAT_TABLE	创建用于保存统计信息的统计表。	要调用此过程创建统计表，需要用户拥有在指定Schema与指定tablespace下创建表和创建索引的权限。
DBE_STATS.DROP_STAT_TABLE	删除用于保存统计信息的统计表。	要调用此过程删除统计表，需要用户拥有在指定Schema下删除表和删除索引的权限。

版权所有 © 华为云计算技术有限公司

接口名称	描述	权限要求
DBE_STATS.EXPORT_INDEX_STATS	检索指定索引的统计信息并存储在用户统计信息表中。	• 使用该接口时，用户需要对指定的表拥有与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。 • 调用此过程将统计信息导出到用户统计信息表，需要对用户统计信息表具有SELECT、INSERT、DELETE和UPDATE的权限。
DBE_STATS.EXPORT_TABLE_STATS	检索指定表的统计信息并存储在用户统计信息表中。	
DBE_STATS.EXPORT_COLUMN_STATS	检索指定列的统计信息并存储在用户统计信息表中。	
DBE_STATS.EXPORT_SCHEMA_STATS	检索指定schema的统计信息并存储在用户统计信息表中。	
DBE_STATS.EXPORT_DATABASE_STATS	检索当前database的统计信息并存储在用户统计信息表中。	• 调用该接口时，会导出当前database下所有当前用户有ANALYZE权限的用户表，对于没有权限的表直接跳过。 • 调用此过程将统计信息导出到用户统计信息表，需要对用户统计信息表具有SELECT、INSERT、DELETE和UPDATE的权限。
DBE_STATS.IMPORT_INDEX_STATS	从用户统计信息表中检索指定索引的统计信息，并将其写回系统表。	• 使用该接口时，用户需要对指定的表拥有与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。 • 调用此过程将统计信息从用户统计信息表导回系统表，需要对用户统计信息表具有SELECT、INSERT、DELETE和UPDATE的权限。
DBE_STATS.IMPORT_TABLE_STATS	从用户统计信息表中检索指定表的统计信息，并将其写回系统表。	
DBE_STATS.IMPORT_COLUMN_STATS	从用户统计信息表中检索指定列的统计信息，并将其写回系统表。	
DBE_STATS.IMPORT_SCHEMA_STATS	从用户统计信息表中检索指定schema的统计信息，并将其写回系统表。	

版权所有 © 华为云计算技术有限公司

接口名称	描述	权限要求
DBE_STATS.IMPORT_DATABASE_STATS	从用户统计信息表中检索指定database的统计信息，并将其写回系统表。	• 调用该接口时，导入当前database下所有当前用户有ANALYZE权限用户表的统计信息，对于没有权限的表直接跳过。 • 调用此过程将统计信息导出到用户统计信息表，需要对用户统计信息表具有SELECT、INSERT、DELETE和UPDATE的权限。
DBE_STATS.SET_COLUMN_STATS	设置列相关的统计信息，包括单列、多列、表达式统计信息。	使用该接口时，用户需要对指定的表拥有与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。
DBE_STATS.SET_INDEX_STATS	设置索引相关的统计信息。	
DBE_STATS.SET_TABLE_STATS	设置表相关的统计信息。	
DBE_STATS.GET_TABLES_BY_SQLID	获取Unique SQL ID对应语句所有涉及的表，如果包含多张表，则返回多行记录。	使用该接口时，用户的权限需要满足以下其中一种情况： • 用户是初始用户。 • 用户拥有MONADMIN访问权限。
DBE_STATS.ANALYZE_SQLID	获取Unique SQL ID对应语句所有涉及的表，以及收集对应表的统计信息。对于收集成功的表，如果包含多张表，则返回多行记录。	使用该接口时，用户的权限需要满足以下其中一种情况： • 用户是初始用户。 • 用户拥有MONADMIN访问权限，同时拥有相应表与ANALYZE命令相同的权限。有关权限的详细信息，请参见ANALYZE | ANALYSE。

版权所有 © 华为云计算技术有限公司

接口名称	描述	权限要求
dbe_stats_tools.py	用于库级统计信息导出和导入，以及计划调优语句元信息导出。	• 导出统计信息时，需要具备相应接口或命令（DBE_STATS.CREATE_STAT_TABLE、DBE_STATS.EXPORT_DATABASE_STATS、\COPY TO、DBE_STATS.DROP_STAT_TABLE）的权限，以及指定导出目录的读写权限。 • 导入统计信息时，需要具备相应接口或命令（DBE_STATS.CREATE_STAT_TABLE、DBE_STATS.EXPORT_DATABASE_STATS、\COPY FROM、DBE_STATS.DROP_STAT_TABLE）的权限，以及指定导出目录下文件的读取权限。 • 导出语句元信息时，需要数据库的系统管理员权限，以及指定目录的读写权限。

- DBE_STATS.LOCK_TABLE_STATS

  锁定一个表的统计信息。锁定后，该表的统计信息无法更新，该表下级索引也会同步被锁定。该接口无返回值。

  DBE_STATS.LOCK_TABLE_STATS原型为：

```
DBE_STATS.LOCK_TABLE_STATS(
  ownname   VARCHAR2,
  tabname   VARCHAR2
);
```

  表 2-370 DBE_STATS.LOCK_TABLE_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	要锁定的表所属的Schema名称，该参数为空时，默认为当前Schema。
tabname	varchar2	否	要锁定的表的名称。

版权所有 © 华为云计算技术有限公司

说明

- 可以通过查看表的reloptions中的stat_state状态获取该表的锁定状态。
- 当一个表处于锁定状态时，其统计信息无法更新。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 建表。
gaussdb=# CREATE SCHEMA dbe_stats_lock;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA=dbe_stats_lock;
SET
gaussdb=# CREATE TABLE t1(a int,b int) distribute by hash(a);
CREATE TABLE

-- 锁定表，查看其锁定状态。
gaussdb=# CALL DBE_STATS.LOCK_TABLE_STATS(ownname=>'dbe_stats_lock',tabname=>'t1');
 lock_table_stats
------------------

(1 row)
gaussdb=# SELECT relname,instr(reloptions::text,'stat_state=locked',1,1) <> 0  AS exist_lock FROM
PG_CLASS WHERE relname='t1' AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_lock');
 relname | exist_lock
---------+------------
 t1      | t
(1 row)

-- 锁定后analyze, 发生报错。
gaussdb=# ANALYZE t1;
ERROR:  The statistics is locked, cannot be updated.

-- 删除表、删除命名空间。
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock;
DROP SCHEMA
```

- DBE_STATS.LOCK_PARTITION_STATS

  锁定一个分区的统计信息。锁定后，该分区的统计信息无法更新，该分区下级索引也会同步锁定。该接口无返回值。

  DBE_STATS.LOCK_PARTITION_STATS原型为：

```
DBE_STATS.LOCK_PARTITION_STATS(
    ownname     VARCHAR2,
    tabname     VARCHAR2,
    partname    VARCHAR2
);
```

表 2-371 DBE_STATS.LOCK_PARTITION_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	要锁定的分区所属的Schema名称，该参数为空时，默认为当前Schema。
tabname	varchar2	否	要锁定的分区所属的表的名称。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
partname	varchar2	否	要锁定的分区的名称。

说明

- 可以通过查看该分区的reloptions中的stat_state状态获取该表的锁定状态。
- 当该分区或该表所属的表处于锁定状态时，其统计信息无法更新。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 建表。
gaussdb=# CREATE SCHEMA dbe_stats_lock;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA=dbe_stats_lock;
SET
gaussdb=# CREATE TABLE upart_table(a int, b int, c int) distribute by hash(a) PARTITION BY
RANGE(a)
 (
    PARTITION p1 VALUES LESS THAN(1200),
    PARTITION p2 VALUES LESS THAN(2400),
    PARTITION p3 VALUES LESS THAN(MAXVALUE)
 );
CREATE TABLE

-- 锁定一个分区，其他分区及表不受影响。
gaussdb=# CALL
DBE_STATS.LOCK_PARTITION_STATS(ownname=>'dbe_stats_lock',tabname=>'upart_table',partname=>'
p1');
 lock_partition_stats
----------------------

(1 row)

gaussdb=# SELECT relname,instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock FROM
PG_CLASS WHERE relname='upart_table';
   relname   | exist_lock
-------------+------------
 upart_table | f
(1 row)

gaussdb=# SELECT relname,instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock FROM
PG_PARTITION WHERE parentid='upart_table'::REGCLASS ORDER BY oid;
   relname   | exist_lock
-------------+------------
 upart_table | f
 p1          | t
 p2          | f
 p3          | f
(4 rows)
-- 删除表、命名空间。
gaussdb=# DROP TABLE upart_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock;
DROP SCHEMA
```

- DBE_STATS.LOCK_COLUMN_STATS

  锁定一个列的统计信息。锁定后，该列的统计信息无法更新。该接口无返回值。

版权所有 © 华为云计算技术有限公司

DBE_STATS.LOCK_COLUMN_STATS原型为：

```
DBE_STATS.LOCK_COLUMN_STATS(
    ownname     VARCHAR2,
    tabname     VARCHAR2,
    colname     VARCHAR2,
    partname    VARCHAR2 DEFAULT NULL
);
```

表 2-372 DBE_STATS.LOCK_COLUMN_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	要锁定的列所属的Schema的名称，该参数为空时，默认为当前Schema。
tabname	varchar2	否	要锁定的列所属的表的名称。
colname	varchar2	否	要锁定的列的名称。
partname	varchar2	是	要锁定的列所属的分区的名称，默认为空。

说明

- 当partname为空时，默认锁定该列表级统计信息。
- 若不存在该表的列级统计信息，则锁定可执行成功，但实际未生效。
- 可查看**PG_STATISTIC**、**PG_STATISTIC_EXT**中的stastate获取锁定状态。
- 锁定多列时，需先查询**PG_STATISTIC_EXT**中的staextname字段，获取多列别名作为colname参数传入。
- 当某一列被锁定或其所属的表或分区被锁定时，其统计信息无法更新，该列被锁定的信息不会回显，需要在日志中通过DEBUG2级别的DBE_STATS的logging_module查看。
- 当ownname参数值设置为null时使用当前schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 建表，收集统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_lock;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA=dbe_stats_lock;
SET
gaussdb=# CREATE TABLE t1(a int,b int) distribute by hash(a);
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(generate_series(1,100),1);
INSERT 0 100
gaussdb=# ANALYZE t1;
ANALYZE

-- 锁定列后，查看列的锁定状态。
gaussdb=# CALL
DBE_STATS.LOCK_COLUMN_STATS(ownname=>'dbe_stats_lock',tabname=>'t1',colname=>'a');
 lock_column_stats
-------------------

(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT staattnum,stastate FROM PG_STATISTIC WHERE starelid='t1'::REGCLASS ORDER BY
1,2;
 staattnum | stastate
-----------+----------
         1 | l
         2 | u
(2 rows)

-- 删除表、命名空间。
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock;
DROP SCHEMA
```

- DBE_STATS.LOCK_SCHEMA_STATS

  锁定一个指定Schema下所有表、表下级索引的统计信息。该接口无返回值。

  DBE_STATS.LOCK_SCHEMA_STATS原型为：

```
DBE_STATS.LOCK_SCHEMA_STATS(
   ownname    VARCHAR2
);
```

表 2-373 DBE_STATS.LOCK_SCHEMA_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	指定的Schema名称，该参数为空时，默认为当前Schema。

说明

当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_lock_schema CASCADE;
NOTICE:  schema "dbe_stats_lock_schema" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表
gaussdb=# CREATE SCHEMA dbe_stats_lock_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_lock_schema;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by hash(a);
CREATE TABLE
gaussdb=# CREATE TABLE t2(a int, b int) distribute by hash(a);
CREATE TABLE
gaussdb=# -- 锁定schema
gaussdb=# CALL DBE_STATS.LOCK_SCHEMA_STATS(ownname=>'dbe_stats_lock_schema');
 lock_schema_stats
-------------------

(1 row)

gaussdb=# -- 验证: t1和t2都应被锁定 (exist_lock = t)
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_CLASS
gaussdb-# WHERE relname IN ('t1','t2')
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_lock_schema')
gaussdb-# ORDER BY 1;
 relname | exist_lock
```


版权所有 © 华为云计算技术有限公司

```
---------+------------
 t1      | t
 t2      | t
(2 rows)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE t2;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock_schema;
DROP SCHEMA
```

- DBE_STATS.UNLOCK_TABLE_STATS

  解锁一个表的统计信息。解锁后，该表的统计信息可以更新。该表下级索引会同步解锁。该接口无返回值。

  DBE_STATS.UNLOCK_TABLE_STATS原型为：

```
DBE_STATS.UNLOCK_TABLE_STATS(
  ownname   VARCHAR2,
  tabname   VARCHAR2
);
```

表 2-374 DBE_STATS.UNLOCK_TABLE_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	要解锁的表所属的Schema的名称，该参数为空时，默认为当前Schema。
tabname	varchar2	否	要解锁的表的名称。

说明

- 可以通过查看该表的reloptions中的stat_state状态获取该表的锁定状态。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_lock_unlock_table CASCADE;
NOTICE:  schema "dbe_stats_lock_unlock_table" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表
gaussdb=# CREATE SCHEMA dbe_stats_lock_unlock_table;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_lock_unlock_table;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by hash(a);
CREATE TABLE
gaussdb=# -- 先锁定表
gaussdb=# CALL DBE_STATS.LOCK_TABLE_STATS(ownname=>'dbe_stats_lock_unlock_table',
tabname=>'t1');
 lock_table_stats
------------------


(1 row)

gaussdb=# -- 验证已锁定
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_CLASS
```


版权所有 © 华为云计算技术有限公司

```
gaussdb-# WHERE relname='t1'
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_lock_unlock_table');
 relname | exist_lock
---------+------------
 t1      | t
(1 row)

gaussdb=# -- 解锁表
gaussdb=# CALL DBE_STATS.UNLOCK_TABLE_STATS(ownname=>'dbe_stats_lock_unlock_table',
tabname=>'t1');
 unlock_table_stats
--------------------


(1 row)

gaussdb=# -- 验证已解锁 (exist_lock = f)
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_CLASS
gaussdb-# WHERE relname='t1'
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_lock_unlock_table');
 relname | exist_lock
---------+------------
 t1      | f
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock_unlock_table;
DROP SCHEMA
```

- DBE_STATS.UNLOCK_PARTITION_STATS

  解锁一个分区的统计信息。解锁后，该分区的统计信息可以更新。该分区下级索引会同步解锁。该接口无返回值。

  DBE_STATS.UNLOCK_PARTITION_STATS原型为：

```
DBE_STATS.UNLOCK_PARTITION_STATS(
    ownname     VARCHAR2,
    tabname     VARCHAR2,
    partname    VARCHAR2
);
```

表 2-375 DBE_STATS.UNLOCK_PARTITION_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	要解锁的分区所属的Schema的名称，该参数为空时，默认为当前Schema。
tabname	varchar2	否	要解锁的分区所属的表的名称。
partname	varchar2	否	要解锁的分区的名称。

版权所有 © 华为云计算技术有限公司

说明

- 可查看该分区的reloptions中stat_state状态获取该表的锁定状态。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_lock_unlock_partition CASCADE;
NOTICE:  schema "dbe_stats_lock_unlock_partition" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建分区表
gaussdb=# CREATE SCHEMA dbe_stats_lock_unlock_partition;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_lock_unlock_partition;
SET
gaussdb=# CREATE TABLE upart_table(a int, b int, c int) distribute by hash(a) PARTITION BY
RANGE(a)
gaussdb-# (
gaussdb(#     PARTITION p1 VALUES LESS THAN(1200),
gaussdb(#     PARTITION p2 VALUES LESS THAN(2400),
gaussdb(#     PARTITION p3 VALUES LESS THAN(MAXVALUE)
gaussdb(# );
CREATE TABLE
gaussdb=# -- 先锁定p1
gaussdb=# CALL DBE_STATS.LOCK_PARTITION_STATS(ownname=>'dbe_stats_lock_unlock_partition',
tabname=>'upart_table', partname=>'p1');
 lock_partition_stats
----------------------

(1 row)

gaussdb=# -- 验证p1已锁定
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_PARTITION
gaussdb-# WHERE parentid='upart_table'::REGCLASS
gaussdb-# ORDER BY 1;
   relname   | exist_lock
-------------+------------
 p1          | t
 p2          | f
 p3          | f
 upart_table | f
(4 rows)

gaussdb=# -- 解锁p1
gaussdb=# CALL
DBE_STATS.UNLOCK_PARTITION_STATS(ownname=>'dbe_stats_lock_unlock_partition',
tabname=>'upart_table', partname=>'p1');
 unlock_partition_stats
------------------------

(1 row)

gaussdb=# -- 验证p1已解锁 (全部 exist_lock = f)
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_PARTITION
gaussdb-# WHERE parentid='upart_table'::REGCLASS
gaussdb-# ORDER BY 1;
   relname   | exist_lock
-------------+------------
 p1          | f
 p2          | f
 p3          | f
 upart_table | f
(4 rows)

gaussdb=# -- 清理
gaussdb=# DROP TABLE upart_table;
```


版权所有 © 华为云计算技术有限公司

```
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock_unlock_partition;
DROP SCHEMA
```

- DBE_STATS.UNLOCK_COLUMN_STATS

  解锁一个列的统计信息。解锁后，该列的统计信息可以更新。该接口无返回值。

  DBE_STATS.UNLOCK_COLUMN_STATS原型为：

```
DBE_STATS.UNLOCK_COLUMN_STATS(
   ownname     VARCHAR2,
   tabname     VARCHAR2,
   colname     VARCHAR2,
   partname    VARCHAR2 DEFAULT NULL
);
```

表 2-376 DBE_STATS.UNLOCK_COLUMN_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	要解锁的列所属的Schema的名称，该参数为空时，默认为当前Schema。
tabname	varchar2	否	要解锁的列所属的表的名称。
colname	varchar2	否	要解锁的列的名称。
partname	varchar2	是	要解锁的列所属的分区的名称，默认为空。

说明

- 当partname为空时，默认解锁该列表级统计信息，以及解锁其级联分区列级统计信息。
- 可查看**PG_STATISTIC**、**PG_STATISTIC_EXT**中的stastate状态获取锁定状态。
- 解锁多列时，需先查询**PG_STATISTIC_EXT**中的staextname字段，获取多列别名作为colname参数传入。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_lock_unlock_column CASCADE;
NOTICE:  schema "dbe_stats_lock_unlock_column" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_lock_unlock_column;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_lock_unlock_column;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by hash(a);
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(generate_series(1,100), 1);
INSERT 0 100
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 先锁定列a
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL DBE_STATS.LOCK_COLUMN_STATS(ownname=>'dbe_stats_lock_unlock_column',
tabname=>'t1', colname=>'a');
 lock_column_stats
-------------------

(1 row)

gaussdb=# -- 验证列a已锁定 (stastate='l')
gaussdb=# SELECT staattnum, stastate
gaussdb-# FROM PG_STATISTIC
gaussdb-# WHERE starelid='t1'::REGCLASS
gaussdb-# ORDER BY 1, 2;
 staattnum | stastate
-----------+----------
         1 | l
         2 | u
(2 rows)

gaussdb=# -- 解锁列a
gaussdb=# CALL DBE_STATS.UNLOCK_COLUMN_STATS(ownname=>'dbe_stats_lock_unlock_column',
tabname=>'t1', colname=>'a');
 unlock_column_stats
---------------------

(1 row)

gaussdb=# -- 验证列a已解锁 (stastate='u')
gaussdb=# SELECT staattnum, stastate
gaussdb-# FROM PG_STATISTIC
gaussdb-# WHERE starelid='t1'::REGCLASS
gaussdb-# ORDER BY 1, 2;
 staattnum | stastate
-----------+----------
         1 | u
         2 | u
(2 rows)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock_unlock_column;
DROP SCHEMA
```

- DBE_STATS.UNLOCK_SCHEMA_STATS

  解锁一个指定Schema下所有表、表下级索引的统计信息。该接口无返回值。

  DBE_STATS.UNLOCK_SCHEMA_STATS原型为：

```
DBE_STATS.UNLOCK_SCHEMA_STATS(
   ownname    VARCHAR2
);
```

表 2-377 DBE_STATS.UNLOCK_SCHEMA_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	是	指定的Schema名称，该参数为空时，默认为当前Schema。

说明

当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_lock_unlock_schema CASCADE;
NOTICE:  schema "dbe_stats_lock_unlock_schema" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表
gaussdb=# CREATE SCHEMA dbe_stats_lock_unlock_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_lock_unlock_schema;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by hash(a);
CREATE TABLE
gaussdb=# CREATE TABLE t2(a int, b int) distribute by hash(a);
CREATE TABLE
gaussdb=# -- 先锁定schema
gaussdb=# CALL DBE_STATS.LOCK_SCHEMA_STATS(ownname=>'dbe_stats_lock_unlock_schema');
 lock_schema_stats
-------------------

(1 row)

gaussdb=# -- 验证已锁定 (exist_lock = t)
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_CLASS
gaussdb-# WHERE relname IN ('t1','t2')
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_lock_unlock_schema')
gaussdb-# ORDER BY 1;
 relname | exist_lock
---------+------------
 t1      | t
 t2      | t
(2 rows)

gaussdb=# -- 解锁schema
gaussdb=# CALL DBE_STATS.UNLOCK_SCHEMA_STATS(ownname=>'dbe_stats_lock_unlock_schema');
 unlock_schema_stats
---------------------

(1 row)

gaussdb=# -- 验证已解锁 (exist_lock = f)
gaussdb=# SELECT relname, instr(reloptions::text,'stat_state=locked',1,1) <> 0 AS exist_lock
gaussdb-# FROM PG_CLASS
gaussdb-# WHERE relname IN ('t1','t2')
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_lock_unlock_schema')
gaussdb-# ORDER BY 1;
 relname | exist_lock
---------+------------
 t1      | f
 t2      | f
(2 rows)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE t2;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_lock_unlock_schema;
DROP SCHEMA
```

- DBE_STATS.RESTORE_TABLE_STATS

  将表的统计信息回退到指定的时间节点，选择该时间节点前最后一次收集的统计信息载入系统表，同时级联其下级未被锁定的分区、索引、列级统计信息。该接口无返回值。

  DBE_STATS.RESTORE_TABLE_STATS原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_STATS.RESTORE_TABLE_STATS(
   ownname                 VARCHAR2,
   tabname                 VARCHAR2,
   as_of_timestamp           TIMESTAMPTZ,
   restore_cluster_index     BOOLEAN DEFAULT NULL,
   force                   BOOLEAN DEFAULT FALSE,
   no_invalidate             BOOLEAN DEFAULT NULL
);
```

表 2-378 DBE_STATS.RESTORE_TABLE_STATS 接口参数介绍

参数	类型	是否允许为空	描述
ownname	VARCHAR2	是	要回退的表所属的Schema名称，该参数为空时，默认为当前Schema。
tabname	VARCHAR2	否	要回退的表的名称。
as_of_timestamp	timestamp with time zone	否	要回退的目标时间节点。
restore_cluster_index	BOOLEAN	是	暂不支持。
force	BOOLEAN	是	是否在统计信息被锁定时强行回退的标志位，默认为false。
no_invalidate	BOOLEAN	是	暂不支持。

说明

- 调用该接口时，同时回退其下级分区、索引、列的统计信息，如果其下级的分区或列被锁定，则直接跳过。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_restore_table CASCADE;
NOTICE:  schema "dbe_stats_restore_table" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集两次统计信息
gaussdb=# CREATE SCHEMA dbe_stats_restore_table;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_restore_table;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 查看历史表 (应有两条记录: reltuples=3 和 reltuples=6)
gaussdb=# SELECT relname, reltuples FROM GS_TABLESTATS_HISTORY WHERE relid='t1'::regclass
ORDER BY 1, 2;
 relname | reltuples
---------+-----------
 t1      |         3
 t1      |         6
(2 rows)

gaussdb=# -- 查看当前系统表 (reltuples=6)
gaussdb=# SELECT relname, reltuples FROM PG_CLASS
gaussdb-# WHERE relname='t1'
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_table');
 relname | reltuples
---------+-----------
 t1      |         6
(1 row)

gaussdb=# -- 回退到最早的统计信息
gaussdb=# CALL DBE_STATS.RESTORE_TABLE_STATS(
gaussdb(#     ownname=>'dbe_stats_restore_table',
gaussdb(#     tabname=>'t1',
gaussdb(#     as_of_timestamp=>(SELECT MIN(reltimestamp) FROM GS_TABLESTATS_HISTORY
WHERE relname='t1'
gaussdb(#     AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_table'))
gaussdb(# );
 restore_table_stats
---------------------


(1 row)

gaussdb=# -- 验证: 回退后 reltuples=3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS
gaussdb-# WHERE relname='t1'
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_table');
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_restore_table;
DROP SCHEMA
```

- DBE_STATS.RESTORE_PARTITION_STATS

  将分区的统计信息回退到指定的时间节点，选择该时间节点前最后一次收集的统计信息载入系统表，同时级联其下级未被锁定的分区、索引、列级统计信息。该接口无返回值。

  DBE_STATS.RESTORE_PARTITION_STATS原型为：

```
DBE_STATS.RESTORE_PARTITION_STATS(
    ownname                 VARCHAR2,
    tabname                 VARCHAR2,
    partname                VARCHAR2,
    as_of_timestamp           TIMESTAMPTZ,
    force                   BOOLEAN DEFAULT FALSE,
    no_invalidate           BOOLEAN DEFAULT NULL
);
```


版权所有 © 华为云计算技术有限公司

表 2-379 DBE_STATS.RESTORE_PARTITION_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	是	要回退的分区所属的Schema名称。
tabname	VARCHAR2	否	要回退的分区所属的表名。
partname	VARCHAR2	否	要回退的分区名。
as_of_timestamp	timestamp with time zone	否	要回退的目标时间节点。
force	BOOLEAN	是	是否在统计信息被锁定时强行回退的标志位。
no_invalidate	BOOLEAN	是	暂不支持。

说明

- 调用该接口时，同时回退其下级分区、索引、列的统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_restore_partition CASCADE;
NOTICE:  schema "dbe_stats_restore_partition" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建分区表，收集两次分区统计信息
gaussdb=# CREATE SCHEMA dbe_stats_restore_partition;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_restore_partition;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication PARTITION BY RANGE(a)
gaussdb-# (
gaussdb(#     PARTITION p1 VALUES LESS THAN(10),
gaussdb(#     PARTITION p2 VALUES LESS THAN(MAXVALUE)
gaussdb(# );
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# ANALYZE t1 PARTITION (p1);
ANALYZE
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# ANALYZE t1 PARTITION (p1);
ANALYZE
gaussdb=# -- 查看历史表 (应有分区的历史记录)
gaussdb=# SELECT relname, relkind, reltuples FROM GS_TABLESTATS_HISTORY
gaussdb-# WHERE relname='p1' AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_partition')
```


版权所有 © 华为云计算技术有限公司

```
gaussdb-# ORDER BY reltuples;
 relname | relkind | reltuples
---------+---------+-----------
 p1      | p       |         3
 p1      | p       |         6
(2 rows)

gaussdb=# -- 查看当前分区统计信息 (reltuples=6)
gaussdb=# SELECT relname, reltuples FROM PG_PARTITION
gaussdb-# WHERE relname='p1' AND parentid = (SELECT oid FROM PG_CLASS WHERE relname='t1'
AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_partition'));
 relname | reltuples
---------+-----------
 p1      |         6
(1 row)

gaussdb=# -- 回退分区p1到最早的统计信息
gaussdb=# CALL DBE_STATS.RESTORE_PARTITION_STATS(
gaussdb(#     ownname=>'dbe_stats_restore_partition',
gaussdb(#     tabname=>'t1',
gaussdb(#     partname=>'p1',
gaussdb(#     as_of_timestamp=>(SELECT MIN(reltimestamp) FROM GS_TABLESTATS_HISTORY
WHERE relname='p1' AND
gaussdb(#     relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_partition'))
gaussdb(# );
 restore_partition_stats
-------------------------

(1 row)

gaussdb=# -- 验证回退结果 (reltuples应回退到3)
gaussdb=# SELECT relname, reltuples FROM PG_PARTITION
gaussdb-# WHERE relname='p1' AND parentid = (SELECT oid FROM PG_CLASS WHERE relname='t1'
AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_partition'));
 relname | reltuples
---------+-----------
 p1      |         3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_restore_partition;
DROP SCHEMA
```

- DBE_STATS.RESTORE_COLUMN_STATS

  将列的统计信息回退到指定的时间节点，选择该时间节点前最后一次收集的统计信息载入系统表。该接口无返回值。

  DBE_STATS.RESTORE_COLUMN_STATS原型为：

```
DBE_STATS.RESTORE_COLUMN_STATS(
    ownname              VARCHAR2,
    tabname              VARCHAR2,
    colname              VARCHAR2,
    partname             VARCHAR2 DEFAULT NULL,
    as_of_timestamp      TIMESTAMPTZ,
    force                BOOLEAN DEFAULT FALSE,
    no_invalidate        BOOLEAN DEFAULT NULL
);
```


版权所有 © 华为云计算技术有限公司

表 2-380 DBE_STATS.RESTORE_COLUMN_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	是	要回退的列所属的Schema名。
tabname	VARCHAR2	否	要回退的列所属的表名。
colname	VARCHAR2	否	要回退的列名。
partname	VARCHAR2	是	要回退的列所属的分区名。
as_of_timestamp	timestamp with time zone	否	要回退的目标时间节点。
force	BOOLEAN	是	是否在统计信息被锁定时强行回退的标志位。
no_invalidate	BOOLEAN	是	暂不支持。

说明

- 当partname为空时，默认回退该列表及统计信息。
- 回退多列时，需先查询**PG_STATISTIC_EXT**中的staextname字段，获取多列别名作为colname参数传入。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_restore_column CASCADE;
NOTICE:  schema "dbe_stats_restore_column" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集两次统计信息
gaussdb=# CREATE SCHEMA dbe_stats_restore_column;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_restore_column;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 确保两次ANALYZE时间戳间隔>1秒
gaussdb=# SELECT pg_sleep(1.1);
 pg_sleep
----------

(1 row)

gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
```


版权所有 © 华为云计算技术有限公司

```
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 查看历史表里的统计信息
gaussdb=# SELECT staattnum, stadistinct, stakind1, stanumbers1, stavalues1
gaussdb-# FROM GS_STATISTIC_HISTORY
gaussdb-# WHERE starelid='t1'::REGCLASS
gaussdb-# ORDER BY statimestamp;
 staattnum | stadistinct | stakind1 | stanumbers1 | stavalues1
-----------+-------------+----------+-------------+------------
         1 |    -.333333 |        1 | {1}         | {1}
         2 |    -.333333 |        1 | {1}         | {1}
         1 |    -.333333 |        1 | {.5,.5}     | {1,2}
         2 |    -.333333 |        1 | {.5,.5}     | {1,2}
(4 rows)

gaussdb=# SELECT staattnum, stadistinct, stakind1, stanumbers1, stavalues1
gaussdb-# FROM PG_STATISTIC
gaussdb-# WHERE starelid='t1'::REGCLASS
gaussdb-# ORDER BY 1;
 staattnum | stadistinct | stakind1 | stanumbers1 | stavalues1
-----------+-------------+----------+-------------+------------
         1 |    -.333333 |        1 | {.5,.5}     | {1,2}
         2 |    -.333333 |        1 | {.5,.5}     | {1,2}
(2 rows)

gaussdb=# -- 回退列a到时间较早的时间节点
gaussdb=# CALL DBE_STATS.RESTORE_COLUMN_STATS(
gaussdb(#     ownname=>'dbe_stats_restore_column',
gaussdb(#     tabname=>'t1',
gaussdb(#     colname=>'a',
gaussdb(#     as_of_timestamp=>((SELECT MIN(reltimestamp) FROM GS_TABLESTATS_HISTORY
WHERE relname='t1'
gaussdb(#     AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_column')) + INTERVAL '1 second')
gaussdb(# );
 restore_column_stats
----------------------

(1 row)

gaussdb=# -- 验证回退后列a的统计信息 (stanumbers1={1}, stavalues1={1})
gaussdb=# SELECT staattnum, stadistinct, stakind1, stanumbers1, stavalues1
gaussdb-# FROM PG_STATISTIC
gaussdb-# WHERE starelid='t1'::REGCLASS
gaussdb-# ORDER BY 1;
 staattnum | stadistinct | stakind1 | stanumbers1 | stavalues1
-----------+-------------+----------+-------------+------------
         1 |    -.333333 |        1 | {1}         | {1}
         2 |    -.333333 |        1 | {.5,.5}     | {1,2}
(2 rows)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_restore_column;
DROP SCHEMA
```

- DBE_STATS.RESTORE_SCHEMA_STATS

  将指定schema下所有表的统计信息回退到指定时间节点。该接口无返回值。

  DBE_STATS.RESTORE_SCHEMA_STATS原型为：

```
DBE_STATS.RESTORE_SCHEMA_STATS(
   ownname                 VARCHAR2,
   as_of_timestamp           TIMESTAMPTZ,
   force                   BOOLEAN DEFAULT FALSE,
```


版权所有 © 华为云计算技术有限公司

```
    no_invalidate           BOOLEAN DEFAULT NULL
);
```

表 2-381 DBE_STATS.RESTORE_SCHEMA_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	是	指定的Schema名称。
as_of_timestamp	timestamp with time zone	否	要回退的目标时间节点。
force	BOOLEAN	是	是否在统计信息被锁定时强行回退的标志位。
no_invalidate	BOOLEAN	是	暂不支持。

说明

当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_restore_schema CASCADE;
NOTICE:  schema "dbe_stats_restore_schema" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，插数据，收集两次统计信息
gaussdb=# CREATE SCHEMA dbe_stats_restore_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_restore_schema;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 查看历史表
gaussdb=# SELECT relname, reltuples FROM GS_TABLESTATS_HISTORY WHERE relid='t1'::regclass
ORDER BY 1, 2;
 relname | reltuples
---------+-----------
 t1      |         3
 t1      |         6
(2 rows)

gaussdb=# -- 查看当前系统表 (reltuples=6)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT relname, reltuples FROM PG_CLASS
gaussdb-# WHERE relname='t1'
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_schema');
 relname | reltuples
---------+-----------
 t1      |         6
(1 row)

gaussdb=# -- 回退schema
gaussdb=# CALL DBE_STATS.RESTORE_SCHEMA_STATS(
gaussdb(#     ownname=>'dbe_stats_restore_schema',
gaussdb(#     as_of_timestamp=>(SELECT MIN(reltimestamp) FROM GS_TABLESTATS_HISTORY
WHERE relname='t1'
gaussdb(#     AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_schema'))
gaussdb(# );
 restore_schema_stats
----------------------

(1 row)

gaussdb=# -- 验证: 回退后 reltuples=3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS
gaussdb-# WHERE relname='t1'
gaussdb-#   AND relnamespace = (SELECT oid FROM PG_NAMESPACE WHERE
nspname='dbe_stats_restore_schema');
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_restore_schema;
DROP SCHEMA
```

- DBE_STATS.PURGE_STATS

  清除指定时间节点之前的所有历史统计信息。该接口无返回值。

  DBE_STATS.PURGE_STATS原型为：

```
DBE_STATS.PURGE_STATS(
    before_timestamp        TIMESTAMPTZ
);
```

表 2-382 DBE_STATS.PURGE_STATS 接口参数说明

参数	类型	是否允许为空	描述
before_timestamp	timestamp with time zone	否	指定要清除的时间节点。

示例：

```
-- 建表，收集两次统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_purge;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA=dbe_stats_purge;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
```

 版权所有 © 华为云计算技术有限公司

```
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(1,1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES(2,2);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE

-- 查看历史表。
gaussdb=# SELECT relname,reltuples FROM GS_TABLESTATS_HISTORY WHERE relname='t1' ORDER
BY 1,2;
 relname | reltuples
---------+-----------
 t1      |         3
 t1      |         6
(2 rows)

-- 清除时间较早的历史统计信息，查看历史表。
gaussdb=# CALL DBE_STATS.PURGE_STATS(before_timestamp=>((SELECT MIN(reltimestamp) FROM
GS_TABLESTATS_HISTORY WHERE relname='t1') + INTERVAL '1 second'));
 purge_stats
-------------


(1 row)

gaussdb=# SELECT relname,reltuples FROM GS_TABLESTATS_HISTORY WHERE relname='t1';
 relname | reltuples
---------+-----------
 t1      |         6
(1 row)

-- 删除表、命名空间。
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_purge;
DROP SCHEMA
```

- DBE_STATS.GET_STATS_HISTORY_RETENTION

  获得当前历史统计信息的保留时间，返回用GUC参数stats_history_retention_time的值，表示历史统计信息的保留时间。该接口没有入参。

  DBE_STATS.GET_STATS_HISTORY_RETENTION原型为：

  ```
  DBE_STATS.GET_STATS_HISTORY_RETENTION()
  returns NUMBER;
  ```

  示例：

  ```
  gaussdb=# CALL DBE_STATS.GET_STATS_HISTORY_RETENTION();
   get_stats_history_retention
  -----------------------------
                            31
  (1 row)
  ```

- DBE_STATS.GET_STATS_HISTORY_AVAILABILITY

  获得当前历史统计信息的最早可用时间，返回当前数据库下所有历史统计信息的最早时间。该接口没有入参。

  DBE_STATS.GET_STATS_HISTORY_AVAILABILITY原型为：

  ```
  DBE_STATS.GET_STATS_HISTORY_AVAILABILITY()
  returns TIMESTAMPTZ;
  ```

  示例：

版权所有 © 华为云计算技术有限公司

```
gaussdb=# ANALYZE;
ANALYZE
gaussdb=# CALL DBE_STATS.GET_STATS_HISTORY_AVAILABILITY();
 get_stats_history_availability
--------------------------------
 2025-05-07 05:11:43.584349-04
(1 row)
```

说明

DBE_STATS高级包接口仅对普通表有效。

- DBE_STATS.CREATE_STAT_TABLE

  创建用于保存用户统计信息的统计表。该接口无返回值。

  DBE_STATS.CREATE_STAT_TABLE原型为：

```
DBE_STATS.CREATE_STAT_TABLE (
   ownname          VARCHAR2,
   stattab          VARCHAR2,
   tblspace         VARCHAR2 DEFAULT NULL,
   global_temporary BOOLEAN DEFAULT NULL
);
```

表 2-383 DBE_STATS.CREATE_STAT_TABLE 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定创建的统计表所在的Schema。如果设置为null，默认在当前所在的Schema创建。
stattab	VARCHAR2	否	统计表的名称。
tblspace	VARCHAR2	是	指定创建的统计表所在的表空间。如果不指定，则使用默认表空间进行创建。
global_temporary	BOOLEAN	是	暂不支持该参数。

说明

- 为了区分普通表和统计表，在创建统计表后，其reloptions中的字段statstable将不为空串。且考虑到统计表结构可能进行版本升级，为实现前向兼容，statstable将被赋予当前统计信息表的版本信息。
- 该表创建时会自动生成一个索引gs_stattab_<name_hash_value>_statid_type_nameinfo_index用于索引扫描，其中<name_hash_value>为该统计信息表名的哈希值。如若被删除，则导入导出export、import接口不可用。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 在dbe_stats_create_drop_schema下创建一个stattab统计表，并手动插入一条数据，可从新建的用户统计信息表中查询到此条数据。
gaussdb=# CREATE SCHEMA dbe_stats_create_drop_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_create_drop_schema;
SET
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE(ownname => 'dbe_stats_create_drop_schema',
stattab => 'stat_tab');
 create_stat_table
-------------------
(1 row)
gaussdb=# INSERT INTO stat_tab(nodeid, statid, type, relname) VALUES(0, '123', 't', 't1');
INSERT 0 1
gaussdb=# SELECT statid, type, relname FROM stat_tab;
 statid | type | relname
--------+------+---------
 123    | t    | t1
(1 row)
```

- DBE_STATS.DROP_STAT_TABLE

  删除用户统计信息表。该接口无返回值。

  DBE_STATS.DROP_STAT_TABLE原型为：

```
DBE_STATS.DROP_STAT_TABLE (
    ownname VARCHAR2,
    stattab VARCHAR2
);
```

表 2-384 DBE_STATS.DROP_STAT_TABLE 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	统计表所在的Schema。如果设置为null，则默认使用当前Schema。
stattab	VARCHAR2	否	统计表的名称。

说明

- 对于statstable属性为空的普通表，无法使用此函数删除。
- 如果在用户统计信息表上已创建索引或者已创建其他依赖于该表的对象，则在调用此函数时，将级联删除所有依赖于该表的对象。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 删除用户统计信息表。将上述所建的用户统计信息表删除，删除新建的Schema。
gaussdb=# CALL DBE_STATS.DROP_STAT_TABLE(ownname => 'dbe_stats_create_drop_schema',
stattab => 'stat_tab');
 drop_stat_table
-----------------
(1 row)
gaussdb=# DROP SCHEMA dbe_stats_create_drop_schema CASCADE;
DROP SCHEMA
```

- DBE_STATS.EXPORT_INDEX_STATS

  在系统表中查找指定索引的统计信息，并将结果存储在用户统计信息表中。该接口无返回值。

  DBE_STATS.EXPORT_INDEX_STATS原型为：

```
DBE_STATS.EXPORT_INDEX_STATS (
    ownname  VARCHAR2,
    indname  VARCHAR2,
    partname VARCHAR2 DEFAULT NULL,
    stattab  VARCHAR2,
```


版权所有 © 华为云计算技术有限公司

```
  statid   VARCHAR2 DEFAULT NULL,
  statown  VARCHAR2 DEFAULT NULL
);
```

表 2-385 DBE_STATS.EXPORT_INDEX_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定索引所在Schema的名称，如果设置为null，会默认使用当前的Schema。
indname	VARCHAR2	否	指定需要导出统计信息的索引名。
partname	VARCHAR2	是	分区索引的名称。如果索引已分区但是partname为null，则导出全局和分区索引统计信息。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。

说明

- 导出索引的reltuples（numrows）、relpages（numlblks）和relallvisible（relallvisible）三类统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 建表，收集统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_export;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_export;
SET
gaussdb=# CREATE TABLE t2(a int, b int, c int) distribute by replication PARTITION BY RANGE(a)
SUBPARTITION BY RANGE(b)
 (
   PARTITION p1 VALUES LESS THAN(200)
   (SUBPARTITION s1 VALUES LESS THAN (500),
    SUBPARTITION s2 VALUES LESS THAN (MAXVALUE)
   ),
   PARTITION p2 VALUES LESS THAN(500)
   (SUBPARTITION s3 VALUES LESS THAN (500),
    SUBPARTITION s4 VALUES LESS THAN (MAXVALUE)
   ),
   PARTITION p3 VALUES LESS THAN(MAXVALUE)
   ( SUBPARTITION s5 VALUES LESS THAN (500),
     SUBPARTITION s6 VALUES LESS THAN (MAXVALUE)
   )
 );
CREATE TABLE
gaussdb=# CREATE INDEX idx_t2_local_a ON t2(a) LOCAL
```


版权所有 © 华为云计算技术有限公司

```
(
    PARTITION p1_idx_a
    (
        SUBPARTITION s1_idx_a,
        SUBPARTITION s2_idx_a
    ),
    PARTITION p2_idx_a
    (
        SUBPARTITION s3_idx_a,
        SUBPARTITION s4_idx_a
    ),
    PARTITION p3_idx_a
    (
        SUBPARTITION s5_idx_a,
        SUBPARTITION s6_idx_a
    ));
CREATE INDEX
gaussdb=# INSERT INTO t2 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (100, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (300, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (300, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (600, 400, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (600, 600, 1);
INSERT 0 1
gaussdb=# ANALYZE t2 WITH ALL;
ANALYZE
gaussdb=# ANALYZE t2((a,b)) WITH ALL;
ANALYZE
-- 建立用户统计信息表，作为统计信息导出的目标表。
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_export', 'export_table');
 create_stat_table
-------------------
(1 row)

-- 导出索引统计信息，连同分区索引的统计信息一并导出。
-- 查看当前的统计信息。
gaussdb=# SELECT reltuples FROM PG_CLASS WHERE relname='idx_t2_local_a';
 reltuples
-----------
         6
(1 row)
gaussdb=# SELECT relname,relpages,reltuples FROM PG_PARTITION WHERE
parentid='idx_t2_local_a'::REGCLASS ORDER BY oid;
 relname  | relpages | reltuples
----------+----------+-----------
 s1_idx_a |        1 |         0
 s2_idx_a |        1 |         0
 s3_idx_a |        1 |         0
 s4_idx_a |        1 |         0
 s5_idx_a |        1 |         0
 s6_idx_a |        1 |         0
(6 rows)
-- 导出统计信息，查看用户统计信息表中的数据。
gaussdb=# CALL DBE_STATS.EXPORT_INDEX_STATS(ownname => 'dbe_stats_export', indname =>
'idx_t2_local_a', partname=> null, stattab => 'export_table', statid => 'idx_s1');
 export_index_stats
--------------------
(1 row)
gaussdb=# SELECT relname,partname,numrows,numblocks FROM export_table WHERE nodeid = 1
ORDER BY relname, partname desc;
    relname     | partname | numrows | numblocks
----------------+----------+---------+-----------
 idx_t2_local_a |          |       6 |         6
 idx_t2_local_a | s6_idx_a |       0 |         1
```


版权所有 © 华为云计算技术有限公司

```
 idx_t2_local_a | s5_idx_a |       0 |         1
 idx_t2_local_a | s4_idx_a |       0 |         1
 idx_t2_local_a | s3_idx_a |       0 |         1
 idx_t2_local_a | s2_idx_a |       0 |         1
 idx_t2_local_a | s1_idx_a |       0 |         1
(7 rows)
-- 删除表、命名空间。
gaussdb=# DROP INDEX idx_t2_local_a;
DROP INDEX
gaussdb=# DROP TABLE t2;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_export;
DROP SCHEMA
```

- DBE_STATS.EXPORT_TABLE_STATS

  检索指定表的统计信息，并存储在用户统计信息表中。该接口无返回值。

  DBE_STATS.EXPORT_TABLE_STATS原型为：

```
DBE_STATS.EXPORT_TABLE_STATS (
  ownname           VARCHAR2,
  tabname           VARCHAR2,
  partname          VARCHAR2 DEFAULT NULL,
  stattab           VARCHAR2,
  statid            VARCHAR2 DEFAULT NULL,
  cascade           BOOLEAN  DEFAULT TRUE,
  statown           VARCHAR2 DEFAULT NULL,
  stat_category     VARCHAR2 DEFAULT NULL
);
```

表 2-386 DBE_STATS.EXPORT_TABLE_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
tabname	VARCHAR2	否	指定需要导出统计信息的表名。
partname	VARCHAR2	是	表分区的名称。如果表已分区但是partname为null，则导出全局和分区表统计信息。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
cascade	BOOLEAN	是	是否一并导出列、索引、多列和索引表达式的统计信息。默认为TRUE。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。
stat_category	VARCHAR2	是	暂不支持。

版权所有 © 华为云计算技术有限公司

说明

- 导出表的reltuples（numrows）、relpages（numblks）和relallvisible（relallvisible）三类统计信息。级联导出列级、索引级的统计信息参考**EXPORT_INDEX_STATS**和**EXPORT_COLUMN_STATS**接口的统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_export_table CASCADE;
NOTICE:  schema "dbe_stats_export_table" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_export_table;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_export_table;
SET
gaussdb=# CREATE TABLE t1(a int, b int, c int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 建立用户统计信息表
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_export_table', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# -- 导出表统计信息 (级联导出列、索引统计信息)
gaussdb=# CALL DBE_STATS.EXPORT_TABLE_STATS(
gaussdb(#     ownname => 'dbe_stats_export_table',
gaussdb(#     tabname => 't1',
gaussdb(#     partname => null,
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 't1_t',
gaussdb(#     cascade => true
gaussdb(# );
 export_table_stats
--------------------

(1 row)

gaussdb=# -- 验证: 用户统计信息表中应有数据
gaussdb=# SELECT count(*) AS row_count FROM export_table WHERE relname='t1' and nodeid=0;
 row_count
-----------
         4
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_export_table;
DROP SCHEMA
```

- DBE_STATS.EXPORT_COLUMN_STATS

  检索指定列（多列/表达式）的统计信息并存储在用户统计信息表中。该接口无返回值。

  DBE_STATS.EXPORT_COLUMN_STATS原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_STATS.EXPORT_COLUMN_STATS (
   ownname  VARCHAR2,
   tabname  VARCHAR2,
   colname  VARCHAR2,
   partname VARCHAR2 DEFAULT NULL,
   stattab  VARCHAR2,
   statid   VARCHAR2 DEFAULT NULL,
   statown  VARCHAR2 DEFAULT NULL
);
```

表 2-387 DBE_STATS.EXPORT_COLUMN_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
tabname	VARCHAR2	否	指定需要导出统计信息的列所属的表名（表达式对应索引名）。
colname	VARCHAR2	否	列名或者多列名（tabname为索引时，colname为表达式名）。
partname	VARCHAR2	是	分区的名称。如果表/索引已分区但是partname为null，则导出全局和分区列/多列/索引表达式统计信息。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。

说明

- 导出列级的统计信息可查询**PG_STATISTIC**和**PG_STATISTIC_EXT**。
- 导出多列统计信息时，需先查询**PG_STATISTIC_EXT**中的staextname字段，获取多列别名作为colname参数传入。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 建表，收集统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_export;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA=dbe_stats_export;
SET
gaussdb=# CREATE TABLE IF NOT EXISTS t1 (a int, b int, c int) distribute by replication;
CREATE TABLE
gaussdb=# CREATE INDEX idx_t1_expr on t1((a*a), (a+a), (a*b));
CREATE INDEX
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# ANALYZE t1((a,b));
ANALYZE
-- 建立用户统计信息表。
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_export', 'export_table');
 create_stat_table
-------------------
(1 row)

-- 查看列级统计信息。
gaussdb=# SELECT staattnum,stadistinct,stakind1,stanumbers1,stavalues1 FROM PG_STATISTIC
WHERE starelid='t1'::REGCLASS ORDER BY 1;
 staattnum | stadistinct | stakind1 | stanumbers1 |  stavalues1
-----------+-------------+----------+-------------+---------------
         1 |          -1 |        2 |             | {100,150,250}
         2 |          -1 |        2 |             | {150,300,550}
         3 |    -.333333 |        1 | {1}         | {1}
(3 rows)
-- 导出列级统计信息，查看用户统计信息表。
gaussdb=# CALL DBE_STATS.EXPORT_COLUMN_STATS(ownname => 'dbe_stats_export', tabname =>
't1', colname=> 'a', partname => null, stattab => 'export_table', statid => 't1_c');
 export_column_stats
---------------------
(1 row)
gaussdb=# SELECT relname,columnname,stadistinct,stakind1,stanumbers1,stavalues1 FROM
export_table WHERE nodeid=0;
 relname | columnname | stadistinct | stakind1 | stanumbers1 |  stavalues1
---------+------------+-------------+----------+-------------+---------------
 t1      | a          |          -1 |        2 |             | {100,150,250}
(1 row)

-- 查看表达式统计信息。
gaussdb=# SELECT staattnum,stadistinct,stakind1,stanumbers1,stavalues1 FROM PG_STATISTIC
WHERE starelid='idx_t1_expr'::REGCLASS ORDER BY 1;
 staattnum | stadistinct | stakind1 | stanumbers1 |     stavalues1
-----------+-------------+----------+-------------+---------------------
         1 |          -1 |        2 |             | {10000,22500,62500}
         2 |          -1 |        2 |             | {200,300,500}
         3 |          -1 |        2 |             | {30000,37500,82500}
(3 rows)
-- 导出表达式统计信息，查看用户统计信息表。
gaussdb=# CALL DBE_STATS.EXPORT_COLUMN_STATS(ownname => 'dbe_stats_export', tabname =>
'idx_t1_expr', colname => 'expr', stattab => 'export_table', statid => 't1_expr');
 export_column_stats
---------------------
(1 row)
gaussdb=# SELECT relname,columnname,stadistinct,stakind1,stanumbers1,stavalues1 FROM
export_table where statid='t1_expr' AND nodeid=0;
   relname   | columnname | stadistinct | stakind1 | stanumbers1 |     stavalues1
-------------+------------+-------------+----------+-------------+---------------------
 idx_t1_expr | expr       |          -1 |        2 |             | {10000,22500,62500}
(1 row)

-- 导出多列统计信息。
-- 查看多列统计信息。
gaussdb=# SELECT stakey,staextname,stadistinct,stakind1,stanumbers1,stavalues1 FROM
PG_STATISTIC_EXT WHERE starelid='t1'::REGCLASS;
 stakey |     staextname      | stadistinct | stakind1 | stanumbers1 | stavalues1
--------+---------------------+-------------+----------+-------------+------------
 1 2    | extname_-1308656218 |          -1 |        8 |             |
(1 row)
-- 导出，查看用户统计信息表。
gaussdb=# CALL DBE_STATS.EXPORT_COLUMN_STATS(ownname => 'dbe_stats_export', tabname =>
't1', colname=> 'extname_-1308656218', partname => null, stattab => 'export_table', statid => 't1_c');
 export_column_stats
---------------------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
gaussdb=# SELECT relname,columnname,stadistinct,stakind1,stanumbers1,stavalues1 FROM
export_table WHERE nodeid=0 AND relname='t1';
 relname |     columnname      | stadistinct | stakind1 | stanumbers1 |  stavalues1
---------+---------------------+-------------+----------+-------------+---------------
 t1      | a                   |          -1 |        2 |             | {100,150,250}
 t1      | extname_-1308656218 |          -1 |        8 |             |
(2 rows)
-- 删除表、命名空间。
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_export;
DROP SCHEMA
```

- DBE_STATS.EXPORT_SCHEMA_STATS

  用于检索指定Schema的统计信息，并存储在用户统计信息表中。该接口无返回值。

  DBE_STATS.EXPORT_SCHEMA_STATS原型为：

```
DBE_STATS.EXPORT_SCHEMA_STATS (
   ownname          VARCHAR2,
   stattab          VARCHAR2,
   statid           VARCHAR2 DEFAULT NULL,
   statown          VARCHAR2 DEFAULT NULL,
   stat_category    VARCHAR2 DEFAULT NULL
);
```

表 2-388 DBE_STATS.EXPORT_SCHEMA_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。
stat_category	VARCHAR2	是	暂不支持。

说明

- 导出当前用户在指定Schema下所有具有权限的表的统计信息，对于无权限的表直接跳过，不报错。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_export_schema CASCADE;
NOTICE:  schema "dbe_stats_export_schema" does not exist, skipping
```


版权所有 © 华为云计算技术有限公司

```
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_export_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_export_schema;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (2,2);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 建立用户统计信息表
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_export_schema', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# -- 导出schema统计信息
gaussdb=# CALL DBE_STATS.EXPORT_SCHEMA_STATS(
gaussdb(#     ownname => 'dbe_stats_export_schema',
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 's1'
gaussdb(# );
 export_schema_stats
---------------------

(1 row)

gaussdb=# -- 验证: 用户统计信息表中应有数据
gaussdb=# SELECT relname,numrows FROM export_table where nodeid=0 and numrows != 0;
 relname | numrows
---------+---------
 t1      |       2
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_export_schema;
DROP SCHEMA
```

- DBE_STATS.EXPORT_DATABASE_STATS

  用于检索指定database的统计信息，并存储在用户统计信息表中。该接口无返回值。

  DBE_STATS.EXPORT_DATABASE_STATS原型：

```
DBE_STATS.EXPORT_DATABASE_STATS (
   stattab         VARCHAR2,
   statid          VARCHAR2 DEFAULT NULL,
   statown         VARCHAR2 DEFAULT NULL,
   stat_category   VARCHAR2 DEFAULT NULL
);
```

表 2-389 DBE_STATS.EXPORT_DATABASE_STATS 接口参数说明

参数	类型	是否允许为空	描述
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用当前Schema。
stat_category	VARCHAR2	是	暂不支持。

说明

- 导出当前用户在当前database下所有具有analyze权限用户表的统计信息，对于无权限的表直接跳过，不报错。
- 当statown参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_export_database CASCADE;
NOTICE:  schema "dbe_stats_export_database" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_export_database;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_export_database;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (2,2);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 建立用户统计信息表
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_export_database', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# -- 导出database统计信息
gaussdb=# CALL DBE_STATS.EXPORT_DATABASE_STATS(
gaussdb(#     statown => 'dbe_stats_export_database',
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 's1'
gaussdb(# );
 export_database_stats
-----------------------

(1 row)

gaussdb=# -- 验证: 用户统计信息表中应有数据
gaussdb=# SELECT count(*) > 0 FROM export_table;
 ?column?
----------
 t
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT count(*) > 0 FROM export_table WHERE relname='t1';
 ?column?
----------
 t
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_export_database;
DROP SCHEMA
```

- DBE_STATS.IMPORT_INDEX_STATS

  从stattab标识的用户统计信息表中检索指定索引的统计信息，并将其写回系统表中。该接口无返回值。

  DBE_STATS.IMPORT_INDEX_STATS原型为：

```
DBE_STATS.IMPORT_INDEX_STATS (
   ownname       VARCHAR2,
   indname       VARCHAR2,
   partname      VARCHAR2 DEFAULT NULL,
   stattab       VARCHAR2,
   statid        VARCHAR2 DEFAULT NULL,
   statown       VARCHAR2 DEFAULT NULL,
   no_invalidate BOOLEAN DEFAULT NULL,
   force         BOOLEAN DEFAULT FALSE
);
```

表 2-390 DBE_STATS.IMPORT_INDEX_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
indname	VARCHAR2	否	指定需要导入统计信息的索引名。
partname	VARCHAR2	是	索引分区的名称。如果索引已分区但是partname为null，则导入全局和分区索引统计信息。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。
no_invalidate	BOOLEAN	是	暂不支持。
force	BOOLEAN	是	是否强制导入，忽略锁的状态，默认false。

版权所有 © 华为云计算技术有限公司

说明

- 导入索引的reltuples（numrows）、relpages（numlblks）和relallvisible（relallvisible）三类统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 建表，收集统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_import;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_import;
SET
gaussdb=# CREATE TABLE t2(a int, b int, c int)distribute by replication PARTITION BY RANGE(a)
SUBPARTITION BY RANGE(b)
 (
   PARTITION p1 VALUES LESS THAN(200)
   (SUBPARTITION s1 VALUES LESS THAN (500),
    SUBPARTITION s2 VALUES LESS THAN (MAXVALUE)
   ),
   PARTITION p2 VALUES LESS THAN(500)
   (SUBPARTITION s3 VALUES LESS THAN (500),
    SUBPARTITION s4 VALUES LESS THAN (MAXVALUE)
   ),
   PARTITION p3 VALUES LESS THAN(MAXVALUE)
   ( SUBPARTITION s5 VALUES LESS THAN (500),
     SUBPARTITION s6 VALUES LESS THAN (MAXVALUE)
   )
 );
CREATE TABLE
gaussdb=# CREATE INDEX idx_t2_local_a ON t2(a) LOCAL
 (
    PARTITION p1_idx_a
    (
       SUBPARTITION s1_idx_a,
       SUBPARTITION s2_idx_a
    ),
    PARTITION p2_idx_a
    (
       SUBPARTITION s3_idx_a,
       SUBPARTITION s4_idx_a
    ),
    PARTITION p3_idx_a
    (
       SUBPARTITION s5_idx_a,
       SUBPARTITION s6_idx_a
    ));
CREATE INDEX
gaussdb=# INSERT INTO t2 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (100, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (300, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (300, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (600, 400, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (600, 600, 1);
INSERT 0 1
gaussdb=# ANALYZE t2 WITH ALL;
ANALYZE
-- 创建用户统计信息表，先将数据导出。
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_import', 'export_table');
 create_stat_table
-------------------
(1 row)
gaussdb=# CALL DBE_STATS.EXPORT_INDEX_STATS(ownname => 'dbe_stats_import', indname =>
```


版权所有 © 华为云计算技术有限公司

```
'idx_t2_local_a', partname=> null, stattab => 'export_table', statid => 'idx_s1');
 export_index_stats
--------------------
(1 row)
gaussdb=# SELECT relname,partname,numrows,numblocks FROM export_table WHERE nodeid=1
ORDER BY relname,partname desc;
    relname     | partname | numrows | numblocks
----------------+----------+---------+-----------
 idx_t2_local_a |          |       6 |         6
 idx_t2_local_a | s6_idx_a |       0 |         1
 idx_t2_local_a | s5_idx_a |       0 |         1
 idx_t2_local_a | s4_idx_a |       0 |         1
 idx_t2_local_a | s3_idx_a |       0 |         1
 idx_t2_local_a | s2_idx_a |       0 |         1
 idx_t2_local_a | s1_idx_a |       0 |         1
(7 rows)
-- 再次插入数据，收集统计信息，查看统计信息。
gaussdb=# INSERT INTO t2 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (100, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (100, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (300, 600, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (600, 400, 1);
INSERT 0 1
gaussdb=# INSERT INTO t2 VALUES (600, 600, 1);
INSERT 0 1
gaussdb=# ANALYZE t2 WITH ALL;
ANALYZE
gaussdb=#  SELECT reltuples FROM PG_CLASS WHERE relname='idx_t2_local_a';
 reltuples
-----------
        12
(1 row)
gaussdb=# SELECT relname,relpages,reltuples FROM PG_PARTITION WHERE
parentid='idx_t2_local_a'::REGCLASS ORDER BY oid;
 relname  | relpages | reltuples
----------+----------+-----------
 s1_idx_a |        1 |         0
 s2_idx_a |        1 |         0
 s3_idx_a |        1 |         0
 s4_idx_a |        1 |         0
 s5_idx_a |        1 |         0
 s6_idx_a |        1 |         0
(6 rows)
-- 将用户统计表中的统计信息导入统计信息系统表。
gaussdb=# CALL DBE_STATS.IMPORT_INDEX_STATS(ownname => 'dbe_stats_import', indname =>
'idx_t2_local_a', stattab => 'export_table', statid => 'idx_s1');
 import_index_stats
--------------------
(1 row)
gaussdb=# SELECT reltuples FROM PG_CLASS WHERE relname='idx_t2_local_a';
 reltuples
-----------
         6
(1 row)
gaussdb=# SELECT relname,relpages,reltuples FROM PG_PARTITION WHERE
parentid='idx_t2_local_a'::REGCLASS ORDER BY oid;
 relname  | relpages | reltuples
----------+----------+-----------
 s1_idx_a |        1 |         0
 s2_idx_a |        1 |         0
 s3_idx_a |        1 |         0
 s4_idx_a |        1 |         0
 s5_idx_a |        1 |         0
 s6_idx_a |        1 |         0
(6 rows)
```


版权所有 © 华为云计算技术有限公司

```
-- 删除表、命名空间。
gaussdb=# DROP TABLE t2;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_import;
DROP SCHEMA
```

- DBE_STATS.IMPORT_TABLE_STATS

  从stattab标识的用户统计信息表中检索指定表的统计信息，并将其写回系统表中。该接口无返回值。

  DBE_STATS.IMPORT_TABLE_STATS原型为：

```
DBE_STATS.IMPORT_TABLE_STATS (
  ownname        VARCHAR2,
  tabname        VARCHAR2,
  partname       VARCHAR2 DEFAULT NULL,
  stattab        VARCHAR2,
  statid         VARCHAR2 DEFAULT NULL,
  cascade        BOOLEAN  DEFAULT TRUE,
  statown        VARCHAR2 DEFAULT NULL,
  no_invalidate BOOLEAN  DEFAULT NULL,
  force          BOOLEAN  DEFAULT FALSE,
  stat_category VARCHAR2 DEFAULT NULL
);
```

表 2-391 DBE_STATS.IMPORT_TABLE_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	varchar2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
tabname	varchar2	否	指定需要导入统计信息的表名。
partname	varchar2	是	表分区的名称。如果表已分区但是partname为null，则导入全局和分区表统计信息。
stattab	varchar2	否	指定存储统计信息用户表的表名。
statid	varchar2	是	指向stattab中统计信息的标识符（可选）。
cascade	boolean	是	是否一并导入列、索引、多列和索引表达式的统计信息。默认为true。
statown	varchar2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。
no_invalidate	boolean	是	暂不支持。
force	boolean	是	是否强制导入，忽略锁的状态，默认false。
stat_category	varchar2	是	暂不支持。

版权所有 © 华为云计算技术有限公司

说明

- 导入表的reltuples（numrows）、relpages（numblks）和relallvisible（relallvisible）三类统计信息。级联导入列级、索引级的统计信息参考**IMPORT_INDEX_STATS**和**IMPORT_COLUMN_STATS**接口的统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_import_table CASCADE;
NOTICE:  schema "dbe_stats_import_table" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_import_table;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_import_table;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (3,3);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 创建用户统计信息表，导出表统计信息
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_import_table', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# CALL DBE_STATS.EXPORT_TABLE_STATS(
gaussdb(#     ownname => 'dbe_stats_import_table',
gaussdb(#     tabname => 't1',
gaussdb(#     partname => null,
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 't1_t',
gaussdb(#     cascade => true
gaussdb(# );
 export_table_stats
--------------------

(1 row)

gaussdb=# -- 验证: 导出前 reltuples=3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# INSERT INTO t1 VALUES (4,4);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (5,5);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (6,6);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
```


版权所有 © 华为云计算技术有限公司

```
 t1      |         6
(1 row)

gaussdb=# CALL DBE_STATS.IMPORT_TABLE_STATS(
gaussdb(#     ownname => 'dbe_stats_import_table',
gaussdb(#     tabname => 't1',
gaussdb(#     partname => null,
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 't1_t',
gaussdb(#     cascade => true,
gaussdb(#     force => false
gaussdb(# );
 import_table_stats
--------------------

(1 row)

gaussdb=# -- 验证: 导入后 reltuples恢复为3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_import_table;
DROP SCHEMA
```

- DBE_STATS.IMPORT_COLUMN_STATS

  从stattab标识的用户统计信息表中检索指定列（多列/表达式）的统计信息，并将其写回系统表中。该接口无返回值。

  DBE_STATS.IMPORT_COLUMN_STATS原型为：

```
DBE_STATS.IMPORT_COLUMN_STATS (
   ownname       VARCHAR2,
   tabname       VARCHAR2,
   colname       VARCHAR2,
   partname      VARCHAR2 DEFAULT NULL,
   stattab       VARCHAR2,
   statid        VARCHAR2 DEFAULT NULL,
   statown       VARCHAR2 DEFAULT NULL,
   no_invalidate BOOLEAN DEFAULT NULL,
   force         BOOLEAN DEFAULT FALSE
);
```

表 2-392 DBE_STATS.IMPORT_COLUMN_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
tabname	VARCHAR2	否	指定需要导入统计信息的列所属的表名（表达式对应索引名）。
colname	VARCHAR2	否	列名或者多列名（tabname为索引时，colname为表达式名）。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
partname	VARCHAR2	是	分区的名称。如果表/索引已分区但是partname为null，则导入全局和分区列/多列/索引表达式统计信息。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。
no_invalidate	BOOLEAN	是	暂不支持。
force	BOOLEAN	是	是否强制导入，忽略锁的状态，默认false。

说明

- 导入列级的统计信息可查询**PG_STATISTIC**和**PG_STATISTIC_EXT**。
- 导入多列时，需先查询**PG_STATISTIC_EXT**中的staextname字段，获取多列别名作为colname参数传入。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_import_column CASCADE;
NOTICE:  schema "dbe_stats_import_column" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_import_column;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_import_column;
SET
gaussdb=# CREATE TABLE IF NOT EXISTS t1 (a int, b int, c int) distribute by replication;
CREATE TABLE
gaussdb=# CREATE INDEX idx_t1_expr on t1((a*a), (a+a), (a*b));
CREATE INDEX
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# ANALYZE t1((a,b));
ANALYZE
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# -- 建立用户统计信息表，导出列统计信息
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_import_column', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# CALL DBE_STATS.EXPORT_COLUMN_STATS(
gaussdb(#     ownname => 'dbe_stats_import_column',
gaussdb(#     tabname => 't1',
gaussdb(#     colname => 'a',
gaussdb(#     partname => null,
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 't1_c'
gaussdb(# );
 export_column_stats
---------------------

(1 row)

gaussdb=# -- 查看导出结果
gaussdb=# SELECT relname, columnname, stadistinct, stakind1, stanumbers1, stavalues1
gaussdb-# FROM export_table order by 1,2,3,4;
 relname | columnname | stadistinct | stakind1 | stanumbers1 | stavalues1
---------+------------+-------------+----------+-------------+------------
 t1      | a          |          -.5 |        1 | {.666667}   | {250}
 t1      | a          |          -.5 |        1 | {.666667}   | {250}
 t1      | a          |          -.5 |        1 | {.666667}   | {250}
(3 rows)

gaussdb=# -- 再次插入数据，收集统计信息
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 查看import前列统计信息 (stadistinct变化)
gaussdb=# SELECT staattnum, stadistinct, stakind1, stanumbers1, stavalues1
gaussdb-# FROM PG_STATISTIC WHERE starelid='t1'::REGCLASS ORDER BY 1;
 staattnum | stadistinct | stakind1 |        stanumbers1         |  stavalues1
-----------+-------------+----------+----------------------------+---------------
         1 |    -.333333 |        1 | {.555556,.222222,.222222} | {250,100,150}
         2 |    -.333333 |        1 | {.555556,.222222,.222222} | {150,300,550}
         3 |    -.111111 |        1 | {1}                       | {1}
(3 rows)

gaussdb=# CALL DBE_STATS.IMPORT_COLUMN_STATS(
gaussdb(#     ownname => 'dbe_stats_import_column',
gaussdb(#     tabname => 't1',
gaussdb(#     colname => 'a',
gaussdb(#     partname => null,
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 't1_c',
gaussdb(#     force => false
gaussdb(# );
 import_column_stats
---------------------

(1 row)

gaussdb=# -- 验证: 导入后列a的统计信息恢复
gaussdb=# SELECT staattnum, stadistinct, stakind1, stanumbers1, stavalues1
gaussdb-# FROM PG_STATISTIC WHERE starelid='t1'::REGCLASS ORDER BY 1;
 staattnum | stadistinct | stakind1 |        stanumbers1         |  stavalues1
-----------+-------------+----------+----------------------------+---------------
         1 |          -.5 |        1 | {.666667}                 | {250}
```


版权所有 © 华为云计算技术有限公司

```
       2 |    -.333333 |        1 | {.555556,.222222,.222222} | {150,300,550}
       3 |    -.111111 |        1 | {1}                       | {1}
(3 rows)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_import_column;
DROP SCHEMA
```

- DBE_STATS.IMPORT_SCHEMA_STATS

  从stattab标识的用户统计信息表中检索指定Schema的统计信息，并将其写回系统表中。该接口无返回值。

  DBE_STATS.IMPORT_SCHEMA_STATS原型为：

```
DBE_STATS.IMPORT_SCHEMA_STATS (
  ownname       VARCHAR2,
  stattab       VARCHAR2,
  statid        VARCHAR2 DEFAULT NULL,
  statown       VARCHAR2 DEFAULT NULL,
  no_invalidate BOOLEAN  DEFAULT NULL,
  force         BOOLEAN DEFAULT FALSE,
  stat_category VARCHAR2 DEFAULT NULL
);
```

表 2-393 DBE_STATS.IMPORT_SCHEMA_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	指定Schema的名称，如果设置为null，会默认使用当前的Schema。
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用ownname参数的值。
no_invalidate	BOOLEAN	是	暂不支持。
force	BOOLEAN	是	是否强制导入，忽略锁的状态，默认false。
stat_category	VARCHAR2	是	暂不支持。

版权所有 © 华为云计算技术有限公司

说明

- 导入当前用户在指定Schema下所有具有权限的表的统计信息，对于无权限的表直接跳过，不报错。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_import_schema CASCADE;
NOTICE:  schema "dbe_stats_import_schema" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_import_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_import_schema;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (3,3);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 创建用户统计信息表，导出schema统计信息
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_import_schema', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# CALL DBE_STATS.EXPORT_SCHEMA_STATS(
gaussdb(#     ownname => 'dbe_stats_import_schema',
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 's1'
gaussdb(# );
 export_schema_stats
---------------------

(1 row)

gaussdb=# -- 验证: 导出前 reltuples=3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 再次插入数据，收集统计信息
gaussdb=# INSERT INTO t1 VALUES (4,4);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (5,5);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (6,6);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 验证: 统计信息已更新 reltuples=6
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         6
(1 row)

gaussdb=# -- 导入schema统计信息
gaussdb=# CALL DBE_STATS.IMPORT_SCHEMA_STATS(
```


版权所有 © 华为云计算技术有限公司

```
gaussdb(#     ownname => 'dbe_stats_import_schema',
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 's1',
gaussdb(#     force => false
gaussdb(# );
 import_schema_stats
---------------------

(1 row)

gaussdb=# -- 验证: 导入后 reltuples恢复为3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_import_schema;
DROP SCHEMA
```

- DBE_STATS.IMPORT_DATABASE_STATS

  从stattab标识的用户统计信息表中检索当前database用户表的统计信息，并将其写回系统表中。该接口无返回值。

  DBE_STATS.IMPORT_DATABASE_STATS原型为：

```
DBE_STATS.IMPORT_DATABASE_STATS (
   stattab         VARCHAR2,
   statid          VARCHAR2 DEFAULT NULL,
   statown         VARCHAR2 DEFAULT NULL,
   no_invalidate   BOOLEAN DEFAULT NULL,
   force           BOOLEAN DEFAULT FALSE,
   stat_category   VARCHAR2 DEFAULT NULL
);
```

表 2-394 DBE_STATS.IMPORT_DATABASE_STATS 接口参数说明

参数	类型	是否允许为空	描述
stattab	VARCHAR2	否	指定存储统计信息用户表的表名。
statid	VARCHAR2	是	指向stattab中统计信息的标识符（可选）。
statown	VARCHAR2	是	指定存储统计信息用户表所在Schema的名称，如果设置为null，会使用当前Schema。
no_invalidate	BOOLEAN	是	暂不支持。
force	BOOLEAN	是	是否强制导入，忽略锁的状态，默认false。
stat_category	VARCHAR2	是	暂不支持。

版权所有 © 华为云计算技术有限公司

说明

- 导入当前用户在指定database下所有具有analyze权限的用户表的统计信息，对于无权限的表直接跳过，不报错。
- 当statown参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_import_database CASCADE;
NOTICE:  schema "dbe_stats_import_database" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_import_database;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_import_database;
SET
gaussdb=# CREATE TABLE t1(a int, b int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1,1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (2,2);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (3,3);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 创建用户统计信息表，导出database统计信息
gaussdb=# CALL DBE_STATS.CREATE_STAT_TABLE('dbe_stats_import_database', 'export_table');
 create_stat_table
-------------------

(1 row)

gaussdb=# CALL DBE_STATS.EXPORT_DATABASE_STATS(
gaussdb(#     statown => 'dbe_stats_import_database',
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 's1'
gaussdb(# );
 export_database_stats
-----------------------

(1 row)

gaussdb=# -- 验证: 导出前 reltuples=3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 再次插入数据，收集统计信息
gaussdb=# INSERT INTO t1 VALUES (4,4);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (5,5);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (6,6);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 验证: 统计信息已更新 reltuples=6
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         6
(1 row)

gaussdb=# -- 导入database统计信息
gaussdb=# CALL DBE_STATS.IMPORT_DATABASE_STATS(
```


版权所有 © 华为云计算技术有限公司

```
gaussdb(#     statown => 'dbe_stats_import_database',
gaussdb(#     stattab => 'export_table',
gaussdb(#     statid => 's1',
gaussdb(#     force => false
gaussdb(# );
 import_database_stats
-----------------------

(1 row)

gaussdb=# -- 验证: 导入后 reltuples恢复为3
gaussdb=# SELECT relname, reltuples FROM PG_CLASS WHERE oid='t1'::regclass;
 relname | reltuples
---------+-----------
 t1      |         3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP TABLE export_table;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_import_database;
DROP SCHEMA
```

- DBE_STATS.SET_COLUMN_STATS

  设置列相关的统计信息，包括设置单列、多列和表达式统计信息。该接口无返回值。

  DBE_STATS.SET_COLUMN_STATS原型为：

```
DBE_STATS.SET_COLUMN_STATS (
   ownname         VARCHAR2,
   tabname         VARCHAR2,
   colname         VARCHAR2,
   partname        VARCHAR2    DEFAULT NULL,
   stattab         VARCHAR2    DEFAULT NULL,
   statid          VARCHAR2    DEFAULT NULL,
   distcnt         NUMBER      DEFAULT NULL,
   density         NUMBER      DEFAULT NULL,
   nullcnt         NUMBER      DEFAULT NULL,
   srec            VARCHAR2    DEFAULT NULL,
   avgclen         NUMBER      DEFAULT NULL,
   flags           NUMBER      DEFAULT NULL,
   statown         VARCHAR2    DEFAULT NULL,
   no_invalidate   BOOLEAN     DEFAULT NULL,
   force           BOOLEAN     DEFAULT FALSE,
   node_name       VARCHAR2    DEFAULT NULL,
   option          VARCHAR2    DEFAULT NULL
);
```

表 2-395 DBE_STATS.SET_COLUMN_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	该表所在的Schema。如果设置为null，则默认使用当前Schema。
tabname	VARCHAR2	否	表名。
colname	VARCHAR2	否	列名。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
partname	VARCHAR2	是	要设置统计信息的表分区的名称。如果表已分区且partname为null，则设置全局级别的列级统计信息。
stattab	VARCHAR2	是	暂不支持此参数。
statid	VARCHAR2	是	暂不支持此参数。
distcnt	NUMBER	是	设置该列中NDV数量。
density	NUMBER	是	暂不支持。
nullcnt	NUMBER	是	设置该列中null值数量。
srec	VARCHAR2	是	暂不支持此参数。
avgclen	NUMBER	是	设置该列中字段的平均长度（以字节为单位）。
flags	NUMBER	是	暂不支持此参数。
statown	VARCHAR2	是	暂不支持此参数。
no_invalidate	BOOLEAN	是	暂不支持此参数。
force	BOOLEAN	是	决定统计信息被锁定时的行为。如果为true，则即使列级统计信息被锁定，该过程也会设置值。默认为false。
node_name	VARCHAR2	是	node_name为null会更新所有CN和DN节点的统计信息； node_name为DN节点名仅会更新指定DN的统计信息； node_name为CN节点名会更新所有CN的统计信息
option	VARCHAR2	是	DN节点NDV的计算方式，仅在node_name为null时生效，当前只允许"p", "s"。 "p"：表示使用泊松分布计算DN节点的NDV值。 "s"：表示DN节点和CN节点使用同一NDV值。

版权所有 © 华为云计算技术有限公司

说明

- 该接口支持设置表达式统计信息，由于表达式的创建依赖于索引，因此当调用该接口设置表达式统计信息时，tabname应该传入该索引的名称。
- 当前仅支持设置列的NDV（distcnt）、nullfrac（nullcnt）和width（avgclen）三类统计信息。
- 当单列、表达式统计信息被删除后，仍然支持调用该接口设置对应的单列、表达式统计信息，此时**PG_STATISTIC**系统表中将插入新的数据。但对于多列统计信息，当其被删除后，不支持使用该接口进行重新设置。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。
- 当前列的行数(reltuples)为0时，nullfrac将会被设置为0，distcnt将会被设置为入参的值。

示例：

```
-- 前置条件，建表，收集统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_set_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_set_schema;
SET
gaussdb=# DROP TABLE IF EXISTS t1;
NOTICE:  table "t1" does not exist, skipping
DROP TABLE
gaussdb=# CREATE TABLE IF NOT EXISTS t1 (a int, b int, c int) distribute by replication;
CREATE TABLE
gaussdb=# CREATE INDEX idx_t1_expr ON t1((a*a), (a+a), (a*b));
CREATE INDEX
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# ANALYZE t1((a, b));
ANALYZE
-- 设置单列统计信息。
-- 查询原有统计信息。
gaussdb=# SELECT staattnum,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC WHERE
starelid='t1'::REGCLASS ORDER BY 1;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           0 |        4 |       -.75
         2 |           0 |        4 |       -.75
         3 |           0 |        4 |       -.25
(3 rows)
-- 设置列级统计信息后重新查询。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
't1', colname => 'a', distcnt => 6, nullcnt => 5, avgclen => 42);
 set_column_stats
------------------

(1 row)
gaussdb=# SELECT staattnum,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC WHERE
starelid='t1'::REGCLASS ORDER BY 1;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           1 |       42 |           6
         2 |           0 |        4 |       -.75
         3 |           0 |        4 |       -.25
(3 rows)
-- 设置多列统计信息。
-- 查询原有统计信息。
gaussdb=# SELECT stakey,staextname,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC_EXT
```


版权所有 © 华为云计算技术有限公司

```
WHERE starelid='t1'::REGCLASS;
 stakey |     staextname      | stanullfrac | stawidth | stadistinct
--------+---------------------+-------------+----------+-------------
 1 2    | extname_-1308656218 |           0 |        8 |        -.75
(1 row)
-- 设置统计信息后重新查询。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
't1', colname => 'extname_-1308656218', distcnt => 5, nullcnt => 5, avgclen => 2);
 set_column_stats
------------------

(1 row)
gaussdb=# SELECT stakey,staextname,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC_EXT
WHERE starelid='t1'::REGCLASS;
 stakey |     staextname      | stanullfrac | stawidth | stadistinct
--------+---------------------+-------------+----------+-------------
 1 2    | extname_-1308656218 |           1 |        2 |           5
(1 row)
-- 设置表达式统计信息。
-- 查询原有统计信息。
gaussdb=# SELECT staattnum,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC WHERE
starelid='idx_t1_expr'::REGCLASS ORDER BY 1;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           0 |        4 |        -.75
         2 |           0 |        4 |        -.75
         3 |           0 |        4 |        -.75
(3 rows)
-- 查询表达式列名。
gaussdb=# SELECT attnum, attname FROM PG_ATTRIBUTE WHERE attrelid='idx_t1_expr'::REGCLASS
ORDER BY 1;
 attnum | attname
--------+---------
      1 | expr
      2 | expr1
      3 | expr2
(3 rows)
-- 设置统计信息后重新查询。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
'idx_t1_expr', colname => 'expr', distcnt => 13, nullcnt => 5, avgclen => 22);
 set_column_stats
------------------

(1 row)
gaussdb=# SELECT staattnum,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC WHERE
starelid='idx_t1_expr'::REGCLASS ORDER BY 1;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           1 |       22 |          13
         2 |           0 |        4 |        -.75
         3 |           0 |        4 |        -.75
(3 rows)
-- 设置指定DN的统计信息。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
't1', colname => 'a', distcnt => 3, nullcnt => 3,avgclen => 3, node_name=>'dn_6007_6008_6009');
 set_column_stats
------------------


(1 row)
gaussdb=# EXECUTE DIRECT ON (dn_6007_6008_6009) $$SELECT
staattnum,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC WHERE starelid='t1'::REGCLASS AND
staattnum = 1;$$;
staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |         .75 |        3 |           3
(1 row)
-- 设置所有CN的统计信息。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
't1', colname => 'a', distcnt => 6, nullcnt => 6,avgclen => 6, node_name=>'cn_5002');
 set_column_stats
------------------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
gaussdb=# EXECUTE DIRECT ON (cn_5002) $$SELECT staattnum,stanullfrac,stawidth,stadistinct FROM
PG_STATISTIC WHERE starelid='t1'::REGCLASS AND staattnum = 1;$$;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           1 |        6 |           6
(1 row)
-- 设置整个集群的统计信息，其中DN的distcnt值和CN保持一致。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
't1', colname => 'a', distcnt => 100, nullcnt => 100,avgclen => 100, option=>'s');
 set_column_stats
------------------

(1 row)
gaussdb=# EXECUTE DIRECT ON (cn_5002) $$SELECT staattnum,stanullfrac,stawidth,stadistinct FROM
PG_STATISTIC WHERE starelid='t1'::REGCLASS AND staattnum = 1;$$;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           1 |      100 |         100
(1 row)
gaussdb=# EXECUTE DIRECT ON (dn_6007_6008_6009) $$SELECT
staattnum,stanullfrac,stawidth,stadistinct FROM PG_STATISTIC WHERE starelid='t1'::REGCLASS AND
staattnum = 1;$$;
 staattnum | stanullfrac | stawidth | stadistinct
-----------+-------------+----------+-------------
         1 |           1 |      100 |         100
(1 row)
-- 设置指定DN的行数和页数。
gaussdb=# CALL DBE_STATS.SET_TABLE_STATS(tabname => 't1', ownname => 'dbe_stats_set_schema',
numrows => 3, numblks => 3, relallvisible => 1, node_name=>'dn_6007_6008_6009');
 set_table_stats
-----------------
(1 row)
-- 设置整个集群的统计信息，其中DN的distcnt值采用泊松分布计算。
gaussdb=# CALL DBE_STATS.SET_COLUMN_STATS(ownname => 'dbe_stats_set_schema', tabname =>
't1', colname => 'a', distcnt => 100, nullcnt => 100,avgclen => 100, option=>'p');
 set_column_stats
------------------

(1 row)
-- 删除表、命名空间。
gaussdb=# DROP INDEX idx_t1_expr;
DROP INDEX
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_set_schema cascade;
DROP SCHEMA
```

- DBE_STATS.SET_INDEX_STATS

  设置索引相关的统计信息。该接口无返回值。

  DBE_STATS.SET_INDEX_STATS原型为：

```
DBE_STATS.SET_INDEX_STATS (
   ownname       VARCHAR2,
   indname       VARCHAR2,
   partname      VARCHAR2  DEFAULT NULL,
   stattab       VARCHAR2  DEFAULT NULL,
   statid        VARCHAR2  DEFAULT NULL,
   numrows       NUMBER    DEFAULT NULL,
   numlblks      NUMBER    DEFAULT NULL,
   relallvisible NUMBER    DEFAULT NULL,
   numdist       NUMBER    DEFAULT NULL,
   avglblk       NUMBER    DEFAULT NULL,
   avgdblk       NUMBER    DEFAULT NULL,
   clstfct       NUMBER    DEFAULT NULL,
   indlevel      NUMBER    DEFAULT NULL,
   flags         NUMBER    DEFAULT NULL,
   statown       VARCHAR2  DEFAULT NULL,
```


版权所有 © 华为云计算技术有限公司

```
  no_invalidate BOOLEAN    DEFAULT NULL,
  guessq        NUMBER     DEFAULT NULL,
  cachedblk     NUMBER     DEFAULT NULL,
  cachehit      NUMBER     DEFAULT NULL,
  force         BOOLEAN    DEFAULT FALSE,
  node_name     VARCHAR2   DEFAULT NULL
);
```

表 2-396 DBE_STATS.SET_INDEX_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	该索引所在的Schema。如果设置为null，则默认使用当前Schema。
indname	VARCHAR2	否	索引名。
partname	VARCHAR2	是	要设置统计信息的索引分区的名称。如果索引已分区且partname为null，则设置全局索引级别的统计信息。
stattab	VARCHAR2	是	暂不支持该参数。
statid	VARCHAR2	是	暂不支持该参数。
numrows	NUMBER	是	设置索引的行数。
numlblks	NUMBER	是	设置索引的页数。
relallvisible	NUMBER	是	设置被标识为全可见的索引中的页的数量。
numdist	NUMBER	是	暂不支持该参数。
avglblk	NUMBER	是	暂不支持此参数。
avgdblk	NUMBER	是	暂不支持此参数。
clstfct	NUMBER	是	暂不支持此参数。
indlevel	NUMBER	是	暂不支持此参数。
flags	NUMBER	是	暂不支持此参数。
statown	VARCHAR2	是	暂不支持此参数。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
no_invalidate	BOOLEAN	是	暂不支持此参数。
guessq	NUMBER	是	暂不支持此参数。
cachedblk	NUMBER	是	暂不支持此参数。
cachehit	NUMBER	是	暂不支持此参数。
force	BOOLEAN	是	决定统计信息被锁定时的行为。如果为TRUE，则即使索引统计信息被锁定，该过程也会设置值。默认为false。
node_name	VARCHAR2	是	node_name为null会更新所有CN和DN节点的统计信息； node_name为DN节点名仅会更新指定DN的统计信息； node_name为CN节点名会更新所有CN的统计信息。

说明

- 当前仅支持设置索引的reltuples（numrows）、relpages（numlblks）和relallvisible（relallvisible）三类统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
-- 前置条件，建表，收集统计信息。
gaussdb=# CREATE SCHEMA dbe_stats_set_schema;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_set_schema;
SET
gaussdb=# DROP TABLE IF EXISTS t1;
NOTICE:  table "t1" does not exist, skipping
DROP TABLE
gaussdb=# CREATE TABLE IF NOT EXISTS t1 (a int, b int, c int) distribute by replication;
CREATE TABLE
gaussdb=# CREATE INDEX idx_t1_a on t1(a);
CREATE INDEX
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
-- 设置索引统计信息。
-- 查看当前索引统计信息。
gaussdb=# SELECT reltuples FROM PG_CLASS WHERE relname='idx_t1_a';
 reltuples
```


版权所有 © 华为云计算技术有限公司

```
-----------
         3
(1 row)
-- 设置统计信息后重新查询。
gaussdb=# CALL DBE_STATS.SET_INDEX_STATS(indname => 'idx_t1_a', ownname =>
'dbe_stats_set_schema', numrows => 1, numlblks => 3, relallvisible => 6);
 set_index_stats
-----------------
(1 row)
gaussdb=# SELECT reltuples,relpages,relallvisible FROM PG_CLASS WHERE relname='idx_t1_a';
 reltuples | relpages | relallvisible
-----------+----------+---------------
         1 |        3 |             6
(1 row)
-- 设置指定DN的统计信息。
gaussdb=# CALL DBE_STATS.SET_INDEX_STATS(indname => 'idx_t1_a', ownname =>
'dbe_stats_set_schema', numrows => 2, numlblks => 2, relallvisible =>
2,node_name=>'dn_6007_6008_6009');
 set_index_stats
-----------------

(1 row)
-- 查询指定DN的统计信息。
gaussdb=# EXECUTE DIRECT ON (dn_6007_6008_6009) $$SELECT reltuples,relpages,relallvisible FROM
PG_CLASS WHERE relname='idx_t1_a'$$;
 reltuples | relpages | relallvisible
-----------+----------+---------------
         2 |        2 |             2
(1 row)
-- 设置指定CN的统计信息。
gaussdb=# CALL DBE_STATS.SET_INDEX_STATS(indname => 'idx_t1_a', ownname =>
'dbe_stats_set_schema', numrows => 10, numlblks => 100, relallvisible => 10,node_name=>'cn_5001');
 set_index_stats
-----------------

(1 row)
-- 查询指定CN的统计信息。
gaussdb=# EXECUTE DIRECT ON (cn_5001) $$SELECT reltuples,relpages,relallvisible FROM PG_CLASS
WHERE relname='idx_t1_a'$$;
 reltuples | relpages | relallvisible
-----------+----------+---------------
        10 |      100 |            10
(1 row)
-- 删除表、命名空间。
gaussdb=# DROP INDEX idx_t1_a;
DROP INDEX
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_set_schema CASCADE;
DROP SCHEMA
```

- DBE_STATS.SET_TABLE_STATS

  设置表级统计信息。该接口无返回值。

  DBE_STATS.SET_TABLE_STATS原型为：

```
DBE_STATS.SET_TABLE_STATS (
    ownname          VARCHAR2,
    tabname          VARCHAR2,
    partname         VARCHAR2 DEFAULT NULL,
    stattab          VARCHAR2  DEFAULT NULL,
    statid           VARCHAR2  DEFAULT NULL,
    numrows          NUMBER    DEFAULT NULL,
    numblks          NUMBER    DEFAULT NULL,
    relallvisible    NUMBER    DEFAULT NULL,
    avgrlen          NUMBER    DEFAULT NULL,
    flags            NUMBER    DEFAULT NULL,
    statown          VARCHAR2  DEFAULT NULL,
    no_invalidate    BOOLEAN   DEFAULT NULL,
    cachedblk        NUMBER    DEFAULT NULL,
```


版权所有 © 华为云计算技术有限公司

```
  cachehit       NUMBER    DEFAULT NULL,
  force          BOOLEAN   DEFAULT FALSE,
  im_imcu_count  NUMBER    DEFAULT NULL,
  im_block_count NUMBER    DEFAULT NULL,
  scanrate       NUMBER    DEFAULT NULL,
  node_name      VARCHAR2  DEFAULT NULL
);
```

表 2-397 DBE_STATS.SET_TABLE_STATS 接口参数说明

参数	类型	是否允许为空	描述
ownname	VARCHAR2	否	该表所在的Schema。如果设置为null，则默认使用当前Schema。
tabname	VARCHAR2	否	表名。
partname	VARCHAR2	是	要设置统计信息的表分区的名称。如果表已分区且partname为null，则设置全局表级别的统计信息。
stattab	VARCHAR2	是	暂不支持该参数。
statid	VARCHAR2	是	暂不支持该参数。
numrows	NUMBER	是	设置表的行数。
numblks	NUMBER	是	设置表的页数。
relallvisible	NUMBER	是	设置被标识为全可见的表中的页的数量。
avgrlen	NUMBER	是	暂不支持该参数。
flags	NUMBER	是	暂不支持此参数。
statown	VARCHAR2	是	暂不支持此参数。
no_invalidate	BOOLEAN	是	暂不支持此参数。
cachedblk	NUMBER	是	暂不支持此参数。
cachehit	NUMBER	是	暂不支持此参数。
force	BOOLEAN	是	决定统计信息被锁定时的行为。如果为TRUE，则即使表统计信息被锁定，该过程也会设置值。默认为false。

版权所有 © 华为云计算技术有限公司

参数	类型	是否允许为空	描述
im_imcu_count	NUMBER	是	暂不支持此参数。
im_block_count	NUMBER	是	暂不支持此参数。
scanrate	NUMBER	是	暂不支持此参数。
node_name	VARCHAR2	是	node_name为null会更新所有CN和DN节点的统计信息； node_name为DN节点名仅会更新指定DN的统计信息； node_name为CN节点名会更新所有CN的统计信息。

说明

- 当前仅支持设置表的reltuples（numrows）、relpages（numblks）和relallvisible（relallvisible）三类统计信息。
- 当ownname参数值设置为null时使用当前Schema，需将behavior_compat_options参数值设置为bind_procedure_searchpath。

示例：

```
gaussdb=# DROP SCHEMA IF EXISTS dbe_stats_set_schema_table CASCADE;
NOTICE:  schema "dbe_stats_set_schema_table" does not exist, skipping
DROP SCHEMA
gaussdb=# -- 前置条件，建表，收集统计信息
gaussdb=# CREATE SCHEMA dbe_stats_set_schema_table;
CREATE SCHEMA
gaussdb=# SET CURRENT_SCHEMA = dbe_stats_set_schema_table;
SET
gaussdb=# DROP TABLE IF EXISTS t1;
NOTICE:  table "t1" does not exist, skipping
DROP TABLE
gaussdb=# CREATE TABLE IF NOT EXISTS t1 (a int, b int, c int) distribute by replication;
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (100, 300, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (150, 550, 1);
INSERT 0 1
gaussdb=# INSERT INTO t1 VALUES (250, 150, 1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# -- 查看当前表统计信息
gaussdb=# SELECT reltuples, relpages, relallvisible FROM PG_CLASS WHERE oid='t1'::regclass;
 reltuples | relpages | relallvisible
-----------+----------+---------------
         3 |        1 |             0
(1 row)

gaussdb=# -- 设置表级统计信息 (numrows=3, numblks=3, relallvisible=3)
gaussdb=# CALL DBE_STATS.SET_TABLE_STATS(
gaussdb(#     tabname => 't1',
gaussdb(#     ownname => 'dbe_stats_set_schema_table',
gaussdb(#     numrows => 3,
```


版权所有 © 华为云计算技术有限公司

```
gaussdb(#     numblks => 3,
gaussdb(#     relallvisible => 3
gaussdb(# );
 set_table_stats
-----------------

(1 row)

gaussdb=# -- 验证: 表统计信息已设置 (reltuples=3, relpages=3, relallvisible=3)
gaussdb=# SELECT reltuples, relpages, relallvisible FROM PG_CLASS WHERE oid='t1'::regclass;
 reltuples | relpages | relallvisible
-----------+----------+---------------
         3 |        3 |             3
(1 row)

gaussdb=# -- 清理
gaussdb=# DROP TABLE t1;
DROP TABLE
gaussdb=# DROP SCHEMA dbe_stats_set_schema_table CASCADE;
DROP SCHEMA
```

- DBE_STATS.GET_TABLES_BY_SQLID

  获取Unique SQL ID对应语句所有涉及的表，如果包含多张表，则返回多行记录。

  DBE_STATS.GET_TABLES_BY_SQLID原型为：

```
DBE_STATS.GET_TABLES_BY_SQLID (
   unique_sqlid    BIGINT
)
RETURNS TABLE (
   nspname         TEXT,
   tabname         TEXT
);
```

表 2-398 DBE_STATS.GET_TABLES_BY_SQLID 接口参数与返回值说明

字段名	数据类型	字段类型	是否允许为空	描述
unique_sqlid	BIGINT	IN	否	唯一SQL ID，可通过pg_catalog.pg_stat_activity（实时会话信息视图）、dbe_perf.statement（归一化SQL语句记录视图）或dbe_perf.statement_history（全量或慢SQL语句记录视图）中查询获得。
nspname	TEXT	OUT	否	语句中涉及表的模式名称。
tabname	TEXT	OUT	否	语句中涉及表的名称。

示例：

```
-- 前置条件
gaussdb=# DROP TABLE IF EXISTS t1;
DROP TABLE
gaussdb=# CREATE TABLE t1 (c1 int) DISTRIBUTE BY HASH (c1);
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1);
INSERT 0 1
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# CREATE OR REPLACE PROCEDURE select_t1()
gaussdb-# AS
gaussdb$# BEGIN
gaussdb$#     PERFORM * FROM t1;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# DROP TABLE IF EXISTS t2;
DROP TABLE
gaussdb=# CREATE TABLE t2 (c1 int) DISTRIBUTE BY HASH (c1) PARTITION BY LIST (c1) (
gaussdb(#     PARTITION t2_p1 VALUES (1),
gaussdb(#     PARTITION t2_p2 VALUES (2)
gaussdb(# );
CREATE TABLE
gaussdb=# INSERT INTO t2 VALUES (1), (2);
INSERT 0 2

-- 使用从pgxc_stat_activity中查询的unique sqlid。
-- 会话1：模拟无法执行的语句。
gaussdb=# SET cursor_sharing = off; -- 避免自动参数化不执行解析。
SET
gaussdb=# SELECT pg_sleep(60) from t1;
 pg_sleep
----------

(1 row)

-- 会话2：在会话一执行期间查询其unique sqlid，并查询对应的tabname，语句需根据实际sql_id调整。
gaussdb=#  SELECT query, sessionid, unique_sql_id FROM pgxc_stat_activity WHERE query LIKE
'%pg_sleep%t1%' AND query NOT LIKE '%pgxc_stat_activity%';
            query              | sessionid | unique_sql_id
-------------------------------+-----------+---------------
 SELECT pg_sleep(60) FROM t1; |         1 |    3101300097
(1 row)

gaussdb=# SELECT * FROM dbe_stats.get_tables_by_sqlid(3101300097);
 nspname | tabname
---------+---------
 public  | t1
(1 row)

-- 2.使用从dbe_perf.statement中查询的unique sqlid。
gaussdb=# SHOW instr_unique_sql_count; -- 查看statement视图中的最大记录数。
 instr_unique_sql_count
------------------------
 200000
(1 row)

gaussdb=# SELECT count(*) FROM dbe_perf.summary_statement; -- 确认statement视图中的记录数是否
超限。
 count
-------
    46
(1 row)

gaussdb=# SELECT reset_unique_sql('GLOBAL','ALL',0); -- 如果达到限制，后续的语句不会再记录，可以
清空记录（注：会导致SQL记录丢失）。
 reset_unique_sql
------------------
 t
(1 row)

gaussdb=# SELECT * FROM t1;
 c1
----
  1
(1 row)

gaussdb=# SELECT unique_sql_id FROM dbe_perf.summary_statement WHERE query LIKE '%SELECT *
```


版权所有 © 华为云计算技术有限公司

```
FROM t1%';
 unique_sql_id
---------------
    4115569441
(1 row)

gaussdb=# SELECT * FROM dbe_stats.get_tables_by_sqlid(4115569441);
 nspname | tabname
---------+---------
 public  | t1
(1 row)

-- 3.使用从dbe_stats.statement_history中查询的unique sqlid。
gaussdb=# SET log_min_duration_statement = '1s'; -- 执行时间>=1s的慢SQL会被记录。
SET
gaussdb=# DELETE FROM pg_catalog.statement_history; -- 避免受残留的慢SQL记录影响。
DELETE 2
gaussdb=# SELECT pg_sleep(1) FROM t1;
 pg_sleep
----------

(1 row)

gaussdb=# SELECT unique_query_id FROM dbe_perf.statement_history WHERE query LIKE '%pg_sleep
%t1%' LIMIT 1;
 unique_query_id
-----------------
      3101300097
(1 row)

gaussdb=# SELECT * FROM dbe_stats.get_tables_by_sqlid(3101300097);
 nspname | tabname
---------+---------
 public  | t1
(1 row)

-- 4.使用存储过程的unique sqlid，从dbe_stats.statement中查询。
gaussdb=# SHOW instr_unique_sql_track_type; -- 确认此时statement视图记录包括存储过程子语句。
 instr_unique_sql_track_type
-----------------------------
 all
(1 row)

gaussdb=# SELECT * FROM select_t1();
 select_t1
-----------

(1 row)

gaussdb=# SELECT s1.query, s1.unique_sql_id FROM dbe_perf.summary_statement s1 WHERE s1.query
= 'SELECT * FROM select_t1();'
gaussdb-# UNION ALL
gaussdb-# SELECT s2.query, s2.unique_sql_id FROM dbe_perf.summary_statement s1,
dbe_perf.summary_statement s2 WHERE s1.query = 'SELECT * FROM select_t1();' AND
s1.unique_sql_id = s2.parent_unique_sql_id;
           query            | unique_sql_id
----------------------------+---------------
 SELECT * FROM select_t1(); |    1861705880
 SELECT * FROM t1           |    4115569441
(2 rows)

gaussdb=# SELECT * FROM dbe_stats.get_tables_by_sqlid(1861705880); -- 顶层语句不包含基表。
 nspname | tabname
---------+---------
(0 rows)

gaussdb=# SELECT * FROM dbe_stats.get_tables_by_sqlid(4115569441); -- 子语句包含基表。
 nspname | tabname
---------+---------
```


版权所有 © 华为云计算技术有限公司

```
 public  | t1
(1 row)

-- 5.在dbe_perf.statement中查询，涉及多张表的语句。
gaussdb=# SELECT * FROM t1, t2 LIMIT 1;
 c1 | c1
----+----
  1 |  1
(1 row)

gaussdb=# SELECT query, unique_sql_id FROM dbe_perf.summary_statement WHERE query LIKE
'%SELECT * FROM t1, t2 LIMIT%';
              query              | unique_sql_id
---------------------------------+---------------
 SELECT * FROM t1, t2 LIMIT ?; |    2763456878
(1 row)

gaussdb=# SELECT * FROM dbe_stats.get_tables_by_sqlid(2763456878);
 nspname | tabname
---------+---------
 public  | t1
 public  | t2
(2 rows)

-- 后置操作
gaussdb=# DROP PROCEDURE IF EXISTS select_t1;
DROP PROCEDURE
gaussdb=# DROP TABLE IF EXISTS t1;
DROP TABLE
gaussdb=# DROP TABLE IF EXISTS t2;
DROP TABLE
```

- DBE_STATS.ANALYZE_SQLID

  获取Unique SQL ID对应语句所有涉及的表，以及收集对应表的统计信息。对于收集成功的表，如果包含多张表，则返回多行记录。

  DBE_STATS.ANALYZE_SQLID原型为：

```
DBE_STATS.ANALYZE_SQLID (
    unique_sqlid    BIGINT,
    all_complete    BOOLEAN DEFAULT FALSE,
    node_name       NAME DEFAULT NULL
)
RETURNS TABLE (
    nspname         TEXT,
    tabname         TEXT
);
```

表 2-399 DBE_STATS.ANALYZE_SQLID 接口参数与返回值说明

字段名	数据类型	字段类型	是否允许为空	描述
unique_sqlid	BIGINT	IN	否	唯一SQL ID，可通过pg_catalog.pg_stat_activity（实时会话信息视图）、dbe_perf.statement（归一化SQL语句记录视图）或dbe_perf.statement_history（全量或慢SQL语句记录视图）中查询获得。

版权所有 © 华为云计算技术有限公司

字段名	数据类型	字段类型	是否允许为空	描述
all_complete	BOOLEAN	IN	是	指定是否使用ALL COMPLETE模式进行统计信息收集，与ANALYZE tabname WITH ALL COMPLETE的行为一致。未指定则使用默认模式，对应ANALYZE tabname。
node_name	NAME	IN	是	CN节点名，对应pgxc_node.node_name。如果指定了CN名，则只查询对应CN中的SQL语句记录，未指定则查询所有CN中的记录。
nspname	TEXT	OUT	否	语句中涉及表的模式名称。
tabname	TEXT	OUT	否	语句中涉及表的名称。仅包含成功收集统计信息的表。

示例：

```
-- ANALYZE_SQLID的使用方法与上述GET_TABLES_SQLID基本相同，除了本接口会实际收集统计信息，以及提供了all_complete选项、node_name节点入参的筛选。

-- 前置条件。
gaussdb=# DROP TABLE IF EXISTS t1;
DROP TABLE
gaussdb=# CREATE TABLE t1 (c1 int) DISTRIBUTE BY HASH (c1);
CREATE TABLE
gaussdb=# INSERT INTO t1 VALUES (1);
INSERT 0 1
gaussdb=# ANALYZE t1;
ANALYZE
gaussdb=# CREATE OR REPLACE PROCEDURE select_t1()
gaussdb-# AS
gaussdb$# BEGIN
gaussdb$#     PERFORM * FROM t1;
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# DROP TABLE IF EXISTS t2;
DROP TABLE
gaussdb=# CREATE TABLE t2 (c1 int) DISTRIBUTE BY HASH (c1) PARTITION BY LIST (c1) (
gaussdb(#     PARTITION t2_p1 VALUES (1),
gaussdb(#     PARTITION t2_p2 VALUES (2)
gaussdb(# );
CREATE TABLE
gaussdb=# INSERT INTO t2 VALUES (1), (2);
INSERT 0 2
gaussdb=# ANALYZE t2;
ANALYZE

-- 1.使用从pgxc_stat_activity中查询的unique sqlid。
-- 会话1：模拟无法执行的语句。
gaussdb=# SET cursor_sharing = off; -- 避免自动参数化不执行解析。
SET
gaussdb=# SELECT pg_sleep(60) FROM t1;
 pg_sleep
----------
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

-- 会话2：在会话一执行期间查询其unique sqlid，并查询对应的tabname，语句需根据实际sql_id调整。
gaussdb=#  SELECT query, sessionid, unique_sql_id FROM pgxc_stat_activity WHERE query LIKE
'%pg_sleep%t1%' AND query NOT LIKE '%pgxc_stat_activity%';
            query              | sessionid | unique_sql_id
-------------------------------+-----------+---------------
 SELECT pg_sleep(60) FROM t1; |         1 |    3101300097
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 此前的analyze次数。
 analyze_count
---------------
             1
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(3101300097);
 nspname | tabname
---------+---------
 public  | t1
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 确认以上操作完成了一次analyze。
 analyze_count
---------------
             2
(1 row)

-- 2.使用从dbe_perf.statement中查询的unique sqlid。
gaussdb=# SHOW instr_unique_sql_count; -- 查看statement视图中的最大记录数。
 instr_unique_sql_count
------------------------
 200000
(1 row)

gaussdb=# SELECT count(*) FROM dbe_perf.summary_statement; -- 确认statement视图中的记录数是否
超限。
 count
-------
    46
(1 row)

gaussdb=# SELECT reset_unique_sql('GLOBAL','ALL',0); -- 如果达到限制，后续的语句不会再记录，可以
清空记录（注：会导致SQL记录丢失）。
 reset_unique_sql
------------------
 t
(1 row)

gaussdb=# SELECT * FROM t1;
 c1
----
  1
(1 row)

gaussdb=# SELECT unique_sql_id FROM dbe_perf.summary_statement WHERE query LIKE '%SELECT *
FROM t1%';
 unique_sql_id
---------------
    4115569441
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 此前的analyze次数。
 analyze_count
---------------
```


版权所有 © 华为云计算技术有限公司

```
           2
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(4115569441);
 nspname | tabname
---------+---------
 public  | t1
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 确认以上操作完成了一次analyze。
 analyze_count
---------------
           3
(1 row)

-- 3.使用从dbe_stats.statement_history中查询的unique sqlid。
gaussdb=# SET log_min_duration_statement = '1s'; -- 执行时间>=1s的慢SQL会被记录。
SET
gaussdb=# DELETE FROM pg_catalog.statement_history; -- 避免受残留的慢SQL记录影响。
DELETE 2
gaussdb=# SELECT pg_sleep(1) FROM t1;
 pg_sleep
----------


(1 row)

gaussdb=# SELECT unique_query_id FROM dbe_perf.statement_history WHERE query LIKE '%pg_sleep
%t1%' LIMIT 1;
 unique_query_id
-----------------
      3101300097
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 此前的analyze次数。
 analyze_count
---------------
           3
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(3101300097);
 nspname | tabname
---------+---------
 public  | t1
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 确认以上操作完成了一次analyze。
 analyze_count
---------------
           4
(1 row)

-- 4.使用存储过程的unique sqlid，从dbe_stats.statement中查询。
gaussdb=# SHOW instr_unique_sql_track_type; -- 确认此时statement视图记录包括存储过程子语句。
 instr_unique_sql_track_type
-----------------------------
 all
(1 row)

gaussdb=# SELECT * FROM select_t1();
 select_t1
-----------


(1 row)

gaussdb=# SELECT s1.query, s1.unique_sql_id FROM dbe_perf.summary_statement s1 WHERE s1.query
= 'SELECT * FROM select_t1();'
```


版权所有 © 华为云计算技术有限公司

```
gaussdb-# UNION ALL
gaussdb-# SELECT s2.query, s2.unique_sql_id FROM dbe_perf.summary_statement s1,
dbe_perf.summary_statement s2 WHERE s1.query = 'SELECT * FROM select_t1();' AND
s1.unique_sql_id = s2.parent_unique_sql_id;
           query            | unique_sql_id
----------------------------+---------------
 SELECT * FROM select_t1(); |    1861705880
 SELECT * FROM t1           |    4115569441
(2 rows)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 此前的analyze次数。
 analyze_count
---------------
             4
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(1861705880); -- 顶层语句不包含基表。
 nspname | tabname
---------+---------
(0 rows)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 以上操作并没有触发t1表的analyze。
 analyze_count
---------------
             4
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(4115569441); -- 子语句包含基表。
 nspname | tabname
---------+---------
 public  | t1
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't1'; -- 确认以上操作完成了一次analyze。
 analyze_count
---------------
             5
(1 row)

-- 5.查询dbe_perf.statement，并对分区表使用all_complete模式进行统计信息收集。
gaussdb=# SELECT * FROM t2;
 c1
----
  1
  2
(2 rows)

gaussdb=# SELECT query, unique_sql_id FROM dbe_perf.summary_statement WHERE query LIKE
'%SELECT * FROM t2%';
       query        | unique_sql_id
--------------------+---------------
 SELECT * FROM t2; |     909667705
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't2'; -- 此前的analyze次数。
 analyze_count
---------------
             1
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(909667705, true); -- 使用all_complete模式，每个分
区会采集更大的样本。
 nspname | tabname
---------+---------
 public  | t2
```


版权所有 © 华为云计算技术有限公司

```
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't2'; -- 确认以上操作完成了一次analyze。
 analyze_count
---------------
             2
(1 row)

-- 6.用node_name筛选，仅查询当前CN节点的SQL记录并收集相关统计信息，假设当前连接节点为
data_coordinator1。
gaussdb=# SELECT node_name, node_type, node_port FROM pgxc_node;
     node_name     | node_type | node_port
-------------------+-----------+-----------
 data_coordinator1 | C         |     36415
 data_coordinator2 | C         |     36419
 data_coordinator3 | C         |     36423
 data_datanode1    | D         |     36815
 data_datanode2    | D         |     36819
 data_datanode3    | D         |     36823
(6 rows)

gaussdb=# select * from t2;
 c1
----
  1
  2
(2 rows)

gaussdb=# SELECT unique_sql_id, node_name FROM dbe_perf.summary_statement WHERE query LIKE
'%SELECT * FROM t2%'; -- 对应的语句记录在CN1节点上。
 unique_sql_id |     node_name
---------------+-------------------
      57714958 | data_coordinator1
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't2'; -- 此前的analyze次数。
 analyze_count
---------------
             2
(1 row)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(57714958, true, 'data_coordinator1');
 nspname | tabname
---------+---------
 public  | t2
(1 row)

gaussdb=# SELECT analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public' AND
relname = 't2'; -- 确认以上操作完成了一次analyze。
 analyze_count
---------------
             3
(1 row)

-- 7.在dbe_perf.statement中查询并收集，涉及多张表的语句。
gaussdb=# SELECT * FROM t1, t2 LIMIT 1;
 c1 | c1
----+----
  1 |  1
(1 row)

gaussdb=# SELECT query, unique_sql_id FROM dbe_perf.summary_statement WHERE query LIKE
'%SELECT * FROM t1, t2 LIMIT%';
             query              | unique_sql_id
--------------------------------+---------------
 SELECT * FROM t1, t2 LIMIT ?; |    2763456878
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT relname, analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public'
AND relname IN ('t1', 't2');
 relname | analyze_count
---------+---------------
 t1      |             5
 t2      |             3
(2 rows)

gaussdb=# SELECT * FROM dbe_stats.analyze_sqlid(2763456878);
 nspname | tabname
---------+---------
 public  | t1
 public  | t2
(2 rows)

gaussdb=# SELECT relname, analyze_count FROM pg_stat_all_tables WHERE schemaname = 'public'
AND relname IN ('t1', 't2');
 relname | analyze_count
---------+---------------
 t1      |             6
 t2      |             4
(2 rows)

-- 后置操作。
gaussdb=# DROP PROCEDURE IF EXISTS select_t1;
DROP PROCEDURE
gaussdb=# DROP TABLE IF EXISTS t1;
DROP TABLE
gaussdb=# DROP TABLE IF EXISTS t2;
DROP TABLE
```

- dbe_stats_tools.py

  该python脚本用于简化库级统计信息的导出和导入流程：

  - 导出统计信息时，需要具备相应接口或命令（DBE_STATS.CREATE_STAT_TABLE、DBE_STATS.EXPORT_DATABASE_STATS、\COPY TO、DBE_STATS.DROP_STAT_TABLE）的权限，以及指定导出目录的读写权限。
  - 导入统计信息时，需要具备相应接口或命令（DBE_STATS.CREATE_STAT_TABLE、DBE_STATS.EXPORT_DATABASE_STATS、\COPY FROM、DBE_STATS.DROP_STAT_TABLE）的权限，以及指定导出目录下文件的读取权限。
  - 导出语句元信息的收集项包括：
    - 输入的SQL语句文本。
    - 在hint生效与不生效场景下，对SQL语句执行EXPLAIN操作得到的信息。
    - 所在DATABASE中所有对象的定义。
    - 与计划调优相关的所有GUC参数配置，不包含应用的会话级别配置。
    - SQL语句所引用表对象的当前统计信息与历史统计信息（仅在生成计划为stream类型时导出此项）。

    导出语句元信息时，需要数据库的系统管理员权限，以及指定目录的读写权限。

  说明

  M-Compatibility兼容模式下不支持该工具。

版权所有 © 华为云计算技术有限公司

表 2-400 dbe_stats_tools.py 接口参数说明

参数	是否允许为空	描述
--mode/-m	否	取值范围：dump、restore、explain、explain_analyze。 ● dump：导出。 ● restore：导入。 ● explain：导出语句元信息，不实际执行语句。 ● explain_analyze：导出语句元信息，实际执行语句。
--database/-d	否	导出或导入的目标数据库名字。
--port/-p	否	数据库CN节点端口号。取值范围是[1, 65535]。
--user/-U	是	用户名，如果不输入-U参数，则会获取当前用户名作为入参。
--passwd/-W	是	用户密码。如果指定该选项，需要单独输入密码。
--host/-H	是	导出或导入目标数据库节点的IP。
--set/-s	是	在语句元信息收集模式下，设置会话级别GUC参数。可指定多次，对应多个GUC参数设置。
--command/-c	否	在语句元信息收集模式下，指定SQL文本。要求是单条可执行EXPLAIN的SQL语句，且不能带有EXPLAIN关键字。explain_analyze模式下仅支持SELECT语句。
--file/-f	否	在语句元信息收集模式下，指定存放SQL文本的文件路径。要求是单条可执行EXPLAIN的SQL语句，且不能带有EXPLAIN关键字。explain_analyze模式下仅支持SELECT语句。
--output/-o	否	存放导出统计信息的目录。

版权所有 © 华为云计算技术有限公司

说明

- 统计信息导出后，会在--output参数所指定的目录下生成gs_stats.meta（元数据文件）和gs_stats.csv（指定DATABASE用户表的统计信息文件）。
- 语句元信息导出后，会在--output参数所指定目录下生成以下目录及文件：
  - query.sql：存放输入的SQL语句文本。
  - metainfo_nohint：该目录默认生成，存放内容是语句无hint生效（当语句本身无hint或工具使其失效）时的信息：
    - explain.log：对SQL语句进行EXPLAIN的结果。
  - metainfo_hinted：仅当语句有hint生效时生成该目录，内容结构与metainfo_nohint一致。
  - objdefs.sql：所在DATABASE中所有对象的定义。
  - querytuning.conf：与计划调优相关的所有GUC参数配置，不包含应用的会话级别配置。
  - currstats：SQL语句所引用表对象的当前统计信息，目录结构与dump模式的输出一致。
  - histstats：SQL语句所引用表对象的历史统计信息，该目录下每一个histstats_<timestamp>子目录对应某个时间戳的统计信息快照，子目录下的文件结构与dump模式的输出一致。histstats下最多拥有50个历史统计信息快照，如果DATABASE中实际查询的快照总数超出该数目，输出的50个历史统计信息快照则为等距采样的结果。

示例一：导出导入统计信息流程。

```
# 导出统计信息流程示例。
[omm@localhost ~]$ python3 dbe_stats_tools.py -m dump -d db1 -p 8000 --output ./stats
export table name: "export_table_1749782634"
execute: "gsql -d db1 -p 8000 -U omm -c "select 1 from pg_catalog.pg_class where relname =
'export_table_1749782634' and relnamespace = 2200;""
execute: "gsql -d db1 -p 8000 -U omm -c "START TRANSACTION;call
dbe_stats.create_stat_table(ownname => 'public', stattab => 'export_table_1749782634');call
dbe_stats.export_database_stats(stattab=>'export_table_1749782634',statown=>'public');\COPY
public.export_table_1749782634TO '/home/omm/stats/gs_stats.csv' WITH (FORMAT CSV,
HEADER);call dbe_stats.drop_stat_table(ownname => 'public', stattab =>
'export_table_1749782634');COMMIT;""
execute: "gsql -d db1 -p 8000 -U omm -c "select pg_catalog.gs_get_current_version() from dual;""
deployment: "M"
execute: "gsql -d db1 -p 8000 -U omm -c "select count(1) from pg_catalog.pgxc_node where
node_type = 'D';""
datanode number: "6"
dump complete!

# 导出统计信息后，stats目录下会生成元数据文件和统计信息的dump文件。
[omm@localhost ~]$ ls stats/
gs_stats.csv  gs_stats.meta

# 导入统计信息流程示例。
[omm@localhost ~]$ python3 dbe_stats_tools.py -m restore -d db1 -p 8000 --output ./stats
export table name: "export_table_1749782643"
gs_stats meta file: "/home/omm/stats/gs_stats.meta"
deployment: "M"
execute: "gsql -d db1 -p 8000 -U omm -c "select pg_catalog.gs_get_current_version() from dual;""
datanode number: "6"
execute: "gsql -d db1 -p 8000 -U omm -c "select count(1) from pg_catalog.pgxc_node where
node_type = 'D';""
execute: "gsql -d db1 -p 8000 -U omm -c "select 1 from pg_catalog.pg_class where relname =
'export_table_1749782643' and relnamespace = 2200;""
execute: "gsql -d db1 -p 8000 -U omm -c "START TRANSACTION;call
dbe_stats.create_stat_table(ownname => 'public', stattab => 'export_table_1749782643');\COPY
public.export_table_1749782643 FROM '/home/omm/stats/gs_stats.csv' WITH (FORMAT CSV,
HEADER);call
dbe_stats.import_database_stats(stattab=>'export_table_1749782643',statown=>'public');call
dbe_stats.drop_stat_table(ownname => 'public', stattab => 'export_table_1749782643');COMMIT;""
restore complete!
```


版权所有 © 华为云计算技术有限公司

示例二：通过语句元信息复现SQL计划不优问题。本示例包含了一个经典的计划不优问题：在数据变更后，未及时收集统计信息，导致选错索引的计划。

前置操作：建表、插入数据、复现慢SQL业务。

```
# 示例中使用的db1为ORA-Compatibility兼容模式数据库，请根据实际需要替换数据库名字。
# 示例中执行命令的节点为集群节点。
[omm@localhost ~]$ gsql -d db1 -p 8000 -r

-- 1.为避免基数反馈、计划复用、语句下推等对该慢SQL问题复现产生影响，设置以下GUC参数。
db1=# SET enable_feedback_cardest = OFF;
SET
db1=# SET cursor_sharing = OFF;
SET
db1=# SET enable_fast_query_shipping = OFF;
SET
db1=# SET explain_perf_mode = 'pretty';
SET

-- 2.预置数据，s表仅用于数据生成，t表是最终查询的业务表。
db1=# DROP TABLE IF EXISTS s;
NOTICE:  table "s" does not exist, skipping
DROP TABLE
db1=# CREATE TABLE s (
    i INT
) DISTRIBUTE BY HASH (i);
CREATE TABLE
db1=# INSERT INTO s SELECT generate_series(1, 550000);
INSERT 0 550000
db1=# ANALYZE s;
ANALYZE
db1=# DROP TABLE IF EXISTS t;
NOTICE:  table "t" does not exist, skipping
DROP TABLE
db1=# CREATE TABLE t (
    idate INT,
    no    INT
) DISTRIBUTE BY HASH (idate) PARTITION BY RANGE (idate) (
    PARTITION p1 VALUES LESS THAN (200),
    PARTITION p2 VALUES LESS THAN (400),
    PARTITION p3 VALUES LESS THAN (MAXVALUE)
);
CREATE TABLE
db1=# CREATE INDEX t_i  ON t(idate) LOCAL;
CREATE INDEX
db1=# CREATE INDEX t_no ON t(no)    LOCAL;
CREATE INDEX
db1=# INSERT INTO t
SELECT floor(random() * 540) AS idate,
       floor(random() * 30)  AS no
  FROM s
 WHERE i <= 540000;
INSERT 0 540000
db1=# ANALYZE t;
ANALYZE

-- 3.插入新数据，但是不重新执行ANALYZE。
db1=# INSERT INTO t
SELECT floor(540 + random() * 10) AS idate,
       floor(random() * 30)       AS no
  FROM s
 WHERE i <= 1000000;
INSERT 0 550000

-- 4.查询新增部分数据（默认计划 vs hint计划）。
db1=# EXPLAIN ANALYZE
SELECT *
  FROM t
 WHERE idate >= 540 AND idate < 550 AND no = 1;
 id |                  operation                  |       A-time       | A-rows | E-rows | Peak Memory | A-width | E-
```

 版权所有 © 华为云计算技术有限公司

```
width | E-costs
----+----------------------------------------------+-------------------+--------+--------+-------------+---------
+---------+---------
  1 | ->  Streaming (type: GATHER)                 | 144.711           |  18204 |      1 | 81KB        |         |
8 | 2.60
  2 |    ->  Partitioned Index Scan using t_i on t | [100.696,130.653] |  18204 |      1 | [78KB,78KB]
|         |       8 | 2.47
(2 rows)

     Predicate Information (identified by plan id)
-----------------------------------------------------------
   2 --Partitioned Index Scan using t_i on t
         Index Cond: ((idate >= 540) AND (idate < 550))
         Filter: (no = 1)
         Rows Removed by Filter: 531796
         Selected Partitions:  3
(5 rows)

 Memory Information (identified by plan id)
--------------------------------------------
 Coordinator Query Peak Memory:
         Query Peak Memory: 1MB
 Datanode:
         Max Query Peak Memory: 0MB
         Min Query Peak Memory: 0MB
(5 rows)

                        User Define Profiling
---------------------------------------------------------------------------
 Plan Node id: 1  Track name: coordinator get datanode connection
  (actual time=[0.062, 0.062], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator serialize plan
  (actual time=[1.085, 1.085], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator send begin command
  (actual time=[0.003, 0.003], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator start transaction and send query
  (actual time=[0.053, 0.053], calls=[1, 1])
(8 rows)

                             ====== Query Summary =====
-------------------------------------------------------------------------------------------
 Datanode executor start time [dn_6001_6002_6003, dn_6004_6005_6006]: [0.229 ms,0.261 ms]
 Datanode executor run time [dn_6007_6008_6009, dn_6001_6002_6003]: [111.230 ms,144.077 ms]
 Datanode executor end time [dn_6004_6005_6006, dn_6001_6002_6003]: [0.012 ms,0.022 ms]
 Coordinator executor start time: 0.095 ms
 Coordinator executor run time: 146.583 ms
 Coordinator executor end time: 0.044 ms
 Planner runtime: 0.474 ms
 Plan size: 3857 byte
 Query Id: 72339069015809757
 Total runtime: 146.743 ms
(10 rows)

db1=# EXPLAIN ANALYZE
SELECT /*+ indexscan(t t_no) */ *
  FROM t
 WHERE idate >= 540 AND idate < 550 AND no = 1;
 id |                   operation                   |    A-time     | A-rows | E-rows | Peak Memory | A-width | E-
width | E-costs
----+-----------------------------------------------+---------------+--------+--------+-------------+---------
+---------+---------
  1 | ->  Streaming (type: GATHER)                  | 25.360        |  18204 |  17652 | 81KB        |         |
8 | 4265.30
  2 |    ->  Partitioned Index Scan using t_no on t | [5.622,8.476] |  18204 |  17652 | [78KB,78KB] |
|       8 | 3529.80
(2 rows)

   Predicate Information (identified by plan id)
-------------------------------------------------------
```


版权所有 © 华为云计算技术有限公司

```
  2 --Partitioned Index Scan using t_no on t
        Index Cond: (no = 1)
        Filter: ((idate >= 540) AND (idate < 550))
        Rows Removed by Filter: 4554
        Selected Partitions:  3
(5 rows)

 Memory Information (identified by plan id)
--------------------------------------------
 Coordinator Query Peak Memory:
        Query Peak Memory: 1MB
 Datanode:
        Max Query Peak Memory: 0MB
        Min Query Peak Memory: 0MB
(5 rows)

                        User Define Profiling
---------------------------------------------------------------------------
 Plan Node id: 1  Track name: coordinator get datanode connection
  (actual time=[0.812, 0.812], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator serialize plan
  (actual time=[1.924, 1.924], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator send begin command
  (actual time=[0.003, 0.003], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator start transaction and send query
  (actual time=[0.058, 0.058], calls=[1, 1])
(8 rows)

                             ====== Query Summary =====
-------------------------------------------------------------------------------------------
 Datanode executor start time [dn_6004_6005_6006, dn_6007_6008_6009]: [0.397 ms,0.584 ms]
 Datanode executor run time [dn_6004_6005_6006, dn_6007_6008_6009]: [12.902 ms,21.639 ms]
 Datanode executor end time [dn_6004_6005_6006, dn_6007_6008_6009]: [0.011 ms,0.019 ms]
 Coordinator executor start time: 0.165 ms
 Coordinator executor run time: 26.601 ms
 Coordinator executor end time: 0.038 ms
 Planner runtime: 1.037 ms
 Plan size: 3886 byte
 Query Id: 72339069015811437
 Total runtime: 26.853 ms
(10 rows)

db1=# \q
```

场景一：通过-c选项收集带HINT SQL语句的元信息，对比HINT生效与不生效情况下的计划信息。

```
# 指定带hint的SQL文本，收集其元信息。
[omm@localhost ~]$ rm -rf ./hint_metainfo
[omm@localhost ~]$ python3 $GAUSSHOME/bin/dbe_stats_tools.py -m explain_analyze -d db1 -p
8000 -s enable_fast_query_shipping=off -c 'SELECT /*+ indexscan(t t_no) */ * FROM t WHERE idate >=
540 AND idate < 550 AND no = 1;' -o ./hint_metainfo
execute: "gsql -d db1 -p 8000 -U omm -q -f -
START TRANSACTION READ ONLY;
set enable_fast_query_shipping=E'off';
set explain_perf_mode = pretty; set cursor_sharing = off;
explain (metadata on, tables on, analyze on) SELECT /*+ indexscan(t t_no) */ * FROM t WHERE idate
>= 540 AND idate < 550 AND no = 1;
ROLLBACK;"
execute: "gsql -d db1 -p 8000 -U omm -q -f -
START TRANSACTION READ ONLY;
set enable_fast_query_shipping=E'off';
set explain_perf_mode = pretty; set cursor_sharing = off;
set parser_ignore_hints = on;
explain (metadata on, tables on, analyze on) SELECT /*+ indexscan(t t_no) */ * FROM t WHERE idate
>= 540 AND idate < 550 AND no = 1;
ROLLBACK;"
explaining query completed!
dumping query tuning settings completed!
dumping object definition completed!
```


版权所有 © 华为云计算技术有限公司

```
dumping current statistics completed!
dumping historical statistics completed!

# 此时得到该语句在hint生效与不生效时的计划信息，以及其它可用于复现的信息。
[omm@localhost ~]$ ls -R ./hint_metainfo/
./hint_metainfo/:
currstats  histstats  metainfo_hinted  metainfo_nohint  objdefs.sql  query.sql  querytuning.conf

./hint_metainfo/currstats:
gs_stats.csv  gs_stats.meta

./hint_metainfo/histstats:
histstats_2026-07-21_15:34:04.161597+08  histstats_2026-07-21_15:35:43.781149+08
histstats_2026-07-21_15:35:43.970977+08  histstats_2026-07-21_15:35:44.108525+08
histstats_2026-07-21_15:35:44.255638+08

'./hint_metainfo/histstats/histstats_2026-07-21_15:34:04.161597+08':
gs_stats.csv  gs_stats.meta

'./hint_metainfo/histstats/histstats_2026-07-21_15:35:43.781149+08':
gs_stats.csv  gs_stats.meta

'./hint_metainfo/histstats/histstats_2026-07-21_15:35:43.970977+08':
gs_stats.csv  gs_stats.meta

'./hint_metainfo/histstats/histstats_2026-07-21_15:35:44.108525+08':
gs_stats.csv  gs_stats.meta

'./hint_metainfo/histstats/histstats_2026-07-21_15:35:44.255638+08':
gs_stats.csv  gs_stats.meta

./hint_metainfo/metainfo_hinted:
explain.log

./hint_metainfo/metainfo_nohint:
explain.log
```

场景二：通过-f选项收集对应语句的元信息，并尝试在测试库中复现当前与历史执行计划。

```
# 通过gsql连接，对t表执行ANALYZE，更新其统计信息。
[omm@localhost ~]$ gsql -d db1 -p 8000 -r

db1=# SET enable_feedback_cardest = OFF;
SET
db1=# SET cursor_sharing = OFF;
SET
db1=# SET enable_fast_query_shipping = OFF;
SET
db1=# SET explain_perf_mode = 'pretty';
SET

-- 执行ANALYZE解决慢SQL问题。
db1=# ANALYZE t;
ANALYZE

-- 重新收集统计信息后，计划默认最优。
db1=# EXPLAIN ANALYZE
SELECT *
  FROM t
 WHERE idate >= 540 AND idate < 550 AND no = 1;
 id |                    operation                    |    A-time     | A-rows | E-rows |  Peak Memory  | A-width |
E-width | E-costs
----+-------------------------------------------------+---------------+--------+--------+---------------
+---------+---------+---------
  1 | ->  Streaming (type: GATHER)                    | 25.861        |  18204 |  18636 | 82KB          |
|       8 | 4204.50
  2 |    ->  Partitioned Bitmap Heap Scan on t        | [8.731,9.053] |  18204 |  18636 | [23KB,23KB]
|         |       8 | 3428.00
  3 |       ->  Partitioned Bitmap Index Scan using t_no | [1.066,1.100] |  22758 |  23400 |
```


版权所有 © 华为云计算技术有限公司

```
[178KB,200KB] |         |       0 | 60.95
(3 rows)

  Predicate Information (identified by plan id)
-----------------------------------------------------
  2 --Partitioned Bitmap Heap Scan on t
        Recheck Cond: (no = 1)
        Filter: ((idate >= 540) AND (idate < 550))
        Rows Removed by Filter: 4554
        Selected Partitions:  3
  3 --Partitioned Bitmap Index Scan using t_no
        Index Cond: (no = 1)
(7 rows)

 Memory Information (identified by plan id)
-----------------------------------------------
 Coordinator Query Peak Memory:
        Query Peak Memory: 1MB
 Datanode:
        Max Query Peak Memory: 0MB
        Min Query Peak Memory: 0MB
(5 rows)

                        User Define Profiling
--------------------------------------------------------------------------------
 Plan Node id: 1  Track name: coordinator get datanode connection
  (actual time=[1.722, 1.722], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator serialize plan
  (actual time=[1.236, 1.236], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator send begin command
  (actual time=[0.003, 0.003], calls=[1, 1])
 Plan Node id: 1  Track name: Coordinator start transaction and send query
  (actual time=[0.048, 0.048], calls=[1, 1])
(8 rows)

                             ====== Query Summary =====
-------------------------------------------------------------------------------------------
 Datanode executor start time [dn_6001_6002_6003, dn_6007_6008_6009]: [0.388 ms,0.550 ms]
 Datanode executor run time [dn_6001_6002_6003, dn_6007_6008_6009]: [19.315 ms,22.296 ms]
 Datanode executor end time [dn_6004_6005_6006, dn_6007_6008_6009]: [0.022 ms,0.033 ms]
 Coordinator executor start time: 0.113 ms
 Coordinator executor run time: 27.502 ms
 Coordinator executor end time: 0.032 ms
 Planner runtime: 0.594 ms
 Plan size: 3924 byte
 Query Id: 72339069015818205
 Total runtime: 27.669 ms
(10 rows)

db1=# \q

# 通过指定SQL文本文件的方式，收集语句元信息。
[omm@localhost ~]$ echo 'SELECT * FROM t WHERE idate >= 540 AND idate < 550 AND no = 1;' > ./
tmp.sql
[omm@localhost ~]$ python3 $GAUSSHOME/bin/dbe_stats_tools.py -m explain_analyze -d db1 -p
8000 -s enable_fast_query_shipping=off -f ./tmp.sql -o ./hist_metainfo
execute: "gsql -d db1 -p 8000 -U omm -q -f -
START TRANSACTION READ ONLY;
set enable_fast_query_shipping=E'off';
set explain_perf_mode = pretty; set cursor_sharing = off;
explain (metadata on, tables on, analyze on) SELECT * FROM t WHERE idate >= 540 AND idate < 550
AND no = 1;
ROLLBACK;"
explaining query completed!
dumping query tuning settings completed!
dumping object definition completed!
dumping current statistics completed!
dumping historical statistics completed!
```


版权所有 © 华为云计算技术有限公司

```
# 此时导出的语句元信息有多个历史统计信息快照。
[omm@localhost ~]$ ls -R ./hist_metainfo
./hist_metainfo:
currstats  histstats  metainfo_nohint  objdefs.sql  query.sql  querytuning.conf

./hist_metainfo/currstats:
gs_stats.csv  gs_stats.meta

./hist_metainfo/histstats:
histstats_2026-07-21_15:34:04.161597+08  histstats_2026-07-21_15:35:43.970977+08
histstats_2026-07-21_15:35:44.255638+08
histstats_2026-07-21_15:35:43.781149+08  histstats_2026-07-21_15:35:44.108525+08
histstats_2026-07-21_15:38:27.837751+08

'./hist_metainfo/histstats/histstats_2026-07-21_15:34:04.161597+08':
gs_stats.csv  gs_stats.meta

'./hist_metainfo/histstats/histstats_2026-07-21_15:35:43.781149+08':
gs_stats.csv  gs_stats.meta

'./hist_metainfo/histstats/histstats_2026-07-21_15:35:43.970977+08':
gs_stats.csv  gs_stats.meta

'./hist_metainfo/histstats/histstats_2026-07-21_15:35:44.108525+08':
gs_stats.csv  gs_stats.meta

'./hist_metainfo/histstats/histstats_2026-07-21_15:35:44.255638+08':
gs_stats.csv  gs_stats.meta

'./hist_metainfo/histstats/histstats_2026-07-21_15:38:27.837751+08':
gs_stats.csv  gs_stats.meta

./hist_metainfo/metainfo_nohint:
explain.log

# 创建一个测试库，导入对象定义与统计信息。
[omm@localhost ~]$ gsql -d postgres -p 8000 -c 'DROP DATABASE IF EXISTS db_test;'
NOTICE:  database "db_test" does not exist, skipping
DROP DATABASE
[omm@localhost ~]$ gsql -d postgres -p 8000 -c "CREATE DATABASE db_test WITH dbcompatibility =
'ORA';"
CREATE DATABASE
[omm@localhost ~]$ gsql -d db_test -p 8000 -q -f ./hist_metainfo/objdefs.sql
total time: 386  ms

# 先尝试导入当前统计信息，复现修复后的执行计划。
[omm@localhost ~]$ python3 $GAUSSHOME/bin/dbe_stats_tools.py -m restore -d db_test -p 8000 --
output ./hist_metainfo/currstats
export table name: "export_table_1784619829"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
select pg_catalog.gs_get_current_version() from dual;"
gs_stats meta file: "/home/omm/hist_metainfo/currstats/gs_stats.meta"
deployment: "M"
datanode number: "3"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
select pg_catalog.count(1) from pg_catalog.pgxc_node where node_type = 'D';"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
select 1 from pg_catalog.pg_class where relname = 'export_table_1784619829' and relnamespace =
2200;"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
START TRANSACTION;call dbe_stats.create_stat_table(ownname => 'public', stattab =>
'export_table_1784619829');\COPY public.export_table_1784619829 FROM '/home/omm/
hist_metainfo/currstats/gs_stats.csv' WITH (FORMAT CSV, HEADER);call
dbe_stats.import_database_stats(stattab=>'export_table_1784619829',statown=>'public');call
dbe_stats.drop_stat_table(ownname => 'public', stattab => 'export_table_1784619829');COMMIT;"
restore complete!
[omm@localhost ~]$ gsql -d db_test -p 8000 -c "SET cursor_sharing = OFF; SET
enable_fast_query_shipping = OFF; SET enable_feedback_cardest = OFF; SET explain_perf_mode =
'pretty'; EXPLAIN SELECT * FROM t WHERE idate >= 540 AND idate < 550 AND no = 1;"
```


版权所有 © 华为云计算技术有限公司

```
SET
SET
SET
SET
 id |                       operation                       | E-rows | E-width | E-costs
----+-------------------------------------------------------+--------+---------+---------
  1 | ->  Streaming (type: GATHER)                          |  18636 |       8 | 4204.50
  2 |    ->  Partitioned Bitmap Heap Scan on t              |  18636 |       8 | 3428.00
  3 |       ->  Partitioned Bitmap Index Scan using t_no on |  23400 |       0 | 60.95
(3 rows)

   Predicate Information (identified by plan id)
-----------------------------------------------------
  2 --Partitioned Bitmap Heap Scan on t
        Recheck Cond: (no = 1)
        Filter: ((idate >= 540) AND (idate < 550))
        Selected Partitions:  3
  3 --Partitioned Bitmap Index Scan using t_no
        Index Cond: (no = 1)
(6 rows)

# 再尝试导入历史统计信息，复现修复前的执行计划（选择最接近问题发生时的时间戳）。
[omm@localhost ~]$ python3 $GAUSSHOME/bin/dbe_stats_tools.py -m restore -d db_test -p 8000 --
output ./hist_metainfo/histstats/histstats_2026-07-21_15\:34\:04.161597+08
export table name: "export_table_1784620247"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
select pg_catalog.gs_get_current_version() from dual;"
gs_stats meta file: "/home/omm/hist_metainfo/histstats/histstats_2026-07-21_15:34:04.161597+08/
gs_stats.meta"
deployment: "M"
datanode number: "3"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
select pg_catalog.count(1) from pg_catalog.pgxc_node where node_type = 'D';"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
select 1 from pg_catalog.pg_class where relname = 'export_table_1784620247' and relnamespace =
2200;"
execute: "gsql -d db_test -p 8000 -U omm -q -f -
START TRANSACTION;call dbe_stats.create_stat_table(ownname => 'public', stattab =>
'export_table_1784620247');\COPY public.export_table_1784620247 FROM '/home/omm/
hist_metainfo/histstats/histstats_2026-07-21_15:34:04.161597+08/gs_stats.csv' WITH (FORMAT CSV,
HEADER);call
dbe_stats.import_database_stats(stattab=>'export_table_1784620247',statown=>'public');call
dbe_stats.drop_stat_table(ownname => 'public', stattab => 'export_table_1784620247');COMMIT;"
restore complete!
[omm@localhost ~]$ gsql -d db_test -p 8000 -c 'SET cursor_sharing = OFF; SET
enable_fast_query_shipping = OFF; SET enable_feedback_cardest = OFF; SET explain_perf_mode =
'pretty'; EXPLAIN SELECT * FROM t WHERE idate >= 540 AND idate < 550 AND no = 1;'
SET
SET
SET
SET
 id |                  operation                   | E-rows | E-width | E-costs
----+----------------------------------------------+--------+---------+---------
  1 | ->  Streaming (type: GATHER)                 |      1 |       8 | 2.60
  2 |    ->  Partitioned Index Scan using t_i on t |      1 |       8 | 2.47
(2 rows)

    Predicate Information (identified by plan id)
-----------------------------------------------------
  2 --Partitioned Index Scan using t_i on t
        Index Cond: ((idate >= 540) AND (idate < 550))
        Filter: (no = 1)
        Selected Partitions:  3
(4 rows)
```

后置操作：清理数据库对象与导出目录文件。

```
# 清理数据库对象。
[omm@localhost ~]$ gsql -d db1 -p 8000 -r
db1=# DROP TABLE s CASCADE;
DROP TABLE
```


版权所有 © 华为云计算技术有限公司

```
db1=# DROP TABLE t CASCADE;
DROP TABLE
db1=# DROP DATABASE db_test;
DROP DATABASE
db1=# \q

# 清理导出文件。
[omm@localhost ~]$ rm -rf ./hint_metainfo
[omm@localhost ~]$ rm -rf ./hist_metainfo
```

2.12.2.19 DBE_TASK

接口介绍

高级功能包DBE_TASK支持的接口请参见**表2-401**。

表 2-401 DBE_TASK

接口名称	描述
DBE_TASK.SUBMIT	提交一个定时任务。作业号由系统自动生成。
DBE_TASK.JOB_SUBMIT	功能等同于**DBE_TASK.SUBMIT**，但提供语法兼容参数。
DBE_TASK.ID_SUBMIT	提交一个定时任务。作业号由用户指定。
DBE_TASK.CANCEL	通过作业号来删除定时任务。
DBE_TASK.RUN	运行定时任务。
DBE_TASK.FINISH	禁用或者启用定时任务。
DBE_TASK.UPDATE	修改定时任务的属性，包括任务内容、下次执行时间和执行间隔。
DBE_TASK.CHANGE	功能等同于**DBE_TASK.UPDATE**，但提供语法兼容参数。
DBE_TASK.CONTENT	修改定时任务的内容属性。
DBE_TASK.NEXT_TIME	修改定时任务的下次执行时间属性。
DBE_TASK.INTERVAL	修改定时任务的执行间隔属性。

说明

- DBE_TASK任务调度在分布式环境下不保证任务执行态信息的高可靠性。仅在任务提交的节点上可以查询到任务调度的执行态信息，其他节点查询到的信息可能尚未同步。
- GaussDB数据库提供了高可用、高可靠性、并且更灵活的调度系统：DBE_SCHEDULER接口，推荐分布式环境下使用该接口。

- DBE_TASK.SUBMIT

  DBE_TASK.SUBMIT提交一个系统提供的定时任务。

版权所有 © 华为云计算技术有限公司

DBE_TASK.SUBMIT原型为：

```
DBE_TASK.SUBMIT(
what          IN   TEXT,
next_time     IN   TIMESTAMP DEFAULT sysdate,
interval_time IN   TEXT  DEFAULT 'null',
id            OUT  INTEGER
)RETURN INTEGER;
```

说明

- 当创建一个定时任务（DBE_TASK）时，系统默认将当前数据库和用户名与当前创建的定时任务（DBE_TASK）绑定起来。
- 该接口函数可以通过CALL或SELECT语句进行调用，如果通过CALL语句调用，需要填写出参，如果通过SELECT语句调用，可以不填写出参。如果在存储过程中则需要用通过PERFORM语句调用该接口函数。
- 如果提交的SQL语句任务使用到非PUBLIC的schema，则需要指定表或者函数的schema，或者在执行SQL语句前先执行SET current_schema = xxx;语句。

表 2-402 DBE_TASK.SUBMIT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
what	text	IN	否	要执行的SQL语句。支持一个或多个DDL（不支持DB相关操作）、DML、匿名块、调用存储过程的语句，或这四种类型混合的场景。
next_time	timestamp	IN	否	下次作业运行时间。默认值为当前系统时间（sysdate）。如果是过去时间，在提交作业时表示立即执行。
interval_time	text	IN	是	用来计算下次作业运行时间的时间表达式，可以是interval表达式，也可以是sysdate加上一个numeric值（例如：sysdate +1.0/24）。如果该值为空或字符串"null"，表示只执行一次。执行后，JOB状态STATUS变成'd'，表示不再执行。
id	integer	OUT	否	作业号。范围为1～3000000。当使用SELECT语句调用时，该参数不能添加；当使用CALL语句调用时，该参数必须添加。

说明

当在TASK的参数what中创建用户时，日志会记录密码的明文。因此不建议在TASK任务中创建用户。该接口创建的任务不能保证高可用，建议使用**PKG_SERVICE.SUBMIT_ON_NODES**创建任务，并将JOB执行节点指定为CCN。

示例：

```
gaussdb=# SELECT DBE_TASK.SUBMIT('call pro_xxx();', to_date('20180101','yyyymmdd'),'sysdate+1');
 submit
--------
  31031
(1 row)
```


版权所有 © 华为云计算技术有限公司

```
gaussdb=# SELECT DBE_TASK.SUBMIT('call pro_xxx();', to_date('20180101','yyyymmdd'),'sysdate
+1.0/24');
 submit
--------
    512
(1 row)

gaussdb=# DECLARE
gaussdb-#     jobid int;
gaussdb-# BEGIN
gaussdb$#     PERFORM DBE_TASK.SUBMIT('call pro_xxx();', sysdate, 'interval ''5 minute''', jobid);
gaussdb$# END;
gaussdb$# /
ANONYMOUS BLOCK EXECUTE
```

- DBE_TASK.JOB_SUBMIT

  DBE_TASK.JOB_SUBMIT提交一个系统提供的定时任务。并提供了额外的兼容性参数。

  DBE_TASK.JOB_SUBMIT原型为：

```
DBE_TASK.JOB_SUBMIT(
job           OUT  INTEGER,
what          IN   TEXT,
next_date     IN   TIMESTAMP  DEFAULT sysdate,
job_interval  IN   TEXT       DEFAULT 'null',
no_parse      IN   BOOLEAN    DEFAULT false,
instance      IN   INTEGER    DEFAULT 0,
force         IN   BOOLEAN    DEFAULT false
)RETURN INTEGER;
```

表 2-403 DBE_TASK.JOB_SUBMIT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job	integer	OUT	否	作业号。范围为1～3000000。当使用SELECT语句调用DBE_TASK.JOB_SUBMIT时，该参数可以省略。
what	text	IN	否	要执行的SQL语句。支持一个或多个DDL（不支持DB相关操作）、DML、匿名块、“调用存储过程的语句，或这四种类型混合的场景。
next_date	timestamp	IN	是	下次作业运行时间。默认值为当前系统时间（sysdate）。如果是过去时间，在提交作业时表示立即执行。
job_interval	text	IN	是	用来计算下次作业运行时间的时间表达式，可以是interval表达式，也可以是sysdate加上一个numeric值（例如：sysdate +1.0/24）。如果该值为空或字符串"null"，表示只执行一次。执行后，JOB状态STATUS变成'd'，表示不再执行。
no_parse	boolean	IN	是	默认值false，仅用于语法上的兼容。
instance	integer	IN	是	默认值0，仅用于语法上的兼容。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
force	boolean	IN	是	默认值false，仅用于语法上的兼容。

示例：

```
gaussdb=# DECLARE
gaussdb-#     id integer;
gaussdb-# BEGIN
gaussdb$#     id = DBE_TASK.JOB_SUBMIT(
gaussdb$#         what => 'insert into t1 values (1, 2)',
gaussdb$#         job_interval => 'sysdate + 1' --daily
gaussdb$#     );
gaussdb$# END;
gaussdb$# /
ANONYMOUS BLOCK EXECUTE
```

- DBE_TASK.ID_SUBMIT

  DBE_TASK.ID_SUBMIT与DBE_TASK.SUBMIT语法功能相同，
  DBE_TASK.ID_SUBMIT的所有参数都是入参，第一个参数表示指定的作业号，
  DBE_TASK.SUBMIT最后一个参数是出参，表示系统自动生成的作业号。

  DBE_TASK.ID_SUBMIT原型为：

  ```
  DBE_TASK.ID_SUBMIT(
  id            IN   BIGINT,
  what          IN   TEXT,
  next_time     IN   TIMESTAMP DEFAULT sysdate,
  interval_time IN   TEXT  DEFAULT 'null');
  ```

  示例：

  ```
  gaussdb=# CALL dbe_task.id_submit(101, 'insert_msg_statistic1;', sysdate, 'sysdate+3.0/24');
   id_submit
  -----------


  (1 row)
  ```

- DBE_TASK.CANCEL

  DBE_TASK.CANCEL删除指定的定时任务。

  DBE_TASK.CANCEL原型为：

  ```
  CANCEL(id  IN  INTEGER);
  ```

  表 2-404 DBE_TASK.CANCEL 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	integer	IN	否	指定的作业号。

  示例：

  ```
  gaussdb=# CALL dbe_task.cancel(101);
   cancel
  --------


  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

- DBE_TASK.RUN

  DBE_TASK.RUN运行定时任务。

  DBE_TASK.RUN原型为：

```
DBE_TASK.RUN(
job         IN   BIGINT,
force       IN   BOOLEAN DEFAULT FALSE);
```

  表 2-405 DBE_TASK.RUN 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job	bigint	IN	否	指定的作业号。
force	boolean	IN	是	仅用于语法上的兼容。

  示例：

```
gaussdb=# BEGIN
gaussdb$#     DBE_TASK.ID_SUBMIT(12345, 'insert_msg_statistic1;', sysdate, 'sysdate+3.0/24');
gaussdb$#     DBE_TASK.RUN(12345);
gaussdb$# END;
gaussdb$# /
ANONYMOUS BLOCK EXECUTE
```

- DBE_TASK.FINISH

  DBE_TASK.FINISH禁用或者启用定时任务。

  DBE_TASK.FINISH原型为：

```
DBE_TASK.FINISH(
id          IN   INTEGER,
broken      IN   BOOLEAN,
next_time   IN   TIMESTAMP  DEFAULT  sysdate);
```

  表 2-406 DBE_TASK.FINISH 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	integer	IN	否	指定的作业号。
broken	boolean	IN	否	状态标志位。true代表禁用，false代表启用。设置具体true或false值以更新当前JOB；如果值为空，则不改变原有JOB的状态。
next_time	timestamp	IN	是	下次运行时间，默认为当前系统时间。该参数可以省略，为默认值。 ● 如果参数broken状态为true，则更新该参数为'4000-1-1'。 ● 如果参数broken状态为false，参数next_time不为空值时，更新指定job的next_time值；next_time为空值时，不更新next_time值。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# CALL dbe_task.id_submit(101, 'insert_msg_statistic1;', sysdate, 'sysdate+3.0/24');
 id_submit
-----------

(1 row)

gaussdb=# CALL dbe_task.finish(101, true);
 finish
--------

(1 row)

gaussdb=# CALL dbe_task.finish(101, false, sysdate);
 finish
--------

(1 row)
```

- DBE_TASK.UPDATE

  DBE_TASK.UPDATE修改定时任务的属性，包括任务内容、下次执行时间和执行间隔。

  DBE_TASK.UPDATE原型为：

```
dbe_task.UPDATE(
id              IN     INTEGER,
content         IN     TEXT,
next_time       IN     TIMESTAMP,
interval_time   IN  TEXT);
```

表 2-407 DBE_TASK.UPDATE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	integer	IN	否	指定的作业号。
content	text	IN	是	执行的存储过程名或者SQL语句块。如果该参数为空值，则不更新指定JOB的content值，否则更新指定JOB的content值。
next_time	timestamp	IN	是	下次运行时间。如果该参数为空值，则不更新指定JOB的next_time值，否则更新指定JOB的next_time值。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
interval_time	text	IN	是	用来计算下次作业运行时间的时间表达式。 ● 如果该参数为空值，则不更新指定JOB的interval_time值。 ● 如果该参数不为空值，会校验interval_time是否为有效的时间类型或interval类型，若是则更新指定JOB的interval_time值。 ● 如果为字符串"null"，表示只执行一次。执行后，JOB状态STATUS变成'd'，表示不再执行。

示例：

```
gaussdb=# CALL dbe_task.update(101, 'call userproc();', sysdate, 'sysdate + 1.0/1440');
 update
--------

(1 row)

gaussdb=# CALL dbe_task.update(101, 'insert into tbl_a values(sysdate);', sysdate, 'sysdate +
1.0/1440');
 update
--------

(1 row)
```

- DBE_TASK.CHANGE

  DBE_TASK.CHANGE修改定时任务的属性，包括任务内容、下次执行时间和执行间隔。

  DBE_TASK.CHANGE原型为：

```
DBE_TASK.CHANGE(
job             IN    INTEGER,
what            IN    TEXT       DEFAULT NULL,
next_date       IN    TIMESTAMP  DEFAULT NULL,
job_interval    IN    TEXT       DEFAULT NULL,
instance        IN    INTEGER    DEFAULT NULL,
force           IN    BOOLEAN    DEFAULT false);
```

表 2-408 DBE_TASK.CHANGE 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
job	integer	IN	否	指定的作业号。
what	text	IN	是	执行的存储过程名或者SQL语句块。如果该参数为空值，则不更新指定JOB的what值，否则更新指定JOB的what值。

版权所有 © 华为云计算技术有限公司

参数	类型	入参/出参	是否可以为空	描述
next_date	timestamp	IN	是	下次运行时间。如果该参数为空值，则不更新指定JOB的next_time值，否则更新指定JOB的next_date值。
job_interval	text	IN	是	用来计算下次作业运行时间的时间表达式。 ● 如果该参数为空值，则不更新指定JOB的job_interval值。 ● 如果该参数不为空值，会校验job_interval是否为有效的时间类型或interval类型，若是则更新指定JOB的job_interval值。 ● 如果为字符串"null"，表示只执行一次。执行后，JOB状态STATUS变成'd' ，表示不再执行。
instance	integer	IN	是	仅用于语法上的兼容。
force	boolean	IN	否	仅用于语法上的兼容。

示例：

```
gaussdb=# BEGIN
gaussdb$#     DBE_TASK.CHANGE(
gaussdb$#         job => 101,
gaussdb$#         what => 'insert into t2 values (2);'
gaussdb$#     );
gaussdb$# END;
gaussdb$# /
ANONYMOUS BLOCK EXECUTE
```

- DBE_TASK.CONTENT

  DBE_TASK.CONTENT修改定时任务的任务内容属性。

  DBE_TASK.CONTENT原型为：

```
DBE_TASK.CONTENT(
id          IN     INTEGER,
content      IN TEXT);
```

表 2-409 DBE_TASK.CONTENT 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	integer	IN	否	指定的作业号。
content	text	IN	否	执行的存储过程调用、SQL语句块或程序块。

版权所有 © 华为云计算技术有限公司

说明

- 当content参数是一个或多个可以执行成功的SQL语句、程序块或调用存储过程时，该接口函数才能被执行成功，否则会执行失败。
- 若content参数为一个简单的INSERT、UPDATE等语句，需要在表前加模式名。

示例：

```
gaussdb=# CALL dbe_task.content(101, 'call userproc();');
 content
---------

(1 row)

gaussdb=# CALL dbe_task.content(101, 'insert into tbl_a values(sysdate);');
 content
---------

(1 row)
```

- DBE_TASK.NEXT_TIME

  DBE_TASK.NEXT_TIME修改定时任务的下次执行时间属性。

  DBE_TASK.NEXT_TIME原型为：

```
DBE_TASK.NEXT_TIME(
id          IN    BIGINT,
next_time   IN    TEXT);
```

表 2-410 DBE_TASK.NEXT_TIME 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	bigint	IN	否	指定的作业号。
next_time	text	IN	否	下次运行时间。

说明

如果输入的next_time的值小于当前日期值，该JOB会立即执行一次。

示例：

```
gaussdb=# CALL dbe_task.next_time(101, sysdate);
 next_time
-----------

(1 row)
```

- DBE_TASK.INTERVAL

  DBE_TASK.INTERVAL修改定时任务的执行间隔属性。

  DBE_TASK.INTERVAL原型为：

```
DBE_TASK.INTERVAL(
id              IN      INTEGER,
interval_time       IN    TEXT);
```


版权所有 © 华为云计算技术有限公司

表 2-411 DBE_TASK.INTERVAL 接口参数说明

参数	类型	入参/出参	是否可以为空	描述
id	integer	IN	否	指定的作业号。
interval_time	text	IN	是	用来计算下次作业运行时间的时间表达式。作业间隔需要为有效的时间类型或interval类型。 如果为空值或字符串"null"，表示只执行一次。执行后，JOB状态STATUS变成'd'，表示不再执行。

示例：

```
gaussdb=# CALL dbe_task.interval(101, 'sysdate + 1.0/1440');
 interval
----------

(1 row)

gaussdb=# CALL dbe_task.cancel(101);
 cancel
--------

(1 row)
```

说明

对于指定JOB正在运行状态（即job_status为'r'）时，不允许通过CANCEL、UPDATE、NEXT_TIME、CONTENT、INTERVAL等接口删除或修改JOB的参数信息。

约束说明

- 使用SUMMIT/ID_SUBMIT创建一个新JOB后，该JOB从属于当前COORDINATOR（即该JOB仅在当前COORDINATOR上调度和执行），其他COORDINATOR不会调度和执行该JOB，如果出现COORDINATOR节点故障，将无法保证JOB正常执行。建议使用**PKG_SERVICE.SUBMIT_ON_NODES**接口将JOB执行节点指定为CCN，以保证节点故障时JOB仍然可用。不是所有COORDINATOR都可以查看、修改、删除其他CN创建的JOB。
- JOB只能通过DBE_TASK高级包提供的接口进行创建、更新和删除操作。这是因为高级包的接口会考虑所有CN间JOB信息的同步，以及PG_JOB与PG_JOB_PROC表主键的关联操作。如果通过DML语句对PG_JOB表进行增删改，会导致JOB信息在CN间不一致，以及系统表无法关联变更，从而严重影响JOB内部的管理。
- 由于用户创建的每个任务和CN绑定，当任务运行过程中，该CN发生故障，则该任务的状态无法实时刷新，仍为‘r’状态，需要等CN启动正常后才能刷新为‘s’状态。如果在任务未执行时CN发生故障，则该CN上的任务都得不到正常的调度和执行，需要人为干预让该CN恢复正常，或进行节点删除/替换，JOB才能正常的调度和执行。
- JOB在定时执行过程中，需要在当前JOB所属的CN上实时更新该JOB的运行状态、最近执行开始时间、最近执行结束时间、下次开始时间、失败次数（如果JOB执行失败）等相关参数信息到PG_JOB系统表中，并同步到其他CN，保证JOB信息的一致性。如果其他CN存在节点故障，那么JOB所属CN会同步超时重发的操作，导致

版权所有 © 华为云计算技术有限公司

JOB执行时间变长，但CN间同步超时失败后，原CN上PG_JOB表中JOB的相关信息仍然能正常更新，且JOB能正常执行成功。当故障CN恢复正常后，可能出现该CN上PG_JOB表中当前JOB的执行时间、运行状态等参数与原CN上不一致的情况，需要原CN上再次执行该JOB后才能保证JOB信息的同步。

- 对于并发同时有多个JOB到达执行时间的场景，由于会为每个JOB创建一个线程来执行JOB，由于系统内部启动每个线程的时间会有延迟，因此会导致同时并发执行的JOB开始时间有延迟，每个JOB的延迟时间在0.1ms左右。

2.12.2.20 DBE_UTILITY

接口介绍

高级功能包DBE_UTILITY支持的所有接口请参见**表2-412**。

表 2-412 DBE_UTILITY

接口名称	描述
DBE_UTILITY.FORMAT_ERROR...	输出函数异常的调用堆栈。
DBE_UTILITY.FORMAT_ERROR...	输出函数异常的具体信息。
DBE_UTILITY.FORMAT_CALL_...DBE_UTILITY.FORMAT_CALL_...	输出函数的调用堆栈。
DBE_UTILITY.GET_TIME	输出当前时间，一般用于做差得到执行时长。
DBE_UTILITY.CANONICALIZE	用于给表名字符串做规范。
DBE_UTILITY.COMMA_TO_TABLE	将用逗号隔开的名字列表的字符串转换为PL/SQL表名称列表。
DBE_UTILITY.DB_VERSION	返回数据库的版本号和兼容性版本号。
DBE_UTILITY.EXEC_DDL_STATEMENT	用于执行用户输入的DDL语句。
DBE_UTILITY.EXPAND_SQL_TEXT_PROC	用于展开SQL查询的视图。
DBE_UTILITY.GET_CPU_TIME	返回当前CPU处理时间的测量值。
DBE_UTILITY.GET_ENDIANNESS	用于获取数据库所在平台字节序的大小端信息。
DBE_UTILITY.GET_HASH_VALUE	返回一个给定字符串的hash值。
DBE_UTILITY.GET_SQL_HASH	输出一个给定字符串的hash值，该存储过程在不打开proc_outparam_override时使用。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_UTILITY.IS_BIT_SET	用于检查参数n是否存在于r。
DBE_UTILITY.IS_CLUSTER_DATABASE	用于判断当前数据库是否在数据库集群模式下运行。
DBE_UTILITY.NAME_RESOLVE	解析给定的对象名称，包括同义词翻译和必要的授权检查。
DBE_UTILITY.NAME_TOKENIZE	用于解析a [. b [. c]][@ dblink]形式的名字。
DBE_UTILITY.OLD_CURRENT_SCHEMA	返回当前用户环境下的数据库模式名称。
DBE_UTILITY.OLD_CURRENT_USER	返回当前用户的名称。
DBE_UTILITY.TABLE_TO_COMMA	将PL/SQL中的表名转换为用逗号隔开的名字列表的字符串。
DBE_UTILITY.GET_SQL_HASH_FUNC	功能同DBE_UTILITY.GET_SQL_HASH，该函数在打开proc_outparam_override时使用。

- DBE_UTILITY.FORMAT_ERROR_BACKTRACE

  函数FORMAT_ERROR_BACKTRACE返回在执行过程中出现错误时，出现错误位置的调用堆栈。DBE_UTILITY.FORMAT_ERROR_BACKTRACE函数原型为：

  ```
  DBE_UTILITY.FORMAT_ERROR_BACKTRACE()
  RETURN TEXT;
  ```

- DBE_UTILITY.FORMAT_ERROR_STACK

  函数FORMAT_ERROR_STACK返回在执行过程中出现错误时，出现错误位置的具体信息。DBE_UTILITY.FORMAT_ERROR_STACK函数原型为：

  ```
  DBE_UTILITY.FORMAT_ERROR_STACK()
  RETURN TEXT;
  ```

- DBE_UTILITY.FORMAT_CALL_STACK

  函数FORMAT_CALL_STACK设置输出函数调用堆栈。
  DBE_UTILITY.FORMAT_CALL_STACK函数原型为：

  ```
  DBE_UTILITY.FORMAT_CALL_STACK()
  RETURN TEXT;
  ```

- DBE_UTILITY.GET_TIME

  函数GET_TIME设置输出时间，通常用于做差，单独的返回值没有意义。
  DBE_UTILITY.GET_TIME函数原型为：

  ```
  DBE_UTILITY.GET_TIME()
  RETURN BIGINT;
  ```

- DBE_UTILITY.CANONICALIZE

  存储过程CANONICALIZE用于给表名字符串做规范。该过程处理单个保留字或关键字，并删除单个标识符的空白，以便“table”成为TABLE。
  DBE_UTILITY.CANONICALIZE函数原型为：

  ```
  DBE_UTILITY.CANONICALIZE(
    name      IN   VARCHAR2,
  ```


版权所有 © 华为云计算技术有限公司

```
  canon_name OUT  VARCHAR2,
  canon_len  IN   BINARY_INTEGER DEFAULT 1024
);
```

表 2-413 DBE_UTILITY.CANONICALIZE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
name	VARCHAR2	IN	否	待规范的字符串。
canon_name	VARCHAR2	OUT	是	规范好的字符串。
canon_len	BINARY_INTEGER	IN	是	要规范的字符串长度，默认值为1024（以字节为单位）。若此参数的值小于待规范的字符串的实际长度（字节），则会以字节为粒度截断字符串。

- DBE_UTILITY.COMMA_TO_TABLE

  存储过程COMMA_TO_TABLE将用逗号隔开的名字列表的字符串转换为PL/SQL表名称列表。DBE_UTILITY.COMMA_TO_TABLE函数原型为：

  ```
  DBE_UTILITY.COMMA_TO_TABLE (
    list   IN  VARCHAR2,
    tablen OUT BINARY_INTEGER,
    tab    OUT VARCHAR2[]
  );
  ```

  表 2-414 DBE_UTILITY.COMMA_TO_TABLE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
list	VARCHAR2	IN	否	逗号隔开的名字列表字符串。
tablen	BINARY_INTEGER	OUT	是	列表名字的个数。
tab	VARCHAR2	OUT	是	转换后的表名称列表。

- DBE_UTILITY.DB_VERSION

  存储过程DB_VERSION返回数据库的版本号和兼容性版本号。
  DBE_UTILITY.DB_VERSION函数原型为：

  ```
  DBE_UTILITY.DB_VERSION (
    version OUT VARCHAR2
  );
  ```


版权所有 © 华为云计算技术有限公司

表 2-415 DBE_UTILITY.DB_VERSION 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
version	VARCHAR2	OUT	否	输出参数，表示内部的数据库软件版本信息，是一个字符串。

- DBE_UTILITY.EXEC_DDL_STATEMENT

  说明

  使用EXECUTE DIRECT ON语法执行DBE_UTILITY.EXEC_DDL_STATEMENT函数，如果GUC参数enable_udf_execute_direct的值为on，则可以执行；否则不可以执行。

  存储过程EXEC_DDL_STATEMENT用于执行用户输入的DDL语句。DBE_UTILITY.EXEC_DDL_STATEMENT函数原型为：

```
DBE_UTILITY.EXEC_DDL_STATEMENT (
    parse_string IN TEXT
);
```

表 2-416 DBE_UTILITY.EXEC_DDL_STATEMENT 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
parse_string	TEXT	IN	是	需要执行的DDL语句。

- DBE_UTILITY.EXPAND_SQL_TEXT_PROC

  存储过程EXPAND_SQL_TEXT_PROC用于展开SQL查询的视图，会递归展开视图中的视图对象，一直展开显示到表。DBE_UTILITY.EXPAND_SQL_TEXT_PROC函数原型为：

```
DBE_UTILITY.EXPAND_SQL_TEXT_PROC (
    input_sql_text  IN  CLOB,
    output_sql_text OUT CLOB
);
```

表 2-417 DBE_UTILITY.EXPAND_SQL_TEXT_PROC 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
input_sql_text	CLOB	IN	否	输入的SQL文本。
output_sql_text	CLOB	OUT	否	输出展开视图的SQL文本。

版权所有 © 华为云计算技术有限公司

说明

用户输入的input_sql_text参数中，SQL语句中的对象需要增加schema前缀，否则函数会报无法找到对象的错误。若设置behavior_compat_options参数值为bind_procedure_searchpath，则无需强制指定schema前缀。

- DBE_UTILITY.GET_CPU_TIME

  函数GET_CPU_TIME返回当前CPU处理时间的测量值（以百分之一秒为单位）。
  DBE_UTILITY.GET_CPU_TIME函数原型为：

```
DBE_UTILITY.GET_CPU_TIME()
RETURN BIGINT;
```

- DBE_UTILITY.GET_ENDIANNESS

  函数GET_ENDIANNESS用于获取数据库所在平台字节序的大小端信息。
  DBE_UTILITY. GET_ENDIANNESS函数原型为：

```
DBE_UTILITY.GET_ENDIANNESS
RETURN INTEGER;
```

- DBE_UTILITY.GET_HASH_VALUE

  函数GET_HASH_VALUE返回一个给定字符串的hash值。
  DBE_UTILITY.GET_HASH_VALUE函数原型为：

```
DBE_UTILITY.GET_HASH_VALUE(
   name      IN VARCHAR2(n),
   base      IN INTEGER,
   hash_size IN INTEGER)
RETURN INTEGER;
```

表 2-418 DBE_UTILITY.GET_HASH_VALUE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
name	VARCHAR2	IN	否	待哈希转换的字符串。
base	INTEGER	IN	否	返回的哈希值的起始值。
hash_size	INTEGER	IN	否	哈希映射到的哈希表的大小。

- DBE_UTILITY.GET_SQL_HASH

  存储过程GET_SQL_HASH通过MD5算法输出一个给定字符串的hash值。
  DBE_UTILITY.GET_SQL_HASH函数原型为：

```
DBE_UTILITY.GET_SQL_HASH(
   name       IN  VARCHAR2,
   hash       OUT RAW,
   last4bytes OUT BIGINT
)RETURN BIGINT;
```

表 2-419 DBE_UTILITY.GET_SQL_HASH 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
name	VARCHAR2	IN	否	待哈希转换的字符串。

版权所有 © 华为云计算技术有限公司

参数名称	类型	入参/出参	是否可以为空	描述
hash	RAW	OUT	否	完整的16进制MD5哈希值。
last4bytes	BIGINT	OUT	否	MD5哈希值的最后四字节，以无符号整数形式展现。

说明

设置behavior_compat_options参数值为非proc_outparam_override值后，请调用DBE_UTILITY.GET_SQL_HASH函数，如调用DBE_UTILITY.GET_SQL_HASH_FUNC则会发生赋值不成功。

- DBE_UTILITY.IS_BIT_SET

  函数IS_BIT_SET用于检查参数n是否存在于r。DBE_UTILITY. IS_BIT_SET函数原型为：

```
DBE_UTILITY.IS_BIT_SET (
   r IN RAW,
   n IN INTEGER)
RETURN INTEGER;
```

表 2-420 DBE_UTILITY.IS_BIT_SET 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
r	RAW	IN	否	4字节加上实际的十六进制字符串。
n	INTEGER	IN	否	用于在二进制中判断是否存在该数值。

- DBE_UTILITY.IS_CLUSTER_DATABASE

  函数IS_CLUSTER_DATABASE用于判断当前数据库是否在数据库集群模式下运行。DBE_UTILITY.IS_CLUSTER_DATABASE函数原型为：

```
DBE_UTILITY.IS_CLUSTER_DATABASE
RETURN BOOLEAN;
```

- DBE_UTILITY.NAME_RESOLVE

  存储过程NAME_RESOLVE解析给定的对象名称，包括同义词翻译和必要的授权检查。DBE_UTILITY.NAME_RESOLVE函数原型为：

```
DBE_UTILITY.NAME_RESOLVE (
   name          IN   VARCHAR2,
   context       IN   INTEGER,
   schema        OUT VARCHAR2,
   part1         OUT VARCHAR2,
   part2         OUT VARCHAR2,
   dblink        OUT VARCHAR2,
   part1_type    OUT INTEGER,
   object_number OUT OID
);
```


版权所有 © 华为云计算技术有限公司

表 2-421 DBE_UTILITY.NAME_RESOLVE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
name	VARCHAR2	IN	否	待解析对象名，结构为[[a.]b.]c[@d]。
context	INTEGER	IN	否	待解析对象类型 ● 0 - table ● 1 - function, procedure, package ● 2 - sequence ● 3 - trigger ● 4 - Java Source ● 5 - Java resource ● 6 - Java class ● 7 - type ● 8 - Java shared data ● 9 - index
schema	VARCHAR2	OUT	否	对象所在的模式。
part1	VARCHAR2	OUT	否	名称的第一部分，该字段的类型由part1_type指定。
part2	VARCHAR2	OUT	是	如果该字段不为空，则为子程序名称。
dblink	VARCHAR2	OUT	是	数据库链接。
part1_type	INTEGER	OUT	否	part1的类型： ● 5：synonym ● 7：procedure(top level) ● 8：function(top level) ● 9：package
object_number	OID	OUT	否	对象标识。object_number在ORA数据库中类型为Number，表示对象标识，而GaussDB中对象标识类型为OID，并且不支持Number到OID的隐式转换。

- DBE_UTILITY.NAME_TOKENIZE

  存储过程NAME_TOKENIZE用于解析a [. b [. c]][@ dblink]形式的名字，如果名字带有双引号则去掉，否则变为大写字母。DBE_UTILITY.NAME_TOKENIZE函数原型为：

  ```
  DBE_UTILITY.NAME_TOKENIZE (
  name     IN  VARCHAR2,
  ```


版权所有 © 华为云计算技术有限公司

```
a       OUT VARCHAR2,
b       OUT VARCHAR2,
c       OUT VARCHAR2,
dblink  OUT VARCHAR2,
nextpos OUT INTEGER
);
```

表 2-422 DBE_UTILITY.NAME_TOKENIZE 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
name	VARCHAR2	IN	否	名字，由SQL标识符组成（比如，scott.foo@dblink）。
a	VARCHAR2	OUT	否	名字的第一个token。
b	VARCHAR2	OUT	是	名字的第二个token。
c	VARCHAR2	OUT	是	名字的第三个token。
dblink	VARCHAR2	OUT	是	数据库链接。
nextpos	INTEGER	OUT	否	字符串经过解析后的下一个位置。

- DBE_UTILITY.OLD_CURRENT_SCHEMA

  函数OLD_CURRENT_SCHEMA返回当前用户环境下的数据库模式名称。
  DBE_UTILITY.OLD_CURRENT_SCHEMA函数原型为：

  ```
  DBE_UTILITY.OLD_CURRENT_SCHEMA()
  RETURN VARCHAR;
  ```

- DBE_UTILITY.OLD_CURRENT_USER

  函数OLD_CURRENT_USER返回当前用户的名称。
  DBE_UTILITY.OLD_CURRENT_USER函数原型为：

  ```
  DBE_UTILITY.OLD_CURRENT_USER()
  RETURN TEXT;
  ```

- DBE_UTILITY.TABLE_TO_COMMA

  存储过程TABLE_TO_COMMA将PL/SQL中的表名转换为用逗号隔开的名字列表的字符串。DBE_UTILITY.TABLE_TO_COMMA函数原型为：

  ```
  DBE_UTILITY.TABLE_TO_COMMA (
     tab    IN  VARCHAR2[],
     tablen OUT BINARY_INTEGER,
     list   OUT VARCHAR2
  );
  ```

表 2-423 DBE_UTILITY.TABLE_TO_COMMA 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
tab	VARCHAR2[]	IN	否	包含表名的结构表数组。
tablen	BINARY_INTEGER	OUT	否	结构表中的表个数。
list	VARCHAR2	OUT	否	以逗号分隔表名的字符串。

版权所有 © 华为云计算技术有限公司

- DBE_UTILITY.GET_SQL_HASH_FUNC

  函数GET_SQL_HASH_FUNC通过MD5算法输出一个给定字符串的hash值。
  DBE_UTILITY.GET_SQL_HASH_FUNC函数原型为：

```
DBE_UTILITY.GET_SQL_HASH_FUNC(
    name       IN  VARCHAR2,
    hash       OUT RAW,
    last4bytes OUT BIGINT
);
```

  表 2-424 DBE_UTILITY.GET_SQL_HASH_FUNC 接口说明

参数名称	类型	入参/出参	是否可以为空	描述
name	VARCHAR2	IN	否	待哈希转换的字符串。
hash	RAW	OUT	否	完整的16进制MD5哈希值。
last4bytes	BIGINT	OUT	否	MD5哈希值的最后四字节，以无符号整数形式展现。

  说明

  设置set behavior_compat_options = 'proc_outparam_override' 参数后，请调用DBE_UTILITY.GET_SQL_HASH_FUNC函数，如调用DBE_UTILITY.GET_SQL_HASH则会报参数不匹配异常。

示例

```
-- 示例1
create or replace procedure print_err() as
DECLARE
    a bool;
BEGIN
    a := not_exist;
exception
    when others then
        dbe_output.print_line('err_stack: ' || DBE_UTILITY.FORMAT_ERROR_STACK());
END;
/
CREATE PROCEDURE

call print_err();
-- 预期结果为：
err_stack: 50360452: column "not_exist" does not exist
 print_err
-----------

(1 row)

-- 清理环境
drop procedure print_err;
DROP PROCEDURE

-- 示例2
create or replace procedure print_err() as
DECLARE
    a bool;
```


版权所有 © 华为云计算技术有限公司

```
BEGIN
    a := not_exist;
exception
    when others then
        dbe_output.print_line('backtrace: ' || DBE_UTILITY.FORMAT_ERROR_BACKTRACE());
END;
/
CREATE PROCEDURE

call print_err();
-- 预期结果为：
backtrace: 50360452: PL/pgSQL function print_err() line 5 at assignment

 print_err
-----------

(1 row)

-- 清理环境
drop procedure print_err;
DROP PROCEDURE

-- 示例3
create or replace procedure print_err() as
DECLARE
    a bool;
BEGIN
    a := not_exist;
exception
    when others then
        dbe_output.print_line('call_stack: ');
        dbe_output.print_line(DBE_UTILITY.FORMAT_CALL_STACK());
END;
/
CREATE PROCEDURE

call print_err();
-- 预期结果为：
call_stack:
      3  dbe_utility.format_call_stack()
      9  print_err()

 print_err
-----------

(1 row)

-- 清理环境
drop procedure print_err;
DROP PROCEDURE

-- 示例4
CREATE OR REPLACE PROCEDURE test_get_time1()
AS
declare
    start_time  bigint;
    end_time  bigint;
BEGIN
    start_time:= dbe_utility.get_time ();
    pg_sleep(1);
    end_time:=dbe_utility.get_time ();
    dbe_output.print_line(end_time - start_time);
END;
/
CREATE PROCEDURE

call test_get_time1();
-- 预期结果为：
101
```


版权所有 © 华为云计算技术有限公司

```
 test_get_time1
----------------

(1 row)

-- 清理环境
drop PROCEDURE test_get_time1;
DROP PROCEDURE

-- 示例5
-- 给表名字符串做规范
declare
    cname varchar2(50);
begin
    dbe_utility.canonicalize('seg1', cname, 50);
    dbe_output.put_line(cname);
end;
/
-- 预期结果为：
SEG1
ANONYMOUS BLOCK EXECUTE

-- 示例6
-- 将输入的字符串转换成一个表名的数组
DECLARE
 tab_list VARCHAR2(100) := 't1,t2';
 len BINARY_INTEGER;
 tab varchar2[];
BEGIN
 dbe_output.put_line('table list is: ' || tab_list);
 dbe_utility.comma_to_table(tab_list, len, tab);
END;
/
-- 预期结果为：
table list is: t1,t2
ANONYMOUS BLOCK EXECUTE

-- 示例7
-- 查看数据库的版本号和兼容性版本号
declare
    v_version varchar2;
begin
    dbe_utility.db_version(v_version);
    v_version:=left(v_version, 8);
    dbe_output.print_line('version:' || v_version);
end;
/
-- 预期结果为：
version:gaussdb
ANONYMOUS BLOCK EXECUTE

-- 示例8
-- 查看当前CPU处理时间的测量值
DECLARE
    cputime NUMBER;
BEGIN
    cputime := dbe_utility.get_cpu_time();
    dbe_output.put_line('cpu time：' || cputime);
END;
/
-- 预期结果为（数值并非固定）：
cpu time：70179
ANONYMOUS BLOCK EXECUTE

-- 示例9
-- 获取数据库所在平台字节序的大小端信息
BEGIN
    dbe_output.PUT_LINE(dbe_utility.GET_ENDIANNESS());
END;
```


版权所有 © 华为云计算技术有限公司

```
/
-- 预期结果为：
2
ANONYMOUS BLOCK EXECUTE

-- 示例10
SET system_function_version TO DEFAULT;
SET
-- 获取一个给定字符串的hash值
DECLARE
    result NUMBER(28);
BEGIN
    result := dbe_utility.get_hash_value('hello',10,10);
    dbe_output.put_line(result);
END;
/
-- 预期结果为：
15
ANONYMOUS BLOCK EXECUTE

-- 示例11
-- 判断当前数据库是否为集群模式
DECLARE
    is_cluster BOOLEAN;
BEGIN
    is_cluster := dbe_utility.IS_CLUSTER_DATABASE();
    dbe_output.put_line('CLUSTER DATABASE: ' || CASE WHEN is_cluster THEN 'TRUE' ELSE 'FALSE' END);
END;
/
-- 预期结果为：
CLUSTER DATABASE: TRUE
ANONYMOUS BLOCK EXECUTE

-- 示例12
-- 获取当前用户环境下的数据库模式名称
DECLARE
    schm varchar2(100);
BEGIN
    schm := dbe_utility.old_current_schema();
    dbe_output.put_line('current schema: ' || schm);
END;
/
-- 预期结果为（结果为当前数据库的模式名，并非固定）：
current schema: public
ANONYMOUS BLOCK EXECUTE

-- 示例13
-- 获取当前用户名称
select dbe_utility.old_current_user() from sys_dummy;
-- 预期结果为（结果为当前数据库的用户名，并非固定）：
 old_current_user
------------------
 test
(1 row)

-- 示例14
DECLARE
    ddl_str VARCHAR2(255);
BEGIN
    dbe_output.print_line('start to test exec_ddl_statement create table.');
    ddl_str := 'CREATE TABLE test_ddl (COL1 INT)';
    dbe_utility.exec_ddl_statement(ddl_str);
END;
/
-- 预期结果为：
start to test exec_ddl_statement create table.
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'col1' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CONTEXT:  SQL statement "CREATE TABLE test_ddl (COL1 INT)"
```


版权所有 © 华为云计算技术有限公司

```
SQL statement "CALL dbe_utility.exec_ddl_statement(ddl_str)"
PL/pgSQL function inline_code_block line 5 at PERFORM
ANONYMOUS BLOCK EXECUTE

select * from test_ddl;
-- 预期结果为：
 col1
------
(0 rows)

-- 清理环境
drop table test_ddl;
DROP TABLE

-- 示例15
create table t1 (c1 int primary key, c2 int);
insert into t1 values(1,1),(2,1),(3,2),(4,2),(5,3),(6,3);
create view v1 as select * from t1 where c1 > 1;
create view v2 as select c1 from v1 where c2 > 1;
create view v3 as select * from v2 where c1 > 2;

declare
    in_sql clob := 'select * from public.v3';
    out_sql clob;
begin
    dbe_output.print_line('start to test expand_sql_text_proc v3 expend sql text.');
    dbe_utility.expand_sql_text_proc(in_sql, out_sql);
    dbe_output.print_line(out_sql);
end;
/
-- 预期结果为：
start to test expand_sql_text_proc v3 expend sql text.
SELECT c1 FROM (SELECT v2.c1 FROM (SELECT v1.c1 FROM (SELECT t1.c1, t1.c2 FROM public.t1 WHERE
t1.c1 > 1) v1 WHERE v1.c2 > 1) v2 WHERE v2.c1 > 2) v3
ANONYMOUS BLOCK EXECUTE

-- 清理环境
drop table t1 cascade;
-- 预期结果为：
NOTICE:  drop cascades to 3 other objects
DETAIL:  drop cascades to view v1
drop cascades to view v2
drop cascades to view v3


-- 示例16
declare
    name varchar2;
    hash raw;
    last4bytes bigint;
BEGIN
    name := '';
    -- return correctly (D41D8CD98F00B204E9800998ECF8427E, 2118318316)
    dbe_utility.get_sql_hash(name,hash,last4bytes);
    raise notice '%',hash;
    raise notice '%',last4bytes;
END;
/
-- 预期结果为：
NOTICE:  D41D8CD98F00B204E9800998ECF8427E
NOTICE:  2118318316
ANONYMOUS BLOCK EXECUTE

-- 示例17
declare
    bitchar raw(8);
begin
    bitchar := '2111111f';
    dbe_output.print('test '|| bitchar ||' bit is_bit_set value from 1 to 32 bit: ');
```

 版权所有 © 华为云计算技术有限公司

```
    for i in reverse 32 .. 1 loop
        dbe_output.print(dbe_utility.is_bit_set(bitchar, i));
    end loop;
    dbe_output.print_line('.');
end;
/
-- 预期结果为：
test 2111111F bit is_bit_set value from 1 to 32 bit: 00100001000100010001000100011111.
ANONYMOUS BLOCK EXECUTE

-- 示例18
create or REPLACE PROCEDURE p_test_pk ( -- for print result
    name in varchar2,
    type in integer
)
as
    schema varchar2;
    part1 varchar2;
    part2 varchar2;
    dblink varchar2;
    part1_type integer;
    object_number integer;
begin
    dbe_utility.name_resolve(name,type,schema,part1,part2,dblink,part1_type,object_number);
    raise notice 'schema: % -- part1: % -- part2: % -- dblink: % -- part1_type: %',
schema,part1,part2,dblink,part1_type;
end;
/
CREATE PROCEDURE

declare begin p_test_pk('a.b.c@aa',3); end;
/
-- 预期结果为：
NOTICE:  schema: a -- part1: b -- part2: c -- dblink: aa -- part1_type: 0
CONTEXT:  SQL statement "CALL p_test_pk('a.b.c@aa',3)"
PL/pgSQL function inline_code_block line 1 at PERFORM
ANONYMOUS BLOCK EXECUTE

-- 示例19
DECLARE
    name varchar;
    a    varchar;
    b    varchar;
    c    varchar;
    dblink varchar;
    nextpos INTEGER;
BEGIN
    name := '我.w#sdfsdf.CD';
    DBE_UTILITY.NAME_TOKENIZE(name, a, b, c, dblink, nextpos);
    RAISE INFO E'dbe_utility.name_tokenize parse error: name:%\na:%\nb:%\nc:%\ndblink:%',
    name, a, b, c, dblink;
    IF nextpos <> OCTET_LENGTH(name) THEN
        RAISE INFO E'dbe_utility.name_tokenize length error: name:%\nlength of name:%\nnextpos:%',
        name, OCTET_LENGTH(name), nextpos;
    END IF;
END;
/
-- 预期结果为：
INFO:  dbe_utility.name_tokenize parse error: name:我.w#sdfsdf.CD
a:我
b:W#SDFSDF
c:CD
dblink:<NULL>
ANONYMOUS BLOCK EXECUTE

-- 示例20
DECLARE
list varchar2(50) := 'aabb,ccdd,eeff,gghh';
len_list integer;
```


版权所有 © 华为云计算技术有限公司

```
tab varchar2[];
get_list varchar2(50);
len_tab integer;
BEGIN
dbe_output.print_line('参数列表：' || list);
dbe_utility.comma_to_table(list,len_list,tab);
dbe_output.print_line('参数长度：' || len_list);
FOR i IN 1 .. len_list LOOP
    dbe_output.print_line('列表名称 ' || i || ' : ' || tab(i));
END LOOP;
dbe_output.print_line('调用table_to_comma：');
dbe_utility.table_to_comma(tab,len_tab,get_list );
dbe_output.print_line('输出结果：' || get_list );
dbe_output.print_line('数组长度：' || len_tab);
END;
/
-- 预期结果为：
参数列表：aabb,ccdd,eeff,gghh
参数长度：4
列表名称 1 : aabb
列表名称 2 : ccdd
列表名称 3 : eeff
列表名称 4 : gghh
调用table_to_comma：
输出结果：aabb,ccdd,eeff,gghh
数组长度：4
ANONYMOUS BLOCK EXECUTE

-- 示例21
declare
    name varchar2;
    hash raw;
    last4bytes bigint;
BEGIN
    name := 'hello world';
    -- return correctly(5EB63BBBE01EEED093CB22BB8F5ACDC3, 3285015183)
    dbe_utility.get_sql_hash_func(name,hash,last4bytes);
    raise notice '%',hash;
    raise notice '%',last4bytes;
END;
/
-- 预期结果为：
NOTICE:  3285015183
NOTICE:  <NULL>
ANONYMOUS BLOCK EXECUTE
```

2.12.2.21 DBE_XMLDOM

接口介绍

高级功能包DBE_XMLDOM用于访问XMLType对象，实现DOM(Document Object Model)，用于访问HTML和XML DOCUMENTS API。高级功能包DBE_XMLDOM支持的所有类型请参见**表2-425**，DBE_XMLDOM支持的所有接口请参见**表2-426**。

说明

DBE_XMLDOM高级包在字符集设置为SQL_ASCII的数据库内使用的情况下，输入超出ASCII范围的字符，会导致报错。

表 2-425 DBE_XMLDOM 数据类型说明

类型名称	描述
DOMATTR	实现DOM Attribute接口。

版权所有 © 华为云计算技术有限公司

类型名称	描述
DOMDOCUMENT	实现DOM Document接口。
DOMELEMENT	实现DOM Element接口。
DOMNAMEDNODEMAP	实现DOM Named Node Map接口。
DOMNODELIST	实现DOM Node List接口。
DOMNODE	实现DOM Node接口。
DOMTEXT	实现DOM Text接口。

表 2-426 DBE_XMLDOM 接口参数说明

接口名称	描述
DBE_XMLDOM.APPENDCHILD	将newchild node添加到parent(n)节点最后面，并返回新添加的Node节点。
DBE_XMLDOM.CREATEELEMENT	创建指定名称的DOMELEMENT对象。
DBE_XMLDOM.CREATETEXTNODE	创建DOMTEXT节点。
DBE_XMLDOM.FREEDOCUMENT	释放DOMDOCUMENT节点相关资源。
DBE_XMLDOM.FREEELEMENT	释放DOMELEMENT节点相关资源。
DBE_XMLDOM.FREENODE	释放DOMNODE节点相关资源。
DBE_XMLDOM.FREENODELIST	释放DOMNODELIST节点相关资源。
DBE_XMLDOM.GETATTRIBUTE	按名称返回DOMELEMENT属性的值。
DBE_XMLDOM.GETATTRIBUTES	将DOMNODE节点属性值作为map返回。
DBE_XMLDOM.GETCHILDNODES	将节点下的若干子节点转换成节点列表。
DBE_XMLDOM.GETCHILDRENBYTAGNAME	按名称返回DOMELEMENT的子节点。
DBE_XMLDOM.GETDOCUMENTELEMENT	返回指定DOCUMENT的首个子节点。
DBE_XMLDOM.GETFIRSTCHILD	返回第一个子节点。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_XMLDOM.GETLASTCHILD	返回最后一个子节点。
DBE_XMLDOM.GETLENGTH	获取给定节点中的节点个数。
DBE_XMLDOM.GETLOCALNAME	检索节点的本地名称。
DBE_XMLDOM.GETNAMEDITEM	检索由名称指定的节点。
DBE_XMLDOM.GETNEXTSIBLING	返回该节点的下一个节点。
DBE_XMLDOM.GETNODENAME	返回节点名称。
DBE_XMLDOM.GETNODETYPE	返回节点类型。
DBE_XMLDOM.GETNODEVALUE	此函数用于获取节点的值，具体取决于其类型。
DBE_XMLDOM.GETPARENTNODE	检索此节点的父节点。
DBE_XMLDOM.GETTAGNAME	返回指定DOMELEMENT的标签名称。
DBE_XMLDOM.HASCHILDNODES	检查DOMNODE对象是否拥有任一子节点。
DBE_XMLDOM.IMPORTNODE	复制节点并为该节点指定所属文档。
DBE_XMLDOM.ISNULL	检测节点是否为空。
DBE_XMLDOM.ITEM	返回映射中与索引参数对应的项。
DBE_XMLDOM.MAKEELEMENT	将DOMNODE对象转换为DOMELEMENT类型。
DBE_XMLDOM.MAKENODE	将节点强制转换为DOMNODE类型。
DBE_XMLDOM.NEWDOMDOCUMENT	返回新的DOMDOCUMENT对象。
DBE_XMLDOM.SETATTRIBUTE	按名称设置DOMELEMENT属性的值。
DBE_XMLDOM.SETCHARSET	设置DOMDOCUMENT的CHARSET字符集。
DBE_XMLDOM.SETDOCTYPE	设置DOMDOCUMENT的外部DTD。
DBE_XMLDOM.SETNODEVALUE	此函数用于向DOMNODE对象中设置节点的值。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_XMLDOM.WRITETOBUFFER	将XML节点写入指定缓冲区。
DBE_XMLDOM.WRITETOCLOB	将XML节点写入指定CLOB。
DBE_XMLDOM.WRITETOFILE	将XML节点写入指定文件。
DBE_XMLDOM.GETSESSIONTREENUM	显示当前session中所有类型的dom树的数量。
DBE_XMLDOM.GETDOCTREESINFO	显示document类型的dom树的内存占用、节点数量等统计信息。
DBE_XMLDOM.GETDETAILDOCTREEINFO	显示特定的document变量的各类型节点数量。
DBE_XMLDOM.GETELEMENTSBYTAGNAM	返回匹配TAGNAME的DOMNODELIST节点列表。

- DBE_XMLDOM.APPENDCHILD

  将newchild node添加到parent(n)节点最后面,并返回新添加的Node节点。
  DBE_XMLDOM.APPENDCHILD函数原型为：

```
DBE_XMLDOM.APPENDCHILD(
   n IN DOMNode,
   newchild IN DOMNode)
RETURN DOMNODE;
```

  表 2-427 DBE_XMLDOM.APPENDCHILD 接口参数说明

参数	描述
n	被添加的node。
newchild	添加的新node。

  说明

  - DOCUMENT类型节点下APPEND ATTR类型节点会报“operation not support”错误，ORA数据库在此场景下不报错，但实际并没有挂载成功。
  - ATTR类型节点下APPEND ATTR类型节点会报“operation not support”错误，ORA数据库在此场景下不报错，但实际并没有挂载成功。
  - 父节点在添加多个ATTR类型子节点时，不允许KEY值相同的子节点同时存在于同一个父节点下。

  示例：

```
--为指定的DOC树添加DOMNODE节点，并通过DBE_XMLDOM.HASCHILDNODES()验证子节点是否添加成功。
DECLARE
   doc DBE_XMLDOM.DOMDocument;
   doc1 DBE_XMLDOM.DOMDocument;
   root DBE_XMLDOM.DOMElement;
```


版权所有 © 华为云计算技术有限公司

```
    rootnode DBE_XMLDOM.DOMNode;
    child1 DBE_XMLDOM.DOMElement;
    child2 DBE_XMLDOM.DOMElement;
    attr DBE_XMLDOM.DOMAttr;
    text DBE_XMLDOM.DOMTEXT;
    node DBE_XMLDOM.DOMNode;
    child1_node DBE_XMLDOM.DOMNode;
    attr_node DBE_XMLDOM.DOMNode;
    parent DBE_XMLDOM.DOMNode;
    buf varchar2(1000);
BEGIN
    doc := DBE_XMLDOM.newDOMDocument();
    root := DBE_XMLDOM.createElement(doc, 'root');
    rootnode := DBE_xmldom.makeNode(root);
    node := DBE_XMLDOM.appendChild(DBE_xmldom.makeNode(doc), rootnode);
    child1 := DBE_XMLDOM.createElement(doc, 'child1');
    child1_node := DBE_XMLDOM.makeNode(child1);
    node := DBE_XMLDOM.appendChild(rootnode, child1_node);
    attr := DBE_XMLDOM.createAttribute(doc, 'abc');
    attr_node := DBE_XMLDOM.makeNode(attr);
    node := DBE_XMLDOM.appendChild(child1_node, attr_node);
    IF DBE_XMLDOM.HASCHILDNODES(child1_node) THEN
        DBE_OUTPUT.print_line('HAS CHILD NODES');
    ELSE
        DBE_OUTPUT.print_line('NOT HAS CHILD NODES ');
    END IF;
    parent := DBE_XMLDOM.GETPARENTNODE(attr_node);
    buf := DBE_XMLDOM.GETNODENAME(parent);
    DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
NOT HAS CHILD NODES
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.CREATEELEMENT

  返回创建指定名称的DOMELEMENT对象。DBE_XMLDOM.CREATEELEMENT的函数原型为：

  ```
  DBE_XMLDOM.CREATEELEMENT(
    doc        IN      DOMDOCUMENT,
    tagName    IN      VARCHAR2)
   RETURN DOMELEMENT;
  ```

  返回创建指定名称和命名空间的DOMELEMENT对象。
  DBE_XMLDOM.CREATEELEMENT的函数原型为：

  ```
  DBE_XMLDOM.CREATEELEMENT(
    doc        IN     DOMDOCUMENT,
    tagName    IN     VARCHAR2,
    ns         IN     VARCHAR2)
   RETURN DOMELEMENT;
  ```

  表 2-428 DBE_XMLDOM.CREATEELEMENT 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT对象。
tagName	新建的DOMELEMENT名称。
ns	命名空间。

版权所有 © 华为云计算技术有限公司

说明

- tagName参数传入NULL和空字符串时，都会抛出异常 "NULL or invalid TagName argument specified"
- tagName和ns默认的最大长度为32767，超过该长度会抛出异常。

示例：

```
--1. 创建指定名称的DOMELEMENT对象。
DECLARE
  doc dbe_xmldom.domdocument;
  attr DBE_XMLDOM.DOMATTR;
  elem DBE_XMLDOM.DOMELEMENT;
  ans DBE_XMLDOM.DOMATTR;
  buf varchar2(1010);
BEGIN
   doc := dbe_xmldom.newdomdocument('<?xml version="1.0" encoding="UTF-8"?>
      <computer size="ITX"><cpu>Ryzen 9 3950X</cpu>
      <ram>32GBx2 DDR4 3200MHz</ram>
      <motherboard>ROG X570i</motherboard>
      <gpu>RTX2070 Super</gpu>
      <ssd>1TB NVMe Toshiba + 2TB NVMe WD Black</ssd>
      <hdd>12TB WD Digital</hdd>
      <psu>CORSAIR SF750</psu>
      <case>LIANLI TU150</case>
      </computer>');
   elem := dbe_xmldom.createelement(doc,'elem');
   DBE_XMLDOM.WRITETOBUFFER(dbe_xmldom.makenode(elem), buf);
   DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE

--2. 创建指定名称和命名空间的DOMELEMENT对象。
DECLARE
  doc dbe_xmldom.domdocument;
  attr DBE_XMLDOM.DOMATTR;
  elem DBE_XMLDOM.DOMELEMENT;
  ans DBE_XMLDOM.DOMNODE;
  buf varchar2(1010);
  list DBE_XMLDOM.DOMNODELIST;
  node DBE_XMLDOM.DOMNODE;
BEGIN
   doc := dbe_xmldom.newdomdocument('<h:data xmlns:h="http://www.w3.org/TR/html4/">
      <h:da1 len="10">test namespace</h:da1><h:da1>bbbbbbbbbb</h:da1></h:data>');
   elem := dbe_xmldom.createelement(doc,'elem','http://www.w3.org/TR/html5/');
   ans := DBE_XMLDOM.APPENDCHILD(dbe_xmldom.makenode(doc), dbe_xmldom.makenode(elem));
   DBE_XMLDOM.WRITETOBUFFER(doc, buf);
   DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
<?xml version="1.0" encoding="UTF-8"?>
<h:data xmlns:h="http://www.w3.org/TR/html4/">
 <h:da1 len="10">test namespace</h:da1>
 <h:da1>bbbbbbbbbb</h:da1>
</h:data>
<elem xmlns="http://www.w3.org/TR/html5/"/>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.CREATETEXTNODE

  创建并返回DOMTEXT对象。DBE_XMLDOM.CREATETEXTNOD的函数原型为：

```
DBE_XMLDOM.CREATETEXTNODE(
  doc IN DOMDocument,
  data IN VARCHAR2)
RETURN DOMTEXT;
```


版权所有 © 华为云计算技术有限公司

表 2-429 DBE_XMLDOM.CREATETEXTNODE 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT。
data	DOMText节点的内容。

说明

- data可以输入空字符串和NULL值。
- data默认的最大长度为32767，超过该长度会抛出异常。

示例：

```
--为DOC树添加DOMTEXT节点，并将DOC树打印输出到缓冲区。
DECLARE
  doc DBE_XMLDOM.DOMDOCUMENT;
  doctext DBE_XMLDOM.DOMTEXT;
  node DBE_XMLDOM.DOMNODE;
  buffer varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
     <!ELEMENT to (#PCDATA)>
     <!ELEMENT from (#PCDATA)>
     <!ELEMENT heading (#PCDATA)>
     <!ELEMENT body (#PCDATA)>]>
     <note>
       <to>中文</to>
       <from>Jani</from>
       <heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body>
     </note>');
  doctext := DBE_XMLDOM.CREATETEXTNODE(doc, 'there is nothing');
  node := DBE_XMLDOM.MAKENODE(doctext);
  dbe_xmldom.writetobuffer(node, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
buffer:
there is nothing
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.FREEDOCUMENT

  释放DOMDOCUMENT节点。DBE_XMLDOM.FREEDOCUMENT的函数原型为：

```
DBE_XMLDOM.FREEDOCUMENT(
  doc     IN     DOMDOCUMENT);
```

表 2-430 DBE_XMLDOM.FREEDOCUMENT 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。

示例：

```
--在DOC树中添加DOMNODE节点后，将整个DOC树的资源释放。
DECLARE
```


版权所有 © 华为云计算技术有限公司

```
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  doc_node dbe_xmldom.DOMNODE;
  root_elmt dbe_xmldom.DOMELEMENT;
  root_node dbe_xmldom.DOMNODE;
  value  varchar(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument();
  doc_node := dbe_xmldom.MAKENODE(doc);
  root_elmt := dbe_xmldom.CREATEELEMENT(doc,'staff');
  root_node:=dbe_xmldom.APPENDCHILD(doc_node, dbe_xmldom.MAKENODE(root_elmt));
  dbe_xmldom.freedocument(doc);
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.FREEELEMENT

  释放DOMELEMENT节点。DBE_XMLDOM.FREEELEMENT的函数原型为：

```
DBE_XMLDOM.FREEELEMENT(
  elem     IN     DOMELEMENT);
```

表 2-431 DBE_XMLDOM.FREEELEMENT 接口参数说明

参数	描述
elem	指定的DOMELEMENT节点。

示例：

```
--从DOC中获取DOMELEMENT节点后对其进行释放，对比其free前后是否为空的情况。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  node dbe_xmldom.domnode;
  node1 dbe_xmldom.domnode;
  nodelist DBE_XMLDOM.DOMNODELIST;
  len INTEGER;
  buffer varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
     <!ELEMENT to (#PCDATA)>
     <!ELEMENT from (#PCDATA)>
     <!ELEMENT heading (#PCDATA)>
     <!ELEMENT body (#PCDATA)>]>
     <note>
       <to>中文</to>
       <from>Jani</from>
       <heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body>
     </note>');
  elem := dbe_xmldom.GETDOCUMENTELEMENT(doc);
  IF DBE_XMLDOM.ISNULL(elem) THEN
      dbe_output.print_line('IS NULL');
  ELSE
      dbe_output.print_line('NOT NULL');
  END IF;
  dbe_xmldom.FREEELEMENT(elem);

  IF DBE_XMLDOM.ISNULL(elem) THEN
       dbe_output.print_line('IS NULL');
  ELSE
       dbe_output.print_line('NOT NULL');
  END IF;
END;
```


版权所有 © 华为云计算技术有限公司

```
/
-- 预期结果为：
NOT NULL
IS NULL
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.FREENODE

  释放DOMNODE节点。DBE_XMLDOM.FREENODE的函数原型为：

```
DBE_XMLDOM.FREENODE(
  n IN DOMNODE);
```

表 2-432 DBE_XMLDOM.FREENODE 接口参数说明

参数	描述
n	指定的DOMNODE节点。

说明

- GaussDB数据库进行FREENODE操作后，被释放的节点不会出现重新可用的情况；ORA数据库在FREENODE后存在被释放的节点重新可用并变成其他节点的情况。
- 其他接口在调用被释放的DOMNODE节点时与ORA数据库存在差异。

示例：

```
--从DOC树中获取一个DOMNODE节点后对其进行释放，对比其free前后是否为空的情况。
DECLARE
  doc dbe_xmldom.domdocument;
  node dbe_xmldom.domnode;
  node1 dbe_xmldom.domnode;
  nodelist DBE_XMLDOM.DOMNODELIST;
  len INTEGER;
  buffer1 varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
     <!ELEMENT to (#PCDATA)>
     <!ELEMENT from (#PCDATA)>
     <!ELEMENT heading (#PCDATA)>
     <!ELEMENT body (#PCDATA)>]>
     <note>
       <to>中文</to>
       <from>Jani</from>
       <heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body>
     </note>');
  node := dbe_xmldom.makenode(doc);
  node := dbe_xmldom.GETFIRSTCHILD(node);
  IF DBE_XMLDOM.ISNULL(node) THEN
        dbe_output.print_line('IS NULL');
  ELSE
        dbe_output.print_line('NOT NULL');
  END IF;
  DBE_XMLDOM.FREENODE(node);
  IF DBE_XMLDOM.ISNULL(node) THEN
        dbe_output.print_line('IS NULL');
  ELSE
        dbe_output.print_line('NOT NULL');
  END IF;
END;
/
-- 预期结果为：
NOT NULL
IS NULL
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

- DBE_XMLDOM.FREENODELIST

  释放DOMNODELIST节点。DBE_XMLDOM.FREENODE的函数原型为：

```
DBE_XMLDOM.GETLENGTH(
  nl     IN    DOMNODELIST);
```

表 2-433 DBE_XMLDOM.FREENODELIST 接口参数说明

参数	描述
nl	指定的DOMNODELIST节点。

说明

- FREENODELIST会彻底释放NODELIST。
- 其他接口在调用被释放的DOMNODELIST节点时与ORA数据库存在差异。
- freenodelist不允许空值入参。

示例：

```
--从DOC树中获取一个DOMNODELIST节点后对其进行释放，对比其free前后的长度。
DECLARE
  doc dbe_xmldom.domdocument;
  node dbe_xmldom.domnode;
  node1 dbe_xmldom.domnode;
  nodelist DBE_XMLDOM.DOMNODELIST;
  len INTEGER;
  buffer1 varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
     <!ELEMENT to (#PCDATA)>
     <!ELEMENT from (#PCDATA)>
     <!ELEMENT heading (#PCDATA)>
     <!ELEMENT body (#PCDATA)>]>
     <note>
       <to>中文</to>
       <from>Jani</from>
       <heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body>
     </note>');
  node := dbe_xmldom.makenode(doc);
  node := dbe_xmldom.GETFIRSTCHILD(node);
  nodelist := DBE_XMLDOM.GETCHILDNODES(node);
  len := DBE_XMLDOM.GETLENGTH(nodelist);
  RAISE NOTICE 'len :  %', len;
  DBE_XMLDOM.FREENODELIST(nodelist);
  len := DBE_XMLDOM.GETLENGTH(nodelist);
  RAISE NOTICE 'len :  %', len;
END;
/
-- 预期结果为：
NOTICE:  len :  4
NOTICE:  len :  0
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETATTRIBUTE

  按名称返回DOMELEMENT属性的值。DBE_XMLDOM.GETATTRIBUTE的函数原型为：

```
DBE_XMLDOM.GETATTRIBUTE(
  elem       IN      DOMELEMENT,
  name       IN      VARCHAR2)
RETURN VARCHAR2;
```


版权所有 © 华为云计算技术有限公司

按名称和命名空间URI返回DOMELEMENT属性的值。
DBE_XMLDOM.GETATTRIBUTE的函数原型为：

```
DBE_XMLDOM.GETATTRIBUTE(
  elem      IN     DOMELEMENT,
  name      IN     VARCHAR2,
  ns        IN     VARCHAR2)
RETURN VARCHAR2;
```

表 2-434 DBE_XMLDOM.GETATTRIBUTE 接口参数说明

参数	描述
elem	指定的DOMELEMENT节点。
name	属性名称。
ns	命名空间。

说明

- DBE_XMLDOM.GETATTRIBUTE接口的参数ns不支持传入参数" * "。
- GaussDB数据库不支持将命名空间前缀作为属性，不允许通过DBE_XMLDOM.GETATTRIBUTE接口查询该前缀的值。

示例：

```
--1. 按名称返回DOMELEMENT属性的值。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  docnode DBE_XMLDOM.DOMNode;
  buffer varchar2(1010);
  value  varchar2(1000);
BEGIN
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(doc, 'root');
  DBE_XMLDOM.setattribute(elem, 'len', '50cm');
  docnode := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(doc),
DBE_XMLDOM.makeNode(elem));
  value := DBE_XMLDOM.getattribute(elem, 'len');
  dbe_output.print_line('value: ');
  dbe_output.print_line(value);
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
value:
50cm
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<root len="50cm"/>

ANONYMOUS BLOCK EXECUTE

--2. 按名称和命名空间URI返回DOMELEMENT属性的值。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  docnode DBE_XMLDOM.DOMNode;
  buffer varchar2(1010);
  value  varchar(1000);
```


版权所有 © 华为云计算技术有限公司

```
BEGIN
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(doc, 'root');
  DBE_XMLDOM.setattribute(elem, 'len', '50cm', 'www.xxxxx.com');
  docnode := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(doc),
DBE_XMLDOM.makeNode(elem));
  value := DBE_XMLDOM.getattribute(elem, 'len', 'www.xxxxx.com');
  dbe_output.print_line('value: ');
  dbe_output.print_line(value);
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
value:
50cm
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<root len="50cm"/>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETATTRIBUTES

  将DOMNode节点属性值作为map返回。DBE_XMLDOM.GETATTRIBUTES的函数原型为：

```
DBE_XMLDOM.GETATTRIBUTES(
  n IN DOMNode)
RETURN DOMNAMEDNODEMAP;
```

表 2-435 DBE_XMLDOM.GETATTRIBUTES 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--获取DOMNODE节点下的属性值并返回DOMNAMEDNODEMAP类型，输出DOMNAMEDNODEMAP的
长度和第一个节点值。
DECLARE
  doc dbe_xmldom.domdocument;
  node dbe_xmldom.domnode;
  node1 dbe_xmldom.domnode;
  len INTEGER;
  map DBE_XMLDOM.DOMNAMEDNODEMAP;
  buffer1 varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <note a="16" b="176" c="asd">
       <to>中文</to>
       <from>Jani</from>
       <heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body>
     </note>');
  node := dbe_xmldom.makenode(doc);
  node := dbe_xmldom.GETFIRSTCHILD(node);
  map := DBE_XMLDOM.GETATTRIBUTES(node);
  IF DBE_XMLDOM.ISNULL(map) THEN
        dbe_output.print_line('IS NULL');
  ELSE
        dbe_output.print_line('NOT NULL');
  END IF;
    len := DBE_XMLDOM.GETLENGTH(map);
  RAISE NOTICE 'len :  %', len;
   node1 := DBE_XMLDOM.ITEM(map, 0);
```


版权所有 © 华为云计算技术有限公司

```
  dbe_xmldom.writetobuffer(node1, buffer1);
  dbe_output.print_line('buffer1: ');
  dbe_output.print_line(buffer1);
END;
/
-- 预期结果为：
NOT NULL
NOTICE:  len :  3
buffer1:
16
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETCHILDNODES

  函数将节点下的若干子节点转换成节点列表。DBE_XMLDOM.GETCHILDNODES的函数原型为：

```
DBE_XMLDOM.GETCHILDNODES(
  n IN DOMNode)
RETURN DOMNodeList;
```

表 2-436 DBE_XMLDOM.GETCHILDNODES 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--获取DOC树的第一个子节点后，将节点下的若干子节点转换成节点列表，输出其长度信息。
DECLARE
    doc dbe_xmldom.domdocument;
    doc_node dbe_xmldom.domnode;
    root_node dbe_xmldom.domnode;
    node_list dbe_xmldom.domnodelist;
    list_len integer;
    node_name varchar2(1000);
    node_type integer;
    buffer varchar2(1010);
BEGIN
   doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
       <note>
          <to>中文</to>
          <from>Jani</from>
          <heading>Reminder</heading>
          <body>Don''t forget me this weekend!</body>
       </note>');
    doc_node := DBE_XMLDOM.MAKENODE(doc);
    root_node := DBE_XMLDOM.GETFIRSTCHILD(doc_node);
    node_name := DBE_XMLDOM.GETNODENAME(root_node);
    node_type := DBE_XMLDOM.GETNODETYPE(root_node);
    dbe_output.print_line(node_name);
    dbe_output.print_line(node_type);
    node_list := DBE_XMLDOM.GETCHILDNODES(root_node);
    list_len := DBE_XMLDOM.GETLENGTH(node_list);
    dbe_output.print_line(list_len);
END;
/
-- 预期结果为：
note
1
4
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETCHILDRENBYTAGNAME

  按名称返回DOMELEMENT的子节点。
  DBE_XMLDOM.GETCHILDRENBYTAGNAME的函数原型为：

 版权所有 © 华为云计算技术有限公司

```
DBE_XMLDOM.GETCHILDRENBYTAGNAME (
  elem       IN      DOMELEMENT,
  name       IN      VARCHAR2)
RETURN DOMNODELIST;
```

按名称和命名空间返回DOMELEMENT的子节点。
DBE_XMLDOM.GETCHILDRENBYTAGNAME的函数原型为：

```
DBE_XMLDOM.GETCHILDRENBYTAGNAME (
  elem      IN     DOMELEMENT,
  name      IN     VARCHAR2,
  ns        IN     VARCHAR2)
 RETURN DOMNODELIST;
```

表 2-437 DBE_XMLDOM.GETCHILDRENBYTAGNAME 接口参数说明

参数	描述
elem	指定的DOMELEMENT节点。
name	属性名称。
ns	命名空间。

说明

DBE_XMLDOM.GETCHILDRENBYTAGNAME接口的参数ns不支持传入参数" * "，如需获取节点下全部属性，可使用DBE_XMLDOM.GETCHILDNODES接口。

示例：

```
--1. 按名称返回DOMELEMENT的子节点。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  docnodelist dbe_xmldom.domnodelist;
  node_elem dbe_xmldom.domelement;
  node dbe_xmldom.domnode;
  buffer varchar2(1010);
  value  varchar2(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0" encoding="UTF-8"?>
     <students age="16" hight="176">
     <student>
       <name>Jerry</name><age>519</age><gender>man</gender><abc>12345</abc>
     </student>
     <student>
       <name>Bob</name><age>245</age><gender>woman</gender><abc>54321</abc>
     </student>
     </students>');
  elem := dbe_xmldom.GETDOCUMENTELEMENT(doc);
  docnodelist := dbe_xmldom.GETCHILDRENBYTAGNAME(elem, 'student');
  node := dbe_xmldom.ITEM(docnodelist, 0);
  node_elem := dbe_xmldom.makeelement(node);
  value := DBE_XMLDOM.gettagname(node_elem);
  dbe_output.print_line('value: ');
  dbe_output.print_line(value);
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
value:
student
buffer:
```


版权所有 © 华为云计算技术有限公司

```
<?xml version="1.0" encoding="UTF-8"?>
<students age="16" hight="176">
  <student>
    <name>Jerry</name>
    <age>519</age>
    <gender>man</gender>
    <abc>12345</abc>
  </student>
  <student>
    <name>Bob</name>
    <age>245</age>
    <gender>woman</gender>
    <abc>54321</abc>
  </student>
</students>

ANONYMOUS BLOCK EXECUTE

--2. 按名称和命名空间返回DOMELEMENT的子节点。
DECLARE
   doc dbe_xmldom.domdocument;
   elem dbe_xmldom.domelement;
   node dbe_xmldom.domnode;
   node_elem dbe_xmldom.domelement;
   docnodelist dbe_xmldom.domnodelist;
   buffer varchar2(1010);
   value varchar2(1000);
BEGIN
   doc := dbe_xmldom.newdomdocument('
      <note xmlns:h="www.xxxxx.com">
      <h:to h:len="50cm">中文</h:to>
      <from>Jani</from>
      <heading>Reminder</heading>
      <body>Don''t forget me this weekend!</body>
      </note>');
   elem := dbe_xmldom.GETDOCUMENTELEMENT(doc);
   docnodelist := dbe_xmldom.GETCHILDRENBYTAGNAME(elem, 'to', 'www.xxxxx.com');
   node := dbe_xmldom.ITEM(docnodelist, 0);
   node_elem := dbe_xmldom.makeelement(node);
   value := DBE_XMLDOM.getattribute(node_elem, 'len');
   dbe_output.print_line('value: ');
   dbe_output.print_line(value);
END;
/
-- 预期结果为：
value:
50cm
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETDOCUMENTELEMENT

  返回指定DOCUMENT的首个子节点。DBE_XMLDOM.GETDOCUMENTELEMENT的函数原型为：

```
DBE_XMLDOM.GETDOCUMENTELEMENT(
   doc      IN      DOMDOCUMENT)
 RETURN DOMELEMENT;
```

表 2-438 DBE_XMLDOM.GETDOCUMENTELEMENT 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。

示例：

```
--获取DOC树中的首个子节点，并输出该节点的名称。
DECLARE
```


版权所有 © 华为云计算技术有限公司

```
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  doc_node dbe_xmldom.DOMNODE;
  root_elmt dbe_xmldom.DOMELEMENT;
  root_node dbe_xmldom.DOMNODE;
  value  varchar(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument();
  doc_node := dbe_xmldom.MAKENODE(doc);
  root_elmt := dbe_xmldom.CREATEELEMENT(doc,'staff');
  root_node:=dbe_xmldom.APPENDCHILD(doc_node, dbe_xmldom.MAKENODE(root_elmt));
  elem := dbe_xmldom.GETDOCUMENTELEMENT(doc);
  value := DBE_XMLDOM.gettagname(elem);
  dbe_output.print_line(value);
END;
/
-- 预期结果为：
staff
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETFIRSTCHILD

  返回节点的第一个子节点。DBE_XMLDOM.GETFIRSTCHILD的函数原型为：

```
DBE_XMLDOM.GETFIRSTCHILD(
  n IN DOMNODE)
RETURN DOMNODE;
```

表 2-439 DBE_XMLDOM.GETFIRSTCHILD 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--获取DOC转换成DOMNODE类型后的第一个子节点后输出其名称和类型；在获取到的第一个子节点基础上，获取该DOMNODE的第一个子节点并输出其名称。
DECLARE
   doc dbe_xmldom.domdocument;
   doc_node dbe_xmldom.domnode;
    root_node dbe_xmldom.domnode;
   inside_node dbe_xmldom.domnode;
   node_name varchar2(1000);
   node_type integer;
BEGIN
   doc := dbe_xmldom.newdomdocument('<?xml version="1.0" encoding="UTF-8"?>
      <students age="16" hight="176">
      <student1>
         <name>Jerry</name><age>519</age><gender>man</gender><abc>12345</abc>
      </student1>
      <student2>
         <name>Bob</name><age>245</age><gender>woman</gender><abc>54321</abc>
      </student2>
      </students>');
   doc_node := DBE_XMLDOM.MAKENODE(doc);
   root_node := DBE_XMLDOM.GETFIRSTCHILD(doc_node);
   node_name := DBE_XMLDOM.GETNODENAME(root_node);
   node_type := DBE_XMLDOM.GETNODETYPE(root_node);
   dbe_output.print_line(node_name);
   dbe_output.print_line(node_type);
   inside_node := DBE_XMLDOM.GETFIRSTCHILD(root_node);
   node_name := DBE_XMLDOM.GETNODENAME(inside_node);
   dbe_output.print_line(node_name);
END;
/
-- 预期结果为：
students
```


版权所有 © 华为云计算技术有限公司

```
1
student1
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETLASTCHILD

  返回节点的最后一个子节点。DBE_XMLDOM.GETLASTCHILD的函数原型为：

```
DBE_XMLDOM.GETLASTCHILD(
  n IN DOMNODE)
RETURN DOMNODE;
```

表 2-440 DBE_XMLDOM.GETLASTCHILD 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--获取DOC转换成DOMNODE类型后的最后一个子节点后输出其名称和类型；在获取到的最后一个子节点基础上，获取该DOMNODE的最后一个子节点并输出其名称。
DECLARE
    doc dbe_xmldom.domdocument;
    doc_node dbe_xmldom.domnode;
    root_node dbe_xmldom.domnode;
    inside_node dbe_xmldom.domnode;
    node_name varchar2(1000);
    node_type integer;
BEGIN
    doc := dbe_xmldom.newdomdocument('<?xml version="1.0" encoding="UTF-8"?>
      <students age="16" hight="176">
      <student1>
        <name>Jerry</name><age>519</age><gender>man</gender><abc>12345</abc>
      </student1>
      <student2>
        <name>Bob</name><age>245</age><gender>woman</gender><abc>54321</abc>
      </student2>
      </students>');
    doc_node := DBE_XMLDOM.MAKENODE(doc);
    root_node := DBE_XMLDOM.GETFIRSTCHILD(doc_node);
    node_name := DBE_XMLDOM.GETNODENAME(root_node);
    node_type := DBE_XMLDOM.GETNODETYPE(root_node);
    dbe_output.print_line(node_name);
    dbe_output.print_line(node_type);
    inside_node := DBE_XMLDOM.GETLASTCHILD(root_node);
    node_name := DBE_XMLDOM.GETNODENAME(inside_node);
    dbe_output.print_line(node_name);
END;
/
-- 预期结果为：
students
1
student2
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETLENGTH

  返回DOMNAMEDNODEMAP类型节点中的节点数。DBE_XMLDOM.GETLENGTH的函数原型为：

```
DBE_XMLDOM.GETLENGTH(
  nnm      IN     DOMNAMEDNODEMAP)
 RETURN NUMBER;
```

  返回DOMNODELIST类型节点中的节点数。DBE_XMLDOM.GETLENGTH的函数原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_XMLDOM.GETLENGTH(
  nl     IN    DOMNODELIST)
 RETURN NUMBER;
```

表 2-441 DBE_XMLDOM.GETLENGTH 接口参数说明

参数	描述
nnm	指定的DOMNAMEDNODEMAP类型节点。
nl	指定的DOMNODELIST类型节点。

示例：

```
--1. DOMNAMEDNODEMAP类型作为函数参数。
DECLARE
   doc DBE_XMLDOM.DOMDocument;
   elem DBE_XMLDOM.DOMElement;
   map DBE_XMLDOM.DOMNAMEDNODEMAP;
   node DBE_XMLDOM.DOMNODE;
   buf varchar2(10000);
   len INTEGER;
BEGIN
   doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
       <bookstore category="web" cover="paperback">
       <book category="cooking">
          <title lang="en">Everyday Italian</title>
          <author>Giada De Laurentiis</author>
          <year>2005</year>
          <price>30.00</price>
       </book>
       </bookstore>');
   elem := DBE_XMLDOM.GETDOCUMENTELEMENT(doc);
   node := DBE_XMLDOM.MAKENODE(elem);
   map := DBE_XMLDOM.GETATTRIBUTES(node);
   len := DBE_XMLDOM.GETLENGTH(map);
   DBE_OUTPUT.print_line(len);
END;
/
-- 预期结果为：
2
ANONYMOUS BLOCK EXECUTE

--2. Nodelist类型作为函数参数
DECLARE
   doc dbe_xmldom.domdocument;
   node dbe_xmldom.domnode;
   node1 dbe_xmldom.domnode;
   nodelist DBE_XMLDOM.DOMNODELIST;
   len INTEGER;
   buffer1 varchar2(1010);
BEGIN
   doc := dbe_xmldom.newdomdocument('<?xml version="1.0" encoding="UTF-8"?>
       <students age="16" hight="176">
       <student>
          <name>Jerry</name><age>519</age><gender>man</gender><abc>12345</abc>
       </student>
       <student>
          <name>Jerry</name><age>519</age><gender>man</gender><abc>12345</abc>
       </student>
       </students>');
   node := dbe_xmldom.makenode(doc);
   node := dbe_xmldom.GETFIRSTCHILD(node);
   nodelist := DBE_XMLDOM.GETCHILDNODES(node);
   len := DBE_XMLDOM.GETLENGTH(nodelist);
```


版权所有 © 华为云计算技术有限公司

```
  RAISE NOTICE 'len :  %', len;
END;
/
-- 预期结果为：
NOTICE:  len :  2
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETLOCALNAME

  函数返回给定的DOMATTR类型节点的本地名称。DBE_XMLDOM.MAKENODE的函数原型为：

```
DBE_XMLDOM.GETLOCALNAME(
  a       IN     DOMATTR)
RETURN VARCHAR2;
```

  函数返回给定的DOMELEMENT类型节点的本地名称。
  DBE_XMLDOM.MAKENODE的函数原型为

```
DBE_XMLDOM.GETLOCALNAME(
  elem       IN     DOMELEMENT)
RETURN VARCHAR2;
```

  存储过程返回给定的DOMNODE类型节点的本地名称。
  DBE_XMLDOM.MAKENODE的函数原型为

```
DBE_XMLDOM.GETLOCALNAME(
  n      IN     DOMNODE,
  data   OUT    VARCHAR2);
```

表 2-442 DBE_XMLDOM.GETLOCALNAME 接口参数说明

参数	描述
a	指定的DOMATTR类型节点。
elem	指定的DOMELEMENT类型节点。
n	指定的DOMNODE类型节点。
data	返回的本地名称。

示例：

```
--1. createAttribute函数生成attr节点，获取它的本地名称。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root DBE_XMLDOM.DOMElement;
  attr1 DBE_XMLDOM.DOMATTR;
  value VARCHAR2(1000);
BEGIN
  doc := DBE_xmldom.newdomdocument('<?xml version="1.0"?>
    <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>]>
    <note><to>中文</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don''t forget me this weekend!</body>
    </note>');
  attr1 := DBE_XMLDOM.createAttribute(doc,'len');
  value := DBE_XMLDOM.getlocalname(attr1);
  DBE_output.print_line('value: ');
  DBE_output.print_line(value);
END;
/
-- 预期结果为：
```


版权所有 © 华为云计算技术有限公司

```
value:
len
ANONYMOUS BLOCK EXECUTE

--2. createElement函数生成elem节点，获取它的本地名称。
DECLARE
  doc DBE_xmldom.domdocument;
  elem DBE_xmldom.domelement;
  value  varchar2(10000);
BEGIN
  doc := DBE_xmldom.newdomdocument();
  elem := DBE_XMLDOM.createELEMENT(doc, 'root');
  value := DBE_XMLDOM.getlocalname(elem);
  DBE_output.print_line('value: ');
  DBE_output.print_line(value);
END;
/
-- 预期结果为：
value:
root
ANONYMOUS BLOCK EXECUTE

--3. Element节点转换成node节点后，取其本地名称。
DECLARE
  doc DBE_xmldom.domdocument;
  elem DBE_xmldom.domelement;
  node DBE_xmldom.domnode;
  value  varchar2(100);
  buf varchar2(100);
BEGIN
  doc := DBE_xmldom.newdomdocument();
  elem := DBE_XMLDOM.createELEMENT(doc, 'root');
  node := DBE_xmldom.makenode(elem);
  DBE_XMLDOM.getlocalname(node, buf);
  DBE_output.print_line('buf: ');
  DBE_output.print_line(buf);
END;
/
-- 预期结果为：
buf:
root
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETNAMEDITEM

  检索由名称指定的节点。DBE_XMLDOM.GETNAMEDITEM的函数原型为：

```
DBE_XMLDOM.GETNAMEDITEM(
  nnm IN DOMNAMEDNODEMAP,
  name IN VARCHAR2)
RETURN DOMNODE;
```

  检索由名称和命名空间指定的节点。DBE_XMLDOM.GETNAMEDITEM的函数原型为：

```
DBE_XMLDOM.GETNAMEDITEM(
  nnm IN DOMNAMEDNODEMAP,
  name IN VARCHAR2,
  ns IN VARCHAR2)
RETURN DOMNODE;
```

表 2-443 DBE_XMLDOM.GETNAMEDITEM 接口参数说明

参数	描述
nnm	DOMNAMEDNODEMAP。
name	要检索的元素名称。
ns	命名空间。

版权所有 © 华为云计算技术有限公司

说明

- name和nnm可以输入NULL值，但不可不入参。
- name和ns默认的最大长度为32767，超出该长度会报错。
- name和ns可输入int类型，长度可超出127位。

示例：

```
--1. 检索由名称指定的节点。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  elem DBE_XMLDOM.DOMElement;
  map DBE_XMLDOM.DOMNAMEDNODEMAP;
  node DBE_XMLDOM.DOMNODE;
  node2 DBE_XMLDOM.DOMNODE;
  buf varchar2(1000);
  buf2 varchar2(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument('<bookstore category="web" cover="paperback">
     <book category="cooking"><title lang="en">Everyday Italian</title>
     <author>Giada De Laurentiis</author><year>2005</year>
     <price>30.00</price></book></bookstore>');
  elem := DBE_XMLDOM.GETDOCUMENTELEMENT(doc);
  node := DBE_XMLDOM.MAKENODE(elem);
  map := DBE_XMLDOM.GETATTRIBUTES(node);
  node2:= DBE_XMLDOM.GETNAMEDITEM(map,'category');
  DBE_XMLDOM.writeToBuffer(node2, buf2);
  dbe_output.print_line(buf2);
END;
/
-- 预期结果为：
web
ANONYMOUS BLOCK EXECUTE

--2. 检索由名称和命名空间指定的节点。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root DBE_XMLDOM.DOMElement;
  elem DBE_XMLDOM.DOMElement;
  map DBE_XMLDOM.DOMNAMEDNODEMAP;
  node DBE_XMLDOM.DOMNODE;
  buf varchar2(1000);
  buf2 varchar2(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument('<h:table xmlns:h="http://www.w3.org/TR/html4/">
      <h:tr h:id="10"><h:td >Apples</h:td>
      <h:td>Bananas</h:td></h:tr></h:table>');
  root := DBE_XMLDOM.getDocumentElement(doc);
  node := DBE_XMLDOM.MAKENODE(root);
  node := dbe_xmldom.GETFIRSTCHILD(node);
  map := DBE_XMLDOM.GETATTRIBUTES(node);
  node := DBE_XMLDOM.GETNAMEDITEM(map,'id','http://www.w3.org/TR/html4/');
  DBE_XMLDOM.writeToBuffer(node, buf2);
  dbe_output.print_line(buf2);
END;
/
-- 预期结果为：
10
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETNEXTSIBLING

  返回下一个节点。DBE_XMLDOM.GETNEXTSIBLING的函数原型为：

```
DBE_XMLDOM.GETNEXTSIBLING(
  n  IN  DOMNODE)
RETURN DOMNODE;
```


版权所有 © 华为云计算技术有限公司

表 2-444 DBE_XMLDOM.GETNEXTSIBLING 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--首先获取DOC转换成DOMNODE类型后的第一个子节点；在获取到的第一个子节点基础上，获取该DOMNODE的第一个子节点；通过DBE_XMLDOM.GETNEXTSIBLING获取该节点的下一个节点，并输出下一个节点的名称。
DECLARE
    doc dbe_xmldom.domdocument;
    doc_node dbe_xmldom.domnode;
    root_node dbe_xmldom.domnode;
    inside_node dbe_xmldom.domnode;
    node_name varchar2(1000);
    node_type integer;
BEGIN
   doc := dbe_xmldom.newdomdocument('<computer size="ITX">
        <cpu>Ryzen 9 3950X</cpu>
        <ram>32GBx2 DDR4 3200MHz</ram>
        <motherboard>X570i</motherboard>
    </computer>');
    doc_node := DBE_XMLDOM.MAKENODE(doc);
    root_node := DBE_XMLDOM.GETFIRSTCHILD(doc_node);
    node_name := DBE_XMLDOM.GETNODENAME(root_node);
    node_type := DBE_XMLDOM.GETNODETYPE(root_node);
    dbe_output.print_line(node_name);
    dbe_output.print_line(node_type);
    inside_node := DBE_XMLDOM.GETFIRSTCHILD(root_node);
    node_name := DBE_XMLDOM.GETNODENAME(inside_node);
    dbe_output.print_line(node_name);
    inside_node := DBE_XMLDOM.GETNEXTSIBLING(inside_node);
    node_name := DBE_XMLDOM.GETNODENAME(inside_node);
    dbe_output.print_line(node_name);
END;
/
-- 预期结果为：
computer
1
cpu
ram
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETNODENAME

  返回NODE节点的名称。DBE_XMLDOM.GETNODENAME的函数原型为：

```
DBE_XMLDOM.GETNODENAME(
   n  IN  DOMNODE)
RETURN  VARCHAR2;
```

表 2-445 DBE_XMLDOM.GETNODENAME 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--在DOC树中获取DOMNODE节点，输出该节点的名称。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root DBE_XMLDOM.DOMElement;
```


版权所有 © 华为云计算技术有限公司

```
  root_node DBE_XMLDOM.DOMNode;
  inside_node DBE_XMLDOM.DOMNode;
  buf VARCHAR2(1000);
BEGIN
   doc := dbe_xmldom.newdomdocument('<bookstore category="web" cover="paperback">
       <book category="cooking"><title lang="en">Everyday Italian</title>
       <author>Giada De Laurentiis</author><year>2005</year>
       <price>30.00</price></book></bookstore>');
   root := DBE_XMLDOM.getDocumentElement(doc);
   root_node := DBE_XMLDOM.MAKENODE(root);
   inside_node := DBE_XMLDOM.GETFIRSTCHILD(root_node);
   buf := DBE_XMLDOM.GETNODENAME(inside_node);
   dbe_output.print_line(buf);
END;
/
-- 预期结果为：
book
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETNODETYPE

  返回NODE节点的类型。DBE_XMLDOM.GETNODETYPE的函数原型为：

```
DBE_XMLDOM.GETNODETYPE(
   n  IN  DOMNODE)
RETURN  NUMBER;
```

表 2-446 DBE_XMLDOM.GETNODETYPE 接口参数说明

参数	描述
n	指定的DOMNODE节点。

示例：

```
--在DOC树中获取DOMNODE节点，输出该节点的类型值。
DECLARE
   doc DBE_XMLDOM.DOMDocument;
   doc_node DBE_XMLDOM.DOMNode;
   num number;
   buf varchar2(1000);
BEGIN
   doc := dbe_xmldom.newdomdocument('<bookstore category="web" cover="paperback">
       <book category="cooking"><title lang="en">Everyday Italian</title>
       <author>Giada De Laurentiis</author><year>2005</year>
       <price>30.00</price></book></bookstore>');
   doc_node := DBE_XMLDOM.makeNode(doc);
   num := DBE_XMLDOM.GETNODETYPE(doc_node);
   dbe_output.print_line(num);
   buf := DBE_XMLDOM.GETNODENAME(doc_node);
   dbe_output.print_line(buf);
END;
/
-- 预期结果为：
9
#document
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETNODEVALUE

  返回NODE节点的值。DBE_XMLDOM.GETNODEVALUE的函数原型为：

```
DBE_XMLDOM.GETNODEVALUE(
   n  IN  DOMNODE)
RETURN  VARCHAR2;
```


版权所有 © 华为云计算技术有限公司

表 2-447 DBE_XMLDOM.GETNODEVALUE 接口参数说明

参数	描述
n	指定的DOMNODE对象。

示例：

```
--将DOMTEXT类型节点转换为DOMNODE类型后获取该节点的值并输出。
DECLARE
  buf VARCHAR2(1000);
  doc DBE_XMLDOM.DOMDocument;
  text DBE_XMLDOM.DOMText;
  elem2 DBE_XMLDOM.DOMElement;
  node DBE_XMLDOM.DOMNode;
begin
  doc := DBE_XMLDOM.NEWDOMDOCUMENT();
  text := DBE_XMLDOM.createTextNode(doc, 'aaa');
  DBE_XMLDOM.SETNODEVALUE(DBE_XMLDOM.makeNode(text), 'ccc');
  buf := DBE_XMLDOM.GETNODEVALUE(DBE_XMLDOM.makeNode(text));
  DBE_OUTPUT.print_line(buf);
end;
/
-- 预期结果为：
ccc
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETPARENTNODE

  返回给定NODE节点的父节点。DBE_XMLDOM.GETPARENTNODE的函数原型为：

```
DBE_XMLDOM.GETPARENTNODE(
   n  IN  DOMNODE)
RETURN  DOMNODE;
```

表 2-448 DBE_XMLDOM.GETPARENTNODE 接口参数说明

参数	描述
n	指定的DOMNODE对象。

示例：

```
--向DOC树中添加子节点后，获取该子节点的父节点，输出父节点的名称。
DECLARE
   doc DBE_XMLDOM.DOMDocument;
   doc1 DBE_XMLDOM.DOMDocument;
   root DBE_XMLDOM.DOMElement;
   child1 DBE_XMLDOM.DOMElement;
   child2 DBE_XMLDOM.DOMElement;
   attr DBE_XMLDOM.DOMAttr;
   text DBE_XMLDOM.DOMTEXT;
   node DBE_XMLDOM.DOMNode;
   parent DBE_XMLDOM.DOMNode;
   buf varchar2(1000);
BEGIN
   doc := DBE_XMLDOM.newDOMDocument();
   root := DBE_XMLDOM.createElement(doc, 'root');
   node := DBE_XMLDOM.appendChild(DBE_xmldom.makeNode(doc),DBE_xmldom.makeNode(root));
   child1 := DBE_XMLDOM.createElement(doc, 'child1');
   node := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(root),
DBE_XMLDOM.makeNode(child1));
   child2 := DBE_XMLDOM.createElement(doc, 'child2');
```


版权所有 © 华为云计算技术有限公司

```
  node := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(child1),
DBE_XMLDOM.makeNode(child2));
  parent := DBE_XMLDOM.GETPARENTNODE(DBE_XMLDOM.makeNode(child2));
  buf := DBE_XMLDOM.GETNODENAME(parent);
  DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
child1
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETTAGNAME

  返回指定DOMELEMENT的标签名称。DBE_XMLDOM.GETTAGNAME的函数原型为：

```
DBE_XMLDOM.GETTAGNAME(
  elem  IN  DOMELEMENT)
RETURN  VARCHAR2;
```

表 2-449 DBE_XMLDOM.GETTAGNAME 接口参数说明

参数	描述
elem	指定的DOMELEMENT节点。

示例：

```
--创建DOMELEMENT节点后，输出其标签名称。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  buffer varchar2(1010);
  value  varchar(1000);
BEGIN
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(DBE_XMLDOM.NEWDOMDOCUMENT(), 'root');
  value := DBE_XMLDOM.gettagname(elem);
  dbe_output.print_line('value: ');
  dbe_output.print_line(value);
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
value:
root
buffer:
<?xml version="1.0" encoding="UTF-8"?>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.HASCHILDNODES

  检查DOMNODE对象是否拥有任一子节点。DBE_XMLDOM.HASCHILDNODES的函数原型为：

```
DBE_XMLDOM.HASCHILDNODES(
  n  IN  DOMNODE)
RETURN  BOOLEAN;
```


版权所有 © 华为云计算技术有限公司

表 2-450 DBE_XMLDOM.HASCHILDNODES 接口参数说明

参数	描述
n	指定的DOMNODE对象。

示例：

```
--创建节点child1并将其挂载到DOC树中，为child1节点添加子节点后，判断其是否拥有任一子节点。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  doc1 DBE_XMLDOM.DOMDocument;
  root DBE_XMLDOM.DOMElement;
  child1 DBE_XMLDOM.DOMElement;
  child2 DBE_XMLDOM.DOMElement;
  attr DBE_XMLDOM.DOMAttr;
  text DBE_XMLDOM.DOMTEXT;
  node DBE_XMLDOM.DOMNode;
  buf varchar2(1000);
BEGIN
  doc := DBE_XMLDOM.newDOMDocument();
  root := DBE_XMLDOM.createElement(doc, 'root');
  node := DBE_XMLDOM.appendChild(DBE_xmldom.makeNode(doc),DBE_xmldom.makeNode(root));
  child1 := DBE_XMLDOM.createElement(doc, 'child1');
  node := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(root),
DBE_XMLDOM.makeNode(child1));
  child2 := DBE_XMLDOM.createElement(doc, 'child2');
  node := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(child1),
DBE_XMLDOM.makeNode(child2));
  IF DBE_XMLDOM.HASCHILDNODES(DBE_XMLDOM.makeNode(child1)) THEN
    DBE_OUTPUT.print_line('HAS CHILD NODES');
  ELSE
    DBE_OUTPUT.print_line('NOT HAS CHILD NODES ');
  END IF;
END;
/
-- 预期结果为：
HAS CHILD NODES
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.IMPORTNODE

  该函数将节点复制到另一节点中，并将复制后的节点挂载到指定document中。若被复制节点的类型不属于xmldom的constants所规定的12种类型，则直接抛出类型不支持异常。DBE_XMLDOM.IMPORTNODE的函数原型为：

```
DBE_XMLDOM.IMPORTNODE(
  doc IN DOMDOCUMENT,
  importedNode IN DOMNODE,
  deep IN BOOLEAN)
RETURN DOMNODE;
```

表 2-451 DBE_XMLDOM.IMPORTNODE 接口参数说明

参数	描述
doc	节点挂载的文档。
importedNode	将要导入的节点。

版权所有 © 华为云计算技术有限公司

参数	描述
deep	设置递归导入： ● 如果为TRUE，则导入该节点及其所有子节点。 ● 如果为FALSE，则只导入节点本身。

示例：

```
--获取将DOC2树中的节点root2_node，并将其复制并挂载到DOC树中。
DECLARE
    doc dbe_xmldom.domdocument;
    doc2 dbe_xmldom.domdocument;
    doc_node dbe_xmldom.domnode;
    doc2_node dbe_xmldom.domnode;
     root_node dbe_xmldom.domnode;
    root2_node dbe_xmldom.domnode;
    import_node dbe_xmldom.domnode;
    result_node dbe_xmldom.domnode;
    buffer varchar2(1010);
BEGIN
    doc := dbe_xmldom.newdomdocument('<bookstore category="web" cover="paperback">
       <book category="cooking"><title lang="en">Everyday Italian</title>
       <author>Giada De Laurentiis</author><year>2005</year>
       <price>30.00</price></book></bookstore>');
    doc2 := dbe_xmldom.newdomdocument('<case>LIANLI TU150</case>');
    doc_node := DBE_XMLDOM.MAKENODE(doc);
    doc2_node := DBE_XMLDOM.MAKENODE(doc2);
    root_node := DBE_XMLDOM.GETFIRSTCHILD(doc_node);
    root2_node := DBE_XMLDOM.GETFIRSTCHILD(doc2_node);
    DBE_XMLDOM.WRITETOBUFFER(doc, buffer);
    dbe_output.print_line(buffer);
    import_node := DBE_XMLDOM.IMPORTNODE(doc, root2_node, TRUE);
    result_node := DBE_XMLDOM.APPENDCHILD(root_node, import_node);
    DBE_XMLDOM.WRITETOBUFFER(doc, buffer);
    dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
<?xml version="1.0" encoding="UTF-8"?>
<bookstore category="web" cover="paperback">
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>

<?xml version="1.0" encoding="UTF-8"?>
<bookstore category="web" cover="paperback">
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <case>LIANLI TU150</case>
</bookstore>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.ISNULL

版权所有 © 华为云计算技术有限公司

检测给定的DOMATTR类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  a      IN     DOMATTR)
RETURN BOOLEAN;
```

检测给定的DOMDOCUMENT类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  doc      IN     DOMDOCUMENT)
RETURN BOOLEAN;
```

检测给定的DOMELEMENT类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  elem     IN     DOMELEMENT)
RETURN BOOLEAN;
```

检测给定的DOMNAMEDNODEMAP类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  nnm      IN     DOMNAMEDNODEMAP)
RETURN BOOLEAN;
```

检测给定的DOMNODE类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  n      IN     DOMNODE)
RETURN BOOLEAN;
```

检测给定的DOMNODELIST类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  nl       IN     DOMNODELIST)
RETURN BOOLEAN;
```

检测给定的DOMTEXT类型节点是否为NULL。如果是返回TRUE，否则返回FALSE。DBE_XMLDOM.ISNULL的函数原型为：

```
DBE_XMLDOM.ISNULL(
  t      IN     DOMTEXT)
RETURN BOOLEAN;
```

表 2-452 DBE_XMLDOM.ISNULL 接口参数说明

参数	描述
a	指定的DOMATTR类型节点。
doc	指定的DOMDOCUMENT类型节点。
elem	指定的DOMELEMENT类型节点。
nnm	指定的DOMNAMEDNODEMAP类型节点。
n	指定的DOMNODE类型节点。
nl	指定的DOMNODELIST类型节点。
t	指定的DOMTEXT类型节点。

版权所有 © 华为云计算技术有限公司

说明

由于DBE_XMLDOM.FREEDOCUMENT的实现差异，DBE_XMLDOM.ISNULL接口在调用释放后的DOMDOCUMENT节点时会发生报错。

示例：

```
--1. 通过createAttribute创建DOMATTR节点，并判断其是否为空。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  attr DBE_XMLDOM.DOMATTR;
  buf VARCHAR2(1000);
BEGIN
  doc := DBE_xmldom.newdomdocument('<?xml version="1.0"?>
    <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
    <!ATTLIST note color CDATA #REQUIRED>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>]>
    <note color="red"><to>中文</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don''t forget me this weekend!</body>
    </note>');
  attr := DBE_XMLDOM.CREATEATTRIBUTE (doc, 'length');
  if DBE_XMLDOM.ISNULL(attr) then
    DBE_OUTPUT.print_line('null');
  else
    DBE_OUTPUT.print_line('not null');
  end if;
END;
/
-- 预期结果为：
not null
ANONYMOUS BLOCK EXECUTE

--2. DOMELEMENT仅声明不初始化，并判断其是否为空。
DECLARE
  docelem   DBE_XMLDOM.DOMElement;
BEGIN
  if DBE_XMLDOM.ISNULL(docelem) then
    DBE_OUTPUT.print_line('null');
  else
    DBE_OUTPUT.print_line('not null');
  end if;
END;
/
-- 预期结果为：
null
ANONYMOUS BLOCK EXECUTE

--3. 通过newDomdocument构建良构的DOMDOCUMENT节点，判断其是否为空。
Declare
 doc dbe_xmldom.domdocument;
BEGIN
 doc := DBE_xmldom.newdomdocument('<?xml version="1.0"?>
    <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
    <!ATTLIST note color CDATA #REQUIRED>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>]>
    <note color="red"><to>中文</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don''t forget me this weekend!</body>
    </note>');
 if DBE_XMLDOM.ISNULL(doc) then
    DBE_OUTPUT.print_line('null');
```


版权所有 © 华为云计算技术有限公司

```
  else
    DBE_OUTPUT.print_line('not null');
  end if;
END;
/
-- 预期结果为：
not null
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.ITEM

  根据索引返回list中与索引对应的元素。DBE_XMLDOM.ITEM的函数原型为：

```
DBE_XMLDOM.ITEM(
  nl IN DOMNODELIST,
  index IN NUMBER)
RETURN DOMNODE;
```

  根据索引返回map中与索引对应的元素。DBE_XMLDOM.ITEM的函数原型为：

```
DBE_XMLDOM.ITEM(
  nnm IN DOMNAMEDNODEMAP,
  index IN NUMBER)
RETURN DOMNODE;
```

表 2-453 DBE_XMLDOM.ITEM 接口参数说明

参数	描述
nl	DOMNODELIST。
nnm	DOMNAMEDNODEMAP。
index	要检索的元素的索引。

说明

map类型函数item对不合理的参数输入：如bool、clob，会默认指向第一个index的值。

示例：

```
--1. 根据索引返回map中与索引对应的元素。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  elem DBE_XMLDOM.DOMElement;
  map DBE_XMLDOM.DOMNAMEDNODEMAP;
  node DBE_XMLDOM.DOMNODE;
  node2 DBE_XMLDOM.DOMNODE;
  buf varchar2(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument('<bookstore category="web" cover="paperback"><book
category="cooking">
      <title lang="en">Everyday Italian</title><author>Giada De Laurentiis</author>
      <year>2005</year><price>30.00</price></book></bookstore');
  elem := DBE_XMLDOM.GETDOCUMENTELEMENT(doc);
  node := DBE_XMLDOM.MAKENODE(elem);
  map := DBE_XMLDOM.GETATTRIBUTES(DBE_XMLDOM.getFirstChild(node));
  node2:= DBE_XMLDOM.item(map,0);
  DBE_XMLDOM.writeToBuffer(node2, buf);
  dbe_output.print_line(buf);
  dbe_xmldom.freedocument(doc);
  RAISE NOTICE '%', buf;
END;
/
-- 预期结果为：
cooking
NOTICE:  cooking
ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

```
--2. 根据索引返回list中与索引对应的元素。
DECLARE
  doc dbe_xmldom.domdocument;
  node dbe_xmldom.domnode;
  node1 dbe_xmldom.domnode;
  nodelist DBE_XMLDOM.DOMNODELIST;
  len INTEGER;
  buffer1 varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<bookstore category="web" cover="paperback"><book
category="cooking">
      <title lang="en">Everyday Italian</title><author>Giada De Laurentiis</author>
      <year>2005</year><price>30.00</price></book></bookstore>');
  node := dbe_xmldom.makenode(doc);
  node := dbe_xmldom.GETFIRSTCHILD(node);
  node := dbe_xmldom.GETFIRSTCHILD(node);
  nodelist := DBE_XMLDOM.GETCHILDNODES(node);
  len := DBE_XMLDOM.GETLENGTH(nodelist);
  RAISE NOTICE 'len :  %', len;
  node1 := DBE_XMLDOM.ITEM(nodelist, 0);
  IF DBE_XMLDOM.ISNULL(node1) THEN
        dbe_output.print_line('IS NULL');
  ELSE
        dbe_output.print_line('NOT NULL');
  END IF;
  dbe_xmldom.writetobuffer(node1, buffer1);
  dbe_output.print_line('buffer1: ');
  dbe_output.print_line(buffer1);
END;
/
-- 预期结果为：
NOTICE:  len :  4
NOT NULL
buffer1:
<title lang="en">Everyday Italian</title>
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.MAKEELEMENT

  返回转换后的DOMELEMENT对象。DBE_XMLDOM.MAKEELEMENT的函数原型为：

```
DBE_XMLDOM.MAKEELEMENT(
  n IN DOMNODE)
RETURN DOMELEMENT;
```

表 2-454 DBE_XMLDOM.MAKEELEMENT 接口参数说明

参数	描述
n	指定的DOMNODE对象。

示例：

```
--将DOMELEMENT类型转换后的DOMNODE类型节点node强制转换回DOMELEMENT类型。
DECLARE
  buf VARCHAR2(1000);
  doc DBE_XMLDOM.DOMDocument;
  elem DBE_XMLDOM.DOMElement;
  elem2 DBE_XMLDOM.DOMElement;
  node DBE_XMLDOM.DOMNode;
BEGIN
  doc := DBE_XMLDOM.NEWDOMDOCUMENT();
  elem := DBE_XMLDOM.createElement(doc, 'aaa');
  node := DBE_XMLDOM.makeNode(elem);
  elem2 := DBE_XMLDOM.makeElement(node);
  buf := DBE_XMLDOM.GETNODENAME(DBE_XMLDOM.makeNode(elem2));
```


版权所有 © 华为云计算技术有限公司

```
  DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
aaa
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.MAKENODE

  将给定的DOMATTR类型节点强制转换为DOMNODE类型，返回DOMNODE节点。DBE_XMLDOM.MAKENODE的函数原型为：

```
DBE_XMLDOM.MAKENODE(
  a       IN     DOMATTR)
 RETURN DOMNODE;
```

  将给定的DOMDOCUMENT类型节点强制转换为DOMNODE类型，返回DOMNODE节点。DBE_XMLDOM.MAKENODE的函数原型为：

```
DBE_XMLDOM.MAKENODE(
  doc      IN     DOMDOCUMENT)
 RETURN DOMNODE;
```

  将给定的DOMELEMENT类型节点强制转换为DOMNODE类型，返回DOMNODE节点。DBE_XMLDOM.MAKENODE的函数原型为：

```
DBE_XMLDOM.MAKENODE(
  elem       IN     DOMELEMENT)
 RETURN DOMNODE;
```

  将给定的DOMTEXT类型节点强制转换为DOMNODE类型，返回DOMNODE节点。DBE_XMLDOM.MAKENODE的函数原型为：

```
DBE_XMLDOM.MAKENODE(
  t       IN     DOMTEXT)
 RETURN DOMNODE;
```

表 2-455 DBE_XMLDOM.MAKENODE 接口参数说明

参数	描述
a	指定的DOMATTR类型节点。
doc	指定的DOMDOCUMENT类型节点。
elem	指定的DOMELEMENT类型节点。
t	指定的DOMTEXT类型节点。

说明

由于语法限制，DBE_XMLDOM.MAKENODE作为函数返回值时，不能直接通过如下命令实现：

```
return DBE_XMLDOM.MAKENODE(doc);
```

建议写为：

```
tmp_node := DBE_XMLDOM.MAKENODE(doc );
return tmp_node;
```

示例：

```
--1. createattr生成ATTR,将其转换为node。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  attr DBE_XMLDOM.DOMATTR;
  dom_node DBE_XMLDOM.DOMNode;
  buf VARCHAR2(1000);
BEGIN
  doc := DBE_xmldom.newdomdocument('<?xml version="1.0"?>
```


版权所有 © 华为云计算技术有限公司

```
    <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>]>
    <note><to>中文</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don''t forget me this weekend!</body>
    </note>');
  attr := DBE_XMLDOM.CREATEATTRIBUTE (doc, 'length');
  dom_node := DBE_XMLDOM.makeNode(attr);
  buf := DBE_XMLDOM.getNodeName(dom_node);
  DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
length
ANONYMOUS BLOCK EXECUTE

--2. getdocumentelement函数生成elem节点后进行makenode。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root DBE_XMLDOM.DOMElement;
  attr DBE_XMLDOM.DOMATTR;
  node DBE_XMLDOM.DOMNODE;
  buf VARCHAR2(1000);
BEGIN
  doc := DBE_xmldom.newdomdocument('<?xml version=''1.0''?>
    <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
    <!ATTLIST note color CDATA #REQUIRED>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>]>
    <note color="red"><to>中文</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don''t forget me this weekend!</body>
    </note>');
  root := DBE_XMLDOM.getDocumentElement(doc);
  node := DBE_XMLDOM.makenode(root);
  DBE_OUTPUT.print_line(DBE_XMLDOM.GETNODENAME(node));
END;
/
-- 预期结果为：
note
ANONYMOUS BLOCK EXECUTE

--3. 通过newdomdocument创建DOMDOCUMENT类型参数，非空内容，并作为MAKENODE的输入参
数。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  buf VARCHAR2(1000);
  dom_node DBE_XMLDOM.DOMNODE;
BEGIN
  doc := DBE_xmldom.newdomdocument('<?xml version=''1.0''?>
            <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)>
            <!ELEMENT to (#PCDATA)>
            <!ELEMENT from (#PCDATA)>
            <!ELEMENT heading (#PCDATA)>
            <!ELEMENT body (#PCDATA)>]>
            <note><to>中文</to>
            <from>Jani</from>
            <heading>Reminder</heading>
            <body>Don''t forget me this weekend!</body>
            </note>');
  DBE_OUTPUT.print_line('doc.id: ');
  DBE_OUTPUT.print_line(doc.id);
```


版权所有 © 华为云计算技术有限公司

```
  dom_node := DBE_XMLDOM.makeNode(doc);
  DBE_OUTPUT.print_line('dom_node.id: ');
  DBE_OUTPUT.print_line(dom_node.id);
  buf := DBE_XMLDOM.GETNODENAME(dom_node);
  DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为（若当前session之前执行过xmldom接口，则结果是不确定的）：
doc.id:
19000000000000001B00000001
dom_node.id:
19000000010000001B00000001
#document
ANONYMOUS BLOCK EXECUTE

--4. DOMTEXT声明变量，不初始化, 并作为MAKENODE的输入参数。
DECLARE
  text DBE_XMLDOM.DOMTEXT;
  buf VARCHAR2(1000);
  dom_node DBE_XMLDOM.DOMNODE;
BEGIN
  dom_node := DBE_XMLDOM.makeNode(text);
  buf := DBE_XMLDOM.GETNODENAME(dom_node);
  DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.NEWDOMDOCUMENT

  返回新的DOMDOCUMENT对象。DBE_XMLDOM.NEWDOMDOCUMENT的函数原型为：

  ```
  DBE_XMLDOM.NEWDOMDOCUMENT
  RETURN DOMDOCUMENT;
  ```

  返回从指定的XMLType类型创建的新DOMDOCUMENT实例对象。DBE_XMLDOM.NEWDOMDOCUMENT的函数原型为：

  ```
  DBE_XMLDOM.NEWDOMDOCUMENT(
    xmldoc    IN SYS.XMLTYPE)
  RETURN DOMDOCUMENT;
  ```

  返回从指定的CLOB类型创建的新DOMDOCUMENT实例对象。DBE_XMLDOM.NEWDOMDOCUMENT的函数原型为

  ```
  DBE_XMLDOM.NEWDOMDOCUMENT(
    cl       IN    CLOB)
  RETURN DOMDOCUMENT;
  ```

  表 2-456 DBE_XMLDOM.NEWDOMDOCUMENT 接口参数说明

参数	描述
xmldoc	指定的XMLType类型。
cl	指定的CLOB类型。

版权所有 © 华为云计算技术有限公司

说明

- 入参大小需限制在1GB以内。
- 目前暂不支持外部DTD解析。
- newdomdocument创建的doc，默认UTF-8字符集。
- 从同一个xmltype实例中解析出的每一个doc都是独立的，对doc的修改也不会影响到xmltype。
- 与ORA数据库差异参见**DBE_XMLPARSER.PARSECLOB**。

示例：

```
--1. 返回新的DOMDOCUMENT对象。
DECLARE
  doc dbe_xmldom.domdocument;
  buffer varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument();
  dbe_xmldom.setdoctype(doc, 'note', 'sysid', 'pubid');
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
  dbe_xmldom.freedocument(doc);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note PUBLIC "pubid" "sysid">

ANONYMOUS BLOCK EXECUTE

--2. 返回从指定的CLOB类型创建的新DOMDOCUMENT实例对象。
DECLARE
  doc dbe_xmldom.domdocument;
  buffer varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
      <note><to>test</to><from>Jani</from><heading>Reminder</heading>
      <body>Don''t forget me this weekend!</body></note>');
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
  dbe_xmldom.freedocument(doc);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<note>
 <to>test</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>

ANONYMOUS BLOCK EXECUTE

--3. 返回从指定的XMLType类型创建的新DOMDOCUMENT实例对象。
DECLARE
doc dbe_xmldom.domdocument;
  xt xmltype;
  buffer varchar2(1010);
BEGIN
  xt := xmltype('<h:data xmlns:h="http://www.w3.org/TR/html4/">
    <h:da1 len="10">test namespace</h:da1>
    <h:da1>bbbbbbbbbb</h:da1>
    </h:data>');
  doc := dbe_xmldom.newdomdocument(xt);
```


版权所有 © 华为云计算技术有限公司

```
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
  dbe_xmldom.freedocument(doc);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<h:data xmlns:h="http://www.w3.org/TR/html4/">
 <h:da1 len="10">test namespace</h:da1>
 <h:da1>bbbbbbbbbb</h:da1>
</h:data>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.SETATTRIBUTE

  按名称设置DOMELEMENT属性的值。DBE_XMLDOM.SETATTRIBUTE的函数原型为：

```
DBE_XMLDOM.SETATTRIBUTE(
  elem    IN  DOMELEMENT,
  name    IN  VARCHAR2,
  value   IN  VARCHAR2);
```

  按名称和命名空间URI设置DOMELEMENT属性的值。
  DBE_XMLDOM.SETATTRIBUTE的函数原型为：

```
DBE_XMLDOM.SETATTRIBUTE(
  elem    IN  DOMELEMENT,
  name    IN  VARCHAR2,
  value   IN  VARCHAR2,
  ns      IN  VARCHAR2);
```

表 2-457 DBE_XMLDOM.SETATTRIBUTE 接口参数说明

参数	描述
elem	指定的DOMELEMENT节点。
name	属性名称。
value	属性值。
ns	命名空间。

说明

DBE_XMLDOM.SETATTRIBUTE接口可以添加多个属性，属性名称不可以为null，且同一个DOMELEMENT节点不能出现同名属性。如需添加同名属性，应显式的为每个同名属性设置命名空间，但是应尽量避免此类操作。如果属性存在于某命名空间下，当修改属性时，应显式指定命名空间，否则视为添加同名属性。

示例：

```
--1. 按名称设置DOMELEMENT属性的值。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  docnode DBE_XMLDOM.DOMNode;
  buffer varchar2(1010);
  value  varchar(1000);
BEGIN
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(doc, 'root');
  DBE_XMLDOM.setattribute(elem, 'len', '50cm');
```


版权所有 © 华为云计算技术有限公司

```
  docnode := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(doc),
DBE_XMLDOM.makeNode(elem));
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<root len="50cm"/>

ANONYMOUS BLOCK EXECUTE

--2. 按名称和命名空间URI设置DOMELEMENT属性的值。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  docnode DBE_XMLDOM.DOMNode;
  buffer varchar2(1010);
  value  varchar(1000);
begin
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(doc, 'root');
  DBE_XMLDOM.setattribute(elem, 'len', '50cm', 'www.xxxxx.com');
  docnode := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(doc),
DBE_XMLDOM.makeNode(elem));
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<root len="50cm"/>

ANONYMOUS BLOCK EXECUTE

--3. 按名称修改DOMELEMENT属性的值。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
  docnode DBE_XMLDOM.DOMNode;
  buffer varchar2(1010);
  value  varchar(1000);
BEGIN
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(doc, 'root');
  DBE_XMLDOM.setattribute(elem, 'len', '50cm');
  DBE_XMLDOM.setattribute(elem, 'len', '55cm');
  docnode := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(doc),
DBE_XMLDOM.makeNode(elem));
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<root len="55cm"/>

ANONYMOUS BLOCK EXECUTE

--4. 按名称和命名空间URI修改DOMELEMENT属性的值。
DECLARE
  doc dbe_xmldom.domdocument;
  elem dbe_xmldom.domelement;
```


版权所有 © 华为云计算技术有限公司

```
  docnode DBE_XMLDOM.DOMNode;
  buffer varchar2(1010);
  value  varchar(1000);
begin
  doc := dbe_xmldom.newDOMDocument();
  elem := DBE_XMLDOM.CREATEELEMENT(doc, 'root');
  DBE_XMLDOM.setattribute(elem, 'len', '50cm', 'www.xxxxx.com');
  DBE_XMLDOM.setattribute(elem, 'len', '55cm', 'www.xxxxx.com');
  docnode := DBE_XMLDOM.appendChild(DBE_XMLDOM.makeNode(doc),
DBE_XMLDOM.makeNode(elem));
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<root len="55cm"/>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.SETCHARSET

  设置DOMDOCUMENT的CHARSET字符集。DBE_XMLDOM.SETCHARSET的函数原型为：

```
DBE_XMLDOM.SETCHARSET(
  doc       IN     DOMDocument,
  charset   IN     VARCHAR2);
```

表 2-458 DBE_XMLDOM.SETCHARSET 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点
charset	字符集

说明

- charset限制为60个字节以内。
- 目前支持的字符集有：UTF-8、UCS-4、UCS-2、ISO-8859-1、ISO-8859-2、ISO-8859-3、ISO-8859-4、ISO-8859-5、ISO-8859-6、ISO-8859-7、ISO-8859-8、ISO-8859-9、ISO-2022-JP、Shift_JIS、EUC-JP、ASCII。输入其他字符集会报错或者可能导致输出乱码。

示例：

```
--为DOC树设置UTF-8字符集后，将DOC树输出到缓冲区。
DECLARE
  doc dbe_xmldom.domdocument;
  buffer varchar2(1010);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)><!ELEMENT to (#PCDATA)>
     <!ELEMENT from (#PCDATA)><!ELEMENT heading (#PCDATA)><!ELEMENT body (#PCDATA)>]>
     <note><to>test</to><from>Jani</from><heading>Reminder</heading>
     <body>Don''t forget me this weekend!</body></note>');
  dbe_xmldom.setcharset(doc, 'utf-8');
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
  dbe_xmldom.freedocument(doc);
END;
/
```


版权所有 © 华为云计算技术有限公司

```
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
<!ELEMENT note (to , from , heading , body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
  <to>test</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.SETDOCTYPE

  设置DOMDOCUMENT的外部DTD。DBE_XMLDOM.SETDOCTYPE的函数原型为：

```
DBE_XMLDOM.SETDOCTYPE(
  doc     IN   DOMDocument,
  name    IN   VARCHAR2,
  sysid   IN   VARCHAR2,
  pubid   IN   VARCHAR2);
```

表 2-459 DBE_XMLDOM.SETDOCTYPE 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。
name	需要初始化doctype的名称。
sysid	需要初始化doctype的system ID。
pubid	需要初始化doctype的public ID。

说明

name、sysid、pubid的总长度限制在32500个字节以内。

示例：

```
--为DOMDOCUMENT的外部DTD分别设置初始化的system ID、public ID和名称后，分别将每次修改后的DOC树输出到缓冲区。
DECLARE
  doc dbe_xmldom.domdocument;
  buffer varchar2(1010);
begin
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
     <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)><!ELEMENT to (#PCDATA)>
     <!ELEMENT from (#PCDATA)><!ELEMENT heading (#PCDATA)><!ELEMENT body (#PCDATA)>]>
     <note><to>test</to><from>Jani</from><heading>Reminder</heading>
     <body>Don''t forget me this weekend!</body></note>');
  dbe_xmldom.setdoctype(doc, 'note', 'sysid', 'pubid');
  dbe_xmldom.writetobuffer(doc, buffer);
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
  dbe_output.print_line('-----------------------------------------------');
  dbe_xmldom.setdoctype(doc, 'n0te', NULL, '');
  dbe_xmldom.setdoctype(doc, 'n0t1e', NULL, '');
  dbe_xmldom.writetobuffer(doc, buffer);
```


版权所有 © 华为云计算技术有限公司

```
  dbe_output.print_line('buffer: ');
  dbe_output.print_line(buffer);
  dbe_xmldom.freedocument(doc);
END;
/
-- 预期结果为：
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note PUBLIC "pubid" "sysid" [
<!ELEMENT note (to , from , heading , body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
  <to>test</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

-----------------------------------------------
buffer:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE n0t1e [
<!ELEMENT note (to , from , heading , body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
  <to>test</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.SETNODEVALUE

  此函数用于向DOMNODE对象中设置节点的值。DBE_XMLDOM.SETNODEVALUE的函数原型为：

```
DBE_XMLDOM.SETNODEVALUE(
 n IN DOMNODE,
 nodeValue IN VARCHAR2);
```

表 2-460 DBE_XMLDOM.SETNODEVALUE 接口参数说明

参数	描述
n	指定的DOMNODE对象。
nodeValue	向DOMNODE对象中设置的字符串。

说明

- nodeValue可以输入空字符串和NULL值，但不会对节点值进行修改。
- nodeValue支持转义字符'&'，序列化时自动转义。
- nodeValue默认的最大长度受限于VARCHAR2类型，为32767字节，超过该长度会抛出异常。

版权所有 © 华为云计算技术有限公司

示例：

```
--对DOMTEXT转换后的DOMNODE节点设置与初始值不同的节点值后，获取并输出该节点的值。
DECLARE
  buf VARCHAR2(1000);
  doc DBE_XMLDOM.DOMDocument;
  text DBE_XMLDOM.DOMText;
  elem2 DBE_XMLDOM.DOMElement;
  node DBE_XMLDOM.DOMNode;
BEGIN
  doc := DBE_XMLDOM.NEWDOMDOCUMENT();
  text := DBE_XMLDOM.createTextNode(doc, 'aaa');
  DBE_XMLDOM.SETNODEVALUE(DBE_XMLDOM.makeNode(text), 'ccc');
  buf := DBE_XMLDOM.GETNODEVALUE(DBE_XMLDOM.makeNode(text));
  DBE_OUTPUT.print_line(buf);
END;
/
-- 预期结果为：
ccc
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.WRITETOBUFFER

  使用数据库字符集将 XML 节点写入指定缓冲区。
  DBE_XMLDOM.WRITETOBUFFER的函数原型为：

```
DBE_XMLDOM.WRITETOBUFFER(
   doc       IN      DOMDOCUMENT,
   buffer    INOUT   VARCHAR2);
```

  使用数据库字符集将 XML 文档写入指定缓冲区。
  DBE_XMLDOM.WRITETOBUFFER的函数原型为：

```
DBE_XMLDOM.WRITETOBUFFER(
   n         IN      DOMNODE,
   buffer    INOUT   VARCHAR2);
```

表 2-461 DBE_XMLDOM.WRITETOBUFFER 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。
buffer	写入操作的缓冲区。
n	指定的DOMNODE节点。

说明

- writetobuffer输出buffer限制在1GB以内。
- 该函数会添加缩进等内容，将输出格式化。输出doc将包含XML声明version和encoding。
- 默认以UTF-8字符集输出xml。

示例：

```
--1. 输入DOMNODE类型参数。
DECLARE
   doc dbe_xmldom.domdocument;
   elem DBE_XMLDOM.DOMELEMENT;
   buf varchar2(1000);
BEGIN
    doc := dbe_xmldom.newdomdocument();
    elem := dbe_xmldom.createelement(doc,'elem');
    DBE_XMLDOM.WRITETOBUFFER(dbe_xmldom.makenode(elem), buf);
    DBE_OUTPUT.print_line(buf);
END;
```


版权所有 © 华为云计算技术有限公司

```
/
-- 预期结果为：
<elem/>
ANONYMOUS BLOCK EXECUTE

--2. 输入DOMDOCUMENT类型参数。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  buf VARCHAR2(1000);
BEGIN
  doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
       <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)><!ELEMENT to (#PCDATA)>
       <!ELEMENT from (#PCDATA)><!ELEMENT heading (#PCDATA)><!ELEMENT body (#PCDATA)>]>
       <note><to>test</to><from>Jani</from><heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body></note>');
  DBE_XMLDOM.WRITETOBUFFER(doc, buf);
  DBE_OUTPUT.print_line('doc: ');
  DBE_OUTPUT.print_line(buf);
  DBE_XMLDOM.FREEDOCUMENT(doc);
END;
/
-- 预期结果为：
doc:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
<!ELEMENT note (to , from , heading , body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
  <to>test</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.WRITETOCLOB

  使用数据库字符集将 XML 节点写入指定的 CLOB。
  DBE_XMLDOM.WRITETOCLOB的函数原型为：

```
DBE_XMLDOM.WRITETOCLOB(
  doc     IN      DOMDOCUMENT,
  cl      INOUT  CLOB);
```

  使用数据库字符集将 XML 节点写入指定的 CLOB。
  DBE_XMLDOM.WRITETOCLOB的函数原型为：

```
DBE_XMLDOM.WRITETOCLOB(
  n       IN      DOMNODE,
  cl      INOUT  CLOB);
```

表 2-462 DBE_XMLDOM.WRITETOCLOB 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。
cl	要写入的CLOB。
n	指定的DOMNODE节点。

版权所有 © 华为云计算技术有限公司

说明

- document入参，writetoclob大小支持1GB以内。
- 该函数会添加缩进等内容，将输出格式化。输出doc将包含XML声明version和encoding。
- 默认以UTF-8字符集输出xml。

示例：

```
--1. 输入DOMNODE类型参数。
DECLARE
  CL  CLOB;
  N   DBE_XMLDOM.DOMNODE;
BEGIN
  DBE_XMLDOM.WRITETOCLOB(N, CL);
  DBE_OUTPUT.PRINT_LINE(CL);
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE

--2. 输入DOMDOCUMENT类型参数。
DECLARE
    doc dbe_xmldom.domdocument;
    mclob clob;
BEGIN
    doc := dbe_xmldom.newdomdocument('<?xml version="1.0"?>
       <!DOCTYPE note [<!ELEMENT note (to,from,heading,body)><!ELEMENT to (#PCDATA)>
       <!ELEMENT from (#PCDATA)><!ELEMENT heading (#PCDATA)><!ELEMENT body (#PCDATA)>]>
       <note><to>test</to><from>Jani</from><heading>Reminder</heading>
       <body>Don''t forget me this weekend!</body></note>');
    dbe_xmldom.writetoclob(doc, mclob);
    dbe_output.print_line('mclob: ');
    dbe_output.print_line(mclob);
    dbe_xmldom.freedocument(doc);
END;
/
-- 预期结果为：
mclob:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
<!ELEMENT note (to , from , heading , body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
  <to>test</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.WRITETOFILE

  使用数据库字符集将 XML 节点写入指定文件。DBE_XMLDOM.WRITETOFILE的函数原型为：

  ```
  DBE_XMLDOM.WRITETOCLOB(
    doc     IN      DOMDOCUMENT,
    fileName   IN      VARCHAR2);
  ```

  使用数据库字符集将 XML 节点写入指定文件。DBE_XMLDOM.WRITETOFILE的函数原型为：

  ```
  DBE_XMLDOM.WRITETOCLOB(
    n       IN      DOMNODE,
    fileName   IN      VARCHAR2);
  ```


版权所有 © 华为云计算技术有限公司

使用指定字符集将 XML 文档写入指定文件。DBE_XMLDOM.WRITETOFILE的函数原型为：

```
DBE_XMLDOM.WRITETOCLOB(
  doc     IN      DOMDOCUMENT,
  fileName   IN      VARCHAR2,
  charset   IN   VARCHAR2);
```

使用指定字符集将 XML 文档写入指定文件。DBE_XMLDOM.WRITETOFILE的函数原型为：

```
DBE_XMLDOM.WRITETOCLOB(
  n       IN      DOMNODE,
  fileName   IN      VARCHAR2,
  charset   IN   VARCHAR2);
```

表 2-463 DBE_XMLDOM.WRITETOFILE 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。
fileName	要写入的文件。
n	指定的DOMNODE节点。
charset	指定字符集。

说明

- document入参，filename长度限制在255个字节以内，charset限制在60个字节以内，charset支持字符集请参考**DBE_XMLDOM.SETCHARSET**接口。
- 该函数会添加缩进等内容，将输出格式化。输出doc将包含XML声明version和encoding。
- 传入newdomdocument()无参创建的doc，在不指定charset时不会报错，默认UTF-8字符集。
- filename需要在pg_directory中创建的路径下，filename中的\会被转换成/，只允许存在一个/。文件名格式应为pg_directory_name/file_name.xml，输出文件仅支持xml类型。
- 在打开guc参数safe_data_path时，用户只能通过高级包读写safe_data_path指定文件路径下的文件。
- 创建目录前需要保证路径为操作系统实际存在的路径，且用户需要拥有该目录的读和写权限。关于目录创建，请参考**CREATE DIRECTORY**。

示例：

```
--创建目录前需要保证路径为操作系统实际存在的路径，且用户需要拥有该目录的读和写权限
create directory dir as '/tmp';
--1. 使用数据库字符集将 XML 节点写入指定文件。
DECLARE
  FPATH VARCHAR2(1000);
  DOC   DBE_XMLDOM.DOMDOCUMENT;
BEGIN
  DOC := DBE_XMLDOM.NEWDOMDOCUMENT('<ROOT>
    <A ATTR1="A_VALUE">
        <ACHILD>ACHILD TXT</ACHILD>
    </A>
    <B>B TXT</B>
    <C/>
  </ROOT>');
  FPATH := 'dir/simplexml.xml';
  DBE_XMLDOM.WRITETOFILE(DOC, FPATH);
END;
```


版权所有 © 华为云计算技术有限公司

```
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE

--2. 使用指定字符集将 XML 文档写入指定文件。
DECLARE
  SRC    VARCHAR(1000);
  FPATH VARCHAR2(1000);
  DOC    DBE_XMLDOM.DOMDOCUMENT;
  ELE    DBE_XMLDOM.DOMELEMENT;
BEGIN
  FPATH := 'dir/simplexml.xml';
  SRC := '<ROOT>
    <A ATTR1="A_VALUE">
        <ACHILD>ACHILD TXT</ACHILD>
    </A>
    <B>B TXT</B>
    <C/>
  </ROOT>';
  DOC := DBE_XMLDOM.NEWDOMDOCUMENT(SRC);
  ELE := DBE_XMLDOM.GETDOCUMENTELEMENT(DOC);
  DBE_XMLDOM.WRITETOFILE(DBE_XMLDOM.MAKENODE(ELE), FPATH, 'ASCII');
  DBE_XMLDOM.FREEDOCUMENT(DOC);
END;
/
-- 预期结果为：
ANONYMOUS BLOCK EXECUTE

-- 清理环境
drop directory dir;
```

- DBE_XMLDOM.GETSESSIONTREENUM

  查询当前session中所有类型的dom树数量。

  DBE_XMLDOM.GETSESSIONTREENUM的函数原型为：

```
DBE_XMLDOM.GETSESSIONTREENUM()
RETURN INTEGER;
```

  说明

  对于使用过FREEELEMENT和FREENODE的dom树，该函数依然会将其统计在内。

  示例：

```
-- 创建三个document，并获取当前session中所有dom树的数量
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  doc2 DBE_XMLDOM.DOMDocument;
  doc3 DBE_XMLDOM.DOMDocument;

  buffer varchar2(1010);
BEGIN
  -- 创建三个document
  doc := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<root>
    <elem1 attr="attrtest">
        <elem2>Im text</elem2>
        <elem3>Im text too</elem3>
    </elem1>
    <elem4>Text</elem4>
</root>
');
  doc2 := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<computer size="ITX" price="19999">
    <cpu>Ryzen 9 3950X</cpu>
    <ram>32GBx2 DDR4 3200MHz</ram>
    <motherboard>ROG X570i</motherboard>
    <gpu>RTX2070 Super</gpu>
    <ssd>1TB NVMe Toshiba + 2TB NVMe WD Black</ssd>
    <hdd>12TB WD Digital</hdd>
    <psu>CORSAIR SF750</psu>
```


版权所有 © 华为云计算技术有限公司

```
    <case>LIANLI TU150</case>
</computer>
');
  doc3 := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<bookstore>
    <book genre="autobiography" publicationdate="1981" ISBN="1-861003-11-0">
        <title>The Autobiography of Benjamin Franklin</title>
        <author>
            <first-name>Benjamin</first-name>
            <last-name>Franklin</last-name>
        </author>
        <price>8.99</price>
    </book>
    <book genre="novel" publicationdate="1967" ISBN="0-201-63361-2">
        <title>The Confidence Man</title>
        <author>
            <first-name>Herman</first-name>
            <last-name>Melville</last-name>
        </author>
        <price>11.99</price>
    </book>
    <book genre="philosophy" publicationdate="1991" ISBN="1-861001-57-6">
        <title>The Gorgias</title>
        <author>
            <name>Plato</name>
        </author>
        <price>9.99</price>
    </book>
</bookstore>
');
  -- 打印id
  DBE_OUTPUT.PRINT_LINE(doc.id);
  DBE_OUTPUT.PRINT_LINE(doc2.id);
  DBE_OUTPUT.PRINT_LINE(doc3.id);
  -- 调用该函数并打印
  DBE_OUTPUT.PRINT_LINE(DBE_XMLDOM.GETSESSIONTREENUM());
  -- 释放document
  DBE_XMLDOM.FREEDOCUMENT(doc);
  DBE_XMLDOM.FREEDOCUMENT(doc2);
  DBE_XMLDOM.FREEDOCUMENT(doc3);
END;
/
-- 预期结果为（若当前session之前执行过xmldom接口，则结果是不确定的）：
00000000000000000200000001
01000000000000000300000001
02000000000000000400000001
3
ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETDOCTREESINFO

  查询当前session中Document类型的dom树信息，如内存占用等。
  DBE_XMLDOM.GETDOCTREESINFO的函数原型为：

```
DBE_XMLDOM.GETDOCTREESINFO()
RETURN VARCHAR2;
```

  说明

  该函数只统计Document类型的dom树节点。

  示例：

```
-- 创建三个document，并获取当前session中document类型的树的信息
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  doc2 DBE_XMLDOM.DOMDocument;
  doc3 DBE_XMLDOM.DOMDocument;

  buffer varchar2(1010);
BEGIN
  -- 创建三个document
```


版权所有 © 华为云计算技术有限公司

```
  doc := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<root>
   <elem1 attr="attrtest">
      <elem2>Im text</elem2>
      <elem3>Im text too</elem3>
   </elem1>
   <elem4>Text</elem4>
</root>
');
  doc2 := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<computer size="ITX" price="19999">
   <cpu>Ryzen 9 3950X</cpu>
   <ram>32GBx2 DDR4 3200MHz</ram>
   <motherboard>ROG X570i</motherboard>
   <gpu>RTX2070 Super</gpu>
   <ssd>1TB NVMe Toshiba + 2TB NVMe WD Black</ssd>
   <hdd>12TB WD Digital</hdd>
   <psu>CORSAIR SF750</psu>
   <case>LIANLI TU150</case>
</computer>
');
  doc3 := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<bookstore>
   <book genre="autobiography" publicationdate="1981" ISBN="1-861003-11-0">
      <title>The Autobiography of Benjamin Franklin</title>
      <author>
         <first-name>Benjamin</first-name>
         <last-name>Franklin</last-name>
      </author>
      <price>8.99</price>
   </book>
   <book genre="novel" publicationdate="1967" ISBN="0-201-63361-2">
      <title>The Confidence Man</title>
      <author>
         <first-name>Herman</first-name>
         <last-name>Melville</last-name>
      </author>
      <price>11.99</price>
   </book>
   <book genre="philosophy" publicationdate="1991" ISBN="1-861001-57-6">
      <title>The Gorgias</title>
      <author>
         <name>Plato</name>
      </author>
      <price>9.99</price>
   </book>
</bookstore>
');
  -- 打印id
  DBE_OUTPUT.PRINT_LINE(doc.id);
  DBE_OUTPUT.PRINT_LINE(doc2.id);
  DBE_OUTPUT.PRINT_LINE(doc3.id);
  -- 调用该函数并打印
  DBE_OUTPUT.PRINT_LINE(DBE_XMLDOM.GETDOCTREESINFO());
  -- 释放document
  DBE_XMLDOM.FREEDOCUMENT(doc);
  DBE_XMLDOM.FREEDOCUMENT(doc2);
  DBE_XMLDOM.FREEDOCUMENT(doc3);
END;
/
-- 预期结果为（若当前session之前执行过xmldom接口，则结果是不确定的）：
00000000000000000200000001
01000000000000000300000001
02000000000000000400000001
|ID:00000000000000000200000001    |Node count:11    |Memory used:151 byte    |
|ID:01000000000000000300000001    |Node count:22    |Memory used:322 byte    |
|ID:02000000000000000400000001    |Node count:48    |Memory used:654 byte    |

ANONYMOUS BLOCK EXECUTE
```


版权所有 © 华为云计算技术有限公司

- DBE_XMLDOM.GETDETAILDOCTREEINFO

  查询传入的document内的各类型子节点的数量。
  DBE_XMLDOM.GETDETAILDOCTREEINFO的函数原型为：

```
DBE_XMLDOM.GETDETAILDOCTREEINFO(
   doc     IN      DOMDOCUMENT
)
RETURN VARCHAR2;
```

表 2-464 DBE_XMLDOM.GETDETAILDOCTREEINFO 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点

说明

该函数只统计Document类型的dom树节点。

示例：

```
-- 创建三个document，并使用该函数分别获取每一个document内的各类型节点数量
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  doc2 DBE_XMLDOM.DOMDocument;
  doc3 DBE_XMLDOM.DOMDocument;

  buffer varchar2(1010);
BEGIN
  -- 创建三个document
  doc := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<root>
   <elem1 attr="attrtest">
      <elem2>Im text</elem2>
      <elem3>Im text too</elem3>
   </elem1>
   <elem4>Text</elem4>
</root>
');
  doc2 := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<computer size="ITX" price="19999">
   <cpu>Ryzen 9 3950X</cpu>
   <ram>32GBx2 DDR4 3200MHz</ram>
   <motherboard>ROG X570i</motherboard>
   <gpu>RTX2070 Super</gpu>
   <ssd>1TB NVMe Toshiba + 2TB NVMe WD Black</ssd>
   <hdd>12TB WD Digital</hdd>
   <psu>CORSAIR SF750</psu>
   <case>LIANLI TU150</case>
</computer>
');
  doc3 := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0"?>
<bookstore>
   <book genre="autobiography" publicationdate="1981" ISBN="1-861003-11-0">
      <title>The Autobiography of Benjamin Franklin</title>
      <author>
         <first-name>Benjamin</first-name>
         <last-name>Franklin</last-name>
      </author>
      <price>8.99</price>
   </book>
   <book genre="novel" publicationdate="1967" ISBN="0-201-63361-2">
      <title>The Confidence Man</title>
      <author>
         <first-name>Herman</first-name>
```


版权所有 © 华为云计算技术有限公司

```
            <last-name>Melville</last-name>
        </author>
        <price>11.99</price>
    </book>
    <book genre="philosophy" publicationdate="1991" ISBN="1-861001-57-6">
        <title>The Gorgias</title>
        <author>
            <name>Plato</name>
        </author>
        <price>9.99</price>
    </book>
</bookstore>
');
  -- 打印id
  DBE_OUTPUT.PRINT_LINE(doc.id);
  DBE_OUTPUT.PRINT_LINE(doc2.id);
  DBE_OUTPUT.PRINT_LINE(doc3.id);
  -- 调用该函数并打印
  buffer := DBE_XMLDOM.GETDETAILDOCTREEINFO(doc);
  DBE_OUTPUT.PRINT_LINE(buffer);
  buffer := DBE_XMLDOM.GETDETAILDOCTREEINFO(doc2);
  DBE_OUTPUT.PRINT_LINE(buffer);
  buffer := DBE_XMLDOM.GETDETAILDOCTREEINFO(doc3);
  DBE_OUTPUT.PRINT_LINE(buffer);
  -- 释放document
  DBE_XMLDOM.FREEDOCUMENT(doc);
  DBE_XMLDOM.FREEDOCUMENT(doc2);
  DBE_XMLDOM.FREEDOCUMENT(doc3);
END;
/
-- 预期结果为（若当前session之前执行过xmldom接口，则结果是不确定的）：
00000000000000000200000001
01000000000000000300000001
02000000000000000400000001
|ID:00000000000000000200000001    |Element count:5    |Attribute count:1    |Text count:4    |
|ID:01000000000000000300000001    |Element count:9    |Attribute count:2    |Text count:10    |
|ID:02000000000000000400000001    |Element count:18    |Attribute count:9    |Text count:20    |

ANONYMOUS BLOCK EXECUTE
```

- DBE_XMLDOM.GETELEMENTSBYTAGNAM

  XML中按名称查找返回DOMNODELIST的节点。
  DBE_XMLDOM.GETELEMENTSBYTAGNAME的函数原型为：

```
DBE_XMLDOM.GETELEMENTSBYTAGNAME(
  doc        IN      DOMDOCUMENT,
  tagname    IN      VARCHAR2)
RETURN DOMNODELIST;
```

  XML中按名称查找返回DOMNODELIST的节点。
  DBE_XMLDOM.GETELEMENTSBYTAGNAME的函数原型为：

```
DBE_XMLDOM.GETELEMENTSBYTAGNAME(
  elem       IN      DOMELEMENT,
  tagname       IN      VARCHAR2)
RETURN DOMNODELIST;
```

  XML中按名称和命名空间查找返回DOMNODELIST的节点。
  DBE_XMLDOM.GETELEMENTSBYTAGNAME的函数原型为：

```
DBE_XMLDOM.GETELEMENTSBYTAGNAME(
  elem       IN      DOMELEMENT,
  tagname       IN      VARCHAR2,
  ns         IN      VARCHAR2)
 RETURN DOMNODELIST;
```


版权所有 © 华为云计算技术有限公司

表 2-465 DBE_XMLDOM.GETELEMENTSBYTAGNAME 接口参数说明

参数	描述
doc	指定的DOMDOCUMENT节点。
elem	指定的DOMELEMENT节点。
tagname	标签名称。使用通配符（*）将匹配任何标签。
ns	命名空间。使用通配符（*）将匹配任何命名空间。

示例：

```
--1. 在DOMDOCUMENT节点通过TAGNAME匹配查找，返回匹配的DOMNODELIST节点列表。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root_elem DBE_XMLDOM.DOMElement;
  child_node DBE_XMLDOM.DOMNODE;
  node_list DBE_XMLDOM.DOMNODELIST;
  buffer VARCHAR2(1000);
BEGIN
  doc := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0" encoding="UTF-8"?>
<computer size="ITX" price="19999">
    <cpu>Ryzen 9 3950X</cpu>
    <cpu>Ryzen 9 5950X_1</cpu>
    <ram>32GBx2 DDR4 3200MHz<cpu>Ryzen <cpu>Ryzen 9 5950X_2</cpu></cpu></ram>
    <motherboard>ROG X570i</motherboard>
    <gpu>RTX2070 Super</gpu>
    <ssd>1TB NVMe Toshiba + 2TB NVMe WD Black</ssd>
    <hdd>12TB WD Digital</hdd>
    <psu>CORSAIR SF750</psu>
    <case>LIANLI TU150</case>
</computer>');
  root_elem := DBE_XMLDOM.GETDOCUMENTELEMENT(doc);
  node_list := DBE_XMLDOM.GETELEMENTSBYTAGNAME(doc, 'cpu');
  child_node := DBE_XMLDOM.ITEM(node_list, 2);
  DBE_XMLDOM.WRITETOBUFFER(child_node, buffer);
  DBE_OUTPUT.PRINT_LINE(buffer);
END;
/
-- 预期结果为：
<cpu>Ryzen <cpu>Ryzen 9 5950X_2</cpu></cpu>
ANONYMOUS BLOCK EXECUTE

--2. 在DOMELEMENT节点通过TAGNAME匹配查找，返回匹配的DOMNODELIST节点列表。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root_elem DBE_XMLDOM.DOMElement;
  child_node DBE_XMLDOM.DOMNODE;
  node_list DBE_XMLDOM.DOMNODELIST;
  buffer VARCHAR2(1000);
BEGIN
  doc := DBE_XMLDOM.NEWDOMDOCUMENT(
    '<?xml version="1.0" encoding="UTF-8"?>' ||
    '<computer size="ITX" price="19999">' ||
    '<cpu>Ryzen 9 3950X</cpu>' ||
    '<cpu>Ryzen 9 5950X_1</cpu>' ||
    '<ram>32GBx2 DDR4 3200MHz' ||
        '<cpu>Ryzen 9 5950X_2' ||
            '<cpu>Ryzen 9 5950X_3' ||
                '<cpu>Ryzen 9 5950X_4</cpu>' ||
            '</cpu>' ||
```


版权所有 © 华为云计算技术有限公司

```
        '</cpu>' ||
    '</ram>' ||
    '</computer>');
  root_elem := DBE_XMLDOM.GETDOCUMENTELEMENT(doc);
  node_list := DBE_XMLDOM.GETELEMENTSBYTAGNAME(root_elem, 'cpu');
  child_node := DBE_XMLDOM.ITEM(node_list, 3);
  DBE_XMLDOM.WRITETOBUFFER(child_node, buffer);
  DBE_OUTPUT.PRINT_LINE(buffer);
END;
/
-- 预期结果为：
<cpu>Ryzen 9 5950X_3<cpu>Ryzen 9 5950X_4</cpu></cpu>
ANONYMOUS BLOCK EXECUTE

--3. 在DOMELEMENT节点通过TAGNAME以及NAMESPACE匹配查找，返回匹配的DOMNODELIST节点列表。
DECLARE
  doc DBE_XMLDOM.DOMDocument;
  root_elem DBE_XMLDOM.DOMElement;
  child_node DBE_XMLDOM.DOMNODE;
  node_list DBE_XMLDOM.DOMNODELIST;
  buffer VARCHAR2(1000);
BEGIN
  doc := DBE_XMLDOM.NEWDOMDOCUMENT('<?xml version="1.0" encoding="UTF-8"?>
<computer size="ITX" price="19999" xmlns:h="www.xxxxx.com">
    <cpu>Ryzen 9 3950X</cpu>
    <cpu>Ryzen 9 5950X_1</cpu>
    <h:cpu>ns Ryzen 9 5950X_2</h:cpu>
    <ram>32GBx2 DDR4 3200MHz<cpu>Ryzen 9 5950X_3<cpu>Ryzen 9 5950X_4<cpu>Ryzen 9
5950X_5</cpu></cpu></cpu></ram>
    <motherboard>ROG X570i</motherboard>
    <gpu>RTX2070 Super</gpu>
    <ssd>1TB NVMe Toshiba + 2TB NVMe WD Black</ssd>
    <hdd>12TB WD Digital</hdd>
    <psu>CORSAIR SF750</psu>
    <case>LIANLI TU150</case>
</computer>');
  root_elem := DBE_XMLDOM.GETDOCUMENTELEMENT(doc);
  node_list := DBE_XMLDOM.GETELEMENTSBYTAGNAME(root_elem, 'cpu', 'www.xxxxx.com');
  child_node := DBE_XMLDOM.ITEM(node_list, 0);
  DBE_XMLDOM.WRITETOBUFFER(child_node, buffer);
  DBE_OUTPUT.PRINT_LINE(buffer);
END;
/
-- 预期结果为：
<h:cpu>ns Ryzen 9 5950X_2</h:cpu>
ANONYMOUS BLOCK EXECUTE
```

2.12.2.22 DBE_XMLGEN

接口介绍

DBE_XMLGEN系统包将SQL查询的结果转换为规范的XML格式，并将结果返回。支持的所有接口参见**表2-467**。

表 2-466 DBE_XMLGEN 数据类型

类型名称	描述
DBE_XMLGEN.CTXHANDLE	用于存储XML输出状态的数据类型。

版权所有 © 华为云计算技术有限公司

说明

- 在同一个session中，context handle最多只允许存在65535个。关闭context handle并不会回收这个数量。
- 分布式中暂不支持connect by语句，因此不可构造newcontextfromhierarchy函数数据。
- 分布式中不允许cursor反向移动，因此restartquery功能不可使用。
- 输出的XML中表字段、类型与用户创建的表字段与类型大小写一致，如果需要大写字段与类型名需要在创建时使用双引号进行指定。
- NEWCONTEXTFROMHIERARCHY初始化时，能够调用SETNULLHANDLING、USENULLATTRIBUTEINDICATOR、SETCONVERTSPECIALCHARS方法设置，但不生效。
- 在不添加ORDER BY进行查询时，查询结果无序，会影响GETXML函数的返回结果。
- 使用EXECUTE DIRECT ON语法执行DBE_XMLGEN包中的函数时，如果GUC参数enable_udf_execute_direct的值为on，则可以执行；如果其值为off，则不可以执行。

表 2-467 DBE_XMLGEN

接口名称	描述
DBE_XMLGEN.CONVERT	将输入的字符串进行XML编码或解码操作。
DBE_XMLGEN.NEWCONTEXT	初始化普通context handle。
DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY	初始化带有递归元素的context handle。
DBE_XMLGEN.SETCONVERTSPECIALCHARS	设置输出的XML是否需要进行XML编码。
DBE_XMLGEN.SETNULLHANDLING	设置XML中null值如何展示。
DBE_XMLGEN.SETROWSETTAG	设置XML根节点名称。
DBE_XMLGEN.SETROWTAG	设置XML中每一行数据的tag名。
DBE_XMLGEN.USENULLATTRIBUTEINDICATOR	对XML中的null值所在的元素添加xsi:nil="true"属性。
DBE_XMLGEN.USEITEMTAGSFORCOLL	对数组类型变量所在的元素中添加_ITEM后缀。
DBE_XMLGEN.GETNUMROWSPROCESSED	查看上一次GETXML或者GETXMLTYPE返回的数据行数。
DBE_XMLGEN.SETMAXROWS	设置GETXML操作最大的返回行数。
DBE_XMLGEN.SETSKIPROWS	设置跳过SQL行数。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_XMLGEN.RESTARTQUERY	重启SQL。
DBE_XMLGEN.GETXMLTYPE	返回XMLTYPE类型的XML文本。
DBE_XMLGEN.GETXML	返回CLOB类型的XML文本。
DBE_XMLGEN.CLOSECONTEXT	关闭context handle。

- DBE_XMLGEN.CONVERT
  将输入的字符串进行XML编码或解码操作。按照以下规则进行转换操作。

  表 2-468 XML 编码规则

原始值	编码值
&	&
<	<
>	>
"	"
'	'

  DBE_XMLGEN.CONVERT函数原型：

```
DBE_XMLGEN.CONVERT(XMLSTR IN VARCHAR2, FLAG IN NUMBER := 0) RETURNS VARCHAR2;
DBE_XMLGEN.CONVERT(XMLCLOB IN CLOB, FLAG IN NUMBER := 0) RETURNS CLOB;
```

  表 2-469 DBE_XMLGEN.CONVERT 接口参数说明

参数	描述
XMLSTR	需要转换的XML字符串，VARCHAR2类型。
XMLCLOB	需要转换的XML字符串， CLOB类型。
FLAG	转码或解码字符串。 0：编码操作。 1：解码操作。

  示例：

```
-- xml解码
SELECT DBE_XMLGEN.CONVERT('<foo/>', 1);
```


版权所有 © 华为云计算技术有限公司

```
 convert
---------
 <foo/>
(1 row)
-- xml编码
SELECT DBE_XMLGEN.CONVERT('<foo><qwe</foo>', 0);
            convert
--------------------------------
 &lt;foo&gt;&lt;qwe&lt;/foo&gt;
(1 row)
```

- DBE_XMLGEN.NEWCONTEXT

  初始化普通context handle，该context handle只能在一个事务中使用，否则使用**DBE_XMLGEN.RESTARTQUERY**和**DBE_XMLGEN.CLOSECONTEXT**关闭对应context handle时会报错。

  DBE_XMLGEN.NEWCONTEXT函数原型：

  ```
  DBE_XMLGEN.NEWCONTEXT(QUERYSTRING IN VARCHAR2) RETURNS DBE_XMLGEN.CTXHANDLE;
  DBE_XMLGEN.NEWCONTEXT(QUERYSTRING IN SYS_REFCURSOR) RETURNS
  DBE_XMLGEN.CTXHANDLE;
  ```

  表 2-470 DBE_XMLGEN.NEWCONTEXT 接口参数说明

参数	描述
QUERYSTRING	用于生成XML的查询SQL语句或SYS_REFCURSOR。

- DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY

  初始化带有递归元素的context handle。数据格式要求为两列，第一列为NUMERIC类型，第二列为XML或XMLTYPE类型。通常情况下由connect by语句生成，第一列指定生成level。由于分布式不支持connect by语句，因此该函数用例无法构造。

  说明

  生成的xml层级嵌套不允许超过5000万层。

  DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY函数原型：

  ```
  DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY(QUERYSTRING IN VARCHAR2);
  ```

  表 2-471 DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY 接口参数说明

参数	描述
QUERYSTRING	需要转换的XML字符串，VARCHAR2类型。

- DBE_XMLGEN.SETCONVERTSPECIALCHARS

  设置输出的XML是否需要进行XML编码。取消xml编码可能会存在xml注入问题，如果出于性能考虑且可以保证xml是安全的情况下可以不进行xml编码。

  DBE_XMLGEN.SETCONVERTSPECIALCHARS函数原型：

  ```
  DBE_XMLGEN.SETCONVERTSPECIALCHARS(CTX IN DBE_XMLGEN.CTXHANDLE, CONV IN BOOLEAN);
  ```


版权所有 © 华为云计算技术有限公司

表 2-472 DBE_XMLGEN.SETCONVERTSPECIALCHARS 接口参数说明

参数	描述
CTX	context handle。
CONV	是否需要对输出的XML进行XML编码。 ● true：进行XML编码。 ● false：不进行XML编码。

- DBE_XMLGEN.SETNULLHANDLING
  设置XML中null值如何展示。
  DBE_XMLGEN.SETNULLHANDLING函数原型：

```
DBE_XMLGEN.SETNULLHANDLING(CTX IN DBE_XMLGEN.CTXHANDLE, FLAG IN NUMBER := 0);
```

表 2-473 DBE_XMLGEN.SETNULLHANDLING 接口参数说明

参数	描述
CTX	context handle。
FLAG	NULL值展示格式。 ● 0：不展示元素。 ● 1：元素上添加 xsi:nil="true" 属性。 ● 2：展示自闭合元素。

- DBE_XMLGEN.SETROWSETTAG
  设置XML根节点名称。
  DBE_XMLGEN.SETROWSETTAG函数原型：

```
DBE_XMLGEN.SETROWSETTAG(CTX IN DBE_XMLGEN.CTXHANDLE, ROWSETTAGNAME IN VARCHAR2);
```

表 2-474 DBE_XMLGEN.SETROWSETTAG 接口参数说明

参数	描述
CTX	context handle。
ROWSETTAGNAME	XML根节点名称。

- DBE_XMLGEN.SETROWTAG
  设置XML中每一行数据的tag名。
  DBE_XMLGEN.SETROWTAG函数原型：

```
DBE_XMLGEN.SETROWTAG(CTX IN DBE_XMLGEN.CTXHANDLE, ROWTAGNAME IN VARCHAR2);
```


版权所有 © 华为云计算技术有限公司

表 2-475 DBE_XMLGEN.SETROWTAG 接口参数说明

参数	描述
CTX	context handle。
ROWTAGNAME	每一行数据的tag名。

- DBE_XMLGEN.USENULLATTRIBUTEINDICATOR
  对XML中的null值所在的元素添加xsi:nil="true"属性。
  DBE_XMLGEN.USENULLATTRIBUTEINDICATOR函数原型：

```
DBE_XMLGEN.USENULLATTRIBUTEINDICATOR(CTX IN DBE_XMLGEN.CTXHANDLE, ATTRIND IN BOOLEAN);
```

表 2-476 DBE_XMLGEN.USENULLATTRIBUTEINDICATOR 接口参数说明

参数	描述
CTX	context handle。
ATTRIND	无意义。

- DBE_XMLGEN.USEITEMTAGSFORCOLL
  对数组类型变量所在的元素中添加_ITEM后缀。
  DBE_XMLGEN.USEITEMTAGSFORCOLL函数原型：

```
DBE_XMLGEN.USEITEMTAGSFORCOLL(CTX IN DBE_XMLGEN.CTXHANDLE);
```

表 2-477 DBE_XMLGEN.USEITEMTAGSFORCOLL 接口参数说明

参数	描述
CTX	context handle。

示例：

```
-- 预置数据
CREATE TABLE test_for_array(idd number, id INT[]);
INSERT INTO test_for_array VALUES(1, ARRAY[1,2,3]);
SELECT DBE_XMLGEN.GETXML('SELECT * from test_for_array');
-- 数组类型添加_ITEM后缀
DECLARE
qryctx DBE_XMLGEN.CTXHANDLE;
result CLOB;
BEGIN
    qryctx := DBE_XMLGEN.NEWCONTEXT('SELECT id from test_for_array');
    DBE_XMLGEN.useItemTagsForColl(qryctx);
    result:=DBE_XMLGEN.GETXML(qryctx);
    DBE_XMLGEN.CLOSECONTEXT(qryctx);
    DBE_OUTPUT.PUT_LINE(result);
END;
/
<?xml version="1.0"?>
<ROWSET>
<ROW>
  <id>
    <int4_ITEM>1</int4_ITEM>
```


版权所有 © 华为云计算技术有限公司

```
  <int4_ITEM>2</int4_ITEM>
  <int4_ITEM>3</int4_ITEM>
 </id>
</ROW>
</ROWSET>
ANONYMOUS BLOCK EXECUTE
DROP TABLE test_for_array;
```

- DBE_XMLGEN.GETNUMROWSPROCESSED

  查看上一次GETXML或者GETXMLTYPE返回的数据行数。

  DBE_XMLGEN.GETNUMROWSPROCESSED函数原型：

  ```
  DBE_XMLGEN.GETNUMROWSPROCESSED(CTX IN DBE_XMLGEN.CTXHANDLE);
  ```

  表 2-478 DBE_XMLGEN.GETNUMROWSPROCESSED 接口参数说明

参数	描述
CTX	context handle。

- DBE_XMLGEN.SETMAXROWS

  设置每一次GETXML最大的返回行数。

  DBE_XMLGEN.SETMAXROWS函数原型：

  ```
  DBE_XMLGEN.SETMAXROWS(CTX IN DBE_XMLGEN.CTXHANDLE, MAXROWS IN NUMBER);
  ```

  表 2-479 DBE_XMLGEN.SETMAXROWS 接口参数说明

参数	描述
CTX	context handle。
MAXROWS	每一次GETXML最大的返回行数。

- DBE_XMLGEN.SETSKIPROWS

  设置跳过SQL行数。

  DBE_XMLGEN.SETSKIPROWS函数原型：

  ```
  DBE_XMLGEN.SETSKIPROWS(CTX IN DBE_XMLGEN.CTXHANDLE, SKIPROWS IN NUMBER);
  ```

  表 2-480 DBE_XMLGEN.SETSKIPROWS 接口参数说明

参数	描述
CTX	context handle。
SKIPROWS	跳过SQL的头部行数。

- DBE_XMLGEN.RESTARTQUERY

  重启SQL。分布式不支持该接口。

  DBE_XMLGEN.RESTARTQUERY函数原型：

  ```
  DBE_XMLGEN.RESTARTQUERY(CTX IN DBE_XMLGEN.CTXHANDLE);
  ```


版权所有 © 华为云计算技术有限公司

表 2-481 DBE_XMLGEN.RESTARTQUERY 接口参数说明

参数	描述
CTX	context handle。

- DBE_XMLGEN.GETXMLTYPE

  返回XMLTYPE类型的XML文本。

  DBE_XMLGEN.GETXMLTYPE函数原型：

  ```
  DBE_XMLGEN.GETXMLTYPE(SQLQUERY IN VARCHAR2, DTDORSCHEMA IN NUMBER := 0) RETURNS
  XMLTYPE;
  DBE_XMLGEN.GETXMLTYPE(CTX IN DBE_XMLGEN.CTXHANDLE, DTDORSCHEMA IN NUMBER := 0)
  RETURNS XMLTYPE;
  ```

  表 2-482 DBE_XMLGEN.GETXMLTYPE 接口参数说明

参数	描述
SQLQUERY	需要转换成XML的查询SQL。
DTDORSCHEMA	无意义
CTX	context handle。

- DBE_XMLGEN.GETXML

  返回CLOB类型的XML文本。

  DBE_XMLGEN.GETXML函数原型：

  ```
  DBE_XMLGEN.GETXML(SQLQUERY IN VARCHAR2, DTDORSCHEMA IN NUMBER := 0) RETURNS CLOB;
  DBE_XMLGEN.GETXML(CTX IN DBE_XMLGEN.CTXHANDLE, DTDORSCHEMA IN NUMBER := 0)
  RETURNS CLOB;
  DBE_XMLGEN.GETXML(CTX IN DBE_XMLGEN.CTXHANDLE, TMPCLOB INOUT CLOB, DTDORSCHEMA
  IN NUMBER := 0);
  ```

  表 2-483 DBE_XMLGEN.GETXML 接口参数说明

参数	描述
SQLQUERY	需要转换成XML的查询SQL。
DTDORSCHEMA	无意义
CTX	context handle。
TMPCLOB	用于保存输出的XML的CLOB变量。

- DBE_XMLGEN.CLOSECONTEXT

  关闭context handle。

  DBE_XMLGEN.CLOSECONTEXT函数原型：

  ```
  DBE_XMLGEN.CLOSECONTEXT(CTX IN DBE_XMLGEN.CTXHANDLE);
  ```


版权所有 © 华为云计算技术有限公司

表 2-484 DBE_XMLGEN.CLOSECONTEXT 接口参数说明

参数	描述
CTX	context handle。

示例：

```
CREATE TABLE IF NOT EXISTS department(department_id NUMBER, department_name
VARCHAR2(30), manager NUMBER, location NUMBER);
INSERT INTO department VALUES(10, 'administrator', 200, 1700);
INSERT INTO department VALUES(11, 'aaa', 200, 1700);
INSERT INTO department VALUES(12, 'bbb', 300, 1600);
INSERT INTO department VALUES(13, 'ccc', 400, 1600);
INSERT INTO department VALUES(14, 'aaa</row><a>asd</a><row>', 400, 1600);
INSERT INTO department VALUES(15, NULL, 500,1600);
INSERT INTO department VALUES(16, '!@#$%^&*()+-=<>/\"a3_啊', 400, 1600);
-- 关闭context
DECLARE
    qryctx DBE_XMLGEN.CTXHANDLE;
    result CLOB;
BEGIN
    qryctx := DBE_XMLGEN.NEWCONTEXT('SELECT * from department where department_id=14');
    DBE_XMLGEN.SETCONVERTSPECIALCHARS(qryctx, false);
    result:=DBE_XMLGEN.GETXML(qryctx);
    DBE_XMLGEN.CLOSECONTEXT(qryctx);
    DBE_OUTPUT.PUT_LINE(result);
END;
/
<?xml version="1.0"?>
<ROWSET>
<ROW>
  <department_id>14</department_id>
  <department_name>aaa</row><a>asd</a><row></department_name>
  <manager>400</manager>
  <location>1600</location>
</ROW>
</ROWSET>
ANONYMOUS BLOCK EXECUTE
DROP TABLE department;
```

2.12.2.23 DBE_XMLPARSER

接口介绍

DBE_XMLPARSER用于将XML字符串反序列化，将存储XML文档的字符串转换为Document节点。高级包DBE_XMLPARSER支持的接口请参见**表2-485**。

XMLPARSER数据类型用于存储XML数据，其存储的数量上限为16777215。XMLPARSER数据类型能够根据输入的字符串解析并建立DOMDocument节点。高级包还提供相应的SET和GET接口，用于解析过程的约束属性。

说明

- DBE_XMLPARSER高级包在字符集设置为SQL_ASCII的数据库内使用时，传入超出ASCII范围的字符，会导致报错。
- DBE_XMLPARSER高级包只支持ORA兼容模式。

版权所有 © 华为云计算技术有限公司

表 2-485 DBE_XMLPARSER

接口名称	描述
DBE_XMLPARSER.FREEPARSER	释放PARSER。
DBE_XMLPARSER.GETDOCUMENT	获取解析的Document节点。
DBE_XMLPARSER.GETVALIDATIONMODE	获取validate属性。
DBE_XMLPARSER.NEWPARSER	新建PARSER实例。
DBE_XMLPARSER.PARSEBUFFER	解析VARCHAR字符串。
DBE_XMLPARSER.PARSECLOB	解析CLOB字符串。
DBE_XMLPARSER.SETVALIDATIONMODE	设置validate属性。

- DBE_XMLPARSER.FREEPARSER

  释放给定的PARSER对象。

  DBE_XMLPARSER.FREEPARSER的原型为：

  ```
  DBE_XMLPARSER.FREEPARSER (
    p     IN     parser);
  ```

  表 2-486 DBE_XMLPARSER.FREEPARSER 接口参数说明

参数	描述
p	指定的PARSER类型对象。

  示例：

  ```
  -- 新建parser，释放parser。
  DECLARE
    l_parser dbe_xmlparser.parser;
    BEGIN
    l_parser := dbe_xmlparser.newparser();
    -- 直接释放l_parser实例。
    dbe_xmlparser.freeparser(l_parser);
  END;
  /
  ```

- DBE_XMLPARSER.GETDOCUMENT

  DBE_XMLPARSER.GETDOCUMENT返回PARSER构建的DOM树文档的根节点。只有在解析文档后，才能调用此函数。

  DBE_XMLPARSER.GETDOCUMENT的原型为：

版权所有 © 华为云计算技术有限公司

```
DBE_XMLPARSER.GETDOCUMENT (
 p     IN     parser)
 RETURN DOMDocument;
```

表 2-487 DBE_XMLPARSER.GETDOCUMENT 接口参数说明

参数	描述
p	指定的PARSER类型对象。

说明

- GETDOCUMENT函数无传入参数时，报错。
- GETDOCUMENT函数参数PARSER传入为空时，返回null。
- GETDOCUMENT函数传入的PARSER还没有解析文档时，返回null。

示例：

```
-- 新建parser解析字符串，GETDOCUMENT获取文档打印出来。
DECLARE
  l_parser dbe_xmlparser.parser;
  l_doc dbe_xmldom.domdocument;
  buffer varchar2 :=
'<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Do not forget me this weekend!</body>
</note>';
  buffer2 varchar2;
  BEGIN
  l_parser := dbe_xmlparser.newparser();
 -- l_parser解析字符串，通过GETDOCUMENT获取domdocument节点。
  dbe_xmlparser.PARSEBUFFER(l_parser, buffer);
  l_doc := dbe_xmlparser.getdocument(l_parser);
 --将l_doc中的内容打印出来。
  dbe_xmldom.writetobuffer(l_doc, buffer2);
  RAISE NOTICE '%', buffer2;

  dbe_xmlparser.freeparser(l_parser);
  dbe_xmldom.freedocument(l_doc);
END;
/
```

执行结果：

```
NOTICE:  <?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Do not forget me this weekend!</body>
</note>
```

- DBE_XMLPARSER.GETVALIDATIONMODE

  获取给定PARSER的解析验证模式。如果DTD验证开启返回true，否则返回false。

  DBE_XMLPARSER.GETVALIDATIONMODE原型为：

```
DBE_XMLPARSER.GETVALIDATIONMODE (
 p     IN     parser)
 RETURN BOOLEAN;
```


版权所有 © 华为云计算技术有限公司

表 2-488 DBE_XMLPARSER.GETVALIDATIONMODE 接口参数说明

参数	描述
p	指定的PARSER类型对象。

示例：

```
-- 新建parser,通过GETVALIDATIONMODE获取parser解析验证模式是否打开。
DECLARE
  l_parser dbe_xmlparser.parser;
BEGIN
  l_parser := dbe_xmlparser.newparser();
    IF (dbe_xmlparser.GETVALIDATIONMODE(l_parser) = true) THEN
 RAISE NOTICE 'validation';
    ELSE
    RAISE NOTICE 'no validation';
  END IF;
  dbe_xmlparser.freeparser(l_parser);
END;
/
```

执行结果：

```
NOTICE:  validation
```

- DBE_XMLPARSER.NEWPARSER

  新建PARSER对象，返回一个新的解析器实例。

  DBE_XMLPARSER.NEWPARSER原型为：

  ```
  DBE_XMLPARSER.NEWPARSER
   RETURN Parser;
  ```

  示例：

  ```
  -- 新建parser解析字符串，随后释放。
  DECLARE
    -- Create a parser.
    l_parser dbe_xmlparser.parser;
    l_doc dbe_xmldom.domdocument;
    buffer varchar2(1000) :=
      '<?xml version="1.0" encoding="UTF-8"?>
      <note>
      <to>Tove</to>
      <from>Jani</from>
      <heading>Reminder</heading>
      <body>Do not forget me this weekend!</body>
      </note>';
    buffer2 varchar2(1000);
    BEGIN
    l_parser := dbe_xmlparser.newparser();
    -- Parse the document and create a new DOM document.
    dbe_xmlparser.PARSEBUFFER(l_parser, buffer);

    dbe_xmlparser.freeparser(l_parser);
  END;
  /
  ```

- DBE_XMLPARSER.PARSEBUFFER

  PARSEBUFFER解析存储在字符串中的XML文档。

  DBE_XMLPARSER.PARSEBUFFER原型为：

  ```
  DBE_XMLPARSER.PARSEBUFFER (
    p     IN     parser,
    doc   IN VARCHAR2);
  ```


版权所有 © 华为云计算技术有限公司

表 2-489 DBE_XMLPARSER.PARSEBUFFER 接口参数说明

参数	描述
p	指定的PARSER类型对象。
doc	存储XML文档的字符串。

说明

- PARSEBUFFER函数能够解析的字符串最大长度为32767，超过最大长度解析报错。
- 与ORA数据库差异：字符串encoding只支持UTF8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。
- 与ORA数据库DTD校验差异：
  - !ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而ORA数据库不报错。
  - <!ENTITY baidu "www.baidu.com">..... &Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而ORA数据库不报错。
- 与ORA数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而ORA数据库会报错。
- 与ORA数据库xml预定义实体解析差异：'"会被解析转义为字符，而ORA数据库中预定义实体统一都没有转义为字符。

示例：

```
-- 新建parser，PARSEBUFFER解析字符串，获取文档打印出来。
DECLARE
  l_parser dbe_xmlparser.parser;
  l_doc dbe_xmldom.domdocument;
  buffer varchar2 :=
'<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Do not forget me this weekend!</body>
</note>';
  buffer2 varchar2;
  BEGIN
  l_parser := dbe_xmlparser.newparser();
  dbe_xmlparser.PARSEBUFFER(l_parser, buffer);
  l_doc := dbe_xmlparser.getdocument(l_parser);

  dbe_xmldom.writetobuffer(l_doc, buffer2);
  RAISE NOTICE '%', buffer2;

  dbe_xmlparser.freeparser(l_parser);
  dbe_xmldom.freedocument(l_doc);
END;
/
```

执行结果：

```
NOTICE:  <?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Do not forget me this weekend!</body>
</note>
```

- DBE_XMLPARSER.PARSECLOB

  PARSECLOB解析存储在Clob中的XML文档。

版权所有 © 华为云计算技术有限公司

DBE_XMLPARSER.PARSECLOB原型为：

```
DBE_XMLPARSER.PARSECLOB (
  p     IN     parser,
  doc   IN CLOB);
```

表 2-490 DBE_XMLPARSER.PARSECLOB 接口参数说明

参数	描述
p	指定的PARSER类型对象。
doc	存储XML文档的CLOB字符串。

说明

- PARSECLOB不支持解析大于1GB的clob。
- 与ORA数据库差异：字符串encoding只支持UTF8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。
- 与ORA数据库DTD校验差异：
  - !ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而ORA数据库不报错。
  - <!ENTITY baidu "www.baidu.com">..... &Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而ORA数据库不报错。
- 与ORA数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而ORA数据库会报错。
- 与ORA数据库xml预定义实体解析差异：'"会被解析转义为字符，而ORA数据库中预定义实体统一都没有转义为字符。

示例：

```
-- 新建parser，parseclob解析字符串，获取文档打印出来。
DECLARE

 l_clob clob :=
'<?xml version="1.0" encoding="UTF-8"?>
 <note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>this weekend!</body>
 </note>';
 -- Create a parser.
 l_parser dbe_xmlparser.parser;
 l_doc dbe_xmldom.domdocument;
 buffer varchar2(1000);
 BEGIN
 l_parser := dbe_xmlparser.newparser();
 -- Parse the document and create a new DOM document.
 dbe_xmlparser.parseclob(l_parser, l_clob);
 l_doc := dbe_xmlparser.getdocument(l_parser);
 dbe_xmldom.writetobuffer(l_doc, buffer);
 RAISE NOTICE '%',buffer;

 dbe_xmlparser.freeparser(l_parser);
 dbe_xmldom.freedocument(l_doc);

 END;
 /
```

执行结果：

 版权所有 © 华为云计算技术有限公司

```
NOTICE:  <?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>this weekend!</body>
</note>
```

- DBE_XMLPARSER.SETVALIDATIONMODE

  设置给定Parser的解析验证模式。

  DBE_XMLPARSER.SETVALIDATIONMODE原型为：

```
DBE_XMLPARSER.SETVALIDATIONMODE(
  p     IN     parser)
  yes   IN BOOLEAN);
```

表 2-491 DBE_XMLPARSER.SETVALIDATIONMODE 接口参数说明

参数	描述
p	指定的PARSER类型对象
yes	要设置的模式： ● true：开启DTD验证。 ● false：不开启验证。

说明

- SETVALIDATIONMODE函数yes传入为空时，不改变PARSER的解析验证模式。
- PARSER初始化默认为开启DTD验证模式。

示例1：

```
-- 新建parser，设置的待解析xml字符串同DTD格式不匹配。
-- setValidationMode设置为false可以正常解析，设置为true后解析报错。
DECLARE
  l_clob clob :=
 '<!DOCTYPE note [
 <!ELEMENT note (to,from,heading,body)>
 <!ELEMENT to (#PCDATA)>
 <!ELEMENT from (#PCDATA)>
 <!ELEMENT heading (#PCDATA)>
 <!ELEMENT body (#PCDATA)>
 ]>
 <table>
 <name attr1="WEB" attr2="web2">African Coffee Table</name>
 <width>80</width>
 <length>120</length>
 </table>';
  l_parser dbe_xmlparser.parser;
  l_doc dbe_xmldom.domdocument;
 buffer varchar2(1000);
  BEGIN
  l_parser := dbe_xmlparser.newparser();
  -- 设为false，去解析。
    dbe_xmlparser.setValidationMode(l_parser, false);
    dbe_xmlparser.parseclob(l_parser, l_clob);
    l_doc := dbe_xmlparser.getdocument(l_parser);
    dbe_xmldom.writetobuffer(l_doc, buffer);
    RAISE NOTICE '%', buffer;
    dbe_xmlparser.freeparser(l_parser);
    dbe_xmldom.freedocument(l_doc);
```


版权所有 © 华为云计算技术有限公司

```
 END;
 /
```

执行结果：

```
NOTICE:  <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
<!ELEMENT note (to , from , heading , body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<table>
 <name attr1="WEB" attr2="web2">African Coffee Table</name>
 <width>80</width>
 <length>120</length>
 </table>
```

示例2：

```
-- 新建parser，设置的待解析xml字符串同DTD格式不匹配。
-- setValidationMode设置为true后解析报错。
DECLARE
  l_clob clob :=
 '<!DOCTYPE note [
 <!ELEMENT note (to,from,heading,body)>
 <!ELEMENT to (#PCDATA)>
 <!ELEMENT from (#PCDATA)>
 <!ELEMENT heading (#PCDATA)>
 <!ELEMENT body (#PCDATA)>
 ]>
 <table>
 <name attr1="WEB" attr2="web2">African Coffee Table</name>
 <width>80</width>
 <length>120</length>
 </table>';
  l_parser dbe_xmlparser.parser;
  l_doc dbe_xmldom.domdocument;
 buffer varchar2(1000);
  BEGIN
 l_parser := dbe_xmlparser.newparser();
 -- 设为true，去解析。
 -- xml字符串不符合DTD格式，预期将报错。
   dbe_xmlparser.setValidationMode(l_parser, true);
   dbe_xmlparser.parseclob(l_parser, l_clob);
   l_doc := dbe_xmlparser.getdocument(l_parser);
   dbe_xmldom.writetobuffer(l_doc, buffer);
   dbe_xmlparser.freeparser(l_parser);
   dbe_xmldom.freedocument(l_doc);
 END;
 /
```

执行结果：

```
xmlparser解析报错
ERROR:  invalid XML document
```

2.12.2.24 PRVT_ILM

PRVT_ILM接口为ILM特性内部使用接口，用户无法直接调用。本章节仅列举接口名称，不提供详细的接口原型描述。

接口名称	描述
be_active_ado_window	维护窗口触发动作，刷新ilmadowindow触发时间及动作。

版权所有 © 华为云计算技术有限公司

接口名称	描述
be_create_ado_window_for_each_db	维护窗口触发动作，在实例的每个数据库中创建ilmadowindow。
be_execute_ilm	ilmadowindow执行动作，进行自动评估。
be_execute_ilm_dn	用于分布式DN上的自动评估。该操作会影响集群整体的ILM后台调度。
flush_task_executestate	自动调度任务刷新task状态。
generate_taskoid	CN生成用于本次调度的taskid。
evaluate_obj_policy	触发调度后对数据对象进行评估。
change_be_ilm	DBE_ILM_ADMIN.ENABLE_ILM()和DISABLE_ILM()帮助函数。该操作会影响集群整体的ILM后台调度。
get_compression_ratio	DBE_COMPRESSION.GET_COMPRESSION_RATIO()帮助函数。
get_compression_type	DBE_COMPRESSION.GET_COMPRESSION_TYPE()帮助函数。
get_lastmodified_time	DBE_HEAT_MAP.ROW_HEAT_MAP()帮助函数。
ilm_job_action	压缩任务执行体，分布式DN使用。
delete_expired_ilm_log	清理过期日志，分布式使用。
delete_gs_ilm_task	操作特性相关系统表，分布式DN使用。
insert_gs_ilm_jobdetail	操作特性相关系统表，分布式DN使用。
insert_gs_ilm_task	操作特性相关系统表，分布式DN使用。
insert_gs_ilm_taskdetail	操作特性相关系统表，分布式DN使用。
update_gs_ilm_jobdetail	操作特性相关系统表，分布式DN使用。
update_gs_ilm_object	操作特性相关系统表，分布式DN使用。
update_gs_ilm_param	操作特性相关系统表，分布式DN使用。该操作会影响集群整体的ILM后台调度。
update_gs_ilm_task	操作特性相关系统表，分布式DN使用。
gs_ilm_ticker	维护窗口执行动作，执行一次打点，记录LSN与时间的映射关系。
compress_blocks	PRVT_ILM.ILM_JOB_ACTION()帮助函数。
compress_block_single	废弃接口，请勿使用。
get_job_status	DBE_ILM.STOP_ILM帮助函数。
insert_gs_ilm_param	操作特性相关系统表。

版权所有 © 华为云计算技术有限公司

接口名称	描述
query_unfinishedjob_num	查询是否存在还没有结束的job，包括初始和运行中的状态。
update_gs_ilm_object_flag2	操作特性相关系统表，分布式DN使用。
ilm_seq_nextval	用于获取ILM的下一个sequence值。
ilm_seq_setval	用于设置ILM的当前sequence值。
check_compatibility	用于统一判断数据库兼容性模式的帮助函数。
set_pg_settings	用于设置数据库兼容性模式特性GUC参数的帮助函数。
check_seq	保证GSILMTASK_SEQ当前值大于GS_ILM_TASK表最大taskoid，分布式CN使用。
get_actual_compression_ratio	DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO()帮助函数。
get__hole_ratio	DBE_COMPRESSION.GET_HOLE_RATIO()帮助函数。
turbo_compress_buffer_stats	DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS()帮助函数。
get_index_compression_type	DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE()帮助函数。
create_ilm_db_policy	DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY()帮助函数。
delete_ilm_db_policy	DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY()帮助函数。
compress_indexes	PRVT_ILM.ILM_JOB_ACTION()帮助函数。
eval_comp_relgrowth	废弃接口，请勿使用。
delete_gs_ilm_jobdetail	操作特性相关系统表，分布式DN使用。

2.12.3 内部函数

以下接口为各高级包内部接口，不建议用户直接调用，否则可能出现异常行为或非预期运行效果。

DBE_DESCRIBE

DBE_DESCRIBE包含的内部接口如**表2-492**所示。

版权所有 © 华为云计算技术有限公司

表 2-492 高级功能包 DBE_DESCRIBE 内部接口功能说明

接口名称	描述
DBE_DESCRIBE.GET_PROCEDURE_NAME	从用户的原始输入中提取出存储过程/函数名称。
DBE_DESCRIBE.IS_NUMBER_TYPE	判断类型是否为数值类型。

DBE_UTILITY

DBE_UTILITY包含的内部接口如**表2-493**所示。

表 2-493 高级功能包 DBE_UTILITY 包内部接口功能说明

接口名称	描述
DBE_UTILITY.EXPAND_SQL_TEXT	将包含视图或同义词的SQL语句展开为仅引用基表和联接的等价查询。
DBE_UTILITY.CANONICALIZE_RET	对数据库对象名（表、视图、列、用户等）进行标准化格式化。
DBE_UTILITY.COMMA_TO_TABLE_FUNC	将逗号分隔的字符串列表转换为PL/SQL索引表，并返回元素个数。
DBE_UTILITY.COMPILE_SCHEMA	推荐使用pkg_util.gs_compile_schema。批量编译指定模式下的存储过程、函数、视图等无效或全部PL/SQL对象。
DBE_UTILITY.NAME_SEPARATE	用于按照数据库对象命名规则，对对象名称进行结构化拆分，解析出模式名、对象名及数据库链（dblink）等组成部分。
DBE_UTILITY.NAME_TOKENIZE_FUNC	将数据库对象名称解析为五个部分（模式名、表名、列名、总长度）。
DBE_UTILITY.NAME_TOKENIZE_LOWER	将数据库对象名称解析为五个部分（模式名、表名、列名、总长度），并将所有部分转换为小写。
DBE_UTILITY.NAME_TOKENIZE_LOWER_FUNC	功能同DBE_UTILITY.NAME_TOKENIZE_LOWER。
DBE_UTILITY.PRIVILEGE_CHECK	检查当前用户对指定数据库对象的权限。
DBE_UTILITY.SEARCH_CLASS_WITH_NSPOID_ONAME_TYPE	在指定命名空间中按名称和类型搜索数据库对象，返回该对象oid及数量。
DBE_UTILITY.SEARCH_OBJECTS	根据模式名、对象名称和对象类型，在数据库中搜索匹配的数据库对象，并返回对象的oid、对象类型以及是否为同义词。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_UTILITY.SEARCH_OBJECTS_SYNONYM_FILL_SCHEMA	根据指定的搜索模式类型，在多个可能的模式中搜索指定名称和类型的数据库对象，返回找到对象的信息。
DBE_UTILITY.SEARCH_PROCEDURE_WITH_NSPOID_ONAME	用于检索指定模式oid和名称对应的函数、存储过程及包。
DBE_UTILITY.SEARCH_SYNONYM_WITH_NSPOID_ONAME	用于检索指定模式oid和名称对应的同义词。
DBE_UTILITY.TABLE_TO_COMMA_FUNC	用于将数组转化为字符串。
DBE_UTILITY.USER_NAME	用于返回当前数据库用户名。
DBE_UTILITY.SEARCH_OBJECTS_SYNONYM_FILL_SECHEMA	与DBE_UTILITY.SEARCH_OBJECTS_SYNONYM_FILL_SCHEMA功能相同，建议使用DBE_UTILITY.SEARCH_OBJECTS_SYNONYM_FILL_SCHEMA。
DBE_UTILITY.SEARCH_SYNONM_WITH_NSPOID_ONAME	与DBE_UTILITY.SEARCH_SYNONYM_WITH_NSPOID_ONAME功能相同，建议使用DBE_UTILITY.SEARCH_SYNONYM_WITH_NSPOID_ONAME。

2.13 Retry 管理

Retry是数据库在SQL或存储过程（包含匿名块）执行失败时，在数据库内部进行重新执行的过程，以提高执行成功率和用户体验。同时也是数据库内部通过检查发生错误时的错误码及Retry相关配置，决定是否进行重试。

失败时回滚之前执行的语句，并重新执行存储过程进行Retry。

示例：

```
gaussdb=# CREATE OR REPLACE PROCEDURE retry_basic ( IN  x INT)
AS
BEGIN
   INSERT INTO t1 (a) VALUES (x);
   INSERT INTO t1 (a) VALUES (x+1);
END;
/

gaussdb=# CALL retry_basic(1);
```

2.14 PACKAGE

PACKAGE是一组相关存储过程、函数、变量、常量和游标等PL/SQL程序的组合，具有面向对象的特点，可以对PL/SQL程序设计元素进行封装。PACKAGE中的函数具有统一性，创建、删除、修改都统一进行。

版权所有 © 华为云计算技术有限公司

PACKAGE包含包头（Package Specification）和Package Body两个部分，其中包头所包含的声明可以被外部函数、匿名块等访问，而在包体中包含的声明不能被外部函数、匿名块等访问，只能被包体内函数和存储过程等访问。

PACKAGE的创建请参见**CREATE PACKAGE**。

须知

- 跨PACKAGE变量不支持作为FOR循环中控制变量使用。
- PACKAGE中定义类型不支持删除、修改等操作，也不支持定义表。
- 不支持以SCHEMA.PACKAGE.CURSOR的形式引用cursor变量。
- 带参数的CURSOR仅支持在当前PACKAGE内打开。
- 在分布式下，仅支持在ORA兼容性数据库中使用PACKAGE。
- 在分布式下，当SQL语句中含有package内的变量、package内的函数时，不支持下推。
- 不支持package变量作为函数或存储过程参数的默认值。
- 如果PACKAGE内的存储过程入参包含伪类型（如：anyelement），不支持调用时传入变长类型的参数。
- 包头、包体的函数in/out/inout参数不一致时，系统将识别为不同的函数。

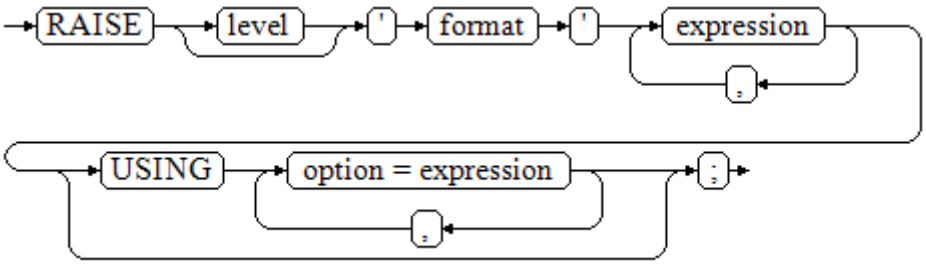
2.15 调试

语法

RAISE有以下五种语法格式：

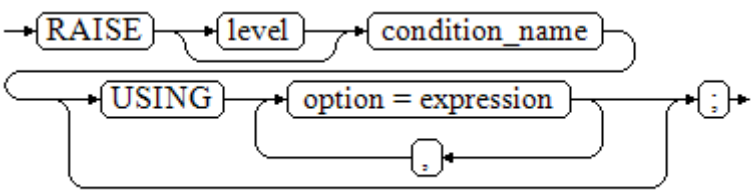
- raise_format::=语法如**图2-51**所示。

  图 2-51 raise_format::=

- raise_condition::=语法如**图2-52**所示。

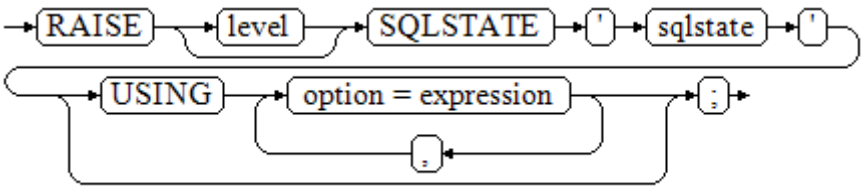
  图 2-52 raise_condition::=

版权所有 © 华为云计算技术有限公司

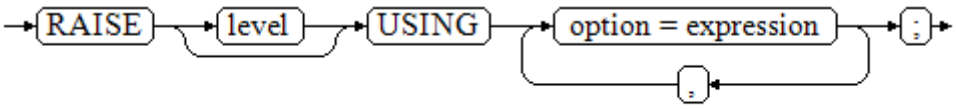
- raise_sqlstate::=语法如图2-53所示。

  图 2-53 raise_sqlstate::=

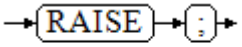
- raise_option::=语法如图2-54所示。

  图 2-54 raise_option::=

- raise::=语法如图2-55所示。

  图 2-55 raise::=

参数说明：

- level选项用于指定错误级别，有DEBUG、LOG、INFO、NOTICE、WARNING以及EXCEPTION（默认值）。EXCEPTION抛出一个正常终止当前事务的异常，其他的仅产生不同异常级别的信息。特殊级别的错误信息是否报告到客户端、写到服务器日志由log_min_messages和client_min_messages这两个GUC参数控制。
- format：格式字符串，指定要报告的错误消息文本。格式字符串后可跟表达式，用于向消息文本中插入。在格式字符串中，%由format后面跟着的参数的值替换，%%用于打印出%。例如：

  ```
  --v_job_id 将替换字符串中的 %：
  RAISE NOTICE 'Calling cs_create_job(%)',v_job_id;
  ```

- option = expression：向错误报告中添加另外的信息。关键字option可以是MESSAGE、DETAIL、HINT以及ERRCODE，并且每一个expression可以是任意的字符串。
  - MESSAGE，指定错误消息文本，这个选项不能用于在USING前包含一个格式字符串的RAISE语句中。
  - DETAIL，说明错误的详细信息。
  - HINT，用于打印出提示信息。
  - ERRCODE，向报告中指定错误码（SQLSTATE）。可以使用条件名称或者直接用五位字符的SQLSTATE错误码。
- condition_name：错误码对应的条件名。
- sqlstate：错误码。

版权所有 © 华为云计算技术有限公司

如果在RAISE EXCEPTION命令中既没有指定条件名也没有指定SQLSTATE，默认用RAISE EXCEPTION (P0001)。如果没有指定消息文本，默认用条件名或者SQLSTATE作为消息文本。

须知

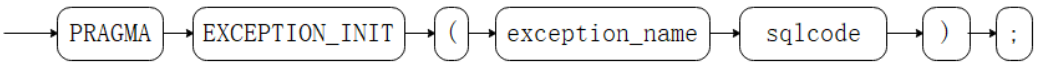
- 当由SQLSTATE指定了错误码，则不局限于已定义的错误码，可以选择任意包含五个数字或者大写的ASCII字母的错误码，而不是00000。建议避免使用以三个0结尾的错误码，因为这种错误码是类别码，会被整个种类捕获。
- 兼容ORA模式下，SQLCODE等于SQLSTATE。
- 图2-55所示的语法不接受任何参数。这种形式仅用于一个BEGIN块中的EXCEPTION语句，它使得错误重新被处理。

EXCEPTION_INIT语法

兼容ORA模式下，支持使用EXCEPTION_INIT语法自定义错误码SQLCODE。

语法图如图2-56所示。

图 2-56 exception_init::=



参数说明：

- exception_name为用户声明的异常名，EXCEPTION_INIT语法必须出现在与声明异常相同部分，位于声明异常之后。
- sqlcode为自定义的SQLCODE，必须为负整数，取值范围-2147483647~-1。

注意事项：

- 支持在ORA兼容性下使用，不支持与系统错误码进行绑定。
- exception关联的sqlcode只能通过exception_init赋值，其他赋值方式无效。
- 不支持跨database.schema.package.exception调用。

须知

- 使用EXCEPTION_INIT语法自定义错误码SQLCODE时，SQLSTATE与SQLCODE相同，SQLERRM格式为" xxx: non-GaussDB Exception"。比如自定义SQLCODE=-1，则SQLSTATE="-1"，SQLERRM=" 1: non-GaussDB Exception"。
- 在同一个存储过程或函数中，通过EXCEPTION_INIT设置相同错误码时，Oracle数据库不支持分别作为条件捕获异常，GaussDB支持此行为，详见示例。

示例

终止事务时，给出错误和提示信息：

版权所有 © 华为云计算技术有限公司

```
CREATE OR REPLACE PROCEDURE proc_raise1(user_id in integer)
AS
BEGIN
RAISE EXCEPTION 'Noexistent ID --> %',user_id USING HINT = 'Please check your user ID';
END;
/

call proc_raise1(300011);

--执行结果
ERROR:  Noexistent ID --> 300011
HINT:  Please check your user ID
```

两种设置SQLSTATE的方式：

```
CREATE OR REPLACE PROCEDURE proc_raise2(user_id in integer)
AS
BEGIN
RAISE 'Duplicate user ID: %',user_id USING ERRCODE = 'unique_violation';
END;
/

\set VERBOSITY verbose
call proc_raise2(300011);

--执行结果
ERROR:  Duplicate user ID: 300011
SQLSTATE: 23505
```

如果主要的参数是条件名或者是SQLSTATE，可以使用：

```
RAISE division_by_zero;
RAISE SQLSTATE '22012';
```

例如：

```
CREATE OR REPLACE PROCEDURE division(div in integer, dividend in integer)
AS
DECLARE
res int;
    BEGIN
    IF dividend=0 THEN
        RAISE division_by_zero;
        RETURN;
    ELSE
        res := div/dividend;
        RAISE INFO 'division result: %', res;
        RETURN;
    END IF;
    END;
/
call division(3,0);

--执行结果
ERROR:  division_by_zero
```

或者另一种方式：

```
RAISE unique_violation USING MESSAGE = 'Duplicate user ID: ' || user_id;
```

在同一个存储过程或函数中，通过EXCEPTION_INIT设置相同错误码时，Oracle数据库不支持分别作为条件捕获异常，GaussDB支持此行为。

```
gaussdb=# CREATE OR REPLACE PACKAGE pkg IS
gaussdb$#   err1 EXCEPTION;
gaussdb$#   err2 EXCEPTION;
gaussdb$#   PRAGMA EXCEPTION_INIT(err1, -10001);
gaussdb$#   PRAGMA EXCEPTION_INIT(err2, -10001);
gaussdb$# PROCEDURE pro;
gaussdb$# END pkg;
gaussdb$# /
```


版权所有 © 华为云计算技术有限公司

```
CREATE PACKAGE
gaussdb=# CREATE OR REPLACE PACKAGE BODY pkg IS
gaussdb$# PROCEDURE pro IS
gaussdb$# BEGIN
gaussdb$#     FOR i IN 1..10 LOOP
gaussdb$#         BEGIN
gaussdb$#         IF i = 1 THEN
gaussdb$#             raise err1;
gaussdb$#         END IF;
gaussdb$#         IF i = 2 THEN
gaussdb$#             raise err2;
gaussdb$#         END IF;
gaussdb$#         EXCEPTION
gaussdb$# -- 相同错误码拆分作为EXCEPTION的两个不同条件。
gaussdb$#             WHEN err1 THEN
gaussdb$#                 CONTINUE;
gaussdb$#             WHEN err2 THEN
gaussdb$#                 EXIT;
gaussdb$#         END;
gaussdb$#     END LOOP;
gaussdb$# END pro;
gaussdb$# END pkg;
gaussdb$# /
CREATE PACKAGE BODY
gaussdb=# CALL pkg.pro();
 pro
-----

(1 row)
gaussdb=# DROP PACKAGE pkg;
NOTICE:  drop cascades to function public.pro()
DROP PACKAGE
```

2.16 关键字

说明

关键字是PL/SQL中具有特殊含义的标识符，不区分大小写，**表2-494**中描述的是PL/SQL域的关键字。需要注意的是，PL/SQL保留关键字和非保留关键字对比SQL关键字并不完全一致，有关SQL关键字的详细信息请参见**关键字**。

表 2-494 PL/SQL 关键字

关键字	关键字类型	是否是SQL关键字
ABSOLUTE	非保留	非保留
ALIAS	非保留	-
ALL	保留	保留
ALTER	非保留	非保留
ARRAY	非保留	保留
AS	非保留	保留
BACKWARD	非保留	非保留
BULK	非保留	-
BEGIN	保留	非保留

版权所有 © 华为云计算技术有限公司

关键字	关键字类型	是否是SQL关键字
BY	保留	非保留
CASE	保留	保留
CLOSE	保留	非保留
COLLATE	保留	保留
COLLECT	非保留	-
COMMIT	非保留	非保留
CONSTANT	非保留	非保留
CONTINUE	非保留	非保留
CURSOR	非保留	非保留
CURRENT	非保留	非保留
DEBUG	非保留	-
DECLARE	保留	非保留
DEFAULT	保留	保留
DELETE	保留	非保留
DETAIL	非保留	-
DIAGNOSTICS	保留	-
DISTINCT	非保留	保留
DUMP	非保留	-
ELSE	保留	保留关键字
ELSEIF	保留	-
ELSIF	保留	-
END	保留	保留关键字
ERRCODE	非保留	-
ERROR	非保留	非保留
EXCEPT	非保留	保留
EXCEPTION	保留	-
EXCEPTIONS	非保留	-
EXECUTE	保留	非保留
EXIT	保留	-
FETCH	保留	保留

版权所有 © 华为云计算技术有限公司

关键字	关键字类型	是否是SQL关键字
FIRST	非保留	非保留
FOR	保留	保留
FORALL	保留	-
FOREACH	保留	-
FORWARD	非保留	非保留
FROM	保留	保留
FUNCTION	保留	非保留
GET	保留	-
GOTO	保留	-
HINT	非保留	-
IF	保留	非保留
IGNORE	非保留	非保留
IMMEDIATE	非保留	非保留
IN	保留	保留
INDEX	非保留	非保留
INFO	非保留	-
INSERT	保留	非保留
INSTANTIATION	非保留	-
INTERSECT	非保留	保留
INTO	保留	保留
IS	非保留	保留
LAST	非保留	非保留
LIMIT	保留	保留
LOAD	非保留	非保留
LOG	非保留	非保留
LOOP	保留	非保留
MERGE	非保留	非保留
MESSAGE	非保留	-
MESSAGE_TEXT	非保留	-
MOVE	保留	非保留

版权所有 © 华为云计算技术有限公司

关键字	关键字类型	是否是SQL关键字
MULTISET	非保留	-
NEXT	非保留	非保留
NO	非保留	非保留
NOT	保留	保留
NOTICE	非保留	-
NULL	保留	保留
OF	保留	非保留
OPEN	保留	非保留
OPTION	非保留	非保留
OR	保留	保留
OUT	保留	非保留（不能是函数或类型）
PACKAGE	非保留	非保留
PERFORM	非保留	-
PG_EXCEPTION_CONTEXT	非保留	-
PG_EXCEPTION_DETAIL	非保留	-
PG_EXCEPTION_HINT	非保留	-
PIPE	非保留	-
PIPELINED	非保留	-
PRAGMA	非保留	-
PRIOR	非保留	非保留
PROCEDURE	保留	保留关键字
QUERY	非保留	非保留
RAISE	保留	-
RANGE	非保留	非保留
RECORD	非保留	-
REF	保留	非保留
RELATIVE	非保留	非保留
RELEASE	非保留	非保留

版权所有 © 华为云计算技术有限公司

关键字	关键字类型	是否是SQL关键字
REPLACE	非保留	非保留
RESULT_OID	非保留	-
RETURN	保留	非保留
RETURNED_SQLSTATE	非保留	-
REVERSE	非保留	-
ROLLBACK	非保留	非保留
ROW_COUNT	非保留	-
ROWTYPE	非保留	非保留
SAVE	非保留	-
SAVEPOINT	非保留	非保留
SCROLL	非保留	非保留
SELECT	保留	保留
SLICE	非保留	非保留
SQLSTATE	非保留	-
STACKED	非保留	-
STRICT	保留	非保留
SUBTYPE	非保留	-
SYS_REFCURSOR	非保留	非保留
TABLE	非保留	保留
THEN	保留	保留
TO	保留	保留
UNION	非保留	保留
UPDATE	保留	非保留
USING	保留	保留
USE_COLUMN	非保留	-
USE_VARIABLE	非保留	-
VARIABLE_CONFLICT	非保留	-
VARRAY	非保留	-
WARNING	非保留	-

版权所有 © 华为云计算技术有限公司

关键字	关键字类型	是否是SQL关键字
WHEN	保留	保留
WHILE	保留	-
WITH	非保留	保留

版权所有 © 华为云计算技术有限公司

3 兼容性说明

3.1 GaussDB 数据库兼容性概述

在GaussDB中，创建数据库可以选择不同的兼容性模式，如ORA、MYSQL、TD、PG、M，分别代表Oracle兼容性、MySQL兼容性MYSQL模式、TD兼容性、Postgres兼容性和MySQL兼容性M-Compatibility模式。不同兼容性模式会直接影响SQL语法、数据类型、系统函数、存储过程等行为，部分兼容性接口可能仅在相应的兼容性模式下支持。

Oracle兼容性、MySQL兼容性M-Compatibility模式、MySQL兼容性MYSQL模式的兼容性说明可参考**Oracle兼容性说明**、**MySQL兼容性M-Compatibility模式**、**MySQL兼容性MYSQL模式**等章节内容，用户可根据实际需要，选择合适的兼容性。

3.2 Oracle 兼容性说明

3.2.1 Oracle 数据库兼容性概述

GaussDB数据库在基本功能（数据类型、SQL、数据库对象等）和PL/SQL方面与Oracle数据库基本兼容。但是由于架构设计方面的差异，还是存在一些不兼容的项。

本章节主要介绍GaussDB数据库的Oracle兼容模式与Oracle数据库19c版本的兼容性对比信息。

3.2.2 SQL 的基本元素

版权所有 © 华为云计算技术有限公司

3.2.2.1 数据类型

表 3-1 数值类型

Oracle数据库	GaussDB数据库	差异
NUMBER [(p [, s])]	支持，有差异	精度和用法存在差异。 ● NUMBER带参数时，GaussDB的精度p与标度s的最大边界值比Oracle更大。 ● NUMBER不带参数时，GaussDB的精度p的默认值远大于带参数时的最大边界值；而在Oracle中，精度p的默认值等于带参数时的最大边界值。 ● GaussDB不支持标度s为负值；在Oracle中，标度s为负值时会精确到相应的整数位。 ● GaussDB支持插入NaN值；在Oracle中，插入NaN值会报错。 ● GaussDB中未指定标度s时，不限制标度；Oracle中，未指定标度s时，默认标度为0。 ● GaussDB中不支持number(*, s)的写法；Oracle支持该写法，*表示取最大精度（即不限制精度）。
FLOAT [(p)]	支持	-
BINARY_FLOAT	不支持	-
BINARY_DOUBLE	支持	-

表 3-2 日期时间类型

Oracle数据库	GaussDB数据库	差异
DATE	支持，有差异	GaussDB在未开启mapping_date_to_datea参数时精度有差异，GaussDB支持的公元时间相较于Oracle范围更大；GaussDB在开启mapping_date_to_datea参数时，GaussDB支持的语法、操作符、格式控制与Oracle数据库中DATE类型有差异。
TIMESTAMP [(fractional_seconds_precision)]	支持，有差异	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE	支持，有差异	GaussDB的timestamptz等价于Oracle的timestampwithlocaltimezone，缺少Oracle对应的timestamptz类型。 时区更新：部分国家或地区经常会更新时区信息， 数据库系统也因此常常需要同步修改时区文件以确保时间内容的正确性。 GaussDB时区类型目前只涉及timestamp with timezone，当新的时区文件生效时，不会对已有的数据进行变更，新数据会随时区文件信息进行同步调整。与Oracle的同类型数据能力存在差异。
TIMESTAMP [(fractional_seconds_precision)] WITH LOCAL TIME ZONE	不支持	-
INTERVAL YEAR [(year_precision)] TO MONTH	支持	-
INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]	支持	-

版权所有 © 华为云计算技术有限公司

说明

- ORA兼容模式下，GaussDB中日历系统和Oracle日历系统存在以下差异，包括：
  - Oracle中存在两套日历系统：儒略历和格里历（公历）。公元1582年10月5日前使用儒略历，闰年计算规则是满足年份能被4整除；公元1582年10月5日后使用格里历，闰年计算规则为满足年份能被4整除且不能被100整除，或者能被400整除。GaussDB中只有格里历（公历）日历系统，闰年计算规则为满足年份能被4整除且不能被100整除，或者能被400整除。
  - Oracle中公元1582年包含355天，缺少的10天为公元1582年10月5号到公元1582年10月14号，公元1582年10月4号的下一天是公元1582年10月15号；GaussDB中公元1582年包含365天。
  - Oracle存在公元0年，即'-0001-01-01 00:00:00'到'0001-01-01 00:00:00'包含731天；GaussDB中不存在公元0年，即'-0001-01-01 00:00:00'到'0001-01-01 00:00:00'包含366天。
- ORA兼容模式下，在未开启mapping_date_to_datea参数时DATE类型被替换为TIMESTAMP(0) WITHOUT TIME ZONE，差异同TIMESTAMP(0) WITHOUT TIME ZONE。当向DATE类型字段插入带小数点的timestamp数值时，如timestamp '2023-01-01 23:59:59.5'，GaussDB会四舍五入，插入表中的结果为'2023-01-02 00:00:00'，Oracle会截断，插入表中的结果为'2023-01-01 23:59:59'。
- ORA兼容模式下，在开启mapping_date_to_datea参数时DATE类型被替换为DATEA，差异同DATEA。
- 对于TIMESTAMP [（fractional_seconds_precision）] / TIMESTAMP [（fractional_seconds_precision）] WITH TIME ZONE与Oracle存在以下差异。
  - fractional_seconds_precision：GaussDB支持的精度范围为0~6，Oracle支持的精度范围为0~9。
  - GaussDB通过DateStyle设置日期和时间值的显示格式，以及有歧义的输入值的解析规则。具体请参见**日期/时间类型**中日期输入的说明介绍。Oracle在一般情况下的输入格式校验及输出显示由NLS_TIMESTAMP_FORMAT/NLS_TIMESTAMP_TZ_FORMAT参数进行控制。
  - 对于秒的小数部分的显示，GaussDB默认去除末尾的零，Oracle依据格式化参数中的设置（FF/FF1-FF9）进行显示控制。如 '2017-09-01 10:32:19.212000'，GaussDB显示为'2017-09-01 10:32:19.212'，Oracle在format参数中含FF时显示为'2017-09-01 10:32:19.212000'，在format参数中含FF9时显示为'2017-09-01 10:32:19.212000000'。
  - 支持的时间范围存在差异，GaussDB支持的公元时间较Oracle范围更大。
  - 对于超过6位精度的字面量会首先按照第7位精度遵循四舍六入五成双的规则做转换，即第7位为5的时候，第6位将取最接近的偶数。6位精度内的舍入规则以时间2000-01-01 00:00:00为基线，基线前时间五舍六入，基线后时间四舍五入，其中TIMESTAMP WITH TIME ZONE类型的时间基线按照UTC-0时区计算。
- ORA兼容模式下，在开启mapping_date_to_datea参数时DATE与Oracle的DATE类型存在以下差异。
  - 支持字面量定义DATEA类型，如“SELECT datea '2024-10-01';”，与Oracle存在差异，GaussDB中的字面量字符串受nls_date_format校验，若不符合nls_date_format值则会报错或输出错误数据，而Oracle中字面量格式校验则受到固定格式YYYY-MM-DD的校验。
  - 在使用CREATE TABLE、CREATE VIEW、PREPARE创建对象时，GaussDB与Oracle存在差异。

    GaussDB中，如果存在未明确类型（UNKNOWN）时，会对UNKNOWN类型向明确类型进行一次转换，如果转换失败，则创建对象报错，如果转换成功则会将该值保存为确定类型值，并且之后的操作不再受额外参数的影响。例如，有如下的典型场景：

    ```
    WHERE datea_column > 'unknown_text'
    'unknown_text'::datea
    cast('unknown_text' as datea)
    datea'unknown text'
    ```


版权所有 © 华为云计算技术有限公司

而如果原始定义中存在函数等其他表达式时，则使用宽松校验。定义对象时，只要该表达式返回值类型能够转换成目标类型，则可以创建成功，不需要在定义时执行该表达式。例如：

```
CREATE TABLE t (a DATE DEFAULT to_date('2024-10-01'));
```

上述语句在创建表时不会受nls_date_format格式校验。

- 当UNKNOWN、CHAR、CHARACTER、NCHAR、BPCHAR、VARCHAR、CHARACTER VARYING、VARCHAR2、NVARCHAR2、TEXT、CLOB类型格式符合数值格式，GaussDB支持DATEA加以上类型（左右操作数可交换）和DATEA减以上类型，不支持以上类型减DATEA类型；当CHAR、CHARACTER、NCHAR、VARCHAR、CHARACTER VARYING、VARCHAR2、NVARCHAR2类型格式符合数值格式，Oracle支持DATE类型加以上类型（左右操作数可交换）和DATE类型减以上类型，不支持以上类型减DATE类型。
- 当UNKNOWN、CHAR、CHARACTER、NCHAR、BPCHAR、VARCHAR、CHARACTER VARYING、VARCHAR2、NVARCHAR2、TEXT、CLOB类型格式符合时间格式，GaussDB支持以上类型减DATEA类型，不支持DATEA加以上类型（左右操作数可交换）和DATEA减以上类型；当CHAR、CHARACTER、NCHAR、VARCHAR、CHARACTER VARYING、VARCHAR2、NVARCHAR2类型格式符合时间格式，Oracle不支持DATE类型加以上类型（左右操作数可交换），不支持DATE类型减以上类型和以上类型减DATE类型。
- GaussDB中DATEA类型格式控制只支持nls_date_format参数，Oracle中除了nls_date_format参数还支持nls_date_language参数。
- GaussDB中DATEA类型和CLOB类型之间支持互相隐式转换；Oracle在某些场景下不支持。
- Oracle数据库在日期时间类型与数值类型之间进行算术运算（+、-）时，会将日期时间类型操作数隐式转换为DATE类型，运算结果也为DATE类型。GaussDB遵循自身的类型提升规则，运算结果类型与左操作数类型保持一致，不进行向DATEA类型的隐式降级转换。
- GaussDB中支持DATEA类型和INTERVAL类型、数值类型、RAW等类型之间的比较操作，Oracle不支持，具体请参见**比较操作符**。
- GaussDB中支持的nls_date_format元素与Oracle存在一些行为差异，包括：
  1. GaussDB不支持E/EE，Oracle支持。
  2. DL/DS/TS在GaussDB中不能用于to_date，Oracle支持用于to_date。
  3. 不支持输入的字符串带后置空格，如nls_date_format = 'yyyy-mm-dd'; 在将下面字符串转为date类型时会报错：“SELECT date '2024 -10-01' FROM dual;”，Oracle支持该写法。
  4. 在Oracle中对于"text"分隔符，需要校验输入，但是在当前GaussDB中，只影响输出，不影响输入。
  5. FX和FM与Oracle行为存在差异，GaussDB中的FM功能和Oracle一样，都用来转为字符串时抑制前导0和后置空格，但GaussDB中其只能作为前缀，且作用范围只是在每一个基础元素上，与Oracle中的作用范围有差异；FX功能与Oracle中一致，都是用于精确校验是否符合nls格式，但GaussDB中FX作用范围为只要FX出现1次，后面的所有元素都需要精确校验，直到遇到FM为止，与Oracle中的作用范围有差异。
  6. GaussDB支持TH作为后缀；Oracle实际不起作用。
  7. GaussDB不支持THSP、SPTH作为后缀，SP可以设置为后缀，但实际不起作用。
  8. GaussDB中前缀、后缀不允许在一个格式化模型元素中连续多次出现，如FMFMYYYY、YYYYTHTH。
- GaussDB中的DATEA类型仅支持btree/ubtree/psort/cbtree索引，不支持gin/ugin/gist索引。
- GaussDB中的DATEA类型仅支持JDBC、ODBC、libpq驱动，不支持GO、Python驱动。
- GaussDB中的DATEA类型不支持作为范围类型使用。
- GaussDB中的DATEA类型的算术操作符（包括DATEA相减、DATEA与数值类型的加减运算等）内部使用float8（双精度浮点数）进行中间计算，再转换为结果类型输出。由于float8的有效精度约为15位十进制数字，当运算结果的小数部分位数较多时，可能出

版权所有 © 华为云计算技术有限公司

现末尾精度损失。例如DATEA相减返回numeric类型时，有效位数最多约15位；DATEA与数值加减返回DATEA类型时，结果中的时分秒部分可能出现微秒级精度偏差。Oracle中DATE类型的算术运算返回NUMBER类型，不存在此精度限制。如需高精度日期计算，建议使用EXTRACT等函数分别提取时间元素后计算。

- GaussDB的gsql端不按照nls_date_format格式显示DATEA数据，而是按照固定格式'YYYY-MM-DD HH24:MI:SS AD/BC'显示。Oracle的sqlplus端按照nls_date_format格式显示DATE数据。

- ORA兼容模式下，支持AT TIME ZONE语法，但是语义与Oracle不一致。在GaussDB中，AT TIME ZONE是类型转换操作符。timestamp AT TIME ZONE zone将不带时区的timestamp按照指定时区解释为UTC时间点，返回timestamptz类型（显示时转换为当前会话时区）；timestamptz AT TIME ZONE zone将带时区的时间戳转换为指定时区的本地时间，返回timestamp类型。但在Oracle中，AT TIME ZONE是时间语义重解释操作符，保持返回类型不变（TIMESTAMP WITH TIME ZONE输入仍返回TIMESTAMP WITH TIME ZONE），仅将时间值按目标时区重新呈现。
- ORA兼容模式下，在使用CREATE TABLE创建对象时，GaussDB与Oracle存在差异：

  GaussDB不支持REF关键字，Oracle支持。

```
CREATE TABLE test_floor_ref (
   ref_id NUMBER,
   col_ref REF obj_ref_test_floor
);
```

表 3-3 字符类型

Oracle数据库	GaussDB数据库	差异
VARCHAR2 [(size [BYTE | CHAR])]	支持，有差异	在GaussDB中，size的单位为字节，最大值为10M。 在Oracle中，当MAX_STRING_SIZE=EXTENDED时，最大长度为32767字节；当MAX_STRING_SIZE=STANDARD时，最大长度为4000字节。
NVARCHAR2 (size)	支持，有差异	在GaussDB中，size的单位为字节，最大值为10M，实际能容纳的字符数与使用的字符集有关；在Oracle中，当MAX_STRING_SIZE=EXTENDED时，最大长度为32767字节，当MAX_STRING_SIZE=STANDARD时，最大长度为4000字节，实际能容纳的字符数与使用的字符集有关。 在GaussDB中，NVARCHAR2结尾的空格在插入时会被忽略；在Oracle中，NVARCHAR2结尾的空格在插入时不会被忽略。
CHAR [(size [BYTE | CHAR])]	支持，有差异	在GaussDB中，size的单位为字节，最大值为10M；在Oracle中，最大长度为2000字节。
NCHAR [(size)]	支持，有差异	在GaussDB中，size单位为字节，最大10MB，实际能容纳的字符数与使用的字符集有关；而在Oracle中，NCHAR是用于存储固定长度Unicode字符的数据类型，size单位为字符，最大容量为2000个字节。
CLOB	支持，有差异	不支持定位器。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
NCLOB	不支持	-
LONG	不支持	-

表 3-4 二进制类型

Oracle数据库	GaussDB数据库	差异
RAW（size）	支持，有差异	在GaussDB中，size是指字节长度建议值，不会用于校验输入raw类型的字节长度。
LONG RAW	不支持	-
BLOB	支持	-
BFILE	不支持	-

表 3-5 ROWID 类型

Oracle数据库	GaussDB数据库	差异
ROWID	支持，有差异	在GaussDB中，ROWID类型是按照node_slice、table_oid、bucket_id、row_no字段数值依次比较；在Oracle中，ROWID类型是按照data object number、relate file number、block number、row number字段数值依次比较。ROWID类型操作符比较规则参见**操作符**。
UROWID	不支持	-

表 3-6 用户自定义类型

Oracle数据库	GaussDB数据库
对象类型	不支持
REF数据类型	不支持
可变数组	支持
嵌套表	支持

版权所有 © 华为云计算技术有限公司

表 3-7 伪类型

Oracle数据库	GaussDB数据库
anytype	不支持
anydata	不支持
anydataset	不支持

表 3-8 xml 类型

Oracle数据库	GaussDB数据库	差异
XMLTYPE	支持，有差异	GaussDB中不支持部分操作，如不能通过使用XMLELEMENT函数将字符串转变为XMLTYPE类型，而是转变成XML类型。具体请参见**XMLTYPE类型**。
URITYPE	不支持	-

表 3-9 空间类型

Oracle数据库	GaussDB数据库
SDO_GEOMETRY	不支持
SDO_TOPO_GEOMETRY	不支持
SDO_GEORASTER	不支持

表 3-10 锁模式

Oracle数据库	GaussDB数据库
none	-
null	AccessShare
RS	RowShare
RX	RowExclusive
S	ShareUpdateExclusive
SRX	Share
-	ShareRowExclusive
X	Exclusive
-	AccessExclusive

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
-	INVALID **说明** GaussDB的INVALID表示被赋予了非法锁。仅在运行过程中出现了GaussDB无法识别的锁时会被赋予INVALID锁。

3.2.2.2 数据类型比较规则

数据类型比较（排序）规则是指相同数据类型的值之间发生比较（排序）时遵循的比较（排序）规则。

表 3-11 比较规则

Oracle数据库	GaussDB数据库	差异
Numeric值	支持	-
日期时间值	支持	-
二进制值	支持	-
字符值	支持，有差异	• 在GaussDB和Oracle中，支持的比较规则不完全相同，相同比较规则的名称也可能不同。 • GaussDB和Oracle的比较规则在可指定性上有差异，例如GaussDB不支持指定表级别的比较规则，而Oracle支持。 • GaussDB和Oracle在指定比较规则的语法上有差异，例如在GaussDB中，使用ENCODING、LC_CTYPE和LC_COLLATE三个参数决定创建数据库时使用的字符集、字符分类和比较规则，具体请参见**CREATE DATABASE**。而在Oracle中，各级别的比较规则通常由一系列带有NLS前缀的参数确定。
对象值	不支持	-
Varrays和嵌套表	支持，有差异	GaussDB和Oracle均支持Varrays的比较，与Oracle不同的是，GaussDB不仅支持比较两个Varrays中的元素个数，还支持同类型的Varrays之间的比较。
数据类型优先级	支持	-
数据转换（显式/隐式类型转换）	支持	-

版权所有 © 华为云计算技术有限公司

3.2.2.3 字面量

表 3-12 字面量

Oracle数据库	GaussDB数据库
文本字面量	支持。
数值字面量	支持，存在如下差异： 不支持123.4f、123.4d（带f/d后缀）的数值字面量作为函数入参。
日期时间字面量	支持。
区间字面量	支持。

3.2.2.4 格式模型

表 3-13 格式

Oracle数据库	GaussDB数据库	差异
数字格式	支持，有差异	GaussDB仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，支持$、C、TM、TM9、TME、U格式。同时在该参数下，不支持TH、PL、SG格式。 GaussDB具体支持情况请参见**类型转换函数**章节的“number类型fmt参数表”。
日期时间格式	支持，有差异	GaussDB中用于时间截断和时间四舍五入的参数，仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 GaussDB具体支持情况请参见**时间和日期处理函数和操作符**章节的“用于日期/时间格式化的模式”。
格式模型修饰符	支持	-
字符串到日期转换规则	支持，有差异	GaussDB中to_timestamp_tz函数在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 GaussDB具体支持情况请参见**类型转换函数**章节的“to_date/to_timestamp/to_timestamp_tz”。
xml格式模式	不支持	-

版权所有 © 华为云计算技术有限公司

3.2.2.5 空值

表 3-14 空值

Oracle数据库	GaussDB数据库
IS NULL和IS NOT NULL	支持
NULLS in conditions	支持

3.2.2.6 注释

表 3-15 comment

Oracle数据库	GaussDB数据库	差异
斜杠和星号（/*）	支持	-
两个连字符（--）	支持	-
COMMENT命令	支持	-
HINT	支持，有差异	GaussDB不支持'--+'hint形式。 具体信息请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。

3.2.2.7 数据库对象

表 3-16 schema 对象

Oracle数据库	GaussDB数据库	差异
分析视图	不支持	-
属性维度	不支持	-
集群	支持	-
约束	支持	-
数据库链接	支持	-
数据库触发器	支持	-
尺寸	支持	-
外部过程库	不支持	-
分层结构	不支持	-
索引组织表	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
索引	支持	Oracle数据库的唯一索引为组合列时会认为NULL == NULL，而GaussDB默认遵守SQL标准，认为NULL != NULL。如需兼容这一行为，可以开启GUC参数enable_uniq_idx_a_compat进行兼容，该参数在GaussDB ORA兼容模式下默认关闭。
索引类型	不支持	-
Java类	不支持	-
Java资源	不支持	-
Java源码	不支持	-
join groups	不支持	-
物化视图	支持	-
物化视图日志	不支持	-
挖掘模型	不支持	-
对象表	不支持	-
对象类型	不支持	-
对象视图	不支持	-
operators	支持	-
包	支持	-
序列	支持	-
存储函数	支持	-
存储过程	支持	-
同义词	支持，有差异	Oracle数据库的数据库对象在同一namespace内不能重名。GaussDB同一namespace内同义词可与表、视图、函数、package重名，如果重名，会优先访问该名称对象，未寻找到该名称对象时才会寻找该名称的同义词指向的对象；同义词指向的对象的Schema名是用户名时才会搜索PUBLIC同义词。详细的搜索顺序参考**CREATE SYNONYM**中的注意事项，同义词的搜索过程。
表	支持	-
视图	支持	-
zone map	不支持	-

版权所有 © 华为云计算技术有限公司

表 3-17 nonschema 对象

Oracle数据库	GaussDB数据库
上下文	不支持
目录	支持
版本	不支持
闪回存档	不支持
锁定配置文件	不支持
配置文件	不支持
还原点	支持
角色	支持
回滚段	• ustore支持回滚段 • astore不支持回滚段
表空间	支持
表空间集	不支持
统一审计策略	支持
用户	支持

3.2.2.8 数据库对象名称和限定符

表 3-18 命名规则

Oracle数据库	GaussDB数据库	差异
数据库对象命名规则	支持，有差异	GaussDB默认小写。
模式对象命名规则	支持	-

3.2.2.9 SQL 语句中的引用架构对象和部件的语法

表 3-19 对象引用

Oracle数据库	GaussDB数据库
引用对象的一般语法	支持
解析对象的引用	支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
引用外部模式对象	支持
引用外部数据库对象	支持
引用表和索引分区和子分区	支持

3.2.3 伪列

分层查询伪列

表 3-20 分层查询伪列

Oracle数据库	GaussDB数据库
connect_by_iscycle	支持
connect_by_isleaf	支持
level伪列	支持

序列伪列

表 3-21 序列

Oracle数据库	GaussDB数据库	差异
currval	支持，有差异	GaussDB以函数形式实现。兼容Oracle调用方式。
nextval	支持，有差异	GaussDB以函数形式实现。兼容Oracle调用方式。

rowid 伪列

表 3-22 rowid

Oracle数据库	GaussDB数据库	差异
rowid	支持，有差异	Oracle在创建表时支持rowid伪列；GaussDB在创建表时使用hasrowid选项支持rowid伪列。

版权所有 © 华为云计算技术有限公司

rownum 伪列

表 3-23 rownum

Oracle数据库	GaussDB数据库	差异
rownum	支持，有差异	Oracle在left、right、full join的条件中使用rownum进行过滤时，不同的条件下表现不尽相同，可能存在忽略或部分忽略rownum条件的现象，而GaussDB在此情况下则表现为对left、right、full join后的结果进行过滤。

xmldata 伪列

表 3-24 xmldata

Oracle数据库	GaussDB数据库
xmldata	不支持

3.2.4 操作符

GaussDB数据库兼容除分层查询以外的运算符。

SQL 运算符

表 3-25 SQL 运算符

Oracle数据库	GaussDB数据库
一元运算符和二元运算符	支持
运算符优先级	支持

算术运算符

表 3-26 算术运算符

Oracle数据库	GaussDB数据库
正负（+-）一元运算符	支持
加减（+-）二元运算符	支持
乘除（*/）二元运算符	支持

版权所有 © 华为云计算技术有限公司

COLLATE 运算符

表 3-27 COLLATE 运算符

Oracle数据库	GaussDB数据库
COLLATE collation_name	支持

连接运算符

表 3-28 连接运算符

Oracle数据库	GaussDB数据库	差异
||	支持，有差异。	对于非法字符的拼接，GaussDB不会检查字符的合法性，直接拼接所有字符；而Oracle若检测到非法字符，会忽略该字符。 示例： -- GaussDB gaussdb=# SET a_format_version='10c'; SET gaussdb=# SET a_format_dev_version='s2'; SET gaussdb=# SELECT dump(CHR(239) || 'test') FROM dual; dump ---------- Typ=25 Len=5: 239,116,101,115,116 (1 row) -- Oracle SQL> SELECT dump(CHR(239) || 'test') FROM dual; dump(CHR(239)||'TEST') ---------- Typ=1 Len=4: 116,101,115,116

分层查询运算符

表 3-29 分层查询运算符

Oracle数据库	GaussDB数据库	差异
prior	支持，有差异。	GaussDB中仅支持对普通列调用，不支持对函数等调用。
connect_by_root	支持，有差异。	GaussDB中，connect_by_root调用时，用括号修饰操作值时，行为与Oracle一致；不使用括号时，仅支持对普通列调用此运算符。

版权所有 © 华为云计算技术有限公司

集合运算符

表 3-30 集合运算符

Oracle数据库	GaussDB数据库
union	支持
union all	支持
intersect	支持
minus	支持

多集合运算符

表 3-31 多集合运算符

Oracle数据库	GaussDB数据库
multiset except	支持
multiset intersect	支持
multiset union	支持

用户自定义运算符

表 3-32 用户自定义运算符

Oracle数据库	GaussDB数据库
CREATE OPERATOR	不支持

比较操作符

Oracle数据库	GaussDB数据库
< =	支持
< >	支持
> =	支持
^ =	支持
! =	不支持，!=中间存在空格时，!会被识别为阶乘。

版权所有 © 华为云计算技术有限公司

当比较操作符（<=、<>、>=、^=）中间存在空格时，也可以识别成没有空格进行正常操作。!=中间存在空格时，!会被识别为阶乘，可能会导致结果与预期不一致。

3.2.5 表达式

GaussDB数据库兼容大部分数据库表达式。

表 3-33 表达式

Oracle数据库	GaussDB数据库	差异
简单表达式	支持	-
分析视图表达式	不支持	-
复合表达式	支持	-
case表达式	支持	-
列表示式	支持	-
cursor表达式	不支持	-
日期时间表达式	支持，有差异	GaussDB的输出结果中不会带时区信息，Oracle会带有类似PM AMERICA/LOS_ANGELES的时区信息。
函数表达式	支持	-
区间表达式	部分支持	GaussDB支持形如SELECT INTERVAL '999999999 23:59:59.999' day(9) to second FROM DUAL;的语句，而不支持形如SELECT(SYSDATE- SYSDATE) DAY TO SECOND FROM DUAL;的语句。Oracle均支持。
json对象访问表达式	部分支持，有差异	• GaussDB支持通过“->'key'”的方式从JSON对象中提取value，而Oracle支持通过“.key”方式来提取value。 • 对于JSON数组对象而言，Oracle支持通过“.key”方式一次性提取所有key对应的value，但GaussDB目前不支持。
模型表达式	不支持	-
对象表达式	不支持	-
占位符表达式	部分支持	对于形如“:var”的一般占位符表达式，GaussDB支持，但不支持通过INDICATOR关键字将两个一般占位符表达式结合起来。
标量子查询表达式	支持	-
类型构造器表达式	部分支持	GaussDB不支持在类型构造器前指定New关键字，而Oracle支持。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
表达式list	支持	-

版权所有 © 华为云计算技术有限公司

<table>
<tr><th>Oracle数据库</th><th>GaussDB数据库</th><th>差异</th></tr>
<tr><td>含(+)操作符表达式</td><td>部分支持，有差异</td><td>对于使用OR操作符连接的外连接表达式，GaussDB仅支持OR操作符左右两侧均为同一单表的外连接表达式。
在(+)和OR操作符混用的场景下，GaussDB与Oracle的行为存在差异，以下是与Oracle 19c的对比说明：
公共语句
-- 执行以下建表语句并插入数据
CREATE TABLE a(id int, b int);
CREATE TABLE a2(id int, c int);
INSERT INTO a VALUES(1,1);
INSERT INTO a VALUES (null, 11);
INSERT INTO a VALUES (2, null);
INSERT INTO a VALUES (123,123);
INSERT INTO a VALUES (5, 500);
INSERT INTO a VALUES (null, 600);
INSERT INTO a2 VALUES(1,2);
INSERT INTO a2 VALUES (3,null);
INSERT INTO a2 VALUES (null, 22);
INSERT INTO a2 VALUES (234,234);
INSERT INTO a2 VALUES (5, 700);
INSERT INTO a2 VALUES (null, 800);
● 示例一
SELECT * FROM a,a2 WHERE (a.id(+) = 1 AND a.id(+) <= 2) OR a2.id(+) = 1 ORDER BY 1,2,3,4;
– GaussDB
gaussdb=# CREATE DATABASE gaussdb_o DBCOMPATIBILITY = 'ORA';
gaussdb=# \c gaussdb_o
-- 执行公共语句后再执行查询语句
gaussdb_o=# SELECT * FROM a,a2 WHERE (a.id(+) = 1 AND a.id(+) <= 2) OR a2.id(+) = 1 ORDER BY 1,2,3,4;
ERROR: Operator "(+)" is not allowed used with "OR" together
-- 切换至默认数据库，并删除gaussdb_o数据库（需按照实际情况修改数据库名）
gaussdb_o=# \c postgres
gaussdb=# DROP DATABASE gaussdb_o;
– Oracle
-- 执行公共语句后再执行查询语句
SQL> SELECT * FROM a,a2 WHERE (a.id(+) = 1 AND a.id(+) <= 2) OR a2.id(+) = 1 ORDER BY 1,2,3,4;

ID| B| ID| C
----------|----------|----------|----------
1| 1| 1| 2
1| 1| 3|
1| 1| 5| 700
1| 1| 234| 234
1| 1| | 22
1| 1| | 800
2| | 1| 2
5| 500| 1| 2
123| 123| 1| 2
| 11| 1| 2
| 600| 1| 2

11 rows selected.</td></tr>
</table>

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

		-- 清理本次示例新建的数据表（需按照实际情况修改表名）
		SQL> DROP TABLE a;
		SQL> DROP TABLE a2;
		GaussDB因不满足OR操作符左右两侧为同一单表的外连接表达式的限制，报错；Oracle不报错。
		• 示例二
		SELECT * FROM a,a2 WHERE a.id(+) = a2.id AND a.id(+) != 2 OR a.id(+) IS NULL AND a.id(+) = a2.id ORDER BY 1,2,3,4;
		– GaussDB
		gaussdb=# CREATE DATABASE gaussdb_o DBCOMPATIBILITY = 'ORA';
		gaussdb=# \c gaussdb_o
		-- 执行公共语句后再执行查询语句
		gaussdb_o=# SELECT * FROM a,a2 WHERE a.id(+) = a2.id OR a.id IS NULL AND a.id(+) = a2.id ORDER BY 1,2,3,4;
		id | b | id | c
		----+-----+-----+-----
		1 | 1 | 1 | 2
		5 | 500 | 5 | 700
		| | 3 |
		| | 234 | 234
		| | | 22
		| | | 800
		(6 rows)
		-- 提取公共表达式后，执行转化后的SQL语句
		gaussdb_o=# SELECT * FROM a,a2 WHERE a.id(+) = a2.id AND (a.id(+) != 2 OR a.id(+) IS NULL) ORDER BY 1,2,3,4;
		id | b | id | c
		----+-----+-----+-----
		1 | 1 | 1 | 2
		5 | 500 | 5 | 700
		| | 3 |
		| | 234 | 234
		| | | 22
		| | | 800
		(6 rows)
		-- 切换至默认数据库，并删除gaussdb_o数据库（需按照实际情况修改数据库名）
		gaussdb_o=# \c postgres
		gaussdb=# DROP DATABASE gaussdb_o;
		– Oracle
		-- 执行公共语句后再执行查询语句
		SQL> SELECT * FROM a,a2 WHERE a.id(+) = a2.id OR a.id IS NULL AND a.id(+) = a2.id ORDER BY 1,2,3,4;
		ERROR at line 1:
		ORA-01719: outer join operator (+) not allowed in operand of OR or IN
		-- 提取公共表达式后，执行转化后的SQL语句
		SQL> SELECT * FROM a,a2 WHERE a.id(+) = a2.id AND (a.id(+) != 2 OR a.id(+) IS NULL) ORDER BY 1,2,3,4;
		
		ID| B| ID| C
		----------|----------|----------|----------
		1| 1| 1| 2
		5| 500| 5| 700

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
		| | 3| | | 234| 234 | | | 22 | | | 800 6 rows selected. -- 清理本次示例新建的数据表（需按照实际情况修改表名） SQL> DROP TABLE a; SQL> DROP TABLE a2; GaussDB因满足OR操作符左右两侧均为同一单表的外连接表达式，因此不报错；Oracle则直接报错（ORA-01719）。 通过提取公共表达式a.id(+) = a2.id，转化后的SQL语句为 SELECT * FROM a,a2 WHERE a.id(+) = a2.id AND (a.id(+) != 2 OR a.id(+) IS NULL) ORDER BY 1,2,3,4; 在两种数据库中执行结果一致。

3.2.6 条件

本章节描述常见的条件兼容项，包含比较条件、浮点条件、逻辑条件、模型条件、多集合条件、模式匹配条件、NULL值条件、XML条件、SQL/JSON条件、复合条件、BETWEEN条件、EXISTS条件、IN条件、IS OF TYPE条件，如表3-34所示。

表 3-34 条件

Oracle数据库	GaussDB数据库	差异
比较条件	支持，有差异	语句中存在ANY、SOME、ALL操作符时存在差异，Oracle支持对list对象进行操作，而GaussDB中需要将list对象转换成数组表达式的形式后再进行操作。
浮点条件	不支持	-
逻辑条件	支持	-
模型条件	不支持	-
多集合条件	不支持	-
模式匹配条件	支持	-
NULL值条件	支持	-
XML条件	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
SQL/JSON条件	部分支持，有差异	• GaussDB不支持IS JSON条件和JSON_TEXTCONTAINS条件。 • GaussDB中JSONB_EQ条件等同于Oracle中JSON_EQUAL条件，但GaussDB不支持ERROR子句。 • GaussDB中JSONB_EXISTS条件等同于Oracle中JSON_EXISTS条件，但GaussDB不支持ERROR子句、EMPTY子句和PASSING子句。
复合条件	支持	-
BETWEEN条件	支持	-
EXISTS条件	支持	-
IN条件	支持	-
IS OF TYPE条件	不支持	-

3.2.7 驱动

3.2.7.1 JDBC

3.2.7.1.1 Array

本章节描述在JDBC驱动使用java.sql.Array类型时，Oracle与GaussDB的差异。

版权所有 © 华为云计算技术有限公司

表 3-35 构造方法参考

构造方式	Oracle数据库	GaussDB数据库	接口差异
1. ArrayDescriptor的静态构造方法构造ArrayDescriptor对象； 2. 通过ArrayDescriptor构造Array对象。	String typeName = "XXX"; Connection conn = getConnection(); Object[] elements = null; ArrayDescriptor desc = ArrayDescriptor.createDescriptor(typeName, conn); Array array = new ARRAY(desc, conn, elements);	String typeName = "xxx"; Connection conn = getConnection(); Object[] elements = null; ArrayDescriptor desc = ArrayDescriptor.getDescriptor(typeName, conn); Array array = new GaussArray(desc, elements);	• 构造ArrayDescriptor的静态方法名不一致，Oracle的方法名为createDescriptor，GaussDB的方法名为getDescriptor。 • Array的构造方法名定义不一致，Oracle为ARRAY(ArrayDescriptor, Connection, Object)，GaussDB为GaussArray(ArrayDescriptor, Object)。 • 变量说明： typeName为类型名称，且大小写敏感，通常Oracle的类型名大写，GaussDB的类型名小写。 conn为对应数据库的连接对象。 elements为对应类型的元素数据。

说明

1. 对于上述表格中未列举出的构造方式，GaussDB暂不支持。
2. 若元素类型为字符类型，构造传入的字符串长度大于元素类型定义的长度，Oracle会在绑定入参时报错；

   若数组类型为varray类型时，若元素个数大于varray类型的长度上限，Oracle会在绑定入参时报错；

   GaussDB在构造或者绑定入参时不会对类型修饰符进行校验，在数据库接收Array对象并执行SQL时，由数据库决定是否报错。

版权所有 © 华为云计算技术有限公司

表 3-36 接口参考

方法名	返回值类型	throws	GaussDB数据库
getBaseTypeName()	String	SQLException	支持
getBaseType()	int	SQLException	支持
getArray()	Object	SQLException	支持
getArray(java.util.Map<String,Class<?>> map)	Object	SQLException	不支持
getArray(long index, int count)	Object	SQLException	支持
getArray(long index, int count, java.util.Map<String,Class<?>> map)	Object	SQLException	不支持
getResultSet()	ResultSet	SQLException	不支持
getResultSet(java.util.Map<String,Class<?>> map)	ResultSet	SQLException	不支持
getResultSet(long index, int count)	ResultSet	SQLException	不支持
getResultSet (long index, int count, java.util.Map<String,Class<?>> map)	ResultSet	SQLException	不支持
free()	void	SQLException	不支持

表 3-37 getArray()接口详细差异

元素数据库类型	getArray接口实际返回值类型（Oracle）	getArray接口实际返回值类型（GaussDB）
CHAR	java.lang.String[]	java.lang.String[]
VARCHAR/ VARCHAR2	java.lang.String[]	java.lang.String[]
NCHAR	java.lang.String[]	java.lang.String[]
NVARCHAR2	java.lang.String[]	java.lang.String[]
NUMBER	java.math.BigDecimal[]	java.math.BigDecimal[]

版权所有 © 华为云计算技术有限公司

元素数据库类型	getArray接口实际返回值类型（Oracle）	getArray接口实际返回值类型（GaussDB）
NUMERIC	java.math.BigDecimal[]	java.math.BigDecimal[]
DECIMAL	java.math.BigDecimal[]	java.math.BigDecimal[]
INTEGER	java.math.BigDecimal[]	java.lang.Integer[]
SMALLINT	java.math.BigDecimal[]	java.lang.Short[]
DOUBLE PRECISION	java.math.BigDecimal[]	java.lang.Double[]
FLOAT	java.math.BigDecimal[]	java.lang.Double[]
REAL	java.math.BigDecimal[]	java.lang.Float[]
BINARY_DOUBLE	java.lang.Double[]	java.lang.Double[]
BINARY_INTEGER	java.math.BigDecimal[]	java.lang.Integer[]
BOOLEAN	java.math.BigDecimal[]	java.lang.Boolean[]
TIMESTAMP	java.sql.Timestamp[]	java.sql.Timestamp[]
TIMESTAMP WITH TIME ZONE	java.time.OffsetDateTime[]	java.sql.Timestamp[]
BLOB	oracle.sql.BLOB[]	java.sql.Blob[]
CLOB	oracle.sql.CLOB[]	java.sql.Clob[]
集合类型/数组类型	java.lang.Object[]	java.sql.Array[]
RECORD类型	java.lang.Object[]	java.sql.Struct[]

说明

1. 对于上述表格中未列举出的类型，GaussDB暂不支持。
2. getArray(long index, int count)接口返回值差异也参考上述表格。
3. getArray接口(long index, int count)，参数index的差异如下：
   - Oracle支持范围为[1, Long.MAX_VALUE]；GaussDB支持范围为[1, Integer.MAX_VALUE]；
   - Oracle在index > Integer.MAX_VALUE时index会被截断；GaussDB在index > Integer.MAX_VALUE时报错。

表 3-38 getBaseType()接口详细差异

元素数据库类型	getBaseType接口返回值（Oracle）	getBaseType接口返回值（GaussDB）
CHAR	java.sql.Types.CHAR	java.sql.Types.CHAR

版权所有 © 华为云计算技术有限公司

元素数据库类型	getBaseType接口返回值（Oracle）	getBaseType接口返回值（GaussDB）
VARCHAR/ VARCHAR2	java.sql.Types.VARCHAR	java.sql.Types.VARCHAR
NCHAR	java.sql.Types.NCHAR	java.sql.Types.CHAR
NVARCHAR2	java.sql.Types.NVARCHAR	java.sql.Types.VARCHAR
NUMBER	java.sql.Types.NUMERIC	java.sql.Types.NUMERIC
NUMERIC	java.sql.Types.DECIMAL	java.sql.Types.NUMERIC
DECIMAL	java.sql.Types.DECIMAL	java.sql.Types.NUMERIC
INTEGER	java.sql.Types.NUMERIC	java.sql.Types.INTEGER
SMALLINT	java.sql.Types.NUMERIC	java.sql.Types.SMALLINT
DOUBLE PRECISION	java.sql.Types.FLOAT	java.sql.Types.DOUBLE
FLOAT	java.sql.Types.FLOAT	java.sql.Types.DOUBLE
REAL	java.sql.Types.FLOAT	java.sql.Types.REAL
BINARY_DOUBLE	oracle.jdbc.OracleTypes.BINARY_DOUBLE	java.sql.Types.DOUBLE
BINARY_INTEGER	java.sql.Types.NUMERIC	java.sql.Types.INTEGER
BOOLEAN	java.sql.Types.NUMERIC	java.sql.Types.BIT
TIMESTAMP	java.sql.Types.TIMESTAMP	java.sql.Types.TIMESTAMP
TIMESTAMP WITH TIME ZONE	oracle.jdbc.OracleTypes.TIMESTAMPTZ	java.sql.Types.TIMESTAMP
BLOB	java.sql.Types.BLOB	java.sql.Types.BLOB
CLOB	java.sql.Types.CLOB	java.sql.Types.CLOB
集合类型/数组类型	java.sql.Types.ARRAY	java.sql.Types.ARRAY
RECORD类型	java.sql.Types.STRUCT	java.sql.Types.STRUCT

说明

对于上述表格中未列举出的类型，GaussDB暂不支持。

版权所有 © 华为云计算技术有限公司

表 3-39 getBaseTypeName()接口详细差异

元素数据库类型	getBaseTypeName接口返回值（Oracle）	getBaseTypeName接口返回值（GaussDB）
CHAR	"CHAR"	"bpchar"
VARCHAR/ VARCHAR2	"VARCHAR"	"varchar"
NCHAR	"NCHAR"	"bpchar"
NVARCHAR2	"NVARCHAR"	"nvarchar2"
NUMBER	"NUMBER"	"numeric"
NUMERIC	"DECIMAL"	"numeric"
DECIMAL	"DECIMAL"	"numeric"
INTEGER	"NUMBER"	"int4"
SMALLINT	"NUMBER"	"int2"
DOUBLE PRECISION	"FLOAT"	"float8"
FLOAT	"FLOAT"	"float8"
REAL	"FLOAT"	"float4"
BINARY_DOUBLE	"BINARY_DOUBLE"	"float8"
BINARY_INTEGER	"NUMBER"	"int4"
BOOLEAN	"NUMBER"	"bool"
TIMESTAMP	"TIMESTAMP"	"timestamp"
TIMESTAMP WITH TIME ZONE	"TIMESTAMP WITH TIME ZONE"	"timestamptz"
BLOB	"BLOB"	"blob"
CLOB	"CLOB"	"clob"
集合类型/数组类型	见说明部分	见说明部分
RECORD类型	见说明部分	见说明部分

版权所有 © 华为云计算技术有限公司

说明

1. 对于上述表格中未列举出的类型，GaussDB暂不支持。
2. 元素为集合类型、数组类型或record类型，且元素类型是package里定义的类型时，getBaseTypeName返回规则如下：
   - OJDBC11返回schemaName.packageName.typeName。
   - OJDBC8一般返回schemaName.packageName.typeName，在类型名满足以下条件返回"schemaName"."packageName.typeName"：

     schemaName/packageName/typeName中有任意名称不符合首字符为字母，其他字符为字母、数字或下划线。
   - GaussDB一般返回schemaName.packageName.typeName，在类型名满足以下条件返回"schemaName"."packageName"."typeName"：

     schemaName/packageName/typeName中有任意名称不符合首字符为字母或下划线，其他字符为字母、数字或下划线。
3. 元素为集合类型、数组类型或record类型，且元素类型是schema下定义的类型时（非package里定义的类型），getBaseTypeName返回规则如下：
   - OJDBC11返回schemaName.typeName。
   - OJDBC8一般返回schemaName.typeName，在类型名满足以下条件返回"schemaName"."typeName"：

     schemaName/typeName中有任意名称不符合首字符为字母，其他字符为字母、数字或下划线。
   - GaussDB一般返回schemaName.typeName，在类型名满足以下条件返回"schemaName"."typeName"：

     schemaName/typeName中有任意名称不符合首字符为字母或下划线，其他字符为字母、数字或下划线。
4. 若元素类型在创建时不做特殊处理，Oracle的getBaseTypeName接口通常返回的是大写的类型名称，GaussDB的getBaseTypeName接口通常返回的是小写的类型名称。

表 3-40 Array 构造接口详细差异

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
CHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。

版权所有 © 华为云计算技术有限公司

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
VARCHAR/ VARCHAR2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。
NCHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。
NVARCHAR2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。
NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。

版权所有 © 华为云计算技术有限公司

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
NUMERIC	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
DECIMAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 当入参超过Integer范围时GaussDB会报错。
SMALLINT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 当入参超过Short范围时GaussDB会报错。

版权所有 © 华为云计算技术有限公司

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
DOUBLE PRECISION	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
FLOAT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
REAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Float可能会出现精度丢失的情况。
BINARY_DOUBLE	Double、oracle.sql.BINARY_DOUBLE	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。

版权所有 © 华为云计算技术有限公司

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
BINARY_INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAdder、LongAccumulator、LongAdder、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. OJDBC11元素入参支持的类型：Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER 3. 目标类型不一致，参考getArray接口详细差异。 4. 当传入数值超过Integer范围时：GaussDB会报错 Oracle OJDBC8会进行截断处理。
BOOLEAN	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAdder、LongAccumulator、LongAdder、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. OJDBC8元素入参支持的类型：Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER 3. 目标类型不一致，参考getArray接口详细差异。 4. GaussDB目标类型为Boolean类型，仅支持1、0、"true"、"false"等输入。

版权所有 © 华为云计算技术有限公司

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
TIMESTAMP	byte[]、java.sql.Date、Calendar、java.util.Date、LocalDate、LocalDateTime、LocalTime、OffsetDateTime、OffsetTime、String、java.sql.Time、java.sql.Timestamp、oracle.sql.DATE、oracle.sql.TIMESTAMP、oracle.sql.TIMESTAMPTZ、oracle.sql.TIMESTAMPLTZ、ZonedDateTime	java.util.Date、java.sql.Date、java.sql.Time、java.sql.Timestamp、LocalDateTime、String	元素入参支持的类型差异见表格。
TIMESTAMP WITH TIME ZONE	java.sql.Date、Calendar、java.util.Date、LocalDate、LocalDateTime、LocalTime、OffsetDateTime、OffsetTime、String、java.sql.Time、java.sql.Timestamp、oracle.sql.DATE、oracle.sql.TIMESTAMP、oracle.sql.TIMESTAMPTZ、oracle.sql.TIMESTAMPLTZ、ZonedDateTime	java.util.Date、java.sql.Date、java.sql.Time、java.sql.Timestamp、LocalDateTime、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。
BLOB	oracle.sql.BLOB、oracle.jdbc.driver.OracleBlob	PGBlob	元素入参支持的类型差异见表格。

版权所有 © 华为云计算技术有限公司

元素数据库类型	元素入参支持的Java类型列表（Oracle OJDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
CLOB	oracle.sql.CLOB、oracle.jdbc.driver.OracleClob、InputStream、Reader	PGClob	元素入参支持的类型差异见表格。
集合类型/数组类型	ARRAY、Object	GaussArray、Object	1. 元素入参支持的类型差异见表格。 2. Oracle：当元素入参为ARRAY时，ARRAY的类型与元素类型不一致时不会报错。 GaussDB：当元素入参为GaussArray时，GaussArray的类型与元素类型不一致时会报错。 3. 当元素入参为Object时，参考Array构造接口差异。
RECORD类型	STRUCT、Object[]	GaussStruct、Object[]	1. 元素入参支持的类型差异见表格。 2. Oracle：当元素入参为STRUCT时，STRUCT的类型与元素类型不一致时不会报错。 GaussDB：当元素入参为GaussStruct时，GaussStruct的类型与元素类型不一致时会报错。 3. 当元素入参为Object[]时，参考Struct构造接口差异。

说明

1. 构造Array时，若传入元素的Java类型不符合目标类型，会进行相应的隐式转换操作，对于不同的数据库元素类型，支持传入的元素的Java类型差异见上表。
2. 对于上述表格中未列举出的类型，GaussDB暂不支持。
3. 构造方法需要提供元素数组，上述表格描述的是数组中每一个元素对应的差异。

版权所有 © 华为云计算技术有限公司

3.2.7.1.2 Struct

本章节描述在JDBC驱动使用java.sql.Struct类型时，Oracle与GaussDB的差异。

表 3-41 构造方法参考

构造方式	Oracle数据库	GaussDB数据库	接口差异
1. StructDescriptor的静态构造方法构造StructDescriptor对象 2. 通过StructDescriptor构造Struct对象	String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; StructDescriptor desc = StructDescriptor.createDescriptor(typeName, conn); Struct struct = new STRUCT(desc, conn, attributes);	String typeName = "xxx"; Connection conn = getConnection(); Object[] elements = null; StructDescriptor desc = StructDescriptor.getDescriptor(typeName, conn); Struct struct = new GaussStruct(desc, attributes);	• 构造StructDescriptor的静态方法名不一致，Oracle的方法名为createDescriptor，GaussDB的方法名为getDescriptor。 • Struct的构造方法名定义不一致，Oracle为STRUCT(StructDescriptor, Connection, Object[])，GaussDB为GaussStruct(StructDescriptor, Object[])。 • 变量说明： typeName为类型名称，且大小写敏感，通常Oracle的类型名大写，GaussDB的类型名小写。 conn为对应数据库的连接对象。 attributes为元素数据数组。
通过Connection的createStruct标准接口构造Struct对象	String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; Struct struct = conn.createStruct(typeName, attributes);	String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; Struct struct = conn.createStruct(typeName, attributes);	• 变量说明： typeName为类型名称，且大小写敏感，通常Oracle的类型名大写，GaussDB的类型名小写。 conn为对应数据库的连接对象。 attributes为元素数据数组。

版权所有 © 华为云计算技术有限公司

说明

1. 对于上述表格中未列举出的构造方式，GaussDB暂不支持。
2. 若attribute的类型为字符类型，构造传入的字符串长度大于元素类型定义的长度，Oracle会在绑定入参时报错。

   GaussDB在构造或者绑定入参时不会对类型修饰符进行校验，在数据库接收Struct对象并执行SQL时，由数据库决定是否报错。
3. 数组元素数量大于对应类型的实际列数量，则会在创建时报错。

   数组元素数量小于实际列数量时：Oracle能创建成功，在传入参执行时会报错；GaussDB会在创建时报错。

表 3-42 接口参考

方法名	返回值类型	throws	GaussDB数据库
getSQLTypeName()	String	SQLException	支持
getAttributes()	Object[]	SQLException	支持
getAttributes(java.util.Map<String,Class<?>> map)	Object[]	SQLException	不支持

版权所有 © 华为云计算技术有限公司

说明

getSQLTypeName接口差异参考如下说明：

1. 对于package类型，通过packageName.typeName形式构造的Struct，getSQLTypeName接口差异如下：
   - OJDBC11返回packageName.typeName。
   - OJDBC8返回packageName.typeName，在packageName和typeName满足以下条件时返回"packageName"."typeName"：

     packageName/typeName有任意名称不符合首字符为字母，其他字符为字母或者数字或者下划线。
   - GaussDB返回schemaName.packageName.typeName，在schemaName、packageName和typeName满足以下条件时返回"schemaName"."packageName"."typeName"：

     schemaName/packageName/typeName有任意名称不符合首字符为字母或者下划线，其他字符为字母、数字或者下划线。
2. 对于package类型，其他场景下getSQLTypeName接口差异如下：
   - OJDBC11返回schemaName.packageName.typeName。
   - OJDBC8返回schemaName.packageName.typeName，在schemaName、packageName和typeName满足以下条件时返回"schemaName"."packageName.typeName"：

     schemaName/packageName/typeName有任意名称不符合首字符为字母，其他字符为字母或者数字或者下划线。
   - GaussDB返回schemaName.packageName.typeName，在schemaName、packageName和typeName满足以下条件时返回"schemaName"."packageName"."typeName"：

     schemaName/packageName/typeName有任意名称不符合首字符为字母或者下划线，其他字符为字母、数字或者下划线。
3. 对于非package类型，getSQLTypeName接口差异如下：
   - OJDBC11返回schemaName.typeName。
   - OJDBC8返回schemaName.typeName，在schemaName和typeName满足以下条件时返回"schemaName"."typeName"：

     schemaName/typeName有任意名称不符合首字符为字母，其他字符为字母或者数字或者下划线。
   - GaussDB返回schemaName.typeName，在schemaName和typeName满足以下条件时返回"schemaName"."typeName"：

     schemaName/typeName有任意名称不符合首字符为字母或者下划线，其他字符为字母、数字或者下划线。

表 3-43 getAttributes()接口详细差异

Attribute数据库类型	返回值中对应元素的Java类型（Oracle OJDBC8）	返回值中对应元素的Java类型（Oracle OJDBC11）	返回值中对应元素的Java类型（GaussDB）
CHAR	String	String	String
VARCHAR/ VARCHAR2	String	String	String
NCHAR	String	String	String

版权所有 © 华为云计算技术有限公司

Attribute数据库类型	返回值中对应元素的Java类型（Oracle OJDBC8）	返回值中对应元素的Java类型（Oracle OJDBC11）	返回值中对应元素的Java类型（GaussDB）
NVARCHAR2	String	String	String
NUMBER	BigDecimal	BigDecimal	BigDecimal
NUMERIC	BigDecimal	BigDecimal	BigDecimal
DECIMAL	BigDecimal	BigDecimal	BigDecimal
INTEGER	BigDecimal	BigDecimal	Integer
SMALLINT	BigDecimal	BigDecimal	Short
DOUBLE PRECISION	BigDecimal	BigDecimal	Double
FLOAT	BigDecimal	BigDecimal	Double
REAL	BigDecimal	BigDecimal	Float
BINARY_DOUBLE	Double	Double	Double
BINARY_INTEGER	BigDecimal	Integer	Integer
BOOLEAN	BigDecimal	Integer	Boolean
TIMESTAMP	Timestamp	Timestamp	Timestamp
TIMESTAMP WITH TIME ZONE	TIMESTAMPTZ	TIMESTAMPTZ	Timestamp
BLOB	BLOB	BLOB	PGBlob
CLOB	CLOB	CLOB	PGClob
集合类型/数组类型	ARRAY	ARRAY	GaussArray
RECORD类型	STRUCT	STRUCT	GaussStruct

说明

对于上述表格中未列举出的类型，GaussDB暂不支持。

版权所有 © 华为云计算技术有限公司

表 3-44 Struct 构造接口详细差异

Attribute数据库类型	Attribute入参支持的Java类型列表（Oracle）	Attribute入参支持的Java类型列表（GaussDB）	差异说明
CHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	1. attribute入参支持的类型差异见表格。 2. 当入参为String时，Oracle会在字符串末尾补齐空格至字符串长度与类型定义的长度一致；GaussDB不会补空格。
VARCHAR/VARCHAR2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	attribute入参支持的类型差异见表格。
NCHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	attribute入参支持的类型差异见表格。
NVARCHAR2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	attribute入参支持的类型差异见表格。

版权所有 © 华为云计算技术有限公司

Attribute数据库类型	Attribute入参支持的Java类型列表（Oracle）	Attribute入参支持的Java类型列表（GaussDB）	差异说明
NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
NUMERIC	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
DECIMAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 当入参超过Integer范围时GaussDB会报错。

版权所有 © 华为云计算技术有限公司

Attribute数据库类型	Attribute入参支持的Java类型列表（Oracle）	Attribute入参支持的Java类型列表（GaussDB）	差异说明
SMALLINT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 当入参超过Short范围时GaussDB会报错。
DOUBLE PRECISION	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
FLOAT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
REAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Float可能会出现精度丢失的情况。

版权所有 © 华为云计算技术有限公司

Attribute数据库类型	Attribute入参支持的Java类型列表（Oracle）	Attribute入参支持的Java类型列表（GaussDB）	差异说明
BINARY_DOUBLE	byte[]、Double、oracle.sql.BINARY_DOUBLE	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
BINARY_INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAdder、LongAccumulator、LongAdder、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当传入数值超过Integer范围时，GaussDB会报错，Oracle会进行截断处理。
BOOLEAN	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAdder、LongAccumulator、LongAdder、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. GaussDB目标类型为Boolean类型，仅支持1、0、"true"、"false"等输入。

版权所有 © 华为云计算技术有限公司

Attribute数据库类型	Attribute入参支持的Java类型列表（Oracle）	Attribute入参支持的Java类型列表（GaussDB）	差异说明
TIMESTAMP	byte[]、java.sql.Date、String、java.sql.Time、java.sql.Timestamp、oracle.sql.TIMESTAMP、oracle.sql.DATE	java.util.Date、java.sql.Date、java.sql.Time、java.sql.Timestamp、LocalDateTime、String	attribute入参支持的类型差异见表格。
TIMESTAMP WITH TIME ZONE	java.sql.Date、Calendar、java.util.Date、LocalDate、LocalDateTime、LocalTime、OffsetDateTime、OffsetTime、String、java.sql.Time、java.sql.Timestamp、oracle.sql.TIMESTAMP、oracle.sql.TIMESTAMPTZ、oracle.sql.TIMESTAMPLTZ、ZonedDateTime	java.util.Date、java.sql.Date、java.sql.Time、java.sql.Timestamp、LocalDateTime、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。
BLOB	oracle.sql.BLOB、oracle.jdbc.driver.OracleBlob	PGBlob	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。
CLOB	oracle.sql.CLOB、oracle.jdbc.driver.OracleClob	PGClob	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。

版权所有 © 华为云计算技术有限公司

Attribute数据库类型	Attribute入参支持的Java类型列表（Oracle）	Attribute入参支持的Java类型列表（GaussDB）	差异说明
集合类型/数组类型	ARRAY、Object	GaussArray、Object	1. attribute入参支持的类型差异见表格。 2. Oracle：当attribute入参为ARRAY时，ARRAY的类型与attribute实际需要的类型不一致时不会报错。 GaussDB：当attribute入参为GaussArray时，GaussArray的类型与attribute实际需要的类型不一致时会报错。 3. 当attribute入参为Object时，参考Array构造接口差异。
RECORD类型	STRUCT、Object[]	GaussStruct、Object[]	1. attribute入参支持的类型差异见表格。 2. Oracle：当attribute入参为STRUCT时，STRUCT的类型与attribute实际需要的类型不一致时不会报错。 GaussDB：当attribute入参为GaussStruct时，GaussStruct的类型与attribute实际需要的类型不一致时会报错。 3. 当attribute入参为Object[]时，参考Struct构造接口差异。

版权所有 © 华为云计算技术有限公司

说明

1. 构造Struct时，若传入元素的Java类型不符合目标类型，会进行相应的隐式转换操作，对于不同的数据库元素类型，支持传入的元素的Java类型差异见上表。
2. 对于上述表格中未列举出的类型，GaussDB暂不支持。
3. 构造方法需要提供属性数组，上述表格描述的是数组中每一个attribute对应的差异。

3.2.8 常见的 SQL DDL 子句

本章节描述常见的SQL DDL子句兼容项，包含分配扩展子句、约束、取消分配未使用子句、文件规范、日志记录子句、并行子句、物理属性子句、大小子句、存储子句、聚集函数嵌套、创建序列、修改序列，如表3-45所示。

表 3-45 常用 SQL DDL 子句

Oracle数据库	GaussDB数据库	差异
分配扩展子句 语法： ALLOCATE EXTENT [({ SIZE size_clause | DATAFILE 'filename' | INSTANCE integer } ...)] 例如：创建employees表后，改变表的分配扩展size为10M。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. SQL> ALTER TABLE employees ALLOCATE EXTENT (SIZE 10M); Table altered.	不支持。	-
约束 语法： { inline_constraint | out_of_line_constraint | inline_ref_constraint | out_of_line_ref_constraint } 例如：创建表staff，约束子句中指定ID列、NAME列不为空。 SQL> CREATE TABLE staff(ID INT NOT NULL, NAME char(8) NOT NULL, AGE INT, ADDRESS CHAR(50), SALARY REAL); Table created.	支持。	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
取消分配未使用子句 语法： DEALLOCATE UNUSED [KEEP size_clause] 例如：创建employees表，进行了一些插入、删除操作后，希望使用取消分配未使用子句释放employees表未使用的空间。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. - 进行一些插入、删除操作 SQL> ALTER TABLE employees DEALLOCATE UNUSED; Table altered.	不支持。	-
文件规范 语法： {['filename' | 'ASM_filename'] [SIZE size_clause] [REUSE] [autoextend_clause]} | {['filename | ASM_filename' | ('filename | ASM_filename' [, 'filename | ASM_filename']...)] [SIZE size_clause] [BLOCKSIZE size_clause [REUSE]} 例如：创建一个临时表空间tbs_temp_01，SQL语句的文件规范子句中指定在表空间中创建一个临时数据库文件temp01.dbf，可以自动扩展，并将表空间分配给表空间组tbs_grp_01。 SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	不支持。	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

日志记录子句
语法：
{ LOGGING | NOLOGGING | FILESYSTEM_LIKE_LOGGING }
	部分支持，有差异。	• GaussDB不支持LOGGING约束子句和FILESYSTEM_LIKE_LOGGING约束子句。
		例如：
		GaussDB创建表，带LOGGING约束子句，语法报错。
		gaussdb=# CREATE LOGGING TABLE my_tab(id int, name char(16));
		ERROR: syntax error at or near "LOGGING"
		LINE 1: CREATE LOGGING TABLE my_tab(id int, name char(16));
		^
		GaussDB创建表，带FILESYSTEM_LIKE_LOGGING约束子句，语法报错。
		gaussdb=# CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name char(16));
		ERROR: syntax error at or near "FILESYSTEM_LIKE_LOGGING"
		LINE 1: CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name cha...
		^
		• GaussDB仅支持表级的UNLOGGED约束，不支持列级的UNLOGGED约束。
		例如：GaussDB创建表，带列级的UNLOGGED约束子句，语法报错。
		gaussdb=# CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16));
		ERROR: syntax error at or near "UNLOGGED"
		LINE 1: CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16))...
		^
		• GaussDB仅支持在CREATE TABLE、CREATE TABLE AS、SELECT INTO语句中使用日志记录子句。
		例如：GaussDB创建TABLESPACE，带UNLOGGED约束子句，语法报错。
		gaussdb=# CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespace1/tablespace_1';
		ERROR: syntax error at or near "TABLESPACE"
		LINE 1: CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespac...
		^

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
并行子句 语法： { NOPARALLEL | PARALLEL [integer] } 例如：创建表t1，并在并行子句中指定PARALLEL 4，意为查询和更新表t1时最多使用4个并行进程操作。 SQL> CREATE TABLE t1 (id NUMBER, name VARCHAR2(50)) PARALLEL 4; Table created.	不支持。	-

物理属性子句
语法：
[{ PCTFREE integer | PCTUSED integer | INITRANS integer | storage_clause }...]
	部分支持，有差异。	• GaussDB不支持PCTUSED。
		例如：执行在表tbl1中创建一个tbl1_ind的索引的SQL语句，并在该语句的物理属性子句中指定索引的空间利用率PCTUSED为20%，GaussDB执行该SQL语句语法报错。
		gaussdb=# CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20;
		ERROR: syntax error at or near "PCTUSED"
		LINE 1: CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20;
		^
		• GaussDB 仅支持在CREATE TABLE、CREATE INDEX语句中使用物理属性子句。
		例如：尝试从表tbl1中获取数据，创建物化视图tbl1_mv，并在物理属性子句中指定该视图的初始化事务数为30，GaussDB执行该SQL语句语法报错。
		gaussdb=# CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * from tbl1;
		ERROR: syntax error at or near "INITRANS"
		LINE 1: CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * fro...
		^

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
大小子句 语法： integer [K | M | G | T | P | E] 例如：创建一个临时表空间tbs_temp_01，并在表空间中创建一个临时数据库文件temp01.dbf，SQL语句的大小子句中指定初始大小是5M，可以自动扩展，并将表空间分配给表空间组tbs_grp_01。 SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' SIZE 5M AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	不支持。	-
存储子句 语法： STORAGE ({ INITIAL size_clause | NEXT size_clause | MINEXTENTS integer | MAXEXTENTS { integer | UNLIMITED } | maxsize_clause | PCTINCREASE integer | FREELISTS integer | FREELIST GROUPS integer | OPTIMAL [size_clause | NULL] | BUFFER_POOL { KEEP | RECYCLE | DEFAULT } | FLASH_CACHE { KEEP | NONE | DEFAULT } | (CELL_FLASH_CACHE (KEEP | NONE | DEFAULT)) | ENCRYPT } ...)	部分支持，有差异。	• Oracle中由STORAGE子句指定存储参数，而GaussDB中由WITH子句指定存储参数。 例如： Oracle中创建表my_tab1，在存储子句中指定表初始大小为10M，需要更多空间时每次增加5M的SQL语句如下： SQL> CREATE TABLE my_tab1 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) STORAGE (INITIAL 10M NEXT 5M); Table created. GaussDB中创建表my_tab2，在存储子句中指定存储引擎类型为USTORE的SQL语句如下： gaussdb=# CREATE TABLE my_tab2 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) with (storage_type=ustore); NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "my_tab2_pkey" for table "my_tab2" CREATE TABLE • GaussDB中可选的存储参数和Oracle存在很大差异。GaussDB具体可参见**CREATE TABLE**的参数说明部分，“WITH ({storage_parameter = value} [, ...])”中描述了CREATE TABLE语句支持的存储参数。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
聚集函数嵌套 例如：创建由sales表的sales_amount列嵌套聚集函数MIN()、SUM()生成的表revenue。 SQL> CREATE TABLE sales(ID INT, SALES_AMOUNT INT); Table created. SQL> INSERT INTO sales VALUES(1, 100); 1 row created. SQL> INSERT INTO sales VALUES (3, 200); 1 row created. SQL> CREATE TABLE revenue as SELECT SUM(MIN(sales_amount)) as total from sales group by sales_amount; Table created.	支持。	-
删除系统schema 语法： DROP USER schema_name CASCADE; 例如：在SYS用户下删除SYS schema。 SQL> DROP USER SYS; DROP USER SYS * ERROR at line 1: ORA-28050: specified user or role cannot be dropped	支持。	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

删除表的主键约束
语法：
DROP PRIMARY KEY [CASCADE] [(KEEP | DROP) INDEX] [ONLINE]
	部分支持，有差异。	• GaussDB仅支持DROP PRIMARY KEY语法，不支持其他选项。
		• 当进行该操作时，Oracle会同时将创建主键时自动为创建的主键列添加的NOT NULL约束进行删除；GaussDB不会删除，仍然保留NOT NULL约束。
		-- GaussDB:
		gaussdb=# CREATE TABLE t1 (id int PRIMARY KEY);
		NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "t1_pkey" for table "t1"
		CREATE TABLE
		gaussdb=# ALTER TABLE t1 DROP PRIMARY KEY;
		ALTER TABLE
		gaussdb=# INSERT INTO t1 VALUES(null);
		ERROR: The null value in column "id" violates the not-null constraint.
		DETAIL: Failing row contains (null).
		gaussdb=# DROP TABLE t1;
		
		-- Oracle:
		SQL> CREATE TABLE t1 (id int PRIMARY KEY);
		SQL> ALTER TABLE t1 DROP PRIMARY KEY;
		SQL> INSERT INTO t1 VALUES(null);
		-- 该操作可以插入成功，即主键列的NOT NULL约束也同时被删除。
		SQL> DROP TABLE t1;

创建序列 语法： CREATE SEQUENCE [IF NOT EXISTS] [schema.] sequence [SHARING = { METADATA | DATA | NONE }] [{ INCREMENT BY | START WITH } integer | { MAXVALUE integer | NOMAXVALUE } | { MINVALUE integer | NOMINVALUE } | { CYCLE | NOCYCLE } | { CACHE integer | NOCACHE } | { ORDER | NOORDER } | { KEEP | NOKEEP } | { SCALE {EXTEND | NOEXTEND} | NOSCALE } | { SHARD {EXTEND | NOEXTEND} | NOSHARD } | { SESSION | GLOBAL }]...	部分支持，有差异。	• GaussDB不支持SHARING、CACHE、NOCACHE、ORDER、NOORDER、KEEP、NOKEEP、SCALE、NOSCALE、SHARD、NOSHARD、SESSION以及GLOBAL选项。 • 对于步长的绝对值是否小于最大值和最小值的差值的场景，GaussDB不进行检查。 -- GaussDB: gaussdb=# CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 INCREMENT BY 10; CREATE SEQUENCE -- Oracle: SQL> CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 INCREMENT BY 10; CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 INCREMENT BY 10 * ERROR at line 1: ORA-04005: INCREMENT must be less than MAXVALUE minus MINVALUE

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

修改序列
语法：
ALTER SEQUENCE [schema.] sequence { { (INCREMENT BY) | (START WITH) } integer | { MAXVALUE integer | NOMAXVALUE } | { MINVALUE integer | NOMINVALUE } | RESTART | { CYCLE | NOCYCLE } | { CACHE integer | NOCACHE } | { ORDER | NOORDER } | { KEEP | NOKEEP } | { SCALE {EXTEND | NOEXTEND} | NOSCALE } | { SHARD {EXTEND | NOEXTEND} | NOSHARD } | { SESSION | GLOBAL } } ...
	部分支持，有差异。	• GaussDB不支持INCREMENT BY、MINVALUE、NOMINVALUE、RESTART、CYCLE、NOCYLE、CACHE、NOCACHE、ORDER、NOORDER、KEEP、NOKEEP、SCALE、NOSCALE、SHARD、NOSHARD、SESSION以及GLOBAL选项。
		• GaussDB修改最大值时，不检查步长的绝对值是否小于最大值和最小值的差值。
		-- GaussDB:
		gaussdb=# CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 100 INCREMENT BY 10;
		CREATE SEQUENCE
		gaussdb=# ALTER SEQUENCE seq_test MAXVALUE 10;
		ALTER SEQUENCE
		
		-- Oracle:
		SQL> CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 100 INCREMENT BY 10;
		
		Sequence created.
		
