

云数据库 GaussDB

兼容性参考（分布式）

文档版本 01
发布日期 2025-09-09



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目 录

1 GaussDB 数据库兼容性概述.....	1
2 Oracle 兼容性说明.....	2
2.1 Oracle 数据库兼容性概述.....	2
2.2 SQL 的基本元素.....	2
2.2.1 数据类型.....	2
2.2.2 数据类型比较规则.....	7
2.2.3 字面量.....	8
2.2.4 格式模型.....	8
2.2.5 空值.....	9
2.2.6 注释.....	9
2.2.7 数据库对象.....	10
2.2.8 数据库对象名称和限定符.....	12
2.2.9 SQL 语句中的引用架构对象和部件的语法.....	12
2.3 伪列.....	12
2.4 操作符.....	13
2.5 表达式.....	16
2.6 条件.....	17
2.7 驱动.....	18
2.7.1 JDBC.....	18
2.7.1.1 Array.....	18
2.7.1.2 Struct.....	31
2.8 常见的 SQL DDL 子句.....	41
2.9 SQL 查询和子查询.....	47
2.10 PL/SQL 语言.....	48
2.10.1 PL/SQL 基本语法.....	48
2.10.2 数据类型兼容性.....	51
2.10.3 控制语句.....	51
2.10.4 集合和 Record.....	53
2.10.5 静态 SQL.....	58
2.10.6 动态 SQL.....	62
2.10.7 Trigger.....	62
2.11 系统函数.....	71
2.11.1 单行函数.....	71

2.11.2 其它函数.....	88
2.12 系统视图.....	90
2.13 高级包.....	97
3 MySQL 兼容性说明.....	149
3.1 MySQL 数据库兼容性概述.....	149
3.2 MySQL 兼容性 M-Compatibility 模式.....	151
3.2.1 数据类型.....	151
3.2.1.1 数值数据类型.....	151
3.2.1.2 日期与时间数据类型.....	153
3.2.1.3 字符串数据类型.....	155
3.2.1.4 二进制数据类型.....	157
3.2.1.5 数据类型支持的属性.....	161
3.2.1.6 数据类型转换.....	162
3.2.2 系统函数.....	166
3.2.2.1 系统函数兼容性概述.....	166
3.2.2.2 流程控制函数.....	167
3.2.2.3 日期和时间函数.....	167
3.2.2.4 字符串函数.....	172
3.2.2.5 强制转换函数.....	176
3.2.2.6 加密函数.....	177
3.2.2.7 比较函数.....	178
3.2.2.8 聚合函数.....	179
3.2.2.9 数字操作函数.....	181
3.2.2.10 其他函数.....	182
3.2.3 操作符.....	183
3.2.4 字符集.....	192
3.2.5 排序规则.....	193
3.2.6 事务.....	194
3.2.7 SQL.....	197
3.2.7.1 关键字.....	197
3.2.7.2 标识符.....	198
3.2.7.3 DDL.....	200
3.2.7.4 DML.....	216
3.2.7.5 DCL.....	241
3.2.7.6 其他语句.....	241
3.2.7.7 用户与权限.....	243
3.2.7.8 系统表和系统视图.....	248
3.2.8 驱动.....	252
3.2.8.1 ODBC.....	252
3.2.8.1.1 ODBC 接口参考.....	252
3.2.8.2 JDBC.....	253
3.3 MySQL 兼容性 MYSQL 模式.....	253

3.3.1 数据类型.....	254
3.3.1.1 数值数据类型.....	254
3.3.1.2 日期与时间数据类型.....	260
3.3.1.3 字符串数据类型.....	271
3.3.1.4 二进制数据类型.....	274
3.3.1.5 JSON 数据类型.....	277
3.3.1.6 数据类型支持的属性.....	277
3.3.1.7 数据类型转换.....	277
3.3.2 系统函数.....	280
3.3.2.1 流量控制函数.....	280
3.3.2.2 日期和时间函数.....	282
3.3.2.3 字符串函数.....	292
3.3.2.4 强制转换函数.....	296
3.3.2.5 加密函数.....	296
3.3.2.6 JSON 函数.....	297
3.3.2.7 聚合函数.....	299
3.3.2.8 数字操作函数.....	300
3.3.2.9 其他函数.....	301
3.3.3 操作符.....	301
3.3.4 字符集.....	302
3.3.5 排序规则.....	303
3.3.6 SQL.....	303
3.3.6.1 DDL.....	304
3.3.6.2 DML.....	312
3.3.6.3 DCL.....	323
3.3.7 驱动.....	323
3.3.7.1 JDBC.....	324
3.3.7.1.1 JDBC 接口参考.....	324

1 GaussDB 数据库兼容性概述

在GaussDB中，创建数据库可以选择不同的兼容性模式，如ORA、MYSQL、TD、PG、M，分别代表Oracle兼容性，MySQL兼容性MYSQL模式、Teradata兼容性、PostgreSQL兼容性和MySQL兼容性M-Compatibility模式。不同兼容性会直接影响SQL语法、数据类型、系统函数、存储过程等行为，部分兼容性接口可能仅在相应的兼容性下支持。

Oracle兼容性、MySQL兼容性M-Compatibility模式、MySQL兼容性MYSQL模式的兼容性说明可参考[Oracle兼容性说明](#)、[MySQL兼容性M-Compatibility模式](#)、[MySQL兼容性MYSQL模式](#)等章节内容，用户可根据实际需要，选择合适的兼容性。

2 Oracle 兼容性说明

2.1 Oracle 数据库兼容性概述

GaussDB数据库在基本功能（数据类型、SQL、数据库对象等）和PL/SQL方面与Oracle数据库基本兼容。但是由于架构设计方面的差异，还是存在一些不兼容的项。

本章节主要介绍GaussDB数据库的Oracle兼容模式505.2.1版本与Oracle数据库19c版本的兼容性对比信息。

2.2 SQL 的基本元素

2.2.1 数据类型

表 2-1 数值类型

序号	Oracle数据库	GaussDB数据库	差异
1	NUMBER [(p[,s])]	支持，有差异	精度和用法存在差异。 - NUMBER带参数时，GaussDB的精度p与标度s的最大边界值比Oracle更大。 - NUMBER不带参数时，GaussDB的精度p的默认值远大于带参数时的最大边界值；而在Oracle中，精度p的默认值等于带参数时的最大边界值。 - GaussDB不支持标度s为负值；在Oracle中，标度s为负值时会精确到相应的整数位。
2	FLOAT [(p)]	支持	-
3	BINARY_FLOAT	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
4	BINARY_DOUBLE	支持	-

表 2-2 日期时间类型

序号	Oracle数据库	GaussDB数据库	差异
1	DATE	支持，有差异	精度有差异，GaussDB支持的公元时间相较于Oracle范围更大。
2	TIMESTAMP [(fractional_seconds_precision)]	支持，有差异	-
3	TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE	支持，有差异	GaussDB的timestampztz等价于Oracle的timestampwithlocaltimezone，缺少Oracle对应的timestampztz类型。 时区更新：部分国家或地区经常会更新时区信息，数据库系统也因此常常需要同步修改时区文件以确保时间内容的正确性。 GaussDB时区类型目前只涉及timestamp with timezone，当新的时区文件生效时，不会对已有的数据进行变更，新数据会随时区文件信息进行同步调整。与Oracle的同类型数据能力存在差异。
4	TIMESTAMP [(fractional_seconds_precision)] WITH LOCAL TIME ZONE	不支持	-
5	INTERVAL YEAR [(year_precision)] TO MONTH	支持	-
6	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]	支持	-

 说明

- ORA兼容模式下，DATE类型被替换为TIMESTAMP(0) WITHOUT TIME ZONE，差异同TIMESTAMP(0) WITHOUT TIME ZONE。
- 对于TIMESTAMP [(fractional_seconds_precision)] / TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE与Oracle存在以下差异。
 - fractional_seconds_precision GaussDB支持的精度范围为0~6，Oracle支持的精度范围为0~9。
 - GaussDB通过DateStyle设置日期和时间值的显示格式，以及有歧义的输入值的解析规则。具体请参见《开发指南》中“SQL参考 > 数据类型 > 日期/时间类型”章节中日期输入的说明介绍。Oracle在一般情况下的输入格式校验及输出显示由NLS_TIMESTAMP_FORMAT/NLS_TIMESTAMP_TZ_FORMAT参数进行控制。
 - 对于秒的小数部分的显示，GaussDB默认去除结尾的零，Oracle依据格式化参数中的设置（FF/FF1-FF9）进行显示控制。如 '2017-09-01 10:32:19.212000'，GaussDB显示为 '2017-09-01 10:32:19.212'，Oracle在format参数中含FF时显示为'2017-09-01 10:32:19.212000'，在format参数中含FF9时显示为'2017-09-01 10:32:19.212000000'。
- 支持的时间范围存在差异，GaussDB支持的公元时间较Oracle范围更大。

表 2-3 字符类型

序号	Oracle数据库	GaussDB数据库	差异
1	VARCHAR2 (size [BYTE CHAR])	支持，有差异	在GaussDB中，size单位为字节，即仅支持BYTE，不支持在BYTE和CHAR之间选择，最大10MB；而在Oracle中，size的单位可以在BYTE和CHAR之间选择，MAX_STRING_SIZE=EXTENDED时，最大长度为32767字节，MAX_STRING_SIZE=STANDARD时，最大长度为4000字节，实际能容纳的字符数与使用的字符集有关。
2	NVARCHAR2 (size)	支持，有差异	在GaussDB中，size单位为字节，最大值为10MB，实际能容纳的字符数与使用的字符集有关； 在Oracle中，当MAX_STRING_SIZE=EXTENDED时，最大长度为32767字节，当MAX_STRING_SIZE=STANDARD时，最大长度为4000字节，实际能容纳的字符数与使用的字符集有关。
3	CHAR [(size [BYTE CHAR])]	支持，有差异	在GaussDB中，size单位为字节，即仅支持BYTE，不支持在BYTE和CHAR之间选择，最大10MB；而在Oracle中，size的单位可以在BYTE和CHAR之间选择，最大容量为2000个字节，实际能容纳的字符数与使用的字符集有关。

序号	Oracle数据库	GaussDB数据库	差异
4	NCHAR [(size)]	支持，有差异	在GaussDB中，size单位为字节，最大10MB；而在Oracle中，size单位为字符，最大容量为2000个字节，实际能容纳的字符数与使用的字符集有关。
5	CLOB	支持，有差异	不支持定位器。
6	NCLOB	不支持	-
7	LONG	不支持	-

表 2-4 二进制类型

序号	Oracle数据库	GaussDB数据库	差异
1	RAW (size)	支持，有差异	在GaussDB中，size是指字节长度建议值，不会用于校验输入raw类型的字节长度。
2	LONG RAW	不支持	-
3	BLOB	支持	-
4	BFILE	不支持。	-

表 2-5 ROWID 类型

序号	Oracle数据库	GaussDB数据库
1	ROWID	不支持
2	UROWID	不支持

表 2-6 用户自定义类型

序号	Oracle数据库	GaussDB数据库
1	对象类型	不支持
2	REF数据类型	不支持
3	可变数组	支持
4	嵌套表	支持

表 2-7 伪类型

序号	Oracle数据库	GaussDB数据库
1	anytype	不支持
2	anydata	不支持
3	anydataset	不支持

表 2-8 xml 类型

序号	Oracle数据库	GaussDB数据库	差异
1	XMLTYPE	支持，有差异。	GaussDB中不支持部分操作，如不能通过使用XMLELEMENT函数将字符串转变为XMLTYPE类型，而是转变成XML类型。具体请参见《开发指南》中“SQL参考 > 数据类型 > XMLTYPE类型”章节。
2	URITYPE	不支持。	-

表 2-9 空间类型

序号	Oracle数据库	GaussDB数据库
1	SDO_GEOMETRY	不支持
2	SDO_TOPO_GEOMETRY	不支持
3	SDO_GEORASTER	不支持

表 2-10 锁模式

序号	Oracle数据库	GaussDB数据库
1	none	-
2	null	AccessShare
3	RS	RowShare
4	RX	RowExclusive
5	S	ShareUpdateExclusive
6	SRX	Share
7	-	ShareRowExclusive
8	X	Exclusive

序号	Oracle数据库	GaussDB数据库
9	-	AccessExclusive
10	-	INVALID 说明 GaussDB的INVALID表示被赋予了非法锁。仅在运行过程中出现了GaussDB无法识别的锁时会被赋予INVALID锁。

2.2.2 数据类型比较规则

数据类型比较（排序）规则是指相同数据类型的值之间发生比较（排序）时遵循的比较（排序）规则。

表 2-11 比较规则

序号	Oracle数据库	GaussDB数据库	差异
1	Numeric 值	支持	-
2	日期时间 值	支持	-
3	二进制值	支持	-
4	字符值	支持，有差异	- 在GaussDB和Oracle中，支持的比较规则不完全相同，相同比较规则的名称也可能不同。 - GaussDB和Oracle的比较规则在可指定性上有差异，例如GaussDB不支持指定表级别的比较规则，而Oracle支持。 - GaussDB和Oracle在指定比较规则的语法上有差异，例如在GaussDB中，使用ENCODING、LC_CTYPE和LC_COLLATE三个参数决定创建数据库时使用的字符集、字符分类和比较规则，具体请参见《开发指南》中“SQL参考 > SQL语法 > C > CREATE DATABASE”章节。而在Oracle中，各级别的比较规则通常由一系列带有NLS前缀的参数确定。
5	对象值	不支持	-
6	Varrays 和嵌套表	支持，有差异	GaussDB和Oracle均支持Varrays的比较，与Oracle不同的是，GaussDB不仅支持比较两个Varrays中的元素个数，还支持同类型的Varrays之间的比较。

序号	Oracle数据库	GaussDB数据库	差异
7	数据类型 优先级	支持	-
8	数据转换 （显示/ 隐式类型 转换）	支持	-

2.2.3 字面量

表 2-12 字面量

序号	Oracle数据库	GaussDB数据库
1	文本字面量	支持
2	数值字面量	支持
3	日期时间字面量	支持
4	区间字面量	支持

2.2.4 格式模型

表 2-13 格式

序号	Oracle数据库	GaussDB数据库	差异
1	数字格式	支持，有差异	GaussDB仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下，支持\$、C、TM、TM9、TME、U格式。同时在该参数下，不支持TH、PL、SG格式。 GaussDB具体支持情况请参见《开发指南》中“SQL参考>函数和操作符>类型转换函数”章节的“number类型fmt参数表”。
2	日期时间格式	支持，有差异	GaussDB中用于时间截断和时间四舍五入的参数，仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。 GaussDB具体支持情况请参见《开发指南》中“SQL参考>函数和操作符>时间和日期处理函数和操作符”章节的“用于日期/时间格式化的模式”。

序号	Oracle数据库	GaussDB数据库	差异
3	格式模型修饰符	支持	-
4	字符串到日期转换规则	支持，有差异	GaussDB中to_timestamp_tz函数在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。GaussDB具体支持情况请参见《开发指南》中“SQL参考 > 函数和操作符 > 类型转换函数”章节的“to_date/to_timestamp/to_timestamp_tz”。
5	xml格式模式	不支持	-

2.2.5 空值

表 2-14 空值

序号	Oracle数据库	GaussDB数据库
1	IS NULL和IS NOT NULL	支持
2	NULLS in conditions	支持

2.2.6 注释

表 2-15 comment

序号	Oracle数据库	GaussDB数据库	差异
1	斜杠和星号（/*）	支持	-
2	两个连字符（--）	支持	-
3	COMMENT命令	支持	-
4	HINT	支持，有差异	GaussDB不支持'--+'hint形式。具体信息请参见《开发指南》中“SQL调优指南 > 使用Plan Hint进行调优”章节。

2.2.7 数据库对象

表 2-16 schema 对象

序号	Oracle数据库	GaussDB数据库	差异
1	分析视图	不支持	-
2	属性维度	不支持	-
3	集群	支持	-
4	约束	支持	-
5	数据库链接	支持	-
6	数据库触发器	支持	-
7	尺寸	支持	-
8	外部过程库	不支持	-
9	分层结构	不支持	-
10	索引组织表	不支持	-
11	索引	支持	-
12	索引类型	不支持	-
13	java类	不支持	-
14	java资源	不支持	-
15	java源码	不支持	-
16	join groups	不支持	-
17	物化视图	支持	-
18	物化视图日志	不支持	-
19	挖掘模型	不支持	-
20	对象表	不支持	-
21	对象类型	不支持	-
22	对象视图	不支持	-
23	operators	支持	-
24	包	支持	-
25	序列	支持	-
26	存储函数	支持	-

序号	Oracle数据库	GaussDB数据库	差异
27	存储过程	支持	-
28	同义词	支持，有差异	Oracle数据库的数据库对象在同一namespace内不能重名。GaussDB同一namespace内同义词可与表、视图、函数、package重名，如果重名，会优先访问该名称对象，未寻找到该名称对象时才会寻找该名称的同义词指向的对象；同义词指向的对象的Schema名是用户名时才会搜索PUBLIC同义词。详细的搜索顺序参考《开发指南》中的“SQL参考 > SQL语法 > C > CREATE SYNONYM”章节中的注意事项，同义词的搜索过程。
29	表	支持	-
30	视图	支持	-
31	zone map	不支持	-

表 2-17 nonschema 对象

序号	Oracle数据库	GaussDB数据库
1	上下文	不支持
2	目录	支持
3	版本	不支持
4	闪回存档	不支持
5	锁定配置文件	不支持
6	配置文件	不支持
7	还原点	支持
8	角色	支持
9	回滚段	• ustore支持回滚段 • astore不支持回滚段
10	表空间	支持
11	表空间集	不支持
12	统一审计策略	支持
13	用户	支持

2.2.8 数据库对象名称和限定符

表 2-18 命名规则

序号	Oracle数据库	GaussDB数据库	差异
1	数据库对象命名规则	支持，有差异	GaussDB默认小写。
2	模式对象命名规则	支持	-

2.2.9 SQL 语句中的引用架构对象和部件的语法

表 2-19 对象引用

序号	Oracle数据库	GaussDB数据库
1	引用对象的一般语法	支持
2	解析对象的引用	支持
3	引用外部模式对象	支持
4	引用外部数据库对象	支持
5	引用表和索引分区和子分区	支持

2.3 伪列

GaussDB数据库兼容序列伪列、rownum伪列，其余暂不支持。

分层查询伪列

表 2-20 分层查询伪列

序号	Oracle数据库	GaussDB数据库
1	connect_by_iscycle	支持
2	connect_by_isleaf	支持
3	level伪列	支持

序列伪列

表 2-21 序列

序号	Oracle数据库	GaussDB数据库	差异
1	currval	支持，有差异	GaussDB以函数形式实现。兼容Oracle调用方式。
2	nextval	支持，有差异	GaussDB以函数形式实现。兼容Oracle调用方式。

rownum 伪列

表 2-22 rownum

序号	Oracle数据库	GaussDB数据库	差异
1	rownum	支持，有差异。	Oracle在left、right、full join的条件中使用rownum进行过滤时，不同的条件下表现不尽相同，可能存在忽略或部分忽略rownum条件的现象，而GaussDB在此情况下则表现为对left、right、full join后的结果进行过滤。

xmldata 伪列

表 2-23 xmldata

序号	Oracle数据库	GaussDB数据库
1	xmldata	不支持

2.4 操作符

GaussDB数据库兼容除分层查询以外的运算符。

SQL 运算符

表 2-24 SQL 运算符

序号	Oracle数据库	GaussDB数据库
1	一元运算符和二元运算符	支持

序号	Oracle数据库	GaussDB数据库
2	运算符优先级	支持

算术运算符

表 2-25 算术运算符

序号	Oracle数据库	GaussDB数据库
1	正负（+-）一元运算符	支持
2	加减（+-）二元运算符	支持
3	乘除（*/）二元运算符	支持

COLLATE 运算符

表 2-26 COLLATE 运算符

序号	Oracle数据库	GaussDB数据库
1	COLLATE collation_name	支持

连接运算符

表 2-27 连接运算符

序号	Oracle数据库	GaussDB数据库
1		支持

分层查询运算符

表 2-28 分层查询运算符

序号	Oracle数据库	GaussDB数据库	差异
1	prior	支持，有差异。	GaussDB中仅支持对普通列调用，不支持对函数等调用。

序号	Oracle数据库	GaussDB数据库	差异
2	connect_by_root	支持，有差异。	GaussDB中，connect_by_root调用时，用括号修饰操作值时，行为与Oracle一致；不使用括号时，仅支持对普通列调用此运算符。

集合运算符

表 2-29 集合运算符

序号	Oracle数据库	GaussDB数据库
1	union	支持
2	union all	支持
3	intersect	支持
4	minus	支持

多集合运算符

表 2-30 多集合运算符

序号	Oracle数据库	GaussDB数据库
1	multiset except	支持
2	multiset intersect	支持
3	multiset union	支持

用户自定义运算符

表 2-31 用户自定义运算符

序号	Oracle数据库	GaussDB数据库	差异
1	CREATE OPERATOR	支持	- Oracle中提供的CONTEXT_CLAUSE支持自定义功能评估函数，和GaussDB约束选择性评估函数不同。GaussDB不支持自定义功能评估函数。 - GaussDB和Oracle可选参数差异较大。GaussDB具体请参考《开发指南》中“SQL参考 > SQL语法 > C > CREATE OPERATOR”章节的参数说明部分。

比较操作符

序号	Oracle数据库	GaussDB数据库
1	< =	支持
2	< >	支持
3	> =	支持
4	^ =	支持
5	!=	不支持，!=中间存在空格时，!会被识别为阶乘。

当比较操作符（<=、<>、>=、^=）中间存在空格时，也可以识别成没有空格进行正常操作。!=中间存在空格时，!会被识别为阶乘，可能会导致结果与预期不一致。

2.5 表达式

GaussDB数据库兼容大部分数据库表达式。

表 2-32 表达式

序号	Oracle数据库	GaussDB数据库	差异
1	简单表达式	支持	-
2	分析视图表达式	不支持	-
3	复合表达式	支持	-

序号	Oracle数据库	GaussDB数据库	差异
4	case表达式	支持	-
5	列表示式	支持	-
6	cursor表达式	不支持	-
7	日期时间表达式	支持，有差异	GaussDB的输出结果中不会带时区信息，Oracle会带有类似PM AMERICA/LOS_ANGELES的时区信息。
8	函数表达式	支持	-
9	区间表达式	部分支持	GaussDB支持形如SELECT INTERVAL '999999999 23:59:59.999' day(9) to second FROM DUAL;的语句，而不支持形如SELECT(SYSDATE- SYSDATE) DAY TO SECOND FROM DUAL;的语句。Oracle均支持。
10	json对象访问表达式	部分支持，有差异	• GaussDB支持通过 “->'key' “的方式从JSON对象中提取value，而Oracle支持通过 “.key “方式来提取value。 • 对于JSONARRAY对象而言，Oracle支持通过 “.key “方式一次性提取所有key对应的value，但GaussDB目前不支持。
11	模型表达式	不支持	-
12	对象表达式	不支持	-
13	占位符表达式	部分支持	对于形如 “:var “的一般占位符表达式，GaussDB支持，但不支持通过INDICATOR关键字将两个一般占位符表达式结合起来。
14	标量子查询表达式	支持	-
15	类型构造器表达式	部分支持	GaussDB不支持在类型构造器前指定New关键字，而Oracle支持。
16	表达式list	支持	-

2.6 条件

本章节描述常见的条件兼容项，包含比较条件、浮点条件、逻辑条件、模型条件、多集合条件、模式匹配条件、NULL值条件、XML条件、SQL/JSON条件、复合条件、BETWEEN条件、EXISTS条件、IN条件、IS OF TYPE条件，如表2-33所示。

表 2-33 条件

序号	Oracle数据库	GaussDB数据库	差异
1	比较条件	支持，有差异	语句中存在ANY、SOME、ALL操作符时存在差异，Oracle支持对list对象进行操作，而GaussDB中需要将list对象转换成数组表达式的形式后再进行操作。
2	浮点条件	不支持	-
3	逻辑条件	支持	-
4	模型条件	不支持	-
5	多集合条件	不支持	-
6	模式匹配条件	支持	-
7	NULL值条件	支持	-
8	XML条件	不支持	-
9	SQL/JSON条件	部分支持，有差异	● GaussDB不支持IS JSON条件和JSON_TEXTCONTAINS条件。 ● GaussDB中JSONB_EQ条件等同于Oracle中JSON_EQUAL条件，但GaussDB不支持ERROR子句。 ● GaussDB中JSONB_EXISTS条件等同于Oracle中JSON_EXISTS条件，但GaussDB不支持ERROR子句、EMPTY子句和PASSING子句。
10	复合条件	支持	-
11	BETWEEN条件	支持	-
12	EXISTS条件	支持	-
13	IN条件	支持	-
14	IS OF TYPE条件	不支持	-

2.7 驱动

2.7.1 JDBC

2.7.1.1 Array

本章节描述在JDBC驱动使用java.sql.Array类型时，Oracle与GaussDB的差异。

表 2-34 构造方法参考

构造方式	Oracle数据库	GaussDB数据库	接口差异
1. ArrayDescriptor的静态构造方法构造ArrayDescriptor对象； 2. 通过ArrayDescriptor构造Array对象。	<pre>String typeName = "XXX"; Connection conn = getConnection(); Object[] elements = null; ArrayDescriptor desc = ArrayDescriptor.createDescriptor(typeName, conn); Array array = new ARRAY(desc, conn, elements);</pre>	<pre>String typeName = "xxx"; Connection conn = getConnection(); Object[] elements = null; ArrayDescriptor desc = ArrayDescriptor.getDescriptor(typeName, conn); Array array = new GaussArray(desc, elements);</pre>	- 构造ArrayDescriptor的静态方法名不一致，Oracle的方法名为createDescriptor，GaussDB的方法名为getDescriptor。 - Array的构造方法名定义不一致，Oracle为ARRAY(ArrayDescriptor, Connection, Object)，GaussDB为GaussArray(ArrayDescriptor, Object)。 - 变量说明： typeName为类型名称，且大小写敏感，通常Oracle的类型名大写，GaussDB的类型名小写。 conn为对应数据库的连接对象。 elements为对应类型的元素数据。

说明

1. 对于上述表格中未列举出的构造方式，GaussDB暂不支持。
2. 若元素类型为字符类型，构造传入的字符串长度大于元素类型定义的长度，Oracle会在绑定入参时报错；
若数组类型为varray类型时，若元素个数大于varray类型的长度上限，Oracle会在绑定入参时报错；
GaussDB在构造或者绑定入参时不会对类型修饰符进行校验，在数据库接收Array对象并执行SQL时，由数据库决定是否报错。

表 2-35 接口参考

方法名	返回值类型	throws	GaussDB数据库
getBaseTypeName()	String	SQLException	支持
getBaseType()	int	SQLException	支持
getArray()	Object	SQLException	支持
getArray(java.util.Map<String,Class<?>> map)	Object	SQLException	不支持
getArray(long index, int count)	Object	SQLException	支持
getArray(long index, int count, java.util.Map<String,Class<?>> map)	Object	SQLException	不支持
getResultSet()	ResultSet	SQLException	不支持
getResultSet(java.util.Map<String,Class<?>> map)	ResultSet	SQLException	不支持
getResultSet(long index, int count)	ResultSet	SQLException	不支持
getResultSet (long index, int count, java.util.Map<String,Class<?>> map)	ResultSet	SQLException	不支持
free()	void	SQLException	不支持

表 2-36 getArray()接口详细差异

元素数据库类型	getArray接口实际返回值类型（Oracle）	getArray接口实际返回值类型（GaussDB）
CHAR	java.lang.String[]	java.lang.String[]
VARCHAR/ VARCHAR2	java.lang.String[]	java.lang.String[]
NCHAR	java.lang.String[]	java.lang.String[]
NVARCHAR2	java.lang.String[]	java.lang.String[]
NUMBER	java.math.BigDecimal[]	java.math.BigDecimal[]

元素数据库类型	getArray接口实际返回值类型（Oracle）	getArray接口实际返回值类型（GaussDB）
NUMERIC	java.math.BigDecimal[]	java.math.BigDecimal[]
DECIMAL	java.math.BigDecimal[]	java.math.BigDecimal[]
INTEGER	java.math.BigDecimal[]	java.lang.Integer[]
SMALLINT	java.math.BigDecimal[]	java.lang.Short[]
DOUBLE PRECISION	java.math.BigDecimal[]	java.lang.Double[]
FLOAT	java.math.BigDecimal[]	java.lang.Double[]
REAL	java.math.BigDecimal[]	java.lang.Float[]
BINARY_DOUBLE	java.lang.Double[]	java.lang.Double[]
BINARY_INTEGER	java.math.BigDecimal[]	java.lang.Integer[]
BOOLEAN	java.math.BigDecimal[]	java.lang.Boolean[]
TIMESTAMP	java.sql.Timestamp[]	java.sql.Timestamp[]
TIMESTAMP WITH TIME ZONE	java.time.OffsetDateTime[]	java.sql.Timestamp[]
BLOB	oracle.sql.BLOB[]	java.sql.Blob[]
CLOB	oracle.sql.CLOB[]	java.sql.Clob[]
集合类型/数组类型	java.lang.Object[]	java.sql.Array[]
RECORD类型	java.lang.Object[]	java.sql.Struct[]

说明

- 对于上述表格中未列举出的类型，GaussDB暂不支持。
- getArray(long index, int count)接口返回值差异也参考上述表格。
- getArray接口(long index, int count)，参数index的差异如下：
 - Oracle支持范围为[1, Long.MAX_VALUE]；Gauss支持范围为[1, Integer.MAX_VALUE]。
 - Oracle在index > Integer.MAX_VALUE时index会被截断；Gauss在index > Integer.MAX_VALUE时报错。

表 2-37 getBaseType()接口详细差异

元素数据库类型	getBaseType接口返回值（Oracle）	getBaseType接口返回值（GaussDB）
CHAR	java.sql.Types.CHAR	java.sql.Types.CHAR

元素数据库类型	getBaseType接口返回值 (Oracle)	getBaseType接口返回值 (GaussDB)
VARCHAR/ VARCHAR2	java.sql.Types.VARCHAR	java.sql.Types.VARCHAR
NCHAR	java.sql.Types.NCHAR	java.sql.Types.CHAR
NVARCHAR2	java.sql.Types.NVARCHAR R	java.sql.Types.VARCHAR
NUMBER	java.sql.Types.NUMERIC	java.sql.Types.NUMERIC
NUMERIC	java.sql.Types.DECIMAL	java.sql.Types.NUMERIC
DECIMAL	java.sql.Types.DECIMAL	java.sql.Types.NUMERIC
INTEGER	java.sql.Types.NUMERIC	java.sql.Types.INTEGER
SMALLINT	java.sql.Types.NUMERIC	java.sql.Types.SMALLINT
DOUBLE PRECISION	java.sql.Types.FLOAT	java.sql.Types.DOUBLE
FLOAT	java.sql.Types.FLOAT	java.sql.Types.DOUBLE
REAL	java.sql.Types.FLOAT	java.sql.Types.REAL
BINARY_DOUBLE	oracle.jdbc.OracleTypes.B INARY_DOUBLE	java.sql.Types.DOUBLE
BINARY_INTEGER	java.sql.Types.NUMERIC	java.sql.Types.INTEGER
BOOLEAN	java.sql.Types.NUMERIC	java.sql.Types.BIT
TIMESTAMP	java.sql.Types.TIMESTAM P	java.sql.Types.TIMESTAMP
TIMESTAMP WITH TIME ZONE	oracle.jdbc.OracleTypes.T IMESTAMPTZ	java.sql.Types.TIMESTAMP
BLOB	java.sql.Types.BLOB	java.sql.Types.BLOB
CLOB	java.sql.Types.CLOB	java.sql.Types.CLOB
集合类型/数组类型	java.sql.Types.ARRAY	java.sql.Types.ARRAY
RECORD类型	java.sql.Types.STRUCT	java.sql.Types.STRUCT