		SQL> ALTER SEQUENCE seq_test MAXVALUE 10;
		ALTER SEQUENCE seq_test MAXVALUE 10
		*
		ERROR at line 1:
		ORA-04005: INCREMENT must be less than MAXVALUE minus MINVALUE
		• GaussDB修改最大值时，检查序列的起始值是否在最大值和最小值范围内。
		-- GaussDB:
		gaussdb=# CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 START WITH 10 INCREMENT BY -1;
		CREATE SEQUENCE
		gaussdb=# SELECT seq_test.nextval FROM DUAL;
		nextval
		--------- 	
		10
		(1 row)
		
		gaussdb=# SELECT seq_test.nextval FROM DUAL;
		nextval
		--------- 	
		9
		(1 row)
		
		gaussdb=# SELECT seq_test.nextval FROM DUAL;
		nextval
		--------- 	
		8
		(1 row)
		
		gaussdb=# ALTER SEQUENCE seq_test MAXVALUE 9;

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

		ERROR: START value (10) cannot be greater than MAXVALUE (9).
		
		-- Oracle:
		SQL> CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 START WITH 10 INCREMENT BY -1;
		
		Sequence created.
		
		SQL> SELECT seq_test.nextval FROM DUAL;
		
		  NEXTVAL
		---------- 	
		   10
		
		SQL> SELECT seq_test.nextval FROM DUAL;
		
		  NEXTVAL
		---------- 	
		    9
		
		SQL> SELECT seq_test.nextval FROM DUAL;
		
		  NEXTVAL
		---------- 	
		    8
		
		SQL> ALTER SEQUENCE seq_test MAXVALUE 9;
		
		Sequence altered.
		- GaussDB首次执行nextval前内部位置是起始值，修改最大值时，检查序列的起始值是否在最大值和最小值范围内。
		-- GaussDB:
		gaussdb=# CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 START WITH 6 INCREMENT BY -5;
		CREATE SEQUENCE
		gaussdb=# ALTER SEQUENCE seq_test MAXVALUE 9;
		ALTER SEQUENCE
		
		-- Oracle:
		SQL> CREATE SEQUENCE seq_test MINVALUE 1 MAXVALUE 10 START WITH 6 INCREMENT BY -5;
		
		Sequence created.
		
		SQL> ALTER SEQUENCE seq_test MAXVALUE 9;
		ALTER SEQUENCE seq_test MAXVALUE 9
		*
		ERROR at line 1:
		ORA-04009: MAXVALUE cannot be made to be less than the current value

3.2.9 SQL 查询和子查询

GaussDB数据库兼容除分层查询以外的SQL查询和子查询。

版权所有 © 华为云计算技术有限公司

表 3-46 SQL 查询和子查询

Oracle数据库	GaussDB数据库	差异
创建简单查询	支持	-
分层查询	不支持	-
UNION [ALL]，INTERSECT，减运算符	支持	-
查询结果排序	支持，有差异	GaussDB查询不包含分组，且目标列同时包含聚集函数和集合返回函数时，不忽略对集合返回函数列的排序。 当查询列涉及类型转换且排序键为原始列名时，GaussDB因转换结果默认别名为原始列名，会按转换后的类型排序；Oracle因转换结果默认别名为转换表达式，仍按原始列的类型排序。两者排序语义不同，可能导致结果集顺序不一致。
Joins	支持，有差异	GaussDB只支持和Oracle相同的Join Types，如left、right、self 、natural、full outer join等。不支持In-Memory Join Groups等Join Optimizations方法。
使用子查询	支持	-
嵌套子查询的解嵌套	支持，有差异	GaussDB不支持显式指定HASH_AJ或MERGE_AJ。
分布式查询	支持，有差异	GaussDB需要显式DBLINK查询。
聚集函数嵌套	支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

层次查询	支持，有差异	• GaussDB在非CONNECT BY子句中使用PRIOR关键字修饰列时，列回显名称与Oracle有差异。例如对于列a，GaussDB回显为__prior_a__，Oracle回显为PRIORA。
		• 在SELECT的GROUP BY子句中使用带有PRIOR列的表达式，GaussDB表现与Oracle有差异。GaussDB支持在GROUP BY子句中使用带有PRIOR列的表达式，Oracle会报错。
		-- GaussDB:
		gaussdb=# CREATE TABLE test (c1 INT, c2 INT, c3 INT);
		NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by default.
		HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
		CREATE TABLE
		gaussdb=# INSERT INTO test VALUES(1,2,3);
		INSERT 0 1
		gaussdb=# INSERT INTO test VALUES(2,3,4);
		INSERT 0 1
		gaussdb=# SELECT PRIOR c1 + 1 AS result FROM test CONNECT BY PRIOR c2 = c1 GROUP BY PRIOR c1 + 1;
		result
		-------- 	
		2
		(2 rows)
		gaussdb=# DROP TABLE test;
		DROP TABLE
		-- Oracle:
		SQL> CREATE TABLE test (c1 INT, c2 INT, c3 INT);
		SQL> INSERT INTO test VALUES(1,2,3);
		SQL> INSERT INTO test VALUES(2,3,4);
		-- 该操作会报错not a GROUP BY expression
		SQL> SELECT PRIOR c1 + 1 AS result FROM test CONNECT BY PRIOR c2 = c1 GROUP BY PRIOR c1 + 1;
		SQL> DROP TABLE test;
		• GaussDB语法上允许ORDER SIBLINGS BY和ORDER BY同时使用，此时最终输出的排序结果以外层的ORDER BY为准，Oracle不支持该用法。

目前xmltable_clause实现的功能与Oracle数据库和PG数据库还存在如下差异，如表3-47所示。

 版权所有 © 华为云计算技术有限公司

表 3-47 GaussDB 与 ORA 数据库和 PG 数据库对比

GaussDB	PG数据库	Oracle数据库
在row_expression处及columns_clause中的PATH处使用XPath 1.0表达式。	与GaussDB一致。	在与row_expression和columns_clause中的PATH对应处使用XQuery 1.0表达式。
不支持默认命名空间功能。	与GaussDB一致。	支持默认命名空间功能。
支持在passing_clause子句中传入单个数据但不可取别名。	与GaussDB一致（注意传入数据为XML类型）。	支持在对应子句部分传入多个数据且可取别名。
支持省略passing_clause子句。	不支持省略passing_clause子句。	支持省略对应子句。
不支持在passing_clause子句后使用RETURNING SEQUENCE BY REF子句。	与GaussDB一致。	支持在对应子句后使用RETURNING SEQUENCE BY REF子句。
不支持在columns_clause子句中使用(SEQUENCE) BY REF修饰XMLTYPE类型的返回数据。	与GaussDB一致。	支持在对应子句中使用(SEQUENCE) BY REF修饰XMLTYPE类型的返回数据。
在省略columns_clause子句中的PATH部分时，若不使用""将列名包起，在传入XMLTABLE的数据中对应节点的节点名为小写的情况下能正确找到该节点进行后续操作。	与GaussDB一致。	在省略对应部分时，若不使用""将列名包起，在传入XMLTABLE的数据中对应节点的节点名为大写的情况下能正确找到该节点进行后续操作。
不支持省略columns_clause子句中列类型的声明。	与GaussDB一致。	支持省略对应子句中列类型的声明。
在ORA兼容模式下支持省略columns_clause子句，在PG兼容模式下不支持省略columns_clause子句。	不支持省略对应子句。	支持省略对应子句。
在ORA兼容模式下返回数据长度超过类型的typmod时会截断处理，在PG兼容模式下返回数据长度超过类型的typmod时会报错。	返回数据长度超过类型的typmod时会报错。	返回数据长度超过类型的typmod时会截断处理。

示例：

```
gaussdb=# CREATE DATABASE test DBCOMPATIBILITY = 'ORA';
gaussdb=# \c test
test=# SELECT * FROM XMLTABLE(
```


版权所有 © 华为云计算技术有限公司

```
test(# XMLNAMESPACES('nspace1' AS "ns1", 'nspace2' AS "ns2"), -- 声明两个XML的命名空间'nspace1'和
'nspace2'及对应的别名"ns1"和"ns2"
test(# '/ns1:root/*:child' -- 经row_expression从传入的数据中选取命名空间为'nspace1'的root节点，在选取其下
面的所有child节点，忽略child的命名空间；其中ns1为'nspace1'的别名
test(# PASSING xmltype(
test(# '<root xmlns="nspace1">
test'#     <child>
test'#         <name>peter</name>
test'#         <age>11</age>
test'#     </child>
test'#     <child xmlns="nspace1">
test'#         <name>qiqi</name>
test'#         <age>12</age>
test'#     </child>
test'#     <child xmlns="nspace2">
test'#         <name>hacker</name>
test'#         <age>15</age>
test'#     </child>
test'# </root>')
test(# COLUMNS
test(# columns FOR ORDINALITY, -- 该列为行号列
test(# name varchar(10) path 'ns1:name', -- 从row_expression获取的每个child节点中选取命名空间为'nspace1'
的name节点，并将节点中的值转换为varchar(10)返回；其中ns1为'nspace1'的别名
test(# age int); -- 从row_expression获取的每个child节点中选取age节点，并将节点中的值转换为int返回；由于
第一个child节点没有显式指明命名空间，故跟随父节点root的命名空间，故该列一个值都没有返回
 column | name  | age
--------+-------+-----
      1 | peter |
      2 | qiqi  |
      3 |       |
(3 rows)

--切换至默认数据库并删除test数据库（需按照实际情况修改数据库名）。
gaussdb=# \c postgres
gaussdb=# DROP DATABASE test;
```

3.2.10 PL/SQL 语言

GaussDB数据库基本兼容的PL/SQL操作符、表达式，控制语句、集合和record等等，不支持预定义的PL/SQL常量和类型、子类型等。

3.2.10.1 PL/SQL 基本语法

表 3-48 PL/SQL 操作符

Oracle数据库	GaussDB数据库
+	支持
:=	支持
=>	支持
%	支持
'	支持
.	支持
||	支持
/	支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
**	不支持
(	支持
)	支持
:	支持
,	支持
<<	支持
>>	支持
/*	支持
*/	支持
*	支持
"	支持
..	支持
=	支持
<>	支持
!=	支持
~=	支持
^=	支持
<	支持
>	支持
<=	支持
>=	支持
@	支持
--	支持
;	支持
-	支持

表 3-49 逻辑运算符

Oracle数据库	GaussDB数据库
NOT	支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
AND	支持
OR	支持

表 3-50 比较表达式

Oracle数据库	GaussDB数据库
IS [NOT] NULL	支持
LIKE	支持
BETWEEN	支持
IN	支持

表 3-51 条件表达式

Oracle数据库	GaussDB数据库
simple CASE	支持
searched CASE	支持

表 3-52 变量声明相关参数

Oracle数据库	GaussDB数据库	差异
%TYPE	支持，有差异	• GaussDB不支持record变量%type。 • GaussDB不支持pkg.record变量%type、schema.pkg.record变量%type作为出入参类型。 • GaussDB不支持表/视图.column.column%type、schema.表/视图.column.column%type嵌套1层及以上，作为变量类型或者出入参类型。 • GaussDB不支持record变量.column.column %type、pkg.record变量.column.column%type嵌套1层及以上的record的某列类型，作为变量类型或者出入参类型。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
%ROWTYPE	支持，有差异	• GaussDB在多个CN的环境下，存储过程中无法声明临时表的%ROWTYPE及%TYPE属性。因为临时表仅在当前session有效，在编译阶段其他CN无法看到当前CN的临时表。故多个CN的环境下，会提示该临时表不存在。 • GaussDB不支持view%rowtype、schema.view%rowtype作为出入参类型。 • GaussDB不支持package.cursor变量%rowtype作为出入参类型。

3.2.10.2 数据类型兼容性

表 3-53 其他 PL/SQL 数据类型

Oracle数据库	GaussDB数据库	差异
CHARACTER	支持，有差异	• GaussDB中字节长度限制为：1~10485760。 • Oracle中字节长度限制为：1~32767。
VARCHAR	支持，有差异	• GaussDB中字节长度限制为：1~10485760。 • Oracle中字节长度限制为：1~32767。
STRING	不支持	-
PLS_INTEGER	不支持	GaussDB中可使用INT类型替代。
BINARY_INTEGER	支持	-

3.2.10.3 控制语句

表 3-54 条件语句

Oracle数据库	GaussDB数据库
IF THEN	支持
IF THEN ELSE	支持
IF THEN ELSIF	支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
simple CASE: CASE selector WHEN selector_value_1 THEN statements_1 WHEN selector_value_2 THEN statements_2 ... WHEN selector_value_n THEN statements_n [ELSE else_statements END CASE;]	支持
searched CASE: CASE WHEN condition_1 THEN statements_1 WHEN condition_2 THEN statements_2 ... WHEN condition_n THEN statements_n [ELSE else_statements END CASE;]	支持

表 3-55 LOOP 循环语句

Oracle数据库	GaussDB数据库
[label] LOOP statements END LOOP [label];	支持
EXIT;	支持
EXIT WHEN;	支持
CONTINUE;	支持
CONTINUE WHEN;	支持

版权所有 © 华为云计算技术有限公司

表 3-56 FOR 循环语句

Oracle数据库	GaussDB数据库	差异
[label] FOR index IN [REVERSE] lower_bound..upper_bound LOOP statements END LOOP [label];	支持，有差异	GaussDB使用REVERSE关键字时，lower_bound必须大于等于upper_bound，否则循环体不会被执行。
EXIT WHEN;	支持	-
CONTINUE WHEN;	支持	-

表 3-57 WHILE LOOP 循环语句

Oracle数据库	GaussDB数据库
[label] WHILE condition LOOP statements END LOOP [label];	支持

表 3-58 GOTO 语句

Oracle数据库	GaussDB数据库
GOTO	支持

表 3-59 NULL 语句

Oracle数据库	GaussDB数据库
NULL	支持

3.2.10.4 集合和 Record

表 3-60 类型

Oracle数据库	GaussDB数据库
Associative array (or index-by table)	支持
VARRAY (variable-size array)	支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
Nested table	支持
record	支持

表 3-61 语法

Oracle数据库	GaussDB数据库	差异
Associative array (or index-by table)语法： TABLE OF datatype [NOT NULL] INDEX BY { PLS_INTEGER | BINARY_INTEGER | VARCHAR2 (v_size) | data_type }	支持，有差异	• GaussDB不支持 PLS_INTEGER类型，GaussDB内data_type可以为基础数据类型、或存储过程内定义的record类型、集合类型、数组类型，不支持ref cursor类型。 • GaussDB内NOT NULL只支持语法不支持功能，即不会校验元素是否为NULL。 • 详情可参考**集合**章节。
VARRAY (variable-size array)语法： { VARRAY | [VARYING] ARRAY } (size_limit) OF datatype [NOT NULL]	支持，有差异	• GaussDB内不支持NOT NULL语法。 • GaussDB内不支持datatype为varray类型（varray不能嵌套）。 • size_limit功能生效需要在behavior_compat_optionsGUC参数中开启varray_compat参数。 • 详情可参考**数组**章节。
Nested table语法： TABLE OF datatype [NOT NULL]	支持，有差异	• GaussDB内NOT NULL只支持语法不支持功能。 • 详情可参考**集合**章节。
record语法： TYPE record_type IS RECORD (field_definition [, field_definition]...) ;	支持	• record的列可以定义为NOT NULL属性也可以指定默认值。其他类型嵌套record类型，record类型的默认值和NOT NULL不生效；通过package.record_type访问类型的形式来创建record变量，该record变量的默认值和NOT NULL不生效。 • 详情可参考**record**章节。

版权所有 © 华为云计算技术有限公司

表 3-62 构造器

Oracle数据库	GaussDB数据库
collection_type ([value [, value]...])	支持

表 3-63 变量赋值

Oracle数据库	GaussDB数据库	差异
Associative array (or index-by table)	支持	-
VARRAY (variable-size array)	支持，有差异	• GaussDB不同VARRAY类型的数据可以相互赋值，取决于其元素之间是否能相互隐式转换。 • 详情可参考**数组**章节。
Nested table	支持	-
record	支持，有差异	• GaussDB不同record类型的数据可以相互赋值，取决于列与列之间是否能相互隐式转换。 • 详情可参考**record**章节。

表 3-64 集合操作符

Oracle数据库	GaussDB数据库	差异
=	支持，有差异	• Oracle：比较时忽略集合成员先后顺序。 • GaussDB：比较时严格按照集合成员先后顺序。
<>	支持，有差异	• Oracle：比较时忽略集合成员先后顺序。 • GaussDB：比较时严格按照集合成员先后顺序。
IS [NOT] NULL	支持	-
^=	支持，有差异	• Oracle：比较时忽略集合成员先后顺序。 • GaussDB：比较时严格按照集合成员先后顺序。
~=	不支持	-
IS [NOT] A SET	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
IS [NOT] EMPTY	不支持	-
expr [NOT] MEMBER [OF] nested_table	不支持	-
nested_table1 [NOT] SUBMULTISET [OF] nested_table2	不支持	-
[NOT] IN	支持	• Oracle：比较时忽略集合成员先后顺序。 • GaussDB：比较时严格按照集合成员先后顺序。

表 3-65 集合 MULTISET 函数

Oracle数据库	GaussDB数据库
MULTISET UNION [ALL | DISTINCT]	支持
MULTISET EXCEPT [ALL | DISTINCT]	支持
MULTISET INTERSECT [ALL | DISTINCT]	支持

表 3-66 集合类型函数

Oracle数据库	GaussDB数据库	差异
exists(idx)	支持	-
extend[(count[, idx])]	支持，有差异	GaussDB仅支持nesttable类型。
delete[(idx1[, idx2])]	支持	-
trim[(n)]	支持，有差异	GaussDB仅支持nesttable类型。
count	支持	-
first	支持	-
last	支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
prior(idx)	支持	-
next(idx)	支持	-
limit	支持，有差异	GaussDB仅支持nesttable类型。

表 3-67 record 变量操作

Oracle数据库	GaussDB数据库
构造器	支持
%ROWTYPE声明变量	支持
定义常量constant	不支持

表 3-68 集合相关函数

Oracle数据库	GaussDB数据库	差异
unnest_table(anynesttable)	支持	-
unnest_table(anyindexbytable)	支持	-
table(anyarray)	不支持	GaussDB使用unnest(anyarray)函数进行等价改写。

3.2.10.5 静态 SQL

表 3-69 静态查询 SQL 语句

Oracle数据库	GaussDB数据库	差异
SELECT	支持，有差异	GaussDB和Oracle在某些场景下有不同。 GaussDB中FOR SHARE对检索出来的行加共享锁，不同事务的共享锁不会互相阻塞。若数据在一个事务中被FOR SHARE锁定，在另一个事务中使用SELECT FOR SHARE SKIP LOCKED时，SKIP LOCKED不会跳过锁。

版权所有 © 华为云计算技术有限公司

表 3-70 静态 DML SQL 语句

Oracle数据库	GaussDB数据库	差异
INSERT	支持，有差异	• Oracle允许目标表的列数多于子查询结果的列数，但必须显式指定要插入的列名，以确保列数匹配。 • GaussDB可以省略显式指定要插入的列名，该情况下子查询结果的第1列值插入到目标表的第1列中，以此类推，目标表多出的列会被插入NULL值（如果该列允许为NULL）或默认值（如果该列有默认值）。
UPDATE	支持	-
DELETE	支持	-
MERGE	支持	-
LOCK TABLE	支持	-
INSERT ALL	支持，有差异	• Oracle不支持对into_clause的表设置别名，GaussDB支持。 • into_clause指定sequence： – Oracle：首次引用nextval会生成下一个数字，但所有非首次引用的nextval都将返回相同数字。 – GaussDB：引用nextval生成的数字可以正常自增。 • Oracle设置plan_hint语句可以正常生效，GaussDB不生效。 • Oracle允许目标表的列数多于子查询结果的列数，但必须显式指定要插入的列名，以确保列数匹配。GaussDB可以省略显式指定要插入的列名，该情况下子查询结果的第1列值插入到目标表的第1列中，以此类推，目标表多出的列会被插入NULL值（如果该列允许为 NULL）或默认值（如果该列有默认值）。

表 3-71 静态 TCL SQL 语句

Oracle数据库	GaussDB数据库	差异
COMMIT	支持	-
ROLLBACK	支持	-
SAVEPOINT	支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
SET TRANSACTION	支持，有差异	GaussDB不支持NAME string语法、USE ROLLBACK SEGMENT rollback_segment语法。

表 3-72 伪列

Oracle数据库	GaussDB数据库	差异
CURRVAL and NEXTVAL	支持	-
LEVEL	不支持	-
OBJECT_VALUE	不支持	-
ROWID	支持	-
ROWNUM	支持，有差异	不推荐ROWNUM条件用于JOIN ON子句。 GaussDB中ROWNUM条件用于JOIN ON子句时在LEFT JOIN、RIGHT JOIN、FULL JOIN场景下和MERGE INTO场景下与其他数据库行为不一致，直接进行业务迁移存在风险。

表 3-73 隐式游标属性

Oracle数据库	GaussDB数据库	差异
SQL%FOUND	支持，有差异	GaussDB在commit/rollback之后不刷新隐式游标结果，Oracle会在commit/rollback之后刷新隐式游标结果。
SQL %NOTFOUND	支持，有差异	
SQL %ROWCOUNT	支持，有差异	
SQL%ISOPEN	支持，有差异	
SQL %BULK_ROWCOUNT	不支持	
SQL %BULK_EXCEPTIONS	不支持	

版权所有 © 华为云计算技术有限公司

表 3-74 显式游标语法及关键字

Oracle数据库	GaussDB数据库	差异
CURSOR cursor_name [parameter_list] RETURN return_type;	支持	-
CURSOR cursor_name [parameter_list] [RETURN return_type] IS select_statement;	支持	-
OPEN	支持	-
CLOSE	支持，有差异	GaussDB在exception内部会自动关闭，Oracle在exception内部不会自动关闭。
FETCH	支持	-
CURRENT OF CURSOR	支持	-

表 3-75 显式游标属性

Oracle数据库	GaussDB数据库
SQL%FOUND	支持
SQL%NOTFOUND	支持
SQL%ROWCOUNT	支持
SQL%ISOPEN	支持

表 3-76 游标循环

Oracle数据库	GaussDB数据库
FOR LOOP	支持，有差异。 对于FORALL + BULK COLLECT INTO场景下， INTO变量在GaussDB中仅返回单条DML语句执行结果，在Oracle中返回DML语句累计执行结果。

版权所有 © 华为云计算技术有限公司

表 3-77 自治事务支持场景

Oracle数据库	GaussDB数据库
存储过程	支持
匿名块	支持
函数	支持
Package	支持

3.2.10.6 动态 SQL

表 3-78 动态 SQL 语句执行方式

Oracle数据库	GaussDB数据库	差异
EXECUTE IMMEDIATE	支持，有差异	• GaussDB通过dynamic_sql_compat参数控制同名变量是否读取同一参数，并且检查调用存储过程时绑定参数出入参类型是否与语句参数类型一致。 • GaussDB不支持调用匿名块中部分绑定参数场景，例如匿名块中嵌套动态语句，使用表达式绑定参数，具体请参见**动态调用匿名块**章节。 • GaussDB不支持RETURNING、RETURN INTO。
OPEN FOR、FETCH 、CLOSE	支持	GaussDB通过dynamic_sql_compat参数控制变量是否读取同名参数，并且在调用存储过程时，检查出入参类型是否与语句参数类型相一致。

3.2.10.7 Trigger

表 3-79 trigger 类型

Oracle数据库	GaussDB数据库	差异
DML TRIGGER	支持，有差异	GaussDB不支持Compound DML Triggers。
SYSTEM TRIGGER	不支持	-

版权所有 © 华为云计算技术有限公司

表 3-80 create trigger

Oracle数据库	GaussDB数据库	差异
create语法： CREATE [OR REPLACE] [EDITIONABLE | NONEDITIONABLE] TRIGGER plsql_trigger_source	支持，有差异	GaussDB不支持OR REPLACE 、EDITIONABLE | NONEDITIONABLE，支持plsql_trigger_source部分行为。
plsql_trigger_source ::=语法： [schema.] trigger_name [sharing_clause] [default_collation_clause] { simple_dml_trigger | instead_of_dml_trigger | compound_dml_trigger | system_trigger }	支持，有差异	GaussDB不支持schema、sharing_clause、default_collation_clause。
simple_dml_trigger ::=语法： { BEFORE | AFTER } dml_event_clause [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [ENABLE | DISABLE] [WHEN (condition)] trigger_body	支持，有差异	GaussDB不支持referencing_clause（用from referencing_table代替）、trigger_edition_clause、trigger_ordering_clause、 ENABLE | DISABLE，支持trigger_body部分行为。 GaussDB在没有INSTEAD OF TRIGGER的视图上创建语句级BEFORE/AFTER TRIGGER时不会报错，执行DML时报错。
dml_event_clause ::=语法： { DELETE | INSERT | UPDATE [OF column [, column]...] } [OR { DELETE | INSERT | UPDATE [OF column [, column]...] }... ON [schema.] { table | view }	不支持	-

 版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
trigger_body ::=语法： { plsql_block | CALL routine_clause }	支持，有差异	GaussDB不支持plsql_block，仅支持EXECUTE PROCEDURE function_name（arguments);方式执行function，并且function需要用户定义，必须声明为不带参数并返回类型为触发器，在触发器触发时执行。
instead_of_dml_trigger ::=语法： INSTEAD OF { DELETE | INSERT | UPDATE } [OR { DELETE | INSERT | UPDATE }]... ON [NESTED TABLE nested_table_column OF] [schema.] noneditioning_view [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [ENABLE | DISABLE] trigger_body	支持，有差异	GaussDB不支持NESTED TABLE nested_table_column OF、referencing_clause、trigger_edition_clause、trigger_ordering_clause、ENABLE | DISABLE。
compound_dml_trigger ::=语法： CREATE trigger FOR dml_event_clause ON view COMPOUND TRIGGER INSTEAD OF EACH ROW IS BEGIN statement; END INSTEAD OF EACH ROW;	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
system_trigger ::=语法： { BEFORE | AFTER | INSTEAD OF } { ddl_event [OR ddl_event]... | database_event [OR database_event]... } ON { [schema.] SCHEMA | [PLUGGABLE] DATABASE } [trigger_ordering_clause] [ENABLE | DISABLE] trigger_body	不支持	-

表 3-81 alter trigger

Oracle数据库	GaussDB数据库	差异
ALTER TRIGGER [schema.] trigger_name { trigger_compile_clause | { ENABLE | DISABLE } | RENAME TO new_name | { EDITIONABLE | NONEDITIONABLE } } ;	支持，有差异	GaussDB不支持schema、trigger_compile_clause、{ ENABLE | DISABLE }、{ EDITIONABLE | NONEDITIONABLE }。

表 3-82 drop trigger

Oracle数据库	GaussDB数据库	差异
DROP TRIGGER [schema.] trigger ;	支持，有差异	GaussDB不支持schema，需要在trigger_name后面加上ON table_name。

Oracle数据库名为*_TRIGGERS的视图统计了trigger的相关信息，GaussDB相关视图与Oracle存在差异，GaussDB视图具体请参见**DB_TRIGGERS**、**ADM_TRIGGERS**、**MY_TRIGGERS**。

版权所有 © 华为云计算技术有限公司

表 3-83 Nested, Package and Standalone Subprograms 兼容性

Oracle数据库	GaussDB数据库	差异
nested subprogram（子块）	支持，有差异	GaussDB不支持重载。 GaussDB不支持定义为自治事务。 GaussDB不支持SETOF的使用，仅限一个限定符引用嵌套子程序或嵌套子程序的变量。
package subprogram	支持	-
standalone subprogram（包含 Function & Procedure）	支持	-
匿名块	支持	-

表 3-84 RETURN 语句支持情况

Oracle数据库	GaussDB数据库
Function	支持
Procedure	支持
匿名块	支持

表 3-85 Function 相关参数

Oracle数据库	GaussDB数据库	差异
DETERMINISTIC	支持，有差异	GaussDB中为IMMUTABLE。
PARALLEL_ENABLE	不支持	-
PIPELINED	不支持	-
RESULT_CACHE	不支持	-

表 3-86 参数形式支持

Oracle数据库	GaussDB数据库
IN	支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
OUT	支持
IN OUT	支持

表 3-87 CREATE 语句

Oracle数据库	GaussDB数据库	差异
CREATE FUNCTION	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持sharing_clause、仅支持部分指定function属性的子句（属性的子句仅支持 invoker_rights_clause子句）、不支持关键字[EDITIONABLE | NONEDITIONABLE]。 GaussDB的具体语法请参见**CREATE FUNCTION**章节。
CREATE LIBRARY	不支持	-
CREATE PACKAGE	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持sharing_clause、仅支持部分指定package属性的子句（属性的子句仅支持 invoker_rights_clause子句）、不支持关键字[EDITIONABLE | NONEDITIONABLE]。 GaussDB的具体语法请参见**CREATE PACKAGE**章节。
CREATE PACKAGE BODY	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持sharing_clause、不支持关键字[EDITIONABLE | NONEDITIONABLE]。 GaussDB的具体语法请参见**CREATE PACKAGE**章节。
CREATE PROCEDURE	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持sharing_clause子句、不支持关键字[EDITIONABLE | NONEDITIONABLE]。 GaussDB的具体语法请参见**CREATE PROCEDURE**章节。
CREATE TRIGGER	支持，有差异	GaussDB的具体语法请参见**CREATE TRIGGER**章节。
CREATE TYPE	支持，有差异	GaussDB不支持varray、object类型、UNDER语法。 GaussDB的具体语法请参见**CREATE TYPE**章节。
CREATE TYPE BODY	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
CREATE OR REPLACE TYPE	不支持	-

表 3-88 ALTER 语句

Oracle数据库	GaussDB数据库	差异
ALTER FUNCTION	支持，有差异	GaussDB不支持关键字[EDITIONABLE | NONEDITIONABLE]、REUSE 、SETTINGS、DEBUG。 GaussDB的具体语法请参见**ALTER FUNCTION**章节。
ALTER LIBRARY	不支持	-
ALTER PACKAGE	支持，有差异	GaussDB不支持关键字[EDITIONABLE | NONEDITIONABLE]、REUSE 、SETTINGS、DEBUG。 GaussDB的具体语法请参见**ALTER PACKAGE**章节。
ALTER PROCEDURE	支持，有差异	GaussDB不支持关键字[EDITIONABLE | NONEDITIONABLE]、REUSE 、SETTINGS、DEBUG。
ALTER TRIGGER	支持，有差异	GaussDB仅支持修改trigger名字。 GaussDB的具体语法请参见**ALTER TRIGGER**章节。
ALTER TYPE	支持，有差异	GaussDB仅支持部分语句。 GaussDB的具体语法请参见**ALTER TYPE**章节。

表 3-89 DROP 语句

Oracle数据库	GaussDB数据库	差异
DROP FUNCTION	支持	-
DROP LIBRARY	不支持	-
DROP PACKAGE	支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DROP PROCEDURE	支持	-
DROP TRIGGER	支持，有差异	GaussDB的语法与Oracle不同。 GaussDB的具体语法请参见**DROP TRIGGER**章节。
DROP TYPE	支持，有差异	GaussDB不支持关键字FORCE、VALIDATE。 GaussDB的具体语法请参见**DROP TYPE**章节。
DROP TYPE BODY	不支持	-

表 3-90 Function、Procedure、匿名块相关关键字

Oracle数据库	GaussDB数据库	差异
ACCESSIBLE BY	不支持	-
AGGREGATE	支持，有差异	• GaussDB不支持Oracle的aggregate using [schema.] implementation_type用法。 • GaussDB的具体用法请参见**CREATE AGGREGATE**章节。 语法不同，但实现功能相同。
DETERMINISTIC	支持，有差异	GaussDB仅在语法上支持关键字DETERMINISTIC，未实现功能。
PIPE ROW	不支持	-
PIPELINED	不支持	-
SQL_MACRO	不支持	-
RESTRICT_REFERENCES	不支持	-
INLINE	不支持	-

版权所有 © 华为云计算技术有限公司

表 3-91 异常处理相关关键字

Oracle数据库	GaussDB数据库	差异
EXCEPTION_INIT	支持，有差异	GaussDB不支持与系统错误码进行绑定。若在同一个存储过程或函数中通过EXCEPTION_INIT设置相同错误码，Oracle数据库不支持各自分别作为条件捕获异常，GaussDB支持此行为，详见**示例**。
Exception	支持	-
Exception Handler	支持	-
SQLCODE	支持	-
SQLERRM	支持	-

表 3-92 其他 PL/SQL 关键字

Oracle数据库	GaussDB数据库
COVERAGE	不支持
COLLATION	支持
DEPRECATE	不支持
FORALL	支持
NOCOPY	不支持
RETURNING	支持
SERIALLY_REUSABLE	不支持
SHARING	不支持
BULK COLLECT	支持

3.2.11 系统函数

兼容函数分为：单行函数、用户自定义函数、AGG函数、分析函数、对象引用函数、模型函数、OLAP函数。

3.2.11.1 单行函数

Oracle数据库	GaussDB数据库
表3-93	支持，有差异
表3-94	支持，有差异

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
表3-95	支持，有差异
字符集函数	不支持
Collation函数	不支持
表3-96	支持，有差异
表3-97	支持，有差异
表3-98	支持，有差异
表3-99	支持，有差异
集合函数	不支持
表3-100	支持
数据挖掘功能函数	不支持
表3-101	支持，有差异
表3-102	支持，有差异
表3-103	支持，有差异
表3-104	支持
表3-105	支持，有差异

表 3-93 数值函数

Oracle数据库	GaussDB数据库	差异
ABS	支持	-
ACOS	支持，有差异	当参数为BINARY_DOUBLE或BINARY_FLOAT类型且值的范围不在-1~1之间时，GaussDB报错，Oracle返回NaN。
ASIN	支持，有差异	当参数为BINARY_DOUBLE或BINARY_FLOAT类型且值的范围不在-1~1之间时，GaussDB报错，Oracle返回NaN。
ATAN	支持	-
ATAN2	支持，有差异	两个参数都为0时，GaussDB返回0，Oracle报错。
BITAND	支持，有差异	• GaussDB入参最大支持BIGINT范围，Oracle入参可支持NUMBER范围。 • 当参数包含小数时，GaussDB四舍五入取整后进行计算，而Oracle截断取整后进行计算。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
CEIL	支持	-
COS	支持，有差异	● 返回值数据类型和精度不同。 GaussDB：返回值类型是float8，结果精度保留15位有效数字。 Oracle：返回值类型是number。 ● 参数值为Infinity或-Infinity时，GaussDB报错，Oracle返回NaN。
COSH	支持，有差异	返回值数据类型和精度不同。 GaussDB：当输入是numeric时，返回numeric类型；当输入是其他类型时，返回float8类型，结果精度保留15位有效数字。 Oracle：返回值类型是number。
EXP	支持	-
FLOOR	支持	返回值类型不同： Gaussdb: ● 当其中一个输入是numeric时，返回numeric类型。 ● 其他数据类型，返回float8类型。 Oracle: ● 返回值类型是numeric。
LN	支持	返回值不同： ● Oracle返回结果保留小数点后37位，GaussDB返回结果保留小数点后15位。
LOG	支持	入参类型不同： ● GaussDB对非法入参进行了判断，如NaN 0.0 - Infinity 0 Infinity等。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
MOD	支持，有差异	• 返回类型不一致，Oracle数据库类型包括BINARY_DOUBLE，BINARY_FLOAT，NUMBER；GaussDB返回类型包括int2，int4，int8，numeric。 • 返回值不同，当入参为包含小数的字符串类型（如'2.5'）时，Oracle将其隐式转换为NUMBER类型，保留小数部分进行计算；GaussDB将其隐式转换为整数类型，小数部分四舍五入进行计算，导致结果与Oracle不一致。 • 当第一个入参为数值类型时，第二个参数必须为int、numeric类型或能够转换为numeric的类型。在a_format_version为10c，a_format_dev_version为s6时，当第一个参数为能够转为numeric的text类型时，第二个参数必须为不超过int4的数值类型或者纯数字的字符类型。
NANVL	支持，有差异	GaussDB不支持直接声明或浮点数除0得到NaN。
POWER	支持，有差异	返回值数据类型不同。 GaussDB： • 当其中一个输入是numeric时，返回numeric类型。当numeric结果小于10^16时，直接返回0。 • 其他数据类型，返回float8类型。 Oracle： 返回值类型是number。
REMAINDER	支持，有差异	返回值数据类型不同。 GaussDB： • 当一个输入是float4时，另一个是numeric时，返回float4类型。 • 当两个输入都是float4时，返回float4类型。 • 当两个输入都是float8时，返回float8类型。 • 当两个输入都是字符数值类型时，返回float8类型。 • 其他数据类型，返回numeric Oracle： 返回值类型是number。 返回结果不一致： • Oracle用ROUND取商，GaussDB用FLOOR取商，当n2/n1小数部分为0.5时两者取整方向不同，导致余数结果不一致。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
ROUND	支持，有差异	• 第一个参数n的float类型，GaussDB存在精度损失，比Oracle数据库精度低。 • 返回类型不一致。round(n, integer)形式，Oracle数据库NUMBER类型，GaussDB返回numeric类型；round(n)形式，Oracle数据库n的数据类型，GaussDB只能返回float8和numeric类型，缺少float4返回类型。 • GaussDB判断入参有null，执行框架返回null的逻辑与Oracle数据库不一致。 `SELECT round(NULL,'q');` Oracle数据库null，GaussDB报错invalid input syntax for integer: "q"。
SIGN	支持，有差异	返回值数据类型不同。 GaussDB：当输入是numeric时，返回numeric类型，否则返回float8类型。 Oracle：返回值类型是number。
SIN	支持，有差异	• 返回值数据类型和精度不同。 GaussDB：返回值类型是float8，结果精度保留15位有效数字。 Oracle：返回值类型是number。 • 参数值为Infinity或-Infinity时，GaussDB报错，Oracle返回NaN。
SINH	支持，有差异	返回值数据类型和精度不同。 GaussDB：当输入是numeric时，返回numeric类型；当输入是其他类型时，返回float8类型，结果精度保留15位有效数字。 Oracle：返回值类型是number。
SQRT	支持	-
TAN	支持，有差异	• 返回值数据类型和精度不同。 GaussDB：返回值类型是float8，结果精度保留15位有效数字。 Oracle：返回值类型是number。 • 参数值为Infinity或-Infinity时，GaussDB报错，Oracle返回NaN。
TANH	支持，有差异	返回值数据类型和精度不同。 GaussDB：当输入是numeric时，返回numeric类型；当输入是其他类型时，返回float8类型，结果精度保留15位有效数字。 Oracle：返回值类型是number。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TRUNC	支持	入参类型不同： ● GaussDB第二个入参为text类型，第一个入参须为时间类型。 返回值类型不同： ● GaussDB入参个数为1个且不是numetic的数值类型时，返回值类型为loat8类型。
WIDTH_BUCKET	支持	入参类型不同： GaussDB： ● 不支持入参为时间类型。 ● 函数最后一个入参类型不支持科学计数法。

表 3-94 返回字符值的字符函数

Oracle数据库	GaussDB数据库	差异
CHR	支持，有差异	● 输入的数字不符合现有字符集时，在JDBC下GaussDB会报错，Oracle数据库会返回乱码。 ● 输入0、256等时Oracle数据库会返回Ascii码为0的字符，GaussDB会在'\0;处截断。 ● 当入参为(-1,1)区间内的小数时，在GaussDB中，参数会被取整为0，函数输出为null；而在Oracle中，则会报错，提示参数超出范围。
CONCAT	支持，有差异	● 函数返回类型差异：GaussDB返回类型为text；而Oracle则根据输入参数的类型动态决定返回类型。 ● 对于非法字符的拼接，GaussDB不会检查字符的合法性，直接拼接所有字符；而Oracle若检测到非法字符，会忽略该字符。||操作符也存在类似的差异。 示例： -- GaussDB gaussdb=# SET a_format_version='10c'; SET gaussdb=# SET a_format_dev_version='s2'; SET gaussdb=# SELECT dump(CHR(239) || 'test') FROM dual; dump ---------- Typ=25 Len=5: 239,116,101,115,116 (1 row) -- Oracle SQL> SELECT dump(CHR(239) || 'test') FROM dual; dump(CHR(239)||'TEST') ---------- Typ=1 Len=4: 116,101,115,116

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
INITCAP	支持，有差异	返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。当入参为EMPTY_CLOB()时，GaussDB返回空字符串；Oracle返回null。
LOWER	支持，有差异	• 返回值类型不一致，Oracle数据库和输入类型一致的数据类型。 • 对时间格式上隐式转换问题，输入时间类型时，隐式转换为字符串再进行lower操作。 SELECT LOWER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle返回10-DEC-12，GaussDB返回2012-12-10 00:00:00。 • 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
LPAD	支持，有差异	• GaussDB返回类型为text，当第二个参数超过int范围时，会报错；Oracle返回类型为varchar2，第二个参数超过int范围时会根据varchar2的最大长度截断为4000个字符，并正常输出结果。 • 对于非法字符的处理，Oracle会直接填充非法字符；而GaussDB会先检查字符的合法性，如果是非法字符，则会报错。
LTRIM	支持，有差异	• 返回值类型不一致。当输入为字符数据类型时，Oracle返回VARCHAR2类型；输入为数据库创建时指定的国家字符集时，Oracle返回NVARCHAR2类型；输入为LOB类型时，Oracle返回LOB类型，GaussDB返回TEXT类型。 • 当源字符串为EMPTY_CLOB()时，Oracle返回空字符；GaussDB返回null。 • Oracle仅支持删除单个字符；GaussDB支持删除字符串。 • 当删除字符为EMPTY_CLOB()时，Oracle直接返回null；GaussDB将其视为删除空字符，返回源字符串。
NCHR	支持，有差异	• 返回值字节长度与Oracle数据库不一致。 • 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。 • 返回入参对应的字节数组时，单个字节在[0x80-0xFF]范围，会返回“?”，Oracle数据库返回“?”、或者不输出、或者会报错。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
NLS_LOWER	支持，有差异	• 返回值类型不同，输入为字符数据类型时，Oracle返回VARCHAR2类型；输入为LOB类型时，Oracle返回LOB类型，GaussDB返回TEXT类型。 • nlsparam参数Oracle数据库还可以传入除nls_sort外的其他参数种类而不报错，GaussDB只支持nls_sort。 • 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
NLS_UPPER	支持，有差异	• 返回值类型不同，输入为字符数据类型时，Oracle返回VARCHAR2类型；输入为LOB类型时，Oracle返回LOB类型，GaussDB返回TEXT类型。 • nlsparam参数Oracle数据库还可以传入除nls_sort外的其他参数种类而不报错，GaussDB只支持nls_sort。 • 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
NLSSORT	支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

REGEXP_REPLACE	支持，有差异	• GaussDB入参source_char不支持NCLOB、BLOB、XMLTYPE类型。
		• 入参match_param选项 'n' 含义有差异：GaussDB中 'n' 选项与 'm' 选项含义相同，表示采用多行模式匹配；而 Oracle 表示 (.) 能匹配 '\n' 字符，没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项，不需要指定选项。
		• 不同正则表达式匹配结果可能不一致。
		`SELECT REGEXP_REPLACE('abc01234xyz', '(.*?)(\d+)(.*)', '#', 'g') FROM DUAL;`
		Oracle报错，GaussDB返回#####xyz。
		• 在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。
		• 包含部分转义字符的正则表达式匹配结果可能不一致。
		`SELECT REGEXP_REPLACE('abcabc', '\abc', '#', 'g') FROM DUAL;`
		Oracle报错，GaussDB返回abcabc。
		• 匹配规则受aformat_regexp_match参数影响，具体影响规格请参见**字符处理函数和操作符**章节REGEXP_REPLACE函数部分。
		• 正则表达式中如果字符类[]中包含转义字符（例如\t、\n、\r、\s、\d、\w等），GaussDB会将其转义成制表符；Oracle将其视为普通字符，不进行转义。
		• 对于不完整的正则表达式（例如‘{2,’等），GaussDB会严格验证正则表达式的正确性并报错；Oracle将其视为普通字符。
		• 第四个、第五个参数的取整方式不同：GaussDB采取四舍五入处理；Oracle采取截断取整处理。
		• GaussDB不支持正则表达式[A-Z]??问号数量超过三个；Oracle支持。
		• GaussDB支持正则表达式长度超过512；Oracle不支持。
		• 当输入四个参数时，GaussDB的第四个参数不支持字符串形式的数字；Oracle支持。
		• GaussDB支持 (?:...) 非捕获组语法；Oracle不支持。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
REGEXP_SUBSTR	支持，有差异	• 匹配规则受aformat_regexp_match参数影响，具体影响规格请参见**字符处理函数和操作符**章节REGEXP_SUBSTR函数部分。 • GaussDB对于正则表达式A{1,xxx}的最大值限制在255以内。 • GaussDB中对于正则表达式 'x' 模式，会将CHR(9)制表符或CHR(10)换行符视为空白忽略；Oracle将其视为普通匹配字符处理。 • GaussDB不支持六个参数的输入。 • GaussDB支持将0~1的小数作为第三个参数，Oracle不支持。
REPLACE	支持，有差异	• GaussDB不支持XMLTYPE类型的输入，Oracle支持。 • GaussDB不支持BLOB类型的输入，Oracle支持。
RPAD	支持，有差异	• GaussDB返回类型为text，当第二个参数超过int范围时，会报错；Oracle返回类型为varchar2，第二个参数超过int范围时会根据varchar2的最大长度截断为4000个字符，并正常输出结果。 • 对于非法字符的处理，Oracle会直接填充非法字符；而GaussDB会先检查字符的合法性，如果是非法字符，则会报错。
RTRIM	支持，有差异	• 当源字符串为EMPTY_CLOB()时，Oracle返回空字符；GaussDB返回null。 • Oracle仅支持删除单个字符；GaussDB支持删除字符串。 • 当删除字符为EMPTY_CLOB()时，Oracle直接返回null；GaussDB将其视为删除空字符，返回源字符串。
SUBSTR	支持，有差异	• 当源字符串参数为EMPTY_CLOB()时，GaussDB返回null，Oracle返回空clob。 • 当第二个参数或第三个参数为带小数位的数值时，GaussDB采用四舍五入处理，Oracle采用取整处理。 • 当第二个参数初始index超过源字符串的总长度时，GaussDB报错，Oracle返回null。 • GaussDB不支持将字符串的科学表达式（如‘1E2’）作为第二个或第三个参数，Oracle支持。
TRANSLATE	支持，有差异	当源字符串为EMPTY_CLOB()时，GaussDB返回空字符串；Oracle返回null。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TRIM	支持，有差异	• 当源字符串为EMPTY_CLOB()时，Oracle返回空字符；GaussDB返回null。 • Oracle要求FROM前必须指定截取方向关键字或截取字符参数，不能省略；GaussDB支持省略FROM前所有参数。 • Oracle仅支持删除单个字符；GaussDB支持删除字符串。 • 当删除字符为EMPTY_CLOB()时，Oracle直接返回null；GaussDB将其视为删除空字符，返回源字符串。
UPPER	支持，有差异	• 返回值类型不一致，Oracle数据库和输入类型一致的数据类型，GaussDB返回TEXT类型。 • 对时间格式上隐式转换问题，输入时间类型时，隐式转换为字符串再进行upper操作。 `SELECT UPPER(TO_DATE('2012-12-10','YYYY-MM-DD'));` Oracle返回10-DEC-12，GaussDB返回2012-12-10 00:00:00。 • 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
REVERSE	支持，有差异	• Oracle不支持clob、NUMBER、DATE、TIMESTAMP、TIMESTAMPTZ、INTERVAL、float、double数据类型的输入，GaussDB支持。 • 对于多字节字符，Oracle按照字节进行反转，转换后会出现乱码；GaussDB按照字符转换。
INSTRB	支持	-

表 3-95 返回数值的字符函数

Oracle数据库	GaussDB数据库	差异
ASCII	支持，有差异	• 返回值类型不同。Oracle数据库返回类型为uint4，GaussDB为int4。 • 对于非法字符的处理，GaussDB会先检验是否为非法字符（如0x80）；Oracle则相对宽松，会返回非法字符的ASCII码。 • 当入参为EMPTY_CLOB()时，GaussDB返回0；Oracle返回null。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
INSTR	支持，有差异	• GaussDB的第三个参数类型为int；Oracle为numeric。 • GaussDB的第三、四个参数小数点的进位规则不同，GaussDB采用四舍五入处理；Oracle采用整数截断处理。 • Oracle的第四个参数因为是截断整数，当为负数和0~1的小数时，截断结果是0或负数，当为非法次数时，会报错；GaussDB不报错，直接返回0。 • GaussDB不支持json类型作为源参数。
LENGTH	支持，有差异	返回值类型不同，Oracle返回numeric，GaussDB返回int。
LENGTHB	支持，有差异	GaussDB仅支持文本字符串类型入参，不接受BLOB类型；Oracle支持BLOB类型入参。
REGEXP_COUNT	支持，有差异	• GaussDB入参source_char不支持NCLOB类型。 • 入参match_param选项 'n' 含义有差异：GaussDB中 'n' 选项与 'm' 选项含义相同，表示采用多行模式匹配；而 Oracle 表示 (.) 能匹配 '\n' 字符，没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项，不需要指定选项。 • 不同正则表达式匹配结果可能不一致。 • 在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。 • 包含部分转义字符的正则表达式匹配结果可能不一致。 • 匹配规则受aformat_regexp_match参数影响，具体影响规格请参见**字符处理函数和操作符**章节REGEXP_COUNT函数部分。 • GaussDB支持 (?:...) 非捕获组语法，Oracle不支持。 • 在x 扩展模式下，GaussDB会忽略模式中的空白字符（空格、制表符等）和注释；Oracle不支持忽略。 • 正则表达式为未定义的转义字符如'\x'等，GaussDB会报错；Oracle会将其退化为普通字符进行匹配。 • 当标记为m时，如果源字符串中存在拼接的CHR(10)，返回结果存在差异：Oracle不会统计最后一行。 • [[:blank:]]在GaussDB中可以匹配CHR(9)；Oracle无法匹配。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
REGEXP_INSTR	支持，有差异	• 匹配规则受aformat_regexp_match参数影响，具体影响规格请参见字符处理函数和操作符章节REGEXP_INSTR函数部分。 • 正则表达式差异，\s在字符类[...]中的行为：GaussDB正则引擎（基于POSIX ERE/Perl风格）支持在[]内使用\s作为空白字符类，\s = [\t\n\r\f\v]（所有空白字符）；Oracle正则引擎不支持在[]内使用\s作为"空白字符类"。 • 参数类型： – GaussDB不支持将BLOB类型作为源字符串参数；Oracle支持。 – GaussDB的第三个、第四个、第五个参数类型为int，且超过int范围会报错；Oracle为numeric。 • 取整方式：GaussDB的第三个、第四个、第五个参数采用四舍五入处理，Oracle采用截断整数处理。 • 参数个数：GaussDB不支持七个参数的输入，Oracle支持。
REGEXP_LIKE	支持，有差异	• GaussDB不支持'n'等标识参数。 • 对于错误的正则表达式，GaussDB报错，Oracle返回false。 • GaussDB的正则表达式不支持'^X{4000}$' 大于255。
LENGTHC	支持	-
DUMP	支持，有差异	• 第二个、第三个或第四个参数为null时：GaussDB直接返回null；Oracle使用默认值，第二个参数默认为10，第三个参数默认为1，第四个参数默认为全部（即读取表达式的所有字节）。 • 参数类型： – GaussDB不支持xmltype、name类型作为第一个参数；Oracle支持。 – GaussDB的第二个、第三个、第四个参数类型为int；Oracle为numeric。 • 取整方式：GaussDB的第二个、第三个、第四个参数采用四舍五入处理；Oracle采用取整处理。 • 源输入为null时：GaussDB返回null；Oracle返回'NULL'字符串。

版权所有 © 华为云计算技术有限公司

表 3-96 日期时间函数

Oracle数据库	GaussDB数据库	差异
ADD_MONTHS	支持，有差异	• 公元后到公元前，GaussDB会和Oracle数据库相差1年。 • GaussDB的计算结果范围可以到-4714年，Oracle数据库只到-4713年。 • 当输入字符类型字面量时，GaussDB会优先选择timestamp的重载函数，而Oracle只存在入参为date类型的函数。
CURRENT_DATE	支持，有差异	• GaussDB不支持nls_date_format参数设置时间显示格式。 • 开启mapping_date_to_datea时，current_date返回值为datea类型，关闭mapping_date_to_datea时，current_date返回date类型。
CURRENT_TIMESTAMP	支持，有差异	Oracle数据库参数支持范围（0 ~ 9）。 GaussDB支持范围（0 ~ 6），微秒末位零不显示。
DBTIMEZONE	支持，有差异	GaussDB不支持自带tz的timestamp类型接口的调用。
EXTRACT	支持	-
LAST_DAY	支持，有差异	GaussDB存在入参为datea类型和timestamp类型的两个重载函数，当输入字符类型字面量时，GaussDB会优先选择timestamp的重载函数，而Oracle只存在入参为date类型的函数。
LOCALTIMESTAMP	支持，有差异	Oracle数据库参数支持范围（0 ~ 9）。 GaussDB支持范围（0 ~ 6），微秒末位零不显示。
MONTHS_BETWEEN	支持，有差异	GaussDB存在入参为datea类型和timestamp类型的两个重载函数，当输入字符类型字面量时，GaussDB会优先选择timestamp的重载函数，而Oracle只存在入参为date类型的函数。
NEW_TIME	支持，有差异	GaussDB存在入参为datea类型和timestamp类型的两个重载函数，当输入字符类型字面量时，GaussDB会优先选择timestamp的重载函数，而Oracle只存在入参为date类型的函数。
NEXT_DAY	支持	GaussDB存在入参为datea类型和timestamp类型的两个重载函数，输入字符类型字面量时，GaussDB会优先选择timestamp的重载函数，而Oracle只存在入参为date类型的函数。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
NUMTODSINTERVAL	支持，有差异	GaussDB不支持dsinterval类型，暂时用interval兼容dsinterval类型。
NUMTOYMINTERVAL	支持，有差异	• GaussDB不支持yminterval类型，暂时用interval兼容yminterval类型。 • GaussDB不支持BLOB类型作为入参。 • 由于浮点数取整存在差异，导致入参为浮点数时，返回结果存在差异。 -- Oracle SQL> SELECT NUMTOYMINTERVAL(0.01, 'YEAR') FROM DUAL; NUMTOYMINTERVAL(0.01,'YEAR') --- +000000000-01 -- GaussDB gaussdb=# SET a_format_version='10c'; SET gaussdb=# SET a_format_dev_version='s2'; SET gaussdb=# SELECT NUMTOYMINTERVAL(0.01, 'YEAR') FROM DUAL; numtoyminterval --- 00:00:00 (1 row)
SESSIONTIMEZONE	支持，有差异	• 赋值语法差异。GaussDB为set session time zone 8。Oracle为alter session set time_zone= '+08:00'。 • 默认值差异。GaussDB为时区名称形式如:PRC。Oracle为偏移量形式，如：+08:00。
SYS_EXTRACT_UTC	支持，有差异	当入参时间值落在夏令时跳变间隙（DST gap）中时，Oracle数据库报错ORA-01878，GaussDB不报错，以跳变前的时区偏移量为基准计算UTC时间。例如：悉尼时区Australia/Sydney在2024-10-06 02:00:00处于夏令时开始间隙（时钟从02:00 AEST跳至03:00 AEDT），SYS_EXTRACT_UTC(TIMESTAMP '2024-10-06 02:00:00 Australia/Sydney')在Oracle数据库报错，在GaussDB返回2024-10-05 16:00:00（以跳变前偏移量UTC+10计算）。
SYSDATE	支持，有差异	返回值类型不一致。GaussDB中关闭mapping_date_to_datea时，返回值类型为timestamp without time zone；GaussDB中开启mapping_date_to_datea时，返回值类型为datea。Oracle返回值类型为date。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
SYSTIMESTAMP	支持，有差异	• GaussDB毫秒计算只支持6位，Oracle数据库支持9位。 • GaussDB中SYSTIMESTAMP在同一SQL语句内每次调用均实时获取操作系统当前时间，因此同一语句中多次调用可能返回不同的时间值。Oracle数据库中SYSTIMESTAMP在同一SQL语句内返回相同的语句级快照时间。
TO_DSINTERVAL	支持，有差异	• GaussDB不支持dsinterval类型，暂时用interval兼容dsinterval类型。 • GaussDB不支持nlsparam参数，传入nlsparam参数会报函数不存在的错误；Oracle支持nlsparam参数但该参数已无实际作用，仅为向后兼容保留。
TO_TIMESTAMP	支持，有差异	• GaussDB毫秒计算只支持6位，Oracle数据库支持9位。当格式串指定FF7~FF9时，Oracle数据库可存储并返回对应精度的小数秒（如FF7返回7位），GaussDB的TIMESTAMP类型物理精度上限为6位，超出6位的小数秒部分直接截断（而非舍入），返回结果最多保留6位小数秒。 **说明** 此截断行为与TIMESTAMP字面量的舍入规则不同，字面量场景的舍入规则参见**数据类型**章节。 • 对YYYY格式指定输入0000年时：GaussDB中会被修改为当前年份；Oracle中会报错。 • 对于fmt入参中包含的文本分隔符"text"时：GaussDB中不会对输入进行文本分隔符的校验；Oracle中会对输入进行文本分隔符的校验。 • GaussDB中对于fmt入参中包含DAY/DY等星期格式时，可能返回未知结果。 • GaussDB中小数部分值为0时，不显示小数位；Oracle中显示9位小数位0。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TO_TIMESTAMP_TZ	支持，有差异	• GaussDB的timestamptz等价于Oracle的timestampwithlocaltimezone，缺少Oracle对应的timestamptz类型。nls_date_language只支持ENGLISH和AMERICAN两种语言。 • GaussDB底层使用IANA时区数据库（tzdata）。对于部分国家或地区时区（如PRC），在表示远早于标准时间引入日期的时间值（如1900年以前）时，IANA数据库使用LMT（Local Mean Time，本地平均时）定义该地区的本地时间偏移，而非固定的标准时区偏移。Oracle数据库对远早日期采用固定标准时区偏移，不使用LMT。因此，对于远早日期的带时区时间戳，GaussDB与Oracle的时区偏移值可能不一致。 -- Oracle SQL> SELECT DBTIMEZONE AS "Database", SESSIONTIMEZONE AS "Session" FROM DUAL; Dat Session --- ---------- PRC PRC SQL> SELECT TO_TIMESTAMP_TZ('0001-01-01 00:00:00 +00:00', 'YYYY-MM-DD HH24:MI:SS TZH:TZM') FROM DUAL; TO_TIMESTAMP_TZ('0001-01-0100:00:00+00:00','YYYY-MM-DDHH24:MI:SSTZH:TZM') ---------- 0001-01-01 00:00:00.000000000 +00:00 -- GaussDB gaussdb=# SET a_format_version='10c'; SET gaussdb=# SET a_format_dev_version='s1'; SET gaussdb=> SET TIMEZONE = 'Asia/Beijing'; SET gaussdb=> SELECT TO_TIMESTAMP_TZ('0001-01-01 00:00:00 +00:00', 'YYYY-MM-DD HH24:MI:SS TZH:TZM') FROM DUAL; to_timestamp_tz ---------- 0001-01-01 08:00:00+08 (1 row) gaussdb=> SET TIMEZONE = 'PRC'; SET gaussdb=# SELECT TO_TIMESTAMP_TZ('0001-01-01 00:00:00 +00:00', 'YYYY-MM-DD HH24:MI:SS TZH:TZM') FROM DUAL; to_timestamp_tz ---------- 0001-01-01 08:05:43+08:05:43 (1 row)

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TO_YMINTERVAL	支持，有差异	GaussDB不支持yminterval类型，暂时用interval兼容yminterval类型。
TRUNC	支持，有差异	GaussDB返回的类型与第一个入参的类型保持一致，Oracle始终返回date类型，另外支持指定的format也有区别，具体支持的列表详见**时间和日期处理函数和操作符**。
TZ_OFFSET	支持，有差异	• 当接收一个时区名称作为入参时，GaussDB支持的时区名称的类型少于Oracle。例如GaussDB不支持时区缩写（如PST、CST、EST等），请使用IANA时区区域名称（如US/Pacific）替代。 • 入参为时区偏移量格式（如'+hh:mi'或'-hh:mi'）时，当偏移量为零（如'-00:00'），Oracle数据库会将输出规范化为+00:00，GaussDB保留输入的原始符号，返回-00:00。 • 当入参仅包含空白字符（如空格）时，GaussDB返回空，Oracle数据库报错。
ROUND	支持，有差异	返回值类型不一致。GaussDB中关闭mapping_date_to_datea时，返回值类型为timestamp without time zone；GaussDB中开启mapping_date_to_datea时，返回值类型为datea。Oracle返回值类型为date。 GaussDB入参可以为字符类型，而Oracle不支持字符类型作为round函数入参。 第二个入参在指定单位为由ISO 8601标准定义的包含日历周的年份，例如iyyy、iy时，Oracle和GaussDB部分年份返回结果存在差异。

表 3-97 通用比较函数

Oracle数据库	GaussDB数据库	差异
GREATEST	支持，有差异	• GaussDB不支持NLS_SORT参数指定的比较方式，只支持二进制比较。 • GaussDB不支持多语种的表达式。
LEAST	支持，有差异	• GaussDB不支持NLS_SORT参数指定的比较方式，只支持二进制比较。 • GaussDB不支持多语种的表达式。

版权所有 © 华为云计算技术有限公司

表 3-98 转换函数

Oracle数据库	GaussDB数据库	差异
ASCIISTR	支持	-
CAST	支持，有差异	• GaussDB不支持multiset子句。 • GaussDB不支持nlsparam参数。 • 当进行CAST(expr AS NUMBER, fmt)操作时，GaussDB要求输入数据中的逗号个数需不小于fmt中的逗号个数，否则会进行报错；Oracle中会对fmt中多余的逗号进行忽略。 -- Oracle SELECT CAST('10' AS NUMBER, '99,99,99') FROM DUAL; -- 10 -- GaussDB gaussdb=# SET a_format_version='10c'; SET gaussdb=# SET a_format_dev_version='s1'; SET gaussdb=# SELECT CAST('10' AS NUMBER, '99,99,99') FROM DUAL; ERROR: surplus fmt. CONTEXT: referenced column: numeric
HEXTORAW	支持	-

RAWTOHEX	支持，有差异	当函数入参为数值类型（包括整型、浮点型等）、时间类型（DATEA、TIMESATMP等）时，函数返回结果不一致。
		-- Oracle
		SQL> SELECT RAWTOHEX(1), RAWTOHEX(1.12) FROM DUAL;
		RAWT RAWTOH
		---- ------
		C102 C1020D
		
		SQL> SELECT RAWTOHEX(TO_DATE('1970-01-01', 'YYYY-MM-DD')) FROM DUAL;
		RAWTOHEX(TO_DATE
		---------------- 	
		B207010100000000
		
		-- GaussDB
		gaussdb=# SELECT RAWTOHEX(1), RAWTOHEX(1.12) FROM DUAL;
		rawtohex | rawtohex
		----------+----------
		31 | 312e3132
		(1 row)
		
		gaussdb=# SELECT RAWTOHEX(TO_DATE('1970-01-01', 'YYYY-MM-DD')) FROM DUAL;
		rawtohex
		-------------------- 	
		30312d4a414e2d3730
		(1 row)

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

TO_BINARY_DOUBLE	支持，有差异	● GaussDB不支持nlsparam参数。
		● GaussDB中字面量输入按照IEEE 754 浮点标准解析，能正确返回目标类型范围内的值；Oracle中的数值字面量（无f/d后缀）先经过NUMBER类型中转，受到NUMBER类型范围约束（1E-130~ 1E+126）和字符串解析长度限制，超出上限会报错，超出下限会返回0。
		-- Oracle
		SELECT TO_BINARY_DOUBLE(1e+400) FROM DUAL; -- 字面量阶段超NUMBER范围
		-- numeric overflow
		SELECT TO_BINARY_DOUBLE(LPAD('9', 4000, '9')) FROM DUAL;
		-- 字符串解析阶段超长度
		invalid number
		SELECT TO_BINARY_DOUBLE(-2.2250738585072014E-308) FROM DUAL;
		-- 0
		
		-- GaussDB
		gaussdb=# SET a_format_version='10c';
		SET
		gaussdb=# SET a_format_dev_version='s2';
		SET
		gaussdb=# SELECT TO_BINARY_DOUBLE(1e+400) FROM DUAL;
		to_binary_double
		------------------ 	
		Infinity
		(1 row)
		
		gaussdb=# SELECT TO_BINARY_DOUBLE(LPAD('9', 4000, '9')) FROM DUAL;
		to_binary_double
		------------------ 	
		Infinity
		(1 row)
		
		gaussdb=# SELECT TO_BINARY_DOUBLE(-2.2250738585072014E-308) FROM DUAL;
		to_binary_double
		----------------------- 	
		-2.2250738585072e-308
		(1 row)
		● GaussDB与Oracle的fmt选项差异点描述：
		1. GaussDB不支持fmt格式：C
		2. GaussDB不支持fmt格式：EEEE
		3. 对于fmt格式9解析非数字字符时：GaussDB会进行跳过；Oracle会报错。
		● 无fmt参数时，GaussDB要求字符串严格符合浮点数字面量格式，不符合即报错；Oracle对含数值格式相关字符（如-、+）的非标准字符串，可能截断返回有效前缀而并非报错，对含非数值格式字符的字符串报错。
		-- Oracle
		SELECT TO_BINARY_DOUBLE('1-23') FROM DUAL;
		-- 1

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
		-- GaussDB gaussdb=# SET a_format_version='10c'; SET gaussdb=# SET a_format_dev_version='s2'; SET gaussdb=# SELECT TO_BINARY_DOUBLE('1-23') FROM DUAL; ERROR: invalid input syntax for type double precision: "1-23" CONTEXT: referenced column: to_binary_double ● 如果入参中存在null值时：GaussDB直接返回null；Oracle中会正常执行函数功能。 ● TO_BINARY_DOUBLE('NAN')返回的NAN，在与0的数值运算中，GaussDB与Oracle存在差异。 – Oracle中，NAN与0计算的结果恒为NAN。 – GaussDB中，NAN与0的计算符合0值计算的正常算术逻辑。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

TO_BINARY_FLOAT	支持，有差异	● GaussDB不支持nlsparam参数。
		● GaussDB中，字面量输入按照IEEE 754浮点标准解析，可以正确返回目标类型范围内的值；Oracle中的数值字面量（无f/d后缀）需先经过NUMBER类型中转，受到NUMBER类型范围约束（1E-130 ~ 1E+126）和字符串解析长度限制，超出上限时会进行报错，超出下限时会返回0。
		-- Oracle
		SELECT TO_BINARY_FLOAT(1e+400) FROM DUAL; -- 字面量阶段超NUMBER范围
		-- numeric overflow
		SELECT TO_BINARY_FLOAT(LPAD('9', 4000, '9')) FROM DUAL; -- 字符串解析阶段超长度
		-- invalid number
		
		-- GaussDB
		gaussdb=# SET a_format_version='10c';
		SET
		gaussdb=# SET a_format_dev_version='s2';
		SET
		gaussdb=# SELECT TO_BINARY_FLOAT(1e+400) FROM DUAL;
		 to_binary_float
		----------------- 	
		  Infinity
		(1 row)
		
		gaussdb=# SELECT TO_BINARY_FLOAT(LPAD('9', 4000, '9')) FROM DUAL;
		 to_binary_float
		----------------- 	
		  Infinity
		(1 row)
		● GaussDB与Oracle的fmt选项差异点描述：
		1. GaussDB不支持fmt格式：C
		2. GaussDB不支持fmt格式：EEEE
		3. 对于fmt格式9解析非数字字符时：GaussDB会进行跳过；Oracle会报错。
		● 无fmt参数时，GaussDB要求字符串严格符合浮点数字面量格式，不符合即报错；Oracle对含数值格式相关字符（如-、+）的非标准字符串可能截断返回有效前缀而非报错，对含非数值格式字符的字符串报错。
		-- Oracle
		SELECT TO_BINARY_FLOAT('1-23') FROM DUAL;
		-- 1
		
		-- GaussDB
		gaussdb=# SET a_format_version='10c';
		SET
		gaussdb=# SET a_format_dev_version='s2';
		SET
		gaussdb=# SELECT TO_BINARY_FLOAT('1-23') FROM DUAL;
		ERROR: invalid input syntax for type real: "1-23"
		CONTEXT: referenced column: to_binary_float
		● 如果入参中存在null值时：GaussDB直接返回null；Oracle中会正常执行函数功能。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
		● TO_BINARY_FLOAT('NAN')返回的NAN，在与0的数值运算中，GaussDB与Oracle存在差异。 – Oracle中，NAN与0计算结果恒为NAN。 – GaussDB中，NAN与0的计算符合0值计算的正常算术逻辑。
TO_BLOB	支持，有差异	● GaussDB不支持long raw类型。 ● GaussDB不支持bfile、mime_type类型。
TO_CLOB	支持，有差异	GaussDB不支持BLOB类型作为入参，Oracle支持。

TO_DATE	支持，有差异	● 不支持多语种参数。
		● GaussDB不支持BLOB类型作为入参，Oracle支持。
		● 返回类型不一致。GaussDB中打开mapping_date_to_datea时，返回值类型为datea；GaussDB中关闭mapping_date_to_datea时，返回值类型为timestamp。
		● GaussDB中打开mapping_date_to_datea时，受控制参数nls_date_format影响；GaussDB中关闭mapping_date_to_datea时，不受控制参数nls_date_format影响。
		● 部分format格式不支持。
		● fmt = 'j'。1582年10月15日之前Oracle数据库与GaussDB输出不一致。
		● 无分割符时，不保证与Oracle数据库完全一致。如to_date('220725', 'yymmdd')，yy/rr按照固定长度4解析，会解析为2207年25月，25非法月份则会报错。
		● 如果入参中存在null值时：GaussDB直接返回null；Oracle中会正常执行函数功能。
		● 部分场景下，Oracle可忽略fmt的空格，GaussDB则严格校验。因此使用时建议入参字符串与fmt严格匹配。
		`-- Oracle`
		`SQL> SELECT TO_DATE('15-JANUARY -2024', 'DD-MONTH-YYYY') FROM DUAL;`
		`TO_DATE`
		`--------------------`
		`2024-01-15 00:00:00`
		
		`-- GaussDB（月份JANUARY后带空格）`
		`gaussdb=# SELECT TO_DATE('15-JANUARY -2024', 'DD-MONTH-YYYY') FROM DUAL;`
		`ERROR:  Invalid data exists in a string for matching date.`
		`CONTEXT:  referenced column: to_date`

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TO_MULTI_BYTE	支持	-
TO_NCHAR	支持，有差异	● GaussDB：将入参的类型转换为text。 ● Oracle：将入参的类型转换为国家字符集（national character set）。 其余差异请参见TO_CHAR函数。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TO_NUMBER	支持，有差异	GaussDB不支持NLS_PARAM参数。 GaussDB不支持CLOB类型作为入参，Oracle支持。 GaussDB与Oracle的format选项差异点描述： 1. $ GaussDB不支持该format。 2. , (comma) GaussDB：逗号可以出现在format的任意位置。 Oracle： ● 在format中，逗号只能出现在整数部分，且不能出现在数字开头；在原数据中，逗号可以在数字的开头位置。 ● 支持format中的逗号与原数据的逗号个数和位置不一致，但最后一个逗号的位置需一致。 ● 原数据和format中的连续逗号，等同于没有逗号。 ● 当原数据中没有逗号时，format的最后一个逗号后面的数字位数需与原数据相等。 3. B GaussDB未实现该功能。 4. C GaussDB不支持NLS参数。 5. G GaussDB不支持NLS参数。 6. L GaussDB不支持NLS参数。 7. U GaussDB不支持NLS参数。 8. D GaussDB不支持NLS参数。 9. PR GaussDB：等同于S，返回负数。 Oracle： ● 返回<尖括号>中的负值。 ● 返回带前导和尾随空格的正值。 ● 限制：PR格式元素只能出现在数字格式模型的最后一个位置。 10.RN | rn GaussDB未实现该功能。 11.TM| TM9 | TMe GaussDB未实现该功能。 12.V GaussDB未实现该功能。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
		13.FM ● GaussDB：当有FM时，允许format中的逗号比原数据中多，不需要严格保持一致。 ● Oracle：返回值保留前后的空格。 14.EEEE GaussDB未实现该功能。 15.XXXX 对于负数入参的处理，GaussDB和Oracle返回结果不一致。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

TO_CHAR	支持，有差异	• GaussDB不支持nlsparam参数。
		• fmt'5'未在Oracle数据库文档中，未进行适配。
		• 对于入参为字符类型时：Oracle的返回值类型与入参类型保持一致；GaussDB则统一返回text类型的返回值。
		• 入参类型为INTERVAL类型且未显式指定fmt时，GaussDB返回结果与Oracle存在格式差异。
		-- GaussDB
		gaussdb=# SELECT TO_CHAR(INTERVAL '2 12:30:45' DAY TO SECOND) FROM DUAL;
		to_char
		------------------ 	
		2.52135416666667
		(1 row)
		
		-- Oracle
		SQL> SELECT TO_CHAR(INTERVAL '2 12:30:45' DAY TO SECOND) FROM DUAL;
		TO_CHAR(INTERVAL'21
		-------------------- 	
		+02 12:30:45.000000
		• 对于零值占位符处理逻辑不同，Oracle中严格按格式模板保留指定位数的零；GaussDB中则固定返回一位零。
		-- GaussDB
		gaussdb=# SELECT TO_CHAR(0, '999V99') FROM DUAL;
		to_char
		--------- 	
		0
		(1 row)
		
		-- Oracle
		SQL> SELECT TO_CHAR(0, '999V99') FROM DUAL;
		TO_CHA
		------ 	
		00
		• 如果入参中存在null值时：GaussDB直接返回null；Oracle中会正常执行函数功能。
		• GaussDB不支持fmt格式：TZH、TZM。
		-- GaussDB
		gaussdb=# set a_format_version = '10c';
		SET
		gaussdb=# set a_format_dev_version = 's7';
		SET
		gaussdb=# SELECT TO_CHAR(TIMESTAMP '2024-06-15 14:30:00 +08:00', 'TZH:TZM') FROM DUAL;
		ERROR: invalid date format model
		CONTEXT: referenced column: to_char
		
		-- Oracle
		SELECT TO_CHAR(TIMESTAMP '2024-06-15 14:30:00 +08:00', 'TZH:TZM') FROM DUAL;
		-- +08:00
		• 如果数值的整数部分超过fmt格式表示整数的位数，会使用#号填充整改格式宽度，GaussDB和Oracle的返回结果存在差异。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异

		``
		-- GaussDB
		gaussdb=# SET a_format_version='10c';
		SET
		gaussdb=# SET a_format_dev_version='s2';
		SET
		gaussdb=# SELECT TO_NCHAR(1000, '99.99MI') FROM DUAL;
		 to_nchar
		---------- 	
		 #####
		(1 row)
		
		-- Oracle
		SELECT TO_CHAR(-1000, '$99.99MI') FROM DUAL;
		-- #######
		``
		● 如果数值为0，fmt格式B与S、D、.（点）搭配使用时，GaussDB和Oracle的返回结果存在差异。
		``
		-- GaussDB
		gaussdb=# SET a_format_version='10c';
		SET
		gaussdb=# SET a_format_dev_version='s1';
		SET
		gaussdb=# SELECT TO_CHAR(0, 'B9999.99S') FROM DUAL;
		 to_char
		----------- 	
		 . +
		(1 row)
		
		-- Oracle
		SELECT TO_CHAR(0, 'B9999.99S') FROM DUAL;
		-- 全部替换为空格
		``
		● format格式中，不支持SP后缀（将数值转换为英文单词拼写），指定该后缀后不生效，数值仍以数字形式输出。

TO_SINGLE_BYTE	支持	-
TREAT	支持，有差异	GaussDB不支持使用“.”操作符取值，不支持转化为object类型。
UNISTR	支持，有差异	GaussDB只支持UTF-8编码，Oracle数据库支持UTF-8和UTF-16编码。

表 3-99 大对象函数

Oracle数据库	GaussDB数据库	差异
EMPTY_BLOB	支持	-
EMPTY_CLOB	支持，有差异	GaussDB的CLOB类型不支持Oracle数据库中的定位器概念。

版权所有 © 华为云计算技术有限公司

表 3-100 层次函数

Oracle数据库	GaussDB数据库	差异
SYS_CONNECT_BY_PATH	支持，有差异	GaussDB中第一个入参指定的列的类型仅支持CHAR/VARCHAR/NVARCHAR2/TEXT/INT1/INT2/INT4/INT8/FLOAT4/FLOAT8/NUMERIC类型。当前该函数col输入不支持表达式输入，列内容与separator分隔符重复不支持报错。
CONNECT_BY_ROOT	支持，有差异	在GaussDB中，对于CONNECT_BY_ROOT运算存在以下语法限制：若其后指定的列名为非保留关键字时，则必须通过括号将列名进行包裹（如CONNECT_BY_ROOT(column_name)）才能为该表达式赋予别名，直接使用无括号形式会导致语法错误。如：CONNECT_BY_ROOT unreserved_keyword_column_name AS alias_name需要修改为CONNECT_BY_ROOT (unreserved_keyword_column_name) AS alias_name。

表 3-101 XML 类型函数

Oracle数据库	GaussDB数据库	差异
EXISTSNODE	支持，有差异	在入参有命名空间时，xpath和命名空间都需要定义别名。
EXTRACTVALUE	支持，有差异	目前仅支持xpath1.0版本。
SYS_XMLAGG	支持，有差异	xmlagg的别名，可使用xmlagg代替。
XMLAGG	支持，有差异	具体差异如说明所示。
XMLCOMMENT	支持，有差异	
XMLCONCAT	支持，有差异	
XMLELEMENT	支持，有差异	xmlelement和xmlattributes的name字段赋值为NULL时，行为与Oracle不一致。 ● xmlelement的name字段赋值为NULL时，结果显示name信息为空，且不显示属性信息。 ● xmlattributes的name字段赋值为NULL时，不显示属性信息。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
XMLEXISTS	支持，有差异	GaussDB入参为xml类型。
XMLFOREST	支持，有差异	GaussDB返回值为xml类型。 GaussDB不支持EVALNAME语法。
XMLPARSE	支持，有差异	GaussDB返回值为xml类型。 GaussDB不支持WELLFORMED语法。
XMLROOT	支持，有差异	GaussDB返回值为xml类型。
XMLSERIALIZE	支持，有差异	语法规则规则不同，GaussDB入参需要在value后加上AS type，Oracle可以省略不加。
XMLTABLE	支持，有差异	GaussDB： ● 使用XPath 1.0表达式从xml中选取数据。 ● 不支持声明默认命名空间。 ● 不支持多组输入及取别名。 ● 不支持RETURNING SEQUENCE BY REF子句和（SEQUENCE）BY REF子句。 ● XPath查询与输出功能基于libxml2库实现。该实现继承libxml2的字符处理机制，包括对XML特殊字符（如<, >, &, ", '）的转义规则。 ● 使用数字和xml文本组成的序列输入场景下，存在以下差异： – 不支持标准xpath和xml组合成序列输入。 – xml文本中支持携带版本信息。 – 形如'1e10'的科学计数法中，支持'e'或'E'与指数间存在空格。 Oracle：使用XQuery表达式从xml中选取数据。
XMLQUERY	支持，有差异	GaussDB： ● 仅支持XPath1.0语法。 ● 固定xml文本根节点为上下文空间开始寻找文本节点。 Oracle：使用XQuery表达式从xml中选取数据。
XMLCAST	支持，有差异	GaussDB支持的数据类型中，部分数据类型由于与Oracle数据库支持的数据类型本身存在差异，输出的数据格式以及精度上会有差异。
GETSTRINGVAL	支持，有差异	具体差异如说明所示。
GETCLOBVAL	支持，有差异	

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
XMLSEQUENCE	支持，有差异	

说明

- 由于表3-101中罗列的XML类型函数基于libxml2实现，因此在部分场景下与Oracle存在行为差异，主要体现在XML解析、特殊字符转义规则以及XPath查询结果格式等方面。

  以下示例展示了同一XML查询在GaussDB与Oracle中输出的差异：

  GaussDB中的ORA兼容模式下：

```
gaussdb=# SELECT * FROM xmltable('/' passing xmltype('<a>&lt; &gt; &quot; &amp; &apos;</a>'));
        column_value
-----------------------------
 <a>&lt; &gt; " &amp; '</a>+

(1 row)
```

  Oracle：

```
SQL> SELECT * FROM xmltable('/' passing xmltype('<a>&lt; &gt; &quot; &amp; &apos;</a>'));
COLUMN_VALUE
--------------------------------------------------------------------------------
<a>&lt; &gt; &quot; &amp; &apos;</a>
```

- GaussDB所依赖的libxml2库版本升级时，可能引入不兼容变更。

表 3-102 JSON 函数

Oracle数据库	GaussDB数据库	差异
JSON_ARRAY	支持，有差异	GaussDB：入参为可变参数列表([VARIADIC "any"])。 Oracle：入参更复杂，为(expr [FORMAT JSON] [, expr [FORMAT JSON]]... [JSON_on_null_clause] [JSON_returning_clause] [STRICT])。
JSON_OBJECT	支持，有差异	GAUSSDB：支持函数JSON_OBJECT([key, val[, key1, val1] ...])，key重复时，保留第一组key-value。 Oracle：入参更复杂，为(key_name VARCHAR2, value_expr [, key_name VARCHAR2, value_expr]... [JSON_on_null_clause] [JSON_returning_clause] [STRICT])。支持显式的键值对配对、独立的NULL值处理策略（如ABSENT ON NULL或NULL ON NULL）、返回类型控制以及严格模式（STRICT，防止键名重复或类型不匹配，未指定时如果存在重复key，默认全部保留）。

版权所有 © 华为云计算技术有限公司

表 3-103 编码解码函数

Oracle数据库	GaussDB数据库	差异
DECODE	支持，有差异	条件表达式比较时，如果一侧为定长字符类型，另一侧为变长字符类型，GaussDB会忽略尾部空格（与比较操作符行为一致），Oracle不会忽略尾部空格。
DUMP	支持，有差异	因存储格式不同，GaussDB中数值和时间类型的返回结果和Oracle数据库不一致。如： ● GaussDB中“SELECT dump(123);”返回“Typ=23 Len=4: 123,0,0,0”（需要先执行SET a_format_version='10c';以及SET a_format_dev_version='s2';）。 ● Oracle中“SELECT dump(123) FROM dual;”返回“Typ=2 Len=3: 194,2,24”。
ORA_HASH	支持，有差异	GaussDB中有以下行为： ● 时间类型的入参转换成字符串类型再进行hash。 ● 不支持maxbucket参数。
VSIZE	支持，有差异	● 因存储格式不同，GaussDB中数值和时间类型的返回结果和Oracle数据库不一致。如： – GaussDB中“SELECT vsize(999);”返回4（需要先执行SET a_format_version='10c';以及SET a_format_dev_version='s2';）。 – Oracle中“SELECT vsize(999) FROM dual;”返回3。 ● GaussDB不支持xmltype类型的入参。

表 3-104 空值相关的函数

Oracle数据库	GaussDB数据库	差异
COALESCE	支持	-
LNNVL	支持	-
NULLIF	支持，有差异	表达式比较时，如果一侧为定长字符类型，另一侧为变长字符类型，GaussDB会忽略尾部空格（与比较操作符行为一致），Oracle不会忽略尾部空格。
NVL	支持，有差异	第一个表达式为NULL时，GaussDB返回类型为text，Oracle返回第二个表达式的数据类型。
NVL2	支持	-

版权所有 © 华为云计算技术有限公司

表 3-105 环境和标识符函数

Oracle数据库	GaussDB数据库	差异
SYS_CONTEXT	支持，有差异	• GaussDB支持SYS_CONTEXT(namespace, attribute)，不支持通过SYS_CONTEXT(namespace, attribute, len)指定返回值的长度。 • GaussDB对不支持的参数返回NULL。 以下为不支持的参数列表： – 'action' – 'is_application_root' – 'is_application_pdb' – 'audited_cursorid' – 'authenticated_identity' – 'authentication_data' – 'authentication_method' – 'cdb_domain' – 'cdb_name' – 'client_identifier' – 'con_id' – 'con_name' – 'current_sql_length' – 'db_domain' – 'db_supplemental_log_level' – 'dblink_info' – 'drain_status' – 'entryid' – 'enterprise_identity' – 'fg_job_id' – 'global_uid' – 'identification_type' – 'instance' – 'is_dg_rolling_upgrade' – 'ldap_server_type' – 'module' – 'network_protocol' – 'nls_calendar' – 'nls_sort' – 'nls_territory' – 'oracle_home'

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
		– 'os_user' – 'platform_slash' – 'policy_invoker' – 'proxy_enterprise_identity' – 'proxy_user' – 'proxy_userid' – 'scheduler_job' – 'session_edition_id' – 'session_edition_name' – 'sessionid' – 'statementid' – 'terminal' – 'unified_audit_sessionid' – 'session_default_collation' – 'client_info' – 'bg_job_id' – 'client_program_name' – 'current_bind' – 'global_context_memory' – 'host' – 'current_sqln'
SYS_GUID	支持	-
USER	支持，有差异	返回值类型不一致。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
USERENV	支持，有差异	GaussDB对不支持的参数返回NULL。 以下为不支持的参数列表： ● 'action' ● 'is_application_root' ● 'is_application_pdb' ● 'audited_cursorid' ● 'authenticated_identity' ● 'authentication_data' ● 'authentication_method' ● 'cdb_domain' ● 'cdb_name' ● 'client_identifier' ● 'con_id' ● 'con_name' ● 'current_sql_length' ● 'db_domain' ● 'db_supplemental_log_level' ● 'dblink_info' ● 'drain_status' ● 'entryid' ● 'enterprise_identity' ● 'fg_job_id' ● 'global_uid' ● 'identification_type' ● 'is_dg_rolling_upgrade' ● 'ldap_server_type' ● 'module' ● 'network_protocol' ● 'nls_calendar' ● 'nls_sort' ● 'nls_territory' ● 'oracle_home' ● 'os_user' ● 'platform_slash' ● 'policy_invoker' ● 'proxy_enterprise_identity'

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
		• 'proxy_user' • 'proxy_userid' • 'scheduler_job' • 'session_edition_id' • 'session_edition_name' • 'sessionid' • 'statementid' • 'terminal' • 'unified_audit_sessionid' • 'session_default_collation' • 'client_info' • 'bg_job_id' • 'client_program_name' • 'current_bind' • 'global_context_memory' • 'host' • 'current_sqln'

3.2.11.2 其它函数

Oracle数据库	GaussDB数据库
表3-106	支持
表3-107	支持
对象引用函数	不支持
模型函数	不支持
OLAP函数	不支持
数据盒功能函数	不支持
关于用户定义的函数	支持

表 3-106 聚合函数

Oracle数据库	GaussDB数据库	差异
AVG	支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
CORR	支持	-
COUNT	支持	-
COVAR_POP	支持	-
COVAR_SAMP	支持	-
CUME_DIST	支持	-
DENSE_RANK	支持	-
FIRST	支持	GaussDB使用KEEP的语法，兼容Oracle此功能。
GROUPING	支持	-
LAST	支持	GaussDB使用KEEP的语法，兼容Oracle此功能。
LISTAGG	支持	-
MAX	支持	-
MEDIAN	支持	-
MIN	支持	-
PERCENT_RANK	支持	-
PERCENTILE_CONT	支持	-
RANK	支持	-
REGR_ (Linear Regression)	支持	-
STDDEV	支持	-
STDDEV_POP	支持	-
STDDEV_SAMP	支持	-
SUM	支持	-
VAR_POP	支持	-
VAR_SAMP	支持	-
VARIANCE	支持	-
WM_CONCAT	支持	-

版权所有 © 华为云计算技术有限公司

表 3-107 分析函数

Oracle数据库	GaussDB数据库	差异
FIRST_VALUE	支持	-
LAG	支持	-
LAST_VALUE	支持	-
LEAD	支持	-
NTH_VALUE	支持，有差异	• Oracle：支持FROM FIRST|LAST语法格式。 • GaussDB：不支持FROM FIRST|LAST语法格式
NTILE	支持	-
ROW_NUMBER	支持	-
RATIO_TO_REPORT	支持	-

3.2.12 系统视图

GaussDB数据库兼容了部分Oracle数据库的系统视图，兼容的详细列表如下。

更多系统视图的字段说明信息请参见**系统视图**。

表 3-108 系统视图支持列表

Oracle数据库	GaussDB数据库
ALL_ALL_TABLES	DB_ALL_TABLES
ALL_COL_PRIVS	DB_COL_PRIVS
ALL_COLL_TYPES	DB_COLL_TYPES
ALL_IND_COLUMNS	DB_IND_COLUMNS
ALL_COL_COMMENTS	DB_COL_COMMENTS
ALL_CONS_COLUMNS	DB_CONS_COLUMNS
ALL_CONSTRAINTS	DB_CONSTRAINTS
ALL_DEPENDENCIES	DB_DEPENDENCIES
ALL_DIRECTORIES	DB_DIRECTORIES
ALL_IND_EXPRESSIONS	DB_IND_EXPRESSIONS
ALL_IND_PARTITIONS	DB_IND_PARTITIONS
ALL_INDEXES	DB_INDEXES

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
ALL_IND_SUBPARTITIONS	DB_IND_SUBPARTITIONS
ALL_OBJECTS	DB_OBJECTS
ALL_PART_COL_STATISTICS	DB_PART_COL_STATISTICS
ALL_PART_KEY_COLUMNS	DB_PART_KEY_COLUMNS
ALL_PART_TABLES	DB_PART_TABLES
ALL_SCHEDULER_JOB_ARGS	DB_SCHEDULER_JOB_ARGS
ALL_SCHEDULER_PROGRAM_ARGS	DB_SCHEDULER_PROGRAM_ARGS
ALL_SEQUENCES	DB_SEQUENCES
ALL_SUBPART_KEY_COLUMNS	DB_SUBPART_KEY_COLUMNS
ALL_SYNONYMS	DB_SYNONYMS
ALL_TAB_COL_STATISTICS	DB_TAB_COL_STATISTICS
ALL_TAB_COMMENTS	DB_TAB_COMMENTS
ALL_TAB_HISTOGRAMS	DB_TAB_HISTOGRAMS
ALL_TAB_STATS_HISTORY	DB_TAB_STATS_HISTORY
ALL_TYPES	DB_TYPES
ALL_PROCEDURES	DB_PROCEDURES
ALL_SOURCE	DB_SOURCE
ALL_TAB_COLUMNS	DB_TAB_COLUMNS
ALL_TAB_PARTITIONS	DB_TAB_PARTITIONS
ALL_TAB_SUBPARTITIONS	DB_TAB_SUBPARTITIONS
ALL_TABLES	DB_TABLES
ALL_TRIGGERS	DB_TRIGGERS
ALL_USERS	DB_USERS
ALL_VIEWS	DB_VIEWS
DBA_AUDIT_OBJECT	ADM_AUDIT_OBJECT
DBA_AUDIT_SESSION	ADM_AUDIT_SESSION
DBA_AUDIT_STATEMENT	ADM_AUDIT_STATEMENT
DBA_AUDIT_TRAIL	ADM_AUDIT_TRAIL
DBA_COL_COMMENTS	ADM_COL_COMMENTS

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
DBA_COL_PRIVS	ADM_COL_PRIVS
DBA_COLL_TYPES	ADM_COLL_TYPES
DBA_ARGUMENTS	ADM_ARGUMENTS
DBA_CONSTRAINTS	ADM_CONSTRAINTS
DBA_DATA_FILES	ADM_DATA_FILES
DBA_CONS_COLUMNS	ADM_CONS_COLUMNS
DBA_DEPENDENCIES	ADM_DEPENDENCIES
DBA_DIRECTORIES	ADM_DIRECTORIES
DBA_PART_COL_STATISTICS	ADM_PART_COL_STATISTICS
DBA_PART_TABLES	ADM_PART_TABLES
DBA_ROLE_PRIVS	ADM_ROLE_PRIVS
DBA_ROLES	ADM_ROLES
DBA_SCHEDULER_JOB_ARGS	ADM_SCHEDULER_JOB_ARGS
DBA_SCHEDULER_PROGRAMS	ADM_SCHEDULER_PROGRAMS
DBA_SCHEDULER_PROGRAM_ARGS	ADM_SCHEDULER_PROGRAM_ARGS
DBA_HIST_SNAPSHOT	ADM_HIST_SNAPSHOT
DBA_HIST_SQL_PLAN	ADM_HIST_SQL_PLAN
DBA_HIST_SQLSTAT	ADM_HIST_SQLSTAT
DBA_HIST_SQLTEXT	ADM_HIST_SQLTEXT
DBA_ILMDATAMOVEMENTPOLICIES	GS_ADM_ILMDATAMOVEMENTPOLICIES
DBA_ILMEVALUATIONDETAILS	GS_ADM_ILMEVALUATIONDETAILS
DBA_ILMOBJECTS	GS_ADM_ILMOBJECTS
DBA_ILMPARAMETERS	GS_ADM_ILMPARAMETERS
DBA_ILMPOLICIES	GS_ADM_ILMPOLICIES
DBA_ILMRESULTS	GS_ADM_ILMRESULTS
DBA_ILMTASKS	GS_ADM_ILMTASKS
DBA_IND_COLUMNS	ADM_IND_COLUMNS
DBA_IND_EXPRESSIONS	ADM_IND_EXPRESSIONS

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
DBA_IND_PARTITIONS	ADM_IND_PARTITIONS
DBA_INDEXES	ADM_INDEXES
DBA_OBJECTS	ADM_OBJECTS
DBA_PART_INDEXES	ADM_PART_INDEXES
DBA_PROCEDURES	ADM_PROCEDURES
DBA_SCHEDULER_JOBS	ADM_SCHEDULER_JOBS
DBA_SCHEDULER_RUNNING_JOBS	ADM_SCHEDULER_RUNNING_JOBS
DBA_SEGMENTS	ADM_SEGMENTS
DBA_SEQUENCES	ADM_SEQUENCES
DBA_SOURCE	ADM_SOURCE
DBA_IND_SUBPARTITIONS	ADM_IND_SUBPARTITIONS
DBA_SUBPART_KEY_COLUMNS	ADM_SUBPART_KEY_COLUMNS
DBA_SYS_PRIVS	ADM_SYS_PRIVS
DBA_TAB_COL_STATISTICS	ADM_TAB_COL_STATISTICS
DBA_TAB_HISTOGRAMS	ADM_TAB_HISTOGRAMS
DBA_TAB_STATISTICS	ADM_TAB_STATISTICS
DBA_TAB_STATS_HISTORY	ADM_TAB_STATS_HISTORY
DBA_TABLESPACES	ADM_TABLESPACES
DBA_TYPES	ADM_TYPES
DBA_USERS	ADM_USERS
DBA_SYNONYMS	ADM_SYNONYMS
DBA_TAB_COLS	ADM_TAB_COLS
DBA_TAB_COLUMNS	ADM_TAB_COLUMNS
DBA_TAB_COMMENTS	ADM_TAB_COMMENTS
DBA_TABLES	ADM_TABLES
DBA_TAB_PARTITIONS	ADM_TAB_PARTITIONS
DBA_TAB_SUBPARTITIONS	ADM_TAB_SUBPARTITIONS
DBA_TRIGGERS	ADM_TRIGGERS
DBA_TYPE_ATTRS	ADM_TYPE_ATTRS

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
DBA_VIEWS	ADM_VIEWS
ROLE_ROLE_PRIVS	ROLE_ROLE_PRIVS
ROLE_SYS_PRIVS	ROLE_SYS_PRIVS
ROLE_TAB_PRIVS	ROLE_TAB_PRIVS
USER_COL_COMMENTS	MY_COL_COMMENTS
USER_COL_PRIVS	MY_COL_PRIVS
USER_COLL_TYPES	MY_COLL_TYPES
USER_CONSTRAINTS	MY_CONSTRAINTS
USER_DEPENDENCIES	MY_DEPENDENCIES
DICT	DICT
DICTIONARY	DICTIONARY
DUAL	DUAL
NLS_DATABASE_PARAMETERS	NLS_DATABASE_PARAMETERS
NLS_INSTANCE_PARAMETERS	NLS_INSTANCE_PARAMETERS
PLAN_TABLE	PLAN_TABLE
USER_ILMDATAMOVEMENTPOLICIES	GS_MY_ILMDATAMOVEMENTPOLICIES
USER_ILMEVALUATIONDETAILS	GS_MY_ILMEVALUATIONDETAILS
USER_ILMOBJECTS	GS_MY_ILMOBJECTS
USER_ILMPOLICIES	GS_MY_ILMPOLICIES
USER_ILMRESULTS	GS_MY_ILMRESULTS
USER_ILMTASKS	GS_MY_ILMTASKS
USER_IND_COLUMNS	MY_IND_COLUMNS
USER_IND_EXPRESSIONS	MY_IND_EXPRESSIONS
USER_IND_PARTITIONS	MY_IND_PARTITIONS
USER_IND_SUBPARTITIONS	MY_IND_SUBPARTITIONS
USER_INDEXES	MY_INDEXES
USER_JOBS	MY_JOBS
USER_OBJECTS	MY_OBJECTS

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
USER_PART_COL_STATISTICS	MY_PART_COL_STATISTICS
USER_PART_INDEXES	MY_PART_INDEXES
USER_PART_TABLES	MY_PART_TABLES
USER_PROCEDURES	MY_PROCEDURES
USER_SCHEDULER_JOB_ARGS	MY_SCHEDULER_JOB_ARGS
USER_SCHEDULER_PROGRAM_ARGS	MY_SCHEDULER_PROGRAM_ARGS
USER_SEQUENCES	MY_SEQUENCES
USER_SOURCE	MY_SOURCE
USER_SUBPART_KEY_COLUMNS	MY_SUBPART_KEY_COLUMNS
USER_SYNONYMS	MY_SYNONYMS
USER_SYS_PRIVS	MY_SYS_PRIVS
USER_TAB_COL_STATISTICS	MY_TAB_COL_STATISTICS
USER_TAB_COLUMNS	MY_TAB_COLUMNS
USER_TAB_COMMENTS	MY_TAB_COMMENTS
USER_TAB_HISTOGRAMS	MY_TAB_HISTOGRAMS
USER_TAB_PARTITIONS	MY_TAB_PARTITIONS
USER_TAB_STATISTICS	MY_TAB_STATISTICS
USER_TAB_STATS_HISTORY	MY_TAB_STATS_HISTORY
USER_TABLES	MY_TABLES
USER_TABLESPACES	MY_TABLESPACES
USER_TRIGGERS	MY_TRIGGERS
USER_TYPE_ATTRS	MY_TYPE_ATTRS
USER_TYPES	MY_TYPES
USER_VIEWS	MY_VIEWS
V$NLS_PARAMETERS	V$NLS_PARAMETERS
V$SESSION_WAIT	V$SESSION_WAIT
V$SYSSTAT	V$SYSSTAT
V$SYSTEM_EVENT	V$SYSTEM_EVENT
V$VERSION	V$VERSION

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
V$INSTANCE	V_INSTANCE
GV$INSTANCE	GV_INSTANCE
V$MYSTAT	V_MYSTAT
V$SESSION	V_SESSION
GV$SESSION	GV_SESSION
V$SESSION_LONGOPS	DV_SESSION_LONGOPS
V$SESSION	DV_SESSIONS
ALL_ARGUMENTS	DB_ARGUMENTS
USER_CONS_COLUMNS	MY_CONS_COLUMNS
USER_PART_KEY_COLUMNS	MY_PART_KEY_COLUMNS
USER_ROLE_PRIVS	MY_ROLE_PRIVS
DBA_TAB_PRIVS	ADM_TAB_PRIVS
USER_SCHEDULER_JOBS	MY_SCHEDULER_JOBS
V$LOCK	V$LOCK
V$DBLINK	V$DBLINK
V$GLOBAL_TRANSACTION	V$GLOBAL_TRANSACTION
V$OPEN_CURSOR	V$OPEN_CURSOR
V$GLOBAL_OPEN_CURSOR	V$GLOBAL_OPEN_CURSOR
ALL_TAB_PRIVS	DB_TAB_PRIVS
ALL_TAB_MODIFICATIONS	DB_TAB_MODIFICATIONS
USER_TAB_MODIFICATIONS	MY_TAB_MODIFICATIONS
USER_AUDIT_TRAIL	MY_AUDIT_TRAIL

3.2.13 高级包

GaussDB数据库兼容了部分Oracle数据库的高级包，兼容的详细列表如下。

表 3-109 高级包支持列表

Oracle数据库	GaussDB数据库	差异
表3-110	DBE_LOB	GaussDB具体用法请参见DBE_LOB。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
表3-111	DBE_RANDOM	GaussDB具体用法请参见DBE_RANDOM。
表3-112	DBE_OUTPUT	GaussDB具体用法请参见DBE_OUTPUT。
表3-113	DBE_RAW	GaussDB具体用法请参见DBE_RAW。
表3-114	DBE_SCHEDULER	GaussDB具体用法请参见DBE_SCHEDULER。
表3-115	DBE_UTILITY	GaussDB具体用法请参见DBE_UTILITY。
表3-116	DBE_SQL	GaussDB具体用法请参见 DBE_SQL。
表3-118	DBE_FILE	GaussDB具体用法请参见DBE_FILE。
表3-119	DBE_SESSION	GaussDB具体用法请参见DBE_SESSION。
表3-120	DBE_MATCH	GaussDB具体用法请参见DBE_MATCH。
表3-121	DBE_APPLICATION_INFO	GaussDB具体用法请参见DBE_APPLICATION_INFO。
表3-122	DBE_XMLDOM	GaussDB中具体请参见DBE_XMLDOM。
表3-123	DBE_XMLPARSER	GaussDB中具体请参见DBE_XMLPARSER。
表3-124	DBE_ILM	GaussDB中具体请参见 DBE_ILM。
表3-125	DBE_ILM_ADMIN	GaussDB中具体请参见DBE_ILM_ADMIN。
表3-126	DBE_COMPRESSION	GaussDB中具体请参见DBE_COMPRESSION。
表3-127	DBE_HEAT_MAP	GaussDB中具体请参见DBE_HEAT_MAP。
表3-128	DBE_DESCRIBE	GaussDB中具体请参见DBE_DESCRIBE。
表3-130	DBE_XMLGEN	GaussDB中具体请参见DBE_XMLGEN。
表3-129	DBE_STATS	GaussDB中具体请参见DBE_STATS。
表3-131	DBE_OBFUSCATION_TOOLKIT	GaussDB中具体请参见DBE_OBFUSCATION_TOOLKIT。

版权所有 © 华为云计算技术有限公司

表 3-110 DBMS_LOB 兼容性说明

Oracle数据库	GaussDB数据库	差异
APPEND Procedures	APPEND Procedures	-
CLOB2FILE Procedure	不支持	-
CLOSE Procedure	BFILECLOSE Procedure	GaussDB：参数类型为BFILE，不存在函数重载。 Oracle：该过程存在3个重载，3个重载的参数lob_loc、lob_loc和file_loc的类型分别为BLOB、CLOB CHARACTER SET ANY_CS和BFILE。
COMPARE Functions	COMPARE Functions	GaussDB：存在3个重载函数，对于第三个参数（len）均为BIGINT。 Oracle：存在3个重载函数，对于第三个参数（amount）均为INTEGER。
CONVERTTOBLOB Procedure	LOB_CONVERTTOBLOB Procedure	GaussDB：该过程共有5个参数，且第3、4、5个参数类型为BIGINT。 Oracle：该过程共有8个参数，在GaussDB所有参数的基础上增加了blob_csid、lang_context和warning3个参数，类型分别为NUMBER、INTEGER和INTEGER，且第3、4、5个参数类型为INTEGER。
CONVERTTOCLOB Procedure	LOB_CONVERTTOCLOB Procedure	GaussDB：该过程共有5个参数。第3、4、5个参数类型为BIGINT。 Oracle：该过程共有8个参数。第3、4、5个参数类型为INTEGER。Oracle的该过程在GaussDB所有参数的基础上增加了3个参数:blob_csid、lang_context和warning，参数类型分别为NUMBER、INTEGER和INTEGER。
COPY Procedures	LOB_COPY Functions	-
COPY_DBFS_LINK Procedures	不支持	-
COPY_FROM_DBFS_LINK	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
CREATETEMPORARY Procedures	CREATE_TEMPORARY Procedures	GaussDB：该过程存在2个重载。第一个重载过程的第一个参数（lob_loc）为BLOB，第二个重载过程的第一个参数（lob_loc）为CLOB；两个重载过程的第三个参数（dur）为INTEGER，默认值为10。 Oracle：该过程存在2个重载。第一个重载过程的第一个参数（lob_loc）为BLOB，第二个重载过程的第一个参数（lob_loc）为CLOB；两个重载过程的第三个参数（dur）的参数类型为PLS_INTEGER，第一个重载过程的dur默认值为DBMS_LOB.SESSION，第二个重载过程的dur默认值为10。
DBFS_LINK_GENERATE_PATH Functions	不支持	-
ERASE Procedures	LOB_ERASE Procedures	-
FILECLOSE Procedure	不支持	-
FILECLOSEALL Procedure	不支持	-
FILEEXISTS Function	不支持	-
FILEGETNAME Procedure	不支持	-
FILEISOPEN Function	不支持	-
FILEOPEN Procedure	不支持	-
FRAGMENT_DELETE Procedure	不支持	-
FRAGMENT_INSERT Procedures	不支持	-
FRAGMENT_MOVE Procedure	不支持	-
FRAGMENT_REPLACE Procedures	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
FREETEMPORARY Procedures	不支持	-
GET_DBFS_LINK Functions	不支持	-
GET_DBFS_LINK_STATE Procedures	不支持	-
GETCHUNKSIZE Functions	GETCHUNKSIZE Functions	-
GETCONTENTTYPE Functions	不支持	-
GETLENGTH Functions	不支持	-
GETOPTIONS Functions	不支持	-
GET_STORAGE_LIMIT Function	不支持	-
INSTR Functions	MATCH Functions	GaussDB：存在3个重载函数。3个重载函数的第三、四个参数均为BIGINT。 Oracle：存在3个重载函数。3个重载函数的第三、四个参数均为INTEGER。
ISOPEN Functions	不支持	-
ISREMOTE Function	不支持	-
ISSECUREFILE Function	不支持	-
ISTEMPORARY Functions	不支持	-
LOADBLOBFROMFILE Procedure	LOADBLOBFROMFILE Procedure	-
LOADCLOBFROMFILE Procedure	LOADCLOBFROMFILE Procedure	-
LOADFROMFILE Procedure	LOADFROMFILE Procedure	-
MOVE_TO_DBFS_LINK Procedures	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
OPEN Procedures	BFILEOPEN Procedure	GaussDB：该过程不存在重载。第一个参数（bfile）类型为DBE_LOB.BFILE，第二个参数（open_mode）类型为TEXT，且只支持read模式。 Oracle：该过程存在3个重载。第一个重载过程的第一个参数（lob_loc）类型为NOCOPY BLOB，第二个参数（openmode）类型为BINARY_INTEGER；第二个重载过程的第一个参数（lob_loc）类型为NOCOPY CLOB CHARACTER SET ANY_CS，第二个参数（openmode）类型为BINARY_INTEGER；第三个重载过程的第一个参数（file_loc）类型为NOCOPY BFILE,第二个参数（openmode）类型为BINARY_INTEGER，且只能为file_readonly。
READ Procedures	READ Procedures	GaussDB：该过程存在2个重载。 Oracle：该过程存在3个重载。其中前两个重载与GaussDB无差异，第三个过程重载包括4个参数：file_loc、amount、offset和buffer，其类型分别为BFILE、NOCOPY INTEGER、INTEGER和RAW。
SET_DBFS_LINK Procedures	不支持	-
SETCONTENTTYPE Procedure	不支持	-
SETOPTIONS Procedures	不支持	-
SUBSTR Functions	LOB_SUBSTR Functions	-
TRIM Procedures	STRIP Functions	GaussDB：该过程存在2个重载。两个重载过程的第二个参数（newlen）均为BIGINT。 Oracle：该过程存在2个重载。两个重载过程的第二个参数（newlen）均为INTEGER。
WRITE Procedures	WRITE Functions	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
WRITEAPPEND Procedures	WRITEAPPEND Functions	-

表 3-111 DBMS_RANDOM 兼容性说明

Oracle数据库	GaussDB数据库	差异
INITIALIZE Procedure	不支持	-
NORMAL Function	不支持	-
RANDOM Function	不支持	-
SEED Procedures	DBE_RANDOM.SET_SEED Function	GaussDB：该函数无重载，参数类型为INTEGER。 Oracle：该过程存在2个重载，2个重载过程的参数类型分别为VARCHAR2和BINARY_INTEGER。
STRING Function	不支持	-
TERMINATE Procedure	不支持	-
VALUE Functions	DBE_RANDOM.GET_VALUE Function	GaussDB：该函数无重载。 Oracle：存在无参数的VALUE函数重载，返回NUMBER类型。

表 3-112 DBMS_OUTPUT 兼容性说明

Oracle数据库	GaussDB数据库	差异
DISABLE Procedure	DISABLE Function	-
ENABLE Procedure	ENABLE Function	-
GET_LINE Procedure	GET_LINE Function	GaussDB：该函数无重载，首个参数（lines）数据类型为VARCHAR[]。 Oracle：该过程存在2个重载，2个重载过程的首个参数（lines）分别为CHARARR和DBMSOUTPUT_LINESARRAY。
GET_LINES Procedure	GET_LINES Function	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
NEW_LINE Procedure	NEW_LINE Function	-
PUT Procedure	PUT Function	GaussDB：当数据库服务端字符集server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时，该函数不会区分入参的数据类型，都会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出。 Oracle：当数据库服务端字符集server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时，若入参类型是NVARCHAR2，则该过程会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出；若入参为其他字符类型，则会将该字符编码视作非法字符，以占位符的形式输出。
PUT_LINE Procedure	PUT_LINE Function	GaussDB：当数据库服务端字符集server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时，该函数不会区分入参的数据类型，都会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出。 Oracle：当数据库服务端字符集server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时，若入参类型是NVARCHAR2，则该过程会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出；若入参为其他字符类型，则会将该字符编码视作非法字符，以占位符的形式输出。

表 3-113 UTL_RAW 兼容性说明

Oracle数据库	GaussDB数据库	差异
BIT_AND Function	BIT_AND Function	-
BIT_COMPLEMENT Function	BIT_COMPLEMENT Function	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
BIT_OR Function	BIT_OR Function	GaussDB：两个参数类型被定义为TEXT类型并且返回TEXT类型。 Oracle：两个参数为RAW类型并且返回RAW类型。
BIT_XOR Function	BIT_XOR Function	-
CAST_FROM_BINARY_DOUBLE Function	CAST_FROM_BINARY_DOUBLE_TO_RAW Function	-
CAST_FROM_BINARY_FLOAT Function	CAST_FROM_BINARY_FLOAT_TO_RAW Function	GaussDB：参数n为FLOAT4类型，且在转换前会对其进行范围和精度截断处理。 Oracle：参数n为FLOAT类型。
CAST_FROM_BINARY_INTEGER Function	CAST_FROM_BINARY_INTEGER_TO_RAW Function	GaussDB：参数value为BIGINT类型。 Oracle：参数value为INTEGER类型。
CAST_FROM_NUMBER Function	CAST_FROM_NUMBER_TO_RAW Function	GaussDB：参数n为NUMERIC类型。 Oracle：参数n为NUMBER类型。
CAST_TO_BINARY_DOUBLE Function	CAST_FROM_RAW_TO_BINARY_DOUBLE Function	-
CAST_TO_BINARY_FLOAT Function	CAST_FROM_RAW_TO_BINARY_FLOAT Function	GaussDB：函数返回类型为FLOAT4。 Oracle：函数返回类型为FLOAT。
CAST_TO_BINARY_INTEGER Function	CAST_FROM_RAW_TO_BINARY_INTEGER Function	GaussDB：参数endianess为INTEGER类型，函数返回类型为INTEGER。 Oracle：参数endianess为PLS_INTEGER类型，函数返回类型为BINARY_INTEGER。
CAST_TO_NUMBER Function	CAST_FROM_RAW_TO_NUMBER Function	GaussDB：函数返回类型为NUMERIC。 Oracle：函数返回类型为NUMBER。
CAST_TO_NVARCHAR2 Function	CAST_FROM_RAW_TO_NVARCHAR2 Function	-
CAST_TO_RAW Function	CAST_FROM_VARCHAR2_TO_RAW Function	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
CAST_TO_VARCHAR2 Function	CAST_TO_VARCHAR2 Function	-
COMPARE Function	COMPARE Function	GaussDB：函数返回类型为INTEGER。 Oracle：函数返回类型为NUMBER。
CONCAT Function	CONCAT Function	-
CONVERT Function	CONVERT Function	-
COPIES Function	COPIES Function	GaussDB：参数n为NUMERIC类型。 Oracle：参数n为NUMBER类型。
LENGTH Function	GET_LENGTH Function	-
OVERLAY Function	OVERLAY Function	-
REVERSE Function	REVERSE Function	-
SUBSTR Function	SUBSTR Function	GaussDB：参数lob_loc为BLOB类型；参数off_set为INTEGER类型，默认值为1；参数amount为INTEGER类型，默认值为32767。 Oracle：参数r为RAW类型，参数pos为BINARY_INTEGER类型且无默认值，参数len为BINARY_INTEGER类型，默认值为NULL。
TRANSLATE Function	TRANSLATE Function	-
TRANSLITERATE Function	TRANSLITERATE Function	-
XRANGE Function	XRANGE Function	GaussDB：参数start_byte和end_byte无默认值。 Oracle：参数start_byte和end_byte默认为NULL。

表 3-114 DBMS_SCHEDULER 兼容性说明

Oracle数据库	GaussDB数据库
ADD_EVENT_QUEUE_SUBSCRIBER Procedure	不支持
ADD_GROUP_MEMBER Procedure	不支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
ADD_JOB_EMAIL_NOTIFICATION Procedure	不支持
ADD_TO_INCOMPATIBILITY Procedure	不支持
ALTER_CHAIN Procedure	不支持
ALTER_RUNNING_CHAIN Procedure	不支持
CLOSE_WINDOW Procedure	不支持
COPY_JOB Procedure	不支持
CREATE_CHAIN Procedure	不支持
CREATE_CREDENTIAL Procedure	CREATE_CREDENTIAL Procedure
CREATE_DATABASE_DESTINATION Procedure	不支持
CREATE_EVENT_SCHEDULE Procedure	不支持
CREATE_FILE_WATCHER Procedure	不支持
CREATE_GROUP Procedure	不支持
CREATE_INCOMPATIBILITY Procedure	不支持
CREATE_JOB Procedure	CREATE_JOB Procedure
CREATE_JOB_CLASS Procedure	CREATE_JOB_CLASS Procedure
CREATE_JOBS Procedure	不支持
CREATE_PROGRAM Procedure	CREATE_PROGRAM Procedure
CREATE_RESOURCE Procedure	不支持
CREATE_SCHEDULE Procedure	CREATE_SCHEDULE Procedure
CREATE_WINDOW Procedure	不支持
DEFINE_ANYDATA_ARGUMENT Procedure	不支持
DEFINE_CHAIN_EVENT_STEP Procedure	不支持
DEFINE_CHAIN_RULE Procedure	不支持
DEFINE_CHAIN_STEP Procedure	不支持
DEFINE_METADATA_ARGUMENT Procedure	不支持
DEFINE_PROGRAM_ARGUMENT Procedure	DEFINE_PROGRAM_ARGUMENT Procedure

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
DISABLE Procedure	DISABLE Procedure
DROP_AGENT_DESTINATION Procedure	不支持
DROP_CHAIN Procedure	不支持
DROP_CHAIN_RULE Procedure	不支持
DROP_CHAIN_STEP Procedure	不支持
DROP_CREDENTIAL Procedure	DROP_CREDENTIAL Procedure
DROP_DATABASE_DESTINATION Procedure	不支持
DROP_FILE_WATCHER Procedure	不支持
DROP_GROUP Procedure	不支持
DROP_INCOMPATIBILITY Procedure	不支持
DROP_JOB Procedure	DROP_JOB Procedure
DROP_JOB_CLASS Procedure	DROP_JOB_CLASS Procedure
DROP_PROGRAM Procedure	DROP_PROGRAM Procedure
DROP_PROGRAM_ARGUMENT Procedure	不支持
DROP_SCHEDULE Procedure	DROP_SCHEDULE Procedure
DROP_WINDOW Procedure	不支持
ENABLE Procedure	ENABLE Procedure
END_DETACHED_JOB_RUN Procedure	不支持
EVALUATE_CALENDAR_STRING Procedure	EVALUATE_CALENDAR_STRING Procedure
EVALUATE_RUNNING_CHAIN Procedure	不支持
GENERATE_JOB_NAME Function	GENERATE_JOB_NAME Function
GET_AGENT_INFO Function	不支持
GET_AGENT_VERSION Function	不支持
GET_ATTRIBUTE Procedure	不支持
GET_FILE Procedure	不支持
GET_SCHEDULER_ATTRIBUTE Procedure	不支持

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库
OPEN_WINDOW Procedure	不支持
PURGE_LOG Procedure	不支持
PUT_FILE Procedure	不支持
REMOVE_EVENT_QUEUE_SUBSCRIBER Procedure	不支持
REMOVE_FROM_INCOMPATIBILITY Procedure	不支持
REMOVE_GROUP_MEMBER Procedure	不支持
REMOVE_JOB_EMAIL_NOTIFICATION Procedure	不支持
RESET_JOB_ARGUMENT_VALUE Procedure	不支持
RUN_CHAIN Procedure	不支持
RUN_JOB Procedure	RUN_JOB Procedure
SET_AGENT_REGISTRATION_PASS Procedure	不支持
SET_ATTRIBUTE Procedure	SET_ATTRIBUTE Procedure
SET_ATTRIBUTE_NULL Procedure	不支持
SET_JOB_ANYDATA_VALUE Procedure	不支持
SET_JOB_ARGUMENT_VALUE Procedure	SET_JOB_ARGUMENT_VALUE Procedure
SET_JOB_ATTRIBUTES Procedure	不支持
SET_RESOURCE_CONSTRAINT Procedure	不支持
SET_SCHEDULER_ATTRIBUTE Procedure	不支持
STOP_JOB Procedure	STOP_JOB Procedure

表 3-115 DBMS_UTILITY 兼容性说明

Oracle数据库	GaussDB数据库	差异
ACTIVE_INSTANCES Procedure	不支持	-
ANALYZE_DATABASE Procedure	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
ANALYZE_PART_OBJECT Procedure	不支持	-
ANALYZE_SCHEMA Procedure	不支持	-
CANONICALIZE Procedure	CANONICALIZE Procedure	GaussDB：参数canon_len默认为1024字节。 Oracle：参数canon_len无默认值。
COMMA_TO_TABLE Procedures	COMMA_TO_TABLE Procedures	GaussDB：参数tab为VARCHAR2数组。 Oracle：该过程存在2个重载。参数tab可以为两种类型之一：一种为uncl_array，另一种为lname_array。
COMPILE_SCHEMA Procedure	不支持	-
CREATE_ALTER_TYPE_ERROR_TABLE Procedure	不支持	-
CURRENT_INSTANCE Function	不支持	-
DATA_BLOCK_ADDRESS_BLOCK Function	不支持	-
DATA_BLOCK_ADDRESS_FILE Function	不支持	-
DB_VERSION Procedure	DB_VERSION Procedure	GaussDB：只有参数version，类型为VARCHAR2。Oracle：有参数version和compatibility，类型均为VARCHAR2。
EXEC_DDL_STATEMENT Procedure	EXEC_DDL_STATEMENT Function	GaussDB：参数parse_string为TEXT类型。 Oracle：参数parse_string为VARCHAR2类型。
EXPAND_SQL_TEXT Procedure	EXPAND_SQL_TEXT Function	GaussDB： 参数output_sql_text为CLOB。 Oracle：参数output_sql_text为NOCOPY CLOB，通过传引用方式传递OUT参数。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
FORMAT_CALL_STACK Function	FORMAT_CALL_STACK Function	GaussDB：函数返回类型为TEXT。 Oracle：函数返回类型为VARCHAR2。
FORMAT_ERROR_BACKTRACE Function	FORMAT_ERROR_BACKTRACE Function	GaussDB：函数返回类型为TEXT。 Oracle：函数返回类型为VARCHAR2。
FORMAT_ERROR_STACK Function	FORMAT_ERROR_STACK Function	GaussDB：函数返回类型为TEXT。 Oracle：函数返回类型为VARCHAR2。
GET_CPU_TIME Function	GET_CPU_TIME Function	GaussDB：函数返回类型为BIGINT。 Oracle：函数返回类型为NUMBER。
GET_DEPENDENCY Procedure	不支持	-
GET_ENDIANNESS Function	GET_ENDIANNESS Function	GaussDB：函数返回类型为INTEGER。 Oracle：函数返回类型为NUMBER。
GET_HASH_VALUE Function	GET_HASH_VALUE Function	GaussDB：参数base、hash_size和返回类型均为INTEGER。 Oracle：参数base、hash_size和返回类型均为NUMBER。
GET_PARAMETER_VALUE Function	不支持	-
GET_SQL_HASH Function	GET_SQL_HASH Function	GaussDB：参数last4bytes为BIGINT类型，代表MD5哈希值的最后四字节，以无符号整数形式展现，函数返回类型为BIGINT。 Oracle：对应参数pre10ihash为NUMBER类型，用于存储MD5计算得到的16字节中的4字节哈希值。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
GET_TIME Function	GET_TIME Function	GaussDB：函数返回类型为BIGINT。 Oracle：函数返回类型为NUMBER。
GET_TZ_TRANSITIONS Procedure	不支持	-
INVALIDATE Procedure	不支持	-
IS_BIT_SET Function	IS_BIT_SET Function	GaussDB：参数n和返回值类型为INTEGER。 Oracle：参数n和返回值类型为NUMBER。
IS_CLUSTER_DATABASE Function	IS_CLUSTER_DATABASE Function	-
MAKE_DATA_BLOCK_ADDRESS Function	不支持	-
NAME_RESOLVE Procedure	NAME_RESOLVE Procedure	GaussDB：参数context和part1_type为INTEGER，参数 object_number为OID；GaussDB不支持NUMBER到OID的隐式转换。 Oracle：参数context、part1_type和object_number均为NUMBER。
NAME_TOKENIZE Procedure	NAME_TOKENIZE Procedure	GaussDB：参数nextpos为INTEGER类型。 Oracle：参数nextpos为BINARY_INTEGER类型。
OLD_CURRENT_SCHEMA Function	OLD_CURRENT_SCHEMA Function	GaussDB：函数返回类型为VARCHAR。 Oracle：函数返回类型为VARCHAR2。
OLD_CURRENT_USER Function	OLD_CURRENT_USER Function	GaussDB：函数返回类型为TEXT。 Oracle：函数返回类型为VARCHAR2。
PORT_STRING Function	不支持	-
SQLID_TO_SQLHASH Function	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
TABLE_TO_COMMA Procedures	TABLE_TO_COMMA Procedures	GaussDB：参数tab为VARCHAR2数组。 Oracle：该存储过程存在2个重载。参数tab可以为两种类型之一：一种为uncl_array，另一种为lname_array。
VALIDATE Procedure	不支持	-
WAIT_ON_PENDING_DML Function	不支持	-

表 3-116 DBMS_SQL 兼容性说明

Oracle数据库	GaussDB数据库	差异
BIND_ARRAY Procedures	SQL_BIND_ARRAY Function	-
BIND_VARIABLE Procedures	SQL_BIND_VARIABLE Function	-
BIND_VARIABLE_PKG Procedure	不支持	-
CLOSE_CURSOR Procedure	SQL_UNREGISTER_CONTEXT Function	-
COLUMN_VALUE Procedure	GET_RESULT Procedure	-
COLUMN_VALUE_LONG Procedure	不支持	-
DEFINE_ARRAY Procedure	SET_RESULTS_TYPE Procedure	-
DEFINE_COLUMN Procedures	SET_RESULT_TYPE Procedure	-
DEFINE_COLUMN_CHAR Procedure	不支持	-
DEFINE_COLUMN_LONG Procedure	不支持	-
DEFINE_COLUMN_RAW Procedure	不支持	-
DEFINE_COLUMN_ROWID Procedure	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DESCRIBE_COLUMNS Procedure	DESCRIBE_COLUMNS Procedure	-
DESCRIBE_COLUMNS2 Procedure	不支持	-
DESCRIBE_COLUMNS3 Procedure	不支持	-
EXECUTE Function	SQL_RUN Function	• GaussDB：返回值为常量1。当前对于语句中unknown类型之间的比较，无法正确返回结果。 • Oracle：返回值对于INSERT、UPDATE、DELETE语句是影响的行数，对于其他语句则无意义。
EXECUTE_AND_FETCH Function	RUN_AND_NEXT Function	-
FETCH_ROWS Function	NEXT_ROW Function	-
GET_NEXT_RESULT Procedures	不支持	-
IS_OPEN Function	IS_ACTIVE Function	-
LAST_ERROR_POSITION Function	不支持	-
LAST_ROW_COUNT Function	LAST_ROW_COUNT Function	-
LAST_ROW_ID Function	不支持	-
LAST_SQL_FUNCTION_CODE Function	不支持	-
OPEN_CURSOR Functions	REGISTER_CONTEXT Function	-
PARSE Procedures	支持，有差异	GaussDB中为SQL_SET_SQL Function，不支持重载。
RETURN_RESULT Procedures	不支持	-
TO_CURSOR_NUMBER Function	不支持	-
TO_REFCURSOR Function	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
VARIABLE_VALUE Procedures	GET_VARIABLE_RESULT Procedures	-
VARIABLE_VALUE_PKG Procedure	不支持	-

表 3-117 DBMS_SQL 数据类型兼容性说明

Oracle数据库	GaussDB数据库
DBMS_SQL DESC_REC	DBE_SQL.DESC_REC
DBMS_SQL DATE_TABLE	DBE_SQL.DATE_TABLE
DBMS_SQL NUMBER_TABLE	DBE_SQL.NUMBER_TABLE
DBMS_SQL VARCHAR2_TABLE	DBE_SQL.VARCHAR2_TABLE
DBMS_SQL BLOB_TABLE	DBE_SQL.BLOB_TABLE

表 3-118 UTL_FILE 兼容性说明

Oracle数据库	GaussDB数据库	差异
FCLOSE Procedure	CLOSE Procedure	-
FCLOSE_ALL Procedure	CLOSE_ALL Procedure	-
FCOPY Procedure	COPY Procedure	-
FFLUSH Procedure	FLUSH Procedure	-
FGETATTR Procedure	GET_ATTR Procedure	-
FGETPOS Function	GET_POS Function	-
FOPEN Function	FOPEN Function	-
FOPEN_NCHAR Function	FOPEN_NCHAR Function	-
FREMOVE Procedure	REMOVE Procedure	-
FRENAME Procedure	RENAME Procedure	-
FSEEK Procedure	SEEK Procedure	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
GET_LINE Procedure	READ_LINE Procedure	-
GET_LINE_NCHAR Procedure	READ_LINE_NCHAR Procedure	-
GET_RAW Procedure	GET_RAW Procedure	-
IS_OPEN Function	IS_OPEN Function	-
NEW_LINE Procedure	支持，有差异，NEW_LINE Function	GaussDB将接口定义为Function。
PUT Procedure	支持，有差异，WRITE Function	GaussDB将接口定义为Function。
PUT_LINE Procedure	支持，有差异，WRITE_LINE Function	GaussDB将接口定义为Function。
PUT_LINE_NCHAR Procedure	支持，有差异，WRITE_LINE_NCHAR Function	GaussDB将接口定义为Function。
PUT_NCHAR Procedure	支持，有差异，WRITE_NCHAR Function	GaussDB将接口定义为Function。
PUTF Procedure	支持，有差异，FORMAT_WRITE Function	GaussDB将接口定义为Function。
PUTF_NCHAR Procedure	支持，有差异，FORMAT_WRITE_NCHAR Function	GaussDB将接口定义为Function。
PUT_RAW Procedure	支持，有差异，PUT_RAW Function	GaussDB将接口定义为Function。

表 3-119 DBMS_SESSION 兼容性说明

Oracle数据库	GaussDB数据库	差异
CLEAR_ALL_CONTEXT Procedure	不支持	-
CLEAR_CONTEXT Procedure	CLEAR_CONTEXT Function	-
CLEAR_IDENTIFIER Procedure	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
CLOSE_DATABASE_LINK Procedure	不支持	-
CURRENT_IS_ROLE_ENABLED Function	不支持	-
FREE_UNUSED_USER_MEMORY Procedure	不支持	-
GET_PACKAGE_MEMORY_UTILIZATION Procedure	不支持	-
IS_ROLE_ENABLED Function	不支持	-
IS_SESSION_ALIVE Function	不支持	-
LIST_CONTEXT Procedures	不支持	-
MODIFY_PACKAGE_STATE Procedure	MODIFY_PACKAGE_STATE Procedure	GaussDB：仅支持入参flags = 1的场景使用。 Oracle：支持flags=1或flags = 2的场景使用。
RESET_PACKAGE Procedure	不支持	-
SESSION_IS_ROLE_ENABLED Function	不支持	-
SESSION_TRACE_DISABLE Procedure	不支持	-
SESSION_TRACE_ENABLE Procedure	不支持	-
SET_CONTEXT Procedure	SET_CONTEXT Function	GaussDB：仅包括参数namespace，attribute和value，类型均为text。 Oracle：包括参数namespace，attribute，value，username和client_id，类型均为VARCHAR2。
SET_EDITION_DEFERRED Procedure	不支持	-
SET_IDENTIFIER Procedure	不支持	-
SET_NLS Procedure	不支持	-
SET_ROLE Procedure	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
SET_SQL_TRACE Procedure	不支持	-
SLEEP Procedure	不支持	-
SWITCH_CURRENT_CONSUMER_GROUP Procedure	不支持	-
UNIQUE_SESSION_ID Function	不支持	-

表 3-120 UTL_MATCH 兼容性说明

Oracle数据库	GaussDB数据库	差异
EDIT_DISTANCE Function	不支持	-
EDIT_DISTANCE_SIMILARITY Function	EDIT_DISTANCE_SIMILARITY Function	GaussDB：参数str1和str2均为TEXT类型，函数返回类型为INTEGER。 Oracle：参数s1和s2为VARCHAR2类型，函数返回类型为PLS_INTEGER。
JARO_WINKLER Function	不支持	-
JARO_WINKLER_SIMILARITY Function	不支持	-

表 3-121 DBMS_APPLICATION_INFO 兼容性说明

Oracle数据库	GaussDB数据库	差异
READ_CLIENT_INFO Function	READ_CLIENT_INFO Procedure	GaussDB：参数client_info为TEXT类型。 Oracle：参数client_info为VARCHAR2类型。
READ_MODULE Procedure	READ_MODULE Procedure	GaussDB：参数module_name、action_name为TEXT类型。 Oracle：参数module_name、action_name为VARCHAR2类型。
SET_ACTION Procedure	SET_ACTION Procedure	GaussDB：参数action_name为TEXT类型。 Oracle：参数action_name为VARCHAR2类型。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
SET_CLIENT_INFO Function	SET_CLIENT_INFO Procedure	GaussDB：参数str为TEXT类型，且返回类型为void。 Oracle：参数client_info为VARCHAR2类型，无返回值。二者均为写入客户端信息，最大输入64字节，超过64字节将被截断。
SET_MODULE Procedure	SET_MODULE Procedure	GaussDB：参数module_name、action_name为TEXT类型。 Oracle：参数module_name、action_name为VARCHAR2类型。
SET_SESSION_LONGOPS Procedure	不支持	-

表 3-122 DBMS_XMLDOM 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.APPENDCHILD	DBE_XMLDOM.APPENDCHILD	● GaussDB：DOCUMENT类型节点下APPEND ATTR类型节点会报“operation not support”错误。Oracle：在此场景下不报错，但实际并没有挂载成功。 ● GaussDB：ATTR类型节点下APPEND ATTR类型节点会报“operation not support”错误。Oracle：在此场景下不报错，但实际并没有挂载成功。 ● GaussDB：父节点在添加多个ATTR类型子节点时，不允许KEY值相同的子节点同时存在于同一个父节点下。Oracle：允许KEY值相同的子节点同时存在于同一个父节点下。
DBMS_XMLDOM.CREATEELEMENT	DBE_XMLDOM.CREATEELEMENT	-
DBMS_XMLDOM.CREATETEXTNODE	DBE_XMLDOM.CREATETEXTNODE	-
DBMS_XMLDOM.FREEDOCUMENT	支持，有差异，DBE_XMLDOM.FREEDOCUMENT	GaussDB：释放时不会立刻释放对象，累积一定数量后释放。document下全部节点失效。 Oracle：立即释放对象。
DBMS_XMLDOM.FREEELEMENT	DBE_XMLDOM.FREEELEMENT	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.FREENODE	DBE_XMLDOM.FREENODE	-
DBMS_XMLDOM.FREENODELIST	支持，有差异，DBE_XMLDOM.FREENODELIST	GaussDB：nodelist会被释放。 Oracle：释放nodelist后，在原始的doc中还能被查询到。
DBMS_XMLDOM.GETATTRIBUTE	DBE_XMLDOM.GETATTRIBUTE	-
DBMS_XMLDOM.GETATTRIBUTES	DBE_XMLDOM.GETATTRIBUTES	-
DBMS_XMLDOM.GETCHILDNODES	DBE_XMLDOM.GETCHILDNODES	GaussDB：对document的node使用时会包含dtd。 Oracle：不包含dtd。
DBMS_XMLDOM.GETCHILDRENBYTAGNAME	DBE_XMLDOM.GETCHILDRENBYTAGNAME	GaussDB：DBE_XMLDOM.GETCHILDRENBYTAGNAME接口的参数ns不支持传入参数" * "，如需获取节点下全部属性，可使用DBE_XMLDOM.GETCHILDNODES接口。 Oracle：支持传入参数" * "。
DBMS_XMLDOM.GETDOCUMENTELEMENT	DBE.XMLDOM.GETDOCUMENTELEMENT	-
DBMS_XMLDOM.GETFIRSTCHILD	DBE_XMLDOM.GETFIRSTCHILD	-
DBMS_XMLDOM.GETLASTCHILD	DBE_XMLDOM.GETLASTCHILD	-
DBMS_XMLDOM.GETLENGTH	DBE_XMLDOM.GETLENGTH	-
DBMS_XMLDOM.GETLOCALNAME	DBE_XMLDOM.GETLOCALNAME	-
DBMS_XMLDOM.GETNAMEDITEM	DBE_XMLDOM.GETNAMEDITEM	-
DBMS_XMLDOM.GETNEXTSIBLING	DBE_XMLDOM.GETNEXTSIBLING	-
DBMS_XMLDOM.GETNODENAME	DBE_XMLDOM.GETNODENAME	-
DBMS_XMLDOM.GETNODETYPE	DBE_XMLDOM.GETNODETYPE	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.GETTAGNAME	DBE_XMLDOM.GETTAGNAME	-
DBMS_XMLDOM.IMPORTNODE	DBE_XMLDOM.IMPORTNODE	-
DBMS_XMLDOM.ISNULL	DBE_XMLDOM.ISNULL	GaussDB：入参为DOMNODELIST类型时，若对象在哈希表中不存在会发生报错。 Oracle：不会报错。
DBMS_XMLDOM.ITEM	DBE_XMLDOM.ITEM	-
DBMS_XMLDOM.MAKENODE	DBE_XMLDOM.MAKENODE	GaussDB：该函数不支持直接作为函数返回值返回。 Oracle：支持直接作为函数返回值返回。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.NEWDOMDOCUMENT	DBE_XMLDOM.NEWDOMDOCUMENT	• GaussDB入参大小需限制1GB以内。 Oracle：与CLOB类型大小一致。 • GaussDB 目前暂不支持外部DTD解析。 Oracle：支持解析外部DTD。 • GaussDB newdomdocument创建的doc，默认UTF-8字符集。 Oracle：根据服务端字符集生成。 • GaussDB 从同一个xmltype实例中解析出的每一个doc都是独立的，对doc的修改也不会影响到xmltype。 Oracle：从同一个xmltype实例中解析出的每一个doc不独立，有关联关系。 • GaussDB version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。 Oracle：不报错。 • GaussDB 与Oracle数据库DTD校验差异：!ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而Oracle数据库不报错。 <!ENTITY baidu "www.baidu.com">.....&Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应。 Oracle：不报错。 • GaussDB 与Oracle数据库命名空间校验差异：解析未声明的命名空间标签正常执行。 Oracle：报错。
DBMS_XMLDOM.SETATTRIBUTE	DBE_XMLDOM.SETATTRIBUTE	GaussDB：属性key不支持为null或空字符串。 Oracle：属性key允许为null或空字符串。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.SETCHARSET	DBE_XMLDOM.SETCHARSET	GaussDB目前支持的字符集有：UTF-8、UCS-4、UCS-2、ISO-8859-1、ISO-8859-2、ISO-8859-3、ISO-8859-4、ISO-8859-5、ISO-8859-6、ISO-8859-7、ISO-8859-8、ISO-8859-9、ISO-2022-JP、Shift_JIS、EUC-JP、ASCII。输入其他字符集会报错或者可能导致输出乱码。
DBMS_XMLDOM.SETDOCTYPE	DBE_XMLDOM.SETDOCTYPE	GaussDB name、sysid、pubid的总长度限制在32500个字节以内。 Oracle：限制在32767字节内。
DBMS_XMLDOM.WRITETOBUFFER	支持，有差异，DBE_XMLDOM.WRITETOBUFFER	• GaussDB：writetobuffer输出buffer限制在1GB以内。 Oracle：限制在32767字节内。 • GaussDB：输出doc将包含XML声明version和encoding。 Oracle：用户不主动指定将不包含。 • GaussDB：入参为domnode类型时，如果节点是doc转换的，输出节点将包含XML声明version和encoding。 Oracle：用户不主动指定将不包含。 • GaussDB：默认以UTF-8字符集输出xml。 Oracle：根据数据库字符集生成。
DBMS_XMLDOM.WRITETOCLOB	DBE_XMLDOM.WRITETOCLOB	• GaussDB：writetoclob大小支持1GB以内。 Oracle：按CLOB大小支持。 • GaussDB：输出doc将包含XML声明version和encoding。 Oracle：用户不主动指定将不包含。 • GaussDB：入参为domnode类型时，如果节点是doc转换的，输出节点将包含XML声明version和encoding。 Oracle：用户不主动指定将不包含。 • GaussDB 默认以UTF-8字符集输出xml。 Oracle：根据数据库字符集生成。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.WRITETOFILE	DBE_XMLDOM.WRITETOFILE	• GaussDB：document入参，filename长度限制在255个字节以内，charset请参考dbe_xmldom.setcharset接口。 Oracle：filename长度限制受操作系统影响，大于255个字节。 • GaussDB：domnode入参，filename长度限制在255个字节以内，charset请参考dbe_xmldom.setcharset接口。 Oracle：filename长度限制受操作系统影响，大于255个字节。 • GaussDB：该函数会添加缩进等内容，将输出格式化。输出doc将包含XML声明version和encoding。入参为domnode类型时，如果节点是doc转换的，输出节点将包含XML声明version和encoding。 Oracle：用户不主动指定将不包含。 • GaussDB：传入newdomdocument()无参创建的doc，在不指定charset时不会报错，默认UTF-8字符集。 Oracle：会进行报错。 • GaussDB：filename需要在pg_directory中创建的路径下，filename中的\会被转换成/，只允许存在一个/。文件名格式应为pg_directory_name/file_name。 Oracle：按用户输入不进行转义。
DBMS_XMLDOM.GETNODEVALUE	DBE_XMLDOM.GETNODEVALUE	-
DBMS_XMLDOM.GETPARENTNODE	DBE_XMLDOM.GETPARENTNODE	-
DBMS_XMLDOM.HASCHILDNODES	DBE_XMLDOM.HASCHILDNODES	-
DBMS_XMLDOM.MAKEELEMENT	DBE_XMLDOM.MAKEELEMENT	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLDOM.SETNODEVALUE	DBE_XMLDOM.SETNODEVALUE	• GaussDB：nodeValue入参，可以输入空字符串和NULL值，但不会对节点值进行修改。 Oracle：空字符串和NULL会将节点值修改为空字符串。 • GaussDB：nodeValue入参，暂不支持转义字符'&'，如字符串中包含该转义字符，会清空节点值。 Oracle：支持转义字符。
DBMS_XMLDOM.GETELEMENTSBYTAGNAME	DBE_XMLDOM.GETELEMENTSBYTAGNAME	-

表 3-123 DBMS_XMLPARSER 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_XMLPARSER.FREEPARSER	DBE_XMLPARSER.FREEPARSER	-
DBMS_XMLPARSER.GETDOCUMENT	DBE_XMLPARSER.GETDOCUMENT	-
DBMS_XMLPARSER.GETVALIDATIONMODE	DBE_XMLPARSER.GETVALIDATIONMODE	-
DBMS_XMLPARSER.NEWPARSER	支持，有差异，DBE_XMLPARSER.NEWPARSER	GaussDB中parser对象的数量上限为16777215，Oracle中约为1亿。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLPARSER.PARSEBUFFER	支持，有差异，DBE_XMLPARSER.PARSEBUFFER	• 与Oracle数据库解析字段差异：字符串encoding只支持UTF-8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。 • 与Oracle数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而Oracle数据库会报错。 • 与Oracle数据库xml预定义实体解析差异：'"会被解析转义为字符’”，而Oracle中预定义实体统一都没有转义为字符。 • 与Oracle数据库DTD校验差异： – !ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而Oracle数据库不报错。 – <!ENTITY baidu "www.baidu.com">..... &Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而Oracle数据库不报错。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLPARSER.PARSECLOB	支持，有差异，DBE_XMLPARSER.PARSECLOB	• PARSECLOB不支持解析大于等于2GB的clob。 • 与Oracle数据库解析字段差异：字符串encoding只支持UTF-8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。 • 与Oracle数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而Oracle数据库会报错。 • 与Oracle数据库xml预定义实体解析差异：'"会被解析转义为字符’”，而Oracle中预定义实体统一都没有转义为字符。 • 与Oracle数据库DTD校验差异： – !ATTLIST to type (CHECK|check|Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而Oracle数据库不报错。 – <!ENTITY baidu "www.baidu.com">..... &Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而Oracle数据库不报错。
DBMS_XMLPARSER.SETVALIDATIONMODE	DBE_XMLPARSER.SETVALIDATIONMODE	-

表 3-124 DBMS_ILM 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_ILM.ADD_TO_ILM	不支持	-
DBMS_ILM.ARCHIVESTATENAME	不支持	-
DBMS_ILM.EXECUTE_ILM	DBE_ILM.EXECUTE_ILM	• GaussDB数据库的入参Schema在Oracle数据库中对应为owner。 • GaussDB数据库不支持指定ilm_scope（一次指定多个对象）的操作。

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_ILM.EXECUTE_ILM_TASK	不支持	-
DBMS_ILM.PREVIEW_ILM	不支持	-
DBMS_ILM.REMOVE_FROM_ILM	不支持	-
DBMS_ILM.STOP_ILM	DBE_ILM.STOP_ILM	-

表 3-125 DBMS_ILM_ADMIN 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_ALL	不支持	-
DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_TABLE	不支持	-
DBMS_ILM_ADMIN.CUSTOMIZE_ILM	DBE_ILM_ADMIN.CUSTOMIZE_ILM	入参parameter取值对应的特性参数存在差异。 ● GaussDB数据库param取值支持1、2、7、11、12、13、14和15。 ● GaussDB数据库param取值为14时，对应的特性参数为WIND_DURATION，用于控制自动调度中执行窗口的持续时长，而ORACLE数据库无对应的特性参数。
DBMS_ILM_ADMIN.DISABLE_ILM	DBE_ILM_ADMIN.DISABLE_ILM	-
DBMS_ILM_ADMIN.ENABLE_AUTO_OPTIMIZE	不支持	-
DBMS_ILM_ADMIN.ENABLE_ILM	DBE_ILM_ADMIN.ENABLE_ILM	-
DBMS_ILM_ADMIN. IGNORE_AUTO_OPTIMIZE_ CRITERIA	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_ILM_ADMIN.SET_HEAT_MAP_ALL	不支持	-
DBMS_ILM_ADMIN.SET_HEAT_MAP_START	不支持	-
DBMS_ILM_ADMIN.SET_HEAT_MAP_TABLE	不支持	-

表 3-126 DBMS_COMPRESSION 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_COMPRESSION.GET_COMPRESSION_RATIO	DBE_COMPRESSION.GET_COMPRESSION_RATIO	• GaussDB不支持LOBs的压缩率获取。 • 对于单个对象的压缩率获取： – GaussDB入参comptype取值仅支持1（未压缩）和2（高级压缩），Oracle还支持1024、2048等取值。 – GaussDB入参objtype取值仅支持1（表对象），而Oracle还支持2（索引对象）。 – Oracle数据库使用subset_numrows参数直接来决定采样的行数（即为参数的取值），而GaussDB则使用sample_ratio（采样率）来间接确定采样的行数。
DBMS_COMPRESSION.GET_COMPRESSION_TYPE	DBE_COMPRESSION.GET_COMPRESSION_TYPE	• Oracle使用rowid来指定待获取压缩类型的行，而GaussDB则是使用行的ctid来指定。 • 返回值为comptype，其取值差异同GET_COMPRESSION_RATIO。 • GaussDB该接口仅支持在DN调用，详见**DBE_COMPRESSION**。

版权所有 © 华为云计算技术有限公司

表 3-127 DBMS_HEAT_MAP 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_HEAT_MAP.BLOCK_HEAT_MAP	不支持	-
DBMS_HEAT_MAP.EXTENT_HEAT_MAP	不支持	-
DBMS_HEAT_MAP.OBJECT_HEAT_MAP	不支持	-
DBMS_HEAT_MAP.SEGMENT_HEAT_MAP	不支持	-
DBMS_HEAT_MAP.TABLESPACE_HEAT_MAP	不支持	-
不支持	DBE_HEAT_MAP.ROW_HEAT_MAP	GaussDB该接口仅支持在DN调用，请参见**DBE_HEAT_MAP**。

表 3-128 DBMS_DESCRIBE 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_DESCRIBE.DESCRIBE_PROCEDURE	DBE_DESCRIBE.DESCRIBE_PROCEDURE	• datatype参数与Oracle存在差异，GaussDB返回数据类型的oid，Oracle数据库返回Oracle数据库内部的数据类型的编号。 • datalength、dataprecision和scale因GaussDB创建存储过程或函数时无法保留类型的约束（如number(7,2)、varchar2(20)等），该三个参数置0处理；Oracle可使用%type方法获得带约束的数据类型。 • GaussDB的具体使用请参见**DBE_DESCRIBE**。

表 3-129 DBMS_STATS 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.ALTER_STATS_HISTORY_RETENTION	不支持	-
DBMS_STATS.CANCEL_ADVISOR_TASK	不支持	-
DBMS_STATS.CONFIGURE_ADVISOR_FILTER	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER	不支持	-
DBMS_STATS.CONFIGURE_ADVISOR_OPR_FILTER	不支持	-
DBMS_STATS.CONFIGURE_ADVISOR_RULE_FILTER	不支持	-
DBMS_STATS.CREATE_ADVISOR_TASK	不支持	-
DBMS_STATS.CONVERT_RAW_VALUE	不支持	-
DBMS_STATS.CONVERT_RAW_VALUE_NVARCHAR	不支持	-
DBMS_STATS.CONVERT_RAW_VALUE_ROWID	不支持	-
DBMS_STATS.COPY_TABLE_STATS	不支持	-
DBMS_STATS.CREATE_EXTENDED_STATS	不支持	-
DBMS_STATS.CREATE_STAT_TABLE	DBE_STATS.CREATE_STAT_TABLE	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。
DBMS_STATS.DELETE_COLUMN_STATS	不支持	-
DBMS_STATS.DELETE_DATABASE_PREFS	不支持	-
DBMS_STATS.DELETE_DATABASE_STATS	不支持	-
DBMS_STATS.DELETE_DICTIONARY_STATS	不支持	-
DBMS_STATS.DELETE_FIXED_OBJECTS_STATS	不支持	-
DBMS_STATS.DELETE_INDEX_STATS	不支持	-
DBMS_STATS.DELETE_PENDING_STATS	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.DELETE_PROCESSING_RATE	不支持	-
DBMS_STATS.DELETE_SCHEMA_PREFS	不支持	-
DBMS_STATS.DELETE_SCHEMA_STATS	不支持	-
DBMS_STATS.DELETE_SYSTEM_STATS	不支持	-
DBMS_STATS.DELETE_TABLE_PREFS	不支持	-
DBMS_STATS.DELETE_TABLE_STATS	不支持	-
DBMS_STATS.DIFF_TABLE_STATS_IN_HISTORY	不支持	-
DBMS_STATS.DIFF_TABLE_STATS_IN_PENDING	不支持	-
DBMS_STATS.DIFF_TABLE_STATS_IN_STATTAB	不支持	-
DBMS_STATS.DROP_ADVISOR_TASK	不支持	-
DBMS_STATS.DROP_EXTENDED_STATS	不支持	-
DBMS_STATS.DROP_STAT_TABLE	DBE_STATS.DROP_STAT_TABLE	-
DBMS_STATS.EXECUTE_ADVISOR_TASK	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.EXPORT_COLUMN_STATS	DBE_STATS.EXPORT_COLUMN_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见DBE_STATS。 • 导出的列级统计信息与pg_statistic表保持一致，多列和pg_statistic_ext表保持一致。 • 支持导出索引表达式统计信息。要求tabname传的是索引名称，colname传的是索引表达式名称。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.EXPORT_DATABASE_PREFS	不支持	-
DBMS_STATS.EXPORT_DATABASE_STATS	DBE_STATS.EXPORT_DATABASE_STATS	• GaussDB中statown应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见DBE_STATS。 • stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.EXPORT_DICTIONARY_STATS	不支持	-
DBMS_STATS.EXPORT_FIXED_OBJECTS_STATS	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.EXPORT_INDEX_STATS	DBE_STATS.EXPORT_INDEX_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.EXPORT_PENDING_STATS	不支持	-
DBMS_STATS.EXPORT_SCHEMA_PREFS	不支持	-
DBMS_STATS.EXPORT_SCHEMA_STATS	DBE_STATS.EXPORT_SCHEMA_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。导出表相关列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 • 权限：需要具有stattab表的siud权限。
DBMS_STATS.EXPORT_SYSTEM_STATS	不支持	-
DBMS_STATS.EXPORT_TABLE_PREFS	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.EXPORT_TABLE_STATS	DBE_STATS.EXPORT_TABLE_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见DBE_STATS。 • stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。级联导出的列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.FLUSH_DATABASE_MONITORING_INFO	不支持	-
DBMS_STATS.GATHER_DATABASE_STATS	不支持	-
DBMS_STATS.GATHER_DICTIONARY_STATS	不支持	-
DBMS_STATS.GATHER_FIXED_OBJECTS_STATS	不支持	-
DBMS_STATS.GATHER_INDEX_STATS	不支持	-
DBMS_STATS.GATHER_PROCESSING_RATE	不支持	-
DBMS_STATS.GATHER_SCHEMA_STATS	不支持	-
DBMS_STATS.GATHER_SYSTEM_STATS	不支持	-
DBMS_STATS.GATHER_TABLE_STATS	不支持	-
DBMS_STATS.GENERATE_STATS	不支持	-
DBMS_STATS.GET_ADVISOR_OPR_FILTER	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.GET_ADVISOR_RECS	不支持	-
DBMS_STATS.GET_COLUMN_STATS	不支持	-
DBMS_STATS.GET_INDEX_STATS	不支持	-
DBMS_STATS.GET_PARAM	不支持	-
DBMS_STATS.GET_PREFS	不支持	-
DBMS_STATS.GET_STATS_HISTORY_AVAILABILITY	DBE_STATS.GET_STATS_HISTORY_AVAILABILITY	GaussDB查询到的是全库存在的最早历史统计信息的收集时间。
DBMS_STATS.GET_STATS_HISTORY_RETENTION	DBE_STATS.GET_STATS_HISTORY_RETENTION	-
DBMS_STATS.GET_SYSTEM_STATS	不支持	-
DBMS_STATS.GET_TABLE_STATS	不支持	-
DBMS_STATS.IMPLEMENT_ADVISOR_TASK	不支持	-
DBMS_STATS.IMPORT_COLUMN_STATS	DBE_STATS.IMPORT_COLUMN_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • 导出单列col导出的统计信息与pg_statistic表保持一致。多列ext-col导出的统计信息与pg_statistic_ext表保持一致。 • 支持导入索引表达式统计信息。要求tabname传的是索引名称，colname传的是索引表达式名称。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.IMPORT_DATABASE_PREFS	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.IMPORT_DATABASE_STATS	DBE_STATS.IMPORT_DATABASE_STATS	• GaussDB中statown应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。
DBMS_STATS.IMPORT_DICTIONARY_STATS	不支持	-
DBMS_STATS.IMPORT_FIXED_OBJECTS_STATS	不支持	-
DBMS_STATS.IMPORT_INDEX_STATS	DBE_STATS.IMPORT_INDEX_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.IMPORT_SCHEMA_PREFS	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.IMPORT_SCHEMA_STATS	DBE_STATS.IMPORT_SCHEMA_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。导入表相关列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 • 权限：需要具有stattab表的siud权限。
DBMS_STATS.IMPORT_SYSTEM_STATS	不支持	-
DBMS_STATS.IMPORT_TABLE_PREFS	不支持	-
DBMS_STATS.IMPORT_TABLE_STATS	DBE_STATS.IMPORT_TABLE_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。 • stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。级联导入的列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
DBMS_STATS.INTERRUPT_ADVISOR_TASK	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.LOCK_PARTITION_STATS	DBE_STATS.LOCK_PARTITION_STATS	• GaussDB中ownname应传入Schema名称。
DBMS_STATS.LOCK_SCHEMA_STATS	DBE_STATS.LOCK_SCHEMA_STATS	• GaussDB中ownname应传入Schema名称。
DBMS_STATS.LOCK_TABLE_STATS	DBE_STATS.LOCK_TABLE_STATS	• GaussDB中ownname应传入Schema名称。
DBMS_STATS.MERGE_COL_USAGE	不支持	-
DBMS_STATS.PREPARE_COLUMN_VALUES	不支持	-
DBMS_STATS.PREPARE_COLUMN_VALUES_ROWID	不支持	-
DBMS_STATS.PUBLISH_PENDING_STATS	不支持	-
DBMS_STATS.PURGE_STATS	DBE_STATS.PURGE_STATS	-
DBMS_STATS.REMAP_STAT_TABLE	不支持	-
DBMS_STATS.REPORT_ADVISOR_TASK	不支持	-
DBMS_STATS.REPORT_COL_USAGE	不支持	-
DBMS_STATS.REPORT_GATHER_AUTO_STATS	不支持	-
DBMS_STATS.REPORT_GATHER_DATABASE_STATS	不支持	-
DBMS_STATS.REPORT_GATHER_DICTIONARY_STATS	不支持	-
DBMS_STATS.REPORT_GATHER_FIXED_OBJ_STATS	不支持	-
DBMS_STATS.REPORT_GATHER_SCHEMA_STATS	不支持	-
DBMS_STATS.REPORT_STATS_OPERATIONS	不支持	-
DBMS_STATS.RESET_ADVISOR_TASK	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.RESET_COL_USAGE	不支持	-
DBMS_STATS.RESET_GLOBAL_PREF_DEFAULTS	不支持	-
DBMS_STATS.RESET_PARAM_DEFAULTS	不支持	-
DBMS_STATS.RESTORE_DICTIONARY_STATS	不支持	-
DBMS_STATS.RESTORE_FIXED_OBJECTS_STATS	不支持	-
DBMS_STATS.RESTORE_SCHEMA_STATS	DBE_STATS.RESTORE_SCHEMA_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。
DBMS_STATS.RESTORE_SYSTEM_STATS	不支持	-
DBMS_STATS.RESTORE_TABLE_STATS	DBE_STATS.RESTORE_TABLE_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。
DBMS_STATS.RESUME_ADVISOR_TASK	不支持	-
DBMS_STATS.SCRIPT_ADVISOR_TASK	不支持	-
DBMS_STATS.SEED_COL_USAGE	不支持	-
DBMS_STATS.SET_ADVISOR_TASK_PARAMETER	不支持	-
DBMS_STATS.SET_COLUMN_STATS	DBE_STATS.SET_COLUMN_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见**DBE_STATS**。
DBMS_STATS.SET_DATABASE_PREFS	不支持	-
DBMS_STATS.SET_GLOBAL_PREFS	不支持	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_STATS.SET_INDEX_STATS	DBE_STATS.SET_INDEX_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见DBE_STATS。 • GaussDB中新增了relallvisible入参。
DBMS_STATS.SET_PARAM	不支持	-
DBMS_STATS.SET_PROCESSING_RATE	不支持	-
DBMS_STATS.SET_SCHEMA_PREFS	不支持	-
DBMS_STATS.SET_SYSTEM_STATS	不支持	-
DBMS_STATS.SET_TABLE_PREFS	不支持	-
DBMS_STATS.SET_TABLE_STATS	DBE_STATS.SET_TABLE_STATS	• GaussDB中ownname应传入Schema名称。 • GaussDB仅支持部分入参功能，详情请参见DBE_STATS。 • GaussDB中新增了relallvisible入参。
DBMS_STATS.SHOW_EXTENDED_STATS_NAME	不支持	-
DBMS_STATS.TRANSFER_STATS	不支持	-
DBMS_STATS.UNLOCK_PARTITION_STATS	DBE_STATS.UNLOCK_PARTITION_STATS	GaussDB中ownname应传入Schema名称。
DBMS_STATS.UNLOCK_SCHEMA_STATS	DBE_STATS.UNLOCK_SCHEMA_STATS	GaussDB中ownname应传入Schema名称。
DBMS_STATS.UNLOCK_TABLE_STATS	DBE_STATS.UNLOCK_TABLE_STATS	GaussDB中ownname应传入Schema名称。
DBMS_STATS.UPGRADE_STAT_TABLE	不支持	-

版权所有 © 华为云计算技术有限公司

表 3-130 DBMS_XMLGEN 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_XMLGEN.CONVERT	DBE_XMLGEN.CONVERT	-
DBMS_XMLGEN.NEWCONTEXT	DBE_XMLGEN.NEWCONTEXT	-
DBMS_XMLGEN.NEWCONTEXTFROMHIERARCHY	DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY	• GaussDB生成的递归XML最大深度不能超过5000万层。 • Oracle的newcontextfromhierarchy方法对于connect by语句生成的xml是带xml头的，但是对于直接构造的数据不带xml头，GaussDB均带xml头。
DBMS_XMLGEN.SETCONVERTSPECIALCHARS	DBE_XMLGEN.SETCONVERTSPECIALCHARS	-
DBMS_XMLGEN.SETNULLHANDLING	DBE_XMLGEN.SETNULLHANDLING	-
DBMS_XMLGEN.SETROWSETTAG	DBE_XMLGEN.SETROWSETTAG	-
DBMS_XMLGEN.SETROWTAG	DBE_XMLGEN.SETROWTAG	-
DBMS_XMLGEN.USENULLATTRIBUTEINDICATOR	DBE_XMLGEN.USENULLATTRIBUTEINDICATOR	-
DBMS_XMLGEN.USEITEMTAGSFORCOLL	DBE_XMLGEN.USEITEMTAGSFORCOLL	-
DBMS_XMLGEN.GETNUMROWSPROCESSED	DBE_XMLGEN.GETNUMROWSPROCESSED	-
DBMS_XMLGEN.SETMAXROWS	DBE_XMLGEN.SETMAXROWS	-
DBMS_XMLGEN.SETSKIPROWS	DBE_XMLGEN.SETSKIPROWS	-
DBMS_XMLGEN.RESTARTQUERY	DBE_XMLGEN.RESTARTQUERY	GaussDB分布式中不允许cursor反向移动，因此restartquery函数不可使用。
DBMS_XMLGEN.GETXMLTYPE	DBE_XMLGEN.GETXMLTYPE	-

版权所有 © 华为云计算技术有限公司

Oracle数据库	GaussDB数据库	差异
DBMS_XMLGEN.GETXML	DBE_XMLGEN.GETXML	-
DBMS_XMLGEN.CLOSECONTEXT	DBE_XMLGEN.CLOSECONTEXT	-

表 3-131 DBMS_OBFUSCATION_TOOLKIT 兼容性说明

Oracle数据库	GaussDB数据库	差异
DBMS_OBFUSCATION_TOOLKIT.DESGETKEY	DBE_OBFUSCATION_TOOLKIT.DESGETKEY	-
DBMS_OBFUSCATION_TOOLKIT.DESENCRYPT	DBE_OBFUSCATION_TOOLKIT.DESENCRYPT	-
DBMS_OBFUSCATION_TOOLKIT.DESDECRYPT	DBE_OBFUSCATION_TOOLKIT.DESDECRYPT	-
DBMS_OBFUSCATION_TOOLKIT.DES3GETKEY	DBE_OBFUSCATION_TOOLKIT.DES3GETKEY	-
DBMS_OBFUSCATION_TOOLKIT.DES3ENCRYPT	DBE_OBFUSCATION_TOOLKIT.DES3ENCRYPT	-
DBMS_OBFUSCATION_TOOLKIT.DES3DECRYPT	DBE_OBFUSCATION_TOOLKIT.DES3DECRYPT	-
DBMS_OBFUSCATION_TOOLKIT.MD5	DBE_OBFUSCATION_TOOLKIT.MD5	-

3.3 MySQL 兼容性说明

3.3.1 MySQL 数据库兼容性概述

MySQL 兼容性 M-Compatibility 模式概述

MySQL兼容性M-Compatibility模式主要介绍GaussDB数据库的MySQL兼容性M-Compatibility模式（即sql_compatibility='M'）与MySQL 5.7数据库的兼容性对比信息。仅介绍505.2版本后新增的兼容性特性，特性的相关规格和约束建议在《M-Compatibility开发指南》中查看。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

GaussDB的执行计划和优化、EXPLAIN显示结果与MySQL不同。

由于GaussDB数据库与MySQL数据库在底层框架实现上存在差异，GaussDB数据库与MySQL数据库仍存在部分差异。

版权所有 © 华为云计算技术有限公司

> **说明**
>
> 由于GaussDB的底层架构与MySQL存在差异，对于information_schema和m_schema下与MySQL名称相同的Schema（具体请参见《M-Compatibility开发指南》中“Schema”章节），其查询性能可能存在差异。例如count函数无法做执行上的优化，表现为SELECT *和SELECT count(*)语句耗时近似。

MySQL 兼容性 MYSQL 模式概述

MySQL兼容性MYSQL模式主要介绍GaussDB数据库的MySQL兼容性MYSQL模式（即sql_compatibility = 'MYSQL'、且设置参数b_format_version='5.7'、b_format_dev_version='s1'时）与MySQL 5.7数据库的兼容性对比信息。仅介绍503.0.0版本后新增的兼容性特性。

> **说明**
>
> - MySQL兼容性MYSQL模式在语法、数据类型、元数据、协议等功能上对MySQL数据库的兼容度较高。MySQL兼容性MYSQL模式由于架构限制无法很好的与MySQL兼容，后续不再演进。
> - 仅在**MySQL兼容性MYSQL模式**中体现的特性行为与MySQL进行了兼容性适配，未体现的特性行为仍然与GaussDB保持一致。
> - 当前形态下MYSQL模式（即sql_compatibility='MYSQL'）与集中式形态下B模式（即sql_compatibility='B'）实现逻辑相近。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

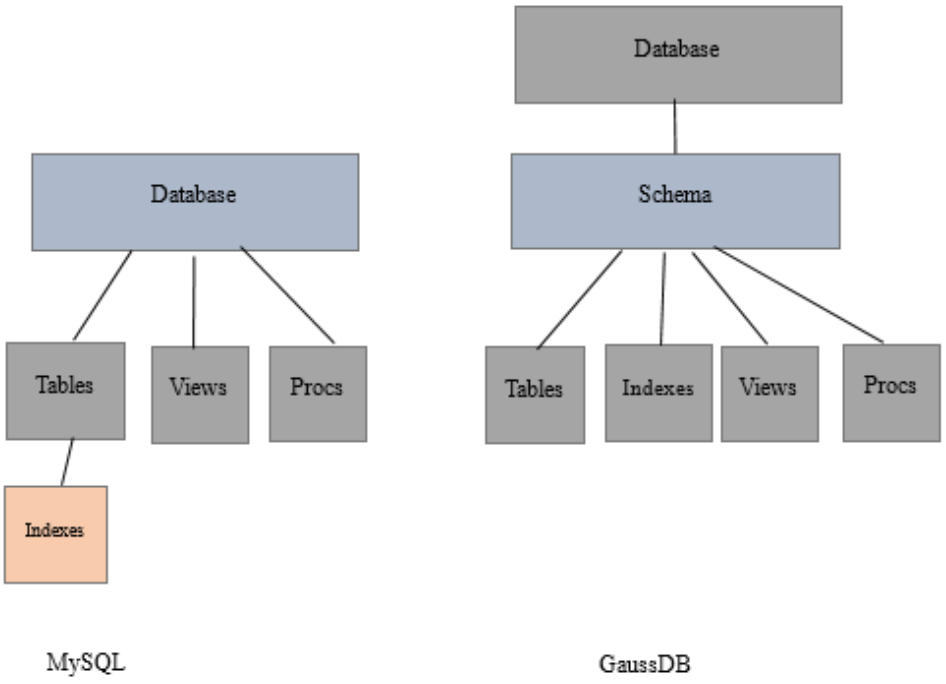
由于GaussDB数据库与MySQL数据库在底层框架实现上存在差异，GaussDB数据库与MySQL数据库仍存在部分差异。

Database 和 Schema 设计

MySQL的数据对象包括DATABASE、TABLE、INDEX、VIEW、TRIGGER、PROC等，MySQL的对象层次跟GaussDB的对应关系是从上至下且一对多包含关系。如**图3-1**所示：

版权所有 © 华为云计算技术有限公司

图 3-1 MySQL 和 GaussDB 中 Database 和 Schema 之间的差异



- 在MySQL中Database和Schema是同义词；而在GaussDB中，一个Database下可以有多个Schema。在该特性中，每个MySQL中的Database都被映射到GaussDB的一个Schema。
- 在MySQL中，INDEX从属于一个TABLE，但在GaussDB中，INDEX从属于一个Schema。这个差异导致INDEX名在GaussDB中要求在Schema内唯一，但在MySQL中仅要在在一个表内唯一。这个差异将作为当前约束予以保留。

3.3.2 MySQL 兼容性 M-Compatibility 模式

3.3.2.1 数据类型

3.3.2.1.1 数值数据类型

除特别说明外，MySQL兼容性M-Compatibility模式中的数据类型精度、标度、位数大小等默认不支持用浮点型数值定义，建议使用合法的整型数值定义。

表 3-132 整数类型

数据类型	与MySQL的差异
BOOL	输出格式：GaussDB中SELECT TRUE/FALSE输出结果为t/f，MySQL为1/0。 MySQL：BOOL/BOOLEAN类型实际映射为TINYINT类型。
BOOLEAN	

版权所有 © 华为云计算技术有限公司

数据类型	与MySQL的差异
TINYINT[(M)] [UNSIGNED] [ZEROFILL] SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	输入格式： ● MySQL： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下MySQL会发生错误解析，抛出WARNING但插入表成功，例如将“1.2.3.4.5”插入表后值为12。 ● GaussDB： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下，会将第二个小数点后的字符当作非法字符全部截断，抛出WARNING但插入表成功，例如将“1.2.3.4.5”插入表后值为1，“1.6.3.4.5”插入表后值为2。
MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]	MySQL存储MEDIUMINT数据需要3字节。 ● 带符号的范围是-8,388,608 ~ +8,388,607。 ● 无符号的范围是0 ~ +16,777,215。 GaussDB映射为INT类型，存储需要4字节，通过边界值判断限制取值范围。 ● 带符号的范围是-8,388,608 ~ +8,388,607。 ● 无符号的范围是0 ~ +16,777,215。 其他差异请参见表格下方说明中的内容。
INT[(M)] [UNSIGNED] [ZEROFILL] INTEGER[(M)] [UNSIGNED] [ZEROFILL] BIGINT[(M)] [UNSIGNED] [ZEROFILL]	输入格式： ● MySQL： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下MySQL会发生错误解析，抛出WARNING但插入表成功，例如将“1.2.3.4.5”插入表后值为12。 ● GaussDB： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下，会将第二个小数点后的字符当作非法字符全部截断，抛出WARNING但插入表成功，例如将“1.2.3.4.5”插入表后值为1，“1.6.3.4.5”插入表后值为2。

表 3-133 任意精度类型

数据类型	与MySQL的差异
DECIMAL[(M[,D])] [ZEROFILL] NUMERIC[(M[,D])] [ZEROFILL] DEC[(M[,D])] [ZEROFILL]	-

版权所有 © 华为云计算技术有限公司

数据类型	与MySQL的差异
FIXED[(M[,D])] [ZEROFILL]	

表 3-134 浮点类型

数据类型	与MySQL的差异
FLOAT[(M,D)] [ZEROFILL]	FLOAT数据类型不支持KEY键值分区策略分区表。
FLOAT(p) [ZEROFILL]	FLOAT数据类型不支持KEY键值分区策略分区表。
DOUBLE[(M,D)] [ZEROFILL]	DOUBLE数据类型不支持KEY键值分区策略分区表。
DOUBLE PRECISION[(M,D)] [ZEROFILL]	DOUBLE PRECISION数据类型不支持KEY键值分区策略分区表。
REAL[(M,D)] [ZEROFILL]	REAL数据类型不支持KEY值分区策略分区表。

3.3.2.1.2 日期与时间数据类型

表 3-135 日期与时间数据类型

数据类型	与MySQL的差异
DATE	GaussDB支持date数据类型，与MySQL相比规格上存在如下差异： 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。
DATETIME[(fsp)]	GaussDB支持datetime数据类型，与MySQL相比规格上存在如下差异： 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。

版权所有 © 华为云计算技术有限公司

数据类型	与MySQL的差异
TIMESTAMP[(fsp)]	GaussDB支持timestamp数据类型，与MySQL相比规格上存在如下差异： ● 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 ● MySQL支持设置explicit_defaults_for_timestamp，当explicit_defaults_for_timestamp设置为off时，对TIMESTAMP类型的字段的默认值、插入NULL等处理为非标准行为。MySQL 5.7下，explicit_defaults_for_timestamp默认值为off；MySQL 8.0下，explicit_defaults_for_timestamp默认值为on。GaussDB不支持设置explicit_defaults_for_timestamp，行为与MySQL设置explicit_defaults_for_timestamp为on时的行为一致。explicit_defaults_for_timestamp相关说明参见表格下方说明中的内容。
TIME[(fsp)]	GaussDB支持time数据类型，与MySQL相比规格上存在如下差异： ● 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 ● 当时间类型的时、分、秒、纳秒为0时，GaussDB和MySQL可能存在符号位不同的情况。
YEAR[(4)]	-

版权所有 © 华为云计算技术有限公司

说明

- GaussDB不支持ODBC语法的字面量：

  { d 'str' }

  { t 'str' }

  { ts 'str' }

- 当给DATETIME、TIME、TIMESTAMP数据类型指定的精度超过其支持的最大精度时，GaussDB会将精度截断成支持的最大精度，MySQL则会报错。
- MySQL中当explicit_defaults_for_timestamp为off时，对TIMESTAMP类型的字段的处理逻辑如下：
  - 未显式指定NULL/NOT NULL属性的字段，将自动添加NOT NULL属性。向此类字段插入NULL值时，会将NULL值替换为当前时间戳。
  - 如果未给表中的第一个TIMESTAMP类型的字段指定NULL属性，会自动给该字段添加DEFAULT CURRENT_TIMESTAMP和ON UPDATE CURRENT_TIMESTAMP属性。
  - 如果未给表中的第二个及之后的TIMESTAMP类型的字段指定NULL属性，会自动给该字段添加DEFAULT '0000-00-00 00:00:00'属性。
- MySQL中当explicit_defaults_for_timestamp为on时，对TIMESTAMP类型的字段的处理逻辑如下：
  - 向TIMESTAMP类型的字段插入NULL值时，不会将NULL值替换为当前时间戳。
  - 未显式指定NULL/NOT NULL属性的字段，将自动添加NULL属性。
  - 向指定了NOT NULL属性的字段插入NULL值时，严格模式下报错，宽松模式下将插入'0000-00-00 00:00:00'。
  - 不会为任何TIMESTAMP类型的字段自动添加DEFAULT CURRENT_TIMESTAMP和ON UPDATE CURRENT_TIMESTAMP属性。

3.3.2.1.3 字符串数据类型

表 3-136 字符串数据类型

数据类型	与MySQL的差异
CHAR(M)	输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。
VARCHAR(M)	输入格式： • GaussDB的自定义函数参数和返回值不支持长度校验，存储过程参数不支持长度校验，MySQL支持。 • GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警，MySQL不支持。 • 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。

版权所有 © 华为云计算技术有限公司

数据类型	与MySQL的差异
TINYTEXT	● 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 ● 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 ● 索引：MySQL中TINYTEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。
TEXT	● 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 ● 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 ● 索引：MySQL中TEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。
MEDIUMTEXT	● 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 ● 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 ● 索引：MySQL中MEDIUMTEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。
LONGTEXT	● 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 ● 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 ● 索引：MySQL中LONGTEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。

版权所有 © 华为云计算技术有限公司

3.3.2.1.4 二进制数据类型

表 3-137 二进制数据类型

数据类型	与MySQL的差异
BINARY[(M)]	● 输入格式： – 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 – 插入字符串长度小于目标长度时，GaussDB填充符是0x20，MySQL是0x00。 ● 字符集：默认字符集为数据库初始化字符集，MySQL默认类型字符集为BINARY字符集。 ● 输出格式： – JDBC协议输出时BINARY类型的末尾空格显示为空格，MySQL末尾空格显示为\x00。 – 宽松模式下，BINARY类型面对输入超过n的字节数的字符输入（例如中文字符），会将超限的整个字符截断。MySQL中会将超限的整个字符的前几位满足n范围内的字节信息保留，但输出时字符信息显示乱码。 – 在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。 **说明** GaussDB中，由于BINARY类型填充符和\0截断与MySQL的差异，在操作符比较计算，字符串相关系统函数计算，索引匹配，数据导入导出等场景下与MySQL的表现会存在差异。差异场景请见本节示例。
VARBINARY(M)	● 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 字符集：默认字符集为数据库初始化字符集，MySQL默认类型字符集为BINARY字符集。 ● 输出格式： – JDBC协议输出时BINARY类型的末尾空格显示为空格，MySQL末尾空格显示为\x00。 – 在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。

版权所有 © 华为云计算技术有限公司

数据类型	与MySQL的差异
TINYBLOB	● 输入格式： – 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 – 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 主键：MySQL中TINYBLOB类型不支持主键，GaussDB支持。 ● 索引：MySQL中TINYBLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 ● 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 ● 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。
BLOB	● 输入格式： – 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 – 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 主键：MySQL中BLOB类型不支持主键，GaussDB支持。 ● 索引：MySQL中BLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 ● 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 ● 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。
MEDIUMBLOB	● 输入格式： – 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 – 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 主键：MySQL中MEDIUMBLOB类型不支持主键，GaussDB支持。 ● 索引：MySQL中MEDIUMBLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 ● 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 ● 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。

版权所有 © 华为云计算技术有限公司

数据类型	与MySQL的差异
LONGBLOB	● 取值范围：只支持不超过1GB字节长度，MySQL支持4GB-1字节长度。 ● 输入格式： – 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 – 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 主键：MySQL中LONGBLOB类型不支持主键，GaussDB支持。 ● 索引：MySQL中LONGBLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 ● 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 ● 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。
BIT[(M)]	输出格式： ● 所有输出按照二进制字符串形式输出。MySQL中根据ASCII码表转义，无法转义的输出为空。 ● 在MySQL 8.0以上版本，默认会开头补0，GaussDB不会补0。

示例：

```
-- GaussDB场景
m_db=# CREATE TABLE test(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test VALUES(0x8000);
INSERT 0 1
m_db=# SELECT hex(a) FROM test;
        hex
----------------------
 80202020202020202020
(1 row)

m_db=# SELECT * FROM test WHERE hex(a) = 80000000000000000000;
 a
---
(0 rows)

m_db=# CREATE TABLE test2(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test2 VALUES(0x80008000);
INSERT 0 1
m_db=# SELECT hex(a) FROM test2;
        hex
----------------------
 80202020202020202020
(1 row)

m_db=# DROP TABLE test;
DROP TABLE
```


版权所有 © 华为云计算技术有限公司

```
m_db=# DROP TABLE test2;
DROP TABLE

-- MySQL场景
mysql> CREATE TABLE test(a BINARY(10));
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO test VALUES(0x8000);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT hex(a) FROM test;
+----------------------+
| hex(a)               |
+----------------------+
| 80000000000000000000 |
+----------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM test WHERE hex(a) = 80000000000000000000;
+------------+
| a          |
+------------+
| ▒          |
+------------+
1 row in set (0.00 sec)

mysql> CREATE TABLE test2(a binary(10));
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO test2 VALUES(0x80008000);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT hex(a) FROM test2;
+----------------------+
| hex(a)               |
+----------------------+
| 80008000000000000000 |
+----------------------+
1 row in set (0.00 sec)
mysql> DROP TABLE test;
Query OK, 0 rows affected (0.00 sec)
mysql> DROP TABLE test2;
Query OK, 0 rows affected (0.00 sec)
```

3.3.2.1.5 数据类型支持的属性

表 3-138 数据类型支持的属性

MySQL数据库	GaussDB数据库
NULL	支持
NOT NULL	支持
DEFAULT	支持
ON UPDATE	支持
PRIMARY KEY	支持
AUTO_INCREMENT	支持
CHARACTER SET name	支持
COLLATE name	支持

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库
ZEROFILL	支持

使用CREATE TABLE AS方式建表，对VARBINARY类型的字段设置默认值，在使用SHOW CREATE TABLE、DESC或\d 查询的时候回显与MySQL存在差异，GaussDB显示为转换成十六进制后的值，而MySQL显示为原值。

示例：

```
m_db=# CREATE TABLE test_int(
       int_col INT
);
m_db=# CREATE TABLE test_varbinary(
       varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
m_db=# SHOW CREATE TABLE test_varbinary;
     Table      |                              Create Table
----------------+-----------------------------------------------------------------------------
 test_varbinary | SET search_path = public;                                                    +
                | CREATE TABLE test_varbinary (                                               +
                |     varbinary_col varbinary(20) DEFAULT X'6761757373',                      +
                |     int_col integer                                                         +
                | )                                                                           +
                | CHARACTER SET = "UTF8" COLLATE = "utf8mb4_general_ci"                       +
                | WITH (orientation=row, compression=no, storage_type=USTORE, segment=off);
(1 row)
m_db=# DROP TABLE test_int, test_varbinary;

mysql> CREATE TABLE test_int(
       int_col INT
);
mysql> CREATE TABLE test_varbinary(
       varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
mysql> SHOW CREATE TABLE test_varbinary;
+----------------
+-------------------------------------------------------------------------------------------------------------------------+
| Table          | Create
Table                                                                                                                    |
+----------------
+-------------------------------------------------------------------------------------------------------------------------+
| test_varbinary | CREATE TABLE `test_varbinary` (
  `varbinary_col` varbinary(20) DEFAULT 'gauss',
  `int_vol` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |
+----------------
+-------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
mysql> DROP TABLE test_int, test_varbinary;
```

3.3.2.1.6 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换：

- 操作符（比较操作符、运算操作符等）的操作数的数据类型不一致。常见于查询条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新（包括INSERT、UPDATE、MERGE、REPLACE等）的目标列，数据的类型和列的定义类型不一致。
- 显式的类型转换：CAST(expr AS datatype)，将expr表达式类型转换为datatype类型。

版权所有 © 华为云计算技术有限公司

- 集合运算（UNION、EXCEPT）确定最终投影列的目标数据类型后，各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景，根据不同表达式的数据类型， 来决定用于比较或者最终结果的目标数据类型。
- 当操作符的操作数为函数时，可能会导致函数的返回值类型发生隐式转换（如GREATEST、LEAST、IF、NULLIF、IFNULL、ELTM、DAYNAME、UUID、ACOS、ASIN、TIMESTAMP、CONVERTTZ、AVGFINALF等），与单独使用函数时的返回值类型存在差异。

数据类型转换差异点主要分为三类：隐式转换，显式转换和UNION/CASE。

双冒号转换差异点

- GaussDB中使用双冒号将函数入参转换为期望类型可能导致结果超出预期，且MySQL中无双冒号功能。

  示例：

```
m_db=# SELECT POW("12"::VARBINARY,"12"::VARBINARY);
ERROR:  The value is out of range, and an overflow occurred.
CONTEXT:  referenced column: pow

varbinary col
m_db=# CREATE TABLE test_varbinary (
        A VARBINARY(10)
);
m_db=# INSERT INTO test_varbinary VALUES ('12');
m_db=# SELECT POW(A, A) FROM test_varbinary;
      pow
----------------
 8916100448256
(1 row)
```

隐式类型转换差异点

- GaussDB中统一平铺成小类型到小类型的转换规则，MySQL中使用小类型转大类型，大类型转小类型的转换规则。
- GaussDB中隐式转换因数据类型本身差异点，输出格式存在部分行为不一致。
- GaussDB中的隐式转换，BIT数据类型到字符数据类型和二进制数据类型转换，输出存在部分行为不一致。GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。

  示例：

```
m_db=# CREATE TABLE bit_storage (
    VS_COL1 BIT(4),
    VS_COL2 BIT(4),
    VS_COL3 BIT(4),
    VS_COL4 BIT(4),
    VS_COL5 BIT(4),
    VS_COL6 BIT(4),
    VS_COL7 BIT(4),
    VS_COL8 BIT(4)
) DISTRIBUTE BY REPLICATION;
m_db=# CREATE TABLE string_storage (
    VS_COL1 BLOB,
    VS_COL2 TINYBLOB,
    VS_COL3 MEDIUMBLOB,
    VS_COL4 LONGBLOB,
    VS_COL5 TEXT,
    VS_COL6 TINYTEXT,
    VS_COL7 MEDIUMTEXT,
    VS_COL8 LONGTEXT
) DISTRIBUTE BY REPLICATION;
```

 版权所有 © 华为云计算技术有限公司

```
m_db=# INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
m_db=# INSERT INTO string_storage SELECT * FROM bit_storage;
m_db=# SELECT * FROM string_storage;
 VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8
---------+---------+---------+---------+---------+---------+---------+---------
 \x05    | \x05    | \x05    | \x05    | \x05    | \x05    | \x05    | \x05
(1 row)
m_db=# DROP TABLE bit_storage, string_storage;

mysql> CREATE TABLE bit_storage (
    VS_COL1 BIT(4),
    VS_COL2 BIT(4),
    VS_COL3 BIT(4),
    VS_COL4 BIT(4),
    VS_COL5 BIT(4),
    VS_COL6 BIT(4),
    VS_COL7 BIT(4),
    VS_COL8 BIT(4)
);
mysql> CREATE TABLE bit_storage (
    VS_COL1 BIT(4),
    VS_COL2 BIT(4),
    VS_COL3 BIT(4),
    VS_COL4 BIT(4),
    VS_COL5 BIT(4),
    VS_COL6 BIT(4),
    VS_COL7 BIT(4),
    VS_COL8 BIT(4)
);
mysql> INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
mysql> INSERT INTO string_storage SELECT * FROM bit_storage;
mysql> SELECT * FROM string_storage;
+---------+---------+---------+---------+---------+---------+---------+---------+
| VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8 |
+---------+---------+---------+---------+---------+---------+---------+---------+
|       |       |       |       |       |       |       |       |
+---------+---------+---------+---------+---------+---------+---------+---------+
1 row in set (0.00 sec)
mysql> DROP TABLE bit_storage, string_storage;
```

- 中间带有0x00的二进制或十六进制字符串插入二进制数据类型时，GaussDB部分插入，0x00后的字符会截断，MySQL可以完整插入。

  示例：

```
m_db=# CREATE TABLE blob_storage (
       A BLOB
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO blob_storage VALUES (0xBB00BB);
m_db=# SELECT hex(A) FROM blob_storage;
 hex
-----
 BB
(1 row)
m_db=# DROP TABLE blob_storage;

mysql> CREATE TABLE blob_storage (
       A BLOB
);
mysql> INSERT INTO blob_storage VALUES (0xBB00BB);
mysql> SELECT hex(A) FROM blob_storage;
+--------+
| hex(a) |
+--------+
| BB00BB |
+--------+
1 row in set (0.01 sec)
mysql> DROP TABLE blob_storage;
```


版权所有 © 华为云计算技术有限公司

- 中间带有0x00的二进制或十六进制字符串插入字符串数据类型时，GaussDB部分插入，0x00后的字符会截断，MySQL严格模式不允许插入，宽松模式插入空字符串。

  示例：

```
m_db=# CREATE TABLE text_storage (
       A TEXT
);
m_db=# INSERT INTO text_storage VALUES (b'10111011000000000010111011');
m_db=# SELECT hex(A) FROM text_storage;
 hex
-----
 BB
(1 row)
m_db=# DROP TABLE text_storage;

mysql> CREATE TABLE text_storage (
       A TEXT
);
mysql> INSERT INTO text_storage VALUES (b'10111011000000000010111011');
ERROR 1366 (HY000): Incorrect string value: '\xBB\x00\xBB' for column 'A' at row 1
mysql> SELECT hex(A) FROM text_storage;
Empty set (0.00 sec)
mysql> SET SQL_MODE='';
mysql> INSERT INTO text_storage VALUES (b'10111011000000000010111011');
mysql> SELECT hex(A) FROM text_storage;
+--------+
| hex(A) |
+--------+
|        |
+--------+
1 row in set (0.01 sec)
mysql> DROP TABLE text_storage;
```

- WHERE子句中只带有普通字符串，GaussDB中't'、'true'、'yes'、 'y'、'on'返回TRUE，'no'、'f'、'off'、'false'、'n'返回FALSE，其余字符串报错。MySQL通过字符串转换为INT1判断返回TRUE/FALSE。

  示例：

```
m_db=# CREATE TABLE test_where (
       A INT
);
m_db=# INSERT INTO test_where VALUES (1);
m_db=# SELECT * FROM test_where WHERE '111';
ERROR:  invalid input syntax for type boolean: "111"
LINE 1: SELECT * FROM test_where WHERE '111';
m_db=# DROP TABLE test_where;

mysql> CREATE TABLE test_where (
       A INT
);
mysql> INSERT INTO test_where VALUES (1);
mysql> SELECT * FROM test_where WHERE '111';
+------+
| a    |
+------+
|    1 |
+------+
1 row in set (0.01 sec)
mysql> DROP TABLE test_where;
```

- 对于YEAR类型的输入，在将字符串转换为整型的过程中，MySQL考虑科学计数法，GaussDB暂不支持，统一做截断处理。

  示例：

```
m_db=# CREATE TABLE test_year (
       A YEAR
);
```


版权所有 © 华为云计算技术有限公司

```
m_db=# SET sql_mode = '';
m_db=# INSERT INTO test_year VALUES ('2E3x');
WARNING:  Data truncated for column.
LINE 1: INSERT INTO test_year VALUES ('2E3x');
                                      ^
CONTEXT:  referenced column: a
m_db=# SELECT * FROM test_year ORDER BY A;
  a
------
 2002
(1 row)
m_db=# DROP TABLE test_year;

mysql> CREATE TABLE test_year (
        A YEAR
);
mysql> INSERT INTO test_year VALUES ('2E3x');
mysql> SELECT * FROM test_year ORDER BY A;
+------+
| a    |
+------+
| 2000 |
+------+
1 row in set (0.01 sec)
mysql> DROP TABLE test_year;
```

- MySQL 5.7字符串转整数时存在两套不同的行为，其对字符串前缀空白的判断标准不一致。会导致同一字符串，在不同使用场景下，转整数的结果不一致。MySQL8.0及GaussDB在字符串转整数时，使用一致的行为，仅将空格及制表符按前缀空白处理。

```
-- GaussDB，各场景下字符串转整数结果一致
m_db=# SELECT cast('\n1' as signed);
WARNING:  The INTEGER value '
1' is incorrect.
CONTEXT:  referenced column: cast
 cast
------
    0
(1 row)

m_db=# SELECT left('Hello', '\n1');
WARNING:  The INTEGER value '
1' is incorrect.
CONTEXT:  referenced column: left
 left
------

(1 row)

m_db=# SELECT left('Hello', concat('\n', '1'));
 left
------

(1 row)

-- MySQL5.7，left嵌套concat函数时，将'\n'按前缀空白跳过处理，其余场景未跳过'\n'。
mysql> SELECT cast('\n1' as signed);
+-----------------------+
| cast('\n1' as signed) |
+-----------------------+
|                     0 |
+-----------------------+
1 row in set, 1 warning (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
mysql>  SELECT left('Hello', '\n1');
+----------------------+
| left('Hello', '\n1') |
```

 版权所有 © 华为云计算技术有限公司

```
+----------------------+
|                      |
+----------------------+
1 row in set, 2 warnings (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
mysql>  SELECT left('Hello', concat('\n', '1'));
+-----------------------------------+
| left('Hello', concat('\n', '1')) |
+-----------------------------------+
| H                                 |
+-----------------------------------+
1 row in set (0.00 sec)

-- MySQL8.0，各场景下字符串转整数结果一致
mysql>  SELECT cast('\n1' as signed);
+-----------------------+
| cast('\n1' as signed) |
+-----------------------+
|                     0 |
+-----------------------+
1 row in set, 1 warning (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
mysql>  SELECT left('Hello', '\n1');
+----------------------+
| left('Hello', '\n1') |
+----------------------+
|                      |
+----------------------+
1 row in set, 2 warnings (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
mysql> SELECT left('Hello', concat('\n', '1'));
+-----------------------------------+
| left('Hello', concat('\n', '1')) |
+-----------------------------------+
|                                   |
+-----------------------------------+
1 row in set, 2 warnings (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
Warning (Code 1292): Truncated incorrect INTEGER value: '
1'
```

显式类型转换差异点

- GaussDB中平铺成对各目标类型的转换规则，MySQL中使用C++多态重载函数，在嵌套场景中存在不一致行为。

  示例：

```
m_db=# SELECT CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED);
WARNING:  Truncated incorrect INTEGER value: '2023-01-01'
CONTEXT:  referenced column: cast
 cast
------
 2023
(1 row)

mysql> SELECT CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED);
+-----------------------------------------------------------+
```

 版权所有 © 华为云计算技术有限公司

```
| CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED) |
+---------------------------------------------------------+
|                                                20230101 |
+---------------------------------------------------------+
```

UNION，CASE 和相关构造差异点

- POLYGON + NULL、POINT + NULL、POLYGON + POINT组合在MySQL中均返回GEOMETRY类型，GaussDB中未涉及，暂时当做报错处理。
- SET和ENUM两种类型暂未支持，暂时当做报错处理。
- 常量类型和其他类型做类型聚合的时候，输出类型的精度为其他类型的精度。如“SELECT "helloworld" UNION SELECT p FROM t;”的结果的精度为属性p的精度。
- 定点常量和不带精度约束的类型（非字符串类型如int、bool、year等，聚合结果类型为定点类型）聚合时，精度约束会按照定点数默认精度31输出。
- merge rule差异：

  MySQL 5.7中YEAR和TINYINT、INT、MEDIUMINT、BIGINT、BOOL聚合的结果类型为带UNSIGNED的类型，GaussDB的结果类型为不带UNSIGNED的类型；MySQL中BIT和INT、NUMERIC、FLOAT、DOUBLE等数值类型的聚合类型为VARBINARY类型，GaussDB中BIT和INT|NUMERIC聚合为NUMERIC类型，和FLOAT或者DOUBLE类型聚合时，结果为DOUBLE类型，和无符号整型聚合时为UINT8类型。
- MySQL中BINARY和CHAR填充字符不相同，BINARY填充'\0'，CHAR填充空格，GaussDB中BINARY和CHAR都是填充空格。

3.3.2.2 系统函数

3.3.2.2.1 系统函数兼容性概述

GaussDB数据库兼容绝大多数MySQL的系统函数，但存在部分差异。

当前存在原GaussDB的系统函数和MySQL系统函数同名，但是M-Compatibility兼容模式下尚未支持这些函数的情况；一部分未支持的同名函数会提示用户在M-Compatibility兼容模式下不支持，另外一部分同名函数仍然保持原GaussDB系统函数的行为。同名函数会与MySQL的行为产生较大差异，因此建议用户尽量避免使用这些同名函数，只使用M-Compatibility兼容模式下的系统函数。

具体同名函数列表如下：

表 3-139 M-Compatibility 兼容模式下提示不支持的同名函数

cot	isEmpty	mod	octet_length	overlaps
point	radians	regexp_instr	regexp_like	regexp_replace
regexp_substr	stddev_pop	stddev_samp	var_pop	var_samp
variance	-	-	-	-

版权所有 © 华为云计算技术有限公司

表 3-140 M-Compatibility 兼容模式下保持原 GaussDB 系统函数行为的同名函数

ceil	decode	encode	format	instr
position	round	stddev	row_num	-

说明

- MySQL数据库支持通过可加载函数接口，向MySQL中添加自定义函数，在调用此类函数时，函数的入参支持指定别名。GaussDB不支持可加载函数接口，在调用函数时，函数入参不支持指定别名。
- M-Compatibility模式下，系统函数存在以下公共差异：
  - 系统函数的返回值类型仅考虑入参node类型为Var（表中数据）和Const（常量输入）类型时的情况与MySQL保持一致，其他情况（如入参为运算表达式、函数表达式等）可能返回值的类型与MySQL有差异。
  - 当聚合函数以其他函数、操作符或SELECT子句等表达式作为入参时（如SELECT sum(abs(n)) FROM t;），聚合函数将获取不到入参表达式传递的精度信息，导致函数的结果精度与MySQL有差异。
  - 系统函数的调用不推荐使用pg_catalog.func_name()形式的调用，当被调用函数存在语法形式的入参时（如SELECT substr('demo' from 1 for 2)），函数的调用可能存在错误。
  - 对于支持多个入参的系统函数，当入参存在NULL时，GaussDB不会对入参进行校验，直接返回NULL；MySQL会对NULL之外的入参进行校验，当存在不合法的入参时，函数返回NULL并产生WARNING。
  - MySQL以带反引号的形式调用系统函数时，不区分大小写。GaussDB以带反引号的形式调用系统函数时，需指定函数名为全小写。

```
-- GaussDB
m_db=# SELECT `abs`(1);
 abs
-----
   1
(1 row)

m_db=# SELECT `Abs`(1);
ERROR:  Function Abs(integer) does not exist.
LINE 1: select `Abs`(1);
               ^
HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
CONTEXT:  referenced column: Abs

-- MySQL
mysql> SELECT `abs`(1);
+----------+
| `abs`(1) |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT `Abs`(1);
+----------+
| `Abs`(1) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)
```


版权所有 © 华为云计算技术有限公司

3.3.2.2.2 流程控制函数

表 3-141 流程控制函数列表

函数名	与MySQL的差异
IF()	当第一个参数为TRUE且第三个参数表达式中存在隐式类型转换错误，或者第一个参数为FALSE且第二个参数表达式中存在隐式类型转换错误时，MySQL会忽略该错误，GaussDB会提示类型转换错误。
IFNULL()	第一个参数不为NULL且第二个参数表达式中存在隐式类型转换错误时，MySQL会忽略该错误，GaussDB会提示类型转换错误。
NULLIF()	函数返回值类型在MySQL 5.7和MySQL 8.0中存在差异，考虑到MySQL 8.0更合理，因此函数返回值类型兼容MySQL 8.0。

3.3.2.2.3 日期和时间函数

以下为GaussDB数据库M-Compatibility兼容性日期时间函数公共说明，与MySQL行为一致。

- 函数入参为时间类型表达式的情况：

  时间类型表达式主要包括TEXT、DATETIME、DATE或TIME，但所有可以隐式转换为时间表达式的类型都可以作为入参，比如数字类型可以通过先隐式转化为TEXT，再作为时间类型表达式生效。

  但是，不同函数的具体生效情况会有所不同。例如：DATEDIFF函数仅计算日期之间的差值，因此时间表达式会被解析为日期；而TIMESTAMPDIFF函数在计算时间差值时，会根据UNIT参数来决定将时间表达式解析为DATE、TIME或DATETIME。

- 函数入参为无效日期的情况：

  一般而言，日期时间函数支持DATE、DATETIME的范围和MySQL保持一致。DATE支持的范围为'0000-01-01'到'9999-12-31'，DATETIME支持的范围为'0000-01-01 00:00:00'到'9999-12-31 23:59:59'。虽然GaussDB支持的DATE、DATETIME范围大于MySQL，但是越界仍然算无效日期。

  大部分时间函数对于入参为无效时间时，会告警并返回NULL，只有能通过cast正常转换的日期，才是正常合理的日期。

GaussDB M-Compatibility兼容性框架下GaussDB的大部分日期时间函数与MySQL一致，一些函数的差异如**表3-142**所示：

表 3-142 日期和时间函数列表

函数名	与MySQL的差异
ADDDATE()	-
ADDTIME()	-

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异

CONVERT_TZ()	• MySQL支持使用'system'指定时区为系统时区，GaussDB不支持使用'system'指定时区。
	• CONVERT_TZ执行转换前，会校验时间是否合法且在TIMESTAMP类型范围内（'1970-01-01 00:00:01' UTC ~ '2038-01-19 03:14:07' UTC），时间合法但不在范围内则不转换，返回原值。MySQL校验第一个入参经第二个入参（from_tz）转换为UTC后的值是否满足要求，GaussDB则仅校验第一个入参的原始值。
	-- GaussDB
	m_db=# SELECT convert_tz('1970-01-01 00:00:01', '-01:00', '+01:00'),
	convert_tz('1970-01-01 00:00:01', '+01:00', '+03:00');
	convert_tz | convert_tz
	---+---
	1970-01-01 02:00:01 | 1970-01-01 02:00:01
	(1 row)
	
	m_db=# SELECT convert_tz('1970-01-01 00:00:00', '-01:00', '+01:00'),
	convert_tz('1970-01-01 00:00:00', '+01:00', '+03:00');
	convert_tz | convert_tz
	---+---
	1970-01-01 00:00:00 | 1970-01-01 00:00:00
	(1 row)
	
	-- MySQL
	mysql> SELECT convert_tz('1970-01-01 00:00:01', '-01:00', '+01:00'),
	convert_tz('1970-01-01 00:00:01', '+01:00', '+03:00');
	+---
	+---+
	| convert_tz('1970-01-01 00:00:01', '-01:00', '+01:00') |
	convert_tz('1970-01-01 00:00:01', '+01:00', '+03:00') |
	+---
	+---+
	| 1970-01-01 02:00:01 | 1970-01-01
	00:00:01 |
	+---
	+---+
	1 row in set (0.00 sec)
	
	mysql> SELECT convert_tz('1970-01-01 00:00:00', '-01:00', '+01:00'),
	convert_tz('1970-01-01 00:00:00', '+01:00', '+03:00');
	+---
	+---+
	| convert_tz('1970-01-01 00:00:00', '-01:00', '+01:00') |
	convert_tz('1970-01-01 00:00:00', '+01:00', '+03:00') |
	+---
	+---+
	| 1970-01-01 02:00:00 | 1970-01-01
	00:00:00 |
	+---
	+---+
	1 row in set (0.00 sec)

CURDATE()	-
CURRENT_DATE() /CURRENT_DATE	-
CURRENT_TIME() /CURRENT_TIME	MySQL入参整型值会按照一字节最大值255整数回绕（例：SELECT CURRENT_TIME(257) == SELECT CURRENT_TIME(1)）。 GaussDB只支持[0,6]合法值，其他值报错。

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
CURRENT_TIMESTAMP()/ CURRENT_TIMESTAMP	
CURTIME()	
LOCALTIME()/ LOCALTIME	
LOCALTIMESTAMP/ LOCALTIMESTAMP()	
NOW()	
SYSDATE()	
UTC_TIME()	
UTC_TIMESTAMP()	
DATE()	-
DATE_ADD()	-
DATE_FORMAT()	-
DATE_SUB()	-
DATEDIFF()	-
DAY()	-
DAYNAME()	-
DAYOFMONTH()	-
DAYOFWEEK()	-
DAYOFYEAR()	-

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异

EXTRACT()	• 当EXTRACT(unit FROM date)中unit入参的取值为DAY_HOUR、DAY_MINUTE、DAY_SECOND或DAY_MICROSECOND时GaussDB会将date入参按datetime类型的格式要求进行解析，MySQL会将date入参按time类型的格式要求进行解析。
	-- GaussDB
	m_db=# SELECT EXTRACT(DAY_HOUR FROM '10:30:45') AS res;
	WARNING: Incorrect datetime value: '10:30:45'
	CONTEXT: referenced column: res
	res
	----- 	
	
	(1 row)
	
	-- MySQL
	mysql> SELECT EXTRACT(DAY_HOUR FROM '10:30:45') AS res;
	+------+
	| res |
	+------+
	| 10 |
	+------+
	1 row in set (0.00 sec)
	• 当EXTRACT(unit FROM date)中date入参的类型为datetime类型并且unit入参的取值为DAY_HOUR、DAY_MINUTE、DAY_SECOND或DAY_MICROSECOND时，GaussDB的结果都会返回DAY的部分，MySQL的结果只有在date入参为表数据时会返回DAY的部分，例如：
	-- GaussDB的行为：
	m_db=# DROP TABLE IF EXISTS t_datetime;
	DROP TABLE
	
	m_db=# CREATE TABLE t_datetime(c1 datetime(6));
	CREATE TABLE
	
	m_db=# INSERT INTO t_datetime VALUES('2023-01-27 10:11:12.400000');
	INSERT 0 1
	
	-- date入参为表数据时，结果会返回DAY的部分。
	m_db=# SELECT EXTRACT(DAY_MICROSECOND FROM c1) FROM t_datetime;
	extract
	---------------- 	
	27101112400000
	(1 row)
	
	-- date入参为常量数据时，结果也会返回DAY的部分。
	m_db=# SELECT EXTRACT(DAY_MICROSECOND FROM cast('2023-01-27 10:11:12.400000' as datetime(6)));
	extract
	---------------- 	
	27101112400000
	(1 row)
	
	m_db=# DROP TABLE IF EXISTS t_datetime;
	DROP TABLE
	
	-- MySQL的行为：
	mysql> DROP TABLE IF EXISTS t_datetime;
	Query OK, 0 rows affected (0.04 sec)
	
	mysql> CREATE TABLE t_datetime(c1 datetime(6));
	Query OK, 0 rows affected (0.05 sec)

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异

	mysql> INSERT INTO t_datetime VALUES('2023-01-27 10:11:12.400000');
	Query OK, 1 row affected (0.00 sec)
	
	-- date入参为表数据时，结果会返回DAY的部分。
	mysql> SELECT EXTRACT(DAY_MICROSECOND FROM c1) FROM t_datetime;
	+----------------------------------+
	| EXTRACT(DAY_MICROSECOND FROM c1) |
	+----------------------------------+
	| 27101112400000 |
	+----------------------------------+
	1 row in set (0.01 sec)
	
	-- date入参为常量数据时，结果不会返回DAY的部分。
	mysql> SELECT EXTRACT(DAY_MICROSECOND FROM cast('2023-01-27 10:11:12.400000' as datetime(6)));
	+---+
	| EXTRACT(DAY_MICROSECOND FROM cast('2023-01-27 10:11:12.400000' as datetime(6))) |
	+---+
	| 101112400000 |
	+---+
	1 row in set (0.00 sec)
	
	mysql> DROP TABLE IF EXISTS t_datetime;
	Query OK, 0 rows affected (0.04 sec)

FROM_DAYS()	-
FROM_UNIXTIME()	-
GET_FORMAT()	-
HOUR()	-
LAST_DAY()	-
MAKEDATE()	-
MAKETIME()	分布式下推场景下当TIME类型秒位无精度时，MYSQL默认补齐6个0，GaussDB不做补齐。
MICROSECOND()	-
MINUTE()	-
MONTH()	-
MONTHNAME()	-

版权所有 © 华为云计算技术有限公司

<table>
<tr><th>函数名</th><th>与MySQL的差异</th></tr>
<tr><td>PERIOD_ADD()
PERIOD_DIFF()</td><td>MySQL 8.0修复了以下问题，在以下场景中，该函数的行为与MySQL 8.0版本保持一致，与MySQL 5.7版本存在差异：
整数溢出处理的行为：
MySQL在5.7版本，此函数入参和结果的最大值都为2^32=4294967296，在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题。GaussDB不存在该问题。
负数period的表现：
MySQL在5.7版本，会将负数年份解析为异常值而不是报错。GaussDB入参或结果（如100年1月减去10000月）出现负数时报错。
period月份越界的表现：
MySQL在5.7版本中，若月份大于12或等于0，例如200013、199900，会将其顺延到之后的年份，或者将0月作为上一年12月处理。GaussDB会对越界月份进行报错。
入参无法严格转换为整数时的行为：
MySQL在5.7版本中，当传入的参数无法严格转换为整数时，会将其强制转换为整数值，GaussDB进行报错。<pre>-- GaussDB
m_db=# SELECT period_add('20.1', 1);
WARNING: The INTEGER value '20.1' is incorrect.
CONTEXT: referenced column: period_add
ERROR: Incorrect arguments to period_add
CONTEXT: referenced column: period_add
m_db=# SELECT period_add('20ab', 1);
WARNING: The INTEGER value '20ab' is incorrect.
CONTEXT: referenced column: period_add
ERROR: Incorrect arguments to period_add
CONTEXT: referenced column: period_add

-- MySQL
mysql> SELECT period_add('20.1', 1);
+-----------------------+
| period_add('20.1', 1) |
+-----------------------+
| 200109 |
+-----------------------+
1 row in set, 1 warning (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '20.1'
mysql> SELECT period_add('20ab', 1);
+-----------------------+
| period_add('20ab', 1) |
+-----------------------+
| 200109 |
+-----------------------+
1 row in set, 1 warning (0.00 sec)

Warning (Code 1292): Truncated incorrect INTEGER value: '20ab'</pre></td></tr>
<tr><td>QUARTER()</td><td>-</td></tr>
<tr><td>SEC_TO_TIME()</td><td>-</td></tr>
<tr><td>SECOND()</td><td>-</td></tr>
<tr><td>STR_TO_DATE()</td><td>返回值类型与MySQL有差异，GaussDB返回的是text，MySQL返回的是datetime、date。</td></tr>
</table>

 版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
SUBDATE()	-
SUBTIME()	-
TIME()	-
TIME_FORMAT()	-
TIME_TO_SEC()	-
TIMEDIFF()	-
TIMESTAMP()	-
TIMESTAMPADD()	-
TIMESTAMPDIFF()	-
TO_DAYS()	-
TO_SECONDS()	-
UNIX_TIMESTAMP()	MySQL会根据入参是否存在小数位，决定返回定点型还是整型。当前GaussDB在内层嵌套操作符或函数时，返回的类型与MySQL可能存在不同。当内层节点返回定点、浮点、字符型、时间类型（不包括DATE类型）时，MySQL可能返回整型，GaussDB会返回定点型。
UTC_DATE()	-
WEEK()	-
WEEKDAY()	-
WEEKOFYEAR()	-
YEAR()	-
YEARWEEK()	-

3.3.2.2.4 字符串函数

表 3-143 字符串函数列表

函数名	与MySQL的差异
ASCII()	-
BIT_LENGTH()	-

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
CHAR()	当返回值为非法字符时，GaussDB会直接返回非法字符，MySQL在宽松模式下返回截断后的字段串，严格模式下返回NULL。
CHAR_LENGTH()	GaussDB此函数如果数据库字符集是SQL_ASCII，CHAR_LENGTH()会返回字节数而非字符数。
CHARACTER_LENGTH()	GaussDB此函数如果数据库字符集是SQL_ASCII，CHARACTER_LENGTH()会返回字节数而非字符数。
CONCAT()	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
CONCAT_WS()	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
HEX()	-
LENGTH()	-
LPAD()	• MySQL默认最大填充长度为1398101，GaussDB默认最大长度为1048576。在不同字符集下，最大填充长度会有差异，例如字符集为GBK时，GaussDB默认最大长度为2097152。 • 当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。 • 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
REPEAT()	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
REPLACE()	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
RPAD()	• MySQL默认最大填充长度为1398101，GaussDB默认最大长度为1048576。在不同字符集下，最大填充长度会有差异，例如字符集为GBK时，GaussDB默认最大长度为2097152。 • 当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。 • 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
SPACE()	-
STRCMP()	当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。
FIND_IN_SET() LCASE() LEFT() LOWER() LTRIM() REVERSE() RIGHT() RTRIM() SUBSTR() SUBSTRING() SUBSTRING_INDEX() TRIM() UCASE() UPPER()	当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
UNHEX()	MySQL的返回值类型为BINARY、VARBINARY、BLOB、MEDIUMBLOB或LONGBLOB；GaussDB返回值类型固定为LONGBLOB。

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
FIELD()	-
FORMAT()	-

版权所有 © 华为云计算技术有限公司

3.3.2.2.5 强制转换函数

表 3-144 强制转换函数列表

函数名	与MySQL的差异

CAST()	• GaussDB不支持使用CAST(expr AS CHAR[(N)] charset_info或者CAST(expr AS NCHAR[(N)])转换字符集。
	• GaussDB支持使用CAST(expr AS FLOAT[(p)])或CAST(expr AS DOUBLE)将表达式转换为浮点类型，MySQL 5.7版本不支持此转换。
	• 使用CAST(expr AS FLOAT[(p)])函数时，若参数p满足0 ≤ p < 24，函数返回值类型为FLOAT，但该函数在CREATE TABLE/VIEW [AS] SELECT语句中使用时，最终创建的表/视图对应字段类型会变为DOUBLE，此行为与MySQL 8.0版本存在不一致。
	• GaussDB不支持使用CAST(expr AS JSON)将表达式转换为JSON。
	• 对于CAST嵌套子查询场景，如果子查询语句返回的是FLOAT类型，GaussDB返回的是准确的数值，MySQL 5.7版本返回失真数值，BINARY函数使用CAST实现，同理。
	--GaussDB
	m_db=# CREATE TABLE sub_query_table(myfloat float) DISTRIBUTE BY REPLICATION;
	CREATE TABLE
	m_db=# INSERT INTO sub_query_table(myfloat) VALUES (1.23);
	INSERT 0 1
	m_db=# SELECT binary(SELECT myfloat FROM sub_query_table) FROM sub_query_table;
	binary
	--------- 	
	1.23
	(1 row)
	m_db=# SELECT cast((SELECT myfloat FROM sub_query_table) AS char);
	cast
	------ 	
	1.23
	(1 row)
	--MySQL 5.7
	mysql> CREATE TABLE sub_query_table(myfloat float);
	Query OK, 0 rows affected (0.02 sec)
	mysql> INSERT INTO sub_query_table(myfloat) VALUES (1.23);
	Query OK, 1 row affected (0.00 sec)
	mysql> SELECT binary(SELECT myfloat FROM sub_query_table) FROM sub_query_table;
	+---+
	| binary(SELECT myfloat FROM sub_query_table) |
	+---+
	| 1.2300000190734863 |
	+---+
	1 row in set (0.00 sec)
	mysql> SELECT cast((SELECT myfloat FROM sub_query_table) AS char);
	+---+
	| cast((SELECT myfloat FROM sub_query_table) AS char) |
	+---+
	| 1.2300000190734863 |
	+---+
	1 row in set (0.00 sec)

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异

	● 当使用CAST(expr as CHAR[(p)])时，MySQL支持指定字符集，GaussDB 不支持。
	-- GaussDB
	m_db=# SELECT CAST('hello' AS CHAR CHARACTER SET utf8);
	ERROR: syntax error at or near "CHARACTER"
	LINE 1: SELECT CAST('hello' AS CHAR CHARACTER SET utf8);
	^
	-- Mysql 5.7
	mysql> SELECT CAST('hello' AS CHAR CHARACTER SET utf8);
	+----------+
	| CAST('hello' AS CHAR CHARACTER SET utf8) |
	+----------+
	| hello |
	+----------+
	1 row in set (0.00 sec)
	● 当使用CAST(expr as CHAR[(p)])，返回结果为非法字符时，MySQL 5.7及GaussDB直接返回非法字符，MySQL 8.0在宽松模式下返回截断后的字段串，严格模式下返回NULL。

CONVERT()	● GaussDB不支持使用CONVERT(expr, CHAR[(N)] charset_info或者CAST(expr, NCHAR[(N)])转换字符集。 ● GaussDB支持使用CONVERT(expr, FLOAT[(p)])或CONVERT(expr, DOUBLE)将表达式转换为浮点类型，MySQL 5.7版本不支持此转换。 ● 使用CONVERT(expr, FLOAT[(p)])函数时，若参数p满足0 ≤ p < 24，函数返回值类型为FLOAT，但该函数在CREATE TABLE/VIEW [AS] SELECT语句中使用时，最终创建的表/视图对应字段类型会变为DOUBLE，此行为与MySQL 8.0版本存在不一致。 ● GaussDB不支持使用CONVERT(expr, JSON)将表达式转换为JSON。

3.3.2.2.6 加密函数

表 3-145 加密函数列表

函数名	与MySQL的差异
AES_DECRYPT()	● ecb为不安全加密模式，GaussDB不支持，默认为cbc模式。 ● GaussDB中，当指定数据库使用的字符编码是SQL_ASCII时，服务器把字节值0~127根据ASCII标准解释，而字节值128~255则当作无法解析的字符；如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符。 ● MySQL的返回值类型为BINARY、VARBINARY、BLOB、MEDIUMBLOB、LONGBLOB，GaussDB返回值类型固定为LONGBLOB。
AES_ENCRYPT()	
SHA()/SHA1()	-

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
SHA2()	-

3.3.2.2.7 比较函数

表 3-146 比较函数列表

函数名	与MySQL的差异
COALESCE()	union distinct场景下，返回值精度与MySQL不完全一致。 ● 当第一个不为NULL的参数的后续参数表达式中存在隐式类型转换错误时，MySQL会忽略该错误，GaussDB会提示类型转换错误。当参数为MIN函数、MAX函数时，返回值类型与MySQL不一致。 ● 当函数传入的参数均为整型，且存在无符号整型的入参时，MySQL 5.7返回值类型为bigint，当结果大于bigint上限时结果会被截断。GaussDB及MySQL 8.0返回值类型为decimal，当参数大于bigint上限时结果不会被截断。
INTERVAL()	-
GREATEST() LEAST()	● 当函数传入的参数存在数据类型不一致的情况时，函数的返回值类型与MySQL存在差异。 ● 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。 ● 当该函数入参含有NULL且在WHERE关键字之后调用，返回结果与MySQL 5.7不一致，此处为MySQL 5.7存在的问题，MySQL 8.0修复了该问题，目前GaussDB和MySQL 8.0保持一致。
ISNULL()	-

版权所有 © 华为云计算技术有限公司

3.3.2.2.8 聚合函数

表 3-147 聚合函数列表

函数名	与MySQL的差异
AVG()	● GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 ● GaussDB中当expr中的列为BIT、BOOL、整数类型，且所有行的和超过BIGINT的范围时，会发生溢出导致整数翻转。 ● GaussDB中当expr中的类型为BIT类型，返回值类型为double，MySQL的返回值类型为decimal。
BIT_AND()	-
BIT_OR()	-
BIT_XOR()	-
COUNT()	GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。
GROUP_CONCAT()	● GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 ● GROUP_CONCAT返回类型为二进制类型时，只返回BLOB类型，其他情况返回TEXT类型。MySQL根据返回长度还会返回longtext/tinytext/longblob/tinyblob类型。 ● GaussDB中当GROUP_CONCAT参数中同时有DISTINCT和ORDER BY语法时，所有ORDER BY后的表达式必须也在DISTINCT的表达式之中。 ● GaussDB中GROUP_CONCAT(... ORDER BY 数字)不代表按照第几个参数的顺序，数字只是一个常量表达式，相当于不排序。 ● GaussDB中使用参数group_concat_max_len限制GROUP_CONCAT最大返回长度，超长截断，目前能返回的最大长度是1073741823，小于MySQL。

版权所有 © 华为云计算技术有限公司

<table>
<tr><th>函数名</th><th>与MySQL的差异</th></tr>
<tr><td>MAX()
MIN()</td><td>

GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。
当参数为非表字段时，MAX函数、MIN函数返回值类型和MySQL 5.7不一致。
当参数为float类型时，MAX函数、MIN函数的返回值结果与MySQL 5.7一致。MySQL5.7和MySQL8.0有行为差异，导致MAX函数、MIN函数嵌套CAST(expr AS FLOAT[(p)])时的返回结果与MySQL8.0有差异。

<pre>
-- GaussDB：
m_db=# CREATE TABLE t1(c1 float);
CREATE TABLE

m_db=# INSERT INTO t1 VALUES(1.2);
INSERT 0 1

m_db=# SELECT MAX(c1) FROM t1;
        max

 1.2000000476837158
(1 row)

m_db=# SELECT MAX(CAST(1.2 AS FLOAT));
        max

 1.2000000476837158
(1 row)

m_db=# DROP TABLE t1;
DROP TABLE

-- MySQL 5.7：
mysql> CREATE TABLE t1(c1 float);
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO t1 VALUES(1.2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT MAX(c1) FROM t1;
+--------------------+
| MAX(c1) |
+--------------------+
| 1.2000000476837158 |
+--------------------+
1 row in set (0.00 sec)
-- MySQL5.7不支持CAST(expr AS FLOAT[(p)])表达式
mysql> SELECT MAX(CAST(1.2 AS FLOAT));
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MySQL server version for the right
syntax to use near 'FLOAT))' at line 1

mysql> DROP TABLE t1;
Query OK, 0 rows affected (0.01 sec)

-- MySQL 8.0：
mysql> CREATE TABLE t1(c1 float);
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO t1 VALUES(1.2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT MAX(c1) FROM t1;
+----------+
</pre>
</td></tr>
</table>

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
		MAX(c1) | +---------+ | 1.2 | +---------+ 1 row in set (0.00 sec) mysql> SELECT MAX(CAST(1.2 AS FLOAT)); +-------------------------+ | MAX(CAST(1.2 AS FLOAT)) | +-------------------------+ | 1.2 | +-------------------------+ 1 row in set (0.00 sec) mysql> DROP TABLE t1; Query OK, 0 rows affected (0.01 sec)
SUM()	● GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 ● GaussDB中当expr中的列为BIT、BOOL、整数类型，且所有行的和超过BIGINT的范围时，会发生溢出导致整数翻转。 ● GaussDB中当expr中的类型为BIT类型，返回值类型为double，MySQL的返回值类型为decimal。

3.3.2.2.9 数字操作函数

表 3-148 数字操作函数列表

函数名	与MySQL的差异
ABS()	-
ACOS()	-
ASIN()	-
ATAN()	-
ATAN2()	-
CEILING()	部分操作结果类型与MySQL不一致。对于推导结果类型为NUMERIC或者整型的，如果能够被整型类型存储，MySQL中结果类型为整型；GaussDB中类型仍然为NUMERIC类型。
COS()	-
DEGREES()	-
EXP()	-

版权所有 © 华为云计算技术有限公司

函数名	与MySQL的差异
FLOOR()	FLOOR函数的返回值类型与MySQL的有差异：入参类型为INT，GaussDB返回值类型为BIGINT，MySQL返回值类型为INT。 部分操作结果类型与MySQL不一致。对于推导结果类型为NUMERIC或者整型的，如果能够被整型类型存储，MySQL中结果类型为整型；GaussDB中类型仍然为NUMERIC类型。
LN()	-
LOG()	-
LOG10()	-
LOG2()	-
PI()	PI函数的返回值精度与MySQL的有差异：MySQL中PI函数的结果仅保留四舍五入之后的小数后6位，而GaussDB的结果会保留四舍五入之后的小数后15位。
POW()	-
POWER()	-
RAND()	-
SIGN()	-
SIN()	-
SQRT()	-
TAN()	-
TRUNCATE()	-
CEIL()	-

3.3.2.2.10 其他函数

表 3-149 其他函数列表

函数名	与MySQL的差异
DATABASE()	-
UUID()	-
UUID_SHORT()	-

3.3.2.3 操作符

GaussDB数据库兼容绝大多数MySQL的操作符，但存在部分差异。如未列出，操作符行为默认为GaussDB原生行为，目前存在MySQL不支持但是GaussDB支持的语句，在

版权所有 © 华为云计算技术有限公司

MySQL兼容性下，这类语句通常为系统内部使用，因此不建议使用。建议按照MySQL中的使用规则对操作符进行组合使用。

操作符差异

- ORDER BY排序对NULL值处理的差异。MySQL在排序时会将NULL值排序在前面；GaussDB默认将NULL值默认排在最后面。GaussDB可以通过NULLS FIRST和NULLS LAST设置NULL值排序顺序。
- 有ORDER BY时，GaussDB输出顺序与MySQL一致。没有ORDER BY时，GaussDB不保证结果有序。
- 使用MySQL操作符时需要用括号表达式严格确保表达式的结合性，否则执行报错。例如：SELECT 1 regexp ('12345' regexp '123');。

  GaussDB M-Compatibility操作符不用括号严格表达的结合性也能成功执行。
- NULL值显示不同。MySQL会将NULL显示为“NULL”；GaussDB将NULL值显示为空值。

  MySQL输出结果：

  ```
  mysql> SELECT NULL;
  +------+
  | NULL |
  +------+
  | NULL |
  +------+
  1 row in set (0.00 sec)
  ```

  GaussDB输出结果：

  ```
  m_db=# SELECT NULL;
   ?column?
  ----------

  (1 row)
  ```
- 操作符执行后，列名显示不一致。MySQL会将NULL显示为“NULL”；GaussDB将NULL值显示为空值。
- 字符串转double遇到非法字符串时，告警信息不一致。MySQL在常量非法字符串报错，字段非法字符串不报错；GaussDB在常量非法字符串和字段非法字符串都报错。
- 比较操作符返回结果显示不同。MySQL返回1/0；GaussDB返回t/f。

表 3-150 操作符

操作符	与MySQL的差异
<>	MySQL支持索引，GaussDB不支持索引。
<=>	MySQL支持索引，GaussDB不支持索引、hash连接和合并连接。

版权所有 © 华为云计算技术有限公司

操作符	与MySQL的差异

行表达式	• MySQL支持<=>操作符行比较、GaussDB不支持<=>操作符行比较。
	• MySQL不支持行表达式与NULL比较。GaussDB支持<、<=、=、>=、>、<>操作符对行表达式与NULL值比较。
	• MySQL不支持IS NULL、ISNULL对行表达式的操作。GaussDB支持IS NULL，但不支持ISNULL对行表达式的操作。
	• 操作符对于行表达式的不支持的操作，GaussDB错误信息与MySQL不一致。
	GaussDB：
	m_db=# SELECT (1,2) <=> row(2,3);
	ERROR: could not determine interpretation of row comparison operator <=>
	LINE 1: SELECT (1,2) <=> row(2,3);
	^
	HINT: unsupported operator.
	m_db=# SELECT (1,2) < NULL;
	?column?
	---------- 	
	
	(1 row)
	m_db=# SELECT (1,2) <> NULL;
	?column?
	---------- 	
	(1 row)
	m_db=# SELECT (1, 2) IS NULL;
	?column?
	---------- 	
	f
	(1 row)
	m_db=# SELECT ROW(0,0) BETWEEN ROW(1,1) AND ROW(2,2);
	ERROR: Unsupport type.
	MySQL:
	mysql> SELECT (1,2) <=> row(2,3);
	+--------------------+
	| (1,2) <=> row(2,3) |
	+--------------------+
	| 0 |
	+--------------------+
	1 row in set (0.00 sec)
	
	mysql> SELECT (1,2) < NULL;
	ERROR 1241 (21000): Operand should contain 2 column(s)
	mysql> SELECT (1,2) <> NULL;
	ERROR 1241 (21000): Operand should contain 2 column(s)
	mysql> SELECT (1, 2) IS NULL;
	ERROR 1241 (21000): Operand should contain 1 column(s)
	mysql> SELECT ISNULL((1, 2));
	ERROR 1241 (21000): Operand should contain 1 column(s)
	mysql> SELECT NULL BETWEEN NULL AND ROW(2,2);
	ERROR 1241 (21000): Operand should contain 1 column(s)

--	MySQL表示对一个操作数进行两次取反，结果等于原操作数；GaussDB表示注释。

版权所有 © 华为云计算技术有限公司

操作符	与MySQL的差异
!!	MySQL：!!含义同!，表示取非。 GaussDB：!表示取非操作，当!与!中间存在空格时，表示连续两次取非（! !）；当!与!中间没有空格时，表示阶乘（!!）。 **说明** • GaussDB中，当同时使用阶乘（!!）和取非（!）时，阶乘（!!）和取非（!）中间需要添加空格，否则会报错。 • GaussDB中，当需要多次取非操作时，!与!之间需使用空格隔开。
[NOT] REGEXP	• GaussDB和MySQL在正则表达式中支持的元字符有所不同。例如，GaussDB支持“\d”表示数字，“\w”表示字母、数字和下划线，“\s”表示空格，而MySQL不支持这些元字符，MySQL会把这些字符当成正常字符串。 • GaussDB中“\b”可以与“\\b”匹配，MySQL会匹配失败。 • GaussDB中使用“\”表示转义字符，而MySQL中使用“\\”。 • MySQL不支持2个操作符连在一起使用。 • 模式字符串pattern非法入参，只存在右单括号“)”时，GaussDB会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 • 在de|abc匹配序列de或abc的匹配规则，当|左右存在空值时，GaussDB不会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 • 制表符“\t”正则匹配字符类[:blank:]，GaussDB可匹配，MySQL 5.7不能匹配，MySQL 8.0可匹配。 • GaussDB支持非贪婪模式匹配，即尽可能少的匹配字符，在部分特殊字符后加“?”问号字符，例如：“??, *?, +?, {n}?, {n,}?, {n,m}?”。MySQL 5.7版本不支持非贪婪模式匹配，并报错：Got error 'repetition-operator operand invalid' from regexp。MySQL 8.0版本已经支持。 • 在BINARY字符集下，TEXT类型、BLOB类型均会转换成BYTEA类型，由于REGEXP操作符不支持BYTEA类型，因此无法匹配。
LIKE	• MySQL：LIKE的左操作数只能是位运算或者算术运算或者由括号组成的表达式，LIKE的右操作数只能是单目运算符（不含NOT）或者括号组成的表达式。 GaussDB：LIKE的左右操作数可以是任意表达式。 • LIKE的右操作数使用JDBC的setFloat接口绑参，且GaussDB和MySQL的JDBC都使用PBE的方式执行时，GaussDB的查询结果与MySQL存在差异，但对比等价场景——将该setFloat绑参替换成表中float类型数据，GaussDB的行为保持一致，而MySQL自身的行为不一致。

版权所有 © 华为云计算技术有限公司

操作符	与MySQL的差异

[NOT] BETWEEN AND	MySQL：[NOT] BETWEEN AND嵌套使用时从右到左结合。[NOT] BETWEEN AND的第1个操作数和第2个操作数只能是位运算或者算术运算或者由括号组成的表达式。
	GaussDB：[NOT] BETWEEN AND嵌套使用时从左到右结合。[NOT] BETWEEN AND的第1个操作数和第2个操作数可以是任意表达式。
	在一些特殊场景下（例如表达式中包含repeat()函数），MySQL的[NOT] BETWEEN AND可能与[NOT]（>= AND <=）的等价操作结果不一致，而GaussDB将会优先保证与[NOT]（>= AND <=)的等价操作结果保持一致，因此会存在差异。
	-- GaussDB BETWEEN AND 与 (>= AND <=) 行为一致
	m_db=# CREATE TABLE test_between(c0 YEAR);
	CREATE TABLE
	m_db=# INSERT INTO test_between(c0) VALUES('1963');
	INSERT 0 1
	m_db=# SELECT (''<>'Lt') BETWEEN '' AND c0 AS ref0 FROM test_between;
	ref0
	------ 	
	f
	(1 row)
	
	m_db=# SELECT (''<>'Lt') >= '' AND (''<>'Lt') <= c0 AS ref0 FROM test_between;
	ref0
	------ 	
	f
	(1 row)
	
	m_db=# DROP TABLE test_between;
	DROP TABLE
	
	-- MYSQL: BETWEEN AND行为与 (>= AND <=) 行为不一致
	mysql> CREATE TABLE test_between(c0 YEAR);
	Query OK, 0 rows affected (0.03 sec)
	
	mysql> INSERT INTO test_between(c0) VALUES('1963');
	Query OK, 1 row affected (0.00 sec)
	
	mysql> SELECT (''<>'Lt') BETWEEN '' AND c0 AS ref0 FROM test_between;
	+------+
	| ref0 |
	+------+
	| 1 |
	+------+
	1 row in set (0.00 sec)
	
	mysql> SELECT (''<>'Lt') >= '' AND (''<>'Lt') <= c0 AS ref0 FROM test_between;
	+------+
	| ref0 |
	+------+
	| 0 |
	+------+
	1 row in set (0.00 sec)
	
	mysql> DROP TABLE test_between;
	Query OK, 0 rows affected (0.01 sec)

IN	MySQL：IN的左操作数只能是位运算或者算术运算或者由括号组成的表达式。 GaussDB：IN的左操作数可以是任意表达式。

版权所有 © 华为云计算技术有限公司

操作符	与MySQL的差异
!	MySQL：!的操作数只能是单目运算符（不含not）或者括号组成的表达式。 GaussDB：!的操作数可以是任意表达式。
#	MySQL支持#注释，GaussDB不支持#注释。
BINARY	GaussDB中支持的表达式和MySQL并不完全一致（包括一些函数、操作符等）。GaussDB独有的表达式“~”、“IS DISTINCT FROM”等，由于BINARY关键字优先级更高，使用BINARY expr会优先将BINARY与“~”、“IS DISTINCT FROM”的左参数合并，导致报错。
取反（-）	连续取反次数超过1次时，GaussDB会将其识别为注释，导致返回结果与MySQL存在差异。

版权所有 © 华为云计算技术有限公司

操作符	与MySQL的差异

XOR、|、&、<、>、<=、>=、=、!=	MySQL执行机制为执行左操作数后，对结果进行判断是否为空，进而决定是否需要执行右操作数。
	GaussDB执行机制是执行左右操作数后，对结果再进行判断是否为空。
	当左操作数结果为空，右操作数执行报错时，MySQL不会报错直接返回，GaussDB会执行报错。
	MySQL行为：
	mysql> SELECT version();
	+-----------+
	| version() |
	+-----------+
	| 5.7.44-debug-log |
	+-----------+
	1 row in set (0.00 sec)
	
	mysql> dROP TABLE IF EXISTS data_type_table;
	Query OK, 0 rows affected (0.02 sec)
	
	mysql> CREATE TABLE data_type_table (
	-> MyBool BOOL,
	-> MyBinary BINARY(10),
	-> MyYear YEAR
	->);
	Query OK, 0 rows affected (0.02 sec)
	
	mysql> INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021');
	Query OK, 1 row affected (0.00 sec)
	
	mysql> SELECT (MyBool % MyBinary) | (MyBool - MyYear) FROM data_type_table;
	+----------+
	| (MyBool % MyBinary) | (MyBool - MyYear) |
	+----------+
	| NULL |
	+----------+
	1 row in set, 2 warnings (0.00 sec)
	GaussDB行为：
	m_db=# DROP TABLE IF EXISTS data_type_table;
	DROP TABLE
	m_db=# CREATE TABLE data_type_table (
	m_db(# MyBool BOOL,
	m_db(# MyBinary BINARY(10),
	m_db(# MyYear YEAR
	m_db(#);
	CREATE TABLE
	m_db=# INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021');
	INSERT 0 1
	m_db=# SELECT (MyBool % MyBinary) | (MyBool - MyYear) FROM data_type_table;
	WARNING: Truncated incorrect double value: '4Vx '
	CONTEXT: referenced column: (MyBool % MyBinary) | (MyBool - MyYear)
	WARNING: division by zero
	CONTEXT: referenced column: (MyBool % MyBinary) | (MyBool - MyYear)
	ERROR: Bigint is out of range.
	CONTEXT: referenced column: (MyBool % MyBinary) | (MyBool - MyYear)

版权所有 © 华为云计算技术有限公司

表 3-151 操作符组合差异

操作符组合示例	MySQL数据库	GaussDB数据库	说明
SELECT 1 LIKE 3 & 1;	不支持	支持	LIKE的右操作数不能是位运算符组成的表达式。
SELECT 1 LIKE 1 +1;	不支持	支持	LIKE的右操作数不能是算术运算符组成的表达式。
SELECT 1 LIKE NOT 0;	不支持	支持	LIKE的右操作数只能是+、-、!等单目操作符或者括号组成的表达式，NOT除外。
SELECT 1 BETWEEN 1 AND 2 BETWEEN 2 AND 3;	从右到左结合	从左到右结合	建议加上括号明确运算的优先级，防止由于运算顺序的差异导致运算结果产生偏差。
SELECT 2 BETWEEN 1=1 AND 3;	不支持	支持	BETWEEN的第2个操作数不能是比较操作符组成的表达式。
SELECT 0 LIKE 0 BETWEEN 1 AND 2;	不支持	支持	BETWEEN的第1个操作数不能是模式匹配操作符组成的表达式。
SELECT 1 IN (1) BETWEEN 0 AND 3;	不支持	支持	BETWEEN的第1个操作数不能是IN操作符组成的表达式。
SELECT 1 IN (1) IN (1);	不支持	支持	第2个IN表达式左操作数不能是IN组成的表达式。
SELECT ! NOT 1;	不支持	支持	!的操作数只能是+、-、!等单目操作符或者括号组成的表达式，NOT除外。

索引差异

- GaussDB当前仅支持UBTree和B-tree索引。
- 针对模糊匹配（LIKE操作符），MySQL创建默认索引可以走索引；GaussDB默认的索引不走索引，需要用户使用以下语法指定opclass，比如指定为text_pattern_ops，LIKE操作符才可以走索引。
  ```
  CREATE INDEX indexname ON tablename(col [opclass]);
  ```
- B-tree/UBTree索引场景保持原生GaussDB原有逻辑，即同一操作符族内的类型比较，支持索引扫描，其余索引类型暂未支持。
- 在使用GaussDB JDBC连接数据库时，GaussDB的YEAR类型在含有绑定参数的PBE场景下无法利用索引。
- 部分函数在操作符表达式中嵌套时，返回值类型会发生隐式转换。如果函数返回值类型转换后与表达式索引的类型不一致，将无法使用索引扫描。
- WHERE子句中，索引字段类型和常量类型操作场景下，GaussDB中索引与MySQL索引支持存在差异，如表3-152所示。例如以下语句GaussDB不支持索引：
  ```
  CREATE TABLE t(_int int);
  CREATE INDEX idx ON t(_int) USING BTREE;
  SELECT * FROM t WHERE _int > 2.0;
  ```


版权所有 © 华为云计算技术有限公司

说明

WHERE子句里索引字段类型和常量类型操作场景中，可以使用cast函数将常数类型显式转换为字段类型，以便实现索引。

```
SELECT * FROM t WHERE _int > cast(2.0 AS signed);
```

表 3-152 索引支持存在差异

索引字段类型	常量类型	GaussDB是否支持	MySQL是否支持
整型	整型	是	是
浮点型	浮点型	是	是
定点型	定点型	是	是
字符串类型	字符串类型	是	是
二进制类型	二进制类型	是	是
带日期的时间类型	带日期的时间类型	是	是
TIME类型	TIME类型	是	是
带日期的时间类型	可转为带日期的时间类型（如20231130等整型）	是	是
带日期的时间类型	TIME类型	是	是
TIME类型	可转为TIME类型的常量（如203008等整型）	是	是
浮点型	整型	是	是
浮点型	定点型	是	是
浮点型	字符串类型	是	是
浮点型	二进制类型	是	是
浮点型	带日期的时间类型	是	是
浮点型	TIME类型	是	是
定点型	整型	是	是
字符串类型	带日期的时间类型	是	否
字符串类型	TIME类型	是	否
二进制类型	字符串类型	是	是

版权所有 © 华为云计算技术有限公司

索引字段类型	常量类型	GaussDB是否支持	MySQL是否支持
二进制类型	带日期的时间类型	是	否
二进制类型	TIME类型	是	否
整型	浮点型	否	是
整型	定点型	否	是
整型	字符串类型	否	是
整型	二进制类型	否	是
整型	带日期的时间类型	否	是
整型	TIME类型	否	是
定点型	浮点型	否	是
定点型	字符串类型	否	是
定点型	二进制类型	否	是
定点型	带日期的时间类型	否	是
定点型	TIME类型	否	是
字符串类型	二进制类型	否	是
带日期的时间类型	整型（不可转为带日期的时间类型）	否	是
带日期的时间类型	浮点型（不可转为带日期的时间类型）	否	是
带日期的时间类型	定点型（不可转为带日期的时间类型）	否	是
TIME类型	整型（不可转为TIME类型）	否	是
TIME类型	字符串类型（不可转为TIME类型）	否	是
TIME类型	二进制类型（不可转为TIME类型）	否	是
TIME类型	带日期的时间类型	否	是

版权所有 © 华为云计算技术有限公司

索引字段类型	常量类型	GaussDB是否支持	MySQL是否支持
YEAR类型	YEAR类型	是	是
YEAR类型	可转为YEAR类型的常量（如2034等整型）	是	是
BIT类型	BIT类型	否	是

表 3-153 是否支持走索引

索引字段类型	常量类型	GaussDB是否走索引	MySQL是否走索引
字符串类型	整型	否	否
字符串类型	浮点型	否	否
字符串类型	定点型	否	否
二进制类型	整型	否	否
二进制类型	浮点型	否	否
二进制类型	定点型	否	否
带日期的时间类型	字符串类型（不可转为带日期的时间类型）	否	否
带日期的时间类型	二进制类型（不可转为带日期的时间类型）	否	否
TIME类型	浮点型（不可转为TIME类型）	否	否
TIME类型	定点型（不可转为TIME类型）	否	否
BIT类型	字符串类型	否	否

3.3.2.4 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集，支持的范围如下。

表 3-154 字符集列表

MySQL数据库	GaussDB数据库
utf8mb4	支持

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库
utf8	支持
gbk	支持
gb18030	支持
binary	支持

说明

- GaussDB数据库将utf8和utf8mb4视为同一个字符集，编码最大长度为4字节。当字符串字符集为utf8，指定其字符序为utf8mb4_bin、utf8mb4_general_ci、utf8mb4_unicode_ci、utf8mb4_0900_ai_ci时（例如SELECT _utf8'a' collate utf8mb4_bin），MySQL会发生报错，GaussDB不报错。当字符串字符集为utf8mb4，指定其字符序为utf8_bin、utf8_general_ci、utf8_unicode_ci时，存在同样差异。
- 词法语法解析按照字节流解析，当多字节字符中包含与'\'、'\''、'\\'等符号一致的编码时，会导致与MySQL行为不一致，建议暂时关闭转义符开关进行规避。
- GaussDB数据库对不属于当前字符集的非法字符未执行严格的编码逻辑校验，严格模式下，使用部分字符函数可能导致此类非法字符成功输入，而MySQL会校验报错。宽松模式下截断后插入，与MYSQL行为一致。严格模式与宽松模式由GUC参数sql_mode的strict_trans_tables配置项决定。
- GaussDB字符集之间的转换逻辑与MySQL存在差异，binary字符集与其他字符集混用时可能存在转码流程导致的行为不一致，建议使用binary字符集时，client_encoding、server_encoding、character_set_connection等字符集相关参数保持一致，且仅输入ASCII范围内的字符。

3.3.2.5 排序规则

GaussDB数据库支持指定模式、表或列的排序规则，支持的范围如下。

表 3-155 排序规则列表

MySQL数据库	GaussDB数据库
binary	支持
gb18030_bin	支持
gb18030_chinese_ci	支持
gbk_bin	支持
gbk_chinese_ci	支持
utf8_bin	支持
utf8_general_ci	支持
utf8_unicode_ci	支持
utf8mb4_0900_ai_ci	支持
utf8mb4_bin	支持

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库
utf8mb4_general_ci	支持
utf8mb4_unicode_ci	支持

说明

排序规则差异说明：

- 当前仅有字符串类型、部分二进制类型支持指定排序规则，其他类型不支持指定排序规则，可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。MySQL中所有类型可以指定字符序，但除字符串、二进制类型其他排序规则无实际意义。
- 当前排序规则（除binary外）仅支持在其对应字符集与库级字符集一致时可以指定，GaussDB数据库中，字符集必须与数据库的字符集一致，且不支持表内多种字符集混合使用。
- utf8mb4字符集下默认字符序为utf8mb4_general_ci，与MySQL 5.7保持一致。

3.3.2.6 事务

GaussDB数据库兼容MySQL的事务，但存在部分差异。本章节介绍GaussDB的M-Compatibility数据库中事务相关的差异。

事务默认隔离级别

M-Compatibility默认隔离级别为READ COMMITTED，MySQL默认隔离级别为REPEATABLE READ。

```
-- 查看当前事务隔离级别。
m_db=# SHOW transaction_isolation;
```

子事务

M-Compatibility中，通过SAVEPOINT用于在当前事务里建立一个新的保存点（子事务），使用ROLLBACK TO SAVEPOINT回滚到一个保存点（子事务），子事务回滚后父事务可以继续运行，子事务的回滚不影响父事务的事务状态。

MySQL不存在创建保存点（子事务）。

嵌套事务

嵌套事务指在事务块中开启新事务。

M-Compatibility中，正常事务块中开启新事务会警告存在一个进行中的事务，忽略开启命令；异常事务块中开启新事务将报错，必须在执行ROLLBACK/COMMIT之后才可以执行，执行ROLLBACK/COMMIT会回滚之前语句。

MySQL中，正常事务块中开启新事务会先把之前事务提交，然后开启新事务；异常事务块中开启新事务会忽略错误，提交之前无错误的语句并开启新事务。

```
-- M-Compatibility正常事务块中，开启新事务会警告并忽略。
m_db=# DROP TABLE IF EXISTS test_t;
m_db=# CREATE TABLE test_t(a int, b int);
m_db=# BEGIN;
m_db=# INSERT INTO test_t values(1, 2);
m_db=# BEGIN;   -- 会警告there is already a transaction in progress。
```


版权所有 © 华为云计算技术有限公司

```
m_db=# SELECT * FROM test_t ORDER BY 1;
m_db=# COMMIT;

-- M-Compatibility异常事务块中，开启新事务会报错，必须ROLLBACK/COMMIT之后才可以执行。
m_db=# BEGIN;
m_db=# ERROR sql; -- 错误语句。
m_db=# BEGIN; -- 报错。
m_db=# COMMIT; -- ROLLBACK/COMMIT之后才可以执行。
```

隐式提交的语句

M-Compatibility使用GaussDB存储，继承GaussDB事务机制，事务中执行DDL、DCL不会自动提交。

MySQL在DDL、DCL、管理类语句，锁相关语句会自动提交。

```
-- M-Compatibility创建表和设置GUC参数可以回滚掉。
m_db=# DROP TABLE IF EXISTS test_table_rollback;
m_db=# BEGIN;
m_db=# CREATE TABLE test_table_rollback(a int, b int);
m_db=# \d test_table_rollback;
m_db=# ROLLBACK;
m_db=# \d test_table_rollback; -- 不存在该表。
```

SET TRANSACTION 差异

M-Compatibility中，SET TRANSACTION同时设置多次隔离级别/事务访问模式时，只有最后一个会生效；多个事务特性支持使用空格和逗号分隔。

MySQL中SET TRANSACTION不允许设置多次隔离级别/事务访问模式；多个事务特性只支持使用逗号分隔。

表 3-156 SET TRANSACTION 差异

语法	功能	差异
SET TRANSACTION	设置事务特性。	M-Compatibility中，SET TRANSACTION在会话级别生效；MySQL中SET TRANSACTION在下一个事务生效。
SET SESSION TRANSACTION	设置会话级事务特性。	-
SET GLOBAL TRANSACTION	设置全局会话级事务特性，该特性适用于后续会话，对当前会话无影响。	M-Compatibility中，GLOBAL是全局会话级别生效，只针对当前数据库实例，其它数据库不影响。 MySQL中，会使所有数据库生效。

```
-- SET TRANSACTION会话级生效。
m_db=# SET TRANSACTION ISOLATION LEVEL READ COMMITTED READ WRITE;
m_db=# SHOW transaction_isolation;
m_db=# SHOW transaction_read_only;
-- M-Compatibility同时设置多次隔离级别/事务访问模式，最后一个生效。
m_db=# SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED, ISOLATION LEVEL
```


版权所有 © 华为云计算技术有限公司

```
REPEATABLE READ, READ WRITE, READ ONLY;
m_db=# SHOW transaction_isolation; -- repeatable read
m_db=# SHOW transaction_read_only; -- on
```

START TRANSACTION 差异

M-Compatibility中，START TRANSACTION开启事务时，同时支持设置隔离级别；同时设置多次隔离级别/事务访问模式时，只有最后一个会生效；当前版本不支持立即开启一致性快照；多个事务特性支持空格和逗号分隔。

MySQL的start transaction 开启事务时，不支持设置隔离级别，不支持设置多次事务访问模式；多个事务特性只支持逗号分隔。

```
-- 开启事务设置隔离级别。
m_db=# START TRANSACTION ISOLATION LEVEL READ COMMITTED;
m_db=# COMMIT;
-- 多次设置访问模式。
m_db=# START TRANSACTION READ ONLY, READ WRITE;
m_db=# COMMIT;
```

事务相关的 GUC 参数

表 3-157 事务相关的 GUC 参数差异

GUC参数	功能	差异
autocommit	设置事务自动提交模式。	-

版权所有 © 华为云计算技术有限公司

GUC参数	功能	差异
transaction_isolation	在M-Compatibility中是设置当前事务的隔离级别。 在MySQL中是设置会话级事务的隔离级别。	• GaussDB中，通过使用SET transaction_isolation = value命令，只能改变当前事务的隔离级别。如果想要改变会话级的隔离级别，可以使用default_transaction_isolation。在MySQL中，通过使用SET命令，可以改变会话级的事务隔离级别。 • 支持范围差异。 MySQL中，支持以下隔离级别设置，对大小写不敏感，对空格敏感： – READ COMMITTED – READ UNCOMMITTED – REPEATABLE READ – SERIALIZABLE GaussDB中，支持以下隔离级别设置，对大小写和空格敏感： – read committed – read uncommitted（可以设置，但实际生效的隔离级别为read committed） – repeatable read – serializable（可以设置，但实际生效的隔离级别为repeatable read） – default（设置和会话中默认隔离级别一样） – 设置m_format_dev_version = 's2'时，支持MySQL的隔离级别设置。 • 在GaussDB中，新事务的transaction_isolation值将被初始化为default_transaction_isolation的值。
tx_isolation	设置事务的隔离级别； tx_isolation和transaction_isolation是同义词。	M-Compatibility中不支持查询，不支持修改。需要时，建议查询transaction_isolation。
default_transaction_isolation	设置事务的隔离级别。	M-Compatibility中通过SET设置会改变会话级事务隔离级别。 MySQL中不支持该系统参数。

版权所有 © 华为云计算技术有限公司

GUC参数	功能	差异
transaction_read_only	设置事务的访问模式。	• 在M-Compatibility中，通过使用SET命令，只能改变当前事务的访问模式。如果想要改变会话级的访问模式，可以使用default_transaction_read_only。 在MySQL中，通过使用SET命令，可以改变会话级的事务隔离级别。 • 在GaussDB中，新事务的transaction_read_only值将被初始化为default_transaction_read_only的值。
tx_read_only	设置事务的访问模式。tx_read_only和transaction_read_only是同义词。	M-Compatibility中不支持查询，不支持修改。需要时，建议查询transaction_read_only。
default_transaction_read_only	设置事务的访问模式。	M-Compatibility中通过SET设置会改变会话级事务访问模式；MySQL中不支持该系统参数。

3.3.2.7 SQL

3.3.2.7.1 SQL 兼容性概述

GaussDB数据库兼容绝大多数MySQL语法，但存在部分差异。本章节介绍GaussDB数据库当前支持的MySQL语法。

- 部分关键字在MySQL可以做标识符但在M-Compatibility不可以或存在限制，如表3-158所示。

表 3-158 受限标识符列表

关键字类型	关键字	约束
保留（可以是类型或函数）	COLLATION、COMPACT	除函数和变量，不可以作为其他数据库标识符。
非保留（不能是类型或函数）	BIT、BOOLEAN、COALESCE、DATE、NATIONAL、NCHAR、NONE、NUMBER、TEXT、TIME、TIMESTAMP、TIMESTAMPDIFF	不可以作为函数或变量的标识符。

版权所有 © 华为云计算技术有限公司

关键字类型	关键字	约束
保留	ANY、ARRAY、BUCKETS、DO、END、LESS、MODIFY、OFFSET、ONLY、RETURNING、SOME、USER	不可以作为任意数据库标识符。

- GaussDB优化器与MySQL的优化器存在差异，由于优化器生成的执行计划的差异，可能导致GaussDB行为与MySQL行为的不一致，不影响业务数据结果。

  例如在下述场景中，GaussDB在计算col1以及使用col1进行WHERE比较时，均会调用cast函数，因此会产生两条WARNING记录；MySQL在计算col1时会调用cast函数，在进行WHERE比较时，直接使用计算好的值直接进行比较，因此只产生一条WARNING记录。

```
-- GaussDB行为：
m_db=# SELECT * FROM (SELECT cast('abc' AS decimal) AS col1) t1 WHERE col1=0;
WARNING:  The DECIMAL value 'abc' is incorrect.
WARNING:  The DECIMAL value 'abc' is incorrect.
CONTEXT:  referenced column: col1
 col1
------
    0
(1 row)

m_db=# EXPLAIN VERBOSE SELECT * FROM (SELECT cast('abc' AS decimal) AS col1) t1 WHERE col1=0;
WARNING:  The DECIMAL value 'abc' is incorrect.
WARNING:  The DECIMAL value 'abc' is incorrect.
CONTEXT:  referenced column: col1
                QUERY PLAN
-------------------------------------------
 Result  (cost=0.00..0.01 rows=1 width=0)
   Output: 0::decimal
(2 rows)

-- MySQL行为：
mysql>  SELECT * FROM (SELECT cast('abc' AS decimal) AS col1) t1 WHERE col1=0;
+------+
| col1 |
+------+
|    0 |
+------+
1 row in set, 1 warning (0.00 sec)

mysql> SHOW warnings;
+---------+------+-----------------------------------------+
| Level   | Code | Message                                 |
+---------+------+-----------------------------------------+
| Warning | 1292 | Truncated incorrect DECIMAL value: 'abc' |
+---------+------+-----------------------------------------+
1 row in set (0.00 sec)

mysql> EXPLAIN SELECT * FROM (SELECT cast('abc' AS decimal) AS col1) t1 WHERE col1=0;
+----+-------------+------------+------------+--------+---------------+------+---------+------+------+----------
+----------------+
| id | select_type | table      | partitions | type   | possible_keys | key  | key_len | ref  | rows | filtered |
Extra          |
+----+-------------+------------+------------+--------+---------------+------+---------+------+------+----------
+----------------+
|  1 | PRIMARY     | <derived2> | NULL       | system | NULL          | NULL | NULL    | NULL |    1 |
100.00 | NULL           |
|  2 | DERIVED     | NULL       | NULL       | NULL   | NULL          | NULL | NULL    | NULL | NULL |
NULL | No tables used |
```


版权所有 © 华为云计算技术有限公司

```
+----+-------------+------------+------------+--------+---------------+------+---------+------+------+----------
+----------------+
2 rows in set, 2 warnings (0.01 sec)

mysql> SHOW warnings;
+---------+------+-----------------------------------------------------------------+
| Level   | Code | Message                                                         |
+---------+------+-----------------------------------------------------------------+
| Warning | 1292 | Truncated incorrect DECIMAL value: 'abc'                        |
| Note    | 1003 | /* select#1 */ select '0' AS `col1` from dual where ('0' = 0) |
+---------+------+-----------------------------------------------------------------+
2 rows in set (0.00 sec)
```

3.3.2.7.2 关键字

约束差异：

- 当关键字在M-Compatibility模式下为保留关键字，在MySQL中为非保留关键字，其差异为：在M-Compatibility模式下不可作为表名、列名、列别名、AS列别名、AS表别名、表别名、函数名和变量名，在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字，在MySQL中为保留关键字，其差异为：在M-Compatibility模式下可作为表名、列名、列别名、AS列别名、AS表别名、表别名、函数名和变量名，在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字（可以是函数或类型），在MySQL中为保留关键字，其差异为：在M-Compatibility模式下可作为列别名、AS列别名、函数名和变量名，在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字（可以是函数或类型），在MySQL中为非保留关键字，其差异为：在M-Compatibility模式下不可作为表名、列名、AS表别名和表别名，在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字（不能是函数或类型），在MySQL中为保留关键字，其差异为：在M-Compatibility模式下可作为表名、列名、列别名、AS列别名、AS表别名、表别名和变量名，在MySQL中不支持。
- 当关键字在M-Compatibility模式下为非保留关键字（不能是函数或类型），在MySQL中为非保留关键字，其差异为：在M-Compatibility模式下不可作为函数名，在MySQL中支持。

  说明

  在M-Compatibility模式下的非保留关键字、保留关键字（可以是函数或类型）以及非保留关键字（不能是函数或类型）之中，以下关键字不能作为列别名进行使用：

  BETWEEN、BIGINT、BLOB、CHAR、CHARACTER、COMPILE、CROSS、DEC、DECIMAL、DIV、DOUBLE、EXISTS、FLOAT、FLOAT4、FLOAT8、GROUPING、INNER、INOUT、INT、INT1、INT2、INT3、INT4、INT8、INTEGER、JOIN、LEFT、LIKE、LONGBLOB、LONGTEXT、MEDIUMBLOB、MEDIUMINT、MEDIUMTEXT、MOD、NATURAL、NUMERIC、OUT、OUTER、PRECISION、REAL、RIGHT、ROW、ROW_NUMBER、SIGNED、SMALLINT、SOUNDS、TINYBLOB、TINYINT、TINYTEXT、VALUES、VARCHAR、VARYING、WITHOUT

  其中，SIGNED和WITHOUT在MySQL中可以作为列别名进行使用。

3.3.2.7.3 标识符

M-Compatibility模式下标识符存在以下差异：

- GaussDB无引号标识符中不支持以美元符号（$）开头，MySQL无引号标识符中支持。
- GaussDB无引号标识符中的支持大小写敏感的数据库对象。

版权所有 © 华为云计算技术有限公司

- GaussDB标识符支持U+0080~U+00FF扩展字符，MySQL标识符支持U+0080~U+FFFF的扩展字符。
- 无引号标识符中，GaussDB不支持创建以数字开头包含一个e或E结尾作为标识符的表，例如：

```
-- GaussDB报错不支持，MySQL支持
m_db=# CREATE TABLE 23e(c1 int);
ERROR:  syntax error at or near "23"
LINE 1: CREATE TABLE 23e(c1 int);
                     ^
m_db=# CREATE TABLE t1(23E int);
ERROR:  syntax error at or near "23"
LINE 1: CREATE TABLE t1(23E int);
                        ^
```

- 有引号标识符中，GaussDB对于创建了列名为纯数字或科学计算法的表，不支持直接使用，需要在引号中使用；对于点操作符（.）场景，列名为纯数字或科学计算法的表也需要在引号中使用。例如：

```
-- 创建列名为纯数字或科学计算法的表
m_db=# CREATE TABLE t1(`123` int, `1e3` int, `1e` int);
CREATE TABLE

-- 向表中插入数据
m_db=# INSERT INTO t1 VALUES(7, 8, 9);
INSERT 0 1

-- 结果非预期，但与MySQL结果一致
m_db=# SELECT 123 FROM t1;
 ?column?
----------
      123
(1 row)

-- 结果非预期，但与MySQL结果一致
m_db=# SELECT 1e3 FROM t1;
 ?column?
----------
     1000
(1 row)

-- 结果非预期，并且与MySQL结果不一致
m_db=# SELECT 1e FROM t1;
 e
---
 1
(1 row)

-- 正确用法
m_db=# SELECT `123` FROM t1;
 123
-----
   7
(1 row)

m_db=# SELECT `1e3` FROM t1;
 1e3
-----
   8
(1 row)

m_db=# SELECT `1e` FROM t1;
 1e
----
  9
(1 row)

-- 点操作符的场景，GaussDB不支持，MySQL支持
m_db=# SELECT t1.123 FROM t1;
ERROR:  syntax error at or near ".123"
```


版权所有 © 华为云计算技术有限公司

```
LINE 1: SELECT t1.123 FROM t1;
                 ^
m_db=# SELECT t1.1e3 FROM t1;
ERROR:  syntax error at or near "1e3"
LINE 1: SELECT t1.1e3 FROM t1;
                  ^
m_db=# SELECT t1.1e FROM t1;
ERROR:  syntax error at or near "1"
LINE 1: SELECT t1.1e FROM t1;
                  ^
-- 点操作符的场景，正确用法：
m_db=# SELECT t1.`123` FROM t1;
 123
-----
   7
(1 row)

m_db=# SELECT t1.`1e3` FROM t1;
 1e3
-----
   8
(1 row)

m_db=# SELECT t1.`1e` FROM t1;
 1e
----
  9
(1 row)

m_db=# DROP TABLE t1;
DROP TABLE
```

- GaussDB分区名使用双引号（需要设置SQL_MODE包含ANSI_QUOTES）或反引号时区分大小写，MySQL不区分。
- MySQL标识符长度限制为64字符，而GaussDB标识符长度限制为63字节。超过标识符的长度限制后，MySQL报错，GaussDB会对标识符截断并告警。

3.3.2.7.4 DDL

表 3-159 DDL 语法兼容介绍

概述	详细语法说明	差异
建表和修改表时支持创建主键、UNIQUE索引、外键约束	ALTER TABLE、CREATE TABLE	• 在GaussDB中，当约束关联的表为ustore，且SQL语句中指定为using btree时，底层会建立为ubtree。 • 在GaussDB中，允许将外键作为分区键。 • 索引名、约束名、key名GaussDB是SCHEMA下唯一，MySQL是表下唯一。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持自增列	ALTER TABLE、CREATE TABLE	• GaussDB的自增列建议为索引的第一个字段，否则建表时产生警告，MySQL自动增长列必须为索引第一个字段，否则建表时会报错。GaussDB含有自动增长列的表进行某些操作时会产生错误，例如：ALTER TABLE EXCHANGE PARTITION。 • GaussDB的AUTO_INCREMENT [=] value语法，value必须为[1，2^{127}-1]的正整数。MySQL可以设置为0（实际为1），GaussDB不可为0。 • GaussDB中当自增值已经达到字段数据类型的最大值时，若sql_mode为严格模式，继续自增将产生错误，若sql_mode为宽松模式，仍可自增为最大值。GaussDB自增计数器和表字段值转换使用类型转换规则与表达式计算相同，对于最大值溢出的处理逻辑是合理的。MySQL自增列为BIGINT UNSIGNED时，达到最大值后自增会产生报错，部分场景在达到最大值-2时就会产生报错，但并无统一的标准说明。MySQL自增列为BIGINT以及更小的整型类型时，达到最大值后自增仍为最大值。 • 不支持innodb_autoinc_lock_mode系统变量，GaussDB的GUC参数auto_increment_cache=0时，批量插入自增列的行为与MySQL系统变量innodb_autoinc_lock_mode=1相似。 • GaussDB的自增列在导入数据或者进行Batch Insert执行计划的插入操作时，对于混合0、NULL和确定值的场景，如果产生错误，后续插入自增值不一定与MySQL

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		完全一致。提供auto_increment_cache参数，可以控制预留自增值的数量。 ● GaussDB的并行导入或插入自增列触发自增时，每个并行线程预留的缓存值也只在其线程中使用，未完全使用完毕的话，也会出现表中自增列的值不连续的情况。并行插入产生的自增值结果无法保证与MySQL完全一致。 ● GaussDB不允许auto_increment_offset的值大于auto_increment_increment的值，会产生错误。MySQL允许，并说明auto_increment_offset会被忽略。 ● 在表有主键或索引的情况下，ALTER TABLE命令重写表数据的顺序与MySQL不一定相同，GaussDB按表数据存储顺序重写，MySQL会按主键或索引顺序重写，导致自增值的顺序可能不同。 ● GaussDB使用ALTER TABLE命令添加或修改自增列时，第一次预留自增值的数量是表统计信息中的行数，统计信息的行数不一定与MySQL一致，表现为自增值不连续，产生的自增值结果无法保证与MySQL完全一致。 ● GaussDB对GUC参数auto_increment_offset和auto_increment_increment设置超出范围的值会产生错误。MySQL会自动修改为边界值。 ● 临时表不支持自增列。 ● 本地自增列在各个DN独立自增，不支持在插入成功时

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		刷新LAST_INSERT_ID函数和GeneratedKey的值。 ● sql_mode设置no_auto_value_on_zero参数，表定义的自增列为非NOT NULL约束，向表中插入数据不指定自增列的值时，GaussDB中自增列不触发自增，插入NULL值；MySQL中自增列触发自增，插入NULL值。 ● 批量插入在不同执行计划下，本地自增列自增的顺序、自增值预留数量可能与MySQL存在差异。如：“INSERT INTO table VALUES(...),(...),...”由于需要分布到不同的DN，在某些执行计划中，DN获取不到即将插入的行数。 ● GaussDB的全局自增列在插入数据时自增时机和更新自增起始值时机比MySQL靠前，在插入报错的场景中更容易产生空洞值。如：在INSERT过程中，违反CHECK约束，MySQL 8.0数据库中的自增列不自增，GaussDB分布式全局自增列自增。 ● GaussDB的ALTER TABLE命令新增全局自增列时，各个DN向GTM申请全局自增值，会导致自增值的顺序与MySQL不一致，空洞值也会有差异。 ● UPDATE语句更新自增列的值，如果新值非0、NULL或DEFAULT，且大于当前自增最大值，会更新自增起始值，其行为与MySQL8.0相同。 ● INSERT INTO ON DUPLICATE KEY语句，单次插入多行数据时，如果违反唯一约束，产生的空洞可能会和MySQL存在差异。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		• GaussDB数据库TRUNCATE表时，不支持重置全局自增列的自增器从自增起始值自增，可以使用AUTOINC_RESETVAL()函数重置全局自增列的自增器从自增起始值开始自增。该函数谨慎使用，如果表中数据未清空使用该函数会导致自增列产生重复值。 • INSERT INTO ON DUPLICATE KEY语句当主键冲突更新全局自增列的值时，对于JDBC的getGeneratedKeys() 接口，GaussDB返回自增列的自增值，MySQL返回自增列的更新值。 • GaussDB的全局自增列在插入数据时，自增时机和更新自增起始值时机比MySQL靠前。如果在插入报错时，GaussDB可能已经刷新了last_insert_id的值，下次查询last_insert_id()函数结果，可能会是上次报错语句产生的自增值，和MySQL存在差异。
支持前缀索引	CREATE INDEX、ALTER TABLE、CREATE TABLE	• GaussDB中前缀长度不得超过2676，键值的实际长度受内部页面限制，若字段中含有多字节字符或者一个索引上有多个键，索引行长度可能会超限报错。 • GaussDB中主键索引中不支持前缀键，创建或添加主键时不支持指定前缀长度。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持指定字符集与排序规则	ALTER SCHEMA、ALTER TABLE、CREATE SCHEMA、CREATE TABLE	• 指定库级字符集时，除BINARY字符集外，暂不支持创建新库/模式的字符集与数据库的server_encoding不同。 • 指定表级、列级字符集和字符序时，MySQL支持指定与库级字符集、字符序不同的字符集和字符序。在GaussDB中，表级、列级字符集和字符序仅支持BINARY字符集、字符序或者与库级字符集、字符序相同的字符集、字符序。
修改表时支持在表第一列前面或者在指定列后面添加列	ALTER TABLE	-
修改列名称/定义语法	ALTER TABLE	暂不支持DROP INDEX | DROP KEY | ORDER BY子项。
创建分区表语法	CREATE TABLE PARTITION	• MySQL在以下场景支持表达式，不支持多个分区键： – 使用LIST分区/RANGE分区策略，不指定COLUMNS关键字。 – 使用HASH分区策略。 • MySQL在以下场景不支持表达式，支持多个分区键： – 使用LIST分区/RANGE分区策略，指定COLUMNS关键字。 – 使用KEY分区策略。 • GaussDB不支持使用表达式作为分区键，不支持指定二级分区。 • GaussDB仅在以下场景支持使用多个分区键：使用LIST分区/RANGE分区策略。 • GaussDB分区表不支持用生成列作为分区键。
建表和修改表时支持指定表级和列级comment	CREATE TABLE、ALTER TABLE	-
创建索引时支持指定索引级comment	CREATE INDEX	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
交换普通表和分区表分区的数据	ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点： ● 对于自增列，MySQL执行alter exchange partition后，自增列会被重置；GaussDB则不会被重置，自增列则按照旧的自增值递增。 ● MySQL表或分区使用tablespace时，则无法进行分区和普通表数据的交换；GaussDB表或分区使用不同的tablespace时，仍可进行分区和普通表数据的交换。 ● 对于列默认值，MySQL不会校验默认值，因此默认值不同时也可进行分区和普通表数据的交换；GaussDB会校验默认值，如果默认值不同，则无法进行分区和普通表数据的交换。 ● MySQL在分区表或普通表上进行DROP列操作后，表结构仍然一致，则可进行分区和普通表数据的交换；GaussDB需要保证普通表和分区表的被删除列严格对齐才能进行分区和普通表数据的交换。 ● MySQL和GaussDB的哈希算法不同，所以两者在相同的hash分区存储的数据可能不一致，导致最后交换的数据也可能不一致。 ● MySQL的分区表不支持外键，普通表包含外键或其他表引用普通表的外键，则无法进行分区和普通表数据的交换；GaussDB的分区表支持外键，在两个表的外键约束一致时，则可进行分区和普通表数据的交换，GaussDB的分区表不带外键，普通表有其他表引用，如果分区表和普通表表一致，则可进行分区和普通表数据的交换。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
修改分区表的分区键信息	ALTER TABLE	MySQL支持修改分区表的分区键信息，GaussDB中不支持。
支持CREATE TABLE ... LIKE语法	CREATE TABLE ... LIKE	• 在MySQL 8.0.16 之前的版本中，CHECK约束会被语法解析但功能会被忽略，表现为不复制CHECK约束，GaussDB支持复制CHECK约束。 • 对于主键约束名称，在建表时，MySQL所有主键约束名称固定为PRIMARY KEY，GaussDB不支持复制。 • 对于唯一键约束名称，在建表时，MySQL支持复制，GaussDB不支持复制。 • 对于CHECK约束名称，在建表时，MySQL 8.0.16 之前的版本无CHECK约束信息，GaussDB支持复制。 • 对于索引名称，在建表时，MySQL支持复制，GaussDB不支持复制。 • 在跨sql_mode模式建表时，MySQL受宽松模式和严格模式控制，GaussDB可能存在严格模式失效的情况。 例如：源表存在默认值“0000-00-00”，在“no_zero_date”严格模式下，GaussDB建表成功，且包含默认值“0000-00-00”，严格模式失效；而MySQL建表失败，受严格模式控制。
TRUNCATE子分区语法	ALTER TABLE [IF EXISTS] table_name truncate_clause;	支持子项有差异，对于truncate_clause： • M-Compatibility模式： TRUNCATE PARTITION { { ALL | partition_name [, ...] } | FOR (partition_value [, ...]) } [UPDATE GLOBAL INDEX] • MySQL支持： TRUNCATE PARTITION {partition_names | ALL}

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
主键索引名	CREATE TABLE table_name (col_definition ,PRIMARY KEY [index_name] [USING method] ({ column_name | (expression) }[ASC | DESC] } [, ...]) index_parameters [USING method| COMMENT 'string'])	GaussDB中的主键指定索引名后创建的索引名为用户所指定的索引名，MySQL索引名为PRIMARY。
删除有依赖的对象	DROP drop_type name CASCADE;	在GaussDB中，删除有依赖的对象需要加CASCADE，MySQL不需要。
NOT NULL约束不允许插入NULL值	CREATE TABLE t1(id int NOT NULL DEFAULT 8); INSERT INTO t1 VALUES(NULL); INSERT INTO t1 VALUES(1),(NULL),(2);	在MySQL宽松模式下，会将NULL进行类型转换，并成功插入数据；在MySQL严格模式下不允许插入NULL值。在GaussDB不支持此特性，在宽松模式和严格模式下均不允许插入NULL值。
CHECK约束生效	CREATE TABLE	• CREATE TABLE带CHECK约束的时候，MySQL 8.0会生效，MySQL 5.7只解析语法但不生效。GaussDB在此功能上同步MySQL 8.0版本，且GaussDB CHECK约束可以引用其他列，而MySQL不能。 • GaussDB一个表中最多只能加32767个CHECK约束。
索引的algorithm和lock选项不起作用	CREATE INDEX ... DROP INDEX ...	M-Compatibility模式的CREATE/DROP INDEX语句中INDEX选项algorithm_option和lock_option目前只在语法上支持，创建时不报错，但实际不起作用。
CREATE TABLE hash分区和二级分区的存储与MySQL不同	CREATE TABLE	GaussDB的CREATE TABLE语句中hash分区表和二级分区表所使用的hash函数与MySQL不一致，因此hash分区表和二级分区表的存储与MySQL有区别。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
分区表索引	CREATE INDEX	• GaussDB的分区表索引分为LOCAL和GLOBAL两种。LOCAL索引与某个具体分区绑定，而GLOBAL索引则对应整个分区表。 • LOCAL和GLOBAL索引的创建方法和默认规则具体说明参见**CREATE INDEX**，例如：在非分区键上创建唯一索引，会默认创建为GLOBAL索引。 • MySQL无GLOBAL索引的概念。在GaussDB中，当分区表索引为GLOBAL索引时，对表分区进行DROP、TRUNCATE、EXCHANGE等操作不会默认更新GLOBAL索引，进而导致GLOBAL索引失效，导致后续语句无法选中该索引。为了避免这种场景，建议用户在使用分区操作语法时在最后显指定UPDATE GLOBAL INDEX子句，或配置全局GUC参数enable_gpi_auto_update为true（推荐），使得在进行分区操作时自定更新GLOBAL索引。也可以配置全局GUC参数enable_drop_truncate_partition_keep_gpi为true，使得在进行删除、清空分区操作时自动更新GLOBAL索引。
CREATE/ALTER TABLE语句中分区表为KEY分区，不支持指定algorithm。部分分区定义入参不支持表达式。	CREATE TABLE、ALTER TABLE	GaussDB的CREATE/ALTER TABLE语句中分区表为KEY分区，不支持指定algorithm。 不支持表达式入参的语法： • PARTITION BY HASH() • PARTITION BY KEY() • VALUES LESS THAN()
分区表不支持LINEAR/KEY hash	CREATE TABLE ... PARTITION ...	GaussDB分区表不支持LINEAR/KEY hash。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
check和auto_increment语法不能作用在同一字段	CREATE TABLE	由于MySQL 5.7的check字段不生效，check和auto_increment同时作用于同一字段只有auto_increment生效，但GaussDB报错。
删除存在依赖关系的表	DROP TABLE	GaussDB删除存在依赖的表必须加上CASCADE才能成功，MySQL不需要。
表定义相关选项	CREATE TABLE... 、ALTER TABLE ...	• GaussDB不支持以下选项：AVG_ROW_LENGTH、CHECKSUM、COMPRESSION、CONNECTION、DATA DIRECTORY、INDEX DIRECTORY、DELAY_KEY_WRITE、ENCRYPTION、INSERT_METHOD、KEY_BLOCK_SIZE、MAX_ROWS、MIN_ROWS、PACK_KEYS、PASSWORD、STATS_AUTO_RECALC、STATS_PERSISTENT、STATS_SAMPLE_PAGES。 • 以下选项在GaussDB中不报错，但实际上也不生效：ENGINE、ROW_FORMAT。
generated always as语句不能再引用由generated always as生成的列。	Generated Always AS	GaussDB的generated always as语句不能再引用由generated always as生成的列，MySQL可以。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持更改表名语法	ALTER TABLE tbl_name RENAME [TO | AS | =] new_tbl_name;	• GaussDB的ALTER RENAME语法仅支持修改表名称功能操作，不能耦合其它功能操作。 • GaussDB中，仅旧表名字段支持使用schema.table_name格式，且新表名与旧表名将属于同一个Schema。 • GaussDB不支持新旧表跨Schema重命名操作；但如有权限，则可在当前Schema下修改其他Schema下表名称。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持CREATE VIEW AS SELECT语法	CREATE VIEW table_name AS query;	• 针对以下类型，不支持CREATE VIEW view_name AS query语法中query包含计算操作（如函数调用、使用操作符计算）。 – BINARY[(n)] – BOOLEAN/BOOL – VARBINARY(n) – CHAR[(n)] – VARCHAR(n) – TIME[(p)] – DATETIME[(p)] – TIMESTAMP[(p)] – BIT[(n)] – NUMERIC[(p[,s])] – DECIMAL[(p[,s])] – DEC[(p[,s])] – FIXED[(p[,s])] – FLOAT4[(p, s)] – FLOAT8[(p,s)] – FLOAT[(p)] – REAL[(p, s)] – FLOAT[(p, s)] – DOUBLE[(p,s)] – DOUBLE PRECISION[(p,s)] – TEXT – TINYTEXT – MEDIUMTEXT – LONGTEXT – BLOB – TINYBLOB – MEDIUMBLOB – LONGBLOB • 在query为简单查询场景下，M-Compatibility模式针对上述类型的计算操作进行报错提示，例如: m_db=# CREATE TABLE TEST (salary int(10)); CREATE TABLE

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		m_db=# INSERT INTO TEST VALUES(8000);
		INSERT 0 1
		
		m_db=# CREATE VIEW view1 AS SELECT salary/10 as te FROM TEST;
		ERROR: Unsupported type numeric used with expression in CREATE VIEW statement.
		
		m_db=# CREATE VIEW view2 AS SELECT sec_to_time(salary) as te FROM TEST;
		ERROR: Unsupported type time used with expression in CREATE VIEW statement.
		• 在query为复合查询，子查询等非简单查询场景下，M-Compatibility模式针对上述类型的计算操作与MySQL存在差异，M-Compatibility模式下新创建表的数据类型列精度属性不保留。

索引名可重名范围	CREATE TABLE、CREATE INDEX	MySQL中索引名在一个表下唯一，在不同的表下可以有相同的索引名。M-Compatibility模式中的索引名在同一个SCHEMA下唯一，在同一的SCHEMA下不可用相同的索引名。在M-Compatibility模式下，针对会自动创建索引的约束和key，也会有相同的规则。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
视图依赖差异	CREATE VIEW、ALTER TABLE	MySQL中视图存储，只记录目标表的表名、列名、数据库名信息，不记录目标表的唯一标识；GaussDB会将创建视图时的SQL解析，并存储目标表的唯一标识。因此存在如下差异： ● 修改存在视图依赖的列的数据类型，MySQL中对应的视图不感知目标表的修改，因此可以修改成功；GaussDB视图中的列禁止修改数据类型，因此无法修改该列的数据类型。 ● 重命名存在视图依赖的列，MySQL中对应的视图不感知目标表的修改，因此可以修改成功，但是后续无法查询该视图；GaussDB视图中，每个列精确存储了其对应的表和列的唯一标识，因此表中的列名可以修改成功，视图中的列名不被修改，且后续可以查询该视图。
外键差异	CREATE TABLE	● GaussDB外键约束对类型不敏感，如果主表和从表对应的字段数据类型存在隐式类型转换就可以建成。MySQL外键类型敏感。如果两个表对应的列类型不同外键无法建成。 ● MySQL不支持通过MODIFY COLUMN或CHANGE COLUMN方式修改表列外键所在列的数据类型或列名等，GaussDB可以。
索引升降序差异	CREATE INDEX	在MySQL 5.7中，ASC | DESC被解析但是被忽略，默认行为为ASC；在MySQL 8.0及GaussDB中，ASC | DESC被解析且生效。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

指定字段的默认值	CREATE TABLE、ALTER TABLE	• MySQL 5.7指定默认值时，仅支持不带括号形式的默认值。MySQL 8.0与GaussDB支持带括号形式的默认值。
		-- GaussDB
		m_db=# DROP TABLE IF EXISTS t1, t2;
		DROP TABLE
		m_db=# CREATE TABLE t1(a DATETIME DEFAULT NOW());
		CREATE TABLE
		m_db=# CREATE TABLE t2(a DATETIME DEFAULT (NOW()));
		CREATE TABLE
		
		-- MySQL 5.7
		mysql> DROP TABLE IF EXISTS t1, t2;
		Query OK, 0 rows affected (0.04 sec)
		
		mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW());
		Query OK, 0 rows affected (0.04 sec)
		
		mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW()));
		ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(NOW()))' at line 1
		
		-- MySQL 8.0
		mysql> DROP TABLE IF EXISTS t1, t2;
		Query OK, 0 rows affected (0.17 sec)
		
		mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW());
		Query OK, 0 rows affected (0.19 sec)
		
		mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW()));
		Query OK, 0 rows affected (0.20 sec)
		• MySQL给BLOB、TEXT、JSON数据类型指定默认值时必须给默认值添加括号，GaussDB给上述数据类型指定默认值时可以不添加括号。
		• GaussDB指定默认值时不会校验默认值是否溢出；MySQL指定不带括号形式的默认值时会校验是否溢

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		出，指定带括号形式的默认值时不会校验是否溢出。
		● GaussDB支持使用DATE/TIME/TIMESTAMP开头的时间常量给字段指定默认值，MySQL使用DATE/TIME/TIMESTAMP开头的时间常量给字段指定默认值时必须在默认值外添加括号。
		-- GaussDB
		m_db=# DROP TABLE IF EXISTS t1, t2;
		DROP TABLE
		m_db=# CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00');
		CREATE TABLE
		m_db=# CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00'));
		CREATE TABLE
		
		-- MySQL 5.7
		mysql> DROP TABLE IF EXISTS t1, t2;
		Query OK, 0 rows affected (0.02 sec)
		
		mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00');
		ERROR 1067 (42000): Invalid default value for 'a'
		mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00'));
		ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(TIMESTAMP '2000-01-01 00:00:00'))' at line 1
		
		-- MySQL 8.0
		mysql> DROP TABLE IF EXISTS t1, t2;
		Query OK, 0 rows affected (0.14 sec)
		
		mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00');
		ERROR 1067 (42000): Invalid default value for 'a'
		mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00'));
		Query OK, 0 rows affected (0.19 sec)

版权所有 © 华为云计算技术有限公司

3.3.2.7.5 DML

表 3-160 DML 语法兼容介绍

概述	详细语法说明	差异
DELETE支持从指定分区（或子分区）删除数据	DELETE	-
SELECT INTO语法	SELECT	• GaussDB可以使用SELECT INTO根据查询结果创建一个新表，MySQL不支持。 • GaussDB的SELECT INTO语法不支持将多个查询进行集合运算后的结果作为查询结果。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

REPLACE INTO语法	REPLACE	时间类型初始值的差异。例如:
		• MySQL不受严格模式和宽松模式的影响，可向表中插入时间0值，即:
		mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL);
		Query OK, 1 row affected (0.00 sec)
		
		mysql> REPLACE INTO test VALUES(f1, f2, f3);
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM test;
		+---------------------
		+---------------------+------------+
		| f1 | f2 | f3 |
		+---------------------
		+---------------------+------------+
		| 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 |
		+---------------------
		+---------------------+------------+
		1 row in set (0.00 sec)
		• GaussDB在宽松模式下才可以成功插入时间0值，即
		gaussdb=# SET sql_mode = '';
		SET
		gaussdb=# CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL);
		CREATE TABLE
		gaussdb=# REPLACE INTO test VALUES(f1, f2, f3);
		REPLACE 0 1
		gaussdb=# SELECT * FROM test;
		f1 | f2 | f3
		---------------------+---------------------+------------
		0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00
		(1 row)
		在严格模式下，则报错Incorrect Date/Time/Datetime/Timestamp/Year value。

SELECT支持指定多分区查询	SELECT	-
UPDATE支持指定多分区更新	UPDATE	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

LOAD DATA导入数据功能	LOAD DATA	在使用LOAD DATA导入数据功能时，GaussDB与MySQL相比有如下差异：
		● LOAD DATA语法执行结果与MySQL严格模式一致，宽松模式暂未适配。例如当导入源数据包含非法字符时，GaussDB报错，MySQL严格模式下报错，宽松模式下对非法字符进行截断处理后导入。
		-- GaussDB
		m_db=# SET sql_mode='';
		SET
		m_db=# DROP TABLE IF EXISTS t1;
		DROP TABLE
		m_db=# CREATE TABLE t1(a text) CHARSET gbk;
		CREATE TABLE
		m_db=# SELECT unhex('1AFF1F') INTO OUTFILE '/tmp/3.data';
		COPY 1
		m_db=# LOAD DATA INFILE '/tmp/3.data' INTO TABLE t1 CHARACTER SET gbk;
		ERROR: invalid byte sequence for encoding "GBK": 0xff 0x1f
		CONTEXT: COPY t1, line 1.
		m_db=# DROP TABLE t1;
		DROP TABLE
		
		-- MySQL
		mysql> SET sql_mode='';
		Query OK, 0 rows affected (0.00 sec)
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected (0.02 sec)
		mysql> CREATE TABLE t1(a text) CHARSET gbk;
		Query OK, 0 rows affected (0.03 sec)
		mysql> SELECT unhex('1AFF1F') INTO OUTFILE '/tmp/3.data';
		Query OK, 1 row affected (0.00 sec)
		mysql> LOAD DATA INFILE '/tmp/3.data' INTO TABLE t1 CHARACTER SET gbk;
		Query OK, 1 row affected, 1 warning (0.01 sec)
		Records: 1 Deleted: 0 Skipped: 0 Warnings: 1
		mysql> SHOW warnings;
		+---------+------
		+----------
		----------+
		| Level
			Code |
			Message
			|
			+---------+------
			+----------
			----------+
			| Warning
				1366	Incorrect string value: '\xFF\x1F' for column 'a' at row 1 |
					+---------+------

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		+----------+ 1 row in set (0.00 sec) mysql> SELECT hex(a) FROM t1; +--------+ | hex(a) | +--------+ | 1A | +--------+ 1 row in set (0.00 sec) mysql> DROP TABLE t1; Query OK, 0 rows affected (0.02 sec) ● IGNORE与LOCAL参数功能仅为当导入数据与表中数据存在冲突时，忽略当前冲突行数据功能和当文件中字段数小于指定表中列数时自动为其余列填充默认值功能，其余功能暂未适配。 ● 指定LOCAL关键字，且文件路径为相对路径时，文件从二进制目录下搜索；不指定LOCAL关键字，且文件路径为相对路径时，文件从数据目录下搜索。 ● LOAD DATA仅支持从服务端导入文件。 ● [(col_name_or_user_var [, col_name_or_user_var] ...)]指定列参数不支持重复指定列。 ● [FIELDS TERMINATED BY 'string']指定换行符不能与[LINES TERMINATED BY 'string']分隔符相同。 ● 执行LOAD DATA语法写入表中的数据若无法转换为表中数据类型格式时报错。 ● LOAD DATA SET表达式中不支持指定列名计算。 ● LOAD DATA只能用于表，不能用于视图。 ● Windows下的文件与Linux环境下文件默认换行符存在差异，LOAD DATA无法识别此场景会报错，建议用户导入时检查导入文件行尾换行符。 ● 当源数据的字段数比目标表的字段数多时，GaussDB提示告警信息，MySQL在以下任一场

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		景提示告警信息，其他场景报错。
		– sql_mode未设置严格模式。
		– 指定了IGNORE选项。
		– 指定了LOCAL选项且未指定REPLACE选项。
		• 当源数据文件尾以列分隔符结束时，MySQL按分隔符后无新列处理，GaussDB下按分隔符后包含一列空字符串处理。
		-- 1.csv源数据
		1,2,
		
		-- GaussDB
		m_db=# DROP TABLE IF EXISTS t1;
		NOTICE: table "t1" does not exist, skipping
		DROP TABLE
		m_db=# CREATE TABLE t1(a char(30), b char(30), c char(30));
		CREATE TABLE
		m_db=# LOAD DATA infile '/tmp/1.csv' INTO TABLE t1 fields terminated by ',';
		COPY 1
		m_db=# SELECT c IS NULL, length(c) FROM t1;
		?column? | length
		----------+--------
		f | 0
		(1 row)
		
		-- MySQL
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected (0.06 sec)
		
		mysql> CREATE TABLE t1(a char(30), b char(30), c char(30));
		Query OK, 0 rows affected (0.04 sec)
		
		mysql> LOAD DATA infile '/tmp/1.csv' INTO TABLE t1 fields terminated by ',';
		ERROR 1261 (01000): Row 1 doesn't contain data for all columns

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
LIMIT限制差异	DELETE、SELECT、UPDATE	各个语句的limit子项与MySQL的limit项当前存在差异。 GaussDB中limit参数最大值为BIGINT类型限制（超过92233720368547758O7报错）。在MySQL中，limit最大值为unsigned LONGLONG类型限制（超过18446744073709551615报错）。 limit可以设置小数值，实际执行时四舍五入。MySQL不能取小数。 GaussDB的DELETE语句中，不允许limit 0。MySQL在DELETE语句中允许limit 0。
反斜杠(\)用法差异	INSERT	反斜杠(\)的用法在GaussDB和MySQL中都可以由参数控制但当前默认用法不同： MySQL中使用参数NO_BACKSLASH_ESCAPES控制字符串和标识符中的反斜杠（\）被解析为普通字符还是转义字符，默认反斜杠字符（\）作为字符串和标识符中的转义字符。使用“SET sql_mode='NO_BACKSLASH_ESCAPES';”语句可以禁用反斜杠字符（\）作为字符串和标识符中的转义字符。 GaussDB中使用参数standard_conforming_strings控制字符串和标识符中的反斜杠（\）被解析为普通字符还是转义字符。默认值为on，在普通字符串文本中按照SQL标准把反斜杠（\）当普通文本。使用“SET standard_conforming_strings=off;”语句将反斜杠字符（\）作为字符串和标识符中的转义字符。
插入值少于字段数目时，MySQL报错，GaussDB补充空值。	INSERT	GaussDB不指定列的列表时，如果插入值少于字段数目，默认按建表时的字段顺序赋值。字段上有非空约束时报错，没有非空约束时，如果指定了默认值则缺省部分补充默认值，若未指定默认值则补充空。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
ORDER BY中排序的列必须包括在结果集的列中。	SELECT	在GaussDB中，在与GROUP BY子句一起使用的情况下，ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。在与DISTINCT关键字一起使用的情况下，ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。
SELECT结果允许存在重复列名。	SELECT	-
NATURAL JOIN与MySQL有差异。	SELECT	在GaussDB中，NATURAL [[LEFT | RIGHT] OUTER] JOIN允许不指定LEFT | RIGHT，不指定时NATURAL OUTER JOIN为NATURAL JOIN。允许连续使用多次JOIN。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

UPDATE/DELETE支持ORDER BY和LIMIT	UPDATE/DELETE	在ORDER BY子句和LIMIT子句和UPDATE/DELETE一起使用的情况下，若发现将要更新/删除的元组被其他会话并发修改，会取该条会话匹配中已排序得到的元组的最新值重新进行匹配，若依然满足条件再对这条元组进行更新。这个过程中MySQL会重新获取排序结果进行更新，而GaussDB仅会对该条会话已排序得到的涉及并发更新的元组进行更新/删除，可能导致数据不一致。
		● GaussDB的行为：
		m_db=# CREATE TABLE t1(a INT, b INT);
		CREATE TABLE
		m_db=# CREATE TABLE t2(a INT, b INT);
		CREATE TABLE
		m_db=# INSERT INTO t1 VALUES(1,1),(2,2);
		INSERT 0 2
		-- session one。
		m_db=# START TRANSACTION;
		START TRANSACTION
		-- session one先进行更新，未提交事务。
		m_db=# UPDATE t1 SET a=3 WHERE a=1;
		UPDATE 1
		-- session two。
		m_db=# START TRANSACTION;
		START TRANSACTION
		-- session two后进行更新，等待session one事务结束后更新成功。
		m_db=# UPDATE t1 SET b=4 ORDER BY a LIMIT 1;
		UPDATE 1
		-- session one提交事务。
		m_db=# COMMIT;
		COMMIT
		-- session two更新成功后查询数据。
		m_db=# SELECT * FROM t1;
		a | b
		---+---
		3 | 4
		2 | 2
		(2 rows)
		● MySQL的行为：
		-- session one。
		mysql> START TRANSACTION;
		Query OK, 0 rows affected (0.00 sec)
		-- session one先进行更新，未提交事务。
		mysql> UPDATE t1 SET a=3 WHERE a=1;
		Query OK, 1 row affected (0.00 sec)
		Rows matched: 1 Changed: 1
		Warnings: 0
		-- session two。
		mysql> START TRANSACTION;

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		Query OK, 0 rows affected (0.00 sec) mysql> UPDATE t1 SET b=4 ORDER BY a LIMIT 1; -- session two后进行更新，等待session one事务结束后更新成功。 Query OK, 1 row affected (5.12 sec) Rows matched: 1 Changed: 1 Warnings: 0 -- session one提交事务。 mysql> COMMIT; Query OK, 0 rows affected (0.00 sec) -- session two更新成功后查询数据。 mysql> SELECT * FROM t1; +------+------+ | a | b | +------+------+ | 3 | 1 | | 2 | 4 | +------+------+ 2 rows in set (0.00 sec)
外键数据类型是timestamp/datetime时，UPDATE/DELETE外表报错。	UPDATE/DELETE	外键数据类型是timestamp/datetime时，UPDATE/DELETE外表报错，MySQL成功。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

natural join和using兼容。	SELECT	• GaussDB join的顺序严格按照从左往右，MySQL可能会调整顺序。
		• GaussDB和MySQL在natural join与using时均不允许左表或右表参与join的字段出现歧义（一般由左或右临时表中重名字段造成）。因为两者join的顺序有差别，故行为上可能有差别。
		– GaussDB的行为：
		m_regression=# CREATE TABLE t1(a int,b int);
		CREATE TABLE
		m_regression=# CREATE TABLE t2(a int,b int);
		CREATE TABLE
		m_regression=# CREATE TABLE t3(a int,b int);
		CREATE TABLE
		m_regression=# SELECT * FROM t1 JOIN t2;
		a | b | a | b
		---+---+---+---
		(0 rows)
		m_regression=# SELECT * FROM t1 JOIN t2 natural join t3; -- failed, 因为:列a,b在t1 join t2 得到的临时表中存在重复，故natural join存在歧义。
		ERROR: common column name "a" appears more than once in left table
		– MySQL的行为：
		mysql> SELECT * FROM t1 JOIN t2 NATURAL JOIN t3;
		Empty set (0.00 sec)
		mysql> SELECT * FROM (t1 join t2) NATURAL JOIN t3;
		ERROR 1052 (23000): Column 'a' in from clause is ambiguous

with clause兼容MySQL 8.0版本	SELECT 、INSERT、UPDATE、DELETE	-
join兼容	SELECT	GaussDB join不支持使用逗号“,”的连接方式，MySQL支持。 GaussDB不支持use index for join。
SELECT语句中列表达式为函数表达式、算数表达式等情形时，查询结果显示的列名为？column?	SELECT	GaussDB SELECT语句中列表达式为函数表达式、算数表达式等情形时，查询结果显示的列名为？column?，MySQL为对应表达式。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SELECT导出文件（into outfile）	SELECT ... INTO OUTFILE ...	SELECT INTO OUTFILE语法，导出文件中FLOAT、DOUBLE、REAL类型的值显示精度和MySQL存在差异，不影响COPY导入和导入后的值。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

SELECT/UPDATE/INSERT/REPLACE指定模式名、表名	SELECT/UPDATE/INSERT/REPLACE	• SELECT语句指定投影列时，MySQL支持模式名.表别名.列名的三段式用法，GaussDB不支持。
		m_db=# CREATE SCHEMA test;
		CREATE SCHEMA
		m_db=# CREATE TABLE test.t1(a int);
		CREATE TABLE
		m_db=# SELECT test.alias1.a FROM t1 alias1;
		ERROR: invalid reference to FROM-clause entry for table "alias1"
		LINE 1: SELECT test.alias1.a FROM t1 alias1;
		^
		HINT: There is an entry for table "alias1", but it cannot be referenced from this part of the query.
		CONTEXT: referenced column: a
		• UPDATE/REPLACE SET中，MySQL的三段式用法为database.table.column；GaussDB的三段式用法为table.column.field，其中field为指定复合类型中的属性。二者存在差异。
		• INSERT ... SET中，MySQL支持使用column、table.column和database.table.column；GaussDB只支持使用column，不支持使用table.column和database.table.column，二者存在差异。
		• INSERT ... SET中，MySQL支持SET子句中等式右边引用列名以及包含列名的表达式，GaussDB不支持：
		– GaussDB的行为：
		m_db=# CREATE TABLE t2 (a int default 3, b int default 5);
		CREATE TABLE
		
		m_db=# INSERT INTO t2 SET a = b + 1;
		ERROR: Column "b" does not exist.
		LINE 1: INSERT INTO t2 SET a = b + 1;
		^
		HINT: There is a column named "b" in table "t2", but it cannot be referenced from this part of the query.
		
		m_db=# INSERT INTO t2 SET a =

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		b + 1, b = 0;
		ERROR: Column "b" does not exist.
		LINE 1: INSERT INTO t2 SET a = b + 1, b = 0;
		^
		HINT: There is a column named "b" in table "t2", but it cannot be referenced from this part of the query.
		
		m_db=# INSERT INTO t2 SET b = 0, a = b + 1;
		ERROR: Column "b" does not exist.
		LINE 1: INSERT INTO t2 SET b = 0, a = b + 1;
		^
		HINT: There is a column named "b" in table "t2", but it cannot be referenced from this part of the query.
		
		m_db=# INSERT INTO t2 SET a = a + 1;
		ERROR: Column "a" does not exist.
		LINE 1: INSERT INTO t2 SET a = a + 1;
		^
		HINT: There is a column named "a" in table "t2", but it cannot be referenced from this part of the query.
		
		m_db=# DROP TABLE t2;
		DROP TABLE
		– MySQL的行为：
		mysql> CREATE TABLE t2 (a int default 3, b int default 5);
		Query OK, 0 rows affected (0.07 sec)
		
		mysql> INSERT INTO t2 SET a = b + 1;
		Query OK, 1 row affected (0.02 sec)
		
		mysql> SELECT * FROM t2;
		+------+------+
		| a | b |
		+------+------+
		| 6 | 5 |
		+------+------+
		1 row in set (0.00 sec)
		
		mysql> INSERT INTO t2 SET a = b + 1, b = 0;
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM t2;
		+------+------+
		| a | b |

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		+------+------+
		| 6 | 5 |
		| 6 | 0 |
		+------+------+
		2 rows in set (0.00 sec)
		
		mysql> INSERT INTO t2 SET b = 0, a = b + 1;
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM t2;
		+------+------+
		| a | b |
		+------+------+
		| 6 | 5 |
		| 6 | 0 |
		| 1 | 0 |
		+------+------+
		3 rows in set (0.00 sec)
		
		mysql> INSERT INTO t2 SET a = a + 1;
		Query OK, 1 row affected (0.02 sec)
		
		mysql> SELECT * FROM t2;
		+------+------+
		| a | b |
		+------+------+
		| 6 | 5 |
		| 6 | 0 |
		| 1 | 0 |
		| 4 | 5 |
		+------+------+
		4 rows in set (0.00 sec)
		
		mysql> DROP TABLE t2;
		Query OK, 4 rows affected (0.40 sec)

UPDATE SET执行顺序与MySQL存在差异	UPDATE ... SET	MySQL中，UPDATE SET的顺序是从前往后依次UPDATE，前面UPDATE的结果会影响后面的结果，且允许多次设置同一列；GaussDB中为先取出原来的所有相关的数据，再一次性UPDATE，且不允许多次设置同一列，二者存在差异。
IGNORE特性	UPDATE/DELETE/INSERT	MySQL数据库和GaussDB执行过程的差异，因此产生的WARNING条数和WARNING信息可能存在不同。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW COLUMNS语法	SHOW	• 用户权限验证与MySQL存在差异。 – GaussDB中需要拥有指定表所在Schema的USAGE权限，同时还需要拥有指定表的任意表级权限或列级权限，仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 – MySQL中需要拥有指定表的任意表级权限或列级权限，仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 • LIKE和WHERE子句中涉及到字符串比较操作时，排序规则与MySQL存在差异。 – MySQL中使用utf8_general_ci。 – GaussDB中排序规则使用当前客户端的collation_connection。GaussDB中建议用户在WHERE子句中，不要对返回字段以外的列进行选择，否则可能会出现非预期的报错。 -- 预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE `b`='pri'; ERROR: Column "b" does not exist. LINE 1: SHOW FULL COLUMNS FROM t02 WHERE `b`='pri'; ^ -- 非预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE `c`='pri'; ERROR: input of anonymous composite types is not implemented LINE 1: SHOW FULL COLUMNS FROM t02 WHERE `c`='pri'; ^

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW CREATE DATABASE语法	SHOW	用户权限验证与MySQL存在差异。 ● GaussDB中需要拥有指定Schema的USAGE权限。 ● MySQL中需要拥有任意库级权限（除GRANT OPTION和USAGE）、任意表级权限（除GRANT OPTION）或任意列级权限。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW CREATE TABLE 语法	SHOW	• 用户权限验证与MySQL存在差异。 – GaussDB中需要拥有指定表所在Schema的USAGE权限和指定表的任意表级权限。 – MySQL中需要拥有指定表的任意表级权限（除GRANT OPTION）。 • 返回的建表语句与MySQL存在差异。 – GaussDB中索引以CREATE INDEX语句的形式返回。MySQL中表的索引在CREATE TABLE语句中返回。主要因为GaussDB中CREATE INDEX语法支持的可选参数范围与CREATE TABLE语法中创建索引不同，因此某些索引无法在CREATE TABLE语句中创建。 – GaussDB中CREATE TABLE语法的ENGINE和ROW_FORMAT选项仅做了语法适配，实际不生效，因此在返回的建表语句中不予显示。 • 设置兼容性参数m_format_dev_version为's2'后，返回的建表语句才兼容MySQL。兼容的内容包括：列注释位置变更、表注释位置变更、全局临时表ON COMMIT选项位置变更、主键与唯一约束位置变更、主键与唯一约束中的USING INDEX TABLESPACE选项不再显示以及索引注释位置变更。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW CREATE VIEW 语法	SHOW	• 用户权限验证与MySQL存在差异。 – GaussDB中需要拥有指定视图所在Schema的USAGE权限和指定视图的任意表级权限。 – MySQL中需要拥有指定视图的表级SELECT和表级SHOW VIEW权限。 • 返回的视图创建语句与MySQL存在差异。以SELECT * FROM tbl_name形式创建的视图，GaussDB中*不会被展开，而MySQL中会展开。 • 返回结果中的character_set_client和collation_connection字段与MySQL存在差异。 – MySQL中显示视图创建时系统变量character_set_client和collation_connection的会话值 – GaussDB中未记录相关元数据，显示为NULL。
SHOW PROCESSLIST 语法	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.processlist视图内字段内容与大小写保持一致，MySQL中可能存在差异。 • GaussDB中用户只能访问自己的线程信息，拥有SYSADMIN权限的用户可以访问所有用户的线程信息。 • MySQL中用户只能访问自己的线程信息，拥有PROCESS权限的用户可以访问所有用户的线程信息。
SHOW [STORAGE] ENGINES	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.engines视图内字段内容与大小写保持一致，MySQL中可能存在差异。因为MySQL与GaussDB的存储引擎不同，所以该指令查询的结果不同。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW [SESSION] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.session_status视图内字段内容与大小写保持一致，MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。
SHOW [GLOBAL] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.global_status视图内字段内容与大小写保持一致，MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。
SHOW INDEX	SHOW	• 用户权限验证与MySQL存在差异。 – GaussDB中需要拥有指定SCHEMA的USAGE权限和指定表的任意表级权限或者任意列级权限。 – MySQL中需要拥有指定表的任意表级权限（除GRANT OPTION）或者任意列级权限。 • GaussDB中临时表存储于独立的临时Schema中，在使用FROM或IN db_name条件来展示指定临时表索引信息时，须指明db_name为临时表所在的Schema才能展示临时表索引信息，否则会提示不存在该临时表，这一点和MySQL在部分情况下存在差异。
SHOW SESSION VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.session_variables视图内字段内容及大小写保持一致，与MySQL可能存在差异。
SHOW GLOBAL VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.global_variables视图内字段内容及大小写保持一致，与MySQL可能存在差异。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW CHARACTER SET	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.character_sets视图内字段内容及大小写保持一致，与MySQL可能存在差异。
SHOW COLLATION	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.collations视图内字段内容及大小写保持一致，与MySQL可能存在差异。
EXCEPT语法	SELECT	-
SELECT支持STRAIGHT_JOIN语法	SELECT	GaussDB中多表关联JOIN场景下生成的执行计划，与MySQL可能存在差异。
SHOW TABLES	SHOW	• LIKE行为存在差异，具体请参见**操作符**章节的“LIKE”。 • WHERE表达式行为存在差异，具体行为请参见GaussDB数据库的“WHERE表达式”。 • GaussDB中：表和数据库的权限需要分开赋予用户，查询的数据库必须是用户在SHOW SCHEMAS上可以查询到，不能仅仅有表的权限，必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。 • GaussDB中：校验逻辑中优先校验Schema是否存在，后校验当前用户是否对Schema具有权限，与MySQL存在差异。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SHOW TABLE STATUS	SHOW	• GaussDB中：该语法展示数据依赖information_schema下的tables视图。MySQL中tables指定的是表。 • GaussDB中：表和数据库的权限需要分开赋予用户，查询的数据库必须是用户在SHOW SCHEMAS上可以查询到，不能仅仅有表的权限，必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。 • GaussDB中：校验逻辑中优先校验Schema是否存在，后校验当前用户是否对Schema具有权限，与MySQL存在差异。
HAVING语法	SELECT	GaussDB的HAVING必须且只能引用GROUP BY子句中的列或聚合函数中使用的列。MySQL支持对此行为的扩展，并允许HAVING引用列表中的SELECT列和外部子查询中的列。

SELECT后跟行表达式	SELECT	MySQL不支持SELECT后跟行表达式，GaussDB支持SELECT后跟行表达式。
		MySQL的行为：
		`mysql> SELECT row(1,2);`
		`ERROR 1241 (21000): Operand should contain 1 column(s)`
		GaussDB的行为：
		`m_db=# SELECT row(1,2);`
		` row(1,2)`
		`----------`
		` (1,2)`
		`(1 row)`

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

SELECT FOR SHARE/FOR UPDATE/ LOCK IN SHARE MODE	SELECT	• GaussDB不支持FOR SHARE/FOR UPDATE/LOCK IN SHARE MODE子句和UNION/EXCEPT/DISTINCT/ GROUP BY/HAVING子句一起使用，MySQL 5.7部分支持（FOR SHARE/EXCEPT语法不支持），MySQL 8.0均支持。
		• 当加锁子句与LEFT/RIGHT [OUTER] JOIN子句连用时，LEFT JOIN不支持给右表上锁，RIGHT JOIN不支持给左表上锁；MySQL可以给JOIN两侧的表同时上锁。
		• 当加锁子句与ORDER BY子句一起使用时，GaussDB若发现将要锁定的元组被其他会话并发修改，会取该条会话匹配中已排序得到的元组的最新值。这个过程中MySQL会重新获取排序结果。
		• GaussDB的行为：
		gaussdb=# CREATE TABLE t1(a INT,b INT);
		CREATE TABLE
		gaussdb=# INSERT INTO t1 VALUES(1,1),(2,2);
		INSERT 0 2
		-- session one。
		gaussdb=# START TRANSACTION;
		START TRANSACTION
		-- session one先进行更新，未提交事务。
		gaussdb=# UPDATE t1 SET a=3 WHERE a=1;
		UPDATE 1
		-- session two开启事务。
		gaussdb=# START TRANSACTION;
		START TRANSACTION
		-- session two后进行查询，等待session one事务结束后查询成功。
		gaussdb=# SELECT * FROM t1 ORDER BY a FOR UPDATE;
		commit a | b
		---+---
		3 | 1
		2 | 2
		(2 rows)
		-- session one提交事务。
		gaussdb=# COMMIT;
		COMMIT
		• MySQL的行为：
		mysql> CREATE TABLE t1(a INT,b INT);
		Query OK, 0 rows affected (0.00 sec)
		mysql> INSERT INTO t1 VALUES(1,1),(2,2);

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		Query OK, 2 rows affected (0.00 sec)
		Records: 2 Duplicates: 0 Warnings: 0
		-- session one。
		mysql> START TRANSACTION;
		Query OK, 0 rows affected (0.00 sec)
		-- session one先进行更新，未提交事务。
		mysql> UPDATE t1 SET a=3 WHERE a=1;
		Query OK, 1 row affected (0.00 sec)
		Rows matched: 1 Changed: 1 Warnings: 0
		-- session two开启事务。
		mysql> START TRANSACTION;
		Query OK, 0 rows affected (0.00 sec)
		-- session two后进行查询，等待session one事务结束后查询成功。
		mysql> SELECT * FROM t1 ORDER BY a FOR UPDATE;
		+------+------+
		| a | b |
		+------+------+
		| 3 | 1 |
		| 2 | 4 |
		+------+------+
		2 rows in set (0.00 sec)
		-- session one提交事务。
		mysql> COMMIT;
		Query OK, 0 rows affected (0.00 sec)
		• MySQL不支持给同一张表指定多次锁子句；GaussDB支持给同一张表指定多次锁子句，实际生效按照最强的锁处理。
		-- GaussDB
		m_db=# DROP TABLE IF EXISTS t1;
		DROP TABLE
		
		m_db=# CREATE TABLE t1(a INT, b INT);
		CREATE TABLE
		
		m_db=# INSERT INTO t1 VALUES(1,2);
		INSERT 0 1
		
		m_db=# SELECT * FROM t1 FOR UPDATE OF t1 LOCK IN SHARE MODE;
		a | b
		---+---
		1 | 2
		(1 row)
		
		m_db=# DROP TABLE t1;
		DROP TABLE
		
		-- MySQL
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected (0.05 sec)
		
		mysql> CREATE TABLE t1(a INT, b INT);
		Query OK, 0 rows affected (0.09 sec)

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.01 sec) mysql> SELECT * FROM t1 FOR UPDATE OF t1 LOCK IN SHARE MODE; ERROR 3569 (HY000): Table t1 appears in multiple locking clauses. mysql> DROP TABLE t1; Query OK, 0 rows affected (0.05 sec)

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

SELECT语法支持范围	SELECT	• GaussDB的HAVING子句必须与GROUP BY子句或聚合函数一同使用，MySQL支持在查询语句中仅指定HAVING子句。
		• GaussDB的HAVING必须且只能引用GROUP BY子句中的列或聚合函数中使用的列。MySQL支持对此行为的扩展，并允许HAVING引用列表中的SELECT列和外部子查询中的列。
		• GaussDB以及MySQL5.7使用GROUP BY WITH ROLLUP时，不支持与DISTINCT、ORDER BY子句共同使用，MySQL8.0支持。
		• 当查询表为空表时，GaussDB在使用指定WITH ROLLUP语句进行查询时，查询结果为一条空行数据，而MySQL查询结果为空。
		• GaussDB指定FROM子句中的表别名时，支持带字段名称。MySQL 5.7不支持指定表别名时带字段名称，MySQL 8.0仅支持给子查询指定表别名时带字段名称。
		-- GaussDB
		m_db=# DROP TABLE IF EXISTS t1;
		DROP TABLE
		m_db=# CREATE TABLE t1(a INT, b INT);
		CREATE TABLE
		m_db=# INSERT INTO t1 VALUES(1,2);
		INSERT 0 1
		m_db=# SELECT * FROM t1 t2(a, b);
		a | b
		---+---
		1 | 2
		(1 row)
		
		m_db=# SELECT * FROM (SELECT * FROM t1) t2(a, b);
		a | b
		---+---
		1 | 2
		(1 row)
		
		-- MySQL 5.7
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected, 1 warning (0.00 sec)
		
		mysql> CREATE TABLE t1(a INT, b INT);

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		Query OK, 0 rows affected (0.03 sec)
		
		mysql> INSERT INTO t1 VALUES(1,2);
		Query OK, 1 row affected (0.01 sec)
		
		mysql> SELECT * FROM t1 t2(a, b);
		ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1
		mysql> SELECT * FROM (SELECT * FROM t1) t2(a, b);
		ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1
		
		-- MySQL 8.0
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected (0.10 sec)
		
		mysql> CREATE TABLE t1(a INT, b INT);
		Query OK, 0 rows affected (0.18 sec)
		
		mysql> INSERT INTO t1 VALUES(1,2);
		Query OK, 1 row affected (0.03 sec)
		
		mysql> SELECT * FROM t1 t2(a, b);
		ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1
		mysql> SELECT * FROM (SELECT * FROM t1) t2(a, b);
		+------+------+
		| a | b |
		+------+------+
		| 1 | 2 |
		+------+------+
		1 row in set (0.00 sec)
		• 当查询语句不带FROM子句时，GaussDB支持带WHERE子句，与MySQL 8.0保持一致，MySQL 5.7不支持。
		-- GaussDB
		m_db=# SELECT 1 WHERE true;
		 1
		--- 	
		 1
		(1 row)
		
		-- MySQL 5.7
		mysql> SELECT 1 WHERE true;
		ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'where true' at line 1

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		-- MySQL 8.0 mysql> SELECT 1 WHERE true; +---+ | 1 | +---+ | 1 | +---+ 1 row in set (0.00 sec)

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
SELECT投影列赋值用户变量差异	SELECT @var_name := expr	• MySQL用户变量名支持使用转义字符或双重引号转义，GaussDB用户变量名不支持。单引号括起的变量名，变量名不能出现单引号，如@'''、@''''、@'\''不支持，解析时匹配不到'或解析报错，如： -- 解析报错 m_db=# SET @'''' = 1; ERROR: syntax error at or near "@" LINE 1: SET @'''' = 1; -- 解析时匹配不到' m_db=# SET @'\'' = 1; m_db'# 双引号括起的变量名，变量名不能出现双引号，如@"""、@""""""、@"\""不支持，解析时匹配不到"或解析报错，如： -- 解析报错 m_db=# SET @"""""" = 1; ERROR: syntax error at or near "@" LINE 1: SET @"""""" = 1; -- 解析时匹配不到" m_db=# SET @"\"" = 1; m_db"# 反引号括起的变量名，变量名不能出现反引号，如@```、@````、@`\``不支持，解析时匹配不到`或解析报错，如： -- 解析报错 m_db=# SET @```` = 1; ERROR: syntax error at or near "@" LINE 1: SET @```` = 1; -- 解析时匹配不到` m_db=# SET @`\`` = 1; m_db`# • 当SQL语句中同时出现为用户变量赋值和读取用户变量时，变量的求值顺序是未定义的，最终结果与MySQL可能会有差异。以赋值用户变量，同时用户变量作为MAX函数的参数为例： -- GaussDB m_db=# SELECT @a := 2; @a ---- 2 (1 row) m_db=# SELECT MAX(@a), @a := 3; max | @a -----+---- 2 | 3 (1 row)

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		m_db=# SELECT @a;
		 @a
		---- 	
		 3
		(1 row)
		
		-- MySQL 5.7
		mysql> SELECT @a := 2;
		+------+
		| @a |
		+------+
		| 2 |
		+------+
		1 row in set (0.00 sec)
		
		mysql> SELECT max(@a), @a := 3;
		+---------+---------+
		| max(@a) | @a := 3 |
		+---------+---------+
		| 3 | 3 |
		+---------+---------+
		1 row in set (0.00 sec)
		
		mysql> SELECT @a;
		+------+
		| @a |
		+------+
		| 3 |
		+------+
		1 row in set (0.00 sec)
		• 设置用户变量为子查询或CASE WHEN表达式且最终结果为子查询表达式时，GaussDB中自定义变量的值与直接查询的结果保持一致，而MySQL使用的中间计算结果。GaussDB中最终自定义变量的值与MySQL可能会有差异。
		CREATE TABLE t1 (a double(6,3));
		INSERT INTO t1 VALUES(1.1);
		
		-- GaussDB，用户变量的值与直接查询一致
		m_db=# SELECT @a := (SELECT a FROM t1);
		 @a
		------- 	
		 1.100
		(1 row)
		
		m_db=# SELECT @a;
		 @a
		------- 	
		 1.100
		(1 row)
		
		-- MySQL 5.7，用户变量的值与直接查询的不一致
		mysql> SELECT @a := (SELECT a FROM t1);
		+--------------------------+
		| @a := (SELECT a FROM t1) |

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		+----------+
		| 1.100 |
		+----------+
		1 row in set (0.00 sec)
		
		mysql> SELECT @a;
		+------+
		| @a |
		+------+
		| 1.1 |
		+------+
		1 row in set (0.00 sec)
		• ORDER BY 子句中引用用户变量表达式相关的列时，GaussDB用户变量的结果与MySQL可能会有差异，具体表现与执行计划有关。
		CREATE TABLE t1 (a int);
		INSERT INTO t1 VALUES (1),(3),(2);
		
		-- GaussDB
		m_db=# SELECT @a := a FROM t1 ORDER BY a;
		@a
		---- 	
		1
		2
		3
		(3 rows)
		
		m_db=# SELECT @a;
		@a
		---- 	
		2
		(1 row)
		
		-- MySQL 5.7
		mysql> SELECT @a := a FROM t1 ORDER BY a;
		+---------+
		| @a := a |
		+---------+
		| 1 |
		| 2 |
		| 3 |
		+---------+
		3 rows in set (0.00 sec)
		
		mysql> SELECT @a;
		+------+
		| @a |
		+------+
		| 3 |
		+------+
		1 row in set (0.00 sec)
		• @var_name :=expr作为IF/IFNULL/COALESCE函数的入参，当此参数未被选中时，MySQL不会给用户变量赋值，而GaussDB会给用户变量赋

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		值，实现不一致。以IF函数为例：
		-- GaussDB，无论@var:=val参数是否被选中，都会为@var赋值
		m_db=# SELECT IF (1, @a:=1, 2);
		 if
		---- 	
		 1
		(1 row)
		
		m_db=# SELECT @a;
		 @a
		---- 	
		 1
		(1 row)
		
		m_db=# SELECT IF (0, @a:=10, 2);
		 if
		---- 	
		 2
		(1 row)
		
		m_db=# SELECT @a;
		 @a
		---- 	
		 10
		(1 row)
		
		-- MySQL 5.7，@var:=val参数被选中时为@var赋值，@var:=val参数未被选中时不会为@var赋值
		mysql> SELECT IF (1, @a:=1, 2);
		+------------------+
		| if (1, @a:=1, 2) |
		+------------------+
		| 1 |
		+------------------+
		1 row in set, 1 warning (0.00 sec)
		
		mysql> SELECT @a;
		+------+
		| @a |
		+------+
		| 1 |
		+------+
		1 row in set (0.00 sec)
		
		mysql> SELECT IF (0, @a:=10, 2);
		+-------------------+
		| if (0, @a:=10, 2) |
		+-------------------+
		| 2 |
		+-------------------+
		1 row in set, 1 warning (0.00 sec)
		
		mysql> SELECT @a;
		+------+
		| @a |
		+------+
		| 1 |
		+------+
		1 row in set (0.00 sec)
		• 当用户变量赋值操作出现在FROM 子查询中时，GaussDB

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		直接把子查询结果传递到上层访问，而MySQL会把FROM子查询的结果全部物化到临时表中供上层访问查询，两者的实现不一致。
		CREATE TABLE t1(a int);
		INSERT INTO t1 VALUES (1),(2),(3);
		
		-- GaussDB
		m_db=# SELECT @a := a, a FROM t1;
		 @a | a
		----+---
		 1 | 1
		 2 | 2
		 3 | 3
		(3 rows)
		
		m_db=# SELECT @a, a FROM
		(SELECT @a := a, a FROM t1)t;
		 @a | a
		----+---
		 1 | 1
		 2 | 2
		 3 | 3
		(3 rows)
		
		m_db=# SELECT @a;
		 @a
		---- 	
		 3
		(1 row)
		
		-- MySQL 5.7
		mysql> SELECT @a := a, a FROM t1;
		+---------+------+
		| @a := a | a |
		+---------+------+
		| 1 | 1 |
		| 2 | 2 |
		| 3 | 3 |
		+---------+------+
		3 rows in set (0.00 sec)
		
		mysql> SELECT @a, a FROM (SELECT
		@a := a, a FROM t1)t;
		+------+------+
		| @a | a |
		+------+------+
		| 3 | 1 |
		| 3 | 2 |
		| 3 | 3 |
		+------+------+
		3 rows in set (0.00 sec)
		
		mysql> SELECT @a;
		+------+
		| @a |
		+------+
		| 3 |
		+------+
		1 row in set (0.00 sec)

 版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

INSERT ... ON DUPLICATE KEY UPDATE语法	INSERT	• GaussDB的ON DUPLICATE KEY UPDATE子句中的VALUES()不支持表名.列名格式；MySQL支持。
		• GaussDB不允许对约束字段用ON DUPLICATE KEY UPDATE进行修改；MySQL允许。
		• INSERT ... query ON DUPLICATE KEY UPDATE语句中，query为子查询时，GaussDB的ON DUPLICATE KEY UPDATE子句不支持引用子查询中的列名；MySQL支持.
		• ON DUPLICATE KEY UPDATE子句更新多列时，MySQL前面UPDATE的结果会影响后面的结果，且允许同一列设置多次。GaussDB中，前面UPDATE的结果不会影响后面的结果，且不支持同一列设置多次，二者存在差异。
		• 插入数据违反唯一约束执行UPDATE操作时，GaussDB和MySQL返回的受影响行数存在差异。更新一条数据时，GaussDB返回1，MySQL返回2；如果将现有行更新为其当前值，GaussDB返回1，MySQL返回0。
		• 对于表字段为VARCHAR类型，隐式转换为数值数据类型的场景：MySQL在字符串转换失败时，转换失败的部分比字符串完整长度少2的情况下（即长度差等于2），不会产生告警或报错，对于长度差不等于2的情况下，产生告警或报错。GaussDB在字符串转换失败时，会始终产生告警或报错。
		-- GaussDB
		m_db=# CREATE TABLE t1(a int PRIMARY KEY, b varchar(10));
		NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "t1_pkey" for table "t1"
		CREATE TABLE
		
		m_db=# INSERT INTO t1 VALUES(1, 'A');

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		INSERT 0 1
		
		m_db=# INSERT INTO t1 VALUES(1, 'X') ON DUPLICATE KEY UPDATE b=values(a)+values(b);
		ERROR: The double value 'X' is incorrect.
		CONTEXT: referenced column: b
		
		m_db=# INSERT INTO t1 VALUES(1, 'YY') ON DUPLICATE KEY UPDATE b=values(a)+values(b);
		ERROR: The double value 'YY' is incorrect.
		CONTEXT: referenced column: b
		
		m_db=# INSERT INTO t1 VALUES(1, 'ZZZ') ON DUPLICATE KEY UPDATE b=values(a)+values(b);
		ERROR: The double value 'ZZZ' is incorrect.
		CONTEXT: referenced column: b
		
		m_db=# DROP TABLE t1;
		DROP TABLE
		
		-- MySQL
		mysql> CREATE TABLE t1(a INT PRIMARY KEY, b VARCHAR(10));
		Query OK, 0 rows affected (0.00 sec)
		
		mysql> TRUNCATE TABLE t1;
		Query OK, 0 rows affected (0.01 sec)
		
		mysql> INSERT INTO t1 VALUES(1, 'A');
		Query OK, 1 row affected (0.00 sec)
		
		mysql> INSERT INTO t1 VALUES(1, 'X') ON DUPLICATE KEY UPDATE b=values(a)+values(b);
		ERROR 1292 (22007): Truncated incorrect DOUBLE value: 'X'
		
		mysql> INSERT INTO t1 VALUES(1, 'YY') ON DUPLICATE KEY UPDATE b=values(a)+values(b);
		Query OK, 2 rows affected, 2 warnings (0.00 sec)
		
		mysql> INSERT INTO t1 VALUES(1, 'ZZZ') ON DUPLICATE KEY UPDATE b=values(a)+values(b);
		ERROR 1292 (22007): Truncated incorrect DOUBLE value: 'ZZZ'
		
		mysql> DROP TABLE t1;
		Query OK, 0 rows affected (0.00 sec)

 版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
UNION/EXCEPT语句	SELECT	GaussDB集合操作中，当前两个查询中合并的数据是字符串字面量的形式时，如果存在第三个查询，集合操作的返回类型直接以第三个查询的数据类型为准。 -- 使用第三个查询中的类型做转换，第二个查询中的数据转换失败。 m_db=# SELECT '123' AS col UNION SELECT 'test' UNION SELECT 9999 ORDER BY col; WARNING: The integer syntax of the input is invalid: "test". CONTEXT: referenced column: col col ------ 0 123 9999 (3 rows)

3.3.2.7.6 DCL

表 3-161 DCL 语法兼容介绍

概述	详细语法说明	差异
SET NAMES指定COLLATE子句	SET [SESSION | LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] | DEFAULT};	GaussDB中暂不支持指定charset_name与数据库字符集不同。具体请参考《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > S > SET”章节。

USE切换当前模式	USE schema_name	使用USE语句指定模式，且用户没有对应模式的权限时，MySQL产生报错，GaussDB会将当前模式指定为空。
		-- MySQL
		mysql> USE test;
		ERROR 1044 (42000): Access denied for user 'u1'@'%' to database 'test'
		
		-- GaussDB
		m_db=> USE test;
		SET
		m_db=> SELECT database();
		ERROR: function returned NULL
		CONTEXT: referenced column: database

版权所有 © 华为云计算技术有限公司

3.3.2.7.7 其他语句

表 3-162 其他语法兼容介绍

概述	详细语法说明	差异
事务相关语法	数据库默认隔离级别	M-Compatibility默认隔离级别为READ COMMITTED，MySQL默认隔离级别为REPEATABLE READ。 M-Compatibility隔离级别只有READ COMMITTED、REPEATABLE READ生效。
事务相关语法	事务嵌套	M-Compatibility中嵌套事务不会自动提交，MySQL会自动提交。
事务相关语法	自动提交	M-Compatibility使用GaussDB存储，继承GaussDB事务机制，事务中执行DDL，DCL不会自动提交。MySQL在DDL、DCL、管理类语句，锁相关语句会自动提交。
事务相关语法	报错后需rollback	M-Compatibility事务中报错，需要执行rollback，MySQL无限制。
事务相关语法	锁机制	M-Compatibility锁机制只能在事务块中使用，MySQL无限制。
锁机制	锁机制	• MySQL获取read锁后，当前会话无法进行写操作，M-Compatibility获取read锁后，当前会话可以进行写操作。 • MySQL给表上锁后，读取其他表报错，M-Compatibility无限制。 • MySQL同一会话中获取同一个表的锁，会自动释放上一个锁，并提交事务，M-Compatibility无该机制。 • M-Compatibility中LOCK TABLE只能在一个事务块的内部有用，且无UNLOCK TABLE命令，锁总是在事务结束时释放。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
PBE	PBE	• 重复创建同名的PREPARE语句，M-Compatibility会报已经存在的错误，需要先删除已有statement，MySQL会覆盖旧的statement。 • M-Compatibility和MySQL在SQL语句执行过程中对异常场景的报错阶段不同，例如解析层、执行层等；而PREPARE语句对预备语句只处理到解析层。因此PBE下对于异常场景，报错位置在PREPARE阶段还是EXECUTE阶段，M-Compatibility和MySQL存在可能差异。

3.3.2.7.8 用户与权限

概述

在M-Compatibility中，用户与权限管控相关的行为、语法整体沿用GaussDB的机制，暂不同步MySQL。

用户与权限的行为与GaussDB保持一致，具体行为说明请参见《开发指南》中“数据库安全 > 用户及权限”章节。

用户与权限的语法在原有GaussDB的基础上，裁剪了部分语法，具体语法说明请参见《M-Compatibility开发指南》中“ SQL参考 > SQL语法 > SQL语句”章节。M-Compatibility与GaussDB的语法差异请参见**表3-163**。

表 3-163 M-Compatibility 与 GaussDB 的语法差异

语法说明	概述	M-Compatibility与GaussDB的差异
CREATE ROLE	创建一个角色。	在M-Compatibility中，不支持指定涉及以下关键字的选项：ENCRYPTED、UNENCRYPTED、RESOURCE POOL、PERM SPACE、TEMP SPACE、SPILL SPACE。 在M-Compatibility中，创建USER时会自动创建与USER同名的Schema，MySQL不创建。
CREATE USER	创建一个用户。	
CREATE GROUP	创建一个新用户组。CREATE GROUP是CREATE ROLE的别名，不推荐使用。	
ALTER ROLE	修改角色属性。	
ALTER USER	修改用户属性。	
ALTER GROUP	修改一个用户组的属性。	-

版权所有 © 华为云计算技术有限公司

语法说明	概述	M-Compatibility与GaussDB的差异
DROP ROLE	删除角色。	-
DROP USER	删除用户。	-
DROP GROUP	删除用户组。	-
DROP OWNED	删除一个数据库角色所拥有的数据库对象。	-
REASSIGN OWNED	修改数据库对象的属主。	M-Compatibility中不支持该语法。
GRANT	对角色和用户进行授权操作。	M-Compatibility中不支持授予或回收函数、存储过程、表空间、DATABASE LINK等对象的权限。
REVOKE	用于撤销一个或多个角色的权限。	
ALTER DEFAULT PRIVILEGES	设置应用于将来创建的对象的权限（这不会影响分配到已有对象中的权限）。	M-Compatibility中不支持函数，类型，密态数据库客户端主密钥和列加密密钥的权限更改。

差异说明

- 语法格式差异

  M-Compatibility的授权语法请参见《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > G > GRANT”章节，MySQL中的授权语法如下：

```
-- 全局级、数据库级、表级、存储过程级赋权语法
GRANT
    priv_type [(column_list)]
      [, priv_type [(column_list)]] ...
    ON [object_type] priv_level
    TO user [auth_option] [, user [auth_option]] ...
    [REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
    [WITH {GRANT OPTION | resource_option} ...]

-- 用户代理赋权语法
GRANT PROXY ON user
    TO user [, user] ...
    [WITH GRANT OPTION]

object_type: {
    TABLE
  | FUNCTION
  | PROCEDURE
}

priv_level: {
    *
  | *.*
  | db_name.*
  | db_name.tbl_name
  | tbl_name
  | db_name.routine_name
}
```


版权所有 © 华为云计算技术有限公司

```
user:
    'user_name'@'host_name'

auth_option: {
    IDENTIFIED BY 'auth_string'
  | IDENTIFIED WITH auth_plugin
  | IDENTIFIED WITH auth_plugin BY 'auth_string'
  | IDENTIFIED WITH auth_plugin AS 'auth_string'
  | IDENTIFIED BY PASSWORD 'auth_string'
}

tls_option: {
    SSL
  | X509
  | CIPHER 'cipher'
  | ISSUER 'issuer'
  | SUBJECT 'subject'
}

resource_option: {
  | MAX_QUERIES_PER_HOUR count
  | MAX_UPDATES_PER_HOUR count
  | MAX_CONNECTIONS_PER_HOUR count
  | MAX_USER_CONNECTIONS count
}
```

- 赋权类型差异

  MySQL支持的赋权类型如下：

表 3-164 MySQL 支持的赋权类型

权限类型	释义及权限级别
ALL [PRIVILEGES]	授予指定访问级别的所有权限，除了**GRANT OPTION**和**PROXY**。
ALTER	启用**ALTER TABLE**。级别：全局、数据库、表。
ALTER ROUTINE	允许更改或删除存储过程。级别：全局、数据库、例程。
CREATE	启用数据库和表创建。级别：全局、数据库、表。
CREATE ROUTINE	启用存储过程创建。级别：全局、数据库。
CREATE TABLESPACE	允许创建、更改或删除表空间和日志文件组。级别：全局。
CREATE TEMPORARY TABLES	启用**CREATE TEMPORARY TABLE**。级别：全局、数据库。
CREATE USER	启用**CREATE USER**、**DROP USER**、**RENAME USER**和**REVOKE ALL PRIVILEGES**。级别：全局。
CREATE VIEW	允许创建或更改视图。级别：全局、数据库、表。

版权所有 © 华为云计算技术有限公司

权限类型	释义及权限级别
DELETE	启用DELETE。级别：全局、数据库、表。
DROP	允许删除数据库、表和视图。级别：全局、数据库、表。
EVENT	启用定时任务。级别：全局、数据库。
EXECUTE	使用户能够执行存储过程。级别：全局、数据库、存储过程。
FILE	使用户能够使服务器读取或写入文件。级别：全局。
GRANT OPTION	允许向其他账户授予权限或从其他账户删除权限。级别：全局、数据库、表、存储过程、代理。
INDEX	允许创建或删除索引。级别：全局、数据库、表。
INSERT	启用INSERT。级别：全局、数据库、表、列。
LOCK TABLES	在具有SELECT权限的表上启用LOCK TABLES。级别：全局、数据库。
PROCESS	使用户能够通过SHOW PROCESSLIST查看所有正在运行的线程。级别：全局。
PROXY	启用用户代理。级别：从用户到用户。
REFERENCES	启用外键创建。级别：全局、数据库、表、列。
RELOAD	启用FLUSH操作的使用。级别：全局。
REPLICATION CLIENT	使用户能够查询源服务器或副本服务器的位置。级别：全局。
REPLICATION SLAVE	允许副本从源读取二进制日志。级别：全局。
SELECT	启用使用SELECT。级别：全局、数据库、表、列。
SHOW DATABASES	启用SHOW DATABASES以显示所有数据库。级别：全局。
SHOW VIEW	启用SHOW CREATE VIEW。级别：全局、数据库、表。

版权所有 © 华为云计算技术有限公司

权限类型	释义及权限级别
SHUTDOWN	启用mysqladmin shutdown的使用。级别：全局。
SUPER	启用其他管理操作，例如CHANGE MASTER TO、KILL、PURGE BINARY LOGS、SET GLOBAL和mysqladmin debug命令。级别：全局。
TRIGGER	启用触发器操作。级别：全局、数据库、表。
UPDATE	启用UPDATE。级别：全局、数据库、表、列。
USAGE	等价于“没有特权”。

M-Compatibility以级别划分支持以下权限：

表 3-165 M-Compatibility 支持的赋权类型

授权对象	支持授予的权限
数据库	CREATE、CONNECT、TEMPORARY、TEMP、ALTER、DROP、COMMENT
模式	CREATE、USAGE、ALTER、DROP、COMMENT
表、视图	SELECT、INSERT、UPDATE、DELETE、TRUNCATE、REFERENCES、TRIGGER、ALTER、DROP、COMMENT、INDEX、VACUUM
列	SELECT、INSERT、UPDATE、REFERENCES、COMMENT
序列	SELECT、USAGE、UPDATE、ALTER、DROP、COMMENT

- MySQL通过'*.*'表示全局层级的授权对象；GaussDB通过'{DATABASE} db_name'表示数据库层级的授权对象。GaussDB的数据库层级对应MySQL的全局层级。
- MySQL通过'schema_name.*'表示数据库/模式层级的授权对象；GaussDB通过'{SCHEMA} schema_name'表示模式层级的授权对象。GaussDB的模式层级对应MySQL的数据库/模式层级。
- MySQL中用户名为两部分：用户名@主机名；M-Compatibility当前仅支持用户名。

版权所有 © 华为云计算技术有限公司

- MySQL支持在GRANT赋权语法中修改用户验证，安全连接，资源参数属性，即auth_option、tls_option和resource option；M-Compatibility赋权语法中不支持以上特性，需使用CREATE USER、ALTER USER设置用户相关属性。
- MySQL支持用户代理赋权，GRANT PROXY ON主要用于对多个用户进行统一的权限管理。MySQL 5.7未提供角色机制，而在MySQL 8.0和M-Compatibility中都提供了角色机制。角色能满足用户对于多个用户权限统一管控的目标，可以替代GRANT PROXY ON。
- M-Compatibility拥有public的概念，所用用户都拥有public的权限，部分系统表、系统视图可供所有用户查询。用户可以对public所拥有的权限进行grant和revoke；MySQL中，新创建的用户只拥有全局的usage权限，这个权限很小，几乎为0，只有连接数据库和查询information_schema数据库的权限。
- M-Compatibility中，对象的所有者缺省具有该对象上的所有权限，出于安全考虑所有者可以舍弃部分权限，但ALTER、DROP、COMMENT、INDEX、VACUUM以及对象的可再授予权限属于所有者固有的权限，隐式拥有，无法回收；MySQL中，没有owner的概念，即使用户创建了表，如果没赋予用户对应权限，那么用户也不能对其创建的表进行IUD等操作。
- 在MySQL中，USAGE实际上表示无权限，所用用户都拥有该权限，当执行revoke或grant usage时，实际上不会进行任何修改；在M-Compatibility中，USAGE权限如下：
  - 对于模式，USAGE允许访问包含在指定模式中的对象，若没有该权限，则只能看到这些对象的名称。
  - 对于序列，USAGE允许使用nextval函数。
- 在M-Compatibility中，支持给用户设置管理员角色，包括系统管理员（SYSADMIN）、安全管理员（CREATEROLE）、审计管理员（AUDITADMIN）、监控管理员（MONADMIN）、运维管理员（OPRADMIN）、安全策略管理员（POLADMIN）。默认情况下拥有SYSADMIN属性的系统管理员，具备系统最高权限。三权分立后，系统管理员将不再具有CREATEROLE属性（安全管理员）和AUDITADMIN属性（审计管理员）能力，即不再拥有创建角色和用户的权限，也不再拥有查看和维护数据库审计日志的权限；在MySQL中，不支持该用户设置管理员角色，也没有三权分立相关设计。
- 在M-Compatibility中，可以给用户赋予ANY权限，表示用户能够在非系统模式下拥有对应的权限，包括CREATE ANY TABLE、SELECT ANY TABLE、TRUNCATE ANY TABLE、CREATE ANY INDEX等；在MySQL中，不支持ANY权限的赋予。
- MySQL中提供SHOW GRANTS查询用户权限；M-Compatibility中，可以通过gsql客户端元命令'\l+'、'\dn+'、'\dp'查询权限信息，也可以通过查询pg_namespace、pg_class、pg_attribute等系统表的权限相关字段查询权限信息。
- MySQL中数据库、表、列被删除时，相关的授权信息在系统表中依然保留，如果重新创建同名对象用户依然拥有权限；M-Compatibility中当数据库、表、列被删除时，相关的授权信息会被删除，在重新创建同名对象后需要重新授权。
- MySQL在授予数据库层级的权限时，支持‘_’和‘%’对数据库名进行模糊匹配；M-Compatibility不支持对象名模糊匹配，‘_’或‘%’等特殊字符被识别为普通字符。
- MySQL中，GRANT语句中指定用户不存在时默认会创建该账户（此特性已在MySQL 8.0中移除）；M-Compatibility不支持给未创建用户赋权。

版权所有 © 华为云计算技术有限公司

3.3.2.7.9 系统表和系统视图

表 3-166 M-Compatibility 与 GaussDB 的系统表或系统视图的差异

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_schema.columns	generation_expression	该字段输出结果因涉及M-Compatibility与MySQL的表达式的字符串拼接逻辑的不同而存在差异。
information_schema.columns	data_type	该字段输出结果因涉及M-Compatibility的数据类型format_type输出，目前未修改，与MySQL存在差异。
information_schema.columns	column_type	该字段输出结果因涉及M-Compatibility的数据类型format_type输出，目前未修改，与MySQL存在差异。
information_schema.tables	engine	M-Compatibility中： ● ENGINE对齐information_schema.engines数据。 ● 部分系统表ENGINE为空。 ● 在默认为ASTORE表且未指定STORAGE_TYPE时ENGINE为空。
information_schema.tables	version	M-Compatibility中不支持该字段。
information_schema.tables	row_format	M-Compatibility中不支持该字段。
information_schema.tables	avg_row_length	M-Compatibility下表示使用数据文件除以所有元组数（包括活元组和死元组）的结果。表中没有元组，值为null。
information_schema.tables	max_data_length	M-Compatibility中不支持该字段。
information_schema.tables	data_free	M-Compatibility中表示死元组在总元组中的比例乘以数据文件大小。如果表中没有元组，则为null。
information_schema.tables	check_time	M-Compatibility中不支持该字段。
information_schema.tables	create_time	在M-Compatibility下，此字段与MySQL行为表现有差异，对于创建视图的情形MySQL中该字段置null，M-Compatibility则显示实际的创建表时间。数据库自带的表，视图设置null。
information_schema.tables	update_time	M-Compatibility数据库自带的表，视图设置null。

版权所有 © 华为云计算技术有限公司

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_schema.statistics	collation	M-Compatibility只有值A、D，不会是NULL。
information_schema.statistics	packed	M-Compatibility中不支持该字段。
information_schema.statistics	sub_part	M-Compatibility中不支持该字段。
information_schema.statistics	comment	M-Compatibility中不支持该字段。
information_schema.partitions	subpartition_name	M-Compatibility中，如果分区不是子分区，则为null。
information_schema.partitions	subpartition_ordinal_position	M-Compatibility中，如果分区不是子分区，则为null。
information_schema.partitions	partition_method	M-Compatibility中， 分区策略。如果分区不是一级分区，则为null。 ● 'r'：范围分区。 ● 'i'：间隔分区。 ● 'l'：list分区。 ● 'h'：hash分区。
information_schema.partitions	subpartition_method	M-Compatibility中， 子分区策略。如果分区不是二级分区，则为null。 ● 'r'：范围分区。 ● 'i'：间隔分区。 ● 'l'：list分区。 ● 'h'：hash分区。
information_schema.partitions	partition_description	M-Compatibility中，是区分一级分区和二级分区的。
information_schema.partitions	partition_expression	M-Compatibility中不支持该字段。
information_schema.partitions	subpartition_expression	M-Compatibility中不支持该字段。

版权所有 © 华为云计算技术有限公司

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_schema.partitions	data_length	M-Compatibility中不支持该字段。
information_schema.partitions	max_data_length	M-Compatibility中不支持该字段。
information_schema.partitions	index_length	M-Compatibility中不支持该字段。
information_schema.partitions	data_free	M-Compatibility中不支持该字段。
information_schema.partitions	create_time	M-Compatibility中不支持该字段。
information_schema.partitions	update_time	M-Compatibility中不支持该字段。
information_schema.partitions	check_time	M-Compatibility中不支持该字段。
information_schema.partitions	checksum	M-Compatibility中不支持该字段。
information_schema.partitions	partition_comment	M-Compatibility中不支持该字段。
information_schema.partitions	nodegroup	M-Compatibility中不支持该字段。

版权所有 © 华为云计算技术有限公司

说明

- 视图中对于整型的类型回显，不支持指定精度范围。如MySQL的bigint(1)，M-Compatibility下对应的是bigint类型，MySQL中bigint(21) unsigned，在M-Compatibility下对应的是bigint unsigned类型。
- MySQL中int类型，在M-Compatibility中是integer类型。
- MySQL中的set、enum类型的字段，M-Compatibility中都不支持。m_schema.columns_priv视图的Column_priv字段、m_schema.tables_priv视图的Table_priv,Column_priv字段、m_schema.procs_priv视图的Routine_type,Proc_priv字段、m_schema.proc视图的type,language,sql_data_access,is_deterministic,security_type,sql_mode字段、m_schema.func视图的type字段均不支持，在此版本中不予显示。
- 由于information_schema.tables中table_rows、avg_row_length、data_length、data_free、index_length及information_schema.statistics中cardinality基于统计信息获取，查看前请先执行ANALYZE，更新统计信息后再查看（如果数据库中更新数据，建议延迟执行ANALYZE）。
- information_schema.statistics包含的索引列需要是创建索引中索引列是完整的表列，如果索引列是表达式，不在这个视图中。
- information_schema.partitions 中table_row基于统计信息获取，查看前请先执行ANALYZE，更新统计信息后再查看（如果数据库中更新数据，建议延迟执行ANALYZE）。
- 对于支持的grantee字段，MySQL的格式是'*user_name*'@'*host_name*'，在M-Compatibility数据库，是被授予权限的用户或角色的名称。
- 对于支持的host字段，在M-Compatibility数据库，返回当前节点的hostname。
- m_schema.tables_priv、information_schema.user_privileges、information_schema.schema_privileges，information_schema.table_privileges、information_schema.column_privileges、m_schema.columns_priv、m_schema.func、m_schema.procs_priv 在MySQL下需要授权后才能查看视图内容，M-Compatibility数据库可以根据默认权限查看到对应的内容。如对于表t1，在MySQL下需要先对t1给对应的用户授权，才能在权限视图中看到对应的权限信息，M-Compatibility数据库下则可以直接在视图中看到t1表相关的权限信息。
- m_schema中的系统视图，但在MySQL是系统表。
- information_schema.views中的VIEW_DEFINITION以及information_schema.routines中的ROUTINE_DEFINITION不做字符序控制。
- 《M-Compatibility开发指南》中“Schema”章节所列举的字符类型的视图字段，字符集使用utf8mb4，字符序使用utf8mb4_bin或utf8mb4_general_ci，字符序优先级为《M-Compatibility开发指南》中“SQL参考 > 字符集与字符序 > 字符集和字符序合并规则”所描述的“支持字符序的数据类型的列”的优先级，和MySQL存在差异。

3.3.3 MySQL 兼容性 MYSQL 模式

3.3.3.1 数据类型

3.3.3.1.1 数值数据类型

整数类型

除特别说明外，MySQL兼容性MYSQL模式中的数据类型精度、标度、位数大小等默认不支持用浮点型数值定义，建议使用合法的整型数值定义。

整数类型公共差异说明：

- 输入格式：

版权所有 © 华为云计算技术有限公司

  - MySQL

    对于类似“asbd”、“12dd”、“12 12”等字符场景的输入，会采取截断或返回0值并上报WARNING处理，在严格模式下，插入表会失败。

  - GaussDB

    - 整数类型（TINYINT、SMALLINT、MEDIUMINT、INT、INTEGER、BIGINT）的输入，当非法字符串部分被截断时，如“12@3”或者“1E09”字符串转成整型的科学计数法，会直接截断并无提示信息，插入表成功。

    - 当整数类型全部被截断（如“@123”）或字符串为空时，返回0，且插入表成功。

- 操作符：
  - +、-、*操作符

    GaussDB：INT/INTEGER/SMALLINT/BIGINT在进行运算时，返回值为类型本身，不会向上提升类型，当返回值超范围时报错。

    MySQL：支持提升类型到BIGINT后计算。

  - |、&、^、~运算符

    GaussDB：在类型所占用BIT位中计算；GaussDB中^表示指数运算，如需使用异或运算符，使用#替换。

    MySQL：提升类型计算。

  - <、<=、<=>、<>、=、>、>=

    GaussDB：使用整数类型与字符串类型比较时，将字符串类型转换成整数类型后再进行比较。当字符串中包含小数点、字母等非数字部分时，会直接截断非整数部分进行比较；当字符串被全部被截断（如“@123”）或字符串为空时，字符串类型将转换成0后再与整数类型进行比较。

    MySQL：使用整数类型与字符串类型比较时，将转换成浮点数类型后再进行比较。

- 负数显式类型转换：

  GaussDB：宽松模式结果为0，严格模式报错。

  MySQL：依据其对应的二进制将最高位替换成数值位计算结果，例如(-1)::uint4 = 4294967295。

- 其他差异：

  INT[(M)]精度，MySQL控制格式化输出，GaussDB仅语法支持，不支持功能。

- 聚集函数：
  - variance：GaussDB表示样本方差，MySQL表示总体方差。
  - stddev：GaussDB表示样本标准差，MySQL表示总体标准差。

- 显示宽度：
  - 在为整型数字列指明宽度信息时，如果不同时指定ZEROFILL，则宽度信息在表结构描述中不显示。
  - INSERT语句插入字符类型字段，GaussDB统一补齐0后插入。
  - JOIN USING语句，涉及类型推导，MySQL默认第一张表列，GaussDB若结果为有符号类型则宽度信息失效，否则为第一张表字段宽度。
  - greatest/least、ifnull/if、case when/decode，MySQL不补齐0，GaussDB在类型及宽度信息一致时补齐0。

版权所有 © 华为云计算技术有限公司

- 作为函数/存储过程出入参、返回值时，MySQL支持功能、GaussDB语法不报错功能不支持。

GaussDB数据库和MySQL数据库整数类型具体差异请参见**表3-167**。

表 3-167 整数类型

MySQL数据库	GaussDB数据库	差异
BOOL	支持，存在差异	MySQL：BOOL/BOOLEAN类型实际映射为TINYINT类型。 GaussDB：支持BOOL，其中： • “真”值的有效文本值是：TRUE、't'、'true'、'y'、'yes'、'1' 、'TRUE'、true、'on'以及所有非0数值。 • “假”值的有效文本值是：FALSE、'f'、'false'、'n'、'no'、'0'、0、'FALSE'、false、'off'。 使用TRUE和FALSE是比较规范的用法（也是SQL兼容的用法）。
BOOLEAN	支持，存在差异	
TINYINT[(M)] [UNSIGNED]	支持，存在差异	详情请参见**整数类型公共差异说明**。
SMALLINT[(M)] [UNSIGNED]	支持，存在差异	详情请参见**整数类型公共差异说明**。
MEDIUMINT[(M)] [UNSIGNED]	支持，存在差异	MySQL存储MEDIUMINT数据需要3字节。 • 带符号的范围是-8,388,608 ~ +8,388,607。 • 无符号的范围是0 ~ +16,777,215。 GaussDB映射为INT类型，存储需要4字节。 • 带符号的范围是-2,147,483,648 ~ +2,147,483,647。 • 无符号的范围是0 ~ +4,294,967,295。 其他差异请参见**整数类型公共差异说明**。
INT[(M)] [UNSIGNED]	支持，存在差异	详情请参见**整数类型公共差异说明**。
INTEGER[(M)] [UNSIGNED]	支持，存在差异	详情请参见**整数类型公共差异说明**。
BIGINT[(M)] [UNSIGNED]	支持，存在差异	详情请参见**整数类型公共差异说明**。

版权所有 © 华为云计算技术有限公司

说明

因不同兼容性导致数据存在差异的，当该列上有索引，且使用该索引与创建该索引时的兼容性环境不同时，可能会导致该索引失效，需要reindex该索引。

任意精度类型

表 3-168 任意精度类型

MySQL数据库	GaussDB数据库	差异
DECIMAL[(M[,D])]	支持，存在差异	• 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 • 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 • 输入格式：当字符串入参全部被截断时不会报错，如“@123”；只有被部分截断时才会报错，如“12@3”。
NUMERIC[(M[,D])]	支持，存在差异	
DEC[(M[,D])]	支持，存在差异	
FIXED[(M[,D])]	不支持	-

浮点类型

表 3-169 浮点类型

MySQL数据库	GaussDB数据库	差异
FLOAT[(M,D)]	支持，存在差异	• 分区表支持：FLOAT数据类型不支持KEY键值分区策略分区表。 • 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 • 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 • 输出格式：对于非法入参一律报错ERROR，不会在sql_mode=''的宽松模式下报WARNING。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
FLOAT(p)	支持，存在差异	● 分区表支持：FLOAT数据类型不支持KEY键值分区策略分区表。 ● 操作符：数值类型使用^操作符，与MySQL不一致，GaussDB中^操作符为取指数运算。 ● 取值范围：定义精度p时，仅支持使用合法的整型数据类型。 ● 输出格式：对于非法入参一律报错ERROR，不会在sql_mode=''的宽松模式下报WARNING。
DOUBLE[(M,D)]	支持，存在差异	● 分区表支持：DOUBLE数据类型不支持KEY键值分区策略分区表。 ● 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 ● 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 ● 输出格式：对于非法入参一律报错ERROR，不会在sql_mode=''的宽松模式下报WARNING。
DOUBLE PRECISION[(M,D)]	支持，存在差异	● 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 ● 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 ● 输出格式：对于非法入参一律报错ERROR，不会在sql_mode=''的宽松模式下报WARNING。
REAL[(M,D)]	支持，存在差异	● 分区表支持：REAL数据类型不支持KEY值分区策略分区表。 ● 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 ● 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 ● 输出格式：对于非法入参一律报错ERROR，不会在sql_mode=''的宽松模式下报WARNING。

版权所有 © 华为云计算技术有限公司

序列整数

表 3-170 序列整数

MySQL数据库	GaussDB数据库	差异

SERIAL	支持，存在差异	GaussDB中SERIAL具体介绍请参见**数值类型**。
		规格上与MySQL的差异如下：
		`CREATE TABLE test(f1 serial, f2 CHAR(20));`
		• 类型定义的差异，MySQL的serial是映射到BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE，GaussDB的serial是映射到INTEGER NOT NULL DEFAULT nextval('test_f1_seq'::regclass)。如：
		-- MySQL serial的定义：
		mysql> SHOW CREATE TABLE test\G
		*************************** 1. row ***************************
		Table: test
		Create Table: CREATE TABLE `test` (
		`f1` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
		`f2` char(20) DEFAULT NULL,
		UNIQUE KEY `f1` (`f1`)
		) ENGINE=InnoDB DEFAULT CHARSET=utf8
		1 row in set (0.00 sec)
		
		-- GaussDB serial的定义
		gaussdb=# \d+ test
		Table "public.test"
		Column | Type |
		Modifiers | Storage | Stats target |
		Description
		--------+---------------
		+-----------+----------
		+--------------+-------------
		f1 | integer | not null default
		nextval('test_f1_seq'::regclass) | plain | |
		f2 | character(20) | |
		extended | |
		Has OIDs: no
		Options: orientation=row, compression=no,
		storage_type=USTORE
		• INSERT场景下serial类型DEFAULT值的差异。如：
		-- MySQL插入serial的DEFAULT值
		mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa');
		Query OK, 1 row affected (0.00 sec)
		
		mysql> INSERT INTO test VALUES(10, 'aaaa');
		Query OK, 1 row affected (0.00 sec)
		
		mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa');
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM test;
		+----+------+
		| f1 | f2 |
		+----+------+
		| 1 | aaaa |
		| 10 | aaaa |
		| 11 | aaaa |
		+----+------+
		3 rows in set (0.00 sec)

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异

		-- GaussDB插入serial的DEFAULT值
		gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa');
		INSERT 0 1
		gaussdb=# INSERT INTO test VALUES(10, 'aaaa');
		INSERT 0 1
		gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa');
		INSERT 0 1
		gaussdb=# SELECT * FROM test;
		f1 | f2
		----+----------
		1 | aaaa
		2 | aaaa
		10 | aaaa
		(3 rows)
		• REPLACE场景下serial类型引用列的差异，GaussDB引用列的介绍请参见**REPLACE**。如：
		-- MySQL插入serial引用列的值
		mysql> REPLACE INTO test VALUES(f1, 'aaaa');
		Query OK, 1 row affected (0.00 sec)
		
		mysql> REPLACE INTO test VALUES(f1, 'bbbb');
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM test;
		+----+------+
		| f1 | f2 |
		+----+------+
		| 1 | aaaa |
		| 2 | bbbb |
		+----+------+
		2 rows in set (0.00 sec)
		
		-- GaussDB插入serial引用列的值
		gaussdb=# REPLACE INTO test VALUES(f1, 'aaaa');
		REPLACE 0 1
		gaussdb=# REPLACE INTO test VALUES(f1, 'bbbb');
		REPLACE 0 1
		gaussdb=# SELECT * FROM test;
		f1 | f2
		----+----------
		0 | aaaa
		0 | bbbb
		(2 rows)

版权所有 © 华为云计算技术有限公司

3.3.3.1.2 日期与时间数据类型

表 3-171 日期与时间数据类型

MySQL数据库	GaussDB数据库	差异
DATE	支持，存在差异	GaussDB支持date数据类型，与MySQL相比规格上存在如下差异： ● 输入格式 – GaussDB只支持字符类型，不支持数值类型。如支持'2020-01-01'或'20200101'字符串格式，不支持20200101数值输入。MySQL支持数值输入转换为date类型。 – 分隔符：GaussDB不支持加号“+”、冒号“:”作为年、月、日之间的分隔符，其他符号都支持。MySQL所有符号均可作为分隔符。 分隔符混合使用的某些场景也不支持，与MySQL也有差异，如'2020-01>01'，'2020/01+01'等，不建议混合使用分隔符，建议使用最常用的“-”、“/”作为分隔符。 – 无分隔符：推荐使用完整格式，如 'YYYYMMDD' 或者 'YYMMDD'。其他不完整的格式（包括超长格式）解析的规则与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 ● 输出格式 GaussDB在sql_mode参数不包含'strict_trans_tables'选项（定义为宽松模式，否则为严格模式）时，允许年、月、日的值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值，如date '0000-00-10' 转换为：0002-12-10 BC。 非法输入或者超过范围时，会报warning信息，并返回0000-00-00值。MySQL对于包含0值年、月、日的date值会原样输出。 ● 取值范围 GaussDB的范围是4713-01-01 BC ~ 5874897-12-31 AD，支持公元前的日期，宽松模式下超过范围时，返回的是0值：0000-00-00，严格模式下会报错。MySQL的范围是 0000-00-00 ~ 9999-12-31，宽松模式下超过范围后，各个场景下的表现并不一致，可能报错（如SELECT查询语句中），也可能返回0000-00-00值（如INSERT时）。此差异会导致date类型作为函数入参时，函数返回的结果存在差异。 ● 操作符

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异

		– GaussDB仅支持date类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；date与interval类型的加法运算，返回结果为date类型；date与interval类型的减法运算，返回结果为date类型；date类型之间的减法运算，返回结果为interval类型。
		– MySQL date类型和其他数值类型运算时，会先将date转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如：
		`-- MySQL： date + 数值，先将date类型转换为数值20200101,再与1相加，结果为数值类型20200102。`
		`mysql> SELECT date'2020-01-01' + 1;`
		`+----------------------+`
		`| date'2020-01-01' + 1 |`
		`+----------------------+`
		`| 20200102 |`
		`+----------------------+`
		`1 row in set (0.00 sec)`
		
		`-- GaussDB： date + 数值，数值类型会转换为interval类型1 day，然后相加得到新的日期。`
		`gaussdb=# SELECT date'2020-01-01' + 1;`
		`?column?`
		`------------`
		`2020-01-02`
		`(1 row)`
		● 类型转换
		相比较MySQL，GaussDB仅支持date类型与char(n)、nchar(n)、datetime、timestamp类型之间的相互转换，不支持与binary、decimal、json、integer、unsigned integer、time 类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考**数据类型转换**章节的描述。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
DATETIME[(fsp)]	支持，存在差异	GaussDB支持datetime数据类型，与MySQL相比规格上存在如下差异： ● 输入格式 – GaussDB只支持字符类型，不支持数值类型。如支持'2020-01-01 10:20:30.123456'或'20200101102030.123456'字符串格式，不支持如20200101102030.123456的数值类型输入。MySQL支持数值输入转换为datetime类型。 – 分隔符：GaussDB不支持加号“+”、冒号“:”作为年、月、日之间的分隔符，其他的符号都支持。仅支持冒号“:”作为时、分、秒之间的分隔符，其他的符号都不支持。分隔符混合使用的某些场景也不支持，与MySQL也有差异，不推荐使用。MySQL支持所有符号作为分隔符。 – 无分隔符：GaussDB推荐使用完整格式'YYYYMMDDhhmiss.ffffff'。其他不完整的格式（包括超长格式）解析的规则可能与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 ● 输出格式： – 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的格式，格式与MySQL无差异，且不受DateStyle参数的影响。但是对于精度部分，如果最后几位为0，GaussDB不显示，MySQL会显示。 – GaussDB在sql_mode参数不包含'strict_trans_tables'选项（定义为宽松模式，否则为严格模式）时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值，如datetime '0000-00-10 00:00:00' 转换为：0002-12-10 00:00:00 BC。 非法输入或者超过范围时，会报warning信息，并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的datetime值会原样输出。 ● 取值范围 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。294277-01-09 04:00:54.775807 AD 返回的是infinity。对于超过范围的值，严格模式下GaussDB会报错，MySQL是否报错取决于使用场景。一般查询场景不报错，而执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00值，MySQL根据使用场景可能报错，也可能返回

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异

		0000-00-00 00:00:00值或者null值。这个差异会导致以datetime类型为入参的函数执行结果与MySQL也存在差异。
		● 精度
		范围0~6，作为表列的类型时缺省为0，与MySQL一致。对于 datetime[(p)] 'str' 表达式场景，GaussDB将(p)作为精度解析，缺省为6，将'str'按照p指定的精度格式化成datetime类型。MySQL不支持datetime[(p)] 'str'表达式。
		● 操作符
		– GaussDB仅支持datetime类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；datetime与interval类型的加法运算，返回结果为datetime类型；datetime与interval类型的减法运算，返回结果为datetime类型；datetime类型之间的减法运算，返回结果为interval类型。
		– MySQL datetime类型和其他数值类型运算时，会先将datetime转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如：
		-- MySQL： datetime + 数值，先将datetime类型转换为数值20201010123456,再与1相加，结果为数值类型20201010123457。
		mysql> SELECT cast('2020-10-10 12:34:56.123456' AS datetime) + 1;
		+--+
		| cast('2020-10-10 12:34:56.123456' AS datetime) + 1 |
		+--+
		| 20201010123457 |
		+--+
		1 row in set (0.00 sec)
		
		-- GaussDB： datetime + 数值，数值类型会转换为interval类型 1 day，然后相加得到新的datetime。
		gaussdb=# SELECT cast('2020-10-10 12:34:56.123456' AS datetime) + 1;
		?column?
		--------------------- 	
		2020-10-11 12:34:56
		(1 row)
		将datetime类型与数值的运算结果作为函数的入参，可能导致函数的结果与MySQL也存在差异。
		● 类型转换
		相比较MySQL，GaussDB仅支持datetime类型与char(n)、varchar(n)、timestamp类型之间的相互转换、datetime到date、time类型的转换（仅赋值和显式转换）。不支持与binary、decimal、json、integer、unsigned integer类

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
		型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考**数据类型转换**章节的描述。 ● 时区 GaussDB支持datetime值中携带时区信息（时区偏移或者时区名），如'2020-01-01 12:34:56.123456 +01:00' 或者 '2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。MySQL不支持（5.7版本不支持，8.0及之后的版本支持）。 ● GaussDB的datetime数据类型的表字段实际上会被转换为timestamp(p) without time zone类型，查询表信息或者使用工具导出的表结构，其字段的数据类型显示的是timestamp(p) without time zone，而不是datetime。MySQL显示的是datetime(p)。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
TIMESTAMP[(fsp)]	支持，存在差异	GaussDB支持timestamp数据类型，与MySQL相比规格上存在如下差异： ● 输入格式： – 只支持字符类型，不支持数值类型。如支持'2020-01-01 10:20:30.123456'或'20200101102030.123456'字符串格式，不支持如20200101102030.123456的数值类型输入。MySQL支持数值输入转换为timestamp类型。 – 分隔符：不支持加号“+”、冒号“:”作为年、月、日之间的分隔符，其他的符号都支持。仅支持冒号“:”作为时、分、秒之间的分隔符，其他的符号都不支持。分隔符混合使用的某些场景也不支持，与MySQL也有差异，不推荐使用。MySQL支持所有符号作为分隔符。 – 无分隔符：推荐使用完整格式'YYYYMMDDhhmiss.ffffff'。其他不完整的格式（包括超长格式）解析的规则可能与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 ● 输出格式： – 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的格式，格式与MySQL无差异，且不受DateStyle参数的影响。但是对于精度部分，如果最后几位为0，GaussDB不显示，MySQL会显示。 – GaussDB在sql_mode参数不包含'strict_trans_tables'选项（定义为宽松模式，否则为严格模式）时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值，如timestamp '0000-00-10 00:00:00' 转换为：0002-12-10 00:00:00 BC。 非法输入或者超过范围时，会报warning信息，并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的timestamp值会原样输出。 ● 取值范围： 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。294277-01-09 04:00:54.775807 AD 返回的是infinity。对于超过范围的值，严格模式下GaussDB会报错，MySQL是否报错取决于使用场景。一般查询场景不报错，而执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00值，MySQL根据使用场景可能报错，也可能返回

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
		0000-00-00 00:00:00值或者null值。这个差异会导致以timestamp类型为入参的函数执行结果与MySQL也存在差异。 ● 精度： 范围0~6，作为表列的类型时缺省为0，与MySQL一致。对于 timestamp[(p)] 'str' 表达式场景： – GaussDB将(p)作为精度解析，缺省为6，将'str'按照p指定的精度格式化成timestamp类型。 – MySQL将timestamp 'str'的含义与GaussDB一致，缺省精度也为6。但是将timestamp(p) 'str'解析为函数调用，p作为timestamp函数的入参，结果返回一个timestamp类型的值，'str'作为投影列的别名。 ● 操作符： – GaussDB仅支持timestamp类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；timestamp与interval类型的加法运算，返回结果为timestamp类型；timestamp与interval类型的减法运算，返回结果为timestamp类型；timestamp类型之间的减法运算，返回结果为interval类型。 – MySQL timestamp类型和其他数值类型运算时，会先将timestamp转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如： -- MySQL： timestamp + 数值，先将timestamp类型转换为数值20201010123456.123456,再与1相加，结果为数值类型20201010123457.123456。 mysql> SELECT timestamp '2020-10-10 12:34:56.123456' + 1; +--+ | timestamp '2020-10-10 12:34:56.123456' + 1 | +--+ | 20201010123457.123456 | +--+ 1 row in set (0.00 sec) -- GaussDB： timestamp + 数值，数值类型会转换为interval类型 1 day，然后相加得到新的timestamp。 gaussdb=# SELECT timestamp '2020-10-10 12:34:56.123456' + 1; ?column? ---------- 2020-10-11 12:34:56.123456 (1 row) 将timestamp类型与数值的运算结果作为函数的入参，可能导致函数的结果与MySQL也存在差异。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
		• 类型转换： 相比较MySQL，GaussDB仅支持timestamp类型与char(n)、varchar(n)、datetime类型之间的相互转换、timestamp到date、time类型的转换（仅赋值和显式转换）。不支持与binary、decimal、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考**数据类型转换**章节的描述。 • 时区： GaussDB支持timestamp值中携带时区信息（时区偏移或者时区名），如'2020-01-01 12:34:56.123456 +01:00' 或者 '2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。如果更改服务器时区，timestamp类型的值输出时会转换为更改后时区的时间戳。MySQL不支持（5.7版本不支持，8.0及之后的版本支持）。 • GaussDB的timestamp数据类型的表字段实际上会被转换为timestamp(p) with time zone类型，查询表信息或者使用工具导出的表结构，其字段的数据类型显示的是timestamp(p) with time zone，而不是timestamp。MySQL显示的是timestamp(p)。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
TIME[(fsp)]	支持，存在差异	GaussDB支持time数据类型，与MySQL相比规格上存在如下差异： ● 输入格式： – 只支持字符类型，不支持数值类型。如支持'1 10:20:30'或'102030'字符串格式，不支持102030数值输入。MySQL支持数值输入转换为time类型。 – 分隔符：GaussDB仅支持冒号“:”作为时、分、秒之间的分隔符，其他的符号都不支持。MySQL支持所有的符号作为分隔符。 – 无分隔符：推荐使用完整格式，如'hhmiss.ffffff'。其他不完整的格式（包括超长格式）解析的规则可能与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 – 分、秒、精度输入负数时，GaussDB数据库可能会忽略第一个负数开始的部分，涉及的部分解析为0，如：'00:00:-10' 解析结果为'00:00:00'。 也可能报错，如：'00:00:-10000' 会解析报错。取决于输入值的范围。而MySQL数据库统一报错。 ● 输出格式： 统一为hh:mi:ss.ffffff的格式，格式与MySQL无差异。但是对于精度部分，如果最后几位为0，GaussDB不显示，MySQL会显示。 ● 取值范围： -838:59:59.000000 ~ 838:59:59.000000，与MySQL一致。对于超过范围的值，GaussDB在宽松模式下执行SELECT、INSERT、UPDATE等DML操作时，返回的值都是就近的边界值：-838:59:59或838:59:59。MySQL是查询时报错，DML操作返回的值才是就近边界值，场景上存在差异。此差异会导致time类型作为函数入参时，函数返回的结果也存在差异。 ● 精度： 范围0~6，作为表列的类型时缺省为0，与MySQL一致。对于 time(p) 'str' 表达式场景，GaussDB将(p)作为精度解析，缺省为6，将'str'按照p指定的精度格式化成time类型。MySQL是解析为time函数，p是入参，'str'是投影列的别名。 ● 操作符： – GaussDB仅支持time类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；time与

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
		interval类型的加法运算，返回结果为time类型；time与interval类型的减法运算，返回结果为time类型；time类型之间的减法运算，返回结果为interval类型。 – MySQL time类型和其他数值类型运算时，会先将time转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如： -- MySQL： time + 数值，先将time类型转换为数值123456，再与1相加，结果为数值类型123457。 mysql> SELECT time '12:34:56' + 1; +---------------------+ | time '12:34:56' + 1 | +---------------------+ | 123457 | +---------------------+ 1 row in set (0.00 sec) -- GaussDB： time + 数值，数值类型会转换为interval类型1 day，然后相加得到新的time，由于是加了24小时，得到的仍然是12:34:56。 gaussdb=# SELECT time '12:34:56' + 1; ?column? ---------- 12:34:56 (1 row) 将time类型与数值的运算结果作为函数的入参，可能导致函数的结果与MySQL也存在差异。 ● 类型转换： 相比较MySQL，GaussDB仅支持time类型与char(n)、nchar(n)类型之间的相互转换、datetime、timestamp到time类型的转换。不支持与binary、decimal、date、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考**数据类型转换**章节的描述。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
YEAR[(4)]	支持，存在差异	GaussDB支持year数据类型，与MySQL相比规格上存在如下差异： ● 操作符： – GaussDB仅支持year类型之间的比较操作符：=、!=、<、<=、>、>=。返回true或者false。 – GaussDB仅支持year类型与int4类型之间的算术操作符：+、-。返回整型值，MySQL是返回无符号整型值。 ● 类型转换： 相比较MySQL，GaussDB仅支持year类型与int4类型的转换，仅支持int4、varchar、numeric、date、time、timestamp、timestamptz类型到year类型的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考**数据类型转换**章节的描述。
INTERVAL	支持，存在差异	GaussDB支持INTERVAL数据类型，但INTERVAL在MySQL中为表达式，同时存在以下差异： ● 不支持字符串类型的日期输入作为运算，如：SELECT '2023-01-01' + interval 1 day。 ● 不支持interval expr unit语法中，expr为负整数或浮点数的输入，如：SELECT date'2023-01-01' + interval -1 day。 ● 不支持interval expr unit语法中，expr为运算表达式的输入，如：SELECT date'2023-01-01' + interval 4/2 day。 ● interval表达式参与运算时，返回值固定为datetime类型，MySQL为datetime或date类型。运算的逻辑与原有GaussDB保持一致，与MySQL有差异。 ● interval expr unit语法中，expr数值支持的范围会根据unit单位的不同有所差异，最大可支持的范围为[-2147483648, 2147483647]。超过范围时，严格模式报error，宽松模式报warning并返回0值。 ● interval expr unit语法中，expr指定的字段数量大于unit预期的字段数量时，在严格模式，报error；在宽松模式，报warning并返回0值。如unit取值为DAY_HOUR，预期的字段数量为2，expr取值为'1-2-3'，字段数量为3。

版权所有 © 华为云计算技术有限公司

3.3.3.1.3 字符串数据类型

表 3-172 字符串数据类型

MySQL数据库	GaussDB数据库	差异
CHAR[(M)]	支持，存在差异	• 输入格式 – GaussDB自定义函数参数和返回值不支持长度校验，存储过程参数不支持长度校验，同时也不支持在PAD_CHAR_TO_FULL_LENGTH打开时补齐正确的空格，MySQL支持。 – GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 • 语法 GaussDB的 Cast（expr as char）语法无法根据输入的字符串长度转成对应的类型，只支持转成varchar类型。不支持cast('' as char) 和 cast('' as char(0))将空串转成char(0)类型。MySQL支持按长度转成对应的类型。 • 操作符 – GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 – GaussDB除以0会报错，MySQL返回null。 – “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 – “^”：GaussDB表示次方幂，MySQL表示按位异或。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
VARCHAR(M)	支持，存在差异	● 输入格式 – GaussDB的自定义函数参数和返回值不支持长度校验，存储过程参数不支持长度校验，MySQL支持。 – GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警，MySQL不支持。 – GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 – GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 – GaussDB除以0会报错，MySQL返回null。 – “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 – “^”：GaussDB表示次方幂，MySQL表示按位异或。
TINYTEXT	支持，存在差异	● 输入格式 – GaussDB中该类型的长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过255字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 – GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 – GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 – GaussDB除以0会报错，MySQL返回null。 – “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 – “^”：GaussDB表示次方幂，MySQL表示按位异或。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
TEXT	支持，存在差异	● 输入格式 – GaussDB中该类型的长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过65535字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 – GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 – GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 – GaussDB除以0会报错，MySQL返回null。 – “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 – “^”：GaussDB表示次方幂，MySQL表示按位异或。
MEDIUMTEXT	支持，存在差异	● 输入格式 – GaussDB中该类型的长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过16777215字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 – GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 – GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 – GaussDB除以0会报错，MySQL返回null。 – “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 – “^”：GaussDB表示次方幂，MySQL表示按位异或。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
LONGTEXT	支持，存在差异	● 输入格式 – GaussDB只支持不超过1GB的长度，MySQL支持4GB-1字节的长度。 – GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 – GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 – GaussDB除以0会报错，MySQL返回null。 – “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 – “^”：GaussDB表示次方幂，MySQL表示按位异或。
ENUM('value1','value2',...)	不支持	-
SET('value1','value2',...)	不支持	-