说明

对于上述表格中未列举出的类型，GaussDB暂不支持。

表 2-38 getBaseTypeName()接口详细差异

元素数据库类型	getBaseTypeName接口 返回值（Oracle）	getBaseTypeName接口返回值 （GaussDB）
CHAR	"CHAR"	"bpchar"
VARCHAR/ VARCHAR2	"VARCHAR"	"varchar"
NCHAR	"NCHAR"	"bpchar"
NVARCHAR2	"NVARCHAR"	"nvarchar2"
NUMBER	"NUMBER"	"numeric"
NUMERIC	"DECIMAL"	"numeric"
DECIMAL	"DECIMAL"	"numeric"
INTEGER	"NUMBER"	"int4"
SMALLINT	"NUMBER"	"int2"
DOUBLE PRECISION	"FLOAT"	"float8"
FLOAT	"FLOAT"	"float8"
REAL	"FLOAT"	"float4"
BINARY_DOUBLE	"BINARY_DOUBLE"	"float8"
BINARY_INTEGER	"NUMBER"	"int4"
BOOLEAN	"NUMBER"	"bool"
TIMESTAMP	"TIMESTAMP"	"timestamp"
TIMESTAMP WITH TIME ZONE	"TIMESTAMP WITH TIME ZONE"	"timestamptz"
BLOB	"BLOB"	"blob"
CLOB	"CLOB"	"clob"
集合类型/数组类型	见说明部分	见说明部分
RECORD类型	见说明部分	见说明部分

说明

1. 对于上述表格中未列举出的类型，GaussDB暂不支持。
2. 元素为集合类型、数组类型或record类型，且元素类型是package里定义的类型时，getBaseTypeName返回规则如下：
 - OJDBC11返回schemaName.packageName.typeName。
 - OJDBC8一般返回schemaName.packageName.typeName，在类型名满足以下条件返回"schemaName"."packageName.typeName":
schemaName/packageName/typeName中有任意名称不符合首字符为字母，其他字符为字母、数字或下划线。
 - GaussDB一般返回schemaName.packageName.typeName，在类型名满足以下条件返回"schemaName"."packageName"."typeName":
schemaName/packageName/typeName中有任意名称不符合首字符为字母或下划线，其他字符为字母、数字或下划线。
3. 元素为集合类型、数组类型或record类型，且元素类型是schema下定义的类型时（非package里定义的类型），getBaseTypeName返回规则如下：
 - OJDBC11返回schemaName.typeName。
 - OJDBC8一般返回schemaName.typeName，在类型名满足以下条件返回"schemaName"."typeName":
schemaName/typeName中有任意名称不符合首字符为字母，其他字符为字母、数字或下划线。
 - GaussDB一般返回schemaName.typeName，在类型名满足以下条件返回"schemaName"."typeName":
schemaName/typeName中有任意名称不符合首字符为字母或下划线，其他字符为字母、数字或下划线。
4. 若元素类型在创建时不做特殊处理，Oracle的getBaseTypeName接口通常返回的是大写的类型名称，GaussDB的getBaseTypeName接口通常返回的是小写的类型名称。

表 2-39 Array 构造接口详细差异

元素数据库类型	元素入参支持的Java类型列表（ Oracle OJDBC8 ）	元素入参支持的Java类型列表（ GaussDB ）	差异说明
CHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。

元素数据库类型	元素入参支持的Java类型列表（Oracle JDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
VARCHAR/ VARCHAR2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。
NCHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。
NVARCHA R2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	元素入参支持的类型差异见表格。
NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。

元素数据库类型	元素入参支持的Java类型列表（ Oracle OJDBC8 ）	元素入参支持的Java类型列表（ GaussDB ）	差异说明
NUMERIC	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
DECIMAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 当入参超过Integer范围时GaussDB会报错。
SMALLINT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 当入参超过Short范围时GaussDB会报错。

元素数据库类型	元素入参支持的Java类型列表（ Oracle OJDBC8 ）	元素入参支持的Java类型列表（ GaussDB ）	差异说明
DOUBLE PRECISION	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
FLOAT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
REAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Float可能会出现精度丢失的情况。
BINARY_DOUBLE	Double、oracle.sql.BINARY_DOUBLE	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。

元素数据库类型	元素入参支持的Java类型列表（ Oracle OJDBC8 ）	元素入参支持的Java类型列表（ GaussDB ）	差异说明
BINARY_INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAddr、LongAccumulator、LongAddr、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. OJDBC11元素入参支持的类型：Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER 3. 目标类型不一致，参考getArray接口详细差异。 4. 当传入数值超过Integer范围时：GaussDB会报错Oracle OJDBC8会进行截断处理。
BOOLEAN	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAddr、LongAccumulator、LongAddr、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. 元素入参支持的类型差异见表格。 2. OJDBC8元素入参支持的类型：Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER 3. 目标类型不一致，参考getArray接口详细差异。 4. GaussDB目标类型为Boolean类型，仅支持1、0、"true"、"false"等输入。

元素数据库类型	元素入参支持的Java类型列表（Oracle JDBC8）	元素入参支持的Java类型列表（GaussDB）	差异说明
TIMESTAMP	byte[]、 java.sql.Date、 Calendar、 java.util.Date、 LocalDate、 LocalDateTime、 LocalTime、 OffsetDateTime、 OffsetTime、 String、 java.sql.Time、 java.sql.Timestamp、 oracle.sql.DATE、 oracle.sql.TIMESTAMP、 oracle.sql.TIMESTAMPTZ、 oracle.sql.TIMESTAMPTZ、 ZonedDateTime	java.util.Date、 java.sql.Date、 java.sql.Time、 java.sql.Timestamp、 LocalDateTime、 String	元素入参支持的类型差异见表格。
TIMESTAMP WITH TIME ZONE	java.sql.Date、 Calendar、 java.util.Date、 LocalDate、 LocalDateTime、 LocalTime、 OffsetDateTime、 OffsetTime、 String、 java.sql.Time、 java.sql.Timestamp、 oracle.sql.DATE、 oracle.sql.TIMESTAMP、 oracle.sql.TIMESTAMPTZ、 oracle.sql.TIMESTAMPTZ、 ZonedDateTime	java.util.Date、 java.sql.Date、 java.sql.Time、 java.sql.Timestamp、 LocalDateTime、 String	1. 元素入参支持的类型差异见表格。 2. 目标类型不一致，参考getArray接口详细差异。
BLOB	oracle.sql.BLOB、 oracle.jdbc.driver.OracleBlob	PGBlob	元素入参支持的类型差异见表格。

元素数据库类型	元素入参支持的Java类型列表（ Oracle OJDBC8 ）	元素入参支持的Java类型列表（ GaussDB ）	差异说明
CLOB	oracle.sql.CLOB、oracle.jdbc.driver.OracleClob、InputStream、Reader	PGClob	元素入参支持的类型差异见表格。
集合类型/ 数组类型	ARRAY、Object	GaussArray、Object	1. 元素入参支持的类型差异见表格。 2. Oracle：当元素入参为ARRAY时，ARRAY的类型与元素类型不一致时不会报错。 GaussDB：当元素入参为GaussArray时，GaussArray的类型与元素类型不一致时会报错。 3. 当元素入参为Object时，参考Array构造接口差异。
RECORD类型	STRUCT、Object[]	GaussStruct、Object[]	1. 元素入参支持的类型差异见表格。 2. Oracle：当元素入参为STRUCT时，STRUCT的类型与元素类型不一致时不会报错。 GaussDB：当元素入参为GaussStruct时，GaussStruct的类型与元素类型不一致时会报错。 3. 当元素入参为Object[]时，参考Struct构造接口差异。

 说明

- 1. 构造Array时，若传入元素的Java类型不符合目标类型，会进行相应的隐式转换操作，对于不同的数据库元素类型，支持传入的元素的Java类型差异见上表。
- 2. 对于上述表格中未列举出的类型，GaussDB暂不支持。
- 3. 构造方法需要提供元素数组，上述表格描述的是数组中每一个元素对应的差异。

2.7.1.2 Struct

本章节描述在JDBC驱动使用java.sql.Struct类型时，Oracle与GaussDB的差异。

表 2-40 构造方法参考

构造方式	Oracle数据库	GaussDB数据库	接口差异
1. StructDescriptor的静态构造方法构造StructDescriptor对象 2. 通过StructDescriptor构造Struct对象	<pre>String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; StructDescriptor desc = StructDescriptor.createDescriptor(typeName, conn); Struct struct = new STRUCT(desc, conn, attributes);</pre>	<pre>String typeName = "xxx"; Connection conn = getConnection(); Object[] elements = null; StructDescriptor desc = StructDescriptor.getDescriptor(typeName, conn); Struct struct = new GaussStruct(desc, attributes);</pre>	- 构造StructDescriptor的静态方法名不一致，Oracle的方法名为createDescriptor，GaussDB的方法名为getDescriptor。 - Struct的构造方法名定义不一致，Oracle为STRUCT(StructDescriptor, Connection, Object[]), GaussDB为GaussStruct(StructDescriptor, Object[])。 - 变量说明： typeName为类型名称，且大小写敏感，通常Oracle的类型名大写，GaussDB的类型名小写。 conn为对应数据库的连接对象。 attributes为元素数据数组。
通过Connection的createStruct标准接口构造Struct对象	<pre>String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; Struct struct = conn.createStruct(typeName, attributes);</pre>	<pre>String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; Struct struct = conn.createStruct(typeName, attributes);</pre>	变量说明： typeName为类型名称，且大小写敏感，通常Oracle的类型名大写，GaussDB的类型名小写。 conn为对应数据库的连接对象。 attributes为元素数据数组。

 说明

1. 对于上述表格中未列举出的构造方式，GaussDB暂不支持。
2. 若attribute的类型为字符类型，构造传入的字符串长度大于元素类型定义的长度，Oracle会在绑定入参时报错。

GaussDB在构造或者绑定入参时不会对类型修饰符进行校验，在数据库接收Struct对象并执行SQL时，由数据库决定是否报错。
3. 数组元素数量大于对应类型的实际列数量，则会在创建时报错。

数组元素数量小于实际列数量时：Oracle能创建成功，在传入参执行时会报错；GaussDB会在创建时报错。

表 2-41 接口参考

方法名	返回值类型	throws	GaussDB数据库
getSQLTypeName()	String	SQLException	支持
getAttributes()	Object[]	SQLException	支持
getAttributes(java.util.Map<String,Class<?>> map)	Object[]	SQLException	不支持

说明

getSQLTypeName接口差异参考如下说明：

1. 对于package类型，通过packageName.typeName形式构造的Struct，getSQLTypeName接口差异如下：
- OJDBC11返回packageName.typeName。
 - OJDBC8返回packageName.typeName，在packageName和typeName满足以下条件时返回"packageName"."typeName":
packageName/typeName有任意名称不符合首字符为字母，其他字符为字母或者数字或者下划线。
 - GaussDB返回schemaName.packageName.typeName，在schemaName、packageName和typeName满足以下条件时返回
"schemaName"."packageName"."typeName":
schemaName/packageName/typeName有任意名称不符合首字符为字母或者下划线，其他字符为字母、数字或者下划线。
2. 对于package类型，其他场景下getSQLTypeName接口差异如下：
- OJDBC11返回schemaName.packageName.typeName。
 - OJDBC8返回schemaName.packageName.typeName，在schemaName、packageName和typeName满足以下条件时返回
"schemaName"."packageName.typeName":
schemaName/packageName/typeName有任意名称不符合首字符为字母，其他字符为字母或者数字或者下划线。
 - GaussDB返回schemaName.packageName.typeName，在schemaName、packageName和typeName满足以下条件时返回
"schemaName"."packageName"."typeName":
schemaName/packageName/typeName有任意名称不符合首字符为字母或者下划线，其他字符为字母、数字或者下划线。
3. 对于非package类型，getSQLTypeName接口差异如下：
- OJDBC11返回schemaName.typeName。
 - OJDBC8返回schemaName.typeName，在schemaName和typeName满足以下条件时返回"schemaName"."typeName":
schemaName/typeName有任意名称不符合首字符为字母，其他字符为字母或者数字或者下划线。
 - GaussDB返回schemaName.typeName，在schemaName和typeName满足以下条件时返回"schemaName"."typeName":
schemaName/typeName有任意名称不符合首字符为字母或者下划线，其他字符为字母、数字或者下划线。

表 2-42 getAttributes()接口详细差异

Attribute数据库类型	返回值中对应元素的Java类型 (Oracle OJDBC8)	返回值中对应元素的Java类型 (Oracle OJDBC11)	返回值中对应元素的Java类型 (GaussDB)
CHAR	String	String	String
VARCHAR/ VARCHAR2	String	String	String
NCHAR	String	String	String

Attribute数据库类型	返回值中对应元素的Java类型（Oracle JDBC8）	返回值中对应元素的Java类型（Oracle JDBC11）	返回值中对应元素的Java类型（GaussDB）
NVARCHAR2	String	String	String
NUMBER	BigDecimal	BigDecimal	BigDecimal
NUMERIC	BigDecimal	BigDecimal	BigDecimal
DECIMAL	BigDecimal	BigDecimal	BigDecimal
INTEGER	BigDecimal	BigDecimal	Integer
SMALLINT	BigDecimal	BigDecimal	Short
DOUBLE PRECISION	BigDecimal	BigDecimal	Double
FLOAT	BigDecimal	BigDecimal	Double
REAL	BigDecimal	BigDecimal	Float
BINARY_DOUBLE	Double	Double	Double
BINARY_INTEGER	BigDecimal	Integer	Integer
BOOLEAN	BigDecimal	Integer	Boolean
TIMESTAMP	Timestamp	Timestamp	Timestamp
TIMESTAMP WITH TIME ZONE	TIMESTAMPTZ	TIMESTAMPTZ	Timestamp
BLOB	BLOB	BLOB	PGBlob
CLOB	CLOB	CLOB	PGClob
集合类型/数组类型	ARRAY	ARRAY	GaussArray
RECORD类型	STRUCT	STRUCT	GaussStruct

 说明

对于上述表格中未列举出的类型，GaussDB暂不支持。

表 2-43 Struct 构造接口详细差异

Attribute 数据库类型	Attribute入参支持的Java类型列表 (Oracle)	Attribute入参支持的Java类型列表 (GaussDB)	差异说明
CHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	1. attribute入参支持的类型差异见表格。 2. 当入参为String时，Oracle会在字符串末尾补齐空格至字符串长度与类型定义的长度一致；GaussDB不会补空格。
VARCHAR/ VARCHAR2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	attribute入参支持的类型差异见表格。
NCHAR	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	attribute入参支持的类型差异见表格。
NVARCHA R2	任意Java类型	Byte、Short、Integer、Long、BigInteger、BigDecimal、Float、Double、Character、Boolean、String、java.sql.Date、java.sql.Time、java.sql.Timestamp、PGClob	attribute入参支持的类型差异见表格。

Attribute 数据库类型	Attribute入参支持的 Java类型列表 (Oracle)	Attribute入参支持的 Java类型列表 (GaussDB)	差异说明
NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
NUMERIC	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
DECIMAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当入参为Float、Double、BigDecimal、String且小数点部分为0时，Oracle会舍去小数部分；GaussDB会保留小数部分。
INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 当入参超过Integer范围时GaussDB会报错。

Attribute 数据库类型	Attribute入参支持的 Java类型列表 (Oracle)	Attribute入参支持的 Java类型列表 (GaussDB)	差异说明
SMALLINT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 当入参超过Short范围时GaussDB会报错。
DOUBLE PRECISION	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
FLOAT	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
REAL	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Float可能会出现精度丢失的情况。

Attribute 数据库类型	Attribute入参支持的 Java类型列表 (Oracle)	Attribute入参支持的 Java类型列表 (GaussDB)	差异说明
BINARY_DOUBLE	byte[]、Double、 oracle.sql.BINARY_DOUBLE	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. 更高精度的类型转Double可能会出现精度丢失的情况。
BINARY_INTEGER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAdder、LongAccumulator、LongAdder、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 当传入数值超过Integer范围时，GaussDB会报错，Oracle会进行截断处理。
BOOLEAN	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、AtomicInteger、AtomicLong、DoubleAccumulator、DoubleAdder、LongAccumulator、LongAdder、Striped64、String、oracle.sql.NUMBER	Byte、Short、Integer、Long、Float、Double、Boolean、BigDecimal、BigInteger、String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考getAttributes()接口详细差异。 3. GaussDB目标类型为Boolean类型，仅支持1、0、"true"、"false"等输入。

Attribute 数据库类型	Attribute入参支持的 Java类型列表 (Oracle)	Attribute入参支持的 Java类型列表 (GaussDB)	差异说明
TIMESTAMP	byte[]、 java.sql.Date、 String、 java.sql.Time、 java.sql.Timestamp 、 oracle.sql.TIMESTAMP、 oracle.sql.DATE	java.util.Date、 java.sql.Date、 java.sql.Time、 java.sql.Timestamp 、 LocalDateTime、 String	attribute入参支持的类型差异见表格。
TIMESTAMP WITH TIME ZONE	java.sql.Date、 Calendar、 java.util.Date、 LocalDate、 LocalDateTime、 LocalTime、 OffsetDateTime、 OffsetTime、 String、 java.sql.Time、 java.sql.Timestamp 、 oracle.sql.TIMESTAMP、 oracle.sql.TIMESTAMPTZ、 oracle.sql.TIMESTAMPTZ、 ZonedDateTime	java.util.Date、 java.sql.Date、 java.sql.Time、 java.sql.Timestamp 、 LocalDateTime、 String	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考 getAttributes()接口详细差异。
BLOB	oracle.sql.BLOB、 oracle.jdbc.driver.OracleBlob	PGBlob	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考 getAttributes()接口详细差异。
CLOB	oracle.sql.CLOB、 oracle.jdbc.driver.OracleClob	PGClob	1. attribute入参支持的类型差异见表格。 2. 目标类型不一致，参考 getAttributes()接口详细差异。

Attribute 数据库类型	Attribute入参支持的 Java类型列表 (Oracle)	Attribute入参支持的 Java类型列表 (GaussDB)	差异说明
集合类型/ 数组类型	ARRAY、Object	GaussArray、Object	1. attribute入参支持的类型差异见表格。 2. Oracle：当 attribute入参为 ARRAY时，ARRAY 的类型与attribute实际需要的类型不一致时不会报错。 GaussDB：当 attribute入参为 GaussArray时，GaussArray的类型与attribute实际需要的类型不一致时会报错。 3. 当attribute入参为 Object时，参考 Array构造接口差异。
RECORD类型	STRUCT、Object[]	GaussStruct、Object[]	1. attribute入参支持的类型差异见表格。 2. Oracle：当 attribute入参为 STRUCT时，STRUCT的类型与 attribute实际需要的类型不一致时不会报错。 GaussDB：当 attribute入参为 GaussStruct时，GaussStruct的类型与attribute实际需要的类型不一致时会报错。 3. 当attribute入参为 Object[]时，参考 Struct构造接口差异。

说明

- 1. 构造Struct时，若传入元素的Java类型不符合目标类型，会进行相应的隐式转换操作，对于不同的数据库元素类型，支持传入的元素的Java类型差异见上表。
- 2. 对于上述表格中未列举出的类型，GaussDB暂不支持。
- 3. 构造方法需要提供属性数组，上述表格描述的是数组中每一个attribute对应的差异。

2.8 常见的 SQL DDL 子句

本章节描述常见的SQL DDL子句兼容项，包含分配扩展子句、约束、取消分配未使用子句、文件规范、日志记录子句、并行子句、物理属性子句、大小子句、存储子句、聚集函数嵌套，如表2-44所示。

表 2-44 常用 SQL DDL 子句

序号	Oracle数据库	GaussDB数据库	差异
1	分配扩展子句 语法： ALLOCATE EXTENT [({ SIZE size_clause DATAFILE 'filename' INSTANCE integer } ...)] 例如：创建employees 表后，改变表的分配扩展size为10M。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. SQL> ALTER TABLE employees ALLOCATE EXTENT (SIZE 10M); Table altered.	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	约束 语法： <pre>{ inline_constraint out_of_line_constraint inline_ref_constraint out_of_line_ref_constraint }</pre> 例如：创建表staff，约束子句中指定ID列、NAME列不为空。 <pre>SQL> CREATE TABLE staff(ID INT NOT NULL, NAME char(8) NOT NULL, AGE INT, ADDRESS CHAR(50), SALARY REAL);</pre> Table created.	支持	-
3	取消分配未使用子句 语法： <pre>DEALLOCATE UNUSED [KEEP size_clause]</pre> 例如：创建employees表，进行了一些插入、删除操作后，希望使用取消分配未使用子句释放employees表未使用的空间。 <pre>SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16));</pre> Table created. - 进行一些插入、删除操作 <pre>SQL> ALTER TABLE employees DEALLOCATE UNUSED;</pre> Table altered.	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
4	文件规范 语法： { ['filename' 'ASM_filename'] [SIZE size_clause] [REUSE] [autoextend_clause] } { ['filename ASM_filename' ('filename ASM_filename' [, 'filename ASM_filename']...)] [SIZE size_clause] [BLOCKSIZE size_clause [REUSE] } 例如：创建一个临时表 空间tbs_temp_01， SQL语句的文件规范子 句中指定在表空间中创 建一个临时数据库文件 templ01.dbf，可以自动 扩展，并将表空间分配 给表空间组 tbs_grp_01。 SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
5	日志记录子句 语法： { LOGGING NOLOGGING FILESYSTEM_LIKE_LOGGING }	部分支持，有差异	- GaussDB不支持LOGGING约束子句和FILESYSTEM_LIKE_LOGGING约束子句。 例如： GaussDB创建表，带LOGGING约束子句，语法报错。 gaussdb=# CREATE LOGGING TABLE my_tab(id int, name char(16)); ERROR: syntax error at or near "LOGGING" LINE 1: CREATE LOGGING TABLE my_tab(id int, name char(16)); ^ - GaussDB创建表，带FILESYSTEM_LIKE_LOGGING约束子句，语法报错。 gaussdb=# CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name char(16)); ERROR: syntax error at or near "FILESYSTEM_LIKE_LOGGING" LINE 1: CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name cha... ^ - GaussDB仅支持表级的UNLOGGED约束，不支持列级的UNLOGGED约束。 例如：GaussDB创建表，带列级的UNLOGGED约束子句，语法报错。 gaussdb=# CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16)); ERROR: syntax error at or near "UNLOGGED" LINE 1: CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16))... ^ - GaussDB仅支持在CREATE TABLE、CREATE TABLE AS、SELECT INTO语句中使用日志记录子句。 例如：GaussDB创建TABLESPACE，带UNLOGGED约束子句，语法报错。 gaussdb=# CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespace1/tablespace_1'; ERROR: syntax error at or near "TABLESPACE" LINE 1: CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespac... ^

序号	Oracle数据库	GaussDB数据库	差异
6	并行子句 语法： { NOPARALLEL PARALLEL [integer] } 例如：创建表t1，并在并行子句中指定 PARALLEL 4，意为查询和更新表t1时最多使用4个并行进程操作。 <pre>SQL> CREATE TABLE t1 (id NUMBER, name VARCHAR2(50)) PARALLEL 4;</pre> Table created.	不支持	-
7	物理属性子句 语法： [{ PCTFREE integer PCTUSED integer INITRANS integer storage_clause }...]	部分支持，有差异	- GaussDB不支持PCTUSED。 例如：执行在表tbl1中创建一个tbl1_ind的索引的SQL语句，并在该语句的物理属性子句中指定索引的空间利用率PCTUSED为20%，GaussDB执行该SQL语句语法报错。 <pre>gaussdb=# CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20; ERROR: syntax error at or near "PCTUSED" LINE 1: CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20; ^</pre> - GaussDB 仅支持在CREATE TABLE、CREATE INDEX语句中使用物理属性子句。 例如：尝试从表tbl1中获取数据，创建物化视图tbl1_mv，并在物理属性子句中指定该视图的初始化事务数为30，GaussDB执行该SQL语句语法报错。 <pre>gaussdb=# CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * from tbl1; ERROR: syntax error at or near "INITRANS" LINE 1: CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * fro... ^</pre>

序号	Oracle数据库	GaussDB数据库	差异
8	大小子句 语法： integer [K M G T P E] 例如：创建一个临时表空间tbs_temp_01，并在表空间中创建一个临时数据库文件templ01.dbf，SQL语句的大小子句中指定初始大小是5M，可以自动扩展，并将表空间分配给表空间组tbs_grp_01。 <pre>SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' SIZE 5M AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01;</pre> Tablespace created.	不支持	-
9	存储子句 语法： <pre>STORAGE ({ INITIAL size_clause NEXT size_clause MINEXTENTS integer MAXEXTENTS { integer UNLIMITED } maxsize_clause PCTINCREASE integer FREELISTS integer FREELIST GROUPS integer OPTIMAL [size_clause NULL] BUFFER_POOL { KEEP RECYCLE DEFAULT } FLASH_CACHE { KEEP NONE DEFAULT } (CELL_FLASH_CACHE (KEEP NONE DEFAULT)) ENCRYPT } ...)</pre>	部分支持，有差异	- Oracle中由STORAGE子句指定存储参数，而GaussDB中由WITH子句指定存储参数。 - 例如： Oracle中创建表my_tab1，在存储子句中指定表初始大小为10M，需要更多空间时每次增加5M的SQL语句如下： <pre>SQL> CREATE TABLE my_tab1 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) STORAGE (INITIAL 10M NEXT 5M);</pre> Table created. GaussDB中创建表my_tab2，在存储子句中指定存储引擎类型为USTORE的SQL语句如下： <pre>gaussdb=# CREATE TABLE my_tab2 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) with (storage_type=ustore);</pre> NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "my_tab2_pkey" for table "my_tab2" <pre>CREATE TABLE</pre> - GaussDB中可选的存储参数和Oracle存在很大差异。GaussDB具体可参见《开发指南》中“SQL参考 > SQL语法 > C > CREATE TABLE”的参数说明部分，“WITH ({storage_parameter = value} [, ...])”中描述了CREATE TABLE语句支持的存储参数。

序号	Oracle数据库	GaussDB数据库	差异
10	聚集函数嵌套 例如：创建由sales表的sales_amount列嵌套聚集函数MIN()、SUM()生成的表revenue。 <pre>SQL> CREATE TABLE sales(ID INT, SALES_AMOUNT INT);</pre> Table created. <pre>SQL> INSERT INTO sales VALUES(1, 100);</pre> 1 row created. <pre>SQL> INSERT INTO sales VALUES (3, 200);</pre> 1 row created. <pre>SQL> CREATE TABLE revenue as SELECT SUM(MIN(sales_amount)) as total from sales group by sales_amount;</pre> Table created.	支持	-
11	删除系统schema。 语法： <pre>DROP USER schema_name CASCADE;</pre> 例如：在SYS用户下删除SYS schema。 <pre>SQL> DROP USER SYS;</pre> <pre>DROP USER SYS</pre> * ERROR at line 1: ORA-28050: specified user or role cannot be dropped	支持	-

2.9 SQL 查询和子查询

GaussDB数据库兼容除分层查询以外的SQL查询和子查询。

表 2-45 SQL 查询和子查询

序号	Oracle数据库	GaussDB数据库	差异
1	创建简单查询	支持	-
2	分层查询	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
3	UNION [ALL], INTERSECT, 减运算符	支持	-
4	查询结果排序	支持，有差异	GaussDB查询不包含分组，且目标列同时包含聚集函数和集合返回函数时，不忽略对集合返回函数列的排序。
5	Joins	支持，有差异	GaussDB只支持和Oracle相同的Join Types，如left、right、self、natural、full outer join等。不支持In-Memory Join Groups等Join Optimizations方法。
6	使用子查询	支持	-
7	嵌套子查询的解嵌套	支持，有差异	GaussDB不支持显式指定HASH_AJ或MERGE_AJ。
8	分布式查询	支持，有差异	GaussDB需要显式DBLINK查询。
9	聚集函数嵌套	支持	-

2.10 PL/SQL 语言

GaussDB数据库基本兼容的PL/SQL操作符、表达式，控制语句、集合和record等等，不支持预定义的PL/SQL常量和类型、子类型等。

2.10.1 PL/SQL 基本语法

表 2-46 PL/SQL 操作符

序号	Oracle数据库	GaussDB数据库
1	+	支持
2	:=	支持
3	=>	支持
4	%	支持
5	'	支持
6	.	支持
7		支持
8	/	支持
9	**	不支持

序号	Oracle数据库	GaussDB数据库
10	(	支持
11	)	支持
12	:	支持
13	,	支持
14	<<	支持
15	>>	支持
16	/*	支持
17	*/	支持
18	*	支持
19	"	支持
20	..	支持
21	=	支持
22	<>	支持
23	!=	支持
24	~=	支持
25	^=	支持
26	<	支持
27	>	支持
28	<=	支持
29	>=	支持
30	@	支持
31	--	支持
32	;	支持
33	-	支持

表 2-47 逻辑运算符

序号	Oracle数据库	GaussDB数据库
1	NOT	支持
2	AND	支持

序号	Oracle数据库	GaussDB数据库
3	OR	支持

表 2-48 比较表达式

序号	Oracle数据库	GaussDB数据库
1	IS [NOT] NULL	支持
2	LIKE	支持
3	BETWEEN	支持
4	IN	支持

表 2-49 条件表达式

序号	Oracle数据库	GaussDB数据库
1	simple CASE	支持
2	searched CASE	支持

表 2-50 变量声明相关参数

序号	Oracle数据库	GaussDB数据库	差异
1	%TYPE	支持，有差异	- GaussDB不支持record变量%type。 - GaussDB不支持pkg.record变量%type、schema.pkg.record变量%type作为出入参类型。 - GaussDB不支持表/视图.column.column %type、schema.表/视图.column.column %type嵌套1层及以上，作为变量类型或者出入参类型。 - GaussDB不支持record变量.column.column %type、pkg.record变量.column.column %type嵌套1层及以上的record的某列类型，作为变量类型或者出入参类型。

序号	Oracle数据库	GaussDB数据库	差异
2	%ROWTYPE	支持，有差异	- GaussDB在多个CN的环境下，存储过程中无法声明临时表的%ROWTYPE及%TYPE属性。因为临时表仅在当前session有效，在编译阶段其他CN无法看到当前CN的临时表。故多个CN的环境下，会提示该临时表不存在。 - GaussDB不支持view%rowtype、schema.view%rowtype作为出入参类型。 - GaussDB不支持package.cursor变量%rowtype作为出入参类型。

2.10.2 数据类型兼容性

表 2-51 其他 PL/SQL 数据类型

序号	Oracle数据库	GaussDB数据库	差异
1	CHARACTER	支持，有差异	- GaussDB中字节长度限制为：1~10485760。 - Oracle中字节长度限制为：1~32767。
2	VARCHAR	支持，有差异	- GaussDB中字节长度限制为：1~10485760。 - Oracle中字节长度限制为：1~32767。
3	STRING	不支持	-
4	PLS_INTEGER	不支持	GaussDB中可使用INT类型替代。
5	BINARY_INTEGER	支持	-

2.10.3 控制语句

表 2-52 条件语句

序号	Oracle数据库	GaussDB数据库
1	IF THEN	支持
2	IF THEN ELSE	支持
3	IF THEN ELSIF	支持

序号	Oracle数据库	GaussDB数据库
4	simple CASE: CASE selector WHEN selector_value_1 THEN statements_1 WHEN selector_value_2 THEN statements_2 ... WHEN selector_value_n THEN statements_n [ELSE else_statements END CASE;]	支持
5	searched CASE: CASE WHEN condition_1 THEN statements_1 WHEN condition_2 THEN statements_2 ... WHEN condition_n THEN statements_n [ELSE else_statements END CASE;]	支持

表 2-53 LOOP 循环语句

序号	Oracle数据库	GaussDB数据库
1	[label] LOOP statements END LOOP [label];	支持
2	EXIT;	支持
3	EXIT WHEN;	支持
4	CONTINUE;	支持
5	CONTINUE WHEN;	支持

表 2-54 FOR 循环语句

序号	Oracle数据库	GaussDB数据库	差异
1	[label] FOR index IN [REVERSE] lower_bound..upper_bound LOOP statements END LOOP [label];	支持，有差异	GaussDB使用REVERSE关键字时，lower_bound必须大于等于upper_bound，否则循环体不会被执行。
2	EXIT WHEN;	支持	-
3	CONTINUE WHEN;	支持	-

表 2-55 WHILE LOOP 循环语句

序号	Oracle数据库	GaussDB数据库
1	[label] WHILE condition LOOP statements END LOOP [label];	支持

表 2-56 GOTO 语句

序号	Oracle数据库	GaussDB数据库
1	GOTO	支持

表 2-57 NULL 语句

序号	Oracle数据库	GaussDB数据库
1	NULL	支持

2.10.4 集合和 Record

表 2-58 类型

序号	Oracle数据库	GaussDB数据库
1	Associative array (or index-by table)	支持

序号	Oracle数据库	GaussDB数据库
2	VARRAY (variable-size array)	支持
3	Nested table	支持
4	record	支持

表 2-59 语法

序号	Oracle数据库	GaussDB数据库	差异
1	Associative array (or index-by table)语法： TABLE OF datatype [NOT NULL] INDEX BY { PLS_INTEGER BINARY_INTEGER VARCHAR2 (v_size) data_type }	支持，有差异	- GaussDB不支持 PLS_INTEGER类型，GaussDB内data_type可以为基础数据类型、或存储过程内定义的record类型、集合类型、数组类型，不支持ref cursor类型。 - GaussDB内NOT NULL只支持语法不支持功能，即不会校验元素是否为NULL。 - 详情可参考《开发指南》中“存储过程 > 数组、集合和 record > 集合”章节。
2	VARRAY (variable-size array)语法： { VARRAY [VARYING] ARRAY } (size_limit) OF datatype [NOT NULL]	支持，有差异	- GaussDB内不支持NOT NULL语法。 - GaussDB内不支持datatype为varray类型（varray不能嵌套）。 - size_limit功能生效需要在behavior_compat_optionsGUC参数中开启varray_compat参数。 - 详情可参考《开发指南》中“存储过程 > 数组、集合和 record > 数组”章节。
3	Nested table语法： TABLE OF datatype [NOT NULL]	支持，有差异	- GaussDB内NOT NULL只支持语法不支持功能。 - 详情可参考《开发指南》中“存储过程 > 数组、集合和 record > 集合”章节。

序号	Oracle数据库	GaussDB数据库	差异
4	record语法： TYPE record_type IS RECORD (field_definition [, field_definition]...) ;	支持	- record的列可以定义为NOT NULL属性也可以指定默认值。其他类型嵌套record类型，record类型的默认值和NOT NULL不生效；通过package.record_type访问类型的形式来创建record变量，该record变量的默认值和NOT NULL不生效。 - 详情可参考《开发指南》中“存储过程 > 数组、集合和record > record ”章节。

表 2-60 构造器

序号	Oracle数据库	GaussDB数据库
1	collection_type ([value [, value]...])	支持

表 2-61 变量赋值

序号	Oracle数据库	GaussDB数据库	差异
1	Associative array (or index-by table)	支持	-
2	VARRAY (variable- size array)	支持，有差异	- GaussDB不同VARRAY类型的数据可以相互赋值，取决于其元素之间是否能相互隐式转换。 - 详情可参考《开发指南》中“存储过程 > 数组、集合和record > 数组 ”章节。
3	Nested table	支持	-
4	record	支持，有差异	- GaussDB不同record类型的数据可以相互赋值，取决于列与列之间是否能相互隐式转换。 - 详情可参考《开发指南》中“存储过程 > 数组、集合和record > record ”章节。

表 2-62 集合操作符

序号	Oracle数据库	GaussDB数据库	差异
1	=	支持，有差异	- Oracle：比较时忽略集合成员先后顺序。 - GaussDB：比较时严格按照集合成员先后顺序。
2	<>	支持，有差异	- Oracle：比较时忽略集合成员先后顺序。 - GaussDB：比较时严格按照集合成员先后顺序。
3	IS[NOT] NULL	支持	-
4	^=	支持，有差异	- Oracle：比较时忽略集合成员先后顺序。 - GaussDB：比较时严格按照集合成员先后顺序。
5	~=	不支持	-
6	IS[NOT] A SET	不支持	-
7	IS [NOT] EMPTY	不支持	-
8	expr [NOT] MEMBER [OF] nested_table	不支持	-
9	nested_table1 [NOT] SUBMULTISET [OF] nested_table2	不支持	-
10	[NOT] IN	支持	- Oracle：比较时忽略集合成员先后顺序。 - GaussDB：比较时严格按照集合成员先后顺序。

表 2-63 集合 MULTISET 函数

序号	Oracle数据库	GaussDB数据库
1	MULTISET UNION [ALL DISTINCT]	支持

序号	Oracle数据库	GaussDB数据库
2	MULTISET EXCEPT [ALL DISTINCT]	支持
3	MULTISET INTERSECT [ALL DISTINCT]	支持

表 2-64 集合类型函数

序号	Oracle数据库	GaussDB数据库	差异
1	exists(idx)	支持	-
2	extend[(count[, idx])]	支持，有差异	GaussDB仅支持nesttable类型。
3	delete[(idx1[, idx2])]	支持	-
4	trim[(n)]	支持，有差异	GaussDB仅支持nesttable类型。
5	count	支持	-
6	first	支持	-
7	last	支持	-
8	prior(idx)	支持	-
9	next(idx)	支持	-
10	limit	支持，有差异	GaussDB仅支持nesttable类型。

表 2-65 record 变量操作

序号	Oracle数据库	GaussDB数据库
1	构造器	支持
2	%ROWTYPE声明变量	支持
3	定义常量constant	不支持

表 2-66 集合相关函数

序号	Oracle数据库	GaussDB数据库	差异
1	unnest_table(anynesttable)	支持	-
2	unnest_table(anyindexbyteable)	支持	-
3	table(anyarray)	不支持	GaussDB使用unnest(anyarray)函数进行等价改写。

2.10.5 静态 SQL

表 2-67 静态查询 SQL 语句

序号	Oracle数据库	GaussDB数据库	差异
1	SELECT	支持，有差异	GaussDB和Oracle在某些场景下有不同。 GaussDB中FOR SHARE对检索出来的行加共享锁，不同事务的共享锁不会互相阻塞。若数据在一个事务中被FOR SHARE锁定，在另一个事务中使用SELECT FOR SHARE SKIP LOCKED时，SKIP LOCKED不会跳过锁。

表 2-68 静态 DML SQL 语句

序号	Oracle数据库	GaussDB数据库	差异
1	INSERT	支持，有差异	Oracle 允许目标表的列数多于子查询结果的列数，但必须显式指定要插入的列名，以确保列数匹配。GaussDB可以省略显式指定要插入的列名，该情况下子查询结果的第1列值插入到目标表的第1列中，依此类推，目标表多出的列会被插入NULL值（如果该列允许为NULL）或默认值（如果该列有默认值）。
2	UPDATE	支持	-
3	DELETE	支持	-
4	MERGE	支持	-
5	LOCK TABLE	支持	-

序号	Oracle数据库	GaussDB数据库	差异
6	INSERT ALL	支持，有差异	• Oracle不支持对into_clause的表设置别名，GaussDB支持。 • into_clause指定sequence： – Oracle：首次引用nextval会生成下一个数字，但所有非首次引用的nextval都将返回相同数字。 – GaussDB：引用nextval生成的数字可以正常自增。 • Oracle设置plan_hint语句可以正常生效，GaussDB不生效。 • Oracle允许目标表的列数多于子查询结果的列数，但必须显式指定要插入的列名，以确保列数匹配。GaussDB可以省略显式指定要插入的列名，该情况下子查询结果的第1列值插入到目标表的第1列中，依此类推，目标表多出的列会被插入NULL值（如果该列允许为NULL）或默认值（如果该列有默认值）。

表 2-69 静态 TCL SQL 语句

序号	Oracle数据库	GaussDB数据库	差异
1	COMMIT	支持	-
2	ROLLBACK	支持	-
3	SAVEPOINT	支持	-
4	SET TRANSACTION	支持，有差异	GaussDB不支持NAME string语法、USE ROLLBACK SEGMENT rollback_segment语法。

表 2-70 伪列

序号	Oracle数据库	GaussDB数据库	差异
1	CURRVAL and NEXTVAL	支持	-
2	LEVEL	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
3	OBJECT_VALUE	不支持	-
4	ROWID	不支持	-
5	ROWNUM	支持，有差异	不推荐ROWNUM条件用于JOIN ON子句。 GaussDB中ROWNUM条件用于JOIN ON子句时在LEFT JOIN、RIGHT JOIN、FULL JOIN场景下和MERGE INTO场景下与其他数据库行为不一致，直接进行业务迁移存在风险。

表 2-71 隐式游标属性

序号	Oracle数据库	GaussDB数据库	差异
1	SQL%FOUND	支持，有差异	GaussDB在commit\rollback之后不刷新隐式游标结果，Oracle会在commit\rollback之后刷新隐式游标结果。
2	SQL%NOTFOUND	支持，有差异	
3	SQL%ROWCOUNT	支持，有差异	
4	SQL%ISOPEN	支持，有差异	
5	SQL%BULK_ROWCOUNT	不支持	
6	SQL%BULK_EXCEPTIONS	不支持	

表 2-72 显式游标语法及关键字

序号	Oracle数据库	GaussDB数据库	差异
1	CURSOR cursor_name [parameter_list] RETURN return_type;	支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	CURSOR cursor_name [parameter_list] [RETURN return_type] IS select_statement;	支持	-
3	OPEN	支持	-
4	CLOSE	支持，有差异	GaussDB在exception内部会自动关闭，Oracle在exception内部不会自动关闭。
5	FETCH	支持	-
6	CURRENT OF CURSOR	支持	-

表 2-73 显式游标属性

序号	Oracle数据库	GaussDB数据库
1	SQL%FOUND	支持
2	SQL%NOTFOUND	支持
3	SQL%ROWCOUNT	支持
4	SQL%ISOPEN	支持

表 2-74 游标循环

序号	Oracle数据库	GaussDB数据库
1	FOR LOOP	支持，有差异。 对于FORALL + BULK COLLECT INTO场景， INTO变量在GaussDB中仅返回单条DML语句执行结果，在Oracle中返回DML语句累计执行结果。

表 2-75 自治事务支持场景

序号	Oracle数据库	GaussDB数据库
1	存储过程	支持
2	匿名块	支持
3	函数	支持

序号	Oracle数据库	GaussDB数据库
4	Package	支持

2.10.6 动态 SQL

表 2-76 动态 SQL 语句执行方式

序号	Oracle数据库	GaussDB数据库	差异
1	EXECUTE IMMEDIATE	支持，有差异	- GaussDB通过dynamic_sql_compat参数控制同名变量是否读取同一参数，并且检查调用存储过程时绑定参数出入参类型是否与语句参数类型一致。 - GaussDB不支持调用匿名块中部分绑定参数场景，例如匿名块中嵌套动态语句，使用表达式绑定参数，具体请参见《开发指南》中“存储过程 > 动态语句 > 动态调用匿名块”章节。 - GaussDB不支持RETURNING、RETURN INTO。
2	OPEN FOR、 FETCH 、 CLOSE	支持	GaussDB通过dynamic_sql_compat参数控制同名变量是否读取同一参数，并且检查调用存储过程时绑定参数出入参类型是否与语句参数类型一致。

2.10.7 Trigger

表 2-77 trigger 类型

序号	Oracle数据库	GaussDB数据库	差异
1	DML TRIGGER	支持，有差异	GaussDB不支持Compound DML Triggers。
2	SYSTEM TRIGGER	不支持	-

表 2-78 create trigger

序号	Oracle数据库	GaussDB数据库	差异
1	create语法： CREATE [OR REPLACE] [EDITIONABLE NONEDITIONABLE] TRIGGER plsql_trigger_source	支持，有差异	GaussDB不支持OR REPLACE、 EDITIONABLE NONEDITIONABLE，支持 plsql_trigger_source部分行为。
2	plsql_trigger_source ::= 语法： [schema.] trigger_name [sharing_clause] [default_collation_clause] { simple_dml_trigger instead_of_dml_trigger compound_dml_trigger system_trigger }	支持，有差异	GaussDB不支持schema、 sharing_clause、 default_collation_clause。
3	simple_dml_trigger ::= 语法： { BEFORE AFTER } dml_event_clause [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [ENABLE DISABLE] [WHEN (condition)] trigger_body	支持，有差异	GaussDB不支持 referencing_clause、 referencing_clause（用from referencing_table代替）、 trigger_edition_clause、 trigger_ordering_clause、 ENABLE DISABLE，支持 trigger_body部分行为。 GaussDB在没有INSTEAD OF TRIGGER的视图上创建语句级 BEFORE/AFTER TRIGGER时不会 报错，执行DML时报错。

序号	Oracle数据库	GaussDB数据库	差异
4	dml_event_clause ::=语法： { DELETE INSERT UPDATE [OF column [, column]...] } [OR { DELETE INSERT UPDATE [OF column [, column]...] }... ON [schema.] { table view }	不支持	-
5	trigger_body ::=语法： { plsql_block CALL routine_clause }	支持，有差异	GaussDB不支持plsql_block，仅支持EXECUTE PROCEDURE function_name (arguments);方式执行function，并且function需要用户定义，必须声明为不带参数并返回类型为触发器，在触发器触发时执行。
6	instead_of_dml_trigger ::=语法： INSTEAD OF { DELETE INSERT UPDATE } [OR { DELETE INSERT UPDATE }]... ON [NESTED TABLE nested_table_column OF] [schema.] nonconditioning_view [referencing_clause] [FOR EACH ROW] [trigger_edition_claus e] [trigger_ordering_clau se] [ENABLE DISABLE] trigger_body	支持，有差异	GaussDB不支持NESTED TABLE nested_table_column OF、referencing_clause、trigger_edition_clause、trigger_ordering_clause、ENABLE DISABLE。

序号	Oracle数据库	GaussDB数据库	差异
7	compound_dml_trigger ::=语法： CREATE trigger FOR dml_event_clause ON view COMPOUND TRIGGER INSTEAD OF EACH ROW IS BEGIN statement; END INSTEAD OF EACH ROW;	不支持	-
8	system_trigger ::=语 法： { BEFORE AFTER INSTEAD OF } { ddl_event [OR ddl_event]... database_event [OR database_event]... } ON { [schema.] SCHEMA [PLUGGABLE] DATABASE } [trigger_ordering_clau se] [ENABLE DISABLE] trigger_body	不支持	-

表 2-79 alter trigger

序号	Oracle数据库	GaussDB数据库	差异
1	ALTER TRIGGER [schema.] trigger_name { trigger_compile_clause { ENABLE DISABLE } RENAME TO new_name { EDITIONABLE NONEDITIONABLE } };	支持，有差异	GaussDB不支持schema、 trigger_compile_clause、 { ENABLE DISABLE }、 { EDITIONABLE NONEDITIONABLE }。

表 2-80 drop trigger

序号	Oracle数据库	GaussDB数据库	差异
1	DROP TRIGGER [schema.] trigger ;	支持，有差异	GaussDB不支持schema，需要在trigger_name后面加上ON table_name。

Oracle数据库名为*_TRIGGERS的视图统计了trigger的相关信息，GaussDB相关视图与Oracle存在差异，GaussDB视图具体请参见《开发指南》中“系统表和系统视图 > 系统视图 > 其他系统视图”中的DB_TRIGGERS、ADM_TRIGGERS、MY_TRIGGERS章节。

表 2-81 Nested, Package, and Standalone Subprograms 兼容性

序号	Oracle数据库	GaussDB数据库	差异
1	nested subprogram（子块）	支持，有差异	GaussDB不支持重载。 GaussDB不支持定义为自治事务。 GaussDB不支持SETOF的使用，仅限一个限定符引用嵌套子程序或嵌套子程序的变量。
2	package subprogram	支持	-
3	standalone subprogram（包含Function & Procedure）	支持	-
4	匿名块	支持	-

表 2-82 RETURN 语句支持情况

序号	Oracle数据库	GaussDB数据库
1	Function	支持
2	Procedure	支持
3	匿名块	支持

表 2-83 Function 相关参数

序号	Oracle数据库	GaussDB数据库	差异
1	DETERMINISTIC	支持，有差异	GaussDB中为IMMUTABLE。
2	PARALLEL_ENABLED	不支持	-
3	PIPELINED	不支持	-
4	RESULT_CACHE	不支持	-

表 2-84 参数形式支持

序号	Oracle数据库	GaussDB数据库
1	IN	支持
2	OUT	支持
3	IN OUT	支持

表 2-85 CREATE 语句

序号	Oracle数据库	GaussDB数据库	差异
1	CREATE FUNCTION	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持sharing_clause、仅支持部分指定function属性的子句（属性的子句仅支持invoker_rights_clause子句）、不支持关键字[EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中“SQL参考 > SQL语法 > C > CREATE FUNCTION ” 章节。
2	CREATE LIBRARY	不支持	-
3	CREATE PACKAGE	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持sharing_clause、仅支持部分指定package属性的子句（属性的子句仅支持invoker_rights_clause子句）、不支持关键字[EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中“SQL参考 > SQL语法 > C > CREATE PACKAGE ” 章节。

序号	Oracle数据库	GaussDB数据库	差异
4	CREATE PACKAGE BODY	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持 sharing_clause、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > C >CREATE PACKAGE ” 章节。
5	CREATE PROCEDURE	支持，有差异	GaussDB不支持IF NOT EXISTS语法、不支持 sharing_clause子句、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > C >CREATE PROCEDURE ” 章节。
6	CREATE TRIGGER	支持，有差异	GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > C >CREATE TRIGGER ” 章节。
7	CREATE TYPE	支持，有差异	GaussDB不支持varray、object类型、UNDER 语法。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > C >CREATE TYPE ” 章节。
8	CREATE TYPE BODY	不支持	-

表 2-86 ALTER 语句

序号	Oracle数据库	GaussDB数据库	差异
1	ALTER FUNCTION	支持，有差异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、SETTINGS、DEBUG。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > A > ALTER FUNCTION ” 章节。
2	ALTER LIBRARY	不支持	-
3	ALTER PACKAGE	支持，有差异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、SETTINGS、DEBUG。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > A > ALTER PACKAGE ” 章节。

序号	Oracle数据库	GaussDB数据库	差异
4	ALTER PROCEDURE	支持，有差异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE、SETTINGS、DEBUG。
5	ALTER TRIGGER	支持，有差异	GaussDB仅支持修改trigger名字。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > A > ALTER TRIGGER ” 章节。
6	ALTER TYPE	支持，有差异	GaussDB仅支持部分语句。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > A > ALTER TYPE ” 章节。

表 2-87 DROP 语句

序号	Oracle数据库	GaussDB数据库	差异
1	DROP FUNCTION	支持	-
2	DROP LIBRARY	不支持	-
3	DROP PACKAGE	支持	-
4	DROP PROCEDURE	支持	-
5	DROP TRIGGER	支持，有差异	GaussDB的语法与Oracle不同。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > D > DROP TRIGGER ” 章节。
6	DROP TYPE	支持，有差异	GaussDB不支持关键字FORCE、VALIDATE。 GaussDB的具体语法请参见《开发指南》中”SQL参考 > SQL语法 > D > DROP TYPE ” 章节。
7	DROP TYPE BODY	不支持	-

表 2-88 Function、Procedure、匿名块相关关键字

序号	Oracle数据库	GaussDB数据库	差异
1	ACCESSIBLE BY	不支持	-
2	AGGREGATE	支持，有差异	- GaussDB不支持Oracle的aggregate using [schema.] implementation_type用法。 - GaussDB的具体用法请参见《开发指南》中“SQL参考 > SQL语法 > C > CREATE AGGREGATE”章节。 语法不同，但实现功能相同。
3	DETERMINISTIC	支持，有差异	GaussDB仅在语法上支持关键字 DETERMINISTIC，未实现功能。
4	PIPE ROW	不支持	-
5	PIPELINED	不支持	-
6	SQL_MACRO	不支持	-
7	RESTRICT_REFERENCES	不支持	-
8	INLINE	不支持	-

表 2-89 异常处理相关关键字

序号	Oracle数据库	GaussDB数据库	差异
1	EXCEPTION_INIT	支持，有差异	GaussDB不支持与系统错误码进行绑定。
2	Exception	支持	-
3	Exception Handler	支持	-
4	SQLCODE	支持	-
5	SQLERRM	支持	-

表 2-90 其他 PL/SQL 关键字

序号	Oracle数据库	GaussDB数据库
1	COVERAGE	不支持
2	COLLATION	支持

序号	Oracle数据库	GaussDB数据库
3	DEPRECATE	不支持
4	FORALL	支持
5	NOCOPY	不支持
6	RETURNING	支持
7	SERIALLY_REUSABLE	不支持
8	SHARING	不支持
9	BULK COLLECT	支持

2.11 系统函数

兼容函数分为：单行函数、用户自定义函数、AGG函数、分析函数、对象引用函数、模型函数、OLAP函数。

2.11.1 单行函数

序号	Oracle数据库	GaussDB数据库
1	数值函数	支持，有差异
2	返回字符值的字符函数	支持，有差异
3	返回数值的字符函数	支持，有差异
4	字符集函数	不支持
5	Collation函数	不支持
6	日期时间函数	支持，有差异
7	通用比较函数	支持，有差异
8	转换函数	支持，有差异
9	大对象函数	支持，有差异
10	集合函数	不支持
11	层次函数	支持
12	数据挖掘功能函数	不支持
13	XML类型函数	支持，有差异
14	JSON函数	不支持
15	编码解码函数	支持，有差异
16	空值相关的函数	支持

序号	Oracle数据库	GaussDB数据库
17	环境和标识符函数	支持，有差异

表 2-91 数值函数

序号	Oracle数据库	GaussDB数据库	差异
1	ABS	支持	-
2	ACOS	支持	-
3	ASIN	支持	-
4	ATAN	支持	-
5	ATAN2	支持	-
6	BITAND	支持	-
7	CEIL	支持	-
8	COS	支持	-
9	COSH	支持	-
10	EXP	支持	-
11	FLOOR	支持	-
12	LN	支持	-
13	LOG	支持	-
14	MOD	支持，有差异	- 返回类型不一致，Oracle数据库类型包括 BINARY_DOUBLE, BINARY_FLOAT, NUMBER; GaussDB返回类型包括INT2, INT4, INT8, NUMERIC。 - 当第一个入参为数值类型时，第二个参数必须为INT、NUMERIC类型或能够转换为NUMERIC的类型。在a_format_version为10c, a_format_dev_version为s6时，当第一个参数为能够转为NUMERIC的TEXT类型时，第二个参数必须为不超过INT4的数值类型。
15	NANVL	支持，有差异	GaussDB不支持直接声明或浮点数除0得到NaN。
16	POWER	支持	-

序号	Oracle数据库	GaussDB数据库	差异
17	REMAINDER	支持，有差异	返回值数据类型不同。 GaussDB: - 当一个输入是FLOAT4时，另一个是NUMERIC时，返回FLOAT4类型。 - 当两个输入都是FLOAT4时，返回FLOAT4类型。 - 当两个输入都是FLOAT8时，返回FLOAT8类型。 - 其他数据类型，返回NUMERIC。 Oracle: 返回值类型是NUMBER。
18	ROUND	支持，有差异	- 第一个参数n的FLOAT类型，GaussDB存在精度损失，比Oracle数据库精度低。 - 返回类型不一致。round(n, integer)形式，Oracle数据库NUMBER类型，GaussDB返回NUMERIC类型；round(n)形式，Oracle数据库n的数据类型，GaussDB只能返回FLOAT8和NUMERIC类型，缺少FLOAT4返回类型。 - GaussDB判断入参有null，执行框架返回null的逻辑与Oracle数据库不一致。 SELECT round(NULL,'q'); Oracle数据库null，GaussDB报错invalid input syntax for integer: "q"。
19	SIGN	支持	-
20	SIN	支持	-
21	SINH	支持	-
22	SQRT	支持	-
23	TAN	支持	-
24	TANH	支持，有差异	返回值数据类型不同。 GaussDB: - 当输入是FLOAT8时，返回FLOAT8类型。 - 当输入是NUMERIC时，返回NUMERIC类型。 Oracle: 返回值类型是NUMBER。
25	TRUNC	支持	-
26	WIDTH_BUCKET	支持	-

表 2-92 返回字符值的字符函数

序号	Oracle数据库	GaussDB数据库	差异
1	CHR	支持，有差异	- 输入的数字不符合现有字符集时，在JDBC下GaussDB会报错，Oracle数据库会返回乱码。 - 输入0、256等时Oracle数据库会返回ASCII码为0的字符，GaussDB会在'\0'处截断。
2	CONCAT	支持	-
3	INITCAP	支持，有差异	返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
4	LOWER	支持，有差异	- 返回值类型不一致，Oracle数据库和输入类型一致的数据类型。 - 对时间格式上隐式转换问题，输入时间类型时，隐式转换为字符串再进行lower操作。 <code>SELECT LOWER(TO_DATE('2012-12-10','YYYY-MM-DD'));</code> Oracle返回10-DEC-12，GaussDB返回2012-12-10 00:00:00。 - 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
5	LPAD	支持	-
6	LTRIM	支持，有差异	返回值类型不一致。 当输入为字符数据类型时，Oracle返回VARCHAR2类型；输入为数据库创建时指定的国家字符集时，Oracle返回NVARCHAR2类型；输入为LOB类型时，Oracle返回LOB类型，GaussDB返回TEXT类型。
7	NCHR	支持，有差异	- 返回值字节长度与Oracle数据库不一致。 - 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。 - 返回入参对应的字节数组时，单个字节在[0x80-0xFF]范围，会返回“？”，Oracle数据库返回“？”、或者不输出、或者会报错。
8	NLS_LOWER	支持，有差异	- 返回值类型不同，输入为字符数据类型时，Oracle返回VARCHAR2类型；输入为LOB类型时，Oracle返回LOB类型，GaussDB返回TEXT类型。 - nlsparam参数Oracle数据库还可以传入除nls_sort外的其他参数种类而不报错，GaussDB只支持nls_sort。 - 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。

序号	Oracle数据库	GaussDB数据库	差异
9	NLS_UPPER	支持，有差异	- 返回值类型不同，输入为字符数据类型时，Oracle返回VARCHAR2类型；输入为LOB类型时，Oracle返回LOB类型，GaussDB返回TEXT类型。 - nlsparam参数Oracle数据库还可以传入除nls_sort外的其他参数种类而不报错，GaussDB只支持nls_sort。 - 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
10	NLSSORT	支持	-
11	REGEXP_REPLACE	支持，有差异	- GaussDB入参source_char不支持NCLOB类型。 - 入参match_param选项 'n' 含义有差异：GaussDB中 'n' 选项与 'm' 选项含义相同，表示采用多行模式匹配；而Oracle表示 (.) 能匹配 '\n' 字符，没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项，不需要指定选项。 - 不同正则表达式匹配结果可能不一致。 SELECT REGEXP_REPLACE('abc0123xyz', '(.*?)(\d+)(.*)', '#', 'g') FROM DUAL; Oracle报错，GaussDB返回#####xyz。 - 在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。 - 包含部分转义字符的正则表达式匹配结果可能不一致。 SELECT REGEXP_REPLACE('abcabc', 'abc', '#', 'g') FROM DUAL; Oracle报错，GaussDB返回abcabc。 - 匹配规则受aformat_regexp_match参数影响，具体影响规格请参见《开发指南》中“SQL参考 > 函数和操作符 > 字符处理函数和操作符”章节REGEXP_REPLACE函数部分。
12	REGEXP_SUBSTR	支持，有差异	匹配规则受aformat_regexp_match参数影响，具体影响规格请参见《开发指南》中“SQL参考 > 函数和操作符 > 字符处理函数和操作符”章节REGEXP_SUBSTR函数部分。
13	REPLACE	支持	-
14	RPAD	支持	-
15	RTRIM	支持	-
16	SUBSTR	支持	-

序号	Oracle数据库	GaussDB数据库	差异
17	TRANSLATE	支持	-
18	TRIM	支持	-
19	UPPER	支持，有差异	- 返回值类型不一致，Oracle数据库和输入类型一致的数据类型，GaussDB返回TEXT类型。 - 对时间格式上隐式转换问题，输入时间类型时，隐式转换为字符串再进行upper操作。 SELECT UPPER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle返回10-DEC-12，GaussDB返回2012-12-10 00:00:00。 - 返回值受限于数据库字符集，导致返回结果与Oracle数据库不一致。
20	INSTRB	支持	-

表 2-93 返回数值的字符函数

序号	Oracle数据库	GaussDB数据库	差异
1	ASCII	支持，有差异	返回值类型不同。Oracle数据库返回类型为UINT4，GaussDB为INT4。
2	INSTR	支持	-
3	LENGTH	支持	-

序号	Oracle数据库	GaussDB数据库	差异
4	REGEXP_COUNT	支持，有差异	- GaussDB入参source_char不支持NCLOB类型。 - 入参match_param选项 'n' 含义有差异：GaussDB中 'n' 选项与 'm' 选项含义相同，表示采用多行模式匹配；而 Oracle 表示 (.) 能匹配 '\n' 字符，没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项，不需要指定选项。 - 不同正则表达式匹配结果可能不一致。 - 在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。 - 包含部分转义字符的正则表达式匹配结果可能不一致。 - 匹配规则受aformat_regexp_match参数影响，具体影响规格请参见《开发指南》中“SQL参考 > 函数和操作符 > 字符处理函数和操作符”章节REGEXP_COUNT函数部分。
5	REGEXP_INSTR	支持，有差异	匹配规则受aformat_regexp_match参数影响，具体影响规格请参见《开发指南》中“SQL参考 > 函数和操作符 > 字符处理函数和操作符”章节REGEXP_INSTR函数部分。
6	LENGTHC	支持	-

表 2-94 日期时间函数

序号	Oracle数据库	GaussDB数据库	差异
1	ADD_MONTHS	支持，有差异	- 公元后到公元前，GaussDB会和Oracle数据库相差1年。 - GaussDB的计算结果范围可以到-4714年，Oracle数据库只到-4713年。
2	CURRENT_DATE	支持，有差异	GaussDB不支持nls_date_format参数设置时间显示格式。
3	CURRENT_TIMESTAMP	支持，有差异	Oracle数据库参数支持范围（0 ~ 9）。GaussDB支持范围（0 ~ 6），微秒末位零不显示。
4	DBTIMEZONE	支持，有差异	GaussDB不支持自带tz的timestamp类型接口的调用。
5	EXTRACT	支持	-