3.3.3.1.4 二进制数据类型

表 3-173 二进制数据类型

MySQL数据库	GaussDB数据库	差异
BINARY[(M)]	不支持	-
VARBINARY(M)	不支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
TINYBLOB	支持，存在差异	● 取值范围：GaussDB中该类型由BYTEA类型映射得来，长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过255字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 ● 输入格式：不支持转义字符输入，不支持""双引号输入。 ● 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 ● 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“||”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“|”、“^”。
BLOB	支持，存在差异	● 取值范围：GaussDB中该类型由BYTEA类型映射得来，长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过65535字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 ● 输入格式：不支持转义字符输入，不支持""双引号输入。 ● 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 ● 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“||”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“|”、“^”。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
MEDIUMBLOB	支持，存在差异	• 取值范围：GaussDB中该类型由BYTEA类型映射得来，长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过16777215字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 • 输入格式：不支持转义字符输入，不支持""双引号输入。 • 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 • 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“||”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“|”、“^”。
LONGBLOB	支持，存在差异	• 取值范围：GaussDB中该类型由BYTEA类型映射得来，只支持不超过1GB的长度，具体范围参照BYTEA数据类型集中式和分布式规格。 • 输入格式：不支持转义字符输入，不支持""双引号输入。 • 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 • 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“||”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“|”、“^”。
BIT[(M)]	不支持	-

3.3.3.1.5 JSON 数据类型

表 3-174 JSON 数据类型

MySQL数据库	GaussDB数据库	差异
JSON	支持，存在差异	• GaussDB数据库MySQL兼容性MYSQL模式中的JSON类型与GaussDB数据库原生的JSON类型行为一致，与MySQL行为差异较大，此处不再逐个列出。 • GaussDB数据库MySQL兼容性MYSQL模式中的JSON类型具体行为请参见**JSON/JSONB类型**。

版权所有 © 华为云计算技术有限公司

3.3.3.1.6 数据类型支持的属性

表 3-175 数据类型支持的属性

MySQL数据库	GaussDB数据库
NULL	支持
NOT NULL	支持
DEFAULT	支持
ON UPDATE	支持
PRIMARY KEY	支持
CHARACTER SET name	支持
COLLATE name	支持

3.3.3.1.7 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换：

- 操作符（比较操作符、运算操作符等）的操作数的数据类型不一致。常见于查询条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新（包括INSERT、UPDATE、MERGE、REPLACE等）的目标列，数据的类型和列的定义类型不一致。
- 显式的类型转换：cast(expr AS datatype)，将expr表达式类型转换为datatype类型。
- 集合运算（UNION、EXCEPT）确定最终投影列的目标数据类型后，各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景，根据不同表达式的数据类型， 来决定用于比较或者最终结果的目标数据类型。
  - DECODE
  - CASE WHEN
  - lexpr [NOT] IN (expr_list)
  - BETWEEN AND
  - JOIN USING(a,b)
  - GREATEST和LEAST
  - NVL 和 COALESCE

GaussDB和MySQL数据库对于数据类型转换、转换的目标数据类型有着完全不同的规则。如下示例体现了两者处理的差异：

- 示例一

```
-- MySQL： in 执行结果为0，表示false。根据规则，会将'1970-01-01'与列表中的表达式依次比较，结果都为0，因此最终结果为0。
mysql> SELECT '1970-01-01' IN ('1970-01-02', 1, '1970-01-02');
+-----------------------------------------------+
```


版权所有 © 华为云计算技术有限公司

```
| '1970-01-01' IN ('1970-01-02', 1, '1970-01-02') |
+-------------------------------------------------+
|                                               0 |
+-------------------------------------------------+

-- GaussDB： in 执行结果为true，与MySQL结果相反：根据规则选定的公共类型为int类型，因此将左表达式'1970-01-01'转换为int类型与列表中的表达式转换为int类型后的值依次比较。
-- '1970-01-01'和'1970-01-02'转换为int类型时都为1970（MySQL兼容性下，转换时遇到非法字符后忽略，将前面部分转换为int类型），比较结果为相等，因此返回结果为true。
gaussdb=# SELECT '1970-01-01' IN ('1970-01-02', 1::int, '1970-01-02') AS result;
 result
--------
 t
(1 row)
```

- 示例二

```
-- MySQL中，整数类型与字符串类型比较时，将转换成浮点数类型后再进行比较。
mysql> SELECT '2.1' = 2, '2.1' > 2, '2.1' < 2, '.123' = 0;
+-----------+-----------+-----------+------------+
| '2.1' = 2 | '2.1' > 2 | '2.1' < 2 | '.123' = 0 |
+-----------+-----------+-----------+------------+
|         0 |         1 |         0 |          0 |
+-----------+-----------+-----------+------------+
1 row in set (0.00 sec)

-- GaussDB中，整型与字符串类型比较时，字符串类型将会转换成整数类型后进行比较，'2.1'在转整型时会截断非整数结果，结果为2；'.123'在转整型时会直接转换为0。
gaussdb=# SELECT '2.1' = 2, '2.1' > 2, '2.1' < 2, '.123' = 0;
 ?column? | ?column? | ?column? | ?column?
----------+----------+----------+----------
 t        | f        | f        | t
(1 row)
```

数据类型转换规则的差异：

- GaussDB数据库对于不同数据类型之间的转换规则有明确的定义：
  - 是否支持转换：pg_cast系统表中是否定义两种类型的转换路径，没有定义则不支持。
  - 支持转换的场景：支持任意场景转换、仅支持显式（cast表达式）转换、仅支持赋值时转换。不支持的场景下即使定义了转换路径，也不能进行数据类型转换。
- MySQL数据库支持任意两种数据类型之间做转换。

由于存在以上差异，基于MySQL数据库的应用程序向GaussDB数据库迁移时，SQL语句可能由于不支持不同数据类型之间的转换而报错。或者支持转换的场景下，转换的规则有差异导致SQL语句执行的结果不同。

推荐的做法是：SQL语句中尽量使用相同的数据类型做比较或者赋值等操作，避免因为数据类型转换导致非预期结果或者性能损耗。

选择目标数据类型的规则差异：

对于有些场景，比较的数据类型或者返回的数据类型需要综合考虑多个表达式的类型才能确定。比如UNION运算中，不同SELECT语句中相同位置的投影列具有不同的数据类型，查询结果的最终数据类型，需要由各个SELECT语句投影列的数据类型共同确定。

确定目标数据类型的规则，GaussDB数据库和MySQL数据库存在体系上的差异。

- GaussDB数据库规则：
  - 操作符的操作数类型不一致时，并不是将操作数的类型统一转换为目标类型再计算。而是直接注册两个数据类型的操作符，操作符处理中定义两个不同

 版权所有 © 华为云计算技术有限公司

类型的处理规则。此方式不存在类型隐式转换，但自定义的处理规则隐含了转换的操作。

- 集合运算和表达式场景，确定目标数据类型的规则：
  - 如果所有类型都相同，则此类型即为目标类型。
  - 两个数据类型如果不同，检查数据类型是否属于同一种类的数据类型，如数值类型、字符类型、日期时间类型等。不属于同一种类的数据类型，无法确定目标类型，此时SQL语句执行会报错。
  - 对于category属性（在pg_type系统表中定义）相同的数据类型，具有preferred属性（在pg_type系统表中定义）的数据类型会被选为目标类型。或者操作数1能转换为操作数2（没有转换路径），而操作数2无法转换为操作数1或数值类型优先级小于操作数2，则选择操作数2作为目标类型。
  - 如果涉及到3个及以上的数据类型，确定目标类型的规则为：common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3)，依次迭代处理，得到最终的结果。
  - 对于IN和NOT IN表达式，如果根据以上规则无法确认目标类型，会将lexpr与expr_list中每一个表达式单独按照等值操作符（=）逐个比较。
  - 精度的确定：以最终选定的表达式的精度作为最终结果。

- MySQL数据库规则：
  - 操作符的操作数类型不一致时，先按照如下规则确定目标类型。确定后将类型不一致的操作数转换成目标类型后再做处理。
    - 两个参数都是string类型，则都按照string类型比较。
    - 两个参数都是integer类型，则都按照integer类型比较。
    - 十六进制数值如果不与数值比较，则当做二进制字符串比较。
    - 一个参数是datetime/timestamp类型，另一个参数是常量，将常量转换为时间戳类型然后比较。
    - 如果其中一个参数是decimal类型，比较时使用的数据类型取决于另外一个参数。另外一个是decimal或者integer类型时，按照decimal类型；另外一个是其他类型，按照real类型比较。
    - 其他场景都转换为 real 类型后比较。
  - 集合运算和表达式场景，确定目标数据类型的规则如下：
    - 建立任意两个类型之间的目标类型矩阵。给定两个类型，通过矩阵即可以确定目标类型。
    - 如果涉及到3个及以上的数据类型，确定目标类型的规则为：common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3)，依次迭代处理，得到最终的结果。

 版权所有 © 华为云计算技术有限公司

- 如果目标类型是integer类型，且各个表达式类型包含有符号和无符号的混合场景，则会将类型提升到更高精度的integer类型。符号的确定：所有表达式都是无符号时，结果才为无符号，否则结果为有符号。
- 精度确定：以表达式中的最大精度作为最终结果。

从以上规则可知：GaussDB和MySQL数据库在数据类型的转换规则上有很大差异，不能直接对比。在上述场景下，SQL语句的执行结果可能和MySQL数据库不一致。当前版本推荐各个表达式使用相同的类型，或提前使用cast转换成需要的类型来规避差异。

3.3.3.2 系统函数

3.3.3.2.1 系统函数兼容性概述

GaussDB数据库兼容绝大多数MySQL的系统函数，但存在部分差异。

除特别说明外，MySQL兼容性MYSQL模式中的函数行为默认为GaussDB原生行为。

说明

GaussDB MySQL兼容性的绝大部分系统函数目前均存在返回值与MySQL精度不一致（结果后面0的位数）的问题，这是由于部分数据类型在某些场景下仍存在精度丢失问题，无法正确获取精度，导致目前部分函数未完全做到适配。

3.3.3.2.2 流量控制函数

表 3-176 流量控制函数列表

MySQL数据库	GaussDB数据库	差异
IF()	支持，存在差异	• expr1入参仅支持bool类型。非bool类型入参若不能转换为bool类型则报错。 • 若expr2、expr3两个入参类型不同且两类型间不存在隐式转换函数则报错。 • 两个入参类型相同时，返回该入参类型。 • 若expr2、expr3两个入参类型分别为NUMERIC、STRING或TIME其中一个，GaussDB输出为TEXT类型，MySQL输出为VARCHAR类型。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
IFNULL()	支持，存在差异	• 若expr1、expr2两个入参类型不同且两类型间不存在隐式转换函数则报错。 • 两个入参类型相同时，返回该入参类型。 • 若expr1、expr2两个入参类型范畴分别为NUMERIC、STRING或TIME其中一个，GaussDB输出为TEXT类型，MySQL输出为VARCHAR类型。 • 两个入参类型中一个为FLOAT4类型，另一个为NUMERIC范畴中任一类型，返回值为DOUBLE类型。MySQL其中一入参为FLOAT4，另一入参为TINYINT、UNSIGNED TINYINT、SMALLINT、UNSIGNED SMALLINT、MEDIUMINT、UNSIGNED MEDIUMINT、BOOL任一类型时，返回FLOAT4类型，第一个入参为FLOAT4，第二个入参为BIGINT或UNSIGNED BIGINT时，返回FLOAT类型。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异

NULLIF()	支持，存在差异	● GaussDB中NULLIF()类型推导遵从以下逻辑：
		– 如果两个参数的数据类型不同，且两个入参类型存在等值比较操作符，则返回对应等值操作符对应的左值类型，否则会对两个入参类型进行强制类型兼容。
		– 若强制类型兼容后，存在等值比较操作符，则返回强制类型兼容后对应等值操作符的左值类型。
		– 若强制类型兼容后，仍找不到对应等值操作符，则报错。
		--两个入参类型存在等值比较操作符
		gaussdb=# SELECT pg_typeof(nullif(1::int2, 2::int8));
		pg_typeof
		----------- 	
		smallint
		(1 row)
		--两个入参类型不存在等值比较操作符，但在强制类型兼容后可以找到等值比较操作符
		gaussdb=# SELECT pg_typeof(nullif(1::int1, 2::int2));
		pg_typeof
		----------- 	
		bigint
		(1 row)
		
		--两个入参类型不存在等值比较操作符，且强制类型兼容后也不存在等值比较操作符
		gaussdb=# SELECT nullif(1::bit, '1'::MONEY);
		ERROR: operator does not exist: bit = money
		LINE 1: SELECT nullif(1::bit, '1'::MONEY);
		^
		HINT: No operator matches the given name and argument type(s). You might need to add explicit type casts.
		CONTEXT: referenced column: nullif
		● MySQL输出类型仅与第一个入参类型有关：
		– 第一个入参为tinyint、smallint、mediumint、int、bool时，输出为int类型。
		– 第一个入参为bigint时，输出为bigint类型。
		– 第一个入参为unsigned tinyint、unsigned smallint、unsigned mediumint、unsigned int、bit时，输出为unsigned int类型。
		– 第一个入参为unsigned bigint时，输出为unsigned bigint。
		– 第一个入参为浮点型即float、double、real时，输出为double类型。
		– 第一个入参类型为decimal或numeric类型时，输出为decimal类型。
		– 第一个入参类型为时间类型或字符串类型即date、time、datetime、timestamp、char、varchar以及tinytext、enum、set时，输出为varchar类型。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
		– 第一个入参类型为text、mediumtext、longtext时，输出为longtext类型。 – 第一个入参类型为tinyblob时，输出为varbinary类型。 – 第一个入参类型为mediumblob或longblob时，输出为longblob类型。 – 第一个入参为blob时，输出为blob类型。
ISNULL()	支持，存在差异	GaussDB中返回值为BOOLEAN类型的t或f，MySQL中返回值为INT类型的1或0。

3.3.3.2.3 日期和时间函数

以下为GaussDB数据库MySQL兼容性MYSQL模式中日期时间函数的公共说明，与MySQL行为一致。

- 函数入参为时间类型表达式的情况：

  时间类型表达式主要包括text、datetime、date或time，但所有可以隐式转换为时间表达式的类型都可以作为入参，比如数字类型可以通过先隐式转化为text，再作为时间类型表达式生效。

  但是，不同函数的具体生效情况会有所不同。例如：datediff函数仅计算日期之间的差值，因此时间表达式会被解析为日期；而timestampdiff函数在计算时间差值时，会根据unit参数来决定将时间表达式解析为date、time或 datetime。

- 函数入参为无效日期的情况：

  一般而言，日期时间函数支持date、datetime的范围和MySQL保持一致。date支持的范围为'0000-01-01'到'9999-12-31'，datetime支持的范围为'0000-01-01 00:00:00'到'9999-12-31 23:59:59'。虽然GaussDB支持的date、datetime范围大于MySQL，但是越界仍然算无效日期。

  大部分时间函数对于入参为无效时间时，会告警并返回NULL，只有能通过cast正常转换的日期，才是正常合理的日期。

- 函数入参的分隔符场景：

  对于时间函数，处理入参时会将所有非数字字符视作分隔符，然后根据数字所处的位置进行计算，推荐使用标准写法，年月日之间使用-分隔符，时分秒之间使用:分隔符，毫秒之前通过.来进行分隔。

  易错场景：在MySQL兼容性MYSQL模式数据库中执行“SELECT timestampdiff(hour, '2020-03-01 00:00:00', '2020-02-28 00:00:00+08');”时，时间函数不会自动计算时区，所以此处+08并未识别为时区，而是+作为分隔符当作秒来进行计算。

GaussDB的日期时间函数的大部分功能场景与MySQL一致，但仍有差异，部分差异如下：

- 函数入参为NULL时，函数返回NULL，无warning或error告警。这些函数包括：

  from_days、date_format、str_to_date、datediff、timestampdiff、date_add、subtime、month、time_to_sec、to_days、to_seconds、dayname、monthname、convert_tz、sec_to_time、addtime、adddate、date_sub、

版权所有 © 华为云计算技术有限公司

timediff、last_day、weekday、from_unixtime、unix_timestamp、subdate、day、year、weekofyear、dayofmonth、dayofyear、week、yearweek、dayofweek、time_format、hour、minute、second、microsecond、quarter、get_format、extract、makedate、period_add、timestampadd、period_diff、utc_time、utc_timestamp、maketime、curtime

示例：

```
gaussdb=# SELECT day(null);
 day
-----

(1 row)
```

- 纯数字入参个别函数与MySQL有差异，不带引号的数字入参统一转成text入参来处理。

  示例：

  ```
  gaussdb=# SELECT day(19231221.123141);
  WARNING:  Incorrect datetime value: "19231221.123141"
  CONTEXT:  referenced column: day
   day
  -----

  (1 row)
  ```

- 时间日期运算函数：adddate、subdate、date_add、date_sub。当运算后的结果为日期时，支持的范围为[0000-01-01, 9999-12-31]，当运算后的结果为日期时间时，支持的范围为[0000-01-01 00:00:00.000000, 9999-12-31 23:59:59.999999]，当运算后的结果超过支持的范围时，在严格模式下报error，在宽松模式下报warning。另外，当运算后的日期结果在范围[0000-01-01, 0001-01-01]中时，GaussDB正常返回结果，MySQL返回'0000-00-00'。

  示例：

  ```
  gaussdb=# SELECT subdate('0000-01-01', interval 1 hour);
  ERROR:  Datetime function: datetime field overflow
  CONTEXT:  referenced column: subdate

  gaussdb=# SELECT subdate('0001-01-01', interval 1 day);
    subdate
  -------------
   0000-12-31

  (1 row)
  ```

- 对于日期和时间函数的date或datetime类型入参，含有0月或0日则为非法值，在严格模式下报error；在宽松模式，当输入为字符串或数字时，报warning，输入为date或datetime类型时视为上一年12月或上一月最后一日处理。

  对于cast函数，转换为date、datetime时，严格模式下会报error；宽松模式下不会报warning，而是视为上一年12月或上一月最后一日处理，需要注意此区别。MySQL对于包含0年、0月或0日的情况会原样输出。

  示例：

  ```
  gaussdb=# SELECT adddate('2023-01-00', 1);-- 严格模式
  ERROR:  Incorrect datetime value: "2023-01-00"
  CONTEXT:  referenced column: adddate

  gaussdb=# SELECT adddate('2023-01-00', 1);-- 宽松模式
  WARNING:  Incorrect datetime value: "2023-01-00"
  CONTEXT:  referenced column: adddate
   adddate
  ---------

  (1 row)

  gaussdb=# SELECT adddate(date'2023-00-00', 1);-- 宽松模式
  ```


版权所有 © 华为云计算技术有限公司

```
 adddate
------------
 2022-12-01
(1 row)

gaussdb=# SELECT cast('2023/00/00' AS date);-- 宽松模式
    date
------------
 2022-11-30
(1 row)

gaussdb=# SELECT cast('0000-00-00' AS datetime);-- 宽松模式
      timestamp
---------------------
 0000-00-00 00:00:00
(1 row)
```

- 若函数入参为numeric数据类型，在非法输入的情况下不会产生报错，会把入参当做0值处理。

  示例：

```
gaussdb=# SELECT from_unixtime('aa');
    from_unixtime
---------------------
 1970-01-01 08:00:00
(1 row)
```

- 最多保留6位小数，不保留后置都为0的小数。

  示例：

```
gaussdb=# SELECT from_unixtime('1234567899.00000');
    from_unixtime
---------------------
 2009-02-14 07:31:39
(1 row)
```

- 时间函数参数为字符串时，只保证年月日之间使用“-”分隔，时分秒之间使用“:”分隔时结果正确。

  示例：

```
gaussdb=# SELECT adddate('20-12-12',interval 1 day);
 adddate
------------
 2020-12-13
(1 row)
```

- 在MySQL中，当函数的返回值为varchar时，在GaussDB中，函数对应的返回值为text。

```
-- GaussDB中函数的返回值。
gaussdb=# SELECT pg_typeof(adddate('2023-01-01', 1));
 pg_typeof
-----------
 text
(1 row)

-- MySQL中函数的返回值。
mysql> CREATE VIEW v1 AS SELECT adddate('2023-01-01', 1);
Query OK, 0 rows affected (0.00 sec)

mysql> DESC v1;
+--------------------------+-------------+------+-----+---------+-------+
| Field                    | Type        | Null | Key | Default | Extra |
+--------------------------+-------------+------+-----+---------+-------+
| adddate('2023-01-01', 1) | varchar(29) | YES  |     | NULL    |       |
+--------------------------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```


版权所有 © 华为云计算技术有限公司

表 3-177 日期和时间函数列表

MySQL数据库	GaussDB数据库	差异
ADDDATE()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见**INTERVAL差异说明**。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
ADDTIME()	支持，存在差异	• MySQL对第二入参为DATETIME样式字符串返回NULL，GaussDB可以计算。 • 入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。 • MySQL中ADDTIME函数如果第一个参数是动态参数（例如在预准备语句中），则返回类型为TIME。否则，函数的解析类型派生自第一个参数的解析类型。GaussDB中ADDTIME函数的返回值规则如下： – 第一个入参为date，第二个入参为date，返回值为time。 – 第一个入参为date，第二个入参为text，返回值为text。 – 第一个入参为date，第二个入参为datetime，返回值为time。 – 第一个入参为date，第二个入参为time，返回值为time。 – 第一个入参为text，第二个入参为date，返回值为text。 – 第一个入参为text，第二个入参为text，返回值为text。 – 第一个入参为text，第二个入参为datetime，返回值为text。 – 第一个入参为text，第二个入参为time，返回值为text。 – 第一个入参为datetime，第二个入参为date，返回值为datetime。 – 第一个入参为datetime，第二个入参为text，返回值为text。 – 第一个入参为datetime，第二个入参为datetime，返回值为datetime。 – 第一个入参为datetime，第二个入参为time，返回值为datetime。 – 第一个入参为time，第二个入参为date，返回值为time。 – 第一个入参为time，第二个入参为text，返回值为text。 – 第一个入参为time，第二个入参为datetime，返回值为time。 – 第一个入参为time，第二个入参为time，返回值为time。
CONVERT_TZ()	支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
CURDATE()	支持	-
CURRENT_DATE(), CURRENT_DATE	支持	-
CURRENT_TIME(), CURRENT_TIME	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，其他均报错；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，超过6的整型值，会告警并按照精度6输出时间值；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
CURTIME()	支持，存在差异	GaussDB此函数输入字符串或者非整型值，会被隐式转成整型，然后再校验精度，[0,6]范围之外的会报错，范围之内会正常输出时间值；MySQL直接报错。GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，其他均报错；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
YEARWEEK()	支持	-
DATE_ADD()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见**INTERVAL差异说明**。
DATE_FORMAT()	支持	-
DATE_SUB()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见**INTERVAL差异说明**。
DATEDIFF()	支持	-
DAY()	支持	-
DAYNAME()	支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
DAYOFMONTH()	支持	-
DAYOFWEEK()	支持	-
DAYOFYEAR()	支持	-
EXTRACT()	支持	-
FROM_DAYS()	支持	-
FROM_UNIXTIME()	支持	-
GET_FORMAT()	支持	-
HOUR()	支持	-
LAST_DAY	支持	-
LOCALTIME(), LOCALTIME	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，其他整型值直接报错；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
LOCALTIMESTAMP, LOCALTIMESTAMP()	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，超过6的整型值，会告警并按照精度6输出时间值；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
MAKEDATE()	支持	-
MAKETIME()	支持，存在差异	与MySQL相比, 入参为NULL时，GaussDB不支持maketime函数自嵌套，MySQL支持。
MICROSECOND()	支持	-
MINUTE()	支持	-
MONTH()	支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
MONTHNAME()	支持	-
NOW()	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，超过6的整型值，会告警并按照精度6输出时间值；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
PERIOD_ADD()	支持，存在差异	当入参period或结果小于0时，GaussDB参考MySQL 8.0.x版本的表现，报错处理。MySQL 5.7会发生整数回绕，导致计算结果异常。
PERIOD_DIFF()	支持，存在差异	当入参或结果小于0时，GaussDB参考MySQL 8.0.x版本的表现，报错处理。MySQL 5.7会发生整数回绕，导致计算结果异常。
QUARTER()	支持	-
SEC_TO_TIME()	支持	-
SECOND()	支持	-
STR_TO_DATE()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是text，MySQL返回的是datetime、date。
SUBDATE()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见**INTERVAL差异说明**。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
SUBTIME()	支持，存在差异	• MySQL对第二入参为DATETIME样式字符串返回NULL，GaussDB可以计算。 • 入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。 • MySQL中SUBTIME函数如果第一个参数是动态参数（例如在预准备语句中），则返回类型为TIME。否则，函数的解析类型派生自第一个参数的解析类型。GaussDB中SUBTIME函数的返回值规则如下： – 第一个入参为date，第二个入参为date，返回值为time。 – 第一个入参为date，第二个入参为text，返回值为text。 – 第一个入参为date，第二个入参为datetime，返回值为time。 – 第一个入参为date，第二个入参为time，返回值为time。 – 第一个入参为text，第二个入参为date，返回值为text。 – 第一个入参为text，第二个入参为text，返回值为text。 – 第一个入参为text，第二个入参为datetime，返回值为text。 – 第一个入参为text，第二个入参为time，返回值为text。 – 第一个入参为datetime，第二个入参为date，返回值为datetime。 – 第一个入参为datetime，第二个入参为text，返回值为text。 – 第一个入参为datetime，第二个入参为datetime，返回值为datetime。 – 第一个入参为datetime，第二个入参为time，返回值为datetime。 – 第一个入参为time，第二个入参为date，返回值为time。 – 第一个入参为time，第二个入参为text，返回值为text。 – 第一个入参为time，第二个入参为datetime，返回值为time。 – 第一个入参为time，第二个入参为time，返回值为time。
SYSDATE()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数回绕，GaussDB不回绕。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
YEAR()	支持	-
TIME_FORMAT()	支持	-
TIME_TO_SEC()	支持	-
TIMEDIFF()	支持	-
WEEKOFYEAR()	支持	-
TIMESTAMPADD()	支持	-
TIMESTAMPDIFF()	支持	-
TO_DAYS()	支持	-
TO_SECONDS()	支持	-
UNIX_TIMESTAMP()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是numeric，MySQL返回的是int。
UTC_DATE()	支持，存在差异	• MySQL支持无括号调用，GaussDB不支持。MySQL入参整型值会按照一字节最大值255整数回绕。 • MySQL入参只支持0-6整数， GaussDB支持可以隐式转换为0-6的输入。
UTC_TIME()	支持，存在差异	
UTC_TIMESTAMP()	支持，存在差异	
WEEK()	支持	-
WEEKDAY()	支持	-

版权所有 © 华为云计算技术有限公司

3.3.3.2.4 字符串函数

表 3-178 字符串函数列表

MySQL数据库	GaussDB数据库	差异
BIN()	支持，存在差异	函数入参支持类型存在差异，GaussDB入参支持类型如下： ● 整数类型：tinyint、smallint、mediumint、int、bigint ● 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned ● 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 ● 浮点类型：float、real、double ● 定点类型：numeric、decimal、dec ● 布尔类型：bool
CONCAT()	支持，存在差异	无论参数的数据类型如何，concat返回值的数据类型始终为text；MySQL的concat在含有二进制类型参数时，返回值为二进制类型。
CONCAT_WS()	支持，存在差异	无论参数的数据类型如何，concat_ws返回值的数据类型始终为text；MySQL的concat_ws在含有二进制类型参数时，返回值为二进制类型，其他情况返回值为字符串类型。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
ELT()	支持，存在差异	• 函数入参1支持类型存在差异，GaussDB入参1支持类型如下： – 整数类型：tinyint、smallint、mediumint、int、bigint – 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned – 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 – 浮点类型：float、real、double – 定点类型：numeric、decimal、dec – 布尔类型：bool • 函数入参2支持类型存在差异，GaussDB入参2支持类型如下： – 整数类型：tinyint、smallint、mediumint、int、bigint – 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned – 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext – 浮点类型：float、real、double – 定点类型：numeric、decimal、dec – 布尔类型：bool – 大对象类型：tinyblob、blob、mediumblob、longblob – 日期类型：datetime、timestamp、date、time
FIELD()	支持，存在差异	函数入参为在bigint最大值 ~ bigint unsigned最大值范围内的数字，存在不兼容。 函数入参为浮点型float(m, d)、double(m, d)、real(m, d)时精度更高，存在不兼容。
FIND_IN_SET()	支持，存在差异	当数据库encoding = 'SQL_ASCII'时，不支持默认的大小写判断规则，即在用户不指定字符集规则的情况下，大写与小写区分判断。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
INSERT()	支持，存在差异	• Int64类型传参有范围限制，一旦超出-9223372036854775808~9223372036854775807范围会直接报错，MySQL对数值类型传参范围无限制，异常会告警按照上限或下限数值处理。 • 字符串传参有限制，入参text类型字符串长度最大为2^30-5字节，入参BYTEA类型字符串长度最大为2^30-512字节。 • s1和s2任意参数为BYTEA类型时，涉及到结果出现非法字符的情况可能展示结果与MySQL有差异但是字符编码与MySQL是一致的。
LOCATE()	支持，存在差异	入参1为BYTEA类型，入参2为text类型时，GaussDB与MySQL行为存在差异。
MAKE_SET()	支持，存在差异	• bits参数为整型时，最大范围支持到int128，低于MySQL范围。 • bits参数为日期类型datetime、timestamp、date、time，由于时间类型转整型与MySQL存在差异，目前均未做支持。 • bit类型或bool类型由于此类数据类型GaussDB与MySQL存在差异，返回结果导致的差异为GaussDB与MySQL固有差异。bits入参为bool类型，str入参为bit类型与bool类型均不做支持。 • bits入参为字符串或文本类型时，仅支持纯整型数字形式，其他形式存在差异。且纯整型数字范围限制在bigint范围。 • str入参整型数值超过正负81个9，返回值与MySQL有差异。 • str入参当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，以科学计数法打印，此为固有差异。

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
QUOTE()	支持，存在差异	● 已知str字符串中含有“\Z”，“\r”，“\%”，“_”，GaussDB未进行转义，与MySQL存在差异。斜线后跟部分数字也会引起差异，如“\563”。由转义字符引起的本函数与MySQL的差异，此为GaussDB与MySQL的转义字符差异。 ● str字符串中的“\b”，输出结果表现形式与MySQL有差异。此为GaussDB与MySQL的固有差异 ● str字符串中含有“\0”时，GaussDB由于UTF-8字符集不识别该字符，输入不成功。此为GaussDB与MySQL的固有差异 ● str为bit或bool类型时，由于GaussDB与MySQL此类型目前有差异，暂不支持此类类型。 ● GaussDB最大支持1GB数据传输，str入参长度最大支持536870908字节，函数返回结果字符串最大支持1GB。 ● str入参整型数值超过正负81个9，返回值与MySQL有差异。 ● str入参当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，以科学计数法打印，此为固有差异。
SPACE()	支持，存在差异	● GaussDB入参最大支持1073741818字节，超出返回空字符串。MySQL的入参默认最大支持4194304字节，超出告警。 ● 函数入参支持类型存在差异，GaussDB入参支持类型如下： – 整数类型：tinyint、smallint、mediumint、int、bigint – 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned – 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 – 浮点类型：float、real、double – 定点类型：numeric、decimal、dec – 布尔类型：bool
SUBSTR()	支持	-
SUBSTRING()	支持	-
SUBSTRING_INDEX()	支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
STRCMP()	支持，存在差异	• 函数入参支持类型存在差异，GaussDB入参支持类型如下： – 字符类型：char、varchar、nvarchar2、text – 二进制类型：BYTEA – 数值类型：tinyint [unsigned]、smallint [unsigned]、integer [unsigned]、bigint [unsigned]、float4、float8、numeric – 日期时间类型：date、time without time zone、datetime、timestamptz • 对于数值类型中的浮点类型，由于连接参数设置不同，精度可能与M有差异，不建议使用该场景，或使用NUMERIC类型代替。
SHA()/ SHA1()	支持	-
SHA2()	支持	-

3.3.3.2.5 强制转换函数

表 3-179 强制转换函数列表

MySQL数据库	GaussDB数据库	差异
CAST()	支持，存在差异	数据类型转换规则和支持的转换类型均以GaussDB支持的转换范围和规则为准。
CONVERT()	支持，存在差异	数据类型转换规则和支持的转换类型均以GaussDB支持的转换范围和规则为准。

3.3.3.2.6 加密函数

表 3-180 加密函数列表

MySQL数据库	GaussDB数据库	差异
AES_DECRYPT()	支持	-
AES_ENCRYPT()	支持	-

版权所有 © 华为云计算技术有限公司

3.3.3.2.7 JSON 函数

JSON函数差异说明：

- 对于JSON函数和其他字符入参函数，如果输入中包含转义字符，默认情况下会与MySQL有一定差异。要实现与MySQL的兼容，需要设置GUC参数standard_conforming_strings取值为off，在这种情况下，转义字符的处理将与MySQL兼容，但是会产生非标准字符输入的warning告警，转义字符\t、\u以及转义数字与MySQL存在差异。JSON_UNQUOTE()函数已经做了兼容处理，即使不设置 GUC 参数，也能与 MySQL 兼容，并且不会产生告警。
- 在处理超长数字（数字的字符长度超过64）时，GaussDB的JSON函数会将数字解析为一个double处理，并使用科学计数法计数。和MySQL的非JSON类型入参相同。但是在JSON类型入参时，由于JSON类型未完全与MySQL兼容，此场景下会产生差异。MySQL会完整显示数字（并且当数字长度超过82时，MySQL会给出错误的结果），GaussDB依然将超长数字解析为一个double精度的值。考虑到超长数字内部都是使用浮点数进行储存，进行运算时无论GaussDB还是MySQL都会有精度丢失，建议您使用字符串来储存超长数字。

```
gaussdb=# SELECT json_insert('[1, 4,
99999999999999999999999999999999999999999999999999999999999999999999999999999999999]','$[6]',json
_insert('[1,4]','$[5]',99999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999));
        json_insert
-------------------------------
 [1, 4, 1e+74, [1, 4, 1e+74]]
(1 row)
```

表 3-181 JSON 函数列表

MySQL数据库	GaussDB数据库	差异
JSON_APPEND()	支持	-
JSON_ARRAY()	支持	-
JSON_ARRAY_APPEND()	支持	-
JSON_ARRAY_INSERT()	支持	-
JSON_CONTAINS()	支持	-
JSON_CONTAINS_PATH()	支持	-
JSON_DEPTH()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是int，MySQL返回的是bigint。
JSON_EXTRACT()	支持	-
JSON_INSERT()	支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
JSON_KEYS()	支持	-
JSON_LENGTH()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是int，MySQL返回的是bigint。
JSON_MERGE()	支持	-
JSON_OBJECT()	支持	-
JSON_QUOTE()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是JSON，MySQL返回的是varchar或者text。
JSON_REMOVE()	支持	-
JSON_REPLACE()	支持	-
JSON_SEARCH()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是text，MySQL返回的是JSON。
JSON_SET()	支持	-
JSON_TYPE()	支持，存在差异	数值类型的JSON值统一识别为number，与MySQL有差异。
JSON_UNQUOTE()	支持	-
JSON_VALID()	支持	-

版权所有 © 华为云计算技术有限公司

3.3.3.2.8 聚合函数

表 3-182 聚合函数列表

MySQL数据库	GaussDB数据库	差异
GROUP_CONCAT()	支持，存在差异	• 当group_concat参数中同时有DISTINCT和ORDER BY语法时，所有ORDER BY后的表达式必须也在DISTINCT的表达式之中。 • group_concat(... order by 数字)不代表按照第几个参数的顺序，数字只是一个常量表达式，相当于不排序。 • 无论入参的数据类型是什么，group_concat返回值的数据类型始终为text；MySQL的group_concat在含有二进制类型参数时，返回值为二进制类型，其他情况返回值为字符串类型，并且返回值长度大于512时，其数据类型为字符串大对象或二进制大对象。 • GUC参数group_concat_max_len有效范围是0-1073741823，最大值比MySQL小。
DEFAULT()	支持，存在差异	• 字段默认值为数组形式，GaussDB返回数组形式，MySQL不支持数组类型。 • GaussDB字段是隐藏列（比如xmin、cmin），default函数返回空值。 • GaussDB支持分区表、临时表、多表连接查询默认值。 • GaussDB支持查询列名包含字符串值节点（表示名称）和A_Star节点（表示出现“*”），如default(tt.t4.id)和default(tt.t4.*)。不合法的查询列名和A_Star节点，GaussDB和MySQL报错信息有差异。 • GaussDB创建字段默认值，没有检验字段类型的范围，使用default函数可能报错。 • 字段的默认值是函数表达式时，GaussDB的default函数返回建表时字段的default表达式的计算值。MySQL的default函数返回NULL。

版权所有 © 华为云计算技术有限公司

3.3.3.2.9 数字操作函数

表 3-183 数字操作函数列表

MySQL数据库	GaussDB数据库	差异
log2()	支持，存在差异	• 小数位显示与MySQL存在差异，受GaussDB浮点数据类型限制，可通过参数extra_float_digits控制小数位个数显示。 • 由于输入精度内部处理差异，GaussDB与MySQL会存在结果计算差异。 • 支持数据类型有： – 整数类型：bigint、int16、int、smallint 、tinyint。 – 无符号整数类型：bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned。 – 浮点数类型：numeric、real。 – 字符串类型：character、character varying、clob、text，仅支持纯数字整数字符串。 – set类型。 – NULL空类型。
log10()	支持，存在差异	• 小数位显示与MySQL存在差异，受GaussDB浮点数据类型限制，可通过参数extra_float_digits控制小数位个数显示； • 由于输入精度内部处理差异，GaussDB与MySQL会存在结果计算差异； • 支持数据类型有： – 整数类型：bigint、int16、int、smallint 、tinyint。 – 无符号整数类型：bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned。 – 浮点数类型：numeric、real。 – 字符串类型：character、character varying、clob、text，仅支持纯数字整数字符串。 – set类型。 – NULL空类型。
RAND([seed])	支持，存在差异	受函数内部使用的随机数生成算法的影响，函数返回的随机数与MYSQL存在差异。

版权所有 © 华为云计算技术有限公司

3.3.3.2.10 其他函数

表 3-184 其他函数列表

MySQL数据库	GaussDB数据库	差异
UUID()	支持	-
UUID_SHORT()	支持	-

3.3.3.3 操作符

GaussDB数据库兼容绝大多数MySQL的操作符，但存在部分差异。除特别说明外，MySQL兼容性MYSQL模式中的操作符行为默认为GaussDB原生行为。

表 3-185 操作符

MySQL数据库	GaussDB数据库	差异
安全等于（<=>）	支持	-

版权所有 © 华为云计算技术有限公司

MySQL数据库	GaussDB数据库	差异
[NOT] REGEXP	支持，存在差异	• 当设置GUC参数b_format_dev_version='s2'时，模式字符串pattern中有“\\a”、“\\d”、“\\e”、“\\n”、“\\Z”、“\\u”等转义字符时，匹配源字符串“\a”、“\d”、“\e”、“\n”、“\Z”、“\u”时，GaussDB行为与MySQL 5.7不一致，与MySQL 8.0一致。 • 当设置GUC参数b_format_dev_version='s2'时，GaussDB中“\b”可以与“\\b”匹配，MySQL会匹配失败。 • 模式字符串pattern非法入参，只存在右单括号“)”时，GaussDB会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 • 在de|abc匹配序列de或abc的匹配规则，当|左右存在空值时，GaussDB不会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 • 制表符“\t”正则匹配字符类[:blank:]，GaussDB可匹配，MySQL 5.7不能匹配，MySQL 8.0可匹配。 • GaussDB支持非贪婪模式匹配，即尽可能少的匹配字符，在部分特殊字符后加“?”问号字符，例如：“??, *?, +?, {n}?, {n,}?, {n,m}?”。MySQL 5.7版本不支持非贪婪模式匹配，并报错：Got error 'repetition-operator operand invalid' from regexp。MySQL 8.0版本已经支持。 • 在BINARY字符集下，TEXT类型、BLOB类型均会转换成BYTEA类型，由于REGEXP操作符不支持BYTEA类型，因此无法匹配。
[NOT] RLIKE	支持，存在差异	同[NOT] REGEXP。

3.3.3.4 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集，支持的范围如下。

表 3-186 字符集列表

MySQL数据库	GaussDB数据库
utf8mb4	支持
gbk	支持
gb18030	支持

版权所有 © 华为云计算技术有限公司

说明

目前GaussDB对不属于当前字符集的非法字符未执行严格的编码逻辑校验，可能导致此类非法字符成功输入。而MySQL会校验报错。

3.3.3.5 排序规则

GaussDB数据库支持指定模式、表或列的排序规则，支持的范围如下。

表 3-187 排序规则列表

MySQL数据库	GaussDB数据库
binary	支持
gb18030_bin	支持
gb18030_chinese_ci	支持
gbk_bin	支持
gbk_chinese_ci	支持
utf8_bin	支持
utf8_general_ci	支持
utf8mb4_0900_ai_ci	支持
utf8mb4_bin	支持
utf8mb4_general_ci	支持
utf8mb4_unicode_ci	支持

说明

排序规则差异说明：

- 当前仅有字符串类型、部分二进制类型支持指定排序规则，其他类型不支持指定排序规则，可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。MySQL中所有类型可以指定字符序，但除字符串、二进制类型其他排序规则无实际意义。
- 当前排序规则仅支持在其对应字符集与库级字符集一致时可以指定。
- utf8mb4字符集下默认字符序为utf8mb4_general_ci，与MySQL 5.7保持一致，utf8mb4_0900_ai_ci为utf8mb4的非默认字符序。
- GaussDB中utf8和utf8mb4为同一个字符集。

3.3.3.6 SQL

版权所有 © 华为云计算技术有限公司

3.3.3.6.1 DDL

表 3-188 DDL 语法兼容介绍

概述	详细语法说明	差异
建表和修改表时支持创建主键、UNIQUE索引	ALTER TABLE、CREATE TABLE	• GaussDB当前不支持UNIQUE INDEX|KEY index_name语法，使用UNIQUE INDEX|KEY index_name语法时会报错。MySQL支持。 • 当约束被建立为全局二级索引，SQL语句中指定using btree时，底层会建立为ubtree。 • 当约束关联的表为ustore，且SQL语句中指定为using btree时，底层会建立为ubtree。
支持前缀索引	CREATE INDEX	• 前缀长度不得超过2676，键值的实际长度受内部页面限制，若字段中含有多字节字符或者一个索引上有多个键，索引行长度可能会超限报错。 • CREATE INDEX语法中，不支持以下关键字作为前缀键的字段名称：COALESCE、EXTRACT、GREATEST、LEAST、LNNVL、NULLIF、NVL、NVL2、OVERLAY、POSITION、REGEXP_LIKE、SUBSTRING、TIMESTAMPDIFF、TREAT、TRIM、XMLCONCAT、XMLELEMENT、XMLEXISTS、XMLFOREST、XMLPARSE、XMLPI、XMLROOT、XMLSERIALIZE。 • 主键和唯一键索引中不支持前缀键。
支持指定字符集与排序规则	ALTER SCHEMA、ALTER TABLE、CREATE SCHEMA、CREATE TABLE	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
创建分区表语法兼容	CREATE TABLE PARTITION	-
建表和修改表时支持指定表级和列级comment	CREATE TABLE、ALTER TABLE	-
创建索引时支持指定索引级comment	CREATE INDEX	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
交换普通表和分区表分区的数据	ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点： ● MySQL表或分区使用tablespace时，则无法进行分区和普通表数据的交换；GaussDB表或分区使用不同的tablespace时，仍可进行分区和普通表数据的交换。 ● 对于列默认值，MySQL不会校验默认值，因此默认值不同时也可进行分区和普通表数据的交换；GaussDB会校验默认值，如果默认值不同，则无法进行分区和普通表数据的交换。 ● MySQL在分区表或普通表上进行DROP列操作后，表结构仍然一致，则可进行分区和普通表数据的交换；GaussDB需要保证普通表和分区表的被删除列严格对齐才能进行分区和普通表数据的交换。 ● MySQL和GaussDB的哈希算法不同，所以两者在相同的hash分区存储的数据可能不一致，导致最后交换的数据也可能不一致。 ● MySQL的分区表不支持外键，普通表包含外键或其他表引用普通表的外键，则无法进行分区和普通表数据的交换；GaussDB的分区表支持外键，在两个表的外键约束一致时，则可进行分区和普通表数据的交换；GaussDB的分区表不带外键，普通表有其他表引用，如果分区表和普通表表一致，则可进行分区和普通表数据的交换。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持自增列	ALTER TABLE、CREATE TABLE	• 目前仅支持各个DN独立自增的本地自动增长列。 • 自动增长列建议为索引（非全局二级索引）的第一个字段，否则建表时产生警告，含有自动增长列的表进行某些操作时会产生错误，例如：ALTER TABLE EXCHANGE PARTITION。MySQL自动增长列必须为索引第一个字段。 • AUTO_INCREMENT = value语法，value必须为小于2^127的正数。MySQL不校验value。 • 当自增值已经达到字段数据类型的最大值时，继续自增将产生错误。MySQL有些场景产生错误或警告，有些场景仍自增为最大值。 • 不支持innodb_autoinc_lock_mode系统变量，GaussDB的GUC参数auto_increment_cache=0时，批量插入自动增长列的行为与MySQL系统变量innodb_autoinc_lock_mode=1相似。 • 自动增长列在导入数据或者进行Batch Insert执行计划的插入操作时，对于混合0、NULL和确定值的场景，如果产生错误，后续插入自增值不一定与MySQL完全一致。提供auto_increment_cache参数，可以控制预留自增值的数量。 • 批量插入在不同执行计划下，自增的顺序、自增值预留数量可能与MySQL不完全相同。例如：“INSERT INTO table VALUES(...),(...),...”由于要分布到不同DN，在某些执行计划中，DN获取不到即将插入的行数。提供

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		auto_increment_cache参数，可以控制预留自增值的数量。 ● 并行导入或插入自动增长列触发自增时，每个并行线程预留的缓存值也只在其线程中使用，未完全使用完毕的话，也会出现表中自动增长列的值不连续的情况。并行插入产生的自增值结果无法保证与MySQL完全一致。 ● SERIAL数据类型为原有的自增列，与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。 ● 不允许auto_increment_offset的值大于auto_increment_increment的值，会产生错误。MySQL允许，并说明auto_increment_offset会被忽略。 ● 在表有主键或索引的情况下，ALTER TABLE命令重写表数据的顺序与MySQL不一定相同，GaussDB按表数据存储顺序重写，MySQL会按主键或索引顺序重写，导致自增值的顺序可能不同。 ● ALTER TABLE命令添加或修改自增列时，第一次预留自增值的数量是表统计信息中的行数，统计信息的行数不一定与MySQL一致。 ● 在触发器或用户自定义函数中自增时，刷新last_insert_id返回值。MySQL不刷新。 ● 对GUC参数auto_increment_offset和auto_increment_increment设置超出范围的值会产生错

 版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		误。MySQL会自动改为边界值。 ● 暂不支持last_insert_id函数。 ● 本地临时表暂不支持自动增长列。 ● sql_mode设置no_auto_value_on_zero参数，表定义的自动增长列为非NOT NULL约束，向表中插入数据不指定自动增长列的值时，GaussDB中自动增长列插入NULL值，且不触发自增；MySQL中自动增长列插入NULL值，触发自增。
支持删除表的主键约束	ALTER TABLE	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持CREATE TABLE ... LIKE语法兼容	CREATE TABLE ... LIKE	• 在MySQL 8.0.16 之前的版本中，CHECK约束会被语法解析但功能会被忽略，表现为不复制CHECK约束，GaussDB支持复制CHECK约束。 • 对于set数据类型，在建表时，MySQL支持复制，GaussDB不支持复制。 • 对于主键约束名称，在建表时，MySQL所有主键约束名称固定为PRIMARY KEY，GaussDB不支持复制。 • 对于唯一键约束名称，在建表时，MySQL支持复制，GaussDB不支持复制。 • 对于CHECK约束名称，在建表时，MySQL 8.0.16 之前的版本无CHECK约束信息，GaussDB支持复制。 • 对于索引名称，在建表时，MySQL支持复制，GaussDB不支持复制。 • 在跨sql_mode模式建表时，MySQL受宽松模式和严格模式控制，GaussDB可能存在严格模式失效的情况。 例如：源表存在默认值“0000-00-00”，在“no_zero_date”严格模式下，GaussDB建表成功，且包含默认值“0000-00-00”，严格模式失效；而MySQL建表失败，受严格模式控制。 • 针对跨数据库创建表，MySQL支持，GaussDB不支持。 • 针对源表为临时表，创建非临时表，MySQL支持，GaussDB不支持。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持更改表名兼容语法	ALTER TABLE[IF EXISTS] tbl_name RENAME [TO | AS | =] new_tbl_name; RENAME {TABLE | TABLES} tbl_name TO new_tbl_name [, tbl_name2 TO new_tbl_name2, ...];	• GaussDB的alter rename语法仅支持修改表名称功能操作，不能耦合其它功能操作。 • GaussDB中，仅旧表名字段支持使用schema.table_name格式，且新表名与旧表名将属于同一个Schema。 • GaussDB的不支持新旧表跨Schema重命名操作；但如有权限，则可在当前Schema下修改其它Schema下表名称。 • GaussDB的rename多组表的语法支持全为本地临时表的重命名，不支持本地临时表和非本地临时表组合的场景。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
支持增加子分区语法兼容	ALTER TABLE [IF EXISTS] { table_name [*] | ONLY table_name | ONLY (table_name)} action [, ...]; action: move_clause | exchange_clause | row_clause | merge_clause | modify_clause | split_clause | add_clause | drop_clause | ilm_clause add_clause: ADD {{partition_less_than_item | partition_start_end_item | partition_list_item} | PARTITION({partition_less_than_item | partition_start_end_item | partition_list_item})}	• 不支持ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition1,…);语法添加多分区。 • 仅支持原有添加多分区语法：ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2[y1]), …;。

3.3.3.6.2 DML

表 3-189 DML 语法兼容介绍

概述	详细语法说明	差异
DELETE支持ORDER BY和LIMIT	DELETE	-
UPDATE支持ORDER BY和LIMIT	UPDATE	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

REPLACE INTO语法兼容	REPLACE	• 时间类型初始值的差异。例如:
		– MySQL不受严格模式和宽松模式的影响，可向表中插入时间0值，即:
		mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL);
		Query OK, 1 row affected (0.00 sec)
		
		mysql> REPLACE INTO test VALUES(f1, f2, f3);
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM test;
		+---------------------+---------------------+------------+
		| f1 | f2 | f3 |
		+---------------------+---------------------+------------+
		| 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 |
		+---------------------+---------------------+------------+
		1 row in set (0.00 sec)
		– GaussDB在宽松模式下才可以成功插入时间0值，即
		gaussdb=# SET b_format_version = '5.7';
		SET
		gaussdb=# SET b_format_dev_version = 's1';
		SET
		gaussdb=# SET sql_mode = '';
		SET
		gaussdb=# CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL) DISTRIBUTE BY HASH(f1);
		CREATE TABLE
		gaussdb=# REPLACE INTO test VALUES(f1, f2, f3);
		REPLACE 0 1
		gaussdb=# SELECT * FROM test;
		f1 | f2 | f3
		--------------------- 	
		+---------------------+------------
		0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00
		(1 row)

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		在严格模式下，则报错date/time field value out of range: "0000-00-00 00:00:00"。
		● BIT类型在有NOT NULL时初始值的差异。例如：
		– MySQL BIT类型的初始值为空串''，即：
		mysql> CREATE TABLE test(f1 BIT(3) NOT NULL);
		Query OK, 0 rows affected (0.01 sec)
		
		mysql> REPLACE INTO test VALUES(f1);
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT f1, f1 IS NULL FROM test;
		+----+------------+
		| f1 | f1 is null |
		+----+------------+
		| | 0 |
		| | 0 |
		+----+------------+
		2 rows in set (0.00 sec)
		– GaussDB BIT类型的初始值为NULL，则报错。
		gaussdb=# CREATE TABLE test(f1 int, f2 BIT(3) NOT NULL) DISTRIBUTE BY HASH(f1);
		CREATE TABLE
		gaussdb=# REPLACE INTO test VALUES(1, f2);
		ERROR: null value in column "f2" violates not-null constraint
		DETAIL: Failing row contains (1, null).

SELECT支持指定多分区查询	SELECT	-
UPDATE支持指定多分区更新	UPDATE	-

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

LOAD DATA导入数据功能	LOAD DATA	• LOAD DATA语法执行结果与MySQL严格模式一致，宽松模式暂未适配。例如当导入源数据包含非法字符时，GaussDB报错，MySQL严格模式下报错，宽松模式下对非法字符进行截断处理后导入。
		-- GaussDB
		gaussdb=# SET sql_mode='';
		SET
		gaussdb=# DROP TABLE IF EXISTS t1;
		DROP TABLE
		gaussdb=# CREATE TABLE t1(a text) CHARSET gbk;
		CREATE TABLE
		gaussdb=# SELECT unhex('1AFF1F') INTO OUTFILE '/tmp/3.data';
		COPY 1
		gaussdb=# LOAD DATA INFILE '/tmp/3.data' INTO TABLE t1 CHARACTER SET gbk;
		ERROR: invalid byte sequence for encoding "GBK": 0xff 0x1f
		CONTEXT: COPY t1, line 1.
		gaussdb=# DROP TABLE t1;
		DROP TABLE
		
		-- MySQL
		mysql> SET sql_mode='';
		Query OK, 0 rows affected (0.00 sec)
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected (0.02 sec)
		mysql> CREATE TABLE t1(a text) CHARSET gbk;
		Query OK, 0 rows affected (0.03 sec)
		mysql> SELECT unhex('1AFF1F') INTO OUTFILE '/tmp/3.data';
		Query OK, 1 row affected (0.00 sec)
		mysql> LOAD DATA INFILE '/tmp/3.data' INTO TABLE t1 CHARACTER SET gbk;
		Query OK, 1 row affected, 1 warning (0.01 sec)
		Records: 1 Deleted: 0 Skipped: 0 Warnings: 1
		mysql> SHOW warnings;
		+---------+------+--+
		| Level | Code | Message |
		+---------+------+--+
		| Warning | 1366 | Incorrect string

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		value: '\xFF\x1F' for column 'a' at row 1 | +---------+------ +---------- ----------+ 1 row in set (0.00 sec) mysql> SELECT hex(a) FROM t1; +--------+ | hex(a) | +--------+ | 1A | +--------+ 1 row in set (0.00 sec) mysql> DROP TABLE t1; Query OK, 0 rows affected (0.02 sec) ● IGNORE与LOCAL参数功能仅为当导入数据与表中数据存在冲突时，忽略当前冲突行数据功能和当文件中字段数小于指定表中列数时自动为其余列填充默认值功能，其余功能暂未适配。 ● 指定LOCAL关键字，且文件路径为相对路径时，文件从二进制目录下搜索；不指定LOCAL关键字，且文件路径为相对路径时，文件从数据目录下搜索。 ● 语法中指定分隔符，转义字符，分行符等符号时，若指定为单引号，将导致词法解析错误。 ● [(col_name_or_user_var [, col_name_or_user_var] ...)]指定列参数不支持重复指定列。 ● [FIELDS TERMINATED BY 'string']指定换行符不能与[LINES TERMINATED BY 'string']分隔符相同。 ● 执行LOAD DATA语法写入表中的数据若无法转换为表中数据类型格式时报错。 ● 指定列时，仅支持指定为列名，不支持指定为用户变量。 ● LOAD DATA SET表达式中不支持指定列名计算。

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		• 若set表达式返回值类型与对应列类型之间不存在隐式转换函数则报错。
		• LOAD DATA不支持INSERT、DELETE触发器。
		• LOAD DATA只能用于表，不能用于视图。
		• Windows下的文件与Linux环境下文件默认换行符存在差异，LOAD DATA无法识别此场景会报错，建议用户导入时检查导入文件行结尾的换行符。
		• 当源数据的字段数比目标表的字段数多时，GaussDB提示告警信息，MySQL在以下任一场景提示告警信息，其他场景报错。
		– sql_mode未设置严格模式。
		– 指定了IGNORE选项。
		– 指定了LOCAL选项且未指定REPLACE选项。
		• 当源数据文件尾以列分隔符结束时，MySQL按分隔符后无新列处理，GaussDB下按分隔符后包含一列空字符串处理。
		-- 1.csv源数据
		1,2,
		
		-- GaussDB
		gaussdb=# DROP TABLE IF EXISTS t1;
		NOTICE: table "t1" does not exist, skipping
		DROP TABLE
		gaussdb=# CREATE TABLE t1(a char(30), b char(30), c char(30));
		CREATE TABLE
		gaussdb=# LOAD DATA infile '/tmp/1.csv' INTO TABLE t1 fields terminated by ',';
		COPY 1
		gaussdb=# SELECT c IS NULL, length(c) FROM t1;
		?column? | length
		----------+--------
		f | 0
		(1 row)
		
		-- MySQL
		mysql> DROP TABLE IF EXISTS t1;
		Query OK, 0 rows affected (0.06

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异
		sec) mysql> CREATE TABLE t1(a char(30), b char(30), c char(30)); Query OK, 0 rows affected (0.04 sec) mysql> LOAD DATA infile '/tmp/1.csv' INTO TABLE t1 fields terminated by ','; ERROR 1261 (01000): Row 1 doesn't contain data for all columns

 版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

INSERT IGNORE兼容	INSERT IGNORE	• GaussDB会返回降级后的错误信息，MySQL则会将降级后的错误信息记录到错误堆栈中，然后调用show warnings;命令查看。例如：
		• 时间类型的差异。例如：
		– GaussDB中date、datetime、timestamp默认零值。
		gaussdb=# CREATE TABLE test(f1 DATE NOT NULL, f2 DATETIME NOT NULL, f3 TIMESTAMP NOT NULL);
		CREATE TABLE
		gaussdb=# INSERT IGNORE INTO test VALUES(NULL, NULL, NULL);
		WARNING: null value in column "f1" violates not-null constraint
		DETAIL: Failing row contains (null, null, null, null).
		WARNING: null value in column "f2" violates not-null constraint
		DETAIL: Failing row contains (null, null, null, null).
		WARNING: null value in column "f3" violates not-null constraint
		DETAIL: Failing row contains (null, null, null, null).
		INSERT 0 1
		gaussdb=#
		SELECT * FROM test;
		f1 | f2 | f3
		------------+---------------------+---------------------
		1970-01-01 | 1970-01-01 00:00:00 | 1970-01-01 00:00:00
		(1 row)
		– MySQL中date、datetime、timestamp默认零值。
		mysql> CREATE TABLE test(f1 DATE NOT NULL, f2 DATETIME NOT NULL, f3 TIMESTAMP NOT NULL);
		Query OK, 0 rows affected (0.00 sec)
		
		mysql> INSERT IGNORE INTO test VALUES(NULL, NULL, NULL);
		Query OK, 1 row affected, 3 warnings (0.00 sec)

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		mysql> show warnings;
		+---------+------
		+----------------------------+
		| Level | Code |
		Message |
		+---------+------
		+----------------------------+
		| Warning | 1048 | Column 'f1'
		cannot be null |
		| Warning | 1048 | Column 'f2'
		cannot be null |
		| Warning | 1048 | Column 'f3'
		cannot be null |
		+---------+------
		+----------------------------+
		3 rows in set (0.00 sec)
		
		mysql> SELECT * FROM test;
		+------------+--------------------
		+--------------------+
		| f1 | f2 |
		f3 |
		+------------+--------------------
		+--------------------+
		| 0000-00-00 | 0000-00-00
		00:00:00 | 0000-00-00
		00:00:00 |
		+------------+--------------------
		+--------------------+
		1 row in set (0.00 sec)
		• 由于GaussDB不支持MySQL的bit类型，因此忽略bit类型NOT NULL约束和插入的bit类型长度与定义不同的场景下不支持INSERT IGNORE错误降级。
		– GaussDB中bit类型
		gaussdb=# CREATE TABLE
		test(f1 BIT(10) NOT NULL);
		CREATE TABLE
		gaussdb=# INSERT IGNORE
		INTO test VALUES(NULL);
		ERROR: Un-support feature
		DETAIL: ignore null for insert
		statement is not supported in
		column f1.
		gaussdb=# INSERT IGNORE
		INTO test VALUES('1010');
		ERROR: bit string length 4
		does not match type bit(10)
		CONTEXT: referenced column:
		f1
		– MySQL中bit类型
		mysql> CREATE TABLE test(f1
		BIT(10) NOT NULL);
		Query OK, 0 rows affected
		(0.00 sec)
		
		mysql> INSERT IGNORE INTO
		test VALUES(NULL);
		Query OK, 1 row affected, 1

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		warning (0.00 sec)
		
		mysql> INSERT IGNORE INTO test VALUES('1010');
		Query OK, 1 row affected, 1 warning (0.01 sec)
		● MySQL数据库时间类型指定精度时，插入时间零值会显示精度，GaussDB则不显示，例如：
		– GaussDB指定时间精度
		gaussdb=# CREATE TABLE test(f1 TIME(3) NOT NULL, f2 DATETIME(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL);
		CREATE TABLE
		gaussdb=# INSERT IGNORE INTO test VALUES(NULL,NULL,NULL);
		WARNING: null value in column "f1" violates not-null constraint
		DETAIL: Failing row contains (null, null, null).
		WARNING: null value in column "f2" violates not-null constraint
		DETAIL: Failing row contains (null, null, null).
		WARNING: null value in column "f3" violates not-null constraint
		DETAIL: Failing row contains (null, null, null).
		INSERT 0 1
		gaussdb=# SELECT * FROM test;
		f1 | f2 | f3
		----------+----------------------+----------------------
		00:00:00 | 1970-01-01 00:00:00 | 1970-01-01 00:00:00
		(1 row)
		– MySQL指定时间精度
		mysql> CREATE TABLE test(f1 TIME(3) NOT NULL, f2 DATETIME(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL);
		Query OK, 0 rows affected (0.00 sec)
		
		mysql> INSERT IGNORE INTO test VALUES(NULL,NULL,NULL);
		Query OK, 1 row affected, 3 warnings (0.00 sec)
		
		mysql> SELECT * FROM test;
		+--------------+-------------------------+-------------------------+

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

			f1	f2	f3 |
					+--------------
					+-------------------------
					+-------------------------+
					| 00:00:00.000
						0000-00-00 00:00:00.000	0000-00-00 00:00:00.000 |
							+--------------
							+-------------------------
							+-------------------------+
							1 row in set (0.00 sec)
							• 由于MySQL数据库和GaussDB执行过程的差异，因此，产生的warnings条数可能不同，例如：
							– GaussDB产生的warnings条数
							gaussdb=# CREATE TABLE test(f1 INT, f2 INT not null);
							CREATE TABLE
							gaussdb=# INSERT INTO test VALUES(1,0),(3,0),(5,0);
							INSERT 0 3
							gaussdb=# INSERT IGNORE INTO test SELECT f1+1, f1/f2 FROM test;
							WARNING: division by zero
							CONTEXT: referenced column: f2
							WARNING: null value in column "f2" violates not-null constraint
							DETAIL: Failing row contains (2, null).
							WARNING: division by zero
							CONTEXT: referenced column: f2
							WARNING: null value in column "f2" violates not-null constraint
							DETAIL: Failing row contains (4, null).
							WARNING: division by zero
							CONTEXT: referenced column: f2
							WARNING: null value in column "f2" violates not-null constraint
							DETAIL: Failing row contains (6, null).
							INSERT 0 3
							– MySQL产生的warnings条数
							mysql> CREATE TABLE test(f1 INT, f2 INT not null);
							Query OK, 0 rows affected (0.01 sec)
							
							mysql> INSERT INTO test VALUES(1,0),(3,0),(5,0);

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		Query OK, 3 rows affected (0.00 sec)
		Records: 3 Duplicates: 0 Warnings: 0
		
		mysql> INSERT IGNORE INTO test SELECT f1+1, f1/f2 FROM test;
		Query OK, 3 rows affected, 4 warnings (0.00 sec)
		Records: 3 Duplicates: 0 Warnings: 4
		• MySQL数据库和GaussDB INSERT IGNORE在触发器中的差异，例如：
		– GaussDB触发器中使用INSERT IGNORE
		gaussdb=# CREATE TABLE test1(f1 INT NOT NULL);
		CREATE TABLE
		gaussdb=# CREATE TABLE test2(f1 INT);
		CREATE TABLE
		gaussdb=# CREATE OR REPLACE FUNCTION trig_test() RETURNS TRIGGER AS $$
		gaussdb$# BEGIN
		gaussdb$# INSERT IGNORE INTO test1 VALUES(NULL);
		gaussdb$# RETURN NEW;
		gaussdb$# END;
		gaussdb$# $$ LANGUAGE plpgsql;
		CREATE FUNCTION
		gaussdb=# CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW EXECUTE PROCEDURE trig_test();
		CREATE TRIGGER
		gaussdb=# INSERT INTO test2 VALUES(NULL);
		WARNING: null value in column "f1" violates not-null constraint
		DETAIL: Failing row contains (null).
		CONTEXT: SQL statement "INSERT IGNORE INTO test1 VALUES(NULL)"
		PL/pgSQL function trig_test() line 3 at SQL statement
		INSERT 0 1
		gaussdb=# SELECT * FROM test1;
		f1
		---- 	
		0
		(1 rows)
		
		gaussdb=# SELECT * FROM test2;

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		f1
		---- 	
		
		(1 rows)
		– MySQL触发器中使用INSERT IGNORE
		mysql> CREATE TABLE test1(f1 INT NOT NULL);
		Query OK, 0 rows affected (0.01 sec)
		
		mysql> CREATE TABLE test2(f1 INT);
		Query OK, 0 rows affected (0.00 sec)
		
		mysql> DELIMITER ||
		mysql> CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW
		-> BEGIN
		-> INSERT IGNORE into test1 values(NULL);
		-> END||
		Query OK, 0 rows affected (0.01 sec)
		
		mysql> DELIMITER ;
		mysql> INSERT INTO test2 VALUES(NULL);
		ERROR 1048 (23000): Column 'f1' cannot be null
		mysql> INSERT IGNORE INTO test2 VALUES(NULL);
		Query OK, 1 row affected (0.00 sec)
		
		mysql> SELECT * FROM test1;
		+----+
		| f1 |
		+----+
		| 0 |
		+----+
		1 row in set (0.00 sec)
		
		mysql> SELECT * FROM test2;
		+------+
		| f1 |
		+------+
		| NULL |
		+------+
		1 row in set (0.00 sec)
		● GaussDB的bool、serial的实现机制与MySQL不同，因此其默认零值与MySQL不同，例如：
		– GaussDB的行为
		gaussdb=# CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL);
		NOTICE: CREATE TABLE will create implicit sequence

版权所有 © 华为云计算技术有限公司

概述	详细语法说明	差异

		"test_f1_seq" for serial column "test.f1"
		CREATE TABLE
		gaussdb=# INSERT IGNORE INTO test values(NULL,NULL);
		WARNING: null value in column "f1" violates not-null constraint
		DETAIL: Failing row contains (null, null).
		WARNING: null value in column "f2" violates not-null constraint
		DETAIL: Failing row contains (null, null).
		INSERT 0 1
		gaussdb=# SELECT * FROM test;
		 f1 | f2
		----+----
		  0 | f
		(1 row)
		– MySQL的行为
		mysql> CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL);
		Query OK, 0 rows affected (0.00 sec)
		
		mysql> INSERT IGNORE INTO test values(NULL,NULL);
		Query OK, 1 row affected, 1 warning (0.00 sec)
		
		mysql> SELECT * FROM test;
		+----+----+
		| f1 | f2 |
		+----+----+
		| 1 | 0 |
		+----+----+
		1 row in set (0.00 sec)

3.3.3.6.3 DCL

表 3-190 DCL 语法兼容介绍

概述	详细语法说明	差异
SET NAMES指定COLLATE子句	SET [SESSION | LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] | DEFAULT};	GaussDB中暂不支持指定charset_name与数据库字符集不同。具体请参见**SET** 。

3.3.3.7 驱动

版权所有 © 华为云计算技术有限公司

3.3.3.7.1 JDBC 接口参考

GaussDB与MySQL的JDBC接口定义一致，均遵循业界规范，本章节主要介绍GaussDB数据库的MySQL兼容性MYSQL模式与MySQL数据库JDBC接口的行为差异。

获取结果集中的数据

ResultSet对象提供了丰富的方法，以获取结果集中的数据。获取数据常用的方法如**表3-191**所示，其他方法请参考JDK官方文档。

表 3-191 ResultSet 对象的常用方法

方法	描述	差异
int getInt(int columnIndex)	按列标获取int型数据。	-
int getInt(String columnLabel)	按列名获取int型数据。	-
String getString(int columnIndex)	按列标获取String型数据。	字段类型为整型且带有ZEROFILL属性时，GaussDB按照ZEROFILL属性要求的宽度信息用0进行补位后输出结果，MySQL直接输出结果。
String getString(String columnLabel)	按列名获取String型数据。	字段类型为整型且带有ZEROFILL属性时，GaussDB按照ZEROFILL属性要求的宽度信息用0进行补位后输出结果，MySQL直接输出结果。
Date getDate(int columnIndex)	按列标获取Date型数据。	-
Date getDate(String columnLabel)	按列名获取Date型数据。	-

版权所有 © 华为云计算技术有限公司

4 工具参考

4.1 下载并安装工具

下载并安装工具

1. 下载表4-1中的驱动包和驱动包校验包。

   表 4-1 驱动包下载列表

版本	下载地址
V2.0-10.x	驱动包 驱动包校验包

2. 校验驱动包。

   为了防止软件包在传递过程或存储期间被恶意篡改，下载软件包时需下载对应的校验包对软件包进行校验，校验方法如下：

   - Linux操作系统软件包完整性校验：
     i. 上传软件包和软件包校验包到虚拟机的同一目录下。
     ii. 执行如下命令，校验软件包完整性。

     ```
     F=$(echo DBS-GaussDB-driver*.tar.gz);echo "$(awk '{print $1}' "$F.sha256")  $F" | sha256sum -c -
     ```

     如果回显OK，则校验通过。例如：

     ```
     DBS-GaussDB-driver_507.0_20260629130806.133858239.tar.gz: OK
     ```

   - Windows操作系统软件包完整性校验：
     i. 使用快捷键“Win+R”打开“运行”窗口。 在“打开”栏，输入“cmd”，按“Enter”回车，打开命令行页面。
     ii. 执行以下命令，获取驱动包的Hash值。

     certutil -hashfile *{驱动包本地目录}\{驱动包名}* sha256

     - {*驱动包本地目录*}：请根据实际下载目录进行替换。例如：C:\Users
     - {*驱动包名*}：请根据实际下载的驱动包名进行替换。例如：DBS-GaussDB-driver_507.0_20260629130806.133858239.tar.gz

版权所有 © 华为云计算技术有限公司

示例：certutil -hashfile *C:|Users|DBS-GaussDB-driver_507.0_20260629130806.133858239.tar.gz* sha256

iii. 将2.ii获取到的Hash值和表4-1中获取到的驱动包校验包的Hash值进行比较。
   - 若一致则通过校验。
   - 若不一致，请重新下载驱动包，重复2.i~2.iii进行校验。

3. 登录服务器，创建用于存放工具包的目录。
```
mkdir -p /opt/bin
```
4. 将1中获取到的驱动包上传到新创建的目录中。
5. 执行以下命令解压驱动包。驱动包名请根据实际下载的包名进行替换，此处以DBS-GaussDB-driver_507.0_20260629130806.133858239.tar.gz为例。
```
cd /opt/bin
tar -zxvf DBS-GaussDB-driver_507.0_20260629130806.133858239.tar.gz
```
6. 执行以下命令获取gsql客户端工具包。
```
cd /opt/bin/DBS-GaussDB-driver_507.0_20260629130806.133858239/Distributed/
tar -zxvf gsql_driver.tar.gz
cd /opt/bin/DBS-GaussDB-driver_507.0_20260629130806.133858239/Distributed/gsql_driver/Euler2.5_X86_64
tar -zxvf GaussDB-Kernel_507.0.0.B071_Gsql_X86_Distributed.tar.gz
```

说明

- 客户端工具包相对位置为解压后位置，请根据实际情况填写。此处以Euler2.5_x86_64系统的工具包为例。
- 客户端工具包名请根据实际下载的包名进行替换。此处以GaussDB-Kernel_507.0.0.B071_Gsql_X86_Distributed.tar.gz为例。

7. 执行以下命令将解压出来的“GaussDB-Kernel_507.0.0_Euler_64bit_Gsql.tar.gz”客户端工具包拷贝到“/opt/bin”路径下。
```
cp GaussDB-Kernel_507.0.0_Euler_64bit_Gsql.tar.gz /opt/bin
```

说明

Gsql工具包中包含了gsql客户端工具、gs_dump、gs_dumpall和gs_restore工具。

8. 执行以下命令解压文件。
```
cd /opt/bin
tar -zxvf GaussDB-Kernel_507.0.0_Euler_64bit_Gsql.tar.gz
```
9. 通过脚本，自动配置环境变量。
```
source gsql_env.sh
```

4.2 数据库连接工具

4.2.1 gsql 连接数据库

4.2.1.1 gsql 使用指导

工具介绍

gsql是GaussDB提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，gsql还提供了若干高级功能，便于用户使用。

基本功能：

版权所有 © 华为云计算技术有限公司

- 连接数据库：默认只支持从服务器本机连接，如果需要连接到远端的数据库，必须在服务端进行配置。详细操作请参见《开发指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节。

  说明

  gsql创建连接时，会有5分钟超时时间。如果在这个时间内，数据库未正确地接受连接并对身份进行认证，gsql将超时退出。针对此问题，可以参考**FAQ**。

- 执行SQL语句：支持交互式地键入并执行SQL语句，也可以执行一个文件中指定的SQL语句。
- 执行元命令：元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化SQL输出结果，以及连接到新的数据库等。元命令的详细说明请参见**元命令参考**。

高级功能：

gsql的高级功能如**表4-2**所示。

表 4-2 gsql 高级功能

特性名称	描述
变量	gsql提供类似于Linux的shell命令的变量特性，可以使用gsql的元命令\set设置一个变量，格式如下： \set *varname value* 删除由\set命令设置的变量请使用如下方式： \unset *varname* **说明** • 变量只是简单的名称/值对，这里的值可以是任意长度。 • 变量名称必须由字母（包括非拉丁字母）、数字和下划线组成，且对大小写敏感。 • 如果使用\set varname的格式（不带第二个参数），则只是设置这个变量而没有给变量赋值。 • 可以使用不带参数的\set来显示所有变量的值。 变量的示例和详细说明请参见**变量**。
SQL代换	利用gsql的变量特性，可以将常用的SQL语句设置为变量，以简化操作。 SQL代换的示例和详细说明请参见**SQL代换**。
自定义提示符	gsql使用的提示符支持用户自定义。可以通过修改gsql预留的三个变量PROMPT1、PROMPT2、PROMPT3来改变提示符。 这三个变量的值可以用户自定义，也可以使用gsql预定义的值。详细说明请参见**提示符**。

版权所有 © 华为云计算技术有限公司

特性名称	描述
客户端操作历史记录	gsql支持客户端操作历史记录，当客户端连接时指定“-r”参数，此功能被打开。可以通过\set设置记录历史的条数，例如，\set HISTSIZE 50，将记录历史的条数设置为50，\set HISTSIZE 0，不记录历史。 **说明** • 客户端操作历史记录条数默认设置为32条，最多支持记录500条。当客户端交互式输入包含中文字符时，只支持UTF-8的编码环境。 • 出于安全考虑，将包含PASSWORD、IDENTIFIED、GS_ENCRYPT_AES128、GS_DECRYPT_AES128、GS_ENCRYPT、GS_DECRYPT、GS_ENCRYPT_BYTEA、GS_DECRYPT_BYTEA、PG_CREATE_PHYSICAL_REPLICATION_SLOT_EXTERN、SECRET_ACCESS_KEY、SECRETKEY、CREATE_CREDENTIAL、ACCESSKEY、SECRET_KEY、DESENCRYPT、DESDECRYPT、DES3ENCRYPT、DES3DECRYPT等字符串（不区分大小写）的SQL语句记录识别为包含敏感信息的语句，不会记录到历史信息中，即不能通过上下翻回显。

- 变量

  可以使用gsql元命令\set设置一个变量。例如，把变量foo的值设置为bar：

  ```
  gaussdb=#  \set foo bar
  ```

  要引用变量的值，在变量前面加冒号。例如，查看变量的值：

  ```
  gaussdb=#  \echo :foo
  bar
  ```

  这种变量的引用方法适用于规则的SQL语句和除\copy、\ef、\help、\sf、\!以外的元命令。

  gsql预定义了一些特殊变量，同时也规划了变量的取值。为了保证和后续版本最大限度兼容，请避免以其他目的使用这些变量。所有特殊变量见表4-3。

  说明

  - 所有特殊变量都由大写字母、数字和下划线组成。
  - 要查看特殊变量的默认值，请使用元命令\echo :*varname*（例如\echo :DBNAME）。

  表 4-3 特殊变量设置

变量	设置方法	变量说明
DBNAME	\set DBNAME *dbname*	当前连接的数据库的名称。每次连接数据库时都会被重新设置。
ECHO	\set ECHO all | queries	• 如果设置为all，只显示查询信息。等效于使用gsql连接数据库时指定-a参数。 • 如果设置为queries，显示命令行和查询信息。等效于使用gsql连接数据库时指定-e参数。

版权所有 © 华为云计算技术有限公司

变量	设置方法	变量说明
ECHO_HIDDEN	\set ECHO_HIDDEN on | off | noexec	当使用元命令查询数据库信息（例如\dg）时，此变量的取值决定了查询的行为： ● 设置为on，先显示元命令实际调用的查询语句，然后显示查询结果。等效于使用gsql连接数据库时指定-E参数。 ● 设置为off，则只显示查询结果。 ● 设置为noexec，则只显示查询信息，不执行查询操作。
ENCODING	\set ENCODING *encoding*	当前客户端的字符集编码。
FETCH_COUNT	\set FETCH_COUNT *variable*	● 如果该变量的值为大于0的整数，假设为n，则执行SELECT语句时，每次从结果集中取n行到缓存并显示到屏幕。 ● 如果不设置此变量，或设置的值小于等于0，则执行SELECT语句时，一次性把结果都取到缓存。 **说明** 设置合理的变量值，将减少内存使用量。一般来说，设为100到1000之间的值比较合理。
HISTCONTROL	\set HISTCONTROL ignorespace | ignoredups | ignoreboth | none	● ignorespace：以空格开始的行将不会写入历史列表。 ● ignoredups：与以前历史记录里匹配的行不会写入历史记录。 ● ignoreboth、none或者其他值：所有以交互模式读入的行都被保存到历史列表。 **说明** none表示不设置HISTCONTROL。
HISTFILE	\set HISTFILE *filename*	此文件用于存储历史命令列表。缺省值是~/.bash_history。
HISTSIZE	\set HISTSIZE *size*	保存在历史命令里命令的个数。缺省值是500。
HOST	\set HOST *hostname*	已连接的数据库主机名称。
IGNOREEOF	\set IGNOREEOF *variable*	● 若设置此变量为数值，假设为10，则在gsql中输入的前9次EOF字符（通常是Ctrl+C）都会被忽略，在第10次按Ctrl+C才能退出gsql程序。 ● 若设置此变量为非数值，则缺省为10。 ● 若删除此变量，则向交互的gsql会话发送一个EOF终止应用。
LASTOID	\set LASTOID *oid*	最后影响的oid值，即为从一条INSERT或lo_import命令返回的值。此变量只保证在下一条SQL语句的结果显示之前有效。

版权所有 © 华为云计算技术有限公司

变量	设置方法	变量说明
ON_ERROR_ROLLBACK	\set ON_ERROR_ROLLBACK on | interactive | off	• on：当一个事务块里的语句发生错误时，自动回滚到上一个命令之前的状态，并继续执行后续命令。该模式是通过在一个事务块的每个命令前隐含地创建一个SAVEPOINT来工作的，在发生错误时，自动回滚到之前创建的SAVEPOINT，从而撤销失败命令的影响。 • interactive：仅在交互式会话中启用自动回滚功能。 • off（缺省值）：事务块里的语句发生错误时，终止整个事务。
ON_ERROR_STOP	\set ON_ERROR_STOP on | off	• on：命令执行错误时会立即停止，在交互模式下，gsql会立即返回已执行命令的结果。 • off（缺省值）：命令执行错误时将会跳过错误继续执行。
PORT	\set PORT *port*	当前连接数据库的端口号。
USER	\set USER *username*	当前用于连接的数据库用户。
VERBOSITY	\set VERBOSITY terse | default | verbose	该选项用于控制错误报告的详细程度。 • terse：仅返回严重且主要的错误文本以及文本位置（一般适合于单行错误信息）。 • default：返回严重且主要的错误文本及其位置，还包括详细的错误细节、错误提示（可能会跨越多行）。 • verbose：返回所有的错误信息。

- SQL代换

  像元命令的参数一样，gsql变量的一个关键特性是可以把gsql变量替换成正规的SQL语句。此外，gsql还提供为变量更换新的别名或其他标识符等功能。使用SQL代换方式替换一个变量的值可在变量前加冒号。例如：

```
gaussdb=#  \set foo 'HR.areaS'
-- 执行以下命令，将会查询HR.areaS表。
gaussdb=#  SELECT * FROM :foo;
 area_id |       area_name
---------+------------------------
       4 | Middle East and Africa
       3 | Asia
       1 | Europe
       2 | Americas
(4 rows)
```

  说明

  变量的值是逐字复制的，甚至可以包含不对称的引号或反斜杠命令。所以必须保证输入的内容有意义。

- 提示符

 版权所有 © 华为云计算技术有限公司

通过表4-4的三个变量可以设置gsql的提示符，这些变量由字符和特殊的转义字符所组成。

表 4-4 提示符变量

变量	描述	示例
PROMPT1	gsql请求一个新命令时使用的正常提示符。 PROMPT1的默认值为：%o%R%#。	使用变量PROMPT1切换提示符： • 提示符变为local： gaussdb=# \set PROMPT1 %M local:/data/huawei/wisequery/perfadm_mppdb • 提示符变为name： gaussdb=# \set PROMPT1 name name • 提示符变为=： gaussdb=# \set PROMPT1 %R =
PROMPT2	在一个命令期待更多输入时（例如，查询没有用一个分号结束或者引号不完整）显示的提示符。	使用变量PROMPT2显示提示符： gaussdb=# \set PROMPT2 TEST gaussdb=# SELECT * FROM HR.areaS TEST; area_id | area_name ---------+--------------------- 1 | Europe 2 | Americas 4 | Middle East and Africa 3 | Asia (4 rows)
PROMPT3	当执行COPY命令，并期望在终端输入数据时（例如，COPY FROM STDIN），显示提示符。	使用变量PROMPT3显示COPY提示符： gaussdb=# \set PROMPT3 '>>>>' gaussdb=# COPY HR.areaS FROM STDIN; Enter data to be copied followed by a newline. End with a backslash and a period on a line by itself. >>>>1 aa >>>>2 bb >>>>\.

提示符变量的值是按实际字符显示的，但是，当设置提示符的命令中出现“%”时，变量的值根据“%”后的字符，替换为已定义的内容，已定义的提示符请参见表4-5。

表 4-5 已定义的替换

符号	符号说明
%M	主机的全名（包含域名），若连接是通过Unix域套接字进行的，则全名为[local]，若Unix域套接字不是编译的缺省位置，则为[local:/dir/name]。
%m	主机名删去第一个点后面的部分。若通过Unix域套接字连接，则为[local]。
%o	数据库名称。如果是postgres库，显示gaussdb；其他库则显示数据库名。

版权所有 © 华为云计算技术有限公司

符号	符号说明
%>	主机正在侦听的端口号。
%n	数据库会话的用户名。
%/	当前数据库名称。
%~	类似%/，如果数据库是缺省数据库时输出的是波浪线“~”。
%#	如果会话用户是数据库系统管理员，使用“#”，否则用“>”。
%R	• 对于PROMPT1，通常是“=”，如果是单行模式则是“^”，如果会话与数据库断开（如果\connect失败可能发生）则是“!”。 • 对于PROMPT2，该序列被“-”、“*”、单引号、双引号或“$”（取决于gsql是否等待更多的输入：查询没有终止、正在一个/* ... */注释里、正在引号或者美元符扩展里）代替。
%x	事务状态： • 如果不在事务块里，则是一个空字符串。 • 如果在事务块里，则是“*”。 • 如果在一个失败的事务块里则是“!”。 • 如果无法判断事务状态时为“?”（比如没有连接）。
%digits	指定字节值的字符将被替换到该位置。
%:name	gsql变量“name”的值。
%`command`	“command”表示指定命令的输出，类似于普通的“反引号”替换。
%[. . . %]	提示可以包含终端控制字符，这些字符可以改变颜色、背景、提示文本的风格、终端窗口的标题。例如： `gaussdb=# \set PROMPT1 '%[%033[1;33;40m%]%n@%/%R%[%033[0m%]%#'` 这个句式的结果是在VT100兼容的可显示彩色的终端上的一个宽体（1;）黑底黄字（33;40）。

环境变量：

与gsql相关的环境变量参见**表4-6**。

表 4-6 与 gsql 相关的环境变量

名称	描述
COLUMNS	如果\set columns为0，则由此参数控制wrapped格式的宽度。这个宽度用于决定在自动扩展的模式下，是否要把宽输出模式变成竖线的格式。

版权所有 © 华为云计算技术有限公司

名称	描述
PAGER	如果查询结果无法在一页显示，它们就会被重定向到这个命令。可以用\pset命令关闭分页器。典型的是用命令more或less来实现逐页查看。缺省值与平台相关。 **说明** less的文本显示，受系统环境变量LC_CTYPE影响。
PSQL_EDITOR EDITOR VISUAL	\e和\ef命令使用环境变量指定的编辑器。变量是按照列出的先后顺序检查的。在Unix系统上默认的编辑工具是vi。

PSQL_EDITOR_LINENUMBER_ARG	当\e和\ef带上一行数字参数使用时，这个变量指定的命令行参数用于向编辑器传递起始行数。对于Emacs或vi编辑器，该参数值为一个加号。如果选项和行号之间需要空白，在变量的值后加一个空格。例如：
	`PSQL_EDITOR_LINENUMBER_ARG = '+'`
	`PSQL_EDITOR_LINENUMBER_ARG='--line '`
	Unix系统默认的是+。

PSQLRC	用户的.gsqlrc文件的存放位置。
SHELL	使用\!命令和shell执行的命令效果相同。
TMPDIR	存储临时文件的目录。缺省是/tmp。

说明

gsql不支持按照nls_date_format格式显示datea类型，当前只支持显示YYYY-MM-DD HH24-MI-SS AD格式。

前提条件

- 连接数据库时使用的用户需要具备访问数据库的权限。
- gsql需要与数据库版本配套。
- 当gsql客户端编码设置为UTF8时，若GaussDB返回的数据编码格式不是UTF8，则数据将无法正确显示在gsql客户端中。

说明

- gsql工具可通过发布包中的gsql_env.sh脚本设置环境变量使用，例如：source gsql_env.sh。
- 如果之前设置过环境变量，执行gsql_env.sh后会以脚本中环境变量优先。

命令格式

```
gsql [OPTION]... [DBNAME [USERNAME]]
```

参数说明

详细的gsql参数请参见表4-7、表4-8、表4-9、表4-10和表4-11。

版权所有 © 华为云计算技术有限公司

表 4-7 常用参数

参数	参数说明	取值范围
-c, --command=COMMAND	声明gsql要执行一条字符串命令然后退出。	-
-d, --dbname=DBNAME	指定连接的数据库名称。若未指定数据库名称，则使用初始化时默认生成的数据库名称。 另外，gsql允许使用扩展的DBNAME，即“postgres[ql]://[user[:password]@][netloc][:port][,...][/dbname][?param1=value1&...]”或“[key=value][...]”形式的连接串作为DBNAME，gsql将从连接串中解析连接信息，并优先使用这些信息。 **说明** gsql使用扩展的DBNAME创建连接时，不支持指定replication参数。	字符串。
-f, --file=FILENAME	使用文件作为命令源而不是交互式输入。gsql将在处理完文件后结束。如果FILENAME是-（连字符），则从标准输入读取。 **说明** 当服务端编码为GB18030/GB18030_2022时，若gsql -f 结合管道符('>') 执行（如：gsql -f xxx.sql > xxx.log），导入文件编码格式需与服务端编码格式保持一致，建议在文件中手动设置client_encoding为服务端编码格式。	绝对路径或相对路径，且满足操作系统路径命名规则。
-l, --list	列出所有可用的数据库，然后退出。	-
-v, --set=, --variable=NAME=VALUE	设置gsql变量NAME为VALUE。 变量的示例和详细说明请参见**变量**。	-
-X, --no-gsqlrc	不读取启动文件（系统范围的gsqlrc或者用户的~/.gsqlrc都不读取）。 **说明** 启动文件默认为~/.gsqlrc，或通过PSQLRC环境变量指定。	-
-1 ("one"), --single-transaction	当gsql使用-f选项执行脚本时，会在脚本的开头和结尾分别加上START TRANSACTION/COMMIT用以把整个脚本当作一个事务执行。这将保证该脚本完全执行成功，或者脚本无效。 **说明** 如果脚本中已经使用了START TRANSACTION、COMMIT、ROLLBACK，则该选项无效。	-

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
-y, --slash-command	将当前语句终结并发送到内核执行或者重新执行已经执行过的语句（不包括gsql元命令）。 **说明** • 该功能仅支持ORA兼容模式数据库，而且不支持-c, --command参数的使用场景。 • ORA兼容模式数据库下默认普通语句分隔符为“;”，PL/SQL为“/”，使用该参数时不支持更换分隔符。	-
-?, --help	显示关于gsql命令行参数的帮助信息然后退出。	-
-V, --version	打印gsql版本信息然后退出。	-

表 4-8 输入和输出参数

参数	参数说明	取值范围
-a, --echo-all	在读取行时向标准输出打印所有内容。 **说明** 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。	-
-B, --base64	表示-c参数传入的是Base64编码后的内容。 **说明** 此参数必须与-c参数同时使用。	-
-e, --echo-queries	把所有发送给服务器的查询同时回显到标准输出。 **说明** 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。	-
-E, --echo-hidden	回显由\d和其他反斜杠命令生成的实际查询。	-
-k, --with-key=KEY	使用gsql对导入的加密文件进行解密。 **说明** • 对于本身就是shell命令中的关键字符如单引号（'）或双引号（"），Linux shell会检测输入的单引号（'）或双引号（"）是否匹配。如果不匹配，shell认为用户没有输入完毕，会一直等待用户输入，从而不会进入到gsql程序。 • 不支持解密导入存储过程和函数。	-

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
-L, --log-file=FILENAME	除了正常的输出源之外，把所有查询输出记录到文件FILENAME中。 **说明** • 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。 • 此参数只保留查询结果到相应文件中，主要目标是为了查询结果能够更好更准确地被其他调用者（例如自动化运维脚本）解析，而不是保留gsql运行过程中的相关日志信息。	绝对路径或相对路径，且满足操作系统路径命名规则。
-m, --maintenance	允许在两阶段事务恢复期间连接集群。 **说明** 该选项是一个开发选项，禁止用户使用，只限专业技术人员使用，功能是：使用该选项时，gsql可以连接到备机，用于校验主备机数据的一致性。	-
-n, --no-libedit	关闭命令行编辑。	-
-o, --output=FILENAME	将所有查询输出重定向到文件FILENAME。	绝对路径或相对路径，且满足操作系统路径命名规则。
-q, --quiet	安静模式，执行时不会打印出额外信息。	缺省时gsql将打印许多其他输出信息。
-s, --single-step	单步模式运行。意味着每个查询在发往服务器之前都要提示用户，用这个选项也可以取消执行。此选项主要用于调试脚本。 **说明** 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。	-
-S, --single-line	单行运行模式，这时每个命令都将由换行符结束。	-
-C，-C1, --enable-client-encryption=1	可通过该选项打开密态数据库开关，此开关为密态等值查询基本能力开关。	-
-C3，--enable-client-encryption=3	可通过该选项打开内存解密逃生通道开关，支持密态等值查询基本能力以及内存解密逃生通道能力。	-

版权所有 © 华为云计算技术有限公司

表 4-9 输出格式参数

参数	参数说明	取值范围
-A, --no-align	切换为非对齐输出模式。缺省为对齐输出模式。	-
-F, --field-separator=STRING	设置域分隔符（默认为“|”）。	-
-H, --html	指定HTML格式输出。	-
-P, --pset=VAR[=ARG]	在命令行上以\pset的风格设置打印选项。 **说明** 这里必须用等号而不是空格分隔名称和值。例如，把输出格式设置为LaTeX，可以键入-P format=latex。	-
-R, --record-separator=STRING	设置记录分隔符。	-
-r	开启在客户端操作中可以进行编辑的模式。缺省为关闭。	-
-t, --tuples-only	只打印行。	-
-T, --table-attr=TEXT	允许声明放在HTML table标签里的选项。 使用时请搭配参数“-H,--html”，指定为HTML格式输出。	-
-x, --expanded	启用扩展格式显示。	-
-z, --field-separator-zero	设置非对齐输出模式的域分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	-
-0, --record-separator-zero	设置非对齐输出模式的记录分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	-
-2, --pipeline	使用管道传输密码，禁止在终端使用，必须和-c或者-f参数一起使用。	-

版权所有 © 华为云计算技术有限公司

表 4-10 连接参数

参数	参数说明	取值范围
-h, --host=HOSTNAME	指定正在运行服务器的主机名、Unix域套接字的路径或者域名。接受以“,”分隔的字符串来指定多个主机地址，支持指定多个主机地址，支持指定IPv6主机地址。 当指定多个主机地址时，默认情况下会跳过故障的节点，自动选择第一个可用的主节点进行连接。可通过设置PGTARGETSESSIONATTRS环境变量的值来选择连接到不同类型的节点，PGTARGETSESSIONATTRS变量取值如下： ● read-write：可读写的节点。 ● read-only：只读节点。 ● primary或者不设定：主节点。 ● standby：备节点。 ● prefer-standby：首选备节点，没有备节点则转为any。 ● any：不进行角色检查。 **说明** 主CN节点和主DN节点都是主节点，一般情况下DN节点都是只读节点，可以通过设置PGTARGETSESSIONATTRS为read-write来选择连接到CN节点。	如果省略主机名，gsql将通过Unix域套接字与本地主机的服务器相连，或者在没有Unix域套接字的机器上，通过TCP/IP与localhost连接。
-p, --port=PORT	指定数据库服务器的端口号。可以配置一个或多个，当配置一个时，所有的主机地址都使用同一个端口连接；当配置多个时，顺序与主机地址顺序相同，个数必须与主机地址数相等，当不相等时会报错。 可以通过port参数修改默认端口号。	默认端口可通过编译参数来指定，不指定时默认为5432。
-U, --username=USERNAME	指定连接数据库的用户。 **说明** ● 通过该参数指定用户连接数据库时，需要同时提供用户密码用以身份验证。可以通过交互方式输入密码，或者通过-W参数指定密码。 ● 用户名中包含有字符$，需要在字符$前增加转义字符才可成功连接数据库。	字符串，默认使用与当前操作系统用户同名的用户。

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
-W, --password=PASSWORD	当使用-U参数连接本地数据库或者连接远端数据库时，可通过该选项指定密码。 **说明** • 登录CN所在服务器后连接本地CN实例时，默认使用trust连接，会忽略此参数。 • 用户密码中包含特殊字符“\”和“`”时，需要增加转义字符才可成功连接数据库。 • 如果用户未输入该参数，但是数据库连接需要用户密码，这时将出现交互式输入，请用户输入当前连接的密码。该密码最长长度为999字节，受限于GUC参数password_max_length的最大值。	字符串。

版权所有 © 华为云计算技术有限公司

表 4-11 Unix Domain Socket 运维通道的运维命令

参数	参数说明	取值范围
-O, --emergency-command=EMERGENCYCOMMAND	Unix Domain Socket运维通道的运维命令。 在极端情况下，gsql工具无法与GaussDB建立连接时，可以使用该命令进行应急运维。此命令通过Unix Domain Socket运维通道与GaussDB通信，实现一键运维功能。 -O选项的参数格式具体如下： "{port} {command} [{param} [...]]" {port}：数据库监听端口。 {command}：具体运维命令。 {param}：该运维命令的参数，可能有0个或多个。 **说明** • 使用此选项时，gsql工具必须与GaussDB进程位于同一台服务器上。 • 如果运维命令的参数包含空格，请在参数两边加上引号。	• "{port} mem_log"：记载全局的内存概况与全局级内存上下文、线程级内存上下文、session级内存上下文的top20内存上下文占用详情。该功能仅在GUC参数use_workload_manager开启时有效，范围同内存过载记录mem_log特性。无参数。示例： -O "19995 mem_log" • "{port} kill_wait_event {wait_event}"：查杀指定等待事件会话。参数{wait_event}表示具体的锁、轻量级锁以及I/O的信息等。示例： -O "19995 kill_wait_event relation" • "{port} kill_business_sql"：查杀非rdsAdmin用户的全部业务SQL语句。无参数。示例： -O "19995 kill_business_sql" • "{port} kill_slow_sql {username}"：批量查杀非默认数据库下，指定用户的状态为非idle的执行超过1分钟的SQL语句。参数{username}表示用户名。示例： -O "19995 kill_slow_sql omm" • "{port} kill_slow_sql {username} {database}"：批量查杀指定用户在指定数据库中的执行超过1分钟的SQL语句。参数{username}表示用户名，{database}表示数据库名。示例： -O "19995 kill_slow_sql omm db1" • "{port} kill_executing_sql {username}"：查杀非默认数据库下，指定用户的状态为active的正在执行的SQL语句。参数{username}表示用户名。示例： -O "19995 kill_executing_sql omm"

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
		• "{port} kill_mem_topn {top n}"：查杀内存占用top n的SQL语句。参数{top n}表示指定的top n。示例： -O "19995 kill_mem_topn 10" • "{port} kill_cpu_topn {top n}"：查杀CPU TIME占用top n的SQL语句。参数{top n}表示指定的top n。示例： -O "19995 kill_cpu_topn 10"

示例

示例一

连接数据库并执行命令。

步骤1 使用gsql连接到GaussDB服务器。

gsql工具使用-d参数指定目标数据库名、-U参数指定数据库用户名、-h参数指定主机名、-p参数指定端口号信息。详细的gsql参数请参见**参数说明**。

```
--使用jack用户连接到远程主机postgres数据库的8000端口。（根据实际需要替换各参数值）
gsql -h 10.180.123.163 -d postgres -U jack -p 8000
```

说明

- 若未指定数据库名称，则使用初始化时默认生成的数据库名称。
- 若未指定数据库用户名，则默认使用当前操作系统用户作为数据库用户名。
- 当某个值没有前面的参数（-d、-U等）时，若连接的命令中没有指定数据库名（-d），则该参数会被解释为数据库名；如果已经指定数据库名（-d）而没有指定数据库用户名（-U），该参数则会被解释为数据库用户名。

  例如，以下命令中，参数postgres和omm不属于任何选项，会被分别解释为数据库名和用户名。

  ```
  gsql postgres omm -p 8000
  ```

  等效于以下命令。

  ```
  gsql -d postgres -U omm -p 8000
  ```

步骤2 执行gsql元命令和SQL语句。

```
--创建数据库human_staff。
gaussdb=# CREATE DATABASE human_staff;
CREATE DATABASE

--列出GaussDB中所有的数据库和描述信息。
gaussdb=# \l
                              List of databases
     Name      |  Owner   | Encoding  | Collate | Ctype |   Access privileges
---------------+----------+-----------+---------+-------+-----------------------
 human_resource | omm | SQL_ASCII | C       | C     |
 postgres       | omm | SQL_ASCII | C       | C     |
 template0      | omm | SQL_ASCII | C       | C     | =c/omm         +
                |     |           |         |       | omm=CTc/omm
 template1      | omm | SQL_ASCII | C       | C     | =c/omm          +
```


版权所有 © 华为云计算技术有限公司

```
               |          |           |         |       | omm=CTc/omm
 human_staff     | omm | SQL_ASCII | C       | C     |
 gaussdb         | omm | SQL_ASCII | C       | C     |
(6 rows)

--删除数据库human_staff。
gaussdb=# DROP DATABASE human_staff;
DROP DATABASE
```

更多gsql元命令请参见**元命令参考**。

----结束

示例二

获取帮助信息。

```
gsql --help
gsql is the GaussDB Kernel interactive terminal.

Usage:
  gsql [OPTION]... [DBNAME [USERNAME]]

General options:
  -c, --command=COMMAND    run only single command (SQL or internal) and exit
  -d, --dbname=DBNAME      database name to connect to (default: "omm")
  -f, --file=FILENAME      execute commands from file, then exit
  -l, --list               list available databases, then exit
  -v, --set=, --variable=NAME=VALUE
                           set gsql variable NAME to VALUE
  -V, --version            output version information, then exit
  -X, --no-gsqlrc          do not read startup file (~/.gsqlrc)
  -1 ("one"), --single-transaction
                           execute command file as a single transaction
  -?, --help               show this help, then exit

Input and output options:
  -a, --echo-all           echo all input from script
  -e, --echo-queries       echo commands sent to server
  -E, --echo-hidden        display queries that internal commands generate
  -k, --with-key=KEY       the key for decrypting the encrypted file
  -L, --log-file=FILENAME  send session log to file
  -m, --maintenance        can connect to cluster during 2-pc transaction recovery
  -n, --no-libedit         disable enhanced command line editing (libedit)
  -o, --output=FILENAME    send query results to file (or |pipe)
  -q, --quiet              run quietly (no messages, only query output)
  -C, -C1, --enable-client-encryption=1
                           enable client encryption feature
  -C2, --enable-client-encryption=2
                           enable client sorting based client encryption feature
  -C3, --enable-client-encryption=3
                           enable Trusted Domain operation based on client encryption feature

  -s, --single-step        single-step mode (confirm each query)
  -S, --single-line        single-line mode (end of line terminates SQL command)
  -y, --slash-command      enable slash command to re-execute query in buffer

Output format options:
  -A, --no-align           unaligned table output mode
  -F, --field-separator=STRING
                           set field separator (default: "|")
  -H, --html               HTML table output mode
  -P, --pset=VAR[=ARG]     set printing option VAR to ARG (see \pset command)
  -R, --record-separator=STRING
                           set record separator (default: newline)
  -r                       if this parameter is set,use libedit
  -t, --tuples-only        print rows only
  -T, --table-attr=TEXT    set HTML table tag attributes (e.g., width, border)
  -x, --expanded           turn on expanded table output
```


版权所有 © 华为云计算技术有限公司

```
  -z, --field-separator-zero
                      set field separator to zero byte
  -0, --record-separator-zero
                      set record separator to zero byte
  -2, --pipeline          use pipeline to pass the password, forbidden to use in terminal
                      must use with -c or -f

Connection options:
  -h, --host=HOSTNAME      database server host or socket directory (default: "/data/huawei/wisequery/
perfadm_mppdb")
                      allow multi host IP address with comma separator, Use the
                      PGTARGETSESSIONATTRS(default: "primary") variable to select the
                      IP address to be connected.
                      (PGTARGETSESSIONATTRS can be {read-write|read-only|primary|
                       standby|prefer-standby|any})
  -p, --port=PORT          database server port (default: "5432")
  -U, --username=USERNAME  database user name (default: "omm")
  -W, --password=PASSWORD  the password of specified database user

Emergency O&M command:
  -O, --emergency-command=EMERGENCYCOMMAND
                      emergency O&M command

For more information, type "\?" (for internal commands) or "\help" (for SQL
commands) from within gsql, or consult the gsql section in the GaussDB Kernel
documentation.
```

示例三

执行Unix Domain Socket运维通道的运维命令。以下为如何通过Unix Domain Socket运维通道查看内存使用情况的简单示例：

步骤1 登录GaussDB节点所在服务器，执行以下gsql命令，确认是否生成内存概况以及内存上下文等使用记录。

```
gsql -O "{port} mem_log"
```

其中，{port}为数据库节点名称。当显示true时，则表示已生成具体的详情信息；如果执行失败，则显示失败原因。

```
-- 示例中19995为数据库监听端口。
gsql -O "19995 mem_log"
true
```

步骤2 查看$GAUSSLOG/mem_log/{nodename}目录下生成的mem_log文件内容，其中{nodename}为数据库节点名称。

```
-- 示例中datanode为数据库节点名称。
ls -l $GAUSSLOG/mem_log/datanode
total 8
-rw-------. 1 omm omm 7543 May 28 15:15 mem_log-2025-05-28_151518.log
```

步骤3 执行以下命令查看文件内容。

```
cat $GAUSSLOG/mem_log/datanode/mem_log-time.log
```

其中，mem_log-time.log为**步骤2**中$GAUSSLOG/mem_log/datanode路径下生成的文件（文件名后缀为.log）。

----结束

4.2.1.2 元命令参考

使用GaussDB数据库命令行交互工具登录数据库后，gsql提供了一些元命令。连接数据库后，可以使用\?命令，显示所有可用的元命令帮助信息。

版权所有 © 华为云计算技术有限公司

注意事项

- 一个gsql元命令的格式是反斜杠后面紧跟一个动词，然后是任意参数。参数命令动词和其他参数以任意个空白字符间隔。
- 要在参数里面包含空白，必须用单引号把它引起来。要在这样的参数里包含单引号，可以在前面加一个反斜杠。任何包含在单引号里的内容都会被进一步进行类似C语言的替换：\n（新行）、\t（制表符）、\b（退格）、\r（回车）、\f（换页）、\digits（八进制表示的字符）、\xdigits（十六进制表示的字符）。
- 用反引号“`”包围的内容被当作一个命令行传入shell。该命令的输出（删除了结尾的新行）会替换原参数中的反引号及其包围的文本，被当作参数值。
- 如果不带引号的参数以冒号（:）开头，它会被当做一个gsql变量，并且该变量的值最终会成为真正的参数值。
- 有些命令以一个SQL标识的名称（比如一个表）为参数。这些参数遵循SQL语法关于双引号的规则：不带双引号的标识强制转换成小写，而双引号保护字母不进行大小写转换，并且允许在标识符中使用空白。在双引号中，成对的双引号在结果名称中分析成一个双引号。比如，FOO"BAR"BAZ解析成fooBARbaz；而"Aweird""name"解析成Aweird"name。
- 对参数的分析在遇到另一个不带引号的反斜杠时停止，这里会认为是一个新的元命令的开始。特殊的双反斜杠序列（\\）标识参数的结尾，并将继续分析后面的SQL语句。这样SQL和gsql命令可以在一行中混合使用。但是在任何情况下，一条元命令的参数不能延续超过行尾。
- M-Compatibility模式数据库不支持\h元命令。

命令格式及参数说明

元命令的详细说明请参见**表4-12**、**表4-13**、**表4-14**、**表4-15**、**表4-17**、**表4-19**、**表4-20**、**表4-21**和**表4-23**。

说明

以下命令中所提到的FILE代表文件路径。此路径可以是绝对路径（如/home/gauss/file.txt），也可以是相对路径（如file.txt，file.txt会默认在用户执行gsql命令所在的路径下创建）。

表 4-12 一般的元命令

参数	参数说明	取值范围
\copyright	显示GaussDB的版本和版权信息。	-
\g [FILE] or ;	执行查询（并将结果发送到文件或管道）。	-
\h(\help) [NAME]	给出指定SQL语句的语法帮助。	如果没有给出NAME，gsql将列出可获得帮助的所有命令。如果NAME是一个星号（*），则显示所有SQL语句的语法帮助。

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
\parallel [on [num]|off]	控制并发执行开关。 ● on：打开控制并发执行开关，且最大并发数为num。num的默认值为1024。 ● off：关闭控制并发执行开关。 **说明** ● 不支持事务中开启并发执行以及并发中开启事务。 ● 不支持\d这类元命令的并发。 ● 并发select返回结果混乱问题可接受，core、进程停止响应不可接受。 ● 不推荐在并发中使用set语句，否则导致结果与预期不一致。 ● 不支持创建临时表。如需使用临时表，需要在开启parallel之前创建好，并在parallel内部使用。parallel内部不允许创建临时表。 ● \parallel执行时最多会启动num个独立的gsql进程连接服务器。 ● \parallel中所有作业的持续时间不能超过session_timeout，否则可能会导致并发执行过程中断开连接。 ● 在\parallel on之后一条或多条命令，会等到\parallel off执行后才会执行，因而，\parallel on之后需要有对应的\parallel off，否则\parallel on后的命令都无法执行。 ● 服务器能接受的最大连接数受max_connection及当前已有连接数限制。设置num时，请根据服务器当前可接受的实际连接数合理指定。	-
\q	退出gsql程序。在一个脚本文件里，只在脚本终止时执行。	-

表 4-13 查询缓存区元命令

参数	参数说明
\e [FILE] [LINE]	使用外部编辑器编辑查询缓冲区（或者文件）。
\ef [FUNCNAME [LINE]]	使用外部编辑器编辑函数定义。如果指定了LINE（即行号），则光标会指到函数体的指定行。
\p	打印当前查询缓冲区到标准输出。
\r	重置（或清空）查询缓冲区。
\w FILE \w |COMMAND	将当前查询缓冲区输出到FILE指定的文件。 将当前查询缓冲区通过管道发送给COMMAND指定的shell命令。

版权所有 © 华为云计算技术有限公司

表 4-14 输入/输出元命令

参数	参数说明
\copy { table [(column_list)] | (query) } { from | to } { filename | stdin | stdout | pstdin | pstdout } [copy_parameter] [parallel integer] [NO_SPLIT_FILE] [FIELD_NO_EOL]	在任何gsql客户端登录数据库成功后可以执行导入导出数据，这是一个运行SQL COPY命令的操作，但不是读取或写入指定文件的服务器，而是读取或写入文件，并在服务器和本地文件系统之间路由数据。这意味着文件的可访问性和权限是本地用户的权限，而不是服务器的权限，并且不需要数据库初始化用户权限。

版权所有 © 华为云计算技术有限公司

参数	参数说明
	说明 ● \COPY只适合小批量，格式良好的数据导入，导入数据应优先选择GDS或COPY。 ● \COPY可以指定数据导入时的客户端数量，从而实现数据文件的并行导入，目前并发数范围为[1, 客户端CPU核数的两倍]。 ● \COPY指定parallel参数进行并行导入时，将启动多个线程导入数据，每个线程都是单独的事务，可能出现数据部分导入的结果。请在充分理解并行可能造成的结果后，谨慎使用并行导入数据。 ● \COPY可以指定pstdin和pstdout。pstdin声明输入是来自gsql的标准输入，pstdout声明输出是来自gsql的标准输出。 ● \COPY并行导入目前存在以下约束： 临时表的并行导入不支持、在事务内的并行导入不支持、对二进制文件的并行导入不支持、数据导入支持AES128加密时不支持以及COPY选项中存在EOL时不支持。在这些情况下，即使指定了parallel参数，仍然会按照非并行流程执行；在数据文件中数据少于100行时，也会按照非并行流程执行。不支持在指定SEQUENCE的场景下进行并行导入，会出现报错。 在对带触发器的表执行并行导入时，若触发器内部存在向带约束的表（记为t1）插入数据的操作，则可能因并行导入导致死锁问题。具体表现为：多个并行导入进程在操作过程中，t1表因约束条件（如唯一键、外键等）导致不同事务间产生锁等待状态，若形成循环等待则触发数据库死锁检测机制，最终导致导入失败并抛出deadlock错误。因此当目标表包含触发器，且触发器逻辑中存在向带约束的表插入数据的操作时，不建议使用并行导入。 在对带约束的表（如唯一约束等）执行并行导入时，如果不使用load语法，若数据文件本身含有违反约束冲突的数据，则可能因并行导入导致死锁问题。具体表现为：多个并行导入线程在操作过程中，t1表因约束条件（如唯一键、外键等）导致不同事务间产生锁等待状态，若形成循环等待则触发数据库死锁检测机制，最终导致数据部分导入失败，并抛出错误。因此当目标表包含约束条件（如唯一键等），建议使用load语法进行并行导入。 FIELD_NO_EOL语法在parallel大于1时生效，在使用FIELD_NO_EOL语法时会加快数据导入的速度，但需要自行确认导入数据文件中的数据行字段内没有包含换行符。如果包含换行符，可能会导致： ● 一个完整的数据行被数据字段内的换行符分割为两部分，两部分被分配给不同的线程向客户端发送数据，导致插入字段与表结构不匹配的错误。 ● 在使用skip语法时会跳过前面指定数量的数据行。字段内的换行符可能让一行数据被误认为是两行数据，导致skip语法跳过的数据行数错误。 ● \COPY新增加NO_SPLIT_FILE语法，当parallel大于1时生效，如果使用该参数，表示使用原有逻辑无法解除并行度最大值为8的限制。默认解除并行度最大值为8的限制，自动拆分数据文件实现并行。 ● \COPY元命令不建议与其它数据库命令放在同一行执行。 ● \COPY语句不允许多行。 ● \COPY中间不能包含注释，否则导致注释后面的语句被截断。 ● \COPY执行时，若遇到\copy: parse error at end of line报错，建议优先校验当前命令是否完全遵循COPY语法规范，重点关注字段分隔符、导入导出格式相关参数的正确性。 ● \COPY执行时，当通过copy_custom_id等参数传递自定义字符串时，建议避免字符串内容包含EOL、BINARY等COPY语法参数，以防止并行导入出现异常。 ● \COPY仅在设置parallel参数大于1，开启并行时才支持参数skip和limit，功能如下： ● skip int_number：指定数据导入时，跳过数据文件的前int_number行。 ● limit int_number：指定从输入数据源中最多处理的行数。 ● \COPY支持COPY的所有命令，在参数列表中用copy_parameter来指代，非M兼容下关于COPY所有命令的详细信息请参见**COPY**，M兼容下

版权所有 © 华为云计算技术有限公司

参数	参数说明
\echo [STRING]	把字符串写到标准输出。
prompt [STRING]	把字符串写到标准输出（等同于\echo）。
\i FILE	从文件FILE中读取内容，并将其当作输入，执行查询。
\i+ FILE KEY	执行加密文件中的命令。
\ir FILE	和\i类似，只是相对于存放当前脚本的路径。
\ir+ FILE KEY	和\i+类似，只是相对于存放当前脚本的路径。
\o [FILE]	把后续所有查询结果发送到FILE指定的文件里。
\o [| COMMAND]	把后续所有查询结果通过管道发送给COMMAND指定的shell命令。
\qecho [STRING]	把字符串写到查询结果输出流里。
\print_copy_lines [on | off]	切换COPY FROM STDIN命令是否打印"COPY XXX"回显，默认开启。 ● on：COPY FROM STDIN命令打印当前导入多少行数据。 ● off：COPY FROM STDIN命令不打印当前导入多少行数据。

表 4-15 显示信息元命令

参数	参数说明	取值范围	示例
\d[S+]	列出当前search_path中模式下所有的表、视图和序列。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。	-	列出当前search_path中模式下所有的表、视图和序列。 gaussdb=# \d
\d[S+] NAME	列出指定表、视图和索引的结构。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。	-	假设存在表a，列出指定表a的结构。 gaussdb=# \dtable+ a

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围	示例
\d+ [PATTERN]	列出所有表、视图和索引。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的表、视图和索引。	-	列出所有名称以f开头的表、视图和索引。 gaussdb=# \d+ f*
\da[S] [PATTERN]	列出所有可用的聚集函数，以及它们操作的数据类型和返回值类型。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的聚集函数。	-	列出所有名称以f开头可用的聚集函数，以及它们操作的数据类型和返回值类型。 gaussdb=# \da f*
\db[+] [PATTERN]	列出所有可用的表空间。如果声明了PATTERN，只显示名称匹配PATTERN的表空间。	-	列出所有名称以p开头的可用表空间。 gaussdb=# \db p*
\dc[S+] [PATTERN]	列出所有字符集之间的可用转换。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的转换。	-	列出所有字符集之间的可用转换。 gaussdb=# \dc *
\dC[+] [PATTERN]	列出所有类型转换。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的转换。	PATTERN需要使用实际类型名，不能使用别名。	列出所有名称以c开头的类型转换。 gaussdb=# \dC c*
\dd[S] [PATTERN]	显示所有可视对象。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配的对象。对象包括：聚集、函数、操作符、类型、关系(表、视图、索引、序列)、规则。	-	列出所有可视对象。 gaussdb=# \dd

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围	示例
\ddp [PATTERN]	显示所有默认的使用权限。如果指定了PATTERN，只显示名称匹配PATTERN的使用权限。	-	列出所有默认的使用权限。 gaussdb=# \ddp
\dD[S+] [PATTERN]	列出所有可用域。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的域。	-	列出所有可用域。 gaussdb=# \dD
\det[+] [PATTERN]	列出所有的外部表。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的表。	-	列出所有的外部表。 gaussdb=# \det
\des[+] [PATTERN]	列出所有的外部服务器。如果声明了PATTERN，只显示名称匹配PATTERN的服务器。	-	列出所有的外部服务器。 gaussdb=# \des
\deu[+] [PATTERN]	列出用户映射信息。如果声明了PATTERN，只显示名称匹配PATTERN的信息。	-	列出用户映射信息。 gaussdb=# \deu
\dew[+] [PATTERN]	列出封装的外部数据。如果声明了PATTERN，只显示名称匹配PATTERN的数据。	-	列出封装的外部数据。 gaussdb=# \dew
\df[antw] [S+] [PATTERN]	列出所有可用函数，以及它们的参数和返回的数据类型。a代表聚集函数，n代表普通函数，t代表触发器，w代表窗口函数。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的函数。	-	列出所有可用函数，以及它们的参数和返回的数据类型。 gaussdb=# \df

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围	示例
\dF[+] [PATTERN]	列出所有的文本搜索配置信息。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的配置信息。	-	列出所有的文本搜索配置信息。 gaussdb=# \dF+
\dFd[+] [PATTERN]	列出所有的文本搜索字典。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的字典。	-	列出所有的文本搜索字典。 gaussdb=# \dFd
\dFp[+] [PATTERN]	列出所有的文本搜索分析器。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的分析器。	-	列出所有的文本搜索分析器。 gaussdb=# \dFp
\dFt[+] [PATTERN]	列出所有的文本搜索模板。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只显示名称匹配PATTERN的模板。	-	列出所有的文本搜索模板。 gaussdb=# \dFt
\dg[+] [PATTERN]	列出所有数据库角色。如果指定了PATTERN，只显示名称匹配PATTERN的角色。 **说明** 因为用户和群组的概念被统一为角色，所以这个命令等价于\du。为了和以前兼容，所以保留两个命令。	-	列出名称为“j?e”所有数据库角色（“?”表示任一字符）。 gaussdb=# \dg j?e
\dL[S+] [PATTERN]	列出可用的程序语言。如果指定了PATTERN，只列出名称匹配PATTERN的语言。	-	列出可用的程序语言。 gaussdb=# \dL

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围	示例
\dm[S+] [PATTERN]	列出物化视图。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果指定了PATTERN，只列出名称匹配PATTERN的物化视图。	-	列出物化视图。 gaussdb=# \dm
\dn[S+] [PATTERN]	列出所有模式（名称空间）。如果向命令追加+，会列出每个模式相关的权限及描述。如果声明了PATTERN，只列出名称匹配PATTERN的模式名。缺省时，只列出用户创建的模式。	-	列出所有名称以d开头的模式以及相关信息。 gaussdb=# \dn+ d*
\do[S] [PATTERN]	列出所有可用的操作符，以及它们的操作数和返回的数据类型。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只列出名称匹配PATTERN的操作符。缺省时，只列出用户创建的操作符。	-	列出所有可用的操作符，以及它们的操作数和返回的数据类型。 gaussdb=# \do
\dO[S+] [PATTERN]	列出排序规则。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果声明了PATTERN，只列出名称匹配PATTERN的规则。缺省时，只列出用户创建的规则。	-	列出排序规则。 gaussdb=# \dO
\dp [PATTERN]	列出可用的表、视图以及相关的权限信息。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果指定了PATTERN，只列出名称匹配PATTERN的表、视图。	-	列出可用的表、视图以及相关的权限信息。 gaussdb=# \dp \dp显示结果如下： rolename=xxxx/yyyy --赋予一个角色的权限 =xxxx/yyyy --赋予public的权限 xxxx表示赋予的权限，yyyy表示授予这个权限的角色。权限的参数说明请参见**表4-16**。

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围	示例
\drds [PATTERN1 [PATTERN2]]	列出所有修改过的配置参数。这些设置可以是针对角色的、针对数据库的或者同时针对两者的。PATTERN1和PATTERN2表示要列出的角色PATTERN和数据库PATTERN。如果声明了PATTERN，只列出名称匹配PATTERN的规则。缺省或指定*时，则会列出所有设置。	-	列出数据库所有修改过的配置参数。 gaussdb=# \drds * *dbname*
\dT[S+] [PATTERN]	列出所有的数据类型。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。如果指定了PATTERN，只列出名称匹配PATTERN的类型。	-	列出所有的数据类型。 gaussdb=# \dT
\du[+] [PATTERN]	列出所有数据库角色。如果指定了PATTERN，则只列出名称匹配PATTERN的角色。 **说明** 因为用户和群组的概念被统一为角色，所以这个命令等价于\dg。为了和以前兼容，所以保留两个命令。	-	列出所有数据库角色。 gaussdb=# \du
\dE[S+] [PATTERN] \di[S+] [PATTERN] \ds[S+] [PATTERN] \dt[S+] [PATTERN] \dv[S+] [PATTERN]	这一组命令，字母E、i、s、t和v分别代表着外部表、索引、序列、表和视图。可以以任意顺序指定其中一个或者它们的组合来列出这些对象。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。例如：\dit列出所有的索引和表。如果指定了PATTERN，只列出名称匹配该PATTERN的对象。默认情况下只会显示用户创建的对象。通过PATTERN或者S修饰符可以把系统对象包括在内。	-	列出所有的索引和视图。 gaussdb=# \div

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围	示例
\dx[+] [PATTERN]	列出安装数据库的扩展信息。如果指定了PATTERN，则只列出名称匹配PATTERN的扩展信息。	-	列出安装数据库的扩展信息。 gaussdb=# \dx
\l[+]	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。	-	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。 gaussdb=# \l
\sf[+] FUNCNAME	显示函数的定义。 **说明** • 对于带圆括号的函数名，需要在函数名两端添加双引号，并且在双引号后面加上参数类型列表。参数类型列表两端添加圆括号。 • 如果存在同名的函数，则会返回多个函数的定义。	-	假设存在函数function_a和函数名带圆括号的函数func()name，列出函数的定义。 gaussdb=# \sf function_a gaussdb=# \sf "func()name"(argtype1, argtype2)
\z [PATTERN]	列出数据库中所有表、视图和序列，以及它们相关的访问特权。如果给出任何PATTERN，则被当成一个正则表达式，只显示匹配的表、视图、序列。	-	列出数据库中所有表、视图和序列，以及它们相关的访问特权。 gaussdb=# \z

版权所有 © 华为云计算技术有限公司

说明

- 表4-15中的选项S表示显示系统对象，+表示显示对象附加的描述信息。
- PATTERN用来指定要被显示的对象名称。关于PATTERN的注意事项如下：
  - 在最简单的情况下，PATTERN是该对象的准确名称。在PATTERN中的字符通常会被变成小写形式（和在SQL名称中相同），例如\dt FOO将会显示名为foo的表。和在SQL名称中相同，把PATTERN放在双引号中可以阻止它被转换成小写形式。如果需要在一个PATTERN中包括一个真正的双引号字符，则需要把它写成两个相邻的双引号，这同样是符合SQL引用标识符的规则。例如，\dt "FOO""BAR"将显示名为FOO"BAR（不是foo"bar）的表。和普通的SQL名称规则不同，不能只在PATTERN的一部分周围放上双引号，例如\dt FOO"FOO"BAR将会显示名为fooFOObar的表。
  - 不使用PATTERN参数时，\d命令会显示当前schema搜索路径中可见的全部对象——这等价于用*作为PATTERN。所谓对象可见是指可以直接用名称引用该对象，而不需要用schema来进行限定。要查看数据库中所有的对象而不管它们的可见性，可以把*.*用作PATTERN。
  - 如果放在一个PATTERN中，*将匹配任意字符序列（包括空序列），而?会匹配任意的单个字符（和Unix shell的文件名PATTERN相同）。例如，\dt int*会显示名称以int开始的表。但是如果被放在双引号内，*和?就会失去这些特殊含义而变成普通的字符。
  - 包含一个点号（.）的PATTERN被解释为一个schema名称模式后面跟上一个对象名称模式。例如，\dt foo*.*bar*会显示名称以foo开始的schema中所有名称包括bar的表。如果没有出现点号，那么模式将只匹配当前schema搜索路径中可见的对象。同样，双引号内的点号会失去其特殊含义并且变成普通的字符。
  - 高级用户可以使用字符类等正则表达式记法，如[0-9]可以匹配任意数字。所有的正则表达式特殊字符都按照POSIX正则表达式使用。以下字符除外：
    - .会按照上文所述作为一种分隔符。
    - *会被翻译成正则表达式记号.*。
    - ?会被翻译成.。
    - $则按字面意思匹配。
  - 根据需要，可以通过书写?、(*R*+|)、(*R*|)和*R*?来分别模拟PATTERN字符.、*R**和*R*?。$不需要作为一个正则表达式字符，因为PATTERN必须匹配整个名称，而不是按照正则表达式的常规用法解释（换句话说，$会被自动地追加到PATTERN上）。如果不希望该PATTERN的匹配位置被固定，可以在开头或者结尾写上*。注意在双引号内，所有的正则表达式特殊字符会失去其特殊含义并且按照其字面意思进行匹配。另外，在操作符名称PATTERN中（即\do的PATTERN参数），正则表达式特殊字符也按照字面意思进行匹配。

表 4-16 权限的参数说明

参数	参数说明
r	SELECT：允许对指定的表、视图读取数据。
w	UPDATE：允许对指定表更新字段。
a	INSERT：允许对指定表插入数据。
d	DELETE：允许删除指定表中的数据。
D	TRUNCATE：允许清理指定表中的数据。
x	REFERENCES：允许创建外键约束，分布式场景暂不支持。
t	TRIGGER：允许在指定表上创建触发器。
X	EXECUTE：允许使用指定的函数，以及利用这些函数实现的操作符。

版权所有 © 华为云计算技术有限公司

参数	参数说明
U	USAGE： ● 对于过程语言，允许用户在创建函数时，指定过程语言。 ● 对于模式，允许访问包含在指定模式中的对象。 ● 对于序列，允许使用nextval函数。
C	CREATE： ● 对于数据库，允许在该数据库里创建新的模式。 ● 对于模式，允许在该模式中创建新的对象。 ● 对于表空间，允许在其中创建表，以及允许创建数据库和模式的时候把该表空间指定为其缺省表空间。
c	CONNECT：允许用户连接到指定的数据库。
T	TEMPORARY：允许创建临时表。
A	ALTER：允许用户修改指定对象的属性。
P	DROP：允许用户删除指定的对象。
m	COMMENT：允许用户定义或修改指定对象的注释。
i	INDEX：允许用户在指定表上创建索引。
v	VACUUM：允许用户对指定的表执行ANALYZE和VACUUM操作。
*	授权以上权限。

表 4-17 格式化元命令

参数	参数说明	取值范围
\a	对齐模式和非对齐模式之间的切换。	-
\C [STRING]	为表输出设置标题，或者取消标题（如果没有参数）。	任意字符串或空。
\f [STRING]	指定在非对齐输出模式下，用于分隔字段的字符或字符串。	任意字符串。
\H	标准输出和HTML表格格式输出之间的切换。	-
\pset NAME [VALUE]	设置影响查询结果表输出的选项。	NAME的取值见**表4-18**。
\t [on | off]	切换输出的表格格式。	● on：只显示数据行，不显示列标题和行计数。 ● off：显示完整的表格，包括列标题和行计数。

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
\T [STRING]	指定在使用HTML输出格式时放在table标签里的属性。如果参数为空，表示不设置。	-
\x [on | off | auto]	切换扩展行显示模式。	● on：启用扩展显示模式，每个字段单独一行显示。 ● off：禁用扩展显示模式，使用标准表格显示。 ● auto：根据查询结果的宽度自动选择是否使用扩展显示模式。

表 4-18 可调节的打印选项

选项	选项说明	取值范围
border	用于控制表格输出时的边框样式。	● 在HTML格式下，取值范围为大于0的整数。 ● 在其他格式下，取值范围： – 0：无边框，表格输出时不显示任何边框线。 – 1：内部边框，显示列之间的分隔线，但不显示表格的外边框。 – 2：完整边框，显示外边框和内部分隔线。
expanded (或x)	在正常和扩展格式之间切换。	● on：启用扩展显示模式，每个字段单独一行显示。 ● off：禁用扩展显示模式，使用标准表格显示。标准格式只对aligned和wrapped格式有意义。 ● auto：根据查询结果的宽度自动选择是否使用扩展显示模式。
fieldsep	用于控制在非对齐输出模式下，各个字段之间的分隔符。	可以设置为任意字符串。常见的取值包括“,”、“\t”、“|”等，如果不设置该选项，默认使用空格作为字段分隔符。
fieldsep_zero	用于设置未对齐输出模式下的字段分隔符为ASCII NUL字符。	-
footer	用于设置查询结果的页脚。	● on：显示页脚信息。 ● off：不显示页脚信息。

版权所有 © 华为云计算技术有限公司

选项	选项说明	取值范围
format	控制查询结果的输出格式。	• unaligned（u）：以未对齐的形式显示，字段之间使用当前的字段分隔符分隔。 • aligned（a）：以对齐的表格形式显示。 • wrapped（w）：与aligned类似，但会自动换行过长的值。 • html（h）：以HTML表格格式显示。 • latex（l）：以LaTeX表格格式显示。 • troff-ms（t）：以troff ms宏格式显示。
null	用于设置NULL值的显示方式。	任意字符串。如果不设置该选项，默认使用空白字符来代替NULL值。
numericlocale	用于控制数字输出格式时是否使用当前系统的区域设置。	• on：数字输出时会使用当前系统的区域设置进行格式化。例如，在英语环境下，数字1,000.00会显示为1,000.00，在德语环境下会显示为1.000,00。 • off：数字输出时不使用区域设置，而是使用标准的数字格式，如1000.00。
pager	控制查询和gsql帮助输出的分页器。如果设置了环境变量PAGER，输出将被指向到指定程序，否则使用系统缺省。	• on：当输出到终端且不适合屏幕显示时，使用分页器。 • off：不使用分页器。 • always：当输出到终端无论是否符合屏幕显示时，都使用分页器。
recordsep	用于设置未对齐输出模式下的记录分隔符。	可以设置为任意字符串。常见的取值包括换行符\n、回车符\r、回车换行符\r\n等。如果不设置该选项，默认使用换行符\n作为记录分隔符。
recordsep_zero	用于设置未对齐输出模式下的记录分隔符为ASCII NUL字符。	-
tableattr（或T）	声明放在HTML输出格式中HTML table标签的属性（例如：cellpadding或bgcolor）。注意：如果已经使用\pset border命令，这里无需再设置border。	例如bgcolor=#FFFFFF等。如果没有给出value，则表示不设置表的属性。

版权所有 © 华为云计算技术有限公司

选项	选项说明	取值范围
title	用于设置查询结果的标题。	可以设置为任意字符串，如果不设置该选项，查询结果将不会显示任何标题。
tuples_only（或t）	在完全显示和只显示实际的表数据之间切换。完全显示将输出像列头、标题、各种脚注等信息。在tuples_only模式下，只显示实际的表数据。	-
feedback	切换是否输出结果行数。	-

表 4-19 连接元命令

参数	参数说明	取值范围
\c[onnect] [DBNAME|- USER|- HOST|- PORT|-]	连接到一个新的数据库。当数据库名称长度超过63个字节时，默认前63个字节有效，连接到前63个字节对应的数据库，但是gsql的命令提示符中显示的数据库对象名仍为截断前的名称。 **说明** 重新建立连接时，如果切换数据库登录用户，可能会出现交互式输入，要求输入新用户的连接密码。该密码最大长度为999字节，受限于GUC参数password_max_length的最大值。	-
\encoding [ENCODING]	设置客户端字符编码格式。	不带参数时，显示当前的编码格式。
\conninfo	输出当前连接的数据库的信息。	-

表 4-20 操作系统元命令

参数	参数说明	取值范围
\cd [DIR]	切换当前的工作目录。	绝对路径或相对路径，且满足操作系统路径命名规则。
\setenv NAME [VALUE]	设置环境变量NAME为VALUE，如果没有给出VALUE值，则不设置环境变量。	-
\timing [on | off]	以毫秒为单位显示每条SQL语句的执行时间（不包括屏显打印时间）。	• on：表示打开显示。 • off：表示关闭显示。

版权所有 © 华为云计算技术有限公司

参数	参数说明	取值范围
\! [COMMAND]	用于执行操作系统的命令。如果不跟任何参数，\!会直接打开一个独立的shell供用户交互。	-

表 4-21 变量元命令

参数	参数说明
\prompt [TEXT] NAME	用于提示用户输入一个值，并将其存储在一个gsql变量中。TEXT表示提示用户输入的文本，NAME表示要将输入值存储的gsql变量。示例如下： gaussdb=# \prompt 'Enter area name: ' area Enter area name: Asia gaussdb=# SELECT * FROM HR.areaS WHERE area_name = :'area'; area_id | area_name ---------+----------- 3 | Asia (1 row)
\set [NAME [VALUE]]	设置内部变量NAME为VALUE，如果给出了多于一个值，变量取值为所有值的连接结果。如果没有给出第二个参数，只设置变量不设置值。 有一些常用变量被gsql特殊对待，它们是一些选项设置，通常所有特殊对待的变量都是由大写字母组成（可能还有数字和下划线）。 表4-22是所有被特殊对待的变量列表。
\unset NAME	不设置（或删除）gsql变量名。

表 4-22 \set 常用命令

名称	命令说明	取值范围
\set VERBOSITY value	该选项用于控制错误报告的冗余行。	• default：返回严重且主要的错误文本及其位置，还包括详细的错误细节、错误提示（可能会跨越多行）。 • verbose：返回所有的错误信息。 • terse：仅返回严重且主要的错误文本以及文本位置（一般适合于单行错误信息）。

版权所有 © 华为云计算技术有限公司

名称	命令说明	取值范围
\set ON_ERROR_STOP value	如果设置了这个变量，脚本处理将马上停止。如果该脚本是从另外一个脚本调用的，另外一个脚本也会按同样的方式停止。如果最外层的脚本不是从一次交互的gsql会话中调用的而是用-f选项调用的，gsql将返回错误代码3，以示这个情况与致命错误条件的区别（错误代码为1）。	• on：命令执行错误时会立即停止，在交互模式下，gsql会立即返回已执行命令的结果。 • off：命令执行错误时将会跳过错误继续执行。
\set AUTOCOMMIT [on | off]	控制当前gsql连接的自动提交行为。默认情况下，gsql连接处于自动提交模式，每个单独的语句都被隐式提交。如果基于性能或者其他方面考虑，关闭自动提交时，需要用户自己显式发出COMMIT命令来保证事务的提交。例如，在指定的业务SQL执行完之后发送COMMIT语句显式提交，特别是gsql客户端退出之前务必保证所有的事务已经提交。 **说明** gsql默认使用自动提交模式，若关闭自动提交，将会导致后面执行的语句都被隐式事务包裹，数据库中不支持在事务中执行的语句不能在此模式下执行。	• on：表示打开自动提交。 • off：表示关闭自动提交。

表 4-23 全密态元命令

参数	参数说明
\send_token	全密态功能，传输密钥到服务端缓存，在开启内存解密逃生通道的情况下使用。
\st	全密态功能，传输密钥到服务端缓存，在开启内存解密逃生通道的情况下使用。
\clear_token	全密态功能，销毁服务端缓存的密钥，在开启内存解密逃生通道的情况下使用。
\ct	全密态功能，销毁服务端缓存的密钥，在开启内存解密逃生通道的情况下使用。
\key_info KEY_INFO	在全密态数据库特性中，设置访问外部密钥管理者的参数。

版权所有 © 华为云计算技术有限公司

说明

- 分布式暂不支持全密态内存解密逃生通道。
- M-Compatibility模式暂不支持全密态数据库。

表 4-24 常用简易运维元命令

简易运维命令	参数解释	实际执行的SQL
\view stmt_track {unique_sql_id}	stmt_track：记录指定SQL的L2级别的FULLSQL信息。 unique_sql_id：需要记录的SQL的unique_sql_id。	SELECT * FROM pg_catalog.dynamic_func_control('LOCAL', 'STMT', 'TRACK', '{"{unique_sql_id}", "L2"}');
\view stmt_untrack {unique_sql_id}	stmt_untrack：取消记录指定SQL的L2级别的FULLSQL信息。 unique_sql_id：需要取消记录的SQL的unique_sql_id。	SELECT * FROM pg_catalog.dynamic_func_control('LOCAL', 'STMT', 'UNTRACK', '{"{unique_sql_id}"}');
\view stmt_list	stmt_list：查询记录列表。	SELECT * FROM pg_catalog.dynamic_func_control('LOCAL', 'STMT', 'LIST', '{}');
\view stmt_clean	stmt_clean：清理记录列表。	SELECT * FROM pg_catalog.dynamic_func_control('LOCAL', 'STMT', 'CLEAN', '{}');
\view explain_and_stack {pid}	explain_and_stack：查询指定会话的运行时计划及堆栈。 pid：指定会话的pid。	SELECT pg_catalog.gs_get_explain(pid), pg_catalog.gs_stack(pid) FROM pg_catalog.pgxc_stat_activity WHERE pid = {pid};
\view slow_sql {top_n}	slow_sql：查询实时慢SQL情况。 top_n：查询的行数。	SELECT pid, sessionid, pg_catalog.SUBSTRING(query, 0, 100), state, usename, pg_catalog.NOW() - query_start AS elp FROM pg_catalog.pgxc_stat_activity WHERE state != 'idle' AND usename NOT IN ('rdsAdmin', 'rdsMetric', 'rdsBackup', 'rdsRepl', 'dbmind_monitor_agent', 'dbmind_monitor') ORDER BY elp DESC LIMIT {top_n};

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\view slow_trans {top_n}	slow_trans：查询实时长事务情况。 top_n：查询的行数。	SELECT pid, sessionid, pg_catalog.SUBSTRING(query, 0, 100), state, usename, pg_catalog.NOW() - xact_start AS elp FROM pg_catalog.pgxc_stat_activity WHERE state != 'idle' AND usename NOT IN ('rdsAdmin', 'rdsMetric', 'rdsBackup', 'rdsRepl', 'dbmind_monitor_agent', 'dbmind_monitor') ORDER BY elp DESC LIMIT {top_n};
\view wait_event	wait_event（不带参数）：查询当前语句和等待事件。	SELECT a.pid, a.sessionid, a.datname, a.usename, a.client_addr, a.xact_start, a.query_start, (pg_catalog.NOW() - a.query_start)::text "query_runtime", w.wait_status, w.wait_event, w.locktag, w.lockmode, w.block_sessionid, a.query FROM pg_catalog.pgxc_stat_activity a JOIN pg_catalog.pgxc_thread_wait_status w ON a.sessionid = w.sessionid WHERE a.pid <> pg_catalog.pg_backend_pid() AND a.state = 'active' AND a.client_addr IS NOT NULL ORDER BY query_runtime DESC;
\view wait_event {unique_sql_id} {start_time} {end_time}	wait_event（带有三个参数）：查询指定慢SQL的历史等待事件。 unique_sql_id：要查询的慢SQL的unique_sql_id。 start_time：开始时间。如'2025-01-01 00:00'。 end_time：结束时间。如'2025-01-01 00:00'。	SELECT *, pg_catalog.statement_detail_decode(details, 'plaintext', true) FROM dbe_perf.statement_history WHERE unique_query_id = {unique_sql_id} AND start_time > {start_time} AND start_time < {end_time} ORDER BY db_time DESC;
\view thr_wait_event	thr_wait_event（不带参数）：查询实时整体等待事件。	SELECT wait_status, wait_event, pg_catalog.COUNT(*) FROM pg_catalog.pgxc_thread_wait_status GROUP BY 1, 2 ORDER BY 3 DESC;

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\view thr_wait_event {sessionid}	thr_wait_event（带有一个参数）：查询指定语句的等待事件。 sessionid：指定的sessionid。	SELECT * FROM pg_catalog.pgxc_thread_wait_status WHERE sessionid = {sessionid};
\view gs_asp {start_time} {end_time}	gs_asp：查询指定时间段的整体等待事件 start_time：开始时间。如'2025-01-01 00:00'。 end_time：结束时间。如'2025-12-31 23:59'。	SELECT wait_status, event, pg_catalog.COUNT(*) FROM pg_catalog.gs_asp WHERE sample_time BETWEEN {start_time} AND {end_time} GROUP BY 1, 2 ORDER BY 3 DESC;
\view pg_stats {table_name} {att_name}	pg_stats：查询指定表的统计信息分布。 table_name：表名。 att_name：列名。	SELECT * FROM pg_catalog.pg_stats WHERE tablename = '{table_name}' AND attname = '{att_name}';
\view autovac {table_name}	autovac：查询统计信息更新阈值。 table_name：表名。	SELECT * FROM pg_catalog.pg_autovac_status('{table_name}'::regclass);
\view thr_pool	thr_pool：查询线程池使用情况。	SELECT node_name, worker_info FROM dbe_perf.global_threadpool_status;
\view failed_proc {proc_name}	failed_proc：查询失效的视图或存储过程。 proc_name：失效的函数。	SELECT re.ev_class, cl.relname FROM pg_catalog.pg_rewrite re, pg_catalog.pg_class cl WHERE re.ev_action LIKE '%{proc_name}%' AND cl.oid = re.ev_class;

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\view failed_proc_pkg	failed_proc_pkg：查询失效存储过程对应的package。	SELECT DISTINCT object_type, pg_catalog.pg_get_userbyid(creator) AS owner, CASE object_type WHEN 'P' THEN (SELECT DISTINCT pg_catalog.UPPER(pkgname) FROM pg_catalog.gs_package WHERE oid = (SELECT propackageid FROM pg_catalog.pg_proc WHERE oid = object_oid)) WHEN 'B' THEN (SELECT DISTINCT pg_catalog.UPPER(pkgname) FROM pg_catalog.gs_package WHERE oid = object_oid) WHEN 'S' THEN (SELECT DISTINCT pg_catalog.UPPER(pkgname) FROM pg_catalog.gs_package WHERE oid = object_oid) END package_name FROM pg_catalog.pg_object WHERE valid = 'f' AND object_type IN ('P', 'B', 'S');
\view failed_index	failed_index：查询不可用的索引。	SELECT a.schemaname, a.relname, a.indexrelname, b.indisusable FROM pg_catalog.pg_stat_user_indexes a, pg_catalog.pg_index b WHERE a.indexrelid = b.indexrelid AND b.indisusable = 'f';
\view table_size	table_size：查询当前库下所有表的大小，按降序输出。	SELECT schemaname, relname, pg_catalog.pg_total_relation_size(pg_catalog.CONCAT(schemaname, '.', relname)) / 1024 / 1024 / 1024 "GB" FROM pg_catalog.pg_stat_user_tables ORDER BY 3 DESC;

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\view dead_tup {top_n}	dead_tup：查询死元组的数量。 top_n：查询的行数。	SELECT relname, n_live_tup, n_dead_tup, (n_dead_tup / (n_live_tup + 1)) AS rating, last_vacuum, last_autovacuum, last_analyze, last_autoanalyze FROM pg_catalog.pg_stat_user_tables ORDER BY rating DESC, n_dead_tup DESC LIMIT {top_n};
\view table_size {table_name}	table_size：查询指定表的大小。 table_name：表名。	SELECT * FROM pg_catalog.pg_size_pretty(pg_catalog.pg_total_relation_size('{table_name}'::regclass));
\view login {start_time} {end_time} {ip} {port}	login：查询指定IP和端口的登录登出信息。 start_time：开始时间。如'2024-01-01 00:00:00'。 end_time：结束时间。如'2025-12-31 23:59:59'。 ip：IP地址。 port：端口。	SELECT *, pg_catalog.SUBSTRING(client_conninfo::varchar, pg_catalog.POSITION(client_conninfo::varchar, '@'::varchar) + 1, 15) AS ip FROM pg_catalog.pg_query_audit({start_time}, {end_time}) WHERE ip = '{ip}' AND remote_port = '{port}';
\view mem	mem：查询当前内存使用信息。	SELECT * FROM pg_catalog.pg_total_memory_detail;
\view thr_mem {top_n}	thr_mem（带有一个参数）：查询数据库所有线程的内存上下文使用情况。 top_n：查询的行数。	SELECT contextname, pg_catalog.SUM(totalsize) / 1024 / 1024 "totalsize", pg_catalog.SUM(freesize) / 1024 / 1024 "freesize", pg_catalog.COUNT(*) "sum" FROM pg_catalog.pv_thread_memory_context GROUP BY contextname ORDER BY sum DESC LIMIT {top_n};
\view thr_mem {context_name} {top_n}	thr_mem（带有两个参数）：查询使用内存最多的线程的线程id。 context_name：内存上下文的名称。 top_n：查询的行数。	SELECT * FROM pg_catalog.pv_thread_memory_detail() WHERE contextname = '{context_name}' ORDER BY totalsize DESC LIMIT {top_n};

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\view sess_mem {top_n}	sess_mem：查询数据库所有session的内存上下文使用情况。 top_n：查询的行数。	SELECT contextname, pg_catalog.SUM(totalsize) / 1024 / 1024 "totalsize", pg_catalog.SUM(freesize) / 1024 / 1024 "freesize", pg_catalog.COUNT(*) "sum" FROM pg_catalog.pv_session_memory_context GROUP BY contextname ORDER BY sum DESC LIMIT {top_n};
\view thr_mem_detail {tid} {context_name}	thr_mem_detail：查询指定线程的内存使用情况。 tid：线程id。 context_name：内存上下文的名称。	SELECT * FROM pg_catalog.gs_get_thread_memctx_detail({tid}, '{context_name}');
\view sess_mem_tp {context_name}	sess_mem_tp：查询指定内存上下文的内存使用情况。 context_name：内存上下文的名称。	SELECT * FROM pg_catalog.gs_get_session_memctx_detail('{context_name}');
\view shared_mem {top_n}	shared_mem：查询数据库全局共享内存上下文的使用情况。 top_n：查询的行数。	SELECT contextname, pg_catalog.SUM(totalsize) / 1024 / 1024 "totalsize MB", pg_catalog.SUM(freesize) / 1024 / 1024 "freesize MB", pg_catalog.COUNT(*) "sum" FROM pg_catalog.pg_shared_memory_detail GROUP BY contextname ORDER BY sum DESC LIMIT {top_n};
\view sess_mem_detail {top_n}	sess_mem_detail：查询session级内存上下文的使用情况。 top_n：查询的行数。	SELECT contextname, pg_catalog.SUM(totalsize) / 1024 / 1024 "totalsize MB", pg_catalog.COUNT(1) "count" FROM pg_catalog.pv_session_memory_detail() GROUP BY contextname ORDER BY 2 DESC LIMIT {top_n};

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\view sess_mem_detail_tp {top_n}	sess_mem_detail_tp：查询session级内存上下文的使用情况(仅在线程池模式开启时生效)。 top_n：查询的行数。	SELECT sessid, pg_catalog.SUM(totalsize) / 1024 / 1024 "totalsize MB", pg_catalog.COUNT(1) "count" FROM pg_catalog.pv_session_memory_context GROUP BY sessid ORDER BY 2 DESC LIMIT {top_n};
\view thr_mem_pv {top_n}	thr_mem_pv：查询线程级内存上下文的使用情况。 top_n：查询的行数。	SELECT contextname, pg_catalog.SUM(totalsize) / 1024 / 1024 "totalsize", pg_catalog.SUM(freesize) / 1024 / 1024 "freesize", pg_catalog.COUNT(1) "count" FROM pg_catalog.pv_thread_memory_detail() GROUP BY 1 ORDER BY 2 DESC LIMIT {top_n};
\view total_mem {top_n}	total_mem：查询当前数据库节点的内存使用情况。 top_n：查询的行数。	SELECT * FROM pg_catalog.pg_total_memory_detail LIMIT {top_n};
\view avg_cpu {top_n}	avg_cpu：查询SQL语句的平均CPU时间，按降序输出。 top_n：查询的行数。	SELECT unique_sql_id, cpu_time / (n_calls) "average_cpu_time" FROM dbe_perf.statement WHERE n_calls != 0 ORDER BY 2 DESC LIMIT {top_n};
\om create_abort_sql_patch {patch_name} {unique_sql_id}	create_abort_sql_patch：创建避险SQL PATCH。 patch_name：SQL PATCH名。 unique_sql_id：指定SQL的unique_sql_id。	SELECT * FROM dbe_sql_util.create_abort_sql_patch('{patch_name}', {unique_sql_id});

版权所有 © 华为云计算技术有限公司

简易运维命令	参数解释	实际执行的SQL
\om kill_sess_thr {username} {database} {context_name}	kill_sess_thr：根据线程内存占用查杀内存占用高的SQL语句。 username：用户名。 database：数据库名。 context_name：内存上下文的名称。	EXECUTE DIRECT ON COORDINATORS 'SELECT pg_catalog.pg_terminate_session(pid, sessionid), pid, sessionid FROM pg_catalog.pgxc_stat_activity WHERE usename = ''{username}'' AND datname != ''postgres'' AND datname = ''{database}'' AND pid IN (SELECT tid FROM pg_catalog.pv_thread_memory_context WHERE contextname = ''{context_name}'') AND state = ''idle'';';
\om kill_sess_sess {username} {database} {context_name}	kill_sess_sess：根据session内存占用查杀内存占用高的SQL语句。 username：用户名。 database：数据库名。 context_name：内存上下文的名称。	EXECUTE DIRECT ON COORDINATORS 'SELECT pg_catalog.pg_terminate_session(pid, sessionid), pid, sessionid FROM pg_catalog.pgxc_stat_activity WHERE usename = ''{usename}'' AND datname != ''postgres'' AND datname = ''{database}'' AND pid IN (SELECT threadid FROM pg_catalog.pv_session_memory_context WHERE contextname = ''{context_name}'');';

示例

示例一

获取帮助信息。

```
--连接数据库。（根据实际需要替换数据库名字和端口号）
gsql -d postgres -p 8000
gsql ((GaussDB Kernel XXX.X.XXX build f521c606) compiled at 2021-09-16 14:55:22 last mr 6385 release)
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

--查看GaussDB支持的所有SQL语句。
gaussdb=#  \h
Available help:
  ABORT
  ALTER AGGREGATE
 ... ...

--查看DROP TABLE语法的参数。
gaussdb=#  \h DROP TABLE
Command:     DROP TABLE
Description: remove a table
Syntax:
```


版权所有 © 华为云计算技术有限公司

```
DROP TABLE [ IF EXISTS ]
    { [schema.]table_name } [, ...] [ CASCADE | RESTRICT ] [ PURGE ];

--查看gsql支持的元命令。
gaussdb=#  \?
General
  \copyright             show GaussDB Kernel usage and distribution terms
  \g [FILE] or ;         execute query (and send results to file or |pipe)
... ...
```

示例二

使用元命令。

```
--连接数据库。（根据实际需要替换数据库名字和端口号）
gsql -d postgres -p 8000
gsql ((GaussDB Kernel XXX.X.XXX build f521c606) compiled at 2021-09-16 14:55:22 last mr 6385 release)
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

--创建表。
gaussdb=# CREATE SCHEMA HR;
CREATE SCHEMA
gaussdb=# CREATE TABLE HR.areaS(area_ID NUMBER,area_NAME VARCHAR2(25));
NOTICE:  The 'DISTRIBUTE BY' clause is not specified. Using 'area_id' as the distribution column by default.
HINT:  Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
CREATE TABLE

--查看表的定义。
gaussdb=#  \d HR.areaS
               Table "hr.areas"
  Column   |          Type          | Modifiers
-----------+------------------------+-----------
 area_id   | numeric                |
 area_name | character varying(25) |


--向HR.areaS表插入四行数据。
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (1, 'Europe');
INSERT 0 1
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (2, 'Americas');
INSERT 0 1
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (3, 'Asia');
INSERT 0 1
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (4, 'Middle East and Africa');
INSERT 0 1

--查看表。
gaussdb=# SELECT * FROM HR.areaS;
area_id |       area_name
---------+------------------------
       4 | Middle East and Africa
       1 | Europe
       2 | Americas
       3 | Asia
(4 rows)

--用\pset命令以不同的方式显示表。
gaussdb=# \pset border 2
Border style is 2.

gaussdb=# SELECT * FROM HR.areaS;
+---------+------------------------+
| area_id |       area_name        |
+---------+------------------------+
|       4 | Middle East and Africa |
|       1 | Europe                 |
|       2 | Americas               |
|       3 | Asia                   |
+---------+------------------------+
```

 版权所有 © 华为云计算技术有限公司

```
(4 rows)

gaussdb=# \pset border 0
Border style is 0.

gaussdb=# SELECT * FROM HR.areaS;
area_id       area_name
------- ----------------------
      4 Middle East and Africa
      1 Europe
      2 Americas
      3 Asia
(4 rows)

--使用元命令。
gaussdb=# \a \t \x
Output format is unaligned.
Showing only tuples.
Expanded display is on.

gaussdb=# SELECT * FROM HR.areaS;
area_id|4
area_name|Middle East and Africa

area_id|1
area_name|Europe

area_id|2
area_name|Americas

area_id|3
area_name|Asia

gaussdb=# \a \t \x
Output format is aligned.
Tuples only is off.
Expanded display is off.

gaussdb=# SELECT * FROM HR.areaS;
area_id       area_name
------- ----------------------
      4 Middle East and Africa
      1 Europe
      2 Americas
      3 Asia
(4 rows)

gaussdb=# DROP TABLE HR.areaS;
DROP TABLE

gaussdb=# DROP SCHEMA HR;
DROP SCHEMA
```

4.2.1.3 FAQ

连接性能问题

- 问题：开启log_hostname，但是配置错误的DNS导致的连接性能问题。

  解决：

  连接数据库，通过“show log_hostname”语句，检查数据库中是否开启log_hostname参数。

  如果开启了相关参数，数据库内核通过DNS反查客户端所在机器的主机名。如果数据库CN所在的主机配置了不正确的/不可达的DNS服务器，会导致数据库建立连接过程较慢。此参数的更多信息详见GUC参数log_hostname。

版权所有 © 华为云计算技术有限公司

- 问题：数据库内核执行初始化语句较慢导致的性能问题。

  解决：

  此种情况定位较难，可以尝试使用Linux命令strace。

  ```
  strace gsql -U MyUserName -d gaussdb -h 127.0.0.1 -p 23508 -r -c '\q'
  Password for MyUserName:
  ```

  在屏幕上打印出数据库的连接过程。比如，如果较长时间停留在下面的操作上，便可以确定是数据库执行“SELECT VERSION()”语句较慢。

  ```
  sendto(3, "Q\0\0\0\25SELECT VERSION()\0", 22, MSG_NOSIGNAL, NULL, 0) = 22
  poll([{fd=3, events=POLLIN|POLLERR}], 1, -1) = 1 ([{fd=3, revents=POLLIN}])
  ```

  在连接数据库后，便可以通过执行“explain performance select version()”语句来确定初始化语句执行较慢的原因。更多信息，详见《SQL调优指南》中“SQL执行计划介绍”章节。

  另外还有一种场景不太常见：由于数据库CN所在机器的磁盘满或故障，此时查询等受影响，无法进行用户认证，导致连接过程挂起。解决此问题的方法是清理数据库CN的数据盘空间。

- 问题：TCP连接创建较慢问题。

  解决：

  此问题可以参考上面的初始化语句较慢问题的排查方法，通过strace侦听，如果长时间停留在：

  ```
  connect(3, {sa_family=AF_FILE, path="/home/test/tmp/gaussdb_llt1/.s.PGSQL.61052"}, 110) = 0
  ```

  或者

  ```
  connect(3, {sa_family=AF_INET, sin_port=htons(61052), sin_addr=inet_addr("127.0.0.1")}, 16) = -1
  EINPROGRESS (Operation now in progress)
  ```

  说明客户端与数据库端建立物理连接过慢，此时应当检查网络是否存在不稳定、吞吐量太大的问题。

- 问题：资源负载满导致连接较慢的问题。

  解决：

  当CPU、内存、I/O负载中的任意一项接近100%时，会出现gsql连接慢的现象。

  - 通过top命令等确认CPU使用率；通过free命令确认内存使用情况；通过iostat命令确认I/O负载；还可以通过cm_agent中的监控日志，以及数据库运维平台中的监测记录进行检查。
  - 针对短时间内大量慢查询导致的峰值负载场景，可通过（数据库服务器的端口号+1）端口连接，查询pg_stat_activity视图；针对慢查询，可以使用系统函数pg_terminate_backend进行查杀会话。
  - 针对业务量长期超负载情况（即无明显慢查询，或慢查询查杀后但新的查询依然会变成慢查询），应考虑降低业务负载、增加数据库资源的方式进行优化。

创建连接故障

- 问题：gsql: could not connect to server: No route to host

  解决：

  此问题一般是因为指定了不可达的地址或者端口。请检查-h参数与-p参数是否添加正确。

- 问题：gsql: FATAL: Invalid username/password,login denied.

  解决：

  此问题一般是因为输入了错误的用户名和密码，请联系数据库管理员，确认用户名和密码的正确性。

版权所有 © 华为云计算技术有限公司

- 问题：gsql: FATAL: Forbid remote connection with trust method!

  解决：

  由于安全问题，数据库禁止远程登录时使用trust模式。这时需要修改gs_hba.conf里的连接认证信息。请联系管理员处理。

  说明

  请不要修改gs_hba.conf中数据库集群主机的相关设置，否则可能导致数据库功能故障。建议业务应用部署在数据库集群之外，而非集群内部。

- 问题：在CN所在的主机连接数据库，添加“-h 127.0.0.1”可以连接，去掉后无法连接。

  解决：

  通过执行SQL语句“show unix_socket_directory”检查数据库CN使用的Unix套接字目录，是否与shell中的环境变量$PGHOST一致。如果检查结果不一致，那么修改PGHOST环境变量到GUC参数unix_socket_directory指向的目录。

  关于unix_socket_directory的更多信息，请联系管理员获取。

- 问题：The "libpq.so" loaded mismatch the version of gsql, please check it.

  解决：

  此问题是由于环境中使用的libpq.so的版本与gsql的版本不匹配导致的，请通过“ldd gsql”命令确认当前加载的libpq.so的版本，并通过修改LD_LIBRARY_PATH环境变量来加载正确的libpq.so。

- 问题：gsql: symbol lookup error: xxx/gsql: undefined symbol: libpqVersionString

  解决：

  此问题是由于环境中使用的libpq.so的版本与gsql的版本不匹配导致的（也有可能是环境中存在PG的libpq.so），请通过“ldd gsql”命令确认当前加载的libpq.so的版本，并通过修改LD_LIBRARY_PATH环境变量来加载正确的libpq.so。

- 问题：gsql: connect to server failed: Connection timed out

  Is the server running on host "xx.xxx.xxx.xxx" and accepting TCP/IP connections on port xxxx?

  解决：

  此问题是由于网络连接故障。请检查客户端与数据库服务器间的网络连接。如果发现从客户端无法PING到数据库服务器端，则说明网络连接出现故障。请联系网络管理人员排查解决。

  ```
  ping -c 4 10.10.10.1
  PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
  From 10.10.10.1: icmp_seq=2 Destination Host Unreachable
  From 10.10.10.1 icmp_seq=2 Destination Host Unreachable
  From 10.10.10.1 icmp_seq=3 Destination Host Unreachable
  From 10.10.10.1 icmp_seq=4 Destination Host Unreachable
  --- 10.10.10.1 ping statistics ---
  4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 2999ms
  ```

- 问题：gsql: FATAL: permission denied for database "gaussdb"

  DETAIL: User does not have CONNECT privilege.

  解决：

  此问题是由于用户不具备访问该数据库的权限，可以使用如下方法解决。

  a. 使用管理员用户连接数据库。

     ```
     gsql -d gaussdb -U dbadmin -p 8000
     ```


版权所有 © 华为云计算技术有限公司

b. 赋予该用户访问数据库的权限。

```
GRANT CONNECT ON DATABASE gaussdb TO user1;
```

另外，许多常见的错误操作也可能导致用户无法连接到数据库。如用户连接的数据库不存在，用户名或密码输入错误等。这些错误操作在客户端工具会有相应的提示信息。

```
gsql -d gaussdb -p 8000
gsql: FATAL:  database "gaussdb" does not exist

gsql -d gaussdb -U user1 -p 8000
Password for user user1:
gsql: FATAL:  Invalid username/password,login denied.
```

- 问题：gsql: FATAL: sorry, too many clients already，active/non-active: 197/3.

  解决：

  此问题是由于系统连接数量超过了最大连接数量。请联系数据库DBA进行会话连接数管理，释放无用会话。

  关于查看用户会话连接数的方法如**表4-25**。

  会话状态可以在视图PG_STAT_ACTIVITY中查看。无用会话可以使用函数pg_terminate_backend进行释放。

  ```
  SELECT datid,pid,state FROM pg_stat_activity;
   datid |       pid       | state
  -------+-----------------+--------
   13205 | 139834762094352 | active
   13205 | 139834759993104 | idle
  (2 rows)
  ```

  其中pid的值即为该会话的线程ID。根据线程ID结束会话。

  ```
  SELECT PG_TERMINATE_BACKEND(139834759993104);
  ```

  显示类似如下信息，表示结束会话成功。

  ```
  PG_TERMINATE_BACKEND
  ----------------------
   t
  (1 row)
  ```

表 4-25 查看会话连接数

描述	命令

查看指定用户的会话连接数上限。	执行如下命令查看连接到指定用户user1的会话连接数上限。其中-1表示没有对用户user1设置连接数的限制。
	`gaussdb=# SELECT ROLNAME,ROLCONNLIMIT FROM PG_ROLES WHERE ROLNAME='user1';`
	` rolname | rolconnlimit`
	`---------+--------------`
	` user1 |  -1`
	`(1 row)`

查看指定用户已使用的会话连接数。	执行如下命令查看指定用户user1已使用的会话连接数。其中，1表示user1已使用的会话连接数。
	`gaussdb=# SELECT COUNT(*) FROM dv_sessions WHERE USERNAME='user1';`
	` count`
	`-------`
	`  1`
	`(1 row)`

 版权所有 © 华为云计算技术有限公司

描述	命令
查看指定数据库的会话连接数上限。	执行如下命令查看连接到指定数据库gaussdb的会话连接数上限。其中-1表示没有对数据库gaussdb设置连接数的限制。 gaussdb=# SELECT DATNAME,DATCONNLIMIT FROM PG_DATABASE WHERE DATNAME='gaussdb'; datname | datconnlimit ----------+-------------- gaussdb | -1 (1 row)
查看指定数据库已使用的会话连接数。	执行如下命令查看指定数据库gaussdb上已使用的会话连接数。其中，1表示数据库gaussdb上已使用的会话连接数。 gaussdb=# SELECT COUNT(*) FROM PG_STAT_ACTIVITY WHERE DATNAME='gaussdb'; count ------- 1 (1 row)
查看所有用户已使用会话连接数。	执行如下命令查看所有用户已使用的会话连接数。 gaussdb=# SELECT COUNT(*) FROM dv_sessions; count ------- 10 (1 row)

- 问题：gsql: wait xxx.xxx.xxx.xxx:xxxx timeout expired

  解决：

  gsql在向数据库发起连接的时候，会有5分钟超时机制，如果在这个超时时间内，数据库未能正常的对客户端请求进行校验和身份认证，那么gsql会退出当前会话的连接过程，并报出该错误。

  一般来说，此问题是由于连接时使用的-h参数及-p参数指定的连接主机及端口有误（即错误信息中的xxx部分），导致通信故障；极少数情况是网络故障导致。要排除此问题，请检查数据库的主机名及端口是否正确。

- 问题：gsql: could not receive data from server: Connection reset by peer.

  解决：

  请检查CN日志中的如下类似日志“ FATAL: cipher file "/data/coordinator/server.key.cipher" has group or world access”，一般是由于数据目录或部分关键文件的权限被误操作篡改导致。请参照其他正常实例下的相关文件权限修改。

- 问题：gsql: FATAL: GSS authentication method is not allowed because XXXX user password is not disabled.

  解决：

  表示目标CN的gs_hba.conf里，配置了当前客户端IP使用“gss”方式来做认证，该认证算法不支持用作客户端的身份认证，请修改到“sha256”后再试。具体操作请联系管理员处理。

版权所有 © 华为云计算技术有限公司

说明

- 请不要修改gs_hba.conf中数据库集群主机的相关设置，否则可能导致数据库功能故障。
- 建议业务应用部署在数据库集群之外，而非集群内部。

其他故障

- 问题：出现因“总线错误”（Bus error）导致的core dump或异常退出。

  解决：

  一般情况下出现此种问题，是进程运行过程中加载的共享动态库（在Linux为.so文件）出现变化，或者进程二进制文件本身出现变化，导致操作系统加载机器的执行码或者加载依赖库的入口发生变化，操作系统出于保护目的将进程终止，产生core dump文件。为解决此问题，请重试。同时请尽可能避免在升级等运维操作过程中，在集群内部运行业务程序，避免升级时因替换文件产生此问题。

  说明

  此故障的core dump文件的堆栈可能是dl_main及其子调用，它是操作系统用来初始化进程做共享动态库加载的。如果进程已经初始化，但是共享动态库还未加载完成，严格意义上来说，此时进程并未完全启动。

4.3 数据导入导出工具

4.3.1 数据导入导出工具概览

表 4-26 数据导入导出工具概览

工具名称	工具简介
COPY	数据库数据批量导入导出语法接口，COPY可用于表级数据的导入和导出，以及查询结果集的导出。详细信息请参见**COPY**。
\COPY	gsql客户端中用作表级数据批量导入导出的元命令，它基于COPY xxx FROM stdin/stdout用法，并通过libpq进行了报文封装，目的是能将数据存放在gsql客户端机器上。详细信息请参见**表4-14**。
gs_dump	数据库逻辑备份工具。导出单个数据库下的相关信息，用户可以自定义导出一个数据库下全部或其中的部分对象（模式、表、视图等）的定义及数据。详细信息请参见**gs_dump导出数据库信息**。
gs_dumpall	数据库逻辑备份工具。导出所有数据库下的相关信息，它可以导出实例下所有数据库中的对象定义及数据，包括默认数据库的数据、自定义数据库的数据、以及所有数据库公共的全局对象。详细信息请参见**gs_dumpall导出所有数据库信息**。
gs_restore	数据库逻辑恢复工具。对gs_dump导出生成的文件进行导入的工具。详细信息请参见**gs_restore导入数据**。

版权所有 © 华为云计算技术有限公司

实例级导入导出

对于想要逻辑备份整个数据库实例的用户，推荐使用gs_dumpall工具进行备份，它可以导出实例下所有数据库的数据，包括默认数据库的数据、自定义数据库的数据、以及所有数据库公共的全局对象的定义。

须知

- gs_dumpall不会对数据库运行参数进行备份及恢复。
- gs_dumpall仅支持纯文本格式导出。所以只能使用gsql恢复gs_dumpall导出的转储内容。
- 恢复后，建议在每个数据库上运行ANALYZE，为优化器提供有用的统计数据信息。

备份时推荐以初始用户/具备sysadmin权限的用户执行如下命令

```
nohup gs_dumpall -U root -W ******** -f /data/backup/bkp2.sql -p 8000 > /data/backup/dumpall_backup.log &
```

恢复时推荐以初始用户/具备sysadmin权限的用户执行如下命令

```
nohup gsql -d postgres -p 8000 -U root -W ******** -f /data/backup/bkp2.sql -a > /data/backup/dumpall_restore.log &
```

数据库级导入导出

对于想要逻辑备份单个database的用户，推荐使用gs_dump工具进行备份。gs_dump支持四种不同的使用归档格式，不同的归档格式适用场景如下，用户可以根据自身适用情况选择合适的归档格式。

表 4-27 导出文件格式

格式名称	-F的参数值	说明	建议	对应导入工具
纯文本归档格式	p	纯文本脚本文件包含SQL语句和命令。命令可以由gsql命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，或者需要对导出的sql文件进行修改，推荐纯文本格式。	使用**gsql**工具恢复数据库对象前，可根据需要使用文本编辑器编辑纯文本导出文件。
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选数据库对象。	中型或大型数据库，需要将备份结果输出到单个文件中，推荐自定义归档格式。	使用**gs_restore**可以选择要从自定义归档/目录归档/tar归档导出文件中导入相应的数据库对象。

版权所有 © 华为云计算技术有限公司

格式名称	-F的参数值	说明	建议	对应导入工具
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是包含数据库对象的目录文件，另一类是每个表和blob对象对应的数据文件。	中型或大型数据库，需要将数据库对象与数据分目录存储导出，推荐使用目录归档格式。	
tar归档格式	t	tar归档文件支持从导出文件中恢复所有或所选数据库对象。tar归档格式不支持压缩且对于单个文件大小应小于8GB。	小型数据库，需要将归档结果导出并打包，可以使用tar归档格式。	

须知

- gs_dump不会对所有数据库公共的全局对象（角色、表空间及对应权限）进行备份和恢复。因此恢复时请确保目标库/新实例上的全局对象已经创建。
- gs_dump和gs_restore不支持跨数据库兼容模式的导入导出，请确保源库与目标库的数据库兼容模式及兼容性配置参数保持一致。
- 禁止修改-F c/d/t 格式导出的文件和内容，否则可能无法恢复成功。对于-F p 格式导出的文件，如有需要修改替换，可根据需要谨慎编辑导出文件。
- 恢复后，建议在数据库上运行ANALYZE，为优化器提供有用的统计数据信息。

备份时推荐以初始用户/具备sysadmin权限的用户执行如下命令，源库为my_database，导出时包含数据和对象定义

```
-- 纯文本归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F p -f /data/backup/my_database_backup.sql > /
data/backup/my_database_backup.log &
-- 自定义归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F c -f /data/backup/my_database_backup.dmp > /
data/backup/my_database_backup.log &
-- 目录归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F d -f /data/backup/my_database_backup > /data/
backup/my_database_backup.log &
-- tar归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F t -f /data/backup/my_database_backup.tar > /
data/backup/my_database_backup.log &
```

恢复时需要先创建好与源库属性相同的目标库，且库内没有任何内容

```
-- 通过以下gsql元命令，查看数据库属性信息
\l+
-- 根据查询到的属性信息，创建目标数据库
create database my_database2 encoding='xxxxx' LC_COLLATE='xxxxx' LC_CTYPE ='xxxxx' TEMPLATE=xxx
DBCOMPATIBILITY 'xxx';
```

再以初始用户/具备sysadmin权限的用户执行如下命令进行恢复

版权所有 © 华为云计算技术有限公司

```
-- 纯文本归档格式
nohup gsql -d my_database2 -p 8000 -U root -W ******** -f /data/backup/my_database_backup.sql -a > /
data/backup/my_database_restore.log &
-- 自定义归档格式
nohup gs_restore /data/backup/my_database_backup.dmp -d my_database2 -p 8000 -U root -W ******** -F c
-v > /data/backup/my_database_restore.log &
-- 目录归档格式
nohup gs_restore /data/backup/my_database_backup -d my_database2 -p 8000 -U root -W ******** -F d -v
> /data/backup/my_database_restore.log &
-- tar归档格式
nohup gs_restore /data/backup/my_database_backup.tar -d my_database2 -p 8000 -U root -W ******** -F t -
v > /data/backup/my_database_restore.log &
```

模式级导入导出

对于想要备份单个schema下的用户，推荐使用gs_dump工具并配合-n参数进行备份，可以连续使用多个-n备份多个schema。

备份时推荐以初始用户/具备sysadmin权限的用户执行如下命令，源库为my_database，导出时会包含数据和对象定义。

```
-- 纯文本归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F p -f /data/backup/my_schema_backup.sql -n
my_schema > /data/backup/my_schema_backup.log &
-- 自定义归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F c -f /data/backup/my_schema_backup.dmp -n
my_schema > /data/backup/my_schema_backup.log &
-- 目录归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F d -f /data/backup/my_schema_backup -n
my_schema > /data/backup/my_schema_backup.log &
-- tar归档格式
nohup gs_dump my_database -U root -W ******** -p 8000 -F t -f /data/backup/my_schema_backup.tar -n
my_schema > /data/backup/my_schema_backup.log &
```

恢复时需要先创建好与源库属性相同的目标库，且库内没有目标模式

```
-- 通过以下gsql元命令，查看数据库属性信息
\l+
-- 根据查询到的属性信息，创建目标数据库
create database my_database2 encoding='xxxxx' LC_COLLATE='xxxxx' LC_CTYPE ='xxxxx' TEMPLATE=xxx
DBCOMPATIBILITY 'xxx';
```

再以初始用户/具备sysadmin权限的用户执行如下命令进行恢复

```
-- 纯文本归档格式
nohup gsql -d my_database2 -p 8000 -U root -W ******** -f /data/backup/my_schema_backup.sql -a > /data/
backup/my_schema_restore.log &
-- 自定义归档格式
nohup gs_restore /data/backup/my_database_backup.dmp -d my_database2 -p 8000 -U root -W ******** -F c
-v -n my_schema > /data/backup/my_schema_restore.log &
-- 目录归档格式
nohup gs_restore /data/backup/my_database_backup -d my_database2 -p 8000 -U root -W ******** -F d -v -
n my_schema > /data/backup/my_schema_restore.log &
-- tar归档格式
nohup gs_restore /data/backup/my_database_backup.tar -d my_database2 -p 8000 -U root -W ******** -F t -
v -n my_schema > /data/backup/my_schema_restore.log &
```

表级导入导出

对于表级的导入与导出，能够使用的工具很多，可基于以下场景切入选择合适的工具。

1. 用户需要导出单个表的定义和表内数据到同一个文件中。推荐使用gs_dump工具的纯文本归档，并配合-t参数使用，可以连续使用-t备份多张表。

   备份时推荐以初始用户/具备sysadmin权限的用户执行如下命令，源库为my_database，目标表为my_schema下的my_table。

版权所有 © 华为云计算技术有限公司

```
nohup gs_dump my_database -U root -W ******** -p 8000 -F p -f /data/backup/my_table_backup.sql -t my_schema.my_table > /data/backup/my_table_backup.log &
```

恢复时需要先创建好与源库属性相同的目标库，且库内已有目标模式，没有目标表，再以初始用户/具备sysadmin权限的用户执行如下命令进行恢复

```
nohup gsql -d my_database2 -p 8000 -U root -W ******** -f /data/backup/my_table_backup.sql -a > /data/backup/my_table_restore.log &
```

2. 用户仅需要导出单个表的定义，不需要表内数据。

   推荐使用gs_dump工具的纯文本归档，配合-s参数使用，命令如下

   ```
   nohup gs_dump my_database -U root -W ******** -p 8000 -F p -f /data/backup/my_table_backup.sql -t my_schema.my_table -s > /data/backup/my_table_backup.log &
   ```

   恢复时需要先创建好与源库属性相同的目标库，且库内已有目标模式，没有目标表，再以初始用户/具备sysadmin权限的用户执行如下命令进行恢复

   ```
   nohup gsql -d my_database2 -p 8000 -U root -W ******** -f /data/backup/my_table_backup.sql -a > /data/backup/my_table_restore.log &
   ```

3. 用户需要导出单个表的数据，不需要表定义。

   a. 将数据导出到数据库节点上，推荐直接使用COPY命令

   须知

   - 普通场景下，出于数据库安全考虑，当参数enable_copy_server_files关闭时，只允许初始用户执行COPY FROM FILENAME或COPY TO FILENAME命令，当参数enable_copy_server_files打开时，允许具有SYSADMIN权限的用户或继承了内置角色gs_role_copy_files权限的用户执行，但默认禁止对数据库配置文件，密钥文件，证书文件和审计日志执行COPY FROM FILENAME或COPY TO FILENAME，以防止用户越权查看或修改敏感文件。enable_copy_server_files打开时，系统管理员可以通过guc参数safe_data_path设置普通用户可以导入导出的路径的白名单，未设置此guc参数时候（默认情况），不对普通用户使用的路径进行拦截。
   - 当数据库部署在沙箱内时，COPY指定的文件为沙箱内的路径，而不是沙箱外的路径。

   推荐使用如下命令将数据导出成标准的CSV格式，此文件路径为数据库节点上的路径。

   ```
   COPY my_schema.my_table TO '/data/backup/my_table.csv' CSV;
   ```

   再通过如下命令将数据导入到目标表中。

   ```
   COPY my_schema.my_table2 FROM '/data/backup/my_table.csv' CSV;
   ```

   b. 将数据导出到客户端上，推荐直接使用\COPY元命令

   使用如下命令将数据导出成标准的CSV格式，此文件路径为客户端上的路径。

   ```
   \COPY my_schema.my_table TO '/data/backup/my_table.csv' CSV;
   ```

   再通过如下命令将数据导入到目标表中。

   ```
   \COPY my_schema.my_table2 FROM '/data/backup/my_table.csv' CSV;
   ```

4. 用户需要自定义导入导出时数据的格式，例如分隔符换行符等

   推荐使用COPY或\COPY。

版权所有 © 华为云计算技术有限公司

4.3.2 gs_dump 导出数据库信息

工具介绍

gs_dump是GaussDB用于导出数据库相关信息的工具，用户可以自定义导出一个数据库或其中的对象（模式、表、视图等）。支持导出的数据库可以是默认数据库postgres，也可以是自定义数据库。

- gs_dump支持SSL加密通信。
- gs_dump工具在进行数据导出时，其他用户可以访问集群数据库（读或写）。
- gs_dump工具导出完整一致的数据。例如，T1时刻启动gs_dump导出A数据库，那么导出数据结果将会是T1时刻A数据库的数据状态，T1时刻之后对A数据库的修改不会被导出。
- gs_dump支持将数据库信息导出至纯文本格式的SQL脚本文件或其他归档文件中。
  - 纯文本格式的SQL脚本文件：包含将数据库恢复为其保存时的状态所需的SQL语句。通过**gsql连接数据库**运行该SQL脚本文件，可以恢复数据库。即使在其他主机和其他数据库产品上，只要对SQL脚本文件稍作修改，也可以用来重建数据库。
  - 归档格式文件：包含将数据库恢复为其保存时的状态所需的数据，可以是tar格式、目录归档格式或自定义归档格式，详见**表4-28**。该导出结果必须与**gs_restore导入数据**配合使用来恢复数据库，使用gs_restore工具导入时，系统允许用户选择需要导入的内容，甚至可以在导入之前对等待导入的内容进行排序。
- gs_dump支持四种不同的导出文件格式，通过[-F或者--format=]选项指定，具体如**表4-28**所示。

表 4-28 导出文件格式

格式名称	-F的参数值	说明	建议	对应导入工具
纯文本格式	p	纯文本脚本文件包含SQL语句和命令。命令可以由gsql命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，或者需要对导出的sql文件进行修改，推荐纯文本格式。	使用**gsql连接数据库**工具恢复数据库对象前，可根据需要使用文本编辑器编辑纯文本导出文件。
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选的数据库对象。	中型或大型数据库，推荐自定义归档格式。	这三种归档格式，必须使用**4.7.5-gs_restore**进行对象和数据的导入。gs_restore也可以选择要导入的数据库对象。

版权所有 © 华为云计算技术有限公司

格式名称	-F的参数值	说明	建议	对应导入工具
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是包含数据库对象的目录文件，另一类是每个表和blob对象对应的数据文件。	需要将数据库对象与数据分目录存储导出，推荐使用目录归档格式。	
tar归档格式	t	tar归档文件支持从导出文件中恢复所有或所选的数据库对象。tar归档格式不支持压缩且对于单个文件大小应小于8GB。	需要将归档结果导出并打包，可以使用tar归档格式。	

说明

可以使用gs_dump工具将文件压缩为目录归档或自定义归档导出文件，减少导出文件的大小。生成目录归档或自定义归档导出文件时，默认进行中等级别的压缩。gs_dump程序无法压缩已归档导出文件。

注意事项

- 使用gs_dump前请确保gs_dump版本与数据库版本保持一致，高版本gs_dump不保证完全兼容低版本内核数据。
- gs_dump不适用于库中对象数量（表、视图、索引等）过多的场景。当库中对象数量超过十万级，或者对象间依赖关系过于复杂时，gs_dump导出时间会很长。
- 禁止修改-F c/d/t格式导出的文件和内容，否则可能无法恢复成功。对于-F p格式导出的文件，如有需要，可谨慎编辑导出的文件。
- 为了保证数据一致性和完整性，gs_dump默认会对需要转储的表设置ACCESS SHARE级别锁。如果表在别的事务中设置了冲突的锁，gs_dump会等待锁释放后锁定表。如果无法在指定时间内锁定某个表，转储会失败。用户可以通过指定--lock-wait-timeout选项，自定义等待锁超时时间。锁与锁之间的冲突请参见**LOCK**。
- 不支持加密导出存储过程和函数。
- 对于物化视图，本工具仅支持物化视图定义的导出，在导入后需手动执行REFRESH命令来进行数据恢复。
- gs_dump不会导出物化视图相关的matviewmap_和mlog_前缀开头的表，不会导出带有internal_mask选项的表，不会导出在回收站中的表。
- 对于临时对象，本工具仅支持导出全局临时表。
- 本工具不支持在备机上使用。
- 由于DN上系统表中分布信息不完整，所以在DN使用gs_dump时，不会转储表的分布信息。
- gs_dump导出分区索引时，部分索引分区的属性无法导出，比如索引分区的unusable状态。可以通过查询系统表PG_PARTITION或者查询视图

版权所有 © 华为云计算技术有限公司

ADM_IND_PARTITIONS获取索引分区的具体属性，通过ALTER INDEX命令可以手动设置索引分区属性。

- 对于定时任务，本工具仅支持导出在MYSQL兼容性数据库中，通过高级包创建的非周期性定时任务。
- gs_dump不支持导出自定义Tokenweight分词词典，可以根据报错WARNING: dictionary xx cannot be automatically exported, please create it manually手动创建对应分词词典。
- 普通用户不支持导出DIRECTORY、SYNONYM，若普通用户进行相关导出，会提示“WARNING: xx not dumped because current user is not a superuser”。需使用初始用户或SYSADMIN权限的用户进行导出。导出后执行导入需遵循语法本身的用户权限、操作约束，详情请参见**CREATE DIRECTORY**、**CREATE SYNONYM**。
- 如果源数据库实例的template1中存在本地数据，请将gs_dump的输出恢复到一个真正的空数据库（从template0而非template1复制），否则可能会因为被添加对象的定义被复制，出现错误。创建一个无本地添加的空数据库，需从template0而非template1复制，例如：
  ```
  CREATE DATABASE foo WITH TEMPLATE template0;
  ```
- tar归档形式的单个文件大小不得超过8GB（tar文件格式的固有限制）。tar文档整体大小和任何其他输出格式没有限制，操作系统可能对此有要求。
- 由gs_dump生成的转储文件不包含优化程序用来做执行计划决定的统计数据。因此，建议从某转储文件恢复之后运行ANALYZE以确保最佳效果。转储文件不包含ALTER DATABASE…SET命令、数据库用户和其他完成安装设置，这些设置由gs_dumpall转储。
- gs_dump不导出失效函数，需要导出函数前请确保函数有效，失效的函数会打印告警，提示对应OID的函数失效。
- PACKAGE外的函数参数类型是表的列类型（table.column%type）或视图的列类型（view.column%type），且列类型是基础类型时，导出的函数定义中，参数类型会被转化为对应的基础类型。通过ALTER TABLE MODIFY COLUMN修改表的列类型并执行ALTER FUNCTION COMPILE，或者修改视图查询表的列类型后重建视图并执行ALTER FUNCTION COMPILE，PG_PROC系统表中该函数PROARGSRC字段中参数类型不会随之变化，导出的函数定义中，参数类型依然是对象列类型变更前的基础类型。
- gs_dump当前设计不兼容三权分立，三权分立下仅可通过初始用户使用gs_dump工具。导入的文件也需要通过初始用户执行。
- 当导出对象中包含触发器时，需要加上参数--use-set-session-authorization --all，并通过文本模式导出。否则恢复时无法准确还原触发器的owner信息。
- 使用gs_dump不会转储model对象。
- 使用系统管理员用户执行gs_dump导出时，不会导出初始用户创建的对象，初始用户创建的对象需要使用初始用户进行导出。使用普通用户执行gs_dump导出时，仅会导出owner归属于该普通用户的所有对象。
- 当导出对象中包含依赖PACKAGE内部函数的视图时，在首次导入目标端时会报视图依赖对象找不到的错误，此时需要使用gs_dump --schema-only备份元数据，并再次导入到目标端即可。
- 在使用gs_dump导出高级压缩库级策略定义时，需满足以下条件：必须执行全库导出操作且用户具备系统管理员权限。若上述条件未同时满足，则相关策略将以表、分区或子分区级别的独立策略形式导出，而非作为库级策略处理。
- gs_dump会把sequence的实际状态导出，其在恢复到目标端后，该sequence会继续上一次的状态。

版权所有 © 华为云计算技术有限公司

- 对一些还未支持ustore老版本上导出的备份文件，在支持ustore的版本上恢复时，表会被创建为ustore格式且受ustore规格的限制，若想恢复为astore格式，需要设置enable_default_ustore_table为off。
- 使用gs_dump导出间隔分区表时，只能导出该分区表的range分区部分，不能导出该分区表的interval分区部分。对于该分区表上的分类索引，遵循同样的原则，即只能导出建立在range分区上的分类索引，不能导出interval分区上的分类索引。
- 当ddl_invalid_mode参数设置为invalid后，对函数和包做drop操作时，不会级联删除依赖其的对象，只是失效其依赖关系，即pg_depend系统表中仍然保有一条依赖关系记录；此时使用gs_dump进行备份时，会因为存在依赖关系但是找不到依赖的对象而报错cache lookup failed。需要修复依赖关系（重建依赖对象/删除被依赖对象）后进行导出。
- gs_dump默认会导出失效视图，如果导出的视图中有失效视图会打印提示，在507.0之前版本导入时可能会报错。可指定--ignore-invalid-views选项，不导出失效视图。
- gs_dump不导出rowid系统列上的数据，仅导出rowid作为数据类型时用户列的数据。
- 导出全文检索相关数据对象之后，如果导入的目标库没有对应的全文检索文件，或者导入导出时带上-c选项，则导入会失败。对于这类对象需要在目标库上创建后重试导入，或者导入导出时不指定-c选项。
- gs_dump导出视图时，若视图定义包含SELECT * 且基表在视图创建后发生过列变更，则导出文件中的 * 在目标库导入后会按目标表当前列结构重新展开。
- gs_redis重分布期间，gs_redis会以CANCELABLE模式加锁并主动向持有冲突锁的gs_dump线程发送SIGTERM，可能导致gs_dump导出被强制终止而失败。
- gs_dump导出的对象中包含用户视图时，请确保导出环境与恢复环境中的behavior_compat_options参数关于proc_outparam_override的配置保持一致，否则可能导入失败。
- gs_dump导出结果不包含所有会话参数设置等状态信息（如：behavior_compat_options等），执行语句结果若依赖状态信息，在目标库导入时，需要原库和目标库的会话参数一致，否则可能报错或者数据不一致。

命令格式

```
gs_dump [OPTION]... [DBNAME]
```

参数说明

- DBNAME

  指定要连接的数据库，前面不需要加短或长选项。例如：

  ```
  gs_dump -p port_number testdb -f dump1.sql
  ```

  或者

  ```
  export PGDATABASE=testdb
  gs_dump -p port_number -f dump1.sql
  ```

  DBNAME详细使用方式参见“gsql使用指导”中**表4-7**。

- OPTION

  分为三类，通用参数、转储参数和连接参数。

  通用参数：

  – -f, --file=FILENAME

 版权所有 © 华为云计算技术有限公司

将输出发送至指定文件或目录。如果省略该参数，则使用标准输出。如果输出格式为-F c、-F d或-F t时，必须指定-f参数。如果-f的参数值含有目录，要求当前用户对该目录具有读写权限。

- -F, --format=c|d|t|p

  选择输出格式。格式如下：

  - c|custom：输出一个自定义格式的归档，并且以目录形式输出，以此作为gs_restore的输入信息。该格式是最灵活的输出格式，因为可以手动选择，而且在恢复过程中可以将归档项重新排序。该格式默认状态下会被压缩。
  - d|directory：该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和blob对象对应的数据文件。
  - t|tar：输出一个tar格式的归档，作为gs_restore输入信息。tar格式与目录格式兼容，tar格式归档形式在提取过程中会生成一个有效的目录格式归档形式。tar格式不支持压缩且对于单独表有8GB的大小限制。此外，表数据项的相应排序在恢复过程中不能更改。
  - p|plain：输出一个文本SQL脚本文件（默认）。

- -v, --verbose

  指定verbose模式。该选项将导致gs_dump向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。

- -V, --version

  打印gs_dump版本，然后退出。

- -Z, --compress=0-9

  指定使用的压缩比级别。

  取值范围：0~9

  - 0表示无压缩。
  - 1表示压缩比最小，处理速度最快。
  - 9表示压缩比最大，处理速度最慢。

  说明

  - 针对自定义归档格式，该选项指定单个表数据片段的压缩，默认方式是以中等级别进行压缩。tar归档格式和纯文本格式目前不支持压缩。
  - 对于参数的取值，系统将取遇到的第一个非数字字符之前的部分。如果字符串以非数字字符开头，则取值为0。

- --lock-wait-timeout=TIMEOUT

  请勿在转储刚开始时一直等待以获取共享表锁（一级锁），如果不能在指定时间内锁定表，则选择失败。参数单位为ms（毫秒）。可以以任何符合SET statement_timeout的格式指定超时时间。

  整个导出过程中会一直持有这些锁，不会阻塞DML语句。

- -?, --help

  显示gs_dump命令行参数帮助，然后退出。

- --ignore-invalid-views

  指定该参数时，不会导出失效视图，导出时发现有失效视图会打印提示。

版权所有 © 华为云计算技术有限公司

转储参数：

- -a, --data-only

  只输出数据，不输出模式（数据定义）。转储表数据和序列值。

- -b, --blobs

  该参数为扩展预留接口，不建议使用。

- -c, --clean

  在将创建数据库对象的指令输出到备份文件之前，先将清理（删除）数据库对象的指令输出到备份文件中。如果目标数据库中没有任何对象，gsql或gs_restore工具可能会输出一些提示性的错误信息。

  该选项只对文本格式有意义。针对归档格式，可以在调用gs_restore时指定选项。

  说明

  - 输出到备份文件中的删除数据库对象指令，均为非级联式删除的DROP命令。
  - 连接database1在t1时刻使用-c参数导出表table1到backup01.dmp文件，t2时刻在database1中创建视图view1并依赖table1，t3使用gs_restore的-c参数将backup01.dmp恢复到database1中时，因为会使用非级联删除，在删除table1时会报错“ERROR: table table cannot be dropped because other objects depend on it.”，并在创建table1时报错“ERROR: Relation "table1" already exists.”。此时需要用户在评估影响后，手动对相关依赖对象进行删除，或手动执行级联删除命令。

- -C, --create

  备份文件的内容以创建数据库和连接到该数据库的命令开始。如果命令脚本以这种方式执行，在运行脚本之前连接的是哪一个数据库都没有影响。

  该选项只对文本格式有意义。针对归档格式，可以在调用gs_restore时指定选项。

  说明

  - 在M-Compatibility模式数据库下，不支持使用该选项。必须先在目标实例上创建出M-Compatibility模式数据库，然后在源实例上导出，最后在目标实例连接新创建的M-Compatibility模式数据库导入。
  - 在指定-n、-t或者--include-table-file之后，-C选项会失效。

- -E, --encoding=ENCODING

  以指定的字符集编码创建转储。默认情况下，以数据库编码创建转储。将环境变量“PGCLIENTENCODING”设置为所需的转储编码，也可以得到相同的结果。

  说明

  当指定转储编码存在转码场景，且表中的数据存在非法编码的数据时，导出会报错invalid byte sequence，建议使用gs_dump的-s参数只导出定义，并单独使用COPY打开编码容错进行数据的导出与导入。

- -g, --exclude-guc=GUC_PARAM

  导出的文本文件中，不包括相关guc参数的set命令，目前只支持enable_cluster_resize，指定其他参数不会报错，但不会生效。

- -j, --jobs=NUM

  以指定的并行数进行导出。

 版权所有 © 华为云计算技术有限公司

说明

- 不指定-j时，默认使用串行导出。
- 指定的并行数不允许超过客户端CPU核数的两倍。当客户端运行在容器中时，由于获取到的是主机的CPU数量，可能比容器实际能使用的多，并发度建议由用户自行控制在实际能使用的客户端CPU数量的两倍范围内。表内并行导出通过多进程并发多个事务实现，实际并发度会受限于服务端的线程池模型，也同时会给服务端增加压力，请按照实际情况合理设置并发度，例如并发度不超过客户端空闲CPU数量，同时不超过服务端最大连接数限制等。建议并行数不低于4。
- 并行导出仅支持目录归档格式。
- 并行导出默认使用表内并行导出，即只导出一个表时，也可以对该表的数据拆分成多个导出任务，并行导出到多个文件中。可以通过同时指定--disable-inter-table-parallel参数关闭表内并行，当关闭表内并行导出时，并行导出是表级别的，即仅在导出多个表的表数据时，并行才会生效。
- 表内并行导出后执行并行导入时，同表数据会放到一个事务中，即并行导入效率没有提升。
- 并行导出与--inserts、--column-inserts、--attribute-inserts选项冲突。
- 并行导出与--query选项冲突。
- bucket表不支持表内并行导出。
- 并行导出性能会受到压缩参数-Z的影响，建议合理设置压缩级别（默认中等压缩、或者设置为0即不压缩）。压缩级别越高，客户端CPU占用会越高，并存在客户端CPU抢占导致导出速度降低的情况。
- 并行导出不支持在升级观察期使用。
- 表内并行导出不支持在DN上使用；即仅允许在DN上同时指定-j和--disable-inter-table-parallel。
- 分布式上并行导出存在一定的损耗。并行数较低时，并行优化效果不明显。

– -n, --schema=SCHEMA

只转储与模式名称匹配的模式，此选项包括模式本身和所有它包含的对象。如果该选项没有指定，所有在目标数据库中的非系统模式将会被转储。支持写入多个-n选项来选择多个模式。此外，根据gsql的\d命令所使用的相同规则，模式参数可以被理解为一个pattern，所以多个模式也可以通过在该pattern中写入通配符来选择。使用通配符时，注意给pattern打引号，防止shell扩展通配符。

说明

- 当指定-n时，gs_dump不会转储已选模式所附着的任何其他数据库对象。因此，无法保证某个指定模式的转储结果能够自行成功地储存到一个空数据库中。
- 当指定-n时，非模式对象不会被转储。
- M-Compatibility模式数据库下， 通过CREATE DATABASE带templatem创建的数据库直接通过指定db_name导出数据； 而通过CREATE DATABASE db_name创建的DATABASE与SCHEMA等效，只能通过-n导出数据。
- GaussDB会自动将对象名称中的大写字母转为小写，当模式名称中包含大写字母时，需要添加额外的引号，如：-n '"Sch1"' 或 -n "\"Sch1\""。
- 当指定-n时，不支持通过pattern "*"去匹配导出数据库中的系统SCHEMA。

转储支持多个模式的转储。例如，以下示例中，sch1和sch2都会被转储。

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -n sch1 -n sch2
```

– -N, --exclude-schema=SCHEMA

不转储任何与模式pattern匹配的模式。Pattern将参照针对-n的相同规则来理解。可以通过输入多个-N，不转储与任何pattern匹配的模式。

版权所有 © 华为云计算技术有限公司

当同时输入-n和-N时，会转储与至少一个-n选项匹配、与-N选项不匹配的模式。如果有-N没有-n，则不转储常规转储中与-N匹配的模式。

转储过程支持排除多个模式。例如，以下示例中，sch1和sch2在转储过程中都会被排除。

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -N sch1 -N sch2
```

说明

GaussDB会自动将对象名称中的大写字母转为小写，当模式名称中包含大写字母时，需要添加额外的引号，如：-N '"Sch1"' 或 -N "\"Sch1\""。

- -o, --oids

  转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于以某种方式（例如：外键约束方式）参照了OID列的情况。如果不是以上这种情况，请勿使用该选项。不能与--inserts/--column-inserts选项一起使用。

- -O, --no-owner

  不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs_dump会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的数据库对象的归属。脚本运行时这些语句会失败，除非是由系统管理员（或是拥有脚本中所有对象的同一个用户）触发。通过指定-O，可以生成一个任何用户都能用于恢复的脚本，且该脚本会授予该用户所有对象的权限。

  该选项只对文本格式有意义。针对归档格式，可以在调用gs_restore时指定选项。

- -s, --schema-only

  只转储对象定义（模式），不转储数据。

- -q, --target=VERSION

  指定导出兼容其他版本数据库的文本文件，目前支持v1和v5参数，指定其他参数不会报错，但不会生效。v1参数用于导出GaussDB v5版本数据库的数据为兼容GaussDB v1版本数据库的文本文件。v5参数用于导出GaussDB v5版本数据库的数据为GaussDB v5版本数据库格式的文本文件，减少了导入GaussDB v5版本数据库时可能的报错情况。

  在使用v1参数时，建议同时设置--exclude-guc="enable_cluster_resize"，--exclude-function，--exclude-with等选项，否则导入到GaussDB v1版本数据库时可能报错。

- -S, --sysadmin=NAME

  该参数为扩展预留接口，不建议使用。

- -t, --table=TABLE

  指定转储的表（或视图、或序列、或外表）对象列表，可以使用多个-t选项来选择多个表，也可以使用通配符指定多个表对象。例如，以下示例中，schema1.table1和schema2.table2都会被转储。

  ```
  gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -t schema1.table1 -t schema2.table2
  ```

  当使用通配符指定多个表对象时，注意给pattern打引号，防止shell扩展通配符。

  当使用-t时，-n和-N没有任何效果，因为由-t选择的表的转储不受-n、-N选项的影响。

版权所有 © 华为云计算技术有限公司

说明

- -t参数选项个数必须小于等于100。
- 如果-t参数选项个数大于100，建议使用参数--include-table-file来替换。
- 当指定-t时，gs_dump不会转储已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地储存到一个空数据库中。
- -t tablename只转储在默认搜索路径中可见的表。-t *.tablename转储数据库中所有模式下的tablename表（系统模式下的表除外）。-t schema.table转储特定模式中的表。
- -t tablename不会导出表上的触发器信息。
- 对于表名中包含大写字母的表，在使用-t参数指定导出时需对表名添加\"来导出。例如对于表"abC"，需指定-t \"abC\"；对于表schema."abC"，导出需指定-t schema.\"abC\"。
- -t "" 不匹配任何表。
- 当-t参数中有多个"."分隔符时，实际上仅最后两个分隔符生效，倒数第二个和倒数第一个分隔符之间的名称即模式名，倒数第一个分隔符之后的名称即表名。例如-t schema_a.schema_b.schema_c.table1，实际上会导出模式schema_c的表table1；-t schema_a.schema_b.schema_c.*.table1，实际上会导出所有模式下的表table1。

– --include-table-file=FILENAME

  指定需要dump的表文件。

– -T, --exclude-table=TABLE

  不转储的表（视图、序列、外表）对象列表，可以使用多个-T选项来选择多个表，也可以使用通配符指定多个表对象。

  当同时输入-t和-T时，会转储在-t列表中，而不在-T列表中的表对象。

  例如，以下示例中，table1和table2在转储过程中都会被排除。

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -T table1 -T table2
```

  说明

  对于自增列的序列，-T参数不会生效。

– --exclude-table-file=FILENAME

  指定不需要dump的表文件。

  说明

  同--include-table-file，其内容格式如下：

  schema1.table1

  schema2.table2

  ......

– -x, --no-acl

  防止转储访问权限命令（授权/撤销命令），只影响acl对象，不影响privilege对象。

– --exclude-function

  不导出函数和存储过程。

– --exclude-with

  导出的表定义，末尾不添加WITH(orientation=row,compression=on)等描述。

– --binary-upgrade

 版权所有 © 华为云计算技术有限公司

该参数为扩展预留接口，不建议使用。

说明

M-Compatibility模式数据库不支持该选项。

- --binary-upgrade-usermap="USER1=USER2"

  该参数为扩展预留接口，不建议使用。

- --column-inserts|--attribute-inserts

  以INSERT命令带列名（INSERT INTO表（列、…）值…）方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致失败语句对应的数据丢失，而非整个表内容。

  说明

  M-Compatibility模式数据库不支持该选项。

- --disable-dollar-quoting

  该选项将禁止在函数体前使用美元符号$，并强制使用SQL标准字符串语法对其进行引用。

- --disable-inter-table-parallel

  该选项仅在并行导出时生效，当指定该参数时，并行导出使用表级别的并行。

- --include-alter-table

  导出表删除列的相关步骤。

- --disable-triggers

  该参数为扩展预留接口，不建议使用。

- --exclude-table-data=TABLE

  指定不转储任何匹配pattern的表的数据。依照针对-t的相同规则理解该pattern。

  可多次输入--exclude-table-data来排除匹配任何pattern的表。当用户需要特定表的定义但不需要其中的数据时，该选项很有帮助。

  排除数据库中所有表的数据，参见**-s, --schema-only**。

- --include-query-file=FILENAME

  导出表内数据时通过该参数在文件内批量配置谓词过滤条件，通过增加过滤条件，只导出满足过滤条件的表数据。

  FILENAME中的内容支持如下格式，每个谓词条件之间以'\n'换行分隔：

  i. [schema_name.][table_name:] query_clause

  ii. query=[schema_name.][table_name:] query_clause

  说明

  - 文件中以'#'开头的行会被当做注释行。
  - --include-query-file和--query可以同时使用，但是对于同一张表只能指定一个谓词条件。
  - 谓词条件其他规则请参见--query参数说明。

- --inserts

  发出INSERT命令（而非COPY命令）时转储数据，会导致恢复缓慢。

 版权所有 © 华为云计算技术有限公司

但是，由于该选项会针对每行生成一个独立分开的命令，所以，重新加载某行时出现的错误只会导致对应的一行数据丢失，而非整个表内容。需要注意的是，如果重排列顺序，可能会导致恢复失败。列顺序改变时，--column-inserts选项不受影响。

说明

M-Compatibility模式数据库不支持该选项。

- --log-level=LOG|DEBUG|DEBUG1|DEBUG2|DEBUG3

  该选项将指定gs_dump的日志级别。LOG记录正常执行信息，DEBUG记录一些调试信息。DEBUG数字越大，日志级别越高，记录的信息越详细。

- --no-security-labels

  该参数为扩展预留接口，不建议使用。

- --no-tablespaces

  不输出选择表空间的命令。若使用该选项，无论默认表空间是什么，在恢复过程中所有对象都会被创建。

  该选项只对文本格式有意义。针对归档格式，可以在调用gs_restore时指定选项。

- --no-unlogged-table-data

  该参数为扩展预留接口，不建议使用。

- --non-lock-table

  屏蔽gs_dump导出过程中对表加锁的行为。该参数仅供软件间接口调用，不推荐用户主动调用。

  说明

  gs_dump导出table过程中，默认在整个事务内对要导出的table添加access share lock，此级别的锁会阻塞表上并发DDL语句的执行，保证导出表的过程中不会因对表并发DDL的影响而报错。

- --query=[schema_name.][table_name:]query_clause

  导出表内数据时通过该参数支持谓词过滤，通过增加过滤条件，只导出满足过滤条件的表数据。

  其中query_clause是一个标准的以'where'开头的查询子句。

版权所有 © 华为云计算技术有限公司

说明

- 当多个表需要添加谓词过滤时，需要同时使用多个--query参数。
- 可以同时指定多个query参数，但是对于同一张表只能指定一个query参数。
- -t参数的优先级高于--query参数，同时使用-t与--query参数时，先按照-t参数过滤需要被导出的表的范围，再根据范围内的表进行谓词筛选。
- --query参数仅对谓词进行拼接，不对谓词进行转义，需要转义时需要用户手动进行转义处理。
- 使用的谓词应符合SQL标准且能被正确执行，否则数据导出时会报错。
- 不支持对视图、物化视图指定谓词条件，否则数据导出时会报错。
- 导出性能受谓词的查询性能影响。
- --query参数中的where子句部分需要使用单引号或双引号进行包围，如果where子句中出现字符类型的过滤条件，则必须使用双引号进行包围。

  例如where子句中的过滤条件不是字符类型，可以使用单引号或双引号进行包围。

  ```
  --query=test_schema.test_table:'where test_attr > 3'
  --query=test_schema.test_table:"where test_attr > 3"
  ```

  例如where子句中的过滤条件是字符类型，必须使用双引号进行包围。

  ```
  --query=test_schema.test_table:"where test_attr = 'a'"
  ```

- GaussDB会自动将对象名称中的大写字母转为小写，当schema_name、table_name或者query_clause中包含大写字母或特殊字符时，需要添加额外的引号。

  例如Schema名称包含大写字符或特殊字符：

  ```
  --query="\"TEST_SCHEMA\"".test_table:"where test_attr > 3"
  ```

  例如谓词条件包含大写字符（列名有大写）：

  ```
  --query=test_schema.test_table:"where \"TEST_attr\" > 3"
  ```

– --quote-all-identifiers

强制对所有标识符加引号。向后续版本迁移场景下，可能涉及引入额外关键词，使用该选项会有帮助。

说明

M-Compatibility模式数据库不支持该选项。

– --section=SECTION

转储指定名称的区段（pre-data、data和post-data）。

– --serializable-deferrable

转储过程中使用可串行化事务，以确保所使用的快照与之后的数据库状态一致。要实现该操作需要在无异常状况的事务流中等待某个点（所有可能与转储存在读写冲突的并发读写事务都已提交或回滚完毕），这样才能保证转储成功，避免引起其他事务出现serialization_failure要重新再做。

该选项对仅用于灾难恢复的转储没有益处（灾备只要求恢复出完整可用的数据，不需要快照严格一致）。它适用于以下场景：原库持续被业务更新的同时，转储一份副本用于报表或其他只读负载共享——此时副本需要是一个严格一致的状态。若不使用该选项，转储可能呈现一个“任何串行执行都不会产生”的异常状态。例如，使用批处理时，转储中可能出现“批次已标记关闭、但批次内条目尚未全部出现”的矛盾结果。

如果当gs_dump启动时，读写事务仍处于非活动状态，即便使用该选项也不会对其产生影响。如果读写事务处于活动状态，转储的开始可能会延迟一段不确定的时间。

版权所有 © 华为云计算技术有限公司

- --use-set-session-authorization

  输出符合SQL标准的SET SESSION AUTHORIZATION命令，而不是ALTER OWNER命令，来确定对象所有权。使用SET SESSION AUTHORIZATION命令的转储，如果转储文件中对象的历史有问题，可能不能正确恢复。并且，使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员权限才能成功，而ALTER OWNER需要的权限比较低。此外，SET SESSION AUTHORIZATION部分支持直接使用密文密码，因此使用此参数导出的脚本可能无法正常恢复，不建议使用此参数导出。

  说明

  SET SESSION AUTHORIZATION使用范围：

  - 系统管理员可以通过SET SESSION AUTHORIZATION语句切换到普通用户，无法切换到初始用户、其他sysadmin、opradmin、monadmin、poladmin和auditadmin。
  - 其他用户无法通过SET SESSION AUTHORIZATION语句切换用户。

- --all

  示本次gs_dump工具被gs_dumpall工具所调用。除了需要导出触发器场景之外，不建议使用此参数。

- --with-copy-format=FILE_FORMAT

  指定gs_dump导出的数据文件格式。

  说明

  - FILE_FORMAT值为字符串类型，只允许输入TEXT和CSV格式，默认取值为TEXT格式。
  - 仅在目录归档模式下，支持FILE_FORMAT值为CSV格式。
  - 指定FILE_FORMAT值为CSV时，不支持与--inserts、--column-inserts、--attribute-inserts参数同时使用。
  - 使用目录归档模式进行导出后，必须使用gs_restore工具进行导入恢复，用户不需要关心导出的数据文件格式。若用户明确需要导出的数据文件格式为CSV，则可以指定--with-copy-format=CSV进行导出。

- --with-encryption=AES128

  指定转储数据用AES128进行加密。

- --with-key=KEY

  AES128密钥规则如下：

  密钥长度为8~16个字符。

  至少包含大写字母（A-Z），小写字母（a-z），数字（0-9），非字母数字字符（限定为~!@#$%^&*()-_=+\|[{}];:,<.>/?）四类字符中的三类字符。

  说明

  不支持加密导出存储过程和函数。

- --with-salt=RANDVALUES

  gs_dumpall使用此参数传递随机值。

- --include-nodes

  将TO NODE/TO GROUP语句包含在已转储的CREATE TABLE/CREATE FOREIGN TABLE语句中。

- --include-extensions

  在转储中包含扩展。

 版权所有 © 华为云计算技术有限公司

须知

扩展功能为内部使用功能，只允许操作数据库内部的高级包或扩展。

- --include-depend-objs

  备份结果包含依赖于指定对象的对象信息。该参数需要同-t/--include-table-file参数关联使用才会生效。

- --exclude-self

  备份结果不包含指定对象自身的信息。该参数需要同-t/--include-table-file参数关联使用才会生效。

- --pipeline

  使用管道传输密码，禁止在终端使用。

- --no-transaction

  默认情况（不开启该选项），导出的文件（纯文本格式）会为CREATE和ALTER ... OWNER TO、SET SESSION AUTHORIZATION和RESET SESSION AUTHORIZATION这类语句单独添加事务块。如果开启该选项，这些语句将直接输出，而不会被额外包裹在BEGIN/COMMIT中。

- --dont-overwrite-file

  文本、tar以及自定义格式情况下会重写现有文件。对目录格式不适用。

  例如：

  设想这样一种情景，即当前目录下backup.sql已存在。如果在输入命令中输入-f backup.sql选项时，当前目录恰好也生成backup.sql，文件就会被重写。

  如果备份文件已存在，且输入--dont-overwrite-file选项，则会报告附带‘转储文件已经存在’信息的错误。

  ```
  gs_dump -p port_number testdb -f backup.sql -F plain --dont-overwrite-file
  ```

  说明

  - -s/--schema-only和-a/--data-only不能同时使用。
  - -c/--clean和-a/--data-only不能同时使用。
  - --inserts/--column-inserts和-o/--oids不能同时使用，因为INSERT命令不能设置OIDS。
  - --role和--rolepassword必须一起使用。
  - --binary-upgrade-usermap和--binary-upgrade必须一起使用。
  - --include-depend-objs/--exclude-self需要同-t/--include-table-file参数关联使用才会生效。
  - --exclude-self必须同--include-depend-objs一起使用。
  - --with-encryption=AES128仅支持-F p/plain。
  - --with-key=KEY仅支持-F p/plain。
  - --with-salt=RANDVALUES由gs_dumpall调用，不需要用户手动输入。

连接参数：

- -h, --host=HOSTNAME

  指定主机名称。如果数值以斜杠开头，则被用作Unix域套接字的路径。缺省从PGHOST环境变量中获取（如果已设置）。

  该参数只针对集群外部连接，对集群内本机IPv4用127.0.0.1，IPv6用::1。

- -p, --port=PORT

版权所有 © 华为云计算技术有限公司

指定主机端口。在开启线程池情况下，建议使用pooler port，即主机端口+1。缺省从PGPORT环境变量中获取（如果已设置）。

- -U, --username=NAME

  指定所连接主机的用户名，跨节点执行不支持使用初始用户。缺省从PGUSER环境变量中获取（如果已设置）。

  说明

  使用普通用户权限导出、普通用户权限导入时，在导入前需要给普通用户赋予对应数据库创建schema权限，如：GRANT CREATE ON DATABASE db_xxx TO user_xxx。

- -w, --no-password

  不出现输入密码提示。如果主机要求密码认证并且密码没有通过其他形式给出，则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- -W, --password=PASSWORD

  指定用户连接的密码。当主机的认证策略是trust时，不会对系统管理员进行密码验证，即无需输入-W选项；如果不加此参数且不是系统管理员，则会提示交互式输入。为了系统安全，推荐使用交互式输入密码方式。

- --role=ROLENAME

  指定创建转储使用的角色名。选择该选项，会使gs_dump连接数据库后，发起一个SET ROLE角色名命令。当所授权用户（由-U指定）没有gs_dump要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以超系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。

- --rolepassword=ROLEPASSWORD

  指定角色名的密码。

示例

以下示例中，“backup/MPPDB_backup.sql”表示导出的文件，其中backup表示相对于当前目录的相对目录；“37300”表示数据库服务器端口；“testdb”表示要访问的数据库名。

说明

导出操作时，请确保该目录存在并且当前的操作系统用户对其具有读写权限。

示例一

执行gs_dump，导出testdb数据库全量信息，导出的MPPDB_backup.sql文件格式为纯文本格式。

```
gs_dump -U omm -f backup/MPPDB_backup.sql -p 37300 testdb -F p
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: The total objects number is 356.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: [100.00%] 356 objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: dump database testdb successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: total time: 1274  ms
```

示例二

执行gs_dump，导出testdb数据库全量信息，导出的MPPDB_backup.tar文件格式为tar格式。

版权所有 © 华为云计算技术有限公司

```
gs_dump -U omm -f backup/MPPDB_backup.tar -p 37300 testdb -F t
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:24]: The total objects number is 1369.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:53]: [100.00%] 1369 objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:53]: dump database testdb successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:53]: total time: 50086  ms
```

示例三

执行gs_dump，导出testdb数据库全量信息，导出的MPPDB_backup.dmp文件格式为自定义归档格式。

```
gs_dump -U omm -f backup/MPPDB_backup.dmp -p 37300 testdb -F c
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:05:40]: The total objects number is 1369.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:06:03]: [100.00%] 1369 objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:06:03]: dump database testdb successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:06:03]: total time: 36620  ms
```

示例四

执行gs_dump，导出testdb数据库全量信息，导出的MPPDB_backup文件格式为目录格式。

```
gs_dump -U omm -f backup/MPPDB_backup -p 37300  testdb -F d
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:04]: The total objects number is 1369.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:23]: [100.00%] 1369 objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:23]: dump database testdb successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:23]: total time: 33977  ms
```

示例五

执行gs_dump，导出testdb数据库信息，但不导出/home/MPPDB_temp.sql中指定的表信息。导出的MPPDB_backup.sql文件格式为纯文本格式。

```
gs_dump -U omm -p 37300 testdb --exclude-table-file=/home/MPPDB_temp.sql -f backup/MPPDB_backup.sql
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:01]: The total objects number is 1367.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:22]: [100.00%] 1367 objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:22]: dump database testdb successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:22]: total time: 37017  ms
```

示例六

执行gs_dump，仅导出依赖于指定表testtable的视图信息。

```
gs_dump -U omm -s -p 37300 testdb -t PUBLIC.testtable --include-depend-objs --exclude-self -f backup/MPPDB_backup.sql -F p
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: The total objects number is 331.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: [100.00%] 331 objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: dump database testdb successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: total time: 327  ms
```

 版权所有 © 华为云计算技术有限公司

相关工具

gs_dumpall导出所有数据库信息，**gs_restore导入数据**，**gsql连接数据库**

4.3.3 gs_dumpall 导出所有数据库信息

工具介绍

gs_dumpall是GaussDB用于导出所有数据库相关信息的工具，它可以导出集群数据库的所有数据，包括默认数据库postgres的数据、自定义数据库的数据、集群所有数据库公共的全局对象。

gs_dumpall工具在进行数据导出时，其他用户可以访问集群数据库（读或写）。

gs_dumpall工具支持导出完整一致的数据。例如，T1时刻启动gs_dumpall导出整个集群数据库，那么导出数据结果将会是T1时刻该集群数据库的数据状态，T1时刻之后对集群数据库的修改不会被导出。

gs_dumpall在导出整个集群所有数据库时分为两部分：

- gs_dumpall自身对所有数据库公共的全局对象进行导出，包括有关数据库用户和组、表空间以及属性（例如，适用于数据库整体的访问权限）信息。
- gs_dumpall通过调用gs_dump来完成集群中各数据库的SQL脚本文件导出，该脚本文件包含将数据库恢复为其保存时的状态所需要的全部SQL语句。

以上两部分导出的结果为纯文本格式的SQL脚本文件，使用**gsql连接数据库**运行该脚本文件可以恢复集群数据库。

gs_dumpall支持SSL加密通信，使用方式同gsql方式。

使用gs_dumpall前请确保gs_dumpall版本与gs_dump版本、数据库版本保持一致，高版本gs_dumpall不保证完全兼容低版本内核数据。

gs_dumpall使用时会调用gs_dump，因此一些关于数据库内对象导出的约束与gs_dump相同。

注意事项

- 禁止修改导出的文件和内容，否则可能无法恢复成功。
- 为了保证数据一致性和完整性，gs_dumpall会对需要转储的表设置共享锁。如果某张表在别的事务中设置了共享锁，gs_dumpall会等待此表的锁释放后锁定此表。如果无法在指定时间内锁定某张表，转储会失败。用户可以通过指定--lock-wait-timeout选项，自定义等待锁超时时间。
- 由于gs_dumpall读取所有数据库中的表，因此必须以数据库集群管理员身份进行连接，才能导出完整文件。在使用gsql执行脚本文件导入时，同样需要管理员权限，以便添加用户和组，以及创建数据库。导入备份前，验证其安全性，防止管理员权限被利用。
- 使用gs_dumpall导出所有数据库对象，并希望在新的实例环境上进行导入时，需要保证导出和导入时使用用户的名称和权限相同，否则会出现名称不一致或权限不足的报错。
- 对于定时任务，本工具仅支持导出在MYSQL兼容性数据库中，通过CREATE EVENT创建的定时任务或通过高级包创建的非周期性定时任务。

版权所有 © 华为云计算技术有限公司

- gs_dumpall不支持导出自定义Tokenweight分词词典，可以根据报错WARNING: dictionary xx cannot be automatically exported, please create it manually手动创建对应分词词典。
- 由于M-compatibility模式数据库全局只能创建一个，当导出的数据库中包含M-compatibility模式数据库时，须保证导入的目标实例环境上不存在其他的M-compatibility模式数据库。
- 当指定转储编码存在转码场景时，且表中的数据存在非法编码的数据，导出会报错invalid byte sequence，建议使用gs_dump的-s参数只导出定义，并单独使用COPY打开编码容错进行数据的导出与导入。
- gs_dumpall当前设计不兼容三权分立，三权分立下仅可通过初始用户使用gs_dumpall工具。导入的文件也需要通过初始用户执行。
- 使用系统管理员用户执行gs_dumpall导出时，不会导出初始用户创建的对象，初始用户创建的对象需要使用初始用户进行导出。
- 当导出对象中包含依赖package内部函数的视图时，在首次导入目标端时会产生视图依赖对象不存在的错误，此时需要使用gs_dumpall --schema-only备份元数据，并再次导入到目标端即可正常。
- gs_dumpall默认会导出失效视图，如果导出的视图中有失效视图会打印提示，在507.0之前版本导入时可能会报错。可指定--ignore-invalid-views选项，不导出失效视图。

命令格式

```
gs_dumpall [OPTION]...
```

参数说明

通用参数：

- -f, --filename=FILENAME

  将输出发送至指定文件。如果这里省略，则使用标准输出。
- -v, --verbose

  指定verbose模式。该选项将导致gs_dumpall向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。
- -V, --version

  打印gs_dumpall版本，然后退出。
- --lock-wait-timeout=TIMEOUT

  请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。参数单位为ms（毫秒）。可以以任何符合SET statement_timeout的格式指定超时时间。
- -?, --help

  显示gs_dumpall命令行参数帮助，然后退出。
- --ignore-invalid-views

  指定该参数时，不会导出失效视图，导出时发现有失效视图会打印提示。

转储参数：

- -a, --data-only

  只转储数据，不转储模式（数据定义）。

版权所有 © 华为云计算技术有限公司

- -c, --clean

  在重新创建数据库之前，执行SQL语句清理（删除）这些数据库。针对角色和表空间的转储命令已添加。

- -g, --globals-only

  只转储全局对象（角色和表空间），无数据库。

- -o, --oids

  转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于应用以某种方式参照了OID列的情况。如果不是以上这种情况，请勿使用该选项。

- -O, --no-owner

  不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs_dumpall会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的模式元素的所属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定-O，编写一个任何用户都能存储的脚本，且该脚本会授予该用户拥有所有对象的权限，因为没有使用ALTER OWNER或SET SESSION AUTHORIZATION语句，导入时一直使用执行用户权限，所以在导入前请确认转储文件内是否存在风险。例如，检查转储文件中是否包含提权语句，且该语句管理员是否知晓。

- -r, --roles-only

  只转储角色，不转储数据库或表空间。

- -s, --schema-only

  只转储对象定义（模式），而非数据。

- -S, --sysadmin=NAME

  该参数为扩展预留接口，不建议使用。

- -t, --tablespaces-only

  只转储表空间，不转储数据库或角色。

- -x, --no-privileges

  防止转储访问权限（授权/撤销命令）。

- --column-inserts|--attribute-inserts

  以INSERT命令带列名的方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致失败语句对应的数据丢失，而非整个表内容。

  说明

  M-Compatibility模式数据库不支持该选项，导出M-Compatibility模式数据库时会忽略该选项。

- --disable-dollar-quoting

  该选项将禁止在函数体前使用美元符号$，并强制使用SQL标准字符串语法对其进行引用。

- --disable-triggers

  该参数为扩展预留接口，不建议使用。

- --inserts

  发出INSERT命令（而非COPY命令）时转储数据。这会导致恢复缓慢。注意如果重排列顺序，可能会导致恢复整个失败。--column-inserts选项更加安全，虽然可能更慢些。

版权所有 © 华为云计算技术有限公司

说明

M-Compatibility模式数据库不支持该选项，导出M-Compatibility模式数据库时会忽略该选项。

- --no-security-labels

  该参数为扩展预留接口，不建议使用。

- --no-tablespaces

  请勿输出创建表空间的命令，也请勿针对对象选择表空间。使用该选项，无论默认表空间是哪一个，在恢复过程中所有对象都会被创建。

- --no-unlogged-table-data

  该参数为扩展预留接口，不建议使用。

- --include-alter-table

  导出表中已删除的列信息。

- --quote-all-identifiers

  强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。

  说明

  M-Compatibility模式数据库不支持该选项，导出M-Compatibility模式数据库时会忽略该选项。

- --dont-overwrite-file

  不重写当前文件。

- --use-set-session-authorization

  输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功，而ALTER OWNER需要的权限则低得多。但是SET SESSION AUTHORIZATION部分支持使用密文密码随意切换用户及权限，因此使用此参数导出的脚本可能无法正常恢复，不建议使用此参数导出。

  说明

  SET SESSION AUTHORIZATION使用范围：

  - 系统管理员可以通过SET SESSION AUTHORIZATION语句切换到普通用户，无法切换到初始用户、其他sysadmin、opradmin、monadmin、poladmin和auditadmin。
  - 其他用户无法通过SET SESSION AUTHORIZATION语句切换用户。

- --with-encryption=AES128

  指定转储数据需用AES128进行加密。

- --with-key=KEY

  AES128密钥规则如下：

  密钥长度为8~16个字符。

  至少包含大写字母（A-Z），小写字母（a-z），数字（0-9），非字母数字字符（限定为~!@#$%^&*()-_=+\|[{}];:,<.>/?）四类字符中的三类字符。

- --include-extensions

  如果include-extensions参数被设置，将备份所有的CREATE EXTENSION语句。

 版权所有 © 华为云计算技术有限公司

须知

扩展功能为内部使用功能，只允许操作数据库内部的高级包或扩展。

- --include-templatedb
  转储过程中包含模板库。
- --dump-nodes
  转储过程中包含节点和Node Group。
- --include-nodes
  将TO NODE语句包含在已转储的CREATE TABLE命令中。
- --include-buckets
  该参数为扩展预留接口，不建议使用。
- --dump-wrm
  存储过程中包含负载资源管理器，具体包括资源池、负载组以及负载组映射。

  说明

  连接M-Compatibility模式数据库进行导出时，不支持该选项。
- --binary-upgrade
  该参数为扩展预留接口，不建议使用。

  说明

  M-Compatibility数据库不支持该选项，导出M-Compatibility数据库时会忽略该选项。
- --binary-upgrade-usermap="USER1=USER2"
  该参数为扩展预留接口，不建议使用。
- --non-lock-table
  该参数仅供软件间接口调用。
- --tablespaces-postfix
  该参数为扩展预留接口，不建议使用。
- --parallel-jobs
  指定备份进程并发数，取值范围为1~1000。
- --no-transaction
  默认情况下，导出的文件(纯文本格式)会为 CREATE 和 ALTER ... OWNER TO 以及SET SESSION AUTHORIZATION和RESET SESSION AUTHORIZATION这类语句单独添加事务块。如果开启该选项，这些语句将直接输出，而不会被额外包裹在 BEGIN/COMMIT 中。
- --pipeline
  使用管道传输密码，禁止在终端使用。

 版权所有 © 华为云计算技术有限公司

说明

- -g/--globals-only和-r/--roles-only不能同时使用。
- -g/--globals-only和-t/--tablespaces-only不能同时使用。
- -r/--roles-only和-t/--tablespaces-only不能同时使用。
- -s/--schema-only和-a/--data-only不能同时使用。
- -r/--roles-only和-a/--data-only不能同时使用。
- -t/--tablespaces-only和-a/--data-only不能同时使用。
- -g/--globals-only和-a/--data-only不能同时使用。
- --tablespaces-postfix和--binary-upgrade必须一起使用。
- --binary-upgrade-usermap和--binary-upgrade必须一起使用。
- --parallel-jobs和-f/--file必须一起使用。

连接参数：

- -h, --host=HOSTNAME

  指定主机的名称。如果取值是以斜线开头，它将用作Unix域套接字的目录。默认值取自PGHOST环境变量；如果没有设置，将启动某个Unix域套接字建立连接。

  该参数只针对集群外，对集群内本机IPv4用127.0.0.1，IPv6用::1。

  环境变量：PGHOST

- -l, --database=DATABASENAME

  指定所连接的转储全局对象的数据库名称，并去寻找还有其他哪些数据库需要被转储。如果没有指定，会使用postgres数据库，如果postgres数据库不存在，会使用template1。

  说明

  无论指定哪一个数据库作为导出时连接的数据库，在新实例环境上使用gsql工具进行导入时，必须连接默认数据库，否则可能出现语法不兼容的情况，导致导入异常。特别地，当错误地连接M-Compatibility模式数据库进行导入时，由于M-compatibility模式数据库下CREATE DATABASE与CREATE SCHEMA效果等价，会导致创建失败。

- -p, --port=PORT

  指定服务器所侦听的TCP端口或本地Unix域套接字后缀，以确保连接。默认值设置为PGPORT环境变量。

  在开启线程池情况下，建议使用 pooler port，即侦听端口+1。

  环境变量：PGPORT

- -U, --username=NAME

  所连接的用户名，跨节点执行不支持使用初始用户。

  环境变量：PGUSER

- -w, --no-password

  不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其他形式给出，则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- -W, --password=PASSWORD

  指定用户连接的密码。如果主机的认证策略是trust，则不会对系统管理员进行密码验证，即无需输入-W选项；如果不加此参数，并且不是系统管理员，则会提示交互式输入，为了系统安全，推荐使用交互式输入密码方式。

 版权所有 © 华为云计算技术有限公司

- --role=ROLENAME

  指定创建转储使用的角色名。选择该选项，会使gs_dumpall连接数据库后，发起一个SET ROLE角色名命令。当所授权用户（由-U指定）没有gs_dumpall要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。

- --rolepassword=ROLEPASSWORD

  指定具体角色用户的角色密码。

说明

- 由于gs_dumpall内部调用gs_dump，所以一些诊断信息参见**gs_dump导出数据库信息**。
- 一旦恢复，建议在每个数据库上运行ANALYZE，优化程序提供有用的统计数据。
- gs_dumpall恢复前需要所有必要的表空间目录为空；否则，对于处在非默认位置的数据库，数据库创建会失败。

示例

使用gs_dumpall一次导出集群的所有数据库。

说明

gs_dumpall仅支持纯文本格式导出。所以只能使用gsql恢复gs_dumpall导出的转储内容。

- 导出所有数据库。

  backup/bkp2.sql表示导出的文件，29900表示数据库服务器端口，omm为用户名。

```
gs_dumpall -U omm -f ./backup/bkp2.sql -p 29900
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:00:01]: The total objects number is 458.
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:00:01]: [100.00%] 458 objects have been dumped.
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:00:01]: dump database dbname='postgres' successfully
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:00:01]: total time: 3239  ms
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:00:01]: dumpall operation successful
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:00:01]: total time: 3305  ms
```

- 设置4并发导出所有数据库。

```
gs_dumpall -U omm -f ./backup/bkp2.sql -p 29900 --parallel-jobs 4
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:04:13]: The total objects number is 458.
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:04:13]: [100.00%] 458 objects have been dumped.
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:04:13]: dump database dbname='postgres' successfully
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:04:13]: total time: 3261  ms
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:04:13]: dumpall operation successful
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:04:13]: total time: 3320  ms
```

- 只导出全局对象（角色和表空间）。

```
gs_dumpall -U omm -f ./backup/bkp2.sql -p 29900 -g
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:06:00]: dumpall operation successful
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:06:00]: total time: 45  ms
```

- 指定导出文件用AES128进行加密。

版权所有 © 华为云计算技术有限公司

使用实际的密码替换********。

```
gs_dumpall -U omm -f ./backup/bkp2.sql -p 29900 --with-encryption=AES128 --with-key='********'
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:07:20]: The total objects number is 459.
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:07:20]: [100.00%] 459 objects have been dumped.
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:07:21]: dump database dbname='postgres' successfully
gs_dump[user='omm'][localhost][port='29900'][dbname='postgres'][2024-11-11 11:07:21]: total time: 3518  ms
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:07:21]: dumpall operation successful
gs_dumpall[user='omm'][localhost][port='29900'][2024-11-11 11:07:21]: total time: 3764  ms
```

相关命令

gs_dump导出数据库信息，**gs_restore导入数据**

4.3.4 gs_restore 导入数据

工具介绍

gs_restore是GaussDB提供的针对gs_dump导出数据的导入工具。通过此工具可将gs_dump导出生成的文件进行导入。

主要功能包含：

- 导入到数据库

  如果连接参数中指定了数据库，则数据将被导入到指定的数据库中。其中，并行导入必须指定连接的密码。
- 导入到归档文件

  如果参数指定"-l"，则生成归档文件，其中包含简略的数据总结。

gs_restore支持SSL加密通信，使用方式同gsql方式。

前提条件

使用gs_restore前请确保gs_restore版本与gs_dump版本、数据库版本保持一致。

注意事项

- gs_restore恢复到新库中时，允许隐藏系统列上rowid的值发生变化，允许隐藏列关联的sequence值发生变化，数据类型为rowid的用户列数据不变。

命令格式

```
gs_restore [OPTION]... FILE
```


版权所有 © 华为云计算技术有限公司

说明

- FILE没有短选项或长选项。用来指定归档文件所处的位置。
- 作为前提条件，需输入dbname或-l选项。不允许用户同时输入dbname和-l选项。
- gs_restore默认是以追加的方式进行数据导入。为避免多次导入造成数据异常，在进行导入时，建议选择使用“-c”和“-e”参数。“-c”表示在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象；“-e”表示当发送SQL语句到数据库时如果出现错误请退出，默认状态下会继续，且在导入后会显示一系列错误信息。
- 在进行导入时，如果schema对象的owner拥有OPRADMIN的系统权限，那么在导入时需要使用初始用户。
- 在进行导入时，如果数据中存在非法编码的数据并且不需要进行转码，可以配置数据库兼容性参数copy_special_character_version为'no_error'将非法编码的数据进行导入，否则会报错，但不会中断导入。
- gs_restore当前设计不兼容三权分立，三权分立下仅可通过初始用户使用gs_restore工具。导入的文件也需要通过初始用户执行。

参数说明

通用参数：

- -d, --dbname=NAME
  连接数据库dbname并直接导入到该数据库中。
- -f, --file=FILENAME
  指定生成归档的输出文件，使用-l时列表的输出文件。
  默认是标准输出。

  说明

  -f不能同-d一起使用。
- -F, --format=c|d|t
  指定归档格式。由于gs_restore会自动决定格式，因此不需要指定格式。
  取值范围：
  - c/custom：该归档形式为gs_dump的自定义格式。
  - d/directory：该归档形式是一个目录归档形式。
  - t/tar：该归档形式是一个tar归档形式。
- -l, --list
  列出归档形式内容。这一操作的输出可用作-L选项的输入。

  说明

  如果类似-n或-t的过滤选项与-l同时使用，过滤选项将会限制列举的项目（即归档形式内容）。
- -v, --verbose
  指定verbose模式。
- -V, --version
  打印gs_restore版本，然后退出。
- -?, --help
  显示gs_restore命令行参数帮助，然后退出。

导入参数：

 版权所有 © 华为云计算技术有限公司

- -a, --data-only

  只导入数据，不导入模式（数据定义）。gs_restore的导入是以追加方式进行的。

- -c, --clean

  在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象。如果目标数据库中没有删除操作涉及的对象，可能会输出一些提示性的错误信息。

  说明

  - 备份文件中的删除数据库对象指令，均为非级联式删除的DROP命令。
  - 连接database1在t1时刻使用-c参数导出表table1到backup01.dmp文件，t2时刻在database1中创建视图view1并依赖table1，t3使用gs_restore的-c参数将backup01.dmp恢复到database1中时，因为会使用非级联删除，在删除table1时会报错ERROR: table table cannot be dropped because other objects depend on it. 并在创建table1时报错ERROR: Relation "table1" already exists. 此时需要用户在评估影响后，手动对相关依赖对象进行删除，或手动执行级联删除命令。

- -C, --create

  导入数据库之前会先使用CREATE DATABASE创建数据库（指定该选项后，-d指定的数据库仅用于执行CREATE DATABASE命令，所有数据依然会导入到创建的数据库中）。

  说明

  在M-Compatibility模式数据库下，不支持使用该选项。必须先在目标实例上创建M-Compatibility模式数据库，然后在源实例上导出，最后在目标实例连接新创建的M-Compatibility模式数据库导入。

- -e, --exit-on-error

  当发送SQL语句到数据库时如果出现错误，请退出。默认状态下会继续，且在导入后会显示一系列错误信息。

- -I, --index=NAME

  只导入已列举的index的定义。允许导入多个index。如果多次输入-I index导入多个index。

  例如：

  ```
  gs_restore -h host_name -p port_number -d testdb -I Index1 -I Index2 backup/MPPDB_backup.tar
  ```

  在上面这个例子中，Index1和Index2会被导入。

- -j, --jobs=NUM

  运行gs_restore最耗时的部分（如加载数据、创建index或创建约束）使用并发任务。该选项能大幅缩短导入时间，即将一个大型数据库导入到某一多处理器的服务器上。

  每个任务可能是一个进程或一个线程，这由操作系统决定。每个任务与服务器进行单独连接。

  该选项的最优值取决于服务器的硬件设置、客户端、以及网络。还包括这些因素，如CPU核数量、硬盘设置。建议是从增加服务器上的CPU核数量入手，更大的值（服务器上CPU核数量）在很多情况下也能导致数据文件更快地被导入。相应地，过高的值会由于超负荷反而导致性能降低。

  该选项只支持自定义归档格式。输入文件必须是常规文件（不能是像pipe的文件）。如果是通过脚本文件，而非直接连接数据库服务器，该选项可忽略。而且，多任务不能与--single-transaction选项一起使用。

版权所有 © 华为云计算技术有限公司

说明

- 此参数适用于多表/多索引/多约束的情况。实际使用过程中，创建的进程数（或线程数）与表、索引、约束等的数量有关，最高并发不会超过给定的jobs数。
- 指定的并行数不允许超过客户端CPU核数的两倍。当客户端运行在容器中时，由于获取到的是主机的CPU数量，可能比容器实际能使用的多，并发度建议由用户自行控制在实际能使用的客户端CPU数量的两倍范围内。并行导入通过多进程并发多个事务实现，实际并发度会受限于服务端的线程池模型，也同时会给服务端增加压力，请按实际情况合理设置并发度，例如并发度不超过客户端空闲CPU数量，同时不超过服务端最大连接数限制等。

- -L, --use-list=FILENAME

  只导入列举在list-file中的那些归档形式元素，导入顺序以它们在文件中的顺序为准。注意如果像-n或-t的过滤选项与-L使用，它们将会进一步限制导入的项目。

  一般情况下，list-file是通过编辑前面提到的某个-l参数的输出创建的。文件行的位置可更改或直接删除行，也可使用分号（;）在行的开始注释。见下文的举例。

- -n, --schema=NAME

  只导入已列举的模式中的对象。

  该选项可与-t选项一起用以导入某个指定的表。

  多次输入-n *schemaname*可以导入多个模式。

  例如：

  ```
  gs_restore -h host_name -p port_number -d testdb -n sch1 -n sch2 backup/MPPDB_backup.tar
  ```

  在上面这个例子中，sch1和sch2会被导入。

  说明

  GaussDB会自动将对象名称中的大写字母转为小写，当模式名称中包含大写字母时，需要添加额外的引号，如：-n '"Sch1"' 或 -n "\"Sch1\""

- -O, --no-owner

  不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs_restore会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的模式元素的所属。除非是由系统管理员（或是拥有脚本中所有对象的同一个用户）进行数据库首次连接的操作，否则语句会失败。使用-O选项，任何用户名都可用于首次连接，且该用户拥有所有已创建的对象。

- -P, --function=NAME(args)

  只导入已列举的函数。请按照函数所在转储文件中的目录，准确拼写函数名称和参数。

  当-P单独使用时，表示导入文件中所有'function-name(args)'函数；当-P同-n一起使用时，表示导入指定模式下的'function-name(args)'函数；多次输入-P，而仅指定一次-n，表示所有导入的函数默认都是位于-n模式下的。

  可以多次输入-n schema-name -P 'function-name(args)'同时导入多个指定模式下的函数。

  例如：

  ```
  ./gs_restore -h host_name -p port_number -d testdb -n test1 -P 'Func1(integer)' -n test2 -P 'Func2(integer)' backup/MPPDB_backup.tar
  ```

  在上面这个例子中，test1模式下的函数Func1(i integer)和test2模式下的函数Func2(j integer)会被一起导入。

- -s, --schema-only

  只导入模式（数据定义），不导入数据（表内容）。当前的序列值也不会导入。

版权所有 © 华为云计算技术有限公司

- -S, --sysadmin=NAME

  该参数为扩展预留接口，不建议使用。

- -t, --table=NAME

  只导入已列举的表定义、数据或定义和数据。该选项与-n选项同时使用时，用来指定某个模式下的表对象。-n参数不输入时，默认为PUBLIC模式。多次输入-n <schemaname> -t <tablename>可以导入指定模式下的多个表。

  例如：

  导入PUBLIC模式下的table1

  ```
  gs_restore -h host_name -p port_number -d testdb -t table1 backup/MPPDB_backup.tar
  ```

  导入test1模式下的test1和test2模式下test2

  ```
  gs_restore -h host_name -p port_number -d testdb -n test1 -t test1 -n test2 -t test2 backup/MPPDB_backup.tar
  ```

  导入PUBLIC模式下的table1和test1 模式下test1

  ```
  gs_restore -h host_name -p port_number -d testdb -n PUBLIC -t table1 -n test1 -t table1 backup/MPPDB_backup.tar
  ```

  须知

  - -t不支持schema_name.table_name的输入格式，指定此格式不会报错，但不会生效。
  - 当-t已指定时，gs_restore不会导入已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地导入到一个空数据库中。
  - -t tablename不会导入表上的触发器信息。

- -T, --trigger=NAME

  该参数为扩展预留接口。

- -x, --no-privileges/--no-acl

  防止导入访问权限（grant/revoke命令）。

- -1, --single-transaction

  执行导入作为一个单独事务。

  该选项确保要么所有命令成功完成，要么没有改变应用。该选项意为--exit-on-error。

- --disable-triggers

  该参数为扩展预留接口，不建议使用。

- --log-level=LOG|DEBUG|DEBUG1|DEBUG2|DEBUG3

  该选项将指定gs_restore的日志级别。LOG记录正常执行信息，DEBUG记录一些调试信息。DEBUG数字越大，日志级别越高，记录的信息越详细。

- --no-data-for-failed-tables

  默认状态下，即使创建表的命令失败（如表已经存在），表数据仍会被导入。使用该选项，像这种表的数据会被跳过。如果目标数据库已包含想要的表内容，这种行为会有帮助。

  该选项只有在直接导入到某数据库中时有效，不针对生成SQL脚本文件输出。

- --no-security-labels

  该参数为扩展预留接口，不建议使用。

版权所有 © 华为云计算技术有限公司

- --no-tablespaces

  不输出选择表空间的命令。使用该选项，无论默认表空间是哪一个，在导入过程中所有对象都会被创建。

- --no-transaction

  默认情况下，CREATE 和 ALTER ... OWNER TO 以及SET SESSION AUTHORIZATION和RESET SESSION AUTHORIZATION这类语句默认会在单独的事务块中执行。如果开启该选项，这些语句将直接执行，而不会额外包裹在 BEGIN/COMMIT 中。

- --section=SECTION

  导入已列举的区段（如pre-data、data、或post-data）。

- --use-set-session-authorization

  该选项用来进行文本格式的备份。

  输出SET SESSION AUTHORIZATION命令，而非ALTER OWNER命令，用以决定对象归属。该选项使转储更加兼容标准，但通过参考转储中对象的记录，导入过程可能会有问题。使用SET SESSION AUTHORIZATION的转储要求必须是系统管理员，同时在导入前还需参考"SET SESSION AUTHORIZATION"，手工对导出文件的密码进行修改验证，只有这样才能进行正确的导入操作，相比之下，ALTER OWNER对权限要求较低。

- --pipeline

  使用管道传输密码，禁止在终端使用。

须知

- 如果安装过程中有任何本地数据要添加到template1数据库，请谨慎将gs_restore的输出载入到一个真正的空数据库中；否则可能会因为被添加对象的定义被复制，而出现错误。要创建一个无本地添加的空数据库，需从template0而非template1复制，例如：

```
gaussdb=# CREATE DATABASE foo WITH TEMPLATE template0;
```

说明

- -d/--dbname 和 -f/--file 不能同时使用。
- -s/--schema-only 和 -a/--data-only不能同时使用。
- -c/--clean 和 -a/--data-only不能同时使用。
- 使用--single-transaction时，-j/--jobs必须为单任务。
- --role 和 --rolepassword必须一起使用。

连接参数：

- -h, --host=HOSTNAME

  指定的主机名称。如果取值是以斜线开头，他将用作Unix域套接字的目录。默认值取自PGHOST环境变量；如果没有设置，将启动某个Unix域套接字建立连接。

  该参数只针对集群外，对集群内本机IPv4用127.0.0.1，IPv6用::1。

  环境变量：PGHOST

- -p, --port=PORT

  指定服务器所侦听的TCP端口或本地Unix域套接字后缀，以确保连接。默认值设置为PGPORT环境变量。

版权所有 © 华为云计算技术有限公司

在开启线程池情况下，建议使用 pooler port，即侦听端口+1。

环境变量：PGPORT

- -U, --username=NAME

  所连接的用户名，跨节点执行不支持使用初始用户。

  环境变量：PGUSER

  说明

  使用普通用户权限导出，普通用户权限导入时，在导入前需要给普通用户赋予对应数据库创建schema权限，如：grant create on database db_xxx to user_xxx;

- -w, --no-password

  不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其他形式给出，则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- -W, --password=PASSWORD

  指定用户连接的密码。如果主机的认证策略是trust，则不会对系统管理员进行密码验证，即无需输入-W参数；如果不加此参数，并且不是系统管理员，则会提示交互式输入，为了系统安全，推荐使用交互式输入密码方式。

- --role=ROLENAME

  指定导入操作使用的角色名。选择该参数，会使gs_restore连接数据库后，发起一个SET ROLE角色名命令。当所授权用户（由-U指定）没有gs_restore要求的权限时，该参数会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以初始用户身份登录，而使用该参数能够在不违反该规定的情况下完成导入。

- --rolepassword=ROLEPASSWORD

  指定导入操作使用的角色密码。

示例

gs_restore用来导入由gs_dump生成的导出文件。

- 示例1：执行gs_restore，将导出的MPPDB_backup.dmp文件（自定义归档格式）导入到testdb数据库。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb
restore operation successful
total time: 13053  ms
```

- 示例2：执行gs_restore，将导出的MPPDB_backup.tar文件（tar格式）导入到testdb数据库。

```
gs_restore backup/MPPDB_backup.tar -p 8000 -d testdb
restore operation successful
total time: 21203  ms
```

- 示例3：执行gs_restore，将导出的MPPDB_backup文件（目录格式）导入到testdb数据库。

```
gs_restore backup/MPPDB_backup -p 8000 -d testdb
restore operation successful
total time: 21003  ms
```

- 示例4：执行gs_restore，使用自定义归档格式的MPPDB_backup.dmp文件来进行如下导入操作。 导入PUBLIC模式下所有对象的定义和数据。在导入时会先删除已经存在的对象，如果原对象存在跨模式的依赖则需手工强制干预。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -c -n PUBLIC
Error while PROCESSING TOC:
Error from TOC entry 313; 1259 337399 TABLE table1 gaussdba
```

 版权所有 © 华为云计算技术有限公司

```
could not execute query: ERROR:  cannot drop table table1 because other objects depend on it
DETAIL:  view t1.v1 depends on table table1
HINT:  Use DROP ... CASCADE to drop the dependent objects too.
   Command was: DROP TABLE IF EXISTS public.table1;
```

手工删除依赖，导入完成后再重新创建。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -c -n PUBLIC
restore operation successful
total time: 2203  ms
```

- 示例5：执行gs_restore，使用自定义归档格式的MPPDB_backup.dmp文件来进行如下导入操作。只导入PUBLIC模式下表table1的定义。

  ```
  gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -c -s -n PUBLIC -t table1
  restore operation successful
  total time: 21000  ms
  ```

- 示例6：执行gs_restore，使用自定义归档格式的MPPDB_backup.dmp文件来进行如下导入操作。只导入PUBLIC模式下表table1的数据。

  ```
  gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -a -n PUBLIC -t table1
  restore operation successful
  total time: 20203  ms
  ```

- 特例：执行gsql程序，使用如下选项将gs_dump/gs_dumpall导出的纯文本格式文件MPPDB_backup.sql导入到testdb数据库。

  ```
  --如下示例-f指定文件绝对路径，“8000”表示数据库服务器端口；“testdb”表示要访问的数据库名。
  gsql -d testdb -p 8000 -f /home/omm/test/MPPDB_backup.sql
  SET
  SET
  SET
  SET
  SET
  ALTER TABLE
  ALTER TABLE
  ALTER TABLE
  ALTER TABLE
  ALTER TABLE
  CREATE INDEX
  CREATE INDEX
  CREATE INDEX
  SET
  CREATE INDEX
  REVOKE
  REVOKE
  GRANT
  GRANT
  total time: 30476  ms
  ```

相关命令

gs_dump导出数据库信息，**gs_dumpall导出所有数据库信息**

版权所有 © 华为云计算技术有限公司

5 日志参考

5.1 日志类型简介

在数据库运行过程中，会出现大量日志，既有保证数据库安全可靠的WAL日志（预写式日志，也称为Xlog），也有用于数据库日常维护的运行和操作日志等。在数据库发生故障时，可以参考这些日志进行问题定位和数据库恢复的操作。

日志类型

日志类型的详细说明请参见下表。

表 5-1 日志类型

类型	说明
系统日志	数据库系统进程运行时产生的日志，记录系统进程的异常信息。
操作日志	通过客户端工具（例如gs_guc）操作数据库时产生的日志。
审计日志	开启数据库审计功能后，将数据库用户的某些操作记录在日志中，这些日志称为审计日志。
WAL日志	又称为REDO日志，在数据库异常损坏时，可以利用WAL日志进行恢复。由于WAL日志的重要性，所以需要经常备份这些日志。
内存日志	保存数据库进程动态内存使用超过90%时的内存上下文统计信息。

日志清单

表 5-2 日志清单

日志分类	日志名称	日志路径
系统日志	gaussdb-创建时间.log	$GAUSSLOG/gs_log

版权所有 © 华为云计算技术有限公司

日志分类	日志名称	日志路径
	system_alarm-创建时间-current.log	$GAUSSLOG/gs_log
	gtm-创建时间.log	$GAUSSLOG/gs_log/gtm
	cm_agent-创建时间-current.log	$GAUSSLOG/cm/cm_agent
	system_alarm-创建时间-current.log	$GAUSSLOG/cm/cm_agent
	system_call-current.log	$GAUSSLOG/cm/cm_agent
	system_stat-current.log	$GAUSSLOG/cm/cm_agent
	cm_server-创建时间-current.log	$GAUSSLOG/cm/cm_server
	key_event-创建时间-current.log	$GAUSSLOG/cm/cm_server
	system_alarm-创建时间-current.log	$GAUSSLOG/cm/cm_server
	cm_ctl-创建时间-current.log	$GAUSSLOG/cm/cm_ctl
	om_monitor-创建时间-current.log	$GAUSSLOG/cm/om_monitor
	system_alarm-创建时间-current.log	$GAUSSLOG/cm/om_monitor
操作日志	gs_install-创建时间.log	$GAUSSLOG/om
	gs_preinstall-创建时间.log	$GAUSSLOG/om
	gs_om-创建时间.log	$GAUSSLOG/om
	gs_expand-创建时间.log	$GAUSSLOG/om
	gs_replace-创建时间.log	$GAUSSLOG/om
	gs_local-创建时间.log	$GAUSSLOG/om
	checkRunStatus-创建时间.log	$GAUSSLOG/om
	JsonToDbclusterInfo-创建时间.log	$GAUSSLOG/om
	killall_log-创建时间.log	$GAUSSLOG/om
	uploader-创建时间.log	$GAUSSLOG/om
	cmd_sender-创建时间.log	$GAUSSLOG/om

版权所有 © 华为云计算技术有限公司

日志分类	日志名称	日志路径
	gs_initdb-创建时间-current.log	$GAUSSLOG/bin/gs_initdb
	gs_guc-创建时间-current.log	$GAUSSLOG/bin/gs_guc
	gs_cgroup-创建时间.log	$GAUSSLOG/bin/gs_cgroup
	gs_clean-创建时间-current.log	$GAUSSLOG/bin/gs_clean
	gs_ctl-创建时间-current.log	$GAUSSLOG/bin/gs_ctl
	gs_encrypt-创建时间-current.log	$GAUSSLOG/bin/gs_encrypt
	gs_initgtm-创建时间-current.log	$GAUSSLOG/bin/gs_initgtm
	gs_obs.run.log	$GAUSSLOG/bin/gs_obs
	gtm_ctl-创建时间-current.log	$GAUSSLOG/bin/gtm_ctl
	gs_redis-创建时间-current.log	$GAUSSLOG/bin/gs_redis
审计日志	–	默认$GAUSSLOG/pg_audit（如果使用om工具部署集群，则审计日志路径为“$GAUSSLOG/pg_audit/实例名称”），支持由GUC参数audit_directory自主配置路径。
WAL日志	pg_xlog	数据目录/pg_xlog。例如DN实例上为/gaussdb/data/data_dn/pg_xlog，其中/gaussdb/data/data_dn代表集群节点的数据目录。
内存日志	mem_log-创建时间.log	$GAUSSLOG/mem_log

5.2 系统日志

集群运行时CN、DN、GTM、CM以及集群安装部署时产生的日志统称为系统日志。如果集群在运行时发生故障，可以通过这些系统日志及时定位故障发生的原因，根据日志内容制定恢复集群的方法。

日志文件存储路径

CN、DN、GTM的运行日志放在“$GAUSSLOG/gs_log”中各自对应的目录下。

CM（包括cm_agent、cm_server、cm_ctl、om_monitor、etcd）的日志放在“$GAUSSLOG/cm”中各自对应的目录下。

版权所有 © 华为云计算技术有限公司

OM集群安装卸载时产生的日志放在“$GAUSSLOG/om”目录下。

日志文件命名格式

- CN和DN运行日志的命名规则：

  gaussdb-创建时间.log

  默认情况下，每日0点或者日志文件大于16MB或者集群（CN、DN）重新启动后，会生成新的日志文件。

  system_alarm-创建时间-current.log

- GTM运行日志的命名规则：

  gtm-创建时间.log

  默认情况下，日志文件大于16MB或者GTM实例重新启动后，会生成新的日志文件。

- CM的日志的命名规则：
  - cm_agent的日志：cm_agent-创建时间.log、cm_agent-创建时间-current.log；system_alarm-创建时间-current.log、system_call-current.log、system_stat-current.log、gaussdb-创建时间_000000.log。
  - cm_server的日志：cm_server-创建时间.log、cm_server-创建时间-current.log；key_event-创建时间-current.log、system_alarm-创建时间-current.log。
  - cm_ctl的日志：cm_ctl-创建时间-current.log。
  - om_monitor的日志：om_monitor-创建时间.log、om_monitor-创建时间-current.log；system_alarm-创建时间-current.log。

  其中，不带current标识符的文件是历史日志文件，带current标识符的文件是当前日志文件。最初调用进程时，进程会先创建一个带current标识符的日志文件，当该日志文件的大小超过16MB时，会将当前日志文件重命名为历史日志文件，并以当前时间生成新的日志文件。

  说明

  - 前缀为cm_agent、cm_server、cm_ctl、om_monitor的日志分别是cm_agent、cm_server、cm_ctl、om_monitor的运行日志。
  - 前缀为system_alarm的日志分别是cm_agent、cm_server、om_monitor的告警日志，记录集群运行时异常情况的告警上报与清除。
  - 前缀为key_event的日志是cm_server的关键事件日志，记录cm_server仲裁相关信息。
  - 前缀为system_call的日志是cm_agent在运行过程中调用工具命令的情况记录日志。
  - 前缀为system_stat的日志是cm_agent在运行过程中系统的性能监控日志，记录当前系统的磁盘I/O性能。
  - 前缀为gaussdb的日志是启动fenced模式的运行日志。
  - 日志命名依赖于系统时间戳，数据库会根据时间对旧的日志进行压缩，如果系统发生时间倒退，可能会把正在写入的日志文件提前压缩，导致日志文件无法继续写入，需要手动清理未来时间点日志进行恢复。

日志内容说明

- CN、DN每一行日志内容的默认格式：

  日期+时间+节点名称+用户名称+数据库名称+IP地址+会话ID+事务ID+应用程序名称+日志级别+日志内容

版权所有 © 华为云计算技术有限公司

表 5-3 CN、DN 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss.ms。
节点名称	表示指定错误上报的节点。
用户名称	当前触发生成日志的用户名。
数据库名称	当前触发生成日志的数据库名。
IP地址	当前触发生成日志的客户端的IP地址。
线程ID	线程ID。
会话ID	当前触发生成日志的session的session id。
事务ID	事务ID（0表示没有分配事务ID）。
线程名	线程名。
查询ID	query_id，用户发起查询时，后台记录此查询的ID。
模块名称	模块名称。
日志级别	日志的级别，不同级别表示的严重程度不一样，有FATAL、ERROR、LOG等级别。
日志内容	日志内容。

- GTM每一行日志内容的默认格式：
  日志行数+线程ID+日期+时间+日志级别+日志内容

表 5-4 GTM 日志参数

字段名称	字段含义
日志行数	当前生成日志可能会包含多行，日志行数表示日志内容处于本条日志的第几行。
线程ID	当前生成日志的线程ID号。
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss.ms。
日志级别	日志的级别，不同级别表示的严重程度不一样，有FATAL、ERROR、LOG等级别。
日志内容	日志内容。

- CN、DN的system_alarm日志每一行内容的默认格式：告警ID+告警名称+告警级别+主机所属领域列表+告警来源标识+告警操作类型+告警描述+清除类型+开始时间戳+结束时间戳，日志参数详细请参见**表5-5**。

版权所有 © 华为云计算技术有限公司

表 5-5 CN、DN、cm_agent、cm_server、om_monitor 的 system_alarm 日志参数

字段名称-英文	字段名称-中文	字段含义
id	告警ID	告警的ID。
name	告警名称	告警的名称。
level	告警级别	告警的级别，不同级别表示不同的严重程度，包括critical、major、minor、notice等级别。
scope	主机所属领域列表	主机所属领域列表。指使用newsql的告警平台，固定取值newsql。
source_tag	告警来源标识	告警的来源标识。
op_type	告警操作类型	告警的操作类型。 ● firing表示告警上报。当为firing时，日志中包含告警描述和清除类型。 ● resolved表示告警清除。当为resolved时，日志中不包含告警描述和清除类型。
details	告警描述	告警的具体信息。
clear_type	清除类型	告警的清除类型。包括ADAC和ADMC。 ● ADAC：系统能够自动检测到该故障发生，也能自动检测到该故障被修复。 ● ADMC：系统能够自动检测到该故障发生，但无法检测到该故障的修复情况。
start_timestamp	开始时间戳	告警开始的时间。 ● 上报告警时，start_timestamp时间戳有值，end_timestamp值为0。 ● 解除告警时，start_timestamp值为0，end_timestamp时间戳有值。
end_timestamp	结束时间戳	告警结束的时间。

- cm_agent、cm_server、cm_ctl、om_monitor日志内容如下所示：
  - cm_agent、cm_server、cm_ctl、om_monitor的运行日志每一行内容的默认格式：日期+时间+线程ID+线程名+日志级别+日志内容。
  - cm_agent、cm_server、om_monitor的system_alarm日志每一行内容的默认格式：告警ID+告警名称+告警级别+主机所属领域列表+告警来源标识+告警操作类型+告警描述+清除类型+开始时间戳+结束时间戳。
  - cm_server的key_event日志每一行内容的默认格式：日期+时间+线程ID+线程名+关键事件类型+仲裁对象实例ID+仲裁细节。

版权所有 © 华为云计算技术有限公司

- cm_agent的system_call日志大部分内容的默认格式：日期+时间+节点名称+用户名称+数据库名称+IP地址+会话ID+事务ID+应用程序名称+日志级别+日志内容。日志参数详细请参见**表5-3**。
- cm_agent的system_stat日志内容的默认格式：时间+操作系统+CPU信息+磁盘读写情况。

表 5-6 cm_agent、cm_server、cm_ctl、om_monitor 的运行日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss.ms。
线程ID	生成当前日志的线程ID号。
线程名	生成当前日志的线程名称。（cm_ctl、om_monitor运行日志无该参数。）
日志级别	日志的级别，不同级别表示不同的严重程度，包括LOG、WARNING、ERROR、FATAL等级别。
日志内容	日志内容。

表 5-7 cm_server 的 key_event 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss.ms。
线程ID	生成当前日志的线程ID号。
线程名	生成当前日志的线程名称。
关键事件类型	关键事件的类型，格式：[KeyEvent: xxxx]。
关键事件对象实例ID	关键事件的对象，格式：[Instance: xxxx]。
关键事件细节	关键事件的细节，格式：[Details: xxxx]。

表 5-8 cm_agent 的 system_stat 日志参数

字段名称	字段含义
时间	日志生成的时间，使用中国标准时间（CST）表示。
操作系统	主机的操作系统。

版权所有 © 华为云计算技术有限公司

字段名称	字段含义
CPU信息	CPU使用率。其中： ● %user：CPU处在用户模式下的时间百分比。 ● %nice：CPU处在带NICE值的用户模式下的时间百分比。 ● %system：CPU处在系统模式下的时间百分比。 ● %iowait：CPU等待输入输出完成时间的百分比。 ● %steal：管理程序维护另一个虚拟处理器时，虚拟CPU的无意识等待时间百分比。 ● %idle：CPU空闲时间百分比。

版权所有 © 华为云计算技术有限公司

字段名称	字段含义
磁盘读写	磁盘I/O信息。其中： ● rrqm/s：每秒合并的读请求次数，即读请求合并为更大的块的次数。 ● wrqm/s：每秒合并的写请求次数，即写请求合并为更大的块的次数。 ● r/s：每秒完成的读请求次数。 ● w/s：每秒完成的写请求次数。 ● rMB/s：每秒读取的数据量，单位为MB/s。 ● wMB/s：每秒写入的数据量，单位为MB/s。 ● r_await：平均每个读请求的等待时间，单位为毫秒。 ● w_await：平均每个写请求的等待时间，单位为毫秒。 ● %util：磁盘利用率，即磁盘I/O操作占用磁盘总时间的百分比。 ● %rrqm：读取请求在发送到设备之前合并的百分比。 ● %wrqm：写入请求在发送到设备之前合并的百分比。 ● rareq-sz：向设备发出的读请求的平均大小，单位为kB。 ● wareq-sz：向设备发出的写请求的平均大小，单位为kB。 ● d/s：设备每秒完成的丢弃请求数。 ● dMB/s：每秒设备丢弃的扇区数。 ● drqm/s：每秒合并到设备排队的丢弃请求数。 ● %drqm：丢弃请求在发送到设备之前合并的百分比。 ● d_await：向要服务的设备发出丢弃请求的平均时间。 ● dareq-sz：向设备发出的丢弃请求的平均大小，单位为kB。 ● f/s：设备每秒完成的刷新请求数（合并后）。 ● f_await：向要服务的设备发出的刷新请求的平均时间。 ● aqu-sz：向设备发出的请求的平均队列长度。

日志示例

- CN、DN
  - 运行日志的典型样例：

版权所有 © 华为云计算技术有限公司

```
2023-09-14 21:28:52.647 dn_6001 rdsAdmin postgres localhost 139983342597888 0[0:0#0] 0 dn_6001 0 [BACKEND] LOG:  will update control file (create checkpoint), shutdown:0
```

  - system_alarm日志的典型样例：

```
{"id":"0000001078919222","name":"AbnormalDataInstConnAuthMethod","level":"critical","scope":["newsql"],"source_tag":"szvphis5033219-cn_5001","op_type":"firing","details":"Data instance cn_5001 remote connection trust method error","clear_type":"ADAC","start_timestamp":1694140757589,"end_timestamp":0}
```

- GTM运行日志的典型样例：

```
1:140027660965632:2023-09-05 09:38:51.267 -LOG:  switched to running as GTM-primary.
```

- cm_agent
  - 运行日志的典型样例：

```
2023-09-05 05:59:59.845 tid=13690 CheckSysStatus ASYN LOG: [MEMORY STAT] memory system usage: 91.60%, application usage: 97.42%.
```

  - system_alarm日志的典型样例：

```
{"id":"0000001078919202","name":"DatanodeFailOver","level":"major","scope":["newsql"],"source_tag":"pekphis351066-dn_6001","op_type":"firing","details":"Datanode standby dn_6001 fail over","clear_type":"ADMC","start_timestamp":1693035351709,"end_timestamp":0}
```

  - system_call日志的典型样例：

```
2023-09-04 15:37:07.509 dn_6001_6002_6003 [unknown] [unknown] localhost 140252967524864 0[0:0#0] 0 [unknown] 0 [BACKEND] LOG:  create gaussdb state file success: db state(STARTING_STATE), server mode(Pending), connection index(1)
```

  - system_stat日志的典型样例：

```
Fri Aug 25 16:58:19 CST 2023
Linux 3.10.0-862.14.1.5.h687.eulerosv2r7.x86_64 (pekphis351066)         08/25/2023      _x86_64_        (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          18.02    0.00   10.07    0.41    0.00   71.50

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
vda               0.01     7.89    0.08    3.67     0.00     0.05    26.96     0.01    2.57    3.41    2.55    0.18   0.07
vdb               0.04    51.39    0.53  103.88     0.09     0.65    14.63     0.09    0.84    4.85    0.82    0.39   4.06
```

  - gs_log目录下的gaussdb日志的典型样例：

```
[UDF] LOG:  Fenced master process start OK.
```

- cm_server
  - 运行日志的典型样例：

```
2023-09-04 16:03:50.693 tid=95109 MAIN LOG: process startup packet, remote_type 7, nodeid 1, node name pekphis351066, postmaster is false.
```

  - system_alarm日志的典型样例：

```
{"id":"0000001078919258","name":"ServerSwitchOver","level":"minor","scope":["newsql"],"source_tag":"pekphis351066-server_1","op_type":"firing","details":"Server standby server_1 switch over","clear_type":"ADMC","start_timestamp":1693031871104,"end_timestamp":0}
```

  - key_event日志的典型样例：

```
2023-09-04 15:45:10.468 tid=95440 AGENT_WORKER: [KeyEvent: KEY_EVENT_SWITCHOVER] [Instance: 6001] [Details: send switchover message, node=1, instance=6001]
```

- cm_ctl
  - 运行日志的典型样例：

```
2023-08-28 16:39:54.113 tid=72942  LOG: set cm_server.conf success.
```

- om_monitor
  - 运行日志的典型样例如下所示，CM Agent启动检查完成，cluster_manual_start、agent_config_file_r和agent_binary_file_x文件存在，config_change_flag文件不存在，CM Agent之前状态为进程不在，拉起CM Agent次数为0：

 版权所有 © 华为云计算技术有限公司

```
2023-09-05 10:50:41.044 tid=100802  LOG: The CM Agent startup check is complete: cluster_manual_start=1, agent_config_file_r=1, agent_binary_file_x=1, config_change_flag=0, previous_status=0, start_count=0.
```

- system_alarm日志的典型样例：

```
{"id":"0000001078919226","name":"AbnormalCMAProcess","level":"critical","scope":["newsql"],"source_tag":"pekphis351066-cma","op_type":"resolved","start_timestamp":0,"end_timestamp":1692953836127}
```

日志丢弃机制

当系统日志写入异常时（如日志盘故障、日志盘负载过高等），系统会先进行重试（默认重试超时60秒），若重试仍失败则执行日志丢弃，以避免因日志写入阻塞而影响业务。当日志发生丢弃时，会在操作系统日志中打印告警信息：WARNING: logger pipe write failed, dropped a log message.

5.3 操作日志

操作日志是指数据库管理员使用工具操作数据库时以及工具被集群调用时产生的日志。如果集群发生故障，可以通过这些日志信息跟踪用户对数据库进行了哪些操作，重现故障场景。

日志文件存储路径

默认在“$GAUSSLOG/om”和“$GAUSSLOG/bin”目录下，如果环境变量$GAUSSLOG不存在或者变量值为空，则工具日志信息不会记录到对应的工具日志文件中，日志信息只会打印到屏幕上。

日志文件命名格式

日志文件命名格式为：

- 工具名-日志创建时间.log
- 工具名-日志创建时间-current.log

其中，“工具名-日志创建时间.log”是历史日志文件，“工具名-日志创建时间-current.log”是当前日志文件。

如果日志大小超过16MB，在下一次调用该工具时，会重命名当前日志文件为历史日志文件，并以当前时间生成新的当前日志文件。例如将“gs_guc-2015-01-16_183728-current.log”重命名为“gs_guc-2015-01-16_183728.log”，然后重新生成“gs_guc-2015-01-17_142216-current.log”。

日志内容说明

- OM工具日志，其包含的gs_install、gs_preinstall、gs_om、gs_expand、gs_replace、gs_local、checkRunStatus（运维操作状态检查接口）、JsonToDbclusterInfo（集群拓扑Json转xml工具）、killall_log、uploader(pscp工具)、cmd_sender(pssh工具)的每一行日志内容的默认格式：

  调用链ID+日期时间+线程ID+日志上报位置+线程名+日志级别+[Step*]+日志内容

版权所有 © 华为云计算技术有限公司

表 5-9 OM 包含的工具或者组件的日志参数

字段名称	字段含义
调用链ID	调用链ID。
日期时间	日志生成的日期和时间，格式：yyyy-mm-dd hh:mm:ss.ms。
线程ID	当前生成日志的线程ID号。
日志上报位置	当前生成日志可能会包含多行，日志上报位置表示日志内容处于执行文件的第几行。
线程名	当前生成日志的线程名称。
日志级别	日志的级别，不同级别表示不同的严重程度，有LOG、DEBUG、WARNING、ERROR等级别。
Step*	当前时间执行的步骤编号。
日志内容	日志内容。

- gs_initdb日志内容的默认格式：
  日期+时间+日志级别+日志内容

  表 5-10 gs_initdb 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss。
日志级别	日志的级别，不同级别表示不同的严重程度，有LOG、WARNING、ERROR、FATAL等级别。
日志内容	日志内容。

- gs_guc日志内容的默认格式：
  日期+时间+日志内容

  表 5-11 gs_guc 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss。
日志内容	日志内容。

- gs_cgroup每一行日志内容的默认格式：
  调用链ID+日期时间+线程ID+日志上报位置+线程名+日志级别+日志内容

版权所有 © 华为云计算技术有限公司

表 5-12 gs_cgroup 日志参数

字段名称	字段含义
调用链ID	调用链ID。
日期时间	日志生成的日期 时间，格式：yyyy-mm-dd hh:mm:ss.ms。
线程ID	当前生成日志的线程ID号。
日志上报位置	当前生成日志可能会包含多行，日志上报位置表示日志内容处于执行文件的第几行。
线程名	当前生成日志的线程名称。
日志级别	日志的级别，不同级别表示不同的严重程度，有LOG、DEBUG、WARNING、ERROR等级别。
日志内容	日志内容。

- gs_clean每一行日志内容的默认格式：
  日期+时间+日志内容

  表 5-13 gs_clean 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss。
日志内容	日志内容。

- gs_ctl每一行日志内容的默认格式：
  日期+时间+线程ID+node_name+线程名+日志内容

  表 5-14 gs_ctl 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss.ms。
线程ID	当前生成日志的线程ID号。
node_name	实例名称。
线程名	当前生成日志的线程名称。
日志内容	日志内容。

- gs_initgtm每一行日志内容的默认格式：
  日期+时间+日志内容

版权所有 © 华为云计算技术有限公司

表 5-15 gs_initgtm 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss。
日志内容	日志内容。

- gs_obs每一行日志内容的默认格式：
  日期+时间+日志级别+线程ID+日志内容

  表 5-16 gs_obs 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss.ms。
日志级别	日志的级别，不同级别表示不同的严重程度，有LOG、WARNING、ERROR、FATAL等级别。
线程ID	当前生成日志的线程ID号。
日志内容	日志内容。

- gtm_ctl日志内容的默认格式：
  日期+时间+日志内容

  表 5-17 gtm_ctl 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss。
日志内容	日志内容。

- gs_redis每一行日志内容的默认格式：
  日期+时间+时间戳+线程名+线程ID+日志上报位置+step[*]+日志内容

  表 5-18 gs_redis 日志参数

字段名称	字段含义
日期	日志生成的日期，格式：yyyy-mm-dd。
时间	日志生成的时间，格式：hh:mm:ss:ms。
时间戳	当前时间的时间戳。

版权所有 © 华为云计算技术有限公司

字段名称	字段含义
线程名	打印日志的线程名。
线程ID	当前生成日志的线程ID号。
step[*]	当前时间执行的步骤编号。
日志内容	日志内容。

日志示例

- OM工具日志的典型样例：
```
[220822164bc111ee9466fa163e8dd000][2023-09-05 08:02:11.098083][11214]
[InstallImpl.py(doStart:864)][gs_install][DEBUG][Step7]:Successfully started the cluster.
```
- gs_initdb日志的典型样例：
```
[2023-09-05 07:57:01]

WARNING: enabling "trust" authentication for local connections
You can change this by editing gs_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run gs_initdb.
```
- gs_guc日志的典型样例：
```
[2023-09-05 09:05:06]
Begin to perform the total nodes: 3.
Popen count is 3, Popen success count is 3, Popen failure count is 0.
Begin to perform gs_guc for datanodes.
[2023-09-05 09:05:07]
expected instance path: [/var/lib/engine/data1/data/dn_6001/gaussdb.conf]
expected instance path: [/var/lib/engine/data1/data/dn_6005/gaussdb.conf]
gs_guc reload: key=value: [/var/lib/engine/data1/data/dn_6001/gaussdb.conf]
server signaled
gs_guc reload: key=value: [/var/lib/engine/data1/data/dn_6005/gaussdb.conf]
server signaled

Total instances: 2. Failed instances: 0.
Success to perform gs_guc!

Command count is 3, Command success count is 3, Command failure count is 0.

Total instances: 6. Failed instances: 0.
ALL: Success to perform gs_guc!
```
- gs_cgroup日志的典型样例：
```
[de86987e4bc011eeb3befa163e8dd000][2023-09-05 15:51:49.372997][947]
[PreInstallUtility.py(setCgroup:1793)][gs_cgroup][DEBUG]:The gs_cgroup run with the following
arguments: [/var/chroot/usr/local/package/dependency/bin/gs_cgroup -U Ruby --upgrade -c -H /var/
chroot/usr/local/tool/Ruby ].
'/sys/fs/cgroup/cpu,cpuacct/CM:Ruby' exists, omit to create this cgroup.
'/sys/fs/cgroup/cpu,cpuacct/CM:Ruby' exists, omit to create this cgroup.
```
- gs_clean日志的典型样例：
```
2023-09-05 09:06:57  gs_clean: connected successfully
```
- gs_ctl日志的典型样例：
```
[2023-09-05 08:00:59.538][4053][dn_6004_6005_6006][gs_ctl]: finish fsync all files.
```
- gs_initgtm日志的典型样例：
```
[2023-09-05 09:05:06] gs_initgtm: input file location must be an absolute path.
```
- gs_obs日志的典型样例：
```
2025-01-21 11:06:42.867| INFO|[140565224322816]|Enter complete_multi_part_upload successfully !
```
- gtm_ctl日志的典型样例：
```
[2023-09-05 08:00:40]
 Set GTM syncmode succeeded.
```


版权所有 © 华为云计算技术有限公司

- gs_redis日志的典型样例：
  ```
  2023-09-05 08:53:01  1693903981339 worker 140463824475904
  [snapshot.snap_summary_stat_all_indexes] step[2]: END successfully
  ```

维护建议

建议定时对过期的日志文件进行转储，以避免大量日志占用太多的磁盘空间和避免重要日志丢失。

5.4 审计日志

数据库安全管理员可以利用审计日志信息，重现导致数据库现状的一系列事件，找出非法操作的用户、时间和内容等。审计功能开启时会不断产生大量的审计日志，占用磁盘空间。用户可以根据磁盘空间的大小设置审计日志维护策略。

日志存储路径

默认$GAUSSLOG/pg_audit，如果使用om工具部署集群，则审计日志路径为“$GAUSSLOG/pg_audit/实例名称”。日志存储路径支持由GUC参数audit_directory自主配置路径。

日志查询方法

只有拥有AUDITADMIN属性的用户才可以查看审计记录。查询方法如下：

- 查询传统审计记录。
  ```
  gaussdb=# SELECT * FROM gs_query_audit('2021-03-03 21:30:00', '2021-03-03 22:00:00') where type = 'login_success' and username = 'user1';
  ```
- 查询所有CN节点传统审计记录。
  ```
  gaussdb=# SELECT * FROM gs_query_audit_dist('2021-03-03 21:30:00', '2021-03-03 22:00:00') where type = 'login_success' and username = 'user1';
  ```
- 查询统一审计记录。

```
gaussdb=# SELECT * FROM gs_query_unified_audit('2025-04-26 08:00:00','2025-04-26 17:00:00')where type = 'login_success' and username = 'user1';
```

starttime和endtime的差值代表要查询的时间段，其有效值为从starttime日期中的00:00:00开始到endtime日期中的23:59:59之间的任何值。请正确指定这两个参数，否则将查不到需要的审计信息。

日志内容说明

日志中各个字段的描述参见gs_query_audit函数定义，如**表5-19**。

表 5-19 传统审计日志参数

名称	描述
time	操作时间。
type	操作类型。
result	操作结果。
userid	用户id。

版权所有 © 华为云计算技术有限公司

名称	描述
username	执行操作的用户名。
database	数据库名称。
client_conninfo	客户端连接信息。
object_name	操作对象名称。
detail_info	执行操作详细信息。
node_name	节点名称。
thread_id	线程id。
local_port	本地端口。
remote_port	远端端口。

日志中各个字段的描述参见gs_query_unified_audit函数定义，如**表5-20**。

表 5-20 统一审计日志参数

名称	描述
time	操作时间。
type	操作类型。
result	操作结果。
userid	用户ID。
username	执行操作的用户名。
database	数据库名称。
client_conninfo	客户端连接信息。
object_name	操作对象名称。
detail_info	执行操作详细信息。
node_name	节点名称。
thread_id	线程ID。
local_port	本地端口。
remote_port	远端端口。
policy_id	统一审计策略ID。
unified_audit_type	统一审计策略类型。

版权所有 © 华为云计算技术有限公司

名称	描述
unified_audit_policy	统一审计策略信息。

5.5 WAL 日志

预写式日志WAL（Write Ahead Log，也称为Xlog）是指如果要修改数据文件，必须是在这些修改操作已经记录到日志文件之后才能进行修改，即在描述这些变化的日志记录刷新到永久存储器之后。在系统崩溃时，可以使用WAL日志对集群进行恢复操作。WAL日志记录的详细程度分为几个级别，由参数**wal_level**控制。

日志运行机制

GaussDB启动时，会分配一段内存用来缓存数据库运行期间生成的日志，这个缓冲区大小由参数wal_buffers控制。要确保事务的持久化，Wal_buffers中的日志就需要落盘。

WAL日志会在以下两种情况下写硬盘：

- Commit 事务提交时。系统会调用fdatasync函数，将wal_buffers中的WAL数据写入WAL日志文件。（wal_sync_method可控制commit时调用的函数，默认为fdatasync）。
- walWriter定期写。 walWriter进程会定期将wal_buffers中的WAL数据写入WAL日志文件。wal_writer_delay 参数用来控制walwriter进程工作的时间间隔，默认200ms。如果时间过长可能造成WAL缓冲区的内存不足，时间过短会引起WAL不断写入，增加磁盘I/O负担。

日志文件存储路径

路径为数据目录/pg_xlog。

- CN的WAL日志存储路径为 /<cn数据目录>/pg_xlog。单个WAL日志文件的大小默认为16MB，通过参数wal_segment_size设置，一般不做调整。

```
gaussdb=# SHOW wal_segment_size;
wal_segment_size
------------------
16MB
(1 row)
```

- 以一个DN为例，数据目录可以通过如下命令查询。其中/opt/gaussdb/data/dn/db_6001代表集群节点的数据目录，当前节点的Xlog日志存放在 /opt/gaussdb/data/dn/db_6001/pg_xlog目录。

```
$ cm_ctl query -Cvd
[    CMServer State    ]
node                    instance                state
-----------------------------------------------------------------------------
1    192.168.10.182    1    /opt/gaussdb/data/cm/cm_server    Primary

[   Cluster State   ]

cluster_state  : Normal
redistributing : No
balanced       : Yes
current_az : AZ_ALL
```


版权所有 © 华为云计算技术有限公司

```
[    Datanode State    ]

node                        instance               state
---------------------------------------------------------------------------
1    192.168.10.182 6001    /opt/gaussdb/data/dn/dn_6001 P Primary Normal
```

日志文件命名格式

日志文件以段文件的形式存储的，每个段为16MB，并分割成若干页，每页8KB。一个段文件的名称由24个十六进制组成，对WAL日志的命名由如下部分组成：时间线+日志文件标号+日志文件段标号，请参考表5-21。

日志文件的段标号上限为FF，若有下一个WAL日志文件，需要将文件标号+1，段标号归零，例如000000010000000000000FF的后一个文件为000000010000000100000000。时间线由1开始，日志文件标号和日志文件的段标号由0开始。

表 5-21 WAL 日志文件命名规范

字段名称	字段含义
timeline	时间线。取值范围：8位16进制数字。
wal_file_num	日志文件标号。取值范围：8位16进制数字。
wal_segment_num	日志文件段标号。取值范围：8位16进制数字。

典型的WAL日志文件名如下所示，每个WAL日志文件大小均为16MB。

```
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 11    12:33 000000010000000100000F2
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 11    18:10 000000010000000100000F3
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 11    22:18 000000010000000100000F4
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 12    03:58 000000010000000100000F5
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 12    07:59 000000010000000100000F6
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 12    13:45 000000010000000100000F7
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 12    17:30 000000010000000100000F8
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 12    23:10 000000010000000100000F9
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 13    03:10 000000010000000100000FA
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 13    08:39 000000010000000100000FB
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 13    12:43 000000010000000100000FC
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 13    17:08 000000010000000100000FD
-rW----- 1 gaussdb d gaussdb d 16777216 Jun 13    16:32 000000010000000100000FE
```

说明

WAL日志文件名的数字一般情况下是顺序增长使用的（要把所有可用数字都用光也需要非常长的时间），但也存在循环使用的情况。

日志内容说明

WAL日志的内容取决于记录事务的类型，在系统崩溃时可以利用WAL日志进行恢复。

默认配置下，GaussDB每次启动时会先读取WAL日志进行恢复。

日志使用场景

数据库崩溃恢复：数据库意外崩溃时，因为Xlog日志中已经完整记录了数据库的变化，在数据库重启后就可以通过回放checkpoint之后的Xlog日志，把之前修改过的数

版权所有 © 华为云计算技术有限公司

据页面（记录在Xlog里，但是还没有来得及修改数据文件）落盘，确保数据的完整性。这个过程是在数据库启动后自动完成的。

主备DN数据同步：GaussDB的备DN通过接收来自主DN的Xlog，然后应用Xlog来达到主备一致。主库和备库之间的数据同步也可以通过Xlog来实现。

数据库备份恢复：用GaussRoach对GaussDB做备份时，不仅要复制数据文件，还需要备份期间产生的Xlog，才能构成一个完整的备份。使用全量备份或增量备份+Xlog，可以做到指定时间点的恢复，即恢复到全量备份之后的指定时间点（即PITR: Point-In-Time Recovery）。

逻辑复制：逻辑复制工具，如DRS，是通过对Xlog的解析来获取数据流并进行增量数据同步的。

Xlog 的回收

数据库运行期间，会持续生成Xlog日志，GaussDB通过checkpoint机制来进行Xlog日志清理（通常被称为Xlog回收）。Xlog日志回收相关参数请参见**表5-22**，参数设置方式请参见**设置参数**。

表 5-22 Xlog 日志回收相关参数

参数名称	参数说明
checkpoint_timeout	设置自动检查点之间的最长时间。在上一次自动检查点后，多长时间必须做下一次自动检查点，默认值为15min。如果在两次自动检查点期间堆积的Xlog日志数量大于checkpoint_segments设置的值，也会触发一次自动检查点。
checkpoint_segments	设置**checkpoint_timeout**周期内所保留WAL日志文件的最大数量。每个日志文件大小为16MB。
enable_incremental_checkpoint	增量检查点开关，默认值为on。
incremental_checkpoint_timeout	开启增量检查点开关（**enable_incremental_checkpoint**）之后，设置自动检查点之间的最长时间。 默认值为1min。即每分钟会启动一次增量检查点，同样会检查是否有Xlog需要清理。
wal_keep_segments	Xlog日志文件的数量，设置“pg_xlog”目录下保留日志文件的最小数量。备机通过获取主机的Xlog日志进行流复制。默认值为128。一般情况下，数据库在执行checkpoint时，会检查当前的pg_xlog目录下的Xlog数量是否大于wal_keep_segments设置的值， 如果数量大于wal_keep_segments值，就从旧的一端开始清理，保留wal_keep_segments设定数量的Xlog日志。

版权所有 © 华为云计算技术有限公司

参数名称	参数说明
enable_xlog_prune	设置在任一备机断连时，主机是否根据Xlog日志的大小超过参数max_size_for_xlog_prune的值而回收日志，on为开启。在备机断联时，为确保主备DN间Xlog日志是连续的，将不对Xlog日志进行清理。但是，如果开启enable_xlog_prune， 且Xlog堆积的容量大于max_size_for_xlog_prune指定的值，一般为256GB，同样会触发Xlog日志清理。
max_size_for_xlog_prune	在备机故障时主机保留的Xlog最大数量。在enable_xlog_prune打开时生效，工作机制如下： 1. replconninfo系列guc参数配置的所有备机都连接主机时，则该参数不生效。 2. replconninfo系列guc参数配置的备机存在断连时，则该参数生效。当主机Xlog日志量大于该参数值，会强制回收。例外：在同步提交模式下（即synchronous_commit参数不是local/off时），如果存在连接中的备机，则主机会考虑保留多数派备机中最小日志接受位置以后的日志，这种情况下，保留的日志可能多于max_size_for_xlog_prune参数值。 3. 若存在build中的备机，则该参数不生效，主机日志会全量保留，防止build操作期间由于日志被回收引发的失败。

维护建议

WAL日志对数据库异常恢复有重要的作用，建议定期对WAL日志进行备份。

5.6 内存日志

内存日志是保存数据库进程动态内存使用超过设定阈值，或动态内存波动超过设定阈值，或统计周期达到设定值时生成的内存上下文统计信息，由GUC参数memory_trace_level控制是否记录，默认为level1记录日志；由GUC参数memory_log_config设置更多功能，如配置动态内存使用量百分比阈值、内存波动百分比阈值、统计周期、内存上下文top数量、SQL语句top数量。

说明

- 内存日志的生成有防振荡机制，当动态内存超过阈值（memory_log_config参数第1部分设置）时，生成内存日志时的动态内存会记为触发内存。此后，若动态内存若持续高于(触发内存-动态内存上限*5%)，将不会再因超过阈值而生成内存日志，从而避免在阈值附近小幅振荡时反复生成日志。
- 按“内存波动百分比”条件（memory_log_config参数第2部分设置）触发生成内存上下文快照是基于3秒检测周期的，若在3秒检测周期内满足生成快照条件，不会生成内存上下文快照。
- 内存阈值触发、内存波动触发和周期性触发不会同时进行。
- 仅支持对名为{模块名}_{数字}格式的内存上下文按模块名合并，不符合此特点的不会合并，例如“pooler_memory_context”、“pooler_core_context”、“pooler_agent_context”是3种不同的内存上下文，显示时不会合并。

版权所有 © 华为云计算技术有限公司

日志文件存储路径

数据库节点的内存日志目录在“$GAUSSLOG/mem_log”目录下。

日志文件命名格式

数据库节点的内存日志的命名规则：mem_log-创建时间.log。

日志内容说明

数据库进程的全局内存上下文、会话级内存上下文、线程级内存上下文的详细使用统计信息，包含占用内存top20的会话级和线程级以及对应的客户端类型和SQL语句等信息，如果存在运行的并行逻辑解码任务，日志中也会包含并行逻辑解码的内存使用信息。

每种类型的内存上下文、每个query的内存上下文按内存使用量对排序最多显示top n个（由memory_log_config参数的第4部分控制），query信息最多显示top n个（由memory_log_config参数的第5部分控制）。

表 5-23 日志参数

字段名称	字段含义
Max_dynamic_memory	数据库进程最大可使用的动态内存。
Dynamic_used_memory	当前进程已使用的动态内存大小。
Dynamic_peak_memory	当前进程动态内存使用的历史最大值。
Dynamic_thread_cxt_used_memory	当前进程线程级内存上下文已使用内存大小。
Dynamic_session_cxt_used_memory	当前进程会话级内存上下文已使用内存大小，仅在启用线程池时显示。
Dynamic_used_shrctx	当前进程全局内存上下文占用的内存大小。
Dynamic_peak_shrctx	当前进程全局内存上下文占用的历史最大值。
Max_backend_memory	HA端口连接的会话申请的最大内存大小。
Backend_used_memory	HA端口连接的会话正在占用的内存大小。
other_used_memory	未在内存上下文上申请的其他内存占用大小。
unit	同级内存量的单位。
Context Type	内存上下文类型，有“Shared Memory Context”，“Thread Memory Context”，“Session Memory Context”三种类型。

版权所有 © 华为云计算技术有限公司

字段名称	字段含义
context	内存上下文的名称。同名、同模块名（格式为{模块名}_{数字}）的内存上下文会合并显示，名称只显示模块名。
freeSize	内存上下文上缓存的空闲内存的大小。
totalSize	内存上下文申请的总的内存大小。
file	内存申请所在的代码的文件名，该参数仅在memory_trace_level=level2时记录。
line	内存申请所在的代码的文件行号，该参数仅在memory_trace_level=level2时记录。
size	某一文件某一行号处申请的内存的累计大小，该参数仅在memory_trace_level=level2时记录。
count	合并统计的同名、同模块名内存上下文数量。
query	执行的语句。
unique query id	语句唯一ID。
debug query id	语句调试ID。
session id	会话ID。
thread id	线程ID。
mem size	语句申请内存大小。
application name	应用名。

日志示例

- memory_trace_level=level1时记录日志如下所示：

```
{
  "Global Memory Statistics": {
  "Max_dynamic_memory": 34012,
  "Dynamic_used_memory": 3645,
  "Dynamic_peak_memory": 3664,
  "Dynamic_thread_cxt_used_memory": 383,
  "Dynamic_session_cxt_used_memory": 2582,
  "Dynamic_used_shrctx": 401,
  "Dynamic_peak_shrctx": 401,
  "Max_backend_memory": 648,
  "Backend_used_memory": 1,
  "other_used_memory": 0,
  "unit": "MB"
 },
 "Memory Context Info": {
  "Memory Context Detail": {
   "Context Type": "Shared Memory Context",
   "Memory Context": {
    "context": "IncreCheckPointContext",
    "freeSize": 2864,
    "totalSize": 263066352,
    "unit": "Byte",
    "count": 1
   },
   ...
```


版权所有 © 华为云计算技术有限公司

```
 },
 "Memory Context Detail": {
  "Context Type": "Session Memory Context",
  "Memory Context": {
   "context": "CachedPlan",
   "freeSize": 68041368,
   "totalSize": 235937792,
   "unit": "Byte",
   "count": 1
  },
  ...
 },
 "Memory Context Detail": {
  "Context Type": "Thread Memory Context",
  "Memory Context": {
   "context": "LocalSysCacheShareMemoryContext",
   "freeSize": 60431360,
   "totalSize": 644141760,
   "unit": "Byte",
   "count": 1
  },
  ...
 }
"Query Info": {
 "Query Context": {
    "query": "select distinct a from t3 order by a;",
    "unique query id": "2286703632",
    "debug query id": "1407374883554034",
    "session id": "2",
    "thread id": "140188551411456",
    "mem size": 1,
    "unit": "MB",
    "application name": "gsql",
    "Memory Context Detail": {
       "Context Type": "Session Memory Context",
       "Memory Context": {
       "context": "SessionTopMemoryContext",
       "freeSize": 97048,
       "totalSize": 460520,
       "unit": "Byte",
       "count": 1
    },
    ...
  },
 "Query Context": {
    "query": "WLM fetch collect info from data nodes",
    "unique query id": "0",
    "debug query id": "0",
    "session id": "140186948200192",
    "thread id": "140186948200192",
    "mem size": 1,
    "unit": "MB",
    "application name": "workload",
    "Memory Context Detail": {
       "Context Type": "Thread Memory Context",
       "Memory Context": {
       "context": "LocalSysCacheShareMemoryContext",
       "freeSize": 150720,
       "totalSize": 1872096,
       "unit": "Byte",
       "count": 1
    },
    ...
  },
  ...
},
"parallel logical decode Info": [{
 "slot name": "slot1",
 "memory info": "memory usage: reader toast memory size: 0, sender out buf memory size: 0, [xid,
```


 版权所有 © 华为云计算技术有限公司

```
first_lsn, memory, entry num] : NULL"
 }]
}
```

- memory_trace_level=level2时记录日志如下所示：

```
{
  "Global Memory Statistics":  {
  "Max_dynamic_memory":  34012,
  "Dynamic_used_memory":  3645,
  "Dynamic_peak_memory":  3664,
  "Dynamic_thread_cxt_used_memory":  383,
  "Dynamic_session_cxt_used_memory":  2582,
  "Dynamic_used_shrctx":  401,
  "Dynamic_peak_shrctx":  401,
  "Max_backend_memory":  648,
  "Backend_used_memory":  1,
  "other_used_memory":  0,
  "unit":  "MB"
 },
 "Memory Context Info":  {
  "Memory Context Detail":  {
   "Context Type":  "Shared Memory Context",
   "Memory Context":  {
    "context":  "IncreCheckPointContext",
    "freeSize":  2864,
    "totalSize":  263066352,
    "unit":  "Byte",
    "count":  1,
    "MemoryDetail":  {
        "file":"xxx.cpp",
        "line":411,
        "size":211131,
     }
     ...
   },
  ...
  },
  "Memory Context Detail":       {
   "Context Type": "Session Memory Context",
   "Memory Context":      {
    "context":      "CachedPlan",
    "freeSize":     68041368,
    "totalSize":    235937792,
    "unit":    "Byte",
    "count":    1,
    "MemoryDetail":  {
        "file":"xxx.cpp",
        "line":411,
        "size":211131,
     }
     ...
   },
   ...
  },
  "Memory Context Detail":  {
   "Context Type":  "Thread Memory Context",
   "Memory Context":  {
    "context":  "LocalSysCacheShareMemoryContext",
    "freeSize":  60431360,
    "totalSize":  644141760,
    "unit":  "Byte",
    "count":  1,
    "MemoryDetail":  {
        "file":"xxx.cpp",
        "line":411,
        "size":211131,
     }
     ...
   },
   ...
```

 版权所有 © 华为云计算技术有限公司

```
  }
 "Query Info":  {
  "Query Context":  {
     "query":    "select distinct a from t3 order by a;",
     "unique query id":  "2286703632",
     "debug query id":  "1407374883554034",
     "session id":  "2",
     "thread id":  "140188551411456",
     "mem size":  1,
     "unit":  "MB",
     "application name":  "gsql",
     "Memory Context Detail":  {
        "Context Type":  "Session Memory Context",
        "Memory Context":  {
        "context":  "SessionTopMemoryContext",
        "freeSize":  97048,
        "totalSize":  460520,
        "unit":  "Byte",
        "count":  1,
        "MemoryDetail":  {
           "file":  "xxx.cpp",
           "line":  461,
           "size":  62976
        },
        ...
     },
     ...
   },
  "Query Context":  {
      "query":  "WLM fetch collect info from data nodes",
      "unique query id":  "0",
      "debug query id":  "0",
      "session id":  "140186948200192",
      "thread id":  "140186948200192",
      "mem size":  1,
      "unit":  "MB",
      "application name":  "workload",
      "Memory Context Detail":  {
         "Context Type":  "Thread Memory Context",
         "Memory Context":  {
         "context":  "LocalSysCacheShareMemoryContext",
         "freeSize":  150720,
         "totalSize":  1872096,
         "unit":  "Byte",
         "count":  1,
         "MemoryDetail":  {
            "file":    "xxx.cpp",
            "line":    352,
            "size":    40153344
         },
         ...
      },
      ...
    },
   ...
 },
 "parallel logical decode Info":  [{
  "slot name":  "slot1",
  "memory info":  "memory usage: reader toast memory size: 0, sender out buf memory size: 0, [xid,
first_lsn, memory, entry num] : NULL"
 }]
}
```


图形化内存信息 html

生成JSON格式的.log文件的同时，还会在同一目录生成名称相同的html文件的.gz压缩包。将压缩包解压、把html文件和同目录下的echarts.min.js文件导出至同一目录下，并使用浏览器打开html文件，可查看图形化的内存信息。

版权所有 © 华为云计算技术有限公司

内存信息与JSON格式的内存日志对应，由以下几个部分组成：

- Global Memory Statistics：显示各类别动态内存占最大动态内存的占比圆环图。
- Shared Memory Context：显示共享内存上下文的内存柱状图。若GUC参数memory_log_config第4部分设置为0则不会显示。
- Session Memory Context：显示会话内存上下文的内存柱状图。若GUC参数memory_log_config第4部分设置为0，或GUC参数enable_thread_pool=off，则不会显示。
- Thread Memory Context：显示线程内存上下文的内存柱状图。若GUC参数memory_log_config第4部分设置为0则不会显示。
- SQL Details：显示执行语句或线程的内存信息。若GUC参数memory_log_config第5部分设置为0则不会显示。当有内存上下文显示时，可通过单击每条语句条目来显示其top n内存上下文的内存柱状图。

说明

GUC参数memory_trace_level设置为level2时相比于level1，会多显示内存上下文申请内存的位置和大小信息，可通过单击每种内存上下文的柱状图来显示其申请内存的位置和大小。

各部分所展示信息的字段含义如下表所示。

表 5-24 展示信息

字段名称	字段含义
Dynamic used share ctx memory	当前进程已使用的动态内存大小。
Dynamic thread cxt used memory	当前进程线程级内存上下文已使用内存大小。
Dynamic session cxt used memory	当前进程会话级内存上下文已使用内存大小，仅在开启线程池时显示。
Dynamic unused memory	当前进程未使用内存。
Context Type	内存上下文类型，有“Shared Memory Context”，“Thread Memory Context”，“Session Memory Context”三种类型。
context	内存上下文的名称。同名、同模块名（格式为{模块名}_{数字}）的内存上下文会合并显示，名称只显示模块名。
totalSize	内存上下文申请的总的内存大小。
file	内存申请所在的代码的文件名，该参数仅在memory_trace_level=level2时记录。
line	内存申请所在的代码的文件行号，该参数仅在memory_trace_level=level2时记录。
size	某一文件某一行号处申请的内存的累计大小，该参数仅在memory_trace_level=level2时记录。
query	执行的语句。

版权所有 © 华为云计算技术有限公司

字段名称	字段含义
unique query id	语句唯一ID。
debug query id	语句调试ID。
session id	会话ID。
thread id	线程ID。
mem size	语句申请内存大小。
application name	应用名。

版权所有 © 华为云计算技术有限公司

6 数据库运行参数说明

6.1 查看参数

查看 GUC 参数

GaussDB安装后，有一套默认的GUC参数，为了使GaussDB与业务的配合度更高，用户需要根据业务场景和数据量的大小调整GUC参数。

操作步骤

步骤1 连接数据库，具体操作请参见《开发指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节。

步骤2 查看当前GUC参数的取值。

- 方法一：使用SHOW命令。
  - 使用如下命令查看单个参数：
    ```
    gaussdb=# SHOW client_encoding;
    ```
    client_encoding参数用于查看客户端的字符编码类型。
  - 使用如下命令查看所有参数：
    ```
    gaussdb=# SHOW ALL;
    ```
- 方法二：使用pg_settings视图。
  - 使用如下命令查看单个参数：
    ```
    gaussdb=# SELECT * FROM pg_settings WHERE NAME='client_encoding';
    ```
  - 使用如下命令查看所有参数：
    ```
    gaussdb=# SELECT * FROM pg_settings;
    ```

----结束

示例

查看客户端的字符编码类型。

```
gaussdb=# SHOW client_encoding;
 client_encoding
-----------------
 UTF8
(1 row)
```

 版权所有 © 华为云计算技术有限公司

查看 CM 参数

CM参数可以直接在文件中查看，也可以通过命令进行查看：

- 在文件中查看：

  cm_agent相关参数可通过cm_agent数据目录下的cm_agent.conf文件查看，cm_server相关参数可通过cm_server数据目录下的cm_server.conf文件查看。

- 通过命令查看：

  ```
  cm_ctl list --param {--agent | --server} [-n NODEID]
  ```

  参数说明

  - --agent：列出cm_agent参数。
  - --server：列出cm_server参数。
  - -n NODEID：列出指定节点的CM参数，NODEID为节点名称，可通过cm_ctl query -Cv命令查询。

  示例

  - 列出cm_agent参数：

    ```
    cm_ctl list --param --agent
    ```

  - 列出cm_server参数：

    ```
    cm_ctl list --param --server
    ```

6.2 设置参数

GUC 参数设置

GaussDB提供了六类GUC参数，以及多种修改GUC参数的方法，用户可以方便地针对数据库、用户、会话进行设置，具体分类和设置方式请参考**表6-1**。

- 参数名称不区分大小写。
- 参数取值有整型、浮点型、字符串、布尔型和枚举类型五类。
  - 布尔值可以是（on，off）、（true，false）、（yes，no）或者（1，0），GaussDB的布尔值推荐使用（on，off）。
  - 枚举类型的取值是系统表pg_settings的enumvals字段取值定义的。
  - 浮点（real）型的参数，必须加小数点，如 1.0e-3 是符合规范的，而1e-3是不符合规范的。
  - 在IEEE 754 标准中定义，对于浮点数，存在+0、-0，在对数值型参数进行设置时，零带负号可以通过校验。
  - gs_guc工具对于浮点数范围检查具有精度阈值，具体阈值为1.0*e-9。
- 对于有单位的参数，在设置时请指定单位，否则将使用默认的单位。
  - 参数的默认单位是系统表pg_settings的unit字段定义的。
  - 内存单位有：KB/kB（千字节）、MB/mB（兆字节）和GB/gB（吉字节）。
  - 时间单位：ms（毫秒）、s（秒）、min（分钟）、h（小时）和d（天）。
- CN和DN参数可以同时进行设置，其他类型的参数不能同时进行设置。

具体参数说明请参见**GUC参数说明**。

版权所有 © 华为云计算技术有限公司

表 6-1 GUC 参数分类

参数类型	说明	设置方式
INTERNAL	固定参数，在创建数据库的时候确定，用户无法修改，只能通过show语法或者pg_settings视图进行查看。	无
POSTMASTER	数据库服务端参数，在数据库启动时确定，可以通过配置文件指定。	支持表6-2中的方式一。
SIGHUP	数据库全局参数，可在数据库启动时设置或者在数据库启动后，发送指令重新加载。	支持表6-2中的方式一、方式二。
BACKEND	会话连接参数。在创建会话连接时指定，连接建立后无法修改。连接断掉后参数失效。内部使用参数，不推荐用户设置。	支持表6-2中的方式一、方式二。 **说明** 设置该参数后，下一次建立会话连接时生效。
SUSET	数据库管理员参数。可在数据库启动时、数据库启动后或者数据库管理员通过SQL进行设置。	支持表6-2中的方式一、方式二或由数据库管理员通过方式三设置。
USERSET	普通用户参数。可被任何用户在任何时刻设置。	支持表6-2中的方式一、方式二或方式三设置。 **说明** 设置USERSET类型的参数时，ALTER DATABASE设置的参数值优先级高于gs_guc设置。如果想要gs_guc设置的参数值生效，则需要执行“alter database *xxx* reset *xxx*”进行重置。

GaussDB提供了三种方式来修改GUC参数，具体操作请参考表6-2：

表 6-2 GUC 参数设置方式

序号	设置方法
方式一	1. 登录管理控制台。 2. 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例基本信息页面。 3. 在左侧导航栏单击“参数管理”，进入参数修改页面，在该页面修改参数。 如果需要修改的参数在管理控制台无法修改，请提前评估风险后再联系客服进行修改。 4. 重启数据库使参数生效。 **说明** 重启数据库集群操作会导致用户执行操作中断，请在操作之前规划好合适的执行窗口。

版权所有 © 华为云计算技术有限公司

序号	设置方法
方式二	1. 登录管理控制台。 2. 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例基本信息页面。 3. 在左侧导航栏单击“参数管理”，进入参数修改页面，在该页面修改参数。 如果需要修改的参数在管理控制台无法修改，请提前评估风险后再联系客服进行修改。
方式三	修改会话级别的参数。 • 设置会话级别的参数： gaussdb=# SET *paraname* TO *value*; 修改本次会话中的取值。退出会话后，设置将失效。

说明

- 使用方式一和方式二设置参数时，若所设参数不属于当前节点，数据库会提示参数不在支持范围内的相关信息。
- 使用方式一和方式二设置参数时，不支持以页面数量为参数单位进行设置。
- 使用方式三设置参数时，若参数值为int整型，则会将整数前导零过滤掉，例如SET *paraname* TO *008192*与SET *paraname* TO *8192*效果相同。

CM 参数设置

GaussDB有两类CM参数，一类在设置后需要重启集群才能生效，一类可以通过reload动态加载生效，具体设置方式请参考**表6-3**。

版权所有 © 华为云计算技术有限公司

表 6-3 CM 参数设置方式

序号	设置方法

方式一	1. 使用如下命令修改参数。
	`cm_ctl set --param {--agent | --server} [-n NODEID] -k "PARAMETER=value"`
	说明
	参数说明
	● --agent：设置cm_agent参数。
	● --server：设置cm_server参数。
	● -n NODEID：设置指定节点的CM参数，NODEID为节点名称，可通过cm_ctl query -Cv命令查询。不设置该参数时，默认设置所有节点。
	● -k "PARAMETER=value"：指定设置的参数和参数的值，只能设置已经存在的参数，不支持增加或删减参数。
	示例
	● 设置cm_agent参数incremental_build值为on：
	`cm_ctl set --param --agent -k "incremental_build=on"`
	● 设置cm_server参数switch_rto值为600：
	`cm_ctl set --param --server -k "switch_rto=600"`
	2. 重启集群使参数生效。
	`cm_ctl stop && cm_ctl start`
	说明
	重启集群操作会导致用户执行操作中断，请在操作之前规划好合适的执行窗口。

版权所有 © 华为云计算技术有限公司

序号	设置方法
方式二	1. 使用如下命令修改参数。 cm_ctl set --param {--agent | --server} [-n *NODEID*] -k "*PARAMETER*=*value*" **说明** 参数说明 • --agent：设置cm_agent参数。 • --server：设置cm_server参数。 • -n NODEID：设置指定节点的CM参数，NODEID为节点名称，可通过cm_ctl query -Cv命令查询。不设置该参数时，默认设置所有节点。 • -k "PARAMETER=value"：指定设置的参数和参数的值，只能设置已经存在的参数，不支持增加或删减参数。 示例 • 设置cm_agent参数incremental_build值为on： cm_ctl set --param --agent -k "incremental_build=on" • 设置cm_server参数switch_rto值为600： cm_ctl set --param --server -k "switch_rto=600" 2. 动态加载CM参数。 cm_ctl reload --param {--agent | --server} [-n *NODEID*] **说明** 参数说明 • --agent：动态加载cm_agent参数。 • --server：动态加载cm_server参数。 • -n NODEID：动态加载指定节点的CM参数，NODEID为节点名称，可通过cm_ctl query -Cv命令查询。 示例 • 动态加载cm_agent参数： cm_ctl reload --param --agent • 动态加载cm_server参数： cm_ctl reload --param --server

6.3 GUC 参数说明

6.3.1 GUC 使用说明

数据库提供了许多GUC参数，配置这些参数可以影响数据库系统的行为。在修改这些参数时请确保用户理解了这些参数本身的规格限制与修改后对数据库的影响，否则可能会导致无法预料的结果：

- 数据库内部部分参数与环境有关，在数据库安装时便已指定，不建议进行修改。如果有一定要修改的场景，设置时必须确保内容、格式正确，否则会导致实例异常。例如：设置侦听地址的参数listen_addresses；设置端口的参数port；设置ip地址的参数local_bind_address；以及各类设置文件输出路径的参数。
- 受限于使用的硬件资源大小，需要谨慎修改的参数。例如：内存、连接数、影响磁盘空间占用的相关参数。用户修改前需确认使用的硬件规格，确保参数修改后，不会超出硬件本身的大小限制，否则可能会使数据库实例申请不到足够的资源，导致实例状态异常。

版权所有 © 华为云计算技术有限公司

- 受限于参数之间的联动，修改参数时需要详细阅读参数对应规格限制。例如：max_wal_senders参数取值必须小于max_connections参数取值，数据库才能正常启动。
- 参数设置不能过于极端，设置过大过小都有使业务受损的可能，需要详细理解参数含义并谨慎修改。例如：session_timeout参数设置1秒，业务与服务器建立连接后，1秒不进行操作，便会断开连接，从而影响业务正常执行。log_min_messages控制打印日志的级别，设置级别过低，会导致记录日志量过大，影响性能。

注意事项

- 参数中如果取值范围为字符串，此字符串应遵循操作系统的路径和文件名命名规则。
- 取值范围最大值为INT_MAX的参数，此选项最大值跟所在的操作系统有关。其值为数据类型INT的最大值，值为2147483647。
- 取值范围最大值为DBL_MAX的参数，此选项最大值跟所在的操作系统有关。其值为数据类型FLOAT的最大值。
- 部分GUC参数会影响函数/操作符的选择、存储过程的编译以及执行计划的生成，从而会影响到视图、函数参数默认值、存储过程编译产物以及计划缓存的行为。由于这些机制原因，后续GUC参数变更可能不会再影响这些行为。
  - 视图：数据库在视图定义时会根据当前GUC参数状态生成重写规则，后续对视图操作时直接使用对应重写规则而不再受GUC参数（会影响重写规则生成的参数）的影响。
  - 函数参数默认值：如果函数参数默认值使用了函数，数据库在创建该函数时会根据当前GUC参数状态为默认值参数选择对应的函数并记录选中的函数oid到对应系统表中（pg_proc），后续执行该函数时会使用系统表中记录的函数oid给默认值参数赋值而不再受GUC参数的影响。
  - 存储过程编译产物：数据库对存储过程进行编译时会根据当前GUC参数状态生成编译产物，执行存储过程时直接使用编译产物而不再受GUC参数（会影响编译产物生成的参数）的影响。
  - 计划缓存：数据库在执行SQL时会根据当前GUC参数状态生成对应的执行计划，如果将该计划缓存起来后，后续再执行该SQL将直接使用该计划而不再受GUC参数（会影响执行计划生成的参数）的影响。

  有影响GUC参数如下表（GUC参数具体功能详见对应GUC参数）：

 版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

convert_string_to_digit	优先将字符串转为数字	不支持打开参数时优先将字符串转为数字的行为。	视图、函数参数默认值、存储过程编译产物、计划缓存	CREATE or REPLACE FUNCTION test(c numeric) RETURN text package AS
				BEGIN
				RETURN 'test(c numeric)';
				END;
				/
				CREATE or REPLACE FUNCTION test(c varchar2) RETURN text package AS
				BEGIN
				RETURN 'test(c varchar2)';
				END;
				/
				
				SET convert_string_to_digit=on;
				CREATE or REPLACE VIEW test_view AS SELECT test('123'::text);
				
				SELECT test('123'::text);
				test
				----------------- 	
				test(c numeric)
				(1 row)
				
				SELECT * FROM test_view;
				test
				----------------- 	
				test(c numeric)
				(1 row)
				
				--关闭参数
				SET convert_string_to_digit=off;
				
				SELECT test('123'::text);
				test
				------------------ 	
				test(c varchar2)
				(1 row)
				
				--视图行为和直接调用函数行为不一致
				SELECT * FROM test_view;
				test
				----------------- 	
				test(c numeric)
				(1 row))

set behavior_compat_options = 'current_sysdate';	SYSDATE函数获取当前操作系统时间（底层使用current_sysdate函数）	SYSDATE函数获取当前数据库时间（底层使用sysdate函数）。	视图、函数参数默认值、存储过程编译产物、计划缓存	SET behavior_compat_options = 'current_sysdate'; CREATE or REPLACE VIEW test_view_SYSDATE AS SELECT SYSDATE AS test; SELECT * FROM test_view_SYSDATE;--返回当前操作系统时间 SELECT SYSDATE;--返回当前操作系统时间 --关闭参数 SET behavior_compat_options = ''; --视图行为和直接调用函数行为不一致 SELECT * FROM test_view_SYSDATE;--返回当前操作系统时间 SELECT SYSDATE;--返回当前数据库时间

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
set a_format_version='10c'; set a_format_dev_version='s1';	• 新增支持NVL2。 • 新增部分系统函数。 • case when返回类型有变更。 • 集合类型构造器优先级高于函数。 • 支持timestamp到timestamptz的隐式转换。	不支持打开参数时功能。	视图、函数参数默认值、存储过程编译产物、计划缓存	SET a_format_version='10c'; SET a_format_dev_version='s1'; CREATE OR REPLACE VIEW test_view_nvl2 AS SELECT nvl2(1,2,3) AS test; SELECT * FROM test_view_nvl2; test ------ 2 (1 row) RESET a_format_dev_version; SELECT * FROM test_view_nvl2; test ------ 2 (1 row)

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
set a_format_version='10c'; set a_format_dev_version='s2';	• 新增支持CURRENT_TIMESTAMP '('FCONST')'。 • 新增支持DBTIMEZONE。 • 新增支持LNNVL。 • CURRENT_DATE a_format_date_timestamp=off且mapping_date_to_datea=off情况下，返回值列名为timesta	不支持打开参数时功能。	视图、函数参数默认值、存储过程编译产物、计划缓存	SET a_format_version='10c'; SET a_format_dev_version='s2'; CREATE or REPLACE VIEW test_view_LNNVL AS SELECT LNNVL(123=123) AS test; SELECT * FROM test_view_LNNVL; RESET a_format_dev_version; SELECT * FROM test_view_LNNVL;

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
	mp;a_format_date_timestamp=on且mapping_date_to_datea=off情况下，返回值列名为current_date;mapping_date_to_datea=on情况下，a_format_date_timestamp不影响列名，返回值列名为curr			

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
	ent_date。 • 新增部分系统函数。			
set a_format_version='10c'; set a_format_dev_version='s4';	新增部分系统函数。	不支持打开参数时功能。	视图、函数参数默认值、存储过程编译产物、计划缓存	-

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

set a_format_version='10c';
set a_format_dev_version='s5';
	• 新增部分系统函数。
	• 复合类型构造器优先级高于函数。
		不支持打开参数时功能。	视图、函数参数默认值、存储过程编译产物、计划缓存	--复合类型构造器优先级高于函数:
				CREATE TYPE tt AS（
				val1 int,
				val2 int
				);
				CREATE OR REPLACE FUNCTION tt(va int, vb int)
				RETURN int IS
				ret int;
				BEGIN
				ret := va;
				RETURN ret;
				END;
				/
				
				SET a_format_version='10c';
				SET a_format_dev_version='s5';
				
				--赋值给tt变量
				CREATE OR REPLACE FUNCTION test0
				RETURN int IS
				va tt;
				ret int;
				BEGIN
				va := tt(1,2);
				RAISE INFO 'tt: %' ,va;
				ret := 9;
				RETURN ret;
				END;
				/
				
				SELECT test0();
				INFO: tt: (1,2)
				test0
				------- 	
				9
				(1 row)
				
				SET a_format_version='';
				SET a_format_dev_version='';
				
				SELECT test0();
				INFO: tt: (1,2)
				test0
				------- 	
				9
				(1 row)
				
				--赋值给tt变量
				CREATE OR REPLACE FUNCTION test0
				RETURN int IS
				va tt;
				ret int;
				BEGIN
				va := tt(1,2);
				RAISE INFO 'tt: %' ,va;
				ret := 9;
				RETURN ret;
				END;
				/
				
				select test0();
				ERROR: cannot assign non-composite value to a row variable

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
set behavior_compat_options = 'enable_bpcharlikebpchar_compare';	启用bpcharlikebpchar和bpcharnlikebpchar操作符。	关闭bpcharlikebpchar和bpcharnlikebpchar操作符。	视图、存储过程编译产物、计划缓存	CREATE TABLE op_test (col BPCHAR(2) DEFAULT NULL); CREATE INDEX op_index ON op_test(col); INSERT INTO op_test VALUES ('a'); INSERT INTO op_test VALUES ('1'); INSERT INTO op_test VALUES ('11'); INSERT INTO op_test VALUES ('12'); INSERT INTO op_test VALUES ('sd'); INSERT INTO op_test VALUES ('aa'); SET behavior_compat_options = 'enable_bpcharlikebpchar_compare'; EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col; QUERY PLAN ------------------------------ Sort Sort Key: col -> Seq Scan on op_test Filter: (col ~~ col) (4 rows) CREATE OR REPLACE VIEW test_view_bpchar AS SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col; EXPLAIN SELECT * FROM test_view_bpchar; QUERY PLAN -- Sort (cost=34.48..34.50 rows=10 width=12) Sort Key: op_test.col -> Seq Scan on op_test (cost=0.00..34.31 rows=10 width=12) Filter: (col OPERATOR(pg_catalog.~~) col) (4 rows) SET behavior_compat_options = ''; EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col; QUERY PLAN --------------------------------------- Sort Sort Key: col -> Seq Scan on op_test Filter: (col ~~ (col)::text) (4 rows) EXPLAIN SELECT * FROM test_view_bpchar; QUERY PLAN -- Sort (cost=34.48..34.50 rows=10

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
				width=12) Sort Key: op_test.col -> Seq Scan on op_test (cost=0.00..34.31 rows=10 width=12) Filter: (col OPERATOR(pg_catalog.~~) col) (4 rows) --关闭开关的情况下重新创建视图： CREATE OR REPLACE VIEW test_view_bpchar AS SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col; explain select * from test_view_bpchar; QUERY PLAN ---------- Sort (cost=39.34..39.37 rows=10 width=12) Sort Key: op_test.col -> Seq Scan on op_test (cost=0.00..39.17 rows=10 width=12) Filter: (col ~~ (col)::text) (4 rows)

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
set b_format_version='5.7';set b_format_dev_version='s1';	• CURDATE、CURRENT_TIME、CURRENT_TIMESTAMP、LOCALTIME、LOCALTIMESTAMP NOW返回语句执行时刻时间戳。 • SYSDATE返回函数执行时刻时间戳。 • DB_JSOBJ、LAST_DAY_FUNC、	• CURDATE、CURRENT_TIME、CURRENT_TIMESTAMP、LOCALTIME、LOCALTIMESTAMP NOW、SYSDATE函数返回事务开始时刻时间戳。 • DB_JSOBJ、LAST_DAY_FUNC、EXTRACT、TIMEST	视图、函数参数默认值、存储过程编译产物、计划缓存	-

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
	EXTRACT、TIMESTAMPDIFF、SUBSTRING函数行为变更为与MYSQL一致。	AMPDIFF、SUBSTRING函数恢复默认行为。		

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
set behavior_compat_options = 'bind_procedure_searchpath';	• 未指定模式名的存储过程中的数据库对象的搜索路径配置项。 • 在存储过程中如果不显式指定模式名，会优先在存储过程所属的模式下搜索。	恢复默认行为。	存储过程编译产物	-

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

set behavior_compat_options='skip_insert_gs_source';	恢复默认行为。	创建PL/SQL对象（存储过程、函数以及package）时，不允许向gs_source中插入PL/SQL对象创建信息。	PL/SQL对象（存储过程、函数以及package）产物	CREATE OR REPLACE PROCEDURE proc_no_parameter
				AS
				ret1 int;
				BEGIN
				ret1 := 0;
				dbe_output.print_line('procedure has no parameter is 1w111.');
				END;
				/
				SELECT * FROM DBE_PLDEVELOPER.gs_source;
				id | owner | nspid | name | type | status | src
				-------+-------+-------+-------------------+-----------+--------
				+--
				24593 | 10 | 2200 | proc_no_parameter | procedure | t
				| CREATE OR REPLACE PROCEDURE proc_no_parameter +
				| | | | | |
				AS
				  +
				| | | | | | ret1
				int;
				  +
				| | | | | |
				BEGIN
				  +
				| | | | | | ret1 :=
				0;
				+
				| | | | | | dbe_output.print_line('procedure has no parameter is 1w111.');+
				| | | | | | END;
				(1 row)

set behavior_compat_options = 'enable_out_param_override';	控制存储过程出参的重载行为。	恢复默认行为。	存储过程编译产物	-

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

enable_bitmapscan	scan算子选择bitmapscan。	scan算子不选择bitmapscan。	计划缓存	SET enable_auto_explain=true;
				SET auto_explain_level=notice;
				CREATE TABLE t1(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				CREATE TABLE t2(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				INSERT INTO t1 SELECT i,10*random()*i,i%20,'atr'||i from generate_series(1,1000) i;
				CREATE INDEX index1 ON t1(a);
				
				SET enable_bitmapscan=on;
				SET enable_seqscan=off;
				SET enable_indexscan=off;
				CREATE OR REPLACE PROCEDURE proc_while_loop()
				AS
				DECLARE
				i int :=1;
				BEGIN
				WHILE i < 5 LOOP
				INSERT INTO t2 SELECT * FROM t1 WHERE a = i ;
				i:=i+1;
				END LOOP;
				END;
				/
				CALL proc_while_loop();
				SET enable_bitmapscan=off;
				CALL proc_while_loop();
				SET enable_bitmapscan=on;
				CALL proc_while_loop();

enable_indexscan	scan算子选择indexscan。	scan算子不选择indexscan。	计划缓存	SET enable_auto_explain=true;
				SET auto_explain_level=notice;
				CREATE TABLE t1(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				CREATE TABLE t2(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				INSERT INTO t1 SELECT i,10*random()*i,i%20,'atr'||i from generate_series(1,1000) i;
				CREATE INDEX index1 on t1(a);
				SET enable_bitmapscan=off;
				SET enable_seqscan=off;
				CREATE OR REPLACE PROCEDURE proc_while_loop()
				AS
				DECLARE
				i int :=1;
				BEGIN
				WHILE i < 5 LOOP
				insert into t2 select * from t1 where a = i ;
				i:=i+1;
				END LOOP;
				END;
				/
				CALL proc_while_loop();
				SET enable_indexscan=off;
				CALL proc_while_loop();

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

query_Dop	用户自定义的查询并行度。开启SMP功能，系统会使用设定的并行度执行。	默认值为1，1表示关闭并行查询。	计划缓存	SET enable_auto_explain=true;
				SET auto_explain_level=notice;
				CREATE TABLE t1(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				CREATE TABLE t2(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				INSERT INTO t1 SELECT i,10*random()*i,i%20,'atr'||i from generate_series(1,1000) i;
				CREATE INDEX index1 ON t1(a);
				SET enable_bitmapscan=f;
				CREATE OR REPLACE PROCEDURE proc_while_loop()
				AS
				DECLARE
				i int :=1;
				BEGIN
				WHILE i < 5 LOOP
				INSERT INTO t2 SELECT /*+ indexscan(t1 index1) */* FROM t1 WHERE a = i ;
				i:=i+1;
				END LOOP;
				END;
				/
				SET enable_auto_explain=true;
				SET auto_explain_level=notice;
				SET client_min_messages=log;
				SET enable_stream_operator=on;
				SET query_dop=4;
				CALL proc_while_loop();
				SET query_Dop=6;
				CALL proc_while_loop();

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

set plan_cache_mode = force_generic_plan	pbe情况下，强制执行gplan。	pbe情况下，不强制执行gplan。	计划缓存	SET enable_auto_explain=true;
				SET auto_explain_level=notice;
				CREATE TABLE t1(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				CREATE TABLE t2(a int, b int, c int,d varchar)WITH(storage_type=ustore);
				INSERT INTO t1 SELECT i,10*random()*i,i%20,'atr'||i from generate_series(1,1000) i;
				CREATE INDEX index1 ON t1(a);
				SET enable_bitmapscan=on;
				SET enable_seqscan=off;
				SET enable_indexscan=off;
				CREATE OR REPLACE PROCEDURE proc_while_loop()
				AS
				DECLARE
				i int :=1;
				BEGIN
				WHILE i < 5 LOOP
				insert into t2 select * from t1 where a = i ;
				i:=i+1;
				END LOOP;
				END;
				/
				CALL proc_while_loop();
				SET plan_cache_mode = force_generic_plan;
				PREPARE aa AS SELECT proc_while_loop();
				EXECUTE aa;
				SET enable_bitmapscan=off;
				EXECUTE aa;

a_format_date_timestamp	CURRENT_DATE返回当前SQL启动的时间戳。	CURRENT_DATE返回事务开启的日期或日期及时间。	视图、函数参数默认值、存储过程编译产物、计划缓存	SET a_format_date_timestamp=on;
				CREATE OR REPLACE PROCEDURE proc_while_loop()
				AS
				DECLARE
				i date;
				BEGIN
				i:=current_date;
				raise info 'step2:%',TO_CHAR(current_date, 'MM-DD-YYYY HH24:MI:SS');
				END;
				/
				CALL proc_while_loop();
				PREPARE p1 AS SELECT current_timestamp;
				EXECUTE p1;
				SET a_format_date_timestamp=off;
				CALL proc_while_loop();
				EXECUTE p1;

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
adjust_systemview_priority	同名视图优先访问用户自定义视图。	同名视图优先访问系统视图。	视图参数默认值	CREATE DATABASE database_test DBCOMPATIBILITY = 'ORA'; SELECT datname,datcompatibility,dattimezone FROM pg_database; \c database_test CREATE TABLE t1(c int); CREATE TABLE CREATE VIEW dual AS SELECT * FROM t1; CREATE VIEW SET adjust_systemview_priority = off; SET SELECT * FROM dual; dummy ------- X (1 row) PREPARE p1 as SELECT * FROM dual; PREPARE EXECUTE p1; dummy ------- X (1 row) SET adjust_systemview_priority = on; SET SELECT * FROM dual; c --- (0 rows) EXECUTE p1; dummy ------- X (1 row)

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
mapping_date_to_datea	在ORA兼容模式下，用datea类型替换date类型。	恢复默认行为。	视图、函数参数默认值、存储过程编译产物、计划缓存。	--新安装数据库中默认开关处于打开状态 SHOW mapping_date_to_datea; mapping_date_to_datea ---------- on (1 row) CREATE TABLE t1(c1 date); CREATE TABLE \d+ t1; Table "public.t1" Column | Type | Modifiers | Storage | Stats target | Description --------+-------+-----------+---------+--------------+------------- c1 | datea | | plain | | Has OIDs: no Options: orientation=row, compression=no, storage_type=USTORE, segment=off --在gaussdb.conf中设置开关为off，重启数据库后开关变为关闭 SHOW mapping_date_to_datea; mapping_date_to_datea ---------- off (1 row) CREATE TABLE t2(c1 date); CREATE TABLE \d+ t2; Table "public.t2" Column | Type | Modifiers | Storage | Stats target | Description --------+-----------------------------+-----------+---------+--------------+------------- c1 | timestamp(0) without time zone | | plain | | Has OIDs: no Options: orientation=row, compression=no, storage_type=USTORE, segment=off

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例
nls_date_format	在ORA兼容模式下，用于datea类型和字符类型转换时的格式控制： 1. 字符类型转换为date类型时，必须满足nls_date_format中设置的时间格式； 2. date类型转换为字符类型时，会按照nls_date_format的格式转换； 3. to_char和to_date默认按照nls_date_format格式转换。	不进行设置时，nls_date_format的默认值为"DD-MON-RR"。	视图、函数参数默认值、计划缓存	-- 默认值为'DD-MON-RR' SHOW nls_date_format; nls_date_format ----------------- DD-MON-RR (1 row) -- 输入date类型不符合nls_date_format格式时校验报错 SELECT '2020-10-10'::date; ERROR: invalid value "20-10-10" for "MON" LINE 1: select '2020-10-10'::date; ^ DETAIL: The given value did not match any of the allowed values for this field. CONTEXT: referenced column: datea SET nls_date_format='YYYY-MM-DD'; SET SELECT '2020-10-10'::date; datea ------------------------- 2020-10-10 00:00:00 AD (1 row) -- to_date函数的输入必须符合nls_date_format格式，否则报错 SELECT to_date('10-10-2020'); ERROR: Invalid data exists in a string for matching date. CONTEXT: referenced column: to_date SELECT to_date('2020-10-10'); to_date ------------------------- 2020-10-10 00:00:00 AD (1 row) -- to_char函数将date类型转为字符串时，字符串结果为nls_date_format设置的格式 SELECT to_char(to_date('05-2020-10', 'MM-YYYY-DD')); to_char ------------ 2020-05-10 (1 row)

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

fixed_date	在ORA兼容模式下，固定sysdate的值，sysdate返回fixed_date所表示的时间。	sysdate返回系统时间。	sysdate的值。	-- 开启fixed_date时返回的结果。
				gaussdb=# SET fixed_date='2024-05-15 11:47:02';
				SET
				gaussdb=# SELECT sysdate;
				timestamp
				--------------------- 	
				2024-05-15 11:47:02
				(1 row)
				
				gaussdb=# PREPARE p1 AS SELECT sysdate;
				PREPARE
				gaussdb=# EXECUTE p1;
				datea
				------------------------ 	
				2024-05-15 11:47:02 AD
				(1 row)
				
				gaussdb=# SET fixed_date='2024-05-16 11:47:02';
				SET
				gaussdb=# SELECT sysdate;
				datea
				------------------------ 	
				2024-05-16 11:47:02 AD
				(1 row)
				
				gaussdb=# EXECUTE p1;
				datea
				------------------------ 	
				2024-05-15 11:47:02 AD
				(1 row)

a_format_func_col_name	在ORA兼容模式下，将select user返回的投影列名从current_user改为user。	在ORA兼容模式下，select user返回的投影列名为current_user。	计划缓存。	gaussdb=# PREPARE p1 AS SELECT user FROM dual;
				PREPARE
				gaussdb=# EXECUTE p1;
				current_user
				-------------- 	
				omm
				(1 row)
				gaussdb=# SET a_format_func_col_name = 'user';
				SET
				gaussdb=# EXECUTE p1;
				current_user
				-------------- 	
				omm
				(1 row)

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

disable_keyword_options	影响各兼容模式关键字的行为，具体内容可参考表6-38	影响各兼容模式关键字的行为，具体内容可参考表6-38	计划缓存	gaussdb=# set disable_keyword_options = 'dbtimezone';
				SET
				gaussdb=# prepare p1 as select 1 dbtimezone;
				PREPARE
				gaussdb=# execute p1;
				dbtimezone
				------------ 	
				1
				(1 row)
				
				gaussdb=# set disable_keyword_options = '';
				SET
				gaussdb=# execute p1;
				dbtimezone
				------------ 	
				1
				(1 row)
				
				gaussdb=# deallocate p1;
				DEALLOCATE

backslash_quote	打开参数允许字符串文本中的单引号使用\'表示	关闭参数不允许字符串文本中的单引号使用\'表示	计划缓存	gaussdb=# set backslash_quote = 'safe_encoding';
				SET
				gaussdb=# prepare p1 as select text E'Don \'t do that.';
				PREPARE
				gaussdb=# execute p1;
				text
				------------------ 	
				Don 't do that.
				(1 row)
				
				gaussdb=# set backslash_quote = off;
				SET
				gaussdb=# execute p1;
				text
				------------------ 	
				Don 't do that.
				(1 row)
				
				gaussdb=# deallocate p1;
				DEALLOCATE

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

set sql_mode='ansi_quotes'	M-Compatibility模式下标识符同时支持反引号（``）和双引号（""）	M-Compatibility模式下标识符仅支持反引号（``）	计划缓存	-- sql_mode选项变化，不会让计划重新生成。
				m_db=# CREATE TABLE t1(a int);
				NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'a' as the distribution column by default.
				HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
				CREATE TABLE
				m_db=# SET sql_mode = 'ansi_quotes';
				SET
				m_db=# PREPARE p1 FROM 'INSERT INTO "t1" VALUES(1)';
				PREPARE
				m_db=# EXECUTE p1;
				INSERT 0 1
				
				m_db=# SET sql_mode='';
				SET
				m_db=# EXECUTE p1;
				INSERT 0 1
				m_db=# DROP TABLE t1;
				DROP TABLE

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

standard_conforming_strings	按照SQL标准把反斜杠当做普通文本	反斜杠在字符串字面值中被解释为转义字符	计划缓存	gaussdb=# SET standard_conforming_strings = on;
				SET
				gaussdb=# PREPARE p1 AS SELECT u&'d\0061t\+000061' AS u&"d\0061t\+000061";
				PREPARE
				gaussdb=# SELECT u&'d\0061t\+000061' AS u&"d\0061t\+000061";
				data
				------ 	
				data
				(1 row)
				
				gaussdb=# EXECUTE p1;
				data
				------ 	
				data
				(1 row)
				
				gaussdb=# SET standard_conforming_strings = off;
				SET
				gaussdb=# SELECT u&'d\0061t\+000061' AS u&"d\0061t\+000061";
				ERROR: unsafe use of string constant with Unicode escapes
				LINE 1: select u&'d\0061t\+000061' as u&"d\0061t\+000061";
				^
				DETAIL: String constants with Unicode escapes cannot be used when standard_conforming_strings is off.
				gaussdb=# EXECUTE p1;
				data
				------ 	
				data
				(1 row)
				
				gaussdb=# DEALLOCATE p1;
				DEALLOCATE

版权所有 © 华为云计算技术有限公司

GUC参数	打开参数	关闭参数	影响点	示例

set behavior_compat_options = 'ignore_like_default_escape';	standard_conforming_strings为on时可以开启该选项，开启后支持在LIKE模糊匹配中将模式串的反斜杠字符视为普通字符进行匹配。	LIKE模糊匹配中模式串的反斜杠字符作为转义字符，用于消除通配符的特殊含义。	计划缓存	gaussdb=# SET standard_conforming_strings = on;
				SET
				gaussdb=# SET behavior_compat_options = 'ignore_like_default_escape';
				SET
				gaussdb=# CREATE TABLE test(c1 text);
				CREATE TABLE
				gaussdb=# INSERT INTO test values('aa\a');
				INSERT 0 1
				gaussdb=# PREPARE p1 AS SELECT * FROM test WHERE c1 LIKE 'aa\a';
				PREPARE
				gaussdb=# EXECUTE p1;
				  c1
				------ 	
				 aa\a
				(1 row)
				
				gaussdb=# SELECT * FROM test WHERE c1 LIKE 'aa\a';
				  c1
				------ 	
				 aa\a
				(1 row)
				
				gaussdb=# SET behavior_compat_options = '';
				SET
				gaussdb=# EXECUTE p1;
				  c1
				------ 	
				 aa\a
				(1 row)
				
				gaussdb=# SELECT * FROM test WHERE c1 LIKE 'aa\a';
				 c1
				---- 	
				(0 rows)
				
				gaussdb=# DEALLOCATE p1;
				DEALLOCATE

6.3.2 文件位置

数据库安装后会自动生成三个配置文件（gaussdb.conf、gs_hba.conf和gs_ident.conf），并统一存放在数据目录（data）下。用户可以使用本节介绍的方法修改配置文件的名称和存放路径。

修改任意一个配置文件的存放目录时，gaussdb.conf里的data_directory参数必须设置为实际数据目录（data）。

说明

考虑到配置文件修改一旦出错对数据库的影响很大，不建议安装后再修改本节的配置文件。

版权所有 © 华为云计算技术有限公司

data_directory

参数说明：设置GaussDB的数据目录（data目录），仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：安装时指定，如果在安装时不指定，则默认不初始化数据库。

设置方式：此参数可以通过如下方式设置：

- 在安装GaussDB时设置。
- 该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

config_file

参数说明：设置主服务器配置文件名称（gaussdb.conf）。

参数类型：字符串

参数单位：无

取值范围：长度大于0的字符串。

默认值：gaussdb.conf（实际安装可能带有绝对目录）。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

hba_file

参数说明：设置基于主机认证（HBA）的配置文件（gs_hba.conf）。此参数只能在配置文件gaussdb.conf中指定，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：长度大于0的字符串。

默认值：gs_hba.conf（实际安装可能带有绝对目录）。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议修改配置文件参数。

设置不当的风险与影响：修改后可能会影响集群升级、备份功能。

版权所有 © 华为云计算技术有限公司

ident_file

参数说明：设置用于客户端认证的配置文件的名称（gs_ident.conf），仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：长度大于0的字符串。

默认值：gs_ident.conf（实际安装可能带有绝对目录）。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议修改配置文件参数。

设置不当的风险与影响：修改后可能会影响集群升级、备份功能。

external_pid_file

参数说明：声明可被服务器管理程序使用的额外PID文件，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法文件名称。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

说明

该参数只能在数据库服务重新启动后生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.3 连接和认证

6.3.3.1 连接设置

介绍设置客户端和服务器连接方式相关的参数。

light_comm

参数说明：设置服务器是否使用轻量通信方式。

该参数指定服务器是否使用基于轻量锁和非阻塞socket的通信方式。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示使用轻量通信方式。
- off：表示不使用轻量通信方式。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

listen_addresses

参数说明：声明服务器侦听客户端的TCP/IP地址。

该参数指定GaussDB服务器使用哪些IP地址进行侦听，如IPv4或IPv6。服务器主机上可能存在多个网卡，每个网卡可以绑定多个IP地址，该参数用来控制GaussDB绑定在哪一个或者哪几个IP地址上。而客户端则可以通过该参数中指定的IP地址来连接GaussDB或者给GaussDB发送请求。

参数类型：字符串

参数单位：无

取值范围：

- 主机名或IP地址，多个值之间用英文逗号分隔。
- “*”或“0.0.0.0”或“::”表示侦听所有IP地址。配置侦听所有IP地址存在安全风险，不推荐用户使用。
- 置空则服务器不会侦听任何IP地址，这种情况下，只有Unix域套接字可以用于连接数据库。

默认值：

集群安装好后，根据public_cloud.conf配置文件中不同实例的IP地址配置不同默认值。CN的默认参数值为：“localhost,mgr.net网卡对应的IP地址，data.net网卡对应的IP地址,virtual.net网卡对应的IP地址”；DN的默认参数值为：“data.net网卡对应的IP”。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

说明

localhost表示只允许进行本地“回环”连接。

public_cloud.conf文件保存的网卡信息，包括：mgr.net（管理网卡）、data.net（数据网卡）、virtual.net（虚拟网卡）。

使用IPv6地址时，若IP为fe80地址块，则配置时需要追加'%zone index'信息。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

listen_address_ext

参数说明：声明服务器侦听客户端的扩展TCP/IP地址。该参数指定GaussDB服务器使用哪些IP地址作为扩展侦听IP。

参数类型：字符串

参数单位：无

取值范围：IPv4或者IPv6格式地址，不支持多IP设置。

默认值：“localhost”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在BMS部署形态下，集群安装启动时自动配置。不建议用户手动配置使用。

说明

- listen_address_ext参数配置涉及整个集群扩展IP的网络拓扑关系和配置，所以set/reload都必须通过OM工具实现。
- listen_address_ext是用于扩展具体IP地址的网络通道，若设置为“*”，“0.0.0.0”，“localhost”，“127.0.0.1”，“::-1”则不生效，建议此类地址设置在listen_addresses参数中。
- listen_address_ext用于配置分布式DN实例，CN配置后不生效。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

local_bind_address

参数说明：声明当前节点连接集群其他节点绑定的本地IP地址。

参数类型：字符串

参数单位：无

取值范围：IPv4或IPv6格式地址，不支持多IP设置。

默认值：

集群安装好后，根据public_cloud.conf配置文件中不同实例的IP地址配置不同默认值。CN/DN的默认参数值为：“data.net网卡对应的IP地址”。

说明

public_cloud.conf文件保存的网卡信息，包括：mgr.net（管理网卡）、data.net（数据网卡）、virtual.net（虚拟网卡）。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数由安装时的配置文件指定，请勿轻易修改，否则修改后会影响数据库正常通信。

port

参数说明：GaussDB服务侦听的TCP端口号。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

说明

- 设置端口号时，请设置一个未被占用的端口号。设置多个实例的端口号，不可冲突。
- 1 ~ 1023为操作系统保留端口号，请不要使用。
- 通过配置文件安装集群时，配置文件中的端口号需要注意通信矩阵预留端口。如：DN还需保留dataPortBase+1作为内部工具使用端口。故集群安装阶段，port最大值为：CN可设置65532，DN可设置65529，GTM可设置65534，同时需要保证端口号不冲突。

默认值：5432（实际值由安装时的配置文件指定）

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数由安装时的配置文件指定，请勿轻易修改，否则修改后会影响数据库正常通信。

max_connections

参数说明：允许和数据库连接的最大并发连接数。此参数会影响集群的并发能力。

参数类型：整型

参数单位：无

取值范围：最小值为10（要大于max_wal_senders），理论最大值为262143，实际最大值为动态值，计算公式为“262143 - job_queue_processes - autovacuum_max_workers - max_inner_tool_connections - max_concurrent_autonomous_transactions - AUXILIARY_BACKENDS - AV_LAUNCHER_PROCS - min(max(newValue/4,64),1024)”，job_queue_processes、autovacuum_max_workers、max_inner_tool_connections和max_concurrent_autonomous_transactions的值取决于对应GUC参数的设置，AUXILIARY_BACKENDS为预留辅助线程数固定为20，AV_LAUNCHER_PROCS为预留autovacuum的launcher线程数固定为2，min(max(newValue/4,64),1024)公式中newValue为新设置的值。

在不同实例的内存规格下，强制该参数取值范围如下：

表 6-4 独立部署模式下，不同实例的内存规格的参数取值范围

内存规格	CN参数取值范围	DN参数取值范围
< 32GB	[10, 100]	[10, 100]
[32GB, 64GB)	[10, 200]	[10, 200]
[64GB, 128GB)	[10, 1000]	[10, 2500]
[128GB, 256GB)	[10, 2000]	[10, 6000]
[256GB, 480GB)	[10, 4000]	[10, 12000]

版权所有 © 华为云计算技术有限公司

内存规格	CN参数取值范围	DN参数取值范围
>= 480GB	[10, 8000]	[10, 24000]

默认值：

- 独立部署：

  CN：8000（128核CPU/1024G内存；60核CPU/480G内存）；4000（32核CPU/256G内存）；2000（16核CPU/128G内存）；1000（8核CPU/64G内存）；100（4核CPU/32G内存，4核CPU/16G内存）

  DN：24000（128核CPU/1024G内存；60核CPU/480G内存）；12000（32核CPU/256G内存）；6000（16核CPU/128G内存）；2500（8核CPU/64G内存）；100（4核CPU/32G内存，4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：数据库主节点中此参数建议保持默认值。该参数调小会导致实际可用连接数变小，若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

设置不当的风险与影响：

- 若配置max_connections过大，超过计算公式所描述的最大动态值，会出现CN、DN拉起失败问题，报错提示“invalid value for parameter "max_connections"”；或在拉起时申请内存失败，报错提示“Cannot allocate memory”。
- 若未按照对外出口规格配置仅调大max_connections参数值，未同比例调整内存参数。业务压力大时，容易出现内存不足，报错提示“memory is temporarily unavailable”。
- 混合部署场景下，若CN的max_connections参数值大于（DN的max_connections/CN个数）时，客户端总压力超过DN端max_connections值时，可能会出现CN节点连接DN失败情况，报错提示“pooler... Too many clients already”。
- 若thread_pool_attr设置值大于等于max_connections，会导致proc资源全部被线程池占用，出现proc资源不足的问题。

 版权所有 © 华为云计算技术有限公司

说明

- 对于管理员用户的连接数限制会略超过max_connections设置，目的是为了让管理员在连接被普通用户占满后仍可以连接上数据库，再超过一定范围（sysadmin_reserved_connections参数控制）后才会报错。即管理员用户的最大连接数等于max_connections + sysadmin_reserved_connections + sysadmin_reserved_connections（通过主端口+1端口登录专属预留数量）。
- 对于普通用户来说，由于内部作业也会使用一些连接，因此会略小于max_connections，具体值取决于内部连接个数。
- 开启线程池后，stream线程数的上限为max_connections设置值。若stream线程数量达到上限时，将产生“Exceed stream thread pool limitation...”报错，此时可以通过调整max_connections参数值将stream线程数上限调高。由于该参数属于POSTMASTER类型，故设置时可以综合业务情况进行合理预估：stream线程数总数 = 业务并发数 * 每一个并发执行的语句所消耗的stream线程数量（可通过执行计划查看）。
- 节点连接数max_connections包含外部连接和内部连接。
  - 外部连接：用户应用程序的连接。
  - 内部连接：内部作业产生的连接。
- 在事务中使用DDL操作时，CN会在集群中的其他节点建立连接以进行同步操作，直至事务结束，此时CN的建连会占用对应节点的连接数。该场景的并发执行DDL数量为N，则某个节点的内部连接数量增加N个，因此该节点外部客户端可使用的连接数量也会相应地减少N个。例如使用gsql在CN1建立3个连接，并在事务内使用DDL操作，CN1会向其他节点分别建立3个内部连接，CN2节点新增3个来自CN1的内部连接，因此CN2节点外部客户端可使用的连接数量减少3个。

max_inner_tool_connections

参数说明：允许和数据库连接的工具的最大并发连接数。此参数会影响GaussDB的工具连接并发能力。

参数类型：整型

参数单位：无

取值范围：最小值为1，最大值的计算公式为“262143 - job_queue_processes - autovacuum_max_workers - max_connections - max_concurrent_autonomous_transactions - AUXILIARY_BACKENDS - AV_LAUNCHER_PROCS - min(max(max_connections/4,64),1024)”，**job_queue_processes**、**autovacuum_max_workers**、**max_connections**和**max_concurrent_autonomous_transactions**的值取决于对应GUC参数的设置，AUXILIARY_BACKENDS为预留辅助线程数固定为20，AV_LAUNCHER_PROCS为预留autovacuum的launcher线程数固定为2。

默认值：50。如果该默认值超过内核支持的最大值（在执行gs_initdb的时候判断），系统会提示错误。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：数据库主节点中此参数建议保持默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

sysadmin_reserved_connections

参数说明：为管理员用户预留的最少连接数，也是额外给管理员用户通过主端口+1端口号登录预留的最少连接数，不建议设置过大。该参数和max_connections参数配合使

版权所有 © 华为云计算技术有限公司

用，管理员用户的最大连接数等于max_connections + sysadmin_reserved_connections + sysadmin_reserved_connections（通过主端口+1端口登录专属预留数量）。

说明

- 当启用线程池功能时，若线程池占满将形成处理瓶颈，导致管理员预留连接无法正常建立；作为逃生手段，此时可使用gsql通过主端口+1端口号连入，清理无用会话，即可正常连入。
- 当管理员预留连接数全部被使用后，新建连接将失败，此时只能通过重启集群来恢复，所以需要谨慎使用该预留连接数。

参数类型：整型

参数单位：无

取值范围：最小值为0，最大值为MIN(262143, max_connections)，max_connections的计算方法见**max_connections**。

默认值：3

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

service_reserved_connections

参数说明：为后台运维用户（带有persistence属性）预留的最少连接数，不建议设置过大。该参数和max_connections参数配合使用，运维用户的最大连接数等于max_connections + service_reserved_connections。

参数类型：整型

参数单位：无

取值范围：0 ~ 262143

默认值：10

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果设置过小，会导致在max_connections占满的情况下，运维用户（带有persistence属性）无法连接数据库，作业无法正常执行。如果设置过大，则会有过多的空闲资源。

extip_reserved_connections

参数说明：为用户在listen_address_ext配置的扩展IP上进行的普通gsql业务连接预留的最少连接数。该参数和max_connections参数配合使用，对应业务连接的最大连接数等于max_connections + service_reserved_connections。

说明

该参数仅在为分布式DN配置了listen_address_ext的具体IP地址时有效。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：无

取值范围：最小值为0，最大值为MIN(262143, max_connections)，max_connections的计算方法见**max_connections**。

默认值：5

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

unix_socket_directory

参数说明：设置GaussDB服务器侦听客户端连接的Unix域套接字目录，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

说明

该参数的取值长度限制于操作系统的最大目录路径长度，Linux系统下，套接字路径名（套接字目录与套接字文件名拼接而成）长度不得超过107bytes，目录最长不得超过92bytes。超过该限制将会导致Unix-domain socket path "xxx" is too long的问题。

默认值：""（实际值由安装时配置文件中tmpMppdbPath指定）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若误设置出错（包括长度超过限制、非法目录等），会影响进程正常拉起，可以通过检索cm_agent路径下system_call日志定界。

unix_socket_group

参数说明：设置Unix域套接字的所属组（套接字的所属用户总是启动服务器的用户）。可以与选项**unix_socket_permissions**一起用于对套接字进行访问控制。

参数类型：字符串

参数单位：无

取值范围：系统中已有的用户组。

默认值：""，表示当前用户的缺省组。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

unix_socket_permissions

参数说明：设置Unix域套接字的访问权限。

Unix域套接字使用普通的Unix文件系统权限集。这个参数的值应该是数值的格式（chmod和umask命令可接受的格式）。如果使用自定义的八进制格式，数字必须以0开头。

参数类型：整型

参数单位：无

取值范围：0000 ~ 0777（8进制计数，转化为十进制为0 ~ 511）

说明

在Linux中，文档具有十个属性，其中第一个属性为文档类型，后面九个为权限属性，分别为Owner，Group及Others这三个组别的read、write、execute属性。

文档的权限属性分别简写为r，w，x，这九个属性三个为一组，也可以使用数字来表示文档的权限，对照表如下：

r：4

w：2

x：1

-：0

同一组（owner/group/others）的三个属性是累加的。

例如，-rwxrwx---表示这个文档的权限为：

owner = rwx = 4+2+1 = 7

group = rwx = 4+2+1 = 7

others = --- = 0+0+0 = 0

所以其权限为0770。

默认值：0700（8进制计数，转化为十进制为448）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为0770（只有当前连接数据库的用户和同组的人可以访问）或者0700（只有当前连接数据库的用户自己可以访问，同组或者其他人都没有权限）。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

application_name

参数说明：当前连接请求当中，所使用的客户端名称。

参数类型：字符串

参数单位：无

取值范围：合法的客户端名称，实际查询结果取决于查询所用的客户端或用户设置。

说明

在备机请求主机进行日志复制时，如果该参数非空串，那么会被用来作为备机在主机上的流复制槽名字。此时，如果该参数长度超过61个字节，那么流复制槽名字只会截取使用前61个字节的字符。

默认值：""（连接到后端的应用名，以实际安装为准）。

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。但是需要注意的是，客户端连接后该参数会被设为客户端名称，且设置为客户端级别，所以只有设置会话级别的参数会生效，其他方式都由于级别低于客户端级别而不会生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：数据库依赖于application_name标识部分内部进程，当用户设置了与内部进程同名的application_name时，可能产生非预期的行为。

connection_info

参数说明：连接数据库的驱动类型、驱动版本号、当前驱动的部署路径和进程属主用户。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""，表示当前连接数据库的驱动不支持自动设置connection_info参数或应用程序未设置。

说明

驱动连接数据库的时候自行拼接的connection_info参数格式如下：

```
{"driver_name":"ODBC","driver_version": "(GaussDB XXX.XXX.XXX build 290d125f) compiled at 2020-05-08 02:59:43 last mr 131 release","driver_path":"/usr/local/lib/gsqlodbcw.so","os_user":"omm"}
```

默认显示driver_name和driver_version，driver_path和os_user的显示由用户通过连接串中的连接参数connectionExtraInfo进行控制（详情请参见《开发指南》中“应用程序开发教程 > 基于JDBC开发 > 开发步骤 > 连接数据库 > 连接参数参考”章节和《开发指南》中“应用程序开发教程 > 附录 > ODBC > 配置参数说明”章节）。

设置方式：该参数属于USERSET类型参数，属于运维类参数，不建议用户设置。该参数仅支持在会话中设置，若要调整参数取值，请参见**表6-2**中“方式三”的设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

backend_version

参数说明：用于CN和CN，CN和DN建立连接时同步连接的版本号，该参数涉及版本号，用户不可以随意设置。

参数类型：整型

参数单位：无

取值范围：0 ~ 100000

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置出错，会导致不同节点之间版本号不一致。

版权所有 © 华为云计算技术有限公司

check_disconnect_query

参数说明：该参数控制的是，当客户端异常断连（如JDBC触发socketTimeout、libpq触发rwtimeout且关闭连接、运行业务过程中客户端进程终止等）后，GaussDB服务端语句是否终止执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示当客户端异常断连后，GaussDB服务端终止运行对应的语句。
- off：表示当客户端异常断连后，GaussDB服务端不会终止运行对应的语句。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

connection_reject_nodes

参数说明：用于表示CN/DN中拒绝接受其他CN连接的黑名单，用逗号进行分隔，例如'cn_5001,cn_5002'。当黑名单中的CN（如cn_5001）连接至本节点时，会拒绝接受其连接，并报错“Reject cn_5001 connection”。

参数类型：字符串

参数单位：无

取值范围：由若干个节点名称和英文逗号组成的字符串。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当设置为节点名时，其他节点会拒绝接受其连接，造成连接报错。

enable_server_listener_thread

参数说明：用于控制是否开启独立监听线程。当启用该功能时，GaussDB Kernel服务侦听的TCP端口**port**将由独立线程进行监听。独立监听线程功能开启需要设置GUC参数**enable_thread_pool**和**enable_connect_thread_pool**为on。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启该功能，GaussDB Kernel服务侦听的TCP端口**port**将由独立线程进行监听。

版权所有 © 华为云计算技术有限公司

- off：不开启该功能，GaussDB Kernel服务侦听的TCP端口**port**将由服务实例进程的Postmaster线程进行监听。

默认值：

- 新安装的数据库：on。
- 从507.0.0之前版本升级的数据库：升级后默认值为off。
- 从507.0.0及之后版本升级进行升级：升级后参数值保留原设置。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。开启独立监听线程会新增运行线程，在硬件资源受限时建议根据实际测试结果决定是否开启该功能。

设置不当的风险与影响：若未开启此功能，当Postmaster线程和其他线程竞争时，服务侦听端口的监听效率可能下降，影响客户端的连接建立耗时。请在经过测试验证后进行修改。

6.3.3.2 安全和认证

介绍设置客户端和服务器的安全认证方式的相关参数。

authentication_timeout

参数说明：完成客户端认证的最长时间。如果一个客户端没有在这段时间里完成与服务器端的认证，则服务器自动中断与客户端的连接，这样就避免了出问题的客户端无限制地占用连接数。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 600

默认值：60

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值600，表示authentication_timeout为600s；带单位取值10min，表示authentication_timeout为10min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：参数设置过小，可能会导致频繁遇到连接失败，影响用户使用。参数设置过大，会导致安全风险增加、资源浪费等问题。

auth_iteration_count

参数说明：认证加密信息生成过程中使用的迭代次数。

参数类型：整型

参数单位：无

取值范围：2048 ~ 134217728

默认值：10000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：请根据实际硬件条件合理设置迭代次数，推荐采用默认迭代次数。

设置不当的风险与影响：迭代次数设置过小会降低口令存储的安全性，设置过大会导致认证、用户创建等涉及口令加密的场景性能劣化。

session_authorization

参数说明：当前会话的用户标识。该参数的取值只能通过SHOW语句进行查询。

参数类型：字符串

参数单位：无

取值范围：合法的用户标识。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**SET SESSION AUTHORIZATION**语法进行设置。直接设置时，只支持设置为DEFAULT，不支持直接设置为其他值。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

session_timeout

参数说明：表明与服务器建立连接后，不进行任何操作的最长时间。当该参数取值不为0，用户不进行任何操作的时间超过该参数取值后，会与服务器断开连接。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 86400（1d）

- 0表示关闭超时设置。
- 正数表示与服务器建立连接后，不进行任何操作的最长时间。当超过设定的值后，会与服务器断开连接。

默认值：1800

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值600，表示session_timeout为600s；带单位取值10min，表示session_timeout为10min。取值如果要带单位，必须为s、min、h、d。

设置建议：参数值设置越小越安全，但用户使用越不便，依据业务安全诉求设定。

设置不当的风险与影响：设置过小可能造成业务连接池中长连接失效，进而导致业务失败。

说明

GaussDB gsql客户端中有自动重连机制，所以针对初始化用户本地连接，超时后gsql表现的现象为断开后重连。

idle_in_transaction_timeout

参数说明：表明与服务器建立连接并开启事务后，事务空闲并不进行任何操作的最长时间。当该参数取值不为0，事务空闲并不进行任何操作的时间超过该参数取值后，会与服务器断开连接。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 86400（1d）

- 0表示关闭超时设置。
- 正数表示与服务器建立连接并开启事务后，事务空闲并不进行任何操作的最长时间。当超过设定的值后，会与服务器断开连接。

默认值：300（即5min）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值600，表示idle_in_transaction_timeout为600s；带单位取值10min，表示idle_in_transaction_timeout为10min。取值如果要带单位，必须为s、min、h、d。

设置建议：参数值设置越小越安全，但用户使用越不便，依据业务安全诉求设定。

设置不当的风险与影响：参数设置过小，可能会使得事务开启后，客户端应用在运行过程中没有及时发送新的请求，从而导致会话超时退出，影响用户使用。

说明

- GaussDB gsql客户端中有自动重连机制，所以针对初始化用户本地连接，超时后gsql表现的现象为断开后重连。
- idle_in_transaction_timeout与**session_timeout**同时开启时，在事务空闲状态下都会生效，以配置较小的超时时间为准。
- 新下发实例该参数取值为300，升级实例默认为0。
- 空闲事务超时后，日志中会打印事务的开始时间、线程id、应用名以及query等信息。该日志用于空闲事务超时退出问题定位，不包含前缀格式**log_line_prefix**。

ssl

参数说明：设置服务端是否开启SSL连接。请在使用这个选项之前阅读《开发指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示服务端开启SSL连接，实际通信过程中是否为SSL加密传输由客户端配置决定。
- off：表示服务端关闭SSL连接。

默认值：on（CN实例）；off（DN实例）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：对安全性要求高，建议开启，对性能要求高，建议关闭。GaussDB目前支持SSL的场景为客户端连接CN场景，该参数目前建议只在CN中开启，DN默认值为off。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：开启此参数需要同时确保ssl_cert_file、ssl_key_file和ssl_ca_file参数配置正确，如果使用国密认证，还需要确保ssl_enc_cert_file和ssl_enc_key_file参数配置正确，不正确的配置可能会导致集群无法正常启动。

说明

主备之间的SSL通信，也会受到GUC参数wal_replication_ssl的影响

comm_ssl

参数说明：设置是否启用主DN之间的SSL连接。请在使用这个选项之前阅读《开发指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示主DN之间启用SSL连接。
- off：表示主DN之间不启用SSL连接。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：该参数目前建议只在DN中开启，CN默认值为off。

设置不当的风险与影响：开启此参数需要同时确保ssl_cert_file、ssl_key_file和ssl_ca_file参数配置正确，不正确的配置可能会导致集群无法正常启动。

require_ssl

参数说明：设置服务端是否强制要求SSL连接，该参数只有当参数ssl为on时才有效。请在使用这个选项之前阅读《开发指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示服务器端强制要求SSL连接。
- off：表示服务器端对是否通过SSL连接不作强制要求。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：GaussDB目前支持SSL的场景为客户端连接CN场景，该参数目前建议只在CN中开启。

设置不当的风险与影响：在SSL相关参数配置有误的场景下，该参数打开可能导致通信建连失败，甚至导致集群启动失败。

版权所有 © 华为云计算技术有限公司

ssl_ciphers

参数说明：指定SSL支持的加密算法列表，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：字符串，如果指定多个加密算法，加密算法之间需要以分号（;）分隔。

默认值："ALL"，表示允许对端使用产品支持的所有加密算法，但不包含ADH、LOW、EXP、MD5算法。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：按照规范进行该参数值的设置，不建议指定不安全算法DHE-RSA-AES128-GCM-SHA256和DHE-RSA-AES256-GCM-SHA384。

设置不当的风险与影响：ssl_ciphers设置错误会导致数据库不能正常启动。

ssl_renegotiation_limit

参数说明：指定在会话密钥重新协商之前，通过SSL加密通道可以传输的流量。这个重新协商流量限制机制可以减少攻击者针对大量数据使用密码分析法破解密钥的几率，但是也带来较大的性能损失。流量是指发送和接受的流量总和。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647。其中0表示禁用重新协商机制。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示ssl_renegotiation_limit为1024kB；带单位取值1MB，表示ssl_renegotiation_limit为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：建议不做修改。使用SSL重协商机制可能引入其他风险，因此已禁用SSL重协商机制，为保持版本兼容保留此参数，修改参数配置不再起作用。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

ssl_cert_file

参数说明：指定包含SSL服务器证书的文件的名称，其相对路径是相对于数据目录的。

参数类型：字符串

参数单位：无

取值范围：合法文件名称，需要以“.crt”结尾。

默认值："server.crt"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：配置文件路径时，需要确保其与其余SSL相关证书/私钥配置文件是配套关系。

设置不当的风险与影响：文件路径错误时，会导致客户端与数据库节点之间无法建立SSL连接，数据库集群节点之间无法进行SSL通信，数据存在泄漏风险。

ssl_key_file

参数说明：指定包含SSL私钥的文件的名称，其相对路径是相对于数据目录的。

参数类型：字符串

参数单位：无

取值范围：合法文件名称，需要以“.key”结尾。

默认值："server.key"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：配置文件路径时，需要确保其与其余SSL相关证书/私钥配置文件是配套关系。

设置不当的风险与影响：文件路径错误时，会导致客户端与数据库节点之间无法建立SSL连接，数据库集群节点之间无法进行SSL通信，数据存在泄漏风险。

ssl_enc_cert_file

参数说明：指定包含SSL服务器国密加密证书的文件的名称，其相对路径是相对于数据目录的。

参数类型：字符串

参数单位：无

取值范围：合法文件名称，需要以“.crt”结尾。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：配置文件路径时，需要确保其与其余国密SSL相关证书/私钥配置文件是配套关系。

设置不当的风险与影响：文件路径错误时，会导致客户端与数据库节点之间无法建立国密SSL连接，数据库集群节点之间无法进行国密SSL通信，数据存在泄漏风险。

ssl_enc_key_file

参数说明：指定包含SSL服务器国密加密证书私钥的文件的名称，其相对路径是相对于数据目录的。

参数类型：字符串

参数单位：无

取值范围：合法文件名称，需要以“.key”结尾。

版权所有 © 华为云计算技术有限公司

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：配置文件路径时，需要确保其与其余国密SSL相关证书/私钥配置文件是配套关系。

设置不当的风险与影响：文件路径错误时，会导致客户端与数据库节点之间无法建立国密SSL连接，数据库集群节点之间无法进行国密SSL通信，数据存在泄漏风险。

ssl_ca_file

参数说明：指定包含CA信息的根证书名称，其相对路径是相对于数据目录的。

参数类型：字符串

参数单位：无

取值范围：合法文件名称，需要以“.pem”结尾。其中空字符串表示没有CA文件被加载，不进行客户端证书验证。

默认值："cacert.pem"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需要进行客户端证书验证时，请正确配置文件路径。

设置不当的风险与影响：文件路径错误时，会导致数据库无法校验客户端证书，通信安全性降低。

ssl_crl_file

参数说明：指定包含证书吊销列表的文件的名称，如果客户端证书在该列表中，则当前客户端证书被视为无效证书，其相对路径是相对于数据目录的。

参数类型：字符串

参数单位：无

取值范围：合法文件名称，空字符串表示没有吊销列表。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若吊销列表中文件名称配置错误，可能导致无效的客户端证书可完成SSL建连，存在数据泄露风险；有效的客户端证书被视为已吊销，无法建立SSL连接。

enable_shared_sslctx

参数说明：是否开启SSL证书全局加载并共享SSL上下文，仅在SSL参数配置为on时生效。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示开启SSL证书全局加载，仅在数据库启动时进行一次SSL上下文初始化动作。
- off：表示关闭SSL证书全局加载，每次会话建连时均进行SSL上下文初始化动作。

默认值：off，操作系统glibc版本小于2.28时为on。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议关闭；仅在对SSL非双向认证场景的性能要求极高时，建议开启。

设置不当的风险与影响：SSL证书全局加载功能生效后，替换证书无法实时生效，需要重启数据库。

说明

- 在操作系统glibc版本小于2.28时，由于证书加载效率较低，因此这些系统下参数的默认值为on。
- 若数据库启动阶段，SSL全局上下文初始化失败，则该参数会被自动设置为off，降级为会话级SSL证书加载。

ssl_cert_notify_time

参数说明：SSL服务器证书到期前提醒的天数。建立连接初始化ssl证书时，若当前时间距离证书到期时间小于设定值，则在日志中打印过期提醒。

参数类型：整型

参数单位：d（天）

取值范围：7 ~ 180

默认值：90

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值如果要带单位，必须为d。

设置建议：推荐使用默认值。

设置不当的风险与影响：参数设置过小，可能导致服务端证书过期没有被提前感知，导致无法及时更换证书；设置过大，可能导致频繁告警。

krb_server_keyfile

参数说明：指定Kerberos服务主配置文件的位置，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法文件路径。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：请联系技术支持设置。

设置不当的风险与影响：设置不当会导致集群间节点无法使用Kerberos认证功能。

krb_srvname

参数说明：设置Kerberos服务名。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值："postgres"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

krb_caseins_users

参数说明：设置Kerberos用户名是否大小写敏感。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示大小写不敏感。
- off：表示大小写敏感。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

modify_initial_password

参数说明：当GaussDB安装成功后，数据库中仅存在一个初始用户（UID为10的用户）。该参数控制的是，客户通过该账户初次登录数据库进行操作时，是否要对该初始账户的密码进行修改。

说明

如果安装过程中未指定初始用户密码，则安装后初始用户密码默认为空，执行其他操作前需要先通过gsql客户端设置初始用户的密码。此参数功能不再生效，保留此参数仅为兼容升级场景。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示集群安装成功后初始用户首次登录操作前需要修改初始密码。
- off：表示集群安装成功后初始用户无需修改初始密码即可进行操作。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

password_policy

参数说明：该参数控制的是，在使用CREATE ROLE/USER或者ALTER ROLE/USER命令创建或者修改GaussDB账户时，是否进行密码复杂度检查。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示不采用密码复杂度校验策略。
- 1：表示采用密码复杂度校验策略。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：从安全性考虑，请勿关闭密码复杂度策略。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

password_reuse_time

参数说明：该参数控制的是，在使用ALTER USER或者ALTER ROLE修改用户密码时，是否对新密码进行可重用天数检查。

说明

修改密码时会检查配置参数**password_reuse_time**和**password_reuse_max**。

- 当**password_reuse_time**和**password_reuse_max**都为正数时，只要满足其中任一个，即可认为密码可以重用。
- 当**password_reuse_time**为0时，表示不限制密码重用天数，仅限制密码重用次数。
- 当**password_reuse_max**为0时，表示不限制密码重用次数，仅限制密码重用天数。
- 当**password_reuse_time**和**password_reuse_max**都为0时，表示不对密码重用进行限制。

参数类型：浮点型

参数单位：d（天）

取值范围：0 ~ 3650

- 0表示不检查密码可重用的天数。
- 正数表示新密码不能为该值指定的天数内使用过的密码。

默认值：0

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：参数值设置越大越安全，但是在使用过程中会带来不便，依据业务安全诉求设定。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

password_reuse_max

参数说明：该参数控制的是，在使用ALTER USER或者ALTER ROLE修改用户密码时，是否对新密码进行可重用次数检查，仅sysadmin用户可以访问。

说明

修改密码时会检查配置参数**password_reuse_time**和**password_reuse_max**。

- 当**password_reuse_time**和**password_reuse_max**都为正数时，只要满足其中任一个，即可认为密码可以重用。
- 当**password_reuse_time**为0时，表示不限制密码重用天数，仅限制密码重用次数。
- 当**password_reuse_max**为0时，表示不限制密码重用次数，仅限制密码重用天数。
- 当**password_reuse_time**和**password_reuse_max**都为0时，表示不对密码重用进行限制。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000

- 0表示不检查密码可重用次数。
- 正整数表示新密码不能为该值指定的次数内使用过的密码。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：参数值设置越大越安全，但是在使用过程中会带来不便，请依据业务安全诉求进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

password_lock_time

参数说明：该参数指定账户被锁定后自动解锁所需的时间。

说明

只有当password_lock_time和**failed_login_attempts**的取值都为正数时，账户锁定和解锁功能才能生效。

参数类型：浮点型

参数单位：d（天）

取值范围：0 ~ 365。取值的整数部分表示天数，小数部分可以换算成时、分、秒，如：password_lock_time=1.5，表示1天零12小时。

- 0表示密码验证失败时，自动锁定功能不生效。
- 正数表示账户被锁定后，当锁定时间超过password_lock_time设定的值时，账户将会被自行解锁。

版权所有 © 华为云计算技术有限公司

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值如果要带单位，必须为d。

设置建议：设置为默认值。自动解锁时间不宜设置过长，需要同**failed_login_attempts**参数设置保持在合理范围内。

设置不当的风险与影响：如果**failed_login_attempts**参数设置过小，自动解锁时间设置过长，会导致因误输入导致账号长时间无法使用，影响用户正常使用。

failed_login_attempts

参数说明：在任意时候，如果输入密码错误的次数达到failed_login_attempts参数所设定的值，则当前账户被锁定，在经过**password_lock_time**参数所设定的时间后才会被自动解锁，期间仅sysadmin用户可以访问。登录时输入密码失败、ALTER USER时修改密码失败等情形，都会被视作输入密码错误。

说明

只有当failed_login_attempts和**password_lock_time**的取值都为正数时，账户锁定和解锁功能才能生效。

参数类型：整型

参数单位：次

取值范围：0 ~ 1000。

- 0表示自动锁定功能不生效。
- 正整数表示当错误密码次数达到failed_login_attempts设定的值时，当前账户将被锁定。

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为5~10。

设置不当的风险与影响：如果参数过大，密码错误次数输入过多（小于最大值1000）但不采取安全措施的话有安全风险，如果参数过小（大于0小于10）则可能会因为误输入导致账户被锁定影响正常使用。

enable_lock_account

参数说明：将**failed_login_attempts**设置为0时，自动锁定功能将被关闭。在此设置下，系统无法记录账号的登录失败次数。若使用enable_lock_account关闭自动锁定功能，账号使用错误密码连续登录失败达到参数**failed_login_attempts**所设定的值后，则不会自动锁定账号。此时，系统将通过上报告警来保障账号安全。管理员手动锁定不受参数控制。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示账号使用错误密码连续登录失败达到参数failed_login_attempts所设定的值后，自动锁定账号。

版权所有 © 华为云计算技术有限公司

- off：表示账号使用错误密码连续登录失败达到参数failed_login_attempts所设定的值后，不自动锁定账号。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：从安全性考虑，如未启用告警功能，请勿关闭自动锁定账号功能。

设置不当的风险与影响：设置为off会降低账号安全性。

password_encryption_type

参数说明：该参数决定采用何种加密方式对用户密码进行加密存储。修改此参数的配置不会自动触发已有用户密码加密方式的修改，只会对新创建用户的密码或已有用户修改的密码采用新的加密方式进行加密。

参数类型：整型

参数单位：无

取值范围：0、1、2、3

- 0：表示采用md5方式对密码加密。
- 1：表示采用sha256和md5两种方式分别对密码加密。
- 2：表示采用sha256方式对密码加密。
- 3：表示采用sm3方式对密码加密。

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：MD5加密算法安全性低，不建议使用，设置为0或1时存在安全风险。

说明

FIPS模式不支持md5和sm3算法，当password_encryption_type参数设置为0、1、3时将失效，强制使用sha256算法加密用户密码。

password_min_length

参数说明：该参数决定账户密码的最小长度，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：6 ~ 999

默认值：8

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求配置，推荐使用默认值。

设置不当的风险与影响：设置长度超过**password_max_length**会导致账户无法正常创建。

版权所有 © 华为云计算技术有限公司

password_max_length

参数说明：该参数决定账户密码的最大长度，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：6 ~ 999

默认值：32

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求配置，推荐使用默认值。

设置不当的风险与影响：设置长度小于**password_min_length**会导致账户无法正常创建。

password_min_uppercase

参数说明：该参数决定账户密码中至少需要包含的大写字母的个数，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：0 ~ 999

- 0：表示没有限制。
- 1 ~ 999：表示创建账户所指定的密码中至少需要包含的大写字母的个数。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求配置，推荐使用默认值。

设置不当的风险与影响：设置长度超过**password_max_length**会导致账户无法正常创建。

password_min_lowercase

参数说明：该参数决定账户密码中至少需要包含的小写字母的个数，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：0 ~ 999

- 0：表示没有限制。
- 1 ~ 999：表示创建账户所指定的密码中至少需要包含的小写字母的个数。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求配置，推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置长度超过**password_max_length**会导致账户无法正常创建。

password_min_digital

参数说明：该参数决定账户密码中至少需要包含的数字的个数，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：0 ~ 999

- 0：表示没有限制。
- 1 ~ 999：表示创建账户所指定的密码中至少需要包含的数字的个数。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求配置，推荐使用默认值。

设置不当的风险与影响：设置长度超过**password_max_length**会导致账户无法正常创建。

password_min_special

参数说明：该参数决定账户密码中至少需要包含的特殊字符的个数，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：0 ~ 999

- 0：表示没有限制。
- 1 ~ 999：表示创建账户所指定的密码中至少需要包含的特殊字符的个数。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求配置，推荐使用默认值。

设置不当的风险与影响：设置长度超过**password_max_length**会导致账户无法正常创建。

password_effect_time

参数说明：该参数决定账户密码的有效时间。

参数类型：浮点型

参数单位：d（天）

取值范围：0 ~ 999。整数部分表示天数，小数部分可以换算成时、分、秒，如：password_effect_time=0.5，表示0天零12小时。

版权所有 © 华为云计算技术有限公司

- 0：表示不开启账户密码有效期限制功能。
- 1 ~ 999：表示创建账户所指定的密码有效期，临近或超过有效期时，系统会提示用户修改密码。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐配置用户密码的有效时间，参数设置越小越安全，但对用户使用带来不便，请依据业务安全诉求进行设置。

设置不当的风险与影响：设置为0时，密码长期不修改会存在安全风险。

password_notify_time

参数说明：该参数决定账户密码到期前提醒的天数。

参数类型：整型

参数单位：d（天）

取值范围：0 ~ 999

- 0：表示不开启提醒功能。
- 1 ~ 999：表示账户密码到期前提醒的天数。

默认值：7

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为0时，可能导致账户密码到期没有被提前感知。

enable_innertool_cert

参数说明：该参数决定内部工具是否使用证书认证。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示内部工具使用证书认证。
- off：表示内部工具不使用证书认证。

默认值：off

说明

- 此参数必须与参数**ssl**同时打开才能生效。
- 此参数打开后，使用的证书默认配置的是$GAUSSHOME/share/sslcert/gsql目录下的证书，证书的Common Name必须与初始用户相同，否则认证失败。
- 此参数打开后，初始用户使用证书认证，其他用户则需要使用密码方式认证。
- 此参数打开后，允许初始用户使用证书认证方式远程连接数据库。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值，如果用户对安全性要求比较高，建议打开参数，并正确配置相关证书。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ldap_ca_file

参数说明：设置LDAP客户端证书路径。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值："/etc/openldap/certs/cacert.pem"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际证书位置设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.3.3 通信库参数

本节介绍通信库相关的参数设置及取值范围等内容。

tcp_keepalives_idle

参数说明：在支持TCP_KEEPIDLE套接字选项的系统上，设置发送保持活跃信号的间隔秒数。不设置发送保持活跃信号，连接就会处于闲置状态。

说明

- 如果操作系统不支持TCP_KEEPIDLE选项，这个参数的值必须为0。
- 在通过Unix域套接字进行的连接的操作系统上，这个参数将被忽略。
- 将该值设置为0时，将使用系统的值。
- 该参数在不同的会话之间不共享，也就是说不同的会话连接可能有不同的值。
- 查看该参数时查出来的是当前会话连接内的参数值，而不是GUC副本的值。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

默认值：60

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示tcp_keepalives_idle为60s；带单位取值1min，表示tcp_keepalives_idle为1min。取值如果要带单位，必须为s、min、h、d。

设置建议：设置时请确认操作系统是否支持TCP_KEEPIDLE套接字选项。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

tcp_keepalives_interval

参数说明：在支持TCP_KEEPINTVL套接字选项的操作系统上，设置重新传输之间等待响应的时间。

说明

- 如果操作系统不支持TCP_KEEPINTVL选项，这个参数的值必须为0。
- 在通过Unix域套接字进行的连接的操作系统上，这个参数将被忽略。
- 将该值设置为0时，将使用系统的值。
- 该参数在不同的会话之间不共享，也就是说不同的会话连接可能有不同的值。
- 查看该参数时查出来的是当前会话连接内的参数值，而不是GUC副本的值。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 180

默认值：30

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。例如，不带单位取值60，表示tcp_keepalives_interval为60s；带单位取值1min，表示tcp_keepalives_interval为1min。取值如果要带单位，必须为s、min、h、d。

设置建议：设置时请确认操作系统是否支持TCP_KEEPINTVL套接字选项。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

tcp_keepalives_count

参数说明：在支持TCP_KEEPCNT套接字选项的操作系统上，设置GaussDB服务端在断开与客户端连接之前可以等待的保持活跃信号个数。

说明

- 如果操作系统不支持TCP_KEEPCNT选项，这个参数的值必须为0。
- 在通过Unix域套接字进行连接的操作系统上，这个参数将被忽略。
- 将该值设置为0时，将使用系统的值。
- 该参数在不同的会话之间不共享，也就是说不同的会话连接可能有不同的值。
- 查看该参数时查出来的是当前会话连接内的参数值，而不是GUC副本的值。

参数类型：整型

参数单位：无

取值范围：0 ~ 100，其中0表示GaussDB未收到客户端反馈的保持活跃信号则立即断开连接。

默认值：20

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：设置时请确认操作系统是否支持TCP_KEEPCNT套接字选项。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

tcp_user_timeout

参数说明：在支持TCP_USER_TIMEOUT套接字选项的操作系统上，设置GaussDB在发送数据时，指定传输的数据在TCP连接被强制关闭之前可以保持未确认状态的最大时长。

说明

- 如果操作系统不支持TCP_USER_TIMEOUT选项，这个参数的值将不生效，默认为0。
- 在通过Unix域套接字进行连接的操作系统上，这个参数将被忽略。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 3600000

默认值：0，表示跟随操作系统设置。

注意，不同操作系统内核下，这个参数生效结果将不同：

- aarch64 EulerOS（Linux内核版本：4.19），超时时间即为该参数设置值。
- x86 Euler2.5（Linux内核版本：3.10），超时时间不是该参数设置值，而是不同区间的最大值，即超时时间取值为：tcp_user_timeout设置值所处“Linux TCP重传总耗时”区间的上限最大值。例如：tcp_user_timeout=40000时，重传总耗时为51秒。

表 6-5 x86 Euler2.5（Linux 内核版本：3.10）tcp_user_timeout 参数取值示意

Linux TCP重传次数	Linux TCP重传总耗时区间（秒）	tcp_user_timeout设置举例（毫秒）	实际Linux TCP重传总耗时（秒）
1	(0.2,0.6]	400	0.6
2	(0.6,1.4]	1000	1.4
3	(1.4,3]	2000	3
4	(3,6.2]	4000	6.2
5	(6.2,12.6]	10000	12.6
6	(12.6,25.4]	20000	25.4
7	(25.4,51]	40000	51
8	(51,102.2]	80000	102.2
9	(102.2,204.6]	150000	204.6
10	(204.6,324.6]	260000	324.6
11	(324.6,444.6]	400000	444.6

注：TCP每次重传耗时随重传次数指数增加，当TCP一次重传到达120秒后，后续每次重传都将耗时120秒不再变化。

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：设置时请确认操作系统是否支持TCP_USER_TIMEOUT套接字选项。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

tcp_syn_retries

参数说明：在支持TCP_SYNCNT套接字选项的操作系统上，设置GaussDB连接其他内部节点时，建立连接三次握手阶段SYN包发送失败而重传的次数。

参数类型：整型

参数单位：无

取值范围：0-127，其中0表示跟随操作系统设置。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：保持默认。

设置不当的风险与影响：该参数设置过小（大于0）时，网络波动情况下，会导致SYN重传次数较少，若重传时长少于网络恢复时长，会导致建连失败；该参数设置过大时，网络不通情况下，会导致SYN重传次数较多，重传时间较长，造成整体报错的时间变长。

须知

- 如果操作系统不支持TCP_SYNCNT选项，这个参数的值将不生效，默认为0。
- 在通过Unix域套接字进行连接的操作系统上，这个参数将被忽略。

comm_tcp_mode

参数说明：通信库使用TCP协议建立数据通道的切换开关，重启集群生效。

说明

SCTP协议的连接不再提供支持，为了保持兼容，提供此参数的接口，但此参数会在设置过程中强制改为on。

参数类型：布尔型

参数单位：无

取值范围：CN设置为on表示使用TCP模式连接DN，DN设置为on表示DN间使用TCP代理通信。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

comm_sctp_port

参数说明：TCP代理通信库使用的TCP协议侦听端口，负责侦听数据报文通道。

参数类型：整型

参数单位：无

取值范围：0 ~ 65535

默认值：25110（实际值为GUC参数**port**值+2，取决于用户配置）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议用户配置，集群安装时会自动下发相关配置。

设置不当的风险与影响：集群部署时会自动分配此端口号，请不要轻易修改此参数，如端口号配置不正确会导致数据库通信失败。

comm_control_port

参数说明：TCP代理通信库使用的TCP协议监听端口。

参数类型：整型

参数单位：无

取值范围：0 ~ 65535

默认值：25111（实际值为GUC参数**port**值+3，取决于用户配置）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议用户配置，集群安装时会自动下发相关配置。

设置不当的风险与影响：集群部署时会自动分配此端口号，请不要轻易修改此参数，如端口号配置不正确会导致数据库通信失败。

comm_max_datanode

参数说明：TCP代理通信库支持的最大DN数。

参数类型：整型

参数单位：无

取值范围：1 ~ 8192

默认值：单个节点支持的最大主DN数量。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置，缩容时该参数需重启集群才会生效。

设置建议：推荐设置为256。

设置不当的风险与影响：参数取值小于当前集群DN数量时，会导致建立连接失败。设置过大会占用过多内存。

版权所有 © 华为云计算技术有限公司

comm_max_stream

参数说明：TCP代理通信库支持的最大并发数据流数。该参数必须大于业务并发数*每并发平均stream算子数***query_dop**的平方。

参数类型：整型

参数单位：无

取值范围：2 ~ 60000

默认值：1024

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 不建议该参数值设置过大，因为comm_max_stream会占用内存（占用内存=380byte*comm_max_stream*(comm_max_datanode+max_coordinators)），若并发数据流过大，查询较为复杂及smp过大都会导致内存不足。
- 如果comm_max_stream参数值较小，进程内存充足，可以适当将comm_max_stream值调大。

设置不当的风险与影响：参数设置过小会限制大并发分布式stream查询，产生“Cannot get stream index,maybe comm_max_stream is not enough”报错；设置过大会占用过多内存底噪，并行运行业务导致消耗更多系统资源。修改必须详细确认参数的规格限制，并考虑硬件资源（如CPU、内存）是否足够，否则可能导致集群异常。业务上线前，需要完成分布式业务改造，并按照实际业务场景压测，避免配置不当导致报错。

comm_max_receiver

参数说明：TCP代理通信库内部接收线程数量。

参数类型：整型

参数单位：无

取值范围：1 ~ 50

默认值：4

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，避免过多的线程资源，带来的线程切换调度。

设置不当的风险与影响：若设置过大会启动过多数据接收线程，造成线程资源浪费。

comm_quota_size

参数说明：TCP代理通信库最大可连续发送包总大小。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2048000

版权所有 © 华为云计算技术有限公司

默认值：1024

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示comm_quota_size为1024kB；带单位取值1MB，表示comm_quota_size为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。使用1GE网卡时，建议取较小值，推荐设置为20kB~40kB。

设置不当的风险与影响：设置过小可能会导致TCP通信库收发阶段等待quota。

comm_usable_memory

参数说明：单个DN内TCP代理通信库缓存最大可使用内存。

参数类型：整型

参数单位：kB

取值范围：102400 ~ 1073741823

默认值：4096000（即4000MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值102400，表示comm_usable_memory为102400kB；带单位取值100MB，表示comm_usable_memory为100MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。此参数需根据环境内存及部署方式具体配置。

设置不当的风险与影响：参数设置过大会造成OOM，过小会降低TCP代理通信库性能。

comm_memory_pool

参数说明：单个DN内TCP代理通信库可使用内存池资源的容量大小。

参数类型：整型

参数单位：kB

取值范围：102400 ~ 1073741823

默认值：2048000（即2000MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值102400，表示comm_memory_pool为102400kB；带单位取值100MB，表示comm_memory_pool为100MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。此参数需根据实际业务情况做调整，若通信库使用内存小，可设置该参数数值较小，反之设置数值较大。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。修改时必须详细确认参数的规格限制，并考虑硬件资源是否足够，否则可能导致集群异常。

版权所有 © 华为云计算技术有限公司

comm_memory_pool_percent

参数说明：单个DN内TCP代理通信库可使用内存池资源的百分比，用于自适应负载预留通信库通信消耗的内存大小。

参数类型：整型

参数单位：无

取值范围：0 ~ 100

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。此参数需根据实际业务情况做调整，若通信库使用内存小，可设置该参数数值较小，反之设置数值较大。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

comm_client_bind

参数说明：通信库客户端发起连接时是否使用bind绑定指定IP。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示绑定指定IP。
- off：表示不绑定指定IP。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果集群某一节点存在多个IP处于同一通信网段时，需设置为on。此时将绑定本地**listen_addresses**指定的IP发起通信，随机端口号不能重复使用，集群并发数量会受到可用随机端口号数量的限制。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

comm_no_delay

参数说明：是否使用通信库连接的NO_DELAY属性。修改参数取值后，重启集群生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用NO_DELAY属性。
- off：表示不使用NO_DELAY属性。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：总体上，时延敏感型业务建议打开；吞吐要求高时，建议关闭。此外，当集群出现因每秒接收数据包过多导致的丢包时，需设置为off，以便小包合并成大包发送，减少数据包总数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

comm_debug_mode

参数说明：TCP代理通信库debug模式开关，该参数设置是否打印通信层详细日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打印通信库详细debug日志。
- off：表示不打印通信库详细debug日志。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置为on时，打印日志量较大，会增加额外的overhead并降低数据库性能，建议仅在调试时打开。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

comm_ackchk_time

参数说明：无数据包接收情况下，该参数设置通信库服务端主动ACK触发时长。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 20000。取值为0表示关闭此功能。

默认值：2000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示comm_ackchk_time为60ms；带单位取值1min，表示comm_ackchk_time为1min。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

comm_timer_mode

参数说明：TCP代理通信库timer模式开关，该参数设置是否打印通信层各阶段时间桩。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示打印通信库详细时间桩日志。
- off：表示不打印通信库详细时间桩日志。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置为on时，打印日志量较大，会增加额外的overhead并降低数据库性能，建议仅在调试时打开。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

comm_stat_mode

参数说明：TCP代理通信库stat模式开关，该参数设置是否打印通信层的统计信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打印通信库统计信息日志。
- off：表示不打印通信库统计信息日志。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置为on时，打印日志量较大，会增加额外的overhead并降低数据库性能，建议仅在调试时打开。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_stateless_pooler_reuse

参数说明：pooler连接池复用切换开关，开启后可对已有的空闲TCP连接进行复用，重启集群生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用pooler复用模式。
- off：表示关闭pooler复用模式。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：CN和DN需要同步设置。如果CN设置enable_stateless_pooler_reuse为off，DN设置enable_stateless_pooler_reuse为on会导致集群不能正常通信，因此必须对该参数做CN和DN全局相同的配置，重启集群生效。

设置不当的风险与影响：当该参数关闭时，pooler连接池复用粒度会降低，DN的内存消耗可能会增加。

版权所有 © 华为云计算技术有限公司

comm_cn_dn_logic_conn

参数说明：CN和DN间逻辑连接特性开关，重启集群生效。

说明

不再提供CN和DN之间的逻辑连接支持，为了保持兼容，提供此参数的接口，但此参数会在设置过程中强制改为off。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示CN和DN之间连接为逻辑连接，使用libcomm组件。
- off：表示CN和DN之间连接为物理连接，使用libpq组件。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数开启时，CN和DN之间会使用逻辑连接用于解决socket数量上限瓶颈，逻辑连接通信库会有性能开销。

6.3.4 资源消耗

6.3.4.1 内存

介绍与内存相关的参数设置。

说明

这些参数只能在数据库服务重新启动后生效，local_syscache_threshold除外。

memorypool_enable

参数说明：设置是否允许使用内存池。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许使用内存池。
- off：表示不允许使用内存池。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

memorypool_size

参数说明：设置内存池大小。

参数类型：整型

参数单位：kB

取值范围：131072 ~ 1073741823

默认值：524288（即512MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值131072，表示memorypool_size为131072kB；带单位取值128MB，表示memorypool_size为128MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_memory_limit

参数说明：设置是否启用逻辑内存管理模块。

说明

由于存在固定开销，即**shared_buffers**和元数据（大约200MB），若**max_process_memory**减去固定开销小于2GB，GaussDB会强制把enable_memory_limit设置为off。其中元数据是GaussDB内部使用的内存，和部分并发参数，如**max_connections**、**thread_pool_attr**、**max_prepared_transactions**等参数相关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用逻辑内存管理模块。
- off：表示不启用逻辑内存管理模块。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当该值为off时，不对数据库使用的内存做限制，在大并发或者复杂查询时，使用内存过多，可能导致操作系统OOM问题。

enable_threads_memory_reserve

参数说明：设置是否启用后台线程内存隔离模块。开启后为后台线程按照后台线程优先级预留内存资源。当开启逻辑内存管理模块（即enable_memory_limit为on）时，设置该参数为on才能启动生效。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示启用后台线程内存隔离模块。
- off：表示不启用后台线程内存隔离模块。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。开启之后不同优先级后台线程预留内存和内存使用上限如下。

内存情况	高优先级	中优先级	低优先级
预留内存情况1（最大可用动态内存 < 10G）	最大可用动态内存/10	最大可用动态内存/10	最大可用动态内存/20
预留内存情况1（最大可用动态内存 < 20G）	0.75GB	1.5GB	0.75GB
预留内存情况1（最大可用动态内存 < 40G）	0.9GB	3.0GB	1.5GB
预留内存情况1（最大可用动态内存 >= 40G）	1.0GB	4.0GB	2.0GB
内存使用上限	无	最大可用动态内存 * 30%	等于预留内存

设置不当的风险与影响：当该参数设置为off时，后台线程内存使用数据会被持续统计，但不对后台线程使用的内存做隔离，在大并发或者复杂查询时，使用内存过多，可能导致后台线程申请不到内存。

max_process_memory

参数说明：设置一个数据库节点可用的最大物理内存。

参数类型：整型

参数单位：kB

取值范围：2097152 ~ 2147483647

默认值：请参见“典型场景设置”。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2097152，表示max_process_memory为2097152kB；带单位取值2GB，表示max_process_memory为2GB。取值如果要带单位，必须为kB、MB、GB。

设置建议：该参数目的是尽可能保证系统的可靠性，不会因数据库内存膨胀导致节点OOM。DN上该数值需要根据系统物理内存及单节点部署主DN个数决定，建议计算公

版权所有 © 华为云计算技术有限公司

式如下：（物理内存大小 – vm.min_free_kbytes）* 0.7 / (n+主DN个数)。这个公式中提到vm.min_free_kbytes，其含义是预留操作系统内存供内核使用，通常用作操作系统内核中通信收发内存分配，至少为5%内存，即max_process_memory = 物理内存 * 0.665 / (n+主DN个数)。其中，当集群规模小于256时，n=1；当集群规模大于256且小于512时，n=2；当集群规模超过512时，n=3；独立部署时，n=0。CN上该数值内存可设置与DN数值一样。

设置不当的风险与影响：若该值设置不合理，比如大于服务器物理内存时，可能导致操作系统OOM问题。

典型场景设置：

以下典型场景设置是针对主推的规格经过专门的测试调优之后，给出的最优出口值配置，具体配置信息如表6-6、表6-7、表6-8、表6-9所示。

- 分布式独立部署（单节点1C或1D）

  表 6-6 分布式独立部署设置

资源配置	CN推荐值	DN推荐值
128C/1024G	900GB	900GB
60C/480G	360GB	360GB
32C/256G	192GB	192GB
16C/128G	96GB	96GB
8C/64G	40GB	40GB
4C/32G	20GB	20GB
4C/16G	10GB	10GB

- 金融版（标准型）（单节点1C2D）

  表 6-7 金融版（标准型）设置

资源配置	CN推荐值	DN推荐值
192C/1536G	300GB	550GB
128C/1024G	200GB	350GB
104C/1024G	200GB	350GB
96C/1024G	200GB	350GB
96C/768G	160GB	260GB
80C/640G	130GB	220GB
72C/576G	120GB	200GB
64C/512G	100GB	180GB
60C/480G	100GB	160GB

版权所有 © 华为云计算技术有限公司

资源配置	CN推荐值	DN推荐值
32C/256G	50GB	80GB
16C/128G	20GB	40GB
8C/64G	10GB	20GB

- 企业版（单节点1C3D）

  表 6-8 企业版设置

资源配置	CN推荐值	DN推荐值
192C/1536G	200GB	400GB
128C/1024G	150GB	250GB
104C/1024G	150GB	250GB
96C/1024G	150GB	250GB
96C/768G	110GB	190GB
80C/640G	90GB	160GB
80C/512G	75GB	125GB
72C/576G	80GB	140GB
64C/512G	75GB	125GB
60C/480G	70GB	120GB
32C/256G	35GB	60GB
16C/128G	15GB	25GB
8C/64G	9GB	15GB

- 金融版（数据计算型）（单节点1C4D）

  表 6-9 金融版（数据计算型）设置

资源配置	CN推荐值	DN推荐值
192C/1536G	160GB	300GB
128C/1024G	120GB	200GB
96C/768G	100GB	150GB
72C/576G	60GB	110GB
64C/512G	60GB	100GB
32C/256G	20GB	40GB

版权所有 © 华为云计算技术有限公司

local_syscache_threshold

参数说明：单个session的系统缓存占用内存的最大值，当实际系统缓存大小超过该阈值时，会触发内存淘汰。如果参数**enable_global_syscache**和**enable_thread_pool**已打开，则该参数描述的是当前线程和绑定到当前线程上的session缓存的总大小。

参数类型：整型

参数单位：kB

取值范围：1024 ~ 524288

默认值：

- 独立部署：16MB

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示local_syscache_threshold为1024kB；带单位取值1MB，表示local_syscache_threshold为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：建议设置为1024的整数倍。底噪内存是2MB，平均每增加一个表或者索引，增加11kB内存。

设置不当的风险与影响：如果设置的值过小，会导致内存频繁淘汰，影响性能。

enable_memory_context_control

参数说明：设置是否启用检查内存上下文是否超过给定限制的功能。仅适用于DEBUG版本。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用最大内存上下文限制检查功能。
- off：表示关闭最大内存上下文限制检查功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

uncontrolled_memory_context

参数说明：启用检查内存上下文是否超过给定限制的功能时，设置不受此功能约束的内存上下文。仅适用于DEBUG版本。

说明

查询该参数时，会在参数值的最前面添加标题含义字符串“MmgrMemoryController white list:”。

版权所有 © 华为云计算技术有限公司

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

huge_pages

参数说明：数据库共享内存的申请启用静态内存大页设置。

说明

要启用静态内存大页功能，需要提前在操作系统内核中设置足够数量的静态内存大页资源以供申请，相关参数为vm.nr_hugepages和vm.nr_overcommit_hugepages，会被设置在/etc/sysctl.conf文件中。

参数类型：字符串

参数单位：无

取值范围：

- on：表示启用静态大页内存申请，如果申请不到内存大页资源，则数据库启动失败。
- off：表示不启用静态大页内存申请，申请普通共享内存。
- try：尝试申请内存静态大页资源，如果申请失败，就申请普通共享内存。

默认值：try

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

shared_buffers

参数说明：设置GaussDB使用的共享内存大小。增加此参数的值会使GaussDB比系统默认设置需要更多的System V共享内存。

参数类型：整型

参数单位：页（8kB）

取值范围：16 ~ 1073741823

默认值：请参见“典型场景设置”。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示shared_buffers为1024*8kB，即8192kB；带单位取值8MB，表示shared_buffers为8MB。取值如果要带单位，必须为kB、MB、GB，且取值必须为8kB的整数倍。

版权所有 © 华为云计算技术有限公司

设置建议：

- 由于GaussDB大部分查询下推，建议DN中此参数设置比CN大。
- 建议设置shared_buffers值为**max_process_memory**的40%以内。
- 如果设置较大的shared_buffers，需要同时增加**checkpoint_segments**的值，因为写入大量新增数据、修改数据需要消耗更多的时间周期。
- 如果调整shared_buffers参数之后，导致进程重启失败，请参见启动失败的报错信息，采用以下解决方案之一：
  - 对应调整操作系统kernel.shmall、kernel.shmmax、kernel.shmmin参数。
  - 执行free -g观察操作系统可用内存和swap空间是否足够，如果内存明显不足，请手动停止其他比较占用内存的用户程序。
  - 对应不同规格，请按照对应建议的默认值设置，不建议修改，避免设置明显不合理（过大或过小）的shared_buffers值。注意应满足条件：**data_replicate_buffer_size** + **segment_buffers** + shared_buffers + **wal_buffers** + **temp_buffers** + **maintenance_work_mem** + **work_mem** + **query_mem** +（备机）**wal_receiver_buffer_size** < **max_process_memory** < 物理机内存大小。

设置不当的风险与影响：如果设置的内存参数数值过大，超过物理内存上限，将会出现数据库由于无法申请到足够的内存而无法成功启动的情况。

典型场景设置：

以下典型场景设置是针对主推的规格经过专门的测试调优之后，给出的最优出口值配置，具体配置信息如**表6-10**、**表6-11**、**表6-12**、**表6-13**所示。

- 分布式独立部署（单节点1C或1D）

  表 6-10 分布式独立部署设置

资源配置	CN推荐值	DN推荐值	CN取值范围	DN取值范围
128C/1024G	8GB	304GB	256MB ~ 409.6GB	2GB ~ 409.6GB
60C/480G	4GB	140GB	256MB ~ 192GB	2GB ~ 192GB
32C/256G	2GB	76GB	256MB ~ 102.4GB	2GB ~ 102.4GB
16C/128G	2GB	40GB	256MB ~ 51.2GB	2GB ~ 51.2GB
8C/64G	1GB	16GB	256MB ~ 25.6GB	2GB ~ 25.6GB
4C/32G	512MB	8GB	256MB ~ 12.8GB	2GB ~ 12.8GB
4C/16G	256MB	4GB	256MB ~ 6.4GB	2GB ~ 6.4GB

- 金融版（标准型）（单节点1C2D）

版权所有 © 华为云计算技术有限公司

表 6-11 金融版（标准型）设置

资源配置	CN推荐值	DN推荐值
192C/1536G	2GB	220GB
128C/1024G	2GB	140GB
104C/1024G	2GB	140GB
96C/1024G	2GB	140GB
96C/768G	2GB	100GB
80C/640G	2GB	80GB
72C/576G	2GB	80GB
64C/512G	2GB	70GB
60C/480G	2GB	60GB
32C/256G	1GB	30GB
16C/128G	1GB	16GB
8C/64G	512MB	8GB

- 企业版（单节点1C3D）

表 6-12 企业版设置

资源配置	CN推荐值	DN推荐值
192C/1536G	2GB	160GB
128C/1024G	2GB	100GB
104C/1024G	2GB	100GB
96C/1024G	2GB	100GB
96C/768G	2GB	76GB
80C/640G	2GB	64GB
80C/512G	2GB	50GB
72C/576G	2GB	56GB
64C/512G	2GB	50GB
60C/480G	2GB	48GB
32C/256G	1GB	24GB
16C/128G	1GB	10GB
8C/64G	512MB	6GB

版权所有 © 华为云计算技术有限公司

- 金融版（数据计算型）（单节点1C4D）

  表 6-13 金融版（数据计算型）设置

资源配置	CN推荐值	DN推荐值
192C/1536G	2GB	120GB
128C/1024G	2GB	80GB
96C/768G	2GB	50GB
72C/576G	1GB	40GB
64C/512G	1GB	30GB
32C/256G	512MB	10GB

enable_scale_cache

参数说明：ScaleCache通过提高缓冲区页面pin和unpin的本地性和SIMD加速哈希检索，以及copy on write的无锁索引扫描技术来实现缓冲区管理的优化，进而提升整体缓冲区的性能。该参数仅支持在ARM 8及以上平台环境设置。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启ScaleCache缓冲区管理模式。
- off：表示关闭ScaleCache缓冲区管理模式。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

pin_buffer_group_cores

参数说明：ScaleCache功能，在pin和unpin优化时会对CPU核进行分组，该参数表示分组时每组CPU的核数。当enable_scale_cache参数为on时，该参数才允许被设置和修改。该参数仅支持在ARM 8及以上平台环境设置。

参数类型：整型

参数单位：个

取值范围：1 ~ 1073741823

默认值：4

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用2的幂次方，如果CPU核数或者设置的值非2的幂次方，那么该参数就会被设置为1。

设置不当的风险与影响：修改前请确认enable_scale_cache参数已开启。请在充分理解参数含义，并经过测试验证后进行修改。

enable_shared_storage_aio

参数说明：该功能用来控制打开或关闭Dorado共享存储模式并行异步写日志优化功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启Dorado共享存储异步写功能。
- off：表示关闭Dorado共享内存异步写功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

page_version_check

参数说明：基于页面版本信息，进行底层存储故障校验和页面漏置脏校验的功能总开关，page_version_check是一个三级开关。其中，底层存储故障校验功能是检测从底层存储读取的页面是否为正确版本，防止磁盘掉电等故障情况导致的页面版本信息丢失；页面漏置脏校验功能是检测被修改页面是否忘记置脏，它有一个单独开关**page_missing_dirty_check**进行控制。

参数类型：枚举类型

参数单位：无

取值范围：

- off：表示完全关闭底层存储故障校验和页面漏置脏校验的功能开关。
- memory：表示打开纯内存态的页面版本校验功能（即底层存储故障校验和页面漏置脏校验），页面版本信息只缓存在内存当中，重启丢失。
- persistence：表示打开持久态的页面版本校验功能（即底层存储故障校验和页面漏置脏校验），页面版本信息会持久化到文件中，重启不丢失。

默认值：persistence（192核CPU/1536GB内存，128核CPU/1024GB内存，104核CPU/1024GB内存，96核CPU/1024GB内存，96核CPU/768GB内存，80核CPU/640GB内存，80核CPU/512GB内存，72核CPU/576GB内存，64核CPU/512GB内存，60核CPU/480GB内存，32核CPU/256GB内存，16核CPU/128GB内存）；off（8核CPU/64GB内存，4核CPU/16GB内存，4核CPU/32GB内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据不同机器规格，建议如下设置：persistence（192核CPU/1536GB内存，128核CPU/1024GB内存，104核CPU/1024GB内存，96核CPU/1024GB内存，96

版权所有 © 华为云计算技术有限公司

核CPU/768GB内存，80核CPU/640GB内存，80核CPU/512GB内存，72核CPU/576GB内存，64核CPU/512GB内存，60核CPU/480GB内存，32核CPU/256GB内存，16核CPU/128GB内存）；off（8核CPU/64GB内存，4核CPU/16GB内存，4核CPU/32GB内存）。

设置不当的风险与影响：开关设置为memory及persistence时，对性能有一定影响，机器规格越小，影响越大（参考：16核CPU/128GB内存，TPCC模型影响7%左右）。在系统需要频繁重启的场景下，建议设置为persistence，能够确保页面版本信息不丢失，但会进一步影响性能。

page_version_full_validation

参数说明：在AntiCache持久态下，页面版本在buffer刷脏时即插入AntiCache，这相比在buffer淘汰时才插入的内存态更为频繁，容易因占满AntiCache内存而导致漏校验。为降低漏校验概率、实现100%校验覆盖，需优化持久态的插入方式与Vlog文件回收机制。page_version_full_validation控制是否开启AntiCache全校验模式。开启此参数后，在插入AntiCache时若内存已满，系统将先对一个已有页面执行原地校验以释放空间，再插入新版本。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启AntiCache全校验模式。
- off：表示关闭AntiCache全校验模式。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

page_missing_dirty_check

参数说明：页面漏置脏校验的功能开关，检测被修改页面是否忘记置脏。page_missing_dirty_check受控于**page_version_check**，如果page_version_check设置为off，则page_missing_dirty_check设置为on时也不生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进行页面漏置脏校验。
- off：表示不进行页面漏置脏校验。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议在测试场景中开启此开关，能够最大限度在非生产环境中排查出漏置脏代码BUG；在现网时默认关闭页面漏置脏，避免带来额外开销，进而影响性能。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

page_version_max_num

参数说明：内存中所能缓存的页面版本信息的最大数量，在**page_version_check**取值不为OFF的情况下生效。每个page_version内存占用36字节，请注意内存使用量。

参数类型：整型

参数单位：个

取值范围：0 ~ 2147483647

- 0：根据**shared_buffers**大小自动计算，page_version_max_num=shared_buffers*2。例如，32M shared_buffers对应4096个buffer，所以页面版本信息数量设置为8192个。
- 非0：将强制使用手动配置的数值。不允许小于**page_version_partitions**的16倍，如果小于，则会被强制设置为page_version_partitions*16。

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：内存有限的情况下，合理的数量应该是**shared_buffers**数量的2倍到4倍。对性能有较高要求且内存足够的情况下，建议手动设置为**shared_buffers**数量的4倍以上，且与**page_version_partitions**的比值范围为[256, 1024]。

设置不当的风险与影响：如果设置的内存参数数值过大，超过物理内存上限，将会出现数据库由于无法申请到足够的内存而无法成功启动的情况。

page_version_partitions

参数说明：内存中缓存页面版本信息的哈希表分区数量，分区数量直接影响哈希查询效率和哈希冲突概率。只在**page_version_check**取值不为OFF的情况下生效。

参数类型：整型

参数单位：个

取值范围：0 ~ 2097152

- 0：根据**page_version_max_num**自动计算，page_version_partitions=page_version_max_num/512。若自动计算结果小于4，则强制设置为4。
- 非0：将强制使用手动配置的数值。最小值为4。如果取值小于4，则会被强制设置为4。

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：对性能有较高要求的情况下，建议手动设置为**page_version_max_num**的256分之一到1024分之一左右。

设置不当的风险与影响：如果设置的分区数量太大，会降低哈希查询效率；设置的分区数量太小，会增加哈希冲突的概率。

版权所有 © 华为云计算技术有限公司

page_version_recycler_thread_num

参数说明：控制回收和校验页面版本信息的线程数量，只在**page_version_check**取值不为OFF的情况下生效。

参数类型：整型

参数单位：个

取值范围：0 ~ 16

- **page_version_check**参数值为memory时：
  - 0：根据**page_version_partitions**自动计算，page_version_recycler_thread_num=page_version_partitions/16384。若自动计算结果大于4，则强制设置为4。
  - 非0：将强制使用手动配置的数值。
  - 不允许大于**page_version_partitions**，如果大于，则会被强制设置为page_version_partitions的值。
- **page_version_check**参数值为persistence时：

  小于2时将设置为2，大于等于2时强制使用手动配置的参数。

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_cached_context

参数说明：控制缓存内存上下文的个数，在为0时内存上下文优化框架不生效。

参数类型：整型

参数单位：个

取值范围：0 ~ 1024

- 设置为0时内存上下文优化框架不生效。
- 设置为非0值时，内存上下文优化框架生效，设置的值为缓存内存上下文的个数。

默认值：128

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

verify_log_buffers

参数说明：控制verifyLog buffer大小，只在**page_version_check**取值为persistence的情况下生效。verifyLog buffer内存按页面管理，每页8kB。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：页（8kB）

取值范围：4 ~ 262144

默认值：1GB（192核CPU/1536G内存，128核CPU/1024G内存，104核CPU/1024G内存，96核CPU/1024G内存，96核CPU/768G内存，80核CPU/640G内存，80核CPU/512G内存，72核CPU/576G内存，64核CPU/512G内存，60核CPU/480G内存，32核CPU/256G内存）；512MB（16核CPU/128G内存）；256MB（8核CPU/64G内存）；128MB（4核CPU/32G内存）；16MB（4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，取值131072表示verify_log_buffers为131072 * 8 kB = 1GB；取值131072kB表示verify_log_buffers为131072kB。取值如果带单位，必须为kB、MB、GB，且必须为8kB整数倍。

设置建议：根据系统硬件规格，进行相应的设置。

1GB（192核CPU/1536G内存，128核CPU/1024G内存，104核CPU/1024G内存，96核CPU/1024G内存，96核CPU/768G内存，80核CPU/640G内存，80核CPU/512G内存，72核CPU/576G内存，64核CPU/512G内存，60核CPU/480G内存，32核CPU/256G内存）；512MB（16核CPU/128G内存）；256MB（8核CPU/64G内存）；128MB（4核CPU/32G内存）；16MB（4核CPU/16G内存）。

设置不当的风险与影响：如果设置的内存参数数值过大，超过物理内存上限，将会出现数据库由于无法申请到足够的内存而无法成功启动的情况。

segment_buffers

参数说明：设置GaussDB段页式元数据页的内存大小。

参数类型：整型

参数单位：页（8kB）

取值范围：16 ~ 1073741823

默认值：1024（即8MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示segment_buffers为1024*8kB，即8192kB；带单位取值8MB，表示segment_buffers为8MB。取值如果要带单位，必须为kB、MB、GB，且取值必须为8kB的整数倍。

设置建议：segment_buffers用来缓存段页式段头的内容，属于关键元数据信息，为了提高性能建议常用的表的段头都能缓存在buffer中，不被置换出去。建议按照表的个数（包括索引和toast表）* 分区数 * 3 + 128 来设置。乘以3是因为每个表（分区）会有一些额外的元数据段，一般一个表有3个段。最后加上128因为段页式表空间管理需要一定数量的buffer。该参数设置过小会导致首次创建段页式表时耗时较久，因此请按照默认值进行设置，不建议修改，避免设置明显不合理（过大或过小）的segment_buffers值。注意应满足条件：**data_replicate_buffer_size** + segment_buffers + **shared_buffers** + **wal_buffers** + **temp_buffers** + **maintenance_work_mem** + **work_mem** + **query_mem** + (备机) **wal_receiver_buffer_size** < **max_process_memory** < 物理机内存大小。

设置不当的风险与影响：如果设置的内存参数数值过大，超过物理内存上限，将会出现数据库由于无法申请到足够的内存而无法成功启动的情况。

版权所有 © 华为云计算技术有限公司

bulk_read_ring_size

参数说明：数据并行导出使用的环形缓冲区的大小。

参数类型：整型

参数单位：kB

取值范围：256 ~ 2147483647

默认值：16384（即16MB）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示bulk_read_ring_size为1024kB；带单位取值1MB，表示bulk_read_ring_size为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bulk_write_ring_size

参数说明：数据并行导入使用的环形缓冲区的大小。

参数类型：整型

参数单位：kB

取值范围：16384 ~ 2147483647

默认值：2097152（即2GB）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值16384，表示bulk_write_ring_size为16384kB；带单位取值16MB，表示bulk_write_ring_size为16MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：建议导入压力大的场景中增加DN中此参数取值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

standby_shared_buffers_fraction

参数说明：备实例所在服务器使用**shared_buffers**内存缓冲区大小的比例。

参数类型：浮点型

参数单位：无

取值范围：0.1 ~ 1.0

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

temp_buffers

参数说明：设置每个数据库会话使用的LOCAL临时缓冲区的大小。

说明

- 在每个会话第一次使用临时表之前可以改变temp_buffers的值，之后的设置将是无效的。
- 一个会话将按照temp_buffers给出的限制，根据需要分配临时缓冲区。如果在一个并不需要大量临时缓冲区的会话里设置一个大的数值，其开销只是一个缓冲区描述符的大小。当缓冲区被使用，就会额外消耗8192字节。

参数类型：整型

参数单位：页（8kB）

取值范围：100 ~ 1073741823

默认值：128（即1MB）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值100，表示temp_buffers为100*8kB，即800kB；带单位取值16MB，表示temp_buffers为16MB。取值如果要带单位，必须为kB、MB、GB，且取值必须为8kB的整数倍。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_prepared_transactions

参数说明：设置可以同时处于“预备”状态的事务的最大数目。增加此参数的值会使GaussDB比系统默认设置需要更多的System V共享内存。

参数类型：整型

参数单位：无

取值范围：0 ~ 262143

默认值：

- 独立部署：

  2400（128核CPU/1024G内存）；1200（60核CPU/480G内存，32核CPU/256G内存）；800（16核CPU/128G内存）；400（8核CPU/64G内存）；300（4核CPU/32G内存）；200（4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，当两阶段事务报错槽位不足时，才需要调整。为避免在准备步骤失败，线程池模式下此参数的值应大于**thread_pool_attr**中工作线程个数，非线程池模式下此参数的值不能小于**max_connections**。

设置不当的风险与影响：当GaussDB部署为主备双机时，在备机上此参数的设置必须要高于或等于主机上的，否则无法在备机上进行查询操作。

版权所有 © 华为云计算技术有限公司

work_mem

参数说明：设置内部排序操作和Hash表在开始写入临时磁盘文件之前使用的内存大小。ORDER BY，DISTINCT和MERGE JOINS都要用到排序操作。Hash表在散列连接、散列为基础的聚集、散列为基础的IN子查询处理中都要用到。

对于复杂的查询，可能会同时并发运行好几个排序或者散列操作，每个都可以使用此参数所声明的内存量，不足时会使用临时文件。同样的，多个正在运行的会话可能会同时进行排序操作。因此使用的总内存可能是work_mem的好几倍。

参数类型：整型

参数单位：kB

取值范围：64 ~ 2147483647

默认值：

- 独立部署：

  256MB（128核CPU/1024G内存）；128MB（60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存）；64MB（8核CPU/64G内存）；32MB（4核CPU/32G内存）；16MB（4核CPU/16G内存）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示work_mem为1024kB；带单位取值1MB，表示work_mem为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：

依据查询特点和并发来确定，一旦work_mem限定的物理内存不够，算子运算数据将写入临时表空间，带来5-10倍的性能下降，查询响应时间从秒级下降到分钟级。

- 对于串行无并发的复杂查询场景，平均每个查询有5-10关联操作，建议work_mem=50%内存/10。
- 对于串行无并发的简单查询场景，平均每个查询有2-5个关联操作，建议work_mem=50%内存/5。
- 对于并发场景，建议work_mem=串行下的work_mem/物理并发数。
- 对于BitmapScan的哈希表也会受到work_mem的限制，但不会被严格管控下盘。完全Lossify的情况下，哈希表每占用1MB的内存，对应一次BitmapHeapScan的16GB的页面，达到work_mem上限后，会按此比例随数据访问量线性增长。

设置不当的风险与影响：默认值即推荐值，不建议修改。修改必须详细确认参数的规格限制，并考虑硬件资源是否足够，否则可能导致集群异常。

query_mem

参数说明：设置执行作业所使用的内存。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647

- 如果设置的query_mem值大于32768（即32MB），在生成执行计划时，优化器会将作业的估算内存调整为该值。

版权所有 © 华为云计算技术有限公司

- 如果设置值小于32768（即32MB），将强制设置为默认值0，此时优化器不会根据该值调整作业的估算内存。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值32768，表示query_mem为32768kB；带单位取值32MB，表示query_mem为32MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

query_max_mem

参数说明：设置执行作业所能够使用的最大内存。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647

- 如果设置的query_max_mem值大于32768（即32MB），当作业执行时所使用内存超过该值时，将报错退出。
- 如果设置值小于32768（即32M），将设置为默认值0，此时不会根据该值限制作业的内存使用。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值32768，表示query_max_mem为32768kB；带单位取值32MB，表示query_max_mem为32MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

maintenance_work_mem

参数说明：设置在维护性操作（比如VACUUM、CREATE INDEX等）中可使用的最大的内存。该参数的设置会影响VACUUM、VACUUM FULL、CLUSTER、CREATE INDEX的执行效率。

参数类型：整型

参数单位：kB

取值范围：1024 ~ 2147483647

默认值：

- 独立部署：

  CN：2GB（128核CPU/1024G内存）；1GB（60核CPU/480G内存）；512MB（32核CPU/256G内存）；256MB（16核CPU/128G内存）；128MB（8核CPU/64G内存）；64MB（4核CPU/32G内存）；32MB（4核CPU/16G内存）

  DN：4GB（128核CPU/1024G内存）；2GB（60核CPU/480G内存）；1GB（32核CPU/256G内存）；512MB（16核CPU/128G内存）；256MB（8核CPU/64G内存）；128MB（4核CPU/32G内存）；64MB（4核CPU/16G内存）

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示maintenance_work_mem为1024kB；带单位取值1MB，表示maintenance_work_mem为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：

- 建议设置此参数的值大于**work_mem**，可以改进清理和恢复数据库转储的速度。因为在一个数据库会话里，任意时刻只有一个维护性操作可以执行，并且在执行维护性操作时不会有太多的会话。
- 当**自动清理**线程运行时，**autovacuum_max_workers**倍数的内存将会被分配，所以此时设置maintenance_work_mem的值应该不小于**work_mem**。
- 如果进行大数据量的cluster等，可以在session中调大该值。
- 请按照默认值进行设置，不建议修改，避免设置明显不合理（过大或过小）的maintenance_work_mem值。注意应满足条件：**data_replicate_buffer_size** + **segment_buffers** + **shared_buffers** + **wal_buffers**+ **temp_buffers** + **maintenance_work_mem** + **work_mem** + **query_mem** + (备机) **wal_receiver_buffer_size** < **max_process_memory** < 物理机内存大小。

设置不当的风险与影响：默认值即推荐值，不建议修改。修改必须详细确认参数的规格限制，并考虑硬件资源是否足够，否则可能导致集群异常。

max_stack_depth

参数说明：设置GaussDB执行堆栈的最大安全深度。之所以需要这个安全界限，是因为在服务器里，并非所有程序都检查了堆栈深度，而是只在可能递归的过程，比如表达式计算之类的过程中才进行检查。

参数类型：整型

参数单位：kB

取值范围：100 ~ 2147483647

默认值：

- 系统线程栈大小 - 640kB的值大于等于2MB时，此参数的默认值为2MB。
- 系统线程栈大小 - 640kB的值小于2MB时，此参数的默认值为系统线程栈大小 - 640kB。

须知

- 系统线程栈大小可通过“ulimit -s”命令查看。
- 修改系统线程栈大小后，需重启集群，数据库进程才会采用新的系统线程栈大小。数据库进程在启动时从父进程继承该限制，运行期间不会重新读取。

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示max_stack_depth为1024kB；带单位取值1MB，表示max_stack_depth为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：

- 数据库需要预留640kB堆栈深度，因此，此参数的最佳设置是等于系统线程栈大小 - 640kB。

版权所有 © 华为云计算技术有限公司

- 不支持设置该参数值大于系统线程栈大小 - 640kB，通过gs_guc工具设置该参数值大于系统线程栈大小 - 640kB时，会设置失败。
- 因为并非所有的操作都能够被检测到，所以建议用户在此设置一个明确的值。
- 默认值最大为2MB，这个值相对比较小，不容易导致系统崩溃。

设置不当的风险与影响：

- 系统线程栈大小 - 640kB小于此参数取值范围的最小值时，会导致数据库启动失败。
- 如果设置此参数的值大于实际的内核限制，则一个正在运行的递归函数可能会导致一个独立的服务器进程崩溃。

enable_early_free

参数说明：控制是否可以进行算子内存的提前释放。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持算子内存提前释放。
- off：表示不支持算子内存提前释放。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

memory_trace_level

参数说明：动态内存使用超过最大动态内存的指定百分比阈值（默认是90%，可在**memory_log_config**参数中更改）后，记录内存申请信息的管控等级。该参数仅在GUC参数**use_workload_manager**和**enable_memory_limit**打开时生效。

参数类型：枚举类型

参数单位：无

取值范围：

- none：表示不记录内存申请信息。
- level1：动态内存使用超过最大动态内存的指定百分比阈值后，会记录以下信息，并将记录的内存信息保存在$GAUSSLOG/mem_log目录下。
  - 全局内存概况。
  - instance，session，thread三种类型的所有内存上下文中内存占用前20的内存上下文的内存使用情况。
  - 每个内存上下文的totalsize、freesize字段。
- level2：动态内存使用超过最大动态内存的指定百分比阈值后，会记录以下信息，并将记录的内存信息保存在$GAUSSLOG/mem_log目录下。

版权所有 © 华为云计算技术有限公司

- 全局内存概况。
- instance，session，thread三种类型的所有内存上下文中内存占用前20的内存上下文的内存使用情况。
- 每个内存上下文的totalsize，freesize字段。
- 每个内存上下文上所有内存申请的详细信息，包含申请内存所在的文件，行号和大小。

默认值：level1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数设置为level2后，会记录每个内存上下文的内存申请详情（file，line，size字段），会对性能影响较大，需慎重设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 记录的内存快照信息可以通过系统函数gs_get_history_memory_detail(cstring)查询，函数详情请参见**gs_get_history_memory_detail(cstring)**。
- **use_workload_manager**参数关闭的情况下，如果打开**bypass_workload_manager**，则该参数也会生效，但是因为bypass_workload_manager是SIGHUP类型，使用reload方式设置后需要重启数据库才会使得当前功能生效。
- 记录的内存上下文是经过将同一类型所有重名的内存上下文进行汇总之后得到的。

memory_log_config

参数说明：在动态内存使用百分比超过阈值，或动态内存波动百分比超过阈值，或达到统计周期时，记录每个类型的top n个内存上下文内存信息、内存使用top n个SQL语句的内存信息，并生成内存日志。该参数仅在GUC参数**use_workload_manager**和**enable_memory_limit**打开、**memory_trace_level**非none时生效。

参数类型：字符串

参数单位：无

取值范围：

该参数分为5个部分：'memory_log_threshold, memory_rush_percent, memory_log_period, topn_memory_contexts, topn_queries'，这5个部分的具体含义如下：

- memory_log_threshold：动态内存使用量百分比阈值，取值范围为[0,100]（含0含100）内的整数，取值为0表示关闭此场景触发功能，单位：%。
- memory_rush_percent：动态内存波动百分比阈值，取值范围为[0,100]内的整数，取值为0表示关闭此场景触发功能，单位：%。
- memory_log_period：统计周期，取值范围为[0,INT_MAX)（含0不含INT_MAX；INT_MAX为数据类型INT的最大值2147483647，下同）的整数，取值为0表示关闭此场景触发功能，单位：分钟。
- topn_memory_contexts：每个类型的内存上下文记录前topn_memory_contexts个内存上下文的信息，取值范围为[0,INT_MAX)（含0不含INT_MAX）的整数，取值为0表示关闭此场景触发功能。
- topn_queries：记录内存使用前topn_queries个SQL语句的内存信息，取值范围为[0,INT_MAX)的整数，取值为0表示关闭此场景触发功能。

版权所有 © 华为云计算技术有限公司

默认值：'90,0,0,20,10'，这表示内存使用量百分比阈值为90%，关闭内存波动百分比场景触发功能，关闭达到统计周期场景触发功能，每个类型的内存上下文记录top 20个，SQL语句记录top 10个。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据参数含义合理设置。

设置不当的风险与影响：参数的每个部分需要用英文逗号”,”分隔，因此参数中必须仅有4个英文逗号，将参数分为5个部分，每个部分需要按照上述取值范围中各部分的要求填写，任何不符合要求的情况都会导致此次设置操作失败，所有参数部分均无法生效。请在充分理解参数含义，并经过测试验证后进行修改。

resilience_memory_reject_percent

参数说明：用于控制内存过载逃生功能的动态内存占用百分比。该参数仅在GUC参数**use_workload_manager**和**enable_memory_limit**打开时生效。

参数类型：字符串

参数单位：无

取值范围：长度大于0的字符串。

该参数分为recover_memory_percent、overload_memory_percent两部分，这2个部分的具体含义如下：

- recover_memory_percent：内存从过载状态恢复到正常状态的动态内存使用占最大动态内存的百分比，当动态内存使用小于最大动态内存乘以该值对应的百分比后，停止过载逃生并放开新连接接入，取值为0 ~ 100，设置为多少表示百分之多少。
- overload_memory_percent：内存过载时动态内存使用占最大动态内存的百分比，当动态内存使用大于最大动态内存乘以该值对应的百分比后，表示当前内存已经过载，触发过载逃生kill会话并禁止新连接接入，取值为0 ~ 100，设置为多少表示百分之多少。

默认值："0,0"，表示关闭内存过载逃生功能。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

例如，如下命令表示内存使用超过最大内存上限的90%后禁止新连接接入并kill堆积的会话，kill会话过程中内存恢复到最大内存的70%以下时停止kill会话并允许新连接接入。

kill策略为：优先kill空闲会话；按照会话占用的内存从大到小进行排序，依次kill正在执行的会话。