序号	Oracle数据库	GaussDB数据库	差异
6	LAST_DAY	支持，有差异	返回值类型不一致。
7	LOCALTIMESTAMP	支持，有差异	Oracle数据库参数支持范围（0 ~ 9）。GaussDB支持范围（0 ~ 6），微秒末位零不显示。
8	MONTHS_BETWEEN	支持，有差异	入参类型不一致。
9	NEW_TIME	支持，有差异	new_time函数的第一个入参为字面量时，字面量的格式以及函数的返回值类型均与Oracle数据库不一致。
10	NEXT_DAY	支持	-
11	NUMTODSINTERVAL	支持，有差异	GaussDB不支持dsinterval类型，暂时用interval兼容dsinterval类型。
12	NUMTOYMINTERVAL	支持，有差异	GaussDB不支持yminterval类型，暂时用interval兼容yminterval类型。
13	SESSIONTIMEZONE	支持，有差异	- 赋值语法差异。GaussDB为set session time zone 8。Oracle为alter session set time_zone= '+08:00'。 - 默认值差异。GaussDB为时区名称形式如:PRC。Oracle为偏移量形式，如：+08: 00。
14	SYS_EXTRACT_UTC	支持	-
15	SYSDATE	支持，有差异	返回值类型不一致。
16	SYSTIMESTAMP	支持，有差异	GaussDB毫秒计算只支持6位，Oracle数据库支持9位。
17	TO_CHAR	支持，有差异	fmt '5' 未在Oracle数据库文档中，未适配。
18	TO_DSINTERVAL	支持，有差异	GaussDB不支持dsinterval类型，暂时用interval兼容dsinterval类型。
19	TO_TIMESTAMP	支持，有差异	GaussDB毫秒计算只支持6位，Oracle数据库支持9位。
20	TO_TIMESTAMP_TZ	支持，有差异	GaussDB的timestamp_tz等价于Oracle的timestampwithlocaltimezone，缺少Oracle对应的timestamp_tz类型。nls_date_language只支持ENGLISH和AMERICAN两种语言。
21	TO_YMINTERVAL	支持，有差异	GaussDB不支持yminterval类型，暂时用interval兼容yminterval类型。

序号	Oracle数据库	GaussDB数据库	差异
22	TRUNC	支持，有差异	GaussDB返回的类型与第一个入参的类型保持一致，Oracle始终返回date类型，另外支持指定的format也有区别，具体支持的列表详见《开发指南》的“SQL参考 > 函数和操作符 > 时间和日期处理函数和操作符”章节。
23	TZ_OFFSET	支持，有差异	接收一个时区名称为入参的时候，时区名称的类型比Oracle数据库要少。

表 2-95 通用比较函数

序号	Oracle数据库	GaussDB数据库	差异
1	GREATEST	支持，有差异	- GaussDB不支持NLS_SORT参数指定的比较方式，只支持二进制比较。 - GaussDB不支持多语种的表达式。
2	LEAST	支持，有差异	- GaussDB不支持NLS_SORT参数指定的比较方式，只支持二进制比较。 - GaussDB不支持多语种的表达式。

表 2-96 转换函数

序号	Oracle数据库	GaussDB数据库	差异
1	ASCIISTR	支持	-
2	CAST	支持，有差异	- GaussDB不支持multiset子句。 - GaussDB不支持nlsparam参数。
3	HEXTORAW	支持	-
4	RAWTOHEX	支持	-
5	TO_BINARY_DOUBLE	支持，有差异	GaussDB不支持nlsparam参数。
6	TO_BINARY_FLOAT	支持，有差异	GaussDB不支持nlsparam参数。
7	TO_BLOB	支持，有差异	- GaussDB不支持long raw类型。 - GaussDB不支持bfile、mime_type类型。

序号	Oracle数据库	GaussDB数据库	差异
8	TO_CLOB	支持	-
9	TO_DATE	支持，有差异	- 不支持多语种参数。 - 返回类型不一致。 - 缺少控制参数NLS_DATE_FORMAT。 - 部分format格式不支持。 - fmt = 'j'。1582年10月15日之前Oracle数据库与GaussDB输出不一致。 - 无分割符时，不保证与Oracle数据库完全一致。如to_date('220725' , 'yymmdd')，yy/rr按照固定长度4解析，会解析为2207年25月，25非法月份则会报错。
10	TO_MULTI_BYTE	支持	-
11	TO_NCHAR	支持，有差异	- GaussDB：将入参的类型转换为text。 - Oracle：将入参的类型转换为国家字符集（ national character set ）。

序号	Oracle数据库	GaussDB数据库	差异
12	TO_NUMBER	支持，有差异	GaussDB不支持NLS_PARAM参数。 GaussDB与Oracle的format选项差异点描述： 1、\$ GaussDB不支持该format。 2、, (comma) GaussDB：逗号可以出现在format的任意位置。 Oracle： - 在format中，逗号只能出现在整数部分，且不能出现在数字开头；在原数据中，逗号可以在数字的开头位置。 - 支持format中的逗号与原数据的逗号个数和位置不一致，但最后一个逗号的位置需一致。 - 原数据和format中的连续逗号，等同于没有逗号。 - 当原数据中没有逗号时，format的最后一个逗号后面的数字位数需与原数据相等。 3、B GaussDB未实现该功能。 4、C GaussDB不支持NLS参数。 5、G GaussDB不支持NLS参数。 6、L GaussDB不支持NLS参数。 7、U GaussDB不支持NLS参数。 8、D GaussDB不支持NLS参数。 9、PR GaussDB：等同于S，返回负数。 Oracle： - 返回<尖括号>中的负值。 - 返回带前导和尾随空格的正值。 - 限制：PR格式元素只能出现在数字格式模型的最后一个位置。 10、RN rn

序号	Oracle数据库	GaussDB数据库	差异
			GaussDB未实现该功能。 TM TM9 TMe GaussDB未实现该功能。 11、V GaussDB未实现该功能。 12、FM GaussDB当有fm时，允许format中的逗号比原数据中多，不需要严格保持一致。 Oracle返回值保留前后的空格。 13、EEEE GaussDB未实现该功能。
13	TO_SINGLE_BYTE	支持	-
14	TREAT	支持，有差异	GaussDB不支持使用“.”操作符取值，不支持转化为OBJECT类型。
15	UNISTR	支持，有差异	GaussDB只支持UTF-8编码，Oracle数据库支持UTF-8和UTF-16编码。

表 2-97 大对象函数

序号	Oracle数据库	GaussDB数据库	差异
1	EMPTY_BLOB	支持	-
2	EMPTY_CLOB	支持，有差异	GaussDB的CLOB类型不支持Oracle数据库中的定位器概念。

表 2-98 层次函数

序号	Oracle数据库	GaussDB数据库
1	SYS_CONNECT_BY_PATH	不支持

表 2-99 XML 类型函数

序号	Oracle数据库	GaussDB数据库	差异
1	EXISTSNODE	支持，有差异	在入参有命名空间时，xpath和命名空间都需要定义别名。
2	EXTRACTVALUE	支持，有差异	目前仅支持xpath1.0版本。
3	SYS_XMLAGG	支持，有差异	xmlagg的别名，可使用xmlagg代替。
4	XMLAGG	支持	-
5	XMLCOMMENT	支持	-
6	XMLCONCAT	支持	-
7	XMLELEMENT	支持，有差异	xmlelement和xmlattributes的name字段赋值为NULL时，行为与Oracle不一致。 • xmlelement的name字段赋值为NULL时，结果显示name信息为空，且不显示属性信息。 • xmlattributes的name字段赋值为NULL时，不显示属性信息。
8	XMLEXISTS	支持，有差异	GaussDB入参为xml类型。
9	XMLFOREST	支持，有差异	GaussDB返回值为xml类型。 GaussDB不支持EVALNAME语法。
10	XMLPARSE	支持，有差异	GaussDB返回值为xml类型。 GaussDB不支持WELLFORMED语法。
11	XMLROOT	支持，有差异	GaussDB返回值为xml类型。
12	JSON_OBJECT	支持	-
13	XMLTABLE	支持，有差异	GaussDB从xml中选取数据使用的为XPath 1.0表达式，不支持声明默认命名空间，不支持多组输入及取别名，不支持省略传入数据的passing_clause子句，不支持RETURNING SEQUENCE BY REF子句和(SEQUENCE) BY REF子句。
14	GETSTRINGVAL	支持	-
15	GETCLOBVAL	支持	-

序号	Oracle数据库	GaussDB数据库	差异
16	XMLSEQUENCE	支持	-

表 2-100 编码解码函数

序号	Oracle数据库	GaussDB数据库	差异
1	DECODE	支持	-
2	DUMP	支持，有差异	因存储格式不同，GaussDB数值和时间类型返回结果和Oracle数据库不一致。如，GaussDB中select dump(123); 返回Typ=23 Len=4: 123,0,0,0。Oracle中select dump(123) from dual; 返回Typ=2 Len=3: 194,2,24。
3	ORA_HASH	支持，有差异	GaussDB中有以下行为： - 时间类型的入参转换成字符串类型再进行hash。 - 不支持maxbucket参数。
4	VSIZE	支持，有差异	因存储格式不同，GaussDB数值和时间类型返回结果和Oracle数据库不一致。如GaussDB中select vsize(999); 返回4。Oracle中select vsize(999) from dual; 返回3。

表 2-101 空值相关的函数

序号	Oracle数据库	GaussDB数据库
1	COALESCE	支持
2	LNNVL	支持
3	NULLIF	支持
4	NVL	支持
5	NVL2	支持

表 2-102 环境和标识符函数

序号	Oracle数据库	GaussDB数据库	差异
1	SYS_CONTEXT	支持，有差异	GaussDB对不支持的参数返回NULL。 以下为不支持的参数列表： • 'action' • 'is_application_root' • 'is_application_pdb' • 'audited_cursorid' • 'authenticated_identity' • 'authentication_data' • 'authentication_method' • 'cdb_domain' • 'cdb_name' • 'client_identifier' • 'con_id' • 'con_name' • 'current_sql_length' • 'db_domain' • 'db_supplemental_log_level' • 'dblink_info' • 'drain_status' • 'entryid' • 'enterprise_identity' • 'fg_job_id' • 'global_uid' • 'identification_type' • 'instance' • 'is_dg_rolling_upgrade' • 'ldap_server_type' • 'module' • 'network_protocol' • 'nls_calendar' • 'nls_sort' • 'nls_territory' • 'oracle_home' • 'os_user' • 'platform_slash'

序号	Oracle数据库	GaussDB数据库	差异
			• 'policy_invoker' • 'proxy_enterprise_identity' • 'proxy_user' • 'proxy_userid' • 'scheduler_job' • 'session_edition_id' • 'session_edition_name' • 'sessionid' • 'statementid' • 'terminal' • 'unified_audit_sessionid' • 'session_default_collation' • 'client_info' • 'bg_job_id' • 'client_program_name' • 'current_bind' • 'global_context_memory' • 'host' • 'current_sqln'
2	SYS_GUID	支持	-
3	USER	支持，有差异	返回值类型不一致。

序号	Oracle数据库	GaussDB数据库	差异
4	USERENV	支持，有差异	GaussDB对不支持的参数返回NULL。 以下为不支持的参数列表： • 'action' • 'is_application_root' • 'is_application_pdb' • 'audited_cursorid' • 'authenticated_identity' • 'authentication_data' • 'authentication_method' • 'cdb_domain' • 'cdb_name' • 'client_identifier' • 'con_id' • 'con_name' • 'current_sql_length' • 'db_domain' • 'db_supplemental_log_level' • 'dblink_info' • 'drain_status' • 'entryid' • 'enterprise_identity' • 'fg_job_id' • 'global_uid' • 'identification_type' • 'is_dg_rolling_upgrade' • 'ldap_server_type' • 'module' • 'network_protocol' • 'nls_calendar' • 'nls_sort' • 'nls_territory' • 'oracle_home' • 'os_user' • 'platform_slash' • 'policy_invoker' • 'proxy_enterprise_identity'

序号	Oracle数据库	GaussDB数据库	差异
			• 'proxy_user' • 'proxy_userid' • 'scheduler_job' • 'session_edition_id' • 'session_edition_name' • 'sessionid' • 'statementid' • 'terminal' • 'unified_audit_sessionid' • 'session_default_collation' • 'client_info' • 'bg_job_id' • 'client_program_name' • 'current_bind' • 'global_context_memory' • 'host' • 'current_sqln'

2.11.2 其它函数

序号	Oracle数据库	GaussDB数据库
1	聚合函数	支持
2	分析函数	支持
3	对象引用函数	不支持
4	模型函数	不支持
5	OLAP函数	不支持
6	数据盒功能函数	不支持
7	关于用户定义的函数	支持

表 2-103 聚合函数

序号	Oracle数据库	GaussDB数据库	差异
1	AVG	支持	-
2	CORR	支持	-
3	COUNT	支持	-
4	COVAR_POP	支持	-
5	COVAR_SAMP	支持	-
6	CUME_DIST	支持	-
7	DENSE_RANK	支持	-
8	FIRST	支持	GaussDB使用KEEP的语法，兼容Oracle此功能。
9	GROUPING	支持	-
10	LAST	支持	GaussDB使用KEEP的语法，兼容Oracle此功能。
11	LISTAGG	支持	-
12	MAX	支持	-
13	MEDIAN	支持	-
14	MIN	支持	-
15	PERCENT_RANK	支持	-
16	PERCENTILE_CONT	支持	-
17	RANK	支持	-
18	REGR_ (Linear Regression)	支持	-
19	STDDEV	支持	-
20	STDDEV_POP	支持	-
21	STDDEV_SAMP	支持	-
22	SUM	支持	-
23	VAR_POP	支持	-
24	VAR_SAMP	支持	-
25	VARIANCE	支持	-

序号	Oracle数据库	GaussDB数据库	差异
26	WM_CONCAT	支持	-

表 2-104 分析函数

序号	Oracle数据库	GaussDB数据库	差异
1	FIRST_VALUE	支持	-
2	LAG	支持	-
3	LAST_VALUE	支持	-
4	LEAD	支持	-
5	NTH_VALUE	支持，有差异	• Oracle：支持FROM FIRST LAST语法格式。 • GaussDB：不支持FROM FIRST LAST语法格式
6	NTILE	支持	-
7	ROW_NUMBER	支持	-
8	RATIO_TO_REPORT	支持	-

2.12 系统视图

GaussDB数据库兼容了部分Oracle数据库的系统视图，兼容的详细列表如下。

更多系统视图的字段说明信息请参考《开发指南》中“系统视图”章节。

表 2-105 系统视图支持列表

序号	Oracle数据库	GaussDB数据库
1	ALL_ALL_TABLES	DB_ALL_TABLES
2	ALL_COL_PRIVS	DB_COL_PRIVS
3	ALL_COLL_TYPES	DB_COLL_TYPES
4	ALL_IND_COLUMNS	DB_IND_COLUMNS
5	ALL_COL_COMMENTS	DB_COL_COMMENTS
6	ALL_CONS_COLUMNS	DB_CONS_COLUMNS

序号	Oracle数据库	GaussDB数据库
7	ALL_CONSTRAINTS	DB_CONSTRAINTS
8	ALL_DEPENDENCIES	DB_DEPENDENCIES
9	ALL_DIRECTORIES	DB_DIRECTORIES
10	ALL_IND_EXPRESSIONS	DB_IND_EXPRESSIONS
11	ALL_IND_PARTITIONS	DB_IND_PARTITIONS
12	ALL_INDEXES	DB_INDEXES
13	ALL_IND_SUBPARTITIONS	DB_IND_SUBPARTITIONS
14	ALL_OBJECTS	DB_OBJECTS
15	ALL_PART_COL_STATISTICS	DB_PART_COL_STATISTICS
16	ALL_PART_KEY_COLUMNS	DB_PART_KEY_COLUMNS
17	ALL_PART_TABLES	DB_PART_TABLES
18	ALL_SCHEDULER_JOB_ARGS	DB_SCHEDULER_JOB_ARGS
19	ALL_SCHEDULER_PROGRAM_ARGS	DB_SCHEDULER_PROGRAM_ARGS
20	ALL_SEQUENCES	DB_SEQUENCES
21	ALL_SUBPART_KEY_COLUMNS	DB_SUBPART_KEY_COLUMNS
22	ALL_SYNONYMS	DB_SYNONYMS
23	ALL_TAB_COL_STATISTICS	DB_TAB_COL_STATISTICS
24	ALL_TAB_COMMENTS	DB_TAB_COMMENTS
25	ALL_TAB_HISTOGRAMS	DB_TAB_HISTOGRAMS
26	ALL_TAB_STATS_HISTORY	DB_TAB_STATS_HISTORY
27	ALL_TYPES	DB_TYPES
28	ALL_PROCEDURES	DB_PROCEDURES
29	ALL_SOURCE	DB_SOURCE
30	ALL_TAB_COLUMNS	DB_TAB_COLUMNS
31	ALL_TAB_PARTITIONS	DB_TAB_PARTITIONS
32	ALL_TAB_SUBPARTITIONS	DB_TAB_SUBPARTITIONS
33	ALL_TABLES	DB_TABLES
34	ALL_TRIGGERS	DB_TRIGGERS

序号	Oracle数据库	GaussDB数据库
35	ALL_USERS	DB_USERS
36	ALL_VIEWS	DB_VIEWS
37	DBA_AUDIT_OBJECT	ADM_AUDIT_OBJECT
38	DBA_AUDIT_SESSION	ADM_AUDIT_SESSION
39	DBA_AUDIT_STATEMENT	ADM_AUDIT_STATEMENT
40	DBA_AUDIT_TRAIL	ADM_AUDIT_TRAIL
41	DBA_COL_COMMENTS	ADM_COL_COMMENTS
42	DBA_COL_PRIVS	ADM_COL_PRIVS
43	DBA_COLL_TYPES	ADM_COLL_TYPES
44	DBA_ARGUMENTS	ADM_ARGUMENTS
45	DBA_CONSTRAINTS	ADM_CONSTRAINTS
46	DBA_DATA_FILES	ADM_DATA_FILES
47	DBA_CONS_COLUMNS	ADM_CONS_COLUMNS
48	DBA_DEPENDENCIES	ADM_DEPENDENCIES
49	DBA_DIRECTORIES	ADM_DIRECTORIES
50	DBA_PART_COL_STATISTICS	ADM_PART_COL_STATISTICS
51	DBA_PART_TABLES	ADM_PART_TABLES
52	DBA_ROLE_PRIVS	ADM_ROLE_PRIVS
53	DBA_ROLES	ADM_ROLES
54	DBA_SCHEDULER_JOB_ARGS	ADM_SCHEDULER_JOB_ARGS
55	DBA_SCHEDULER_PROGRAMS	ADM_SCHEDULER_PROGRAMS
56	DBA_SCHEDULER_PROGRAM_ARGS	ADM_SCHEDULER_PROGRAM_ARGS
57	DBA_HIST_SNAPSHOT	ADM_HIST_SNAPSHOT
58	DBA_HIST_SQL_PLAN	ADM_HIST_SQL_PLAN
59	DBA_HIST_SQLSTAT	ADM_HIST_SQLSTAT
60	DBA_HIST_SQLTEXT	ADM_HIST_SQLTEXT
61	DBA_ILMDATAMOVEMENTPOLICIES	GS_ADM_ILMDATAMOVEMENTPOLICIES

序号	Oracle数据库	GaussDB数据库
62	DBA_ILMEVALUATIONDETAILS	GS_ADM_ILMEVALUATIONDETAILS
63	DBA_ILMOBJECTS	GS_ADM_ILMOBJECTS
64	DBA_ILMPARAMETERS	GS_ADM_ILMPARAMETERS
65	DBA_ILMPOLICIES	GS_ADM_ILMPOLICIES
66	DBA_ILMRESULTS	GS_ADM_ILMRESULTS
67	DBA_ILMTASKS	GS_ADM_ILMTASKS
68	DBA_IND_COLUMNS	ADM_IND_COLUMNS
69	DBA_IND_EXPRESSIONS	ADM_IND_EXPRESSIONS
70	DBA_IND_PARTITIONS	ADM_IND_PARTITIONS
71	DBA_INDEXES	ADM_INDEXES
72	DBA_OBJECTS	ADM_OBJECTS
73	DBA_PART_INDEXES	ADM_PART_INDEXES
74	DBA_PROCEDURES	ADM_PROCEDURES
75	DBA_SCHEDULER_JOBS	ADM_SCHEDULER_JOBS
76	DBA_SCHEDULER_RUNNING_JOBS	ADM_SCHEDULER_RUNNING_JOBS
77	DBA_SEGMENTS	ADM_SEGMENTS
78	DBA_SEQUENCES	ADM_SEQUENCES
79	DBA_SOURCE	ADM_SOURCE
80	DBA_IND_SUBPARTITIONS	ADM_IND_SUBPARTITIONS
81	DBA_SUBPART_KEY_COLUMNS	ADM_SUBPART_KEY_COLUMNS
82	DBA_SYS_PRIVS	ADM_SYS_PRIVS
83	DBA_TAB_COL_STATISTICS	ADM_TAB_COL_STATISTICS
84	DBA_TAB_HISTOGRAMS	ADM_TAB_HISTOGRAMS
85	DBA_TAB_STATISTICS	ADM_TAB_STATISTICS
86	DBA_TAB_STATS_HISTORY	ADM_TAB_STATS_HISTORY
87	DBA_TABLESPACES	ADM_TABLESPACES
88	DBA_TYPES	ADM_TYPES
89	DBA_USERS	ADM_USERS

序号	Oracle数据库	GaussDB数据库
90	DBA_SYNONYMS	ADM_SYNONYMS
91	DBA_TAB_COLS	ADM_TAB_COLS
92	DBA_TAB_COLUMNS	ADM_TAB_COLUMNS
93	DBA_TAB_COMMENTS	ADM_TAB_COMMENTS
94	DBA_TABLES	ADM_TABLES
95	DBA_TAB_PARTITIONS	ADM_TAB_PARTITIONS
96	DBA_TAB_SUBPARTITIONS	ADM_TAB_SUBPARTITIONS
97	DBA_TRIGGERS	ADM_TRIGGERS
98	DBA_TYPE_ATTRS	ADM_TYPE_ATTRS
99	DBA_VIEWS	ADM_VIEWS
100	ROLE_ROLE_PRIVS	ROLE_ROLE_PRIVS
101	ROLE_SYS_PRIVS	ROLE_SYS_PRIVS
102	ROLE_TAB_PRIVS	ROLE_TAB_PRIVS
103	USER_COL_COMMENTS	MY_COL_COMMENTS
104	USER_COL_PRIVS	MY_COL_PRIVS
105	USER_COLL_TYPES	MY_COLL_TYPES
106	USER_CONSTRAINTS	MY_CONSTRAINTS
107	USER_DEPENDENCIES	MY_DEPENDENCIES
108	DICT	DICT
109	DICTIONARY	DICTIONARY
110	DUAL	DUAL
111	NLS_DATABASE_PARAMETERS	NLS_DATABASE_PARAMETERS
112	NLS_INSTANCE_PARAMETERS	NLS_INSTANCE_PARAMETERS
113	PLAN_TABLE	PLAN_TABLE
114	USER_ILMDATAMOVEMENTPOLICIES	GS_MY_ILMDATAMOVEMENTPOLICIES
115	USER_ILMEVALUATIONDETAILS	GS_MY_ILMEVALUATIONDETAILS
116	USER_ILMOBJECTS	GS_MY_ILMOBJECTS
117	USER_ILMPOLICIES	GS_MY_ILMPOLICIES

序号	Oracle数据库	GaussDB数据库
118	USER_ILMRESULTS	GS_MY_ILMRESULTS
119	USER_ILMTASKS	GS_MY_ILMTASKS
120	USER_IND_COLUMNS	MY_IND_COLUMNS
121	USER_IND_EXPRESSIONS	MY_IND_EXPRESSIONS
122	USER_IND_PARTITIONS	MY_IND_PARTITIONS
123	USER_IND_SUBPARTITION S	MY_IND_SUBPARTITIONS
124	USER_INDEXES	MY_INDEXES
125	USER_JOBS	MY_JOBS
126	USER_OBJECTS	MY_OBJECTS
127	USER_PART_COL_STATISTI CS	MY_PART_COL_STATISTICS
128	USER_PART_INDEXES	MY_PART_INDEXES
129	USER_PART_TABLES	MY_PART_TABLES
130	USER_PROCEDURES	MY_PROCEDURES
131	USER_SCHEDULER_JOB_A RGs	MY_SCHEDULER_JOB_ARGS
132	USER_SCHEDULER_PROG RAM_ARGS	MY_SCHEDULER_PROGRAM_ARGS
133	USER_SEQUENCES	MY_SEQUENCES
134	USER_SOURCE	MY_SOURCE
135	USER_SUBPART_KEY_COL UMNS	MY_SUBPART_KEY_COLUMNS
136	USER_SYNONYMS	MY_SYNONYMS
137	USER_SYS_PRIVS	MY_SYS_PRIVS
138	USER_TAB_COL_STATISTIC S	MY_TAB_COL_STATISTICS
139	USER_TAB_COLUMNS	MY_TAB_COLUMNS
140	USER_TAB_COMMENTS	MY_TAB_COMMENTS
141	USER_TAB_HISTOGRAMS	MY_TAB_HISTOGRAMS
142	USER_TAB_PARTITIONS	MY_TAB_PARTITIONS
143	USER_TAB_STATISTICS	MY_TAB_STATISTICS

序号	Oracle数据库	GaussDB数据库
144	USER_TAB_STATS_HISTORY	MY_TAB_STATS_HISTORY
145	USER_TABLES	MY_TABLES
146	USER_TABLESPACES	MY_TABLESPACES
147	USER_TRIGGERS	MY_TRIGGERS
148	USER_TYPE_ATTRS	MY_TYPE_ATTRS
149	USER_TYPES	MY_TYPES
150	USER_VIEWS	MY_VIEWS
151	V\$NLS_PARAMETERS	V\$NLS_PARAMETERS
152	V\$SESSION_WAIT	V\$SESSION_WAIT
153	V\$SYSSTAT	V\$SYSSTAT
154	V\$SYSTEM_EVENT	V\$SYSTEM_EVENT
155	V\$VERSION	V\$VERSION
156	V\$INSTANCE	V_INSTANCE
157	GV\$INSTANCE	GV_INSTANCE
158	V\$MYSTAT	V_MYSTAT
159	V\$SESSION	V_SESSION
160	GV\$SESSION	GV_SESSION
161	V\$SESSION_LONGOPS	DV_SESSION_LONGOPS
162	V\$SESSION	DV_SESSIONS
163	ALL_ARGUMENTS	DB_ARGUMENTS
164	USER_CONS_COLUMNS	MY_CONS_COLUMNS
165	USER_PART_KEY_COLUMNS	MY_PART_KEY_COLUMNS
166	USER_ROLE_PRIVS	MY_ROLE_PRIVS
167	DBA_TAB_PRIVS	ADM_TAB_PRIVS
168	USER_SCHEDULER_JOBS	MY_SCHEDULER_JOBS
169	V\$LOCK	V\$LOCK
170	V\$DBLINK	V\$DBLINK
171	V\$GLOBAL_TRANSACTION	V\$GLOBAL_TRANSACTION

序号	Oracle数据库	GaussDB数据库
172	V\$OPEN_CURSOR	V\$OPEN_CURSOR
173	V\$GLOBAL_OPEN_CURSOR	V\$GLOBAL_OPEN_CURSOR
174	ALL_TAB_PRIVS	DB_TAB_PRIVS
175	ALL_TAB_MODIFICATIONS	DB_TAB_MODIFICATIONS
176	USER_TAB_MODIFICATIONS	MY_TAB_MODIFICATIONS
177	USER_AUDIT_TRAIL	MY_AUDIT_TRAIL

2.13 高级包

GaussDB数据库兼容了部分Oracle数据库的高级包，兼容的详细列表如下。

更多高级包信息请参考开发指南中高级包章节。

表 2-106 高级包支持列表

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_LOB	DBE_LOB	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_LOB”章节。
2	DBMS_RANDOM	DBE_RANDOM	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_RANDOM”章节。
3	DBMS_OUTPUT	DBE_OUTPUT	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_OUTPUT”章节。
4	UTL_RAW	DBE_RAW	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_RAW”章节。
5	DBMS_SCHEDULER	DBE_SCHEDULER	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_SCHEDULER”章节。
6	DBMS_UTILITY	DBE_UTILITY	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_UTILITY”章节。
7	DBMS_SQL	DBE_SQL	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_SQL”章节。

序号	Oracle数据库	GaussDB数据库	差异
8	UTL_FILE	DBE_FILE	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_FILE”章节。
9	DBMS_SESSION	DBE_SESSION	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_SESSION”章节。
10	UTL_MATCH	DBE_MATCH	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_MATCH”章节。
11	DBMS_APPLICATION_INFO	DBE_APPLICATION_INFO	GaussDB具体用法请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_APPLICATION_INFO”章节。
12	DBMS_XMLDOM	DBE_XMLDOM	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_XMLDOM”章节。
13	DBMS_XMLPARSER	DBE_XMLPARSER	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_XMLPARSER”章节。
14	DBMS_ILM	DBE_ILM	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_ILM”章节。
15	DBMS_ILM_ADMIN	DBE_ILM_ADMIN	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_ILM_ADMIN”章节。
16	DBMS_COMPRESSION	DBE_COMPRESSION	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_COMPRESSION”章节。
17	DBMS_HEAT_MAP	DBE_HEAT_MAP	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_HEAT_MAP”章节。
18	DBMS_DESCRIBE	DBE_DESCRIBE	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_DESCRIBE”章节。
19	DBMS_XMLGEN	DBE_XMLGEN	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_XMLGEN”章节。
20	DBMS_STATS	DBE_STATS	GaussDB中具体请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_STATS”章节。

表 2-107 DBMS_LOB 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	APPEND Procedures	APPEND Procedures	-
2	CLOB2FILE Procedure	不支持	-
3	CLOSE Procedure	BFILECLOSE Procedure	GaussDB：参数类型为BFILE，不存在函数重载。 Oracle：该过程存在3个重载，3个重载的参数lob_loc、lob_loc和file_loc的类型分别为BLOB、CLOB CHARACTER SET ANY_CS和BFILE。
4	COMPARE Functions	COMPARE Functions	GaussDB：存在3个重载函数，对于第三个参数（len）均为BIGINT。 Oracle：存在3个重载函数，对于第三个参数（amount）均为INTEGER。
5	CONVERTTOBLOB Procedure	LOB_CONVERTTOBLOB Procedure	GaussDB：该过程共有5个参数，且第3、4、5个参数类型为BIGINT。 Oracle：该过程共有8个参数，在GaussDB所有参数的基础上增加了blob_csid、lang_context和warning3个参数，类型分别为NUMBER、INTEGER和INTEGER，且第3、4、5个参数类型为INTEGER。
6	CONVERTTOCLOB Procedure	LOB_CONVERTTOCLOB Procedure	GaussDB：该过程共有5个参数。第3、4、5个参数类型为BIGINT。 Oracle：该过程共有8个参数。第3、4、5个参数类型为INTEGER。Oracle的该过程在GaussDB所有参数的基础上增加了3个参数:blob_csid、lang_context和warning，参数类型分别为NUMBER、INTEGER和INTEGER。
7	COPY Procedures	LOB_COPY Functions	-