```
gs_guc reload -Z coordinator -Z datanode -N all -I all -c "resilience_memory_reject_percent='70,90'"
```

设置建议：建议根据实际业务需要进行调整。

设置不当的风险与影响：如果该参数设置的百分比过小，则会频繁触发内存过载逃生流程，会使正在执行的会话被强制退出，新连接短时间接入失败，需要根据实际内存使用情况慎重设置。

版权所有 © 华为云计算技术有限公司

说明

- 最大动态内存和已使用的动态内存可以通过pv_total_memory_detail视图查询获得，最大动态内存：max_dynamic_memory，已使用的动态内存：dynamic_used_memory。
- use_workload_manager参数关闭的情况下，如果打开bypass_workload_manager，则该参数也会生效，但是因为bypass_workload_manager是SIGHUP类型，reload方式设置后需要重启数据库才会使得当前功能生效。
- recover_memory_percent和overload_memory_percent的值可以同时为0，除此之外，recover_memory_percent的值必须要小于overload_memory_percent，否则设置不会生效。
- 将参数从其它值动态设为"0,0"后，若此时已启动会话查杀，不会终止查杀动作，可等待10s以上再连接数据库防止新连接被查杀。

resilience_escape_user_permissions

参数说明：设置用户权限，以逗号分隔，可以设置多个，设置多个则表示多个特殊权限的用户都支持逃生能力，只设置一个则只针对一个特权用户进行逃生。sysadmin控制sysadmin用户的作业是否会被该逃生功能进行cancel处理；monadmin控制monadmin用户的作业是否会被该逃生功能进行cancel处理；默认为空，表示关闭sysadmin和monadmin用户的逃生能力。当前取值仅支持sysadmin，monadmin或者空字符串。

参数类型：字符串

参数单位：无

取值范围：长度大于0的字符串。

该参数目前只支持三个取值：sysadmin，monadmin或""，这几个值的具体含义如下：

- sysadmin：控制sysadmin用户的作业是否会被该逃生功能进行cancel处理。
- monadmin：控制monadmin用户的作业是否会被该逃生功能进行cancel处理。
- ""：关闭sysadmin和monadmin用户的逃生能力。

默认值：""，表示关闭sysadmin和monadmin用户的逃生能力。

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。该参数可以同时设置多个值，以逗号分隔，例如resilience_escape_user_permissions = 'sysadmin,monadmin'，也可以只设置一个值，例如resilience_escape_user_permissions = 'monadmin'。

例如，如下命令表示同时开启sysadmin和monadmin用户的逃生功能。

```
gs_guc reload -Z coordinator -Z datanode -N all -I all -c
"resilience_escape_user_permissions='sysadmin,monadmin'"
```

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 若该参数多次设置，以最新的设置生效。
- 该参数设置为取值范围中的任意值，普通用户都支持该逃生功能。
- 当用户同时具有sysadmin和monadmin时，resilience_escape_user_permissions必须设置为"sysadmin,monadmin"才能触发该用户的逃生功能。

版权所有 © 华为云计算技术有限公司

start_with_max_mem

参数说明：对于数据量较大的start with查询语句，可能会由于递归深度过大导致占用内存过大，该参数用于控制start with算子占用的最大内存大小。

参数类型：整型

参数单位：kB

取值范围：-1 ~ 2147483647

默认值：-1，表示start with算子占用内存无限制。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示start_with_max_mem为1024kB；带单位取值1MB，表示start_with_max_mem为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：依据查询数据量来确定，默认-1，如果查询数据量较大，可能导致出现内存占用过大的问题。可按需调整该参数，避免单个查询占用内存过大，具体可参考**work_mem**设置。

设置不当的风险与影响：如果start_with_max_mem设置非默认值，或设置的值过小，会导致查询语句执行失败。

6.3.4.2 磁盘空间

介绍与磁盘空间相关的参数，用于限制临时文件所占用的磁盘空间。

sql_use_spacelimit

参数说明：限制单个SQL在单个DN上，触发写盘操作时，所有类型写盘文件的总空间大小，管控的空间包括普通表、临时表以及中间结果集写盘占用的空间。

参数类型：整型

参数单位：kB

取值范围：-1 ~ 2147483647，其中-1表示没有限制。

默认值：-1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值如果要带单位，必须为kB、MB、GB，不能为TB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

temp_file_limit

参数说明：指定一个会话中，能用于临时文件的最大磁盘空间。例如：一个连接会话中需要进行排序或者使用外存哈希表的临时文件，以及使用游标功能占用的临时文件。

版权所有 © 华为云计算技术有限公司

说明

- SQL查询执行时使用的临时表空间不在此限制。
- 在SMP场景（设置**query_dop** >= 2）下，前台会话中的SQL会被拆分为若干个（个数等于query_dop参数设置的值）后台线程执行。在任意一个线程产生的临时文件达到阈值后，才会触发前台会话中的SQL报错，此时可能出现临时文件大小大于前台会话中设置的本参数值的情况（此场景下单个前台会话占用的临时文件大小峰值不会超过 temp_file_limit * query_dop）。
- 在磁盘使用率超过只读阈值**datastorage_threshold_value_check**后，并且default_transaction_read_only = on的情况下，该参数会被CM接管，设置为0，此时修改temp_file_limit将无法生效。磁盘使用率恢复后，CM会将该参数恢复为节点只读前用户最后一次设置的值。此外，当磁盘使用率超过只读阈值**datastorage_threshold_value_check**且default_transaction_read_only = on时，若用户通过清理磁盘空间将磁盘使用率降低至只读阈值**datastorage_threshold_value_check**以下，并手动设置参数default_transaction_read_only = off，此种恢复方式视为绕过CM监管，用户需手动同步修改temp_file_limit值。
- 该参数设置默认值-1的情况下，磁盘使用率超过只读阈值**datastorage_threshold_value_check**后，CM接管参数生效到kill存量业务期间，业务仍可能持续落盘，可能会导致磁盘写满，此时可以删除数据目录下pgsql_tmp中生成的临时文件以降低磁盘使用率。

参数类型：整型

参数单位：kB

取值范围：-1 ~ 2147483647，其中-1表示没有限制。

默认值：-1

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值如果要带单位，必须为kB、MB、GB，不能为TB。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果设置过小会导致频繁的I/O操作，导致性能劣化；如果设置过大，可能会占用过多的磁盘空间。

6.3.4.3 内核资源使用

介绍与操作系统内核相关的参数，这些参数是否生效依赖于操作系统的设置。

max_files_per_process

参数说明：设置每个数据库线程允许同时打开的最大文件数量。

参数类型：整型

参数单位：无

取值范围：25 ~ 2147483647

默认值：1024

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果操作系统内核强制一个合理的数目，则不需要设置。但是在一些平台上（特别是大多数BSD系统），内核允许独立进程打开比系统真正可以支持的数目大得多的文件数。如果用户发现有“Too many open files”失败现象，请尝试缩小这个

版权所有 © 华为云计算技术有限公司

设置。通常情况下需要满足，系统FD（file descriptor）数量>=最大并发数*当前服务器主DN个数*max_files_per_process*3。

设置不当的风险与影响：请根据实际使用需求进行设置。如果设置过小，有效安全文件句柄可能较少，从而导致集群状态异常。

shared_preload_libraries

参数说明：此参数指定数据库启动时需预加载的共享库（支持逗号分隔多个库名）。例如，'$libdir/mylib' 会优先加载mylib.so（部分平台为mylib.sl）。仅sysadmin用户可修改此参数。

说明

- 如果被声明的库不存在，GaussDB服务将会启动失败。
- 每一个支持GaussDB的库都有一个特殊的标记用于保证兼容性。因此，不支持GaussDB的库不能用这种方法加载。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值："platform_compatibility_b,security_plugin"

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：可以用这个方法预先加载GaussDB的存储过程库，通常是使用'$libdir/plXXX'语法。XXX只能是pgsql、perl、tcl、python之一。

通过预先加载一个共享库并在需要的时候初始化，可以避免第一次使用这个库的加载时间。但是启动每个服务器进程的时间可能会增加，即使进程从来没有使用过这些库。因此建议，对将被大多数会话使用的库才使用这个选项。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.4.4 基于开销的清理延迟

该特性旨在降低VACUUM和ANALYZE等维护操作在高并发数据库中的I/O影响。通过基于开销的清理延迟机制，允许管理员平衡维护任务与正常业务操作的资源分配。该特性的主要作用如下：

- 非关键延迟：维护操作无需即时完成，可适应后台执行。
- 资源隔离：避免维护任务过度占用I/O资源，保障业务查询性能。

须知

有些清理操作会持有关键的锁，这些操作应该尽快结束并释放锁。所以GaussDB的机制是，在这类操作过程中，基于开销的清理延迟不会发生作用。为了避免在这种情况下的长延时，实际的开销限制取下面两者之间的较大值：

- vacuum_cost_delay*accumulated_balance/vacuum_cost_limit
- vacuum_cost_delay*4

版权所有 © 华为云计算技术有限公司

背景信息

在**ANALYZE | ANALYSE**和**VACUUM**语句执行过程中，系统维护一个内部的计数器，跟踪所执行的各种I/O操作的近似开销。当累积的I/O开销达到vacuum_cost_limit设定的阈值时，执行维护操作的线程将休眠vacuum_cost_delay指定的时间。休眠结束后，计数器重置，操作继续执行。该特性默认关闭，需将vacuum_cost_delay设置为非零值以启用。

vacuum_cost_delay

参数说明：vacuum拉起后台服务线程对于数据表死亡元组进行清理，它会和业务线程冲突，为了能够控制其对于业务的影响，设置该参数vacuum_cost_delay表示当vacuum执行的开销达到**vacuum_cost_limit**的值时，vacuum线程就休眠vacuum_cost_delay的时间把资源让给业务线程执行，来确保业务不会被后台vacuum一直阻塞。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 100。0表示关闭基于开销的清理延迟特性，正数值表示打开基于开销的清理延迟特性。在许多系统上，休眠的有效分辨率是10ms，因此把vacuum_cost_delay设置为一个不是10的整数倍数值与设置为下一个10的整数倍数值作用相同。

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数一般设置较小，常见的设置是10ms或20ms。调整此特性资源占用率时，建议调整其他参数，而不是此参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

vacuum_cost_page_hit

参数说明：设置vacuum清理一个缓冲区命中的数据页面的预估代价。

参数类型：整型

参数单位：无

取值范围：0 ~ 10000

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

vacuum_cost_page_miss

参数说明：设置vacuum清理一个缓冲区中未命中的数据页面的预估代价。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 10000

默认值：10

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

vacuum_cost_page_dirty

参数说明：vacuum清理页面会产生脏页，当清理的数据页面原先是干净页面时，由于vacuum清理会产生一个需要刷脏到磁盘的额外代价，通过设置该参数，来表示vacuum清理该类数据页面的代价。

参数类型：整型

参数单位：无

取值范围：0 ~ 10000

默认值：20

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

vacuum_cost_limit

参数说明：设置清理线程休眠的开销限制。当vacuum线程执行的代价达到vacuum_cost_limit时，该线程休眠一次，休眠**vacuum_cost_delay**时间。

参数类型：整型

参数单位：无

取值范围：1 ~ 10000

默认值：1000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。该值越大，Vacuum 的I/O频次限制越小，越不容易进入休眠状态，Vacuum效率越高，对业务I/O的影响就越大。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

vacuum_defer_cleanup_age

参数说明：通过设置该参数，延迟VACUUM操作清理无效行存表记录的时间，延迟等待的时间是以已经执行的事务数量来量化，通过设置vacuum_defer_cleanup_age的数值来控制延迟事务的数量。即VACUUM和VACUUM FULL操作不会立即清理被删除的元组，而是等到该元组已经被一定数量的事务处理后才执行清理和删除。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 1000000，值为0表示不延迟。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致VACUUM和VACUUM FULL不及时。

6.3.4.5 后端写线程

本章节介绍后端写（background writer）线程的参数配置。后端写线程的功能是将共享缓冲区中的脏数据（指共享缓冲区中新增或修改的内容）写入磁盘。其设计目的是让数据库进程在执行用户查询时，尽可能减少或避免因写操作而产生的等待（写操作由后端写线程异步完成）。

此机制还能有效减少检查点引起的性能波动。后端写线程会持续将脏页面刷新到磁盘，因此在检查点触发时，只需刷新少量页面即可。但该机制会增加整体I/O负载：在传统检查点间隔中，重复修改的页面可能仅刷新一次，而后端写线程可能在相同间隔内多次刷新。通常情况下，持续的低负荷运行优于周期性的尖峰负荷，本节讨论的参数可用于根据实际需求调节行为。

bgwriter_delay

参数说明：定期循环逐块扫描缓冲区刷脏的时间间隔。全量checkpoint模式下，根据**bgwriter_lru_maxpages**参数控制每次写的量，随后休眠bgwriter_delay毫秒后重复执行；增量checkpoint模式下，根据设定**candidate_buf_percent_target**计算目标空闲缓冲页面个数，当候选缓冲页面不足时，每隔bgwriter_delay毫秒扫描一次10GB的缓冲区页面，遇到脏页则刷盘后加入候选链表，干净页则直接加入候选链表。多数系统中，休眠延迟的有效分辨率是10毫秒。因此，设置一个不是10倍数的数值与把它设置为一个10的倍数是一样的效果。

参数类型：整型

参数单位：ms（毫秒）

取值范围：10 ~ 10000

默认值：2000（即2s）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 增量checkpoint模式：数据量是**shared_buffers**多倍时，bgwriter_delay设置值不宜大于2s；数据量比shared_buffers小时，如果为了节省I/O，bgwriter_delay可以适当调大。
- 全量checkpoint模式：保持默认值即可，但是磁盘能力差的，可以适当调大。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

candidate_buf_percent_target

参数说明：当增量检查点打开时，设置干净页面候选链表占整体缓冲区页面数量百分比的期望值。若当干净页面链表中的页面比例小于该值，pagewriter线程会启动，扫描缓冲区页面将干净的页面放入链表，把脏页刷盘后再放入链表。

版权所有 © 华为云计算技术有限公司

参数类型：浮点型

参数单位：无

取值范围：0.1 ~ 0.85

默认值：0.3

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数直接影响pagewriter线程刷页个数。当数据量大于**shared_buffers**时，candidate_buf_percent_target设置过小，会影响性能。若数据量是shared_buffers的2倍及以上，该值不宜小于默认值；其他场景下，若要降低I/O，可适当调小该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bgwriter_lru_maxpages

参数说明：设置后端写线程每次脏页刷盘的脏页数量上限。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000。0表示禁用后端写功能，禁用后端写功能不会对checkpoints产生影响。

默认值：100

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bgwriter_lru_multiplier

参数说明：数据库内核根据当前缓冲区大小设置**shared_buffers**，在内核实际运行业务时发现当前设置的shared_buffers过小或过大，通过调整bgwriter_lru_multiplier参数给出最优推荐值，推荐值为该参数和shared_buffers的乘积。设置为推荐值之后，在数据库重新启动时生效。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 10

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 写“脏”缓存区到磁盘的数目取决于服务器最近几次使用的缓存区数目。最近使用buffers数目的平均值乘以bgwriter_lru_multiplier是为了评估下次服务器进程需要使用的buffers数目。在有足够多的干净的、可用的缓存区之前，后端写线程会一直写“脏”缓存区（每次写的缓存区数目不会超过**bgwriter_lru_maxpages**的值）。

版权所有 © 华为云计算技术有限公司

- 设置bgwriter_lru_multiplier为1.0表示一种“实时”策略，其作用是准确预测下次写“脏”缓冲区的数目。设置为较大的值可以应对突然的需求高峰，而较小的值则可以让服务器进程执行更多的写操作。
- 设置较小的bgwriter_lru_maxpages和bgwriter_lru_multiplier，会减小后端写线程导致的额外I/O开销，但是服务器进程会发出写操作，增加了对查询的响应时间。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

pagewriter_thread_num

参数说明：增量检查点打开后，设置后台刷页的线程数，将按照脏页置脏的先后顺序刷盘，产生干净的缓冲区页面，并推进用于打检查点的恢复点。

参数类型：整型

参数单位：无

取值范围：1 ~ 16

默认值：

- 混合部署形态的8核CPU/64G内存规格，默认值为1；
- 独立部署形态的所有规格、混合部署形态的其他规格，默认值为4。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dirty_page_percent_max

参数说明：增量检查点打开后，设置脏页数量占**shared_buffers**的最大期望百分比。达到这个设定值时，后台刷页线程将以设置的**max_io_capacity**刷脏。

参数类型：浮点型

参数单位：无

取值范围：0.1 ~ 1

默认值：0.9

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

pagewriter_sleep

参数说明：增量检查点模式下，设置pagewriter刷页线程的刷脏周期。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 3600000

版权所有 © 华为云计算技术有限公司

默认值：2000（即2s）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议CN保持为默认值。DN不宜设置大于2s，脏页产生很快的情况下，建议设置100ms - 500ms，因为设置过大会导致redo点推进过慢，影响Xlog回收。

设置不当的风险与影响：设置过大时，会导致刷页速度降低，redo点推进过慢，进而影响Xlog回收；同时，由于停止实例时需要等待该参数设置的时间，设置过大时可能导致进程退出超时。

max_io_capacity

参数说明：设置后端pagewriter刷页线程批量刷脏每秒的I/O上限。

参数类型：整型

参数单位：kB

取值范围：30720 ~ 10485760

默认值：512000（即500MB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不带单位，则默认为kB；取值如果要带单位，必须为kB、MB、GB。例如：max_io_capacity=512000、max_io_capacity=512000kB、max_io_capacity=500MB都表示设置max_io_capacity为500MB。

设置建议：根据具体业务场景和磁盘I/O能力进行设置。比如磁盘I/O带宽为500MB，建议设置带宽为500MB以内。要求更小的RTO时间或者在大容量场景，可以调大max_io_capacity，例如设置为2GB。

设置不当的风险与影响：设置过小时，刷脏速度慢于业务产生脏页的速度，会导致缓冲区脏页满，从而导致性能劣化。设置过大时，会导致写放大，影响存储设备寿命。

enable_consider_usecount

参数说明：设置缓冲区的页面淘汰算法是否考虑页面冷热程度。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示考虑页面冷热程度。
- off：表示不考虑页面冷热程度。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议大容量场景等频繁执行单页面淘汰的场景开启此参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_buffer_evictor

参数说明：智能淘汰算法开关。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：开启智能淘汰算法。
- off：关闭智能淘汰算法。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议大容量场景等频繁执行单页面淘汰的场景开启此参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

eviction_clean_page_percent_min

参数说明：智能淘汰算法保留的最少干净页数量。干净页少于此比例，限制淘汰候选页面的产生速率，设置为0不限制。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 0.9

默认值：0.1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议大容量场景等频繁执行单页面淘汰的场景开启此参数。

设置不当的风险与影响：如果参数设置过大，可能会导致刷脏性能下降。

eviction_candidate_list_size

参数说明：智能淘汰算法的淘汰候选队列长度。

参数类型：整型

参数单位：无

取值范围：1024 ~ 65536

默认值：2048

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议大容量场景等频繁执行单页面淘汰的场景开启此参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

pagewriter_flush_mode

参数说明：极致刷脏功能的开关。

参数类型：枚举类型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- extre_mode：表示开启极致刷脏功能。
- default_mode：表示关闭极致刷脏功能。

默认值：default_mode

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议大容量场景等频繁执行单页面淘汰的场景开启此参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dw_file_num

参数说明：设置批量双写文件的数量。

参数类型：整型

参数单位：无

取值范围：1 ~ 16

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数值与**pagewriter_thread_num**有关，不会大于pagewriter_thread_num。如果dw_file_num设置过大，内部会纠正为pagewriter_thread_num大小。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dw_file_size

参数说明：设置每个批量双写文件的大小。

参数类型：整型

参数单位：MB

取值范围：32 ~ 256

默认值：256

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

postmaster_parallel_init_thread_num

参数说明：设置数据库启动时可使用的并行线程数量。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：1 ~ 256

默认值：16

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该数值表示并行初始化的线程数量，因此建议设置值不超过（环境CPU核数量-1）。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.4.6 异步 I/O

checkpoint_flush_after

参数说明：设置checkpointer线程刷脏页个数超过设定的阈值时，告知操作系统将文件缓存中的数据页面异步刷盘。GaussDB中，磁盘页大小为8kB。

参数类型：整型

参数单位：页（8kB）

取值范围：0 ~ 256（0表示关闭异步刷盘功能）。

默认值：256kB（即32个页面）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。可以设置页面的数量或字节的大小，例如，取值32或256kB，表示checkpointer线程连续写32个磁盘页，即32*8=256kB磁盘空间后会进行异步刷盘。gs_guc不支持以页面数量为参数单位进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bgwriter_flush_after

参数说明：设置background writer线程刷脏页个数超过设定的阈值时，告知操作系统将文件缓存中的数据页面异步刷盘。GaussDB中，磁盘页大小为8kB。

参数类型：整型

参数单位：页（8kB）

取值范围：0 ~ 256（0表示关闭异步刷盘功能）。

默认值：512kB（即64个页面）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。可以设置页面的数量或字节的大小，例如，取值64或512kB，表示background writer线程连续写64个磁盘页，即64*8=512kB磁盘空间后会进行异步刷盘。gs_guc不支持以页面数量为参数单位进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

backend_flush_after

参数说明：设置backend线程刷脏页个数超过设定的阈值时，告知操作系统将文件缓存中的数据页面异步刷盘。GaussDB中，磁盘页大小为8kB。

参数类型：整型

参数单位：页（8kB）

取值范围：0 ~ 256（0表示关闭异步刷盘功能）。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。可以设置页面的数量或字节的大小，例如，取值64或512kB，表示backend线程连续写64个磁盘页，即64*8=512kB磁盘空间后会进行异步刷盘。gs_guc不支持以页面数量为参数单位进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_adio_function

参数说明：设置ADIO模式是否开启。关于ADIO模式的详细说明。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启ADIO模式。
- off：表示关闭ADIO模式。

默认值：off

设置方式：该参数属于SIGHUP类型参数。支持通过**表6-2**中的“方式二”设置方法将该参数从off切换为on。设置完成后，可通过系统函数**gs_get_io_type()**获取ADIO模式是否切换成功。如需将该参数从on切换为off，需要重启实例，参考**表6-2**中的“方式一”。

设置建议：DBE_PERF.global_candidate_status系统视图的candidate_slots列在短时间内多次出现小于缓冲区页面15%时，建议将该参数设置为on。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

该开关打开时，若/proc/sys/fs/aio-max-nr配置较低，可能导致AIO Completer线程拉起失败退出，建议修改/proc/sys/fs/aio-max-nr值大于节点最大实例数量 * AIO线程的最大数量（当前版本为5）* 最大AIO完成队列深度（当前版本为10000）+ 5000。

effective_io_concurrency

参数说明：设置磁盘子系统同时有效处理的请求数。对于RAID阵列，此参数是阵列中驱动器主轴的数量。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：0 ~ 1000

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置值和磁盘数量相同。

设置不当的风险与影响：设置过大，运行到相关逻辑时可能会占用整个磁盘的I/O资源，出现I/O peak。

6.3.5 数据并行导入导出

GaussDB提供了并行导入导出功能，以快速、高效地完成大量数据导入导出。介绍GaussDB并行导入导出的相关参数。

raise_errors_if_no_files

参数说明：设置导入时是否区分“导入文件记录数为空”和“导入文件不存在”。该参数开启时，“导入文件不存在”的时候，GaussDB将抛出“文件不存在”的错误。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示导入时区分“导入文件记录数为空”和“导入文件不存在”。
- off：表示导入时不区分“导入文件记录数为空”和“导入文件不存在”。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

raise_errors_if_no_files参数只在分布式GDS工具使用场景有意义。

gds_debug_mod

参数说明：为了增强对Gauss Data Service（以下简称GDS）相关问题的分析定位能力，可以通过此参数选择是否开启GDS的debug功能。参数开启后，将在集群节点对应的日志中输出GDS每次收发的包裹类型、命令交互的对端以及其他交互相关的细节信息，方便记录Gaussdb端状态机的状态跳转，以及目前所处的状态信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启GDS的debug功能。
- off：表示不开启GDS的debug功能。

版权所有 © 华为云计算技术有限公司

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议仅在定位GDS问题时开启。

设置不当的风险与影响：此参数打开会输出额外日志，增加日志I/O开销，进而影响性能和日志的信息有效性。

safe_data_path

参数说明：设置copy、load data、本地节点类型的外表以及非初始用户的高级包进行文件读写操作时的路径限制。对相关路径实行白名单管理，仅允许对白名单路径下的文件进行读写操作。

参数类型：字符串

参数单位：无

取值范围：合法目录路径，需要小于4096个字符。

默认值："copydir_default,copydir_logical_decode"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：尽量控制safe_data_path路径覆盖有限范围，不要设置过大的路径范围。

设置不当的风险与影响：将safe_data_path设置为空或将safe_data_path路径范围设置过大时，可能导致恶意用户访问到服务器上的敏感文件。

说明

- 如果safe_data_path目录下存在软链接文件，则会按软链接实际指向的文件路径进行处理，实际路径如果不在safe_data_path下会报错处理。
- 如果safe_data_path目录下存在硬链接文件，则可以正常使用。为安全起见，请谨慎使用硬链接文件，切勿在safe_data_path目录下创建指向目录以外的硬链接文件，并确保safe_data_path目录权限最小化。
- 将safe_data_path参数设置为空，表示不开启白名单校验，用作前向兼容，请在评估风险和影响后谨慎进行设置。
- safe_data_path参数取值中，多个不同的路径间需要使用“,”进行分隔，且分隔符间不包含空格和制表符等空白字符。
- 参数默认取值中的copydir_default为占位的表示方式，表示白名单中支持使用的默认路径；实际表示的路径为$GAUSSLOG/gs_copy_data；当路径不存在时，可以通过手动执行一次“COPY table_name FROM 'file_name'”或“COPY table_name TO 'file_name'”命令进行创建，或者在对应目录下通过mkdir命令进行创建；在开启白名单功能时，不推荐用户将此参数进行删除。
- 参数默认取值中的copydir_logical_decode为逻辑解码中需要使用到的路径，禁止将其删除，否则可能会对相关功能造成影响。
- 修改safe_data_path参数时，请以“,”分隔的方式对参数取值进行追加写，禁止对默认值中的路径进行删除。

enable_copy_server_files

参数说明：是否开启copy服务器端文件的权限。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示开启copy服务端文件的权限，允许具有SYSADMIN权限的用户或继承了内置角色gs_role_copy_files权限的用户执行COPY FROM FILENAME或COPY TO FILENAME命令。
- off：表示不开启copy服务端文件的权限，只允许初始用户执行COPY FROM FILENAME或COPY TO FILENAME命令。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：仅在需要copy服务端文件时打开，打开后建议同步设置**safe_data_path**。

设置不当的风险与影响：打开该参数后，如果**safe_data_path**设置路径范围过大，可能导致恶意用户访问到服务器上的敏感文件。

support_binary_copy_version

参数说明：使用COPY TO的BINARY模式导出数据时，是否包含当前数据库服务端的编码信息。

参数类型：字符串

参数单位：无

取值范围：""、"header_encoding"。

表 6-14 兼容性配置项

兼容性配置项	兼容性行为控制
"header_encoding"	使用COPY TO的BINARY模式导出数据时，二进制文件头中包含当前数据库服务端的编码信息。
""	做前向兼容处理时配置，以原二进制格式进行导出。

默认值："header_encoding"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：无特殊情况时建议配置为默认取值，仅做前向兼容时需另行配置为空。

设置不当的风险与影响：设置为空字符串时，导出结果中不包含服务端编码信息，在需要编码信息的场景下，需要用户自行查询并记录。

copy_special_character_version

参数说明：控制使用COPY导入导出数据时对非法编码的字符处理的相关表现。

参数类型：字符串

参数单位：无

取值范围：""、"no_error"、"per_byte"。

版权所有 © 华为云计算技术有限公司

表 6-15 兼容性配置项

兼容性配置项	兼容性行为控制
"no_error"	使用COPY导入数据文件时，当文件编码与服务端编码相同时，对数据文件中不满足编码信息的数据进行容错处理，不报错，并直接按照原编码插入表中。
"per_byte"	使用COPY导出文本格式的数据文件时，对文件编码为GBK或ZHS16GBK文件的处理方式。 设置"per_byte"后，在导出数据时一次导出一个字节数据，否则一次导出两个字节数据（GBK编码下，一个字符占两个字节）。
""	参数默认配置，不影响任何功能使用，做前向兼容处理。即COPY存在非法编码时报错。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。使用gsql连接数据库，使用set方式设置时不区分大小写，使用gs_guc设置时仅能设置小写。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 为保证导入表中数据的合法性，COPY中需要对数据编码的合法性进行校验。当开启此参数后，会屏蔽非法校验操作并导致字段内出现非法字符，请酌情考虑开启此参数。
- 当前屏蔽数据编码校验仅在服务端编码与数据编码保持一致时生效，即copy_special_character_version值为"no_error"时，数据库服务端编码须与数据文件编码保持一致，否则会报错；未指定数据编码时，数据编码默认为客户端编码。
- 二进制模式下，copy_special_character_version值为"no_error"时，仅对TEXT、CHAR、VARCHAR、NVARCHAR2、CLOB类型的字段生效。
- 此参数仅在UTF-8、GB18030、GB18030_2022、ZHS16GBK、LATIN1字符集数据库下使用生效。
- 客户端编码与服务端编码都为GBK或ZHS16GBK时，当数据库内存在非法编码的数据时，如果不设置copy_special_character_version值为"per_byte"，导出的数据文件中可能会存在预期外的数据。
- 设置copy_special_character_version值为"no_error"的优先级高于COPY内COMPATIBLE_ILLEGAL_CHARS参数。

enable_log_copy_illegal_chars

参数说明：使用COPY导入导出数据时，遇到非法编码的字符时，是否向数据库运行日志中写入记录。

参数类型：布尔型

参数单位：无

取值范围：on、off。

版权所有 © 华为云计算技术有限公司

表 6-16 兼容性配置项

兼容性配置项	兼容性行为控制
on	每出现一行包括非法编码字符的数据，都会向数据库运行日志中写入一条记录。
off	不向数据库运行日志中写入非法编码字符相关记录。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：无特殊情况时建议配置为默认值，仅对数据库性能及磁盘带宽有要求时设置为off。

设置不当的风险与影响：设置为off时将不会在运行日志中写入非法编码字符相关记录，影响可定位能力。

batch_insert_index_types

参数说明：在特定场景下执行批量插入时，是否开启批量插入索引性能优化。

参数类型：字符串

参数单位：无

取值范围："rcr_ubtree"、""。

表 6-17 兼容性配置项

兼容性配置项	兼容性行为控制
"rcr_ubtree"	rcr_ubtree类别的索引在批量插入时开启批量插入索引特性。
""	任何类别的索引不开启批量插入索引特性。

默认值：""。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议配置为默认值，如果出现性能大幅下降则将本项目设为""。

版权所有 © 华为云计算技术有限公司

说明

当以下任何一个场景满足所有的约束条件时，打开GUC参数可以提升批量插入索引性能：

场景1：使用JDBC连接数据库，通过java的批量插入数据接口executeBatch进行批量插入索引；

场景2：执行语句insert into t1 select * from t2，其中t1和t2表schema相同，t1表包含索引，此语句支持在t1表上批量插入索引；必须要开启stream计划（enable_fast_query_shipping参数被设置成off），并在批量插入表数据的前提（使用explain verbose命令查看）下才会批量插入索引；

场景3：执行语句copy t1 from 'file.csv' delimiter ',' csv header，其中t1和file.csv文件中的schema相同，t1表包含索引，此语句支持在t1表上批量插入索引。

约束1：插入的索引类型为ubtree RCR索引；

约束2：插入数据不递增（这种情况下原来的机制会利用缓存，无性能提升）；

约束3：enable_stream_operator参数被设置成on；

约束4：不是SMP的批量索引插入。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

a_format_enable_copy_empty_lobs

参数说明：在Oracle兼容模式下，执行COPY FROM批量插入时，是否允许通过空串输入向BLOB及CLOB类型中插入空字符串。

参数类型：布尔型

参数单位：无

取值范围：on、off。

表 6-18 兼容性配置项

兼容性配置项	兼容性行为控制
on	Oracle兼容模式下，允许通过空串输入向CLOB及BLOB类型中插入空串。 比如将1,"",3插入到表中后，第二个字段的结果为空，但是无法通过where col2 is null进行过滤，且字段的长度为0。
off	Oracle兼容模式下，允许通过空串输入向CLOB及BLOB类型中插入空串。空串会被转换为NULL进行存储。 比如将1,"",3插入到表中后，第二个字段的结果为NULL，可以通过where col2 is null进行过滤。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：建议配置为默认值，如果需要前向兼容再设置为off。

版权所有 © 华为云计算技术有限公司

说明

- 关于空串输入与空值输入：

  在COPY的CSV格式下，1,,3表示空值输入，1,"",3表示空串输入。参数开启时，前者的第二个字段会被插入为NULL，后者的第二个字段会被插入为空串；参数关闭时，前者和后者的第二个字段都会被插入为NULL。
- 对于新安装部署的数据库，参数默认取值为on；对于升级上来的数据库，出于前向兼容考虑，参数默认取值为off。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_copy_case_sensitive

参数说明：使用gs_loader导入数据及COPY...TRANSFORM()导入数据需要表达式处理时，遇到大写字符列或包含特殊的列名时，是否保留列名的大小写敏感。

参数类型：布尔型

参数单位：无

取值范围：on、off。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议配置为默认值，在表列中存在大写列名、特殊字符列名且需要在导入过程使用表达式处理相关列时，设置为on。

说明

- 在gs_loader控制文件中，对大写字符列、包含特殊字符的列进行表达式处理，会默认转为小写匹配表中的列，若存在大写列名、特殊字符列名，通过开启参数enable_copy_case_sensitive=on后，在控制文件中通过双引号""转义标识保留列名，参考示例如下：

  ```
  LOAD DATA
  CHARACTERSET UTF8
  truncate into table "IMP_COLNAME_TEST"
  FIELDS TERMINATED BY ","
  (
  id integer external,
  name char(32),
  "Column_Mixed" "TRIM(:""Column_Mixed"")",
  "Col With Space" "TRIM(:""Col With Space"")",
  "COL_UPPER" "TRIM(:""COL_UPPER"")",
  "Col""With""Quote" "TRIM(:""Col""""With""""Quote"")",
  )
  ```
- 在COPY...TRANSFORM()导入数据需要表达式处理时，使用单引号'标识表达式中的列名，参考示例如下：

  ```
  COPY "IMP_COLNAME_TEST" ( id, name, "Column_Mixed","Col With Space", "COL_UPPER", "Col""With""Quote", con, col_lower )
  FROM 'case.csv'
  TRANSFORM("Column_Mixed" AS TRIM(:"Column_Mixed"),"Col With Space" AS TRIM(:"Col With Space"),"COL_UPPER" AS TRIM(:"COL_UPPER"),"Col""With""Quote" AS TRIM(:"Col""With""Quote")) ;
  ```


版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：默认off时对表达式中的列名大小写不敏感，统一转义为小写处理，若实际列名为大写字段可能出现导入报错。设置为on时，则以表达式中的列名执行表达式处理后，进行导入操作。

6.3.6 预写式日志

6.3.6.1 设置

wal_level

参数说明：设置写入WAL信息量的级别，不能为空或被注释掉。

参数类型：枚举类型

参数单位：无

取值范围：

- minimal

  优点：一些重要操作（包括创建表、创建索引、簇操作和表的复制）都能安全跳过，这样就可以使操作变得更快。

  缺点：WAL仅提供从数据库服务器崩溃或者紧急关闭状态恢复时所需要的基本信息，无法用WAL归档日志恢复数据。

- archive

  这个参数级别增加了WAL归档需要的日志信息，从而可以支持数据库的归档恢复。

- hot_standby
  - 这个参数级别进一步增加了在备机上运行的SQL查询的信息，这个参数只能在数据库服务重新启动后生效。
  - 为了在备机上开启只读查询，wal_level的级别必须在主机上设置成大于或者等于hot_standby级别，并且备机必须打开**hot_standby**参数。hot_standby和archive级别之间的性能差异很小。

- logical

  设置为logical后才可以进行逻辑日志解析，设置后Xlog日志中会额外记录主键信息。

默认值：hot_standby

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：开启逻辑复制功能时设置为logical，其他场景设置为hot_standby。参数级别满足优先级logical > hot_standby > archive > minimal。

版权所有 © 华为云计算技术有限公司

须知

- 如果需要启用WAL日志归档和主备机的数据流复制，必须将此参数设置为archive、hot_standby或者logical。
- 如果该参数设置为archive或minimal，则**hot_standby**参数必须设置为off。在分布式环境中，因为会对双机其他一些功能产生影响，hot_standby参数不能设置成off。因此不建议该参数设置为archive或minimal，否则数据库将无法启动。

设置不当的风险与影响：当备机读开启时，设置级别小于hot_standby，会导致节点无法启动。

fsync

参数说明：设置GaussDB服务器是否使用fsync()系统函数确保数据的更新及时写入物理磁盘中（fsync()系统函数实现的种类，请参见参数**wal_sync_method**）。

须知

- 使用fsync()系统函数可以保证在操作系统或者硬件崩溃的情况下将数据恢复到一个已知的状态。
- 如果将此参数关闭，可能会在系统崩溃时无法恢复原来的数据，导致数据库不可用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用fsync()系统函数。
- off：表示不使用fsync()系统函数。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

wal_sync_method

参数说明：参数**fsync**开启的前提情况下，设置向磁盘强制更新WAL数据的方法。

须知

如果参数**fsync**关闭，这个参数的设置就没有意义，因为所有数据更新都不会强制写入磁盘。

参数类型：枚举类型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- open_datasync：表示用带O_DSYNC选项的open()打开WAL文件。
- fdatasync：表示每次提交的时候都调用fdatasync()（支持suse10和suse11）。
- fsync_writethrough：表示每次提交的时候调用fsync()强制把缓冲区任何数据写入磁盘。
- fsync：表示每次提交的时候调用fsync()（支持suse10和suse11）。
- open_sync：表示用带O_SYNC选项的open()写WAL文件（支持suse10和suse11）。

说明

- 不是所有的平台都支持以上参数。
- 由于历史原因，Windows平台支持将wal_sync_method设置为fsync_writethrough。在windows平台上fsync_writethrough和fsync等效。

默认值：fdatasync

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

synchronous_commit

参数说明：设置当前事务的同步方式。

通常情况下，一个事务产生的日志同步顺序如下：

1. 主机将日志内容写入本地内存。
2. 主机将本地内存中的日志写入本地文件系统。
3. 主机将本地文件系统中的日志内容刷盘。
4. 主机将日志内容发送给备机。
5. 备机接收到日志内容，存入备机内存。
6. 备机将备机内存中的日志写入备机文件系统。
7. 备机将备机文件系统中的日志内容刷盘。
8. 备机回放日志，完成对数据文件的增量更新。

参数类型：枚举类型

参数单位：无

取值范围：

- on：表示主机事务提交需要等待备机将对应日志刷新到磁盘。
- off：表示主机事务提交无需等待主机自身将对应日志刷新到磁盘，通常也称为异步提交。
- local：表示主机事务提交需要等待主机自身将对应日志刷新到磁盘，通常也称为本地提交。
- remote_write：表示主机事务提交需要等待备机将对应日志写到文件系统（无需刷新到磁盘）。

版权所有 © 华为云计算技术有限公司

- remote_receive：表示主机事务提交需要等待备机接收到对应日志数据（无需写入文件系统）。
- remote_apply：表示主机事务提交需要等待备机完成对应日志的回放操作。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。synchronous_commit参数已被CM接管，正常情况是默认开启自动升降副本功能（即CM参数enable_az_auto_switchover为on），此时修改synchronous_commit参数不能生效，只支持多AZ场景。如要修改此参数，则需关闭自动升降副本功能（即CM参数enable_az_auto_switchover为off），此时修改synchronous_commit参数才能生效。

设置建议：不建议进行修改。如果手动修改该参数为非默认值，则存在丢失数据的风险，不具备商用能力。

设置不当的风险与影响：该参数由CM维护，如果手动修改，则存在丢失数据的风险。

full_page_writes

参数说明：设置GaussDB服务器在检查点之后对页面的第一次修改时，是否将每个磁盘页面的全部内容写到WAL日志中。当**enable_incremental_checkpoint**和**enable_double_write**同时打开时，则不开启full_page_writes。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用此特性。
- off：表示关闭此特性。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。启用此特性后，完整的页面写到磁盘，会在崩溃重启过程中，保证页面被正确还原进而确保数据完整，代价是增加了写入WAL日志的数据量。

设置不当的风险与影响：关闭此特性后，操作系统崩溃可能导致磁盘页面部分写入，从而无法恢复原始数据。若服务器硬件（如带电池供电的磁盘控制器）或文件系统特性（如Reiser4）能有效降低页面写入风险，且用户明确了解相关风险，可关闭此参数。

wal_log_hints

参数说明：在检查点之后，如果对页面的第一次修改是对页面上元组hint bits的修改时，是否将整个页面的全部内容写到WAL日志中。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示整个页面全部内容写到WAL日志中。

版权所有 © 华为云计算技术有限公司

- off：表示整个页面内容不会写到WAL日志中。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不推荐用户修改此设置。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

wal_buffers

参数说明：设置用于存放WAL数据的共享内存空间的大小，可以用XLOG_BLCKSZ数量表示，同时也可以用实际的大小表示。XLOG_BLCKSZ是Xlog块的大小，一般默认为8kB。

参数类型：整型

参数单位：8kB（XLOG_BLCKSZ）

取值范围：-1 ~ 262144（XLOG_BLCKSZ数量）

- 如果设置为-1，表示wal_buffers的大小随着参数**shared_buffers**自动调整。wal_buffers默认为shared_buffers的1/32，最小值为8个XLOG_BLCKSZ，最大值为2048个XLOG_BLCKSZ，自动调整后的值小于最小值时会被强制设置为最小值，大于最大值时会被强制设置为最大值。
- 如果设置为其他值，当取值小于4时（0、1、2、3），会被强制设置为4。

默认值：

- 独立部署：1GB（128核CPU/1024G内存，60核CPU/480G内存，32核CPU/256G内存）；512MB（16核CPU/128G内存）；256MB（8核CPU/64G内存）；128MB（4核CPU/32G内存）；64MB（4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，取值2048表示wal_buffers为2048 * 8kB；取值20480kB表示wal_buffers为20480kB。取值如果要带单位，必须为kB、MB、GB。如果取值带单位且不为8kB的整数倍，会自动向下取整为8kB的整数倍。

设置建议：每次事务提交时，WAL缓冲区的内容都写入到磁盘中，因此设置为很大的值不会带来明显的性能提升。如果将它设置成几百兆，在有很多即时事务提交的服务器上提高写入磁盘的性能。根据经验，默认值可以满足大多数的情况，对应不同规格，请按照对应建议的默认值设置，不建议修改。避免设置明显不合理（过大或过小）的wal_buffers值，应满足条件：data_replicate_buffer_size + segment_buffers + shared_buffers + wal_buffers + temp_buffers + maintenance_work_mem + work_mem + query_mem +（备机）wal_receiver_buffer_size < max_process_memory < 物理机内存大小。如果设置的内存参数数值过大，超过物理内存上限，将会导致数据库由于无法申请到足够的内存而启动失败的情况。

设置不当的风险与影响：默认值即推荐值，不建议修改。修改必须详细确认参数的规格限制，并考虑硬件资源是否足够，否则可能导致集群异常。

wal_writer_delay

参数说明：WalWriter线程的写间隔时间。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：ms（毫秒）

取值范围：1 ~ 10000

默认值：200

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：时间过长可能造成WAL缓冲区的内存不足；时间过短会引起WAL不断写入，增加磁盘I/O负担。

commit_delay

参数说明：表示一个已经提交的数据在WAL缓冲区中存放的时间。

须知

- 设置为非0值时事务执行commit后不会立即写入WAL中，而仍存放在WAL缓冲区中，等待WalWriter线程周期性写入磁盘。
- 如果系统负载很高，在延迟时间内，其他事务可能已经准备好提交。但如果没有事务准备提交，这个延迟可能会造成不必要的等待。

参数类型：整型

参数单位：us（微秒）

取值范围：0 ~ 100000，其中0表示无延迟。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：参考须知。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

commit_siblings

参数说明：当一个事务发出提交请求时，如果数据库中正在执行的事务数量大于此参数的值，则该事务将等待一段时间（**commit_delay**的值），否则该事务则直接写入WAL。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000

默认值：5

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

wal_block_size

参数说明：表示WAL日志段文件中日志页面的大小。

参数类型：整型

参数单位：Byte

取值范围：8192

默认值：8192

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

wal_segment_size

参数说明：表示WAL日志段文件的大小。可以用XLOG_BLCKSZ数量表示，同时也可以用实际的大小表示。XLOG_BLCKSZ是Xlog块的大小，一般默认为8kB。

参数类型：整型

参数单位：8kB

取值范围：2048（即16MB）

默认值：16MB（2048 * 8kB）

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

force_promote

参数说明：备机强切功能开关。

须知

备机强切在集群故障状态下，以丢失部分数据为代价换取集群尽可能快地恢复服务；是集群状态为不可用时的一个逃生方法，不建议频繁触发。如果操作者不清楚备机强切后丢失数据对业务的影响，请勿使用本功能。

参数类型：整型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- 0：表示关闭。
- 1：表示开启。

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。使用备机强切功能时，DN和cmserver都需要将此参数设置为开启，重启集群后生效。

设置建议：不推荐开启此参数。

设置不当的风险与影响：开启后，触发强切会导致丢失部分数据。

wal_file_init_num

参数说明：WAL writer辅助线程启动时一次创建Xlog段文件的数量。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000000

默认值：10

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，会占用过多的磁盘空间，占用量为wal_file_init_num * **wal_segment_size**。

wal_file_preinit_bounds

参数说明：业务运行时，WAL writer辅助线程每秒可预扩的WAL日志段文件的数量限制（WAL日志段文件的大小为16MB）。当该参数值为0时，表示不进行限制。

参数类型：整型

参数单位：无

取值范围：0 ~ 1024

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐设置为默认值，不做调整。当把该参数设置为有效值时，需基于磁盘I/O能力进行设置，可参考参数**max_io_capacity**的数值进行设置。

- 若max_io_capacity为500MB，可设置为25；
- 若max_io_capacity为1GB，可设置为50。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

walwriter_sleep_threshold

参数说明：xlogflusher进入休眠之前空闲Xlog刷新的次数，达到阈值会休眠。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：无

取值范围：1 ~ 50000

默认值：500

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

wal_flush_timeout

参数说明：遍历walInsertStatusTable的超时时间。Xlog刷盘自适应控制的刷盘I/O遍历walInsertStatusTable等待的最大时间。

参数类型：整型

参数单位：us（微秒）

取值范围：0 ~ 90000000

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置值过大可能造成Xlog刷盘频率降低，降低Xlog处理性能。

xlog_file_path

参数说明：双集群共享存储场景下，Xlog日志共享盘的路径。本参数在数据库系统初始化时由OM进行配置，不建议用户自行修改。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会进行设定，不建议用户修改此配置。

设置不当的风险与影响：设置不当会使数据库无法启动。

xlog_file_size

参数说明：双集群共享存储场景下，Xlog日志共享盘的大小。本参数在数据库系统初始化时由OM进行配置，不建议用户自行修改。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：Byte

取值范围：134217728 ~ 576460752303423487

默认值：549755813888

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：安装时会进行设定，不建议用户修改此配置。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

xlog_lock_file_path

参数说明：双集群共享存储场景下，Xlog日志共享盘抢占的锁文件的路径。本参数在数据库系统初始化时由OM进行配置，不建议用户自行修改。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会进行设定，不建议用户修改此配置。

设置不当的风险与影响：设置不当会使数据库无法启动。

max_size_for_shared_storage_xlog_write

参数说明：控制共享盘单次写入的最大数据量。

参数类型：整型

参数单位：kB

取值范围：8 ~ 131072

默认值：1024

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会进行设定，不建议用户修改此配置。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

shared_storage_write_mode

参数说明：用于控制共享存储（比如dorado存储设备）下主集群的最大Xlog可写入位置。

参数类型：枚举类型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- none：无限制。
- standby：集群内已连接备机的本地最大Xlog LSN + xlog_file_size*（双集群共享存储场景下，Xlog日志共享盘的大小）*。
- disaster_standby：灾备集群已连接备机的本地最大LSN + xlog_file_size。
- disaster_standby_strict：灾备集群备机的本地最大LSN + xlog_file_size（和disaster_standby的区别在于，即使主备集群之间数据面网络隔离，也不会覆盖写入）。
- standby_and_disaster_standby：集群内已连接备机的本地最大LSN和灾备集群已连接备机的本地最大LSN中的较小值+xlog_file_size。

默认值：disaster_standby

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：不建议用户修改此配置。

设置不当的风险与影响：修改该参数可能会导致备集群触发重建（build），不建议手动修改。

enable_shared_storage_optimization

参数说明：用于控制共享存储（比如dorado存储设备）下是否开启性能优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启共享存储下性能优化。
- off：表示关闭共享存储下性能优化。

默认值：

- 新安装的数据库：on。
- 从507.0之前版本升级的数据库：升级后默认值为off。
- 从507.0及之后版本升级的数据库：升级后的参数值保留原设置。

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：不建议用户修改此配置。

设置不当的风险与影响：关闭后将影响低并发下共享存储下的性能。

enable_wal_insert_record_single

参数说明：用于控制ARM架构服务器上的实例是否开启日志插入Single模式，x86架构的服务器上的实例不受影响。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示开启日志插入Single模式。
- off：表示关闭日志插入Single模式。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：

- 小规格或256C场景下建议开启，避免无效的调度开销。
- 高并发多核场景下建议关闭，避免大量CAS锁竞争。
- 256C/2048G出口参数默认为on。

设置不当的风险与影响：开启后将影响高并发多核使用场景下的性能。

6.3.6.2 检查点

checkpoint_segments

参数说明：设置**checkpoint_timeout**周期内所保留WAL日志文件的最大数量。每个日志文件大小为16MB。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483646

默认值：1024

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。提升此参数可加快大数据的导入速度，但需要结合**checkpoint_timeout**、**shared_buffers**这两个参数统一考虑。这个参数同时影响WAL日志文件复用数量，通常情况下pg_xlog文件夹下最大的复用文件数量为checkpoint_segments的2倍，复用的文件被改名为后续即将使用的WAL日志文件，不会被删除。

设置不当的风险与影响：该参数设置过小会导致频繁触发checkpoint；设置过大可能会增大RTO，且会导致WAL日志保留的文件数量增多，增大占用磁盘空间。

checkpoint_timeout

参数说明：设置自动检查点之间的最长时间。

参数类型：整型

参数单位：s（秒）

取值范围：30 ~ 3600

默认值：900（即15min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在提升**checkpoint_segments**以加快大数据导入的场景也需将此参数调大，同时这两个参数提升会加大**shared_buffers**的负担，需要综合考虑。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

checkpoint_completion_target

参数说明：预计每个检查点执行完毕的时间与**checkpoint_timeout**的比值。

参数类型：浮点型

参数单位：无

取值范围：0.0 ~ 1.0

默认值：0.5

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

checkpoint_warning

参数说明：由于写检查点文件导致检查点发生的时间间隔小于此参数，就写入建议增加**checkpoint_segments**参数值的警告信息。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647，其中0表示关闭警告。

默认值：300（5min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

checkpoint_wait_timeout

参数说明：设置请求检查点等待checkpointer线程启动的最长时间。

参数类型：整型

参数单位：s（秒）

取值范围：2 ~ 3600

默认值：60（1min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_incremental_checkpoint

参数说明：增量检查点开关。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启增量检查点开关。
- off：表示关闭增量检查点开关。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在小内存或磁盘性能较差的环境建议关闭，其他场景建议按默认值打开。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

当enable_incremental_checkpoint关闭后手动执行checkpoint，会使用full_page_writes生成第一条Xlog，触发备份恢复时可能会产生“xlog contains no data”字样的日志。

enable_double_write

参数说明：双写特性开关。**enable_incremental_checkpoint**增量检查点开关打开时，同时enable_double_write打开，则使用enable_double_write双写特性保护，不再使用**full_page_writes**防止半页写问题。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启双写开关。
- off：表示关闭双写开关。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在增量检查点模式下设置为on。

设置不当的风险与影响：在增量检查点开关打开时，关闭该参数可能导致数据出现半页写，进而导致持久化数据不可用。

incremental_checkpoint_timeout

参数说明：开启增量检查点开关（**enable_incremental_checkpoint**）之后，设置自动检查点之间的最长时间。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 3600

默认值：60

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_xlog_prune

参数说明：设置在任一备机断连时，主机是否根据Xlog日志的大小超过参数**max_size_for_xlog_prune**的值而回收日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：如果任一备机断连，主机回收日志。
- off：如果任一备机断连，主机不回收日志。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_size_for_xlog_prune

参数说明：控制触发Xlog回收时的强制回收量。

场景一：在备机故障时主机保留的Xlog最大数量。在**enable_xlog_prune**打开时生效，工作机制如下：

1. replconninfo系列GUC参数配置的所有备机都连接主机时，则该参数不生效。
2. replconninfo系列GUC参数配置的备机存在断连时，则该参数生效。当主机Xlog日志量大于该参数值，会强制回收。以下情况例外：

   在同步提交模式下（即**synchronous_commit**参数值不为local或off时），如果存在连接中的备机，则主机会考虑保留多数派备机中最小日志接受位置以后的日志，这种情况下，保留的日志可能多于max_size_for_xlog_prune参数值。
3. 若存在build中的备机，则该参数不生效，主机日志会全量保留，防止build操作期间由于日志被回收引发的失败。

场景二：在开启PITR归档功能时保留归档Xlog的最大数量，当归档Xlog长时间未上传导致本地保留的Xlog超过该参数大小时强制回收。此场景不要求**enable_xlog_prune**打开。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2199023255551

默认值：256GB

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不带单位，则默认为kB；取值如果要带单位，必须为kB、MB、GB。

版权所有 © 华为云计算技术有限公司

设置建议：根据磁盘可用空间大小设置，未开启PITR归档时建议最大设置值不超过256GB。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

- 若参数设置过小，可能导致备机连接中断，或在备机完成build重新连接主机时，因无法持续从主机获取日志而触发新的build重建流程。
- 在开启PITR归档功能时，该参数设置较小有可能触发Xlog强制回收，导致后续XBSA归档失败。
- 若参数设置过大，在备机长期断连时可能导致磁盘空间膨胀。

max_redo_log_size

参数说明：对于备DN，此参数表示最新日志回放点到最新检查点位置之间日志量的期望值；对于主DN，此参数表示最新日志插入点到最新检查点位置之间日志量的期望值。该参数用来参与控制检查点刷页速度。

参数类型：整型

参数单位：kB

取值范围：163840 ~ 2147483647

默认值：1048576

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。关注RTO的情况下，这个值建议不宜过大。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

checkpoint_lag_alarm_threshold

参数说明：控制“检查点推进恢复起始点的进度滞后”告警的触发阈值。

- 对于主实例，该阈值表示日志写入位置与恢复起始点之间的日志量差距。
- 对于备实例，该阈值表示日志回放位置与恢复起始点之间的日志量差距。

参数类型：整型

参数单位：GB

取值范围：-1 ~ 2000

- -1：表示按照CPU核数自动配置触发阈值（CPU核数<16时为80GB；16≤核数<128时为100GB；核数≥128时为128GB）。
- 0：表示关闭“检查点推进恢复起始点的进度滞后”告警。
- N（正整数）：表示触发阈值设置为N GB。

默认值：128（192核CPU/1536G内存，128核CPU/1024G内存）；100（104核CPU/1024G内存，96核CPU/1024G内存，96核CPU/768G内存，80核CPU/640G内存，80核CPU/512G内存，72核CPU/576G内存，64核CPU/512G内存，60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存）；80（8核CPU/64G内存，4核CPU/32G内存，4核CPU/16G内存）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.6.3 日志回放

recovery_time_target

参数说明：备机完成日志写入和回放的流控阈值。当备机的回放完成预期时间大于此参数，主机会触发日志流控，将降低主机向备机发送日志的速率。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

- 0：表示不开启日志流控。
- 1 ~ 3600：表示备机能够在recovery_time_target时间内完成日志的写入和回放，可以保证主机与备机切换时能够在recovery_time_target完成日志写入和回放，保证备机能够快速升主机。

默认值：60

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：一般设置为默认值，不作调整。

设置不当的风险与影响：设置过小会影响主机的性能，例如设置为10，说明备机回放相对备机接收日志有延迟，允许延迟最大为10s，如果超过了10s，则会限制主机往备机段发送日志，对主机的性能会有限制，在主备场景下，主机的事务需要有备机日志落盘才能提交，限制了日志的发送会限制主机的事务执行影响到性能。设置过大会失去流控效果。关闭流控后，可能会出现RTO/RPO升高的情况。

recovery_max_workers

参数说明：设置并行回放线程的最大数量。当设置值大于1时，为并行回放模式。

须知

- **recovery_parse_workers**和**recovery_redo_workers**同时为1，即不开启极致RTO，此参数才能生效。
- **recovery_parse_workers**或者**recovery_redo_workers**大于1，即开启极致RTO，此参数不生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 20

默认值：

- 混合部署形态的8核CPU/64G内存规格，默认值为1；
- 独立部署形态的所有规格、混合部署形态的其他规格，默认值为4。

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，会导致CPU、内存资源不足，造成回放速度降低。

recovery_parallelism

参数说明：查询并行回放线程的实际数量。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：4

设置方式：该参数属于POSTMASTER类型参数，但是不能直接修改。**recovery_max_workers**、**recovery_parse_workers**、**recovery_redo_workers**这3个参数的任意一个被修改时，recover_parallelism都将被重新计算。

设置建议：该参数为只读参数，无法修改。

设置不当的风险与影响：无

queue_item_size

参数说明：设定每个redo replay线程的任务队列最大长度。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：560

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，会占用较多的内存资源。

enable_parallel_reader

参数说明：开启极致RTO新框架特性的开关。

- 需要配合参数**recovery_readers**、**recovery_redo_workers**、**recovery_parse_workers**、**recovery_undo_workers**使用。
- 开启极致RTO新框架时，请确保参数**replication_type**取值为1。
- 开启极致RTO新框架说明请参见**表6-19**。

版权所有 © 华为云计算技术有限公司

表 6-19 开启极致 RTO 新框架说明

参数/模式	并行回放	极致RTO	极致RTO新框架
recovery_parse_workers	1	recovery_parse_workers>1 或者 recovery_redo_workers>1	recovery_parse_workers>1 或者 recovery_redo_workers>1
recovery_redo_workers	1		
enable_parallel_reader	off **说明** 此时若设置enable_parallel_reader为on，系统在内存中会自动将其修改为off，该参数不生效。	off	on

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启极致RTO新框架。
- off：表示关闭极致RTO新框架。

默认值：off

设置方式：该参数属于POSTMASTER类型参数请参见表6-1中对应设置方法进行设置。

设置建议：小于256U不建议开启。

设置不当的风险与影响：小于256U开启极致RTO新框架可能导致集群无法启动。

recovery_readers

参数说明：设置极致RTO新框架特性中readpage线程的数量。

- recovery_readers需要配合enable_parallel_reader、recovery_redo_workers、recovery_parse_workers、recovery_undo_workers使用。如不期望开启极致RTO新框架，请保持recovery_readers为1。
- enable_parallel_reader为on时且开启极致RTO时，recovery_readers需要大于1。若enable_parallel_reader为on、recovery_readers为1，且recovery_parse_workers或recovery_redo_workers大于1，系统会在内存中将enable_parallel_reader修改为false。

参数类型：整型

参数单位：无

取值范围：1 ~ 50

默认值：1

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不同CPU、内存和部署模式下，**recovery_readers**、**recovery_undo_workers**、**recovery_parse_workers**和**recovery_redo_workers**的设置值请参见**表6-21**。

设置不当的风险与影响：readpage线程自带解析Xlog的item，每增加一个readpage线程会增加对应的内存消耗。随着业务增加，item不足时会持续初始化item，导致内存增长。item数量存在上限，当达到上限后内存不再大幅度增长，但是item本身大小受到业务影响，超大事务等操作可能导致item本身内存膨胀，因此达到item上限后依然可能存在内存增长的情况。设置过大，会导致CPU、内存资源不足，造成回放速度降低或回放停滞。

极致RTO新框架对内存有要求，启动时会依据配置的max_process_memory和shared_buffer进行检测，计算该参数可以支撑的最大reader个数，若设置的recovery_readers大于该值，系统会在内存中将recovery_readers修改为计算后的值。

parallel_max_item_count

参数说明：设置极致RTO新框架特性中单个readpage线程可以使用的redo_item上限。该参数仅在极致RTO新框架下，即**enable_parallel_reader**为on时生效。

参数类型：整型

参数单位：无

取值范围：40000 ~ 400000

默认值：60000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议参照**表6-21**。

设置不当的风险与影响：readpage线程自带解析Xlog的item，每增加一个readpage线程会增加对应的内存消耗。随着业务增加item不足时会持续初始化item导致内存增长。item数量存在上限，当达到上限后内存不再大幅度增长，但是item本身大小受到业务影响，超大事务等操作可能导致item本身内存膨胀，故达到item上限后依然可能存在内存增长的情况。设置过大，会导致CPU、内存资源不足，造成回放性能下降。设置过小，会影响回放性能，不建议设置为100000以下的值，如因参数配置较小导致回放性能问题，需要根据机器实际配置调大此参数。

极致RTO新框架存在内存要求，启动时会依据配置的max_process_memory和shared_buffer进行检测，计算该参数可以支撑的最大reader个数，若设置的recovery_readers大于该reader，则会在内存中修改recovery_readers为计算后的reader。

recovery_undo_workers

参数说明：设置极致RTO新框架特性中undo回放线程的数量。

recovery_undo_workers需要配合**enable_parallel_reader**、**recovery_redo_workers**、**recovery_parse_workers**、**recovery_readers**使用。如果不期望开启极致RTO新框架，请保持recovery_undo_workers为1。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：1 ~ 50

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不同CPU、内存和部署模式下，**recovery_readers**、**recovery_undo_workers**、**recovery_parse_workers**和**recovery_redo_workers**的设置值请参见**表6-21**。

设置不当的风险与影响：设置过大，会导致CPU、内存资源不足，造成回放速度降低。

recovery_parse_workers

参数说明：设置极致RTO特性中ParseRedoRecord线程的数量。

- 需要配合**recovery_redo_workers**使用。若recovery_parse_workers或**recovery_redo_workers**取值大于1，则开启极致RTO。若不希望开启极致RTO，请保持recovery_parse_workers取值为1。
- 在开启极致RTO时确保参数**replication_type**取值为1。
- 若同时开启极致RTO和并行回放，则极致RTO特性生效，并行回放特性失效。

参数类型：整型

参数单位：无

取值范围：1 ~ 16

默认值：

- 独立部署：

  2（128核CPU/1024G内存，60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存）；1（8核CPU/64G内存，4核CPU/32G内存，4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不同CPU、内存和部署模式下，recovery_parse_workers和recovery_redo_workers的设置值可参考**表6-20**。

设置不当的风险与影响：设置过大，会导致CPU、内存资源不足，造成回放速度降低。

说明

- 打开极致RTO后，备机会额外启动的回放线程总数=recovery_parse_workers * (recovery_redo_workers + 2) + 5。回放线程越多，占用CPU、内存和I/O资源越多。请根据实际硬件配置合理设置参数，避免因为参数设置过大，导致CPU和内存占用过高，进而导致系统启动异常。混合部署场景可能对主机性能有影响。
- 在极致RTO新框架模式下，**recovery_parse_workers**取值范围为2~16，在该模式下若设置recovery_parse_workers为1，系统会自动在内存中将其设置为2。

版权所有 © 华为云计算技术有限公司

recovery_redo_workers

参数说明：设置极致RTO特性中每个ParseRedoRecord线程对应的PageRedoWorker数量。recovery_redo_workers需要配合**recovery_parse_workers**使用，recovery_parse_workers或者recovery_redo_workers大于1，则开启极致RTO。如不期望开启极致RTO，请保持recovery_redo_workers为默认值1。

参数类型：整型

参数单位：无

取值范围：

- 极致RTO：1~8
- 极致RTO新框架：1~50

说明

在极致RTO模式下，若设置的参数值大于8，系统会在内存中自动修正为上限值8。

默认值：

- 独立部署：

  8（128核CPU/1024G内存，60核CPU/480G内存，32核CPU/256G内存）；3（16核CPU/128G内存，8核CPU/64G内存）；1（4核CPU/32G内存，4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不同CPU、内存和部署模式下，recovery_parse_workers和recovery_redo_workers的设置值可参考**表6-20**、**表6-21**。

表 6-20 极致 RTO 框架下不同 CPU、内存和部署模式下的参数设置参考

编号	CPU个数	内存（GB)	是否混合部署	recovery_parse_workers	recovery_redo_workers	回放线程总数	备注
1	4	-	-	1	1	recovery_max_workers参数值	不推荐开极致RTO。
2	8	-	是	1	1	recovery_max_workers参数值	不推荐开极致RTO。
3	8	64	否	1	3	10	-
4	16	128	是	1	2	9	-
5	16	128	否	2	3	15	-
6	32	256	是	2	2	13	-
7	32	256	否	2	8	25	-

版权所有 © 华为云计算技术有限公司

编号	CPU个数	内存（GB)	是否混合部署	recovery_parse_workers	recovery_redo_workers	回放线程总数	备注
8	64	512	是	2	4	17	-
9	64	512	否	2	8	25	大于此硬件规格的系统均参考此参数设置值。
10	96	768	-	2	8	25	大于此硬件规格的系统均参考此参数设置值。

表 6-21 极致 RTO 新框架下不同 CPU、内存和部署模式下的参数设置参考

编号	CPU个数	内存（GB)	是否混合部署	recovery_readers	recovery_undo_workers	recovery_parse_workers	recovery_redo_workers	parall_max_item_count	回放线程总数	备注
1	<256	-	-	-	-	-	-	-	-	不推荐开极致RTO新框架。
2	256	2048	否	40	30	3	50	400000	174	-

版权所有 © 华为云计算技术有限公司

说明

- 开启极致RTO新框架时，推荐同时设置如下参数：
  - redo_bind_cpu_attr='cpuorderbind:0-255'
  - standby_recycle_interval=1
  - base_page_saved_interval=2000
  - pagewriter_sleep=1000
  - pagewriter_thread_num=6
  - max_io_capacity=3GB
- 在极致RTO新框架下，若回放速度降低，可执行如下命令查看buffer情况。
  ```
  select * from dbe_perf.global_candidate_status;
  ```
  如果candidate_slot列的值非常小，说明buffer中干净页面很少；get_buf_clock_sweep列的值很大，说明工作线程在自行淘汰页面，此时可以通过调大shared_buffer以增加buffer数量，或者提升磁盘性能规格从而提高回放速度。

设置不当的风险与影响：设置过大，会导致CPU、内存资源不足，造成回放速度降低。

enable_page_lsn_check

参数说明：数据页lsn检查开关。回放时，检查数据页当前的lsn是否是期望的lsn。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启数据页lsn检查开关。
- off：表示关闭数据页lsn检查开关。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数关闭后，可能无法快速发现数据页错误，导致预期外的报错。

promote_lsn_check

参数说明：控制回放结束升主时是否校验本地日志回放完整性。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启本地日志回放校验机制。
- off：表示关闭本地日志回放校验机制。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数关闭后，可能无法拦截因日志故障导致的提前升主操作，进而造成数据丢失风险。

recovery_min_apply_delay

参数说明：设置备节点回放的延迟时间。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：0（不增加延迟）

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：见下方须知。

设置不当的风险与影响：见下方须知。

须知

- 此参数主节点设置无效，必须设置在需要延迟的备节点上，推荐设置在异步备节点上，设置了延时的异步备节点如果升主RTO时间会比较长。
- 延迟时间是根据主服务器上事务提交的时间戳与备机上的当前时间来计算，因此需要保证主备系统时钟一致。
- 延迟时间设置过长时，可能会导致该备机Xlog文件所在的磁盘满，需要平衡考虑磁盘大小来设置延迟时间。
- 没有事务的操作不会被延迟。
- 主备切换之后，原主机若需延迟，需要再手动配置此参数。
- 当synchronous_commit被设置为remote_apply时，同步复制会受到这个延时的影响，每一个COMMIT都需要等待备机回放结束后才会返回。
- 使用这个特性也会让hot_standby_feedback被延迟，这可能导致主服务器的膨胀，因此这两个参数同时使用时要权衡风险。
- 主机执行了持有AccessExclusive锁的DDL操作，比如DROP和TRUNCATE操作，在备机延迟回放该条记录期间，如果在备机上对该对象执行查询操作，等待锁释放之后才会返回。

dcf_truncate_dump_info_level

参数说明：用于控制dcf做日志截断（truncate）时是否打印被截断lsn及之后的lsn。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：不开启。

版权所有 © 华为云计算技术有限公司

- 1：打印被dcf截断的所有lsn内容（大于等于截断lsn的Xlog日志）。
- 2：打印被dcf截断的所有lsn内容，且当落盘lsn大于截断lsn时打印warning级别日志。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

redo_bind_cpu_attr

参数说明：用于控制回放线程的绑核操作，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：长度大于0的字符串，该参数不区分大小写。

- "autobind"：日志回放线程自动绑定到一个NUMA组, 默认绑定到最后一个NUMA组。只针对ARM服务器生效，对于x86服务器及小规格的服务器不绑核。
- "nobind"：线程不做绑核。
- "nodebind: 1, 2"：利用NUMA组1,2中的CPU core进行绑核。
- "cpubind: 0-30"：利用0-30号CPU core进行绑核。
- "cpuorderbind: 16-32": 从16号核开始一个线程绑定一个CPU，区间内核不足就不参与绑定。建议区间大小设置为大于等于**recovery_parallelism** + 1。

默认值："autobind"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：见下方说明。

设置不当的风险与影响：错误的设置会造成数据库节点无法启动。

说明

- 本参数主要用于arm环境下的绑核操作。推荐将所有的回放线程绑定到一个NUMA组内，性能会更好，针对混合部署的场景，推荐将同一个机器上的不同节点的回放线程绑定到不同的NUMA组。非arm环境下使用NUMA绑核相关参数将导致集群启动失败。
- 通过本参数设置的绑核区间应与通过GUC参数**thread_pool_attr**设置的线程池绑核区间错开。

enable_wal_page_header_crc

参数说明：控制是否对wal日志页面头进行crc计算赋值和校验。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示开启wal日志页面头的crc计算赋值和校验。
- off：表示关闭wal日志页面头的crc计算赋值和校验。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数关闭后，可能无法校验wal日志页面头是否正确，进而造成备机接收到有误的wal日志，备机回放失败无法同步。

6.3.6.4 归档

archive_interval

参数说明：表示归档间隔时间。超过该参数设定的时间，则强制归档日志文件。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 1000

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致频繁归档，占用存储空间；设置过大则影响PITR的RPO。

time_to_target_rpo

参数说明：双集群异地灾备模式下，主集群日志将被归档到OBS。该参数设置主集群发生异常时至已归档到OBS的恢复点所允许的时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

- 0是指不开启日志流控。
- 1 ~ 3600表示主集群发生异常时至已归档到OBS的恢复点所允许的时间，保证主集群因灾难崩溃时，最多可能丢失数据的时长在允许范围内。

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会影响主机性能，设置过大会失去流控效果。

6.3.7 双机复制

版权所有 © 华为云计算技术有限公司

6.3.7.1 发送端服务器

max_wal_senders

参数说明：指定Xlog日志发送线程的并发连接最大数量，必须小于**max_connections**。

须知

设置max_wal_senders的前提：**wal_level**必须设置为archive、hot_standby或者logical以允许备机的连接。

参数类型：整型

参数单位：无

取值范围：0 ~ 1024（建议取值范围：8 ~ 100）

默认值：20

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 只有当使用单DN实例无主备场景下，该参数才可以设置为0。
- 当使用双机复制、备份恢复、逻辑解码时，该参数值建议设为：当前备机个数+备份连接数+所需的逻辑复制连接数。

设置不当的风险与影响：如果实际设置值小于建议取值范围，可能造成这些功能不可用或异常。

wal_keep_segments

参数说明：Xlog日志文件的数量，设置“pg_xlog”目录下保留日志文件的最小数量。备机通过获取主机的Xlog日志进行流复制。

参数类型：整型

参数单位：无

取值范围：2 ~ 2147483647

默认值：128

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 当服务器开启日志归档或者从检查点恢复时，保留的日志文件数量可能大于wal_keep_segments设定的值。
- 当双机为异步传输时，以COPY方式连续导入4G以上数据需要增大wal_keep_segments配置。以T6000单板为例，如果导入数据量为50G，建议调整参数为1000。可以在导入完成并且日志同步正常后，动态恢复此参数设置。
- 若**synchronous_commit**级别小于LOCAL_FLUSH，重建备机时，建议调大该参数为1000，避免重建过程中，主机日志回收导致重建失败。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：如果设置过大，可能会导致pg_xlog文件夹占用过大的磁盘空间。如果设置过小，则在备机请求事务日志时，此事务日志可能已经被产生的新事务日志覆盖，导致请求失败，主备关系断开。

wal_sender_timeout

参数说明：本节点等待事务日志接收端接收日志的最大等待时间。

须知

- 如果主机数据较大，重建备机数据库时需要增大此参数的值，主机数据在500GB时，此参数的参考值为600s。
- 此值不能大于**wal_receiver_timeout**或数据库重建时的超时参数。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：6000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，会对网络波动、闪断越敏感，可能会导致主备连接异常。设置过大，导致对异常的响应更缓慢。

max_replication_slots

参数说明：主机端的日志复制slot最大数量。

参数类型：整型

参数单位：无

取值范围：0 ~ 1024（建议取值范围：8 ~ 100）。

默认值：20

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

当使用双机复制、备份恢复、逻辑解码时，该参数值建议设为：当前物理流复制槽数+备份槽数+所需的逻辑复制槽数。如果实际设置值比上述建议值小，可能造成这些功能不可用或异常。

- 物理流复制槽提供了一种自动化的方法来确保主DN在所有备DN收到Xlog之前，Xlog不会被移除。因此物理流复制槽用于支撑集群HA。集群所需要的物理流复制槽数为：一组DN中，备DN数量与主DN的比例。例如，假设集群的DN高可用方案为1主3备，则所需物理流复制槽数为3。
- 备份槽：记录备份执行过程中的一些复制信息，全量备份和增量备份各自对应单独的备份槽，共2个。

版权所有 © 华为云计算技术有限公司

- 关于逻辑复制槽数，请按如下规则考虑。
  - 一个逻辑复制槽只能解码一个Database的修改，如果需要解码多个Database，则需要创建多个逻辑复制槽。
  - 如果需要多路逻辑复制同步给多个目标数据库，在源端数据库需要创建多个逻辑复制槽，每个逻辑复制槽对应一条逻辑复制链路。
  - 同一实例上，最多支持同时开启20个逻辑复制槽进行解码。

设置不当的风险与影响：设置过小，会导致备机无法连接，或逻辑解码、备份归档功能等无法启用；设置过大，会一定程度影响性能。容灾搭建时，主集群和灾备集群的max_replication_slots数量需要达成一致，否则可能会导致容灾搭建失败。

enable_wal_shipping_compression

参数说明：在流式容灾模式下，开启跨集群日志压缩功能。

须知

该参数仅作用于流式容灾中跨集群传输的一对walsender与walreceiver中，在主集群上配置。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开流式容灾跨集群日志压缩。
- off：表示关闭流式容灾跨集群日志压缩。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：无需单独设置，容灾搭建流程中会默认开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

repl_auth_mode

参数说明：设置主备复制和备机重建的验证模式。

版权所有 © 华为云计算技术有限公司

须知

- 如果主机上开启了UUID验证功能、且配置了非空字符串的repl_uuid验证码，那么备机也需要开启UUID验证功能、且配置相同的repl_uuid验证码，否则主备日志复制和备机重建请求将被主机拒绝。
- 该参数支持SIGHUP动态加载新值。修改之后不影响已建连的主备连接，对后续主备复制请求和主备重建请求生效。
- 支持Quorum、DCF协议下的备机重建验证；支持Quorum协议下的主备复制验证；不支持DCF协议下的主备复制验证。
- 不支持跨集群主、备之间的认证，包括Dorado主备集群和容灾主备集群。
- UUID验证功能主要为了防止主、备误连导致的数据串扰和污染，不是用于安全目的。
- 该参数不支持主、备间自动同步。

参数类型：枚举类型

参数单位：无

取值范围：

- off：表示关闭UUID验证功能。
- default：表示关闭UUID验证功能。
- uuid：表示开启UUID验证功能。

默认值：default

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据须知中的约束，结合需求判断是否开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

repl_uuid

参数说明：设置用于主备UUID验证的UUID码。

须知

- 如果主机上开启了UUID验证功能、且配置了非空字符串的repl_uuid验证码，那么备机也需要开启UUID验证功能、且配置相同的repl_uuid验证码，否则主备日志复制和备机重建请求将被主机拒绝。
- 该参数支持SIGHUP动态加载新值。修改之后，不影响已建连的主备连接，对后续主备复制请求和主备重建请求生效。
- 支持Quorum、DCF协议下的备机重建验证；支持Quorum协议下的主备复制验证；不支持DCF协议下的主备复制验证。
- 不支持跨集群主、备之间的认证，包括Dorado主备集群和容灾主备集群。
- UUID验证功能主要为了防止主、备误连导致的数据串扰和污染，不是用于安全目的。
- 该参数不支持主、备间自动同步。

版权所有 © 华为云计算技术有限公司

参数类型：字符串

参数单位：无

取值范围：字符串。长度0 ~ 63个字符，字母和数字的组合，大小写不敏感，内部统一转换为小写存储。空字符串表示不启用UUID验证功能。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据须知中的约束，结合需求判断是否开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

replconninfo1

参数说明：设置本端侦听和鉴权的第一个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo1 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第一个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo2

参数说明：设置本端侦听和鉴权的第二个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo2 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第二个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

版权所有 © 华为云计算技术有限公司

replconninfo3

参数说明：设置本端侦听和鉴权的第三个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo3 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第三个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo4

参数说明：设置本端侦听和鉴权的第四个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo4 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第四个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo5

参数说明：设置本端侦听和鉴权的第五个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo5 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第五个节点信息。

版权所有 © 华为云计算技术有限公司

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo6

参数说明：设置本端侦听和鉴权的第六个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo6 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第六个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo7

参数说明：设置本端侦听和鉴权的第七个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo7 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第七个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo8

参数说明：设置本端侦听和鉴权的第八个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

版权所有 © 华为云计算技术有限公司

```
replconninfo8 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第八个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo9

参数说明：设置本端侦听和鉴权的第九个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo9 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第九个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo10

参数说明：设置本端侦听和鉴权的第十个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo10 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

版权所有 © 华为云计算技术有限公司

replconninfo11

参数说明：设置本端侦听和鉴权的第十一个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo11 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十一个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo12

参数说明：设置本端侦听和鉴权的第十二个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo12 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十二个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo13

参数说明：设置本端侦听和鉴权的第十三个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo13 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十三个节点信息。

版权所有 © 华为云计算技术有限公司

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo14

参数说明：设置本端侦听和鉴权的第十四个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo14 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十四个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo15

参数说明：设置本端侦听和鉴权的第十五个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo15 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十五个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo16

参数说明：设置本端侦听和鉴权的第十六个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

版权所有 © 华为云计算技术有限公司

```
replconninfo16 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十六个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo17

参数说明：设置本端侦听和鉴权的第十七个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo17 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十七个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

replconninfo18

参数说明：设置本端侦听和鉴权的第十八个节点信息。集群安装成功后自动配置，无需手动修改。

示例：

```
replconninfo18 = "localhost=XXXX localport=XXXX localheartbeatport=XXXX localservice=XXXX
remotehost=XXXX remoteport=XXXX remoteheartbeatport=XXXX remoteservice=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第十八个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致集群状态异常。

版权所有 © 华为云计算技术有限公司

cross_cluster_replconninfo1

参数说明：设置跨集群的本端侦听和鉴权的第一个节点信息。

示例：

```
cross_cluster_replconninfo1 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第一个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo2

参数说明：设置跨集群的本端侦听和鉴权的第二个节点信息。

示例：

```
cross_cluster_replconninfo2 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第二个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo3

参数说明：设置跨集群的本端侦听和鉴权的第三个节点信息。

示例：

```
cross_cluster_replconninfo3 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第三个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo4

参数说明：设置跨集群的本端侦听和鉴权的第四个节点信息。

示例：

```
cross_cluster_replconninfo4 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第四个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo5

参数说明：设置跨集群的本端侦听和鉴权的第五个节点信息。

示例：

```
cross_cluster_replconninfo5 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第五个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo6

参数说明：设置跨集群的本端侦听和鉴权的第六个节点信息。

示例：

```
cross_cluster_replconninfo6 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第六个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo7

参数说明：设置跨集群的本端侦听和鉴权的第七个节点信息。

示例：

```
cross_cluster_replconninfo7 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第七个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

cross_cluster_replconninfo8

参数说明：设置跨集群的本端侦听和鉴权的第八个节点信息。

示例：

```
cross_cluster_replconninfo8 = "localhost=XXXX localport=XXXX remotehost=XXXX remoteport=XXXX"
```

参数类型：字符串

参数单位：无

取值范围：字符串。其中空字符串表示没有配置第八个节点信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：安装时会自动设置，不建议用户进行修改。

设置不当的风险与影响：错误的连接信息会导致灾备集群状态异常。

enable_time_report

参数说明：设定是否记录每条redo时间开销。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示记录每条redo时间开销。
- off：表示不记录每条redo时间开销。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：非调试模式不建议开启，开启后性能会有所下降。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

thread_top_level

参数说明：提高WALWRITERAUXILIARY、WALWRITER、STARTUP、WALRECEIVER、WAL_NORMAL_SENDER、PGSTAT线程的优先级到最高。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示提高上述线程优先级到最高。
- off：表示不提高上述线程优先级。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

page_work_queue_size

参数说明：每一个redo worker的阻塞队列长度。

参数类型：整型

参数单位：无

取值范围：1 ~ 100000

默认值：4096

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：过小会导致流水线能力失效，过大会占用过多内存。

parallel_build_thread_num

参数说明：代表并行重建的并行度。当取值为1时表示运行原有单线程重建代码，当取值为n时会建立n条线程完成重建时的数据传输。

参数类型：整型

参数单位：个

取值范围：1 ~ 10

默认值：1

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 建议设置为默认值。只有在重建数据文件量非常大，且磁盘I/O性能好、网络带宽、内存以及CPU资源充足的环境上，根据具体业务场景和机器磁盘I/O能力进行设置。并行度越大，占用的I/O、内存及CPU资源越多，重建速度也越快。
- 并行度与cpu及内存占用成正比（2个并行度时，CPU最高占用为3核，后续每增加1个并行度，CPU占用增加1核）。
- 假设单线程下build速度为 a MB/s, 则在enable_parallel_build_compression=off的情况下，每新增一个并行度，服务侧读io上升不超过 a MB/s；在enable_parallel_build_compression=on的情况下，每新增一个并行度，服务侧读io上升不超过0.7*a MB/s。

设置不当的风险与影响：

在不合适的环境下开启本开关，并行重建的速度提升效果不明显，且会占用额外资源。

全量并行重建每多加一个并行度会与服务端的一个新的walsender建立连接， 占用服务端guc参数**max_wal_senders**的额度， 并行度设置过大会因无法新建连接导致并行重建失败， 之后转单线程重建重试。

enable_parallel_build_compression

参数说明：控制是否在并行重建传输数据的过程中进行LZ4压缩，仅在**parallel_build_thread_num**>1情况下生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在并行重建传输数据的过程中进行LZ4压缩。
- off：表示在并行重建传输数据的过程中不进行LZ4压缩。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

建议设置为默认值。当网络带宽受限时，推荐配置值为on。

设置不当的风险与影响：

在不合适的环境下开启本开关，并行重建的速度提升效果不明显，且会占用额外资源。

replay_delay

参数说明：开启容灾延迟回放特性时，用于设置灾备集群延迟回放的时间。

RTO表示灾难发生时数据恢复所需的时长。RTO=（目标回放点-当前回放点）/回放速率，即待回放日志和回放速率的比值。而延迟回放特性使灾备集群成为主集群在延迟回放时长前的一个快照。该灾备集群执行failover升主操作时，目标回放点（目标回放

版权所有 © 华为云计算技术有限公司

点=当前时间点-replay_delay参数值）之后的日志将被截断，以确保failover升主的RTO不受replay_delay参数影响。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 4294967295，其中0表示不开启灾备集群延迟回放功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际需求设置。

须知

- 延迟回放时间配置跨度从分钟到年，当满足约束条件时设置的延迟回放时间可精确到秒，需满足以下时间同步约束：必须使用NTP等保证主集群与灾备集群之间时间同步精度<=100ms，灾备集群内节点间时间同步精度<=1ms。
- 灾备集群failover升主后，replay_delay将会被重置为默认值0。

设置不当的风险与影响：

- 若设置值过大，可能导致灾备集群长时间不执行回放，引发日志堆积，进而造成磁盘空间满。
- 当设置值不为0时，灾备集群无法执行switchover、容灾模拟、断恢容灾。

replay_mode

参数说明：容灾延迟回放特性开启时，用于控制灾备集群的回放暂停行为。

参数类型：字符串

参数单位：无

取值范围：

- ""：表示不开启暂停回放。
- stop：表示立即暂停回放。
- 时间戳：设置为时间戳表示暂停回放到该时间戳对应的barrier点。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际需求设置，需基于barrier点的时间戳进行配置。

须知

灾备集群failover升主后，replay_mode将会被重置为默认值""。

设置不当的风险与影响：若设置为已回放的barrier时间戳或不存在的barrier时间戳，将导致回放进程卡住。

版权所有 © 华为云计算技术有限公司

hadr_replconninfo_config_mode

参数说明：用于控制容灾链路使用的管理模式。

参数类型：枚举类型

参数单位：无

取值范围：

- local_file：表示容灾链路管理模式为基于GUC配置的模式。
- cluster_manager：表示容灾链路管理模式为基于集群管理配置的模式。

默认值：local_file

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：容灾流程会自动设置，不建议用户进行修改。

设置不当的风险与影响：容灾链路属性设置失效。

hadr_source_info

参数说明：用于为流式容灾日志接收端集群配置日志发送端集群的链路信息。

参数类型：字符串

参数单位：无

取值范围：字符串。其中""表示没有配置日志发送端集群的链路信息。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：容灾流程会自动设置，不建议用户进行修改；日志接收端集群hadr_source_info应和日志发送端集群配置的dest_name匹配。

设置不当的风险与影响：如果首备hadr_source_info未正确配置，会导致日志接收端集群与日志发送端集群无法建连。

6.3.7.2 主服务器

synchronous_standby_names

参数说明：潜在同步复制的备机名称列表，每个名称用逗号分隔。

须知

- 当前连接的同步备机是列表中的第一个名称。如果当前同步备机失去连接，则它会立即更换下一个优先级更高的备机，并将此备机的名称放入列表中。
- 备机名称可以通过设置环境变量PGAPPNAME指定。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：当取值为*，表示匹配任意提供同步复制的备机名称。支持按如下格式配置：

- ANY *num_sync* (*standby_name* [, ...])
- [FIRST] *num_sync* (*standby_name* [, ...])
- *standby_name* [, ...]

说明

- 其中*num_sync*是事务需要等待其回复的同步复制的备机的数量，*standby_name*是备机的名称，FIRST以及ANY指定从所列服务器中选取同步复制的备机的策略。
- ANY N (dn_instanceId1, dn_instanceId2,...)表示在括号内任选N个主机名称作为同步复制的备机名称列表。例如，ANY 1(dn_instanceId1, dn_instanceId2)表示在dn_instanceId1和dn_instanceId2中任选一个作为同步复制的备机名称。
- FIRST N (dn_instanceId1, dn_instanceId2,...)表示在括号内按出现顺序的先后作为优先级选择前N个主机名称作为同步复制的备机名称列表。例如，FIRST 1 (dn_instanceId1, dn_instanceId2)表示选择dn_instanceId1作为同步复制的备机名称。
- dn_instanceId1, dn_instanceId2,...和FIRST 1 (dn_instanceId1, dn_instanceId2,...)的含义相同。

若使用gs_guc工具设置该参数，需要如下设置：

```
gs_guc reload -Z datanode -N @NODE_NAME@ -D @DN_PATH@ -c "synchronous_standby_names='ANY 1(dn_instanceId1, dn_instanceId2)'"
```

或者：

```
gs_guc reload -Z datanode -N @NODE_NAME@ -D @DN_PATH@ -c "synchronous_standby_names='ANY 1(AZ1, AZ2)'"
```

默认值：*

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：错误的同步备列表会导致主备同步多数派协议失效。

须知

该参数由CM维护，如果手动修改，则存在丢失数据的风险。

most_available_sync

参数说明：指定在备机同步失败时，是否阻塞主机的事务提交。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在备机同步失败时，不阻塞主机的事务提交。
- off：表示在备机同步失败时，阻塞主机的事务提交。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：开启后，主机不会等待备机提交验证。主机处于单主时也会提交成功，而不是多数派写入成功。

设置不当的风险与影响：客户需要针对自身使用场景谨慎开启。

enable_stream_replication

参数说明：控制主备是否进行数据和日志同步。

须知

- 此参数属于性能测试参数，用于测试带有DN备机和不带DN备机的性能参数。关闭参数后，不能进行切换、故障等异常场景测试，否则会出现主备不一致的情况。
- 此参数属于受控参数，不建议正常业务场景下关闭此参数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开主备同步。
- off：表示关闭主备同步。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后，集群将失去多副本能力，会降低可用性，造成备份失败，不建议用户关闭。

enable_mix_replication

参数说明：控制主备之间WAL日志及数据复制的方式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开WAL日志、数据页混合复制模式。
- off：表示关闭WAL日志、数据页混合复制模式。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，默认值为off，不允许外部修改。

设置建议：不建议修改。此参数目前不允许正常业务场景下改变其值，即默认关闭WAL日志、数据页混合复制模式。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_wal_replication_compression

参数说明：在主DN和备DN间物理复制时设置启动Xlog日志压缩功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开日志压缩功能。
- off：表示关闭日志压缩功能。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数仅作用于主DN和备DN间传输的一对walsender与walreceiver线程中，在主DN上配置。小Xlog和时延敏感的场景下，不建议打开该参数。

设置不当的风险与影响：该参数开启后存在性能和资源的消耗，需合理配置**wal_replication_compression_thread_settings**、**wal_replication_compression_settings**参数。

- 性能影响：参见wal_replication_compression_settings参数详情。
- 不同资源场景下的配置：参见wal_replication_compression_thread_settings参数详情。

须知

该参数有以下约束：

- 升级过程中不允许修改压缩算法；
- 不支持DCF；
- 暂不支持容灾集群；
- 在小规格下（例如4U/8U）不建议打开该功能。

wal_replication_compression_thread_settings

参数说明：主备集群的并行压缩线程数和单线程队列深度。前提是**enable_wal_replication_compression**要开启，此参数才能生效。

参数类型：字符串

参数单位：无

取值范围：以','分隔的两个整型组成的字符串，两个整型参数取值范围都为[1,16]。

默认值：'1,6'

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数与**walsender_max_send_size**共同决定压缩线程内存占用，需合理配合使用，低性能规格机器建议设置较小值。

版权所有 © 华为云计算技术有限公司

编号	CPU个数	内存（GB）	是否混合部署	walsender_max_send_size（MB）	Xlog压缩最大线程数	压缩buffer最大队列深度	wal_replication_compression_thread_settings	备注
1	4	-	-	8	0	0	-	一般情况不推荐开Xlog压缩。客户场景存在网络流量限制时，可结合硬件规格对该参数进行设置，开启Xlog压缩来缓解流量压力。
2	8	-	是	8	0	0	-	
3	8	64	否	8	1	1	1,1	
4	16	128	是	8	2	2	2,2	
5	16	128	否	8	2	2	2,2	
6	32	256	是	8	4	2~4（性能测试后根据性能最优来确认推荐值）	4,2~4,4	
7	32	256	否	8	4	2~4（性能测试后根据性能最优来确认推荐值）	4,2~4,4	
8	64	512	是	8	8	2~4（性能测试后根据性能最优来确认推荐值）	8,2~8,4	

版权所有 © 华为云计算技术有限公司

编号	CPU个数	内存（GB）	是否混合部署	walsender_max_send_size（MB）	Xlog压缩最大线程数	压缩buffer最大队列深度	wal_replication_compression_thread_settings	备注
9	64	512	否	8	8	2~4（性能测试后根据性能最优来确认推荐值）	8,2~8,4	
10	96	768	-	8	12	2~4（性能测试后根据性能最优来确认推荐值）	12,2~12,4	

设置不当的风险与影响：如果该参数设置较大，则会占用较多动态内存，导致进程无法启动或者业务申请内存失败等问题。

wal_replication_compression_settings

参数说明：主备集群Xlog压缩采用的算法规则。

参数类型：字符串

参数单位：无

取值范围：参数值由压缩算法和压缩参数组成，格式为'string,int'，如'zstd,5'。其他格式为错误格式，会导致设置失败。第一个参数（压缩算法）可选lz4或zstd，其他值为无效值。

- 若第一个参数为lz4，则第二个参数为加速因子，取值范围为[1,300]。
- 若第一个参数为zstd，则第二个参数为压缩等级，取值范围为[1,15]。

默认值：'lz4,3'

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：

- lz4算法加速因子越大，则压缩率越低，性能越好。
- zstd压缩等级越大，则压缩率越高，性能越差。

版权所有 © 华为云计算技术有限公司

- 提高压缩率会减小网络带宽，但是TPS性能也会相应下降，建议设置为满足带宽限制的最小压缩率。
- 两种算法相较而言，zstd算法压缩率更高，性能下降更多；lz4算法性能下降较少，压缩率较低。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

data_replicate_buffer_size

参数说明：发送端与接收端传递数据页时，队列占用内存的大小。此参数会影响主备之间复制的缓冲区大小。

参数类型：整型

参数单位：kB

取值范围：4096 ~ 1072693248

默认值：131072（即128MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请按照建议的默认值设置，避免设置明显不合理（过大或过小）的data_replicate_buffer_size值。应满足条件：data_replicate_buffer_size + segment_buffers + shared_buffers + wal_buffers + temp_buffers + maintenance_work_mem + work_mem + query_mem +（备机）wal_receiver_buffer_size < max_process_memory < 物理机内存大小。

设置不当的风险与影响：设置的内存参数数值过大，超过物理内存上限，将会导致数据库由于无法申请到足够的内存而无法成功启动的情况。

walsender_max_send_size

参数说明：设置主机端日志或数据发送缓冲区的大小。

参数类型：整型

参数单位：kB

取值范围：8 ~ 2147483647

默认值：8192（即8MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会占用过多内存，设置过小会导致主备同步性能下降。

wal_buffer_reserve_percent

参数说明：设置主机端wal日志缓存保留窗口的比例，该窗口保留缓存中已落盘的日志数据。该参数设置为非0的有效值时，需满足**wal_buffers**设置值不小于1GB，该参数设置值才能生效。

参数类型：浮点型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：0.0 ~ 0.5

默认值：0.25

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ha_module_debug

参数说明：用于查看数据复制时具体数据块的复制状态日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示日志中将打印数据复制时每个数据块的状态。
- off：表示日志中不打印数据复制时每个数据块的状态。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：打印日志会占用不必要的磁盘、CPU、I/O带宽，不建议用户开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

sync_config_strategy

参数说明：主机和备机、备机和级联备之间配置文件的同步策略。

须知

在低于503.0.0版本升级到503.0.0版本后配置了该参数，回滚后该参数将无法识别。请勿在升级503.0.0版本期间配置该参数。

参数类型：枚举类型

参数单位：无

取值范围：

- all_node：主机配置为all_node时，表示允许主机向所有备机主动同步配置文件；备机配置为all_node时，表示允许当前备机向其主机发送同步请求，允许当前备机向其所有级联备主动同步配置文件；级联备配置为all_node时，表示允许当前级联备向其备机发送同步请求。
- only_sync_node：主机配置为only_sync_node时，表示仅允许主机向所有同步备机主动同步配置文件；备机配置为only_sync_node时，表示允许当前备机向其主机发送同步请求，不允许当前备机向其所有级联备主动同步配置文件；级联备配置为only_sync_node时，表示允许当前级联备向其备机发送同步请求。

版权所有 © 华为云计算技术有限公司

- none_node：主机配置为none_node时，表示不允许主机向任何备机主动同步配置文件；备机配置为none_node时，表示不允许当前备机向其主机发送同步请求，不允许当前备机向其所有级联备主动同步配置文件；级联备配置为none_node时，表示不允许当前级联备向其备机发送同步请求。

默认值：all_node

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：关闭后会使主机停止向备机同步GUC参数，谨慎设置为非all_node。

设置不当的风险与影响：设置成none_node后，主备之间不会再进行GUC参数的同步，请谨慎使用。

hadr_recovery_time_target

参数说明：在流式容灾模式下，设置hadr_recovery_time_target能够让备数据库实例完成日志写入和回放。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

- 0是指不开启日志流控。
- 1 ~ 3600是指备机能够在hadr_recovery_time_target时间内完成日志的写入和回放。可以保证主数据库实例与备数据库实例切换时能够在hadr_recovery_time_target时间内完成日志写入和回放，保证备数据库实例能够快速升主。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，频繁触发流控会影响主机性能。设置过大或关闭此参数，如果出现备机回放不及时，会导致备机RTO时间超出预期，降低高可用的RTO、RPO指标。

hadr_recovery_point_target

参数说明：在流式容灾模式下，设置hadr_recovery_point_target能够让备数据库实例完成日志刷盘的RPO时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

- 0是指不开启日志流控。
- 1 ~ 3600是指备机能够在hadr_recovery_point_target时间内完成日志的刷盘。可以保证主数据库实例与备数据库实例切换时，能够在hadr_recovery_point_target时间内完成日志回放，保障备数据库实例升主。

默认值：0

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，频繁触发流控会影响主机性能。设置过大或关闭此参数，如果出现备机回放不及时，会导致备机RTO时间超出预期，降低高可用的RTO、RPO指标。

hadr_super_user_record_path

参数说明：该参数为流式异地容灾参数，表示备集群中hadr_disaster用户的加密文件存放路径。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：由流式容灾密码传递工具自动设置，不需要用户手动添加。不建议修改。

设置不当的风险与影响：错误的修改会导致灾备集群状态异常，无法连接主集群导致无法同步数据等问题。

check_sync_standby

参数说明：备机检查的控制开关。备机检查开启的情况，主备场景下配置正确的**synchronous_standby_names**参数后，当同步备故障时，主机写业务直接报错写失败。

说明

- 该参数不支持在job work和自治事务中同步，有可能导致检查不生效。
- 若指定用户或session中未设置备机检查，开启强同步提交模式下备机故障，执行一个表的写操作会导致另一个用户或session中的同一个表的查询hang，此时需要备机恢复或者手动终止hang住的客户端。
- 不支持非写操作中触发写日志的场景中（vacuum analyze，gs_clean等）开启备机检查开关。若备机不满足同步备配置，则该场景会导致业务hang，需要手动终止。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启备机检查。
- off：表示关闭备机检查。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_wal_sender_crc_check

参数说明：开启walsender线程CRC校验，在主机发送日志前进行CRC校验。该参数开启后可能会导致性能下降，影响在5%以内。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启校验的功能。
- off：表示关闭校验的功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数保证日志有效性，用户根据场景需求确认是否开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

DCF模式下发送日志的线程为dcf_rep_appendlog，打开enable_wal_sender_crc_check支持在DCF模式下开启dcf_rep_appendlog线程发送日志前的CRC校验。

syncrep_wait_proceed_timeout

参数说明：在事务提交过程中，主机线程等待多数派确认时收到终止信号后进行超时检测的时间阈值。当多数派备机落盘位置持续未发生更新的时长超过该阈值，主机进程将主动退出。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000。0表示收到终止信号后不进行超时检测。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。若需启用该超时检测，设置值需不低于自动降副本流程的用时，建议不小于60。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。若设置值过小，可能会导致在自动降副本执行过程中主机进程提前退出。

async_build

参数说明：控制是否使用异步build流程重建故障DN。开启时采用异步流程重建，在剩余副本数不足多数派时，可优先恢复业务写入能力；关闭时采用同步逻辑重建，剩余副本数不足多数派时需等待数据与日志完全同步方可恢复业务。启用本功能后，若要求重建中的备DN参与多数派提交，需满足被重建的备DN在参数**synchronous_standby_names**配置的同步列表中，且除被重建DN外至少有一个同步列表中的DN存活。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示开启异步build参与多数派提交的功能。
- off：表示关闭异步build参与多数派提交的功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：该参数保障故障场景业务的快速恢复能力，典型场景示例：4副本部署中当故障DN达2个时，因多数派无法满足将阻塞写入新增业务。启用异步build参与多数派提交的功能恢复故障DN后，在重建故障DN过程中即可恢复新增业务的写入。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

须知

1. 业务恢复的时间由各个正在build实例与主机间的Xlog差距和网络速率决定，如Xlog相差1GB，网络速率为100MB/s，则业务应在build开始传输Xlog后10s左右恢复。
2. build状态下的新增事务可能暂时损失一定的数据可靠性，即在可用副本已无法满足多数派提交条件时，build接收日志达成多数派，但正在build的实例还未拥有完整的数据。若此时叠加故障导致剩余的所有副本同时无法工作且无法恢复，可能导致build过程中新写入的数据无法恢复。在正在build的实例恢复完成或触发CM的自动降副本能力后，可消除该风险。例如，在4副本场景中，仅剩余一主一备两副本Normal，其他两个副本处于build或待修复状态时，build接收日志可使主机事务达成多数派提交条件，但此时如果Normal状态的主备副本均发生无法恢复的损坏，可能造成故障期间的新增日志无法恢复。
3. 该参数在集群内各节点可以独立设置，只有主机和备机同时设为on时才生效。建议集群内所有节点保持参数一致。

6.3.7.3 备服务器

hot_standby

参数说明：设置是否允许备机在恢复到minrecovery点后接受连接和查询。

版权所有 © 华为云计算技术有限公司

须知

- 如果此参数设置为on，**wal_level**级别必须设置为hot_standby或以上，否则将导致数据库无法启动。
- 在分布式环境中，因为会对双机其他一些功能产生影响，hot_standby参数不能设置成off。
- 如果hot_standby参数曾经被关闭，且wal_level参数曾被设置低于hot_standby等级，那么，再次打开hot_standby参数之前，为了确保主备环境下备机上待回放的日志都可以支持备机查询功能，需要进行如下操作：
  1. 将主、备的wal_level参数调整到hot_standby等级或以上，并重启实例生效。
  2. 在主机上执行checkpoint操作，并通过查询pg_stat_get_wal_senders()系统函数，确认各个备机的receiver_replay_location追上主机当前的sender_flush_location，保证wal_level的调整同步到备机并生效，且备机不需要再回放之前低等级的日志。
  3. 将主、备的hot_standby参数打开（设为on），并重启实例生效。
- 备机读开启后，因备机回放与查询冲突，可能会出现查询被取消的报错，报错信息有：
  - ERROR: canceling statement due to conflict with recovery.
  - ERROR: terminating connection due to conflict with recovery.
- 在串行和并行回放备机读下，主机在进行在线重建索引时，备机读可能会报错，备机读请稍后重试。报错信息包含：
  - ERROR: could not open relation with OID xxx during recovery delete object, please try again later.
  - ERROR: Catalog is missing xxx attribute(s) for relid xxx.
  - ERROR: cache lookup failed for index xxx, refilenode:xxx, name:"xxx".
  - ERROR: could not find pg_class entry for xxx.
  - ERROR: could not open block during recovery delete object, please try again later.
- 在极致RTO备机读的情况下，内存配置较小的虚拟机可能出现内存不足的报错，可通过开启GUC参数**exrto_standby_read_opt**（默认开启）有效降低内存和I/O开销，解决此问题。
- 备机不支持自治事务功能。
- 在并行回放和极致RTO备机读场景下，Astore类型的表不支持生成IndexOnlyScan执行计划。
- 在极致RTO备机读场景下，若事务需使用读已提交（READ COMMITTED）的隔离级别，**exrto_transaction_isolation_force_rr**参数必须设置为off。当**exrto_transaction_isolation_force_rr**参数设置为on时，极致RTO备机读场景仅支持可重复读（REPEATABLE READ）的事务隔离级别，在可重复读隔离级别下使用备机读时需注意以下两项。若这两项均不满足，会话建立连接后无法读到新回放的数据。
  - 使用驱动连接数据库的场景下，需打开autocommit功能。
  - 在需关闭autocommit的场景下，如果执行长事务过程中需要读到最新回放的数据，需手动commit提交本次查询事务，然后再执行新的查询。
- 典型场景下的备机读性能：使用sysbench进行测试，主机执行100并发的update类型业务，主备机同时执行200并发的读类型业务，在I/O和CPU不受限的条件下，串

版权所有 © 华为云计算技术有限公司

行回放备机读的性能不低于主机读业务性能的80%，极致RTO备机读的性能和串行回放备机读的性能相比劣化不超过10%。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许备机在恢复到minrecovery点后接受连接和查询。
- off：表示不允许备机在恢复到minrecovery点后接受连接和查询。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：备机读功能开启后，即使没有响应业务，也会有一定的资源消耗，包括CPU、内存、磁盘空间、I/O占用。如果没有备机读业务，建议关闭此参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_standby_archive_delay

参数说明：当开启双机热备模式时，备机读场景下进行查询就会产生冲突，此参数就是设置备机取消查询之前所等待的时间。当前版本设置暂不生效，统一由参数**max_standby_streaming_delay**控制。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 2147483647。-1表示允许备机一直等待冲突的查询完成。

默认值：3000（即3s）

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_standby_streaming_delay

参数说明：当开启双机热备模式时，如果备机正通过流复制接收WAL日志数据，这时进行查询就会产生冲突，这个参数就是设置备机取消查询之前所等待的时间。当参数值较大，或业务压力大时，概率出现查询与回放冲突的报错。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 2147483647

版权所有 © 华为云计算技术有限公司

须知

-1表示允许备机一直等待冲突的查询完成。在开启串行或并行回放的场景下，当检测到查询线程与回放线程已处于死锁状态时，仍会通过取消查询来避免阻塞回放。

0表示备机在出现回放与查询的冲突时，立即取消查询，不会等待。

默认值：3000（即3s）

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：需根据业务需求谨慎调整，建议设置为默认值。设置值越大，备机的查询业务优先级越高，但会影响备机回放；设置值越小，长查询越容易被取消，但回放的优先级越高。

设置不当的风险与影响：设置过大时，如果出现与回放冲突的长查询，有可能导致回放长时间被阻塞，影响此备机的RTO和RPO指标。

wal_receiver_status_interval

参数说明：WAL日志接收线程的状态通知给主机的最大时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483

默认值：5

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当该参数设置为0时，表示关闭备机向主机反馈日志接收位置等信息，可能会导致主机事务提交阻塞、switchover操作失败等异常现象。正常业务场景，不建议将该参数设置为0。

hot_standby_feedback

参数说明：将备机上执行查询的最小事务号反馈给主机，可以避免查询冲突。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许将备机上执行查询的最小事务号反馈给主机。
- off：表示不允许将备机上执行查询的最小事务号反馈给主机。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当该参数为on时，主机的旧版本数据的清理会受限于备机正在读的事务，即主机只允许清理小于备机反馈的事务号所作的更改。因此若该参数开

版权所有 © 华为云计算技术有限公司

启时，会影响主机的性能。若备机回放与查询冲突，会出现查询报错，因此建议适当调大**max_standby_streaming_delay**。

wal_receiver_timeout

参数说明：设置从主机接收数据的最大等待时间。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：6000（即6s）

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，会导致网络波动时，walreceiver线程频繁断开。

wal_receiver_connect_timeout

参数说明：设置连接主机的最大等待超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

wal_receiver_connect_retries

参数说明：设置连接主机的最大尝试次数。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

wal_receiver_buffer_size

参数说明：备机接收Xlog存放到内存缓冲区的大小。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：kB

取值范围：4096 ~ 1047552

默认值：65536（即64MB）

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

primary_slotname

参数说明：设置备机对应主机的slot name，用于主备校验。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：错误设置可能会导致备机无法连接上主机，集群状态异常。

max_standby_base_page_size

参数说明：开启极致RTO功能后，备机上允许base page类型文件的最大存储空间。

参数类型：整型

参数单位：kB

取值范围：1048576 ~ 562949953421311

默认值：268435456（即256GB）

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据实际物理磁盘大小，对此参数进行设置。

设置不当的风险与影响：设置过小，会导致备机读报错概率增加。

max_standby_lsn_info_size

参数说明：开启极致RTO功能后，备机上允许lsn info类型文件的最大存储空间。

参数类型：整型

参数单位：kB

取值范围：1048576 ~ 562949953421311

默认值：268435456（即256GB）

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据实际物理磁盘大小，对此参数进行设置。

设置不当的风险与影响：设置过小，会导致备机读报错概率增加。

max_keep_csn_info_size

参数说明：开启极致RTO功能后，备机上每个DN允许的csn info链表的最大存储空间。

参数类型：整型

参数单位：kB

取值范围：16384 ~ 131072

默认值：16384（即16MB）

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

base_page_saved_interval

参数说明：开启极致RTO功能后，备机上生成base page的间隔次数。对同一个页面来说，每回放该参数值的次数，生成一次base page。

参数类型：整型

参数单位：无

取值范围：5 ~ 2000

默认值：400

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：间隔过大，会使得每次备机读需要回放的Xlog较多，性能变差；间隔过小，会占用更多的磁盘空间，更快到达强制回收点。

standby_force_recycle_ratio

参数说明：开启极致RTO功能后，备机读相关文件触发强制回收的比例。当base page文件总大小超过**max_standby_base_page_size*** standby_force_recycle_ratio，或者lsn info文件总大小超过**max_standby_lsn_info_size*** standby_force_recycle_ratio时，触发强制回收，会有查询被取消。当standby_force_recycle_ratio = 0时，不会启动强制回收，max_standby_base_page_size和max_standby_lsn_info_size也不会生效。

参数类型：浮点型

参数单位：无

取值范围：0.0 ~ 1.0

版权所有 © 华为云计算技术有限公司

默认值：0.8

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

standby_recycle_interval

参数说明：开启极致RTO功能后，备机读相关文件回收的时间间隔。备机读的资源回收线程，每间隔该参数值的时间，尝试清理一次备机读相关文件。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 86400。0表示不会回收历史版本页面，同时也会影响备机Xlog的回收，会导致备机占用过多的磁盘空间。

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置间隔过长，会导致历史页面回收不及时，可能会导致过多的磁盘占用。

standby_max_query_time

参数说明：开启极致RTO功能后，支持备机上查询的最大时间，超过该时间会被取消。

说明

何时取消查询受回收线程的时间间隔参数**standby_recycle_interval**和查询取快照的时间影响，因此备机上查询的实际执行时间要大于该参数。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 86400。0表示回收线程在回收历史页面版本时，若发现查询的快照小于当前回收点，则立即取消查询。

默认值：600

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：建议根据实际业务需要进行调整。

设置不当的风险与影响：如果设置过高，回收线程在计算回收点时会受到影响，可能会阻塞历史版本和Xlog日志的回收，从而导致磁盘空间占用增加。

exrto_transaction_isolation_force_rr

参数说明：在极致RTO备机读场景下，指定事务是否被强制设置为可重复读隔离级别。

参数类型：布尔类型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：在极致RTO备机读场景下，事务被强制设置为可重复读隔离级别。
- off：在极致RTO备机读场景下，事务可设置为任意隔离级别。

须知

在极致RTO备机读场景下，需要先确认exrto_transaction_isolation_force_rr参数值为off，再通过**transaction_isolation**参数设置事务隔离级别。

默认值：

- 新安装的数据库默认值为off。
- 若升级的基础版本不存在该GUC参数，默认值为on；若升级的基础版本存在该GUC参数，保留原设置。

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

exrto_standby_read_opt

参数说明：支持极致RTO备机读优化。主机和备机间不同步该参数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启优化。
- off：表示关闭优化。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

walrcv_writer_crc_check_level

参数说明：支持流式容灾场景下，主备集群内备机侧的Xlog校验，默认仅在灾备集群内开启。

参数类型：整型

参数单位：无

取值范围：0 ~ 3

版权所有 © 华为云计算技术有限公司

- 0：关闭校验。
- 1：灾备集群内校验生效。
- 2：主集群内备机侧的校验生效。
- 3：主集群内的备机和灾备集群内节点全部生效。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：容灾搭建流程会自动设置，无需用户手动设置。校验失败会导致主备机断连、备集群状态短暂异常，重新拉起后即可恢复。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

standby_read_use_az_info

参数说明：GTM-Lite备机读是否使用AZ信息，优先选择和CN相同的AZ的备机，默认关闭。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：建议根据实际业务需要进行设置，打开开关后，由于需要遍历所有可用备DN节点，再去从中选择，可能会导致查询性能下降。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

standby_read_use_load_balance

参数说明：GTM-Lite备机读是否使用负载均衡功能，随机挑选满足一致性点（和AZ信息）的DN，默认关闭。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：建议根据实际业务需要进行设置，打开开关后，由于需要遍历所有可用备DN节点，再去从中选择，可能会导致查询性能下降。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_standby_bufferpool

参数说明：启用极致RTO备机读独立缓冲区功能的开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

启用该功能开关时，开关生效需满足**enable_incremental_checkpoint**增量检查点打开，且参数**standby_shared_buffers_fraction**减去参数**standby_bufferpool_scale**的差值不小于0.1。

standby_bufferpool_scale

参数说明：设置用于开启极致RTO备机读独立缓冲区功能后，独立缓冲区使用**shared_buffers**内存缓冲区大小的比例。

参数类型：浮点型

参数单位：无

取值范围：0.1-0.9

默认值：0.5

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若该值设置不合理，会导致刷页速度加快。

wal_replication_ssl

参数说明：与GUC参数**ssl**同时使用控制主备之间是否开启SSL通信，只有两个参数均为on时，主备才会走ssl加密传输，如果关闭可以提高主备之间通信性能。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若该值设置不合理，会导致主备之间通讯安全性下降。

6.3.7.4 逻辑复制

max_changes_in_memory

参数说明：逻辑解码时，单个事务在内存中缓存的DML语句数量上限。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：4096

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，解码到大事务会占用较多的动态内存；设置过小，解码效率会降低。

max_cached_tuplebufs

参数说明：逻辑解码时，总元组信息在内存中缓存的数量上限。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：8192

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。建议设置为**max_changes_in_memory**的两倍以上。

设置不当的风险与影响：设置过大，解码到大事务会占用较多的动态内存；设置过小，解码效率会降低。

logical_decode_options_default

参数说明：逻辑解码任务启动时的全局默认参数。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：当前支持指定的逻辑解码选项包括：parallel-decode-num、parallel-queue-size、max-txn-in-memory、max-reorderbuffer-in-memory、exclude-users、skip-generated-columns、decode-sequence、data-limit、max-control-memory-size、max-toast-memory-size、max-queue-cache-num、desc-memory-limit、enable-cn-decode。选项的详细说明，请参见《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写） > 逻辑解码选项”章节。

须知

- 通过逗号分隔的key=value字符串，例如："parallel-decode-num=4,parallel-queue-size=128,exclude-users=userA,skip-generated-columns=on"。
- logical_decode_options_default配置的exclude-users选项和逻辑解码启动的exclude-users选项存在差异，不允许指定多个黑名单用户。
- logical_decode_options_default配置的decode-sequence仅支持设置为false。

默认值：""。空字符串表示采用程序默认值。

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

须知

该参数SIGHUP生效并不会影响已经启动的逻辑解码流程；后续逻辑解码启动将使用该参数设置作为其默认配置，如果解码任务启动命令中有相同配置项，则logical_decode_options_default对应选项的默认配置值不生效，按启动命令中的设定值进行配置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

logical_decode_options_extend

参数说明：逻辑解码任务启动时的全局默认参数。该参数可在PDB级别设置。

参数类型：字符串

参数单位：无

取值范围：包含事务xid的字符串（3~18446744073709551615），不同的事务以'/'为分隔符进行隔离。例如："skip-xid-list=9737819/9737823"。

默认值：""，表示采用程序预设的默认值。

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。当前支持指定的逻辑解码选项包括：skip-xid-list，选项详细说明请参见《开发指南》中“特性使用指导 > 逻辑复制 > 逻辑解码（非透明多写） > 逻辑解码选项”章节。

版权所有 © 华为云计算技术有限公司

须知

该参数SIGHUP生效并不会影响已经启动的逻辑解码流程，后续逻辑解码启动将使用该参数设置作为其默认配置，如果解码任务启动命令中有相同配置项，则logical_decode_options_extend对应选项的默认配置值不生效，按启动命令中的设定值进行配置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，如果设置xid有误可能会导致不可预期的事务被跳过导致解码数据丢失。

logical_sender_timeout

参数说明：设置本端等待逻辑日志接收端接收日志的最大等待时间。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：30000（30s）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_logicalrepl_xlog_prune

参数说明：设置是否开启逻辑复制槽强制失效功能。当GUC参数enable_logicalrepl_xlog_prune=on、**max_size_for_xlog_retention**为非零值时，如果max_size_for_xlog_retention大于0且当前逻辑复制槽导致保留日志的段数（每段日志大小为16MB）超过max_size_for_xlog_retention，或者max_size_for_xlog_retention小于0且磁盘使用率达到(-max_size_for_xlog_retention)/100，则该逻辑复制槽会强制失效，restart_lsn被置为7FFFFFFF/FFFFFFFF，该复制槽不阻塞Xlog日志的回收。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启逻辑复制槽强制失效功能。
- off：表示关闭逻辑复制槽强制失效功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需要确认是否开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_logical_replication_ddl

参数说明：设置逻辑解码是否支持DDL，是否反解析，是否生成日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：逻辑复制可支持DDL，对DDL执行结果进行反解析，并生成DDL的WAL日志。
- off：不支持DDL，不反解析也不生成WAL日志。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_keep_log_seg

参数说明：流控参数，逻辑复制在DN本地会解析物理日志转换成逻辑日志，当未被解析的物理日志文件数量大于该参数时会触发限流。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647。0表示关闭限流功能。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求，判断是否需要进行流控。

设置不当的风险与影响：设置过大，流控一直不触发。

logical_replication_dictionary_retention_time

参数说明：GS_TXN_LSN_TIME以及数据字典相关系统表数据保留时间。

参数类型：整型

参数单位：d（天）

取值范围：1 ~ 3650

默认值：365

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求进行设置。

设置不当的风险与影响：设置过小，会导致指定位点解码的可选区间过小；设置过大，会影响解码性能。

版权所有 © 华为云计算技术有限公司

logical_replica_identity_force

参数说明：控制全局逻辑复制旧元组的记录级别。

参数类型：枚举类型

参数单位：无

取值范围：

- 'full'：表示所有列都记录为副本标识。
- 'unique'：表示副本标识为主键和唯一键。
- 'default'：表示复制副本标识为默认选择（主键或无）。
- 'nothing'：表示没有记录此关系的副本标识。
- 'none'：表示此全局控制参数不生效。

默认值：'none'

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

说明

如需将已设置为非默认值的logical_replica_identity_force参数重新设置成'none'，须使用将GUC参数恢复成默认值的方法。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.8 查询规划

介绍查询优化器方法配置、开销常量、规划算法以及一些配置参数。

说明

- 优化器中涉及的两个参数：
  - INT_MAX数据类型INT的最大值，其值为2147483647。
  - DBL_MAX数据类型FLOAT的最大值。
- 全局设置查询规划相关参数除了客户业务外也会对数据库自身运维和监控业务造成影响，如WDR报告生成、扩容、重分布、数据导入导出等。

6.3.8.1 优化器方法配置

这些配置参数提供了影响查询优化器选择查询规划的原始方法。如果优化器为特定的查询选择的缺省规划并不是最优的，可以通过使用这些配置参数强制优化器选择一个不同的规划来临时解决这个问题。更好的方法包括调节优化器开销常量、手动运行ANALYZE、增加配置参数default_statistics_target的值、增加使用ALTER TABLE SET STATISTICS为指定列增加收集的统计信息。

enable_bitmapscan

参数说明：控制优化器对位图扫描规划类型的使用。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后会全局禁止bitmap scan算子，可能导致部分场景性能下降。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

force_bitmapand

参数说明：控制优化器对bitmap and规划类型的强制使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开该参数后当满足bitmap and索引扫描算子使用条件时，优化器生成计划会强制走bitmap and索引扫描，部分场景下可能影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

enable_hashagg

参数说明：控制优化器对Hash聚集规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后会导致部分场景下Hash聚集算子无法使用，影响查询改写逻辑，使查询性能下降。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

版权所有 © 华为云计算技术有限公司

enable_hashjoin

参数说明：控制优化器对Hash连接规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后会全局禁止使用Hash Join算子，优先生成其他Join类型计划，可能导致部分场景性能下降。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

enable_indexscan

参数说明：控制优化器对索引扫描规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后会全局禁止使用索引扫描算子，可能导致部分场景性能下降。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

enable_indexonlyscan

参数说明：控制优化器对仅索引扫描规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后会全局禁止使用仅索引扫描算子，一些仅需要走索引扫描不需要回表的计划将无法生成，可能导致部分场景性能下降。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

enable_gsiscan

参数说明：控制优化器对仅全局二级索引扫描规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：在有全局二级索引的场景下，如果关闭该参数，会导致无法生成仅全局二级索引扫描计划，可能影响查询性能。

enable_gsitablescan

参数说明：控制优化器对全局二级索引回表扫描规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：在有全局二级索引的场景下，如果关闭该参数，会导致无法生成全局二级索引回表扫描计划，可能影响查询性能。

gsi_fastpath_level

参数说明：控制优化器是否执行全局二级索引简单查询的规则判断系统，全局二级索引的简单查询规则如下：

- 单个表、没有子查询、没有子链接、目标表不是视图。
- 查询语句无法生成普通索引的执行计划。

版权所有 © 华为云计算技术有限公司

- 查询语句无法生成GSI的IndexOnlyScan执行计划。
- 查询语句不包含Scan类的Hint，如TableScan、IndexScan等。
- WHERE条件中包含等值条件且等值条件包含候选GSI（唯一且不是部分GSI索引）的索引键。

参数类型：枚举类型

参数单位：无

取值范围：

- strict：等值条件包含GSI所有索引键。
- partial：等值条件包含GSI的部分索引键（最左匹配），性能不一定最优。
- none：关闭。

默认值：strict

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能影响部分GSI场景下的计划生成，影响查询性能。

说明

partial的使用参考如下场景：

```
gaussdb=# CREATE TABLE t1(a INT, b INT, c INT);
gaussdb=# CREATE GLOBAL UNIQUE INDEX t1_gsi ON t1(b,c);
-- Q1
gaussdb=# SELECT * FROM t1 WHERE b=1;
-- Q2
gaussdb=# SELECT * FROM t1 WHERE b=1 LIMIT 1;
```

上述场景中，GSI索引建在(b,c)两列上，gsi_fastpath_level参数设置为partial时，查询语句Q1和Q2才可以生成GSI非Stream回表计划。索引唯一性由(b,c)两列保证，对于查询语句Q1，返回结果可能有多条，GSI非Stream回表计划的性能不一定是最优的，对于查询语句Q2，由于包含limit 1的条件，GSI非Stream回表计划的性能依然是最优的。

enable_default_gsi

参数说明：（1）控制在创建表时，对于不包含分布键的列添加主键约束或者唯一约束时是否默认创建全局二级索引；（2）控制hashbucket表是否可以创建全局二级索引。

参数类型：布尔型

参数单位：无

取值范围：

- on：（1）创建表时，对于不包含分布键的列添加主键约束或者唯一约束时默认创建全局二级索引；（2）hashbucket表可以创建全局二级索引。
- off：（1）创建表时，对于不包含分布键的列添加主键约束或者唯一约束时不会默认创建全局二级索引，创建表失败；（2）hashbucket表不可以创建全局二级索引。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。用户需按照实际使用场景决策是否打开此参数。

设置不当的风险与影响：

- 如果此参数为off，（1）创建表时无法对不包含分布键的列添加主键约束或者唯一约束，建议优先通过业务改造添加分布键来解决；（2）无法对hashbucket表创建全局二级索引。
- 如果此参数为on，（1）通过默认创建全局二级索引，表上可以对不包含分布键的列添加主键约束或者唯一约束，需注意表上如果创建了全局二级索引，DML操作会有潜在的性能下降；（2）hashbucket表上的部分索引、表达式索引形式的GSI会在重分布过程中进行离线重建，重建期间会阻塞业务，非部分索引、非表达式索引形式的GSI，在重分布过程中进行在线重建，不会阻塞业务。加节点、重分布的全程GSI保持有效。

enable_gsi_iud_returning

参数说明：控制是否打开对于建有全局二级索引的基表上执行INSERT、UPDATE、DELETE操作时支持RETURNING功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开。
- off：表示关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：在建有全局二级索引的基表上执行INSERT、UPDATE、DELETE操作的场景下，如果关闭该参数将无法支持RETURNING功能。

enable_gsi_distkey_update

参数说明：控制是否打开全局二级索引分布键更新（UPDATE、UPSERT、MERGE INTO）功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开。
- off：表示关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果关闭该参数将无法支持全局二级索引分布键更新。

版权所有 © 华为云计算技术有限公司

enable_gsi_dml_lp_opt

参数说明：控制优化器是否执行全局二级索引简单DML的规则判断系统，对于简单DML的执行进行优化。全局二级索引的简单DML优化规则如下：

- 针对INSERT语句，仅支持以下形式的简单INSERT：
  - 仅支持单条VALUES子句（不支持多条VALUES）；
  - VALUES中的值仅限于常量或参数（不支持表达式）；
  - 不支持RETURNING子句、任何形式的子查询以及各类UPSERT语句；
  - 目标表必须为单个表，该表上存在不超过一个全局二级索引，且该索引必须为普通全局二级索引（不支持表达式索引或部分索引），同时该索引不能包含自定义类型列。
- 其他DML：目前不支持。
- 该参数需要CN和DN同时设置，保持一致。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开。
- off：表示关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果关闭该参数，无法支持GSI DML Light Proxy优化，可能导致部分GSI DML场景性能下降。

enable_material

参数说明：控制优化器对实体化的使用。消除整个实体化是不可能的，但是可以关闭这个参数以防止优化器插入实体节点。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后会禁止优化器插入实体节点，部分场景下无法生成Material算子，可能导致部分场景性能下降。

版权所有 © 华为云计算技术有限公司

enable_mergejoin

参数说明：控制优化器对融合连接规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开该参数优化器会对满足条件的Join连接生成融合连接路径，可能导致该场景下选择的连接算子非实际最优。

enable_nestloop

参数说明：控制优化器对内表全表扫描嵌套循环连接规划类型的使用。完全消除嵌套循环连接是不可能的，但是，若关闭这个参数，当存在其他方法时，优化器会优先选择其他方法。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开该参数优化器会对满足条件的Join连接生成NestloopJoin路径，可能导致该场景下选择的连接算子非实际最优。

enable_index_nestloop

参数说明：控制优化器对内表参数化索引扫描嵌套循环连接规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会导致部分场景下无法生成参数化路径，通常参数化路径可以提升查询性能，因此一般建议开启。

enable_remote_param

参数说明：控制是否使用XC参数化路径特性，打开后能够生成XC参数化路径计划。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：

- on：新安装的数据库的默认值。
- off：从505.2.0之前版本升级后的默认值。

说明

该参数自505.2.0版本引入，在507.0.0修改默认值，由off改为on。从505.2.0之前版本升级后该参数为off；若升级前存在该参数并且已设置取值，升级后将保留原参数值。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在相关子链接和Nestloop场景中，若外表规模较小且内表规模较大，同时内表在关联条件上存在符合最左前缀匹配原则的索引时，建议将参数值设置为on。此配置下，若代价模型选择参数化路径，能够提升性能。

设置不当的风险与影响：在相关子链接和Nestloop场景中，若外表规模较大而内表规模较小，将此参数设置为on，且代价模型选择了参数化路径，可能会导致性能下降。

版权所有 © 华为云计算技术有限公司

须知

- XC参数化路径生成条件包括：
  - 条件中不能包含不可下推表达式。（如PROVOLATILE属性为volatile的函数）。
  - 条件能够遵循最左匹配规则匹配索引。
  - 外连接的WHERE条件不支持参数化。
- XC参数化支持场景包括：
  - 支持生成Nestloop的参数化路径，将Join条件下推到内表RemoteQuery（包括单表RemoteQuery和RemoteJoinQuery）上，以便将外表的查询结果作为参数传递到内表。
  - 支持生成Subplan的参数化路径，将Subplan上的Coordinator Qual下推到Subplan的RemoteQuery上，以便将Subplan的父查询的结果作为参数传递到内表。
- 使用XC参数化路径特性存在以下约束：
  - 仅支持XC计划，不支持Stream和FQS计划。
  - 当且仅当基表生成索引参数化路径时，才对基表生成带参数rescan的计划。
  - Nestloop参数化路径仅支持内表为RemoteQuery的场景。
  - Nestloop参数化路径不支持将参数下推到子查询上。
  - 不支持predpush Hint。
  - M-Compatibility模式数据库下，max_datanode_for_plan参数设置大于0时，不支持显示参数化路径DN计划。
  - 不支持remote query动态裁剪DN。

enable_remote_pbe_reuse

参数说明：控制XC计划中CN下发DN的PBE报文是否复用。打开后XC计划中CN下发DN的报文会由未命名报文变成带“_disqry_sys_cached”前缀的报文，并在执行过程中进行复用，进而提升查询性能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示复用。
- off：表示不复用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会在部分场景下影响XC计划查询性能。

版权所有 © 华为云计算技术有限公司

enable_seqscan

参数说明：控制优化器对顺序扫描规划类型的使用。完全消除顺序扫描是不可能的，但是，若关闭这个参数，当存在其他方法时，优化器会优先选择其他方法。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会导致优化器倾向于选择其他扫描方法，降低顺序扫描方法的优先级，部分场景下可能会影响查询性能。

enable_sort

参数说明：控制优化器对排序方法的使用。完全消除明确的排序是不可能的，但是，若关闭这个参数，当存在其他方法时，优化器会优先选择其他方法。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会导致优化器倾向于选择其他方法，降低排序方法使用的优先级，产生的不良影响较小，通常建议打开。

enable_tidscan

参数说明：控制优化器对TID扫描规划类型的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会导致优化器倾向于选择其他方法，降低TID扫描方法使用的优先级，产生的不良影响较小，通常建议打开。

enable_kill_query

参数说明：CASCADE模式删除用户时，会删除此用户拥有的所有对象。此参数标识在删除用户的时候，是否允许取消锁定此用户所属对象的query。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许取消锁定。
- off：表示不允许取消锁定。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开后，用户被删除时，该用户对应query可能会暴露，对系统的一致性和安全性带来不利影响。

enable_stream_concurrent_update

参数说明：控制优化器在并发更新场景下对stream的使用，该参数受限于**enable_stream_operator**参数。如果出现并发update冲突，有可能会出现报错，可以通过本参数控制。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许优化器对update语句生成stream计划。
- off：表示优化器对update语句仅能生成非stream计划。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会导致优化器倾向于选择其他方法，对update语句仅能生成非stream计划，产生的不良影响较小，通常建议打开。

enable_stream_operator

参数说明：控制优化器对stream的使用。当enable_stream_operator参数关闭时，会有大量关于计划不能下推的日志记录到日志文件中。如果用户不需要这些日志内容，

版权所有 © 华为云计算技术有限公司

建议用户在enable_stream_operator参数关闭时，也同时关闭enable_unshipping_log参数。裸机安装时，参数会被设置为off。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：

- 独立部署：off

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数优化器将无法生成Stream计划，可能导致查询性能下降。

enable_stream_recursive

参数说明：控制是否将with recursive关联查询下推DN分布式执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持使用with recursive关联查询下推DN分布式执行。
- off：表示不支持使用with recursive下推。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致查询性能下降。

max_recursive_times

参数说明：控制with recursive的最大迭代次数。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：200

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐根据实际业务评估合理值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：对于with recursive语句，此参数用于避免陷入死循环，层次查询可用nocycle跳出循环。若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

enforce_a_behavior

参数说明：控制正则表达式的规则匹配模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示正则表达式采用A格式的匹配规则。
- off：表示正则表达式采用POSIX格式的匹配规则。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务场景进行设置。

设置不当的风险与影响：查询语句中正则表达式无法按预期匹配。

enable_vector_engine

参数说明：控制优化器对向量化执行引擎的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致优化器未能针对特定场景生成向量化的查询计划，从而影响查询性能。

enable_broadcast

参数说明：控制优化器在stream代价估算时对broadcast分布方式的使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

版权所有 © 华为云计算技术有限公司

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能会影响stream计划生成的准确性，从而影响查询性能。

enable_change_hjcost

参数说明：控制优化器在Hash Join代价估算路径选择时，是否使用将内表运行时代价排除在Hash Join节点运行时代价外的估算方式。如果使用，则有利于选择条数少，但运行代价大的表做内表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能会导致Hash Join代价估算有误，从而影响带有Hash Join计划生成的准确性。

best_agg_plan

参数说明：本参数用于控制优化器生成哪种hashagg的计划。对于stream下的Agg操作，优化器会生成三种计划：

1. hashagg+gather(redistribute)+hashagg。
2. redistribute+hashagg(+gather)。
3. hashagg+redistribute+hashagg(+gather)。

参数类型：整型

参数单位：无

取值范围：0 ~ 3

- 0：优化器会根据以上三种计划的估算cost选择最优的一种计划生成。
- 1：强制生成第一种计划。
- 2：如果group by列可以重分布，强制生成第二种计划，否则生成第一种计划。
- 3：如果group by列可以重分布，强制生成第三种计划，否则生成第一种计划。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：该参数会影响hashagg的生成方式，通常建议使用默认值，设置其他值会倾向于某一类hashagg，设置不当可能会影响部分场景下查询的性能。

agg_redistribute_enhancement

参数说明：进行Agg操作时，如果包含多个group by列且均不为分布列，进行重分布时会选择某一group by列进行重分布。本参数控制选择重分布列的策略。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示会选择估算distinct值最多的一个可重分布列作为重分布列。
- off：表示会选择第一个可重分布列作为重分布列。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开会增加计划生成时间，但可能会选择更合理的重分布列。

enable_absolute_tablespace

参数说明：控制表空间是否可以使用绝对路径。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示可以使用绝对路径。
- off：表示不可以使用绝对路径。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数创建表空间时将不允许使用绝对路径，建议根据业务实际需要设置，无其他风险。

enable_valuepartition_pruning

参数说明：是否对DFS分区表进行静态/动态优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示对DFS分区表进行静态/动态优化。

版权所有 © 华为云计算技术有限公司

- off：表示不对DFS分区表进行静态/动态优化。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后DFS分区表相关查询性能可能下降。

expected_computing_nodegroup

参数说明：标识选定的计算Node Group模式或目标计算Node Group。Node Group目前为内部机制，用户无需设置。共四种计算Node Group模式，用于关联操作和聚集操作时选定计算Node Group。在每一种模式中，优化器有针对性地选定几个候选计算Node Group，然后根据代价，从中为当前算子挑选最佳计算Node Group。

参数类型：字符串

参数单位：无

取值范围：

- optimal：候选计算Node Group列表包含算子操作对象所在的Node Group和由当前用户具有COMPUTE权限的所有Node Group包含的所有DN构成的Node Group。
- query：候选计算Node Group列表包含算子操作对象所在的Node Group和由当前查询涉及的所有基表所在Node Group包含的所有DN构成的Node Group。
- Node Group名（**enable_nodegroup_debug**被设置为off）：候选计算Node Group列表包含算子操作对象所在的Node Group和该指定的Node Group。
- Node Group名（**enable_nodegroup_debug**被设置为on）：候选计算Node Group为指定的Node Group。

默认值：query

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能会导致优化器不能选择更优的计算Node Group，从而影响查询性能。

enable_nodegroup_debug

参数说明：控制优化器在多Node Group环境下，是否使用强制弹性计算。Node Group目前为内部机制，用户无需设置。该参数只在**expected_computing_nodegroup**被设置为具体Node Group时生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示强制将计算弹性到**expected_computing_nodegroup**所指定的Node Group进行计算。
- off：表示不强制使用某个Node Group进行计算。

版权所有 © 华为云计算技术有限公司

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能影响查询性能。

enable_default_index_deduplication

参数说明：设置btree索引默认情况下是否对键值重复的元组进行去重压缩。去重压缩功能对主键索引和唯一索引不生效。在重复键值的索引较多时，去重压缩功能可以有效降低索引占用空间。若创建索引时带有with (deduplication=on/off)语法时，优先根据deduplication参数决定该索引是否使用去重压缩功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示btree索引默认开启索引去重压缩功能。
- off：表示btree索引默认关闭索引去重压缩功能。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，对于非唯一索引、索引键值重复度较高且索引占用空间较大的情况，可以打开。

设置不当的风险与影响：非唯一索引且索引键值重复度很低或者唯一的场景，去重压缩功能会使索引插入性能小幅度劣化。

expr_engine

参数说明：控制表达式的执行方式。

参数类型：枚举类型

参数单位：无

取值范围：

- EXPR_TREE：表示使用表达式的递归引擎。

默认值：EXPR_TREE

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数仅可设置为EXPR_TREE。

设置不当的风险与影响：无。

说明

当前版本暂不支持表达式执行优化，仅能设置为默认值EXPR_TREE。

版权所有 © 华为云计算技术有限公司

sql_result_check

参数说明：控制表达式结果自验证功能。

参数类型：枚举类型

参数单位：无

取值范围：

- off：关闭基于表达式的结果校验功能。
- single_select：在支持进行表达式结果校验的类型范围中，对于数值精度类型列校验失败时，则产生告警；其他类型列的校验失败时，则抛出错误异常

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：使用single_check时，会少量增加查询运行的时间开销。

stream_multiple

参数说明：设置优化器计算Stream算子开销时的加权。在原代价模型的基础上，最终Stream代价将被乘以此加权参数。

说明

此参数仅对Redistribute和Broadcast类型的Stream有效。

参数类型：浮点型

参数单位：无

取值范围：0 ~ DBL_MAX

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，并依据部署环境适时调整。

设置不当的风险与影响：设置不当会导致优化器错估Stream算子开销，从而影响查询性能。

qrw_inlist2join_optmode

参数说明：控制是否使用inlist-to-join查询重写。

参数类型：字符串

参数单位：无

取值范围：

- disable：关闭inlist-to-join查询重写。
- cost_base：基于代价的inlist-to-join查询重写，即基于代价自动评估是否使用inlist-to-join查询重写。

版权所有 © 华为云计算技术有限公司

- rule_base：基于规则的inlist-to-join查询重写，即强制使用inlist-to-join查询重写。
- 任意正整数（1 ~ 2147483647）：inlist-to-join查询重写阈值，即list内元素个数大于该阈值，进行inlist-to-join查询重写。

默认值：cost_base

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，除非有异常场景或者发现优化器选择的计划查询性能低，需要手动调优才使用其他值。

设置不当的风险与影响：该参数会影响优化器是否进行inlist-to-join查询重写，设置不当会导致未生成更优计划。

须知

以下数据类型可以使用inlist-to-join查询重写功能：TINYINT、SMALLINT、INTEGER、BIGINT、NUMERIC、FLOAT4、FLOAT8、BOOLEAN、CHAR、BPCHAR、VARCHAR、NVARCHAR、TEXT、DATE、TIME、TIMEZ、TIMESTAMP、TIMESTAMPTZ、SMALLDATETIME、INTERVAL、TINTERVAL、UNSIGNED TINYINT、UNSIGNED SMALLINT、UNSIGNED INTEGER、UNSIGNED BIGINT。

skew_option

参数说明：控制是否使用优化策略。

参数类型：枚举类型

参数单位：无

取值范围：off、normal、lazy

- off：关闭策略。
- normal：采用激进策略。对于不确定是否出现倾斜的场景，认为存在倾斜，并进行相应优化。
- lazy：采用保守策略。对于不确定是否出现倾斜场景，认为不存在倾斜，不进行优化。

默认值：normal

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：会影响优化器是否针对倾斜进行优化，设置不当会影响计划生成性能或查询执行性能。

default_limit_rows

参数说明：设置生成genericplan的缺省limit估算行数。

参数类型：浮点型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：-100 ~ DBL_MAX。建议取值范围为0.1 ~ 10000，超过这个范围可能会导致业务风险。

- 设置为非负数时，表示直接将设置的值作为估算limit的行数。
- 设置为负数时，表示使用百分比的形式设置默认的估算值，负数转换为默认百分比，即-5代表5%。

默认值：-10

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能影响带有limit算子genericplan查询的性能。

check_implicit_conversions

参数说明：控制对查询中有隐式类型转换的索引列是否会生成候选索引路径进行检查。关于该参数的使用场景请参见《SQL调优指南》中“检查隐式转换的性能问题”章节。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示对查询中有隐式类型转换的索引列会生成候选索引路径进行检查。
- off：表示不进行相关检查。

默认值：off

说明

- 将该参数设置为on时，需要同时将参数**enable_fast_query_shipping**设置为off，检查索引列的隐式数据类型转换的识别机制才会生效。
- 该参数用于检查隐式转换性能问题，在正式生产环境中请关闭该参数。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：不进行检查可能导致查询语句中正则表达式无法按预期匹配。

cost_weight_index

参数说明：设置index_scan的代价权重。

参数类型：浮点型

参数单位：无

取值范围：1e-6 ~ 1e+10

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：会影响优化器是否选择index_scan，设置不当可能影响优化器选择非最优计划从而影响查询性能。

try_vector_engine_strategy

参数说明：设置行存表选择向量化执行引擎的策略。通过设置该参数，可以使包含行存表的查询转换为向量化的执行计划执行计算，从而提升类AP场景的复杂查询的执行性能。

参数类型：枚举类型

参数单位：无

取值范围：

- off：表示关闭本功能，即行存表不会转换为向量的执行计划执行。
- force：表示只要查询中不包含向量化引擎不支持的类型或者表达式，若查询的基表为行存表，强制将查询转换为向量化的执行计划执行计算。在这种情况下，针对不同的查询场景可能出现性能下降。
- optimal：表示在force的基础上，由优化器根据查询的复杂度进行选择是否将查询语句转换为向量化的执行计划，尽可能避免转换为向量化的执行计划后出现性能下降。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：会影响优化器是否选择向量化的执行计划，设置不当可能会影响部分场景下查询的性能。

enable_opfusion_reuse

参数说明：控制复用opfusion obj内存特性的开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在满足条件情况下，复用opfusion obj内存。
- off：表示关闭本功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致计划重新生成，查询性能下降。

enable_iud_fusion

参数说明：控制IUD语句优化特性的开关。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示对输入进行类型转换时做优化处理。
- off：表示关闭本功能。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，IUD语句性能需优化时可考虑开启。

设置不当的风险与影响：打开可能会增加优化器的开销，无明显风险。

enable_cached_operator

参数说明：控制算子执行态缓存特性的开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许缓存执行算子。
- off：表示关闭本功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭则无法对算子执行态进行缓存，从而影响查询性能。

enable_smp_partitionwise

参数说明：控制是否在SMP场景下，开启Partition-wise Join的能力。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在SMP场景下，开启Partition-wise Join的能力。
- off：表示在SMP场景下，不开启Partition-wise Join的能力。

默认值：

- on：新安装的数据库的默认值。
- off：该参数自505.2.0版本添加，从505.2.0版本之前的数据库升级后此参数为off。升级前若存在该参数并且设置了值，升级后默认值不变。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：可能影响SMP场景下数据查询操作的性能。

版权所有 © 华为云计算技术有限公司

force_smp_partitionwise_scan

参数说明：在SMP场景下，打开Partition-wise Join开关enable_smp_partitionwise后，是否在全局范围内开启Partition-wise Scan的并行扫描方式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在SMP场景下，开启Partition-wise Scan的并行扫描方式。
- off：表示在SMP场景下，不开启Partition-wise Scan的并行扫描方式。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：可能影响SMP场景下读取分区表数据的性能。

enable_nestagg

参数说明：开启嵌套聚集功能，嵌套聚集为形如sum(min(col))的双重聚集语法。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启嵌套聚集功能。
- off：表示关闭嵌套聚集功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：影响嵌套聚集功能的使用。

enable_index_skip_scan

参数说明：控制索引跳跃扫描特性的开关。若关闭该特性，当用户使用组合索引查询且前导列为空时，会退化为全表扫描。若开启该特性，当用户使用组合索引查询且前导列为空时，数据库会根据NDV（Number of Distinct Values，表示某一列的Distinct统计值）比例决定是否走索引跳跃扫描。NDV的比例越低，索引跳跃扫描优势越大，当NDV的比例较高时，索引跳跃扫描的代价可能比全表扫描的代价更大，从而导致数据库还是选择全表扫描。

版权所有 © 华为云计算技术有限公司

说明

- 不支持间隔前导列，例如在表t1的(a,b,c,d)列创建索引后，使用“SELECT * FROM t1 WHERE b = xxx AND d = xxx”语句进行查询的情况。
- 不支持前导列为范围的情况。
- 不支持表达式索引。
- 仅支持RCR版本的btree、ubtree索引。
- 仅支持IndexScan和IndexOnlyScan，不支持BitmapIndexScan。
- 由于SQL BYPASS当前停止演进，该特性暂不适配SQL BYPASS。
- 不支持开启SMP。
- 不支持hashbucket表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启索引跳跃扫描特性。
- off：表示关闭索引跳跃扫描特性。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭索引跳跃扫描特性，当用户使用组合索引查询且前导列为空时，会无法使用索引扫描，从而降低查询效率。

enable_gsi_stream_dml

参数说明：控制具有全局二级索引的基表DML操作是否开启STREAM优化特性。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示对于具有全局二级索引的基表的DML操作，开启STREAM优化，即允许生成STREAM计划。
- off：表示对于具有全局二级索引的基表的DML操作，关闭STREAM优化，即不允许生成STREAM计划。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会带来性能劣化。

版权所有 © 华为云计算技术有限公司

enable_partition_statistic

参数说明：控制在优化器阶段，是否使用分区级统计信息。可以使用分区级统计信息的具体场景为：在cplan中，查询涉及到的分区表被静态剪枝到单个分区，包括一级分区表剪枝到单个一级分区。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示可以使用分区级统计信息。
- off：表示不可以使用分区级统计信息。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：影响分区表被静态剪枝到单个分区时的统计信息，进而影响计划的生成。

dist_query_adaptive_mode

参数说明：用于控制分布式自适应计划所选择的模式，其取值范围为auto、plan、sql、off，默认值为auto。当参数值不为off时，会进行Remote Query算子的代价计算。

参数类型：枚举类型

参数单位：无

取值范围：

- auto：分布式计划自适应选择，基于代价或规则选择分布式计划。当规则存在时（设置参数dist_query_adaptive_rows不为0），其优先级高于基于代价的选择。
- plan：强制执行下发计划的分布式计划。
- sql：强制执行下发语句的分布式计划。
- off：关闭分布式计划自适应选择。

默认值：

- auto：新安装的数据库的默认值。
- off：该参数自507.0.0版本引入，从其他版本升级后该参数为off；若升级前存在该参数并且已设置取值，升级后将保留原参数值。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

版权所有 © 华为云计算技术有限公司

说明

分布式计划自适应选择流程如下：首先生成下发计划的分布式计划（Stream计划），通过分析该Stream计划的特征判断是否触发自适应选择分布式计划功能，若需启用自适应优化，则重新生成下发语句的分布式计划。

- 基于代价的分布式计划自适应约束：
  - 仅SELECT语句支持自适应选择分布式计划。
  - 语句中包含Stream的hint时，该计划无法启用自适应分布式优化。
  - 当Stream计划中仅包含Stream Gather算子且无其他Stream算子时，该计划无法启用自适应分布式优化。
  - 当Stream计划是SMP计划时，该计划无法启用自适应分布式优化。
  - 当Stream计划不包含Join操作且无相关子链接时，该计划无法启用自适应分布式优化。
  - 当Stream计划代价相对较小时，该计划无法启用自适应分布式优化。
  - 当CN节点内存或CPU使用率高于80%时，通过分布式自适应计划生成的下发语句的分布式计划，如果是非参数化路径计划，系统会回退至Stream计划。该约束受参数opt_behavior_knob选项DIST_QUERY_ADAPTIVE_FALLBACK控制，默认开启。
- 基于规则的分布式计划自适应约束：
  - 仅SELECT语句支持自适应选择分布式计划。
  - 语句中包含Stream的hint时，该计划无法启用自适应分布式优化。
  - 当Stream计划中仅包含Stream Gather算子且无其他Stream算子时，该计划无法启用自适应分布式优化。
  - 当Stream计划是SMP计划时，该计划无法启用自适应分布式优化。

dist_query_adaptive_rows

参数说明：基于Stream算子行数大小选择分布式计划规则。所有Stream算子（除Stream Gather算子）的总行数小于等于该参数值时，选择下发语句的分布式计划；大于该参数值时，选择下发计划的分布式计划。

参数类型：整型

参数单位：无

取值范围：[0, 2147483647]

默认值：0（表示禁用该规则）。该参数自507.0.0版本引入，从其他版本升级后该参数为0；若升级前存在该参数并且已设置取值，升级后将保留原参数值。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议在DN节点数量较多的场景下使用。使用时通过业务数据规模调整，通常建议设置不超过500。

设置不当的风险与影响：该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

说明

Stream算子要和所有DN节点建立连接传输数据，在DN节点数量较多、高并发的业务场景下易引发数据通信瓶颈，因此该参数建议在集群主DN节点数量大于8个且并发量大于100的业务中使用，使用前需要根据业务实际情况测试并调整参数取值。

版权所有 © 华为云计算技术有限公司

enable_constraint_optimization

参数说明：控制是否启用基于信息性约束的查询优化功能，该功能开启时，优化器能够利用表上的约束信息生成更高效的查询执行计划。主要场景为：外部数据源扫描优化、冗余非空约束消除、包含唯一性约束的扫描与连接优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启基于信息性约束的查询优化功能。
- off：表示关闭基于信息性约束的查询优化功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能会影响部分场景下查询的性能。

6.3.8.2 优化器开销常量

介绍优化器开销常量。这里描述的开销可以按照任意标准度量。只关心其相对值，因此以相同的系数缩放它们将不会对优化器的选择产生任何影响。缺省时，以抓取顺序页的开销为基本单位。也就是说将seq_page_cost设为1.0，同时其他开销参数以该参数为基准设置。也可以使用其他基准，比如以毫秒计的实际执行时间。

seq_page_cost

参数说明：设置优化器计算一次顺序磁盘页面抓取的开销。

参数类型：浮点型

参数单位：无

取值范围：0 ~ DBL_MAX

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数影响一次查询中对顺序抓取磁盘页面的性能开销评估，错误地设置该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

random_page_cost

参数说明：设置优化器计算一次非顺序抓取磁盘页面的开销。

参数类型：浮点型

参数单位：无

取值范围：0.1 ~ DBL_MAX。建议取值范围为0.1 ~ 10，超过这个范围可能会导致业务风险。

版权所有 © 华为云计算技术有限公司

默认值：4

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：参见说明部分。

设置不当的风险与影响：该参数影响一次查询中对随机抓取磁盘页面的性能开销评估，错误地设置该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

说明

- 虽然服务器允许将random_page_cost设置的比**seq_page_cost**小，但是物理上实际不受影响。如果所有数据库都位于随机访问内存中，两者设置为相等很合理。因为在此种情况下，非顺序抓取页并没有副作用。同样，在缓冲率很高的数据库上，应该相对于CPU参数同时降低这两个值，因为获取内存中的页要比通常情况下开销小很多。
- 对于指定表空间中的表和索引，可以通过设置同名的表空间参数覆盖库级别的该参数值，即表空间级别该参数的优先级高于库级别该参数的优先级。
- 相对于**seq_page_cost**，减少这个值将导致系统更倾向于使用索引扫描，而增加这个值使得索引扫描开销比较高。可以通过同时增加或减少这两个值来调整磁盘I/O相对于CPU的开销。
- 参数值1.1适用于固态硬盘配置，若磁盘为机械硬盘，建议修改为4。

cpu_tuple_cost

参数说明：设置优化器计算在一次查询中处理每一行数据的开销。

参数类型：浮点型

参数单位：无

取值范围：0 ~ DBL_MAX

默认值：0.01

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数影响一次查询中对处理一行数据的性能开销评估，错误地设置该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

cpu_index_tuple_cost

参数说明：设置优化器计算在一次索引扫描中处理每条索引的开销。

参数类型：浮点型

参数单位：无

取值范围：0 ~ DBL_MAX

默认值：0.005

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：该参数影响一次查询中对索引扫描的性能开销评估，错误地设置该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

cpu_operator_cost

参数说明：设置优化器计算一次查询中执行一个操作符或函数的开销。

参数类型：浮点型

参数单位：无

取值范围：0 ~ DBL_MAX

默认值：0.0025

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数影响一次查询中对操作符或函数的性能开销评估，错误地设置该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

effective_cache_size

参数说明：设置优化器在一次单一的查询中可用的磁盘缓冲区的有效大小。

设置这个参数，要考虑GaussDB的共享缓冲区以及内核的磁盘缓冲区，还要考虑预计在不同表之间的并发查询数目，因为它们将共享可用的空间。

这个参数对GaussDB实际运行时分配的共享内存大小没有影响，它只用于计划生成阶段的估算。该数值用磁盘页来计算，通常每个页面是8192字节。

参数类型：整型

参数单位：页面（8kB）

取值范围：1 ~ 2147483647

默认值：

独立部署：

CN：4GB（128核CPU/1024G内存）；2GB（60核CPU/480G内存）；1GB（32核CPU/256G内存，16核CPU/128G内存）；512MB（8核CPU/64G内存）；256MB（4核CPU/32G内存）；128MB（4核CPU/16G内存）

DN：152GB（128核CPU/1024G内存）；70GB（60核CPU/480G内存）；38GB（32核CPU/256G内存）；20GB（16核CPU/128G内存）；8GB（8核CPU/64G内存）；4GB（4核CPU/32G内存）；2GB（4核CPU/16G内存）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为200，表示200个页面，即200*8kB的大小；设置为200MB，带单位，表示200MB的大小。取值如果要带单位，必须为kB、MB、GB。

设置建议：较大的数值使优化器倾向于选择索引扫描，较小的数值使优化器倾向于选择全表扫描。一般情况下，可以设为shared_buffers大小的1/2，较为激进地，可以设为shared_buffers大小的3/4。内存充足的情况下，该值越大性能越优。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：错误地设置该参数会影响优化器对生成计划的选择，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

allocate_mem_cost

参数说明：设置优化器计算Hash Join创建Hash表开辟内存空间所需的开销，供Hash join估算不准时调优使用。

参数类型：浮点型

参数单位：无

取值范围：0 ~ DBL_MAX

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，仅在Hash Join估算不准确时调整该参数。

设置不当的风险与影响：错误地设置该参数可能导致Hash Join估算不准确，部分场景下影响查询性能。若要更改，请谨慎操作，避免因误操作产生意料之外的风险。

6.3.8.3 基因查询优化器

介绍基因查询优化器相关的参数。基因查询优化器（GEQO）是一种启发式的查询规划算法。这个算法减少了对复杂查询规划的时间，而且生成规划的开销有时也小于正常的详尽的查询算法。

geqo

参数说明：控制基因查询优化的使用。

说明

该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：通常情况下在执行过程中不要关闭，**geqo_threshold**参数提供了更精细的控制GEQO的方法。

设置不当的风险与影响：若关闭geqo参数，则当查询语句涉及的表数量较大时，会采用动态规划枚举所有组合，优化器生成计划的开销极大。

版权所有 © 华为云计算技术有限公司

geqo_threshold

参数说明：如果执行语句的数量超过设计的FROM的项数，则会使用基因查询优化来执行查询。

说明

- 对于简单的查询，通常用详尽搜索方法，当涉及多个表的查询的时候，用GEQO可以更好地管理查询。
- 一个FULL OUTER JOIN构造仅作为一个FROM项。

参数类型：整型

参数单位：无

取值范围：2 ~ 2147483647

默认值：12

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，若可以接受动态规划的性能开销，可以调大此值；若较难接受动态规划的开销，可以调小此值。

设置不当的风险与影响：设置过大，可能导致动态规划枚举所有计划的开销难以接受；设置过小，可能损失计划的准确性。

geqo_effort

参数说明：控制GEQO在规划时间和规划质量之间的平衡。

参数类型：整型

参数单位：无

取值范围：1 ~ 10

默认值：5

说明

- geqo_effort实际上并没有直接做任何事情，只是用于计算其他影响GEQO的变量的默认值。也可以手动设置其他参数。
- 比默认值大的数值增加了查询规划的时间，但是也增加了选中有效查询的几率。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，可能导致查询规划的开销过大；设置过小，可能导致生成的查询计划质量较差。

geqo_pool_size

参数说明：控制GEQO使用池的大小，也就是基因全体中的个体数量。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 2147483647

默认值：0

说明

设置为0，表示使用系统自适应方式，GaussDB会基于**geqo_effort**和表的个数选取合适的值。其余情况至少设置为2，且有用的值一般在100到1000之间。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，可能导致查询规划的开销过大；设置过小，可能导致生成的查询计划质量较差。

geqo_generations

参数说明：控制GEQO使用的算法的迭代次数。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

说明

如果设置为0，则基于**geqo_pool_size**选取合适的值。其余情况至少设置为1，且有用的值介于100和1000之间。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，可能导致查询规划的开销过大；设置过小，可能导致生成的查询计划质量较差。

geqo_selection_bias

参数说明：控制GEQO的选择性偏好，即一个种群中的选择性压力。

参数类型：浮点型

参数单位：无

取值范围：1.5 ~ 2

默认值：2

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置较高可能会导致GEQO算法更快地集中在一些优秀的计划上，而不是探索可行计划中的多样性。这可能会降低算法找到全局最优解的可能性，但可以提高找到可行的较优解的速度。

版权所有 © 华为云计算技术有限公司

geqo_seed

参数说明：控制GEQO使用的随机数生产器的初始化值，用来从顺序连接在一起的查询空间中查找随机路径。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0

说明

不同的值会改变搜索的连接路径，从而影响所找路径的优劣。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：不同值会带来不同的随机性，可能影响连接路径生成的优劣。

6.3.8.4 基于 DSL 的规则引擎

enable_dsl_ruleengine

参数说明：标识是否开启基于DSL的规则引擎，如果开启该功能，优化器则会应用DSL规则对SQL进行等价改写。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭参数，无法使用DSL规则引擎提供的内置规则进行查询改写优化，可能导致生成的查询计划非最优。

dsl_engine_max_iteration

参数说明：用于控制DSL规则引擎在一次SQL改写中应用规则集迭代的最大次数。

参数类型：整型

参数单位：无

取值范围：1 ~ 100

默认值：10

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，可能导致基于DSL的规则引擎开销过大；设置过小，可能导致生成的查询计划非最优。

dsl_rulelist

参数说明：控制会话级DSL规则启用禁用状态，通过规则名称指定需要变更状态的DSL规则，通过on或off状态指示规则的开关状态。

参数类型：字符串

参数单位：无

取值范围：

- 规则名称：通过查询GS_DSL_REWRITERULE系统表的rulename字段获取。
- 开关状态：
  - on表示开启规则。
  - off表示关闭规则。

```
gaussdb=# SHOW dsl_rulelist;
 dsl_rulelist
--------------

(1 row)

gaussdb=# SET dsl_rulelist = 'compute_const:on';
SET
gaussdb=# SHOW dsl_rulelist;
    dsl_rulelist
--------------------
 compute_const:on
(1 row)
```

默认值：’’，表示规则是否开启受GS_DSL_REWRITERULE系统表中的enabled字段控制。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：如需手动设置，需要明确开启或关闭的规则是否符合需求。

设置不当的风险与影响：设置规则为off，可能导致生成的查询计划非最优。

6.3.8.5 其他优化器选项

cost_model_version

参数说明：此参数用来指定优化器代价模型的版本。可以视作一个保护参数，用来禁用最新的优化器代价模型，保持和指定值所对应的版本计划一致，如果和其他GUC控制相同功能，具体生效优先级参见对应GUC参数说明。

参数类型：整型

参数单位：无

取值范围：0、1、2、3、4、5、6、7、8

版权所有 © 华为云计算技术有限公司

- 0：表示使用最新的cost估算模型。当前版本等价于8。
- 1：表示使用原始的cost估算模型。
- 2：表示在1的基础上，使用增强的coalesce表达式估算、hash join代价估算、semi/anti join代价估算。
- 3：表示在2的基础上，使用边界矫正估计器估算NDV，indexscan的hint可以作用于indexonlyscan。
- 4：表示在3的基础上，使用分区级统计信息参与代价估算。
- 5：表示在4的基础上，增强outer join计算Filter的代价估算，使得基于代价的查询重写更加准确；增强计算选择率时对outer join外表的filter条件的计算优化，可用enable_poisson_outer_optimization参数单独控制。
- 6：表示在5的基础上，增加了以下功能：
  - 当使用子查询更新多列时，多个subplan可优化至仅一个，此功能也可通过enable_shared_set_update单独开关。
  - 支持查询重写阶段消除IN非相关子链接中冗余的DISTINCT、GROUP BY条件，消除Exists简单子链接中冗余的LIMIT条件，消除有AGG的表达式相关子链接中冗余的LIMIT条件以及消除子查询中冗余的投影列等优化场景。
  - 支持子链接满足内外表分布列关联条件时生成FQS计划。
  - 支持视图INSERT，UPDATE，DELETE语句生成FQS计划，并且能根据对分布列的简单过滤条件裁剪涉及的DN节点（对于SELECT视图语句，在6之前已支持FQS计划，但不可裁剪DN节点，6之后也支持裁剪DN节点）。
  - 支持查询重写阶段消除sublink中冗余的Left Join/Right Join、消除主键或非空唯一键上的自连接以及将主外键上的外连接转换为内连接等优化场景。
  - 支持查询重写阶段对满足条件的简单IN/ANY相关子链接，在能转Semi Join的场景下，由Semi Join切换为Inner Join连接。
  - 支持使用静态统计信息，可以直接从PG_CLASS系统表中读取行数和页面数，跳过对基表采样环节。此功能也可通过opt_use_static_stats单独开关。
  - 支持查询重写阶段IN相关子链接提升场景，本参数生效时，该功能通过GUC参数rewrite_rule的'pullup_in_related'值单独开关。
  - 支持对OR条件基于代价展开，改写成UNION ALL或UNION等集合操作，本参数生效时，该功能通过GUC参数costbased_rewrite_rule的'enable_or_expand'枚举值单独开关。
  - 对record array in（多列复合类型表达式是否存在于给定数组集合中）条件生成bitmap index scan计划时，因谓词个数过多，可能导致生成的执行计划在全局上并非最优。开启参数后，可优化该类场景，会倾向于保留覆盖更多列的索引。
- 7：表示在6的基础上，增加了以下功能：
  - 支持对于左右操作数均为普通列的等值连接谓词，在路径生成阶段可使用选择率缓存功能加速计算过程，提升优化器性能，该功能也可通过**enable_sel_cache**单独进行控制。
  - 支持对于左右操作数均为普通列或左右操作数为普通列与常量值的=、<>、>、>=、<以及<=等操作符连接的谓词（仅在WHERE后且不包含子查询提升以及join下推的谓词），根据推导规则进行谓词推导消减，简化谓词逻辑并提升查询性能。该功能可以通过**opt_behavior_knob的PREDICATE_INFER_ENHANCE选项**单独控制开关。
  - 支持M-Compatibility模式数据库下，count(DISTINCT *多列*)生成并行执行计划，提升查询性能。

版权所有 © 华为云计算技术有限公司

  - 支持冗余ORDER BY的消除增强，提升查询性能。
  - 支持ORA兼容模式下，复制表START WITH CONNECT BY递归查询FQS下推DN执行。
  - 支持在查询改写阶段使用基于DSL的规则引擎，提升查询性能。引擎默认包含四个场景的查询重写规则，规则详见《开发指南》中“特性使用指导 > 基于DSL的规则引擎 > DSL规则引擎整体流程”章节的“内置规则介绍”表，对应场景一至场景四相关内容。
- 8：表示在7的基础上，增加了以下功能：
  - 支持部分场景（xmlagg不包含ORDER BY子句、xmlagg包含ORDER BY子句且同层查询存在GROUP BY语句以及xmlagg位于相关子查询内部等场景）将聚集函数xmlagg下推到DN节点执行，提升查询性能。
  - 支持在507.0.0版本新增DSL查询改写规则，提升查询性能。新增规则详见《开发指南》中“特性使用指导 > 基于DSL的规则引擎 > DSL规则引擎整体流程”章节的“内置规则介绍”表，对应场景五至场景九相关内容。
  - 支持Partition Wise Join支持更多场景，优化查询性能。新增能力详见《开发指南》中“特性使用指导 > 分区表 > 分区表查询优化 > Partition-wise Join”。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：数据库升级时，建议和升级前版本保持一致；新安装环境时，建议设置为默认值。

设置不当的风险与影响：改变此参数，可能会导致很多SQL计划的改变，因此修改前请谨慎评估。

enable_fast_query_shipping

参数说明：控制查询优化器是否使用分布式框架。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示执行计划在CN和DN上各自生成。
- off：表示使用分布式框架，即执行计划在CN上生成，然后发送到DN中执行。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭该参数，可能导致生成的计划不能更高效地利用分布式集群性能，建议打开该参数。

enable_trigger_shipping

参数说明：控制触发器场景是否允许将触发器下推到DN执行。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示允许将触发器下推到DN执行。
- off：表示不允许将触发器下推到DN执行，在CN执行。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭该参数，会将性能负载集中在CN节点，不能最大化利用集群性能，建议打开该参数。

enable_remotejoin

参数说明：控制是否允许将连接操作计划下推到DN执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许将连接操作计划下推到DN执行。
- off：表示不允许将连接操作计划下推到DN执行。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果参与JOIN的表数据量很大，这些数据已经根据JOIN条件分散在不同的数据节点上，并且数据节点具备足够的计算资源，可以在本地高效地完成JOIN操作，那么关闭该参数将可能会带来查询性能的下降。

enable_remotegroup

参数说明：控制是否允许将group by与aggregates执行计划下推到DN执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许将group by与aggregates执行计划下推到DN执行。
- off：表示不允许将group by与aggregates执行计划下推到DN执行。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果查询涉及到大量数据的分组，并且数据分布在多个数据节点上，关闭该参数可能会带来查询性能的下降。

版权所有 © 华为云计算技术有限公司

enable_remotelimit

参数说明：控制是否允许将LIMIT子句执行计划下推到DN执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许将LIMIT子句执行计划下推到DN执行。
- off：表示不允许将LIMIT子句执行计划下推到DN执行。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果查询涉及到对大数据量的快速裁剪，并且数据分布在多个节点上，关闭此参数可能会降低查询性能。

enable_remotesort

参数说明：控制是否允许将ORDER BY子句操作计划下推到DN执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许将ORDER BY子句操作计划下推到DN执行。
- off：表示不允许将ORDER BY子句操作计划下推到DN执行。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果查询涉及到大量数据的排序，并且数据分布在多个数据节点上，关闭该参数可能会带来性能的下降。

enable_rep_table_strict_shipping

参数说明：进行复制表DML操作时，控制在复制表的下推充要条件判断失败时的行为。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进行复制表DML操作时，如果对复制表的下推充要条件判断失败，则禁止下推。
- off：表示进行复制表DML操作时，如果对复制表的下推充要条件判断失败，不禁止下推，仅在日志中打印提示信息。

版权所有 © 华为云计算技术有限公司

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_csqual_pushdown

参数说明：设置进行查询时，是否要将过滤条件下推，进行Rough Check。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进行查询时，将过滤条件下推，进行Rough Check。
- off：表示进行查询时，不将过滤条件下推，进行Rough Check。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果查询涉及大量数据的过滤，并且这些数据分布在不同的节点上，关闭过滤条件下推可能会带来性能的下降。

explain_dna_file

参数说明：当**explain_perf_mode**为run时，指定导出的目标文件。

参数类型：字符串

参数单位：无

取值范围：绝对路径加上.csv格式的文件名。

默认值：“ ”

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，如有需要根据业务场景进行设置。

设置不当的风险与影响：可能会带来文件写入的开销。

explain_perf_mode

参数说明：此参数用来指定explain的显示格式。

参数类型：枚举类型

参数单位：无

取值范围：normal、pretty、summary、run

- normal：代表使用默认的打印格式。
- pretty：代表使用GaussDB改进后的新显示格式。新的格式层次清晰，计划包含了plan node id，性能分析简单直接。

版权所有 © 华为云计算技术有限公司

- summary：是在pretty的基础上增加了对打印信息的分析。
- run：在summary的基础上，将统计的信息输出到csv格式的文件中，以便于进一步分析。

默认值：pretty

说明

pretty模式当前只支持包含stream算子的计划，不支持下发语句到DN节点的计划。因此显示格式会受**enable_stream_operator**参数影响，当**enable_stream_operator**设置为off时无法生成包含stream算子的计划。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，如有需要可另行设置。

设置不当的风险与影响：不同格式会影响explain提供的信息量。

analysis_options

参数说明：通过开启对应选项使用相应的定位功能，包括数据校验，性能统计等，具体参见取值范围中的描述。

参数类型：字符串

参数单位：无

取值范围：

该参数每次设置将会通过集合操作的方式对当前值做修改：

```
gaussdb=# show analysis_options;
                    analysis_options
------------------------------------------------------------
 ALL,on(),off(LLVM_COMPILE,HASH_CONFLICT,STREAM_DATA_CHECK)
(1 row)

gaussdb=# SET analysis_options = 'on(LLVM_COMPILE)';
SET
gaussdb=# show analysis_options;
                   analysis_options
-------------------------------------------------------------
 ALL,on(LLVM_COMPILE),off(HASH_CONFLICT,STREAM_DATA_CHECK)
(1 row)

gaussdb=# SET analysis_options = 'on(HASH_CONFLICT)';
SET
gaussdb=# show analysis_options;
                   analysis_options
--------------------------------------------------------------
 ALL,on(LLVM_COMPILE,HASH_CONFLICT),off(STREAM_DATA_CHECK)
(1 row)
gaussdb=# SET analysis_options = 'off(ALL)';
SET
gaussdb=# show analysis_options;
                    analysis_options
------------------------------------------------------------
 ALL,on(),off(LLVM_COMPILE,HASH_CONFLICT,STREAM_DATA_CHECK)
(1 row)
```

- LLVM_COMPILE：表示在explain performance显示界面中显示每个线程的codegen编译时间。
- HASH_CONFLICT：表示在DN进程的gs_log目录的log日志中显示hash表的统计信息，包括hash表大小、hash链长、hash冲突情况。

版权所有 © 华为云计算技术有限公司

- STREAM_DATA_CHECK：表示对网络传输前后的数据进行CRC校验。

默认值：off(ALL)，表示不开启任何定位功能。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如需手动设置，需要明确预期显示的结果是否符合需求。

cost_param

参数说明：该参数用于控制在特定的客户场景中，使用不同的估算方法，使得代价模型估算更准确。通过将参数修改为不同的值，选择开启不同的方法。此参数可以同时控制多种方法，每一种方法由不同的数字编号控制，参数取值与方法对应的数字编号做“与”操作的结果不为0时，表示该方法开启。

- 当cost_param & 1不为0，表示对于求不等值连接选择率时选择一种改良机制，此方法在自连接（两个相同的表之间连接）的估算中更加准确。当前版本已弃用cost_param & 1不为0时的路径，默认选择更优的估算公式。
- 当cost_param & 2不为0，表示求多个过滤条件（Filter）的选择率时，选择最小的作为总的选择率，而非两者乘积，此方法在过滤条件的列之间关联性较强时估算更加准确。
- 当cost_param & 4不为0，表示在进行stream节点估算时，选用调试模型，该模型不推荐用户使用。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在明确需要调整选择率的场景后，设置对应参数值。

设置不当的风险与影响：设置不当可能会导致最终的代价估算不符合预期，建议在充分测试后确认应当设置的参数值。

enable_partitionwise

参数说明：分区表连接操作是否选择智能算法。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示选择智能算法。
- off：表示不选择智能算法。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：打开前确认是否开启SMP，否则可能影响非SMP场景下分区表Join操作的性能。

partition_page_estimation

参数说明：分区表页面是否通过剪枝结果进行页面估算优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用剪枝结果进行页面估算优化。
- off：表示不使用剪枝结果进行页面估算优化。

默认值：

- on：新安装的数据库的默认值。
- off：从506.0.0之前版本升级到当前版本，此参数默认值为off；升级前若存在该参数，无论如何设置，升级后默认值均为off。

  在PDB场景内，若未设置该参数，则继承来自全局的设置。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_partition_pseudo_predicate

参数说明：在指定分区查询场景下，是否使用伪谓词改写的方式计算指定分区查询的选择率。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用伪谓词改写。
- off：表示不使用伪谓词改写。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

partition_iterator_elimination

参数说明：分区表在分区剪枝结果为一个分区时，是否消除分区迭代算子来提升执行效率。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示消除分区迭代算子。
- off：表示不消除分区迭代算子。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数可能会导致查询性能下降。

rewrite_rule

参数说明：标识开启的可选查询重写规则。有部分查询重写规则是可选的，开启它们并不能总是对查询效率有提升效果。在特定的客户场景中，通过此GUC参数对查询重写规则进行设置，使得查询效率最优。

此参数可以控制查询重写规则的组合。比如有多个重写规则：rule1、rule2、rule3、rule4，可以设置：

```
set rewrite_rule=rule1;          --启用查询重写规则rule1
set rewrite_rule=rule2,rule3;    --启用查询重写规则rule2和rule3
set rewrite_rule=none;           --关闭所有可选查询重写规则
```

参数类型：枚举类型

参数单位：无

取值范围：

- none：不使用任何可选查询重写规则。
- lazyagg：使用Lazy Agg查询重写规则（消除子查询中的聚集运算）。
- magicset：使用Magic Set查询重写规则（将带有聚集算子的子查询提前和主查询进行关联，减少子链接的重复扫描）。
- partialpush：使用Partial Push查询重写规则（对于不可下推的语句，下推部分子查询到DN执行，剩余不下推的部分在CN执行）。
- uniquecheck：使用Unique Check查询重写规则（提升目标列中无agg的子查询语句，在执行时检查返回行数是否为1行）。
- disablerep：使用Disable Replicate查询重写规则（由于复制表提升之后可能劣化，开启此规则之后，禁止复制表的子查询提升）。
- intargetlist：使用In Target List查询重写规则（提升目标列中的子查询）。
- predpushnormal：使用Predicate Push查询重写规则（下推谓词条件到子查询中，可能会添加BROADCAST算子来支持分布式执行），当开启predpush后不会生成向量化计划。
- predpushforce：使用Predicate Push查询重写规则（下推谓词条件到子查询中，尽可能地利用索引加速），当开启predpush后不会生成向量化计划。
- predpush：在predpushnormal和predpushforce中根据代价选择最优计划，当开启predpush后不会生成向量化计划。
- disable_pullup_expr_sublink：禁止优化器将expr_sublink类型的子链接提升，关于sublink的分类和提升原理详见《SQL调优指南》中“典型SQL调优点 > 子查询调优”章节。

版权所有 © 华为云计算技术有限公司

- enable_sublink_pullup_enhanced：使用增强后的sublink查询重写规则，包括where、having子句的非相关子链接提升和winmagic重写优化。
- disable_pullup_not_in_sublink：禁止优化器对not in相关的子链接进行提升，关于sublink的分类和提升原理详见《SQL调优指南》中“典型SQL调优点 > 子查询调优”章节。
- disable_rownum_pushdown：禁止将父查询中的rownum过滤条件下推到子查询中。
- disable_windowagg_pushdown：禁止将父查询中的窗口函数的过滤条件下推到子查询中。
- cse_rewrite_opt：使用公共表达式CSE，将Having子查询中的公共表达式替换成windowAgg。
- groupby_pushdown_subquery：将Group By子句与聚集函数下推到子查询中。
- enable_sublink_pullup_rownum：允许SQL语句中有ROWNUM伪列的情况下优化器进行子链接提升。
- disable_in_sublink_reduce_distinct：禁止消除IN/ANY非相关子链接提升场景下冗余的DISTINCT和GROUP BY条件。
- disable_sublink_reduce_limit：禁止消除满足约束条件的子链接中冗余的LIMIT条件，如：简单的Exists子链接中不影响查询结果的LIMIT条件等。
- pullup_in_related：允许对满足条件的简单IN/ANY或NOT IN相关子链接提升为Semi Join或Anti Join连接。
- disable_reduce_unused_targetlist: 禁止消除子查询投影列中冗余的函数计算处理。
- disable_sublink_outer_join_elimination: 禁止对满足条件的sublink中冗余的left/right join进行消除。
- disable_fk_pk_outer_join_to_inner_join: 禁止主外键上的外连接转换为内连接。
- disable_self_join_elimination: 禁止消除主键或非空唯一键上的自连接。
- enable_corr_any_sublink_semijoin_to_innerjoin：允许在查询重写阶段对满足条件的简单IN/ANY相关子链接，在能转Semi Join的场景下，由Semi Join切换为Inner Join连接。
- disable_reduce_orderby：禁止消除满足条件的冗余ORDER BY语句。
- enable_not_in_sublink_reduce_distinct：允许消除NOT IN非相关子链接提升场景下冗余的DISTINCT和GROUP BY条件。

默认值：magicset,groupby_pushdown_subquery,pullup_in_related。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。如有其他改写必要，建议明确查询改写场景后，仅打开对应重写规则。

设置不当的风险与影响：可能会导致某些场景改写的语句代价并非最优，建议充分测试后使用对应规则。

版权所有 © 华为云计算技术有限公司

说明

- 当cost_model_version>=6或为0且GUC参数中未设置disable_in_sublink_reduce_distinct时，IN非相关子链接消除冗余DISTINCT和GROUP BY的相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中未设置disable_sublink_reduce_limit时，Exists子链接消除冗余LIMIT的相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中未设置disable_reduce_unused_targetlist时，子查询消除冗余投影列的相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中设置pullup_in_related时，IN相关子链接提升相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中未设置disable_sublink_outer_join_elimination时，sublink中冗余的left/right join消除相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中未设置disable_fk_pk_outer_join_to_inner_join时，主外键上的外连接转内连接相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中未设置disable_self_join_elimination时，主键或非空唯一键上的自连接消除相关功能生效。
- 当cost_model_version>=6或为0且GUC参数中设置enable_corr_any_sublink_semijoin_to_innerjoin时，简单IN/ANY相关子链接由Semi Join切换为Inner Join连接生效。
- 当cost_model_version>=6或为0且GUC参数中设置enable_not_in_sublink_reduce_distinct时，NOT IN非相关子链接消除冗余DISTINCT和GROUP BY的相关功能生效。

costbased_rewrite_rule

参数说明：该参数用于标识对指定的查询改写规则开启基于代价的评估策略。部分查询改写规则支持基于代价评估是否对SQL进行改写，从而使内核生成更好的查询计划，提升SQL执行效率。

该参数与GUC参数rewrite_rule互斥，如果同时设置了costbased_rewrite_rule和rewrite_rule，则rewrite_rule的相关规则配置不生效。

该参数可以控制多个基于代价的查询重写规则进行组合设置，且参数间无互斥关系。比如有多个重写规则：rule1、rule2、rule3、rule4，可以设置：

```
set costbased_rewrite_rule=rule1;          --启用查询重写规则rule1基于代价的评估策略
set costbased_rewrite_rule=rule2,rule3;    --启用查询重写规则rule2和rule3基于代价的评估策略
set costbased_rewrite_rule=none;           --关闭所有可选查询重写规则基于代价的评估策略
```

参数类型：字符串

参数单位：无

取值范围：

- none：不使用任何基于代价的查询改写策略。
- pullup_subquery：简单子查询展开改写规则开启使用基于代价的查询改写策略。
- pullup_sublink_any_exists：单个或AND条件里的ANY非相关子链接和[NOT] EXISTS相关子链接改写规则开启使用基于代价的查询改写策略。
- pullup_not_in_sublink：单个或AND条件里的NOT IN非相关子链接改写规则开启使用基于代价的查询改写策略。该选项与GUC参数rewrite_rule的disable_pullup_not_in_sublink选项互斥，当开启该选项时，rewrite_rule的disable_pullup_not_in_sublink选项功能不生效。
- pullup_expr_sublink：表达式子链接以及OR条件中的ANY非相关子链接和[NOT] EXISTS相关子链接改写场景生效。该选项与GUC参数rewrite_rule的

版权所有 © 华为云计算技术有限公司

disable_pullup_expr_sublink、enable_sublink_pullup_enhanced、magicset选项互斥，当开启该选项时rewrite_rule的相关选项功能不生效。

- intargetlist：TargetList中的相关表达式子链接改写规则开启使用基于代价的查询改写策略。该选项与GUC参数rewrite_rule的intargetlist、magicset选项互斥，当开启该选项时，rewrite_rule的intargetlist、magicset选项功能不生效。
- enable_sublink_pullup_enhanced：表达式子链接改写功能增强场景使用基于代价的查询改写策略。该选项受pullup_expr_sublink选项影响，AND条件中的表达式子链接改写场景需开启pullup_expr_sublink选项，该选项功能才生效。且与GUC参数rewrite_rule的enable_sublink_pullup_enhanced选项互斥，当开启该选项时，rewrite_rule的enable_sublink_pullup_enhanced选项功能不生效。
- enable_or_expand：对OR条件基于代价展开，改写成UNION ALL或 UNION等集合操作。

默认值：

- 新安装的数据库默认值："intargetlist,pullup_expr_sublink,pullup_not_in_sublink,enable_sublink_pullup_enhanced,enable_or_expand"。
- 从历史版本升级到当前版本时的默认值：该参数自505.2.0版本添加，505.2.0版本之前的数据库升级到当前版本后此参数为none。自505.2.0版本升级到其他更高版本（如505.2.1），该GUC默认值与升级前保持一致。

说明

- 相关参数使用场景案例请参见《SQL调优指南》中“实际调优案例 > 案例：调整基于代价的查询重写GUC参数costbased_rewrite_rule”章节。
- 当cost_model_version>=6或为0且GUC参数中设置enable_or_expand时，OR条件基于代价展开相关功能生效。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

对于简单子查询展开（受pullup_subquery参数控制）、ANY非相关子链接和[NOT] EXISTS相关子链接（受pullup_sublink_any_exists参数控制）通常按默认规则进行改写生成执行计划更优，因为开启代价改写策略会带来一些性能开销，因此对于此类规则不建议开启基于代价的改写策略。

在优化器查询重写阶段，开启该参数后相比基于规则的改写模型会生成更多候选计划，因此会引入额外的时间性能消耗；该模式下每走一次基于代价的改写策略相比基于规则的改写策略会增加约1倍的计划生成时间，但可能带来更好的执行性能，因此如果SQL使用基于规则改写策略生成的执行计划已经很好，则建议关闭该参数。

设置不当的风险与影响：关闭该参数相关改写规则将退化为基于规则进行改写，缺少代价评估可能导致部分场景下生成的计划较差，影响查询性能。

costbased_rewrite_rule_max_iterations

参数说明：执行的SQL在计划生成阶段时，如果对同一规则中满足做代价评估的条件数超过该参数的取值，则对本次请求中超过阈值的条件关闭基于代价的评估策略，转化为基于规则的改写策略。当查询改写阶段开启基于代价的评估策略后该参数生效。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 1000

默认值：10

说明

- 该参数受GUC参数**costbased_rewrite_rule**影响，当costbased_rewrite_rule值不为none时，该参数生效。
- 如果设置该参数值为0，表示对当前SQL关闭基于代价的改写策略。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：可根据实际业务需求评估调整，当该值设置过大或过小时都可能带来额外的性能开销，影响最终的查询性能。

costbased_rewrite_rule_timeout

参数说明：执行的SQL在计划生成阶段时，如果使用各规则做代价评估的整体时间超过该参数设置的超时时间，则对本次请求后续流程关闭基于代价的评估策略，转化为基于规则的改写策略。当查询改写阶段开启基于代价的评估策略后该参数生效。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 300000

默认值：-1

说明

- 该参数受GUC参数**costbased_rewrite_rule**影响，当costbased_rewrite_rule值不为none时，该参数生效。
- 如果设置该参数值为0，表示对当前SQL关闭基于代价的改写策略。
- 如果设置该参数值为-1，表示对当前SQL关闭超时时间控制。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：可根据实际业务需求评估调整，当该值设置过大或过小时都可能带来额外的性能开销，影响最终的查询性能。

optrbo_max_or_expr_numbers

参数说明：用于设置OR展开允许应用的最大次数。

参数类型：整型

参数单位：无

取值范围：2 ~ 100

默认值：10

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：可根据实际业务需求评估调整，当该值设置过大时,可能会对OR条件较多的查询进行尝试改写，影响最终的查询性能。

enable_pbe_optimization

参数说明：设置优化器是否对以PBE（Parse Bind Execute）形式执行的语句进行查询计划的优化，优化原理是使FQS选择gplan计划。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示优化器将优化PBE语句的查询计划，在FQS下选择gplan。
- off：表示不使用优化。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，在不希望FQS选择gplan时可设置为off。

设置不当的风险与影响：关闭可能会导致某些场景走cplan。

enable_light_proxy

参数说明：设置优化器是否对CN上简单查询的执行进行优化，应用端和内核端字符集不匹配时，该参数不生效，建议建库时将字符集设为UTF8。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示优化器将优化CN上简单查询的执行。
- off：表示不使用优化。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后可能会导致简单查询执行压力集中在CN上。

说明

该参数在存储过程内部通过SET方式设置不生效。

enable_global_plancache

参数说明：设置是否对PBE查询和存储过程中语句的执行计划进行缓存共享，开启该功能可以节省高并发下CN和DN上的内存使用。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示对PBE查询和存储过程中语句的执行计划进行缓存共享。
- off：表示不共享。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。该参数在CN和DN上必须保持一致。

设置不当的风险与影响：CN和DN上该参数设置不一致会使得CN下发给DN的报文不匹配，从而导致报错等问题。开启该参数后，频繁DDL可能导致计划失效，影响CPU开销，进而影响性能。

enable_plancache_mprotect

参数说明：用于控制执行阶段计划写保护开启。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启执行阶段计划写保护。
- off：表示关闭执行阶段计划写保护。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数适用于调试版本，主要用于问题定位、排查及调试。生产环境请按照默认值设置，避免影响正常业务。

设置不当的风险与影响：

- 开启该参数时，数据库内核在生成执行计划后，会严格防护并禁止在执行阶段修改计划。
- 该参数通过系统调用mprotect实现内存保护，开启该参数时，在高负载、频繁或大范围使用修改内存场景下会带来额外耗时。
- 如果内存不足导致mprotect资源不足或者出现修改执行计划的动作，会触发进程保护性重启防护。

disable_gpc_store

参数说明：GUC参数enable_global_plancache为on时，用于阻止后续执行计划存入全局计划缓存。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示后续执行计划不可存入全局计划缓存。

版权所有 © 华为云计算技术有限公司

- off：表示后续计划计划可存入全局计划缓存。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：enable_global_plancache为off，该GUC设置无意义。该参数设置后，后续执行计划不再存入全局共享缓存，可能导致本会话内存开销变大，请谨慎设置。

enable_stream_pbe

参数说明：设置优化器在数据节点上是否对STREAM分布式计划使用缓存执行计划的优化。

说明

- enable_global_plancache = on时，不支持stream pbe优化。
- 需依赖参数enable_stream_operator开启，enable_stream_pbe的优化功能才会生效。
- 由不支持stream pbe的版本升级至支持stream pbe的版本，升级未提交期间不支持stream pbe优化执行方式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示优化器在数据节点上对STREAM分布式计划使用缓存执行计划的优化。
- off：表示不使用优化。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后可能会增加生成计划的开销。

gpc_clean_timeout

参数说明：本参数用于控制不使用的共享计划的保留时间。在开启**enable_global_plancache**的情况下，如果共享计划列表里的计划超过gpc_clean_timeout的时间没有被使用，则会每隔一段时间被清理一次。

参数类型：整型

参数单位：s（秒）

取值范围：300 ~ 86400

默认值：1800（即30min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置过大可能导致GPC占用内存过多，建议充分测试后设置合理值。

enable_gpc_grayrelease_mode

参数说明：分布式集群下，开启GPC需要重启集群。如果需要在不重启集群的情况下开启GPC，需要用enable_gpc_grayrelease_mode控制滚动开启GPC或关闭GPC。

须知

分布式集群上操作方式。

开启GPC：

1. 在所有DN节点开启enable_gpc_grayrelease_mode。
2. 在所有CN节点开启enable_gpc_grayrelease_mode。
3. 开启GPC参数，由于GPC是POSTMASTER参数，需要先reload参数，之后轮询kill节点，使重新拉起的节点上GPC生效。

关闭GPC：

1. 首先确定enable_gpc_grayrelease_mode是on的状态，reload关闭GPC参数，然后轮询kill节点，使重新拉起的节点上GPC生效。
2. 在所有CN节点关闭enable_gpc_grayrelease_mode。
3. 在所有DN节点关闭enable_gpc_grayrelease_mode。

参数类型：布尔型

参数单位：无

取值范围：

- on：滚动开启GPC。
- off：关闭GPC。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数打开会影响pbe报文收发逻辑，需严格按照参数说明指定顺序设置，避免不合理报错。

enable_opfusion

参数说明：控制是否对简单查询进行查询优化。

说明

该开关优化DN的查询性能，可以设置**max_datanode_for_plan**查看一个查询DN的执行计划，如果DN的执行计划中带有[Bypass]标识则代表该查询在该DN可以查询优化。

简单查询限制如下：

- 只支持indexscan和indexonlyscan，且全部WHERE语句的过滤条件都在索引上。

版权所有 © 华为云计算技术有限公司

- 只支持单表增删改查，不支持join、using。
- 只支持行存表，不支持分区表，表不支持有触发器。
- 不支持active sql、QPS等信息统计特性。
- 不支持正在扩容和缩容的表。
- 不支持查询或者修改系统列。
- 只支持简单SELECT语句，例如：
  ```
  SELECT c3 FROM t1 WHERE c1 = ? and c2 =10;
  ```
  仅可以查询目标表的列，c1和c2列为索引列，后边可以是常量或者参数，可以使用 for update。
- 只支持简单INSERT语句，例如：
  ```
  INSERT INTO t1 VALUES (?,10,?);
  ```
  仅支持一个VALUES，VALUES里面的类型可以是常量和参数，不支持returning。
- 只支持简单DELETE语句，例如：
  ```
  DELETE FROM t1 WHERE c1 = ? and c2 = 10;
  ```
  c1和c2列为索引列，后边可以是常量或者参数。
- 只支持简单UPDATE语句，例如：
  ```
  UPDATE t1 SET c3 = c3+? WHERE c1 = ? and c2 = 10;
  ```
  c3列修改的值可以是常量和参数，也可以是一个简单的表达式，c1和c2列为索引列，后边可以是常量或者参数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后将不使用SQLBypass特性。SQLBypass针对简单增删改查语句采用简化的执行路径，以减少条件过滤、投影列计算、算子初始化和执行等带来的执行开销。关闭该参数后，参数说明中给定的简单增删改查语句将使用通用的执行流程，从而影响对应场景下的性能。

sql_beta_feature

参数说明：标识开启的可选SQL引擎Beta特性，其中包括对行数估算、查询等价估算等优化。开启它们可以对特定的场景进行优化，但也可能会导致部分没有被测试覆盖的场景发生性能劣化。在特定的客户场景中，通过此GUC参数对查询重写规则进行设置，使得查询效率最优。

此参数可以控制SQL引擎Beta特性的组合，比如有多个Beta特性：feature1、feature2、feature3、feature4，可以设置：

```
set sql_beta_feature=feature1;          --启用SQL引擎Beta特性feature1
set sql_beta_feature=feature2,feature3;    --启用SQL引擎Beta特性feature2和feature3
set sql_beta_feature=none;          --关闭所有可选SQL引擎Beta特性
```


版权所有 © 华为云计算技术有限公司

参数类型：字符串

参数单位：无

取值范围：

- none：不使用任何Beta优化器特性。
- sel_semi_poisson：使用泊松分布对等值的半连接和反连接选择率进行校准。
- sel_expr_instr：使用字符串匹配的行数估算方法对instr(col, 'const') > 0, = 0, = 1进行更准确的估算。
- param_path_gen：生成更多可能的参数化路径。
- rand_cost_opt：对小数据量表的随机读取代价进行优化。
- param_path_opt：利用表的膨胀系数优化索引analyze信息。
- page_est_opt：优化对表索引analyze信息的relpages估算。
- no_unique_index_first：关闭主键索引扫描路径优先的优化。
- join_sel_with_cast_func：估算join行数的时候支持类型转换函数。
- canonical_pathkey：正则化pathkey生成置前（pathkey：标记数据有序性键值的集合）。

  说明

  该选项打开之后，可能会导致带order by等语句，在有外连接的情况下，输出数据语义和标准不一样。请联系华为工程师再确定是否打开该参数。
- index_cost_with_leaf_pages_only：估算索引代价时考虑索引叶子节点。
- partition_opfusion：支持数据库在运行过程中轻量缓存分区表的一些数据结构，执行处理轻量化。当前版本不生效。
- a_style_coerce：开启Decode类型转换规则兼容ORA，详见**UNION，CASE和相关构造**章节中的“对于case，在ORA兼容模式下的处理”部分内容。
- plpgsql_stream_fetchall：在存储过程中for loop或cursor上执行的sql走stream场景下，开启获取所有tuple结果。
- partition_fdw_on：支持基于分区表创建GaussDB foreign table下的相关SQL。
- predpush_same_level：开启predpush hint控制同层参数化路径的功能。
- disable_bitmap_cost_with_lossy_pages：关闭bitmap路径代价中对lossy pages代价的计算。
- enable_upsert_execute_gplan：pbe场景下，on duplicate key update语句中update子句带有参数时，设置enable_upsert_execute_gplan允许通过gplan执行。
- disable_text_expr_flatten：关闭text和数值类型（numeric, bigint）进行比较运算时自动内联表达式的功能。

默认值：
"sel_semi_poisson,sel_expr_instr,rand_cost_opt,param_path_opt,page_est_opt"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。该参数控制多种查询优化和兼容性行为，部分选项设置存在风险，建议谨慎使用。若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

设置不当的风险与影响：需充分测试，否则可能产生意料之外的风险。

版权所有 © 华为云计算技术有限公司

enable_global_stats

参数说明：标识当前统计信息模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示采用全局统计信息收集模式。
- off：表示收集集群第一个节点的统计信息，此时可能会影响生成查询计划的质量，但信息收集性能较优，建议谨慎设置。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会影响集群环境下查询计划生成质量。

default_statistics_target

参数说明：为没有用ALTER TABLE SET STATISTICS设置字段目标的表设置缺省统计目标。此参数仅影响统计信息的目标采样行数，实际采样行数还会受到内存参数**maintenance_work_mem**的限制。

参数类型：整型

参数单位：无

取值范围：-100 ~ 10000

- 此参数设置为正数时，代表统计信息直方图预期桶的数量，统计信息采样行数为default_statistics_target * 300。
- 此参数设置为负数时，代表使用百分比的形式设置统计目标，负数转换为对应的百分比，即-5代表5%，采样行数为总行数 * 5%。

默认值：100

说明

- 比默认值大的正数数值增加了ANALYZE所需的时间，但是可能会改善优化器的估计质量。
- 调整此参数可能存在性能劣化的风险，如果某个查询劣化，可以考虑：
  1. 恢复默认的统计信息。
  2. 使用plan hint调整到之前的查询计划。详细请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。
- 当此guc参数设置为负数时，如果计算的采样样本数大于等于总数据量的2%，且用户表的数据量小于1600000时，ANALYZE所需时间相比guc参数为默认值的时间会有所增加。
- 当此guc参数设置为负数时，autoanalyze不生效。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在调整此参数之前，应根据具体的工作负载和查询模式进行综合评估。对于数据特征分布不均匀或数据量大的表，可能需要手动调整该参数值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：

- 如果default_statistics_target设置过高，可能会增加 ANALYZE 操作的时间和资源消耗，因为需要收集更多的样本数据来生成统计信息。这可能导致数据库维护的开销增大，特别是在大型表上。
- 如果default_statistics_target 设置过低，可能会降低统计信息的准确性，从而影响查询优化器生成高效查询计划的能力，导致查询性能下降。

default_gsi_statistics_target

参数说明：为全局二级索引设置缺省统计目标。

参数类型：整型

参数单位：无

取值范围：-100 ~ 3000000

- 此参数设置为正数时，代表统计信息的样本数量。
- 此参数设置为负数时，代表使用百分比的形式设置统计目标，负数转换为对应的百分比，即-5代表5%。

默认值：0

说明

- 默认值为0表示采用内部默认计算采样率的方法进行采样。
- 调整此参数可能存在性能劣化的风险，如果某个查询劣化，可以考虑：
  1. 恢复默认的统计信息。
  2. 使用plan hint来调整到之前的查询计划。详细请参见《SQL调优指南》中“使用Plan Hint进行调优”章节。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在调整此参数之前，应根据具体的工作负载和查询模式进行综合评估。对于数据特征分布不均匀或数据量大的表，可能需要手动调整该参数值。

设置不当的风险与影响：

- 如果default_gsi_statistics_target设置过高，可能会增加ANALYZE操作的时间和资源消耗，因为需要收集更多的样本数据来生成统计信息。这可能导致数据库维护的开销增大。
- 如果default_gsi_statistics_target设置过低，可能会降低统计信息的准确性，从而影响查询优化器生成高效查询计划的能力，导致查询性能下降。

opt_mcv_buckets_global

参数说明：为未通过“ALTER TABLE <table> ALTER COLUMN <attr> SET (OPT_MCV_BUCKETS = <val>)”设置的字段，设置高频值（MCV）统计信息的最大存储桶数量。该参数仅指定高频值存储个数的上限，实际高频值个数由实际采样数据分布决定。该参数可在PDB级别设置。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 10000。设置为0时，表示自动根据采样行数计算高频值（MCV）个数。

默认值：

- 新安装的数据库：100。
- 从历史版本升级上来的数据库：若历史版本不支持该GUC参数，升级后的默认值为0；若历史版本支持该GUC参数，升级后默认值保留原设置。
- 在PDB场景内，若未设置该参数，则继承来自全局的设置。

说明

- 参数修改后仅对后续ANALYZE生效，存量表旧统计信息不会自动刷新，需手动执行“ANALYZE 表名;”，重新采集统计或等待自动触发统计信息采集。
- 若单表个别列需要更高桶数上限，可通过列级自定义统计参数单独放开，语法为：
  ```
  ALTER TABLE <table> ALTER COLUMN <attr> SET (OPT_MCV_BUCKETS = <val>)
  ```
  设置后该列不受全局参数opt_mcv_buckets_global约束。
- 升级观察期期间设置此参数无效。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。在调整此参数之前，应根据具体的工作负载和查询模式进行综合评估。对于数据分布里高频值特别多的表，可能需要手动调整该参数值。

设置不当的风险与影响：

- 如果opt_mcv_buckets_global设置得过高，会收集更多的高频值（MCV），可能会增加ANALYZE操作的时间和资源消耗，从而导致数据库维护开销增大，特别是对于大型表。
- 如果opt_mcv_buckets_global设置得过低，可能会降低代价估算的准确性，从而影响查询优化器生成高效查询计划的能力，导致查询性能下降。

auto_statistic_ext_columns

参数说明：表示会根据数据表上的组合索引的前auto_statistic_ext_columns列，收集多列统计信息。例如：某组合索引为(a,b,c,d,e)，此GUC参数设为3，则会在组合列(a,b)、(a,b,c)上产生多列统计信息。多列统计信息可以在组合条件查询时，使优化器更准确地估计基数。

版权所有 © 华为云计算技术有限公司

说明

- 系统表不会生效。
- 组合列中的所有列的类型，都支持比较函数'='和'<'，统计信息才会生效。
- 索引中的系统伪列，如tableoid、ctid不会被收集。
- 默认会收集distinct值、不带NULL的高频值、带NULL的高频值。自动创建的多列统计的信息不收集智能统计信息，只收集高频值和带NULL的高频值。
- 如果创建某多列统计信息的索引被删除，并且没有其它索引包含该多列组合，在下一次analyze的时候，该多列统计信息会被删除。
- 如果该参数由大变小，新的索引会依据该参数产生多列统计信息，而已经产生的超过该参数长度的多列统计信息，不会被删除。
- 如果用户希望禁用某一特定组合的多列统计信息，又希望使用其它多列统计信息，可以不修改该参数，而使用DDL命令'ALTER TABLE tablename disable statistics ((column list))'的方式，禁用特定的多列组合。
- 自动创建多列统计信息会跳过相关性不强的多列。如果单列的NDV（去重后的数量）乘积与多列的NDV达到80%以上相接近，则判定该多列相关性不强。手动创建的多列统计信息或自动创建的多列统计信息NDV为-1不需判断相关性。
- 如果分布式下采用百分比采样的方式，则CN上不会自动创建多列统计信息。
- 自动创建多列统计信息的功能打开后，因增加了计算统计信息阶段的开销，当组合索引数量较多时，analyze的性能会较之前有轻微下降。

参数类型：整型

参数单位：无

取值范围：1 ~ 4。1表示不会自动收集多列统计信息。

默认值：

- 新安装的数据库：4。
- 从506.1之前版本升级的数据库：升级后默认值为1。
- 从506.1及之后版本升级的数据库：升级后的参数值保留原设置。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在调整该参数值时，可考虑以下因素：

- 表的大小和数据特性，例如多列相关性。
- 查询模式，特别是查询中涉及的列和索引。
- 系统可用的资源，例如CPU、内存和存储空间。

设置不当的风险与影响：

- 设置的值过高可能会使得ANALYZE操作变得频繁，增加了数据库的维护开销，尤其是在大型表上。
- 如果设置的值过低，可能无法充分捕捉到表中列之间的相关性，导致查询优化器不能生成最优的查询计划。

constraint_exclusion

参数说明：控制查询优化器使用表约束查询的优化。

参数类型：枚举类型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示检查所有表的约束。
- off：表示不检查约束。
- partition：表示只检查继承的子表和UNION ALL子查询。

默认值：partition

说明

- 当constraint_exclusion为on时，优化器用查询条件和表的CHECK约束比较，并且在查询条件和约束冲突的时候忽略对表的扫描。
- 目前，constraint_exclusion默认被打开，通常用来实现表分区。为所有的表打开该参数时，对于简单的查询强加了额外的规划，并且对简单查询没有什么好处。如果不用分区表，可以关掉该参数。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。如果业务查询中经常使用到表的约束，并且这些约束有助于优化查询，则保持该参数开启可能是有益的。反之，如果这些约束对于查询优化没有太大帮助，关闭该参数可能更有利于性能。

设置不当的风险与影响：设置不当可能会影响查询性能。

cursor_tuple_fraction

参数说明：优化器估计游标获取行数在总行数中的占比。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0.1

说明

比默认值小的值与使用“fast start”为游标规划的值相偏离，从而使得前几行恢复的很快而抓取全部的行需要很长的时间。比默认值大的值加大了总的估计时间。在最大的值1.0处，像正常的查询一样规划游标，只考虑总的估计时间和传送第一行的时间。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值，如果实际应用中游标操作经常访问表中的大部分行，可以适当增加此参数的值。

设置不当的风险与影响：设置不当可能导致查询性能下降。

from_collapse_limit

参数说明：根据生成的FROM列表的项数来判断优化器是否将子查询合并到上层查询，如果FROM列表项个数小于等于该参数值，优化器会将子查询合并到上层查询。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

版权所有 © 华为云计算技术有限公司

默认值：8

说明

比默认值小的数值将降低规划时间，但是可能生成差的执行计划。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，如果查询中经常需要处理多个子查询，并且这些子查询可以有效地合并以提高查询性能，可以考虑增大此参数的值。反之，如果合并子查询导致性能下降或增加了优化器生成执行计划的时间，可以考虑减小该值。

设置不当的风险与影响：设置不当可能会导致查询性能下降。

join_collapse_limit

参数说明：根据得出的列表项数，来判断优化器是否把除FULL JOIN之外的JOIN构造重写到FROM列表中。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：8

说明

- 设置为1会避免任何JOIN重排。这样就使得查询中指定的连接顺序就是实际的连接顺序。查询优化器并不是总能选取最优的连接顺序，高级用户可以选择暂时把这个参数设置为1，然后指定它们需要的连接顺序。
- 比默认值小的数值减少规划时间但也降低了执行计划的质量。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。降低参数值减少了规划时间，但可能降低计划生成质量；提高参数值增加了规划时间，但有可能生成更优计划。

设置不当的风险与影响：规划时间与计划生成质量的权衡，设置不当可能导致其中一方效果难以接受。

plan_mode_seed

参数说明：该参数为调测参数，目前仅支持OPTIMIZE_PLAN和RANDOM_PLAN两种。

参数类型：整型

参数单位：无

取值范围：-1 ~ 2147483647

- 0：表示OPTIMIZE_PLAN模式，即通过动态规划算法进行代价估算生成最优plan。
- -1：表示RANDOM_PLAN模式，即随机生成plan，且用户不指定生成随机数的种子标识符seed值，由优化器随机生成[1, 2147483647]范围内的整数值，并根据随机数生成随机的执行计划。

版权所有 © 华为云计算技术有限公司

- [1, 2147483647]范围内的整数值：表示RANDOM_PLAN模式，即随机生成plan，且用户指定的生成随机数的种子标识符seed值，优化器根据seed值生成随机的执行计划。

默认值：0

说明

当该参数不为0时，查询指定的plan hint不会生效。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：当该参数设置为RANDOM_PLAN模式时，优化器会随机生成执行计划，该执行计划可能不是最优计划，从而对查询性能产生影响。所以，在升级、扩容、缩容等正常业务操作或运维过程中，建议将该参数设置为0。

设置不当的风险与影响：设置不当可能导致查询性能下降。

enable_random_datanode

参数说明：表示是否允许开启复制表DN随机查找功能，复制表在每个DN存放一份完整数据，随机选取可以缓解节点压力。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许开启复制表DN随机查找功能。
- off：表示不允许开启复制表DN随机查找功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致节点压力激增。

hashagg_table_size

参数说明：用于设置执行HASH JOIN操作时HASH表的大小。

参数类型：整型

参数单位：无

取值范围：0 ~ 1073741823，其中0表示数据库将根据需要自动调整HASH表的大小。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在实际应用中，如果遇到特定的查询场景，比如处理大量数据的聚合操作，可能需要手动调整此参数以优化性能。

设置不当的风险与影响：增加HASH表的大小可以减少HASH AGG操作中的磁盘I/O，因为更多的数据可以保留在内存中。然而，如果设置HASH表大小过大，可能会占用过多内存，导致内存不足。如果设置过小，可能无法有效地利用内存，导致更多的磁盘I/O操作，从而降低查询速度。

版权所有 © 华为云计算技术有限公司

enable_codegen

参数说明：标识是否允许开启代码生成优化，目前代码生成使用的是LLVM优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许开启代码生成优化。
- off：表示不允许开启代码生成优化。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启参数，可能会导致表达式查询场景资源使用上涨。

codegen_compile_thread_num

参数说明：该参数用于设置Codegen编译线程的线程数量。

参数类型：整型

参数单位：无

取值范围：1 ~ 8

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，当业务的并发量较大且业务的性能瓶颈在表达式执行流程时再做修改。

设置不当的风险与影响：线程数量设置过多，可能会导致系统性能下降。但当业务并发量较大时，可适当调大线程数以提升吞吐性能。

llvm_max_memory

参数说明：此参数设置Codegen编译产生的IR（包括缓存的和正在使用的）所占用的最大内存。Codegen使用的内存不采用预占的方式申请，属于max_dynamic_memory的一部分，且受llvm_max_memory参数的限制。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647，超过设定值则不再使用Codegen执行逻辑，降级为原有递归执行逻辑。当到达该上限触发执行降级后，调小llvm_max_memory的值无法立即释放多余IR所占用的内存，实际IR所占用的内存随着对应的SQL执行完毕后释放。

默认值：131072（即128MB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为100，不带单位，表示100kB；设置为16MB，带单位，表示16MB。取值如果要带单位，必须为kB、MB、GB。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值，当系统视图pv_total_memory_detail中llvm_used_memory项到达默认值上限且业务的性能瓶颈在表达式执行流程时，再做修改。

设置不当的风险与影响：

- 最大占用内存如果设置过小，则易使系统不再使用Codegen执行逻辑，影响功能使用。
- 最大占用内存如果设置过大，可能会造成LLVM的编译过多占用其他线程的资源，造成系统整体性能下降。

enable_codegen_print

参数说明：标识是否允许在log日志中打印所生成的LLVM IR函数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许在log日志中打印IR函数。
- off：表示不允许在log日志中打印IR函数。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，当定位问题时再开启。

设置不当的风险与影响：若开启则会产生大量的log日志，占用磁盘I/O，从而使数据库系统前台查询的读写性能下降。

codegen_cost_threshold

参数说明：由于LLVM编译生成最终的可执行机器码需要一定时间，因此只有当实际执行的代价大于编译生成机器码所需要的代价和优化后的执行代价之和时，利用代码生成才有收益。codegen_cost_threshold标识代价的阈值，当执行估算代价大于该代价时，使用LLVM优化。codegen使用执行算子的plan_rows作为代价与codegen_cost_threshold进行比较，plan_rows的值可以通过explain命令查看。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：100000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，当需要调整codegen机制的触发阈值时，再做修改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_bloom_filter

参数说明：标识是否允许使用BloomFilter优化。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许使用BloomFilter优化。
- off：表示不允许使用BloomFilter优化。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bloom_filter_build_max_rows

参数说明：启用BloomFilter优化时，HashJoin的Build侧可以创建BloomFilter的最大数据量。如果build侧的数据量高于该值，则不会建立BloomFilter。

参数类型：整型

参数单位：条

取值范围：1 ~ 2147483647

默认值：2000000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。该参数仅在参数**enable_bloom_filter**打开的情况下生效。

设置建议：在存储与计算资源充足的情况下，可以适当调大此值以获得更佳性能。

设置不当的风险与影响：当存储与计算资源不足，**query_dop**较高（>16）并且该值设定过大（>2000000）的情况下，有可能引起系统性能下降。

bloom_filter_apply_threshold

参数说明：启用BloomFilter优化时，HashJoin的Apply侧可以创建BloomFilter的最小数据量。如果apply侧的数据量低于该值，则不会建立BloomFilter。

参数类型：整型

参数单位：条

取值范围：1 ~ 2147483647

默认值：10000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。该参数仅在参数**enable_bloom_filter**打开的情况下生效。

设置建议：建议设置为1000~20000之间的整数。

设置不当的风险与影响：该值设置过大时（>20000），会丢失添加BloomFilter后可以优化性能的场景。该值设置过小时（<1000），会对数据规模较小的查询也进行BloomFilter的添加，反而引起性能劣化。

版权所有 © 华为云计算技术有限公司

说明

- 开启/设置enable_bloom_filter、bloom_filter_build_max_rows参数后，仍需要满足BloomFilter实现/原理限制，才会生成BloomFilter。
- 关于BloomFilter实现/原理限制：
  - build端需要在HashJoin内侧。
  - apply端必须是IMCVScan。
  - 不支持Left Join、Full Join和Left Anti Join种类的Join。
  - BloomFilter在apply/build时跨越了无法下推的算子，如Func Agg、Limit等。
  - BloomFilter调用Semi Join接口进行估算，只有外表选择率小于2/3才可以创建BloomFilter，外表选择率通过统计信息计算得到，详见《开发指南》中“特性使用指导 > CBO优化器 > 技术原理 > 统计信息”章节。

build端节点与apply端节点可以通过观察计划确认。对于计划中的HashJoin节点，下方的子节点为build侧，上方的子节点为apply侧。以下述计划为例，Imcv Scan on public.row_hashjoin_table_01 a为build侧，Imcv Scan on public.row_hashjoin_table_02 b为apply侧：

```
                                                                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
 Row Adapter  (cost=3033.82..3033.82 rows=59804 width=48)
   Output: a.c_int, a.c_bigint, a.c_smallint, b.c_int, b.c_bigint, b.c_smallint
   ->  Vector Sonic Hash Join  (cost=374.00..3033.82 rows=59804 distinct=[200, 600] width=48)
         Output: a.c_int, a.c_bigint, a.c_smallint, b.c_int, b.c_bigint, b.c_smallint
         Hash Cond: (b.c_int = a.c_int)
         Generate Bloom Filter On Expr: a.c_int
         Generate Bloom Filter On Index: 0
         ->  Imcv Scan on public.row_hashjoin_table_02 b  (cost=0.00..924.00 rows=75852 width=24)
               Output: b.c_int, b.c_bigint, b.c_smallint
               Filter: (b.c_int > 2400)
               Filter By Bloom Filter On Expr: b.c_int
               Filter By Bloom Filter On Index: 0
         ->  Imcv Scan on public.row_hashjoin_table_01 a  (cost=0.00..299.00 rows=6000 width=24)
               Output: a.c_int, a.c_bigint, a.c_smallint
               Filter: (a.c_int > 2400)
```

enable_extrapolation_stats

参数说明：对于日期类型，标识是否允许基于历史统计信息使用推理估算的逻辑。若使用该逻辑，对于未及时收集统计信息的表，可以提高估算的准确性，但也存在推理错误导致估算过大的可能。在日期类型数据定期插入的场景下，需要开启此开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许基于历史统计信息使用推理估算的逻辑。
- off：表示不允许基于历史统计信息使用推理估算的逻辑。

默认值：

- 新安装的数据库默认值为on。
- 若升级的基础版本不存在该GUC参数，默认值为on；若升级的基础版本存在该GUC参数，保留原设置。

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。在启用enable_extrapolation_stats之前，应该评估数据的变化频率和查询模式。如果数据变化很快，而且查询优化器经常需要处理不完整的统计信息，启用这个参数可能会有所帮助。在正式启用之前，应该在测试环境中进行充分的测试，以确定启用外推统计信息对性能的具体影响。启用外推统计信息后，应密切监控查询性能和数据库的统计信息准确性，确保没有引入性能问题或导致统计信息的准确性大幅下降。

设置不当的风险与影响：打开该参数后，可能会存在推理错误导致查询性能下降的问题。

query_dop

参数说明：用户自定义的查询并行度。

参数类型：整型

参数单位：无

取值范围：-64 ~ 64

- [1,64]：打开固定SMP功能，系统会使用固定并行度。
- 0：打开SMP自适应功能，系统会根据资源情况和计划特征动态选取最优并行度。
- [-64,-1]：打开SMP自适应功能，并限制自适应选取的最大并行度。

说明

- 在开启并行查询后，请保证系统CPU、内存、网络、I/O等资源充足，以达到最佳效果。
- 为了避免用户设置不合理的过大值造成性能劣化，系统会计算出该DN可用最大CPU核数，并以此来作为query_dop的上限。如果用户设置query_dop超过4并且同时超过该上限，那么系统会重置query_dop为该上限值。

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，在 CPU、内存、I/O 和网络带宽等资源充足的情况下，并行度越高，性能提升效果越好。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_analyze_check

参数说明：标识在生成计划的时候，对于在pg_class中reltuples和relpages均为0的表，是否允许检查这些表是否曾进行过统计信息的收集。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许检查。
- off：表示不允许检查。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。启用检查可能会带来计划生成的开销，但会确保是否进行过统计信息的收集；如果确定大多数在pg_class中reltuples和relpages均为0的表无需额外收集统计信息，可保持关闭。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_sonic_hashagg

参数说明：标识是否依据规则约束使用基于面向列的hash表设计的Hash Agg算子。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在满足约束条件时，使用基于面向列的hash表设计的Hash Agg算子。
- off：表示不使用面向列的hash表设计的Hash Agg算子。

说明

- 当开启enable_sonic_hashagg，且查询达到约束条件使用基于面向列的hash表设计的Hash Agg算子时，查询对应的Hash Agg算子内存使用通常可获得精简。但对于代码生成技术可获得显著性能提升的场景(**enable_codegen**打开后获得较大性能提升)，对应的算子查询性能可能会出现劣化。
- 当开启enable_sonic_hashagg，且查询达到约束条件使用基于面向列的hash表设计的Hash Agg算子时，在Explain Analyze/Performance的执行计划和执行信息中，算子显示为“Sonic Hash Aggregation”，而当未达到该约束条件时，算子名称将显示为“Hash Aggregation”，Explain详解请参见《SQL调优指南》中“SQL执行计划介绍 > 执行计划详解”章节。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致该场景下查询性能劣化。

enable_sonic_hashjoin

参数说明：标识是否依据规则约束使用基于面向列的hash表设计的Hash Join算子。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在满足约束条件时，使用基于面向列的hash表设计的Hash Join算子。
- off：表示不使用面向列的hash表设计的Hash Join算子。

版权所有 © 华为云计算技术有限公司

说明

- 当前开关仅适用于Inner Join的场景。
- 当开启enable_sonic_hashjoin时，查询对应的Hash Inner算子内存使用通常可获得精简。但对于代码生成技术可获得显著性能提升的场景，对应的算子查询性能可能会出现劣化。
- 当开启enable_sonic_hashjoin，且查询达到约束条件使用基于面向列的hash表设计的Hash Join算子时，在Explain Analyze/Performance的执行计划和执行信息中，算子显示为“Sonic Hash Join”，而当未达到该约束条件时，算子名称将显示为“Hash Join”，Explain详解请参见《SQL调优指南》中“SQL执行计划介绍 > 执行计划详解”章节。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致该场景下查询性能劣化。

vector_engine_features

参数说明：标识向量化引擎可选特性的开关。开启这些特性不一定对执行性能有提升效果，在特定的业务场景中，通过此GUC参数对向量化引擎相关特性进行设置，使得执行性能最优。

此参数可以设置向量化引擎特性开关的组合。比如有多个特性：rule1、rule2、rule3、rule4，可以设置：

```
set vector_engine_features='rule1';          --启用向量化引擎特性rule1。
set vector_engine_features='rule2,rule3';    --启用向量化引擎特性rule2和rule3。
set vector_engine_features=none;           --关闭所有可选向量化引擎特性。
```

参数类型：枚举类型

参数单位：无

取值范围：

- none：不使用任何可选向量化引擎特性。
- enable_ca_hashjoin：使用面向Cache-Aware HashTable设计的Ca Hash Join算子。
- enable_ca_hashagg：使用面向Cache-Aware HashTable设计的Ca Hash Agg算子。

默认值：enable_ca_hashjoin, enable_ca_hashagg

设置方式：该参数属于USERSET类型参数，请参见中表6-2对应设置方法进行设置。

设置建议：推荐使用默认值。如有其它必要，建议明确执行场景后，仅打开对应特性开关。

设置不当的风险与影响：关闭特性开关可能会导致某些场景下执行性能劣化，建议充分测试后使用对应特性开关。

版权所有 © 华为云计算技术有限公司

说明

- GUC参数中设置enable_ca_hashjoin时，该开关和基础开关enable_hashjoin存在依赖关系，enable_hashjoin关闭时，计划中不存在HashJoin，enable_ca_hashjoin无效。
  - 当前开关仅适用于以下场景：
    - 连接键
      - 同类型连接键。如：int = varchar， 不支持。
      - 非表达式连接键。如：a - 1 = b，不支持。
      - 连接键支持的数据类型请参见**表6-22**。
    - 连接类型
      - Inner Join：支持。
      - Right Join：部分支持。带有连接条件的情况下不支持，其他情况下支持。
      - Semi Join：部分支持。非投影情况下支持，仅输出外表数据情况下支持，投影情况下、有输出内表数据情况下不支持。
      - Anti Join：部分支持。非投影情况下支持，仅输出外表数据情况下支持，投影情况下、有输出内表数据情况下不支持。
      - Right Semi Join：部分支持。仅输出内表数据情况下支持，有输出外表数据情况下不支持。
      - Right Anti Join：部分支持。仅输出内表数据情况下支持，有输出外表数据情况下不支持。
      - Full Join：不支持。
      - Left Join：不支持。
    - 其它
      - 内表数据输出小于等于64列支持，大于64列不支持。
  - 当开启enable_ca_hashjoin，且查询达到约束条件，使用面向Cache-Aware HashTable设计的Ca Hash Join算子时，在Explain Analyze/Performance的执行计划和执行信息中，算子显示为“Vector Ca Hash Join”，而当未达到该约束条件时，算子名称将显示为“Vector Sonic Hash Join”或者“Vector Hash Join”，Explain详解请参见《SQL调优指南》中“SQL执行计划介绍 > 执行计划详解”章节。
- GUC参数中设置enable_ca_hashagg时，该开关和基础开关enable_hashagg存在依赖关系，enable_hashagg关闭时，计划中不存在hashagg，enable_ca_hashagg无效。
  - 当前开关仅适用于以下场景：
    - 原向量化Agg支持的非numeric类型。
    - Agg函数个数不超过21个。
    - 下层算子返回的列仅含作为HashKey的列、用于聚集的列和与HashAgg无关的列（例外情况如SELECT col1::numeric FROM test_table GROUP BY col1; 这种情况下，下层算子可能会返回col1和col1::numeric，而只有col1会被作为HashKey）。
    - Agg节点的unique_check标记为false（当前只有子查询的部分情况下可能为true）。
    - 连接键支持的数据类型请参见**表6-22**。
  - 当开启enable_ca_hashagg，且查询达到约束条件，使用面向Cache-Aware HashTable设计的Ca Hash Agg算子时，在Explain Analyze/Performance的执行计划和执行信息中，算子显示为“Vector Ca Hash Aggregate”，而当未达到约束条件时，算子名称将显示为“Vector Sonic Hash Aggregate”或者“Vector Hash Aggregate”，Explain详解请参见《SQL调优指南》中“SQL执行计划介绍 > 执行计划详解”章节。

版权所有 © 华为云计算技术有限公司

表 6-22 连接键支持的数据类型

类别	数据类型
数值类型	TINYINT [UNSIGNED]
	SMALLINT [UNSIGNED]
	MEDIUMINT [UNSIGNED]
	BIGINT [UNSIGNED]
	FLOAT4
	FLOAT8
日期/时间类型	DATE
	DATEA
	SMALLDATETIME
	ABSTIME
	INTERVAL
	RELTIME
	TIME [WITHOUT TIME ZONE]
	TIMESTAMP [WITHOUT TIME ZONE]
字符类型	BPCHAR
	CHAR
	VARCHAR
	TEXT
其他数据类型	BOOL
	OID

enable_sonic_optspill

参数说明：标识是否对面向列的hash表设计的Hash Join算子进行下盘文件数优化。该参数打开时，在Hash Join算子下盘文件较多的时候，下盘文件数不会显著增加。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示优化面向列的hash表设计的Hash Join算子的下盘文件数。
- off：表示不优化面向列的hash表设计的Hash Join算子的下盘文件数。

默认值：on

 版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致该场景下查询性能劣化。

plan_cache_mode

参数说明：标识在prepare语句中，选择生成执行计划的策略。

参数类型：枚举类型

参数单位：无

取值范围：auto、force_generic_plan、force_custom_plan

- auto：表示由优化器自动选择custom plan或者generic plan。
- force_generic_plan：表示强制走generic plan（软解析）。generic plan是指对于prepare语句生成计划，该计划策略会在执行execute语句的时候把参数bind到plan中，然后执行计划。这种方案的优点是每次执行可以省去重复的优化器开销；缺点是当bind参数字段上数据存在倾斜时该计划可能不是最优的，部分bind参数场景下执行性能较差。bind会绑定第一次传入的参数类型，如果同一占位符两次传入的参数类型不一致会导致报错。
- force_custom_plan：表示强制走custom plan（硬解析）。custom plan是指对于prepare语句，在执行execute的时候，把execute语句中的参数嵌套到语句之后生成的计划。custom plan会根据execute语句中具体的参数生成计划，这种方案的优点是每次都按照具体的参数生成优选计划，执行性能比较好；缺点是每次执行前都需要重新生成计划，存在大量的重复的优化器开销。

说明

此参数只对prepare语句生效，一般用在prepare语句中参数化字段存在比较严重的数据倾斜的场景下。

默认值：auto

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据实际业务场景进行设置。

设置不当的风险与影响：设置不当可能会导致计划生成开销增加或计划生成质量下降。

enable_router

参数说明：是否打开手动设置下推节点功能，能将支持的查询语句下推到特定DN上执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需手动设置下推节点以针对特定分布列时启用。

设置不当的风险与影响：此功能打开后，若设置错分布列，可能导致数据存入错误的DN，需谨慎使用。

router

参数说明：用于控制router功能的详细属性，仅在打开**enable_router**和**enable_light_proxy**后生效。该参数会根据表的hash分布列，计算给定的分布列所在DN，设置router后将支持的sql下推到该DN上执行。

参数类型：字符串

参数单位：无

取值范围：

该参数分为两部分，'schema_name.table_name,"distribute_keys"'（注意双引号），其具体含义如下：

- schema_name.table_name：表示schema名和表名，如不显示设置schema_name，则默认为current_schema。
- distribute_keys：分布表的所有分布列值，用逗号间隔开，且分布列值的顺序必须和表中分布列顺序一致。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需手动设置下推节点以针对特定分布列时启用。

设置不当的风险与影响：如果设置错router，可能导致数据存入错误的DN，导致不可预测的问题，需谨慎使用。

enable_force_vector_engine

参数说明：对于支持向量化的执行器算子，如果其子节点是非向量化的算子，通过设置此参数为on，强制生成向量化的执行计划。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示可以向量化的算子强制生成向量化。
- off：表示由向量化算子优化器决定是否向量化。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，在向量化业务场景下推荐打开。

设置不当的风险与影响：启用该参数可能导致查询性能劣化。

版权所有 © 华为云计算技术有限公司

enable_auto_explain

参数说明：控制是否开启自动打印执行计划。该参数可用来定位慢存储过程或慢查询，只对当前连接的CN有效，不影响其他CN的执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，若需要查看执行计划，则开启，但会降低当前系统性能。

设置不当的风险与影响：开启后可能会降低当前系统性能。

auto_explain_level

参数说明：控制自动打印执行计划的日志等级。

参数类型：枚举类型

参数单位：无

取值范围：

- log：表示log级别及以上的日志会被打印。
- notice：表示notice级别及以上的日志会被打印。

默认值：log

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

auto_explain_log_min_duration

参数说明：控制自动打印执行计划的耗时阈值，整体耗时大于auto_explain_log_min_duration的执行计划才会被打印。比如该参数设置为0时，所有执行过的执行计划都会输出；设置为3000时，单次语句执行耗时超过3000毫秒的所有执行的执行计划会输出。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：0

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。例如，设置为100，不带单位，表示100ms；设置为2min，带单位，表示2min。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：推荐使用默认值。可根据业务需要调整数值，以输出慢查询语句。

设置不当的风险与影响：设置过小可能会导致输出内容过多。

auto_explain_analyze_options

参数说明：控制自动打印执行计划是否显示运行时计划信息。该参数可以在DN日志中查看FQS和PGXC计划下推语句的实际执行计划，只对当前连接的CN有效，不影响其他CN的执行。

说明

- 该参数功能需要enable_auto_explain打开才能使用，继承enable_auto_explain约束，对于EXPLAIN语句不打印，只支持正常执行的语句计划打印。
- DN上计划日志仅在enable_auto_explain = true且auto_explain_level = log场景下生效。
- 设置除None外的任何值，都会包括ANALYZE功能，显示基本的运行时计划信息。
- 不支持Stream计划、SMP计划、SqlBypass计划。如果计划没有显示运行时信息，检查是否生成了Stream计划、SMP计划、SqlBypass计划。
- 每个节点只能显示当前节点的计划信息，例如CN不会在日志或客户端中显示DN的计划信息。
- 当前版本不支持auto explain显示cpu运行时计划信息，参数取值范围中Cpu作为预留选项使用。

参数类型：枚举类型

参数单位：无

取值范围：

- None：设置自动打印执行计划不显示运行时计划信息。
- Analyze：收集基本的运行时计划信息，包括rows、time。
- Buffers：收集运行时rows、time、buffer相关信息。
- Cpu：收集运行时rows、time、cpu相关信息。

默认值：None

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值，若需要查看执行时计划信息，则开启，但会降低当前系统性能。

设置不当的风险与影响：开启后可能会降低当前系统性能。

max_datanode_for_plan

参数说明：设置生成FQS计划时DN上执行计划显示的个数。DN上计划显示的个数由集群中的DN数和该参数值的更小者决定。

参数类型：整型

参数单位：无

取值范围：0 ~ 8192

版权所有 © 华为云计算技术有限公司

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在需要更详细的多节点执行计划时可修改此值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_inner_unique_opt

参数说明：控制是否对嵌套循环连接、哈希连接、排序归并连接进行Inner Unique优化，即在连接条件中内表对应的属性满足唯一性约束的情况下，减少匹配次数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致对应场景查询性能下降。

enable_indexscan_optimization

参数说明：控制是否对astore存储引擎下的btree索引扫描（IndexScan和IndexOnlyScan）进行优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用。
- off：表示不使用。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致对应场景查询性能下降。

enable_uniq_idx_a_compat

参数说明：控制复合唯一索引针对Null值是否兼容ORA数据库。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示兼容。
- off：表示不兼容。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数仅在ORA兼容性的数据库中适用，建议使用默认值。

设置不当的风险与影响：设置不当可能会影响兼容性或导致查询规划开销增加。

create_gsi_opt

参数说明：控制全局二级索引GSI创建模式。注意，除none模式外，其他模式需要打开**enable_stream_operator**参数才会生效。

参数类型：枚举类型

参数单位：无

取值范围：

- none：使用PGXC的方式创建。
- build：使用STREAMING+GSI BUILD算子的方式创建。
- parallel：使用STREAMING+GSI BUILD算子+并行排序的方式创建（需要设置基表的parallel_workers参数）。

默认值：build

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能导致全局二级索引创建性能下降。

session_sequence_cache

参数说明：在当前会话下，一次性交互申请的sequence数值，会话结束后会自动丢弃未用完的值。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：10

说明

默认值10，在高并发场景下，兼顾单条插入和批量插入，性能都会有比较好的表现。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：用户在使用sequence大批量导入数据时，可以通过调大该参数，提高插入速度，增加高并发性能；用户在高并发单条插入数据时，将该参数设置为1，减少sequence的跳变。如对连续性有强要求，需要在创建sequence的时候指定需要的cache，该参数如果大于cache指定的值会自动失效。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置过大时，可能导致单条插入性能下降；设置过小时，可能导致批量插入性能下降。

enable_invisible_indexes

参数说明：设置优化器是否可以使用不可见索引。

说明

索引在被设置为不可见状态后，可能会对查询语句的性能产生影响。此时若不想变更索引可见性状态，且打算使用不可见索引，可设置enable_invisible_indexes参数为on。

参数类型：布尔型

参数单位：无

取值范围：

- on：优化器可以使用不可见索引。
- off：优化器不可以使用不可见索引。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：使用不可见索引但未开启此参数时，可能导致忽略不可见索引，因此可能有更优计划未被考虑。

enable_dynamic_samplesize

参数说明：是否动态调整采样行数。对于超过一百万行的大表，收集统计信息时动态调整采样行数，可以提高统计信息准确性。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示该功能打开。
- off：表示该功能关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后可能会降低统计信息准确性。

说明

动态调整采样行数的功能仅支持绝对值采样。

default_statistic_granularity

参数说明：默认情况下，当未指定PARTITION_MODE时，指定默认收集哪些分区表分区级统计信息。对非分区表不生效。

版权所有 © 华为云计算技术有限公司

参数类型：枚举类型

参数单位：无

取值范围：

- all：收集整表、一级分区的统计信息。
- global：收集整表的统计信息。
- partition：收集一级分区的统计信息。
- global_and_partition：收集整表、一级分区的统计信息。
- all_complete：收集整表、一级分区的统计信息，相对于取值为all的场景，采样率更高，可能带来更长的统计信息耗时。

默认值：global

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。如果需要收集分区级统计信息，可根据实际需要设置，但可能会对ANALYZE的性能产生一定影响。

设置不当的风险与影响：需要在维护开销与统计信息准确性间权衡，设置不当可能导致一方代价过大。

说明

升级场景下，升级后该GUC参数值和升级前保持一致。

enable_fast_numeric_agg

参数说明：该参数控制是否开启针对numeric数据类型的agg优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启numeric数据类型的agg优化。
- off：表示不开启numeric数据类型的agg优化。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后可能会导致对应场景查询性能下降。

enable_poisson_outer_optimization

参数说明：该参数用于控制计划生成阶段计算选择率时，是否用泊松修正outer join中外表上的filter条件的选择率。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示泊松不修正outer join中外表的filter条件的选择率，即计算选择率时不将外表的filter条件按照可下推计算。
- off：outer join外表的filter条件下推计算选择率。

默认值：升级上来的集群默认为off，新安装的集群默认为on。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：可能无法选中较好的执行计划，出现慢sql。

说明

当GUC参数cost_model_version>=5或为0且enable_poisson_outer_optimization为on时，相关功能才会生效。

enable_predreorder_by_rank

参数说明：设置进行计划生成时，是否要将单一基表的多个过滤条件基于其选择率、计算代价进行重排序，以减小执行开销。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进行计划生成时，将多个过滤条件基于其选择率、计算代价进行重排序。
- off：表示进行计划生成时，不将多个过滤条件基于其选择率、计算代价进行重排序。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭后将导致部分计划执行无法得到优化。

说明

当cost_model_version>=6或为0且enable_predreorder_by_rank为on时，相关功能才生效。

enable_shared_set_update

参数说明：对于UPDATE语句中使用子查询更新多个列的场景，该参数可以控制优化器是否对每个待更新列分别生成一个子计划，当打开该参数时，将只生成一个子计划在多个列之间共享。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示仅根据子查询生成一个子计划。

版权所有 © 华为云计算技术有限公司

- off：表示为每个待更新列，分别生成一个子计划。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会降低使用子查询更新多个列的查询执行效率，且增加查询优化和执行开销。

说明

- 当cost_model_version>=6或为0且enable_shared_set_update为on时，相关功能才生效。
- 当enable_shared_set_update为on且生成shared set update计划时，如果语句中存在IMCV表，则不会生成IMCV SCAN计划。

cursor_sharing

参数说明：设置是否启用自动参数化或相同语句计划复用功能。

参数类型：枚举类型

参数单位：无

取值范围：

- force：启用特性自动参数化和特性相同语句计划复用功能。如果满足自动参数化条件，则优先通过自动参数化缓存；满足不了则通过相同语句计划复用缓存。例如在MYSQL或M兼容模式下，不支持参数化，会自动按照cursor_sharing = exact的行为实现。
- exact：仅启用特性相同语句计划复用功能。
- off：自动参数化和特性相同语句计划复用两个功能均关闭。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数设置为force时，语句中的部分常量将被替换为参数。可能会导致SQL计划的改变，可能引起计划性能下降，并可能会影响语法树，造成unique_sql_id的变化，导致sqlpatch失效。

设置为force或exact，可能会导致占用内存空间增大，且如果计划模板无法复用的，无法通过计划复用带来性能提升，在设置参数时请确认业务模型是否适合。在计划内存上限管控的情况下，如果存入大量不同的sql模板，会引起计划缓存内存占用过多，触发计划内存上限淘汰，导致性能下降，极端场景下可能会引起掉零。

请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 启用自动参数化后，其执行机制与手动绑定参数保持一致。
- 启用自动参数化后，由于查询语句被自动改写为参数化形式，执行过程中输出的WARNING或ERROR信息（如提示文本、行号定位等）可能与未启用自动参数化时的输出存在差异，属于预期行为。

版权所有 © 华为云计算技术有限公司

opt_use_static_stats

参数说明：当开启此功能时，可以直接从PG_CLASS系统表中读取行数和页面数等统计信息，而不是对基表进行页面采样来获取，常用于手动验证优化器在给定统计信息的前提下，计划生成的效果。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示从PG_CLASS中直接读取统计信息。
- off：表示按照默认方式，对基表进行采样来获取行数和页面数等统计信息。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开该开关，会导致优化器不感知实际表大小，统计信息仅参考PG_CLASS系统表。

说明

当cost_model_version>=6或为0且opt_use_static_stats为on时，相关功能才生效。

enable_sel_cache

参数说明：当开启此功能时，在路径生成阶段，会针对经过优化器推导所得的、左右操作数均为普通列的、等值连接谓词进行选择率缓存，在路径搜索阶段进行谓词选择率的重复计算时，会使用该缓存加速计算。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用此功能。
- off：表示不使用此功能。

默认值：

- on：新安装的数据库默认值。
- off：从506.1.0之前版本升级到当前版本，此参数默认值为off；升级前若存在该参数，无论如何设置，升级后默认值均为off。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该开关，可能会导致优化器路径生成阶段性能下降。

版权所有 © 华为云计算技术有限公司

说明

- 不支持对多列统计信息的结果进行缓存。
- 当cost_model_version>=7或为0且enable_sel_cache为on时，相关功能才生效。
- 当且仅当遇到普通列构成的等值连接谓词时，才会构建缓存加速选择率计算。
- 如果左右操作数包含类型转换，且被转换的类型之间是二进制强制转化的，也可以使用此功能加速计算，具体可参考**PG_CAST**中对于castmethod的描述。
- 当同时关闭**enable_hashjoin**和**enable_mergejoin**时，此功能不生效。

stats_history_record_limit

参数说明：每个对象（包括表、列、分区、索引）的历史统计信息保留数目上限。每个对象在收集统计信息时会同时存入历史统计信息表中，当历史统计信息表中该对象的统计信息数量达到该阈值，再次收集新的统计信息时，时间较早的统计信息将被清理。

参数类型：整型

参数单位：条

取值范围：0 ~ 100

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，如果需要记录更多历史版本的统计信息可以适当调大该参数，但可能会对analyze的性能产生一定影响。

设置不当的风险与影响：设置过大可能会对analyze的性能产生一定影响。

stats_history_retention_time

参数说明：每个对象（包括表、列、分区、索引）的历史统计信息保留时间。每个对象在收集统计信息时会同时存入历史统计信息表中，当历史统计信息表中该对象的统计信息的保留时间超过该阈值后，再次收集新的统计信息时，超过保留时间的统计信息将被清理。

参数类型：浮点型

参数单位：d（天）

取值范围：-1、0 ~ 365000，-1表示历史统计信息不因时间而清除。

默认值：31

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，如果需要记录更早时间的历史版本的统计信息可以适当调大该参数，但可能会对analyze的性能产生一定影响。

设置不当的风险与影响：设置过大可能会对analyze的性能产生一定影响。

opt_behavior_knob

参数说明：用于控制优化器在特定场景下的行为变更，以进行手动查询调优。

参数类型：字符串

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：所有特定场景行为变更开启，可由SHOW opt_behavior_knob查看：

```
ALL,on(ROWID_OPT,DIST_QUERY_ADAPTIVE_FALLBACK,XC_REMOTE_JOIN_PARAM,MERGE_INTO_SHIP,SUBLINK_QUERY_SHIP,SUBPLAN_SORTED_PATH_FIX,PARAM_PATH_SKIP_CACHING_VARRATIO,PREFER_UNIQUE_INDEX,PREFER_LONGER_INDEX,PPI_COST_MODEL1,PREDICATE_INFER_ENHANCE,HIERARCHY_QRY_CHECK_SUBLINK_PULLUP,ACTUAL_RANGE_BY_INDEX,FIX_PART_INDEX_SCAN_COST,OPT_KNOB_20260703_1,OPT_KNOB_20260807_1,OPT_KNOB_20260728_1,OPT_KNOB_20260812_1),off()
```

- PARAM_PATH_SKIP_CACHING_VARRATIO：控制在生成参数化路径时，进行基数估计阶段，跳过相关谓词选择率的缓存，以保证上层连接基数估计的准确性。
- SUBPLAN_SORTED_PATH_FIX：控制子链接sorted path相关路径选择，避免代价低的路径直接淘汰，保证路径选择的准确性。
- DIST_QUERY_ADAPTIVE_FALLBACK：控制在CN节点资源紧张状态下（内存/CPU使用率超过80%），如果通过分布式自适应计划生成的计划是下发语句的分布式计划，且该计划为非参数化路径计划，则系统将其回退至Stream计划。
- XC_REMOTE_JOIN_PARAM：控制内表为RemoteJoin算子时，支持外表传递参数给内表执行。
- MERGE_INTO_SHIP：控制MERGE INTO下推功能，MERGE INTO USING SELECT FROM dual表场景下，可下推至DN执行CN轻量化/FQS计划。
- SUBLINK_QUERY_SHIP：控制子链接FQS下推功能，在满足可下推条件时，允许把子链接下推至DN执行。
- ROWID_OPT：控制ROWID系统列谓词条件改写、ORDER BY语句改写、max/min优化、分布剪枝和分区剪枝是否生效。详情可见《开发指南》中“特性使用指导 > ROWID表 > 使用指导 > DML”章节内容。
- PREFER_UNIQUE_INDEX：控制唯一索引扫描路径优先规则是否生效。该参数与sql_beta_feature中no_unique_index_first存在兼容性，两者为或的关系，唯一索引扫描路径优先规则在sql_beta_feature中未设置no_unique_index_first或PREFER_UNIQUE_INDEX开启时生效。仅支持等值场景，规则优先级低于PREFER_LONGER_INDEX。
- PREFER_LONGER_INDEX：控制最长最左索引扫描路径优先规则是否生效。仅支持等值场景，规则优先级高于PREFER_UNIQUE_INDEX。
- PPI_COST_MODEL1：控制LOCAL索引参数化路径场景下代价的修正，避免参数化路径场景选择LOCAL索引且查询条件不包含分区键时，代价估计过小导致计划不优。从不支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。
- PREDICATE_INFER_ENHANCE：控制谓词推导消减，简化谓词逻辑并提升查询性能。详情见参数cost_model_version的**谓词推导消减描述**。
- HIERARCHY_QRY_CHECK_SUBLINK_PULLUP：控制层次查询同层子链接提升的行为，当开启该参数时，将进行更严格的子链接提升检查。从不支持该参数的旧版本升级至当前版本场景下，该参数值保持默认关闭，且升级观察期该功能不生效。
- ACTUAL_RANGE_BY_INDEX：控制优化器在做范围条件选择性估算时，如果发现条件落在统计直方图的最左/最右端，是否直接通过索引查询真实的最大/最小值，避免统计过时导致估算偏差。
- FIX_PART_INDEX_SCAN_COST：控制分区表LOCAL索引扫描的代价修正，避免估算偏差导致计划不优。从不支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。
- OPT_KNOB_20260703_1：控制优化器在使用多列统计信息计算NDV时，是否将NULL值比例计入。打开该开关，不将其计入，可以获得更准确的基数估计。从不

 版权所有 © 华为云计算技术有限公司

支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。

- OPT_KNOB_20260807_1：控制优化器估算分区表索引页面数的修正，避免估算偏差导致计划不优。从不支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。
- OPT_KNOB_20260728_1：控制优化器agg代价模型，打开时使用超几何分布概率算法，关闭时使用等比例缩放算法，影响hashagg中agg行数估算，避免估算偏差导致计划不优。从不支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。
- OPT_KNOB_20260812_1：控制优化器生成nestloop路径时，外表带sort算子场景的代价修正，避免估算偏差导致计划不优。从不支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。

默认值：

- 新安装的数据库默认值为：

  'ALL,on(ROWID_OPT,DIST_QUERY_ADAPTIVE_FALLBACK,XC_REMOTE_JOIN_PARAM,MERGE_INTO_SHIP,SUBLINK_QUERY_SHIP,SUBPLAN_SORTED_PATH_FIX,PARAM_PATH_SKIP_CACHING_VARRATIO,PREFER_UNIQUE_INDEX,PREFER_LONGER_INDEX,PPI_COST_MODEL1,PREDICATE_INFER_ENHANCE,HIERARCHY_QRY_CHECK_SUBLINK_PULLUP,ACTUAL_RANGE_BY_INDEX,FIX_PART_INDEX_SCAN_COST,OPT_KNOB_20260703_1,OPT_KNOB_20260807_1,OPT_KNOB_20260728_1,OPT_KNOB_20260812_1),off()'。
- 参数PPI_COST_MODEL1、FIX_PART_INDEX_SCAN_COST、HIERARCHY_QRY_CHECK_SUBLINK_PULLUP、OPT_KNOB_20260703_1、OPT_KNOB_20260807_1、OPT_KNOB_20260728_1、OPT_KNOB_20260812_1行为说明：从不支持上述参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。
- 在PDB场景内，若未设置该参数，则继承来自全局的设置。

设置方式：该参数属于USERSET类型参数，不建议通过gs_guc工具来设置。设置方法如下：

首先，通过SHOW opt_behavior_knob来查看哪些行为变更是支持可控制的。例如，查询输出结果为：

```
gaussdb=# SHOW opt_behavior_knob;
              opt_behavior_knob
-----------------------------------------------


ALL,on(ROWID_OPT,DIST_QUERY_ADAPTIVE_FALLBACK,XC_REMOTE_JOIN_PARAM,MERGE_INTO_SHIP,SUBLINK_QUERY_SHIP,SUBPLAN_SORTED_PATH_FIX,PARAM_PATH_SKIP_CACHING_VARRATIO,PREFER_UNIQUE_INDEX,PREFER_LONGER_INDEX,PPI_COST_MODEL1,PREDICATE_INFER_ENHANCE,HIERARCHY_QRY_CHECK_SUBLINK_PULLUP,ACTUAL_RANGE_BY_INDEX,FIX_PART_INDEX_SCAN_COST,OPT_KNOB_20260703_1,OPT_KNOB_20260807_1,OPT_KNOB_20260728_1,OPT_KNOB_20260812_1),off()
(1 row)
```

支持可控制的行为变更使用大写来标识，特殊标识ALL用于对所有行为变更进行设置。可以使用on/off来控制行为变更是否生效。设置“参数化路径跳过缓存选择率”行为变更关闭，可使用如下命令：

```
gaussdb=# SET opt_behavior_knob = 'off(PARAM_PATH_SKIP_CACHING_VARRATIO)';
gaussdb=# SHOW opt_behavior_knob;
              opt_behavior_knob
-----------------------------------------------
```

 版权所有 © 华为云计算技术有限公司

```
ALL,on(ROWID_OPT,DIST_QUERY_ADAPTIVE_FALLBACK,XC_REMOTE_JOIN_PARAM,MERGE_INTO_SHIP,SUBLINK_QUERY_SHIP,SUBPLAN_SORTED_PATH_FIX,PREFER_UNIQUE_INDEX,PREFER_LONGER_INDEX,PPI_COST_MODEL1,PREDICATE_INFER_ENHANCE,HIERARCHY_QRY_CHECK_SUBLINK_PULLUP,ACTUAL_RANGE_BY_INDEX,FIX_PART_INDEX_SCAN_COST,OPT_KNOB_20260703_1,OPT_KNOB_20260807_1,OPT_KNOB_20260728_1,OPT_KNOB_20260812_1),off(PARAM_PATH_SKIP_CACHING_VARRATIO)
(1 row)
```

可以看到“参数化路径跳过缓存选择率”行为变更被关闭，其他行为变更保持原状态。

ALL标识相当于一个快捷操作，即对所有行为变更进行开启或关闭。

```
gaussdb=# SET opt_behavior_knob = 'off(ALL)';
SET
gaussdb=# SHOW opt_behavior_knob;
              opt_behavior_knob
-------------------------------------------------

ALL,on(),off(ROWID_OPT,DIST_QUERY_ADAPTIVE_FALLBACK,XC_REMOTE_JOIN_PARAM,MERGE_INTO_SHIP,SUBLINK_QUERY_SHIP,SUBPLAN_SORTED_PATH_FIX,PARAM_PATH_SKIP_CACHING_VARRATIO,PREFER_UNIQUE_INDEX,PREFER_LONGER_INDEX,PPI_COST_MODEL1,PREDICATE_INFER_ENHANCE,HIERARCHY_QRY_CHECK_SUBLINK_PULLUP,ACTUAL_RANGE_BY_INDEX,FIX_PART_INDEX_SCAN_COST,OPT_KNOB_20260703_1,OPT_KNOB_20260807_1,OPT_KNOB_20260728_1,OPT_KNOB_20260812_1)
(1 row)

gaussdb=# SET opt_behavior_knob = 'on(ALL)';
SET
gaussdb=# SHOW opt_behavior_knob;
              opt_behavior_knob
-------------------------------------------------

ALL,on(ROWID_OPT,DIST_QUERY_ADAPTIVE_FALLBACK,XC_REMOTE_JOIN_PARAM,MERGE_INTO_SHIP,SUBLINK_QUERY_SHIP,SUBPLAN_SORTED_PATH_FIX,PARAM_PATH_SKIP_CACHING_VARRATIO,PREFER_UNIQUE_INDEX,PREFER_LONGER_INDEX,PPI_COST_MODEL1,PREDICATE_INFER_ENHANCE,HIERARCHY_QRY_CHECK_SUBLINK_PULLUP,ACTUAL_RANGE_BY_INDEX,FIX_PART_INDEX_SCAN_COST,OPT_KNOB_20260703_1,OPT_KNOB_20260807_1,OPT_KNOB_20260728_1,OPT_KNOB_20260812_1),off()
(1 row)
```

设置建议：推荐使用默认值，根据需要关闭指定行为变更。

设置不当的风险与影响：关闭某个或者所有行为变更时会导致优化器输出非预期计划，可能影响系统性能。

说明

当cost_model_version>=5或为0且opt_behavior_knob指定行为变更为on时，相关功能才生效。

exec_behavior_knob

参数说明：用于控制执行器在特定场景下的行为变更。

参数类型：字符串

参数单位：无

取值范围：可由SHOW exec_behavior_knob查看参数值：

```
ALL,on(SHOW_BATCH_ERROR_DETAIL,INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION),off(AUTO_INCR_LP_INSERT)
```

- SHOW_BATCH_ERROR_DETAIL：batch批量执行场景下，如果在循环执行过程中抛出异常，则打印当前批次数目和报错批次中的绑参值。
- INDEXBY_FUNC_OPTIMIZE：在存储过程中调用带索引的集合类型变量的first/last/prior/next函数时，使用红黑树进行性能优化。开启后，为了维护红黑树，占用内存会增加，部分场景性能会劣化。

版权所有 © 华为云计算技术有限公司

- SYS_CB_PATH_NULL_CTRL：控制层次查询系统函数SYS_CONNECT_BY_PATH执行时，如果读取到NULL值，是否进行特殊处理。开启后，对于NULL值会输出空；否则，输出“null”字符串。从不支持该参数的旧版本升级上来，观察期内开关默认为开启状态且参数不生效；观察期结束后，开关自动切换为关闭状态。
- AUTO_INCR_LP_INSERT：用于控制分区键为自增列的简单插入场景下，是否启用light proxy下推优化。关闭后，该场景下通过explain展示执行计划时，由light proxy变为pgxc，导致部分场景性能劣化。
- PGXC_MERGE_UPDATE：用于控制在PGXC计划下，单条SQL语句，执行MERGE INTO语法时，是否允许对同一行数据进行多次更新。如果开启该参数，且开启enable_cid_sync参数，则报错。
- SORT_KEYS_CACHE：用于控制是否启用Sort算子多列信息缓存机制以提升排序性能操作。开启时，可动态缓存排序键相关信息，从而提升行执行器排序性能。

  说明

  Sort算子多列信息缓存使用限制如下：

  - 仅支持全量排序

    本特性针对行执行器高频交叉比对的全量排序场景进行优化。对于带有LIMIT子句的Top-N查询，由于采用堆排序（Heap Sort），其整体比对次数较少，额外缓存多列数据的性价比低，因此此类场景下该特性不生效。若带LIMIT子句的查询采用快速排序，则可启用本优化特性。例如，执行计划输出结果为：

    ```
    gaussdb=# EXPLAIN ANALYZE SELECT * FROM test_sort ORDER BY 1 LIMIT 1;
                                                        QUERY PLAN
    ------------------------------------------------------------------------------------------------------------------------
    --------
     [Parameterized]
     Limit  (cost=46.03..46.03 rows=1 width=4) (actual time=0.098..0.098 rows=1 loops=1)
       ->  Sort  (cost=46.03..52.04 rows=1 width=4) (actual time=0.095..0.095 rows=1 loops=1)
             Sort Key: a
             Sort Method: top-N heapsort  Memory: 25kB
             ->  Seq Scan on test_sort  (cost=0.00..34.02 rows=2402 width=4) (actual
    time=0.045..0.050 rows=10 loops=1)
     Total runtime: 0.397 ms
    (7 rows)
    gaussdb=# EXPLAIN ANALYZE SELECT * FROM test_sort ORDER BY 1 LIMIT 6;
                                                        QUERY PLAN
    ------------------------------------------------------------------------------------------------------------------------
    --------
     [Parameterized]
     Limit  (cost=77.08..77.09 rows=6 width=4) (actual time=0.132..0.137 rows=6 loops=1)
       ->  Sort  (cost=77.08..83.08 rows=6 width=4) (actual time=0.127..0.129 rows=6 loops=1)
             Sort Key: a
             Sort Method: quicksort  Memory: 25kB
             ->  Seq Scan on test_sort  (cost=0.00..34.02 rows=2402 width=4) (actual
    time=0.060..0.070 rows=10 loops=1)
     Total runtime: 0.551 ms
    (7 rows)
    ```

  - 唯一键场景

    在一个单数据节点（DN）上，如果排序键仅有一列，或者第一排序键的数据具备唯一性（只要实际数据唯一，不需要唯一性约束），说明不会发生二次比对。此时该特性不生效。

  - 内存溢出保护

    当原始排序数据所需内存已超出系统配置的工作内存（**work_mem**），需溢出磁盘处理时；或计入多列缓存带来的额外内存开销后，整体占用超出工作内存（**work_mem**）时，多列缓存特性将自动失效。

  - 排序键数量限制

    为避免缓存结构过度膨胀，当查询语句中的排序键总数超过32个时，系统将不启用该特性。

版权所有 © 华为云计算技术有限公司

- PLSQL_RETURN_EXCEPTION：用于控制ORA兼容模式下，存储过程RETURN语句类型转换报错能否被EXCEPTION语句捕获。详细可见**错误捕获语句**。

  说明

  从不支持该参数选项的版本升级至当前版本时，升级观察期该参数设置不生效，升级提交后生效且升级提交后该选项默认为off。

默认值：

- 新安装的数据库默认值为：

  'ALL,on(SHOW_BATCH_ERROR_DETAIL,INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION),off(AUTO_INCR_LP_INSERT)'。
- 该参数自507.0.0版本添加，从507.0.0之前的版本升级到当前版本，数据库默认值为：

  'ALL,on(SHOW_BATCH_ERROR_DETAIL),off(INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,AUTO_INCR_LP_INSERT,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION)'。
- 507.0.0之后的版本升级到当前版本，原有行为变更保持与升级前版本一致，新增行为变更默认关闭。

设置方式：该参数属于USERSET类型参数，不建议通过gs_guc工具来设置。设置方法如下：

首先，通过SHOW exec_behavior_knob来查看哪些行为变更是支持可控制的。例如，查询输出结果为：

```
gaussdb=# SHOW exec_behavior_knob;
        exec_behavior_knob
----------------------------------------

ALL,on(SHOW_BATCH_ERROR_DETAIL,INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,PGXC_MERGE_
UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION),off(AUTO_INCR_LP_INSERT)
(1 row)
```

支持可控制的行为变更使用大写来标识，特殊标识ALL用于对所有行为变更进行设置。可以使用on/off来控制行为变更是否生效。设置SHOW_BATCH_ERROR_DETAIL行为变更关闭，可使用如下命令：

```
gaussdb=# SET exec_behavior_knob='off(SHOW_BATCH_ERROR_DETAIL)';
gaussdb=# SHOW exec_behavior_knob;
        exec_behavior_knob
----------------------------------------

ALL,on(INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLS
QL_RETURN_EXCEPTION),off(SHOW_BATCH_ERROR_DETAIL,AUTO_INCR_LP_INSERT)
(1 row)
```

可以看到SHOW_BATCH_ERROR_DETAIL行为变更被关闭，其他行为变更保持原状态。

ALL标识相当于一个快捷操作，即对所有行为变更进行开启或关闭。

```
gaussdb=# SET exec_behavior_knob='off(ALL)';
SET
gaussdb=# SHOW exec_behavior_knob;
        exec_behavior_knob
----------------------------------------

ALL,on(),off(SHOW_BATCH_ERROR_DETAIL,INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,AUTO_INC
R_LP_INSERT,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION)
```


版权所有 © 华为云计算技术有限公司

```
(1 row)
gaussdb=# SET exec_behavior_knob = 'on(ALL)';
SET
gaussdb=# SHOW exec_behavior_knob;
         exec_behavior_knob
----------------------------------------

ALL,on(SHOW_BATCH_ERROR_DETAIL,INDEXBY_FUNC_OPTIMIZE,SYS_CB_PATH_NULL_CTRL,AUTO_INCR_LP
_INSERT,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION),off()
(1 row)
```

SHOW_BATCH_ERROR_DETAIL打开时，对于配置文件中未显式指定的参数选项将加载系统默认值。系统默认值中，默认打开的参数有SHOW_BATCH_ERROR_DETAIL和SYS_CB_PATH_NULL_CTRL。因此，可以看到SHOW_BATCH_ERROR_DETAIL行为变更被打开时，SYS_CB_PATH_NULL_CTRL参数为打开，INDEXBY_FUNC_OPTIMIZE、AUTO_INCR_LP_INSERT、PGXC_MERGE_UPDATE、PLSQL_RETURN_EXCEPTION参数为关闭。

```
gs_guc reload -Z datanode -N all -I all -c "exec_behavior_knob='on(SHOW_BATCH_ERROR_DETAIL)'"
gaussdb=# SHOW exec_behavior_knob;
         exec_behavior_knob
----------------------------------------

ALL,on(SHOW_BATCH_ERROR_DETAIL,SYS_CB_PATH_NULL_CTRL),off(INDEXBY_FUNC_OPTIMIZE,AUTO_INC
R_LP_INSERT,PGXC_MERGE_UPDATE,SORT_KEYS_CACHE,PLSQL_RETURN_EXCEPTION)
(1 row)
```

设置建议：

- SORT_KEYS_CACHE：请结合当前业务系统配置的**work_mem**和数据分布进行设置。

  须知

  开启多列缓存会按比例增加Sort算子的内存占用。增加的内存大小与数据行数及额外缓存的列数呈严格的线性关系，通过以下公式进行评估：

  额外内存增量 ≈ 总行数 * 10 *（总排序键个数 - 1）字节

  （注：10为单列缓存预估的字节开销基数）

  - 建议开启场景：

    前导列重复度高：当ORDER BY中靠前的排序列（前导列）包含大量重复值（如状态、部门等低基数字段）时，数据重复度越高，该特性的多列比对优化效果越显著。

    后续列物理位置靠后：当用于二次比对的后续排序列在表结构中的位置较靠后（即列下标较大）时。列下标越大，该特性的优化收益就越明显。
  - 建议关闭场景：

    内存资源紧张：当系统配置的**work_mem**较小，或者数据库整体内存资源较为紧张时，不建议开启。
- 其它：推荐使用默认值。

设置不当的风险与影响：关闭某个或者所有选项会引起对应选项的行为变更。

enable_cid_sync

参数说明：当开启此功能时，在执行顶层分布式PGXC计划时，下发DN的非只读语句会同步使用CN上的CID，保证本事务内修改元组的可见性。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启此功能。
- off：表示关闭此功能。

默认值：

- 新安装的数据库：on。
- 从历史版本升级上来的数据库：若历史版本不支持该GUC参数，升级后的默认值为off；若历史版本支持该GUC参数，升级后默认值保留原设置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭该开关，在分布式PGXC计划涉及一行SQL下发多个语句的场景下，可能会影响本事务内修改元组的可见性。

enable_startwith_prior_enhance

参数说明：该参数标识是否增强对PRIOR修饰列的控制。若打开该参数，在带有CONNECT BY子句的SELECT语句中，支持在SELECT查询列、DISTINCT子句、WHERE子句、GROUP BY子句、HAVING子句、WINDOW子句、ORDER BY子句使用PRIOR关键字修饰列，显示父行列值。同时，在下述三个场景下，会报错：

（1）在带有CONNECT BY子句的SELECT语句中，ON条件、ORDER SIBLINGS BY子句使用PRIOR修饰列。

（2）在不带有CONNECT BY子句的语句中，使用PRIOR修饰列。

（3）在支持子句（SELECT查询列、DISTINCT子句、WHERE子句、GROUP BY子句、HAVING子句、WINDOW子句、ORDER BY子句）中，使用PRIOR修饰伪列LEVEL、CONNECT_BY_ISCYCLE、CONNECT_BY_ISLEAF。

若关闭该参数，在CONNECT BY子句外使用PRIOR关键字修饰列等同于不添加PRIOR关键字，会显示当前行列值。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，在恢复前向兼容行为时关闭。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

ustore_consistent_read_opt

参数说明：控制Ustore顺序扫描一致性读策略。

参数类型：枚举类型

参数单位：无

取值范围：

- pbrcr：基于页面的行一致性读策略。
- fast_pbrcr: 优化了页面持锁逻辑和Undo遍历算法的基于页面的行一致性读策略。

默认值：pbrcr

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

parser_ignore_hints

参数说明：控制是否禁用对HINT的解析。

参数类型：布尔类型

参数单位：无

取值范围：

- on：在解析阶段禁用了对HINT的处理，使语句中的HINT不生效。该参数打开时影响SQL Patch的生效，不影响Outline与SPM Baseline。
- off: 在解析阶段正常对HINT处理。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启该参数后，可能导致经HINT或SQL Patch调优的语句回退到差的查询计划，从而导致性能劣化。

enable_partition_statistics_min_rows

参数说明：对分区表执行partition_mode为ALL的ANALYZE操作时，是否使用分区表上设置的partition_statistics_min_rows选项。

参数类型：布尔类型

参数单位：无

取值范围：

- on：可使用分区表上指定的partition_statistics_min_rows值作为非空分区采样的最小采样行数。
- off：不使用分区表上指定的partition_statistics_min_rows值来控制非空分区采样的最小行数。

版权所有 © 华为云计算技术有限公司

默认值：

- 新安装的数据库：on。
- 从历史版本升级上来的数据库：若历史版本不支持该GUC参数，升级后默认值为off；若历史版本支持该GUC参数，升级后默认值保留原设置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数后，可能导致partition_mode为ALL的ANALYZE操作，得到的分区级统计信息不够准确。具体行为可参考**ANALYZE | ANALYSE**章节。

6.3.9 错误报告和日志

6.3.9.1 记录日志的位置

log_destination

参数说明：GaussDB支持多种方法记录服务器日志，log_destination的取值为一个逗号分隔开的列表（如log_destination="stderr,csvlog"）。

参数类型：字符串

参数单位：无

取值范围：有效值为"stderr"、"csvlog"、"syslog"、"eventlog"。

- "stderr"，表示日志打印到屏幕。
- "csvlog"，表示日志的输出格式为“逗号分隔值”即CSV（Comma Separated Value）格式。使用csvlog记录日志的前提是将**logging_collector**设置为on，请参见**使用CSV格式写日志**。
- "syslog"，表示通过操作系统的syslog记录日志。GaussDB使用syslog的LOCAL0 ~ LOCAL7记录日志，请参见**syslog_facility**。使用syslog记录日志需在操作系统后台服务配置文件中添加代码：
  local0.* */var/log/gaussdb*
- "eventlog"，表示通过事件日志组件记录日志。仅在windows系统下支持。

默认值："stderr"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

logging_collector

参数说明：控制开启后端日志收集线程logger进行日志收集。该线程捕获到stderr或csvlog日志消息，将其写入日志文件。

这种记录日志的方法比将日志记录到syslog更加有效，因为某些类型的消息在syslog的输出中无法显示。例如动态链接库加载失败消息和脚本产生的错误消息。

版权所有 © 华为云计算技术有限公司

须知

将服务器日志发送到stderr时可以不使用logging_collector参数，此时日志消息会被发送到服务器的stderr指向的空间。这种方法的缺点是日志回滚困难，只适用于较小的日志容量。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启日志收集功能。
- off：表示关闭日志收集功能。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭时无法使用csvlog记录日志。

log_directory

参数说明：logging_collector设置为on时，log_directory决定存放服务器日志文件的目录。它可以是绝对路径，或者是相对路径（相对于数据目录的路径）。log_directory支持动态修改，可以通过gs_guc reload实现，仅sysadmin用户可以访问。

须知

- 当配置文件中log_directory的值为非法路径时，会导致集群无法重新启动。
- 通过gs_guc reload动态修改log_directory时，当指定路径为合法路径时，日志输出到新的路径下。当指定路径为非法路径时，日志输出到上一次合法的日志输出路径下而不影响数据库正常运行。此时即使指定的log_directory的值非法，也会写入到配置文件中。
- 在沙箱环境，路径中不可以包含/var/chroot，例如log的绝对路径是/var/chroot/var/lib/log/Ruby/gs_log/cn_log，则只需要设置为/var/lib/log/Ruby/gs_log/cn_log。

说明

- 合法路径：用户对此路径有读写权限。
- 非法路径：用户对此路径无读写权限。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：安装时指定。

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_filename

参数说明：logging_collector设置为on时，log_filename决定服务器运行日志文件的名称。通常日志文件名是按照strftime模式生成，因此可以用系统时间定义日志文件名，用%转义字符实现，仅sysadmin用户可以访问。

须知

- 建议使用%转义字符定义日志文件名称，否则难以对日志文件进行有效的管理。
- 当log_destination设为csvlog时，系统会生成附加了时间戳的日志文件名，文件格式为csv格式，例如“server_log.1093827753.csv”。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：gaussdb-%Y-%m-%d_%H%M%S.log

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_file_mode

参数说明：logging_collector设置为on时，log_file_mode设置服务器日志文件的权限。通常log_file_mode的取值是能够被chmod和umask系统调用接受的数字。

须知

- 使用此选项前请设置log_directory，将日志存储到数据目录之外的地方。
- 因日志文件可能含有敏感数据，故不能将其设为对外可读。

参数类型：整型

参数单位：无

取值范围：0000 ~ 0777（8进制计数，转化为十进制为0 ~ 511）。

说明

- 0600表示只允许服务器管理员读写日志文件。
- 0640表示允许管理员所在用户组成员只能读日志文件。

默认值：0600

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_truncate_on_rotation

参数说明：logging_collector设置为on时，log_truncate_on_rotation设置日志消息的写入方式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示GaussDB以覆盖写入的方式写服务器日志消息。
- off：表示GaussDB将日志消息附加到同名的现有日志文件上。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_rotation_age

参数说明：logging_collector设置为on时，log_rotation_age表示创建一个新日志文件的时间间隔。当现在的时间减去上次创建服务器日志的时间超过了log_rotation_age时，将生成一个新的日志文件。

说明

假设日志需要保留7天，每天生成一个日志文件，日志文件名设置为server_log.Mon、server_log.Tue等。第二周的周二生成的日志消息会覆盖写入到server_log.Tue。

设置方法：将**log_filename**设置为server_log.%a ，**log_truncate_on_rotation**设置为on，log_rotation_age设置为1440。

参数类型：整型

参数单位：min（分钟）

取值范围：0 ~ 35791394。其中0表示关闭基于时间的新日志文件的创建。

默认值：1440（即1d）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_rotation_size

参数说明：logging_collector设置为on时，log_rotation_size决定服务器日志文件的最大容量。当日志消息的总量超过日志文件最大容量时，服务器将生成一个新的日志文件。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：kB

取值范围：0 ~ 2097151。0表示关闭基于容量的新日志文件的创建。

默认值：20480（20MB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

syslog_facility

参数说明：**log_destination**设置为syslog时，syslog_facility配置使用syslog记录日志的“设备”。

参数类型：枚举类型

参数单位：无

取值范围：local0、local1、local2、local3、local4、local5、local6、local7。

默认值：local0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

syslog_ident

参数说明：**log_destination**设置为syslog时，syslog_ident表示在syslog日志中GaussDB日志消息的标识。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值："postgres"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

event_source

参数说明：该参数仅在windows环境下生效。**log_destination**设置为eventlog时，event_source设置在日志中GaussDB日志消息的标识。

参数类型：字符串

参数单位：无

取值范围：字符串

版权所有 © 华为云计算技术有限公司

默认值："PostgreSQL"

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.9.2 记录日志的时间

client_min_messages

参数说明：用于控制发送到客户端的消息级别，级别不低于参数设置值的所有消息会被发送。

参数类型：枚举类型

参数单位：无

取值范围：消息级别从低到高依次为debug5、debug4、debug3、debug2、debug1、log、info、notice、warning、error、fatal和panic。参数的详细信息请参见表6-23。

须知

- 通常，设置的级别越低，发送给客户端的消息就越多。
- 对于客户端日志消息：如果client_min_messages参数值设置为fatal或panic，该参数的实际值会被强制指定为error。client_min_messages无论设置为哪个级别，info级别消息都会发送到客户端。
- 客户端连接至CN节点时，debug（debug5、debug4、debug3、debug2、debug1）和log级别消息因安全和性能原因不会发送至客户端，如需查看这两类日志，请直接查看CN节点的服务器日志。

默认值：notice

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_min_messages

参数说明：用于控制写入服务器日志的消息级别，级别不低于参数设置值的所有消息会被写入。

参数类型：枚举类型

参数单位：无

取值范围：消息级别从低到高依次为debug5、debug4、debug3、debug2、debug1、log、info、notice、warning、error、fatal和panic。参数的详细信息请参见表6-23。

版权所有 © 华为云计算技术有限公司

须知

- 通常，设置的级别越低，写入服务器日志的消息就越多。
- 对于服务器日志消息：当log_min_messages设置级别不高于error时，log级别消息都会写入服务器日志。
- 由于频繁打印会影响系统性能和稳定性，参数级别debug1至debug5一般作为客户现场问题定位手段使用，且建议优先按照模块（参见**logging_module**）开启这些级别，并在后续及时恢复原有日志级别。

默认值：warning

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_min_error_statement

参数说明：用于控制写入服务器日志的SQL语句对应的消息级别，级别不低于参数设置值的消息对应的所有SQL语句会被写入。

参数类型：枚举类型

参数单位：无

取值范围：消息级别从低到高依次为debug5、debug4、debug3、debug2、debug1、log、info、notice、warning、error、fatal和panic。参数的详细信息请参见**表6-23**。

须知

- 通常，设置的级别越低，写入服务器日志的SQL语句就越多。
- 对于服务器日志SQL消息：当设置级别不高于error时，log级别消息对应的SQL语句都会写入服务器日志。
- 当SQL语句中包含用户个人信息时，若不需要记录报错的SQL语句，可以将log_min_error_statement设置为panic。

默认值：error

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_min_duration_statement

参数说明：当某条SQL语句的执行时间大于或者等于该参数设定的值时，这条SQL语句和其执行时间会一并被记录。

设置log_min_duration_statement可以很方便地跟踪需要优化的查询语句。对于使用扩展查询协议的客户端，语法分析、绑定、执行每一步所花的时间将被独立记录。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：ms（毫秒）

须知

当此参数与**log_statement**同时使用时，已经被log_statement记录的SQL语句不会被重复记录。在没有使用syslog情况下，推荐使用**log_line_prefix**记录PID或会话ID，方便通过PID或会话ID关联语句消息和执行时间消息。

取值范围：-1 ~ 2147483647

- 设置为-1，关闭此功能。
- 设置为0，记录所有SQL语句和其执行时间。
- 设置为正整数，当某条SQL语句的执行时间大于等于设置值时，这条SQL语句和其执行时间会一并被记录。

默认值：3000（即3s）

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值为整数时，默认单位为ms；取值如果带单位，必须为ms、s、min、h、d。例如，不带单位取值600，表示log_min_duration_statement为600ms；带单位取值10min，表示log_min_duration_statement为10min。

设置建议：设置为默认值。

设置不当的风险与影响：取值若超出范围，或者取值带有除ms、s、min、h、d以外的单位时，会有报错提示。

backtrace_min_messages

参数说明：用于控制写入服务器日志的函数堆栈信息的消息级别，级别不低于参数设置值的消息对应的所有堆栈信息会被写入。

参数类型：枚举类型

参数单位：无

取值范围：消息级别从低到高为debug5、debug4、debug3、debug2、debug1、log、info、notice、warning、error、fatal、panic。参数的详细信息请参见**表6-23**。

须知

- 通常，设置的级别越低，写入服务器日志的堆栈信息就越多。
- 该参数作为客户现场问题定位手段使用。由于频繁打印函数堆栈会影响到系统开销及稳定性，问题定位时建议将backtrace_min_messages设置为fatal或panic。

默认值：panic

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。如果设置的级别过低，在高并发场景下，线程并发打印堆栈可能影响业务正常执行。例如，如果在高并发建连时此参数设置不合理，会拖慢建连线程池处理效率，甚至导致数据库实例不可连接。

表6-23解释GaussDB中使用的消息安全级别。当日志输出到syslog或者eventlog(仅windows环境下，GaussDB版本不涉及该参数)时，GaussDB进行如表中的转换。

表 6-23 信息严重程度分类

信息严重程度类型	详细说明	系统日志	事件日志
debug[1-5]	报告详细调试信息。	DEBUG	INFORMATION
log	报告对数据库管理员有用的信息， 比如检查点操作统计信息。	INFO	INFORMATION
info	报告用户可能需求的信息， 比如在VACUUM VERBOSE过程中的信息。	INFO	INFORMATION
notice	报告可能对用户有帮助的信息， 比如， 长标识符的截断， 作为主键一部分创建的索引等。	NOTICE	INFORMATION
warning	报告警告信息， 比如在事务块范围之外的COMMIT。	NOTICE	WARNING
error	报告导致当前命令退出的错误。	WARNING	ERROR
fatal	报告导致当前会话终止的原因。	ERR	ERROR
panic	报告导致整个数据库被关闭的原因。	CRIT	ERROR

6.3.9.3 记录日志的内容

debug_print_parse

参数说明：用于控制打印解析树结果。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启打印解析树结果的功能。
- off：表示关闭打印解析树结果的功能。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

debug_print_rewritten

参数说明：用于控制打印查询重写结果。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启打印查询重写结果的功能。
- off：表示关闭打印查询重写结果的功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

debug_print_plan

参数说明：用于设置是否将查询的执行计划打印到日志中。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示将查询的执行计划打印到日志中。
- off：表示不将查询的执行计划打印到日志中。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

须知

- 只有当日志的级别为log及以上时，debug_print_parse、debug_print_rewritten和debug_print_plan的调试信息才会输出。当这些选项打开时，调试信息只会记录在服务器的日志中，而不会输出到客户端的日志中。通过设置**client_min_messages**和**log_min_messages**参数可以改变日志级别。
- 在打开debug_print_plan开关的情况下需尽量避免调用gs_encrypt_aes128及gs_decrypt_aes128函数，避免敏感参数信息在日志中泄露的风险。同时建议用户在打开debug_print_plan开关生成的日志中对gs_encrypt_aes128及gs_decrypt_aes128函数的参数信息进行过滤后再提供给外部维护人员定位，日志使用完成后请及时删除。

debug_pretty_print

参数说明：打开此参数对**debug_print_parse**、**debug_print_rewritten**和**debug_print_plan**产生的日志进行缩进，会生成易读且更长的输出格式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进行缩进。
- off：表示不进行缩进。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_checkpoints

参数说明：控制在服务器日志中记录检查点和重启点的信息。打开此参数时，服务器日志消息包含涉及检查点和重启点的统计量，其中包含需要写的缓存区的数量及写入所花费的时间等。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开此参数，服务器日志消息包含涉及检查点和重启点的统计量。
- off：表示关闭此参数，服务器日志消息不包含涉及检查点和重启点的统计量。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

log_connections

参数说明：控制是否记录客户端连接请求的日志信息。日志信息包括连接的IP、PORT、用户名、数据库名、GaussDB数据库响应建立连接流程中关键步骤的耗时等。

说明

部分客户端程序（例如gsql），在判断是否需要口令的时候会尝试连接两次，因此日志消息中可能存在重复的“connection receive”（收到连接请求）。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示记录日志信息。
- off：表示不记录日志信息。

默认值：off

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_disconnections

参数说明：控制记录客户端结束连接信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示记录信息。
- off：表示不记录信息。

默认值：off

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_duration

参数说明：控制是否记录每个已完成SQL语句的执行时间。对使用扩展查询协议的客户端，会记录语法分析、绑定和执行每一步所花费的时间。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示记录所有已完成SQL语句的执行时间。

版权所有 © 华为云计算技术有限公司

- off：表示不记录SQL语句的执行时间。

说明

该参数设置为on或**log_min_duration_statement**设置为0都会记录每个已完成SQL语句的执行时间，不同之处在于log_min_duration_statement会强制记录SQL语句。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_error_verbosity

参数说明：控制服务器日志中每条记录的消息写入的详细度。

参数类型：枚举类型

参数单位：无

取值范围：

- terse：表示不包括DETAIL、HINT、QUERY及CONTEXT错误信息的记录。
- verbose：表示包括SQLSTATE错误代码 、源代码文件名、函数名及产生错误所在的行号。
- default：表示包括DETAIL、HINT、QUERY及CONTEXT错误信息的记录，不包括SQLSTATE错误代码 、源代码文件名、函数名及产生错误所在的行号。

默认值：default

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据对日志消息的详细度要求进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_hostname

参数说明：此参数关闭状态下，连接消息日志只显示正在连接主机的IP地址。打开此参数，表示会同时记录主机名。由于解析主机名需要一定的时间，可能影响数据库的性能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示记录主机名。
- off：表示不记录主机名。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：开启时可能影响数据库的性能。

log_line_prefix

参数说明：控制每条日志信息的前缀格式。日志前缀类似于printf风格的字符串，在日志的每行开头输出。用以%为开头的“转义字符”表示状态信息，具体请参见**表6-24**。

表 6-24 转义字符表

转义字符	状态信息
%a	应用程序名称。
%u	用户名。
%d	数据库名。
%r	远端主机名或者IP地址以及远端端口，在不启动log_hostname时显示IP地址及远端端口。
%h	远端主机名或者IP地址，在不启动log_hostname时只显示IP地址。
%p	线程ID。
%t	时间戳（不带毫秒，Windows上没有时区）。
%m	带毫秒的时间戳。
%n	表示指定错误上报的节点。
%i	命令标签：会话当前执行的命令类型。
%e	SQLSTATE错误码。
%c	会话ID，详见说明。
%l	每个会话或线程的日志编号，从1开始。
%s	线程启动时间。
%v	虚拟事务ID（backendID/ localXID）
%x	事务ID（0表示没有分配事务ID）。
%q	不产生任何输出。如果当前线程是后端线程，忽略这个转义序列，继续处理后面的转义序列；如果当前线程不是后端线程，忽略这个转义序列和它后面的所有转义序列。
%S	会话ID。
%T	Trace ID。
%%	字符%。

版权所有 © 华为云计算技术有限公司

说明

转义字符%c打印的会话ID，由两个4字节的十六进制数组成，通过字符“.”分开。这两个十六进制数分别表示线程的启动时间及线程编号，所以%c也可以用作一种更节省空间的记录线程启动时间和编号的方式。比如，可以用下面的查询从pg_stat_activity中生成这种十六进制的会话ID：

```
SELECT to_hex(EXTRACT(EPOCH FROM backend_start)::integer) || '.' ||
       to_hex(pid)
FROM pg_stat_activity;
```

- 当log_line_prefix设置为非空值时，需要保证最后一个字符是一个空格，以此来直观地与后续的日志行进行区分，也可以使用一个标点符号。
- Syslog会生成时间戳及线程ID信息，所以当登录日志时，不需要包含这些转义字符。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：'%m %n %u %d %h %p %S %x %a '

说明

%m %n %u %d %h %p %S %x %a 表示会话开始时间戳、错误上报节点、用户名、数据库名、远程主机名或IP、线程ID、会话ID、事务ID、应用程序名称。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据对日志状态信息的需求进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_lock_waits

参数说明：当一个会话的等待获得一个锁的时间超过**deadlock_timeout**的值时，此参数控制在数据库日志中记录此消息，有助于判断锁等待事件是否会导致死锁等问题。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示记录此信息。
- off：表示不记录此信息。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_statement

参数说明：控制记录SQL语句。对于使用扩展查询协议的客户端，记录接收到执行消息的事件和绑定参数的值（内置单引号要双写）。

版权所有 © 华为云计算技术有限公司

须知

- 即使log_statement设置为all，包含简单语法错误的语句也不会被记录，因为仅在完成基本的语法分析并确定了语句类型之后才记录日志。在使用扩展查询协议的情况下，执行阶段之前（语法分析或规划阶段）同样不会记录。将log_min_error_statement设为ERROR或更低才能记录这些语句。
- 设置该参数为非none时，可视为开启相关语句审计功能，数据库DBA可以访问服务端日志查看SQL执行记录。

参数类型：枚举类型

参数单位：无

取值范围：

- none：表示不记录SQL语句。
- ddl：表示记录所有的数据定义语句，比如CREATE、ALTER和DROP语句。
- mod：表示记录所有DDL语句，还包括数据修改语句INSERT、UPDATE、DELETE、TRUNCATE和COPY FROM 。
- all：表示记录所有语句，PREPARE、EXECUTE和EXPLAIN ANALYZE语句也同样被记录。

默认值：none

设置方式：该参数属于SUSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据是否需要记录SQL语句进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_temp_files

参数说明：控制记录比该参数设定值大的临时文件的删除信息。临时文件可以用来排序、哈希及临时查询结果。当一个临时文件被删除时，将会产生一条日志消息。

参数类型：整型

参数单位：kB

取值范围：-1 ~ 2147483647

- 正整数表示只记录比log_temp_files设定值大的临时文件的删除信息。
- 0表示记录所有的临时文件的删除信息。
- -1表示不记录任何临时文件的删除信息。

默认值：-1

设置方式：该参数属于SUSET类型参数，请参见表6-1中对应设置方法进行设置。取值为整数时，默认单位为kB；如果带单位，必须为kB、MB、GB。例如，不带单位取值16384，表示log_temp_files为16384kB；带单位取值16MB，表示log_temp_files为16MB。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置为非-1的值时，当满足记录要求的临时文件越多，日志记录也会越多，则会影响系统性能。取值若超出范围，或者取值带有除kB、MB、GB以外的单位时，会有报错提示。

log_timezone

参数说明：设置服务器写日志文件时使用的时区。与**TimeZone**不同，这个值是数据库范围的，针对所有连接到本数据库的会话生效。

参数类型：字符串

参数单位：无

取值范围：字符串，可查询视图**PG_TIMEZONE_NAMES**获得。

默认值：根据OS时区设置

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

gs_initdb进行相应系统环境设置时会对默认值进行修改。

logging_module

参数说明：用于设置或者显示模块日志在服务端的可输出性。该参数依赖于**log_min_messages**参数的设置，低于log_min_messages设置级别的模块日志不会打印。

参数类型：字符串

参数单位：无

取值范围：

ALL,on(),off(COMMAND,DFS,GUC,GSCLEAN,HDFS,ORC,SLRU,MEM_CTL,AUTOVAC,CACHE,ADIO,SSL,GDS,TBLSPC,WLM,OBS,INDEX,EXECUTOR,OPFUSION,GPC,GSC,VEC_EXECUTOR,STREAM,LLVM,OPT,OPT_REWRITE,CBOPT_REWRITE,OPT_JOIN,OPT_AGG,OPT_CHOICE,OPT_SUBPLAN,OPT_SETOP,OPT_SKEW,OPT_PLANNER,DISTOPT_PLANNER,OPT_STAT_EXT,OPT_SHIPPING,OPT_DSL_REWRITE,SPM,SPM_BG,SPM_KEY_FLOW,UDF,COOP_ANALYZE,WLMCP,ACCELERATE,MOT,PLANHINT,PARQUET,PGSTAT,CARBONDATA,SNAPSHOT,XACT,HANDLE,CLOG,EC,REMOTE,CN_RETRY,PLSQL,PLSQL_COMPILE,PLSQL_TRANS,TEXTSEARCH,SEQ,REDO,FUNCTION,PARSER,INSTR,WDR_SNAPSHOT,GSMONITOR,WDR_REPORT,ASP_REPORT,BLACKBOX,INCRE_CKPT,INCRE_BG_WRITER,DBL_WRT,RTO_RPO,HEARTBEAT,COMM_IPC,COMM_PARAM,TIMESERIES,SCHEMA,GTT,SEGMENT_PAGE,LIGHTPROXY,HOTKEY,THREAD_POOL,OPT_AI,WALRECEIVER,USTORE,UPAGE,UBTREE,UNDO,TIMECAPSULE,GEN_COL,DCF,AI4DB,DB4AI,ABO,MOD_ABOFEEDBACK,ASCEND,PLDEBUGGER,ADVISOR,SEC,SEC_FE,SEC_LEGER,SEC_POLICY,SEC_SDD,SEC_TDE,COMM_FRAMEWORK,COMM_PROXY,COMM_POOLER,COMM_STATUS,VACUUM,JOB,SPI,NEST_COMPILE,RESOWNER,GSSTACK,LOGICAL_DECODE,GPRC,DISASTER_READ,STANDBY_READ,SQL_APPLY,REPSYNC,SQLPATCH,PARTITION,UBT_NEWPAGE,GS_DEPENDENCY,GPI,LWLOCK,LOCK,UNIQUE_SQL,GSI,GLC,SRF,DBLINK,BARRIER_CREATOR,EXRTO_PAGE_P,SEQ_TUP_P,BT_TUP_P,DISPATCH_VERIFY,HBKT,HBKT_CBI,DBE_STATS,DBE_XMLGEN,MESSAGE_QUEUE,DBE_ALERT,UPDATE_DIST_COL,GS_ILM,GSPERF,PARAMETERIZE,EXEC_REMOTE,UBTREE_PARA,BTREE_PARA,ANTI_CACHE,ANTI_RECYCLER,VERIFYLOG,PARALLEL_BUILD,GS_REPAIR,AUTHID,CCINDEX,ONLINE_DDL,MTD,EXRTO_BUFFER,WAL,STREAM_PBE,HTAP,UALT,GTM_LITE_READ_S,UGIN,LOCAL_CACHE,BIINDEX,ONDEMAND,DATEA,TURBO_COMPRESS,GS_SIGNAL,AUTOHINT,EXRTO_READ_RECY,CHARACTER,BUFFILE,BUFFER,PAGE,SELCACHE,BYTECODE,UTIL,FUNC_DEPEND,MOD_DBE_LOB,EXEC_CACHE,ROWID,WAL_IMAGE_CHECK,CHECK_XLOG_GEN,SHARE_STORAGE,MOD_TOOLKIT,RESULT_CHECK)

版权所有 © 华为云计算技术有限公司

表 6-25 logging_module 部分模块说明

模块名称	模块说明
COMM_IPC	通信性能问题定位开关，该参数设置是否打印各个节点之间通信的报文收发情况。
COMM_PARAM	通信性能问题定位开关，该参数设置是否打印节点通信过程中session参数设置情况。

须知

当前版本CN_RETRY、BYTECODE不生效。

默认值："off(ALL)"。即默认关闭所有模块日志。

设置方式：该参数属于会话级参数，不建议通过gs_guc工具来设置。

首先，可以通过SHOW logging_module来查看哪些模块是支持可控制的。例如，查询输出结果为：

```
gaussdb=# show logging_module;
logging_module
------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------
ALL,on(),off(COMMAND,DFS,GUC,GSCLEAN,HDFS,ORC,SLRU,MEM_CTL,AUTOVAC,CACHE,ADIO,SSL,GDS,TBL
SPC,WLM,OBS,INDEX,EXECUTOR,OPFUSION,GPC,GSC,VEC_EXECUTOR,STREAM,LLVM,OPT,OPT_REWRITE,CB
OPT_REWRITE,OPT_JOIN,OPT_AGG,OPT_CHOICE,OPT_SUBPLAN,OPT_SETOP,OPT_SKEW,OPT_PLANNER,DIST
OPT_PLANNER,OPT_STAT_EXT,OPT_SHIPPING,OPT_DSL_REWRITE,SPM,SPM_BG,SPM_KEY_FLOW,UDF,COOP_
ANALYZE,WLMCP,ACCELERATE,MOT,PLANHINT,PARQUET,PGSTAT,CARBONDATA,SNAPSHOT,XACT,HANDLE,C
LOG,EC,REMOTE,CN_RETRY,PLSQL,PLSQL_COMPILE,PLSQL_TRANS,TEXTSEARCH,SEQ,REDO,FUNCTION,PARS
ER,INSTR,WDR_SNAPSHOT,GSMONITOR,WDR_REPORT,ASP_REPORT,BLACKBOX,INCRE_CKPT,INCRE_BG_WRI
TER,DBL_WRT,RTO_RPO,HEARTBEAT,COMM_IPC,COMM_PARAM,TIMESERIES,SCHEMA,GTT,SEGMENT_PAGE,L
IGHTPROXY,HOTKEY,THREAD_POOL,OPT_AI,WALRECEIVER,USTORE,UPAGE,UBTREE,UNDO,TIMECAPSULE,GE
N_COL,DCF,AI4DB,DB4AI,ABO,MOD_ABOFEEDBACK,ASCEND,PLDEBUGGER,ADVISOR,SEC,SEC_FE,SEC_LEGER,
SEC_POLICY,SEC_SDD,SEC_TDE,COMM_FRAMEWORK,COMM_PROXY,COMM_POOLER,COMM_STATUS,VACU
UM,JOB,SPI,NEST_COMPILE,RESOWNER,GSSTACK,LOGICAL_DECODE,GPRC,DISASTER_READ,STANDBY_READ,
SQL_APPLY,REPSYNC,SQLPATCH,PARTITION,UBT_NEWPAGE,GS_DEPENDENCY,GPI,LWLOCK,LOCK,UNIQUE_S
QL,GSI,GLC,SRF,DBLINK,BARRIER_CREATOR,EXRTO_PAGE_P,SEQ_TUP_P,BT_TUP_P,DISPATCH_VERIFY,HBKT,HB
KT_CBI,DBE_STATS,DBE_XMLGEN,MESSAGE_QUEUE,DBE_ALERT,UPDATE_DIST_COL,GS_ILM,GSPERF,PARAME
TERIZE,EXEC_REMOTE,UBTREE_PARA,BTREE_PARA,ANTI_CACHE,ANTI_RECYCLER,VERIFYLOG,PARALLEL_BUIL
D,GS_REPAIR,AUTHID,CCINDEX,ONLINE_DDL,MTD,EXRTO_BUFFER,WAL,STREAM_PBE,HTAP,UALT,GTM_LITE_
READ_S,UGIN,LOCAL_CACHE,BIINDEX,ONDEMAND,DATEA,TURBO_COMPRESS,GS_SIGNAL,AUTOHINT,EXRT
O_READ_RECY,CHARACTER,BUFFILE,BUFFER,PAGE,SELCACHE,BYTECODE,UTIL,FUNC_DEPEND,MOD_DBE_LO
B,EXEC_CACHE,ROWID,WAL_IMAGE_CHECK,CHECK_XLOG_GEN,SHARE_STORAGE,MOD_TOOLKIT,RESULT_CH
ECK)
(1 row)
```

支持可控制的模块使用大写来标识，特殊标识ALL用于对所有模块日志进行设置。可以使用on/off来控制模块日志的输出。设置SSL模块日志为可输出，使用如下命令：

```
gaussdb=# set logging_module='on(SSL)';
SET
gaussdb=# show
logging_module;
        logging_module
------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------
```


版权所有 © 华为云计算技术有限公司

```
-------------------------------------------------
ALL,on(SSL),off(COMMAND,DFS,GUC,GSCLEAN,HDFS,ORC,SLRU,MEM_CTL,AUTOVAC,CACHE,ADIO,GDS,TBL
SPC,WLM,OBS,INDEX,EXECUTOR,OPFUSION,GPC,GSC,VEC_EXECUTOR,STREAM,LLVM,OPT,OPT_REWRITE,CB
OPT_REWRITE,OPT_JOIN,OPT_AGG,OPT_CHOICE,OPT_SUBPLAN,OPT_SETOP,OPT_SKEW,OPT_PLANNER,DIST
OPT_PLANNER,OPT_STAT_EXT,OPT_SHIPPING,OPT_DSL_REWRITE,SPM,SPM_BG,SPM_KEY_FLOW,UDF,COOP_
ANALYZE,WLMCP,ACCELERATE,MOT,PLANHINT,PARQUET,PGSTAT,CARBONDATA,SNAPSHOT,XACT,HANDLE,C
LOG,EC,REMOTE,CN_RETRY,PLSQL,PLSQL_COMPILE,PLSQL_TRANS,TEXTSEARCH,SEQ,REDO,FUNCTION,PARS
ER,INSTR,WDR_SNAPSHOT,GSMONITOR,WDR_REPORT,ASP_REPORT,BLACKBOX,INCRE_CKPT,INCRE_BG_WRI
TER,DBL_WRT,RTO_RPO,HEARTBEAT,COMM_IPC,COMM_PARAM,TIMESERIES,SCHEMA,GTT,SEGMENT_PAGE,L
IGHTPROXY,HOTKEY,THREAD_POOL,OPT_AI,WALRECEIVER,USTORE,UPAGE,UBTREE,UNDO,TIMECAPSULE,GE
N_COL,DCF,AI4DB,DB4AI,ABO,MOD_ABOFEEDBACK,ASCEND,PLDEBUGGER,ADVISOR,SEC,SEC_FE,SEC_LEGER,
SEC_POLICY,SEC_SDD,SEC_TDE,COMM_FRAMEWORK,COMM_PROXY,COMM_POOLER,COMM_STATUS,VACU
UM,JOB,SPI,NEST_COMPILE,RESOWNER,GSSTACK,LOGICAL_DECODE,GPRC,DISASTER_READ,STANDBY_READ,
SQL_APPLY,REPSYNC,SQLPATCH,PARTITION,UBT_NEWPAGE,GS_DEPENDENCY,GPI,LWLOCK,LOCK,UNIQUE_S
QL,GSI,GLC,SRF,DBLINK,BARRIER_CREATOR,EXRTO_PAGE_P,SEQ_TUP_P,BT_TUP_P,DISPATCH_VERIFY,HBKT,HB
KT_CBI,DBE_STATS,DBE_XMLGEN,MESSAGE_QUEUE,DBE_ALERT,UPDATE_DIST_COL,GS_ILM,GSPERF,PARAME
TERIZE,EXEC_REMOTE,UBTREE_PARA,BTREE_PARA,ANTI_CACHE,ANTI_RECYCLER,VERIFYLOG,PARALLEL_BUIL
D,GS_REPAIR,AUTHID,CCINDEX,ONLINE_DDL,MTD,EXRTO_BUFFER,WAL,STREAM_PBE,HTAP,UALT,GTM_LITE_
READ_S,UGIN,LOCAL_CACHE,BIINDEX,ONDEMAND,DATEA,TURBO_COMPRESS,GS_SIGNAL,AUTOHINT,EXRT
O_READ_RECY,CHARACTER,BUFFILE,BUFFER,PAGE,SELCACHE,BYTECODE,UTIL,FUNC_DEPEND,MOD_DBE_LO
B,EXEC_CACHE,ROWID,WAL_IMAGE_CHECK,CHECK_XLOG_GEN,SHARE_STORAGE,MOD_TOOLKIT,RESULT_CH
ECK)
(1 row)
```

可以看到模块SSL的日志输出被打开。

ALL标识相当于一个快捷操作，即对所有模块的日志可输出进行开启或关闭。

```
gaussdb=# set logging_module='off(ALL)';
SET
gaussdb=# show
logging_module;
         logging_module
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------
ALL,on(),off(COMMAND,DFS,GUC,GSCLEAN,HDFS,ORC,SLRU,MEM_CTL,AUTOVAC,CACHE,ADIO,SSL,GDS,TBL
SPC,WLM,OBS,INDEX,EXECUTOR,OPFUSION,GPC,GSC,VEC_EXECUTOR,STREAM,LLVM,OPT,OPT_REWRITE,CB
OPT_REWRITE,OPT_JOIN,OPT_AGG,OPT_CHOICE,OPT_SUBPLAN,OPT_SETOP,OPT_SKEW,OPT_PLANNER,DIST
OPT_PLANNER,OPT_STAT_EXT,OPT_SHIPPING,OPT_DSL_REWRITE,SPM,SPM_BG,SPM_KEY_FLOW,UDF,COOP_
ANALYZE,WLMCP,ACCELERATE,MOT,PLANHINT,PARQUET,PGSTAT,CARBONDATA,SNAPSHOT,XACT,HANDLE,C
LOG,EC,REMOTE,CN_RETRY,PLSQL,PLSQL_COMPILE,PLSQL_TRANS,TEXTSEARCH,SEQ,REDO,FUNCTION,PARS
ER,INSTR,WDR_SNAPSHOT,GSMONITOR,WDR_REPORT,ASP_REPORT,BLACKBOX,INCRE_CKPT,INCRE_BG_WRI
TER,DBL_WRT,RTO_RPO,HEARTBEAT,COMM_IPC,COMM_PARAM,TIMESERIES,SCHEMA,GTT,SEGMENT_PAGE,L
IGHTPROXY,HOTKEY,THREAD_POOL,OPT_AI,WALRECEIVER,USTORE,UPAGE,UBTREE,UNDO,TIMECAPSULE,GE
N_COL,DCF,AI4DB,DB4AI,ABO,MOD_ABOFEEDBACK,ASCEND,PLDEBUGGER,ADVISOR,SEC,SEC_FE,SEC_LEGER,
SEC_POLICY,SEC_SDD,SEC_TDE,COMM_FRAMEWORK,COMM_PROXY,COMM_POOLER,COMM_STATUS,VACU
UM,JOB,SPI,NEST_COMPILE,RESOWNER,GSSTACK,LOGICAL_DECODE,GPRC,DISASTER_READ,STANDBY_READ,
SQL_APPLY,REPSYNC,SQLPATCH,PARTITION,UBT_NEWPAGE,GS_DEPENDENCY,GPI,LWLOCK,LOCK,UNIQUE_S
QL,GSI,GLC,SRF,DBLINK,BARRIER_CREATOR,EXRTO_PAGE_P,SEQ_TUP_P,BT_TUP_P,DISPATCH_VERIFY,HBKT,HB
KT_CBI,DBE_STATS,DBE_XMLGEN,MESSAGE_QUEUE,DBE_ALERT,UPDATE_DIST_COL,GS_ILM,GSPERF,PARAME
TERIZE,EXEC_REMOTE,UBTREE_PARA,BTREE_PARA,ANTI_CACHE,ANTI_RECYCLER,VERIFYLOG,PARALLEL_BUIL
D,GS_REPAIR,AUTHID,CCINDEX,ONLINE_DDL,MTD,EXRTO_BUFFER,WAL,STREAM_PBE,HTAP,UALT,GTM_LITE_
READ_S,UGIN,LOCAL_CACHE,BIINDEX,ONDEMAND,DATEA,TURBO_COMPRESS,GS_SIGNAL,AUTOHINT,EXRT
O_READ_RECY,CHARACTER,BUFFILE,BUFFER,PAGE,SELCACHE,BYTECODE,UTIL,FUNC_DEPEND,MOD_DBE_LO
B,EXEC_CACHE,ROWID,WAL_IMAGE_CHECK,CHECK_XLOG_GEN,SHARE_STORAGE,MOD_TOOLKIT,RESULT_CH
ECK)
(1 row)

gaussdb=# set logging_module='on(ALL)';
SET
gaussdb=# show
logging_module;
        logging_module
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------
ALL,on(COMMAND,DFS,GUC,GSCLEAN,HDFS,ORC,SLRU,MEM_CTL,AUTOVAC,CACHE,ADIO,SSL,GDS,TBLSPC,
WLM,OBS,INDEX,EXECUTOR,OPFUSION,GPC,GSC,VEC_EXECUTOR,STREAM,LLVM,OPT,OPT_REWRITE,CBOPT_
REWRITE,OPT_JOIN,OPT_AGG,OPT_CHOICE,OPT_SUBPLAN,OPT_SETOP,OPT_SKEW,OPT_PLANNER,DISTOPT_
```

 版权所有 © 华为云计算技术有限公司

```
PLANNER,OPT_STAT_EXT,OPT_SHIPPING,OPT_DSL_REWRITE,SPM,SPM_BG,SPM_KEY_FLOW,UDF,COOP_ANAL
YZE,WLMCP,ACCELERATE,MOT,PLANHINT,PARQUET,PGSTAT,CARBONDATA,SNAPSHOT,XACT,HANDLE,CLOG,E
C,REMOTE,CN_RETRY,PLSQL,PLSQL_COMPILE,PLSQL_TRANS,TEXTSEARCH,SEQ,REDO,FUNCTION,PARSER,INS
TR,WDR_SNAPSHOT,GSMONITOR,WDR_REPORT,ASP_REPORT,BLACKBOX,INCRE_CKPT,INCRE_BG_WRITER,DB
L_WRT,RTO_RPO,HEARTBEAT,COMM_IPC,COMM_PARAM,TIMESERIES,SCHEMA,GTT,SEGMENT_PAGE,LIGHTP
ROXY,HOTKEY,THREAD_POOL,OPT_AI,WALRECEIVER,USTORE,UPAGE,UBTREE,UNDO,TIMECAPSULE,GEN_COL
,DCF,AI4DB,DB4AI,ABO,MOD_ABOFEEDBACK,ASCEND,PLDEBUGGER,ADVISOR,SEC,SEC_FE,SEC_LEGER,SEC_P
OLICY,SEC_SDD,SEC_TDE,COMM_FRAMEWORK,COMM_PROXY,COMM_POOLER,COMM_STATUS,VACUUM,JO
B,SPI,NEST_COMPILE,RESOWNER,GSSTACK,LOGICAL_DECODE,GPRC,DISASTER_READ,STANDBY_READ,SQL_A
PPLY,REPSYNC,SQLPATCH,PARTITION,UBT_NEWPAGE,GS_DEPENDENCY,GPI,LWLOCK,LOCK,UNIQUE_SQL,GSI,
GLC,SRF,DBLINK,BARRIER_CREATOR,EXRTO_PAGE_P,SEQ_TUP_P,BT_TUP_P,DISPATCH_VERIFY,HBKT,HBKT_CBI,
DBE_STATS,DBE_XMLGEN,MESSAGE_QUEUE,DBE_ALERT,UPDATE_DIST_COL,GS_ILM,GSPERF,PARAMETERIZE,
EXEC_REMOTE,UBTREE_PARA,BTREE_PARA,ANTI_CACHE,ANTI_RECYCLER,VERIFYLOG,PARALLEL_BUILD,GS_RE
PAIR,AUTHID,CCINDEX,ONLINE_DDL,MTD,EXRTO_BUFFER,WAL,STREAM_PBE,HTAP,UALT,GTM_LITE_READ_S,
UGIN,LOCAL_CACHE,BIINDEX,ONDEMAND,DATEA,TURBO_COMPRESS,GS_SIGNAL,AUTOHINT,EXRTO_READ_
RECY,CHARACTER,BUFFILE,BUFFER,PAGE,SELCACHE,BYTECODE,UTIL,FUNC_DEPEND,MOD_DBE_LOB,EXEC_C
ACHE,ROWID,WAL_IMAGE_CHECK,CHECK_XLOG_GEN,SHARE_STORAGE,MOD_TOOLKIT,RESULT_CHECK),off(
)
(1 row)
```

设置建议：根据需要开启对应模块日志的打印。不建议开启所有模块的日志打印（即：'on(ALL)'）。

设置不当的风险与影响：开启时可能会打印较多对应模块相关的日志，影响系统性能。

enable_unshipping_log

参数说明：用于控制是否打印语句不下推的日志，主要用于帮助用户定位不下推语句可能导致的性能问题。当**enable_stream_operator**参数关闭时，如果此参数设置为on，会有大量关于计划不能下推的日志记录到日志文件中。如果用户不需要这些日志内容，建议用户在enable_stream_operator参数关闭时，也同时关闭enable_unshipping_log参数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打印日志。
- off：表示不打印日志。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

opfusion_debug_mode

参数说明：用于调试简单查询是否进行查询优化。设置成log级别可以在DN的执行计划中看到没有查询优化的具体原因。

参数类型：枚举类型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- off：表示不打开该功能。
- log：表示打开该功能，可以在DN的执行计划中看到没有查询优化的具体原因。

须知

- 需要设置参数**max_datanode_for_plan**才能看到DN的执行计划。
- 提供在log中显示语句没有查询优化的具体原因，需要将参数设置成log级别，**log_min_messages**设置成debug4级别，**logging_module**设置'on(OPFUSION)'，注意log内容可能会比较多，尽可能在调优期间执行少量作业使用。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_debug_vacuum

参数说明：允许输出一些与VACUUM相关的日志，便于定位VACUUM相关问题。开发人员专用，不建议普通用户使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启此日志开关。
- off：表示关闭此日志开关。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_err_stack

参数说明：控制ERROR级别中ACTION为"Contact Huawei technical support."的错误码报错是否打印堆栈。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启此日志开关。
- off：表示关闭此日志开关。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_show_err_code

参数说明：控制gsql等前端连接是否显示报错的错误码。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启此开关。
- off：表示关闭此开关。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.9.4 使用 CSV 格式写日志

前提条件

- **log_destination**的值设置为csvlog。
- **logging_collector**的值设置为on。

csvlog 定义

以“逗号分隔值” 即CSV（Comma Separated Value）的形式发出日志。

以下是简单的用来存储CSV形式日志输出的表定义：

```
CREATE TABLE gaussdb_log
(
log_time timestamp(3) with time zone,
node_name text,
user_name text,
database_name text,
process_id bigint,
connection_from text,
"session_id" text,
session_line_num bigint,
command_tag text,
session_start_time timestamp with time zone,
virtual_transaction_id text,
transaction_id bigint,
query_id bigint,
module text,
error_severity text,
sql_state_code text,
message text,
detail text,
hint text,
internal_query text,
internal_query_pos integer,
context text,
query text,
```


版权所有 © 华为云计算技术有限公司

```
query_pos integer,
location text,
application_name text
);
```

详细说明请参见表6-26。

表 6-26 csvlog 字段含义表

字段名	字段含义	字段名	字段含义
log_time	毫秒级的时间戳	module	日志所属模块
node_name	节点名称	error_severity	ERRORSTATE代码
user_name	用户名	sql_state_code	SQLSTATE代码
database_name	数据库名	message	错误消息
process_id	线程ID	detail	详细错误消息
connection_from	客户主机:端口号	hint	提示
session_id	会话ID	internal_query	内部查询（查询那些导致错误的信息，如果有的话）
session_line_num	每个会话的行数	internal_query_pos	内部查询指针
command_tag	命令标签	context	环境
session_start_time	会话开始时间	query	错误发生位置的字符统计
virtual_transaction_id	常规事务	query_pos	错误发生位置指针
transaction_id	事务ID	location	在GaussDB源代码中错误的位置（如果log_error_verbosity的值设为verbose ）
query_id	查询ID	application_name	应用名称

使用COPY FROM命令将日志文件导入这个表：

```
COPY postgres_log FROM '/opt/data/gs_log/logfile.csv' WITH csv;
```

说明

此处的日志名“logfile.csv”要换成实际生成的日志的名称。

版权所有 © 华为云计算技术有限公司

简化输入

简化输入到CSV日志文件，可以通过如下操作：

- 设置**log_filename**和**log_rotation_age**，为日志文件提供一个一致的、可预测的命名方案。通过日志文件名，预测一个独立的日志文件完成并进入准备导入状态的时间。
- 将**log_rotation_size**设为0来终止基于尺寸的日志回滚，因为基于尺寸的日志回滚让预测日志文件名变得非常的困难。
- 将**log_truncate_on_rotation**设为on以便区分在同一日志文件中旧的日志数据和新的日志数据。

6.3.10 告警上报

在集群运行的过程中，会对数据库中的错误场景进行上报，便于用户及早感知到数据库集群的错误。告警写入的system_alarm日志可以在$GAUSSLOG/cm、$GAUSSLOG/gs_log/gtm、$GAUSSLOG/gs_log或$GAUSSLOG/roach/agent路径下查看。

enable_alarm

参数说明：设置是否打开告警上报线程，上报数据库中可能的故障场景。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开告警上报线程。
- off：表示关闭告警上报线程。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数会导致用户无法感知数据库中的某些故障场景。

说明

该参数生效范围节点仅为CN、DN。

connection_alarm_rate

参数说明：设置允许和数据库连接的最大并发连接数的比率限制。数据库连接的最大并发连接数为**max_connections** * connection_alarm_rate。当并发连接数超过**max_connections** * connection_alarm_rate时，后台会打印连接信息的日志connection_log，日志中可以查看pg_stat_activity视图和线程池视图LOCAL_THREADPOOL_STATUS（查看pg_stat_activity视图需要开启追踪功能，查看线程池视图LOCAL_THREADPOOL_STATUS需要开启线程池功能）。

参数类型：浮点型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 1

默认值：0.9

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

alarm_report_interval

参数说明：指定告警上报的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。设置时取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：

- 该值设置过大时，若指定告警上报时间间隔内检测出多次相同告警，会丢失关键告警。
- 该值设置过小时，若指定告警上报时间间隔内持续检测出同一告警，告警会频繁上报。

alarm_component

参数说明：在告警上报时，会进行告警抑制，即同一个实例的同一个告警项在**alarm_report_interval**（默认值为10s）内不做重复上报。在这种情况下设置用于处理告警内容的告警组件的位置，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法文件名称。

- 若前置脚本gs_preinstall中的--alarm-type参数设置为5时，表示未对接第三方组件，告警写入system_alarm日志，此时GUC参数alarm_component的取值为：/opt/huawei/snas/bin/snas_cm_cmd。
- 若前置脚本gs_preinstall中的--alarm-type参数设置为1时，表示对接第三方组件，此时GUC参数alarm_component的值为第三方组件的可执行程序的绝对路径。

默认值："/opt/huawei/snas/bin/snas_cm_cmd"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。设置时取值不能带单位。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

table_skewness_warning_threshold

参数说明：设置用于表倾斜告警的阈值。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，并根据业务场景对表倾斜的敏感度适时调整。

设置不当的风险与影响：设置较小时，可能对表倾斜更敏感，无显著风险。

table_skewness_warning_rows

参数说明：设置用于表倾斜告警的行数。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：100000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，并根据业务场景对表倾斜的敏感度适时调整。

设置不当的风险与影响：设置较小时，可能对表倾斜更敏感，无显著风险。

6.3.11 运行时统计

6.3.11.1 查询和索引统计收集器

查询和索引统计收集器负责收集数据库系统运行中的统计数据，如在一个表和索引上进行了多少次插入与更新操作、磁盘块的数量和元组的数量、每个表上最近一次执行清理和分析操作的时间等。可以通过查询系统视图pg_stats和系统表pg_statistic查看统计数据。下面的参数设置服务器范围内的统计收集特性。

track_activities

参数说明：控制是否收集每个会话中当前正在执行命令的统计数据。对于存储过程，打开该参数后，可以通过pg_stat_activity视图看到存储过程内正在执行的perform语句、调用存储过程语句、存储过程内的SQL语句、OPEN CURSOR语句。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示开启收集功能。
- off：表示关闭收集功能。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数设置为off会影响存储引擎空间回收能力，导致空间膨胀。

track_counts

参数说明：控制是否收集数据库活动的统计数据。在AutoVacuum自动清理线程中选择清理的数据库时，需要数据库的统计数据，故默认值设置为on。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启收集功能。
- off：表示关闭收集功能。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数设置为off会影响存储引擎空间回收能力，导致空间膨胀。

track_procedure_sql

参数说明：该参数控制pg_stat_activity系统视图中的query列是否同时打印该存储过程中正在执行的SQL语句。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示调用存储过程时，pg_stat_activity的query列会同时打印存储过程正在执行的语句。
- off：表示调用存储过程时，pg_stat_activity的query列只打印存储过程调用语句。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

track_io_timing

参数说明：控制是否收集数据库I/O调用时序的统计数据。I/O时序统计数据可以在pg_stat_database中查询。开启该参数时，收集器会在当前时间重复查询操作系统的信息，这可能会引起某些平台的重大开销，故默认值设置为off。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启收集功能。
- off：表示关闭收集功能。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_functions

参数说明：控制是否收集函数的调用次数和调用耗时的统计数据。

说明

当SQL语言函数设置为调用查询的“内联”函数时，不管是否开启该参数，这些SQL语言函数都无法被追踪到。

参数类型：枚举类型

参数单位：无

取值范围：

- pl：表示只追踪过程语言函数。
- all：表示追踪SQL语言函数。
- none：表示关闭函数追踪功能。

默认值：none

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_activity_query_size

参数说明：用于跟踪每一个活动会话中当前正在执行命令的字节数。如果实际命令字节数大于该值，会进行截断。

参数类型：整型

参数单位：Byte

取值范围：100 ~ 102400

版权所有 © 华为云计算技术有限公司

默认值：1024

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务场景进行设置。

设置不当的风险与影响：调高该参数会导致共享内存占用增大，参数设置过大可能会导致其他模块内存不足。请在充分理解参数含义，并经过测试验证后进行修改。

update_process_title

参数说明：控制是否收集每次服务器因接收到一个新的SQL语句而产生的线程名称更新的统计数据。

线程名称可以通过ps命令进行查看，在Windows下通过任务管理器查看。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启收集功能。
- off：表示关闭收集功能。

默认值：off

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

stats_temp_directory

参数说明：设置存储临时统计数据的目录，仅sysadmin用户可以访问。

说明

将该参数设置为一个基于RAM的文件系统目录会减少实际的I/O开销并提升其性能。

参数类型：字符串

参数单位：无

取值范围：合法的目录。

默认值："pg_stat_tmp"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_gsstat_cache

参数说明：是否对PgStat函数启用缓存加速。

版权所有 © 华为云计算技术有限公司

PgStat函数包含：pg_stat_get_numscans、pg_stat_get_tuples_returned、pg_stat_get_tuples_fetched、pg_stat_get_tuples_inserted、pg_stat_get_tuples_updated、pg_stat_get_tuples_deleted、pg_stat_get_tuples_hot_updated、pg_stat_get_live_tuples、pg_stat_get_dead_tuples、pg_stat_get_autovacuum_count、pg_stat_get_autoanalyze_count、pg_stat_get_blocks_fetched、pg_stat_get_blocks_hit、pg_timestamp stat_get_last_autoanalyze_time、pg_timestamp stat_get_last_autovacuum_time、pg_timestamp stat_get_last_updated。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示针对PgStat函数开启缓存加速功能。
- off：表示针对PgStat函数关闭缓存加速功能。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在该参数关闭的情况下，使用上述函数的PgStat的相关视图查询比较慢时，建议开启该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gsstat_cache_size

参数说明：获取特定分区的统计信息时缓存加速所占用的内存最大值。

参数类型：整型

参数单位：兆字节(Megabytes)

取值范围：1 ~ 10240

默认值：1024

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务场景进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_thread_wait_status_interval

参数说明：用来定期收集thread状态信息的时间间隔。

参数类型：整型

参数单位：min（分钟）

取值范围：0 ~ 1440

默认值：30

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为100，不带单位，表示100min；设置为1d，带单位，表示1d。取值如果要带单位，必须为min、h、d。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_save_datachanged_timestamp

参数说明：确定是否收集insert/update/delete、exchange/truncate/drop partition操作对表数据改动的时间。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许收集相关操作对表数据改动的时间。
- off：表示禁止收集相关操作对表数据改动的时间。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

plan_collect_thresh

参数说明：控制收集每个会话中当前正在执行计划的统计数据。

参数类型：整型

参数单位：无

取值范围：-1 ~ 2147483647

- -1：表示不收集运行态计划。
- 0：表示若当前SQL执行时间大于等于GUC参数log_min_duration_statement指定值，且计划中所有算子处理的tuple数量和大于等于10000时，开始收集运行态计划；且后续每当所有算子处理tuple数量的和的增量超过10000时，进行一次收集。
- >0：表示当计划中所有算子增量返回tuple数量之和大于等于该值时，收集一次运行态计划。

默认值：0

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_sql_count

参数说明：控制是否对每个会话中当前正在执行的SELECT、INSERT、UPDATE、DELETE、MERGE INTO语句进行计数。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示开启计数功能。
- off：表示关闭计数功能。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

当参数关闭时，查询视图的结果为0行。

6.3.11.2 热点 key 统计

分布式架构下，如果应用短时间内集中访问某一节点，会导致该节点资源使用过高，从而影响数据库正常运行。GaussDB提供的热点key快速检测功能可以用来快速定位是否有热点key以及热点key的分布来帮助定位问题。

enable_hotkeys_collection

参数说明：开关打开后，自动对数据库内被访问的键值进行统计。

说明

- 如果是通过gs_guc set 方式设置参数，需要重启数据库使得GUC参数生效，重启数据库时会清理热点key信息。
- 当GUC参数关闭时，调用热点key查询结果将会返回空，并且提示GUC参数关闭。但是开关关闭时，热点key清理接口仍可以正常使用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启计数功能。
- off：表示关闭计数功能。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需要使用热点检测功能时打开开关。

设置不当的风险与影响：开关打开后，TPCC性能略微下降，控制在5%以内，每个实例会占用500M内存资源。

6.3.12 负载管理

未对数据库资源做控制时，容易出现并发任务抢占资源导致操作系统过载甚至最终崩溃。操作系统过载时，其响应用户任务的速度会变慢甚至无响应；操作系统崩溃时，

版权所有 © 华为云计算技术有限公司

整个系统将无法对用户提供任何服务。GaussDB的负载管理功能能够基于可用资源的多少均衡数据库的负载，以避免数据库系统过载。

use_workload_manager

参数说明：设置是否开启资源管理功能。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开资源管理。
- off：表示关闭资源管理。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：资源管理的总开关，关闭会影响很多资源管理的相关功能，不建议关闭。

说明

- 当使用表6-1中的方式二来修改参数值时，新参数值只能对更改操作执行后启动的线程生效。此外，对于后台线程以及线程复用执行的新作业，该参数值的改动不会生效。如果希望这类线程即时识别参数变化，可以使用kill session或重启节点的方式来实现。
- use_workload_manager参数由off变为on状态后，不会统计off时的存储资源。如果需要统计off时用户使用的存储资源，请在数据库中执行以下SQL语句：

```
SELECT gs_wlm_readjust_user_space(0);
```

enable_control_group

参数说明：设置是否开启Cgroups功能。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开Cgroups功能。
- off：表示关闭Cgroups功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置，当使用表中的方式二来修改参数值时，新参数值只能对更改操作执行后启动的线程生效。

设置建议：对于后台线程以及线程复用执行的新作业，该值的改动不会生效，如果希望这类线程即时识别参数变化，可以使用kill session或重启节点的方式来实现。

设置不当的风险与影响：同时开启该参数和资源管控功能（use_workload_manager）后会使用户资源管控功能生效，降低用户执行的作业的性能。

版权所有 © 华为云计算技术有限公司

enable_cgroup_switch

参数说明：控制数据库执行语句时是否根据类型自动切换到TopWD组。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示数据库执行语句时根据类型自动切换到TopWD组。
- off：表示数据库执行语句时根据类型不自动切换到TopWD组。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

当且仅当**enable_control_group**参数取值为on时，该参数设置有效。

cgroup_name

参数说明：设置当前使用的Cgroups的名称以及调整当前group下排队的优先级。

如果先设置cgroup_name，再设置session_respool，那么**session_respool**关联的控制组起作用，如果再修改cgroup_name，那么新修改的cgroup_name起作用。

修改cgroup_name的过程中如果指定Workload控制组级别，数据库不对级别进行验证。级别的范围只要在1-10范围内都可以。

参数类型：字符串

参数单位：无

取值范围：已有控制组的名称。

默认值："DefaultClass:Medium"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：尽量不要混合使用cgroup_name和**session_respool**。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

DefaultClass:Medium表示DefaultClass下Timeshare控制组中的Medium控制组。

enable_backend_control

参数说明：控制数据库常驻线程是否绑定到DefaultBackend和HaBackend控制组。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示常驻线程绑定到DefaultBackend和HaBackend控制组。
- off：表示常驻线程不绑定到DefaultBackend和HaBackend控制组。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

当且仅当**enable_control_group**参数取值为on时，该参数设置有效。HaBackend控制组用于心跳线程和Wal相关后台线程的CPU管控，保证集群高可用。

enable_vacuum_control

参数说明：控制数据库常驻线程中的autoVacuumWorker是否绑定到Vacuum控制组。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示数据库常驻线程中的autoVacuumWorker绑定到Vacuum控制组。
- off：表示数据库常驻线程中的autoVacuumWorker不绑定到Vacuum控制组。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

当且仅当**enable_control_group**参数取值为on时，该参数设置有效。

enable_perm_space

参数说明：设置是否开启perm space功能。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开perm space管理。
- off：表示关闭perm space管理。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

parctl_min_cost

参数说明：设置语句受到资源池并发控制的最小执行代价。

参数类型：整型

参数单位：无

取值范围：-1 ~ 2147483647

- 值为-1时或者执行语句的代价小于10时，不受资源池并发控制。
- 值大于等于0时，当**enable_dynamic_workload**为off时，如果执行语句的代价大于或等于10并且超过这个参数值就会受到资源池并发控制。

默认值：100000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cpu_collect_timer

参数说明：设置语句执行时在DN上收集CPU时间的周期。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：30

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：数据库管理员需根据系统资源（如CPU资源、I/O资源和内存资源）情况，调整此数值大小，使得系统支持较合适的收集周期，太小会影响执行效率，太大会影响异常处理的精确度。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

memory_tracking_mode

参数说明：设置记录内存信息的模式。

参数类型：枚举类型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- none：不启动内存统计功能。
- peak：统计query级内存peak值；此数值记入数据库log，也可由explain analyze输出。
- normal：仅做内存实时统计，不生成文件。
- executor：生成统计文件，包含执行层使用过的所有已分配内存的上下文信息。
- fullexec：生成统计文件，包含执行层申请过的所有内存上下文信息。

默认值：none

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

memory_detail_tracking

参数说明：设置需要的线程内分配内存上下文的顺序号以及当前线程所在query的plannodeid，仅用在DEBUG版本。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数不允许用户进行设置，建议保持默认值。

设置不当的风险与影响：无

enable_resource_track

参数说明：设置是否开启资源实时监控功能。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开资源监控功能。
- off：表示关闭资源监控功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_resource_record

参数说明：设置是否开启资源监控记录归档功能。开启时，对于history视图（GS_WLM_SESSION_HISTORY和GS_WLM_OPERATOR_HISTORY）中的记录，每隔

版权所有 © 华为云计算技术有限公司

3分钟会分别被归档到相应的info视图（GS_WLM_SESSION_INFO和GS_WLM_OPERATOR_INFO），归档后history视图中的记录会被清除。此参数需在CN和DN同时应用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启资源监控记录归档功能。
- off：表示关闭资源监控记录归档功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_logical_io_statistics

参数说明：设置是否开启资源监控逻辑I/O统计功能。开启时，对于PG_TOTAL_USER_RESOURCE_INFO视图中的read_kbytes、write_kbytes、read_counts、write_counts、read_speed和write_speed字段，会统计对应用户的逻辑读写字节数、次数以及速率；对于GS_WLM_USER_RESOURCE_HISTORY与GS_WLM_INSTANCE_HISTORY系统表中的逻辑读写相关字段，会统计相关用户、实例的逻辑读写对应值。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启资源监控逻辑I/O统计功能。
- off：表示关闭资源监控逻辑I/O统计功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_instance_metric_persistent

参数说明：设置是否开启实例资源监控转存功能。开启时，对实例的监控数据会保存到GS_WLM_INSTANCE_HISTORY系统表中。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启实例资源监控转存功能。

版权所有 © 华为云计算技术有限公司

- off：表示关闭实例资源监控转存功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

instance_metric_retention_time

参数说明：设置实例历史资源监控数据的保存天数。该参数仅在**enable_instance_metric_persistent**参数取值为on时有效。

参数类型：整型

参数单位：d（天）

取值范围：0 ~ 3650

- 值等于0时，实例历史资源监控数据将永久保存。
- 值大于0时，实例历史资源监控数据将保存对应设置天数。

默认值：7

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值如果要带单位，必须为d。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

resource_track_level

参数说明：设置当前会话的资源监控的等级。该参数仅在**enable_resource_track**参数取值为on时才有效。

参数类型：枚举类型

参数单位：无

取值范围：

- none：不开启资源监控功能。
- query：开启query级别资源监控功能。
- operator：开启query级别和算子级别资源监控功能。

默认值：query

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

resource_track_cost

参数说明：设置对当前会话的语句进行资源监控的最小执行代价。该参数仅在**enable_resource_track**参数取值为on时才有效。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：无

取值范围：-1 ~ 2147483647

- 值为-1时，不进行资源监控。
- 值大于等于0且小于等于9时，对执行代价大于等于10的语句进行资源监控。
- 值大于等于10时，对执行代价超过该参数值的语句进行资源监控。

默认值：100000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

resource_track_duration

参数说明：设置资源监控实时视图中记录的语句执行结束后进行历史信息转存的最小执行时间。当执行完成的作业，其执行时间不小于此参数值时，作业信息会从实时视图（以statistics为后缀的视图）转存到相应的历史视图（以history为后缀的视图）中。该参数仅在**enable_resource_track**参数取值为on时才有效。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

- 值为0时，资源监控实时视图中记录的所有语句都进行历史信息归档。
- 值大于0时，资源监控实时视图中记录的语句的执行时间超过这个值就会进行历史信息归档。

默认值：60

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值3600，表示resource_track_duration为3600s；带单位取值1h，表示resource_track_duration为1h。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dynamic_memory_quota

参数说明：自适应负载场景下，设置内存控制的比重，即可以使用系统最大可用内存的比例。

参数类型：整型

参数单位：无

取值范围：1 ~ 100

默认值：80

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

disable_memory_protect

参数说明：设置是否禁止内存保护功能。当系统内存不足时如果需要查询系统视图，可以先将此参数置为on，禁止内存保护功能，保证视图可以正常查询。该参数只适用于在系统内存不足时进行系统诊断和调试，正常运行时请保持该参数配置为off。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示禁止内存保护功能。
- off：表示启动内存保护功能。

默认值：off

设置方式：该参数属于USERSET类型参数，且只对当前会话有效，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

query_band

参数说明：用于标识当前会话的作业类型，由用户自定义。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

memory_fault_percent

参数说明：设置内存故障测试时内存申请失败的比例，仅用在DEBUG版本。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_bbox_dump

参数说明：设置是否开启黑匣子功能，在系统不配置core机制的情况下，是否产生core文件。此功能只对CN或DN有效，CMA、CMS、GTM、fenced UDF等仍需配置系统core机制才能捕获core文件。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开黑匣子功能。
- off：表示关闭黑匣子功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

黑匣子功能生成core文件依赖操作系统开放ptrace接口；若发生权限不足(errno = 1)，请确保/proc/sys/kernel/yama/ptrace_scope配置合理。

bbox_dump_count

参数说明：在**bbox_dump_path**定义的路径下，允许存储的GaussDB所产生core文件最大数。超过此数量时，旧的core文件会被删除。此参数仅在**enable_bbox_dump**参数取值为on时才生效。

说明

在并发产生core文件时，core文件的产生个数可能大于bbox_dump_count。

参数类型：整型

参数单位：无

取值范围：1 ~ 20

默认值：8

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bbox_dump_path

参数说明：黑匣子core文件的生成路径。此参数仅在**enable_bbox_dump**参数取值为on时才生效。

参数类型：字符串

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：合法文件路径。

默认值：""。默认生成黑匣子core文件的路径为读取"/proc/sys/kernel/core_pattern"下的路径，如果这个路径不是一个目录，或者用户对此目录没有写权限，黑匣子core文件将生成在数据库的data目录下。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bbox_blanklist_items

参数说明：黑匣子core文件的脱敏数据选项。此参数仅在**enable_bbox_dump**参数取值为on时才生效。

参数类型：字符串

参数单位：无

取值范围：以逗号分隔的敏感数据选项组成的字符串。

目前支持脱敏的数据项：

- SHARED_BUFFER：buffer数据缓冲区。
- XLOG_BUFFER：redo日志缓冲区。
- DW_BUFFER：双写数据缓冲区。
- XLOG_MESSAGE_SEND：主备复制日志发送缓冲区。
- WALRECEIVER_CTL_BLOCK：主备复制日志接收缓冲区。
- DATA_MESSAGE_SEND：主备复制数据发送缓冲区。
- DATA_WRITER_QUEUE：主备复制数据接收缓冲区。
- BUCKET_XLOG_MESSAGE_SEND：bucket扩容回放日志发送缓冲区。

默认值：""。表示bbox生成的core文件脱敏所有支持的敏感数据项。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_ffic_log

参数说明：设置是否开启FFIC（First Failure Info Capture）功能。此功能只对CN或DN有效，CMA、CMS、GTM、fenced UDF等仍需配置系统core机制才能捕获core文件。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示打开FFIC功能。
- off：表示关闭FFIC功能。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_dynamic_workload

参数说明：设置是否开启内存自适应功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开内存自适应功能。
- off：表示关闭内存自适应功能。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 开启内存自适应后，不再需要使用work_mem进行算子内存使用调优，由系统根据当前负载情况，为每个语句生成计划，并估算每个算子的内存使用量和整个语句的内存使用量。系统根据负载情况和整个语句内存使用量进行队列调度，所以多并发场景会出现语句排队的情况。
- 由于优化器行数估算不准现象的存在，会出现语句内存使用量低估或高估的情况。低估时，执行时内存会自动扩展。高估时，会导致系统内存利用不足，排队语句增多，可能导致性能非最优。此时需要识别语句估算内存远大于实际DN峰值内存的语句，通过设置query_mem进行调优，详情请参见《SQL调优指南》中“SQL调优关键参数调整”章节。

enable_reaper_backend

参数说明：设置是否启用单独线程来回收子线程退出时的信号。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启此功能。
- off：表示关闭此功能。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

bypass_workload_manager

参数说明：设置是否打开I/O管控独立的开关，此参数需在CN和DN同时应用。

在不开启**use_workload_manager**的情况下，通过此参数单独控制I/O管控功能的开启。打开之后可通过设置**io_limits**或**io_priority**进行I/O管控。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开I/O管控的独立开关。
- off：表示关闭I/O管控的独立开关。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

io_limits

参数说明：设置每秒触发I/O的上限。

参数类型：整型

参数单位：无

取值范围：0 ~ 1073741823

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

io_priority

参数说明：I/O利用率高达50%时，重消耗I/O作业进行I/O资源管控时关联的优先级等级。

参数类型：枚举类型

参数单位：无

取值范围：

- None：表示不受控。

版权所有 © 华为云计算技术有限公司

- Low：表示限制iops为该作业原始触发数值的10%。
- Medium：表示限制iops为该作业原始触发数值的20%。
- High：表示限制iops为该作业原始触发数值的50%。

默认值：None

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

io_control_unit

参数说明：行存场景下，I/O管控时用来对I/O次数进行计数的单位，此参数需在CN和DN同时应用。

记多少次I/O触发为一计数单位，通过此计数单位所记录的次数进行I/O管控。

参数类型：整型

参数单位：无

取值范围：1000 ~ 1000000

默认值：6000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

session_respool

参数说明：设置当前的session关联的resource pool。

如果先设置**cgroup_name**，再设置session_respool，那么session_respool关联的控制组起作用，如果再修改cgroup_name，那么新修改的cgroup_name起作用。

修改cgroup_name的过程中如果指定Workload控制组级别，数据库不对级别进行验证。级别的范围只要在1-10范围内都可以。

参数类型：字符串

参数单位：无

取值范围：通过create resource pool所设置的资源池的名称。

默认值："invalid_pool"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：尽量不要混合使用**cgroup_name**和session_respool。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

session_statistics_memory

参数说明：设置实时查询视图的内存大小。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：kB

取值范围：5120 ~ 2147483647

默认值：5120（即5MB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值16384，表示session_statistics_memory为16384kB；带单位取值16MB，表示session_statistics_memory为16MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：建议取值不大于**max_process_memory**的50%。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

session_history_memory

参数说明：设置历史查询视图的内存大小。

参数类型：整型

参数单位：kB

取值范围：10240 ~ 2147483647

默认值：10240（即10MB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值16384，表示session_history_memory为16384kB；带单位取值16MB，表示session_history_memory为16MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：建议取值不大于**max_process_memory**的50%。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_transaction_parctl

参数说明：设置是否管控事务块语句和存储过程语句。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示对事务块及存储过程语句进行管控。
- off：表示不对事务块及存储过程语句进行管控。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

topsql_retention_time

参数说明：设置历史TopSQL中gs_wlm_session_query_info_all和gs_wlm_operator_info表中数据的保存时间。

参数类型：整型

参数单位：d（天）

取值范围：0 ~ 3650

- 值为0时，表示数据永久保存。
- 值大于0时，表示数据能够保存的对应天数。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

transaction_pending_time

参数说明：设置事务块语句和存储过程语句排队的最大时间。此参数仅在**enable_transaction_parctl**参数取值为on时才生效。

参数类型：整型

参数单位：s（秒）

取值范围：-1 ~ 1073741823

- 值为-1或0时，事务块语句和存储过程语句无超时判断，排队至资源满足可执行条件。
- 值大于0时，事务块语句和存储过程语句排队超过所设数值的时间后，无视当前资源情况强制执行。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

node_group_mode

参数说明：显示当前的node group模式。

参数类型：字符串

参数单位：无

取值范围：node group模式名称。

默认值："node group"

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

resilience_ctrlslot_available_maxpercent

参数说明：最多允许慢SQL占用线程池线程数的百分比。仅对非sysadmin/monadmin用户执行的select类型的语句生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 100

默认值：100，表示允许慢SQL使用所有的线程池线程。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

resilience_ctrlstmt_detect_timelimit

参数说明：正常SQL语句被标记为慢SQL所需的执行时间，0表示不做慢SQL识别，大于0表示当SQL执行时间超过该时间时被标记为慢SQL。仅对非sysadmin/monadmin用户执行的select类型的语句生效。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647，0表示关闭慢SQL超时检测功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值6000，表示resilience_ctrlstmt_detect_timelimit为6000ms；带单位取值6s，表示resilience_ctrlstmt_detect_timelimit为6s。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：根据实时数据库执行sql的平均时长来设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

resilience_ctrlstmt_control_iopslimit

参数说明：正常SQL语句被标记为慢SQL后，慢SQL可使用的最大IOPS上限。仅对非sysadmin/monadmin用户执行的select类型的语句生效。

参数类型：字符串

参数单位：无

取值范围："Low"、"Medium"、"High"、"None"、0 ~ 2147483647。级别越低管控越严格。

版权所有 © 华为云计算技术有限公司

- "Low"：低级。
- "Medium"：中级。
- "High"：高级。
- "None"：关闭慢SQL逻辑I/O管控功能。
- 0 ~ 2147483647：数值越小越严格，越大越宽松。

默认值："None"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。如果设置为数值，需要使用字符串的形式，如resilience_ctrlstmt_control_iopslimit = '1024'。

设置建议：I/O访问满时设置为较小值，I/O空闲时可设置为较大值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

session_max_dynamic_memory

参数说明：管控会话级内存上限，会话内存使用超过该值后会报错中断。此参数仅在**enable_memory_limit**参数取值为on时才生效。

参数类型：整型

参数单位：kB

取值范围：-1 ~ 2147483647

默认值：-1，表示不管控。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：设为默认值，不做调整。

设置不当的风险与影响：该参数不能设置太小，设置太小会导致该会话执行的语句因内存不足失败，若是通过**表6-2**中的方式三设置值，且因为内存不足导致无法调大该参数时，只能通过退出会话来消除该限制。当设置的值小于16MB时会给出对应的提示信息。

max_concurrency

参数说明：管控全局并发数上限。

参数类型：整型

参数单位：无

取值范围：-1 ~ 2147483647

默认值：-1，表示不管控。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：数据库管理员需根据系统资源（如CPU资源、I/O资源和内存资源）的使用情况，调整此参数大小，使得系统支持最大限度的并发作业操作，并且防止由于并发执行作业过多导致系统崩溃的情况。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_workload_rule

参数说明：设置是否启用SQL限流功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用SQL限流功能。
- off：表示不启用SQL限流功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果不需要SQL限流功能，可以设置成off，关闭该功能。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_sysadmin_workload_rule

参数说明：设置是否启用管理员SQL限流功能。该参数对superuser无效，superuser始终不受限流规则影响。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用管理员SQL限流功能。
- off：表示不启用管理员SQL限流功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议保持默认值off。仅在需要对管理员用户执行的SQL进行限流时，设置为on。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

workload_uncontrolled_users

参数说明：负载管理的用户白名单列表，位于列表中的用户，不对其进行负载管理，目前仅对SQL限流生效。

参数类型：字符串

参数单位：无

取值范围：用户列表，以“,”分隔，例如“user1, user2, user3”。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要设置，对于一些不具备sysadmin权限的用户，但不希望其执行SQL语句时被负载管理限制时，可以添加到该参数中。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_emergency_access

参数说明：控制是否开启Unix Domain Socket运维通道，后称“应急通道”。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启应急通道功能。
- off：表示不开启应急通道功能。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：建议使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

max_plancache_memory_ratio

参数说明：用于控制session级prepare statement和cursor_sharing的计划缓存占用动态内存的百分比阈值。

参数类型：整型

参数单位：无

取值范围：0 ~ 100

该参数是一个取值范围在0 ~ 100的整型数，其含义为当前session中计划缓存的上限占“预期为每个session预留内存”的百分比阈值。设置为0表示禁用该功能。

系统动态内存预期为每个session预留的内存大小遵从如下**表6-27**规则：

表 6-27 系统动态内存为每个 session 预留的内存大小对应表

CurConnCount/ max_connections	预期为每个session预留内存
0 ~ 30%	max_dynamic_memory / (max_connections * 30%)
30% ~ 60%	max_dynamic_memory / (max_connections * 60%)
60% ~ 100%	max_dynamic_memory / (max_connections * 100%)

其中，CurConnCount表示当前session个数。max_connections表示最大并发session数，由GUC参数max_connections控制。max_dynamic_memory表示系统最大的动态内存。

版权所有 © 华为云计算技术有限公司

当系统监测到计划缓存占用的内存大小与Session预留内存的比值达到或超过参数max_plancache_memory_ratio设定的百分比阈值时，将触发缓存淘汰机制。该机制通过LRU（最近最少使用）算法释放计划缓存条目，直至当前计划缓存内存占比回落到阈值以下。被淘汰的缓存条目在后续重新执行时，系统将自动生成新的执行计划以确保查询正常执行

默认值：

- 独立部署：

  CN：60（60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存）；30（8核CPU/64G内存）；50（4核CPU/32G内存）；40（4核CPU/16G内存）

  DN：60（60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存）；50（8核CPU/64G内存，4核CPU/32G内存）；30（4核CPU/16G内存）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果该参数设置的百分比过小，则会频繁触发计划缓存淘汰、重建，有导致性能劣化的风险，需要根据实际内存使用情况慎重设置。对于混合部署场景需要基于单个进程分配的内存考虑，调整参数配置。该参数仅控制session级prepare statement和cursor_sharing的计划缓存占用的动态内存，其他内存不在管控范围内。

当前参数配置基于内存配置、连接数以及性能考虑给出。当计划内存占用过多，可能导致max_dynamic_memory占满，或触发内存淘汰导致性能劣化。在这类特殊场景下，建议根据业务调整内存比例大小。

另外，在GUC参数enable_opfusion取值为on的场景下，session级别的opfusion相关内存暂未纳入该参数控制，存在这部分内存不受控制的风险，如果出现此类问题，建议降低GUC参数max_plancache_memory_ratio的取值以缓解此类问题。

设置不当的风险与影响：如果该参数设置的百分比过小，则会频繁触发计划缓存淘汰、重建，有导致性能劣化的风险；如果该参数设置的百分比过大，可能会导致内存资源紧张，仍有资源不可用的风险。请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 该参数设置后，需要将参数**enable_memory_limit**打开才能生效。
- 最大动态内存可以通过pv_total_memory_detail视图查询获得，对应字段为：max_dynamic_memory。最大连接数可以通过GUC参数max_connections查看。
- 缓存淘汰只在执行计划创建、重建后内存大小达到或超过上限后触发。当新session连接后，占比从**表6-27**中给出的区间之间跳变时（例如从29%增加到31%），为每个session预留内存随之改变，注意此时并不会被动触发淘汰。
- 版本升级后，该参数默认关闭，即值为0。

global_plancache_memory_ratio

参数说明：用于控制全局共享计划缓存占用动态内存的百分比阈值，0表示关闭GPC内存管控。

参数类型：整型

参数单位：无

取值范围：0 ~ 80

默认值：

版权所有 © 华为云计算技术有限公司

- 独立部署：

  CN：30（60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存）；20（8核CPU/64G内存）；30（4核CPU/32G内存）；20（4核CPU/16G内存）

  DN：30（60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存，8核CPU/64G内存，4核CPU/32G内存）；30（4核CPU/16G内存）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果该参数设置的百分比过小，可能会导致计划缓存无法共享，需要根据实际内存使用情况慎重设置。

对于不同内存规格环境建议调整参数设置，可参考出口参数设置。

混合部署场景需要基于单个进程分配的内存考虑，调整参数配置。

当前参数配置基于内存配置以及性能考虑给出。当计划内存占用过多，可能导致max_dynamic_memory占满，或触发内存淘汰导致性能劣化。在这类特殊场景下，建议根据业务调整内存比例大小。

设置不当的风险与影响：如果该参数设置的百分比过小，可能会导致大量计划缓存无法共享；如果该参数设置的百分比过大，可能会导致内存资源紧张，仍有资源不可用的风险。请在充分理解参数含义，并经过测试验证后进行修改。

说明

该参数设置后，需要将参数**enable_memory_limit**打开才能生效。

current_logic_cluster

参数说明：显示当前的逻辑数据库实例名称。

参数类型：字符串

参数单位：无

取值范围：逻辑数据库实例名称。

默认值：""

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无。

6.3.13 自动清理

系统自动清理线程（autovacuum）包含VACUUM和ANALYZE两个过程，回收被标识为删除状态的记录空间，并更新表的统计数据。两次autovacuum的执行间隔时间由参数autovacuum_naptime控制，默认时间为10分钟，可根据实际业务场景进行配置。该间隔时间不能保证完全精确，一方面取决于环境的硬件条件和负载，当负载较高时，可能由于需要清理的数据量较大，造成延迟；另一方面取决于相关参数autovacuum_naptime和autovacuum_max_workers的配置值，autovacuum_naptime为执行间隔时间，配置值越小，执行间隔越小，但由于涉及数据的清理和统计信息的计算，占用的CPU、内存和I/O开销均会增大；autovacuum_max_workers为同时运行的自动清理线程的最大数量，配置值越大，执

版权所有 © 华为云计算技术有限公司

行间隔越符合autovacuum_naptime的配置，但占用的CPU、内存和I/O开销同样会增大。

autovacuum

参数说明：控制数据库自动清理线程（autovacuum）的启动。自动清理线程运行的前提是将**track_counts**设置为on。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启数据库自动清理线程。
- off：表示关闭数据库自动清理线程。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 如果希望系统在故障恢复后，具备自动清理两阶段事务的功能，请将autovacuum设置为on。
- 当设置autovacuum为on，**autovacuum_max_workers**为0时，表示系统不会自动进行autovacuum，只会在故障恢复后自动清理两阶段事务。
- 当设置autovacuum为on，**autovacuum_max_workers**大于0时，表示系统不仅会在故障恢复后自动清理两阶段事务，还可以自动进行autovacuum。

设置不当的风险与影响：在ASTORE表中，关闭该参数会导致表中有过多的不活跃行存在，可能会影响查询性能。

autovacuum_mode

参数说明：该参数仅在**autovacuum**设置为on的场景下生效，它控制autoanalyze和autovacuum的打开情况。

参数类型：枚举类型

参数单位：无

取值范围：

- analyze：表示只做autoanalyze。
- vacuum：表示只做autovacuum。
- mix：表示autoanalyze和autovacuum都做。
- none：表示autoanalyze和autovacuum都不做。

默认值：mix

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

autoanalyze_timeout

参数说明：设置autoanalyze的超时时间。在对某张表做autoanalyze时，如果该表的analyze时长超过了autoanalyze_timeout，则自动取消该表此次analyze。

这里的时间检查不能保证完全精确，原则上要保证各个CN上统计信息一致，因此在CN间同步信息时，即便超时也不会被打断。这导致实际的执行时间有可能超过用户定义的时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483，0表示没有超时限制。

默认值：300

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：设置值取决于数据库的负载和性能需求。如果数据库的负载较轻，可以适当地将autoanalyze_timeout设置大些，以便自动分析操作有足够的时间来完成。如果数据库的负载较重，则应该将autoanalyze_timeout设置小些，以便自动分析操作可以在短时间内完成，从而避免影响其他重要的数据库操作。

设置不当的风险与影响：autoanalyze操作会消耗一定的系统资源，如果该参数设置太大，可能会导致系统资源被过度消耗，从而影响数据库的性能。如果设置太小，autoanalyze操作可能无法完成，可能会导致统计信息不准确，从而影响查询的执行计划和性能。

autovacuum_io_limits

参数说明：控制autovacuum线程每秒触发I/O的上限。

参数类型：整型

参数单位：无

取值范围：-1 ~ 1073741823。其中-1表示不控制，而是使用系统默认控制组。

默认值：-1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 服务器有足够的磁盘I/O资源，可以将autovacuum_io_limits设置为较高的值，这将允许autovacuum线程更快地完成清理操作。
- 服务器磁盘I/O资源有限，可以将autovacuum_io_limits设置为较低的值。这将限制autovacuum线程使用的磁盘I/O资源，以避免对其他线程的影响。

设置不当的风险与影响：

- 设置为过高的值，可能会导致autovacuum线程使用过多的磁盘I/O资源，从而影响其他线程的性能。
- 设置为过低的值，可能会导致autovacuum线程无法有效地清理表数据，从而导致表数据过度膨胀，影响数据库性能。

版权所有 © 华为云计算技术有限公司

log_autovacuum_min_duration

参数说明：当自动清理的执行时间大于或者等于某个特定的值时，向服务器日志中记录自动清理执行的每一步操作。设置此选项有助于追踪自动清理的行为。

例如：将log_autovacuum_min_duration设置为250ms，表示记录所有运行大于或者等于250ms的自动清理命令的相关信息。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 2147483647

- 当参数设置为0时，表示所有的自动清理操作都记录到日志中。
- 当参数设置为-1时，表示所有的自动清理操作都不记录到日志中。
- 当参数设置大于等于0时，当由于锁冲突的存在导致一个自动清理操作被跳过，记录一条消息（记录跳过autovacuum原因，便于审计）。

默认值：-1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值600，表示log_autovacuum_min_duration为600ms；带单位取值10min，表示log_autovacuum_min_duration为10min。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：一般默认为-1。如果需要监控自动VACUUM操作的性能，可以将log_autovacuum_min_duration设置为一个大于0的值，以记录持续时间超过该值的自动VACUUM操作的日志信息。

设置不当的风险与影响：如果设置为0或者过小，则可能会记录大量的日志信息，影响系统性能；如果设置过大，可能无法及时感知自动清理时间过久，导致清理速度不足或异常情况。

autovacuum_max_workers

参数说明：设置能同时运行的自动清理线程的最大数量。

参数类型：整型

参数单位：无

取值范围：最小值为0（表示不会自动进行AUTOVACUUM），理论最大值为1024，用户设置超过1024时会内部自动调整为1024，但设置的参数值仍可能超过实际取值上限；该参数的实际取值上限与GUC参数**max_connections**和**job_queue_processes**大小有关，计算公式为“262143 - max_inner_tool_connections - max_connections - max_concurrent_autonomous_transactions - job_queue_processes - 辅助线程数 – autovacuum的launcher线程数[x(1] - 1”，其中辅助线程数和autovacuum的launcher线程数由两个宏来指定，当前版本的默认值分别为20和2。

默认值：3

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置越大，自动清理功能创建的线程数越多，占用的系统CPU和内存资源越多，所以设置时不建议设置过大，避免由于此参数设置过大导

版权所有 © 华为云计算技术有限公司

致内存无法分配或者占用过多CPU资源，导致数据库启动报错或业务受到影响。如果将该参数设置过小，可能会导致autovacuum线程无法及时清理和回收不再使用的表空间，从而导致数据库膨胀和性能下降。参数实际设置值需要根据取值范围的实际上限评估，一般可达到1024；如果设置值超过实际上限则会设置失败。此参数设置后对新线程生效，而对当前运行中的线程无效。

autovacuum_naptime

参数说明：设置两次自动清理操作的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483

默认值：600（即10min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值600，表示autovacuum_naptime为600s；带单位取值10min，表示autovacuum_naptime为10min。取值如果要带单位，必须为s、min、h、d。

设置建议：仅用Ustore表时推荐使用默认值。对于Astore表，设置值取决于用户的数据库环境和应用程序负载。如果数据库环境中有大量表的更新或删除操作，建议将该参数设置为较短的时间间隔，以确保及时清理无用的数据并避免数据库空间不足的情况。如果数据库环境中的更新或删除操作较少，则可以将该参数设置为较长的时间间隔，以减少自动清理线程的开销。

设置不当的风险与影响：

- 如果将autovacuum_naptime设置为过短的时间间隔，可能会导致autovacuum线程的开销过大，从而影响数据库的性能。
- 如果将autovacuum_naptime设置为过长的时间间隔，可能会导致无用的数据堆积，从而占用数据库空间，影响数据库的性能。

autovacuum_vacuum_threshold

参数说明：触发ASTORE表和USTORE表VACUUM操作的基准值。触发VACUUM的阈值=autovacuum_vacuum_threshold+reltuples（表上元组的个数）* **autovacuum_vacuum_scale_factor**。当表上被删除和更新的记录数总和超过阈值时，才会对这个表执行VACUUM操作。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：50

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：

- 该参数设置得过低，会导致频繁的VACUUM操作，增加系统开销，降低性能。
- 该参数设置得过高，会导致表中过多的死亡行存在，可能会影响查询性能。

版权所有 © 华为云计算技术有限公司

autovacuum_analyze_threshold

参数说明：触发ASTORE表和USTORE表ANALYZE操作的基准值。触发ANALYZE的阈值=autovacuum_analyze_threshold+reltuples（表上元组的个数）*autovacuum_analyze_scale_factor。当表上被删除、插入和更新的记录数总和超过阈值时才会对这个表执行ANALYZE操作。若表上存在全局二级索引，则当表上被删除、插入和更新的记录数总和超过阈值的20倍时，才会对全局二级索引执行ANALYZE操作。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：50

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：

- 该参数设置得过低，会导致频繁的ANALYZE操作，增加系统开销，降低性能。
- 该参数设置得过高，表中的统计信息可能会过时，从而导致查询计划不准确，影响查询性能。

autovacuum_vacuum_scale_factor

参数说明：触发ASTORE表和USTORE表VACUUM操作的规模因子。触发VACUUM的阈值=autovacuum_vacuum_threshold+reltuples（表上元组的个数）*autovacuum_vacuum_scale_factor。当表上被删除和更新的记录数总和超过阈值时，才会对这个表执行VACUUM操作。

参数类型：浮点型

参数单位：无

取值范围：0.0 ~ 100.0

默认值：0.2

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：设置值取决于表的负载和使用情况；如果是高写入负载的，则用户需要将该参数设置得更低，以便更频繁地清理表以避免性能下降。另一方面，如果是高读取负载的，则可以将该参数设置得更高，以避免在读取时不必要地浪费资源。

设置不当的风险与影响：

- 该参数设置得太低，那么autovacuum线程可能会过于频繁地清理表，导致性能下降。
- 该参数设置得太高，那么表可能无法及时清理，会变得非常大，占用过多的磁盘空间。

autovacuum_analyze_scale_factor

参数说明：触发ASTORE表和USTORE表ANALYZE操作的规模因子。触发ANALYZE的阈值=autovacuum_analyze_threshold+reltuples（表上元组的个数）*

版权所有 © 华为云计算技术有限公司

autovacuum_analyze_scale_factor。当表上被删除、插入和更新的记录数总和超过阈值时，才会对这个表执行ANALYZE操作。

参数类型：浮点型

参数单位：无

取值范围：0.0 ~ 100.0

默认值：0.1，即表上元组数的10%。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：

- 该参数设置得过低，会导致频繁的ANALYZE操作，增加系统开销，降低性能。
- 该参数设置得过高，可能会导致ANALYZE的频率过低，表中的统计信息可能会过时，从而导致查询计划不准确，影响查询性能。

autovacuum_analyze_window

参数说明：在周一至周日范围内指定一个或多个时间窗口，在给定时间窗口中以**autovacuum_analyze_window_scale_factor**作为触发ASTORE表和USTORE表ANALYZE操作的规模因子。

参数类型：字符串

参数单位：无

取值范围：空字符串或由一个或多个指定时间窗口组成的字符串，由逗号分割。

说明

- 指定时间窗口的字符串为键值对格式，即key=value。
- 时间窗口的键必须为MONDAY_WINDOW、TUESDAY_WINDOW、WEDNESDAY_WINDOW、THURSDAY_WINDOW、FRIDAY_WINDOW、SATURDAY_WINDOW或SUNDAY_WINDOW，大小写不敏感。
- 时间窗口的值的格式为HH:MM-HH:MM，按“-”分隔。左边为窗口开始时间，右边为窗口结束时间。该格式中不能包含空格，最小时间为00:00，最大时间为23:59。
- 时间窗口的结束时间如果小于开始时间表明时间窗口出现跨天，如"MONDAY_WINDOW=22:00-02:00"，表明时间窗口从周一的晚上22点到周二的凌晨2点。
- 多天的时间窗口不能出现重叠。如无法设置"MONDAY_WINDOW=22:00-05:00,TUESDAY_WINDOW=03:00-09:00"，可以设置"MONDAY_WINDOW=22:00-02:59,TUESDAY_WINDOW=03:00-09:00"。
- 该参数不受**autovacuum_mode**参数控制，只要该参数设置成功，在给定时间窗口内会一直收集统计信息。
- 该参数同一天只能配置一个窗口，如果同一天配置多个窗口，只有最后配置的窗口生效。
- 时间窗口按照系统本地时间进行判断，改变系统时间时区时，请确认是否需要修改本参数配置的时间窗口。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，根据业务场景适当调整。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：单个时间窗口设置的时间间隔需大于**autovacuum_naptime**取值，否则时间窗口可能无法生效。

autovacuum_analyze_window_scale_factor

参数说明：时间窗口内触发ASTORE表和USTORE表ANALYZE操作的规模因子。触发ANALYZE的阈值 = **autovacuum_analyze_threshold** + reltuples（表上元组的个数）* autovacuum_analyze_window_scale_factor。当表上被删除、插入和更新的记录数总和超过阈值时，才会对这个表执行ANALYZE操作。

参数类型：浮点型

参数单位：无

取值范围：0.0 ~ 100.0

默认值：0.1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：

- 该参数设置得过低，会导致时间窗口内频繁的ANALYZE操作，增加系统开销，降低性能。
- 该参数设置得过高，可能会导致时间窗口内ANALYZE的频率过低，表中的统计信息可能会过时，从而导致查询计划不准确，影响查询性能。

autovacuum_freeze_max_age

参数说明：指定ASTORE表和USTORE表pg_class.relfrozenxid64字段在超过多少个事务后，会强制执行VACUUM操作。即使自动清理被禁用，系统也会启动AUTOVACUUM线程。强制VACUUM结束后会清理pg_clog和pg_csnlog子目录中可以被删除的旧文件，防止pg_clog和pg_csnlog目录无限膨胀。

参数类型：整型

参数单位：无

取值范围：100000 ~ 576460752303423487

默认值：4000000000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。如果数据库中的ASTORE表经常进行大量的更新和删除操作，那么建议将该参数设置为较小的值，以便及时回收垃圾。如果数据库中的ASTORE表很少进行更新和删除操作，那么可以将该参数设置为较大的值，以减少自动垃圾回收的频率。

设置不当的风险与影响：

- 该参数设置得过低，会导致频繁的自动垃圾回收操作，增加系统开销，降低性能。
- 该参数设置得过高，会导致自动垃圾回收的延迟，从而可能导致大量的无效数据占用磁盘空间，影响数据库性能。

版权所有 © 华为云计算技术有限公司

autovacuum_vacuum_cost_delay

参数说明：设置在自动VACUUM操作里使用的开销延迟数值。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 100。其中-1表示使用常规的**vacuum_cost_delay**。

默认值：20

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

autovacuum_vacuum_cost_limit

参数说明：设置在自动VACUUM操作里使用的开销限制数值。

参数类型：整型

参数单位：无

取值范围：-1 ~ 10000。其中-1表示使用常规的**vacuum_cost_limit**。

默认值：-1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

twophase_clean_workers

参数说明：该参数用来控制内核调度gs_clean工具的并发清理数。

参数类型：整型

参数单位：无

取值范围：1 ~ 10

默认值：3

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

clog_keep_xids

参数说明：设置clog事务文件保留的事务数。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 576460752303423487

- 值为0时： 按照原有回收逻辑进行回收，不额外保留事务文件。
- 值为非0且在取值范围内时： 按照新的回收逻辑进行回收（配置保留事务数的计算基于当前最新的事务ID，且仅当新配置的参数范围包含原本回收逻辑需要回收的事务文件时，才会触发保留机制）

默认值：0

说明：谨慎评估并确保环境预留足够的磁盘空间。每节点磁盘空间估算方法如下：

（配置保留事务数的计算基于当前最新的事务ID，且仅当新配置的参数范围包含原本回收逻辑需要回收的事务文件时，才会触发保留机制）

clog保留的空间占用（字节）：(clog_keep_xids / 4)

clog保留的空间占用(hashbucket)（字节）：db数 * buckets数 * (clog_keep_xids / 4)

例：clog_keep_xids配置4294967295的情况，普通场景共占用4294967295 / 4 = 1073741823字节，约1GB空间。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值（根据上面公式计算磁盘空间占用，在保证磁盘空间充足的前提下，如果不存在hashbucket业务建议设置为8589934590，如存在hashbucket业务建议设置为8388607）。

设置不当的风险与影响：设置不合理可能导致磁盘空间占用偏大，所有节点都涉及，请谨慎评估。

csnlog_keep_xids

参数说明：设置csnlog事务文件保留的事务数。

参数类型：整型

参数单位：无

取值范围：0 ~ 576460752303423487

- 值为0时： 按照原有回收逻辑进行回收，不额外保留事务文件。
- 值为非0且在取值范围内时： 按照新的回收逻辑进行回收（配置保留事务数的计算基于当前最新的事务ID，且仅当新配置的参数范围包含原本回收逻辑需要回收的事务文件时，才会触发保留机制）

默认值：0

说明：谨慎评估并确保环境预留足够的磁盘空间。每节点磁盘空间估算方法如下：

（配置保留事务数的计算基于当前最新的事务ID，且仅当新配置的参数范围包含原本回收逻辑需要回收的事务文件时，才会触发保留机制）

csnlog_keep_xids空间估算：

csnlog保留的空间占用（字节）：(csnlog_keep_xids * 8)

例：csnlog_keep_xids配置4294967295的情况，共占用4294967295 * 8 = 34359738360字节，约34GB空间。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值（根据上面公式计算磁盘空间占用，在保证磁盘空间充足的前提下，建议设置为252645135）。

设置不当的风险与影响：设置不合理可能导致磁盘空间占用偏大，所有节点都涉及，请谨慎评估。

defer_csn_cleanup_time

参数说明：用来指定本地回收时间间隔。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：5000（即5s）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。一般来说，如果系统的负载较轻，可以将defer_csn_cleanup_time设置为较长的时间。如果系统的负载较重，建议将defer_csn_cleanup_time设置为较短的时间。这样可以确保系统在忙碌时也能保持良好的性能。

设置不当的风险与影响：defer_csn_cleanup_time参数用于控制系统何时清理已过期的CSN信息。它指定了系统在多长时间内保留已过期的CSN信息。如果设置得太短，系统可能会频繁地清理已过期的CSN信息，从而导致性能下降。如果设置得太长，系统可能会占用过多的内存，从而导致系统变慢或崩溃。

multixact_keep_mxids

参数说明：设置multixact事务文件保留的事务数。

参数类型：整型

参数单位：无

取值范围：0 ~ 5764607523034234 87

- 值为0时： 按照原有回收逻辑进行回收，不额外保留事务文件。
- 值为非0且在取值范围内时： 按照新的回收逻辑进行回收（配置保留事务数的计算基于当前最新的multixact ID，且仅当新配置的参数范围包含原本回收逻辑需要回收的事务文件时，才会触发保留机制）

默认值：0

说明：谨慎评估并确保环境预留足够的磁盘空间。每节点磁盘空间估算方法如下：

（配置保留事务数的计算基于当前最新的事务ID，且仅当新配置的参数范围包含原本回收逻辑需要回收的事务文件时，才会触发保留机制）

multixact_keep_mxids空间估算：

multixact保留的空间占用（字节）：(multixact_keep_mxids * 8 + multixact_keep_mxids * members * 8)

例：multixact_keep_mxids配置4294967295的情况，multixact平均3并发(3个members)场景下，共占用

版权所有 © 华为云计算技术有限公司

(4294967295 * 8 + 4294967295 * 3 * 8) = 137438953440字节，约137GB空间。

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值（根据上面公式计算磁盘空间占用，在保证磁盘空间充足的前提下，建议设置为63161283）。

设置不当的风险与影响：设置不合理可能导致磁盘空间占用偏大，所有节点都涉及，请谨慎评估。

6.3.14 客户端连接缺省设置

6.3.14.1 语句行为

介绍SQL语句执行过程的相关默认参数。

search_path

参数说明：当一个被引用对象没有指定模式时，此参数设置模式搜索顺序。

参数类型：字符串

参数单位：无

取值范围：由一个或多个模式名构成的字符串，不同的模式名用逗号隔开。

说明

- 设置为'"$user",public'时，支持共享数据库（没有用户具有私有模式和所有共享使用public），用户私有模式和这些功能的组合使用。可以通过改变默认搜索路径来获得其他效果，无论是全局化的还是私有化的。
- 设置为空字符串（''）的时候，系统会自动转换成一对双引号。
- 设置的内容中包含双引号时，系统会认为是不安全字符，会将每个双引号转换成一对双引号。

默认值：'"$user",public'

说明

$user表示与当前会话用户名同名的模式名，如果这样的模式不存在，$user将被忽略。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：

- 当前会话存放临时表的模式时，可以使用别名pg_temp将它列在搜索路径中，如'pg_temp,public'。存放临时表的模式始终会作为第一个被搜索的对象，排在pg_catalog和search_path中所有模式的前面，即具有第一搜索优先级。建议用户不要在search_path中显式设置pg_temp。如果在search_path中指定了pg_temp，但不是在最前面，系统会提示设置无效，pg_temp仍被优先搜索。通过使用别名pg_temp，系统只会在存放临时表的模式中搜索表、视图和数据类型这样的数据库对象，不会在里面搜索函数或运算符这样的数据库对象。
- 系统表所在的模式pg_catalog，总是排在search_path中指定的所有模式前面被搜索，即具有第二搜索优先级（pg_temp具有第一搜索优先级）。建议用户不要在search_path中显式设置pg_catalog。如果在search_path中指定了pg_catalog，但不是在最前面，系统会提示设置无效，pg_catalog仍被第二优先搜索。

版权所有 © 华为云计算技术有限公司

- 当没有指定一个特定模式而创建一个对象时，它们被放置到以search_path为命名的第一个有效模式中。当搜索路径为空时，会报错误。
- 可以通过SQL函数current_schema()检测当前搜索路径的有效值。这和search_path的值不完全相同，因为current_schema()显示search_path中首位有效的模式名称。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

current_schema

参数说明：设置当前的模式。

参数类型：字符串

参数单位：无

取值范围：由一个或多个模式名构成的字符串，不同的模式名用逗号隔开。

默认值：'"$user",public'

说明

$user表示与当前会话用户名同名的模式名，如果这样的模式不存在，$user将被忽略。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_tablespace

参数说明：当CREATE命令没有明确声明表空间时，所创建对象（表和索引等）的缺省表空间。

说明

- 值是一个表空间的名称或者一个表示使用当前数据库缺省表空间的空字符串。若指定的是一个非默认表空间，用户必须具有它的CREATE权限，否则尝试创建会失败。
- 临时表不使用此参数，可以用**temp_tablespaces**代替。
- 创建数据库时不使用此参数。默认情况下，一个新的数据库从模板数据库继承表空间配置。

参数类型：字符串

参数单位：无

取值范围：已有表空间的名称或空字符串，其中空字符串表示使用默认表空间。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

temp_tablespaces

参数说明：当一个CREATE命令没有明确指定一个表空间时，temp_tablespaces指定了创建临时对象（临时表和临时表的索引）所在的表空间。在这些表空间中创建临时文件用来做大型数据的排序工作。

其值是一系列表空间名的列表。如果列表中有多个表空间时，每次临时对象的创建，GaussDB会在列表中随机选择一个表空间；如果在事务中，连续创建的临时对象被放置在列表里连续的表空间中。如果选择的列表中的元素是一个空字符串，GaussDB将自动将当前的数据库设为默认的表空间。

参数类型：字符串

参数单位：无

取值范围：已有表空间的名称或空字符串。空字符串表示所有的临时对象仅在当前数据库默认的表空间中创建，请参见**default_tablespace**。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_storage_nodegroup

参数说明：设置当前默认建表所在的Node Group，目前只适用普通表。

参数类型：字符串

参数单位：无

取值范围：已有Node Group的名称或“installation”。

默认值："installation"

- 值为“installation”表示建表会默认建在安装的Node Group上。
- 值为其他字符串表示建表会默认建在设置的Node Group上。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

check_function_bodies

参数说明：设置是否在CREATE FUNCTION执行过程中进行函数体字符串的合法性验证。为了避免产生问题（比如避免从转储中恢复函数定义时向前引用的问题），偶尔会禁用验证。开启后主要验证存储过程中PL/SQL的语法问题，包括数据类型、语句和表达式等，对于其中出现的SQL则在CREATE阶段不做检查而采用了运行时检查的方式。

说明

O兼容模式下，设置check_function_bodies值为off的条件下，创建带有自定义类型和自治事务的存储过程或者函数，调用该存储过程或函数时，会触发死锁。设置参数check_function_bodies值为on，重建函数或存储过程后调用正常。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在CREATE FUNCTION执行过程中进行函数体字符串的合法性验证。
- off：表示在CREATE FUNCTION执行过程中不进行函数体字符串的合法性验证。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_transaction_isolation

参数说明：设置默认的事务隔离级别。

说明

当前版本暂不支持设置默认的事务隔离级别，默认为read committed，请勿自行修改。

参数类型：枚举类型

参数单位：无

取值范围：

- read committed：表示事务读已提交。
- read uncommitted：表示事务读未提交。
- repeatable read：表示事务可重复读。
- serializable：GaussDB目前功能上不支持此隔离级别，等价于repeatable read。

默认值：read committed

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_transaction_read_only

参数说明：设置每个新创建事务是否是只读状态。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示只读状态。
- off：表示非只读状态。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数设为on后，无法执行dml和写事务。

default_transaction_deferrable

参数说明：控制每个新事务的默认延迟状态。只读事务或者那些比序列化更加低的隔离级别的事务除外。

GaussDB不支持可串行化的隔离级别，因此，该参数无实际意义。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示默认延迟。
- off：表示默认不延迟。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

session_replication_role

参数说明：控制当前会话与复制相关的触发器和规则的行为。

说明

修改该参数取值会丢弃之前任何缓存的执行计划。

参数类型：枚举类型

参数单位：无

取值范围：

- origin：表示当前会话是数据源，会正常执行触发器、外键、约束和规则检查。该取值为所有普通会话的默认设置，适用于常规数据库操作。
- replica：表示当前会话是数据接收端（如从数据库），仅会开启ENABLE REPLICA模式的触发器、外键和规则，其他模式的触发器、规则和外键都被会禁用。
- local：表示函数执行复制时会检测当前登录数据库的角色并采取相应的操作。该取值主要用于特定的复制函数场景，使用频率较低，行为通常与origin一致。

默认值：origin

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

statement_timeout

参数说明：当语句执行时间超过该参数设置的时间（从服务器收到命令时开始计时）时，该语句将会报错并退出执行。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：0，表示不开启超时报错功能。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2000，表示statement_timeout为2000ms；带单位取值2s，表示statement_timeout为2s。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

vacuum_freeze_min_age

参数说明：VACUUM在扫描一个表时，用于判断是否用FrozenXID替换记录的xmin字段（在同一个事务中）的最小时间。

参数类型：整型

参数单位：无

取值范围：0 ~ 576460752303423487

说明

尽管随时可以将此参数设为上述取值范围之间的任意值，但是，VACUUM将默认其有效值范围限制在**autovacuum_freeze_max_age**的50%以内。

默认值：2000000000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置偏大会使得freeze时间推迟，导致回收效率降低。

vacuum_freeze_table_age

参数说明：指定VACUUM扫描全表以冻结元组的时间。如果当前事务号与表pg_class.relfrozenxid64字段的差值已经大于参数指定的时间，则VACUUM会对全表进行扫描。

参数类型：整型

参数单位：无

取值范围：0 ~ 576460752303423487

版权所有 © 华为云计算技术有限公司

说明

尽管随时可以将此参数设为上述取值范围之间的任意值，但是，VACUUM将默认其有效值范围限制在**autovacuum_freeze_max_age**的95%以内。定期手动执行VACUUM可以在对此表的反重叠自动清理启动之前运行。

默认值：4000000000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置偏大可能导致表空间膨胀。

bytea_output

参数说明：设置bytea类型值的输出格式。

参数类型：枚举类型

参数单位：无

取值范围：

- hex：将二进制数据编码为每字节2位十六进制数字。
- escape：采用以ASCII字符序列表示二进制串的方法，同时将那些无法表示成ASCII字符的二进制串转换成特殊的转义序列。当sql_compatibility = 'B'时，设置参数b_format_version = '5.7'和参数b_format_dev_version = 's1'后，escape将不再转义二进制数据编码，而是直接返回原始字节。

须知

当sql_compatibility = 'B'时，设置参数b_format_version = '5.7'且参数bytea_outpur='escape'时，根据参数b_format_dev_version取值不同，在新建表、使用转义字符串E'\\x'插入hex格式数据后，查询时有以下规则：

- 当参数b_format_dev_version = 's1'时，如果客户端字符集和服务端字符集一致，查询不会报错；如果客户端字符集和服务端字符集不一致，当插入数据的编码是服务端字符集的不合法编码时，查询报错。
- 当参数b_format_dev_version = 's2'时，当插入数据的编码是客户端字符集的不合法编码时，查询报错。

默认值：hex

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

xmlbinary

参数说明：设置二进制值在XML中的编码格式。

说明

此参数目前不支持XML类型数据。

版权所有 © 华为云计算技术有限公司

参数类型：枚举类型

参数单位：无

取值范围：

- base64：使用base64编码格式。
- hex：使用hex编码格式。

默认值：base64

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

xmloption

参数说明：对XML数据进行隐式转换或序列化操作时，将其视为文档还是普通的字符串。

说明

此参数目前不支持XML类型数据。

参数类型：枚举类型

参数单位：无

取值范围：

- document：表示HTML格式的文档。
- content：表示普通的字符串。

默认值：content

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_compile_functions

参数说明：设置节点可存储的存储过程和函数的编译结果的最大数量，包括系统自带的存储过程、以及用户自定义的存储过程和函数。编译结果超过该参数的取值后，系统会自动触发内存淘汰，删除未使用的普通存储过程和函数。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：1000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，有助于减少内存占用，提升系统性能。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：编译结果数量超过该参数设定的值后，若系统一直找不到适合删除的存储过程和函数，会占用大量内存，导致性能下降。

gin_pending_list_limit

参数说明：设置当GIN或UGIN索引启用fastupdate时，GIN、UGIN索引融合pending list的阈值。当pending list的容量大于设置值时，会把pending list中的数据批量移动到索引的数据结构中以进行清理。单个索引可通过更改索引存储参数覆盖此设置值。

参数类型：整型

参数单位：kB

取值范围：64 ~ 2147483647

默认值：4096（即4MB）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示gin_pending_list_limit为1024kB；带单位取值1MB，表示gin_pending_list_limit为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：根据更新频率将此参数设置为合理的值，有助于提升索引同步修改的性能。

设置不当的风险与影响：

- 该参数取值设置过大的情况下（大于work_mem），同步融合增量数据的性能以及扫描性能可能会变差；
- 该参数取值设置过小的情况下，会导致pending_list频繁融合，从而影响DML执行性能。

6.3.14.2 区域和格式化

介绍时间格式设置的相关参数。

DateStyle

参数说明：设置日期和时间值的显示格式，以及有歧义的输入值的解析规则。

这个变量包含两个独立的加载部分：输出格式声明（ISO、Postgres、SQL、German）和输入输出的年/月/日顺序（DMY、MDY、YMD、Euro、European、US、NonEuro、NonEuropean、Default）。两个部分可以独立设置或者一起设置。关键字Euro和European等价于DMY；关键字US、NonEuro、NonEuropean等价于MDY。

参数类型：字符串

参数单位：无

取值范围：输出格式声明与输入输出的年/月/日顺序的排列组合（两个部分也可以单独设置）。

- 输出格式声明：ISO、Postgres、SQL、German。
- 输入输出的年/月/日顺序：DMY（Euro、European）、MDY（US、NonEuro、NonEuropean）、YMD。

默认值："ISO, MDY"

版权所有 © 华为云计算技术有限公司

说明

gs_initdb会将这个参数初始化成与**lc_time**一致的值。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

设置建议：优先推荐使用ISO格式。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

IntervalStyle

参数说明：设置区间值的显示格式。

参数类型：枚举类型

参数单位：无

取值范围：

- sql_standard：表示产生与SQL标准规定相匹配的输出。
- postgres：表示产生与PostgreSQL 8.4版本相匹配的输出，当**DateStyle**参数被设为ISO时有效。
- postgres_verbose：表示产生与PostgreSQL 8.4版本相匹配的输出，当**DateStyle**参数被设为non_ISO时有效。
- iso_8601：表示产生与在ISO 8601中定义的“格式与代号”相匹配的输出。
- oracle：表示numtodsinterval函数产生与Oracle格式相匹配的输出结果，详细请参考**numtodsinterval(num, interval_unit)**。

说明

IntervalStyle参数也会影响不明确的间隔输入的说明。

默认值：postgres

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

TimeZone

参数说明：设置显示和解释时间类型数值时使用的时区。

参数类型：字符串

参数单位：无

取值范围：可通过查询视图PG_TIMEZONE_NAMES查看取值范围，请参见**PG_TIMEZONE_NAMES**。

版权所有 © 华为云计算技术有限公司

默认值："PRC"

说明

gs_initdb将设置一个与其系统环境一致的时区值。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

timezone_abbreviations

参数说明：设置服务器接受的时区缩写值。

参数类型：字符串

参数单位：无

取值范围：当前支持的取值有“India”、“Australia”、“Default”。

默认值："Default"

说明

- “Default”表示通用时区的缩写，适合绝大部分情况。但也有其他诸如“Australia”和“India”等用来定义特定时区的安装。
- 不建议用户修改对应取值文件，改写或替换时区缩写文件后，无法保证修改内容实时更新。部分场景需重启数据库集群，修改生效。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

extra_float_digits

参数说明：这个参数为浮点数值调整显示的数据位数，浮点型包括float4、float8以及几何数据类型。参数值加在标准的数据位数上（FLT_DIG或DBL_DIG，视情况而定）。

参数类型：整型

参数单位：无

取值范围：-15 ~ 3

版权所有 © 华为云计算技术有限公司

说明

- 设置为>0时，表示包括部分关键的数据位，但包括的数据位不一定是有效值，可能会导致输出的数字不一定完全有效。该功能适用于转储需要精确恢复浮点数据的场景。
- 设置为<=0时，表示消除不需要的数据位，能够保证输出的数字完全有效。
- M-Compatibility模式数据库下，extra_float_digits仅影响不带标度的float4数据类型显示的数据位数。float8数据类型及带标度的float4数据类型显示的数据位数不受该GUC参数的影响。
- 在通过JDBC和ODBC建立连接的场景下，extra_float_digits的默认值为0。

默认值：3

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

float_shortest_precision

参数说明：这个参数控制是否开启最小精度方法替代sprintf()来实现浮点数的序列化功能。该最小精度方法用于提升浮点数序列化系统函数的性能，即缩短float4out()和float8out()两个系统函数单线程的执行时间。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启最小精度方法。
- off：表示不开启最小精度方法。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表 GUC参数分类**中对应设置方法进行设置。

设置建议：建议新开局点或者对浮点序列化有性能诉求，可以设置此参数。

设置不当的风险与影响：该参数打开后，在extra_float_digits设置不同取值时，float有效数字的精度与未打开前有所差异，extra_float_digits的取值范围为[-15,3]，每个取值需要的有效数字数量如下所示：

extra_float_digits的值	需要的有效数字数量	
	Float4	Float8
-15	1	1
-14	1	1
-13	1	2
-12	1	3
-11	1	4

版权所有 © 华为云计算技术有限公司

-10	1	5
-9	1	6
-8	1	7
-7	1	8
-6	1	9
-5	1	10
-4	2	11
-3	3	12
-2	4	13
-1	5	14
0	6	15
1	7	16
2	8	17
3	9	18

说明

- 对于调用snprintf_s()的原逻辑来说，“需要的有效数字数量”并不代表输出的数字都是有效值。
  1. 当extra_float_digits>0 时，输出的数字不保证完全有效。
  2. 当extra_float_digits<=0 时，输出的数字保证完全有效。
- 对于使用了最小精度方法的新逻辑来说，只输出有效数字。
  1. 当extra_float_digits>0 时，不保证输出的数字达到“需要的有效数字数量”，但是输出的都是有效数字。
  2. 当extra_float_digits<=0 时，输出的数字保证完全有效。

client_encoding

参数说明：设置客户端的字符编码类型。

参数类型：字符串

参数单位：无

取值范围：兼容PG所有的字符编码类型。其中UTF8表示使用数据库的字符编码类型。

版权所有 © 华为云计算技术有限公司

说明

- 使用命令locale -a查看当前系统支持的区域和相应的编码格式，并可以选择进行设置；不同操作系统之间，或者相同操作系统的不同版本之间，排序规则可能存在差异和变化。当应用业务在上述这些操作系统之间进行逻辑迁移时，操作系统排序规则的差异和变化可能会导致索引、分区、排序算子等数据库功能行为的差异，如相同的范围查询语句返回的数据结果集不同、相同的排序查询语句返回的查询结果不同等。应用业务需要根据操作系统发布的locale差异，在业务上排查业务数据是否涉及相关的字符。
- 默认情况下，gs_initdb会根据当前的系统环境初始化此参数，通过locale命令可以查看当前的配置环境。
- 参数建议保持默认值，不建议通过gs_guc工具或其他方式直接在gaussdb.conf文件中设置client_encoding参数，即使设置也不会生效，以保证集群内部通信编码格式一致。
- client_encoding支持设置为GB18030_2022字符编码，当client_encoding=gb18030_2022时，需要保证客户端操作系统支持的GB18030字符集已升级到2022版本。
- 当数据库字符集编码为UTF8，且当前系统支持的区域下相应的编码格式为GB18030时，若客户端操作系统支持的GB18030字符集已升级到2022版本，且数据库中存在客户端GB18030字符集升级前存入的历史数据，则存在以下情况：
  1. set client_encoding=gb18030时，UTF8数据库中的历史数据返回给客户端的字符编码与客户端字符集升级前保持一致，但字符集升级过程中发生映射关系改变的38个字符的字形显示与2022版本保持一致。
  2. set client_encoding=gb18030_2022时，UTF8数据库中的历史数据返回给客户端的字符编码与客户端字符集升级后保持一致，字形与升级前保持一致。这是因为GB18030字符集自身的各版本间存在不完全兼容，可能导致数据不一致。因此，若客户端涉及GB18030字符集升级且存在历史数据的场景，在升级前应进行历史数据的升级转换。
- client_encoding支持设置为ZHS16GBK字符编码，下表展示了ZHS16GBK、GB18030、GB18030_2022字符集所有转换关系对应的server_encoding与client_encoding设置方式：

server_encoding	client_encoding	locale	设置方法
zhs16gbk	utf8	utf8	数据库自动获取locale取值。
utf8	zhs16gbk	gbk	• gsql通过用户手动执行：set client_encoding = zhs16gbk。 • JDBC连接时通过url参数直接指定：characterEncoding=zhs16gbk。

版权所有 © 华为云计算技术有限公司

server_encoding	client_encoding	locale	设置方法
zhs16gbk	zhs16gbk	gbk	• gsql连接时数据库自动获取locale取值，内部进行处理后自动设置client_encoding = zhs16gbk。 • JDBC连接时通过url参数直接指定：characterEncoding=zhs16gbk。
zhs16gbk	gb18030	gb18030	• gsql连接时数据库自动获取locale取值。 • JDBC连接时通过url参数直接指定：characterEncoding=gb18030。
gb18030	zhs16gbk	gbk	• gsql不支持该设置。 • JDBC连接时通过url参数直接指定characterEncoding=zhs16gbk。
zhs16gbk	gb18030-2022	gb18030	• gsql连接时，用户手动执行：set client_encoding = gb18030_2022 。 • JDBC连接时通过url参数直接指定：characterEncoding=gb18030_2022。
gb18030-2022	zhs16gbk	gbk	• gsql不支持该设置。 • JDBC连接时通过url参数直接指定characterEncoding=zhs16gbk。

版权所有 © 华为云计算技术有限公司

server_encoding	client_encoding	locale	设置方法
gb18030	utf8	utf8	数据库自动获取locale取值。
utf8	gb18030	gb18030	• 数据库自动获取locale取值。 • JDBC连接时通过url参数直接指定：characterEncoding=gb18030。
gb18030_2022	utf8	utf8	数据库自动获取locale取值。
utf8	gb18030_2022	gb18030	• gsql连接时，用户手动执行：set client_encoding = gb18030_2022 。 • JDBC连接时通过url参数直接指定：characterEncoding=gb18030_2022。
gb18030	gb18030	gb18030	• 数据库自动获取locale取值。 • JDBC连接时通过url参数直接指定：characterEncoding=gb18030。

版权所有 © 华为云计算技术有限公司

server_encoding	client_encoding	locale	设置方法
gb18030_2022	gb18030_2022	gb18030	• gsql连接时数据库自动获取locale取值，内部进行处理后自动设置client_encoding=gb18030_2022。 • JDBC连接时通过url参数直接指定：characterEncoding=gb18030_2022。

默认值："UTF8"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请根据前端业务的情况确定。尽量客户端编码和服务器端编码一致，提高效率。建议设置为“SQL_ASCII”或“UTF8”。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

lc_messages

参数说明：设置信息显示的语言。

参数类型：字符串

参数单位：无

取值范围：取值范围与系统相关，可通过使用命令locale -a查看。

说明

- 使用命令locale -a查看当前系统支持的区域和相应的编码格式，并可以选择进行设置。
- 在一些系统上，取值范围并不存在，不过仍然允许设置这个变量，只是不会有任何效果。同样，也有可能是所期望的语言的翻译信息不存在。在这种情况下，用户仍然能看到英文信息。
- 默认情况下，gs_initdb会根据当前的系统环境初始化此参数，通过locale命令可以查看当前的配置环境。

默认值："C"

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

lc_monetary

参数说明：设置货币值的显示格式，影响to_char()之类的函数的输出。可接受的值是系统相关的。

参数类型：字符串

参数单位：无

取值范围：取值范围与系统相关，可通过使用命令locale -a查看。

说明

- 使用命令locale -a查看当前系统支持的区域和相应的编码格式，并可以选择进行设置。
- 默认情况下，gs_initdb会根据当前的系统环境初始化此参数，通过locale命令可以查看当前的配置环境。
- 该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

默认值："C"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

lc_numeric

参数说明：设置数值的显示格式，影响to_char之类的函数的输出。

参数类型：字符串

参数单位：无

取值范围：取值范围与系统相关，可通过使用命令locale -a查看。

说明

- 使用命令locale -a查看当前系统支持的区域和相应的编码格式，并可以选择进行设置。
- 默认情况下，gs_initdb会根据当前的系统环境初始化此参数，通过locale命令可以查看当前的配置环境。
- 该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

默认值："C"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

lc_time

参数说明：设置时间和区域的显示格式，影响to_char()之类的函数的输出。

参数类型：字符串

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：取值范围与系统相关，可通过使用命令locale -a查看。

说明

- 使用命令locale -a查看当前系统支持的区域和相应的编码格式，并可以选择进行设置。
- 默认情况下，gs_initdb会根据当前的系统环境初始化此参数，通过locale命令可以查看当前的配置环境。
- 该参数通过执行gs_guc reload修改时，如果当前节点上的某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需与连接节点断开连接后重新连接才会生效。

默认值："C"

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

lc_time_names

参数说明：控制用于显示日期和月份名称及缩写的语言。影响DATE_FORMAT()、DAYNAME()以及MONTHNAME()函数输出结果。不影响STR_TO_DATE()和GET_FORMAT()的表现。

参数类型：字符串

参数单位：无

取值范围：参数应当为简易格式的IETF语言标记，例如en_US、zh_CN等。

默认值："en_US"

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据业务需要，切换合适的语言。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_week_format

参数说明：用于week()函数的默认mode值，取值范围与表现如下表所示。

default_week_format	一周的第一天	范围	第1周定义
0	周日	0~53	今年第一个有周日的周。
1	周一	0~53	今年第一个有4天及以上天数的周。
2	周日	1~53	今年第一个有周日的周。
3	周一	1~53	今年第一个有4天及以上天数的周。

版权所有 © 华为云计算技术有限公司

default_week_format	一周的第一天	范围	第1周定义
4	周日	0~53	今年第一个有4天及以上天数的周。
5	周一	0~53	今年第一个有周一的周。
6	周日	1~53	今年第一个有4天及以上天数的周。
7	周一	1~53	今年第一个有周一的周。

说明

一年的第1周，其中的1代表序号值。根据week_format不同，如第三列所示，一年的第一个周序号可能为0。

参数类型：整型

参数单位：无

取值范围：-2147483648 ~ 2147483647

- 当取值小于0时，会被强制调整为0。
- 当取值大于7时，会被强制调整为7。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据业务需要，选择合适的mode。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.14.3 其他缺省

主要介绍数据库系统默认的库加载参数。

dynamic_library_path

参数说明：设置数据查找动态加载的共享库文件的路径。当需要打开一个可以动态装载的模块并且在CREATE FUNCTION或LOAD命令里面声明的名称没有目录部分时，系统将搜索这个目录以查找声明的文件，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：用冒号分隔（Windows下是分号分隔）的绝对路径列表。

当一个路径名称以特殊变量$libdir为开头时，会替换为GaussDB发布提供的模块安装路径。例如：