序号	Oracle数据库	GaussDB数据库	差异
8	COPY_DBFS_LINK Procedures	不支持	-
9	COPY_FROM_DBFS_LINK	不支持	-
10	CREATE_TEMPORARY Procedures	CREATE_TEMPORARY Procedures	GaussDB: 该过程存在2个重载。第一个重载过程的第一个参数 (lob_loc) 为BLOB, 第二个重载过程的第一个参数 (lob_loc) 为CLOB; 两个重载过程的第三个参数 (dur) 为INTEGER, 默认值为10。 Oracle: 该过程存在2个重载。第一个重载过程的第一个参数 (lob_loc) 为BLOB, 第二个重载过程的第一个参数 (lob_loc) 为CLOB; 两个重载过程的第三个参数 (dur) 的参数类型为PLS_INTEGER, 第一个重载过程的dur默认值为DBMS_LOB.SESSION, 第二个重载过程的dur默认值为10。
11	DBFS_LINK_GENERATE_PATH Functions	不支持	-
12	ERASE Procedures	LOB_ERASE Procedures	-
13	FILECLOSE Procedure	不支持	-
14	FILECLOSEALL Procedure	不支持	-
15	FILEEXISTS Function	不支持	-
16	FILEGETNAME Procedure	不支持	-
17	FILEISOPEN Function	不支持	-
18	FILEOPEN Procedure	不支持	-
19	FRAGMENT_DELETE Procedure	不支持	-
20	FRAGMENT_INSERT Procedures	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
21	FRAGMENT_MOVE Procedure	不支持	-
22	FRAGMENT_REPLACE Procedures	不支持	-
23	FREETEMPORARY Procedures	不支持	-
24	GET_DBFS_LINK Functions	不支持	-
25	GET_DBFS_LINK_STATE Procedures	不支持	-
26	GETCHUNKSIZE Functions	GETCHUNKSIZE Functions	-
27	GETCONTENTTYPE Functions	不支持	-
28	GETLENGTH Functions	不支持	-
29	GETOPTIONS Functions	不支持	-
30	GET_STORAGE_LIMIT Function	不支持	-
31	INSTR Functions	MATCH Functions	GaussDB：存在3个重载函数。3个重载函数的第三、四个参数均为BIGINT。 Oracle：存在3个重载函数。3个重载函数的第三、四个参数均为INTEGER。
32	ISOPEN Functions	不支持	-
33	ISREMOTE Function	不支持	-
34	ISSECUREFILE Function	不支持	-
35	ISTEMPORARY Functions	不支持	-
36	LOADBLOBFROMFILE Procedure	LOADBLOBFROMFILE Procedure	-
37	LOADCLOBFROMFILE Procedure	LOADCLOBFROMFILE Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
38	LOADFROMFILE Procedure	LOADFROMFILE Procedure	-
39	MOVE_TO_DBFS _LINK Procedures	不支持	-
40	OPEN Procedures	BFILEOPEN Procedure	GaussDB：该过程不存在重载。 第一个参数（bfile）类型为 DBE_LOB.BFILE，第二个参数 （open_mode）类型为TEXT， 且只支持read模式。 Oracle：该过程存在3个重载。 第一个重载过程的第一个参数 （lob_loc）类型为NOCOPY BLOB，第二个参数 （openmode）类型为 BINARY_INTEGER；第二个重载 过程的第一个参数（lob_loc） 类型为NOCOPY CLOB CHARACTER SET ANY_CS，第 二个参数（openmode）类型为 BINARY_INTEGER；第三个重载 过程的第一个参数（file_loc） 类型为NOCOPY BFILE,第二个 参数（openmode）类型为 BINARY_INTEGER，且只能为 file_readonly。
41	READ Procedures	READ Procedures	GaussDB：该过程存在2个重 载。 Oracle：该过程存在3个重载。 其中前两个重载与GaussDB无差 异，第三个过程重载包括4个参 数：file_loc、amount、offset 和buffer，其类型分别为 BFILE、NOCOPY INTEGER、 INTEGER和RAW。
42	SET_DBFS_LINK Procedures	不支持	-
43	SETCONTENTTY PE Procedure	不支持	-
44	SETOPTIONS Procedures	不支持	-
45	SUBSTR Functions	LOB_SUBSTR Functions	-

序号	Oracle数据库	GaussDB数据库	差异
46	TRIM Procedures	STRIP Functions	GaussDB：该过程存在2个重载。两个重载过程的第二个参数（newlen）均为BIGINT。 Oracle：该过程存在2个重载。两个重载过程的第二个参数（newlen）均为INTEGER。
47	WRITE Procedures	WRITE Functions	-
48	WRITEAPPEND Procedures	WRITEAPPEND Functions	-

表 2-108 DBMS_RANDOM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	INITIALIZE Procedure	不支持	-
2	NORMAL Function	不支持	-
3	RANDOM Function	不支持	-
4	SEED Procedures	DBE_RANDOM.SET_SEED Function	GaussDB：该函数无重载，参数类型为INTEGER。 Oracle：该过程存在2个重载，2个重载过程的参数类型分别为VARCHAR2和BINARY_INTEGER。
5	STRING Function	不支持	-
6	TERMINATE Procedure	不支持	-
7	VALUE Functions	DBE_RANDOM.GET_VALUE Function	GaussDB：该函数无重载。 Oracle：存在无参数的VALUE函数重载，返回NUMBER类型。

表 2-109 DBMS_OUTPUT 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DISABLE Procedure	DISABLE Function	-

序号	Oracle数据库	GaussDB数据库	差异
2	ENABLE Procedure	ENABLE Function	-
3	GET_LINE Procedure	GET_LINE Function	GaussDB：该函数无重载，首个参数（lines）数据类型为 VARCHAR[]。 Oracle：该过程存在2个重载，2个重载过程的首个参数（lines）分别为CHARARR和 DBMSOUTPUT_LINESARRAY。
4	GET_LINES Procedure	GET_LINES Function	-
5	NEW_LINE Procedure	NEW_LINE Function	-
6	PUT Procedure	PUT Function	GaussDB：当数据库服务端字符集 server_encoding不是UTF8编码格式且入参的字符编码是合法的 UTF8编码时，该函数不会区分入参的数据类型，都会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出。 Oracle：当数据库服务端字符集 server_encoding不是UTF8编码格式且入参的字符编码是合法的 UTF8编码时，若入参类型是 NVARCHAR2，则该过程会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出；若入参为其他字符类型，则会将该字符编码视作非法字符，以占位符的形式输出。

序号	Oracle数据库	GaussDB数据库	差异
7	PUT_LINE Procedure	PUT_LINE Function	GaussDB：当数据库服务端字符集 server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时，该函数不会区分入参的数据类型，都会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出。 Oracle：当数据库服务端字符集 server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时，若入参类型是NVARCHAR2，则该过程会先把该字符编码按照“UTF8 > server_encoding”的转换关系进行转换后再输出；若入参为其他字符类型，则会将该字符编码视作非法字符，以占位符的形式输出。

表 2-110 UTL_RAW 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	BIT_AND Function	BIT_AND Function	-
2	BIT_COMPLEME NT Function	BIT_COMPLEMENT Function	-
3	BIT_OR Function	BIT_OR Function	GaussDB：两个参数类型被定义为TEXT类型并且返回TEXT类型。 Oracle：两个参数为RAW类型并且返回RAW类型。
4	BIT_XOR Function	BIT_XOR Function	-
5	CAST_FROM_BIN ARY_DOUBLE Function	CAST_FROM_BINAR Y_DOUBLE_TO_RA W Function	-
6	CAST_FROM_BIN ARY_FLOAT Function	CAST_FROM_BINAR Y_FLOAT_TO_RAW Function	GaussDB：参数n为FLOAT4类型。 Oracle：参数n为FLOAT类型。

序号	Oracle数据库	GaussDB数据库	差异
7	CAST_FROM_BINARY_INTEGER Function	CAST_FROM_BINARY_INTEGER_TO_RAW Function	GaussDB: 参数value为BIGINT类型。 Oracle: 参数value为INTEGER类型。
8	CAST_FROM_NUMBER Function	CAST_FROM_NUMBER_TO_RAW Function	GaussDB: 参数n为NUMERIC类型 Oracle: 参数n为NUMBER类型。
9	CAST_TO_BINARY_DOUBLE Function	CAST_FROM_RAW_TO_BINARY_DOUBLE Function	-
10	CAST_TO_BINARY_FLOAT Function	CAST_FROM_RAW_TO_BINARY_FLOAT Function	GaussDB: 函数返回类型为FLOAT4。 Oracle: 函数返回类型为FLOAT。
11	CAST_TO_BINARY_INTEGER Function	CAST_FROM_RAW_TO_BINARY_INTEGER Function	GaussDB: 参数endianess为INTEGER类型，函数返回类型为INTEGER。 Oracle: 参数endianess为PLS_INTEGER类型，函数返回类型为BINARY_INTEGER。
12	CAST_TO_NUMBER Function	CAST_FROM_RAW_TO_NUMBER Function	GaussDB: 函数返回类型为NUMERIC。 Oracle: 函数返回类型为NUMBER。
13	CAST_TO_NVARCHAR2 Function	CAST_FROM_RAW_TO_NVARCHAR2 Function	-
14	CAST_TO_RAW Function	CAST_FROM_VARCHAR2_TO_RAW Function	-
15	CAST_TO_VARCHAR2 Function	CAST_TO_VARCHAR2 Function	-
16	COMPARE Function	COMPARE Function	GaussDB: 函数返回类型为INTEGER。 Oracle: 函数返回类型为NUMBER。
17	CONCAT Function	CONCAT Function	-
18	CONVERT Function	CONVERT Function	-

序号	Oracle数据库	GaussDB数据库	差异
19	COPIES Function	COPIES Function	GaussDB: 参数n为NUMERIC类型。 Oracle: 参数n为NUMBER类型。
20	LENGTH Function	GET_LENGTH Function	-
21	OVERLAY Function	OVERLAY Function	-
22	REVERSE Function	REVERSE Function	-
23	SUBSTR Function	SUBSTR Function	GaussDB: 参数lob_loc为BLOB类型; 参数off_set为INTEGER类型, 默认值为1; 参数amount为INTEGER类型, 默认值为32767。 Oracle: 参数r为RAW类型, 参数pos为BINARY_INTEGER类型且无默认值, 参数len为BINARY_INTEGER类型, 默认值为NULL。
24	TRANSLATE Function	TRANSLATE Function	-
25	TRANSLITERATE Function	TRANSLITERATE Function	-
26	XRANGE Function	XRANGE Function	GaussDB: 参数start_byte和end_byte无默认值。 Oracle: 参数start_byte和end_byte默认为NULL。

表 2-111 DBMS_SCHEDULER 兼容性说明

序号	Oracle数据库	GaussDB数据库
1	ADD_EVENT_QUEUE_SUBSCRIBER Procedure	不支持
2	ADD_GROUP_MEMBER Procedure	不支持
3	ADD_JOB_EMAIL_NOTIFICATION Procedure	不支持
4	ADD_TO_INCOMPATIBILITY Procedure	不支持

序号	Oracle数据库	GaussDB数据库
5	ALTER_CHAIN Procedure	不支持
6	ALTER_RUNNING_CHAIN Procedure	不支持
7	CLOSE_WINDOW Procedure	不支持
8	COPY_JOB Procedure	不支持
9	CREATE_CHAIN Procedure	不支持
10	CREATE_CREDENTIAL Procedure	CREATE_CREDENTIAL Procedure
11	CREATE_DATABASE_DESTINATION Procedure	不支持
12	CREATE_EVENT_SCHEDULE Procedure	不支持
13	CREATE_FILE_WATCHER Procedure	不支持
14	CREATE_GROUP Procedure	不支持
15	CREATE_INCOMPATIBILITY Procedure	不支持
16	CREATE_JOB Procedure	CREATE_JOB Procedure
17	CREATE_JOB_CLASS Procedure	CREATE_JOB_CLASS Procedure
18	CREATE_JOBS Procedure	不支持
19	CREATE_PROGRAM Procedure	CREATE_PROGRAM Procedure
20	CREATE_RESOURCE Procedure	不支持
21	CREATE_SCHEDULE Procedure	CREATE_SCHEDULE Procedure
22	CREATE_WINDOW Procedure	不支持
23	DEFINE_ANYDATA_ARGUMENT Procedure	不支持
24	DEFINE_CHAIN_EVENT_STEP Procedure	不支持
25	DEFINE_CHAIN_RULE Procedure	不支持
26	DEFINE_CHAIN_STEP Procedure	不支持
27	DEFINE_METADATA_ARGUMENT Procedure	不支持
28	DEFINE_PROGRAM_ARGUMENT Procedure	DEFINE_PROGRAM_ARGUMENT Procedure
29	DISABLE Procedure	DISABLE Procedure

序号	Oracle数据库	GaussDB数据库
30	DROP_AGENT_DESTINATION Procedure	不支持
31	DROP_CHAIN Procedure	不支持
32	DROP_CHAIN_RULE Procedure	不支持
33	DROP_CHAIN_STEP Procedure	不支持
34	DROP_CREDENTIAL Procedure	DROP_CREDENTIAL Procedure
35	DROP_DATABASE_DESTINATION Procedure	不支持
36	DROP_FILE_WATCHER Procedure	不支持
37	DROP_GROUP Procedure	不支持
38	DROP_INCOMPATIBILITY Procedure	不支持
39	DROP_JOB Procedure	DROP_JOB Procedure
40	DROP_JOB_CLASS Procedure	DROP_JOB_CLASS Procedure
41	DROP_PROGRAM Procedure	DROP_PROGRAM Procedure
42	DROP_PROGRAM_ARGUMENT Procedure	不支持
43	DROP_SCHEDULE Procedure	DROP_SCHEDULE Procedure
44	DROP_WINDOW Procedure	不支持
45	ENABLE Procedure	ENABLE Procedure
46	END_DETACHED_JOB_RUN Procedure	不支持
47	EVALUATE_CALENDAR_STRING Procedure	EVALUATE_CALENDAR_STRING Procedure
48	EVALUATE_RUNNING_CHAIN Procedure	不支持
49	GENERATE_JOB_NAME Function	GENERATE_JOB_NAME Function
50	GET_AGENT_INFO Function	不支持
51	GET_AGENT_VERSION Function	不支持
52	GET_ATTRIBUTE Procedure	不支持
53	GET_FILE Procedure	不支持
54	GET_SCHEDULER_ATTRIBUTE Procedure	不支持
55	OPEN_WINDOW Procedure	不支持

序号	Oracle数据库	GaussDB数据库
56	PURGE_LOG Procedure	不支持
57	PUT_FILE Procedure	不支持
58	REMOVE_EVENT_QUEUE_SUBSCRIBER Procedure	不支持
59	REMOVE_FROM_INCOMPATIBILITY Procedure	不支持
60	REMOVE_GROUP_MEMBER Procedure	不支持
61	REMOVE_JOB_EMAIL_NOTIFICATION Procedure	不支持
62	RESET_JOB_ARGUMENT_VALUE Procedure	不支持
63	RUN_CHAIN Procedure	不支持
64	RUN_JOB Procedure	RUN_JOB Procedure
65	SET_AGENT_REGISTRATION_PASS Procedure	不支持
66	SET_ATTRIBUTE Procedure	SET_ATTRIBUTE Procedure
67	SET_ATTRIBUTE_NULL Procedure	不支持
68	SET_JOB_ANYDATA_VALUE Procedure	不支持
69	SET_JOB_ARGUMENT_VALUE Procedure	SET_JOB_ARGUMENT_VALUE Procedure
70	SET_JOB_ATTRIBUTES Procedure	不支持
71	SET_RESOURCE_CONSTRAINT Procedure	不支持
72	SET_SCHEDULER_ATTRIBUTE Procedure	不支持
73	STOP_JOB Procedure	STOP_JOB Procedure

表 2-112 DBMS_UTILITY 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	ACTIVE_INSTANCES Procedure	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	ANALYZE_DATABASE Procedure	不支持	-
3	ANALYZE_PART_OBJECT Procedure	不支持	-
4	ANALYZE_SCHEMA Procedure	不支持	-
5	CANONICALIZE Procedure	CANONICALIZE Procedure	GaussDB：参数 canon_len默认为1024字节。 Oracle：参数canon_len无默认值。
6	COMMA_TO_TABLE Procedures	COMMA_TO_TABLE Procedures	GaussDB：参数tab为VARCHAR2数组。 Oracle：该过程存在2个重载。参数tab可以为两种类型之一：一种为 uncl_array，另一种为 lname_array。
7	COMPILE_SCHEMA Procedure	不支持	-
8	CREATE_ALTER_TYPE_ERROR_TABLE Procedure	不支持	-
9	CURRENT_INSTANCE Function	不支持	-
10	DATA_BLOCK_ADDRESS_BLOCK Function	不支持	-
11	DATA_BLOCK_ADDRESS_FILE Function	不支持	-
12	DB_VERSION Procedure	DB_VERSION Procedure	GaussDB：只有参数 version，类型为 VARCHAR2。Oracle：有参数version和 compatibility，类型均为 VARCHAR2。
13	EXEC_DDL_STATEMENT Procedure	EXEC_DDL_STATEMENT Function	GaussDB：参数 parse_string为TEXT类型。 Oracle：参数 parse_string为 VARCHAR2类型。

序号	Oracle数据库	GaussDB数据库	差异
14	EXPAND_SQL_TEXT Procedure	EXPAND_SQL_TEX T Function	GaussDB：参数 output_sql_text为 CLOB。 Oracle：参数 output_sql_text为 NOCOPY CLOB，通过传 引用方式传递OUT参数。
15	FORMAT_CALL_STACK Function	FORMAT_CALL_ST ACK Function	GaussDB：函数返回类型 为TEXT。 Oracle：函数返回类型为 VARCHAR2。
16	FORMAT_ERROR_BACKT RACE Function	FORMAT_ERROR_ BACKTRACE Function	GaussDB：函数返回类型 为TEXT。 Oracle：函数返回类型为 VARCHAR2。
17	FORMAT_ERROR_STACK Function	FORMAT_ERROR_ STACK Function	GaussDB：函数返回类型 为TEXT。 Oracle：函数返回类型为 VARCHAR2。
18	GET_CPU_TIME Function	GET_CPU_TIME Function	GaussDB：函数返回类型 为BIGINT。 Oracle：函数返回类型为 NUMBER。
19	GET_DEPENDENCY Procedure	不支持	-
20	GET_ENDIANNES S Function	GET_ENDIANNES S Function	GaussDB：函数返回类型 为INTEGER。 Oracle：函数返回类型为 NUMBER。
21	GET_HASH_VALUE Function	GET_HASH_VALUE Function	GaussDB：参数base、 hash_size和返回类型均为 INTEGER。 Oracle：参数base、 hash_size和返回类型均为 NUMBER。
22	GET_PARAMETER_VA LUE Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
23	GET_SQL_HASH Function	GET_SQL_HASH Function	GaussDB: 参数 last4bytes 为BIGINT类型, 代表MD5哈希值的最后四字节, 以无符号整数形式展现, 函数返回类型为BIGINT。 Oracle: 对应参数 pre10ihash为NUMBER类型, 用于存储MD5计算得到的16字节中的4字节哈希值。
24	GET_TIME Function	GET_TIME Function	GaussDB: 函数返回类型为BIGINT。 Oracle: 函数返回类型为NUMBER。
25	GET_TZ_TRANSITIONS Procedure	不支持	-
26	INVALIDATE Procedure	不支持	-
27	IS_BIT_SET Function	IS_BIT_SET Function	GaussDB: 参数n和返回值类型为INTEGER。 Oracle: 参数n和返回值类型为NUMBER。
28	IS_CLUSTER_DATABASE Function	IS_CLUSTER_DATA BASE Function	-
29	MAKE_DATA_BLOCK_AD DRESS Function	不支持	-
30	NAME_RESOLVE Procedure	NAME_RESOLVE Procedure	GaussDB: 参数context和part1_type为INTEGER, 参数 object_number为OID; GaussDB不支持NUMBER到OID的隐式转换。 Oracle: 参数context、part1_type和 object_number均为NUMBER。
31	NAME_TOKENIZE Procedure	NAME_TOKENIZE Procedure	GaussDB: 参数nextpos为INTEGER类型。 Oracle: 参数nextpos为BINARY_INTEGER类型。

序号	Oracle数据库	GaussDB数据库	差异
32	OLD_CURRENT_SCHEMA Function	OLD_CURRENT_SC HEMA Function	GaussDB：函数返回类型为VARCHAR。 Oracle：函数返回类型为VARCHAR2。
33	OLD_CURRENT_USER Function	OLD_CURRENT_U SER Function	GaussDB：函数返回类型为TEXT。 Oracle：函数返回类型为VARCHAR2。
34	PORT_STRING Function	不支持	-
35	SQLID_TO_SQLHASH Function	不支持	-
36	TABLE_TO_COMMA Procedures	TABLE_TO_COMM A Procedures	GaussDB：参数tab为VARCHAR2数组。 Oracle：该存储过程存在2个重载。参数tab可以为两种类型之一：一种为uncl_array，另一种为lname_array。
37	VALIDATE Procedure	不支持	-
38	WAIT_ON_PENDING_DM L Function	不支持	-

表 2-113 DBMS_SQL 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	BIND_ARRAY Procedures	SQL_BIND_ARRAY Function	-
2	BIND_VARIABLE Procedures	SQL_BIND_VARIABL E Function	-
3	BIND_VARIABLE_PKG Procedure	不支持	-
4	CLOSE_CURSOR Procedure	SQL_UNREGISTER_C ONTEXT Function	-
5	COLUMN_VALUE Procedure	GET_RESULT Procedure	-
6	COLUMN_VALUE_LO NG Procedure	不支持	-
7	DEFINE_ARRAY Procedure	SET_RESULTS_TYPE Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
8	DEFINE_COLUMN Procedures	SET_RESULT_TYPE Procedure	-
9	DEFINE_COLUMN_CHAR Procedure	不支持	-
10	DEFINE_COLUMN_LONG Procedure	不支持	-
11	DEFINE_COLUMN_RAW Procedure	不支持	-
12	DEFINE_COLUMN_ROWID Procedure	不支持	-
13	DESCRIBE_COLUMNS Procedure	DESCRIBE_COLUMNS Procedure	-
14	DESCRIBE_COLUMNS2 Procedure	不支持	-
15	DESCRIBE_COLUMNS3 Procedure	不支持	-
16	EXECUTE Function	SQL_RUN Function	GaussDB: 返回值为常量 1。当前对于语句中 unknown 类型之间的比较, 无法正确返回结果。 Oracle: 对于 INSERT、UPDATE、DELETE 语句, 返回值是影响的行数, 对于其他语句则无意义。
17	EXECUTE_AND_FETCH Function	RUN_AND_NEXT Function	-
18	FETCH_ROWS Function	NEXT_ROW Function	-
19	GET_NEXT_RESULT Procedures	不支持	-
20	IS_OPEN Function	IS_ACTIVE Function	-
21	LAST_ERROR_POSITION Function	不支持	-
22	LAST_ROW_COUNT Function	LAST_ROW_COUNT Function	-
23	LAST_ROW_ID Function	不支持	-
24	LAST_SQL_FUNCTION_CODE Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
25	OPEN_CURSOR Functions	REGISTER_CONTEXT Function	-
26	PARSE Procedures	支持，有差异	GaussDB中为SQL_SET_SQL Function，不支持重载。
27	RETURN_RESULT Procedures	不支持	-
28	TO_CURSOR_NUMBER Function	不支持	-
29	TO_REFCURSOR Function	不支持	-
30	VARIABLE_VALUE Procedures	GET_VARIABLE_RESULT Procedures	-
31	VARIABLE_VALUE_PACKAGE Procedure	不支持	-

表 2-114 DBMS_SQL 数据类型兼容性说明

序号	Oracle数据库	GaussDB数据库
1	DBMS_SQL DESC_REC	DBE_SQL.DESC_REC
2	DBMS_SQL DATE_TABLE	DBE_SQL.DATE_TABLE
3	DBMS_SQL NUMBER_TABLE	DBE_SQL.NUMBER_TABLE
4	DBMS_SQL VARCHAR2_TABLE	DBE_SQL.VARCHAR2_TABLE
5	DBMS_SQL BLOB_TABLE	DBE_SQL.BLOB_TABLE

表 2-115 UTL_FILE 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	FCLOSE Procedure	CLOSE Procedure	-
2	FCLOSE_ALL Procedure	CLOSE_ALL Procedure	-
3	FCOPY Procedure	COPY Procedure	-
4	FFLUSH Procedure	FLUSH Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
5	FGETATTR Procedure	GET_ATTR Procedure	-
6	FGETPOS Function	GET_POS Function	-
7	FOPEN Function	FOPEN Function	-
8	FOPEN_NCHAR Function	FOPEN_NCHAR Function	-
9	FREMOVE Procedure	REMOVE Procedure	-
10	FRENAME Procedure	RENAME Procedure	-
11	FSEEK Procedure	SEEK Procedure	-
12	GET_LINE Procedure	READ_LINE Procedure	-
13	GET_LINE_NCHAR Procedure	READ_LINE_NCHAR Procedure	-
14	GET_RAW Procedure	GET_RAW Procedure	-
15	IS_OPEN Function	IS_OPEN Function	-
16	NEW_LINE Procedure	支持，有差异，NEW_LINE Function	GaussDB将接口定义为Function。
17	PUT Procedure	支持，有差异，WRITE Function	GaussDB将接口定义为Function。
18	PUT_LINE Procedure	支持，有差异，WRITE_LINE Function	GaussDB将接口定义为Function。
19	PUT_LINE_NCHAR Procedure	支持，有差异，WRITE_LINE_NCHAR Function	GaussDB将接口定义为Function。
20	PUT_NCHAR Procedure	支持，有差异，WRITE_NCHAR Function	GaussDB将接口定义为Function。
21	PUTF Procedure	支持，有差异，FORMAT_WRITE Function	GaussDB将接口定义为Function。
22	PUTF_NCHAR Procedure	支持，有差异，FORMAT_WRITE_NCHAR Function	GaussDB将接口定义为Function。

序号	Oracle数据库	GaussDB数据库	差异
23	PUT_RAW Procedure	支持，有差异，PUT_RAW Function	GaussDB将接口定义为Function。

表 2-116 DBMS_SESSION 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	CLEAR_ALL_CONTEXT Procedure	不支持	-
2	CLEAR_CONTEXT Procedure	CLEAR_CONTEXT Function	-
3	CLEAR_IDENTIFIER Procedure	不支持	-
4	CLOSE_DATABASE_LINK Procedure	不支持	-
5	CURRENT_IS_ROLE_ENABLED Function	不支持	-
6	FREE_UNUSED_USER_MEMORY Procedure	不支持	-
7	GET_PACKAGE_MEMORY_UTILIZATION Procedure	不支持	-
8	IS_ROLE_ENABLED Function	不支持	-
9	IS_SESSION_ALIVE Function	不支持	-
10	LIST_CONTEXT Procedures	不支持	-
11	MODIFY_PACKAGE_STATE Procedure	MODIFY_PACKAGE_STATE Procedure	GaussDB：仅支持入参 flags = 1的场景使用。 Oracle：支持flags=1或 flags = 2的场景使用。
12	RESET_PACKAGE Procedure	不支持	-
13	SESSION_IS_ROLE_ENABLED Function	不支持	-
14	SESSION_TRACE_DISABLE Procedure	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
15	SESSION_TRACE_ENABLE Procedure	不支持	-
16	SET_CONTEXT Procedure	SET_CONTEXT Function	GaussDB：仅包括参数 namespace, attribute和 value, 类型均为text。 Oracle：包括参数 namespace, attribute, value, username和 client_id, 类型均为 VARCHAR2。
17	SET_EDITION_DEFERRED Procedure	不支持	-
18	SET_IDENTIFIER Procedure	不支持	-
19	SET-NLS Procedure	不支持	-
20	SET_ROLE Procedure	不支持	-
21	SET_SQL_TRACE Procedure	不支持	-
22	SLEEP Procedure	不支持	-
23	SWITCH_CURRENT_CONSUMER_GROUP Procedure	不支持	-
24	UNIQUE_SESSION_ID Function	不支持	-

表 2-117 UTL_MATCH 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	EDIT_DISTANCE Function	不支持	-
2	EDIT_DISTANCE_SIMILARITY Function	EDIT_DISTANCE_SIMILARITY Function	GaussDB：参数str1和str2均为 TEXT类型，函数返回类型为 INTEGER。 Oracle：参数s1和s2为 VARCHAR2类型，函数返回类型为PLS_INTEGER。
3	JARO_WINKLER Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
4	JARO_WINKLER_SIMILARITY Function	不支持	-

表 2-118 DBMS_APPLICATION_INFO 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	READ_CLIENT_INFO Function	READ_CLIENT_INFO Procedure	GaussDB: 参数client_info为TEXT类型。 Oracle: 参数client_info为VARCHAR2类型。
2	READ_MODULE Procedure	READ_MODULE Procedure	GaussDB: 参数module_name、action_name为TEXT类型。 Oracle: 参数module_name、action_name为VARCHAR2类型。
3	SET_ACTION Procedure	SET_ACTION Procedure	GaussDB: 参数action_name为TEXT类型。 Oracle: 参数action_name为VARCHAR2类型。
4	SET_CLIENT_INFO Function	SET_CLIENT_INFO Procedure	GaussDB: 参数str为TEXT类型，且返回类型为void。 Oracle: 参数client_info为VARCHAR2类型，无返回值。二者均为写入客户端信息，最大输入64字节，超过64字节将被截断。
5	SET_MODULE Procedure	SET_MODULE Procedure	GaussDB: 参数module_name、action_name为TEXT类型。 Oracle: 参数module_name、action_name为VARCHAR2类型。
6	SET_SESSION_LONGOPS Procedure	不支持	-

表 2-119 DBMS_XMLDOM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLDOM. APPENDCHILD	DBE_XMLDOM.AP PENDCHILD	- GaussDB: DOCUMENT类型节点下APPEND ATTR类型节点会报“operation not support”错误。 Oracle: 在此场景下不报错, 但实际并没有挂载成功。 - GaussDB: ATTR类型节点下APPEND ATTR类型节点会报“operation not support”错误。 Oracle: 在此场景下不报错, 但实际并没有挂载成功。 - GaussDB: 父节点在添加多个ATTR类型子节点时, 不允许KEY值相同的子节点同时存在于同一个父节点下。 Oracle: 允许KEY值相同的子节点同时存在于同一个父节点下。
2	DBMS_XMLDOM. CREATEELEMENT	DBE_XMLDOM.CR EATELEMENT	-
3	DBMS_XMLDOM. CREATETEXTNO DE	DBE_XMLDOM.CR EATETEXTNODE	-
4	DBMS_XMLDOM. FREEDOCUMENT	支持, 有差异, DBE_XMLDOM.FR EEDOCUMENT	GaussDB: 释放时不会立刻释放对象, 累积一定数量后释放。 document下全部节点失效。 Oracle: 立即释放对象。
5	DBMS_XMLDOM. FREEELEMENT	DBE_XMLDOM.FR EEELEMENT	-
6	DBMS_XMLDOM. FREENODE	DBE_XMLDOM.FR EENODE	-
7	DBMS_XMLDOM. FREENODELIST	支持, 有差异, DBE_XMLDOM.FR EENODELIST	GaussDB: nodelist会被释放。 Oracle: 释放nodelist后, 在原始的doc中还能被查询到。
8	DBMS_XMLDOM. GETATTRIBUTE	DBE_XMLDOM.GE TATTRIBUTE	-
9	DBMS_XMLDOM. GETATTRIBUTES	DBE_XMLDOM.GE TATTRIBUTES	-