```
dynamic_library_path = '/usr/local/lib/gaussdb:/opt/testgs/lib:$libdir'
```


版权所有 © 华为云计算技术有限公司

说明

设置为空字符串时，表示关闭自动路径搜索。

默认值："$libdir"

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：由于参数**local_preload_libraries**和**shared_preload_libraries**配置的动态库的搜索路径也依赖此参数配置，如果设置此参数为空或其他非法路径会导致动态库加载失败，从而导致数据库服务异常。

gin_fuzzy_search_limit

参数说明：设置UGIN或GIN索引返回的集合大小的软上限。当此参数为非默认值时，UGIN索引扫描时将会报错。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_astore_gin

参数说明：GIN索引功能开关。当开关打开时，Astore支持创建GIN索引；当开关关闭时，GIN索引功能不可用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示Astore支持GIN功能打开。
- off：表示Astore支持GIN功能关闭。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议保持默认值，根据实际业务调整。

设置不当的风险与影响：无

enable_astore_gist

参数说明：GiST索引功能开关。当开关打开时，Astore支持创建GiST索引；当开关关闭时，GiST索引功能不可用。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示Astore支持GiST功能打开。
- off：表示Astore支持GiST功能关闭。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议保持默认值，根据实际业务调整。

设置不当的风险与影响：无

local_preload_libraries

参数说明：指定一个或多个共享库，它们在开始连接前预先加载。多个共享库之间用逗号分隔，除了双引号内的库名，所有的库名都转换为小写。

- 并非只有系统管理员才能更改此选项，因此只能加载安装的标准库目录下plugins子目录中的库文件，数据库管理员有责任确保该目录中的库都是安全的。local_preload_libraries中指定的项可以明确含有该目录，例如$libdir/plugins/mylib；也可以仅指定库的名称，例如mylib（等价于$libdir/plugins/mylib）。
- 与shared_preload_libraries不同，在会话开始之前加载模块与在会话中使用到该模块的时候临时加载相比并不具有性能优势。相反，这个特性的目的是为了调试或者测量在特定会话中不明确使用LOAD加载的库。例如针对某个用户将该参数设为ALTER USER SET来进行调试。
- 每一个支持GaussDB的库都有一个“magic block”用于确保兼容性，因此不支持GaussDB的库不能通过这个方法加载。

参数类型：字符串

参数单位：无

取值范围：用逗号分隔的共享库列表，或空字符串。

默认值：""

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当指定的库未找到时，连接会失败。若误设置出错（包括长度超过限制、非法目录等），会影响进程正常拉起。

enable_driver_param_sync

参数说明：控制驱动创建数据库连接时指定的GUC参数是否从CN同步到DN。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示开启。
- off：表示关闭。

默认值：升级上来的数据库实例默认为off，新安装的数据库实例默认为on。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，升级上来的数据库建议在业务评估影响后手动设置为on。

设置不当的风险与影响：当参数为off时，驱动创建连接时设置的参数在执行RESET ALL命令后，CN上的参数取值还原为连接中设置的参数，而DN上的参数取值还原为参数默认值，此时存在CN和DN取值不一样的问题。因此建议升级上来的数据库在业务评估影响后手动设置为on，避免出现参数不一致问题。

6.3.15 锁管理

在GaussDB中，并发执行的事务由于竞争资源会导致死锁。本节介绍的参数主要管理事务锁的机制。

deadlock_timeout

参数说明：设置死锁超时检测时间。当申请的锁超过设定值时，系统会检查是否产生了死锁。该参数仅针对常规锁生效。

参数类型：整型

参数单位：ms（毫秒）

取值范围：1 ~ 2147483647

默认值：1000（即1s）

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 死锁的检查代价是比较高的，服务器不会在每次等待锁的时候都运行这个过程。在系统运行过程中，不经常出现死锁情况，因此在检查死锁前只需等待一个相对较短的时间。deadlock_timeout增加，可以减少死锁检查浪费的时间，但是会减慢真正的死锁错误报告的速度。在负载过重的服务器上，用户可能需要增大deadlock_timeout。这个值的设置应该超过事务持续时间，这样可以减少在锁释放之前就开始死锁检查的问题。
- 当设置**log_lock_waits**为on时，deadlock_timeout决定一个等待时间来将查询执行过程中的锁等待耗时信息写入日志。如果要研究锁延时情况，可以设置deadlock_timeout值比正常情况小。

设置不当的风险与影响：设置过大时，会减慢死锁错误报告的速度；设置过小时，会出现锁释放前就开始检测死锁的问题。

说明

仅支持单节点引发的局部死锁检测，对于涉及多节点协同的分布式全局死锁，暂未提供检测能力。

版权所有 © 华为云计算技术有限公司

lockwait_timeout

参数说明：除update_lockwait_timeout和ddl_lockwait_timeout参数的控制范围之外，其余所有锁类型的申请过程中，单个锁的最长等待时间均由此参数统一控制。若上述锁的等待时长超出该参数设定值，系统会报错。该参数仅针对常规锁生效。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：1200000（20min）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值900，表示lockwait_timeout为900ms；带单位取值20min，表示lockwait_timeout为20min。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：一般设置为默认值。根据业务场景中上述锁冲突超时时间设定该参数的具体数值。若设置小于1s且不为0，则上述锁也会至少等待1s才报告超时。如需实现所有锁类型无限期等待，则需将此参数设置为0（update_lockwait_timeout和ddl_lockwait_timeout参数不具备控制无限等锁行为的功能）。

设置不当的风险与影响：申请上述锁时若发生锁冲突，参数设置过大将导致阻塞时间延长；设置过小则可能引发更多的上述锁超时报错。

update_lockwait_timeout

参数说明：在业务允许并发更新的情况下，该参数控制并发更新同一行数据时，申请单个transactionid锁或tuple锁的最长等待时间。当申请上述锁的等待时间超过设定值时，系统会报错。该参数仅针对常规锁生效。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：120000（2min）

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值900，表示update_lockwait_timeout为900ms；带单位取值2min，表示update_lockwait_timeout为2min。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：一般设置为默认值。设置值主要取决于并发更新同一行的事务的workload。一般TP业务，执行时间较短小于两分钟，因此等待并发更新事务提交时间较短，按默认值2min的配置误报锁超时概率较小；如果业务出现大量并发更新同一行事务，执行时间超过两分钟，可以调大该参数避免等锁超时误报。若设置小于1s，则上述锁也会至少等待1s才报告超时。

设置不当的风险与影响：并发更新同一行数据时，若申请transactionid锁或tuple锁发生锁冲突，参数设置过大将导致阻塞时间延长；设置过小则可能引发更多transactionid锁或tuple锁超时报错。

版权所有 © 华为云计算技术有限公司

ddl_lockwait_timeout

参数说明：该参数控制部分DDL业务在申请表对象、索引对象、存储过程/package/函数对象的高级别（5~8级）对象锁时，单个锁的最长等待时间。当申请的锁等待时间超过设定值时，系统会报错。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：120000（2min）

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值900，表示ddl_lockwait_timeout为900ms；带单位取值2min，表示ddl_lockwait_timeout为2min。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：一般设置为默认值。根据DDL业务对上述高级锁超时时间的需要，预期最大等待时间是多少，该参数就设置为多少。若设置小于1s，则上述锁也会至少等待1s才报告超时。

设置不当的风险与影响：DDL业务申请上述高级锁时若发生锁冲突，参数设置过大将导致阻塞时间延长；设置过小则可能引发更多的高级锁超时报错。

max_locks_per_transaction

参数说明：当数据库连接数达到最大时，每个事务能够持有对象锁的平均数量。

> **说明**
>
> 这个参数并不严格限制每个事务最多可以持有多少对象锁。如果连接数没有达到最大值**max_connections**，那么每个事务能够持有对象锁的平均数量会增大。

参数类型：整型

参数单位：无

取值范围：10 ~ 2147483647

默认值：256

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 共享锁的哈希表的大小是以假设任意时刻最多只有max_locks_per_transaction*(**max_connections**+**max_prepared_transactions**)个独立的对象需要被锁住为基础进行计算的，进而推导出max_locks_per_transaction >= 业务事务的并发数 * 业务的每个事务加对象锁的数目/(max_connections+max_prepared_transactions)，计算结果不超过参数取值范围的上限。
- 不超过设定数量的多个对象可以在任意时刻同时被锁定。当在一个事务里面修改很多不同的表时，可能需要提高这个参数值。
- 增大这个参数可能导致GaussDB请求更多的System V共享内存，有可能超过操作系统的缺省配置，导致数据库启动失败。建议配置该参数大小不超过**max_process_memory** * 10% / (**max_connections** + **job_queue_processes** +

版权所有 © 华为云计算技术有限公司

autovacuum_max_workers + **max_inner_tool_connections** + **max_concurrent_autonomous_transactions** + 1280 + **max_prepared_transactions** * TWOPHASE_PART) / (512字节)。其中1280为预留辅助线程固定数及其他线程占用数的预估值，max_process_memory的单位需换算成字节。

- 当运行备机时，请将此参数设置不小于主机上的值，否则不允许在备机上查询操作。
- 设置该参数变小后，若在重启前备机节点故障，则该备机会故障恢复失败。建议参数设置回原值后重启恢复环境。
- 在降规格及容灾场景下，变更此参数时不建议频繁DDL，否则会导致内存报错。

设置不当的风险与影响：设置过大时，可能导致数据库启动失败。

max_pred_locks_per_transaction

参数说明：当数据库连接数达到最大时，每个事务能够持有断定锁的平均数量。

说明

这个参数并不严格限制每个事务最多可以持有多少断定锁。如果连接数没有达到最大值**max_connections**，那么每个事务能够持有断定锁的平均数量会增大。

参数类型：整型

参数单位：无

取值范围：10 ~ 2147483647

默认值：64

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- 共享的断定锁表的大小是以假设任意时刻最多只有max_pred_locks_per_transaction*(**max_connections** +**max_prepared_transactions**) 个独立的对象需要被锁住为基础进行计算的。不超过设定数量的多个对象可以在任一时刻同时被锁定。当在一个事务里面修改很多不同的表时，可能需要提高这个默认数值。只能在服务器启动的时候设置。
- 增大这个参数可能导致GaussDB请求更多的System V共享内存，有可能超过操作系统的缺省配置。

设置不当的风险与影响：设置过大时，可能导致数据库启动失败。

gs_clean_timeout

参数说明：GaussDB数据库中事务处理使用的是两阶段提交方法，当有两阶段事务残留时，该事务通常会持有表级锁，导致其它连接无法加锁，此时需要调用gs_clean工具对集群中两阶段事务进行清理，gs_clean_timeout是控制CN周期性调用gs_clean的时间。gs_clean也可用来定时清理残留临时表。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483

版权所有 © 华为云计算技术有限公司

默认值：60

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如：不带单位取值1，表示gs_clean_timeout为1s；带单位取值1min，表示gs_clean_timeout为1min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大时，会使两阶段事务和残留临时表清理周期延长，可能导致清理前其他连接无法加锁。

partition_lock_upgrade_timeout

参数说明：在执行某些查询语句的过程中，需要将分区表上的锁级别由允许读的ExclusiveLock级别升级到读写阻塞的AccessExclusiveLock级别。如果此时已经存在并发的读事务，那么该锁升级操作将阻塞等待。partition_lock_upgrade_timeout为尝试锁升级的等待超时时间。

举例：在分区表上进行MERGE PARTITION和CLUSTER PARTITION操作时，都利用了临时表进行数据重排和文件交换，为了最大程度提高分区上的操作并发度，在数据重排阶段给相关分区加锁ExclusiveLock，在文件交换阶段加锁AccessExclusiveLock。加锁方式是尝试性加锁，加锁成功了则立即返回，不成功则等待50ms后继续下次尝试，加锁超时时间使用会话级设置参数partition_lock_upgrade_timeout。

参数类型：整型

参数单位：s（秒）

取值范围：-1 ~ 3000。-1表示无限等待，即不停地尝试锁升级，直到加锁成功。

默认值：1800

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为-1时，长时间无法升级锁可能导致业务阻塞。

fault_mon_timeout

参数说明：轻量级死锁检测周期。

参数类型：整型

参数单位：min（分钟）

取值范围：0 ~ 1440

默认值：5

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大时，会无法及时发现死锁问题而减慢死锁错误报告的速度。

版权所有 © 华为云计算技术有限公司

enable_online_ddl_waitlock

参数说明：控制DDL是否会阻塞等待pg_advisory_lock/pgxc_lock_for_backup等集群锁。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：主要用于OM在线操作场景，不建议用户设置。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

xloginsert_locks

参数说明：控制用于并发写预写式日志锁的个数。主要用于提高写预写式日志的效率。

参数类型：整型

参数单位：无

取值范围：1 ~ 1000。若CPU为NUMA架构，数值必须为NUMA节点数量的整数倍。

默认值：16

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：ARM CPU架构在并发压力大的场景下，可能因为原子操作的竞争而导致概率性的性能抖动，可以适当降低该参数值而获取相对稳定的运行性能。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

num_internal_lock_partitions

参数说明：控制内部轻量级锁分区的个数。主要用于各类场景的性能调优。

说明

- 参数内容以关键字和数字的KV方式组织，各个不同类型锁之间以逗号隔开。
- 先后顺序对设置结果不影响。例如"CLOG_PART=256,CSNLOG_PART=512"等同于"CSNLOG_PART=512,CLOG_PART=256"。
- 重复设置同一关键字时，以最后一次设置为准。例如"CLOG_PART=256,CLOG_PART=2"，设置的结果为CLOG_PART=2。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- CLOG_PART：CLOG文件控制器的个数。最小值为1，最大值为256。
- CSNLOG_PART：CSNLOG文件控制器的个数。最小值为1，最大值为512。
- LOG2_LOCKTABLE_PART：常规锁表锁分区个数的2对数。最小值4，即锁分区数为16；最大值为12，即锁分区数为4096。
- TWOPHASE_PART：两阶段事务锁的分区数。最小值为1，最大值为64。
- FASTPATH_PART：每个线程可以不通过主锁表拿锁的最大锁个数。最小值为20，最大值为10000。
- FASTPATH_PART_PER_SHARD：每个线程通过FASTPATH申请锁时，单个FASTPATH分区的大小。最小值20，最大值10000。
- FASTPATH_PROCEDURE_PART：每个线程可以不通过主锁表申请的存储过程对象锁的最大锁个数。最小值为0，最大值为10000。

默认值：
"CLOG_PART=256,CSNLOG_PART=512,LOG2_LOCKTABLE_PART=4,TWOPHASE_PART=1,FASTPATH_PART=20,FASTPATH_PART_PER_SHARD=20,FASTPATH_PROCEDURE_PART=100"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：

- CLOG_PART：增大该值可以提高CLOG日志写入效率，提升事务提交性能，但是会增大内存使用；减小该值会减少相应内存使用，但可能使得CLOG日志写入冲突变大，影响性能。
- CSNLOG_PART：增大该值可以提高CSNLOG日志写入效率，提升事务提交性能，但是会增大内存使用；减小该值会减少相应内存使用，但可能使得CSNLOG日志写入冲突变大，影响性能。
- LOG2_LOCKTABLE_PART：增大该值可以提升正常流程常规锁获取锁的并行度，但是会增大内存使用，对于等待事件在LockMgrLock时，可以调大该锁增加性能。
- TWOPHASE_PART：调大该值可以提高两阶段事务提交的并发数，但是会增大内存使用。当两阶段事务报错槽位不足时（CN/DN出现日志：Maximum number of prepared transactions reached, please increase max_parepared_transactions (currently max_parepared_transactions) or wait gs_clean to clean reserved prepared transactions.），建议调大该值，避免事务锁冲突严重。
- FASTPATH_PART：对于分区表读取、更新、插入、删除操作且等待事件在LockMgrLock时，可以通过调大该值避免获取LockMgrLock提升性能，建议调整数量大于等于分区数*（1+本地索引数量）+全局索引数量+10。调大该值会额外增加内存，fastpath增量内存=((fastpath增加量/20)*8 + fastpath增加量*12) * 线程池大小，单位是字节。
- FASTPATH_PART_PER_SHARD：该值表示一个FASTPATH分区的大小，配置较大时会增加遍历FASTPATH分区的耗时，可能导致性能劣化，建议使用默认值。
- FASTPATH_PROCEDURE_PART：对于执行存储过程等待事件在LockMgrLock时，可以通过调大该值避免获取LockMgrLock以提升性能，建议调整数量大于存储过程嵌套调用的最大深度。调大该值会额外增加内存，增量内存=((参数值增加量/20)*8 + 参数值增加量*12) * 线程池大小，单位是字节。

设置不当的风险与影响：设置不当可能会影响数据库性能。

版权所有 © 华为云计算技术有限公司

enable_wait_exclusive_lock

参数说明：控制ProcArrayLock的排他锁hang死检测与治愈功能的开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

barrier_lock_timeout

参数说明：持有barrier锁的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

默认值：30

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_activity_history_query_number

参数说明：发生死锁后，日志可打印事务内历史命令的最大数量。

参数类型：整型

参数单位：无

取值范围：1 ~ 1000

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际单事务命令数量设置。建议设置10以内的值。数据库会预先分配内存，内存大小为track_activity_query_size * 最大预估线程数量 * track_activity_history_query_number。max_connections为影响线程数量最大因素，当max_connections参数设置较大时，会导致最大预估线程数量变大（最大可能线程数超50万），最终分配较大内存，这种情况下需要谨慎增加track_activity_history_query_number数量。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置过大会导致死锁时日志大量打印及分配较大内存。

enable_xid_abort_check

参数说明：事务提交时校验事务ID回滚开关是否开启。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_partition_ddl_lowlevel_lock

参数说明：执行分区表DDL (ADD/DROP/TRUNCATE/EXCHANGE/MOVE/RENAME)的过程中，会对表加锁。该参数用于控制表的锁级别。该参数为on时，对表加4级锁，此时允许对该表的其他分区并发执行DDL/DML；该分区为off时，对表加8级锁，此时不允许对该表的其他分区并发执行DDL/DML。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：升级上来的数据库实例默认为off，新安装的数据库实例默认为on。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当参数为on时，允许对表的其他分区并发执行DDL/DML，具有性能优势，但是由于网络延时等因素，并发的DDL和DML在不同DN上的实际执行顺序可能不同，会导致DN间的分布式死锁。当参数为off时，不允许对表的其他分区并发执行DDL/DML，无性能优势，但是没有死锁风险。

6.3.16 版本和平台兼容性

6.3.16.1 历史版本兼容性

GaussDB介绍数据库的向下兼容性和对外兼容性特性的参数控制。数据库系统的向后兼容性能够为旧版本的数据库应用提供支持。本节介绍的参数主要控制数据库的向后兼容性。

版权所有 © 华为云计算技术有限公司

array_nulls

参数说明：控制数组输入解析器是否将没有用引用的NULL识别为数组的一个NULL元素。M兼容模式下不支持该参数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许向数组中输入空元素。
- off：表示向下兼容旧模式。仍然能够创建包含NULL值的数组。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

backslash_quote

参数说明：控制字符串文本中的单引号是否能够用\'表示。

说明

在字符串文本符合SQL标准的情况下，\没有任何其他含义。这个参数影响的是如何处理不符合标准的字符串文本，包括明确的字符串转义语法（E'...'）。

参数类型：枚举类型

参数单位：无

取值范围：

- on：表示一直允许用\'表示。
- off：表示不能用\'表示。
- safe_encoding：表示当客户端字符集编码在多字节字符末尾不包含\的ASCII值时，允许使用。

默认值：safe_encoding

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

escape_string_warning

参数说明：控制是否对普通字符串中使用的反斜杠转义发出警告。

- 如果需要使用反斜杠作为转义，可以调整为使用转义字符串语法（E'...'）来做转义，因为在SQL标准中，普通字符串的默认行为将反斜杠作为一个普通字符。
- 这个变量可以帮助定位需要改变的代码。
- 使用E转义会导致部分场景下日志记录不全。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：对普通字符串中使用的反斜杠转义发出警告。
- off：不对普通字符串中使用的反斜杠转义发出警告。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

quote_all_identifiers

参数说明：当数据库生成SQL时，控制是否强制引用所有的标识符（包括非关键字）。这将影响到EXPLAIN的输出及函数的结果，例如pg_get_viewdef。详细说明请参见**gs_dump导出数据库信息**章节中的--quote-all-identifiers选项。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开强制引用。
- off：表示关闭强制引用。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

sql_inheritance

参数说明：用来控制继承表的访问策略，该参数是为了兼容旧版本而设置的。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示可以访问子表。
- off：表示不可以访问子表，即默认使用ONLY关键字。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

standard_conforming_strings

参数说明：在非M-Compatibility兼容模式数据库中，该参数控制普通字符串文本（'...'）是否按照SQL标准将反斜杠视为普通文本。应用程序可通过检查此参数判断字符串文本的处理方式。建议明确使用转义字符串语法（E'...'）来转义字符，该语法优先级高于本参数，即使本参数设置为on，转义字符串中的反斜杠仍会被解释为转义字符。M-Compatibility兼容模式数据库下该参数设置行为不生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示按照SQL标准把反斜杠当作普通文本。
- off：表示反斜杠在字符串字面值中被解释为转义字符。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

synchronize_seqscans

参数说明：控制并行执行的序列扫描是否相互同步。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示扫描可能从表的中间开始，然后选择“环绕”方式来覆盖所有的行，目的是与已在执行的扫描保持同步。这种同步机制可以提高并行序列扫描的性能，但可能导致没有ORDER BY的查询返回乱序结果。
- off：禁用序列扫描的同步行为，每个扫描都从表的起始位置开始。可以确保没有ORDER BY的查询返回与物理存储顺序一致的结果，但并行扫描效率可能会降低。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_beta_features

参数说明：控制是否开启某些非正式发布的特性，仅用于POC验证，例如GDS表关联操作。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示开启这些功能受限的特性，保持前向兼容。
- off：表示禁止使用这些特性。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：这些特性属于延伸特性，建议用户谨慎开启，在某些功能场景下可能存在问题。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_recordtype_check_strict

参数说明：控制是否要对PL/SQL中的record类型进行严格性校验，详见说明部分。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示会在编译procedure、function、package时，校验创建的record类型中不支持的功能，以及提供procedure、function里创建的record类型not null功能。
- off：表示向下兼容旧式模式，不会对record类型进行校验。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为off时，部分指定了not null约束和定义了列默认值的record变量在赋值时功能不生效。

说明

- 直接安装当前版本数据库时，该参数默认值为on。从历史版本升级上来的数据库，若升级前的基础版本不支持该参数，则升级后该参数默认值为off；若升级前的基础版本支持该参数，升级后保持该参数当前设置。
- 打开该参数后有以下三点行为变更：
  - 存储过程、函数内部创建的record类型，列约束not null功能生效。
  - package里创建的record类型，如果有一列指定了not null或default，则会编译报错。通过访问package.rec类型来创建的变量不支持not null和default功能。
  - 其他类型嵌套一个record类型，如果有一列指定了not null或default，则会编译报错。创建嵌套record类型的变量，该变量的record元素不支持not null和default功能。

plsql_block_concat_typename

参数说明：控制PL/SQL的子块中自定义类型名字的拼接方式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示PL/SQL子块中的自定义类型，其在SQL中实际创建的名字拼接格式为OID.[m.n…].typename，其中OID为存储过程的OID，typename为自定义类型的

版权所有 © 华为云计算技术有限公司

名称，m,n则为子块的序号，多层子块嵌套时，则需要多层序号，如果仅有单层子块，则仅需m一个数字拼接即可。

- off：表示向下兼容旧式模式，名字拼接方式为OID.typename。

默认值：

- 直接安装新版本时，默认值为on。
- 从历史版本升级上来的数据库，如果升级的基础版本不支持该GUC参数，默认值为off；如果升级的基础版本支持该GUC参数，默认值为on。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为off时，当子块内的自定义类型重名时，SQL中无法区分不同子块的自定义类型。

default_with_oids

参数说明：在没有声明WITH OIDS和WITHOUT OIDS的情况下，这个选项控制在新创建的表中CREATE TABLE和CREATE TABLE AS是否包含一个OID字段。它还决定SELECT INTO创建的表里面是否包含OID 。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在新创建的表中，CREATE TABLE和CREATE TABLE AS包含一个OID字段。
- off：表示在新创建的表中，CREATE TABLE和CREATE TABLE AS不包含一个OID字段。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不推荐在用户表中使用OID，故默认设置为off。需要带有OID字段的表应该在创建时声明WITH OIDS 。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

system_view_version

参数说明：控制系统视图的版本。具体详见**表1 系统视图版本参数说明**。所有版本向下兼容，例如system_view_version=3时同时满足版本2与版本1的所有特性。

参数类型：整型

参数单位：无

取值范围：0 ~ 9999

默认值：1

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：升级场景不做修改，安装场景配置为**表1 系统视图版本参数说明**中的最大版本。

设置不当的风险与影响：系统视图部分字段行为存在变更，参考**表1 系统视图版本参数说明**。

表 6-28 系统视图版本参数说明

取值	说明
0	默认行为。
1	当GaussDB中不存在已准备好的事务时，V$GLOBAL_TRANSACTION视图的PREPARECOUNT字段由NULL变为0。

enable_subtype_inherit_notnull_constraint

参数说明：当subtype作为集合类型和数组类型的子元素时，控制subtype的非空约束是否生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：当subtype作为集合类型和数组类型的子元素时，subtype的非空约束生效，无法给元素赋空值。
- off：当subtype作为集合类型和数组类型的子元素时，subtype的非空约束不生效，可以给元素赋空值。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为off时，当subtype作为集合类型和数组类型的子元素时，subtype的非空约束不生效。

说明

直接安装当前版本数据库时，该参数默认值为on。从历史版本升级上来的数据库，若升级前的基础版本不支持该参数，则升级后该参数默认值为off；若升级前的基础版本支持该参数，升级后保持该参数当前设置。

enable_create_rule_on_select_to_table

参数说明：在表上创建名为“_RETURN”的“ON SELECT TO ... DO INSTEAD”规则来将表转为视图时，控制该行为是否执行成功。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：在表上创建名为“_RETURN”的“ON SELECT TO ... DO INSTEAD”规则来将表转为视图时，创建成功。
- off：在表上创建名为“_RETURN”的“ON SELECT TO ... DO INSTEAD”规则来将表转为视图时，创建失败。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：在表上创建名为“_RETURN”的“ON SELECT TO ... DO INSTEAD”规则来将表转为视图是不安全的操作，可能导致意外错误。以这种方式创建的视图在失效后将无法重新编译。请在充分理解参数含义，并经过测试验证后进行修改。

6.3.16.2 平台和客户端兼容性

很多平台都使用数据库系统，数据库系统的对外兼容性给平台提供了很大的便利。

a_format_date_timestamp

参数说明：控制返回的日期时间。在ORA模式下，当事务开启时，函数current_date()、current_timestamp()、localtimestamp()返回当前SQL启动的时间戳。

参数类型：布尔型

参数单位：无

取值范围：

- on：返回当前SQL启动的时间戳。
- off：返回事务开启的日期或日期及时间。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在开启事务需要返回系统时间时可开启该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

在参数**a_format_version**值为10c和**a_format_dev_version**值为s7的情况下，支持开启该GUC参数的行为，具体请参见**a_format_dev_version**。

compatiable_o_function_version

参数说明：通过指定版本号，改变对重载的系统函数的选择逻辑。该参数仅在ORA模式下生效。

参数类型：字符串

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- "auto"：表示选择范围为从507.1.0开始添加的重载的系统函数。
- "*具体的版本号*"：版本号为507.1.0及之后的版本，表示选择范围为从507.1.0开始到该版本之间的重载的系统函数。

默认值："auto"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：调整该参数会改变重载的系统函数的选择逻辑，可能导致选择到错误的重载的系统函数，请谨慎操作。

说明

假如选择的版本号为507.1.0.SPC1100，则需将该参数的值设置为"507.1.0.1100"；假如选择的版本号为507.1.0（没有SPC版本号），则需将该参数的值设置为"507.1.0.0"。

convert_string_to_digit

参数说明：设置隐式转换优先级，是否优先将字符串转为数字。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示优先将字符串转为数字。
- off：表示不优先将字符串转为数字。

默认值：on

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：调整该参数会修改内部数据类型转换规则，导致不可预期的行为，请谨慎操作。

character_set_connection

参数说明：用于设置常量字符串的字符集。修改该参数会同步将**collation_connection**修改成该字符集的默认字符序。

- 在sql_compatibility='MYSQL'兼容模式下，设置b_format_version='5.7'、b_format_dev_version='s2'时生效。
- 在M-Compatibility模式下生效。

参数类型：字符串

参数单位：无

取值范围：

- sql_compatibility='MYSQL'兼容性模式下支持的字符集，请参见**COLLATE collation**。

版权所有 © 华为云计算技术有限公司

- 在M-Compatibility模式下取值范围为其支持的字符集，请参见《M-Compatibility开发指南》中“SQL参考 > 字符集与字符序”。

默认值：与当前**server_encoding**保持一致。

设置方式：该参数属于USERSET类型参数，但不支持通过GUC工具设置，仅支持方式三设置，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置的字符集和当前环境不符时，可能导致多次转码，影响性能。

collation_connection

参数说明：用于设置常量字符串的字符序。修改该参数会同步将**character_set_connection**修改成该字符序的默认字符集。

- 在sql_compatibility='MYSQL'兼容模式下，设置b_format_version='5.7'、b_format_dev_version='s2'时生效。
- 在M-Compatibility模式下生效。

参数类型：字符串

参数单位：无

取值范围：

- sql_compatibility='MYSQL'兼容性模式下支持的字符序，请参见 **COLLATE collation**。
- 在M-Compatibility模式下取值范围为其支持的字符序，请参见《M-Compatibility开发指南》中“SQL参考 > 字符集与字符序”

默认值：当前**server_encoding**的默认字符序，若没有server_encoding的默认字符序，则为“default”。

设置方式：该参数属于USERSET类型参数，但不支持通过GUC工具设置，仅支持方式三设置，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置的字符序和当前环境不符时，可能导致多次转码，影响性能。

character_set_results

参数说明：用于设置返回结果的字符集。

- 在sql_compatibility='MYSQL'兼容模式下，设置b_format_version='5.7'、b_format_dev_version='s2'时生效。
- 在M-Compatibility模式下生效。

参数类型：字符串

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- sql_compatibility='MYSQL'兼容性模式下支持的字符集，请参见**COLLATE collation**。
- 在M-Compatibility模式下取值范围为其支持的字符集，请参见《M-Compatibility开发指南》中“SQL参考 > 字符集与字符序”。
- 还可以设置成null或''，此时直接以数据库的字符集输出。

默认值：与当前**client_encoding**保持一致。

设置方式：该参数属于USERSET类型参数，但不支持通过GUC工具设置，仅支持方式三设置，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置的字符集和当前环境不符时，可能会出现乱码。

enable_multiple_charset

参数说明：该参数用于控制在M-Compatibility模式下，是否可以创建表、列、Schema指定字符集和当前数据库字符集不同。

参数类型：布尔型

参数单位：无

取值范围：

- on：可以创建表、列、Schema指定字符集和当前数据库字符集不同。
- off：不可以创建表、列、Schema指定字符集和当前数据库字符集不同。

默认值：on

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_view_invalidation

参数说明：用于控制视图失效功能的开启与关闭。开启该参数后，若视图依赖对象发生变更，上层依赖视图将级联失效。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开视图失效功能。
- off：关闭视图失效功能。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

版权所有 © 华为云计算技术有限公司

说明

视图失效重编译请参见《开发指南》中“特性使用指导 > 失效重编译”章节。

enable_proc_exec_check

参数说明：用于控制存储过程执行时，对依赖对象有效性进行检查。参数打开后，若依赖对象已经被失效，则不会继续执行当前存储过程，返回报错。

参数类型：布尔型

参数单位：无

取值范围：

- on：设置为on时，会在存储过程执行时进行有效性检查。
- off：设置为off时，不会在存储过程执行时进行有效性检查。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据需求场景进行配置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

说明

若从507.0版本之前升级至507.0及之后版本，该参数会默认设置为off。若新安装版本为507.0及之后版本，该参数默认设置为on。

enable_show_view_original_definition

参数说明：该参数用于控制在ORA模式数据库下查询视图定义时，显示用户输入的原始SELECT语句。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示查询视图定义时，显示用户输入的原始SELECT语句。
- off：表示查询视图定义时，按照将QUERY TREE反解析方式显示SELECT语句。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

在ORA模式下，该参数会影响\d+、dba_views、all_views、user_views、information_schema.views、pg_views、pg_get_viewdef等查询视图信息的函数或视图的显示结果。

版权所有 © 华为云计算技术有限公司

fixed_date

参数说明：该参数用于控制在ORA模式数据库下，设置参数后可固定sysdate的值。

参数类型：字符串

参数单位：无

取值范围：格式为timestamp所支持的所有格式，数值范围为datea所表示的时间范围（0001-9999）。日期设置不支持负数，负数表示公元前时间，如需要设置可在时间后加上'BC'。

默认值：""

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。该参数不允许在生产环境开启，仅用于调试。

mapping_date_to_datea

参数说明：该参数用于控制在ORA兼容模式下，是否用datea类型替换date类型。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示用datea类型替换date类型。
- off：表示date类型的行为维持默认状态。

默认值：新安装数据库为on；若升级的基础版本不存在该guc参数，默认为off；若升级的基础版本存在该GUC参数，保留原设置。

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。支持在数据库安装时指定，如未指定，默认值为on，表示用datea类型替换date类型。可通过以下方式用于安装时打开或关闭该参数：

```
gs_install --dn-guc="mapping_date_to_datea=off" --cn-guc="mapping_date_to_datea=off" --gsinit-parameter="-- mapping_date_to_datea off"
```

设置不当的风险与影响：在ORA兼容模式下，可能导致date类型的范围、精度、格式等与预期不符。

说明

- 该参数在升级观察期设置不生效。
- 该参数在安装数据库时指定，不建议在数据库启动后对参数进行修改。若在数据库运行过程中修改，系统视图、系统表以及修改参数之前与date数据类型相关数据将保持修改参数前的状态，不会发生变化。新建立的对象则会使用datea类型替换date类型。
- 升级场景下，升级提交后若修改该参数值，低版本上创建的数据将保持参数修改前的状态，不会发生变化。例如低版本该参数值设置为off，升级到高版本后设置该参数为on，低版本上创建的date类型数据仍为映射为timestamp，如需使用新的datea类型，请按照数据类型中的说明重新创建对象。

版权所有 © 华为云计算技术有限公司

nls_date_format

参数说明：在ORA兼容模式下，用于datea类型和字符类型转换时的格式控制。

参数类型：字符串

参数单位：无

取值范围：nls_date_format支持的格式请参见**表1-12**，且支持'-'、'/'、','、'.'、'%'、':'或用""包含的字符串作为分隔符。

默认值："DD-MON-RR"

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求设置，建议不要频繁变更。

设置不当的风险与影响：可能导致datea类型输入时格式校验失败，产生报错，或输入时间结果与预期不符。

nls_timestamp_format

参数说明：设置时间戳默认格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值："DD-Mon-YYYY HH:MI:SS.FF AM"

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

nls_timestamp_tz_format

参数说明：在ORA兼容模式下，设置带时区时间戳默认格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值："DD-Mon-YYYY HH:MI:SS.FF AM"

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

此参数在参数**a_format_version**值为10c和**a_format_dev_version**值为s1的情况下有效。

版权所有 © 华为云计算技术有限公司

nls_nchar_characterset

参数说明：搭配nchr(cvalue int|bigint)系统函数使用，设置国家字符集。

参数类型：字符串

参数单位：无

取值范围：“AL16UTF16”、“UTF8”，不区分大小写。

默认值："AL16UTF16"

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 数据库不支持国家字符集，该参数只是为了兼容ORA数据库，获取用户想使用的国家字符集。
- 该GUC参数仅限于nchr(cvalue int|bigint)函数的使用。

group_concat_max_len

参数说明：搭配函数group_concat()使用，限制其返回值长度，超长截断，并产生相应报错/告警。当sql_mode设置含有STRICT_TRANS_TABLES时，DML语句中会产生报错，其余场景中提示告警。

参数类型：整型

参数单位：无

取值范围：0 ~ 9223372036854775807，目前实际生效的最大长度是1073741823，超出此长度后会有out of memory的报错。

默认值：1024

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小可能导致group_concat()返回的结果被截断。

lastval_supported

参数说明：控制是否可以使用lastval()函数。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持lastval()函数，同时nextval()函数不支持下推。
- off：表示不支持lastval()函数，同时nextval()函数可以下推。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_function_args

参数说明：函数参数最大个数。

参数类型：整型

参数单位：个

取值范围：8192

默认值：8192

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

lower_case_table_names

参数说明：该参数用于控制在M-Compatibility模式数据库下，查询用户表名（包括表别名）、用户视图名、Schema名（内部库名）以及information_schema系统视图中以上对象对应字段的值的大小写是否敏感。当前版本该参数取值为0，不支持修改。

参数类型：整型

参数单位：无

取值范围：

- 0：用户表名、用户视图名、Schema名（内部库名）使用CREATE TABLE、CREATE VIEW、CREATE SCHEMA（CREATE DATABASE）语句中指定的字母大小写格式存储在系统表上。information_schema系统视图中以上对象对应字段的值名称比较区分大小写。
- 1：用户表名在系统表上以小写格式存储，名称比较不区分大小写；在存储和查找操作时，将所有用户表名转换为小写格式。此行为也适用于Schema名（内部库名）、表别名、用户视图名和information_schema系统视图中以上对象对应字段的值。

默认值：0

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无。

max_subpro_nested_layers

参数说明：嵌套子程序的最大嵌套深度。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：无

取值范围：0 ~ 100

说明

当设置为0时，不允许使用嵌套子程序。

默认值：3

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据最大嵌套情况设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

transform_null_equals

参数说明：是否控制表达式expr=NULL（或NULL=expr）当作expr IS NULL处理。正确的SQL标准兼容的expr=NULL总是返回NULL（未知）。Microsoft Access里的过滤表单生成的查询使用expr=NULL来测试空值。打开这个选项后，可以使用该接口来访问数据库。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示控制表达式expr=NULL（或NULL=expr）当作expr IS NULL处理，如果expr得出NULL值则返回真，否则返回假。
- off：表示不控制，即expr=NULL总是返回NULL（未知）。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

新用户经常在涉及NULL的表达式上语义混淆，故默认值设为off。

support_extended_features

参数说明：控制是否支持数据库的扩展特性。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持数据库的扩展特性。
- off：表示不支持数据库的扩展特性。

版权所有 © 华为云计算技术有限公司

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

sql_compatibility

参数说明：控制数据库的SQL语法和语句行为同哪一个主流数据库兼容。

参数类型：枚举类型

参数单位：无

取值范围：

- ORA：表示同Oracle兼容。
- TD：表示同Teradata兼容。
- MYSQL：表示同MySQL兼容。
- PG：表示同PostgreSQL兼容。
- M：表示M-Compatibility。

默认值：MYSQL

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

说明

- 该参数只能在执行**CREATE DATABASE**命令创建数据库的时候设置。
- 在数据库中，该参数只能是确定的一个值，要么始终设置为ORA，要么始终设置为TD，请勿任意改动，否则会导致数据库行为不一致。
- 在M-Compatibility模式数据库分布式场景下，如果告警是在CN上产生的，则会产生WARNING信息；如果告警是在DN上产生的，暂不支持产生WARNING信息。

b_format_behavior_compat_options

参数说明：数据库MYSQL模式兼容性行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前只支持**表1 MYSQL模式兼容性配置项**，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开。

默认值：""

版权所有 © 华为云计算技术有限公司

说明

当b_format_version参数不为""时，b_format_behavior_compat_options会被设置为"all"，而且不能被修改。

表 6-29 MYSQL 模式兼容性配置项

兼容性配置项	兼容性行为控制
default_collation	默认字符序前向兼容开关。 • 若不设置此配置项，在未显式指定字符类型字段的字符集或字符序且表级字符序也为空时，字段为default字符序。 • 若设置此配置项，当表级字符序不为空时，字符类型字段的字符序继承表级字符序，当表级字符序为空时，字符类型字段的字符序设置为数据库编码对应的默认字符序。
all	打开所有语法控制开关。 all不能同时和其他配置项一起指定。表格中除all外所有配置项逗号隔开的含义和取值为all含义相同。

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：控制一些MYSQL兼容性特性是否可用，若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

m_format_behavior_compat_options

参数说明：数据库M-Compatibility模式的兼容性行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前只支持**表2 M-compatibility模式兼容性配置项**，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开。

默认值：

- 新安装的数据库："grant_database_nomapping"。
- 从历史版本升级上来的数据库，若历史版本不支持该GUC参数，升级后的默认值为""；若历史版本支持该GUC参数，升级后默认值保留原设置。

版权所有 © 华为云计算技术有限公司

表 6-30 M-Compatibility 模式兼容性配置项

兼容性配置项	兼容性行为控制

grant_database_nomapping	M-Compatibility模式数据库下GRANT | REVOKE ON DATABASE语法行为配置项。
	• 不设置此配置项时：GRANT | REVOKE ON DATABASE语法映射成GRANT | REVOKE ON SCHEMA行为，赋予或回收指定的用户或角色对模式层级对象的访问权限。
	• 设置此配置项时：GRANT | REVOKE ON DATABASE语法保留非M-Compatibility模式数据库（即GaussDB原有）默认兼容性库中GRANT | REVOKE ON DATABASE语法的行为，赋予或回收指定的用户或角色对数据库层级对象的访问权限。
	-- 创建用户和数据库。
	\c postgres
	DROP USER IF EXISTS testUser cascade;
	DROP DATABASE IF EXISTS m_db;
	CREATE USER testUser WITH PASSWORD '********';
	CREATE DATABASE m_db DBCOMPATIBILITY='M';
	\c m_db
	
	-- 未打开选项，映射成GRANT SCHEMA，提示找不到Schema。
	m_db=# SET m_format_behavior_compat_options='';
	SET
	m_db=# GRANT CREATE ON DATABASE m_db TO testUser;
	ERROR: schema "m_db" does not exist
	m_db=# SET SESSION AUTHORIZATION testUser PASSWORD '********';
	SET
	m_db=> CREATE SCHEMA s1;
	ERROR: Permission denied for database m_db.
	DETAIL: N/A.
	m_db=> \c -
	
	-- 打开选项，成功授予CREATE SCHEMA权限。
	m_db=# SET m_format_behavior_compat_options='grant_database_nomapping';
	SET
	m_db=# GRANT CREATE ON DATABASE m_db TO testUser;
	GRANT
	m_db=# SET SESSION AUTHORIZATION testUser PASSWORD '********';
	SET
	m_db=> CREATE SCHEMA s1;
	CREATE SCHEMA
	m_db=> \c -
	
	-- 清理。
	\c postgres
	DROP USER IF EXISTS testUser cascade;
	DROP DATABASE IF EXISTS m_db;

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

system_function_version

参数说明：数据库系统函数行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前只支持表6-31，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开，例如：“SET system_function_version='get_hash_value_fix,pbe_trunc_fix,empty_array_fix';”。

默认值：

- 安装时：get_hash_value_fix,pbe_trunc_fix,empty_array_fix,regex_like_m_fix,parse_phrase_fix
- 升级时：“”

表 6-31 兼容性配置项

兼容性配置项	兼容性行为控制

get_hash_value_fix	dbe_utility.get_hash_value前向兼容开关。
	• 若不设置此配置项，对于用户输入的字符串会忽略后4个字节，导致小于4个字节的字符串的hash值一致。
	• 若设置此配置项，修复该问题，保证hash值的正确性。
	gaussdb=# SET system_function_version = '';
	gaussdb=# SELECT dbe_utility.get_hash_value(1, 1024, 1024);
	get_hash_value
	---------------- 	
	1444
	(1 row)
	
	gaussdb=# SELECT dbe_utility.get_hash_value(2, 1024, 1024);
	get_hash_value
	---------------- 	
	1444
	(1 row)
	
	gaussdb=# SET system_function_version = 'get_hash_value_fix';
	gaussdb=# SELECT dbe_utility.get_hash_value(1, 1024, 1024);
	get_hash_value
	---------------- 	
	1785
	(1 row)
	
	gaussdb=# SELECT dbe_utility.get_hash_value(2, 1024, 1024);
	get_hash_value
	---------------- 	
	1538
	(1 row)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

pbe_trunc_fix	pbe场景时lpad/rpad/chr/round/trunc/add_months/instrb函数前向兼容开关。
	● 若不设置此配置项，在pbe场景时上述函数的入参小数点会按照四舍五入转换整型。
	● 若设置此配置项，在pbe场景时上述函数的入参小数点会按照整数截断转换整型，保证与ORA结果一致。
	gaussdb=# SET a_format_version='10c';
	gaussdb=# SET a_format_dev_version='s1';
	gaussdb=# PREPARE pround1 AS SELECT round(123.456,$1);
	gaussdb=# EXECUTE pround1('2.6');
	round
	--------- 	
	123.456
	(1 row)
	
	gaussdb=# SET system_function_version = 'pbe_trunc_fix';
	gaussdb=# PREPARE pround1 AS SELECT round(123.456,$1);
	gaussdb=# EXECUTE pround1('2.6');
	round
	--------- 	
	123.46
	(1 row)
	说明
	该参数仅在数据库兼容性为ORA时有效。

empty_array_fix	控制空数组的前向兼容开关。 ● 若不设置此配置项，类似array_positions函数等操作构建的空数组与'{}'::int[]构建的空数组不相等。 ● 若设置此配置项，类似array_positions函数等操作构建的空数组与'{}'::int[]构建的空数组相等。 gaussdb=# SET system_function_version = 'empty_array_fix'; SET gaussdb=# SELECT array_eq(array_positions(array[1, 1, 1], 3), '{}'::int[]); array_eq ---------- t (1 row) gaussdb=# SET system_function_version = ''; SET gaussdb=# SELECT array_eq(array_positions(array[1, 1, 1], 3), '{}'::int[]); array_eq ---------- f (1 row)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
regex_like_m_fix	控制regex_like_m函数行为的前向兼容开关。（regexp_like函数的实现中调用了regex_like_m，所以也受影响。） • 若不设置此配置项，在多行匹配且正则规则中有‘|’时，匹配结果错误。 • 若设置此配置项，在多行匹配且正则规则中有‘|’时，匹配结果正确。 gaussdb=# SET system_function_version = 'regex_like_m_fix'; SET gaussdb=# SELECT 'abc' FROM dual WHERE regex_like_m('111 2 222','^1+$\|^2+$'); ?column? ---------- abc (1 row) gaussdb=# SET system_function_version = ''; SET gaussdb=# SELECT 'abc' FROM dual WHERE regex_like_m('111 2 222','^1+$\|^2+$'); ?column? ---------- (0 row) **说明** 该配置项仅对从匹配结果错误的版本升级上来的数据库生效。对于新安装的数据库，无论是否设置该配置项，行为均与设置该配置项时的行为一致。
parse_phrase_fix	tsquery中控制是否解析词组操作符的开关。 • 若不设置此配置项，to_tsquery等涉及解析tsquery的操作不会解析词组操作符。 • 若设置此配置项，to_tsquery等涉及解析tsquery的操作将解析词组操作符。 gaussdb=# SET system_function_version = 'parse_phrase_fix'; SET gaussdb=# SELECT to_tsquery('hello <-> world'); to_tsquery ---------- 'hello' <-> 'world' (1 row) gaussdb=# SET system_function_version = ''; SET gaussdb=# SELECT to_tsquery('hello <-> world'); ERROR: syntax error in tsquery: "hello <-> world" CONTEXT: referenced column: to_tsquery

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。升级时默认为空，若客户需要打开设置时，建议修改GUC配置文件，保证数据库重启时不会丢失设置项。

设置建议：控制一些系统函数兼容性特性是否可用，若要更改，请在理解参数含义后谨慎操作。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

behavior_compat_options

参数说明：数据库兼容性行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前只支持表6-32，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开，例如：set behavior_compat_options='end_month_calculate,display_leading_zero';。

默认值：

- 新安装的数据库："enable_bpcharlikebpchar_compare,enable_crosstype_integer_operator,forbid_skip_tableof_empty_str_elem,enable_ora_joinop_in_updatestmt"。
- 从历史版本升级上来的数据库，若历史版本不支持该GUC参数，升级后的默认值为""；若历史版本支持该GUC参数，升级后默认值保留原设置。

说明

在参数a_format_version值为10c和a_format_dev_version值为s7的情况下，支持部分兼容性配置项的行为，具体参数介绍请参见a_format_dev_version。

表 6-32 兼容性配置项

兼容性配置项	兼容性行为控制

display_leading_zero	浮点数显示配置项。控制数值类型中char、character、nchar、varchar、character varying、varchar2、nvarchar2、text、clob等所有字符串类型和float4、float8、numeric等任意精度类型的小数点前0的显示，length计算数字长度时是否包含小数点前的0。
	• 不设置此配置项时，对于-1~0和0~1之间的小数，不显示小数点前的0。比如：
	gaussdb=# SELECT 0.1231243 as a, 0.1231243::numeric as b,0.1231243::integer(10,3) as c, length(0.1242343) as d;
	a | b | c | d
	----------+----------+------+---
	.1231243 | .1231243 | .123 | 8
	(1 row)
	• 设置此配置项时，对于-1~0和0~1之间的小数，显示小数点前的0。比如：
	gaussdb=# select 0.1231243 as a, 0.1231243::numeric as b,0.1231243::integer(10,3) as c, length(0.1242343) as d;
	a | b | c | d
	-----------+-----------+-------+---
	0.1231243 | 0.1231243 | 0.123 | 9
	(1 row)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
end_month_calculate	add_months函数计算逻辑配置项。 此参数在M-Compatibility模式数据库下不生效。 假定函数add_months的两个参数分别为param1和param2，param1的月份和param2的和为月份result。 ● 不设置此配置项时，如果param1的日期（Day字段）为月末，并且param1的日期（Day字段）比result月份的月末日期小，计算结果中的日期字段（Day字段）和param1的日期字段保持一致。比如： gaussdb=# select add_months('2018-02-28',3) from *sys_dummy*; add_months --------------------- 2018-05-28 00:00:00 (1 row) ● 设置此配置项时，如果param1的日期（Day字段）为月末，并且param1的日期（Day字段）比result月份的月末日期比小，计算结果中的日期字段（Day字段）和result的月末日期保持一致。比如： gaussdb=# select add_months('2018-02-28',3) from *sys_dummy*; add_months --------------------- 2018-05-31 00:00:00 (1 row)
compat_analyze_sample	analyze采样行为配置项。 设置此配置项时，会优化analyze的采样行为，主要体现在analyze时全局采样会更精确的控制在三万条左右，更好地控制analyze时Coordinator端的内存消耗，保证analyze性能的稳定性。
bind_schema_tablespace	绑定模式与同名表空间配置项。 如果存在与模式名schema_name相同的表空间名，那么如果设置search_path为schema_name，default_tablespace也会同步切换到schema_name。
bind_procedure_searchpath	未指定模式名的存储过程中的数据库对象的搜索路径配置项。 在存储过程中如果不显式指定模式名，会优先在存储过程所属的模式下搜索。 如果找不到，则有两种情况： ● 若不设置此参数，报错退出。 ● 若设置此参数，按照search_path中指定的顺序继续搜索。如果还是找不到，报错退出。
skip_insert_gs_source	开启此参数后，创建PL/SQL对象时不再插入dbe_pldeveloper.gs_source表中。
a_hash_bpchar	开启此参数后，hashbpchar函数会保留字符串末尾空格来计算哈希值。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
correct_to_number	控制to_number()结果兼容性的配置项。 ● 若不设置此配置项，则to_number()函数结果默认与ORA数据库保持一致。 gaussdb=# select '' AS to_number_14, to_number('34,50','999,99'); ERROR: invalid data. CONTEXT: referenced column: to_number ● 若设置此配置项，则to_number()函数结果与pg11保持一致。 gaussdb=# select '' AS to_number_14, to_number('34,50','999,99'); to_number_14 | to_number ---+--- | 3450 (1 row)
unbind_divide_bound	控制对整数除法的结果进行范围校验。 此参数在M-Compatibility模式数据库下不生效。 ● 若不设置此配置项，则会对除法结果做范围校验，例如，INT_MIN/(-1)会因为输出结果大于INT_MAX而报越界错误。 gaussdb=# select (-2147483648)::int4 / (-1)::int4; ERROR: integer out of range ● 若设置此配置项，则不需要对除法结果做范围校验，例如，INT_MIN/(-1)可以得到输出结果为INT_MAX+1。 gaussdb=# select (-2147483648)::int4 / (-1)::int4; ?column? --- 2147483648 (1 row)

convert_string_digit_to_numeric	控制当表中以字符串形式表示的numeric常量和数字类型做比较时，是否都统一转换为numeric类型再进行比较。
	此参数在M-Compatibility模式数据库下不生效。
	● 若不设置此配置项，不会将以字符串形式表示的numeric常量转换为numeric类型。
	● 若设置此配置项，会将以字符串形式表示的numeric常量转换为numeric类型。
	gaussdb=# create table test1 (c1 int, c2 varchar);
	gaussdb=# insert into test1 values (2, '1.1');
	gaussdb=# set behavior_compat_options='';
	gaussdb=# select * from test1 where c2 > 1;
	ERROR: invalid input syntax for type bigint: "1.1"
	
	gaussdb=# set behavior_compat_options='convert_string_digit_to_numeric';
	gaussdb=# select * from test1 where c2 > 1;
	c1 | c2
	---+---
	2 | 1.1
	(1 row)
	
	gaussdb=# drop table test1;

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
return_null_string	控制函数lpad()和rpad()结果为空字符串''的显示配置项。 此参数在M-Compatibility模式数据库下不生效。 ● 不设置此配置项时，空字符串显示为NULL。 gaussdb=# select length(lpad('123',0,'*')) from *sys_dummy*; length -------- (1 row) ● 设置此配置项时，空字符串显示为''。 gaussdb=# select length(lpad('123',0,'*')) from *sys_dummy*; length -------- 0 (1 row)
compat_concat_variadic	控制函数concat()和concat_ws()对variadic类型结果兼容性的配置项。由于MYSQL数据库无variadic类型，所以该选项对MYSQL数据库无影响。 ● 若不设置此配置项，当concat函数参数为variadic类型时，默认ORA数据库和TD数据库兼容模式下结果相同，且与ORA数据库保持一致。 gaussdb=# select concat(variadic NULL::int[]) is null; ?column? ---------- t (1 row) ● 若设置此配置项，当concat函数参数为variadic类型时，保留ORA数据库和TD数据库兼容模式下不同的结果形式。 --ORA数据库下： gaussdb=# select concat(variadic NULL::int[]) is null; ?column? ---------- t (1 row) --TD数据库下： gaussdb=# select concat(variadic NULL::int[]) is null; ?column? ---------- f (1 row)
merge_update_multi	控制在使用MERGE INTO ... WHEN MATCHED THEN UPDATE（请参见**MERGE INTO**）和INSERT ... ON DUPLICATE KEY UPDATE（请参见**INSERT**）时，当目标表中一条目标数据与多条源数据冲突时UPDATE行为。 ● 若设置此配置项，当存在上述场景时，该冲突行将会多次执行UPDATE。 ● 若不设置此配置项（默认），则报错，即MERGE或INSERT操作失败。
plstmt_implicit_savepoint	控制存储过程中更新语句的执行是否拥有独立的子事务。 若设置此配置项，存储过程中每条更新语句前开启隐式保存点，EXCEPTION块中默认回退到最近的保存点，从而保证只回退失败语句的修改。该选项是为了兼容ORA数据库的EXCEPTION行为。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
hide_tailing_zero	numeric显示配置项。 ● 若设置此配置项，所有输出numeric的场景均隐藏小数点后的末尾0，包括指定format精度情况。 ● 若不设置此配置项，numeric按照指定精度显示。 例如： gaussdb=# set behavior_compat_options='hide_tailing_zero'; gaussdb=# select cast(123.123 as numeric(15,10)) as a, to_char(cast(123.123 as numeric(15,10)), '999D999999'); a | to_char ---------+---------- 123.123 | 123.123 (1 row) gaussdb=# set behavior_compat_options=''; gaussdb=# select cast(123.123 as numeric(15,10)) as a, to_char(cast(123.123 as numeric(15,10)), '999D999999'); a | to_char ----------------+------------- 123.1230000000 | 123.123000 (1 row)
truncate_numeric_tail_zero	numeric显示配置项。 ● 若设置此配置项，除to_char(numeric, format)这种显式设置精度的情况外，所有输出numeric的场景均会隐藏小数点后的末尾0。 ● 若不设置此配置项，numeric按照默认精度显示。 例如： gaussdb=# set behavior_compat_options='truncate_numeric_tail_zero'; gaussdb=# select cast(123.123 as numeric(15,10)) as a, to_char(cast(123.123 as numeric(15,10)), '999D999999'); a | to_char ---------+------------- 123.123 | 123.123000 (1 row) gaussdb=# set behavior_compat_options=''; gaussdb=# select cast(123.123 as numeric(15,10)) as a, to_char(cast(123.123 as numeric(15,10)), '999D999999'); a | to_char ----------------+------------- 123.1230000000 | 123.123000 (1 row)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

char_coerce_compat	控制char(n)类型向其它变长字符串类型转换时的行为。该参数仅在sql_compatibility参数的值为ORA时生效，并且开启该参数后无论是隐式转换、显式转换还是通过调用text(bpchar)函数转换类型都不再省略尾部空格。
	● 若设置此配置项，转换时不再省略尾部的空格，并且如果char(n)类型的长度超过其它变长字符串类型，将会报错。
	● 若不设置此配置项，char(n)类型转换其它变长字符串类型时，会省略尾部的空格。
	gaussdb=# set behavior_compat_options='';
	gaussdb=# create table tab_1(col1 varchar(3));
	gaussdb=# create table tab_2(col2 char(3));
	gaussdb=# insert into tab_2 values(' ');
	gaussdb=# insert into tab_1 select col2 from tab_2;
	gaussdb=# select * from tab_1 where col1 is null;
	col1
	------ 	
	
	(1 row)
	gaussdb=# select * from tab_1 where col1=' ';
	col1
	------ 	
	(0 rows)
	gaussdb=# delete from tab_1;
	gaussdb=# set behavior_compat_options = 'char_coerce_compat';
	gaussdb=# insert into tab_1 select col2 from tab_2;
	gaussdb=# select * from tab_1 where col1 is null;
	col1
	------ 	
	(0 rows)
	gaussdb=# select * from tab_1 where col1=' ';
	col1
	------ 	
	
	(1 row)
	
	gaussdb=# drop table tab_1;
	gaussdb=# drop table tab_2;

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
aformat_regexp_match	控制正则表达式函数的匹配行为。 设置此项，且sql_compatibility参数的值为ORA或MYSQL时，正则表达式的flags参数支持的选项含义： 1. 默认不能匹配'\n'字符。 2. flags中包含n选项时，'.'能够匹配'\n'字符。 3. regexp_replace(source, pattern replacement)函数替换所有匹配的子串。 4. regexp_replace(source, pattern, replacement, flags)函数在 flags值为null时，返回值为null。 5. 当sql_compatibility参数值为ORA，regexp_replace(source, pattern, replacement, flags)函数flags值为''时，返回值为null。 否则，正则表达式的flags参数支持的选项含义： 1. 默认能匹配'\n'字符。 2. flags中的n选项表示按照多行模式匹配。 3. regexp_replace(source, pattern replacement)函数仅替换第一个匹配到的子串。 4. regexp_replace(source, pattern, replacement, flags)函数在flags值为null时，返回值为替换后的字符串。 5. 当sql_compatibility参数值为ORA，regexp_replace(source, pattern, replacement, flags)函数flags值为''时，返回值为替换后的字符串。 **须知** 不建议在执行业务期间变更该参数的状态。该参数的状态变更可能会使得相同入参的regexp_replace函数的返回结果发生变化，如果regexp_replace函数在表达式索引中被引用，该参数的状态变更可能会导致数据异常。
disable_emptystr2null	ORA兼容模式绑参场景中，开启此参数后，关闭默认将字符类型传参的值从空串转换为null功能。包括text、clob、blob、raw、bytea、varchar、nvarchar2、bpchar、char、name、byteawithoutorderwithqualcol、byteawithoutordercol类型。该参数为逃生参数，非必要用户不要自行设置。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

select_into_return_null	PG兼容模式下，开启此参数后，存储过程语句SELECT *select_expressions* INTO [STRICT] *target* FROM ... 允许在不指定STRICT并且查询结果为空时给变量赋NULL值。
	开启此参数后，对于返回结果集为空的场景，对变量赋NULL值。比如：
	gaussdb=# SET behavior_compat_options = 'select_into_return_null';
	SET
	gaussdb=# CREATE TABLE t1 (a int, b int) DISTRIBUTE BY hash(a);
	CREATE TABLE
	gaussdb=# DECLARE
	gaussdb-# var int;
	gaussdb-# BEGIN
	gaussdb$# SELECT b INTO var FROM t1;
	gaussdb$# RAISE INFO '%', var;
	gaussdb$# END;
	gaussdb$# /
	INFO: <NULL>
	ANONYMOUS BLOCK EXECUTE
	gaussdb=# DROP TABLE t1;
	DROP TABLE

plsql_security_definer	• 开启此参数后，创建存储过程时默认为定义者权限。 • package默认为SECURITY INVOKER权限，如果想将默认行为改为SECURITY DEFINER权限，需要设置此参数。
proc_outparam_override	• 控制存储过程出参的重载行为，打开该参数后，对于存储过程只有out出参部分不同的情况，也可以正常创建和调用。目前只有gsql与jdbc连接数据库时可以使用该参数，对于其他工具打开该参数连接数据库时无法正常调用带有out的存储过程。 • 支持带有out出参的函数返回record类型，且out出参正常赋值。 • behavior_compat_options='proc_outparam_override' 重载参数仅在ORA兼容模式可用。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

proc_uncheck_default_param	函数调用时不检查默认参数省略情况配置项。
	● 若不设置此配置项，调用带有默认参数的函数时，入参从左往右排入函数，如果有非默认参数的入参缺失则会报错。比如：
	gaussdb=# create or replace function test(f1 int, f2 int default 20, f3 int, f4 int default 40, f5 int default 50) return int
	gaussdb-# as
	gaussdb$# begin
	gaussdb$# raise info 'f1:%',f1;
	gaussdb$# raise info 'f2:%',f2;
	gaussdb$# raise info 'f3:%',f3;
	gaussdb$# raise info 'f4:%',f4;
	gaussdb$# raise info 'f5:%',f5;
	gaussdb$# return 1;
	gaussdb$# end;
	gaussdb$# /
	CREATE FUNCTION
	gaussdb=# select test(1,2);
	ERROR: function test(integer, integer) does not exist
	LINE 1: select test(1,2);
	^
	HINT: No function matches the given name and argument types. You might need to add explicit type casts.
	CONTEXT: referenced column: test
	
	gaussdb=# drop function test;
	● 若设置此配置项，调用带有默认参数的函数时，入参从左往右排入函数，允许缺省默认参数的入参，如果有非默认参数的入参缺失，则会用错位的默认值填充该参数。比如：
	gaussdb=# create or replace function test(f1 int, f2 int default 20, f3 int, f4 int default 40, f5 int default 50) return int
	gaussdb-# as
	gaussdb$# begin
	gaussdb$# raise info 'f1:%',f1;
	gaussdb$# raise info 'f2:%',f2;
	gaussdb$# raise info 'f3:%',f3;
	gaussdb$# raise info 'f4:%',f4;
	gaussdb$# raise info 'f5:%',f5;
	gaussdb$# return 1;
	gaussdb$# end;
	gaussdb$# /
	CREATE FUNCTION
	gaussdb=# select test(1,2);
	INFO: f1:1
	CONTEXT: referenced column: test
	INFO: f2:2
	CONTEXT: referenced column: test
	INFO: f3:20
	CONTEXT: referenced column: test
	INFO: f4:40
	CONTEXT: referenced column: test
	INFO: f5:50
	CONTEXT: referenced column: test
	test
	------ 	
	1
	(1 row)
	
	gaussdb=# drop function test;
	如上，f3被错误的默认值填充。
	说明
	该场景下，非默认参数会被错位的默认值填充。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
dynamic_sql_compat	开启此参数后： ● 动态语句不会将模板SQL中的同名模板参数视为同一个变量，而是按照顺序依次匹配using子句中的变量值。 ● 动态语句执行语句时调用存储过程的场景，会对存储过程中参数的IN/OUT属性和using子句中的IN/OUT属性进行检查。 ● 应用场景：Execute immediate、open cursor for、return query、for var in execute sql绑定参数场景。 **说明** 动态语句执行匿名块语句时调用存储过程的场景，只针对IN参数进行矫正，如果需要对OUT参数进行检查需要设置proc_outparam_override选项。
dynamic_sql_check	开启此参数后，动态语句模板SQL中的不同模板参数个数与using子句中的变量个数不同时，将会在动态语句执行期间报错。 应用场景：Execute immediate、open cursor for、return query、for var in execute sql绑定参数场景。 **说明** ● 不推荐同时使用dynamic_sql_check选项和dynamic_sql_compat选项，开启dynamic_sql_compat选项后dynamic_sql_check选项将不再生效。 ● 动态语句执行匿名块语句时调用存储过程的场景，只针对IN参数进行检查，如果需要对OUT参数进行检查需要设置proc_outparam_override选项。 ● 动态语句执行匿名块语句时调用存储过程的场景，开启参数后不会对存储过程中参数的IN/OUT属性和using子句中的IN/OUT属性进行检查。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

enable_funcname_with_argsname	开启参数后，使用select调用函数时投影别名显示完整函数。
	● 若不设置此配置项，使用select调用函数时投影别名仅显示函数名。比如：
	gaussdb=# select power(2,3);
	power
	------- 	
	8
	(1 row)
	
	gaussdb=# select count(*) from db_ind_columns;
	count
	------- 	
	611
	(1 row)
	
	gaussdb=# select count(index_name) from db_ind_columns;
	count
	------- 	
	611
	(1 row)
	
	gaussdb=# select left('abcde', 2);
	left
	------ 	
	ab
	(1 row)
	
	gaussdb=# select pg_client_encoding();
	pg_client_encoding
	-------------------- 	
	UTF8
	(1 row)
	● 若设置此配置项，使用select调用函数时投影别名显示完整函数。比如：
	gaussdb=# set behavior_compat_options =
	'enable_funcname_with_argsname';
	SET
	gaussdb=# select power(2,3);
	power(2,3)
	------------ 	
	8
	(1 row)
	
	gaussdb=# select count(*) from db_ind_columns;
	count(*)
	---------- 	
	611
	(1 row)
	
	gaussdb=# select count(index_name) from db_ind_columns;
	count(index_name)
	------------------- 	
	611
	(1 row)
	
	gaussdb=# select left('abcde', 2);
	left('abcde',2)
	----------------- 	
	ab
	(1 row)
	
	gaussdb=# select pg_client_encoding();
	pg_client_encoding()

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	---------- 	
	UTF8
	(1 row)
	• 目前仅支持func_name(args_list)、func_name()、func_name(*)三种形式投影别名显示完整函数，且参数args类型仅支持字符型常量、数值型常量、列名、函数。列名支持带有表名，但是func_name(*)不支持*前面带有表名。函数名支持带有schema、包名。不支持参数为表达式、仅支持参数带有DISTINCT关键字，不支持带有其他关键字时显示完整函数。
	• 函数入参为以下场景时，投影别名不支持显示完整函数。
	-- 创建ORA兼容模式数据库。
	gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA' ENCODING = 'UTF8';
	CREATE DATABASE
	gaussdb=# \c ora_db
	ora_db=# SET behavior_compat_options = 'enable_funcname_with_argsname';
	SET
	-- 函数入参为null值或者ORA兼容性下的空字符串。
	ora_db=# SELECT length(null);
	length
	-------- 	
	
	(1 row)
	
	ora_db=# SHOW sql_compatibility;
	sql_compatibility
	------------------- 	
	ORA
	(1 row)
	
	ora_db=# SELECT length('');
	length
	-------- 	
	
	(1 row)
	
	-- 函数入参为带双冒号的类型强制转换。
	ora_db=# SELECT length('abc'::text);
	length
	-------- 	
	3
	(1 row)
	
	-- 函数入参为指定类型转换。
	ora_db=# SELECT length(text'abc');
	length
	-------- 	
	3
	(1 row)
	
	-- 函数入参为pbe占位符。
	ora_db=# PREPARE p1 AS SELECT length($1);
	PREPARE
	ora_db=# EXECUTE p1('abc');
	length
	-------- 	
	3
	(1 row)
	
	ora_db=# DEALLOCATE p1;
	DEALLOCATE

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	-- 函数入参为数组构造器。
	ora_db=# SELECT array_ndims(ARRAY[1, 2, 3]);
	array_ndims
	------------- 	
	1
	(1 row)
	• 函数带不同子句时，投影别名的差异如下：
	– 函数入参带有其他子句（如ORDER BY子句）时，投影别名仅支持显示函数名。
	– 函数后接OVER、WITHIN子句时，投影别名支持显示函数名和入参，不支持显示OVER、WITHIN子句。
	– 函数后接KEEP子句时，投影别名支持显示函数名，不支持显示入参和KEEP子句。
	gaussdb=# \c ora_db
	ora_db=# SET behavior_compat_options =
	'enable_funcname_with_argsname';
	SET
	ora_db=# DROP TABLE IF EXISTS tbl;
	ora_db=# CREATE TABLE tbl (id int, lower_col text, col1 int, col2
	numeric);
	ora_db=# INSERT INTO tbl VALUES (1, 'hello', 42, 3.14);
	
	-- 入参带ORDER BY子句
	ora_db=# SELECT string_agg(lower_col, '1' ORDER BY id) FROM tbl;
	string_agg
	------------ 	
	hello
	(1 row)
	
	-- OVER子句
	ora_db=# SELECT min(col1) OVER(PARTITION BY id ORDER BY col2)
	FROM tbl;
	min(col1)
	----------- 	
	42
	(1 row)
	
	-- WITHIN子句
	ora_db=# SELECT listagg(lower_col, ',') WITHIN GROUP(ORDER BY id)
	FROM tbl;
	listagg(lower_col,',')
	------------------------ 	
	hello,world,test
	(1 row)
	
	-- KEEP子句
	ora_db=# SELECT min(col1) KEEP (DENSE_RANK FIRST ORDER BY
	col2) FROM tbl;
	min
	----- 	
	42
	(1 row)
	
	ora_db=# DROP TABLE IF EXISTS tbl;
	• 部分函数既支持语法函数调用形式，又支持系统函数调用形式，不同调用方式会导致投影别名显示差异，函数列表参见**表1-29**。
	– 仅使用函数名称调用时，投影别名不显示完整函数。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	– 使用pg_catalog前缀或者双引号包裹调用函数时，投影别名显示完整函数。
	• 投影列包含纯语法函数时，不支持完整投影别名显示，函数列表参见**表1-29**。纯语法函数的投影列名不一定显示函数名本身，而是显示该函数在语法解析阶段推导出的输出标识。
	– 当最外层函数为纯语法函数时，投影列名仅显示该函数的名称。
	– 若外层函数支持投影别名显示，但其参数中嵌套了纯语法函数，则仅显示外层函数名，而不显示纯语法函数名。
	• 一些安全加解密函数、脱敏函数，投影别名显示完整函数可能存在安全问题，在这里就还是仅显示函数名。包括函数：gs_encrypt_aes128、gs_decrypt_aes128、gs_encrypt、gs_decrypt、gs_encrypt_bytea、gs_decrypt_bytea、aes_encrypt、aes_decrypt、pg_create_physical_replication_slot_extern、creditcardmasking、basicemailmasking、fullemailmasking、alldigitsmasking、shufflemasking、randommasking、regexpmasking、gs_digest、desencrypt、desdecrypt、des3encrypt、des3decrypt。
	• 不支持=>传参方式调用函数时投影别名显示完整函数，不支持投影别名显示""，如：select "power"(2,3)。
	• 为了让投影别名显示完整函数，本功能不受去除末尾0等参数影响。
	• 投影别名超过63个字节时，会截断后面部分。
	• 参数args为数值型常量时，不支持投影别名显示前导0。
	`gaussdb=# SET behavior_compat_options = 'enable_funcname_with_argsname';`
	`SET`
	`gaussdb=# SELECT abs(0010);`
	` abs(10)`
	`---------`
	`  10`
	`(1 row)`
	• 在如下特殊场景中，投影别名有所差异：
	– 场景一：部分函数既是语法函数又是系统函数，函数列表参见**表1-29**，在支持不带括号调用的场景下，使用pg_catalog前缀调用时，不会显示pg_catalog前缀。
	– 场景二：xmltype在带pg_catalog前缀或者双引号包裹调用时，会被当作系统函数调用，投影别名支持显示完整函数；仅使用xmltype调用时，会被当作语法函数调用，投影别名不支持显示完整函数。
	`gaussdb=# \c ora_db`
	`ora_db=# SET behavior_compat_options = 'enable_funcname_with_argsname';`
	`SET`
	`-- 场景一（当前用户为omm）：`
	`ora_db=# SELECT pg_catalog.current_user;`
	` current_user`
	`--------------`

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	omm
	(1 row)
	
	-- 场景二：
	ora_db=# SELECT pg_catalog.xmltype('<a>1');
	pg_catalog.xmltype('<a>1')
	--- 	
	<a>1
	(1 row)
	
	ora_db=# SELECT "xmltype"('<a>1');
	xmltype('<a>1')
	--- 	
	<a>1
	(1 row)
	
	ora_db=# SELECT xmltype('<a>1');
	xmltype
	--- 	
	<a>1
	(1 row)
	
	ora_db=# \c postgres
	gaussdb=# DROP DATABASE ora_db;

proc_outparam_transfer_length	开启此参数后，存储过程和函数支持out出参长度传递，会在内层存储过程或者函数中报错。例如：
	gaussdb=# SET behavior_compat_options='proc_outparam_override,proc_outparam_transfer_length';
	SET
	gaussdb=# CREATE OR REPLACE PROCEDURE out_param_test1(m in int, v inout varchar2,v1 inout varchar2) is
	gaussdb$# begin
	gaussdb$# v := 'aaaddd';
	gaussdb$# v1 := 'aaaddd';
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# CREATE OR REPLACE PROCEDURE call_out_param_test1 is
	gaussdb$# v varchar2(5) := 'aabbb';
	gaussdb$# v1 varchar2(6) := 'aabb';
	gaussdb$# begin
	gaussdb$# out_param_test1(5,v,v1);
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# CALL call_out_param_test1();
	ERROR: value too long for type character varying(5)
	CONTEXT: PL/SQL function out_param_test1(integer,character varying,character varying) line 3 at assignment
	PL/SQL function call_out_param_test1() line 4 at SQL statement
	
	gaussdb=# DROP PROCEDURE call_out_param_test1;
	gaussdb=# DROP PROCEDURE out_param_test1;

compat_cursor	控制错误码变量兼容行为。设置此项，且兼容ORA，错误码变量（SQLCODE、SQLSTATE）由原先的仅在当前执行的函数有效，拓展到包括本函数调用的子函数有效。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

tableof_elem_constraints	开启此参数后：
	1. 集合类型会对元素合法性进行校验。
	2. 索引为varchar的集合类型，索引长度也会进行合法性校验。
	行为差异详情和示例具体可参考**集合**。
	说明
	• 不要在同session内切换并使用该参数。
	• 对于升级新版本的数据库：
	  • 已经在存储过程或PACKAGE内定义好的集合类型，开启tableof_elem_constraints参数不会使上述功能生效。需要重新创建集合类型，即需要重新创建存储过程或PACKAGE，执行CREATE OR REPLACE或DROP + CREATE操作来重新创建存储过程或PACKAGE从而重新创建集合类型后该参数才能生效。
	  • 通过CREATE TYPE type_name is table of data_type语法创建出来的集合类型，需要DROP类型后重新创建该功能才能生效。
	元素合法性校验示例：
	未开启此参数时调用如下存储过程，元素不校验。
	`gaussdb=# CREATE OR REPLACE procedure p1 is`
	`gaussdb$#  type t1 is table of varchar(5);`
	`gaussdb$#  v t1 := t1();`
	`gaussdb$# begin`
	`gaussdb$#  v.extend();`
	`gaussdb$#  v(1) := '123456';`
	`gaussdb$#  raise info '%', v;`
	`gaussdb$# end;`
	`gaussdb$# /`
	`CREATE PROCEDURE`
	`gaussdb=# CALL p1();`
	`INFO:  {123456}`
	` p1`
	`----`
	`(1 row)`
	`gaussdb=# DROP procedure p1;`
	开启此参数后调用如下存储过程，元素校验会报错。
	`gaussdb=# SET behavior_compat_options = 'tableof_elem_constraints';`
	`SET`
	`gaussdb=# CREATE OR REPLACE procedure p1 is`
	`gaussdb$#  type t1 is table of varchar(5);`
	`gaussdb$#  v t1 := t1();`
	`gaussdb$# begin`
	`gaussdb$#  v.extend();`
	`gaussdb$#  v(1) := '123456';`
	`gaussdb$#  raise info '%', v;`
	`gaussdb$# end;`
	`gaussdb$# /`
	`CREATE PROCEDURE`
	`gaussdb=# CALL p1();`
	`ERROR:  value too long for type character varying(5)`
	`CONTEXT:  PL/SQL function p1() line 5 at assignment`
	索引值合法性校验示例：
	未开启该参数时，索引值超过定义长度不报错。
	`gaussdb=# CREATE OR REPLACE procedure p1 is`
	`gaussdb$#  type t1 is table of int index by varchar(5);`
	`gaussdb$#  v t1;`

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	gaussdb$# begin
	gaussdb$# v('123456') := 1;
	gaussdb$# raise info '%', v;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# CALL p1();
	INFO: {indexbyType:varchar,"123456"=>1}
	 p1
	---- 	
	
	(1 row)
	
	gaussdb=# DROP procedure p1;
	
	开启该参数后，索引值超过定义长度报错。
	gaussdb=# SET behavior_compat_options = 'tableof_elem_constraints';
	SET
	gaussdb=# CREATE OR REPLACE procedure p1 is
	gaussdb$# type t1 is table of int index by varchar(5);
	gaussdb$# v t1;
	gaussdb$# begin
	gaussdb$# v('123456') := 1;
	gaussdb$# raise info '%', v;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# CALL p1();
	ERROR: value too long for type character varying(5)
	CONTEXT: PL/SQL function p1() line 4 at assignment

allow_rownum_alias	开启此参数后，将允许ROWNUM在SQL语句中通过AS语法用作列的别名，ROWNUM将作为普通标识符使用，不再能够作为伪列使用。 此参数在M-Compatibility模式数据库下不生效。 **说明** 不建议在执行业务期间变更该参数的状态。开启参数的状态下，在数据库中用ROWNUM作为名称创建的数据库对象（如表名，列名，数据库名等），在参数关闭后，将会产生歧义，行为不可预期。关闭参数的状态下，在数据库中ROWNUM作为伪列使用的行为，在参数开启后，将失效，行为不可预期。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

current_sysdate	开启此参数后，执行sysdate时，会获取当前操作系统时间。关闭参数后，sysdate返回当前本条SQL执行时刻的系统日期和时间。
	gaussdb=# set behavior_compat_options='current_sysdate';
	SET
	gaussdb=# select sysdate;
	current_sysdate
	--------------------- 	
	2023-06-20 20:15:27
	(1 row)
	开启GUC参数mapping_date_to_datea后，ORA兼容性下，sysdate会默认返回datea类型，获取当前操作系统时间，current_sysdate不再生效。
	gaussdb=# set behavior_compat_options='';
	SET
	gaussdb=# show mapping_date_to_datea;
	mapping_date_to_datea
	---------------------- 	
	on
	(1 row)
	gaussdb=# set NLS_DATE_FORMAT = 'YYYY-MM-DD HH24:MI:SS';
	SET
	gaussdb=# begin
	gaussdb$# dbe_output.print_line('time1:' || sysdate);
	gaussdb$# pg_sleep(2);
	gaussdb$# dbe_output.print_line('time2:' || sysdate);
	gaussdb$# end;
	gaussdb$# /
	time1:2025-02-27 10:45:57
	time2:2025-02-27 10:45:59

版权所有 © 华为云计算技术有限公司

<table>
<tr><th>兼容性配置项</th><th>兼容性行为控制</th></tr>
<tr><td>allow_function_procedure_replace</td><td>默认场景下，对于ORA和PG兼容模式，当有一个包外的存储过程（或函数）时，创建一个同名的函数（或存储过程），与已有对象类型不同，不支持创建并报错。开启此参数后，通过create or replace的方式可以实现包外同名的存储过程和函数对象之间的类型替换。
<pre>
gaussdb=# create or replace function proc_test return varchar2 as
gaussdb$# begin
gaussdb$# return '1';
gaussdb$# end;
gaussdb$# /
CREATE FUNCTION
gaussdb=# create or replace procedure proc_test as
gaussdb$# begin
gaussdb$# null;
gaussdb$# end;
gaussdb$# /
ERROR: cannot change routine kind
DETAIL: "proc_test" is a function.
gaussdb=# \df+ proc_test
                                                          List of functions
 Schema | Name | Result data type | Argument data types | Type |
Volatility | Owner | Language | Source code | Description | fencedmode |
propackage | prokind
--------+-----------+-------------------+---------------------+--------+------------
+-----------+----------+-------------+-------------+------------+------------+---------
 public | proc_test | character varying | | normal | volatile |
wangxinyu | plpgsql | DECLARE   +| | f | f | f
        | | | | | |
begin   +| | | |
        | | | | | |
return '1';+| | | |
        | | | | | |
end | | | |
(1 row)

-- 设置参数后允许替换类型
gaussdb=# set behavior_compat_options='allow_function_procedure_replace';
SET
gaussdb=# create or replace procedure proc_test as
gaussdb$# begin
gaussdb$# null;
gaussdb$# end;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# \df+ proc_test
                                                          List of functions
 Schema | Name | Result data type | Argument data types | Type |
Volatility | Owner | Language | Source code | Description | fencedmode |
propackage | prokind
--------+-----------+------------------+---------------------+--------+------------
+-----------+----------+-------------+-------------+------------+------------+---------
 public | proc_test | void | | normal | volatile | wangxinyu
| plpgsql | DECLARE   +| | f | f | p
        | | | | | |
begin   +| | | |
        | | | | | |
null;   +| | | |
        | | | | | |
end | | | |
(1 row)

gaussdb=# drop procedure proc_test;
</pre></td></tr>
</table>

兼容性配置项	兼容性行为控制

collection_exception_backcompat	控制在PL/SQL中集合类型相关报错抛出的异常值，当前控制三种异常值，对应关系如下：
	未开启该参数 开启该参数
	collection_is_null program_limit_exceeded
	subscript_beyond_count program_limit_exceeded
	subscript_outside_limit program_limit_exceeded
	用例：
	gaussdb=# create or replace procedure p1 is
	gaussdb$# type t1 is table of int;
	gaussdb$# v t1;
	gaussdb$# v_int int;
	gaussdb$# begin
	gaussdb$# v_int := v.count();
	gaussdb$# exception when collection_is_null then
	gaussdb$# raise info '%', sqlerrm;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# call p1();
	INFO: Reference to uninitialized collection
	p1
	---- 	
	
	(1 row)
	
	gaussdb=# create or replace procedure p1 is
	gaussdb$# type t1 is table of int;
	gaussdb$# v t1 := t1(1, 2, 3);
	gaussdb$# v_int int;
	gaussdb$# begin
	gaussdb$# v_int := v(4);
	gaussdb$# exception when subscript_beyond_count then
	gaussdb$# raise info '%', sqlerrm;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# call p1();
	INFO: Subscript beyond count
	p1
	---- 	
	
	(1 row)
	
	gaussdb=# create or replace procedure p1 is
	gaussdb$# type t1 is table of int;
	gaussdb$# v t1 := t1(1, 2, 3);
	gaussdb$# v_int int;
	gaussdb$# begin
	gaussdb$# v_int := v(-1);
	gaussdb$# exception when subscript_outside_limit then
	gaussdb$# raise info '%', sqlerrm;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# call p1();
	INFO: Subscript outside of limit
	p1
	---- 	
	
	(1 row)
	
	gaussdb=# set behavior_compat_options = 'collection_exception_backcompat';
	SET

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	gaussdb=# create or replace procedure p1 is
	gaussdb$# type t1 is table of int;
	gaussdb$# v t1;
	gaussdb$# v_int int;
	gaussdb$# begin
	gaussdb$# v_int := v.count();
	gaussdb$# exception when program_limit_exceeded then
	gaussdb$# raise info '%', sqlerrm;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# call p1();
	INFO: Reference to uninitialized collection
	 p1
	---- 	
	
	(1 row)
	
	gaussdb=# create or replace procedure p1 is
	gaussdb$# type t1 is table of int;
	gaussdb$# v t1 := t1(1, 2, 3);
	gaussdb$# v_int int;
	gaussdb$# begin
	gaussdb$# v_int := v(4);
	gaussdb$# exception when program_limit_exceeded then
	gaussdb$# raise info '%', sqlerrm;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# call p1();
	INFO: Subscript beyond count
	 p1
	---- 	
	
	(1 row)
	
	gaussdb=# create or replace procedure p1 is
	gaussdb$# type t1 is table of int;
	gaussdb$# v t1 := t1(1, 2, 3);
	gaussdb$# v_int int;
	gaussdb$# begin
	gaussdb$# v_int := v(-1);
	gaussdb$# exception when program_limit_exceeded then
	gaussdb$# raise info '%', sqlerrm;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# call p1();
	INFO: Subscript outside of limit
	 p1
	---- 	
	
	(1 row)
	
	gaussdb=# drop procedure p1;

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

enable_case_when_alias	开启该参数后，case when和decode语法的别名为以__unnamed_开头的字符串。
	用例：
	gaussdb=# set behavior_compat_options='enable_case_when_alias';
	SET
	gaussdb=# create table test(c1 varchar2);
	CREATE TABLE
	gaussdb=# insert into test values('x');
	INSERT 0 1
	gaussdb=# select decode(c1,'x','0','default') from test;
	__unnamed_decode__
	--- 	
	0
	(1 row)
	
	gaussdb=# select (case c1 when 'x' then '0' else 'default' end) from test;
	__unnamed_case_when__
	--- 	
	0
	(1 row)
	
	gaussdb=# drop table test;

plsql_rollback_keep_user	控制在PL/SQL中rollback和rollback to savepoint是否修改当前用户，以及rollback to定义者权限函数内的savepoint是否修改当前用户。当开启此参数时，rollback将不会修改当前用户。
	用例：
	gaussdb=# create user plsql_rollback1 password '********';
	gaussdb=# create user plsql_rollback2 password '********';
	gaussdb=# grant plsql_rollback1 to plsql_rollback2;
	gaussdb=# create or replace procedure plsql_rollback1.p1 () authid definer
	gaussdb-# as
	gaussdb$# va int;
	gaussdb$# begin
	gaussdb$# raise info 'current usr:%', current_user;
	gaussdb$# rollback;
	gaussdb$# raise info 'current usr:%', current_user;
	gaussdb$# end;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# set session AUTHORIZATION plsql_rollback2 PASSWORD '********';
	SET
	gaussdb=> set behavior_compat_options = 'plsql_rollback_keep_user';
	SET
	gaussdb=> call plsql_rollback1.p1 ();
	INFO: current usr:plsql_rollback1
	INFO: current usr:plsql_rollback1
	p1
	---- 	
	
	(1 row)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
enable_bpcharlikebpchar_compare	启用/关闭bpcharlikebpchar和bpcharnlikebpchar操作符。 ● 新安装的数据库默认开启该参数。 ● 505.1.0版本之前的数据库通过升级操作后，该参数默认关闭。

```
gaussdb=# SELECT bpcharlikebpchar('455'::BPCHAR(10), '455  '::BPCHAR);
 bpcharlikebpchar
------------------
 f
(1 row)
gaussdb=# SELECT bpcharlikebpchar('455 '::BPCHAR(10), '455 '::BPCHAR(10));
 bpcharlikebpchar
------------------
 t
(1 row)
gaussdb=# SELECT bpcharlikebpchar('455 '::BPCHAR(10), '455'::BPCHAR(10));
 bpcharlikebpchar
------------------
 t
(1 row)
gaussdb=# SELECT bpcharlikebpchar('455 '::BPCHAR(10), '455'::BPCHAR(11));
 bpcharlikebpchar
------------------
 f
(1 row)
gaussdb=# CREATE TABLE op_test (
  col BPCHAR(2) DEFAULT NULL
 );
CREATE TABLE
gaussdb=# CREATE INDEX op_index ON op_test(col);
CREATE INDEX
gaussdb=# INSERT INTO op_test VALUES ('a');
INSERT 0 1
gaussdb=# INSERT INTO op_test VALUES ('1');
INSERT 0 1
gaussdb=# INSERT INTO op_test VALUES ('11');
INSERT 0 1
gaussdb=# INSERT INTO op_test VALUES ('12');
INSERT 0 1
gaussdb=# INSERT INTO op_test VALUES ('sd');
INSERT 0 1
gaussdb=# INSERT INTO op_test VALUES ('aa');
INSERT 0 1
gaussdb=# SHOW behavior_compat_options;
 behavior_compat_options
-------------------------
(1 row)
--当behavior_compat_options不包含enable_bpcharlikebpchar_compare时，则未启用最新bpcharlikebpchar操作符，其bpchar匹配bpchar返回结果集非预期（正常应返回全部数据）。
gaussdb=# EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col;
             QUERY PLAN
-------------------------------------
 Sort
   Sort Key: col
   ->  Seq Scan on op_test
         Filter: (col ~~ (col)::text)
(4 rows)
gaussdb=# SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col;
 col
-----
 11
 12
```


版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	aa
	sd
	(4 rows)
	gaussdb=# SET behavior_compat_options =
	'enable_bpcharlikebpchar_compare';
	SET
	gaussdb=# SHOW behavior_compat_options;
	behavior_compat_options
	--------------------------------- 	
	enable_bpcharlikebpchar_compare
	(1 row)
	--开启参数后，将启用最新bpcharlikebpchar操作符，其匹配时返回行为符合预期行为。
	gaussdb=# EXPLAIN (COSTS OFF) SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col;
	QUERY PLAN
	------------------------------ 	
	Sort
	Sort Key: col
	-> Seq Scan on op_test
	Filter: (col ~~ col)
	(4 rows)
	gaussdb=# SELECT * FROM op_test WHERE col LIKE col::BPCHAR ORDER BY col;
	col
	----- 	
	1
	11
	12
	a
	aa
	sd
	(6 rows)
	gaussdb=# DROP TABLE op_test;
	DROP TABLE
	说明
	● 该参数仅在sql_compatibility兼容性参数设置为ORA时生效。
	● 启用该参数，会影响BPCHAR类型之间进行LIKE或NOT LIKE模式匹配时的结果集与执行计划。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

cursor_asensitive	打开该参数后，游标数据敏感行为兼容505.1之前版本。默认关闭该参数，数据库实现数据非敏感游标。
	● 若设置此配置项，游标数据敏感行为兼容505.1之前版本，某些游标可以看到OPEN后的修改。
	gaussdb=# set behavior_compat_options='cursor_asensitive';
	SET
	gaussdb=# create table t1(c1 int, c2 varchar,c3 varchar) distribute by hash(c1);
	CREATE TABLE
	gaussdb=# insert into t1 values(1,'Tom','Jones');
	INSERT 0 1
	gaussdb=# create or replace function func1() returns sys_refcursor
	gaussdb-# as $$
	gaussdb$# declare
	gaussdb$# x sys_refcursor;
	gaussdb$# cursor cur1 for select count(1) from t1 where c2 = 'Joe';
	gaussdb$# begin
	gaussdb$# open cur1;
	gaussdb$# update t1 set c2='Joe' where c3='Jones';
	gaussdb$# fetch cur1 into x;
	gaussdb$# close cur1;
	gaussdb$# return x;
	gaussdb$# end;$$
	gaussdb-# LANGUAGE plpgsql;
	CREATE FUNCTION
	gaussdb=# call func1();
	func1
	------- 	
	1
	(1 row)
	
	gaussdb=# drop function func1;
	gaussdb=# drop table t1;
	● 若不设置此配置项，游标OPEN确定游标的结果集，看不到OPEN后的修改。
	gaussdb=# show behavior_compat_options;
	behavior_compat_options
	------------------------ 	
	
	(1 row)
	gaussdb=# create table t1(c1 int, c2 varchar,c3 varchar) distribute by hash(c1);
	CREATE TABLE
	gaussdb=# insert into t1 values(1,'Tom','Jones');
	INSERT 0 1
	gaussdb=# create or replace function func1() returns sys_refcursor
	gaussdb-# as $$
	gaussdb$# declare
	gaussdb$# x sys_refcursor;
	gaussdb$# cursor cur1 for select count(1) from t1 where c2 = 'Joe';
	gaussdb$# begin
	gaussdb$# open cur1;
	gaussdb$# update t1 set c2='Joe' where c3='Jones';
	gaussdb$# fetch cur1 into x;
	gaussdb$# close cur1;
	gaussdb$# return x;
	gaussdb$# end;$$
	gaussdb-# LANGUAGE plpgsql;
	CREATE FUNCTION
	gaussdb=# call func1();
	func1
	------- 	
	0

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	(1 row)
	
	gaussdb=# drop function func1;
	gaussdb=# drop table t1;

 版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

enable_crosstype_integer_operator	启用/关闭跨类型整型操作符。
	● 全新安装的数据库默认开启该参数。
	● 505.1.0版本之前的数据库通过升级操作后，该参数默认关闭。
	● 涉及操作符范围=、<>、<、>、<=、>=。
	● 开启该参数后，涉及的跨类型整型比较，不再需要隐式转换，可直接进行比较。如下以(int1 op int2)举例：
	ora_db=# SET max_datanode_for_plan=1;
	SET
	ora_db=# DROP TABLE IF EXISTS implicit_index;
	DROP TABLE
	ora_db=# CREATE TABLE implicit_index(c1 int1);
	NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by default.
	HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
	CREATE TABLE
	ora_db=# SET behavior_compat_options='enable_crosstype_integer_operator';
	SET
	ora_db=# EXPLAIN (COSTS OFF) SELECT * FROM implicit_index WHERE c1 > 1::int2;
	QUERY PLAN
	--- 	
	Data Node Scan
	Node/s: All datanodes
	
	Remote SQL: SELECT c1 FROM public.implicit_index WHERE c1 > 1::smallint
	Datanode Name: datanode1--该节点信息为当前测试示例执行计划节点信息，可能与用户实际存在差异，具体实际执行展示信息以实际信息为准。
	Seq Scan on implicit_index
	Filter: (c1 > 1::smallint)
	
	(8 rows)
	
	ora_db=# SET behavior_compat_options='';
	SET
	ora_db=# EXPLAIN (COSTS OFF) SELECT * FROM implicit_index WHERE c1 > 1::int2;
	QUERY PLAN
	--- 	
	Data Node Scan
	Node/s: All datanodes
	
	Remote SQL: SELECT c1 FROM public.implicit_index WHERE c1::bigint > 1::smallint
	Datanode Name: datanode1--该节点信息为当前测试示例执行计划节点信息，可能与用户实际存在差异，具体实际执行展示信息以实际信息为准。
	Seq Scan on implicit_index
	Filter: ((c1)::bigint > 1::smallint)
	
	(8 rows)
	
	ora_db=# DROP TABLE implicit_index;
	DROP TABLE

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
	说明 ● 该参数仅在sql_compatibility兼容性参数设置为ORA时生效。 ● 启用该参数，会影响涉及如下场景的操作符的执行计划： ● (int1 op int2) ● (int1 op int4) ● (int1 op int8) ● (int1 op int16) ● (int1 op numeric) ● (int2 op int1) ● (int2 op int16) ● (int2 op numeric) ● (int4 op int1) ● (int4 op int16) ● (int4 op numeric) ● (int8 op int1) ● (int8 op int16) ● (int8 op numeric) ● (int16 op int1) ● (int16 op int2) ● (int16 op int4) ● (int16 op int8) ● (numeric op int1) ● (numeric op int2) ● (numeric op int4) ● (numeric op int8)

forbid_package_function_with_prefix	打开该参数后，创建package内的函数时，如果函数有前缀，则会报错。
	例如：
	gaussdb=# SET behavior_compat_options='forbid_package_function_with_prefix';
	SET
	gaussdb=# CREATE SCHEMA schema1;
	gaussdb=# CREATE OR REPLACE package pck1
	IS
	procedure schema1.proc1();
	END pck1;
	/
	ERROR: not support procedure name use *.* format in package at or near ";"
	CONTEXT: compilation of PL/pgSQL package "pck1" near line 2
	
	gaussdb=# DROP SCHEMA schema1;

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
time_constexpr_compact	打开该参数后，执行时间表达式场景，会根据常量是否含有时区，自动返回with timezone类型或without timezone类型。 目前支持timestamp类型、time类型。 --执行不带时区的时间戳表达式。 gaussdb=# SELECT timestamp '1999-03-15 8:00:00'; timestamp --------------------- 1999-03-15 08:00:00 (1 row) --执行带时区的时间戳表达式 返回timestamp with time zone类型。 gaussdb=# SELECT timestamp '1999-03-15 8:00:00 -8:00:00'; timestamptz ------------------------ 1999-03-16 00:00:00+08 (1 row)
enable_use_ora_timestamptz	开启后TIMESTAMP AT TIME ZONE语法得到的日期时间跟ORA数据库保持一致。由于GaussDB只能显示session时区，ORA可以显示输入的时区，当时区不一致时显示的结果不同，将日期时间转换到相同时区后是同一时间点。 --打开参数前 gaussdb=# SELECT timestamp '2024-03-20 01:30:00' at time zone 'Europe/Moscow' FROM dual; timezone --------------------- 2024-03-20 06:30:00+08 (1 row) -- 打开参数后 gaussdb=# SET behavior_compat_options='enable_use_ora_timestamptz'; SET gaussdb=# SELECT timestamp '2024-03-20 01:30:00' at time zone 'Europe/Moscow' FROM dual; timezone --------------------- 2024-03-20 01:30:00+08 (1 row) **说明** 该参数仅在数据库兼容性为ORA时有效。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

forbid_skip_tableof_empty_str_elem	用于控制无索引的集合类型对于'{1,,3}'格式字符串的输入行为。
	• 打开该参数后，无索引的集合型将不支持'{1,,3}'格式字符串的输入。
	gaussdb=# CREATE TYPE t1 IS TABLE OF int;
	CREATE TYPE
	gaussdb=# CREATE OR REPLACE PROCEDURE p1(v t1) IS
	gaussdb$# BEGIN
	gaussdb$# FOR i IN 1..v.count LOOP
	gaussdb$# dbe_output.print_line('v(' || i || ') is ' || v(i));
	gaussdb$# END LOOP;
	gaussdb$# END;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# SET behavior_compat_options =
	'forbid_skip_tableof_empty_str_elem';
	SET
	gaussdb=# CALL p1('{1,,3}');
	ERROR: Malformed nesttable literal: "{1,,3}".
	LINE 1: call p1('{1,,3}');
	^
	gaussdb=# CALL p1('{1,}');
	ERROR: Malformed nesttable literal: "{1, }".
	LINE 1: call p1('{1, }');
	^
	gaussdb=# CALL p1('{,3}');
	ERROR: Malformed nesttable literal: "{ ,3}".
	LINE 1: call p1('{ ,3}');
	^
	gaussdb=# DROP PROCEDURE p1;
	DROP PROCEDURE
	gaussdb=# DROP TYPE t1;
	DROP TYPE
	• 关闭该参数后，无索引的集合型将支持'{1,,3}'格式字符串的输入，在进行转换时会自动过滤空串。
	gaussdb=# CREATE TYPE t1 IS TABLE OF int;
	CREATE TYPE
	gaussdb=# CREATE OR REPLACE PROCEDURE p1(v t1) IS
	gaussdb$# BEGIN
	gaussdb$# FOR i IN 1..v.count LOOP
	gaussdb$# dbe_output.print_line('v(' || i || ') is ' || v(i));
	gaussdb$# END LOOP;
	gaussdb$# END;
	gaussdb$# /
	CREATE PROCEDURE
	gaussdb=# SET behavior_compat_options = '';
	SET
	gaussdb=# CALL p1('{1,,3}');
	v(1) is 1
	v(2) is 3
	p1
	---- 	
	
	(1 row)
	
	gaussdb=# CALL p1('{1,}');
	v(1) is 1
	p1
	---- 	
	
	(1 row)
	
	gaussdb=# CALL p1('{,3}');

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

	v(1) is 3
	p1
	---- 	
	
	(1 row)
	
	gaussdb=# DROP PROCEDURE p1;
	DROP PROCEDURE
	gaussdb=# DROP TYPE t1;
	DROP TYPE

sys_function_without_brackets	开启该参数后，对于ORA兼容模式下无入参系统函数，可以不带括号进行调用。 gaussdb=# SET behavior_compat_options='sys_function_without_brackets'; gaussdb=# SELECT systimestamp; systimestamp --- 2024-04-16 15:55:21.614342+08 (1 row)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
array_count_compat	用于决定变长数组类型的变量为NULL时，调用array_count函数的返回值为NULL值还是0值。 • 开启参数后，变长数组类型的变量为NULL时，array_count函数返回值为0值。 gaussdb=# create type t1 as table of int; CREATE TYPE gaussdb=# set behavior_compat_options='array_count_compat'; SET gaussdb=# gaussdb=# declare gaussdb-# type typ is varray(5) of int; gaussdb-# va typ; gaussdb-# begin gaussdb$# raise info 'va count:%',va.count; gaussdb$# if 1 > va.count then gaussdb$# raise info ' 1 > null is true'; gaussdb$# else gaussdb$# raise info ' 1 > null is false'; gaussdb$# end if; gaussdb$# end; gaussdb$# / INFO: va count:0 INFO: 1 > null is true ANONYMOUS BLOCK EXECUTE • 关闭参数后，变长数组类型的变量为NULL时，array_count函数返回值为NULL值。 gaussdb=# set behavior_compat_options=''; SET gaussdb=# create type t1 as table of int; CREATE TYPE gaussdb=# declare gaussdb-# type typ is varray(5) of int; gaussdb-# va typ; gaussdb-# begin gaussdb$# raise info 'va count:%',va.count; gaussdb$# if 1 > va.count then gaussdb$# raise info ' 1 > null is true'; gaussdb$# else gaussdb$# raise info ' 1 > null is false'; gaussdb$# end if; gaussdb$# end; gaussdb$# / INFO: va count:<NULL> INFO: 1 > null is false ANONYMOUS BLOCK EXECUTE

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
search_path_behavior_optimization	启用/关闭存储过程搜索路径增强功能。 ● 新安装的数据库默认关闭该参数。 ● 开启该选项后，选项包含bind_procedure_searchpath选项能力。存储过程中会绑定会话级别search_path设置中的Schema，形成一份Schema列表。存储过程内未显式指定Schema的对象，会在这份Schema列表中按照顺序依次查找。如果同时开启bind_procedure_searchpath和search_path_behavior_optimization选项，默认使用search_path_behavior_optimization选项。 ● 开启该选项后，与单独开启bind_procedure_searchpath选项的区别在于：搜索路径编译产物中$user的刷新逻辑变化，以及搜索路径编译产物中pg_temp schema的存在变化。 ● 开启该选项后，具体行为变更如下： – 嵌套调用SECURITY DEFINER函数刷新搜索路径编译产物中的$user为嵌套最内层的函数owner。 – 本地临时表的namespace不加入搜索路径编译产物。 – 不支持存储过程/函数使用本地临时表%type、%rowtype写法（该限制不适用于本地临时表%type、%rowtype作为入参、出参场景）。例如： gaussdb=# SET behavior_compat_options='search_path_behavior_optimization'; SET -- 创建本地临时表temp_table gaussdb=# CREATE TEMP TABLE tmp_table(c1 int); CREATE TABLE -- 存储过程内使用本地临时表%rowtype写法，报错 gaussdb=# CREATE or REPLACE PROCEDURE test_temp_tab_proc AS gaussdb$# a tmp_table%ROWTYPE; gaussdb$# BEGIN gaussdb$# NULL; gaussdb$# END; gaussdb$# / ERROR: UNSUPPORTED FEATURE with temporary table when setting behavior_compat_options: search_path_behavior_optimization. CONTEXT: compilation of PL/pgSQL function "test_temp_tab_proc" near line 2

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

enable_ora_joinop_in_updatestmt	用于控制在UPDATE语句中SET子句带子查询时，子查询是否支持“+”操作符。
	开启该参数后，子查询返回的是带“+”的结果：
	gaussdb=# CREATE DATABASE a_db dbcompatibility = 'ora';
	CREATE DATABASE
	gaussdb=# \c a_db
	a_db=# CREATE TABLE t1(a1 int, b int);
	NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'a1' as the distribution column by default.
	HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
	CREATE TABLE
	a_db=# CREATE TABLE t2(c1 int, d int);
	NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'c1' as the distribution column by default.
	HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
	CREATE TABLE
	a_db=# CREATE TABLE t3(a2 int, c2 int, e int);
	NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'a2' as the distribution column by default.
	HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
	CREATE TABLE
	a_db=# INSERT INTO t1 VALUES(1, 11),(2, 12);
	INSERT 0 2
	a_db=# INSERT INTO t2 VALUES(3, 23),(4, 24);
	INSERT 0 2
	a_db=# INSERT INTO t3 VALUES(1, 3, 31),(5, 6, 32);
	INSERT 0 2
	a_db=# CREATE TABLE result(col1 int, col2 int, col3 int, col4 int, col5 int,col6 int, col7 int);
	NOTICE: The 'DISTRIBUTE BY' clause is not specified. Using 'col1' as the distribution column by default.
	HINT: Please use 'DISTRIBUTE BY' clause to specify suitable data distribution column.
	CREATE TABLE
	a_db=# INSERT INTO result SELECT * FROM t1, t2, t3 WHERE t1.a1=t3.a2 AND t2.c1=t3.c2;
	INSERT 0 1
	a_db=# UPDATE result SET (col2, col3, col4 ,col5,col6,col7) = (SELECT b,c1,d,a2,c2,e FROM t1, t2, t3 WHERE t1.a1=t3.a2(+) AND t3.c2=t2.c1(+));
	ERROR: The subquery in the expression returns more than one row of result.
	CONTEXT: referenced column: col2

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

disable_insert_void_value	用于控制void类型的返回值能否执行INSERT或UPDATE，ORA兼容性下生效。
	开启该参数后void类型执行INSERT或UPDATE失败，关闭时允许执行INSERT或UPDATE。
	gaussdb=# SELECT pg_typeof(dbe_application_info.set_action(''::text));
	 pg_typeof
	----------- 	
	 void
	(1 row)
	gaussdb=# SET behavior_compat_options = 'disable_insert_void_value';
	gaussdb=# CREATE TABLE insert_void (id int,col text) DISTRIBUTE BY HASH(id);
	gaussdb=# INSERT INTO insert_void (id,col) VALUES(1,dbe_application_info.set_client_info(''::text));
	ERROR: column "col" is of type text but expression is of type void
	LINE 1: INSERT INTO insert_void (id,col) VALUES(1,dbe_application_in...
	^
	HINT: You will need to rewrite or cast the expression.
	CONTEXT: referenced column: col
	gaussdb=# UPDATE insert_void SET col = dbe_application_info.set_client_info(''::text);
	ERROR: column "col" is of type text but expression is of type void
	LINE 1: UPDATE insert_void SET col = dbe_application_info.set_clie...
	^
	HINT: You will need to rewrite or cast the expression.
	CONTEXT: referenced column: col

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制

ignore_like_default_escape	用于LIKE模糊匹配时，控制模式串中的反斜杠'\'是否被视为转义字符。
	该参数仅在standard_conforming_strings为on时才可以被设置。
	• 开启参数后，LIKE模糊匹配中，支持将模式串中的反斜杠字符'\'视为普通字符进行匹配，'\'不起转义作用，无法消除通配符的特殊含义。
	gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
	CREATE DATABASE
	gaussdb=# \c ora_db
	ora_db=# SET standard_conforming_strings = on;
	SET
	ora_db=# SET behavior_compat_options = 'ignore_like_default_escape';
	SET
	ora_db=# CREATE TABLE test(c1 text) DISTRIBUTE BY HASH(c1);
	CREATE TABLE
	ora_db=# INSERT INTO test VALUES('aa\a');
	INSERT 0 1
	ora_db=# SELECT * FROM test WHERE c1 LIKE 'aa\a';
	c1
	------ 	
	aa\a
	(1 row)
	
	ora_db=# DROP TABLE test;
	DROP TABLE
	ora_db=# \c postgres
	gaussdb=# DROP DATABASE ora_db;
	• 关闭参数后，LIKE模糊匹配中，模式串中的反斜杠字符'\'视为转义字符，'\'起转义作用，可以消除通配符的特殊含义。
	gaussdb=# CREATE DATABASE ora_db DBCOMPATIBILITY = 'ORA';
	CREATE DATABASE
	gaussdb=# \c ora_db
	ora_db=# SET behavior_compat_options = '';
	SET
	ora_db=# CREATE TABLE test(c1 text) DISTRIBUTE BY HASH(c1);
	CREATE TABLE
	ora_db=# INSERT INTO test VALUES('aaa');
	INSERT 0 1
	ora_db=# SELECT * FROM test WHERE c1 LIKE 'aa\a';
	c1
	------ 	
	aaa
	(1 row)
	
	ora_db=# DROP TABLE test;
	DROP TABLE
	ora_db=# \c postgres
	gaussdb=# DROP DATABASE ora_db;

forbid_update_multi_same_tables	分布式场景不生效。
disable_rewrite_nesttable	分布式场景不生效。
aformat_null_test	分布式场景不生效。

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：根据数据库兼容对象进行调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

该参数的部分选项值影响缓存计划，在prepare语句下，修改该参数的值，执行结果不会改变。

a_format_version

参数说明：数据库平台兼容性行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前支持的兼容性配置项如表6-33所示。

默认值："10c"

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

表 6-33 兼容性配置项

兼容性配置项	兼容性行为控制
10c	ORA平台兼容10C版本。由于a_format_version和a_format_dev_version一起使用，详细的行为控制见表6-34。

a_format_dev_version

参数说明：数据库平台迭代小版本兼容性行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前支持的兼容性配置项如表6-34所示。

默认值："s7"

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

表 6-34 兼容性配置项

兼容性配置项	兼容性行为控制
s1	● ORA平台兼容迭代小版本，影响函数（round(date, fmt),nvl2,lpad,rpad,add_months,months_between,regexp_replace,regexp_count,empty_clob,instrb,trunc(numeric, int),greatest,least,mod,round(number,int),cast,to_date,to_timestamp,chr,to_char,to_number，to_timestamp_tz）。 – round(date, fmt)：此函数的入参为timestamp without time zone时，在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 – nvl2：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 – lpad：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下： – 参数length表示字符串显示长度，单个字符的显示长度按照ORA兼容处理。 – 函数执行过程中出现length剩余长度为1且下一个字符显示长度为2，在字符串左侧添加一个空格字符。 – 参数length入参若为小数则不会被四舍五入，而是被截断。 – 参数string与fill不支持不符合编码规范的输入。 – rpad：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下： – 参数length表示字符串显示长度，单个字符的显示长度按照ORA兼容处理。 – 函数执行过程中出现length剩余长度为1且下一个字符显示长度为2，在字符串右侧添加一个空格字符。 – 参数length入参若为小数则不会被四舍五入，而是被截断。 – 参数string与fill不支持不符合编码规范的输入。 – 参数string有长度限制，若填充后大小超出1G会报错。 – add_months：此函数在GUC参数mapping_date_to_datea值为off的情况下，数据库会将入参中的date类型解析为timestamp(0) without time zone类型，此时返回值为timestamp类型。当该函数入参类型被解析为timestamp(0) without time zone时，此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下： – 当计算结果大于公元9999年时会报错。 – 参数n入参若为小数则不会被四舍五入，而是被截断。 – months_between：此函数在GUC参数mapping_date_to_datea值为off的情况下，数据库会将入参中的date类型解析为timestamp(0) without time zone类型。当该函数入参类型被解析为timestamp(0) without time zone

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
	时，此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 – regexp_replace：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下，occurrence参数默认值为0，表示替换所有匹配到的子串。并且以'\'字符结尾的pattern参数为合法的。 – regexp_count：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下，以'\'字符为结尾的pattern参数为合法的。 – empty_clob：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 – instrb：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 – trunc(numeric, int)：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1时，参数s若入参为小数则不会被四舍五入，而是被截断。 – greatest：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下： – 如果参数中有任意一个参数的值为NULL，函数返回NULL。 – 如果第一个参数是数值类型，函数将第一个参数和其他参数都转换为numeric类型，然后进行比较，对于不能转换的，提示出错信息；第一个参数是其他类型的，函数将其他参数都转换为第一个参数的类型进行比较，对于不能转换的，提示出错信息。 – least：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s1的情况下： – 如果参数中有任意一个参数的值为NULL，函数返回NULL。 – 如果第一个参数是数值类型，函数将第一个参数和其他参数都转换为numeric类型，然后进行比较，对于不能转换的，提示出错信息；第一个参数是其他类型的，函数将其他参数都转换为第一个参数的类型进行比较，对于不能转换的，提示出错信息。 – mod：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s6的情况下，支持mod(text, int)重载，在以(text, int)为入参类型调用mod函数时，会优先选择mod(text, int)。 – round(numeric, int)： – 控制参数s输入为小数时：在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下s截断为整数，否则s四舍五入为整数。 – 在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，round函数支持round(timestamp, text)

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
	重载，在以(text, text)或(text, '')为入参调用round函数时会优先选择round(timestamp, text)。 – cast：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s1的情况下，支持DEFAULT z ON CONVERSION ERROR的可选语法及fmt可选参数 – to_date：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s1的情况下，年份的输入超过9999时，系统可能不报错。如to_date('99999-12-12', 'yyyy-mm-dd hh24:mi:ss')结果为9999-09-12 12:00:00。年输入超过9999时，超过4位后面的数字会解析为下一个fmt。 – to_timestamp：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s1的情况下，fmt支持FF[7-9]，在FF[7-9]的情况下允许转换string中的对应位置长度小于等于FF紧跟的数字，但最终转换结果长度最大保留6位。 – chr：此函数在ORA兼容模式数据库中且参数a_format_version值为10c和a_format_dev_version值为s1的情况下：参数integer入参若为小数则不会被四舍五入，而是被截断。 – to_char(numeric/smallint/integer/bigint/double precision/real[, fmt])：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s1的情况下，支持$、C、TM、TM9、TME、U格式。同时在该参数下，不支持TH、PL、SG格式的fmt。 – to_number：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s1的情况下，to_number函数的格式fmt详情可参考《开发指南》中的to_number函数介绍。 – to_timestamp_tz：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s1的情况下有效。 ● 数据类型转换：小数字符串转换成整数类型（int1/int2/int4/int8/int16）时进行四舍五入。 ● 数据类型转换：支持timestamp with time zone到timestamp without time zone和datea，以及timestamp到datea的隐式转换。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
s2	● ORA平台兼容迭代小版本，影响函数（dump, to_single_byte, to_multi_byte, nls_upper, nls_lower, initcap, ascii2, asciistr, unistr, vsize, cosh, remainder, sinh, tanh, nanvl, current_date, current_timestamp, dbtimezone, numtodsinterval, numtoyminterval, new_time, sessiontimezone, sys_extract_utc, tz_offset, to_binary_double, to_binary_float, to_dsinterval, to_yminterval, lnnvl, ora_hash）。 – dump：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – to_single_byte：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – to_multi_byte：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – nls_upper：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下支持使用。 – nls_lower：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下支持使用。 – initcap：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下，如中文等无大小写区分的字符后接区分大小写的字符首字母会转换为大写，其他字母转换为小写。因此建议开启参数a_format_version值为10c和a_format_dev_version值为s2。 – ascii2：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – asciistr：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – unistr：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – vsize：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – cosh：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – remainder：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – sinh：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – tanh：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – nanvl：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – current_date：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下返回值类型为timestamp。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
	– current_timestamp：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2时，precision参数支持numeric类型的整值，否则仅支持int输入。 – dbtimezone：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – numtodsinterval：此函数在ORA兼容模式下，_format_version值为10c和a_format_dev_version值为s2的情况下：当参数interval_unit为 'DAY' 时，参数num超过1000000000会报错。 – numtoyminterval：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – new_time：此函数GUC参数mapping_date_to_datea值为off的情况下，数据库会将入参中的date类型解析为timestamp(0) without time zone类型，此时返回值为timestamp类型。当该函数入参类型被解析为timestamp(0) without time zone时，此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – sessiontimezone：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – sys_extract_utc：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – tz_offset：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – to_binary_double：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s2的情况下有效。 – to_binary_float：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s2的情况下有效。 – to_dsinterval：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s2的情况下有效。 – to_yminterval：此函数在ORA兼容模式下，a_format_version值为10c且a_format_dev_version值为s2的情况下有效。 – lnnvl：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下有效。 – ora_hash：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s2的情况下才能有效。 ● 兼容配置项为s1时的所有行为。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
s3	● ORA平台兼容迭代小版本，开启参数，支持无参函数嵌套调用。 ● 兼容配置项为s2时的所有行为。
s4	● ORA平台兼容迭代小版本，影响函数nchr(cvalue int|bigint)、to_timestamp_tz、getclobval(xml)、getstringval(xml)。 – nchr(cvalue int|bigint)：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s4的情况下有效。 – to_timestamp_tz(string [DEFAULT return_value ON CONVERSION ERROR], fmt, nlsparam)：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s4的情况下有效。 – getclobval(xml)：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s4的情况下有效。 – getstringval(cml)：此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s4的情况下有效。 ● 兼容配置项为s3时的所有行为。
s5	● ORA平台兼容迭代小版本，开启参数，支持复合类型与函数同名，优先解析为复合类型，影响函数sys_guid()。 – 此函数在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s5的情况下才能生效。 ● 兼容配置项为s4时的所有行为。
s6	● ORA平台兼容迭代小版本，开启参数，影响函数mod(text, int)。 – mod(text, int)：在ORA兼容模式下，a_format_version值为10c和a_format_dev_version值为s6的情况下，mod函数支持mod(text, int)重载，在以(text, int)为入参类型调用mod函数时，会优先选择mod(text, int)。 ● 兼容配置项为s5时的所有行为。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
s7	• ORA平台兼容迭代小版本，开启参数。 • 当开启该参数时，存在以下行为兼容： 1. 兼容GUC参数配置为behavior_compat_options='display_leading_zero, end_month_calculate, plstmt_implicit_savepoint, truncate_numeric_tail_zero, char_coerce_compat, aformat_regexp_match, compat_cursor, plsql_security_definer, proc_outparam_override, dynamic_sql_compat, proc_outparam_transfer_length, tableof_elem_constraints, current_sysdate, plsql_rollback_keep_user, enable_bpcharlikebpchar_compare, show_full_error_lineno, enable_crosstype_integer_operator, forbid_package_function_with_prefix, enable_case_when_alias, time_constexpr_compact, enable_use_ora_timestamptz, forbid_skip_tableof_empty_str_elem, sys_function_without_brackets'的所有行为。 2. 兼容GUC参数配置为a_format_date_timestamp=on的所有行为。 • 兼容配置项为s6时的所有行为。 • JDBC驱动连接数据库的场景，若需要设置该参数，需同步更新JDBC驱动。

b_format_version

参数说明：数据库平台兼容性行为配置项，用于控制MySQL兼容模式下的前向兼容性。

须知

- 仅当**sql_compatibility**兼容性参数设置为MYSQL时，b_format_version参数才会生效。
- 当该参数设置为非空字符串时，会同时设置参数**b_format_behavior_compat_options**为“ALL”值，参数**bytea_output**为“escape”值。当该参数重新设置为空字符串时，会将b_format_behavior_compat_options、bytea_output参数设为原来的值。
- 该参数需要CN、DN同时设置，保持一致。
- 为保障数据一致性，当用户创建了带表字符序或列字符序的表后，须搭配b_format_version='5.7'与b_format_dev_version='s2'使用，且在生产环境运行过程中不允许更改。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：空字符串""或"5.7"

默认值：""

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：与b_format_dev_version一起设置，在安装部署后立即设置，后期非必要无需变更。

设置不当的风险与影响：影响部分SQL的行为，导致兼容性问题。

b_format_dev_version

参数说明：数据库平台迭代小版本兼容性行为配置项。

须知

- 当sql_compatibility兼容性参数设置为MYSQL且b_format_version设置为“5.7”时，b_format_dev_version才会生效。
- 该参数需要CN、DN同时设置，保持一致。
- 为保障数据一致性，当用户创建了带表字符序或列字符序的表后，须搭配b_format_version='5.7'与b_format_dev_version='s2'使用，且在生产环境运行过程中不允许更改。

参数类型：字符串

参数单位：无

取值范围：空字符串""或表6-35中的兼容性配置项。

版权所有 © 华为云计算技术有限公司

表 6-35 兼容性配置项

兼容性配置项	兼容性行为控制
s1	1. 影响如下参数： 参数打开会将disable_keyword_options参数中的datetime选项去除，表示datetime作为非保留关键字使用。datetime可以作为表名、列名、别名使用，但不能作为函数名、存储过程名、函数或存储过程的参数名使用。 2. 影响如下函数： curdate，from_days，date_format，str_to_date，current_date，datediff，timestampdiff，date_add，subtime，month，time_to_sec，to_days，to_seconds，dayname，monthname，convert_tz，sec_to_time，addtime，adddate，date_sub，timediff，last_day，weekday，from_unixtime，unix_timestamp，subdate，day，year，weekofyear，dayofmonth，dayofyear，week，yearweek，dayofweek，time_format，hour，minute，second，microsecond，quarter，utc_date，get_format，extract，makedate，period_add，timestamppadd，period_diff，utc_time，utc_timestamp，sysdate，current_timestamp，maketime，curtime，current_time，localtime，localtimestamp，now，lc_time_names，default_week_format，json_object，substring。 3. 影响如下类型： a. 兼容整型类型tinyint，范围变更为-128 ~ 127。 b. 兼容字符串类型char、varchar、char(n)和varchar(n)的精度n由字节长度变为字符长度。 c. 兼容文本类型tinytext、mediumtext和longtext。这些类型在开启参数后会被映射为text类型。 d. 兼容二进制类型tinyblob、blob、mediumblob和longblob。这些类型在开启参数后会被映射为bytea类型。 e. 兼容浮点类型double、float和real。double映射为float8类型，double(p, s)映射为numeric类型，float(p, s)映射为numeric类型，real映射为float8，real(p, s)映射为numeric类型。 f. 兼容高精度类型numeric、dec和decimal。未指定精度和标度时变更默认精度为10，标度为0。 g. 兼容时间类型datetime、timestamp、time和date。datetime[(p)]替换成timestamp without time zone[(p)]类型。timestamp[(p)] 替换成timestamp with time zone[(p)]类型。这些类型的输入输出、范围、精度等规格有变更，具体请参见**数据类型**章节下的相关章节。 h. 兼容整型类型tinyint、smallint、int、bigint，这些类型在开启参数后显示宽度和ZEROFILL属性功能生效。 4. 影响如下语法： 支持指定列约束ON UPDATE update_expr。 5. 影响如下操作符：

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
	a. 使用bytea类型间的||操作符报错。 b. 支持REGEXP操作符。 6. 影响如下GUC参数： 参数bytea_output='escape'时，处理逻辑发生变化，具体请参见**bytea_output**参数内容。
s2	包含s1控制的兼容性行为，此外还包含以下影响： 1. 影响如下参数： a. 此选项打开时，standard_conforming_strings、escape_string_warning参数被置为off。 b. 此选项打开时，collation_connection、character_set_connection参数功能才会生效。 2. 影响如下功能： a. like操作符对于匹配串末尾是转义符的场景不再报错。 b. 字符集、字符序的排序规则优先级将发生变更，详细信息请参见**字符集和字符序合并规则**。 c. 打开参数后，不支持外键字符序和列上字符序不一致。 3. 影响如下语法： a. 支持“CREATE TABLE table_name LIKE source_table”语法。 b. “CREATE TABLE table_name LIKE source_table”和“CREATE TABLE table_name (LIKE source_table)”语法不支持指定INCLUDING和EXCLUDING选项，缺省等同于指定INCLUDING ALL。 c. 支持LOAD DATA语法。 d. 涉及ALTER TABLE，RENAME TABLE等更改表名的语法，影响场景如下： 新表名对应字符串开头为“#MySQL50#”，且其后有其他字符，“#MySQL50#”将被忽略；如果新旧表名一致，不会报错。 e. 支持set names指定collate子句。

默认值：""

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：与**b_format_version**一起设置，在安装部署后立即设置，后期非必要无需变更。

设置不当的风险与影响：影响部分SQL的行为，导致兼容性问题。

m_format_dev_version

参数说明：数据库平台迭代小版本兼容性行为配置项。

版权所有 © 华为云计算技术有限公司

说明

当**sql_compatibility**兼容性参数设置为M时，m_format_dev_version才会生效。

参数类型：字符串

参数单位：无

取值范围：表6-36中的兼容性配置项。目前只可以设置"s1"。

表 6-36 兼容性配置项

兼容性配置项	兼容性行为控制
s1	影响如下语法： 1. 禁用SELECT FETCH FIRST ROWS ONLY语法。 2. 禁用TRUNCATE CASCADE/RESTRICT语法。 3. 禁用CREATE TABLE中定义主键时指定USING INDEX TABLESPACE tablespace_name语法。 4. DROP TABLE/VIEW/COLUMN中CASCADE/RESTRICT语法语义改变，改为忽略CASCADE/RESTRICT的语义，且默认行为为RESTRICT。 5. CREATE TABLE REFERENCES作为列约束时，语法语义改变，改为忽略该外键定义。

默认值："s1"

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：在安装部署后立即设置，后期非必要无需变更。

设置不当的风险与影响：影响部分SQL的行为，导致兼容性问题。

sql_mode

参数说明：MYSQL兼容模式下和M-Compatibility模式下SQL行为控制配置项。

说明

- 当**sql_compatibility**兼容性参数设置为MYSQL、**b_format_version**设置为'5.7'、**b_format_dev_version**设置为's1'时，sql_mode才会生效；或者sql_compatibility兼容性参数设置为M时生效。
- 使用SHOW sql_mode和SELECT @@sql_mode进行查询时，会依据REAL_AS_FLOAT、ANSI_QUOTES、ONLY_FULL_GROUP_BY、NO_UNSIGNED_SUBTRACTION、NO_AUTO_VALUE_ON_ZERO、NO_BACKSLASH_ESCAPES、STRICT_TRANS_TABLES、STRICT_ALL_TABLES、NO_ZERO_IN_DATE、NO_ZERO_DATE、ALLOW_INVALID_DATES、ERROR_FOR_DIVISION_BY_ZERO、TRADITIONAL、NO_AUTO_CREATE_USER、NO_ENGINE_SUBSTITUTION及PAD_CHAR_TO_FULL_LENGTH的顺序，大写显示，不因设置时的顺序与大小写而改变。

参数类型：字符串

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- MYSQL模式：空字符串""或"strict_trans_tables,only_full_group_by,no_zero_in_date,no_zero_date,error_for_division_by_zero,no_auto_create_user,no_engine_substitution,pad_char_to_full_length,no_auto_value_on_zero"中的选项值。
- M-Compatibility模式：除支持MYSQL模式参数外，还新增了no_unsigned_subtraction、ansi_quotes、allow_invalid_dates、real_as_float、strict_all_tables、traditional。

每个选项值的含义如表6-37所述。

表 6-37 兼容性配置项

兼容性配置项	兼容性行为控制	MYSQL模式参数配置说明	M-Compatibility模式参数配置说明
strict_trans_tables	当前仅支持对MYSQL兼容数据库和M-Compatibility数据库数据类型及系统函数做校验。 设置时，对输入的参数格式和范围做严格校验。无效值输入或者超过范围时，解析报error。 不设置时，对输入的参数格式和范围做宽松校验，无效值输入或者超过范围时，解析报warning，返回对应数据宽松处理后的值。 处理方法： 1、异常数据 字符串插入数值类型时，从第一个非数字开始截断，如'1a'->1。 目标类型为时间类型时，返回对应数据的0值。 2、超出最大范围 返回对应数据的最大/最小值。	可独立设置和取消。	可独立设置和取消。
only_full_group_by	不允许出现非GROUP BY的key值、且不是常量和聚集函数的投影列。	可独立设置和取消。	可独立设置和取消。
no_zero_in_date	不允许日期中的年、月、日是0值。	可独立设置和取消，但无实际作用。	可独立设置和取消。
no_zero_date	不允许日期是0值（0000-00-00）。	可独立设置和取消，但无实际作用。	可独立设置和取消。
error_for_division_by_zero	不允许被0除。	可独立设置和取消，但无实际作用。	可独立设置和取消。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制	MYSQL模式参数配置说明	M-Compatibility模式参数配置说明
no_auto_create_user	无实际作用，仅用作SET SQL_MODE语句包含此选项时不报错的兼容作用。	可独立设置和取消。	可独立设置和取消。
no_engine_substitution	无实际作用，仅用作SET SQL_MODE语句包含此选项时不报错的兼容作用。	可独立设置和取消。	可独立设置和取消。
pad_char_to_full_length	用于表中的char类型列格式化输出，包含时输出带有尾部空格的字符串，否则输出不带尾部空格字符串。	可独立设置和取消。	可独立设置和取消。
no_auto_value_on_zero	包含此选项后，向AUTO_INCREMENT自动增长列插入0值将不会自增。	可独立设置和取消。	可独立设置和取消。
no_unsigned_subtraction	禁止无符号数减法，如果使用无符号整数进行减法操作，且结果为负数，则会返回一个错误。	可独立设置和取消，但无实际作用。	可独立设置和取消。
ansi_quotes	启用ANSI_QUOTES模式，该模式下双引号被视为标识符引用符号，而不是字符串引用符号。	可独立设置和取消，但无实际作用。双引号总被视为标识符引用符号。	可独立设置和取消。
allow_invalid_dates	允许插入无效日期，如果启用该选项，则MySQL将允许插入无效的日期，例如'0000-00-00'。	可独立设置和取消，但无实际作用。	可独立设置和取消。
real_as_float	默认将REAL类型视为DOUBLE类型，如果启用该选项，则将REAL类型视为FLOAT类型。	可独立设置和取消，但无实际作用。	可独立设置和取消。
strict_all_tables	与strict_trans_tables效果一致。	可独立设置和取消。	可独立设置和取消。
traditional	与同时设置strict_trans_tables、strict_all_tables、no_zero_in_date、no_zero_date、error_for_division_by_zero、no_auto_create_user和no_engine_substitution效果一致。	可独立设置和取消。 **说明** 具体是否生效取决于MYSQL兼容模式本身是否支持。	可独立设置和取消。

默认值：
"strict_trans_tables,only_full_group_by,no_zero_in_date,no_zero_date,error_for_division_by_zero,no_auto_create_user,no_engine_substitution"

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

auto_increment_increment

参数说明：AUTO_INCREMENT自动增长列的自增步长，自增值将由auto_increment_offset + N × auto_increment_increment计算而来，N为正整数。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：1

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据用户需要设置。

设置不当的风险与影响：设置不当会使自增值不符合预期。若参数值小于**auto_increment_offset**，自动增长列自增时会产生错误。

auto_increment_offset

参数说明：AUTO_INCREMENT自动增长列的起始值，自增值将由auto_increment_offset + N × auto_increment_increment计算而来，N为正整数。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：1

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据用户需要设置。

设置不当的风险与影响：设置不当会使自增值不符合预期。若参数值大于**auto_increment_increment**，自动增长列自增时会产生错误。

auto_increment_cache

参数说明：对自动增长列批量插入或导入触发自增时，预留自增缓存值的个数。预留自增值时，刷新自增计数器为自增缓存值的最大值，缓存值使用完毕之前，自增计数器值不变，触发自增使用缓存的值。

版权所有 © 华为云计算技术有限公司

说明

- 预留的缓存值仅在语句内有效，若预留的自增缓存值没有全部被使用，后续插入语句基于自增计数器触发自增，会出现表中自动增长列的值不连续的情况。
- 并行导入或插入自动增长列触发自增时，每个并行线程预留的缓存值也只在其线程中使用，未完全使用完毕的话，也会出现表中自动增长列的值不连续的情况。
- 对已有数据的表添加自动增长列或修改某列为自动增长列时，已有数据触发自增，预留自增缓存值也受此参数影响。
- 此参数不影响本地临时表中的自动增长列。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

- 参数值为0时，自动预留自增缓存值。
  - 第一次触发自增时，若已知即将插入自动增长列的行数，预留已知数量的值。
    - 例如：“INSERT INTO table SELECT ...”和COPY FROM触发自增值无法获取即将插入的行数。ALTER TABLE命令表数据重写时，若触发自增，使用统计信息中的reltuples作为即将预留的行数。“INSERT INTO table VALUES(...),(...),...”由于要分布到不同DN，在某些执行计划中，DN获取不到即将插入的行数。
  - 若行数未知，每次预留2^n个值，例如，第一次自增预留1个值，第二次自增预留2个值，第三次预留4个值，第四次预留8个值，以此类推，若预留值个数超过65535，预留65535个值。
- 参数值不为0时，预留缓存值的数量为参数值。
  - 第一次触发自增时，若已知即将插入自动增长列的行数，预留已知数量的值。
  - 若行数未知，每次预留auto_increment_cache数量的自增值。

默认值：0

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，若默认情况产生自增值不连续的情况，可以根据用户即将批量插入数据量调整参数值。

设置不当的风险与影响：参数值越大，批量插入性能越优，也更可能产生不连续的自增值。

auto_increment_mode

参数说明：在M-Compatibility模式数据库中进行建表时，如果AUTO_INCREMENT字段不指定LOCAL和GLOBAL关键字，可通过auto_increment_mode参数指定该字段为全局自增列或本地自增列。

参数类型：字符串

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- 'global'：全局自增列。
- 'local'：本地自增列。

默认值：'global'

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据用户实际需求进行设置。

设置不当的风险与影响：设置不当会使自增值不符合预期。

disable_keyword_options

参数说明：数据库兼容性行为配置项，该参数的值由若干个配置项用逗号隔开构成。设置该参数的标识符将不会作为关键字使用。

参数类型：字符串

参数单位：无

取值范围：不同兼容模式的数据库支持设置的关键字有所差异，具体请参见表6-38。

表 6-38 各兼容模式数据库支持设置 disable_keyword_options 选项的关键字

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
absolute	支持	不支持	不支持
access	支持	不支持	支持
account	支持	不支持	不支持
action	支持	不支持	不支持
active	不支持	不支持	支持
add	支持	不支持	不支持
adddate	不支持	支持	不支持
admin	不支持	不支持	支持
advanced	支持	支持	不支持
algorithm	支持	不支持	不支持
always	支持	不支持	不支持
app	支持	不支持	不支持
append	支持	不支持	不支持
archive	支持	不支持	不支持
array	不支持	不支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
assertion	支持	不支持	不支持
assignment	支持	不支持	不支持
async	支持	不支持	不支持
at	支持	不支持	不支持
audit	支持	不支持	不支持
authentication	不支持	不支持	支持
authid	不支持	不支持	支持
auto	支持	支持	不支持
auto_increment	支持	支持	不支持
autoextend	支持	不支持	不支持
autohint	支持	支持	支持
automapped	支持	不支持	不支持
automatic	支持	支持	不支持
backward	支持	不支持	不支持
bad_path	支持	不支持	不支持
barrier	支持	不支持	不支持
before	支持	不支持	不支持
begin_non_anoyblock	支持	不支持	不支持
blanks	支持	不支持	不支持
blockchain	支持	不支持	不支持
body	支持	不支持	不支持
buckets	不支持	不支持	支持
bulk	不支持	不支持	支持
byte	支持	支持	不支持
byteawithoutorder	支持	不支持	不支持
byteawithoutorderwithequal	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
cache	支持	不支持	不支持
called	支持	不支持	支持
cancelable	支持	不支持	支持
catalog	支持	不支持	不支持
chain	支持	不支持	不支持
challenge_response	不支持	不支持	支持
change	支持	支持	不支持
characteristics	支持	不支持	不支持
charset	支持	支持	不支持
class	支持	不支持	不支持
clean	支持	不支持	不支持
clone	不支持	不支持	支持
client	支持	不支持	不支持
client_master_key	支持	不支持	不支持
client_master_keys	支持	不支持	不支持
columns	支持	支持	不支持
column_encryption_key	支持	不支持	不支持
column_encryption_keys	支持	不支持	不支持
colview	支持	支持	不支持
comment	支持	不支持	不支持
comments	支持	不支持	不支持
compact	支持	不支持	不支持
compatible_illegal_chars	支持	不支持	不支持
compile	支持	支持	不支持
complete	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
completion	支持	支持	不支持
component	不支持	不支持	支持
concurrently	支持	不支持	不支持
condition	支持	不支持	不支持
configuration	支持	不支持	不支持
conflict	不支持	支持	不支持
connection	支持	不支持	不支持
constant	支持	不支持	不支持
containing	支持	支持	不支持
content	支持	不支持	支持
conversion	支持	不支持	不支持
convert	不支持	支持	不支持
coordinator	支持	不支持	不支持
coordinators	支持	不支持	不支持
copy_custom_id	支持	支持	不支持
cost	支持	不支持	不支持
crossbucket	支持	支持	不支持
csn	支持	支持	不支持
csv	支持	不支持	不支持
cube	支持	不支持	不支持
current	支持	不支持	不支持
current_schema	支持	不支持	不支持
cume_dist	不支持	不支持	支持
cycle	支持	不支持	不支持
data	支持	不支持	不支持
datatype_cl	支持	不支持	不支持
datafiles	支持	支持	不支持
date_add	不支持	支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
date_format	支持	不支持	不支持
date_sub	不支持	支持	不支持
datetime	支持	支持	不支持
day	支持	不支持	不支持
day_hour	支持	支持	不支持
day_microsecond	支持	支持	不支持
day_minute	支持	支持	不支持
day_second	支持	支持	不支持
days	支持	支持	不支持
db4aishot	支持	支持	不支持
dbcompatibility	支持	不支持	不支持
dbtimezone	支持	支持	不支持
deallocate	支持	不支持	不支持
decode	支持	不支持	不支持
deferred	支持	不支持	不支持
definer	支持	不支持	不支持
definition	不支持	不支持	支持
delete_all	支持	支持	不支持
delimiter	支持	不支持	不支持
delimiters	支持	不支持	不支持
delta	支持	不支持	不支持
deltamerge	支持	不支持	不支持
dense_rank	支持	支持	支持
description	不支持	不支持	支持
deterministic	支持	不支持	不支持
dictionary	支持	不支持	不支持
direct	支持	不支持	不支持
directory	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
disable_all	支持	支持	不支持
discard	支持	不支持	不支持
discard_path	支持	支持	不支持
disconnect	支持	不支持	不支持
distribute	支持	不支持	不支持
distributed	支持	支持	不支持
distribution	支持	不支持	不支持
document	不支持	不支持	支持
dumpfile	支持	支持	不支持
duplicate	支持	不支持	不支持
each	支持	不支持	不支持
elastic	支持	不支持	不支持
empty	支持	支持	支持
enable_all	支持	支持	不支持
enclosed	支持	不支持	不支持
encoding	支持	不支持	支持
encrypted	支持	不支持	不支持
encrypted_value	支持	不支持	不支持
encryption	支持	不支持	不支持
encryption_type	支持	不支持	不支持
ends	支持	支持	不支持
enforced	支持	不支持	支持
engine	支持	支持	不支持
engine_attribute	不支持	不支持	支持
entityescaping	支持	支持	不支持
eol	支持	不支持	不支持
errors	支持	不支持	不支持
escaped	支持	支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
escaping	支持	不支持	不支持
evalname	支持	支持	不支持
event	支持	支持	不支持
events	支持	支持	不支持
every	支持	不支持	不支持
exchange	支持	不支持	不支持
except	不支持	不支持	支持
exclude	支持	不支持	支持
excluding	支持	不支持	不支持
execute	支持	不支持	不支持
exclusive	不支持	不支持	支持
expdp	支持	支持	不支持
expire	支持	支持	不支持
expired	支持	不支持	不支持
extend	支持	支持	不支持
external	支持	不支持	不支持
family	支持	不支持	不支持
fast	支持	不支持	不支持
features	支持	不支持	不支持
factor	不支持	不支持	支持
failed_login_attempts	不支持	不支持	支持
fenced	不支持	不支持	支持
fields	支持	不支持	不支持
fileheader	支持	不支持	不支持
fill_missing_fields	支持	不支持	不支持
filler	支持	不支持	不支持
filter	支持	不支持	不支持
finish	支持	支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
first	支持	不支持	不支持
first_value	不支持	不支持	支持
fixed	支持	不支持	不支持
following	支持	不支持	支持
force	不支持	不支持	支持
formatter	支持	不支持	不支持
forward	支持	不支持	不支持
freeze	支持	不支持	不支持
functions	支持	不支持	不支持
generated	支持	不支持	不支持
generate	不支持	不支持	支持
geomcollection	不支持	不支持	支持
get_format	不支持	支持	不支持
get_master_public_key	不支持	不支持	支持
get_source_public_key	不支持	不支持	支持
granted	支持	不支持	不支持
greatest	支持	不支持	不支持
grouping	支持	不支持	支持
groups	不支持	不支持	支持
gsidead	支持	支持	不支持
gsigetxid	支持	支持	不支持
gsimerge	支持	支持	不支持
gsiready	支持	支持	不支持
gsiswitch	支持	支持	不支持
gsiusable	支持	支持	不支持
gsivalid	支持	支持	不支持
gsiwaitall	支持	支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
gtid_only	不支持	不支持	支持
hdfsdirectory	支持	不支持	不支持
header	支持	不支持	不支持
high	支持	支持	不支持
hold	支持	不支持	不支持
histogram	不支持	不支持	支持
history	不支持	不支持	支持
hour	支持	不支持	不支持
hour_microsecond	支持	支持	不支持
hour_minute	支持	支持	不支持
hour_second	支持	支持	不支持
hours	支持	支持	不支持
identity	支持	不支持	不支持
ifnull	不支持	支持	不支持
ignore	支持	支持	不支持
ignore_extra_data	支持	不支持	不支持
ilm	支持	支持	不支持
ilm_pidx_list	支持	支持	不支持
imcvtable	支持	支持	不支持
immediate	支持	不支持	不支持
immutable	支持	不支持	不支持
impdp	支持	支持	不支持
implicit	支持	不支持	不支持
inactive	不支持	不支持	支持
include	支持	不支持	不支持
including	支持	不支持	不支持
increment	支持	不支持	不支持
incremental	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
indexes	支持	不支持	不支持
infile	支持	不支持	不支持
inherit	支持	不支持	不支持
inherits	支持	不支持	不支持
initial	支持	不支持	支持
initrans	支持	不支持	不支持
initiate	不支持	不支持	支持
inline	支持	不支持	不支持
inner	支持	不支持	不支持
inout	支持	不支持	不支持
input	支持	不支持	不支持
insensitive	支持	不支持	不支持
instead	支持	不支持	不支持
intersect	不支持	不支持	支持
internal	支持	不支持	不支持
interval	支持	支持	不支持
invisible	支持	支持	支持
invoker	支持	不支持	不支持
isnull	支持	不支持	不支持
json_object	不支持	支持	不支持
json_table	不支持	不支持	支持
json_value	不支持	不支持	支持
keep	支持	支持	不支持
key_path	支持	不支持	不支持
key_store	支持	不支持	不支持
kill	支持	不支持	不支持
keyring	不支持	不支持	支持
label	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
large	支持	不支持	不支持
lag	不支持	不支持	支持
last	支持	不支持	不支持
last_day	不支持	支持	不支持
last_value	不支持	不支持	支持
lateral	不支持	不支持	支持
lc_collate	不支持	不支持	支持
lead	不支持	不支持	支持
leakproof	支持	不支持	支持
least	支持	不支持	不支持
left	支持	不支持	不支持
lines	支持	支持	不支持
link	支持	支持	不支持
listen	支持	不支持	不支持
lnnvl	支持	支持	不支持
load_bad	支持	不支持	支持
load_discard	支持	支持	支持
load_integer_len	支持	支持	不支持
load_session_id	支持	支持	不支持
load_smallint_len	支持	支持	不支持
load_unfixed_len	支持	支持	不支持
load_unfixed_start_pos	支持	支持	不支持
local	支持	不支持	不支持
location	支持	不支持	支持
locked	支持	支持	支持
logging	支持	不支持	不支持
login_any	支持	不支持	不支持
login_failure	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
login_success	支持	不支持	不支持
logout	支持	不支持	不支持
loop	支持	不支持	不支持
low	支持	支持	不支持
manual	支持	支持	不支持
mapping	支持	不支持	不支持
mark	支持	支持	不支持
masking	支持	不支持	不支持
master	支持	不支持	不支持
master_compression_algorithms	不支持	不支持	支持
master_public_key_path	不支持	不支持	支持
master_tls_ciphersuites	不支持	不支持	支持
master_zstd_compression_level	不支持	不支持	支持
match	支持	不支持	支持
matched	支持	不支持	不支持
materialized	支持	不支持	不支持
maxextents	支持	不支持	不支持
maxsize	支持	不支持	不支持
maxtrans	支持	不支持	不支持
medium	支持	支持	不支持
mediumint	支持	支持	不支持
member	不支持	不支持	支持
microsecond	支持	支持	不支持
minextents	支持	不支持	不支持
minute	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
minute_microsecond	支持	支持	不支持
minute_second	支持	支持	不支持
minvalue	支持	不支持	支持
model	支持	不支持	不支持
modification	支持	支持	不支持
month	支持	不支持	不支持
months	支持	支持	不支持
move	支持	不支持	支持
movement	支持	不支持	不支持
names	支持	不支持	不支持
national	支持	不支持	不支持
nested	不支持	不支持	支持
next	支持	不支持	不支持
nocache	支持	支持	不支持
nocolview	支持	支持	不支持
nocycle	不支持	不支持	支持
node	支持	不支持	支持
noentityescaping	支持	支持	不支持
noextend	支持	支持	不支持
nologging	支持	不支持	不支持
nomaxvalue	支持	不支持	支持
nominvalue	支持	不支持	支持
none	支持	不支持	不支持
noscale	支持	支持	不支持
nothing	支持	不支持	不支持
notify	支持	不支持	不支持
notnull	支持	不支持	不支持
now	支持	支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
nowait	不支持	不支持	支持
nth_value	不支持	不支持	支持
ntile	不支持	不支持	支持
nullcols	支持	不支持	不支持
nullif	支持	不支持	不支持
nulls	支持	不支持	支持
numstr	支持	不支持	不支持
nvl	支持	不支持	不支持
nvl2	支持	支持	不支持
object	支持	不支持	不支持
of	不支持	不支持	支持
off	不支持	不支持	支持
offline	支持	支持	不支持
oids	支持	不支持	支持
oj	不支持	不支持	支持
old	不支持	不支持	支持
online	支持	支持	不支持
open	支持	支持	不支持
operator	支持	不支持	支持
optional	不支持	不支持	支持
ordinality	支持	支持	支持
organization	支持	不支持	支持
optionally	支持	不支持	不支持
options	支持	不支持	不支持
others	不支持	不支持	支持
out	支持	不支持	不支持
outfile	支持	支持	不支持
over	支持	不支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
overlaps	支持	不支持	不支持
overlay	支持	不支持	不支持
owned	支持	不支持	支持
owner	支持	不支持	不支持
ordinality	支持	不支持	不支持
package	支持	不支持	不支持
packages	支持	不支持	不支持
parser	支持	不支持	不支持
partition	支持	不支持	不支持
partitioning	支持	支持	不支持
partitions	支持	不支持	不支持
passing	支持	不支持	不支持
password_lock_time	不支持	不支持	支持
path	不支持	不支持	支持
per	支持	不支持	不支持
percent	支持	不支持	不支持
perm	支持	不支持	不支持
percent_rank	不支持	不支持	支持
performance	支持	支持	不支持
persist	不支持	不支持	支持
persist_only	不支持	不支持	支持
pivot	支持	支持	不支持
pluggable	支持	支持	不支持
plan	支持	支持	支持
plans	支持	不支持	不支持
policy	支持	不支持	不支持
pool	支持	不支持	不支持
populate	支持	支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
position	不支持	不支持	支持
preceding	支持	不支持	支持
predict	支持	不支持	不支持
preferred	支持	不支持	不支持
prefix	支持	不支持	不支持
prepared	不支持	不支持	支持
preserve	支持	不支持	不支持
priority	支持	支持	不支持
private	支持	不支持	不支持
procedural	支持	不支持	不支持
profile	支持	不支持	不支持
publication	支持	不支持	不支持
privilege_checks_user	不支持	不支持	支持
process	不支持	不支持	支持
public	支持	支持	不支持
publish	支持	不支持	不支持
purge	支持	不支持	不支持
quarter	支持	支持	不支持
query	支持	不支持	不支持
quote	支持	不支持	不支持
random	不支持	不支持	支持
randomized	支持	不支持	不支持
range	支持	不支持	不支持
rank	不支持	不支持	支持
ratio	支持	不支持	不支持
reassign	支持	不支持	不支持
rebuild	支持	不支持	不支持
recheck	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
recover	支持	支持	不支持
recursive	支持	不支持	支持
recyclebin	支持	不支持	支持
redisanyvalue	支持	不支持	不支持
ref	支持	不支持	不支持
refresh	支持	不支持	不支持
reference	不支持	不支持	支持
regexp	支持	支持	不支持
regexp_like	支持	支持	不支持
registration	不支持	不支持	支持
reindex	支持	不支持	支持
reject	不支持	不支持	支持
relative	支持	不支持	支持
reloptions	支持	不支持	不支持
remote	支持	不支持	不支持
remove	支持	不支持	不支持
repeatable	支持	不支持	不支持
replace	支持	不支持	不支持
replica	支持	不支持	支持
replica_type	支持	不支持	不支持
replicas	不支持	不支持	支持
require_row_format	不支持	不支持	支持
resize	支持	不支持	不支持
resource	不支持	不支持	支持
respect	支持	支持	支持
restart	支持	不支持	支持
return	支持	不支持	不支持
returns	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
retain	不支持	不支持	支持
returning	不支持	不支持	支持
reuse	支持	不支持	支持
right	支持	不支持	不支持
rlike	支持	支持	不支持
role	支持	不支持	支持
roles	支持	不支持	不支持
rollup	支持	不支持	不支持
rotation	支持	不支持	不支持
row	支持	不支持	不支持
rows	支持	不支持	不支持
row_number	不支持	不支持	支持
rowtype	支持	不支持	不支持
rule	支持	不支持	不支持
sample	支持	不支持	不支持
scale	支持	支持	不支持
schedule	支持	支持	不支持
scroll	支持	不支持	支持
search	支持	不支持	不支持
second	支持	不支持	不支持
second_microsecond	支持	支持	不支持
security	支持	不支持	不支持
secondary	不支持	不支持	支持
secondary_engine	不支持	不支持	支持
secondary_engine_attribute	不支持	不支持	支持
secondary_load	不支持	不支持	支持
secondary_unload	不支持	不支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
separator	支持	支持	不支持
sequence	支持	不支持	支持
sequences	支持	不支持	不支持
serializable	支持	不支持	不支持
server	支持	不支持	不支持
sessiontimezone	支持	支持	不支持
setof	支持	不支持	支持
sets	支持	不支持	不支持
sessiontimezone	支持	不支持	不支持
shippable	支持	不支持	支持
show	支持	不支持	不支持
shrink	支持	支持	不支持
siblings	支持	不支持	不支持
signed	支持	支持	不支持
similar	支持	不支持	不支持
simple	支持	不支持	不支持
size	不支持	不支持	支持
skip	支持	不支持	支持
slave	支持	支持	不支持
slice	支持	不支持	支持
smalldatetime	支持	不支持	支持
smalldatetime_format	支持	不支持	支持
smartroute	支持	支持	不支持
source	支持	不支持	不支持
source_auto_position	不支持	不支持	支持
source_bind	不支持	不支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
source_compression_algorithms	不支持	不支持	支持
source_connect_retry	不支持	不支持	支持
source_delay	不支持	不支持	支持
source_heartbeat_period	不支持	不支持	支持
source_host	不支持	不支持	支持
source_log_file	不支持	不支持	支持
source_log_pos	不支持	不支持	支持
source_password	不支持	不支持	支持
source_port	不支持	不支持	支持
source_public_key_path	不支持	不支持	支持
source_retry_count	不支持	不支持	支持
source_ssl	不支持	不支持	支持
source_ssl_ca	不支持	不支持	支持
source_ssl_capath	不支持	不支持	支持
source_ssl_cert	不支持	不支持	支持
source_ssl_cipher	不支持	不支持	支持
source_ssl_crl	不支持	不支持	支持
source_ssl_crlpath	不支持	不支持	支持
source_ssl_key	不支持	不支持	支持
source_ssl_verify_server_cert	不支持	不支持	支持
source_tls_ciphersuites	不支持	不支持	支持
source_tls_version	不支持	不支持	支持
source_user	不支持	不支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
source_zstd_compression_level	不支持	不支持	支持
space	支持	不支持	不支持
specification	支持	支持	不支持
spill	支持	不支持	不支持
split	支持	不支持	支持
srid	不支持	不支持	支持
stable	支持	不支持	支持
standalone	支持	不支持	不支持
starting	支持	支持	不支持
starts	支持	支持	不支持
statement	支持	不支持	不支持
statement_id	支持	不支持	不支持
statistics	支持	不支持	不支持
stdin	支持	不支持	支持
stdout	支持	不支持	支持
storage	支持	不支持	不支持
store	支持	不支持	不支持
stored	支持	不支持	不支持
str	支持	支持	不支持
stratify	支持	不支持	不支持
stream	不支持	不支持	支持
strict	支持	不支持	支持
strip	支持	不支持	不支持
subdate	不支持	支持	不支持
subpartition	支持	不支持	不支持
subpartitioning	支持	支持	不支持
subpartitions	支持	支持	不支持
subscription	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
substr	支持	支持	不支持
substring	支持	不支持	支持
sys_refcursor	支持	不支持	不支持
sysid	支持	不支持	不支持
sysdate	不支持	不支持	支持
system	不支持	不支持	支持
table_compress_clause	支持	支持	不支持
thread_priority	不支持	不支持	支持
ties	不支持	不支持	支持
timestampadd	支持	支持	不支持
tls	不支持	不支持	支持
trim	不支持	不支持	支持
turbo	支持	支持	不支持
table_compress_clause	支持	不支持	不支持
tables	支持	不支持	不支持
tablesample	支持	不支持	不支持
tablespace	支持	不支持	不支持
target	支持	不支持	不支持
temp	支持	不支持	不支持
template	支持	不支持	不支持
temporary	支持	不支持	不支持
terminated	支持	不支持	不支持
than	支持	不支持	不支持
time_format	支持	不支持	不支持
timecapsule	支持	不支持	不支持
timestampadd	不支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
timestamp_format	支持	不支持	不支持
timestampdiff	支持	不支持	不支持
transform	支持	不支持	不支持
treat	支持	不支持	不支持
trim	支持	不支持	不支持
trusted	支持	不支持	不支持
tsfield	支持	不支持	不支持
tstag	支持	不支持	不支持
tstime	支持	不支持	不支持
turbo	支持	不支持	不支持
types	支持	不支持	不支持
unbounded	支持	不支持	支持
uncommitted	支持	不支持	不支持
unencrypted	支持	不支持	不支持
unit	支持	支持	不支持
unknown	支持	不支持	不支持
unlimited	支持	不支持	不支持
unlisten	支持	不支持	不支持
unlock	支持	不支持	不支持
unlogged	支持	不支持	不支持
unpivot	支持	支持	不支持
unregister	不支持	不支持	支持
unsigned	支持	支持	不支持
until	支持	不支持	不支持
unusable	支持	不支持	支持
url	不支持	不支持	支持
useeof	支持	不支持	不支持
vacuum	不支持	不支持	支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
valid	支持	不支持	支持
validate	支持	不支持	不支持
validation	支持	不支持	不支持
validator	支持	不支持	不支持
varchar2	不支持	不支持	支持
variables	支持	不支持	不支持
varying	支持	不支持	不支持
vcgroup	支持	不支持	不支持
vcpu	不支持	不支持	支持
verbose	支持	不支持	支持
verify	支持	支持	不支持
version	支持	不支持	支持
visible	支持	支持	支持
volatile	支持	不支持	支持
wait	支持	不支持	不支持
weak	支持	不支持	不支持
week	支持	支持	不支持
wellformed	支持	支持	不支持
whitespace	支持	不支持	不支持
window	不支持	不支持	支持
within	支持	不支持	支持
without	支持	不支持	不支持
workload	支持	不支持	不支持
wrapper	支持	不支持	不支持
xmlattributes	支持	不支持	不支持
xmlconcat	支持	不支持	不支持
xmlelement	支持	不支持	不支持
xmlexists	支持	不支持	不支持

版权所有 © 华为云计算技术有限公司

关键字	ORA兼容模式数据库	MYSQL兼容模式数据库、TD兼容模式数据库、PG兼容模式数据库	M-Compatibility模式数据库
xmlforest	支持	不支持	不支持
xmlnamespaces	支持	支持	不支持
xmlparse	支持	不支持	不支持
xmlpi	支持	不支持	不支持
xmlroot	支持	不支持	不支持
xmltable	支持	支持	不支持
xmlquery	支持	支持	不支持
xmlcast	支持	支持	不支持
xmlserialize	支持	不支持	不支持
xmltype	支持	支持	不支持
year_month	支持	支持	不支持
years	支持	支持	不支持
zerofill	支持	支持	不支持
zone	支持	不支持	支持

默认值：

- 非M-Compatibility模式数据库下："datetime,regexp,rlike,zerofill"。
- M-Compatibility模式数据库下：""。

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数打开后，作为关键字使用的部分功能会失效，请谨慎使用。如果要屏蔽的关键字在当前兼容模式数据库的待屏蔽关键字数组里不存在，其他兼容模式数据库里存在时，则产生WARNING；如果其他兼容模式数据库里也不存在则产生ERROR。

说明

MYSQL兼容模式下，datetime、regexp、rlike、zerofill是否作为保留关键字由系统确定，用户无法修改和删除。具体情况为：初始状态下系统会将datetime、regexp、rlike、zerofill添加到参数disable_keyword_options中，此时datetime、regexp、rlike、zerofill不作为保留关键字；设置set b_format_version = '5.7'和set b_format_dev_version = 's1';（或者高于s1的版本）后，系统会将datetime、regexp、rlike、zerofill从参数disable_keyword_options中删除，此时可作为保留关键字使用。

版权所有 © 华为云计算技术有限公司

disable_plsql_keyword_options

参数说明：用于将设置的标识符当作非关键字使用。数据库兼容性行为配置项，该参数的值由若干个配置项用逗号隔开构成。

参数类型：字符串

参数单位：无

取值范围：支持的配置项包括PIPE、PIPELINED、RANGE、REPLACE、SUBTYPE。

默认值：""

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 该参数打开后，作为关键字使用的部分功能会失效，请谨慎使用。
- 如需取消屏蔽plsql关键字功能，请将该参数设置为空值。

td_compatible_truncation

参数说明：控制是否开启与Teradata数据库相应兼容的特征。在用户连接上与TD兼容的数据库时，可以将该参数设置成为on（即超长字符串自动截断功能启用），该功能启用后，在后续的insert语句中，对目标表中char和varchar类型的列插入超长字符串时，会按照目标表中相应列定义的最大长度对超长字符串进行自动截断。保证数据都能插入目标表中，而不是报错。

说明

- 超长字符串自动截断功能不适用于insert语句包含外表的场景。
- 如果向字符集为字节类型编码（SQL_ASCII，LATIN1等）的数据库中插入多字节字符数据（如汉字等），且字符数据跨越截断位置，这种情况下，按照字节长度自动截断，自动截断后会在尾部产生非预期结果。如果用户有对于截断结果正确性的要求，建议用户采用UTF8等能够按照字符截断的输入字符集作为数据库的编码集。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开超长字符串自动截断功能。
- off：表示关闭超长字符串自动截断功能。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

uppercase_attribute_name

参数说明：设置列名以大写形式返回给客户端。该参数仅限于ORA兼容模式下使用。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示列名以大写形式返回给客户端。
- off：表示列名不以大写形式返回给客户端。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 不允许通过gs_guc set、gs_guc reload、修改gaussdb.conf、alter database set、alter user set进行全局化修改本参数。
- 当开启该参数且对投影列指定了带引号的别名时，是否以大写形式返回列名还会受**uppercase_attribute_name_respect_quote**参数的控制。
- 当开启该参数后，仅当列名为全小写时，会将其转为全大写返回给客户端。当列名为大小写混用时，不会对列名转大写。

```
gaussdb=# SET uppercase_attribute_name=off;
SET
gaussdb=# SELECT 1 AS ab, 2 AS Ab, 3 AS AB, 4 AS "ab", 5 AS "Ab", 6 AS "AB";
 ab | ab | ab | ab | Ab | AB
----+----+----+----+----+----
  1 |  2 |  3 |  4 |  5 |  6
(1 row)

gaussdb=# SET uppercase_attribute_name=on;
SET
gaussdb=# SELECT 1 AS ab, 2 AS Ab, 3 AS AB, 4 AS "ab", 5 AS "Ab", 6 AS "AB";
 AB | AB | AB | AB | Ab | AB
----+----+----+----+----+----
  1 |  2 |  3 |  4 |  5 |  6
(1 row)
```

uppercase_attribute_name_respect_quote

参数说明：开启uppercase_attribute_name参数后，控制带引号的列别名，是否保留原始大小写形式返回客户端。该参数仅限于ORA兼容模式下使用。该参数可在PDB级别设置。

参数类型：枚举类型

参数单位：无

取值范围：

- off：带引号且全小写的列别名转为大写返回，带引号且全大写或大小写混用的列别名保留原始大小写返回。
- select_alias：对于最外层为SELECT查询且投影列指定了带引号别名的场景，保留原始大小写返回；其余场景行为与off一致。

默认值：off。在PDB场景内，若未设置该参数，则继承来自全局的设置。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

 版权所有 © 华为云计算技术有限公司

说明

- 当从不支持该参数的版本升级到当前版本时，升级未提交情况下设置该参数不生效。
- 不允许通过gs_guc set、gs_guc reload、修改gaussdb.conf、alter database set、alter user set进行全局化修改本参数。
- 该参数仅当开启uppercase_attribute_name参数后才生效，当设置该参数为select_alias后，仅影响SELECT查询语句，不影响PBE、CREATE VIEW、DECLARE CURSOR等场景。

```
gaussdb=# SET uppercase_attribute_name=on;
SET
gaussdb=# SET uppercase_attribute_name_respect_quote=off;
SET
gaussdb=# SELECT 1 AS "a";
 A
---
 1
(1 row)

gaussdb=# SET uppercase_attribute_name_respect_quote=select_alias;
SET
gaussdb=# SELECT 1 AS "a";
 a
---
 1
(1 row)
gaussdb=# PREPARE p1 AS SELECT 1 AS "a";
PREPARE
gaussdb=# EXECUTE p1;
 A
---
 1
(1 row)

gaussdb=# CREATE VIEW v1 AS SELECT 1 AS "a";
CREATE VIEW
gaussdb=# SELECT * FROM v1;
 A
---
 1
(1 row)

gaussdb=# CREATE OR REPLACE PROCEDURE p_test_quote(rc INOUT REFCURSOR) AS
gaussdb$# BEGIN
gaussdb$#     OPEN rc FOR SELECT 1 AS "ab";
gaussdb$# END;
gaussdb$# /
CREATE PROCEDURE
gaussdb=# BEGIN;
BEGIN
gaussdb=# CALL p_test_quote('rc');
 RC
----
 rc
(1 row)

gaussdb=# FETCH ALL FROM rc;
 AB
----
  1
(1 row)

gaussdb=# COMMIT;
COMMIT
gaussdb=# DECLARE c CURSOR FOR SELECT 1 AS "ab", 2 AS "cd";
DECLARE CURSOR
gaussdb=# FETCH 1 FROM c;
 AB | CD
----+----
```


版权所有 © 华为云计算技术有限公司

```
 1 |  2
(1 row)

gaussdb=# CLOSE c;
CLOSE CURSOR
gaussdb=# COMMIT;
COMMIT
```

enable_object_special_character

参数说明：控制执行CREATE EXTENSION语句时，若脚本文件中使用"@extschema@"，此时control文件中schema参数的值是否可以包含["$'\]中任意特殊字符。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启，可以包含["$'\]中任意特殊字符。
- off：表示关闭，不可以包含["$'\]中任意特殊字符。

默认值：

- off：新安装的数据库的默认值。
- on：该参数自505.0版本添加，505.0版本之前的数据库升级后此参数为on。升级前若存在该参数并且设置了值，升级后默认值不变。

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：扩展功能为内部使用功能，只允许操作数据库内部的高级包或扩展。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

enable_implicit_cast_str2num

参数说明：在整型数据类型与字符串数据类型进行比较操作符或者算术运算操作符运算时，控制字符串是否能够转换为NUMERIC。

说明

仅在a_format_version='10c'，a_format_dev_version='s1'时生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。

  比较操作符（>、>=、<、<=、=、<>）：字符串数据类型会被隐式转换成NUMERIC类型（在未开启**behavior_compat_options**参数的enable_crosstype_integer_operator选项时，整型数据类型也会被隐式转换成NUMERIC类型）。

  算术操作符（+、-、*、/、%、^）：整型数据类型和字符串数据类型都会被隐式转换成NUMERIC类型。

版权所有 © 华为云计算技术有限公司

- off：表示关闭。字符串数据类型会被隐式转换成整型数据类型。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

a_format_disable_func

参数说明：禁用指定的系统函数。该参数的值由若干个系统函数OID值用逗号隔开组成。设置该参数的系统函数将不能被调用。当某一系统函数不能满足用户需求并且需要使用相同函数名的自定义函数时，可使用该功能。

参数类型：字符串

参数单位：无

取值范围：字符串，由若干个系统函数OID值用逗号隔开组成。

默认值：""

设置方式：该参数属于SUSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_convert_illegal_char

参数说明：对输出结果中的非法编码是否不进行校验并以占位符替换显示。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启，非法编码将替换为**convert_illegal_char_mode**参数指定符号的编码。
- off：表示关闭，查询结果中包含的非法编码会校验后报错。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，仅在数据中存在特殊字符且需要对特殊字符进行强制不报错处理时开启该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

1. 当数据库字符集为utf8、zhs16gbk、gb18030、gb18030-2022、latin1，enable_convert_illegal_char=on时生效，且当数据库客户端与服务端字符集不相同时，非法编码将以占位符形式显示。
2. 特殊字符范围：本文所指的特殊字符范围仅包括完全异常编码和混合编码，不支持\u0000字符。若字符编码中包含\u0000字符，在使用时通常会在\u0000处截断，无法保证数据的完整性。
3. 在PBE场景下，当数据库客户端与服务端字符集相同时，绑定参数无需编码转换，但仍需校验合法性，并在校验时对绑定参数中的非法字符进行处理。当关闭enable_convert_illegal_char参数时，将绑定参数中的非法字符（不包含\u0000）替换成占位符（**convert_illegal_char_mode**参数指定的字符）。当开启enable_convert_illegal_char参数时，保留原字符。
4. 若数据库客户端与服务端字符集不相同，在enable_convert_illegal_char=off时，数据库将超出当前字符集编码外的特殊字符（不包含\x0000字符）替换为\x3F（即显示为?）编码写入数据库中；当enable_convert_illegal_char=on时，只能使用函数dbe_raw.cast_to_varchar2()将非法编码原编码插入数据库中，并以占位符形式显示。
5. 若数据库客户端与服务端字符集相同，当enable_convert_illegal_char=off时，数据库将超出当前字符集编码外的特殊字符（不包含\x0000字符）替换为\x3F（即显示为?）编码写入数据库中；当enable_convert_illegal_char=on时，只能使用函数dbe_raw.cast_to_varchar2()将非法编码原编码插入数据库中。
6. 仅支持客户端字符集与服务端字符集之间的编码处理，非客户端字符集与服务端字符集之间的编码校验与关闭参数时行为一致。
7. 当开启GUC参数时，特殊字符在下表所列的函数、高级包函数中的表现为：
   - 在客户端与服务端字符集一致时，特殊字符在查询时不报错，与开启GUC参数前的行为保持一致。
   - 在客户端与服务端字符集不一致时，特殊字符在查询时显示为占位符（默认为?）。
   - 不建议客户使用函数处理包含特殊字符的字符串，下表所列函数处理包含特殊字符的字符串不会产生报错，但不保证结果的正确性。

表 6-39 特殊字符支持的函数及高级包函数

序号	函数名/高级包函数名
1	bit_length(string)
2	btrim(string text [, characters text])
3	char_length(string) character_length(string)
4	chr(cvalue int|bigint) chr(integer)
5	concat(str1,str2)
6	concat_ws(sep text, str"any" [, str"any" [, ...]])
7	decode(string text, format text)
8	dump(expr[, return_fmt [, start_position [, length]]])
9	encode(data bytea, format text)
10	find_in_set(text, set)

版权所有 © 华为云计算技术有限公司

序号	函数名/高级包函数名
11	format(formatstr text [, str"any" [, ...]])
12	left(str text, n int)
13	length(string)
14	lengthb(text/bpchar)
15	ltrim(string [, characters])
16	md5(string)
17	notlike(x bytea name text, y bytea text)
18	octet_length(string)
19	overlay(string placing string FROM int [for int])
20	quote_ident(string text)
21	quote_literal(string text)
22	quote_nullable(string text)
23	rawcat(raw,raw)
24	regexp_count(string text, pattern text [, position int [, flags text]])
25	regexp_instr(string text, pattern text [, position int [, occurrence int [, return_opt int [, flags text]]]])
26	regexp_like(source_string, pattern [, match_parameter]) regexp_like(text,text,text)
27	regexp_matches(string text, pattern text [, flags text])
28	regexp_replace(string, pattern, replacement [,flags]) regexp_replace(string text, pattern text [, replacement text [, position int [, occurrence int [, flags text]]]])
29	regexp_split_to_array(string text, pattern text [, flags text])
30	regexp_split_to_table(string text, pattern text [, flags text])
31	regexp_substr(source_char, pattern) regexp_substr(string text, pattern text [, position int [, occurrence int [, flags text]]])
32	repeat(string text, number int)
33	replace(string text, from text, to text)
34	replace(string, substring)
35	reverse(str)

版权所有 © 华为云计算技术有限公司

序号	函数名/高级包函数名
36	right(str text, n int)
37	rtrim(string [, characters])
38	sha(string)
39	sha1(string)
40	sha2(string，hash_length)
41	split_part(string text, delimiter text, field int)
42	substring(string [from int] [for int]) substring(string from pattern for escape) substring(string from pattern)
43	substring_inner(string [from int] [for int])
44	tconvert(key text, value text)
45	to_single_byte(char)
46	translate(string text, from text, to text)
47	trim([leading |trailing |both] [characters] from string)
48	unistr(string)
49	vsize(expr)
50	PKG_UTIL.RAW_CAST_FROM_VARCHAR2
51	PKG_UTIL.LOB_CONVERTTOCLOB
52	PKG_UTIL.LOB_RAWTOTEXT
53	PKG_UTIL.LOB_TEXTTORAW
54	PKG_UTIL.RAW_CAST_TO_VARCHAR2
55	DBE_OUTPUT.PUT
56	DBE_OUTPUT.PUT_LINE

fix_func_selection

参数说明：控制是否优化函数匹配策略。

catlist顺序问题：若用户自定义函数与系统函数存在冲突，数据库系统在选择函数时，会受该系统函数在数据库系统中的注册顺序影响，从而导致选择函数结果不同。

参数类型：字符串

参数单位：无

取值范围：""、"catlist"。

版权所有 © 华为云计算技术有限公司

- ""：不进行优化，与505.1.0版本之前的版本保持一致。
- "catlist"：优化catlist顺序问题，始终优先选择系统函数并执行。

默认值：

- "catlist"：新安装的数据库的默认值。
- ""：505.1.0版本之前的数据库升级后的默认值。

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_allowed_packet

参数说明：M-Compatibility模式数据库行为配置项，用于表示字符串函数返回值上限。若实际结果超过该上限，则会告警，并返回NULL值。

参数类型：整型

参数单位：Byte

取值范围：1024 ~ 1073741824（取值只能是1024的倍数，非倍数时将向下舍入到最接近的倍数）。

默认值：4194304

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

div_precision_increment

参数说明：M-Compatibility模式数据库行为配置项。用于设置除法结果的精度提升值，即除法运算结果的小数位数等于被除数的小数位数加上该参数值。

参数类型：整型

参数单位：无

取值范围：0 ~ 30

默认值：4

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_m_format_hook

参数说明：M-Compatibility模式数据库行为配置项。该参数控制M-Compatibility模式hook是否生效。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示M-Compatibility模式数据库允许挂载对应解析执行hook。
- off：表示M-Compatibility模式数据库不挂载对应解析执行hook。

默认值：on

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：当前参数仅针对当前未完全适配的外部工具使用，不允许用户使用。

设置不当的风险与影响：无

gs_format_behavior_compat_options

参数说明：gs_format_behavior_compat_options用于选择GaussDB内部系统函数配置项。

参数类型：字符串

参数单位：无

取值范围：当前只支持**表6-40**中的兼容性配置项，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开。

默认值：""

表 6-40 gs_format_behavior 兼容性配置项

兼容性配置项	兼容性行为控制
sqrt_karatsuba	• 设置此参数：表示在调用sqrt平方根函数时，使用Karatsuba平方根算法，Karatsuba算法性能更高但是在极少数情况下精度可能会和牛顿迭代算法不一致。 • 不设置此参数：在计算sqrt平方根算法时，会使用默认的牛顿迭代算法进行平方根的计算。
allow_textconcat_null	• 设置此参数：在PG兼容模式下，字符串和null值拼接会返回对应的字符串的值。 -- 在PG兼容模式下执行 gaussdb=# set gs_format_behavior_compat_options='allow_textconcat_null'; SET gaussdb=# select 'a' || null || 'b'; ?column? ---------- ab (1 row) • 不设置此参数：在PG兼容模式下，字符串和null值拼接会返回NULL。

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：控制一些兼容性特性是否可用，若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

support_zero_character

参数说明：当数据中包含Unicode编码为\u0000的字符（以下简称空字符）时，数据库是否取消对空字符的校验并且正确处理而不进行截断操作。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启，数据库取消对空字符的校验，能够正确插入和处理空字符，使数据在处理过程中不发生截断。
- off：表示关闭，数据库保留对空字符的校验，部分函数在处理空字符时会发生报错或数据截断。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，仅在数据中存在空字符且需要对空字符进行正确处理时开启该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

- 该参数仅在数据库兼容性为ORA兼容模式且所有节点都设置为support_zero_character=on时生效。
- 如果需要向数据库中写入空字符，需确认support_zero_character=on生效，空字符数据才能正常插入并存储。
- 当support_zero_character=on时，如果数据类型为CHAR(n)、CHARACTER(n)、NCHAR(n)、VARCHAR(n)、CHARACTER VARYING(n)、VARCHAR2(n)、NVARCHAR2(n)、TEXT或CLOB类型，数据库可以正确插入并处理空字符；如果包含空字符的字符串数据可以显式转换为JSON/JSONB、XML、XMLTYPE、数值类型、货币类型、时间类型、几何类型、网络地址类型、位串类型、文本搜索类型、HLL类型、范围类型、对象标识符类型、账本数据库类型、aclitem类型或向量类型时，会在空字符处发生截断；如果字符串类型显式转换为UUID类型时，会在空字符处发生报错。
- 数据库仅支持以下写入空字符的方式：
  - 通过JDBC连接数据库时，空字符可以通过文本方式写入数据库。
  - 通过dbe_raw.cast_to_varchar2函数直接将空字符编码写入数据库。
  - 通过copy工具将空字符编码写入数据库。
- 在开启GUC参数时，不同字符集之间进行转换时，空字符会直接返回，不再发生报错。
- gsql客户端版本应与数据库版本匹配，在开启GUC参数时，若gsql端接收的数据中包含空字符，gsql工具会将空字符转换为空格后进行打印输出。若使用505.2.0之前版本的gsql工具连接数据库，包含空字符的数据仍存在显示被截断的情况。
- 当开启GUC参数时，空字符在**表 空字符支持的系统函数和高级包函数列表**所列的函数、高级包函数中的表现为：函数能够正常处理空字符，不会发生在空字符处截断的情况。
- 部分高级包接口入参为Schema名或object名，理论上不能含有空字符，因此对**表 入参包含空字符进行报错处理的函数**中的函数添加空字符校验，当入参包含空字符时进行报错处理（原表现为空字符后数据被截断）。
- 当support_zero_character=on时，若处理的数据中不包含空字符，函数性能无明显劣化；若处理的数据中包含空字符，对比相同数量的有效字符时，initcap、upper、lower、nls_upper和nls_lower函数性能劣化程度与数据中空字符个数紧密相关，即空字符个数越多，性能劣化越严重。
- 当support_zero_character=on时，若逻辑解码工具读取到数据中包含的空字符，则会将空字符转换为空格后进行打印输出（原表现为空字符后数据被截断）。

表 6-41 空字符支持的系统函数和高级包函数列表

序号	函数名/高级包函数名
1	btrim(string text [, characters text])
2	char_length(string) 或 character_length(string)
3	chr(cvalue int|bigint)
4	chr(integer)
5	coalesce(expr1, expr2, ..., exprn)
6	concat(str1,str2)
7	concat_ws(sep text, str"any" [, str"any" [, ...]])
8	convert(string bytea, src_encoding name, dest_encoding name)
9	convert_from(string bytea, src_encoding name)

版权所有 © 华为云计算技术有限公司

序号	函数名/高级包函数名
10	convert_to(string text, dest_encoding name)
11	decode(base_expr, compare1, value1, Compare2,value2, … default)
12	dump(expr[, return_fmt [, start_position [, length]]])
13	encode(data bytea, format text)
14	format(formatstr text [, str"any" [, ...]])
15	greatest(expr1 [, ...])
16	group_concat([DISTINCT | ALL] expression [,expression ...] [ORDER BY { expression [[ASC | DESC | USING operator] | nlssort_expression_clause] [NULLS { FIRST | LAST }] } [,...]] [SEPARATOR str_val])
17	initcap(string)
18	instr(string,substring[,position,occurrence]) instr(text,text,int,int)
19	instrb(text,text,int,int)
20	least(expr1 [, ...])
21	left(str text, n int)
22	length(string) length(string bytea, encoding name)
23	lengthb(text/bpchar)
24	listagg(expression [, delimiter]) WITHIN GROUP(ORDER BY order-list)
25	lower(string)
26	lpad(string text, length int [, fill text])
27	ltrim(string [, characters])
28	nls_lower(string [, nlsparam])
29	nls_upper(string [, nlsparam])
30	nullif(expr1, expr2)
31	nvl(expr1 , expr2)
32	nvl2(expr1 , expr2， expr3)
33	overlay(string placing string FROM int [for int])
34	position(substring in string)

版权所有 © 华为云计算技术有限公司

序号	函数名/高级包函数名
35	quote_nullable(string text)
36	regexp_count(string text, pattern text [, position int [, flags text]])
37	regexp_instr(string text, pattern text [, position int [, occurrence int [, return_opt int [, flags text]]]])
38	regexp_like(text,text,text) regexp_like(source_string, pattern [, match_parameter])
39	regexp_matches(string text, pattern text [, flags text])
40	regexp_replace(string, pattern, replacement [,flags])
41	regexp_split_to_array(string text, pattern text [, flags text])
42	regexp_split_to_table(string text, pattern text [, flags text])
43	regexp_substr(source_char, pattern) regexp_substr(string text, pattern text [, position int [, occurrence int [, flags text]]])
44	repeat(string text, number int)
45	replace(string text, from text, to text) replace(string, substring)
46	reverse(str)
47	right(str text, n int)
48	rpad(string text, length int [, fill text])
49	rtrim(string [, characters])
50	similar_escape(pat text, esc text)
51	split_part(string text, delimiter text, field int)
52	string [NOT] LIKE pattern [ESCAPE escape-character]
53	string || string 或 string || non-string 或 non-string || string
54	string_agg(expression, delimiter)
55	strpos(string, substring)
56	substr(string,from) substr(bytea,from,count) substr(string,from,count)

版权所有 © 华为云计算技术有限公司

序号	函数名/高级包函数名
57	substrb(text,int) substrb(text,int,int)
58	substring(string [from int] [for int]) substring(string from pattern for escape) substring(string from pattern) substring_inner(string [from int] [for int])
59	translate(string text, from text, to text)
60	trim([leading |trailing |both] [characters] from string)
61	unistr(string)
62	upper(string)
63	PKG_UTIL.LOB_GET_LENGTH
64	PKG_UTIL.LOB_READ
65	PKG_UTIL.LOB_WRITE
66	PKG_UTIL.LOB_APPEND
67	PKG_UTIL.LOB_COMPARE
68	PKG_UTIL.LOB_MATCH
69	PKG_UTIL.LOB_READ_HUGE
70	PKG_UTIL.LOB_WRITEAPPEND_HUGE
71	PKG_UTIL.LOB_APPEND_HUGE
72	PKG_UTIL.LOB_COPY_HUGE
73	PKG_UTIL.LOB_CONVERTTOBLOB_HUGE
74	PKG_UTIL.IO_PRINT
75	PKG_UTIL.RAW_CAST_FROM_VARCHAR2
76	PKG_UTIL.FILE_READ
77	PKG_UTIL.FILE_READLINE
78	PKG_UTIL.APP_READ_CLIENT_INFO
79	PKG_UTIL.APP_SET_CLIENT_INFO
80	PKG_UTIL.LOB_CONVERTTOBLOB
81	PKG_UTIL.LOB_CONVERTTOCLOB
82	PKG_UTIL.MATCH_EDIT_DISTANCE_SIMILARITY
83	PKG_UTIL.RAW_CAST_TO_VARCHAR2

版权所有 © 华为云计算技术有限公司

序号	函数名/高级包函数名
84	PKG_UTIL.APP_SET_MODULE
85	PKG_UTIL.APP_READ_MODULE
86	PKG_UTIL.APP_SET_ACTION
87	PKG_UTIL.APP_READ_ACTION
88	DBE_OUTPUT.PRINT_LINE
89	DBE_OUTPUT.PRINT
90	DBE_OUTPUT.GET_LINE
91	DBE_OUTPUT.GET_LINES
92	DBE_OUTPUT.PUT
93	DBE_OUTPUT.PUT_LINE
94	DBE_UTILITY.CANONICALIZE
95	DBE_UTILITY.COMMA_TO_TABLE
96	DBE_UTILITY.NAME_TOKENIZE
97	DBE_UTILITY.TABLE_TO_COMMA
98	DBE_UTILITY.CANONICALIZE_RET
99	DBE_UTILITY.COMMA_TO_TABLE_FUNC
100	DBE_UTILITY.NAME_SEPARATE
101	DBE_UTILITY.NAME_TOKENIZE_FUNC
102	DBE_UTILITY.NAME_TOKENIZE_LOWER
103	DBE_UTILITY.NAME_TOKENIZE_LOWER_FUNC
104	DBE_UTILITY.TABLE_TO_COMMA_FUNC

表 6-42 入参包含空字符进行报错处理的函数

序号	高级包函数名
1	PKG_UTIL.SESSION_SET_CONTEXT
2	PKG_UTIL.UTILITY_COMPILE_SCHEMA
3	PKG_UTIL.GS_COMPILE_SCHEMA
4	DBE_UTILITY.NAME_RESOLVE
5	DBE_UTILITY.COMPILE_SCHEMA

版权所有 © 华为云计算技术有限公司

序号	高级包函数名
6	DBE_UTILITY.SEARCH_CLASS_WITH_NSPOID_ONAME_TYPE
7	DBE_UTILITY.SEARCH_OBJECTS
8	DBE_UTILITY.SEARCH_OBJECTS_SYNONYM_FILL_SCHEMA
9	DBE_UTILITY.SEARCH_PROCEDURE_WITH_NSPOID_ONAME
10	DBE_UTILITY.SEARCH_SYNONYM_WITH_NSPOID_ONAME
11	PG_CATALOG.GS_COMPILE_SCHEMA

enable_case_intervaltonumeric

参数说明：控制CASE语句中是否支持INTERVAL转换为NUMERIC，该参数仅在ORA兼容模式下生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：CASE语句中支持INTERVAL转换为NUMERIC。
- off：CASE语句中不支持INTERVAL转换为NUMERIC。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为on，后续修改为off，可能导致现有CASE语句报错。

enable_ecpg_cursor_duplicate_operation

参数说明：ECPG控制在ORA兼容下是否支持重复打开/关闭游标，仅在ECPG连接ORA兼容数据库时生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：ECPG连接ORA兼容数据库时，支持重复打开/关闭游标。
- off：ECPG连接ORA兼容数据库时，不支持重复打开/关闭游标。

默认值：on

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为off，重复打开/关闭游标时会报错。

说明

直接安装当前版本数据库时，该参数默认值为on。从历史版本升级上来的数据库，若升级前的基础版本不支持该参数，则升级后该参数默认值为off；若升级前的基础版本支持该参数，升级后保持该参数当前设置。

format_pg_to_timestamp

参数说明：在PG模式下，控制to_timestamp(string [, fmt])函数的行为。开启后，如果有不支持的格式，自动跳过，当前版本不支持FF，FF7，FF8，FF9，RR，RRRR，SYYYY，x，X等格式；支持早于294277-01-01 00:00:00之前的时间日期和负数年份（转化成公元前）；FF1-FF6会把超出精确度的毫秒四舍五入；当时间日期中相同项冲突时（例如DDD和DD所指的天数不同时）返回一个格式正确的时间不会报错，具体冲突行为如表6-43所示。

参数类型：布尔型

参数单位：无

取值范围：

- on：实现上述功能。
- off：取消实现上述功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启会导致GaussDB不支持的fmt失效。

表 6-43 to_timestamp fmt 冲突

模式	描述
SSSSS与时间冲突	保留时间（分钟，小时）中的值以及SSSSS中的秒数部分的值。
W，J与时间日期冲突	优先级：J，W，mmdd按顺序进行覆盖。 W：未指定时，默认为星期三。 默认行为：W覆盖mmdd。
SSSSS与SS冲突	保留SS，舍弃SSSSS。
DDD与日期冲突	舍弃DDD。

synonym_priv_strict_check

参数说明：控制通过同义词搜索对象时，是否检查同义词关联对象的访问权限。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：通过同义词检索对象时，检查是否具有同义词关联对象的访问权限。
- off：通过同义词检索对象时，不检查是否具有同义词关联对象的访问权限。

默认值：

- 直接安装新版本时，默认值为on。
- 从历史版本升级上来的数据库，如果升级的基础版本不支持该GUC参数，默认值为off；如果升级的基础版本支持该GUC参数，默认值为on。

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为off，通过同义词访问对象时，不检查对象的访问权限，可能构成越权风险。

forbid_public_funcname_same_with_sysfunc

参数说明：控制在创建函数及RENAME函数时，是否禁止PUBLIC函数与pg_catalog下的系统函数名称相同。

参数类型：布尔型

参数单位：无

取值范围：

- on：创建函数及RENAME函数时，禁止PUBLIC函数与pg_catalog下的系统函数名称相同。
- off：创建函数及RENAME函数时，允许PUBLIC函数与pg_catalog下的系统函数名称相同。

默认值：

- 直接安装新版本时，默认值为on。
- 从历史版本升级上来的数据库，如果升级的基础版本不支持该GUC参数，默认值为off；如果升级的基础版本支持该GUC参数，默认值为on。

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为off，允许PUBLIC函数与pg_catalog下的系统函数名称相同。

a_format_func_col_name

参数说明：数据库ORA兼容模式系统函数兼容性行为配置项。

参数类型：字符串

参数单位：无

取值范围：当前只支持**表6-44**的信息，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开。

版权所有 © 华为云计算技术有限公司

默认值：""（空字符串）

表 6-44 系统函数兼容性配置项

系统函数兼容性配置项	兼容性行为控制
user	开启该参数后，SELECT user;返回的投影列名为user。

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据数据库兼容对象进行调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_cursor_with_hold

参数说明：在ORA兼容模式下，控制游标是否具有WITH HOLD功能。当开启enable_cursor_with_hold后，在创建游标时，未明确指定WITHOUT HOLD关键字且SELECT子句不包含FOR UPDATE或FOR SHARE子句时，创建的游标默认行为具有WITH HOLD功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：创建游标时，未明确指定WITHOUT HOLD关键字且SELECT子句不包含FOR UPDATE或FOR SHARE子句时，创建的游标默认行为具有WITH HOLD功能。此时，CURSOR cursor_name FOR query;语义与CURSOR cursor_name WITH HOLD FOR query;等效。
- off：创建游标时，未指定WITH HOLD或WITHOUT HOLD关键字时，默认行为是WITHOUT HOLD。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。仅支持在会话中设置该参数。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为on，所有游标不会被释放，在会话中可能会存在内存膨胀的风险。

modify_function_property

参数说明：在非M兼容模式下，该参数用于控制相关系统函数的属性。

参数类型：字符串

参数单位：无

取值范围：当前只支持表6-45中的配置项，当设置多个兼容性配置项时，相邻配置项之间用逗号隔开。

版权所有 © 华为云计算技术有限公司

默认值：新安装数据库默认值为"3257,5562,4164,4073"；若升级的基础版本不存在该GUC参数，默认为""；若升级的基础版本存在该GUC参数，保留原设置。

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值，不建议修改。

设置不当的风险与影响：影响系统函数的属性。

表 6-45 系统函数配置项

系统函数对应的OID	描述
3257	修改函数pg_catalog.current_timestamp(numeric)的易变性，当该配置项存在于GUC参数中时，函数易变性为 'stable'，不存在时为 'immutable'。
4073	修改函数pg_catalog.text_timestamp(text)的易变性，当该配置项存在于GUC参数中时，函数易变性为 'stable'，不存在时为 'immutable'。
4164	修改函数pg_catalog.text_date(text)的易变性，当该配置项存在于GUC参数中时，函数易变性为 'stable'，不存在时为 'immutable'。
5562	修改函数pg_catalog.DBTimezone()的易变性，当该配置项存在于GUC参数中时，函数易变性为 'stable'，不存在时为 'immutable'。

enable_check_insertall_target

参数说明：该参数用于控制在ORA兼容模式下，是否检查insert all语法的目标表属性的合法性。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示会检查insert all语法的目标表属性的合法性：目标表不支持复制表；所有目标表的分布方式必须一致；目标表不支持带有全局二级索引（GSI）的表。
- off：表示不检查insert all语法的目标表属性的合法性。

默认值：新安装数据库为on；若升级的基础版本不存在该GUC参数，默认为off；若升级的基础版本存在该GUC参数，保留原设置。

设置方式：该参数属于USERSET类型参数，请参考表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为off，插入到目标表的数据存在不符合预期的风险。

版权所有 © 华为云计算技术有限公司

forbid_hold_cursor_with_forupdate

参数说明：用于控制存储过程中带有FOR UPDATE/FOR SHARE特性的游标循环或for loop查询语句行为，是否在执行事务提交/回滚操作后的下次循环访问游标时报错。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启参数说明中的报错行为。
- off：关闭参数说明中的报错行为。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为off，存储过程在执行事务提交/回滚操作期间可能报错，或提交/回滚后下次循环访问游标时丢失FOR UPDATE/FOR SHARE特性。

说明

直接安装当前版本数据库时，该参数默认值为on。从历史版本升级上来的数据库，若升级前的基础版本不支持该参数，则升级后该参数默认值为off；若升级前的基础版本支持该参数，升级后保持该参数当前设置。

adjust_systemview_priority

参数说明：ORA兼容模式下，开启该参数后，同名视图优先访问用户自定义的视图。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启参数，同名视图优先访问用户自定义的视图。
- off：表示关闭参数，同名视图优先访问系统视图。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若设置为on，同名视图会优先访问用户自定义的视图。

using_danger_feature

参数说明：用于控制是否开启危险特性。

参数类型：字符串

参数单位：无

取值范围：当前只支持**表6-46**中的信息，当设置多个值时，相邻的值使用逗号隔开。

默认值："none"，表示不开启危险特性。

版权所有 © 华为云计算技术有限公司

表 6-46 危险特性配置项

危险特性配置项	行为控制
operator_class	设置该选项时，允许系统管理员创建操作符类。在创建操作符类时，要求传入函数的入参类型和索引类型保持一致或能正常进行类型转换，否则会造成不可预知的后果。
alter_function	• 设置该选项时，可以对高级包所属模式、m_schema、information_schema、sqladvisor、sys、pg_toast以及dbe_sql_util中的函数，进行修改其函数名称、函数所有者以及函数所属模式的操作。 • 未设置该选项时，禁止对以上模式中的函数名称、函数所有者以及函数所属模式进行修改。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

说明

- 直接安装当前版本数据库时，该参数默认值为"none"。从历史版本升级到当前版本时，如果升级的基础版本不支持该参数，该参数默认值为"operator_class, alter_function"；如果升级的基础版本支持该参数，升级后保持该参数当前设置。
- 当"none"和其它配置项共同设置时，"none"配置项不生效。

disable_operator

参数说明：禁用指定的操作符，该参数设置的操作符将不能被调用。当某一系统操作符不能满足用户需求并且需要使用自定义操作符时，可使用该功能。

参数类型：字符串

参数单位：无

取值范围：字符串，由若干个操作符OID值用逗号隔开组成。操作符OID可以通过查询系统表**PG_OPERATOR**获得。

默认值：""

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

enable_xmltype_ignore_format

参数说明：用于控制xmltype类型的数据是否使用content格式校验。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启参数，支持xmltype类型的数据使用content格式校验。
- off：表示关闭参数，支持xmltype类型的数据使用document格式校验。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：该参数为逃生参数，非必要用户不要自行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

说明

开启enable_xmltype_ignore_format参数时，目前仅支持以下两种场景使用content格式校验输入数据：

- 从unknown、unspecified类型转换至xmltype类型，转换原理请参见**类型转换**。
- 通过调用xmltype函数，例如：xmltype('123')。

enable_force_create_obj

参数说明：用于控制视图一次性入库功能的开启与关闭。开启该参数后，创建视图，如果视图依赖的对象不存在或缺少查询权限，该视图仍能够创建成功。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开视图一次性入库功能。
- off：关闭视图一次性入库功能。

默认值：off

设置方式：该参数属于USERSET类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：根据需求场景进行配置。若无特殊情况，建议开启对应功能。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

distribute_default_compatibility

参数说明：控制CREATE DATABASE操作中不指定DBCOMPATIBILITY关键字和TEMPLATE关键字时的默认兼容模式。

参数类型：枚举类型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- ora：表示创建数据库的默认兼容模式为ORA兼容模式。
- mysql：表示创建数据库的默认兼容模式为MYSQL兼容模式。

默认值：

- 新安装的数据库：ora。
- 从507.0之前版本升级的数据库：升级后默认值为mysql。
- 从507.0及之后版本升级的数据库：升级后的参数值保留原设置。

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：根据实际业务需求设置。支持在数据库安装时指定，若指定的值为NULL会报错提醒，若指定的值不在取值范围内，会取默认为ora。可通过以下方式用于安装时设置该参数：

```
gs_install -X /data/software/XmlConfig/Dist_3C3D3REP.xml --init-password --gsinit-parameter="--distribute_default_compatibility ora"
```

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

load_data_behavior_compat_options

参数说明：控制sql_compatibility='MYSQL'以及M-Compatibility模式数据库行为配置项。用于控制LOAD DATA语法导入数据的行为方式。

参数类型：枚举类型

参数单位：无

取值范围：如表6-47所示：

表 6-47 load_data_behavior_compat_options 配置项说明

兼容性配置项	兼容性行为控制
none	不使用任何兼容性配置项。 当使用none与其他兼容性配置项组合设置时，none选项不生效。如“SET load_data_behavior_compat_options='none,new_escape_string';”等价于“SET load_data_behavior_compat_options='new_escape_string';”。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
new_escape_string	控制LOAD DATA导入数据时，支持的转义序列。默认的转义符为'\'。 ● 未设置new_escape_string选项时，支持的转义序列如下： – \a[b[c]]：该转义序列中a、b、c均为八进制数字（0~7），表示一个由八进制数字构成的数值，系统会将其解析为对应的字节值。例如，\141的八进制值为141，对应十进制97，表示ASCII字符（字符a的ASCII编码值为97）。 – \xhh：该转义序列中h表示一个十六进制数字（0~9、a~f、A~F）。hh表示由两位十六进制数字构成的数值，系统会将其解析为对应的字节值。例如，\x6a的十六进制值为6A，对应十进制106，表示ASCII字符'j'（字符'j'的ASCII编码值为106）。 – \r：回车符。 – \n：换行符。 – \b：退格符。 – \v：垂直制表符。 – \t：水平制表符。 – \f：换页符。 – \0：0字符。 ● 设置new_escape_string选项时，支持的转义序列如下： – \r：回车符。 – \n：换行符。 – \b：退格符。 – \t：水平制表符。 – \0：0字符。 – \Z：ASCII 26(Control Z)。 控制LOAD DATA导入数据时，是否对指定的行终结符做转义处理。 ● 未设置new_escape_string选项时，行终结符指定为如下内容时，会将其转义为对应的控制字符： – 0x0A或\n：换行符。 – 0x0D0A或\r\n：回车符和换行符。 – 0x0D或\r：回车符。 – 0x09或\t：水平制表符。 – 0x0B或\v：垂直制表符。 – 0x0C或\f：换页符。 ● 设置new_escape_string选项时，不会对指定的行终结符做转义处理。

版权所有 © 华为云计算技术有限公司

兼容性配置项	兼容性行为控制
fixed_row_column_division	在控制LOAD DATA导入数据时，可通过该选项调整对源数据文件的行、列切分行为。包括但不限于以下场景的处理方式受该选项控制： ● 当源数据文件的行末无行分隔符（lines_terminated_by），且最后一行的最后一列数据以包围符（enclosed_by）开头时： – 未设置该选项时，导入过程中会丢失1个字节。 – 设置该选项后，导入过程中不会丢失数据。 ● 当转义符后紧接列分隔符（fields_terminated_by）时： – 未设置该选项时，列分隔符不会被转义为普通字符。 – 设置该选项后，列分隔符会被转义为普通字符。 ● 当转义符后紧接包围符时： – 未设置该选项时，包围符不会被转义为普通字符。 – 设置该选项后，包围符会被转义为普通字符。 ● 当转义符后紧接非显式指定的行分隔符时： – 未设置该选项时，行分隔符不会被转义为普通字符。 – 设置该选项后，行分隔符会被转义为普通字符。 ● 当不指定行分隔符时： – 未设置该选项时，行分隔符为文件中首次匹配的'\r\n'，'\n'，或'\r'。 – 设置该选项时，行分隔符为'\n'。 ● 当指定转义符和包围符为相同字符时： – 未设置该选项时，会对'\t'、'\n'等转义序列进行转义。 – 设置该选项时，不会对'\t'、'\n'等转义序列进行转义。 ● 在M-Compatibility模式下，指定LOCAL选项进行远程传输时： – 未设置该选项时，指定的行起始符（lines_starting_by）不会生效。当源数据存在整行的'\.'字符串时，'\.'及后续的数据不会被导入。 – 设置该选项时，指定的行起始符能够生效。当源数据存在整行的'\.'字符串时，'\.'及后续的数据会被导入。

默认值：

- 新安装的数据库："new_escape_string, fixed_row_column_division"。
- 从历史版本升级上来的数据库，若历史版本不支持该GUC参数，升级后的默认值为"none"；若历史版本支持该GUC参数，升级后默认值保留原设置。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

版权所有 © 华为云计算技术有限公司

modify_function_property_json

参数说明：该参数以JSON格式控制系统函数的属性。

- oid：目标系统函数的oid。 取值范围为0~10000。
- compatibility：数据库兼容模式。取值范围为A、M、OTHER。
- property_list：函数属性列表，可配置多个属性项。
  - property_name：属性名称，目前支持proshippable、provolatile。
  - property_value：属性值，取值需符合对应属性的取值范围。proshippable的取值范围为t、f，provolatile的取值范围i、s、v。

参数类型：字符串

参数单位：无

取值范围：修改配置文件：0 ~ 50000个字符；gs_guc设置：0 ~ 4099个字符。

默认值：

- 新安装数据库默认值为

  '[{"oid": 4162,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4163,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4186,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4187,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4905,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4906,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4907,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4908,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4909,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4910,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4911,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4912,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4913,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 4914,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

  {"oid": 6286,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},

 版权所有 © 华为云计算技术有限公司

{"oid": 6287,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},
{"oid": 6288,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},
{"oid": 6289,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "s"}]},
{"oid": 6307,"compatibility":"A", "property_list": [{"property_name": "provolatile", "property_value": "v"}]},
{"oid": 4079,"compatibility":"M", "property_list": [{"property_name": "provolatile", "property_value": "s"}]}]'。

- 若升级的基础版本不存在该GUC参数，默认为""。
- 若升级的基础版本存在该GUC参数，保留原设置。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，不建议修改。

设置不当的风险与影响：影响系统函数的属性。

说明

若modify_function_property和modify_function_property_json的配置有冲突，以modify_function_property的配置为准。

illegal_character_strict_check

参数说明：用于控制是否开启非法字符校验。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启参数，支持非法字符校验。当字符串的字节序列无法按编码规则完整解析为合法字符（即实际字节长度不足以构成完整字符）时，判定为非法字符并报错。
- off：表示关闭参数，不支持非法字符校验。

默认值：

- 新安装的数据库：on。
- 从历史版本升级上来的数据库：若历史版本不支持该GUC参数，升级后的默认值为off；若历史版本支持该GUC参数，升级后默认值保留原设置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：非必要用户不要自行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

disable_unsupported_copy_load_options

参数说明：用于控制COPY/LOAD DATA 非推荐语法对于导入导出时产生异常结果的处理行为。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

取值范围：

- on：表示禁用非推荐语法中可能导致异常结果的操作。
- off：表示不禁用非推荐语法中可能导致异常结果的操作。

默认值：

- 新安装的数据库：on。
- 从历史版本升级上来的数据库：若历史版本不支持该GUC参数，升级后的默认值为off；若历史版本支持该GUC参数，升级后默认值保留原设置。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，不建议修改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.16.3 云服务产品版本号

设置云服务相关的版本号信息。

product_version

参数说明：云服务产品版本号信息。

参数类型：字符串

参数单位：无

取值范围：无

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置该参数时，需要遵循以下约束，否则会直接报错。

- 字符串长度不超过50个字符。
- 不能包含“|”、“;”、“&”、“$”、“>”、“<”、“`”、“\”、“!”和换行符。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

hotpatch_version

参数说明：云服务热补丁版本号信息。

参数类型：字符串

参数单位：无

取值范围：无

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：设置该参数时，需要遵循以下约束，否则会直接报错。

- 字符串长度不超过1500个字符。
- 不能包含“|”、“;”、“&”、“$”、“>”、“<”、“`”、“\”、“!”和换行符。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.17 容错性

当数据库系统发生错误时，以下参数控制服务器处理错误的方式。

exit_on_error

参数说明：若打开该开关，ERROR级别报错会升级为PANIC报错，从而可以产生core堆栈。主要用于问题定位和业务测试。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示ERROR级别报错会升级为PANIC报错。
- off：表示不会对ERROR级别报错进行升级。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

restart_after_crash

参数说明：若打开该开关，后端线程崩溃时，GaussDB将自动重新初始化此后端线程。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示GaussDB能够在后端线程崩溃时自动重新初始化此后端线程。在某些情况（比如当采用管理工具（例如xCAT）管理GaussDB时），能够最大限度地提高数据库的可用性。
- off：表示能够使得管理工具在后端线程崩溃时获取控制权并采取适当的措施进行处理。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

omit_encoding_error

参数说明：若打开该开关，数据库的客户端字符集编码为UTF-8时，出现的字符编码转换错误将打印在日志中，有转换错误的被转换字符会被忽略，以"?"代替。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示有转换错误的字符将被忽略，以"?"代替，打印错误信息到日志中。
- off：表示有转换错误的字符不能被转换，打印错误信息到终端。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

该参数通过执行gs_guc reload修改时，如果当前节点上某个session的连接不是来自于客户端，而是来自于该节点所属集群上的其他节点，那么执行gs_guc reload后该参数在该session上不会立即生效，需要与连接节点断开连接后重新连接才会生效。

cn_send_buffer_size

参数说明：指定CN端数据发送数据缓存区的大小。

参数类型：整型

参数单位：kB

取值范围：8 ~ 128

默认值：8

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为100，不带单位，表示100kB；设置为128kB，带单位，表示128kB。取值如果要带单位，必须为kB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_cn_temp_file_size

参数说明：指定SQL语句出错自动重试功能中CN端使用临时文件的最大值，设定为0表示不使用临时文件。

参数类型：整型

参数单位：kB

取值范围：0 ~ 10485760

默认值：5242880（即5GB）

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为100，不带单位，表示100kB；设置为16MB，带单位，表示16MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

data_sync_retry

参数说明：控制fsync到磁盘失败后是否继续运行数据库。在某些操作系统的场景下，fsync失败后，即使重试阶段再次fsync失败也不会报错，从而导致数据丢失。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示fsync同步到磁盘失败后采取重试机制，继续运行数据库。
- off：表示fsync同步到磁盘失败后直接报panic，停止数据库。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

remote_read_mode

参数说明：远程读功能开关。读取主机上的页面失败时，可以从备机上读取对应的页面。

参数类型：枚举类型

参数单位：无

取值范围：off、non_authentication、authentication

- off：表示关闭远程读功能。
- non_authentication：表示开启远程读功能，但不进行证书认证。
- authentication：表示开启远程读功能，但要进行证书认证。

默认值：authentication

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.18 连接池参数

当使用连接池访问数据库时，在系统运行过程中，数据库连接是被当作对象存储在内存中的，当用户需要访问数据库时，并非建立一个新的连接，而是从连接池中取出一

版权所有 © 华为云计算技术有限公司

个已建立的空闲连接来使用。用户使用完毕后，数据库并非将连接关闭，而是将连接放回连接池中，以供下一个请求访问使用。

pooler_port

参数说明：cm_agent、cm_ctl等内部工具运维管理端口，初始化用户或系统管理员通过客户端连接数据库所使用端口。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535。

默认值：5433（实际值为GUC参数**port**值+1，取决于用户配置）

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

pooler_maximum_idle_time

参数说明：Pooler连接自动清理功能使用，当连接池中连接空闲时间超过所设置值时，会触发自动清理机制，清理各节点的空闲连接数到**minimum_pool_size**。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：600（即10min）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为100，不带单位，表示100s；设置为20min，带单位，表示20min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：配置过大，会降低pooler连接池中空闲连接的清理频率，降低回收速率；配置过小，会提高清理频率，导致业务频繁建立连接，对性能有一定损耗。

minimum_pool_size

参数说明：Pooler连接自动清理功能使用，表示自动清理后各pooler连接池对应节点的连接数最小剩余量。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：50

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：配置过大，会减少pooler连接池中空闲连接的清理数量，降低回收速率；配置过小，会增大清理数量，导致业务频繁建立连接，对性能有一定损耗。

max_pool_size

参数说明：连接池与其它某个CN/DN的最大连接数，当集群规模有变化，如增加节点、减少节点时，可能需要调整该参数。

参数类型：整型

参数单位：无

取值范围：0 ~ 65535。此参数在DN上不生效，仅DN上允许设置为0。

默认值：

- 独立部署：

  CN：32768（128核CPU/1024G内存，60核CPU/480G内存）；16384（32核CPU/256G内存）；8192（16核CPU/128G内存）；4096（8核CPU/64G内存）；2048（4核CPU/32G内存）；1000（4核CPU/16G内存）

  DN：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：按照不同实例规格下的默认值进行设置，该参数值需要大于**max_connections**，调整时需预留内部线程所消耗的连接。

设置不当的风险与影响：当业务并发高时，会消耗连接池中CN至其他CN/DN的连接，如果该参数配置过小，连接数达到上限时会产生报错，导致业务失败。由于CN启动时，会根据参数值提前申请内存，所以当该参数值变大，系统会消耗更多内存资源，但总体来说对CN内存影响不大。

persistent_datanode_connections

参数说明：会话是否会释放获得的连接。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示不会释放获得的连接。
- off：表示会释放获得的连接。

默认值：off

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启该参数后，会存在会话持有连接但并未运行查询的情况，导致其他查询申请不到连接报错。

版权所有 © 华为云计算技术有限公司

max_coordinators

参数说明：集群中CN的最大数目，当集群规模有变化，如增加节点、减少节点时，可能需要调整该参数。扩容时，请确保此参数值大于需要扩容到的集群中CN的个数，否则会导致扩容失败。缩容时，若此参数值偏大，CN启动时会消耗更多的内存资源，但总体来说对CN内存影响不大。

参数类型：整型

参数单位：无

取值范围：2 ~ 1024

默认值：128

设置方式：该参数属于POSTMASTER类型参数，不建议修改此参数，若需修改，请参见**表6-1**中对应设置方法进行设置。

设置建议：按照集群的实际规格进行设置。

设置不当的风险与影响：该参数小于集群当前CN个数时，会导致节点创建失败。由于CN启动时，会根据参数值提前申请内存，所以当该参数值变大，系统会消耗更多内存资源，但总体来说对CN内存影响不大。

max_datanodes

参数说明：集群中DN的最大数目，当集群规模有变化，如增加节点、减少节点时，可能需要调整该参数。扩容时，请确保此参数值大于需要扩容到的集群中DN总分片数，否则会导致扩容失败。缩容时，若此参数值偏大，CN启动时会消耗更多的内存资源，但总体来说对CN内存影响不大。

参数类型：整型

参数单位：无

取值范围：2 ~ 65535

默认值：256

设置方式：该参数属于POSTMASTER类型参数，不建议修改此参数，若需修改，请参见**表6-1**中对应设置方法进行设置。

设置建议：按照集群的实际规格进行设置。

设置不当的风险与影响：该参数小于集群当前DN个数时，会导致节点创建失败。由于CN启动时，会根据参数值提前申请内存，所以当该参数值变大，系统会消耗更多内存资源，但总体来说对CN内存影响不大。

cache_connection

参数说明：是否回收连接池的连接。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示回收连接池的连接。

版权所有 © 华为云计算技术有限公司

- off：表示不回收连接池的连接。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭该参数后，会话执行语句结束后会直接清理掉持有的连接，导致业务频繁建立连接，对性能有一定损耗。开启该参数后，在多库场景下，当内存规格较小并且连接数max_connections配置较大时，可能会导致内存不足，DN无法启动。

enable_force_reuse_connections

参数说明：会话是否强制重用新的连接。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示强制使用新连接。
- off：表示使用现有连接。

默认值：off

设置方式：该参数属于BACKEND类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启该参数后，每次执行语句前，会对已持有的连接强制发送reset参数的动作，对性能有一定损耗。

pooler_connect_max_loops

参数说明：该参数由pooler建立连接重试功能使用，主备切换场景增强建立连接的稳定性，若节点间pooler建连失败会尝试跟备机重新建立连接，此时若恰好备机升主机成功，则可以在重试阶段建立连接成功。该参数可以设置重试总轮数，增强建立连接的稳定性。当参数设置为0时，表示关闭重试功能，业务只跟主机建立连接而不跟备机重试。

参数类型：整型

参数单位：无

取值范围：0 ~ 20

默认值：1

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数是高可用参数，设置为0后，会关闭重试功能，降低高可用能力；设置大于1后，会增加重试次数。

版权所有 © 华为云计算技术有限公司

pooler_connect_interval_time

参数说明：该参数由pooler建立连接重试功能使用，当参数pooler_connect_max_loops设置大于1时，该参数可以控制不同重试轮数之间执行时间间隔。参数设置方面，建议略大于当前集群主备切换恢复时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 7200

默认值：15

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。例如，设置为100，不带单位，表示100s；设置为20min，带单位，表示20min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数是高可用参数，若设置过长，会导致重试周期拉长，反而降低高可用能力。

pooler_timeout

参数说明：CN连接池中的连接与其它CN/DN通讯时的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 7200

默认值：600（即10min）

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。例如，设置为100，不带单位，表示100s；设置为20min，带单位，表示20min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，当网络波动时，会导致收发数据报错；设置过大，当网络不通时，会导致收发数据报错的时间变长，降低高可用能力。

pooler_connect_timeout

参数说明：CN连接池与集群中其他CN/DN建立连接时的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 7200

默认值：60（即1min）

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。例如，设置为100，不带单位，表示100s；设置为20min，带单位，表示20min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置过小，当网络波动时，会导致建立连接报错；设置过大，当网络不通时，会导致建立连接报错的时间变长，降低高可用能力。

pooler_cancel_timeout

参数说明：CN连接池在错误处理时cancel某连接的超时时间。如果在子事务或存储过程异常捕获的过程中发生该类超时，那么包含子事务或存储过程的整个事务将发生回滚。在此基础上，在子事务或存储过程异常捕获的过程中，如果错误源自COPY FROM操作中源数据与目标表表结构的不一致，则只要该参数值不为0，就总会触发超时报错。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 7200，其中0（一般不建议）表示关闭此开关，不做超时限制。

默认值：15

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为100，不带单位，表示100s；设置为20min，带单位，表示20min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，当网络波动时，会导致cancel动作报错；设置过大，当网络不通时，会导致cancel动作报错的时间变长，降低高可用能力。

6.3.19 集群事务

介绍集群事务隔离、事务只读、最大prepared事务数、集群维护模式目的参数设置及取值范围等内容。

transaction_isolation

参数说明：设置当前事务的隔离级别。对于开启的新事务，隔离级别固定为**default_transaction_isolation**的值。

参数类型：字符串

参数单位：无

取值范围：只支持以下字符串，对大小写和空格敏感。

- "serializable"：GaussDB目前功能上不支持此隔离级别，设置该隔离级别时，与"repeatable read"效果相同。
- "read committed"：仅读取已提交事务的数据，无法读取未提交的数据。
- "read uncommitted"：与"read committed"表现相同。
- "repeatable read"：仅读取事务开始之前提交的数据，无法读取未提交的数据以及在事务执行期间由其他并发事务提交的修改。
- "default"：设置为**default_transaction_isolation**所设的隔离级别。

默认值："default"。default_transaction_isolation参数使用默认值"read committed"的情况下，通过以下命令查询的结果为"read committed"。

```
show transaction_isolation;
```


版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

transaction_read_only

参数说明：设置当前事务是否为只读事务。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示设置当前事务为只读事务。
- off：表示设置当前事务为读写事务。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

autocommit

参数说明：是否开启自动提交模式。如果设置为on，会话为自动提交模式，每一个单独的语句都会被隐式提交。如果设置为off，则必须使用commit提交事务或使用rollback取消事务。如果autocommit从off修改为on，则对所有未完成的事务执行自动commit。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启自动提交模式。
- off：表示关闭自动提交模式。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置，不支持gs_guc工具设置此参数。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果设置为off，事务中执行DDL或DCL时，不会自动提交。

说明

- 当dbcompatibility='M'时，支持修改该参数为off，否则只能设置为on。
- GaussDB中，pg_settings视图不能查询该参数，可以通过“SHOW autocommit;”查看。
- 使用autocommit参数，要求打开内部自动提交模式，可通过“\echo :AUTOCOMMIT”方式确认内部自动提交是否为打开状态。如果不是自动提交模式，使用“\set AUTOCOMMIT on”进行打开。

版权所有 © 华为云计算技术有限公司

xc_maintenance_mode

参数说明：设置系统是否进入维护模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进入维护模式。
- off：表示不进入维护模式。

须知

谨慎打开这个功能，避免引起集群数据不一致。

默认值：off

设置方式：该参数属于SUSET类型参数，仅支持表6-2中的方式三进行设置。

设置建议：该参数一般供维护人员排查问题使用，不建议用户开启。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

allow_concurrent_tuple_update

参数说明：设置是否允许并发更新。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许并发更新。
- off：表示不允许并发更新。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host

参数说明：主GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

版权所有 © 华为云计算技术有限公司

默认值：主GTM的IP地址。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_port

参数说明：主GTM进程的侦听端口。仅sysadmin用户可见。

说明

该参数由安装时的配置文件指定，请勿轻易修改，否则修改后会影响数据库正常通信。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：安装时指定。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

gtm_host1

参数说明：备GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：备GTM的IP地址。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_port1

参数说明：备GTM进程的侦听端口。仅sysadmin用户可见。

说明

该参数由安装时的配置文件指定，请勿轻易修改，否则修改后会影响数据库正常通信。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：1 ~ 65535

默认值：如有备1，安装时指定。否则为6665。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

gtm_backup_barrier

参数说明：指定是否为GTM启动点创建还原点。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示创建还原点。
- off：表示不创建还原点。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_conn_check_interval

参数说明：设置检查CN本地线程与主GTM连接是否正常的间隔时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_show_any_tuples

参数说明：该参数只有在只读事务中可用，用于分析。当这个参数被设置为on时，表中元组的所有版本都会可见。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：表示表中元组的所有版本都会可见。
- off：表示表中元组的所有版本都不可见。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

在toast场景下，如果reindex前后均有DML操作（主要是insert+update或insert+delete），在reindex操作后，开启只读事务并打开该GUC参数，release版本下单独查询toast表或toast索引表历史数据正常，但是通过主表查询toast列历史数据会报错“missing chunk number xxx”。

gtm_connect_timeout

参数说明：控制GTM连接超时时间，如果GTM的连接时间超过此参数设置的值，则超时返回。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_connect_retries

参数说明：控制GTM连接重试的次数。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：30

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_rw_timeout

参数说明：控制GTM反馈超时时间，如果GTM的反馈时间超过此参数设置的值，也就是等待时间超过了此参数值，则超时返回。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：60（即1min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_redistribute

参数说明：节点不匹配时是否重新分配。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示节点不匹配时重新分配。
- off：表示节点不匹配时不重新分配。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

replication_type

参数说明：标记当前部署模式是一主多备模式或单主机模式。

参数类型：整型

参数单位：无

取值范围：0 ~ 3

- 0：是预留参数。
- 1：表示使用一主多备模式，全场景覆盖，推荐使用。
- 2：表示使用单主机模式，此模式无法扩展备机。
- 3：是预留参数。

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。该参数是CM部署时使用的参数，用户不能对该参数进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

pgxc_node_name

参数说明：指定节点名称。

版权所有 © 华为云计算技术有限公司

在备机请求主机进行日志复制时，如果**application_name**参数没有被设置，那么pgxc_node_name参数会被用来作为备机在主机上的流复制槽名字。该流复制槽的命名方式为“该参数值_备机ip_备机port”。其中，备机ip和备机port取自replconninfo参数中指定的备机ip和端口号。该流复制槽最大长度为61个字符，如果拼接后的字符串超过该长度，则会使用截断后的pgxc_node_name进行拼接，以保证流复制槽名字长度小于等于61个字符。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：当前节点名称。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改后会导致连接集群失败。

enable_twophase_commit

参数说明：当前云数据库解决SDS替换问题，采用的模式为GTM Free，为防止业务滥用导致不可靠问题，提供GUC参数开关enable_twophase_commit允许或禁用分布式两阶段写事务。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示GTM-FREE模式下，允许分布式两阶段写事务。
- off：表示GTM-FREE模式下，禁用分布式两阶段写事务。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

application_type

参数说明：此参数仅在**enable_gtm_free**为on时有效。此参数表示用户的业务类型。

参数类型：枚举类型

参数单位：无

取值范围：

- not_perfect_sharding_type：表示跨节点的业务。设置此值时，允许执行跨节点的语句。
- perfect_sharding_type：表示单节点的业务。设置此值时，如果SQL语句需要多个节点参与，会直接报错，对应的SQL语句会同时打印到系统日志中。设置为此

版权所有 © 华为云计算技术有限公司

值时，使用/*+ multinode */ hint可以显式允许SQL语句在多个节点执行。multinode hint可以加到select、insert、update、delete、merge关键字之后。

默认值：perfect_sharding_type

设置方式：该参数属于USERSET类型参数，此参数不支持使用gs_guc设置，只支持以下两种方式设置：

- 使用gsql等客户端在session级别设置。
- 使用jdbc连接数据库时，在连接字符串中指定ApplicationType参数。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host2

参数说明：如有2号GTM，参数值为2号GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：如有2号GTM，为GTM的IP地址，否则为“”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host3

参数说明：如有3号GTM，参数值为3号GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：如有3号GTM，为GTM的IP地址，否则为“”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host4

参数说明：如有4号GTM，参数值为4号GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：字符串

默认值：如有4号GTM，为GTM的IP地址，否则为“”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host5

参数说明：如有5号GTM，参数值为5号GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：如有5号GTM，为GTM的IP地址，否则为“”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host6

参数说明：如有6号GTM，参数值为6号GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：如有6号GTM，为GTM的IP地址，否则为“”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_host7

参数说明：如有7号GTM，参数值为7号GTM进程所在的IP地址。仅sysadmin用户可见。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：如有7号GTM，为GTM的IP地址，否则为“”。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_port2

参数说明：如有2号GTM，参数值为2号GTM进程的侦听端口。仅sysadmin用户可见。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：如有2号GTM，安装时指定。否则为6666。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

gtm_port3

参数说明：如有3号GTM，参数值为3号GTM进程的侦听端口。仅sysadmin用户可见。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：如有3号GTM，安装时指定。否则为6666。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

gtm_port4

参数说明：如有4号GTM，参数值为4号GTM进程的侦听端口。仅sysadmin用户可见。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：如有4号GTM，安装时指定。否则为6666。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

版权所有 © 华为云计算技术有限公司

gtm_port5

参数说明：如有5号GTM，参数值为5号GTM进程的侦听端口。仅sysadmin用户可见。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：如有5号GTM，安装时指定。否则为6666。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

gtm_port6

参数说明：如有6号GTM，参数值为6号GTM进程的侦听端口。仅sysadmin用户可见。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：如有6号GTM，安装时指定。否则为6666。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

gtm_port7

参数说明：如有7号GTM，参数值为7号GTM进程的侦听端口。仅sysadmin用户可见。

参数类型：整型

参数单位：无

取值范围：1 ~ 65535

默认值：如有7号GTM，安装时指定。否则为6666。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议进行修改。

设置不当的风险与影响：修改端口号会影响数据库正常通信。

enable_defer_calculate_snapshot

参数说明：延迟计算快照的xmin和oldestxmin，执行1000个事务或者间隔1s才触发计算。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示延迟计算快照xmin和oldestxmin。
- off：表示实时计算快照xmin和oldestxmin。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为on时可以在高负载场景下减少计算快照的开销，但是会导致oldestxmin推进较慢，影响垃圾元组回收。设置为off时xmin和oldestxmin可以实时推进，但是会增加计算快照时的开销。

seqscan_csn_cache_num

参数说明：此参数控制csn cache的大小，此cache只有在seqscan通过MVCC快照扫描heap表页面判断可见性会使用到。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000。参数值为0时，表示不使用csn缓存机制。

默认值：100

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

reject_gtm_address

参数说明：设置拒绝连接的GTM IP地址。

参数类型：字符串

参数单位：无

取值范围：以逗号分隔，分隔后的每个子串格式为IP[:port]。

说明

上述的[:port]表示可选。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：仅由CM组件在特定流程中自动设置，不建议手动进行修改。

设置不当的风险与影响：不当设置可能导致集群与GTM之间无法进行通信，进而影响集群事务。

版权所有 © 华为云计算技术有限公司

verify_clog_csnlog_mode

参数说明：控制Clog与Csnlog页面数据正确性校验报错日志级别和告警输出。

参数类型：枚举类型

参数单位：无

取值范围：

- off：表示关闭报错和告警输出。
- warning：表示开启warning级别报错和告警输出。
- error：表示开启error级别报错和告警输出。
- panic：表示开启panic级别报错和告警输出。

默认值：

- 在使用支持该功能的版本进行全新安装时，默认配置为warning。
- 从不支持该参数的版本升级至支持该参数的版本时，默认配置为off，升级完成后需手动配置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请根据实际需求进行配置。若设置为error或panic，在页面故障的情况下，会直接影响业务甚至数据库运行。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.20 双集群复制参数

enable_slot_log

参数说明：是否开启复制槽主备同步特性，仅涉及归档槽和备份槽。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启复制槽主备同步特性。
- off：表示不开启复制槽主备同步特性。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：主机上创建归档/备份槽建议开启，备机创建归档/备份槽建议关闭。

设置不当的风险与影响：设置不当时，主备间复制同步除外的备份、归档、build等功能会受到影响。

RepOriginId

参数说明：该参数是一个会话级别的GUC参数，在双向逻辑复制的场景下，为避免数据循环复制，需要设置为一个非0的值。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-2**中的方式三进行设置。

设置建议：在双向逻辑复制的场景下，需要设置为一个非0的值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

auto_csn_barrier

参数说明：流式容灾的主集群和GTM LITE集群是否开启barrier打点功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：容灾流程中会自动设置开启，不建议用户手动关闭。GTM LITE集群开启备机读功能，需要设置该参数为on。

设置不当的风险与影响：关闭后，影响灾备集群和GTM LITE集群备机读数据一致性。

stream_cluster_run_mode

参数说明：流式容灾双集群容灾场景，表示CN/DN节点属于主集群还是备集群。

参数类型：枚举类型

参数单位：无

取值范围：

- cluster_primary：表示节点是主集群的节点。
- cluster_standby：表示节点是备集群的节点。

默认值：cluster_primary

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数为容灾集群的状态参数，内部流程使用，用户可读，不建议用户修改。

设置不当的风险与影响：手动修改会造成灾备集群异常。

hadr_process_type

参数说明：基于流式复制异地容灾解决方案或者同城双中心高可用方案的流程标识。

版权所有 © 华为云计算技术有限公司

参数类型：枚举类型

参数单位：无

取值范围：

- none：表示当前无流程。
- failover：表示当前处于灾备集群升主流程。
- switchover_promote：表示主备集群倒换流程中灾备集群升主流程。
- switchover_demote：表示主备集群倒换流程中主集群降为灾备集群流程。
- dorado_failover：表示dorado灾备集群升主流程。
- dorado_switchover_demote：表示dorado主备集群倒换流程中主集群降为灾备集群流程。
- dorado_failover_abnormal：表示dorado主集群共享盘故障时，灾备集群升主流程。
- dorado_failover_in_standby_stream：表示同城双中心高可用支持基于流式复制异地容灾解决方案中异地集群为容灾主集群时，dorado备集群升为dorado主集群（同时为异地容灾-灾备集群）流程。
- dorado_failover_abnormal_in_standby_stream：表示同城双中心高可用支持基于流式复制异地容灾解决方案中异地集群为异地容灾主集群且dorado主集群（同时为异地容灾-灾备集群）共享盘故障时，dorado备集群升为dorado主集群（同时为异地容灾-灾备集群）流程。

默认值：none

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：容灾集群的状态参数，内部流程使用，用户可读，不建议用户修改。

设置不当的风险与影响：手动修改会造成灾备集群异常。

cluster_name

参数说明：标识当前集群的集群名称。

须知

集群名称cluster_name不可设置为“invalid_cluster_uuid”。

参数类型：字符串

参数单位：无

取值范围：合法的集群名称。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数在容灾流程使用，容灾集群依赖该参数进行集群识别，用户可读，不建议用户修改。

设置不当的风险与影响：手动修改可能造成容灾功能异常。

版权所有 © 华为云计算技术有限公司

hadr_primary_cluster_name

参数说明：标识容灾中当前物理日志发送端集群的集群名称。

须知

当流式容灾日志接收端集群的首备的hadr_primary_cluster_name配置为“invalid_cluster_uuid”时，将不再启动与日志发送端集群的日志同步，日志接收端集群将持续处于断连状态。

参数类型：字符串

参数单位：无

取值范围：合法的集群名称。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：日志接收端集群的识别参数，容灾流程使用，用户可读，不建议用户修改。

设置不当的风险与影响：手动修改可能造成容灾功能异常。

hadr_standby_cluster_num

参数说明：当前流式容灾的灾备集群数量。

须知

- hadr_standby_cluster_num参数在基于流式复制的异地容灾解决方案（一主集群+一备集群）下无效。
- 当hadr_replconninfo_config_mode参数设置为cluster_manager，即日志发送端集群与日志接收端集群间的容灾链路由集群管理配置时，该参数无效。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：容灾集群的状态参数，容灾流程使用，用户可读，不建议用户修改。

设置不当的风险与影响：手动修改可能造成容灾功能异常。

switchover_abort_timeout

参数说明：switchover主备消息交互完成后，若主机退出线程超过该参数设定的超时阈值，将触发进程异常终止。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647，其中0表示不使用超时abort功能。

默认值：10000

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若参数值设置过低，可能导致switchover过程中主机更容易异常退出。

6.3.21 开发人员选项

allow_system_table_mods

参数说明：设置是否允许修改系统表的结构和系统自带模式的名称。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许修改系统表的结构和系统自带模式的名称。
- off：表示不允许修改系统表的结构和系统自带模式的名称。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议用户修改该参数设置。

设置不当的风险与影响：该参数打开后，可能导致系统表损坏，甚至数据库无法启动等严重问题。

allow_create_sysobject

参数说明：设置是否允许在系统模式下创建或修改函数、存储过程、同义词等对象。此处的系统模式指数据库初始后自带的模式，但不包含public模式。系统模式的oid通常小于16384。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许初始用户和系统管理员在系统模式下创建或修改函数、存储过程、同义词等对象。其他用户是否允许创建这些对象请参考对应模式的权限要求。
- off：表示禁止所有用户在系统模式下创建或修改函数、存储过程、同义词等对象。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置该参数为off可以提高系统的安全性，如果不需要在系统模式下创建或修改对象，推荐设置该参数为off。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

新安装数据库为off；若升级的基础版本不存在该GUC参数，默认为off；若升级的基础版本存在该GUC参数，保留原设置。

debug_assertions

参数说明：设置是否打开各种断言检查。要使用这个参数，必须在编译GaussDB的时候定义宏USE_ASSERT_CHECKING（通过configure选项 --enable-cassert完成）。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开断言检查。
- off：表示不打开断言检查。

默认值：off

说明

当启用断言选项编译GaussDB时，debug_assertions默认值为on 。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该功能能够协助调试，当遇到异常或者崩溃的问题时，请打开此参数，用以定位程序的错误。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ignore_checksum_failure

参数说明：设置是否忽略数据校验错误。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示忽略数据校验错误。
- off：表示数据校验错误正常报错。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议用户修改该参数设置。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：该参数打开后会导致系统忽略失败（但仍然会告警），继续执行可能导致崩溃，传播或保存损坏数据，无法从远程节点恢复数据及其他严重问题。

ignore_system_indexes

参数说明：设置读取系统表时是否忽略系统索引（但是修改系统表时依然同时修改索引）。

说明

这个参数在从系统索引被破坏的表中恢复数据的时候非常有用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示忽略系统索引。
- off：表示不忽略系统索引。

默认值：off

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ignore_invalid_pages

参数说明：控制回放结束时检测到无效页面是否触发PANIC级别错误。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示回放结束时忽略无效页面，检测到无效页面时不会触发PANIC级别错误，仅记录WARNING级别日志。
- off：表示回放结束时不忽略无效页面，检测到无效页面时会触发PANIC级别错误。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当数据库出现页面校验的PANIC级别错误时，如果需要快速恢复业务，则可以将该参数设置为on尝试恢复。但此行为可能导致后续进程异常终止、数据丢失、数据错误同步到其他备机等问题。

post_auth_delay

参数说明：在认证成功后，延迟指定的时间，再启动服务器连接。允许调试器附加到启动进程上。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147

默认值：0，表示无延迟。

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示post_auth_delay为60s；带单位取值1min，表示post_auth_delay为1min。取值如果要带单位，必须为s、min、h、d。

设置建议：该参数只用于调试和问题定位，为避免影响正常业务运行，生产环境下请确保参数值为默认值0。

设置不当的风险与影响：参数设置为非0时可能会因认证延迟时间过长导致集群状态异常。

pre_auth_delay

参数说明：启动服务器连接后，延迟指定的时间，再进行认证。允许调试器附加到认证过程上。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 60

默认值：0，表示无延迟。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示pre_auth_delay为60s；带单位取值1min，表示pre_auth_delay为1min。取值如果要带单位，必须为s、min、h、d。

设置建议：该参数只用于调试和问题定位，为避免影响正常业务运行，生产环境下请确保参数值为默认值0。

设置不当的风险与影响：参数设置为非0时可能会因认证延迟时间过长导致集群状态异常。

trace_notify

参数说明：设置是否为LISTEN和NOTIFY命令生成大量调试输出。**client_min_messages**或**log_min_messages**级别必须是DEBUG1或者更低，才能把这些输出分别发送到客户端或者服务器日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开输出功能。
- off：表示关闭输出功能。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

trace_recovery_messages

参数说明：启用与恢复相关的调试输出的日志记录，否则不记录日志。该参数允许覆盖正常设置的**log_min_messages**，但是仅限于特定的消息，建议只在调试备机时使用。

参数类型：枚举类型

参数单位：无

取值范围：有效值有debug5、debug4、debug3、debug2、debug1、log。参数的详细信息请参考**表6-23**。

默认值：log

说明

- 默认值log表示不影响记录决策。
- 除默认值外，其他值会导致优先级更高的恢复相关调试信息被记录，因为它们有log优先权。对于常见的**log_min_messages**设置，这会导致无条件地将它们记录到服务器日志上。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议只在调试备机时修改该参数取值，其余情况推荐使用默认值。

设置不当的风险与影响：设置除了log以外的值时，会打印较多日志，占用一定io资源。

trace_sort

参数说明：设置是否在日志中打印排序操作中的资源使用相关信息。这个选项只有在编译GaussDB的时候定义了TRACE_SORT宏才可以使用，不过目前TRACE_SORT只能缺省，无法进行调整。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打印排序操作中的资源使用相关信息。
- off：表示不打印排序操作中的资源使用相关信息。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

zero_damaged_pages

参数说明：控制检测到导致GaussDB报告错误的损坏的页头后，是否终止当前事务。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：系统发出一个警告而非抛出一个错误，把损坏的页面清零然后继续处理。这种行为会破坏数据，也就是所有在已经损坏页面上的行记录。但是它允许绕开坏页面然后从表中尚存的未损坏页面上继续检索数据行。因此它在因为硬件或者软件错误导致的崩溃中进行恢复是很有用的。
- off：系统不会将损坏页面填充为零。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：通常不应该设置为on，除非不需要从崩溃的页面中恢复数据。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

string_hash_compatible

参数说明：该参数用来控制char类型和varchar/text类型的hash值计算方式是否相同，以此来判断进行分布列从char类型到相同值的varchar/text类型转换，数据分布变化时，是否需要进行重分布。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示计算方式相同，不需要进行重分布。
- off：表示计算方式不同，需要进行重分布。

说明

计算方式的不同主要体现在字符串计算hash值时传入的字节长度上。（如果为char，则会忽略字符串后面空格的长度，如果为text或varchar，则会保留字符串后面空格的长度。）hash值的计算会影响到查询的计算结果，因此此参数一旦设置后，在整个数据库使用过程中不能再对其进行修改，以避免查询错误。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

remotetype

参数说明：设置远程连接类型。

参数类型：枚举类型

参数单位：无

取值范围：有效值有application，coordinator，datanode，gtm，gtmproxy，internaltool，gtmtool。

版权所有 © 华为云计算技术有限公司

默认值：application

设置方式：该参数属于BACKEND类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_user_defined_exception

参数说明：异常最大个数，默认值不可更改。

参数类型：整型

参数单位：无

取值范围：当前只能取固定值1000。

默认值：1000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

enable_fast_numeric

参数说明：标识是否开启Numeric类型数据运算优化。Numeric数据运算是较为耗时的操作之一，通过将Numeric转化为int64/int128类型，提高Numeric运算的性能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启Numeric优化。
- off：表示关闭Numeric优化。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会导致涉及到Numeric类型的查询语句出现性能劣化。

enable_numeric_optimization

参数说明：设置是否开启Numeric类型数据运算优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在向量化引擎中，启用NUMERIC数据类型优化。

版权所有 © 华为云计算技术有限公司

- off：表示在向量化引擎中，不启用NUMERIC数据类型优化。

默认值：off

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无。

说明

当前版本暂不支持此优化，当前仅能查看enable_numeric_optimization默认的值。

enable_fast_vecop

参数说明：标识是否开启向量化引擎运算优化。对向量化引擎中Timestamp，int32，int64，float8类型的运算优化，并减少行存调用开销，支持自动向量化运算。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启向量化运算优化。
- off：表示关闭向量化运算优化。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置关闭影响向量化引擎中Timestamp，int32，int64，float8类型的运算性能。

enable_compress_spill

参数说明：设置是否开启下盘压缩功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启下盘压缩功能。
- off：表示关闭下盘压缩功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

resource_track_log

参数说明：控制自诊断的日志级别。目前仅对多列统计信息进行控制。

参数类型：枚举类型

参数单位：无

取值范围：

- summary：显示简略的诊断信息。
- detail：显示详细的诊断信息。

目前这两个参数值只在显示多列统计信息未收集的告警的情况下有差别，summary不显示未收集多列统计信息的告警，detail会显示这类告警。

默认值：summary

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_parallel_ddl

参数说明：控制多CN对同一数据库对象是否能安全的并发执行DDL操作。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示可以安全的并发执行DDL操作，不会出现分布式死锁。
- off：表示不能安全的并发执行DDL操作，可能会出现分布式死锁。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

support_batch_bind

参数说明：控制是否允许通过JDBC、ODBC、Libpq等接口批量绑定和执行PBE形式的语句。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许批量绑定和执行。
- off：表示不允许批量绑定和执行。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

numa_distribute_mode

参数说明：用于控制部分共享数据和线程在NUMA节点间分布的属性。用于大型多NUMA节点的ARM服务器性能调优，一般不用设置。

参数类型：字符串

参数单位：无

取值范围：当前有效取值为“none”，“all”。

- "none"：表示不启用本特性。
- "all"：表示将部分共享数据和线程分布到不同的NUMA节点下，减少远端访存次数，提高性能。目前仅适用于拥有多个NUMA节点的ARM服务器，并且要求全部NUMA节点都可用于数据库进程，不支持仅选择一部分NUMA节点。

说明

- 当前版本x86平台下不支持numa_distribute_mode设置为“all”。
- 启用此参数的情况下，如果发生CPU扩缩容操作，需要同步执行“gs_cgroup -u -T Gaussdb -f 新的CPU核数范围”命令更新cgroup绑核信息，保持cgroup控制组可用核数范围和扩缩容后可用的CPU核数范围一致，否则将导致集群启动失败。

默认值："none"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_pagewriter

参数说明：设置增量检查点打开后，是否显示线程的刷页信息以及增量检查点的详细信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示显示详细信息。
- off：表示不显示详细信息。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：一般不建议设置为on。

设置不当的风险与影响：设置为on后，会导致显示信息增多，影响性能。

版权所有 © 华为云计算技术有限公司

advance_xlog_file_num

参数说明：用于控制在后台周期性地提前初始化Xlog文件的数目。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000000（0表示不提前初始化）。例如，取值10，表示后台线程会周期性地根据当前Xlog写入位置提前初始化10个Xlog文件。

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数可用来避免事务提交时执行Xlog文件初始化影响性能，但该情况仅在超重负载时才可能出现，因此一般不用配置。

设置不当的风险与影响：设置过大会过多占用磁盘空间，导致磁盘占用率异常，因此调整该参数时需考虑实际磁盘大小，调整参数时占用磁盘大小可按此公式计算：advance_xlog_file_num * wal_segment_size(16MB)。建议预扩占用空间不超过可用空间10%，设置过大可能造成集群不可用且无法恢复。

comm_sender_buffer_size

参数说明：用于设置Stream计划中CN与DN之间，DN与DN之间每次交互的BUFFER大小，在一些情况下不同的取值会对Stream性能产生影响，设置后需要重启集群生效。

参数类型：整型

参数单位：kB

取值范围：1 ~ 1024

默认值：8

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，容易频繁收发，时延小，但是吞吐率低。设置过大，吞吐率会提升，但是时延增加。

ustore_attr

参数说明：该参数主要用来控制USTORE存储引擎表的信息统计、回滚类型、UBTree索引类型、重点模块（包括数据、索引、回滚段、回放等）运行时数据的校验，主要用于协助研发问题定位。

参数类型：字符串

参数单位：无

取值范围：该参数值的设置方式采用key-value模式，key和value取值对应关系和说明如下。如果是多个key-value组合设置，中间使用“;”作为分隔符，例如：ustore_attr='ustore_verify_level=FAST;ustore_verify_module=UPAGE:UBTREE:UNDO:REDO'。

版权所有 © 华为云计算技术有限公司

- ustore_verify_level：控制校验的级别。

  取值范围：取值不区分大小写，见下述表格详细描述。

  表 6-48 ustore_verify_level 取值含义说明

参数取值	含义
NONE	NONE表示关闭校验，性能测试场景下推荐开启。
FAST	FAST表示快速校验，校验内容少，性能影响最小。
COMPLETE	COMPLETE表示全量校验，校验内容最多，性能影响比较大。
NORMAL	该级别与FAST级别作用一致，仅为保证前向兼容而保留，建议使用FAST级别。
SLOW	该级别与COMPLETE级别作用一致，仅为保证前向兼容而保留，建议使用COMPLETE级别。

  默认值：FAST

- ustore_verify_module：控制校验的模块。

  取值范围：设置值UPAGE、UBTREE、UNDO、REDO、ROACH中的一个或者多个，或者单独设置ALL或者NULL（不区分大小写）。当设置UPAGE、UBTREE、UNDO、REDO、ROACH中的多个值时，使用":"作为连接符。例如ustore_verify_module=UPAGE:UBTREE:UNDO:REDO。

  表 6-49 ustore_verify_module 取值含义说明

参数取值	含义
UPAGE	表示开启数据页面校验。
UBTREE	表示开启UBTREE索引校验。
UNDO	表示开启回滚段数据校验。
REDO	表示开启REDO流程的数据页面校验。
ROACH	此参数在当前版本无效，底层逻辑已去掉校验，ustore_verify_module=roach依然能设置成功但实际不生效。
ALL	表示同时开启UPAGE、UBTREE、UNDO、REDO、ROACH模块数据的校验。
NULL	表示同时关闭UPAGE、UBTREE、UNDO、REDO、ROACH模块数据的校验。

  默认值：UPAGE:UBTREE:UNDO

- index_trace_level：控制开启索引追踪并控制打印级别，开启后在索引扫描的过程中，会根据不同的打印级别对符合条件的索引元组的信息进行打印。

 版权所有 © 华为云计算技术有限公司

取值范围：取值见表格描述。

默认值：NO

表 6-50 index_trace_level 取值含义说明

参数取值	含义
NO	不打印任何额外信息。
NORMAL	打印可见索引元组相关信息，包括： ● 当前索引元组所在索引页面号以及偏移。 ● 当前元组状态。 ● 当前元组对应的TID以及partOid。 ● 当前元组对应的xmin和xmax信息。 ● 当前元组内容（如果开启enable_log_tuple）。
VISIBILITY	在NORMAL的基础上，额外打印没有通过可见性检查的索引元组的信息，并标明是否可见。
SHOWHIKEY	在VISIBILITY的基础上，尝试打印页面上HIKEY元组的信息。
ALL	打印扫描的索引页面上所有元组的相关信息。

- enable_log_tuple：打印日志级提示信息时，是否允许同时将相关元组的内容打印出来，以便进行问题排查和定位。

  取值范围：on或者off（不区分大小写）

  默认值：off
- enable_ustore_sync_rollback：控制USTORE表是否开启同步回滚。

  取值范围：布尔值

  默认值：true
- enable_ustore_async_rollback：控制USTORE表是否开启异步回滚。

  取值范围：布尔值

  默认值：true
- enable_ustore_page_rollback：控制USTORE表是否开启页面回滚。

  取值范围：布尔值

  默认值：true
- enable_ustore_partial_seqscan：是否允许USTORE表开启部分扫描。

  取值范围：布尔值

  默认值：false
- enable_candidate_buf_usage_count：是否开启缓存区使用计数统计。

  取值范围：布尔值

  默认值：false
- ustats_tracker_naptime：控制USTORE表统计信息周期。

  取值范围：1 ~ 1073741

  默认值：20，单位为秒。

版权所有 © 华为云计算技术有限公司

- umax_search_length_for_prune：控制USTORE表prune操作搜索的最大深度。
  取值范围：1 ~ 1073741
  默认值：10，单位为次。
- ustore_td_reservation_timeout：控制DML语句在USTORE表页面等待TD时的最大超时时间。
  取值范围：0 ~ 120000
  默认值：60000，单位为毫秒。

说明

- ustore_attr设置参数值时，key和value之间的“=”前后不要有空格或者其他字符，例如“ustore_attr='ustore_verify_level = FAST;'”，内核代码校验时会发现参数不合法，导致参数设置失败。
- 默认开启在线校验功能，若要关闭在线校验功能，请将ustore_attr设置为""。

默认值："ustore_verify_level=FAST;ustore_verify_module=UPAGE:UBTREE:UNDO"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请慎重设置ustore_attr参数，建议在工程师协助下修改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

index_txntype

参数说明：控制UBTree的索引类型为PCR或RCR，PCR支持通过索引进行闪回查询，RCR不支持。空字符串与“RCR”在大部分场景等效，reindex场景下，新索引保持原索引类型，不受该参数影响。

参数类型：字符串

参数单位：无

取值范围：""、"PCR"、"RCR"

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方式进行设置。

设置建议：如果需要使用PCR索引可以改成“PCR”，否则不建议修改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_index_kind

参数说明：控制分区表创建索引的默认行为。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：表示对分布式部署方式不开启全局分区索引功能。
- 1：表示默认创建局部索引。
- 2：表示默认创建全局索引。

默认值：2

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方式进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为0时，无法创建全局分区索引。

enable_auto_segment_remain_cleanup

参数说明：该参数用来是否启动段页式残留自动清理方式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用段页式残留自动清理方式。
- off：表示禁用段页式残留自动清理方式。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请避免随意切换清理方式，否则可能会出现无法正常清理段页式残留或其他异常。

enable_stream_noblock_memcopy

参数说明：设置是否开启向量化引擎中stream算子内存复制优化，如果开启该参数，向量化引擎中local stream算子中数据发送将采用非等待的方式进行，提升local stream算子的执行效率。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启stream算子内存复制优化。
- off：表示关闭stream算子内存复制优化。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为off后，可能导致local stream算子的执行效率下降。

enable_remove_redundant_update_col

参数说明：控制是否开启伪更新分布列优化功能。该参数功能详细内容请参见**伪更新分布列**。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示开启伪更新分布列优化功能。
- off：表示关闭伪更新分布列优化。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需求进行设置。

设置不当的风险与影响：该参数取值设置为on时，开启伪更新分布列优化功能，会导致真更新场景存在少量性能劣化。不使用时请设置取值为off。

convert_illegal_char_mode

参数说明：用于在**enable_convert_illegal_char**参数开启时，设置客户端显示非法编码的占位符。

参数类型：字符串

参数单位：无

取值范围：ascii编码表中十进制编码为32至126的95个可显示字符。

默认值："?"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

default_segment

参数说明：该参数用来控制是否默认创建段页式表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示未指定segment字段时，默认创建段页式表。
- off：表示未指定segment字段时，默认创建页式表。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

pldebugger_timeout

参数说明：设置pldebugger server端等待debug端响应的超时时间。

参数类型：整型

参数单位：s（秒）

版权所有 © 华为云计算技术有限公司

取值范围：1 ~ 86400

默认值：900（即15min）

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值3600，表示pldebugger_timeout为3600s；带单位取值1h，表示pldebugger_timeout为1h。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.22 审计

6.3.22.1 审计开关

audit_enabled

参数说明：控制传统审计的开启和关闭。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启动审计功能。
- off：表示关闭审计功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启审计功能会对数据库性能有一定影响，影响程度取决于记录审计的频率。若设置为on，请根据实际业务和追溯需要设置其他审计开关项；若设置为off，则无法通过审计日志追溯历史操作。为确保系统稳定性，不应在高频执行审计操作期间关闭审计功能。

audit_directory

参数说明：审计文件的存储目录。可以是相对于数据目录data的相对路径，也可以是绝对路径，且仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法目标路径。

默认值：“pg_audit”。如果使用om工具部署集群，则审计日志路径为“$GAUSSLOG/pg_audit/实例名称”。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：若设置的值为非法路径，或路径空间已满，会导致审计功能无法正常使用；不同的CN或DN实例需要设置不同的审计文件存储目录，否则会导致审计日志异常。

说明

- 合法路径：用户对此路径有读写权限。
- 非法路径：用户对此路径无读写权限。

audit_data_format

参数说明：审计日志文件的格式。当前仅支持二进制或文本格式，且仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：

- binary：二进制格式。
- text：文本格式。

默认值："binary"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当前仅支持二进制或文本格式，非法配置按默认值处理。

audit_rotation_interval

参数说明：指定创建一个新审计日志文件的时间间隔。当当前时间与上次创建审计日志文件的时间差超过此参数值时，服务器将生成一个新的审计日志文件。考虑到性能影响，间隔时间会引入随机延迟。

参数类型：整型

参数单位：min（分钟）

取值范围：1 ~ 35791394（INT_MAX/60）

默认值：1d（即1440min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如：不带单位取值1440，表示audit_rotation_interval参数的值为1440min；带单位取值2d，表示audit_rotation_interval参数的值为2880min。取值如果要加上单位，必须为“min”、“h”、“d”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致审计目录下文件过多；设置过大会导致单个审计文件过大。请勿随意调整此参数，否则可能会导致**audit_resource_policy**无法生效。如需控制审计日志的存储空间和时间，可通过**audit_resource_policy**、**audit_space_limit**和**audit_file_remain_time**等参数实现。

版权所有 © 华为云计算技术有限公司

audit_rotation_size

参数说明：指定审计日志文件的最大容量（实际文件大小可能会超过设置上限1~10MB）。当审计日志消息的总量超过此参数值时，服务器将生成一个新的审计日志文件。

参数类型：整型

参数单位：kB

取值范围：1024 ~ 1048576。

默认值：10MB（即10240kB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如：不带单位取值131072，表示audit_rotation_size参数的值为131072kB；带单位取值128MB，表示audit_rotation_size参数的值为128MB。取值如果要加上单位，必须为“kB”、“MB”、“GB”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致审计目录下文件过多；设置过大会导致单个审计文件过大。请勿随意调整此参数，否则可能会导致**audit_resource_policy**无法生效。如需控制审计日志的存储空间和时间，可通过**audit_resource_policy**、**audit_space_limit**和**audit_file_remain_time**参数实现。审计日志文件中记录的单条日志占用空间大小超过此参数值时会被作为无效日志文件。

audit_resource_policy

参数说明：控制审计日志的保存策略，即优先采用空间限制还是时间限制。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示采用空间优先策略，最多保存**audit_space_limit**所设定大小的审计日志。
- off：表示采用时间优先策略，至少保存**audit_file_remain_time**所设定时长的审计日志。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：选择空间优先策略时，请勿随意调整时间策略相关的GUC参数：**audit_rotation_interval**、**audit_file_remain_time**；选择时间优先策略时，请勿随意调整空间策略相关的GUC参数：**audit_rotation_size**、**audit_space_limit**、**audit_file_remain_threshold**。

设置不当的风险与影响：此参数需配合对应策略相关的其他GUC参数共同使用，设置不当会导致审计日志的保存策略不符合预期。

audit_file_remain_time

参数说明：表示审计日志需保存的最短时间要求，该参数在**audit_resource_policy**为off时生效。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：d（天）

取值范围：0 ~ 730，0表示无时间限制。

默认值：90

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：请根据实际需要设置审计日志的保留时间。**audit_resource_policy**为on时，请勿随意调整此参数。

设置不当的风险与影响：该参数在**audit_resource_policy**为off时生效。设置过小会导致保留的审计日志量过少、审计追溯有效时长降低；设置过大会导致审计目录下保存过多审计日志，影响审计查询/删除的性能并占用磁盘空间。

audit_space_limit

参数说明：审计文件占用的磁盘空间总量。

参数类型：整型

参数单位：kB

取值范围：1024 ~ 1073741824（1024GB）

默认值：1GB（即1048576kB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值131072，表示audit_space_limit参数的值为131072kB；带单位取值128MB，表示audit_space_limit参数的值为128MB。取值如果要加上单位，必须为“kB”、“MB”、“GB”。

设置建议：请根据实例实际可用空间及需要保存的审计日志量来设置。

设置不当的风险与影响：该参数在**audit_resource_policy**为on时生效。设置过小会导致保留的审计日志量过少，审计追溯有效时长降低；设置过大会导致审计目录下保存过多审计日志，影响审计查询/删除的性能并占用磁盘空间。

说明

- 此参数的生效范围是pg_audit目录下的单个进程实例文件夹。即默认情况下，每一个CN、DN目录下的审计文件占用的磁盘空间总量为1GB。
- 多审计线程场景下，审计文件占用的磁盘空间的最小值为**audit_thread_num**与**audit_rotation_size**的乘积，请保确audit_space_limit的值大于audit_thread_num与audit_rotation_size的乘积。

audit_file_remain_threshold

参数说明：审计目录下审计文件的最大个数。

参数类型：整型

参数单位：个

取值范围：100 ~ 1048576

默认值：1048576

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中的对应设置方法进行设置。

设置建议：推荐使用默认值，不建议随意调整此参数。如需控制审计日志的存储空间和保留时间，请通过**audit_resource_policy**、**audit_space_limit**和**audit_file_remain_time**参数进行控制。

设置不当的风险与影响：设置不当可能会导致**audit_resource_policy**无法生效。

audit_thread_num

参数说明：该参数表示用于审计的线程个数。

参数类型：整型

参数单位：个

取值范围：1 ~ 48

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中的对应设置方法进行设置。

设置建议：当**audit_dml_state**开关打开且系统处于高性能场景时，建议增大此参数，以保证审计消息能够被及时处理和记录；其他场景下建议保持默认值。

设置不当的风险与影响：线程数量设置得越多，对系统性能的影响越大，占用的系统CPU和I/O资源越多。

6.3.22.2 用户和权限审计

audit_login_logout

参数说明：该参数决定是否审计用户的登录（包括登录成功和登录失败）、注销。

参数类型：整型

参数单位：无

取值范围：0 ~ 7

- 0：表示关闭用户登录、注销审计功能。
- 1：表示只审计用户登录成功。
- 2：表示只审计用户登录失败。
- 3：表示只审计用户登录成功和失败。
- 4：表示只审计用户注销。
- 5：表示只审计用户注销和登录成功。
- 6：表示只审计用户注销和登录失败。
- 7：表示审计用户登录成功、失败和注销。

默认值：7

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置值越小，审计的用户登录、注销行为种类越少，不可追溯的风险越大；设置值越大，可记录的用户登录、注销行为种类越多，当有大量登录、注销行为时，会影响数据库性能。

audit_database_process

参数说明：该参数决定是否对数据库的启动、停止、切换和恢复进行审计。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭数据库启动、停止、切换和恢复的审计功能。
- 1：表示开启数据库启动、停止、切换和恢复的审计功能。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为0时，存在无法追溯数据库启动、停止、切换和恢复行为的风险。

audit_user_locked

参数说明：该参数决定是否审计用户的锁定和解锁。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭用户锁定和解锁审计功能。
- 1：表示开启审计用户锁定和解锁功能。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为0时，存在无法追溯用户锁定和解锁行为的风险。

audit_user_violation

参数说明：该参数决定是否审计用户的越权访问操作。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭用户越权操作审计功能。
- 1：表示开启用户越权操作审计功能。

版权所有 © 华为云计算技术有限公司

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需要对用户越权操作进行审计的情况下建议设置为1。

设置不当的风险与影响：设置为0时，存在无法追溯用户越权审计行为的风险；设置为1时，当进行大量越权操作，会对数据库性能有一定影响。

audit_grant_revoke

参数说明：该参数决定是否审计用户权限授予和回收的操作。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭用户权限授予和回收操作的审计功能。
- 1：表示开启用户权限授予和回收操作的审计功能。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为0时，存在无法追溯用户权限授予和回收行为的风险。设置为1时，当进行大量权限授予和回收操作时，会对数据库性能有一定影响。

audit_security_label

参数说明：该参数决定是否审计用户安全标签的创建、删除和应用操作。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭用户安全标签的创建、删除和应用操作的审计功能。
- 1：表示开启用户安全标签的创建、删除和应用操作的审计功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置为0时，存在无法追溯用户安全标签的创建、删除和应用行为的风险；设置为1时，当进行大量用户安全标签的创建、删除和应用行为时，会对数据库性能有一定影响。

audit_internal_event

参数说明：该参数决定是否对初始用户审计内部工具cm_agent、gs_clean、WDRXdb的连接及进行的操作、DN上是否对来自CN的连接进行审计。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示对初始用户使用内部工具cm_agent、gs_clean、WDRXdb的登录、退出登录及操作进行审计，DN上对来自CN的登录、退出登录进行审计。
- off：表示不对初始用户使用内部工具cm_agent、gs_clean、WDRXdb的登录、退出登录及操作进行审计，DN上不对来自CN的登录、退出登录进行审计。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，可以节约审计日志占用空间，提升审计日志查询性能。

设置不当的风险与影响：开启此参数会记录数据库服务端内部工具或节点之间登录和退出登录操作，导致性能下降。

full_audit_users

参数说明：该参数用于配置全量审计用户列表，对列表中的用户执行的所有可被审计的操作记录审计日志。

参数类型：字符串

参数单位：无

取值范围：字符串，多个用户名需使用逗号分隔。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需要对特定用户的所有行为记录审计时设置。

设置不当的风险与影响：列表中用户过多时，这些用户的所有操作都会记录审计日志，导致数据库性能下降。

no_audit_client

参数说明：该参数用于配置不需要审计的客户端名称及IP地址列表，该IP地址支持IPv6地址格式。参数格式为：客户端名称@IP，同gs_query_audit函数中的client_conninfo字段，例如“cm_agent@127.0.0.1, gs_clean@127.0.0.1”。

参数类型：字符串

参数单位：无

取值范围：字符串，多个配置项需使用逗号分隔。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：数据库服务端内部工具或节点之间通信也会产生审计日志，针对这些风险较低的审计场景的可以通过配置no_audit_client参数不记录审计，以节约审计日志占用空间，提升审计日志查询性能。

设置不当的风险与影响：设置不当可能会导致某客户端连接进行操作无法被审计，导致危险操作无法追溯。

版权所有 © 华为云计算技术有限公司

说明

当执行的SQL语句同时满足full_audit_users和no_audit_client参数配置时，以no_audit_client配置优先，不记录审计日志。

6.3.22.3 操作审计

audit_system_object

参数说明：该参数表示是否对数据库对象的CREATE、DROP、ALTER等操作进行审计。数据库对象包括DATABASE、USER、SCHEMA、TABLE等。通过修改该配置参数的值，可以只审计需要的数据库对象的操作。

参数类型：整型

参数单位：无

取值范围：0 ~ 1073741823

- 0：表示关闭数据库对象的CREATE、DROP、ALTER等操作审计功能。
- 非0：表示只审计某类或者某些数据库对象的CREATE、DROP、ALTER等操作。

取值说明：

该参数的值由30个二进制位的组合求出，这30个二进制位分别代表30类数据库对象。如果对应的二进制位取值为0，表示不审计对应的数据库对象的CREATE、DROP、ALTER等操作；取值为1，表示审计对应的数据库对象的CREATE、DROP、ALTER等操作。这30个二进制位代表的具体审计内容请参见**表6-51**。

用于记录SQL PATCH的参数存在特殊性，如果对该对象进行审计且audit_dml_state_select也开启时，对于一条SQL PATCH操作的审计日志会作为DML和DDL被记录两次，如果调用remote接口，则会在入参对应的节点上产生DDL日志，而不是在下发语句的节点上。

默认值：67121159（十进制），对应二进制为：00 0100 0000 0000 0011 0000 0000 0111，表示对DATABASE、SCHEMA、USER、NODE GROUP、SQLPatch这五种数据库对象的DDL操作进行审计。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要设置记录审计的数据库对象类型。在主备强制选主场景建议audit_system_object取最大值，审计表1中的所有对象。

设置不当的风险与影响：需要审计的对象类型越多，对系统性能的影响越大，占用的系统CPU和I/O越多。

表 6-51 audit_system_object 取值含义说明

二进制位	含义	取值说明
第0位	是否审计DATABASE对象的CREATE、DROP、ALTER操作。	• 0表示不审计该对象的CREATE、DROP、ALTER操作。 • 1表示审计该对象的CREATE、DROP、ALTER操作。

版权所有 © 华为云计算技术有限公司

二进制位	含义	取值说明
第1位	是否审计SCHEMA对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第2位	是否审计USER和USER MAPPING对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第3位	是否审计TABLE对象的CREATE、DROP、ALTER、TRUNCATE、VACUUM、ANALYZE操作。	● 0表示不审计该对象的CREATE、DROP、ALTER、TRUNCATE、VACUUM、ANALYZE操作。 ● 1表示审计该对象的CREATE、DROP、ALTER、TRUNCATE、VACUUM、ANALYZE操作。
第4位	是否审计INDEX对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第5位	● 是否审计VIEW对象的CREATE、DROP、ALTER操作。 ● 是否审计MATVIEW对象的CREATE、DROP操作。	● 0表示不审计VIEW对象的CREATE、DROP、ALTER操作和MATVIEW对象的CREATE、DROP操作。 ● 1表示审计VIEW对象的CREATE、DROP、ALTER操作和MATVIEW对象的CREATE、DROP操作。
第6位	是否审计TRIGGER对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第7位	是否审计PROCEDURE/FUNCTION对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第8位	是否审计TABLESPACE对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第9位	是否审计RESOURCE POOL对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。

版权所有 © 华为云计算技术有限公司

二进制位	含义	取值说明
第10位	是否审计WORKLOAD对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第11位	是否审计SERVER对象的CREATE、DROP、ALTER操作	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第12位	保留	-
第13位	是否审计NODE GROUP对象的CREATE、DROP操作。	● 0表示不审计该对象的CREATE、DROP操作。 ● 1表示审计该对象的CREATE、DROP操作。
第14位	是否审计ROW LEVEL SECURITY对象的CREATE、DROP、ALTER操作。	● 0表示不审计该对象的CREATE、DROP、ALTER操作。 ● 1表示审计该对象的CREATE、DROP、ALTER操作。
第15位	是否审计TYPE对象的CREATE、DROP、ALTER操作。	● 0表示不审计TYPE对象的CREATE、DROP、ALTER操作。 ● 1表示审计TYPE对象的CREATE、DROP、ALTER操作。
第16位	是否审计TEXT SEARCH对象（CONFIGURATION和DICTIONARY）的CREATE、DROP、ALTER操作。	● 0表示不审计TEXT SEARCH对象的CREATE、DROP、ALTER操作。 ● 1表示审计TEXT SEARCH对象的CREATE、DROP、ALTER操作。
第17位	是否审计DIRECTORY对象的CREATE、DROP、ALTER操作。	● 0表示不审计DIRECTORY对象的CREATE、DROP、ALTER操作。 ● 1表示审计DIRECTORY对象的CREATE、DROP、ALTER操作。
第18位	是否审计SYNONYM对象的CREATE、DROP、ALTER操作。	● 0表示不审计SYNONYM对象的CREATE、DROP、ALTER操作。 ● 1表示审计SYNONYM对象的CREATE、DROP、ALTER操作。
第19位	是否审计SEQUENCE对象的CREATE、DROP、ALTER操作。	● 0表示不审计SEQUENCE对象的CREATE、DROP、ALTER操作。 ● 1表示审计SEQUENCE对象的CREATE、DROP、ALTER操作。

版权所有 © 华为云计算技术有限公司

二进制位	含义	取值说明
第20位	是否审计CMK、CEK对象的CREATE、ALTER、DROP操作。	● 0表示不审计CMK、CEK对象的CREATE、ALTER、DROP操作。 ● 1表示审计CMK、CEK对象的CREATE、ALTER、DROP操作。
第21位	是否审计PACKAGE对象的CREATE、DROP、ALTER操作。	● 0表示不审计PACKAGE对象的CREATE、DROP、ALTER操作。 ● 1表示审计PACKAGE对象的CREATE、DROP、ALTER操作。
第22位	保留	-
第23位	保留	-
第24位	是否审计对gs_global_config全局对象的ALTER、DROP操作。	● 0表示不审计对系统表gs_global_config全局对象的ALTER、DROP操作。 ● 1表示审计对系统表gs_global_config全局对象的ALTER、DROP操作。
第25位	是否审计FOREIGN DATA WRAPPER对象的CREATE、DROP、ALTER操作，目前该功能暂不支持。	● 0表示不审计FOREIGN DATA WRAPPER对象的CREATE、DROP、ALTER操作。 ● 1表示审计FOREIGN DATA WRAPPER对象的CREATE、DROP、ALTER操作。
第26位	是否审计SQL PATCH对象的CREATE、ENABLE、DISABLE、DROP操作。	● 0表示不审计SQL PATCH对象的CREATE、ENABLE、DISABLE、DROP操作。 ● 1表示审计SQL PATCH对象的CREATE、ENABLE、DISABLE、DROP操作。
第27位	保留	-
第28位	是否审计DBLINK对象的CREATE、ALTER、DROP操作。	● 0表示不审计DBLINK对象的CREATE、ALTER、DROP操作。 ● 1表示审计DBLINK对象的CREATE、ALTER、DROP操作。
第29位	是否审计RESOURCE LABEL、MASKING POLICY、AUDIT POLICY对象的CREATE、DROP、ALTER操作。	● 0表示不审计RESOURCE LABEL、MASKING POLICY、AUDIT POLICY对象的CREATE、ALTER、DROP操作。 ● 1表示审计RESOURCE LABEL、MASKING POLICY、AUDIT POLICY对象的CREATE、ALTER、DROP操作。

版权所有 © 华为云计算技术有限公司

audit_dml_state

参数说明：该参数表示是否对所有表的INSERT、UPDATE、DELETE、MERGE操作进行审计。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭所有表的INSERT、UPDATE、DELETE、MERGE操作审计功能。
- 1：表示开启所有表的INSERT、UPDATE、DELETE、MERGE操作审计功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为1，当数据库频繁进行INSERT、UPDATE、DELETE、MERGE操作时会频繁记录审计日志，导致数据库性能下降。

audit_dml_state_select

参数说明：该参数表示是否对SELECT操作进行审计。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭SELECT操作审计功能。
- 1：表示开启SELECT操作审计功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为1，当数据库频繁进行SELECT操作时会频繁记录审计日志，导致数据库性能下降。

audit_function_exec

参数说明：该参数表示在执行存储过程、匿名块或自定义函数（不包括系统自带函数）时是否记录审计信息。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭对存储过程、匿名块或自定义函数（不包括系统自带函数）执行的审计功能。

版权所有 © 华为云计算技术有限公司

- 1：表示开启对存储过程、匿名块或自定义函数（不包括系统自带函数）执行的审计功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为1，当数据库频繁执行存储过程、匿名块或自定义函数时会频繁记录审计日志，导致数据库性能下降。

audit_system_function_exec

参数说明：这个参数决定在执行白名单内的系统函数时是否记录审计日志。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭系统函数执行的审计功能。
- 1：表示开启系统函数执行的审计功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为1，当数据库频繁执行白名单内系统函数时会频繁记录审计日志，导致数据库性能下降。

支持记录审计的系统函数白名单如下表所示：

set_working_grand_version_num_manually	set_config	pg_terminate_backend	pg_cancel_backend	pg_cancel_session	pg_cancel_invalid_query
pg_reload_conf	pg_rotate_logfile	pg_terminate_session	pg_terminate_backend	pg_start_backup	pg_stop_backup
pg_create_restore_point	pg_switch_xlog	pg_cbm_get_merged_file	pg_cbm_recycle_file	pg_enable_delay_ddl_recycle	pg_disable_delay_ddl_recycle
pg_cbm_rotate_file	gs_roach_enable_delay_ddl_recycle	gs_roach_disable_delay_ddl_recycle	gs_roach_stop_backup	pg_last_xlog_receive_location	pg_xlog_replay_pause

版权所有 © 华为云计算技术有限公司

pg_xlog_replay_resume	gs_roach_switch_xlog	gs_pitr_archive_slot_force_advance	gs_pitr_clean_history_global_barriers	gs_download_obs_file	gs_upload_obs_file
gs_set_obs_file_context	gs_set_obs_delete_location	gs_hadr_do_switchover	gs_set_obs_delete_location_with_slotname	gs_streaming_dr_in_switchover	pg_advisory_lock
pg_advisory_lock_shared	pg_advisory_unlock	pg_advisory_unlock_shared	pg_advisory_unlock_all	pg_advisory_xact_lock	pg_advisory_xact_lock_shared
pg_try_advisory_lock	pg_try_advisory_lock_shared	pg_try_advisory_xact_lock	pg_try_advisory_xact_lock_shared	gs_get_hadr_key_cn	pg_create_physical_replication_slot_extern
pg_create_logical_replication_slot	pg_drop_replication_slot	pg_logical_slot_peek_changes	pg_logical_slot_get_changes	pg_logical_slot_get_binary_changes	pg_replication_origin_drop
pg_replication_origin_session_reset	local_space_shrink	gs_space_shrink	global_space_shrink	pg_free_remain_segment	gs_fault_inject
sqladvisor.init	sqladvisor.set_weight_params	sqladvisor.set_cost_params	sqladvisor.assign_table_type	gs_repair_file	local_clear_bad_block_info
gs_repair_page	clean_abnormal_sql	abnormal_sql_manage_allowlist	dsl_add_rule	dsl_del_rule	dsl_set_rule
dsl_reload_rule	pgxc_fenced_udf_process	pgxc_terminate_all_fenced_udf_process	-	-	-

说明

在系统函数执行这类审计事件的审计记录中，object_name字段的内容为系统函数名，不再带函数参数。

audit_copy_exec

参数说明：这个参数决定是否对COPY操作进行审计。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0、1

- 0：表示关闭COPY审计功能。
- 1：表示开启COPY审计功能。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为1，当数据库频繁执行COPY操作时会频繁记录审计日志，导致数据库性能下降。

audit_set_parameter

参数说明：该参数表示是否对SET操作进行审计。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭SET审计功能。
- 1：表示开启SET审计功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为1，当数据库频繁执行SET操作时会频繁记录审计日志，导致数据库性能下降。

audit_xid_info

参数说明：该参数决定是否在审计日志字段detail_info中记录SQL语句的事务ID。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭审计日志记录事务ID功能。
- 1：表示开启审计日志记录事务ID功能。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

如果开启此开关，审计日志中detail_info信息则以xid开始，例如：

```
detail_info: xid=14619 , create table t1(id int);
```

对于不存在事务ID的审计行为，则记录xid=NA。

audit_rollback

参数说明：该参数表示是否对rollback操作进行审计。该参数可在PDB级别设置。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭rollback操作审计功能。
- 1：表示开启rollback操作审计功能。

默认值：0。在PDB场景内，若未设置该参数，则继承来自全局的设置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启后rollback操作会记录审计。

audit_sql_error

参数说明：该参数表示是否对执行错误的SQL进行审计。该参数可在PDB级别设置。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示关闭执行错误的SQL操作审计功能。
- 1：表示开启执行错误的SQL操作审计功能。

默认值：0。在PDB场景内，若未设置该参数，则继承来自全局的设置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：开启后执行错误的SQL操作会记录审计。

6.3.23 事务监控

通过设置事务超时预警，可以监控自动回滚的事务并定位其中的语句问题，并且也可以监控执行时间过长的语句。

transaction_sync_naptime

参数说明：为保证数据一致性，当本地事务与GTM上snapshot中状态不一样时会阻塞其他事务的运行，需要等待本地节点上事务状态与GTM状态一致后再运行。当CN上等

版权所有 © 华为云计算技术有限公司

待时长超过transaction_sync_naptime时会主动触发gs_clean进行清理，缩短不一致时的阻塞时长。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483

说明

若该值设为0，则不会在阻塞达到时长时主动调用gs_clean进行清理，而是依据gs_clean_timeout间隔来调用gs_clean，默认是5分钟。

默认值：30

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大时，会使CN触发清理的时间延长，不一致时的阻塞时间更久。

transaction_sync_timeout

参数说明：为保证数据一致性，当本地事务与GTM上snapshot中状态不一样时会阻塞其他事务的运行，需要等待本地节点上事务状态与GTM状态一致后再运行。当CN上等待时长超过transaction_sync_timeout时会报错，回滚事务，避免由于sync lock等其他长时间线程停止响应的情况造成对系统的阻塞。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483

说明

- 若该值设为0，则不会在阻塞超时的情况时报错，回滚事务。
- 该值不为0时，必须大于gs_clean_timeout，避免DN由于还未被gs_clean清理的残留事务阻塞超时，引起不必要的事务回滚。

默认值：600（10min）

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.24 GTM 相关参数

GTM相关参数可以在gtm.conf文件中进行设置，或通过gs_guc进行设置。

nodename

参数说明：主GTM或备GTM的名称。

参数类型：字符串

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：字符串，要符合标识符的命名规范。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

port

参数说明：主GTM或备GTM侦听的主机端口号。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647，建议设置为1024 ~ 65535。

默认值：6666

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_file

参数说明：日志文件名。

参数类型：字符串

参数单位：无

取值范围：字符串，要符合标识符的命名规范。

默认值："gtm-%Y-%m-%d_%H%M%S.log"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

active_host

参数说明：目标GTM的地址。即在主GTM上时为备GTM的地址，在备GTM上时为主GTM的地址。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串，要符合标识符的命名规范。

默认值：""

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

local_host

参数说明：HA本地地址，根据集群配置文件进行设置，不需要手动设置。该地址支持IPv4与IPv6格式。

参数类型：字符串

参数单位：无

取值范围：字符串，要符合标识符的命名规范。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

active_port

参数说明：目标GTM的服务器端口号。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647，建议设置为1024 ~ 65535。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数由安装时的配置文件指定，请勿轻易修改，否则修改后会影响数据库正常通信。

local_port

参数说明：HA本地端口。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647，建议设置为1024 ~ 65535。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数由安装时的配置文件指定，请勿轻易修改，否则修改后会影响数据库正常通信。

版权所有 © 华为云计算技术有限公司

standby_connection_timeout

参数说明：设置GTM主备之间的超时时间。此参数控制GTM主备机之间的超时设置，增大该参数可以增加GTM主备之间的网络容错能力，但是也会增加故障场景下GTM主备断连的检测时长。

参数类型：整型

参数单位：s（秒）

取值范围：5 ~ 2147483647

默认值：5

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大时，可能无法及时检测到断连故障。

keepalives_count

参数说明：在支持TCP_KEEPCNT套接字选项的操作系统上，此参数控制GTM服务端在断开与客户端连接之前，等待未响应的保持活跃信号的最大数量。该参数仅在备GTM上生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0，表示GTM未收到客户端反馈的保持活跃信号则立即断开连接。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

keepalives_idle

参数说明：在支持TCP_KEEPIDLE套接字选项的操作系统上，此参数控制GTM服务端连接上客户端之后，如果没有数据发送，多长时间发送保持活跃信号。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

keepalives_interval

参数说明：在支持TCP_KEEPINTVL套接字选项的操作系统上，此参数控制GTM服务端保持活跃信号发送的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

synchronous_backup

参数说明：指定是否以同步方式备份到GTM备机。

参数类型：枚举类型

参数单位：无

取值范围：

- on：表示开启同步方式。
- off：表示关闭同步方式。
- auto：表示自动同步方式。

默认值：auto

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

query_memory_limit

参数说明：设置查询可以使用的内存百分比限制。该参数仅适用于默认资源组。对于其它的资源组，没有查询内存限制影响。

参数类型：浮点型

参数单位：无

取值范围：0.0 ~ 1.0

默认值：0.25

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

wlm_max_mem

参数说明：设置GTM执行时可使用的最大内存。

参数类型：整型

参数单位：MB

取值范围：512 ~ 2147483647

默认值：2048

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

config_file

参数说明：GTM配置文件名，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法的文件名称。

默认值："gtm.conf"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

data_dir

参数说明：GTM数据文件目录。

参数类型：字符串

参数单位：无

取值范围：合法的目录路径。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

listen_addresses

参数说明：声明服务器侦听客户端的TCP/IP地址。该地址支持IPv4与IPv6格式。

参数类型：字符串

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- 主机名或IP地址，多个值之间用英文逗号分隔。支持IPv4与IPv6格式。
- 星号（*）表示所有IP地址。
- 置空则服务器不会侦听任何IP地址，这种情况下，只有Unix域套接字可以用于连接数据库。

默认值："*"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_directory

参数说明：当**logging_collector**设置为on时，log_directory决定存放服务器日志文件的目录。 它可以是绝对路径，或者是相对路径（相对于数据目录的路径）。

须知

- 当配置文件中log_directory的值为非法路径（即用户对此路径无读写权限）时，会导致集群无法重新启动。
- 修改log_directory时，当指定路径为合法路径（即用户对此路径有读写权限）时，日志输出到新的路径下。当指定路径为非法路径时，日志输出到上一次的合法日志输出路径下而不影响数据库正常运行。此时即使指定的log_directory的值非法，也会写入到配置文件中。

参数类型：字符串

参数单位：无

取值范围：合法的目录路径。

默认值：“gtm_log”，表示在数据目录下的“gtm_log/”目录下生成服务器日志。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_min_messages

参数说明：控制写到服务器日志文件夹下GTM日志文件中的消息级别，当产生该设置参数级别相等或更高级别的消息时，将这些信息记入日志文件。因此级别越低，服务器运行日志中记录的消息就越多。

参数类型：枚举类型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：消息级别从低到高为debug5、debug4、debug3、debug2（debug）、debug1、info、notice、warning、error、log、fatal、panic。参数的详细信息请参见**表6-23**。

说明

- debug的消息级别等效于debug2。
- 对于log_min_messages，log的消息级别高于error，低于fatal。

默认值：warning

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

alarm_component

参数说明：在对告警做上报时，会进行告警抑制，即同一个实例的同一个告警项在**alarm_report_interval**（默认值为10s）内不做重复上报。在这种情况下，该参数设置用于处理告警内容的告警组件的位置。

参数类型：字符串

参数单位：无

取值范围：

- 若前置脚本gs_preinstall中的--alarm-type参数设置为5，表示未对接第三方组件，告警写入system_alarm日志，此时GUC参数alarm_component的取值为：/opt/huawei/snas/bin/snas_cm_cmd。
- 若前置脚本gs_preinstall中的--alarm-type参数设置为1，表示对接第三方组件，此时GUC参数alarm_component的值为第三方组件的可执行程序的绝对路径。

默认值："/opt/huawei/snas/bin/snas_cm_cmd"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

alarm_report_interval

参数说明：指定告警上报的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

standby_only

参数说明：是否强制同步信息到备机，在一主多备模式下只强制同步到ETCD。

参数类型：整型

参数单位：无

取值范围：

- 0：表示不强制同步信息到备机。
- 1：表示强制同步信息到备机。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_max_trans

参数说明：设置gtm最大可接收连接数，不建议用户修改该参数。

参数类型：整型

参数单位：无

取值范围：256 ~ 200000

默认值：8192

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。如果需要改动，此参数不能小于（最大连接数+100）。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_connect_control

参数说明：设置gtm开启连接控制，检测连接IP是否来自集群内部。

参数类型：布尔型

参数单位：无

取值范围：

- on：检测连接IP是否来自集群内部，非集群内部的IP连接则拒绝访问。
- off：不会检测连接IP是否来自集群内部。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_authentication_type

参数说明：GTM模块端口认证方式。

参数类型：枚举类型

参数单位：无

取值范围：

- trust：表示未配置端口认证。
- gss：表示采用kerberos端口认证。

> **须知**
>
> 只有当kerberos服务端和客户端成功安装后才能修改为gss，否则GTM模块无法正常通信，将影响集群状态。

默认值：trust

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_krb_server_keyfile

参数说明：kerberos服务端key文件所在位置，需要配置为绝对路径。该文件通常为${GAUSSHOME}/kerberos路径下，以keytab格式结尾，文件名与集群运行所在用户名相同。与上述**gtm_authentication_type**参数是配对的，当gtm_authentication_type参数修改为gss时，该参数也必须配置为正确路径，否则将影响集群状态。

参数类型：字符串

参数单位：无

取值范围：合法的文件路径。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_gtm_free

参数说明：大并发场景下同一时刻存在活跃事务较多，GTM下发的快照变大且快照请求变多的情况下，瓶颈卡在GTM与CN通讯的网络上。为消除该瓶颈，引入GTM-FREE模式。取消CN和GTM的交互，取消CN下发GTM获取的事务信息给DN。CN只向各个DN发送query，各个DN由本地产生快照及xid等信息，开启该参数支持分布式事务读最终一致性，即分布式事务只有写外部一致性，不具有读外部一致性。

版权所有 © 华为云计算技术有限公司

须知

业务使用GTM-Free模式时，建议将**application_type**设置成perfect_sharding_type，以便及时发现可能导致数据不一致的SQL语句。否则，系统不会拦截可能导致数据不一致的语句，造成数据不一致。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启GTM-FREE模式，集群状态为读最终一致性。
- off：表示非GTM-FREE模式。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：GTM-FREE模式有可能出现数据不一致问题。

gtm_option

参数说明：GTM模式选项，用来指定选用的GTM模式，需要在GTM/CN/DN所有实例上配置，配置的值要一致，共三种模式：GTM模式，GTM-Lite模式，GTM-Free模式。其中GTM模式和GTM-Lite模式要在**enable_gtm_free**参数设置为off的情况下生效，当前版本暂不支持安装好的集群进行不同GTM模式之间的切换。这三种不同的GTM模式介绍，参见《开发指南》中“特性使用指导 > GTM > GTM模式”章节。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：表示GTM模式；
- 1：表示GTM-Lite模式；
- 2：表示GTM-Free模式。

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

csn_sync_interval

参数说明：用来指定GTM主备之间同步CSN的时间间隔。

参数类型：整型

参数单位：s（秒）

版权所有 © 华为云计算技术有限公司

取值范围：1 ~ 2147483647

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

restore_duration

参数说明：该参数用来指定GTM上XID或CSN的恢复间隔（个数）。

参数类型：整型

参数单位：无

取值范围：1000000 ~ 2147483647

默认值：1000000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_enable_threadpool

参数说明：用来指定GTM是否开启GTM线程池功能，设置后需要重启才能生效。

参数类型：布尔型

参数单位：无

取值范围：

- true：表示开启GTM线程池功能。
- false：表示不开启GTM线程池功能。

默认值：true

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_num_threads

参数说明：当**gtm_enable_threadpool**线程池功能开启时，用来控制线程池工作线程的数量。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0 ~ 16384之间。该数值与**gtm_max_trans**大小相关，不应该超过（gtm_max_trans - 1 - 辅助线程数），其中辅助线程数当前版本为2。

默认值：1024

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

check_gtm_primary_time

参数说明：该参数设置GTM检测其主节点与ETCD主节点一致性的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647。0表示检测功能关闭，不进行检测。

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

seq_cache_coefficient

参数说明：该参数用来指定GTM上sequence缓存的系数，该缓存系数用于计算sequence的恢复值。

参数类型：整型

参数单位：无

取值范围：3 ~ 20000

默认值：20000

设置方式：该参数属于POSTMASTER类型参数，请参考**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，最小值不能低于节点数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_dcf_run_mode

参数说明：该参数设置GTM在使用DCC时，DCC的DCF运行模式。

参数类型：整型

参数单位：无

取值范围：0 ~ 2。0表示开启选举，1表示手动选举，2表示不开启选举。

默认值：2

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由CM设置，不建议手动设置或更改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。可能导致不必要的主备切换或主备状态异常。

gtm_dcc_id

参数说明：该参数设置GTM在使用DCC时，本实例的DCC编号。

参数类型：整型

参数单位：无

取值范围：1 ~ 127

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由OM设置与维护，不允许用户修改。

设置不当的风险与影响：无。

gtm_dcf_extreme_mode

参数说明：该参数设置GTM在使用DCC时，DCC的DCF是否开启最大可用模式。

参数类型：整型

参数单位：无

取值范围：0 ~ 2。0表示最大保护模式，1表示最大可用模式，2表示最大性能模式。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由CM设置，不建议手动设置或更改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。可能影响业务的一致性。

gtm_dcf_static_leader_timeout

参数说明：该参数设置GTM在使用DCC时，优先等待DCF静态主或者原主升主的超时时间，主要用于集群启动阶段。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：此参数由CM或OM设置，不建议手动设置或更改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。可能造成集群启动异常。

gtm_dcc_config

参数说明：该参数设置GTM在使用DCC时，DCC的集群配置信息。

参数类型：字符串

参数单位：无

取值范围：包含合法信息的json格式字符串。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由OM设置与维护，不允许用户修改。

设置不当的风险与影响：无。

gtm_dcc_data_dir

参数说明：该参数设置GTM在使用DCC时，DCC的数据目录。

参数类型：字符串

参数单位：无

取值范围：有效的绝对路径。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由OM设置与维护，不允许用户修改。

设置不当的风险与影响：无。

gtm_dcf_sync_nodes

参数说明：该参数设置GTM在使用DCC时，DCC之间的同步方式。

参数类型：字符串

参数单位：无

取值范围：

- ANY num_sync:standby1,standby2：指定复制备机的策略。ANY N:1,2,...表示任选“：”后面的N个备节点同步日志。
- FIRST num_sync:standby1,standby2：指定复制备机的策略。FIRST N:1,2,...表示按“：”后面顺序进行日志优先级同步。

其中num_sync指的需要把日志同步给备节点的数量，standby是备节点id。

默认值：""

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由CM或OM设置，不建议手动设置或更改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。可能导致不必要的主备切换或主备状态异常。

gtm_dcf_candidate_nodes

参数说明：该参数设置GTM在使用DCC时，DCC的主节点候选者列表。

参数类型：字符串

参数单位：无

取值范围：以逗号分隔的数字。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由CM或OM设置，不建议手动设置或更改。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。可能导致不必要的主备切换或主备状态异常。

gtm_dcc_log_level

参数说明：该参数设置GTM在使用DCC时，DCC的日志打印级别。

参数类型：字符串

参数单位：无

取值范围：

- “NONE”：表示关闭日志打印，不能与以下日志级别混合使用。
- “RUN_ERR|RUN_WAR|RUN_INF|DEBUG_ERR|DEBUG_WAR|DEBUG_INF|TRACE|PROFILE|OPER”：表示开启对应级别日志。

默认值：“RUN_ERR|RUN_WAR|DEBUG_ERR|OPER|RUN_INF|PROFILE”

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gtm_ddb

参数说明：该参数设置GTM当前使用ETCD还是DCC。

参数类型：字符串

参数单位：无

取值范围：etcd或dcc。

默认值：etcd

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：此参数由OM在集群安装时设置，不允许安装后手动修改。

设置不当的风险与影响：无。

6.3.25 升级参数

IsInplaceUpgrade

参数说明：是否在升级的过程中。该参数属于升级参数，用户无法修改，仅sysadmin用户可以访问。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在升级过程中。
- off：表示不在升级过程中。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

inplace_upgrade_next_system_object_oids

参数说明：就地升级过程中，新增系统对象的OID。该参数属于升级参数，用户无法修改。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

om_operation_mode

参数说明：集群处于运维或者正常运行状态。该参数用于内部组件交互，不允许用户使用。

参数类型：枚举类型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- upgrade_system_object_mode：表示升级正处于系统对象更新阶段。
- running_mode：表示升级正常运行阶段。

默认值：running_mode

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数配置为upgrade_system_object_mode，会导致不能重命名数据库名称。

upgrade_mode

参数说明：升级模式。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

- 0：表示不在升级过程中。
- 1：表示在就地升级过程中。
- 2：表示在灰度升级过程中。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数属于升级参数，不建议用户自己修改。推荐使用默认值。

设置不当的风险与影响：严禁升级过程中自行修改该参数，将会导致升级失败且无法回滚的严重问题。非升级场景下，自行修改该参数会导致数据库使用报错或其他不可预知后果。

说明

特殊情况：在使用灰度升级的情况下，若选择策略为大版本升级，即需要执行升级脚本和替换二进制包，会将upgrade_mode设置为2，选择策略为小版本升级，只替换二进制包，则不会设置upgrade_mode设置为2。

6.3.26 其它选项

enable_default_ustore_table

参数说明：指定是否默认使用Ustore存储引擎。仅当**enable_ustore**参数开启时，该参数设置有效。该参数为on时，创建的表默认都为Ustore表；该参数为off时，创建的表默认都为Astore表。升级场景下该参数的默认值会优先保证兼容性，即升级后的默认值保持原默认值。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示默认使用Ustore存储引擎。

版权所有 © 华为云计算技术有限公司

- off：表示默认不使用Ustore存储引擎。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：需要特别注意的是，使用Ustore表时，必须要开启**track_counts**和**track_activities**参数，否则会引起空间膨胀。

enable_ustore

参数说明：指定是否开启Ustore存储引擎，该参数为on时，支持创建Ustore表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启Ustore存储引擎。
- off：表示不开启Ustore存储引擎。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：用于USTORE的支持版本控制，不建议修改。

设置不当的风险与影响：需要特别注意的是，使用Ustore表时，必须要开启**track_counts**和**track_activities**参数，否则会引起空间膨胀。

enable_segment_datafile_preallocate

参数说明：指定是否开启段页式文件扩展时立即分配磁盘空间模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启立即分配磁盘空间模式，文件扩展时优先使用fallocate立即分配磁盘空间，如果系统不支持fallocate，则使用逐字节写零方式申请磁盘空间。
- off：表示不开启立即分配磁盘空间模式，文件扩展时不立即分配磁盘空间，文件以空洞方式增长。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

consistency_check_module

参数说明：用于设置需要进行一致性校验的模块。

版权所有 © 华为云计算技术有限公司

参数类型：字符串

参数单位：无

取值范围：模块在开启状态（on）下会进行一致性校验，在关闭状态（off）下不会进行校验，可由SHOW consistency_check_module查看：

```
ALL,on(),off(INDEX)
```

当前支持校验的模块只有INDEX模块，INDEX模块的校验是指在Astore表B-Tree的索引扫描IndexScan的过程中，会校验索引元组与堆表元组的一致性。

默认值：'off(ALL)'。即默认关闭所有模块的校验。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

1. 首先，可以通过SHOW consistency_check_module来查看哪些模块是支持校验的。当前支持校验模块的输出结果如下所示：
   ```
   gaussdb=# SHOW consistency_check_module;
    consistency_check_module
   --------------------------
    ALL,on(),off(INDEX)
   (1 row)
   ```
2. 支持可校验的模块通过大写标识，特殊标识ALL用于对所有模块进行设置。可以使用on/off来控制是否进行校验。比如设置校验索引与堆表的一致性校验可使用如下命令：
   ```
   gaussdb=# SET consistency_check_module='on(INDEX)';
   SET
   gaussdb=# SHOW consistency_check_module;
    consistency_check_module
   --------------------------
    ALL,on(INDEX),off()
   (1 row)
   ```
   可以看到模块INDEX的校验被打开。
3. ALL标识相当于一个快捷操作，即对所有模块的校验进行开启或关闭。
   ```
   gaussdb=# SET consistency_check_module='off(ALL)';
   SET
   gaussdb=# SHOW consistency_check_module;
    consistency_check_module
   --------------------------
    ALL,on(),off(INDEX)
   (1 row)

   gaussdb=# SET consistency_check_module='on(ALL)';
   SET
   gaussdb=# SHOW consistency_check_module;
    consistency_check_module
   --------------------------
    ALL,on(INDEX),off()
   (1 row)
   ```

设置建议：根据需要开启对应模块的校验。

设置不当的风险与影响：开启时会在一些SQL的执行流程中进行一致性校验操作，会有额外的耗时或性能影响，请谨慎使用。

reserve_space_for_nullable_atts

参数说明：指定是否为Ustore表的可空属性预留空间。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示为Ustore表的可空属性预留空间。
- off：表示不为Ustore表的可空属性预留空间。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

server_version

参数说明：用于查看服务器版本号（字符串形式）。

参数类型：字符串

参数单位：无

取值范围：当前有效值仅有“9.2.4”。

默认值："9.2.4"

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。该参数只用于数据库内部工具，不推荐用户使用。

设置建议：该参数不建议使用，如需查询内核版本号，可通过函数opengauss_version()获取。

设置不当的风险与影响：无

server_version_num

参数说明：用于查看服务器版本号（整数形式）。

参数类型：整型

参数单位：无

取值范围：当前有效值仅有90204。

默认值：90204

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。该参数只用于数据库内部工具，不推荐用户使用。

设置建议：该参数不建议使用，如需查询内核版本号，可通过函数opengauss_version()获取。

设置不当的风险与影响：无

block_size

参数说明：用于查看当前数据库所使用的块大小。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：当前有效值仅有8192。

默认值：8192

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

segment_size

参数说明：用于查看当前数据库所使用的段文件大小。

参数类型：整型

参数单位：8kB

取值范围：当前有效值仅有131072。

默认值：131072（即1GB）

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

max_index_keys

参数说明：用于查看当前数据库能够支持的索引键值的最大数目。

参数类型：整型

参数单位：无

取值范围：当前有效值仅有32。

默认值：32

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

integer_datetimes

参数说明：用于查看是否支持64位整数形式的日期和时间格式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持64位整数形式的日期和时间格式。

 版权所有 © 华为云计算技术有限公司

- off：表示不支持64位整数形式的日期和时间格式。

默认值：on

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

lc_collate

参数说明：用于查看当前数据库的字符串排序规则区域设置。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：依赖于集群安装部署时的配置。

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

lc_ctype

参数说明：用于查看当前数据库中字符分类和大小写转换的区域设置。如：哪些字符属于哪些字母，它对应的大写形式是什么。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：依赖于集群安装部署时的配置。

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

max_identifier_length

参数说明：用于查看当前系统允许的标识符最大长度。

参数类型：整型

参数单位：无

取值范围：当前有效值仅有63。

默认值：63

 版权所有 © 华为云计算技术有限公司

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

server_encoding

参数说明：用于查看当前数据库的服务端编码字符集。

默认情况下，gs_initdb会根据创建数据库时的系统环境初始化此参数，通过locale命令可以查看当前的配置环境。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：在创建数据库的时候由系统环境决定。

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

datanode_heartbeat_interval

参数说明：设置心跳线程间发送心跳消息的时间间隔。

参数类型：整型

参数单位：ms（毫秒）

取值范围：1000 ~ 60000

默认值：1000（即1s）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1000，表示datanode_heartbeat_interval为1000ms；带单位取值1s，表示datanode_heartbeat_interval为1s。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：建议取值不超过**wal_receiver_timeout**/2。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dfs_partition_directory_length

参数说明：在HDFS文件系统上，构造HDFS VALUE分区表的分区目录时，目录名长度的上限值。

参数类型：整型

参数单位：无

取值范围：92 ~ 7999

版权所有 © 华为云计算技术有限公司

默认值：512

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_concurrent_autonomous_transactions

参数说明：自治事务最大连接数，即同一时间自治事务执行的最大并发数。当设置为0时，将无法执行自治事务。

参数类型：整型

参数单位：无

取值范围：0 ~ 10000，理论最大值为10000，实际最大值为动态值，计算公式为“262143 - job_queue_processes - autovacuum_max_workers - max_inner_tool_connections - max_connections - AUXILIARY_BACKENDS - AV_LAUNCHER_PROCS”。其中，**job_queue_processes**、**autovacuum_max_workers**、**max_inner_tool_connections**和**max_connections**的值取决于对应GUC参数的设置；AUXILIARY_BACKENDS为预留辅助线程数，固定为20；AV_LAUNCHER_PROCS为预留autovacuum的launcher线程数，固定为2。

默认值：

- 独立部署：160（128核CPU/1024G内存）；80（60核CPU/480G内存）；40（32核CPU/256G内存）；20（16核CPU/128G内存）；10（8核CPU/64G内存，4核CPU/32G内存，4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见表**表6-1**中对应设置方法进行设置。

设置建议：根据实际业务需要和硬件配置设置此参数，建议不超过max_connections的1/10。若仅调大此参数，未同比例调整内存参数，业务压力大时，容易出现内存不足，报错提示“memory is temporarily unavailable”。

设置不当的风险与影响：若升级过程中涉及此参数范围变更，并且在commit前修改了此参数，则如果执行升级回滚，需要将此参数调整至升级前允许的范围，否则可能导致数据库无法启动。

cluster_run_mode

参数说明：双集群容灾场景标识CN/DN节点属于主集群还是备集群。单集群使用默认值。

参数类型：枚举类型

参数单位：无

取值范围：

- cluster_primary：表示节点是主集群的节点。
- cluster_standby：表示节点是备集群的节点。

默认值：cluster_primary

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：标记灾备集群使用的参数，不建议客户设置。

设置不当的风险与影响：修改后可能导致灾备集群异常。

max_resource_package

参数说明：加速数据库实例每个DN可同时运行任务的线程数的上限。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_gpi_auto_update

参数说明：控制在分区DDL命令中是否默认更新Global索引。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示默认更新Global索引，此时无论分区DDL带不带UPDATE GLOBAL INDEX子句，都会更新Global索引。
- off：表示默认不更新Global索引，此时只有当分区DDL带UPDATE GLOBAL INDEX子句时，才会更新Global索引，或enable_drop_truncate_partition_keep_gpi参数开启时，才会在DROP/TRUNCATE PARTITION操作中更新Global索引。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要进行设置。

设置不当的风险与影响：设置为off，当分区DDL未带UPDATE GLOBAL INDEX子句，且enable_drop_truncate_partition_keep_gpi参数关闭时，会使Global索引不可用。该参数必须在CN和DN设置为相同值，否则可能会导致Global索引功能异常。

enable_drop_truncate_partition_keep_gpi

参数说明：指定分区表DROP/TRUNCATE PARTITION操作中的Global索引是否保持有效。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示保持Global索引有效。此时无论DROP/TRUNCATE PARTITION语句是否带有UPDATE GLOBAL INDEX子句，都会保持Global索引有效。此优先级高于enable_gpi_auto_update参数和用户显式指定的UPDATE GLOBAL INDEX。
- off：表示不保持Global索引有效。此时仅当DROP/TRUNCATE PARTITION语句带UPDATE GLOBAL INDEX子句，或enable_gpi_auto_update参数开启时，才会保持Global索引有效。

默认值：

- 新安装的数据库默认值为on。
- 升级的数据库默认值为off。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要进行设置。

设置不当的风险与影响：设置为off时，当DROP/TRUNCATE PARTITION语句未带UPDATE GLOBAL INDEX子句且enable_gpi_auto_update参数关闭时，会使Global索引失效。该参数必须在CN和DN节点上设置为相同值，否则可能导致Global索引功能异常。

enable_gpi_fast_prune

参数说明：控制Ustore表全局分区索引在DML操作期间是否清理DDL操作遗留的垃圾元组。

参数类型：布尔型

参数单位：无

取值范围：

- on：启用清理机制，即除AutoVacuum线程清理DDL产生的垃圾元组外，在DML过程中如果索引页空间不足，也会触发DDL产生的垃圾元组清理。
- off：关闭清理机制，即只有AutoVacuum线程会清理DDL产生的垃圾元组。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要进行设置。存在高频DDL/DML操作且使用全局分区索引的分区表业务场景建议开启；未使用全局分区索引或无分区表DDL操作的业务场景建议关闭。

设置不当的风险与影响：设置为off时，只有AutoVacuum线程会清理DDL产生的垃圾元组，则GPI索引空间可能因无效元组累积引发膨胀，影响查询性能。

enable_cluster_resize

参数说明：扩容工具在内核的唯一标识，此参数开启表示当前session为扩容工具连接。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示当前session为扩容工具连接。
- off：表示当前session不是扩容工具连接。

默认值：off

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。仅允许重分布用户和初始用户在会话中设置该参数。

设置建议：此参数用于内部运维场景，请勿随意开启。

设置不当的风险与影响：用户强制设置为on时，内核认为当前session为扩容工具，可以执行扩容相关系统函数或DDL接口，存在误用风险。

change_cluster_mode

参数说明：用于表示集群是否处于模式切换过程中，其中“模式切换”是指：一主两备集群切换为一主一备一日志集群，或一主一备一日志集群切换为一主两备集群，或一主一备一日志集群中备DN和日志DN互切。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

- 0表示不处于集群模式切换过程中。
- 1表示处于集群模式切换过程中。
- 其他值无实际意义，效果等同于设置为0。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数用于控制“分布式备机读”功能是否正常可用。当设置为1时，“分布式备机读”功能将无法使用，请谨慎设置该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_partrouting_optimization

参数说明：控制是否支持分区表插入优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持分区表插入优化，针对INSERT SELECT语句，当SELECT包含常量分区键时，INSERT只需要做一次分区路由，提升性能。
- off：表示不支持分区表插入优化，插入每一条数据前都需要做一次分区路由去确定插入哪一张分区表。

默认值：on

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果关闭该参数，部分插入分区表的场景会没有该参数带来的性能优化效果，无其他影响。

说明

分区表插入优化约束：

- 只支持Insert select语句。
- insert into ta select from tb中，插入a表中所有分区键的值，对应select结果集中的列值必须是常量。
  - insert into ta select c,d from b where tb.c='1'或select '1' as c, d from tb则可以判断结果列tb.c为常量。
  - insert into ta select c,d from b where tb.c=func('1'); 如果func()不是volatile函数，也不是包含非常量参数的stable/immutable函数，则可以判断tb.c为常量。
  - 如果列值为聚合函数的返回值，如：insert into ta select count(c),d from b where tb.c='1' group by c, d; 不能判断count(c)为常量。
- 分区键为自增列、生成列、或带默认值的列时，不支持该优化。
- 不支持upsert子句。
- 不支持包含before和instead触发器的表，因为可能改变要插入的数据。

enable_cross_partition_dead_row_cache

参数说明：设置是否开启跨分区死行清理。主要针对Astore表大分区表场景，通过将不同分区的死元组TID合并清理，减少GPI的访问频次，将清理耗时与分区数解耦，仅与数据量有关，即在同数据量的前提下，清理分区表的耗时与普通表的耗时接近或为同数量级，该特性在大分区表以及GPI数量较多的场景有相对明显的优化。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启跨分区死行清理。
- off：表示关闭跨分区死行清理。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会造成带GPI的Astore表的VACUUM/AUTOVACUUM清理缓慢。

enable_art_tid_cache

参数说明：设置是否开启使用ART缓存死行TID。支持使用Adaptive Radix Tree结构存储TID以减少VACUUM内存消耗，使得相同内存开销下可以缓存更多的TID，减少大表的索引扫描次数；同时降低搜索TID的时间复杂度，提升索引清理效率。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示开启ART缓存死行TID。
- off：表示关闭ART缓存死行TID。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能会造成Astore表的VACUUM/AUTOVACUUM清理速度变慢。

storage_debug_parameter

参数说明：控制存储模块的调试参数，仅用在DEBUG版本。

参数类型：字符串

参数单位：无

取值范围：由存储配置参数组成的字符串，以分号分隔。例如："oid_threshold=100000;xid_stepsize=10"。

- oid_threshold：申请oid的最大值。最小为100000，最大为4200000000，默认为4200000000。
- oid_stepsize：申请oid的推进步长。最小为1，最大为4200000000，默认为1。
- xid_stepsize：申请事务xid的推进步长。最小为1，最大为100000，默认为1。
- toast_tuple_threshold：触发创建toast表需要的元组宽度。最小为100，最大为7000，默认为2000。

默认值：""。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_unique_checking_of_unusable_index

参数说明：控制当表中存在UNUSABLE的唯一索引时，插入和更新操作是否需要阻塞。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示当表中存在UNUSABLE的唯一索引时，INSERT语句和涉及到索引的UPDATE语句会报错，提示先处理UNUSABLE的唯一索引。
- off：表示当表中存在UNUSABLE的唯一索引时，INSERT语句和涉及到索引的UPDATE语句不会报错。

默认值：off

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据业务需要进行设置，推荐设置为on。

设置不当的风险与影响：如果打开该参数，为了保证唯一性约束，业务可能会存在大量报错；如果关闭该参数，会导致表中插入唯一索引列的重复数据，导致该索引无法直接重建。

说明

该参数打开时：

- 对于astore表，更新非UNUSABLE索引的索引列时也会报错，因为会产生新的元组。
- 对于ustore非分区表，更新非UNUSABLE索引的索引列时不会报错；如果是非原地更新，更新非UNUSABLE索引的索引列时也会报错。
- 对于ustore分区表，UNUSABLE的唯一索引是全局索引时，更新分区键后若新数据与老数据不在同一分区，也会报错，因为会产生新的元组。
- 对于ustore分区表，UNUSABLE的唯一索引是包含分区键的LOCAL索引时，更新分区键也会报错。

enable_verify_datafiles

参数说明：控制是否开启数据文件误删校验功能。该功能仅支持集群正常运行期间对数据文件（base、global、pg_tblspc、undo目录中文件）进行误删除检测；误删检测仅包括文件被误删除（rm）或者被重命名（mv）两种场景检测。误删数据文件和检测之间存在时间差，若在此期间数据库停机，则无法检测到误删文件。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开数据文件误删校验功能。
- off：表示关闭数据文件误删校验功能。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议根据实际业务需要进行设置。开启后，因需维护正常删除文件列表并进行遍历校验，可能对性能产生一定影响。此外，检测能力依赖系统内核inotify能力。最大监控目录数（含每层子目录）为fs.inotify.max_user_watches，最大并发处理消息数为fs.inotify.max_queued_events；若同时删除的文件数超过max_queued_events，可能导致消息队列溢出与丢失文件删除告警。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

disable_indirect_table_updates

参数说明：控制是否禁止在同一个query内，通过主语句、CTE、子查询、Trigger等对同一张表执行多次DML修改操作。

该参数主要用于避免同一个query内多次修改同一张表可能带来的执行顺序依赖、Trigger递归修改、数据结果不确定以及一致性风险。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：禁止同一个query内对同一张表执行多次DML修改操作。
- off：允许同一个query内对同一张表执行多次DML修改操作。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议根据实际业务需要进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 开启参数时，对于同一张表，行级触发器中禁止再次执行DML操作；但当外层DML为INSERT且触发器内的同表DML也为INSERT时，该场景允许执行。
- 触发器嵌套自更新场景下，如果两个触发器分别在CN和DN上调用，该校验不生效。

enable_hashbucket

参数说明：指定CREATE TABLE语句是否默认创建hashbucket表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示默认创建hashbucket表。
- off：表示默认不创建hashbucket表。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，仅在需要默认使用hashbucket表的场景下可设置为on。

设置不当的风险与影响：当该参数修改为on后，执行CREATE TABLE语句将默认创建段页式的hashbucket表而非普通页式表。hashbucket表与普通页式表在数据存储方式、空间占用、缓冲区使用及其他特性支持情况等规格和约束方面有一定差异（详情请参见《开发指南》中“特性使用指导 > bucket分布表”章节）。请在明确需要使用hashbucket表的场景下，充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

- enable_hashbucket参数的影响范围仅包括CREATE TABLE语句与CREATE TABLE AS语句。
- 在下列场景中，enable_hashbucket参数将不生效：
  - 在PG_CATALOG Schema下创建表。
  - 集群初始用户执行相关建表语句。
  - 执行CREATE TABLE LIKE语句且指定了INCLUDING RELOPTIONS或INCLUDING ALL（此时新表是否为hashbucket表将与源表保持一致）。
  - 要创建的表不是持久化表（例如：临时表、全局临时表、UNLOGGED表）。
  - 要创建的表不是HASH分布表（例如：复制表、RANGE分布表等）。
  - 用户在建表语句中通过WITH子句指定了存储参数segment为off或hashbucket为off。
- 当通过**表6-1**中的方式一或方式二设置enable_hashbucket参数时，建议将集群所有节点设置为相同值。若各个节点上的参数值不同，参数是否生效将由下发建表语句的CN上的参数值来决定。
- 升级观察期请勿设置该参数为on。

vacuum_truncate

参数说明：控制是否开启表的清理时截断连续空闲尾页功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启表的清理时截断连续空闲尾页功能。
- off：表示关闭表的清理时截断连续空闲尾页功能。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：参数设置为off时，清理流程不会截断空闲尾页，可能会造成Astore表的空间膨胀。

说明

- 该参数仅对Astore表生效，对hashbucket表和segment表不生效。
- 该参数与表级选项vacuum_truncate冲突时，表级选项优先：
  - 表级vacuum_truncate设为on，全局参数设为off时，该表打开VACUUM TRUNCATE功能。
  - 表级vacuum_truncate设为off，全局参数设为on时，该表关闭VACUUM TRUNCATE功能。
  - 未设置表级参数时，由全局参数控制是否打开VACUUM TRUNCATE功能。

6.3.27 等待事件

enable_instr_track_wait

参数说明：设置是否开启等待事件信息实时收集功能。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开等待事件信息收集功能。
- off：表示关闭等待事件信息收集功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为on。

设置不当的风险与影响：不开启该参数会导致无法实时收集等待事件，不便于及时定位问题。

说明

enable_instr_track_wait参数受**track_activities**约束：track_activities参数关闭时，不记录等待事件。

6.3.28 Query

instr_unique_sql_count

参数说明：控制系统中unique sql信息实时收集功能的最大收集数量。

该值由大变小时，系统将会清空原有的数据重新统计，由小变大则不会重新统计。

当系统中产生的unique sql信息（由dbe_perf.statement/dbe_perf.summary_statement统计）大于instr_unique_sql_count时，系统产生的unique sql信息不被统计。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647，0表示不启用unique sql信息收集功能。

默认值：200000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置较大会增加dbe_perf.statement视图占用的内存上限；设置过小可能出现视图中记录语句过少，导致缺少问题定位手段，影响对慢SQL的诊断。

instr_unique_sql_size

参数说明：用于控制unique sql、full sql、asp功能记录query信息的长度上限。如果实际语句字节长度大于该值，会进行截断，其中字节长度按照执行该query时所在数据库的字符集计算。

参数类型：整型

参数单位：字节（byte）

版权所有 © 华为云计算技术有限公司

取值范围：100 ~ 102400

默认值：1024

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置较大会增加dbe_perf.statement视图占用的内存上限；设置过小可能出现视图中记录语句不完整，导致缺少问题定位手段，影响对慢SQL的诊断。

instr_unique_sql_combination_options

参数说明：Unique SQL同类语句归并功能配置项，该参数的值由若干个配置项用逗号隔开构成。

开启该特性会将同类语句的Unique SQL ID归一，并将生成的Unique SQL String做归一化处理。

参数类型：字符串

参数单位：无

取值范围：参照**表6-52**。

说明

- 归并功能配置时，按照**表6-52**设置字符串，多个配置项之间用逗号隔开，例如：SET instr_unique_sql_combination_options='in_clause';
- 参数设置为空时，即为不开启该功能，例如：SET instr_unique_sql_combination_options='';

版权所有 © 华为云计算技术有限公司

表 6-52 归并功能配置项

归并功能配置项	归并功能行为控制
in_clause	控制select in()语句的in中仅含有固定参数与预编译绑定参数的情况进行归并。 例1：select * from example_table where column in (1,2,3)； 归并后Unique SQL String：select * from example_table where column in (1…n); 例2：select * from example_table where column in ($1,$2,$3)； 归并后Unique SQL String：select * from example_table where column in ($1…$n); 例3：select * from example_table where column in (1,2,$1,3,$2); 归并后Unique SQL String：select * from example_table where column in (1...n,$1...$n); 例4：select * from example_table where (column1, column2) in ((1, 'a'), (2, 'b'), (3, 'c')); 归并后Unique SQL String：select * from example_table where (column1, column2) in ((1...n));
forbid_select_for_update	控制select语句for update子句是否参与归一化计算。 该选项打开后for update / for share / for update wait n / for update nowait / for update skip lock在归一化视图中分别展示，且生成不同的unique_sql_id；该选项关闭后，上述所有子句与不带for update子句的语句归一成同一句，共用同一个unique_sql_id。
temp_table	控制local临时表unique_sql_id计算方式。 若该选项关闭，local临时表dml语句在计算unique_sql_id时采用表oid作为表的标识，不同会话创建的同名临时表执行相同操作时会生成不同unique_sql_id，在unique sql相关视图与函数中会分开独立展示。若业务语句创建临时表数量可控，或有会话力度分析临时表相关操作执行情况的诉求，建议关闭该选项。 若该选项开启，local临时表dml语句在计算unique_sql_id时采用表名字符串哈希值作为表的标识（此时同表名不同表结构的表无法进行区分），不同会话创建的同名临时表执行相同操作时会生成相同unique_sql_id，在unique sql相关视图与函数中可以合并展示。若业务语句中有大量同名local临时表的相同操作，建议开启该选项以避免生成过多记录。

默认值："in_clause,forbid_select_for_update,temp_table"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：修改该参数值将使涉及归并的查询语句的Unique SQL ID发生改变，会对已创建SQLpatch的SQL语句产生影响。

说明

- 升级时默认保持升级前参数设置值，新实例下发时默认值为"in_clause, forbid_select_for_update,temp_table"。
- 对于bigint、real、float4、blob、numeric、decimal、number、dec、integer类型的值，在in()中含有单个参数与多个参数的两种情况，会生成两种不同的unique_sql_id。
- 打开forbid_select_for_update选项后，for update wait n子句中常量n不参与归一化，常量不同记为不同语句。（其中for update wait 0与for update nowait两个子句语义相同，归一为同一条，dbe_perf.statement/gs_asp/statement_history等视图中语句字段记录执行时间更早的那一条语句。）
- 更改instr_unique_sql_combination_options后，变更选项相关语句生成的unique_sql_id与之前不同，原有sql patch可能失效，需要根据更改后新的unique_sql_id重新打sql patch。

instr_unique_sql_track_type

参数说明：unique sql记录sql方式。

参数类型：枚举类型

参数单位：无

取值范围：

- top：只记录顶层sql。
- all：记录所有sql。

默认值：all

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

- 该参数默认为all，与设置为top时相比，参数值为all时会额外记录存储过程内语句的统计信息，对存储过程的执行会产生一定的性能损耗，对于存储过程内简单表达式等执行较快的语句产生的性能影响会更加严重。
- 该参数为top时，存储过程内的open cursor语句会被记录。

unique_sql_retention_time

参数说明：清理unique sql哈希表的间隔，默认为30分钟。

参数类型：整型

参数单位：min（分钟）

取值范围：1 ~ 3650

默认值：30

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

版权所有 © 华为云计算技术有限公司

设置建议：不建议用户修改该参数，该参数当前版本无功能控制点。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

enable_instr_rt_percentile

参数说明：控制是否开启计算系统中80%和95%的SQL响应时间的功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示打开sql响应时间信息计算功能。
- off：表示关闭sql响应时间信息计算功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议开启该参数。

设置不当的风险与影响：若关闭该参数无法记录80%和95%的SQL响应时间，导致无法直观监测系统性能。

percentile

参数说明：sql响应时间百分比信息，后台计算线程会根据该参数取值计算相应的百分比信息。

参数类型：字符串

参数单位：无

取值范围：当前有效值只有“80,95”。

默认值："80,95"

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

instr_rt_percentile_interval

参数说明：sql响应时间信息计算间隔，sql响应时间信息计算功能打开后，后台计算线程每隔设置的时间进行一次计算。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

默认值：10

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示instr_rt_percentile_interval为60s；带单位取值1min，表示instr_rt_percentile_interval为1min。取值如果要带单位，必须为s、min、h。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_instr_cpu_timer

参数说明：控制是否捕获sql执行的cpu时间消耗。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示捕获sql执行的cpu时间消耗。
- off：表示不捕获sql执行的cpu时间消耗。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议开启该参数。

设置不当的风险与影响：若关闭该参数，可能导致无法记录系统中sql执行消耗的cpu时间，不利于慢sql分析信息。

perf_directory

参数说明：perf_directory决定性能视图打点任务输出文件的目录，仅sysadmin用户可以访问。它可以是绝对路径，或者是相对路径（相对于数据目录的路径）。

说明

- 合法路径：用户对此路径有读写权限。
- 非法路径：用户对此路径无读写权限。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：安装时指定。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：当配置文件中perf_directory的值为非法路径时，会导致集群无法重新启动。

enable_stmt_track

参数说明：控制是否开启Full/Slow SQL捕获功能。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启Full/Slow SQL捕获功能。
- off：表示关闭Full/Slow SQL捕获功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_stmt_parameter

参数说明：开启track_stmt_parameter后，在statement_history中记录的执行语句不再进行归一化操作，可以显示完整SQL语句信息，辅助DBA进行问题定位；其中对于简单查询，显示完整语句信息；对于PBE语句，显示完整语句信息的同时，追加每个变量数值信息，格式为“query string；parameters:$1=value1,$2=value2,...”，该参数提供目的是为用户呈现完整SQL信息，为避免query参数为长字符串或大对象类型时占用过多内存，可以通过设置GUC参数进行内存控制。全量SQL信息的内存控制分为两部分，SQL语句部分受**instr_unique_sql_size**参数控制，SQL参数部分受**track_stmt_mem_ctl**参数控制。开启该参数后，statement_history中P报文可能不会记录query信息。修改该参数值不会对正在执行的SQL记录产生影响。当语句为PBE类型且执行SQL Bypass逻辑时，参数会直接下发到DN，因此在CN查询statement_history无法获取完整语句信息；同时由于DN无Query字符串信息，因此在DN查询statement_history也无法获取完整语句信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启显示完整SQL语句信息的功能。
- off：表示关闭显示完整SQL语句信息的功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若SQL语句普遍较长，会有更大的内存和磁盘的空间占用。

track_stmt_session_slot

参数说明：设置一个session缓存的最大的全量SQL和慢SQL的数量，超过这个数量，新的语句执行将不会被跟踪，直到落盘线程将缓存语句落盘，留出空闲的空间。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

版权所有 © 华为云计算技术有限公司

默认值：1000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：每个被占用的slot内存约800bit左右，具体值由语句复杂度决定。业务压力大的场景，若该值过大，会有内存快速膨胀的影响。该值过小时，可能会遗漏需要分析的语句。

track_stmt_details_size

参数说明：设置单语句可以收集的最大的执行事件的大小。该值影响statement_history表中details字段的锁信息。

参数类型：整型

参数单位：byte

取值范围：0 ~ 100000000

默认值：4096

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_stmt_retention_time

参数说明：组合参数，用于控制全量SQL/慢SQL在表statement_history中的记录保留时间和备机已落盘文件的保留时间。系统以60秒为周期读取该参数，并执行清理超过保留时间的记录，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：

该参数分为两部分，形式为'full sql retention time, slow sql retention time'，这两个部分的具体含义如下：

- full sql retention time：全量SQL保留时间，取值范围为0 ~ 86400，单位为秒。
- slow sql retention time：慢SQL的保留时间，取值范围为0 ~ 604800，单位为秒。

默认值："3600,604800"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果设置保留时间太长，可能会占用较多空间；太短可能会影响数据保留时间太短，导致缺少问题定位手段，影响诊断。

版权所有 © 华为云计算技术有限公司

track_stmt_stat_level

参数说明：控制语句执行跟踪的级别。

参数类型：字符串

参数单位：无

取值范围：

该参数分为两部分，形式为'full sql stat level, slow sql stat level'，这两个部分的具体含义如下：

- 第一部分为全量SQL跟踪级别，取值范围为OFF、L0、L1、L2。
- 第二部分为慢SQL的跟踪级别，取值范围为OFF、L0、L1、L2。

说明

若全量SQL跟踪级别值为非OFF，则当前SQL跟踪级别值为全量SQL和慢SQL的较高级别（L2 > L1 >L0）（query_plan字段仍以原跟踪级别为准），级别说明请参见**表7-153**。

默认值："OFF,L0"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：打开全量SQL跟踪功能会影响性能，并可能会占用大量的磁盘空间。

track_stmt_standby_chain_size

参数说明：组合参数，控制备机快/慢SQL记录的最大占用内存与磁盘空间。仅SysAdmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：

该参数分为四部分，形式为'Full SQL memory size, Full SQL disk size, Slow SQL memory size, Slow SQL disk size' ，这四个部分的具体含义如下：

Full SQL与Slow SQL分开存放于不同位置，因此使用了四个值进行控制。

- Full SQL memory size：保留的快SQL的最大内存占用空间，取值范围为[16, 1024]，单位为MB。
- Full SQL disk size：保留的快SQL的最大磁盘占用空间，取值范围为[512, 1048576]，单位为MB。
- Slow SQL memory size：保留的慢SQL的最大内存占用空间，取值范围为[16, 1024]，单位为MB。
- Slow SQL disk size：保留的慢SQL的最大磁盘占用空间，取值范围为[512, 1048576]，单位为MB。

其中内存值不可大于磁盘值。

默认值："32, 1024, 16, 512"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：如果设置参数值过大，可能会占用大量的内存或磁盘空间；参数值过小，可能会遗漏需要分析的语句。

track_stmt_flush_mode

参数说明：控制全量SQL的存储模式。

参数类型：字符串

参数单位：无

取值范围：

该参数分为两部分，形式为'full sql flush mode, slow sql flush mode'，这两个部分的具体含义如下：

- 第一部分为全量SQL跟踪模式，取值可为MEMORY、FILE。参数设置为MEMORY，会记录全量SQL语句到内存中；参数设置为FILE，会记录全量SQL语句到磁盘文件中。
- 第二部分为慢SQL跟踪模式，当前版本取值仅为FILE。参数设置为FILE，会记录慢SQL语句到磁盘文件中。

默认值："FILE,FILE"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果设置参数为“MEMORY,FILE”会打开内核支持全量SQL功能，会占用一定共享内存，内存大小由**track_stmt_shm_size**参数控制。

track_stmt_shm_size

参数说明：控制全量SQL共享内存的大小。

参数类型：整型

参数单位：byte

取值范围：134217728 ~ 1073741824

默认值：134217728

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

track_stmt_mem_ctl

参数说明：控制运维各子功能的内存上限。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：该参数分为3个部分，'MESSAGE_MEM,GSTRACE_MEM,SQL_PARAM_MEM'，具体含义如下：

- MESSAGE_MEM：语句在JDBC侧执行时，会收集各个时间点，发送到数据库。该key用于控制数据库中最大接收字节，接收的信息存储在表DBE_PERF.statement_history的driver_info字段。取值范围256 ~ 4096，单位为字节。
- GSTRACE_MEM：函数打点信息会被记录到内存。该key用于控制每个sql语句记录函数打点记录的内存大小，信息存储在表DBE_PERF.statement_history的details字段。取值范围0 ~ 104857600，单位为字节。
- SQL_PARAM_MEM：控制SQL语句参数部分占用的内存大小，即仅在开启**track_stmt_parameter**参数后有效，对于PBE语句，显示完整语句信息的同时，追加每个变量数值信息，格式为“query string；parameters: $1=value1,$2=value2...”此时控制parameters部分占用的内存。取值范围为0或者100 ~ 102400，单位为字节，0表示追加的参数长度不受控制，记录全量参数内容。

默认值："MESSAGE_MEM=256,GSTRACE_MEM=2048,SQL_PARAM_MEM=1024"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致占用过多内存。设置过小可能出现视图中记录语句过少，导致缺少问题定位手段，影响对慢SQL的诊断。

driver_traceid_level

参数说明：控制表DBE_PERF.statement_history的traceid的生成方式。

参数类型：字符串

参数单位：无

取值范围："none"、"transaction"

- "none"：表示启用业务的自定义traceid，保持前向兼容性。
- "transaction"：表示同一事务内的语句使用相同的traceid。

默认值："none"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请根据业务需要调整取值。

设置不当的风险与影响：无

track_stmt_wait_driver_threshold

参数说明：语句在JDBC执行时间的统计信息，在该参数时间范围内传输到数据库时，统计信息会在数据库处理并落盘，记录在表DBE_PERF.statement_history的driver_info字段。如果超时，则丢弃这部分信息。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 60000，0表示不启用unique sql信息收集功能。

版权所有 © 华为云计算技术有限公司

默认值：5000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60000，表示track_stmt_wait_driver_threshold为60000ms；带单位取值1min，表示track_stmt_wait_driver_threshold为1min。取值如果要带单位，必须为ms、s、min。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置较大会导致执行异常的SQL语句长时间占用内存，导致无法统计新执行的SQL语句的信息，同时占用大量内存。

enable_gstrace

参数说明：内核GSTrace全链路开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启内核GSTrace全链路。
- off：关闭内核GSTrace全链路。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无

concurrent_dml_mode

参数说明：对于启用了更新分布列功能的表（即表属性“enable_update_distkey”为“on”），针对该表上出现的DML并发执行冲突（锁冲突），指定冲突的处理方式。

该参数仅在DN节点上生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：宽松模式，即出现冲突后，后获取锁的DML语句不报错。
- 1：严格模式，即出现冲突后，后获取锁的DML语句将会终止，并输出错误信息。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：可参考取值范围描述进行设置。若预期冲突后，后获取锁DML语句不报错，则可设置为0；若预期冲突后，后获取锁的DML语句终止且报错，则可设置为1。

设置不当的风险与影响：当参数设置为0时，可能引发后获取锁的DML语句无法正确获取需要更新的数据，在并发更新时可能发生预期以外的行为，比如：后执行的语句虽执行成功但未实际生效或使用MERGE INTO语法误插入多余一条数据等行为，请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

sql_hash_abort_list

参数说明：基于GUC创建SQL异常阻断补丁。

参数类型：字符串

参数单位：无

取值范围：单个SQL Hash或者用逗号隔开的多个SQL Hash。

默认值：''

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置该参数后可能影响性能，但不超过5%。

说明

- 仅在CN上检查是否阻断，DN上不生效。
- 支持的SQL Hash有效范围是[1，4294967295]。
- 可配置的SQL Hash上限为1000条，仅前1000条生效。
- 基于SQL Hash配置的异常阻断补丁可能会出现极小概率（1 /4294967295）的哈希冲突，导致不相关语句被阻断，建议用户自行评估风险。
- 所配置的SQL Hash不区分库和Schema，不同库和Schema下的相同SQL都会被阻断。
- 当多条SQL执行路径中包含同一条内部执行的SQL语句时，若对该内部SQL语句执行阻断操作，将导致所有依赖此内部SQL语句的外部SQL语句均被阻断，请谨慎使用。

6.3.29 系统性能快照

enable_perf_trigger

参数说明：设置是否开启秒级CPU冲高火焰图功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开数据库秒级CPU冲高火焰图功能。
- off：关闭数据库秒级CPU冲高火焰图功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

perf_trigger_limit

参数说明：设置由CPU使用率冲高触发火焰图采集的阈值。当系统CPU资源占用率超过该参数取值时，会生成CPU冲高火焰图。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：%

取值范围：0 ~ 100

默认值：90

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

perf_trigger_increase

参数说明：设置由CPU使用率秒级升高触发火焰图采集的阈值。当系统CPU资源占用率相比前一秒升高的比例超过该参数取值时，会生成CPU秒级冲高火焰图。

参数类型：整型

参数单位：%

取值范围：0 ~ 100

默认值：30

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_wdr_snapshot

参数说明：设置是否开启数据库监控快照功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开数据库监控快照功能。
- off：关闭数据库监控快照功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议在以下场景打开：

- 在多表和多库的场景下，WDR snapshot会在每个库串行记录快照，记录快照时间会很长，不建议打开。
- 在多表的场景下，由于内部的一些表的性能问题，pg_stat_all_tables在多表的场景下，查询很慢，这时WDR snapshot生成速度就会很慢，所以不建议打开。
- 如果在大量DDL期间执行WDR snapshot生成操作可能会执行失败，不建议打开。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_wdr_snapshot_standby

参数说明：设置分布式备机是否支持WDR功能，开启该参数后，WDR snapshot会获取备机的性能数据，并存储在主机。该参数需要在GTM FREE模式下生效，即**enable_gtm_free**取值为on时生效。

参数类型：布尔型

参数单位：无

取值范围：

- on：分布式备机支持WDR功能。
- off：分布式备机不支持WDR功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

设置enable_wdr_snapshot_standby参数为on需要满足以下条件：

- **wal_level**参数为logical或者hot_standby。
- **hot_standby**参数取值为on。
- 分布式备机读的模式下**gtm_option**=2或者**enable_gtm_free**=on。

enable_show_standby_name

参数说明：设置是否开启显示区分同一分片主备机名字的功能，该名字区分主备节点。在主备DN上设置该参数，查询性能视图，视图中的node_name可以区分主备机，开启该功能后在同一分片的不同节点上查询pgxc_node_str()，返回不同的名字。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启视图区分主备机名字功能。
- off：关闭视图区分主备机名字功能。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_wdr_cache

参数说明：该参数的功能为，针对WDR报告后台线程调整GUC参数**enable_gsstat_cache**状态。当enable_wdr_cache设置为on时，WDR报告后台线程会将enable_gsstat_cache设置为on，否则设置为off。

版权所有 © 华为云计算技术有限公司

参数类型：布尔型

参数单位：无

取值范围：

- on：将enable_gsstat_cache设置为on。
- off：将enable_gsstat_cache设置为off。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置为默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

wdr_snapshot_retention_days

参数说明：系统中数据库监控快照数据的保留天数。当数据库运行过程期间所生成的快照数量超过保留天数内允许生成的快照数量的最大值（默认为24*8=192）时，系统将每隔wdr_snapshot_interval时间间隔，清理一次snapshot_id最小的快照数据。

说明

当且仅当**enable_wdr_snapshot**参数取值为on时，该参数设置有效。

参数类型：整型

参数单位：d（天）

取值范围：1 ~ 30

默认值：8

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数取值越大，磁盘占用越高。

wdr_snapshot_query_timeout

参数说明：系统执行数据库监控快照操作时，设置快照操作相关的sql语句的执行超时时间。如果语句超过设置的时间没有执行完并返回结果，则本次快照操作失败，但会将已经生成的部分快照保留下来。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：100

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置，0表示不作超时限制。例如，不带单位取值60，表示wdr_snapshot_query_timeout为60s；带单位取值1min，表示wdr_snapshot_query_timeout为1min。取值如果要带单位，必须为s、min、h、d。

版权所有 © 华为云计算技术有限公司

设置建议：根据需求场景进行配置。如果不关心快照中的数据，希望快速生成快照，可以适当地将该参数取值调小；如果需要快照中的数据尽量完备，可以相应地将该参数取值调大。

设置不当的风险与影响：在生成快照超时的前提下，该参数取值越大，超时时间越长，会导致生成快照的时间变长，但快照内的数据更完备。

wdr_snapshot_interval

参数说明：后台线程Snapshot自动对数据库监控数据执行各项快照操作的时间间隔。

说明

当且仅当**enable_wdr_snapshot**参数取值为on时，该参数设置有效。

参数类型：整型

参数单位：min（分钟）

取值范围：10 ~ 60

默认值：60

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示wdr_snapshot_interval为60min；带单位取值1h，表示wdr_snapshot_interval为1h。取值如果要带单位，必须为min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：在固定保留天数下，该参数取值越小，磁盘占用量越大。

wdr_snapshot_space_threshold

参数说明：快照空间占用维度的控制阈值。当快照占用空间达到该参数取值的50%时，会启用数据库的控制逻辑，使快照占用空间趋于平稳。

说明

- 当且仅当**enable_wdr_snapshot**参数取值为on时，该参数设置有效。
- 若将该参数取值改得比当前快照占用空间要小，并不会使快照占用空间回缩，而是开启控制逻辑，控制快照增长，使快照占用空间趋于平稳。

参数类型：整型

参数单位：kB

取值范围：0 ~ 107374182400

默认值：0，表示不开启按空间阈值控制快照占用空间的功能。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值1024，表示wdr_snapshot_space_threshold为1024kB；带单位取值1MB，表示wdr_snapshot_space_threshold为1MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：根据需求场景进行配置。

设置不当的风险与影响：设置不当可能导致存量快照数量仅剩最新一组快照。

版权所有 © 华为云计算技术有限公司

wdr_snapshot_full_backup_interval

参数说明：WDR snapshot做全量快照的间隔。该参数控制的不是时间间隔，而是间隔个数。例如，当取值为12时，每一组先生成一个全量快照，再生成11个增量快照；当取值为1时，每个快照都是全量快照。

说明

- 当且仅当**enable_wdr_snapshot**参数取值为on时，该参数设置有效。
- 若在已经生成若干个快照的情况下，更改该参数取值，下一个快照会是全量快照。例如，WDR snapshot当前已经生成了5个增量快照，此时更改参数取值为10，那么下一个快照会为全量快照，然后开启新的周期。
- 在删除快照的时候，会遵循一组快照一起删除的规则。删除快照的数量受最早的一组快照产生时该参数的取值影响，与该参数的当前取值无关。

参数类型：整型

参数单位：个

取值范围：1 ~ 24

默认值：12

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：根据需求场景进行配置。

设置不当的风险与影响：该参数取值越小，全量快照越多，同样空间占用下能保留的总快照数量越少。

enable_asp

参数说明：设置是否开启ASP（Active Session Profile，活跃会话报告）功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开ASP功能。
- off：关闭ASP功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_asp_standby

参数说明：设置是否开启ASP支持备机功能。开启该功能后，ASP报告进行采样时，会同步获取备机的采集数据。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：开启ASP支持备机功能，ASP获取所有备机上的采集数据。
- off：关闭ASP支持备机功能。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asp_flush_mode

参数说明：ASP刷新到磁盘上的方式分为写文件和写系统表，当取值为“file”时，默认写文件，为“table”时写系统表，为“all”时，即写文件也写系统表，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：当前有效值有“table”、“file”、“all”。

默认值："table"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asp_flush_rate

参数说明：当内存中样本个数达到**asp_sample_num**时，会按一定比例把内存中样本刷新到磁盘上，asp_flush_rate为刷新比例。该参数为10时表示按10:1进行刷新。

参数类型：整型

参数单位：无

取值范围：1 ~ 10

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置值过小会导致刷盘数据量变大。

asp_flush_interval

参数说明：设置ASP定时刷盘的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：300 ~ 3600

版权所有 © 华为云计算技术有限公司

默认值：300

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示asp_flush_interval为60s；带单位取值1min，表示asp_flush_interval为1min。取值如果要带单位，必须为s、min、h。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asp_log_filename

参数说明：当ASP写文件时，生成文件名的格式，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：合法的文件名格式字符串。

默认值："asp-%Y-%m-%d_%H%M%S.log"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asp_retention_days

参数说明：当ASP样本写到系统表时，样本保留的最大天数。

参数类型：整型

参数单位：d（天）

取值范围：1 ~ 7

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：取值过大会导致系统表gs_asp占用磁盘变大。

asp_sample_interval

参数说明：每次采样的间隔。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 10

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值10，表示asp_sample_interval为10s；带单位取值10s，表示asp_sample_interval为10s。取值如果要带单位，必须为s。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asp_sample_num

参数说明：LOCAL_ACTIVE_SESSION视图的最大样本个数，仅sysadmin用户可以访问。

参数类型：整型

参数单位：无

取值范围：10000 ~ 100000

默认值：100000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asp_log_directory

参数说明：**asp_flush_mode**设置为all或者file时，asp_log_directory决定存放服务器asp日志文件的目录。它可以是绝对路径，或者是相对路径（相对于数据目录的路径），仅sysadmin用户可以访问。

说明

- 合法路径：用户对此路径有读写权限。
- 非法路径：用户对此路径无读写权限。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：安装时指定。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请在充分测试验证后进行设置。

设置不当的风险与影响：当配置文件中asp_log_directory的值为非法路径时，会导致集群无法重新启动。

gs_perf_interval

参数说明：设置自动采集堆栈功能的采集时间间隔。

参数类型：浮点型

参数单位：min（分钟）

取值范围：0 ~ 60。0表示关闭自动采集堆栈功能。

版权所有 © 华为云计算技术有限公司

说明

有效值为0.2 ~ 60，若尝试设置为小于0.2的非0值，会自动将该参数调整到默认值5。

默认值：5

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示gs_perf_interval为60min；带单位取值1h，表示gs_perf_interval为1h。取值如果要带单位，必须为min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：若参数设置过小，会导致采集频率较高，对系统性能可能有一定影响，且日志数量较多，会占用大量存储空间。请在充分理解参数含义，并经过测试验证后进行修改。

gs_perf_deploy_mode

参数说明：获取gs_perf_service采集进程部署状态。

参数类型：字符串

参数单位：无

取值范围："ROOT"、"DBUSER"、"DISABLE"

- "ROOT"：采集进程部署为ROOT状态，支持所有的火焰图采集功能。
- "DBUSER"：采集进程部署为DBUSER状态，只支持oncpu相关的火焰图采集功能。
- "DISABLE"：采集进程部署为DISABLE状态，不提供火焰图功能。

默认值："ROOT"

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

gs_perf_freq

参数说明：设置自动采集堆栈功能的采集频率。

参数类型：整型

参数单位：Hz

取值范围：10 ~ 1000

默认值：999

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

gs_perf_retention_days

参数说明：设置火焰图文件的保留时长。

参数类型：整型

参数单位：d（天）

取值范围：1 ~ 8

默认值：3

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.30 黑匣子相关参数

blackbox_number

参数说明：控制“$GAUSSLOG/gs_blackbox/{nodename}”目录下最多能够保留的黑匣子文件个数。

参数类型：整型

参数单位：无

取值范围：0 ~ 10，其中0表示不使用黑匣子功能。

默认值：5（256核CPU/2048G内存，192核CPU/1536G内存，128核CPU/1024G内存，104核CPU/1024G内存，96核CPU/1024G内存，96核CPU/768G内存，80核CPU/640G内存，64核CPU/512G内存，60核CPU/480G内存，32核CPU/256G内存，16核CPU/128G内存，8核CPU/64G内存，4核CPU/32G内存）；0（4核CPU/16G内存）。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：参数值设置为0时，黑匣子功能关闭，不会对$GAUSSLOG/gs_blackbox/{nodename}目录下的文件进行管理控制，gs_blackbox工具功能不受影响。

blackbox_directory

参数说明：黑匣子临时文件的存储目录。

参数类型：字符串

参数单位：无

取值范围：合法目录的格式字符串。目录的文件系统需要是tmpfs类型。

默认值："/dev/shm"

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。可以通过“df -TH [path]”命令查询所设置的文件系统类型。当前用户需要有新指定目录的读写权限。设置参数成功重启节点后，原来目录下的文件需要手动清理。

6.3.31 安全配置

enable_security_policy

参数说明：安全策略开关，控制统一审计和动态数据脱敏策略是否生效。

说明

- 统一审计：统一审计机制是一种通过定制化制定审计策略而实现高效安全审计管理的一种技术。当管理员定义审计对象和审计行为后，用户执行的任务如果关联到对应的审计策略，则生成对应的审计行为，并记录审计日志。
- 动态数据脱敏：动态数据脱敏机制是一种通过定制化制定脱敏策略从而实现对隐私数据保护的一种技术，可以有效地在保留原始数据的前提下解决非授权用户对敏感信息的访问问题。

参数类型：布尔型

参数单位：无

取值范围：

- on：安全策略开关打开。
- off：安全策略开关关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：需要使用安全策略时设置为on，会占用系统资源，对系统性能产生影响。

unified_audit_location

参数说明：控制统一审计日志记录到具体的位置。

参数类型：整型

参数单位：无

取值范围：0、1

- 0：表示统一审计日志记录到rsyslog配置的位置。
- 1：表示统一审计日志记录到传统审计日志文件。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。当使用统一审计记录到审计日志文件的功能，先确认GUC参数audit_enabled为开启状态。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：需要统一审计记录到审计日志文件时设置为1，会对传统审计功能使用的资源产生挤占。

use_elastic_search

参数说明：统一审计发送日志至Elastic Search系统，**enable_security_policy**打开且本参数打开后，统一审计日志会通过http（https）传递至Elastic Search系统（默认使用https安全协议）。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启统一审计日志发送至Elastic Search。
- off：关闭统一审计日志发送至Elastic Search。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，需要将统一审计日志发送至Elastic Search系统时开启。

设置不当的风险与影响：此参数打开后需要保证elastic_search_ip_addr对应的es服务可正常连通，否则use_elastic_search参数配置不生效。

elastic_search_ip_addr

参数说明：Elastic Search系统IP地址，使用https协议格式为：https://ip:port:username；使用http协议格式为：http://ip:port。其中，ip为Elastic Search服务器的IP，port为Elastic Search HTTP通信的侦听端口，范围为9200 - 9299，username为用户在Elastic Search注册账号所使用的用户名，初始用户为elastic。使用https协议需要配置相关证书。

参数类型：字符串

参数单位：无

取值范围：合法IP字符串或空字符串。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，需要将统一审计日志发送至Elastic Search系统时且use_elastic_search为on时设置。

设置不当的风险与影响：需要保证该参数取值对应的es服务可正常连通，否则use_elastic_search参数配置不生效。

is_sysadmin

参数说明：表示当前用户是否是初始用户。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示是初始用户。
- off：表示不是初始用户。

默认值：off

设置方式：该参数属于INTERNAL类型参数，为固定参数，用户无法修改此参数，只能查看。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

enable_tde

参数说明：透明数据加密功能开关。创建加密表前需要将此参数置为on，并且通过**tde_key_info**参数配置密钥信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启透明数据加密功能。
- off：关闭透明数据加密功能。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置此参数为on，需要保证tde_key_info参数中密钥信息配置正确，并且密钥服务可以正常访问，否则透明加密功能无法使用。

设置不当的风险与影响：已存在加密数据前提下，请谨慎关闭该功能。关闭后无法读取加密表数据并回放加密日志，进而可能导致节点FATAL/PANIC故障、数据库重启失败或不可用问题。

tde_key_info

参数说明：开启透明加密功能后，透明数据运行时需访问外部密钥服务实现密钥管理，该参数配置用于访问密钥服务的相关信息，如服务地址、身份认证信息、项目信息等。

参数类型：字符串

参数单位：无

取值范围：请联系技术支持。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请根据实际使用的密钥服务进行设置。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：已存在加密数据前提下，请谨慎修改该参数值。错误配置意味着密钥丢失，无法读取加密表数据并回放加密日志，进而可能导致节点FATAL/PANIC故障、数据库重启失败或不可用问题。

tde_request_kms_count

参数说明：开启透明加密功能后，用于设置透明加密访问密钥管理服务尝试的次数。

参数类型：整型

参数单位：次

取值范围：1~10

默认值：3

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：请保持默认值。

设置不当的风险与影响：设置为1时，只会尝试请求密钥管理服务1次，无法抵御网络闪断、抖动等问题。

tde_index_default_encrypt

参数说明：创建索引时，如果开启本参数，并且索引的基表是加密表，则数据库会自动将索引设置为加密索引，并自动为索引复制基表的加密算法、密钥等加密参数，对索引中的数据先加密再存储。

参数类型：布尔型

参数单位：无

取值范围：

- on：自动对加密表的索引设置加密参数。
- off：不自动对加密表的索引设置加密参数。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果所有加密表的索引都需要被加密，可开启该参数，无需手动将索引设置为加密索引。

设置不当的风险与影响：如果将参数设置为on，在加密表上创建的索引将被自动加密，影响查询索引的性能。

tde_encrypt_config

参数说明：用于设置透明加密支持Xlog和undo-log相关功能的参数。

参数类型：字符串

参数单位：无

取值范围：该参数配置格式为“key=value, key=value, ...”，合法的key和value如下：

版权所有 © 华为云计算技术有限公司

key取值	value取值	缺省值	功能
log_encrypt	on或off	on	控制是否对Xlog和undo-log加密。
log_algorithm	aes_128_ctr或sm4_ctr	aes_128_ctr	对Xlog和undo-log加密的算法。
table_algorithm	aes_128_ctr、sm4_ctr、aes_128_ctr_sha256_hmac、sm4_ctr_sm3_hmac或aes_128_gcm	aes_128_ctr	在该种语法中：CREATE .. SET (enable_tde=on, encrypt_algo=aes_128_ctr)，如果不主动设置encrypt_algo参数，默认使用table_algorithm中设置的参数。

默认值：""

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

block_encryption_mode

参数说明：aes_encrypt和aes_decrypt函数进行加解密时使用的块加密模式。

参数类型：枚举类型

参数单位：无

取值范围：有效值有aes-128-cbc、aes-192-cbc、aes-256-cbc、aes-128-cfb1、aes-192-cfb1、aes-256-cfb1、aes-128-cfb8、aes-192-cfb8、aes-256-cfb8、aes-128-cfb128、aes-192-cfb128、aes-256-cfb128、aes-128-ofb、aes-192-ofb、aes-256-ofb。其中aes表示加/解密算法，128/192/256表示密钥长度（单位：bit），cbc/cfb1/cfb8/cfb128/ofb表示块加/解密模式。

默认值：aes-128-cbc

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：影响aes_encrypt和aes_decrypt函数的使用。

enable_mac_check

参数说明：表示基于标签的强制访问控制是否生效。

参数类型：布尔型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：

- on：表示基于标签的强制访问控制生效。
- off：表示基于标签的强制访问控制不生效。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

allow_initialuser_exec_others_udf

参数说明：表示是否允许初始用户执行其他用户定义的函数或存储过程。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许初始用户执行其他用户定义的函数或存储过程。
- off：表示不允许初始用户执行其他用户定义的函数或存储过程。

默认值：

- 直接安装新版本时，默认值为off
- 从历史版本升级上来的数据库，如果升级的基础版本不支持该GUC参数，默认值为on；若升级前的基础版本支持该参数，升级后保持该参数当前已设置值。

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置该参数为off，可以降低越权风险。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

allow_sysadmin_exec_others_udf

参数说明：表示是否允许系统管理员执行其他用户定义的函数或存储过程。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许系统管理员执行其他用户定义的函数或存储过程。
- off：表示不允许系统管理员执行其他用户定义的函数或存储过程。

默认值：on

设置方式：该参数属于SUSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置该参数为off，可以降低越权风险，系统管理员可以根据需要在会话中设置参数为on。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_rls_match_index

参数说明：控制是否支持在行级访问控制特性使用场景中，基于目标谓词条件对基表执行索引扫描。目标场景为：基表设置并开启了RLS（Row Level Security）策略，查询谓词中包含unleakproof类型系统函数或like操作符。

参数类型：布尔型

参数单位：无

取值范围：

- on：支持目标场景下的基表索引扫描。
- off：不支持目标场景下的基表索引扫描。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。若业务热点查询语句为目标场景，且基表行级访问策略数量较少，可打开该开关以显著提升查询性能。

设置不当的风险与影响：此参数设置为on，在位图扫描算子切换为lossy模式且RLS策略数目较多时，会导致过滤谓词重复执行，影响性能。

说明

- 该参数的修改，会影响目标场景下执行计划的生成，可通过重连、创建操作符等方式手动使缓存计划失效。
- 该参数开启后，会影响到位图扫描算子的生成，RLS策略谓词将被插入到Recheck过滤条件中；因此，在该算子切换为lossy模式且RLS策略数目较多时，会对性能产生一定影响。

restrict_nonsystem_relation_kind

参数说明：控制是否能访问用户创建的关系对象类型。

参数类型：字符串

参数单位：无

取值范围：

- ""：表示不进行任何限制。
- "view"：表示禁用对用户创建的视图进行访问。
- "foreign-table"：表示禁用对用户创建的外表进行访问。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置为默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

在gs_dump执行导出的过程中，为防止恶意攻击，需要禁用访问用户创建的视图和外表，工具会将此参数的值会话级设置为'view, foreign-table'，无需用户手动配置。

enable_tde_async_encryption

参数说明：透明数据异步加密功能开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启透明数据异步加密功能。
- off：关闭透明数据异步加密功能。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：若业务在开启透明加密功能后有显著性能下降，且接受所有新建数据表使用同一异步加密数据密钥，建议开启该参数以有效提升性能。

设置不当的风险与影响：无。

tde_async_dek

参数说明：用于存储透明数据异步加密密钥信息，包括主密钥ID与数据密钥密文两部分内容。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：""

设置方式：该参数属于SIGHUP类型参数，不支持用户设置该参数。

设置建议：该参数由数据库自行填充，不支持用户设置该参数。

设置不当的风险与影响：无。

tde_dkcache_remain_time

参数说明：用于设置透明加密数据密钥缓存有效时长。

参数类型：整型

参数单位：h（时）

取值范围：0 ~ 1440

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：建议设置为默认值，不做调整。

设置不当的风险与影响：设置为0或过大值可能导致数据库内存中长期驻留不被使用的透明加密数据密钥缓存。

说明

- 该参数设置为零值，则透明加密数据密钥缓存永久生效，不会从内存中清理。
- 该参数设置为非零值，则透明加密数据密钥缓存仅在超过tde_dkcache_remain_time设置的小时数而未被使用时，才会从内存中清理（每小时周期性清理）。

enable_fips

参数说明：FIPS模式功能开关。启用FIPS模式需要将此参数置为on，并确保运行环境已开启FIPS模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启FIPS模式。
- off：关闭FIPS模式。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置此参数为on，需要确保存量数据中没有使用FIPS不支持的算法，否则会导致不可预期的异常。

设置不当的风险与影响：设置此参数为on后，存量使用FIPS不支持算法的数据将无法使用（包括但不限于导入导出、备份恢复等数据场景），会导致不可预期的异常。不建议在有数据的环境中开启FIPS模式，如需开启务必确保存量数据未使用FIPS不支持算法。

说明

- 该参数设置为on后，GUC参数password_encryption_type参数设置为0、1、3时将失效，强制使用sha256算法加密用户密码 。历史密码使用md5、sm3加密的用户将无法登录。
- GUC参数设置为on，仅支持对加解密函数、用户创建、登录、TLS通信算法依据FIPS标准进行拦截报错，如需进入FIPS模式还需通过配置使OpenSSL进入FIPS模式，二者缺一不可，否则可能导致不可预期行为。

enable_thread_permission_down

参数说明：后台线程权限控制开关，开启后后台线程使用gs_role_signal_backend角色权限，关闭后使用初始用户权限。

参数类型：布尔型

参数单位：无

取值范围：

- on：开启低权限模式。

版权所有 © 华为云计算技术有限公司

- off：关闭低权限模式。

默认值：

- on：新安装的数据库的默认值。
- off：从507.0.0之前版本升级到当前版本时的默认值。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：设置此参数为on，防止可能出现的后台线程越权风险。

设置不当的风险与影响：无。

enableSeparationOfDuty

参数说明：是否开启三权分立选项。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启三权分立。
- off：表示不开启三权分立。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：此参数设置为on，系统管理员权限会受限制，部分之前可以执行的操作会无法执行。

enable_nonsysadmin_execute_direct

参数说明：是否允许非系统管理员和非监控管理员执行EXECUTE DIRECT ON语句。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许任意用户执行EXECUTE DIRECT ON语句。
- off：表示只允许系统管理员和监控管理员执行EXECUTE DIRECT ON语句。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_access_server_directory

参数说明：是否允许非初始用户创建、修改和删除DIRECTORY对象。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许非初始用户创建、修改和删除DIRECTORY对象。
- off：表示不允许非初始用户创建、修改和删除DIRECTORY对象。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

须知

- 用户在使用高级包UTL_FILE访问服务器端文件时，要求必须拥有所指定的DIRECTORY对象的权限。
- 出于安全考虑，默认情况下，只有初始用户才能够创建、修改、删除DIRECTORY对象。
- 如果开启了enable_access_server_directory，具有SYSADMIN权限的用户和继承了内置角色gs_role_directory_create权限的用户可以创建directory对象；具有SYSADMIN权限的用户、directory对象的属主、被授予了该directory的DROP权限的用户或者继承了内置角色gs_role_directory_drop权限的用户可以删除directory对象；具有SYSADMIN权限的用户和directory对象的属主可以修改directory对象的所有者，且要求该用户是新属主的成员。

6.3.32 HyperLogLog

hll_default_log2m

参数说明：该参数可以指定hll数据结构桶的个数。桶的个数会影响hll计算distinct值的精度，桶的个数越多，误差越小。误差范围为：$[-1.04/2^{hll\_default\_log2m*1/2},+1.04/2^{hll\_default\_log2m*1/2}]$。

参数类型：整型

参数单位：无

取值范围：10 ~ 16

默认值：14

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：设置过小可能导致无法达到所需的精度要求，而设置过大则可能导致资源浪费。

版权所有 © 华为云计算技术有限公司

hll_default_log2explicit

参数说明：该参数可以用来设置从Explicit模式到Sparse模式的默认阈值大小。

参数类型：整型

参数单位：无

取值范围：0 ~ 12

- 0：表示跳过Explicit模式。
- 1 ~ 12：表示在基数到达$2^{hll_default_log2explicit}$时切换模式。

默认值：10

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：设置过小可能导致无法达到所需的精度要求，而设置过大则可能导致资源浪费。

hll_default_log2sparse

参数说明：该参数可以用来设置从Sparse模式到Full模式的默认阈值大小。

参数类型：整型

参数单位：无

取值范围：0 ~ 14

- 0：表示跳过Sparse模式。
- 1 ~ 14：表示在基数到达$2^{hll_default_log2sparse}$时切换模式。

默认值：12

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：设置过小可能导致无法达到所需的精度要求，而设置过大则可能导致资源浪费。

hll_duplicate_check

参数说明：该参数可以用来指定是否默认开启duplicate check。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示默认关闭。
- 1：表示默认开启。

默认值：0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值，可以根据实际业务需要进行调整。

设置不当的风险与影响：开启可能会消耗更多资源来进行duplicate check，但会提高distinct值计算的准确性。

6.3.33 用户自定义函数

udf_memory_limit

参数说明：控制每个CN、DN执行UDF时可用的最大物理内存量。该参数当前版本不生效。

参数类型：整型

参数单位：kB

取值范围：200*1024 ~ 2147483647

默认值：204800（即200MB）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为204800，不带单位，表示204800kB；设置为300MB，带单位，表示300MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

FencedUDFMemoryLimit

参数说明：控制每个fenced udf worker进程使用的虚拟内存。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647

默认值：0，表示不做内存控制。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为200，不带单位，表示200kB；设置为3MB，带单位，表示3MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

UDFWorkerMemHardLimit

参数说明：控制FencedUDFMemoryLimit的最大值。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647

默认值：1048576（即1GB）

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。例如，设置为200，不带单位，表示200kB；设置为3MB，带单位，表示3MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_udf_execute_direct

参数说明：是否允许EXECUTE DIRECT ON语句执行用户自定义函数，高级包DBE_SQL、PKG_SERVICE、DBE_XMLGE中的函数和DBE_UTILITY.EXEC_DDL_STATEMENT。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许EXECUTE DIRECT ON语句执行用户自定义函数、高级包DBE_SQL、PKG_SERVICE、DBE_XMLGE中的函数和DBE_UTILITY.EXEC_DDL_STATEMENT。
- off：表示禁止EXECUTE DIRECT ON语句执行用户自定义函数、高级包DBE_SQL、PKG_SERVICE、DBE_XMLGE中的函数和DBE_UTILITY.EXEC_DDL_STATEMENT。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，避免出现意料之外的结果。

6.3.34 定时任务

job_queue_processes

参数说明：表示系统可以并发执行的job数目。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000

- 当job_queue_processes设置为0时，表示不启用定时任务功能，任何job都不会被执行（因为开启定时任务的功能会对系统的性能有影响，有些局点可能不需要定时任务的功能，可以通过设置为0不启用定时任务功能）。
- 当job_queue_processes大于0时，表示启用定时任务功能且系统能够并发处理的最大任务数。

启用定时任务功能后，job_scheduler线程会在定时时间间隔轮询pg_job系统表，系统设置定时任务检查周期默认为1s。

由于并行运行的任务数太多会消耗更多的系统资源，因此需要设置系统并发处理的任务数，当前并发的任务数达到job_queue_processes，且同时又有任务到期时，这些任

版权所有 © 华为云计算技术有限公司

务将因为本次得不到执行而延期到下一轮询周期。因此，建议用户需要根据每个任务的执行时长合理设置任务的时间间隔（即submit接口中的interval参数），来避免由于任务执行时间太长而导致下个轮询周期无法正常执行。

默认值：10

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果同一时间内并行的job数很多，过小的参数值会导致job等待，而过大的参数值则消耗更多的系统资源，因此建议设置此参数为100，用户可以根据系统资源情况合理调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_prevent_job_task_startup

参数说明：控制是否可以启动job线程。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示不能启动job线程。
- off：表示可以启动job线程。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.35 线程池

enable_thread_pool

参数说明：控制是否使用线程池功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启线程池功能。
- off：表示不开启线程池功能。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在低并发长连接性能敏感场景下，建议关闭该参数，其他场景下建议打开该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

enable_connect_thread_pool

参数说明：控制是否使用认证线程池功能。认证线程池是针对线程池建连模式进行的优化，存在对线程池功能的依赖。使用该功能需要先开启线程池参数**enable_thread_pool**。当且仅当enable_connect_thread_pool和**enable_thread_pool**都为true时才会开启认证线程池功能，enable_connect_thread_pool为ture而**enable_thread_pool**为false时不开启。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启认证线程池功能。
- off：表示不开启认证线程池功能。

默认值：

- on：新安装的数据库的默认值。
- off：505.2.1.SPC0800版本之前的数据库升级后的默认值。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在通常场景下建议打开该参数。

设置不当的风险与影响：开启后可优化高并发建连对SQL业务抖动的影响，但是会降低连接接入性能，如果对连接性能要求较高，应合理配置**connect_thread_pool_attr**参数及客户端超时时间。

thread_pool_attr

参数说明：用于控制线程池功能的详细属性，该参数仅在**enable_thread_pool**打开后生效，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：

该参数分为四个部分，'thread_num, group_num, cpubind_info, enable_thread_pool_auto_scaling'，这四个部分的具体含义如下：

- thread_num：线程池中的初始线程总数，可以动态扩充，取值范围是0~4096。其中0的含义是数据库根据系统CPU core的数量来自动配置线程池的线程数，如果参数值大于0，线程池中的线程数等于thread_num。线程池大小建议根据硬件配置进行设置，计算公式如下：thread_num = CPU核数*(3~5)，thread_num最大值为4096。
- group_num：线程池中的线程分组个数，取值范围是0~64。其中0的含义是数据库根据系统NUMA组的个数来自动配置线程池的线程分组个数，如果参数值大于0，线程池中的线程组个数等于group_num。
- cpubind_info：线程池是否绑核的配置参数。可选择的配置方式有：1. '(nobind)' ，线程不做绑核；2. '(allbind)'，利用当前系统所有能查询到的CPU core做线程绑核；3. '(nodebind: 1, 2)'，利用NUMA组1、2中的CPU core进行绑核；4. '(cpubind: 0-30)'，利用0-30号CPU core进行绑核；5. '(numabind:

版权所有 © 华为云计算技术有限公司

0-30)'，在NUMA组内利用0-30号CPU core进行绑核。当指定了CPU绑核时，为了保证绑核生效，应保证NUMA组和group_num一致。该参数不区分大小写。

- enable_thread_pool_auto_scaling：是否开启线程池自动扩缩容功能。可以不配置，配置1表示开启，配置0表示关闭，默认不配置此字段，表示开启线程池自动扩缩功能。

默认值：

- 独立部署：'2048,2,(nobind)'（128核CPU/1024G内存）；'1024,2,(nobind)'（60核CPU/480G内存，32核CPU/256G内存）；'512,2,(nobind)'（16核CPU/128G内存）；'256,2,(nobind)'（8核CPU/64G内存）；'128,2,(nobind)'（4核CPU/32G内存）；'64,2,(nobind)'（4核CPU/16G内存）

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：内存充足且CPU性能好的情况，当业务需要更多连接时可以增加该参数值，修改前需要在测试环境验证通过。

设置不当的风险与影响：若thread_pool_attr设置值大于等于max_connections，会导致proc资源全部被线程池占用，出现proc资源不足的问题。默认值即推荐值，不建议修改。修改必须详细确认参数的规格限制，并考虑硬件资源是否足够，否则可能导致集群异常。

说明

必须确保**thread_pool_attr**配置的线程数与static_thread_pool_num配置的线程数之和不超过min(4096, **max_connections**)，且static_thread_pool_num必须大于DN分片数乘以group_num，否则无法成功创建静态池。

connect_thread_pool_attr

参数说明：用于控制建连线程池功能的详细属性，该参数仅在**enable_thread_pool**和**enable_connect_thread_pool**打开后生效，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：'auto'或由'thread_num, group_num, cpubind_info, enable_adjust_connect_pool'配置项组成的字符串两种形式。

- 取值为'auto'形式时，表示建连线程池参数将由数据库根据部署机器的规格自动计算后设置。其中4核CPU规格场景下，实际生效为'8,1,(cpubind: *绑定任意两个有效CPU*)'；其他规格场景下，实际生效为'8,1,(nobind)'。
- 取值为由配置项组成的字符串形式时，格式为'thread_num, group_num, cpubind_info, enable_adjust_connect_pool'，配置项的具体含义如下：
  - thread_num：建连线程池中的初始线程总数，可以动态扩充，取值范围是[1, Max(8, CPU核数)]，下限为1，上限为8和当前部署机器CPU核数中的较大值，即如果当前部署机器的总CPU核数为4，理论计算出的建连线程池线程上限为4，小于8，则上限为8，此时thread_num取值范围为[1, 8]；如果当前部署机器的总CPU核数为64，理论计算出的建连线程池线程上限为64，大于8，则上限为64，此时thread_num取值范围为[1, 64]。支持超配，超配上限为500，如果thread_num配置值大于扩容上限，则扩容功能自动关闭，建连线程池线程数将不会变化，如果thread_num配置值小于扩容上限，则根据enable_adjust_connect_pool字段值进行判断是否开启扩容功能。

版权所有 © 华为云计算技术有限公司

  - group_num：建连线程池中的线程分组个数，固定为1，不允许设置为其他值。
  - cpubind_info：建连线程池是否绑核。可选择的配置方式有：
    - '(nobind)'：建连线程池不进行绑核。
    - '(allbind)'：利用当前系统所有能查询到的CPU进行线程绑核。
    - '(nodebind: 1, 2)'：利用NUMA组1、2中的CPU进行线程绑核。
    - '(cpubind: 0-30)'：利用0~30号CPU进行线程绑核。
    - '(numabind: 0-30)'：在NUMA组内利用0~30号CPU进行线程绑核。该参数不区分大小写。
  - enable_adjust_connect_pool：是否关闭建连线程池扩容功能。该配置项可缺省，默认为0，表示开启线程池扩容功能；配置为1时，表示主动关闭。若触发建连线程池过载，过载时的应急扩容不受该参数控制。

默认值：'auto'

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：内存充足且CPU性能好的情况下，当业务需要更多连接时可以增加该参数值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改，当参数设置超过取值范围时，会导致gaussdb进程无法正常启动。

说明

- 调整该参数中的thread_num值对系统会产生相应的影响，当thread_num调小时，建连线程池占用的资源少，在大并发建连场景下，建立连接性能会下降，系统的抗过载能力会提升；当thread_num调大时，建连线程池会占用更多的系统资源，在大并发建连场景下，建立连接性能会提升，系统的抗过载能力会受到部分影响。请根据实际业务需求在参数允许的范围内进行合理配置。
- 当建连线程池持续2秒无法分配线程处理建立连接的会话时，将触发线程池过载。在过载状态下，系统优先响应最新的建连会话，同时强制关闭等待时间已超过2秒的建连会话。

static_thread_pool_num

参数说明：线程个数，表示用于创建静态线程池（静态池）的线程个数。该参数仅在**enable_thread_pool**打开后的分布式CN上生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 1024

默认值：10% * group_num * thread_num，其中group_num和thread_num请参见**thread_pool_attr**。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

- 创建静态线程池用于防止DN实例耗尽CN线程池线程，保证有可用线程连接DN执行业务。
- 必须确保**thread_pool_attr**配置的线程数与static_thread_pool_num配置的线程数之和不超过min(4096, **max_connections**)，且static_thread_pool_num必须大于DN分片数乘以group_num，否则无法成功创建静态池。
- static_thread_pool_num为虚拟值，设置后不会立刻生效。当CN线程池满载或者耗尽，CN会主动创建static_thread_pool_num个线程，这些线程构成静态池。
- 静态线程池包含在整个线程池内部，配置了静态线程池后，业务线程可用的最大数目就会被相应地减少，例如配置静态线程池大小为100，线程池使用上限为500时，实际可使用的正常线程数目上限为400。
- 静态线程池也占用连接数，因此会存在静态线程池导致连接达不到配置的max_connections的情况，此时可以关闭静态线程池来规避此情况。
- CN静态池用于单DN连接业务，静态池对集群每个DN分片线程数量均分，计算公式为：static_thread_pool_num/n（n为DN分片数量），使用超过计算公式的固定配额线程数则报错。
- 若静态池线程被多DN连接业务持有则报错并归还线程。
- 静态池创建不支持多租模式下非默认group线程组。
- static_thread_pool_num参数为0表示关闭静态池功能，若业务不希望由静态池管控多DN连接可以考虑关闭该功能。
- 创建静态线程池功能约束：前提为分布式CN开启线程池，当前功能解决DN分片故障的场景，保障CN仍有可用的线程给业务使用，实现业务不中断。由单CN查询导致的线程池占满不在设计解决范围内，同时对业务仅在单CN查询的情况不纳入管控和统计。

thread_pool_stream_attr

参数说明：用于控制stream线程池功能的详细属性，stream线程只在DN生效，该参数仅在**enable_thread_pool**打开后生效，仅sysadmin用户可以访问。

参数类型：字符串

参数单位：无

取值范围：

该参数分为四个部分，'stream_thread_num, stream_proc_ratio, group_num, cpubind_info'，这四个部分的具体含义如下：

- stream_thread_num：stream线程池中的线程总数，取值范围是0~4096。其中0的含义是数据库根据系统CPU core的数量来自动配置线程池的线程数，如果参数值大于0，线程池中的线程数等于stream_thread_num。线程池大小推荐根据硬件配置设置，计算公式如下：stream_thread_num = CPU核数*(3~5)，stream_thread_num最大值为4096。
- stream_proc_ratio：预留给stream线程的proc数量比例，浮点型，默认为0.2，预留proc计算方式为：stream_proc_ratio * stream_thread_num。
- group_num：线程池中的线程分组个数，取值范围是0~64。其中0的含义是数据库根据系统NUMA组的个数来自动配置线程池的线程分组个数，如果参数值大于0，线程池中的线程组个数等于group_num。thread_pool_stream_attr的group_num需与thread_pool_attr的group_num配置和使用保持一致，若设置为不同值，以thread_pool_attr的group_num为准。
- cpubind_info：线程池是否绑核的配置参数。可选择的配置方式有：1. '(nobind)'，线程不做绑核；2. '(allbind)'，利用当前系统所有能查询到的CPU core做线程绑核；3. '(nodebind: 1, 2)'，利用NUMA组1、2中的CPU core进行绑

版权所有 © 华为云计算技术有限公司

核；4. '(cpubind: 0-30)'，利用0-30号CPU core进行绑核；5. '(numabind: 0-30)'，在NUMA组内利用0-30号CPU core进行绑核。该参数不区分大小写。thread_pool_stream_attr的cpubind_info需与thread_pool_attr的cpubind_info配置和使用保持一致，若设置为不同值，以thread_pool_attr的cpubind_info为准。

默认值：

- stream_thread_num：16
- stream_proc_ratio：0.2
- group_num、cpubind_info：详情请参见**thread_pool_attr**。

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在内存充足且CPU性能好的情况下，当业务需要更多并发性能时可以增加该参数值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_stream_threads

参数说明：当运行stream线程中除sync quit状态外的线程数（活跃stream线程数）超过max_stream_threads上限，等待降低到max_stream_threads范围内再进行复用或拉起新stream线程。

当stream线程数超过max_stream_threads后，默认分批退出空闲stream，直到stream线程数小于等于max_stream_threads或不存在空闲stream线程。

该参数仅在**enable_thread_pool**打开后生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 4096

默认值：

独立部署：

4096（192核CPU/1536G内存，128核CPU/1024G内存，104核CPU/1024G内存，96核CPU/1024G内存）；2048（96核CPU/768G内存，80核CPU/640G内存）；1024（64核CPU/512G内存，60核CPU/480G内存，32核CPU/256G内存）；128（16核CPU/128G内存）；64（8核CPU/64G内存）；32（4核CPU/32G内存）；16（4核CPU/16G内存）

混合部署：

2048（192核CPU/1536G内存，128核CPU/1024G内存，104核CPU/1024G内存，96核CPU/1024G内存）；1024（96核CPU/768G内存，80核CPU/640G内存）；512（80核CPU/512G内存，72核CPU/576G内存，64核CPU/512G内存，60核CPU/480G内存，32核CPU/256G内存）；64（16核CPU/128G内存）；32（8核CPU/64G内存）；16（4核CPU/32G内存）；16（4核CPU/16G内存）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在内存充足且CPU性能良好的情况下，若业务需要更高的并行执行性能，可适当增加该参数值，反之，则减少。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：

- 若设置该参数值过小，会导致无法生成并行执行计划，或使执行计划因等待可用stream线程而排队，增加查询延迟。
- 若设置该参数值过大，而CPU核数较少时，可能会引发stream线程占用大量CPU资源，影响其他业务性能。

说明

max_stream_threads用于单个DN的stream线程上限管控，设置为0表示关闭管控。

stream_resource_control_attr

参数说明：包含cpu_control_threshold、stop_stream_create_threshold和os_run_queue_threshold三部分，分别控制进行CPU上限控制的阈值，对新stream业务默认报错的阈值，操作系统中处于可运行状态的总线程数超过CPU核数的倍数阈值。

该参数仅在**enable_thread_pool**打开后生效。

参数类型：字符串

参数单位：无

取值范围：

该参数分为三个部分，'cpu_control_threshold, stop_stream_create_threshold, os_run_queue_threshold'，这三个部分的具体含义如下：

- cpu_control_threshold：控制进行CPU上限控制的阈值，取值范围为0~100。
- stop_stream_create_threshold：对新stream业务默认报错的阈值，取值范围为0~100。
- os_run_queue_threshold：操作系统中处于可运行状态的总线程数超过CPU核数的倍数阈值，取值范围为0~100。

cpu_control_threshold、stop_stream_create_threshold、run_queue_threshold均为0时表示该功能关闭。

当cpu_control_threshold、stop_stream_create_threshold均生效时，需满足cpu_control_threshold <= stop_stream_create_threshold。

默认值：

- cpu_control_threshold：0
- stop_stream_create_threshold：0
- os_run_queue_threshold：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

当CPU使用率超过stop_stream_create_threshold时，尝试启动新stream业务会产生报错“Stream worker is temporarily unavailable. Try later...”。

版权所有 © 华为云计算技术有限公司

thread_pool_idle_time

参数说明：用于限制线程池线程的空闲时间，当线程池线程空闲时间超过该参数的设置时，则线程退出，并释放相关的内存资源。该参数仅在**enable_thread_pool**打开后生效，仅sysadmin用户可以访问。该参数不作用于认证线程池和静态线程池的线程。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：300

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数影响线程池的线程数量，不建议设置过小。设置该参数越小，线程池的线程启停可能越频繁，会影响性能。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

resilience_threadpool_reject_cond

参数说明：用于控制线程池过载逃生的线程池使用率比例。该参数仅在GUC参数**enable_thread_pool**和**use_workload_manager**打开时生效。

参数类型：字符串

参数单位：无

取值范围：

该参数分为两部分，'recover_threadpool_percent, overload_threadpool_percent'，这两个部分的具体含义如下：

- recover_threadpool_percent：线程池恢复正常状态时的线程池使用率，当线程池使用率小于该值时，停止过载逃生并放开新连接接入，取值为0~INT_MAX，设置为多少表示百分之多少。
- overload_threadpool_percent：线程池过载时的线程池使用率，当线程池使用率大于该值时，表示当前线程池已经过载，触发过载逃生kill会话并禁止新连接接入，取值为0~INT_MAX，设置为多少表示百分之多少。

示例：