序号	Oracle数据库	GaussDB数据库	差异
10	DBMS_XMLDOM. GETCHILDNODE S	DBE_XMLDOM.GE TCHILDNODES	GaussDB：对document的node使用 时会包含dtd Oracle：不包含dtd。
11	DBMS_XMLDOM. GETCHILDRENBY TAGNAME	DBE_XMLDOM.GE TCHILDRENBYTAG NAME	GaussDB： DBE_XMLDOM.GETCHILDRENBY TAGNAME接口的参数ns不支持传 入参数" * "，如需获取节点下全部 属性，可使用 DBE_XMLDOM.GETCHILDNODE S接口。 Oracle：支持传入参数" * "。
12	DBMS_XMLDOM. GETDOCUMENT ELEMENT	DBE.XMLDOM.GE TDOCUMENTELE MENT	-
13	DBMS_XMLDOM. GETFIRSTCHILD	DBE_XMLDOM.GE TFIRSTCHILD	-
14	DBMS_XMLDOM. GETLASTCHILD	DBE_XMLDOM.GE TLASTCHILD	-
15	DBMS_XMLDOM. GETLENGTH	DBE_XMLDOM.GE TLENGTH	-
16	DBMS_XMLDOM. GETLOCALNAME	DBE_XMLDOM.GE TLOCALNAME	-
17	DBMS_XMLDOM. GETNAMEDITEM	DBE_XMLDOM.GE TNAMEDITEM	-
18	DBMS_XMLDOM. GETNEXTSIBLING	DBE_XMLDOM.GE TNEXTSIBLING	-
19	DBMS_XMLDOM. GETNODENAME	DBE_XMLDOM.GE TNODENAME	-
20	DBMS_XMLDOM. GETNODETYPE	DBE_XMLDOM.GE TNODETYPE	-
21	DBMS_XMLDOM. GETTAGNAME	DBE_XMLDOM.GE TTAGNAME	-
22	DBMS_XMLDOM. IMPORTNODE	DBE_XMLDOM.IM PORTNODE	-
23	DBMS_XMLDOM. ISNULL	DBE_XMLDOM.ISN ULL	GaussDB：入参为 DOMNODELIST类型时，若对象 在哈希表中不存在会发生报错。 Oracle：不会报错。
24	DBMS_XMLDOM. ITEM	DBE_XMLDOM.ITE M	-

序号	Oracle数据库	GaussDB数据库	差异
25	DBMS_XMLDOM. MAKENODE	DBE_XMLDOM.MA KENODE	GaussDB: 该函数不支持直接作为函数返回值返回。 Oracle: 支持直接作为函数返回值返回。
26	DBMS_XMLDOM. NEWDOMDOC UMENT	DBE_XMLDOM.NE WDOMDOCUMEN T	- GaussDB入参大小需限制1GB以内。 Oracle: 与CLOB类型大小一致。 - GaussDB 目前暂不支持外部DTD解析。 Oracle: 支持解析外部DTD。 - GaussDB newdomdocument创建的doc, 默认UTF-8字符集。 Oracle: 根据服务端字符集生成。 - GaussDB 从同一个xmltype实例中解析出的每一个doc都是独立的, 对doc的修改也不会影响到xmltype。 Oracle: 从同一个xmltype实例中解析出的每一个doc不独立, 有关联关系。 - GaussDB version字段只支持1.0, 1.0-1.9解析警告但正常执行, 1.9以上报错。 Oracle: 不报错。 - GaussDB 与Oracle数据库DTD校验差异: !ATTLIST to type (CHECK check Check) "Ch..."将报错, 因默认值"Ch..."不属于括号中枚举值, 而Oracle数据库不报错。<!ENTITY baidu "www.baidu.com">.....&Baidu;&writer将报错, 因区分字母大小写, Baidu无法与baidu对应。 Oracle: 不报错。 - GaussDB 与Oracle数据库命名空间校验差异: 解析未声明的命名空间标签正常执行。 Oracle: 报错。
27	DBMS_XMLDOM. SETATTRIBUTE	DBE_XMLDOM.SE TATTRIBUTE	GaussDB: 属性key不支持为null或空字符串。 Oracle: 属性key允许为null或空字符串。

序号	Oracle数据库	GaussDB数据库	差异
28	DBMS_XMLDOM. SETCHARSET	DBE_XMLDOM.SE TCHARSET	GaussDB目前支持的字符集有： UTF-8、UCS-4、UCS-2、 ISO-8859-1、ISO-8859-2、 ISO-8859-3、ISO-8859-4、 ISO-8859-5、ISO-8859-6、 ISO-8859-7、ISO-8859-8、 ISO-8859-9、ISO-2022-JP、 Shift_JIS、EUC-JP、ASCII。输入 其他字符集会报错或者可能导致输 出乱码。
29	DBMS_XMLDOM. SETDOCTYPE	DBE_XMLDOM.SE TDOCTYPE	GaussDB name、sysid、pubid的 总长度限制在32500个字节以内。 Oracle：限制在32767字节内。
30	DBMS_XMLDOM. WRITETOBUFFER	支持，有差异， DBE_XMLDOM.WR ITETOBUFFER	- GaussDB：writetobuffer输出 buffer限制在1GB以内。 Oracle：限制在32767字节 内。 - GaussDB：输出doc将包含XML 声明version和encoding。 Oracle：用户不主动指定将不 包含。 - GaussDB：入参为domnode类 型时，如果节点是doc转换的， 输出节点将包含XML声明 version和encoding。 Oracle：用户不主动指定将不 包含。 - GaussDB：默认以UTF-8字符 集输出xml。 Oracle：根据数据库字符集生 成。

序号	Oracle数据库	GaussDB数据库	差异
31	DBMS_XMLDOM. WRITETOCLOB	DBE_XMLDOM.WR ITETOCLOB	• GaussDB: writetoclob大小支持1GB以内。 Oracle: 按CLOB大小支持。 • GaussDB: 输出doc将包含XML声明version和encoding。 Oracle: 用户不主动指定将不包含。 • GaussDB: 入参为domnode类型时, 如果节点是doc转换的, 输出节点将包含XML声明version和encoding。 Oracle: 用户不主动指定将不包含。 • GaussDB 默认以UTF-8字符集输出xml。 Oracle: 根据数据库字符集生成。

序号	Oracle数据库	GaussDB数据库	差异
32	DBMS_XMLDOM. WRITETOFILE	DBE_XMLDOM.WR ITETOFILE	- GaussDB: document入参, filename长度限制在255个字节以内, charset请参考 dbe_xmldom.setcharset接口。 - Oracle: filename长度限制受操作系统影响, 大于255个字节。 - GaussDB: domnode入参, filename长度限制在255个字节以内, charset请参考 dbe_xmldom.setcharset接口。 - Oracle: filename长度限制受操作系统影响, 大于255个字节。 - GaussDB: 该函数会添加缩进等内容, 将输出格式化。输出 doc将包含XML声明version和 encoding。入参为domnode类型时, 如果节点是doc转换的, 输出节点将包含XML声明 version和encoding。 - Oracle: 用户不主动指定将不包含。 - GaussDB: 传入 newdomdocument()无参创建的doc, 在不指定charset时不会报错, 默认UTF-8字符集。 - Oracle: 会进行报错。 - GaussDB: filename需要在 pg_directory中创建的路径下, filename中的\会被转换成/, 只允许存在一个/。文件名格式应为pg_directory_name/file_name。 - Oracle: 按用户输入不进行转义。
33	DBMS_XMLDOM. GETNODEVALUE	DBE_XMLDOM.GE TNODEVALUE	-
34	DBMS_XMLDOM. GETPARENTNOD E	DBE_XMLDOM.GE TPARENTNODE	-
35	DBMS_XMLDOM. HASCHILDNODE S	DBE_XMLDOM.HA SCHILDNODES	-

序号	Oracle数据库	GaussDB数据库	差异
36	DBMS_XMLDOM. MAKEELEMENT	DBE_XMLDOM.MA KEELEMENT	-
37	DBMS_XMLDOM. SETNODEVALUE	DBE_XMLDOM.SE TNODEVALUE	- GaussDB: nodeValue入参，可以输入空字符串和NULL值，但不会对节点值进行修改。 - Oracle: 空字符串和NULL会将节点值修改为空字符串。 - GaussDB: nodeValue入参，暂不支持转义字符'&'，如字符串中包含该转义字符，会清空节点值。 - Oracle: 支持转义字符。
38	DBMS_XMLDOM. GETELEMENTSBY TAGNAM	DBE_XMLDOM.GE TELEMENTSBYTAG NAME	-

表 2-120 DBMS_XMLPARSER 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLPARS ER.FREEPARSER	DBE_XMLPARSER.F REEPARSER	-
2	DBMS_XMLPARS ER.GETDOCUME NT	DBE_XMLPARSER.G ETDOCUMENT	-
3	DBMS_XMLPARS ER.GETVALIDATI ONMODE	DBE_XMLPARSER.G ETVALIDATIONMO DE	-
4	DBMS_XMLPARS ER.NEWPARSER	支持，有差异， DBE_XMLPARSER. NEWPARSER	GaussDB中parser对象的数量上限为16777215，Oracle中约为1亿。

序号	Oracle数据库	GaussDB数据库	差异
5	DBMS_XMLPARSER.PARSEBUFFER	支持，有差异，DBE_XMLPARSER.PARSEBUFFER	- 与Oracle数据库解析字段差异：字符串encoding只支持UTF-8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。 - 与Oracle数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而Oracle数据库会报错。 - 与Oracle数据库xml预定义实体解析差异：&apos;&quot;会被解析转义为字符'”，而Oracle中预定义实体统一都没有转义为字符。 - 与Oracle数据库DTD校验差异： - !ATTLIST to type (CHECK check Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而Oracle数据库不报错。 - <!ENTITY baidu "www.baidu.com">.....&Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而Oracle数据库不报错。

序号	Oracle数据库	GaussDB数据库	差异
6	DBMS_XMLPARSER.PARSECLOB	支持，有差异，DBE_XMLPARSER.PARSECLOB	• PARSECLOB不支持解析大于等于2GB的clob。 • 与Oracle数据库解析字段差异：字符串encoding只支持UTF-8；version字段只支持1.0，1.0-1.9解析警告但正常执行，1.9以上报错。 • 与Oracle数据库命名空间校验差异：解析未声明的命名空间标签正常执行，而Oracle数据库会报错。 • 与Oracle数据库xml预定义实体解析差异：&quot;会被解析转义为字符'”，而Oracle中预定义实体统一都没有转义为字符。 • 与Oracle数据库DTD校验差异： - !ATTLIST to type (CHECK check Check) "Ch..."将报错，因默认值"Ch..."不属于括号中枚举值，而Oracle数据库不报错。 - <!ENTITY baidu "www.baidu.com">.....&Baidu;&writer将报错，因区分字母大小写，Baidu无法与baidu对应，而Oracle数据库不报错。
7	DBMS_XMLPARSER.SETVALIDATIONMODE	DBE_XMLPARSER.SETVALIDATIONMODE	-

表 2-121 DBMS_ILM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_ILM.ADD_TO_ILM	不支持	-
2	DBMS_ILM.ARCHIVESTATENAME	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
3	DBMS_ILM.EXECUTE_ILM	DBE_ILM.EXECUTE_ILM	- GaussDB数据库的入参Schema名在Oracle数据库中对应为owner。 - GaussDB数据库不支持指定ilm_scope（一次指定多个对象）的操作。
4	DBMS_ILM.EXECUTE_ILM_TASK	不支持	-
5	DBMS_ILM.PREVIEWS_ILM	不支持	-
6	DBMS_ILM.REMOVE_FROM_ILM	不支持	-
7	DBMS_ILM.STOP_ILM	DBE_ILM.STOP_ILM	-

表 2-122 DBMS_ILM_ADMIN 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_ALL	不支持	-
2	DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_TABLE	不支持	-
3	DBMS_ILM_ADMIN.CUSTOMIZE_ILM	DBE_ILM_ADMIN.CUSTOMIZE_ILM	入参parameter取值对应的特性参数存在差异。 - GaussDB数据库param取值支持1、2、7、11、12、13、14和15。 - GaussDB数据库param取值为14时，对应的特性参数为WIND_DURATION，用于控制自动调度中执行窗口的持续时长，而ORACLE数据库无对应的特性参数。
4	DBMS_ILM_ADMIN.DISABLE_ILM	DBE_ILM_ADMIN.DISABLE_ILM	-
5	DBMS_ILM_ADMIN.ENABLE_AUTO_OPTIMIZE	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
6	DBMS_ILM_ADMIN.ENABLE_ILM	DBE_ILM_ADMIN.ENABLE_ILM	-
7	DBMS_ILM_ADMIN.IGNORE_AUTO_OPTIMIZE_CRITERIA	不支持	-
8	DBMS_ILM_ADMIN.SET_HEAT_MAP_ALL	不支持	-
9	DBMS_ILM_ADMIN.SET_HEAT_MAP_START	不支持	-
10	DBMS_ILM_ADMIN.SET_HEAT_MAP_TABLE	不支持	-

表 2-123 DBMS_COMPRESSION 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_COMPRESSION.GET_COMPRESSION_RATIO	DBE_COMPRESSION.GET_COMPRESSION_RATIO	● GaussDB不支持LOBs的压缩率获取。 ● 对于单个对象的压缩率获取： - GaussDB入参comptype取值仅支持1（未压缩）和2（高级压缩），Oracle还支持1024、2048等取值。 - GaussDB入参objtype取值仅支持1（表对象），而Oracle还支持2（索引对象）。 - Oracle数据库使用subset_numrows参数直接来决定采样的行数（即为参数的取值），而GaussDB则使用sample_ratio（采样率）来间接确定采样的行数。

序号	Oracle数据库	GaussDB数据库	差异
2	DBMS_COMPRESSION.GET_COMPRESSION_TYPE	DBE_COMPRESSION.GET_COMPRESSION_TYPE	- Oracle使用rowid来指定待获取压缩类型的行，而GaussDB则是使用行的ctid来指定。 - 返回值为comptype，其取值差异同GET_COMPRESSION_RATIO。 - GaussDB该接口仅支持在DN调用，详见GaussDB《开发指南》中“存储过程>高级包>二次封装接口（推荐）>DBE_COMPRESSION”章节。

表 2-124 DBMS_HEAT_MAP 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_HEAT_MAP.BLOCK_HEAT_MAP	不支持	-
2	DBMS_HEAT_MAP.EXTENT_HEAT_MAP	不支持	-
3	DBMS_HEAT_MAP.OBJECT_HEAT_MAP	不支持	-
4	DBMS_HEAT_MAP.SEGMENT_HEAT_MAP	不支持	-
5	DBMS_HEAT_MAP.TABLESPACE_HEAT_MAP	不支持	-
6	不支持	DBE_HEAT_MAP.ROW_HEAT_MAP	GaussDB该接口仅支持在DN调用，详见GaussDB《开发指南》中“存储过程>高级包>二次封装接口（推荐）>DBE_HEAT_MAP”章节。

表 2-125 DBMS_DESCRIBE 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_DESCRIBE.DESCRIBE_PROCEDURE	DBE_DESCRIBE.DESCRIBE_PROCEDURE	- datatype参数与Oracle存在差异，GaussDB返回数据类型的oid，Oracle数据库返回Oracle数据库内部的数据类型的编号。 - datalength、dataprecision和scale因GaussDB创建存储过程或函数时无法保留类型的约束（如number(7,2)、varchar2(20)等），该三个参数置0处理；Oracle可使用%type方法获得带约束的数据类型。 - GaussDB的具体使用请参见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_DESCRIBE”章节。

表 2-126 DBMS_STATS 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_STATS.ALTER_STATS_HISTORY_RETENTION	不支持	-
2	DBMS_STATS.CANCEL_ADVISOR_TASK	不支持	-
3	DBMS_STATS.CONFIGURE_ADVISOR_FILTER	不支持	-
4	DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER	不支持	-
5	DBMS_STATS.CONFIGURE_ADVISOR_OPR_FILTER	不支持	-
6	DBMS_STATS.CONFIGURE_ADVISOR_RULE_FILTER	不支持	-
7	DBMS_STATS.CREATE_ADVISOR_TASK	不支持	-
8	DBMS_STATS.CONVERT_RAW_VALUE	不支持	-
9	DBMS_STATS.CONVERT_RAW_VALUE_NVARCHAR	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
10	DBMS_STATS.CONVERT_ROW_VALUE_ROWID	不支持	-
11	DBMS_STATS.COPY_TABLE_STATS	不支持	-
12	DBMS_STATS.CREATE_EXTENDED_STATS	不支持	-
13	DBMS_STATS.CREATE_STAT_TABLE	DBE_STATS.CREATE_STAT_TABLE	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
14	DBMS_STATS.DELETE_COLUMN_STATS	DBE_STATS.DELETE_COLUMN_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - GaussDB中，使用该接口可以设置表达式统计信息，但tablename应传该表达式对应的索引名。
15	DBMS_STATS.DELETE_DATABASE_PREFS	不支持	-
16	DBMS_STATS.DELETE_DATABASE_STATS	不支持	-
17	DEDBMS_STATS.DELETE_DICTIONARY_STATS	不支持	-
18	DBMS_STATS.DELETE_FIXED_OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
19	DBMS_STATS.DELETE_INDEX_STATS	DBE_STATS.DELETE_INDEX_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
20	DBMS_STATS.DELETE_PENDING_STATS	不支持	-
21	DBMS_STATS.DELETE_PROCESSING_RATE	不支持	-
22	DBMS_STATS.DELETE_SCHEMA_PREFS	不支持	-
23	DBMS_STATS.DELETE_SCHEMA_STATS	DBE_STATS.DELETE_SCHEMA_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
24	DBMS_STATS.DELETE_SYSTEM_STATS	不支持	-
25	DBMS_STATS.DELETE_TABLE_PREFS	不支持	-
26	DBMS_STATS.DELETE_TABLE_STATS	DBE_STATS.DELETE_TABLE_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
27	DBMS_STATS.DIFF_TABLE_STATS_IN_HISTORY	不支持	-
28	DBMS_STATS.DIFF_TABLE_STATS_IN_PENDING	不支持	-
29	DBMS_STATS.DIFF_TABLE_STATS_IN_STATTAB	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
30	DBMS_STATS.DROP_ADVISOR_TASK	不支持	-
31	DBMS_STATS.DROP_EXTENDED_STATS	不支持	-
32	DBMS_STATS.DROP_STAT_TABLE	DBE_STATS.DROP_STAT_TABLE	-
33	DBMS_STATS.EXECUTE_ADVISOR_TASK	不支持	-
34	DBMS_STATS.EXPORT_COLUMN_STATS	DBE_STATS.EXPORT_COLUMN_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 • 导出的列级统计信息与pg_statistic表保持一致，多列和pg_statistic_ext表保持一致。 • 支持导出索引表达式统计信息。要求tablename传的是索引名称，colname传的是索引表达式名称。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
35	DBMS_STATS.EXPORT_DATABASE_PREFS	不支持	-
36	DBMS_STATS.EXPORT_DATABASE_STATS	不支持	-
37	DBMS_STATS.EXPORT_DICTIONARY_STATS	不支持	-
38	DBMS_STATS.EXPORT_FIXED_OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
39	DBMS_STATS.EXPORT_INDEX_STATS	DBE_STATS.EXPORT_INDEX_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。 - 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
40	DBMS_STATS.EXPORT_PENDING_STATS	不支持	-
41	DBMS_STATS.EXPORT_SCHEMA_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
42	DBMS_STATS.EXPORT_SCHEMA_STATS	DBE_STATS.EXPORT_SCHEMA_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples, relpages, relallvisible。导出表相关列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 - 权限：需要具有stattab表的siud权限。
43	DBMS_STATS.EXPORT_SYSTEM_STATS	不支持	-
44	DBMS_STATS.EXPORT_TABLE_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
45	DBMS_STATS.EXPORT_TABLE_STATS	DBE_STATS.EXPORT_TABLE_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - stattab表中，导出的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples, relpages, relallvisible。级联导出的列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 - 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
46	DBMS_STATS.FLUSH_DATABASE_MONITORING_INFO	不支持	-
47	DBMS_STATS.GATHER_DATABASE_STATS	不支持	-
48	DBMS_STATS.GATHER_DICTIONARY_STATS	不支持	-
49	DBMS_STATS.GATHER_FIXED_OBJECTS_STATS	不支持	-
50	DBMS_STATS.GATHER_INDEX_STATS	不支持	-
51	DBMS_STATS.GATHER_PROCESSING_RATE	不支持	-
52	DBMS_STATS.GATHER_SCHEMA_STATS	不支持	-
53	DBMS_STATS.GATHER_SYSTEM_STATS	不支持	-
54	DBMS_STATS.GATHER_TABLE_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
55	DBMS_STATS.GENERATE_STATS	不支持	-
56	DBMS_STATS.GET_ADVISOR_OPR_FILTER	不支持	-
57	DBMS_STATS.GET_ADVISOR_RECS	不支持	-
58	DBMS_STATS.GET_COLUMN_STATS	不支持	-
59	DBMS_STATS.GET_INDEX_STATS	不支持	-
60	DBMS_STATS.GET_PARAM	不支持	-
61	DBMS_STATS.GET_PREFS	不支持	-
62	DBMS_STATS.GET_STATS_HISTORY_AVAILABILITY	DBE_STATS.GET_STATS_HISTORY_AVAILABILITY	GaussDB查询到的是全库存在的最早历史统计信息的收集时间。
63	DBMS_STATS.GET_STATS_HISTORY_RETENTION	DBE_STATS.GET_STATS_HISTORY_RETENTION	-
64	DBMS_STATS.GET_SYSTEM_STATS	不支持	-
65	DBMS_STATS.GET_TABLE_STATS	不支持	-
66	DBMS_STATS.IMPLEMENT_ADVISOR_TASK	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
67	DBMS_STATS.IMPORT_COLUMN_STATS	DBE_STATS.IMPORT_COLUMN_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - 导出单列col导出的统计信息与pg_statistic表保持一致。多列ext-col导出的统计信息与pg_statistic_ext表保持一致。 - 支持导入索引表达式统计信息。要求tablename传的是索引名称，colname传的是索引表达式名称。 - 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
68	DBMS_STATS.IMPORT_DATABASE_PREFS	不支持	-
69	DBMS_STATS.IMPORT_DATABASE_STATS	不支持	-
70	DBMS_STATS.IMPORT_DICTIONARY_STATS	不支持	-
71	DBMS_STATS.IMPORT_FIXED_OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
72	DBMS_STATS.IMPORT_INDEX_STATS	DBE_STATS.IMPORT_INDEX_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_STATS”章节。 - stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples, relpages, relallvisible。 - 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
73	DBMS_STATS.IMPORT_SCHEMA_PREFS	不支持	-
74	DBMS_STATS.IMPORT_SCHEMA_STATS	DBE_STATS.IMPORT_SCHEMA_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐）> DBE_STATS”章节。 - stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples, relpages, relallvisible。导入表相关列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 - 权限：需要具有stattab表的siud权限。

序号	Oracle数据库	GaussDB数据库	差异
75	DBMS_STATS.IMPORT_SYST EM_STATS	不支持	-
76	DBMS_STATS.IMPORT_TABL E_PREFS	不支持	-
77	DBMS_STATS.IMPORT_TABL E_STATS	DBE_STATS.I MPORT_TABL E_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 • stattab表中，导入的表、分区级统计信息为numrows、numblocks、relallvisible，分别对应系统表pg_class、pg_partition的reltuples, relpages, relallvisible。级联导入的列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 • 权限：需要具有查询表的ANALYZE权限以及stattab表的siud权限。
78	DBMS_STATS.INTERRUPT_A DVISOR_TASK	不支持	-
79	DBMS_STATS.LOCK_PARTITI ON_STATS	DBE_STATS.L OCK_PARTITI ON_STATS	• GaussDB中ownname应传Schema名。
80	DBMS_STATS.LOCK_SCHEM A_STATS	DBE_STATS.L OCK_SCHEM A_STATS	• GaussDB中ownname应传Schema名。
81	DBMS_STATS.LOCK_TABLE_ STATS	DBE_STATS.L OCK_TABLE_S TATS	• GaussDB中ownname应传Schema名。
82	DBMS_STATS.MERGE_COL_ USAGE	不支持	-
83	DBMS_STATS.PREPARE_COL UMN_VALUES	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
84	DBMS_STATS.PREPARE_COLUMN_VALUES_ROWID	不支持	-
85	DBMS_STATS.PUBLISH_PENDING_STATS	不支持	-
86	DBMS_STATS.PURGE_STATS	DBE_STATS.PURGE_STATS	-
87	DBMS_STATS.REMAP_STAT_TABLE	不支持	-
88	DBMS_STATS.REPORT_ADVISOR_TASK	不支持	-
89	DBMS_STATS.REPORT_COLUMN_USAGE	不支持	-
90	DBMS_STATS.REPORT_GATHER_AUTO_STATS	不支持	-
91	DBMS_STATS.REPORT_GATHER_DATABASE_STATS	不支持	-
92	DBMS_STATS.REPORT_GATHER_DICTIONARY_STATS	不支持	-
93	DBMS_STATS.REPORT_GATHER_FIXED_OBJ_STATS	不支持	-
94	DBMS_STATS.REPORT_GATHER_SCHEMA_STATS	不支持	-
95	DBMS_STATS.REPORT_STATISTICS_OPERATIONS	不支持	-
96	DBMS_STATS.RESET_ADVISOR_TASK	不支持	-
97	DBMS_STATS.RESET_COLUMN_USAGE	不支持	-
98	DBMS_STATS.RESET_GLOBAL_PREF_DEFAULTS	不支持	-
99	DBMS_STATS.RESET_PARAMETER_DEFAULTS	不支持	-
100	DBMS_STATS.RESTORE_DICTIONARY_STATS	不支持	-
101	DBMS_STATS.RESTORE_FIXED_OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
102	DBMS_STATS.RESTORE_SCHEMA_STATS	DBE_STATS.RESTORE_SCHEMA_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
103	DBMS_STATS.RESTORE_SYSTEM_STATS	不支持	-
104	DBMS_STATS.RESTORE_TABLE_STATS	DBE_STATS.RESTORE_TABLE_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
105	DBMS_STATS.RESUME_ADVISOR_TASK	不支持	-
106	DBMS_STATS.SCRIPT_ADVISOR_TASK	不支持	-
107	DBMS_STATS.SEED_COLUMN_USAGE	不支持	-
108	DBMS_STATS.SET_ADVISOR_TASK_PARAMETER	不支持	-
109	DBMS_STATS.SET_COLUMN_STATS	DBE_STATS.SET_COLUMN_STATS	• GaussDB中ownname应传Schema名。 • GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。
110	DBMS_STATS.SET_DATABASE_PREFS	不支持	-
111	DBMS_STATS.SET_GLOBAL_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
112	DBMS_STATS.SET_INDEX_STATS	DBE_STATS.SET_INDEX_STATS	- GaussDB中ownname应传Schema名。GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - GaussDB中新增了relallvisible入参。
113	DBMS_STATS.SET_PARAM	不支持	-
114	DBMS_STATS.SET_PROCESSING_RATE	不支持	-
115	DBMS_STATS.SET_SCHEMA_PREFS	不支持	-
116	DBMS_STATS.SET_SYSTEM_STATS	不支持	-
117	DBMS_STATS.SET_TABLE_PREFS	不支持	-
118	DBMS_STATS.SET_TABLE_STATS	DBE_STATS.SET_TABLE_STATS	- GaussDB中ownname应传Schema名。 - GaussDB仅支持部分入参功能，详见《开发指南》中“存储过程 > 高级包 > 二次封装接口（推荐） > DBE_STATS”章节。 - GaussDB中新增了relallvisible入参。
119	DBMS_STATS.SHOW_EXTENDED_STATS_NAME	不支持	-
120	DBMS_STATS.TRANSFER_STATS	不支持	-
121	DBMS_STATS.UNLOCK_PARTITION_STATS	DBE_STATS.UNLOCK_PARTITION_STATS	GaussDB中ownname应传Schema名。
122	DBMS_STATS.UNLOCK_SCHEMA_STATS	DBE_STATS.UNLOCK_SCHEMA_STATS	GaussDB中ownname应传Schema名。

序号	Oracle数据库	GaussDB数据库	差异
123	DBMS_STATS.UNLOCK_TABLE_STATS	DBE_STATS.UNLOCK_TABLE_STATS	GaussDB中ownname应传Schema名。
124	DBMS_STATS.UPGRADE_STAT_TABLE	不支持	-

表 2-127 DBMS_XMLGEN 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLGEN.CONVERT	DBE_XMLGEN.CONVERT	-
2	DBMS_XMLGEN.NEWCONTEXT	DBE_XMLGEN.NEWCONTEXT	-
3	DBMS_XMLGEN.NEWCONTEXTFROMHIERARCHY	DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY	● GaussDB生成的递归XML最大深度不能超过5000万层。 ● Oracle的newcontextfromhierarchy方法对于connect by语句生成的xml是带xml头的，但是对于直接构造的数据不带xml头，GaussDB均带xml头。
4	DBMS_XMLGEN.SETCONVERTSPECIALCHARS	DBE_XMLGEN.SETCONVERTSPECIALCHARS	-
5	DBMS_XMLGEN.SETNULLHANDLING	DBE_XMLGEN.SETNULLHANDLING	-
6	DBMS_XMLGEN.SETROWSETTAG	DBE_XMLGEN.SETROWSETTAG	-
7	DBMS_XMLGEN.SETROWTAG	DBE_XMLGEN.SETROWTAG	-
8	DBMS_XMLGEN.USENULLATTRIBUTEINDICATOR	DBE_XMLGEN.USENULLATTRIBUTEINDICATOR	-
9	DBMS_XMLGEN.USEITEMTAGSFORCOLL	DBE_XMLGEN.USEITEMTAGSFORCOLL	-

序号	Oracle数据库	GaussDB数据库	差异
10	DBMS_XMLGEN.GETNUMROWSPROCESSED	DBE_XMLGEN.GETNUMROWSPROCESSED	-
11	DBMS_XMLGEN.SETMAXROWS	DBE_XMLGEN.SETMAXROWS	-
12	DBMS_XMLGEN.SETSKIPROWS	DBE_XMLGEN.SETSKIPROWS	-
13	DBMS_XMLGEN.RESTARTQUERY	DBE_XMLGEN.RESTARTQUERY	GaussDB分布式中不允许cursor反向移动，因此restartquery函数不可使用。
14	DBMS_XMLGEN.GETXMLTYPE	DBE_XMLGEN.GETXMLTYPE	-
15	DBMS_XMLGEN.GETXML	DBE_XMLGEN.GETXML	-
16	DBMS_XMLGEN.CLOSECONTEXT	DBE_XMLGEN.CLOSECONTEXT	-

3 MySQL 兼容性说明

3.1 MySQL 数据库兼容性概述

MySQL 兼容性 M-Compatibility 模式概述

MySQL兼容性M-Compatibility模式主要介绍GaussDB数据库的MySQL兼容性M-Compatibility模式（即`sql_compatibility='M'`）与MySQL 5.7数据库的兼容性对比信息。仅介绍505.2版本后新增的兼容性特性，特性的相关规格和约束建议在《M-Compatibility开发指南》中查看。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

GaussDB的执行计划和优化、EXPLAIN显示结果与MySQL不同。

由于GaussDB数据库与MySQL数据库在底层框架实现上存在差异，GaussDB数据库与MySQL数据库仍存在部分差异。

说明

由于GaussDB的底层架构与MySQL存在差异，对于`information_schema`和`m_schema`下与MySQL名称相同的Schema（具体请参见《M-Compatibility开发指南》中的“Schema”章节），其查询性能可能存在差异。例如`count`函数无法做执行上的优化，表现为`SELECT *`和`SELECT count(*)`语句耗时近似。

MySQL 兼容性 MYSQL 模式概述

MySQL兼容性MYSQL模式主要介绍GaussDB数据库的MySQL兼容性MYSQL模式（即`sql_compatibility = 'MYSQL'`、且设置参数`b_format_version='5.7'`、`b_format_dev_version='s1'`时）与MySQL 5.7数据库的兼容性对比信息。仅介绍503.0.0版本后新增的兼容性特性，特性的相关规格和约束建议在《开发指南》中查看。

说明

- MySQL兼容性MYSQL模式在语法、数据类型、元数据、协议等功能上对MySQL数据库的兼容度较高。MySQL兼容性MYSQL模式由于架构限制无法很好的与MySQL兼容，后续不再演进。
- 仅在**MySQL兼容性MYSQL模式**中体现的特性行为与MySQL进行了兼容性适配，未体现的特性行为仍然与GaussDB保持一致。
- 当前形态下MYSQL模式（即sql_compatibility='MYSQL'）与集中式形态下B模式（即sql_compatibility='B'）实现逻辑相近。

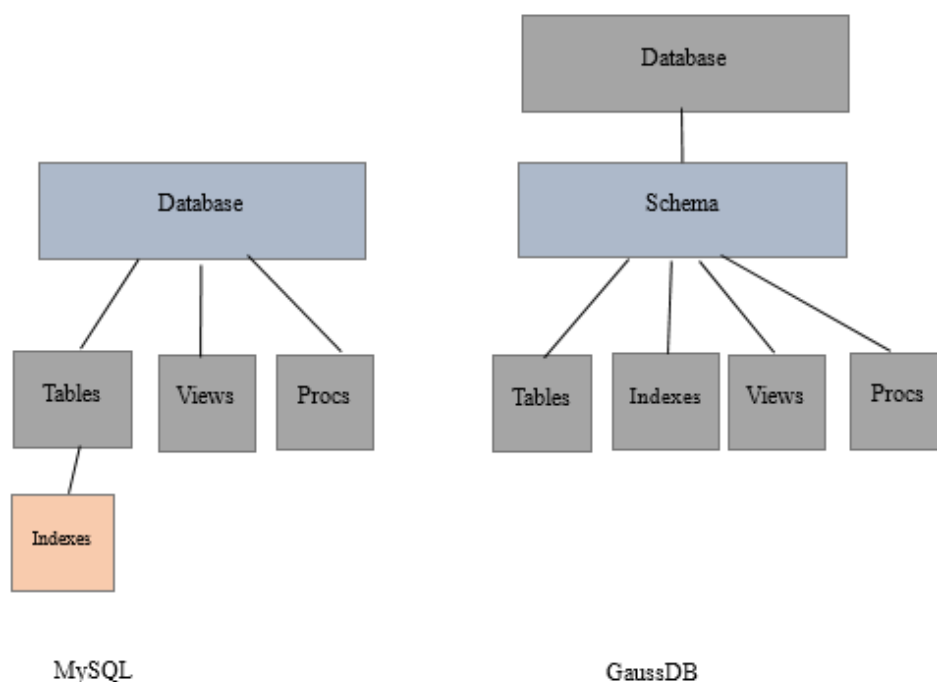
GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

由于GaussDB数据库与MySQL数据库在底层框架实现上存在差异，GaussDB数据库与MySQL数据库仍存在部分差异。

Database 和 Schema 设计

MySQL的数据对象包括DATABASE、TABLE、INDEX、VIEW、TRIGGER、PROC等，MySQL的对象层次跟GaussDB的对应关系是从上至下且一对多包含关系。如下图所示：

图 3-1 MySQL 和 GaussDB 中 Database 和 Schema 之间的差异



- 在MySQL中Database和Schema是同义词；而在GaussDB中，一个Database 下可以有多个Schema。在该特性中，每个MySQL中的Database都被映射到GaussDB的一个Schema。
- 在MySQL中，INDEX从属于一个TABLE，但在GaussDB中，INDEX从属于一个Schema。这个差异导致INDEX名在GaussDB中要求在Schema内唯一，但在MySQL中仅要在在一个表内唯一。这个差异将作为当前约束予以保留。

3.2 MySQL 兼容性 M-Compatibility 模式

3.2.1 数据类型

3.2.1.1 数值数据类型

除特别说明外，MySQL兼容性M-Compatibility模式中的数据类型精度、标度、位数大小等默认不支持用浮点型数值定义，建议使用合法的整型数值定义。

表 3-1 整数类型

MySQL数据库	GaussDB数据库	差异
BOOL	支持，存在差异	输出格式：GaussDB中SELECT TRUE/FALSE输出结果为t/f，MySQL为1/0。
BOOLEAN	支持，存在差异	MySQL：BOOL/BOOLEAN类型实际映射为TINYINT类型。
TINYINT[(M)] [UNSIGNED] [ZEROFILL]	支持，存在差异	输入格式： - MySQL： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下MySQL会发生错误解析，抛出WARNING并插表成功，例如将“1.2.3.4.5”插入表后值为12。 - GaussDB： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下，会将第二个小数点后的字符当作非法字符全部截断，抛出WARNING并插表成功，例如将“1.2.3.4.5”插入表后值为1，“1.6.3.4.5”插入表后值为2。
SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	支持，存在差异	
MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]	支持，存在差异	MySQL存储MEDIUMINT数据需要3字节。 - 带符号的范围是-8,388,608 ~ +8,388,607。 - 无符号的范围是0 ~ +16,777,215。 GaussDB映射为INT类型，存储需要4字节，通过边界值判断限制取值范围。 - 带符号的范围是-8,388,608 ~ +8,388,607。 - 无符号的范围是0 ~ +16,777,215。 其他差异请参见表格下方说明中的内容。

MySQL数据库	GaussDB数据库	差异
INT[(M)] [UNSIGNED] [ZEROFILL]	支持，存在差异	输入格式： - MySQL： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下MySQL会发生错误解析，抛出WARNING并插表成功，例如将“1.2.3.4.5”插入表后值为12。 - GaussDB： 整型类型对于类似“1.2.3.4.5”有多个小数点的字符串形式输入，在宽松模式下，会将第二个小数点后的字符当作非法字符全部截断，抛出WARNING并插表成功，例如将“1.2.3.4.5”插入表后值为1，“1.6.3.4.5”插入表后值为2。
INTEGER[(M)] [UNSIGNED] [ZEROFILL]	支持，存在差异	
BIGINT[(M)] [UNSIGNED] [ZEROFILL]	支持，存在差异	

表 3-2 任意精度类型

MySQL数据库	GaussDB数据库	差异
DECIMAL[(M[,D])] [ZEROFILL]	支持，存在差异	-
NUMERIC[(M[,D])] [ZEROFILL]	支持，存在差异	
DEC[(M[,D])] [ZEROFILL]	支持，存在差异	
FIXED[(M[,D])] [ZEROFILL]	支持，存在差异	

表 3-3 浮点类型

MySQL数据库	GaussDB数据库	差异
FLOAT[(M,D)] [ZEROFILL]	支持，存在差异	FLOAT数据类型不支持KEY键值分区策略分区表。
FLOAT(p) [ZEROFILL]	支持，存在差异	FLOAT数据类型不支持KEY键值分区策略分区表。
DOUBLE[(M,D)] [ZEROFILL]	支持，存在差异	DOUBLE数据类型不支持KEY键值分区策略分区表。

MySQL数据库	GaussDB数据库	差异
DOUBLE PRECISION[(M,D)] [ZEROFILL]	支持，存在差异	DOUBLE PRECISION数据类型不支持KEY键值分区策略分区表。
REAL[(M,D)] [ZEROFILL]	支持，存在差异	REAL数据类型不支持KEY值分区策略分区表。

3.2.1.2 日期与时间数据类型

表 3-4 日期与时间数据类型

MySQL数据库	GaussDB数据库	差异
DATE	支持，存在差异	GaussDB支持DATE数据类型，与MySQL相比规格上存在如下差异： 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。
DATETIME[(fsp)]	支持，存在差异	GaussDB支持DATETIME数据类型，与MySQL相比规格上存在如下差异： 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。

MySQL数据库	GaussDB数据库	差异
TIMESTAMP[(fsp)]	支持，存在差异	GaussDB支持TIMESTAMP数据类型，与MySQL相比规格上存在如下差异： - 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 - MySQL支持设置explicit_defaults_for_timestamp，当explicit_defaults_for_timestamp设置为off时，对TIMESTAMP类型的字段的默认值、插入NULL等处理为非标准行为。MySQL 5.7下，explicit_defaults_for_timestamp默认值为off；MySQL 8.0下，explicit_defaults_for_timestamp默认值为on。GaussDB不支持设置explicit_defaults_for_timestamp，行为与MySQL设置explicit_defaults_for_timestamp为on时的行为一致。explicit_defaults_for_timestamp相关说明参见表格下方内容。
TIME[(fsp)]	支持，存在差异	GaussDB支持TIME数据类型，与MySQL相比规格上存在如下差异： - 反斜杠\在MySQL和GaussDB中都视为转义，但MySQL支持\0，GaussDB暂不支持，因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 - 当时间类型的时、分、秒、纳秒为0时，GaussDB和MySQL可能存在符号位不同的情况。
YEAR[(4)]	支持	-

📖 说明

- GaussDB不支持ODBC语法的字面量：
 { d 'str' }
 { t 'str' }
 { ts 'str' }
- 当给DATETIME、TIME、TIMESTAMP数据类型指定的精度超过其支持的最大精度时，GaussDB会将精度截断成支持的最大精度，MySQL则会报错。
- MySQL中当explicit_defaults_for_timestamp为off时，对TIMESTAMP类型的字段的处理逻辑如下：
 - 未显式指定NULL/NOT NULL属性的字段，将自动添加NOT NULL属性。向此类字段插入NULL值时，会将NULL值替换为当前时间戳。
 - 如果未给表中的第一个TIMESTAMP类型的字段指定NULL属性，会自动给该字段添加DEFAULT CURRENT_TIMESTAMP和ON UPDATE CURRENT_TIMESTAMP属性。
 - 如果未给表中的第二个及之后的TIMESTAMP类型的字段指定NULL属性，会自动给该字段添加DEFAULT '0000-00-00 00:00:00'属性。
- MySQL中当explicit_defaults_for_timestamp为on时，对TIMESTAMP类型的字段的处理逻辑如下：
 - 向TIMESTAMP类型的字段插入NULL值时，不会将NULL值替换为当前时间戳。
 - 未显式指定NULL/NOT NULL属性的字段，将自动添加NULL属性。
 - 向指定了NOT NULL属性的字段插入NULL值时，严格模式下报错，宽松模式下将插入'0000-00-00 00:00:00'。
 - 不会为任何TIMESTAMP类型的字段自动添加DEFAULT CURRENT_TIMESTAMP和ON UPDATE CURRENT_TIMESTAMP属性。

3.2.1.3 字符串数据类型

表 3-5 字符串数据类型

MySQL数据库	GaussDB数据库	差异
CHAR(M)	支持，存在差异	输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。
VARCHAR(M)	支持，存在差异	输入格式： • GaussDB的自定义函数参数和返回值不支持长度校验，存储过程参数不支持长度校验，MySQL支持。 • GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警，MySQL不支持。 • 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。

MySQL数据库	GaussDB数据库	差异
TINYTEXT	支持，存在差异	- 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 - 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 - 索引：MySQL中TINYTEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。
TEXT	支持，存在差异	- 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 - 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 - 索引：MySQL中TEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。
MEDIUMTEXT	支持，存在差异	- 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 - 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 - 索引：MySQL中MEDIUMTEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。
LONGTEXT	支持，存在差异	- 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 默认值：MySQL 5.7不允许设置默认值，GaussDB及MySQL 8.0允许设置默认值。 - 主键：MySQL中创建主键时必须指定前缀长度，GaussDB创建主键时不支持指定前缀长度。 - 索引：MySQL中LONGTEXT类型不支持除前缀索引外其他索引方法，GaussDB支持。

3.2.1.4 二进制数据类型

表 3-6 二进制数据类型

MySQL数据库	GaussDB数据库	差异
BINARY[(M)]	支持，存在差异	- 输入格式： - 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 插入字符串长度小于目标长度时，GaussDB填充符是0x20，MySQL是0x00。 - 字符集：默认字符集为数据库初始化字符集，MySQL默认类型字符集为BINARY字符集。 - 输出格式： - JDBC协议输出时BINARY类型的末尾空格显示为空格，MySQL末尾空格显示为\x00。 - 宽松模式下，BINARY类型面对输入超过n的字节数的字符输入(例如中文字符)，会将超限的整个字符截断。MySQL中会将超限的整个字符的前几位满足n范围内的字节信息保留，但输出时字符信息显示乱码。 - 在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。 说明 GaussDB中，由于BINARY类型填充符和\0截断与MySQL的差异，在操作符比较计算，字符串相关系统函数计算，索引匹配，数据导入导出等场景下与MySQL的表现会存在差异。差异场景请见本节示例。
VARBINARY(M)	支持，存在差异	- 输入格式：输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 字符集：默认字符集为数据库初始化字符集，MySQL默认类型字符集为BINARY字符集。 - 输出格式： - JDBC协议输出时BINARY类型的末尾空格显示为空格，MySQL末尾空格显示为\x00。 - 在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。

MySQL数据库	GaussDB数据库	差异
TINYBLOB	支持，存在差异	- 输入格式： - 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 - 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 主键：MySQL中TINYBLOB类型不支持主键，GaussDB支持。 - 索引：MySQL中TINYBLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 - 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 - 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。
BLOB	支持，存在差异	- 输入格式： - 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 - 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 主键：MySQL中BLOB类型不支持主键，GaussDB支持。 - 索引：MySQL中BLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 - 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 - 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。

MySQL数据库	GaussDB数据库	差异
MEDIUMBLOB	支持，存在差异	- 输入格式： - 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 - 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 主键：MySQL中MEDIUMBLOB类型不支持主键，GaussDB支持。 - 索引：MySQL中MEDIUMBLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 - 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 - 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。
LONGBLOB	支持，存在差异	- 取值范围：只支持不超过1GB字节长度，MySQL支持4GB-1字节长度。 - 输入格式： - 默认值：创建表列时语法上允许设置默认值，MySQL不允许设置默认值。 - 输入二进制或十六进制字符串时，GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。 - 主键：MySQL中LONGBLOB类型不支持主键，GaussDB支持。 - 索引：MySQL中LONGBLOB类型不支持除前缀索引外其他索引方法，GaussDB支持。 - 外键：MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列，GaussDB支持。 - 输出格式：在MySQL 8.0以上版本，默认以0x开头形式返回，GaussDB以多个\x形式返回。
BIT[(M)]	支持，存在差异	输出格式： - 所有输出按照二进制字符串形式输出。MySQL中根据ASCII码表转义，无法转义的输出为空。 - 在MySQL 8.0以上版本，默认会开头补0，GaussDB不会补0。

示例：

```
-- GaussDB场景
m_db=# CREATE TABLE test(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test VALUES(0x8000);
INSERT 0 1
```

```

m_db=# SELECT hex(a) FROM test;
      hex
-----
80202020202020202020
(1 row)

m_db=# SELECT * FROM test WHERE hex(a) = 80000000000000000000;
 a
--
(0 rows)

m_db=# CREATE TABLE test2(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test2 VALUES(0x80008000);
INSERT 0 1
m_db=# SELECT hex(a) FROM test2;
      hex
-----
80202020202020202020
(1 row)

m_db=# DROP TABLE test;
DROP TABLE
m_db=# DROP TABLE test2;
DROP TABLE

-- MySQL场景
mysql> CREATE TABLE test(a BINARY(10));
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO test VALUES(0x8000);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT hex(a) FROM test;
+-----+
| hex(a) |
+-----+
| 80000000000000000000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM test WHERE hex(a) = 80000000000000000000;
+-----+
| a |
+-----+
| 80 |
+-----+
1 row in set (0.00 sec)

mysql> CREATE TABLE test2(a binary(10));
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO test2 VALUES(0x80008000);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT hex(a) FROM test2;
+-----+
| hex(a) |
+-----+
| 80008000000000000000 |
+-----+
1 row in set (0.00 sec)
mysql> DROP TABLE test;
Query OK, 0 rows affected (0.00 sec)
mysql> DROP TABLE test2;
Query OK, 0 rows affected (0.00 sec)

```

3.2.1.5 数据类型支持的属性

表 3-7 数据类型支持的属性

MySQL数据库	GaussDB数据库
NULL	支持
NOT NULL	支持
DEFAULT	支持
ON UPDATE	支持
PRIMARY KEY	支持
AUTO_INCREMENT	支持
CHARACTER SET name	支持
COLLATE name	支持
ZEROFILL	支持

使用CREATE TABLE AS方式建表，对VARBINARY类型的字段设置默认值，在使用SHOW CREATE TABLE、DESC或\d 查询的时候回显与MySQL存在差异，GaussDB显示为转换成十六进制后的值，而MySQL显示为原值。

示例：

```
m_db=# CREATE TABLE test_int(
    int_col INT
);
m_db=# CREATE TABLE test_varbinary(
    varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
m_db=# SHOW CREATE TABLE test_varbinary;
      Table      | Create Table
+-----+-----+
test_varbinary | SET search_path = public;
               | CREATE TABLE test_varbinary (
               |   varbinary_col varbinary(20) DEFAULT X'6761757373',
               |   int_col integer
               | )
               | CHARACTER SET = "UTF8" COLLATE = "utf8mb4_general_ci"
               | WITH (orientation=row, compression=no, storage_type=USTORE, segment=off);
(1 row)
m_db=# DROP TABLE test_int, test_varbinary;

mysql> CREATE TABLE test_int(
    int_col INT
);
mysql> CREATE TABLE test_varbinary(
    varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
mysql> SHOW CREATE TABLE test_varbinary;
+-----+
| Table      | Create
+-----+
| test_varbinary | CREATE TABLE `test_varbinary` (
```

```
`varbinary_col` varbinary(20) DEFAULT 'gauss',
`int_vol` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |
+-----+
+-----+
1 row in set (0.00 sec)
mysql> DROP TABLE test_int, test_varbinary;
```

3.2.1.6 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换：

- 操作符（比较操作符、运算操作符等）的操作数的数据类型不一致。常见于查询条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新（包括INSERT、UPDATE、MERGE、REPLACE等）的目标列，数据的类型和列的定义类型不一致。
- 显式的类型转换：CAST(expr AS datatype)，将expr表达式类型转换为datatype类型。
- 集合运算（UNION以及EXCEPT）确定最终投影列的目标数据类型后，各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景，根据不同表达式的数据类型，来决定用于比较或者最终结果的目标数据类型。

数据类型转换差异点主要分为三类：隐式转换，显式转换和UNION/CASE。

双冒号转换差异点

- GaussDB中使用双冒号将函数入参转换为期望类型可能导致结果超出预期，且MySQL中无双冒号功能。

示例：

```
m_db=# SELECT POW("12"::VARBINARY,"12"::VARBINARY);
ERROR: value out of range: overflow
CONTEXT: referenced column: pow

varbinary col
m_db=# CREATE TABLE test_varbinary (
    A VARBINARY(10)
);
m_db=# INSERT INTO test_varbinary VALUES ('12');
m_db=# SELECT POW(A, A) FROM test_varbinary;
    pow
-----
8916100448256
(1 row)
```

隐式类型转换差异点

- GaussDB中统一平铺成小类型到小类型的转换规则，MySQL中使用小类型转大类型，大类型转小类型的转换规则。
- GaussDB中隐式转换因数据类型本身差异点，输出格式存在部分行为不一致。
- GaussDB中的隐式转换，BIT数据类型到字符数据类型和二进制数据类型转换，输出存在部分行为不一致。GaussDB输出为十六进制，MySQL中根据ASCII码表转义，无法转义的输出为空。

示例：

```
m_db=# CREATE TABLE bit_storage (
    VS_COL1 BIT(4),
```

```
VS_COL2 BIT(4),
VS_COL3 BIT(4),
VS_COL4 BIT(4),
VS_COL5 BIT(4),
VS_COL6 BIT(4),
VS_COL7 BIT(4),
VS_COL8 BIT(4)
) DISTRIBUTE BY REPLICATION;
m_db=# CREATE TABLE string_storage (
  VS_COL1 BLOB,
  VS_COL2 TINYBLOB,
  VS_COL3 MEDIUMBLOB,
  VS_COL4 LONGBLOB,
  VS_COL5 TEXT,
  VS_COL6 TINYTEXT,
  VS_COL7 MEDIUMTEXT,
  VS_COL8 LONGTEXT
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
m_db=# INSERT INTO string_storage SELECT * FROM bit_storage;
m_db=# SELECT * FROM string_storage;
VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8
-----+-----+-----+-----+-----+-----+-----+-----
\x05   | \x05   | \x05   | \x05   | \x05   | \x05   | \x05   | \x05
(1 row)
m_db=# DROP TABLE bit_storage, string_storage;

mysql> CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS_COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS_COL6 BIT(4),
  VS_COL7 BIT(4),
  VS_COL8 BIT(4)
);
mysql> CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS_COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS_COL6 BIT(4),
  VS_COL7 BIT(4),
  VS_COL8 BIT(4)
);
mysql> INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
mysql> INSERT INTO string_storage SELECT * FROM bit_storage;
mysql> SELECT * FROM string_storage;
-----+-----+-----+-----+-----+-----+-----+-----
| VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8 |
-----+-----+-----+-----+-----+-----+-----+-----
|         |         |         |         |         |         |         |         |
-----+-----+-----+-----+-----+-----+-----+-----
1 row in set (0.00 sec)
mysql> DROP TABLE bit_storage, string_storage;
```

- 中间带有0x00的二进制或十六进制字符串插入二进制数据类型时，GaussDB部分插入，0x00后的字符会截断，MySQL可以完整插入。

示例：

```
m_db=# CREATE TABLE blob_storage (
  A BLOB
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO blob_storage VALUES (0xBB00BB);
m_db=# SELECT hex(A) FROM blob_storage;
hex
----
BB
```

```
(1 row)
m_db=# DROP TABLE blob_storage;

mysql> CREATE TABLE blob_storage (
    A BLOB
);
mysql> INSERT INTO blob_storage VALUES (0xBB00BB);
mysql> SELECT hex(A) FROM blob_storage;
+-----+
| hex(a) |
+-----+
| BB00BB |
+-----+
1 row in set (0.01 sec)
mysql> DROP TABLE blob_storage;
```

- 中间带有0x00的二进制或十六进制字符串插入字符串数据类型时，GaussDB部分插入，0x00后的字符会截断，MySQL严格模式不允许插入，宽松模式插入空字符串。

示例：

```
m_db=# CREATE TABLE text_storage (
    A TEXT
);
m_db=# INSERT INTO text_storage VALUES (b'101110110000000010111011');
m_db=# SELECT hex(A) FROM text_storage;
hex
-----
BB
(1 row)
m_db=# DROP TABLE text_storage;

mysql> CREATE TABLE text_storage (
    A TEXT
);
mysql> INSERT INTO text_storage VALUES (b'101110110000000010111011');
ERROR 1366 (HY000): Incorrect string value: '\xBB\x00\xBB' for column 'A' at row 1
mysql> SELECT hex(A) FROM text_storage;
Empty set (0.00 sec)
mysql> SET SQL_MODE="";
mysql> INSERT INTO text_storage VALUES (b'101110110000000010111011');
mysql> SELECT hex(A) FROM text_storage;
+-----+
| hex(A) |
+-----+
|      |
+-----+
1 row in set (0.01 sec)
mysql> DROP TABLE text_storage;
```

- WHERE子句中只带有普通字符串，GaussDB中't'、'true'、'yes'、'y'、'on'返回TRUE，'no'、'f'、'off'、'false'、'n'返回FALSE，其余字符串报错。MySQL通过字符串转换为INT1判断返回TRUE/FALSE。

示例：

```
m_db=# CREATE TABLE test_where (
    A INT
);
m_db=# INSERT INTO test_where VALUES (1);
m_db=# SELECT * FROM test_where WHERE '111';
ERROR: invalid input syntax for type boolean: "111"
LINE 1: SELECT * FROM test_where WHERE '111';
m_db=# DROP TABLE test_where;

mysql> CREATE TABLE test_where (
    A INT
);
mysql> INSERT INTO test_where VALUES (1);
mysql> SELECT * FROM test_where WHERE '111';
```

```
+-----+
| a |
+-----+
| 1 |
+-----+
1 row in set (0.01 sec)
mysql> DROP TABLE test_where;
```

- 对于YEAR类型的输入，在将字符串转换为整型的过程中，MySQL考虑科学计数法，GaussDB暂不支持，统一做截断处理。

示例：

```
m_db=# CREATE TABLE test_year (
      A YEAR
);
m_db=# SET sql_mode = '';
m_db=# INSERT INTO test_year VALUES ('2E3x');
WARNING: Data truncated for column.
LINE 1: INSERT INTO test_year VALUES ('2E3x');
      ^
CONTEXT: referenced column: a
m_db=# SELECT * FROM test_year ORDER BY A;
      a
-----
2002
(1 row)
m_db=# DROP TABLE test_year;

mysql> CREATE TABLE test_year (
      A YEAR
);
mysql> INSERT INTO test_year VALUES ('2E3x');
mysql> SELECT * FROM test_year ORDER BY A;
+-----+
| a |
+-----+
| 2000 |
+-----+
1 row in set (0.01 sec)
mysql> DROP TABLE test_year;
```

显式类型转换差异点

- GaussDB中平铺成对各目标类型的转换规则，MySQL中使用C++多态重载函数，在嵌套场景中存在不一致行为。

示例：

```
m_db=# SELECT CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED);
WARNING: Truncated incorrect INTEGER value: '2023-01-01'
CONTEXT: referenced column: cast
      cast
-----
2023
(1 row)

mysql> SELECT CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED);
+-----+
| CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED) |
+-----+
| 20230101 |
+-----+
```

UNION，CASE 和相关构造差异点

- POLYGON + NULL、POINT + NULL、POLYGON + POINT组合在MySQL中均返回GEOMETRY类型，GaussDB中未涉及，暂时当做报错处理。

- SET和ENUM两种类型暂未支持，暂时当做报错处理。
- 常量类型和其他类型做类型聚合的时候，输出类型的精度为其他类型的精度。如“SELECT "helloworld" UNION SELECT p FROM t;”的结果的精度为属性p的精度。
- 定点常量和不带精度约束的类型（非字符串类型如int、bool、year等，聚合结果类型为定点类型）聚合时，精度约束会按照定点数默认精度31输出。
- merge rule差异：
MySQL 5.7中YEAR和TINYINT、INT、MEDIUMINT、BIGINT、BOOL聚合的结果类型为带UNSIGNED的类型，GaussDB的结果类型为不带UNSIGNED的类型；MySQL中BIT和INT、NUMERIC、FLOAT、DOUBLE等数值类型的聚合类型为VARBINARY类型，GaussDB中BIT和INT|NUMERIC聚合为NUMERIC类型，和FLOAT或者DOUBLE类型聚合时，结果为DOUBLE类型，和无符号整型聚合时为UINT8类型。
- MySQL中BINARY和CHAR填充字符不相同，BINARY填充'\0'，CHAR填充空格，GaussDB中BINARY和CHAR都是填充空格。

3.2.2 系统函数

3.2.2.1 系统函数兼容性概述

GaussDB数据库兼容绝大多数MySQL的系统函数，但存在部分差异。

当前存在原GaussDB的系统函数和MySQL系统函数同名，但是M-Compatibility兼容模式下尚未支持这些函数的情况；一部分未支持的同名函数会提示用户在M-Compatibility兼容模式下不支持，另外一部分同名函数仍然保持原GaussDB系统函数的行为。同名函数会与MySQL的行为产生较大差异，因此建议用户尽量避免使用这些同名函数，只使用M-Compatibility兼容模式下的系统函数。

具体同名函数列表如下：

表 3-8 M-Compatibility 兼容模式下提示不支持的同名函数

cot	isEmpty	last_insert_id	mod	octet_length
overlaps	point	radians	regexp_instr	regexp_like
regexp_replace	regexp_substr	stddev_pop	stddev_samp	var_pop
var_samp	variance	-	-	-

表 3-9 M-Compatibility 兼容模式下保持原 GaussDB 系统函数行为的同名函数

ceil	decode	encode	format	instr
position	round	stddev	row_num	-

说明

- MySQL数据库支持通过可加载函数接口，向MySQL中添加自定义函数，在调用此类函数时，函数的入参支持指定别名。GaussDB不支持可加载函数接口，在调用函数时，函数入参不支持指定别名。
- M-Compatibility模式下，系统函数存在以下公共差异：
 - 系统函数的返回值类型仅考虑入参node类型为Var（表中数据）和Const（常量输入）类型时的情况与MySQL保持一致，其他情况（如入参为运算表达式、函数表达式等）可能返回值的类型与MySQL有差异。
 - 当聚合函数以其他函数、操作符或SELECT子句等表达式作为入参时（如SELECT sum(abs(n)) FROM t;），聚合函数将获取不到入参表达式传递的精度信息，导致函数的结果精度与MySQL有差异。
 - 系统函数的调用不推荐使用pg_catalog.func_name()形式的调用，当被调用函数存在语法形式的入参时（如SELECT substr('demo' from 1 for 2)），函数的调用可能存在错误。

3.2.2.2 流程控制函数

表 3-10 流程控制函数列表

MySQL数据库	GaussDB数据库	差异
IF()	支持，存在差异	当第一个参数为TRUE且第三个参数表达式中存在隐式类型转换错误，或者第一个参数为FALSE且第二个参数表达式中存在隐式类型转换错误时，MySQL会忽略该错误，GaussDB会提示类型转换错误。
IFNULL()	支持，存在差异	第一个参数不为NULL且第二个参数表达式中存在隐式类型转换错误时，MySQL会忽略该错误，GaussDB会提示类型转换错误。
NULLIF()	支持，存在差异	函数返回值类型在MySQL 5.7和MySQL 8.0中存在差异，考虑到MySQL 8.0更合理，因此函数返回值类型兼容MySQL 8.0。

3.2.2.3 日期和时间函数

以下为GaussDB数据库M-Compatibility兼容性日期时间函数公共说明，与MySQL行为一致。

- 函数入参为时间类型表达式的情况：

时间类型表达式主要包括TEXT、DATETIME、DATE或TIME，但所有可以隐式转换为时间表达式的类型都可以作为入参，比如数字类型可以通过先隐式转化为TEXT，再作为时间类型表达式生效。

但是，不同函数的具体生效情况会有所不同。例如：DATEDIFF函数仅计算日期之间的差值，因此时间表达式会被解析为日期；而TIMESTAMPDIFF函数在计算时间差值时，会根据UNIT参数来决定将时间表达式解析为DATE、TIME或DATETIME。
- 函数入参为无效日期的情况：

一般而言，日期时间函数支持DATE、DATETIME的范围和MySQL保持一致。DATE支持的范围为'0000-01-01'到'9999-12-31'，DATETIME支持的范围为'0000-01-01 00:00:00'到'9999-12-31 23:59:59'。虽然GaussDB支持的DATE、DATETIME范围大于MySQL，但是越界仍然算无效日期。