```
resilience_threadpool_reject_cond = '50,90'
```

表示线程池使用率超过90%后，禁止新连接接入并kill堆积的会话，kill会话过程中线程池使用率下降到50%时，停止kill会话并允许新连接接入。kill会话策略为：优先kill空闲会话；再将会话按照会话状态（会话状态相同则按照会话开始时间从早到晚）排序，依次kill会话。

默认值：'0,0'，表示关闭线程池逃生功能。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

说明

- 线程池使用率可以通过DBE_PERF.local_threadpool_status视图查询获得；线程池设置的初始线程池线程数目可以通过查询thread_pool_attr参数获得。
- 该参数如果设置的百分比过小，则会频繁触发线程池过载逃生流程，会使正在执行的会话被强制退出，新连接短时间接入失败，需要根据实际线程池使用情况慎重设置。
- recover_threadpool_percent和overload_threadpool_percent的值可以同时为0，除此之外，recover_threadpool_percent的值必须要小于overload_threadpool_percent，否则不会生效。
- use_workload_manager参数关闭的情况下，如果打开bypass_workload_manager，该参数也会生效，但是因为bypass_workload_manager是SIGHUP类型，用reload方式设置后需要重启数据库才会使得当前功能生效。
- 将参数从其它值动态设为"0,0"后，若此时已启动会话查杀，不会终止查杀动作，可等待10s以上再连接数据库防止新连接被查杀。

6.3.36 全文检索

ngram_gram_size

参数说明：ngram解析器分词的长度。

参数类型：整型

参数单位：无

取值范围：1 ~ 4

默认值：2

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无。

说明

该参数无法影响pg_catalog.ngram和其他显示指定了gram_size的全文检索配置的解析行为。需要通过ALTER TEXT SEARCH CONFIGURATION来修改已创建的配置的相关选项。

ngram_grapsymbol_ignore

参数说明：ngram解析器是否忽略图形化字符。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示忽略图形化字符。
- off：表示不忽略图形化字符。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：无。

ngram_punctuation_ignore

参数说明：ngram解析器是否忽略标点符号。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示忽略标点符号。
- off：表示不忽略标点符号。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无。

default_text_search_config

参数说明：设置全文检索的配置信息。当被文本搜索函数使用时，这些函数并没有一个明确指定的配置。

参数类型：字符串

参数单位：无

取值范围：“pg_catalog.english”、“pg_catalog.simple”、“pg_catalog.ngram”、“pg_catalog.pound”。

- pg_catalog.english、pg_catalog.simple：适用于英文分词场景。
- pg_catalog.ngram：适用于无语义中文分词场景。
- pg_catalog.pound：适用于无语意但待解析文本以固定分隔符分割开来的中英文分词场景。

说明

当与环境相匹配的配置文件确定时，gs_initdb会选择一个与环境相对应的设置来初始化配置文件。

默认值：pg_catalog.english

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：无

设置不当的风险与影响：如果设置为不存在的文本搜索配置时将会报错。如果default_text_search_config对应的文本搜索配置被删除，需要重新设置default_text_search_config，否则会报设置错误。

tsearch_options

参数说明：控制全文检索相关功能的使用。

版权所有 © 华为云计算技术有限公司

参数类型：字符串

参数单位：无

取值范围：可选配置项有configuration、dictionary、debug_function。若配置多个选项，由逗号隔开组成。

- configuration：允许使用全文检索文本配置项，包含CREATE/ALTER/DROP TEXT SEARCH CONFIGURATION相关语法。
- dictionary：允许使用全文检索词典相关语法，包含CREATE/ALTER/DROP TEXT SEARCH DICTIONARY。
- debug_function：允许使用全文检索调测函数，包含ts_debug、ts_parse、ts_token_type、ts_lexize、ts_stat函数。

默认值：

- 新安装集群默认值： ""。
- 506.0.0版本之前的数据库升级后的默认值："configuration,dictionary,debug_function"。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无。

6.3.37 备份恢复

operation_mode

参数说明：系统是否进入备份恢复模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进入备份恢复模式。
- off：表示不进入备份恢复模式。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议用户设置该参数。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

enable_cbm_tracking

参数说明：cbm追踪功能的控制开关。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示追踪功能开启。
- off：表示追踪功能关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：当使用roach执行集群的全量和增量备份时需要开启此参数，如果关闭会导致备份失败。

设置不当的风险与影响：关闭会导致增量备份失败。

max_size_for_xlog_retention

参数说明：用于控制何时触发备份复制槽或逻辑复制槽的强制推进，以避免由于备份操作或逻辑解码过程中日志无法回收导致磁盘满、集群只读等影响。

参数类型：整型

参数单位：无

取值范围：-100 ~ 2147483647

- 0表示关闭该功能。
- 负值表示按磁盘阈值触发，如-80，表示当磁盘阈值超过80%，且日志回收是由于备份操作或逻辑解码被阻塞，那么会触发备份复制槽或逻辑复制槽的强制推进。
- 正值表示按日志积压大小触发，如32，表示距离当前检查点redo位置，当备份复制槽落后了大于32段日志（每段日志大小为16MB）时，且日志回收是由于备份操作或逻辑解码被阻塞，那么会触发备份复制槽或逻辑复制槽的强制推进。

默认值：-80

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议比cm_server组件的datastorage_threshold_value_check略小一点，以避免集群进入只读状态。

设置不当的风险与影响：设置过小可能会导致备份频繁失败。

max_cbm_retention_time

参数说明：用于控制何时触发备份CBM文件的强制回收，以避免由于备份操作过程中CBM文件无法回收导致磁盘满、集群只读等影响。

参数类型：整型

参数单位：s（秒）

取值范围：86400 ~ 2147483647

默认值：1209600（即14d）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为86400，不带单位，表示86400s；设置为1d，带单位，表示1d。取值如果要带单位，必须为s、min、h、d。

设置建议：建议按照全备的时间间隔设置。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置过小可能会导致备份频繁失败。

enable_standby_move_buckets

参数说明：用于控制是否允许CN连接备机执行move buckets操作，仅在集群级恢复时使用。在恢复hashbucket重分布期间的备份集时，该参数作为CN向DN发送buckets搬迁命令时，与备机模式的DN建连使用，仅在该连接中生效，其余情况下均为off。

参数类型：布尔型

参数单位：无

取值范围：

- on：备机可接收连接。
- off：备机不可接收连接。

默认值：off

设置方式：该参数为BACKEND类型参数，但不支持gs_guc set/reload方式设置，且该参数写入配置文件中不会生效，仅作为数据恢复时的运维参数使用。

设置建议：不建议用户修改该参数。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

hadr_max_size_for_xlog_receiver

参数说明：控制归档日志下载和日志回放之间的同步差距阈值。当两者进度差超过该阈值时，系统将自动暂停归档日志下载，以避免日志堆积和资源过载。

参数类型：整型

参数单位：kB

取值范围：0 ~ 2147483647

默认值：268435456

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：不建议用户修改该参数。

设置不当的风险与影响：设置过小可能会导致归档日志下载频繁暂停。

6.3.38 Undo

Undo的GUC参数仅对Ustore有效。

undo_space_limit_size

参数说明：用于控制undo强制回收阈值，达到阈值的80%启动强制回收。建议设置undo_space_limit_size参数不小于undo_limit_size_per_transaction参数。

参数类型：整型

参数单位：页面（8kB）

取值范围：102400 ~ 2147483647，对应的默认单位为页面。

版权所有 © 华为云计算技术有限公司

默认值：33554432（即256GB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，取值1000000，表示undo_space_limit_size为1000000页；取值1000000kB，表示undo_space_limit_size为1000000kB。取值如果要带单位，必须为kB、MB、GB，不能为TB。

设置建议：一般情况保持默认值即可。如果磁盘空间很小，无法满足默认值的需求，可以按需将该参数调小；如果业务中存在长事务或大事务，需要根据磁盘情况将undo_space_limit_size调大，具体调整值可以参考gs_stat_undo系统函数的“info”列，详情请参见**表1-149**。

设置不当的风险与影响：当undo记录无法及时回收，设置偏小可能会发生undo记录强制回收，导致低概率报错 “snapshot too old! the undo record has been forcibly discard”。当磁盘空间不足时，如果undo记录无法及时回收，该参数设置偏大可能会导致数据库系统保留较多undo记录，与其他数据抢占磁盘物理空间，磁盘空间不足时有可能影响业务。

undo_limit_size_per_transaction

参数说明：用于控制单事务undo分配空间阈值，达到阈值时事务报错回滚。建议设置undo_limit_size_per_transaction参数不大于**undo_space_limit_size**参数。若设置的undo_limit_size_per_transaction参数大于undo_space_limit_size参数，用户调用show undo_limit_size_per_transaction命令查询参数值时，显示出来的值和用户设置的值仍保持一致，只是在使用时会取undo_space_limit_size和undo_limit_size_per_transaction两者的较小值，作为实际的单事务undo分配空间阈值。

参数类型：整型

参数单位：页面（8kB）

取值范围：256 ~ 2147483647，对应的默认单位为页面。如果设置的值超过了134217728（即1TB），则实际生效的值为134217728。

默认值：4194304（即32GB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，取值10000，表示undo_limit_size_per_transaction为10000页；取值10000kB，表示undo_limit_size_per_transaction为10000kB。取值如果要带单位，必须为kB、MB、GB，不能为TB。

设置建议：一般情况保持默认值即可。如果磁盘空间/内存空间很小，无法满足默认值的需求，可以按需将该参数调小；如果业务中存在长事务或大事务，单个事务内产生Undo 记录量大于该参数设置值，需要根据磁盘情况将undo_limit_size_per_transaction调大，具体调整值可以参考 gs_stat_undo 系统函数的“info”列，详情请参见**表1-149**。如果设置undo_limit_size_per_transaction超过1TB，可能会影响系统的性能和稳定性，因此如果设置的值超过了134217728（即1TB），用户调用show undo_limit_size_per_transaction命令查询参数值时，显示的值和用户设置的值保持一致，但实际生效的值为134217728。

设置不当的风险与影响：设置偏小可能会导致单事务内生成undo记录超过该阈值，报错“undo size of the transaction exceeds the threshold ”，进而事务回滚。

版权所有 © 华为云计算技术有限公司

6.3.39 DCF 参数设置

enable_dcf

参数说明：是否开启DCF模式，不建议手动修改。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示当前日志复制模式为DCF模式。
- off：表示当前日志复制模式为非DCF模式。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需要使用DCF日志复制模式时，请使用模式切换命令，切勿手动修改enable_dcf，且该参数只在DN中生效，在CN中不生效。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_ssl

参数说明：此参数不再使用，DCF复用GUC参数**ssl**。详情请参见**安全和认证**。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示使用SSL。
- off：表示不使用SSL。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_config

参数说明：DCF集群配置信息，安装阶段由OM配置，安装后不允许修改。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：安装时由OM配置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

dcf_data_path

参数说明：DCF数据路径，安装阶段由OM配置，安装后不允许修改。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：DN数据目录下的dcf_data目录。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

dcf_log_path

参数说明：DCF日志路径，安装阶段由OM配置，安装后不允许修改。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：DN数据目录下的dcf_log目录。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数不允许用户进行设置，只能查看。

设置不当的风险与影响：无

dcf_node_id

参数说明：DCF所在DN节点ID，用户安装和模式切换时自定义。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：安装时用户自定义配置。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_max_workers

参数说明：DCF回调函数线程最大个数。DCF通过回调函数调用到DN向DCF注册的函数时，需要申请共享内存以及信号量等。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：个

取值范围：0 ~ 262143

默认值：40

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致资源不足，集群安装失败。

dcf_truncate_threshold

参数说明：DN对DCF日志进行truncate的门限阈值。

参数类型：整型

参数单位：无

取值范围：1 ~ 2147483647

默认值：100000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：阈值过大可能会导致日志长期不被truncate从而挤占内存。

dcf_election_timeout

参数说明：DCF leader和follower选举超时时间。选举超时时间数值依赖于当前DN之间的网络状况，在超时时间较小且网络极差的情形下，会有超时选举发生，待网络恢复选举恢复正常。建议根据当前网络状态合理设置超时时间。对DCF节点时钟的约束：DCF节点间最大时钟差异小于选举超时时间的一半。在DCF手动选举模式下，为了不影响CM及时仲裁选举，禁止对该参数配置修改，按默认选举超时时间设置即可。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 600

默认值：3

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值60，表示dcf_election_timeout为60s；带单位取值1min，表示dcf_election_timeout为1min（60s）。取值如果要加上单位，必须为“s”，“min”。

设置建议：根据当前网络状态合理设置超时时间。

设置不当的风险与影响：网络环境不佳且超时时间较小的情况下，可能会发生超时选举。

版权所有 © 华为云计算技术有限公司

dcf_enable_auto_election_priority

参数说明：DCF优先级选主是否允许内部自动调整优先级值。

参数类型：整型

参数单位：无

取值范围：

- 0：表示不允许。
- 1：表示允许内部自动调整。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_election_switch_threshold

参数说明：DCF防频繁切主门限。推荐根据用户业务可接受的最大故障时间配置。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：根据业务可接受的最大故障时间配置。

设置不当的风险与影响：设置过大可能会导致业务长时间不可用。

dcf_run_mode

参数说明：DCF选举模式，包含自动选举模式、手动选举模式和去使能选举模式。目前去使能选举模式只限定少数派恢复场景使用，修改会导致数据库实例不可用。

说明

集群在正常状态下进行工作模式切换才能保证切换后工作正常。GUC参数设置和cm_ctl设置的DCF工作模式需要保持一致，即两者需要同步设置为DCF手动或自动模式。

例如，设置DCF手动模式如下：

```
cm_ctl set --param --server -k dn_arbitrate_mode=quorum
cm_ctl reload --param --server
gs_guc reload -Z datanode -I all -N all  -c "dcf_run_mode=1"
```

设置DCF自动模式如下：

```
cm_ctl set --param --server -k dn_arbitrate_mode=paxos
cm_ctl reload --param --server
gs_guc reload -Z datanode -I all -N all  -c "dcf_run_mode=0"
```

参数类型：枚举类型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：0、1、2

- 0：表示自动选举模式。
- 1：表示手动选举模式。
- 2：表示去使能选举模式。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：目前去使能选举模式只限定少数派恢复场景使用，修改会导致数据库实例不可用。

dcf_log_level

参数说明：DCF日志级别。

参数类型：字符串

参数单位：无

取值范围：

- 关闭日志：“NONE”，NONE表示关闭日志打印，不能与以下日志级别混合使用。
- 开启日志：“RUN_ERR|RUN_WAR|RUN_INF|DEBUG_ERR|DEBUG_WAR|DEBUG_INF|TRACE|PROFILE|OPER”

  日志级别可以从上述字符串中选取字符串并使用竖线组合使用，不能配置空字符串。

默认值："RUN_ERR|RUN_WAR|DEBUG_ERR|OPER|RUN_INF|PROFILE"

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：日志级别设置过低可能会导致日志信息缺失。

dcf_log_backup_file_count

参数说明：DCF运行日志备份保留个数。

参数类型：整型

参数单位：个

取值范围：1 ~ 1000

默认值：100

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设定值过小会导致备份较少、日志丢失。

版权所有 © 华为云计算技术有限公司

dcf_max_log_file_size

参数说明：DCF运行日志单个文件最大大小。

参数类型：整型

参数单位：MB

取值范围：1 ~ 1000。

默认值：10

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值10，表示dcf_max_log_file_size为10MB；带单位取值10MB，表示dcf_max_log_file_size为10MB。取值如果要加上单位，必须为“MB”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致单个日志文件打开、查询效率降低。

dcf_socket_timeout

参数说明：DCF通信模块连接socket超时时间，参数重启生效。对于网络环境比较差的环境，若配置很小的超时时间，可能会导致建立连接不成功，此时需要适当增大此值。

参数类型：整型

参数单位：ms（毫秒）

取值范围：10 ~ 600000（10min）

默认值：5000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值5000，表示dcf_socket_timeout为5000ms；带单位取值1min，表示dcf_socket_timeout为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”。

设置建议：推荐使用默认值。

设置不当的风险与影响：网络环境不佳时，配置过小的超时时间可能导致建立连接不成功。

dcf_connect_timeout

参数说明：DCF通信模块建立连接超时时间，参数重启生效。对于网络环境比较差的环境，若配置很小的超时时间，可能会导致建立连接不成功，此时需要适当增大此值。

参数类型：整型

参数单位：ms（毫秒）

取值范围：10 ~ 600000（10min）

默认值：60000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值5000，表示dcf_connect_timeout为5000ms；带单位取值

版权所有 © 华为云计算技术有限公司

1min，表示dcf_connect_timeout为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”。

设置建议：推荐使用默认值。

设置不当的风险与影响：网络环境不佳时，配置过小的超时时间可能导致建立连接不成功。

dcf_mec_fragment_size

参数说明：DCF通信模块fragment大小，参数重启生效。

参数类型：整型

参数单位：kB

取值范围：32 ~ 10240

默认值：64

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_stg_pool_max_size

参数说明：DCF存储模块内存池最大值，参数重启生效。

参数类型：整型

参数单位：MB

取值范围：32 ~ 2147483647

默认值：2048

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2048，表示dcf_stg_pool_max_size为2048MB；带单位取值5GB，表示dcf_stg_pool_max_size为5GB（5120MB）。取值如果要加上单位，必须为“MB”、“GB”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致内存资源紧张，设置过小可能会导致存储模块资源不足。

dcf_stg_pool_init_size

参数说明：DCF存储模块内存池最小值，参数重启生效。

参数类型：整型

参数单位：MB

取值范围：32 ~ 2147483647

默认值：32

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2048，表示dcf_stg_pool_init_size为2048MB；带单位取值5GB，表示dcf_stg_pool_init_size为5GB（5120MB）。取值如果要加上单位，必须为“MB”、“GB”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致内存资源紧张，设置过小可能会导致存储模块资源不足。

dcf_mec_pool_max_size

参数说明：DCF通信模块内存池最大值，参数重启生效。

参数类型：整型

参数单位：MB

取值范围：32 ~ 2147483647

默认值：200

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2048，表示dcf_mec_pool_max_size为2048MB；带单位取值5GB，表示dcf_mec_pool_max_size为5GB（5120MB）。取值如果要加上单位，必须为“MB”、“GB”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致内存资源紧张，设置过小可能会导致通信模块资源不足。

dcf_flow_control_disk_rawait_threshold

参数说明：DCF流控功能的磁盘等待阈值。

参数类型：整型

参数单位：us（微秒）

取值范围：0 ~ 2147483647

默认值：100000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致性能劣化，内存升高。

dcf_flow_control_net_queue_message_num_threshold

参数说明：DCF流控功能的网络队列消息数阈值。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

版权所有 © 华为云计算技术有限公司

默认值：1024

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小可能会导致性能劣化严重。

dcf_flow_control_cpu_threshold

参数说明：DCF CPU流控阈值。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：100

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小可能会导致性能劣化严重。

dcf_mec_batch_size

参数说明：DCF通信批量消息数，数值为0时，DCF会根据网络以及写入数据量自适应调整，参数重启生效。

参数类型：整型

参数单位：无

取值范围：0 ~ 1024

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致节点间通信效率下降。

dcf_mem_pool_max_size

参数说明：DCF内存最大值，参数重启生效。

参数类型：整型

参数单位：MB

取值范围：32 ~ 2147483647

默认值：2048

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2048，表示dcf_mem_pool_max_size为2048MB；带单位取值5GB，表示dcf_mem_pool_max_size为5GB（5120MB）。取值如果要加上单位，必须为“MB”、“GB”。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致集群安装失败。

dcf_mem_pool_init_size

参数说明：DCF内存初始化大小，参数重启生效。

参数类型：整型

参数单位：MB

取值范围：32 ~ 2147483647

默认值：32

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值2048，表示dcf_mem_pool_init_size为2048MB；带单位取值5GB，表示dcf_mem_pool_init_size为5GB（5120MB）。取值如果要加上单位，必须为“MB”、“GB”。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致集群安装失败。

dcf_compress_algorithm

参数说明：DCF运行日志传输压缩算法。

参数类型：整型

参数单位：无

取值范围：

- 0：表示不压缩。
- 1：表示LZ4压缩算法。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：如果需要开启LZ4压缩算法，则将dcf_compress_algorithm置为1。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_compress_level

参数说明：DCF日志传输压缩级别，此参数生效前提必须配置有效的压缩算法，即设置合法的**dcf_compress_algorithm**参数。

参数类型：整型

参数单位：无

取值范围：1 ~ 22。若不开启压缩，配置的压缩级别将不生效。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_mec_channel_num

参数说明：DCF通信通道数量，参数重启生效。

参数类型：整型

参数单位：个

取值范围：1 ~ 64

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小可能会导致节点间通信效率下降。

dcf_rep_append_thread_num

参数说明：DCF日志复制线程数量，参数重启生效。

参数类型：整型

参数单位：个

取值范围：1 ~ 1000

默认值：取每个分片中的DN副本个数

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致线程空闲，资源利用率下降。

dcf_mec_agent_thread_num

参数说明：DCF通信工作线程数量，参数重启生效。

参数类型：整型

参数单位：个

取值范围：1 ~ 1000

默认值：10

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议不少于2*节点数***dcf_mec_channel_num**。

设置不当的风险与影响：设置过大可能会导致线程空闲，资源利用率下降。

版权所有 © 华为云计算技术有限公司

dcf_mec_reactor_thread_num

参数说明：DCF使用reactor线程数量，参数重启生效。

参数类型：整型

参数单位：个

取值范围：1 ~ 100

默认值：1

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：dcf_mec_reactor_thread_num与**dcf_mec_agent_thread_num**比例建议为1:40。

设置不当的风险与影响：设置过大可能会导致线程空闲，资源利用率下降。

dcf_log_file_permission

参数说明：DCF运行日志文件属性，参数重启生效，参数安装阶段配置，后续不支持修改。若用户需要支持同组的其他用户访问日志，首先需要所有的父目录都支持同组的其他用户也能访问。即若参数dcf_log_path_permission配置为750，dcf_log_file_permission只能为600或者640。若参数dcf_log_path_permission配置为700，dcf_log_file_permission只能为600。

参数类型：枚举类型

参数单位：无

取值范围：600、640

默认值：600

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：**dcf_log_path_permission**的权限需要大于dcf_log_file_permission。

设置不当的风险与影响：设置不当可能导致用户无法访问。

dcf_log_path_permission

参数说明：DCF运行日志目录属性，参数重启生效，参数安装阶段配置，后续不支持修改。若用户需要支持同组的其他用户访问日志路径，需选择参数750，否则选择700。

参数类型：枚举类型

参数单位：无

取值范围：700、750

默认值：700

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：若用户需要支持同组的其他用户访问日志路径，需选择参数750，否则选择700。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_majority_groups

参数说明：DCF策略化多数派功能设置。对于需要配置此参数的group，该group内至少有一台备机收到日志。即该group内存在一台同步备机。若对DCF实例内做了增删节点或者对实例内节点group值进行了调整修改，需同步修改此配置。修改此参数时需保证group数值实际存在且有效。

参数类型：字符串

参数单位：无

取值范围：

- 关闭：""，空字符串表示策略化多数派功能关闭。
- 开启：配置有效的group值，使用逗号分隔，group值需在**dcf_config**中存在。例如将group值分别为1和2，加入DCF的策略化多数派配置时，可以设置为"1,2"；若配置了dcf_config中不存在的group值或者其他字符，DCF将认为该配置的group无效。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

须知

若配置了参数后某一group内所有节点均故障，在对其中某个节点做涉及节点build相关操作（节点修复、不换ip的节点替换）时，需要将该group从此参数列表中移除，待节点恢复正常后可将该group再次配置到此参数。

dcf_node_id_map

参数说明：DN备机名称与DCF node_id映射字典，参数重启生效，参数需要在安装阶段配置，后续不支持修改。在DCF集群安装、升级、节点替换场景会涉及使用此参数。GUC参数**synchronous_standby_names**中配置的standby_name需包含在此字典内。

参数类型：字符串

参数单位：无

取值范围：配置格式例如：'standby_name1:dcf_node_id1,standby_name2:dcf_node_id2'，DN备机名称对应的DCF node_id数值，且使用逗号分隔。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能会导致集群安装、升级失败。

dcf_candidate_names

参数说明：DCF候选者列表参数。即有可能被选为主机的DN节点名称。DCF自动模式下选举策略受此参数控制，不在此列表内的DN无法当选为主机。

参数类型：字符串

参数单位：无

取值范围：配置格式例如：'dn_name1,dn_name2,dn_name3'。参数依赖dcf_node_id_map，DN名称必须位于**dcf_node_id_map**中，且使用逗号分隔。

默认值：""

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_thread_effective_time

参数说明：DCF刷盘线程存活有效时间。参数用于判断磁盘是否处于I/O hang，若超过此时间DCF控制日志无法刷新，无法访问I/O资源则认为线程I/O hang，则会触发降备动作。参数配置为0表示关闭I/O hang检测。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

默认值：160

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dcf_pri_leader_timeout

参数说明：优先级选主超时时间。DCF自动模式下，开启优先级选主功能后，备机触发优先级选主，如果超出此超时时间仍然未当选为主，则放弃此轮选举。参数配置为0表示优先级选主会一直等待选举成功为止。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 3600

默认值：60

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致集群长时间处于无主状态。

dcf_static_leader_timeout

参数说明：优选旧主超时时间。DCF自动模式下，重启集群后优先选举旧主为新主，超出此超时时间仍然未成功则放弃此轮选举。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 600

默认值：60

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能会导致集群长时间处于无主状态。

6.3.40 闪回相关参数

本章节介绍闪回功能相关参数。

enable_recyclebin

参数说明：用来控制回收站的实时打开和关闭。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示实时打开回收站。
- off：表示实时关闭回收站。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：需要使用闪回表功能时，将enable_recyclebin设置为on。

设置不当的风险与影响：关闭时无法使用闪回功能。

须知

开启回收站功能后，当数据库执行表或索引删除操作时，将自动生成以“BIN”开头的隐藏对象。这些对象用于在后续恢复过程中定位原始数据。所有“BIN”对象均创建于原业务表所在的SCHEMA中。

recyclebin_retention_time

参数说明：设置回收站对象保留时间，超过该时间的回收站对象将被自动清理。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：900（即15min）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值900，表示recyclebin_retention_time为900s；带单位取值15min，表示recyclebin_retention_time为15min。取值如果要带单位，必须为s、min、h、d。

设置建议：根据用户需求对回收站对象保留时间进行设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

undo_retention_time

参数说明：设置undo旧版本保留时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 259200。0表示不保留闪回点快照信息。

默认值：0

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值900，表示undo_retention_time为900s；带单位取值15min，表示undo_retention_time为15min。取值如果要带单位，必须为s、min、h、d。

设置建议：推荐使用默认值。

设置不当的风险与影响：取值若超出范围，或者取值带有除s、min、h、d以外的单位时，会有报错提示。

须知

- 在进行Ustore闪回查询时，如果中途设置该参数为0，则会清理闪回点快照信息，之前的任何版本不允许再做闪回查询。执行闪回查询会报错："cannot find the restore point"。
- 如果想要保留的undo记录旧版本时间为time1，闪回查询执行的SQL时间为time2，需要设置参数undo_retention_time大于两者之和。即设置undo_retention_time > time1 + time2 + 3s。建议设置 undo_retention_time = time1 + 1.5 * time2。例如：想要保留3h的旧版本，闪回查询执行时间为1h，则undo_retention_time = 3h + 1.5 * 1h = 4.5h。

6.3.41 回滚相关参数

max_undo_workers

参数说明：异步回滚调用的undoworker线程数量，参数重启生效。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：无

取值范围：1 ~ 100

默认值：5

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：只影响同时并发进行异步回滚的最大线程数，设置过小可能会导致异步回滚速度慢，在系统中存在大量需要异步回滚的事务时，极低概率会影响undo 回收的速度。当机器配置不高，系统中又存在大量需要异步回滚的事务时，该值设置过大会导致同时拉起的异步回滚线程数多，占用一定的系统资源。

6.3.42 AI 特性

enable_hypo_index

参数说明：该参数控制数据库的优化器进行EXPLAIN时是否创建虚拟索引。通过对特定的查询语句执行explain，用户可根据优化器给出的执行计划评估该索引是否能够提升该查询语句的执行效率。该参数可在PDB级别设置。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示在执行EXPLAIN命令时创建虚拟索引。
- off：表示在执行EXPLAIN命令时不创建虚拟索引。

默认值：off。在PDB场景内，若未设置该参数，则继承来自全局的设置。

设置方式：该参数属于USERSET类型参数，请参见**表6-2**中对应设置方法进行设置。

设置建议：根据业务需要进行调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_ai_stats

参数说明：该参数用于指定是否创建或者使用智能统计信息。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示创建并且使用智能统计信息。
- off：表示不创建或者不适用智能统计信息。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：数据库运行过程中修改可能会造成计划劣化。

multi_stats_type

参数说明：该参数用于指定在**enable_ai_stats**为on状态下创建的统计信息类别。

参数类型：枚举类型

参数单位：无

取值范围：有效值为“BAYESNET”、“MCV”、“ALL”。

- "BAYESNET"：只创建智能统计信息。
- "MCV"：只创建传统统计信息。
- "ALL"：同时创建传统统计信息和智能统计信息。

默认值："BAYESNET"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_stats_cache_limit

参数说明：该参数用于指定在**enable_ai_stats**为on状态下最多缓存的模型数量。

参数类型：整型

参数单位：个

取值范围：30 ~ 1000

默认值：100

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议根据使用多列统计信息模型数量设置。

设置不当的风险与影响：最大缓存个数设置过大会占用不必要的内存，参数设置过小会频繁从硬盘读取模型，造成系统性能下降。

enable_operator_prefer

参数说明：该参数用于指定是否开启算子倾向性规则，在估计代价相近的情况下，倾向于选择参数化路径执行表连接。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启参数化路径优先。
- off：表示不开启参数化路径优先。

默认值：off

版权所有 © 华为云计算技术有限公司

说明

此参数生效有两个必要的前置条件：

- 参数化路径被生成。
- 参数化路估计的代价和其他索引扫描算子类似。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为默认值，推荐只在session或者hint内指定该参数设置。

设置不当的风险与影响：修改此参数会有严重的计划劣化风险。

enable_adaptive_cost

参数说明：该参数为基于反馈的优化器基数和代价矫正功能的总开关，打开后开启算子信息收集流程和基数估计流程，后端模型维护的启动线程会随此参数开关启动或退出。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启算子信息流程和基数估计流程。
- off：表示关闭算子信息流程和基数估计流程。

默认值：

- on：新安装的数据库的默认值。
- off：506.0.0版本之前的数据库升级后的默认值。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。请根据查询语句是否需要自动调优开启/关闭此参数。

设置不当的风险与影响：不需要该功能的场景下，比如业务中不存在需要调优的复杂语句时，打开功能会占用更多系统资源。

enable_feedback_cardest

参数说明：该参数为另外设置的基于反馈的优化器基数和代价矫正功能的开关，用于开发人员诊断模型相关问题。当**enable_adaptive_cost**参数设置为off，该参数设置为on时，算子信息仍会被收集，反馈基数估计接口仍会被调用，但此时后端自动模型维护的线程不会被启用，开发人员可以通过gs_acm_analyze_workload_manual()函数手动训练模型诊断问题。

参数类型：布尔型

参数单位：无

取值范围：

- on：算子信息会被收集，且反馈基数估计接口仍会被调用，但后端自动模型维护的线程并不会被启用。
- off：控制完全交由**enable_adaptive_cost**参数控制。

默认值：

版权所有 © 华为云计算技术有限公司

- on：新安装的数据库的默认值。
- off：506.0.0版本之前的数据库升级后的默认值。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

adaptive_cardest_strategy

参数说明：该参数用于设置选择基数估计模型偏好。

参数类型：枚举类型

参数单位：无

取值范围："auto"、"use_statistics"、"use_feedback"

- "auto"：自适应模式，自动根据估计的历史准确性判断是使用统计方法还是反馈方法。
- "use_statistics"：优先使用统计信息做基数估计。
- "use_feedback"：优先使用反馈模型做基数估计。

默认值："auto"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：运行过程中修改该参数有计划劣化风险。

maximal_feedback_model_num

参数说明：该参数用于设置基数反馈模型数量上限，超过此上限后不再训练新的模型。

参数类型：整型

参数单位：个

取值范围：-1 ~ 1000000。-1表示无上限。

默认值：10000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果基数反馈模型数量上限设置过小，可能会因无法训练新的模型而导致反馈基数估计功能失效；如果数量上限设置过大，可能会占用过多的磁盘空间，造成性能下降。

feedback_model_cache_limit

参数说明：该参数用于设置控制基数反馈模型在全局内存最大缓存个数。

参数类型：整型

参数单位：个

版权所有 © 华为云计算技术有限公司

取值范围：10 ~ 100000

默认值：500

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：最大缓存个数设置过大会占用不必要的内存，参数设置过小会频繁从硬盘读取模型，造成系统性能下降。

feedback_model_expired_time

参数说明：该参数用于设置基数反馈模型（可通过gs_abo_model_statistic查看）超时时间，过期的模型会定期被清理。

参数类型：整型

参数单位：ms（毫秒）

取值范围：500 ~ 2147483647

默认值：86400000（即1d）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值6000，表示feedback_model_expired_time为6000ms；带单位取值1min，表示feedback_model_expired_time为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”，“h”，“d”。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果超时时间设置过长会造成系统性能下降；如果设置过短会导致模型被清理频率过高而影响功能。

feedback_collection_expired_time

参数说明：该参数用于设置未被训练的算子反馈信息的超时时间（算子反馈信息可通过gs_stat_get_acm_feedback_operator_info()查看），过期的算子信息会定期被清理。

参数类型：整型

参数单位：ms（毫秒）

取值范围：50 ~ 2147483647

默认值：3600000（即1h）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值6000，表示feedback_collection_expired_time为6000ms；带单位取值1min，表示feedback_collection_expired_time为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”，“h”，“d”。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果超时时间设置过长会造成系统性能下降；如果设置过短会导致反馈数据被清理频率过高而影响功能。

版权所有 © 华为云计算技术有限公司

adaptive_cost_min_time

参数说明：该参数用于设置基数反馈收集的SQL时间阈值，只有执行时间大于该值的语句反馈会被收集。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647

默认值：1000

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值6000，表示adaptive_cost_min_time为6000ms；带单位取值1min，表示adaptive_cost_min_time为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”，“h”，“d”。

设置建议：建议设置为需要调优的慢查询执行时长的20%。

设置不当的风险与影响：如果时间参数设置过短，最严重造成1-2%系统性能下降；如果设置过长则会导致能够被自动调优的查询不在功能生效范围内。

cost_update_window_size

参数说明：该参数用于调整收集回归数据的滑动窗口大小。

参数类型：整型

参数单位：个

取值范围：1 ~ 20

默认值：5

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：参数值小于3可能会导致估计代价参数过拟合与实际情况产生误差，使优化器生成错误的执行计划。

adaptive_costest_strategy

参数说明：该参数用于设置代价评估使用新/旧代价的策略。

参数类型：枚举类型

参数单位：无

取值范围：

- l0：只有在基数估计正确（比如使用反馈基数估计）的时候才会触发新代价。
- l1：优先使用新的代价模型计算。

默认值：l0

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为默认值，推荐只在session或者hint内修改该参数设置。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：参数设置为l1时有严重的计划劣化风险。

adaptive_costmodel_calibration_interval

参数说明：该参数用于设置代价模型矫正逻辑的间隔时间。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0 ~ 2147483647。0表示关闭自动触发代价模型矫正功能。

默认值：3600000（即1h）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值6000，表示adaptive_costmodel_calibration_interval为6000ms；带单位取值1min，表示adaptive_costmodel_calibration_interval为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”，“h”，“d”。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果该参数设置过小会导致代价模型经常触发矫正功能从而造成系统性能劣化；如果设置过大会导致模型矫正不及时从而使功能优化效果不佳。

adaptcost_extended_feature

参数说明：该参数用于设置开启自适应代价的扩展功能。

参数类型：字符串

参数单位：无

取值范围：

- none：不使用任何自适应代价扩展功能。
- support_or：支持含or和in条件的算子。
- support_left_join：支持含leftjoin/rightjoin条件的算子。
- support_stream_cost：支持stream算子的自适应代价估计。
- use_knn_support：优先使用knn模型支持in/or和leftjoin/rightjoin算子。

默认值："support_or,support_left_join,support_stream_cost,use_knn_support"

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：可能会导致部分算子无法使用自适应代价，无法优化计划。

enable_collect_abo_history

参数说明：该参数用于设置是否开启反馈基数/代价估计信息持久化。

参数类型：布尔型

参数单位：无

取值范围：

版权所有 © 华为云计算技术有限公司

- on：反馈基数/代价估计信息会被收集。
- off：反馈基数/代价估计信息不会被收集。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：不需要该功能的场景下，比如业务中不存在需要调优的复杂语句时，打开功能会占用更多系统资源。

feedback_record_expired_time

参数说明：该参数用于设置系统表gs_abo_history记录的超时时间，超过该时间时间的记录会定期被清理。

参数类型：整型

参数单位：ms（毫秒）

取值范围：500 ~ 2147483647

默认值：86400000（即1d）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，不带单位取值6000，表示feedback_record_expired_time为6000ms；带单位取值1min，表示feedback_record_expired_time为1min（60000ms）。取值如果要加上单位，必须为“ms”，“s”，“min”，“h”，“d”。

设置建议：推荐使用默认值。

设置不当的风险与影响：如果超时时间设置过长会造成系统性能下降；如果设置过短会导致记录被清理频率过高而影响功能。

enable_ai_watchdog

参数说明：开启或关闭AI Watchdog功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：功能打开会占用系统资源。

enable_ai_watchdog_forcible_oom_detection

参数说明：强制开启或关闭AI Watchdog的OOM探测功能，若关闭该参数，则会自动根据当前数据库的规格判断是否需要启动OOM探测功能。自动判断模式下，对于

版权所有 © 华为云计算技术有限公司

max_process_memory设置为64GB及以上的场景，才会启动OOM探测功能。由于OOM探测功能依赖内存管理模块获取的信息，因此，如果内存管理模块未开启或失效，则OOM探测功能也不会启用。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：功能打开会占用系统资源。

enable_ai_watchdog_healing

参数说明：开启或关闭AI Watchdog的自愈功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_watchdog_max_cpu_usage

参数说明：预期的数据库CPU使用率上限，该值会根据多核情况进行归一化。该参数值设置为0时，表示不判断CPU使用率情况。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0.8

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，可能会导致无法及时发现异常。

版权所有 © 华为云计算技术有限公司

ai_watchdog_oom_dynamic_used_threshold

参数说明：预期的数据库动态内存使用率上限。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0.95

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大，可能会导致无法及时发现异常。

ai_watchdog_oom_growth_confidence

参数说明：OOM检测算法置信度。

参数类型：浮点型

参数单位：无

取值范围：0.1 ~ 1

默认值：0.95

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_watchdog_oom_malloc_failures

参数说明：容忍的最大连续内存分配失败数量，超过该数量可能会触发OOM探测功能。

参数类型：整型

参数单位：个

取值范围：1 ~ 32000

默认值：50

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，会导致频繁报警；设置过大，会导致无法及时发现异常。

ai_watchdog_oom_other_used_memory_threshold

参数说明：预期的数据库的其他部分内存使用上限。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：MB

取值范围：-1 ~ 1048576

默认值：分为以下2种情况：

1. 如果该参数是通过OM安装进行设置，其默认值为-1。

   说明

   当参数值为-1时，对其他部分内存使用上限按照如下规则判断是否发生泄露：
   1. 若max_process_memory小于300GB，则其他部分内存使用上限为20GB；
   2. 若max_process_memory大于等于300GB小于600GB，则其他部分内存使用上限为40GB；
   3. 若max_process_memory大于等于600GB，则其他部分内存使用上限为60GB。

   若用户未采用默认值-1，则其他部分内存使用上限将采用用户设置的参数。
2. 如果该参数不通过OM安装进行设置，其默认值为20480。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，会导致频繁报警；设置过大，会导致无法及时发现异常。

ai_watchdog_oom_process_threshold

参数说明：预期的数据库进程使用占**max_process_memory**的使用比例，到达该阈值时，会触发内存泄漏判断。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 10

默认值：分为以下2种情况：

1. 如果该参数是通过OM安装进行设置，其默认值为1。
2. 如果该参数不通过OM安装进行设置，其默认值为1.1。

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_watchdog_oom_shared_threshold

参数说明：预期的数据库共享内存使用比例上限。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0.4

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_watchdog_rto_restriction_time

参数说明：AI Watchdog自愈功能的RTO限制，超过该RTO阈值，则不进行自愈操作。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 36000

默认值：600

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_watchdog_tolerance_times

参数说明：AI Watchdog启动自愈前最多能容忍多少次连续异常事件，通过该参数可以避免错误操作。

参数类型：整型

参数单位：次

取值范围：0 ~ 100

默认值：4

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ai_watchdog_tps_threshold

参数说明：数据库实例的预期TPS使用下限，低于该值，会触发异常判断逻辑。

参数类型：整型

参数单位：次

取值范围：0 ~ 32000

默认值：2

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

ai_watchdog_wait_time

参数说明：为了避免数据库频繁进行自愈操作，会在数据库启动后一段时间进行等待，该值即用来调整等待时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 36000

默认值：1800

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。取值不能带单位。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，可能会导致数据库刚启动时误报；设置过大，可能会导致无法发现等待时间内的异常。

ai_watchdog_warning_retention

参数说明：AI Watchdog在dbe_perf.ai_watchdog_detection_warnings视图中保留的告警记录数上限。

参数类型：整型

参数单位：个

取值范围：0 ~ 32000

默认值：20

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.43 Global SysCache 参数

enable_global_syscache

参数说明：控制是否使用全局系统缓存功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启全局系统缓存功能。
- off：表示不开启全局系统缓存功能。

默认值：on

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值，开启全局系统缓存功能可以降低系统缓存内存占用，提高并发扩展能力。若打开该参数，需要结合使用**global_syscache_threshold**，此时推荐使用线程池模式。若打开该参数后需要访问备机，建议设置备机**wal_level**级别为hot_standby以上。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

global_syscache_threshold

参数说明：全局系统缓存占用内存的最大值，使用时需要打开**enable_global_syscache**参数。

参数类型：整型

参数单位：kB

取值范围：16384 ~ 1073741824

默认值：163840（即160MB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为163840，不带单位，表示163840kB；设置为160MB，带单位，表示160MB。取值如果要带单位，必须为kB、MB、GB。

设置建议：建议设置为热点DB个数和线程个数的最小值乘以每个DB分配的内存大小，即global_syscache_threshold = min(count(hot dbs)，count(threads)) * memofdb。

热点DB数即访问较为频繁的数据库数量。线程数在线程池模式下取线程池线程个数和后台线程个数之和，非线程池模式不需要计算这个值，直接使用热点DB数。

memofdb即平均每个DB应该分配的内存，每个DB的底噪内存是2M，平均每增加一个表或者索引，增加11k内存。

设置不当的风险与影响：如果设置的值过小，会导致内存频繁淘汰，内存存在大量碎片无法回收，导致内存控制失效。

6.3.44 分布式备机读参数

enable_standby_read

参数说明：控制session是否开启备机读功能。

须知

仅在GTM-Free、GTM-Lite模式、非灾备集群、非事务块内、hot-standby模式下支持开启。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示该session开启分布式备机读功能。
- off：表示该session不开启分布式备机读功能。

版权所有 © 华为云计算技术有限公司

默认值：off

设置方式：该参数属于USERSET类型参数，只能在session级别设置（set enable_standby_read = on），不支持gs_guc set/reload方式进行设置。

设置建议：无法通过gs_guc进行设置，如需要进行备机查询，请在session会话中开启。

设置不当的风险与影响：若在非session级别设置（如在配置文件中将该参数打开），在参数冲突时会导致集群无法拉起；在参数不冲突时，会导致后台线程如autovacuum、WorkloadMonitor等也开启分布式备机读模式，其DDL、DML等作业会受到影响而报错。

enable_direct_standby_datanodes

参数说明：控制session是否开启execute direct on扩展能力。该参数开启后，execute direct on语句可以连接到任意一个备DN或者一个分片所有备DN上执行查询，execute direct on语法具体详见**EXECUTE DIRECT**。

须知

- 仅在gtm-free模式、非事务块内、hot-standby模式下支持开启。
- 同一个session上，若set enable_standby_read = on 之后设置 set enable_direct_standby_datanodes = on则报错。反之，若set enable_direct_standby_datanodes = on 之后设置 set enable_standby_read= on也报错。
- 该参数打开之后当前session为只读状态，其DDL、DML等作业会受到影响而报错。
- 该参数打开之后执行execute direct on仅支持查询系统视图、系统函数或者系统表等系统DFX信息，不支持查看普通用户表。
- enable_direct_standby_datanodes参数打开后execute direct on支持分布式备机查询，在升级过程使用该功能时必须保证CN和DN实例进程都已替换到最新的二进制，否则会出现报文校验失败执行报错。
- 该参数打开之后查询dbe_perf.statement、dbe_perf.summary_statement等系统视图数据，均为单独的CN上的数据，不包含主DN数据，不建议客户开启该参数后进行数据查询。
- 如果需要查询DN上数据可通过execute direct on datanodes进行查询。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示该session开启execute direct on扩展能力。
- off：表示该session不开启execute direct on扩展能力。

默认值：off

设置方式：该参数属于USERSET类型参数，只能在session级别设置（set enable_direct_standby_datanodes = on），不支持gs_guc set/reload方式进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：该参数用来控制在分布式备机读场景下execute direct on执行连接行为。主要用于WDR、以及数据库测试和运维使用，其他场景不建议打开使用。该参数无法通过gs_guc进行设置，如需要进行备机查询，请在session会话中开启。

设置不当的风险与影响：若配置文件中设置打开该参数，将无法启动此数据库节点。

standby_read_delay

参数说明：控制备机读时主备的最大差异，超过该参数不支持备机读。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 2147483647

- -1：表示备机读时不需要做主备差异的校验。
- 0：表示主备没有差异的时候才能进行备机读。

默认值：10000000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为86400，不带单位，表示86400ms；设置为1d，带单位，表示1d。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：根据实际业务需求合理设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

standby_read_rto

参数说明：控制备机读时系统最大RTO，超过该参数不支持备机读。

参数类型：整型

参数单位：ms（毫秒）

取值范围：-1 ~ 2147483647

- -1：表示备机读时不需要做系统RTO的校验。
- 0：表示至少有一台备机的RTO为0，才能进行备机读。

默认值：60000

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为86400，不带单位，表示86400ms；设置为1d，带单位，表示1d。取值如果要带单位，必须为ms、s、min、h、d。

设置建议：根据实际业务需求合理设置。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.45 备机数据修复

standby_page_repair

参数说明：控制备机回放时，回放目标页发现故障后是否进行页面自动修复。当前版本支持CRC校验失败、页面未初始化、页面LSN校验错误共三种类型的修复。不支持

版权所有 © 华为云计算技术有限公司

VM、Xlog、clog、MOT和UNDO元信息的坏块修复，以及FSM的LSN错误修复；UNDO记录仅支持CRC校验失败时的修复；临时表、全局临时表以及unlogged表不涉及主备修复。页面文件级丢失无法修复。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示备机回放时会自动检测修复页面。
- off：表示备机回放时不会自动检测修复页面。

默认值：on

须知

- 如果备机异常导致存在大量坏块，会对备机回放性能产生影响，可能造成备机Xlog堆积。且每次回放到坏块上的Xlog时都会打印异常日志。
- 为保证主备一致性，备机实例如果存在未修复的坏块需要阻止检查点推进，可能造成Xlog日志堆积；坏块上的回放将被暂停，可能引起主备不一致。备机重启或者升主时有坏块将导致实例异常退出，退出前会搜集坏块上的Xlog链，可能引起RTO时间延长。

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：不开启自动修复的情况下一旦发现坏块，坏块会滞留在坏块表中，并造成”须知”中的不良影响。

6.3.46 账本数据库参数

enable_ledger

参数说明：控制是否启用账本数据库功能。若此参数设置为打开（on）状态，则启用账本数据库功能，允许创建新的防篡改模式和修改普通模式为防篡改模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用账本数据库功能。
- off：表示不启用账本数据库功能。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：该参数默认关闭，需要开启账本数据库功能时，将enable_ledger设置为on。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

ledger_hist_level

参数说明：全局区块表记录控制开关，用于控制全局区块表中是否记录SQL语句。

参数类型：整型

参数单位：无

取值范围：

- 0：表示在全局区块表中不记录SQL语句。
- 1：表示在全局区块表中记录SQL语句。

默认值：1

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.47 在线创建索引

delete_cctmp_table

参数说明：控制是否删除Ustore在线创建/重建索引过程中产生的临时表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示删除临时表。
- off：表示保留临时表。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数默认开启，需要保留在线创建/重建索引过程中产生的临时表时设置为off。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_bypass_insert_sequence

参数说明：控制执行含自增列表的INSERT语句时，是否支持SQLBYPASS优化。

说明

- 当GUC参数enable_opfusion开启时，该参数设置才会生效。
- 自增列的类型限制为：smallint、int、bigint。
- 继承SQLBYPASS插入语句的约束。
- 只支持INSERT场景下的序列自增，不支持UPDATE等语句。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示支持SQLBYPASS优化。
- off：表示不支持SQLBYPASS优化。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**18.2-表 GUC参数分类**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：若关闭该参数，含自增列表的INSERT语句将无法使用SQLBYPASS优化，从而会降低INSERT语句的执行效率。

6.3.48 在线 DDL

enable_online_ddl

参数说明：该参数与ONLINE/OFFLINE关键字（关键字详情请参考**ALTER TABLE**，**VACUUM**）结合使用，控制是否开启在线DDL或在线VACUUM。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用在线DDL，当前版本不生效。
- off：表示不启用在线DDL，当前版本不生效。

默认值：off

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议关闭此参数，此参数当前版本不生效。需要使用在线DDL新增特性可在语法里加ONLINE关键字。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.49 数据生命周期管理-OLTP 表压缩

enable_ilm

参数说明：控制是否开启数据生命周期管理-OLTP表压缩特性。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：该参数默认off，如果需要开启数据生命周期管理-OLTP表压缩特性，请联系华为工程师购买license后再设置为on。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ilm_instance_compress

参数说明：将指定ILM压缩策略设定为实例级压缩策略。当前仅支持将跨页透明压缩（TURBO）策略指定为实例级策略，当ilm_instance_compress=turbo时，实例下所有支持跨页透明压缩的对象均会自动添加TURBO压缩策略。

参数类型：枚举类型

参数单位：无

取值范围：

- off：表示关闭。
- turbo：表示开启，指定实例级压缩策略类型为turbo。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数默认关闭，需要开启实例级压缩功能时，可设置为turbo。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ilm_instance_compress_configuration

参数说明：指定实例级压缩策略的默认配置，由压缩级别和压缩冷热管理时间两部分组成。以默认值为例，其代表对压缩对象中1天内未修改的数据使用low级别进行压缩。时间范围不能超过3年。

参数类型：字符串

参数单位：无

取值范围：

- 压缩级别：low、medium、high
- 冷热管理时间：整型数
- 冷热管理时间单位：hour、day、month、year

默认值：low,1,day

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数默认low,1,day。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

turbo_compress_shared_buffers

参数说明：跨页透明压缩专用元信息缓冲区大小。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：8kB（页面）

取值范围：0 ~ 1073741823

默认值：0

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。例如：不带单位取值1024，表示turbo_compress_shared_buffers为1024*8kB，即8192kB；带单位取值8MB，表示turbo_compress_shared_buffers为8MB。若取值需带单位，必须为kB、MB或GB，且取值必须为8kB的整数倍。

设置建议：该参数默认取值0，表示禁用跨页透明压缩功能，当使用此功能时，建议设置为shared_buffers大小的1/128。已经存在透明压缩表后，不能将该参数再设置为0。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

turbo_compress_default_configuration

参数说明：跨页透明压缩默认配置(';'隔离不同级别，依次为low、medium、high；','隔离不同级别下的压缩配置，依次为编码器、压缩器和压缩单元大小)。

参数类型：字符串

参数单位：无

取值范围：

- 编码器：tiling、smartdiff、pattern
- 压缩器：zstd、lz4
- 压缩单元大小：1、2、4、8、16、32、64

默认值：tiling,zstd,4;tiling,zstd,16;pattern,zstd,16

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数默认tiling,zstd,4;tiling,zstd,16;pattern,zstd,16。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

turbo_compress_pattern_encoder

参数说明：跨页透明压缩的模式编码器，当turbo_compress_default_configuration中的编码器配置为pattern时，可通过此参数配置具体使用的模式编码算法，支持指定多种模式编码算法的组合，通过竖线“|”连接。

参数类型：字符串

参数单位：无

取值范围：

- none：表示无编码
- transpose：表示行列转置。
- transpose|column_encode：表示行列转置加列编码。
- transpose|dedup|column_encode：表示行列转置加列编码加去重。

版权所有 © 华为云计算技术有限公司

默认值：transpose

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：该参数默认transpose。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

须知

- 模式编码是否有正收益依赖于数据特性，当评估不产生正收益时，将自动关闭模式编码。
- 模式编码需要额外申请内存，当内存资源不足时，将自动关闭模式编码。

turbo_compress_enable_double_write

参数说明：是否开启跨页透明压缩双写。关闭之后在异常宕机场景下可能导致压缩数据损坏。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

说明

- 完整的跨页透明压缩双写需要记录元信息和实际数据两部分。
- 当参数turbo_compress_enable_double_write与enable_double_write都为on时，两部分都会记录双写；当两个参数都是off时，两个部分都不会记录双写；当仅有一个参数开启时，会仅记录元信息的双写。
- 当完全不记录双写时，出现元信息损坏则无法进行恢复；当仅记录元信息双写时，出现的数据损坏可以尝试通过HA进行恢复。

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：该参数默认on。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

turbo_compress_preload_workers_num

参数说明：并行回放/极致RTO下压缩页面预读线程个数。

参数类型：整型

参数单位：个

取值范围：0 ~ 16

默认值：0

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于POSTMASTER类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：该参数默认0。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

turbo_compress_force_fast_allocate

参数说明：压缩表扩展时是否使用快速预占进行空间分配，默认使用填0的方式进行分配。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见表6-1中对应设置方法进行设置。

设置建议：在ext4等支持快速预占的文件系统上可设置为on，来提高压缩表的扩展速度。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.50 session 级事务下推及相关参数

session_type

参数说明：指定当前会话的类型，大小写不敏感，根据用户实际取值生成执行计划。

参数类型：枚举类型

参数单位：无

取值范围：

- single_node：单节点会话，读写操作支持下推，根据GUC参数exec_target_node判断下推节点并生成执行计划。
- global_read_only：跨节点只读会话，正常生成分布式计划，但禁止分布式写操作。
- normal：默认会话，和原有行为保持一致，根据数据库原有逻辑执行计划。

默认值：normal

设置方式：该参数属于USERSET类型参数，请参见表6-2中对应设置方法进行设置。注意该参数不支持使用gs_guc全局设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：在单节点会话模式下（即session_type设置为single_node）执行数据插入时，数据分布的正确性由应用层保证，如果指定了错误的节点可能会发生数据不一致问题。

版权所有 © 华为云计算技术有限公司

exec_target_node

参数说明：指定当前会话的DML以及部分DDL（TRUNCATE、ALTER TABLE...TRUNCATE PARTITION...）语句执行的DN节点名，大小写敏感。如果DN节点名不合法，则在设置时报错。

说明

仅GUC参数**session_type**设置为single_node时生效。

参数类型：字符串

参数单位：无

取值范围：当前集群内的所有DN节点名，可以通过查询PGXC_NODE系统表获取所有DN节点名。

默认值：""

设置方式：该参数属于USERSET类型参数，请参见**表6-2**中对应设置方法进行设置。注意该参数不支持使用gs_guc全局设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：单节点会话模式下，若没有按照约束使用，可能导致数据不一致问题。

说明

单节点会话模式具有如下约束：

- router功能在此场景下不生效。
- node_name hint指定的DN不生效，按exec_target_node下发。
- 不支持涉及行级访问控制的语句。
- 不支持分布式备机读。
- 不支持高级包语句。
- 不支持GPC。
- 在单节点会话模式下，使用PBE语句时，会导致计划类型直接选择GPlan。而如果已经生成计划后再指定单节点会话模式，则保持原有计划类型不变。

示例

```
gaussdb=# SET log_min_messages = debug2;
gaussdb=# SET logging_module = 'on(opt_choice)';
gaussdb=# CREATE TABLE t1(a int ,b int);
gaussdb=# PREPARE x(int) AS SELECT * FROM t1;
gaussdb=# EXPLAIN EXECUTE x(10);
NOTICE:  [Choosing C/G/A Plan]: GPlan, reason: Using pbe optimization.
NOTICE:  [Choosing C/G/A Plan]: CPlan, reason: First 5 times using CPlan.
                  QUERY PLAN
--------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: All datanodes
(2 rows)
gaussdb=# DEALLOCATE x;

gaussdb=# PREPARE x(int) AS SELECT * FROM t1;
gaussdb=# SET session_type=single_node;
gaussdb=# SET exec_target_node=datanode1;
gaussdb=# EXPLAIN EXECUTE x(10);
NOTICE:  [Choosing C/G/A Plan]: GPlan, reason: Using pbe optimization.
```


版权所有 © 华为云计算技术有限公司

```
NOTICE:  [Choosing C/G/A Plan]: GPlan, reason: Using pbe optimization.
                QUERY PLAN
---------------------------------------------------
 Data Node Scan  (cost=0.00..0.00 rows=0 width=0)
   Node/s: datanode1
(2 rows)
gaussdb=# DEALLOCATE x;
gaussdb=# DROP TABLE t1;
```

6.3.51 Enhanced Toast

enable_enhance_toast_table

参数说明：是否使用Enhanced Toast线外存储表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数取值默认为on，需要关闭Enhanced Toast线外存储功能时设置为off。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.52 全局临时表

max_active_global_temporary_table

参数说明：全局临时表功能开关，控制是否可以创建全局临时表。该参数的取值决定了共享缓存中预留给全局临时表所需的哈希表的内存使用，并不会强制限制所有会话中的活跃全局临时表总数。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000000

- 0：全局临时表功能关闭。
- > 0：全局临时表功能打开。

默认值：1000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

vacuum_gtt_defer_check_age

参数说明：VACUUM执行后检查当前数据库所有全局临时表的relfrozenxid最小值与所有普通表的relfrozenxid最小值的差异。如果全局临时表的relfrozenxid最小值落后超过指定参数值，就会产生告警“global temp table oldest relfrozenxid %lu is the oldest in the entire db”，提醒用户手动执行VACUUM，否则可能导致clog日志不回收。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000000

默认值：10000。

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为默认值，不做调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_gtt_concurrent_truncate

参数说明：是否支持全局临时表truncate table和DML的并发执行，以及全局临时表truncate table和truncate table的并发执行。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示支持上述操作并发。
- off：表示不支持上述操作并发。

默认值：on

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：建议设置为默认值，不做调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

global_gtt_cn_session_key

参数说明：表示当前会话的会话ID，即分布式全局临时表的全局唯一性会话标识。

参数类型：字符串

参数单位：无

取值范围：字符串

默认值：无

设置方式：该参数为内部参数，不支持用户设置，用户查询返回结果为空。

设置建议：不支持设置。

设置不当的风险与影响：无

版权所有 © 华为云计算技术有限公司

gtt_reuse_wait_timeout

参数说明：分布式全局临时表REUSE操作中，当前会话复用GTT session mapping entry时，等待上一个会话释放相关资源的最大超时时间。

参数类型：整型

参数单位：ms（毫秒）

取值范围：0~1200000

默认值：5000（即5s）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在网络延迟较高或业务中存在长事务频繁使用全局临时表的语句报错场景，可适当增大该值，避免因等待超时导致REUSE失败。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。设置过小可能导致在正常清理周期内REUSE超时报错，影响业务连续性；设置过大可能导致DN线程长时间阻塞在REUSE等待中，影响该线程对其他请求的响应。

6.3.53 单节点表

default_nodegroup

参数说明：创建Schema时，是否开启默认设置单节点属性的功能。

参数类型：字符串

参数单位：无

取值范围：

- NONE：创建Schema时，默认关闭设置单节点属性的功能。
- SCHEMA：创建Schema时，默认开启设置单节点属性的功能。

默认值：NONE

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改

说明

当default_nodegroup的值设置为SCHEMA时，只允许普通用户默认设置单节点属性，不支持管理员、运维、初始用户等角色默认设置单节点属性。

single_node_table_strategy

参数说明：控制创建单节点表的行为。

参数类型：字符串

参数单位：无

取值范围：FORBID、ALLOW_STRICT、ALLOW_DEFAULT_NON_PART、ALLOW_DEFAULT_ALL

版权所有 © 华为云计算技术有限公司

- FORBID：关闭创建单节点表功能，包括其他表转换为单节点表的能力。
- ALLOW_STRICT：打开显式创建单节点表功能，包括分区表和非分区表。
- ALLOW_DEFAULT_NON_PART：在支持显式创建单节点表功能基础上，打开非分区表在未指定分布策略以及未指定TO GROUP子句时，根据Schema的单节点属性，默认创建为新的单节点表的功能。
- ALLOW_DEFAULT_ALL：在支持显式创建单节点表功能基础上，打开一级分区表和非分区表，在未指定分布策略以及未指定TO GROUP子句时，根据Schema的单节点属性，默认创建为单节点表的功能。

默认值：ALLOW_STRICT

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

说明

单节点表支持备份恢复。如果原库打开GUC参数single_node_table_strategy且存在单节点表，而目标库关闭GUC参数single_node_table_strategy时，当进行备份恢复操作会存在以下情况：

- 对于库级备份恢复：可正常恢复，表带有单节点属性。
- 对于表级备份恢复：产生报错，需要修改参数（打开GUC参数single_node_table_strategy）才能进行恢复。

6.3.54 HTAP 混合负载

本章节介绍HTAP（Hybrid Transactional/Analytical Processing，混合事务/分析处理）特性相关参数。

enable_htap

参数说明：是否开启HTAP特性，开启HTAP特性后，会加载COLVIEW关键字指定列的存量行存数据到内存IMCV（In-memory Column View，列式内存引擎）中，用于提高AP（Analytical Processing，分析处理）查询效率。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开HTAP特性。
- off：关闭HTAP特性。

默认值：off

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：该参数默认关闭，如果需要开启，请结合数据库使用场景，存在大量AP查询时，开启该参数可以优化查询效率。纯TP查询场景不建议开启该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

htap_max_mem_size

参数说明：依赖HTAP特性开启状态，设置HTAP特性可用的内存上限。当GaussDB动态内存使用达到上限后，即使HTAP内存使用未达到上限，也无法继续分配内存。

参数类型：整型

参数单位：kB

取值范围：102400 ~ 1073741824

默认值：1048576（即1GB）

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。例如，设置为1000000，不带单位，表示1000000kB；设置为1GB，带单位，表示1GB。取值如果要带单位，必须为kB、MB、GB。在不重启的情况下，可增大该参数值。如果改小该参数值，不报错，但仅重启后生效，即重启前为修改前的值，重启后为修改后的值。

设置建议：该参数不应该超过**max_process_memory**和最大共享内存，并根据实际使用场景合理分配空闲内存作为该参数的值。htap_memctl_policy=swap或者escape时，在加载和重建过程中可能看到htap_used_memory值略大于htap_max_mem_size，加载或重建结束后，htap_used_memory会自动回落到htap_max_mem_size以下。

手动加载/重建时内存计数极端情况下的溢出量为：同时加载表数量 * **htap_rowgroup_pagenum** * 8KB * 并行度（参考**gs_imcv_flush**）。

业务运行时极端情况下自动重建导致的溢出量为：**htap_flush_workers**（后台线程数）* **htap_rowgroup_pagenum** * 8KB。

设置不当的风险与影响：htap_max_mem_size的设置基于GaussDB的动态内存和内存管理机制。当max_process_memory设置不当、htap_max_mem_size设置过大、关闭逻辑内存管理时，可能会导致系统内存不足。当htap_max_mem_size设置过小时，根据htap_memctl_policy的取值，可能会导致创建IMCV时频繁触发内存不足异常或者触发IMCV表的落盘。

说明

- 该参数的最大可用内存，指的是HTAP中列存形态的用户数据占用的内存，除此之外，还有IMCU的描述符等元数据的内存，不统计在该范围内。因此HTAP实际使用的内存会比这个值要大。
- 由于HTAP列存的行组是和行存一定数量的页面做映射的，即参数**htap_rowgroup_pagenum**，一个行组包含的IMCU个数又和表的列数对应，因此元数据底噪受到表的数据量大小、列数和行存页面映射数的影响。在页面映射数为8192，单表列数为100，加载到列存的数据量为1TB的大数据量场景，HTAP元数据底噪在1.5GB以内。
- 最大动态内存和已使用的动态内存可以通过gs_total_memory_detail视图查询获得，max_dynamic_memory表示最大动态内存，dynamic_used_memory表示已使用的动态内存。

htap_rowgroup_pagenum

参数说明：依赖HTAP特性开启状态，设置HTAP特性的行组和行存页面映射数的关系。

参数类型：整型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：1000、2048、4096、8192

默认值：1000

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐100GB列存数据以下的小数据量场景使用默认值，1TB以上的列存数据场景使用最大值8192。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

htap_eviction_threshold

参数说明：依赖HTAP特性开启状态，设置HTAP特性缓冲区的换入换出阈值。

参数类型：整型

参数单位：%

取值范围：0 ~ 100

- 0：表示没有任何IMCU会被加载到共享缓冲区。
- 100：表示缓冲区只会在内存使用量超出内存上限时执行淘汰，其余时间不执行任何淘汰或加载动作。
- 其他值：表示HTAP内存占用到**htap_max_mem_size** * **htap_eviction_threshold**/ 100时，缓冲区才会开始淘汰任务。

默认值：90

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，设置0或100两个特殊值时，需要充分理解参数含义，评估影响。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

htap_memctl_policy

参数说明：依赖HTAP特性开启状态，设置HTAP特性内存管理的策略模式。

参数类型：枚举类型

参数单位：无

取值范围：

- delete：废弃参数值，强制修改后，实际生效值为escape。
- swap：表示当HTAP内存不足时，会发生数据交换，使用的策略为Anti-Cache和懒加载，不常用数据会异步落盘到本地磁盘中。如果是DDL加载表到列存，内存不足时超内存部分会直接落盘。
- escape：表示当HTAP内存不足时，新增列存数据会仅存储元数据，查询时这部分数据会回行读取而不是以列存形态存储，和所有页面全为脏页的行组类似。

默认值：swap

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

该策略在以下两种内存不足场景下生效：

1. HTAP内存使用达到htap_max_mem_size (HTAP内存上限) 设置的上限；
2. 全局动态内存使用量达到动态内存管控上限。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_imcvscan

参数说明：依赖HTAP特性开启状态，设置是否打开HTAP内存列扫描算子IMCVScan。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开HTAP内存列扫描算子IMCVScan。
- off：关闭HTAP内存列扫描算子IMCVScan。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_hybrid_imcvscan

参数说明：依赖HTAP特性开启状态和enable_imcvscan参数取值，设置是否打开HTAP列回行混合扫描特性。

参数类型：布尔型

参数单位：无

取值范围：

- on：打开HTAP列回行混合扫描特性，只有当enable_imcvscan取值为on时才会真正开启该特性。
- off：关闭HTAP列回行混合扫描特性，不受enable_imcvscan的取值影响。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

imcvscan_consistency

参数说明：依赖HTAP特性开启状态和enable_imcvscan参数取值，设置列存扫描的一致性语义。

强一致语义下，无论列存数据是否存在未同步的行存变更，列存扫描始终保持和行存结果的一致。

版权所有 © 华为云计算技术有限公司

弱一致语义下，仅扫描列存数据，当列存数据存在未同步的行存变更时，不保证和行存结果的一致，不支持列回行、备机IMCVScan。

参数类型：枚举类型

参数单位：无

取值范围：

- strong：强一致性，保证和行存扫描结果的强一致。
- weak：弱一致性，仅扫描列存数据，当列存数据存在未同步的行存变更时，不保证和行存结果一致。

默认值：strong

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值，建议仅当列存扫描结果和行存数据没有强一致诉求的场景下使用弱一致性语义。弱一致性语义不支持列回行，当enable_hybrid_imcvscan=on时，将忽略imcvscan_consistency=weak设置，并使用强一致性语义。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。弱一致性语义下，注意不要将列存扫描的结果作为业务操作的依据，以免造成数据问题。

htap_router_mode

参数说明：依赖HTAP特性开启状态，设置HTAP透明路由的模式。

参数类型：枚举类型

参数单位：无

取值范围：

- row：强制使查询负载选择行存计划。
- column：当查询请求中的列加载到IMCV时，无视代价高低，强制执行列存（IMCVScan）计划；若存在未加载到IMCV的列时，执行行存计划。
- auto：根据查询请求中的列是否加载到IMCV，以及行列计划代价高低，自动选择行、列、行列混合计划。

默认值：row

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：当使用场景有混合负载时，推荐该参数设置为auto。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

htap_flush_workers

参数说明：依赖HTAP特性开启状态，设置HTAP最大后台工作线程数，后台线程用于执行同步增量行存数据到IMCV行组，异步加载，缓冲区换入换出任务。

参数类型：整型

参数单位：无

取值范围：1 ~ 256

默认值：3

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：在需要加载多个表的场景下，可以调高该参数，加载完成后，调低该参数至默认值。

设置不当的风险与影响：设置过大可能影响TP性能，设置过小可能影响AP性能。

htap_flush_pagenum

参数说明：依赖HTAP特性开启状态，设置同步增量行存数据到IMCV行组的脏页阈值，脏页数量大于该值时触发同步过程。

参数类型：整型

参数单位：无

取值范围：-1, 1 ~ 8192（其中-1取值表示阈值无穷大，不按照脏页数量做同步。）

默认值：512

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：一般不建议修改，降低该参数可以提高AP性能，但是可能会影响TP性能；增加该参数会影响AP性能。该参数的值和**htap_rowgroup_pagenum**关联，如果大于页面映射数，生效的值为**htap_rowgroup_pagenum**。

设置不当的风险与影响：设置过小可能影响TP性能，设置过大可能影响AP性能，建议使用默认值。

htap_flush_interval

参数说明：依赖HTAP特性开启状态，设置同步增量行存数据到IMCV行组的最大时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：-1, 1 ~ 36000（其中-1取值表示阈值无穷大，不按照时间间隔做同步。）

默认值：300

设置方式：该参数属于SIGHUP类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：一般不建议修改，降低该参数可以提高AP性能，但是可能会影响TP性能；增加该参数会影响AP性能。

设置不当的风险与影响：设置过小可能影响TP性能，设置过大可能影响AP性能，建议使用默认值。

enable_async_populate

参数说明：依赖HTAP特性开启状态，控制ALTER .. COLVIEW命令是否同步等待列存数据加载后返回。分布式下不支持同步等待列存数据加载，无论如何设置，实际生效值为on。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：ALTER ... COLVIEW命令创建元数据后直接返回，实际加载数据到IMCV的动作在后台执行。
- off：ALTER ... COLVIEW命令创建元数据后，等待后台线程加载数据到IMCV后返回。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

htap_deploy_mode

参数说明：依赖HTAP特性开启状态，设置HTAP部署模式。分布式下仅支持主机部署HTAP。分布式下无论如何设置，实际生效值为primary。

参数类型：枚举类型

参数单位：无

取值范围：

- primary：表示仅在主机上部署。只在主机上开启HTAP功能，备机不会回放IMCV DDL日志。当主备切换，主降备时会卸载IMCV，备升主时会重新加载IMCV。
- all：表示主备机全部部署，主备切换时保留主机和备机的IMCV数据。
- standby：表示仅在备机上部署。只在备机上开启HTAP功能，主上创建IMCV时只创建元数据，不加载数据。当主备切换，主降备时会加载IMCV，备升主时会卸载IMCV，只创建元数据。

默认值：primary

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：推荐使用默认值。在主备集群中，该参数需保持主备节点设置一致。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

htap_compression_methods

参数说明：依赖HTAP特性开启状态，设置IMCV列存压缩特性是否开启以及启用的压缩算法列表。

参数类型：枚举类型

参数单位：无

取值范围：

- none：表示不开启IMCV列存压缩特性。
- delta_plus_rle：表示开启IMCV列存压缩特性，并且启用delta_plus_rle算法。
- dict：表示开启IMCV列存压缩特性，并且启用dict算法。
- lz4：表示开启IMCV列存压缩特性，并且启用lz4算法。

版权所有 © 华为云计算技术有限公司

- all：表示开启IMCV列存压缩特性，并且同时启用delta_plus_rle、dict、lz4三种算法。

默认值：none

设置方式：该参数属于POSTMASTER类型参数，请参见**表6-1**中对应设置方法进行设置。

设置建议：内存充足时，推荐使用默认值。针对内存小、数据量大的环境，开启IMCV列存压缩特性，让尽可能多的数据在内存中。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.3.55 预留参数

说明

下列参数为预留参数，该版本不生效。

acce_min_datasize_per_thread

dfs_partition_directory_length

enable_fstream

enable_hdfs_predicate_pushdown

enable_orc_cache

schedule_splits_threshold

enable_hadoop_env

undo_zone_count

walwriter_cpu_bind

walwriteraux_bind_cpu

wal_rec_writer_bind_cpu

wal_receiver_bind_cpu

wal_sender_bind_cpu_attr

autoanalyze

cmserver_self_vote_timeout

transaction_deferrable

character_set_client

character_set_database

collation_database

character_set_server

collation_server

max_que_size

enable_global_plsqlcache

版权所有 © 华为云计算技术有限公司

max_execute_functions

max_compile_packages

net_write_timeout

plat_compat_allow_public_key_retrieval

plat_compat_server_port

delimiter_name

enable_parallel_populate

enable_startup_populate

catchup2normal_wait_time

废弃参数

enable_user_metric_persistent

user_metric_retention_time

enable_verify_active_statements

enable_dywlm_adjust

enable_force_memory_control

streaming_batch_memory

streaming_batch_size

streaming_batch_timeout

streaming_collect_memory

streaming_flush_interval

streaming_gather_window_interval

streaming_num_collectors

streaming_num_queues

streaming_num_workers

streaming_router_port

enable_slow_query_log

tde_cmk_id

transparent_encrypted_string

transparent_encrypt_kms_url

transparent_encrypt_kms_region

crypto_module_info

plpgsql_dependency

enable_gs_opt

 版权所有 © 华为云计算技术有限公司

prefetch_quantity

backwrite_quantity

fast_extend_file_size

enable_adio_debug

wal_flush_delay

query_log_file

query_log_directory

hll_default_regwidth

hll_default_expthresh

hll_default_sparseon

hll_max_sparse

enable_compress_hll

az_switchover_threshold

az_check_and_arbitrate_interval

az_connect_check_interval

az_connect_check_delay_time

enable_segment_remain_cleanup

enable_fast_allocate

wait_dummy_time

enable_incremental_catchup

enable_expr_fusion

enable_roach_standby_cluster

6.3.56 向量数据库参数

diskann_probe_ncandidates

参数说明：指定利用gsdiskann索引检索向量时的候选集合大小。

参数类型：整型

参数单位：无

取值范围：1 ~ 32768

默认值：128

设置方式：该参数属于USERSET类型参数，请参见《参考》中的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：

版权所有 © 华为云计算技术有限公司

- 推荐使用默认值，建议在使用中通过实验获得最优的参数配置。
- diskann_probe_ncandidates参数可以对于个别使用gsdiskann索引的查询进行设置和生效，建议只在会话中用set进行设置，不建议使用gs_guc进行全局设置。

设置不当的风险与影响：该参数设置过大会导致查询性能下降；设置过小会导致召回率不足。

diskann_hybrid_search_type

参数说明：指定向标混合场景下的查询策略。该参数可以对向标混合场景的查询进行设置和生效。

说明

本参数为会话级生效，建议使用《参考》中“数据库运行参数说明 > 设置参数”中方式二进行GUC设置，否则DN端无法选择指定策略。

参数类型：字符串

参数单位：无

取值范围：

- "AUTO"：默认参数，根据**diskann_hybrid_pre_filter_selectivity**和**diskann_hybrid_flat_upper_bound**的值选取策略。
- "PRE_FILTER_FLAT"：表示指定为先过滤后暴搜策略。
- "PRE_FILTER_ANN"：表示指定为先过滤后图搜策略。
- "POST_FILTER_ANN"：表示指定为先图搜后过滤策略。

默认值："AUTO"

设置方式：该参数属于USERSET类型参数，请参见《参考》中“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：推荐使用默认值，建议在使用中通过实验获得最优的参数配置。

设置不当的风险与影响：不同数据集场景下该参数设置不当可能会导致召回率不足或查询性能下降。

diskann_hybrid_pre_filter_selectivity

参数说明：指定向标混合场景下前过滤策略的选择率阈值。当选择率小于该阈值时执行前过滤查询，大于等于该阈值执行后过滤查询。

说明

- 当**diskann_hybrid_search_type**参数非默认值时，本参数不生效。
- 本参数为会话级生效，建议使用《参考》中“数据库运行参数说明 > 设置参数”中方式二进行guc设置，否则DN端无法选择指定策略。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0.15

版权所有 © 华为云计算技术有限公司

设置方式：该参数属于USERSET类型参数，请参见《参考》中“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：推荐使用默认值，建议在使用中通过实验获得最优的参数配置。

设置不当的风险与影响：该参数设置不当会导致召回率不足或查询性能下降。

diskann_hybrid_pullup_selectivity

参数说明：指定向标混合前过滤场景下，将部分前过滤条件上提到后过滤的选择率阈值。该参数设置越小，越容易将更多前过滤条件上提到后过滤。默认1.0取值下，前过滤条件不会上提到后过滤。

说明

- 上提后过滤条件时，会将最外层由AND连接的每个子条件视为一个整体，决定是否上提。例如A > 5 AND (B > 3 OR C > 3)中，A > 5是一个整体，(B > 3 OR C > 3)是一个整体。(B > 3 OR C > 3)要么统一放在前过滤，要么统一放在后过滤，不会拆开。
- 该参数的优先级低于**diskann_hybrid_search_type**参数和**diskann_hybrid_pre_filter_selectivity**参数。只有当**diskann_hybrid_search_type**设置为默认值，且过滤条件的整体选择率<**diskann_hybrid_pre_filter_selectivity**时，才有可能触发前过滤上提后过滤。
- 算法会按照选择率从高到低的顺序，尽可能将更多条件从前过滤上提至后过滤，直到不再满足后过滤选择率 > diskann_hybrid_pullup_selectivity且后过滤选择率 >= diskann_hybrid_pre_filter_selectivity的约束条件。
- 整体过滤条件选择率为0时，本参数不生效。
- diskann_hybrid_pullup_selectivity < diskann_hybrid_pre_filter_selectivity的行为等价于diskann_hybrid_pullup_selectivity = diskann_hybrid_pre_filter_selectivity的行为。
- 算法保证不会将所有前过滤条件上提为后过滤。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：1

设置方式：该参数属于USERSET类型参数，请参见《参考》中“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：前过滤爆搜/前过滤图搜策略中，根据过滤条件生成bitmap的开销和数据集大小*每个过滤条件的选择率累加和成正比。因此当数据集大小*选择率累加和过大，导致先过滤策略查询时延过高时，可以适当调小该参数，让部分先过滤条件上提到后过滤，减小bitmap生成开销。

设置不当的风险与影响：该参数设置会引起执行计划变更，设置不当可能导致查询没有走最优的执行计划。

diskann_hybrid_flat_upper_bound

参数说明：指定向标混合前过滤场景下暴搜策略最大数据量阈值。小于该数据量时执行暴力搜索，大于该数据量时执行向量检索。

版权所有 © 华为云计算技术有限公司

说明

- 当该参数设置为0时，arm64架构下实际生效值为8000，其他架构下实际生效值为11000。
- 当**diskann_hybrid_search_type**参数非默认值时，本参数不生效。
- 本参数为会话级生效，建议使用《参考》手册的“数据库运行参数说明 > 设置参数”中方式二进行guc设置，否则DN端无法选择指定策略。

参数类型：整型

参数单位：无

取值范围：0 ~ 1000000

默认值：0

设置方式：该参数属于USERSET类型参数，请参见《参考》手册的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：推荐使用默认值，建议在使用中通过实验获得最优的参数配置。

设置不当的风险与影响：该参数设置不当会导致召回率不足或查询性能下降。

diskann_hybrid_neighbor_search_prune_factor

参数说明：向标混合索引选择前过滤后ANN策略场景下，在搜索过程中，对不符合查询条件的邻居的裁剪比例因子。0为不裁剪，参数值越大，裁剪比例越高，1为全裁剪。

参数类型：浮点型

参数单位：无

取值范围：0 ~ 1

默认值：0。

设置方式：该参数属于USERSET类型参数，请参见《参考》手册的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：

- 推荐使用0.3，一般场景下，裁剪因子越大，可以提升查询性能，但会损失召回率，反之，召回率提升，损失查询性能，建议在使用中通过实验获得最优的参数配置。
- 本参数为会话级生效，建议使用《参考》手册的“数据库运行参数说明 > 设置参数”中方式二进行guc设置，否则DN端无法使用指定过滤。

设置不当的风险与影响：该参数设置过大会导致召回率不足。

diskann_hybrid_enable_lazy_flip_bitmap

参数说明：控制是否开启lazy flip bitmap特性。在前过滤爆搜或前过滤图搜策略下，对于一个选择率大于50%的标量查询或IS NOT NULL查询条件，如果其选择率大于50%，lazy flip bitmap就会计算它的补集，以此减少向bitmap插入数据的开销。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示开启。
- off：表示关闭。

默认值：on

设置方式：该参数属于USERSET类型参数，请参见《参考》手册的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gsivfflat_probes

参数说明：指定利用gsivfflat索引检索向量时待搜索的倒排列表的数量，超过gsivfflat索引的倒排列表总数时执行全表搜索。gsivfflat索引的倒排列表总数由创建索引时参数ivf_nlist指定。

参数类型：整型

参数单位：无

取值范围：1 ~ 32768

默认值：5

设置方式：该参数属于USERSET类型参数，请参见《参考》手册的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：推荐设置为索引创建时参数ivf_nlist取值的3%。建议在使用中通过实验获得最优的参数配置。

设置不当的风险与影响：该参数设置越大，搜索时间越长，但结果越准确。

gsivfflat_secondary_probes

参数说明：指定利用gsivfflat索引检索向量时待搜索的二级倒排列表的数量，超过二级倒排列表总数时执行全表搜索。二级倒排列表总数由创建索引时参数ivf_nlist2指定。

参数类型：整型

参数单位：无

取值范围：1 ~ 32768

默认值：5

设置方式：该参数属于USERSET类型参数，请参见《参考》手册的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：推荐设置为索引创建时参数ivf_nlist2取值的1/4~1/2之间。建议在使用中通过实验获得最优的参数配置。

设置不当的风险与影响：该参数设置越大，搜索时间越长，但结果越准确。

说明

gsivfflat_secondary_probes效果和gsivfflat_probes相同，只有向量索引为双层索引时生效，可以有效加速查询速度。建议在使用中通过实验获得最优的参数配置。

版权所有 © 华为云计算技术有限公司

enable_vectordb

参数说明：表示是否允许创建向量索引，是否允许对向量索引增加、修改和查询。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示允许。
- off：表示不允许。

默认值：off

设置方式：该参数属于SIGHUP类型参数，请参见《参考》手册的“数据库运行参数说明 > 设置参数”中对应设置方法进行设置。

设置建议：当不允许用户使用向量索引等向量数据库功能时，请设置为off，否则请设置为on。

设置不当的风险与影响：参数取值决定了用户能否使用向量数据库的功能，设置错误会导致用户使用数据库功能范围错误。

表 6-53 涉及向量数据库的 GUC 参数

GUC参数名	级别	取值范围（默认值）	描述
maintenance_work_mem	会话级（session）	[1024,2147483647]/[1MB,2048GB) (65536/64MB)	指定在维护性操作中使用的最大的内存量。默认单位是KB。
diskann_probe_ncandidates	会话级（session）	[1, 32768](128)	表示利用gsdiskann索引检索向量时的候选集合大小。
gsivfflat_probes	会话级（session）	[1, 32768](5)	表示待搜索的倒排列表的数量，超过倒排列表总数时执行全表搜索。
gsivfflat_secondary_probes	会话级（session）	[1, 32768](5)	表示待搜索的二级倒排列表的数量，超过二级倒排列表总数时执行全表搜索。
enable_vectordb	全局参数（SIGHUP）	[off,on](off)	高阶特性，表示是否允许创建向量索引，是否允许对向量索引增加、修改和查询。

6.4 CM 参数说明

CM相关参数的修改对GaussDB的运行机制有影响，建议由GaussDB的工程师协助修改。修改CM相关参数的方法，请参见表6-2中方式一或表6-3中对应方式进行设置。所有CM参数都可以通过表6-3方式进行设置，建议优先使用cm_ctl工具进行设置。

版权所有 © 华为云计算技术有限公司

cm_agent相关参数可通过cm_agent数据目录下的cm_agent.conf文件查看，cm_server相关参数可通过cm_server数据目录下的cm_server.conf文件查看。

6.4.1 cm_agent 参数

log_dir

参数说明：log_dir决定存放cm_agent日志文件的目录。可以是绝对路径，或者是相对路径（相对于cm_agent数据目录的路径）。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：“$GAUSSLOG/cm/cm_agent”，表示在$GAUSSLOG对应的cm目录下生成cm_agent日志。

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_file_size

参数说明：控制日志文件的大小。当日志文件达到指定大小时，则重新创建一个日志文件记录日志信息。

参数类型：整型

参数单位：MB

取值范围：0 ~ 2047

默认值：16

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。例如，设置为16，不带单位，表示16MB；设置为16MB，带单位，表示16MB。取值如果要带单位，必须为MB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_min_messages

参数说明：控制写到cm_agent日志文件中的消息级别。每个级别都包含排在它后面的所有级别中的信息。级别越低，服务器运行日志中记录的消息就越少。

参数类型：枚举类型

参数单位：无

取值范围：有效值有debug5、debug1、log、warning、error、fatal。

默认值：warning

版权所有 © 华为云计算技术有限公司

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：考虑硬盘大小后适当设置。

设置不当的风险与影响：设置级别低可能导致调试信息丢失。

log_threshold_check_interval

参数说明：控制日志压缩和清除的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：1800

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。如场景需要，可根据磁盘空间做适当调整。

设置不当的风险与影响：设置过大可能导致硬盘空间占用过多。

log_max_size

参数说明：控制日志最大存储容量。

参数类型：整型

参数单位：MB

取值范围：0 ~ 2147483647

默认值：10240

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。如场景需要，可根据磁盘空间做适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_max_count

参数说明：控制硬盘上可存储的最大日志数量。

参数类型：整型

参数单位：个

取值范围：0 ~ 2147483647

默认值：10000

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。如场景需要，可根据磁盘空间做适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

log_saved_days

参数说明：控制日志保存的天数。

参数类型：整型

参数单位：d（天）

取值范围：0 ~ 1000

默认值：90

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。如场景需要，可根据磁盘空间做适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_log_compress

参数说明：控制是否开启压缩日志功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启压缩日志功能。
- off：表示关闭压缩日志功能。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

incremental_build

参数说明：控制重建备DN模式是否为增量。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示增量重建备DN。
- off：表示全量重建备DN。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

alarm_component

参数说明：设置用于处理告警内容的告警组件的位置。

参数类型：字符串

参数单位：无

取值范围：合法文件路径。

- 若前置脚本gs_preinstall中的--alarm-type参数设置为5，表示未对接第三方组件，告警写入system_alarm日志，此时GUC参数alarm_component的取值为“/opt/huawei/snas/bin/snas_cm_cmd”。
- 若前置脚本gs_preinstall中的--alarm-type参数设置为1，表示对接第三方组件，此时GUC参数alarm_component的值为第三方组件的可执行程序的绝对路径。

默认值："/opt/huawei/snas/bin/snas_cm_cmd"

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

alarm_report_interval

参数说明：指定告警上报的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致告警上报不及时，过小可能导致cpu负载变高。

alarm_report_max_count

参数说明：指定告警上报的最大次数。

参数类型：整型

参数单位：无

取值范围：1 ~ 2592000

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致cpu负载变高。

版权所有 © 华为云计算技术有限公司

agent_report_interval

参数说明：cm_agent上报实例状态的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时。

agent_phony_dead_check_interval

参数说明：cm_agent检测CN/DN/GTM进程是否僵死的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时，设置过小会导致cpu负载变高。

agent_check_interval

参数说明：cm_agent查询DN、CN、GTM等实例状态的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：2

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时，设置过小会导致cpu负载变高。

agent_heartbeat_timeout

参数说明：cm_agent检测cm_server心跳的超时时间。当cm_agent没有在超时时间内接收到cm_server的心跳信息时，将终止与cm_server的连接。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：s（秒）

取值范围：2 ~ 2147483647

默认值：8

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

agent_connect_timeout

参数说明：cm_agent连接cm_server的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

agent_connect_retries

参数说明：cm_agent连接cm_server的重试次数。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：15

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

agent_conn_db_timeout

参数说明：实例状态检测连接超时时间。cm_agent检测CN、DN实例状态的长连接超时时间，超过该时间未获取连接，则认为连接失败。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：2

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

agent_conn_db_phony_dead_timeout

参数说明：实例僵死检测连接超时时间。cm_agent检测CN、DN是否处于僵死状态的短连接超时时间，超过该时间未获取连接，则认为连接失败。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

agent_status_check_connection_type

参数说明：cm_agent进行节点状态检测时的连接类型。

参数类型：整型

参数单位：无

取值范围：

- 0：默认走libpq通道，在高负载状态下，存在libpq通信失败的可能，这时使用UDS通信管道，通过系统预留的内存状态进行查询。
- 1：仅使用libpq通道，即与506.0及之前版本逻辑一致。
- 2：仅使用UDS通道。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

agent_kill_instance_timeout

参数说明：当cm_agent在无法连接cm_server主节点后，发起一次终止本节点上所有实例的操作之前，所需等待的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0，表示不发起终止本节点上所有实例的操作。

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：如果要设置为0以外的值，则需要大于cm_server升级为主节点所耗费的时间。

设置不当的风险与影响：设置过大可能导致CM故障恢复不及时。

agent_backup_open

参数说明：灾备集群设置，控制CM是否按照灾备集群模式运行。

参数类型：整型

参数单位：无

取值范围：0 ~ 3

- 0：表示未搭建灾备关系。
- 1：表示搭建了dorado灾备关系。
- 2：表示搭建了流式灾备关系。
- 3：表示搭建了逻辑灾备关系。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：此参数为内部调用参数，不建议用户修改。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

enable_gtm_phony_dead_check

参数说明：控制是否开启gtm僵死检查功能。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭gtm僵死检查功能。
- 1：表示开启gtm僵死检查功能。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

security_mode

参数说明：控制是否以安全模式启动CN、DN。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示以安全模式启动CN、DN。
- off：表示以非安全模式启动CN、DN。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

upgrade_from

参数说明：就地升级过程中使用，用于标识升级前集群的内部版本号，此参数禁止手动修改。

参数类型：整型

参数单位：无

取值范围：0 ~ 4294967295

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：手动修改可能会导致就地升级失败。

process_cpu_affinity

参数说明：控制是否以绑核优化模式启动主DN进程。仅支持ARM。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 取值为0时，表示不进行绑核优化。
- 取值大于0时，表示进行绑核优化，且物理CPU片数为2^n个。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群、cm_agent才能生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

dilatation_shard_count_for_disk_capacity_alarm

参数说明：扩容场景下，设置新增的扩容分片数，用于上报磁盘容量告警时的阈值计算。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：个

取值范围：0 ~ 2147483647

- 取值为0时，表示关闭磁盘扩容告警上报。
- 取值大于0时，表示开启磁盘扩容告警上报，且告警上报的阈值根据此参数设置的分片数量进行计算。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：请将该分片数与实际扩容分片数设置为一致。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_cn_auto_repair

参数说明：控制是否开启CN自动修复功能。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启CN自动修复功能，即CN被剔除后，agent会尝试自动修复并加回CN。
- off：表示关闭CN自动修复功能。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：CN自动修复功能默认开启，关闭后CN不会自动加回。若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_xc_maintenance_mode

参数说明：在集群只读模式下，控制是否可以修改pgxc_node系统表。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示可以修改pgxc_node系统表。
- off：表示不可以修改pgxc_node系统表。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

unix_socket_directory

参数说明：unix套接字的目录位置。

参数类型：字符串

参数单位：无

取值范围：合法目录路径。

默认值：""

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_dcf

参数说明：DCF模式开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启DCF模式。
- off：表示关闭DCF模式。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_e2e_rto

参数说明：端到端RTO功能开关，开启后僵死检测周期及网络检测超时时间将缩短，CM可以达到端到端RTO指标（单实例故障RTO<=10s，叠加故障RTO<=30s）。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭端到端RTO功能。
- 1：表示开启端到端RTO功能。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。如场景需要，可做适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

environment_threshold

参数说明：agent所监控的物理环境和节点状态信息的阈值，超过阈值会打印日志。具体分别表示为内存使用率阈值，cpu占用率阈值，磁盘使用率阈值，实例的内存使用率阈值，实例的线程池使用率阈值。

参数类型：字符串

参数单位：无

取值范围：该参数由五个部分组成，分别是内存使用率阈值、cpu占用率阈值、磁盘使用率阈值、实例的内存使用率阈值、实例的线程池使用率阈值，每个阈值的取值范围为[0,100]，单位为%，0表示关闭检测。不同阈值之间用“,”进行连接，取值要以括号开头和结束。例如，“(0,0,0,0,0)”表示对五个阈值都不进行检测。

默认值："(0,0,0,0,0)"

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

thread_effective_time

参数说明：线程运行一轮的有效超时时间。超时说明此线程卡住，线程将自动退出。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1200，0表示关闭线程监控能力。

默认值：60

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：不建议调整参数值大小，如果需要调整请联系华为技术支持。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

ping_fault_kill_inst_timeout

参数说明：主DN实例与所有备机ping不通后，重启该实例的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000，0表示关闭ping fault处理能力。

默认值：30

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

版权所有 © 华为云计算技术有限公司

high_cpu_threshold

参数说明：CPU高使用率的阈值，当CPU使用率超过阈值，cm_agent会根据检测周期将CPU使用信息打印到日志中，检测周期默认为1s/次。

参数类型：整型

参数单位：%

取值范围：0 ~ 100

默认值：95

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

high_mem_threshold

参数说明：内存高使用率的阈值，当内存使用率超过阈值，cm_agent会根据检测周期将内存使用信息打印到日志中，检测周期默认为1s/次。

参数类型：整型

参数单位：%

取值范围：0 ~ 100

默认值：95

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

high_disk_threshold

参数说明：磁盘高I/O的阈值，当I/O超过阈值，cm_agent会根据检测周期将I/O信息打印到日志中，检测周期默认为1s/次。连续打印10次，频率为1s/次，并将iostat -xm信息输出到system_stat日志中。

参数类型：整型

参数单位：%

取值范围：0 ~ 100

默认值：95

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

high_load_io_rw_times

参数说明：判断高负载的I/O读写次数。数据库I/O读写次数达到该阈值后，会认为处于数据库高负载状态，不进行磁盘hang、D状态等故障的处理。

参数类型：整型

参数单位：次/秒

取值范围：0 ~ 2147483647，0表示不进行I/O读写次数判断。

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

high_load_io_traffic

参数说明：判断高负载的I/O流量。数据库I/O流量达到该阈值后，会认为处于数据库高负载状态，不进行磁盘hang、D状态等故障的处理。

参数类型：整型

参数单位：kB/秒

取值范围：0 ~ 2147483647，0表示不进行I/O读写流量判断。

默认值：256

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

sys_check_interval

参数说明：系统资源检测周期，即CPU、内存、磁盘I/O检测的间隔时间。配置成0则关闭检测。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 10

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致告警不及时。

network_check_concurrency

参数说明：网络检测的并发数，即同一时刻检测多少条网络链路。配置成0则关闭检测。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：条

取值范围：0 ~ 1000

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致cm故障检测和恢复功能不及时，设置过大会导致cpu负载变高。

network_check_interval

参数说明：网络检测的周期，即间隔多久检测一次网络。配置成0则关闭检测。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时，设置过小会导致cpu负载变高。

asyn_log_enable

参数说明：异步日志开关，打开后cm_agent启动异步日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示cm_agent开启异步日志。
- off：表示cm_agent不开启异步日志，使用串行日志。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能导致日志丢失。

asyn_log_max_buf_size

参数说明：异步日志缓存大小。当异步日志发生堆积，且积压的日志超过了缓存大小，后续的日志无法记录。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：kB

取值范围：1 ~ 1048576

默认值：1024

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：根据内存和硬盘大小适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asyn_log_fault_solution

参数说明：识别到日志磁盘故障后，异步日志处理方案。当CM识别到日志盘故障或异步日志缓存溢出，CM会根据配置来处理异步日志。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：表示不对日志做多余的处理，缓存区保留所有日志，直到缓存大小达到上限，达到上限后不会记录更多日志。
- 1：表示对日志进行筛选，识别到日志磁盘故障后，只将关键日志、告警和ERROR级别以上的日志保存到缓存区，直到缓存大小达到上限，达到上限后不会记录更多日志。
- 2：表示丢弃日志，识别到日志磁盘故障后，只将关键日志、告警保存到缓存区，直到缓存大小达到上限，达到上限后不会记录更多日志。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：根据内存和硬盘大小适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

disk_check_hung_timeout

参数说明：磁盘I/O hang的超时时间，磁盘检测的单次I/O，如果超过这个时间，则认为磁盘已经I/O hang。配置成0则关闭磁盘hang检测。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

默认值：20，但如果安装时无法判断被检测磁盘是否为条带化，则会将默认值改为60。

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

版权所有 © 华为云计算技术有限公司

disk_check_block_timeout

参数说明：如果磁盘因文件系统某些原因，导致open这类文件操作阻塞，当这类操作阻塞超过此参数设定的时间时，会判定磁盘故障。配置为0，表示关闭此判断。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

默认值：40

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

disk_check_interval

参数说明：控制磁盘I/O hang检测的流量，即最少需要在多长时间内写入指定的数据。默认检测周期是5秒，配置成0则关闭磁盘检测。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能导致故障检测不准确、故障恢复不及时。

disk_check_io_size

参数说明：控制磁盘I/O hang检测的流量，即在检测周期内，给磁盘中写入的数据量，这个是直接对磁盘的I/O操作。默认是写入512B的数据，如果检测的磁盘为条带化，则安装的时候会把默认参数改为262144B（256KB）。最大写入1G的数据。配置成0则关闭磁盘hang检测。磁盘检测白名单包括：CM，CN，DN，GTM数据目录，日志目录所在磁盘以及自定义表空间目录所在磁盘。

参数类型：整型

参数单位：B

取值范围：0 ~ 1073741824

默认值：512

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能导致故障检测不准确、故障恢复不及时。

版权所有 © 华为云计算技术有限公司

disk_check_shared_storage_hung_timeout

参数说明：指定dorado共享磁盘I/O hang检测的超时时间。若dorado共享磁盘I/O hang检测时间超过该参数值，则系统会判定dorado共享磁盘已经I/O hang。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000。0表示关闭dorado共享磁盘I/O hang检测功能。

默认值：30

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能会导致dorado共享磁盘I/O hang检测出现假阴性或假阳性的误判情况。

process_unhealthy_alarm_time

参数说明：进程状态检测超时告警参数，当进程长时间处于某种异常状态(D/T/Z)时，超过此参数时间则触发进程异常告警。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647。0表示关闭此功能。

默认值：50

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

enable_tblspc_disk_check

参数说明：表空间磁盘检测功能开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启表空间磁盘检测功能，即表空间所在磁盘hang，中止对应DN。
- off：表示不开启表空间磁盘检测功能。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式一进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

max_network_package_loss

参数说明：检测DN主备之间在数据面的丢包率，主DN丢包率超过阈值时，触发主DN重启操作。

参数类型：整型

参数单位：%

取值范围：0 ~ 99。0表示关闭此功能。

默认值：30

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致故障恢复不及时。

enable_auto_start

参数说明：控制cm_agent是否自动拉起DN、CN。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭自动拉起。
- 1：表示开启自动拉起。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_check_az_network

参数说明：是否开启az级数据面网络检测。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启az级数据面网络检测。
- off：表示关闭az级数据面网络检测。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭会导致故障检测功能降低。

版权所有 © 华为云计算技术有限公司

expect_stop_az_name

参数说明：az级网络检测功能，期望停止的az名称。

参数类型：字符串

参数单位：无

取值范围：环境中已有的az名称。取值为空时，与cmserver主相反的az上的实例停止。

默认值：""

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

check_az_network_concurrency

参数说明：az级网络检测支持并发数。

参数类型：整型

参数单位：个

取值范围：1 ~ 10

默认值：2

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致cm故障检测和恢复功能不及时，设置过大会导致cpu负载变高。

cm_hadr_process_type

参数说明：容灾流程标识。

参数类型：枚举类型

参数单位：无

取值范围：

- none：表示当前无流程。
- failover：表示当前处于灾备集群升主流程。
- switchover_promote：表示主备集群倒换流程中灾备集群升主流程。
- switchover_demote：表示主备集群倒换流程中主集群降为灾备集群流程。

默认值：none

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

emergence_sql_check_max_cn_count

参数说明：急救室SQL检测支持的集群中最大CN个数，CN个数超过该值，则不进行SQL检测。该参数设置为0表示关闭CN检测。

参数类型：整型

参数单位：个

取值范围：0 ~ 256

默认值：16

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_topology_network_gtm_gtm_collect

参数说明：控制是否开启网络拓扑gtm与gtm之间的网络检测。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启网络检测。
- off：表示关闭网络检测。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式一或方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_collecting_group_size

参数说明：数据面网络拓扑每个采集线程采集DN分片数。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：个

取值范围：8 ~ 32

默认值：16

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式一进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

topology_network_collecting_failed_times

参数说明：数据面网络拓扑连续断连次数，断连失败次数超过此参数值，则判断为数据面网络拓扑异常。该参数仅支持QUORUM模式。

需要和cm_server参数中的topology_network_collecting_failed_times值保持一致。

参数类型：整型

参数单位：次

取值范围：3 ~ 50

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_collecting_interval

参数说明：数据面网络拓扑检测周期。该参数设置为0表示关闭网络拓扑检测。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 600

默认值：2

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_check_max_cn_count

参数说明：数据面网络拓扑检测支持的集群中最大CN个数。CN个数超过该值，则不进行数据面网络拓扑检测。该参数设置为0表示关闭网络拓扑检测。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：个

取值范围：0 ~ 256

默认值：16

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

max_cma_check_inst_timeout

参数说明：CMA检测实例的单轮超时时间，超时后，上报告警AIM_AI_CheckInstanceTooLong。该参数设置为0表示不上报此告警。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

默认值：20

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致频繁上报告警。

enable_gsmonitor

参数说明：控制GSMonitor系统资源统计工具是否开启。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启GSMonitor功能。
- off：表示关闭GSMonitor功能。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gsmonitor_max_store_duration

参数说明：控制GSMonitor日志的最大保存天数。

参数类型：整型

参数单位：d（天）

取值范围：2 ~ 30

默认值：14

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gsmonitor_max_store_space

参数说明：控制GSMonitor日志的最大保存空间。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：MB

取值范围：512 ~ 10240

默认值：2048

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

gsmonitor_interval

参数说明：控制GSMonitor的采集间隔。

参数类型：字符串

参数单位：无

取值范围：

- "interval1,interval2,interval3,interval4"

  该参数分为4个部分，形式为"interval1,interval2,interval3,interval4"

  interval1表示CPU使用率0 ~ 49%时的采集间隔，范围是2 ~ 2147483647，单位为秒。

  interval2表示CPU使用率50 ~ 69%时的采集间隔，范围是2 ~ 2147483647，单位为秒。

  interval3表示CPU使用率70 ~ 89%时的采集间隔，范围是2 ~ 2147483647，单位为秒。

  interval4表示CPU使用率90 ~ 100%时的采集间隔，范围是2 ~ 2147483647，单位为秒。
- "auto"，等同于"30,10,5,2"

默认值："30,10,5,2"

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

hadr_replconninfo_config_mode

参数说明：当前数据库实例的容灾类型。

参数类型：字符串

参数单位：无

取值范围："local_file"、"cluster_manager"

默认值："local_file"

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：此参数为内部调用参数，不建议用户修改。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

default_build_mode

参数说明：指定cm_agent自动发起主备重建时采用的重建方式。

参数类型：枚举类型

参数单位：无

取值范围：

- auto：表示主备重建时会先进行三次增量重建，若三次增量重建均失败且不是寻主失败原因所致，则改用全量重建方式进行重建。
- incremental：表示使用增量重建方式进行主备重建。
- full：表示使用全量重建方式进行主备重建。

默认值：auto

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值，如果需要调整请联系华为技术支持。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_preparse

参数说明：DN预解析检测的开关。开启后，将检测DN预解析结果，作为DN仲裁选主的依据。

参数类型：整型

参数单位：无

取值范围：

- 0：表示关闭。
- 1：表示开启。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值，如果需要调整请联系华为技术支持。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

6.4.2 cm_server 参数

log_dir

参数说明：log_dir决定存放cm_server日志文件的目录。可以是绝对路径，或者是相对路径（相对于cm_server数据目录的路径）。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：合法目录路径。

默认值：“$GAUSSLOG/cm/cm_server”，表示在$GAUSSLOG对应的cm目录下生成cm_server日志。

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_file_size

参数说明：控制日志文件的大小。当日志文件达到指定大小时，则重新创建一个日志文件记录日志信息。

参数类型：整型

参数单位：MB

取值范围：0 ~ 2047

默认值：16

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。例如，设置为16，不带单位，表示16MB；设置为16MB，带单位，表示16MB。取值如果要带单位，必须为MB。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

log_min_messages

参数说明：控制写到cm_server日志文件中的消息级别。每个级别都包含排在它后面的所有级别中的信息。级别越低，服务器运行日志中记录的消息就越少。

参数类型：枚举类型

参数单位：无

取值范围：有效值有debug5、debug1、log、warning、error、fatal。

默认值：warning

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：考虑硬盘大小后适当设置。

设置不当的风险与影响：设置级别低可能导致调试信息丢失。

thread_count

参数说明：agent线程池的线程数。

参数类型：字符串

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：该参数分为2个部分，'worker_thread_count, io_ratio'，这2个部分的具体含义如下：

- worker_thread_count：AgentWorker的线程数量，取值范围2~1000。
- io_ratio：io和worker线程的比例，值为n代表一个AgentIO线程对应n个AgentWorker线程，取值范围1~100。AgentIO线程最少一个。

说明

如果第一部分的参数，即AgentWorker的线程数量大于集群节点数与处理cm_ctl请求的线程数（集群节点数小于32默认1个线程，否则4个线程）之和，实际生效值为集群节点数与处理cm_ctl请求的线程数之和。

默认值："(1000,1)"

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：不做调整设置为默认值。若要更改，请在理解参数含义后谨慎操作，避免因误操作产生意料之外的风险。

设置不当的风险与影响：设置不当可能导致CM故障检测和恢复不及时、cpu负载变高。

thread_effective_time

参数说明：线程运行一轮的有效超时时间。超时说明此线程卡住，进程将自动退出。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1200，0表示关闭线程监控能力。

默认值：20

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式一、方式二进行设置。

设置建议：不建议调整参数值大小，如果需要调整请联系华为技术支持。

设置不当的风险与影响：设置不当可能导致故障检测不准确、故障恢复不及时。

ctl_thread_count

参数说明：ctl线程池的线程数。

参数类型：字符串

参数单位：无

取值范围：

长度大于0的字符串。该参数分为2个部分，'worker_thread_count, io_ratio'，这2个部分的具体含义如下：

- worker_thread_count：CtlWorker的线程数量，取值范围0~1000。取值为0时，CtlWorker的线程数量等于集群内节点数。
- io_ratio：io和worker线程的比例，值为n代表一个CtlIO线程对应n个CtlWorker线程，取值范围1~100。CtlIO线程最少一个。

版权所有 © 华为云计算技术有限公司

默认值："(0,1)"

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式一进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能导致CM性能降低。

alarm_component

参数说明：设置用于处理告警内容的告警组件的位置。

参数类型：字符串

参数单位：无

取值范围：合法文件路径。

- 若前置脚本gs_preinstall中的--alarm-type参数设置为5，表示未对接第三方组件，告警写入system_alarm日志，此时GUC参数alarm_component的取值为“/opt/huawei/snas/bin/snas_cm_cmd”。
- 若前置脚本gs_preinstall中的--alarm-type参数设置为1，表示对接第三方组件，此时GUC参数alarm_component的值为第三方组件的可执行程序的绝对路径。

默认值："/opt/huawei/snas/bin/snas_cm_cmd"

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

alarm_report_interval

参数说明：指定告警上报的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：3

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致告警上报不及时，过小可能导致cpu负载变高。

alarm_report_max_count

参数说明：指定告警上报的最大次数。

参数类型：整型

参数单位：次

取值范围：1 ~ 2592000

版权所有 © 华为云计算技术有限公司

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致cpu负载变高。

instance_failover_delay_timeout

参数说明：cm_server检测到主机宕机后，failover备机的延迟时间。对GTM仲裁生效。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致数据库故障恢复不及时。

instance_heartbeat_timeout

参数说明：cm_server检测实例的心跳超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：6

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

instance_keep_heartbeat_timeout

参数说明：cm_agent会定期检测实例状态并上报给cm_server，若实例状态长时间无法成功检测，累积时间超出该数值，则cm_server将下发命令给agent重启该实例。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：40

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

instance_phony_dead_restart_interval

参数说明：当CN/DN/GTM实例僵死时，会被cm_agent重启，相同的实例连续因僵死被杀时，其间隔时间不能小于该参数数值，否则cm_agent不会下发命令。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

coordinator_heartbeat_timeout

参数说明：CN故障自动剔除的心跳超时时间。设置后立即生效，不需要重启集群。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2592000，0表示CN故障后不会自动剔除。

默认值：25

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致CN长时间无法剔除，过小可能导致CN无法正常剔除。

phony_dead_effective_time

参数说明：用于CN/DN/GTM进程的僵死检测，当检测到的僵死次数大于该参数值时，认为进程僵死，将进程重启。

参数类型：整型

参数单位：次

取值范围：1 ~ 2147483647

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

版权所有 © 华为云计算技术有限公司

enable_transaction_read_only

参数说明：控制是否打开cm_server磁盘阈值检测功能，该功能打开后，当磁盘使用率大于**datastorage_threshold_value_check**值时，cm_server会自动将数据库设置为只读模式。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启阈值检测功能。
- off：表示关闭阈值检测功能。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭此功能可能导致磁盘占用过高。

enable_disk_full_check

参数说明：磁盘disk full（磁盘使用率超过**disk_full_threshold**，会停止故障节点的实例进程）检测开关，打开后，允许磁盘disk full check。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启磁盘disk full check。
- off：表示关闭磁盘disk full check。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式一进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

disk_full_threshold

参数说明：设置数据库磁盘最大的占用阈值，当数据目录所在磁盘占用超过这个阈值，会自动将数据库停止。当磁盘占用恢复到**datastorage_threshold_value_check**以下时，会自动将数据库启动。disk_full_threshold需要大于datastorage_threshold_value_check才能生效。

参数类型：整型

参数单位：%

取值范围：1 ~ 99

默认值：98

版权所有 © 华为云计算技术有限公司

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大可能导致其他进程不可用。

datastorage_threshold_check_interval

参数说明：检测磁盘占用的时间间隔。每间隔用户指定时间，检测一次磁盘占用。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2592000

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时，设置过小会导致cpu负载变高。

datastorage_threshold_value_check

参数说明：设置数据库只读模式的磁盘占用阈值，当数据目录所在磁盘占用超过这个阈值，自动将数据库设置为只读模式。由于数据库只读后无法限制日志回放，磁盘占用会继续增加，因此应当关注集群磁盘容量，及时处理数据库只读相关告警，防止出现磁盘空间写满、数据库无法快速恢复的场景。

参数类型：整型

参数单位：%

取值范围：1 ~ 99

默认值：85

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。根据磁盘空间可做适当调整，调整该参数时，建议同步调整dn的**max_size_for_xlog_retention**参数，避免因备份操作触发集群只读阈值。

该默认值设置为85%主要考虑以下因素：

- GaussDB一般用于承载重要核心业务，支持的存储容量上限较大，这类业务的并发通常会比较高，预留15%的空间以便在磁盘空间紧张的情况下，有足够的空间承载磁盘扩容前这段时间的数据增量。
- 为数据库进入只读模式后的运维操作预留磁盘空间，如处理日志、空间回收、备份等，避免发生问题后无空间余量支撑运维操作。
- 根据客户的默认阈值诉求，大部分客户默认使用85%的配置。

设置不当的风险与影响：设置过大会导致磁盘空间占用过高、出现无法快速恢复的场景。

版权所有 © 华为云计算技术有限公司

datastorage_threshold_recover

参数说明：数据库取消只读模式的缓冲阈值。数据目录所在磁盘占用超过datastorage_threshold_value_check后，被设置为只读模式。当磁盘占用恢复到小于（datastorage_threshold_value_check - datastorage_threshold_recover）后，才会取消只读模式。对于被剔除的CN，为了防止反复剔除加回的情况出现，当磁盘占用小于（datastorage_threshold_value_check - datastorage_threshold_recover）时，该CN才能被加回，超过则不加回。主数据库只读后，只有当备数据库磁盘占用小于（datastorage_threshold_value_check - datastorage_threshold_recover）时，才会将主切换到该备机。

参数类型：整型

参数单位：%

取值范围：0 ~ 99

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

max_datastorage_threshold_check

参数说明：设置磁盘使用率的最大检测间隔时间。当用户手动修改只读模式参数后，会自动在指定间隔时间后开启磁盘满检测操作。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2592000

默认值：43200

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时。

enable_az_auto_switchover

参数说明：指定是否开启自动降副本功能。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭自动降副本功能。
- 1：表示开启自动降副本功能。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

版权所有 © 华为云计算技术有限公司

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭此功能，会导致多数派故障时CM无法通过降副本功能恢复集群。

enable_az_conn_check

参数说明：AZ网络仲裁开关。cm_server会停掉被网络隔离的AZ。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭。
- 1：表示开启。开启后，cm_server会进行AZ网络仲裁。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cm_auth_method

参数说明：CM模块端口认证方式。

参数类型：枚举类型

参数单位：无

取值范围：

- trust：表示未配置端口认证。
- gss：表示采用kerberos端口认证。

默认值：trust

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：只有当kerberos服务端和客户端成功安装后才能修改为“gss”，否则CM模块无法正常通信，将影响集群状态。

cm_krb_server_keyfile

参数说明：kerberos服务端key文件所在位置，需要配置为绝对路径。该文件通常在“${GAUSSHOME}/kerberos路径”下，以keytab格式结尾，文件名与集群运行所在用户名相同。

参数类型：字符串

参数单位：无

取值范围：服务器上kerberos服务端key文件所在位置的绝对路径。

版权所有 © 华为云计算技术有限公司

默认值：“${GAUSSHOME}/kerberos/{UserName}.keytab”，默认值无法生效，仅作为提示。

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该参数与上述**cm_auth_method**参数是配对的，当cm_auth_method参数修改为“gss”时，该参数也必须配置为正确路径，否则将影响集群状态。

cm_hadr_process_type

参数说明：容灾流程标识。

参数类型：枚举类型

参数单位：无

取值范围：

- none：表示当前无流程。
- failover：表示当前处于灾备集群升主流程。
- switchover_promote：表示主备集群倒换流程中灾备集群升主流程。
- switchover_demote：表示主备集群倒换流程中主集群降为灾备集群流程。

默认值：none

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cma_fault_kill_instance_timeout

参数说明：DN实例无法上报，cmserver主内部状态置为unknown后，等待超时，cmserver通过pssh命令远程发送kill操作，重启不上报DN实例进程。此参数默认不添加到cmserver配置文件中，需要手动添加。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 21474836。0表示关闭。

默认值：不配置此参数，默认开启，默认值为：（cm_server_num * **agent_connect_timeout*** **agent_connect_retries** + **agent_heartbeat_timeout**+ 10）。

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能导致故障恢复不及时。

版权所有 © 华为云计算技术有限公司

cma_disconnect_to_cms_alarm_timeout

参数说明：CMA与CMS主断链的时间阈值。超时后，集群上报断链告警（ALM_AI_AbnormalConn）。当持续断链超过双倍阈值后，会持续上报长时间断链告警（ALM_AI_LongTimeAbnormalConn）。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

- 0：表示不上报告警（ALM_AI_AbnormalConn，ALM_AI_LongTimeAbnormalConn）。
- 1 ~ 1000：表示CMA与CMS主断链的时间。

默认值：40

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无

cm_server_arbitrate_delay_base_time_out

参数说明：cm_server仲裁延迟基础时长。cm_server主机断连后，仲裁启动计时开始，经过仲裁延迟时长后，将选出新的cm_server主机。其中仲裁延迟时长由仲裁延迟基础时长、节点index（server ID序号）和增量时长共同决定。公式为：仲裁延迟时长=仲裁延迟基础时长+节点index*仲裁延迟增量时长。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致cm_server断连故障假阳性，过大会导致CM故障恢复不及时。

cm_server_arbitrate_delay_incrememtal_time_out

参数说明：cm_server仲裁延迟增量时长。cm_server主机断连后，仲裁启动计时开始，经过仲裁延迟时长后，将选出新的cm_server主机。其中仲裁延迟时长由仲裁延迟基础时长、节点index（server ID序号）和增量时长共同决定。公式为：仲裁延迟时长=仲裁延迟基础时长+节点index*仲裁延迟增量时长。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：3

版权所有 © 华为云计算技术有限公司

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致cm_server断连故障假阳性，过大会导致CM故障恢复不及时。

cmserver_ha_connect_timeout

参数说明：cm_server连接到其他cm_server的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：2

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

cmserver_ha_heartbeat_timeout

参数说明：cm_server从其他cm_server接收心跳信号的超时时间。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：6

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大和过小会导致故障检测的假阴性和假阳性。

cmserver_ha_status_interval

参数说明：cm_server主备同步状态信息的时间间隔。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致cm故障检测和恢复功能不及时，设置过小会导致cpu负载变高。

版权所有 © 华为云计算技术有限公司

cmserver_demote_delay_on_etcd_fault

参数说明：因为etcd不健康而导致cm_server从主降为备的时间延迟。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2147483647

默认值：8

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致etcd故障检测的假阳性，过大会导致CM故障恢复不及时。

force_promote

参数说明：控制cm_server是否开启强切逻辑功能（集群状态为Unknown的时候以丢失部分数据为代价保证集群基本功能可用）。该参数同时适用于CN和DN。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭强切逻辑功能。
- 1：表示开启强切逻辑功能。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：如果用户可以接受业务（系统）发生强切导致部分数据丢失，以恢复业务为最高优先级，则可以设置为开启。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

switch_rto

参数说明：cm_server强切逻辑等待时延。在**force_promote**参数取值设置为1时，当集群的某一分片处于无主状态开始计时，等待该延迟时间后开始执行强切逻辑。

参数类型：整型

参数单位：s（秒）

取值范围：60 ~ 2147483647

默认值：600

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小会导致数据库故障假阳性，过大会导致故障恢复不及时。

版权所有 © 华为云计算技术有限公司

enable_finishredo_retrieve

参数说明：控制cm_server强切逻辑后是否对redo切除的Xlog进行数据找回。该参数取值设置为on时，发生强切后进行数据自动找回。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示进行数据找回。
- off：表示不进行数据找回。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：按照实际需要进行调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

backup_open

参数说明：灾备集群设置，控制CM是否按照灾备集群模式运行。

参数类型：整型

参数单位：无

取值范围：0 ~ 3

- 0：表示未搭建灾备关系。
- 1：表示搭建了dorado灾备关系。
- 2：表示搭建了流式灾备关系。
- 3：表示搭建了逻辑灾备关系。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：非灾备集群不能开启该参数。此参数为内部调用参数，不建议用户修改。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

install_type

参数说明：容灾集群相关的设置，用来区分集群的类型。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：表示未搭建容灾关系的集群。

版权所有 © 华为云计算技术有限公司

- 1：表示基于dorado的集群。
- 2：表示基于流式的集群。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：非灾备集群不能开启该参数。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_ssl

参数说明：ssl证书开关。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示启用ssl证书。
- off：表示不启用ssl证书。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：建议默认关闭。CM组件主要存储集群控制信息，不涉及用户数据，因此默认关闭不影响用户数据的安全性。

设置不当的风险与影响：无。

ssl_cert_expire_alert_threshold

参数说明：ssl证书过期告警时间。证书过期时间少于该时间时，上报证书即将过期告警。

参数类型：整型

参数单位：d（天）

取值范围：7 ~ 180

默认值：90

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：按照实际需要进行调整。

设置不当的风险与影响：设置过小可能会导致告警不及时。

ssl_cert_expire_check_interval

参数说明：ssl证书过期检测周期。

参数类型：整型

版权所有 © 华为云计算技术有限公司

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：86400（即1天）

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：按照实际需要进行调整。

设置不当的风险与影响：设置过小会增加CPU负载，过大会导致告警不及时。

delay_arbitrate_timeout

参数说明：设置等待跟主DN同AZ节点redo回放后升主的时间。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过大会导致DN故障恢复不及时。

dcf_heartbeat_timeout

参数说明：CM&DCF联合仲裁中DCF实例心跳超时时间。如果DCF的主心跳超时，则表示该DCF实例处于CM的脱管状态，CM会自动触发DCF的主备切换。仅在DCF自选主模式下生效。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647，设置为0表示不开启联合仲裁dcf实例的心跳超时检测。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：该功能关闭后，CM不会再处理脱管中的DCF实例，DCF实例在脱管状态下出现异常将无法得到CM的辅助仲裁。请在充分理解参数含义，并经过测试验证后进行修改。

enable_dcf

参数说明：DCF模式开关。

参数类型：布尔型

参数单位：无

版权所有 © 华为云计算技术有限公司

取值范围：

- on：表示开启DCF模式。
- off：表示关闭DCF模式。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_type

参数说明：etcd，dcc模式切换开关。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示开启etcd模式。
- 1：表示开启dcc模式。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一进行设置，修改后需要重启集群才能生效。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_log_level

参数说明：设置ddb日志级别。

参数类型：字符串

参数单位：无

取值范围：

- “NONE”：表示关闭日志打印，不能与以下日志级别混合使用。
- “RUN_ERR|RUN_WAR|RUN_INF|DEBUG_ERR|DEBUG_WAR|DEBUG_INF|TRACE|PROFILE|OPER”：表示开启日志，日志级别可以从上述字符串中选取字符串并使用竖线组合使用，不能配置空字符串。

默认值："RUN_ERR|RUN_WAR|DEBUG_ERR|OPER|RUN_INF|PROFILE"

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

ddb_log_backup_file_count

参数说明：最大保存日志文件个数。

参数类型：整型

参数单位：无

取值范围：1 ~ 100

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_max_log_file_size

参数说明：单条日志最大字节数。

参数类型：字符串

参数单位：无

取值范围：该参数取值由数字部分和单位组成，数字部分取值范围为1 ~ 1000，单位为M。

默认值："10M"

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_log_suppress_enable

参数说明：控制是否开启日志抑制功能。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭日志抑制功能。
- 1：表示开启日志抑制功能。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_election_timeout

参数说明：dcc选举超时时间。

版权所有 © 华为云计算技术有限公司

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 600

默认值：3

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_dcf_sync_nodes

参数说明：设置dcc同步复制的备机列表。

参数类型：字符串

参数单位：无

取值范围：合法的备机列表，支持按如下格式配置：

ANY *num_sync:node_id*[, ...]

说明

*num_sync*表示ddb需要同步复制的备机数量，node_id表示备机的序号。例如：ddb_dcf_sync_nodes='ANY 1:2,3'。

默认值：""

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

ddb_dcf_extreme_mode

参数说明：设置dcc的日志提交策略。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：表示最大保护，多数派提交。
- 1：表示最大可用，备机故障时主机可直接提交，备机正常时多数派提交。
- 2：表示最大性能，本地提交。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

版权所有 © 华为云计算技术有限公司

alarm_ddb_unhealth_threshold

参数说明：ddb不健康告警上报阈值。

参数类型：整型

参数单位：无

取值范围：0 ~ 2592000

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_e2e_rto

参数说明：端到端RTO功能开关，开启后僵死检测周期及网络检测超时时间将缩短，CM可以达到端到端RTO指标（单实例故障RTO<=10s，叠加故障RTO<=30s）。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭端到端RTO功能。
- 1：表示开启端到端RTO功能。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：建议设置为默认值，不做调整。如场景需要，可做适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_synclist_single_inst

参数说明：控制是否开启降副本降至一主零备功能。开启后降副本会降至一主零备，参数设置错误时，按默认值处理。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示开启降副本降至一主零备功能。
- off：表示关闭降副本降至一主零备功能。

默认值：off

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。如果需要调整请联系华为技术支持。

版权所有 © 华为云计算技术有限公司

设置不当的风险与影响：打开此参数时，在所有备机故障情况下，主机会自动转为本地提交方式，此时如果主机再故障，则可能产生不可恢复的影响。

asyn_log_enable

参数说明：异步日志开关，打开后cm_server启动异步日志。

参数类型：布尔型

参数单位：无

取值范围：

- on：表示cm_server开启异步日志。
- off：表示cm_server不开启异步日志，使用串行日志。

默认值：on

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：关闭可能导致日志丢失。

asyn_log_max_buf_size

参数说明：异步日志缓存大小，当异步日志发生堆积，且积压的日志超过了缓存大小，后续的日志无法记录。

参数类型：整型

参数单位：kB

取值范围：1 ~ 1048576

默认值：1024

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：根据内存和硬盘大小适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

asyn_log_fault_solution

参数说明：识别到日志磁盘故障后，异步日志处理方案。当CM识别到日志盘故障或异步日志缓存溢出，CM会根据配置来处理异步日志。

参数类型：整型

参数单位：无

取值范围：0 ~ 2

- 0：表示不对日志做多余的处理，缓存区保留所有日志，直到缓存大小达到上限，达到上限后不会记录更多日志。
- 1：表示对日志进行筛选，识别到日志磁盘故障后，只将关键日志、告警和ERROR级别以上的日志保存到缓存区，直到缓存大小达到上限，达到上限后不会记录更多日志。

版权所有 © 华为云计算技术有限公司

- 2：表示丢弃日志，识别到日志磁盘故障后，只将关键日志、告警保存到缓存区，直到缓存大小达到上限，达到上限后不会记录更多日志。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：根据内存和硬盘大小适当调整。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

enable_isolate

参数说明：隔离实例开关。

参数类型：整型

参数单位：无

取值范围：0 ~ 1

- 0：表示关闭该功能。
- 1：表示开启该功能。开启后，cm_server会自动隔离故障实例。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

barrier_restart_threshold

参数说明：灾备集群barrier重启等待时间。当灾备集群barrier推进持续停滞超过该时间后，会重启barrier推进。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2592000，0表示禁止barrier重启。

默认值：30

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置不当可能导致故障检测不准确、故障恢复不及时。

enable_topology_network_arbitrate

参数说明：网络拓扑仲裁参数，开启该参数GaussDB将根据网络拓扑结构进行仲裁。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：无

取值范围：

 版权所有 © 华为云计算技术有限公司

- 0：表示关闭。
- 1：表示开启。开启后，cm_server会自动进行网络拓扑仲裁。

默认值：1

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_arbitrate_wait_time

参数说明：网络拓扑仲裁等待网络拓扑数据稳定的时间。如果在该参数时间内，网络拓扑数据发生变化，则不会进行网络拓扑仲裁。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：s（秒）

取值范围：5 ~ 600

默认值：10

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_collecting_failed_times

参数说明：数据面网络拓扑连续断连次数，断连失败次数超过此参数，则判断为数据面网络拓扑异常。需要和cma的topology_network_collecting_failed_times值保持一致。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：次

取值范围：3 ~ 50

默认值：5

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_stable_cn_percent

参数说明：数据面网络拓扑判定CN可负载的剩余CN比例，网络仲裁根据该参数计算最少保留的CN个数（所有CN个数 * 该比例，如果计算个数小于1则取1）。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：无

取值范围：0 ~ 100

版权所有 © 华为云计算技术有限公司

默认值：50

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

topology_network_check_max_cn_count

参数说明：数据面网络拓扑检测支持的集群中最大CN个数，CN个数超过该值，则不进行数据面网络拓扑仲裁。参数设置为0表示关闭网络仲裁。该参数仅支持QUORUM模式。

参数类型：整型

参数单位：个

取值范围：0 ~ 256

默认值：16

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

hardware_max_cn_delete_threshold

参数说明：硬件故障最大剔除CN的阈值，计算数值向下取整。

参数类型：整型

参数单位：%

取值范围：0 ~ 99

- 0：表示不剔除CN。
- 1 ~ 99：表示剔除CN的百分比。最少要保留一个可用的CN。

默认值：50

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cluster_starting_aribt_delay

参数说明：cm_server在集群启动阶段，等待DN静态主升主的时间。

参数类型：整型

参数单位：s（秒）

取值范围：1 ~ 2592000

默认值：180

版权所有 © 华为云计算技术有限公司

设置方式：该参数为CM参数，参数修改请参考**表6-2**中方式一、方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cluster_unavailable_alarm_timeout

参数说明：集群处于不可用状态的时间阈值。超时后，集群上报不可用告警（ALM_AI_ClusterUnavailable）。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

- 0：表示不上报此告警。
- 1 ~ 1000：表示集群处于不可用状态的时间。

默认值：40

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无

cluster_degrade_alarm_timeout

参数说明：集群处于Degrade状态的时间阈值。超时后，集群上报Degrade告警（ALM_AI_ClusterDegrade）。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

- 0：表示不上报此告警。
- 1 ~ 1000：表示集群处于Degrade状态的时间。

默认值：40

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无

max_frequency_heal_timeout

参数说明：集群在此时间内触发治愈次数达到10次后，会触发告警（ALM_AI_ClusterHealFrequent）。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 1000

版权所有 © 华为云计算技术有限公司

- 0：表示不上报此告警。
- 1 ~ 1000：表示连续触发治愈的时间范围。

默认值：300

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：无

cn_auto_repair_max_concurrency

参数说明：并发加回CN的最大个数。

参数类型：整型

参数单位：无

取值范围：0 ~ 256，设置为0表示同时并发加回的CN数量与集群内Normal CN的数量一致且最大不超过10个。

默认值：0

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cn_auto_repair_timeout

参数说明：CN加回超时时间。CN加回失败后，如果持续时间已经超过该值，并且超过CN加回重试次数后，会放弃对故障CN的自动加回。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

默认值：1800

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cn_auto_repair_retry_times

参数说明：CN加回重试次数。CN加回失败后，如果重试次数超过该值，并且超过CN加回超时时间后，会放弃对故障CN的自动加回。放弃对故障CN自动加回后，调整该值，会重启对故障CN的自动加回。

参数类型：整型

参数单位：无

取值范围：0 ~ 2147483647

版权所有 © 华为云计算技术有限公司

默认值：3

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

cn_dn_disconnect_to_delete_time

参数说明：CN在所有DN主执行检测SQL都失败的次数。当失败次数超过该值时，则剔除该CN。

参数类型：整型

参数单位：次

取值范围：1 ~ 600

默认值：20

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

majority_standby_lock_timeout

参数说明：多数派备机被锁超时时间。集群进入仲裁选主流程，锁住多数派备机，当主出现异常，仲裁流程会持续阻塞，超过该值后，会尝试停掉原主，恢复仲裁流程。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647，0表示关闭该功能。

默认值：120

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：请在充分理解参数含义，并经过测试验证后进行修改。

replica_reduce_delay_time

参数说明：降副本的延迟时间。当检测到多数派数据节点处于unknown状态时，需持续观测该状态达到此延迟时间后，才会触发降副本操作；当检测到多数派数据节点处于building/need_repair/starting等不可用状态时，需持续观测该状态达到此延迟时间的5倍后，才会触发降副本操作。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：12

版权所有 © 华为云计算技术有限公司

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，短时间网络断连等故障导致误降副本的事件概率增加；设置过大，多数派节点故障时集群会长时间不可用。请在充分理解参数含义，并经过测试验证后进行修改。

replica_increase_delay_time

参数说明：升副本的延迟时间。当检测到副本状态恢复可以加回（即副本状态为normal且日志同步大于等于99%）时，需等待此延迟时间，才会触发自动升副本操作。此参数用于避免降副本后立即升副本导致的资源震荡。

参数类型：整型

参数单位：s（秒）

取值范围：0 ~ 2147483647

默认值：60

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：推荐使用默认值。

设置不当的风险与影响：设置过小，降副本后立即升副本会导致资源震荡；设置过大，副本恢复正常后升副本时间会变长。请在充分理解参数含义，并经过测试验证后进行修改。

dr_dest_name

参数说明：当前集群可以给哪些目标集群同步日志，该参数只在集群为单节点形态下才有效，且该参数设置后不需要reload加载参数。

参数类型：字符串

参数单位：无

取值范围：合法的目标集群实例ID。

默认值：""

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。该参数设置后不需要reload加载参数。

设置建议：此参数为内部调用参数，不建议用户修改。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

hadr_replconninfo_config_mode

参数说明：当前数据库实例的容灾类型。

参数类型：字符串

参数单位：无

取值范围："local_file"/"cluster_manager"

版权所有 © 华为云计算技术有限公司

默认值："local_file"

设置方式：该参数为CM参数，参数修改请参考**表6-3**中方式二进行设置。

设置建议：此参数为内部调用参数，不建议用户修改。

设置不当的风险与影响：修改该参数取值可能导致集群出现意料之外的结果，请保持默认值。

版权所有 © 华为云计算技术有限公司

7 系统表和系统视图

7.1 系统表和系统视图概述

系统表是GaussDB存放结构元数据的地方，它是GaussDB数据库系统运行控制信息的来源，是数据库系统的核心组成部分。

系统视图提供了查询系统表和访问数据库内部状态的方法。

系统表和系统视图要么只对管理员可见，要么对所有用户可见。下面的系统表和视图有些标识了需要管理员权限，这些系统表和视图只有管理员可以查询。

用户可以删除后重新创建这些表、增加列、插入和更新数值，但是用户修改系统表会导致系统信息的不一致，从而导致系统控制紊乱。正常情况下不应该由用户手工修改系统表或系统视图，或者手工重命名系统表或系统视图所在的模式，而是由SQL语句关联的系统表操作自动维护系统表信息。

须知

- 不建议用户修改系统表和系统视图的权限。
- 用户应该禁止对系统表进行增删改等操作，人为对系统表的修改或破坏可能会导致系统各种异常情况甚至集群不可用。
- 系统表和系统视图中与外键相关的字段暂不支持。
- 系统表和系统视图中的字段类型详见**数据类型**章节介绍。
- 对于ADM类视图，由于访问到的对象是数据库下所有的该类型的对象，考虑对ADM类视图进行统一权限管理，默认只有系统管理员有权限访问该类型视图，部分ADM类的视图数据来自基表中公开、非敏感的字段；对于DB类视图，查询到的是数据库内当前用户有权限访问的对象，普通用户即可访问该类视图；对于MY类视图，查询到的是当前用户所属的对象，普通用户即可访问该类视图。

7.2 系统表

7.2.1 分区表

版权所有 © 华为云计算技术有限公司

7.2.1.1 PG_PARTITION

PG_PARTITION系统表存储数据库内所有分区表（partitioned table）、分区（table partition）和索引分区（index partition）三类对象的信息。分区表索引（partitioned index）的信息不在PG_PARTITION系统表中保存。由于分区表（partitioned table）没有实际的物理文件，所以在pg_partition中不会记录其relfilenode、relpages、reltuples、reltoastrelid、reltoastidxid等信息。

表 7-1 PG_PARTITION 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
relname	name	分区表、分区、分区上toast表和索引分区的名称。
parttype	"char"	对象类型： ● 'r'：partitioned table ● 'p'：table partition ● 's'：table subpartition ● 'x'：index partition
parentid	oid	● 当对象为分区表或分区时，此字段表示分区表在PG_CLASS中的OID。 ● 当对象为index partition时，此字段表示所属分区表索引(partitioned index)的OID。
rangenum	integer	保留字段。
intervalnum	integer	保留字段。
partstrategy	"char"	分区表分区策略，现在仅支持： ● 'r'：范围分区。 ● 'l'：list分区。 ● 'h'：hash分区。 ● 'n'：无分区策略，该对象不是表分区。
relfilenode	oid	table partition、index partition、分区上toast表的物理存储位置。
reltablespace	oid	table partition、index partition、分区上toast表所属表空间的OID。
relpages	double precision	统计信息：table partition、index partition的数据页数量。
reltuples	double precision	统计信息：table partition、index partition的元组数。
relallvisible	integer	统计信息：table partition、index partition的可见数据页数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
reltoastrelid	oid	table partition所对应toast表的OID。
reltoastidxid	oid	table partition所对应toast表的索引的OID。
indextblid	oid	index partition对应table partition的OID。
indisusable	boolean	索引分区是否可用。 ● t（true）：表示可用。 ● f（false）：表示不可用。
reldeltarelid	oid	暂不支持。
reldeltaidx	oid	暂不支持。
relcudescrelid	oid	暂不支持。
relcudescidx	oid	暂不支持。
relfrozenxid	xid32	冻结事务ID号。 为保持前向兼容，保留此字段，新增relfrozenxid64用于记录此信息。
intspnum	integer	间隔分区所属表空间的个数。
partkey	int2vector	分区键的列号。
intervaltablespace	oidvector	间隔分区所属的表空间，间隔分区以round-robin方式落在这些表空间内。
interval	text[]	间隔分区的间隔值。
boundaries	text[]	范围分区和间隔分区的上边界。
transit	text[]	间隔分区的跳转点。
reloptions	text[]	设置partition的存储属性，与pg_class.reloptions的形态一样，用"keyword=value"格式的字符串来表示，目前用于在线扩容的信息搜集。
relfrozenxid64	xid	冻结事务ID号。
relminmxid	xid	冻结多事务ID号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
partitionno	integer	用于维护分区表中的分区Map结构。 • 当对象为分区时，此字段表示分区ID，从1开始自增。 • 当对象为分区表时，此字段表示分区ID的最大值，并使用负值来特殊标记，该值会随着部分分区DDL语法不断递增。 • 当对象为其他类型时，此字段为空值，没有任何含义。 partitionno是一个永久自增列，可以通过语法ALTER TABLE t_name RESET PARTITION命令重置/回收。
subpartitionno	integer	保留字段。
partkeyexpr	pg_node_tree	暂不支持。

7.2.1.2 GS_SUBPARTITION_TEMPLATE

GS_SUBPARTITION_TEMPLATE系统表记录了二级分区模板的信息。

表 7-2 GS_SUBPARTITION_TEMPLATE 字段

名称	类型	描述
table_oid	oid	模板所属基表的OID。
position	integer	模板定义的分区编号，从1开始。
spart_name	name	模板定义的分区名。
spart_tablespace	oid	模板定义的分区表空间OID。
spart_boundaries	text[]	模板定义的分区边界。对于RANGE分区，定义为上边界；对于LIST分区，定义为列表值；对于HASH分区，定义为HASH取余值。

7.2.2 负载管理

7.2.2.1 GS_ABNORMAL_SQL_INFO

GS_ABNORMAL_SQL_INFO系统表提供了数据库CPU使用异常的SQL的信息。

版权所有 © 华为云计算技术有限公司

表 7-3 GS_ABNORMAL_SQL_INFO 字段

名称	类型	描述
unique_sql_id	bigint	sql唯一id。
abnormal_count	integer	异常次数。
current_controllable	boolean	CPU使用当前是否被管控。
is_logged_when_controlled	boolean	被管控时是否打印日志。
control_expiration_time	timestamp without time zone	管控过期时间。
abnormal_type	name	异常类型。
user_id	oid	异常SQL所属的用户oid。

7.2.2.2 GS_ABNORMAL_SQL_WHITE_LIST

GS_ABNORMAL_SQL_WHITE_LIST系统表提供了数据库CPU使用异常SQL白名单的信息。

表 7-4 GS_ABNORMAL_SQL_INFO 字段

名称	类型	描述
unique_sql_id	bigint	sql唯一id。
user_id	oid	异常SQL所属的用户oid。

7.2.2.3 PG_APP_WORKLOADGROUP_MAPPING

PG_APP_WORKLOADGROUP_MAPPING系统表提供了数据库负载映射组的信息。

表 7-5 PG_APP_WORKLOADGROUP_MAPPING 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
appname	name	应用名称。
workload_gpname	name	映射到的负载组名称。

7.2.3 OLTP 表压缩

版权所有 © 华为云计算技术有限公司

7.2.3.1 GS_ILM

GS_ILM系统表提供了ILM策略的主体信息，其中包括策略名称、策略属主、策略类型、策略编号和策略状态。

表 7-6 GS_ILM 字段

名称	类型	描述
pidx	integer	策略序号，全局唯一以1开始的流水号。
creator	oid	策略属主。
name	name	策略名称，目前不支持自定义名称，默认名称为：p+pidx。
ptype	"char"	策略类型： ● m：DATA MOVEMENT
flag	smallint	取值范围： ● 0：策略整体启用状态。 ● 1：策略整体被禁用。 ● 2：库级策略。

7.2.3.2 GS_ILM_JOBDETAIL

GS_ILM_JOBDETAIL系统表记录了Job类型、Job状态、Job统计信息等。

表 7-7 GS_ILM_JOBDETAIL 字段

名称	类型	描述
taskoid	bigint	所属ADO Task的Oid。
jobtype	"char"	当前Job类型：压缩（‘c’）。
jobstatus	smallint	当前Job状态： ● 1：JOB CREATED ● 2：COMPLETED SUCCESSFULLY ● 3：FAILED ● 4：STOPPED ● 5：JOB CREATION FAILED
jobname	text	ADO Job名称表内唯一，关联系统表PG_JOB的job_name。
starttime	timestamp with time zone	启动时间。
completetime	timestamp with time zone	完成时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
payload	text	当前Job的具体执行内容PL/SQL代码。
statistics	text	当前Job的统计信息。
comments	text	记录当前Job执行的描述信息，如失败原因等。

7.2.3.3 GS_ILM_OBJECT

GS_ILM_OBJECT记录数据对象与策略的关系，同时记录策略在该数据对象上的调度信息等。注意当给分区表的表上设置策略时，此表会为每一个分区/子分区都生成独立的一条记录。

表 7-8 GS_ILM_OBJECT 字段

名称	类型	描述
pidx	integer	策略编号。
objoid	oid	逻辑数据对象oid有分区的表。
dataobjoid	oid	物理数据对象oid，真正需要执行策略的实体，树级分区关系的叶子。
objtype	"char"	当前数据对象类型： ● 表（‘r’） ● 分区（‘p’） ● 二级分区（‘s’） ● 普通索引（‘i’） ● 全局索引（‘I’） ● 全局二级索引（‘G’） ● 索引分区（‘x’）
origobjoid	oid	定义该策略的原始对象oid。
origobjtype	"char"	定义该策略的原始数据对象类型： ● 表（‘r’） ● 分区（‘p’） ● 二级分区（‘s’） ● 普通索引（‘i’） ● 全局索引（‘I’） ● 全局二级索引（‘G’）
lastchktime	timestamp with time zone	最近一次ADO Task时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
lastexetime	timestamp with time zone	最近一次ADO Job时间。
roundcnt	smallint	成功回合的次数。
failcnt	smallint	失败次数。
lastjobstatus	"char"	上一次Job的执行状态。
lastroundstarttime	timestamp with time zone	该轮全表扫描开始的时间。
lastjobblkid	bigint	上一次Job处理完成的BlockId。
flag	smallint	取值范围： • 0：启用对象压缩策略。 • 1：禁用对象压缩策略。 • 2：启用对象压缩策略，上次压缩调度完成1次全表扫描。
lasthashbktpos	oidvector	记录上一轮压缩调度最后压缩的位置信息，记录两个值[bucketid, blockid]。

7.2.3.4 GS_ILM_PARAM

GS_ILM_PARAM系统表为特性参数表，记录控制ILM特性相关参数，如后台调度的各种控制参数。

表 7-9 GS_ILM_PARAM 字段

名称	类型	描述
idx	smallint	参数编号。
name	name	参数名称。
value	double precision	参数值，详见表7-10的描述列。

表 7-10 GS_ILM_PARAM 特性参数范围

参数编号	参数名称	描述
1	EXECUTION_INTERVAL	ADO Task的执行频率，单位分钟，默认值15。取值范围为大于等于1小于等于2147483647的整数或浮点数，作用时向下取整。

版权所有 © 华为云计算技术有限公司

参数编号	参数名称	描述
2	RETENTION_TIME	ADO相关历史的保留时长，单位天，默认值30。取值范围大于等于1小于等于2147483647的整数或浮点数，作用时向下取整。
7	ENABLE	后台调度的状态。该参数不支持使用DBE_ILM_ADMIN.CUSTOMIZE_ILM修改，应使用disable_ilm()和enable_ilm()修改。
11	POLICY_TIME	控制ADO的条件单位是天还是秒，秒仅用来做测试用。 取值范围： ● 0：单位为天，默认值为0。 ● 1：单位为秒，一般用于测试场景。
12	ABS_JOBLIMIT	控制一次ADO Task最多生成的ADO Job个数，默认值10。取值范围大于等于0小于等于2147483647的整数或浮点数，作用时向下取整。
13	JOB_SIZELIMIT	控制单个ADO Job可以处理的最大字节数，单位兆，默认值1024。取值范围大于等于0小于等于2147483647的整数或浮点数，作用时向下取整。
14	WIND_DURATION	维护窗口持续时长，单位分钟，默认240分钟（4小时）；取值范围为大于等于0小于1440（24小时）的整数或浮点数，作用时向下取整。
15	BLOCK_LIMITS	控制实例级的行存压缩速率上限，默认是40；取值范围是0到10000（0表示不限制）；单位是block/ms，表示每毫秒最多压缩多少个block。
16	ENABLE_META_COMPRESSION	是否开启header压缩，默认为0，0表示关闭，1表示开启。 **说明** 设置此参数为1时，对于单行数据较短的表，压缩率会有一定提升，但是访问压缩行的性能会有较大幅度的下降。若数据库多是单行数据较长的表，不建议开启此参数。
17	SAMPLE_MIN	常量编码和等值编码采样步长最小值，默认为10，取值范围[1, 100]，支持小数输入，小数会自动向下取整。
18	SAMPLE_MAX	常量编码和等值编码采样步长最大值，默认为10，取值范围[1, 100]，支持小数输入，小数会自动向下取整。
19	CONST_PRIO	常量编码优先级，默认为40，取值范围[0, 100]，100表示关闭常量编码，支持小数输入，小数会自动向下取整。
20	CONST_THRESHOLD	常量编码阈值，默认为90，取值范围[1, 100]，表示一列常量值的占比超过该阈值时进行常量编码，支持小数输入，小数会自动向下取整。

版权所有 © 华为云计算技术有限公司

参数编号	参数名称	描述
21	EQVALUE_PRIO	等值编码优先级，默认为60，取值范围[0, 100]，100表示关闭等值编码，支持小数输入，小数会自动向下取整。
22	EQVALUE_THRESHOLD	等值编码阈值，默认为80，取值范围[1, 100]，表示两列数据的等值比例超过该阈值时进行等值编码，支持小数输入，小数会自动向下取整。
23	ENABLE_DELTA_ENCODE_SWITCH	差值编码开关，默认为1，支持小数输入，0表示关闭，1表示开启，小数会自动向下取整。
24	LZ4_COMPRESSION_LEVEL	lz4压缩等级，默认为0，取值范围[0, 16]，支持小数输入，小数会自动向下取整。
25	ENABLE_LZ4_PARTIAL_DECOMPRESSION	部分解压开关，默认为1，支持小数输入，0表示关闭，1表示开启，小数会自动向下取整。
26	INDEXJOB_SIZELIMIT	控制索引压缩单个ADO Job可以处理的最大字节数，单位兆，默认值1024。取值范围大于等于0小于等于2147483647的整数或浮点数，作用时向下取整。
27	ENABLE_INDEX_COMPRESS	索引压缩开关，默认为0，取值为0或1，0表示关闭，1表示开启。

7.2.3.5 GS_ILM_POLICY

GS_ILM_POLICY系统表提供了记录策略动作类型、判定条件等。

表 7-11 GS_ILM_POLICY 字段

名称	类型	描述
pidx	integer	策略序号。
action	"char"	策略动作，仅支持压缩（‘c’）。
ctype	"char"	压缩类型：仅支持高级行压缩（‘a’），跨页透明压缩（‘t’）。
condition	"char"	判定条件类型，仅支持LAST MODIFICATION（‘m’）。
days	smallint	判定天数量。
scope	"char"	策略作用域：仅支持Row（‘r’），Unit（‘u’）。
predicate	pg_node_tree	策略行级表达式。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
csubtype	"char"	压缩级别：高级压缩仅支持Medium（‘m’）和High（‘h’）；跨页透明压缩支持Low（‘l’），Medium（‘m’），High（‘h’）。

7.2.3.6 GS_ILM_TASK

GS_ILM_TASK系统表记录ADO Task的主体信息（创建时间、开始完成时间、执行状态等）。

表 7-12 GS_ILM_TASK 字段

名称	类型	描述
taskoid	bigint	ADO Task的oid。
creator	oid	ADO Task发起人。
createtime	timestamp with time zone	创建时间。
starttime	timestamp with time zone	启动时间。
completetime	timestamp with time zone	完成时间。
executestate	smallint	取值范围： ● 1：'INACTIVE' ● 2：'ACTIVE' ● 3：'COMPLETED'
flag	smallint	预留字段。

7.2.3.7 GS_ILM_TASKDETAIL

GS_ILM_TASKDETAIL系统表记录一次指定数据对象与策略的评估结果。

表 7-13 GS_ILM_TASKDETAIL 字段

名称	类型	描述
pidx	integer	策略序号。
objoid	oid	数据对象oid。
objtype	"char"	数据对象类型。
taskoid	bigint	所属ADO Task的oid。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
evalresult	smallint	评估结果： • 0：SELECTED FOR EXECUTION，评估通过。 • 1：PRECONDITION NOT SATISFIED，条件不满足。 • 2：JOB ALREADY EXISTS，Job已存在。
jobname	text	评估通过后生成的ADO Job Name关联系统表GS_ILM_JOBDETAIL的jobname字段。

7.2.3.8 GS_ILM_TICKER

GS_ILM_TICKER系统表记录LSN以及时间的映射关系，最多保留3650行记录。

表 7-14 GS_ILM_TICKER 字段

名称	类型	描述
ilm_ticker_lsn	bigint	触发打点操作时的当前LSN。
ilm_ticker_time	timestamp with time zone	触发打点操作时的当前时间。

7.2.3.9 GS_ILM_TURBO_OBJECT

GS_ILM_TURBO_OBJECT系统表记录跨页透明压缩策略信息。

表 7-15 GS_ILM_TURBO_OBJECT 字段

名称	类型	描述
objoid	oid	定义该策略的原始对象oid。
objtype	"char"	当前数据对象类型： • 表（‘r’） • 分区（‘p’） • 二级分区（‘s’） • 普通索引（‘i’） • 全局索引（‘I’） • 全局二级索引（‘G’） • 索引分区（‘x’）

版权所有 © 华为云计算技术有限公司

名称	类型	描述
csubtype	"char"	跨页透明压缩策略级别： • 低（‘l’） • 中（‘m’） • 高（‘h’）
encodetype	tinyint	编码类型 • 平铺（‘1’） • 模糊分列（‘2’） • 模式编码（‘3’）
compresstype	tinyint	通用压缩器类型 • lz4（‘1’） • zstd（‘2’）
compressunit	tinyint	压缩单元长度 • 1（‘1’） • 2（‘2’） • 4（‘3’） • 8（‘4’） • 16（‘5’） • 32（‘6’） • 64（‘7’）
lmtinterval	integer	冷热管理时间间隔，单位：小时

7.2.4 密态等值查询

7.2.4.1 GS_CLIENT_GLOBAL_KEYS

GS_CLIENT_GLOBAL_KEYS系统表记录密态等值特性中客户端加密主密钥相关信息，每条记录对应一个客户端加密主密钥。

表 7-16 GS_CLIENT_GLOBAL_KEYS 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
global_key_name	name	客户端加密主密钥(cmk)名称。
key_namespace	oid	包含此客户端加密主密钥(cmk)的命名空间OID。
key_owner	oid	客户端加密主密钥(cmk)的所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
key_acl	aclitem[]	创建该密钥时所拥有的访问权限。
create_date	timestamp without time zone	创建密钥的时间。

7.2.4.2 GS_CLIENT_GLOBAL_KEYS_ARGS

GS_CLIENT_GLOBAL_KEYS_ARGS系统表记录密态等值特性中客户端加密主密钥相关元数据信息，每条记录对应客户端加密主密钥的一个键值对信息。

表 7-17 GS_CLIENT_GLOBAL_KEYS_ARGS 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
global_key_id	oid	客户端加密主密钥(cmk)oid。
function_name	name	值为encryption。
key	name	客户端加密主密钥(cmk)的元数据信息对应的名称。
value	bytea	客户端加密主密钥(cmk)的元数据信息名称的值。

7.2.4.3 GS_COLUMN_KEYS

GS_COLUMN_KEYS系统表记录密态等值特性中列加密密钥相关信息，每条记录对应一个列加密密钥。

表 7-18 GS_COLUMN_KEYS 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
column_key_name	name	列加密密钥(cek)名称。
column_key_distributed_id	oid	根据加密密钥(cek)全称域名哈希值得到的id。
global_key_id	oid	外键。客户端加密主密钥(cmk)的OID。
key_namespace	oid	包含此列加密密钥(cek)的命名空间OID。
key_owner	oid	列加密密钥(cek)的所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
create_date	timestamp without time zone	创建列加密密钥的时间。
key_acl	aclitem[]	创建该列加密密钥时所拥有的访问权限。

7.2.4.4 GS_COLUMN_KEYS_ARGS

GS_COLUMN_KEYS_ARGS系统表记录密态等值特性中，客户端加密主密钥相关元数据信息，每条记录对应客户端加密主密钥的一个键值对信息。

表 7-19 GS_COLUMN_KEYS_ARGS 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
column_key_id	oid	列加密密钥(cek)oid。
function_name	name	值为encryption。
key	name	列加密密钥(cek)的元数据信息对应的名称。
value	bytea	列加密密钥(cek)的元数据信息名称的值。

7.2.4.5 GS_ENCRYPTED_COLUMNS

GS_ENCRYPTED_COLUMNS系统表记录密态等值特性中表的加密列相关信息，每条记录对应一条加密列信息。

表 7-20 GS_ENCRYPTED_COLUMNS 字段

名称	类型	描述
rel_id	oid	表的OID。
column_name	name	加密列的名称。
column_key_id	oid	外键，列加密密钥的OID。
encryption_type	tinyint	加密类型。取值为2(DETERMINISTIC)或者1(RANDOMIZED)。
data_type_original_oid	oid	加密列的原始数据类型id。该值参考系统表**PG_TYPE**的oid字段。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
data_type_original_mod	integer	加密列的原始数据类型修饰符。该值参考原始数据类型对应的系统表PG_ATTRIBUTE中的atttypmod字段。
create_date	timestamp without time zone	创建加密列的时间。

7.2.4.6 GS_ENCRYPTED_PROC

GS_ENCRYPTED_PROC系统表提供了密态函数/存储过程函数参数、返回值的原始数据类型，加密列等信息。

表 7-21 GS_ENCRYPTED_PROC 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
func_id	oid	function的oid，对应系统表PG_PROC中的oid行标识符。
prorettype_orig	integer	返回值的原始数据类型。
last_change	timestamp without time zone	密态函数上次修改的时间。
proargcachedcol	oidvector	函数INPUT参数对应的加密列的oid，对应系统表GS_ENCRYPTED_COLUMNS中的oid行标识符。
proallargtypes_orig	oid[]	所有函数参数的原始数据类型。

7.2.5 通信

7.2.5.1 PGXC_NODE

PGXC_NODE系统表存储集群节点信息。

须知

- PGXC_NODE系统表存储数据库实例节点信息。PGXC_NODE系统表仅在CN下有具体含义且数据有效正确。
- 在DN节点查询PGXC_NODE系统表结果数据中只有node_id字段是有意义，其他字段值没有意义且为无效数据。

版权所有 © 华为云计算技术有限公司

表 7-22 PGXC_NODE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
node_name	name	节点名称。
node_type	"char"	节点类型，其中R/M/O类型为灾备集群特有类型。 ● C：协调节点。 ● D：数据节点。 ● S：数据节点的备节点。 ● R：灾备集群的协调读节点。 ● M：灾备集群的静态主数据节点。 ● O：灾备集群的其他数据节点。
node_port	integer	节点的端口号。
node_host	name	节点的主机名称或者IP（如配置为虚拟IP，则为虚拟IP）。
node_port1	integer	复制节点的端口号。
node_host1	name	复制节点的主机名称或者IP（如配置为虚拟IP，则为虚拟IP）。
hostis_primary	boolean	表明当前节点是否发生主备切换。 ● t（true）：表示发生。 ● f（false）：表示不发生。
nodeis_primary	boolean	在replication表下，是否优选当前节点作为优先执行的节点进行非查询操作。 ● t（true）：表示优选。 ● f（false）：表示不优选。
nodeis_preferred	boolean	在replication表下，是否优选当前节点作为首选的节点进行查询。 ● t（true）：表示优选。 ● f（false）：表示不优选。
node_id	integer	节点标识符。由node_name经过哈希函数计算后得到。
sctp_port	integer	主节点使用TCP代理通信库的数据通道侦听端口（当前版本已经不再支持SCTP通信库）。
control_port	integer	主节点使用TCP代理通信库的控制通道侦听端口。
sctp_port1	integer	备节点使用TCP代理通信库的数据通道侦听端口（当前版本已经不再支持SCTP通信库）。
control_port1	integer	备节点使用TCP代理通信库的控制通道侦听端口。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
nodeis_central	boolean	表明当前节点是否为中心控制节点，只用于CN，对DN无效。 ● t（true）：表示是。 ● f（false）：表示不是。
nodeis_active	boolean	表明当前节点是否是正常状态，用于标记CN是否被剔除，对DN无效，灾备集群节点此字段无效。 ● t（true）：表示是。 ● f（false）：表示不是。
cluster_name	text	集群名称。
slice_order	integer	节点归属分片的顺序序号。

7.2.6 账本数据库

7.2.6.1 GS_GLOBAL_CHAIN

GS_GLOBAL_CHAIN系统表记录用户对防篡改用户表的修改操作信息，每条记录对应一次表级修改操作。具有审计管理员权限的用户可以查询此系统表，所有用户均不允许修改此系统表。

表 7-23 GS_GLOBAL_CHAIN 字段

名称	类型	描述
blocknum	bigint	区块号，当前用户操作在账本中记录的序号。
dbname	name	数据库名称，被修改的防篡改用户表所属的database。
username	name	用户名，执行用户表修改操作的用户名。
starttime	timestamp with time zone	用户执行操作的时间。
relid	oid	用户表oid，被修改的防篡改用户表oid。
relnsp	name	模式名，被修改的防篡改用户表所属的模式名。
relname	name	用户表名，被修改的防篡改用户表名。
relhash	hash16	操作产生的表级哈希变化量。
globalhash	hash32	全局摘要，由当前行信息与前一行globalhash计算而来，将整个表串联起来，用于验证GS_GLOBAL_CHAIN数据完整性。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
txcommand	text	记录操作的SQL语句。

7.2.7 SPM 计划管理

7.2.7.1 GS_SPM_SQL

GS_SPM_SQL是SPM计划管理特性中的系统表，当前该特性在分布式中不支持。它用于存储SPM SQL相关信息的系统表，具备sysadmin权限的用户可以对该系统表进行读操作，但只有初始用户才可以对该系统表进行写操作。

表 7-24 GS_SPM_SQL 字段

名称	类型	描述
sql_namespace	oid	schema oid。
sql_hash	bigint	SPM中SQL的唯一标识。
sql_text	text	SQL的文本串。
param_num	integer	SQL执行所需参数的数量。
user	oid	登录用户的oid。
creation_time	timestamp with time zone	本记录的创建时间。

7.2.7.2 GS_SPM_PARAM

GS_SPM_PARAM是SPM计划管理特性中的系统表，当前该特性在分布式中不支持。它用于存储SQL参数相关信息的系统表，每个SQL只存储一组参数，具备sysadmin权限的用户可以对该系统表进行读操作，但只有初始用户才可以对该系统表进行写操作。

表 7-25 GS_SPM_PARAM 字段

名称	类型	描述
sql_namespace	oid	schema oid。
sql_hash	bigint	SPM中SQL的唯一标识。
position	integer	参数在SQL中的位置索引，从0开始。
datatype	integer	参数类型的oid。
datatype_string	text	参数类型的字符串形式。
value_string	text	参数值的字符串形式。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
is_null	boolean	参数值是否是NULL。
hash_value	bigint	参数值的hash value。
user	oid	本记录的创建用户。
creation_time	timestamp with time zone	本记录的创建时间。
namespace_oids	text	当前search_path中的OID列表，其中OID满足以下特征： ● 用户所拥有权限的Schema的OID。 ● 存在的Schema的OID。 ● 非本地临时表所在Schema的OID。

7.2.7.3 GS_SPM_BASELINE

GS_SPM_BASELINE是SPM计划管理特性中的系统表，当前该特性在分布式中不支持。它用于存储baseline相关信息的系统表，具备sysadmin权限的用户可以对该系统表进行读操作，但只有初始用户才可以对该系统表进行写操作。

表 7-26 GS_SPM_BASELINE 字段

名称	类型	描述
sql_namespace	oid	schema oid。
sql_hash	bigint	SPM中SQL的唯一标识。
plan_hash	bigint	当前SQL下的plan的唯一标识。
outline	text	outline文本，可固定当前计划的一组Hint。
cost	double precision	计划的总代价。
user	oid	创建baseline的用户。
status	integer	baseline的状态，可选取值有： ● 0（UNACC）：表示未接受的计划。 ● 1（ACC）：表示已经接受的计划。 ● 2（FIXED）：一种特殊的ACC状态的计划，计划的匹配优先级高于ACC状态的计划。
source	text	baseline的来源。
gplan	boolean	是否是gplan。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
creation_time	timestamp with time zone	baseline的创建时间。
last_used_time	timestamp with time zone	计划最近使用时间。
modification_time	timestamp with time zone	baseline的修改时间。
jump_intercept_cnt	bigint	当前baseline拦截计划跳变次数。
invalid	boolean	当前baseline是否无效。
namespace_oids	text	当前search_path中的OID列表，其中OID满足以下特征： ● 用户所拥有权限的Schema的OID。 ● 存在的Schema的OID。 ● 非本地临时表所在Schema的OID。

7.2.7.4 GS_SPM_EVOLUTION

GS_SPM_EVOLUTION是SPM计划管理特性中的系统表，当前该特性在分布式中不支持。它用于存储计划演进结果的系统表，具备sysadmin权限的用户可以对该系统表进行读操作，但只有初始用户才可以对该系统表进行写操作。

表 7-27 GS_SPM_EVOLUTION 字段

名称	类型	描述
sql_namespace	oid	schema oid。
sql_hash	bigint	SPM中SQL的唯一标识。
plan_hash	bigint	plan id。
better	boolean	是否是正向收益演进： ● t表示正向收益演进。 ● f表示负向收益演进。
refer_plan	bigint	报告生成主要参考的plan hash。
status	integer	演进过程中是否出现异常： ● 0表示没有发生异常。 ● 1表示有异常发生。
reason	text	演进报告的内容。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
user	oid	生成演进结果的用户。
creation_time	timestamp with time zone	创建演进结果的时间。

7.2.7.5 GS_SPM_ID_HASH_JOIN

GS_SPM_ID_HASH_JOIN是SPM计划管理特性中的系统表，当前该特性在分布式中不支持。它用于存储unique_sql_id和sql_hash关系的系统表，具备sysadmin权限的用户可以对该系统表进行读操作，但只有初始用户才可以对该系统表进行写操作。

表 7-28 GS_SPM_ID_HASH_JOIN 字段

名称	类型	描述
unique_sql_id	bigint	数据库中SQL的唯一标识。
sql_hash	bigint	SPM中SQL的唯一标识。

7.2.8 AI

7.2.8.1 GS_MODEL_WAREHOUSE

GS_MODEL_WAREHOUSE系统表用于存储AI引擎训练模型，其中包含模型、训练过程的详细描述。

表 7-29 GS_MODEL_WAREHOUSE 字段

名称	数据类型	描述
oid	oid	隐含列。
modelname	name	唯一约束。
modelowner	oid	模型拥有者的OID。
createtime	timestamp without time zone	模型创建的时间。
processedtuples	integer	训练涉及的元组数。
discardedtuples	integer	未参加训练的不合格元组数。
preprocesstime	real	数据预处理时长。
exectime	real	训练时长。

版权所有 © 华为云计算技术有限公司

名称	数据类型	描述
iterations	integer	迭代轮次。
outputtype	oid	模型输出的数据类型OID。
modeltype	text	AI算子的类型名称。
query	text	创建模型所执行的query语句。
modeldata	bytea	保存的二进制模型信息。
weight	real[]	目前只适用于GD算子模型。
hyperparametersnames	text[]	涉及的超参名称。
hyperparametersvalues	text[]	超参所对应的取值。
hyperparametersoids	oid[]	超参对应的数据类型OID。
coefnames	text[]	模型参数名称。
coefvalues	text[]	模型参数对应的取值。
coefoids	oid[]	模型参数对应的数据类型OID。
trainingscoresname	text[]	度量模型性能方法的名称。
trainingscoresvalue	real[]	度量模型性能方法的数值。
modeldescribe	text[]	模型的描述信息。

7.2.8.2 GS_OPT_MODEL

GS_OPT_MODEL是启用AiEngine执行计划时间预测功能时的数据表，记录机器学习模型的配置、训练结果、功能、对应系统函数、训练历史等相关信息。

说明

分布式场景下提供此系统表，但AI能力不可用。

7.2.8.3 GS_ABO_MODEL_STATISTIC

GS_ABO_MODEL_STATISTIC系统表用于存储ABO基于反馈基数估计模型的元信息、模型名称、算子信息等。

版权所有 © 华为云计算技术有限公司

表 7-30 GS_ABO_MODEL_STATISTIC 字段

名称	数据类型	描述
oid	oid	隐含列。
modelname	name	模型名称，唯一约束。
createtime	timestamp without time zone	模型创建/更新时间。
groupkey	oid	模型对应算子哈希。
joinkey	oid	模型对应算子连接形状哈希。
fixed_selectivity	double precision	不包含数值条件的算子固定选择率。
avg_qerror	double precision	模型平均qerror误差。
max_qerror	double precision	模型最大qerror误差。
feedback_num	integer	模型训练数据量。
status	integer	模型状态。 ● 0：表示模型无效。 ● 1：表示模型有效。 ● 2：表示模型待更新。 ● 3：表示模型处于黑名单中。 ● 4：表示该模型下次使用时将被替换为默认模型。
train_count	integer	模型总训练次数。
train_failure_count	integer	模型训练失败次数。
prediction_count	integer	模型总预测次数。
prediction_failure_count	integer	模型预测失败次数。
rtable_ids	oidvector	模型条件输入基表顺序。
attnums	int2vector	模型输入条件所涉及列的顺序。

7.2.8.4 GS_ABO_HISTORY

GS_ABO_HISTORY系统表用于存储ABO基于反馈基数估计模型使用情况统计信息等。

版权所有 © 华为云计算技术有限公司

表 7-31 GS_ABO_HISTORY 字段

名称	数据类型	描述
sql_id	bigint	sql id。
user_id	oid	用户id。
acm_sql_status	tinyint	是否在黑名单中。 ● 0：不在黑名单中。 ● 1：在黑名单中。 ● 2：不在黑名单中且正在执行。
explore_count	bigint	用于黑名单机制的计划的探索次数。
execution_time	double precision	本次查询的执行时间，单位为毫秒。
usage_count	bigint	使用反馈基数的次数。
create_time	timestamp without time zone	本条记录的创建时间。

7.2.9 审计

7.2.9.1 GS_AUDITING_POLICY

GS_AUDITING_POLICY系统表记录统一审计的主体信息，每条记录对应一个审计策略。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-32 GS_AUDITING_POLICY 字段

名称	类型	描述
oid	oid	行标识符（隐藏属性，必须明确选择）。
polname	name	策略名称，需要唯一，不可重复。
polcomments	name	策略描述字段，记录策略相关的描述信息，通过COMMENTS关键字体现。
modifydate	timestamp without time zone	策略创建或修改的最新时间戳。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
polenabled	boolean	用来表示策略启动开关。 ● t（true）：表示策略启动。 ● f（false）：表示策略没有启动。

7.2.9.2 GS_AUDITING_POLICY_ACCESS

GS_AUDITING_POLICY_ACCESS系统表记录与ACCESS数据库相关操作的统一审计信息。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-33 GS_AUDITING_POLICY_ACCESS 字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
accesstype	name	ACCESS相关类型。例如SELECT、INSERT、DELETE等。
labelname	name	资源标签名称。对应系统表GS_AUDITING_POLICY中的polname字段。
policyoid	oid	所属审计策略的OID。对应系统表12.2.9.1 GS_AUDITING_POLICY中的oid。
modifydate	timestamp without time zone	创建或修改的最新时间戳。

7.2.9.3 GS_AUDITING_POLICY_FILTERS

GS_AUDITING_POLICY_FILTERS系统表记录统一审计相关的过滤策略相关信息，每条记录对应一个审计策略。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-34 GS_AUDITING_POLICY_FILTERS 字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
filtertype	name	过滤类型。目前值仅为logical_expr。
labelname	name	名称。目前值仅为logical_expr。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
policyoid	oid	所属审计策略的OID，对应审计策略系统表GS_AUDITING_POLICY中的oid。
modifydate	timestamp without time zone	创建或修改的最新时间戳。
logicaloperator	text	过滤条件的逻辑字符串。

7.2.9.4 GS_AUDITING_POLICY_PRIVILEGES

GS_AUDITING_POLICY_PRIVILEGES系统表记录统一审计PRIVILEGES数据库相关操作信息，每条记录对应一个审计策略。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-35 GS_AUDITING_POLICY_PRIVI 字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
privilegetype	name	PRIVILEGES相关类型。例如CREATE、ALTER、DROP等。
labelname	name	资源标签名称。对应系统表GS_AUDITING_POLICY中的polname字段。
policyoid	oid	对应审计策略系统表GS_AUDITING_POLICY中的oid。
modifydate	timestamp without time zone	创建或修改的最新时间戳。

7.2.10 用户和权限管理

7.2.10.1 GS_DB_PRIVILEGE

GS_DB_PRIVILEGE系统表记录ANY权限的授予情况，每条记录对应一条授权信息。

表 7-36 GS_DB_PRIVILEGE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
roleid	oid	用户标识。
privilege_type	text	用户拥有的ANY权限，取值参考**表1-402**。
admin_option	boolean	是否具有privilege_type列记录的ANY权限的再授权权限。 • t：表示具有。 • f：表示不具有。

7.2.10.2 PG_DB_ROLE_SETTING

PG_DB_ROLE_SETTING系统表存储数据库运行时每个角色与数据绑定的配置项的默认值 。

表 7-37 PG_DB_ROLE_SETTING 字段

名称	类型	描述
setdatabase	oid	配置项所对应的数据库，如果未指定数据库，则为0。
setrole	oid	配置项所对应的角色，如果未指定角色，则为0。
setconfig	text[]	运行时配置项的默认值。

7.2.10.3 PG_DEFAULT_ACL

PG_DEFAULT_ACL系统表存储为新建对象设置的初始权限。

表 7-38 PG_DEFAULT_ACL 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
defaclrole	oid	与此权限相关的角色ID。
defaclnamespace	oid	与此权限相关的名称空间，如果没有，则为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
defaclobjtype	"char"	此权限的对象类型。 ● r：表示表或视图。 ● S：表示序列。 ● f：表示函数。 ● T：表示类型。 ● K：表示客户端主密钥。 ● k：表示列加密密钥。
defaclacl	aclitem[]	创建该类型时所拥有的访问权限。

7.2.10.4 PG_RLSPOLICY

PG_RLSPOLICY系统表存储行级访问控制策略。

表 7-39 PG_RLSPOLICY 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
polname	name	行级访问控制策略的名称。
polrelid	oid	行级访问控制策略作用的表对象oid。
polcmd	"char"	行级访问控制策略影响的SQL操作。
polpermissive	boolean	行级访问控制策略的属性。 ● t：表达式OR条件拼接。 ● f：表达式AND条件拼接。
polroles	oid[]	行级访问控制策略影响的用户oid列表，不指定表示影响所有的用户。
polqual	pg_node_tree	行级访问控制策略的表达式。

7.2.10.5 PG_SECLABEL

PG_SECLABEL系统表存储数据对象上的安全标签。

PG_SHSECLABEL的作用类似，只是它是用于在一个数据库集群内共享的数据库对象的安全标签上的。

版权所有 © 华为云计算技术有限公司

表 7-40 PG_SECLABEL 字段

名称	类型	引用	描述
objoid	oid	任意OID属性	安全标签所属的对象的OID。
classoid	oid	PG_CLASS.oid	当前对象的系统目录的OID。
objsubid	integer	-	当前对象中的列的序号。
provider	text	-	与当前标签相关的标签提供程序。
label	text	-	应用于对象的安全标签。

7.2.10.6 PG_SHSECLABEL

PG_SHSECLABEL系统表存储在共享数据库对象上的安全标签。安全标签可以用SECURITY LABEL命令操作。

查看安全标签的简单点的方法，请参阅PG_SECLABELS。

与PG_SECLABEL的作用类似，区别是它是用在单个数据库内部的对象的安全标签的。

不同于大多数的系统表，PG_SHSECLABEL在一个集群中的所有数据库中共享：每个数据库集群只有一个PG_SHSECLABEL，而不是每个数据库一个。

表 7-41 PG_SHSECLABEL 字段

名称	类型	引用	描述
objoid	oid	任意OID属性	安全标签所属的对象的OID。
classoid	oid	PG_CLASS.oid	当前对象的系统目录的OID。
provider	text	-	与当前标签相关的标签提供程序。
label	text	-	应用于当前对象的安全标签。

7.2.10.7 PG_USER_MAPPING

PG_USER_MAPPING系统表存储从本地用户到远程的映射。

需要有系统管理员权限才可以访问此系统表。普通用户可以使用视图PG_USER_MAPPINGS进行查询。

表 7-42 PG_USER_MAPPING 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
umuser	oid	PG_AUTHID.oid	被映射的本地用户的OID，如果用户映射是公共的则为0。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
umserver	oid	PG_FOREIGN_SERVER.oid	包含当前映射的外部服务器的OID。
umoptions	text[]	-	用户映射指定选项，使用“keyword=value”格式的字符串。

7.2.10.8 PG_USER_STATUS

PG_USER_STATUS系统表存储访问数据库用户的状态信息。需要有系统管理员权限才可以访问此系统表。

表 7-43 PG_USER_STATUS 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
roloid	oid	角色的标识。
failcount	integer	尝试失败次数。
locktime	timestamp with time zone	默认显示角色的创建日期，如果角色被管理员锁定，或者登录失败次数超过阈值被锁定，则显示角色被锁定的日期。
rolstatus	smallint	角色的状态。 ● 0：正常状态。 ● 1：由于登录失败次数超过阈值被锁定了一定的时间。 ● 2：被管理员锁定。 ● 3：登录失败次数超过阈值，但不锁定账户。
permspace	bigint	角色已经使用的永久表存储空间大小。
tempspace	bigint	角色已经使用的临时表存储空间大小。
passwordexpired	smallint	密码是否失效。 ● 0：密码有效。 ● 1：密码失效。
passwordlifetime	smallint	密码有效天数。不设置时默认值为0，表示密码长期有效。

版权所有 © 华为云计算技术有限公司

7.2.11 连接和认证

7.2.11.1 PG_AUTHID

PG_AUTHID系统表存储有关数据库认证标识符（角色）的信息。角色把“用户”的概念包含在内。一个用户实际上就是一个rolcanlogin标志被设置的角色。任何角色（不管rolcanlogin设置与否）都能够把其他角色作为成员。

在一个集群中只有一份pg_authid，不是每个数据库有一份。需要有系统管理员权限才可以访问此系统表。

表 7-44 PG_AUTHID 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
rolname	name	角色名称。
rolsuper	boolean	角色是否是拥有最高权限的初始系统管理员。 ● t（true）：表示是。 ● f（false）：表示不是。
rolinherit	boolean	角色是否自动继承其所属角色的权限。 ● t（true）：表示自动继承。 ● f（false）：表示不自动继承。
rolcreaterole	boolean	角色是否可以创建更多角色。 ● t（true）：表示可以。 ● f（false）：表示不可以。
rolcreatedb	boolean	角色是否可以创建数据库。 ● t（true）：表示可以。 ● f（false）：表示不可以。
rolcatupdate	boolean	角色是否可以直接更新系统表。只有usesysid=10的初始系统管理员拥有此权限。其他用户无法获得此权限。 ● t（true）：表示可以。 ● f（false）：表示不可以。
rolcanlogin	boolean	角色是否可以登录，也就是说，这个角色可以给予会话认证标识符。 ● t（true）：表示可以。 ● f（false）：表示不可以。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
rolreplication	boolean	角色是否具有复制权限。 ● t（true）：表示有。 ● f（false）：表示没有。
rolauditadmin	boolean	角色是否具有审计管理员权限。 ● t（true）：表示有。 ● f（false）：表示没有。
rolsystemadmin	boolean	角色是否具有系统管理员权限。 ● t（true）：表示有。 ● f（false）：表示没有。
rolconnlimit	integer	对于可以登录的角色，限制其最大并发连接数量。 -1 表示没有限制。
rolpassword	text	密码密文，如果没有密码，则为NULL。
rolvalidbegin	timestamp with time zone	账户的有效开始时间，如果没有开始时间，则为NULL。
rolvaliduntil	timestamp with time zone	账户的有效结束时间，如果没有结束时间，则为NULL。
rolrespool	name	用户所能够使用的resource pool。
roluseft	boolean	角色是否可以操作外表。 ● t（true）：表示可以。 ● f（false）：表示不可以。
rolparentid	oid	用户所在组的用户OID。
roltabspace	text	用户数据表的最大空间限额。
rolkind	"char"	特殊用户种类。 ● n：表示普通用户。 ● p：表示永久用户。
rolnodegroup	oid	角色所关联的node_group oid，如果没有关联node group，该值为0。
roltempspace	text	用户临时表的最大空间限额，单位 KB。
rolspillspace	text	用户执行作业时下盘数据的最大空间限额，单位 KB。
rolexcpdata	text	用户可以设置的查询规则(当前未使用)。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
rolmonitoradmin	boolean	角色是否具有监控管理员权限。 ● t（true）：表示有。 ● f（false）：表示没有。
roloperatoradmin	boolean	角色是否具有运维管理员权限。 ● t（true）：表示有。 ● f（false）：表示没有。
rolpolicyadmin	boolean	角色是否具有安全策略管理员权限。 ● t（true）：表示有。 ● f（false）：表示没有。

7.2.11.2 PG_AUTH_HISTORY

PG_AUTH_HISTORY系统表记录了角色的认证历史。需要有系统管理员权限才可以访问此系统表。

表 7-45 PG_AUTH_HISTORY 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
roloid	oid	角色标识。
passwordtime	timestamp with time zone	创建和修改密码的时间。
rolpassword	text	角色密码密文，加密方式由GUC参数password_encryption_type确定。

7.2.11.3 PG_AUTH_MEMBERS

PG_AUTH_MEMBERS系统表存储显示角色之间的成员关系。

表 7-46 PG_AUTH_MEMBERS 字段

名称	类型	描述
roleid	oid	拥有成员的角色ID。
member	oid	属于ROLEID角色的一个成员的角色ID。
grantor	oid	赋予此成员关系的角色ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
admin_option	boolean	如果MEMBER可以把ROLEID角色的成员关系赋予其他角色，则为真，不可以则为假。

7.2.12 动态脱敏

7.2.12.1 GS_MASKING_POLICY

GS_MASKING_POLICY系统表记录动态数据脱敏策略的主体信息，每条记录对应一个脱敏策略。需要有系统管理员或安全策略管理员权限才可以访问此系统表。

表 7-47 GS_MASKING_POLICY 表字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
polname	name	策略名称，唯一不可重复。
polcomments	name	策略描述字段，记录策略相关的描述信息，通过COMMENTS关键字体现。
modifydate	timestamp without time zone	策略创建或修改的最新时间戳。
polenabled	boolean	策略启动开关。 ● t（true）：表示策略启动。 ● f（false）：表示策略没有启动。

7.2.12.2 GS_MASKING_POLICY_ACTIONS

GS_MASKING_POLICY_ACTIONS系统表记录动态数据脱敏策略中相应的脱敏策略包含的脱敏行为，一个脱敏策略对应着该表的一行或多行记录。需要有系统管理员或安全策略管理员权限才可以访问此系统表。

表 7-48 GS_MASKING_POLICY_ACTIONS 表字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
actiontype	name	脱敏函数，标识脱敏策略使用的脱敏函数。
actparams	name	向脱敏函数中传递的参数信息。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
actlabelname	name	被脱敏的label名称。
policyoid	oid	该条记录所属的脱敏策略的oid，对应GS_MASKING_POLICY中的oid。
actmodifydate	timestamp without time zone	该条记录创建或修改的最新时间戳。

7.2.12.3 GS_MASKING_POLICY_FILTERS

GS_MASKING_POLICY_FILTERS系统表记录动态数据脱敏策略对应的用户过滤条件，当用户条件满足FILTER条件时，对应的脱敏策略才会生效。需要有系统管理员或安全策略管理员权限才可以访问此系统表。

表 7-49 GS_MASKING_POLICY_FILTERS 表字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
filtertype	name	过滤类型。目前值仅为logical_expr。
filterlabelname	name	过滤范围。目前值仅为logical_expr。
policyoid	oid	该条记录所属的脱敏策略的oid，对应GS_MASKING_POLICY中的oid。
modifydate	timestamp without time zone	该条用户过滤条件创建或修改的最新时间戳。
logicaloperator	text	过滤条件的波兰表达式。

7.2.13 DATABASE LINK

7.2.13.1 GS_DATABASE_LINK

GS_DATABASE_LINK系统表是用于存储DATABASE LINK信息的系统表，主要记录的是DATABASE LINK对象的详细信息。只有具备sysadmin权限的用户才可以对该系统表进行读操作。

版权所有 © 华为云计算技术有限公司

表 7-50 GS_DATABASE_LINK 字段

名称	类型	描述
oid	oid	当前DATABASE LINK对象的唯一id（隐含属性，必须明确选择）。
dlname	name	当前DATABASE LINK的名称。
dlowner	oid	当前DATABASE LINK的拥有者id，显示public则为0。
dlfdw	oid	当前DATABASE LINK的外部数据封装器的OID。
dlcreater	oid	当前DATABASE LINK创建者的id。
options	text[]	当前DATABASE LINK连接信息，使用"keyword=value"。
useroptions	text[]	当前DATABASE LINK连接远端所使用的用户信息，使用"keyword=value"。
dlacl	aclitem[]	当前DATABASE LINK访问权限。

7.2.14 物化视图

7.2.14.1 GS_MATVIEW

GS_MATVIEW系统表提供了关于数据库中每一个物化视图的信息。

表 7-51 GS_MATVIEW 字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
matviewid	oid	物化视图的oid。
mapid	oid	物化视图map表的oid，map表为物化视图关联表，与物化视图一一对应。全量物化视图不存在对应的map表，该字段为0。
ivm	boolean	物化视图的类型，t为增量物化视图，f为全量物化视图。
needrefresh	boolean	保留字段。
refreshtime	timestamp without time zone	物化视图上一次刷新时间，若未刷新则为null。仅对DN上的增量物化视图维护该字段，其余情况均为null。

版权所有 © 华为云计算技术有限公司

7.2.14.2 GS_MATVIEW_DEPENDENCY

GS_MATVIEW_DEPENDENCY系统表提供了关于数据库中每一个增量物化视图、基表和mlog表的关联信息。全量物化视图不存在与基表对应的mlog表，不会写入记录。

表 7-52 GS_MATVIEW_DEPENDENCY 字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
matviewid	oid	物化视图的oid。
relid	oid	物化视图基表的oid。
mlogid	oid	物化视图mlog表的oid，mlog表为物化视图日志表，与基表一一对应。
mxmin	integer	保留字段。

7.2.15 逻辑解码

在逻辑解码独立多版本字典中，使用的逻辑解码字典表是系统表。逻辑解码系统表均为实例级系统表，仅管理员权限用户可以查看和修改系统表。

7.2.15.1 GS_LOGICAL_ATTRDEF

GS_LOGICAL_ATTRDEF系统表记录逻辑解码需要使用的存储列的默认值。

表 7-53 GS_LOGICAL_ATTRDEF 字段

字段名称	字段类型	说明
adid	oid	属性描述符的标识符。
adrelid	oid	所属表的OID。
adnum	smallint	属性号。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	说明
dbnode	oid	数据库OID。
adbin	pg_node_tree	默认值表达式。
adgencol	"char"	是否为GENERATED列。

7.2.15.2 GS_LOGICAL_ATTRIBUTE

GS_LOGICAL_ATTRIBUTE系统表记录逻辑解码需要使用的关于表字段的信息。

表 7-54 GS_LOGICAL_ATTRIBUTE 字段

字段名称	字段类型	说明
attrelid	oid	包含该属性的RELATION OID。
attname	name	属性名称。
atttypid	oid	属性的数据类型的OID。
attlen	smallint	属性的长度。
attnum	smallint	表示属性的编号，用于唯一标识关系中的每个属性。
atthasdef	boolean	属性是否有默认值。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。
dbnode	oid	数据库OID。
attndims	integer	如果是数组类型，则为声明的维数，否则为零。
attcacheoff	integer	用于函数缓存属性在堆元组中的字节偏移。
attbyval	boolean	表示属性是否是按值传递。
attstorage	"char"	属性存储方式。
attalign	"char"	表示属性的对齐方式。
attisdropped	boolean	是否已删除（即逻辑上不可见）。
attinitdefval	bytea	存储属性的默认值表达式。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	说明
attcollation	oid	这个字段类型的PG_ATTRIBUTE中attcollation的复制。
atttypmod	integer	这个字段类型的PG_ATTRIBUTE中atttypmod的复制。

7.2.15.3 GS_LOGICAL_AUTHID

GS_LOGICAL_AUTHID系统表记录逻辑解码需要使用的用户信息。

表 7-55 GS_LOGICAL_AUTHID 字段

字段名称	字段类型	说明
rolid	oid	用户ID。
rolname	name	用户名称。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。
rolkind	"char"	用户类型。
pdbid	oid	用户对应的租户ID。

7.2.15.4 GS_LOGICAL_CLASS

GS_LOGICAL_CLASS系统表记录逻辑解码需要使用的数据库对象信息及其之间的关系。

表 7-56 GS_LOGICAL_CLASS 字段

字段名称	字段类型	说明
relname	name	表名。
relnamespace	oid	命名空间OID。
relpersistence	"char"	表的持久性（例如，'p'表示永久表）。
relkind	"char"	表的类型（例如，'r'表示关系）。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	说明
relnatts	smallint	表中的列数。
relhaspkey	bool	表是否有（或曾经有过）主键索引。
relcmprs	tinyint	表的行的压缩属性。
relreplident	"char"	Replica identity复制方式。
spacenode	oid	表空间OID。
dbnode	oid	数据库OID。
relnode	oid	表对应的物理存储文件的标识符。
createtime	timestamp with time zone	插入元组的时间戳。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
originid	integer	元组的来源标识符。
relid	oid	关系标识符（表OID）。
relowner	oid	表所有者的标识符。
reltoastrelid	oid	TOAST表OID（如果存在）。
relhasoids	boolean	表示表是否具有OID（对象标识符）列。
relam	oid	表示表使用的索引方法的标识符。
reloptions	text[]	保存表的附加选项，以文本格式存储。
relbucket	oid	表示表所属的哈希分区桶的标识符。
parttype	"char"	表示该表是否是分区表，'p'表示为分区表；'n'表示为非分区表。

7.2.15.5 GS_LOGICAL_CONSTRAINT

GS_LOGICAL_CONSTRAINT系统表记录逻辑解码需要使用的表上的检查约束、主键和唯一约束。

表 7-57 GS_LOGICAL_CONSTRAINT 字段

名称	类型	描述
conname	name	约束名称（不一定是唯一的）。
connamespace	oid	包含这个约束的名称空间的OID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
contype	"char"	• c：检查约束。 • p：主键约束。 • u：唯一约束。 • t：触发器约束。 • x：互斥约束。 • f：外键约束。 • s：聚簇约束。 • i：无效约束。
condeferrable	boolean	这个约束是否可以推迟。 • true：表示可以。 • false：表示不可以。
condeferred	boolean	缺省时这个约束是否可以推迟。 • true：表示可以。 • false：表示不可以。
convalidated	boolean	约束是否有效。目前，只有外键和CHECK约束可将其设置为false。 • true：表示有效。 • false：表示无效。
conrelid	oid	这个约束所在的表，如果不是表约束则为0。
contypid	oid	这个约束所在的域，如果不是一个域约束则为0。
conindid	oid	与约束关联的索引ID。
confrelid	oid	如果是外键，则为参考的表，否则为0。
confupdtype	"char"	外键更新动作代码。 • a：没动作。 • r：限制。 • c：级联。 • n：设置为null。 • d：设置为缺省。
confdeltype	"char"	外键删除动作代码。 • a：没动作。 • r：限制。 • c：级联。 • n：设置为null。 • d：设置为缺省。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
confmatchtype	"char"	外键匹配类型。 ● f：全部。 ● p：部分。 ● u：未指定（在f的基础上允许匹配NULL值）。
conislocal	boolean	是否是为关系创建的本地约束。 ● true：表示是。 ● false：表示不是。
coninhcount	integer	约束直接继承父表的数目。继承父表数非零时，不能删除或重命名该约束。
connoinherit	boolean	是否可以被继承。 ● true：表示可以。 ● false：表示不可以。
consoft	boolean	是否为信息约束(Informational Constraint)。 ● true：表示是。 ● false：表示不是。
conopt	boolean	是否使用信息约束优化执行计划。 ● true：表示使用。 ● false：表示不使用。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。
dbnode	oid	数据库OID。
constroid	oid	行标识符。
conkey	smallint[]	如果是表约束，则是约束控制的字段列表。
confkey	smallint[]	如果是一个外键，是参考的字段的列表。
conpfeqop	oid[]	如果是一个外键，是做PK=FK比较的相等操作符ID的列表。
conppeqop	oid[]	如果是一个外键，是做PK=PK比较的相等操作符ID的列表。
conffeqop	oid[]	如果是一个外键，是做FK=FK比较的相等操作符ID的列表。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
conexclop	oid[]	如果是一个排他约束，是列的排他操作符ID列表。
consrc	text	如果是检查约束，则是表达式的可读形式。
conincluding	smallint[]	不用做约束，但是会包含在INDEX中的属性列。

7.2.15.6 GS_LOGICAL_DATABASE

GS_LOGICAL_DATABASE系统表存储关于可用数据库的信息。

表 7-58 GS_LOGICAL_DATABASE 字段

字段名称	字段类型	说明
datname	name	数据库名称。
datid	oid	数据库OID。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。
datdba	oid	数据库所有人，通常为其创建者。
encoding	integer	数据库的字符编码方式。
datcollate	name	数据库使用的字符分类。
datctype	name	数据库使用的排序顺序。
dattablespace	oid	数据库的缺省表空间。
datcompatibility	name	数据库兼容模式，当前支持五种兼容模式：ORA、MYSQL、TD、M、PG，分别表示兼容Oracle、MySQL、TD（Teradata）、M-Compatibility和PostgreSQL。
dattimezone	name	数据库时区信息，默认为PRC时区。

7.2.15.7 GS_LOGICAL_DICTIONARY

GS_LOGICAL_DICTIONARY系统表记录逻辑解码数据字典信息。

版权所有 © 华为云计算技术有限公司

表 7-59 GS_LOGICAL_DICTIONARY 字段

字段名称	字段类型	说明
status	integer	表示逻辑解码数据字典是否有效可用。
restart_lsn	bigint	预留字段。
restart_lsn_time	timestamp with time zone	基线化完成时间。
dictionary_csn_min	bigint	使用的最小版本号。
last_clean_time	timestamp with time zone	数据字典表最后一次清理的时间。
last_clean_result	boolean	数据字典表最后一次清理的结果。
originid	integer	元组的来源标识符。
last_flush_csn	bigint	上一次元信息修改的CSN。
pdbid	oid	该行数据对应的租户ID。

7.2.15.8 GS_LOGICAL_INDEX

GS_LOGICAL_INDEX系统表存储了索引的部分信息，其他的信息存储在GS_LOGICAL_CLASS系统表中。

表 7-60 GS_LOGICAL_INDEX 字段

字段名称	字段类型	说明
indexrelid	Oid	索引的OID。
indrelid	Oid	它所引用的RELATION OID。
indnatts	smallint	索引中的总列数。
indisunique	boolean	是否是唯一索引。
indisprimary	boolean	这个索引是否为主键。
csnmin	bigint	插入元组时的 CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的 CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。
dbnode	oid	数据库OID。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	说明
indimmediate	boolean	是否立即执行唯一性约束。
indisvalid	boolean	此索引是否可用于查询。
indisready	boolean	此索引是否准备好接受插入。
indisusable	boolean	此索引是否可用于插入和选择。
indkey	int2vector	索引列的列编号。
indcollation	oidvector	排序标识符。
indpred	pg_node_tree	如果是部分索引，则为谓词的表达式树；否则为 NULL。
indisreplident	boolean	索引是否被标识为复制标识索引。

7.2.15.9 GS_LOGICAL_NAMESPACE

GS_LOGICAL_NAMESPACE系统表存储名称空间相关的信息。

表 7-61 GS_LOGICAL_NAMESPACE 字段

字段名称	字段类型	说明
nspid	oid	命名空间的OID。
nspname	name	命名空间名称。
nspowner	oid	命名空间所有者的OID。
nsptimeline	bigint	命名空间的时间线标识符。
csnmin	bigint	插入元组时的 CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的 CSN。
createtime	timestamp with time zone	插入元组的时间戳。
originid	integer	元组的来源标识符。
dbnode	oid	数据库OID。

7.2.15.10 GS_LOGICAL_PARTITION

GS_LOGICAL_PARTITION系统表存储分区表相关的信息。

版权所有 © 华为云计算技术有限公司

表 7-62 GS_LOGICAL_PARTITION 字段

字段名称	字段类型	说明
partitionid	Oid	分区表OID。
relname	name	分区表的名称。
parttype	char	分区类型。
parentid	oid	父表OID。
relfilenode	oid	分区表relfilenode。
reltoastrelid	oid	TOAST 表的OID。
reltoastidxid	oid	TOAST索引的OID。
csnmin	bigint	插入元组时的 CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的 CSN。
originid	integer	元组的来源标识符。
createtime	timestamp with time zone	插入元组的时间戳。
reltablespace	oid	表空间的OID。
dbnode	oid	数据库OID。

7.2.15.11 GS_LOGICAL_TYPE

GS_LOGICAL_TYPE系统表存储数据类型的相关信息。

表 7-63 GS_LOGICAL_TYPE 字段

字段名称	字段类型	说明
typname	name	类型名称。
typnamespace	oid	包含此类型的命名空间的OID。
typlen	smallint	该类型值所使用的字节数。
typbyval	boolean	该类型的值是按值还是按引用传递。
typtype	"char"	表示表的行类型。
typisdefined	boolean	表示类型是否被完全定义。
typreceive	regproc	二进制格式的接收函数。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	说明
typsend	regproc	发送函数。
typmodin	regproc	用于可选类型修饰符的输入函数。
typmodout	regproc	输出函数。
typid	oid	类型的OID。
dbnode	oid	数据库OID。
originid	integer	元组的来源标识符。
createtime	timestamp with time zone	插入元组的时间戳。
csnmin	bigint	插入元组时的 CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的 CSN。
typstorage	"char"	指定字段的数据存储方式，具体包含以下几个类型： ● p：普通存储（plain）。数值总是以简单方式存储。 ● e：外部存储（external）。数据实际存储在主表以外的地方（如TOAST表），主表存储的是指向该外部存储数据的引用。 ● m：主体存储（main）。该方式与“普通存储”类似，可以以内联的压缩方式存储。通常用于存储需要自定义存储方式的类型，如自定义的复合类型。 ● x：扩展存储（extended）。值可以被压缩和/或移动到一个二级关系表中。 x一般是TOAST类型存储的方式。
typelem	oid	如果typelem不为零，那么它表示gs_logical_type表中的另一行，该行定义了通过下标获取的类型（如数组）。如果typsubscript为零，则typelem应该为零。如果处理程序不需要typelem来确定下标类型，那么当typsubscript不为零时typelem也可以为零。请注意，typelem依赖关系表示了该类型中元素类型物理上的约束；因此，DDL修改类型时可能会因这种类型的存在而受到限制。
typinput	regproc	输入转换函数（文本格式）。
typoutput	regproc	输出转换函数（文本格式）。
typrelid	oid	如果是复合类型（请参见typtype），则这个字段指向GS_LOGICAL_CLASS中定义该表的行。对于复合类型，GS_LOGICAL_CLASS的类型不是一个表。对于非复合类型，该字段为零。

版权所有 © 华为云计算技术有限公司

7.2.15.12 GS_SQLAPPLY_SKIP

GS_SQLAPPLY_SKIP系统表存储逻辑回放可跳过执行的规则，当前版本暂无对应功能，仅作预留。

表 7-64 GS_SQLAPPLY_SKIP 字段

字段名称	字段类型	说明
createtime	timestamp with time zone	新建时间戳。
originid	integer	复制源标识。
skiptype	integer	SKIP类型： ● 1：SKIP OBJ ● 2：SKIP TXN ● 3：SKIP ERR
dumpcontext	"char"	转储事务到文件： ● 'a'：转储整个事务。 ● 'n'：不转储。 ● 's'：仅转储事务跳过的变更。
param1	text	参考**表7-65**、**表7-66**中的字段说明。
param2	text	参考**表7-65**、**表7-66**中的字段说明。
param3	text	参考**表7-65**中的字段说明。
param4	text	参考**表7-65**中的字段说明。
param5	text	预留字段。

表 7-65 SKIP OBJ 或 SKIP ERR 时，param1-4 的字段

字段名称	字段类型	说明
db_name	text	数据库名称。
schema_name	text	模式名称。
object_name	text	对象名称。
stmt_type	text	SQL语句类型。

版权所有 © 华为云计算技术有限公司

表 7-66 SKIP TXN 时，param1-2 字段

字段名称	字段类型	说明
csn	text	事务序列号。
xid	text	事务ID。

7.2.15.13 GS_TXN_LSN_TIME

GS_TXN_LSN_TIME系统表记录逻辑解码提交序列号、日志序列号、时间戳的对应关系。该表在分布式的环境中，只在DN节点存在数据。

表 7-67 GS_TXN_LSN_TIME 字段

字段名称	字段类型	说明
create_csn	bigint	提交序列号。
create_lsn	bigint	日志序列号。
snp_xmin	bigint	快照xmin。
create_time	timestamp with time zone	插入元组的时间戳。
snp_snapshot	text	快照序列化文本。
barrier	text	barrier信息。

说明

- create_lsn在该系统表中是以数字形式存储的，该LSN可以通过SQL语法转换为常见的LSN格式（仅供参考），即XXXXXXXX/XXXXXXXX，转换语法如下所示：

```
SELECT CONCAT(to_hex(({lsn_num}::bigint >> 32) & x'ffffffff'::bigint), '/', to_hex(({lsn_num}::bigint << 32 >> 32) & x'ffffffff'::bigint));
```

使用上述SQL语句时需将lsn_num转为自己需要的数字，以607906688为例如下所示：

```
gaussdb=# SELECT CONCAT(to_hex((607906688::bigint >> 32) & x'ffffffff'::bigint), '/', to_hex((607906688::bigint << 32 >> 32) & x'ffffffff'::bigint));
concat
------------
0/243beb80
(1 row)
```

- M兼容库需要使用以下转换语句：

```
SELECT CONCAT(hex(({lsn_num} >> 32) & 4294967295), '/', hex((({lsn_num}<< 32) >> 32) & 4294967295));
```

使用上述SQL语句时需将lsn_num转为自己需要的数字，以607906688为例如下所示：

```
gaussdb_m=# SELECT CONCAT(hex((607906688 >> 32) & 4294967295), '/', hex(((607906688 << 32) >> 32) & 4294967295));
   concat
------------
 0/243BEB80
(1 row)
```


版权所有 © 华为云计算技术有限公司

7.2.15.14 GS_LOGICAL_SET

GS_LOGICAL_SET系统表存储SET数据类型定义的元数据。

表 7-68 GS_LOGICAL_SET 字段

字段名称	字段类型	说明
setid	oid	行标识符。
settypid	oid	SET数据类型的OID。
setnum	tinyint	SET数据类型的成员数量，最大64个成员。
setsortorder	tinyint	SET数据类型定义时成员的排序位置，从0开始编号。
setlabel	text	SET数据类型的成员名称。
dbnode	oid	数据库OID。
originid	integer	元组的来源标识符。
createtime	timestamp with time zone	插入元组的时间戳。
csnmin	bigint	插入元组时的CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的CSN。
numbers	integer	用于M-Compatibility模式兼容SET和ENUM数据类型的成员数量，仅M-Compatibility兼容模式用到该字段。
sortorder	integer	用于M-Compatibility模式兼容SET和ENUM数据类型定义时成员的排序位置，仅M-Compatibility兼容模式用到该字段。

7.2.15.15 GS_LOGICAL_ENUM

GS_LOGICAL_ENUM系统表存储了每个枚举类型值和标签的记录。

表 7-69 GS_LOGICAL_ENUM 字段

字段名称	字段类型	说明
enumid	oid	行标识符。
enumtypid	oid	拥有这个枚举值的**PG_TYPE**记录的OID。
enumlabel	name	这个枚举值的文本标签。
enumsortorder	real	这个枚举值在它的枚举类型中的排序位置。
dbnode	oid	数据库OID。

版权所有 © 华为云计算技术有限公司

字段名称	字段类型	说明
originid	integer	元组的来源标识符。
createtime	timestamp with time zone	插入元组的时间戳。
csnmin	bigint	插入元组时的 CSN（提交序列号）。
csnmax	bigint	更新或删除元组时的 CSN。

7.2.15.16 GS_SQLAPPLY_HISTORY

GS_SQLAPPLY_HISTORY系统表存储了逻辑回放开始或结束时刻的解码位点。

表 7-70 GS_SQLAPPLY_HISTORY 字段

字段名称	字段类型	说明
occurtime	timestamp with time zone	逻辑回放开始或结束时间。
originid	integer	逻辑回放解码元信息标识。
opkind	char	逻辑回放的开始信息'b'或结束信息'e'。
l_xid	bigint	逻辑回放本地集群开始或结束的事务号。
l_lsn	bigint	逻辑回放本地集群开始或结束的日志位点。
l_csn	bigint	逻辑回放本地集群开始或结束的事务提交序。
r_restartlsn	bigint	远端集群逻辑解码的开始或结束日志流位点。
r_confirmlsn	bigint	远端集群逻辑解码返回事务的最早或最晚提交日志位点。
r_confirmcsn	bigint	远端集群逻辑解码返回事务的最早或最晚提交序。
uuid	text	逻辑备集群的标识。

7.2.16 基于 DSL 的规则引擎

7.2.16.1 GS_DSL_REWRITERULE

GS_DSL_REWRITERULE系统表记录DSL规则的信息，初始用户才可以修改此表。

版权所有 © 华为云计算技术有限公司

表 7-71 GS_DSL_REWRITERULE 字段

名称	类型	描述
rulename	name	规则name，全局唯一，由用户增加规则时输入填充。字母开头，不支持中文；支持字母、数字、下划线、中划线组合；不支持其他特殊字符；长度不超过64个字节。
ruledesc	name	规则实现功能的简要描述，长度不超过64个字节。
enabled	boolean	是否启用此规则。
externalrule	boolean	外置规则被设置为TRUE，内置规则设置为FALSE。
costbased	boolean	是否基于代价启用该规则，该功能当前不支持。
priority	integer	优先级，全局唯一，支持范围1-9999，数字越大优先级越低。
dbcompatibility	tinyint	此字段使用bit位映射的方式存储规则的数据库兼容模式，具体bit位标识转为十进制后如下： ● 01：标识ORA兼容。 ● 02：标识MYSQL兼容。 ● 04：标识TD兼容。 ● 08：标识PG兼容。 ● 16：标识M兼容。 ● 31：标识所有数据库兼容。
planmode	tinyint	规则支持的计划类型，可选：STREAM/REMOTEQUERY/ANY；默认值：ANY。 此字段使用bit位映射的方式存储规则支持的计划类型，具体bit位标识如下： ● 01：标识支持STREAM计划。 ● 02：标识支持REMOTEQUERY计划。 ● 03：标识支持所有计划。
createtime	timestamp with time zone	规则首次创建时间，系统自动生成。
lastupdatetime	timestamp with time zone	规则内容最后一次被更新时间，系统自动生成。
rulehash	text	规则文本的HASH值，用于检查文本内容唯一性。
ruletext	text	规则文本，用于存储外置DSL规则的定义。对于内置DSL规则，该字段为空。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
conflictrule	text	冲突规则，若检测到已应用的冲突规则，则跳过该规则的使用。 示例：当该值为(rulea,ruleb,rulec)时，表示若已进行rulea、ruleb或rulec的改写，则跳过此规则的改写。
rulegroup	name	规则分组，全局唯一，由用户添加规则时输入填充。需以字母开头，不支持中文；支持字母、数字、下划线、中划线组合；不支持其他特殊字符；长度不超过64个字节。该字段仅用于内置规则，外置规则暂不可用。
updateversion	integer	规则更新版本号，根据版本号控制规则是否生效。

7.2.17 HTAP

7.2.17.1 GS_IMCV

GS_IMCV系统表提供了所有IMCV表的元信息。

表 7-72 GS_IMCV 字段

名称	类型	描述
reloid	oid	表id。
relname	name	表名。
dbname	name	数据库名。
username	name	IMCV表的owner。
parentoid	oid	父表id。如果是分区，则表示父表id，否则显示0。
imcvispart	boolean	是否是分区表。
imcvnattr	smallint	需要加载的列数量。
imcvkey	int2vector	加载的列数组。
priority	smallint	加载优先级。 ● '1'：PRIORITY_NONE无优先级。 ● '2'：PRIORITY_LOW低优先级。 ● '3'：PRIORITY_HIGH高优先级。
childoids	oidvector_extend	如果是分区表，记录该表的子分区的id列表。如果不是，则为空。
dboid	oid	数据库id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
rootoid	oid	如果是普通表，则为该表的id，如果是分区表，则为该分区表基表的id。
schemaname	name	schema的名字。

7.2.18 其他系统表

7.2.18.1 GS_ASP

GS_ASP显示被持久化的ACTIVE SESSION PROFILE样本。该系统表只能在系统库中查询。

表 7-73 GS_ASP 字段

名称	类型	描述
sampleid	bigint	采样ID。
sample_time	timestamp with time zone	采样的时间。
need_flush_sample	boolean	该样本是否需要刷新到磁盘。 ● t（true）：表示需要。 ● f（false）：表示不需要。
databaseid	oid	数据库ID。
thread_id	bigint	线程的ID。
sessionid	bigint	会话的ID。
start_time	timestamp with time zone	会话的启动时间。
event	text	具体的事件名称。内核中关键的事件信息请参见**表7-470**、**表7-471**、**表7-472**和**表7-473**。关于每种事务锁对业务的影响程度，请参见**LOCK**语法小节的详细描述。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	streaming线程的父线程。
tlevel	integer	streaming线程的层级。与执行计划的层级(ID)相对应。
smpid	integer	smp执行模式下并行线程的并行编号。
userid	oid	session用户的ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
application_name	text	应用的名字。
client_addr	inet	client端的地址。
client_hostname	text	client端的名字。
client_port	integer	客户端用于与后端通讯的TCP端口号。
query_id	bigint	唯一SQL id。
unique_query_id	bigint	归一化SQL id。
user_id	oid	unique query的key中的user_id。
cn_id	integer	表示该unique sql来自哪一个CN节点。unique query的key中的cn_id。
unique_query	text	规范化后的Unique SQL文本串。
locktag	text	会话等待锁信息，可通过locktag_decode解析。
lockmode	text	会话等待锁模式： ● LW_EXCLUSIVE：排他锁。 ● LW_SHARED：共享锁。 ● LW_WAIT_UNTIL_FREE：等待LW_EXCLUSIVE可用。
block_sessionid	bigint	如果会话正在等待锁，阻塞该会话获取锁的会话标识。
wait_status	text	描述event列的更多详细信息。
global_sessionid	text	全局会话ID，结构为：nodeid:sessionid#seqid。 ● nodeid：表示下发节点的ID。 ● sessionid：表示下发节点，会话的ID。 ● seqid：递增序列号，在下发节点上该ID为0。同一个会话下发到其他节点的会话，该字段的nodeid和sessionid相同。
xact_start_time	timestamp with time zone	事务开始时间。
query_start_time	timestamp with time zone	语句开始执行时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
state	text	当前语句状态。 可能取值为： ● active：后台正在执行一个查询。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。
event_start_time	timestamp with time zone	wait event的开始时间。
current_xid	xid	当前事务的ID。
top_xid	xid	当前执行事务的顶层事务ID。
instance_name	text	指定该条记录所属CN或DN。
parent_unique_sql_id	bigint	调用本条语句的父语句sql_id。
control_status	text	当进行异常SQL CPU资源管控时为under_cpu_throttle，否则为normal。
top_unique_query_id	bigint	该字段展示存储过程顶层SQL的unique_query_id。
top_unique_query	text	该字段展示存储过程顶层SQL的unique_query语句。
sql_hash	bigint	当前SQL语句对应的hash值，为0。
plan_hash	bigint	当前SQL语句执行计划对应的hash值，为0。

7.2.18.2 GS_COPY_ERROR_OLD

GS_COPY_ERROR_OLD系统表用于捕捉COPY FROM过程的相关错误行，以便后续查阅。

此系统表仅用作前向兼容场景，即从旧版本升级上来的数据库中，此表中可能会存在数据。在此版本上，该系统表已失效，COPY语句不会再向此表中记录任何数据。

版权所有 © 华为云计算技术有限公司

表 7-74 GS_COPY_ERROR_OLD 字段

名称	类型	描述
relname	character varying	表名称。以模式名.表名形式显示。
begintime	timestamp with time zone	出现数据格式错误的时间。
filename	character varying	出现数据格式错误的数据源文件名。
lineno	bigint	在数据源文件中，出现数据格式错误的行号。
rawrecord	text	在数据源文件中，出现数据格式错误的原始记录。
detail	text	详细错误信息。

说明

- pg_catalog.gs_copy_error_old表当前没有自动清理机制，需要管理员用户手动执行TRUNCATE语句进行清理。
- pg_catalog.gs_copy_error_old表当前没有行级访问控制。

7.2.18.3 GS_COPY_SUMMARY_LOG

分布式部署环境，此系统表使用不生效，即不记录数据。

表 7-75 GS_COPY_SUMMARY_LOG 字段

名称	类型	描述
relname	text	本次导入的目标表名。
begintime	timestamp with time zone	导入任务开始时间。
endtime	timestamp with time zone	导入任务结束时间。
id	bigint	本次导入的事务id。
pid	bigint	本次导入工作线程id。
readrows	bigint	本次导入任务共计读取的数据行数。
skiprows	bigint	本次导入任务共计跳过的数据行数。
loadrows	bigint	本次导入任务成功导入的数据行数。
errorrows	bigint	本次导入任务的错误数据行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
whenrows	bigint	本次导入任务违反when过滤条件的数据行数。
allnullrows	bigint	全部字段都为空的数据行数。
detail	text	本次导入任务的导入情况总结。主要包括成功导入行数、错误数据行数、违反when条件行数和字段全空行数等。
custom_id	text	关联的COPY语法中copy_custom_id输入的值，表示本次导入的自定义标识信息。

说明

- pg_catalog.gs_copy_summary_log表当前没有自动清理机制，需要管理员用户手动执行TRUNCATE语句进行清理。
- pg_catalog.gs_copy_summary_log表当前没有行级访问控制。

7.2.18.4 GS_COPY_ERROR_LOG

GS_COPY_ERROR_LOG系统表用于捕捉COPY FROM过程的相关错误行，以便后续查阅。

- 开启三权分立（即GUC参数enableSeparationOfDuty=on）时，使用者可以是数据库普通用户或管理员用户。
- 不开启三权分立（即GUC参数enableSeparationOfDuty=off）时，使用者可以是数据库普通用户或管理员用户。当使用者为普通用户的时候，需要管理员用户对普通用户赋权（该赋权动作需要**allow_create_sysobject**参数为on）。管理员用户可以直接使用。

表 7-76 GS_COPY_ERROR_LOG 字段说明

名称	类型	描述
relname	text	表名称。以模式名/表名形式显示。
begintime	timestamp with time zone	出现数据格式错误的时间。
filename	text	出现数据格式错误的数据源文件名。
lineno	bigint	在数据源文件中，出现数据格式错误的行号。
rawrecord	text	在数据源文件中，出现数据格式错误的原始记录。
detail	text	详细错误信息。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
custom_id	text	关联的COPY语法中copy_custom_id输入的值，表示本次导入的自定义标识信息。

说明

- pg_catalog.gs_copy_error_log表当前没有自动清理机制，需要管理员用户手动执行TRUNCATE语句进行清理。
- pg_catalog.gs_copy_error_log表当前没有行级访问控制。
- 缩容后，被缩容DN节点上pg_catalog.gs_copy_error_log表中的数据会丢失。缩容前请先确认表内数据是否需要，如有需要则提前进行备份。
- 三权分立机制下，错误日志信息不会记录至pg_catalog.pgxc_copy_error_log和pg_catalog.gs_copy_error_log。若用户load_user需记录错误信息，需通过以下语句在其同名Schema下创建：

  ```
  CREATE TABLE load_user.pgxc_copy_error_log (relname varchar, begintime timestamptz, filename varchar, lineno int8, rawrecord text, detail text);
  CREATE INDEX copy_error_log_relname_idx ON load_user.pgxc_copy_error_log(relname);
  ```

  （可选）在表上添加custom_id列：

  ```
  ALTER TABLE load_user.pgxc_copy_error_log ADD COLUMN custom_id varchar(64);
  ```

7.2.18.5 GS_DEP_SOURCE

GS_DEP_SOURCE系统表记录元信息和对象的部分编译结果。仅系统管理员可以查看。

表 7-77 GS_DEP_SOURCE 字段

名称	类型	引用	描述
object_oid	oid	PG_OBJECT.object_oid	对象标识符。
object_type	"char"	-	对象类型：v表示视图。
source	text	-	定义的元信息。
compile	pg_node_tree	-	部分编译结果。

版权所有 © 华为云计算技术有限公司

说明

针对source字段，将视图定义语句中SELECT *的星号展开以及视图创建时对象使用的schema补充写入，示例如下：

```
gaussdb=# CREATE USER user01 WITH SYSADMIN PASSWORD 'xxxx';
gaussdb=# SET ROLE user01 PASSWORD 'xxxx';
-- 在user01 SCHEMA下创建表t1。
gaussdb=# CREATE TABLE t1(c1 INT, c2 INT);
-- 在user01 SCHEMA下创建视图v1，v1查询的表为user01.t1。
gaussdb=# CREATE VIEW v1 AS SELECT * FROM t1;
-- 查询source字段。
gaussdb=# SELECT source FROM gs_dep_source WHERE object_oid = xxx;
 source
--------
CREATE OR REPLACE VIEW user01.v1(c1, c2) AS SELECT t1.c1, t1.c2 FROM user01.t1;
(1 row)
```

7.2.18.6 GS_DEPENDENCIES

GS_DEPENDENCIES系统表记录对象的依赖项信息，和**GS_DEPENDENCIES_OBJ**表是一个一对多的关系。

表 7-78 GS_DEPENDENCIES 字段

名称	类型	描述
schemaname	name	名称空间的名称。
packagename	name	package的名称。
refobjpos	integer	被依赖体引用的位置。 • 1：类型。 • 2：包头。 • 4：函数头。 • 8：函数体。 • 16：包体。 • 32：视图。
refobjoid	oid	被依赖体的oid。
objectname	text	依赖体名称。

7.2.18.7 GS_DEPENDENCIES_OBJ

GS_DEPENDENCIES_OBJ系统表记录对象的被依赖项详细信息。

表 7-79 GS_DEPENDENCIES_OBJ 字段

名称	类型	描述
schemaname	name	名称空间的名称。
packagename	name	package的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
type	integer	被依赖体的类型。 ● 1：未知类型。 ● 2：变量。 ● 3：类型。 ● 4：函数。 ● 5：视图。 ● 6：函数头。
name	text	被依赖体名称。
objnode	pg_node_tree	被依赖体的详细信息。

7.2.18.8 GS_GLOBAL_CONFIG

GS_GLOBAL_CONFIG记录了集群初始化时，用户指定的参数值。除此之外，还存放了用户设置的弱口令、集群是否处于扩容状态（redis_bucket_expansion），支持数据库初始用户通过ALTER和DROP语法对系统表中的参数进行写入、修改和删除。此系统表默认只有初始用户、系统管理员和安全管理员可以访问，其他用户默认无权访问。

表 7-80 GS_GLOBAL_CONFIG 字段

名称	类型	描述
name	name	集群初始化时系统内置的指定参数名称、弱口令名称、扩容重分布参数名称、或用户需要使用的参数。
value	text	集群初始化时系统内置的指定参数值、弱口令、扩容重分布参数、或用户需要使用的参数值。

须知

不建议用户写入、修改和删除“redis_bucket_expansion”参数。该参数名称被占用，在扩容期间用来表示集群处于扩容状态。

7.2.18.9 GS_JOB_ATTRIBUTE

GS_JOB_ATTRIBUTE系统表提供了DBE_SCHEDULER定时任务的相关属性信息，其中包括定时任务、定时任务类、证书、授权、程序和调度的基本属性。新安装数据库集群普通用户无权限访问。

版权所有 © 华为云计算技术有限公司

表 7-81 GS_JOB_ATTRIBUTE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
job_name	text	定时任务、定时任务类、证书、程序、调度的名字和授权的用户名。
attribute_name	text	定时任务、定时任务类、证书、程序、调度的属性名和授权的内容。
attribute_value	text	定时任务、定时任务类、证书、程序和调度的属性值。

7.2.18.10 GS_JOB_ARGUMENT

GS_JOB_ARGUMENT系统表提供了DBE_SCHEDULER定时任务和程序的参数属性。

表 7-82 GS_JOB_ARGUMENT 字段

名称	类型	描述
oid	oid	行标识符（隐含字段）。
argument_position	integer	定时任务或程序的参数位置。
argument_type	name	定时任务或程序的参数类型。
job_name	text	定时任务或程序名。
argument_name	text	定时任务或程序的参数名（定时任务继承了程序的参数名，所以为空）。
argument_value	text	定时任务的参数值（程序本身无法绑定值）。
default_value	text	程序的参数默认值。

7.2.18.11 GS_PACKAGE

GS_PACKAGE系统表记录PACKAGE内的信息。

表 7-83 GS_PACKAGE 字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
pkgnamespace	oid	package所属schema。
pkgowner	oid	package的所属者。
pkgname	name	package的名字。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pkgspecsrc	text	package specification的内容。
pkgbodydeclsrc	text	package body的内容。
pkgbodyinitsrc	text	package init的内容。
pkgacl	aclitem[]	访问权限。
pkgsecdef	boolean	package是否拥有定义者权限。

7.2.18.12 GS_PLAN_TRACE

GS_PLAN_TRACE系统表是用于存储plan trace的系统表，主要记录的是DML语句生成计划过程的详情，只有具有sysadmin权限的用户才可以使用该系统表。分布式场景下不支持plan trace特性，所以在分布式下该视图中无数据。

表 7-84 GS_PLAN_TRACE 字段

名称	类型	描述
query_id	text	当前请求的唯一id。
query	text	当前请求的SQL语句，该字段大小不会超过系统参数track_activity_query_size指定的大小。
unique_sql_id	bigint	当前请求SQL的唯一id。
plan	text	当前请求SQL对应的查询计划文本，该字段大小不会超过10K。
plan_trace	text	当前请求SQL对应的查询计划生成过程的明细，该字段大小不会超过300M。
owner	oid	当前请求SQL用户的OID。
modifydate	timestamp with time zone	当前plan trace的更新时间（当前指的是plan trace创建时间）。

7.2.18.13 GS_POLICY_LABEL

GS_POLICY_LABEL系统表记录资源标签配置信息，一个资源标签对应着一条或多条记录，每条记录标记了数据库资源所属的资源标签。需要有系统管理员或安全策略管理员权限才可以访问此系统表。

FQDN（Fully Qualified Domain Name）标识了数据库资源所属的绝对路径。

版权所有 © 华为云计算技术有限公司

表 7-85 GS_POLICY_LABEL 表字段

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
labelname	name	资源标签名称。
labeltype	name	资源标签类型，目前仅为RESOURCE。
fqdnnamespace	oid	被标识的数据库资源所属的namespace oid。
fqdnid	oid	被标识的数据库资源的oid，若数据库资源为列，则该列为所属表的oid。
relcolumn	name	列名，若被标识的数据库资源为列，该列指出列名，否则该列为空。
fqdntype	name	被标识的数据库资源的类型名称，例如schema、table、column、view等。

7.2.18.14 GS_RECYCLEBIN

gs_recyclebin描述了闪回特性回收站对象的详细信息。

表 7-86 gs_recyclebin 字段

名称	类型	描述
oid	oid	系统列。
rcybaseid	oid	基表对象id，引用gs_recyclebin.oid。
rcydbid	oid	当前对象所属数据库OID。
rcyrelid	oid	当前对象OID。
rcyname	name	回收站对象名称，格式“BIN$unique_id$oid$0”，其中unique_id为最多16字符唯一标识，oid为对象标识符。
rcyoriginname	name	原始对象名称。
rcyoperation	"char"	操作类型： ● d表示drop ● t表示truncate

版权所有 © 华为云计算技术有限公司

名称	类型	描述
rcytype	integer	对象类型： • 0表示table。 • 1表示index。 • 2表示toast table。 • 3表示toast index。 • 4表示sequence，指serial、bigserial、smallserial、largeserial类型自动关联的序列对象。 • 5表示partition。 • 6表示global index。 • 7表示物化视图。 • 8表示global secondary index。
rcyrecyclecsn	bigint	对象drop、truncate时csn。
rcyrecycletime	timestamp with time zone	对象drop、truncate时间。
rcycreatecsn	bigint	对象创建时csn。
rcychangecsn	bigint	对象定义改变的csn。
rcynamespace	oid	包含这个关系的名字空间的OID。
rcyowner	oid	关系所有者。
rcytablespace	oid	这个关系存储所在的表空间。如果为0，则意味着使用该数据库的缺省表空间。如果关系在磁盘上没有文件，则这个字段没有什么意义。
rcyrelfilenode	oid	回收站对象在磁盘上的文件名称，如果没有则为0，用于TRUNCATE对象恢复时物理文件还原。
rcycanrestore	boolean	是否可以被单独闪回。
rcycanpurge	boolean	是否可以被单独purge。
rcyfrozenxid	xid32	该表中所有在这个之前的事务id已经被一个固定的（"frozen"）事务id替换。
rcyfrozenxid64	xid	该表中所有在这个之前的事务id已经被一个固定的（"frozen"）事务id替换。
rcybucket	oid	12.2.15.42 PG_HASHBUCKET中的桶信息。

7.2.18.15 GS_SECURITY_LABEL

GS_SECURITY_LABEL是共享系统表，记录安全标签的相关信息，每条记录对应一个安全标签。

版权所有 © 华为云计算技术有限公司

表 7-87 GS_SECURITY_LABEL 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
label_name	name	安全标签名称。
label_content	text	安全标签内容。

7.2.18.16 GS_SQL_PATCH

GS_SQL_PATCH系统表存储当前节点上的SQL_PATCH的状态信息。

表 7-88 GS_SQL_PATCH 字段

名称	类型	描述
patch_name	name	PATCH名称。
unique_sql_id	bigint	查询全局唯一id。
owner	oid	PATCH的创建用户id。
enable	boolean	PATCH是否生效。
status	"char"	标识SQL PATCH是基于Unique SQL ID还是SQL HASH创建的，'d'代表Unique SQL ID，'h'代表SQL HASH。
abort	boolean	是否是AbortHint。
hint_string	text	Hint文本。
hint_node	pg_node_tree	Hint解析&序列化的结果。
original_query	text	原始语句（预留字段）。
patched_query	text	PATCH之后的语句（预留字段）。
original_query_tree	pg_node_tree	原始语句的解析结果（预留字段）。
patched_query_tree	pg_node_tree	PATCH之后语句的解析结果（预留字段）。
description	text	PATCH的备注。
parent_unique_sql_id	bigint	PATCH生效的SQL外层语句的全局唯一id，存储过程外的语句该值为0，存储过程内的语句该值为调用该存储过程语句的全局唯一id。

版权所有 © 华为云计算技术有限公司

7.2.18.17 GS_STATISTIC_EXT_HISTORY

GS_STATISTIC_EXT_HISTORY是多列历史统计信息管理表，存储有关该数据库中表的历史扩展统计数据，包括多列统计数据和表达式统计数据（后续支持）。

表 7-89 GS_STATISTIC_EXT_HISTORY 字段

名称	类型	描述
starelid	oid	所描述字段所属的表或者索引。
starelkind	"char"	所属对象的类型，'c'表示表，'p'表示分区。
stainherit	boolean	是否统计有继承关系的对象。
statimestamp	timestamp with time zone	该条统计信息收集的时间。
stanullfrac	real	所描述字段中为NULL的记录的比率。
stawidth	integer	所描述字段非NULL记录的平均存储宽度，以字节计。
stadistinct	real	标识全局统计信息中数据库节点上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
stadndistinct	real	标识DN1上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadndistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
standvfunc	"char"	该条统计信息计算NDV时采用的算法。 ● d：采用原来的DUJ1算法估算。 ● c：采用C19算法估算。
staorigin	"char"	该条统计信息的收集方式来源。 ● a：标识由autoanalyze触发收集。 ● m：标识由手动analyze触发收集。 ● g：标识由插入大量数据触发了gsstat线程做analyze收集。
stakindN	smallint	一个编码，表示这种类型的统计存储在pg_statistic行的第N个“槽位”。 N的取值范围：1～5。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
staopN	oid	一个用于生成这些存储在第N个“槽位”的统计信息的操作符。比如，一个柱面图槽位会显示<操作符，该操作符定义了该数据的排序顺序。 N的取值范围：1～5。
stakey	int2vector	所描述的字段编号的数组。
stanumbersN	real[]	第N个“槽位”的相关类型的数值类型统计，如果该槽位和数值类型没有关系，则返回值NULL。 N的取值范围：1～5。
stavaluesN	anyarray	第N个“槽位”类型的字段数据值，如果该槽位类型不存储任何数据值，则返回值NULL。每个数组的元素值实际上都是指定字段的数据类型，因此，除了把这些字段的类型定义成anyarray之外，没有更好的办法。 N的取值范围：1～5。
staexprs	pg_node_tree	扩展统计信息对应的表达式。
stasource	"char"	扩展统计信息的来源： ● 'a'：表示自动创建，由GUC参数auto_statistic_ext_columns控制。 ● 'm'：表示手动创建，用户通过 analyze tablename ((column list)) 或者 alter table tablename add statistics ((column list)) 来创建。
stastatus	"char"	扩展统计信息的状态。 ● 'a'：表示活跃可用。 ● 'd'：表示被禁用，相关信息不被收集，优化器在生成计划的时候也不使用。用户可以使用语法 alter table tablename disable/enable statistics((column list)) 来修改扩展统计信息的状态。
staextname	name	多列统计信息多列组的别名。

7.2.18.18 GS_STATISTIC_HISTORY

GS_STATISTIC_HISTORY是单列历史统计信息管理表，存储有关该数据库中表和索引列的历史统计数据。

表 7-90 GS_STATISTIC_HISTORY 字段

名称	类型	描述
starelid	oid	所描述字段所属的表或者索引。
starelkind	"char"	所属对象的类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
staattnum	smallint	所描述字段在表中的编号，从1开始。
stainherit	boolean	是否统计有继承关系的对象。
statimestamp	timestamp with time zone	该条统计信息收集的时间。
stanullfrac	real	所描述字段中为NULL的记录的比率。
stawidth	integer	所描述字段非NULL记录的平均存储宽度，以字节计。
stadistinct	real	标识全局统计信息中数据库节点上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
standvfunc	"char"	该条统计信息计算NDV时采用的算法。 ● d：采用原来的DUJ1算法估算。 ● c：采用C19算法估算。 ● u：表示计算ndv值的算法在当前CN节点无法查询获得，可以从其他CN节点的历史表查询相同时间戳来获取。
staorigin	"char"	该条统计信息的收集方式来源。 ● a：标识由autoanalyze触发收集。 ● m：标识由手动analyze触发收集。 ● g：标识由插入大量数据触发了gsstat线程做analyze收集。
stakindN	smallint	一个编码，表示这种类型的统计存储在pg_statistic行的第N个“槽位”。 N的取值范围：1～5。
staopN	oid	一个用于生成这些存储在第N个“槽位”的统计信息的操作符。比如，一个柱面图槽位会显示<操作符，该操作符定义了该数据的排序顺序。 N的取值范围：1～5。
stanumbersN	real[]	第N个“槽位”的相关类型的数值类型统计，如果该槽位和数值类型没有关系，则返回值NULL。 N的取值范围：1～5。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stavaluesN	anyarray	第N个“槽位”类型的字段数据值，如果该槽位类型不存储任何数据值，则返回值NULL。每个数组的元素值实际上都是指定字段的数据类型，因此，除了把这些字段的类型定义成anyarray之外，没有更好的办法。 N的取值范围：1～5。
stadndistinct	real	标识DN1上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadndistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
staextinfo	text	统计信息的扩展信息。预留字段。

7.2.18.19 GS_SYSTEM_OBJECTS_VERSION

GS_SYSTEM_OBJECTS_VERSION记录了系统对象的版本号信息，用于在升级中判断变更及新增的系统对象，从而更新系统对象定义或创建系统对象。

此系统表中写入的系统对象类型包含视图、函数、表、索引、类型、规则及类型转换关系CAST。

表 7-91 GS_SYSTEM_OBJECTS_VERSION 字段

名称	类型	描述
obj_id	oid	系统对象的标识符，不同类型系统对象的对应关系如下： 视图、表、索引在PG_CLASS内的唯一标识，类型、函数、规则、类型转换关系CAST分别在PG_TYPE、PG_PROC、PG_REWRITE、PG_CAST内的唯一标识。
obj_type	char	系统对象的类型。 ● v：标识视图。 ● f：标识函数。 ● r：标识普通表。 ● i：标识索引。 ● c：标识复合类型。 ● d：标识域类型。 ● R：标识规则。 ● C：标识类型转换。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
obj_version	int8	系统对象的版本号。 0：初始版本号。 -1：已删除对象的版本号。 其他版本号为后续对象变更的版本号。

7.2.18.20 GS_TABLESTATS_HISTORY

GS_TABLESTATS_HISTORY是表、索引、分区级别的历史统计信息管理表，存储有关该数据库中表、索引、分区的历史统计数据。

表 7-92 GS_TABLESTATS_HISTORY 字段

名称	类型	描述
relid	oid	表、索引、分区在12.2.15.28 PG_CLASS、12.2.1.1 PG_PARTITION内的唯一标识。
relname	name	表、索引、分区的名称。
relnamespace	oid	包含此对象的命名空间的oid。
relkind	"char"	对象类型。 ● r：标识普通表。 ● I：标识表级索引。 ● i：标识分区索引。 ● p：标识一级分区。 ● s：标识二级分区。
reltimestamp	timestamp with time zone	该条统计信息收集的时间。
relpages	double precision	以页为单位的此表在磁盘上的大小，它只是优化器用的一个近似值。
reltuples	double precision	表中行的数目，只是优化器使用的一个估计值。
relallvisible	integer	被标识为全可见的表中的页的数量。

7.2.18.21 GS_TS_DICT_DEFINITION

GS_TS_DICT_DEFINITION是自定义词典特性存储词典内容的系统表，用于自定义词典的创建和使用，当前版本仅支持查看系统表。

版权所有 © 华为云计算技术有限公司

表 7-93 GS_TS_DICT_DEFINITION 字段

名称	类型	描述
dict_oid	oid	词典内容归属词典的oid。
definition_type	"char"	词典内容类型。
definition	text	词典内容定义。

7.2.18.22 GS_TXN_SNAPSHOT

GS_TXN_SNAPSHOT是“时间戳-CSN”映射表，周期性采样，并维护适当的时间范围，用于估算范围内的时间戳对应的CSN值。

表 7-94 GS_TXN_SNAPSHOT 字段

名称	类型	描述
snptime	timestamp with time zone	快照捕获时间。
snpxmin	bigint	快照xmin。
snpcsn	bigint	快照csn。
snpsnapshot	text	快照序列化文本。

7.2.18.23 GS_UID

GS_UID系统表存储了数据库中使用hasuids属性表的唯一标识元信息。

表 7-95 GS_UID 字段

名称	类型	描述
relid	oid	表的oid信息。
uid_backup	bigint	当前可以为表分配唯一标识的最大值。

7.2.18.24 GS_WORKLOAD_RULE

GS_WORKLOAD_RULE系统表存储与SQL限流规则相关的信息。该系统表没有权限限制，所有用户可查询。

版权所有 © 华为云计算技术有限公司

表 7-96 GS_WORKLOAD_RULE 字段

名称	类型	描述
rule_id	bigint	限流规则标识列，系统自动生成。
rule_name	name	限流规则的名称，用于检索限流规则，不保证唯一性，可以为NULL。
databases	name[]	限流规则作用的数据库列表，为NULL表示所有库生效。
max_workload	bigint	限制规则设置的最大并发数。
is_valid	boolean	限流规则是否生效，超时的限流规则会设为false。
start_time	timestamp with time zone	限流规则开始的时间，为NULL表示从现在开始生效。
end_time	timestamp with time zone	限流规则结束的时间，为NULL表示一直生效。
rule_type	text	限流规则类型，当前仅支持：“sqlid”、“select”、“insert”、“update”、“delete”、“merge”、“resource”，“client”，其他的为非法值。
option_val	text[]	限流规则的参数值，包括：sqlid，关键字列表，资源限制情况，客户端IP、ApplicationName。 详细请参见**gs_add_workload_rule**接口说明。
node_names	text[]	预留字段，限流规则生效的节点名称列表，当前不生效。
user_names	text[]	预留字段，限流规则生效的用户名称列表，当前不生效。

须知

针对客户端限流规则，在option_val字段中，ApplicationName将保留并显示用户原始输入的完整内容。在执行客户端限流规则匹配时，ApplicationName的有效长度上限为64字节（含63位可用字节加结束符）。若原始输入长度超过63字节，将自动截断并仅使用前63字节进行规则匹配。

7.2.18.25 PG_AGGREGATE

PG_AGGREGATE系统表存储与聚集函数有关的信息。PG_AGGREGATE里的每条记录都是一条pg_proc里面的记录的扩展。PG_PROC记录承载该聚集的名称、输入和输出数据类型，以及其它一些和普通函数类似的信息。

版权所有 © 华为云计算技术有限公司

表 7-97 PG_AGGREGATE 字段

名称	类型	引用	描述
aggfnoid	regproc	PG_PROC.proname	此聚集函数的PG_PROC proname。
aggtransfn	regproc	PG_PROC.proname	转换函数。
aggcollectfn	regproc	PG_PROC.proname	收集函数。
aggfinalfn	regproc	PG_PROC.proname	最终处理函数（如果没有则为零）。
aggsortop	oid	PG_OPERATOR.oid	关联排序操作符（如果没有则为零）。
aggtranstype	oid	PG_TYPE.oid	此聚集函数的内部转换（状态）数据的数据类型。 可能取值及其含义见于pg_type.h中诸type定义，主要分为多态（isPolymorphicType）和非多态两类。
agginitval	text	-	转换状态的初始值。这是一个文本数据域，它包含初始值的外部字符串表现形式。如果数据域是null，则转换状态值从null开始。
agginitcollect	text	-	收集状态的初始值。这是一个文本数据域，它包含初始值的外部字符串表现形式。如果数据域是null，则收集状态值从null开始。
aggkind	"char"	-	此聚集函数类型： • 'n'：表示Normal Agg • 'o'：表示Ordered Set Agg
aggnumdirectargs	smallint	-	Ordered Set Agg类型聚集函数的直接参数（非聚集相关参数）数量。对Normal Agg类型聚集函数，该值为0。
agginitfn	regproc	PG_PROC.proname	初始化函数。

7.2.18.26 PG_AM

PG_AM系统表存储有关索引访问方法的信息。系统支持的每种索引访问方法都有一行。

版权所有 © 华为云计算技术有限公司

表 7-98 PG_AM 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含属性，必须明确选择）。
amname	name	-	访问方法的名称。
amstrategies	smallint	-	访问方法的操作符策略个数，或者如果访问方法没有一个固定的操作符策略集则为0。
amsupport	smallint	-	访问方法的支持过程个数。
amcanorder	boolean	-	这种访问方式是否支持通过索引字段值的命令扫描排序。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amcanorderbyop	boolean	-	这种访问方式是否支持通过索引字段上操作符的结果的命令扫描排序。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amcanbackward	boolean	-	访问方式是否支持向后扫描。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amcanunique	boolean	-	访问方式是否支持唯一索引。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amcanmulticol	boolean	-	访问方式是否支持多字段索引。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amoptionalkey	boolean	-	访问方式是否支持第一个索引字段上没有任何约束的扫描。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amsearcharray	boolean	-	访问方式是否支持ScalarArrayOpExpr搜索。 ● t（true）：表示支持。 ● f（false）：表示不支持。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
amsearchnulls	boolean	-	访问方式是否支持IS NULL/NOT NULL搜索。 ● t（true）：表示支持。 ● f（false）：表示不支持。
amstorage	boolean	-	是否允许索引存储的数据类型与列的数据类型不同。 ● t（true）：表示允许。 ● f（false）：表示不允许。
amclusterable	boolean	-	是否允许在一个这种类型的索引上聚簇。 ● t（true）：表示允许。 ● f（false）：表示不允许。
ampredlocks	boolean	-	是否允许这种类型的一个索引管理细粒度的谓词锁定。 ● t（true）：表示允许。 ● f（false）：表示不允许
amkeytype	oid	PG_TYPE.oid	存储在索引里数据的类型，如果不是一个固定的类型则为0。
aminsert	regproc	PG_PROC.proname	“插入这个行”函数。
ambeginscan	regproc	PG_PROC.proname	“准备索引扫描” 函数。
amgettuple	regproc	PG_PROC.proname	“下一个有效行”函数，如果没有则为0。
amgetbitmap	regproc	PG_PROC.proname	“抓取所有的有效行” 函数，如果没有则为0。
amrescan	regproc	PG_PROC.proname	“（重新）开始索引扫描”函数。
amendscan	regproc	PG_PROC.proname	“索引扫描后清理” 函数。
ammarkpos	regproc	PG_PROC.proname	“标记当前扫描位置”函数。
amrestrpos	regproc	PG_PROC.proname	“恢复已标记的扫描位置”函数。
ammerge	regproc	PG_PROC.proname	“归并多个索引对象”函数。
ambuild	regproc	PG_PROC.proname	“建立新索引”函数。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
ambuildempty	regproc	PG_PROC.proname	“建立空索引”函数。
ambulkdelete	regproc	PG_PROC.proname	批量删除函数。
amvacuumcleanup	regproc	PG_PROC.proname	VACUUM后的清理函数。
amcanreturn	regproc	PG_PROC.proname	检查是否索引支持唯一索引扫描的函数，如果没有则为0。
amcostestimate	regproc	PG_PROC.proname	估计一个索引扫描开销的函数。
amoptions	regproc	PG_PROC.proname	为一个索引分析和确认reloptions的函数。

7.2.18.27 PG_AMOP

PG_AMOP系统表存储有关和访问方法操作符族关联的信息。如果一个操作符是一个操作符族中的成员，则在这个表中会占据一行。一个族成员是一个search操作符或一个ordering操作符。一个操作符可以在多个族中出现，但是不能在一个族中的多个搜索位置或多个排序位置中出现。

表 7-99 PG_AMOP 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
amopfamily	oid	PG_OPFAMILY.oid	这个项的操作符族。
amoplefttype	oid	PG_TYPE.oid	操作符的左输入类型。取值及其描述请参见数据类型。
amoprighttype	oid	PG_TYPE.oid	操作符的右输入类型。取值及其描述请参见数据类型。
amopstrategy	smallint	-	操作符策略数。
amoppurpose	"char"	-	操作符目的。 • s：表示搜索。 • o：表示排序。
amopopr	oid	PG_OPERATOR.oid	该操作符的OID。
amopmethod	oid	PG_AM.oid	索引访问方式操作符族。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
amopsortfamily	oid	PG_OPFAMILY.oid	如果是一个排序操作符，则为这个项排序所依据的btree操作符族；如果是一个搜索操作符，则为0。

search操作符表明这个操作符族的一个索引可以被搜索，找到所有满足WHERE indexed_column operator constant的行。这样的操作符必须返回布尔值，并且它的左输入类型必须匹配索引的字段数据类型。

ordering操作符表明这个操作符族的一个索引可以被扫描，返回以ORDER BY indexed_column operator constant顺序表示的行。这样的操作符可以返回任意可排序的数据类型，它的左输入类型也必须匹配索引的字段数据类型。 ORDER BY的确切语义是由amopsortfamily字段指定的，该字段必须为操作符的返回类型引用一个btree操作符族。

7.2.18.28 PG_AMPROC

PG_AMPROC系统表存储有关与访问方法操作符族相关联的支持过程的信息。每个属于操作符族的支持过程都占有一行。

表 7-100 PG_AMPROC 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
amprocfamily	oid	PG_OPFAMILY.oid	该项的操作符族。
amproclefttype	oid	PG_TYPE.oid	相关操作符的左输入数据类型。 常见的数据类型请参见**数据类型**。
amprocrighttype	oid	PG_TYPE.oid	相关操作符的右输入数据类型。 常见的数据类型请参见**数据类型**。
amprocnum	smallint	-	支持过程编号。
amproc	regproc	PG_PROC.proname	过程的OID。

amproclefttype和amprocrighttype字段的习惯解释，标识一个特定支持过程支持的操作符的左和右输入类型。对于某些访问方式，匹配支持过程本身的输入数据类型，对其他的则不这样。有一个对索引的“缺省”支持过程的概念，amproclefttype和amprocrighttype都等于索引操作符类的opcintype。

版权所有 © 华为云计算技术有限公司

7.2.18.29 PG_ATTRDEF

PG_ATTRDEF系统表存储列的默认值。

表 7-101 PG_ATTRDEF 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
adrelid	oid	该列的所属表。
adnum	smallint	该列的数目。
adbin	pg_node_tree	字段缺省值的内部表现形式。
adsrc	text	可读的缺省值的内部表现形式。
adgencol	"char"	标识该列是否为生成列。取值为's'表示该列为生成列，取值为'\0'表示该列为普通列，默认值为'\0'。
adbin_on_update	pg_node_tree	字段on update current_timestamp属性表达式的内部表现形式。
adsrc_on_update	text	可读on update current_timestamp属性表达式的内部表现形式。
adbin_extend	pg_node_tree	生成表达式的内部表现形式。在生成列引用其他生成列的场景中，生成表达式中不含有生成列引用，是adbin字段将生成表达式中引用的其他生成列展开为其生成表达式的等价形式。

7.2.18.30 PG_ATTRIBUTE

PG_ATTRIBUTE系统表存储关于表字段的信息。

表 7-102 PG_ATTRIBUTE 字段

名称	类型	描述
attrelid	oid	此字段所属表。
attname	name	字段名。当该字段表示索引字段时，若原表修改了字段名，当前字段的信息不会更新，需要使用pg_get_indexdef函数获取准确字段名。
atttypid	oid	字段类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
attstattarget	integer	控制ANALYZE为这个字段积累的统计细节的级别。 ● 零值表示不收集统计信息。 ● 负数表示使用系统缺省的统计对象。 ● 正数值的确切信息是和数据类型相关的。 对于标量数据类型，ATTSTATTARGET既是要收集的“最常用数值”的目标数目，也是要创建的柱状图的目标数量。
attlen	smallint	是本字段类型的12.2.15.78 PG_TYPE中typlen的复制。
attnum	smallint	字段编号。
attndims	integer	如果该字段是数组，则是维数，否则是0 。
attcacheoff	integer	在磁盘上的时候总是-1 ，但是如果加载入内存中的行描述器中，它可能会被更新以缓冲在行中字段的偏移量。
atttypmod	integer	记录创建新表时支持的类型特定的数据（比如一个varchar字段的最大长度）。它传递给类型相关的输入和长度转换函数当做第三个参数。其值对那些不需要ATTTYPMOD的类型通常为-1。
attbyval	boolean	这个字段类型的12.2.15.78 PG_TYPE中typbyval的复制。
attstorage	"char"	这个字段类型的12.2.15.78 PG_TYPE中typstorage的复制。
attalign	"char"	这个字段类型的12.2.15.78 PG_TYPE中typalign的复制。
attnotnull	boolean	这代表一个非空约束。可以改变这个字段以打开或者关闭这个约束。
atthasdef	boolean	这个字段有一个缺省值，此时它对应12.2.15.25 PG_ATTRDEF表里实际定义此值的记录。
attisdropped	boolean	这个字段已经被删除了，不再有效。一个已经删除的字段物理上仍然存在表中，但会被分析器忽略，因此不能再通过SQL访问。
attislocal	boolean	这个字段是局部定义在关系中的。请注意一个字段可以同时是局部定义和继承的。
attcmprmode	tinyint	对某一列指定压缩方式。压缩方式包括： ● 0：ATT_CMPR_NOCOMPRESS，不压缩 ● 1：ATT_CMPR_DELTA，DELTA压缩算法 ● 2：ATT_CMPR_DICTIONARY，字典压缩算法 ● 3：ATT_CMPR_PREFIX，前缀压缩算法 ● 4：ATT_CMPR_NUMSTR，数字字符串压缩算法

版权所有 © 华为云计算技术有限公司

名称	类型	描述
attinhcount	integer	这个字段所拥有的直接父表的个数。如果一个字段的父表个数非零，则它就不能被删除或重命名。
attcollation	oid	对此列定义的校对列。
attacl	aclitem[]	列级访问权限控制。
attoptions	text[]	字段属性。目前支持以下两种属性： ● n_distinct，表示该字段的distinct值数量（不包含子表）。 ● n_distinct_inherited，表示该字段的distinct值数量（包含子表）。
attfdwoptions	text[]	外表字段属性。当前支持的dist_fdw、file_fdw、log_fdw未使用外表字段属性。
attinitdefval	bytea	存储了此列默认的值表达式。行存表的ADD COLUMN需要使用此字段。
attkvtype	tinyint	对某一列指定key value类型。类型包括： 0：ATT_KV_UNDEFINED，默认。 1：ATT_KV_TAG，维度。 2：ATT_KV_FIELD，指标。 3：ATT_KV_TIMETAG，时间列。 4：ATT_KV_HIDETAG，隐藏分布列。
attidentity	"char"	表示该列的identity类型。类型包括： '0'或'\0'：非IDENTITY列。 'a'：IDENTITY列属性为ALWAYS类型。 'd'：IDENTITY列属性为BY DEFAULT类型。 'n'：IDENTITY列属性为BY DEFAULT ON NULL类型。

7.2.18.31 PG_CAST

PG_CAST系统表存储数据类型之间的转换关系。

表 7-103 PG_CAST 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
castsource	oid	源数据类型的OID。
casttarget	oid	目标数据类型的OID。
castfunc	oid	转换函数的OID。如果为零表明不需要转换函数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
castcontext	"char"	源数据类型和目标数据类型间的转换方式： ● 'e'：表示只能进行显式转换（使用CAST或::语法）。 ● 'i'：表示能进行隐式转换。 ● 'a'：表示赋值转换，在对目标列进行赋值时，同时支持隐式转换和显式转换。
castmethod	"char"	转换方法： ● 'f'：使用castfunc字段中指定的函数进行转换。 ● 'b'：类型间是二进制强制转换，不使用castfunc。

7.2.18.32 PG_CLASS

PG_CLASS系统表存储数据库对象信息及其之间的关系。

表 7-104 PG_CLASS 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
relname	name	表、索引、视图等对象的名称。
relnamespace	oid	包含这个关系的名称空间的OID。
reltype	oid	对应这个表的行类型的数据类型（索引为零，因为索引没有pg_type记录）。
reloftype	oid	复合类型的OID，0表示其他类型。
relowner	oid	关系所有者。
relam	oid	如果行是索引，则就是所用的访问模式（B-tree等）。
relfilenode	oid	这个关系在磁盘上的文件的名称，如果没有则为0。
reltablespace	oid	这个关系存储所在的表空间。如果为零，则意味着使用该数据库的缺省表空间。如果关系在磁盘上没有文件，则这个字段没有什么意义。
relpages	double precision	以页（大小为BLCKSZ）为单位的该表在磁盘上的大小，它只是优化器用的一个近似值。
reltuples	double precision	表中行的数目，只是优化器使用的一个估计值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relallvisible	integer	被标识为全可见的表中的页的数量。此字段是优化器用来做SQL执行优化使用的。VACUUM、ANALYZE和一些DDL语句（例如，CREATE INDEX）会引起此字段更新。
reltoastrelid	oid	与该表关联的TOAST表的OID，如果没有则为0。 TOAST表在一个从属表里“离线”存储大字段。
reltoastidxid	oid	对于TOAST表是它的索引的OID，如果不是TOAST表则为0。
reldeltarelid	oid	暂不支持。
reldeltaidx	oid	暂不支持。
relcudescrelid	oid	暂不支持。
relcudescidx	oid	暂不支持。
relhasindex	boolean	如果它是一个表而且至少有（或者最近有过）一个索引，则为true。 它由CREATE INDEX设置，但DROP INDEX不会立即将其清除。如果VACUUM线程检测到某个表没有索引，将会清理relhasindex字段，并将其值设置为false。
relisshared	boolean	如果该表在整个集群中由所有数据库共享则为true，否则为false。只有某些系统表（比如pg_database）是共享的。
relpersistence	"char"	• p：表示永久表。 • u：表示非日志表。 • t：表示临时表。 • g：表示全局临时表。
relkind	"char"	• r：表示普通表。 • i：表示索引。 • G：表示全局二级索引。 • S：表示序列。 • v：表示视图。 • c：表示复合类型。 • t：表示TOAST表。 • f：表示外表。 • m：表示物化视图。 • e：表示STREAM对象。 • o：表示CONTVIEW对象。
relnatts	smallint	关系中用户字段数目（除了系统字段以外）。在12.2.15.26 PG_ATTRIBUTE里有相同数目对应行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relchecks	smallint	表里的检查约束的数目，参阅12.2.15.30 PG_CONSTRAINT表。
relhasoids	boolean	如果为关系中每行都生成一个OID则为true，否则为false。
relhaspkey	boolean	如果这个表有一个（或者曾经有一个）主键，则为true，否则为false。
relhasrules	boolean	如表有规则就为true。是否有规则可参考系统表12.2.15.63 PG_REWRITE。
relhastriggers	boolean	True表示表中有触发器，或者曾经有过触发器。系统表12.2.15.72 PG_TRIGGER中记录了表和视图的触发器。
relhassubclass	boolean	如果有（或者曾经有）任何继承的子表，为true，否则为false。
relcmprs	tinyint	表示是否启用表的压缩特性。需要特别注意，当且仅当批量插入才会触发压缩，普通的CRUD并不能够触发压缩。 • 0表示其他不支持压缩的表（主要是指系统表，不支持压缩属性的修改操作）。 • 1表示表数据的压缩特性为NOCOMPRESS或者无指定关键字。 • 2表示表数据的压缩特性为COMPRESS。
relhasclusterkey	boolean	暂不支持。
relrowmovement	boolean	针对分区表进行update操作时，是否允许行迁移。 • true：表示允许行迁移。 • false：表示不允许行迁移。
parttype	"char"	表或者索引是否具有分区表的性质。 • p：表示带有分区表性质。 • n：表示没有分区表特性。 • s：表示该表为二级分区表。
relfrozenxid	xid32	该表中所有在这个之前的事务ID已经被一个固定的（"frozen"）事务ID替换。该字段用于跟踪该表是否需要为了防止事务ID重叠（或者允许收缩pg_clog）而进行清理。如果该关系不是表则为零（InvalidTransactionId）。 为保持前向兼容，保留此字段，新增relfrozenxid64用于记录此信息。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relacl	aclitem[]	访问权限。aclitem类型说明可以参考**aclitem类型**。 查询的回显结果为以下形式： *user1=privs/user2* 表示user2赋予user1的权限为*privs* *=privs/user3* 表示user3赋予public角色的权限为*privs* 其中user1、user2和user3为数据库中已存在的用户/角色名，privs为数据库中支持的权限。权限的参数说明请参见**表7-105**。
reloptions	text[]	表、索引、视图等对象的访问方法，使用"keyword=value"格式的字符串。
relreplident	"char"	逻辑解码中解码列的标识： ● d：默认 (主键，如果存在)。 ● n：无。 ● f：所有列。 ● i：索引的indisreplident被设置或者为默认。
relfrozenxid64	xid	该表中所有在这个之前的事务ID已经被一个固定的（"frozen"）事务ID替换。该字段用于跟踪该表是否需要为了防止事务ID重叠（或者允许收缩pg_clog）而进行清理。如果该关系不是表则为零（InvalidTransactionId）。 对于全局临时表，该字段无实际意义。各会话的全局临时表的relfrozenxid64可使用pg_catalog.gs_global_list_gtt_relfrozenxids函数查看。
relbucket	oid	当前表是否包含hash bucket分片。有效的OID指向pg_hashbucket表中记录的具体分片信息。NULL表示不包含hash bucket分片。
relbucketkey	int2vector	表示hash分区列信息，NULL表示不包含。
relminmxid	xid	该表中所有在这个之前的多事务ID已经被一个事务ID替换。该字段用于跟踪该表是否需要为了防止多事务ID重叠或者允许收缩pg_clog而进行清理。如果该关系不是表则为零（InvalidTransactionId）。

表 7-105 权限的参数说明

参数	参数说明
r	SELECT（读）
w	UPDATE（写）
a	INSERT（插入）
d	DELETE

版权所有 © 华为云计算技术有限公司

参数	参数说明
D	TRUNCATE
x	REFERENCES
t	TRIGGER
X	EXECUTE
U	USAGE
C	CREATE
c	CONNECT
T	TEMPORARY
A	ALTER
P	DROP
m	COMMENT
i	INDEX
v	VACUUM
*	给前面权限的授权选项

7.2.18.33 PG_COLLATION

PG_COLLATION系统表描述可用的排序规则，本质上从一个SQL名称映射到操作系统本地类别。

表 7-106 PG_COLLATION 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
collname	name	-	排序规则名（每个名称空间和编码唯一）。
collnamespace	oid	PG_NAMESPACE.oid	包含这个排序规则的名称空间的OID。
collowner	oid	PG_AUTHID.oid	排序规则的所有者。
collencoding	integer	-	排序规则可用的编码，兼容PostgreSQL所有的字符编码类型，如果适用于任意编码为-1。
collcollate	name	-	这个排序规则对象的LC_COLLATE。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
collctype	name	-	这个排序规则对象的LC_CTYPE。
collpadattr	text	-	这个排序规则的填充属性。 ● NULL：不适用。 ● NO PAD：无填充。 ● PAD SPACE：末尾空白填充。
collisdef	boolean	-	这个排序规则是否是所属字符集的默认字符序。
id	integer	-	当前版本暂不支持。

7.2.18.34 PG_CONSTRAINT

PG_CONSTRAINT系统表存储表上的检查约束、主键、唯一约束和外键约束。

表 7-107 PG_CONSTRAINT 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
conname	name	约束名称（不一定是唯一的）。
connamespace	oid	包含这个约束的名称空间的OID。
contype	"char"	● c：检查约束。 ● p：主键约束。 ● u：唯一约束。 ● t：触发器约束。 ● x：互斥约束。 ● f：外键约束。 ● s：聚簇约束。 ● i：无效约束。
condeferrable	boolean	这个约束是否可以推迟。 ● true：表示可以。 ● false：表示不可以。
condeferred	boolean	缺省时这个约束是否可以推迟。 ● true：表示可以。 ● false：表示不可以。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
convalidated	boolean	约束是否有效。目前，只有外键和CHECK约束可将其设置为false。 ● true：表示有效。 ● false：表示无效。
conrelid	oid	这个约束所在的表，如果不是表约束则为0。
contypid	oid	这个约束所在的域，如果不是一个域约束则为0。
conindid	oid	与约束关联的索引ID。
confrelid	oid	如果是外键，则为参考的表，否则为0。
confupdtype	"char"	外键更新动作代码。 ● a：没动作。 ● r：限制。 ● c：级联。 ● n：设置为null。 ● d：设置为缺省。
confdeltype	"char"	外键删除动作代码。 ● a：没动作。 ● r：限制。 ● c：级联。 ● n：设置为null。 ● d：设置为缺省。
confmatchtype	"char"	外键匹配类型。 ● f：全部。 ● p：部分。 ● u：未指定（在f的基础上允许匹配NULL值）。
conislocal	boolean	是否是为关系创建的本地约束。 ● true：表示是。 ● false：表示不是。
coninhcount	integer	约束直接继承父表的数目。继承父表数非零时，不能删除或重命名该约束。
connoinherit	boolean	是否可以被继承。 ● true：表示可以。 ● false：表示不可以。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
consoft	boolean	是否为信息约束(Informational Constraint)。 ● true：表示是。 ● false：表示不是。
conopt	boolean	是否使用信息约束优化执行计划。 ● true：表示使用。 ● false：表示不使用。
conkey	smallint[]	如果是表约束，则是约束控制的字段列表。
confkey	smallint[]	如果是一个外键，是参考的字段的列表。
conpfeqop	oid[]	如果是一个外键，是做PK=FK比较的相等操作符ID的列表。
conppeqop	oid[]	如果是一个外键，是做PK=PK比较的相等操作符ID的列表。
conffeqop	oid[]	如果是一个外键，是做FK=FK比较的相等操作符ID的列表。
conexclop	oid[]	如果是一个排他约束，是列的排他操作符ID列表。
conbin	pg_node_tree	如果是检查约束，那就是其表达式的内部形式。
consrc	text	如果是检查约束，则是表达式的可读形式。
conincluding	smallint[]	不用做约束，但是会包含在INDEX中的属性列。

须知

- consrc在被引用的对象改变之后不会被更新，它不会跟踪字段的名称修改。建议使用pg_get_constraintdef()来抽取一个检查约束的定义。
- PG_CLASS的relchecks需要和在该表上为给定关系找到的检查约束的数目一致。

7.2.18.35 PG_CONVERSION

PG_CONVERSION系统表描述编码转换信息。

表 7-108 PG_CONVERSION 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
conname	name	-	转换名称（在一个名称空间里是唯一的）。
connamespace	oid	PG_NAMESPACE.oid	包含这个转换的名称空间的OID。
conowner	oid	PG_AUTHID.oid	编码转换的属主。
conforencoding	integer	-	源编码ID。
contoencoding	integer	-	目的编码ID。
conproc	regproc	PG_PROC.proname	转换过程。
condefault	boolean	-	如果这是缺省转换则为真，否则为假。

7.2.18.36 PG_DATABASE

PG_DATABASE系统表存储关于可用数据库的信息。

表 7-109 PG_DATABASE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
datname	name	数据库名称。
datdba	oid	数据库所有人，通常为其创建者。
encoding	integer	数据库的字符编码方式。
datcollate	name	数据库使用的排序顺序。
datctype	name	数据库使用的字符分类。
datistemplate	boolean	是否允许作为模板数据库。 ● true：表示允许。 ● false：表示不允许。
datallowconn	boolean	这个字段用于保护template0/templatea数据库不被更改。 ● true：表示用户可以连接到这个数据库。 ● false：表示没有用户可以连接到这个数据库。
datconnlimit	integer	该数据库上允许的最大并发连接数，-1表示无限制。
datlastsysoid	oid	数据库里最后一个系统OID 。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
datfrozenxid	xid32	用于跟踪该数据库是否需要为了防止事务ID重叠而进行清理。当前版本该字段已经废弃使用，为保持前向兼容，保留此字段，新增datfrozenxid64用于记录此信息。
dattablespace	oid	数据库的缺省表空间。
datcompatibility	name	数据库兼容模式。当前支持四种兼容模式：PG、ORA、MYSQL、TD。
datacl	aclitem[]	访问权限。
datfrozenxid64	xid	用于跟踪该数据库是否需要为了防止事务ID重叠而进行清理。
datminmxid	xid	该数据库中所有在这个之前的多事务ID已经被一个事务ID替换。这用于跟踪该数据库是否需要为了防止事务ID重叠或者允许收缩pg_clog而进行清理。它是此数据库中所有表的12.2.15.28 PG_CLASS中relminmxid的最小值。
dattimezone	name	数据库时区信息，默认为PRC时区。
dattype	"char"	数据库类型。“D”表示普通数据库。

7.2.18.37 PG_DEPEND

PG_DEPEND系统表记录数据库对象之间的依赖关系。这个信息允许DROP命令找出哪些其它对象必须由DROP CASCADE删除，或者是在DROP RESTRICT的情况下避免删除。

这个表的功能类似PG_SHDEPEND，用于记录那些在数据库集群之间共享的对象之间的依赖性关系。

表 7-110 PG_DEPEND 字段

名称	类型	引用	描述
classid	oid	PG_CLASS.oid	有依赖对象所在系统表的OID。
objid	oid	任意OID属性	指定的依赖对象的OID。
objsubid	integer	-	对于表字段，这个是该属性的字段数（objid和classid引用表本身）。对于所有其它对象类型，目前这个字段是0。
refclassid	oid	PG_CLASS.oid	被引用对象所在的系统表的OID。
refobjid	oid	任意OID属性	指定的被引用对象的OID。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
refobjsubid	integer	-	对于表字段，这个是该字段的字段号（refobjid和refclassid引用表本身）。对于所有其它对象类型，目前这个字段是0。
deptype	"char"	-	一个定义这个依赖关系特定语义的代码。
depsrc	text	-	记录被引用对象的名称。

在所有情况下，一个PG_DEPEND记录表示被引用的对象不能在有依赖的对象被删除前删除。不过，这里还有几种由deptype定义的情况：

- DEPENDENCY_NORMAL (n)：独立创建的对象之间的一般关系。有依赖的对象可以在不影响被引用对象的情况下删除。被引用对象只有在声明了CASCADE的情况下删除，这时有依赖的对象也被删除。例子：一个表字段对其数据类型有一般依赖关系。
- DEPENDENCY_AUTO (a)：有依赖对象可以和被引用对象分别删除，并且如果删除了被引用对象则应该被自动删除（不管是RESTRICT或CASCADE模式）。例子：一个表上面的命名约束是在该表上的自动依赖关系，因此如果删除了表，它也会被删除。
- DEPENDENCY_INTERNAL (i)：有依赖的对象是作为被引用对象的一部分创建的，实际上只是它的内部实现的一部分。 DROP有依赖对象是不能直接允许的（将告诉用户发出一条删除被引用对象的DROP）。一个对被引用对象的DROP将传播到有依赖对象，不管是否声明了CASCADE。例子：一个创建来强制外键约束的触发器在该约束的**PG_CONSTRAINT**记录上是标记为内部依赖的。
- DEPENDENCY_EXTENSION (e)：依赖对象是被依赖对象extension的一个成员（请参见**PG_EXTENSION**）。依赖对象只可以通过在被依赖对象上DROP EXTENSION删除。函数上这个依赖类型和内部依赖一样动作，但是它为了清晰和简化gs_dump保持分开。
- DEPENDENCY_PIN (p)：没有依赖对象，这种类型的记录标志着系统本身依赖于被引用对象，因此这个对象绝不能被删除。这种类型的记录只有在initdb的时候创建。有依赖对象的字段里是零。
- DEPENDENCY_COMPILE (c)：依赖对象是视图时，记录编译过程中对被依赖对象的依赖关系。例如，若视图的定义中使用表列，创建视图和重编译时会记录视图对表的编译依赖关系。

7.2.18.38 PG_DESCRIPTION

PG_DESCRIPTION系统表可以给每个数据库对象存储一个可选的描述（注释）。许多内置的系统对象的描述提供了PG_DESCRIPTION的初始内容。

这个表的功能类似**PG_SHDESCRIPTION**，用于记录整个集群范围内共享对象的注释。

表 7-111 PG_DESCRIPTION 字段

名称	类型	引用	描述
objoid	oid	任意OID属性	这条描述所描述的对象的OID。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
classoid	oid	PG_CLASS.oid	这个对象出现的系统表的OID。
objsubid	integer	-	对于一个表字段的注释，它是字段号（objoid和classoid指向表自身）。对于其它对象类型，它是零。
description	text	-	对该对象描述的任意文本。

7.2.18.39 PG_DIRECTORY

PG_DIRECTORY系统表用于保存用户添加的directory对象，可以通过CREATE DIRECTORY语句向该表中添加记录，当enable_access_server_directory=off时，只允许初始用户创建directory对象；当enable_access_server_directory=on时，具有SYSADMIN权限的用户和继承了内置角色gs_role_directory_create权限的用户可以创建directory对象。普通用户需要授权才能访问该表。

表 7-112 PG_DIRECTORY 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
dirname	name	目录对象的名称。
owner	oid	目录对象的所有者。
dirpath	text	目录路径。
diracl	aclitem[]	访问权限。

7.2.18.40 PG_ENUM

PG_ENUM系统表包含显示每个枚举类型值和标签的记录。给定枚举类型的内部表示实际上是PG_ENUM里面相关行的OID。

表 7-113 PG_ENUM 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
enumtypid	oid	PG_TYPE.oid	拥有这个枚举值的PG_TYPE记录的OID。
enumsortorder	real	-	这个枚举值在它的枚举类型中的排序位置。
enumlabel	name	-	这个枚举值的文本标签。

版权所有 © 华为云计算技术有限公司

PG_ENUM行的OID跟着一个特殊规则：偶数的OID保证用和它们的枚举类型一样的排序顺序。也就是，如果两个偶数OID属于相同的枚举类型，那么较小的OID必须有较小enumsortorder值。奇数OID需要毫无关系的排序顺序。这个规则允许枚举比较例程在许多常见情况下避开目录查找。创建和修改枚举类型的例程只要可能就尝试分配偶数OID给枚举值。

当创建了一个枚举类型时，它的成员赋予了排序顺序位置1到n。但是随后添加的成员可能会分配enumsortorder的负值或分数值。对这些值的唯一要求是它们要正确的排序和在每个枚举类型中唯一。

7.2.18.41 PG_EXTENSION

PG_EXTENSION系统表存储关于所安装扩展的信息。GaussDB默认有以下扩展：plpgsql、DIST_FDW、FILE_FDW、GC_FDW、DBLINK_FDW、PACKAGES、ROACH_API、STREAMING、TSDB、GSREDISTRIBUTE、SECURITY_PLUGIN、GSSTAT_PLUGIN、NUMERIC_ENHANCE、PKG_DBE_RAW、PKG_DBE_OUTPUT、PKG_DBE_UTILITY、PKG_DBE_XML、PKG_DBE_OBFUSCATION_TOOLKIT、PKG_ILM、PKG_DBE_XMLGEN、PKG_DBE_STATS、PKG_DBE_DESCRIBE和PKG_DBE_LICENSE。

表 7-114 PG_EXTENSION

名称	类型	描述
oid	oid	行标识符（隐含属性，必须明确选择）。
extname	name	扩展名。
extowner	oid	扩展的所有者。
extnamespace	oid	扩展导出对象的名称空间。
extrelocatable	boolean	标识此扩展是否可迁移到其他名称空间，true表示可以，false表示不可以。
extversion	text	扩展的版本号。
extconfig	oid[]	扩展的配置信息。
extcondition	text[]	扩展配置信息的过滤条件。

7.2.18.42 PG_EXTENSION_DATA_SOURCE

PG_EXTENSION_DATA_SOURCE系统表存储外部数据源对象的信息。一个外部数据源对象（Data Source）包含了外部数据库的一些密码编码等信息，主要配合Extension Connector使用，默认仅系统管理员可访问此表。由于规格变更，当前版本已经不再支持本特性，请不要使用。

版权所有 © 华为云计算技术有限公司

表 7-115 PG_EXTENSION_DATA_SOURCE 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
srcname	name	-	外部数据源对象的名称。
srcowner	oid	PG_AUTHID.oid	外部数据源对象的所有者。
srctype	text	-	外部数据源对象的类型，缺省为空。
srcversion	text	-	外部数据源对象的版本，缺省为空。
srcacl	aclitem[]	-	访问权限。
srcoptions	text[]	-	外部数据源对象的指定选项，使用“keyword=value”格式的字符串。

7.2.18.43 PG_FOREIGN_DATA_WRAPPER

PG_FOREIGN_DATA_WRAPPER系统表存储外部数据封装器定义。一个外部数据封装器是在外部服务器上驻留外部数据的机制，是可以访问的。

表 7-116 PG_FOREIGN_DATA_WRAPPER 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
fdwname	name	-	外部数据封装器名。
fdwowner	oid	PG_AUTHID.oid	外部数据封装器的所有者。
fdwhandler	oid	PG_PROC.oid	引用一个负责为外部数据封装器提供扩展例程的处理函数。如果没有提供处理函数则为零。
fdwvalidator	oid	PG_PROC.oid	引用一个验证器函数，这个验证器函数负责验证给予外部数据封装器的选项、外部服务器选项和使用外部数据封装器的用户映射的有效性。如果没有提供验证器函数则为零。
fdwacl	aclitem[]	-	访问权限。
fdwoptions	text[]	-	外部数据封装器指定选项，使用“keyword=value”格式的字符串。

版权所有 © 华为云计算技术有限公司

7.2.18.44 PG_FOREIGN_SERVER

PG_FOREIGN_SERVER系统表存储外部服务器定义。一个外部服务器描述了一个外部数据源，例如一个远程服务器。外部服务器通过外部数据封装器访问。

表 7-117 PG_FOREIGN_SERVER 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
srvname	name	-	外部服务器名。
srvowner	oid	PG_AUTHID.oid	外部服务器的所有者。
srvfdw	oid	PG_FOREIGN_DATA_WRAPPER.oid	外部服务器的外部数据封装器的OID。
srvtype	text	-	服务器的类型（可选）。
srvversion	text	-	服务器的版本（可选）。
srvacl	aclitem[]	-	访问权限。
srvoptions	text[]	-	外部服务器指定选项，使用“keyword=value”格式的字符串。

7.2.18.45 PG_FOREIGN_TABLE

PG_FOREIGN_TABLE系统表存储外部表的辅助信息。

表 7-118 PG_FOREIGN_TABLE 字段

名称	类型	描述
ftrelid	oid	外部表的ID。
ftserver	oid	外部表的所在服务器。
ftwriteonly	boolean	外部表是否可写，取值如下： • t（true）：表示可写。 • f（false）：表示不可写。
ftoptions	text[]	外部表的可选项，具体参考7.13.8.22 CREATE FOREIGN TABLE语法说明。

7.2.18.46 PG_HASHBUCKET

PG_HASHBUCKET系统表存储hashbucket信息。

版权所有 © 华为云计算技术有限公司

表 7-119 PG_HASHBUCKET 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
bucketid	oid	CN上为PG_HASHBUCKET系统表所在DATABASE绑定的node group。DN上此列为空。
bucketcnt	integer	CN上不使用此参数，DN上为当前DN所拥有的bucket数量。
bucketmapsize	integer	当前版本不使用此参数。
bucketref	integer	预留字段，默认值为1。
bucketvector	oidvector_extend	CN上不使用此参数，DN上为当前DN所拥有的bucket列表。
bucketmap	text	用来存储逻辑bucket到物理bucket的映射关系，即16384到1024的映射关系。
bucketversion	oidvector_extend	记录后续hashbucket扩容过程中发生改变的信息版本号。
bucketcsn	text	hashbucket重分布前源节点每个bucket的对应的最大CSN，用于新节点可见性判断。
bucketxid	text	hashbucket扩容，新节点上线设置的next_xid，用于校验是否在阈值范围内。
bucketclogmode	smallint	bucket类型表采用的CLOG目录类型： • 空：表示采用pg_clog/<bucketid>目录； • 1：表示采用pg_clog/base/<dbid>/<bucketid>目录。
cbibuckets	oidvector_extend	记录CBI在线重建过程中正在进行的bucket列表。

7.2.18.47 PG_INDEX

PG_INDEX系统表存储索引的一部分信息，其他的信息大多数在PG_CLASS中。

表 7-120 PG_INDEX 字段

名称	类型	描述
indexrelid	oid	这个索引在12.2.15.28 PG_CLASS里的记录的OID。
indrelid	oid	使用这个索引的表在12.2.15.28 PG_CLASS里的记录的OID。
indnatts	smallint	索引中的字段数目。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
indisunique	boolean	是否为唯一索引。 ● true：是唯一索引。 ● false：不是唯一索引。
indisprimary	boolean	该索引是否为该表的主键。 ● true：该索引是该表的主键。这个字段为真的时候indisunique总是为真。 ● false：该索引不是该表的主键。
indisexclusion	boolean	该索引是否支持排他约束。 ● true：该索引支持排他约束。 ● false：该索引不支持排他约束。
indimmediate	boolean	插入数据时是否进行唯一性检查。 ● true：在插入数据时会立即进行唯一性检查。 ● false：在插入数据时不会进行唯一性检查。
indisclustered	boolean	该表是否在这个索引上建簇。 ● true：该表在这个索引上建簇。 ● false：该表没有在这个索引上建簇。
indisusable	boolean	该索引对insert/select是否可用。 ● true：该索引对insert/select可用。 ● false：该索引对insert/select不可用。
indisvalid	boolean	● true：该索引可以用于查询。 ● false：该索引可能不完整，仍然必须在INSERT/UPDATE操作时进行更新，不过不能安全的用于查询。如果是唯一索引，则唯一属性也将不为真。
indcheckxmin	boolean	● true：查询不能使用索引，直到pg_index此行的xmin低于其快照的TransactionXmin，因为该表可能包含它们能看到的不兼容行断开的HOT链。 ● false：查询可以用索引。
indisready	boolean	● true：此索引对插入数据是可用的。 ● false：在插入或修改数据时忽略此索引。
indkey	int2vector	这是一个包含indnatts值的数组，这些数组值表示这个索引所建立的表字段。比如一个值为1 3的意思是第一个字段和第三个字段组成这个索引键字。这个数组里的零表明对应的索引属性是在这个表字段上的一个表达式，而不是一个简单的字段引用。
indcollation	oidvector	索引各列对应的排序规则的OID，详情请参见PG_COLLATION。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
indclass	oidvector	对于索引键字里面的每个字段，这个字段都包含一个指向所使用的操作符类的OID，详情请参见12.2.15.53 PG_OPCLASS。
indoption	int2vector	存储列前标识位，该标识位是由索引的访问方法定义。
indexprs	pg_node_tree	表达式树（以nodeToString()形式表现）用于那些非简单字段引用的索引属性。它是一个列表，个数与indkey中的零值个数相同。如果所有索引属性都是简单的引用，则为空。
indpred	pg_node_tree	部分索引断言的表达式树（以nodeToString()的形式表现）。如果不是部分索引，则是空字符串。
indisreplident	boolean	此索引的列是否为逻辑解码的解码列。 • true：此索引的列成为逻辑解码的解码列。 • false：此索引的列不是逻辑解码的解码列。
indnkeyatts	smallint	索引中的总字段数，超出indnatts的部分不参与索引查询。
indcctmpid	oid	Ustore在线创建索引时临时表OID。
indisvisible	boolean	• true：此索引状态为可见，即优化器可以使用此索引。 • false：此索引状态为不可见，此时若enable_invisible_indexes参数为off，则优化器不可以使用此索引；若enable_invisible_indexes参数为on，则优化器可以使用此索引。

7.2.18.48 PG_INHERITS

PG_INHERITS系统表记录关于表继承层次的信息。数据库里每个直接的子系表都有一条记录。间接的继承可以通过追溯记录链来判断。

表 7-121 PG_INHERITS 字段

名称	类型	引用	描述
inhrelid	oid	PG_CLASS.oid	子表的OID。
inhparent	oid	PG_CLASS.oid	父表的OID。
inhseqno	integer	-	如果一个子表存在多个直系父表（多重继承），这个数字表明此继承字段的排列顺序。计数从1开始。

版权所有 © 华为云计算技术有限公司

7.2.18.49 PG_JOB

PG_JOB系统表存储用户创建的定时任务的详细信息，定时任务线程定时轮询PG_JOB系统表中的时间，当任务到期会触发任务的执行，并更新PG_JOB表中的任务状态。该系统表属于Shared Relation，所有创建的job记录对所有数据库可见。普通用户需授权后才能访问。

表 7-122 PG_JOB 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
job_id	bigint	作业id，主键，是唯一的（有唯一索引）。
current_postgres_pid	bigint	如果当前任务已被执行，那么此处记录运行此任务的线程ID。默认为-1，表示此任务未被执行过。
log_user	name	创建者的UserName。
priv_user	name	作业执行者的UserName。
dbname	name	标识作业要在哪一个数据库执行的数据库名称。
node_name	name	标识当前作业是在哪一个CN上创建和执行。
job_status	"char"	当前任务的执行状态，默认为's'，各取值含义： ● 'r'：running ● 's'：successfully finished ● 'f'：job failed ● 'd'：disable 当job连续执行失败16次，会将job_status自动设置为失效状态'd'，后续不再执行该job。 注：当用户将定时任务关闭（即：GUC参数job_queue_processes为0时），由于监控job执行的线程不会启动，所以该状态不会根据job的实时状态进行设置，用户不需要关注此状态。只有当开启定时任务功能（即：GUC参数job_queue_processes为非0时），系统才会根据当前job的实时状态刷新该字段值。
start_date	timestamp without time zone	作业第一次开始执行时间，时间精确到毫秒。
next_run_date	timestamp without time zone	定时任务下次执行的时间，时间精确到毫秒。
failure_count	smallint	失败计数，作业连续执行失败16次，不再继续执行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
interval	text	作业执行的重复时间间隔。
last_start_date	timestamp without time zone	上次运行开始时间，时间精确到毫秒。
last_end_date	timestamp without time zone	上次运行的结束时间，时间精确到毫秒。
last_suc_date	timestamp without time zone	上次成功运行的开始时间，时间精确到毫秒。
this_run_date	timestamp without time zone	正在运行任务的开始时间，时间精确到毫秒。
nspname	name	标识作业执行时的schema的名称。
job_name	text	DBE_SCHEDULER定时任务专用，定时任务名称。
end_date	timestamp without time zone	DBE_SCHEDULER定时任务专用，定时任务失效时间，时间精确到毫秒。
enable	boolean	DBE_SCHEDULER定时任务专用，定时任务启用状态： • true：启用。 • false：未启用。
failure_msg	text	最新一次执行任务报错信息。

7.2.18.50 PG_JOB_PROC

PG_JOB_PROC系统表对应**PG_JOB**表中每个任务的作业内容（包括：PL/SQL代码块、匿名块）。将存储过程信息独立出来，是因为如果放到PG_JOB中，被加载到共享内存的时候，会占用不必要的空间，所以在使用的时候再进行查询获取。普通用户需授权后才能访问。

表 7-123 PG_JOB_PROC 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
job_id	integer	外键，关联**PG_JOB**中的job_id。
what	text	作业内容，DBE_SCHEDULER定时任务中的程序内容。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
job_name	text	DBE_SCHEDULER定时任务专用，定时任务或程序名称。

7.2.18.51 PG_LANGUAGE

PG_LANGUAGE系统表登记编程语言，用户可以用这些语言或接口写函数或者存储过程。

表 7-124 PG_LANGUAGE 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段；必须明确选择）。
lanname	name	-	语言的名称。
lanowner	oid	PG_AUTHID.oid	语言的所有者。
lanispl	boolean	-	• true：表示用户定义的语言。 • false：表示内部语言，比如SQL。 目前，gs_dump仍然使用该字段判断哪种语言需要转储，但是这些可能在将来被其它机制取代。
lanpltrusted	boolean	-	• true：这是可信语言，意味着系统相信它不会被授予任何正常SQL执行环境之外的权限。 • false：这是不可信语言。只有初始用户可以用不可信语言创建函数。
lanplcallfoid	oid	PG_PROC.oid	对于非内部语言，这是指向该语言处理器的引用，语言处理器是一个特殊函数， 负责执行以某种语言写的所有函数。
laninline	oid	PG_PROC.oid	这个字段引用一个负责执行“inline”匿名代码块的函数（DO块）。如果不支持内联块则为零。
lanvalidator	oid	PG_PROC.oid	这个字段引用一个语言校验器函数，它负责检查新创建的函数的语法和有效性。如果没有提供校验器，则为零。
lanacl	aclitem[]	-	访问权限。

版权所有 © 华为云计算技术有限公司

7.2.18.52 PG_NAMESPACE

PG_NAMESPACE系统表存储名称空间，即存储schema相关的信息。如果开启数据库对象隔离属性，用户只能查看自己有权限访问的schema信息。

表 7-125 PG_NAMESPACE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
nspname	name	名称空间的名称。
nspowner	oid	名称空间的所有者。
nsptimeline	bigint	在DN上创建此命名空间时的时间线。此字段为内部使用，仅在DN上有效。
nspacl	aclitem[]	访问权限。具体请参见**GRANT**和**REVOKE**。
in_redistribution	"char"	是否处于重发布状态。
nspblockchain	boolean	该模式是否为防篡改模式。 ● true：该模式为防篡改模式。 ● false：该模式为非防篡改模式。
nspcollation	oid	名称空间的默认字符序（仅在sql_compatibility='MYSQL'时可能有值）。
nspgroupoid	oid	单节点属性，在该模式下隐式创建单节点表时默认创建到该字段记录的节点组Node Group上。

7.2.18.53 PG_OBJECT

PG_OBJECT系统表存储限定类型对象（普通表、索引、序列、视图、存储过程和函数）的创建用户、创建时间和最后修改时间。

表 7-126 PG_OBJECT 字段

名称	类型	描述
object_oid	oid	对象标识符。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
object_type	"char"	对象类型： ● r： 表示普通表。 ● i： 表示索引。 ● s： 表示序列。 ● v： 表示视图。 ● P： 表示存储过程和函数。 ● S： 表示包头。 ● B： 表示包体。
creator	oid	对象的所有者。
ctime	timestamp with time zone	对象的创建时间。
mtime	timestamp with time zone	对象的最后修改时间，修改行为包括ALTER操作和GRANT、REVOKE操作。
createcsn	bigint	对象创建时的CSN。
changecsn	bigint	对表或索引执行DDL操作时的CSN。
valid	boolean	对象的有效性，t为有效，f为无效。
relchangecsn	bigint	对表结构变更或执行TRUNCATE DDL操作时的CSN。

须知

- 无法记录初始化数据库（initdb）过程中所创建或修改的对象，即PG_OBJECT无法查询到该对象记录。
- 对于升级至该版本的数据库，无法记录升级以前所创建的对象，即PG_OBJECT无法查询到该对象记录。
- 对于上述两类对象再次修改时，会记录其修改时间（mtime），由于无法得知该对象创建时间，因此ctime为空。
- 对于升级前创建的对象，再次修改时会记录其修改时间（mtime），对表或索引执行DDL操作时会记录其所属事务的事务提交序列号（changecsn）。由于无法得知该对象创建时间，因此ctime和createcsn为空。
- ctime和mtime所记录的时间为用户当次操作所属事务的起始时间。
- 由扩容引起的对象修改时间也会被记录。
- createcsn和changecsn记录的是用户当次操作所属事务的事务提交序列号。
- 执行对象创建语句时，存在未定义的对象，或所依赖的对象有修改或删除动作时，对象会为无效状态。

版权所有 © 华为云计算技术有限公司

7.2.18.54 PG_OPCLASS

PG_OPCLASS系统表定义索引访问方法操作符类。

每个操作符类为一种特定数据类型和一种特定索引访问方法定义索引字段的语义。一个操作符类本质上指定一个特定的操作符族适用于一个特定的可索引的字段数据类型。索引的字段实际可用的族中的操作符集是接受字段中左侧字段的数据类型。

表 7-127 PG_OPCLASS 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
opcmethod	oid	PG_AM.oid	操作符类所服务的索引访问方法。
opcname	name	-	这个操作符类的名称。
opcnamespace	oid	PG_NAMESPACE.oid	这个操作符类的名称空间。
opcowner	oid	PG_AUTHID.oid	操作符类属主。
opcfamily	oid	PG_OPFAMILY.oid	包含该操作符类的操作符族。
opcintype	oid	PG_TYPE.oid	操作符类索引的数据类型。
opcdefault	boolean	-	如果操作符类是opcintype的缺省，则为真（t），否则为假（f）。
opckeytype	oid	PG_TYPE.oid	索引数据的类型，如果和opcintype相同则为零。

一个操作符类的opcmethod必须匹配包含它的操作符族的opfmethod。

7.2.18.55 PG_OPERATOR

PG_OPERATOR系统表存储有关操作符的信息。

表 7-128 PG_OPERATOR 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
oprname	name	-	操作符的名称。
oprnamespace	oid	PG_NAMESPACE.oid	包含此操作符的名称空间的OID。
oprowner	oid	PG_AUTHID.oid	操作符所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
oprkind	"char"	-	• b：表示中缀(“两边”)。 • l：表示前缀(“左边”)。 • r：表示后缀(“右边”)。
oprcanmerge	boolean	-	这个操作符是否支持合并连接。 • t（true）：表示支持合并连接。 • f（false）：表示不支持合并连接。
oprcanhash	boolean	-	这个操作符是否支持Hash连接。 • t（true）：表示支持Hash连接。 • f（false）：表示不支持Hash连接。
oprleft	oid	**PG_TYPE**.oid	左操作数的类型。
oprright	oid	**PG_TYPE**.oid	右操作数的类型。
oprresult	oid	**PG_TYPE**.oid	结果类型。
oprcom	oid	**PG_OPERATOR**.oid	如果存在的话，值为此操作符的交换符。不存在的话，值为0。
oprnegate	oid	**PG_OPERATOR**.oid	如果存在的话，值为此操作符的反转器。不存在的话，值为0。
oprcode	regproc	**PG_PROC**.proname	实现这个操作符的函数。
oprrest	regproc	**PG_PROC**.proname	此操作符的约束选择性计算函数。
oprjoin	regproc	**PG_PROC**.proname	此操作符的连接选择性计算函数。

7.2.18.56 PG_OPFAMILY

PG_OPFAMILY系统表定义操作符族。

每个操作符族是一个操作符和相关支持例程的集合，其中的例程实现为一个特定的索引访问方式指定的语义。另外，族中的操作符都是“兼容的”，通过由访问方式指定的方法。操作符族的概念允许交叉数据类型操作符和索引一起使用，并且合理的使用访问方式的语义知识。

版权所有 © 华为云计算技术有限公司

表 7-129 PG_OPFAMILY 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
opfmethod	oid	PG_AM.oid	操作符族使用的索引方法。
opfname	name	-	这个操作符族的名称。
opfnamespace	oid	PG_NAMESPACE.oid	这个操作符的名称空间。
opfowner	oid	PG_AUTHID.oid	操作符族的所有者。

定义一个操作符族的大多数信息不在它的PG_OPFAMILY行里面，而是在相关的行PG_AMOP，PG_AMPROC和PG_OPCLASS里。

7.2.18.57 PG_PLTEMPLATE

PG_PLTEMPLATE系统表存储过程语言的“模板”信息。

表 7-130 PG_PLTEMPLATE 字段

名称	类型	描述
tmplname	name	这个模板所应用的语言的名称。
tmpltrusted	boolean	如果语言被认为是可信的，则为真。否则为假。
tmpldbacreate	boolean	如果语言是由数据库所有者创建的，则为真。否则为假。
tmplhandler	text	调用处理器函数的名称。
tmplinline	text	匿名块处理器的名称，若没有则为NULL。
tmplvalidator	text	校验函数的名称，如果没有则为NULL。
tmpllibrary	text	实现语言的共享库的路径。
tmplacl	aclitem[]	模板的访问权限（未使用）。

7.2.18.58 PG_PROC

PG_PROC系统表存储函数或过程的信息。

版权所有 © 华为云计算技术有限公司

表 7-131 PG_PROC 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
proname	name	函数名称。
pronamespace	oid	包含该函数名称空间的OID。
proowner	oid	函数的所有者。
prolang	oid	这个函数的实现语言或调用接口。
procost	real	估算的执行成本。
prorows	real	估算的影响行的数目。
provariadic	oid	参数元素的数据类型。
protransform	regproc	此函数的简化调用方式。
proisagg	boolean	函数是否是聚集函数。 ● t（true）：表示是。 ● f（false）：表示不是。
proiswindow	boolean	函数是否是窗口函数。 ● t（true）：表示是。 ● f（false）：表示不是。
prosecdef	boolean	函数是否是一个安全定义器（也就是一个“setuid”函数）。 ● t（true）：表示是。 ● f（false）：表示不是。
proleakproof	boolean	函数是否没有副作用。如果函数没有对参数进行防泄露处理，则会抛出错误。 ● t（true）：表示没副作用。 ● f（false）：表示有副作用。
proisstrict	boolean	● t（true）：使用该函数时，如果入参有空值，则函数实际上不调用直接返回空。 ● f（false）：使用该函数时，即使入参有空值，也要调用。所以该函数必须处理空输入。
proretset	boolean	函数返回值是否为一个集合（也就是说，指定数据类型的多个数值）。 ● true：函数返回值是一个集合。 ● false：函数返回值不是一个集合。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
provolatile	"char"	该函数的结果是否只依赖于它的输入参数，或者还会被外界因素影响。 ● i：“不可变的”（immutable）函数，对于相同的输入总是产生相同的结果。 ● s：“稳定的”（stable）函数，（对于固定输入）其结果在一次扫描里不变。 ● v：“易变”（volatile）函数，其结果可能在任何时候变化，也用于那些有副作用的函数，因此调用它们无法得到优化。
pronargs	smallint	参数数目。
pronargdefaults	smallint	有默认值的参数数目。
prorettype	oid	返回值的数据类型。
proargtypes	oidvector	一个存放函数参数的数据类型的数组。数组里只包括输入参数（包括INOUT参数），代表该函数的调用签名（接口）。
proallargtypes	oid[]	一个包含函数参数的数据类型的数组。数组里包括所有参数的类型（包括OUT和INOUT参数），如果所有参数都是IN参数，则这个字段就会是空。请注意数组下标是以1为起点的，而因为历史原因，proargtypes的下标起点为0。
proargmodes	"char"[]	一个保存函数参数模式的数组，编码如下： ● i：表示IN参数。 ● o：表示OUT参数。 ● b：表示INOUT参数。 ● v：表示VARIADIC参数。 如果所有参数都是IN参数，则这个字段为空。请注意，下标对应的是proallargtypes的位置，而不是proargtypes。
proargnames	text[]	一个保存函数参数的名称的数组。没有名称的参数在数组里设置为空字符串。如果没有一个参数有名称，这个字段将是空。请注意，此数组的下标对应proallargtypes而不是proargtypes。
proargdefaults	pg_node_tree	默认值的表达式树。是PRONARGDEFAULTS元素的列表。
prosrc	text	描述函数或存储过程的定义。例如，对于解释型语言来说就是函数的源程序，或者一个连接符号，一个文件名，或者函数和存储过程创建时指定的其他任何函数体内容，具体取决于语言/调用习惯的实现。
probin	text	关于如何调用该函数的附加信息。同样，其含义也是和语言相关的。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
proconfig	text[]	函数针对运行时配置变量的本地设置。
proacl	aclitem[]	访问权限。具体请参见**GRANT**和**REVOKE**。
prodefaultargpos	int2vector	函数具有默认值的入参的位置。
fencedmode	boolean	函数的执行模式，表示函数是在fence还是not fence模式下执行。 ● true：fence模式，函数的执行会在重新fork的进程中执行。 ● false：not fence模式。 系统内建函数，fencedmode字段均为false，即not fence模式。
proshippable	boolean	表示该函数是否可以下推到DN上执行，默认值是false。 ● 对于IMMUTABLE类型的函数，函数始终可以下推到DN上执行。 ● 对于STABLE/VOLATILE类型的函数，仅当函数的属性是SHIPPABLE的时候，函数可以下推到DN执行。
propackage	boolean	表示该函数是否支持重载，默认值是false。 ● t（true）：表示支持。 ● f（false）：表示不支持。
prokind	"char"	表示该对象为函数还是存储过程。 ● 'f'：表示该对象为函数。 ● 'p'：表示该对象为存储过程。
proargsrc	text	描述兼容ORA语法定义的函数或存储过程的参数输入字符串，包括参数注释。默认值为NULL。
proisprivate	boolean	描述函数是否是PACKAGE内的私有函数，默认为false。
propackageid	oid	函数所属的package oid，如果不在package内，则为0。
proargtypesext	oidvector_extend	当函数参数较多时，用来存放函数参数的数据类型的数组。数组里只包括输入参数（包括INOUT参数），代表该函数的调用签名（接口）。
prodefaultargposext	int2vector_extend	当函数参数较多时，函数具有默认值的入参的位置。
allargtypes	oidvector	用来存放存储过程参数的数据类型的数组，包含存储过程所有参数（入参、出参、INOUT参数）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
allargtypesext	oidvector_extend	当函数参数较多（大于666个）时，用来存放存储过程参数的数据类型的数组，包含所有参数（入参、出参、INOUT参数）。

说明

新建函数时，会向pg_proc表中插入数据，更新索引。当出入参个数很多时，索引的长度可能会超过页面的三分之一，进而可能会产生“Index row size xxx exceeds maximum xxx for index "pg_proc_proname_all_args_nsp_index"”的报错，此为预期情况。您可以通过减少参数个数，避免该报错的产生。

7.2.18.59 PG_RANGE

PG_RANGE系统表存储关于范围类型的信息，除了**PG_TYPE**里类型的记录。

表 7-132 PG_RANGE 字段

名称	类型	引用	描述
rngtypid	oid	**PG_TYPE**.oid	范围类型的OID。
rngsubtype	oid	**PG_TYPE**.oid	这个范围类型的元素类型（子类型）的OID。
rngcollation	oid	**PG_COLLATION**.oid	用于范围比较的排序规则的OID，如果没有则为零。
rngsubopc	oid	**PG_OPCLASS**.oid	用于范围比较的子类型的操作符类的OID。
rngcanonical	regproc	**PG_PROC**.proname	转换范围类型为规范格式的函数名，如果没有则为0。
rngsubdiff	regproc	**PG_PROC**.proname	表示用于计算范围两元素差值的函数的名字，该函数的返回值为双精度类型，如果不存在该函数则rngsubdiff字段值为0。

若元素类型是离散的，则rngcanonical决定用于范围类型的排序顺序；若元素类型是不可排序的，则rngsubopc决定用于范围类型的排序顺序；若元素类型是可排序的，则rngsubopc和rngcollation决定用于范围类型的排序顺序。

7.2.18.60 PG_REPLICATION_ORIGIN

PG_REPLICATION_ORIGIN系统表包含所有已创建的复制源，该表在一个集群的所有数据库之间共享，即每个集群只有一份，而不是每个数据库一份。

版权所有 © 华为云计算技术有限公司

表 7-133 PG_REPLICATION_ORIGIN 字段

名称	类型	描述
roident	oid	一个集群范围内唯一的复制源标识符。
roname	text	外部的由用户定义的复制源名称。

7.2.18.61 PG_RESOURCE_POOL

PG_RESOURCE_POOL系统表提供了数据库资源池的信息。

表 7-134 PG_RESOURCE_POOL 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
respool_name	name	资源池名称。
mem_percent	integer	内存配置的百分比。
cpu_affinity	bigint	CPU绑定core的数值。
control_group	name	资源池所在的control group名称。
active_statements	integer	保留字段。
max_dop	integer	只用于扩容的接口，表示数据重分布时，扫描并发度。
memory_limit	name	资源池最大的内存。
parentid	oid	父资源池OID。
io_limits	integer	每秒触发I/O的次数上限。单位是万次/秒。
io_priority	name	I/O利用率高达90%时，重消耗I/O作业进行I/O资源管控时关联的优先级等级。
nodegroup	name	表示资源池所在的node_group的名称。
is_foreign	boolean	表示资源池是否用于控制逻辑数据库之外的用户。 ● true：表示资源池用来控制不属于当前资源池的普通用户的资源。 ● false：表示不控制不属于当前资源池的普通用户的资源。
max_worker	integer	只用于扩容的接口，表示扩容数据重分布时，表内插入并发度。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_connections	integer	最大连接数，用来限制资源池可使用的最大连接数。
max_dynamic_memory	name	资源池允许使用的最大动态内存值。
max_shared_memory	name	资源池允许使用的最大共享内存值。
max_concurrency	integer	资源池允许使用的最大并发数。

7.2.18.62 PG_REWRITE

PG_REWRITE系统表存储表和视图定义的重写规则。

表 7-135 PG_REWRITE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
rulename	name	规则名称。
ev_class	oid	使用这条规则的表名称。
ev_attr	smallint	这条规则适用的字段（目前总是为零，表示整个表）。
ev_type	"char"	规则适用的事件类型。 ● 1：SELECT ● 2：UPDATE ● 3：INSERT ● 4：DELETE
ev_enabled	"char"	用于控制复制的触发。 ● O：“origin”和“local”模式时触发。 ● D：禁用触发。 ● R：“replica”时触发。 ● A：任何模式是都会触发。
is_instead	boolean	如果该规则是INSTEAD规则，则为真，否则为假。
ev_qual	pg_node_tree	规则的资格条件的表达式树（以nodeToString() 形式存在）。
ev_action	pg_node_tree	规则动作的查询树（以nodeToString() 形式存在）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
ev_action_def	text	保存视图定义时的原始SELECT语句。

7.2.18.63 PG_SET

PG_SET系统表存储SET数据类型定义的元数据。

当前分布式下暂不支持该功能。

7.2.18.64 PG_SHDEPEND

PG_SHDEPEND系统表记录数据库对象和共享对象（比如角色）之间的依赖性关系。这些信息允许GaussDB保证在企图删除这些对象之前，这些对象是没有被引用的。

与**PG_DEPEND**的作用类似，只是它是用于在一个数据库内部的对象的依赖性关系的。

PG_SHDEPEND是在集群的所有数据库之间共享的，即每个集群只有一个，而不是每个数据库一个。

表 7-136 PG_SHDEPEND 字段

名称	类型	引用	描述
dbid	oid	**PG_DATABASE**.oid	依赖对象所在的数据库OID，如果是共享对象，则为零。
classid	oid	**PG_CLASS**.oid	依赖对象所在的系统表的OID。
objid	oid	任意OID属性	指定的依赖对象的OID。
objsubid	integer	-	对于一个表字段，这是字段号（根据objid和classid可以确定指向的表）。对于所有其他对象类型，这个字段为零。
refclassid	oid	**PG_CLASS**.oid	被引用对象所在的系统表的OID(必须是一个共享表)。
refobjid	oid	任意OID属性	指定的被引用对象的OID。
deptype	"char"	-	一段代码，定义了这个依赖性关系的特定语义。 取值范围为o、a、p、d、l。
objfile	text	-	用户定义C函数库文件路径。

在任何情况下，一条PG_SHDEPEND记录就表明这个被引用的对象不能在未删除依赖对象的前提下删除。不过，deptype同时还标出了几种不同的子风格：

- SHARED_DEPENDENCY_OWNER (o)
  被引用的对象（必须是一个角色）是依赖对象的所有者。
- SHARED_DEPENDENCY_ACL (a)

版权所有 © 华为云计算技术有限公司

被引用的对象（必须是一个角色）在依赖对象的ACL（访问控制列表，也就是权限列表）里提到。SHARED_DEPENDENCY_ACL不会在对象的所有者上添加，因为所有者会有一个SHARED_DEPENDENCY_OWNER记录。

- SHARED_DEPENDENCY_PIN (p)

  这类记录标识系统自身依赖于该被依赖对象，因此这样的对象不能被删除。这种类型的记录只是由initdb创建。这样的依赖对象的字段都是零。

- SHARED_DEPENDENCY_ DBPRIV(d)

  被引用的对象（必须是一个角色）具有依赖对象所对应的ANY权限（指定的依赖对象的OID对应的是系统表12.2.10.1 GS_DB_PRIVILEGE中一行）。

- SHARED_DEPENDENCY_ SECLABEL(l)

  被引用的对象（必须是一个安全标签）被应用在依赖对象上。

7.2.18.65 PG_SHDESCRIPTION

PG_SHDESCRIPTION系统表为共享数据库对象存储可选的注释。可以使用COMMENT命令操作注释的内容，使用\d命令查看注释内容。

PG_DESCRIPTION提供了类似的功能，它记录了单个数据库中对象的注释。

PG_SHDESCRIPTION是在集群的所有数据库之间共享的，即每个集群只有一个，而不是每个数据库一个。

表 7-137 PG_SHDESCRIPTION 字段

名称	类型	引用	描述
objoid	oid	任意OID属性	所描述的对象的OID。
classoid	oid	PG_CLASS.oid	该对象出现的系统表的OID。
description	text	-	该对象的描述信息。

7.2.18.66 PG_STATISTIC

PG_STATISTIC系统表存储有关该数据库中表和索引列的统计数据。默认只有系统管理员权限才可以访问此系统表，普通用户需要授权才可以访问。

表 7-138 PG_STATISTIC 字段

名称	类型	描述
starelid	oid	所描述字段所属的表或者索引。
starelkind	"char"	所属对象的类型。
staattnum	smallint	所描述字段在表中的编号，从1开始。
stainherit	boolean	是否统计有继承关系的对象。
stanullfrac	real	所描述字段中为NULL的记录的比率。
stawidth	integer	所描述字段非NULL记录的平均存储宽度，以字节计。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stadistinct	real	标识全局统计信息中所有DN上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
stakindN	smallint	stakindN记录统计信息的类型，N是一个编码，表示这种类型的统计存储在pg_statistic行的第N个“槽位”。 N的取值范围：1～5，如stakind1、stakind2或stakind3等。 针对stakindN，不同编号代表的不同统计信息类型对应如下： ● 1：MCV ● 2：直方图 ● 3：相关性 ● 4：MCELEM（常见元素，针对数组使用） ● 5：不同元素计数直方图（针对数组使用）
staopN	oid	一个用于生成这些存储在第N个“槽位”的统计信息的操作符。比如，一个柱面图槽位会显示<操作符，该操作符定义了该数据的排序顺序。 N的取值范围：1～5
stanumbersN	real[]	第N个“槽位”的相关类型的数值类型统计，如果该槽位和数值类型没有关系，则就是NULL。 N的取值范围：1～5
stavaluesN	anyarray	第N个“槽位”类型的字段数据值，如果该槽位类型不存储任何数据值，则就是NULL。每个数组的元素值实际上都是指定字段的数据类型，因此，除了把这些字段的类型定义成anyarray之外，没有更好的办法。 N的取值范围：1～5
stadndistinct	real	标识DN1上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadndistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
staextinfo	text	统计信息的扩展信息。预留字段。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stastate	"char"	标识统计信息是否被锁定，若被锁定则无法更新。该参数目前在分布式下无实际意义。 ● l：标识已经被锁定。 ● u：标识没有被锁定。

须知

PG_STATISTIC系统表存储了统计对象的一些敏感信息，如高频值MCV。系统管理员和授权后的其他用户可以通过访问PG_STATISTIC系统表，查询到统计对象的这些敏感信息。

7.2.18.67 PG_STATISTIC_EXT

PG_STATISTIC_EXT系统表存储有关该数据库中表的扩展统计数据，包括多列统计数据和表达式统计数据（后续支持）。由用户指定收集哪些扩展统计数据。需要有系统管理员权限才可以访问此系统表。

表 7-139 PG_STATISTIC_EXT 字段

名称	类型	描述
starelid	oid	所描述字段所属的表或者索引。
starelkind	"char"	所属对象的类型。'c'表示表，'p'表示分区。
stainherit	boolean	是否统计有继承关系的对象。 ● t（true）：表示统计。 ● f（false）：表示不统计。
stanullfrac	real	所描述字段中为NULL的记录的比率。
stawidth	integer	所描述字段非NULL记录的平均存储宽度，以字节计。
stadistinct	real	标识全局统计信息中所有DN上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stadndistinct	real	标识DN1上字段里唯一的非NULL数据值的数目。 ● 一个大于零的数值是独立数值的实际数目。 ● 一个小于零的值是distinct值所占总行数的比例，比如stadndistinct=-0.5时，它的实际distinct值是总行数*0.5。 ● 零值表示独立数值的数目未知。
stakindN	smallint	stakindN记录统计信息的类型，N是一个编码，表示这种类型的统计存储在pg_statistic行的第N个“槽位”。 N的取值范围：1～5，如stakind1、stakind2或stakind3等。 针对stakindN，不同编号代表的不同统计信息类型对应如下： ● 1：MCV ● 6：带有空值的MCV ● 8：贝叶斯网络
staopN	oid	一个用于生成这些存储在第N个“槽位”的统计信息的操作符。比如，一个柱面图槽位会显示<操作符，该操作符定义了该数据的排序顺序。 N的取值范围：1～5
stakey	int2vector	所描述的字段编号的数组。
stanumbersN	real[]	第N个“槽位”的相关类型的数值类型统计，如果该槽位和数值类型没有关系，则就是NULL。 N的取值范围：1～5
stavaluesN	anyarray	第N个“槽位”类型的字段数据值，如果该槽位类型不存储任何数据值，则就是NULL。每个数组的元素值实际上都是指定字段的数据类型，因此，除了把这些字段的类型定义成anyarray之外，没有更好的办法。 N的取值范围：1～5
staexprs	pg_node_tree	扩展统计信息对应的表达式。
stasource	"char"	扩展统计信息的来源： ● 'a'：表示自动创建，详见GUC参数auto_statistic_ext_columns。 ● 'm'：表示手动创建，用户通过analyze tablename ((column list))或者alter table tablename add statistics ((column list))来创建。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stastatus	"char"	扩展统计信息的状态： ● 'a'：表示活跃可用。 ● 'd'：表示被禁用，相关信息不被收集，优化器在生成计划的时候也不使用。用户可以使用语法alter table tablename disable/enable statistics((column list))来修改扩展统计信息的状态。
stastate	"char"	标识统计信息是否被锁定，若被锁定则无法更新。该参数目前在分布式下无实际意义。 ● l：标识已经被锁定。 ● u：标识没有被锁定。
staextname	name	多列统计信息多列组的别名。

须知

PG_STATISTIC_EXT系统表存储了统计对象的一些敏感信息，如高频值MCV。系统管理员和授权后的其他用户可以通过访问PG_STATISTIC_EXT系统表，查询到统计对象的这些敏感信息。

7.2.18.68 PG_SYNONYM

PG_SYNONYM系统表存储同义词对象名与其他数据库对象名间的映射信息。

表 7-140 PG_SYNONYM 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
synname	name	同义词名称。
synnamespace	oid	包含该同义词的名字空间的OID。PUBLIC同义词的名字空间的OID为0。
synowner	oid	同义词的所有者，通常是创建它的用户OID。PUBLIC同义词的所有者为0。
synobjschema	name	关联对象指定的模式名。
synobjname	name	关联对象名。
syndblinkname	name	DATABASE LINK对象名。

7.2.18.69 PG_TABLESPACE

PG_TABLESPACE系统表存储表空间信息。

版权所有 © 华为云计算技术有限公司

表 7-141 PG_TABLESPACE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
spcname	name	表空间名称。
spcowner	oid	表空间的所有者，通常是创建它的人。
spcacl	aclitem[]	访问权限。具体请参见**GRANT**和**REVOKE**。
spcoptions	text[]	表空间的选项。
spcmaxsize	text	可使用的最大磁盘空间大小，单位Byte。
relative	boolean	标识表空间指定的存储路径是否为相对路径。 • t（true）：表示是。 • f（false）：表示不是。

7.2.18.70 PG_TRIGGER

PG_TRIGGER系统表存储触发器信息。

表 7-142 PG_TRIGGER 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
tgrelid	oid	触发器所在表的OID。
tgname	name	触发器名。
tgfoid	oid	要被触发器调用的函数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tgtype	smallint	触发器类型。 ● 4：AFTER INSERT FOR EACH STATEMENT。 ● 5：AFTER INSERT FOR EACH ROW。 ● 6：BEFORE INSERT FOR EACH STATEMENT。 ● 7：BEFORE INSERT FOR EACH ROW。 ● 8：AFTER DELETE FOR EACH STATEMENT。 ● 9：AFTER DELETE FOR EACH ROW。 ● 10：BEFORE DELETE FOR EACH STATEMENT。 ● 11：BEFORE DELETE FOR EACH ROW。 ● 16：AFTER UPDATE FOR EACH STATEMENT。 ● 17：AFTER UPDATE FOR EACH ROW。 ● 18：BEFORE UPDATE FOR EACH STATEMENT。 ● 19：BEFORE UPDATE FOR EACH ROW。 ● 32：AFTER TRUNCATE FOR EACH STATEMENT。 ● 34：BEFORE TRUNCATE FOR EACH STATEMENT。 ● 69：INSTEAD OF INSERT FOR EACH ROW。 ● 73：INSTEAD OF DELETE FOR EACH ROW。 ● 81：INSTEAD OF UPDATE FOR EACH ROW。
tgenabled	"char"	● O：表示触发器在“origin”和“local”模式下触发。 ● D：表示触发器被禁用。 ● R：表示触发器在“replica”模式下触发。 ● A：表示触发器始终触发。
tgisinternal	boolean	内部触发器标识，如果为true表示内部触发器。
tgconstrrelid	oid	完整性约束引用的表。
tgconstrindid	oid	完整性约束的索引。
tgconstraint	oid	约束触发器在**PG_CONSTRAINT**中的OID。
tgdeferrable	boolean	约束触发器是否为DEFERRABLE类型。 ● t（true）：表示是。 ● f（false）：表示不是。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tginitdeferred	boolean	约束触发器是否为INITIALLY DEFERRED类型。 • t（true）：表示是。 • f（false）：表示不是。
tgnargs	smallint	触发器函数入参个数。
tgattr	int2vector	当触发器指定列时的列号，未指定则为空数组。
tgargs	bytea	传递给触发器的参数。
tgqual	pg_node_tree	表示触发器的WHEN条件，如果没有则为null。
tgowner	oid	触发器所有者。

7.2.18.71 PG_TS_CONFIG

PG_TS_CONFIG系统表包含表示文本搜索配置的记录。其中的每一条配置包含一个特定的文本搜索解析器和字典映射的列表。

解析器在PG_TS_CONFIG记录中显示，但是字典映射的标记是由PG_TS_CONFIG_MAP里面的辅助记录定义的。

表 7-143 PG_TS_CONFIG 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
cfgname	name	-	文本搜索配置名。
cfgnamespace	oid	PG_NAMESPACE.oid	包含这个配置的名称空间的OID。
cfgowner	oid	PG_AUTHID.oid	配置的所有者。
cfgparser	oid	PG_TS_PARSER.oid	这个配置的文本搜索解析器的OID。
cfoptions	text[]	-	分词相关配置选项。

7.2.18.72 PG_TS_CONFIG_MAP

PG_TS_CONFIG_MAP系统表包含每个文本搜索配置的解析器符号映射关系。

版权所有 © 华为云计算技术有限公司

表 7-144 PG_TS_CONFIG_MAP 字段

名称	类型	引用	描述
mapcfg	oid	PG_TS_CONFIG.oid	拥有这个映射记录的PG_TS_CONFIG记录的OID。
maptokentype	integer	-	由配置的解析器产生的一个符号类型值。
mapseqno	integer	-	在相同mapcfg或maptokentype值的情况下，该符号类型的顺序号。
mapdict	oid	PG_TS_DICT.oid	要咨询的文本搜索字典的OID。

7.2.18.73 PG_TS_DICT

PG_TS_DICT系统表包含定义文本搜索字典的记录。字典取决于文本搜索模板，该模板声明所有需要的实现函数；字典本身提供模板支持的用户可设置的参数的值。

这种分工允许字典通过非权限用户创建。参数由文本字符串dictinitoption指定，参数的格式和意义取决于模板。

表 7-145 PG_TS_DICT 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
dictname	name	-	文本搜索字典名。
dictnamespace	oid	PG_NAMESPACE.oid	包含字典的名称空间的OID。
dictowner	oid	PG_AUTHID.oid	字典的所有者。
dicttemplate	oid	PG_TS_TEMPLATE.oid	字典的文本搜索模板的OID。
dictinitoption	text	-	该模板的初始化选项字符串。

7.2.18.74 PG_TS_PARSER

PG_TS_PARSER系统表包含文本解析器的定义。解析器负责将输入文本分割为词位，并且为每个词位分配标记类型。因为解析器必须通过C语言级别的函数实现，所以新解析器必须由数据库系统管理员创建。

版权所有 © 华为云计算技术有限公司

表 7-146 PG_TS_PARSER 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
prsname	name	-	文本搜索解析器名。
prsnamespace	oid	PG_NAMESPACE.oid	包含解析器的名称空间的OID。
prsstart	regproc	PG_PROC.proname	解析器的启动函数名。
prstoken	regproc	PG_PROC.proname	解析器的下一个标记函数名。
prsend	regproc	PG_PROC.proname	解析器的关闭函数名。
prsheadline	regproc	PG_PROC.proname	解析器的标题函数名。
prslextype	regproc	PG_PROC.proname	解析器的lextype函数名。

7.2.18.75 PG_TS_TEMPLATE

PG_TS_TEMPLATE系统表包含定义文本搜索模板的记录。模板是文本搜索字典的类的实现框架。因为模板必须通过C语言级别的函数实现，索引新模板的创建必须由数据库系统管理员创建。

表 7-147 PG_TS_TEMPLATE 字段

名称	类型	引用	描述
oid	oid	-	行标识符（隐含字段，必须明确选择）。
tmplname	name	-	文本搜索模板名。
tmplnamespace	oid	PG_NAMESPACE.oid	包含这个模板的名称空间的OID。
tmplinit	regproc	PG_PROC.proname	模板的初始化函数名。
tmpllexize	regproc	PG_PROC.proname	模板的lexize函数名。

7.2.18.76 PG_TYPE

PG_TYPE系统表存储数据类型的相关信息。

版权所有 © 华为云计算技术有限公司

表 7-148 PG_TYPE 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
typname	name	数据类型名称。
typnamespace	oid	包含此类型的名称空间的OID。
typowner	oid	该类型的所有者。
typlen	smallint	对于定长类型是该类型内部表现形式的字节数目。对于变长类型是负数。 ● -1：表示一种“变长”（有长度字属性的数据）。 ● -2：表示这是一个NULL结尾的C字符串。
typbyval	boolean	● true：指定内部传递此类型的数值时是传值。 ● false：指定内部传递此类型的数值时是传引用。 如果该类型的typlen不是1、2、4、8， typbyval建议传引用，也可以传值。变长类型通常是传引用，也可以传值。
typtype	"char"	● b：基础类型。 ● c：复合类型（比如，一个表的行类型）。 ● d：域类型。 ● p：伪类型。 ● r：范围类型。 ● e：枚举类型。 ● u：未定义类型。 ● o：集合类型。 参见typrelid和typbasetype。
typcategory	"char"	是数据类型的模糊分类，可用于解析器作为数据转换的依据。
typispreferred	boolean	● true：数据符合typcategory所指定的转换规则时进行转换。 ● false：不进行转换。
typisdefined	boolean	● true：表示类型已定义。 ● false：表示是一种尚未定义的类型的占位符，此时，除了该类型的名称、名称空间和OID之外没有可靠的信息。
typdelim	"char"	当分析数组输入时，分隔两个此类型数值的字符请注意该分隔符是与数组元素数据类型相关联的，而不是和数组数据类型关联。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
typrelid	oid	如果是复合类型（请参见typtype），则这个字段指向PG_CLASS中定义该表的行。对于自由存在的复合类型，pg_class记录并不表示一个表，但是总需要它来查找该类型连接的PG_ATTRIBUTE记录。对于非复合类型为零。
typelem	oid	如果不为0，则它标识pg_type里面的另外一行。当前类型可以当做一个产生类型为typelem的数组来描述。一个“真正的”数组类型是变长的（typlen= -1），但是一些定长的（typlen > 0）类型也拥有非零的typelem（比如name和point）。如果一个定长类型拥有一个typelem，则他的内部形式必须是typelem数据类型的某个数目的个数值，不能有其它数据。变长数组类型有一个该数组子过程定义的头（文件）。
typarray	oid	如果不为0，则表示在pg_type中有对应的类型记录。
typinput	regproc	输入转换函数（文本格式）。
typoutput	regproc	输出转换函数（文本格式）。
typreceive	regproc	输入转换函数（二进制格式），如果没有则为0。
typsend	regproc	输出转换函数（二进制格式），如果没有则为0。
typmodin	regproc	输入类型修改符函数，如果没有则为0。
typmodout	regproc	输出类型修改符函数，如果没有则为0。
typanalyze	regproc	自定义的ANALYZE函数，如果使用标准函数，则为0。
typalign	"char"	当存储此类型的数值时要求的对齐性质。它应用于磁盘存储以及该值的大多数形式。如果数值是连续存放的，比如在磁盘上以完全的裸数据的形式存放时，则先在此类型的数据前填充空白，这样它就可以按照要求的界限存储。对齐引用是该序列中第一个数据的开头。可能的值包含： • c：表示char对齐，也就是不需要对齐。 • s：表示short对齐（在大多数机器上是2字节）。 • i：表示int对齐（在大多数机器上是4字节）。 • d：表示double对齐（在大多数机器上是8字节，但不一定是全部）。 **须知** 对于在系统表里使用的类型，在pg_type里定义的尺寸和对齐必须和编译器在一个表示表的一行的结构里的布局一样。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
typstorage	"char"	指定字段的数据存储方式，具体包含以下几个类型： • p：普通存储（plain）。数值总是以简单方式存储。 • e：外部存储（external）。数据实际存储在主表以外的地方（如TOAST表），主表存储的是指向该外部存储数据的引用。 • m：主体存储（main）。该方式与“普通存储”类似，可以以内联的压缩方式存储。通常用于存储需要自定义存储方式的类型，如自定义的复合类型。 • x：扩展存储（extended）。值可以被压缩和/或移动到一个二级关系表中。 x一般是TOAST类型存储的方式。
typnotnull	boolean	该类型是否存在NOTNULL约束。目前只用于域。
typbasetype	oid	如果这是一个衍生类型（请参见typtype），则该标识作为这个类型的基础的类型。如果不是衍生类型则为零。
typtypmod	integer	域使用typtypmod记录要作用到它们的基础类型上的typmod（如果基础类型不使用typmod则为-1）。如果这种类型不是域，则为-1。 如果该类型是数组类型，则typtypmod是数组类型的最大容量。 如果该类型是带索引的集合类型，则typtypmod是该集合类型的索引最大长度。
typndims	integer	如果一个域是数组，则typndims是数组维数的数值。非域非数组域为零。
typcollation	oid	指定类型的排序规则，取值参考**PG_COLLATION**系统表。如果为0，则表示不支持排序。
typdefaultbin	pg_node_tree	如果为非NULL，则它是该类型缺省表达式的nodeToString() 表现形式。目前这个字段只用于域。
typdefault	text	如果某类型没有相关缺省值，则取值是NULL。如果typdefaultbin不是NULL，则typdefault必须包含一个typdefaultbin代表的缺省表达式。如果typdefaultbin为NULL但typdefault不是，typdefault则是该类型缺省值的外部表现形式，可以把它交给该类型的输入转换器生成一个常量。
typacl	aclitem[]	访问权限。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
typelemmod	integer	• -1：集合和数组类型对应元素类型不需要typmod。 • >=0的数值：集合和数组类型对应元素类型的typmod值。 • NULL：集合和数组类型对应元素类型的typmod未知。 typmod通常指类型的最大长度。

7.2.18.77 PGXC_CLASS

PGXC_CLASS系统表存储每张表的复制或分布信息。

表 7-149 PGXC_CLASS 字段

名称	类型	描述
pcrelid	oid	表的OID。
pclocatortype	"char"	定位器类型。 • H：hash • G：Range • L：List • M：Modulo • N：Round Robin • R：Replication • U：MurmurHash
pchashalgorithm	smallint	使用哈希算法分布元组。 • 1：默认的哈希算法。 • 2：MURMURHASH算法。
pchashbuckets	smallint	哈希容器的值。
pgroup	name	节点群的名称。
redistributed	"char"	表已经完成重分布。
redis_order	integer	重分布的顺序。该值等于0的表在本轮重分布过程中不进行重分布。
pcattnum	int2vector	用作分布键的列标号。
nodeoids	oidvector_extend	表分布的节点OID列表。
options	text	系统内部保留字段，存储扩展状态信息。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
diskey	text[]	SQL建表语句中，DISTRIBUTE BY子句括号中分布列表达式的字符串内容。目前只支持MURMURHASH分布。
diskeyexprs	pg_node_tree	diskey通过pack_node_to_string()函数序列化得到的字符串信息。

7.2.18.78 PGXC_GROUP

PGXC_GROUP系统表存储节点组信息。

表 7-150 PGXC_GROUP 字段

名称	类型	描述
oid	oid	行标识符（隐含字段，必须明确选择）。
group_name	name	节点组名称。
in_redistribution	"char"	是否需要重分布。取值包括： ● n：表示NodeGroup没有再进行重分布。 ● y：表示NodeGroup是重分布过程中的源节点组。 ● t：表示NodeGroup是重分布过程中的目的节点组。
group_members	oidvector_extend	节点组的节点OID列表。
group_buckets	text	分布数据桶的集合。
is_installation	boolean	是否安装子集群。 ● t（true）：表示安装。 ● f（false）：表示不安装。
group_acl	aclitem[]	访问权限。
group_kind	"char"	node group类型，取值包括： ● i：表示installation node group。 ● n：表示普通集群node group。 ● e：表示弹性集群。 ● s：表示系统内置单DN节点集群。
group_parent	oid	如果是子node group，该字段表示父node group的OID，如果是父node group，该字段值为空。
bucket_map	text	物理bucket与逻辑bucket的映射关系。

版权所有 © 华为云计算技术有限公司

7.2.18.79 PGXC_SLICE

PGXC_SLICE表是针对range范围分布和list分布创建的系统表，用来记录分布具体信息，当前不支持range interval自动扩展分片，不过在系统表中预留。

表 7-151 PGXC_SLICE 字段

名称	类型	描述
relname	name	表名或者分片名，通过type区分。
type	"char"	● 't'：relname是表名。 ● 's'：relname是分片的名字。
strategy	"char"	● 'r'：为range分布表。 ● 'l'：为list分布表。 后续interval分片会扩展该值。
relid	oid	该tuple隶属的分布表oid。
referenceoid	oid	所参考分布表的oid，用于slice reference建表语法。
sindex	integer	当为list分布表时，表示当前boundary在某个分片内的位置。
interval	text[]	预留字段。
transitboundary	text[]	预留字段。
transitno	integer	预留字段。
nodeoid	oid	当relname为分片名时，表示该分片的数据存放在哪一个DN上，nodeoid表示这个DN的oid。
boundaries	text[]	当relname为分片名时，对应该分片的边界值。
specified	boolean	当前分片对应的DN是否是用户在DDL中显示指定的。
sliceorder	integer	用户定义分片的顺序。

7.2.18.80 PLAN_TABLE_DATA

PLAN_TABLE_DATA存储了用户通过执行EXPLAIN PLAN收集到的计划信息。与PLAN_TABLE视图不同的是PLAN_TABLE_DATA表存储了所有session和user执行EXPLAIN PLAN收集的计划信息。

版权所有 © 华为云计算技术有限公司

表 7-152 PLAN_TABLE_DATA 字段

名称	类型	描述
session_id	text	表示插入该条数据的会话，由服务线程启动时间戳和服务线程ID组成。受非空约束限制。
user_id	oid	用户ID，用于标识触发插入该条数据的用户。受非空约束限制。
statement_id	character varying(30)	用户输入的查询标签。
plan_id	bigint	查询计划标识。该标识在计划生成阶段自动产生，供内核工程师调试使用。
id	integer	计划中的节点编号。
operation	character varying(30)	操作描述。
options	character varying(255)	操作选项。
object_name	name	操作对应的对象名，来自于用户定义。
object_type	character varying(30)	对象类型。
object_owner	name	对象所属schema，来自于用户定义。
projection	character varying(4000)	操作输出的列信息。
cost	double precision	优化器对算子估算的执行代价。
cardinality	double precision	优化器对算子估算的结果行数。

说明

- PLAN_TABLE_DATA中包含了当前节点所有用户、所有会话的数据，仅管理员有访问权限。普通用户可以通过**PLAN_TABLE**视图查看属于自己的数据。
- 对于不活跃（已退出）的会话，其在PLAN_TABLE_DATA中的数据会在一定时间（默认5min）后被gs_clean清理。用户也可以手动执行gs_clean -C选项对表中不活跃的会话数据进行清理。
- PLAN_TABLE_DATA中的数据是用户通过执行EXPLAIN PLAN命令后由系统自动插入表中，因此禁止用户手动对数据进行插入或更新，否则会引起表中的数据混乱。需要对表中数据删除时，建议通过**PLAN_TABLE**视图来实现。
- statement_id、object_name、object_owner和projection字段内容遵循用户定义的大小写存储，其它字段内容采用大写存储。

版权所有 © 华为云计算技术有限公司

7.2.18.81 STATEMENT_HISTORY

STATEMENT_HISTORY获得当前节点的执行语句的信息。查询系统表必须具有sysadmin权限。只可在系统库中查询到结果，用户库中无法查询。

对于此系统表查询有如下约束：

- 必须在postgres库内查询，其它库中不存数据。
- 此系统表受track_stmt_stat_level控制，默认为"OFF,L0"，第一部分控制Full SQL，第二部分控制Slow SQL，具体字段记录级别见下表。考虑性能影响，更改该参数的值时建议通过set方式设置，使该参数仅对当前会话生效。
- 当track_stmt_stat_level中Full SQL和Slow SQL不全为OFF时，对于超过log_min_duration_statement阈值的SQL，is_slow_sql字段会记录为true，表明这是一个超过阈值的SQL，其他字段以track_stmt_stat_level中的级别为记录标准。

表 7-153 STATEMENT_HISTORY 字段

名称	类型	描述	最低记录级别
db_name	name	数据库名称。	L0
schema_name	name	模式名称。	L0
origin_node	integer	节点名称。	L0
user_name	name	用户名。	L0
application_name	text	用户发起的请求的应用程序名称。	L0
client_addr	text	用户发起的请求的客户端地址。	L0
client_port	integer	用户发起的请求的客户端端口。	L0
unique_query_id	bigint	归一化SQL ID。	L0
debug_query_id	bigint	唯一SQL ID。部分语句存在不唯一的情况，如Parse报文、DCL和TCL等语句的debug_query_id值为0。存储过程内的语句使用同一个ID。	L0
query	text	归一化SQL(仅在CN上有值)，track_stmt_parameter参数开启时，显示完整SQL。	L0
start_time	timestamp with time zone	语句启动的时间。	L0
finish_time	timestamp with time zone	语句结束的时间。	L0

版权所有 © 华为云计算技术有限公司

名称	类型	描述	最低记录级别
slow_sql_threshold	bigint	语句执行时慢SQL的标准。	L0
transaction_id	bigint	事务ID。	L0
thread_id	bigint	执行线程ID。	L0
session_id	bigint	用户session id。	L0
n_soft_parse	bigint	软解析次数, n_soft_parse + n_hard_parse可能大于n_calls, 因为子查询未计入n_calls。（该字段对非DML语句无意义）	L0
n_hard_parse	bigint	硬解析次数, n_soft_parse + n_hard_parse可能大于n_calls, 因为子查询未计入n_calls。对复杂数据类型的解析也会额外增加解析次数，例如smalldatetime。（该字段对非DML语句无意义）	L0
query_plan	text	语句执行计划。	L1(Full SQL) L0(Slow SQL)
n_returned_rows	bigint	SELECT返回的结果集行数。	L0
n_tuples_fetched	bigint	随机扫描行，即回表次数。回表是指在使用索引进行查询时，如果查询的列不在索引中，数据库引擎需要通过索引中的主键或唯一标识符再次访问主表以获取所需的完整数据行的过程。	L0
n_tuples_returned	bigint	顺序扫描行，通过顺序扫描获取的活动行数与通过索引扫描返回的索引条目数之和。	L0
n_tuples_inserted	bigint	插入行。	L0
n_tuples_updated	bigint	更新行。	L0
n_tuples_deleted	bigint	删除行。	L0

版权所有 © 华为云计算技术有限公司

名称	类型	描述	最低记录级别
n_blocks_fetched	bigint	buffer的块访问次数，表示从缓冲区和磁盘中读取block的总数。	L0
n_blocks_hit	bigint	buffer的块命中次数，表示从缓冲区中读取block的数量。	L0
db_time	bigint	有效的DB时间花费，多线程将累加（单位：微秒）。	L0
cpu_time	bigint	CPU消耗时间（单位：微秒）。	L0
execution_time	bigint	执行器内执行时间（单位：微秒）。	L0
parse_time	bigint	SQL解析时间（单位：微秒）。（若解析阶段出现异常，不记录parse_time）	L0
plan_time	bigint	SQL生成计划时间（单位：微秒）。（若计划生成阶段出现异常，不记录plan_time）	L0
rewrite_time	bigint	SQL重写时间（单位：微秒）。（若重写阶段出现异常，不记录rewrite_time）	L0
pl_execution_time	bigint	plpgsql上的执行时间（单位：微秒）。	L0
pl_compilation_time	bigint	plpgsql上的编译时间（单位：微秒）。	L0
data_io_time	bigint	IO上的时间花费（单位：微秒）。	L0
net_send_info	text	通过物理连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。	L0
net_recv_info	text	通过物理连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。	L0

版权所有 © 华为云计算技术有限公司

名称	类型	描述	最低记录级别
net_stream_send_info	text	通过逻辑连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。	L0
net_stream_recv_info	text	通过逻辑连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。	L0
lock_count	bigint	加锁次数。	L0
lock_time	bigint	加锁耗时。	L1
lock_wait_count	bigint	加锁等待次数。	L0
lock_wait_time	bigint	加锁等待耗时。	L1
lock_max_count	bigint	最大持锁数量。	L0
lwlock_count	bigint	轻量级加锁次数（预留）。	L0
lwlock_wait_count	bigint	轻量级加锁等待次数。	L0
lwlock_time	bigint	轻量级加锁时间（预留）。	L1
lwlock_wait_time	bigint	轻量级加锁等待耗时。	L1

版权所有 © 华为云计算技术有限公司

名称	类型	描述	最低记录级别
details	bytea	等待时间、语句锁事件和GSTrace函数运行时间的列表。 记录级别的值>=L0时，开始记录等待事件的列表。显示当前节点wait event的相关统计信息。内核中关键的事件信息请参见**表7-470**、**表7-471**、**表7-472**和**表7-473**，或从视图wait_event_info中查看系统中所有的事件列表。关于每种事务锁对业务的影响程度，请参考**LOCK**语法小节的详细描述。通过GStrace全链路跟踪函数，GStrace面板中的function字段表示函数名，以树形结构表达函数父子关系；GStrace面板中的duration字段表示函数运行时间，单位为微秒。 记录级别的值是L2时，开始记录语句锁事件的列表，该列表按时间顺序记录事件，记录的锁事件数量受参数track_stmt_details_size的影响。 该字段为二进制，需要借助解析函数pg_catalog.statement_detail_decode进行读取，请参见（**表1-204**）。 事件包括： ● 加锁开始 LOCK_START ● 加锁结束 LOCK_END ● 等锁开始 LOCK_WAIT_START ● 等锁结束 LOCK_WAIT_END ● 释放锁 LOCK_RELEASE ● 轻量级等锁开始 LWLOCK_WAIT_START ● 轻量级等锁结束 LWLOCK_WAIT_END	L0/L2
is_slow_sql	boolean	该SQL是否为Slow SQL。 ● t（true）：表示是。 ● f（false）：表示不是。	L0
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。	L0

版权所有 © 华为云计算技术有限公司

名称	类型	描述	最低记录级别
advise	text	可能导致该SQL为Slow SQL的风险信息。 **说明** • 如果语句通过SQLBypass方式执行（可通过explain查看计划判断），可以认为无慢SQL风险，本字段信息可能会缺失。 • 本字段长度限制256字符，如果超过256字符会被截断，先根据已有的提示信息进行SQL优化。 • advise涉及表别名的以表别名展示，涉及列别名的以列的原名展示。	L0
parent_unique_sql_id	bigint	当前语句的外层SQL的归一化SQL ID，对于存储过程内执行的语句，该值为调用存储过程语句的归一化SQL ID，存储过程外的语句该值为0。	L0
finish_status	text	语句完成状态。 • normal：表示正常。默认为normal。 • cancelled：表示被取消或被terminate。 • error：表示除cancelled外，其他异常导致的结束。 **说明** 当语句完成状态不为normal时，表示语句执行异常，其他字段均无参考价值。	L0
used_memory	integer	单条语句内存使用量（单位：MB）。	L0
lock_max_local_count	bigint	经过locallock机制持锁的最大持锁数量。	L0
lock_max_fastpath_count	bigint	经过fastpath机制持锁的最大持锁数量。	L0
lock_max_global_count	bigint	经过主锁表持锁的最大持锁数量。	L0
sql_hash	bigint	当前SQL语句的哈希值。	L0
plan_hash	bigint	当前SQL计划的哈希值。	L0
plan_hash_prev	bigint	如果是Generic Plan，此值为上一次使用的计划的哈希值。	L0
driver_start_time	timestamp with time zone	JDBC驱动开始执行的时间点。	L0
driver_wait_response	timestamp with time zone	JDBC驱动发送报文后的开始等待接收数据库返回数据的时间点。	L0

版权所有 © 华为云计算技术有限公司

名称	类型	描述	最低记录级别
driver_finish_time	timestamp with time zone	JDBC驱动接收数据库返回数据完成的时间点。	L0
driver_info	json	L0级别默认记录为"{}"。 L1级别开始，记录JDBC驱动各报文发送和接收的时间点。 参数track_stmt_stat_level修改后，会在新建立连接的会话生效，现有会话不生效。	L1
kernel_info	json	数据库各报文发送和接收的时间点。	L0
adaptive_join_states	text	记录自适应连接算子最终选用的连接方式，记录格式为[plan_node_id, use_hashjoin]，第一位代表对应的plan_node_id，第二位代表是否使用HashJoin，1代表是，0代表否。	L0
aplan_count	bigint	aplan执行次数。	L0
aplan_parse_time	bigint	aplan解析时间。	L0
aplan_execution_time	bigint	aplan执行时间。	L0
relids	text	语句中涉及表的pg_class.oid，多表按“,”隔开。	L0

版权所有 © 华为云计算技术有限公司

说明

- query_plan字段的记录级别对于Full SQL为L1，对于Slow SQL为L0。
- 对于db_time时间模型，在statement_history中各维度时间统计满足db_time >= max(cpu_time, parse_time, plan_time, rewrite_time, data_io_time, net_send_info.time, net_recv_info.time, net_stream_send_info.time, net_stream_recv_info.time)。
- statement_history中的query字段来自unique sql hash table，获取query时会产生lwlock等待事件UniqueSQLMappingLock，此时db_time已经结束，该段时间不计入db_time中，因此当instr_unique_sql_count设置较大时可能出现lwlock_wait_time大于db_time的情况。
- 对于db_time时间模型公式，如下维度不纳入统计范围：
  - execution_time
  - pl_execution_time
  - pl_compilation_time
- 等待事件的嵌套深度最大值是20，如超过最大深度，新的等待事件会覆盖最后一次等待事件。
- 由于自适应连接不支持分布式，分布式下adaptive_join_states不生效。
- 由于查询计划管理（SQL Plan Management）不支持分布式，分布式下sql_hash、plan_hash和plan_hash_prev不生效。
- DN上执行的存储过程内语句的统计信息不汇聚。
- 存储过程内语句CPU_TIME值为0。
- 分布式存储过程子语句锁相关指标不会向顶层语句进行汇聚。
- 当查询计划记录失败时，会在query_plan字段中提示记录失败的原因，如以下场景：
  - 在LightProxy场景下语句会下推，不会在CN节点上记录计划，会在DN节点上记录。提示：“The current plan is lightproxy. Please check the query_plan on the DN node.”。
  - 在Stream场景下计划会下推，会在CN节点上记录计划，不会在DN节点上记录。提示：“The current plan is streaming. Please check the query_plan on the CN node.”。
  - 对于DML语句自动提交的COMMIT，不会记录相关计划。提示：“The current record is a transaction commit of a DML statement. Please check the plan of this DML statement.”。
  - 非DML/DQL语句不记录计划。提示：“Do not record plans for non-DML/DQL statements.”。对于DDL语句，在多CN集群上，query_plan会记录节点间的交互信息。
  - 存储过程中未适配或没有计划的子语句，例如简单表达式、OPEN CURSOR、CLOSE以及BYPASS路径语句，不记录计划。存储过程中执行的子语句不在DN上记录计划。提示：“Statement has no plan or is unsupported for plan recording.”。
  - 对于进入到PARSE阶段的SQL，还未生成计划，将不进行记录。提示：“This statement is a PARSE statement and has not yet generated a query plan.”。
  - 在检查是否需要上报query_plan时，SQL语句执行并不慢，后续清理等任务影响导致SQL语句整体执行慢。提示：“Check the reported query_plan time is 2026-01-01 08:05:01.735619+08, at this time the SQL is not slow.”。
  - 语句执行状态异常时，计划记录有可能失败，需要检查日志分析原因。提示：“An error occurred during statement execution. Please check the log for analysis.”。
  - 其他未记录成功的场景，需要检查日志分析原因。提示：“The query plan failed to be recorded. Please check the log analysis.”。

7.2.18.82 STREAMING_STREAM

STREAMING_STREAM系统表存储所有STREAM对象的元数据信息。

 版权所有 © 华为云计算技术有限公司

表 7-154 STREAMING_STREAM 字段

名称	类型	描述
relid	oid	STREAM对象的标识。
queries	bytea	该STREAM对应CONTVIEW的位图映射。

7.2.18.83 STREAMING_CONT_QUERY

STREAMING_CONT_QUERY系统表存储所有CONTVIEW对象的元数据信息。

表 7-155 STREAMING_CONT_QUERY 字段

名称	类型	描述
id	integer	CONTVIEW对象唯一的标识符，不可重复。
type	"char"	标识CONTVIEW的类型。 ● 'r'：表示该CONTVIEW是基于行存存储模型。
relid	oid	CONTVIEW对象的标识。
defrelid	oid	CONTVIEW对应的持续计算规则VIEW的标识。
active	boolean	标识CONTVIEW是否处于持续计算状态。 ● t（true）：表示是。 ● f（false）：表示不是。
streamrelid	oid	CONTVIEW对应的STREAM的标识。
matrelid	oid	CONTVIEW对应物化表的标识。
lookupidxid	oid	CONTVIEW对应GROUP LOOK UP INDEX的标识，此字段内部使用，仅行存具有。
step_factor	smallint	标识CONTVIEW的步进模式。主要取值为0（无重叠窗口）和1（滑动窗口，步长为1）。
ttl	integer	CONTVIEW设置的ttl_interval参数值。
ttl_attno	smallint	CONTVIEW设置的TTL功能对应时间列的字段编号。
dictrelid	oid	CONTVIEW对应字典表的标识。
grpnum	smallint	CONTVIEW持续计算规则中维度列的个数，此字段内部使用。
grpidx	int2vector	CONTVIEW持续计算规则中维度列在TARGET LIST的索引，此字段内部使用。

7.3 系统视图

版权所有 © 华为云计算技术有限公司

7.3.1 分区表

7.3.1.1 ADM_IND_PARTITIONS

ADM_IND_PARTITIONS视图显示数据库中所有一级分区表Local索引的索引分区信息。数据库中每个一级分区表的Local索引的索引分区（如果存在的话）都会在ADM_IND_PARTITIONS里有一行记录。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-156 ADM_IND_PARTITIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引分区所属分区表索引的所有者的名称。
index_name	character varying(64)	索引分区所属分区表索引的名称。
partition_name	character varying(64)	索引分区的名称。
def_tablespace_name	name	索引分区的表空间名称。
high_value	text	索引分区所对应分区的上边界。 ● 对于范围分区，显示各分区的上边界值。 ● 对于列表分区，显示各分区的取值列表。 ● 对于哈希分区，显示各分区的编号。
index_partition_usable	boolean	索引分区是否可用。 ● t（true）：表示可用。 ● f（false）：表示不可用。
schema	character varying(64)	索引分区所属分区表索引的模式。
high_value_length	integer	索引分区所对应分区的边界的字符长度。
composite	character varying(3)	索引是否属于二级分区表上的本地索引，该表不存储二级分区信息，所以该值为NO。
subpartition_count	numeric	分区中的二级分区数，该表不存储二级分区信息，所以该值为0。
partition_position	numeric	索引分区在索引中的位置。
status	character varying(8)	索引分区是否可用。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tablespace_name	name	分区所在表空间的名称。
pct_free	numeric	块中最小可用空间百分比。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对索引的更改。
compression	character varying(13)	分区索引是否启用索引压缩。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	根据索引的值表示表中行的顺序。需要通过执行analyze进行统计。
num_rows	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	分区的实际缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
pct_direct_access	numeric	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	索引分区段是否已创建。
orphaned_entries	character varying(3)	暂不支持，值为NULL。

7.3.1.2 ADM_IND_SUBPARTITIONS

ADM_IND_SUBPARTITIONS视图显示数据库中所有二级分区表Local索引的索引分区信息（不包含分区表全局索引）。数据库中每个二级分区表的Local索引的索引分区（如果存在的话）都会在ADM_IND_SUBPARTITIONS里有一行记录。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-157 ADM_IND_SUBPARTITIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引分区所属分区表索引的所有者的名称。
index_name	character varying(64)	索引分区所属分区表索引的名称。
partition_name	character varying(64)	索引所在一级分区的名称。
subpartition_name	character varying(64)	索引所在二级分区的名称。
def_tablespace_name	name	索引分区的表空间名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
high_value	text	索引分区所对应分区的边界值。 ● 对于范围分区，显示各分区的上边界值。 ● 对于列表分区，显示各分区的取值列表。 ● 对于哈希分区，显示各分区的编号。
index_partition_usable	boolean	索引分区是否可用。 ● t（true）：可用。 ● f（false）：不可用。
schema	character varying(64)	索引分区所属分区表索引的模式。
high_value_length	integer	索引分区所对应分区的边界的字符长度。
partition_position	numeric	索引分区在索引中的位置。
subpartition_position	numeric	二级分区在分区中的位置。
status	character varying(8)	索引分区是否可用。
tablespace_name	name	索引分区的表空间名称。
pct_free	numeric	块中最小可用空间百分比。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对索引的更改。
compression	character varying(13)	用于二级分区的压缩类型。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	根据索引的值表示表中行的顺序。需要通过执行analyze进行统计。
num_rows	numeric	二级分区中的行数。需要通过执行vacuum进行统计。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	二级分区的缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	索引分区段是否已创建。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。

7.3.1.3 ADM_PART_COL_STATISTICS

ADM_PART_COL_STATISTICS视图显示数据库中所有表分区的列统计信息和直方图信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-158 ADM_PART_COL_STATISTICS 字段

名称	类型	描述
owner	character varying(128)	分区表的所有者。
table_name	character varying(128)	表名。
partition_name	character varying(128)	表分区名称。
column_name	character varying(4000)	列名。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的低值。
high_value	raw	列中的高值。
density	numeric	• 如果列上有直方图，则该字段将显示直方图中跨越少于2个端点的值的选择性。它不代表跨越2个或更多端点的值的选择性。 • 如果列上没有可用的直方图，则该字段的值为1/NUM_DISTINCT。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
sample_size	numeric	用于分析此列的样本量。
last_analyzed	date	最近分析此列的日期。数据库重启后，数据会丢失。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
notes	character varying(63)	暂不支持，值为NULL。
avg_col_len	numeric	列的平均长度（以字节为单位）。
histogram	character varying(15)	直方图是否存在以及存在的类型： • NONE：不存在直方图。 • FREQUENCY：频率直方图。 • EQUI-WIDTH：等宽直方图。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schema	character varying(64)	列所属的名称空间的名称。

说明

在ORA兼容模式下，初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea，若指定GUC参数mapping_date_to_datea的值为off，则该视图的last_analyzed字段类型会变更为timestamp(0) without time zone。

7.3.1.4 ADM_PART_INDEXES

ADM_PART_INDEXES视图显示数据库中所有分区表索引的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-159 ADM_PART_INDEXES 字段

名称	类型	描述
def_tablespace_name	name	分区表索引的表空间名称。
index_owner	character varying(64)	分区表索引的所有者名称。
index_name	character varying(64)	分区表索引的名称。
partition_count	bigint	分区表索引的索引分区的个数。
partitioning_key_count	integer	分区表的分区键个数。
partitioning_type	text	分区表的分区策略。 **说明** 当前分区表策略支持范围见**CREATE TABLE PARTITION**。
schema	character varying(64)	分区表索引所属模式名称。
table_name	character varying(64)	分区表索引所属的分区表名称。
subpartitioning_type	text	二级分区表的分区策略。如果分区表是一级分区表，则显示NONE。 **说明** 当前二级分区表策略支持范围见**CREATE TABLE SUBPARTITION**。
def_subpartition_count	integer	默认创建二级分区的个数。二级分区表为1，一级分区表为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
subpartitioning_key_count	integer	分区表二级分区键的个数。

7.3.1.5 ADM_PART_TABLES

ADM_PART_TABLES视图显示数据库中所有分区表的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-160 ADM_PART_TABLES 字段

名称	类型	描述
table_owner	character varying(64)	分区表的所有者名称。
table_name	character varying(64)	分区表的名称。
partitioning_type	text	分区表的分区策略。 **说明** 当前分区表策略支持范围见**CREATE TABLE PARTITION**。
partition_count	bigint	分区表的分区个数。
partitioning_key_count	integer	分区表的分区键个数。
def_tablespace_name	name	分区表的表空间名称。
schema	character varying(64)	分区表的模式。
subpartitioning_type	text	二级分区表的分区策略。如果分区表是一级分区表，则显示NONE。
def_subpartition_count	integer	默认创建二级分区的个数，二级分区表为1，一级分区表为0。
subpartitioning_key_count	integer	分区表二级分区键的个数。
status	character varying(8)	暂不支持，值为valid。
def_pct_free	numeric	添加分区时使用的PCTFREE默认值。
def_pct_used	numeric	暂不支持，值为NULL。
def_ini_trans	numeric	添加分区时使用的INITRANS默认值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
def_max_trans	numeric	添加分区时使用的MAXTRANS默认值。
def_initial_extent	character varying(40)	暂不支持，值为NULL。
def_next_extent	character varying(40)	暂不支持，值为NULL。
def_min_extents	character varying(40)	暂不支持，值为NULL。
def_max_extents	character varying(40)	暂不支持，值为NULL。
def_max_size	character varying(40)	暂不支持，值为NULL。
def_pct_increase	character varying(40)	暂不支持，值为NULL。
def_freelists	numeric	暂不支持，值为NULL。
def_freelist_groups	numeric	暂不支持，值为NULL。
def_logging	character varying(7)	暂不支持，值为NULL。
def_compression	character varying(8)	添加分区时使用的默认压缩： • NONE • ENABLED • DISABLED
def_compress_for	character varying(30)	添加分区时使用的默认压缩。 **说明** 可用的压缩方法和压缩级别见**•WITH ({ storage_parameter = value } [, ...])**。
def_buffer_pool	character varying(7)	暂不支持，值为DEFAULT。
def_flash_cache	character varying(7)	暂不支持，值为NULL。
def_cell_flash_cache	character varying(7)	暂不支持，值为NULL。
ref_ptn_constraint_name	character varying(128)	暂不支持，值为NULL。
interval	character varying(1000)	区间值字符串。
autolist	character varying(3)	暂不支持，值为NO。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
interval_subpartition	character varying(1000)	暂不支持，值为NULL。
autolist_subpartition	character varying(3)	暂不支持，值为NO。
is_nested	character varying(3)	暂不支持，值为NO。
def_segment_creation	character varying(4)	暂不支持段页式设置，当启用segment时，值为YES。
def_indexing	character varying(3)	暂不支持，值为ON。
def_inmemory	character varying(8)	暂不支持，值为NONE。
def_inmemory_priority	character varying(8)	暂不支持，值为NULL。
def_inmemory_distribute	character varying(15)	暂不支持，值为NULL。
def_inmemory_compression	character varying(17)	暂不支持，值为NULL。
def_inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
def_read_only	character varying(3)	暂不支持，值为NO。
def_cellmemory	character varying(24)	暂不支持，值为NULL。
def_inmemory_service	character varying(12)	暂不支持，值为NULL。
def_inmemory_service_name	character varying(1000)	暂不支持，值为NULL。

7.3.1.6 ADM_SUBPART_KEY_COLUMNS

ADM_SUBPART_KEY_COLUMNS视图显示数据库中所有的二级分区表或分区索引的分区键列的相关信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。分布式暂不支持二级分区表，该视图所有字段值为null。

版权所有 © 华为云计算技术有限公司

表 7-161 ADM_SUBPART_KEY_COLUMNS 字段

名称	类型	描述
owner	character varying(128)	二级分区表或索引的拥有者。
name	character varying(128)	二级分区表名或索引名。
object_type	character(5)	对象类型。 ● 若分区为二级分区表，此列为table。 ● 若分区为二级分区索引，此列为index。
column_name	character varying(4000)	二级分区表的二级分区键列名。
column_position	numeric	列在分区中的位置。
collated_column_id	numeric	暂不支持，值为NULL。

7.3.1.7 ADM_TAB_PARTITIONS

ADM_TAB_PARTITIONS视图显示数据库下所有的一级分区信息（包括二级分区表）。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-162 ADM_TAB_PARTITIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	关系表名称。
partition_name	character varying(64)	分区名称。
high_value	text	分区的边界值。 ● 对于范围分区，显示各分区的上边界值。 ● 对于列表分区，显示各分区的取值列表。 ● 对于哈希分区，显示各分区的编号。
tablespace_name	name	分区表的表空间名称。
schema	character varying(64)	名称空间的名称。
composite	character varying(3)	表是否为二级分区表。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
subpartition_count	bigint	二级分区的个数。
high_value_length	integer	分区边界值表达式的长度。
partition_position	numeric	分区在表中的位置。
pct_free	numeric	块中可用空间的最小百分比。
pct_used	numeric	暂不支持，值为NULL。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对表的更改。
compression	character varying(8)	表分区的实际压缩属性。
compress_for	character varying(30)	暂不支持，值为NULL。
num_rows	numeric	分区中的行数。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	用于分区块的缓冲池。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
is_nested	character varying(3)	这是否是嵌套表分区。
parent_table_partition	character varying(128)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(4)	表分区是否创建了段。
indexing	character varying(4)	暂不支持，值为NULL。
read_only	character varying(4)	暂不支持，值为NULL。
inmemory	character varying(8)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。
inmemory_service_name	character varying(100)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.1.8 ADM_TAB_SUBPARTITIONS

ADM_TAB_SUBPARTITIONS视图显示数据库下所有的二级分区信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-163 ADM_TAB_SUBPARTITIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	关系表名称。
partition_name	character varying(64)	分区名称。
subpartition_name	character varying(64)	二级分区名称。
high_value	text	二级分区的边界值。 • 对于范围分区，显示各分区的上边界值。 • 对于列表分区，显示各分区的取值列表。 • 对于哈希分区，显示各分区的编号。
tablespace_name	name	二级分区表的表空间名称。
schema	character varying(64)	名称空间的名称。
high_value_length	integer	二级分区的边界值的字符长度。

7.3.1.9 DB_IND_PARTITIONS

DB_IND_PARTITIONS视图显示当前用户所能访问的一级分区表Local索引的索引分区信息（不包含分区表全局索引）。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-164 DB_IND_PARTITIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引分区所属分区表索引的所有者的名称。
index_name	character varying(64)	索引分区所属分区表索引的名称。
partition_name	character varying(64)	索引分区的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
def_tablespace_name	name	索引分区的表空间名称。
high_value	text	索引分区所对应分区的上边界。 ● 对于范围分区，显示各分区的上边界值。 ● 对于列表分区，显示各分区的取值列表。 ● 对于哈希分区，显示各分区的编号。
index_partition_usable	boolean	索引分区是否可用。 ● t（true）：表示可用。 ● f（false）：表示不可用。
schema	character varying(64)	索引分区所属分区表索引的模式。
high_value_length	integer	索引分区所对应分区的边界的字符长度。
composite	character varying(3)	索引是否属于二级分区表上的本地索引，该表不存储二级分区信息，所以该值为NO。
subpartition_count	numeric	分区中的二级分区数，该表不存储二级分区信息，所以该值为0。
partition_position	numeric	索引分区在索引中的位置。
status	character varying(8)	索引分区是否可用。
tablespace_name	name	分区所在表空间的名称。
pct_free	numeric	块中最小可用空间百分比。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对索引的更改。
compression	character varying(13)	分区索引是否启用索引压缩。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	根据索引的值指示表中行的顺序。需要通过执行analyze进行统计。
num_rows	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	分区的实际缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
pct_direct_access	numeric	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	索引分区段是否已创建。
orphaned_entries	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.1.10 DB_IND_SUBPARTITIONS

DB_IND_SUBPARTITIONS视图显示当前用户所能访问的二级分区表Local索引的索引分区信息（不包含分区表全局索引）。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-165 DB_IND_SUBPARTITIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引分区所属分区表索引的所有者的名称。
index_name	character varying(64)	索引分区所属分区表索引的名称。
partition_name	character varying(64)	索引所在一级分区的名称。
subpartition_name	character varying(64)	索引所在二级分区的名称。
def_tablespace_name	name	索引分区的表空间名称。
high_value	text	索引分区所对应分区的边界值。
index_partition_usable	boolean	索引分区是否可用。 ● t（true）：可用。 ● f（false）：不可用。
schema	character varying(64)	索引分区所属分区表索引的模式。
high_value_length	integer	索引分区所对应分区的边界的字符长度。
partition_position	numeric	索引分区在索引中的位置。
subpartition_position	numeric	二级分区在分区中的位置。
status	character varying(8)	索引分区是否可用。
tablespace_name	name	索引分区的表空间名称。
pct_free	numeric	块中最小可用空间百分比。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对索引的更改。
compression	character varying(13)	用于二级分区的压缩类型。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	根据索引的值表示表中行的顺序。需要通过执行analyze进行统计。
num_rows	numeric	二级分区中的行数。需要通过执行vacuum进行统计。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	二级分区的缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
segment_created	character varying(3)	索引分区段是否已创建。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。

7.3.1.11 DB_PART_COL_STATISTICS

DB_PART_COL_STATISTICS视图显示当前用户可访问的表分区的列统计信息和直方图信息。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-166 DB_PART_COL_STATISTICS 字段

名称	类型	描述
owner	character varying(128)	分区表的所有者。
table_name	character varying(128)	表名。
partition_name	character varying(128)	表分区名称。
column_name	character varying(4000)	列名。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的低值。
high_value	raw	列中的高值。
density	numeric	• 如果列上有直方图，则该字段将显示直方图中跨越少于2个端点的值的选择性。它不代表跨越2个或更多端点的值的选择性。 • 如果列上没有可用的直方图，则该字段的值为1/NUM_DISTINCT。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
sample_size	numeric	用于分析此列的样本量。
last_analyzed	date	最近分析此列的日期。数据库重启后，数据会丢失。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
notes	character varying(63)	暂不支持，值为NULL。
avg_col_len	numeric	列的平均长度（以字节为单位）。
histogram	character varying(15)	直方图是否存在以及存在的类型： • NONE：不存在直方图。 • FREQUENCY：频率直方图。 • EQUI-WIDTH：等宽直方图。
schema	character varying(64)	列所属的名称空间的名称。

说明

在ORA兼容模式下，初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea，若指定GUC参数mapping_date_to_datea的值为off，则该视图的last_analyzed字段类型会变更为timestamp(0) without time zone。

7.3.1.12 DB_PART_INDEXES

DB_PART_INDEXES视图显示当前用户所能访问的分区表索引的信息（不包含分区表全局索引）。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-167 DB_PART_INDEXES 字段

名称	类型	描述
def_tablespace_name	name	分区表索引的表空间名称。
index_owner	character varying(64)	分区表索引的所有者名称。
index_name	character varying(64)	分区表索引的名称。
partition_count	bigint	分区表索引的索引分区的个数。
partitioning_key_count	integer	分区表的分区键个数。
partitioning_type	text	分区表的分区策略。 **说明** 当前分区表策略支持范围见**CREATE TABLE PARTITION**。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schema	character varying(64)	分区表索引所属模式名称。
table_name	character varying(64)	分区表索引所属的分区表名称。
subpartitioning_type	text	二级分区表的分区策略。如果分区表是一级分区表，则显示NONE。 **说明** 当前二级分区表策略支持范围见CREATE TABLE SUBPARTITION。
def_subpartition_count	integer	默认创建二级分区的个数，二级分区表为1，一级分区表为0。
subpartitioning_key_count	integer	分区表二级分区键的个数。

7.3.1.13 DB_PART_KEY_COLUMNS

DB_PART_KEY_COLUMNS视图显示了当前用户可访问的分区表或分区索引的分区键列的相关信息。该视图所有用户可访问，显示当前用户可访问的所有信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-168 DB_PART_KEY_COLUMNS 字段

名称	类型	描述
owner	character varying(128)	分区表或索引的拥有者。
name	character varying(128)	分区表名或索引名。
object_type	character(5)	对象类型。 ● 若分区为分区表，此列为table。 ● 若分区为分区索引，此列为index。
column_name	character varying(4000)	分区表或索引的键列名。
column_position	numeric	列在分区中的位置。
collated_column_id	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.1.14 DB_PART_TABLES

DB_PART_TABLES视图显示当前用户所能访问的分区表的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-169 DB_PART_TABLES 字段

名称	类型	描述
table_owner	character varying(64)	分区表的所有者名称。
table_name	character varying(64)	分区表的名称。
partitioning_type	text	分区表的分区策略。 **说明** 当前分区表策略支持范围见**CREATE TABLE PARTITION**。
partition_count	bigint	分区表的分区个数。
partitioning_key_count	integer	分区表的分区键个数。
def_tablespace_name	name	分区表的表空间名称。
schema	character varying(64)	分区表的模式。
subpartitioning_type	text	二级分区表的分区策略。如果分区表是一级分区表，则显示NONE。
def_subpartition_count	integer	默认创建二级分区的个数，二级分区表为1，一级分区表为0。
subpartitioning_key_count	integer	分区表二级分区键的个数。
status	character varying(8)	暂不支持，值为valid。
def_pct_free	numeric	添加分区时使用的PCTFREE默认值。
def_pct_used	numeric	暂不支持，值为NULL。
def_ini_trans	numeric	添加分区时使用的INITRANS默认值。
def_max_trans	numeric	添加分区时使用的MAXTRANS默认值。
def_initial_extent	character varying(40)	暂不支持，值为NULL。
def_next_extent	character varying(40)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
def_min_extents	character varying(40)	暂不支持，值为NULL。
def_max_extents	character varying(40)	暂不支持，值为NULL。
def_max_size	character varying(40)	暂不支持，值为NULL。
def_pct_increase	character varying(40)	暂不支持，值为NULL。
def_freelists	numeric	暂不支持，值为NULL。
def_freelist_groups	numeric	暂不支持，值为NULL。
def_logging	character varying(7)	暂不支持，值为NULL。
def_compression	character varying(8)	添加分区时使用的默认压缩： • NONE - Not specified • ENABLED • DISABLED
def_compress_for	character varying(30)	添加分区时使用的默认压缩。 **说明** 可用的压缩方法和压缩级别见**•WITH ({ storage_parameter = value } [, ...])**。
def_buffer_pool	character varying(7)	暂不支持，值为DEFAULT。
def_flash_cache	character varying(7)	暂不支持，值为NULL。
def_cell_flash_cache	character varying(7)	暂不支持，值为NULL。
ref_ptn_constraint_name	character varying(128)	暂不支持，值为NULL。
interval	character varying(1000)	区间值字符串。
autolist	character varying(3)	暂不支持，值为NO。
interval_subpartition	character varying(1000)	暂不支持，值为NULL。
autolist_subpartition	character varying(3)	暂不支持，值为NO。
is_nested	character varying(3)	暂不支持，值为NO。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
def_segment_creation	character varying(4)	暂不支持段页式设置，当启用segment时，赋YES值。
def_indexing	character varying(3)	暂不支持，值为ON。
def_inmemory	character varying(8)	暂不支持，值为NONE。
def_inmemory_priority	character varying(8)	暂不支持，值为NULL。
def_inmemory_distribute	character varying(15)	暂不支持，值为NULL。
def_inmemory_compression	character varying(17)	暂不支持，值为NULL。
def_inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
def_read_only	character varying(3)	暂不支持，值为NO。
def_cellmemory	character varying(24)	暂不支持，值为NULL。
def_inmemory_service	character varying(12)	暂不支持，值为NULL。
def_inmemory_service_name	character varying(1000)	暂不支持，值为NULL。

7.3.1.15 DB_SUBPART_KEY_COLUMNS

DB_SUBPART_KEY_COLUMNS视图显示了当前用户可访问的二级分区表或分区索引的分区键列的相关信息。该视图所有用户可访问，显示当前用户可访问的所有信息。该视图同时存在于PG_CATALOG和SYS Schema下。分布式暂不支持二级分区表，该视图所有字段值为NULL。

表 7-170 DB_SUBPART_KEY_COLUMNS 字段

名称	类型	描述
owner	character varying(128)	二级分区表或索引的拥有者。
name	character varying(128)	二级分区表名或索引名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
object_type	character(5)	对象类型。 ● 若分区为分区表，此列为table。 ● 若分区为分区索引，此列为index。
column_name	character varying(4000)	二级分区表的二级分区键列名。
column_position	numeric	列在分区中的位置。
collated_column_id	numeric	暂不支持，值为NULL。

7.3.1.16 DB_TAB_PARTITIONS

DB_TAB_PARTITIONS视图显示当前用户所能访问的一级分区信息（包括二级分区表）。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-171 DB_TAB_PARTITIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表的名称。
partition_name	character varying(64)	分区名称。
high_value	text	分区的边界值。 ● 对于范围分区，显示各分区的上边界值。 ● 对于列表分区，显示各分区的取值列表。 ● 对于哈希分区，显示各分区的编号。
tablespace_name	name	分区表的表空间名称。
schema	character varying(64)	名称空间的名称。
composite	character varying(3)	表是否为二级分区表。
subpartition_count	bigint	二级分区的个数。
high_value_length	integer	分区边界值表达式的长度。
partition_position	numeric	分区在表中的位置。
pct_free	numeric	块中可用空间的最小百分比。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pct_used	numeric	暂不支持，值为NULL。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对表的更改。
compression	character varying(8)	表分区的实际压缩属性。
compress_for	character varying(30)	暂不支持，值为NULL。
num_rows	numeric	分区中的行数。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	用于分区块的缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
is_nested	character varying(3)	是否是嵌套表分区。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
parent_table_partition	character varying(128)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(4)	表分区是否创建了段。
indexing	character varying(4)	暂不支持，值为NULL。
read_only	character varying(4)	暂不支持，值为NULL。
inmemory	character varying(8)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。
inmemory_service_name	character varying(100)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。

7.3.1.17 DB_TAB_SUBPARTITIONS

DB_TAB_SUBPARTITIONS视图显示当前用户所能访问的二级分区表的信息。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-172 DB_TAB_SUBPARTITIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	关系表名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
partition_name	character varying(64)	分区名称。
subpartition_name	character varying(64)	二级分区名称。
high_value	text	二级分区的边界值。 • 对于范围分区，显示各分区的上边界值。 • 对于列表分区，显示各分区的取值列表。 • 对于哈希分区，显示各分区的编号。
tablespace_name	name	二级分区表的表空间名称。
schema	character varying(64)	名称空间的名称。
high_value_length	integer	二级分区边界值的字符长度。
partition_position	numeric	分区在表中的位置。
subpartition_position	numeric	二级分区在分区中的位置。
pct_free	numeric	块中可用空间的最小百分比。
pct_used	numeric	暂不支持，值为NULL。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(3)	是否记录对表的更改。
compression	character varying(8)	表分区的实际压缩属性。
compress_for	character varying(30)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
num_rows	numeric	二级分区中的行数。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	二级分区的缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	表二级分区是否创建了段。
indexing	character varying(3)	暂不支持，值为NULL。
read_only	character varying(3)	暂不支持，值为NULL。
inmemory	character varying(8)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。
inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。

7.3.1.18 MY_IND_PARTITIONS

MY_IND_PARTITIONS视图显示当前用户下的一级分区表Local索引的索引分区信息（不包含分区表全局索引）。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-173 MY_IND_PARTITIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引分区所属分区表索引的所有者的名称。
index_name	character varying(64)	索引分区所属分区表索引的名称。
partition_name	character varying(64)	索引分区的名称。
def_tablespace_name	name	索引分区的表空间名称。
high_value	text	索引分区所对应分区的上边界。 • 对于范围分区，显示各分区的上边界值。 • 对于列表分区，显示各分区的取值列表。 • 对于哈希分区，显示各分区的编号。
index_partition_usable	boolean	索引分区是否可用。 • t（true）：表示可用。 • f（false）：表示不可用。
schema	character varying(64)	索引分区所属分区表索引的模式。
high_value_length	integer	索引分区所对应分区的边界的字符长度。
composite	character varying(3)	索引是否属于二级分区表上的本地索引，该表不存储二级分区信息，所以该值为NO。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
subpartition_count	numeric	分区中的二级分区数，该表不存储二级分区信息，所以该值为0。
partition_position	numeric	索引分区在索引中的位置。
status	character varying(8)	指示索引分区是否可用。
tablespace_name	name	分区所在表空间的名称。
pct_free	numeric	块中最小可用空间百分比。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对索引的更改。
compression	character varying(13)	分区索引是否启用索引压缩。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	根据索引的值指示表中行的顺序。需要通过执行analyze进行统计。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
num_rows	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。数据库重启后数据会丢失。
buffer_pool	character varying(7)	分区的实际缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
pct_direct_access	numeric	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	索引分区段是否已创建。
orphaned_entries	character varying(3)	暂不支持，值为NULL。

7.3.1.19 MY_IND_SUBPARTITIONS

MY_IND_SUBPARTITIONS描述了当前用户拥有的二级分区表Local索引的索引分区信息（不包含分区表全局索引）。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-174 MY_IND_SUBPARTITIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引分区所属分区表索引的所有者的名称。
index_name	character varying(64)	索引分区所属分区表索引的名称。
partition_name	character varying(64)	索引所在一级分区的名称。
subpartition_name	character varying(64)	索引所在二级分区的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
def_tablespace_name	name	索引分区的表空间名称。
high_value	text	索引分区所对应分区的边界值。
index_partition_usable	boolean	索引分区是否可用： ● t（true）：表示可用。 ● f（false）：表示不可用。
schema	character varying(64)	索引分区所属分区表索引的模式。
high_value_length	integer	索引分区所对应分区的边界的字符长度。
partition_position	numeric	索引分区在索引中的位置。
subpartition_position	numeric	二级分区在分区中的位置。
status	character varying(8)	指示索引分区是否可用。
tablespace_name	name	索引分区的表空间名称。
pct_free	numeric	块中最小可用空间百分比。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	指示是否记录对索引的更改。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
compression	character varying(13)	用于二级分区的压缩类型。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	根据索引的值指示表中行的顺序。需要通过执行analyze进行统计。
num_rows	numeric	二级分区中的行数。需要通过执行vacuum进行统计。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。数据库重启后，数据会丢失。
buffer_pool	character varying(7)	二级分区的缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	指示索引分区段是否已创建。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。

7.3.1.20 MY_PART_COL_STATISTICS

MY_PART_COL_STATISTICS视图显示当前用户拥有的表分区的列统计信息和直方图信息。所有用户可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-175 MY_PART_COL_STATISTICS 字段

名称	类型	描述
table_name	character varying(128)	表名。
partition_name	character varying(128)	表分区名称。
column_name	character varying(4000)	列名。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的低值。
high_value	raw	列中的高值。
density	numeric	• 如果列上有直方图，则该字段将显示直方图中跨越少于2个端点的值的选择性。它不代表跨越2个或更多端点的值的选择性。 • 如果列上没有可用的直方图，则该字段的值为1/NUM_DISTINCT。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
sample_size	numeric	用于分析此列的样本量。
last_analyzed	date	最近分析此列的日期。数据库重启后，数据会丢失。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
notes	character varying(63)	暂不支持，值为NULL。
avg_col_len	numeric	列的平均长度（以字节为单位）。
histogram	character varying(15)	直方图是否存在以及存在的类型： • NONE：不存在直方图。 • FREQUENCY：频率直方图。 • EQUI-WIDTH：等宽直方图。
schema	character varying(64)	列所属的名称空间的名称。

版权所有 © 华为云计算技术有限公司

说明

在ORA兼容模式下，初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea，若指定GUC参数mapping_date_to_datea的值为off，则该视图的last_analyzed字段类型会变更为timestamp(0) without time zone。

7.3.1.21 MY_PART_INDEXES

MY_PART_INDEXES视图显示当前用户下分区表索引的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-176 MY_PART_INDEXES 字段

名称	类型	描述
def_tablespace_name	name	分区表索引的表空间名称。
index_owner	character varying(64)	分区表索引的所有者名称。
index_name	character varying(64)	分区表索引的名称。
partition_count	bigint	分区表索引的索引分区的个数。
partitioning_key_count	integer	分区表的分区键个数。
partitioning_type	text	分区表的分区策略。 **说明** 当前分区表策略支持范围请参见**CREATE TABLE PARTITION**。
schema	character varying(64)	分区表索引的模式。
table_name	character varying(64)	分区表索引所属的分区表名称。
subpartitioning_type	text	二级分区表的分区策略。如果分区表是一级分区表，则显示NONE。 **说明** 当前二级分区表策略支持范围见**CREATE TABLE SUBPARTITION**。
def_subpartition_count	integer	默认创建二级分区的个数，二级分区表为1，一级分区表为0。
subpartitioning_key_count	integer	分区表二级分区键的个数。

7.3.1.22 MY_PART_KEY_COLUMNS

MY_PART_KEY_COLUMNS视图显示了当前用户拥有的分区表或分区索引的分区键列的相关信息。该视图所有用户可访问，仅可查看当前用户所属信息。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-177 MY_PART_KEY_COLUMNS 字段

名称	类型	描述
name	character varying(128)	分区表名或索引名。
object_type	character(5)	对象类型。 • 若分区为分区表，此列为table。 • 若分区为分区索引，此列为index。
column_name	character varying(4000)	分区表或索引的键列名。
column_position	numeric	列在分区中的位置。
collated_column_id	numeric	暂不支持，值为NULL。

7.3.1.23 MY_PART_TABLES

MY_PART_TABLES视图显示当前用户下分区表的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-178 MY_PART_TABLES 字段

名称	类型	描述
table_owner	character varying(64)	分区表的所有者名称。
table_name	character varying(64)	分区表的名称。
partitioning_type	text	分区表的分区策略。 **说明** 当前分区表策略支持范围请参见 **CREATE TABLE PARTITION**。
partition_count	bigint	分区表的分区个数。
partitioning_key_count	integer	分区表的分区键个数。
def_tablespace_name	name	分区表的表空间名称。
schema	character varying(64)	分区表的模式。
subpartitioning_type	text	二级分区表的分区策略。如果分区表是一级分区表，则显示NONE。
def_subpartition_count	integer	默认创建二级分区的个数，二级分区表为1，一级分区表为0。
subpartitioning_key_count	integer	分区表二级分区键的个数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
status	character varying(8)	暂不支持，值为valid。
def_pct_free	numeric	添加分区时使用的PCTFREE默认值。
def_pct_used	numeric	暂不支持，值为NULL。
def_ini_trans	numeric	添加分区时使用的INITRANS默认值。
def_max_trans	numeric	添加分区时使用的MAXTRANS默认值。
def_initial_extent	character varying(40)	暂不支持，值为NULL。
def_next_extent	character varying(40)	暂不支持，值为NULL。
def_min_extents	character varying(40)	暂不支持，值为NULL。
def_max_extents	character varying(40)	暂不支持，值为NULL。
def_max_size	character varying(40)	暂不支持，值为NULL。
def_pct_increase	character varying(40)	暂不支持，值为NULL。
def_freelists	numeric	暂不支持，值为NULL。
def_freelist_groups	numeric	暂不支持，值为NULL。
def_logging	character varying(7)	暂不支持，值为NULL。
def_compression	character varying(8)	添加分区时使用的默认压缩： • NONE • ENABLED • DISABLED
def_compress_for	character varying(30)	添加分区时使用的默认压缩。 **说明** 可用的压缩方法和压缩级别请参见**•WITH ({ storage_parameter = value } [, ...])**。
def_buffer_pool	character varying(7)	暂不支持，值为DEFAULT。
def_flash_cache	character varying(7)	暂不支持，值为NULL。
def_cell_flash_cache	character varying(7)	暂不支持，值为NULL。
ref_ptn_constraint_name	character varying(128)	暂不支持，值为NULL。
interval	character varying(1000)	区间值字符串。
autolist	character varying(3)	暂不支持，值为NO。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
interval_subpartition	character varying(1000)	暂不支持，值为NULL。
autolist_subpartition	character varying(3)	暂不支持，值为NO。
is_nested	character varying(3)	暂不支持，值为NO。
def_segment_creation	character varying(4)	暂不支持段页式设置，当启用segment时，值为YES。
def_indexing	character varying(3)	暂不支持，值为ON。
def_inmemory	character varying(8)	暂不支持，值为NONE。
def_inmemory_priority	character varying(8)	暂不支持，值为NULL。
def_inmemory_distribute	character varying(15)	暂不支持，值为NULL。
def_inmemory_compression	character varying(17)	暂不支持，值为NULL。
def_inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
def_read_only	character varying(3)	暂不支持，值为NO。
def_cellmemory	character varying(24)	暂不支持，值为NULL。
def_inmemory_service	character varying(12)	暂不支持，值为NULL。
def_inmemory_service_name	character varying(1000)	暂不支持，值为NULL。

7.3.1.24 MY_SUBPART_KEY_COLUMNS

MY_SUBPART_KEY_COLUMNS视图显示了当前用户所拥有的二级分区表或分区索引的分区键列的相关信息。该视图所有用户可访问，仅可查看当前用户所属信息。该视图同时存在于PG_CATALOG和SYS Schema下。分布式暂不支持二级分区表，该视图所有字段值为null。

表 7-179 MY_SUBPART_KEY_COLUMNS 字段

名称	类型	描述
name	character varying(128)	二级分区表名或索引名。
object_type	character(5)	对象类型。 • 若分区为分区表，此列为table。 • 若分区为分区索引，此列为index。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
column_name	character varying(4000)	二级分区表的二级分区键列名。
column_position	numeric	列在分区中的位置。
collated_column_id	numeric	暂不支持，值为NULL。

7.3.1.25 MY_TAB_PARTITIONS

MY_TAB_PARTITIONS视图显示当前用户下所有一级分区信息（包括二级分区表）。当前用户下每个分区表的一级分区都会在MY_TAB_PARTITIONS中有一条记录。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-180 MY_TAB_PARTITIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表的名称。
partition_name	character varying(64)	分区名称。
high_value	text	分区的边界值。 • 对于范围分区，显示各分区的上边界值。 • 对于列表分区，显示各分区的取值列表。 • 对于哈希分区，显示各分区的编号。
tablespace_name	name	分区表的表空间名称。
schema	character varying(64)	名称空间的名称。
composite	character varying(3)	表是否为二级分区表。
subpartition_count	bigint	二级分区的个数。
high_value_length	integer	分区边界值表达式的长度。
partition_position	numeric	分区在表中的位置。
pct_free	numeric	块中可用空间的最小百分比。
pct_used	numeric	暂不支持，值为NULL。
ini_trans	numeric	初始事务数，默认值为4，非USTORE分区表时为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_trans	numeric	最大事务数，默认值为128，非USTORE分区表时为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对表的更改。
compression	character varying(8)	表分区的实际压缩属性。
compress_for	character varying(30)	暂不支持，值为NULL。
num_rows	numeric	分区中的行数。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此分区的日期。
buffer_pool	character varying(7)	用于分区块的缓冲池。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
is_nested	character varying(3)	指示这是否是嵌套表分区。
parent_table_partition	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(4)	表分区是否创建了段。
indexing	character varying(4)	暂不支持，值为NULL。
read_only	character varying(4)	暂不支持，值为NULL。
inmemory	character varying(8)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。
inmemory_service_name	character varying(100)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。

7.3.1.26 MY_TAB_SUBPARTITIONS

MY_TAB_SUBPARTITIONS视图显示当前用户可访问的所有子分区表信息。该视图所有用户可访问，仅可查看当前用户所属信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-181 MY_TAB_SUBPARTITIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
schema	character varying(64)	模式名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
table_name	character varying(64)	表名。
partition_name	character varying(64)	分区名称。
subpartition_name	character varying(64)	子分区名称。
high_value	text	子分区边界值表达式。
high_value_length	integer	子分区边界值表达式的长度。
partition_position	numeric	分区在表内的位置。
subpartition_position	numeric	子分区在分区中的位置。
tablespace_name	name	子分区所在的表空间名称。
pct_free	numeric	块中最小可用空间百分比。
pct_used	numeric	暂不支持，值为NULL。
ini_trans	numeric	初始事务数。
max_trans	numeric	最大事务数。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extent	numeric	暂不支持，值为NULL。
max_extent	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
logging	character varying(7)	是否记录对表的更改： ● YES ● NO
compression	character varying(8)	子分区是否压缩。取值范围：ENABLED、DISABLED。
compress_for	character varying(30)	暂不支持，值为NULL。
num_rows	numeric	暂不支持，值为NULL。
blocks	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	timestamp with time zone	最近分析此表的日期。数据库重启后，数据会丢失。
buffer_pool	character varying(7)	子分区的缓冲池： ● DEFAULT ● KEEP ● RECYCLE ● NULL
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
interval	character varying(3)	分区是否在间隔分区表的间隔节中。分布式暂不支持间隔分区，该字段值为NO。
segment_created	character varying(3)	表子分区段是否已创建： ● YES：已创建。 ● NO：未创建。 ● N/A：此表没有子分区。
indexing	character varying(3)	暂不支持，值为NULL。
read_only	character varying(5)	暂不支持，值为NULL。
inmemory	character varying(8)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。
inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。

7.3.1.27 GS_STATIO_ALL_PARTITIONS

GS_STATIO_ALL_PARTITIONS视图包含当前数据库中每个分区表分区的I/O统计信息，此视图信息由gs_statio_get_all_partitions_stats()函数查询得到。

表 7-182 GS_STATIO_ALL_PARTITIONS 字段

名称	类型	描述
partition_oid	oid	分区OID。
schemaname	name	该分区模式名。
relname	name	该分区所在表的表名。
partition_name	name	该分区所在一级分区名。
sub_partition_name	name	该分区所在二级分区名。分布式不支持二级分区，值为NULL。
heap_blks_read	bigint	从该分区中读取的磁盘块数。
heap_blks_hit	bigint	该分区缓存命中数。
idx_blks_read	bigint	从分区中所有索引读取的磁盘块数。
idx_blks_hit	bigint	分区中所有索引命中缓存数。
toast_blks_read	bigint	该分区的TOAST表分区读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该分区的TOAST表分区命中缓冲区数（如果存在）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tidx_blks_read	bigint	该分区的TOAST表分区索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该分区的TOAST表分区索引命中缓冲区数（如果存在）。

7.3.1.28 GS_STAT_XACT_ALL_PARTITIONS

GS_STAT_XACT_ALL_PARTITIONS视图显示命名空间中所有分区表分区的事务状态信息，此视图信息由gs_stat_get_xact_all_partitions_stats()函数查询得到。

表 7-183 GS_STAT_XACT_ALL_PARTITIONS 字段

名称	类型	描述
partition_oid	oid	分区的OID。
schemaname	name	该分区的模式名。
relname	name	该分区所在表的表名。
partition_name	name	该分区所在一级分区名。
sub_partition_name	name	该分区所在二级分区名。分布式不支持二级分区，值为NULL。
seq_scan	bigint	该分区发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该分区发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	热更新行数（比如没有更新所需的单独索引）。

7.3.1.29 GS_STAT_ALL_PARTITIONS

GS_STAT_ALL_PARTITIONS视图包含当前数据库中所有分区表中每个分区的信息，每个分区各占一行，显示该分区访问情况的统计信息。当前版本不支持二级分区统计信息更新。此视图信息由gs_stat_get_all_partitions_stats()函数查询得到。

版权所有 © 华为云计算技术有限公司

表 7-184 GS_STAT_ALL_PARTITIONS 字段

名称	类型	描述
partition_oid	oid	分区的OID。
schemaname	name	该分区所在表的模式名。
relname	name	该分区所在表的表名。
partition_name	name	该分区所在一级分区名。
sub_partition_name	name	该分区所在二级分区名。
seq_scan	bigint	该分区发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该分区发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	热更新行数（比如没有更新所需的单独索引）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计非活跃行数。在Ustore表中仅代表死行指针数量。
last_vacuum	timestamp with time zone	该分区最后一次被清理的时间。
last_autovacuum	timestamp with time zone	该分区最后一次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	该分区最后一次被分析的时间。
last_autoanalyze	timestamp with time zone	该分区最后一次被autovacuum守护线程分析的时间。
vacuum_count	bigint	该分区被清理的次数。
autovacuum_count	bigint	该分区被autovacuum守护线程清理的次数。
analyze_count	bigint	该分区被分析的次数。
autoanalyze_count	bigint	该分区被autovacuum守护线程分析的次数。

版权所有 © 华为云计算技术有限公司

7.3.2 负载管理

7.3.2.1 GS_WLM_CGROUP_INFO

GS_WLM_CGROUP_INFO视图显示当前执行作业的控制组的信息。

说明

当查询视图无数据显示时，请联系华为工程师提供技术支撑。

表 7-185 GS_WLM_CGROUP_INFO 字段

名称	类型	描述
cgroup_name	text	控制组的名称。
priority	integer	作业的优先级。
usage_percent	integer	控制组占用的百分比。
shares	bigint	控制组分配的CPU资源配额。
cpuacct	bigint	CPU配额分配。
cpuset	text	CPU限额分配。
relpath	text	控制组的相对路径。
valid	text	该控制组是否有效。
node_group	text	保留字段。

7.3.2.2 GS_WLM_REBUILD_USER_RESOURCE_POOL

该视图用于在当前连接节点上重建内存中用户的资源池信息。只在资源池信息缺失或错乱时作为补救措施。查询该视图需要sysadmin权限。

表 7-186 GS_WLM_REBUILD_USER_RESOURCE_POOL 字段

名称	类型	描述
gs_wlm_rebuild_user_resource_pool	boolean	重建内存中用户资源池信息结果。t为成功，f为失败。

7.3.2.3 GS_WLM_USER_INFO

GS_WLM_USER_INFO用户统计信息视图。查询该视图需要sysadmin权限。

说明

当查询视图无数据显示时，请联系华为工程师提供技术支撑。

版权所有 © 华为云计算技术有限公司

表 7-187 GS_WLM_USER_INFO 字段

名称	类型	描述
userid	oid	用户OID。
username	name	用户名。
sysadmin	boolean	是否是管理员用户。 • t（true）：是。 • f（false）：不是。
rpoid	oid	关联的资源池的OID。
respool	name	关联的资源池的名称。
parentid	oid	用户组的OID。
totalspace	bigint	用户的可用空间上限。
spacelimit	bigint	用户表空间限制。
childcount	integer	子用户的个数。
childlist	text	子用户列表。

7.3.2.4 GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO

该视图用于查看分布式下所有节点实时的资源池并发数使用情况。查询该视图需要sysadmin权限或者monadmin权限。

该视图只能查询主DN节点上的数据，备节点需要登录到备节点上后使用系统函数**gs_wlm_respool_concurrency_info()**查询并发数使用情况。

表 7-188 GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO 字段

名称	类型	描述
node_name	name	节点名称。
respool_name	text	资源池名称。
max_concurrency	integer	资源池支持的最大并发数。
running_concurrency	integer	当前资源池正在执行的并发数。
waiting_concurrency	integer	当前资源池正在等待的并发数。

示例：

```
gaussdb=# SELECT * FROM GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO;
 node_name   | respool_name | max_concurrency | running_concurrency | waiting_concurrency
-------------+--------------+-----------------+---------------------+---------------------
 coordinator1 | default_pool |              -1 |                   1 |                   0
```


版权所有 © 华为云计算技术有限公司

```
 datanode3    | default_pool |           -1 |                1 |                0
 datanode1    | default_pool |           -1 |                1 |                0
 datanode2    | default_pool |           -1 |                1 |                0
(4 rows)
```

7.3.2.5 GLOBAL_WLM_RESPOOL_CONNECTION_INFO

该视图用于查看分布式下所有节点实时的资源池连接数使用情况。查询该视图需要sysadmin权限或者monadmin权限。

该视图只能查询主DN节点上的数据，备节点需要登录到备节点上后使用系统函数**gs_wlm_respool_connection_info()**查询并发数使用情况。

表 7-189 GLOBAL_WLM_RESPOOL_CONNECTION_INFO 字段

名称	类型	描述
node_name	name	节点名称。
respool_name	text	资源池名称。
max_connections	integer	资源池最大连接数。
curr_connections	integer	当前资源池已有连接数。

示例：

```
gaussdb=# SELECT * FROM GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO;
 node_name   | respool_name | max_connections | curr_connections
--------------+--------------+-----------------+------------------
 coordinator1 | default_pool |              -1 |                1
 datanode2    | default_pool |              -1 |                2
 datanode3    | default_pool |              -1 |                2
 datanode1    | default_pool |              -1 |                2
(4 rows)
```

7.3.2.6 GLOBAL_WLM_RESPOOL_CPU_INFO

该视图用于查看分布式下所有节点实时的资源池CPU使用情况。查询该视图需要sysadmin权限或者monadmin权限。

该视图只能查询主DN节点上的数据，备节点需要登录到备节点上后使用系统函数**gs_wlm_respool_cpu_info()**查询CPU使用情况。

表 7-190 GLOBAL_WLM_RESPOOL_CPU_INFO 字段

名称	类型	描述
node_name	name	节点名称。
respool_name	text	资源池名称。
control_group	text	cgroup名称。
cpu_affinity	text	CPU绑定core的数值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cpu_usage	integer	资源池的CPU使用率。 **说明** 对于CN和DN混合部署的场景，CN和DN共享同一份CPU资源，因此CN和DN的cpu_usage显示相同，对于CN和DN独立部署场景，会独立显示。

示例：

```
gaussdb=# SELECT * FROM GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO;
 node_name   | respool_name |    control_group    | cpu_affinity | cpu_usage
-------------+--------------+---------------------+--------------+-----------
 coordinator1 | default_pool | DefaultClass:Medium | 0-71         |         0
 datanode1    | default_pool | DefaultClass:Medium | 0-71         |         0
 datanode2    | default_pool | DefaultClass:Medium | 0-71         |         0
 datanode3    | default_pool | DefaultClass:Medium | 0-71         |         0
(4 rows)
```

7.3.2.7 GLOBAL_WLM_RESPOOL_IO_INFO

该视图用于查看分布式下所有节点实时的资源池I/O使用情况。查询该视图需要sysadmin权限或者monadmin权限。

该视图只能查询主DN节点上的数据，备节点需要登录到备节点上后使用系统函数**gs_wlm_respool_io_info()**查询I/O使用情况。

表 7-191 GLOBAL_WLM_RESPOOL_IO_INFO 字段

名称	类型	描述
node_name	name	节点名称。
respool_name	text	资源池名称。
io_limits	integer	每秒触发I/O的次数上限。0表示不控制。 • 行存单位：GUC参数io_control_unit，I/O管控时用来对I/O次数进行计数的单位。1个io_control_unit次为1次IOPS。
io_priority	text	I/O利用率高达90%时，消耗I/O作业进行I/O资源管控时关联的优先级等级。None表示不控制。
current_iops	integer	当前I/O已经触发的次数。 当前I/O统计值存在偶尔超过上限的情况，与I/O统计算法有关，属于正常波动。

版权所有 © 华为云计算技术有限公司

示例：

```
gaussdb=# SELECT * FROM GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO;
 node_name   | respool_name | io_limits | io_priority | current_iops
-------------+--------------+-----------+-------------+--------------
 coordinator1 | default_pool |         0 | None        |            0
 datanode2    | default_pool |         0 | None        |            0
 datanode3    | default_pool |         0 | None        |            0
 datanode1    | default_pool |         0 | None        |            0
(4 rows)
```

7.3.2.8 GLOBAL_WLM_RESPOOL_MEMORY_INFO

该视图用于查看分布式下所有节点实时的资源池内存使用情况。查询该视图需要sysadmin权限或者monadmin权限。

该视图只能查询主DN节点上的数据，备节点需要登录到备节点上后使用系统函数**gs_wlm_respool_memory_info()**查询内存使用情况。

表 7-192 GLOBAL_WLM_RESPOOL_MEMORY_INFO 字段

名称	类型	描述
node_name	name	节点名称。
respool_name	text	资源池名称。
max_dynamic_memory	text	最大可使用的动态内存。
current_dynamic_memory	text	当前已经使用的动态内存。
max_shared_memory	text	最大可使用的共享内存。
current_shared_memory	text	当前已经使用的共享内存。
shared_memory_hits_percent	integer	该功能不可用，留待后续演进。字段保留，恒为0。

示例：

```
gaussdb=# SELECT * FROM GLOBAL_WLM_RESPOOL_CONCURRENCY_INFO;
 node_name   | respool_name | max_dynamic_memory | current_dynamic_memory | max_shared_memory |
current_shared_memory | shared_memory_hits_percent
-------------+--------------+--------------------+------------------------+-------------------+-----------------------
+---------------------------
 coordinator1 | default_pool | -1                 | 9415kB                 | -1                | 3856kB
|                       0
 datanode1    | default_pool | -1                 | 10724kB                | -1                | 2808kB
|                       0
 datanode2    | default_pool | -1                 | 10724kB                | -1                | 2808kB
|                       0
 datanode3    | default_pool | -1                 | 10724kB                | -1                | 560kB
|                       0
(4 rows)
```


版权所有 © 华为云计算技术有限公司

7.3.2.9 PGXC_WLM_REBUILD_USER_RESPOOL

该视图用于在所有的节点上重建内存中用户的资源池信息。只在资源池信息缺失或错乱时作为补救措施。查询该视图需要sysadmin权限。

说明

当查询视图无数据显示时，请联系华为工程师提供技术支撑。

表 7-193 PGXC_WLM_REBUILD_USER_RESPOOL 字段

名称	类型	描述
gs_wlm_rebuild_user_resource_pool	boolean	重建内存中用户资源池信息结果。t为成功，f为失败。

7.3.3 OLTP 表压缩

7.3.3.1 GS_ADM_ILMDATAMOVEMENTPOLICIES

GS_ADM_ILMDATAMOVEMENTPOLICIES视图反映ILM策略的数据移动概要信息，包含策略名称、动作类型、条件等。需要有系统管理员权限才可以访问此系统视图。

表 7-194 GS_ADM_ILMDATAMOVEMENTPOLICIES 字段

名称	类型	描述
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
action_type	character varying(11)	动作类型，当前版本仅支持压缩。
scope	character varying(7)	作用域，当前版本仅支持行。
compression_level	character varying(30)	压缩级别，动作类型为压缩时存在。
tier_tablespace	character varying(128)	目标空间，动作类型为迁移时存在。当前版本值为null。
tier_status	character varying(9)	迁移目标空间是否是只读。当前版本值为null。
condition_type	character varying(22)	条件类型，当前版本仅支持最后修改时间。
condition_days	numeric	条件天数。
custom_function	character varying(128)	自定义函数名称。当前版本值为null。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
policy_subtype	character varying(10)	策略子类型。当前版本值为null。
action_clause	clob	策略执行时可以自动执行的文本。当前版本值为null。
tier_to	character varying(10)	迁移动作的目的地类型。当前版本值为null。
node_name	name	数据来源节点的名称。

7.3.3.2 GS_ADM_ILMOBJECTS

GS_ADM_ILMOBJECTS视图反映所有存在ILM策略应用的数据对象与相应策略的概要信息，包含策略名称、数据对象名称、策略的来源、策略的启用删除状态。需要有系统管理员权限才可以访问此系统视图。

表 7-195 GS_ADM_ILMOBJECTS 字段

名称	类型	描述
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
object_owner	character varying(128)	数据对象的属主。
object_name	character varying(128)	数据对象名称。
subobject_name	character varying(128)	数据对象分区名称。
object_type	character varying(24)	数据对象类型。取值范围： ● TABLE：表。 ● TABLE PARTITION：分区。 ● TABLE SUBPARTITION：二级分区。 ● INDEX：普通索引。 ● GLOBAL PARTITIONED INDEX：全局索引。 ● GLOBAL SECONDARY INDEX：全局二级索引。 ● INDEX PARTITION：索引分区。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
inherited_from	character varying(20)	当前策略是继承哪一个对象上的策略。取值范围： ● TABLE：表。 ● TABLE PARTITION：分区。 ● TABLE SUBPARTITION：二级分区。 ● INDEX：普通索引。 ● POLICY NOT INHERITED：不继承。
tbs_inherited_from	character varying(30)	当前策略是继承哪一个表空间上的策略，当前版本值为NULL。
enabled	character varying(7)	该策略对当前对象是否开启。
deleted	character varying(7)	标识该对象上的ILM策略被删除。实际操作结果是删除分区或者是删除分区上的策略，该条记录都会被删除（ILMOBJ）。
node_name	name	数据来源节点的名称。

7.3.3.3 GS_ADM_ILMPOLICIES

GS_ADM_ILMPOLICIES视图反映ILM策略的概要信息，包含策略名称、类型、启用禁用状态、删除状态。需要有系统管理员权限才可以访问此系统视图。

表 7-196 GS_ADM_ILMPOLICIES 字段

名称	类型	描述
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
policy_type	character varying(13)	策略类型。
tablespace	character varying(30)	表空间名称。当前版本值为null。
enabled	character varying(6)	策略是否开启。
deleted	character varying(7)	策略是否被删除。
node_name	name	数据来源节点的名称。

版权所有 © 华为云计算技术有限公司

7.3.3.4 GS_ADM_ILMEVALUATIONDETAILS

GS_ADM_ILMEVALUATIONDETAILS视图描述ADO Task的评估详情信息，包含Task ID，策略信息、对象信息、评估结果以及ADO JOB名称。默认只有系统管理员权限才可以访问此系统视图。

表 7-197 GS_ADM_ILMEVALUATIONDETAILS 字段

名称	类型	描述
task_id	bigint	ADO Task的ID。
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
object_owner	character varying(128)	数据对象所在Schema名称。
object_name	character varying(128)	数据对象名称。
subobject_name	character varying(128)	数据对象分区名称。
object_type	character varying(24)	数据对象类型： ● TABLE：表。 ● TABLE PARTITION：分区。 ● TABLE SUBPARTITION：二级分区。 ● INDEX：普通索引。 ● GLOBAL PARTITIONED INDEX：全局索引。 ● GLOBAL SECONDARY INDEX：全局二级索引。 ● INDEX PARTITION：索引分区。
selected_for_execution	character varying(42)	评估结果： ● SELECTED FOR EXECUTION：评估通过。 ● PRECONDITION NOT SATISFIED：评估不通过。 ● JOB ALREADY EXISTS：任务已存在。
job_name	character varying(128)	执行具体ADO Job的任务名称，当评估结果为评估通过时，此处有值。
comments	character varying(4000)	预留字段。
node_name	name	数据来源节点的名称。

版权所有 © 华为云计算技术有限公司

7.3.3.5 GS_ADM_ILMPARAMETERS

GS_ADM_ILMPARAMETERS视图描述ILM调度与执行的相关环境参数，该参数可通过DBE_ILM_ADMIN.CUSTOMIZE_ILM接口进行修改。默认只有系统管理员权限才可以访问此系统视图。

表 7-198 GS_ADM_ILMPARAMETERS 字段

名称	类型	描述
name	character varying(128)	ADO相关的环境参数，通过DBE_ILM_ADMIN.CUSTOMIZE_ILM接口进行修改。
value	numeric	参数取值。

7.3.3.6 GS_ADM_ILMRESULTS

GS_ADM_ILMRESULTS视图反映ADO JOB的执行详情信息，包含Task ID，JOB名称、JOB状态、JOB时间信息等。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。

表 7-199 GS_ADM_ILMRESULTS 字段

名称	类型	描述
task_id	bigint	ADO Task的ID。
job_name	character varying(128)	ADO Job的任务名称。
job_state	character varying(35)	ADO Job的状态。
start_time	timestamp with time zone	JOB开始被调度的时间。
completion_time	timestamp with time zone	完成时间。
comments	character varying(4000)	JOB失败后此处记录失败原因。
statistics	clob	统计信息。
node_name	name	数据来源节点的名称。

版权所有 © 华为云计算技术有限公司

7.3.3.7 GS_ADM_ILMTASKS

GS_ADM_ILMTASKS视图反映ADO Task的概要信息，包含Task ID，Task Owner，状态以及时间信息。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。

表 7-200 GS_ADM_ILMTASKS 字段

名称	类型	描述
task_id	bigint	ADO Task的ID。
task_owner	character varying(128)	ADO Task发起的用户，仅ADM视图存在此列。
state	character varying(9)	状态： ● INACTIVE：非活动。 ● ACTIVE：活动。 ● COMPLETED：完成。 ● UNKNOWN：评估中。
creation_time	timestamp with time zone	创建时间。
start_time	timestamp with time zone	变成活动状态的时间。
completion_time	timestamp with time zone	完成时间。
node_name	name	数据来源节点的名称。

7.3.3.8 GS_MY_ILMEVALUATIONDETAILS

GS_MY_ILMEVALUATIONDETAILS视图描述ADO Task的评估详情信息，包含Task ID，策略信息、对象信息、评估结果以及ADO JOB名称。

表 7-201 GS_MY_ILMEVALUATIONDETAILS 字段

名称	类型	描述
task_id	bigint	ADO Task的ID。
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
object_owner	character varying(128)	数据对象Schema名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
object_name	character varying(128)	数据对象名称。
subobject_name	character varying(128)	数据对象分区名称。
object_type	character varying(24)	数据对象类型： ● TABLE：表。 ● TABLE PARTITION：分区。 ● TABLE SUBPARTITION：二级分区。 ● INDEX：普通索引。 ● GLOBAL PARTITIONED INDEX：全局索引。 ● GLOBAL SECONDARY INDEX：全局二级索引。 ● INDEX PARTITION：索引分区。
selected_for_execution	character varying(42)	评估结果： ● SELECTED FOR EXECUTION：评估通过。 ● PRECONDITION NOT SATISFIED：评估不通过。 ● JOB ALREADY EXISTS：任务已存在。
job_name	character varying(128)	执行具体ADO Job的任务名称，当评估结果为评估通过时，此处有值。
comments	character varying(4000)	预留字段。
node_name	name	数据来源节点的名称。

7.3.3.9 GS_MY_ILMRESULTS

GS_MY_ILMRESULTS视图反映ADO JOB的执行详情信息，包含Task ID，JOB名称、JOB状态、JOB时间信息等。

表 7-202 GS_MY_ILMRESULTS 字段

名称	类型	描述
task_id	bigint	ADO Task的ID。
job_name	character varying(128)	ADO Job的任务名称。
job_state	character varying(35)	ADO Job的状态。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
start_time	timestamp with time zone	JOB开始被调度的时间。
completion_time	timestamp with time zone	完成时间。
comments	character varying(4000)	JOB失败后此处记录失败原因。
statistics	clob	统计信息。
node_name	name	数据来源节点的名称。

7.3.3.10 GS_MY_ILMTASKS

GS_MY_ILMTASKS视图反映ADO Task的概要信息，包含Task ID，Task Owner，状态以及时间信息。

表 7-203 GS_MY_ILMTASKS 字段

名称	类型	描述
task_id	bigint	ADO Task的ID。
task_owner	character varying(128)	ADO Task发起的用户。
state	character varying(9)	状态： ● INACTIVE：非活动。 ● ACTIVE：活动。 ● COMPLETED：完成。 ● UNKNOWN：评估中。
creation_time	timestamp with time zone	创建时间。
start_time	timestamp with time zone	变成活动状态的时间。
completion_time	timestamp with time zone	完成时间。
node_name	name	数据来源节点的名称。

版权所有 © 华为云计算技术有限公司

7.3.3.11 GS_MY_ILMDATAMOVEMENTPOLICIES

GS_MY_ILMDATAMOVEMENTPOLICIES视图反映ILM策略的数据移动概要信息，包含策略名称、动作类型、条件等。

表 7-204 GS_MY_ILMDATAMOVEMENTPOLICIES 字段

名称	类型	描述
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
action_type	character varying(11)	动作类型，当前版本仅支持压缩。
scope	character varying(7)	作用域，当前版本仅支持行。
compression_level	character varying(30)	压缩级别，动作类型为压缩时存在。
tier_tablespace	character varying(128)	目标空间，动作类型为迁移时存在。当前版本值为null。
tier_status	character varying(9)	迁移目标空间是否是只读。当前版本值为null。
condition_type	character varying(22)	条件类型，当前版本仅支持最后修改时间。
condition_days	numeric	条件天数。
custom_function	character varying(128)	自定义函数名称。当前版本值为null。
policy_subtype	character varying(10)	策略子类型。当前版本值为null。
action_clause	clob	策略执行时可以自动执行的文本。当前版本值为null。
tier_to	character varying(10)	迁移动作的目的地类型。当前版本值为null。
node_name	name	数据来源节点的名称。

7.3.3.12 GS_MY_ILMOBJECTS

GS_MY_ILMOBJECTS视图反映所有存在ILM策略应用的数据对象与相应策略的概要信息，包含策略名称、数据对象名称、策略的来源、策略的启用删除状态。

版权所有 © 华为云计算技术有限公司

表 7-205 GS_MY_ILMOBJECTS

名称	类型	描述
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
object_owner	character varying(128)	数据对象Schema名称。
object_name	character varying(128)	数据对象名称。
subobject_name	character varying(128)	数据对象分区名称。
object_type	character varying(24)	数据对象类型： ● TABLE：表。 ● TABLE PARTITION：分区。 ● TABLE SUBPARTITION：二级分区。 ● INDEX：普通索引。 ● GLOBAL PARTITIONED INDEX：全局索引。 ● GLOBAL SECONDARY INDEX：全局二级索引。 ● INDEX PARTITION：索引分区。
inherited_from	character varying(20)	当前策略是继承哪一个对象上的策略，取值范围： ● TABLE：表。 ● TABLE PARTITION：分区。 ● TABLE SUBPARTITION：二级分区。 ● INDEX：普通索引。 ● POLICY NOT INHERITED：不继承。
tbs_inherited_from	character varying(30)	当前策略是继承哪一个表空间上的策略。当前版本值为null。
enabled	character varying(7)	该策略对当前对象是否开启。
deleted	character varying(7)	标识该对象上的ILM策略被删除，实际操作结果是删除分区或者是删除分区上的策略，该条记录都会被删除（ILMOBJ）。
node_name	name	数据来源节点的名称。

7.3.3.13 GS_MY_ILMPOLICIES

GS_MY_ILMPOLICIES视图反映ILM策略的概要信息，包含策略名称、类型、启用禁用状态、删除状态。

版权所有 © 华为云计算技术有限公司

表 7-206 GS_MY_ILMPOLICIES

名称	类型	描述
policy_name	character varying(128)	ADO策略名称，系统自动生成，规则为：p+策略ID。
policy_type	character varying(13)	策略类型。
tablespace	character varying(30)	表空间名称，当该策略制定在表空间上时此处有值。当前版本值为null。
enabled	character varying(6)	策略是否开启。
deleted	character varying(7)	策略是否被删除。
node_name	name	数据来源节点的名称。

7.3.4 通信

7.3.4.1 COMM_CLIENT_INFO

COMM_CLIENT_INFO视图显示单个节点活跃的客户端连接信息（DN上查询该视图显示CN连接DN的信息），默认只有系统管理员权限才可以访问此系统视图。

表 7-207 COMM_CLIENT_INFO 字段

名称	类型	描述
node_name	text	当前DN节点的名称，如dn_6001_6002_6003。
app	text	DN上查询该视图，app显示为连接当前DN的客户端，如coordinator(CN)、GTM、DN等。
tid	bigint	当前线程的线程号。
lwtid	integer	当前线程的轻量级线程号。
query_id	bigint	查询ID，对应debug_query_id。
socket	integer	如果是物理连接，显示socket fd。
remote_ip	text	对端节点IP。
remote_port	text	对端节点port。
logic_id	integer	如果是逻辑连接，显示sid。

版权所有 © 华为云计算技术有限公司

7.3.4.2 GLOBAL_COMM_CLIENT_INFO

GLOBAL_COMM_CLIENT_INFO用来查询整个集群全局节点活跃的客户端连接信息，默认只有系统管理员权限才可以访问此系统视图。

表 7-208 GLOBAL_COMM_CLIENT_INFO 字段

名称	类型	描述
node_name	text	当前节点的名称。
app	text	连接当前DN的客户端，如coordinator(CN)、GTM、DN等。
tid	bigint	当前线程的线程号。
lwtid	integer	当前线程的轻量级线程号。
query_id	bigint	查询ID，对应debug_query_id。
socket	integer	如果是物理连接，显示socket。
remote_ip	text	对端节点IP。
remote_port	text	对端节点port。
logic_id	integer	如果是逻辑连接，显示sid。

7.3.4.3 GS_COMM_LISTEN_ADDRESS_EXT_INFO

GS_COMM_LISTEN_ADDRESS_EXT_INFO描述了查询连接当前DN扩展IP的相关线程、会话以及socket等DFX信息。使用场景参见系统函数描述 **gs_comm_listen_address_ext_info()**。

表 7-209 GS_COMM_LISTEN_ADDRESS_EXT_INFO 字段

名称	类型	描述
node_name	text	当前实例名。
app	text	当前连接DN的客户端。
tid	bigint	当前线程的线程号。
lwtid	integer	当前线程的轻量级线程号。
query_id	bigint	当前线程的查询ID。
socket	integer	当前物理连接的socket fd。
remote_ip	text	当前连接对端IP。
remote_port	text	当前连接对端port。
local_ip	text	当前连接本端IP。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
local_port	text	当前连接本端port。

7.3.4.4 GS_GET_LISTEN_ADDRESS_EXT_INFO

GS_GET_LISTEN_ADDRESS_EXT_INFO视图描述了查询当前DN实例扩展IP配置信息。具体使用场景参见**gs_get_listen_address_ext_info()**。

表 7-210 GS_GET_LISTEN_ADDRESS_EXT_INFO 字段

名称	类型	描述
node_name	text	DN实例名。
host	text	DN实例侦听IP。
port	bigint	DN实例侦听的port。
ext_listen_ip	text	DN实例配置扩展IP。

7.3.4.5 GS_LIBCOMM_FD_INFO

记录libcomm通信库socket长连接信息。查询该视图需要PUBLIC权限。

表 7-211 GS_LIBCOMM_FD_INFO 字段

名称	类型	描述
node_name	text	DN实例名。
ip	text	DN的IP。
ctrl_port	bigint	控制通道端口。
data_port	bigint	数据通道端口。
remote_name	text	对端DN示例名。
remote_ip	text	对端IP。
remote_port	bigint	对端端口号。
local_ip	text	本地IP。
local_port	bigint	本地端口号。
fd	bigint	socket连接。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
type	text	TCP连接通道类型，包括： ● DATA_SEND_FD：数据通道发送端的socket。 ● DATA_RECV_FD：数据通道接收端的socket。 ● CTL_SEND_FD：控制通道发送端的socket。 ● CTL_RECV_FD：控制通道接收端的socket。

7.3.4.6 GS_LIBCOMM_MEMORY_INFO

记录当前libcomm通信库内部内存使用的基础配置信息。查询该视图需要PUBLIC权限。

表 7-212 GS_LIBCOMM_MEMORY_INFO 字段

名称	类型	描述
node_name	text	DN实例名。
current_used_memory	bigint	libcomm通信库使用内存，单位为字节（byte）。
current_data_item_num	bigint	libcomm无锁队列数据块个数。
init_mailbox_memory	bigint	libcomm初始化mailbox使用的内存，单位为字节（byte）。
max_datanode	bigint	libcomm通信库支持的最大DN个数。
max_stream	bigint	libcomm单个TCP连接支持的最大数据流个数。
max_quota_size	bigint	libcomm逻辑连接的buffer总大小，单位为字节（byte）。
max_usable_memory	bigint	libcomm通信库缓存最大可使用内存，单位为字节（byte）。
max_memory_pool	bigint	libcomm通信库在每个DN上可以使用的内存资源池总大小，单位为字节（byte）。

7.3.4.7 GS_STATIC_THREADPOOL_CTRL_STATUS

GS_STATIC_THREADPOOL_CTRL_STATUS返回当前实例线程池相关静态线程的统计信息。线程池开启状态下，CN线程池支持该视图查询。连接DN查询返回空行；线程池关闭下返回空行；参数static_thread_pool_num设置为0或小于线程池group数时静态线程池默认值为0。查询该视图需要PUBLIC权限。

版权所有 © 华为云计算技术有限公司

表 7-213 GS_STATIC_THREADPOOL_CTRL_STATUS 字段

名称	类型	描述
node_name	text	实例名。
group_id	integer	线程池group组id。
worker_info	text	描述当前group组线程池运行中的动态的统计信息。包括如下信息： default、default_s：线程数量。动态线程池默认值，静态线程池默认值。 expect、expect_s：线程数量。动态线程池预期值，静态线程池预期值。 actual：实际运行的线程数量，包括动态池线程和静态池线程。 static threads limit：当前group组配置的静态池线程数。 has static threads：当前group组是否创建静态池。默认为0，表示没有创建。 idle static threads：空闲的静态池线程个数。 wait session num：等待的会话数。

7.3.4.8 PG_COMM_DELAY

PG_COMM_DELAY视图展示单个DN的通信库时延状态。

表 7-214 PG_COMM_DELAY 字段

名称	类型	描述
node_name	text	节点名称。
remote_name	text	连接对端节点名称。
remote_host	text	连接对端IP地址。
stream_num	integer	当前物理连接使用的stream逻辑连接数量。
min_delay	integer	当前物理连接一分钟内探测到的最小时延。单位：微秒。 **说明** 负数结果无效，请重新等待时延状态更新后再执行。
average	integer	当前物理连接一分钟内探测时延的平均值。单位：微秒。
max_delay	integer	当前物理连接一分钟内探测到的最大时延。单位：微秒。

版权所有 © 华为云计算技术有限公司

7.3.4.9 PG_COMM_RECV_STREAM

PG_COMM_RECV_STREAM视图展示单个DN上所有的通信库接收流状态。

表 7-215 PG_COMM_RECV_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的tcp通信socket。
state	text	通信流当前的状态。 ● UNKNOWN：当前逻辑连接状态未知。 ● READY：逻辑连接已就绪。 ● RUN：逻辑连接发送报文正常。 ● HOLD：逻辑连接发送报文等待中。 ● CLOSED：关闭逻辑连接。 ● TO_CLOSED：表示将会关闭逻辑连接。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
recv_bytes	bigint	通信流接收的数据总量。单位：Byte。
time	bigint	通信流当前生命周期使用时长。单位：ms。
speed	bigint	通信流的平均接收速率。单位：Byte/s。
quota	bigint	通信流当前的通信配额值。单位：Byte。
buff_usize	bigint	通信流当前缓存的数据大小。单位：Byte。

7.3.4.10 PG_COMM_SEND_STREAM

PG_COMM_SEND_STREAM视图展示单个DN上所有的通信库发送流状态。

版权所有 © 华为云计算技术有限公司

表 7-216 PG_COMM_SEND_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的tcp通信socket。
state	text	通信流当前的状态。 ● UNKNOWN：当前逻辑连接状态未知。 ● READY：逻辑连接已就绪。 ● RUN：逻辑连接发送报文正常。 ● HOLD：逻辑连接发送报文等待中。 ● CLOSED：关闭逻辑连接。 ● TO_CLOSED：将会关闭逻辑连接。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
send_bytes	bigint	通信流发送的数据总量。单位：Byte。
time	bigint	通信流当前生命周期使用时长。单位：ms。
speed	bigint	通信流的平均发送速率。单位：Byte/s。
quota	bigint	通信流当前的通信配额值。单位：Byte。
wait_quota	bigint	通信流等待quota值产生的额外时间开销。单位：ms。

7.3.4.11 PG_COMM_STATUS

PG_COMM_STATUS视图展示单个DN的通信库状态。

表 7-217 PG_COMM_STATUS 字段

名称	类型	描述
node_name	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
rxpck_rate	integer	节点通信库接收速率，单位Byte/s。
txpck_rate	integer	节点通信库发送速率，单位Byte/s。
rxkbyte_rate	bigint	节点通信库接收速率，单位KByte/s。
txkbyte_rate	bigint	节点通信库发送速率，单位KByte/s。
buffer	bigint	cmailbox的缓存大小。
memkbyte_libcomm	bigint	libcomm线程通信内存大小，单位Byte。
memkbyte_libpq	bigint	libpq线程通信内存大小，单位Byte。
used_pm	integer	postmaster线程实时使用率。
used_sflow	integer	gs_sender_flow_controller线程实时使用率。
used_rflow	integer	gs_receiver_flow_controller线程实时使用率。
used_rloop	integer	多个gs_receivers_loop线程中高的实时使用率。
stream	integer	当前使用的逻辑连接总数。

7.3.4.12 PG_GET_INVALID_BACKENDS

PG_GET_INVALID_BACKENDS视图显示CN上连接到当前DN备机的后台线程信息，只有系统管理员和监控管理员才可以访问。

表 7-218 PG_GET_INVALID_BACKENDS 字段

名称	类型	描述
pid	bigint	线程ID。
node_name	text	后台线程中连接的节点信息。
dbname	name	当前连接的数据库。
backend_start	timestamp with time zone	后台线程启动的时间。
query	text	后台线程正在执行的查询语句。

7.3.4.13 PG_POOLER_STATUS

PG_POOLER_STATUS视图查询pooler中的缓存连接状态。该视图只能在CN上查询，显示本地CN的pooler模块的连接缓存信息。

版权所有 © 华为云计算技术有限公司

表 7-219 PG_POOLER_STATUS 字段

名称	类型	描述
database	text	数据库名称。
user_name	text	用户名。
tid	bigint	• 非线程池逻辑下为连接CN的线程id。 • 线程池逻辑下为连接CN的sessionid。
node_oid	bigint	连接的实例节点OID。
node_name	name	连接的实例节点名称。
in_use	boolean	连接是否正被使用。 • t（true）：表示连接正在使用。 • f（false）：表示连接没有使用。
local_host	text	本端IP。
local_port	bigint	本端端口号。
remote_host	text	连接的实例节点IP。
node_port	bigint	连接的实例节点端口号。
fdsock	bigint	对端socket。
remote_pid	bigint	对端处于非线程池逻辑下为对端的线程id，对端处于线程池逻辑下为对端的sessionid。
session_params	text	由此连接下发的GUC session参数。
used_count	bigint	该连接的复用次数。
idx	bigint	连接的实例节点逻辑连接id。
streamid	bigint	每个逻辑连接对应的流标识id。

7.3.4.14 PGXC_COMM_DELAY

PGXC_COMM_DELAY视图展示所有DN的通信库时延状态，仅系统管理员和监控管理员可以查看。

表 7-220 PGXC_COMM_DELAY 字段

名称	类型	描述
node_name	text	节点名称。
remote_name	text	连接对端节点的名称。
remote_host	text	连接对端IP的地址。
stream_num	integer	当前物理连接使用的stream逻辑连接数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
min_delay	integer	当前物理连接一分钟内探测到的最小时延。单位：微秒。 **说明** 负数结果无效，请重新等待时延状态更新后再执行查询。
average	integer	当前物理连接一分钟内探测时延的平均值。单位：微秒。
max_delay	integer	当前物理连接一分钟内探测到的最大时延。单位：微秒。

7.3.4.15 PGXC_COMM_RECV_STREAM

PGXC_COMM_RECV_STREAM视图展示所有DN上的通信库接收流状态。

表 7-221 PGXC_COMM_RECV_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的tcp通信socket。
state	text	通信流当前的状态。 ● UNKNOWN：当前逻辑连接状态未知。 ● READY：逻辑连接已就绪。 ● RUN：逻辑连接发送报文正常。 ● HOLD：逻辑连接发送报文等待中。 ● CLOSED：关闭逻辑连接。 ● TO_CLOSED：将会关闭逻辑连接。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
recv_bytes	bigint	通信流接收的数据总量。单位：Byte。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
time	bigint	通信流当前生命周期使用时长。单位：ms。
speed	bigint	通信流的平均接收速率。单位：Byte/s。
quota	bigint	通信流当前的通信配额值。单位：Byte。
buff_usize	bigint	通信流当前缓存的数据大小。单位：Byte。

7.3.4.16 PGXC_COMM_SEND_STREAM

PGXC_COMM_SEND_STREAM视图展示所有DN上的通信库发送流状态。

表 7-222 PGXC_COMM_SEND_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的tcp通信socket。
state	text	通信流当前的状态。 状态值选择如下： ● UNKNOWN：当前连接状态未知。 ● READY：连接已就绪。 ● RUN：连接当前正常发送报文。 ● HOLD：连接正在等待报文发送。 ● CLOSED：连接关闭。 ● TO_CLOSED：将会关闭连接。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
send_bytes	bigint	通信流发送的数据总量。单位：Byte。
time	bigint	通信流当前生命周期使用时长。单位：ms。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
speed	bigint	通信流的平均发送速率。单位：Byte/s。
quota	bigint	通信流当前的通信配额值。单位：Byte。
wait_quota	bigint	通信流等待quota值产生的额外时间开销。单位：ms。

7.3.4.17 PGXC_COMM_STATUS

PGXC_COMM_STATUS视图展示所有DN的通信库状态。

表 7-223 PGXC_COMM_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
rxpck_rate	integer	节点通信库接收速率。单位：Byte/s。
txpck_rate	integer	节点通信库发送速率。单位：Byte/s。
rxkbyte_rate	bigint	bigint节点通信库接收速率。单位：KByte/s。
txkbyte_rate	bigint	bigint节点通信库发送速率。单位：KByte/s。
buffer	bigint	cmailbox的缓存大小。
memkbyte_libcomm	bigint	libcomm进程通信内存大小。单位：Byte。
memkbyte_libpq	bigint	libpq进程通信内存大小。单位：Byte。
used_pm	integer	postmaster线程实时使用率。
used_sflow	integer	gs_sender_flow_controller线程实时使用率。
used_rflow	integer	gs_receiver_flow_controller线程实时使用率。
used_rloop	integer	多个gs_receivers_loop线程中高的实时使用率。
stream	integer	当前使用的逻辑连接总数。

7.3.5 段页式存储

7.3.5.1 GLOBAL_GS_SEG_DATAFILES

GLOBAL_GS_SEG_DATAFILES在CN上执行，用于查看所有节点所有表空间的数据文件信息。只支持管理员权限用户查询。

版权所有 © 华为云计算技术有限公司

表 7-224 GLOBAL_GS_SEG_DATAFILES 字段

名称	类型	描述
node_name	text	节点名称。
file_name	text	数据文件名。例如，base/17467/2_fsm。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
forknum	integer	段对象的分支类型。取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。
tablespace_name	name	数据文件所属的表空间名称。
contents	text	数据文件的存储内容。取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
extent_size	integer	数据文件的扩展大小。
meta_blocks	bigint	数据文件的已分配的元页面数。
data_blocks	bigint	数据文件的已分配的数据页面数。
total_blocks	bigint	数据文件的总物理页面数。
high_water_mark	bigint	数据文件使用页数的高水位线。
utilization	real	使用的block数占总block数的百分比。即(data_blocks+meta_blocks)/total_blocks。

7.3.5.2 GLOBAL_GS_SEG_EXTENTS

GLOBAL_GS_SEG_EXTENTS在CN上执行，查看所有节点所有表空间的扩展信息。该视图输出用户段对象的所有扩展，包含1号文件中的segment head、fork head、level1 page，2~5号文件中的data extent。只支持管理员权限用户查询。

版权所有 © 华为云计算技术有限公司

表 7-225 GLOBAL_GS_SEG_EXTENTS 字段

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	段对象所属的表空间。
bucketnode	integer	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
head_block_id	bigint	段头页面号。
extent_id	integer	逻辑扩展号。
file_id	integer	扩展所在的数据文件标识。取值范围：[1,5]的int4值。
forknum	integer	段对象的分支。取值范围： • 0表示main。 • 1表示fsm。 • 2表示vm。
block_id	bigint	扩展所在的数据文件中的起始页面号。
blocks	integer	扩展大小。取值：1，8，128，1024，4096。
usage_type	text	扩展的使用类型。取值范围： • segment head表示段头。 • fork head表示分支头。 • level1 page表示level页面。 • data extent表示数据扩展。

7.3.5.3 GLOBAL_GS_SEG_SEGMENTS

GLOBAL_GS_SEG_SEGMENTS在CN上执行，查看所有节点所有表空间的段信息，包含表、索引、TOAST、TOAST INDEX段及fsm fork、vm fork段。只支持管理员权限用户查询。

版权所有 © 华为云计算技术有限公司

表 7-226 GLOBAL_GS_SEG_SEGMENTS 字段

名称	类型	描述
node_name	text	节点名称。
schema_name	name	段对象所属的名称空间。
segment_name	name	段对象名，来源：pg_class、pg_partition.relname。
partition_name	name	段对象分区名称，非分区为NULL。来源：pg_partition.relname。
forknum	integer	段对象的分支类型，取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。
segment_type	text	段对象类型。取值范围： ● table：段页式普通表。 ● table partition：段页式分区表（主表和子表）、段页式二级分区表（其中的一级分区表）。 ● table subpartition：段页式二级分区表（其中的顶层节点表和二级分区表）。 ● index：段页式普通表索引。 ● index partition：段页式分区表、二级分区表的索引。 ● global partition index：段页式分区表、二级分区表的全局索引。 ● toast：段页式toast表。 ● toast index：段页式toast表的索引。
tablespace_name	name	段对象所属的表空间。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
head_block_id	bigint	段头页面号。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
contents	text	数据文件的存储内容，取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
table_name	name	段对象所属的基表名。
blocks	bigint	段对象的逻辑页面数。
total_blocks	bigint	段对象的物理页面数。
extents	integer	段对象的逻辑扩展数。
total_extents	integer	段对象的物理扩展数。
head_lsn	text	段头lsn标识。
level0_slots	bigint[]	段扩展映射level0槽位数组。
level1_slots	bigint[]	段扩展映射level1槽位数组。
fork_head	bigint[]	段对象的分支段头数组。

7.3.5.4 GLOBAL_GS_SEG_SPC_EXTENTS

GLOBAL_GS_SEG_SPC_EXTENTS在CN上执行，获取所有节点所有表空间已使用的扩展信息。输出包含segment head、fork head、level1 page、data extent。只支持管理员权限用户查询。

表 7-227 GLOBAL_GS_SEG_SPC_EXTENTS

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	表空间名称。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bucketnode	integer	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
forknum	integer	段对象的分支类型。取值范围： • 0表示main fork。 • 1表示fsm fork。 • 2表示vm fork。
block_id	bigint	数据扩展的起始页面号。
blocks	integer	数据扩展大小。取值：1，8，128，1024，4096。
contents	text	数据文件的存储内容。取值范围： • permanent：永久。 • unlogged：不记录日志。 • temporary：全局临时。 • temporary2：本地临时。
in_used	text	是否已分配。取值Y/N。
mapblock_location	text	扩展在map block中的位置，格式：(page_id, offset)。
head_file_id	integer	段头文件标识。
head_block_id	bigint	段头页面号。
usage_type	text	扩展的使用类型，取值范围： • segment head表示段头。 • fork head表示分支头。 • level1 page表示level页面。 • data extent表示数据扩展。
remain_flag	text	是否为shrink残留扩展。取值Y/N。
special_data	integer	扩展对应反向指针的特殊数据区。
ipblock_location	text	扩展反向指针位置，格式：(block_id, offset)。

版权所有 © 华为云计算技术有限公司

7.3.5.5 GLOBAL_GS_SEG_SPC_SEGMENTS

GLOBAL_GS_SEG_SPC_SEGMENTS在CN上执行，获取所有节点所有表空间已使用的段信息。只支持管理员权限用户查询。

表 7-228 GLOBAL_GS_SEG_SPC_SEGMENTS 字段

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	表空间名称。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
forknum	integer	段对象的分支类型，取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。
block_id	bigint	数据扩展的起始页面号。
blocks	integer	数据扩展大小，取值：1、8、128、1024、4096。
contents	text	数据文件的存储内容。取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
in_used	text	是否已分配。取值Y/N。
mapblock_location	text	扩展在map block中的位置，格式：(page_id, offset)。
head_file_id	integer	段头文件标识。
head_block_id	bigint	段头页面号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
usage_type	text	扩展的使用类型，取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。
remain_flag	text	是否为shrink残留扩展，取值Y/N。
special_data	integer	扩展对应反向指针的特殊数据区。
ipblock_location	text	扩展反向指针位置，格式：(block_id, offset)。

7.3.5.6 GS_SEG_DATAFILES

GS_SEG_DATAFILES查看所有表空间的数据文件信息。只支持管理员权限用户查询。

表 7-229 GS_SEG_DATAFILES 字段

名称	类型	描述
node_name	text	节点名称。
file_name	text	数据文件名，例如，base/17467/2_fsm。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
forknum	integer	段对象的分支类型。取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。
tablespace_name	name	数据文件所属的表空间名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
contents	text	数据文件的存储内容。取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
extent_size	integer	数据文件的扩展大小。
meta_blocks	bigint	数据文件已分配的元页面数。
data_blocks	bigint	数据文件已分配的数据页面数。
total_blocks	bigint	数据文件的总物理页面数。
high_water_mark	bigint	数据文件使用页数的高水位线。
utilization	real	使用的block数占总block数的百分比，即(data_blocks+meta_blocks)/total_blocks。

7.3.5.7 GS_SEG_DATAFILE_LAYOUT

GS_SEG_DATAFILE_LAYOUT查看段页式1~5号数据文件的静态布局。只支持管理员权限用户查询。

表 7-230 GS_SEG_DATAFILE_LAYOUT 字段

名称	类型	描述
version	text	段页式版本。默认：1.0。
seg_storage_type	text	● segment表示普通段页式数据。 ● hashbucket表示hashbucket数据。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
section_id	integer	数据文件划分的数据区号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
section_type	text	数据文件区类型。取值范围： ● file_header表示文件头。 ● spc_header表示空间头。 ● map_header表示映射头。 ● map_pages表示映射页面。 ● ip_pages(inverse pointer pages)表示反向指针页面。 ● data_pages表示数据页面。
page_start	bigint	数据区域起始页面号。
page_end	bigint	数据区域结束页面号。
page_count	bigint	数据区域页面数。
total_size	bigint	数据区的大小。单位为字节。

7.3.5.8 GS_SEG_EXTENTS

GS_SEG_EXTENTS查看所有表空间的扩展信息。该视图输出用户段对象的所有扩展，包含1号文件中的segment head、fork head、level1 page，2~5号文件中的data extent。只支持管理员权限用户查询。

表 7-231 GS_SEG_EXTENTS 字段

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	段对象所属的表空间。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
head_block_id	bigint	段头页面号。
extent_id	integer	逻辑扩展号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
file_id	integer	扩展所在的数据文件标识。取值范围：[1,5]的int4值。
forknum	integer	段对象的分支类型。取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。
block_id	bigint	扩展所在的数据文件中的起始页面号。
blocks	integer	扩展大小。取值：1，8，128，1024，4096。
usage_type	text	扩展的使用类型。取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。

7.3.5.9 GS_SEG_SEGMENTS

GS_SEG_SEGMENTS查看所有表空间的段信息，包含表、索引、TOAST、TOAST INDEX段及fsm fork、vm fork段。只支持管理员权限用户查询。

表 7-232 GS_SEG_SEGMENTS 字段

名称	类型	描述
node_name	text	节点名称。
schema_name	name	段对象所属的名称空间。
segment_name	name	段对象名，来源：pg_class、pg_partition.relname。
partition_name	name	段对象分区名称，非分区为NULL。来源：pg_partition.relname。
forknum	integer	段对象的分支类型。取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
segment_type	text	段对象类型。取值范围： ● table：段页式普通表。 ● table partition：段页式分区表（主表和子表）、段页式二级分区表（其中的一级分区表）。 ● table subpartition：段页式二级分区表（其中的顶层节点表和二级分区表）。 ● index：段页式普通表索引。 ● index partition：段页式分区表、二级分区表的索引。 ● global partition index：段页式分区表、二级分区表的全局索引。 ● toast：段页式toast表。 ● toast index：段页式toast表的索引。
tablespace_name	name	段对象所属的表空间。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
head_block_id	bigint	段头页面号。
contents	text	数据文件的存储内容。取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
table_name	name	段对象所属的基表名。
blocks	bigint	段对象的逻辑页面数。
total_blocks	bigint	段对象的物理页面数。
extents	integer	段对象的逻辑扩展数。
total_extents	integer	段对象的物理扩展数。
head_lsn	text	段头lsn标识。
level0_slots	bigint[]	段扩展映射level0槽位数组。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
level1_slots	bigint[]	段扩展映射level1槽位数组。
fork_head	bigint[]	段对象的分支段头数组。

7.3.5.10 GS_SEG_SEGMENT_LAYOUT

GS_SEG_SEGMENT_LAYOUT输出段页式文件的静态布局。只支持管理员权限用户查询。

表 7-233 GS_SEG_SEGMENT_LAYOUT 字段

名称	类型	描述
version	text	段页式版本。默认：1.0。
section_id	integer	段划分的数据区号。
section_type	text	段数据区扩展类型。取值范围： ● meta：段头。 ● data：数据。
extent_size	integer	扩展大小。单位为字节。
extent_page_count	integer	扩展页面数。
extent_count_start	bigint	起始扩展号。
extent_count_end	bigint	终止扩展号。
total_size	bigint	段数据区的大小。单位为字节。

7.3.5.11 GS_SEG_SPC_EXTENTS

GS_SEG_SPC_EXTENTS获取所有表空间已使用的扩展信息。输出包含segment head、fork head、level1 page、data extent。只支持管理员权限用户查询。

表 7-234 GS_SEG_SPC_EXTENTS

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	表空间名称。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bucketnode	integer	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
forknum	integer	数据文件分支。
block_id	bigint	数据扩展的起始页面号。
blocks	integer	数据扩展大小。取值：1、8、128、1024、4096。
contents	text	数据文件的存储内容，取值范围： • permanent：永久。 • unlogged：不记录日志。 • temporary：全局临时。 • temporary2：本地临时。
in_used	text	是否已分配。取值Y/N。
mapblock_location	text	扩展在map block中的位置，格式：(page_id, offset)。
head_file_id	integer	段头文件标识。
head_block_id	bigint	段头页面号。
usage_type	text	扩展的使用类型，取值范围： • segment head表示段头。 • fork head表示分支头。 • level1 page表示level页面。 • data extent表示数据扩展。
remain_flag	text	是否为shrink残留扩展。取值Y/N。
special_data	integer	扩展对应反向指针的特殊数据区。
ipblock_location	text	扩展反向指针位置，格式：(block_id, offset)。

7.3.5.12 GS_SEG_SPC_SEGMENTS

GS_SEG_SPC_SEGMENTS获取所有表空间已使用的段信息。只支持管理员权限用户查询。

版权所有 © 华为云计算技术有限公司

表 7-235 GS_SEG_SPC_SEGMENTS 字段

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	表空间名称。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
forknum	integer	段对象的分支类型。取值范围： ● 0表示main fork。 ● 1表示fsm fork。 ● 2表示vm fork。
block_id	bigint	数据扩展的起始页面号。
blocks	integer	数据扩展大小。取值：1、8、128、1024、4096。
contents	text	数据文件的存储内容。取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
in_used	text	是否已分配。取值Y/N。
mapblock_location	text	扩展在map block中的位置，格式：(page_id, offset)。
head_file_id	integer	段头文件标识。
head_block_id	bigint	段头页面号。
usage_type	text	扩展的使用类型，取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。
remain_flag	text	是否为shrink残留扩展。取值Y/N。
special_data	integer	扩展对应反向指针的特殊数据区。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
ipblock_location	text	扩展反向指针位置，格式：(block_id, offset)。

7.3.5.13 GS_SEG_SPC_REMAIN_EXTENTS

GS_SEG_SPC_REMAIN_EXTENTS获取索引表空间残留的孤立的扩展信息。这些扩展信息不包含在残留段中，可以作为独立的清理单元。只支持管理员权限用户查询。

表 7-236 GS_SEG_SPC_REMAIN_EXTENTS 字段

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	表空间名称。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
bucketnode	integer	• 0~1023表示hash bucket表的bucketnode。 • 1024表示段页式普通表的bucketnode。 • 1025表示段页式全局临时表的bucketnode。 • 1026表示段页式unlogged表的bucketnode。 • 1027表示段页式本地临时表的bucketnode。
forknum	integer	段对象的分支类型。取值范围： • 0表示main fork。 • 1表示fsm fork。 • 2表示vm fork。
block_id	bigint	数据扩展的起始页面号。
blocks	integer	数据扩展大小。取值：1、8、128、1024、4096。
contents	text	数据文件的存储内容。取值范围： • permanent：永久。 • unlogged：不记录日志。 • temporary：全局临时。 • temporary2：本地临时。
in_used	text	是否已分配。取值Y/N。
mapblock_location	text	扩展在map block中的位置，格式：（page_id, offset）。
head_file_id	integer	段头文件标识。
head_block_id	bigint	段头页面号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
usage_type	text	扩展的使用类型。取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。
remain_flag	text	是否为shrink残留扩展。取值Y/N。
special_data	integer	扩展对应反向指针的特殊数据区。
ipblock_location	text	扩展反向指针位置，格式：（block_id，offset）。

7.3.5.14 GS_SEG_SPC_REMAIN_SEGMENTS

GS_SEG_SPC_REMAIN_SEGMENTS获取索引表空间残留的段信息，包含main fork、fsm fork、vm fork段。只支持管理员权限用户查询。

表 7-237 GS_SEG_SPC_REMAIN_SEGMENTS 字段

名称	类型	描述
node_name	text	节点名称。
tablespace_name	name	表空间名称。
file_id	integer	数据文件标识。取值范围：[1,5]的int4值。
bucketnode	integer	● 0~1023表示hash bucket表的bucketnode。 ● 1024表示段页式普通表的bucketnode。 ● 1025表示段页式全局临时表的bucketnode。 ● 1026表示段页式unlogged表的bucketnode。 ● 1027表示段页式本地临时表的bucketnode。
forknum	integer	数据文件分支。
block_id	bigint	数据扩展的起始页面号。
blocks	integer	数据扩展大小。取值：1、8、128、1024、4096。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
contents	text	数据文件的存储内容。取值范围： ● permanent：永久。 ● unlogged：不记录日志。 ● temporary：全局临时。 ● temporary2：本地临时。
in_used	text	是否已分配。取值Y/N。
mapblock_location	text	扩展在map block中的位置，格式：（page_id, offset）。
head_file_id	integer	段头文件标识。
head_block_id	bigint	段头页面号。
usage_type	text	扩展的使用类型。取值范围： ● segment head表示段头。 ● fork head表示分支头。 ● level1 page表示level页面。 ● data extent表示数据扩展。
remain_flag	text	是否为shrink残留扩展。取值Y/N。
special_data	integer	扩展对应反向指针的特殊数据区。
ipblock_location	text	扩展反向指针位置，格式：（block_id, offset）。

说明

由于事务可见性的判断机制，当事务A创建了段页式对象且事务未提交时，其它会话执行GS_SEG_SPC_REMAIN_SEGMENTS时会查询到事务A产生的段。实际这个段并非残留，自动或手动执行残留清理函数也不会被清理。待事务A结束之后才能继续判定这个段是否为残留。

7.3.6 SPM 计划管理

7.3.6.1 GS_SPM_SQL_BASELINE

GS_SPM_SQL_BASELINE是SPM计划管理特性下的一个系统视图，当前该特性在分布式下不支持。该视图用于查看当前用户的baseline信息，普通用户以上权限均可以访问该视图。

表 7-238 GS_SPM_SQL_BASELINE 字段

名称	类型	描述
sql_namespace	oid	Schema的oid。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sql_hash	bigint	当前Schema中SQL的唯一标识。
plan_hash	bigint	当前SQL中的plan的唯一标识。
outline	text	outline文本，可固定当前计划的一组Hint。
status	text	计划的状态。取值范围： ● UNACC：表示未接受的计划。 ● ACC：表示已经接受的计划。 ● FIXED：一种特殊的ACC状态的计划，计划的匹配优先级高于ACC状态的计划。
gplan	boolean	当前outline对应的计划是否是gplan。
cost	double precision	计划的总代价。
sql_text	text	SQL的文本串。
param_num	integer	SQL的参数数量。
source	text	baseline的来源。
creation_time	timestamp with time zone	baseline的创建时间。
modification_time	timestamp with time zone	baseline的修改时间。
jump_intercept_cnt	bigint	当前baseline拦截计划跳变次数。
invalid	boolean	当前baseline是否无效。
namespace_oids	text	当前search_path中的OID列表，其中OID满足以下特征： ● 用户所拥有权限的Schema的OID。 ● 存在的Schema的OID。 ● 非本地临时表所在Schema的OID。

7.3.6.2 GS_SPM_SQL_PARAM

GS_SPM_SQL_PARAM是SPM计划管理特性下的系统视图，当前该特性在分布式下不支持。该视图用于查看当前用户的SQL参数信息，普通用户以上权限均可以访问该视图。

版权所有 © 华为云计算技术有限公司

表 7-239 GS_SPM_SQL_PARAM 字段

名称	类型	描述
sql_namespace	oid	Schema的oid。
sql_hash	bigint	当前Schema中SQL的唯一标识。
sql_text	text	SQL的文本串。
position	integer	参数在SQL中的位置索引，从0开始。
datatype	integer	参数类型的oid。
datatype_string	text	参数类型的字符串形式。
value_string	text	参数值的字符串形式。
is_null	boolean	参数值是否是NULL。
hash_value	bigint	参数值的哈希值。
creation_time	timestamp with time zone	本记录的创建时间。
namespace_oids	text	当前search_path中的OID列表，其中OID满足以下特征： ● 用户所拥有权限的Schema的OID。 ● 存在的Schema的OID。 ● 非本地临时表所在Schema的OID。

7.3.6.3 GS_SPM_SQL_EVOLUTION

GS_SPM_SQL_EVOLUTION是SPM计划管理特性下的系统视图，当前该特性在分布式下不支持。该视图用于查看当前用户的计划演进结果信息，普通用户以上权限均可以访问该视图。

表 7-240 GS_SPM_SQL_EVOLUTION 字段

名称	类型	描述
sql_namespace	oid	Schema的oid。
sql_hash	bigint	当前Schema中SQL的唯一标识。
plan_hash	bigint	当前SQL中的plan的唯一标识。
better	boolean	是否是正向收益演进。
status	text	演进过程中是否出现异常。取值范围： ● SUCCESS：演进成功。 ● FAILED：演进失败。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
refer_plan	bigint	报告生成的主要参考的plan hash。
sql_text	text	SQL的文本串。
outline	text	当前计划的Hint字符串列表。
reason	text	演进报告的内容。
gplan	boolean	是否是gplan。
creation_time	timestamp with time zone	创建演进结果的时间。
namespace_oids	text	当前search_path中的OID列表，其中OID满足以下特征： ● 用户所拥有权限的Schema的OID。 ● 存在的Schema的OID。 ● 非本地临时表所在Schema的OID。

7.3.6.4 GS_SPM_SYS_BASELINE

GS_SPM_SYS_BASELINE是SPM计划管理特性下的系统视图，该特性在分布式下不支持。该视图用于查看数据库中的baseline信息，SYSADMIN和初始用户权限均可以访问该视图。

表 7-241 GS_SPM_SYS_BASELINE 字段

名称	类型	描述
sql_hash	bigint	SPM中SQL的唯一标识。
plan_hash	bigint	当前SQL中的plan的唯一标识。
unique_sql_id	bigint	数据库中SQL的唯一标识
outline	text	outline文本，可固定当前计划的一组Hint。
status	integer	计划的状态。取值范围： ● 0（UNACC）：表示未接受的计划。 ● 1（ACC）：表示已经接受的计划。 ● 2（FIXED）：一种特殊的ACC状态的计划，计划的匹配优先级高于ACC状态的计划。
gplan	boolean	当前outline对应的计划是否是gplan。
cost	double precision	计划的总代价。
sql_text	text	SQL的文本串。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
param_num	integer	SQL的参数数量。
source	text	baseline的来源。
creation_time	timestamp with time zone	baseline的创建时间。
modification_time	timestamp with time zone	baseline的修改时间。
jump_intercept_cnt	bigint	当前baseline拦截计划跳变次数。
invalid	boolean	当前baseline是否无效。
namespace_oids	text	当前search_path中的OID列表，其中OID满足以下特征： ● 用户所拥有权限的Schema的OID。 ● 存在的Schema的OID。 ● 非本地临时表所在Schema的OID。

7.3.7 审计

7.3.7.1 ADM_AUDIT_OBJECT

ADM_AUDIT_OBJECT显示数据库中所有对象的审计跟踪记录，该视图同时存在于pg_catalog和sys schema下。仅拥有AUDITADMIN属性及ADM_AUDIT_OBJECT的SELECT权限的用户才可以查看审计信息；三权分立未开启时，拥有SYSADMIN属性的用户也可以查看审计信息。

表 7-242 ADM_AUDIT_OBJECT 字段

名称	类型	描述
os_username	character varying(255)	暂不支持，值为NULL。
username	character varying(128)	操作被审计的用户的名称，不是用户id。
userhost	character varying(128)	暂不支持，值为NULL。
terminal	character varying(255)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
timestamp	date	在本地数据库会话时区中创建审计跟踪条目的日期和时间（审计会话创建条目的用户登录日期和时间）。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。 **说明** 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该字段的类型会变更为datea。
owner	character varying(128)	受操作影响的对象的创建者。
obj_name	character varying(128)	受操作影响的对象的名称。
action_name	character varying(28)	DBA_AUDIT_TRAIL中的“操作”列中的数字代码对应的动作类型名称。 **说明** GaussDB的action_name字段与ORA数据库审计动作不一致。
new_owner	character varying(128)	暂不支持，值为NULL。
new_name	character varying(128)	暂不支持，值为NULL。
ses_actions	character varying(19)	暂不支持，值为NULL。
comment_text	character varying(4000)	暂不支持，值为NULL。
sessionid	numeric	暂不支持，值为NULL。
entryid	numeric	暂不支持，值为NULL。
statementid	numeric	暂不支持，值为NULL。
returncode	numeric	暂不支持，值为NULL。
priv_used	character varying(40)	暂不支持，值为NULL。
client_id	character varying(128)	暂不支持，值为NULL。
econtext_id	character varying(64)	暂不支持，值为NULL。
session_cpu	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
extended_timestamp	timestamp(6) with time zone	在UTC（协调世界时）时区创建审计跟踪条目的时间戳（审计会话创建条目的用户的登录时间戳）。
proxy_sessionid	numeric	暂不支持，值为NULL。
global_uid	character varying(32)	暂不支持，值为NULL。
instance_number	numeric	暂不支持，值为NULL。
os_process	character varying(16)	暂不支持，值为NULL。
transactionid	text	访问或修改对象的事务的事务标识符。 **说明** GaussDB的transactionid字段与ORA数据库中transactionid数据的类型保持一致。
scn	numeric	暂不支持，值为NULL。
sql_bind	nvarchar2(2000)	暂不支持，值为NULL。
sql_text	character varying	查询的SQL文本。 **说明** GaussDB的sql_text字段为解析后sql描述语句，不完全与执行的sql语句相同。
obj_edition_name	character varying(128)	暂不支持，值为NULL。

7.3.7.2 ADM_AUDIT_SESSION

ADM_AUDIT_SESSION显示所有连接断开数据库的审计信息，GaussDB审计信息主要通过gs_query_audit函数查询，该视图同时存在于PG_CATALOG和SYS schema下。仅拥有AUDITADMIN属性及ADM_AUDIT_SESSION的SELECT权限的用户才可以查看审计信息；三权分立未开启时，拥有SYSADMIN属性的用户也可以查看审计信息。

表 7-243 ADM_AUDIT_SESSION 字段

名称	类型	描述
os_username	character varying(255)	暂不支持，值为NULL。
username	character varying(128)	操作被审计的用户的名称，不是用户id。
userhost	character varying(128)	暂不支持，值为NULL。
terminal	character varying(255)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
timestamp	date	创建审核跟踪条目的日期和时间（用户登录创建条目的日期和时间AUDIT SESSION）。 **说明** 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该字段的类型会变更为datea。
action_name	character varying(28)	DBA_AUDIT_TRAIL中的ACTION列中的数字代码对应的动作类型的名称。 **说明** GaussDB的action_name字段与ORA数据库审计动作不一致。
logoff_time	date	暂不支持，值为NULL。
logoff_lread	numeric	暂不支持，值为NULL。
logoff_pread	numeric	暂不支持，值为NULL。
logoff_lwrite	numeric	暂不支持，值为NULL。
logoff_dlock	numeric	暂不支持，值为NULL。
sessionid	numeric	暂不支持，值为NULL。
returncode	numeric	暂不支持，值为NULL。
client_id	character varying(128)	暂不支持，值为NULL。
session_cpu	numeric	暂不支持，值为NULL。
extended_timestamp	timestamp(6) with time zone	在UTC（协调世界时）时区创建审核跟踪条目的时间戳（创建条目的用户的登录时间戳AUDIT SESSION）。
proxy_sessionid	numeric	暂不支持，值为NULL。
global_uid	character varying(32)	暂不支持，值为NULL。
instance_number	numeric	暂不支持，值为NULL。
os_process	character varying(16)	暂不支持，值为NULL。

7.3.7.3 ADM_AUDIT_STATEMENT

ADM_AUDIT_STATEMENT显示所有GRANT、REVOKE审计跟踪条目，GaussDB审计信息主要通过gs_query_audit函数查询，该视图同时存在于PG_CATALOG和SYS schema下。仅拥有AUDITADMIN属性及ADM_AUDIT_STATEMENT的SELECT权限的用户才可以查看审计信息；三权分立未开启时，拥有SYSADMIN属性的用户也可以查看审计信息。

版权所有 © 华为云计算技术有限公司

表 7-244 ADM_AUDIT_STATEMENT 字段

名称	类型	描述
os_username	character varying(255)	暂不支持，值为NULL。
username	character varying(128)	操作被审计的用户的名称，不是用户id。
userhost	character varying(128)	暂不支持，值为NULL。
terminal	character varying(255)	暂不支持，值为NULL。
timestamp	date	在本地数据库会话时区中创建审计跟踪条目的日期和时间（审计会话创建条目的用户的登录日期和时间）。 **说明** 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该字段的类型会变更为datea。
owner	character varying(128)	受操作影响的对象的创建者。
obj_name	character varying(128)	受操作影响的对象的名称。
action_name	character varying(28)	action列中的数字代码对应的action类型名称。 **说明** GaussDB的action_name字段与ORA数据库审计动作不一致。
new_name	character varying(128)	暂不支持，值为NULL。
obj_privilege	character varying(32)	暂不支持，值为NULL。
sys_privilege	character varying(40)	暂不支持，值为NULL。
admin_option	character varying(1)	暂不支持，值为NULL。
grantee	character varying(128)	暂不支持，值为NULL。
audit_option	character varying(40)	暂不支持，值为NULL。
ses_actions	character varying(19)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
comment_text	character varying(4000)	暂不支持，值为NULL。
sessionid	numeric	暂不支持，值为NULL。
entryid	numeric	暂不支持，值为NULL。
statementid	numeric	暂不支持，值为NULL。
returncode	numeric	暂不支持，值为NULL。
priv_used	character varying(40)	暂不支持，值为NULL。
client_id	character varying(128)	暂不支持，值为NULL。
econtext_id	character varying(64)	暂不支持，值为NULL。
session_cpu	numeric	暂不支持，值为NULL。
extended_timestamp	timestamp(6) with time zone	创建审计跟踪条目的时间戳（创建条目的用户登录时间戳UTC（协调通用）中的审计会话时间）时区。
proxy_sessionid	numeric	暂不支持，值为NULL。
global_uid	character varying(32)	暂不支持，值为NULL。
instance_number	numeric	暂不支持，值为NULL。
os_process	character varying(16)	暂不支持，值为NULL。
transactionid	text	访问或修改对象事务的事务标识符。 **说明** GaussDB的transactionid字段与ORA数据库中transactionid数据的类型保持一致。
scn	numeric	暂不支持，值为NULL。
sql_bind	nvarchar2(2000)	暂不支持，值为NULL。
sql_text	character varying(2000)	查询的SQL文本。 **说明** GaussDB的sql_text字段为解析后sql描述语句，不完全与执行的sql语句相同。
obj_edition_name	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.7.4 ADM_AUDIT_TRAIL

ADM_AUDIT_TRAIL显示所有标准审计跟踪条目，GaussDB审计信息主要通过gs_query_audit函数查询，该视图同时存在于PG_CATALOG和SYS schema下。仅拥有AUDITADMIN属性及ADM_AUDIT_TRAIL的SELECT权限的用户才可以查看审计信息。三权分立未开启时，拥有SYSADMIN属性的用户也可以查看审计信息。

⚠ 注意

在分布式下，gs_query_audit函数只能查询当前节点的审计信息。

表 7-245 ADM_AUDIT_TRAIL 字段

名称	类型	描述
os_username	character varying(255)	暂不支持，值为NULL。
username	character varying(128)	操作被审计的用户的名称，不是用户id。
userhost	character varying(128)	暂不支持，值为NULL。
terminal	character varying(255)	暂不支持，值为NULL。
timestamp	date	在本地数据库会话时区中创建审计跟踪条目的日期和时间（由审计会话创建的条目的用户登录日期和时间）。 **说明** 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该字段的类型会变更为datea。
owner	character varying(128)	受操作影响的对象的创建者。
obj_name	character varying(128)	受操作影响的对象的名称。
action	numeric	暂不支持，值为NULL。
action_name	character varying(28)	action列中的数字代码对应的action类型名称。 **说明** GaussDB的action_name字段与ORA数据库审计动作不一致。
new_owner	character varying(128)	暂不支持，值为NULL。
new_name	character varying(128)	暂不支持，值为NULL。
obj_privilege	character varying(32)	暂不支持，值为NULL。
sys_privilege	character varying(40)	暂不支持，值为NULL。
admin_option	character varying(1)	暂不支持，值为NULL。
grantee	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
audit_option	character varying(40)	暂不支持，值为NULL。
ses_actions	character varying(19)	暂不支持，值为NULL。
logoff_time	date	暂不支持，值为NULL。
logoff_lread	numeric	暂不支持，值为NULL。
logoff_pread	numeric	暂不支持，值为NULL。
logoff_lwrite	numeric	暂不支持，值为NULL。
logoff_dlock	character varying(40)	暂不支持，值为NULL。
comment_text	character varying(4000)	暂不支持，值为NULL。
sessionid	numeric	暂不支持，值为NULL。
entryid	numeric	暂不支持，值为NULL。
statementid	numeric	暂不支持，值为NULL。
returncode	numeric	暂不支持，值为NULL。
priv_used	character varying(40)	暂不支持，值为NULL。
client_id	character varying(128)	暂不支持，值为NULL。
econtext_id	character varying(64)	暂不支持，值为NULL。
session_cpu	numeric	暂不支持，值为NULL
extended_timestamp	timestamp(6) with time zone	创建审计跟踪条目的时间戳（创建条目的用户登录时间戳UTC（协调通用）中的审计会话时间）时区。
proxy_sessionid	numeric	暂不支持，值为NULL。
global_uid	character varying(32)	暂不支持，值为NULL。
instance_number	numeric	暂不支持，值为NULL。
os_process	character varying(16)	暂不支持，值为NULL。
transactionid	text	访问或修改对象事务的事务标识符。 **说明** GaussDB的transactionid字段与ORA数据库中transactionid数据的类型保持一致。
scn	numeric	暂不支持，值为NULL。
sql_bind	nvarchar2(2000)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sql_text	nvarchar2	查询的SQL文本。 **说明** GaussDB的sql_text字段为解析后sql描述语句，不完全与执行的sql语句相同。
obj_edition_name	character varying(128)	暂不支持，值为NULL。
dbid	numeric	暂不支持，值为NULL。
rls_info	clob	暂不支持，值为NULL。
current_user	character varying(128)	暂不支持，值为NULL。

7.3.7.5 GS_AUDITING

GS_AUDITING视图显示对数据库相关操作的所有审计信息。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-246 GS_AUDITING 字段

名称	类型	描述
polname	name	策略名称，需要唯一，不可重复。
pol_type	text	审计策略类型，值为‘access’或者‘privilege’。 ● access：表示审计DML操作。 ● privilege：表示审计DDL操作。
polenabled	boolean	用来表示策略开关是否启动。 ● t（true）：表示启动。 ● f（false）：表示不启动。
access_type	name	DML数据库操作相关类型，例如SELECT、INSERT、DELETE等；或是DDL数据库操作相关类型，例如CREATE、ALTER、DROP等。
label_name	name	资源标签名称。对应系统表gs_auditing_policy中的polname字段。
priv_object	text	数据库资产的路径。
filter_name	text	过滤条件的逻辑字符串。

7.3.7.6 GS_AUDITING_ACCESS

GS_AUDITING_ACCESS视图显示对数据库ACCESS相关操作的所有审计信息。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

版权所有 © 华为云计算技术有限公司

表 7-247 GS_AUDITING_ACCESS 字段

名称	类型	描述
polname	name	策略名称，需要唯一，不可重复。
pol_type	text	审计策略类型。值为‘access’，表示审计DML操作。
polenabled	boolean	用来表示策略开关是否启动。 ● t（true）：表示启动。 ● f（false）：表示不启动。
access_type	name	ACCESS数据库操作相关类型。例如，SELECT、INSERT、DELETE等。
label_name	name	资源标签名称。对应系统表gs_auditing_policy中的polname字段。
access_object	text	数据库资产的路径。
filter_name	text	过滤条件的逻辑字符串。

7.3.7.7 GS_AUDITING_PRIVILEGE

GS_AUDITING_PRIVILEGE视图显示对数据库PRIVILEGE相关操作的所有审计信息。默认只有系统管理员权限或安全策略管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-248 GS_AUDITING_PRIVILEGE 字段

名称	类型	描述
polname	name	策略名称，需要唯一，不可重复。
pol_type	text	审计策略类型。值为‘privilege’，表示审计DDL操作。
polenabled	boolean	用来表示策略启动开关。 ● t（true）：表示启动。 ● f（false）：表示不启动。
access_type	name	PRIVILEGE数据库操作相关类型。例如CREATE、ALTER、DROP等。
label_name	name	资源标签名称。对应系统表gs_auditing_policy中的polname字段。
priv_object	text	包含数据库对象的全称域名。
filter_name	text	过滤条件的逻辑字符串。

版权所有 © 华为云计算技术有限公司

7.3.8 用户和权限管理

7.3.8.1 ADM_COL_PRIVS

ADM_COL_PRIVS视图显示所有的列权限授予信息。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-249 ADM_COL_PRIVS 字段

名称	类型	描述
grantor	character varying(128)	执行授权的用户名。
owner	character varying(128)	对象的所有者。
grantee	character varying(128)	被授予权限的用户或角色的名称。
table_schema	character varying(128)	对象的Schema。
table_name	character varying(128)	对象的名称。
column_name	character varying(128)	列的名称。
privilege	character varying(40)	列的权限。
grantable	character varying(3)	是否授予特权。 ● YES：授予。 ● NO：不授予。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.2 ADM_ROLE_PRIVS

ADM_ROLE_PRIVS视图显示授予所有用户和角色的权限的信息。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-250 ADM_ROLE_PRIVS 字段

名称	类型	描述
grantee	character varying(128)	被授予权限的用户或角色名称。
granted_role	character varying(128)	被授予的角色。
admin_option	character varying(3)	该授权是否包含ADMIN选项。 ● YES：包含ADMIN选项。 ● NO：不包含ADMIN选项。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
delegate_option	character varying(3)	暂不支持，值为NULL。
default_role	character varying(3)	暂不支持，值为NULL。
os_granted	character varying(3)	暂不支持，值为NULL。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.3 ADM_ROLES

ADM_ROLES视图显示数据库角色的相关信息，该视图同时存在于PG_CATALOG和SYS Schema下，默认只有系统管理员权限才可以访问。

表 7-251 ADM_ROLES 字段

名称	类型	描述
role	character varying(128)	角色名称。
role_id	oid	角色ID号。
authentication_type	text	角色的身份验证机制。 ● password：需要密码验证。 ● null：不需要验证。
common	character varying(3)	暂不支持，值为NULL。
oracle_maintained	character varying(1)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。
implicit	character varying(3)	暂不支持，值为NULL。
external_name	character varying(4000)	暂不支持，值为NULL。

7.3.8.4 ADM_SYS_PRIVS

ADM_SYS_PRIVS视图显示授予用户和角色的系统权限信息。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-252 ADM_SYS_PRIVS 字段

名称	类型	描述
grantee	character varying(128)	被授予权限的用户或角色的名称。
privilege	character varying(40)	用户拥有的系统权限或ANY权限。 • 系统权限包括rolsuper、rolinherit、rolcreaterole、rolcreatedb、rolcatupdate、rolcanlogin、rolreplication、rolauditadmin、rolsystemadmin、roluseft、rolmonitoradmin、roloperatoradmin、rolpolicyadmin。 • ANY权限的取值请参考**表 1-402**。
admin_option	character varying(3)	该授权是否包含ADMIN选项。 • YES：包含ADMIN选项。 • NO：不包含ADMIN选项。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.5 ADM_TAB_PRIVS

ADM_TAB_PRIVS视图显示数据库中所有对象的授权信息。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-253 ADM_TAB_PRIVS 字段

名称	类型	描述
grantee	character varying(128)	被授予权限的用户或角色的名称。
owner	character varying(128)	对象的所有者。
table_name	character varying(128)	对象的名称。
grantor	character varying(128)	执行授权的用户名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
privilege	character varying(40)	对象上的权限，包括USAGE、UPDATE、DELETE、INSERT、CONNECT、SELECT、EXECUTE。
grantable	character varying(3)	该授权是否包含GRANT选项。 ● YES：包含GRANT选项。 ● NO：不包含GRANT选项。
type	character varying(24)	对象的类型。包括NODE GROUP、COLUMN_ENCRYPTION_KEY、PACKAGE、COLUMN、TABLE、VIEW、SEQUENCE、TYPE、INDEX、DATABASE、DIRECTORY、FOREIGN DATA WRAPPER、FOREIGN SERVER、LANGUAGE、SCHEMA、TEMPLATE、FUNCTION、PROCEDURE、TABLESPACE。
hierarchy	character varying(3)	暂不支持，值为NULL。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.6 ADM_USERS

ADM_USERS视图显示所有数据库用户的信息。默认只有系统管理员权限可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-254 ADM_USERS 字段

名称	类型	描述
username	character varying(128)	用户名称。
user_id	oid	用户ID。
account_status	character varying(32)	账户状态。 ● NULL：该账户是拥有最高权限的初始系统管理员。 ● 0：正常状态。 ● 1：由于登录失败次数超过阈值被锁定了一定的时间。 ● 2：被管理员锁定。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
lock_date	timestamp with time zone	默认显示账户的创建日期，如果账户被管理员锁定，或者登录失败次数超过阈值被锁定，则显示账户被锁定的日期。初始系统管理员该字段为NULL。
expiry_date	timestamp with time zone	账户的到期日期。
default_tablespace	character varying(4000)	数据的默认表空间。
temporary_tablespace	character varying(4000)	临时表的默认表空间名称或表空间组的名称。
local_temp_tablespace	character varying(30)	暂不支持，默认值为NULL。
created	timestamp with time zone	用户创建日期。
profile	character varying(128)	暂不支持，默认值为NULL。
initial_rsrc_consumer_group	character varying(128)	暂不支持，默认值为NULL。
external_name	character varying(4000)	暂不支持，默认值为NULL。
password_versions	character varying(12)	显示账户密码的加密方式，取值为MD5、SHA256或SM3。
editions_enabled	character varying(1)	暂不支持，默认值为NULL。
authentication_type	text	指示用户的身份验证机制。
proxy_only_connect	character varying(1)	暂不支持，默认值为NULL。
common	character varying(3)	暂不支持，默认值为NULL。
last_login	timestamp with time zone	用户最后一次登录的时间。
oracle_maintained	character varying(1)	暂不支持，默认值为NULL。
inherited	character varying(3)	暂不支持，默认值为NULL。
default_collation	character varying(100)	用户Schema的默认字符序。
implicit	character varying(3)	暂不支持，默认值为NULL。
all_shard	character varying(3)	暂不支持，默认值为NULL。
password_change_date	timestamp with time zone	用户上次设置密码的日期。

版权所有 © 华为云计算技术有限公司

7.3.8.7 DB_COL_PRIVS

DB_COL_PRIVS视图显示以下授权信息：

- 当前用户作为对象所有者、授予者或被授予者时的列权限授予信息。
- 已启用角色或PUBLIC角色作为被授予者时的列权限授予信息。

默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-255 DB_COL_PRIVS 字段

名称	类型	描述
grantor	character varying(128)	执行授权的用户名。
owner	character varying(128)	对象的所有者。
grantee	character varying(128)	被授予权限的用户或角色的名称。
table_schema	character varying(128)	对象的Schema。
table_name	character varying(128)	对象的名称。
column_name	character varying(128)	列的名称。
privilege	character varying(40)	列上的权限。
grantable	character varying(3)	是否授予特权。 • YES：授予特权。 • NO：不授予特权。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.8 DB_DIRECTORIES

DB_DIRECTORIES功能为查看当前用户有操作权限的所有目录对象，管理员用户能够查询到所有的目录对象，普通用户只能查询当前有操作权限的目录对象。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-256 DB_DIRECTORIES 字段

名称	类型	描述
owner	oid	目录对象的所属者OID。
directory_name	name	目录对象名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
directory_path	text	目录对象所代表的目录路径。
origin_con_id	character varying(256)	目录对象创建时所在容器的ID。 暂不支持，值为NULL。

7.3.8.9 DB_TAB_PRIVS

DB_TAB_PRIVS视图记录当前用户可访问的所有对象的授权信息。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-257 DB_TAB_PRIVS 字段

名称	类型	描述
grantor	character varying(128)	执行授权的用户名。
grantee	character varying(128)	被授予权限的用户或角色的名称。
table_schema	character varying(128)	对象的模式。
table_name	character varying(128)	对象的名称。
privilege	character varying(40)	对象上的权限，包括USAGE、UPDATE、DELETE、INSERT、CONNECT、SELECT、EXECUTE。
grantable	character varying(3)	该授权是否包含GRANT选项。 ● YES：包含GRANT选项。 ● NO：不包含GRANT选项。
type	character varying(24)	对象的类型，包括NODE GROUP、COLUMN_ENCRYPTION_KEY、PACKAGE、COLUMN、TABLE、VIEW、SEQUENCE、TYPE、INDEX、DATABASE、DIRECTORY、FOREIGN DATA WRAPPER、FOREIGN SERVER、LANGUAGE、SCHEMA、TEMPLATE、FUNCTION、PROCEDURE、TABLESPACE。
hierarchy	character varying(3)	暂不支持，值为NULL。
common	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.10 DB_USERS

DB_USERS视图存储记录数据库中所有用户，但不对用户信息进行详细的描述。默认只有系统管理员可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-258 DB_USERS 字段

名称	类型	描述
user_id	oid	用户的OID。
username	name	用户的名称。

7.3.8.11 GS_DB_PRIVILEGES

GS_DB_PRIVILEGES系统视图记录ANY权限的授予情况，每条记录对应一条授权信息。

表 7-259 GS_DB_PRIVILEGES 字段

名称	类型	描述
rolename	name	用户名。
privilege_type	text	用户拥有的ANY权限，取值请参见**表1-402**。
admin_option	text	是否具有privilege_type列记录的ANY权限的再授权权限。 • yes：具有。 • no：不具有。

7.3.8.12 GS_LABELS

GS_LABELS视图显示所有已配置的资源标签信息。需要有系统管理员或安全策略管理员权限才可以访问此视图。

表 7-260 GS_LABELS 字段

名称	类型	描述
labelname	name	资源标签的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
labeltype	name	资源标签的类型。对应系统表GS_POLICY_LABEL中的labeltype字段。
fqdntype	name	数据库资源的类型。如table、schema、index等。
schemaname	name	数据库资源所属的schema名称。
fqdnname	name	数据库资源名称。
columnname	name	数据库资源列名称。若标记的数据库资源不为表的列则该项为空。

7.3.8.13 MY_COL_PRIVS

MY_COL_PRIVS视图显示当前用户作为对象所有者、授权者或被授予者时的列权限授予信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-261 MY_COL_PRIVS 字段

名称	类型	描述
grantor	character varying(128)	执行授权的用户名。
owner	character varying(128)	对象的所有者。
grantee	character varying(128)	被授予权限的用户或角色的名称。
table_schema	character varying(128)	对象的Schema。
table_name	character varying(128)	对象的名称。
column_name	character varying(128)	列的名称。
privilege	character varying(40)	列的权限。
grantable	character varying(3)	是否授予特权。 • YES：授予特权。 • NO：不授予特权。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.14 MY_ROLE_PRIVS

MY_ROLE_PRIVS视图显示授予当前用户角色（包括public角色）的权限的信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-262 MY_ROLE_PRIVS 字段

名称	类型	描述
grantee	character varying(128)	被授予权限的用户或角色名称。
granted_role	character varying(128)	被授予的角色。
admin_option	character varying(3)	该授权是否包含ADMIN选项。 • YES：包含ADMIN选项。 • NO：不包含ADMIN选项。
delegate_option	character varying(3)	暂不支持，值为NULL。
default_role	character varying(3)	暂不支持，值为NULL。
os_granted	character varying(3)	暂不支持，值为NULL。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.15 MY_SYS_PRIVS

MY_SYS_PRIVS视图显示授予当前用户的系统权限信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-263 MY_SYS_PRIVS 字段

名称	类型	描述
grantee	character varying(128)	被授予权限的用户或角色的名称。
privilege	character varying(40)	用户拥有的系统权限或ANY权限。 • 系统权限包括rolsuper、rolinherit、rolcreaterole、rolcreatedb、rolcatupdate、rolcanlogin、rolreplication、rolauditadmin、rolsystemadmin、roluseft、rolmonitoradmin、roloperatoradmin、rolpolicyadmin。 • ANY权限的取值请参考**表 1-402**。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
admin_option	character varying(3)	该授权是否包含ADMIN选项。 ● YES：包含ADMIN选项。 ● NO：不包含ADMIN选项。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.16 PG_RLSPOLICIES

PG_RLSPOLICIES视图显示行级访问控制策略的信息。初始化用户和具有sysadmin属性的用户可以查看全部的策略信息，其他用户只能查看自己所拥有表上的策略信息。

表 7-264 PG_RLSPOLICIES 字段

名称	类型	描述
schemaname	name	行级访问控制策略作用的表对象所属的模式名称。
tablename	name	行级访问控制策略作用的表对象名称。
policyname	name	行级访问控制策略名称。
policypermissive	text	行级访问控制策略的表达式拼接方式。取值范围： ● PERMISSIVE：宽容性策略，用OR表达式拼接。 ● RESTRICTIVE：限制性策略，用AND表达式拼接。
policyroles	name[]	行级访问控制策略影响的用户列表，不指定表示影响所有的用户。
policycmd	text	行级访问控制策略影响的SQL操作。
policyqual	text	行级访问控制策略的表达式。

7.3.8.17 GS_RLSPOLICIES

GS_RLSPOLICIES视图显示行级访问控制策略的信息。初始化用户和具有sysadmin属性的用户可以查看全部的策略信息，其他用户只能查看自己所拥有表上的策略信息。

表 7-265 GS_RLSPOLICIES 字段

名称	类型	描述
schemaname	name	行级访问控制策略作用的表对象所属的模式名称。
tablename	name	行级访问控制策略作用的表对象名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
policyname	name	行级访问控制策略名称。
policypermissive	text	行级访问控制策略的表达式拼接方式。 取值范围： ● PERMISSIVE：宽容性策略，用OR表达式拼接。 ● RESTRICTIVE：限制性策略，用AND表达式拼接。
policyroles	name[]	行级访问控制策略影响的用户列表，不指定表示影响所有的用户。
policycmd	text	行级访问控制策略影响的SQL操作。
policyqual	text	行级访问控制策略的表达式。

7.3.8.18 PG_ROLES

PG_ROLES视图显示数据库角色的相关信息。初始化用户和具有sysadmin属性或createrole属性的用户可以查看全部角色的信息，其他用户只能查看自己的信息。

表 7-266 PG_ROLES 字段

名称	类型	引用	描述
rolname	name	-	角色名称。
rolsuper	boolean	-	该角色是否是拥有最高权限的初始系统管理员。 ● t（true）：是。 ● f（false）：否。
rolinherit	boolean	-	该角色是否继承角色的权限。 ● t（true）：是。 ● f（false）：否。
rolcreaterole	boolean	-	该角色是否可以创建其他的角色。 ● t（true）：是。 ● f（false）：否。
rolcreatedb	boolean	-	该角色是否可以创建数据库。 ● t（true）：是。 ● f（false）：否。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
rolcatupdate	boolean	-	该角色是否可以直接更新系统表。只有usesysid=10的初始系统管理员拥有此权限。其他用户无法获得此权限。 ● t（true）：是。 ● f（false）：否。
rolcanlogin	boolean	-	该角色是否可以登录数据库。 ● t（true）：是。 ● f（false）：否。
rolreplication	boolean	-	该角色是否可以复制。 ● t（true）：是。 ● f（false）：否。
rolauditadmin	boolean	-	该角色是否为审计管理员。 ● t（true）：是。 ● f（false）：否。
rolsystemadmin	boolean	-	该角色是否为系统管理员。 ● t（true）：是。 ● f（false）：否。
rolconnlimit	integer	-	对于可以登录的角色，此字段限制了该角色允许发起的最大并发连接数。-1表示无限制。
rolpassword	text	-	密文存储后的用户密码，始终为********。
rolvalidbegin	timestamp with time zone	-	账户的有效开始时间。如果没有设置有效开始时间，则为NULL。
rolvaliduntil	timestamp with time zone	-	账户的有效结束时间。如果没有设置有效结束时间，则为NULL。
rolrespool	name	-	用户所能够使用的resource pool。
rolparentid	oid	PG_AUTHID.rolparentid	用户所在组用户的OID。
roltabspace	text	-	用户永久表存储空间限额，单位KB。
rolconfig	text[]	PG_DB_ROLE_SETTING.setconfig	运行时配置项的默认值。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
oid	oid	PG_AUTHID.oid	角色的ID。
roluseft	boolean	PG_AUTHID.roluseft	角色是否可以操作外表。 ● t（true）：是。 ● f（false）：否。
rolkind	"char"	-	角色类型。 ● n：普通用户，即非永久用户。 ● p：永久用户。
nodegroup	name	-	角色所关联的node_group名称，如果没有关联node group，该值为空。
roltempspace	text	-	用户临时表存储空间的限额，单位KB。
rolspillspace	text	-	用户算子落盘空间的限额，单位KB。
rolmonitoradmin	boolean	-	该角色是否为监控管理员。 ● t（true）：是。 ● f（false）：否。
roloperatoradmin	boolean	-	该角色是否为运维管理员。 ● t（true）：是。 ● f（false）：否。
rolpolicyadmin	boolean	-	该角色是否为安全策略管理员。 ● t（true）：是。 ● f（false）：否。

7.3.8.19 GS_ROLES

GS_ROLES视图显示数据库角色的相关信息。初始化用户和具有sysadmin属性或createrole属性的用户可以查看全部角色的信息，其他用户只能查看自己的信息。

表 7-267 GS_ROLES 字段

名称	类型	引用	描述
rolname	name	-	角色名称。
rolsuper	boolean	-	该角色是否是拥有最高权限的初始系统管理员。 ● t（true）：是。 ● f（false）：否。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
rolinherit	boolean	-	该角色是否继承角色的权限。 ● t（true）：是。 ● f（false）：否。
rolcreaterole	boolean	-	该角色是否可以创建其他的角色。 ● t（true）：是。 ● f（false）：否。
rolcreatedb	boolean	-	该角色是否可以创建数据库。 ● t（true）：是。 ● f（false）：否。
rolcatupdate	boolean	-	该角色是否可以直接更新系统表。只有usesysid=10的初始系统管理员拥有此权限。其他用户无法获得此权限。 ● t（true）：是。 ● f（false）：否。
rolcanlogin	boolean	-	该角色是否可以登录数据库。 ● t（true）：是。 ● f（false）：否。
rolreplication	boolean	-	该角色是否可以复制。 ● t（true）：是。 ● f（false）：否。
rolauditadmin	boolean	-	该角色是否为审计管理员。 ● t（true）：是。 ● f（false）：否。
rolsystemadmin	boolean	-	该角色是否为系统管理员。 ● t（true）：是。 ● f（false）：否。
rolconnlimit	integer	-	对于可以登录的角色，此字段限制了该角色允许发起的最大并发连接数。-1表示无限制。
rolpassword	text	-	密文存储后的用户密码，始终为********。
rolvalidbegin	timestamp with time zone	-	账户的有效开始时间。如果没有设置有效开始时间，则为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
rolvaliduntil	timestamp with time zone	-	账户的有效结束时间。如果没有设置有效结束时间，则为NULL。
rolrespool	name	-	用户所能够使用的resource pool。
rolparentid	oid	PG_AUTHID.rolparentid	用户所在组用户的OID。
roltabspace	text	-	用户永久表存储空间限额，单位KB。
rolconfig	text[]	PG_DB_ROLE_SETTING.setconfig	运行时配置项的默认值。
oid	oid	PG_AUTHID.oid	角色的id。
roluseft	boolean	PG_AUTHID.roluseft	角色是否可以操作外表。 • t（true）：是。 • f（false）：否。
rolkind	"char"	-	角色类型。 • n：普通用户，即非永久用户。 • p：永久用户。
nodegroup	name	-	角色所关联的node_group名称，如果没有关联node group，该值为空。
roltempspace	text	-	用户临时表存储空间的限额，单位KB。
rolspillspace	text	-	用户算子落盘空间的限额，单位KB。
rolmonitoradmin	boolean	-	该角色是否为监控管理员。 • t（true）：是。 • f（false）：否。
roloperatoradmin	boolean	-	该角色是否为运维管理员。 • t（true）：是。 • f（false）：否。
rolpolicyadmin	boolean	-	该角色是否为安全策略管理员。 • t（true）：是。 • f（false）：否。

7.3.8.20 PG_SECLABELS

PG_SECLABELS视图显示安全标签的信息。

版权所有 © 华为云计算技术有限公司

表 7-268 PG_SECLABELS 字段

名称	类型	引用	描述
objoid	oid	任意OID属性	该安全标签所属的对象的OID。
classoid	oid	PG_CLASS.oid	该安全标签所属的对象所在系统表的OID。
objsubid	integer	-	• 对于一个在表字段上的安全标签，该字段是字段序号（引用表本身的objoid和classoid）。 • 对于所有其他对象类型，该字段为0。
objtype	text	-	该标签所属的对象的类型，文本格式。例如： • table：表类型。 • column：列类型。
objnamespace	oid	PG_NAMESPACE.oid	该对象的名称空间的OID，如果不适用，取值为NULL。
objname	text	-	该标签所属的对象的名称，文本格式。
provider	text	PG_SECLABEL.provider	该标签的提供者。
label	text	PG_SECLABEL.label	安全标签名称。

7.3.8.21 GS_SECLABELS

GS_SECLABELS视图显示安全标签的信息。

表 7-269 GS_SECLABELS 字段

名称	类型	引用	描述
objoid	oid	任意OID属性	该安全标签所属的对象的OID。
classoid	oid	PG_CLASS.oid	该安全标签所属的对象所在系统表的OID。
objsubid	integer	-	• 对于一个在表字段上的安全标签，该字段是字段序号（引用表本身的objoid和classoid）。 • 对于所有其他对象类型，该字段为0。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
objtype	text	-	该标签所属的对象的类型，文本格式。 ● table：表类型。 ● column：列类型。
objnamespace	oid	PG_NAMESPACE.oid	该对象的名称空间的OID，如果不适用，取值为NULL。
objname	text	-	该标签所属的对象的名称，文本格式。
provider	text	PG_SECLABEL.provider	该标签的提供者。
label	text	PG_SECLABEL.label	安全标签名称。

7.3.8.22 PG_SHADOW

PG_SHADOW视图显示了所有在PG_AUTHID中标记了rolcanlogin的角色的属性，只有系统管理员权限才可以访问此系统视图。

该视图的信息与PG_USER是基本一致的，区别在于后者对密码做了敏感化处理，统一显示为********。

表 7-270 PG_SHADOW 字段

名称	类型	引用	描述
usename	name	PG_AUTHID.rolname	用户名。
usesysid	oid	PG_AUTHID.oid	用户的id。
usecreatedb	boolean	-	用户是否可以创建数据库。 ● t（true）：表示是。 ● f（false）：表示否。
usesuper	boolean	-	用户是否是系统管理员。 ● t（true）：表示是。 ● f（false）：表示否。
usecatupd	boolean	-	用户是否可以更新视图。即使是系统管理员，如果这个字段值不是t，也不能更新视图。 ● t（true）：表示是。 ● f（false）：表示否。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
userepl	boolean	-	用户是否可以复制数据流。 ● t（true）：表示是。 ● f（false）：表示否。
passwd	text	PG_AUTHID.rolpassword	密码密文，如果没有密码，则为NULL。
valbegin	timestamp with time zone	-	账户的有效开始时间。如果没有设置有效开始时间，则为NULL。
valuntil	timestamp with time zone	-	账户的有效结束时间。如果没有设置有效结束时间，则为NULL。
respool	name	-	用户所在的资源池。
parent	oid	-	父用户OID。
spacelimit	text	-	永久表存储空间的限额，单位KB。
useconfig	text[]	PG_DB_ROLE_SETTING.setconfig	运行时配置项的默认值。
tempspacelimit	text	-	临时表存储空间的限额，单位KB。
spillspacelimit	text	-	算子落盘空间的限额，单位KB。
usemonitoradmin	boolean	-	用户是否是监控管理员。 ● t（true）：表示是。 ● f（false）：表示否。
useoperatoradmin	boolean	-	用户是否是运维管理员。 ● t（true）：表示是。 ● f（false）：表示否。
usepolicyadmin	boolean	-	用户是否是安全策略管理员。 ● t（true）：表示是。 ● f（false）：表示否。

7.3.8.23 GS_SHADOW

GS_SHADOW视图显示了所有在PG_AUTHID中标记了rolcanlogin的角色的属性，只有系统管理员权限才可以访问此系统视图。

该视图的信息与GS_USER是基本一致的，区别在于后者对密码做了敏感化处理，统一显示为********。

版权所有 © 华为云计算技术有限公司

表 7-271 GS_SHADOW 字段

名称	类型	引用	描述
usename	name	PG_AUTHID.rolname	用户名。
usesysid	oid	PG_AUTHID.oid	用户的id。
usecreatedb	boolean	-	用户是否可以创建数据库。 • t（true）：表示是。 • f（false）：表示否。
usesuper	boolean	-	用户是否是系统管理员。 • t（true）：表示是。 • f（false）：表示否。
usecatupd	boolean	-	用户是否可以更新视图。即使是系统管理员，如果这个字段值不是t，也不能更新视图。 • t（true）：表示是。 • f（false）：表示否。
userepl	boolean	-	用户是否可以复制数据流。 • t（true）：表示是。 • f（false）：表示否。
passwd	text	PG_AUTHID.rolpassword	密码密文，如果没有密码，则为NULL。
valbegin	timestamp with time zone	-	账户的有效开始时间。如果没有设置有效开始时间，则为NULL。
valuntil	timestamp with time zone	-	账户的有效结束时间。如果没有设置有效结束时间，则为NULL。
respool	name	-	用户所在的资源池。
parent	oid	-	父用户OID。
spacelimit	text	-	永久表存储空间的限额，单位KB。
useconfig	text[]	PG_DB_ROLE_SETTING.setconfig	运行时配置项的默认值。
tempspacelimit	text	-	临时表存储空间的限额，单位KB。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
spillspacelimit	text	-	算子落盘空间的限额，单位KB。
usemonitoradmin	boolean	-	用户是否是监控管理员。 ● t（true）：表示是。 ● f（false）：表示否。
useoperatoradmin	boolean	-	用户是否是运维管理员。 ● t（true）：表示是。 ● f（false）：表示否。
usepolicyadmin	boolean	-	用户是否是安全策略管理员。 ● t（true）：表示是。 ● f（false）：表示否。

7.3.8.24 PG_USER

PG_USER视图显示数据库用户的信息。默认只有初始化用户和具有sysadmin属性的用户可以查看，其余用户需要赋权后才可以查看。

表 7-272 PG_USER 字段

名称	类型	描述
usename	name	用户名。
usesysid	oid	用户的id。
usecreatedb	boolean	用户是否可以创建数据库。 ● t（true）：表示是。 ● f（false）：表示否。
usesuper	boolean	用户是否是拥有最高权限的初始系统管理员。 ● t（true）：表示是。 ● f（false）：表示否。
usecatupd	boolean	用户是否可以直接更新系统表。只有usesysid=10的初始系统管理员拥有此权限。其他用户无法获得此权限。 ● t（true）：表示是。 ● f（false）：表示否。
userepl	boolean	用户是否可以复制数据流。 ● t（true）：表示是。 ● f（false）：表示否。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
passwd	text	密文存储后的用户密码，始终为********。
valbegin	timestamp with time zone	账户的有效开始时间。如果没有设置有效开始时间，则为NULL。
valuntil	timestamp with time zone	账户的有效结束时间。如果没有设置有效结束时间，则为NULL。
respool	name	用户所在的资源池。
parent	oid	父用户OID。
spacelimit	text	永久表存储空间的限额，单位KB。
useconfig	text[]	运行时配置项的默认值。参考PG_DB_ROLE_SETTING.setconfig。
nodegroup	name	用户关联的node_group名称，如果该用户没有关联node_group，则该字段为空。
tempspacelimit	text	临时表存储空间的限额，单位KB。
spillspacelimit	text	算子落盘空间的限额，单位KB。
usemonitoradmin	boolean	用户是否是监控管理员。 • t（true）：表示是。 • f（false）：表示否。
useoperatoradmin	boolean	用户是否是运维管理员。 • t（true）：表示是。 • f（false）：表示否。
usepolicyadmin	boolean	用户是否是安全策略管理员。 • t（true）：表示是。 • f（false）：表示否。

7.3.8.25 GS_USER

GS_USER视图显示数据库用户的信息。默认只有初始化用户和具有sysadmin属性的用户可以查看，其余用户需要赋权后才可以查看。

表 7-273 GS_USER 字段

名称	类型	描述
usename	name	用户名。
usesysid	oid	用户的id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
usecreatedb	boolean	用户是否可以创建数据库。 ● t（true）：表示是。 ● f（false）：表示否。
usesuper	boolean	用户是否是拥有最高权限的初始系统管理员。 ● t（true）：表示是。 ● f（false）：表示否。
usecatupd	boolean	用户是否可以直接更新系统表。只有usesysid=10的初始系统管理员拥有此权限。其他用户无法获得此权限。 ● t（true）：表示是。 ● f（false）：表示否。
userepl	boolean	用户是否可以复制数据流。 ● t（true）：表示是。 ● f（false）：表示否。
passwd	text	密文存储后的用户密码，始终为********。
valbegin	timestamp with time zone	账户的有效开始时间。如果没有设置有效开始时间，则为NULL。
valuntil	timestamp with time zone	账户的有效结束时间。如果没有设置有效结束时间，则为NULL。
respool	name	用户所在的资源池。
parent	oid	父用户OID。
spacelimit	text	永久表存储空间的限额，单位KB。
useconfig	text[]	运行时配置项的默认值。参考**PG_DB_ROLE_SETTING**.setconfig。
nodegroup	name	用户关联的node_group名称，如果该用户没有关联node_group，则该字段为空。
tempspacelimit	text	临时表存储空间的限额，单位KB。
spillspacelimit	text	算子落盘空间的限额，单位KB。
usemonitoradmin	boolean	用户是否是监控管理员。 ● t（true）：表示是。 ● f（false）：表示否。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
useoperatoradmin	boolean	用户是否是运维管理员。 • t（true）：表示是。 • f（false）：表示否。
usepolicyadmin	boolean	用户是否是安全策略管理员。 • t（true）：表示是。 • f（false）：表示否。

7.3.8.26 PG_USER_MAPPINGS

PG_USER_MAPPINGS视图显示用户映射的信息。所有用户均可查看。

表 7-274 PG_USER_MAPPINGS 字段

名称	类型	引用	描述
umid	oid	PG_USER_MAPPING.oid	用户映射的OID。
srvid	oid	PG_FOREIGN_SERVER.oid	包含这个映射的外部服务器的OID。
srvname	name	PG_FOREIGN_SERVER.srvname	外部服务器的名称。
umuser	oid	PG_AUTHID.oid	被映射的本地角色的OID，如果用户映射是公共的则为0。
usename	name	-	被映射的本地用户的名称。
umoptions	text[]	-	如果当前用户是外部服务器的所有者，则为用户映射指定选项， 使用“keyword=value”字符串，否则为NULL。

7.3.8.27 ROLE_ROLE_PRIVS

ROLE_ROLE_PRIVS视图显示授予其他角色的角色，仅提供用户有权访问的角色的信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-275 ROLE_ROLE_PRIVS 字段

名称	类型	描述
role	character varying(128)	角色名称。
granted_role	character varying(128)	被授予的角色。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
admin_option	character varying(3)	该授权是否包含ADMIN选项。 ● YES：包含ADMIN选项。 ● NO：不包含ADMIN选项。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.8.28 ROLE_SYS_PRIVS

ROLE_SYS_PRIVS视图显示授予角色的系统特权信息，仅提供用户有权访问的角色的信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-276 ROLE_SYS_PRIVS 字段

名称	类型	描述
role	character varying(128)	角色名称。
privilege	character varying(40)	用户拥有的系统权限或ANY权限。 ● 系统权限包括rolsuper、rolinherit、rolcreaterole、rolcreatedb、rolcatupdate、rolcanlogin、rolreplication、rolauditadmin、rolsystemadmin、roluseft、rolmonitoradmin、roloperatoradmin、rolpolicyadmin。 ● ANY权限的取值请参考**表 1-402**。
admin_option	character varying(3)	该授权是否包含ADMIN选项。 ● YES：包含ADMIN选项。 ● NO：不包含ADMIN选项。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.8.29 ROLE_TAB_PRIVS

ROLE_TAB_PRIVS视图显示授予角色的对象权限信息，仅提供用户有权访问的角色的信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-277 ROLE_TAB_PRIVS 字段

名称	类型	描述
role	character varying(128)	角色名称。
owner	character varying(128)	对象的所有者。
table_name	character varying(128)	对象的名称。对象类型包括表、包、索引、序列、等等。
column_name	character varying(128)	暂不支持，值为NULL。
privilege	character varying(40)	对象上的权限，包括USAGE、UPDATE、DELETE、INSERT、CONNECT、SELECT、EXECUTE。
grantable	character varying(3)	该授权是否包含GRANT选项。 ● YES：包含GRANT选项。 ● NO：不包含GRANT选项。
common	character varying(3)	暂不支持，值为NULL。
inherited	character varying(3)	暂不支持，值为NULL。

7.3.9 动态脱敏

7.3.9.1 GS_MASKING

GS_MASKING视图显示所有已配置的动态脱敏策略信息。需要有系统管理员或安全策略管理员权限才可以访问此视图。

表 7-278 GS_MASKING 字段

名称	类型	描述
polname	name	脱敏策略名称。
polenabled	boolean	脱敏策略开关。
maskaction	name	脱敏函数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
labelname	name	脱敏函数作用的标签名称。
masking_object	text	脱敏数据库资源对象。
filter_name	text	过滤条件的逻辑表达式。

7.3.10 透明加密

7.3.10.1 PG_TDE_INFO

PG_TDE_INFO视图显示整个集群的加密信息。

表 7-279 PG_TDE_INFO 字段

名称	类型	描述
is_encrypt	boolean	是否加密集群。 ● f：非加密集群。 ● t：加密集群。
g_tde_algo	text	加密算法。 ● SM4-CTR-128 ● AES-CTR-128
remain	text	保留字段。

7.3.11 DATABASE LINK

7.3.11.1 GS_DB_LINKS

GS_DB_LINKS系统视图显示DATABASE LINK对象的相关信息，用户可以查看属于自己和PUBLIC级别的DATABASE LINK信息。

表 7-280 GS_DB_LINKS 字段

名称	类型	描述
dblinkid	oid	当前DATABASE LINK对象的oid。
dlname	name	当前DATABASE LINK对象的名称。
dlowner	oid	当前DATABASE LINK对象拥有者的id。拥有者为public时值为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dlownername	name	当前DATABASE LINK对象拥有者的名称。
options	text[]	当前DATABASE LINK对象的连接信息，使用“keyword=value”格式的字符串。
useroptions	text	当前DATABASE LINK对象连接的远程用户信息。
heterogeneous	text	暂不支持，值为NULL。
protocol	text	暂不支持，值为NULL。
opencursors	text	暂不支持，值为NULL。
intransaction	boolean	当前DATABASE LINK对象是否在事务中。
updatasent	boolean	当前DATABASE LINK对象是否在事务中使用了更新数据的语句且处于未提交阶段。

7.3.11.2 V$DBLINK

V$DBLINK系统视图显示DATABASE LINK对象的相关信息，用户可以查看属于自己和PUBLIC级别的DATABASE LINK信息。默认只有管理员用户可以查看该视图。

说明

本视图数据取自于**GS_DB_LINKS**，但只记录在本次session中曾被使用过或正在使用的DATABASE LINK信息，DATABASE LINK具体的使用方式请参见《开发指南》中“特性使用指导 > DATABASE LINK”章节。

表 7-281 V$DBLINK 字段

名称	类型	描述
db_link	character varying(128)	当前DATABASE LINK对象的名称。
owner_id	numeric	当前DATABASE LINK对象拥有者的ID。DATABASE LINK对象的拥有者是PUBLIC时，owner_id为0。
logged_on	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
heterogeneous	character varying(3)	暂不支持，值为NULL。
protocol	character varying(6)	暂不支持，值为NULL。
open_cursors	numeric	暂不支持，值为NULL。
in_transaction	character varying(3)	当前DATABASE LINK对象是否在事务中。 ● YES：在事务中。 ● NO：不在事务中。
update_sent	character varying(3)	当前DATABASE LINK对象是否使用了更新数据的语句。 ● YES：使用了更新数据的语句。 ● NO：没有使用更新数据的语句。
commit_point_strength	numeric	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。

7.3.12 物化视图

7.3.12.1 GS_MATVIEWS

GS_MATVIEWS视图显示数据库中每一个物化视图的信息。

表 7-282 GS_MATVIEWS 字段

名称	类型	引用	描述
schemaname	name	PG_NAMESPACE.nspname	物化视图的模式名。
matviewname	name	PG_CLASS.relname	物化视图名。
matviewowner	name	PG_AUTHID.rolname	物化视图的所有者。
tablespace	name	PG_TABLESPACE.spcname	物化视图的表空间名（如果使用数据库默认表空间则为空）。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
hasindexes	boolean	-	如果物化视图有（或者最近有过）任何索引，则此列为真。
definition	text	-	物化视图的定义（一个重构的SELECT查询）。

7.3.13 其他系统视图

说明

访问SYS SCHEMA下的视图需要用户拥有对它的访问权限。普通用户需要授权后才可访问，即管理员用户执行GRANT USAGE ON SCHEMA SYS TO username;后才可访问。

7.3.13.1 ADM_ARGUMENTS

ADM_ARGUMENTS视图显示所有存储过程和函数的参数信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-283 ADM_ARGUMENTS 字段

名称	类型	描述
owner	character varying(128)	函数或存储过程的所有者。
object_name	character varying(128)	函数或存储过程的名称。
package_name	character varying(128)	暂不支持，值为NULL。
object_id	oid	函数或存储过程的oid。
overload	character varying(40)	表示该函数是该名称的第n个重载函数。
subprogram_id	numeric	暂不支持，值为NULL。
argument_name	character varying(128)	参数名称。
position	numeric	该参数在参数列表中的位置。函数返回值的位置默认为0。
sequence	numeric	定义参数的顺序。从1开始，返回类型在前，然后是每个参数。
data_level	numeric	复合类型参数的嵌套深度。此列的值始终为0，因为每个参数现只显示一行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
data_type	character varying(64)	参数的数据类型。
defaulted	character varying(1)	参数是否有默认值。 ● y：有默认值。 ● n：没有默认值。
default_value	text	暂不支持，值为NULL。
default_length	numeric	暂不支持，值为NULL。
in_out	character varying(9)	参数出入属性。 ● in：入参。 ● out：出参。 ● in/out：出入参。 ● variadic：VARIADIC参数。
data_length	numeric	暂不支持，值为NULL。
data_precision	numeric	暂不支持，值为NULL。
data_scale	numeric	暂不支持，值为NULL。
radix	numeric	数字的参数基数。smallint、integer、bigint、numeric、float为10，其余值为NULL。
character_set_name	character varying(44)	暂不支持，值为NULL。
type_owner	character varying(128)	数据类型所有者。
type_name	character varying(128)	参数类型名，仅显示自定义类型。
type_subname	character varying(128)	暂不支持，值为NULL。
type_link	character varying(128)	暂不支持，值为NULL。
type_object_type	character varying(7)	由type_owner、type_name和type_subname列描述的类型的类型： ● table：参数为表类型。 ● view：参数为视图类型。 ● 其余值为NULL。
pls_type	character varying(128)	对于数字类型参数，为参数的PL/SQL类型的名称，否则为空。
char_length	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
char_used	character varying(1)	暂不支持。 ● b：varchar，nvarchar2，bpchar，char类型。 ● NULL：其余类型。
origin_con_id	character varying(256)	暂不支持，值为0。

7.3.13.2 ADM_COL_COMMENTS

ADM_COL_COMMENTS视图显示数据库表中字段的注释信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-284 ADM_COL_COMMENTS 字段

名称	类型	描述
owner	character varying(128)	表的所有者。
table_name	character varying(128)	表名。
column_name	character varying(128)	列名。
comments	text	注释。
origin_con_id	numeric	暂不支持，值为0。
schema	character varying(64)	列所属的名称空间的名称。

7.3.13.3 ADM_COLL_TYPES

ADM_COLL_TYPES视图显示所有集合类型的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-285 ADM_COLL_TYPES 字段

名称	类型	描述
owner	character varying(128)	集合的所有者。
type_name	character varying(128)	集合的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
coll_type	character varying(128)	集合的描述。
upper_bound	numeric	暂不支持，值为NULL。
elem_type_mod	character varying(7)	暂不支持，值为NULL。
elem_type_owner	character varying(128)	集合基于的元素类型的所有者。该值主要用于用户定义的类型。
elem_type_name	character varying(128)	集合所依据的数据类型或用户定义类型的名称。
length	numeric	暂不支持，值为NULL。
precision	numeric	暂不支持，值为NULL。
scale	numeric	暂不支持，值为NULL。
character_set_name	character varying(44)	暂不支持，值为NULL。
elem_storage	character varying(7)	暂不支持，值为NULL。
nulls_stored	character varying(3)	暂不支持，值为NULL。
char_used	character varying(1)	暂不支持，值为NULL。

7.3.13.4 ADM_CONS_COLUMNS

ADM_CONS_COLUMNS视图显示数据库表中约束的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-286 ADM_CONS_COLUMNS 字段

名称	类型	描述
owner	character varying(64)	约束的创建者。
constraint_name	character varying(64)	约束名。
table_name	character varying(64)	约束相关的表名。
column_name	character varying(64)	约束相关的列名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
position	smallint	表中列的位置。

7.3.13.5 ADM_CONSTRAINTS

ADM_CONSTRAINTS视图显示数据库表中约束的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-287 ADM_CONSTRAINTS 字段

名称	类型	描述
constraint_name	character varying(64)	约束名。
constraint_type	text	约束类型。 ● c：检查约束。 ● f：外键约束。 ● p：主键约束。 ● u：唯一约束。
table_name	character varying(64)	约束相关的表名。
index_owner	character varying(64)	约束相关的索引的所有者（只针对唯一约束和主键约束）。
index_name	character varying(64)	约束相关的索引名（只针对唯一约束和主键约束）。
owner	character varying(64)	约束的创建者。
status	character varying(8)	约束的状态。
generated	character varying(14)	暂不支持，值为NULL。
search_condition	text	暂不支持，值为NULL。
search_condition_vc	character varying(4000)	暂不支持，值为NULL。
r_owner	character varying(128)	暂不支持，值为NULL。
r_constraint_name	character varying(128)	暂不支持，值为NULL。
delete_rule	character varying(9)	暂不支持，值为NULL。
con_deferrable	character varying(14)	暂不支持，值为NULL。
deferred	character varying(9)	暂不支持，值为NULL。
validated	character varying(13)	暂不支持，值为NULL。
bad	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
rely	character varying(4)	暂不支持，值为NULL。
last_change	date	暂不支持，值为NULL。
invalid	character varying(7)	暂不支持，值为NULL。
view_related	character varying(14)	暂不支持，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为NULL。

7.3.13.6 ADM_DATA_FILES

ADM_DATA_FILES视图显示关于数据库文件的描述。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-288 ADM_DATA_FILES 字段

名称	类型	描述
tablespace_name	name	文件所属的表空间的名称。
bytes	double precision	文件的字节长度。

7.3.13.7 ADM_DEPENDENCIES

ADM_DEPENDENCIES视图显示数据库中的类型、表、视图、存储过程、函数、触发器之间的依赖关系。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-289 ADM_DEPENDENCIES 字段

名称	类型	描述
owner	name	对象的所有者。
name	name	对象的名称。
type	character varying(18)	对象的类型。
referenced_owner	name	被引用对象的所有者。
referenced_name	name	被引用对象的名称。
referenced_type	character varying(18)	被引用对象的类型。
referenced_link_name	character varying(128)	暂不支持，值为NULL。
dependency_type	character varying(4)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.13.8 ADM_DIRECTORIES

ADM_DIRECTORIES显示数据库中所有目录对象的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-290 ADM_DIRECTORIES 字段

名称	类型	描述
owner	oid	目录的所有者。
directory_name	name	目录名称。
directory_path	text	目录的操作系统路径名。
origin_con_id	character varying(256)	暂不支持，值为NULL。

7.3.13.9 ADM_HIST_SNAPSHOT

ADM_HIST_SNAPSHOT视图记录当前系统中存储的WDR快照数据信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图位于PG_CATALOG和SYS Schema下。该视图只有在GUC参数enable_wdr_snapshot为on时才可以访问。访问PG_CATALOG.ADM_HIST_SNAPSHOT和SYS.ADM_HIST_SNAPSHOT视图除需要本视图访问权限外，还需要snapshot schema、snapshot表和tables_snap_timestamp表的访问权限。

表 7-291 ADM_HIST_SNAPSHOT 字段

名称	类型	描述
snap_id	bigint	唯一快照ID。
dbid	oid	快照的数据库ID。
instance_number	oid	暂不支持，取值同DBID。
startup_time	timestamp(3) without time zone	实例启动时间。
begin_interval_time	timestamp without time zone	快照间隔开始的时间（即上次快照的结束时间）。
end_interval_time	timestamp without time zone	快照间隔结束的时间。拍摄快照的实际时间（即本次快照的结束时间）。
flush_elapsed	interval	执行快照所花费的时间。
snap_level	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
error_count	numeric	暂不支持，值为NULL。
snap_flag	numeric	暂不支持，值为NULL。
snap_timezone	interval day to second(0)	快照时区表示为与UTC（协调世界时）时区的偏移量。
begin_interval_time_tz	timestamp with time zone	快照间隔开始的时间（即上次快照的结束时间），带时区。
end_interval_time_tz	timestamp with time zone	快照间隔结束的时间。拍摄快照的实际时间（即本次快照的结束时间），带时区。
con_id	numeric	暂不支持，值为0。

7.3.13.10 ADM_HIST_SQL_PLAN

ADM_HIST_SQL_PLAN视图描述当前用户通过执行EXPLAIN PLAN收集到的计划信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在PG_CATALOG和SYS Schema下。

表 7-292 ADM_HIST_SQL_PLAN 字段

名称	类型	描述
dbid	text	数据库ID。
sql_id	character varying(30)	表示插入该条数据的会话，由服务线程启动时间戳和服务线程ID组成。受非空约束限制。
plan_hash_value	bigint	查询标识。
id	integer	分配给执行计划中的每个步骤编号。
operation	character varying(30)	操作描述。
options	character varying(255)	操作选项。
object_node	character varying(128)	暂不支持，值为NULL。
object#	numeric	暂不支持，值为NULL。
object_owner	name	表或索引的对象编号。
object_name	name	操作对应的对象名，来自于用户定义。
object_alias	character varying(261)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
object_type	character varying(30)	对象类型。
optimizer	character varying(20)	暂不支持，值为NULL。
parent_id	numeric	暂不支持，值为NULL。
depth	numeric	暂不支持，值为NULL。
position	numeric	暂不支持，值为NULL。
search_columns	numeric	暂不支持，值为NULL。
cost	double precision	优化器对算子估算的执行代价。
cardinality	double precision	优化器对算子估算访问表记录基数大小。
bytes	numeric	暂不支持，值为NULL。
other_tag	character varying(35)	暂不支持，值为NULL。
partition_start	character varying(64)	暂不支持，值为NULL。
partition_stop	character varying(64)	暂不支持，值为NULL。
partition_id	numeric	暂不支持，值为NULL。
other	character varying(4000)	暂不支持，值为NULL。
distribution	character varying(20)	暂不支持，值为NULL。
cpu_cost	numeric	暂不支持，值为NULL。
io_cost	numeric	暂不支持，值为NULL。
temp_space	numeric	暂不支持，值为NULL。
access_predicates	character varying(4000)	暂不支持，值为NULL。
filter_predicates	character varying(4000)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
projection	character varying(4000)	操作输出的列信息。
time	numeric	暂不支持，值为NULL。
qblock_name	character varying(128)	暂不支持，值为NULL。
remarks	character varying(4000)	暂不支持，值为NULL。
timestamp	date	暂不支持，值为NULL。
other_xml	clob	暂不支持，值为NULL。
con_dbid	text	容器数据库ID，目前与dbid取值相同。
con_id	numeric	容器ID，目前不支持容器，值为0。

7.3.13.11 ADM_HIST_SQLSTAT

ADM_HIST_SQLSTAT视图描述当前节点的执行语句的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在PG_CATALOG和SYS Schema下。

WDR Snapshot启动（即GUC参数enable_wdr_snapshot为on时）后，用户可以查看此视图中的数据。

表 7-293 ADM_HIST_SQLSTAT 字段

名称	类型	描述
instance_number	integer	快照的实例编号。
plan_hash_value	integer	归一化SQL ID。
module	integer	包含第一次解析SQL语句时正在执行的模块的名称。
apwait_delta	integer	应用程序等待时间的Delta值。
sql_id	bigint	查询标识。
snap_id	bigint	唯一快照ID。
elapsed_time_delta	bigint	有效的DB时间花费，多线程将累加（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cpu_time_delta	bigint	CPU的时间花费（单位：微秒）。
executions_delta	bigint	自从它被带入库缓存以来在此对象上发生的执行次数增量。
iowait_delta	bigint	IO的时间花费（单位：微秒）。
rows_processed_delta	bigint	SELECT返回的结果集行数。
parsing_schema_name	character varying	暂不支持，值为NULL。
disk_reads_delta	bigint	暂不支持，值为NULL。
buffer_reads_delta	bigint	暂不支持，值为NULL。
clwait_delta	bigint	暂不支持，值为NULL。

7.3.13.12 ADM_HIST_SQLTEXT

ADM_HIST_SQLTEXT视图描述当前节点的执行语句的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在PG_CATALOG和SYS Schema下。

WDR Snapshot启动（即GUC参数enable_wdr_snapshot为on时）后，用户可以查看此视图中的数据。

表 7-294 ADM_HIST_SQLTEXT 字段

名称	类型	描述
dbid	integer	数据库ID。
sql_id	bigint	查询标识。
sql_text	clob	查询对应文本。
command_type	integer	暂不支持，值为0。
con_dbid	integer	容器数据库ID，目前与dbid取值相同。
con_id	integer	容器ID，目前不支持容器，值为0。

版权所有 © 华为云计算技术有限公司

7.3.13.13 ADM_IND_COLUMNS

ADM_IND_COLUMNS视图显示数据库中索引字段的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-295 ADM_IND_COLUMNS 字段

名称	类型	描述
index_owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名。
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
column_name	name	列名。
column_position	smallint	索引中列的位置。
column_length	numeric	列的长度。如果列是变长类型，该字段取值为NULL。
char_length	numeric	列的最大字节长度。
descend	character varying(4)	列是降序（DESC）排序还是升序（ASC）排序。
collated_column_id	numeric	暂不支持，值为NULL。

7.3.13.14 ADM_IND_EXPRESSIONS

ADM_IND_EXPRESSIONS视图显示数据库中表达式索引的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-296 ADM_IND_EXPRESSIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
index_owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名。
column_expression	text	定义列的基于函数的索引表达式。
column_position	smallint	索引中列的位置。

版权所有 © 华为云计算技术有限公司

7.3.13.15 ADM_INDEXES

ADM_INDEXES视图显示数据库中所有索引的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-297 ADM_INDEXES 字段

名称	类型	描述
owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名称。
table_name	character varying(64)	索引对应的表名。
uniqueness	text	该索引是否为唯一索引。 ● UNIQUE：唯一索引。 ● NONUNIQUE：非唯一索引。
partitioned	character(3)	该索引是否具有分区表的性质。 ● Yes：索引具有分区表的性质。 ● No：索引不具有分区表的性质。
generated	character varying(1)	该索引的名称是否为系统生成。 ● y：索引名称为系统生成。 ● n：索引名称非系统生成。
index_type	character varying(27)	索引类型。 ● NORMAL：索引属性都是简单的引用，表达式树为空。 ● FUNCTION-BASED NORMAL：存在表达式树用于非简单字段引用的索引属性。
table_owner	character varying(128)	索引对象的所有者。
table_type	character(11)	索引对象的类型。 ● TABLE：索引对象为表类型。
tablespace_name	character varying(30)	包含索引的表空间名称。
status	character varying(8)	非分区索引状态。 ● VALID：非分区索引可以用于查询。 ● UNUSABLE：非分区索引不可用。 ● N/A：索引具有分区表性质。
compression	character varying(13)	暂不支持，值为NULL。
prefix_length	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
ini_trans	numeric	暂不支持，值为NULL。
max_trans	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
pct_threshold	numeric	暂不支持，值为NULL。
include_column	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
pct_free	numeric	暂不支持，值为NULL。
logging	character varying(3)	暂不支持，值为NULL。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	暂不支持，值为NULL。
num_rows	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	date	暂不支持，值为NULL。
degree	character varying(40)	暂不支持，值为NULL。
instances	character varying(40)	暂不支持，值为NULL。
temporary	character varying(1)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
buffer_pool	character varying(7)	暂不支持，值为NULL。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
user_stats	character varying(3)	暂不支持，值为NULL。
duration	character varying(15)	暂不支持，值为NULL。
pct_direct_access	numeric	暂不支持，值为NULL。
ityp_owner	character varying(128)	暂不支持，值为NULL。
ityp_name	character varying(128)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
domidx_status	character varying(12)	暂不支持，值为NULL。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
funcidx_status	character varying(8)	暂不支持，值为NULL。
join_index	character varying(3)	暂不支持，值为NULL。
iot_redundant_pkey_elim	character varying(3)	暂不支持，值为NULL。
dropped	character varying(3)	暂不支持，值为NULL。
visibility	character varying(9)	该索引对优化器是否可见。 ● VISIBLE：索引对优化器可见。 ● INVISIBLE：索引对优化器不可见。
domidx_management	character varying(14)	暂不支持，值为NULL。
segment_created	character varying(3)	暂不支持，值为NULL。
orphaned_entries	character varying(3)	暂不支持，值为NULL。
indexing	character varying(7)	暂不支持，值为NULL。
auto	character varying(3)	暂不支持，值为NULL。

7.3.13.16 ADM_OBJECTS

ADM_OBJECTS视图显示数据库中所有数据库对象的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-298 ADM_OBJECTS 字段

名称	类型	描述
owner	name	对象的所有者。
object_name	name	对象的名称。
object_id	oid	对象的OID。
object_type	name	对象的类型。例如table，schema，index等。
namespace	oid	对象所在的命名空间。
temporary	character(1)	对象是否为临时对象。
status	character varying(7)	对象的状态。 ● valid：有效。 ● invalid：已失效。
subobject_name	name	对象的子对象名称。
generated	character(1)	对象名称是否是系统生成。
created	timestamp with time zone	对象的创建时间。
last_ddl_time	timestamp with time zone	对象的最后修改时间。
default_collation	character varying(100)	对象的默认排序规则。
data_object_id	numeric	暂不支持，值为NULL。
timestamp	character varying(19)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
edition_name	character varying(128)	暂不支持，值为NULL。
sharing	character varying(18)	暂不支持，值为NULL。
editionable	character varying(1)	暂不支持，值为NULL。
oracle_maintained	character varying(1)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
application	character varying(1)	暂不支持，值为NULL。
duplicated	character varying(1)	暂不支持，值为NULL。
sharded	character varying(1)	暂不支持，值为NULL。
created_appid	numeric	暂不支持，值为NULL。
modified_appid	numeric	暂不支持，值为NULL。
created_vsnid	numeric	暂不支持，值为NULL。
modified_vsnid	numeric	暂不支持，值为NULL。

须知

created和last_ddl_time支持的范围请参见**PG_OBJECT**中的记录范围。

7.3.13.17 ADM_PROCEDURES

ADM_PROCEDURES视图显示数据库中所有存储过程、函数和触发器的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-299 ADM_PROCEDURES 字段

名称	类型	描述
owner	character varying(64)	存储过程、函数、触发器或包的所有者。
object_name	character varying(64)	存储过程、函数或触发器的名称。若为包中函数或存储过程，则为包名。
procedure_name	character varying(128)	若object_name为包名，则为包中函数或存储过程名称，否则为空。
object_id	oid	存储过程、函数、触发器或包的OID。
subprogram_id	numeric	包中函数或存储过程的位置。
overload	character varying(40)	该函数是该名称的第n个重载函数。
object_type	character varying(13)	对象的类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
aggregate	character varying(3)	是否为聚合函数。 ● yes：是。 ● no：不是。
pipelined	character varying(3)	暂不支持，值为no。
impltypeowner	character varying(128)	实现类型的所有者。
impltypename	character varying(128)	实现类型的名称。
parallel	character varying(3)	暂不支持，值为no。
interface	character varying(3)	暂不支持，值为no。
deterministic	character varying(3)	暂不支持，值为no。
authid	character varying(12)	使用的权限类型。 ● definer：使用创建者权限。 ● current_user：使用调用者权限。 因该字段与保留关键字冲突，调用该字段需加视图名。
result_cache	character varying(3)	暂不支持，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为0。
polymorphic	character varying(5)	暂不支持，值为NULL。
argument_number	smallint	存储过程的入参个数。

7.3.13.18 ADM_SCHEDULER_JOB_ARGS

ADM_SCHEDULER_JOB_ARGS视图显示数据库中所有任务的有关参数信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-300 ADM_SCHEDULER_JOB_ARGS 字段

名称	类型	描述
owner	character varying(128)	参数所属作业的拥有者。
job_name	character varying(128)	参数所属作业名。
argument_name	character varying(128)	参数名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
argument_position	numeric	参数在参数列表中的位置。
argument_type	character varying(257)	参数的数据类型，可以是用户的自定义数据类型。
value	character varying(4000)	参数值。
anydata_value	character varying(4000)	暂不支持，值为NULL。
out_argument	character varying(5)	保留字段，值为NULL。

7.3.13.19 ADM_SCHEDULER_JOBS

ADM_SCHEDULER_JOBS视图显示数据库中所有DBE_SCHEDULER定时任务的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-301 ADM_SCHEDULER_JOBS 字段

名称	类型	描述
owner	name	定时任务的所有者。
job_name	text	定时任务名。
job_subname	character varying(128)	暂不支持，值为NULL。
job_style	text	定时任务的行为模式。创建时指定，仅支持指定为“REGULAR”，不指定时为NULL。
job_creator	name	定时任务的创建者。
client_id	character varying(65)	暂不支持，值为NULL。
global_uid	character varying(33)	暂不支持，值为NULL。
program_owner	character varying(4000)	定时任务引用的程序的所有者。
program_name	text	定时任务引用的程序的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
job_type	character varying(16)	定时任务内联程序类型，可用类型为： ● PLSQL_BLOCK：匿名存储过程快。 ● STORED_PROCEDURE：保存的存储过程。 ● EXTERNAL_SCRIPT：外部脚本。
job_action	text	定时任务的程序内容。
number_of_arguments	text	定时任务的参数个数。范围为0 ~ 255。
schedule_owner	character varying(4000)	暂不支持，值为NULL。
schedule_name	text	定时任务引用的调度的名称。
schedule_type	character varying(12)	暂不支持，值为NULL。
start_date	timestamp without time zone	定时任务的起始时间。
repeat_interval	text	定时任务的任务周期。
event_queue_owner	character varying(128)	暂不支持，值为NULL。
event_queue_name	character varying(128)	暂不支持，值为NULL。
event_queue_agent	character varying(523)	暂不支持，值为NULL。
event_condition	character varying(4000)	暂不支持，值为NULL。
event_rule	character varying(261)	暂不支持，值为NULL。
file_watcher_owner	character varying(261)	暂不支持，值为NULL。
file_watcher_name	character varying(261)	暂不支持，值为NULL。
end_date	timestamp without time zone	定时任务的失效时间。
job_class	text	定时任务所属的定时任务类的名称。
enabled	boolean	定时任务的启用状态。
auto_drop	text	定时任务的自动删除功能状态。
restart_on_recovery	character varying(5)	暂不支持，值为NULL。
restart_on_failure	character varying(5)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
state	"char"	定时任务的状态。
job_priority	numeric	暂不支持，值为NULL。
run_count	numeric	暂不支持，值为NULL。
uptime_run_count	numeric	暂不支持，值为NULL。
max_runs	numeric	暂不支持，值为NULL。
failure_count	smallint	定时任务失败次数统计。
uptime_failure_count	numeric	暂不支持，值为NULL。
max_failures	numeric	定时任务标记为破坏之前允许失败的最大次数。
retry_count	numeric	暂不支持，值为NULL。
last_start_date	timestamp without time zone	定时任务上次拉起时间。
last_run_duration	interval day to second(6)	定时任务上次执行的时长。
next_run_date	timestamp without time zone	定时任务下次执行时间。
schedule_limit	interval day to second(0)	暂不支持，值为NULL。
max_run_duration	interval day to second(0)	暂不支持，值为NULL。
logging_level	character varying(11)	暂不支持，值为NULL。
store_output	character varying(5)	是否存储所有定时任务的输出信息。
stop_on_window_close	character varying(5)	暂不支持，值为NULL。
instance_stickiness	character varying(5)	暂不支持，值为NULL。
raise_events	character varying(4000)	暂不支持，值为NULL。
system	character varying(5)	暂不支持，值为NULL。
job_weight	numeric	暂不支持，值为NULL。
nls_env	character varying(4000)	暂不支持，值为NULL。
source	character varying(128)	暂不支持，值为NULL。
number_of_destinations	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
destination_owner	character varying(261)	暂不支持，值为NULL。
destination	text	定时任务目标名称。 如果destination值为“CCN”则说明该任务是CCN的JOB，相关说明请参见DBE_SCHEDULER。
credential_owner	character varying(128)	暂不支持，值为NULL。
credential_name	text	定时任务的证书名称。
instance_id	oid	当前数据库的oid。
deferred_drop	character varying(5)	暂不支持，值为NULL。
allow_runs_in_restricted_mode	character varying(5)	暂不支持，值为NULL。
comments	text	定时任务的备注。
flags	numeric	暂不支持，值为NULL。
restartable	character varying(5)	暂不支持，值为NULL。
has_constraints	character varying(5)	暂不支持，值为NULL。
connect_credential_owner	character varying(128)	暂不支持，值为NULL。
connect_credential_name	character varying(128)	暂不支持，值为NULL。
fail_on_script_error	character varying(5)	暂不支持，值为NULL。

7.3.13.20 ADM_SCHEDULER_PROGRAM_ARGS

ADM_SCHEDULER_PROGRAM_ARG视图显示数据库中所有程序的有关参数信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-302 ADM_SCHEDULER_PROGRAM_ARGS 字段

名称	类型	描述
owner	character varying(128)	参数所属程序的拥有者。
program_name	character varying(128)	参数所属程序的名称。
argument_name	character varying(128)	参数名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
argument_position	numeric	参数在参数列表中的位置。
argument_type	character varying(257)	参数的数据类型，可以是用户的自定义数据类型。
metadata_attribute	character varying(19)	暂不支持，值为NULL。
default_value	character varying(4000)	参数默认值。
default_anydata_value	character varying(4000)	暂不支持，值为NULL。
out_argument	character varying(5)	保留字段，值为NULL。

7.3.13.21 ADM_SCHEDULER_PROGRAMS

ADM_SCHEDULER_PROGRAMS视图显示数据库中所有可以调度的程序信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-303 ADM_SCHEDULER_PROGRAMS 字段

名称	类型	描述
owner	name	调度程序的所有者。
program_name	text	调度程序的名称。
program_type	character varying(16)	调度程序的类型。可用类型为： ● PLSQL_BLOCK：匿名存储过程快。 ● STORED_PROCEDURE：保存的存储过程。 ● EXTERNAL_SCRIPT：外部脚本。
program_action	text	调度程序执行的操作。
number_of_arguments	numeric	调度程序参数个数。范围为0 ~ 255。
enabled	character varying(5)	调度程序是否启用。
comments	text	调度程序的备注。
detached	character varying(5)	暂不支持，值为NULL。
schedule_limit	interval day to second(0)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
priority	numeric	暂不支持，值为NULL。
weight	numeric	暂不支持，值为NULL。
max_runs	numeric	暂不支持，值为NULL。
max_failures	numeric	暂不支持，值为NULL。
max_run_duration	interval day to second(0)	暂不支持，值为NULL。
has_constraints	character varying(5)	暂不支持，值为NULL。
nls_env	character varying(4000)	暂不支持，值为NULL。

7.3.13.22 ADM_SCHEDULER_RUNNING_JOBS

ADM_SCHEDULER_RUNNING_JOBS视图显示数据库中所有正在执行的DBE_SCHEDULER定时任务的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-304 ADM_SCHEDULER_RUNNING_JOBS 字段

名称	类型	描述
owner	character varying(128)	定时任务所有者。
job_name	character varying(128)	定时任务的名称。
job_subname	character varying(128)	暂不支持，值为NULL。
job_style	character varying(17)	定时任务行为模式。创建时指定，仅支持指定为“REGULAR”，不指定时为NULL。
detached	character varying(5)	暂不支持，值为NULL。
session_id	numeric	执行该定时任务的会话ID。
slave_process_id	numeric	暂不支持，值为NULL。
slave_os_process_id	character varying(12)	执行该任务的进程号。
running_instance	numeric	暂不支持，值为NULL。
resource_consumer_group	character varying(32)	暂不支持，值为NULL。
elapsed_time	interval day to second(2)	定时任务本次已执行的时长。
cpu_used	interval day to second(2)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
destination_owner	character varying(261)	暂不支持，值为NULL。
destination	character varying(261)	定时任务目标名称。 如果destination值为“CCN”则说明该任务是CCN的JOB，相关说明详见**DBE_SCHEDULER**。
credential_owner	character varying(128)	暂不支持，值为NULL。
credential_name	character varying(128)	定时任务证书名称。
log_id	numeric	暂不支持，值为NULL。

7.3.13.23 ADM_SEGMENTS

ADM_SEGMENTS视图显示数据库中所有段分配的存储空间。同时存在于PG_CATALOG和SYS Schema下。仅系统管理员可访问。信息无法通过系统表获取。该视图为空视图。

表 7-305 ADM_SEGMENTS 字段

名称	类型	描述
owner	character varying(128)	暂不支持，值为NULL。
segment_name	character varying(128)	暂不支持，值为NULL。
partition_name	character varying(128)	暂不支持，值为NULL。
segment_type	character varying(18)	暂不支持，值为NULL。
segment_subtype	character varying(10)	暂不支持，值为NULL。
tablespace_name	character varying(30)	暂不支持，值为NULL。
header_file	numeric	暂不支持，值为NULL。
header_block	numeric	暂不支持，值为NULL。
bytes	numeric	暂不支持，值为NULL。
blocks	numeric	暂不支持，值为NULL。
extents	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_size	numeric	暂不支持，值为NULL。
retention	character varying(7)	暂不支持，值为NULL。
minretention	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
relative_fno	numeric	暂不支持，值为NULL。
buffer_pool	character varying(7)	暂不支持，值为NULL。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
inmemory	character varying(8)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。

7.3.13.24 ADM_SEQUENCES

ADM_SEQUENCES视图显示数据库中所有序列的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-306 ADM_SEQUENCES 字段

名称	类型	描述
sequence_owner	character varying(64)	序列的所有者。
sequence_name	character varying(64)	序列名称。
min_value	int16	序列的最小值。
max_value	int16	序列的最大值。
increment_by	int16	序列的递增值。
last_number	int16	该序列从GTM中一次性批量申请的最后一个值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cache_size	int16	序列磁盘缓存大小。
cycle_flag	character(1)	序列是否是循环序列，取值为y或n： ● y：是循环序列。 ● n：不是循环序列。
order_flag	character(1)	序列是否是有序序列。取值范围： ● y：有序序列。 ● n：不是有序序列。

7.3.13.25 ADM_SOURCE

ADM_SOURCE视图显示数据库中所有存储过程、函数、触发器的定义信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-307 ADM_SOURCE 字段

名称	类型	描述
owner	name	对象的所有者。
name	name	对象的名字。
type	name	对象的类型。取值范围：function、procedure、trigger。
line	numeric	此行在定义信息中的行号。
text	text	存储对象的文本来源。
origin_con_id	character varying(256)	暂不支持，值为0。

7.3.13.26 ADM_SYNONYMS

ADM_SYNONYMS视图显示数据库中所有同义词的信息。默认只有系统管理员权限才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-308 ADM_SYNONYMS 字段

名称	类型	描述
owner	text	同义词的所有者。PUBLIC同义词的所有者为PUBLIC。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schema_name	text	同义词所属模式名。PUBLIC同义词的所属模式名NULL。
synonym_name	text	同义词的名称。
table_owner	text	关联对象的所有者。尽管该列称为table_owner，但它拥有的该关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
table_name	text	关联对象名。尽管该列称为table_name，但此关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
table_schema_name	text	关联对象所属的模式名。尽管该列称为table_schema_name，但此Schema下的该关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。

7.3.13.27 ADM_TAB_COL_STATISTICS

ADM_TAB_COL_STATISTICS视图显示从ADM_TAB_COLUMNS中提取的列统计信息和直方图信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-309 ADM_TAB_COL_STATISTICS 字段

名称	类型	描述
owner	character varying(128)	表的所有者。
table_name	character varying(128)	表名。
column_name	character varying(128)	列名。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的低值。
high_value	raw	列中的高值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
density	numeric	● 如果COLUMN_NAME上有直方图，则此列将显示直方图中跨越少于2个端点的值的选择性。它不代表跨越2个或更多端点的值的选择性。 ● 如果COLUMN_NAME上没有可用的直方图，则该列的值为1/NUM_DISTINCT。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
sample_size	numeric	用于分析此列的样本量。
last_analyzed	date	最近分析此列的日期。数据库重启后，数据会丢失。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
notes	character varying(99)	暂不支持，值为NULL。
avg_col_len	numeric	列的平均长度（以字节为单位）。
histogram	character varying(15)	直方图是否存在以及存在的类型： ● NONE：不存在直方图。 ● FREQUENCY：频率直方图。 ● EQUI-WIDTH：等宽直方图。
scope	character varying(7)	对于在除全局临时表之外的任何表上收集的统计信息，该值是SHARED（表示统计信息在所有会话之间共享）。
schema	character varying(64)	列所属的名称空间的名称。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。

7.3.13.28 ADM_TAB_COLS

ADM_TAB_COLS视图显示表和视图列的相关信息。数据库中每个表和视图的每一个字段在ADM_TAB_COLS里各有一行对应的数据。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。该视图与ADM_TAB_COLUMNS视图显示行数相同，仅存在字段差异。

版权所有 © 华为云计算技术有限公司

表 7-310 ADM_TAB_COLS 字段

名称	类型	描述
owner	character varying(64)	表或视图的所有者。
table_name	character varying(128)	表或视图的名称。
column_name	character varying(128)	列名。
data_type	character varying(128)	列的数据类型，可以是用户自定义的数据类型。
data_type_mod	character varying(3)	暂不支持，值为NULL。
data_type_owner	character varying(128)	列的数据类型的所有者。
data_length	numeric	列的字节长度。
data_precision	numeric	数据类型的精度。对于numeric数据类型有效，其他类型为NULL。
data_scale	numeric	小数点右边的位数。对于numeric数据类型有效，其他类型为0。
nullable	character varying(1)	该列是否允许为空。对于主键约束和非空约束，该值为n。
column_id	numeric	创建表时列的序号。
default_length	numeric	列的默认值字节长度。
data_default	text	列的默认值。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的最小值。
high_value	raw	列中的最大值。
density	numeric	列密度。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
last_analyzed	date	上次分析的日期。
sample_size	numeric	用于分析此列的样本量。
character_set_name	character varying(44)	暂不支持，值为NULL。
char_col_decl_length	numeric	字符类型列的声明长度。
global_stats	character varying(3)	暂不支持，值为NO。
user_stats	character varying(3)	暂不支持，值为NO。
avg_col_len	numeric	列的平均长度（单位字节）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
char_length	numeric	列的长度（以字符计）。只对varchar，nvarchar2，bpchar，char类型有效。
char_used	character varying(1)	暂不支持。 ● B：varchar、bpchar与char类型。 ● C：nvarchar2类型。 ● NULL：其余类型。
v80_fmt_image	character varying(3)	暂不支持，值为NULL。
data_upgraded	character varying(3)	暂不支持，值为YES。
hidden_column	character varying(3)	暂不支持，值为NULL。
virtual_column	character varying	列是否为虚拟列（即生成列）： ● YES：是。 ● NO：不是。
segment_column_id	numeric	暂不支持，值为NULL。
internal_column_id	numeric	列的内部序列号，内容同COLUMN_ID。
histogram	character varying(15)	直方图是否存在以及存在的类型： ● NONE：不存在直方图。 ● FREQUENCY：频率直方图。 ● EQUI_WIDTH：等宽直方图。
qualified_col_name	character varying(64)	限定列名，同COLUMN_NAME。
user_generated	character varying(3)	暂不支持，值为YES。
default_on_null	character varying(3)	暂不支持，值为NULL。
identity_column	character varying(3)	列是否为标识列： ● yes：表示为标识列。 ● no：表示为非标识列。
sensitive_column	character varying(3)	暂不支持，值为NULL。
evaluation_edition	character varying(128)	暂不支持，值为NULL。
unusable_before	character varying(128)	暂不支持，值为NULL。
unusable_beginning	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
collation	character varying(100)	列的排序规则。因该字段与保留关键字冲突，调用该字段需加视图名。
collated_column_id	numeric	暂不支持，值为NULL。
schema	character varying(64)	列所属的名称空间的名称。

说明

- 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。
- 目前CREATE TABLE语法不支持创建具有标识列字段的表，当进行查询ADM_TAB_COLS视图操作时，返回结果中identity_column字段值恒为no。

7.3.13.29 ADM_TAB_COLUMNS

ADM_TAB_COLUMNS视图显示关于表和视图的字段的信息。数据库中每个表和视图的每一个字段在ADM_TAB_COLUMNS里各有一行对应的数据。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-311 ADM_TAB_COLUMNS 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表的名称。
column_name	character varying(64)	列名。
data_type	character varying(128)	列的数据类型。
data_type_mod	character varying(3)	暂不支持，值为NULL。
data_type_owner	character varying(128)	列的数据类型的所有者。
data_length	integer	列的字节长度。
data_precision	integer	数据类型的精度。对于numeric数据类型有效，其他类型为NULL。
data_scale	integer	小数点右边的位数。对于numeric数据类型有效，其他类型为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
nullable	bpchar	该列是否允许为空。取值范围： ● y：允许。 ● n：不允许。对于主键约束和非空约束，该值为n。
column_id	integer	创建表时列的序号。
default_length	numeric	列的默认值字节长度，无默认值时为空。
data_default	text	列的默认值。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的最小值。
high_value	raw	列中的最大值。
density	numeric	列密度。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
last_analyzed	date	上次分析的日期。
sample_size	numeric	用于分析此列的样本量。
character_set_name	character varying(44)	暂不支持，值为NULL。
char_col_decl_length	numeric	字符类型列的声明长度。
global_stats	character varying(3)	暂不支持，值为no。
user_stats	character varying(3)	暂不支持，值为no。
avg_col_len	numeric	列的平均长度（单位字节）。
char_length	numeric	列的长度（以字符计）。只对varchar，nvarchar2，bpchar，char类型有效。
char_used	character varying(1)	暂不支持。 ● b：varchar、bpchar与char类型。 ● c：nvarchar2类型。 ● NULL：其余类型。
v80_fmt_image	character varying(3)	暂不支持，值为NULL。
data_upgraded	character varying(3)	暂不支持，值为yes。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
histogram	character varying(15)	直方图是否存在以及存在的类型： ● none：不存在直方图。 ● frequency：频率直方图。 ● equi-width：等宽直方图。
default_on_null	character varying(3)	暂不支持，值为NULL。
identity_column	character varying(3)	列是否为标识列： ● yes：表示为标识列。 ● no：表示为非标识列。
sensitive_column	character varying(3)	暂不支持，值为NULL。
evaluation_edition	character varying(128)	暂不支持，值为NULL。
unusable_before	character varying(128)	暂不支持，值为NULL。
unusable_beginning	character varying(128)	暂不支持，值为NULL。
collation	character varying(100)	列的排序规则。因该字段与保留关键字冲突，调用该字段需加视图名。
comments	text	列的注释。
schema	character varying(64)	列所属的名称空间的名称。

说明

- 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。
- 目前CREATE TABLE语法不支持创建具有标识列字段的表，当进行查询ADM_TAB_COLUMNS视图操作时，返回结果中identity_column字段值恒为no。

7.3.13.30 ADM_TAB_COMMENTS

ADM_TAB_COMMENTS视图显示数据库中所有表和视图的注释信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-312 ADM_TAB_COMMENTS 字段

名称	类型	描述
owner	character varying(64)	表或视图的所有者。
table_name	character varying(64)	表或视图的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
comments	text	注释。
schema	character varying(64)	表所属的名称空间的名称。

7.3.13.31 ADM_TAB_HISTOGRAMS

ADM_TAB_HISTOGRAMS系统视图显示数据库所有表和视图的直方图信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-313 ADM_TAB_HISTOGRAMS 字段

名称	类型	描述
owner	character varying(128)	表的所有者。
table_name	character varying(128)	表名。
column_name	character varying(4000)	列名。
endpoint_number	numeric	直方图的桶号。
endpoint_value	numeric	暂不支持，值为NULL。
endpoint_actual_value	character varying(4000)	桶端点的实际值。
endpoint_actual_value_raw	raw	暂不支持，值为NULL。
endpoint_repeat_count	numeric	暂不支持，值为NULL。
scope	character varying(7)	暂不支持，值为shared。

7.3.13.32 ADM_TAB_STATISTICS

ADM_TAB_STATISTICS显示数据库中所有表的优化程序统计信息。该视图同时存在于PG_CATALOG和SYS Schema下。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-314 ADM_TAB_STATISTICS 字段

名称	类型	描述
owner	character varying(128)	对象的所有者。
table_name	character varying(128)	表名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
partition_name	character varying(128)	暂不支持，值为NULL。
partition_position	numeric	暂不支持，值为NULL。
subpartition_name	character varying(128)	暂不支持，值为NULL。
subpartition_position	numeric	暂不支持，值为NULL。
object_type	character varying(12)	对象类型： ● TABLE ● PARTITION ● SUBPARTITION
num_rows	numeric	对象中的行数。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	integer	平均行长，包括行开销。
avg_space_freelist_blocks	numeric	暂不支持，值为NULL。
num_freelist_blocks	numeric	暂不支持，值为NULL。
avg_cached_blocks	numeric	暂不支持，值为NULL。
avg_cache_hit_ratio	numeric	暂不支持，值为NULL。
im_imcu_count	numeric	暂不支持，值为NULL。
im_block_count	numeric	暂不支持，值为NULL。
im_stat_update_time	timestamp(6) without time zone	暂不支持，值为NULL。
scan_rate	numeric	暂不支持，值为NULL。
sample_size	numeric	分析表格时使用的样本量。
last_analyzed	timestamp with time zone	最近分析表的日期。数据库重启后，数据会丢失。
global_stats	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
user_stats	character varying(3)	暂不支持，值为NULL。
stattype_locked	character varying(5)	暂不支持，值为NULL。
stale_stats	character varying(7)	暂不支持，值为NULL。
notes	character varying(25)	暂不支持，值为NULL。
scope	character varying(7)	暂不支持，默认值SHARED。

7.3.13.33 ADM_TAB_STATS_HISTORY

ADM_TAB_STATS_HISTORY系统视图提供数据库所有表的表统计信息历史。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-315 ADM_TAB_STATS_HISTORY 字段

名称	类型	描述
owner	character varying(128)	对象的拥有者。
table_name	character varying(128)	表名。
partition_name	character varying(128)	暂不支持，值为NULL。
subpartition_name	character varying(128)	暂不支持，值为NULL。
stats_update_time	timestamp(6) with time zone	统计信息更新的时间。数据库重启后，数据会丢失。

7.3.13.34 ADM_TABLES

ADM_TABLES视图显示关于数据库下的所有表信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-316 ADM_TABLES 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名称。
tablespace_name	character varying(64)	存储表的表空间名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dropped	character varying	当前表是否已删除。 ● yes：已删除。 ● no：未删除。
num_rows	numeric	表的估计行数。
status	character varying(8)	当前表是否有效。 ● valid：当前表有效。 ● unusable：当前表不可用。
sample_size	numeric	分析表使用的样本数量。
temporary	character(1)	是否为临时表。 ● y：是临时表。 ● n：不是临时表。
pct_free	numeric	块中空闲空间的最小比例。
ini_trans	numeric	事务的初始数量。
max_trans	numeric	事务数量的最大值。
avg_row_len	integer	平均每行的字节数。
partitioned	character varying(3)	表是否为分区表。 ● yes：是分区表。 ● no：不是分区表。
last_analyzed	timestamp with time zone	上次分析表的时间。数据库重启后，数据会丢失。
row_movement	character varying(8)	是否允许分区行移动。 ● enabled：允许分区行移动。 ● disabled：不允许分区行移动。
compression	character varying(8)	是否启用表压缩。 ● enabled：启用表压缩。 ● disabled：不启用表压缩。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
duration	character varying(15)	临时表的期限。 ● NULL：非临时表。 ● sys$session：会话临时表。 ● sys$transaction：事务临时表。
logical_replication	character varying(8)	表是否启用逻辑复制。 ● enabled：启用逻辑复制。 ● disabled：不启用逻辑复制。
external	character varying(3)	表是否为外表。 ● yes：是外表。 ● no：不是外表。
logging	character varying(3)	表的更改是否记录日志。 ● yes：表的更改记录日志。 ● no：表的更改不记录日志。
default_collation	character varying(100)	表的默认排序规则。 ● default
degree	character varying(10)	扫描表的实例数量。
table_lock	character varying(8)	是否启用表级锁。 ● enabled：启用表级锁。 ● disabled：不启用表级锁。
nested	character varying(3)	是否为嵌套表。 ● yes：是嵌套表。 ● no：不是嵌套表。
buffer_pool	character varying(7)	表的默认缓冲池。
flash_cache	character varying(7)	用于表块的数据库智能闪存提示。
cell_flash_cache	character varying(7)	用于表块的单元闪存缓存提示。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
skip_corrupt	character varying(8)	扫描表是否跳过损坏的块。 ● enabled：跳过损坏的块。 ● disabled：不跳过损坏的块。
has_identity	character varying(3)	表是否具有标识列。 ● yes：有标识列。 ● no：没有标识列。
inmemory	character varying(8)	暂不支持，取值为disabled。
segment_created	character varying(3)	表段是否已被创建。 ● yes：表段已被创建。 ● no：表段未被创建。
monitoring	character varying(3)	是否跟踪表的修改。 ● yes：跟踪表的修改。 ● no：不跟踪表的修改。
cluster_name	character varying(128)	暂不支持，值为NULL。
iot_name	character varying(128)	暂不支持，值为NULL。
pct_used	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
backed_up	character varying(1)	暂不支持，值为NULL。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_space_freelist_blocks	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
num_freelist_blocks	numeric	暂不支持，值为NULL。
instances	character varying(10)	暂不支持，值为NULL。
cache	character varying(5)	暂不支持，值为NULL。
iot_type	character varying(12)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
cluster_owner	character varying(30)	暂不支持，值为NULL。
dependencies	character varying(8)	暂不支持，值为NULL。
compression_for	character varying(30)	暂不支持，值为NULL。
read_only	character varying(3)	暂不支持，值为NULL。
result_cache	character varying(7)	暂不支持，值为NULL。
clustering	character varying(3)	暂不支持，值为NULL。
activity_tracking	character varying(23)	暂不支持，值为NULL。
dml_timestamp	character varying(25)	暂不支持，值为NULL。
container_data	character varying(3)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
duplicated	character varying(1)	暂不支持，值为NULL。
sharded	character varying(1)	暂不支持，值为NULL。
hybrid	character varying(3)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
containers_default	character varying(3)	暂不支持，值为NULL。
container_map	character varying(3)	暂不支持，值为NULL。
extended_data_link	character varying(3)	暂不支持，值为NULL。
extended_data_link_map	character varying(3)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
container_map_object	character varying(3)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。
has_sensitive_column	character varying(3)	暂不支持，值为NULL。
admit_null	character varying(3)	暂不支持，值为NULL。
data_link_dml_enabled	character varying(3)	暂不支持，值为NULL。
object_id_type	character varying(16)	暂不支持，值为NULL。
table_type_owner	character varying(128)	暂不支持，值为NULL。
table_type	character varying(128)	暂不支持，值为NULL。
compress_for	character varying(30)	暂不支持，值为NULL。

7.3.13.35 ADM_TABLESPACES

ADM_TABLESPACES视图显示有关可用的表空间的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。ORA数据库与GaussDB数据库逻辑结构特性不一致。

表 7-317 ADM_TABLESPACES 字段

名称	类型	描述
tablespace_name	character varying(64)	表空间名称。
block_size	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
min_extlen	numeric	暂不支持，值为NULL。
contents	character varying(9)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
status	character varying(9)	暂不支持，值为online。
logging	character varying(9)	暂不支持，值为NULL。
force_logging	character varying(3)	暂不支持，值为NULL。
extent_management	character varying(10)	暂不支持，值为NULL。
allocation_type	character varying(9)	暂不支持，值为NULL。
plugged_in	character varying(3)	暂不支持，值为NULL。
segment_space_management	character varying(6)	暂不支持，值为NULL。
def_tab_compression	character varying(8)	暂不支持，值为NULL。
retention	character varying(11)	暂不支持，值为NULL。
bigfile	character varying(3)	暂不支持，值为NULL。
predicate_evaluation	character varying(7)	暂不支持，值为NULL。
encrypted	character varying(3)	暂不支持，值为NULL。
compress_for	character varying(30)	暂不支持，值为NULL。
def_inmemory	character varying(8)	暂不支持，值为NULL。
def_inmemory_priority	character varying(8)	暂不支持，值为NULL。
def_inmemory_distribute	character varying(15)	暂不支持，值为NULL。
def_inmemory_compression	character varying(17)	暂不支持，值为NULL。
def_inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
shared	character varying(12)	暂不支持，值为NULL。
def_index_compression	character varying(8)	暂不支持，值为NULL。
index_compress_for	character varying(13)	暂不支持，值为NULL。
def_cellmemory	character varying(14)	暂不支持，值为NULL。
def_inmemory_service	character varying(12)	暂不支持，值为NULL。
def_inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
lost_write_protect	character varying(7)	暂不支持，值为NULL。
chunk_tablespace	character varying(1)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.13.36 ADM_TRIGGERS

ADM_TRIGGERS视图显示数据库中的触发器信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-318 ADM_TRIGGERS 字段

名称	类型	描述
owner	character varying(128)	触发器的所有者。
trigger_name	character varying(64)	触发器名称。
trigger_type	character varying	触发器触发的时机。取值范围：before statement、before each row、after statement、after each row、instead of。
triggering_event	character varying	触发触发器的事件。取值范围：update、insert、delete、truncate。
table_owner	character varying(64)	定义触发器的表的所有者。
base_object_type	character varying(18)	定义触发器的基础对象。取值范围：table，view。
table_name	character varying(64)	定义触发器的表或视图名称。
column_name	character varying(4000)	暂不支持，值为NULL。
referencing_name	character varying(422)	暂不支持，值为referencing new as new old as old。
when_clause	character varying(4000)	when子句的内容，必须求值为true才能执行trigger_body。
status	character varying(64)	触发器的状态。 ● O：触发器在“origin”和“local”模式下触发。 ● D：触发器被禁用。 ● R：触发器在“replica”模式下触发。 ● A：触发器始终触发。
description	character varying(4000)	触发器描述。用于重新创建触发器创建语句。
action_type	character varying(11)	触发体的动作类型，仅支持call。
trigger_body	text	触发器触发时执行的语句。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
crossedition	character varying(7)	暂不支持，值为no。
before_statement	character varying(3)	暂不支持，值为no。
before_row	character varying(3)	暂不支持，值为no。
after_row	character varying(3)	暂不支持，值为no。
after_statement	character varying(3)	暂不支持，值为no。
instead_of_row	character varying(3)	暂不支持，值为no。
fire_once	character varying(3)	暂不支持，值为NULL。
apply_server_only	character varying(3)	暂不支持，值为NULL。

7.3.13.37 ADM_TYPE_ATTRS

ADM_TYPE_ATTRS视图描述当前数据库对象类型的属性。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-319 ADM_TYPE_ATTRS 字段

名称	类型	描述
owner	oid	该类型的所有者。
type_name	name	数据类型名称。
attr_name	name	字段名。
attr_type_mod	integer	记录创建新表时支持的类型特定的数据（比如一个varchar字段的最大长度）。它传递给类型相关的输入和长度转换函数当做第三个参数。其值对那些不需要ATTTYPMOD的类型通常为-1。
attr_type_owner	oid	该类型属性的所有者。
attr_type_name	name	数据类型属性名称。目前该字段记录的是转换后的类型名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
length	smallint	对于定长类型是该类型内部表现形式的字节数目。对于变长类型是负数。 ● -1表示一种“变长”（有长度字属性的数据）。 ● -2表示这是一个NULL结尾的C字符串。
precision	integer	数字类型的精度。
scale	integer	数字类型的范围。
character_set_name	character(1)	属性的字符集名称。 ● C：CHAR_CS。 ● N：NCHAR_CS。
attr_no	smallint	属性编号。
inherited	character(1)	表示属性是否继承自超级类型（y或n）。
attr_length	integer	记录创建新表时支持的类型特定的数据（比如一个varchar字段的最大长度）。对于raw类型，因内核实现原因，暂未记录。

7.3.13.38 ADM_TYPES

ADM_TYPES视图用于描述数据库中的所有对象类型。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-320 ADM_TYPES 字段

名称	类型	描述
owner	character varying(128)	类型的所有者。
type_name	character varying(128)	类型名称。
type_oid	raw	类型的标识符（OID）。
typecode	character varying(128)	类型的类型代码。
attributes	numeric	类型中的属性数。
methods	numeric	暂不支持，值为0。
predefined	character varying(3)	该类型是否是内置类型。
incomplete	character varying(3)	该类型是否是不完整类型。
final	character varying(3)	暂不支持，值为NULL。
instantiable	character varying(3)	暂不支持，值为NULL。
persistable	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
supertype_owner	character varying(128)	暂不支持，值为NULL。
supertype_name	character varying(128)	暂不支持，值为NULL。
local_attributes	numeric	暂不支持，值为NULL。
local_methods	numeric	暂不支持，值为NULL。
typeid	raw	暂不支持，值为NULL。

7.3.13.39 ADM_VIEWS

ADM_VIEWS视图显示数据库中的视图信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-321 ADM_VIEWS 字段

名称	类型	描述
owner	character varying(64)	视图的所有者。
view_name	character varying(64)	视图名称。
text	text	视图文本。
text_length	integer	视图文本长度。
text_vc	character varying(4000)	视图创建语句。此列可能会截断视图文本。BEQUEATH子句将不会作为此视图中的TEXT_VC列的一部分出现。
type_text_length	numeric	暂不支持，值为NULL。
type_text	character varying(4000)	暂不支持，值为NULL。
oid_text_length	numeric	暂不支持，值为NULL。
oid_text	character varying(4000)	暂不支持，值为NULL。
view_type_owner	character varying(128)	暂不支持，值为NULL。
view_type	character varying(128)	暂不支持，值为NULL。
superview_name	character varying(128)	暂不支持，值为NULL。
editioning_view	character varying(1)	暂不支持，值为NULL。
read_only	character varying(1)	暂不支持，值为NULL。
container_data	character varying(1)	暂不支持，值为NULL。
bequeath	character varying(12)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
origin_con_id	character varying(256)	暂不支持，值为NULL。
default_collation	character varying(100)	暂不支持，值为NULL。
containers_default	character varying(3)	暂不支持，值为NULL。
container_map	character varying(3)	暂不支持，值为NULL。
extended_data_link	character varying(3)	暂不支持，值为NULL。
extended_data_link_map	character varying(3)	暂不支持，值为NULL。
has_sensitive_column	character varying(3)	暂不支持，值为NULL。
admit_null	character varying(3)	暂不支持，值为NULL。
pdb_local_only	character varying(3)	暂不支持，值为NULL。

7.3.13.40 DB_ARGUMENTS

DB_ARGUMENTS视图显示当前用户可访问的存储过程和函数的参数信息。该视图同时存在于PG_CATALOG和SYS Schema下。该视图所有用户都可以访问，显示当前用户可访问的所有信息。

表 7-322 DB_ARGUMENTS 字段

名称	类型	描述
owner	character varying(128)	函数或存储过程的所有者。
object_name	character varying(128)	函数或存储过程的名称。
package_name	character varying(128)	暂不支持，值为NULL。
object_id	oid	函数或存储过程的oid。
overload	character varying(40)	表示该函数是该名称的第n个重载函数。
subprogram_id	numeric	暂不支持，值为NULL。
argument_name	character varying(128)	参数名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
position	numeric	该参数在参数列表中的位置，函数返回值的位置默认为0。
sequence	numeric	定义参数的顺序，从1开始，返回类型在前，然后是每个参数。
data_level	numeric	复合类型参数的嵌套深度，此列的值始终为0，因为每个参数现只显示一行。
data_type	character varying(64)	参数的数据类型。
defaulted	character varying(1)	参数是否有默认值。 ● y：有默认值。 ● n：没有默认值。
default_value	text	暂不支持，值为NULL。
default_length	numeric	暂不支持，值为NULL。
in_out	character varying(9)	参数出入属性。取值范围： ● in：入参。 ● out：出参。 ● in/out：出入参。 ● variadic：VARIADIC参数。
data_length	numeric	暂不支持，值为NULL。
data_precision	numeric	暂不支持，值为NULL。
data_scale	numeric	暂不支持，值为NULL。
radix	numeric	数字的参数基数。smallint、integer、bigint、numeric、float为10，其余值为NULL。
character_set_name	character varying(44)	暂不支持，值为NULL。
type_owner	character varying(128)	数据类型所有者。
type_name	character varying(128)	参数类型名，仅显示自定义类型。
type_subname	character varying(128)	暂不支持，值为NULL。
type_link	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
type_object_type	character varying(7)	由type_owner、type_name和type_subname列描述的类型的类型。 ● table：参数为表类型。 ● view：参数为视图类型。 ● NULL：参数为非表、非视图类型。
pls_type	character varying(128)	对于数字类型参数，为参数的PL/SQL类型的名称，否则为空。
char_length	numeric	暂不支持，值为NULL。
char_used	character varying(1)	暂不支持。 ● b：varchar、nvarchar2、bpchar、char类型。 ● NULL：其余类型。
origin_con_id	character varying(256)	暂不支持，值为0。

7.3.13.41 DB_ALL_TABLES

DB_ALL_TABLES视图显示当前用户所能访问的表和视图的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-323 DB_ALL_TABLES 字段

名称	类型	描述
owner	name	表或视图的所有者。
table_name	name	表或视图的名称。
tablespace_name	name	表或视图所在的表空间。

7.3.13.42 DB_COL_COMMENTS

DB_COL_COMMENTS视图显示当前用户可访问的表中字段的注释信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-324 DB_COL_COMMENTS 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
column_name	character varying(64)	列名。
comments	text	注释。
schema	character varying(64)	列所属的名称空间的名称。

7.3.13.43 DB_COLL_TYPES

DB_COLL_TYPES视图显示当前用户可访问的所有集合类型的信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-325 DB_COLL_TYPES 字段

名称	类型	描述
owner	character varying(128)	集合的所有者。
type_name	character varying(128)	集合的名称。
coll_type	character varying(128)	集合的描述。
upper_bound	numeric	暂不支持，值为NULL。
elem_type_mod	character varying(7)	元素的类型修饰。
elem_type_owner	character varying(128)	集合基于的元素类型的所有者。该值主要用于用户定义的类型。
elem_type_name	character varying(128)	集合所依据的数据类型或用户定义类型的名称。
length	numeric	暂不支持，值为NULL。
precision	numeric	暂不支持，值为NULL。
scale	numeric	暂不支持，值为NULL。
character_set_name	character varying(44)	暂不支持，值为NULL。
elem_storage	character varying(7)	暂不支持，值为NULL。
nulls_stored	character varying(3)	暂不支持，值为NULL。
char_used	character varying(1)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.13.44 DB_CONS_COLUMNS

DB_CONS_COLUMNS视图显示当前用户可访问的约束字段的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-326 DB_CONS_COLUMNS 字段

名称	类型	描述
constraint_name	character varying(64)	约束名。
table_name	character varying(64)	约束相关的表名。
column_name	character varying(64)	约束相关的列名。
owner	character varying(64)	约束的创建者。
position	smallint	表中列的位置。

7.3.13.45 DB_CONSTRAINTS

DB_CONSTRAINTS视图显示当前用户可访问的约束的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-327 DB_CONSTRAINTS 字段

名称	类型	描述
constraint_name	character varying(64)	约束名。
constraint_type	text	约束类型： ● c表示检查约束。 ● f表示外键约束。 ● p表示主键约束。 ● u表示唯一约束。
table_name	character varying(64)	约束相关的表名。
index_owner	character varying(64)	约束相关的索引的所有者（只针对唯一约束和主键约束）。
index_name	character varying(64)	约束相关的索引名（只针对唯一约束和主键约束）。
owner	character varying(64)	约束的创建者。

版权所有 © 华为云计算技术有限公司

7.3.13.46 DB_DEPENDENCIES

DB_DEPENDENCIES视图显示当前用户可访问的类型、表、视图、存储过程、函数、触发器之间的依赖关系。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-328 DB_DEPENDENCIES 字段

名称	类型	描述
owner	name	对象的所有者。
name	name	对象的名称。
type	character varying(18)	对象的类型。
referenced_owner	name	被引用对象的所有者。
referenced_name	name	被引用对象的名称。
referenced_type	character varying(18)	被引用对象的类型。
referenced_link_name	character varying(128)	暂不支持，值为NULL。
dependency_type	character varying(4)	暂不支持，值为NULL。

7.3.13.47 DB_IND_COLUMNS

DB_IND_COLUMNS视图显示当前用户可访问的所有索引的字段信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-329 DB_IND_COLUMNS 字段

名称	类型	描述
index_owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名。
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
column_name	name	列名。
column_position	smallint	索引中列的位置。
column_length	numeric	列的长度。
char_length	numeric	列的长度（字符类型）。
descend	character varying(4)	列的排序方式： • 降序（DESC）。 • 升序（ASC）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
collated_column_id	numeric	此列提供语言排序的列的内部序列号。

7.3.13.48 DB_IND_EXPRESSIONS

DB_IND_EXPRESSIONS视图显示当前用户可访问的表达式索引的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-330 DB_IND_EXPRESSIONS 字段

名称	类型	描述
index_owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名。
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
column_expression	text	定义列的基于函数的索引表达式。
column_position	smallint	索引中列的位置。

7.3.13.49 DB_INDEXES

DB_INDEXES视图显示当前用户可访问的索引信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-331 DB_INDEXES 字段

名称	类型	描述
owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名称。
table_name	character varying(64)	索引对应的表名。
uniqueness	text	该索引是否为唯一索引。 • UNIQUE：唯一索引。 • NONUNIQUE：非唯一索引。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
partitioned	character(3)	该索引是否具有分区表的性质。 ● Yes：索引具有分区表的性质。 ● No：索引不具有分区表的性质。
generated	character varying(1)	该索引的名称是否为系统生成。 ● y：索引名称为系统生成。 ● n：索引名称非系统生成。
index_type	character varying(27)	索引类型。 ● NORMAL：索引属性都是简单的引用，表达式树为空。 ● FUNCTION-BASED NORMAL：存在表达式树用于非简单字段引用的索引属性。
table_owner	character varying(128)	索引对象的所有者。
table_type	character(11)	索引对象的类型。 ● TABLE：索引对象为表类型。
tablespace_name	character varying(30)	包含索引的表空间名称。
status	character varying(8)	非分区索引状态。 ● VALID：非分区索引可以用于查询。 ● UNUSABLE：非分区索引不可用。 ● N/A：索引具有分区表性质。
compression	character varying(13)	暂不支持，值为NULL。
prefix_length	numeric	暂不支持，值为NULL。
ini_trans	numeric	暂不支持，值为NULL。
max_trans	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
pct_threshold	numeric	暂不支持，值为NULL。
include_column	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
pct_free	numeric	暂不支持，值为NULL。
logging	character varying(3)	暂不支持，值为NULL。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。
clustering_factor	numeric	暂不支持，值为NULL。
num_rows	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	date	暂不支持，值为NULL。
degree	character varying(40)	暂不支持，值为NULL。
instances	character varying(40)	暂不支持，值为NULL。
temporary	character varying(1)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
buffer_pool	character varying(7)	暂不支持，值为NULL。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
duration	character varying(15)	暂不支持，值为NULL。
pct_direct_access	numeric	暂不支持，值为NULL。
ityp_owner	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
ityp_name	character varying(128)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
domidx_status	character varying(12)	暂不支持，值为NULL。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
funcidx_status	character varying(8)	暂不支持，值为NULL。
join_index	character varying(3)	暂不支持，值为NULL。
iot_redundant_pkey_elim	character varying(3)	暂不支持，值为NULL。
dropped	character varying(3)	暂不支持，值为NULL。
visibility	character varying(9)	该索引对优化器是否可见。 ● VISIBLE：索引对优化器可见。 ● INVISIBLE：索引对优化器不可见。
domidx_management	character varying(14)	暂不支持，值为NULL。
segment_created	character varying(3)	暂不支持，值为NULL。
orphaned_entries	character varying(3)	暂不支持，值为NULL。
indexing	character varying(7)	暂不支持，值为NULL。
auto	character varying(3)	暂不支持，值为NULL。

7.3.13.50 DB_OBJECTS

DB_OBJECTS视图显示当前用户可访问的数据库对象的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-332 DB_OBJECTS 字段

名称	类型	描述
owner	name	对象的所有者。
object_name	name	对象的名称。
object_id	oid	对象的OID。
object_type	name	对象的类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
namespace	oid	对象所在的命名空间的ID。
temporary	character(1)	对象是否为临时对象。
status	character varying(7)	对象的状态。 ● valid：有效。 ● invalid：已失效。
subobject_name	name	对象的子对象名称。
generated	character(1)	对象名称是否是系统生成。
created	timestamp with time zone	对象的创建时间。
last_ddl_time	timestamp with time zone	对象的最后修改时间。
default_collation	character varying(100)	对象的默认排序规则。
data_object_id	numeric	暂不支持，值为NULL。
timestamp	character varying(19)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
edition_name	character varying(128)	暂不支持，值为NULL。
sharing	character varying(18)	暂不支持，值为NULL。
editionable	character varying(1)	暂不支持，值为NULL。
oracle_maintained	character varying(1)	暂不支持，值为NULL。
application	character varying(1)	暂不支持，值为NULL。
duplicated	character varying(1)	暂不支持，值为NULL。
sharded	character varying(1)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
created_appid	numeric	暂不支持，值为NULL。
modified_appid	numeric	暂不支持，值为NULL。
created_vsnid	numeric	暂不支持，值为NULL。
modified_vsnid	numeric	暂不支持，值为NULL。

须知

created和last_ddl_time支持的范围请参见**PG_OBJECT**中的记录范围。

7.3.13.51 DB_PROCEDURES

DB_PROCEDURES视图显示当前用户可访问的所有存储过程和函数的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-333 DB_PROCEDURES 字段

名称	类型	描述
owner	name	对象的所有者。
object_name	name	对象的名称。

7.3.13.52 DB_SCHEDULER_JOB_ARGS

DB_SCHEDULER_JOB_ARG视图显示当前用户可访问任务的有关参数信息。该视图所有用户可访问，显示当前用户可访问的所有信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-334 DB_SCHEDULER_JOB_ARGS 字段

名称	类型	描述
owner	character varying(128)	参数所属作业的拥有者。
job_name	character varying(128)	参数所属作业名。
argument_name	character varying(128)	参数名称。
argument_position	numeric	参数在参数列表中的位置。
argument_type	character varying(257)	参数的数据类型，可以是用户的自定义数据类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
value	character varying(4000)	参数值。
anydata_value	character varying(4000)	暂不支持，值为NULL。
out_argument	character varying(5)	保留字段，值为NULL。

7.3.13.53 DB_SCHEDULER_PROGRAM_ARGS

DB_SCHEDULER_PROGRAM_ARG视图显示当前用户可访问程序的有关参数信息。该视图所有用户可访问，显示当前用户可访问的所有信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-335 DB_SCHEDULER_PROGRAM_ARGS 字段

名称	类型	描述
owner	character varying(128)	参数所属程序的拥有者。
program_name	character varying(128)	参数所属程序名。
argument_name	character varying(128)	参数名称。
argument_position	numeric	参数在参数列表中的位置。
argument_type	character varying(257)	参数的数据类型，可以是用户的自定义数据类型。
metadata_attribute	character varying(19)	暂不支持，值为NULL。
default_value	character varying(4000)	参数默认值。
anydata_default_value	character varying(4000)	暂不支持，值为NULL。
out_argument	character varying(5)	保留字段，值为NULL。

7.3.13.54 DB_SEQUENCES

DB_SEQUENCES视图显示当前用户能够访问的所有序列的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-336 DB_SEQUENCES 字段

名称	类型	描述
sequence_owner	name	序列所有者。
sequence_name	name	序列的名称。
min_value	int16	序列最小值。
max_value	int16	序列最大值。
increment_by	int16	序列的增量。
cycle_flag	character(1)	序列是否是循环序列。取值范围： ● y表示是循环序列。 ● n表示不是循环序列。
order_flag	character(1)	序列是否是有序序列。取值范围： ● y：有序序列。 ● n：不是有序序列。
cache_size	int16	序列磁盘缓存大小。
last_number	int16	该序列从GTM中一次性批量申请的最后一个值。
scale_flag	character varying(1)	标志是否为可扩展序列。暂不支持，值为NULL。
extend_flag	character varying(1)	标志可扩展序列生成的值是否超出序列最大值、最小值范围。暂不支持，值为NULL。
sharded_flag	character varying(1)	标志是否为分片序列。暂不支持，值为NULL。
session_flag	character varying(1)	标志序列是否为私有会话。暂不支持，值为NULL。
keep_value	character varying(1)	标志在失败后的replay期间是否保留序列值。暂不支持，值为NULL。

7.3.13.55 DB_SOURCE

DB_SOURCE视图显示当前用户可访问的存储过程、函数、触发器的定义信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-337 DB_SOURCE 字段

名称	类型	描述
owner	name	对象的所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
name	name	对象名字。
type	name	对象类型：function、procedure、trigger。
line	numeric	此行在定义信息中的行号。
text	text	存储对象的文本来源。
origin_con_id	character varying(256)	暂不支持，值为0。

7.3.13.56 DB_SYNONYMS

DB_SYNONYMS视图显示当前用户可访问的所有同义词信息。

表 7-338 DB_SYNONYMS 字段

名称	类型	描述
owner	text	同义词的所有者。PUBLIC同义词的所有者为PUBLIC。
schema_name	text	同义词所属模式名。PUBLIC同义词的所属模式名NULL。
synonym_name	text	同义词的名称。
table_owner	text	关联对象的所有者。尽管该列称为table_owner，但它拥有的该关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
table_name	text	关联对象名。尽管该列称为table_name，但此关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
table_schema_name	text	关联对象所属模式名。尽管该列称为table_schema_name，但此schema下的该关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
db_link	character varying(128)	保留字段，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为0。

版权所有 © 华为云计算技术有限公司

7.3.13.57 DB_TAB_COL_STATISTICS

DB_TAB_COL_STATISTICS视图显示从DB_TAB_COLUMNS中提取的列统计信息和直方图信息。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-339 DB_TAB_COL_STATISTICS 字段

名称	类型	描述
owner	character varying(128)	表的所有者。
table_name	character varying(128)	表名。
column_name	character varying(128)	列名。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的低值。
high_value	raw	列中的高值。
density	numeric	• 如果COLUMN_NAME上有直方图，则此列将显示直方图中跨越少于2个端点的值的选择性。它不代表跨越2个或更多端点的值的选择性。 • 如果COLUMN_NAME上没有可用的直方图，则该列的值为1/NUM_DISTINCT。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图中的桶数。
sample_size	numeric	用于分析此列的样本量。
last_analyzed	date	最近分析此列的日期。数据库重启后，数据会丢失。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
notes	character varying(99)	暂不支持，值为NULL。
avg_col_len	numeric	列的平均长度（以字节为单位）。
histogram	character varying(15)	直方图是否存在以及其类型： • NONE：不存在直方图。 • FREQUENCY：频率直方图。 • EQUI-WIDTH：等宽直方图。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
scope	character varying(7)	对于在除全局临时表之外的任何表上收集的统计信息，该值是SHARED（表示统计信息在所有会话之间共享）。
schema	character varying(64)	列所属的名称空间的名称。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。

7.3.13.58 DB_TAB_COLUMNS

DB_TAB_COLUMNS视图显示当前用户可访问的表和视图的列的描述信息。该视图同时存在于PG_CATALOG和SYS Schema下。该视图所有用户可访问，显示当前用户可访问的所有信息。

表 7-340 DB_TAB_COLUMNS 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表的名称。
column_name	character varying(64)	列的名称。
data_type	character varying(128)	列的数据类型。
data_type_mod	character varying(3)	暂不支持，值为NULL。
data_type_owner	character varying(128)	列的数据类型的所有者。
data_length	integer	列的字节长度。
data_precision	integer	数据类型的精度。对于numeric数据类型有效，其他类型为NULL。
data_scale	integer	小数点右边的位数。对于numeric数据类型有效，其他类型为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
nullable	bpchar	该列是否允许为空。取值范围： • y：允许。 • n：不允许。对于主键约束和非空约束，该值为n。
column_id	integer	创建表时列的序号。
default_length	numeric	列的默认值字节长度。
data_default	text	列的默认值。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的最小值。
high_value	raw	列中的最大值。
density	numeric	列密度。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
last_analyzed	date	上次分析的日期。
sample_size	numeric	用于分析此列的样本量。
character_set_name	character varying(44)	暂不支持，值为NULL。
char_col_decl_length	numeric	字符类型列的声明长度。
global_stats	character varying(3)	暂不支持，值为no。
user_stats	character varying(3)	暂不支持，值为no。
avg_col_len	numeric	列的平均长度（单位字节）。
char_length	numeric	列的长度（以字符计），只对varchar，nvarchar2，bpchar，char类型有效。
char_used	character varying(1)	暂不支持，varchar、bpchar与char类型值为b；nvarchar2类型值为c；其余值为NULL。
v80_fmt_image	character varying(3)	暂不支持，值为NULL。
data_upgraded	character varying(3)	暂不支持，值为yes。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
histogram	character varying(15)	直方图是否存在以及其类型： ● none：不存在直方图。 ● frequency：频率直方图。 ● equi-width：等宽直方图。
default_on_null	character varying(3)	暂不支持，值为NULL。
identity_column	character varying(3)	列是否为标识列： ● yes：表示为标识列。 ● no：表示为非标识列。
evaluation_edition	character varying(128)	暂不支持，值为NULL。
unusable_before	character varying(128)	暂不支持，值为NULL。
unusable_beginning	character varying(128)	暂不支持，值为NULL。
collation	character varying(100)	列的排序规则。因该字段与保留关键字冲突，调用该字段需加视图名。
comments	text	注释。
schema	character varying(64)	列所属的名称空间的名称。

说明

- 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。
- 目前CREATE TABLE语法不支持创建具有标识列字段的表，当进行查询DB_TAB_COLUMNS视图操作时，返回结果中identity_column字段值恒为no。

7.3.13.59 DB_TAB_COMMENTS

DB_TAB_COMMENTS视图显示当前用户可访问的所有表和视图的注释信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-341 DB_TAB_COMMENTS 字段

名称	类型	描述
owner	character varying(128)	表或视图的所有者。
table_name	character varying(128)	表或视图的名称。
table_type	character varying(11)	对象类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
comments	text	注释。
origin_con_id	numeric	暂不支持，值为0。
schema	character varying(64)	表所属的名称空间的名称。

7.3.13.60 DB_TAB_HISTOGRAMS

DB_TAB_HISTOGRAMS视图显示当前用户可访问的表和视图的直方图统计信息，即表各列数据的分布情况。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-342 DB_TAB_HISTOGRAMS 字段

名称	类型	描述
owner	character varying(128)	表的拥有者。
table_name	character varying(128)	表名。
column_name	character varying(4000)	列名。
endpoint_number	numeric	直方图的桶号。
endpoint_value	numeric	暂不支持，值为NULL。
endpoint_actual_value	character varying(4000)	桶端点的实际值。
endpoint_actual_value_raw	raw	暂不支持，值为NULL。
endpoint_repeat_count	numeric	暂不支持，值为NULL。
scope	character varying(7)	暂不支持，值为shared。

7.3.13.61 DB_TAB_MODIFICATIONS

DB_TAB_MODIFICATIONS视图显示自上次收集表统计信息以来当前用户可访问的表的修改信息的统计数据。该视图目前只显示执行了INSERT、DELETE和UPDATE操作的表的统计数据。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-343 DB_TAB_MODIFICATIONS 字段

名称	类型	描述
table_owner	character varying(128)	表的所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
table_name	character varying(128)	表的名称。
partition_name	character varying(128)	分区的名称。
subpartition_name	character varying(128)	子分区的名称。
inserts	numeric	自上次收集统计信息以来插入次数的估算值。
updates	numeric	自上次收集统计信息以来更新次数的估算值。
deletes	numeric	自上次收集统计信息以来删除次数的估算值。
timestamp	date	上次修改表的时间。 暂不支持分区表的修改时间，值为NULL。 • O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
truncated	character varying(3)	暂不支持，值为NULL。
drop_segments	numeric	暂不支持，值为NULL。
schema_name	character varying(128)	表所属模式名。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的timestamp字段类型会变更为datea。

7.3.13.62 DB_TAB_STATS_HISTORY

DB_TAB_STATS_HISTORY视图显示表统计信息涉及的表、分区或者子分区以及执行收集表统计信息的时间。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-344 DB_TAB_STATS_HISTORY 字段

名称	类型	描述
owner	character varying(128)	对象的拥有者。
table_name	character varying(128)	表名。
partition_name	character varying(128)	暂不支持，值为NULL。
subpartition_name	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stats_update_time	timestamp(6) with time zone	统计信息更新的时间，数据库重启后，数据会丢失。

7.3.13.63 DB_TABLES

DB_TABLES视图显示当前用户可访问的所有表的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-345 DB_TABLES 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
tablespace_name	character varying(64)	存储表的表空间名称。
dropped	character varying	当前表是否已删除： ● yes：已删除。 ● no：未删除。
num_rows	numeric	表的估计行数。
status	character varying(8)	当前表是否有效。 ● valid：当前表有效。 ● unusable：当前表不可用。
sample_size	numeric	分析表使用的样本数量。
temporary	character(1)	表是否为临时表： ● y：是临时表。 ● n：不是临时表。
pct_free	numeric	块中空闲空间的最小比例。
ini_trans	numeric	事务的初始数量。
max_trans	numeric	事务数量的最大值。
avg_row_len	integer	平均每行的字节数。
partitioned	character varying(3)	表是否为分区表。 ● yes：是分区表。 ● no：不是分区表。
last_analyzed	timestamp with time zone	上次分析表的时间。数据库重启后，数据会丢失。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
row_movement	character varying(8)	是否允许分区行移动。 ● enabled：允许分区行移动。 ● disabled：不允许分区行移动。
compression	character varying(8)	是否启用表压缩。 ● enabled：启用表压缩。 ● disabled：不启用表压缩。
duration	character varying(15)	临时表的期限。 ● NULL：表示非临时表。 ● sys$session：表示会话临时表。 ● sys$transaction：表示事务临时表。
logical_replication	character varying(8)	表是否启用逻辑复制。 ● enabled：启用逻辑复制。 ● disabled：不启用逻辑复制。
external	character varying(3)	表是否为外表。 ● yes：是外表。 ● no：不是外表。
logging	character varying(3)	表的更改是否记录日志。 ● yes：表的更改记录日志。 ● no：表的更改不记录日志。
default_collation	character varying(100)	表的默认排序规则。
degree	character varying(10)	扫描表的实例数量。
table_lock	character varying(8)	是否启用表级锁。 ● enabled：启用表级锁。 ● disabled：不启用表级锁。
nested	character varying(3)	是否为嵌套表。 ● yes：是嵌套表。 ● no：不是嵌套表。
buffer_pool	character varying(7)	表的默认缓冲池。
flash_cache	character varying(7)	用于表块的数据库智能闪存提示。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cell_flash_cache	character varying(7)	用于表块的单元闪存缓存提示。
skip_corrupt	character varying(8)	扫描表是否跳过损坏的块。 ● enabled：跳过损坏的块。 ● disabled：不跳过损坏的块。
has_identity	character varying(3)	表是否具有标识列。 ● yes：有标识列。 ● no：没有标识列。
inmemory	character varying(8)	暂不支持，取值为disabled。
segment_created	character varying(3)	表段是否已被创建。 ● yes：表段已被创建。 ● no：表段未被创建。
monitoring	character varying(3)	是否跟踪表的修改。 ● yes：跟踪表的修改。 ● no：不跟踪表的修改。
cluster_name	character varying(128)	暂不支持，值为NULL。
iot_name	character varying(128)	暂不支持，值为NULL。
pct_used	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
backed_up	character varying(1)	暂不支持，值为NULL。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_space_freelist_blocks	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
num_freelist_blocks	numeric	暂不支持，值为NULL。
instances	character varying(10)	暂不支持，值为NULL。
cache	character varying(5)	暂不支持，值为NULL。
iot_type	character varying(12)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
cluster_owner	character varying(30)	暂不支持，值为NULL。
dependencies	character varying(8)	暂不支持，值为NULL。
compression_for	character varying(30)	暂不支持，值为NULL。
read_only	character varying(3)	暂不支持，值为NULL。
result_cache	character varying(7)	暂不支持，值为NULL。
clustering	character varying(3)	暂不支持，值为NULL。
activity_tracking	character varying(23)	暂不支持，值为NULL。
dml_timestamp	character varying(25)	暂不支持，值为NULL。
container_data	character varying(3)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
duplicated	character varying(1)	暂不支持，值为NULL。
sharded	character varying(1)	暂不支持，值为NULL。
hybrid	character varying(3)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
containers_default	character varying(3)	暂不支持，值为NULL。
container_map	character varying(3)	暂不支持，值为NULL。
extended_data_link	character varying(3)	暂不支持，值为NULL。
extended_data_link_map	character varying(3)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
container_map_object	character varying(3)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。
has_sensitive_column	character varying(3)	暂不支持，值为NULL。
admit_null	character varying(3)	暂不支持，值为NULL。
data_link_dml_enabled	character varying(3)	暂不支持，值为NULL。
object_id_type	character varying(16)	暂不支持，值为NULL。
table_type_owner	character varying(128)	暂不支持，值为NULL。
table_type	character varying(128)	暂不支持，值为NULL。
compress_for	character varying(30)	暂不支持，值为NULL。

7.3.13.64 DB_TRIGGERS

DB_TRIGGERS视图显示当前用户能访问到的触发器的信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-346 DB_TRIGGERS 字段

名称	类型	描述
trigger_name	character varying(64)	触发器名称。
table_name	character varying(64)	关系表名称。
table_owner	character varying(64)	角色名称。
status	character varying(64)	触发器的状态。 ● O：触发器在“origin”和“local”模式下触发。 ● D：触发器被禁用。 ● R：触发器在“replica”模式下触发。 ● A：触发器始终触发。

7.3.13.65 DB_TYPES

DB_TYPES视图显示当前用户可访问的对象类型的信息。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。

版权所有 © 华为云计算技术有限公司

表 7-347 DB_TYPES 字段

名称	类型	描述
owner	character varying(128)	类型的所有者。
type_name	character varying(128)	类型名称。
type_oid	raw	类型的标识符（OID）。
typecode	character varying(128)	类型的类型代码。
attributes	numeric	类型中的属性数。
methods	numeric	暂不支持，值为0。
predefined	character varying(3)	该类型是否是内置类型。
incomplete	character varying(3)	该类型是否为不完整类型。
final	character varying(3)	暂不支持，值为NULL。
instantiable	character varying(3)	暂不支持，值为NULL。
persistable	character varying(3)	暂不支持，值为NULL。
supertype_owner	character varying(128)	暂不支持，值为NULL。
supertype_name	character varying(128)	暂不支持，值为NULL。
local_attributes	numeric	暂不支持，值为NULL。
local_methods	numeric	暂不支持，值为NULL。
typeid	raw	暂不支持，值为NULL。

7.3.13.66 DB_VIEWS

DB_VIEWS视图显示当前用户可访问的所有视图的描述信息。该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-348 DB_VIEWS 字段

名称	类型	描述
owner	name	视图的所有者。
view_name	name	视图的名称。
text	text	视图文本。
text_length	integer	视图文本长度。
text_vc	character varying(4000)	视图创建语句。此列可能会截断视图文本。BEQUEATH子句将不会作为此视图中的TEXT_VC列的一部分出现。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
type_text_length	numeric	暂不支持，值为NULL。
type_text	character varying(4000)	暂不支持，值为NULL。
oid_text_length	numeric	暂不支持，值为NULL。
oid_text	character varying(4000)	暂不支持，值为NULL。
view_type_owner	character varying(128)	暂不支持，值为NULL。
view_type	character varying(128)	暂不支持，值为NULL。
superview_name	character varying(128)	暂不支持，值为NULL。
editioning_view	character varying(1)	暂不支持，值为NULL。
read_only	character varying(1)	暂不支持，值为NULL。
container_data	character varying(1)	暂不支持，值为NULL。
bequeath	character varying(12)	暂不支持，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为NULL。
default_collation	character varying(100)	暂不支持，值为NULL。
containers_default	character varying(3)	暂不支持，值为NULL。
container_map	character varying(3)	暂不支持，值为NULL。
extended_data_link	character varying(3)	暂不支持，值为NULL。
extended_data_link_map	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
has_sensitive_column	character varying(3)	暂不支持，值为NULL。
admit_null	character varying(3)	暂不支持，值为NULL。
pdb_local_only	character varying(3)	暂不支持，值为NULL。

7.3.13.67 DICT

DICT视图显示数据库中的数据字典表和系统视图的描述信息。所有用户都可以访问，该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-349 DICT 字段

名称	类型	描述
table_name	character varying(64)	对象的名称。
comments	character varying(4000)	对象上的文本注释。

7.3.13.68 DICTIONARY

DICTIONARY视图显示数据库中的数据字典表和系统视图的描述信息。所有用户都可以访问，该视图同时存在于PG_CATALOG和SYS Schema下。

表 7-350 DICTIONARY 字段

名称	类型	描述
table_name	character varying(64)	对象的名称。
comments	character varying(4000)	对象上的文本注释。

7.3.13.69 DUAL

DUAL视图是一个由系统自定创建的虚拟视图，具有一行一列的结构，主要用于执行不需要实际数据表的查询操作。任何用户都可以访问。该视图同时存在于PG_CATALOG和SYS schema中。

版权所有 © 华为云计算技术有限公司

表 7-351 DUAL 字段

名称	类型	描述
dummy	text	固定内容"X"。

7.3.13.70 DV_SESSIONS

DV_SESSIONS视图显示当前所有活动的后台线程的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。

表 7-352 DV_SESSIONS 字段

名称	类型	描述
sid	bigint	当前活动的后台线程的OID。
serial#	integer	当前活动的后台线程的序号，在GaussDB中为0。
user#	oid	登录此后台线程的用户的OID。oid为0表示此后台线程为全局辅助线程（auxiliary）。
username	name	登录此后台线程的用户名。username为空表示此后台线程为全局辅助线程（auxiliary）。 可以通过和pg_stat_get_activity()关联查询，识别出application_name。 例如： SELECT s.*,a.application_name FROM DV_SESSIONS AS s LEFT JOIN pg_stat_get_activity(NULL) AS a ON s.sid=a.sessionid;

7.3.13.71 DV_SESSION_LONGOPS

DV_SESSION_LONGOPS视图显示当前正在执行的操作的进度。该视图需要授权访问。

表 7-353 DV_SESSION_LONGOPS 字段

名称	类型	描述
sid	bigint	当前正在执行的后台线程的OID。
serial#	integer	当前正在执行的后台线程的序号，在GaussDB中为0。
sofar	integer	目前完成的工作量，在GaussDB中为空。
totalwork	integer	工作总量，在GaussDB中为空。

7.3.13.72 GET_GLOBAL_PREPARED_XACTS

GET_GLOBAL_PREPARED_XACTS视图显示全局所有节点上的两阶段残留事务信息。

版权所有 © 华为云计算技术有限公司

表 7-354 GET_GLOBAL_PREPARED_XACTS 字段

名称	类型	描述
transaction	xid	残留事务的xid。
gid	text	残留两阶段事务全局gid。
prepared	timestamp with time zone	残留两阶段事务的准备时间。
owner	name	残留两阶段事务的所有者。
database	name	残留两阶段事务所属的数据库。
node_name	text	残留事务所在的节点名称。

7.3.13.73 GLOBAL_BAD_BLOCK_INFO

GLOBAL_BAD_BLOCK_INFO视图，在CN上执行，统计所有实例数据页面损坏的情况，查询信息会显示损坏页面的基本信息。在DN上执行结果为空。根据这些信息利用**数据损坏检测修复函数**中的页面检测修复函数进行修复操作。默认只有初始用户、具有系统管理员属性的用户、在运维模式下具有运维管理员属性的用户以及监控用户可以查看，其余用户需要赋权后才可以使用。

表 7-355 GLOBAL_BAD_BLOCK_INFO 字段

名称	类型	描述
node_name	text	当前损坏页面的节点信息。
spc_node	oid	当前损坏页面对应的表空间id。
db_node	oid	当前损坏页面对应的数据库id。
rel_node	oid	当前损坏页面对应的relation的relfilenode。
bucket_node	integer	当前损坏页面的bucket_node，该字段在修复时作为是否是段页式表的判断依据。 • -1：非段页式表。 • 非0值：段页式表。
block_num	oid	当前损坏页面的页面号。
fork_num	integer	当前损坏页面的文件forknum。
file_path	text	当前损坏页面的相对路径。段页式表显示的是逻辑路径，非真实存在的文件。
check_time	timestamp with time zone	当前损坏页面检测出问题的时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
repair_time	timestamp with time zone	当前损坏页面被修复的时间。

7.3.13.74 GLOBAL_CLEAR_BAD_BLOCK_INFO

GLOBAL_CLEAR_BAD_BLOCK_INFO视图，在CN上执行，用于清理所有实例中已修复的损坏页面的信息，在DN上执行结果为空。默认只有初始用户、具有系统管理员属性的用户、在运维模式下具有运维管理员属性的用户以及监控用户可以查看，其余用户需要赋权后才可以使用。

表 7-356 GLOBAL_BAD_BLOCK_INFO 字段

名称	类型	描述
node_name	text	当前清理修复页面的节点信息。
result	boolean	当前实例清理修复页面执行的结果。

7.3.13.75 GLOBAL_SQL_PATCH

GLOBAL_SQL_PATCH视图显示所有SQL PATCH的信息，该视图仅在PG_CATALOG模式下存在。

表 7-357 GLOBAL_SQL_PATCH 字段

名称	类型	描述
node_name	text	SQL PATCH所在节点的名称。
patch_name	name	SQL PATCH的名称。
unique_sql_id	bigint	查询全局唯一ID。
owner	oid	SQL PATCH创建用户的ID。
enable	boolean	SQL PATCH是否生效。
status	"char"	SQL PATCH的状态（预留字段）。
abort	boolean	是否是废弃的Hint。
hint_string	text	Hint文本。
description	text	SQL PATCH的备注。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
parent_unique_sql_id	bigint	SQL PATCH生效的SQL语句的外层语句的全局唯一ID。 • 对于存储过程外的语句，该值为0。 • 对于存储过程内的语句，该值为调用该存储过程语句的全局唯一ID。
original_query	text	SQL文本。

7.3.13.76 GLOBAL_STAT_HOTKEYS_INFO

GLOBAL_STAT_HOTKEYS_INFO视图可用来查询整个集群中热点key的统计信息，查询结果按照count从大到小排序。

表 7-358 GLOBAL_STAT_HOTKEYS_INFO 字段

名称	类型	描述
database_name	text	热点key所在数据库的名称。
schema_name	text	热点key所在模式的名称。
table_name	text	热点key所在表的名称。
key_value	text	热点key的值。
hash_value	bigint	热点key在数据库中的哈希值。如果是List/Range分布表，该字段为0。
count	numeric	热点key被访问频次。

7.3.13.77 GLOBAL_WAL_SENDER_STATUS

GLOBAL_WAL_SENDER_STATUS视图显示当前集群主DN实例的redo日志传输和回放状态。该视图只有monitor admin和sysadmin权限可以查看。

表 7-359 GLOBAL_WAL_SENDER_STATUS 字段

名称	类型	描述
nodename	text	主节点名。
source_ip	text	主节点IP。
source_port	integer	主节点端口。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dest_ip	text	备节点IP。
dest_port	integer	备节点端口。
sender_pid	integer	发送线程PID。
local_role	text	主节点类型。 ● UNKNOWN_MODE： 状态未知。 ● NORMAL_MODE：单主机节点类型。 ● PRIMARY_MODE：节点类型为主节点。 ● STANDBY_MODE：节点类型为备节点。 ● CASCADE_STANDBY_MODE：节点类型为级联备节点。 ● PENDING_MODE：该节点在仲裁阶段。 ● RECOVERY_MODE：该节点在恢复阶段。 ● STANDBY_CLUSTER_MODE：该节点类型为备集群节点。 ● MAIN_STANDBY_MODE：该节点类型为首备集群节点。 **说明** 主节点类型期望值为: NORMAL_MODE、PRIMARY_MODE、PENDING_MODE、RECOVERY_MODE。如出现其他节点类型，请联系华为技术工程师提供技术支持。
peer_role	text	备节点类型。 ● UNKNOWN_MODE： 状态未知。 ● NORMAL_MODE：单主机节点类型。 ● PRIMARY_MODE：节点类型为主节点。 ● STANDBY_MODE：节点类型为备节点。 ● CASCADE_STANDBY_MODE：节点类型为级联备节点。 ● PENDING_MODE：该节点在仲裁阶段。 ● RECOVERY_MODE：该节点在恢复阶段。 ● STANDBY_CLUSTER_MODE：该节点类型为备集群节点。 ● MAIN_STANDBY_MODE：该节点类型为首备集群节点。 **说明** 备节点类型期望值为: NORMAL_MODE、STANDBY_MODE、CASCADE_STANDBY_MODE、PENDING_MODE、RECOVERY_MODE、STANDBY_CLUSTER_MODE、MAIN_STANDBY_MODE。如出现其他节点类型，请联系华为技术工程师提供技术支持。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
peer_state	text	备节点状态。 ● UNKNOWN_STATE：节点状态未知。 ● NORMAL_STATE：表示节点启动正常。 ● NEEDREPAIR_STATE：当前节点需要修复。 ● STARTING_STATE：节点正在启动中。 ● WAITING_STATE：节点正等待升级中。 ● DEMOTING_STATE：节点正在降级中。 ● PROMOTING_STATE：备节点正在升级为主节点的状态 ● BUILDING_STATE：备机启动失败, 需要重建。 ● CATCHUP_STATE：备节点正在追赶主节点。 ● COREDUMP_STATE：节点崩溃。
state	text	wal sender状态。 ● WALSNDSTATE_STARTUP：启动状态。 ● WALSNDSTATE_BACKUP：备份状态。 ● WALSNDSTATE_CATCHUP：追赶状态。 ● WALSNDSTATE_STREAMING：流复制状态。
sender_sent_location	text	主节点发送位置。
sender_write_location	text	主节点落盘位置。
sender_flush_location	text	主节点将缓存写入磁盘的位置。
sender_replay_location	text	主节点redo位置。
receiver_received_location	text	备节点接收位置。
receiver_write_location	text	备节点落盘位置。
receiver_flush_location	text	备节点将缓存写入磁盘的位置。
receiver_replay_location	text	备节点redo位置。

版权所有 © 华为云计算技术有限公司

7.3.13.78 GS_ALL_CONTROL_GROUP_INFO

GS_ALL_CONTROL_GROUP_INFO视图显示数据库内所有控制组的信息。

表 7-360 GS_ALL_CONTROL_GROUP_INFO 字段

名称	类型	描述
name	text	控制组的名称。
type	text	控制组的类型。 ● GROUP_NONE：无分组。 ● GROUP_TOP：顶级分组。 ● GROUP_CLASS：该资源的类分组，不控制任何线程。 ● GROUP_BAKWD：后端线程控制组。 ● GROUP_DEFWD： 默认控制组，仅控制该级别的查询线程。 ● GROUP_TSWD：每个用户的分时控制组，控制最底层的查询线程。
gid	bigint	控制组ID。
classgid	bigint	Workload所属Class的控制组ID。
class	text	Class控制组。
workload	text	Workload控制组。
shares	bigint	控制组分配的CPU资源配额。
limits	bigint	控制组分配的CPU资源限额。
wdlevel	bigint	Workload控制组层级。
cpucores	text	控制组使用的CPU核的信息。

7.3.13.79 GS_ALL_PREPARED_STATEMENTS

GS_ALL_PREPARED_STATEMENTS视图显示所有会话中可用的预备语句的信息。默认只有系统管理员权限才可以访问。

表 7-361 GS_ALL_PREPARED_STATEMENTS 字段

名称	类型	描述
pid	bigint	后台线程ID。 **说明** 线程池模式下pid显示的是当前会话绑定的线程ID，当会话在不同线程上执行时pid会随之改变。线程池模式下statement与sessionid相关联，与pid无关联，关联查询时建议使用sessionid。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sessionid	bigint	当前会话ID。
global_sessionid	text	全局会话ID。
name	text	预备语句的标识符。
statement	text	创建该预备语句的查询字符串。对于从SQL创建的预备语句而言是客户端提交的PREPARE语句；对于通过前/后端协议创建的预备语句而言是预备语句自身的文本。对于stream pbe功能产生的计划缓存，当前字段在CN记录的是上述内容，而在DN记录的是首次执行时的语句。
prepare_time	timestamp with time zone	创建该预备语句的时间戳。
parameter_types	regtype[]	该预备语句期望的参数类型，以regtype类型的数组格式出现。与该数组元素相对应的OID可以通过把regtype转换为OID值得到。对于stream pbe功能产生的计划缓存，当前字段在DN暂时为空。
from_sql	boolean	● 如果该预备语句是通过PREPARE语句创建的则为true。 ● 如果是通过前/后端协议创建的则为false。
is_deleted	boolean	表示计划缓存是否被清理，通过GUC参数max_plancache_memory_ratio设置清理阈值。 ● 如果被清理则为TRUE。 ● 如果未被清理则为FALSE。
size	bigint	表示计划缓存占用内存大小，单位为Byte。
valid	boolean	表示计划缓存的失效状态。 ● 如果有效则为TRUE。 ● 如果被失效则为FALSE。
is_reused	boolean	表示该缓存是否被重用。 ● 如果被重用则为TRUE。 ● 如果未被重用则为FALSE。

7.3.13.80 GS_AUTONOMOUS_TRANSACTION_DETAIL

GS_AUTONOMOUS_TRANSACTION_DETAIL视图用来显示当前CN节点所有会话中用户所属的自治事务会话、及其父子事务关系。若当前用户具备MONADMIN权限或者SYSADMIN权限，则可以查看当前CN节点所有自治事务会话及其父子事务关系。不允许在DN节点调用。

版权所有 © 华为云计算技术有限公司

表 7-362 GS_AUTONOMOUS_TRANSACTION_DETAIL 字段

名称	类型	描述
autonomous_transaction_session_id	bigint	自治事务会话ID。
autonomous_transaction_thread_id	bigint	自治事务线程ID。
parent_autonomous_transaction_session_id	bigint	自治事务父会话ID。

7.3.13.81 GS_BUCKET_LOCKS

GS_BUCKET_LOCKS视图存储各打开事务所持有的bucket锁信息。仅支持分布式下的CN查询。

表 7-363 GS_BUCKET_LOCKS 字段

名称	类型	引用	描述
node_name	text	-	节点名称。
database	oid	PG_DATABASE.oid	被锁定对象所在数据库的OID。
virtualtransaction	text	-	持有此锁或者在等待此锁的事务的虚拟ID。
pid	bigint	-	持有或者等待这个锁的服务器线程的逻辑ID。如果锁是被一个预备事务持有的，则为NULL。
sessionid	bigint	-	持有或者等待这个锁的会话ID。
mode	text	-	这个线程持有的或者是期望的锁模式。 可能取值为：AccessShareLock、AccessExclusiveLock。
granted	boolean	-	• 如果锁是持有锁，则为TRUE。 • 如果锁是等待锁，则为FALSE。
fastpath	boolean	-	如果通过fast-path获得锁，则为TRUE；如果通过库级别锁信息获得锁或者两阶段提交时的锁，则为FALSE。
global_sessionid	text	-	全局会话ID。
bucketlist	text	-	被锁定的bucket列表。
bucket_group	smallint	-	被锁定的bucket所在的bucket组，当前版本暂不支持，值恒为0。

版权所有 © 华为云计算技术有限公司

7.3.13.82 GS_GET_CONTROL_GROUP_INFO

GS_GET_CONTROL_GROUP_INFO视图显示所有控制组的信息。默认只有系统管理员权限才可以访问。

表 7-364 GS_GET_CONTROL_GROUP_INFO 字段

名称	类型	描述
group_name	text	控制组的名称。
group_type	text	控制组的类型。 ● GROUP_NONE：无分组。 ● GROUP_TOP：顶级分组。 ● GROUP_CLASS：该资源的类分组，不控制任何线程。 ● GROUP_BAKWD：后端线程控制组。 ● GROUP_DEFWD：默认控制组，仅控制该级别的查询线程。 ● GROUP_TSWD：每个用户的分时控制组，控制最底层的查询线程。
gid	bigint	控制组ID。
classgid	bigint	Workload所属Class的控制组ID。
class	text	Class控制组。
group_workload	text	Workload控制组。
shares	bigint	控制组分配的CPU资源配额。
limits	bigint	控制组分配的CPU资源限额。
wdlevel	bigint	Workload控制组层级。
cpucores	text	控制组使用的CPU核的信息。
nodegroup	text	node group的名称。
group_kind	text	node group的类型，取值包括： ● i：表示installation node group。 ● n：表示普通集群node group。 ● e：表示弹性集群。

版权所有 © 华为云计算技术有限公司

7.3.13.83 GS_GLC_MEMORY_DETAIL

GS_GLC_MEMORY_DETAIL视图显示所有数据库内global plpgsql cache全局缓存的内存使用情况，仅在enable_global_plsqlcache=on时可用。

表 7-365 GS_GLC_MEMORY_DETAIL 字段

名称	类型	描述
contextname	text	内存对象名称。
database	text	内存对象所属的数据库，"pkg_bucket" 与"func_bucket"显示为"NULL"。
schema	text	内存对象所属的模式，"pkg_bucket" 与"func_bucket"显示为"NULL"。
type	text	对象类型： ● "pkg_bucket"：代表该对象为package对象的父节点。 ● "func_bucket"：代表该对象为函数或存储过程的父节点。 ● "pkg"：代表该对象为package对象。 ● "func"：代表该对象为函数或存储过程。
status	text	缓存对象当前状态，可用显示为“valid”，不可用显示为'invalid'。"pkg_bucket"与"func_bucket"对象无状态，显示为"NULL"。
location	text	缓存对象当前位置，在缓存哈希表内显示为"in_global_hash_table"，在失效链表中显示为"in_global_expired_list"。 "pkg_bucket"与"func_bucket"对象显示为"NULL"。
env	bigint	创建对象时的环境参数，即behavior_compat_flags的值。"pkg_bucket" 与"func_bucket"对象显示为0。
usedsize	bigint	缓存对象大小。
usecount	bigint	有多少对象正在引用该全局缓存，当没有对象引用该全局编译产物时，数量为0。
oid	integer	内存对象oid。
isspec	boolean	内存对象是否是包头。
searchpath	text	内存对象访问编译产物的schema及其他环境变量。
invalidreason	text	内存对象的失效原因。

版权所有 © 华为云计算技术有限公司

7.3.13.84 GS_GLOBAL_ARCHIVE_STATUS

GS_GLOBAL_ARCHIVE_STATUS视图描述CN和所有分片的归档进度，获取分片名称（node_name）、归档位置（restart_lsn）、实际进行归档的主/备机名称（archive_node）和当前日志位置（current_xlog_location）。查询此视图需要数据库开启归档功能，并从CN节点进行查询，需要monitor admin和sysadmin权限。具体字段信息如表7-366所示。

表 7-366 GS_GLOBAL_ARCHIVE_STATUS 字段

名称	类型	描述
node_name	text	分片名称。
restart_lsn	text	归档位置。
archive_node	text	实际进行归档的主/备机名称。
current_xlog_location	text	当前日志位置。

7.3.13.85 GS_GSC_MEMORY_DETAIL

GS_GSC_MEMORY_DETAIL视图显示当前节点当前进程的全局SysCache的内存占用情况。仅在开启GSC的模式下有数据。具体字段如表7-367所示。

由于这个查询是以数据库内存上下文分隔的，因此会缺少一部分内存的统计，缺失的内存统计对应的内存上下文名称为GlobalSysDBCache。

表 7-367 GS_GSC_MEMORY_DETAIL 字段

名称	类型	描述
db_id	text	数据库id。
totalsize	numeric	共享内存总大小，单位Byte。
freesize	numeric	共享内存剩余大小，单位Byte。
usedsize	numeric	共享内存使用大小，单位Byte。

7.3.13.86 GS_GTT_ATTACHED_PIDS

GS_GTT_ATTACHED_PIDS视图可用来在DN上查看哪些会话正在使用全局临时表，调用gs_gtt_attached_pid()函数。

表 7-368 GS_GTT_ATTACHED_PIDS 字段

名称	类型	描述
schemaname	name	schema名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tablename	name	全局临时表名称。
relid	oid	全局临时表的OID。
pids	bigint[]	会话ID。
cn_session_ids	text[]	分布式下CN的全局会话ID列表。

7.3.13.87 GS_GLOBAL_AUTONOMOUS_TRANSACTION_DETAIL

GS_GLOBAL_AUTONOMOUS_TRANSACTION_DETAIL视图用来显示每个CN节点所有会话中用户所属的自治事务会话、及其父子事务关系。若当前用户具备MONADMIN权限或者SYSADMIN权限，则可以查看每个CN节点所有自治事务会话及其父子事务关系。不允许在DN节点调用。

说明

当GUC参数enable_nonsysadmin_execute_direct=off时，该视图仅允许具备MONADMIN权限或者SYSADMIN权限的用户调用。

表 7-369 GS_GLOBAL_AUTONOMOUS_TRANSACTION_DETAIL 字段

名称	类型	描述
node_name	text	CN节点名称。
autonomous_transaction_session_id	bigint	自治事务会话ID。
autonomous_transaction_thread_id	bigint	自治事务线程ID。
parent_autonomous_transaction_session_id	bigint	自治事务父会话ID。

7.3.13.88 GS_GLOBAL_GTT_ATTACHED_PIDS

GS_GLOBAL_GTT_ATTACHED_PIDS视图用来在CN上查看所有DN上哪些会话正在使用全局临时表。

表 7-370 GS_GLOBAL_GTT_ATTACHED_PIDS 字段

名称	类型	描述
node_name	text	节点名称。
schemaname	name	schema名称。
tablename	name	全局临时表名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relid	oid	全局临时表的OID。
pids	bigint[]	会话ID。
cn_session_ids	text[]	分布式下CN的全局会话ID列表。

7.3.13.89 GS_GLOBAL_DIST_GTT_REMAIN_SESSIONS

GS_GLOBAL_DIST_GTT_REMAIN_SESSIONS视图可用来在CN上查看所有残留在DN上的会话信息。

表 7-371 GS_GTT_ATTACHED_PIDS 字段

名称	类型	描述
dn_name	text	DN的名称。
cn_name	text	CN的名称。
cn_timeline	integer	CN的时间线。
cn_sequence	integer	CN的会话序列。
cn_session_id	bigint	CN的会话ID。

7.3.13.90 GS_LOGICALSTANDBY_SKIP

GS_LOGICALSTANDBY_SKIP视图展示定义的SKIP规则，当skiptype为1或者3时（具体请参见**GS_SQLAPPLY_SKIP**系统表字段），从系统表GS_SQLAPPLY_SKIP中获取如下字段创建或更新系统视图。分布式场景暂无对应功能，视图为空，当前仅作预留。

表 7-372 GS_LOGICALSTANDBY_SKIP 字段

名称	类型	说明
createtime	timestamp with time zone	SKIP定义时间。
error	text	是否错误发生才触发。
dump	"char"	转储类型。
stmt	text	SKIP语句或KEYWORD。
dbname	text	数据库名称。
schemaname	text	模式名称。
objectname	text	对象名称。

版权所有 © 华为云计算技术有限公司

7.3.13.91 GS_LOGICALSTANDBY_SKIP_TRANSACTION

GS_LOGICALSTANDBY_SKIP_TRANSACTION视图展示定义的SKIP规则，当skiptype为2时（具体请参见GS_SQLAPPLY_SKIP系统表字段），从系统表GS_SQLAPPLY_SKIP中获取如下字段创建或更新系统视图。分布式场景暂无对应功能，视图为空，当前仅作预留。

表 7-373 GS_LOGICALSTANDBY_SKIP_TRANSACTION 字段

名称	类型	属性
createtime	timestamp with time zone	SKIP定义时间。
dump	"char"	转储类型。
commitcsn	text	事务提交序号。
xid	text	事务ID。

7.3.13.92 GS_LSC_MEMORY_DETAIL

GS_LSC_MEMORY_DETAIL视图显示所有线程的本地SysCache的内存占用情况，以MemoryContext节点来统计。仅在开启GSC的模式下有数据。具体字段如表7-374所示。

表 7-374 GS_LSC_MEMORY_DETAIL 字段

名称	类型	描述
threadid	text	线程启动时间+线程标识（字符串信息为timestamp.sessionid）。
tid	bigint	线程标识。
thrdtype	text	线程类型。可以是系统内存在的任何线程类型，如postgresql、wlmmonitor等。
contextname	text	内存上下文名称。
level	smallint	当前上下文在整体内存上下文中的层级。
parent	text	父内存上下文名称。
totalsize	bigint	当前内存上下文的内存总数，单位Byte。
freesize	bigint	当前内存上下文中已释放的内存总数，单位Byte。
usedsize	bigint	当前内存上下文中已使用的内存总数，单位Byte。

版权所有 © 华为云计算技术有限公司

7.3.13.93 GS_MY_PLAN_TRACE

GS_MY_PLAN_TRACE是系统表GS_PLAN_TRACE的视图，该视图主要用来查看当前用户的plan trace。具体字段信息如表7-375所示。分布式场景下不支持plan trace特性，所以在分布式下该视图中无数据。

表 7-375 GS_MY_PLAN_TRACE 字段

名称	类型	描述
query_id	text	当前请求的唯一id。
query	text	当前请求的sql语句，该字段大小不会超过系统参数track_activity_query_size指定的大小。
unique_sql_id	bigint	当前请求sql的唯一id。
plan	text	当前请求sql对应的查询计划文本。该字段大小不会超过10K。
plan_trace	text	当前请求sql对应的查询计划生成过程的明细，该字段大小不会超过300M。
modifydate	timestamp with time zone	当前plan trace的更新时间（当前指的是plan trace创建时间）。

7.3.13.94 GS_SESSION_ALL_SETTINGS

GS_SESSION_ALL_SETTINGS显示本节点上所有session的全量GUC参数配置。该视图只有sysadmin和monadmin权限可以查看。具体字段信息如表7-376所示。

表 7-376 GS_SESSION_ALL_SETTINGS 字段

名称	类型	描述
sessionid	bigint	会话的id。
pid	bigint	后端线程的id。
name	text	参数名称。
setting	text	参数当前值。
unit	text	参数的隐式单位。

7.3.13.95 GS_SQL_COUNT

GS_SQL_COUNT视图显示数据库当前节点当前时刻执行的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）统计信息。具体字段信息如表7-377所示。

版权所有 © 华为云计算技术有限公司

- 普通用户查询GS_SQL_COUNT视图仅能看到该用户当前节点的统计信息；管理员权限用户查询GS_SQL_COUNT视图则能看到所有用户当前节点的统计信息。
- 当集群或该节点重启时，计数将清零，并重新开始计数。
- 计数以节点收到的查询数为准，包括集群内部进行的查询。例如，CN收到一条查询，若下发多条查询DN，那将在DN上进行相应次数的计数。

表 7-377 GS_SQL_COUNT 字段

名称	类型	描述
node_name	name	节点名称。
user_name	name	用户名。
select_count	bigint	SELECT语句统计结果。
update_count	bigint	UPDATE语句统计结果。
insert_count	bigint	INSERT语句统计结果。
delete_count	bigint	DELETE语句统计结果。
mergeinto_count	bigint	MERGE INTO语句统计结果。
ddl_count	bigint	DDL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。
dml_count	bigint	DML语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dcl_count	bigint	DCL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。
total_select_elapse	bigint	总SELECT的时间花费（单位：微秒）。
avg_select_elapse	bigint	平均SELECT的时间花费（单位：微秒）。
max_select_elapse	bigint	最大SELECT的时间花费（单位：微秒）。
min_select_elapse	bigint	最小SELECT的时间花费（单位：微秒）。
total_update_elapse	bigint	总UPDATE的时间花费（单位：微秒）。
avg_update_elapse	bigint	平均UPDATE的时间花费（单位：微秒）。
max_update_elapse	bigint	最大UPDATE的时间花费（单位：微秒）。
min_update_elapse	bigint	最小UPDATE的时间花费（单位：微秒）。
total_insert_elapse	bigint	总INSERT的时间花费（单位：微秒）。
avg_insert_elapse	bigint	平均INSERT的时间花费（单位：微秒）。
max_insert_elapse	bigint	最大INSERT的时间花费（单位：微秒）。
min_insert_elapse	bigint	最小INSERT的时间花费（单位：微秒）。
total_delete_elapse	bigint	总DELETE的时间花费（单位：微秒）。
avg_delete_elapse	bigint	平均DELETE的时间花费（单位：微秒）。
max_delete_elapse	bigint	最大DELETE的时间花费（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
min_delete_elapse	bigint	最小DELETE的时间花费（单位：微秒）。
dbid	oid	统计的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）所属的数据库id。
user_dml_count	bigint	用户执行的DML语句的数量。 user_dml_count的数量等于dml_count的数量减去bg_dml_count的数量。
bg_dml_count	bigint	数据库后台线程执行的DML语句的数量，包含全场景下后台线程（WORKER和THREADPOOL_WORKER以外的线程）执行的DML语句。 **说明** 当前存储过程中的命令不支持统计DML语句。

7.3.13.96 GS_SYSTEM_OBJECT_VERSION_DETAILS

GS_SYSTEM_OBJECT_VERSION_DETAILS视图记录了系统对象的版本号信息，以及对应系统对象的详细信息。

此系统视图中包含的系统对象类型包含视图、函数、表、索引、类型、规则及类型转换关系CAST。

表 7-378 GS_SYSTEM_OBJECTS_VERSION 字段

名称	类型	描述
object_oid	oid	系统对象的标识符，不同类型系统对象的对应关系如下： 视图、表、索引在**PG_CLASS**内的唯一标识，类型、函数、规则、类型转换关系CAST分别在**PG_TYPE**、**PG_PROC**、**PG_REWRITE**、**PG_CAST**内的唯一标识
object_namespace	oid	系统对象Schema的oid。 系统对象为类型转换关系CAST时，该字段为空。
object_name	character varying	系统对象的名称。 系统对象为类型转换关系CAST时，该字段的格式为“源数据类型(源数据类型OID) AS 目标数据类型(目标数据类型OID)”。
object_schema	name	系统对象Schema的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
object_args	text	系统对象为函数时，该字段为函数对象参数的数据类型数组，只包括输入参数（包括INOUT参数），显示格式为“[arg1, ..., argn]”。 系统对象为其他类型时，该字段为空。
object_type	text	系统对象的类型。 ● view：标识视图。 ● function：标识函数。 ● table：标识普通表。 ● index：标识索引。 ● composite type：标识复合类型。 ● domain：标识域类型。 ● rule：标识规则。 ● cast：标识类型转换。
object_version	bigint	系统对象的版本号。 0：初始版本号。 -1：已删除对象的版本号。 其他版本号为后续对象变更的版本号。

7.3.13.97 GS_THRDSTREAM_WORKER_INFO

GS_THRDSTREAM_WORKER_INFO视图显示stream线程池复用相关信息。仅支持在以线程池模式运行的DN上查询。具有PUBLIC权限的用户才可以访问该视图。

表 7-379 GS_THRDSTREAM_WORKER_INFO 字段

名称	类型	描述
node_name	text	实例名。
group_id	integer	线程池group组id。
db_info	text	描述当前group组stream线程池运行中的动态的DATABASE统计信息。包括如下信息： ● total stream reusing db：stream线程池当前使用或复用中的DATABASE数量。 ● free session db：stream线程池当前暂未运行session的DATABASE数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stream_info	text	描述当前group组stream线程池运行中的动态的stream线程统计信息。包括如下信息： • actual：实际启动的stream线程数。 • idle：空闲的stream线程数。 • using：使用中的stream线程数。 • init session threads：初始化session的stream线程数。 • reuse session threads：复用session的stream线程数。

7.3.13.98 GS_TOTAL_NODEGROUP_MEMORY_DETAIL

GS_TOTAL_NODEGROUP_MEMORY_DETAIL返回当前数据库node group使用内存的信息，单位为MB，若GUC参数enable_memory_limit设置为off，则该视图不能使用。具体字段信息如**表7-380**所示。

表 7-380 GS_TOTAL_NODEGROUP_MEMORY_DETAIL 字段

名称	类型	描述
ngname	text	node group名称。
memorytype	text	内存类型，包括以下几种： • ng_total_memory：该node group的总内存大小。 • ng_used_memory：该node group的实际使用内存大小。 • ng_estimate_memory：该node group的估算使用内存大小。 • ng_foreignrp_memsize：该node group的外部资源池的总内存大小。 • ng_foreignrp_usedsize：该node group的外部资源池实际使用内存大小。 • ng_foreignrp_peaksize：该node group的外部资源池使用内存的峰值。 • ng_foreignrp_mempct：该node group的外部资源池占该node_group总内存大小的百分比。 • ng_foreignrp_estmsize：该node group的外部资源池估算使用内存大小。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memorymbytes	integer	内存类型分配内存的大小。

7.3.13.99 GS_WORKLOAD_RULE_STAT

GS_WORKLOAD_RULE_STAT系统视图记录SQL限流规则相关的信息。只有具有sysadmin权限的用户才可以访问此系统视图。具体字段信息如**表7-381**所示。

表 7-381 GS_WORKLOAD_RULE_STAT 字段

名称	类型	描述
rule_id	bigint	限流规则id。
rule_name	name	限流规则的名称，用于检索限流规则。
databases	name[]	限流规则作用的数据库列表，为NULL表示对所有库生效。
rule_type	text	限流规则类型，当前仅支持：sqlid、select、insert、update、delete、merge和resource、client，其他取值为非法值。
start_time	timestamp with time zone	限流规则开始的时间，为NULL表示从现在开始生效。
end_time	timestamp with time zone	限流规则结束的时间，为NULL表示一直生效。
max_workload	bigint	限制规则设置的最大并发数。
option_val	text[]	限流规则的参数值，包括：sqlid、关键字列表、资源限制情况、客户端IP、ApplicationName。 详细请参见**•gs_add_workload_rule(rule_type, rule_name, databases, start_time, end_time, max_workload, option_val)**接口说明。
is_valid	boolean	限流规则是否生效，超时的限流规则会设为false。
validate_count	bigint	限流规则拦截SQL的次数。
node_names	text[]	预留字段，限流规则生效的节点名称列表，当前不生效。
user_names	text[]	预留字段，限流规则生效的用户名称列表，当前不生效。

版权所有 © 华为云计算技术有限公司

7.3.13.100 GV_INSTANCE

GV_INSTANCE视图显示当前数据库实例的信息。默认只有系统管理员权限才可以访问，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-382所示。

表 7-382 GV_INSTANCE 字段

名称	类型	描述
inst_id	oid	当前数据库oid。
instance_number	oid	当前数据库oid。
instance_name	character varying(16)	当前数据库名。
host_name	character varying(64)	暂不支持，值为NULL。
version	character varying(17)	暂不支持，值为NULL。
version_legacy	character varying(17)	暂不支持，值为NULL。
version_full	character varying(17)	暂不支持，值为NULL。
startup_time	date	暂不支持，值为NULL。
status	character varying(12)	暂不支持，值为NULL。
parallel	character varying(3)	暂不支持，值为NULL。
thread#	numeric	暂不支持，值为NULL。
archiver	character varying(7)	暂不支持，值为NULL。
log_switch_wait	character varying(15)	暂不支持，值为NULL。
logins	character varying(10)	暂不支持，值为NULL。
shutdown_pending	character varying(3)	暂不支持，值为NULL。
database_status	character varying(17)	暂不支持，值为NULL。
instance_role	character varying(18)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
active_state	character varying(9)	暂不支持，值为NULL。
blocked	character varying(3)	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。
instance_mode	character varying(11)	暂不支持，值为NULL。
edition	character varying(7)	暂不支持，值为NULL。
family	character varying(80)	暂不支持，值为NULL。
database_type	character varying(15)	暂不支持，值为NULL。

7.3.13.101 GV_SESSION

GV_SESSION视图显示当前所有的会话信息。该视图只有管理员可以访问，普通用户需要授权才能访问，该视图同时存在于PG_CATALOG和SYS Schema下。在开启线程池（enable_thread_pool = on）时，显示所有会话信息；当线程池关闭（enable_thread_pool = off）时，用户连接的会话不显示。具体字段信息如**表7-383**所示。

表 7-383 GV_SESSION 字段

名称	类型	描述
inst_id	numeric	暂不支持，值为NULL。
saddr	raw	暂不支持，值为NULL。
sid	bigint	会话ID。
serial#	integer	当前活动的后台线程的序号，在GaussDB中为0。
audsid	numeric	暂不支持，值为NULL。
paddr	raw	暂不支持，值为NULL。
schema#	numeric	暂不支持，值为NULL。
schemaname	name	登录该后台的用户名。
user#	oid	登录此后台线程的用户的OID。oid为0表示此后台线程为全局辅助线程(auxiliary)。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
username	name	登录此后台线程的用户名。username为空表示此后台线程为全局辅助线程(auxiliary)。
command	numeric	暂不支持，值为NULL。
ownerid	numeric	暂不支持，值为NULL。
taddr	character varying(16)	暂不支持，值为NULL。
lockwait	character varying(16)	暂不支持，值为NULL。
machine	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
sql_id	bigint	查询语句的ID。
client_info	text	客户端信息。
event	text	语句当前排队状态。可能值是： ● waiting in queue：表示语句在排队中。 ● 空：表示语句正在运行。
sql_exec_start	timestamp with time zone	开始当前活跃查询的时间， 如果state的值不是active，则这个值是上一个查询的开始时间。
program	text	连接到该后台的应用名。
status	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。
server	character varying(9)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pdml_status	character varying(8)	当前会话是否启用DML的并行执行。
port	numeric	当前会话的端口号。
process	character varying(24)	当前会话的进程号。
logon_time	date	当前会话的登录时间。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
last_call_et	integer	当前会话上次状态发生改变的时长。
osuser	text	服务端操作系统用户名。
terminal	character varying(30)	暂不支持，值为NULL。
type	character varying(10)	暂不支持，值为NULL。
sql_address	raw	暂不支持，值为NULL。
sql_hash_value	numeric	暂不支持，值为NULL。
sql_child_number	numeric	暂不支持，值为NULL。
sql_exec_id	numeric	暂不支持，值为NULL。
prev_sql_addr	raw	暂不支持，值为NULL。
prev_hash_value	numeric	暂不支持，值为NULL。
prev_sql_id	character varying(13)	暂不支持，值为NULL。
prev_child_number	numeric	暂不支持，值为NULL。
prev_exec_start	date	暂不支持，值为NULL。 • O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
prev_exec_id	numeric	暂不支持，值为NULL。
plsql_entry_object_id	numeric	暂不支持，值为NULL。
plsql_entry_subprogram_id	numeric	暂不支持，值为NULL。
plsql_object_id	numeric	暂不支持，值为NULL。
plsql_subprogram_id	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
module	text	当前正在运行的模块名称，通过DBE_APPLICATION_INFO.SET_MODULE接口设置。
module_hash	numeric	暂不支持，值为NULL。
action	text	当前模块中当前操作的名称DBE_APPLICATION_INFO.SET_MODULE或DBE_APPLICATION_INFO.SET_ACTION接口设置。
action_hash	numeric	暂不支持，值为NULL。
fixed_table_sequence	numeric	暂不支持，值为NULL。
row_wait_obj#	numeric	暂不支持，值为NULL。
row_wait_file#	numeric	暂不支持，值为NULL。
row_wait_block#	numeric	暂不支持，值为NULL。
row_wait_row#	numeric	暂不支持，值为NULL。
top_level_call#	numeric	暂不支持，值为NULL。
pdml_enabled	character varying(3)	暂不支持，值为NULL。
failover_type	character varying(13)	暂不支持，值为NULL。
failover_method	character varying(10)	暂不支持，值为NULL。
failed_over	character varying(3)	暂不支持，值为NULL。
resource_consumer_group	character varying(32)	暂不支持，值为NULL。
pddl_status	character varying(8)	暂不支持，值为NULL。
pq_status	character varying(8)	暂不支持，值为NULL。
current_queue_duration	numeric	暂不支持，值为NULL。
client_identifier	character varying(64)	暂不支持，值为NULL。
blocking_session_status	character varying(11)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
blocking_instance	numeric	暂不支持，值为NULL。
blocking_session	numeric	暂不支持，值为NULL。
final_blocking_session_status	character varying(11)	暂不支持，值为NULL。
final_blocking_instance	numeric	暂不支持，值为NULL。
final_blocking_session	numeric	暂不支持，值为NULL。
seq#	numeric	暂不支持，值为NULL。
event#	numeric	暂不支持，值为NULL。
p1text	character varying(64)	暂不支持，值为NULL。
p1	numeric	暂不支持，值为NULL。
p1raw	raw	暂不支持，值为NULL。
p2text	character varying(64)	暂不支持，值为NULL。
p2	numeric	暂不支持，值为NULL。
p2raw	raw	暂不支持，值为NULL。
p3text	character varying(64)	暂不支持，值为NULL。
p3	numeric	暂不支持，值为NULL。
p3raw	raw	暂不支持，值为NULL。
wait_class_id	numeric	暂不支持，值为NULL。
wait_class#	numeric	暂不支持，值为NULL。
wait_class	character varying(64)	暂不支持，值为NULL。
wait_time	numeric	暂不支持，值为NULL。
seconds_in_wait	numeric	暂不支持，值为NULL。
state	character varying(19)	暂不支持，值为NULL。
wait_time_micro	numeric	暂不支持，值为NULL。
time_remaining_micro	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
time_since_last_wait_micro	numeric	暂不支持，值为NULL。
service_name	character varying(64)	暂不支持，值为NULL。
sql_trace	character varying(8)	暂不支持，值为NULL。
sql_trace_waits	character varying(5)	暂不支持，值为NULL。
sql_trace_binds	character varying(5)	暂不支持，值为NULL。
sql_trace_plan_stats	character varying(10)	暂不支持，值为NULL。
session_edition_id	numeric	暂不支持，值为NULL。
creator_addr	raw	暂不支持，值为NULL。
creator_serial#	numeric	暂不支持，值为NULL。
ecid	character varying(64)	暂不支持，值为NULL。
sql_translation_profile_id	numeric	暂不支持，值为NULL。
pga_tunable_mem	numeric	暂不支持，值为NULL。
shard_ddl_status	character varying(8)	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。
external_name	character varying(1024)	暂不支持，值为NULL。
plsql_debugger_connected	character varying(5)	暂不支持，值为NULL。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的logon_time字段类型会变更为datea。

7.3.13.102 MPP_TABLES

MPP_TABLES视图显示PGXC_CLASS中表的信息。具体字段信息如表7-384所示。

版权所有 © 华为云计算技术有限公司

表 7-384 MPP_TABLES 字段

名称	类型	描述
schemaname	name	表的模式名。
tablename	name	表名。
tableowner	name	表的所有者。
tablespace	name	表所在的表空间。
pgroup	name	节点群的名称。
nodeoids	oidvector_extend	表分布的节点OID列表。

7.3.13.103 MY_AUDIT_TRAIL

MY_AUDIT_TRAIL视图显示与当前用户有关的标准审计跟踪条目，GaussDB审计信息主要通过gs_query_audit()函数。该视图同时存在于PG_CATALOG和SYS Schema下，仅拥有AUDITADMIN属性及MY_AUDIT_TRAIL的SELECT权限的用户才可以查看审计信息；三权分立未开启时，拥有SYSADMIN属性的用户也可以查看审计信息。GaussDB的action_name字段与ORA数据库审计动作不一致，transactionid字段与ORA数据库中transactionid数据的类型保持一致，GaussDB的sql_text字段为解析后SQL描述语句，不完全与执行的SQL语句相同。具体字段信息如表7-385所示。

⚠ 注意

在分布式下，gs_query_audit()函数只能查询当前节点的审计信息。

表 7-385 MY_AUDIT_TRAIL 字段

名称	类型	描述
os_username	character varying(255)	暂不支持，值为NULL。
username	character varying(128)	操作被审计的用户的名称，不是用户id。
userhost	character varying(128)	暂不支持，值为NULL。
terminal	character varying(255)	暂不支持，值为NULL。
timestamp	date	在本地数据库会话时区中创建审计跟踪条目的日期和时间（由审计会话创建的条目的用户登录日期和时间）。
owner	character varying(128)	受操作影响的对象的创建者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
obj_name	character varying(128)	受操作影响的对象的名称。
action	numeric	暂不支持，值为NULL。
action_name	character varying(28)	action列中的数字代码对应的action类型名称。
new_owner	character varying(128)	暂不支持，值为NULL。
new_name	character varying(128)	暂不支持，值为NULL。
obj_privilege	character varying(32)	暂不支持，值为NULL。
sys_privilege	character varying(40)	暂不支持，值为NULL。
admin_option	character varying(1)	暂不支持，值为NULL。
grantee	character varying(128)	暂不支持，值为NULL。
audit_option	character varying(40)	暂不支持，值为NULL。
ses_actions	character varying(19)	暂不支持，值为NULL。
logoff_time	date	暂不支持，值为NULL。
logoff_lread	numeric	暂不支持，值为NULL。
logoff_pread	numeric	暂不支持，值为NULL。
logoff_lwrite	numeric	暂不支持，值为NULL。
logoff_dlock	character varying(40)	暂不支持，值为NULL。
comment_text	character varying(4000)	暂不支持，值为NULL。
sessionid	numeric	暂不支持，值为NULL。
entryid	numeric	暂不支持，值为NULL。
statementid	numeric	暂不支持，值为NULL。
returncode	numeric	暂不支持，值为NULL。
priv_used	character varying(40)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
client_id	character varying(128)	暂不支持，值为NULL。
econtext_id	character varying(64)	暂不支持，值为NULL。
session_cpu	numeric	暂不支持，值为NULL。
extended_timestamp	timestamp(6) with time zone	创建审计跟踪条目的时间戳（由创建的条目的用户登录时间戳UTC（协调通用）中的审计会话时间）时区。
proxy_sessionid	numeric	暂不支持，值为NULL。
global_uid	character varying(32)	暂不支持，值为NULL。
instance_number	numeric	暂不支持，值为NULL。
os_process	character varying(16)	暂不支持，值为NULL。
transactionid	text	访问或修改对象的事务的事务标识符。
scn	numeric	暂不支持，值为NULL。
sql_bind	nvarchar2(2000)	暂不支持，值为NULL。
sql_text	nvarchar2	查询的SQL文本。
obj_edition_name	character varying(128)	暂不支持，值为NULL。
dbid	numeric	暂不支持，值为NULL。
rls_info	clob	暂不支持，值为NULL。
current_user	character varying(128)	暂不支持，值为NULL。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的timestamp字段类型会变更为datea。

7.3.13.104 MY_COL_COMMENTS

MY_COL_COMMENTS视图显示当前用户下表的列注释信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-386**所示。

版权所有 © 华为云计算技术有限公司

表 7-386 MY_COL_COMMENTS 字段

名称	类型	描述
owner	character varying(128)	表的所有者。
table_name	character varying(128)	表的名称。
column_name	character varying(128)	列名称。
comments	text	注释。
origin_con_id	numeric	暂不支持，值为0。
schema	character varying(64)	列所属的名称空间的名称。

7.3.13.105 MY_COLL_TYPES

MY_COLL_TYPES视图显示当前用户创建的集合类型信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-387**所示。

表 7-387 MY_COLL_TYPES 字段

名称	类型	描述
owner	character varying(128)	集合的所有者。
type_name	character varying(128)	集合的名称。
coll_type	character varying(128)	集合的描述。
upper_bound	numeric	暂不支持，值为NULL。
elem_type_mod	character varying(7)	暂不支持，值为NULL。
elem_type_owner	character varying(128)	集合基于的元素类型的所有者。该值主要用于用户定义的类型。
elem_type_name	character varying(128)	集合所依据的数据类型或用户定义类型的名称。
length	numeric	暂不支持，值为NULL。
precision	numeric	暂不支持，值为NULL。
scale	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
character_set_name	character varying(44)	暂不支持，值为NULL。
elem_storage	character varying(7)	暂不支持，值为NULL。
nulls_stored	character varying(3)	暂不支持，值为NULL。
char_used	character varying(1)	暂不支持，值为NULL。

7.3.13.106 MY_CONS_COLUMNS

MY_CONS_COLUMNS视图显示当前用户下表中主键约束列的信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-388所示。

表 7-388 MY_CONS_COLUMNS 字段

名称	类型	描述
table_name	character varying(64)	约束相关的表名。
column_name	character varying(64)	约束相关的列名。
constraint_name	character varying(64)	约束名。
owner	character varying(64)	约束创建者。
position	smallint	表中列的位置。

7.3.13.107 MY_CONSTRAINTS

MY_CONSTRAINTS视图显示当前用户下表中约束的信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-389所示。

表 7-389 MY_CONSTRAINTS 字段

名称	类型	描述
constraint_name	character varying(64)	约束名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
constraint_type	text	约束类型： • c：表示检查约束。 • f：表示外键约束。 • p：表示主键约束。 • u：表示唯一约束。
table_name	character varying(64)	约束相关的表名。
index_owner	character varying(64)	约束相关的索引的所有者（只针对唯一约束和主键约束）。
index_name	character varying(64)	约束相关的索引的名称（只针对唯一约束和主键约束）。
owner	character varying(64)	约束创建者。

7.3.13.108 MY_DEPENDENCIES

MY_DEPENDENCIES显示用户拥有对象对其他对象的依赖关系。所有用户都可以访问，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-390**所示。

表 7-390 MY_DEPENDENCIES 字段

名称	类型	描述
name	name	对象的名称。
type	character varying(18)	对象类型。
referenced_owner	name	被引用对象的所有者。
referenced_name	name	被引用对象的名称。
referenced_type	character varying(18)	被引用对象的类型。
referenced_link_name	character varying(128)	指向父对象的链接的名称（如果是远程）。
dependency_type	character varying(4)	指示依赖关系是否为REF依赖关系，默认取值为HARD，表示硬引用。

版权所有 © 华为云计算技术有限公司

7.3.13.109 MY_IND_COLUMNS

MY_IND_COLUMNS视图显示当前用户下所有索引的字段信息。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-391所示。

表 7-391 MY_IND_COLUMNS 字段

名称	类型	描述
index_owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名。
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
column_name	name	列名。
column_position	smallint	索引中列的位置。
column_length	numeric	列的长度，如果列是变长类型，该字段取值为NULL。
char_length	numeric	列的最大字节长度。
descend	character varying(4)	表示列是按降序（DESC）还是升序（ASC）排序。
collated_column_id	numeric	暂不支持，值为NULL。

7.3.13.110 MY_IND_EXPRESSIONS

MY_IND_EXPRESSIONS视图显示当前用户下基于函数的表达式索引的信息。所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-392所示。

表 7-392 MY_IND_EXPRESSIONS 字段

名称	类型	描述
table_owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名。
index_owner	character varying(64)	索引的所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
index_name	character varying(64)	索引名。
column_expression	text	定义列的基于函数的索引表达式。
column_position	smallint	索引中列的位置。

7.3.13.111 MY_INDEXES

MY_INDEXES视图显示当前用户的索引信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-393所示。

表 7-393 MY_INDEXES 字段

名称	类型	描述
owner	character varying(64)	索引的所有者。
index_name	character varying(64)	索引名称。
table_name	character varying(64)	索引对应的表名。
uniqueness	text	表示该索引是否为唯一索引。 • UNIQUE：唯一索引。 • NONUNIQUE：非唯一索引。
partitioned	character(3)	表示该索引是否具有分区表的性质。 • Yes：索引具有分区表的性质。 • No：索引不具有分区表的性质。
generated	character varying(1)	表示该索引的名称是否为系统生成。 • y：索引名称为系统生成。 • n：索引名称非系统生成。
index_type	character varying(27)	索引类型。 • NORMAL：索引属性都是简单的引用，表达式树为空。 • FUNCTION-BASED NORMAL：存在表达式树用于非简单字段引用的索引属性。
table_owner	character varying(128)	索引对象的所有者。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
table_type	character(11)	索引对象的类型。 ● TABLE：索引对象为表类型。
tablespace_name	character varying(30)	包含索引的表空间名称。
status	character varying(8)	非分区索引状态。 ● VALID：非分区索引可以用于查询。 ● UNUSABLE：非分区索引不可用。 ● N/A：索引具有分区表性质。
compression	character varying(13)	暂不支持，值为NULL。
prefix_length	numeric	暂不支持，值为NULL。
ini_trans	numeric	暂不支持，值为NULL。
max_trans	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
pct_threshold	numeric	暂不支持，值为NULL。
include_column	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
pct_free	numeric	暂不支持，值为NULL。
logging	character varying(3)	暂不支持，值为NULL。
blevel	numeric	暂不支持，值为NULL。
leaf_blocks	numeric	暂不支持，值为NULL。
distinct_keys	numeric	暂不支持，值为NULL。
avg_leaf_blocks_per_key	numeric	暂不支持，值为NULL。
avg_data_blocks_per_key	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
clustering_factor	numeric	暂不支持，值为NULL。
num_rows	numeric	暂不支持，值为NULL。
sample_size	numeric	暂不支持，值为NULL。
last_analyzed	date	暂不支持，值为NULL。
degree	character varying(40)	暂不支持，值为NULL。
instances	character varying(40)	暂不支持，值为NULL。
temporary	character varying(1)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
buffer_pool	character varying(7)	暂不支持，值为NULL。
flash_cache	character varying(7)	暂不支持，值为NULL。
cell_flash_cache	character varying(7)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
duration	character varying(15)	暂不支持，值为NULL。
pct_direct_access	numeric	暂不支持，值为NULL。
ityp_owner	character varying(128)	暂不支持，值为NULL。
ityp_name	character varying(128)	暂不支持，值为NULL。
parameters	character varying(1000)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
domidx_status	character varying(12)	暂不支持，值为NULL。
domidx_opstatus	character varying(6)	暂不支持，值为NULL。
funcidx_status	character varying(8)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
join_index	character varying(3)	暂不支持，值为NULL。
iot_redundant_pkey_elim	character varying(3)	暂不支持，值为NULL。
dropped	character varying(3)	暂不支持，值为NULL。
visibility	character varying(9)	表示该索引对优化器是否可见。 ● VISIBLE：索引对优化器可见。 ● INVISIBLE：索引对优化器不可见。
domidx_management	character varying(14)	暂不支持，值为NULL。
segment_created	character varying(3)	暂不支持，值为NULL。
orphaned_entries	character varying(3)	暂不支持，值为NULL。
indexing	character varying(7)	暂不支持，值为NULL。
auto	character varying(3)	暂不支持，值为NULL。

7.3.13.112 MY_JOBS

MY_JOBS视图显示当前用户拥有的定时任务的详细信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-394所示。

表 7-394 MY_JOBS 字段

名称	类型	描述
job	bigint	作业id。
log_user	name	创建者的UserName。
priv_user	name	作业执行者的UserName。
dbname	name	创建作业的数据库名称。
schema_user	name	定时任务的默认模式名。
start_date	timestamp without time zone	作业第一次开始执行的时间。
start_suc	text	作业第一次成功执行对应的开始时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_date	timestamp without time zone	上次运行的开始时间。
last_suc	text	上次成功运行的开始时间。
last_sec	text	上次成功运行的开始时间，提供兼容性支持。
this_date	timestamp without time zone	正在运行任务的开始时间。
this_suc	text	正在运行任务的开始时间。
this_sec	text	正在运行任务的开始时间，提供兼容性支持。
next_date	timestamp without time zone	任务下次执行时间。
next_suc	text	任务下次执行时间。
next_sec	text	任务下次执行时间，提供兼容性支持。
total_time	numeric	任务最近一次的执行时长。
broken	text	如果status字段取值为d，broken则为y，否则broken为n。
status	"char"	本步骤的执行状态，取值范围：（r、s、f、d），默认为r，取值含义： ● r：运行中。 ● s：执行成功。 ● f：执行失败。 ● d：取消执行。
interval	text	用来计算下次运行时间的时间表达式，如果为null则表示定时任务只执行一次。
failures	smallint	失败计数，若作业连续执行失败16次，则不再继续执行。
what	text	可执行的作业。
nls_env	character varying(4000)	暂不支持，值为NULL。
misc_env	raw	暂不支持，值为NULL。
instance	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

7.3.13.113 MY_OBJECTS

MY_OBJECTS视图描述了当前用户拥有的数据库对象。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-395所示。

表 7-395 MY_OBJECTS 字段

名称	类型	描述
object_name	name	对象的名称。
object_id	oid	对象的OID。
object_type	name	对象的类型，包括table、index、sequence、view、global secondary index等。
namespace	oid	对象所属的名称空间。
temporary	character(1)	对象是否为临时对象。
status	character varying(7)	对象的状态。
subobject_name	name	对象的子对象名称。
generated	character(1)	对象名称是否是系统生成。
created	timestamp with time zone	对象的创建时间。
last_ddl_time	timestamp with time zone	对象的最后修改时间。
default_collation	character varying(100)	对象的默认排序规则。
data_object_id	numeric	暂不支持，值为NULL。
timestamp	character varying(19)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
edition_name	character varying(128)	暂不支持，值为NULL。
sharing	character varying(18)	暂不支持，值为NULL。
editionable	character varying(1)	暂不支持，值为NULL。
oracle_maintained	character varying(1)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
application	character varying(1)	暂不支持，值为NULL。
duplicated	character varying(1)	暂不支持，值为NULL。
sharded	character varying(1)	暂不支持，值为NULL。
created_appid	numeric	暂不支持，值为NULL。
modified_appid	numeric	暂不支持，值为NULL。
created_vsnid	numeric	暂不支持，值为NULL。
modified_vsnid	numeric	暂不支持，值为NULL。

须知

created和last_ddl_time支持的范围，请参见**PG_OBJECT**中的记录范围。

7.3.13.114 MY_PROCEDURES

MY_PROCEDURES视图描述了当前用户拥有的存储过程、函数或触发器的信息。该视图同时存在于PG_CATALOG和SYS Schema下。该视图所有用户可访问，仅可查看当前用户所属信息。具体字段信息如**表7-396**所示。

表 7-396 MY_PROCEDURES 字段

名称	类型	描述
owner	character varying(64)	存储过程、函数、触发器或包的所有者。
object_name	character varying(64)	存储过程、函数或触发器的名称，若为包中函数或存储过程，则为包名。
procedure_name	character varying(128)	若object_name为包名，则为包中函数或存储过程名称，否则为空。
object_id	oid	存储过程、函数、触发器或包的OID。
subprogram_id	numeric	包中函数或存储过程的位置。
overload	character varying(40)	表示该函数是该名称的第n个重载函数。
object_type	character varying(13)	对象的类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
aggregate	character varying(3)	表示是否为聚合函数： ● yes：表示是。 ● no：表示不是。
pipelined	character varying(3)	暂不支持，值为no。
impltypeowner	character varying(128)	实现类型的所有者。
impltypename	character varying(128)	实现类型的名称。
parallel	character varying(3)	暂不支持，值为no。
interface	character varying(3)	暂不支持，值为no。
deterministic	character varying(3)	暂不支持，值为no。
authid	character varying(12)	表示是使用创建者权限还是调用者权限： ● definer：表示使用创建者权限。 ● current_user：表示使用调用者权限。 因该字段与保留关键字冲突，调用该字段需加视图名。
result_cache	character varying(3)	暂不支持，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为0。
polymorphic	character varying(5)	暂不支持，值为NULL。
argument_number	smallint	存储过程入参个数。

7.3.13.115 MY_SCHEDULER_JOB_ARGS

MY_SCHEDULER_JOB_ARG视图显示当前用户拥有的任务的有关参数信息。该视图所有用户可访问，仅可查看当前用户所属信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-397**所示。

版权所有 © 华为云计算技术有限公司

表 7-397 MY_SCHEDULER_JOB_ARGS 字段

名称	类型	描述
job_name	character varying(128)	参数所属作业名。
argument_name	character varying(128)	参数名称。
argument_position	numeric	参数在参数列表中的位置。
argument_type	character varying(257)	参数的数据类型，可以是用户的自定义数据类型。
value	character varying(4000)	参数值。
anydata_value	character varying(4000)	暂不支持，值为NULL。
out_argument	character varying(5)	保留字段，值为NULL。

7.3.13.116 MY_SCHEDULER_JOBS

MY_SCHEDULER_JOBS视图显示数据库下当前用户拥有的所有DBE_SCHEDULER定时任务的信息。默认所有用户都可以访问，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-398**所示。

表 7-398 MY_SCHEDULER_JOBS 字段

名称	类型	描述
job_name	text	定时任务名。
job_subname	character varying(128)	暂不支持，值为NULL。
job_style	text	定时任务行为模式，创建时指定，仅支持指定为“REGULAR”，不指定时为NULL。
job_creator	name	定时任务创建者。
client_id	character varying(65)	暂不支持，值为NULL。
global_uid	character varying(33)	暂不支持，值为NULL。
program_owner	character varying(4000)	定时任务引用的程序的所有者。
program_name	text	定时任务引用的程序的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
job_type	character varying(16)	定时任务内联程序类型。
job_action	text	定时任务的程序内容。
number_of_arguments	text	定时任务的参数个数。范围为0 ~ 255。
schedule_owner	character varying(4000)	暂不支持，值为NULL。
schedule_name	text	定时任务引用的调度的名称。
schedule_type	character varying(12)	暂不支持，值为NULL。
start_date	timestamp without time zone	定时任务的起始时间。
repeat_interval	text	定时任务的任务周期。
event_queue_owner	character varying(128)	暂不支持，值为NULL。
event_queue_name	character varying(128)	暂不支持，值为NULL。
event_queue_agent	character varying(523)	暂不支持，值为NULL。
event_condition	character varying(4000)	暂不支持，值为NULL。
event_rule	character varying(261)	暂不支持，值为NULL。
file_watcher_owner	character varying(261)	暂不支持，值为NULL。
file_watcher_name	character varying(261)	暂不支持，值为NULL。
end_date	timestamp without time zone	定时任务的失效时间。
job_class	text	定时任务所属的定时任务类的名称。
enabled	boolean	定时任务的启用状态。
auto_drop	text	定时任务的自动删除功能状态。
restart_on_recovery	character varying(5)	暂不支持，值为NULL。
restart_on_failure	character varying(5)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
state	"char"	定时任务的状态。
job_priority	numeric	暂不支持，值为NULL。
run_count	numeric	暂不支持，值为NULL。
uptime_run_count	numeric	暂不支持，值为NULL。
max_runs	numeric	暂不支持，值为NULL。
failure_count	smallint	定时任务失败次数统计。
uptime_failure_count	numeric	暂不支持，值为NULL。
max_failures	numeric	定时任务标记为破坏之前允许失败的最大次数。
retry_count	numeric	暂不支持，值为NULL。
last_start_date	timestamp without time zone	定时任务上次拉起时间。
last_run_duration	interval day to second(6)	定时任务上次执行的时长。
next_run_date	timestamp without time zone	定时任务下次执行时间。
schedule_limit	interval day to second(0)	暂不支持，值为NULL。
max_run_duration	interval day to second(0)	暂不支持，值为NULL。
logging_level	character varying(11)	暂不支持，值为NULL。
store_output	character varying(5)	是否存储所有定时任务的输出信息。
stop_on_window_close	character varying(5)	暂不支持，值为NULL。
instance_stickiness	character varying(5)	暂不支持，值为NULL。
raise_events	character varying(4000)	暂不支持，值为NULL。
system	character varying(5)	暂不支持，值为NULL。
job_weight	numeric	暂不支持，值为NULL。
nls_env	character varying(4000)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
source	character varying(128)	暂不支持，值为NULL。
number_of_destinations	numeric	暂不支持，值为NULL。
destination_owner	character varying(261)	暂不支持，值为NULL。
destination	text	定时任务目标名称。 如果destination值为“CCN”则说明该任务是CCN的JOB，相关说明详见**DBE_SCHEDULER**。
credential_owner	character varying(128)	暂不支持，值为NULL。
credential_name	text	定时任务的证书名称。
instance_id	oid	当前数据库的oid。
deferred_drop	character varying(5)	暂不支持，值为NULL。
allow_runs_in_restricted_mode	character varying(5)	暂不支持，值为NULL。
comments	text	定时任务的备注。
flags	numeric	暂不支持，值为NULL。
restartable	character varying(5)	暂不支持，值为NULL。
has_constraints	character varying(5)	暂不支持，值为NULL。
connect_credential_owner	character varying(128)	暂不支持，值为NULL。
connect_credential_name	character varying(128)	暂不支持，值为NULL。
fail_on_script_error	character varying(5)	暂不支持，值为NULL。

7.3.13.117 MY_SCHEDULER_PROGRAM_ARGS

MY_SCHEDULER_PROGRAM_ARG视图显示当前用户拥有的程序的有关参数信息。该视图所有用户可访问，仅可查看当前用户所属信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-399**所示。

版权所有 © 华为云计算技术有限公司

表 7-399 MY_SCHEDULER_PROGRAM_ARGS 字段

名称	类型	描述
program_name	character varying(128)	参数所属程序名。
argument_name	character varying(128)	参数名称。
argument_position	numeric	参数在参数列表中的位置。
argument_type	character varying(257)	参数的数据类型，可以是用户的自定义数据类型。
metadata_attribute	character varying(19)	暂不支持，值为NULL。
default_value	character varying(4000)	参数默认值。
default_anydata_value	character varying(4000)	暂不支持，值为NULL。
out_argument	character varying(5)	保留字段，值为NULL。

7.3.13.118 MY_SEQUENCES

MY_SEQUENCES视图显示当前用户的序列信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-400**所示。

表 7-400 MY_SEQUENCES 字段

名称	类型	描述
sequence_owner	name	序列所有者。
sequence_name	name	序列的名称。
min_value	int16	序列最小值。
max_value	int16	序列最大值。
increment_by	int16	序列的增量。
cycle_flag	character(1)	表示序列是否是循环序列，取值为： • y：表示是循环序列。 • n：表示不是循环序列。
order_flag	character(1)	表示序列是否是有序序列，取值为： • y表示是有序序列。 • n表示不是有序序列。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
cache_size	int16	序列磁盘缓存大小。
last_number	int16	该序列从GTM中一次性批量申请的最后一个值。
scale_flag	character varying(1)	表示是否为可扩展序列，暂不支持，值为NULL。
extend_flag	character varying(1)	表示可扩展序列生成的值是否超出序列最大值、最小值范围。暂不支持，值为NULL。
sharded_flag	character varying(1)	表示是否为分片序列。暂不支持，值为NULL。
session_flag	character varying(1)	表示序列是否为会话私有，暂不支持，值为NULL。
keep_value	character varying(1)	表示在失败后的replay期间是否保留序列值。暂不支持，值为NULL。

7.3.13.119 MY_SOURCE

MY_SOURCE视图显示当前用户拥有的存储过程、函数、触发器的定义信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-401**所示。

表 7-401 MY_SOURCE 字段

名称	类型	描述
owner	name	对象的所有者。
name	name	对象名字。
type	name	对象类型：function、procedure、trigger。
line	numeric	此行在定义信息中的行号。
text	text	存储对象的文本来源。
origin_con_id	character varying(256)	暂不支持，值为0。

7.3.13.120 MY_SYNONYMS

MY_SYNONYMS视图显示当前模式下同义词的信息。该视图所有用户可访问，仅可查看当前用户所属信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-402**所示。

版权所有 © 华为云计算技术有限公司

表 7-402 MY_SYNONYMS 字段

名称	类型	描述
schema_name	text	同义词所属模式名。
synonym_name	text	同义词的名称。
table_owner	text	关联对象的所有者。尽管该列称为table_owner，但拥有的该关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程或同义词等。
table_name	text	关联对象名。尽管该列称为table_name，但此关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
table_schema_name	text	关联对象所属模式名。尽管该列称为table_schema_name，但此schema下的该关联对象不一定是表，可以是任何数据库通用对象，例如视图、存储过程、同义词等。
db_link	character varying(128)	保留字段，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为0。

7.3.13.121 MY_TAB_COL_STATISTICS

MY_TAB_COL_STATISTICS视图显示从MY_TAB_COLUMNS 中提取的列统计信息和直方图信息。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-403**所示。

表 7-403 MY_TAB_COL_STATISTICS 字段

名称	类型	描述
table_name	character varying(128)	表名。
column_name	character varying(128)	列名。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的低值。
high_value	raw	列中的高值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
density	numeric	• 如果COLUMN_NAME上有直方图，则此列将显示直方图中跨越少于2个端点的值的选择性。它不代表跨越2个或更多端点的值的选择性。 • 如果COLUMN_NAME上没有可用的直方图，则该列的值为1/NUM_DISTINCT。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图中的桶数。
last_analyzed	date	最近分析此列的日期，数据库重启后，数据会丢失。
sample_size	numeric	用于分析此列的样本量。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
notes	character varying(99)	暂不支持，值为NULL。
avg_col_len	numeric	列的平均长度（以字节为单位）。
histogram	character varying(15)	表示直方图是否存在，如果存在的话是什么类型： • none：表示不存在直方图。 • frequency：表示频率直方图。 • equi-width：表示等宽直方图。
scope	character varying(7)	对于在除全局临时表之外的任何表上收集的统计信息，该值是shared（表示统计信息在所有会话之间共享）。
schema	character varying(64)	列所属的名称空间的名称。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。

7.3.13.122 MY_TAB_COLUMNS

MY_TAB_COLUMNS视图显示当前用户拥有的表和视图的字段信息。该视图同时存在于PG_CATALOG和SYS Schema下。该视图所有用户可访问，仅显示该用户所属的信息。具体字段信息如**表7-404**所示。

版权所有 © 华为云计算技术有限公司

表 7-404 MY_TAB_COLUMNS 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名称。
column_name	character varying(64)	列名。
data_type	character varying(128)	列的数据类型。
data_type_mod	character varying(3)	暂不支持，值为NULL。
data_type_owner	character varying(128)	列的数据类型的所有者。
data_length	integer	列的字节长度。
data_precision	integer	数据类型的精度，对于numeric数据类型有效，其他类型为NULL。
data_scale	integer	小数点右边的位数，对于numeric数据类型有效，其他类型为0。
nullable	bpchar	该列是否允许为空，对于主键约束和非空约束，该值为n。
column_id	integer	创建表时列的序号。
default_length	numeric	列的默认值字节长度。
data_default	text	列的默认值。
num_distinct	numeric	列中不同值的数量。
low_value	raw	列中的最小值。
high_value	raw	列中的最大值。
density	numeric	列密度。
num_nulls	numeric	列中空值数。
num_buckets	numeric	列的直方图的桶数。
last_analyzed	date	上次分析的日期。
sample_size	numeric	用于分析此列的样本量。
character_set_name	character varying(44)	暂不支持，值为NULL。
char_col_decl_length	numeric	字符类型列的声明长度。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
global_stats	character varying(3)	暂不支持，值为no。
user_stats	character varying(3)	暂不支持，值为no。
avg_col_len	numeric	列的平均长度（单位字节）。
char_length	numeric	列的长度（单位字符），只对varchar，nvarchar2，bpchar，char类型有效。
char_used	character varying(1)	暂不支持，varchar、bpchar与char类型值为B；nvarchar2类型值为C；其余值为NULL。
v80_fmt_image	character varying(3)	暂不支持，值为NULL。
data_upgraded	character varying(3)	暂不支持，值为yes。
histogram	character varying(15)	表示直方图是否存在，如果存在的话是什么类型： ● none：表示不存在直方图。 ● frequency：表示频率直方图。 ● equi-width：表示等宽直方图。
default_on_null	character varying(3)	暂不支持，值为NULL。
identity_column	character varying(3)	列是否为标识列： ● yes：表示为标识列。 ● no：表示为非标识列。
sensitive_column	character varying(3)	暂不支持，值为NULL。
evaluation_edition	character varying(128)	暂不支持，值为NULL。
unusable_before	character varying(128)	暂不支持，值为NULL。
unusable_beginning	character varying(128)	暂不支持，值为NULL。
collation	character varying(100)	列的排序规则。因该字段与保留关键字冲突，调用该字段需加视图名。
comments	text	列的注释。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schema	character varying(64)	列所属的名称空间的名称。

说明

- 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的last_analyzed字段类型会变更为datea。
- 目前CREATE TABLE语法不支持创建具有标识列字段的表，当进行查询MY_TAB_COLUMNS视图操作时，返回结果中identity_column字段值恒为no。

7.3.13.123 MY_TAB_COMMENTS

MY_TAB_COMMENTS视图显示当前用户拥有的所有表和视图的注释信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-405所示。

表 7-405 MY_TAB_COMMENTS 字段

名称	类型	描述
owner	character varying(64)	表或视图的所有者。
table_name	character varying(64)	表或视图的名称。
comments	text	注释。
schema	character varying(64)	表所属的名称空间的名称。

7.3.13.124 MY_TAB_HISTOGRAMS

MY_TAB_HISTOGRAMS系统视图显示当前用户拥有的表和视图的直方图信息。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-406所示。

表 7-406 MY_TAB_HISTOGRAMS 字段

名称	类型	描述
table_name	character varying(128)	表名。
column_name	character varying(4000)	列名。
endpoint_number	numeric	直方图的桶号。
endpoint_value	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
endpoint_actual_value	character varying(4000)	桶端点的实际值。
endpoint_actual_value_raw	raw	暂不支持，值为NULL。
endpoint_repeat_count	numeric	暂不支持，值为NULL。
scope	character varying(7)	暂不支持，值为shared。

7.3.13.125 MY_TAB_MODIFICATIONS

MY_TAB_MODIFICATIONS视图记录自上次在表上收集统计信息以来当前用户拥有的表的修改信息的统计数据。该视图目前只显示执行了INSERT、DELETE和UPDATE操作的表。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-407**所示。

表 7-407 MY_TAB_MODIFICATIONS 字段

名称	类型	描述
table_name	character varying(128)	表的名称。
partition_name	character varying(128)	分区的名称。
subpartition_name	character varying(128)	子分区的名称。
inserts	numeric	自上次收集统计信息以来的大致插入次数。
updates	numeric	自上次收集统计信息以来的大致更新次数。
deletes	numeric	自上次收集统计信息以来的大致删除次数。
timestamp	date	上次修改表的时间。 暂不支持分区表的修改时间，值为NULL。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
truncated	character varying(3)	暂不支持，值为NULL。
drop_segments	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schema_name	character varying(128)	表所属模式名。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的timestamp字段类型会变更为datea。

7.3.13.126 MY_TAB_STATS_HISTORY

MY_TAB_STATS_HISTORY视图提供当前用户所拥有的表的表统计信息历史。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-408所示。

表 7-408 MY_TAB_STATS_HISTORY 字段

名称	类型	描述
table_name	character varying(128)	表名。
partition_name	character varying(128)	暂不支持，值为NULL。
subpartition_name	character varying(128)	暂不支持，值为NULL。
stats_update_time	timestamp(6) with time zone	统计信息更新的时间，数据库重启后，数据会丢失。

7.3.13.127 MY_TAB_STATISTICS

MY_TAB_STATISTICS视图显示数据库中有关当前用户拥有的表的统计信息。所有用户均可访问该视图，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-409所示。

表 7-409 MY_TAB_STATISTICS 字段

名称	类型	描述
table_name	character varying(128)	表名。
partition_name	character varying(128)	暂不支持，值为NULL。
partition_position	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
subpartition_name	character varying(128)	暂不支持，值为NULL。
subpartition_position	numeric	暂不支持，值为NULL。
object_type	character varying(12)	对象类型： ● TABLE ● PARTITION ● SUBPARTITION
num_rows	numeric	对象中的行数。
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_row_len	integer	平均行长，包括行开销。
avg_space_freelist_blocks	numeric	暂不支持，值为NULL。
num_freelist_blocks	numeric	暂不支持，值为NULL。
avg_cached_blocks	numeric	暂不支持，值为NULL。
avg_cache_hit_ratio	numeric	暂不支持，值为NULL。
im_imcu_count	numeric	暂不支持，值为NULL。
im_block_count	numeric	暂不支持，值为NULL。
im_stat_update_time	timestamp(6) without time zone	暂不支持，值为NULL。
scan_rate	numeric	暂不支持，值为NULL。
sample_size	numeric	分析表格时使用的样本量。
last_analyzed	timestamp with time zone	最近分析表的日期。数据库重启后，数据会丢失。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
stattype_locked	character varying(5)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stale_stats	character varying(7)	暂不支持，值为NULL。
notes	character varying(25)	暂不支持，值为NULL。
scope	character varying(7)	暂不支持，默认值SHARED。

7.3.13.128 MY_TABLES

MY_TABLES视图显示当前用户拥有的表的信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-410所示。

表 7-410 MY_TABLES 字段

名称	类型	描述
owner	character varying(64)	表的所有者。
table_name	character varying(64)	表名称。
tablespace_name	character varying(64)	存储表的表空间名称。
dropped	character varying	当前表是否已删除： • yes：表示已删除。 • no：表示未删除。
num_rows	numeric	表的估计行数。
status	character varying(8)	当前表是否有效。 • valid：表示当前表有效。 • unusable：表示当前表不可用。
sample_size	numeric	分析表使用的样本数量。
temporary	character(1)	表是否为临时表： • y：表示是临时表。 • n：表示不是临时表。
pct_free	numeric	块中空闲空间的最小比例。
ini_trans	numeric	事务的初始数量。
max_trans	numeric	事务数量的最大值。
avg_row_len	integer	平均每行的字节数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
partitioned	character varying(3)	表是否为分区表。 ● yes：是分区表。 ● no：不是分区表。
last_analyzed	timestamp with time zone	上次分析表的时间。数据库重启后，数据会丢失。
row_movement	character varying(8)	是否允许分区行移动。 ● enabled：允许分区行移动。 ● disabled：不允许分区行移动。
compression	character varying(8)	是否启用表压缩。 ● enabled：启用表压缩。 ● disabled：不启用表压缩。
duration	character varying(15)	临时表的期限。 ● NULL：表示非临时表。 ● sys$session：表示会话临时表。 ● sys$transaction：表示事务临时表。
logical_replication	character varying(8)	表是否启用逻辑复制。 ● enabled：启用逻辑复制。 ● disabled：不启用逻辑复制。
external	character varying(3)	表是否为外表。 ● yes：是外表。 ● no：不是外表。
logging	character varying(3)	表的更改是否记录日志。 ● yes：表的更改记录日志。 ● no：表的更改不记录日志。
default_collation	character varying(100)	表的默认排序规则。
degree	character varying(10)	扫描表的实例数量。
table_lock	character varying(8)	是否启用表级锁。 ● enabled：启用表级锁。 ● disabled：不启用表级锁。
nested	character varying(3)	是否为嵌套表。 ● yes：是嵌套表。 ● no：不是嵌套表。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
buffer_pool	character varying(7)	表的默认缓冲池。
flash_cache	character varying(7)	用于表块的数据库智能闪存提示。
cell_flash_cache	character varying(7)	用于表块的单元闪存缓存提示。
skip_corrupt	character varying(8)	扫描表是否跳过损坏的块。 ● enabled：跳过损坏的块。 ● disabled：不跳过损坏的块。
has_identity	character varying(3)	表是否具有标识列。 ● yes：有标识列。 ● no：没有标识列。
inmemory	character varying(8)	暂不支持，取值为disabled。
segment_created	character varying(3)	表段是否已被创建。 ● yes：表段已被创建。 ● no：表段未被创建。
monitoring	character varying(3)	是否跟踪表的修改。 ● yes：跟踪表的修改。 ● no：不跟踪表的修改。
cluster_name	character varying(128)	暂不支持，值为NULL。
iot_name	character varying(128)	暂不支持，值为NULL。
pct_used	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
freelists	numeric	暂不支持，值为NULL。
freelist_groups	numeric	暂不支持，值为NULL。
backed_up	character varying(1)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
blocks	numeric	暂不支持，值为NULL。
empty_blocks	numeric	暂不支持，值为NULL。
avg_space	numeric	暂不支持，值为NULL。
chain_cnt	numeric	暂不支持，值为NULL。
avg_space_freelist_blocks	numeric	暂不支持，值为NULL。
num_freelist_blocks	numeric	暂不支持，值为NULL。
instances	character varying(10)	暂不支持，值为NULL。
cache	character varying(5)	暂不支持，值为NULL。
iot_type	character varying(12)	暂不支持，值为NULL。
secondary	character varying(1)	暂不支持，值为NULL。
global_stats	character varying(3)	暂不支持，值为NULL。
user_stats	character varying(3)	暂不支持，值为NULL。
cluster_owner	character varying(30)	暂不支持，值为NULL。
dependencies	character varying(8)	暂不支持，值为NULL。
compression_for	character varying(30)	暂不支持，值为NULL。
read_only	character varying(3)	暂不支持，值为NULL。
result_cache	character varying(7)	暂不支持，值为NULL。
clustering	character varying(3)	暂不支持，值为NULL。
activity_tracking	character varying(23)	暂不支持，值为NULL。
dml_timestamp	character varying(25)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
container_data	character varying(3)	暂不支持，值为NULL。
inmemory_priority	character varying(8)	暂不支持，值为NULL。
inmemory_distribute	character varying(15)	暂不支持，值为NULL。
inmemory_compression	character varying(17)	暂不支持，值为NULL。
inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
duplicated	character varying(1)	暂不支持，值为NULL。
sharded	character varying(1)	暂不支持，值为NULL。
hybrid	character varying(3)	暂不支持，值为NULL。
cellmemory	character varying(24)	暂不支持，值为NULL。
containers_default	character varying(3)	暂不支持，值为NULL。
container_map	character varying(3)	暂不支持，值为NULL。
extended_data_link	character varying(3)	暂不支持，值为NULL。
extended_data_link_map	character varying(3)	暂不支持，值为NULL。
inmemory_service	character varying(12)	暂不支持，值为NULL。
inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
container_map_object	character varying(3)	暂不支持，值为NULL。
memoptimize_read	character varying(8)	暂不支持，值为NULL。
memoptimize_write	character varying(8)	暂不支持，值为NULL。
has_sensitive_column	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
admit_null	character varying(3)	暂不支持，值为NULL。
data_link_dml_enabled	character varying(3)	暂不支持，值为NULL。
object_id_type	character varying(16)	暂不支持，值为NULL。
table_type_owner	character varying(128)	暂不支持，值为NULL。
table_type	character varying(128)	暂不支持，值为NULL。
compress_for	character varying(30)	暂不支持，值为NULL。

7.3.13.129 MY_TABLESPACES

MY_TABLESPACES视图显示用户拥有存储对象的表空间的描述信息。默认所有用户都可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-411**所示。

表 7-411 MY_TABLESPACES 字段

名称	类型	描述
tablespace_name	character varying(64)	表空间名称。
block_size	numeric	暂不支持，值为NULL。
initial_extent	numeric	暂不支持，值为NULL。
next_extent	numeric	暂不支持，值为NULL。
min_extents	numeric	暂不支持，值为NULL。
max_extents	numeric	暂不支持，值为NULL。
max_size	numeric	暂不支持，值为NULL。
pct_increase	numeric	暂不支持，值为NULL。
min_extlen	numeric	暂不支持，值为NULL。
contents	character varying(9)	暂不支持，值为NULL。
status	character varying(9)	表空间状态，默认为online（在线）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
logging	character varying(9)	暂不支持，值为NULL。
force_logging	character varying(3)	暂不支持，值为NULL。
extent_management	character varying(10)	暂不支持，值为NULL。
allocation_type	character varying(9)	暂不支持，值为NULL。
segment_space_management	character varying(6)	暂不支持，值为NULL。
def_tab_compression	character varying(8)	暂不支持，值为NULL。
retention	character varying(11)	暂不支持，值为NULL。
bigfile	character varying(3)	暂不支持，值为NULL。
predicate_evaluation	character varying(7)	暂不支持，值为NULL。
encrypted	character varying(3)	暂不支持，值为NULL。
compress_for	character varying(30)	暂不支持，值为NULL。
def_inmemory	character varying(8)	暂不支持，值为NULL。
def_inmemory_priority	character varying(8)	暂不支持，值为NULL。
def_inmemory_distribute	character varying(15)	暂不支持，值为NULL。
def_inmemory_compression	character varying(17)	暂不支持，值为NULL。
def_inmemory_duplicate	character varying(13)	暂不支持，值为NULL。
shared	character varying(12)	暂不支持，值为NULL。
def_index_compression	character varying(8)	暂不支持，值为NULL。
index_compress_for	character varying(13)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
def_cellmemory	character varying(14)	暂不支持，值为NULL。
def_inmemory_service	character varying(12)	暂不支持，值为NULL。
def_inmemory_service_name	character varying(1000)	暂不支持，值为NULL。
lost_write_protect	character varying(7)	暂不支持，值为NULL。
chunk_tablespace	character varying(1)	暂不支持，值为NULL。

7.3.13.130 MY_TRIGGERS

MY_TRIGGERS视图显示当前用户拥有的触发器的信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-412所示。

表 7-412 MY_TRIGGERS 字段

名称	类型	描述
owner	character varying(128)	触发器的所有者。
trigger_name	character varying(64)	触发器名称。
trigger_type	character varying	触发器触发的时机：before statement、before each row、after statement、after each row和instead of。
triggering_event	character varying	触发触发器的事件：update、insert、delete和truncate。
table_owner	character varying(64)	定义触发器的表的所有者。
base_object_type	character varying(18)	定义触发器的基础对象：table和view。
table_name	character varying(64)	定义触发器的表或视图名称。
column_name	character varying(4000)	暂不支持，值为NULL。
referencing_name	character varying(422)	暂不支持，值为referencing new as new old as old。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
when_clause	character varying(4000)	WHEN子句的内容，必须求值为true才能执行trigger_body。
status	character varying(64)	• O：触发器在“origin”和“local”模式下触发。 • D：触发器被禁用。 • R：触发器在“replica”模式下触发。 • A：触发器始终触发。
description	character varying(4000)	触发器描述，用于重新创建触发器创建语句。
action_type	character varying(11)	触发体的动作类型，仅支持call。
trigger_body	text	触发器触发时执行的语句。
crossedition	character varying(7)	暂不支持，值为NULL。
before_statement	character varying(3)	暂不支持，值为NULL。
before_row	character varying(3)	暂不支持，值为NULL。
after_row	character varying(3)	暂不支持，值为NULL。
after_statement	character varying(3)	暂不支持，值为NULL。
instead_of_row	character varying(3)	暂不支持，值为NULL。
fire_once	character varying(3)	暂不支持，值为NULL。
apply_server_only	character varying(3)	暂不支持，值为NULL。

7.3.13.131 MY_TYPE_ATTRS

MY_TYPE_ATTRS视图显示数据库中当前用户所拥有的所有类型的属性。所有用户均可访问该视图，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-413**所示。

版权所有 © 华为云计算技术有限公司

表 7-413 MY_TYPE_ATTRS 字段

名称	类型	描述
type_name	character varying(128)	数据类型名称。
attr_name	character varying(128)	属性名称。
attr_type_mod	character varying(7)	属性的类型修饰符： • REF • POINT
attr_type_owner	character varying(128)	属性类型的所有者。
attr_type_name	character varying(128)	属性类型的名称。
length	numeric	CHAR属性的长度，或VARCHAR和character varying属性的最大长度。
precision	numeric	数字或DECIMAL属性的十进制精度，或FLOAT属性的二进制精度。
scale	numeric	NUMERIC或DECIMAL属性的小数位。
character_set_name	character varying(44)	属性的字符集名称（CHAR_CS或NCHAR_CS）。
attr_no	numeric	类型规范或CREATE TYPE语句中指定的属性的语法顺序编号或位置（不用作id编号）。
inherited	character varying(3)	表示属性是否继承自超类型。 • YES：表示继承自超类型。 • NO：表示没有继承自超类型。

7.3.13.132 MY_TYPES

MY_TYPES视图用于描述当前用户所拥有的所有对象类型。所有用户都可以访问该视图。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-414**所示。

表 7-414 MY_TYPES 字段

名称	类型	描述
type_name	character varying(128)	类型名称。
type_oid	raw	类型的标识符（OID）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
typecode	character varying(128)	类型的类型代码。
attributes	numeric	类型中的属性数。
methods	numeric	暂不支持，值为0。
predefined	character varying(3)	表示该类型是否是内置类型。
incomplete	character varying(3)	表示类型是否为不完整类型。
final	character varying(3)	暂不支持，值为NULL。
instantiable	character varying(3)	暂不支持，值为NULL。
persistable	character varying(3)	暂不支持，值为NULL。
supertype_owner	character varying(128)	暂不支持，值为NULL。
supertype_name	character varying(128)	暂不支持，值为NULL。
local_attributes	numeric	暂不支持，值为NULL。
local_methods	numeric	暂不支持，值为NULL。
typeid	raw	暂不支持，值为NULL。

7.3.13.133 MY_VIEWS

MY_VIEWS视图显示当前用户的所有视图信息。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-415所示。

表 7-415 MY_VIEWS 字段

名称	类型	描述
owner	character varying(64)	视图的所有者。
view_name	character varying(64)	视图名称。
text	text	视图文本。
text_length	integer	视图文本长度。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
text_vc	character varying(4000)	视图创建语句。此列可能会截断视图文本。BEQUEATH子句将不会作为此视图中的TEXT_VC列的一部分出现。
type_text_length	numeric	暂不支持，值为NULL。
type_text	character varying(4000)	暂不支持，值为NULL。
oid_text_length	numeric	暂不支持，值为NULL。
oid_text	character varying(4000)	暂不支持，值为NULL。
view_type_owner	character varying(128)	暂不支持，值为NULL。
view_type	character varying(128)	暂不支持，值为NULL。
superview_name	character varying(128)	暂不支持，值为NULL。
editioning_view	character varying(1)	暂不支持，值为NULL。
read_only	character varying(1)	暂不支持，值为NULL。
container_data	character varying(1)	暂不支持，值为NULL。
bequeath	character varying(12)	暂不支持，值为NULL。
origin_con_id	character varying(256)	暂不支持，值为NULL。
default_collation	character varying(100)	暂不支持，值为NULL。
containers_default	character varying(3)	暂不支持，值为NULL。
container_map	character varying(3)	暂不支持，值为NULL。
extended_data_link	character varying(3)	暂不支持，值为NULL。
extended_data_link_map	character varying(3)	暂不支持，值为NULL。
has_sensitive_column	character varying(3)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
admit_null	character varying(3)	暂不支持，值为NULL。
pdb_local_only	character varying(3)	暂不支持，值为NULL。

7.3.13.134 NLS_DATABASE_PARAMETERS

NLS_DATABASE_PARAMETERS列出数据库服务器端的永久NLS参数。该视图同时存在于PG_CATALOG和SYS Schema下。所有用户都可以访问。由于数据库内核不同、参数的设置格式不同等原因，该视图对相同参数的查询结果可能会和ORA数据库有明显差异。具体字段信息如表7-416所示。

表 7-416 NLS_DATABASE_PARAMETERS 字段

名称	类型	描述
parameter	character varying(128)	参数名。
value	character varying(64)	参数值。

7.3.13.135 NLS_INSTANCE_PARAMETERS

NLS_INSTANCE_PARAMETERS列出数据库客户端的永久NLS参数。该视图同时存在于PG_CATALOG和SYS Schema下。所有用户都可以访问。由于数据库内核不同、参数的设置格式不同等原因，该视图对相同参数的查询结果中可能会和ORA数据库有明显差异。具体字段信息如表7-417所示。

表 7-417 NLS_INSTANCE_PARAMETERS 字段

名称	类型	描述
parameter	character varying(128)	参数名。
value	character varying(64)	参数值。

7.3.13.136 PG_AVAILABLE_EXTENSION_VERSIONS

PG_AVAILABLE_EXTENSION_VERSIONS视图显示数据库中某些特性的扩展版本信息。具体字段信息如表7-418所示。

版权所有 © 华为云计算技术有限公司

表 7-418 PG_AVAILABLE_EXTENSION_VERSIONS 字段

名称	类型	描述
name	name	扩展名。
version	text	版本名。
installed	boolean	如果这个扩展的版本是当前已经安装了的则为真。否则为假。
superuser	boolean	如果只允许系统管理员安装这个扩展则为真。否则为假。
relocatable	boolean	如果扩展可以重新加载到另一个模式则为真。否则为假。
schema	name	扩展必须安装到的模式名，如果部分或全部可重新定位则为NULL。
requires	name[]	先决条件扩展的名称，如果没有则为NULL。
comment	text	扩展的控制文件中的评论。

7.3.13.137 PG_AVAILABLE_EXTENSIONS

PG_AVAILABLE_EXTENSIONS视图显示数据库中某些特性的扩展信息。具体字段信息如**表7-419**所示。

表 7-419 PG_AVAILABLE_EXTENSIONS 字段

名称	类型	描述
name	name	扩展名。
default_version	text	缺省版本的名称，如果没有指定则为NULL。
installed_version	text	扩展当前安装版本，如果没有安装任何版本则为NULL。
comment	text	扩展的控制文件中的评论。

7.3.13.138 PG_CONTROL_GROUP_CONFIG

PG_CONTROL_GROUP_CONFIG视图显示系统的控制组配置信息。查询该视图需要sysadmin权限。具体字段信息如**表7-420**所示。

版权所有 © 华为云计算技术有限公司

表 7-420 PG_CONTROL_GROUP_CONFIG 字段

名称	类型	描述
pg_control_group_config	text	控制组的配置信息。

7.3.13.139 PG_CURSORS

PG_CURSORS视图列出了当前可用的游标。具体字段信息如表7-421所示。

表 7-421 PG_CURSORS 字段

名称	类型	描述
name	text	游标名。
statement	text	声明该游标时的查询语句。
is_holdable	boolean	如果该游标是持久的（就是在声明该游标的事务结束后仍然可以访问该游标）则为TRUE，否则为FALSE。
is_binary	boolean	如果该游标被声明为BINARY则为TRUE，否则为FALSE。
is_scrollable	boolean	如果该游标可以滚动（就是允许以不连续的方式检索）则为TRUE，否则为FALSE。
creation_time	timestamp with time zone	声明该游标的时间戳。

7.3.13.140 PG_EXT_STATS

PG_EXT_STATS视图可用来访问存储在PG_STATISTIC_EXT表里面的扩展统计信息。扩展统计信息目前包括多列统计信息。具体字段信息如表7-422所示。

表 7-422 PG_EXT_STATS 字段

名称	类型	引用	描述
schemaname	name	PG_NAMESPACE.nspname	表的模式名。
tablename	name	PG_CLASS.relname	表名。
attname	int2vector	PG_STATISTIC_EXT.stakey	统计信息扩展的多列信息。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
inherited	boolean	-	暂不支持继承表，该字段为false。
null_frac	real	-	记录中字段组合为空的百分比。
avg_width	integer	-	字段组合记录以字节记的平均宽度。
n_distinct	real	-	• 如果大于零，表示字段组合中独立数值的估计数目。 • 如果小于零，表示独立数值的数目除以行数后乘-1得到的负数。比如，-1表示一个字段组合中独立数值的个数和行数相同。 1. 用负数形式是因为ANALYZE认为独立数值的数目是随着表增长而增长； 2. 正数的形式用于在字段看上去好像有固定的可能值数目的情况下。 • 如果等于零，表示独立数值的数目未知。
n_dndistinct	real	-	标识dn1上字段组合中非NULL的独立数值的数目。 • 如果大于零，表示独立数值的实际数目。 • 如果小于零，表示独立数值的数目除以行数后乘-1得到的负数。比如，一个字段组合的数值平均出现概率为两次，则可以表示为n_dndistinct=-0.5。 • 如果等于零，表示独立数值的数目未知。
most_common_vals	anyarray	-	一个字段组合里最常用数值的列表。如果该字段组合不存在最常用数值，则为NULL。本列保存的多列常用数值均不为NULL。
most_common_freqs	real[]	-	一个记录字段组合里最常用数值的出现频率的列表，频率由每个数值出现的次数除以行数得到。如果most_common_vals取值为NULL，则该字段取值也为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
most_common_vals_null	anyarray	-	一个字段组合里最常用数值的列表。如果该字段组合不存在最常用数值，则为NULL。本列保存的多列常用数值中至少有一个值为NULL。
most_common_freqs_null	real[]	-	一个记录字段组合里最常用数值的出现频率的列表，频率由每个数值出现的次数除以行数得到。如果most_common_vals_null取值为NULL，则该字段取值也为NULL。
histogram_bounds	anyarray	-	直方图的边界值列表。
partitionname	name	PG_PARTITION.relname	分区表的一级分区名。对于非分区表，该字段为空。
subpartitionname	name	PG_PARTITION.relname	分区表的二级分区名。对于分布式，该字段为空。

7.3.13.141 PG_GET_SENDERS_CATCHUP_TIME

PG_GET_SENDERS_CATCHUP_TIME视图显示DN上当前活跃的主备发送线程的追赶信息。具体字段信息如表7-423所示。

表 7-423 PG_GET_SENDERS_CATCHUP_TIME 字段

名称	类型	描述
pid	bigint	当前sender的线程ID。
lwpid	integer	当前sender的lwpid。
local_role	text	本地的角色。
peer_role	text	对端的角色。
state	text	当前sender的复制状态。 • Startup：启动状态。 • Backup：备份状态。 • Catchup：追赶状态，表示备节点正在追赶主节点。 • Streaming：流复制状态，当备节点追上主节点后维持Streaming状态。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
type	text	当前sender的类型。 • Wal：预写入日志类型。 • Data：数据类型。
catchup_start	timestamp with time zone	catchup启动的时间。
catchup_end	timestamp with time zone	catchup结束的时间。

7.3.13.142 PG_GROUP

PG_GROUP视图可用来查看数据库认证角色和组之间的成员关系。具体字段信息如表7-424所示。

表 7-424 PG_GROUP 字段

名称	类型	描述
groname	name	组的名称。
grosysid	oid	组的id。
grolist	oid[]	包含该组里面所有角色的id的数组。

7.3.13.143 PG_GTT_STATS

PG_GTT_STATS视图可用来查看当前会话所有全局临时表的单列统计信息，调用pg_get_gtt_statistics()函数。使用时，需先对表进行ANALYZE。

表 7-425 PG_GTT_STATS 字段

名称	类型	描述
schemaname	name	schema名称。
tablename	name	全局临时表名称。
attname	name	属性名称。
inherited	boolean	是否统计有继承关系的对象。
null_frac	real	该字段中为NULL的记录的比率。
avg_width	integer	非NULL记录的平均存储宽度，以字节计算。
n_distinct	real	标识全局统计信息中字段里唯一的非NULL数据值的数目。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
most_common_vals	text[]	高频值列表，按照出现的频率排序。
most_common_freqs	real[]	高频值的频率。
histogram_bounds	text[]	等频直方图描述列中的数据分布（不包含高频值）。
correlation	real	相关系数。
most_common_elems	text[]	类型高频值列表，用于数组类型或一些其他类型。
most_common_elem_freqs	real[]	类型高频值的频率。
elem_count_histogram	real[]	数组类型直方图。

7.3.13.144 PG_GTT_RELSTATS

PG_GTT_RELSTATS视图可用来查看当前会话所有全局临时表的基本信息，调用pg_get_gtt_relstats()函数。分布式全局临时表在CN上查询结果不具有参考价值，DN上查询结果为空，因此不建议在分布式上使用。

表 7-426 PG_GTT_RELSTATS 字段

名称	类型	描述
schemaname	name	schema名称。
tablename	name	全局临时表名称。
relfilenode	oid	文件对象的ID。
relpages	integer	全局临时表的磁盘页面数。
reltuples	real	全局临时表的记录数。
relallvisible	integer	被标识为全可见的页面数。
relfrozenxid	xid	该表中所有在这个之前的事务ID已经被一个固定的（frozen）事务ID替换。
relminmxid	xid	预留接口，暂未启用。

7.3.13.145 PG_GTT_ATTACHED_PIDS

PG_GTT_ATTACHED_PIDS视图可用来查看哪些会话正在使用全局临时表，调用pg_get_attached_pid()函数。分布式下推荐使用GS_GTT_ATTACHED_PIDS或GS_GLOBAL_GTT_ATTACHED_PIDS。

版权所有 © 华为云计算技术有限公司

表 7-427 PG_GTT_ATTACHED_PIDS 字段

名称	类型	描述
schemaname	name	schema名称。
tablename	name	全局临时表名称。
relid	oid	全局临时表的oid。
pids	bigint[]	线程pid列表。
sessionids	bigint[]	会话id列表。

7.3.13.146 PG_INDEXES

PG_INDEXES视图显示数据库中每个索引的信息。具体字段信息如**表7-428**所示。

表 7-428 PG_INDEXES 字段

名称	类型	引用	描述
schemaname	name	**PG_NAMESPACE**.nspname	包含表和索引的模式的名称。
tablename	name	**PG_CLASS**.relname	此索引所在的表的名称。
indexname	name	**PG_CLASS**.relname	索引的名称。
tablespace	name	**PG_TABLESPACE**.nspname	包含索引的表空间的名称。
indexdef	text	-	索引定义（一个重建的CREATE INDEX命令）。

7.3.13.147 PG_LOCKS

PG_LOCKS视图显示各打开事务所持有的锁的信息。具体字段信息如**表7-429**所示。

表 7-429 PG_LOCKS 字段

名称	类型	引用	描述
locktype	text	-	被锁定对象的类型：relation、extend、page、tuple、transactionid、virtualxid、object、userlock、advisory或tablespace。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
database	oid	PG_DATABASE.oid	被锁定对象所在数据库的OID。 • 如果被锁定的对象是共享对象，则OID为0。 • 如果被锁定的对象是一个事务，则OID为NULL。
relation	oid	PG_CLASS.oid	关系的OID，如果锁定的对象不是关系，也不是关系的一部分，则为NULL。
page	integer	-	关系内部的页面编号，如果对象不是关系页或者不是行页，则为NULL。
tuple	smallint	-	页面里边的行编号，如果对象不是行，则为NULL。
bucket	integer	-	哈希桶编号。
virtualxid	text	-	虚拟事务的id，如果对象不是一个虚拟事务，则为NULL。
transactionid	xid	-	事务的id，如果对象不是一个事务，则为NULL。
classid	oid	PG_CLASS.oid	包含该对象的系统表的OID，如果对象不是普通的数据库对象，则为NULL。
objid	oid	-	对象在其系统表内的OID，如果对象不是普通的数据库对象，则为NULL。
objsubid	smallint	-	对于表的一个字段，这是字段编号；对于其他对象类型，这个字段是零；如果这个对象不是普通数据库对象，则为NULL。
virtualtransaction	text	-	持有此锁或者在等待此锁的虚拟事务的虚拟id。
pid	bigint	-	持有或者等待这个锁的服务器线程的逻辑id。如果锁是被一个预备事务持有的，则为NULL。
sessionid	bigint	-	持有或者等待这个锁的会话的id。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
mode	text	-	这个线程持有的或者是期望的锁模式。 取值为：AccessShareLock、RowShareLock、RowExclusiveLock、ShareLock、ShareRowExclusiveLock、ExclusiveLock或AccessExclusiveLock。
granted	boolean	-	• 如果锁是持有锁，则为TRUE。 • 如果锁是等待锁，则为FALSE。
fastpath	boolean	-	如果通过fast-path获得锁，则为TRUE；如果通过主要的锁表获得，则为FALSE。
locktag	text	-	会话等待锁信息，可通过locktag_decode()函数解析。
global_sessionid	text	-	全局会话id。

7.3.13.148 PG_NODE_ENV

PG_NODE_ENV视图显示当前节点的环境变量信息。该视图只有monitor admin和sysadmin权限可以查看。具体字段信息如表7-430所示。

表 7-430 PG_NODE_ENV 字段

名称	类型	描述
node_name	text	当前节点的名称。
host	text	当前节点的主机名称。
process	integer	当前节点的进程号。
port	integer	当前节点的端口号。
installpath	text	当前节点的安装目录。
datapath	text	当前节点的数据目录。
log_directory	text	当前节点的日志目录。

7.3.13.149 PG_OS_THREADS

PG_OS_THREADS视图提供当前节点下所有线程的状态信息。具体字段信息如表7-431所示。

版权所有 © 华为云计算技术有限公司

表 7-431 PG_OS_THREADS 字段

名称	类型	描述
node_name	text	当前节点的名称。
pid	bigint	当前节点进程中正在运行的线程号。
lwpid	integer	与pid对应的轻量级线程号。
thread_name	text	与pid对应的线程名称。
creation_time	timestamp with time zone	与pid对应的线程创建的时间。
dbid	oid	线程所属的数据库id。

7.3.13.150 PG_PREPARED_STATEMENTS

PG_PREPARED_STATEMENTS视图显示当前会话所有可用的预备语句的信息。具体字段信息如**表7-432**所示。

表 7-432 PG_PREPARED_STATEMENTS 字段

名称	类型	描述
name	text	预备语句的标识符。
statement	text	创建该预备语句的查询字符串。对于从SQL创建的预备语句而言是客户端提交的PREPARE语句；对于通过前/后端协议创建的预备语句而言是预备语句自身的文本。
prepare_time	timestamp with time zone	创建该预备语句的时间戳。
parameter_types	regtype[]	该预备语句期望的参数类型，以regtype类型的数组格式出现。与该数组元素相对应的OID可以通过把regtype转换为OID值得到。
from_sql	boolean	• 如果该预备语句是通过PREPARE语句创建的则为true。 • 如果是通过前/后端协议创建的则为false。

7.3.13.151 PG_PREPARED_XACTS

PG_PREPARED_XACTS视图显示当前准备好进行两阶段提交的事务的信息。具体字段信息如**表7-433**所示。

版权所有 © 华为云计算技术有限公司

表 7-433 PG_PREPARED_XACTS 字段

名称	类型	引用	描述
transaction	xid	-	预备事务的数字标识。
gid	text	-	预备事务的全局标识。
prepared	timestamp with time zone	-	事务准备好提交的时间。
owner	name	PG_AUTHID.rolname	执行该事务的用户的名称。
database	name	PG_DATABASE.datname	执行该事务的数据库的名称。

7.3.13.152 PG_REPLICATION_SLOTS

PG_REPLICATION_SLOTS视图显示复制槽的信息。具体字段信息如表7-434所示。

表 7-434 PG_REPLICATION_SLOTS 字段

名称	类型	描述
slot_name	text	复制槽的名称
plugin	text	逻辑复制槽对应的输出插件名称。
slot_type	text	复制槽的类型。 • physical：物理复制槽。 • logical：逻辑复制槽。
datoid	oid	复制槽所在的数据库OID。
database	name	复制槽所在的数据库名称。
active	boolean	复制槽是否为激活状态。 • t（true）：表示是。 • f（false）：表示不是。
xmin	xid	数据库需要为复制槽保留的最早事务的事务号。
catalog_xmin	xid	数据库需要为逻辑复制槽保留的最早的涉及系统表的事务的事务号。
restart_lsn	text	复制槽需要的最早Xlog的物理位置。
dummy_standby	boolean	预留参数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
confirmed_flush	text	逻辑复制槽专用，客户端确认接收到的日志位置。
confirmed_csn	xid	逻辑复制槽专用，客户端确认接收到的日志中最后一个事务对应的CSN。

示例：

```
--在DN上执行查询。
gaussdb=# SELECT * FROM pg_replication_slots;
 slot_name |      plugin      | slot_type | datoid | database | active | xmin | catalog_xmin | restart_lsn |
dummy_standby | confirmed_flush | confirmed_csn
-----------+----------------+-----------+--------+----------+--------+------+--------------+-------------+---------------
+-----------------+---------------
 dn_6002   |                | physical |      0 |          | t      |      |              | 0/3622B528  | f               |               |
 dn_6003   |                | physical |      0 |          | t      |      |              | 0/3622B528  | f               |               |
 slot_lsn  | mppdb_decoding | logical  | 131072 | db_test  | f      |      |        66658 | 0/36252350  | f               |
0/362523D0      |
 slot_test | mppdb_decoding | logical  | 131072 | db_test  | f      |      |        66658 | 0/36251718  | f
|                 |      10025527
(4 rows)

--在CN上执行查询。
gaussdb=# SELECT * FROM pg_replication_slots;
 slot_name |      plugin      | slot_type | datoid | database | active | xmin | catalog_xmin | restart_lsn |
dummy_standby | confirmed_flush | confirmed_csn
-----------+----------------+-----------+--------+----------+--------+------+--------------+-------------+---------------
+-----------------+---------------
 slot_test | mppdb_decoding | logical  | 139264 | db_test  | f      |      |              |             | f
|                 |
(1 row)
```

须知

在DN上执行查询，LSN序逻辑复制槽的confirmed_csn查询结果为空，CSN序逻辑复制槽的confirmed_flush查询结果为空；在CN上执行查询，CSN序逻辑复制槽的catalog_xmin、restart_lsn、confirmed_flush和confirmed_csn均不予显示，查询结果为空。

7.3.13.153 PG_RULES

PG_RULES视图可用来查询重写规则的有用信息。具体字段信息如表7-435所示。

表 7-435 PG_RULES 字段

名称	类型	描述
schemaname	name	表的模式名称。
tablename	name	规则作用的表的名称。
rulename	name	规则的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
definition	text	规则的定义（由CREATE语句重新构造获得）。

7.3.13.154 PG_RUNNING_XACTS

PG_RUNNING_XACTS视图显示当前节点运行事务的信息。具体字段信息如表7-436所示。

表 7-436 PG_RUNNING_XACTS 字段

名称	类型	描述
handle	integer	事务在GTM对应的句柄。
gxid	xid	事务id号。
state	tinyint	事务状态。 ● 3：prepared。 ● 0：starting。
node	text	节点名称。
xmin	xid	节点上当前数据涉及的最小事务号。
vacuum	boolean	表示当前事务是否是lazy vacuum事务。 ● t（true）：表示是。 ● f（false）：表示不是。
timeline	bigint	数据库重启次数。
prepare_xid	xid	处于prepared状态的事务的id，若事务不在prepared状态，值为0。
pid	bigint	事务对应的线程id。
next_xid	xid	CN传给DN的事务id。
dbid	oid	事务对应的数据库id。

7.3.13.155 PG_SESSION_IOSTAT

PG_SESSION_IOSTAT视图显示当前用户执行作业正在运行时的I/O负载管理相关信息。查询该视图需要sysadmin权限或者monitor admin权限。具体字段信息如表7-437所示。

涉及到iops，以万次/s为单位。

版权所有 © 华为云计算技术有限公司

说明

当查询视图无数据显示时，请联系华为工程师提供技术支撑。

表 7-437 PG_SESSION_IOSTAT 字段

名称	类型	描述
query_id	bigint	作业id。
mincurriops	integer	该作业当前I/O在各DN中的最小值。
maxcurriops	integer	该作业当前I/O在各DN中的最大值。
minpeakiops	integer	作业运行时，在各DN的最小作业I/O峰值。
maxpeakiops	integer	作业运行时，在各DN的最大作业I/O峰值。
io_limits	integer	该作业所设GUC参数io_limits的取值。
io_priority	text	该作业所设GUC参数io_priority的取值。
query	text	作业。
node_group	text	作业所属用户对应的node_group。
curr_io_limits	integer	使用io_priority管控I/O时的实时io_limits值。

7.3.13.156 PG_SETTINGS

PG_SETTINGS视图显示数据库运行时参数的相关信息。具体字段信息如表7-438所示。

表 7-438 PG_SETTINGS 字段

名称	类型	描述
name	text	参数名称。
setting	text	参数当前值。
unit	text	参数的单位。
category	text	参数的逻辑组。
short_desc	text	参数的简单描述。
extra_desc	text	参数的详细描述。
context	text	设置参数值的上下文，包括：internal、postmaster、sighup、backend、superuser、user。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
vartype	text	参数类型，包括：bool、enum、integer、real或string。
source	text	参数的赋值方式。
min_val	text	参数最小值。如果参数类型不是数值型，那么该字段值为null。
max_val	text	参数最大值。如果参数类型不是数值型，那么该字段值为null。
enumvals	text[]	enum类型参数合法值。如果参数类型不是enum型，那么该字段值为null。
boot_val	text	数据库启动时参数默认值。
reset_val	text	数据库重置时参数默认值。
sourcefile	text	设置参数值的配置文件。如果参数不是通过配置文件赋值，那么该字段值为null。
sourceline	integer	设置参数值的配置文件的行号。如果参数不是通过配置文件赋值，那么该字段值为null。

说明

在ORA兼容模式下，PG_SETTINGS字段的查询结果不会为空字符串，而在其他兼容模式下，PG_SETTINGS字段的查询结果可能为空字符串而非null值。

7.3.13.157 PG_SHARED_MEMORY_DETAIL

PG_SHARED_MEMORY_DETAIL视图显示所有已产生的共享内存上下文的使用信息。具体字段信息如表7-439所示。

表 7-439 PG_SHARED_MEMORY_DETAIL 字段

名称	类型	描述
contextname	text	内存上下文的名称。
level	smallint	当前上下文在整体内存上下文中的层级。
parent	text	上级内存上下文。
totalsize	bigint	共享内存总大小，单位Byte。
freesize	bigint	共享内存剩余大小，单位Byte。
usedsize	bigint	共享内存使用大小，单位Byte。

版权所有 © 华为云计算技术有限公司

7.3.13.158 PG_STATS

PG_STATS视图可用来查看存储在pg_statistic表里面的单列统计信息。该视图记录的统计信息更新时间间隔由GUC参数autovacuum_naptime设置。具体字段信息如**表7-440**所示。

表 7-440 PG_STATS 字段

名称	类型	引用	描述
schemaname	name	**PG_NAMESPACE**.nspname	表的模式名。
tablename	name	**PG_CLASS**.relname	表名。
attname	name	**PG_ATTRIBUTE**.attname	字段的名称。
inherited	boolean	-	暂不支持继承表，该字段为f（false），表示否。
null_frac	real	-	记录中字段为空的百分比。
avg_width	integer	-	字段记录以字节记的平均宽度。
n_distinct	real	-	• 如果大于零，表示字段中独立数值的估计数目。 • 如果小于零，表示独立数值的数目除以行数后乘-1得到的负数。比如，-1表示一个唯一字段，独立数值的个数和行数相同。 1. 用负数形式是因为ANALYZE认为独立数值的数目是随着表增长而增长； 2. 正数的形式用于在字段看上去好像有固定的可能值数目的情况下。 • 如果等于零，表示独立数值的数目未知。
n_dndistinct	real	-	标识dn1上字段中非NULL的独立数值的数目。 • 如果大于零，表示独立数值的实际数目。 • 如果小于零，表示独立数值的数目除以行数后乘-1得到的负数。比如，一个字段的数值平均出现概率为两次，则可以表示为n_dndistinct=-0.5。 • 如果等于零，表示独立数值的数目未知。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
most_common_vals	anyarray	-	一个字段里最常用数值的列表。如果该字段不存在最常用数值，则为NULL。
most_common_freqs	real[]	-	一个记录字段里最常用数值的出现频率的列表，频率由每个数值出现的次数除以行数得到。如果most_common_vals是NULL，则为NULL。
histogram_bounds	anyarray	-	由排除了空值和MCV值之外的取值组成的等频直方图。如果某个数值出现在most_common_vals中，则不出现在直方图里。如果字段数据类型没有<操作符或者most_common_vals列表包含了该字段所有取值，则这个字段的直方图信息为NULL。
correlation	real	-	字段值的物理行序和逻辑行序的相关性。取值范围从-1到+1。该值接近-1或者+1的时候，因为减少了对磁盘的随机访问，索引扫描的开销比接近零的时候更少。如果字段数据类型没有<操作符，则这个字段的相关性为NULL。
most_common_elems	anyarray	-	一个最常用的非空元素的列表。
most_common_elem_freqs	real[]	-	一个记录最常用的非空元素的出现频率的列表。
elem_count_histogram	real[]	-	对于独立的非空元素的统计直方图。
partitionname	name	PG_PARTITION.relname	分区表的一级分区名。对于非分区表，该字段为空。
subpartitionname	name	PG_PARTITION.relname	分区表的二级分区名。对于分布式，该字段为空。

7.3.13.159 PG_STAT_ACTIVITY

PG_STAT_ACTIVITY视图显示和当前用户查询相关的信息，字段保存的是上一次执行的信息。具体字段信息如**表7-441**所示。

说明

查询PG_STAT_ACTIVITY时，GaussDB会根据current_user的权限来决定是否允许访问。如果current_user是初始用户或系统管理员，则可以查询所有会话信息。如果current_user是普通用户，则只能查看自己的会话信息（即session_user是当前用户的信息）。

版权所有 © 华为云计算技术有限公司

表 7-441 PG_STAT_ACTIVITY 字段

名称	类型	描述
datid	oid	用户会话在后台连接到的数据库OID。
datname	name	用户会话在后台连接到的数据库名称。
pid	bigint	后台线程id。
sessionid	bigint	会话id。
usesysid	oid	登录该后台的用户OID。
usename	name	登录该后台的用户名。
application_name	text	连接到该后台的应用名。
client_addr	inet	连接到该后台的客户端的IP地址。 如果此字段是null，它表明通过服务器机器上UNIX套接字连接客户端或者这是内部线程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
backend_start	timestamp with time zone	该会话开始的时间，即客户端连接服务器的时间。
xact_start	timestamp with time zone	当前活跃事务开始的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	当前活跃查询开始的时间， 如果state的值不是active，则这个值是上一个查询的开始时间。如果是存储过程、或函数或package，则显示的是第一个查询时间，不会随着存储过程内语句运行而改变。
state_change	timestamp with time zone	上次状态改变的时间。
waiting	boolean	当前后台是否正在等待锁。 ● t（true）： 表示是。 ● f（false）： 表示否。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
enqueue	text	语句当前排队状态。可能值是： ● waiting in queue：表示语句在排队中。 ● 空：表示语句正在运行。
state	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。 **说明** 普通用户只能查看到自己账户所对应的会话状态。即其他账户的state信息为空。例如以judy用户连接数据库后，在pg_stat_activity中查看到的普通用户joe及初始用户omm的state信息为空： SELECT datname, usename, usesysid, state,pid FROM pg_stat_activity; datname | usename | usesysid | state | pid ----------+---------+----------+--------+----------------- testdb | omm | 10 | | 139968752121616 testdb | omm | 10 | | 139968903116560 db_tpcds | judy | 16398 | active | 139968391403280 testdb | omm | 10 | | 139968643069712 testdb | omm | 10 | | 139968680818448 testdb | joe | 16390 | | 139968563377936 (6 rows)
resource_pool	name	用户使用的资源池。
query_id	bigint	查询语句的id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。 当执行存储过程时，此字段显示内容受GUC参数track_procedure_sql的影响： • 当GUC参数track_procedure_sql为on时，此字段会同时打印存储过程正在执行的语句。 • 当GUC参数track_procedure_sql为off时，此字段只打印存储过程调用语句。
connection_info	text	json格式字符串，记录当前连接数据库的驱动类型、驱动版本号、当前驱动的部署路径、进程属主用户等信息（参见GUC参数connection_info）。
global_sessionid	text	全局的会话id。
unique_sql_id	bigint	语句的unique sql id。 当执行存储过程时，此字段显示的值受GUC参数instr_unique_sql_track_type的影响： • 当GUC参数instr_unique_sql_track_type设置为ALL时，此字段显示存储过程内部执行语句的unique_sql_id。 • 当GUC参数instr_unique_sql_track_type设置为TOP时，此字段显示顶层执行语句的unique_sql_id。
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。
top_xid	xid	事务的顶层事务id。
current_xid	xid	事务的当前事务id。
xlog_quantity	bigint	事务当前使用的Xlog量，单位为字节。

7.3.13.160 PG_STAT_ACTIVITY_NG

PG_STAT_ACTIVITY_NG视图显示在当前用户所属的node group下，所有查询的相关信息。具体字段信息如**表7-442**所示。

版权所有 © 华为云计算技术有限公司

表 7-442 PG_STAT_ACTIVITY_NG 字段

名称	类型	描述
datid	oid	用户会话在后台连接到的数据库OID。
datname	name	用户会话在后台连接到的数据库名称。
pid	bigint	后台线程id。
sessionid	bigint	会话id。
global_sessionid	text	全局会话id。
usesysid	oid	登录该后台的用户OID。
usename	name	登录该后台的用户名。
application_name	text	连接到该后台的应用名。
client_addr	inet	连接到该后台的客户端的IP地址。 如果该字段取值是null，表明是通过服务器机器上UNIX套接字连接客户端或者这是内部线程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
backend_start	timestamp with time zone	该会话开始的时间，即当客户端连接服务器的时间。
xact_start	timestamp with time zone	当前活跃事务开始的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	当前活跃查询开始的时间， 如果state的值不是active，则这个值是上一个查询的开始时间。如果是存储过程、函数或package，则显示的是第一个查询时间，不会随着存储过程内语句运行而改变。
state_change	timestamp with time zone	上次状态改变的时间。
waiting	boolean	当前后台是否正在等待锁。 ● t（true）： 表示是。 ● f（false）： 表示否。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
enqueue	text	语句当前排队状态。可能值是： ● waiting in queue：表示语句在排队中。 ● 空：表示语句正在运行。
state	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。 **说明** 普通用户只能查看到自己账户所对应的会话状态。即其他账户的state信息为空。例如以judy用户连接数据库后，在pg_stat_activity中查看到的普通用户joe及初始用户omm的state信息为空： SELECT datname, usename, usesysid, state,pid FROM pg_stat_activity_ng; datname | usename | usesysid | state | pid ----------+---------+----------+--------+----------------- testdb | omm | 10 | | 139968752121616 testdb | omm | 10 | | 139968903116560 db_tpcds | judy | 16398 | active | 139968391403280 testdb | omm | 10 | | 139968643069712 testdb | omm | 10 | | 139968680818448 testdb | joe | 16390 | | 139968563377936 (6 rows)
resource_pool	name	用户使用的资源池。
query_id	bigint	查询语句的id。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。
node_group	text	语句所属用户对应的node group。

版权所有 © 华为云计算技术有限公司

7.3.13.161 PG_STAT_ALL_INDEXES

PG_STAT_ALL_INDEXES视图可用来查询当前数据库中的每个索引行，显示访问特定索引的统计。具体字段信息如表7-443所示。

索引可以通过简单的索引扫描或位图索引扫描进行使用。位图扫描中索引的输出可通过AND或OR规则进行组合，因此当使用位图扫描的时候，很难将独立堆行抓取与特定索引进行组合，因此，每一次位图扫描都会增加pg_stat_all_indexes.idx_tup_read使用索引的计数，并且增加pg_stat_all_tables.idx_tup_fetch表的计数，该操作不影响pg_stat_all_indexes.idx_tup_fetch。

表 7-443 PG_STAT_ALL_INDEXES 字段

名称	类型	描述
relid	oid	索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引的模式名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	索引上开始的索引扫描数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。

7.3.13.162 PG_STAT_ALL_TABLES

PG_STAT_ALL_TABLES视图可用来查询当前数据库中每个表的信息（包括TOAST表），显示特定表的统计信息。具体字段信息如表7-444所示。

表 7-444 PG_STAT_ALL_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表的模式名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（比如没有更新所需的单独索引）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次清理该表的时间。
last_autovacuum	timestamp with time zone	这个表上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次分析该表的时间。
last_autoanalyze	timestamp with time zone	这个表上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	这个表被清理的次数。
autovacuum_count	bigint	这个表被autovacuum清理的次数。
analyze_count	bigint	这个表被手动分析的次数。
autoanalyze_count	bigint	这个表被autovacuum守护线程分析的次数。
last_data_changed	timestamp with time zone	记录这个表上一次数据发生变化的时间（引起数据变化的操作包括对表的修改insert/update/delete/truncate和对表的分区(partition/subpartition)的修改exchange/truncate/drop，系统表不记录该字段），该列数据仅在本地CN记录。

7.3.13.163 PG_STAT_BAD_BLOCK

PG_STAT_BAD_BLOCK视图显示自节点启动后，读取数据时出现Page校验失败的统计信息。具体字段信息如**表7-445**所示。

版权所有 © 华为云计算技术有限公司

表 7-445 PG_STAT_BAD_BLOCK 字段

名称	类型	描述
nodename	text	节点名。
databaseid	integer	数据库OID。
tablespaceid	integer	表空间OID。
relfilenode	integer	文件对象id。
bucketid	smallint	一致性hash bucket id。
forknum	integer	文件类型，具体取值如下： ● 0：数据主文件。 ● 1：FSM文件。 ● 2：VM文件。 ● 3：BCM文件。
error_count	integer	出现校验失败的次数。
first_time	timestamp with time zone	第一次出现时间。
last_time	timestamp with time zone	最后一次出现时间。

7.3.13.164 PG_STAT_BGWRITER

PG_STAT_BGWRITER视图显示后端写进程活动的统计信息。具体字段信息如**表7-446**所示。

表 7-446 PG_STAT_BGWRITER 字段

名称	类型	描述
checkpoints_timed	bigint	定期执行的检查点数。
checkpoints_req	bigint	主动执行的检查点数。
checkpoint_write_time	double precision	将文件写入到磁盘时，在检查点处理部分花费的时间总量，以毫秒为单位。
checkpoint_sync_time	double precision	将文件同步到磁盘时，在检查点处理部分花费的时间总量，以毫秒为单位。
buffers_checkpoint	bigint	检查点写入的缓冲区的数量。
buffers_clean	bigint	后端写进程写入的缓冲区的数量。
maxwritten_clean	bigint	后端写进程因写入的缓冲区过多导致的清理扫描停止的次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
buffers_backend	bigint	后端直接写入的缓冲区的数量。
buffers_backend_fsync	bigint	后端自己执行fsync调用的次数（通常情况下，即使后端自己执行了这些写入动作，后端写进程也会再处理一次）。
buffers_alloc	bigint	分配的缓冲区数量。
stats_reset	timestamp with time zone	这些统计被重置的时间。

7.3.13.165 PG_STAT_DATABASE

PG_STAT_DATABASE视图显示集群中每个数据库的统计信息。具体字段信息如表7-447所示。

表 7-447 PG_STAT_DATABASE 字段

名称	类型	描述
datid	oid	数据库的OID。
datname	name	数据库的名称。
numbackends	integer	当前连接到该数据库的后端数。 这是该视图中唯一一个返回当前状态值的字段，其他字段返回的都是自上次重置之后的累计值。
xact_commit	bigint	该数据库中已经提交的事务数。
xact_rollback	bigint	该数据库中已经回滚的事务数。
blks_read	bigint	在该数据库中读取的磁盘块的数量。
blks_hit	bigint	已在缓冲区缓存中找到磁盘块的次数，因此不需要读取（只统计在缓冲区缓存找到的，不包括在操作系统的文件系统缓存中找到的）。
tup_returned	bigint	通过数据库查询返回的行数。
tup_fetched	bigint	通过数据库查询抓取的行数。
tup_inserted	bigint	通过数据库查询插入的行数。
tup_updated	bigint	通过数据库查询更新的行数。
tup_deleted	bigint	通过数据库查询删除的行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
conflicts	bigint	由于数据库恢复冲突取消的查询数量（只在备用服务器发生的冲突）。请参见PG_STAT_DATABASE_CONFLICTS获取更多信息。
temp_files	bigint	通过数据库查询创建的临时文件数量。计算所有临时文件， 无论该临时文件为什么创建（比如排序或者哈希），也不管log_temp_files参数如何设置。
temp_bytes	bigint	通过数据库查询写入临时文件的数据总量。计算所有临时文件，无论该临时文件为什么创建，也不管log_temp_files参数如何设置。
deadlocks	bigint	该数据库中检测到自上次重置统计信息以来所有的死锁数。
blk_read_time	double precision	通过数据库后端读取数据文件块花费的时间，以毫秒计算。
blk_write_time	double precision	通过数据库后端写入数据文件块花费的时间，以毫秒计算。
stats_reset	timestamp with time zone	当前状态统计被重置的时间。

7.3.13.166 PG_STAT_DATABASE_CONFLICTS

PG_STAT_DATABASE_CONFLICTS视图显示数据库冲突状态的统计信息。具体字段信息如表7-448所示。

表 7-448 PG_STAT_DATABASE_CONFLICTS 字段

名称	类型	描述
datid	oid	数据库标识。
datname	name	数据库名称。
confl_tablespace	bigint	冲突的表空间的数目。
confl_lock	bigint	冲突的锁数目。
confl_snapshot	bigint	冲突的快照数目。
confl_bufferpin	bigint	冲突的缓冲区数目。
confl_deadlock	bigint	冲突的死锁数目。

版权所有 © 华为云计算技术有限公司

7.3.13.167 PG_STAT_REPLICATION

PG_STAT_REPLICATION视图显示日志同步线程的信息，如发起端发送日志位置，接收端接收日志位置等。具体字段信息如表7-449所示。

表 7-449 PG_STAT_REPLICATION 字段

名称	类型	描述
pid	bigint	线程的PID。
usesysid	oid	用户系统id。
usename	name	用户名。
application_name	text	程序名称。
client_addr	inet	客户端地址。
client_hostname	text	客户端名。
client_port	integer	客户端端口。
backend_start	timestamp with time zone	程序启动时间。
state	text	日志同步线程的状态。 • startup：线程正在启动。 • catchup：线程正在建立备用服务器和主服务器的连接。 • streaming：线程已建立备用服务器和主服务器的连接，正在进行数据的流复制。 • backup：线程正在发送备份。 • stopping：线程正在停止。
sender_sent_location	text	发送端发给接收端的最后一个xlog位置。
receiver_write_location	text	接收端写入文件系统的最后一个xlog位置。
receiver_flush_location	text	接收端下刷到磁盘的最后一个xlog位置。
receiver_replay_location	text	接收端回放完成的最后一个xlog位置。
sync_priority	integer	同步复制的优先级（0表示异步）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sync_state	text	同步状态： • async：异步复制。 • sync：同步复制。 • potential：该备用服务器现在是异步的，但假如一个当前的同步服务器发生故障，该服务器会变成同步的。 • quorum：在同步与异步之间切换，保证备机中有大于一定数量的同步备机，同步备机数量一般为(n+1)/2-1，n为总副本个数。是否为同步备机取决于是否先接到了日志。详情可参考synchronous_standby_names参数描述。

7.3.13.168 PG_STAT_SYS_INDEXES

PG_STAT_SYS_INDEXES视图显示pg_catalog、information_schema模式中所有系统表的索引状态信息。具体字段信息如表7-450所示。

表 7-450 PG_STAT_SYS_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引的模式名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	索引上开始的索引扫描数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。

7.3.13.169 PG_STAT_SYS_TABLES

PG_STAT_SYS_TABLES视图显示pg_catalog、information_schema模式的所有命名空间中系统表的统计信息。具体字段信息如表7-451所示。

版权所有 © 华为云计算技术有限公司

表 7-451 PG_STAT_SYS_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表的模式名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（比如没有更新所需的单独索引）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	上次手动清理该表的时间（不计算VACUUM FULL）。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析这个表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	这个表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	这个表被autovacuum守护线程清理的次数。
analyze_count	bigint	这个表被手动分析的次数。
autoanalyze_count	bigint	这个表被autovacuum守护线程分析的次数。
last_data_changed	timestamp with time zone	这个表数据最近修改时间。

版权所有 © 华为云计算技术有限公司

7.3.13.170 PG_STAT_USER_FUNCTIONS

PG_STAT_USER_FUNCTIONS视图显示命名空间中用户自定义函数（函数语言为非内部语言）的状态信息。具体字段信息如表7-452所示。

表 7-452 PG_STAT_USER_FUNCTIONS 字段

名称	类型	描述
funcid	oid	函数标识。
schemaname	name	模式的名称。
funcname	name	函数名称。
calls	bigint	函数被调用的次数。
total_time	double precision	函数的总执行时长。
self_time	double precision	当前线程调用函数的总的时长。

7.3.13.171 PG_STAT_USER_INDEXES

PG_STAT_USER_INDEXES视图显示数据库中用户自定义普通表和toast表的索引状态信息。具体字段信息如表7-453所示。

表 7-453 PG_STAT_USER_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引的模式名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	索引上开始的索引扫描数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。

7.3.13.172 PG_STAT_USER_TABLES

PG_STAT_USER_TABLES视图显示所有命名空间中用户自定义普通表和toast表的状态信息。具体字段信息如表7-454所示。

版权所有 © 华为云计算技术有限公司

表 7-454 PG_STAT_USER_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表的模式名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新所需的单独索引）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	上次手动清理该表的时间（不计算VACUUM FULL）。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析这个表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	这个表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	这个表被autovacuum守护线程清理的次数。
analyze_count	bigint	这个表被手动分析的次数。
autoanalyze_count	bigint	这个表被autovacuum守护线程分析的次数。
last_data_changed	timestamp with time zone	这个表数据最近修改时间。

版权所有 © 华为云计算技术有限公司

7.3.13.173 PG_STAT_XACT_ALL_TABLES

PG_STAT_XACT_ALL_TABLES视图显示命名空间中所有普通表和toast表的事务状态信息。具体字段信息如表7-455所示。

表 7-455 PG_STAT_XACT_ALL_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表的模式名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（比如没有更新所需的单独索引）。

7.3.13.174 PG_STAT_XACT_SYS_TABLES

PG_STAT_XACT_SYS_TABLES视图显示命名空间中系统表的事务状态信息。具体字段信息如表7-456所示。

表 7-456 PG_STAT_XACT_SYS_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表的模式名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（比如没有更新所需的单独索引）。

7.3.13.175 PG_STAT_XACT_USER_FUNCTIONS

PG_STAT_XACT_USER_FUNCTIONS视图包含每个函数的执行的统计信息。具体字段信息如表7-457所示。

表 7-457 PG_STAT_XACT_USER_FUNCTIONS 字段

名称	类型	描述
funcid	oid	函数标识。
schemaname	name	模式的名称。
funcname	name	函数名称。
calls	bigint	函数被调用的次数。
total_time	double precision	函数的总执行时长。
self_time	double precision	当前线程调用函数的总的时长。

7.3.13.176 PG_STAT_XACT_USER_TABLES

PG_STAT_XACT_USER_TABLES视图显示命名空间中用户表的事务状态信息。具体字段信息如表7-458所示。

表 7-458 PG_STAT_XACT_USER_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表的模式名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（比如没有更新所需的单独索引）。

7.3.13.177 PG_STATIO_ALL_INDEXES

PG_STATIO_ALL_INDEXES视图可用来查询当前数据库中的每个索引行的信息，显示特定索引的I/O的统计信息。具体字段信息如表7-459所示。

表 7-459 PG_STATIO_ALL_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	bigint	从索引中读取的磁盘块数。
idx_blks_hit	bigint	索引命中缓存数。

7.3.13.178 PG_STATIO_ALL_SEQUENCES

PG_STATIO_ALL_SEQUENCES视图显示当前数据库中每个序列的I/O的统计信息。具体字段信息如表7-460所示。

表 7-460 PG_STATIO_ALL_SEQUENCES 字段

名称	类型	描述
relid	oid	序列OID。
schemaname	name	序列的模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列命中缓存数。

版权所有 © 华为云计算技术有限公司

7.3.13.179 PG_STATIO_ALL_TABLES

PG_STATIO_ALL_TABLES视图可用来查询当前数据库中每个表（包括TOAST表）的I/O统计信息。具体字段信息如表7-461所示。

表 7-461 PG_STATIO_ALL_TABLES 字段

名称	类型	描述
relid	oid	表OID。
schemaname	name	该表的模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表命中缓存数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	从该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓存数（如果存在）。
tidx_blks_read	bigint	从该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓存数（如果存在）。

7.3.13.180 PG_STATIO_SYS_INDEXES

PG_STATIO_SYS_INDEXES视图显示命名空间中所有系统表索引的I/O状态信息。具体字段信息如表7-462所示。

表 7-462 PG_STATIO_SYS_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	bigint	从索引中读取的磁盘块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
idx_blks_hit	bigint	索引命中缓存数。

7.3.13.181 PG_STATIO_SYS_SEQUENCES

PG_STATIO_SYS_SEQUENCES视图显示命名空间中所有序列的I/O状态信息。具体字段信息如表7-463所示。

表 7-463 PG_STATIO_SYS_SEQUENCES 字段

名称	类型	描述
relid	oid	序列OID。
schemaname	name	序列的模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列命中缓存数。

7.3.13.182 PG_STATIO_SYS_TABLES

PG_STATIO_SYS_TABLES视图显示命名空间中所有系统表的I/O状态信息。具体字段信息如表7-464所示。

表 7-464 PG_STATIO_SYS_TABLES 字段

名称	类型	描述
relid	oid	表OID。
schemaname	name	该表的模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表命中缓存数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	从该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓存数（如果存在）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tidx_blks_read	bigint	从该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓存数（如果存在）。

7.3.13.183 PG_STATIO_USER_INDEXES

PG_STATIO_USER_INDEXES视图显示命名空间中所有用户关系表索引的I/O状态信息。具体字段信息如表7-465所示。

表 7-465 PG_STATIO_USER_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名.
indexrelname	name	索引名称。
idx_blks_read	bigint	从索引中读取的磁盘块数。
idx_blks_hit	bigint	索引命中缓存数。

7.3.13.184 PG_STATIO_USER_SEQUENCES

PG_STATIO_USER_SEQUENCES视图显示命名空间中所有用户关系表序列的I/O状态信息。具体字段信息如表7-466所示。

表 7-466 PG_STATIO_USER_SEQUENCES 字段

名称	类型	描述
relid	oid	序列OID。
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列命中缓存数。

版权所有 © 华为云计算技术有限公司

7.3.13.185 PG_STATIO_USER_TABLES

PG_STATIO_USER_TABLES视图显示命名空间中所有用户关系表的I/O状态信息。具体字段信息如表7-467所示。

表 7-467 PG_STATIO_USER_TABLES 字段

名称	类型	描述
relid	oid	表OID。
schemaname	name	该表的模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表命中缓存数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	从该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓存数（如果存在）。
tidx_blks_read	bigint	从该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓存数（如果存在）。

7.3.13.186 PG_TABLES

PG_TABLES视图可用来查询数据库中每个表的有用信息。具体字段信息如表7-468所示。

表 7-468 PG_TABLES 字段

名称	类型	引用	描述
schemaname	name	PG_NAMESPACE.nspname	表的模式名。
tablename	name	PG_CLASS.relname	表名。
tableowner	name	pg_get_userbyid(PG_CLASS.relowner)	表的所有者。
tablespace	name	PG_TABLESPACE.spcname	包含表的表空间，默认为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	引用	描述
hasindexes	boolean	PG_CLASS.relhasindex	如果表上有索引（或者最近拥有）则为t，否则为f。
hasrules	boolean	PG_CLASS.relhasrules	如果表上有规则，则为t，否则为f。
hastriggers	boolean	PG_CLASS.RELHASTRIGGERS	如果表上有触发器，则为t，否则为f。
tablecreator	name	pg_get_userbyid(PG_OBJECT.creator)	表的所有者。
created	timestamp with time zone	PG_OBJECT.ctime	表的创建时间。
last_ddl_time	timestamp with time zone	PG_OBJECT.mtime	最后一次对该表执行DDL操作的时间。

7.3.13.187 PG_THREAD_WAIT_STATUS

通过PG_THREAD_WAIT_STATUS视图可以检测当前实例中工作线程（backend thread）以及辅助线程（auxiliary thread）的阻塞等待情况。具体字段信息如表7-469所示。

表 7-469 PG_THREAD_WAIT_STATUS 字段

名称	类型	描述
node_name	text	当前节点的名称。
db_name	text	数据库名称。
thread_name	text	线程名称。
query_id	bigint	查询id，对应debug_query_id。
tid	bigint	当前线程的线程号。
sessionid	bigint	当前会话id。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	父会话ID。
tlevel	integer	streaming线程的层级。
smpid	integer	并行线程的id。
wait_status	text	当前线程的等待状态。等待状态的详细信息请参见表7-470。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
wait_event	text	如果wait_status是acquire lock、acquire lwlock、wait io三种类型，此列描述具体的锁、轻量级锁、I/O的信息。否则是空。
locktag	text	当前线程正等待获取的锁的信息。
lockmode	text	当前线程正等待获取的锁的模式。包含表级锁、行级锁、页级锁下的各模式。
block_sessionid	bigint	阻塞当前线程获取锁的会话标识。
global_sessionid	text	全局会话id。

wait_status列的等待状态如表7-470所示。

表 7-470 等待状态列表

wait_status值	含义	出现原因	调优建议
wait cmd	等待完成读取网络通信包。	本端等待对端的下一条指令。	通常情况下，相比其他操作，网络通信本身耗时较长。如果发现某些连接长时间处于该状态，可以结合pg_stat_activity、gs_stack、dbe_perf.pooler_status、pg_thread_wait_status等视图分析对端长时间未发送指令的原因。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
wait pooler get conn	等待pooler完成获取连接。	本端节点尝试从连接池获取与其他节点的连接，如果连接池里没有符合条件的连接，则尝试创建新的连接。如果连接能直接从连接池中复用，这种情况下获取连接的耗时不会很大。如果业务量大、且连接重新创建的情况较多（如连接池中连接被清理、连接因出错而直接被销毁没有归还到连接池中等），会出现获取连接耗时较长。	根据业务场景调整GUC参数minimum_pool_size、pooler_maximum_idle_time，并结合日志排查当前连接情况，如果日志中出现大量连接销毁的日志则需要分析原因。
wait pooler abort conn	等待pooler完成终止连接。	由clean connection触发，用于等待符合条件的事务退出、释放连接。	清理连接前停止业务。
wait pooler clean conn	等待pooler完成清理连接。	由clean connection触发，用于等待符合条件的线程退出、释放连接。	清理连接前停止业务。
pooler create conn	等待pooler建立连接，当前正在与nodename指定节点建立连接，且仍有N个连接等待建立。	与指定节点建连，如果网络状态较差、或对端业务较忙，会影响建连耗时。	检查网络状态，或降低业务量。
set cmd	在连接上执行SET/RESET/TRANSACTION BLOCK LEVEL PARA SET/SESSION LEVEL PARA SET，当前正在nodename指定节点上执行。	正在往指定节点执行SET/RESET/TRANSACTION BLOCK LEVEL PARA SET/SESSION LEVEL PARA SET。	检查网络状态，或降低业务量。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
cancel query	取消某连接上正在执行的SQL语句。	本端向对端发送取消语句指令后，等待对端返回执行结果。	检查网络状态，或降低业务量。
stop query	停止某连接上正在执行的查询。	本端向对端发送取消语句指令后，等待对端返回执行结果。	检查网络状态，或降低业务量。
wait node	等待接收与某节点的连接上的数据，当前正在等待nodename节点plevel线程的数据，且仍有N个连接的数据待返回。如果状态包含phase信息，则可能的阶段状态有： ● begin：表示处于事务开始阶段。 ● commit：表示处于事务提交阶段。 ● rollback：表示处于事务回滚阶段。	本端发送指令后，等待对端返回执行结果。	通常情况下，相比其他操作，网络通信本身耗时较长。如果发现某些连接长时间处于该状态，可以结合pg_stat_activity、gs_stack、dbe_perf.pooler_status、pg_thread_wait_status等视图分析对端长时间未发送指令的原因。
wait transaction sync	等待xid指定事务同步。	等待本地事务完成。	修改transaction_sync_naptime参数。
wait wal sync	等待特定lsn的WAL LOG完成到备机的同步。	开启了主备日志强同步。	修改为本地提交。
wait data sync	等待完成数据页到备机的同步。	开启了数据页强同步到从备。	关闭混合流复制功能enable_mix_replication。
wait wal flush	主机中等待Xlog落盘以及DCF模式下与备机达成共识，备机中等待刷新完最小恢复点以及DCF模式下跟主机达成共识。	等待日志落盘或刷新最小恢复点。	检查I/O性能。
wait data sync queue	等待把行存的数据页放入同步队列。	开启了数据页强同步从备。	关闭混合流复制功能enable_mix_replication。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
flush data	等待向网络中nodename指定节点的plevel对应线程发送数据。如果状态包含phase信息，则可能的阶段状态为wait quota，即当前通信流正在等待quota值。	正在向其他节点发送数据。	检查网络状态，或降低业务量。
wait reserve td	用于在对Uheap页面进行DML时分配Uheap页面的tdslot。	对Ustore表执行DML。	减少对Ustore表的并发DML操作。
wait td rollback	用于在对Uheap页面tdslot上的待回滚的事务做页内回滚。	对Ustore表执行DML时无直接可使用的tdslot，需要释放abort且未回滚事务的tdslot。	减少对Ustore表的并发DML操作。
wait available td	用于等待Uheap和Ubtree页面的空闲td。	对Ustore表进行DML时，Uheap、Ubtree页面上暂时无可分配的td。	减少对Ustore表的并发DML操作。
wait transaction rollback	用于等待某事务ID对应的undo回滚完成。	对Ustore表进行DML时，事务abort。	减少对Ustore表进行DML操作的事务abort。
prune table	等待堆表清理历史删除数据。	触发了页面清理。	减少Ustore表的并发DML。
prune index	等待索引清理历史删除数据。	执行vacuum。	减少并行vacuum操作。
stream get conn	初始化stream flow时，等待与nodename节点的consumer对象建立连接，且当前有N个待建连对象。	发送建连请求，等待对端返回结果。	在对端通过PG_COMM_RECV_STREAM检查逻辑连接状态，或检查网络状态。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
wait producer ready	初始化stream flow时，等待每个producer都准备好，当前正在等待nodename节点plevel对应线程的producer对象准备好，且仍有N个producer对象处于等待状态。	stream consumer等待producer完成执行并接收其发送的数据。	检查网络状态，或降低业务量。
synchronize quit	stream plan结束时，等待stream线程组内的线程统一退出。当wait_status中显示形如“synchronize quit, total *m*, remain *n*”的内容时，表示共有m个并发任务，其中仍存在n个并发任务未处于统一退出状态。	等待stream线程组内的线程统一退出。	stream计划和smp计划的SQL均会出现该等待事件，可通过增加非stream计划的hint或关闭enable_stream_operator功能规避。
wait stream group destroy	等待销毁stream node group。	stream plan结束时，等待销毁stream node group。	stream计划和smp计划的SQL均会出现该等待事件，可通过增加非stream计划的hint或关闭enable_stream_operator功能规避。
wait stream thread ready	stream producer阻塞，直至其线程id被成功设置。	stream producer未成功刷新线程id。	stream计划和smp计划的SQL均会出现该等待事件，可通过增加非stream计划的hint或关闭enable_stream_operator功能规避。
wait create stream thread	等待stream线程启动。	由于资源上限等原因，等待stream线程启动。	stream计划和smp计划的SQL均会出现该等待事件，可通过增加非stream计划的hint或关闭enable_stream_operator功能规避。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
wait active statement	等待作业执行，正在资源负载管控中。	超过并发上限。	减少并发操作。
wait memory	等待内存获取。	有线程处于等待mem状态	-
Sort	当前是Sort算子做排序。	1.查询的数据量大； 2.计划不合理，如过滤条件没有下推。	尝试更改SQL语句调整执行计划。
Sort - fetch tuple	当前是Sort算子做排序，fetch tuple表示Sort算子正在获取tuple。	1.查询的数据量大； 2.计划不合理，比如大小表Join的顺序不合理。	尝试更改SQL语句调整执行计划。
Sort - write file	当前是Sort算子做排序，write file表示Sort算子正在将数据写入磁盘。	1.查询的数据量大； 2.计划不合理，如过滤条件没有下推； 3.由于内存设置较小，或其他线程占用，导致当前业务线程内存不够。	1.尝试更改SQL语句调整执行计划； 2.调大内存。
Material	当前是Material算子。	1.查询的数据量大； 2.计划不合理，比如过滤条件没有下推。	1.尝试更改SQL语句调整执行计划； 2.调大内存。
Material - write file	当前是Material算子，write file表示Material算子正在将数据写入磁盘。	1.查询的数据量大； 2.计划不合理，如过滤条件没有下推； 3.由于内存设置较小，或其他线程占用，导致当前业务线程内存不够。	1.尝试更改SQL语句调整执行计划； 2.调大内存。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
HashJoin - build hash	当前是HashJoin算子，主要关注耗时的执行阶段。 ● build hash：表示当前HashJoin算子正在建立哈希表。	1.查询的数据量大； 2.计划不合理，比如大小表Join的顺序不合理。	尝试更改SQL语句调整执行计划，添加对应的索引，加快Hash Join构建内表Hash table的速度。
HashJoin - write file	当前是HashJoin算子，主要关注耗时的执行阶段。 ● write file：表示当前HashJoin算子正在将数据写入磁盘。	1.查询的数据量大； 2.计划不合理，如大小表Join的顺序不合理； 3.由于内存设置较小，或其他线程占用，导致当前业务线程内存不够。	1.尝试更改SQL语句调整执行计划； 2.调大内存。
HashAgg - build hash	当前是HashAgg算子，主要关注耗时的执行阶段。 ● build hash：表示当前HashAgg算子正在建立哈希表。	1.查询的数据量大； 2.计划不合理，比如大小表Join的顺序不合理。	1.尝试更改SQL语句调整执行计划； 2.调大内存。
HashAgg - write file	当前是HashAgg算子，主要关注耗时的执行阶段。 ● write file：表示当前HashAgg算子正在将数据写入磁盘。	1.查询的数据量大； 2.计划不合理，比如大小表Join的顺序不合理； 3.由于内存设置较小，或其他线程占用，导致当前业务线程内存不够。	1.尝试更改SQL语句调整执行计划； 2.调大内存。
HashSetop - build hash	当前是HashSetop算子，主要关注耗时的执行阶段。 ● build hash：表示当前HashSetop算子正在建立哈希表。	1.查询的数据量大； 2.计划不合理，比如大小表Join的顺序不合理。	尝试更改SQL语句调整执行计划。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
HashSetop - write file	当前是HashSetop算子，主要关注耗时的执行阶段。 • write file：表示当前HashSetop算子正在将数据写入磁盘。	1.查询的数据量大； 2.计划不合理，比如大小表Join的顺序不合理； 3.由于内存设置较小，或其他线程占用，导致当前业务线程内存不够。	1.尝试更改SQL语句调整执行计划； 2.调大内存。
create index	当前正在创建索引。	创建索引。	尽量避免在非关键列或区分性较低的列建立索引。
analyze	当前正在对表relname执行analyze。如果状态包含phase信息，则为autovacuum，表示是数据库自动开启AutoVacuum线程执行的analyze分析操作。	触发autoanalyze或用户手动下发的analyze。	调整default_statistics_target减小采样率，或调高autovacuum_analyze_scale_factor和autovacuum_analyze_threshold，减少analyze的触发间隔。
vacuum	当前正在对表relname执行vacuum。如果状态包含phase信息，则为autovacuum，表示是数据库自动开启AutoVacuum线程执行的vacuum清理操作。	执行vacuum或analyze。	取消vacuum或analyze操作，对于autovacuum线程，可通过减少页面并发DML实现。
vacuum full	当前正在对表relname执行vacuum full清理。	执行vacuum full。	避免对表执行重复无效的update操作，导致表膨胀。
vacuum gpi	vacuum或autovacuum流程中对分区表global partition index清理。	分区表上有GPI索引，执行过分区表DDL后，GPI索引产生的死元组依赖autovacuum/vacuum做异步清理。	减少分区表DDL频率，或者使用local索引。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
gtm connect	等待与GTM建连。	正在与GTM建连。	检查GTM和网络状态。
gtm get csn	等待从GTM获取csn。	事务commit时从GTM获取csn。	检查GTM和网络状态。
gtm get snapshot	等待从GTM获取事务快照snapshot。	事务commit时从GTM获取snapshot。	检查GTM和网络状态。
gtm create sequence	等待GTM创建sequence。	正在发消息给GTM创建sequence。	检查GTM和网络状态。
gtm alter sequence	等待GTM修改sequence。	正在发消息给GTM修改sequence。	检查GTM和网络状态。
gtm get sequence val	等待从GTM获取sequence的下一个值。	正在发消息给GTM获取sequence	检查GTM和网络状态。
gtm set sequence val	等待GTM设置sequence的值。	正在发消息给GTM设置sequence。	检查GTM和网络状态。
gtm drop sequence	等待GTM删除sequence。	正在发消息给GTM删除sequence。	检查GTM和网络状态。
gtm rename sequence	等待GTM重命名sequence。	正在发消息给GTM重命名sequence。	检查GTM和网络状态。
gtm set disaster cluster	等待在GTM设置容灾集群的静态配置信息。	搭建容灾集群，主集群和容灾集群倒换；或GTM通信异常。	正常情况时间小于1s，若长时间处于此等待事件，需检查GaussDB与GTM通信是否异常。
gtm get disaster cluster	等待从GTM获取容灾集群的静态配置信息。	搭建容灾集群，主集群和容灾集群倒换，主集群节点的拓扑结构发生变化；或GTM通信异常。	正常情况时间小于1s，若长时间处于此等待事件，需检查GaussDB与GTM通信是否异常。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
gtm del disaster cluster	等待在GTM删除容灾集群的静态配置信息。	搭建容灾集群，主集群和容灾集群倒换；或GTM通信异常。	正常情况时间小于1s，若长时间处于此等待事件，需检查GaussDB与GTM通信是否异常。
gtm set standby read info	等待GTM写入备机读信息。	备机从gtm中读取csn信息以及commit time时。	检查GTM和网络状态。
gtm get standby read info	等待从GTM获取备机读信息。	备机从gtm中获取上次信息。	检查GTM和网络状态。
gtm set target standby info	新特性预留。	新特性预留。	新特性预留。
gtm get target standby info	新特性预留。	新特性预留。	新特性预留。
gtm reset standby read csn	新特性预留。	新特性预留。	新特性预留。
wait sync consumer next step	消费者等待下一次迭代。	RecursiveUnion算子执行过程中，等待下一次迭代的数据；仅在分布式下会出现。	设置enable_stream_recursive=off。
wait sync producer next step	生产者等待下一次迭代。	RecursiveUnion算子对应的生产者执行过程中，等待下一次迭代的开始；仅在分布式下会出现。	设置enable_stream_recursive=off。
gtm set consistency point	等待在GTM设置容灾集群的全局一致性快照。	容灾集群正常运行过程中，CN收集并计算一致性快照，上报给GTM。	正常情况时间小于1s，若长时间处于此等待事件，需检查GaussDB与GTM通信是否异常。
wait sync bgworkers	等待并行创建索引的子线程完成本地扫描和排序。	执行并行创建索引。	关闭并行创建索引。
flush local buffer for GTT	等待本地buffer刷脏完成。	等待本地buffer刷脏。	减少业务量，检查I/O性能。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
standby get snapshot	并行回放备机读场景下等待获取快照。	并行回放模式下，备机上的查询在获取快照时，如果备机正在回放页面清理类日志，需要等相关的日志回放完成，否则可能会出现数据不一致的问题。	主机减少vacuum的执行频率，或者开启极致RTO回放模式。
recovery conflict with lock	备机回放八级锁日志与查询发生冲突，等待查询结束。	主机执行DDL。	主机减少DDL的执行频率。
recovery conflict with snapshot	备机回放页面清理日志与查询发生冲突，等待查询结束。	主机执行vacuum。	主机减少vacuum的执行频率。
recovery conflict with tablespace	备机回放删除tablespace日志与查询发生冲突，等待查询结束。	主机执行删除tablespace。	主机减少DDL的执行频率。
recovery conflict with database	备机回放删除database日志与查询发生冲突，等待查询结束。	主机执行删除database。	主机减少DDL的执行频率。
recovery conflict with truncate	备机回放文件truncate日志与查询发生冲突，等待查询结束。	主机执行vacuum产生的文件truncate日志。	主机减少vacuum的执行频率。
recovery conflict with standby query timeout	极致RTO备机读，查询超过时间阈值，查询取消。	查询超过最大执行时间。	减少查询的时间。
recovery conflict with force recycle	极致RTO备机读，发生强制回收，查询取消。	极致RTO备机读历史文件空间占用超过阈值。	调大GUC参数max_standby_base_page_size。
recovery conflict with buffer pin	回放线程在解决buffer pin类型的查询与回放冲突。	在数据库恢复过程与缓冲区锁定发生冲突。	优化缓冲区管理策略，减少恢复操作与缓冲区锁定的冲突。
wait io control	等待作业执行，正在I/O管控中。	I/O操作被系统管控机制限制，如I/O资源配额或优先级控制。	检查和调整I/O管控策略，确保关键I/O操作能够及时执行。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
wait gs_sleep	等待服务器线程延迟时间。	数据库内部的调度机制，用于控制线程的执行节奏。	检查数据库的配置参数，如是否有必要调整线程的调度策略。
wait vacuum delay	等待vacuum延迟时间。	自动vacuum进程延迟执行，或手动vacuum操作等待合适的执行窗口。	调整vacuum的执行策略，如增加自动vacuum的频率或在系统负载较低时执行。
wait seq scan	seq scan执行时间。	数据库执行顺序扫描操作，由于缺乏索引或查询优化不足。	优化查询语句，增加适当的索引以减少全表扫描，提高查询效率。
wait index scan	index scan执行时间。	数据库执行索引扫描操作，由于索引设计不当或查询条件不匹配索引。	检查索引的有效性和查询条件，优化索引结构，提高索引扫描效率。
wait checkpoint start	checkpoint开始阶段时间。	数据库执行checkpoint操作的开始阶段，可能由于I/O性能瓶颈。	优化I/O性能，调整checkpoint的频率和时间，减少I/O等待。
wait checkpoint done	checkpoint结束阶段时间。	数据库执行checkpoint操作的结束阶段，可能由于I/O性能瓶颈。	优化I/O性能，调整checkpoint的配置以减少等待时间。
wait xact start command	xact开始时间。	事务开始阶段，由于事务日志瓶颈或锁等待。	优化事务日志的处理，减少事务开始的延迟。
wait xact commit command	xact提交时间。	事务提交阶段，由于锁等待或事务日志瓶颈。	优化事务处理流程，减少锁竞争，提高事务日志处理效率。
Accept client conn - Total Time	等待客户端连接时间。	用于统计客户端连接时间。	减少业务并发。
Accept client conn - ThrdPool - add epoll	等待监听事件加入epoll。	用于统计加入监听事件。	减少业务并发。

版权所有 © 华为云计算技术有限公司

wait_status值	含义	出现原因	调优建议
Accept client conn - ThrdPool - wait worker	等待worker处理会话。	用于统计等待到worker的时间。	减少业务并发。
Accept client conn - ThrdPool - init session	等待会话初始化。	用于统计会话初始化时间。	减少业务并发。
Accept client conn - Worker - init proc	等待proc初始化。	用于统计proc初始化时间。	减少业务并发。
Accept client conn - Worker - init session	等待会话初始化。	用于统计会话初始化时间。	减少业务并发。
security audit write pipe	等待将审计日志写入管道。	安全审计日志系统在写入日志时遇到瓶颈。	优化审计日志系统，增加资源或调整审计策略，减少写入等待。
wait ugin cleanup proc	等待UGIN索引清空缓冲区。	正在清理UGIN索引缓冲区。	清理UGIN索引缓冲区时调用，适当减少gin_pending_list_limit大小。
wait gin cleanup proc	等待GIN索引清空缓冲区。	正在清理GIN索引缓冲区。	清理GIN索引缓冲区时调用，适当减少gin_pending_list_limit大小。
wait fetch undo record	等待读取目标UNDO记录。	事务同步或异步回滚时。	减少DML操作的事务abort。
deadlock check	等待死锁检测完成。	等锁时间超过死锁检测配置时间，触发死锁检测。	优化业务流程减少锁冲突，或增加deadlock_timeout避免频繁死锁检测。
invalid wait event	意料之外的等待事件。	发生了不应该发生的等待事件。	联系华为技术支持。

当wait_status为acquire lwlock、acquire lock或wait io时，表示有等待事件。正在等待获取wait_event列对应类型的轻量级锁、事务锁或者正在进行I/O操作。

其中，wait_status值为acquire lwlock（轻量级锁）时对应的wait_event等待事件类型与描述信息如**表7-471**所示。（wait_event为extension时，表示此时的轻量级锁是动态分配的锁，未被监控。）

版权所有 © 华为云计算技术有限公司

表 7-471 轻量级锁等待事件列表

wait_event类型	类型描述	出现原因	调优建议
OidGenLock	用于避免不同线程产生相同的OID。	分配OID并发量大。	减少并发操作。
XidGenLock	用于避免两个事务获得相同的XID。	分配XID并发量大。	减少并发DDL、DML等操作。
ProcArrayLock	用于避免并发访问或修改ProcArray共享数组。	当前系统线程数量大。	减少并发线程数量。
SInvalReadLock	用于避免与清理失效消息并发执行。	读取失效队列的失效消息时，加读锁。	减少DDL执行频率，降低失效消息发送次数。
SInvalWriteLock	用于避免与其它写失效消息、清理失效消息并发执行。	插入失效队列的失效消息时，或对失效队列有非只读操作时，加写锁。	减少DDL执行频率，降低失效消息发送次数。
WALBufMappingLock	保护共享内存中各个WAL缓存页面与lsn偏移的映射关系，用于避免将记录插入未初始化的WAL文件。	插入记录与初始化WAL文件并发。	增大wal_buffers设置。
WALWriteLock	用于避免并发WAL写盘。	并发WAL文件写盘。	减少业务量，检查I/O性能。
ControlFileLock	用于避免pg_control文件的读写并发、写写并发。	数据库初始化，checkpoint、备份恢复、日志归档、主备同步等场景，并发访问pg_control文件。	减少并发访问pg_control文件。
CheckpointLock	用于避免多个checkpoint并发执行。	触发checkpoint。	减小并发。
MultiXactGenLock	用于串行分配唯一MultiXact id。	分配MultiXact id并发量大。	减少并发DML等操作。
MultiXactOffsetControlLock	用于避免对pg_multixact/offset的写写并发和读写并发。	对pg_multixact/offset并发量过大。	减小并发。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
MultiXactMemberControlLock	用于避免对pg_multixact/members的写写并发和读写并发。	对pg_multixact/members并发量过大。	减小并发。
RelCacheInitLock	用于失效消息场景对init文件进行操作时加锁。	初始化relcache时，或失效relcache init file时，对init文件加锁。	避免频繁销毁创建线程。
CheckpointerCommLock	用于向checkpointer发起文件刷盘请求场景，需要串行的向请求队列插入请求结构。	脏页落盘、shrink、truncate操作等产生脏页后，收集sync请求。	减少此类操作的并发执行。
TablespaceCreateLock	用于确定tablespace是否已经存在。	drop或创建表空间。	减少并发drop或create表空间。
BtreeVacuumLock	用于防止vacuum清理B-tree中还在使用的页面。	执行vacuum。	减少并行vacuum操作。
AutovacuumLock	用于串行化访问autovacuum worker数组。	Autovacuum Launcher正在调度worker线程。	调整Autovacuum worker线程信息，无需调优。
AutovacuumScheduleLock	用于串行化分配需要vacuum的table。	Autovacuum线程判断是否表需要清理。	减少Astore表的并发DML操作。
AutoanalyzeLock	用于获取和释放允许执行Autoanalyze的任务资源。	在生成计划时，如果基表没有收集过统计信息，则会进行自动收集。单个节点只允许同时存在10个线程进行自动统计信息收集。本轻量锁用于控制统计信息自动收集的资源分配。	如果本轻量锁存在大量冲突，说明业务库中长期大量存在reltuples/relpages为0的表，无需进行自动统计信息收集。可以关闭autoanalyze参数进行调优。
SyncScanLock	用于确定heap扫描时某个relfilenode的起始位置。	执行synchronized sequential scan。	减少并发扫描操作。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
BarrierLock	用于保证当前只有一个线程在创建barrier。	有两个线程同时在创建barrier。	减少create barrier操作。
NodeTableLock	用于保护存放CN和DN节点信息的共享结构。	分布式扩容/缩容阶段，调用pgxc_pool_reload函数，对节点pool中缓存的集群拓扑结构进行刷新，需要加排他锁，此时分布式语句解析和执行阶段的读锁会被阻塞。	将集群扩容放在业务波谷期进行，减少业务冲突。
PoolerLock	用于保证两个线程不会同时从连接池里取到相同的连接。	加锁从连接池中获取连接。	通常情况下，业务稳定运行期间，不会触发此等待事件。如果频繁触发请降低业务量。
AlterPortLock	用于保护协调节点更改注册端口号的操作。	线程初始化时会进行端口号注册。	通常情况下，业务稳定运行期间，不会触发此等待事件。如果频繁触发请排查后台线程是否频繁拉起。
RelationMappingLock	用于等待更新系统表到存储位置之间映射的文件。	并发创建pg_filenode.map。	减少对pg_filenode.map的并发访问。
SerializableXactHashLock	用于避免对于可串行事务共享结构的写写并发和读写并发。	并发访问可串行事务。	减少并发访问可串行事务。
SerializableFinishedListLock	用于避免对于已完成可串行事务共享链表的写写并发和读写并发。	并发访问可串行事务。	减少并发访问可串行事务。
SerializablePredicateLockListLock	用于保护对于可串行事务持有的锁链表。	并发访问可串行事务。	减少并发访问可串行事务。
OldSerXidLock	用于保护记录冲突可串行事务的结构。	并发访问可串行事务。	减少并发访问可串行事务。
FileStatLock	用于保护存储统计文件信息的数据结构。	存储读取文件时调用pgstat上报统计信息。	内置统计信息数组有固定大小，会很快达到限制，无需规避。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
SyncRepLock	用于在主备复制时保护Xlog同步信息。	等待备机返回Xlog刷盘信息。	与主备间网络延迟，备机I/O性能相关。
DataSyncRepLock	用于在主备复制时保护数据页同步信息。	等待备机返回数据页刷盘信息。	关闭混合流复制功能enable_mix_replication。
ExtensionConnectorLibLock	1. 初始化ODBC连接场景，在加载与卸载特定动态库时进行加锁。（EC弃用） 2. 异构DBLINK用于初始化OCI连接场景，在加载与卸载OCI相关动态库时进行加锁。	使用异构DBLINK连接Oracle数据库。	不使用异构DBLINK。
GTMHostInfoLock	用于避免并发访问或者修改GTM主机信息。	并发访问GTM主机。	检查GTM和网络状态。
ReplicationSlotAllocationLock	用于主备复制时保护主机端的流复制槽的分配。	创建或者清理复制槽，修改复制槽状态。	仅创建或者清理复制槽，修改复制槽状态时会调用，无需调优。
ReplicationSlotControlLock	用于主备复制时避免并发更新流复制槽状态。	修改复制槽状态。	仅修改复制槽状态时会调用，无需调优。
FullBuildXlogCopyStartPtrLock	控制全量build起点位置的并发访存，用于保护全量build中Xlog复制的操作。	全量build。	用于build操作，无需调优。
LogicalReplicationSlotPersistentDataLock	控制逻辑复制槽持久化的并发顺序，用于保护逻辑复制过程中修改逻辑复制槽位的数据。	逻辑复制槽数据需要修改。	仅修改逻辑复制槽数据时需要调用，无需调优。
ResourcePoolHashLock	用于避免并发访问或者修改资源池哈希表。	读取或更新资源池哈希表时会申请锁。	增删查改资源池时会申请该锁，业务高峰期也会频繁申请该锁，可以减少此两种操作，降低锁等待的可能。
WorkloadCGroupHashLock	用于避免并发访问或者修改Cgroup信息构成的哈希表。	读取或更新Cgroup信息哈希表时会申请锁。	减轻业务压力。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
OBSGetPathLock	用于避免对obs路径的写写并发和读写并发。	-	-
WorkloadUserInfoLock	用于避免并发访问或修改负载管理的用户信息哈希表。	读取或更新负载管理用户信息哈希表时会申请锁。	减轻业务压力。
WorkloadIOUtilLock	用于保护记录iostat、CPU等负载信息的结构。	读取或更新IOSTAT、CPU等负载信息时会申请锁。	减轻业务压力。
WorkloadNodeGroupLock	用于避免并发访问或者修改内存中的nodegroup信息构成的哈希表。	读取或更新nodegroup信息时会申请锁。	减轻业务压力。
JobShmemLock	用于MPP兼容ORACLE定时任务功能中保护定时读取的全局变量。	定时任务scheduler线程与worker线程访问定时任务列表的锁。	关闭定时任务。
OBSRuntimeLock	用于获取环境变量，如GASSHOME。	-	-
LLVMDumpIRLock	用于导出动态生成函数所对应的汇编语言。	开启enable_codegen_print参数后，触发codegen执行时会出现该等待事件。	设置enable_codegen_print=off（默认为off，通常在问题定位时才开启）。
LLVMParseIRLock	用于在查询开始时，从IR文件中编译并解析预先编译好的IR函数。	查询触发codegen执行时，会出现该等待事件，通常等待时间非常短暂。	设置enable_codegen=off。
RPNumberLock	已废弃。	已废弃。	已废弃。
ClusterRPLock	已废弃。	已废弃。	已废弃。
WaitCountHashLock	用于保护用户语句计数功能场景中的共享结构。	在更新或查询等待计数哈希表时，可能因锁竞争而导致等待。	优化等待计数哈希表的访问逻辑，减少锁竞争，或者优化查询性能。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
InstrWorkloadLock	保护负载管理统计信息的内存hash表访问。	读取或更新workload事务统计信息的哈希表时会申请锁。	可以通过GUC参数enable_resource_track关闭workload事务统计功能。
CBMParseXlogLock	cbm解析Xlog时的保护锁，控制cbm文件的并发访存。	有两个线程在同时使用该锁。	重试该操作。
DelayDDLLock	控制删除类DDL和延迟备份功能的并发顺序。	调用了延迟DDL的操作。	等待延迟DDL操作完成后，执行disable delay DDL操作。
BadBlockStatHashLock	保护共享损坏页面统计哈希表的并发访存。	在更新或查询坏块统计信息时，可能因锁竞争而导致等待。	优化对坏块统计信息的访问频率，减少锁竞争，或者增加处理坏块的并行度。
RowPageReplicationLock	控制行存页面复制的并发顺序，用于管理行存储的数据页复制。	-	-
RcvWriteLock	控制wal receiver writer的并发顺序，防止并发调用WalDataRcvWrite。	写入WAL数据。	减少业务量，检查I/O性能。
InstanceTimeLock	保护全局变量g_instance.stat_cxt.gInstanceTimeInfo，用于统计节点级别时间模型。	刷新或读取节点级别时间模型信息。	节点级别时间模型指标统计，无需规避。
PercentileLock	用于保护全局percentileBuffer。	读取或更新percentile统计数据或结果的全局缓存时会申请锁。	关闭percentile功能或适当增大计算周期。
XlogRemoveSegLock	控制最新删除WAL文件信息的并发访存，保护Xlog段文件的回收操作。	回放checkpoint的Xlog。	减少checkpoint。
DnUsedSpaceHashLock	用于更新会话对应的空间使用信息。	读取或更会话对应的空间使用信息哈希表时会申请锁。	减少会话。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
CsnMinLock	控制csn min信息的并发访存。	当前系统线程数量大。	减少并发线程数量。
GPCClearLock	用于保护全局Plan Cache hash表的清除操作。	GPC清理待淘汰计划。	属于GPC正常清理取锁操作。可通过视图dbe_perf.global_plancache_clean手动清理，或关闭enable_global_plancache功能。
CaptureViewFileHashLock	开启性能视图采集时，保护哈希表的并发访存。	同时对多个不同的表进行采集。	主要用于慢SQL定位使用，无需规避。
UndoZoneLock	锁定UndoZone的bitmap，保护undoZone的并发访存。	每个业务线程需要分配一个UndoZone，UndoZone的分配通过bitmap来管理。分配时需要对bitmap加锁来确保每个线程都分配到了不同的UndoZone。	减少对Ustore表的DML操作的并发数。
RollbackReqHashLock	对异步回滚哈希表加锁，管理共享内存中存储回滚请求信息的哈希表的并发访存。	异步回滚机制在后台执行回滚，不影响用户继续执行其他语句。异步回滚时需先添加异步回滚任务到一个任务哈希表中。对哈希表操作时需加锁以防止并发问题。	尽量减少回滚事务。
HypoIndexLock	虚拟索引创建、删除、重置等动作中使用的轻量级锁。	避免多会话下global级别虚拟索引的并发增删导致地址访问异常等。	默认不开启，该特性在SQL调优阶段使用，不影响数据库性能。
WALWritePaxosLock	保护向paxos复制组件写wal日志的并发顺序。	避免多线程并发写导致Xlog错乱。	一般只有Xlog刷盘线程调用无并发、WAL BUF满等情况下会有线程临时调用需要该锁保护，不影响性能。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
BlockchainVersionLock	账本数据库中控制全局区块号的并发读写。	在账本数据库的防篡改表特性中，为了保证全局区块表区块号的唯一性，需要在每一条记录生成的时候加锁生成递增的区块号。	为避免频繁对账本表进行增删改操作，可以根据业务需求调整防篡改表的范围，并采用分库策略以降低对全局区块表的并发修改。
GlobalPrevHashLock	账本数据库中控制全局校验哈希的并发读写。	在账本数据库的防篡改表特性中，为了保证全局区块表的记录能够与上一行生成链式篡改校验信息，需要在每条记录生成时加锁来生成连续的哈希值，加锁机制可以保证并发写入的安全性。	为避免频繁对账本表进行增删改操作，可以根据业务需求调整防篡改表的范围，并采用分库策略以降低对全局区块表的并发修改。
PldebugLock	用于存储过程调试功能中锁处理操作。	pldebugger进行turnon、turnoff时加锁。	-
LocalBarrierBufferLock	控制本地barrier文件修改，避免并发更新。	同时对本地barrier进行更新。	等待正在更新的操作执行完成。
RoleIdLock	控制pg_user_status系统表的并发读写。	提供密码保护策略。用户使用错误密码登录时，会更新系统表pg_user_status，记录失败次数和更新锁定状态。保证数据库账户安全。	使用错误密码登录是不安全的行为，可能是密码被修改，但用户未能感知；或者存在恶意用户不断尝试数据库密码。数据库管理员可定期检查所有用户的锁定状态，并分析被锁定账户，减少错误密码登录的次数，有以下几种情况： 1. 数据库遭受恶意攻击：管理员应及时分析原因，找出攻击源并采取相应措施。 2. 密码被修改但用户未察觉：提醒用户及时更新密码，以确保账户安全。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
InstrStmtTrackCtlLock	在动态开启全量SQL时, 保护哈希表的并发访存。	动态抓取FullSQL场景中，高频率对不同uniqueQueryId进行track，或是对同一uniqueQueryId频繁track和untrack。	主要用于慢SQL定位使用，无需规避。
HashUidLock	保护Uid的增删改查。	访问带有用户的表。	-
ParallelDecodeLock	并行解码分配全局槽位用锁，保护并行解码的并发顺序。	使用并行解码。	减少并行解码业务的启动。
XLogMaxCSNLock	保护容灾模式下本地最大可恢复CSN信息。	当前系统线程数量大。	减少并发线程数量。
DisasterCacheLock	容灾集群节点映射表的锁，保护共享内存中容灾信息缓存的并发访存。	搭建容灾集群，主集群和容灾集群倒换，主集群节点的拓扑结构发生变化，重新创建节点映射关系。	正常场景，无需调优。
MaxCSNArrayLock	保护共享内存中各个分片备机CSN恢复进度信息。	容灾读引入。	-
RepairBadBlockStatHashLock	保护共享内存中损坏页面哈希表的并发访存。	CRC损坏/LSN损坏页面被检测到，记录页面损坏信息。	避免产生坏块。
ReplicationOriginLock	发布订阅创建、删除、变更状态用锁。	使用发布订阅。	减少发布订阅功能的启动。
LogicalRepWorkerLock	发布订阅后台回放线程启动、停止、关联槽位用锁。	使用发布订阅。	减少发布订阅功能的启动。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
XGBoostLibLock	DB4AI特性调用xgboost库时启用的锁。	xgboost开源库与openElur系统不兼容，需要动态调用库。dlopen加载时需要加锁。	xgboost库的加载属于特殊事件的被动调用。在未进行DB4AI相关操作（如创建模型、执行预测任务等）时，该锁不会影响数据库性能。
NgroupDestoryLock	对于保护nodegroup哈希表并发修改加锁。	pgxc_pool_reload等操作清空整个哈希表时加X锁，与其他S锁互斥发生等待事件。	仅在进行pgxc_pool_reload或其他涉及nodegroup操作时会出现，其他场景不会发生此等待事件。
SQLAdvisorLock	用于管理分布列推荐中hash表的并发访存。	-	-
AuditIndexFileLock	控制审计日志index文件的并发读写。	审计日志的管理依赖于内存中审计索引信息。审计线程中审计日志的老化和轮转，会更新索引信息，worker线程进行审计查询或删除过程会触发读取索引信息。加锁是为了索引信息更新或读取的正确性。	正常情况下，当触发审计查询、删除时才会出现审计索引信息读写的临界区。出现长时间等锁时，可以先停止执行审计的查询或删除，或者通过GUC参数开关控制产生审计日志的速度。
GsStackLock	控制gs_stack函数不被并发调用。	防止gs_stack并发调用。	使用gs_stack时才会申请该锁，不使用gs_stack则对性能无影响，无需规避。
DropArchiveSlotLock	保护删除归档槽信息的并发顺序。	同时对归档槽进行并发操作。	重试该操作。
AboCacheLock	控制ABO多列智能统计信息模型缓存的并发读写。	在进行多列智能统计信息模型训练与调用时会尝试缓存并读取已缓存模型以优化性能。需要加锁保证模型读取的正确性。	设置enable_ai_stats = off。
DistributeDecodeSlotLock	分布式逻辑解码的复制槽的创建互斥。	使用分布式逻辑解码，并发。	-

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
ExrtoSnapshotLock	极致RTO备机读快照锁。	正常回放生成快照会持有该锁，回放DDL日志取消查询时，可能持有的时间比较长。	主机减少DDL的执行频率。
LocalBarrierRecordFlushLock	将内存中的barrier信息刷盘。	同时调用了barrier刷盘的操作。	等待正在刷盘的操作执行完成。
ClogTruncateLock	控制clog文件删除和读写并发的锁。	避免clog文件删除和读写并发错误。	-
RedoTruncateLock	回放truncate文件时使用的锁。	避免checkpoint和truncate并发。	DDL操作频率低对性能影响较小。
SysToastOidGeneratorLock	细粒度恢复时，如果涉及toast表，用来生成toast表的oid信息。	同时调用了impdp table操作。	等待细粒度恢复结束。
RedoStatisticLock	回放dfx统计数据锁。	避免查询和修改并发。	-
DisasterCSNInfoListLock	极致RTO容灾读csn和lsn映射表的锁。	回放barrier日志时会生成csn和lsn的映射。	不会出现持锁很长的时间，轻量的操作。
WorkloadRuleLock	workload锁，避免并发操作。	读写workload资源时加锁。	减少业务并发。
AcmFeedbackLock	控制反馈基数估计模型算子收集相关全局变量的并发读写。	在收集算子与模型训练时需要维护收集的算子链表与算子数量的全局信息，需要加锁保证信息正确。	设置enable_adaptive_cost = off。
AcmFeedbackModelCacheLock	控制ABO反馈基数估计模型缓存的并发读写。	在进行反馈基数估计模型训练与调用时通过缓存、尝试读取已缓存模型以优化性能，在进行该操作时需要加锁保证模型读取的正确性。	设置enable_adaptive_cost = off。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
AcmParameterLock	控制智能代价估计模型训练更新时对全局变量的并发读写。	在进行训练与更新智能代价模型时需更新全局变量acm_trainer中的模型参数，需要加锁保证更新与读取正确。	设置enable_adaptive_cost = off。
ArchiveTaskLock	归档时，向归档任务队列中增删任务。	有多个归档任务同时向归档队列中增删数据。	等待下一次任务。
TruncateFileLock	备机回放truncate文件日志时持有的锁。	主机执行vacuum产生的文件truncate日志。	主机避免短时间内删除大量数据。
TmpBarrierBufferLock	清理、合并本地barrier。	有多个清理、合并本地barrier的操作。	等待本地barrier清理、合并结束。
RedoBarrierBufferLock	对内存中的barrier信息加锁更新。	读取或更新redo barrier中的信息。	等待更新完成。
AdioToggleLock	AdioToggleLock是Adio特性开关时加写锁，pread_wrapper、pwrite_wrapper加共享锁。	开启Adio特性时与读写调用互斥。	非性能敏感的锁，无需优化。
LedgerRepairLock	账本数据库中控制归档和修复操作的并发执行。	加锁机制可以保证账本数据库归档和修复操作并发写入的安全性。	避免对账本表进行并发归档和修复操作。
ConsistencyVerifyStatHashLock	保护共享内存中用于数据-索引一致性校验条目登记的哈希表的并发访存。	执行数据-索引一致性校验。	避免发生数据索引不一致（如注入故障）；避免并发执行数据-索引一致性校验。
PreLogicalStandbyControlLock	逻辑备系统函数执行并发控制，以及playback落盘并发控制。	物理转逻辑，或逻辑回放。	非性能敏感的锁，无需优化。
LogicalStandbyMetaFlushLock	物理转逻辑准备解码元信息过程用于并发写控制。	物理转逻辑收集解码元信息。	非性能敏感的锁，无需优化。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
LogicalStandbyMetaControlLock	逻辑备元信息目录内容修改并发控制。	逻辑回放过程元信息维护并发控制。	非性能敏感的锁，无需优化。
WALReceiverDieLock	walrcv退出锁。	避免在写入buffer时walrcv退出并发。	非性能敏感的锁，无需优化。
TSDictionaryDefinitionLock	词典系统表gs_ts_dict_definition的全局锁。	控制gs_ts_dict_definition系统表的读写互斥，保证读取词典内容的完整性和一致性。	减少对TokenWeight类型词典的并发ALTER TEXT SEARCH、gs_ts_dict_add_definition、缓存加载操作。
RespoolBufferContextBlockLock	bufferpool锁。	读写bufferpool。	减少业务并发。
ParallelBuildIdentifierLock	并行build分配build id的锁。	开启并行build。	不会出现持锁很长的时间，轻量操作。
ParallelBuildQueueLock	并行build处理文件名队列的锁。	开启并行build。	不会出现持锁很长的时间，轻量操作。
InstrStmtGstraceCtlLock	控制gstrace注册跟踪函数时不被并发调用。	防止gstrace注册跟踪函数时被并发调用。	调用 gstrace 跟踪接口时才会申请该锁。无需规避。
InstrStmtGstraceSQLCtlLock	控制gstrace注册跟踪SQL不被并发调用。	防止gstrace注册跟踪SQL被并发调用。	调用 gstrace 注册SQL接口、以及注册了SQL在跟踪时才会申请该锁。无需规避。
InconsistencyAlarmHashLock	保护共享内存中用于数据-索引一致性校验条目登记的哈希表的并发访存。	执行数据-索引一致性校验。	避免发生数据索引不一致（如注入故障）；避免并发执行数据-索引一致性校验。
StackAddrToNameLock	控制gs_stack将符号地址转换为符号名不被并发调用。	防止gs_stack将符号地址转换为符号名被并发调用。	调用gs_stack或perf火焰图功能时才会申请该锁，无需调优。
BufMappingLock	用于保护对共享缓冲映射表的操作。	缓冲区映射hash表的分区锁。	可以调整分区数，避免过多的冲突。
BucketBufMappingLock	bucket扩容期间私有缓冲区分区锁。	bucket扩容期间竞争私有缓冲区。	仅在扩容流程内部涉及，不影响业务，暂无调整手段。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
LockMgrLock	用于保护常规锁结构信息。	获取常规锁时竞争写入主锁表。	调节GUC参数num_internal_lock_partitions。
PredicateLockMgrLock	用于保护可串行事务锁结构信息。	用于保护可串行事务锁结构信息。	-
OperatorRealTLock	用于避免并发访问或者修改记录算子级实时数据的全局结构。	-	-
OperatorHistLock	用于避免并发访问或者修改记录算子级历史数据的全局结构。	在更新或查询操作符历史时，可能因锁竞争而导致等待。	优化操作符历史信息的访问逻辑，减少锁竞争，或者优化查询性能。
SessionRealTLock	用于避免并发访问或者修改记录query级实时数据的全局结构。	读取或更新session实时统计信息的哈希表时会申请锁。	减少session数。
SessionHistLock	用于避免并发访问或者修改记录query级历史数据的全局结构。	读取或更新session历史统计信息的哈希表时会申请锁。	减少session数。
InstanceRealTLock	用于保护共享实例统计信息哈希表的更新操作。	读取或更新实例统计信息哈希表时会申请锁。	减轻业务压力。
CSNBufMappingLock	控制共享csnlog页面映射的并发访存。	并发访问csn页面。	减少并发事务数量。
CLogBufMappingLock	用于提交日志缓存管理。	用于提交日志缓存管理。	-

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
UniqueSQLMappingLock	unique SQL统计信息存放在内存的unique SQL哈希表中，UniqueSQLMappingLock为LWLOCK用于保护该哈希表中数据。	unique SQL统计信息存放在内存的unique SQL哈希表中，当前版本共有128个哈希分区。当一个分区中更新数据时，需对相应分区加S锁以避免更新过程中删除动作触发内存问题，此时若有新的语句模板落入该分区，需等待更新操作完成放锁后才可执行添加；当某语句模板需要加入哈希表或从哈希表中删除时，需对该分区加X锁避免对该分区、添加、删除、写入操作，此时此分区内语句信息任何操作需等待插入、删除完成后执行。出现UniqueSQLMappingLock等待事件可能有以下原因： 1. 高并发产生大量新语句模板； 2. 业务高并发时间段调用unique_sql重置函数reset_unique_sql()。	通常情况下，业务稳定运行期间不涉及增减语句模板，不会触发此等待事件，若频繁出现此等待事件，可考虑下列方法规避此： 1. 若观察到业务模型中有大量select in语句，可设置instr_unique_sql_combination_options = 'in_clause'。 原理：将select in()语句多参数情况进行归一，减少新生成语句模板数量，减少哈希表加X锁次数。 影响：原有select in()语句的unique_sql_id发生变化，若已对该类语句打过sql patch，需重新统计相应语句新的id并重新打patch。 2. 若等待事件由上述原因2引起（日志中频繁打印[UniqueSQL] clean unique sql finished.），在业务低谷期再执行reset_unique_sql。 原理：业务高峰期unique sql哈希表频繁更新，reset的删除操作会对哈希表加x锁，造成大规模阻塞，在低谷期做reset可减小加锁对业务的影响。 影响：无。 3. 若等待事件由上述原因2引起（日志中频繁打印[UniqueSQL] Auto-cleanup over, %d uniquesqls are recycled.），可调整enable_auto_clean_uni

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
			que_sql = off或适当增大 instr_unique_sql_count。 原理：关闭自动淘汰功能可避免运行过程中对哈希表加锁删除操作，适当调大 instr_unique_sql_count 可增大unique sql哈希表容量，减少自动淘汰发生频率。 影响：若关闭自动淘汰功能，待哈希表记满后不会再对新语句模板做记录，导致部分统计信息丢失（不影响已有模板继续记录）；若增大 instr_unique_sql_count，会增大unique_sql哈希表内存占用，需结合当前内存使用情况调整。 4. 若观察到大量业务频繁被 UniqueSQLMappingLock阻塞，且不符合上述情况，采用兜底手段关闭unique sql功能，设置 instr_unique_sql_count = 0。 原理：手动关闭unique sql功能，功能关闭后不再操作unique SQL哈希表，不会触发此等待事件。 影响：语句级别统计信息缺失，会影响问题定位。（full sql、asp、wdr report中sql statistics面板均依赖unique sql功能，关闭unique sql会同时影响上述功能使用。）

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
InstrUserLockId	对保护用户登录或者退出登录哈希表并发修改加锁。	读取或更新用户登录、退出信息时对哈希表并发使用。	用户登录、退出或查询用户登录、退出信息时会申请该锁，可以减少使用用户或避免用户同时登录、退出，降低等待锁的可能。
GPCMappingLock	用于全局Plan Cache缓存管理。	获取、存入计划或计划失效清理取锁。	属于GPC执行过程中正常取锁操作。计划失效清理时写锁可能耗时较长，可降低DDL频率，或关闭enable_global_plancache功能。
ProcXactMappingLock	保护事务号-线程信息映射哈希表的并发访存。	当前系统线程数量大。	减少并发线程数量。
ASPMappingLock	用于管理ASP的hash表的并发访存。	ASP中拿取uniqueSQL中数据。	ASP中用于拿取uniqueSQL数据，无需规避。
RecoveryXidMappingLock	保护两阶段事务的恢复所需要的共享内存。	故障前存在已prepared事务。	减少进程重启前prepare commit事务。
BufferIOLock	保护共享缓冲区单个页面加载或淘汰的IO并发。	缓冲区映射hash表的访问锁。	减少并发对同一页面进行I/O操作。
BufferContentLock	保护共享缓冲区单个页面的读取、修改。	缓冲区页面内容锁。	减少并发读写同一页面。
BucketBufferIOLock	bucket扩容期间私有缓冲区IO锁。	bucket扩容期间竞争私有缓冲区。	-
BucketBufferContentLock	bucket扩容期间私有缓冲区内容锁。	bucket扩容期间竞争私有缓冲区。	-

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
UndoPerZoneLock	保护每个undoZone内信息的并发访存。	每个业务线程绑定一个UndoZone，更改UndoZone的元信息内容时，需要对UndoZone加锁，以防止业务线程和回收线程的并发修改产生问题。	减少对Ustore表的DML操作。
UndoSpaceLock	保护undoSpace的并发访存。	每个UndoZone绑定一个UndoSpace，用于管理物理文件。对UndoSpace的元信息内容进行更改时，需要对UndoSpace加锁，以防止业务线程和回收线程的并发修改产生问题。	减少对Ustore表的DML操作。
PGPROCLock	保护全局共享线程状态数组的并发访存。	新建连接以及连接退出分配proc资源，与快照获取之间锁冲突。	避免频繁建连。
ReplicationSlotLock	用于保护ReplicationSlot。	将复制槽持久化到磁盘。	后台任务无需调优。
Async Ctl	用于保护异步控制模块的共享内存结构。	确保多进程并发访问时的数据一致性。	减少并发操作。
CLOG Ctl	保护clog的并发访存。	每个节点更新事务状态信息都需要访问clog。	减少并发事务数量。
CSNLOG Ctl	保护csn log的并发访存。	每个节点事务提交需要将事务的csn更新到csn log里。	减少并发事务数量。
MultiXactOffset Ctl	保护multixact日志中事务id的偏移量的并发访存。	当多个事务同时操作一个元组时，需要分配出multixactid,并保留到日志中。	减少多个事务对同一个元组进行修改或加锁。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
MultiXactMember Ctl	保护multixact日志中事务id的并发访存。	当多个事务同时操作一个元组时，需要分配出multixactid,并保留到日志中。	减少多个事务对同一个元组进行修改或加锁。
OldSerXid SLRU Ctl	保护序列化事务的共享内存。	当设置事务隔离级别为序列化时，多个事务同时对tuple进行修改。	减少序列化事务修改同一个tuple的场景。
WALInsertLock	用于避免与其它WAL插入操作并发执行。	插入WAL记录。	减少业务量，检查I/O性能。
SnapshotBlockLock	控制快照备份与页面刷盘的并发顺序。快照备份期间，阻塞双写页面落盘。	正在执行快照备份操作。	等待快照备份结束后执行。
DoubleWriteLock	用于双写的管理操作。	写双写文件。	pagewriter个数和双写文件个数相等不再维护。
DWSingleFlushFirstLock	控制非段页式单页面双写文件的并发顺序。	写单写文件。	避免并发页面淘汰。
DWSingleFlushSecondLock	控制段页式单页面双写文件的并发顺序。	写单写文件。	避免并发页面淘汰。
DWSingleFlushSecondBufTagLock	控制段页式单页面双写文件的元信息并发访存。	写单写文件。	避免并发页面淘汰。
RestartPointQueueLock	增量检查点checkpoint redo推进锁并发控制回放过程的redo推进位置。	并发控制回放过程redo点推进位置。	减少主机checkpoint频率，减少恢复过程和按需回放并发访问。
PruneDirtyQueueLock	增量检查点下脏页队列清理锁。	避免恢复过程做检查点更新脏页尾位置和pagewriter主线程清理脏页队列冲突。	-

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
UnlinkRelHashTblLock	保护共享内存中待删除文件哈希表的并发访存。	删除表的所有物理文件。	减少DDL操作。
UnlinkRelForkHashTblLock	保护共享内存中待删除文件fork哈希表的并发访存。	删除指定fork的物理文件。	减少DDL操作。
LWTRANCHE_ACCOUNT_TABLE	控制备机账号锁定记录的hash表的并发读写。	提供密码保护策略，用户使用错误密码登录备机时，会更新hash表，记录失败次数和更新锁定状态并存放在系统内存中，达到最大失败次数后，会禁止该用户登录备机，保证数据库备机的安全。	调优建议：使用错误密码登录是不安全的行为，可能是密码被修改，但用户未能感知；或者存在恶意用户不断尝试数据库密码。数据库管理员可定期检查所有用户的锁定状态，并分析被锁定账户，减少错误密码登录的次数，有以下几种情况： 1）数据库遭受恶意攻击：管理员可通过日志分析登录的账号，找出攻击源并采取相应措施； 2）密码被修改但用户未察觉：提醒用户及时更新密码，以保障业务正常运行。
MPFLLOCK	回放完成且落盘的最大lsn。	更新回放完成且落盘的最大lsn。	-
GlobalTempTableControl	保护全局临时表的本地共享hash表。	插入、删除、修改、truncate、访问全局临时表。	减少全局临时表的DDL、DML并发操作。
NGroupMappingLock	用于保护nodegroup哈希表的单个分桶并发修改加锁。	插入或删除单个分桶内的nodegroup时加X锁，与其他加X锁或S锁场景互斥产生等待事件。	仅在进行涉及nodegroup操作时会出现，其他场景不会发生此等待事件。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
MatviewSeqnoLock	用于维护增量物化视图功能中的mlog表seqno列。	定义增量物化视图时，会通过mlog表维护基表的增量数据。基表增量数据写入mlog表时会申请该锁。seqno字段保证写入的增量数据有唯一且递增的序列号（标记增量数据的顺序）。	定义增量物化视图时，基表增量数据写入mlog表会调用该锁。在某表IUD性能较为重要的场景下，应该避免对该表创建增量物化视图。
IOStatLock	用于资源管理IO统计信息哈希表并发维护操作。	读取或更新IO统计信息哈希表时会申请锁。	减少I/O操作。
StandbyStmtHistLock	主要用于备机FullSQL定位能力。	在慢SQL文件落盘时，保护standby_statement_history函数对变更的同一资源避免读越界。	主要用于慢SQL定位使用，无需规避。
WALBufferInitWait	保护WAL共享内存页面的初始化和刷盘顺序。	初始化WAL Buffer。	仅初始化时触发，无需调优。
WALInitSegment	保护WAL日志段文件的初始化顺序。	Xlog预初始化。	仅初始化时触发，无需调优。
SegmentHeadPartitionLock	保护段页式segment头部元信息的分区锁。	段页式对象的extent发生变化时，需要锁住segment header防止并发产生异常。	-
TwoPhaseStatePartLock	保护（各个分区）两阶段事务状态信息。	保护（各个分区）两阶段事务状态信息。	-
RoleIdPartLock	保护进程角色Id的增删查。	线程池模式下对进程角色Id增删查。	-
GPRCMappingLock	用于管理自治事务全局缓存哈希表的访问和修改操作。	高并发自治事务session场景，概率出现该锁等待事件。	减少自治事务session并发数量。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
PgwrSyncQueueLock	保护待刷盘文件队列的并发访存。	增量checkpoint。	减少checkpoint频率。
BarrierHashTblLock	保护共享内存中barrier列表信息的并发访存。	对barrier_hash_table的并发访问，正常分布式容灾运行过程中，预解析线程解析barrier插入到hash表，回放线程回放完barrier日志后从hash表移除。	-
PageRepairHashTblLock	保护页面修复哈希表的并发访存。	CRC损坏/LSN损坏页面被检测到，记录坏块。	避免产生坏块。
FileRepairHashTblLock	保护文件修复哈希表的并发访存。	发现文件损坏时，记录信息。	避免发生文件损坏。
ReplicationOriginLock	发布订阅创建、删除、变更状态用锁。	使用发布订阅。	减少发布订阅功能的启动。
PCABufferContentLock	跨页透明压缩特性，需要一个cache页面缓存cmap page，保护cache页面的一致性，功能类似于shared_buffer的content lock。	为了保护cache页面的一致性。	-
WALConsensusWait	保护日志达成一致才进行事务提交或日志回放操作。	需要确保WAL日志在多个节点上达成共识，可能因为网络延迟或节点性能差异导致等待。	优化网络配置和节点性能，以减少WAL日志共识的时间，或者调整事务的提交策略以减少对共识机制的依赖。
SegMappingLock	保护段页式cache哈希表的并发访问。	段页式维护了一张哈希表，保存逻辑位置和物理位置的对应关系。哈希表存在并发访问的情况，需要加锁。	-

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
PendingDelLock	保护线程上的pendinglist并发修改。	事务中创建、删除、修改等操作过多时，pendinglist可能过长，操作pendinglist耗时可能较长。	减少事务中DDL操作的次数。
AutonomousSessionLock	用于自治事务建联时加锁。	高并发下保证自治事务顺序建联。	减少自治事务并发使用。
CgroupHashLock	用于防止并发初始化锁。	并发初始化。	减少并发业务。
IMCVInfoHashTblLock	用于保护htap存放全部imcvdesc的哈希表。	用于保护htap存放全部imcvdesc的哈希表。	使用分区表，构造多条gs_imcv，减少冲突。
IMCVDescExpandLock	用于保护单个IMCVDesc的并发安全。	用于保护单个IMCVDesc的并发安全。	减少并发加载COLVIEW操作。
LogicalStandbyWalLock	逻辑备拉取处理远端wal日志互斥。	使用逻辑回放。	-
DelayDdlInfoLock	备机进行延迟DDL持有的锁。	主机执行DDL。	主机减少DDL的执行频率。
NvmeLSPosLock	用于保护nvme哈希的4096个分区每个哈希分区的读写。	用来统计分区锁范围的时间，用以现网hang问题和性能问题定位。	调整索引结构分散热点页面分布。
BucketWriterLock	bucket扩容期间私有缓冲区sync锁。	bucket扩容期间竞争私有缓冲区。	-
ClogBucketLock	用于保护bucket clog控制文件。	大量bucket表业务并发扩展clog页面。	-
BufFreelistLock	已废弃。	已废弃。	已废弃。
CUSlotListLock	已废弃。	已废弃。	已废弃。
ShmemIndexLock	不涉及。	不涉及。	不涉及。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
CLogControlLock	已废弃。	已废弃。	已废弃。
SubtransControlLock	已废弃。	已废弃。	已废弃。
TwoPhaseStateLock	已废弃。	已废弃。	已废弃。
AddinShmemInitLock	已废弃。	已废弃。	已废弃。
AsyncCtlLock	已废弃。	已废弃。	已废弃。
AsyncQueueLock	已废弃。	已废弃。	已废弃。
DataFileIdCacheLock	已废弃。	已废弃。	已废弃。
CStoreColspaceCacheLock	已废弃。	已废弃。	已废弃。
CStoreCUCacheSweepLock	已废弃。	已废弃。	已废弃。
MetaCacheSweepLock	已废弃。	已废弃。	已废弃。
FdwPartitionCacheLock	已废弃。	已废弃。	已废弃。
DfsConnectorCacheLock	已废弃。	已废弃。	已废弃。
dummyServerInfoCacheLock	已废弃。	已废弃。	已废弃。
SearchServerLibLock	已废弃。	已废弃。	已废弃。
DfsUserLoginLock	已废弃。	已废弃。	已废弃。
DfsSpaceCacheLock	已废弃。	已废弃。	已废弃。
LsnXlogChkFileLock	已废弃。	已废弃。	已废弃。
WorkloadStatHashLock	已废弃。	已废弃。	已废弃。
WorkloadIoStatHashLock	已废弃。	已废弃。	已废弃。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
WorkloadSessionInfoLock	已废弃。	已废弃。	已废弃。
WorkloadRecordLock	已废弃。	已废弃。	已废弃。
PgfdwLock	已废弃。	已废弃。	已废弃。
RelfilenodeReuseLock	已废弃。	已废弃。	已废弃。
GPCCommitLock	已废弃。	已废弃。	已废弃。
GPCTimelineLock	已废弃。	已废弃。	已废弃。
TsTagsCacheLock	已废弃。	已废弃。	已废弃。
MaxPageFlushLsnFileLock	已废弃。	已废弃。	已废弃。
StreamingEngineConnLock	已废弃。	已废弃。	已废弃。
StreamingEngineExecLock	已废弃。	已废弃。	已废弃。
PartIdCacheLock	已废弃。	已废弃。	已废弃。
PartOidCacheLock	已废弃。	已废弃。	已废弃。
DeleteCompactionLock	已废弃。	已废弃。	已废弃。
DeleteConsumerLock	已废弃。	已废弃。	已废弃。
UHeapStatLock	不涉及。	不涉及。	不涉及。
ConsumerStateLock	已废弃。	已废弃。	已废弃。
SyncPaxosLock	已废弃。	已废弃。	已废弃。
BackgroundWorkerLock	已废弃。	已废弃。	已废弃。
HadrSwitchoverLock	已废弃。	已废弃。	已废弃。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
UniqueSqlEvictLock	不涉及。	不涉及。	不涉及。
ExrtoRecycleResidualUndoLock	已废弃。	已废弃。	已废弃。
DblinkFdwRelationLock	已废弃。	已废弃。	已废弃。
SPMGlobalLock	不涉及。	不涉及。	不涉及。
SlotsArrayLock	已废弃。	已废弃。	已废弃。
MtdStatusLock	已废弃。	已废弃。	已废弃。
AltLogicalMapClearLock	不涉及。	不涉及。	不涉及。
AltSnapTrackerXminLock	不涉及。	不涉及。	不涉及。
AltSnapTrackerCsnLock	已废弃。	已废弃。	已废弃。
OndemandQueueStatLock	不涉及。	不涉及。	不涉及。
AscendHashLock	不涉及。	不涉及。	不涉及。
CacheSlotMappingLock	已废弃。	已废弃。	已废弃。
LtxidMappingLock	不涉及。	不涉及。	不涉及。
UspagrpMappingLock	已废弃。	已废弃。	已废弃。
GlobalSeqLock	已废弃。	已废弃。	已废弃。
GlobalWorkloadLock	已废弃。	已废弃。	已废弃。
NormalizedSqlLock	已废弃。	已废弃。	已废弃。
StartBlockMappingLock	已废弃。	已废弃。	已废弃。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
DataCacheLock	已废弃。	已废弃。	已废弃。
MetaCacheLock	已废弃。	已废弃。	已废弃。
GeneralExtendedLock	已废弃。	已废弃。	已废弃。
PLdebugger	已废弃。	已废弃。	已废弃。
GlobalCompilationProductHashTblLock	不涉及。	不涉及。	不涉及。
WALFlushWait	已废弃。	已废弃。	已废弃。
AuditIndextblLock	已废弃。	已废弃。	已废弃。
GlobalPlanCacheLock	已废弃。	已废弃。	已废弃。
QueryPlanCacheLock	已废弃。	已废弃。	已废弃。
SPMMappingLock	不涉及。	不涉及。	不涉及。
SPMGlobalInfoLock	已废弃。	已废弃。	已废弃。
AntiCacheLock	不涉及。	不涉及。	不涉及。
SQLFeedbackLock	不涉及。	不涉及。	不涉及。
IMCUDataCacheLock	已废弃。	已废弃。	已废弃。
IMCUMetaCacheLock	已废弃。	已废弃。	已废弃。
IMCVDescLatch	已废弃。	已废弃。	已废弃。
WalRemoteWaitLock	不涉及。	不涉及。	不涉及。
AbnormalSQLMappingLock	不涉及。	不涉及。	不涉及。

当wait_status值为wait io时对应的wait_event等待事件类型与描述信息如**表7-472**所示。

版权所有 © 华为云计算技术有限公司

表 7-472 I/O 等待事件列表

wait_event类型	类型描述	出现原因	调优建议
buffile_read	等待从临时文件读取数据。	查询语句产生临时文件。	增大work_mem内存。
buffile_write	等待写临时文件。	查询语句产生临时文件。	增大work_mem内存。
BufHashTableSearch	共享缓冲区hash表搜索（可能会触发页面淘汰）。	从缓冲区查找指定buffer。	增大共享缓冲区大小，增加buffer命中率。
buffer_strategy_get	缓冲区替换策略中找到buffer耗时。	获取可用buffer。	减小checkpoint时间，pagewriter_sleep、bgwriter_delay时间，加快扫描缓存区的频率；减少并发。
ControlFileRead	读取pg_control文件。主要在数据库启动、执行checkpoint和主备校验过程中发生。	在数据库启动、执行checkpoint和主备校验过程中，从控制文件加载数据。	减少启停数据库、checkpoint、主备切换操作。
ControlFileSync	将pg_control文件持久化到磁盘。数据库初始化时发生。	数据库启动时，将pg_control文件中的数据落盘。	减少启停数据库。
ControlFileSyncUpdate	将pg_control文件持久化到磁盘。主要在数据库启动、执行checkpoint和主备校验过程中发生。	在数据库启动、执行checkpoint和主备校验过程中，将pg_control文件中的数据落盘。	减少启停数据库、checkpoint、主备切换操作。
ControlFileWrite	写入pg_control文件。数据库初始化时发生。	数据库启动时，会初始化控制文件，并写入磁盘。	减少启停数据库。
ControlFileWriteUpdate	更新pg_control文件。主要在数据库启动、执行checkpoint和主备校验过程中发生。	在数据库启动、执行checkpointer、主备校验过程中，需要将控制文件变更信息，写入pg_control文件。	减少启停数据库、checkpoint、主备切换操作。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
CopyFileRead	copy文件时读取文件内容。	由于文件I/O性能瓶颈或源文件被其他操作锁定。	优化I/O性能，或确保源文件未被其他操作长时间锁定，减少锁竞争。
CopyFileWrite	copy文件时写入文件内容。	由于文件I/O性能瓶颈或目标文件被其他操作锁定。	优化I/O性能，或确保目标文件未被其他操作长时间锁定，减少锁竞争。
DataFileExtend	扩展文件时向文件写入内容。	触发扩展文件操作。	减少业务量，检查I/O性能。
DataFileImmediateSync	将表数据文件立即持久化到磁盘。	触发checkpoint。	调整checkpoint间隔并检查I/O性能。
DataFilePrefetch	异步读取表数据文件。	内存里缺少表数据文件信息，需要从磁盘加载。	调整shared_buffers大小并检查I/O性能。
DataFileRead	同步读取表数据文件。	内存里缺少表数据文件信息，需要从磁盘加载。	调整shared_buffers大小并检查I/O性能。
DataFileSync	将表数据文件的修改持久化到磁盘。	触发checkpoint。	调整checkpoint间隔并检查I/O性能。
DataFileTruncate	表数据文件truncate。	触发truncate操作。	减少对truncate表的并发操作。
DataFileWrite	向表数据文件写入内容。	触发checkpoint。	调整checkpoint间隔并检查I/O性能。
SegFileExtend	段页式文件扩展。	数据库在需要更多存储空间时会扩展段文件，由于文件系统性能限制会导致等待。	优化文件系统的I/O性能，或者预先分配足够的存储空间以减少扩展操作。
SegReadDisk	段页式文件读取。	数据库在访问数据时需要从磁盘读取段文件，由于I/O性能限制会导致等待。	优化存储设备的I/O性能，或者增加缓存以减少读取操作。
SegWriteDisk	段页式文件写入。	数据库在更新数据时需要写入段文件，由于I/O性能限制会导致等待。	优化存储设备的I/O性能，或者增加写入操作的并行度。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
SegSync	段页式文件刷盘。	在数据修改后需要确保数据持久化，因此需要等待文件同步。	优化存储设备的I/O性能，或者调整同步操作的频率以减少等待时间。
SegFileShrink	段页式文件收缩。	数据库在释放不再需要的存储空间时会收缩段文件，由于文件系统性能限制会导致等待。	优化文件系统的I/O性能，或者减少段文件收缩操作的频率。
LsnInfoPageDiskRead	极致RTO备机读，读lsn info页面的I/O延迟。	开启极致RTO备机读。	增强I/O能力。
BlkInfoMetaDiskRead	极致RTO备机读，读block info页面的I/O延迟。	开启极致RTO备机读。	增强I/O能力。
BasePageDiskRead	极致RTO备机读，读base page页面的I/O延迟。	开启极致RTO备机读。	增强I/O能力。
LockFileAddToDataDirRead	读取"postmaster.pid"文件。	并发读取"postmaster.pid"文件。	减少并发业务。
LockFileAddToDataDirSync	将"postmaster.pid"内容持久化到磁盘。	并发持久"postmaster.pid"文件。	减少并发业务。
LockFileAddToDataDirWrite	将pid信息写到"postmaster.pid"文件。	并发写"postmaster.pid"文件。	减少并发业务。
LockFileCreateRead	读取LockFile文件"%s.lock"。	并发读取LockFile文件。	减少并发业务。
LockFileCreateSync	将LockFile文件"%s.lock"内容持久化到磁盘。	并发持久LockFile文件。	减少并发业务。
LockFileCreateWRITE	将pid信息写到LockFile文件"%s.lock"。	并发写LockFile文件。	减少并发业务。
ReplicationSlotRead	读取流复制槽文件。重新启动时发生。	读取复制槽持久化信息。	仅进程启动，加载复制槽信息时会调用，无需调优。
ReplicationSlotRestoreSync	将流复制槽文件持久化到磁盘。重新启动时发生。	从磁盘同步复制槽持久化信息。	仅进程启动，加载复制槽信息时会调用，无需调优。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
ReplicationSlotSync	checkpoint时将流复制槽临时文件持久化到磁盘。	将复制槽信息持久化到磁盘。	仅创建或者刷新复制槽状态时会调用，无需调优。
ReplicationSlotWrite	checkpoint时写流复制槽临时文件。	将复制槽信息写入临时文件。	仅创建或者刷新复制槽状态时会调用，无需调优。
SLRUFlushSync	将pg_clog、pg_subtrans和pg_multixact文件持久化到磁盘。主要在执行checkpoint和数据库停机时发生。	触发checkpoint。	适当调小checkpoint间隔并检查I/O性能。
SLRURead	读取pg_clog、pg_subtrans和pg_multixact文件。	内存里缺少clog、csnlog信息，需要从磁盘加载。	适当调大shared_buffers大小并检查I/O性能。
SLRUSync	将脏页写入文件pg_clog、pg_subtrans和pg_multixact并持久化到磁盘。主要在执行checkpoint和数据库停机时发生。	触发checkpoint。	适当调小checkpoint间隔并检查I/O性能。
SLRUWrite	写入pg_clog、pg_subtrans和pg_multixact文件。	触发checkpoint。	适当调小checkpoint间隔并检查I/O性能。
TwophaseFileRead	读取pg_twophase文件。在两阶段事务提交、两阶段事务恢复时发生。	内存里缺少pg_twophase信息，需要从磁盘加载。	适当调大shared_buffers大小并检查I/O性能。
TwophaseFileSync	将pg_twophase文件持久化到磁盘。在两阶段事务提交、两阶段事务恢复时发生。	在两阶段事务期间，将脏页持久化到磁盘。	适当调小checkpoint间隔并检查I/O性能。
TwophaseFileWrite	写入pg_twophase文件。在两阶段事务提交、两阶段事务恢复时发生。	在两阶段事务期间，重建磁盘文件。	适当调小checkpoint间隔并检查I/O性能。
UndoFileExtend	undo文件扩展。	DML会产生undo记录，当空间不足时需要扩展undo文件，用于存储新的undo记录。	减少对Ustore表的DML操作。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
UndoFilePrefetch	undo文件异步读取。	当系统支持posix_fadvise()系统调用时，数据库会对扫描进行预读取以提升性能。	减少对Ustore表的DML操作。
UndoFileRead	undo文件读取。	当业务需要访问元组历史版本，或者事务需要回滚时，会读取undo记录。	减少对Ustore表的DML操作。
UndoFileWrite	undo文件写入。	DML会产生undo记录，写undo记录会先写入缓冲区，然后在pagewriter线程中落盘。此等待事件产生在落盘流程。	减少对Ustore表的DML操作。
UndoFileSync	undo文件同步落盘。	同步将undo记录落盘到磁盘。	减少对Ustore表的DML操作。
UndoFileUnlink	undo文件删除。	每个Ustore表的DML操作均会产生undo记录，对无用的undo记录需要及时删除，以防undo空间不断膨胀。	减少对Ustore表的DML操作。
UndoSegFileExtend	undo段页式文件扩展。	DML会产生undo记录，当空间不足时需要扩展undo段页式文件，用于存储新的undo记录。	减少对Ustore表的DML操作。
UndoSegReadDisk	undo段页式文件读取。	当业务需要访问元组历史版本，或者事务需要回滚时，会读取undo记录。	减少对Ustore表的DML操作。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
UndoSegWriteDisk	undo段页式文件写入。	DML会产生undo记录，写undo记录会先写入缓冲区，然后在pagewriter线程中落盘。此等待事件产生在落盘流程。	减少对Ustore表的DML操作。
UndoSegSync	undo段页式文件同步落盘。	同步将undo记录落盘到磁盘。	减少对Ustore表的DML操作。
UndoMetaSync	undo元数据文件同步落盘。	undo元数据会随着每轮checkpoint刷盘一次。确保checkpoint点更新之后，undo元数据的内容已经被实时持久化。	调大每轮增量checkpoint的时间间隔incremental_checkpoint_timeout。
WALBootstrapSync	将初始化的WAL文件持久化到磁盘。在数据库初始化发生。	初始化数据库。	仅初始化时触发，无需调优。
WALBootstrapWrite	写入初始化的WAL文件。在数据库初始化发生。	初始化数据库。	仅初始化时触发，无需调优。
WALCopyRead	读取已存在的WAL文件并进行复制时产生的读操作。在执行归档恢复完后发生。	执行归档恢复。	与I/O性能相关，可从硬盘优化。
WALCopySync	将复制的WAL文件持久化到磁盘。在执行归档恢复完后发生。	执行归档恢复。	与I/O性能相关，可从硬盘优化。
WALCopyWrite	读取已存在WAL文件并进行复制时产生的写操作。在执行归档恢复完后发生。	执行归档恢复。	与I/O性能相关，可从硬盘优化。
WALInitSync	将新初始化的WAL文件持久化磁盘。在日志回收或写日志时发生。	创建新的日志文件。	与I/O性能相关，可从硬盘优化。
WALInitWrite	将新创建的WAL文件初始化为0。在日志回收或写日志时发生。	创建新的日志文件。	与I/O性能相关，可从硬盘优化。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
WALRead	从Xlog日志读取数据。两阶段文件redo相关的操作产生。	日志传输或逻辑解码操作。	与I/O性能相关，可从硬盘优化。
ShareStorageWalRead	共享盘读取日志文件。	共享存储WAL读取速度较慢，导致读取操作耗时。	优化网络和存储性能，或者增加WAL文件的读取并行度。
ShareStorageWalWrite	共享盘写入日志文件。	共享存储WAL写入速度较慢，导致写入操作耗时。	优化存储性能，或者调整WAL日志生成的配置以减少写入操作的等待。
ShareStorageCtlInfoRead	共享盘读取控制信息。	共享存储控制信息读取速度较慢，导致读取操作耗时。	优化存储性能，或者减少控制信息读取的频率。
ShareStorageCtlInfoWrite	共享盘写入控制信息。	共享存储控制信息写入速度较慢，导致写入操作耗时。	优化存储性能，或者减少控制信息写入的频率。
WALSyncMethodAssign	将当前打开的所有WAL文件持久化到磁盘。	wal_sync_method参数变更。	与I/O性能相关，可从硬盘优化。
WALWrite	写入WAL文件。	写入Xlog文件。	与I/O性能相关，可从硬盘优化。
WALBufferAccess	WAL Buffer访问（出于性能考虑，内核代码里只统计访问次数，未统计其访问耗时）。	插入WAL记录。	不统计耗时，无需调优。
WALBufferFull	WAL Buffer满时，写WAL文件相关的处理。	WAL Buffer满，等待下刷。	增大wal_buffers设置。
DoubleWriteFileRead	等待读取双写文件。	数据库启动阶段，扫描双写文件中的页面，对半写页面进行修复。	减少启停数据库，避免出现半写。
DoubleWriteFileWrite	等待写入双写文件。	触发checkpoint。	减少checkpoint操作。
DWSingleFlushGetPos	单页面双写文件查找可用位置。	增量检查点做全量checkpoint时，截断单写文件。	检查手动checkpoint操作。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
DWSingleFlushWrite	单页面双写文件刷盘。	执行脏页淘汰的时候写单写文件。	减少页面淘汰，适当增加shared_buffers。
CkptWaitPageWriterFlush	执行全量checkpoint时等待刷页。	脏页刷新速度较慢，导致检查点操作受阻。	优化I/O性能，或增加写入器数量以提高写入效率。
CkptWaitPageWriterSync	checkpoint打点前等待有修改的文件同步到磁盘。	脏页同步速度较慢，导致检查点操作受阻。	优化I/O性能，或调整检查点配置以减少同步操作的等待。
CkptWaitCommitTransactionFinish	checkpoint打点前等待所有已经提交的事务提交完毕。	检查点需要等待所有活跃事务提交，以确保数据一致性。	优化事务处理流程，减少长事务，或者调整检查点的触发频率。
PredoProcessPending	并行日志回放中，当前记录回放等待其它记录回放完成。	出现需要等待同步的日志，如DDL相关日志。	减少DDL操作。
PredoApply	并行日志回放中，当前工作线程等待其他线程回放至本线程lsn。	出现需要等待同步的日志，如DDL相关日志。	减少DDL操作。
DisableConnectFileRead	HA锁分片逻辑文件读取。	HA build成功，同步主备配置文件。	与I/O性能相关，可从硬盘优化。
DisableConnectFileSync	HA锁分片逻辑文件强制刷盘。	HA build成功，同步主备配置文件。	与I/O性能相关，可从硬盘优化。
DisableConnectFileWrite	HA锁分片逻辑文件写入。	HA build成功，同步主备配置文件。	与I/O性能相关，可从硬盘优化。
MPFL_INIT	初始化max_page_flush_lsn。	初始化控制文件。	与I/O性能相关，可从硬盘优化。
MPFL_READ	读取max_page_flush_lsn。	读取控制文件。	与I/O性能相关，可从硬盘优化。
MPFL_WRITE	写入max_page_flush_lsn。	写入控制文件。	与I/O性能相关，可从硬盘优化。
LOGCTRL_SLEEP	等待备机回放追赶。	打开流控场景。	关闭流控。
WALRcvBufFull	备机walreceiver接收Xlog日志放到缓冲区过程，缓冲区已满时发生。	WAL接收缓冲区容量不足，无法及时处理接收到的WAL日志。	与I/O性能相关，可从硬盘优化。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
WALFileSync	将WAL文件持久化到磁盘。	在执行WAL写入操作后，需要确保数据持久化，因此需要等待文件同步。	与I/O性能相关，可从硬盘优化。
VlogInitSync	等待vlog文件初始化。	打开页面校验持久化后，创建新的vlog文件。	与I/O性能相关，可从硬盘优化。
VlogWrite	等待将vlog记录写入文件。	打开页面校验持久化后，将vlog记录写入文件。	与I/O性能相关，可从硬盘优化。
OBSWrite	已废弃。	已废弃。	已废弃。
invalid io event	意料之外的读写事件。	发生了不应该发生的读写事件。	联系华为技术支持。

当wait_status值为acquire lock（事务锁）时对应的wait_event等待事件类型与描述信息如表7-473所示。

表 7-473 事务锁等待事件列表

wait_event类型	类型描述	出现原因	调优建议
relation	对表加锁。	业务语句获取表锁冲突。	避免对同一个表并发DML和DDL。
extend	对表扩展空间时加锁。	频繁对同一个表扩展空间锁冲突。	避免对同一个表并发插入大量数据。
partition	对分区表加锁。	业务语句获取分区表锁冲突。	避免对同一个分区表并发DML和DDL。
partition_seq	对分区表的分区加锁。	业务语句获取分区表的分区表锁冲突。	避免对同一个分区表的分区并发DML和DDL。
page	对表页面加锁。	业务语句增删改向量元数据冲突。	减少对向量表向量列的并发DML。
tuple	对页面上的tuple加锁。	执行DML时会对tuple加锁。	减少并发DML操作。
transactionid	对事务id加锁。	并发访问相同事务。	减少并发访问相同事务。
virtualxid	对虚拟事务id加锁。	并发访问相同虚拟事务。	减少并发访问相同虚拟事务。

版权所有 © 华为云计算技术有限公司

wait_event类型	类型描述	出现原因	调优建议
object	加对象锁。	数据库内部对象锁冲突。	分析具体流程代码。
advisory	加advisory锁。	调用咨询锁函数发生锁冲突，例如session1调用pg_advisory_lock_int8并长时间不放锁，session2调用pg_advisory_lock_int8同样参数就会进入等待状态。	优化业务流程，避免锁冲突。
filenode	对文件名加锁，控制文件级操作的并发顺序。	由于多个会话并发访问同一文件时，系统需要对文件加锁以控制并发顺序，防止数据冲突。	优化访问逻辑，减少对同一文件的并发操作。
subtransactionid	对子事务号加锁。	并发访问相同子事务。	减少并发访问相同子事务。
tuple_uid	对元组头部的uid隐藏字段加锁。	对Ustore表执行DML。	减少对Ustore表的并发DML操作。
spectoken	保护upsert过程。	事务里并发执行upsert。	避免事务并发执行upsert或者优化upsert性能。
bucket	加bucket锁。	业务语句获取bucket锁冲突。	-
vector_ccbuild	对索引的tuple加锁。	向量索引创建时会对索引的元组加锁。	减少并发创建操作。
cstore_freespace	已废弃。	已废弃。	已废弃。
userlock	已废弃。	已废弃。	已废弃。

7.3.13.188 PG_TIMEZONE_ABBREVS

PG_TIMEZONE_ABBREVS视图显示所有可用时区的信息。具体字段信息如**表7-474**所示。

版权所有 © 华为云计算技术有限公司

表 7-474 PG_TIMEZONE_ABBREVS 字段

名称	类型	描述
abbrev	text	时区名缩写。
utc_offset	interval	相对于UTC的偏移量。
is_dst	boolean	如果当前正实行夏令时则为t，否则为f。

7.3.13.189 PG_TIMEZONE_NAMES

PG_TIMEZONE_NAMES视图显示所有能够被SET TIMEZONE语法识别的时区名及其缩写、UTC偏移量、是否实行夏令时。具体字段信息如表7-475所示。

表 7-475 PG_TIMEZONE_NAMES 字段

名称	类型	描述
name	text	时区名。
abbrev	text	时区名缩写。
utc_offset	interval	相对于UTC的偏移量。
is_dst	boolean	如果当前正实行夏令时则为t，否则为f。

7.3.13.190 PG_TOTAL_MEMORY_DETAIL

PG_TOTAL_MEMORY_DETAIL视图显示某个数据库节点内存使用情况。当GUC参数enable_memory_limit的值为off时，本视图不可用。具体字段信息如表7-476所示。

表 7-476 PG_TOTAL_MEMORY_DETAIL 字段

名称	类型	描述
nodename	text	节点名称。
memorytype	text	内存的名称。
memorymbytes	integer	内存使用的大小，单位为MB。

7.3.13.191 PG_TOTAL_USER_RESOURCE_INFO

PG_TOTAL_USER_RESOURCE_INFO视图显示所有用户的资源使用情况，需要使用管理员用户进行查询。此视图在GUC参数use_workload_manager为on时才有效。其中，IO相关监控项在参数enable_logical_io_statistics为on时才有效。具体字段信息如表7-477所示。

版权所有 © 华为云计算技术有限公司

表 7-477 PG_TOTAL_USER_RESOURCE_INFO 字段

名称	类型	描述
username	name	用户名。
used_memory	integer	当前版本已废弃，查询结果无参考意义。
total_memory	integer	当前版本已废弃，查询结果无参考意义。
used_cpu	double precision	当前版本已废弃，查询结果无参考意义。
total_cpu	integer	当前版本已废弃，查询结果无参考意义。
used_space	bigint	已使用的永久表存储空间大小，单位KB。
total_space	bigint	可使用的永久表存储空间大小，单位KB，值为-1表示未限制最大存储空间。
used_temp_space	bigint	已使用的临时空间大小，单位KB。
total_temp_space	bigint	可使用的临时空间总大小，单位KB，值为-1表示未限制。
used_spill_space	bigint	已使用的算子落盘空间大小，单位KB。
total_spill_space	bigint	可使用的算子落盘空间总大小，单位KB，值为-1表示未限制。
read_kbytes	bigint	CN：过去5秒内，该用户在所有DN上复杂作业read的字节总数（单位KB）。 DN：实例启动至当前时间为止，该用户复杂作业read的字节总数（单位KB）。
write_kbytes	bigint	CN：过去5秒内，该用户在所有DN上复杂作业write的字节总数（单位KB）。 DN：实例启动至当前时间为止，该用户复杂作业write的字节总数（单位KB）。
read_counts	bigint	CN：过去5秒内，该用户在所有DN上复杂作业read的次数之和（单位次）。 DN：实例启动至当前时间为止，该用户复杂作业read的次数之和（单位次）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
write_counts	bigint	CN：过去5秒内，该用户在所有DN上复杂作业write的次数之和（单位次）。 DN：实例启动至当前时间为止，该用户复杂作业write的次数之和（单位次）。
read_speed	double precision	CN：过去5秒内，该用户在单个DN上复杂作业read平均速率（单位KB/s）。 DN：过去5秒内，该用户在该DN上复杂作业read平均速率（单位KB/s）。
write_speed	double precision	CN：过去5秒内，该用户在单个DN上复杂作业write平均速率（单位KB/s）。 DN：过去5秒内，该用户在该DN上复杂作业write平均速率（单位KB/s）。

说明

- PG_TOTAL_USER_RESOURCE_INFO视图中统计的用户使用空间与磁盘实际使用空间可能存在约±5%的误差，用户可通过执行SELECT gs_wlm_readjust_user_space(0)语句进行修正。数据库也会定期（7天）对统计的用户使用空间进行修正。在执行该语句的过程中，如果在表中插入数据，可能会出现统计结果不够准确的情况。
- 执行DROP DATABASE操作后，PG_TOTAL_USER_RESOURCE_INFO视图中统计到的用户空间信息不会立即清理，在CPU使用率很高的情况下，最多需要1~2小时完成清理并恢复统计数据的准确性。

7.3.13.192 PG_TOTAL_USER_RESOURCE_INFO_OID

PG_TOTAL_USER_RESOURCE_INFO_OID视图显示所有用户的资源使用情况，需要使用管理员用户进行查询。此视图在GUC参数use_workload_manager为on时才有效。具体字段信息如**表7-478**所示。

表 7-478 PG_TOTAL_USER_RESOURCE_INFO_OID 字段

名称	类型	描述
userid	oid	用户id。
used_memory	integer	当前版本已废弃，查询结果无参考意义。
total_memory	integer	当前版本已废弃，查询结果无参考意义。
used_cpu	double precision	当前版本已废弃，查询结果无参考意义。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
total_cpu	integer	当前版本已废弃，查询结果无参考意义。
used_space	bigint	已使用的存储空间大小，单位KB。
total_space	bigint	可使用的存储空间大小，单位KB，值为-1表示未限制最大存储空间。
used_temp_space	bigint	已使用的临时空间大小，单位KB
total_temp_space	bigint	可使用的临时空间总大小，单位KB，值为-1表示未限制。
used_spill_space	bigint	已使用的下盘空间大小。单位KB。
total_spill_space	bigint	可使用的下盘空间总大小，单位KB，值为-1表示未限制。
read_kbytes	bigint	读磁盘数据量，单位KB。
write_kbytes	bigint	写磁盘数据量，单位KB。
read_counts	bigint	读磁盘次数。
write_counts	bigint	写磁盘次数。
read_speed	double precision	读磁盘速率，单位B/ms。
write_speed	double precision	写磁盘速率，单位B/ms。

7.3.13.193 PG_VARIABLE_INFO

PG_VARIABLE_INFO视图用于查询集群中当前节点的xid、oid的状态。具体字段信息如表7-479所示。

表 7-479 PG_VARIABLE_INFO 字段

名称	类型	描述
node_name	text	节点名称。
next_oid	oid	该节点下一次生成的oid。
next_xid	xid	该节点下一次生成的事务号。
oldest_xid	xid	该节点最旧的事务号。
xid_vac_limit	xid	强制autovacuum的临界点。
oldest_xid_db	oid	该节点datafrozenxid最小的数据库oid。
last_extend_csn_logpage	xid	最后一次扩展csnlog的页面号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
start_extend_csn_logpage	xid	csnlog扩展的起始页面号。
next_commit_seqno	xid	该节点下次生成的csn号。
latest_completed_xid	xid	该节点提交或者回滚后节点上的最新事务号。
startup_max_xid	xid	该节点关机前的最后一个事务号。

7.3.13.194 PG_VIEWS

PG_VIEWS视图显示数据库中每个视图的有用信息。具体字段信息如表7-480所示。

表 7-480 PG_VIEWS 字段

名称	类型	引用	描述
schemaname	name	PG_NAMESPACE.nspname	视图的模式名。
viewname	name	PG_CLASS.relname	视图名。
viewowner	name	PG_AUTHID.rolname	视图的所有者。
definition	text	-	视图的定义。

7.3.13.195 PGXC_COPY_ERROR_LOG

PGXC_COPY_ERROR_LOG系统视图用于捕捉COPY FROM过程的相关错误行，以便后续查阅。

表 7-481 PGXC_COPY_ERROR_LOG 字段

名称	类型	描述
relname	text	表名称。以模式名.表名形式显示。
begintime	timestamp with time zone	出现数据格式错误的时间。
filename	text	出现数据格式错误的数据源文件名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
lineno	bigint	在数据源文件中，出现数据格式错误的行号。
rawrecord	text	在数据源文件中，出现数据格式错误的原始记录。
detail	text	详细错误信息。
custom_id	text	关联的COPY语法中copy_custom_id输入的值，表示本次导入的自定义标识信息。

说明

- 查询pg_catalog.pgxc_copy_error_log系统视图时，显示所有CN和DN节点上pg_catalog.gs_copy_error_log表中的数据。等同于执行以下语句：
  ```
  SELECT * FROM pg_catalog.pgxc_copy_error_func(false, false);
  ```
- 视图默认不对pg_catalog.gs_copy_error_old表做UNION ALL操作，如果想要兼容查询pg_catalog.gs_copy_error_old表中的数据，可以执行以下语句进行查询：
  ```
  SELECT * FROM pg_catalog.pgxc_copy_error_func(true, false);
  ```
- 从此版本开始，pgxc_copy_error_log由系统表改成系统视图，因此无法对其执行TRUNCATE操作进行清理，执行TRUNCATE操作会抛出无法对视图进行此操作的相关报错。清理copy相关错误数据的两种情况如下。
  - 非三权分立下，需要系统管理员用户执行以下语句进行清理：
    ```
    TRUNCATE TABLE pg_catalog.gs_copy_error_log;
    ```
  - 三权分立下，使用普通用户清理同名schema下的pgxc_copy_error_log表即可。
- 三权分立机制下，错误日志信息不会记录至pg_catalog.pgxc_copy_error_log和pg_catalog.gs_copy_error_log。若用户load_user需记录错误信息，需通过以下语句在其同名Schema下创建：
  ```
  CREATE TABLE load_user.pgxc_copy_error_log (relname varchar, begintime timestamptz, filename varchar, lineno int8, rawrecord text, detail text);
  CREATE INDEX copy_error_log_relname_idx ON load_user.pgxc_copy_error_log(relname);
  ```
  （可选）在表上添加custom_id列：
  ```
  ALTER TABLE load_user.pgxc_copy_error_log ADD COLUMN custom_id varchar(64);
  ```

7.3.13.196 PGXC_GET_STAT_ALL_TABLES

PGXC_GET_STAT_ALL_TABLES视图显示各表的插入、更新、删除以及脏页率信息。对于高脏页率的系统表，建议在确认当前没有人操作该系统表时，再执行VACUUM FULL。此视图为GaussDB新增功能，升级到该版本后，升级前的插入、更新、删除信息不会被统计。建议对脏页率超过30%的非系统表执行VACUUM FULL，用户也可根据业务场景自行选择是否执行VACUUM FULL。该视图只有system admin和monitor admin用户有权限查看。具体字段信息如表7-482所示。

表 7-482 PGXC_GET_STAT_ALL_TABLES 字段

名称	类型	描述
relid	oid	表的oid。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relname	name	表名。
schemaname	name	表的模式名。
n_tup_ins	numeric	插入的元组条数。
n_tup_upd	numeric	更新的元组条数。
n_tup_del	numeric	删除的元组条数。
n_live_tup	numeric	存活元组的条数。
n_dead_tup	numeric	死亡元组的条数。在Ustore表中仅代表不活跃行指针数量。
dirty_page_rate	numeric(5,2)	表的脏页率信息(%)。

7.3.13.197 PGXC_GET_TABLE_SKEWNESS

PGXC_GET_TABLE_SKEWNESS视图显示当前数据库中表的数据分布倾斜情况。只有系统管理员权限可以访问。具体字段信息如表7-483所示。

表 7-483 PGXC_GET_TABLE_SKEWNESS 字段

名称	类型	描述
schemaname	name	表所在的模式名。
tablename	name	表名。
totalsize	numeric	表的总大小，单位Byte。
avgsize	numeric(1000,0)	表大小平均值(totalsize/DN个数，该值为平均分布的理想情况下，表在各DN占用空间大小)。
maxratio	numeric(4,3)	单DN表大小最大值占比（表在各DN占用空间的最大值/totalsize）。
minratio	numeric(4,3)	单DN表大小最小值占比（表在各DN占用空间的最小值/totalsize）。
skewsize	bigint	表分布倾斜值（单DN表大小最大值 - 单DN表大小最小值）。
skewratio	numeric(4,3)	表分布倾斜率（skewsize/totalsize）。
skewstddev	numeric(1000,0)	表分布标准方差（在表大小一定的情况下，该值越大表明表的整体分布情况越倾斜）。

版权所有 © 华为云计算技术有限公司

7.3.13.198 PGXC_NODE_ENV

PGXC_NODE_ENV视图显示集群中所有节点的环境变量信息。该视图只有monitor admin和sysadmin权限可以查看。具体字段信息如表7-484所示。

表 7-484 PGXC_NODE_ENV 字段

名称	类型	描述
node_name1	text	集群中节点的名称。
host1	text	集群中节点的主机名称。
process1	integer	集群中节点的进程号。
port1	integer	集群中节点的端口号。
installpath1	text	集群中节点的安装目录。
datapath1	text	集群中节点的数据目录。
log_directory1	text	集群中节点的日志目录。

7.3.13.199 PGXC_OS_THREADS

PGXC_OS_THREADS视图显示当前集群中所有正常节点下的线程状态信息。该视图只有monitor admin和sysadmin权限可以查看。具体字段信息如表7-485所示。

表 7-485 PGXC_OS_THREADS 字段

名称	类型	描述
node_name	text	当前集群中正常节点的名称。
pid	bigint	当前集群中正常节点进程中正在运行的线程号。
lwpid	integer	与pid对应的轻量级线程号。
thread_name	text	与pid对应的线程名称。
creation_time	timestamp with time zone	与pid对应的线程创建的时间。
dbid	oid	线程所属的数据库id。

7.3.13.200 PGXC_PREPARED_XACTS

PGXC_PREPARED_XACTS视图显示当前处于prepared阶段的两阶段事务。只有system admin和monitor admin用户有权限查看。具体字段信息如表7-486所示。

版权所有 © 华为云计算技术有限公司

表 7-486 PGXC_PREPARED_XACTS 字段

名称	类型	描述
pgxc_prepared_xact	text	当前处于prepared阶段的两阶段事务。

7.3.13.201 PGXC_RUNNING_XACTS

PGXC_RUNNING_XACTS视图显示集群中各个节点运行事务的信息，字段内容和PG_RUNNING_XACTS相同。只有system admin和monitor admin用户有权限查看。具体字段信息如**表7-487**所示。

表 7-487 PGXC_RUNNING_XACTS 字段

名称	类型	描述
handle	integer	事务在GTM对应的句柄。
gxid	xid	事务id号。
state	tinyint	事务状态。（3：prepared；0：starting。）
node	text	节点名称。
xmin	xid	节点上当前数据涉及的最小事务号xmin。
vacuum	boolean	表示当前事务是否是lazy vacuum事务。 ● t（true）：表示是。 ● f（false）：表示否。
timeline	bigint	数据库重启次数。
prepare_xid	xid	处于prepared状态事务的id号，若不在prepared状态，值为0。
pid	bigint	事务对应的线程id。
next_xid	xid	CN传给DN的事务id号。
dbid	oid	事务对应的数据库id。

7.3.13.202 PGXC_STAT_ACTIVITY

PGXC_STAT_ACTIVITY视图显示当前集群下所有CN的当前用户查询相关的信息，该视图只有monitor admin和sysadmin权限可以查看。505.0.0版本新增三列信息，灰度升级观察期中执行会报错，需使用PG_STAT_ACTIVITY直连目标节点进行查询。具体字段信息如**表7-488**所示。

版权所有 © 华为云计算技术有限公司

表 7-488 PGXC_STAT_ACTIVITY 字段

名称	类型	描述
coorname	text	当前集群下的CN名称。
datid	oid	用户会话在后台连接到的数据库OID。
datname	text	用户会话在后台连接到的数据库名称。
pid	bigint	后台线程id。
sessionid	bigint	会话id。
usesysid	oid	登录该后台的用户OID。
usename	text	登录该后台的用户名。
application_name	text	连接到该后台的应用名。
client_addr	inet	连接到该后台的客户端的IP地址。 如果此字段是null，它表明通过服务器机器上UNIX套接字连接客户端或者这是内部进程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
backend_start	timestamp with time zone	该过程开始的时间，即当客户端连接服务器的时间。
xact_start	timestamp with time zone	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	开始当前活跃查询的时间， 如果state的值不是active，则这个值是上一个查询的开始时间。
state_change	timestamp with time zone	上次状态改变的时间。
waiting	boolean	如果后台当前正等待锁则为true。否则为false。如果等待的锁发生在DN，则需要根据pg_thread_wait_status视图的wait_status字段确定发生的DN，在对应的DN查询pg_stat_activity的waiting字段确认是否发生了等待锁的行为。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
enqueue	text	语句当前排队状态。可能值是： ● waiting in queue：表示语句在排队中。 ● 空：表示语句正在运行。
state	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。 **说明** 只有系统管理员能查看到自己账户所对应的会话状态。其他账户的state信息为空。例如以judy用户连接数据库后，在pgxc_stat_activity中查看到的普通用户joe及初始用户omm的state信息为空： SELECT datname, usename, usesysid, state,pid FROM pgxc_stat_activity; datname | usename | usesysid | state | pid ----------+---------+----------+--------+----------------- testdb | omm | 10 | | 139968752121616 testdb | omm | 10 | | 139968903116560 db_tpcds | judy | 16398 | active | 139968391403280 testdb | omm | 10 | | 139968643069712 testdb | omm | 10 | | 139968680818448 testdb | joe | 16390 | | 139968563377936 (6 rows)
resource_pool	name	用户使用的资源池。
query_id	bigint	查询语句的id。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
connection_info	text	json格式字符串，记录当前连接数据库的驱动类型、驱动版本号、当前驱动的部署路径、进程属主用户等信息（参见GUC参数connection_info）。
global_sessionid	text	全局会话id。
unique_sql_id	bigint	语句的unique sql id。
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。
top_xid	xid	事务的顶层事务ID。
current_xid	xid	事务的当前事务ID。
xlog_quantity	bigint	事务当前使用的Xlog量，单位为字节。

7.3.13.203 PGXC_STAT_BAD_BLOCK

PGXC_STAT_BAD_BLOCK视图显示集群所有节点从启动后，在读取数据时出现Page校验失败的统计信息。该视图只有monitor admin和sysadmin权限可以查看。具体字段信息如表7-489所示。

表 7-489 PGXC_STAT_BAD_BLOCK 字段

名称	类型	描述
nodename	text	节点名。
databaseid	integer	数据库OID。
tablespaceid	integer	表空间OID。
relfilenode	integer	文件对象id。
forknum	integer	文件类型，具体取值如下所示： • 0：数据主文件。 • 1：FSM文件。 • 2：VM文件。 • 3：BCM文件。
error_count	integer	出现校验失败的次数。
first_time	timestamp with time zone	第一次出现的时间。
last_time	timestamp with time zone	最近一次出现的时间。

版权所有 © 华为云计算技术有限公司

7.3.13.204 PGXC_SQL_COUNT

PGXC_SQL_COUNT视图可用来查看SELECT、INSERT、UPDATE、DELETE、MERGE INTO五种SQL的节点级和用户级统计结果，识别当前业务负载较重的query类型，衡量整个集群和单个节点执行某种类型查询的能力。通过对以上五类SQL查询进行计数，获得指定时刻的统计结果，经计算可以得到指定QPS等统计信息。例如，T1时刻，USER1的SELECT计数结果为X1，T2时刻为X2，则可计算得到该用户SELECT查询的QPS值为(X2-X1)\(T2-T1)。由此，可获得集群用户级QPS曲线图和集群吞吐情况，追踪每个用户的业务负载是否发生剧烈变化。如果有剧烈变化，可以定位具体的语句类型(SELECT/INSERT/UPDATE/DELETE/MERGE INTO)。同时观测QPS曲线可以获知问题发生时间点，结合其它工具，定位问题点。能够为集群性能调优、问题定位等提供依据。该视图只有monitor admin和sysadmin权限可以查看。只能在CN上查询，不支持execute direct on (dn) 'select * from PGXC_SQL_COUNT';语句。

PGXC_SQL_COUNT视图的字段与GS_SQL_COUNT一致，具体请参见**GS_SQL_COUNT**的字段信息。

说明

当执行用户的MERGE INTO语句时，若能下推，在DN上收到的是MERGE INTO语句，将在DN节点上进行MERGE INTO类型计数，相应mergeinto_count列计数增加；若不能下推，在DN上收到的是UPDATE或INSERT语句，将在DN节点上进行UPDATE或INSERT类型计数，相应的update_count列或insert_count列计数增加。

7.3.13.205 PGXC_THREAD_WAIT_STATUS

PGXC_THREAD_WAIT_STATUS视图可用来查看集群全局各个节点上所有SQL语句产生的线程之间的调用层次关系，以及各个线程的阻塞等待状态，从而更容易定位进程停止响应问题以及类似现象的原因。具体字段信息如**表7-490**所示。

PGXC_THREAD_WAIT_STATUS视图和PG_THREAD_WAIT_STATUS视图列定义完全相同，这是由于PGXC_THREAD_WAIT_STATUS视图本质是到集群中各个节点上查询PG_THREAD_WAIT_STATUS视图汇总的结果。

表 7-490 PGXC_THREAD_WAIT_STATUS 字段

名称	类型	描述
node_name	text	当前节点的名称。
db_name	text	数据库名称。
thread_name	text	线程名称。
query_id	bigint	查询id，对应debug_query_id。
tid	bigint	当前线程的线程号。
sessionid	bigint	会话id。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	父会话id。
tlevel	integer	streaming线程的层级。
smpid	integer	并行线程的id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
wait_status	text	当前线程的等待状态。等待状态的详细信息请参见**表7-470**。
wait_event	text	如果wait_status是acquire lock、acquire lwlock、wait io三种类型，此列描述具体的锁、轻量级锁、I/O的信息。否则是空。
locktag	text	当前线程正在等待锁的信息。
lockmode	text	当前线程正等待获取的锁模式。
block_sessionid	bigint	阻塞当前线程获取锁的会话标识。
global_sessionid	text	全局会话id。

例如：

在coordinator1执行一条语句之后长时间没有响应。可以创建另外一个连接到coordinator1上，查询coordinator1上的线程状态。

```
gaussdb=# SELECT * FROM pg_thread_wait_status WHERE query_id > 0;
-[ RECORD 1 ]----+---------------------------------------
node_name        | cn_5001
db_name          | tpcc_row
thread_name      | PostgreSQL JDBC Driver
query_id         | 72620544050065400
tid              | 140650239031040
sessionid        | 11680
lwtid            | 26762
psessionid       |
tlevel           | 0
smpid            | 0
wait_status      | wait node: dn_6007_6008_6009, total 1
wait_event       | wait node
locktag          |
lockmode         |
block_sessionid  |
global_sessionid | 1120683504:11680#0
(1 rows)
gaussdb=# SELECT * FROM pgxc_thread_wait_status WHERE query_id > 0;
-[ RECORD 1 ]----+---------------------------------------------------
node_name        | cn_5001
db_name          | tpcc_row
thread_name      | PostgreSQL JDBC Driver
query_id         | 72620544050081616
tid              | 140648290055936
sessionid        | 11680
lwtid            | 26839
psessionid       |
tlevel           | 0
smpid            | 0
wait_status      | wait node: dn_6004_6005_6006, total 2
wait_event       | wait node
locktag          |
lockmode         |
block_sessionid  |
global_sessionid | 1120683504:11680#0
(1 rows)
```


版权所有 © 华为云计算技术有限公司

7.3.13.206 PGXC_TOTAL_MEMORY_DETAIL

PGXC_TOTAL_MEMORY_DETAIL视图显示集群内存使用情况。只能在CN上查询，不支持EXECUTE DIRECT ON (dn) 'SELECT * FROM PGXC_TOTAL_MEMORY_DETAIL';语句。该视图只有sysadmin和monitor admin权限可以查看。当GUC参数enable_memory_limit的值为off时，本视图不可用。具体字段信息如**表7-491**所示。

表 7-491 PGXC_TOTAL_MEMORY_DETAIL 字段

名称	类型	描述
nodename	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memorytype	text	内存使用的名称。 ● max_process_memory：GaussDB实例所占用的内存大小。 ● process_used_memory：GaussDB进程所使用的内存大小。 ● max_dynamic_memory：最大动态内存。 ● dynamic_used_memory：已使用的动态内存。 ● dynamic_peak_memory：内存的动态峰值。 ● dynamic_used_shrctx：最大动态共享内存上下文。 ● dynamic_peak_shrctx：共享内存上下文的动态峰值。 ● max_backend_memory：使用HA端口执行业务可使用的最大内存上限。 ● backend_used_memory：使用HA端口执行业务已使用的内存。 ● high_backend_thread_memory：高优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● high_backend_thread_max_memory：高优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_memory：中优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_max_memory：中优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_memory：低优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_max_memory：低优先级线程预留内存，enable_threads_memory_reserve=on时生效。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
		• max_shared_memory：最大共享内存。 • shared_used_memory：已使用的共享内存。 • max_sctpcomm_memory：通信库所允许使用的最大内存。 • sctpcomm_used_memory：通信库已使用的内存大小。 • sctpcomm_peak_memory：通信库的内存峰值。 • other_used_memory：其他已使用的内存大小。 • llvm_used_memory：当前系统中Codegen执行生成的且查询未释放的表达式IR所占用内存大小。
memorymbytes	integer	内存使用的大小，单位为MB。

7.3.13.207 PGXC_VARIABLE_INFO

PGXC_VARIABLE_INFO视图用于查询集群中所有节点的xid、oid的状态。该视图只有monitor admin和sysadmin权限可以查看。只能在CN上查询，不支持execute direct on (dn) 'select * from PGXC_VARIABLE_INFO';语句。具体字段信息如表7-492所示。

表 7-492 PGXC_VARIABLE_INFO 字段

名称	类型	描述
node_name	text	节点名称。
next_oid	oid	该节点下一次生成的oid。
next_xid	xid	该节点下一次生成的事务号。
oldest_xid	xid	该节点最旧的事务号。
xid_vac_limit	xid	强制autovacuum的临界点。
oldest_xid_db	oid	该节点datafrozenxid最小的数据库oid。
last_extend_csn_logpage	xid	最后一次扩展csnlog的页面号。
start_extend_csn_logpage	xid	csnlog扩展的起始页面号。
next_commit_seqno	xid	该节点下次生成的csn号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
latest_completed_xid	xid	该节点提交或者回滚后节点上的最新事务号。
startup_max_xid	xid	该节点关机前的最后一个事务号。

7.3.13.208 PLAN_TABLE

PLAN_TABLE显示用户通过执行EXPLAIN PLAN收集到的计划信息。计划信息的生命周期是session级别，session退出后相应的数据将被清除。同时不同session和不同user间的数据是相互隔离的。该视图同时存在PG_CATALOG和SYS schema下。

表 7-493 PLAN_TABLE 字段

名称	类型	描述
statement_id	character varying(30)	用户输入的查询标签。
plan_id	bigint	查询标识。
id	integer	查询生成的计划中的每一个执行算子的编号。
operation	character varying(30)	计划中算子的操作描述。
options	character varying(255)	操作选项。
object_name	name	操作对应的对象名，非查询中使用到的对象别名。来自于用户定义。
object_type	character varying(30)	对象类型。
object_owner	name	对象所属schema，来自于用户定义。
projection	character varying(4000)	操作输出的列信息。
cost	double precision	优化器对算子估算的执行代价。
cardinality	double precision	优化器对算子估算的结果行数。
remarks	character varying(4000)	暂不支持，值为NULL。
timestamp	date	暂不支持，值为NULL。 **说明** 在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该字段的类型会变更为datea。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
object_node	character varying(128)	暂不支持，值为NULL。
object_alias	character varying(261)	暂不支持，值为NULL。
object_instance	numeric	暂不支持，值为NULL。
optimizer	character varying(255)	暂不支持，值为NULL。
search_columns	numeric	暂不支持，值为NULL。
parent_id	numeric	暂不支持，值为NULL。
depth	numeric	暂不支持，值为NULL。
position	numeric	暂不支持，值为NULL。
bytes	numeric	暂不支持，值为NULL。
other_tag	character varying(255)	暂不支持，值为NULL。
partition_start	character varying(255)	暂不支持，值为NULL。
partition_stop	character varying(255)	暂不支持，值为NULL。
partition_id	numeric	暂不支持，值为NULL。
other	character varying	暂不支持，值为NULL。
other_xml	clob	暂不支持，值为NULL。
distribution	character varying(20)	暂不支持，值为NULL。
cpu_cost	numeric	暂不支持，值为NULL。
io_cost	numeric	暂不支持，值为NULL。
temp_space	numeric	暂不支持，值为NULL。
access_predicates	character varying(4000)	暂不支持，值为NULL。
filter_predicates	character varying(4000)	暂不支持，值为NULL。
time	numeric	暂不支持，值为NULL。
qblock_name	character varying(128)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

说明

- object_type取值范围为PG_CLASS中定义的relkind类型（TABLE普通表，INDEX索引，SEQUENCE序列，VIEW视图，FOREIGN TABLE外表，COMPOSITE TYPE复合类型，TOASTVALUE TOAST表）和计划使用到的rtekind(SUBQUERY, JOIN, FUNCTION, VALUES, CTE, REMOTE_QUERY)。
- object_owner对于RTE来说是计划中使用的对象描述，非用户定义的类型不存在object_owner。
- statement_id、object_name、object_owner、projection字段内容遵循用户定义的大小写存储，其它字段内容采用大写存储。
- 支持用户对PLAN_TABLE进行SELECT和DELETE操作，不支持其它DML操作。

7.3.13.209 PV_FILE_STAT

PV_FILE_STAT视图通过对数据文件I/O的统计，反映数据的I/O性能，用以发现I/O操作异常等性能问题。具体字段信息如表7-494所示。

表 7-494 PV_FILE_STAT 字段

名称	类型	描述
filenum	oid	文件标识。
dbid	oid	数据库标识。
spcid	oid	表空间标识。
phyrds	bigint	读物理文件的数目。
phywrts	bigint	写物理文件的数目。
phyblkrd	bigint	读物理文件块的数目。
phyblkwrt	bigint	写物理文件块的数目。
readtim	bigint	读文件的总时长，单位微秒。
writetim	bigint	写文件的总时长，单位微秒。
avgiotim	bigint	读写文件的平均时长，单位微秒。
lstiotim	bigint	最后一次读文件时长，单位微秒。
miniotim	bigint	读写文件的最小时长，单位微秒。
maxiowtm	bigint	读写文件的最大时长，单位微秒。

7.3.13.210 PV_INSTANCE_TIME

提供当前节点下的各种时间消耗信息，具体字段信息如表7-495所示，主要分为以下类型:

- DB_TIME: 作业在多核下的有效时间花费。
- CPU_TIME：CPU时间的消耗。
- EXECUTION_TIME：执行器内花费的时间。

版权所有 © 华为云计算技术有限公司

- PARSE_TIME：SQL解析的时间花费。
- PLAN_TIME：生成Plan的时间花费。
- REWRITE_TIME：SQL重写的时间消耗。
- PL_EXECUTION_TIME ： PL/SQL（存储过程）的执行时间.
- PL_COMPILATION_TIME：PL/SQL（存储过程）的编译时间。
- NET_SEND_TIME：网络上的时间花销。
- DATA_IO_TIME：I/O时间上的花销。

表 7-495 PV_INSTANCE_TIME 字段

名称	类型	描述
stat_id	integer	统计编号。
stat_name	text	类型名称。
value	bigint	时间值(单位：微秒)。

7.3.13.211 PV_OS_RUN_INFO

PV_OS_RUN_INFO视图显示当前操作系统运行的状态信息。具体字段信息如**表7-496**所示。

表 7-496 PV_OS_RUN_INFO 字段

名称	类型	描述
id	integer	编号。
name	text	操作系统运行状态名称。
value	numeric	操作系统运行状态值。
comments	text	操作系统运行状态注释。
cumulative	boolean	操作系统运行状态的值是否为累加值。

7.3.13.212 PV_REDO_STAT

PV_REDO_STAT视图显示会话线程的日志回放情况。具体字段信息如**表7-497**所示。

表 7-497 PV_REDO_STAT 字段

名称	类型	描述
phywrts	bigint	日志回放过程中写数据的次数。
phyblkwrt	bigint	日志回放过程中写数据的块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
writetim	bigint	日志回放过程中写数据消耗的总时间。
avgiotim	bigint	日志回放过程中写一次数据平均消耗的时间。
lstiotim	bigint	日志回放过程中最后一次写数据消耗的时间。
miniotim	bigint	日志回放过程中单次写数据消耗的最短时间。
maxiowtm	bigint	日志回放过程中单次写数据消耗的最长时间。

7.3.13.213 PV_SESSION_MEMORY

PV_SESSION_MEMORY视图显示Session级别的内存使用情况，包含执行作业在数据节点上由gaussdb线程和Stream线程分配的所有内存。具体字段信息如表7-498所示。

表 7-498 PV_SESSION_MEMORY 字段

名称	类型	描述
sessid	text	线程启动时间+线程标识。
init_mem	integer	当前正在执行作业进入执行器前已分配的内存，单位MB。
used_mem	integer	当前正在执行作业已分配的内存，单位MB。
peak_mem	integer	当前正在执行作业已分配的内存峰值，单位MB。

7.3.13.214 PV_SESSION_MEMORY_CONTEXT

PV_SESSION_MEMORY_CONTEXT视图显示所有会话的内存使用情况，以MemoryContext节点来统计。该视图仅在开启线程池（enable_thread_pool = on）时生效。

其中内存上下文“TempSmallContextGroup”，记录当前线程中所有内存上下文字段“totalsize”小于8192字节的信息汇总，并且内存上下文统计计数记录到“usedsize”字段中。所以在视图中，“TempSmallContextGroup”内存上下文中的“totalsize”和“freesize”是该线程中所有内存上下文“totalsize”小于8192字节的汇总总和，“usedsize”字段表示统计的内存上下文个数。具体字段信息如表7-499所示。

版权所有 © 华为云计算技术有限公司

表 7-499 PV_SESSION_MEMORY_CONTEXT 字段

名称	类型	描述
sessid	text	会话启动时间+会话标识（字符串信息为timestamp.sessionid）。
threadid	bigint	会话绑定的线程标识，如果未绑定线程，该值为-1。
contextname	text	内存上下文名称。
level	smallint	当前上下文在整体内存上下文中的层级。
parent	text	父内存上下文名称。
totalsize	bigint	当前内存上下文的内存总数，单位Byte。
freesize	bigint	当前内存上下文中已释放的内存总数，单位Byte。
usedsize	bigint	当前内存上下文中已使用的内存总数，单位Byte； “TempSmallContextGroup”内存上下文中该字段含义为统计计数。

须知

该视图为运维视图，用于定位内存问题时使用，不要并发查询该视图，并发查询该视图会随着并发数的增多导致新连接接入等待时间增加，长时间无法接入。

7.3.13.215 PV_SESSION_MEMORY_DETAIL

PV_SESSION_MEMORY_DETAIL视图显示会话的内存使用情况，以MemoryContext节点来统计。当开启线程池（enable_thread_pool = on）时，该视图包含所有的线程和会话的内存使用情况。

其中内存上下文“TempSmallContextGroup”，记录当前线程中所有内存上下文字段“totalsize”小于8192字节的信息汇总，并且内存上下文统计计数记录到“usedsize”字段中。所以在视图中，“TempSmallContextGroup”内存上下文中的“totalsize”和“freesize”是该线程中所有内存上下文“totalsize”小于8192字节的汇总总和，“usedsize”字段表示统计的内存上下文个数。

可通过SELECT * FROM PV_SESSION_MEMCTX_DETAIL(threadid, '');将某个线程所有内存上下文信息记录到$GAUSSLOG/gs_log/${node_name}/dumpmem目录下的threadid_timestamp.log文件中。其中threadid可通过表7-500sessid中获得。

版权所有 © 华为云计算技术有限公司

表 7-500 PV_SESSION_MEMORY_DETAIL 字段

名称	类型	描述
sessid	text	• 关闭线程池（enable_thread_pool = off）时该字段表示线程启动时间+session标识（字符串信息为timestamp.sessionid）。 • 开启线程池（enable_thread_pool = on）时，内存上下文是线程级别的，则对应的该字段表示线程启动时间+线程标识（字符串信息为timestamp.threadid），内存上下文是session级别的，则对应的该字段表示线程启动时间+session标识（字符串信息为timestamp.sessionid）。
sesstype	text	线程名称。
contextname	text	内存上下文名称。
level	smallint	当前上下文在整体内存上下文中的层级。
parent	text	父内存上下文名称。
totalsize	bigint	当前内存上下文的内存总数，单位Byte。
freesize	bigint	当前内存上下文中已释放的内存总数，单位Byte。
usedsize	bigint	当前内存上下文中已使用的内存总数，单位Byte； “TempSmallContextGroup”内存上下文中该字段含义为统计计数。

须知

该视图为运维视图，用于定位内存问题时使用，不要并发查询该视图，并发查询该视图会随着并发数的增多导致新连接接入等待时间增加，长时间无法接入。

7.3.13.216 PV_SESSION_STAT

PV_SESSION_STAT视图显示以会话线程或AutoVacuum线程为单位的统计会话状态信息。具体字段信息如**表7-501**所示。

版权所有 © 华为云计算技术有限公司

表 7-501 PV_SESSION_STAT 字段

名称	类型	描述
sessid	text	线程标识+线程启动时间。
statid	integer	统计编号。
statname	text	统计会话名称。
statunit	text	统计会话单位。
value	bigint	统计会话值。

7.3.13.217 PV_SESSION_TIME

PV_SESSION_TIME视图显示会话线程的运行时间信息及各执行阶段所消耗的时间。具体字段信息如表7-502所示。

表 7-502 PV_SESSION_TIME 字段

名称	类型	描述
sessid	text	线程标识+线程启动时间。
stat_id	integer	统计编号。
stat_name	text	会话类型名称。 • DB_TIME：作业在多核下的有效时间花费。 • CPU_TIME：CPU时间的消耗。 • EXECUTION_TIME：执行器内花费的时间。 • PARSE_TIME：SQL解析的时间花费。 • PLAN_TIME：生成Plan的时间花费。 • REWRITE_TIME：SQL重写的时间消耗。 • PL_EXECUTION_TIME：plpgsql（存储过程）的执行时间。 • PL_COMPILATION_TIME：plpgsql（存储过程）编译时间。 • NET_SEND_TIME：网络上的时间花销。 • DATA_IO_TIME：I/O时间上的花销。
value	bigint	会话值。

版权所有 © 华为云计算技术有限公司

7.3.13.218 PV_THREAD_MEMORY_CONTEXT

PV_THREAD_MEMORY_CONTEXT视图显示所有线程的内存使用情况，以MemoryContext节点来统计。该视图在关闭线程池（enable_thread_pool = off）时等价于PV_SESSION_MEMORY_DETAIL视图。具体字段信息如表7-503所示。

其中内存上下文“TempSmallContextGroup”，记录当前线程中所有内存上下文字段“totalsize”小于8192字节的信息汇总，并且内存上下文统计计数记录到“usedsize”字段中。所以在视图中，“TempSmallContextGroup”内存上下文中的“totalsize”和“freesize”是该线程中所有内存上下文“totalsize”小于8192字节的汇总总和，“usedsize”字段表示统计的内存上下文个数。

表 7-503 PV_THREAD_MEMORY_CONTEXT 字段

名称	类型	描述
threadid	text	线程启动时间+线程标识（字符串信息为timestamp.sessionid）。
tid	bigint	线程标识。
thrdtype	text	线程类型。可以是系统内存在的任何线程类型，如wlmmonitor等。
contextname	text	内存上下文名称。
level	smallint	当前上下文在整体内存上下文中的层级。
parent	text	父内存上下文名称。
totalsize	bigint	当前内存上下文的内存总数，单位Byte。
freesize	bigint	当前内存上下文中已释放的内存总数，单位Byte。
usedsize	bigint	当前内存上下文中已使用的内存总数，单位Byte； “TempSmallContextGroup”内存上下文中该字段含义为统计计数。

7.3.13.219 PV_TOTAL_MEMORY_DETAIL

PV_TOTAL_MEMORY_DETAIL视图显示当前数据库节点使用内存的信息，单位为MB。当GUC参数enable_memory_limit的值为off时，本视图不可用。具体字段信息如表7-504所示。

表 7-504 PV_TOTAL_MEMORY_DETAIL 字段

名称	类型	描述
nodename	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memorytype	text	内存类型，包括以下几种： ● max_process_memory：GaussDB集群实例所占用的内存大小。 ● process_used_memory：GaussDB进程所使用的内存大小。 ● max_dynamic_memory：最大动态内存。 ● dynamic_used_memory：已使用的动态内存。 ● dynamic_peak_memory：内存的动态峰值 ● dynamic_used_shrctx：已使用的动态共享内存上下文。 ● dynamic_peak_shrctx：共享内存上下文的动态峰值。 ● max_backend_memory：使用HA端口执行业务可使用的最大内存上限。 ● backend_used_memory：使用HA端口执行业务已使用的内存。 ● high_backend_thread_memory：高优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● high_backend_thread_max_memory：高优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_memory：中优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_max_memory：中优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_memory：低优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_max_memory：低优先级线程预留内存，enable_threads_memory_reserve=on时生效。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
		• max_shared_memory：最大共享内存。 • shared_used_memory：已使用的共享内存。 • max_sctpcomm_memory：通信库所允许使用的最大内存。 • sctpcomm_used_memory：通信库已使用的内存大小。 • sctpcomm_peak_memory：通信库的内存峰值。 • other_used_memory：其他已使用的内存大小。 • llvm_used_memory：当前系统中Codegen执行生成的且查询未释放的表达式IR所占用内存大小。
memorymbytes	integer	内存类型分配内存的大小。

7.3.13.220 SYS_DUMMY

SYS_DUMMY视图是数据库根据数据字典自动创建的，它只有一个文本字段，且只有一行，用于保存表达式计算结果。任何用户都可以访问它。该视图同时存在于PG_CATALOG和SYS schema下。具体字段信息如表7-505所示。

表 7-505 SYS_DUMMY 字段

名称	类型	描述
dummy	text	表达式计算结果。

7.3.13.221 V_INSTANCE

V_INSTANCE视图显示当前数据库的实例信息。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-506所示。

表 7-506 V_INSTANCE 字段

名称	类型	描述
instance_number	oid	当前数据库oid。
instance_name	character varying(16)	当前数据库名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
host_name	character varying(64)	主机名。
version	character varying(17)	暂不支持，值为NULL。
version_legacy	character varying(17)	暂不支持，值为NULL。
version_full	character varying(17)	暂不支持，值为NULL。
startup_time	date	暂不支持，值为NULL。
status	character varying(12)	暂不支持，值为NULL。
parallel	character varying(3)	暂不支持，值为NULL。
thread#	numeric	暂不支持，值为NULL。
archiver	character varying(7)	暂不支持，值为NULL。
log_switch_wait	character varying(15)	暂不支持，值为NULL。
logins	character varying(10)	暂不支持，值为NULL。
shutdown_pending	character varying(3)	暂不支持，值为NULL。
database_status	character varying(17)	暂不支持，值为NULL。
instance_role	character varying(18)	暂不支持，值为NULL。
active_state	character varying(9)	暂不支持，值为NULL。
blocked	character varying(3)	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。
instance_mode	character varying(11)	暂不支持，值为NULL。
edition	character varying(7)	暂不支持，值为NULL。
family	character varying(80)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
database_type	character varying(15)	暂不支持，值为NULL。

7.3.13.222 V_MYSTAT

V_MYSTAT视图显示数据库所有会话的统计信息。该视图只有系统管理员可以访问，普通用户需要授权才能访问，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-507所示。

表 7-507 V_MYSTAT 字段

名称	类型	描述
sid	numeric	当前会话的id。
statistic#	numeric	暂不支持，值为NULL。
value	numeric	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。

7.3.13.223 V_SESSION

V_SESSION视图显示当前所有会话的信息，该视图只有管理员可以访问，普通用户需要授权才能访问，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-508所示。

表 7-508 V_SESSION 字段

名称	类型	描述
saddr	raw	暂不支持，值为NULL。
sid	bigint	会话id。
serial#	integer	当前活动的后台线程的序号，在GaussDB中为0。
audsid	numeric	暂不支持，值为NULL。
paddr	raw	暂不支持，值为NULL。
schema#	numeric	暂不支持，值为NULL。
schemaname	name	登录该后台的用户名。
user#	oid	登录此后台线程的用户的OID。oid 为0表示此后台线程为全局辅助线程（auxiliary）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
username	name	登录此后台线程的用户名。username为空表示此后台线程为全局辅助线程（auxiliary）。
command	numeric	暂不支持，值为NULL。
ownerid	numeric	暂不支持，值为NULL。
taddr	character varying(16)	暂不支持，值为NULL。
lockwait	character varying(16)	暂不支持，值为NULL。
machine	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
sql_id	bigint	查询语句的id。
client_info	text	客户端信息。
event	text	语句当前排队状态。可能值是： ● waiting in queue：表示语句在排队中。 ● 空：表示语句正在运行。
sql_exec_start	timestamp with time zone	开始当前活跃查询的时间，如果state的值不是active，则这个值是上一个查询的开始时间。
program	text	连接到该后台的应用名。
status	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但事务中没有语句在执行。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。
server	character varying(9)	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pdml_status	character varying(8)	当前会话是否启用DML的并行执行。
port	numeric	当前会话的端口号。
process	character varying(24)	当前会话的进程号。
logon_time	date	当前会话的登录时间。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
last_call_et	integer	当前会话上次状态发生改变的时长。
osuser	character varying(128)	暂不支持，值为NULL。
terminal	character varying(30)	暂不支持，值为NULL。
type	character varying(10)	暂不支持，值为NULL。
sql_address	raw	暂不支持，值为NULL。
sql_hash_value	numeric	暂不支持，值为NULL。
sql_child_number	numeric	暂不支持，值为NULL。
sql_exec_id	numeric	暂不支持，值为NULL。
prev_sql_addr	raw	暂不支持，值为NULL。
prev_hash_value	numeric	暂不支持，值为NULL。
prev_sql_id	character varying(13)	暂不支持，值为NULL。
prev_child_number	numeric	暂不支持，值为NULL。
prev_exec_start	date	暂不支持，值为NULL。 • O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
prev_exec_id	numeric	暂不支持，值为NULL。
plsql_entry_object_id	numeric	暂不支持，值为NULL。
plsql_entry_subprogram_id	numeric	暂不支持，值为NULL。
plsql_object_id	numeric	暂不支持，值为NULL。
plsql_subprogram_id	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
module	character varying(64)	暂不支持，值为NULL。
module_hash	numeric	暂不支持，值为NULL。
action	character varying(64)	暂不支持，值为NULL。
action_hash	numeric	暂不支持，值为NULL。
fixed_table_sequence	numeric	暂不支持，值为NULL。
row_wait_obj#	numeric	暂不支持，值为NULL。
row_wait_file#	numeric	暂不支持，值为NULL。
row_wait_block#	numeric	暂不支持，值为NULL。
row_wait_row#	numeric	暂不支持，值为NULL。
top_level_call#	numeric	暂不支持，值为NULL。
pdml_enabled	character varying(3)	暂不支持，值为NULL。
failover_type	character varying(13)	暂不支持，值为NULL。
failover_method	character varying(10)	暂不支持，值为NULL。
failed_over	character varying(3)	暂不支持，值为NULL。
resource_consumer_group	character varying(32)	暂不支持，值为NULL。
pddl_status	character varying(8)	暂不支持，值为NULL。
pq_status	character varying(8)	暂不支持，值为NULL。
current_queue_duration	numeric	暂不支持，值为NULL。
client_identifier	character varying(64)	暂不支持，值为NULL。
blocking_session_status	character varying(11)	暂不支持，值为NULL。
blocking_instance	numeric	暂不支持，值为NULL。
blocking_session	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
final_blocking_session_status	character varying(11)	暂不支持，值为NULL。
final_blocking_instance	numeric	暂不支持，值为NULL。
final_blocking_session	numeric	暂不支持，值为NULL。
seq#	numeric	暂不支持，值为NULL。
event#	numeric	暂不支持，值为NULL。
p1text	character varying(64)	暂不支持，值为NULL。
p1	numeric	暂不支持，值为NULL。
p1raw	raw	暂不支持，值为NULL。
p2text	character varying(64)	暂不支持，值为NULL。
p2	numeric	暂不支持，值为NULL。
p2raw	raw	暂不支持，值为NULL。
p3text	character varying(64)	暂不支持，值为NULL。
p3	numeric	暂不支持，值为NULL。
p3raw	raw	暂不支持，值为NULL。
wait_class_id	numeric	暂不支持，值为NULL。
wait_class#	numeric	暂不支持，值为NULL。
wait_class	character varying(64)	暂不支持，值为NULL。
wait_time	numeric	暂不支持，值为NULL。
seconds_in_wait	numeric	暂不支持，值为NULL。
state	character varying(19)	暂不支持，值为NULL。
wait_time_micro	numeric	暂不支持，值为NULL。
time_remaining_micro	numeric	暂不支持，值为NULL。
time_since_last_wait_micro	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
service_name	character varying(64)	暂不支持，值为NULL。
sql_trace	character varying(8)	暂不支持，值为NULL。
sql_trace_waits	character varying(5)	暂不支持，值为NULL。
sql_trace_binds	character varying(5)	暂不支持，值为NULL。
sql_trace_plan_stats	character varying(10)	暂不支持，值为NULL。
session_edition_id	numeric	暂不支持，值为NULL。
creator_addr	raw	暂不支持，值为NULL。
creator_serial#	numeric	暂不支持，值为NULL。
ecid	character varying(64)	暂不支持，值为NULL。
sql_translation_profile_id	numeric	暂不支持，值为NULL。
pga_tunable_mem	numeric	暂不支持，值为NULL。
shard_ddl_status	character varying(8)	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。
external_name	character varying(1024)	暂不支持，值为NULL。
plsql_debugger_connected	character varying(5)	暂不支持，值为NULL。

说明

在初始化数据库时，若指定GUC参数mapping_date_to_datea的值为on，则该视图的logon_time字段类型会变更为datea。

7.3.13.224 V$GLOBAL_OPEN_CURSOR

V$GLOBAL_OPEN_CURSOR视图显示当前所有节点中所有打开的游标的相关信息。具体字段信息如表7-509所示。

版权所有 © 华为云计算技术有限公司

表 7-509 V$GLOBAL_OPEN_CURSOR 字段

名称	类型	描述
node_name	name	节点名称。
saddr	raw	暂不支持，值为NULL。
sid	numeric	会话id。
user_name	character varying(128)	用户名称。
address	raw	暂不支持，值为NULL。
hash_value	numeric	暂不支持，值为NULL。
sql_id	character varying(13)	查询语句的id。
sql_text	character varying(60)	游标的SQL文本的前60个字节。
last_sql_active_time	date	暂不支持，值为NULL。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
sql_exec_id	numeric	暂不支持，值为NULL。
cursor_type	character varying(64)	游标类型。 • OPEN-PL/SQL：打开的PL/SQL游标。 • OPEN：其他打开的游标。
child_address	raw	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。
dbid	oid	游标所属的数据库id。

7.3.13.225 V$GLOBAL_TRANSACTION

V$GLOBAL_TRANSACTION视图显示有关当前活动全局事务的信息。默认只有系统管理员权限才可以访问此系统视图，普通用户若要访问需要得到授权。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-510**所示。

表 7-510 V$GLOBAL_TRANSACTION 字段

名称	类型	描述
formatid	numeric	全局事务格式标识符。暂不支持，值为NULL。
globalid	raw	全局事务标识符。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
branchid	raw	全局事务分支标识符。暂不支持，值为NULL。全局事务的每一个单独事务称为分支。
branches	numeric	全局事务分支总数。
refcount	numeric	全局事务的同级数（必须与分支相同）。
preparecount	numeric	已准备的全局事务分支数。当system_view_version参数大于0时，不存在已准备的全局事务分支时为0，否则为NULL。
state	character varying(38)	全局事务的分支的状态。
flags	numeric	状态的数字表示形式。
coupling	character varying(15)	表示分支是自由(`FREE`)、松散耦合(`LOOSELY COUPLED`)、还是紧密耦合(TIGHTLY COUPLED)。暂不支持，值为NULL。
con_id	numeric	与数据相关的容器的id。暂不支持，值为0。

7.3.13.226 V$LOCK

V$LOCK视图显示各打开事务所持有的锁信息。默认只有系统管理员权限才可以访问此系统视图，普通用户若要访问需要得到授权。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-511**所示。

表 7-511 V$LOCK 字段

名称	类型	描述
addr	raw	暂不支持，值为NULL。
kaddr	raw	暂不支持，值为NULL。
sid	numeric	拥有锁的sessionid。
type	character varying(2)	TM或TX。 • TM对应pg_locks中的relation锁。 • TX对应其余锁。
id1	numeric	• TM锁：对象号对应relationId。 • TX锁：暂不支持，值为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
id2	numeric	• TM锁：默认为0。 • TX锁：暂不支持，值为0。
lmode	numeric	具体信息如表7-512所示。
request	numeric	具体信息如表7-512所示。
ctime	numeric	暂不支持，值为NULL。
block	numeric	是否被其余session阻塞，1表示阻塞，0表示未阻塞。
con_id	numeric	暂不支持，值为NULL。
blocksid	bigint	当前线程阻塞的另一个线程ID。

表 7-512 锁模式

级别	GaussDB
-	NULL
0	INVALID
1	AccessShare
2	RowShare
3	RowExclusive
4	ShareUpdateExclusive
5	Share
6	ShareRowExclusive
7	Exclusive
8	AccessExclusive

7.3.13.227 V$NLS_PARAMETERS

V$NLS_PARAMETERS视图显示数据库当前配置的（National Language Support）NLS参数和参数的值。所有用户都可以访问，该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-513所示。

版权所有 © 华为云计算技术有限公司

表 7-513 V$NLS_PARAMETERS 字段

名称	类型	描述
parameter	character varying(64)	（National Language Support）NLS参数名。
value	character varying(64)	（National Language Support）NLS参数的值。
con_id	numeric	暂不支持，值为0。

7.3.13.228 V$OPEN_CURSOR

V$OPEN_CURSOR视图显示当前节点所有会话打开的游标的相关信息。默认只有系统管理员权限才可以访问此系统视图，普通用户若要访问需要得到授权。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-514所示。

表 7-514 V$OPEN_CURSOR 字段

名称	类型	描述
saddr	raw	暂不支持，值为NULL。
sid	numeric	会话id。
user_name	character varying(128)	用户名称。
address	raw	暂不支持，值为NULL。
hash_value	numeric	暂不支持，值为NULL。
sql_id	character varying(13)	查询语句的id。
sql_text	character varying(60)	游标的SQL文本的前60个字节。
last_sql_active_time	date	暂不支持，值为NULL。 O兼容性库下显示为TIMESTAMP(0) WITHOUT TIME ZONE。
sql_exec_id	numeric	暂不支持，值为NULL。
cursor_type	character varying(64)	游标类型。 • OPEN-PL/SQL：打开的PL/SQL游标。 • OPEN：其他打开的游标。
child_address	raw	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为NULL。
dbid	oid	游标所属的数据库id。

版权所有 © 华为云计算技术有限公司

7.3.13.229 V$SESSION_WAIT

V$SESSION_WAIT视图显示了每一个用户每一个会话的当前正在等待的事件或者最后一次等待的事件。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如**表7-515**所示。

表 7-515 V$SESSION_WAIT 字段

名称	类型	描述
sid	numeric	会话识别标记，映射到V$SESSION.SID字段。
seq#	numeric	暂不支持，值为NULL。
event	character varying(64)	如果会话在等待中，则显示目前在等待的资源或者事件，如果会话没有在等待，则显示最后一次等待的资源或者事件。
p1text	character varying(64)	暂不支持，值为NULL。
p1	numeric	暂不支持，值为NULL。
p1raw	raw	暂不支持，值为NULL。
p2text	character varying(64)	暂不支持，值为NULL。
p2	numeric	暂不支持，值为NULL。
p2raw	raw	暂不支持，值为NULL。
p3text	character varying(64)	暂不支持，值为NULL。
p3	numeric	暂不支持，值为NULL。
p3raw	raw	暂不支持，值为NULL。
wait_class_id	numeric	暂不支持，值为NULL。
wait_class#	numeric	暂不支持，值为NULL。
wait_class	character varying(64)	等待事件的种类命名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
wait_time	numeric	如果会话当前正在等待，则值为 0。如果会话不在等待中，则值如下所示： ● >0：值是最后一次等待的持续时间（以百分之一秒为单位）。 ● -1：最后一次等待的持续时间不到百分之一秒。 ● -2：参数TIMED_STATISTICS设置为false。 此列已被弃用，取而代之的是列为WAIT_TIME_MICRO和STATE。
second_in_wait	numeric	如果会话当前正在等待，则该值是等待当前等待的时间量。如果会话未处于等待状态，则该值是自上次等待开始以来的时间量。 此列已被弃用，取而代之的是WAIT_TIME_MICRO列和TIME_SINCE_LAST_WAIT_MICRO列。
state	character varying(64)	等待状态： ● WAITING：会话当前正在等待。 ● WAITED UNKNOWN TIME：最后等待的持续时间未知；这是参数 TIMED_STATISTICS 设置为 false 时的值。 ● WAITED SHORT TIME：最后一次等待不到百分之一秒。 ● WAITED KNOWN TIME：在WAIT_TIME列中指定的最后一次等待的持续时间。
wait_time_micro	numeric	等待的时间（以微秒为单位）。如果会话当前正在等待，则该值是在当前等待中花费的时间。如果会话当前未处于等待状态，则该值是上次等待的等待时间量。
time_remaining_micro	numeric	暂不支持，值为NULL。
time_since_last_wait_micro	numeric	自上次等待结束以来经过的时间（以微秒为单位）。如果会话当前处于等待状态，则值为 0。
con_id	numeric	暂不支持，值为0。

版权所有 © 华为云计算技术有限公司

7.3.13.230 V$SYSSTAT

V$SYSSTAT视图显示自数据库实例启动那刻起就开始累计的全实例的资源使用情况。默认只有初始用户或监控管理员可以访问，其它用户需授予MONADMIN权限才可访问。该视图同时存在于PG_CATALOG和SYS Schema下。在ARM环境下，需先开启内存资源管理功能，详细操作请联系管理员处理。具体字段信息如表7-516所示。

表 7-516 V$SYSSTAT 字段

名称	类型	描述
statistic#	numeric	统计编号。
name	character varying(64)	统计名称。
class	numeric	暂不支持，值为NULL。
value	numeric	统计值。
stat_id	numeric	暂不支持，值为NULL。
con_id	numeric	暂不支持，值为0。

7.3.13.231 V$SYSTEM_EVENT

V$SYSTEM_EVENT视图显示有关事件总等待的信息（自实例启动后各个等待事件的概括）。默认只有系统管理员权限才可以访问此系统视图，普通用户需要授权才可以访问。该视图同时存在于PG_CATALOG和SYS Schema下。具体字段信息如表7-517所示。

表 7-517 V$SYSTEM_EVENT 字段

名称	类型	描述
event	character varying(64)	等待事件的名称。
total_waits	numeric	等待事件的总次数。
total_timeouts	numeric	事件的超时总数。
time_waited	numeric	等待事件的总时间（以百分之一秒为单位）。
average_wait	numeric	等待事件的平均时间（以百分之一秒为单位）。
time_waited_micro	numeric	等待事件的总时间（以微秒为单位）。
total_waits_fg	numeric	暂不支持，值为NULL。
total_timeouts_fg	numeric	暂不支持，值为NULL。
time_waited_fg	numeric	暂不支持，值为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
average_wait_fg	numeric	暂不支持，值为NULL。
time_waited_micro_fg	numeric	暂不支持，值为NULL。
event_id	numeric	暂不支持，值为NULL。
wait_class_id	numeric	暂不支持，值为NULL。
wait_class#	numeric	暂不支持，值为NULL。
wait_class	character varying(64)	等待事件的等待类名称。
con_id	numeric	暂不支持，值为0。

7.3.13.232 V$VERSION

V$VERSION视图显示数据库的版本号。所有用户都可以访问，该视图同时存在于PG_CATALOG和SYS schema下。具体字段信息如表7-518所示。

表 7-518 V$VERSION 字段

名称	类型	描述
banner	character varying(80)	组件名称和版本号。
banner_full	character varying(160)	显示数据库版本和版本号。
banner_legacy	character varying(80)	显示数据库版本。
con_id	numeric	暂不支持，置0。

7.3.14 废弃系统视图/系统表

- GaussDB中下列视图在当前版本中已废弃：
  - GS_IMCU_SLOT_STATUS
  - GS_IMCV_STATUS
  - GS_DATABASE_DETAILS
  - MTD_USER
- GaussDB中下列表在当前版本中已废弃：
  - GS_PDB
  - GS_RESOURCE_PLANS
  - GS_RESOURCE_PLAN_DIRECTIVES

版权所有 © 华为云计算技术有限公司

8 Schema

GaussDB的Schema如表8-1所示。

说明

数据库禁止在提供功能接口的Schema下创建用户的业务数据，同样也不允许依赖这些Schema下的函数、数据类型等系统对象，包括但不限于表、函数等(dbe_*, pkg_*）。

表 8-1 GaussDB 支持的 Schema

Schema名称	描述
blockchain	用于存储账本数据库特性中创建防篡改表时自动创建的用户历史表。
dbe_perf	DBE_PERF Schema内视图主要用来诊断性能问题，也是WDR Snapshot的数据来源。数据库安装后，默认只有初始用户和监控管理员具有模式dbe_perf的权限，有权查看该模式下的视图和函数。
snapshot	用于管理WDR snapshot的相关的数据信息，默认初始化用户或监控管理员用户可以访问。
sqladvsior	用于分布列推荐，具体使用方法见分布列推荐函数。
sys	用于提供系统信息视图接口。
pg_catalog	用于维护系统的catalog信息，包含系统表和所有内置数据类型、函数、操作符。
public	公共模式，用于存储公共对象。search_path参数缺省时，如果存在用户同名的模式则将创建的表（以及其他对象）默认创建到同名模式下，不存在用户同名模式则自动放入public模式。
pkg_service	用于管理package服务相关信息。
pkg_util	用于管理package工具相关信息。
dbe_raw	高级功能包dbe_raw，用于raw类型数据的转化、取子串、求长度等操作。
dbe_session	高级功能包dbe_session，用于设置指定属性的值，并支持用户查询校验。

版权所有 © 华为云计算技术有限公司

Schema名称	描述
dbe_lob	高级功能包dbe_lob，用于大文件（clob/blob）的读取、写入、复制等操作。
dbe_match	高级功能包dbe_match，用于字符串相似度比较。
dbe_task	高级功能包dbe_task，用于作业任务的调度包括提交任务、取消任务、同步任务状态、更新任务信息等可以使数据库定期执行特定的任务。
dbe_sql	高级功能包dbe_sql，用于执行动态sql，可以在应用的运行时间构建查询和其它的命令。
dbe_file	高级功能包dbe_file，用于数据库外部文件的读取、复制、写入、删除、重命名等。
dbe_output	高级功能包dbe_output，用于打印输出信息。
dbe_random	高级功能包dbe_random，用于生成随机种子和随机数。
dbe_application_info	高级功能包dbe_application_info，用于记录客户端信息。
dbe_utility	高级功能包dbe_utility，用于存储过程调用调试工具，例如打印错误堆栈等。
dbe_scheduler	高级功能包dbe_scheduler，用于创建定时任务，通过程序（program）、调度（schedule）使数据库定期执行特定的任务。也可以通过授权、提供证书执行数据库外部任务。
information_schema	用于存储有关当前数据库中定义的对象的信息。
dbe_sql_util	SQL运维功能，目前包含SQL Patch的运维接口。
dbe_xmlgen	高级功能包dbe_xmlgen，用于将查询结果转换成xml字符串。
dbe_describe	高级功能包dbe_describe，用于查询存储过程或函数的参数信息。
dbe_stats	高级功能包dbe_stats，用于管理统计信息。

表 8-2 GaussDB 目前禁用的 Schema

Schema名称	描述
dbe_pldebugger	用于调试PL/SQL函数及存储过程，目前暂不支持，该视图下接口调用报错unsupported。
db4ai	用于管理AI训练中不同版本的数据信息。

8.1 Information Schema

信息模式本身是一个名为information_schema的模式。这个模式自动存在于所有数据库中。信息模式由一组视图构成，它们包含定义在当前数据库中对象的信息。这个模

版权所有 © 华为云计算技术有限公司

式的拥有者是初始数据库用户，但是所有用户仅有使用权限，没有创建表、函数等对象的权限。

信息模式兼容PG，包括：constraint_table_usage、domain_constraints、domain_udt_usage、domains、enabled_roles、key_column_usage、parameters、referential_constraints、applicable_roles、administrable_role_authorizations、attributes、character_sets、check_constraint_routine_usage、check_constraints、collations、collation_character_set_applicability、column_domain_usage、column_privileges、column_udt_usage、columns、constraint_column_usage、role_column_grants、routine_privileges、role_routine_grants、routines、schemata、sequences、table_constraints、table_privileges、role_table_grants、tables、triggered_update_columns、udt_privileges、role_udt_grants、usage_privileges、role_usage_grants、user_defined_types、view_column_usage、view_routine_usage、view_table_usage、views、data_type_privileges、element_types、column_options、foreign_data_wrapper_options、foreign_data_wrappers、foreign_server_options、foreign_servers、foreign_table_options、foreign_tables、user_mapping_options、user_mappings、sql_features、sql_implementation_info、sql_languages、sql_packages、sql_parts、sql_sizing、sql_sizing_profiles。

下面章节只显示未在上述描述内的视图信息。

须知

只有初始用户可通过CREATE OR REPLACE语法重建information_schema模式下的视图，可能产生越权行为，请谨慎操作。例如，在重建视图定义时绑定自定义函数，有可能被用于实现提权。

8.1.1 INFORMATION_SCHEMA._PG_FOREIGN_DATA_WRAPPERS

显示外部数据封装器的信息，如表8-3所示。该视图只有sysadmin权限可以查看。

表 8-3 INFORMATION_SCHEMA._PG_FOREIGN_DATA_WRAPPERS 字段

名称	类型	描述
oid	oid	外部数据封装器的oid。
fdwowner	oid	外部数据封装器的所有者的oid。
fdwoptions	text[]	外部数据封装器指定选项，使用“keyword=value”格式的字符串。
foreign_data_wrapper_catalog	information_schema.sql_identifier	外部封装器所在的数据库名称（永远为当前数据库）。
foreign_data_wrapper_name	information_schema.sql_identifier	外部数据封装器名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
authorization_identifier	information_schema.sql_identifier	外部数据封装器所有者的角色名称。
foreign_data_wrapper_language	information_schema.character_data	外部数据封装器的实现语言。

8.1.2 INFORMATION_SCHEMA._PG_FOREIGN_SERVERS

显示外部服务器的信息，如表8-4所示。该视图只有sysadmin权限可以查看。

表 8-4 INFORMATION_SCHEMA._PG_FOREIGN_SERVERS 字段

名称	类型	描述
oid	oid	外部服务器的oid。
srvoptions	text[]	外部服务器指定选项，使用“keyword=value”格式的字符串。
foreign_server_catalog	information_schema.sql_identifier	外部服务器所在database名称（永远为当前数据库）。
foreign_server_name	information_schema.sql_identifier	外部服务器名称。
foreign_data_wrapper_catalog	information_schema.sql_identifier	外部数据封装器所在database名称（永远为当前数据库）。
foreign_data_wrapper_name	information_schema.sql_identifier	外部数据封装器名称。
foreign_server_type	information_schema.character_data	外部服务器的类型。
foreign_server_version	information_schema.character_data	外部服务器的版本。
authorization_identifier	information_schema.sql_identifier	外部服务器的所有者的角色名称。

8.1.3 INFORMATION_SCHEMA._PG_FOREIGN_TABLE_COLUMNS

显示外部表的列信息，如表8-5所示。该视图只有sysadmin权限可以查看。

版权所有 © 华为云计算技术有限公司

表 8-5 INFORMATION_SCHEMA._PG_FOREIGN_TABLE_COLUMNS 字段

名称	类型	描述
nspname	name	schema名称。
relname	name	表名称。
attname	name	列名称。
attfdwoptions	text[]	外部数据封装器的属性选项，使用“keyword=value”格式的字符串。

8.1.4 INFORMATION_SCHEMA._PG_FOREIGN_TABLES

存储所有定义在本数据库的外部表的信息，如表8-6所示。只显示当前用户有权访问的外部表的信息。该视图只有sysadmin权限可以查看。

表 8-6 INFORMATION_SCHEMA._PG_FOREIGN_TABLES 字段

名称	类型	描述
foreign_table_catalog	information_schema.sql_identifier	外部表所在的数据库名称（永远是当前数据库）。
foreign_table_schema	name	外部表的schema名称。
foreign_table_name	name	外部表的名称。
ftoptions	text[]	外部表的可选项。
foreign_server_catalog	information_schema.sql_identifier	外部服务器所在的数据库名称（永远是当前数据库）。
foreign_server_name	information_schema.sql_identifier	外部服务器的名称。
authorization_identifier	information_schema.sql_identifier	所有者的角色名称。

8.1.5 INFORMATION_SCHEMA._PG_USER_MAPPINGS

存储从本地用户到远程的映射，如表8-7所示。该视图只有sysadmin权限可以查看。

表 8-7 INFORMATION_SCHEMA._PG_USER_MAPPINGS 字段

名称	类型	描述
oid	oid	从本地用户到远程的映射的oid。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
umoptions	text[]	用户映射指定选项，使用“keyword=value”格式的字符串。
umuser	oid	被映射的本地用户的oid，如果用户映射是公共的则为0。
authorization_identifier	information_schema.sql_identifier	本地用户角色名称。
foreign_server_catalog	information_schema.sql_identifier	外部服务器所在的数据库名称（永远是当前数据库）。
foreign_server_name	information_schema.sql_identifier	外部服务器名称。
srvowner	information_schema.sql_identifier	外部服务器所有者。

8.1.6 INFORMATION_SCHEMA.INFORMATION_SCHEMA_CATALOG_NAME

用来显示当前所在的database的名称，如表8-8所示。

表 8-8 INFORMATION_SCHEMA.INFORMATION_SCHEMA_CATALOG_NAME 字段

名称	类型	描述
catalog_name	information_schema.sql_identifier	当前database的名称。

8.1.7 INFORMATION_SCHEMA.TRIGGERS

显示所有定义在当前数据库中表和视图上的触发器，并且只显示当前用户拥有的触发器或当前用户在其上具有某种除了SELECT之外特权的触发器，如表8-9所示。

表 8-9 INFORMATION_SCHEMA.TRIGGERS 字段

名称	类型	描述
trigger_catalog	information_schema.sql_identifier	包含该触发器的数据库名。
trigger_schema	information_schema.sql_identifier	包含该触发器的模式名。
trigger_name	information_schema.sql_identifier	触发器名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
event_manipulation	information_schema.character_data	触发该触发器的事件类型（INSERT、UPDATE、DELETE）。
event_object_catalog	information_schema.sql_identifier	包含该触发器所在表的数据库名。
event_object_schema	information_schema.sql_identifier	包含该触发器所在表的模式名。
event_object_table	information_schema.sql_identifier	触发器所在表的名称。
action_order	information_schema.cardinal_number	同一个表上具有相同event_manipulation、action_timing、action_orientation的触发器之间的触发顺序。
action_condition	information_schema.character_data	触发器的WHEN条件，默认为空。
action_statement	information_schema.character_data	触发器执行的语句。
action_orientation	information_schema.character_data	表示触发器是对每个被处理的行触发一次还是为每个语句触发一次。
action_timing	information_schema.character_data	表示触发器在什么时候触发（BEFORE、AFTER、INSTEAD OF）。
action_reference_old_table	information_schema.sql_identifier	无实际意义，取值为空。
action_reference_new_table	information_schema.sql_identifier	无实际意义，取值为空。
action_reference_old_row	information_schema.sql_identifier	无实际意义，取值为空。
action_reference_new_row	information_schema.sql_identifier	无实际意义，取值为空。
created	information_schema.time_stamp	无实际意义，取值为空。

说明

INFORMATION_SCHEMA.TRIGGERS视图只显示事件类型为INSERT、UPDATE、DELETE的触发器，不显示事件类型为TRUNCATE的触发器。

版权所有 © 华为云计算技术有限公司

8.2 DBE_PERF Schema

DBE_PERF Schema内视图主要用来诊断性能问题，也是WDR Snapshot的数据来源。数据库安装后，默认只有初始用户和监控管理员具有模式dbe_perf的权限，有权查看该模式下的视图和函数。若是由老版本升级而来，为保持权限的前向兼容，模式dbe_perf的权限与老版本保持一致，当前版本禁止所有用户在该模式下创建操作符，已创建的操作符不影响使用。从OS、Instance、Memory等多个维度划分组织视图，并且符合如下命名规范：

- GLOBAL_开头的视图，代表从CN/DN请求数据，并将数据追加对外返回，不会处理数据。
- SUMMARY_开头的视图，代表是将集群内的数据概述，多数情况下是返回CN/DN（有时只有CN的）的数据，会对数据进行加工和汇聚。
- 非这两者开头的视图，一般代表本地视图，不会向其它CN/DN请求数据。

8.2.1 OS

8.2.1.1 DBE_PERF.OS_RUNTIME

显示当前操作系统运行的状态信息，如表8-10所示。

表 8-10 DBE_PERF.OS_RUNTIME 字段

名称	类型	描述
id	integer	编号。
name	text	操作系统运行状态名称。
value	numeric	操作系统运行状态值。
comments	text	操作系统运行状态注释。
cumulative	boolean	操作系统运行状态的值是否为累加值。

8.2.1.2 DBE_PERF.GLOBAL_OS_RUNTIME

提供整个集群中所有正常节点下的操作系统运行状态信息，如表8-11所示。

表 8-11 DBE_PERF.GLOBAL_OS_RUNTIME 字段

名称	类型	描述
node_name	name	节点名称。
id	integer	编号。
name	text	操作系统运行状态名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
value	numeric	操作系统运行状态值。
comments	text	操作系统运行状态注释。
cumulative	boolean	操作系统运行状态的值是否为累加值。

8.2.1.3 DBE_PERF.OS_THREADS

提供当前节点下所有线程的状态信息，如表8-12所示。

表 8-12 DBE_PERF.OS_THREADS 字段

名称	类型	描述
node_name	text	当前节点的名称。
pid	bigint	当前节点进程中正在运行的线程号。
lwpid	integer	与pid对应的轻量级线程号。
thread_name	text	与pid对应的线程名称。
creation_time	timestamp with time zone	与pid对应的线程创建的时间。
dbid	oid	线程所属的数据库ID。

8.2.1.4 DBE_PERF.GLOBAL_OS_THREADS

提供整个集群中所有正常节点下的线程状态信息，如表8-13所示。

表 8-13 DBE_PERF.GLOBAL_OS_THREADS 字段

名称	类型	描述
node_name	text	节点名称。
pid	bigint	当前节点进程中正在运行的线程号。
lwpid	integer	与pid对应的轻量级线程号。
thread_name	text	与pid对应的线程名称。
creation_time	timestamp with time zone	与pid对应的线程创建的时间。
dbid	oid	线程所属的数据库ID。

版权所有 © 华为云计算技术有限公司

8.2.1.5 DBE_PERF.PERF_QUERY

提供当前节点下采集到堆栈信息的名称、树状结构和百分比，如表8-14所示。需要有monadmin权限。

使用该视图查询堆栈信息之前，需要先执行gs_perf_start函数，采集火焰图数据。

表 8-14 DBE_PERF.PERF_QUERY 字段

名称	类型	描述
backtrace	text	堆栈信息树状结构文本。
overhead	double precision	当前堆栈信息在整个堆栈采集过程中所占时间比重。

8.2.1.6 DBE_PERF.SHAKING_COLLECT_STATUS

在CN上执行时返回所有节点的性能抖动监控状态，在DN上执行时返回本实例的性能抖动状态。需要有monadmin权限。监控状态如表8-15所示。升级部分分片时，该新增视图不可用。

表 8-15 DBE_PERF.SHAKING_COLLECT_STATUS 返回值

名称	类型	描述
nodename	text	节点名称。
status	text	该节点所在机器是否已开启性能抖动监控。 • NOT IN SHAKE MODE：未在监控中。 • IN SHAKE MODE：已在监控中。 • NO PERF SERVICE：采集进程未响应。 • Unexpected Result: get_shaking_connect_status error. Detail: 详细报错消息：表示查询失败。

8.2.2 Instance

8.2.2.1 DBE_PERF.INSTANCE_TIME

提供当前集群节点下的各种时间消耗信息，如表8-16所示。

表 8-16 DBE_PERF.INSTANCE_TIME 字段

名称	类型	描述
stat_id	integer	统计编号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stat_name	text	类型名称。 • DB_TIME：作业在多核下的有效时间花销。 • CPU_TIME：CPU的时间花销。 • EXECUTION_TIME：执行器内的时间花销。 • PARSE_TIME：SQL解析的时间花销。 • PLAN_TIME：生成Plan的时间花销。 • REWRITE_TIME：SQL重写的时间花销。 • PL_EXECUTION_TIME：PL/SQL（存储过程）执行的时间花销。 • PL_COMPILATION_TIME：PL/SQL（存储过程）编译的时间花销。 • NET_SEND_TIME：网络上的时间花销。 • DATA_IO_TIME：I/O上的时间花销。
value	bigint	时间值（单位：微秒）。

8.2.2.2 DBE_PERF.GLOBAL_INSTANCE_TIME

提供整个集群中所有正常节点下的各种时间消耗信息，如表8-17所示。

表 8-17 DBE_PERF.GLOBAL_INSTANCE_TIME 字段

名称	类型	描述
node_name	name	节点名称。
stat_id	integer	统计编号。
stat_name	text	类型名称。 • DB_TIME：作业在多核下的有效时间花销。 • CPU_TIME：CPU的时间花销。 • EXECUTION_TIME：执行器内的时间花销。 • PARSE_TIME：SQL解析的时间花销。 • PLAN_TIME：生成Plan的时间花销。 • REWRITE_TIME：SQL重写的时间花销。 • PL_EXECUTION_TIME：PL/SQL（存储过程）执行的时间花销。 • PL_COMPILATION_TIME：PL/SQL（存储过程）编译的时间花销。 • NET_SEND_TIME：网络上的时间花销。 • DATA_IO_TIME：I/O上的时间花销。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
value	bigint	时间值（单位：微秒）。

8.2.3 Memory

8.2.3.1 DBE_PERF.MEMORY_NODE_DETAIL

显示当前数据库节点内存使用情况，如表8-18所示。

表 8-18 DBE_PERF.MEMORY_NODE_DETAIL 字段

名称	类型	描述
nodename	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memorytype	text	内存的名称。 ● max_process_memory：数据库节点可用内存的最大值。 ● process_used_memory：进程所使用的内存大小。 ● max_dynamic_memory：最大动态内存。 ● dynamic_used_memory：已使用的动态内存。 ● dynamic_peak_memory：内存的动态峰值。 ● dynamic_used_shrctx：已使用的动态共享内存上下文。 ● dynamic_peak_shrctx：共享内存上下文的动态峰值。 ● max_shared_memory：最大共享内存。 ● shared_used_memory：已使用的共享内存。 ● high_backend_thread_memory：高优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● high_backend_thread_max_memory：高优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_memory：中优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_max_memory：中优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_memory ： 低优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_max_memory ： 低优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● max_sctpcomm_memory：TCP代理通信所允许使用的最大内存。 ● sctpcomm_used_memory：TCP代理通信已使用的内存大小。 ● sctpcomm_peak_memory：TCP代理通信的内存峰值。 ● other_used_memory：其他已使用的内存大小。 ● gpu_max_dynamic_memory：GPU最大动态内存。 ● gpu_dynamic_used_memory：GPU已使用的动态内存。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
		• gpu_dynamic_peak_memory：GPU内存的动态峰值。 • pooler_conn_memory：连接池申请内存计数。 • pooler_freeconn_memory：连接池空闲连接的内存计数。 • storage_compress_memory：存储模块压缩使用的内存大小。 • udf_reserved_memory：UDF预留的内存大小。
memorymbytes	integer	内存使用的大小，单位为MB。

8.2.3.2 DBE_PERF.GLOBAL_MEMORY_NODE_DETAIL

显示当前集群中所有正常节点下的内存使用情况，如**表8-19**所示。

表 8-19 DBE_PERF.GLOBAL_MEMORY_NODE_DETAIL 字段

名称	类型	描述
nodename	text	节点名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
memorytype	text	内存的名称。 ● max_process_memory：数据库节点可用内存的最大值。 ● process_used_memory：进程所使用的内存大小。 ● max_dynamic_memory：最大动态内存。 ● dynamic_used_memory：已使用的动态内存。 ● dynamic_peak_memory：内存的动态峰值。 ● dynamic_used_shrctx：已使用的动态共享内存上下文。 ● dynamic_peak_shrctx：共享内存上下文的动态峰值。 ● max_shared_memory：最大共享内存。 ● shared_used_memory：已使用的共享内存。 ● high_backend_thread_memory：高优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● high_backend_thread_max_memory：高优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_memory：中优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● medium_backend_thread_max_memory：中优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_memory ： 低优先级线程使用内存，enable_threads_memory_reserve=on时生效。 ● low_backend_thread_max_memory ： 低优先级线程预留内存，enable_threads_memory_reserve=on时生效。 ● max_sctpcomm_memory：TCP代理通信所允许使用的最大内存。 ● sctpcomm_used_memory：TCP代理通信已使用的内存大小。 ● sctpcomm_peak_memory：TCP代理通信的内存峰值。 ● other_used_memory：其他已使用的内存大小。 ● gpu_max_dynamic_memory：GPU最大动态内存。 ● gpu_dynamic_used_memory：GPU已使用的动态内存。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
		• gpu_dynamic_peak_memory：GPU内存的动态峰值。 • pooler_conn_memory：连接池申请内存计数。 • pooler_freeconn_memory：连接池空闲连接的内存计数。 • storage_compress_memory：存储模块压缩使用的内存大小。 • udf_reserved_memory：UDF预留的内存大小。
memorymbytes	integer	内存使用的大小，单位为MB。

8.2.3.3 DBE_PERF.MEMORY_NODE_NG_DETAIL

nodegroup内存使用情况，如表8-20所示。

表 8-20 DBE_PERF.MEMORY_NODE_NG_DETAIL 字段

名称	类型	描述
ngname	text	node group名称。
memorytype	text	内存的名称： • ng_total_memory：node group中设置的总内存。 • ng_used_memory：已经用的内存。 • ng_estimate_memory：优化器评估已经用的内存。 • ng_foreignrp_memsize：外部资源池设置的内存大小。 • ng_foreignrp_usedsize：外部资源池当前已用内存。 • ng_foreignrp_peaksize：外部资源池已使用的内存峰值。 • ng_foreignrp_mempct：外部资源池属性中设置的占用系统总内存的百分比。 • ng_foreignrp_estmsize：外部资源池执行作业优化器评估的内存使用。
memorymbytes	integer	内存使用的大小（单位：MB）。

版权所有 © 华为云计算技术有限公司

8.2.3.4 DBE_PERF.SHARED_MEMORY_DETAIL

查询当前节点所有已产生的共享内存上下文的使用信息，如表8-21所示。

表 8-21 DBE_PERF.SHARED_MEMORY_DETAIL 字段

名称	类型	描述
contextname	text	内存上下文的名称。
level	smallint	内存上下文的级别。
parent	text	上级内存上下文。
totalsize	bigint	共享内存总大小（单位：字节）。
freesize	bigint	共享内存剩余大小（单位：字节）。
usedsize	bigint	共享内存使用大小（单位：字节）。

8.2.3.5 DBE_PERF.GLOBAL_SHARED_MEMORY_DETAIL

查询整个集群中所有正常节点下的共享内存上下文的使用信息，如表8-22所示。

表 8-22 DBE_PERF.GLOBAL_SHARED_MEMORY_DETAIL 字段

名称	类型	描述
node_name	name	节点名称。
contextname	text	内存上下文的名称。
level	smallint	内存上下文的级别。
parent	text	上级内存上下文。
totalsize	bigint	共享内存总大小（单位：字节）。
freesize	bigint	共享内存剩余大小（单位：字节）。
usedsize	bigint	共享内存使用大小（单位：字节）。

8.2.3.6 DBE_PERF.TRACK_MEMORY_CONTEXT_DETAIL

查询DBE_PERF.TRACK_MEMORY_CONTEXT设置的内存上下文上的内存申请详细信息，如表8-23所示。只有初始用户或者具有monadmin权限的用户可以查询该视图。

版权所有 © 华为云计算技术有限公司

表 8-23 DBE_PERF.TRACK_MEMORY_CONTEXT_DETAIL 字段

名称	类型	描述
context_name	text	内存上下文的名称。
file	text	内存申请位置所属的文件。
line	integer	内存申请位置的行号。
size	bigint	内存申请的总大小（单位：字节）。

8.2.4 File

8.2.4.1 DBE_PERF.FILE_IOSTAT

通过对数据文件I/O的统计，反映数据的I/O性能，用以发现I/O操作异常等性能问题，如表8-24所示。

表 8-24 DBE_PERF.FILE_IOSTAT 字段

名称	类型	描述
filenum	oid	文件标识。
dbid	oid	数据库标识。
spcid	oid	表空间标识。
phyrds	bigint	读物理文件的数目。
phywrts	bigint	写物理文件的数目。
phyblkrd	bigint	读物理文件块的数目。
phyblkwrt	bigint	写物理文件块的数目。
readtim	bigint	读文件的总时长（单位：微秒）。
writetim	bigint	写文件的总时长（单位：微秒）。
avgiotim	bigint	读写文件的平均时长（单位：微秒）。
lstiotim	bigint	最后一次读文件时长（单位：微秒）。
miniotim	bigint	读写文件的最小时长（单位：微秒）。
maxiowtm	bigint	读写文件的最大时长（单位：微秒）。

8.2.4.2 DBE_PERF.SUMMARY_FILE_IOSTAT

通过集群内各节点数据文件I/O统计的汇总结果，反映数据的I/O性能，用以发现I/O操作异常等性能问题，如表8-25所示。

版权所有 © 华为云计算技术有限公司

其中phyrds、phywrts、phyblkrd、phyblkwrt、readtim、writetim字段按照各节点的数据累加求和，avgiotim为各节点的平均值（总时长/总次数），lstiotim、maxiowtm取各节点的最大值，miniotim取各节点的最小值。

表 8-25 DBE_PERF.SUMMARY_FILE_IOSTAT 字段

名称	类型	描述
filenum	oid	文件标识。
dbid	oid	数据库标识。
spcid	oid	表空间标识。
phyrds	numeric	读物理文件的数目。
phywrts	numeric	写物理文件的数目。
phyblkrd	numeric	读物理文件块的数目。
phyblkwrt	numeric	写物理文件块的数目。
readtim	numeric	读文件的总时长（单位：微秒）。
writetim	numeric	写文件的总时长（单位：微秒）。
avgiotim	bigint	读写文件的平均时长（单位：微秒）。
lstiotim	bigint	最后一次读文件时长（单位：微秒）。
miniotim	bigint	读写文件的最小时长（单位：微秒）。
maxiowtm	bigint	读写文件的最大时长（单位：微秒）。

8.2.4.3 DBE_PERF.GLOBAL_FILE_IOSTAT

显示集群上所有节点的数据文件I/O统计信息，如表8-26所示。

表 8-26 DBE_PERF.GLOBAL_FILE_IOSTAT 字段

名称	类型	描述
node_name	name	节点名称
filenum	oid	文件标识。
dbid	oid	数据库标识。
spcid	oid	表空间标识。
phyrds	bigint	读物理文件的数目。
phywrts	bigint	写物理文件的数目。
phyblkrd	bigint	读物理文件块的数目。
phyblkwrt	bigint	写物理文件块的数目。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
readtim	bigint	读文件的总时长（单位：微秒）。
writetim	bigint	写文件的总时长（单位：微秒）。
avgiotim	bigint	读写文件的平均时长（单位：微秒）。
lstiotim	bigint	最后一次读文件时长（单位：微秒）。
miniotim	bigint	读写文件的最小时长（单位：微秒）。
maxiowtm	bigint	读写文件的最大时长（单位：微秒）。

8.2.4.4 DBE_PERF.FILE_REDO_IOSTAT

本节点Redo（WAL）相关的统计信息，如表8-27所示。

表 8-27 DBE_PERF.FILE_REDO_IOSTAT 字段

名称	类型	描述
phywrts	bigint	向wal buffer中写的次数。
phyblkwrt	bigint	向wal buffer中写的block的块数。
writetim	bigint	向Xlog文件中写操作的时间（单位：微秒）。
avgiotim	bigint	平均写Xlog的时间（writetim/phywrts，单位：微秒）。
lstiotim	bigint	最后一次写Xlog的时间（单位：微秒）。
miniotim	bigint	最小的写Xlog时间（单位：微秒）。
maxiowtm	bigint	最大的写Xlog时间（单位：微秒）。

8.2.4.5 DBE_PERF.SUMMARY_FILE_REDO_IOSTAT

集群内汇总所有节点的Redo（WAL）相关的统计信息，如表8-28所示。其中phywrts、phyblkwrt、writetim字段按照各节点的数据累加求和，avgiotim为各节点的平均值（汇总的writetim/汇总的phywrts），lstiotim、maxiowtm取各节点的最大值，miniotim取各节点的最小值。

表 8-28 DBE_PERF.SUMMARY_FILE_REDO_IOSTAT 字段

名称	类型	描述
phywrts	numeric	向wal buffer中写的次数。
phyblkwrt	numeric	向wal buffer中写的block的块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
writetim	numeric	向Xlog文件中写操作的时间（单位：微秒）。
avgiotim	bigint	平均写Xlog的时间（writetim/phywrts，单位：微秒）。
lstiotim	bigint	最后一次写Xlog的时间（单位：微秒）。
miniotim	bigint	最小的写Xlog时间（单位：微秒）。
maxiowtm	bigint	最大的写Xlog时间（单位：微秒）。

8.2.4.6 DBE_PERF.GLOBAL_FILE_REDO_IOSTAT

显示集群内各节点的Redo（WAL）相关统计信息，如表8-29所示。

表 8-29 DBE_PERF.GLOBAL_FILE_REDO_IOSTAT 字段

名称	类型	描述
node_name	name	节点名称。
phywrts	bigint	向wal buffer中写的次数。
phyblkwrt	bigint	向wal buffer中写的block的块数。
writetim	bigint	向Xlog文件中写操作的时间（单位：微秒）。
avgiotim	bigint	平均写Xlog的时间（writetim/phywrts，单位：微秒）。
lstiotim	bigint	最后一次写Xlog的时间（单位：微秒）。
miniotim	bigint	最小的写Xlog时间（单位：微秒）。
maxiowtm	bigint	最大的写Xlog时间（单位：微秒）。

8.2.4.7 DBE_PERF.LOCAL_REL_IOSTAT

获取当前节点中数据文件I/O状态的累计值，显示为所有数据文件I/O状态的总和，如表8-30所示。

表 8-30 DBE_PERF.LOCAL_REL_IOSTAT 字段

名称	类型	描述
phyrds	bigint	读物理文件的数目。
phywrts	bigint	写物理文件的数目。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
phyblkrd	bigint	读物理文件的块的数目。
phyblkwrt	bigint	写物理文件的块的数目。
dbid	oid	在当前版本中，该值始终为0。

8.2.4.8 DBE_PERF.SUMMARY_REL_IOSTAT

获取集群上所有节点的数据文件I/O统计信息的汇总求和结果，如表8-31所示。

表 8-31 DBE_PERF.SUMMARY_REL_IOSTAT 字段

名称	类型	描述
phyrds	numeric	读物理文件的数目。
phywrts	numeric	写物理文件的数目。
phyblkrd	numeric	读物理文件的块的数目。
phyblkwrt	numeric	写物理文件的块的数目。
dbid	oid	在当前版本中，该值始终为0。

8.2.4.9 DBE_PERF.GLOBAL_REL_IOSTAT

获取集群上所有节点的数据文件I/O统计信息，如表8-32所示。

表 8-32 DBE_PERF.GLOBAL_REL_IOSTAT 字段

名称	类型	描述
node_name	name	节点名称
phyrds	bigint	读物理文件的数目。
phywrts	bigint	写物理文件的数目。
phyblkrd	bigint	读物理文件块的数目。
phyblkwrt	bigint	写物理文件块的数目。
dbid	oid	在当前版本中，该值始终为0。

8.2.5 Object

版权所有 © 华为云计算技术有限公司

8.2.5.1 DBE_PERF.STAT_USER_TABLES

显示当前节点所有Schema中用户自定义普通表的状态信息，如表8-33所示。

表 8-33 DBE_PERF.STAT_USER_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	该表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	该表被autovacuum守护线程清理的次数。
analyze_count	bigint	该表被手动分析的次数。
autoanalyze_count	bigint	该表被autovacuum守护线程分析的次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.5.2 DBE_PERF.SUMMARY_STAT_USER_TABLES

显示集群内各节点所有Schema中用户自定义普通表的状态信息的汇总求和结果（在CN节点使用。对每个DN节点中用户自定义普通表的状态信息进行汇总求和，其中timestamp类型字段不进行求和，仅取所有节点该字段的最新值），如表8-34所示。

表 8-34 DBE_PERF.SUMMARY_STAT_USER_TABLES

名称	类型	描述
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	numeric	该表发起的顺序扫描数。
seq_tup_read	numeric	顺序扫描抓取的活跃行数。
idx_scan	numeric	该表发起的索引扫描数。
idx_tup_fetch	numeric	索引扫描抓取的活跃行数。
n_tup_ins	numeric	插入行数。
n_tup_upd	numeric	更新行数。
n_tup_del	numeric	删除行数。
n_tup_hot_upd	numeric	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	numeric	估计活跃行数。
n_dead_tup	numeric	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	numeric	该表被手动清理的次数（不计算VACUUM FULL）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
autovacuum_count	numeric	该表被autovacuum守护线程清理的次数。
analyze_count	numeric	该表被手动分析的次数。
autoanalyze_count	numeric	该表被autovacuum守护线程分析的次数。

8.2.5.3 DBE_PERF.GLOBAL_STAT_USER_TABLES

显示集群内各节点所有Schema中用户自定义普通表的状态信息（在CN节点使用。显示DN节点用户自定义普通表的状态信息，不进行汇总），如表8-35所示。

表 8-35 DBE_PERF.GLOBAL_STAT_USER_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	该表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	该表被autovacuum守护线程清理的次数。
analyze_count	bigint	该表被手动分析的次数。
autoanalyze_count	bigint	该表被autovacuum守护线程分析的次数。

8.2.5.4 DBE_PERF.STAT_USER_INDEXES

显示当前节点内用户自定义普通表的索引状态信息，如表8-36所示。

表 8-36 DBE_PERF.STAT_USER_INDEXES 字段

名称	类型	描述
relid	oid	该索引的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	该索引上执行的索引扫描次数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

版权所有 © 华为云计算技术有限公司

8.2.5.5 DBE_PERF.SUMMARY_STAT_USER_INDEXES

显示集群内各节点所有Schema中用户自定义普通表的索引状态信息的汇总求和结果（在CN节点使用，对每个节点的用户自定义普通表的索引状态信息汇总求和），如表8-37所示。

表 8-37 DBE_PERF.SUMMARY_STAT_USER_INDEXES 字段

名称	类型	描述
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	numeric	该索引上执行的索引扫描次数。
idx_tup_read	numeric	该索引上扫描返回的索引项数。
idx_tup_fetch	numeric	使用该索引的简单索引扫描在原表中抓取的活跃行数。

8.2.5.6 DBE_PERF.GLOBAL_STAT_USER_INDEXES

显示集群内各节点所有Schema中用户自定义普通表的索引状态信息（包含CN与DN节点的信息，在CN节点使用，不汇总），如表8-38所示。

表 8-38 DBE_PERF.GLOBAL_STAT_USER_INDEXES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	该索引上执行的索引扫描次数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。

版权所有 © 华为云计算技术有限公司

8.2.5.7 DBE_PERF.STAT_SYS_TABLES

显示当前节点内pg_catalog、information_schema以及pg_toast模式下所有系统表的状态信息，如表8-39所示。

表 8-39 DBE_PERF.STAT_SYS_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	这个表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	该表被autovacuum守护线程清理的次数。
analyze_count	bigint	该表被手动分析的次数。
autoanalyze_count	bigint	该表被autovacuum守护线程分析的次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.5.8 DBE_PERF.SUMMARY_STAT_SYS_TABLES

显示集群内各节点pg_catalog、information_schema以及pg_toast模式下所有系统表的状态信息的汇总求和结果（在CN节点使用，统计信息包含CN节点与DN节点的数据。对每个节点下系统表的状态信息汇总求和，其中timestamp类型字段不进行求和，仅取所有节点该字段的最新值），如表8-40所示。

表 8-40 DBE_PERF.SUMMARY_STAT_SYS_TABLES 字段

名称	类型	描述
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	numeric	该表发起的顺序扫描数。
seq_tup_read	numeric	顺序扫描抓取的活跃行数。
idx_scan	numeric	该表发起的索引扫描数。
idx_tup_fetch	numeric	索引扫描抓取的活跃行数。
n_tup_ins	numeric	插入行数。
n_tup_upd	numeric	更新行数。
n_tup_del	numeric	删除行数。
n_tup_hot_upd	numeric	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	numeric	估计活跃行数。
n_dead_tup	numeric	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	numeric	该表被手动清理的次数（不计算VACUUM FULL）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
autovacuum_count	numeric	该表被autovacuum守护线程清理的次数。
analyze_count	numeric	该表被手动分析的次数。
autoanalyze_count	numeric	该表被autovacuum守护线程分析的次数。

8.2.5.9 DBE_PERF.GLOBAL_STAT_SYS_TABLES

显示集群内各个节点pg_catalog、information_schema以及pg_toast模式下所有系统表的状态信息（包含CN与DN节点的状态信息，在CN节点使用，不汇总），如表8-41所示。

表 8-41 DBE_PERF.GLOBAL_STAT_SYS_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析的时间。
vacuum_count	bigint	该表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	该表被autovacuum守护线程清理的次数。
analyze_count	bigint	该表被手动分析的次数。
autoanalyze_count	bigint	该表被autovacuum守护线程分析的次数。

8.2.5.10 DBE_PERF.STAT_SYS_INDEXES

显示当前节点pg_catalog、information_schema以及pg_toast模式中所有系统表的索引状态信息，如表8-42所示。

表 8-42 DBE_PERF.STAT_SYS_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	该索引上执行的索引扫描次数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

版权所有 © 华为云计算技术有限公司

8.2.5.11 DBE_PERF.SUMMARY_STAT_SYS_INDEXES

显示集群内各节点pg_catalog、information_schema以及pg_toast模式中所有系统表的索引状态信息的汇总求和结果（在CN节点使用，对每个节点内系统表的索引状态信息汇总求和），如表8-43所示。

表 8-43 DBE_PERF.SUMMARY_STAT_SYS_INDEXES 字段

名称	类型	描述
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	numeric	该索引上执行的索引扫描次数。
idx_tup_read	numeric	该索引上扫描返回的索引项数。
idx_tup_fetch	numeric	使用该索引的简单索引扫描在原表中抓取的活跃行数。

8.2.5.12 DBE_PERF.GLOBAL_STAT_SYS_INDEXES

显示集群内各节点pg_catalog、information_schema以及pg_toast模式中所有系统表的索引状态信息（包含CN与DN节点的索引状态信息，在CN节点使用，不汇总），如表8-44所示。

表 8-44 DBE_PERF.GLOBAL_STAT_SYS_INDEXES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	该索引上执行的索引扫描次数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。

版权所有 © 华为云计算技术有限公司

8.2.5.13 DBE_PERF.STAT_ALL_TABLES

显示数据库当前节点内每个表（包括TOAST表）的状态信息，如表8-45所示。

表 8-45 DBE_PERF.STAT_ALL_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）的时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析时间。
vacuum_count	bigint	该表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	该表被autovacuum守护线程清理的次数。
analyze_count	bigint	该表被手动分析的次数。
autoanalyze_count	bigint	该表被autovacuum守护线程分析的次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.5.14 DBE_PERF.SUMMARY_STAT_ALL_TABLES

显示集群内各节点中每个表（包括TOAST表）的状态信息的汇总求和结果（在CN节点使用。对每个节点中表的状态信息汇总求和，其中timestamp类型字段不进行求和，仅取所有节点中该字段的最新值），如表8-46所示。

表 8-46 DBE_PERF.SUMMARY_STAT_ALL_TABLES 字段

名称	类型	描述
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	numeric	该表发起的顺序扫描数。
seq_tup_read	numeric	顺序扫描抓取的活跃行数。
idx_scan	numeric	该表发起的索引扫描数。
idx_tup_fetch	numeric	索引扫描抓取的活跃行数。
n_tup_ins	numeric	插入行数。
n_tup_upd	numeric	更新行数。
n_tup_del	numeric	删除行数。
n_tup_hot_upd	numeric	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	numeric	估计活跃行数。
n_dead_tup	numeric	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）的时间。
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析时间。
vacuum_count	numeric	该表被手动清理的次数（不计算VACUUM FULL）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
autovacuum_count	numeric	该表被autovacuum清理的次数。
analyze_count	numeric	该表被手动分析的次数。
autoanalyze_count	numeric	该表被autovacuum守护线程分析的次数。
last_updated	timestamp with time zone	视图中该表监控数据最后一次更新的时间。

8.2.5.15 DBE_PERF.GLOBAL_STAT_ALL_TABLES

显示集群内各节点中每个表（包括TOAST表）的状态信息（在CN节点使用，每个表在每个节点下的状态信息不汇总），如表8-47所示。

表 8-47 DBE_PERF.GLOBAL_STAT_ALL_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。
n_live_tup	bigint	估计活跃行数。
n_dead_tup	bigint	估计不活跃行数。在Ustore表中仅代表不活跃行指针数量。
last_vacuum	timestamp with time zone	最后一次手动清理该表的（不计算VACUUM FULL）的时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_autovacuum	timestamp with time zone	上次被autovacuum守护线程清理的时间。
last_analyze	timestamp with time zone	上次手动分析该表的时间。
last_autoanalyze	timestamp with time zone	上次被autovacuum守护线程分析时间。
vacuum_count	bigint	该表被手动清理的次数（不计算VACUUM FULL）。
autovacuum_count	bigint	该表被autovacuum守护线程清理的次数。
analyze_count	bigint	该表被手动分析的次数。
autoanalyze_count	bigint	该表被autovacuum守护线程分析的次数。
last_updated	timestamp with time zone	视图中该表监控数据最后一次更新的时间。

8.2.5.16 DBE_PERF.STAT_ALL_INDEXES

显示数据库当前节点中的每个索引的访问信息，如表8-48所示。

表 8-48 DBE_PERF.STAT_ALL_INDEXES 字段

名称	类型	描述
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	该索引上执行的索引扫描次数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.5.17 DBE_PERF.SUMMARY_STAT_ALL_INDEXES

显示集群内数据库各节点的每个索引的访问统计信息的汇总求和结果（在CN节点使用。对每个节点索引的访问统计信息汇总求和，其中timestamp类型字段不求和，仅取所有节点中该字段最新值），如表8-49所示。

表 8-49 DBE_PERF.SUMMARY_STAT_ALL_INDEXES 字段

名称	类型	描述
schemaname	name	索引所在的Schema名。
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	numeric	该索引上执行的索引扫描次数。
idx_tup_read	numeric	该索引上扫描返回的索引项数。
idx_tup_fetch	numeric	使用该索引的简单索引扫描在原表中抓取的活跃行数。
last_updated	timestamp with time zone	视图中该索引监控数据最后一次更新的时间。

8.2.5.18 DBE_PERF.GLOBAL_STAT_ALL_INDEXES

显示集群内数据库各节点中每个索引的访问信息（在CN节点使用，每个索引在每个节点下的状态信息不汇总），如表8-50所示。

表 8-50 DBE_PERF.GLOBAL_STAT_ALL_INDEXES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	该索引所在的表的OID。
indexrelid	oid	索引的OID。
schemaname	name	索引所在的Schema名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
relname	name	索引的表名。
indexrelname	name	索引名。
idx_scan	bigint	该索引上执行的索引扫描次数。
idx_tup_read	bigint	该索引上扫描返回的索引项数。
idx_tup_fetch	bigint	使用该索引的简单索引扫描在原表中抓取的活跃行数。
last_updated	timestamp with time zone	视图中该索引监控数据最后一次更新的时间。

8.2.5.19 DBE_PERF.STAT_DATABASE

显示数据库当前节点的统计信息，如表8-51所示。

表 8-51 DBE_PERF.STAT_DATABASE 字段

名称	类型	描述
datid	oid	数据库的OID。
datname	name	该数据库的名称。
numbackends	integer	当前连接到该数据库的后端数。
xact_commit	bigint	该数据库中已经提交的事务数。
xact_rollback	bigint	该数据库中已经回滚的事务数。
blks_read	bigint	在该数据库中读取的磁盘块的数量。
blks_hit	bigint	高速缓存中已经命中的磁盘块的次数，这种情况下不需要从磁盘读取（高速缓存只包括数据库内核缓冲区高速缓存，不包括操作系统的文件系统缓存）。
tup_returned	bigint	该数据库中顺序扫描获取的活跃行数和索引扫描返回的索引行数。
tup_fetched	bigint	当前数据库通过索引返回的行数。
tup_inserted	bigint	插入的行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
tup_updated	bigint	更新的行数。
tup_deleted	bigint	删除的行数。
conflicts	bigint	由于与数据库回放发生冲突而取消的查询数量（冲突仅在备机上发生）。请参见DBE_PERF.STAT_DATABASE_CONFLICTS获取更多信息。
temp_files	bigint	该数据库中查询语句创建的临时文件数量。统计所有临时文件，不受GUC参数log_temp_files设置值影响。
temp_bytes	bigint	该数据库中查询语句写入临时文件的数据总量。统计所有临时文件，不受GUC参数log_temp_files设置值影响。
deadlocks	bigint	在该数据库中检索的死锁数。
blk_read_time	double precision	通过数据库后端读取数据文件块花费的时间，以毫秒计算。
blk_write_time	double precision	通过数据库后端写入数据文件块花费的时间，以毫秒计算。
stats_reset	timestamp with time zone	重置当前状态统计的时间。

8.2.5.20 DBE_PERF.SUMMARY_STAT_DATABASE

显示集群内各节点的状态信息的汇总求和结果（在CN节点使用。按数据库名对数据库每个节点的状态信息汇总求和，其中timestamp类型字段不进行求和，仅取所有节点该字段的最新值），如表8-52所示。

表 8-52 DBE_PERF.SUMMARY_STAT_DATABASE 字段

名称	类型	描述
datname	name	数据库的名称。
numbackends	bigint	当前连接到该数据库的后端数。
xact_commit	numeric	该数据库中已经提交的事务数。
xact_rollback	numeric	该数据库中已经回滚的事务数。
blks_read	numeric	在该数据库中读取的磁盘块的数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
blks_hit	numeric	高速缓存中已经命中的磁盘块的次数，这种情况下不需要从磁盘读取（高速缓存只包括数据库内核缓冲区高速缓存，不包括操作系统的文件系统缓存）。
tup_returned	numeric	该数据库中顺序扫描获取的活跃行数和索引扫描返回的索引行数。
tup_fetched	numeric	当前数据库通过索引返回的行数。
tup_inserted	bigint	插入的行数。
tup_updated	bigint	更新的行数。
tup_deleted	bigint	删除的行数。
conflicts	bigint	由于与数据库回放发生冲突而取消的查询数量（冲突仅在备机上发生）。请参见DBE_PERF.STAT_DATABASE_CONFLICTS获取更多信息。
temp_files	numeric	该数据库中查询语句创建的临时文件数量。统计所有临时文件，不受GUC参数log_temp_files设置值影响。
temp_bytes	numeric	该数据库中查询语句写入临时文件的数据总量。统计所有临时文件，不受GUC参数log_temp_files设置值影响。
deadlocks	bigint	在该数据库中检索的死锁数。
blk_read_time	double precision	通过数据库后端读取数据文件块花费的时间，以毫秒计算。
blk_write_time	double precision	通过数据库后端写入数据文件块花费的时间，以毫秒计算。
stats_reset	timestamp with time zone	重置当前状态统计的时间。

8.2.5.21 DBE_PERF.GLOBAL_STAT_DATABASE

显示集群内各节点的统计信息（在CN节点使用，每个数据库的状态统计信息不汇总），如表8-53所示。

表 8-53 DBE_PERF.GLOBAL_STAT_DATABASE 字段

名称	类型	描述
node_name	name	节点名称。
datid	oid	数据库的OID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
datname	name	该数据库的名称。
numbackends	integer	当前连接到该数据库的后端数。
xact_commit	bigint	该数据库中已经提交的事务数。
xact_rollback	bigint	该数据库中已经回滚的事务数。
blks_read	bigint	在这个数据库中读取的磁盘块的数量。
blks_hit	bigint	高速缓存中已经命中的磁盘块的次数，这种情况下不需要从磁盘读取（高速缓存只包括数据库内核缓冲区高速缓存，不包括操作系统的文件系统缓存）。
tup_returned	bigint	该数据库中顺序扫描获取的活跃行数和索引扫描返回的索引行数。
tup_fetched	bigint	当前数据库通过索引返回的行数。
tup_inserted	bigint	插入的行数。
tup_updated	bigint	更新的行数。
tup_deleted	bigint	删除的行数。
conflicts	bigint	由于与数据库回放发生冲突而取消的查询数量（冲突仅在备机上发生）。请参见 **DBE_PERF.STAT_DATABASE_CONFLICTS** 获取更多信息。
temp_files	bigint	该数据库中查询语句创建的临时文件数量。统计所有临时文件，不受GUC参数log_temp_files设置值影响。
temp_bytes	bigint	该数据库中查询语句写入临时文件的数据总量。统计所有临时文件，不受GUC参数log_temp_files设置值影响。
deadlocks	bigint	在该数据库中检索的死锁数。
blk_read_time	double precision	通过数据库后端读取数据文件块花费的时间，以毫秒计算。
blk_write_time	double precision	通过数据库后端写入数据文件块花费的时间，以毫秒计算。
stats_reset	timestamp with time zone	重置当前状态统计的时间。

版权所有 © 华为云计算技术有限公司

8.2.5.22 DBE_PERF.STAT_DATABASE_CONFLICTS

显示数据库当前节点冲突状态的统计信息，如表8-54所示。

表 8-54 DBE_PERF.STAT_DATABASE_CONFLICTS 字段

名称	类型	描述
datid	oid	数据库标识。
datname	name	数据库名称。
confl_tablespace	bigint	冲突的表空间的数目。
confl_lock	bigint	冲突的锁数目。
confl_snapshot	bigint	冲突的快照数目。
confl_bufferpin	bigint	冲突的缓冲区数目。
confl_deadlock	bigint	冲突的死锁数目。

8.2.5.23 DBE_PERF.SUMMARY_STAT_DATABASE_CONFLICTS

显示集群内各节点冲突状态的统计信息的汇总求和结果（在CN节点使用，按数据库名对每个节点的冲突状态的统计信息汇总求和），如表8-55所示。

表 8-55 DBE_PERF.SUMMARY_STAT_DATABASE_CONFLICTS 字段

名称	类型	描述
datname	name	数据库名称。
confl_tablespace	bigint	冲突的表空间的数目。
confl_lock	bigint	冲突的锁数目。
confl_snapshot	bigint	冲突的快照数目。
confl_bufferpin	bigint	冲突的缓冲区数目。
confl_deadlock	bigint	冲突的死锁数目。

8.2.5.24 DBE_PERF.GLOBAL_STAT_DATABASE_CONFLICTS

显示集群内各节点冲突状态的统计信息（在CN节点使用，每个数据库状态信息不汇总），如表8-56所示。

版权所有 © 华为云计算技术有限公司

表 8-56 DBE_PERF.GLOBAL_STAT_DATABASE_CONFLICTS 字段

名称	类型	描述
node_name	name	节点名称
datid	oid	数据库标识。
datname	name	数据库名称。
confl_tablespace	bigint	冲突的表空间的数目。
confl_lock	bigint	冲突的锁数目。
confl_snapshot	bigint	冲突的快照数目。
confl_bufferpin	bigint	冲突的缓冲区数目。
confl_deadlock	bigint	冲突的死锁数目。

8.2.5.25 DBE_PERF.STAT_XACT_ALL_TABLES

显示当前节点Schema中所有普通表和toast表的事务状态信息，如表8-57所示。

表 8-57 DBE_PERF.STAT_XACT_ALL_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。

版权所有 © 华为云计算技术有限公司

8.2.5.26 DBE_PERF.SUMMARY_STAT_XACT_ALL_TABLES

显示集群内各节点Schema中所有普通表和toast表的事务状态信息的汇总求和结果（在CN节点使用，按表名对每个节点中表的事务状态统计信息汇总），如表8-58所示。

表 8-58 DBE_PERF.SUMMARY_STAT_XACT_ALL_TABLES 字段

名称	类型	描述
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	numeric	该表发起的顺序扫描数。
seq_tup_read	numeric	顺序扫描抓取的活跃行数。
idx_scan	numeric	该表发起的索引扫描数。
idx_tup_fetch	numeric	索引扫描抓取的活跃行数。
n_tup_ins	numeric	插入行数。
n_tup_upd	numeric	更新行数。
n_tup_del	numeric	删除行数。
n_tup_hot_upd	numeric	HOT更新行数（即没有更新索引列的行数）。

8.2.5.27 DBE_PERF.GLOBAL_STAT_XACT_ALL_TABLES

显示集群内各节点的Schema中所有普通表和toast表的事务状态信息（在CN节点使用，不同节点下相同表名的事务状态信息不进行汇总），如表8-59所示。

表 8-59 DBE_PERF.GLOBAL_STAT_XACT_ALL_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。

8.2.5.28 DBE_PERF.STAT_XACT_SYS_TABLES

显示当前节点Schema中系统表的事务状态信息，如表8-60所示。

表 8-60 DBE_PERF.STAT_XACT_SYS_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。

8.2.5.29 DBE_PERF.SUMMARY_STAT_XACT_SYS_TABLES

显示集群内各节点Schema中系统表的事务状态信息的汇总求和结果（在CN节点使用，按表名对每个节点中表的事务状态统计信息汇总求和），如表8-61所示。

版权所有 © 华为云计算技术有限公司

表 8-61 DBE_PERF.SUMMARY_STAT_XACT_SYS_TABLES 字段

名称	类型	描述
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	numeric	该表发起的顺序扫描数。
seq_tup_read	numeric	顺序扫描抓取的活跃行数。
idx_scan	numeric	该表发起的索引扫描数。
idx_tup_fetch	numeric	索引扫描抓取的活跃行数。
n_tup_ins	numeric	插入行数。
n_tup_upd	numeric	更新行数。
n_tup_del	numeric	删除行数。
n_tup_hot_upd	numeric	HOT更新行数（即没有更新索引列的行数）。

8.2.5.30 DBE_PERF.GLOBAL_STAT_XACT_SYS_TABLES

显示集群内各节点Schema中系统表的事务状态信息（在CN节点使用，不同节点下相同表名的事务状态信息不进行汇总），如表8-62所示。

表 8-62 DBE_PERF.GLOBAL_STAT_XACT_SYS_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。

8.2.5.31 DBE_PERF.STAT_XACT_USER_TABLES

显示当前节点Schema中用户表的事务状态信息，如表8-63所示。

表 8-63 DBE_PERF.STAT_XACT_USER_TABLES 字段

名称	类型	描述
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。

8.2.5.32 DBE_PERF.SUMMARY_STAT_XACT_USER_TABLES

显示集群内各节点Schema中用户表的事务状态信息的汇总求和结果（在CN节点使用，按表名对每个节点中表的事务状态统计信息汇总求和），如表8-64所示。

表 8-64 DBE_PERF.SUMMARY_STAT_XACT_USER_TABLES 字段

名称	类型	描述
schemaname	name	该表所在的Schema名。
relname	name	表名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
seq_scan	numeric	该表发起的顺序扫描数。
seq_tup_read	numeric	顺序扫描抓取的活跃行数。
idx_scan	numeric	该表发起的索引扫描数。
idx_tup_fetch	numeric	索引扫描抓取的活跃行数。
n_tup_ins	numeric	插入行数。
n_tup_upd	numeric	更新行数。
n_tup_del	numeric	删除行数。
n_tup_hot_upd	numeric	HOT更新行数（即没有更新索引列的行数）。

8.2.5.33 DBE_PERF.GLOBAL_STAT_XACT_USER_TABLES

显示集群内各节点Schema中用户表的事务状态信息（在CN节点使用，不同节点下相同表名的事务状态信息不进行汇总），如表8-65所示。

表 8-65 DBE_PERF.GLOBAL_STAT_XACT_USER_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表的OID。
schemaname	name	该表所在的Schema名。
relname	name	表名。
seq_scan	bigint	该表发起的顺序扫描数。
seq_tup_read	bigint	顺序扫描抓取的活跃行数。
idx_scan	bigint	该表发起的索引扫描数。
idx_tup_fetch	bigint	索引扫描抓取的活跃行数。
n_tup_ins	bigint	插入行数。
n_tup_upd	bigint	更新行数。
n_tup_del	bigint	删除行数。
n_tup_hot_upd	bigint	HOT更新行数（即没有更新索引列的行数）。

版权所有 © 华为云计算技术有限公司

8.2.5.34 DBE_PERF.STAT_XACT_USER_FUNCTIONS

显示当前节点本事务内函数执行的统计信息，如表8-66所示。

表 8-66 DBE_PERF.STAT_XACT_USER_FUNCTIONS 字段

名称	类型	描述
funcid	oid	函数标识。
schemaname	name	函数所在Schema名。
funcname	name	函数名称。
calls	bigint	函数被调用的次数。
total_time	double precision	此函数及其调用的所有其他函数所花费的总时间。
self_time	double precision	在此函数本身中花费的时间（不包括它调用的其他函数）。

8.2.5.35 DBE_PERF.SUMMARY_STAT_XACT_USER_FUNCTIONS

显示集群内各节点，本事务内函数执行的统计信息的汇总求和结果（在CN节点使用），如表8-67所示。

表 8-67 DBE_PERF.SUMMARY_STAT_XACT_USER_FUNCTIONS 字段

名称	类型	描述
schemaname	name	函数所在的Schema名。
funcname	name	函数名称。
calls	numeric	函数被调用的次数。
total_time	double precision	此函数及其调用的所有其他函数所花费的总时间。
self_time	double precision	在此函数本身中花费的时间（不包括它调用的其他函数）。

8.2.5.36 DBE_PERF.GLOBAL_STAT_XACT_USER_FUNCTIONS

显示集群内各节点，本事务内函数执行的统计信息（在CN节点使用），如表8-68所示。

版权所有 © 华为云计算技术有限公司

表 8-68 DBE_PERF.GLOBAL_STAT_XACT_USER_FUNCTIONS 字段

名称	类型	描述
node_name	name	节点名称。
funcid	oid	函数标识。
schemaname	name	函数所在的Schema名。
funcname	name	函数名称。
calls	bigint	函数被调用的次数。
total_time	double precision	此函数及其调用的所有其他函数所花费的总时间。
self_time	double precision	在此函数本身中花费的时间（不包括它调用的其他函数）。

8.2.5.37 DBE_PERF.STAT_BAD_BLOCK

获得当前节点表、索引等文件的读取失败信息，如表8-69所示。

表 8-69 DBE_PERF.STAT_BAD_BLOCK 字段

名称	类型	描述
nodename	text	节点名称。
databaseid	integer	database的oid。
tablespaceid	integer	tablespace的oid。
relfilenode	integer	relation的file node。
bucketid	smallint	一致性hash bucket ID。
forknum	integer	fork编号。
error_count	integer	error的数量。
first_time	timestamp with time zone	页面损坏第一次出现的时间。
last_time	timestamp with time zone	页面损坏最后出现的时间。

8.2.5.38 DBE_PERF.SUMMARY_STAT_BAD_BLOCK

获得集群内各节点的表、索引等文件的读取失败信息的汇总求和结果（在CN节点使用。对不同DN的统计信息汇总求和，first_time取最早的时间，last_time取最新的时间），如表8-70所示。

版权所有 © 华为云计算技术有限公司

表 8-70 DBE_PERF.SUMMARY_STAT_BAD_BLOCK 字段

名称	类型	描述
databaseid	integer	database的oid。
tablespaceid	integer	tablespace的oid。
relfilenode	integer	relation的file node。
forknum	bigint	fork编号。
error_count	bigint	error的数量。
first_time	timestamp with time zone	页面损坏第一次出现的时间。
last_time	timestamp with time zone	页面损坏最后出现的时间。

8.2.5.39 DBE_PERF.GLOBAL_STAT_BAD_BLOCK

获得集群内各节点的表、索引等文件的读取失败信息（在CN节点使用。仅显示DN的读取失败信息，不进行汇总），如表8-71所示。

表 8-71 DBE_PERF.GLOBAL_STAT_BAD_BLOCK 字段

名称	类型	描述
node_name	text	节点名称。
databaseid	integer	database的oid。
tablespaceid	integer	tablespace的oid。
relfilenode	integer	relation的file node。
forknum	integer	fork编号。
error_count	integer	error的数量。
first_time	timestamp with time zone	页面损坏第一次出现的时间。
last_time	timestamp with time zone	页面损坏最后出现的时间。

8.2.5.40 DBE_PERF.STAT_USER_FUNCTIONS

显示当前节点的Schema中用户自定义函数（函数语言为非内部语言）的统计信息，如表8-72所示。

版权所有 © 华为云计算技术有限公司

表 8-72 DBE_PERF.STAT_USER_FUNCTIONS 字段

名称	类型	描述
funcid	oid	函数标识。
schemaname	name	Schema的名称。
funcname	name	用户自定义函数的名称。
calls	bigint	函数被调用的次数。
total_time	double precision	调用此函数花费的总时间，包含调用其它函数的时间（单位：毫秒）。
self_time	double precision	调用此函数本身花费的时间，不包含调用其它函数的时间（单位：毫秒）。

8.2.5.41 DBE_PERF.SUMMARY_STAT_USER_FUNCTIONS

显示集群内各节点的用户自定义函数的统计信息的汇总求和结果（在CN节点使用，按函数名对每个节点中用户自定义函数的状态信息汇总求和），如表8-73所示。

表 8-73 DBE_PERF.SUMMARY_STAT_USER_FUNCTIONS 字段

名称	类型	描述
schemaname	name	Schema的名称。
funcname	name	用户自定义函数的名称。
calls	numeric	该函数被调用的次数。
total_time	double precision	调用此函数花费的总时间，包含调用其它函数的时间（单位：毫秒）。
self_time	double precision	调用此函数本身花费的时间，不包含调用其它函数的时间（单位：毫秒）。

8.2.5.42 DBE_PERF.GLOBAL_STAT_USER_FUNCTIONS

显示集群内各节点的用户自定义函数的统计信息（在CN节点使用，不同节点同名函数状态信息不汇总），如表8-74所示。

表 8-74 DBE_PERF.GLOBAL_STAT_USER_FUNCTIONS 字段

名称	类型	描述
node_name	name	节点名称。
funcid	oid	函数的id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schemaname	name	此函数所在Schema的名称。
funcname	name	用户自定义函数的名称。
calls	bigint	该函数被调用的次数。
total_time	double precision	调用此函数花费的总时间，包含调用其它函数的时间（单位：毫秒）。
self_time	double precision	调用此函数本身花费的时间，不包含调用其它函数的时间（单位：毫秒）。

8.2.6 Workload

8.2.6.1 DBE_PERF.WORKLOAD_SQL_COUNT

显示当前节点workload上的SQL数量分布，如表8-75所示。

表 8-75 DBE_PERF.WORKLOAD_SQL_COUNT 字段

名称	类型	描述
workload	name	负载名称。
select_count	bigint	SELECT语句数量。
update_count	bigint	UPDATE语句数量。
insert_count	bigint	INSERT语句数量。
delete_count	bigint	DELETE语句数量。
ddl_count	bigint	DDL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dml_count	bigint	DML语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。
dcl_count	bigint	DCL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。
dbid	oid	统计信息所属的数据库ID。
user_dml_count	bigint	用户执行的DML语句的数量。user_dml_count的数量等于dml_count的数量减去bg_dml_count的数量。
bg_dml_count	bigint	数据库后台线程执行的DML语句的数量，包含全场景下后台线程（WORKER和THREADPOOL_WORKER以外的线程）执行的DML语句。 **说明** 当前存储过程中的命令不支持统计DML语句。
fqs_count	bigint	FQS（Fast Query Shipping）语句的数量。该统计结果包含用户执行的FQS语句和数据库后台线程执行的FQS语句。
lightproxy_count	bigint	CN轻量化语句的数量。该统计结果包含用户执行的CN轻量化语句和数据库后台线程执行的CN轻量化语句。
stream_count	bigint	Stream语句的数量。该统计结果包含用户执行的Stream语句和数据库后台线程执行的Stream语句。
pgxc_count	bigint	PGXC语句的数量。该统计结果包含用户执行的PGXC语句和数据库后台线程执行的Stream语句。

8.2.6.2 DBE_PERF.SUMMARY_WORKLOAD_SQL_COUNT

显示集群内各CN的workload上的SQL数量分布，如表8-76所示。

版权所有 © 华为云计算技术有限公司

表 8-76 DBE_PERF.SUMMARY_WORKLOAD_SQL_COUNT 字段

名称	类型	描述
node_name	name	节点名称。
workload	name	负载名称。
select_count	bigint	SELECT语句数量。
update_count	bigint	UPDATE语句数量。
insert_count	bigint	INSERT语句数量。
delete_count	bigint	DELETE语句数量。
ddl_count	bigint	DDL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。
dml_count	bigint	DML语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。
dcl_count	bigint	DCL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。
dbid	oid	统计信息所属的数据库ID。
user_dml_count	bigint	用户执行的DML语句的数量。user_dml_count的数量等于dml_count的数量减去bg_dml_count的数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bg_dml_count	bigint	数据库后台线程执行的DML语句的数量，包含全场景下后台线程（WORKER和THREADPOOL_WORKER以外的线程）执行的DML语句。 **说明** 当前存储过程中的命令不支持统计DML语句。
fqs_count	bigint	FQS（Fast Query Shipping）语句的数量。该统计结果包含用户执行的FQS语句和数据库后台线程执行的FQS语句。
lightproxy_count	bigint	CN轻量化语句的数量。该统计结果包含用户执行的CN轻量化语句和数据库后台线程执行的CN轻量化语句。
stream_count	bigint	Stream语句的数量。该统计结果包含用户执行的Stream语句和数据库后台线程执行的Stream语句。
pgxc_count	bigint	PGXC语句的数量。该统计结果包含用户执行的PGXC语句和数据库后台线程执行的PGXC语句。

8.2.6.3 DBE_PERF.WORKLOAD_TRANSACTION

当前节点上负载的事务信息，如表8-77所示。

表 8-77 DBE_PERF.WORKLOAD_TRANSACTION 字段

名称	类型	描述
workload	name	负载的名称。
commit_counter	bigint	用户事务commit数量。
rollback_counter	bigint	用户事务rollback数量。
resp_min	bigint	用户事务最小响应时间（单位：微秒）。
resp_max	bigint	用户事务最大响应时间（单位：微秒）。
resp_avg	bigint	用户事务平均响应时间（单位：微秒）。
resp_total	bigint	用户事务总响应时间（单位：微秒）。
bg_commit_counter	bigint	后台事务commit数量。
bg_rollback_counter	bigint	后台事务rollback数量。
bg_resp_min	bigint	后台事务最小响应时间（单位：微秒）。
bg_resp_max	bigint	后台事务最大响应时间（单位：微秒）。
bg_resp_avg	bigint	后台事务平均响应时间（单位：微秒）。
bg_resp_total	bigint	后台事务总响应时间（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dist_commit_counter	bigint	分布式事务commit数量。
dist_rollback_counter	bigint	分布式事务rollback数量。
dist_resp_min	bigint	分布式事务最小响应时间（单位：微秒）。
dist_resp_max	bigint	分布式事务最大响应时间（单位：微秒）。
dist_resp_avg	bigint	分布式事务平均响应时间（单位：微秒）。
dist_resp_total	bigint	分布式事务总响应时间（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

说明

只统计主事务信息，子事务信息不会被统计在内。

8.2.6.4 DBE_PERF.SUMMARY_WORKLOAD_TRANSACTION

显示集群内汇聚的负载事务信息，如表8-78所示。

表 8-78 DBE_PERF.SUMMARY_WORKLOAD_TRANSACTION 字段

名称	类型	描述
workload	name	负载的名称。
commit_counter	numeric	用户事务commit数量。
rollback_counter	numeric	用户事务rollback数量。
resp_min	bigint	用户事务最小响应时间（单位：微秒）。
resp_max	bigint	用户事务最大响应时间（单位：微秒）。
resp_avg	bigint	用户事务平均响应时间（单位：微秒）。
resp_total	numeric	用户事务总响应时间（单位：微秒）。
bg_commit_counter	numeric	后台事务commit数量。
bg_rollback_counter	numeric	后台事务rollback数量。
bg_resp_min	bigint	后台事务最小响应时间（单位：微秒）。
bg_resp_max	bigint	后台事务最大响应时间（单位：微秒）。
bg_resp_avg	bigint	后台事务平均响应时间（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bg_resp_total	numeric	后台事务总响应时间（单位：微秒）。
dist_commit_counter	bigint	分布式事务commit数量。
dist_rollback_counter	bigint	分布式事务rollback数量。
dist_resp_min	bigint	分布式事务最小响应时间（单位：微秒）。
dist_resp_max	bigint	分布式事务最大响应时间（单位：微秒）。
dist_resp_avg	bigint	分布式事务平均响应时间（单位：微秒）。
dist_resp_total	bigint	分布式事务总响应时间（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

说明

只统计主事务信息，子事务信息不会被统计在内。

8.2.6.5 DBE_PERF.GLOBAL_WORKLOAD_TRANSACTION

显示集群内各节点上的workload的负载信息，如表8-79所示。

表 8-79 DBE_PERF.GLOBAL_WORKLOAD_TRANSACTION 字段

名称	类型	描述
node_name	name	节点名称。
workload	name	负载的名称。
commit_counter	bigint	用户事务commit数量。
rollback_counter	bigint	用户事务rollback数量。
resp_min	bigint	用户事务最小响应时间（单位：微秒）。
resp_max	bigint	用户事务最大响应时间（单位：微秒）。
resp_avg	bigint	用户事务平均响应时间（单位：微秒）。
resp_total	bigint	用户事务总响应时间（单位：微秒）。
bg_commit_counter	bigint	后台事务commit数量。
bg_rollback_counter	bigint	后台事务rollback数量。
bg_resp_min	bigint	后台事务最小响应时间（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bg_resp_max	bigint	后台事务最大响应时间（单位：微秒）。
bg_resp_avg	bigint	后台事务平均响应时间（单位：微秒）。
bg_resp_total	bigint	后台事务总响应时间（单位：微秒）。
dist_commit_counter	bigint	分布式事务commit数量。
dist_rollback_counter	bigint	分布式事务rollback数量。
dist_resp_min	bigint	分布式事务最小响应时间（单位：微秒）。
dist_resp_max	bigint	分布式事务最大响应时间（单位：微秒）。
dist_resp_avg	bigint	分布式事务平均响应时间（单位：微秒）。
dist_resp_total	bigint	分布式事务总响应时间（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

说明

只统计主事务信息，子事务信息不会被统计在内。

8.2.6.6 DBE_PERF.WORKLOAD_SQL_ELAPSE_TIME

该视图用于统计workload（业务负载）上的SUID（查询/更新/插入/删除）信息，如表8-80所示。

表 8-80 DBE_PERF.WORKLOAD_SQL_ELAPSE_TIME 字段

名称	类型	描述
workload	name	workload（业务负载）名称。
total_select_elapse	bigint	SELECT语句总的时间花费（单位：微秒）。
max_select_elapse	bigint	SELECT语句最大的时间花费（单位：微秒）。
min_select_elapse	bigint	SELECT语句最小的时间花费（单位：微秒）。
avg_select_elapse	bigint	SELECT语句平均的时间花费（单位：微秒）。
total_update_elapse	bigint	UPDATE语句总的时间花费（单位：微秒）。
max_update_elapse	bigint	UPDATE语句最大的时间花费（单位：微秒）。
min_update_elapse	bigint	UPDATE语句最小的时间花费（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
avg_update_elapse	bigint	UPDATE语句平均的时间花费（单位：微秒）。
total_insert_elapse	bigint	INSERT语句总的时间花费（单位：微秒）。
max_insert_elapse	bigint	INSERT语句最大的时间花费（单位：微秒）。
min_insert_elapse	bigint	INSERT语句最小的时间花费（单位：微秒）。
avg_insert_elapse	bigint	INSERT语句平均的时间花费（单位：微秒）。
total_delete_elapse	bigint	DELETE语句总的时间花费（单位：微秒）。
max_delete_elapse	bigint	DELETE语句最大的时间花费（单位：微秒）。
min_delete_elapse	bigint	DELETE语句最小的时间花费（单位：微秒）。
avg_delete_elapse	bigint	DELETE语句平均的时间花费（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

8.2.6.7 DBE_PERF.SUMMARY_WORKLOAD_SQL_ELAPSE_TIME

该视图用于统计所有CN节点上workload（业务负载）的SUID（查询/更新/插入/删除）信息，如表8-81所示。

表 8-81 DBE_PERF.SUMMARY_WORKLOAD_SQL_ELAPSE_TIME 字段

名称	类型	描述
node_name	name	节点名称。
workload	name	workload（业务负载）名称。
total_select_elapse	bigint	SELECT语句总的时间花费（单位：微秒）。
max_select_elapse	bigint	SELECT语句最大的时间花费（单位：微秒）。
min_select_elapse	bigint	SELECT语句最小的时间花费（单位：微秒）。
avg_select_elapse	bigint	SELECT语句平均的时间花费（单位：微秒）。
total_update_elapse	bigint	UPDATE语句总的时间花费（单位：微秒）。
max_update_elapse	bigint	UPDATE语句最大的时间花费（单位：微秒）。
min_update_elapse	bigint	UPDATE语句最小的时间花费（单位：微秒）。
avg_update_elapse	bigint	UPDATE语句平均的时间花费（单位：微秒）。
total_insert_elapse	bigint	INSERT语句总的时间花费（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_insert_elapse	bigint	INSERT语句最大的时间花费（单位：微秒）。
min_insert_elapse	bigint	INSERT语句最小的时间花费（单位：微秒）。
avg_insert_elapse	bigint	INSERT语句平均的时间花费（单位：微秒）。
total_delete_elapse	bigint	DELETE语句总的时间花费（单位：微秒）。
max_delete_elapse	bigint	DELETE语句最大的时间花费（单位：微秒）。
min_delete_elapse	bigint	DELETE语句最小的时间花费（单位：微秒）。
avg_delete_elapse	bigint	DELETE语句平均的时间花费（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

8.2.6.8 DBE_PERF.USER_TRANSACTION

该视图用于统计用户执行的事务信息，如表8-82所示。monadmin用户能看到所有用户执行事务的信息。

表 8-82 DBE_PERF.USER_TRANSACTION 字段

名称	类型	描述
username	name	用户的名称。
commit_counter	bigint	用户事务commit数量。
rollback_counter	bigint	用户事务rollback数量。
resp_min	bigint	用户提交事务最小响应时间（单位：微秒）。
resp_max	bigint	用户提交事务最大响应时间（单位：微秒）。
resp_avg	bigint	用户提交及回滚事务平均响应时间（单位：微秒）。
resp_total	bigint	用户提交事务总响应时间（单位：微秒）。
bg_commit_counter	bigint	后台事务commit数量。
bg_rollback_counter	bigint	后台事务rollback数量。
bg_resp_min	bigint	后台事务最小响应时间（单位：微秒）。
bg_resp_max	bigint	后台事务最大响应时间（单位：微秒）。
bg_resp_avg	bigint	后台事务平均响应时间（单位：微秒）。
bg_resp_total	bigint	后台事务总响应时间（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dist_commit_counter	bigint	分布式事务commit数量。
dist_rollback_counter	bigint	分布式事务rollback数量。
dist_resp_min	bigint	分布式提交事务最小响应时间（单位：微秒）。
dist_resp_max	bigint	分布式提交事务最大响应时间（单位：微秒）。
dist_resp_avg	bigint	分布式提交及回滚事务平均响应时间（单位：微秒）。
dist_resp_total	bigint	分布式提交事务总响应时间（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

说明

只统计主事务信息，子事务信息不会被统计在内。

8.2.6.9 DBE_PERF.GLOBAL_USER_TRANSACTION

该视图用于统计全局用户执行的事务信息，如表8-83所示。

表 8-83 DBE_PERF.GLOBAL_USER_TRANSACTION 字段

名称	类型	描述
node_name	name	节点名称。
username	name	用户的名称。
commit_counter	bigint	用户事务commit数量。
rollback_counter	bigint	用户事务rollback数量。
resp_min	bigint	用户提交事务最小响应时间（单位：微秒）。
resp_max	bigint	用户提交事务最大响应时间（单位：微秒）。
resp_avg	bigint	用户提交及回滚事务平均响应时间（单位：微秒）。
resp_total	bigint	用户提交事务总响应时间（单位：微秒）。
bg_commit_counter	bigint	后台事务commit数量。
bg_rollback_counter	bigint	后台事务rollback数量。
bg_resp_min	bigint	后台事务最小响应时间（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bg_resp_max	bigint	后台事务最大响应时间（单位：微秒）。
bg_resp_avg	bigint	后台事务平均响应时间（单位：微秒）。
bg_resp_total	bigint	后台事务总响应时间（单位：微秒）。
dist_commit_counter	bigint	分布式事务commit数量。
dist_rollback_counter	bigint	分布式事务rollback数量。
dist_resp_min	bigint	分布式提交事务最小响应时间（单位：微秒）。
dist_resp_max	bigint	分布式提交事务最大响应时间（单位：微秒）。
dist_resp_avg	bigint	分布式提交及回滚事务平均响应时间（单位：微秒）。
dist_resp_total	bigint	分布式提交事务总响应时间（单位：微秒）。
dbid	oid	统计信息所属的数据库ID。

说明

只统计主事务信息，子事务信息不会被统计在内。

8.2.7 Session/Thread

8.2.7.1 DBE_PERF.SESSION_STAT

用于统计当前节点以会话线程或AutoVacuum线程为单位的会话状态信息，如表8-84所示。

表 8-84 DBE_PERF.SESSION_STAT 字段

名称	类型	描述
sessid	text	线程启动时间+线程标识。
statid	integer	统计编号。
statname	text	统计会话名称。
statunit	text	统计会话单位。
value	bigint	统计会话值。

版权所有 © 华为云计算技术有限公司

8.2.7.2 DBE_PERF.GLOBAL_SESSION_STAT

用于统计各节点上以会话线程或AutoVacuum线程为单位的会话状态信息，如表8-85所示。

表 8-85 DBE_PERF.GLOBAL_SESSION_STAT 字段

名称	类型	描述
node_name	name	节点名称。
sessid	text	线程启动时间+线程标识。
statid	integer	统计编号。
statname	text	统计会话名称。
statunit	text	统计会话单位。
value	bigint	统计会话值。

8.2.7.3 DBE_PERF.SESSION_TIME

用于统计当前节点会话线程的运行时间信息，及各执行阶段所消耗时间，如表8-86所示。

表 8-86 DBE_PERF.SESSION_TIME 字段

名称	类型	描述
sessid	text	线程启动时间+线程标识。
stat_id	integer	统计编号。
stat_name	text	统计会话类型名称。
value	bigint	统计会话值。

8.2.7.4 DBE_PERF.GLOBAL_SESSION_TIME

用于统计各节点会话线程的运行时间信息，及各执行阶段所消耗时间，如表8-87所示。

表 8-87 DBE_PERF.GLOBAL_SESSION_TIME 字段

名称	类型	描述
node_name	name	节点名称。
sessid	text	线程启动时间+线程标识。
stat_id	integer	统计编号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
stat_name	text	统计会话类型名称。
value	bigint	统计会话值。

8.2.7.5 DBE_PERF.SESSION_MEMORY

用于统计Session级别的内存使用情况，包含执行作业在当前节点上GaussDB线程和Stream线程分配的所有内存，单位为MB，如表8-88所示。

表 8-88 DBE_PERF.SESSION_MEMORY 字段

名称	类型	描述
sessid	text	线程启动时间+线程标识。
init_mem	integer	当前正在执行作业进入执行器前已分配的内存。
used_mem	integer	当前正在执行作业已分配的内存。
peak_mem	integer	当前正在执行作业已分配的内存峰值。

8.2.7.6 DBE_PERF.GLOBAL_SESSION_MEMORY

用于统计各节点的Session级别的内存使用情况，包含执行作业在数据节点上GaussDB线程和Stream线程分配的所有内存，单位为MB，如表8-89所示。

表 8-89 DBE_PERF.GLOBAL_SESSION_MEMORY 字段

名称	类型	描述
node_name	name	节点名称。
sessid	text	线程启动时间+线程标识。
init_mem	integer	当前正在执行作业进入执行器前已分配的内存。
used_mem	integer	当前正在执行作业已分配的内存。
peak_mem	integer	当前正在执行作业已分配的内存峰值。

8.2.7.7 DBE_PERF.SESSION_MEMORY_DETAIL

用于统计线程的内存使用情况，以MemoryContext粒度来统计当前节点的内存，如表8-90所示。

版权所有 © 华为云计算技术有限公司

表 8-90 DBE_PERF.SESSION_MEMORY_DETAIL 字段

名称	类型	描述
sessid	text	线程启动时间+线程标识。
sesstype	text	线程名称。
contextname	text	内存上下文名称。
level	smallint	内存上下文的重要级别。
parent	text	父级内存上下文名称。
totalsize	bigint	总申请内存大小（单位：字节）。
freesize	bigint	空闲内存大小（单位：字节）。
usedsize	bigint	使用内存大小（单位：字节）。

8.2.7.8 DBE_PERF.GLOBAL_SESSION_MEMORY_DETAIL

用于统计各节点的线程的内存使用情况，以MemoryContext节点来统计，如表8-91所示。

表 8-91 DBE_PERF.GLOBAL_SESSION_MEMORY_DETAIL 字段

名称	类型	描述
node_name	name	节点名称。
sessid	text	线程启动时间+线程标识。
sesstype	text	线程名称。
contextname	text	内存上下文名称。
level	smallint	内存上下文的重要级别。
parent	text	父级内存上下文名称。
totalsize	bigint	总申请内存大小（单位：字节）。
freesize	bigint	空闲内存大小（单位：字节）。
usedsize	bigint	使用内存大小（单位：字节）。

8.2.7.9 DBE_PERF.SESSION_STAT_ACTIVITY

显示当前节点上正在运行的线程相关的信息，如表8-92所示。

版权所有 © 华为云计算技术有限公司

表 8-92 DBE_PERF.SESSION_STAT_ACTIVITY 字段

名称	类型	描述
datid	oid	用户会话在后台连接到的数据库OID。
datname	name	用户会话在后台连接到的数据库名称。
pid	bigint	后台线程ID。
usesysid	oid	登录该后台的用户OID。
usename	name	登录该后台的用户名。
application_name	text	连接到该后台的应用名。
client_addr	inet	连接到该后台的客户端的IP地址。如果此字段为null，则表明客户端是通过服务器上的Unix套接字连接的，或者是内部线程，如果是autovacuum线程会显示autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
backend_start	timestamp with time zone	该过程开始的时间，即当客户端连接服务器时间。
xact_start	timestamp with time zone	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	开始当前活跃查询的时间，如果state的值不是active，则这个值是上一个查询的开始时间。如果是存储过程、函数、package，则查询的是第一个查询时间，不会随着存储过程内语句运行而改变。
state_change	timestamp with time zone	上次状态改变的时间。
waiting	boolean	如果后台当前正等待锁则为true。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
state	text	该后台当前总体状态。可能值是： • active：后台正在执行一个查询。 • idle：后台正在等待一个新的客户端命令。 • idle in transaction：后台在事务中，但是目前无法执行查询。 • idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 • fastpath function call：后台正在执行一个fast-path函数。 • disabled：如果后台禁用track_activities，则报告这个状态。 **说明** 普通用户只能查看到自己账户所对应的会话状态。即其他账户的state信息为空。例如以judy用户连接数据库后，在pg_stat_activity中查看到的普通用户joe及初始用户omm的state信息为空。 gaussdb=# SELECT datname, usename, usesysid,state,pid FROM pg_stat_activity; datname | usename | usesysid | state | pid ----------+---------+----------+-------- +-----------------testdb | omm | 10 | | 139968752121616 testdb | omm | 10 | | 139968903116560 db_tpcds | judy | 16398 | active | 139968391403280 testdb | omm | 10 | | 139968643069712 testdb | omm | 10 | | 139968680818448 testdb | joe | 16390 | | 139968563377936 (6 rows)
resource_pool	name	用户使用的资源池。
query_id	bigint	查询语句的ID。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。
unique_sql_id	bigint	语句的unique sql id。
trace_id	text	驱动传入的trace id，用于标识应用的一次请求。

8.2.7.10 DBE_PERF.GLOBAL_SESSION_STAT_ACTIVITY

显示集群内各节点上正在运行的线程相关的信息，如表8-93所示。

版权所有 © 华为云计算技术有限公司

表 8-93 DBE_PERF.GLOBAL_SESSION_STAT_ACTIVITY 字段

名称	类型	描述
coorname	text	CN节点名称。
datid	oid	用户会话在后台连接到的数据库OID。
datname	text	用户会话在后台连接到的数据库名称。
pid	bigint	后台线程ID。
usesysid	oid	登录该后台的用户OID。
username	text	登录该后台的用户名。
application_name	text	连接到该后台的应用名。
client_addr	inet	连接到该后台的客户端的IP地址。如果此字段为null，则表明客户端是通过服务器上的Unix套接字连接的，或者是内部线程，如果是autovacuum线程会显示autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后台通讯的TCP端口号，如果使用Unix套接字，则为-1。
backend_start	timestamp with time zone	该过程开始的时间，即当客户端连接服务器时间。
xact_start	timestamp with time zone	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
query_start	timestamp with time zone	开始当前活跃查询的时间，如果state的值不是active，则这个值是上一个查询的开始时间。如果是存储过程、函数、package，则查询的是第一个查询时间，不会随着存储过程内语句运行而改变。
state_change	timestamp with time zone	上次状态改变的时间。
waiting	boolean	如果后台当前正等待锁则为true，否则为false。如果等待的锁发生在DN，需要根据pg_thread_wait_status视图的wait_status字段确定发生的DN，在对应的DN查询pg_stat_activity的waiting字段确认是否发生了等待锁的行为。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
state	text	该后台当前总体状态。可能值是： ● active：后台正在执行一个查询。 ● idle：后台正在等待一个新的客户端命令。 ● idle in transaction：后台在事务中，但是目前无法执行查询。 ● idle in transaction (aborted)：后台在事务中，但事务中有语句执行失败。 ● fastpath function call：后台正在执行一个fast-path函数。 ● disabled：如果后台禁用track_activities，则报告这个状态。 **说明** 普通用户只能查看到自己账户所对应的会话状态。即其他账户的state信息为空。例如以judy用户连接数据库后，在pg_stat_activity中查看到的普通用户joe及初始用户omm的state信息为空。 gaussdb=# SELECT datname, usename, usesysid,state,pid FROM pg_stat_activity; datname | usename | usesysid | state | pid ----------+---------+----------+-------- +-----------------testdb | omm | 10 | | 139968752121616 testdb | omm | 10 | | 139968903116560 db_tpcds | judy | 16398 | active | 139968391403280 testdb | omm | 10 | | 139968643069712 testdb | omm | 10 | | 139968680818448 testdb | joe | 16390 | | 139968563377936 (6 rows)
resource_pool	name	用户使用的资源池。
query_id	bigint	查询语句的ID。
query	text	该后台的最新查询。如果state状态是active（活跃的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。
unique_sql_id	bigint	语句的unique sql id。
trace_id	text	驱动传入的trace id，用于标识应用的一次请求。

版权所有 © 华为云计算技术有限公司

8.2.7.11 DBE_PERF.THREAD_WAIT_STATUS

通过该视图可以检测当前节点上工作线程（backend thread）以及辅助线程（auxiliary thread）的阻塞等待情况，如表8-94所示。具体事件信息请参见表8-192。

表 8-94 DBE_PERF.THREAD_WAIT_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
db_name	text	数据库名称。
thread_name	text	线程名称。
query_id	bigint	查询ID，对应debug_query_id。
tid	bigint	当前线程的线程号。
sessionid	bigint	session的ID。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	streaming线程的父线程。
tlevel	integer	streaming线程的层级。
smpid	integer	并行线程的ID。
wait_status	text	当前线程的等待状态。等待状态的详细信息请参见表8-192。
wait_event	text	如果wait_status是acquire lock、acquire lwlock、wait io三种类型，此列描述具体的锁、轻量级锁、I/O的信息；否则为空。
locktag	text	当前线程正在等待锁的信息。
lockmode	text	当前线程正等待获取的锁模式。包含表级锁、行级锁、页级锁下的各模式。
block_sessionid	bigint	阻塞当前线程获取锁的会话标识。
global_sessionid	text	全局会话ID。

8.2.7.12 DBE_PERF.GLOBAL_THREAD_WAIT_STATUS

通过该视图可以检测所有节点上工作线程（backend thread）以及辅助线程（auxiliary thread）的阻塞等待情况，如表8-95所示。具体事件信息见表8-192。

通过GLOBAL_THREAD_WAIT_STATUS视图，可以查看集群全局各个节点上所有SQL语句产生的线程之间的调用层次关系，以及各个线程的阻塞等待状态，从而更容易定位hang以及类似现象的原因。

GLOBAL_THREAD_WAIT_STATUS视图和THREAD_WAIT_STATUS视图列定义完全相同，这是由于GLOBAL_THREAD_WAIT_STATUS视图本质是到集群中各个节点上查询THREAD_WAIT_STATUS视图汇总的结果。

版权所有 © 华为云计算技术有限公司

表 8-95 DBE_PERF.GLOBAL_THREAD_WAIT_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
db_name	text	数据库名称。
thread_name	text	线程名称。
query_id	bigint	查询ID，对应debug_query_id。
tid	bigint	当前线程的线程号。
sessionid	bigint	session的ID。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	streaming线程的父线程。
tlevel	integer	streaming线程的层级。
smpid	integer	并行线程的ID。
wait_status	text	当前线程的等待状态。等待状态的详细信息请参见表8-192。
wait_event	text	如果wait_status是acquire lock、acquire lwlock、wait io三种类型，此列描述具体的锁、轻量级锁、I/O的信息。否则是空。
locktag	text	当前线程正在等待锁的信息。
lockmode	text	当前线程正等待获取的锁模式。包含表级锁、行级锁、页级锁下的各模式。
block_sessionid	bigint	阻塞当前线程获取锁的会话标识。
global_sessionid	text	全局会话ID。

8.2.7.13 DBE_PERF.LOCAL_THREADPOOL_STATUS

该视图显示线程池中工作线程及会话的状态信息，如表8-96所示。该视图仅在线程池开启（enable_thread_pool=on）时生效。

表 8-96 DBE_PERF.LOCAL_THREADPOOL_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
group_id	integer	线程池组ID。
bind_numa_id	integer	该线程池组绑定的NUMA ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bind_cpu_number	integer	该线程池组绑定的CPU信息。如果未绑定CPU，该值为NULL。
listener	integer	该线程池组的Listener线程数量。
worker_info	text	线程池中线程相关信息，包括以下信息： • default：该线程池组中的初始线程数量。 • new：该线程池组中新增线程的数量。 • expect：该线程池组中预期线程的数量。 • actual：该线程池组中实际线程的数量。 • idle：该线程池组中空闲线程的数量。 • pending：该线程池组中等待线程的数量。
session_info	text	线程池中会话相关信息，包括以下信息： • total：该线程池组中所有的会话数量。 • waiting：该线程池组中等待调度的会话数量。 • running：该线程池中正在执行的会话数量。 • idle：该线程池组中空闲的会话数量。
stream_info	text	Stream线程池相关信息，包括以下信息： • total：该线程池组中所有的stream线程数量。 • running：该线程池组中当前正在工作的stream线程数量。 • idle：该线程池组中空闲的stream线程数量。

说明

线程池存在动态扩缩容机制，当扩容出来的新增线程空闲时，线程池会在3分钟左右进行缩容。动态缩容后，线程池总线程数量不会低于初始线程数量。

8.2.7.14 DBE_PERF.GLOBAL_THREADPOOL_STATUS

该视图显示在所有节点上的线程池中工作线程及会话的状态信息。

具体的字段请参见表8-96。

8.2.7.15 DBE_PERF.SESSION_CPU_RUNTIME

该视图显示当前用户执行复杂作业（正在运行）时的CPU使用信息，如表8-97所示。

版权所有 © 华为云计算技术有限公司

表 8-97 DBE_PERF.SESSION_CPU_RUNTIME 字段

名称	类型	描述
datid	oid	连接后端的数据库OID。
usename	name	登录到该后端的用户名。
pid	bigint	后端线程ID。
start_time	timestamp with time zone	语句执行的开始时间。如果是存储过程、函数、package，则查询的是第一个查询时间，不会随着存储过程内语句运行而改变。
min_cpu_time	bigint	语句在所有DN上的最小CPU时间，单位为ms。
max_cpu_time	bigint	语句在所有DN上的最大CPU时间，单位为ms。
total_cpu_time	bigint	语句在所有DN上的CPU总时间，单位为ms。
query	text	正在执行的语句。
node_group	text	语句所属用户对应的node group。
top_cpu_dn	text	CPU使用量topN信息。

8.2.7.16 DBE_PERF.SESSION_MEMORY_RUNTIME

该视图显示当前用户执行复杂作业（正在运行）时的内存使用信息，如表8-98所示。

表 8-98 DBE_PERF.SESSION_MEMORY_RUNTIME 字段

名称	类型	描述
datid	oid	连接后端的数据库OID。
usename	name	登录到该后端的用户名。
pid	bigint	后端线程ID。
start_time	timestamp with time zone	语句执行的开始时间。如果是存储过程、函数、package，则查询的是第一个查询时间，不会随着存储过程内语句运行而改变。
min_peak_memory	integer	语句在所有DN上的最小内存峰值大小，单位MB。
max_peak_memory	integer	语句在所有DN上的最大内存峰值大小，单位MB。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
spill_info	text	语句在所有DN上的下盘信息： • None：所有DN均未下盘。 • All：所有DN均下盘。 • [a:b]：数量为b个DN中有a个DN下盘。
query	text	正在执行的语句。
node_group	text	语句所属用户对应的node group。
top_mem_dn	text	mem使用量topN信息。

8.2.7.17 DBE_PERF.STATEMENT_IOSTAT_COMPLEX_RUNTIME

该视图显示当前用户执行作业正在运行时的I/O负载管理相关信息，如表8-99所示。以下涉及到iops，均以万次/s为单位。

表 8-99 DBE_PERF.STATEMENT_IOSTAT_COMPLEX_RUNTIME 字段

名称	类型	描述
query_id	bigint	作业ID。
mincurriops	integer	该作业当前I/O在各DN中的最小值。
maxcurriops	integer	该作业当前I/O在各DN中的最大值。
minpeakiops	integer	在作业运行时，作业I/O峰值中，各DN的最小值。
maxpeakiops	integer	在作业运行时，作业I/O峰值中，各DN的最大值。
io_limits	integer	该作业所设GUC参数io_limits。
io_priority	text	该作业所设GUC参数io_priority。
query	text	作业。
node_group	text	作业所属用户对应的node_group。
curr_io_limits	integer	使用io_priority管控I/O时的实时iolimits值。

8.2.7.18 DBE_PERF.LOCAL_ACTIVE_SESSION

该视图显示本节点上的ACTIVE SESSION PROFILE内存中的样本，如表8-100所示。

表 8-100 DBE_PERF.LOCAL_ACTIVE_SESSION 字段

名称	类型	描述
sampleid	bigint	采样ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sample_time	timestamp with time zone	采样的时间。
need_flush_sample	boolean	该样本是否需要刷新到磁盘。
databaseid	oid	数据库ID。
thread_id	bigint	线程的ID。
sessionid	bigint	会话的ID。
start_time	timestamp with time zone	会话的启动时间。
event	text	具体的事件名称。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	streaming线程的父线程。
tlevel	integer	streaming线程的层级。与执行计划的层级（id）相对应。
smpid	integer	smp执行模式下并行线程的并行编号。
userid	oid	session用户的ID。
application_name	text	应用的名称。
client_addr	inet	client端的地址。
client_hostname	text	client端的名称。
client_port	integer	客户端用于与后端通讯的TCP端口号。
query_id	bigint	唯一SQL id。
unique_query_id	bigint	归一化SQL id。
user_id	oid	unique query的key中的user_id。
cn_id	integer	cn id，在DN上表示该unique sql来自该CN节点，unique query的key中的cn_id。
unique_query	text	规范化后的UniqueSQL文本串。
locktag	text	会话等待锁信息，可通过locktag_decode解析。
lockmode	text	会话等待锁模式。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
block_sessionid	bigint	如果会话正在等待锁，阻塞该会话获取锁的会话标识。
final_block_sessionid	bigint	表示源头阻塞会话ID。
wait_status	text	描述event列的更多详细信息。
global_sessionid	text	全局会话ID。
xact_start_time	timestamp with time zone	事务开始时间。
query_start_time	timestamp with time zone	语句开始执行时间。
state	text	当前语句状态。 可能取值为：active、idle in transaction、fastpath function call、idle in transaction (aborted)、disabled、retrying。
event_start_time	timestamp with time zone	wait event的开始时间。
current_xid	xid	当前事务ID。
top_xid	xid	顶层事务ID。
parent_unique_sql_id	bigint	调用本条语句的父语句SQL_Id。
control_status	text	当进行异常SQL CPU资源管控时为under_cpu_throttle，否则为normal。
top_unique_query_id	bigint	被采样语句如果属于存储过程，该字段则展示存储过程顶层SQL的unique_query_id。
top_unique_query	text	被采样语句如果属于存储过程，该字段则显示存储过程顶层SQL的unique_query语句。
sql_hash	bigint	SQL语句对应的hash值，为0。
plan_hash	bigint	SQL语句执行计划对应的hash值，为0。

8.2.7.19 DBE_PERF.GLOBAL_ACTIVE_SESSION

该视图显示CN节点和主DN节点上的ACTIVE SESSION PROFILE内存中的样本的汇总，如**表8-101**所示。505.0版本新增event_start_time、current_xid、top_xid三个字段信息，由低于505.0版本升级至505.0或更高版本时，升级观察期无法查询。

版权所有 © 华为云计算技术有限公司

表 8-101 DBE_PERF.GLOBAL_ACTIVE_SESSION 字段

名称	类型	描述
node_name	text	节点名称。
sampleid	bigint	采样ID。
sample_time	timestamp without time zone	采样的时间。
need_flush_sample	boolean	该样本是否需要刷新的磁盘。
databaseid	oid	数据库ID。
thread_id	bigint	线程的ID。
sessionid	bigint	会话的ID。
start_time	timestamp without time zone	会话的启动时间。
event	text	具体的事件名称。
lwtid	integer	当前线程的轻量级线程号。
psessionid	bigint	streaming线程的父线程。
tlevel	integer	streaming线程的层级。与执行计划的层级（id）相对应。
smpid	integer	smp执行模式下并行线程的并行编号。
userid	oid	session用户的ID。
application_name	text	应用的名称。
client_addr	inet	client端的地址。
client_hostname	text	client端的名称。
client_port	integer	客户端用于与后端通讯的TCP端口号。
query_id	bigint	唯一SQL id。
unique_query_id	bigint	归一化SQL id。
user_id	oid	unique query的key中的user_id。
cn_id	integer	cn id，在DN上表示该unique sql来自该CN节点，unique query的key中的cn_id。
unique_query	text	规范化后的UniqueSQL文本串。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
locktag	text	会话等待锁信息，可通过locktag_decode解析。
lockmode	text	会话等待锁模式。
block_sessionid	bigint	如果会话正在等待锁，阻塞该会话获取锁的会话标识。
final_block_sessionid	bigint	表示源头阻塞会话ID。
wait_status	text	描述event列的更多详细信息。
global_sessionid	text	全局会话ID。
xact_start_time	timestamp with time zone	事务开始时间。
query_start_time	timestamp with time zone	语句开始执行时间。
state	text	当前语句状态。 可能取值为：active、idle in transaction、fastpath function call、idle in transaction (aborted)、disabled、retrying。
event_start_time	timestamp with time zone	wait event的开始时间。
current_xid	xid	当前事务ID。
top_xid	xid	顶层事务ID。
parent_unique_sql_id	bigint	调用本条语句的父语句SQL_Id。
sql_hash	bigint	SQL语句对应的hash值，为0。
plan_hash	bigint	SQL语句执行计划对应的hash值，为0。

8.2.8 Transaction

8.2.8.1 DBE_PERF.TRANSACTIONS_RUNNING_XACTS

显示当前节点运行事务的信息，如表8-102所示。

版权所有 © 华为云计算技术有限公司

表 8-102 DBE_PERF.TRANSACTIONS_RUNNING_XACTS 字段

名称	类型	描述
handle	integer	事务在GTM对应的句柄。
gxid	xid	事务id号。
state	tinyint	事务状态（3：prepared或者0：starting）。
node	text	节点名称。
xmin	xid	节点上当前数据涉及的最小事务号xmin。
vacuum	boolean	标志当前事务是否是lazy vacuum事务（lazy vacuum是一种vacuum机制，在需要时进行vacuum）。 ● true：表示是。 ● false：表示否。
timeline	bigint	标志数据库重启次数。
prepare_xid	xid	处于prepared状态的事务的id号，若不在prepared状态，值为0。
pid	bigint	事务对应的线程id。
next_xid	xid	CN传给DN的事务id号。
dbid	oid	事务对应的数据库id。

8.2.8.2 DBE_PERF.SUMMARY_TRANSACTIONS_RUNNING_XACTS

显示集群中各个CN节点运行事务的信息汇总，如**表8-103**所示。

表 8-103 DBE_PERF.SUMMARY_TRANSACTIONS_RUNNING_XACTS 字段

名称	类型	描述
handle	integer	事务在GTM对应的句柄。
gxid	xid	事务id号。
state	tinyint	事务状态（3：prepared或者0：starting）。
node	text	节点名称。
xmin	xid	节点上当前数据涉及的最小事务号xmin。
vacuum	boolean	标志当前事务是否是lazy vacuum事务（lazy vacuum是一种vacuum机制，在需要时进行vacuum）。 ● true：表示是。 ● false：表示否。
timeline	bigint	标志数据库重启次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
prepare_xid	xid	处于prepared状态的事务的id号，若不在prepared状态，值为0。
pid	bigint	事务对应的线程id。
next_xid	xid	CN传给DN的事务id号。
dbid	oid	事务对应的数据库id。

8.2.8.3 DBE_PERF.GLOBAL_TRANSACTIONS_RUNNING_XACTS

显示集群中各个CN和DN节点运行事务的信息，如表8-104所示。

表 8-104 DBE_PERF.GLOBAL_TRANSACTIONS_RUNNING_XACTS 字段

名称	类型	描述
handle	integer	事务在GTM对应的句柄。
gxid	xid	事务id号。
state	tinyint	事务状态（3：prepared或者0：starting）。
node	text	节点名称。
xmin	xid	节点上当前数据涉及的最小事务号xmin。
vacuum	boolean	标志当前事务是否是lazy vacuum事务（lazy vacuum是一种vacuum机制，在需要时进行vacuum）。 ● true：表示是。 ● false：表示否。
timeline	bigint	标志数据库重启次数。
prepare_xid	xid	处于prepared状态的事务的id号，若不在prepared状态，值为0。
pid	bigint	事务对应的线程id。
next_xid	xid	CN传给DN的事务id号。
dbid	oid	事务对应的数据库id。

8.2.8.4 DBE_PERF.TRANSACTIONS_PREPARED_XACTS

显示当前准备好进行两阶段提交的事务的信息，如表8-105所示。

版权所有 © 华为云计算技术有限公司

表 8-105 DBE_PERF.TRANSACTIONS_PREPARED_XACTS 字段

名称	类型	描述
transaction	xid	预备事务的数字事务标识。
gid	text	赋予该事务的全局事务标识。
prepared	timestamp with time zone	事务准备好提交的时间。
owner	name	执行该事务的用户的名称。
database	name	执行该事务所在的数据库名。

8.2.8.5 DBE_PERF.SUMMARY_TRANSACTIONS_PREPARED_XACTS

显示集群中各CN节点当前准备好进行两阶段提交的事务的信息汇总，如表8-106所示。

表 8-106 DBE_PERF.SUMMARY_TRANSACTIONS_PREPARED_XACTS 字段

名称	类型	描述
transaction	xid	预备事务的数字事务标识。
gid	text	赋予该事务的全局事务标识。
prepared	timestamp with time zone	事务准备好提交的时间。
owner	name	执行该事务的用户的名称。
database	name	执行该事务所在的数据库名。

8.2.8.6 DBE_PERF.GLOBAL_TRANSACTIONS_PREPARED_XACTS

显示集群中各个CN和DN节点当前准备好进行两阶段提交的事务的信息汇总，如表8-107所示。

表 8-107 DBE_PERF.GLOBAL_TRANSACTIONS_PREPARED_XACTS 字段

名称	类型	描述
transaction	xid	预备事务的数字事务标识。
gid	text	赋予该事务的全局事务标识。
prepared	timestamp with time zone	事务准备好提交的时间。
owner	name	执行该事务的用户的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
database	name	执行该事务所在的数据库名。

8.2.9 Query

8.2.9.1 DBE_PERF.STATEMENT

获得当前节点的执行语句（归一化SQL）的信息，如表8-108所示。CN上可以看到此CN接收到的归一化的SQL的全量统计信息（包含DN）；DN上仅可看到归一化的SQL的此节点执行的统计信息。

说明

不同的savepoint_name所生成的unique_sql_id不同，大量使用savepoint_name时会导致系统中产生的unique_sql_id信息快速上涨，若unique_sql_id数量高于instr_unique_sql_count数量时，新产生的unique_sql_id信息将不被统计。

表 8-108 DBE_PERF.STATEMENT 字段

名称	类型	描述
node_name	name	节点名称。
node_id	integer	节点的ID（pgxc_node中的node_id）。
user_name	name	用户名称，若查询时执行query语句的用户已不存在，则显示为*REMOVED_USER*。
user_id	oid	用户OID。
unique_sql_id	bigint	归一化的SQL ID。
query	text	归一化的SQL。 备注：长度受instr_unique_sql_size控制。
n_calls	bigint	实际调用次数。对于存储过程中的FETCH语句，其实际触发次数，与其关联cursor实际执行语句的n_calls增量一致。
min_elapse_time	bigint	SQL在内核内的最小运行时间（单位：微秒）。
max_elapse_time	bigint	SQL在内核内的最大运行时间（单位：微秒）。
total_elapse_time	bigint	SQL在内核内的总运行时间（单位：微秒）。
n_returned_rows	bigint	SELECT返回的结果集行数。
n_tuples_fetched	bigint	随机扫描行。
n_tuples_returned	bigint	顺序扫描行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
n_tuples_inserted	bigint	插入行。
n_tuples_updated	bigint	更新行。
n_tuples_deleted	bigint	删除行。
n_blocks_fetched	bigint	逻辑读次数。
n_blocks_hit	bigint	内存中命中次数。
n_soft_parse	bigint	软解析次数，n_soft_parse + n_hard_parse可能大于n_calls，因为子查询未计入n_calls。（该字段对非DML语句无意义）
n_hard_parse	bigint	硬解析次数，n_soft_parse + n_hard_parse可能大于n_calls，因为子查询未计入n_calls。对复杂数据类型的解析也会额外增加解析次数，例如smalldatetime类型。（该字段对非DML语句无意义）
db_time	bigint	有效的数据库内部时间花费，多线程将累加（单位：微秒）。
cpu_time	bigint	CPU时间（单位：微秒）。
execution_time	bigint	执行器内执行时间（单位：微秒）。
parse_time	bigint	SQL解析时间（单位：微秒）。（若解析阶段出现异常，不记录parse_time）
plan_time	bigint	SQL生成计划时间（单位：微秒）。（若计划生成阶段出现异常，不记录plan_time）
rewrite_time	bigint	SQL重写时间（单位：微秒）。（若重写阶段出现异常，不记录rewrite_time）
pl_execution_time	bigint	plpgsql上的执行时间（单位：微秒）。
pl_compilation_time	bigint	plpgsql上的编译时间（单位：微秒）。
data_io_time	bigint	I/O上的时间花费（单位：微秒）。
net_send_info	text	通过物理连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}

版权所有 © 华为云计算技术有限公司

名称	类型	描述
net_recv_info	text	通过物理连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}
net_stream_send_info	text	通过逻辑连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}
net_stream_recv_info	text	通过逻辑连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}
last_updated	timestamp with time zone	最后一次更新该语句的时间。
sort_count	bigint	排序执行的次数。
sort_time	bigint	排序执行的时间（单位：微秒），包括了sort算子和子算子的时间。
sort_mem_used	bigint	排序过程中使用的work memory大小（单位：KB）。
sort_spill_count	bigint	排序过程中，若发生落盘，写文件的次数。
sort_spill_size	bigint	排序过程中，若发生落盘，使用的文件大小（单位：KB）。
hash_count	bigint	hash执行的次数。
hash_time	bigint	hash执行的时间（单位：微秒）。
hash_mem_used	bigint	hash过程中使用的work memory大小（单位：KB）。
hash_spill_count	bigint	hash过程中，若发生落盘，写文件的次数。
hash_spill_size	bigint	hash过程中，若发生落盘，使用的文件大小（单位：KB）。
parent_unique_sql_id	bigint	父语句的unique_sql_id，非存储过程子语句该值为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
lock_wait_time	bigint	SQL的等锁时间（单位：微秒）。
total_used_memory	bigint	归一化语句的内存使用总量（单位：MB）。
max_used_memory	integer	单条语句的最大内存使用量（单位：MB）。
min_used_memory	integer	单条语句的最小内存使用量（单位：MB）。
db_name	name	连接后端的数据库名称，若查询时执行query语句的数据库已不存在，则显示为*REMOVED_DB*。
db_id	oid	连接后端的数据库OID。
eliminated_count	bigint	prepare语句产生的计划缓存淘汰次数。
plan_count	text	SQL的执行计划统计，包括LP、FQS、STREAM、PGXC（单位：次）。
aplan_count	bigint	aplan执行次数。
aplan_parse_time	bigint	aplan解析时间。
aplan_execution_time	bigint	aplan执行时间。
hard_parse_reason	text	硬解析原因。具体原因包括： ● first_execute：首次执行的硬解析次数。 ● cplan：生成cplan的硬解析次数。 ● gplan：生成gplan的硬解析次数。 ● aplan：生成aplan的硬解析次数。 ● revalid：生成计划失效后重建的硬解析次数。 ● memory_control：内存管控淘汰后重建计划的硬解析次数。 以上原因存在逻辑重叠，例如首次执行且生成cplan，则hard_parse_reason为{"first_execute":1, "cplan":1, "gplan":0, "aplan":0, "revalid":0, "memory_control":0}。 当revalid项不为0时，将显示计划失效的原因及次数，例如："revalid":1{invalid reason: guc changed:1}。
sql_hash	bigint	当前SQL语句的哈希值。当前模式记录为0。
plan_hash	bigint	当前SQL计划的哈希值。当前模式记录为0。

 版权所有 © 华为云计算技术有限公司

说明

- 由于查询计划管理（SQL Plan Management）不支持分布式，因此分布式下sql_hash、plan_hash不生效。
- DN上存储过程内语句的统计信息只汇聚到最外层语句，中间层嵌套调用的语句不包含子语句的统计信息。
- 存储过程内支持记录以下类型语句：**DML语句**、**动态语句**、**OPEN/FETCH 游标语句**、**嵌套子程序**、**RETURN QUERY语句**、PERFORM语句（通过PERFORM关键字调用存储过程的语句，或没有OUT出参的存储过程调用语句）。
- 当动态语句作为循环变量、游标、RETURN QUERY使用时，不会生成相应动态语句的记录。
- 执行存储过程后，清理DBE_PERF.STATEMENT表的记录并再次执行相同的存储过程，由于计划复用，不支持生成新的执行记录。
- 用户A执行过存储过程后，切换其他用户执行相同的存储过程，由于计划复用，不支持生成新的执行记录。
- 存储过程内语句CPU_TIME值为0。
- 如果存储过程内语句在统计信息上报阶段执行报错，DBE_PERF.STATEMENT中该语句记录的统计信息字段可能会都为0。

8.2.9.2 DBE_PERF.SUMMARY_STATEMENT

获得各CN节点的执行语句（归一化SQL）的全量信息（包含DN），如**表8-109**所示。

表 8-109 DBE_PERF.SUMMARY_STATEMENT 字段

名称	类型	描述
node_name	name	节点名称。
node_id	integer	节点的ID（pgxc_node中的node_id）。
user_name	name	用户名称，若查询时执行query语句的用户已不存在，则显示为*REMOVED_USER*。
user_id	oid	用户OID。
unique_sql_id	bigint	归一化的SQL id。
query	text	归一化的SQL。 备注：长度受instr_unique_sql_size控制。
n_calls	bigint	实际调用次数。对于存储过程中的FETCH语句，其实际触发次数，与其关联cursor实际执行语句的n_calls增量一致。
min_elapse_time	bigint	SQL在内核内的最小运行时间（单位：微秒）。
max_elapse_time	bigint	SQL在内核内的最大运行时间（单位：微秒）。
total_elapse_time	bigint	SQL在内核内的总运行时间（单位：微秒）。
n_returned_rows	bigint	SELECT返回的结果集行数。
n_tuples_fetched	bigint	随机扫描行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
n_tuples_returned	bigint	顺序扫描行。
n_tuples_inserted	bigint	插入行。
n_tuples_updated	bigint	更新行。
n_tuples_deleted	bigint	删除行。
n_blocks_fetched	bigint	逻辑读次数。
n_blocks_hit	bigint	内存中命中次数。
n_soft_parse	bigint	软解析次数，n_soft_parse + n_hard_parse可能大于n_calls，因为子查询未计入n_calls。（该字段对非DML语句无意义）
n_hard_parse	bigint	硬解析次数，n_soft_parse + n_hard_parse可能大于n_calls，因为子查询未计入n_calls。对复杂数据类型的解析也会额外增加解析次数，例如smalldatetime类型。（该字段对非DML语句无意义）
db_time	bigint	有效的数据库内部时间花费，多线程将累加（单位：微秒）。
cpu_time	bigint	CPU时间（单位：微秒）。
execution_time	bigint	执行器内执行时间（单位：微秒）。
parse_time	bigint	SQL解析时间（单位：微秒）。（若解析阶段出现异常，不记录parse_time）
plan_time	bigint	SQL生成计划时间（单位：微秒）。（若计划生成阶段出现异常，不记录plan_time）
rewrite_time	bigint	SQL重写时间（单位：微秒）。（若重写阶段出现异常，不记录rewrite_time）
pl_execution_time	bigint	plpgsql上的执行时间（单位：微秒）。
pl_compilation_time	bigint	plpgsql上的编译时间（单位：微秒）。
data_io_time	bigint	I/O上的时间花费（单位：微秒）。
net_send_info	text	通过物理连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}

版权所有 © 华为云计算技术有限公司

名称	类型	描述
net_recv_info	text	通过物理连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}
net_stream_send_info	text	通过逻辑连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}
net_stream_recv_info	text	通过逻辑连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。 例如：{"time":xxx, "n_calls":xxx, "size":xxx}
last_updated	timestamp with time zone	最后一次更新该语句的时间。
sort_count	bigint	排序执行的次数。
sort_time	bigint	排序执行的时间（单位：微秒），包括了sort算子和子算子的时间。
sort_mem_used	bigint	排序过程中使用的work memory大小（单位：KB）。
sort_spill_count	bigint	排序过程中，若发生落盘，写文件的次数。
sort_spill_size	bigint	排序过程中，若发生落盘，使用的文件大小（单位：KB）。
hash_count	bigint	hash执行的次数。
hash_time	bigint	hash执行的时间（单位：微秒）。
hash_mem_used	bigint	hash过程中使用的work memory大小（单位：KB）。
hash_spill_count	bigint	hash过程中，若发生落盘，写文件的次数。
hash_spill_size	bigint	hash过程中，若发生落盘，使用的文件大小（单位：KB）。
parent_unique_sql_id	bigint	父语句的unique_sql_id，非存储过程子语句该值为0。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
lock_wait_time	bigint	SQL的等锁时间（单位：微秒）。
total_used_memory	bigint	归一化语句的内存使用总量（单位：MB）。
max_used_memory	integer	单条语句的最大内存使用量（单位：MB）。
min_used_memory	integer	单条语句的最小内存使用量（单位：MB）。
db_name	name	连接后端的数据库名称，若查询时执行query语句的数据库已不存在，则显示为*REMOVED_DB*。
db_id	oid	连接后端的数据库OID。
eliminated_count	bigint	prepare语句产生的计划缓存的淘汰次数。
plan_count	text	SQL的执行计划统计，包括LP、FQS、STREAM、PGXC（单位：次）。
aplan_count	bigint	aplan执行次数。
aplan_parse_time	bigint	aplan解析时间。
aplan_execution_time	bigint	aplan执行时间。
hard_parse_reason	text	硬解析原因。具体原因包括： ● first_execute：首次执行的硬解析次数。 ● cplan：生成cplan的硬解析次数。 ● gplan：生成gplan的硬解析次数。 ● aplan：生成aplan的硬解析次数。 ● revalid：生成计划失效后重建的硬解析次数。 ● memory_control：内存管控淘汰后重建计划的硬解析次数。 以上原因存在逻辑重叠，例如首次执行且生成cplan，则hard_parse_reason为{"first_execute":1, "cplan":1, "gplan":0, "aplan":0, "revalid":0, "memory_control":0}。 当revalid项不为0时，将显示计划失效的原因及次数，例如："revalid":1{invalid reason: guc changed:1}。
sql_hash	bigint	当前SQL语句的哈希值。当前模式记录为0。
plan_hash	bigint	当前SQL计划的哈希值。当前模式记录为0。

 版权所有 © 华为云计算技术有限公司

8.2.9.3 DBE_PERF.STATEMENT_COUNT

显示数据库当前节点当前时刻执行的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）和（DDL、DML、DCL）统计信息，如表8-110所示。

说明

管理员权限用户查询STATEMENT_COUNT视图则能看到所有用户当前节点的统计信息。当集群或该节点重启时，计数将清零，并重新开始计数。计数以节点收到的查询数为准，包括集群内部进行的查询。例如，CN收到一条查询，如果一条查询语句包含多个子查询，这些子查询内容下发到DN时会被分别计数（无论单个DN还是多个DN）。

表 8-110 DBE_PERF.STATEMENT_COUNT 字段

名称	类型	描述
node_name	text	节点名称。
user_name	text	用户名。
select_count	bigint	SELECT语句统计结果。
update_count	bigint	UPDATE语句统计结果。
insert_count	bigint	INSERT语句统计结果。
delete_count	bigint	DELETE语句统计结果。
mergeinto_count	bigint	MERGE INTO语句统计结果。
ddl_count	bigint	DDL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。
dml_count	bigint	DML语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dcl_count	bigint	DCL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。
total_select_elapse	bigint	SELECT语句总的时间花费（单位：微秒）。
avg_select_elapse	bigint	SELECT语句平均的时间花费（单位：微秒）。
max_select_elapse	bigint	SELECT语句最大的时间花费（单位：微秒）。
min_select_elapse	bigint	SELECT语句最小的时间花费（单位：微秒）。
total_update_elapse	bigint	UPDATE语句总的时间花费（单位：微秒）。
avg_update_elapse	bigint	UPDATE语句平均的时间花费（单位：微秒）。
max_update_elapse	bigint	UPDATE语句最大的时间花费（单位：微秒）。
min_update_elapse	bigint	UPDATE语句最小的时间花费（单位：微秒）。
total_insert_elapse	bigint	INSERT语句总的时间花费（单位：微秒）。
avg_insert_elapse	bigint	INSERT语句平均的时间花费（单位：微秒）。
max_insert_elapse	bigint	INSERT语句最大的时间花费（单位：微秒）。
min_insert_elapse	bigint	INSERT语句最小的时间花费（单位：微秒）。
total_delete_elapse	bigint	DELETE语句总的时间花费（单位：微秒）。
avg_delete_elapse	bigint	DELETE语句平均的时间花费（单位：微秒）。
max_delete_elapse	bigint	DELETE语句最大的时间花费（单位：微秒）。
min_delete_elapse	bigint	DELETE语句最小的时间花费（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
dbid	oid	统计的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）所属的数据库ID。
user_dml_count	bigint	用户执行的DML语句的数量。user_dml_count的数量等于dml_count的数量减去bg_dml_count的数量。
bg_dml_count	bigint	数据库后台线程执行的DML语句的数量，包含全场景下后台线程（WORKER和THREADPOOL_WORKER以外的线程）执行的DML语句。 **说明** 当前存储过程中的命令不支持统计DML语句。

8.2.9.4 DBE_PERF.GLOBAL_STATEMENT_COUNT

显示数据库各节点当前时刻执行的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）和（DDL、DML、DCL）统计信息，如表8-111所示。

表 8-111 DBE_PERF.GLOBAL_STATEMENT_COUNT 字段

名称	类型	描述
node_name	text	节点名称。
user_name	text	用户名。
select_count	bigint	SELECT语句统计结果。
update_count	bigint	UPDATE语句统计结果。
insert_count	bigint	INSERT语句统计结果。
delete_count	bigint	DELETE语句统计结果。
mergeinto_count	bigint	MERGE INTO语句统计结果。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
ddl_count	bigint	DDL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。
dml_count	bigint	DML语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。
dcl_count	bigint	DCL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。
total_select_elapse	bigint	SELECT语句总的时间花费（单位：微秒）。
avg_select_elapse	bigint	SELECT语句平均的时间花费（单位：微秒）。
max_select_elapse	bigint	SELECT语句最大的时间花费（单位：微秒）。
min_select_elapse	bigint	SELECT语句最小的时间花费（单位：微秒）。
total_update_elapse	bigint	UPDATE语句总的时间花费（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
avg_update_elapse	bigint	UPDATE语句平均的时间花费（单位：微秒）。
max_update_elapse	bigint	UPDATE语句最大的时间花费（单位：微秒）。
min_update_elapse	bigint	UPDATE语句最小的时间花费（单位：微秒）。
total_insert_elapse	bigint	INSERT语句总的时间花费（单位：微秒）。
avg_insert_elapse	bigint	INSERT语句平均的时间花费（单位：微秒）。
max_insert_elapse	bigint	INSERT语句最大的时间花费（单位：微秒）。
min_insert_elapse	bigint	INSERT语句最小的时间花费（单位：微秒）。
total_delete_elapse	bigint	DELETE语句总的时间花费（单位：微秒）。
avg_delete_elapse	bigint	DELETE语句平均的时间花费（单位：微秒）。
max_delete_elapse	bigint	DELETE语句最大的时间花费（单位：微秒）。
min_delete_elapse	bigint	DELETE语句最小的时间花费（单位：微秒）。
dbid	oid	统计的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）所属的数据库ID。
user_dml_count	bigint	用户执行的DML语句的数量。user_dml_count的数量等于dml_count的数量减去bg_dml_count的数量。
bg_dml_count	bigint	数据库后台线程执行的DML语句的数量，包含全场景下后台线程（WORKER和THREADPOOL_WORKER以外的线程）执行的DML语句。 **说明** 当前存储过程中的命令不支持统计DML语句。

8.2.9.5 DBE_PERF.SUMMARY_STATEMENT_COUNT

显示数据库各节点执行的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）和（DDL、DML、DCL）的汇总统计信息，如表8-112所示。

版权所有 © 华为云计算技术有限公司

表 8-112 DBE_PERF.SUMMARY_STATEMENT_COUNT 字段

名称	类型	描述
user_name	text	用户名。
select_count	numeric	SELECT语句统计结果。
update_count	numeric	UPDATE语句统计结果。
insert_count	numeric	INSERT语句统计结果。
delete_count	numeric	DELETE语句统计结果。
mergeinto_count	numeric	MERGE INTO语句统计结果。
ddl_count	numeric	DDL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“TRUNCATE”、“CLUSTER”、“VACUUM”、“ANALYZE”、“CREATE”（不包括“CREATE ROLE”和“CREATE USER”）、“DROP”（不包括“DROP ROLE”和“DROP USER”）、“ALTER”（不包括“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”和“ALTER SESSION”）、“REINDEX”、“COMMENT”、“BARRIER”、“SECURITY LABEL ON”、“PURGE”、“TIMECAPSULE”、“REFRESH”、“IMPDP”和“EXPDP”语句。
dml_count	numeric	DML语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SELECT”、“UPDATE”、“INSERT”、“DELETE”、“MERGE”、“PREPARE”、“EXPLAIN”、“LOCK”、“COPY”、“ANONYMOUS”、“DELTA”、“EXECUTE”、“MOVE”、“FETCH”、“CLOSE”、“DEALLOCATE”、“DECLARE”、“CURSOR”和“LOAD”语句。
dcl_count	numeric	DCL语句的数量。该统计结果包含用户执行的SQL语句和数据库后台线程执行的SQL语句，包括“SHOW”、“CREATE ROLE”、“CREATE USER”、“DROP ROLE”、“DROP USER”、“ALTER ROLE”、“ALTER USER”、“ALTER DEFAULT PRIVILEGES”、“GRANT”、“REVOKE”、“REASSIGN”、“SET”、“RESET”和“ALTER SESSION”语句。
total_select_elapse	numeric	SELECT语句总的时间花费（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
avg_select_elapse	bigint	SELECT语句平均的时间花费（单位：微秒）。
max_select_elapse	bigint	SELECT语句最大的时间花费（单位：微秒）。
min_select_elapse	bigint	SELECT语句最小的时间花费（单位：微秒）。
total_update_elapse	numeric	UPDATE语句总的时间花费（单位：微秒）。
avg_update_elapse	bigint	UPDATE语句平均的时间花费（单位：微秒）。
max_update_elapse	bigint	UPDATE语句最大的时间花费（单位：微秒）。
min_update_elapse	bigint	UPDATE语句最小的时间花费（单位：微秒）。
total_insert_elapse	numeric	INSERT语句总的时间花费（单位：微秒）。
avg_insert_elapse	bigint	INSERT语句平均的时间花费（单位：微秒）。
max_insert_elapse	bigint	INSERT语句最大的时间花费（单位：微秒）。
min_insert_elapse	bigint	INSERT语句最小的时间花费（单位：微秒）。
total_delete_elapse	numeric	DELETE语句总的时间花费（单位：微秒）。
avg_delete_elapse	bigint	DELETE语句平均的时间花费（单位：微秒）。
max_delete_elapse	bigint	DELETE语句最大的时间花费（单位：微秒）。
min_delete_elapse	bigint	DELETE语句最小的时间花费（单位：微秒）。
dbid	oid	统计的五类语句（SELECT、INSERT、UPDATE、DELETE、MERGE INTO）所属的数据库ID。
user_dml_count	bigint	用户执行的DML语句的数量。 user_dml_count的数量等于dml_count的数量减去bg_dml_count的数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
bg_dml_count	bigint	数据库后台线程执行的DML语句的数量，包含全场景下后台线程（WORKER和THREADPOOL_WORKER以外的线程）执行的DML语句。 **说明** 当前存储过程中的命令不支持统计DML语句。

8.2.9.6 DBE_PERF.STATEMENT_RESPONSETIME_PERCENTILE

获取集群SQL响应时间P80，P95分布信息，如表8-113所示。

表 8-113 DBE_PERF.STATEMENT_RESPONSETIME_PERCENTILE 字段

名称	类型	描述
p80	bigint	集群80%的SQL的响应时间（单位：微秒）。
p95	bigint	集群95%的SQL的响应时间（单位：微秒）。

8.2.9.7 DBE_PERF.STATEMENT_HISTORY

获得当前节点的执行语句的信息。只可在系统库中查询到结果，用户库中无法查询，如表8-114所示。

表 8-114 DBE_PERF.STATEMENT_HISTORY 字段

名称	类型	描述
db_name	name	数据库名称。
schema_name	name	模式名称。
origin_node	integer	节点名称。
user_name	name	用户名。
application_name	text	用户发起的请求的应用程序名称。
client_addr	text	用户发起的请求的客户端地址。
client_port	integer	用户发起的请求的客户端端口。
unique_query_id	bigint	归一化SQL ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
debug_query_id	bigint	唯一SQL ID。部分语句存在不唯一的情况，如Parse报文、DCL和TCL等语句的debug_query_id值为0。存储过程内的语句使用同一个ID。
query	text	归一化SQL（仅CN上有值）。
start_time	timestamp with time zone	语句启动的时间。
finish_time	timestamp with time zone	语句结束的时间。
slow_sql_threshold	bigint	语句执行时慢SQL的标准。
transaction_id	bigint	事务ID。
thread_id	bigint	执行线程ID。
session_id	bigint	用户session id。
n_soft_parse	bigint	软解析次数，n_soft_parse + n_hard_parse可能大于n_calls，因为子查询未计入n_calls。（该字段对非DML语句无意义）
n_hard_parse	bigint	硬解析次数，n_soft_parse + n_hard_parse可能大于n_calls，因为子查询未计入n_calls。对复杂数据类型的解析也会额外增加解析次数，例如smalldatetime类型。（该字段对非DML语句无意义）
query_plan	text	语句执行计划。
n_returned_rows	bigint	SELECT返回的结果集行数。
n_tuples_fetched	bigint	随机扫描行，即回表次数。回表是指在使用索引进行查询时，如果查询的列不在索引中，数据库引擎需要通过索引中的主键或唯一标识符再次访问主表以获取所需的完整数据行的过程。
n_tuples_returned	bigint	顺序扫描行，通过顺序扫描获取的活动行数与通过索引扫描返回的索引条目数之和。
n_tuples_inserted	bigint	插入行。
n_tuples_updated	bigint	更新行。
n_tuples_deleted	bigint	删除行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
n_blocks_fetched	bigint	buffer的块访问次数，表示从缓冲区和磁盘中读取block的总数。
n_blocks_hit	bigint	buffer的块命中次数，表示从缓冲区中读取block的数量。
db_time	bigint	有效的DB时间花费，多线程将累加（单位：微秒）。
cpu_time	bigint	CPU消耗时间（单位：微秒）。
execution_time	bigint	执行器内执行时间（单位：微秒）。
parse_time	bigint	SQL解析时间（单位：微秒）。（若解析阶段出现异常，不记录parse_time）
plan_time	bigint	SQL生成计划时间（单位：微秒）。（若计划生成阶段出现异常，不记录plan_time）
rewrite_time	bigint	SQL重写时间（单位：微秒）。（若重写阶段出现异常，不记录rewrite_time）
pl_execution_time	bigint	plpgsql上的执行时间（单位：微秒）。
pl_compilation_time	bigint	plpgsql上的编译时间（单位：微秒）。
data_io_time	bigint	I/O上的时间花费（单位：微秒）。
net_send_info	text	通过物理连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。
net_recv_info	text	通过物理连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，CN与CN、CN与客户端以及CN与DN之间都是通过物理连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。
net_stream_send_info	text	通过逻辑连接发送消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
net_stream_recv_info	text	通过逻辑连接接收消息的网络状态，包含时间（微秒）、调用次数、吞吐量（字节）。在分布式数据库中，不同分片的DN之间通过逻辑连接进行通信，通过该字段可以分析SQL在分布式系统下的网络开销。例如：{"time":xxx, "n_calls":xxx, "size":xxx}。
lock_count	bigint	加锁次数。
lock_time	bigint	加锁耗时。
lock_wait_count	bigint	加锁等待次数。
lock_wait_time	bigint	加锁等待耗时。
lock_max_count	bigint	最大持锁数量。
lwlock_count	bigint	轻量级加锁次数（预留）。
lwlock_wait_count	bigint	轻量级加锁等待次数。
lwlock_time	bigint	轻量级加锁时间（预留）。
lwlock_wait_time	bigint	轻量级加锁等待耗时。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
details	bytea	等待事件、语句锁事件和GSTrace函数运行时间的列表。 记录级别的值>=L0时，开始记录等待事件的列表。显示当前节点wait event的相关统计信息。内核中关键的事件信息见**表8-192**。或从视图wait_event_info中查看系统中所有的事件列表。关于每种事务锁对业务的影响程度，请参考**LOCK**语法小节的详细描述。通过GStrace全链路跟踪函数，GStrace字段函数名function，以树形结构表达函数父子关系；GStrace字段函数运行时间duration字段表示函数运行时间，单位为微秒。 记录级别的值是L2时，开始记录语句锁事件的列表，该列表按时间顺序记录事件，记录的锁事件数量受参数track_stmt_details_size的影响。 该字段为二进制，需要借助解析函数pg_catalog.statement_detail_decode进行读取，请参见（**表1-204**）。 事件包括： • 加锁开始 LOCK_START • 加锁结束 LOCK_END • 等锁开始 LOCK_WAIT_START • 等锁结束 LOCK_WAIT_END • 释放锁 LOCK_RELEASE • 轻量级等锁开始 LWLOCK_WAIT_START • 轻量级等锁结束 LWLOCK_WAIT_END
is_slow_sql	boolean	该SQL是否为slow SQL。
trace_id	text	驱动传入的trace id，与应用的一次请求相关联。
advise	text	可能导致该SQL为slow SQL的风险信息。 **说明** 如果语句通过SQLBypass方式执行（可通过explain查看计划判断，请参见**enable_opfusion**的参数说明），可以认为无慢SQL风险，本字段信息可能会缺失。
parent_unique_sql_id	bigint	当前语句的外层SQL的归一化SQL ID，对于存储过程内执行的语句，该值为调用存储过程语句的归一化SQL ID，存储过程外的语句该值为0。
finish_status	text	语句完成状态，默认值normal。cancelled表示被取消或被terminate。error表示除cancelled外，其他异常导致的结束。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
used_memory	integer	单条语句内存使用量（单位：MB）。
lock_max_local_count	bigint	经过locallock机制持锁的最大持锁数量。
lock_max_fastpath_count	bigint	经过fastpath机制持锁的最大持锁数量。
lock_max_globall_count	bigint	经过主锁表持锁的最大持锁数量。
driver_start_time	timestamp with time zone	JDBC驱动开始执行的时间点。
driver_wait_response	timestamp with time zone	JDBC驱动发送报文后的开始等待接收数据库返回数据的时间点。
driver_finish_time	timestamp with time zone	JDBC驱动接收数据库返回数据完成的时间点。
driver_info	json	JDBC驱动各报文发送和接收的时间点。 参数track_stmt_stat_level修改后，会在新建立连接的会话生效，现有会话不生效。
kernel_info	json	数据库各报文发送和接收的时间点。
adaptive_join_states	text	记录自适应连接算子最终选用的连接方式，记录格式为[plan_node_id, use_hashjoin]，第一位代表对应的plan_node_id，第二位代表是否使用HashJoin，1代表是，0代表否。
aplan_count	bigint	aplan执行次数。
aplan_parse_time	bigint	aplan解析时间。
aplan_execution_time	bigint	aplan执行时间。
sql_hash	bigint	当前SQL语句的哈希值。
plan_hash	bigint	当前SQL计划的哈希值。
plan_hash_prev	bigint	如果是Generic Plan，此值为上一次使用的计划的哈希值。
relids	text	语句中涉及表的pg_class.oid，多表按“,”隔开。

版权所有 © 华为云计算技术有限公司

说明

- 当track_stmt_stat_level中Full SQL和Slow SQL不全为OFF时，对于超过log_min_duration_statement阈值的SQL，is_slow_sql字段会记录为true，表明这是一个超过阈值的SQL，其他字段以track_stmt_stat_level中的级别为记录标准。
- 由于自适应连接不支持分布式，分布式下adaptive_join_states不生效。
- 由于查询计划管理（SQL Plan Management）不支持分布式，分布式下sql_hash、plan_hash和plan_hash_prev不生效。
- 分布式存储过程子语句锁相关指标不会向顶层语句进行汇聚。
- 当查询计划记录失败时，会在query_plan字段中提示记录失败的原因，如以下场景：
  - 在LightProxy场景下语句会下推，不在CN节点上记录计划，会在DN节点上记录。提示：“The current plan is lightproxy. Please check the query_plan on the DN node.”。
  - 在Stream场景下计划会下推，会在CN节点上记录计划，不会在DN节点上记录。提示：“The current plan is streaming. Please check the query_plan on the CN node.”。
  - 对于DML语句自动提交的COMMIT，不会记录相关计划。提示：“The current record is a transaction commit of a DML statement. Please check the plan of this DML statement.”。
  - 非DML/DQL语句不记录计划。提示：“Do not record plans for non-DML/DQL statements.”。对于DDL语句，在多CN集群上，query_plan会记录节点间的交互信息。
  - 存储过程中未适配或没有计划的子语句，例如简单表达式、OPEN CURSOR、CLOSE以及BYPASS路径语句，不记录计划。存储过程中执行的子语句不在DN上记录计划。提示：“Statement has no plan or is unsupported for plan recording.”。
  - 对于进入到PARSE阶段的SQL，还未生成计划，将不进行记录。提示：“This statement is a PARSE statement and has not yet generated a query plan.”。
  - 在检查是否需要上报query_plan时，SQL语句执行并不慢，后续清理等任务影响导致SQL语句整体执行慢。提示：“Check the reported query_plan time is 2026-01-01 08:05:01.735619+08, at this time the SQL is not slow.”。
  - 语句执行状态异常时，计划记录有可能失败，需要检查日志分析原因。提示：“An error occurred during statement execution. Please check the log for analysis.”。
  - 其他未记录成功的场景，需要检查日志分析原因。提示：“The query plan failed to be recorded. Please check the log analysis.”。

8.2.9.8 DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY

显示各个节点执行作业结束后的负载管理记录，如表8-115所示。

表 8-115 DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY 字段

名称	类型	描述
datid	oid	连接后端的数据库OID。
dbname	text	连接后端的数据库名称。
schemaname	text	模式的名称。
nodename	text	语句执行的CN名称。
username	text	连接到后端的用户名。
application_name	text	连接到后端的应用名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
client_addr	inet	连接到后端的客户端的IP地址。 如果此字段是null，它表明通过服务器机器上UNIX套接字连接客户端或者这是内部线程，如autovacuum。
client_hostname	text	客户端的主机名，这个字段是通过client_addr的反向DNS查找得到。这个字段只有在启动log_hostname且使用IP连接时才非空。
client_port	integer	客户端用于与后端通讯的TCP端口号，如果使用UNIX套接字，则为-1。
query_band	text	用于标示作业类型，可通过GUC参数query_band进行设置，默认为空字符串。
block_time	bigint	语句执行前的阻塞时间，包含语句解析和优化时间，单位ms。
start_time	timestamp with time zone	语句执行的开始时间。
finish_time	timestamp with time zone	语句执行的结束时间。
duration	bigint	语句实际执行的时间，单位ms。
estimate_total_time	bigint	语句预估执行时间，单位ms。
status	text	语句执行结束状态：正常为finished，异常为aborted。
abort_info	text	语句执行结束状态为aborted时显示异常信息。
resource_pool	text	用户使用的资源池。
control_group	text	语句所使用的Cgroup。
estimate_memory	integer	语句预估使用内存。
min_peak_memory	integer	语句在所有DN上的最小内存峰值，单位MB。
max_peak_memory	integer	语句在所有DN上的最大内存峰值，单位MB。
average_peak_memory	integer	语句执行过程中的内存使用平均值，单位MB。
memory_skew_percent	integer	语句各DN间的内存使用倾斜率。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
spill_info	text	语句在所有DN上的下盘信息： ● None：所有DN均未下盘。 ● All：所有DN均下盘。 ● [a:b]：数量为b个DN中有a个DN下盘。
min_spill_size	integer	若发生下盘，所有DN上下盘的最小数据量，单位MB，默认为0。
max_spill_size	integer	若发生下盘，所有DN上下盘的最大数据量，单位MB，默认为0。
average_spill_size	integer	若发生下盘，所有DN上下盘的平均数据量，单位MB，默认为0。
spill_skew_percent	integer	若发生下盘，DN间下盘倾斜率。
min_dn_time	bigint	语句在所有DN上的最小执行时间，单位ms。
max_dn_time	bigint	语句在所有DN上的最大执行时间，单位ms。
average_dn_time	bigint	语句在所有DN上的平均执行时间，单位ms。
dntime_skew_percent	integer	语句在各DN间的执行时间倾斜率。
min_cpu_time	bigint	语句在所有DN上的最小CPU时间，单位ms。
max_cpu_time	bigint	语句在所有DN上的最大CPU时间，单位ms。
total_cpu_time	bigint	语句在所有DN上的CPU总时间，单位ms。
cpu_skew_percent	integer	语句在DN间的CPU时间倾斜率。
min_peak_iops	integer	语句在所有DN上的每秒最小I/O峰值（单位是万次/s）。
max_peak_iops	integer	语句在所有DN上的每秒最大I/O峰值（单位是万次/s）。
average_peak_iops	integer	语句在所有DN上的每秒平均I/O峰值（单位是万次/s）。
iops_skew_percent	integer	语句在DN间的I/O倾斜率。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
warning	text	主要显示如下几类告警信息： ● Spill file size larger than 256MB ● Broadcast size larger than 100MB ● Early spill ● Spill times is greater than 3 ● Spill on memory adaptive ● Hash table conflict
queryid	bigint	语句执行使用的内部query id。
query	text	执行的语句。
query_plan	text	语句的执行计划。
cpu_top1_node_name	text	cpu使用率第1的节点名称。
cpu_top2_node_name	text	cpu使用率第2的节点名称。
cpu_top3_node_name	text	cpu使用率第3的节点名称。
cpu_top4_node_name	text	cpu使用率第4的节点名称。
cpu_top5_node_name	text	cpu使用率第5的节点名称。
mem_top1_node_name	text	内存使用量第1的节点名称。
mem_top2_node_name	text	内存使用量第2的节点名称。
mem_top3_node_name	text	内存使用量第3的节点名称。
mem_top4_node_name	text	内存使用量第4的节点名称。
mem_top5_node_name	text	内存使用量第5的节点名称。
cpu_top1_value	bigint	cpu使用率第1的值。
cpu_top2_value	bigint	cpu使用率第2的值。
cpu_top3_value	bigint	cpu使用率第3的值。
cpu_top4_value	bigint	cpu使用率第4的值。
cpu_top5_value	bigint	cpu使用率第5的值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
mem_top1_value	bigint	内存使用量第1的值。
mem_top2_value	bigint	内存使用量第2的值。
mem_top3_value	bigint	内存使用量第3的值。
mem_top4_value	bigint	内存使用量第4的值。
mem_top5_value	bigint	内存使用量第5的值。
top_mem_dn	text	内存使用量topN信息。
top_cpu_dn	text	cpu使用量topN信息。
node_group	text	语句所属用户对应的node group。

8.2.9.9 DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY_TABLE

显示各个节点执行作业结束后的负载管理记录。此数据是从内核中转储到系统表中的数据。当设置GUC参数enable_resource_record为on时，系统会定时（周期为3分钟）将**DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY**中的记录导入此系统表，开启此功能会占用系统存储空间并对性能有一定影响，不建议用户使用。

具体的字段请参考**DBE_PERF.GLOBAL_STATEMENT_COMPLEX_HISTORY**中的字段。

8.2.9.10 DBE_PERF.STATEMENT_WLMSTAT_COMPLEX_RUNTIME

显示当前用户执行作业正在运行时的负载管理相关信息，如**表8-116**所示。

表 8-116 DBE_PERF.STATEMENT_WLMSTAT_COMPLEX_RUNTIME 字段

名称	类型	描述
datid	oid	连接后端的数据库OID。
datname	name	连接后端的数据库名称。
threadid	bigint	后端线程id。
processid	integer	后端线程的pid。
usesysid	oid	登录后端的用户OID。
appname	text	连接到后端的应用名。
usename	name	登录到该后端的用户名。
priority	bigint	语句所在Cgroups的优先级。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
attribute	text	语句的属性： ● Ordinary：语句发送到数据库后被解析前的默认属性。 ● Simple：简单语句。 ● Complicated：复杂语句。 ● Internal：数据库内部语句。
block_time	bigint	语句当前为止的pending的时间，单位s。
elapsed_time	bigint	语句当前为止的实际执行时间，单位s。
total_cpu_time	bigint	语句在上一时间周期内的DN上CPU使用的总时间，单位s。
cpu_skew_percent	integer	语句在上一时间周期内的DN上CPU使用的倾斜率。
statement_mem	integer	语句执行使用的statement_mem，预留字段。
active_points	integer	语句占用的资源池并发点数。
dop_value	integer	语句的从资源池中获取的dop值。
control_group	text	语句当前所使用的Cgroups。
status	text	语句当前的状态，包括： ● pending：执行前状态。 ● running：执行进行状态。 ● finished：执行正常结束。（当enqueue字段为StoredProc或Transaction时，仅代表语句中的部分作业已经执行完毕，该状态会持续到该语句完全执行完毕。） ● aborted：执行异常终止。 ● active：非以上四种状态外的正常状态。 ● unknown：未知状态。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
enqueue	text	语句当前的排队情况，包括： • Global：在全局队列中排队。 • Respool：在资源池队列中排队。 • CentralQueue：在中心协调节点（CCN）中排队。 • Transaction：语句处于一个事务块中。 • StoredProc：语句处于一个存储过程中。 • None：未在排队。 • Forced None：事务块语句或存储过程语句由于超出设定的等待时间而强制执行。
resource_pool	name	语句当前所在的资源池。
query	text	该后端的最新查询。如果state状态是active（活的），此字段显示当前正在执行的查询。所有其他情况表示上一个查询。

8.2.9.11 DBE_PERF.SESSION_SQL_MEMORY

显示当前用户在当前节点上正在执行的语句的内存使用量，如表8-117所示。

表 8-117 DBE_PERF.SESSION_SQL_MEMORY 字段

名称	类型	描述
pid	bigint	线程ID。
sessionid	bigint	会话ID。
unique_sql_id	bigint	语句的unique sql id。
query_id	bigint	语句的debug query id。
query	text	该会话当前运行语句。
peak_used_memory	integer	该语句内存使用峰值（单位：MB）。
current_used_memory	integer	该语句当前内存使用值（单位：MB）。

8.2.9.12 DBE_PERF.GLOBAL_SESSION_SQL_MEMORY

显示当前用户在各个节点上正在执行的语句的内存使用量，如表8-118所示。

版权所有 © 华为云计算技术有限公司

表 8-118 DBE_PERF.GLOBAL_SESSION_SQL_MEMORY 字段

名称	类型	描述
node_name	text	当前集群下的节点名称。
pid	bigint	线程ID。
sessionid	bigint	会话ID。
unique_sql_id	bigint	语句的unique sql id。
query_id	bigint	语句的debug query id。
query	text	该会话当前运行语句。
peak_used_memory	integer	该语句内存使用峰值（单位：MB）。
current_used_memory	integer	该语句当前内存使用值（单位：MB）。

8.2.9.13 DBE_PERF.GS_PLANCACHE_STAT

显示当前进程内全部session的参数化语句信息，如表8-119所示。查看该视图需要MONADMIN权限。

说明

同一语句受GUC参数影响可能生成不同计划，该语句在视图中将对应展示多次。

表 8-119 DBE_PERF.GS_PLANCACHE_STAT 字段

名称	类型	描述
pid	bigint	后台线程ID。
sessionid	bigint	当前会话ID。
global_sessionid	text	全局会话ID。
database_id	oid	数据库ID。
parameterized_sql	text	参数化后的模板SQL。
saved_time	timestamp with time zone	保存该模板SQL语句缓存时的时间戳。
last_used_time	timestamp with time zone	保存该模板SQL语句最近访问的时间戳。
parameter_types	regtype[]	该模板SQL期望的参数类型。
search_path	text	搜索路径。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
used_count	bigint	使用该模板SQL的次数。
is_deleted	bool	该SQL语句的计划缓存是否已被内存管控淘汰。
size	bigint	该SQL语句缓存计划占用的内存大小，单位为字节。
valid	bool	该SQL语句缓存计划是否有效。
is_parameterized	bool	该SQL语句是否被参数化。
opcache_reuse_cnt	integer	执行缓存复用的次数。
opcache_mem_size	bigint	执行缓存占用的内存大小。

8.2.9.14 DBE_PERF.GS_DIST_PLANCACHE_STAT

显示全部进程（包含备机）的参数化语句信息，如表8-120所示。查看该视图需要MONADMIN权限。

说明

同一语句受GUC参数影响可能生成不同计划，该语句在视图中将对应展示多次。

表 8-120 DBE_PERF.GS_DIST_PLANCACHE_STAT 字段

名称	类型	描述
node_name	name	节点名称。
pid	bigint	后台线程ID。
sessionid	bigint	当前会话ID。
global_sessionid	text	全局会话ID。
database_id	oid	数据库ID。
parameterized_sql	text	参数化后的模板SQL。
saved_time	timestamp with time zone	保存该模板SQL语句缓存时的时间戳。
last_used_time	timestamp with time zone	保存该模板SQL语句最近访问的时间戳。
parameter_types	regtype[]	该模板SQL期望的参数类型。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
search_path	text	搜索路径。
used_count	bigint	使用该模板SQL的次数。
is_deleted	bool	该SQL语句的计划缓存是否已经被内存管控淘汰。
size	bigint	该SQL语句缓存计划占用的内存大小，单位为字节。
valid	bool	该SQL语句缓存计划是否有效。

8.2.10 Cache/IO

8.2.10.1 DBE_PERF.STATIO_USER_TABLES

该视图显示命名空间中所有用户关系表的I/O状态信息，如表8-121所示。

表 8-121 DBE_PERF.STATIO_USER_TABLES 字段

名称	类型	描述
relid	oid	表OID。
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表缓存命中数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	bigint	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓冲区数（如果存在）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.10.2 DBE_PERF.SUMMARY_STATIO_USER_TABLES

该视图显示集群内各节点的用户关系表的I/O状态汇总信息，如表8-122所示。

表 8-122 DBE_PERF.SUMMARY_STATIO_USER_TABLES 字段

名称	类型	描述
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	numeric	从该表中读取的磁盘块数。
heap_blks_hit	numeric	该表缓存命中数。
idx_blks_read	numeric	从表中所有索引读取的磁盘块数。
idx_blks_hit	numeric	表中所有索引命中缓存数。
toast_blks_read	numeric	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	numeric	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	numeric	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	numeric	该表的TOAST表索引命中缓冲区数（如果存在）。

8.2.10.3 DBE_PERF.GLOBAL_STATIO_USER_TABLES

该视图显示各节点的命名空间中所有用户关系表的I/O状态信息，如表8-123所示。

表 8-123 DBE_PERF.GLOBAL_STATIO_USER_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表OID。
schemaname	name	该表模式名。
relname	name	表名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表缓存命中数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	bigint	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓冲区数（如果存在）。

8.2.10.4 DBE_PERF.STATIO_USER_INDEXES

该视图显示当前节点命名空间中所有用户关系表索引的I/O状态信息，如表8-124所示。

表 8-124 DBE_PERF.STATIO_USER_INDEXES 字段

名称	类型	描述
relid	oid	索引的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	bigint	从索引中读取的磁盘块数。
idx_blks_hit	bigint	索引命中缓存数。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

版权所有 © 华为云计算技术有限公司

8.2.10.5 DBE_PERF.SUMMARY_STATIO_USER_INDEXES

该视图显示集群命名空间中所有用户关系表上索引的I/O状态汇总信息，如表8-125所示。

表 8-125 DBE_PERF.SUMMARY_STATIO_USER_INDEXES 字段

名称	类型	描述
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	numeric	从索引中读取的磁盘块数。
idx_blks_hit	numeric	索引命中缓存数。

8.2.10.6 DBE_PERF.GLOBAL_STATIO_USER_INDEXES

该视图显示各节点的命名空间中所有用户关系表索引的I/O状态信息，如表8-126所示。

表 8-126 DBE_PERF.GLOBAL_STATIO_USER_INDEXES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	索引的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	numeric	从索引中读取的磁盘块数。
idx_blks_hit	numeric	索引命中缓存数。

8.2.10.7 DBE_PERF.STATIO_USER_SEQUENCES

该视图显示当前节点的命名空间中所有用户关系表类型为序列的I/O状态信息，如表8-127所示。

版权所有 © 华为云计算技术有限公司

表 8-127 DBE_PERF.STATIO_USER_SEQUENCE 字段

名称	类型	描述
relid	oid	序列OID。
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列中缓存命中数。

8.2.10.8 DBE_PERF.SUMMARY_STATIO_USER_SEQUENCES

该视图显示集群内汇聚的命名空间中所有用户关系表类型为序列的I/O状态信息，如表8-128所示。

表 8-128 DBE_PERF.SUMMARY_STATIO_USER_SEQUENCES 字段

名称	类型	描述
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	numeric	从序列中读取的磁盘块数。
blks_hit	numeric	序列中缓存命中数。

8.2.10.9 DBE_PERF.GLOBAL_STATIO_USER_SEQUENCES

该视图显示各节点的命名空间中所有用户关系表类型为序列的I/O状态信息，如表8-129所示。

表 8-129 DBE_PERF.GLOBAL_STATIO_USER_SEQUENCES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	序列OID。
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
blks_hit	bigint	序列中缓存命中数。

8.2.10.10 DBE_PERF.STATIO_SYS_TABLES

该视图显示当前节点的命名空间中所有系统表的I/O状态信息，如表8-130所示。

表 8-130 DBE_PERF.STATIO_SYS_TABLES 字段

名称	类型	描述
relid	oid	表OID。
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表缓存命中数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	bigint	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓冲区数（如果存在）。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.10.11 DBE_PERF.SUMMARY_STATIO_SYS_TABLES

该视图显示集群内各节点的系统表的I/O状态汇总信息，如表8-131所示。

版权所有 © 华为云计算技术有限公司

表 8-131 DBE_PERF.SUMMARY_STATIO_SYS_TABLES 字段

名称	类型	描述
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	numeric	从该表中读取的磁盘块数。
heap_blks_hit	numeric	该表缓存命中数。
idx_blks_read	numeric	从表中所有索引读取的磁盘块数。
idx_blks_hit	numeric	表中所有索引命中缓存数。
toast_blks_read	numeric	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	numeric	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	numeric	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	numeric	该表的TOAST表索引命中缓冲区数（如果存在）。

8.2.10.12 DBE_PERF.GLOBAL_STATIO_SYS_TABLES

该视图显示各节点的命名空间中所有系统表的I/O状态信息，如**表8-132**所示。

表 8-132 DBE_PERF.GLOBAL_STATIO_SYS_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表OID。
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表缓存命中数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	该表的TOAST表读取的磁盘块数（如果存在）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
toast_blks_hit	bigint	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	bigint	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓冲区数（如果存在）。

8.2.10.13 DBE_PERF.STATIO_SYS_INDEXES

该视图显示命名空间中所有系统表索引的I/O状态信息，如表8-133所示。

表 8-133 DBE_PERF.STATIO_SYS_INDEXES 字段

名称	类型	描述
relid	oid	索引的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	bigint	从索引中读取的磁盘块数。
idx_blks_hit	bigint	索引命中缓存数。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.10.14 DBE_PERF.SUMMARY_STATIO_SYS_INDEXES

该视图显示集群内命名空间中所有系统表上索引的I/O状态汇总信息，如表8-134所示。

表 8-134 DBE_PERF.SUMMARY_STATIO_SYS_INDEXES 字段

名称	类型	描述
schemaname	name	该索引的模式名。
relname	name	该索引的表名。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
indexrelname	name	索引名称。
idx_blks_read	numeric	从索引中读取的磁盘块数。
idx_blks_hit	numeric	索引命中缓存数。

8.2.10.15 DBE_PERF.GLOBAL_STATIO_SYS_INDEXES

该视图显示各节点的命名空间中所有系统表索引的I/O状态信息，如表8-135所示。

表 8-135 DBE_PERF.GLOBAL_STATIO_SYS_INDEXES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	索引的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	numeric	从索引中读取的磁盘块数。
idx_blks_hit	numeric	索引命中缓存数。

8.2.10.16 DBE_PERF.STATIO_SYS_SEQUENCES

该视图显示命名空间中所有系统序列的I/O状态信息，如表8-136所示。

表 8-136 DBE_PERF.STATIO_SYS_SEQUENCES 字段

名称	类型	描述
relid	oid	序列OID。
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
blks_hit	bigint	序列中缓存命中数。

8.2.10.17 DBE_PERF.SUMMARY_STATIO_SYS_SEQUENCES

该视图显示集群内汇聚的命名空间中所有系统序列的I/O状态信息，如表8-137所示。

表 8-137 DBE_PERF.SUMMARY_STATIO_SYS_SEQUENCES 字段

名称	类型	描述
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	numeric	从序列中读取的磁盘块数。
blks_hit	numeric	序列中缓存命中数。

8.2.10.18 DBE_PERF.GLOBAL_STATIO_SYS_SEQUENCES

该视图显示各节点的命名空间中所有系统序列的I/O状态信息，如表8-138所示。

表 8-138 DBE_PERF.GLOBAL_STATIO_SYS_SEQUENCES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	序列OID。
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列中缓存命中数。

8.2.10.19 DBE_PERF.STATIO_ALL_TABLES

该视图显示数据库当前节点每张表（包括TOAST表）的I/O状态信息，如表8-139所示。

版权所有 © 华为云计算技术有限公司

表 8-139 DBE_PERF.STATIO_ALL_TABLES 字段

名称	类型	描述
relid	oid	表OID。
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表缓存命中数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	bigint	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓冲区数（如果存在）。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.10.20 DBE_PERF.SUMMARY_STATIO_ALL_TABLES

该视图显示集群内各节点每张表（包括TOAST表）的I/O状态信息的汇总结果，如**表 8-140**所示。

表 8-140 DBE_PERF.SUMMARY_STATIO_ALL_TABLES 字段

名称	类型	描述
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	numeric	从该表中读取的磁盘块数。
heap_blks_hit	numeric	该表缓存命中数。
idx_blks_read	numeric	从表中所有索引读取的磁盘块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
idx_blks_hit	numeric	表中所有索引命中缓存数。
toast_blks_read	numeric	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	numeric	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	numeric	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	numeric	该表的TOAST表索引命中缓冲区数（如果存在）。
last_updated	timestamp with time zone	视图中该表监控数据最后一次更新的时间。

8.2.10.21 DBE_PERF.GLOBAL_STATIO_ALL_TABLES

该视图显示集群内各节点每张表（包括TOAST表）的I/O状态信息，如表8-141所示。

表 8-141 DBE_PERF.GLOBAL_STATIO_ALL_TABLES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	表OID。
schemaname	name	该表模式名。
relname	name	表名。
heap_blks_read	bigint	从该表中读取的磁盘块数。
heap_blks_hit	bigint	该表缓存命中数。
idx_blks_read	bigint	从表中所有索引读取的磁盘块数。
idx_blks_hit	bigint	表中所有索引命中缓存数。
toast_blks_read	bigint	该表的TOAST表读取的磁盘块数（如果存在）。
toast_blks_hit	bigint	该表的TOAST表命中缓冲区数（如果存在）。
tidx_blks_read	bigint	该表的TOAST表索引读取的磁盘块数（如果存在）。
tidx_blks_hit	bigint	该表的TOAST表索引命中缓冲区数（如果存在）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
last_updated	timestamp with time zone	视图中该表监控数据最后一次更新的时间。

8.2.10.22 DBE_PERF.STATIO_ALL_INDEXES

该视图包含数据库中的每个索引行，显示特定索引的I/O的统计，如表8-142所示。

表 8-142 DBE_PERF.STATIO_ALL_INDEXES 字段

名称	类型	描述
relid	oid	索引的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	bigint	从索引中读取的磁盘块数。
idx_blks_hit	bigint	索引命中缓存数。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.10.23 DBE_PERF.SUMMARY_STATIO_ALL_INDEXES

该视图包含集群数据库中的每个索引行，显示特定索引I/O的统计汇总结果，如表8-143所示。

表 8-143 DBE_PERF.SUMMARY_STATIO_ALL_INDEXES 字段

名称	类型	描述
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	numeric	从索引中读取的磁盘块数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
idx_blks_hit	numeric	索引命中缓存数。
last_updated	timestamp with time zone	视图中该索引监控数据最后一次更新的时间。

8.2.10.24 DBE_PERF.GLOBAL_STATIO_ALL_INDEXES

该视图包含各节点的数据库中的每个索引行，显示特定索引的I/O的统计，如表8-144所示。

表 8-144 DBE_PERF.GLOBAL_STATIO_ALL_INDEXES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	索引的表的OID。
indexrelid	oid	该索引的OID。
schemaname	name	该索引的模式名。
relname	name	该索引的表名。
indexrelname	name	索引名称。
idx_blks_read	numeric	从索引中读取的磁盘块数。
idx_blks_hit	numeric	索引命中缓存数。
last_updated	timestamp with time zone	视图中该索引监控数据最后一次更新的时间。

8.2.10.25 DBE_PERF.STATIO_ALL_SEQUENCES

该视图包含数据库中每个序列的每一行，显示特定序列关于I/O的统计，如表8-145所示。

表 8-145 DBE_PERF.STATIO_ALL_SEQUENCES 字段

名称	类型	描述
relid	oid	序列OID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列中缓存命中数。

8.2.10.26 DBE_PERF.SUMMARY_STATIO_ALL_SEQUENCES

该视图包含集群内汇聚的数据库中每个序列的每一行，显示特定序列关于I/O的统计，如表8-146所示。

表 8-146 DBE_PERF.SUMMARY_STATIO_ALL_SEQUENCES 字段

名称	类型	描述
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	numeric	从序列中读取的磁盘块数。
blks_hit	numeric	序列中缓存命中数。

8.2.10.27 DBE_PERF.GLOBAL_STATIO_ALL_SEQUENCES

该视图包含各节点的数据库中每个序列的每一行，显示特定序列关于I/O的统计，如表8-147所示。

表 8-147 DBE_PERF.GLOBAL_STATIO_ALL_SEQUENCES 字段

名称	类型	描述
node_name	name	节点名称。
relid	oid	序列OID。
schemaname	name	序列中模式名。
relname	name	序列名。
blks_read	bigint	从序列中读取的磁盘块数。
blks_hit	bigint	序列中缓存命中数。

版权所有 © 华为云计算技术有限公司

8.2.11 Comm

8.2.11.1 DBE_PERF.COMM_DELAY

该视图展示单个DN的TCP代理通信库时延状态，如表8-148所示。

表 8-148 DBE_PERF.COMM_DELAY 字段

名称	类型	描述
node_name	text	节点名称。
remote_name	text	连接对端节点名称。
remote_host	text	连接对端IP地址。
stream_num	integer	当前物理连接使用的stream逻辑连接数量。
min_delay	integer	当前物理连接一分钟内探测到的最小时延（单位：微秒）。 **说明** 负数结果无效，请重新等待时延状态更新后再执行。
average	integer	当前物理连接一分钟内探测时延的平均值（单位：微秒）。
max_delay	integer	当前物理连接一分钟内探测到的最大时延（单位：微秒）。

8.2.11.2 DBE_PERF.GLOBAL_COMM_DELAY

该视图展示所有DN的TCP代理通信库时延状态，如表8-149所示。

表 8-149 DBE_PERF.GLOBAL_COMM_DELAY 字段

名称	类型	描述
node_name	text	节点名称。
remote_name	text	连接对端节点名称。
remote_host	text	连接对端IP地址。
stream_num	integer	当前物理连接使用的stream逻辑连接数量。
min_delay	integer	当前物理连接一分钟内探测到的最小时延（单位：微秒）。 **说明** 负数结果无效，请重新等待时延状态更新后再执行。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
average	integer	当前物理连接一分钟内探测时延的平均值（单位：微秒）。
max_delay	integer	当前物理连接一分钟内探测到的最大时延（单位：微秒）。

8.2.11.3 DBE_PERF.COMM_RECV_STREAM

该视图展示单个DN上所有的TCP代理通信库接收流状态，如表8-150所示。

表 8-150 DBE_PERF.COMM_RECV_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的TCP通信socket。
state	text	通信流当前的状态。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
recv_bytes	bigint	通信流接收的数据总量（单位：Byte）。
time	bigint	通信流当前生命周期使用时长（单位：毫秒）。
speed	bigint	通信流的平均接收速率（单位：Byte/s）。
quota	bigint	通信流当前的通信配额值（单位：Byte）。
buff_usize	bigint	通信流当前缓存的数据大小（单位：Byte）。

8.2.11.4 DBE_PERF.GLOBAL_COMM_RECV_STREAM

该视图展示所有DN上所有的TCP代理通信库接收流状态，如表8-151所示。

版权所有 © 华为云计算技术有限公司

表 8-151 DBE_PERF.GLOBAL_COMM_RECV_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的TCP通信socket。
state	text	通信流当前的状态。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
recv_bytes	bigint	通信流接收的数据总量（单位：Byte）。
time	bigint	通信流当前生命周期使用时长（单位：毫秒）。
speed	bigint	通信流的平均接收速率（单位：Byte/s）。
quota	bigint	通信流当前的通信配额值（单位：Byte）。
buff_usize	bigint	通信流当前缓存的数据大小（单位：Byte）。

8.2.11.5 DBE_PERF.COMM_SEND_STREAM

该视图展示单个DN上所有的TCP代理通信库发送流状态，如表8-152所示。

表 8-152 DBE_PERF.COMM_SEND_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的TCP通信socket。
state	text	通信流当前的状态。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。
recv_smp	integer	通信流所执行查询recv端的smpid编号。
send_bytes	bigint	通信流发送的数据总量（单位：Byte）。
time	bigint	通信流当前生命周期使用时长（单位：ms）。
speed	bigint	通信流的平均发送速率（单位：Byte/s）。
quota	bigint	通信流当前的通信配额值（单位：Byte）。
wait_quota	bigint	通信流等待quota值产生的额外时间开销（单位：毫秒）。

8.2.11.6 DBE_PERF.GLOBAL_COMM_SEND_STREAM

该视图展示所有DN上所有的TCP代理通信库发送流状态，如表8-153所示。

表 8-153 DBE_PERF.GLOBAL_COMM_SEND_STREAM 字段

名称	类型	描述
node_name	text	节点名称。
local_tid	bigint	使用此通信流的线程ID。
remote_name	text	连接对端节点名称。
remote_tid	bigint	连接对端线程ID。
idx	integer	通信对端DN在本DN内的标识编号。
sid	integer	通信流在物理连接中的标识编号。
tcp_sock	integer	通信流所使用的TCP通信socket。
state	text	通信流当前的状态。
query_id	bigint	通信流对应的debug_query_id编号。
pn_id	integer	通信流所执行查询的plan_node_id编号。
send_smp	integer	通信流所执行查询send端的smpid编号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
recv_smp	integer	通信流所执行查询recv端的smpid编号。
send_bytes	bigint	通信流发送的数据总量（单位：Byte）。
time	bigint	通信流当前生命周期使用时长（单位：毫秒）。
speed	bigint	通信流的平均发送速率（单位：Byte/s）。
quota	bigint	通信流当前的通信配额值（单位：Byte）。
wait_quota	bigint	通信流等待quota值产生的额外时间开销（单位：毫秒）。

8.2.11.7 DBE_PERF.COMM_STATUS

该视图展示单个DN的TCP代理通信库状态，如表8-154所示。

表 8-154 DBE_PERF.COMM_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
rxpck_rate	integer	节点通信库接收速率，单位Byte/s。
txpck_rate	integer	节点通信库发送速率，单位Byte/s。
rxkbyte_rate	bigint	bigint节点通信库接收速率，单位KByte/s。
txkbyte_rate	bigint	bigint节点通信库发送速率，单位KByte/s。
buffer	bigint	cmailbox的buffer大小。
memkbyte_libcomm	bigint	libcomm线程通信内存大小，单位Byte。
memkbyte_libpq	bigint	libpq线程通信内存大小，单位Byte。
used_pm	integer	postmaster线程实时使用率。
used_sflow	integer	gs_sender_flow_controller线程实时使用率。
used_rflow	integer	gs_receiver_flow_controller线程实时使用率。
used_rloop	integer	多个gs_receivers_loop线程中高的实时使用率。
stream	integer	当前使用的逻辑连接总数。

8.2.11.8 DBE_PERF.GLOBAL_COMM_STATUS

该视图展示所有DN的TCP代理通信库状态，如表8-155所示。

版权所有 © 华为云计算技术有限公司

表 8-155 DBE_PERF.GLOBAL_COMM_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
rxpck_rate	integer	节点通信库接收速率，单位Byte/s。
txpck_rate	integer	节点通信库发送速率，单位Byte/s。
rxkbyte_rate	bigint	bigint节点通信库接收速率，单位KByte/s。
txkbyte_rate	bigint	bigint节点通信库发送速率，单位KByte/s。
buffer	bigint	cmailbox的buffer大小。
memkbyte_libcomm	bigint	libcomm线程通信内存大小，单位Byte。
memkbyte_libpq	bigint	libpq线程通信内存大小，单位Byte。
used_pm	integer	postmaster线程实时使用率。
used_sflow	integer	gs_sender_flow_controller线程实时使用率。
used_rflow	integer	gs_receiver_flow_controller线程实时使用率。
used_rloop	integer	多个gs_receivers_loop线程中高的实时使用率。
stream	integer	当前使用的逻辑连接总数。

8.2.11.9 DBE_PERF.GLOBAL_COMM_CHECK_CONNECTION_STATUS

该视图用于显示所有CN和所有活跃节点（CN和主DN）的连接情况，权限控制继承DBE_PERF schema，如**表8-156**所示。

表 8-156 DBE_PERF.GLOBAL_COMM_CHECK_CONNECTION_STATUS 字段

名称	类型	描述
node_name	name	实例名称。
remote_name	name	对端实例名称。
remote_host	text	对端实例的IP。
remote_port	bigint	对端实例的PORT。
is_connected	boolean	探测当前实例与对端实例之间的网络情况： ● t（true）：表示实例之间网络正常。 ● f（false）：表示实例之间网络异常。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
no_error_occur	boolean	当前实例与对端实例的pooler建连结果： • t（true）：表示pooler连接正常。 • f（false）：表示pooler连接异常。

8.2.12 Utility

8.2.12.1 DBE_PERF.REPLICATION_STAT

该视图用于描述日志同步状态信息，如发起端发送日志位置、接收端接收日志位置等，如表8-157所示。

表 8-157 DBE_PERF.REPLICATION_STAT 字段

名称	类型	描述
pid	bigint	线程的PID。
usesysid	oid	用户系统ID。
usename	name	用户名。
application_name	text	程序名称。
client_addr	inet	客户端地址。
client_hostname	text	客户端名。
client_port	integer	客户端端口。
backend_start	timestamp with time zone	程序启动时间。
state	text	日志复制的状态： • 追赶状态 • 一致的流状态
sender_sent_location	text	发送端发送日志位置。
receiver_write_location	text	接收端write日志位置。
receiver_flush_location	text	接收端flush日志位置。
receiver_replay_location	text	接收端replay日志位置。
sync_priority	integer	同步复制的优先级（0表示异步）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sync_state	text	同步状态： • 异步复制 • 同步复制 • 潜在同步者
dbid	oid	统计信息所属的数据库id。

8.2.12.2 DBE_PERF.GLOBAL_REPLICATION_STAT

该视图用于获得各节点描述日志同步状态信息，如发起端发送日志位置、接收端接收日志位置等，如表8-158所示。

表 8-158 DBE_PERF.GLOBAL_REPLICATION_STAT 字段

名称	类型	描述
node_name	name	节点名称。
pid	bigint	线程的PID。
usesysid	oid	用户系统ID。
usename	name	用户名。
application_name	text	程序名称。
client_addr	inet	客户端地址。
client_hostname	text	客户端名。
client_port	integer	客户端端口。
backend_start	timestamp with time zone	程序启动时间。
state	text	日志复制的状态： • 追赶状态 • 一致的流状态
sender_sent_location	text	发送端发送日志位置。
receiver_write_location	text	接收端write日志位置。
receiver_flush_location	text	接收端flush日志位置。
receiver_replay_location	text	接收端replay日志位置。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sync_priority	integer	同步复制的优先级（0表示异步）。
sync_state	text	同步状态： • 异步复制 • 同步复制 • 潜在同步者
dbid	oid	统计信息所属的数据库id。

8.2.12.3 DBE_PERF.REPLICATION_SLOTS

该视图用于查看复制槽的信息，如表8-159所示。

表 8-159 DBE_PERF.REPLICATION_SLOTS 字段

名称	类型	描述
slot_name	text	复制槽的名称。
plugin	text	逻辑复制槽对应的输出插件名称。
slot_type	text	复制槽的类型。 • physical：物理复制槽。 • logical：逻辑复制槽。
datoid	oid	复制槽所在的数据库OID。
database	name	复制槽所在的数据库名称。
active	boolean	复制槽是否为激活状态。 • t（true）：表示是。 • f（false）：表示不是。
xmin	xid	数据库需要为复制槽保留的最早事务的事务号。
catalog_xmin	xid	数据库需要为逻辑复制槽保留的最早的涉及系统表的事务的事务号。
restart_lsn	text	复制槽需要的最早Xlog的物理位置。
dummy_standby	boolean	预留参数。 复制槽的连接对端是否为从备。 • t（true）：表示是。 • f（false）：表示不是。

8.2.12.4 DBE_PERF.GLOBAL_REPLICATION_SLOTS

该视图用于查看集群各节点的复制槽的信息，如表8-160所示。

版权所有 © 华为云计算技术有限公司

表 8-160 DBE_PERF.GLOBAL_REPLICATION_SLOTS 字段

名称	类型	描述
node_name	name	节点名称。
slot_name	text	复制槽的名称。
plugin	text	逻辑复制槽对应的输出插件名称。
slot_type	text	复制槽的类型。 ● physical：物理复制槽。 ● logical：逻辑复制槽。
datoid	oid	复制槽所在的数据库OID。
database	name	复制槽所在的数据库名称。
active	boolean	复制槽是否为激活状态。 ● t（true）：表示是。 ● f（false）：表示不是。
x_min	xid	数据库需要为复制槽保留的最早事务的事务号。
catalog_xmin	xid	数据库需要为逻辑复制槽保留的最早的涉及系统表的事务的事务号。
restart_lsn	text	复制槽需要的最早Xlog的物理位置。
dummy_standby	boolean	预留参数。 复制槽的连接对端是否为从备。 ● t（true）：表示是。 ● f（false）：表示不是。

8.2.12.5 DBE_PERF.BGWRITER_STAT

该视图显示关于后端写线程活动的统计信息，如表8-161所示。

表 8-161 DBE_PERF.BGWRITER_STAT 字段

名称	类型	描述
checkpoints_timed	bigint	执行的定期检查点数。
checkpoints_req	bigint	执行的需求检查点数。
checkpoint_write_time	double precision	在检查点处理过程中执行写入操作所花费的总时间，单位为毫秒。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
checkpoint_sync_time	double precision	在检查点处理过程中执行同步（sync）操作所花费的总时间，单位为毫秒。
buffers_checkpoint	bigint	检查点写缓冲区的数量。
buffers_clean	bigint	BGWRITER线程写缓冲区的数量。
maxwritten_clean	bigint	BGWRITER线程停止清理扫描过程的次数（由于单次扫描刷页数量过多导致）。
buffers_backend	bigint	后端直接写缓冲区的数量。
buffers_backend_fsync	bigint	后端线程自行执行fsync的次数（通常情况下fsync调用由BGWRITER线程执行）。
buffers_alloc	bigint	分配的缓冲区数量。
stats_reset	timestamp with time zone	本行统计信息上次被重置的时间。

8.2.12.6 DBE_PERF.GLOBAL_BGWRITER_STAT

该视图显示所有实例关于后端写线程活动的统计信息，如表8-162所示。

表 8-162 DBE_PERF.GLOBAL_BGWRITER_STAT 字段

名称	类型	描述
node_name	name	实例名称。
checkpoints_timed	bigint	执行的定期检查点数。
checkpoints_req	bigint	执行的需求检查点数。
checkpoint_write_time	double precision	在检查点处理过程中执行写入操作所花费的总时间，单位为毫秒。
checkpoint_sync_time	double precision	在检查点处理过程中执行同步（sync）操作所花费的总时间，单位为毫秒。
buffers_checkpoint	bigint	检查点写缓冲区的数量。
buffers_clean	bigint	BGWRITER线程写缓冲区的数量。
maxwritten_clean	bigint	BGWRITER线程停止清理扫描过程的次数（由于单次扫描刷页数量过多导致）。
buffers_backend	bigint	后端直接写缓冲区的数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
buffers_backend_fsync	bigint	后端线程自行执行fsync的次数（通常情况下fsync调用由BGWRITER线程执行）。
buffers_alloc	bigint	分配的缓冲区数量。
stats_reset	timestamp with time zone	本行统计信息上次被重置的时间。

8.2.12.7 DBE_PERF.POOLER_STATUS

该视图用于查询本地CN 的pooler中的缓存连接状态，如表8-163所示。

表 8-163 DBE_PERF.POOLER_STATUS 字段

名称	类型	描述
database	text	数据库名称。
user_name	text	用户名。
tid	bigint	非线程池逻辑下为连接CN的线程id，线程池逻辑下为连接CN的sessionid。
node_oid	bigint	连接的实例节点OID。
node_name	name	连接的实例节点名称。
in_use	boolean	连接是否正被使用： • t（true）：表示连接正在使用。 • f（false）：表示连接没有使用。
local_host	text	本端IP。
local_port	bigint	本端端口号。
remote_host	text	连接的节点IP。
node_port	bigint	连接的节点端口。
fdsock	bigint	端口文件描述符。
remote_pid	bigint	对端处于非线程池逻辑下为对端的线程id，对端处于线程池逻辑下为对端的sessionid。
session_params	text	会话参数。
used_count	bigint	该连接的复用次数。
idx	bigint	连接的实例节点逻辑连接id。
streamid	bigint	每个逻辑连接对应的流标识id。

版权所有 © 华为云计算技术有限公司

8.2.12.8 DBE_PERF.POOLER_AGENT

查询当前连接的CN上Pooler agent的信息，用于辅助连接池问题定位。如果连接DN进行查询，将会返回空表。访问该视图需要系统管理员权限。返回值如表8-164所示。

表 8-164 DBE_PERF.POOLER_AGENT 字段

名称	类型	描述
index	bigint	当前agent在全局agent中的下标。
database_name	text	当前会话的数据库名称。
user_name	text	当前会话的用户名。
session_id	bigint	• 非线程池逻辑下为连接CN的线程id。 • 线程池逻辑下为连接CN的sessionid。
dn_oids	oid[]	记录的DN OID列表。
cn_oids	oid[]	记录的CN OID列表。
agent_init	timestamp with time zone	agent初始化的时间，agent在会话创建或pgxc表信息reload时进行初始化。

8.2.12.9 DBE_PERF.GLOBAL_CKPT_STATUS

该视图用于显示整个集群所有实例的检查点信息和各类日志刷页情况，如表8-165所示。

表 8-165 DBE_PERF.GLOBAL_CKPT_STATUS 字段

名称	类型	描述
node_name	text	实例名称。
ckpt_redo_point	text	当前实例的检查点。
ckpt_clog_flush_num	bigint	从启动到当前时间clog刷盘页面数。
ckpt_csnlog_flush_num	bigint	从启动到当前时间csnlog刷盘页面数。
ckpt_multixact_flush_num	bigint	从启动到当前时间multixact刷盘页面数。
ckpt_predicate_flush_num	bigint	从启动到当前时间predicate刷盘页面数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
ckpt_twophase_flush_num	bigint	从启动到当前时间twophase刷盘页面数。

8.2.12.10 DBE_PERF.GLOBAL_DOUBLE_WRITE_STATUS

该视图显示整个集群所有实例的双写文件的情况，如表8-166所示。

表 8-166 DBE_PERF.GLOBAL_DOUBLE_WRITE_STATUS 字段

名称	类型	描述
node_name	text	实例名称。
curr_dwn	bigint	当前双写文件的序列号。
curr_start_page	bigint	当前双写文件恢复起始页面。
file_trunc_num	bigint	当前双写文件复用的次数。
file_reset_num	bigint	当前双写文件写满后发生重置的次数。
total_writes	bigint	当前双写文件总的I/O次数。
low_threshold_writes	bigint	低效率写双写文件的I/O次数。
high_threshold_writes	bigint	高效率写双写文件的I/O次数。
total_pages	bigint	当前刷页到双写文件区的总的页面个数。
low_threshold_pages	bigint	低效率刷页的页面个数。
high_threshold_pages	bigint	高效率刷页的页面个数。
file_id	bigint	当前双写文件的id号。

8.2.12.11 DBE_PERF.GLOBAL_PAGEWRITER_STATUS

该视图显示整个集群所有实例的刷页信息和检查点信息，如表8-167所示。

表 8-167 DBE_PERF.GLOBAL_PAGEWRITER_STATUS 字段

名称	类型	描述
node_name	text	实例名称。
pgwr_actual_flush_total_num	bigint	从启动到当前时间刷脏页的总计数量。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pgwr_last_flush_num	integer	上一批刷脏页数量。
remain_dirty_page_num	bigint	当前预计剩余的脏页数量。
queue_head_page_rec_lsn	text	当前实例的脏页队列第一个脏页的recovery_lsn。
queue_rec_lsn	text	当前实例的脏页队列的recovery_lsn。
current_xlog_insert_lsn	text	当前实例Xlog写入的位置，备机查询时该值无效。
ckpt_redo_point	text	当前实例的检查点。

8.2.12.12 DBE_PERF.GLOBAL_POOLER_STATUS

该视图用于查询全局CN 的pooler中的缓存连接状态，如表8-168所示。

表 8-168 DBE_PERF.GLOBAL_POOLER_STATUS 字段

名称	类型	描述
source_node_name	name	源节点名称。
database	text	数据库名称。
user_name	text	用户名。
tid	bigint	非线程池逻辑下为连接CN的线程id，线程池逻辑下为连接CN的sessionid。
node_oid	bigint	连接的实例节点OID。
node_name	name	连接的实例节点名称。
in_use	boolean	连接是否正被使用： • t（true）：表示连接正在使用。 • f（false）：表示连接没有使用。
local_host	text	本端IP。
local_port	bigint	本端端口号。
remote_host	text	连接的节点IP。
node_port	bigint	连接的节点端口。
fdsock	bigint	端口文件描述符。
remote_pid	bigint	连接的远端节点线程号。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
session_params	text	会话参数。
used_count	bigint	该连接的复用次数。
idx	bigint	连接的实例节点逻辑连接id。
streamid	bigint	每个逻辑连接对应的流标识id。

8.2.12.13 DBE_PERF.GLOBAL_RECORD_RESET_TIME

该视图用于获取集群中各节点的“重置（重启，主备倒换，数据库删除）时间”的统计信息，如表8-169所示。

表 8-169 DBE_PERF.GLOBAL_RECORD_RESET_TIME 字段

名称	类型	描述
node_name	text	节点名称。
reset_time	timestamp with time zone	重置时间点。

8.2.12.14 DBE_PERF.GLOBAL_REDO_STATUS

该视图显示整个集群所有实例的日志回放情况，如表8-170所示。

表 8-170 DBE_PERF.GLOBAL_REDO_STATUS 字段

名称	类型	描述
node_name	text	实例名称。
redo_start_ptr	bigint	当前实例日志回放的起始点。
redo_start_time	bigint	当前实例日志回放的起始UTC时间。
redo_done_time	bigint	当前实例日志回放的结束UTC时间。
curr_time	bigint	当前实例的当前UTC时间。
min_recovery_point	bigint	当前实例日志完成回放后可对外提供服务的最小一致性点。
read_ptr	bigint	当前实例日志的读取位置。
last_replayed_read_ptr	bigint	当前实例的日志回放位置。
recovery_done_ptr	bigint	当前实例启动完成时的回放位置。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
read_xlog_io_counter	bigint	当前实例读取回放日志的I/O次数计数。
read_xlog_io_total_dur	bigint	当前实例读取回放日志的I/O总用时。
read_data_io_counter	bigint	当前实例日志回放过程中，读取数据页面的I/O次数计数。
read_data_io_total_dur	bigint	当前实例日志回放过程中，读取数据页面的I/O总用时。
write_data_io_counter	bigint	当前实例日志回放过程中，写数据页面的I/O次数计数。
write_data_io_total_dur	bigint	当前实例日志回放过程中，写数据页面的I/O总用时。
process_pending_counter	bigint	当前实例日志回放过程中，日志分发线程的同步次数计数。
process_pending_total_dur	bigint	当前实例日志回放过程中，日志分发线程的同步总用时。
apply_counter	bigint	当前实例日志回放过程中，回放线程的同步次数计数。
apply_total_dur	bigint	当前实例日志回放过程中，回放线程的同步总用时。
speed	bigint	当前实例日志回放速率，每回放256MB日志该值更新一次，单位byte/s。
local_max_ptr	bigint	当前实例启动成功后本地收到的回放日志的最大值。
primary_flush_ptr	bigint	主机落盘日志的位置。
worker_info	text	当前实例回放线程信息，若没有开启并行回放则该值为空。

8.2.12.15 DBE_PERF.GLOBAL_RECOVERY_STATUS

该视图显示关于主机和备机的日志流控信息，如表8-171所示。

表 8-171 DBE_PERF.GLOBAL_RECOVERY_STATUS 字段

名称	类型	描述
node_name	text	节点的名称，包含主机和备机。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
standby_node_name	text	备机节点的名称。
source_ip	text	主机的IP地址。
source_port	integer	主机的端口号。
dest_ip	text	备机的IP地址。
dest_port	integer	备机的端口号。
current_rto	bigint	备机当前的日志流控时间，单位秒。
target_rto	bigint	备机通过GUC参数设置的预期流控时间，单位秒。
current_sleep_time	bigint	为了达到此RTO预期，主机所需要执行的睡眠时间，单位微秒。

8.2.12.16 DBE_PERF.CLASS_VITAL_INFO

该视图用于做WDR时校验相同的表或者索引的Oid是否一致，如表8-172所示。

表 8-172 DBE_PERF.CLASS_VITAL_INFO 字段

名称	类型	描述
relid	oid	表的oid。
schemaname	name	schema名称。
relname	name	表名。
relkind	"char"	表示对象类型，取值范围如下： ● r：表示普通表。 ● t：表示toast表。 ● i：表示索引。 ● G：表示全局二级索引。
last_updated	timestamp with time zone	视图中该对象监控数据最后一次更新的时间。

8.2.12.17 DBE_PERF.USER_LOGIN

该视图用来记录用户登录和退出次数的相关信息，如表8-173所示。

版权所有 © 华为云计算技术有限公司

表 8-173 DBE_PERF.USER_LOGIN 字段

名称	类型	描述
node_name	text	节点名称。
user_name	text	用户名称。
user_id	integer	用户oid（同pg_authid中的oid字段）。
login_counter	bigint	登录次数。
logout_counter	bigint	退出次数。
dbid	oid	用户登录的数据库id。

8.2.12.18 DBE_PERF.SUMMARY_USER_LOGIN

该视图用来记录所有CN节点上用户登录和退出次数的相关信息，如表8-174所示。

表 8-174 DBE_PERF.SUMMARY_USER_LOGIN 字段

名称	类型	描述
node_name	text	节点名称。
user_name	text	用户名称。
user_id	integer	用户oid（同pg_authid中的oid字段）。
login_counter	bigint	登录次数。
logout_counter	bigint	退出次数。
dbid	oid	用户登录的数据库id。

8.2.12.19 DBE_PERF.GLOBAL_GET_BGWRITER_STATUS

该视图显示整个集群所有实例bgwriter线程刷页信息、候选buffer链中页面个数和buffer淘汰信息，如表8-175所示。

表 8-175 DBE_PERF.GLOBAL_GET_BGWRITER_STATUS 字段

名称	类型	描述
node_name	text	实例名称。
bgwr_actual_flush_total_num	bigint	从启动到当前时间bgwriter线程总计刷脏页数量。
bgwr_last_flush_num	integer	bgwriter线程上一批刷脏页数量。
candidate_slots	integer	当前候选buffer链中页面个数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
get_buffer_from_list	bigint	buffer淘汰从候选buffer链中获取页面的次数。
get_buf_clock_sweep	bigint	buffer淘汰从原淘汰方案中获取页面的次数。

8.2.12.20 DBE_PERF.GLOBAL_SINGLE_FLUSH_DW_STATUS

该视图显示整个集群所有实例单页面淘汰双写文件信息，如表8-176所示。显示内容中，/前是第一个版本双写文件刷页情况，/后是第二个版本双写文件刷页情况。这里的第一个版本双写刷页情况指的是对非段页式页面进行单页面淘汰双写，第二个版本双写刷页情况指的是对段页式页面进行单页面淘汰双写。

表 8-176 DBE_PERF.GLOBAL_SINGLE_FLUSH_DW_STATUS 字段

名称	类型	描述
node_name	text	实例名称。
curr_dwn	text	当前双写文件的序列号。
curr_start_page	text	当前双写文件start位置。
total_writes	text	当前双写文件总计写数据页面个数。
file_trunc_num	text	当前双写文件复用的次数。
file_reset_num	text	当前双写文件写满后发生重置的次数。

8.2.12.21 DBE_PERF.GLOBAL_CANDIDATE_STATUS

该视图显示整个数据库所有实例候选buffer个数和buffer淘汰信息，如表8-177所示。

表 8-177 DBE_PERF.GLOBAL_CANDIDATE_STATUS 字段

名称	类型	描述
node_name	text	节点名称。
candidate_slots	integer	当前Normal Buffer Pool候选buffer链中页面个数。
get_buf_from_list	bigint	Normal Buffer Pool，buffer淘汰从候选buffer链中获取页面的次数。
get_buf_clock_sweep	bigint	Normal Buffer Pool，buffer淘汰从原淘汰方案中获取页面的次数。
seg_candidate_slots	integer	当前Segment Buffer Pool候选buffer链中页面个数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
seg_get_buf_from_list	bigint	Segment Buffer Pool，buffer淘汰从候选buffer链中获取页面的次数。
seg_get_buf_clock_sweep	bigint	Segment Buffer Pool，buffer淘汰从原淘汰方案中获取页面的次数。

8.2.12.22 DBE_PERF.PARALLEL_DECODE_STATUS

该视图用于查看当前节点上的复制槽的并行解码信息，如表8-178所示。该视图需在DN上查询，在CN上查询返回空。

表 8-178 DBE_PERF.PARALLEL_DECODE_STATUS 字段

名称	类型	描述
slot_name	text	复制槽的名称。
parallel_decode_num	integer	该复制槽的并行解码线程数。
read_change_queue_length	text	将每个解码线程读取日志队列的当前长度拼接后输出。
decode_change_queue_length	text	将每个解码线程解码结果队列的当前长度拼接后输出。
reader_lsn	text	当前reader线程读取的日志位置。
working_txn_cnt	bigint	当前拼接-发送线程中正在拼接的事务个数。
working_txn_memory	bigint	拼接-发送线程中拼接事务占用总内存（单位：字节）。
decoded_time	timestamp with time zone	该复制槽最后一次读取逻辑日志的时间。

8.2.12.23 DBE_PERF.GLOBAL_PARALLEL_DECODE_STATUS

该视图用于查看集群各主DN节点上的复制槽的并行解码信息，如表8-179所示。该视图需在CN上查询，在DN上查询报错。

表 8-179 DBE_PERF.GLOBAL_PARALLEL_DECODE_STATUS 字段

名称	类型	描述
node_name	name	节点名称。
slot_name	text	复制槽的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
parallel_decode_num	integer	该复制槽的并行解码线程数。
read_change_queue_length	text	将每个解码线程读取日志队列的当前长度拼接后输出。
decode_change_queue_length	text	将每个解码线程解码结果队列的当前长度拼接后输出。
reader_lsn	text	当前reader线程读取的日志位置。
working_txn_cnt	bigint	当前拼接-发送线程中正在拼接的事务个数。
working_txn_memory	bigint	拼接-发送线程中拼接事务占用总内存（单位：字节）。
decoded_time	timestamp with time zone	该复制槽最后一次读取逻辑日志的时间。

8.2.12.24 DBE_PERF.PARALLEL_DECODE_THREAD_INFO

该视图用于查看当前节点上进行并行解码的线程信息，如表8-180所示。该视图需在DN上查询，在CN上查询返回空。

表 8-180 DBE_PERF.PARALLEL_DECODE_THREAD_INFO 字段

名称	类型	描述
thread_id	bigint	线程id。
slot_name	text	复制槽名。
thread_type	text	线程种类（共三种，sender代表发送线程，reader代表读取线程，decoder代表解码线程）。
seq_number	integer	当前复制槽中同种线程的序号（从1开始）。

8.2.12.25 DBE_PERF.GLOBAL_PARALLEL_DECODE_THREAD_INFO

该视图用于查看集群各主DN节点上的进行并行解码的线程信息，如表8-181所示。该视图需在CN上查询，在DN上查询报错。

表 8-181 DBE_PERF.GLOBAL_PARALLEL_DECODE_THREAD_INFO 字段

名称	类型	描述
node_name	name	节点名称。
thread_id	bigint	线程id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
slot_name	text	复制槽名。
thread_type	text	线程种类（共三种，sender代表发送线程，reader代表读取线程，decoder代表解码线程）。
seq_number	integer	当前复制槽中同种线程的序号（从1开始）。

8.2.12.26 DBE_PERF.GLOBAL_ADIO_COMPLETER_STATUS

该视图显示集群所有实例中AIO Completer线程的相关统计信息，如表8-182所示。该视图需在CN上查询，在DN上查询报错。

表 8-182 DBE_PERF.GLOBAL_ADIO_COMPLETER_STATUS 字段

参数	类型	描述
node_name	text	实例名称。
tid	bigint	AIO Completer线程id。
thread_type	text	AIO Completer线程类型（读/写）。
aio_submitted_num	bigint	AIO Completer线程已提交的异步I/O请求数量。
aio_completed_num	bigint	AIO Completer线程已完成的异步I/O请求数量。
aio_incompleted_num	bigint	AIO Completer线程未完成的异步I/O请求数量。
slot_count_left	bigint	空闲slot数量。

8.2.12.27 DBE_PERF.GLOBAL_AIO_SLOT_USAGE_STATUS

该视图显示集群所有实例中异步I/O提交槽位的相关统计信息，如表8-183所示。该视图需在CN上查询，在DN上查询报错。

表 8-183 DBE_PERF.GLOBAL_AIO_SLOT_USAGE_STATUS 字段

参数	类型	描述
node_name	text	实例名称。
slot_id	integer	槽位id。
slot_type	"char"	槽位类型（r：读，w：写）。
status	boolean	槽位占用状态。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
buffer_id	bigint	该槽位对应的buffer id。
relfilenode_blocknum	text	该槽位对应的buffer所在的物理页面位置。
lsn	bigint	页面对应的LSN。
submitted_time	bigint	页面异步提交时的时间。
elapsed_time	bigint	页面已经等待的时间。

8.2.12.28 DBE_PERF.GLOBAL_XLOG_REDO_STAT

该视图显示集群各备DN节点已经回放的各类型日志的统计信息，如表8-184所示。

表 8-184 DBE_PERF.GLOBAL_XLOG_REDO_STAT 字段

参数	类型	描述
node_name	text	实例名称。
xlog_type	text	日志类型。
rmid	xid	资源管理器id。
num	xid	已回放数量。
extra	xid	针对page回放日志和xact日志，存在不同的有效值： • 如果是page页回放日志类型，则表示从磁盘读取page的个数。 • 如果是xact日志类型，则表示删除文件的个数。

8.2.12.29 DBE_PERF.EMERGENCY_OPERATION_RECORDS

该视图用于查看各CN/DN节点对Unix Domain Socket运维通道的使用情况，如表8-185所示。

表 8-185 DBE_PERF.EMERGENCY_OPERATION_RECORDS 字段

参数	类型	描述
record_time	timestamptz	使用运维通道的时间。
action	text	使用运维通道执行的操作。
source	text	接入运维通道的来源，取值为gsql和cm。
detail	text	执行操作的具体结果。

版权所有 © 华为云计算技术有限公司

8.2.12.30 DBE_PERF.KERNEL_INFO

该视图用于查看各CN/DN节点上对应模块关键数据信息，如表8-186所示。

表 8-186 DBE_PERF.KERNEL_INFO 字段

参数	类型	描述
node_name	text	节点名。
module	text	模块名。 • XACT（事务模块） • STANDBY（备机模块） • UNDO（undo模块） • HOTPATH（热补丁模块） • LOCK（锁模块） • FULL_SQL • SQL

版权所有 © 华为云计算技术有限公司

参数	类型	描述
name	text	探查内存态关键数据名字。 ● startup_max_xid（XACT：线程启动时最大的xid值） ● recent_local_xmin（XACT：本地活跃事务最小xid值） ● recent_global_xmin（XACT：全局活跃事务最小xid值） ● standby_xmin（STANDBY：备机活跃事务最小xid值） ● standby_redo_cleanup_xmin（STANDBY：备机redo时cleanup日志最小xid值） ● standby_redo_cleanup_xmin_lsn（STANDBY：备机redo时cleanup日志最小xid的LSN值） ● local_csn_min（XACT：本地活跃事务最小CSN值） ● replication_slot_xmin（XACT：复制槽的最小xid值） ● replication_slot_catalog_xmin（XACT：catalog复制槽最小xid值） ● global_recycle_xid（UNDO：全局undo回收事务的最小xid值） ● global_frozen_xid（XACT：全局冻结事务的最小的xid值） ● recent_xmin（XACT：当前会话活跃事务最小xid值） ● next_csn（XACT：下一个事务的CSN值） ● hotpatch_additional_info（HOTPATH：热补丁预留字段） ● stmt_session_discard_records（FULL SQL：内核支持全量SQL因slot满丢弃数据量） ● stmt_shm_flush_discard_records（FULL SQL：内核支持全量SQL因ringbuf满丢弃数据量） ● idle_in_trans_timeout_records（XACT：内核支持空闲事务超时次数累计数据量） ● total_lock_usage（LOCK：当前通过主锁表申请锁的数量） ● total_lock_usage_rate（LOCK：主锁表空间的使用率） ● fastpath_strong_relation_locks_count（LOCK：存在FASTPATH强锁冲突的分区及该分区对应强锁数量）

版权所有 © 华为云计算技术有限公司

参数	类型	描述
		• fastpath_failure_rate（LOCK：由于对应分区存在强锁冲突导致无法通过FASTPATH申请锁的比例） • lock_acquire_lack_fastpath_count（LOCK：由于FASTPATH槽位占满导致通过FASTPATH申请锁失败的总次数） • n_soft_parse（SQL：软解析次数） • n_hard_parse（SQL：硬解析次数）
value	text	探查内存态关键数据值。

8.2.13 Lock

8.2.13.1 DBE_PERF.LOCKS

该视图用于查看各打开事务所持有的锁信息，如表8-187所示。

表 8-187 DBE_PERF.LOCKS 字段

名称	类型	描述
locktype	text	被锁定对象的类型：relation、extend、page、tuple、transactionid、virtualxid、object、userlock、advisory或tablespace。
database	oid	被锁定对象所在数据库的OID： • 如果被锁定的对象是共享对象，则OID为0。 • 如果是一个事务ID，则为NULL。
relation	oid	关系的OID，如果锁定的对象不是关系，也不是关系的一部分，则为NULL。
page	integer	关系内部的页面编号，如果对象不是关系页或者不是行页，则为NULL。
tuple	smallint	页面里边的行编号，如果对象不是行，则为NULL。
bucket	integer	哈希桶号。
virtualxid	text	事务的虚拟ID，如果对象不是一个虚拟事务ID，则为NULL。
transactionid	xid	事务的ID，如果对象不是一个事务ID，则为NULL。
classid	oid	包含该对象的系统表的OID，如果对象不是普通的数据库对象，则为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
objid	oid	对象在其系统表内的OID，如果对象不是普通数据库对象，则为NULL。
objsubid	smallint	对于表的一个字段，这是字段编号；对于其他对象类型，这个字段是零；如果这个对象不是普通数据库对象，则为NULL。
virtualtransaction	text	持有此锁或者在等待此锁的事务的虚拟ID。
pid	bigint	持有或者等待这个锁的服务器线程的逻辑ID。如果锁是被一个预备事务持有的，则为NULL。
sessionid	bigint	持有或者等待这个锁的会话ID。如果锁是被一个预备事务持有的，则为NULL。
mode	text	这个线程持有的或者是期望的锁模式。
granted	boolean	• 如果锁是持有锁，则为TRUE。 • 如果锁是等待锁，则为FALSE。
fastpath	boolean	如果通过fast-path获得锁，则为TRUE；如果通过主要的锁表获得，则为FALSE。
locktag	text	会话等待锁信息，可通过locktag_decode()函数解析。
global_sessionid	text	全局会话ID。

8.2.13.2 DBE_PERF.GLOBAL_LOCKS

该视图用于查看各节点各打开事务所持有的锁信息，如表8-188所示。

表 8-188 DBE_PERF.GLOBAL_LOCKS 字段

名称	类型	描述
node_name	name	节点名称。
locktype	text	被锁定对象的类型：relation、extend、page、tuple、transactionid、virtualxid、object、userlock、advisory或tablespace。
database	oid	被锁定对象所在数据库的OID： • 如果被锁定的对象是共享对象，则OID为0。 • 如果是一个事务ID，则为NULL。
relation	oid	关系的OID，如果锁定的对象不是关系，也不是关系的一部分，则为NULL。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
page	integer	关系内部的页面编号，如果对象不是关系页或者不是行页，则为NULL。
tuple	smallint	页面里边的行编号，如果对象不是行，则为NULL。
bucket	integer	哈希桶号。
virtualxid	text	事务的虚拟ID，如果对象不是一个虚拟事务ID，则为NULL。
transactionid	xid	事务的ID，如果对象不是一个事务ID，则为NULL。
classid	oid	包含该对象的系统表的OID，如果对象不是普通的数据库对象，则为NULL。
objid	oid	对象在其系统表内的OID，如果对象不是普通数据库对象，则为NULL。
objsubid	smallint	对于表的一个字段，这是字段编号；对于其他对象类型，这个字段是零；如果这个对象不是普通数据库对象，则为NULL。
virtualtransaction	text	持有此锁或者在等待此锁的事务的虚拟ID。
pid	bigint	持有或者等待这个锁的服务器线程的逻辑ID。如果锁是被一个预备事务持有的，则为NULL。
sessionid	bigint	持有或者等待这个锁的会话ID。如果锁是被一个预备事务持有的，则为NULL。
global_sessionid	text	全局会话ID。
mode	text	这个线程持有的或者是期望的锁模式。
granted	boolean	● 如果锁是持有锁，则为TRUE。 ● 如果锁是等待锁，则为FALSE。
fastpath	boolean	如果通过fast-path获得锁，则为TRUE；如果通过主要的锁表获得，则为FALSE。
locktag	text	会话等待锁信息，可通过locktag_decode()函数解析。

8.2.14 Wait Events

版权所有 © 华为云计算技术有限公司

8.2.14.1 DBE_PERF.WAIT_EVENTS

该视图显示当前节点wait event的相关统计信息，如表8-189所示。内核中关键的事件信息请参见表8-192。或从视图wait_event_info中查看系统中所有的事件列表。关于每种事务锁对业务的影响程度，请参见LOCK语法小节的详细描述。

对于此系统表查询有如下约束：

此系统表受track_activities和enable_instr_track_wait控制，若关闭参数，则系统中不再记录等待事件，此表中不再能够查询到数据。

表 8-189 DBE_PERF.WAIT_EVENTS 字段

名称	类型	描述
nodename	text	数据库节点名称。
type	text	event类型。
event	text	event名称。
wait	bigint	等待次数。
failed_wait	bigint	等待锁失败的次数。
total_wait_time	bigint	总等待时间（单位：微秒）。
avg_wait_time	bigint	平均等待时间（单位：微秒）。
max_wait_time	bigint	最大等待时间（单位：微秒）。
min_wait_time	bigint	最小等待时间（单位：微秒）。
last_updated	timestamp with time zone	最后一次更新该事件的时间。

8.2.14.2 DBE_PERF.GLOBAL_WAIT_EVENTS

该视图显示各节点wait event的相关统计信息，如表8-190所示。

表 8-190 DBE_PERF.GLOBAL_WAIT_EVENTS 字段

名称	类型	描述
nodename	text	数据库节点名称。
type	text	event类型。
event	text	event名称。
wait	bigint	等待次数。
failed_wait	bigint	等待锁失败的次数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
total_wait_time	bigint	总等待时间（单位：微秒）。
avg_wait_time	bigint	平均等待时间（单位：微秒）。
max_wait_time	bigint	最大等待时间（单位：微秒）。
min_wait_time	bigint	最小等待时间（单位：微秒）。
last_updated	timestamp with time zone	最后一次更新该事件的时间。

8.2.14.3 DBE_PERF.WAIT_EVENT_INFO

该视图显示wait event的具体信息，如表8-191、表8-192所示。

表 8-191 DBE_PERF.WAIT_EVENT_INFO 字段

名称	类型	描述
module	text	event所属的模块名。
type	text	event类型。
event	text	event名称。

表 8-192 Wait Events 事件信息列表

模块分类	事件分类	事件	说明
Lock	等待事件	acquire lock	等待加锁，加锁成功或加锁等待超时。
SharedMemory	LWLOCK事件	ShmemIndexLock	用于保护共享内存中的主索引哈希表
Shared buffer	LWLOCK事件	BufMappingLock	用于保护对共享缓冲映射表的操作。
Lmgr	LWLOCK事件	LockMgrLock	用于保护常规锁结构信息。
LWLock	等待事件	acquire lwlock	等待获取轻量级锁。
I/O	等待事件	wait io	等待I/O完成。
COMM	等待事件	wait cmd	等待完成读取网络通信包。
COMM	等待事件	wait pooler get conn	等待pooler完成获取连接。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
COMM	等待事件	wait pooler abort conn	等待pooler完成终止连接。
COMM	等待事件	wait pooler clean conn	等待pooler完成清理连接。
COMM	等待事件	get conn	获取到其他节点的连接。
COMM	等待事件	set cmd	在连接上执行SET/RESET/TRANSACTION BLOCK LEVEL。
COMM	等待事件	cancel query	取消某连接上正在执行的SQL语句。
COMM	等待事件	stop query	停止某连接上正在执行的查询。
COMM	等待事件	wait node	等待接收与某节点的连接上的数据。
COMM	等待事件	flush data	等待向网络中的其他节点发送数据。
COMM	等待事件	stream get conn	初始化stream flow时，等待建立到consumer nodes的连接。
COMM	等待事件	wait producer ready	初始化stream flow时，等待每个producer都准备好。
Stream	等待事件	synchronize quit	stream plan结束时，等待stream线程组内的线程统一退出。当wait_status中显示形如“synchronize quit, total *m*, remain *n*”的内容时，表示共有m个并发任务，其中仍存在n个并发任务未处于统一退出状态。
Stream	等待事件	wait stream group destroy	stream plan结束时，等待销毁stream node group。
Stream	等待事件	wait stream thread ready	stream producer阻塞，直至其线程id被成功设置。
Stream	等待事件	wait create stream thread	由于资源上限等原因，等待stream线程启动。
Transaction	等待事件	wait transaction sync	等待事务同步。
Transaction	等待事件	wait data sync	等待完成数据页到备机的同步。
Transaction	等待事件	wait data sync queue	等待把行存的数据页放入同步队列。
Transaction	LWLOCK事件	OidGenLock	用于避免不同线程产生相同的OID。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
Transaction	LWLOCK事件	XidGenLock	用于避免两个事务获得相同的xid。
Transaction	LWLOCK事件	ProcArrayLock	用于避免并发访问或修改ProcArray共享数组。
Transaction	LWLOCK事件	SubtransControlLock	用于避免并发访问或者修改子事务控制数据结构。
Transaction	LWLOCK事件	MultiXactGenLock	用于串行分配唯一MultiXactid。
Transaction	LWLOCK事件	TwoPhaseStateLock	用于避免并发访问或者修改两阶段信息共享数组。
Transaction	LWLOCK事件	SerializableXactHashLock	用于避免对于可串行事务共享结构的写写并发和读写并发。
Transaction	LWLOCK事件	SerializableFinishedListLock	用于避免对于已完成可串行事务共享链表的写写并发和读写并发。
Transaction	LWLOCK事件	SerializablePredicateLockListLock	用于保护对于可串行事务持有的锁链表。
Transaction	LWLOCK事件	PredicateLockMgrLock	用于保护可串行事务锁结构信息。
Transaction	LWLOCK事件	OldSerXid SLRU lwlock	保护old xids的slru buffer。
Transaction	LWLOCK事件	OldSerXidLock	用于保护记录冲突可串行事务的结构。
Transaction	LOCK事件	transactionid	对事务ID加锁。
Transaction	LOCK事件	virtualxid	对虚拟事务ID加锁。
Checkpoint	LWLOCK事件	CheckpointLock	用于避免多个checkpoint并发执行。
Checkpoint	LWLOCK事件	CheckpointerCommLock	用于向checkpointer发起文件刷盘请求场景，需要串行的向请求队列插入请求结构。
Analyze	LWLOCK事件	AutoanalyzeLock	用于获取和释放允许执行Autoanalyze的任务资源。
Vacuum	LWLOCK事件	BtreeVacuumLock	用于防止vacuum清理btree中还在使用的页面。
Vacuum	LWLOCK事件	AutovacuumLock	用于串行化访问autovacuum worker数组。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
Vacuum	LWLOCK事件	AutovacuumScheduleLock	用于串行化分配需要vacuum的table。
Auto vacuum	LWLOCK事件	AutovacuumLock	用于保护autovacuum shmem struct。
Auto vacuum	LWLOCK事件	AutovacuumScheduleLock	用于保护auto vacuum worker信息。
Auto analyze	LWLOCK事件	AutoanalyzeLock	用于保护autoAnalyzeFreeProcess变量，保证最多有10个auto analyze线程同时运行。
WAL	等待事件	wait wal sync	等待特定LSN的wal log完成到备机的同步。
WAL	I/O事件	WALBootstrapSync	将初始化的WAL文件持久化到磁盘。在数据库初始化发生。
WAL	I/O事件	WALBootstrapWrite	写入初始化的WAL文件。在数据库初始化发生。
WAL	I/O事件	WALCopyRead	读取已存在的WAL文件并进行复制时产生的读操作。在执行归档恢复完后发生。
WAL	I/O事件	WALCopySync	将复制的WAL文件持久化到磁盘。在执行归档恢复完后发生。
WAL	I/O事件	WALCopyWrite	读取已存在WAL文件并进行复制时产生的写操作。在执行归档恢复完后发生。
WAL	I/O事件	WALInitSync	将新初始化的WAL文件持久化磁盘。在日志回收或写日志时发生。
WAL	I/O事件	WALInitWrite	将新创建的WAL文件初始化为0。在日志回收或写日志时发生。
WAL	I/O事件	WALRead	从Xlog日志读取数据。两阶段文件redo相关的操作产生。
WAL	I/O事件	WALSyncMethodAssign	将当前打开的所有WAL文件持久化到磁盘。
WAL	I/O事件	WALWrite	写入WAL文件。
WAL	I/O事件	LOGCTRL_SLEEP	统计流控发生了多少次、日志流控的睡眠时间。
WAL	LWLOCK事件	RcvWriteLock	防止并发调用WalDataRcvWrite。
WAL	LWLOCK事件	WALBufMappingLock	初始化Xlogbufer的下一个page时需要加x锁。
WAL	LWLOCK事件	WALInsertLock	用于防止多个程序同时向同一个Xlog缓冲区写。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
WAL	LWLOCK事件	WALWriteLock	用于避免并发刷写wal日志。
Relation	LWLOCK事件	SInvalReadLock	用于避免与清理失效消息并发执行。
Relation	LWLOCK事件	SinvalWriteLock	用于避免与其它写失效消息、清理失效消息并发执行。
Relation	LWLOCK事件	RelCacheInitLock	用于失效消息场景对init文件进行操作时加锁。
Relation	LWLOCK事件	TablespaceCreateLock	用于确定tablespace是否已经存在。
Relation	LWLOCK事件	RelfilenodeReuseLock	避免错误地取消已重用的列属性文件的链接。
Relation	LOCK事件	relation	对表加锁。
Relation	LOCK事件	extend	对表扩展空间时加锁。
Relation	LOCK事件	partition	对分区表加锁。
Relation	LOCK事件	partition_seq	对分区表的分区加锁。
WLM	等待事件	wait active statement	WLM等待active statement。
WLM	等待事件	wait memory	WLM等待空余内存。
DDL/DCL	等待事件	create index	等待index创建成功。
DDL/DCL	等待事件	analyze	等待analyze完成。
DDL/DCL	等待事件	vacuum	等待vacuum完成。
DDL/DCL	LWLOCK事件	DelayDDLLock	防止并发ddl。
DDL/DCL	等待事件	vacuum full	等待vacuum full操作完成。
Executor	等待事件	Sort	等待tuple排序完成。
Executor	等待事件	Sort - write file	归并排序时内存受限，将排序完的数据暂时写入文件。
Executor	等待事件	Material	等待将tuple物化。
Executor	等待事件	Material - write file	等待将物化的tuple写入文件。

 版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
Executor	等待事件	HashJoin - build hash	执行hashJoin时，等待hashtable建立完成。
Executor	等待事件	HashJoin - write file	执行hashJoin时，等待将tuple的hash结果写到磁盘。
Executor	等待事件	HashAgg - build hash	执行hashagg时，等待hashtable建立完成。
Executor	等待事件	HashAgg - write file	执行hashagg时，等待将tuple的hash结果写到磁盘。
Executor	等待事件	HashSetop - build hash	通过hash方式执行OP操作时，等待建立hash表。
Executor	等待事件	HashSetop - write file	通过hash方式执行OP操作时，等待将tuple的hash结果写到磁盘。
Executor	等待事件	wait sync consumer next step	等待stream consumer进行下一步处理。
Executor	等待事件	wait sync producer next step	等待stream producer进行下一步处理。
GTM	等待事件	gtm connect	等待与gtm连接。
GTM	等待事件	gtm reset xmin	等待gtm重新设置xmin完成。
GTM	等待事件	gtm get xmin	等待从gtm获取xmin。
GTM	等待事件	gtm get gxid	启动事务时等待从gtm获取gxid。
GTM	等待事件	gtm get csn	事务提交时等待从gtm获取CSN。
GTM	等待事件	gtm get snapshot	事务启动时等待从gtm获取快照。
GTM	等待事件	gtm begin trans	等待GTM开启事务。
GTM	等待事件	gtm commit trans	等待GTM提交事务。
GTM	等待事件	gtm rollback trans	等待GTM回滚事务。
GTM	等待事件	gtm start preprare trans	两阶段提交时，等待GTM完成一阶段。
GTM	等待事件	gtm prepare trans	两阶段提交时，等待GTM完成第二阶段。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
GTM	等待事件	gtm open sequence	等待gtm创建sequence。
GTM	等待事件	gtm close sequence	等待gtm完成alter sequence操作。
GTM	等待事件	gtm set sequence val	等待gtm设置sequence的值。
GTM	等待事件	gtm drop sequence	等待gtm删除sequence。
GTM	等待事件	gtm rename sequence	等待gtm重命名sequence。
GTM	LWLOCK事件	GTMHostInfoLock	用于保护gtm的信息。
Temp File	I/O事件	BufFileRead	从临时文件中读取数据到指定buffer。
Temp File	I/O事件	BufFileWrite	向临时文件中写入指定buffer中的内容。
Pg_control	I/O事件	ControlFileRead	读取pg_control文件。主要在数据库启动、执行checkpoint和主备校验过程中发生。
Pg_control	I/O事件	ControlFileSync	将pg_control文件持久化到磁盘。数据库初始化时发生。
Pg_control	I/O事件	ControlFileSyncUpdate	将pg_control文件持久化到磁盘。主要在数据库启动、执行checkpoint和主备校验过程中发生。
Pg_control	I/O事件	ControlFileWrite	写入pg_control文件。数据库初始化时发生。
Pg_control	I/O事件	ControlFileWriteUpdate	更新pg_control文件。主要在数据库启动、执行checkpoint和主备校验过程中发生。
Pg_control	LWLOCK事件	ControlFileLock	用于避免pg_control文件的读写并发、写写并发。
文件操作	I/O事件	CopyFileRead	copy文件时读取文件内容。
文件操作	I/O事件	CopyFileWrite	copy文件时写入文件内容。
文件操作	I/O事件	DataFileExtend	扩展文件时向文件写入内容。
表数据文件	I/O事件	DataFileImmediateSync	将表数据文件立即持久化到磁盘。
表数据文件	I/O事件	DataFilePrefetch	异步读取表数据文件。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
表数据文件	I/O事件	DataFileRead	同步读取表数据文件。
表数据文件	I/O事件	DataFileSync	将表数据文件持久化磁盘。
表数据文件	I/O事件	DataFileTruncate	表数据文件truncate。
表数据文件	I/O事件	DataFileWrite	向表数据文件写入内容。
表数据文件	LWLOCK事件	SyncScanLock	用于确定heap扫描时某个relfilenode的起始位置。
表数据文件	LWLOCK事件	RelationMappingLock	用于等待更新系统表到存储位置之间映射的文件。
元数据	LWLOCK事件	MetaCacheSweepLock	用于元数据循环淘汰。
postmaster.pid	I/O事件	LockFileAddToDataDirRead	读取"postmaster.pid"文件。
postmaster.pid	I/O事件	LockFileAddToDataDirSync	将"postmaster.pid"内容持久化到磁盘。
postmaster.pid	I/O事件	LockFileAddToDataDirWrite	将pid信息写到"postmaster.pid"文件。
Pid File	I/O事件	LockFileCreateRead	读取LockFile文件"%s.lock"。
Pid File	I/O事件	LockFileCreateSync	将LockFile文件"%s.lock"内容持久化到磁盘。
Pid File	I/O事件	LockFileCreateWRITE	将pid信息写到LockFile文件"%s.lock"。
系统表映射文件	I/O事件	RelationMapRead	读取系统表到存储位置之间的映射文件。
系统表映射文件	I/O事件	RelationMapSync	将系统表到存储位置之间的映射文件持久化到磁盘。
系统表映射文件	I/O事件	RelationMapWrite	写入系统表到存储位置之间的映射文件。
Streaming replication	I/O事件	ReplicationSlotRead	读取流复制槽文件。重新启动时发生。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
Streaming replication	I/O事件	ReplicationSlotRestoreSync	将流复制槽文件持久化到文件。
Streaming replication	I/O事件	ReplicationSlotSync	checkpoint时将流复制槽临时文件持久化到磁盘。
Streaming replication	I/O事件	ReplicationSlotWrite	checkpoint时写流复制槽临时文件。
Streaming replication	LWLOCK事件	ReplicationSlotAllocationLock	方式分配同一个复制槽。
Streaming replication	LWLOCK事件	ReplicationSlotControlLock	检测复制槽名称冲突，识别出具体可分配的复制槽，需要ReplicationSlotControlLock锁。
Clog	I/O事件	SLRUFlushSync	将pg_clog文件持久化到磁盘。主要在执行checkpoint和数据库停机时发生。
Clog	I/O事件	SLRURead	读取pg_clog文件。
Clog	I/O事件	SLRUSync	将脏页写入文件pg_clog并持久化到磁盘。主要在执行checkpoint和数据库停机时发生。
Clog	I/O事件	SLRUWrite	写入pg_clog文件。
Clog	LWLOCK事件	CLogControlLock	用于避免并发访问或者修改Clog控制数据结构。
Clog	LWLOCK事件	MultiXactOffsetControlLock	用于避免对pg_multixact/offset的写写并发和读写并发。
Clog	LWLOCK事件	MultiXactMemberControlLock	用于避免对pg_multixact/members的写写并发和读写并发。
timelinehistory	I/O事件	TimelineHistoryRead	读取timelinehistory文件。在数据库启动时发生。
timelinehistory	I/O事件	TimelineHistorySync	将timelinehistory文件持久化到磁盘。在数据库启动时发生。
timelinehistory	I/O事件	TimelineHistoryWrite	写入timelinehistory文件。
pg_twophase	I/O事件	TwophaseFileRead	读取pg_twophase文件。在两阶段事务提交、两阶段事务恢复时发生。
pg_twophase	I/O事件	TwophaseFileSync	将pg_twophase文件持久化到磁盘。在两阶段事务提交、两阶段事务恢复时发生。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
pg_twophase	I/O事件	TwophaseFileWrite	写入pg_twophase文件。在两阶段事务提交、两阶段事务恢复时发生。
Cluster	LWLOCK事件	NodeTableLock	用于保护存放CN和DN节点信息的共享结构。
Concurrency	LWLOCK事件	PoolerLock	用于保证两个线程不会同时从连接池里取到相同的连接。
Concurrency	LWLOCK事件	AsyncCtlLock	用于避免并发访问或者修改共享通知状态。
Concurrency	LWLOCK事件	AsyncQueueLock	用于避免并发访问或者修改共享通知信息队列。
Double write	I/O事件	DoubleWriteFileWrite	Double write过程中将页面写到Double write文件中。
Double write	I/O事件	DoubleWriteFileRead	发生半写进行恢复时读取Double write文件。
Statistics file	LWLOCK事件	FileStatLock	用于保护存储统计文件信息的数据结构。
Master-slave replication	LWLOCK事件	SyncRepLock	用于在主备复制时保护Xlog同步信息。
Master-slave replication	LWLOCK事件	ReplicationSlotAllocationLock	用于主备复制时保护主机端的流复制槽的分配。
Master-slave replication	LWLOCK事件	ReplicationSlotControlLock	用于主备复制时避免并发更新流复制槽状态。
Master-slave replication	LWLOCK事件	LsnXlogChkFileLock	用于串行更新特定结构中记录的主备机的xlogflush位置点。
Master-slave replication	LWLOCK事件	DataSyncRepLock	用于在主备复制时保护数据页同步信息。
Speed up the cluster	LWLOCK事件	SearchServerLibLock	用于GPU加速场景初始化加载特定动态库时，对读文件操作进行加锁。
Resource manage	LWLOCK事件	ResourcePoolHashLock	用于避免并发访问或者修改资源池哈希表。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
OBS	LWLOCK事件	OBSGetPathLock	用于避免对obs路径的写写并发和读写并发。
OBS	LWLOCK事件	OBSRuntimeLock	用于获取环境变量，如GAUSSHOME。
LLVM	LWLOCK事件	LLVMDumpIRLock	用于导出动态生成函数所对应的汇编语言。
LLVM	LWLOCK事件	LLVMParseIRLock	用于在查询开始处从IR文件中编译并解析已写好的IR函数。
MPP is compatible with ORACLE scheduled task function	LWLOCK事件	JobShmemLock	用于MPP兼容ORACLE定时任务功能中保护定时读取的全局变量。
Operator history information statistics	LWLOCK事件	OperatorRealTLock	用于避免并发访问或者修改记录算子级实时数据的全局结构。
Operator history information statistics	LWLOCK事件	OperatorHistLock	用于避免并发访问或者修改记录算子级历史数据的全局结构。
query history information statistics	LWLOCK事件	SessionRealTLock	用于避免并发访问或者修改记录query级实时数据的全局结构。
query history information statistics	LWLOCK事件	SessionHistLock	用于避免并发访问或者修改记录query级历史数据的全局结构。
query history information statistics	LWLOCK事件	WaitCountHashLock	用于保护用户语句计数功能场景中的共享结构。
barrier	LWLOCK事件	BarrierLock	用于保证当前只有一个线程在创建Barrier。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
CSN	LWLOCK事件	CSNBufMappingLock	保护csn页面。
instrumentation	LWLOCK事件	UniqueSQLMappingLock	用于保护uniquesql hash table。
instrumentation	LWLOCK事件	InstrUserLock	用于保护InstrUserHTAB。
instrumentation	LWLOCK事件	PercentileLock	用于保护全局PercentileBuffer。
instrumentation	LWLOCK事件	InstrWorkloadLock	保护全局instr workload xact info hash table。
Pgproc	LWLOCK事件	Pgproc lwlock	用于保护pgproc。
Async buffer	LWLOCK事件	AsyncCtlLock	保护Async buffer。
MultiXact	LWLOCK事件	MultiXactOffset lwlock	保护MultiXact offset的slru buffer。
MultiXact	LWLOCK事件	MultiXactMemberlwlock	保护MultiXact member的slrubuffer。
CBM	LWLOCK事件	CBMParseXlogLock	Cbm 解析Xlog时的保护锁。
BadBlock	LWLOCK事件	BadBlockStatHashLock	用于保护global_bad_block_stat hash表。
Page	LOCK事件	page	对表页面加锁。
Tuple	LOCK事件	tuple	对页面上的tuple加锁。
object	LOCK事件	object	加对象锁。
user	LOCK事件	userlock	加用户锁。
advisor	LOCK事件	advisory	加advisory锁。
ODBC	LWLOCK事件	ExtensionConnectorLibLock	用于初始化ODBC连接场景，在加载与卸载特定动态库时进行加锁。
COMM	等待事件	Accept client conn - Total Time	GaussDB数据库接受客户端的连接请求时，从建立连接开始到连接成功过程的整体耗时。

版权所有 © 华为云计算技术有限公司

模块分类	事件分类	事件	说明
COMM	等待事件	Accept client conn - ThrdPool - add epoll	GaussDB数据库接受客户端的连接请求时，线程池模式下，从建立连接开始，到线程池将session的句柄加入到epoll的耗时。
COMM	等待事件	Accept client conn - ThrdPool - wait worker	GaussDB数据库接受客户端的连接请求时，线程池模式下，从线程池将session的句柄加入到epoll开始，到线程池中工作线程开始工作的耗时。
COMM	等待事件	Accept client conn - ThrdPool - init session	GaussDB数据库接受客户端的连接请求时，线程池模式下，从线程池中工作线程开始工作，到session初始化结束的耗时。
COMM	等待事件	Accept client conn - Worker - init proc	GaussDB数据库接受客户端的连接请求时，非线程池模式下，从建立连接开始，到线程初始化结束的耗时。
COMM	等待事件	Accept client conn - Worker - init session	GaussDB数据库接受客户端的连接请求时，非线程池模式下，从线程初始化结束，到session初始化结束的耗时。

8.2.14.4 DBE_PERF.GLOBAL_DATANODES_WAIT_EVENTS

该视图显示DN节点wait event的相关统计信息，如表8-193所示。当参数enable_direct_standby_datanodes开启时，显示的是备DN节点wait event的统计信息。

表 8-193 DBE_PERF.GLOBAL_DATANODES_WAIT_EVENTS 字段

名称	类型	描述
nodename	text	数据库节点名称。
type	text	event类型。
event	text	event名称。
wait	bigint	等待锁失败的次数。
failed_wait	bigint	失败的等待次数。
total_wait_time	bigint	总等待时间（单位：微秒）。
avg_wait_time	bigint	平均等待时间（单位：微秒）。
max_wait_time	bigint	最大等待时间（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
min_wait_time	bigint	最小等待时间（单位：微秒）。
last_updated	timestamp with time zone	最后一次更新该事件的时间。

8.2.15 Configuration

8.2.15.1 DBE_PERF.CONFIG_SETTINGS

该视图显示数据库运行时参数的相关信息，如表8-194所示。

表 8-194 DBE_PERF.CONFIG_SETTINGS 字段

名称	类型	描述
name	text	参数名称。
setting	text	参数当前值。
unit	text	参数的隐式结构。
category	text	参数的逻辑组。
short_desc	text	参数的简单描述。
extra_desc	text	参数的详细描述。
context	text	设置参数值的上下文，包括internal、postmaster、sighup、backend、superuser和user。
vartype	text	参数类型，包括bool、enum、integer、real和string。
source	text	参数的赋值方式。
min_val	text	参数最小值。如果参数类型不是数值型，那么该字段值为null。
max_val	text	参数最大值。如果参数类型不是数值型，那么该字段值为null。
enumvals	text[]	enum类型参数合法值。如果参数类型不是enum型，那么该字段值为null。
boot_val	text	数据库启动时参数默认值。
reset_val	text	数据库重置时参数默认值。
sourcefile	text	设置参数值的配置文件。如果参数不是通过配置文件赋值，那么该字段值为null。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
sourceline	integer	设置参数值的配置文件的行号。如果参数不是通过配置文件赋值，那么该字段值为null。

8.2.15.2 DBE_PERF.GLOBAL_CONFIG_SETTINGS

该视图显示各节点数据库运行时参数的相关信息，如表8-195所示。

表 8-195 DBE_PERF.GLOBAL_CONFIG_SETTINGS 字段

名称	类型	描述
node_name	text	节点名称。
name	text	参数名称。
setting	text	参数当前值。
unit	text	参数的隐式结构。
category	text	参数的逻辑组。
short_desc	text	参数的简单描述。
extra_desc	text	参数的详细描述。
context	text	设置参数值的上下文，包括internal、postmaster、sighup、backend、superuser和user。
vartype	text	参数类型，包括bool、enum、integer、real和string。
source	text	参数的赋值方式。
min_val	text	参数最大值。如果参数类型不是数值型，那么该字段值为null。
max_val	text	参数最小值。如果参数类型不是数值型，那么该字段值为null。
enumvals	text[]	enum类型参数合法值。如果参数类型不是enum型，那么该字段值为null。
boot_val	text	数据库启动时参数默认值。
reset_val	text	数据库重置时参数默认值。
sourcefile	text	设置参数值的配置文件。如果参数不是通过配置文件赋值，那么该字段值为null。
sourceline	integer	设置参数值的配置文件的行号。如果参数不是通过配置文件赋值，那么该字段值为null。

版权所有 © 华为云计算技术有限公司

8.2.16 Operator

8.2.16.1 DBE_PERF.OPERATOR_HISTORY_TABLE

该视图显示执行作业结束后的算子相关的记录。此数据是从内核中转储到视图中的数据，如表8-196所示。

表 8-196 DBE_PERF.OPERATOR_HISTORY_TABLE 字段

名称	类型	描述
queryid	bigint	语句执行使用的内部query_id。
pid	bigint	后端线程id。
plan_node_id	integer	查询对应的执行计划的plan node id。
plan_node_name	text	对应于plan_node_id的算子的名称。
start_time	timestamp with time zone	该算子处理第一条数据的开始时间。
duration	bigint	该算子到结束时候总的执行时间（ms）。
query_dop	integer	当前算子执行时的并行度。
estimated_rows	bigint	优化器估算的行数信息。
tuple_processed	bigint	当前算子返回的元素个数。
min_peak_memory	integer	当前算子在所有DN上的最小内存峰值（MB）。
max_peak_memory	integer	当前算子在所有DN上的最大内存峰值（MB）。
average_peak_memory	integer	当前算子在所有DN上的平均内存峰值（MB）。
memory_skew_percent	integer	当前算子在各DN间的内存使用倾斜率。
min_spill_size	integer	若发生下盘，所有DN上下盘的最小数据量（MB），默认为0。
max_spill_size	integer	若发生下盘，所有DN上下盘的最大数据量（MB），默认为0。
average_spill_size	integer	若发生下盘，所有DN上下盘的平均数据量（MB），默认为0。
spill_skew_percent	integer	若发生下盘，DN间下盘倾斜率。
min_cpu_time	bigint	该算子在所有DN上的最小执行时间（ms）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_cpu_time	bigint	该算子在所有DN上的最大执行时间（ms）。
total_cpu_time	bigint	该算子在所有DN上的总执行时间（ms）。
cpu_skew_percent	integer	DN间执行时间的倾斜率。
warning	text	主要显示如下几类告警信息： • Sort/SetOp/HashAgg/HashJoin spill • Spill file size large than 256MB • Broadcast size large than 100MB • Early spill • Spill times is greater than 3 • Spill on memory adaptive • Hash table conflict

8.2.16.2 DBE_PERF.OPERATOR_RUNTIME

该视图显示当前用户正在执行的作业的算子相关信息，如表8-197所示。

表 8-197 DBE_PERF.OPERATOR_RUNTIME 字段

名称	类型	描述
queryid	bigint	语句执行使用的内部query_id。
pid	bigint	后端线程id。
plan_node_id	integer	查询对应的执行计划的plan node id。
plan_node_name	text	对应于plan_node_id的算子的名称。
start_time	timestamp with time zone	该算子处理第一条数据的开始时间。
duration	bigint	该算子到结束时候总的执行时间（ms）。
status	text	当前算子的执行状态，包括finished和running。
query_dop	integer	当前算子执行时的并行度。
estimated_rows	bigint	优化器估算的行数信息。
tuple_processed	bigint	当前算子返回的元素个数。
min_peak_memory	integer	当前算子在所有DN上的最小内存峰值（MB）。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
max_peak_memory	integer	当前算子在所有DN上的最大内存峰值（MB）。
average_peak_memory	integer	当前算子在所有DN上的平均内存峰值（MB）。
memory_skew_percent	integer	当前算子在各DN间的内存使用倾斜率。
min_spill_size	integer	若发生下盘，所有DN上下盘的最小数据量（MB），默认为0。
max_spill_size	integer	若发生下盘，所有DN上下盘的最大数据量（MB），默认为0。
average_spill_size	integer	若发生下盘，所有DN上下盘的平均数据量（MB），默认为0。
spill_skew_percent	integer	若发生下盘，DN间下盘倾斜率。
min_cpu_time	bigint	该算子在所有DN上的最小执行时间（ms）。
max_cpu_time	bigint	该算子在所有DN上的最大执行时间（ms）。
total_cpu_time	bigint	该算子在所有DN上的总执行时间（ms）。
cpu_skew_percent	integer	DN间执行时间的倾斜率。
warning	text	主要显示如下几类告警信息： • Sort/SetOp/HashAgg/HashJoin spill • Spill file size large than 256MB • Broadcast size large than 100MB • Early spill • Spill times is greater than 3 • Spill on memory adaptive • Hash table conflict

8.2.16.3 DBE_PERF.GLOBAL_OPERATOR_HISTORY

该视图显示的是当前用户在所有CN节点上执行作业结束后的算子的相关记录，如**表8-198**所示。

表 8-198 DBE_PERF.GLOBAL_OPERATOR_HISTORY 字段

名称	类型	描述
queryid	bigint	语句执行使用的内部query_id。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
pid	bigint	后端线程id。
plan_node_id	integer	查询对应的执行计划的plan node id。
plan_node_name	text	对应于plan_node_id的算子的名称。
start_time	timestamp with time zone	该算子处理第一条数据的开始时间。
duration	bigint	该算子到结束时候总的执行时间（ms）。
query_dop	integer	当前算子执行时的并行度。
estimated_rows	bigint	优化器估算的行数信息。
tuple_processed	bigint	当前算子返回的元素个数。
min_peak_memory	integer	当前算子在所有DN上的最小内存峰值（MB）。
max_peak_memory	integer	当前算子在所有DN上的最大内存峰值（MB）。
average_peak_memory	integer	当前算子在所有DN上的平均内存峰值（MB）。
memory_skew_percent	integer	当前算子在各DN间的内存使用倾斜率。
min_spill_size	integer	若发生下盘，所有DN上下盘的最小数据量（MB），默认为0。
max_spill_size	integer	若发生下盘，所有DN上下盘的最大数据量（MB），默认为0。
average_spill_size	integer	若发生下盘，所有DN上下盘的平均数据量（MB），默认为0。
spill_skew_percent	integer	若发生下盘，DN间下盘倾斜率。
min_cpu_time	bigint	该算子在所有DN上的最小执行时间（ms）。
max_cpu_time	bigint	该算子在所有DN上的最大执行时间（ms）。
total_cpu_time	bigint	该算子在所有DN上的总执行时间（ms）。
cpu_skew_percent	integer	DN间执行时间的倾斜率。

 版权所有 © 华为云计算技术有限公司

名称	类型	描述
warning	text	主要显示如下几类告警信息： • Sort/SetOp/HashAgg/HashJoin spill • Spill file size large than 256MB • Broadcast size large than 100MB • Early spill • Spill times is greater than 3 • Spill on memory adaptive • Hash table conflict

8.2.16.4 DBE_PERF.GLOBAL_OPERATOR_HISTORY_TABLE

该视图显示所有CN节点执行作业结束后的算子相关的记录。此数据是从内核中转储到系统表GS_WLM_OPERATOR_INFO中的数据。该视图是查询所有CN节点系统表GS_WLM_OPERATOR_INFO的汇聚视图。

表字段同**表8-198**。

8.2.16.5 DBE_PERF.GLOBAL_OPERATOR_RUNTIME

该视图显示当前用户在所有CN节点上正在执行的作业的算子相关信息，如**表8-199**所示。

表 8-199 DBE_PERF.GLOBAL_OPERATOR_RUNTIME 的字段

名称	类型	描述
queryid	bigint	语句执行使用的内部query_id。
pid	bigint	后端线程id。
plan_node_id	integer	查询对应的执行计划的plan node id。
plan_node_name	text	对应于plan_node_id的算子的名称。
start_time	timestamp with time zone	该算子处理第一条数据的开始时间。
duration	bigint	该算子到结束时候总的执行时间（ms）。
status	text	当前算子的执行状态，包括finished和running。
query_dop	integer	当前算子执行时的并行度。
estimated_rows	bigint	优化器估算的行数信息。
tuple_processed	bigint	当前算子返回的元素个数。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
min_peak_memory	integer	当前算子在所有DN上的最小内存峰值（MB）。
max_peak_memory	integer	当前算子在所有DN上的最大内存峰值（MB）。
average_peak_memory	integer	当前算子在所有DN上的平均内存峰值（MB）。
memory_skew_percent	integer	当前算子在各DN间的内存使用倾斜率。
min_spill_size	integer	若发生下盘，所有DN上下盘的最小数据量（MB），默认为0。
max_spill_size	integer	若发生下盘，所有DN上下盘的最大数据量（MB），默认为0。
average_spill_size	integer	若发生下盘，所有DN上下盘的平均数据量（MB），默认为0。
spill_skew_percent	integer	若发生下盘，DN间下盘倾斜率。
min_cpu_time	bigint	该算子在所有DN上的最小执行时间（ms）。
max_cpu_time	bigint	该算子在所有DN上的最大执行时间（ms）。
total_cpu_time	bigint	该算子在所有DN上的总执行时间（ms）。
cpu_skew_percent	integer	DN间执行时间的倾斜率。
warning	text	主要显示如下几类告警信息： • Sort/SetOp/HashAgg/HashJoin spill • Spill file size large than 256MB • Broadcast size large than 100MB • Early spill • Spill times is greater than 3 • Spill on memory adaptive • Hash table conflict

8.2.17 Workload Manager

8.2.17.1 DBE_PERF.WLM_CGROUP_CONFIG

该视图显示当前执行作业的控制组的信息，如表8-200所示。

版权所有 © 华为云计算技术有限公司

表 8-200 DBE_PERF.WLM_CGROUP_CONFIG 字段

名称	类型	描述
cgroup_name	text	控制组的名称。
priority	integer	作业的优先级。
usage_percent	integer	控制组占用的百分比。
shares	bigint	控制组分配的CPU资源配额。
cpuacct	bigint	CPU配额分配。
cpuset	text	CPU限额分配。
relpath	text	控制组的相对路径。
valid	text	该控制组是否有效。
node_group	text	node group名称。

8.2.17.2 DBE_PERF.WLM_CLUSTER_RESOURCE_RUNTIME

该视图显示的是所有DN资源的汇总信息字段，如表8-201所示。

表 8-201 DBE_PERF.WLM_CLUSTER_RESOURCE_RUNTIME 字段

名称	类型	描述
min_mem_util	integer	DN最小内存使用率。
max_mem_util	integer	DN最大内存使用率。
min_cpu_util	integer	DN最小CPU使用率。
max_cpu_util	integer	DN最大CPU使用率。
min_io_util	integer	DN最小I/O使用率。
max_io_util	integer	DN最大I/O使用率。
used_mem_rate	integer	物理节点最大内存使用率。

8.2.17.3 DBE_PERF.WLM_CONTROLGROUP_CONFIG

该视图显示数据库内所有的控制组信息，如表8-202所示。

表 8-202 DBE_PERF.WLM_CONTROLGROUP_CONFIG 字段

名称	类型	描述
name	text	控制组的名称。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
type	text	控制组的类型。
gid	bigint	控制组id。
classgid	bigint	Workload所属Class的控制组id。
class	text	Class控制组。
workload	text	Workload控制组。
shares	bigint	控制组分配的CPU资源配额。
limits	bigint	控制组分配的CPU资源限额。
wdlevel	bigint	Workload控制组层级。
cpucores	text	控制组使用的CPU核的信息。

8.2.17.4 DBE_PERF.WLM_CONTROLGROUP_NG_CONFIG

该视图显示数据库内所有的node group的控制组信息，如表8-203所示。

表 8-203 DBE_PERF.WLM_CONTROLGROUP_NG_CONFIG 字段

名称	类型	描述
group_name	text	控制组的名称。
group_type	text	控制组的类型。
gid	bigint	控制组id。
classgid	bigint	Workload所属Class的控制组id。
class	text	Class控制组。
group_workload	text	Workload控制组。
shares	bigint	控制组分配的CPU资源配额。
limits	bigint	控制组分配的CPU资源限额。
wdlevel	bigint	Workload控制组层级。
cpucores	text	控制组使用的CPU核的信息。
group_kind	text	node group类型，取值包括i，n，v和e： ● i：表示installation node group。 ● n：表示普通集群node group。 ● e：表示弹性集群。

版权所有 © 华为云计算技术有限公司

8.2.17.5 DBE_PERF.WLM_RESOURCEPOOL_RUNTIME

该视图显示资源池上的一些统计信息，如表8-204所示。

表 8-204 DBE_PERF.WLM_RESOURCEPOOL_RUNTIME 的字段

名称	类型	描述
rpoid	oid	资源池的OID。
respool	name	资源池的名称。
control_group	name	资源池的控制组。
parentid	oid	父资源池的OID。
ref_count	integer	关联到该资源池上的作业数量。
active_points	integer	资源池上已经使用的点数。
running_count	integer	正在资源池上运行的作业数量。
waiting_count	integer	正在资源池上排队的作业数量。
io_limits	integer	资源池的iops上限。
io_priority	integer	资源池的I/O优先级。

8.2.17.6 DBE_PERF.WLM_USER_RESOURCE_CONFIG

该视图显示用户的资源配置信息，如表8-205所示。

表 8-205 DBE_PERF.WLM_USER_RESOURCE_CONFIG 字段

名称	类型	描述
userid	oid	用户oid。
username	name	用户名称。
sysadmin	boolean	用户是否是sysadmin。
rpoid	oid	资源池的oid。
respool	name	资源池的名称。
parentid	oid	父用户的oid。
totalspace	bigint	占用总空间大小。
spacelimit	bigint	空间最大上限。
childcount	integer	子用户数量。
childlist	texto	子用户的列表。

版权所有 © 华为云计算技术有限公司

8.2.17.7 DBE_PERF.WLM_USER_RESOURCE_RUNTIME

该视图显示所有用户资源使用情况，需要使用管理员用户进行查询，如表8-206所示。此视图在GUC参数use_workload_manager为on时才有效。

表 8-206 DBE_PERF.WLM_USER_RESOURCE_RUNTIME 字段

名称	类型	描述
username	name	用户名。
used_memory	integer	正在使用的内存大小，单位MB。
total_memory	integer	可以使用的内存大小，单位MB。值为0表示未限制最大可用内存，其限制取决于数据库最大可用内存。
used_cpu	integer	正在使用的CPU核数。
total_cpu	integer	在该机器节点上，用户关联控制组的CPU核数总和。
used_space	bigint	已使用的存储空间大小，单位KB。
total_space	bigint	可使用的存储空间大小，单位KB，值为-1表示未限制最大存储空间。
used_temp_space	bigint	已使用的临时空间大小（预留字段，暂未使用），单位KB。
total_temp_space	bigint	可使用的临时空间大小（预留字段，暂未使用），单位KB，值为-1表示未限制最大临时存储空间。
used_spill_space	bigint	已使用的下盘空间大小（预留字段，暂未使用），单位KB。
total_spill_space	bigint	可使用的下盘空间大小（预留字段，暂未使用），单位KB，值为-1表示未限制最大下盘空间。

8.2.17.8 DBE_PERF.WLM_WORKLOAD_HISTORY_INFO

该视图显示作业结束后或已被处理异常后的负载管理相关信息，如表8-207所示。

表 8-207 DBE_PERF.WLM_WORKLOAD_HISTORY_INFO 字段

名称	类型	描述
statement	text	执行了异常处理的语句。
block_time	bigint	语句执行前的阻塞时间。
elapsed_time	bigint	语句的实际执行时间。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
total_cpu_time	bigint	语句执行异常处理时DN上CPU使用的总时间。
qualification_time	bigint	语句检查倾斜率的时间周期。
cpu_skew_percent	integer	语句在执行异常处理时DN上CPU使用的倾斜率。
control_group	text	语句执行异常处理时所使用的Cgroups。
status	text	语句执行异常处理后的状态，包括： • pending：执行前预备状态。 • running：执行进行状态。 • finished：执行正常结束。 • abort：执行异常终止。
action	text	语句执行的异常处理动作，包括： • abort：执行终止操作。 • adjust：执行Cgroups调整操作，目前只有降级操作。 • finish：正常结束。

8.2.17.9 DBE_PERF.WLM_WORKLOAD_RUNTIME

该视图显示当前用户在CN上执行作业时当前CN的作业状态信息，如表8-208所示。

表 8-208 DBE_PERF.WLM_WORKLOAD_RUNTIME 字段

名称	类型	描述
node_name	text	作业执行所在CN的名称。
thread_id	bigint	后端线程ID。
processid	integer	后端线程的pid。
time_stamp	bigint	语句执行的开始时间。
username	name	登录到该后端的用户名。
memory	integer	语句所需的内存大小。
active_points	integer	语句在资源池上消耗的资源点数。
max_points	integer	语句在资源池上消耗的总资源点数。
priority	integer	作业的优先级。
resource_pool	text	作业所在资源池。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
status	text	作业执行的状态，包括： • pending： 阻塞状态。 • running：执行状态。 • finished：结束状态。 • aborted：终止状态。 • unknown：未知状态。
control_group	text	作业所使用的Cgroups。
enqueue	text	作业的排队信息，包括： • GLOBAL：全局排队。 • RESPOOL：资源池排队。 • ACTIVE：不排队。
query	text	正在执行的语句。
node_group	text	node group名称。

8.2.17.10 DBE_PERF.GLOBAL_WLM_WORKLOAD_RUNTIME

显示当前用户在每个CN上执行作业时在CN上的状态信息，如表8-209所示。需要有监控管理员权限才可以访问。

表 8-209 DBE_PERF.GLOBAL_WLM_WORKLOAD_RUNTIME 字段

名称	类型	描述
node_name	text	作业执行所在的CN的名称。
thread_id	bigint	后端线程ID。
processid	integer	线程的lwpid。
time_stamp	bigint	语句执行的开始时间。
username	name	登录到该后端的用户名。
memory	integer	语句所需的内存大小。
active_points	integer	语句在资源池上消耗的资源点数。
max_points	integer	资源在资源池上的最大资源数。
priority	integer	作业的优先级。
resource_pool	text	作业所在资源池。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
status	text	作业执行的状态，包括： • pending：阻塞状态。 • running：执行状态。 • finished：结束状态。 • aborted：终止状态。 • unknown：未知状态。
control_group	name	作业所使用的Cgroups。
enqueue	text	作业的排队信息，包括： • GLOBAL：全局排队。 • RESPOOL：资源池排队。 • ACTIVE：不排队。
query	text	正在执行的语句。
node_group	text	node group名称。

8.2.17.11 DBE_PERF.LOCAL_IO_WAIT_INFO

显示当前节点I/O管控的实时统计信息，如表8-210所示。

表 8-210 DBE_PERF.LOCAL_IO_WAIT_INFO 字段

名称	类型	描述
node_name	text	节点名称。
device_name	text	节点挂载的数据磁盘名称。
read_per_second	double precision	读完成每秒次数。
write_per_second	double precision	写完成每秒次数。
write_ratio	double precision	写磁盘占总的I/O使用的比例。
io_util	double precision	每秒I/O所占CPU总时间的百分比。
total_io_util	integer	过去三次I/O所占CPU总时间的等级（取值为0~6）。
tick_count	integer	更新磁盘I/O信息的周期，固定为1秒，每次读取数据前都会被清零。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
io_wait_list_len	integer	I/O请求线程等待队列的大小，若为0，则表示当前没有I/O被管控。

8.2.17.12 DBE_PERF.GLOBAL_IO_WAIT_INFO

显示所有节点I/O管控的实时统计信息，如表8-211所示。

表 8-211 DBE_PERF.GLOBAL_IO_WAIT_INFO 字段

名称	类型	描述
node_name	text	节点名称。
device_name	text	节点挂载的数据磁盘名称。
read_per_second	double precision	读完成每秒次数。
write_per_second	double precision	写完成每秒次数。
write_ratio	double precision	写磁盘占总的I/O使用的比例。
io_util	double precision	每秒I/O所占CPU总时间的百分比。
total_io_util	integer	过去三次I/O所占CPU总时间的等级（取值为0~6）。
tick_count	integer	更新磁盘I/O信息的周期，固定为1秒，每次读取数据前都会被清零。
io_wait_list_len	integer	I/O请求线程等待队列的大小，若为0，则表示当前没有I/O被管控。

8.2.18 Global Plancache

GPC相关视图在GUC参数enable_global_plancache打开的状态下才有效。

8.2.18.1 DBE_PERF.LOCAL_PLANCACHE_STATUS

该视图显示当前节点的GPC全局计划缓存状态信息，如表8-212所示。查询该视图必须具有MONADMIN权限。

版权所有 © 华为云计算技术有限公司

表 8-212 DBE_PERF.LOCAL_PLANCACHE_STATUS 字段

名称	类型	描述
nodename	text	所属节点名称。
query	text	查询语句text。
refcount	integer	被引用次数。
valid	bool	是否合法。
databaseid	oid	所属数据库id。
schema_name	text	所属schema。
params_num	integer	参数数量。
func_id	oid	该plancache所在存储过程oid，如果不属于存储过程则为0。
pkg_id	oid	该plancache所在存储过程所属的Package，如果不属于Package则为0。
stmt_id	integer	显示存储过程内语句计划的序号。
size	bigint	表示计划占用内存大小。
parameterized_info	text	该列表示计划存入GPC的途径： ● 自动参数化途径：强制绑参force或相同语句复用exact。 ● 非自动参数化途径：normal。
sql_hash	bigint	spm中sql的唯一标识，分布式下为0。
plan_hash	bigint	spm中当前sql下的plan的唯一标识，分布式下为0。
unique_sql_id	bigint	数据库中SQL的唯一标识。
outline	text	当前计划对应的outline的所有hint的组合字符串，分布式下为空。
last_used_time	timestamp with time zone	该计划缓存最近使用的时间，失效计划不记录最近使用时间。
depend_object_ids	text	该计划缓存依赖的oid对象列表。
hash_value	bigint	参数的哈希值，不保证唯一。
plan_info	text	支持显示有效计划的详细计划信息。视图默认此列内容为空。
opcache_reuse_cnt	integer	执行缓存复用的次数。
opcache_mem_size	bigint	执行缓存占用的内存大小。

版权所有 © 华为云计算技术有限公司

8.2.18.2 DBE_PERF.GLOBAL_PLANCACHE_STATUS

该视图显示全部节点上GPC全局计划缓存的状态信息。查询该视图必须具有MONADMIN权限。

具体的字段见**DBE_PERF.LOCAL_PLANCACHE_STATUS**。

8.2.18.3 DBE_PERF.LOCAL_PREPARE_STATEMENT_STATUS

该视图显示当前节点的GPC全局计划缓存状态对应的prepare statement信息，如**表8-213**所示。

表 8-213 LOCAL_PREPARE_STATEMENT_STATUS 字段

名称	类型	描述
nodename	text	所属节点名称。
cn_sess_id	bigint	其来自的cn的sessionid。
cn_node_id	integer	其来自的cn的node_id。
cn_time_line	integer	其来自的cn的重启次数。
statement_name	text	其statement name。
refcount	integer	其对应的plancache的被引用次数。
is_shared	bool	其对应plancache是否共享。
query	text	对应的query语句。

8.2.18.4 DBE_PERF.GLOBAL_PREPARE_STATEMENT_STATUS

该视图显示全部节点的GPC全局计划缓存状态对应的prepare statement信息。

具体字段请参见**表8-213**。

8.2.18.5 DBE_PERF.LOCAL_PLANCACHE_CLEAN

该视图用于清理当前节点上无人使用的全局计划缓存。返回值为Boolean类型。

集群创建后的默认情况下，查询该视图必须具有MONADMIN权限。

8.2.18.6 DBE_PERF.GLOBAL_PLANCACHE_CLEAN

该视图用于清理全部节点上无人使用的全局计划缓存。返回值为Boolean类型。

集群创建后的默认情况下，查询该视图必须具有MONADMIN权限。

8.2.19 RTO & RPO

版权所有 © 华为云计算技术有限公司

8.2.19.1 DBE_PERF.GLOBAL_RTO_STATUS

该视图显示关于主机和备机的日志流控信息（本节点除外、DN上不可使用），如表8-214所示。

表 8-214 DBE_PERF.GLOBAL_RTO_STATUS 字段

参数	类型	描述
node_name	text	节点的名称，包含主机和备机。
rto_info	text	流控的信息，包含了备机当前的日志流控时间（单位：秒），备机通过GUC参数设置的预期流控时间（单位：秒），为了达到这个预期主机所需要的睡眠时间（单位：微秒）。

8.2.19.2 DBE_PERF.GLOBAL_STREAMING_HADR_RTO_AND_RPO_STAT

该视图显示流式容灾的主集群和备集群日志流控信息（只可在主集群的CN使用，DN以及备集群上均不可获取到统计信息），如表8-215所示。

表 8-215 DBE_PERF.GLOBAL_STREAMING_HADR_RTO_AND_RPO_STAT 字段

参数	类型	描述
hadr_sender_node_name	text	节点的名称，包含主集群和备集群首备。
hadr_receiver_node_name	text	备集群首备名称。
current_rto	int	流控的信息，当前主备集群的日志rto时间（单位：秒）。-1标识灾备集群多数派异常。
target_rto	int	流控的信息，目标主备集群间的rto时间（单位：秒）。
current_rpo	int	流控的信息，当前主备集群的日志rpo时间（单位：秒）。-1标识灾备集群多数派异常。
target_rpo	int	流控的信息，目标主备集群间的rpo时间（单位：秒）。
rto_sleep_time	int	RTO流控信息，为了达到目标RTO，预期主机walsender所需要的睡眠时间（单位：微秒）。
rpo_sleep_time	int	RPO流控信息，为了达到目标RPO，预期主机xlogInsert所需要的睡眠时间（单位：微秒）。

8.2.20 AI Watchdog

版权所有 © 华为云计算技术有限公司

8.2.20.1 DBE_PERF.AI_WATCHDOG_MONITOR_STATUS

表 8-216 DBE_PERF.AI_WATCHDOG_MONITOR_STATUS 字段

参数	类型	描述
metric_name	text	metric指标名称： ● tps：TPS。 ● tps_hourly：每小时的TPS均值。 ● shared_used_mem：共享内存使用量（MB）。 ● dynamic_used_shrctx：共享内存上下文使用量（MB）。 ● other_used_mem：其他内存使用量（MB）。 ● process_used_mem：系统常驻内存使用量（MB）。 ● dynamic_used_mem：动态内存使用量（MB）。 ● malloc_failures：每个采集间隔内的内存分配失败次数。 ● D_state_rate：D状态线程比例。 ● R_state_rate：R状态线程比例。 ● S_state_rate：S状态线程比例。 ● db_state：数据库的状态（68表示D、82表示R、83表示S）。 ● cpu_usage：CPU使用率，上限100。 ● disk_io：两个采集间隔内的磁盘I/O延迟。 ● network_io：两个采集间隔内的网络I/O延迟。 ● threadpool_usage：线程池使用率。 ● threadpool_hang_rate：线程池group处于hang状态的比例。
max_length	int	采集队列长队。
current_length	int	当前采集到的样本数。
collection_interval	int	采集间隔，单位秒。
latest_value	int	上次采集到的值，没采集到为null。
last_report	timestamp	上次采集时刻。

版权所有 © 华为云计算技术有限公司

8.2.20.2 DBE_PERF.AI_WATCHDOG_DETECTION_WARNINGS

表 8-217 DBE_PERF.AI_WATCHDOG_DETECTION_WARNINGS 字段

参数	类型	描述
event	text	事件名称。
cause	text	事件原因。
details	text	事件详情。
time	timestamp	报告时刻。
need_to_handle	bool	是否需要自动处理。

8.2.20.3 DBE_PERF.AI_WATCHDOG_PARAMETERS

表 8-218 DBE_PERF.AI_WATCHDOG_PARAMETERS 字段

参数	类型	描述
name	text	参数名称，包括如下常用参数： ● enable_ai_watchdog：是否开启本功能。 ● ai_watchdog_max_consuming_time_ms：最大耗时。 ● ai_watchdog_used_memory_kb：本功能当前内存使用。 ● ai_watchdog_detection_times：已检测次数。 ● enable_self_healing：发现问题后是否可以自愈。 ● oom_detected_times：已检测到的OOM次数。 ● hang_detected_times：已检测到的hang次数。 ● enable_oom_detection：是否自动启动了OOM探测功能。 ● in_wait_time：是否处于等待时间。 ● other_used_memory_has_risk：其他内存使用部分是否存在风险。 ● shared_used_mem_has_risk：共享内存上下文使用是否存在风险。 ● dynamic_used_shrctx_has_risk：动态内存使用是存在风险。
value	text	参数值。

版权所有 © 华为云计算技术有限公司

8.2.20.4 DBE_PERF.AI_WATCHDOG_FTASK_STATUS

表 8-219 DBE_PERF.AI_WATCHDOG_FTASK_STATUS 字段

参数	类型	描述
name	text	计数器清0任务名。
timeout_threshold	int	计数器清0任务超时阈值。
register_time	timestamp	计数器清0任务注册时间。
thread_id	bigint	该计数器清0任务在哪一个线程ID上使用。
thread_name	text	该计数器清0任务在哪一个线程名上使用。
last_feed	timestamp	计数器清0任务最近一次计数器清0时间，默认初始值等于register_time。
restart_count	int	同一后台线程异常重复注册同一计数器清0任务名的次数，该参数在任务调用取消注册计数器清0任务接口后清零。
timeout_action	text	注册任务时指定的超时处理动作。当前提供的处理动作如下： • WATCHDOG_REPORT_WARNING：超时定位定界提示。 • WATCHDOG_SEND_SIGTERM：向线程发送SIGTERM信号。 • WATCHDOG_SEND_SIGUSR2：向线程发送SIGUSR2信号。 • WATCHDOG_SEND_SIGQUIT：向线程发送SIGQUIT信号。 • WATCHDOG_EXIT_PROCESS：进程退出。

8.2.21 Trace Function

8.2.21.1 DBE_PERF.pl_trace_func_start

开启某节点trace功能的接口函数。字段信息详见**表8-220**。

表 8-220 DBE_PERF.pl_trace_func_start 入参列表

参数	类型	描述
session_id	bigint	需要开启trace功能的会话ID。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
node	text	需要开启trace功能的节点名称。
max_num	integer	本次trace所能记录的最大信息数量（取值范围：100~10000）。
level	text	本次trace的等级： • session：会话级别，在调用pl_trace_func_end函数或会话关闭前一直开启trace功能。 • portal：portal级别，执行一次存储过程后自动关闭trace功能。 • time：会话级别，在会话级别上添加语句执行时间显示。 • threshold：会话级别，在time级别上只记录大于等于阈值的语句。
threshold	float8	本次trace的阈值（单位：毫秒，仅在level为threshold时生效）。

8.2.21.2 DBE_PERF.pl_trace_func_end

关闭某节点trace功能的接口函数。字段信息详见表8-221。

表 8-221 DBE_PERF.pl_trace_func_end 入参列表

参数	类型	描述
session_id	bigint	需要关闭trace功能的会话ID。
node	text	需要关闭trace功能的节点名称。

8.2.21.3 DBE_PERF.pl_trace_info_clean

清理某节点trace功能记录信息的接口函数。字段信息详见表8-222。

表 8-222 DBE_PERF.pl_trace_info_clean 入参列表

参数	类型	描述
node	text	需要清理trace功能记录信息的节点名称。
session_id	bigint	需要清理trace功能记录信息的会话ID（为空时清理所有信息）。

8.2.21.4 DBE_PERF.pl_get_trace_info

查看当前节点trace功能记录信息的接口函数。字段信息详见表8-223。

版权所有 © 华为云计算技术有限公司

表 8-223 DBE_PERF.pl_get_trace_info 返回值列表

参数	类型	描述
session_id	bigint	本条SQL语句所属的会话ID。
level	text	本条SQL语句的等级。
database	text	本条SQL语句所属数据库名。
func	oid	本条SQL语句所属存储过程的OID。
source	text	本条SQL语句。
plan	text	本条SQL语句的执行计划。
time	text	本条SQL语句的执行时间。
threshold	float8	本条SQL语句的阈值（单位：毫秒）。

8.2.21.5 DBE_PERF.pl_get_global_trace_info

查看全部节点trace功能记录信息的接口函数。字段信息详见表8-224。

表 8-224 DBE_PERF.pl_get_global_trace_info 返回值列表

参数	类型	描述
session_id	bigint	本条SQL语句所属的会话ID。
node	text	本条SQL语句所属的节点名称。
level	text	本条SQL语句的等级。
database	text	本条SQL语句所属数据库名。
func	oid	本条SQL语句所属存储过程的OID。
source	text	本条SQL语句。
plan	text	本条SQL语句的执行计划。
time	text	本条SQL语句的执行时间。
threshold	float8	本条SQL语句的阈值（单位：毫秒）。

8.2.22 废弃视图

8.2.22.1 Query

GaussDB中下列视图在当前版本中已废弃：

- DBE_PERF.GS_SLOW_QUERY_HISTORY

版权所有 © 华为云计算技术有限公司

- DBE_PERF.GLOBAL_SLOW_QUERY_HISTORY
- DBE_PERF.GLOBAL_SLOW_QUERY_INFO
- DBE_PERF.GLOBAL_STATEMENT_COMPLEX_RUNTIME

8.2.22.2 Operator

GaussDB中下列视图在当前版本中已废弃：

DBE_PERF.OPERATOR_HISTORY

8.3 WDR Snapshot Schema

WDR Snapshot在启动后（打开GUC参数enable_wdr_snapshot）,会在用户表空间“pg_default”，数据库“postgres”下的Snapshot Schema中创建对象，用于持久化WDR快照数据，默认初始化用户或监控管理员用户可以访问和操作Snapshot Schema下的对象。

根据GUC参数wdr_snapshot_retention_days来自动管理快照的生命周期。

8.3.1 WDR Snapshot 原信息

8.3.1.1 SNAPSHOT.SNAPSHOT

SNAPSHOT表记录当前系统中存储的WDR快照数据的索引信息、开始和结束时间。只能在系统库中查询到结果，用户库中无法查询，WDR Snapshot在启动后（打开GUC参数enable_wdr_snapshot）会触发创建该表，如**表8-225**所示。只有初始化用户或监控管理员用户有权限查看。

表 8-225 SNAPSHOT 字段

名称	类型	描述	示例
snapshot_id	bigint	WDR快照序号。	1
start_ts	timestamp	WDR快照的开始时间。	2019-12-28 17:11:27.423742+08
end_ts	timestamp	WDR快照的结束时间。	2019-12-28 17:11:43.67726+08
version	int	WDR快照的版本信息。	1
snap_flag	bigint	WDR快照的属性。 • NULL：代表该快照为旧版本快照。 • 0：表示该次快照为全量快照。 • 1：表示该次快照为增量快照。	0

版权所有 © 华为云计算技术有限公司

名称	类型	描述	示例
base_snapshot_id	bigint	WDR快照对应的全量快照ID。	1

8.3.1.2 SNAPSHOT.TABLES_SNAP_TIMESTAMP

TABLES_SNAP_TIMESTAMP表记录所有存储的WDR Snapshot中数据库、表对象以及WDR快照的开始和结束时间，WDR Snapshot在启动后（打开GUC参数enable_wdr_snapshot）会触发创建该表，如表8-226所示。只有初始化用户或监控管理员用户有权限查看。

表 8-226 TABLES_SNAP_TIMESTAMP 字段

名称	类型	描述	示例
snapshot_id	bigint	WDR快照序号。	1
db_name	text	WDR Snapshot对应的database。	tpcc1000
tablename	text	WDR Snapshot对应的table。	snap_summary_statement
start_ts	timestamp	WDR快照的开始时间。	2019-12-28 17:11:27.425849+08
end_ts	timestamp	WDR快照的结束时间。	2019-12-28 17:11:27.707398+08

8.3.1.3 SNAPSHOT.SNAP_SEQ

SNAP_SEQ是一个递增的sequence，其为WDR Snapshot提供快照的ID。

WDR Snapshot在启动后（打开GUC参数enable_wdr_snapshot）会触发创建该sequence。

8.3.1.4 SNAPSHOT.SNAP_QUERY

SNAP_QUERY表记录所有存储的WDR Snapshot中的SQL语句信息，在WDR Snapshot启动后（即GUC参数enable_wdr_snapshot值为on时）会触发创建该表，如表8-227所示。

表 8-227 SNAP_QUERY 字段

名称	类型	描述
snap_node_name	name	节点名称。
snap_node_id	integer	节点的ID。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
snap_user_name	name	用户名称，若查询时执行query语句的用户已不存在，则显示为*REMOVED_USER*。
snap_user_id	oid	用户OID。
snap_unique_sql_id	bigint	归一化的SQL id。
snap_query	text	归一化的SQL。 备注：长度受GUC参数instr_unique_sql_size控制。
snap_last_updated	timestamp with time zone	最后一次更新该语句的时间。
snap_parent_unique_sql_id	bigint	父语句的unique_sql_id，非存储过程子语句该值为0。
snap_db_name	name	连接后端的数据库名称，若查询时执行query语句的数据库已不存在，则显示为*REMOVED_DB*。
snap_db_id	oid	连接后端的数据库OID。

8.3.1.5 SNAPSHOT.SNAP_PLAN

SNAP_PLAN表记录所有存储的WDR Snapshot中的SQL语句信息，在WDR Snapshot启动后（即GUC参数enable_wdr_snapshot值为on时）会触发创建该表，如表8-228所示。

表 8-228 SNAP_PLAN 字段

名称	类型	描述
snap_node_id	integer	节点的ID。
snap_user_id	oid	用户OID。
snap_unique_sql_id	bigint	归一化的SQL id。
snap_last_updated	timestamp with time zone	最后一次更新该语句的时间。
snap_db_id	oid	连接后端的数据库OID。
snap_sql_hash	bigint	SQL语句对应的hash值，为0。
snap_plan_hash	bigint	SQL语句执行计划对应的hash值，为0。
snap_query_plan	text	SQL语句的执行计划。

版权所有 © 华为云计算技术有限公司

8.3.2 WDR Snapshot 数据表

WDR Snapshot数据表命名原则：snap_{源数据表}。

WDR Snapshot数据表来源请参见**DBE_PERF Schema**章节下的视图，WDR Snapshot在启动后（打开GUC参数enable_wdr_snapshot）会触发创建WDR Snapshot数据表。

说明

初始用户或者监控管理员用户具有权限查看WDR Snapshot数据表。

8.4 DBE_PLDEBUGGER Schema

DBE_PLDEBUGGER Schema下的系统函数用于调试存储过程，仅管理员有权限执行这些调试接口，但无修改和创建新函数的权限。目前支持的接口及其描述如下所示。

须知

- 当在函数体中创建用户时，调用attach、next、continue、info_code、step、info_breakpoint、backtrace、finish中会返回密码的明文。因此不建议用户在函数体中创建用户。
- 仅支持在CN节点上调试存储过程，并且调试端和被调试端需连接在同一CN节点。
- 调试端和被调试端需连接在同一database，否则使用函数oid获取函数信息的接口将不可用。
- 用户在当前会话使用dbe_pldebugger.turn_on函数开启存储过程调试后，当前会话的所有PL/SQL类型存储过程将不能下推至DN节点。因此，请在调试结束后，及时使用dbe_pldebugger.turn_off函数关闭当前会话的存储过程。
- 不支持直接调试触发器，不支持EXECUTE DIRECT ON将语句下发至DN节点触发调试。
- 在使用dbe_pldebugger.turn_on函数开启存储过程调试前生成的预编译语句可能无法被调试。
- dbe_pldebugger.error_*类型函数，只能在报错断住时才能使用。
- 当存储过程调试时，如果被调试的存储过程中涉及加锁的操作，请注意在调试端勿执行可能导致死锁的操作。
- 三权分立开启时，仅允许初始用户和属主可以调试定义者权限的函数；三权分立关闭时，允许系统管理员、属主、初始用户调试定义者权限的函数。

对应权限角色为gs_role_pldebugger，可以由管理员用户通过如下命令将debugger权限赋权给该用户。

```
GRANT gs_role_pldebugger to user;
```

需要有两个客户端连接数据库，一个客户端负责执行调试接口作为debug端，另一个客户端执行调试函数，控制server端存储过程执行。示例如下：

- 准备调试

  通过PG_PROC，查找到待调试存储过程的oid，并执行DBE_PLDEBUGGER.turn_on(oid)。本客户端会作为server端使用。

  ```
  gaussdb=# CREATE OR REPLACE PROCEDURE test_debug ( IN  x INT)
  AS
  ```


版权所有 © 华为云计算技术有限公司

```
BEGIN
    INSERT INTO t1 (a) VALUES (x);
    DELETE FROM t1 WHERE a = x;
END;
/
CREATE PROCEDURE
gaussdb=# SELECT OID FROM PG_PROC WHERE PRONAME='test_debug';
  oid
-------
 16389
(1 row)
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.turn_on(16389);
 nodename | port
----------+------
 datanode |    0
(1 row)
```

- 开始调试

  server端执行存储过程，会在存储过程内第一条SQL语句前hang住，等待debug端发送的调试消息。仅支持直接执行存储过程的调试，不支持通过trigger调用执行的存储过程调试。

  ```
  gaussdb=# call test_debug(1);
  ```

  再起一个客户端，作为debug端，通过turn_on返回的数据，调用DBE_PLDEBUGGER.attach关联到该存储过程上进行调试。

  ```
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.attach('datanode',0);
   funcoid |  funcname  | lineno |              query
  ---------+------------+--------+----------------------------------
     16389 | test_debug |      3 |   INSERT INTO t1 (a) VALUES (x);
  (1 row)
  ```

  在执行attach的客户端调试，执行下一条statement。

  ```
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.next();
   funcoid |  funcname  | lineno |        query
  ---------+------------+--------+----------------------
     16389 | test_debug |      0 | [EXECUTION FINISHED]
  (1 row)
  ```

  在执行attach的客户端调试，可以执行以下变量操作。

  ```
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.info_locals(); --打印全部变量
   varname | vartype | value | package_name | isconst
  ---------+---------+-------+--------------+---------
   x       | int4    | 1     |              | f
  (1 row)
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.set_var('x', 2); --变量赋值
   set_var
  ---------
   t
  (1 row)
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.print_var('x'); --打印单个变量
   varname | vartype | value | package_name | isconst
  ---------+---------+-------+--------------+---------
   x       | int4    | 2     |              | f
  (1 row)
  ```

  直接执行完成当前正在调试的存储过程。

  ```
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.continue();
   funcoid |  funcname  | lineno |        query
  ---------+------------+--------+----------------------
     16389 | test_debug |      0 | [EXECUTION FINISHED]
  (1 row)
  ```

  当存储过程报错时，会出现以下提示，此时进入报错断住逻辑。

  ```
  gaussdb=# SELECT * FROM DBE_PLDEBUGGER.continue();
   funcoid |  funcname  | lineno |            query
  ---------+------------+--------+-------------------------------
     16389 | test_debug |      0 | [EXECUTION HAS ERROR OCCURRED!]
  (1 row)
  ```


版权所有 © 华为云计算技术有限公司

此时进入报错断住逻辑，可以调用error_info_locals,error_backtrace,error_end,print_var接口查看信息，其他接口不能再使用，需要使用error_end结束当前报错断住。

```
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.error_end();
 funcoid |  funcname  | lineno |      query
---------+------------+--------+------------------
   16389 | test_debug |      0 | [END HANG ERROR!]
(1 row)
```

直接退出当前正在调试的存储过程(报错断住逻辑不可用)，不执行尚未执行的语句。

```
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.abort();
 abort
-------
 t
(1 row)
```

client端查看代码信息并识别可以设置断点行号。

```
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.info_code(16389);
 lineno |                          query                          | canbreak
--------+---------------------------------------------------------+----------
        | CREATE OR REPLACE PROCEDURE public.test_debug( IN  x INT) | f
      1 | AS  DECLARE                                               | f
      2 | BEGIN                                                     | f
      3 |     INSERT INTO t1 (a) VALUES (x);                        | t
      4 |     DELETE FROM t1 WHERE a = x;                           | t
      5 | END;                                                      | f
      6 | /                                                         | f
(7 rows)
```

设置断点。

```
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.add_breakpoint(16389,4);
 breakpointno
--------------
            0
(1 row)
```

查看断点信息。

```
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.info_breakpoints();
 breakpointno | funcoid | lineno |              query               | enable
--------------+---------+--------+----------------------------------+--------
            0 |   16389 |      4 |     DELETE FROM t1 WHERE a = x; | t
(1 row)
```

执行至断点。

```
gaussdb=# SELECT * FROM DBE_PLDEBUGGER.continue();
 funcoid |  funcname  | lineno |              query
---------+------------+--------+----------------------------------
   16389 | test_debug |      4 |     DELETE FROM t1 WHERE a = x;
(1 row)
```

存储过程执行结束后，调试会自动退出，再进行调试需要重新attach关联。如果server端不需要继续调试，可执行turn_off关闭，或退出session。具体调试接口如表8-229所示。

表 8-229 DBE_PLDEBUGGER

接口名称	描述
DBE_PLDEBUGGER.turn_on	server端调用，标记存储过程可以调试，调用后执行该存储过程时会hang住等待调试信息。
DBE_PLDEBUGGER.turn_off	server端调用，标记存储过程关闭调试。

版权所有 © 华为云计算技术有限公司

接口名称	描述
DBE_PLDEBUGGER.local_debug_server_info	server端调用，打印本session内所有已turn_on的存储过程。
DBE_PLDEBUGGER.attach	debug端调用，关联到正在调试存储过程。
DBE_PLDEBUGGER.info_locals	debug端调用，打印正在调试的存储过程中的变量当前值。
DBE_PLDEBUGGER.next	debug端调用，单步执行。
DBE_PLDEBUGGER.continue	debug端调用，继续执行，直到断点或存储过程结束。
DBE_PLDEBUGGER.abort	debug端调用，停止调试，server端报错长跳转。
DBE_PLDEBUGGER.print_var	debug端调用，打印正在调试的存储过程中指定的变量当前值。
DBE_PLDEBUGGER.info_code	debug和server端都可以调用，打印指定存储过程的源语句和各行对应的行号。
DBE_PLDEBUGGER.step	debug端调用，单步进入执行。
DBE_PLDEBUGGER.add_breakpoint	debug端调用，新增断点。
DBE_PLDEBUGGER.delete_breakpoint	debug端调用，删除断点。
DBE_PLDEBUGGER.info_breakpoints	debug端调用，查看当前的所有断点。
DBE_PLDEBUGGER.backtrace	debug端调用，查看当前的调用栈。
DBE_PLDEBUGGER.enable_breakpoint	debug端调用，激活被禁用的断点。
DBE_PLDEBUGGER.disable_breakpoint	debug端调用，禁用已激活的断点。
DBE_PLDEBUGGER.finish	debug端调用，继续调试，直到断点或返回上一层调用栈。
DBE_PLDEBUGGER.set_var	debug端调用，为变量进行赋值操作。

8.4.1 DBE_PLDEBUGGER.turn_on

该函数用于标记某一存储过程为可调试，执行turn_on后server端可以执行该存储过程来进行调试，如表8-230所示。需要用户根据系统表PG_PROC手动获取存储过程oid，传入函数中。turn_on后本session内执行该存储过程会停在第一条sql前等待debug端

版权所有 © 华为云计算技术有限公司

的调试操作。该设置会在session断连后默认被清理掉。目前不支持对启用自治事务的存储过程/函数进行调试。同时turn_on的存储过程不超过100个。

函数原型为：

```
DBE_PLDEBUGGER.turn_on(Oid)
RETURN Record;
```

表 8-230 DBE_PLDEBUGGER.turn_on 入参和返回值列表

名称	类型	描述
func_oid	IN oid	函数oid。
nodename	OUT text	节点名称。
port	OUT integer	连接端口号。

8.4.2 DBE_PLDEBUGGER.turn_off

仅用于去掉当前session被turn_on添加的调试标记，返回值表示成功或失败，如表8-231所示。可通过DBE_PLDEBUGGER.local_debug_server_info查找已经turn_on的存储过程oid。

函数原型为：

```
DBE_PLDEBUGGER.turn_off(Oid)
RETURN boolean;
```

表 8-231 DBE_PLDEBUGGER.turn_off 入参和返回值列表

名称	类型	描述
func_oid	IN oid	函数oid。
turn_off	OUT boolean	表示turn off是否成功。

8.4.3 DBE_PLDEBUGGER.local_debug_server_info

用于查找当前连接中已经turn_on的存储过程oid。便于用户确认在调试哪些存储过程，需要通过funcoid和pg_proc配合使用，如表8-232所示。

表 8-232 DBE_PLDEBUGGER.local_debug_server_info 返回值列表

名称	类型	描述
nodename	OUT text	节点名称。
port	OUT bigint	端口号。
funcoid	OUT oid	存储过程oid。

版权所有 © 华为云计算技术有限公司

8.4.4 DBE_PLDEBUGGER.attach

server端执行存储过程，停在第一条语句前，等待debug端关联。debug端调用attach，传入nodename和port，关联到该存储过程上，如表8-233所示。

如果调试过程中报错，attach会自动失效；如果调试过程中attach到其他存储过程上，当前attach的调试也会失效；重复attach会导致当前存储过程断联。

表 8-233 DBE_PLDEBUGGER.attach 入参和返回值列表

名称	类型	描述
nodename	IN text	节点名称。
port	IN integer	连接端口号。
funcoid	OUT oid	函数id。
funcname	OUT text	函数名。
lineno	OUT integer	当前调试运行的下一行行号。
query	OUT text	当前调试的下一行函数源码。

8.4.5 DBE_PLDEBUGGER.info_locals

debug端调试过程中，调用info_locals，打印当前存储过程内变量。该函数入参frameno表示查询遍历的栈层数，支持无入参调用，缺省为查看最上层栈变量，如表8-234所示。

表 8-234 DBE_PLDEBUGGER.info_locals 入参和返回值列表

名称	类型	描述
frameno	IN integer（可选）	指定的栈层数，缺省为最顶层。
varname	OUT text	变量名。
vartype	OUT text	变量类型。
value	OUT text	变量值。
package_name	OUT text	变量对应的package名，非package时为空。
isconst	OUT boolean	是否为常量。

8.4.6 DBE_PLDEBUGGER.next

执行存储过程中当前的sql，返回执行的下一条的行数和对应query，如表8-235所示。

版权所有 © 华为云计算技术有限公司

表 8-235 DBE_PLDEBUGGER.next 返回值列表

名称	类型	描述
funcoid	OUT oid	函数id。
funcname	OUT text	函数名。
lineno	OUT integer	当前调试运行的下一行行号。
query	OUT text	当前调试的下一行函数源码。

8.4.7 DBE_PLDEBUGGER.continue

执行当前存储过程，直到下一个断点或结束，返回执行的下一条的行数和对应query，如表8-236所示。

函数原型为：

```
DBE_PLDEBUGGER.continue()
RETURN Record;
```

表 8-236 DBE_PLDEBUGGER.continue 返回值列表

名称	类型	描述
funcoid	OUT oid	函数id。
funcname	OUT text	函数名。
lineno	OUT integer	当前调试运行的下一行行号。
query	OUT text	当前调试的下一行函数源码。

8.4.8 DBE_PLDEBUGGER.abort

令server端执行的存储过程报错跳出。返回值表示是否成功发送abort，如表8-237所示。

函数原型为：

```
DBE_PLDEBUGGER.abort()
RETURN boolean;
```

表 8-237 DBE_PLDEBUGGER.abort 返回值列表

名称	类型	描述
abort	OUT boolean	表示是否成功。

版权所有 © 华为云计算技术有限公司

8.4.9 DBE_PLDEBUGGER.print_var

debug端调试过程中，调用print_var，打印当前存储过程内变量中指定的变量名及其取值，也可在报错断住时，使用此函数来打印变量。该函数入参frameno表示查询遍历的栈层数，支持不加入该参数调用，缺省为查看最上层栈变量，如表8-238所示。

表 8-238 DBE_PLDEBUGGER.print_var 入参和返回值列表

名称	类型	描述
var_name	IN text	变量。
frameno	IN integer（可选）	指定的栈层数，缺省为最顶层。
varname	OUT text	变量名。
vartype	OUT text	变量类型。
value	OUT text	变量值。
package_name	OUT text	变量对应的package名，预留使用，当前均为空。
isconst	OUT boolean	是否为常量。

8.4.10 DBE_PLDEBUGGER.info_code

debug端调试过程中，调用info_code，查看指定存储过程的源语句和各行对应的行号，行号从函数体开始，函数头部分行号为空，如表8-239所示。

表 8-239 DBE_PLDEBUGGER.info_code 入参和返回值列表

名称	类型	描述
funcoid	IN oid	函数id。
lineno	OUT integer	行号。
query	OUT text	源语句。
canbreak	OUT bool	当前行是否支持断点。

8.4.11 DBE_PLDEBUGGER.step

debug端调试过程中，如果当前执行的是一个存储过程，则进入该存储过程继续调试，返回该存储过程第一行的行号等信息，如果当前执行的不是存储过程，则和next行为一致，执行该sql后返回下一行的行号等信息，如表8-240所示。

版权所有 © 华为云计算技术有限公司

表 8-240 DBE_PLDEBUGGER.step 入参和返回值列表

名称	类型	描述
funcoid	OUT oid	函数id。
funcname	OUT text	函数名。
lineno	OUT integer	当前调试运行的下一行行号。
query	OUT text	当前调试的下一行函数源码。

8.4.12 DBE_PLDEBUGGER.add_breakpoint

debug端调试过程中，调用add_breakpoint增加新的断点。如果返回-1则说明指定的断点不合法，请参考canbreak字段确定合适的断点位置。进入内层函数前，外层函数无断点，则不支持在内层函数执行时，给外层函数添加断点，如表8-241所示。

表 8-241 DBE_PLDEBUGGER.add_breakpoint 入参和返回值列表

名称	类型	描述
funcoid	IN text	函数id。
lineno	IN integer	行号。
breakpointno	OUT integer	断点编号。

8.4.13 DBE_PLDEBUGGER.delete_breakpoint

debug端调试过程中，调用delete_breakpoint删除已有的断点，如表8-242所示。

表 8-242 DBE_PLDEBUGGER.delete_breakpoint 入参和返回值列表

名称	类型	描述
breakpointno	IN integer	断点编号。
result	OUT bool	是否成功。

8.4.14 DBE_PLDEBUGGER.info_breakpoints

debug端调试过程中，调用info_breakpoints，查看当前的函数断点，如表8-243所示。

版权所有 © 华为云计算技术有限公司

表 8-243 DBE_PLDEBUGGER.info_breakpoints 返回值列表

名称	类型	描述
breakpointno	OUT integer	断点编号。
funcoid	OUT oid	函数id。
lineno	OUT integer	行号。
query	OUT text	断点内容。
enable	OUT boolean	是否有效。

8.4.15 DBE_PLDEBUGGER.backtrace

debug端调试过程中，调用backtrace，查看当前的调用堆栈，如表8-244所示。

表 8-244 DBE_PLDEBUGGER.backtrace 返回值列表

名称	类型	描述
frameno	OUT integer	调用栈编号。
funcname	OUT text	函数名。
lineno	OUT integer	行号。
query	OUT text	断点内容。
funcoid	OUT oid	函数oid。

8.4.16 DBE_PLDEBUGGER.enable_breakpoint

debug端调试过程中，调用enable_breakpoint激活已被禁用的断点，如表8-245所示。

表 8-245 DBE_PLDEBUGGER.enable_breakpoint 入参和返回值列表

名称	类型	描述
breakpointno	IN integer	断点编号。
result	OUT bool	是否成功。

8.4.17 DBE_PLDEBUGGER.disable_breakpoint

debug端调试过程中，调用disable_breakpoint禁用已被激活的断点，如表8-246所示。

版权所有 © 华为云计算技术有限公司

表 8-246 DBE_PLDEBUGGER.disable_breakpoint 入参和返回值列表

名称	类型	描述
breakpointno	IN integer	断点编号。
result	OUT bool	是否成功。

8.4.18 DBE_PLDEBUGGER.finish

执行存储过程中当前的SQL直到下一个断点触发或执行到上层栈的下一行，如表8-247所示。

表 8-247 DBE_PLDEBUGGER.finish 入参和返回值列表

名称	类型	描述
funcoid	OUT oid	函数id。
funcname	OUT text	函数名。
lineno	OUT integer	当前调试运行的下一行行号。
query	OUT text	当前调试的下一行函数源码。

8.4.19 DBE_PLDEBUGGER.set_var

将指定的调试的存储过程中最上层栈上的变量修改为入参的取值。如果存储过程中包含同名的变量，set_var只支持第一个变量值的设置，如表8-248所示。

表 8-248 DBE_PLDEBUGGER.set_var 入参和返回值列表

名称	类型	描述
var_name	IN text	变量名。
value	IN text	修改值。
result	OUT boolean	是否成功。

8.4.20 DBE_PLDEBUGGER.error_backtrace

server端因为存储过程报错断住，可以在debug端调用error_backtrace，查看当前的函数调用堆栈（限制报错断住时使用），如表8-249所示。

版权所有 © 华为云计算技术有限公司

表 8-249 DBE_PLDEBUGGER.error_backtrace 返回值列表

名称	类型	描述
frameno	OUT integer	调用栈编号。
funcname	OUT text	函数名。
lineno	OUT integer	行号。
query	OUT text	断点内容。
funcoid	OUT oid	函数oid。

8.4.21 DBE_PLDEBUGGER.error_end

server端因为存储过程报错断住，可以在debug端调用error_end结束报错断住流程，结束调试流程，返回结束报错断住，停止调试（限制报错断住时使用），如表8-250所示。

表 8-250 DBE_PLDEBUGGER.error_end 返回值列表

名称	类型	描述
funcoid	OUT oid	函数id。
funcname	OUT text	函数名。
lineno	OUT integer	当前调试运行的下一行行号（固定值：0）。
query	OUT text	结束报错断住，停止调试。

8.4.22 DBE_PLDEBUGGER.error_info_locals

server端因为存储过程报错断住，可以在debug端调用error_info_locals，打印当前存储过程内变量。该函数入参frameno表示查询遍历的栈层数，支持无入参调用，缺省为查看最上层栈变量（限制报错断住时使用），如表8-251所示。

表 8-251 DBE_PLDEBUGGER.error_info_locals 返回值列表

名称	类型	描述
frameno	IN integer（可选）	指定的栈层数，缺省为最顶层。
varname	OUT text	变量名。
vartype	OUT text	变量类型。
value	OUT text	变量值。

版权所有 © 华为云计算技术有限公司

名称	类型	描述
package_name	OUT text	变量对应的package名，非package时为空。
isconst	OUT boolean	是否为常量。

8.5 DBE_PLDEVELOPER

DBE_PLDEVELOPER下系统表用于记录PL/SQL包、函数及存储过程编译过程中需要记录的信息。

8.5.1 DBE_PLDEVELOPER.GS_SOURCE

用于记录PL/SQL对象（存储过程、函数、包、包体）相关信息，如表8-252所示。

说明

- gs_source表中只记录用户定义的原始对象语句，即使用户使用了ALTER改变了创建的SCHEMA或者名字，gs_source表中的信息也不会发生变化，如果用户更改了对象的SCHEMA或者名字，用户在删除对象后，对象会从gs_source表中被删除。
- gs_source表中的owner表示创建的用户，不是用户创建存储过程或package时指定的用户。
- 设置GUC参数behavior_compat_options="skip_insert_gs_source"后，PL/SQL对象（存储过程、函数、包、包体）相关信息将不在gs_source中记录。
- gs_source权限与pg_catalog下的系统表保持一致（系统管理员与普通用户不允许DROP、ALTER、DELETE、UPDATE、INSERT，允许SELECT），利用GUC参数allow_system_table_mods控制操作gs_source的权限，此GUC参数默认为off，修改此GUC参数生效需要重启数据库。

表 8-252 DBE_PLDEVELOPER.GS_SOURCE 字段

名称	类型	描述
id	oid	对象id。
owner	bigint	对象创建用户id。
nspid	oid	对象的模式id。
name	name	对象名。
type	text	对象类型（procedure/function/package/package body）。
status	boolean	是否创建成功。
src	text	对象创建的原始语句。

8.6 DBE_COMPRESSION Schema

DBE_COMPRESSION是由ILM特性创建的一种模式，其中定义了根据输入的参数评估指定数据对象的采样压缩率或者获取指定行数据的压缩类型的系统函数。

版权所有 © 华为云计算技术有限公司

DBE_COMPRESSION支持的接口如表8-253所示，用户只有使用权限。具体说明请参见DBE_COMPRESSION。

表 8-253 DBE_COMPRESSION

接口名称	描述
DBE_COMPRESSION.GET_COMPRESSION_RATIO	根据输入参数评估指定数据对象的采样压缩率。
DBE_COMPRESSION.GET_COMPRESSION_TYPE	根据输入参数获取指定行数据的压缩类型。
DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO	根据输入参数评估指定数据对象的真实压缩率。
DBE_COMPRESSION.GET_HOLE_RATIO	根据输入参数评估指定跨页透明压缩数据对象的采样空洞率。
DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS	输出turbo_compress_shared_buffers的使用情况。
DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE	根据输入参数获取指定索引页的压缩类型。

8.7 DBE_HEAT_MAP Schema

DBE_HEAT_MAP是由ILM特性创建的一种模式，根据输入的参数返回目标数据块中行的最后修改时间等信息，用于直观浏览每一行被判定为冷、热行的依据。该接口属于运维类接口，不做可见性判断，即传入的ctid为已删除的行时，该接口依然会返回当前行在页面上最新的状态。DBE_HEAT_MAP支持的接口如表8-254所示，用户只有使用权限。跨页透明压缩不支持此模式。具体说明请参见DBE_HEAT_MAP。

表 8-254 DBE_HEAT_MAP

接口名称	描述
DBE_HEAT_MAP.ROW_HEAT_MAP	根据对象所属Schema、数据对象名称、数据对象分区名及ctid获取行的最后修改时间等信息。

8.8 DBE_ILM Schema

DBE_ILM是由ILM特性创建的一种模式，服务于ILM策略实施，实现ILM策略的评估以及压缩Job的停用接口。DBE_ILM支持的接口如表8-255所示，用户只有使用权限。具体说明请参见DBE_ILM。

版权所有 © 华为云计算技术有限公司

表 8-255 DBE_ILM

接口名称	描述
DBE_ILM.EXECUTE_ILM	根据参数执行对指定的数据和ILM策略进行评估，评估通过则会生成对应的压缩Job。
DBE_ILM.STOP_ILM	根据参数停止正在执行的压缩Job。

8.9 DBE_ILM_ADMIN Schema

DBE_ILM_ADMIN是由ILM特性创建的一种模式，服务于ILM策略实施，实现ADO的后台调度以及各个限流参数的控制。DBE_ILM_ADMIN支持的接口如表8-256所示，用户只有使用权限。具体说明请参见DBE_ILM_ADMIN。

表 8-256 DBE_ILM_ADMIN

接口名称	描述
DBE_ILM_ADMIN.CUSTOMIZE_ILM	根据输入参数定制ILM策略属性。
DBE_ILM_ADMIN.DISABLE_ILM	关闭后台调度。
DBE_ILM_ADMIN.ENABLE_ILM	开启后台调度。
DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY	创建库级压缩策略。
DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY	删除库级压缩策略。

8.10 PRVT_ILM Schema

PRVT_ILM接口为ILM特性内部使用接口，用户无法直接调用。本章节仅列举接口名称，不提供详细的接口原型描述。

表 8-257 PRVT_ILM

接口名称	描述
be_active_ado_window	维护窗口触发动作，刷新ILM window触发时间及动作。
be_create_ado_window_for_each_db	维护窗口触发动作，在实例的每个数据库中创建ILM window。
be_execute_ilm	ILM window执行动作，进行自动评估。
be_execute_ilm_dn	用于分布式DN上的自动评估。该操作会影响集群整体的ILM后台调度。

版权所有 © 华为云计算技术有限公司

接口名称	描述
flush_task_executestate	自动调度任务刷新task状态。
generate_taskoid	CN生成用于本次调度的taskid。
evaluate_obj_policy	触发调度后对数据对象进行评估。
change_be_ilm	DBE_ILM_ADMIN.ENABLE_ILM()和DISABLE_ILM()帮助函数。该操作会影响集群整体的ILM后台调度。
get_compression_ratio	DBE_COMPRESSION.GET_COMPRESSION_RATIO()帮助函数。
get_compression_type	DBE_COMPRESSION.GET_COMPRESSION_TYPE()帮助函数。
get_lastmodified_time	DBE_HEAT_MAP.ROW_HEAT_MAP()帮助函数。
ilm_job_action	压缩任务执行体，分布式DN使用。
delete_expired_ilm_log	清理过期日志，分布式使用。
delete_gs_ilm_task	操作特性相关系统表，分布式DN使用。
insert_gs_ilm_jobdetail	操作特性相关系统表，分布式DN使用。
insert_gs_ilm_task	操作特性相关系统表，分布式DN使用。
insert_gs_ilm_taskdetail	操作特性相关系统表，分布式DN使用。
update_gs_ilm_jobdetail	操作特性相关系统表，分布式DN使用。
update_gs_ilm_object	操作特性相关系统表，分布式DN使用。
update_gs_ilm_param	操作特性相关系统表，分布式DN使用。该操作会影响集群整体的ILM后台调度。
update_gs_ilm_task	操作特性相关系统表，分布式DN使用。
gs_ilm_ticker	维护窗口执行动作，执行一次打点，记录LSN与时间的映射关系。
compress_blocks	PRVT_ILM.ILM_JOB_ACTION()帮助函数。
compress_block_single	当前版本不可用。
get_job_status	DBE_ILM.STOP_ILM帮助函数。
insert_gs_ilm_param	操作特性相关系统表。
query_unfinishedjob_num	查询是否存在还没有结束的Job，包括初始和运行中的状态。
update_gs_ilm_object_flag2	操作特性相关系统表，分布式DN使用。
ilm_seq_nextval	用于获取ILM的下一个Sequence值。
ilm_seq_setval	用于设置ILM的当前Sequence值。

版权所有 © 华为云计算技术有限公司

接口名称	描述
check_compatibility	用于统一判断数据库兼容性模式的帮助函数。
set_pg_settings	用于设置数据库兼容性模式特性GUC参数的帮助函数。
check_seq	保证GSILMTASK_SEQ当前值大于GS_ILM_TASK表最大taskoid，分布式CN使用。
get_actual_compression_ratio	DBE_COMPRESSION.GET_ACTUAL_COMPRESSION_RATIO()帮助函数。
get_turbo_compress_hole	DBE_COMPRESSION.GET_HOLE_RATIO()帮助函数。
turbo_compress_buffer_stats	DBE_COMPRESSION.TURBO_COMPRESS_BUFFER_STATS()帮助函数。
get_index_compression_type	DBE_COMPRESSION.GET_INDEX_COMPRESSION_TYPE()帮助函数。
create_ilm_db_policy	DBE_ILM_ADMIN.CREATE_ILM_DB_POLICY()帮助函数。
delete_ilm_db_policy	DBE_ILM_ADMIN.DELETE_ILM_DB_POLICY()帮助函数。
compress_indexes	PRVT_ILM.ILM_JOB_ACTION()帮助函数。
eval_comp_relgrowth	当前版本不可用。
delete_gs_ilm_jobdetail	操作特性相关系统表，分布式DN使用。

版权所有 © 华为云计算技术有限公司

9 WDR 报告

WDR报表主要内容如表9-1所示。

表 9-1 WDR 报表主要内容

项目	描述
Database Stat	• 数据库维度性能统计信息：事务、读写、行活动、写冲突、死锁等。 • 集群范围报表，仅cluster模式下可查看此报表。
Load Profile	• 集群维度的性能统计信息：CPU时间、DB时间、逻辑读/物理读、I/O性能、登录登出、负载强度和负载性能表现等。 • 集群范围报表，仅cluster模式下可查看此报表。 • 仅非库级WDR报告可查看此报表。
Instance Efficiency Percentages	• 集群级或者节点缓冲命中率。 • 集群、节点范围报表，cluster模式和node模式下均可查看此报表。 • 仅非库级WDR报告可查看此报表。
Top 10 Events by Total Wait Time	• 最消耗时间的事件。 • 节点范围报表，仅node模式下可查看此报表。 • 仅非库级WDR报告可查看此报表。
Wait Classes by Total Wait Time	• 最消耗时间的等待时间分类。 • 节点范围报表，仅node模式下可查看此报表。 • 仅非库级WDR报告可查看此报表。
Host CPU	• 主机CPU消耗。 • 节点范围报表，仅node模式下可查看此报表。 • 仅非库级WDR报告可查看此报表。

版权所有 © 华为云计算技术有限公司

项目	描述
IO Profile	● 集群或者节点维度的I/O的使用情况。 ● 集群、节点范围报表，cluster模式和node模式下均可查看此报表。 ● 仅非库级WDR报告可查看此报表。
Memory Statistics	● 内核内存使用分布。 ● 节点范围报表，仅node模式下可查看此报表。 ● 仅非库级WDR报告可查看此报表。
Time Model	● 数据库各项操作的时间分布信息。 ● 节点范围报表，仅node模式下可查看此报表。 ● 仅非库级WDR报告可查看此报表。
SQL Statistics	● SQL语句各个维度性能统计，按以下维度排序展示：总时间、平均时间、CPU耗时、等锁时间、总内存使用、最大内存使用、平均内存使用、返回的行数、扫描的行数、执行次数、逻辑读、物理读。 ● 集群、节点范围报表，cluster模式和node模式下均可查看此报表。
Wait Events	● 节点级别的等待事件的统计信息。 ● 节点范围报表，仅node模式下可查看此报表。 ● 仅非库级WDR报告可查看此报表。 关键信息请参见《参考》中“系统表和系统视图 > 系统视图 > 其他系统视图 > PG_THREAD_WAIT_STATUS”章节中“等待状态列表”、“轻量级锁等待事件列表”、“I/O等待事件列表”和“事务锁等待事件列表”。
Cache IO Stats	● 用户的表、索引的I/O的统计信息。 ● 集群、节点范围报表，cluster模式和node模式下均可查看此报表。 ● 仅非库级WDR报告可查看此报表。
Utility status	● 复制槽和后台checkpoint的状态信息。 ● 节点范围报表，仅node模式下可查看此报表。 ● 仅非库级WDR报告可查看此报表。
Object stats	● 表、索引维度的性能统计信息。 ● 集群、节点范围报表，cluster模式和node模式下均可查看此报表。
Configuration settings	● 节点GUC参数配置信息：参数名称、参数当前值和默认值等。 ● 节点范围报表，仅node模式下可查看此报表。 ● 仅非库级WDR报告可查看此报表。

版权所有 © 华为云计算技术有限公司

项目	描述
SQL Detail	• SQL语句文本详情。 • SQL语句对应的执行计划文本详情。 • 集群、节点范围报表，cluster模式和node模式下均可查看此报表。

9.1 Database Stat

Database Stat列名称及描述如表9-2所示。

表 9-2 Database Stat 报表主要内容

列名称	描述
DB Name	数据库名称。
Backends	连接到该数据库的后端数。
Xact Commit	此数据库中已经提交的事务数。
Xact Rollback	此数据库中已经回滚的事务数。
Blks Read	在这个数据库中读取磁盘块的数量。
Blks Hit	高速缓存中已经发现磁盘块的次数。
Tuple Returned	顺序扫描的行数。
Tuple Fetched	随机扫描的行数。
Tuple Inserted	通过数据库查询插入的行数。
Tuple Updated	通过数据库查询更新的行数。
Tup Deleted	通过数据库查询删除的行数。
Conflicts	由于数据库恢复冲突取消的查询数量。
Temp Files	通过数据库查询创建的临时文件数量。
Temp Bytes	通过数据库查询写入临时文件的数据总量。
Deadlocks	该数据库中检索的死锁数。
Blk Read Time	通过数据库后端读取数据文件块花费的时间，以毫秒计算。

版权所有 © 华为云计算技术有限公司

列名称	描述
Blk Write Time	通过数据库后端写入数据文件块花费的时间，以毫秒计算。
Stats Reset	重置当前状态统计的时间。

9.2 Load Profile

Load Profile指标名称及描述如表9-3所示。

表 9-3 Load Profile 报表主要内容

指标名称	描述
DB Time(us)	作业运行的elapsed time总和。
CPU Time(us)	作业运行的CPU时间总和。
Redo size(blocks)	产生的WAL的大小（块数）。
Logical read (blocks)	表或者索引文件的逻辑读（块数）。
Physical read (blocks)	表或者索引的物理读（块数）。
Physical write (blocks)	表或者索引的物理写（块数）。
Read IO requests	表或者索引的读次数。
Write IO requests	表或者索引的写次数。
Read IO (MB)	表或者索引的读大小（MB）。
Write IO (MB)	表或者索引的写大小（MB）。
Logins	登录次数。
Executes (SQL)	SQL执行次数。
Rollbacks	回滚事务数。
Transactions	事务数。
n_soft_parse	软解析次数。
n_hard_parse	硬解析次数。
SQL response time P95(us)	95%SQL的响应时间（微秒）。
SQL response time P80(us)	80%SQL的响应时间（微秒）。

版权所有 © 华为云计算技术有限公司

9.3 Instance Efficiency Percentages

Instance Efficiency Percentages指标名称及描述如表9-4所示。

表 9-4 Instance Efficiency Percentages 报表主要内容

指标名称	描述
Buffer Hit %	Buffer Pool命中率。
Effective CPU %	CPU time占DB time的比例。
WalWrite NoWait %	访问WAL Buffer的event次数占总wait event的比例。
Soft Parse %	软解析次数占总解析次数的比例。
Non-Parse CPU %	非parse时间占执行总时间的比例。

9.4 Top 10 Events by Total Wait Time

Top 10 Events by Total Wait Time列名称及描述如表9-5所示。

表 9-5 Top 10 Events by Total Wait Time 报表主要内容

列名称	描述
Event	Wait Event名称。
Waits	wait次数。
Total Wait Time(us)	总wait时间（微秒）。
Avg Wait Time(us)	平均wait时间（微秒）。
Type	Wait Event类别。

9.5 Wait Classes by Total Wait Time

Wait Classes by Total Wait Time列名称及描述如表9-6所示。

版权所有 © 华为云计算技术有限公司

表 9-6 Wait Classes by Total Wait Time 报表主要内容

列名称	描述
Type	Wait Event类别名称： • STATUS。 • LWLOCK_EVENT。 • LOCK_EVENT。 • IO_EVENT。
Waits	Wait次数。
Total Wait Time(us)	总Wait时间（微秒）。
Avg Wait Time(us)	平均Wait时间（微秒）。

9.6 Host CPU

Host CPU列名称及描述如表9-7所示。

表 9-7 Host CPU 报表主要内容

列名称	描述
Cpus	CPU数量。
Cores	CPU核数。
Sockets	CPU Sockets数量。
Load Average Begin	开始snapshot的Load Average值。
Load Average End	结束snapshot的Load Average值。
%User	用户态在CPU时间上的占比。
%System	内核态在CPU时间上的占比。
%WIO	Wait IO在CPU时间上的占比。
%Idle	空闲时间在CPU时间上的占比。

9.7 IO Profile

IO Profile指标名称及描述如表9-8所示。

版权所有 © 华为云计算技术有限公司

表 9-8 IO Profile 指标表主要内容

指标名称	描述
Database requests	Database IO次数。
Database (MB)	Database IO数据量。
Database (blocks)	Database IO数据块。
Redo requests	Redo IO次数。
Redo (MB)	Redo IO量。

9.8 Memory Statistics

Memory Statistics指标名称及描述如表9-9所示。

表 9-9 Memory Statistics 报表主要内容

指标名称	描述
shared_used_memory	已经使用共享内存大小（MB）。
max_shared_memory	最大共享内存（MB）。
process_used_memory	进程已经使用内存（MB）。
max_process_memory	最大进程内存（MB）。

9.9 Time Model

Time Model名称及描述如表9-10所示。

表 9-10 Time Model 报表主要内容

名称	描述
DB_TIME	所有线程端到端的时间消耗总和（单位：微秒）。
EXECUTION_TIME	消耗在执行器上的时间总和（单位：微秒）。

版权所有 © 华为云计算技术有限公司

名称	描述
PL_EXECUTION_TIME	消耗在PL/SQL执行上的时间总和（单位：微秒）。
CPU_TIME	所有线程CPU时间消耗总和（单位：微秒）。
PLAN_TIME	消耗在执行计划生成上的时间总和（单位：微秒）。
REWRITE_TIME	消耗在查询重写上的时间总和（单位：微秒）。
PL_COMPILATION_TIME	消耗在SQL编译上的时间总和（单位：微秒）。
PARSE_TIME	消耗在SQL解析上的时间总和（单位：微秒）。
NET_SEND_TIME	消耗在网络发送上的时间总和（单位：微秒）。
DATA_IO_TIME	消耗在数据读写上的时间总和（单位：微秒）。

9.10 SQL Statistics

SQL Statistics列名称及描述如表9-11所示。

表 9-11 SQL Statistics 报表主要内容

列名称	描述
Unique SQL Id	归一化的SQL id。
DB Name	数据库名称。
Parent Unique SQL Id	父语句的归一化SQL id，对于非存储过程子语句的场景，该值为0。
Node Name	节点名称。
User Name	用户名称。
Avg Tuples Fetched	平均随机扫描行。
Avg Tuples Returned	平均顺序扫描行。
Calls	调用次数。
Total Elapsed Time(us)	总执行时间（us）。

版权所有 © 华为云计算技术有限公司

列名称	描述
Avg Elapsed Time(us)	平均执行时间（us）。
Total Used Memory(MB)	总内存使用量（MB）。
Max Used Memory(MB)	单语句最大内存使用量（MB）。
Min Used Memory(MB)	单语句最小内存使用量（MB）。
Returned Rows	SELECT返回行数。
Avg Returned Rows	平均SELECT返回行数。
Tuples Affected	Insert/Update/Delete行数。
Avg Tuples Affected	平均Insert/Update/Delete行数。
Logical Read	Buffer逻辑读次数。
Avg Logical Read	平均Buffer逻辑读次数。
Physical Read	Buffer物理读次数。
Avg Physical Read	平均Buffer物理读次数。
CPU Time(us)	CPU时间（us）。
Avg CPU Time(us)	平均CPU时间（us）。
Data IO Time(us)	IO上的时间花费（us）。
Avg Data IO Time(us)	平均IO上的时间花费（us）。
Lock Wait Time(us)	SQL锁等待时间（us）。
Avg Lock Wait Time(us)	平均每条SQL锁等待时间（us）。
Sort Count	排序执行的次数。
Sort Time(us)	排序执行的时间（us）。

版权所有 © 华为云计算技术有限公司

列名称	描述
Sort Mem Used(KB)	排序过程中使用work memory的大小（KB）。
Sort Spill Count	排序过程中，若发生落盘，写文件的次数。
Sort Spill Size(KB)	排序过程中，若发生落盘，使用的文件大小（KB）。
Hash Count	hash执行的次数。
Hash Time(us)	hash执行的时间（us）。
Hash Mem Used(KB)	hash过程中使用的work memory大小（KB）。
Hash Spill Count	hash过程中，若发生落盘，写文件的次数。
Hash Spill Size(KB)	hash过程中，若发生落盘，使用的文件大小（KB）。
SQL Text	归一化SQL字符串。
SQL Hash	SQL语句对应的hash值。
Plan Hash	SQL语句的执行计划对应的hash值。

9.11 Wait Events

Wait Events列名称及描述如表9-12所示。

表 9-12 Wait Events 报表主要内容

列名称	描述
Type	Wait Event类别名称： • STATUS。 • LWLOCK_EVENT。 • LOCK_EVENT。 • IO_EVENT。
Event	Wait Event名称。
Total Wait Time (us)	总Wait时间（us）。
Waits	总Wait次数。

版权所有 © 华为云计算技术有限公司

列名称	描述
Failed Waits	Wait失败次数。
Avg Wait Time (us)	平均Wait时间（us）。
Max Wait Time (us)	最大Wait时间（us）。

9.12 Cache IO Stats

Cache IO Stats包含User table IO activity部分和User index IOactivity部分，两部分表格列名称及描述如表9-13及表9-14所示。

User table IO activity

表 9-13 User table IO activity 表格字段

列名称	描述
DB Name	Database名称。
Schema Name	Schema名称。
Table Name	Table名称。
%Heap Blks Hit Ratio	此表Buffer Pool的命中率。
Heap Blks Read	该表中读取的磁盘块数。
Heap Blks Hit	此表缓存命中数。
Idx Blks Read	表中所有索引读取的磁盘块数。
Idx Blks Hit	表中所有索引命中缓存数。
Toast Blks Read	此表中TOAST表读取的磁盘块数（如果存在）。
Toast Blks Hit	此表中TOAST表命中缓冲区数（如果存在）。
Tidx Blks Read	此表中TOAST表索引读取的磁盘块数（如果存在）。

版权所有 © 华为云计算技术有限公司

列名称	描述
Tidx Blks Hit	此表中TOAST表索引命中缓冲区数（如果存在）。

User index IO activity

表 9-14 User index IO activity 表格字段

列名称	描述
DB Name	Database名称。
Schema Name	Schema名称。
Table Name	Table名称。
Index Name	Index名称。
%Idx Blks Hit Ratio	Index的命中率。
Idx Blks Read	所有索引读取的磁盘块数。
Idx Blks Hit	所有索引命中缓存数。

9.13 Utility status

Utility status包含Replication slot和Replication stat两张表，列名称及描述如表9-15及表9-16所示。

Replication slot

表 9-15 Replication slot 报表主要内容

列名称	描述
Slot Name	复制节点名。
Slot Type	复制节点类型。
DB Name	复制节点数据库名称。
Active	复制节点状态。
Xmin	复制节点事务标识。

版权所有 © 华为云计算技术有限公司

列名称	描述
Restart Lsn	复制节点的Xlog文件信息。
Dummy Standby	复制节点假备。

Replication stat

表 9-16 Replication stat 报表主要内容

列名称	描述
Thread Id	线程的PID。
Usesys Id	用户系统ID。
Usename	用户名称。
Application Name	应用程序。
Client Addr	客户端地址。
Client Hostname	客户端主机名。
Client Port	客户端端口。
Backend Start	程序起始时间。
State	日志复制状态。
Sender Sent Location	发送端发送日志位置。
Receiver Write Location	接收端write日志位置。
Receiver Flush Location	接收端flush日志位置。
Receiver Replay Location	接收端replay日志位置。
Sync Priority	同步优先级。
Sync State	同步状态。

9.14 Object Stats

Object Stats包含User Tables stats、User index stats和Bad lock stats三张表，列名称及描述如下所示。

版权所有 © 华为云计算技术有限公司

User Tables stats

表 9-17 User Tables stats 报表主要内容

列名称	描述
DB Name	Database名称。
Schema	Schema名称。
Relname	Relation名称。
Seq Scan	此表发起的顺序扫描数。
Seq Tup Read	顺序扫描抓取的活跃行数。
Index Scan	此表发起的索引扫描数。
Index Tup Fetch	索引扫描抓取的活跃行数。
Tuple Insert	插入行数。
Tuple Update	更新行数。
Tuple Delete	删除行数。
Tuple Hot Update	HOT更新行数（即没有更新所需的单独索引）。
Live Tuple	估计活跃行数。
Dead Tuple	估计不活跃行数。
Last Vacuum	最后一次该表是手动清理的（不计算VACUUM FULL）时间。
Last Autovacuum	上次被autovacuum守护线程清理的时间。
Last Analyze	上次手动分析这个表的时间。
Last Autoanalyze	上次被autovacuum守护线程分析的时间。
Vacuum Count	该表被手动清理的次数（不计算VACUUM FULL）。
Autovacuum Count	该表被autovacuum清理的次数。
Analyze Count	该表被手动分析的次数。
Autoanalyze Count	该表被autovacuum守护线程分析的次数。

版权所有 © 华为云计算技术有限公司

User index stats

表 9-18 User index stats 报表主要内容

列名称	描述
DB Name	Database名称。
Schema	Schema名称。
Relname	Relation名称。
Index Relname	Index名称。
Index Scan	索引上开始的索引扫描数。
Index Tuple Read	通过索引上扫描返回的索引项数。
Index Tuple Fetch	通过使用索引的简单索引扫描抓取的表行数。

Bad lock stats

表 9-19 Bad lock stats 报表主要内容

列名称	描述
DB Id	数据库的OID。
Tablespace Id	表空间的OID。
Relfilenode	文件对象ID。
Fork Number	文件类型。
Error Count	失败计数。
First Time	第一次发生时间。
Last Time	最近一次发生时间。

9.15 Configuration settings

Configuration settings列名称及描述如下表9-20所示。

表 9-20 Configuration settings 报表主要内容

列名称	描述
Name	GUC参数名称。

版权所有 © 华为云计算技术有限公司

列名称	描述
Abstract	GUC参数描述。
Type	数据类型。
Current Value	当前值。
Min Value	合法最小值。
Max Value	合法最大值。
Category	GUC参数类别。
Enum Values	如果是枚举值，列举所有枚举值。
Default Value	数据库启动时参数默认值。
Reset Value	数据库重置时参数默认值。

9.16 SQL Detail

SQL Detail列名称及描述如下表9-21所示。

表 9-21 SQL Detail 报表主要内容

列名称	描述
Unique SQL Id	归一化SQL id。
Parent Unique SQL Id	父语句的归一化SQL id，单独执行的语句该值为0。
DB Name	数据库名称，若生成快照时执行query语句的数据库已不存在，则在报告中显示为*REMOVED_DB*。
User Name	用户名称，若生成快照时执行query语句的用户已不存在，则在报告中显示为*REMOVED_USER*。
Node Name	节点名称。Node模式下不显示该字段。
SQL Hash	SQL语句对应的hash值。
Plan Hash	SQL语句的执行计划对应的hash值。
SQL Text	归一化SQL文本。
Plan Text	SQL语句的执行计划文本。

版权所有 © 华为云计算技术有限公司

10 ASP 报告

表 10-1 Report Header 信息

列名称	描述
Start Time	开始时间。
End Time	结束时间。
Slot Count	指标展示时分割的时间段的个数。
Elapsed Time	生成报告的时间段。
Sample Count	生成ASP的总样本数。
Slice Count	总采样次数。
Average Active Sessions	平均活跃会话数。
Boundary Time	内存中样本和磁盘样本的分割时间点。
Data Source	生成ASP的样本中多少来自内存，多少来自磁盘。

表 10-2 System Load 信息

列名称	描述
Host Node Name	节点名称。
CPUs	CPU核心总量。
Cores	单个CPU核心数量。
Sockets	CPU插槽数量。
Physical Memory	内存大小。
GaussDB Version	GaussDB版本。

版权所有 © 华为云计算技术有限公司

表 10-3 Top Client 信息

列名称	描述
Client Addr	在抽样活跃会话中占高百分比的客户端地址。在GS_ASP表中相关字段有client_hostname、client_port，为空时表示通过服务器机器上Unix套接字连接客户端或者这是内部线程。 **说明** 客户端（Client）是指与数据库建立连接与会话，与服务器进程交互的应用程序或工具。
User Id	用户id。
Application Name	客户端连接程序名称。 **说明** 除已屏蔽的后台线程外，可能存在的后台进程或客户端（内核内）有Autovacuum，StandByRead， Asp，gsql ，Archiver, WalReader，WalSender等，可能存在的分布式相关进程有GTM，CN，DN等，可能存在的外部（内核外）客户端进程有JDBC，ODBC，PSYCOPG2，DRS，DBMind等。
%Activity	活跃百分比（活跃会话总数/样本数量）。
Avg Active Sessions	平均活跃数量（活跃会话总数/切面个数）。

表 10-4 Active sessions 信息

列名称	描述
sample_time（x_axis）	采样时间。
count（y_axis）	采样事件数量。

表 10-5 Top Events Summary 信息

列名	描述
Wait Status	wait status名称。
Wait Status Count	wait status数量。
%Wait Status	wait status百分比（wait status数量/样本数量）。
Avg Active Session	wait status平均活跃会话数（wait status数量/切面数量）。
Event	wait event名称。

版权所有 © 华为云计算技术有限公司

列名	描述
Event Count	wait event数量。

表 10-6 Top Events Graph 信息

列名	描述
sample_time（x_axis）	采样时间。
count（y_axis）	采样事件数量。

表 10-7 Top Sessions with Top event 信息

列名	描述
Session Id	会话ID。
Session Start	会话开始事件。
Session Count	活跃会话数量。
%Activity	活跃百分比（活跃会话总数/样本数量）。
User Id	用户ID。
Application Name	连接程序名称。
Client Addr	客户端地址。
Client Port	客户端端口信息。
Event	活跃会话事件。
Event Count	该会话的wait event的数量。
%Event	会话事件百分比（事件数量/样本数量）。

表 10-8 Top Sessions with Top event split by slots 信息

列名	描述
Slot time	分片时间。
Session Id	会话ID。
Session Start	会话开始事件。
Session Count	会话事件数量。

版权所有 © 华为云计算技术有限公司

列名	描述
%Activity	活跃百分比（活跃会话总数/样本数量）。
User Id	用户ID。
Application Name	连接程序名称。
Client Addr	客户端地址。
Client Port	客户端端口信息。
Event	活跃会话事件。
Event Count	该会话的wait event的数量。
%Event	会话事件百分比（事件数量/样本数量）。

表 10-9 Top Sessions with Top SQL 信息

列名	描述
Session Id	会话ID。
Session Start	会话开始事件。
Session Count	会话事件数量。
%Activity	活跃百分比。
User Id	用户ID。
Application Name	连接程序名称。
Client Addr	客户端地址。
Client Port	客户端端口信息。
SQL Id	sqlid。
Parent SQL Id	调用本条语句的父语句SQL Id。
Node Id	节点ID。
SQL Text	SQL语句。

表 10-10 Top Sessions with Top SQL split by slots 信息

列名	描述
Slot Time	分片时间。
Session Id	会话ID。

版权所有 © 华为云计算技术有限公司

列名	描述
Session Start	会话开始事件。
Session Count	会话事件数量。
%Activity	活跃百分比（活跃会话总数/样本数量）。
User Id	用户ID。
Application Name	连接程序名称。
Client Addr	客户端地址。
Client Port	客户端端口信息。
SQL Id	SQL id。
Parent SQL Id	调用本条语句的父语句SQL Id。
Node Id	节点ID。
SQLText	SQL语句。

表 10-11 Top final blocking sessions with Top event split by slots 信息

列名	描述
Slot Time	分片时间。
Final Blocking Session Id	阻塞会话ID。
Session Start	持锁会话开始时间。
Blocked count	被该会话阻塞的会话的数量。
Blocked%Activity	阻塞会话占活跃会话百分比（阻塞会话总数/样本数量）。
Event Caused	final Blocking Session等待事件。
Event Caused Count	final Blocking Session等待事件的数量。
%Event	等待事件百分比（等待事件数量/样本数量）。
PID	线程ID。
User Id	用户ID。
Application Name	连接程序名称。
Client Addr	客户端地址。
Client Port	客户端端口信息。
Current Xid	当前事务的ID。

版权所有 © 华为云计算技术有限公司

列名	描述
Top Xid	顶层事务的ID。
Xact Start Time	持锁事务开始时间。
Global Session Id	全局会话ID。

表 10-12 Top blocked sessions with Top SQL split by slots 信息

列名	描述
Slot Time	分片时间。
Lock Tag	锁tag。
Lock Mode	锁模式。
Waiting Session Count	等待会话数。
Waiting Session Id	等锁会话ID。
Session Start	等锁会话开始时间。
Session Count	等锁会话数量。
Block Session Id	阻塞会话ID。
Final Block Session Id	阻塞队列头部会话ID。
PID	线程ID。
User Id	用户ID。
Application Name	连接程序名称。
Client Addr	客户端地址。
Client Port	客户端端口信息。
Current Xid	当前事务的ID。
Top Xid	顶层事务的ID。
Xact Start Time	持锁事务开始时间。
Global Session Id	全局会话ID。
SQL Id	SQL id。
Parent SQL Id	调用本条语句的父语句SQL Id。
Query Start	SQL开始的时间。
SQL Count	SQL的数量。
Node Id	节点ID。

版权所有 © 华为云计算技术有限公司

列名	描述
SQL Text	SQL语句。

表 10-13 Top SQL with Top event 信息

列名	描述
SQL Id	SQL id。
Parent SQL Id	调用本条语句的父语句SQL Id。
Node Id	节点ID。
User Id	用户ID。
Count	会话事件数量。
%Activity	活跃百分比（SQL会话总数/样本数量）。
Event	活跃会话事件。
Event Count	wait event数量。
%Event	会话事件百分比（SQL事件数量/样本数量）。
SQL Text	SQL语句。

表 10-14 Top SQL with Top event split by slots 信息

列名	描述
Slot Time	分片时间。
SQL Id	SQL id。
Parent SQL Id	调用本条语句的父语句SQL Id。
Node Id	节点ID。
User Id	用户ID。
Count	会话事件数量。
%Activity	活跃百分比（SQL会话总数/样本数量）。
Event	活跃会话事件。
Event Count	wait event数量。
%Event	会话事件百分比（SQL事件数量/样本数量）。

版权所有 © 华为云计算技术有限公司

列名	描述
SQL Text	SQL语句。

表 10-15 Top SQL Text 信息

列名	描述
SQL Id	SQL id。
Parent SQL Id	调用本条语句的父语句SQL Id。
Node Id	节点ID。
User Id	用户ID。
SQL Text	SQL语句。

版权所有 © 华为云计算技术有限公司