大部分时间函数对于入参为无效时间时，会告警并返回NULL，只有能通过cast正常转换的日期，才是正常合理的日期。

GaussDB M-Compatibility兼容性框架下GaussDB的大部分日期时间函数与MySQL一致，一些函数的差异如下表所示：

表 3-11 日期与和时间函数列表

MySQL数据库	GaussDB数据库	差异
ADDDATE()	支持	-
ADDTIME()	支持	-
CONVERT_TZ()	支持	-
CURDATE()	支持	-
CURRENT_DATE() / CURRENT_DATE	支持	-
CURRENT_TIME() / CURRENT_TIME	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数环绕（例：SELECT CURRENT_TIME(257) == SELECT CURRENT_TIME(1)）。 GaussDB只支持[0,6]合法值，其他值报错。
CURRENT_TIMESTAMP() / CURRENT_TIMESTAMP	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数环绕（例：SELECT CURRENT_TIMESTAMP(257) == SELECT CURRENT_TIMESTAMP(1)）。 GaussDB只支持[0,6]合法值，其他值报错。
CURTIME()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数环绕（例：SELECT CURTIME(257) == SELECT CURTIME(1)）。 GaussDB只支持[0,6]合法值，其他值报错。
DATE()	支持	-
DATE_ADD()	支持	-
DATE_FORMAT()	支持	-
DATE_SUB()	支持	-
DATEDIFF()	支持	-
DAY()	支持	-

MySQL数据库	GaussDB数据库	差异
DAYNAME()	支持	-
DAYOFMONTH()	支持	-
DAYOFWEEK())	支持	-
DAYOFYEAR()	支持	-
EXTRACT()	支持	-
FROM_DAYS())	支持	-
FROM_UNIXTIME()	支持	-
GET_FORMAT()	支持	-
HOUR()	支持	-
LAST_DAY()	支持	-
LOCALTIME() /LOCALTIME	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数环绕（例SELECT LOCALTIME(257) == SELECT LOCALTIME(1)）。 GaussDB只支持[0,6]合法值，其他值报错。
LOCALTIMESTAMP/ LOCALTIMESTAMP()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数环绕（例SELECT LOCALTIMESTAMP(257) == SELECT LOCALTIMESTAMP(1)）。 GaussDB只支持[0,6]合法值，其他值报错。
MAKEDATE()	支持	-
MAKETIME()	支持，存在差异	分布式下推场景下当TIME类型秒位无精度时，MySQL默认补齐6个0，GaussDB不做补齐。
MICROSECOND()	支持	-
MINUTE()	支持	-
MONTH()	支持	-
MONTHNAME()	支持	-
NOW()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数环绕（例SELECT NOW(257) == SELECT NOW(1)）。 GaussDB只支持[0,6]合法值，其他值报错。

MySQL数据库	GaussDB数据库	差异
PERIOD_ADD()	支持，存在差异	- 整数溢出处理的行为： MySQL在5.7版本，此函数入参和结果的最大值都为$2^{32}=4294967296$，在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题；在MySQL 8.0中已修复此问题。GaussDB下此函数的表现与MySQL 8.0版本保持一致。 - 负数period的表现： MySQL在5.7版本，会将负数年份解析为异常值而不是报错。GaussDB入参或结果（如100年1月减去10000月）出现负数时报错。在MySQL 8.0中已修复此问题。GaussDB下此函数的表现与MySQL 8.0版本保持一致。 - period月份越界的表现： MySQL在5.7版本中，若月份大于12或等于0，例如200013、199900，会将其顺延到之后的年份，或者将0月作为上一年12月处理。GaussDB会对越界月份进行报错。在MySQL 8.0中已修复此问题。GaussDB下此函数的表现与MySQL 8.0版本保持一致。
PERIOD_DIFF()	支持，存在差异	- 整数溢出处理的行为： MySQL在5.7版本，此函数入参和结果的最大值都为$2^{32}=4294967296$，在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题；在MySQL 8.0中已修复此问题。GaussDB下此函数的表现与MySQL 8.0版本保持一致。 - 负数period的表现： MySQL在5.7版本，会将负数年份解析为异常值而不是报错。GaussDB入参或结果（如100年1月减去10000月）出现负数时报错。在MySQL 8.0中已修复此问题。GaussDB下此函数的表现与MySQL 8.0版本保持一致。 - period月份越界的表现： MySQL在5.7版本中，若月份大于12或等于0，例如200013、199900，会将其顺延到之后的年份，或者将0月作为上一年12月处理。GaussDB会对越界月份进行报错。在MySQL 8.0中已修复此问题。GaussDB下此函数的表现与MySQL 8.0版本保持一致。
QUARTER()	支持	-
SEC_TO_TIME()	支持	-
SECOND()	支持	-

MySQL数据库	GaussDB数据库	差异
STR_TO_DATE()	支持，存在差异	返回值类型与MySQL有差异，GaussDB返回的是text，MySQL返回的是datetime、date。
SUBDATE()	支持	-
SUBTIME()	支持	-
SYSDATE()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数回绕。 GaussDB不回绕。
TIME()	支持	-
TIME_FORMAT()	支持	-
TIME_TO_SEC()	支持	-
TIMEDIFF()	支持	-
TIMESTAMP()	支持	-
TIMESTAMPADD()	支持	-
TIMESTAMPDIFF()	支持	-
TO_DAYS()	支持	-
TO_SECONDS()	支持	-
UNIX_TIMESTAMP()	支持，存在差异	MySQL会根据入参是否存在小数位，决定返回定点型还是整型。当前GaussDB在内层嵌套操作符或函数时，返回的类型与MySQL可能存在不同。当内层节点返回定点、浮点、字符型、时间类型（不包括date类型）时，MySQL可能返回整型，GaussDB会返回定点型。
UTC_DATE()	支持	-
UTC_TIME()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数回绕，GaussDB只支持[0,6]合法值，其他值报错。
UTC_TIMESTAMP()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数回绕，GaussDB只支持[0,6]合法值，其他值报错。
WEEK()	支持	-
WEEKDAY()	支持	-
WEEKOFYEAR()	支持	-

MySQL数据库	GaussDB数据库	差异
YEAR()	支持	-
YEARWEEK()	支持	-

3.2.2.4 字符串函数

表 3-12 字符串函数列表

MySQL数据库	GaussDB数据库	差异
ASCII()	支持	-
BIT_LENGTH())	支持	-
CHAR_LENGTH())	支持，存在差异	GaussDB此函数如果数据库字符集是SQL_ASCII，CHAR_LENGTH()会返回字节数而非字符数。
CHARACTER_LENGTH())	支持，存在差异	GaussDB此函数如果数据库字符集是SQL_ASCII，CHARACTER_LENGTH()会返回字节数而非字符数。
CONCAT()	支持，存在差异	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
CONCAT_WS())	支持，存在差异	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
HEX()	支持	-
LENGTH()	支持	-

MySQL数据库	GaussDB数据库	差异
LPAD()	支持，存在差异	- MySQL默认最大填充长度为1398101，GaussDB默认最大长度为1048576。在不同字符集下，最大填充长度会有差异，例如字符集为GBK时，GaussDB默认最大长度为2097152。 - 当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。 - 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
REPEAT()	支持，存在差异	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
REPLACE()	支持，存在差异	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。

MySQL数据库	GaussDB数据库	差异
RPAD()	支持，存在差异	- MySQL默认最大填充长度为1398101，GaussDB默认最大长度为1048576。在不同字符集下，最大填充长度会有差异，例如字符集为GBK时，GaussDB默认最大长度为2097152。 - 当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。 - 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。
SPACE()	支持	-
STRCMP()	支持，存在差异	当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。
FIND_IN_SET() LCASE() LEFT() LOWER() LTRIM() REVERSE() RIGHT() RTRIM() SUBSTR() SUBSTRING() SUBSTRING_INDEX() TRIM()	支持，存在差异	当GaussDB使用的字符编码是SQL_ASCII时，服务器会根据ASCII标准对字节值0~127进行解释，而字节值128~255则当作无法解析的字符。如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符，从而与MySQL的行为产生较大差异。 当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。

MySQL数据库	GaussDB数据库	差异
UCASE()		
UPPER()		
UNHEX()	支持，存在差异	MySQL的返回值类型为BINARY、VARBINARY、BLOB、MEDIUMBLOB或LONGBLOB；GaussDB返回值类型固定为LONGBLOB。
FIELD()	支持	-
FORMAT()	支持	-

3.2.2.5 强制转换函数

表 3-13 强制转换函数列表

MySQL数据库	GaussDB数据库	差异
CAST()	支持，存在差异	- GaussDB不支持使用CAST(expr AS CHAR[(N)] charset_info或者CAST(expr AS NCHAR[(N)]) 转换字符集。 - GaussDB支持使用CAST(expr AS FLOAT[(p)]) 或CAST(expr AS DOUBLE)将表达式转换为浮点类型，MySQL 5.7版本不支持此转换。 - GaussDB不支持使用CAST(expr AS JSON)将表达式转换为JSON。 - 对于CAST嵌套子查询场景，如果子查询语句返回的是FLOAT类型，GaussDB返回的是准确的数值，MySQL 5.7版本返回失真数值，BINARY函数使用CAST实现，同理。 <pre>--GaussDB m_db=# CREATE TABLE sub_query_table(myfloat float) DISTRIBUTE BY REPLICATION; CREATE TABLE m_db=# INSERT INTO sub_query_table(myfloat) VALUES (1.23); INSERT 0 1 m_db=# SELECT binary(SELECT myfloat FROM sub_query_table) FROM sub_query_table; binary ----- 1.23 (1 row) m_db=# SELECT cast((SELECT myfloat FROM sub_query_table) AS char); cast ----- 1.23 (1 row) --MySQL 5.7 mysql> CREATE TABLE sub_query_table(myfloat float); Query OK, 0 rows affected (0.02 sec) mysql> INSERT INTO sub_query_table(myfloat) VALUES (1.23); Query OK, 1 row affected (0.00 sec) mysql> SELECT binary(SELECT myfloat FROM sub_query_table) FROM sub_query_table; +-----+ binary(SELECT myfloat FROM sub_query_table) +-----+ 1.2300000190734863 +-----+ 1 row in set (0.00 sec) mysql> SELECT cast((SELECT myfloat FROM sub_query_table) AS char); +-----+ cast((SELECT myfloat FROM sub_query_table) AS char) +-----+ 1.2300000190734863 +-----+ 1 row in set (0.00 sec)</pre>

MySQL数据库	GaussDB数据库	差异
CONVERT()	支持，存在差异	• GaussDB不支持使用CONVERT(expr, CHAR[(N)] charset_info或者CAST(expr, NCHAR[(N)])转换字符集。 • GaussDB支持使用CONVERT(expr, FLOAT[(p)])或CONVERT(expr, DOUBLE)将表达式转换为浮点类型，MySQL 5.7版本不支持此转换。 • GaussDB不支持使用CONVERT(expr, JSON)将表达式转换为JSON。

3.2.2.6 加密函数

表 3-14 加密函数列表

MySQL数据库	GaussDB数据库	差异
AES_DECRYPT()	支持，存在差异	• ecb为不安全加密模式，GaussDB不支持，默认为cbc模式。
AES_ENCRYPT()	支持，存在差异	• GaussDB中，当指定数据库使用的字符编码是SQL_ASCII时，服务器把字节值0~127根据ASCII标准解释，而字节值128~255则当作无法解析的字符；如果该函数的输入输出包含了任何非ASCII数据，数据库将无法帮助用户转换或者校验非ASCII字符。 • MySQL的返回值类型为BINARY、VARBINARY、BLOB、MEDIUMBLOB、LONGBLOB，GaussDB返回值类型固定为LONGBLOB。
SHA()/SHA1()	支持	-
SHA2()	支持	-

3.2.2.7 比较函数

表 3-15 比较函数列表

MySQL数据库	GaussDB数据库	差异
COALESCE()	支持，存在差异	union distinct场景下，返回值精度与MySQL不完全一致。 当第一个不为NULL的参数的后续参数表达式中存在隐式类型转换错误时，MySQL会忽略该错误，GaussDB会提示类型转换错误。当参数为MIN函数、MAX函数时，返回值类型与MySQL不一致。
INTERVAL()	支持	-
GREATEST()	支持，存在差异	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。 当该函数入参含有NULL且在WHERE关键字之后调用，返回结果与MySQL 5.7不一致，此处为MySQL 5.7存在的问题，MySQL 8.0修复了该问题，目前GaussDB和MySQL 8.0保持一致。
LEAST()	支持，存在差异	当MySQL返回值类型为二进制字符串类型（BINARY、VARBINARY、BLOB等）时，GaussDB对应的返回值类型为LONGBLOB；当MySQL返回值类型为非二进制字符串类型（CHAR、VARCHAR、TEXT等）时，GaussDB对应的返回值类型为TEXT。 当该函数入参含有NULL且在WHERE关键字之后调用，返回结果与MySQL 5.7不一致，此处为MySQL 5.7存在的问题，MySQL 8.0修复了该问题，目前GaussDB和MySQL 8.0保持一致。
ISNULL()	支持	-

3.2.2.8 聚合函数

表 3-16 聚合函数列表

MySQL数据库	GaussDB数据库	差异
AVG()	支持，存在差异	- GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 - GaussDB中当expr中的列为BIT、BOOL、整数类型，且所有行的和超过BIGINT的范围时，会发生溢出导致整数翻转。
BIT_AND()	支持	-
BIT_OR()	支持	-
BIT_XOR()	支持	-
COUNT()	支持，存在差异	GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。
GROUP_CONCAT()	支持，存在差异	- GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 - GROUP_CONCAT返回类型为二进制类型时，只返回BLOB类型，其他情况返回TEXT类型。MySQL还会根据返回长度返回LONGTEXT/TINYTEXT/LONGBLOB/TINYBLOB类型。 - GaussDB中当GROUP_CONCAT参数中同时有DISTINCT和ORDER BY语法时，所有ORDER BY后的表达式必须也在DISTINCT的表达式之中。 - GaussDB中GROUP_CONCAT(... ORDER BY 数字)不代表按照第几个参数的顺序，数字只是一个常量表达式，相当于不排序。 - GaussDB中使用参数group_concat_max_len限制GROUP_CONCAT最大返回长度，超长截断，目前能返回的最大长度是1073741823，小于MySQL。

MySQL数据库	GaussDB数据库	差异
MAX() MIN()	支持，存在差异	- GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 - 当参数为非表字段时，MAX函数、MIN函数返回值类型和MySQL 5.7不一致。 - 当参数为FLOAT类型时，MAX函数、MIN函数的返回值结果与MySQL 5.7一致。MySQL 5.7和MySQL 8.0有行为差异，导致MAX函数、MIN函数嵌套CAST(expr AS FLOAT[(p)])时的返回结果与MySQL 8.0有差异。 <pre> -- GaussDB: m_db=# CREATE TABLE t1(c1 float); CREATE TABLE m_db=# INSERT INTO t1 VALUES(1.2); INSERT 0 1 m_db=# SELECT MAX(c1) FROM t1; max ----- 1.2000000476837158 (1 row) m_db=# SELECT MAX(CAST(1.2 AS FLOAT)); max ----- 1.2000000476837158 (1 row) m_db=# DROP TABLE t1; DROP TABLE -- MySQL 5.7: mysql> CREATE TABLE t1(c1 float); Query OK, 0 rows affected (0.02 sec) mysql> INSERT INTO t1 VALUES(1.2); Query OK, 1 row affected (0.00 sec) mysql> SELECT MAX(c1) FROM t1; +-----+ MAX(c1) +-----+ 1.2000000476837158 +-----+ 1 row in set (0.00 sec) -- MySQL5.7不支持CAST(expr AS FLOAT[(p)])表达式 mysql> SELECT MAX(CAST(1.2 AS FLOAT)); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'FLOAT))' at line 1 mysql> DROP TABLE t1; Query OK, 0 rows affected (0.01 sec) -- MySQL 8.0: mysql> CREATE TABLE t1(c1 float); Query OK, 0 rows affected (0.03 sec) </pre>

MySQL数据库	GaussDB数据库	差异
		<pre>mysql> INSERT INTO t1 VALUES(1.2); Query OK, 1 row affected (0.00 sec) mysql> SELECT MAX(c1) FROM t1; +-----+ MAX(c1) +-----+ 1.2 +-----+ 1 row in set (0.00 sec) mysql> SELECT MAX(CAST(1.2 AS FLOAT)); +-----+ MAX(CAST(1.2 AS FLOAT)) +-----+ 1.2 +-----+ 1 row in set (0.00 sec) mysql> DROP TABLE t1; Query OK, 0 rows affected (0.01 sec)</pre>
SUM()	支持，存在差异	- GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时，不对结果进行排序，MySQL会进行排序。 - GaussDB中当expr中的列为BIT类型、BOOL类型、整数类型，且所有行的和超过BIGINT的范围时，会发生溢出导致整数翻转。

3.2.2.9 数字操作函数

表 3-17 数字操作函数列表

MySQL数据库	GaussDB数据库	差异
ABS()	支持	-
ACOS()	支持	-
ASIN()	支持	-
ATAN()	支持	-
ATAN2()	支持	-
CEILING()	支持，存在差异	部分操作结果类型与MySQL不一致。对于推导结果类型为NUMERIC或者整型的，如果能够被整型类型存储，MySQL中结果类型为整型；GaussDB中类型仍然为NUMERIC类型。
COS()	支持	-
DEGREES()	支持	-

MySQL数据库	GaussDB数据库	差异
EXP()	支持	-
FLOOR()	支持，存在差异	FLOOR函数的返回值类型与MySQL的有差异：入参类型为INT，GaussDB返回值类型为BIGINT，MySQL返回值类型为INT。 部分操作结果类型与MySQL不一致。对于推导结果类型为NUMERIC或者整型的，如果能够被整型类型存储，MySQL中结果类型为整型；GaussDB中类型仍然为NUMERIC类型。
LN()	支持	-
LOG()	支持	-
LOG10()	支持	-
LOG2()	支持	-
PI()	支持，存在差异	PI函数的返回值精度与MySQL的有差异：MySQL中PI函数的结果仅保留四舍五入之后的小数后6位，而GaussDB的结果会保留四舍五入之后的小数后15位。
POW()	支持	-
POWER()	支持	-
RAND()	支持	-
SIGN()	支持	-
SIN()	支持	-
SQRT()	支持	-
TAN()	支持	-
TRUNCATE()	支持	-
CEIL()	支持	-

3.2.2.10 其他函数

表 3-18 其他函数列表

MySQL数据库	GaussDB数据库	差异
DATABASE()	支持	-
UUID()	支持	-

MySQL数据库	GaussDB数据库	差异
UUID_SHORT()	支持	-

3.2.3 操作符

GaussDB数据库兼容绝大多数MySQL的操作符，但存在部分差异。如未列出，操作符行为默认为GaussDB原生行为，目前存在MySQL不支持但是GaussDB支持的语句，在MySQL兼容性下，这类语句通常为系统内部使用，因此不建议使用。

操作符差异

- ORDER BY排序对NULL值处理的差异。MySQL在排序时会将NULL值排序在前面；GaussDB默认将NULL值默认排在最后面。GaussDB可以通过NULLS FIRST和NULLS LAST设置NULL值排序顺序。
- 有ORDER BY时，GaussDB输出顺序与MySQL一致。没有ORDER BY时，GaussDB不保证结果有序。
- 使用MySQL操作符时需要用括号表达式严格确保表达式的结合性，否则执行报错。例如：SELECT 1 regexp ('12345' regexp '123');。

GaussDB M-Compatibility操作符不用括号严格表达的结合性也能成功执行。

- NULL值显示不同。MySQL会将NULL显示为“NULL”；GaussDB将NULL值显示为空值。

MySQL输出结果：

```
mysql> SELECT NULL;
+-----+
| NULL |
+-----+
| NULL |
+-----+
1 row in set (0.00 sec)
```

GaussDB输出结果：

```
m_db=# SELECT NULL;
?column?
-----
(1 row)
```

- 操作符执行后，列名显示不一致。MySQL会将NULL显示为“NULL”；GaussDB将NULL值显示为空值。
- 字符串转double遇到非法字符串时，告警信息不一致。MySQL在常量非法字符串报错，字段非法字符串不报错；GaussDB在常量非法字符串和字段非法字符串都报错。
- 比较操作符返回结果显示不同。MySQL返回1/0；GaussDB返回t/f。

表 3-19 操作符

MySQL数据库	GaussDB数据库	差异
<>	支持，存在差异	MySQL支持索引，GaussDB不支持索引。
<=>	支持，存在差异	MySQL支持索引，GaussDB不支持索引、hash连接和合并连接。

MySQL数据库	GaussDB数据库	差异
行表达式	支持，存在差异	- MySQL支持<=>操作符行比较、GaussDB不支持<=>操作符行比较。 - MySQL不支持行表达式与NULL比较。GaussDB支持<、<=、=、>=、>、<>操作符对行表达式与NULL值比较。 - MySQL不支持IS NULL、ISNULL对行表达式的操作。GaussDB支持。 - 操作符对于行表达式的不支持的操作，GaussDB错误信息与MySQL不一致。 GaussDB: m_db=# SELECT (1,2) <=> row(2,3); ERROR: could not determine interpretation of row comparison operator <=> LINE 1: SELECT (1,2) <=> row(2,3); ^ HINT: unsupported operator. m_db=# SELECT (1,2) < NULL; ?column? ----- (1 row) m_db=# SELECT (1,2) <> NULL; ?column? ----- (1 row) m_db=# SELECT (1, 2) IS NULL; ?column? ----- f (1 row) m_db=# SELECT ISNULL((1, 2)); ?column? ----- f (1 row) m_db=# SELECT ROW(0,0) BETWEEN ROW(1,1) AND ROW(2,2); ERROR: un support type MySQL: mysql> SELECT (1,2) <=> row(2,3); +-----+ (1,2) <=> row(2,3) +-----+ 0 +-----+ 1 row in set (0.00 sec) mysql> SELECT (1,2) < NULL; ERROR 1241 (21000): Operand should contain 2 column(s) mysql> SELECT (1,2) <> NULL; ERROR 1241 (21000): Operand should contain 2 column(s) mysql> SELECT (1, 2) IS NULL; ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT ISNULL((1, 2)); ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT NULL BETWEEN NULL AND ROW(2,2); ERROR 1241 (21000): Operand should contain 1 column(s)

MySQL数据库	GaussDB数据库	差异
--	支持，存在差异	MySQL表示对一个操作数进行两次取反，结果等于原操作数；GaussDB表示注释。
!!	支持，存在差异	MySQL：!!含义同!，表示取非。 GaussDB：!表示取非操作，当!与!中间存在空格时，表示连续两次取非(!!)；当!与!中间没有空格时，表示阶乘(!!)。 说明 - GaussDB中，当同时使用阶乘(!!)和取非(!)时，阶乘(!!)和取非(!)中间需要添加空格，否则会报错。 - GaussDB中，当需要多次取非操作时，!与!之间需使用空格隔开。
[NOT] REGEXP	支持，存在差异	- GaussDB和MySQL在正则表达式中支持的元字符有所不同。例如，GaussDB支持“\d”表示数字，“\w”表示字母、数字和下划线，“\s”表示空格，而MySQL不支持这些元字符，MySQL会把这些字符当成正常字符串。 - GaussDB中“\b”可以与“\\b”匹配，MySQL会匹配失败。 - GaussDB中使用“\”表示转义字符，而MySQL中使用“\\”。 - MySQL不支持2个操作符连在一起使用。 - 模式字符串pattern非法入参，只存在右单括号“)”时，GaussDB会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 - 在de abc匹配序列de或abc的匹配规则，当 左右存在空值时，GaussDB不会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 - 制表符“\t”正则匹配字符类[:blank:]，GaussDB可匹配，MySQL 5.7不能匹配，MySQL 8.0可匹配。 - GaussDB支持非贪婪模式匹配，即尽可能少的匹配字符，在部分特殊字符后加“?”问号字符，例如：“??, *, +?, {n}?, {n,}?, {n,m}?”。MySQL 5.7版本不支持非贪婪模式匹配，并报错：Got error 'repetition-operator operand invalid' from regexp。MySQL 8.0版本已经支持。 - 在BINARY字符集下，text类型、blob类型均会转换成bytea类型，由于REGEXP操作符不支持bytea类型，因此无法匹配。

MySQL数据库	GaussDB数据库	差异
LIKE	支持，存在差异	MySQL：LIKE的左操作数只能是位运算或者算术运算或者由括号组成的表达式，LIKE的右操作数只能是单目运算符(不含NOT)或者括号组成的表达式。 GaussDB：LIKE的左右操作数可以是任意表达式。
[NOT] BETWEEN AND	支持，存在差异	MySQL：[NOT] BETWEEN AND嵌套使用时从右到左结合。[NOT] BETWEEN AND的第1个操作数和第2个操作数只能是位运算或者算术运算或者由括号组成的表达式。 GaussDB：[NOT] BETWEEN AND嵌套使用时从左到右结合。[NOT] BETWEEN AND的第1个操作数和第2个操作数可以是任意表达式。
IN	支持，存在差异	MySQL：IN的左操作数只能是位运算或者算术运算或者由括号组成的表达式。 GaussDB：IN的左操作数可以是任意表达式。
!	支持，存在差异	MySQL：!的操作数只能是单目运算符（不含not）或者括号组成的表达式。 GaussDB：!的操作数可以是任意表达式。
#	不支持	MySQL支持#注释，GaussDB不支持#注释。
BINARY	支持，存在差异	GaussDB中支持的表达式和MySQL并不完全一致（包括一些函数、操作符等）。GaussDB独有的表达式“~”、“IS DISTINCT FROM”等，由于BINARY关键字优先级更高，使用BINARY expr会优先将BINARY与“~”、“IS DISTINCT FROM”的左参数合并，导致报错。
取反（-）	支持，存在差异	连续取反次数超过1次时，GaussDB会将其识别为注释，导致返回结果与MySQL存在差异。

MySQL数据库	GaussDB数据库	差异
XOR、 、&、<、>、<=、>=、=、!=	支持，存在差异	MySQL执行机制为执行左操作数后，对结果进行判断是否为空，进而决定是否需要执行右操作数。 GaussDB执行机制是执行左右操作数后，对结果再进行判断是否为空。 当左操作数结果为空，右操作数执行报错时，MySQL不会报错直接返回，GaussDB会执行报错。 MySQL行为： <pre>mysql> SELECT version(); +-----+ version() +-----+ 5.7.44-debug-log +-----+ 1 row in set (0.00 sec)</pre> <pre>mysql> DROP TABLE IF EXISTS data_type_table; Query OK, 0 rows affected (0.02 sec)</pre> <pre>mysql> CREATE TABLE data_type_table (-> MyBool BOOL, -> MyBinary BINARY(10), -> MyYear YEAR ->); Query OK, 0 rows affected (0.02 sec)</pre> <pre>mysql> INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021'); Query OK, 1 row affected (0.00 sec)</pre> <pre>mysql> SELECT (MyBool % MyBinary) (MyBool - MyYear) FROM data_type_table; +-----+ (MyBool % MyBinary) (MyBool - MyYear) +-----+ NULL +-----+ 1 row in set, 2 warnings (0.00 sec)</pre> GaussDB行为： <pre>m_db=# DROP TABLE IF EXISTS data_type_table; DROP TABLE m_db=# CREATE TABLE data_type_table (m_db=# MyBool BOOL, m_db=# MyBinary BINARY(10), m_db=# MyYear YEAR m_db=#); CREATE TABLE m_db=# INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021'); INSERT 0 1 m_db=# SELECT (MyBool % MyBinary) (MyBool - MyYear) FROM data_type_table; WARNING: Truncated incorrect double value: '4Vx ' CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear) WARNING: division by zero CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear) ERROR: Bigint is out of range. CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear)</pre>

表 3-20 操作符组合差异

操作符组合示例	MySQL数据库	GaussDB数据库	说明
SELECT 1 LIKE 3 & 1;	不支持	支持	LIKE的右操作数不能是位运算符组成的表达式。
SELECT 1 LIKE 1 +1;	不支持	支持	LIKE的右操作数不能是算术运算符组成的表达式。
SELECT 1 LIKE NOT 0;	不支持	支持	LIKE的右操作数只能是+、-、!等单目操作符或者括号组成的表达式，NOT除外。
SELECT 1 BETWEEN 1 AND 2 BETWEEN 2 AND 3;	从右到左结合	从左到右结合	建议加上括号明确运算的优先级，防止由于运算顺序的差异导致运算结果产生偏差。
SELECT 2 BETWEEN 1=1 AND 3;	不支持	支持	BETWEEN的第2个操作数不能是比较操作符组成的表达式。
SELECT 0 LIKE 0 BETWEEN 1 AND 2;	不支持	支持	BETWEEN的第1个操作数不能是模式匹配操作符组成的表达式。
SELECT 1 IN (1) BETWEEN 0 AND 3;	不支持	支持	BETWEEN的第1个操作数不能是IN操作符组成的表达式。
SELECT 1 IN (1) IN (1);	不支持	支持	第2个IN表达式左操作数不能是IN组成的表达式。
SELECT ! NOT 1;	不支持	支持	!的操作数只能是+、-、!等单目操作符或者括号组成的表达式，NOT除外。

索引差异

- GaussDB当前仅支持UBTree和B-tree索引。
- 针对模糊匹配（LIKE操作符），MySQL创建默认索引可以走索引；GaussDB默认的索引不走索引，需要用户使用以下语法指定opclass，比如指定为text_pattern_ops，LIKE操作符才可以走索引。

CREATE INDEX indexname ON tablename(col [opclass]);
- B-tree/UBTree索引场景保持原生GaussDB原有逻辑，即同一操作符族内的类型比较，支持索引扫描，其余索引类型暂未支持。
- 在使用GaussDB JDBC连接数据库时，GaussDB的YEAR类型在含有绑定参数的PBE场景下无法利用索引。
- WHERE子句中，索引字段类型和常量类型操作场景下，GaussDB中索引与MySQL索引支持存在差异，如表3-21所示。例如以下语句GaussDB不支持索引：

CREATE TABLE t(_int int);
CREATE INDEX idx ON t(_int) USING BTREE;
SELECT * FROM t WHERE _int > 2.0;

说明

WHERE子句里索引字段类型和常量类型操作场景中，可以使用cast函数将常数类型显式转换为字段类型，以便实现索引。

```
SELECT * FROM t WHERE _int > cast(2.0 AS signed);
```

表 3-21 索引支持存在差异

索引字段类型	常量类型	GaussDB是否支持	MySQL是否支持
整型	整型	是	是
浮点型	浮点型	是	是
定点型	定点型	是	是
字符串类型	字符串类型	是	是
二进制类型	二进制类型	是	是
带日期的时间类型	带日期的时间类型	是	是
TIME类型	TIME类型	是	是
带日期的时间类型	可转为带日期的时间类型（如20231130等整型）	是	是
带日期的时间类型	TIME类型	是	是
TIME类型	可转为TIME类型的常量（如203008等整型）	是	是
浮点型	整型	是	是
浮点型	定点型	是	是
浮点型	字符串类型	是	是
浮点型	二进制类型	是	是
浮点型	带日期的时间类型	是	是
浮点型	TIME类型	是	是
定点型	整型	是	是
字符串类型	带日期的时间类型	是	否
字符串类型	TIME类型	是	否
二进制类型	字符串类型	是	是

索引字段类型	常量类型	GaussDB是否支持	MySQL是否支持
二进制类型	带日期的时间类型	是	否
二进制类型	TIME类型	是	否
整型	浮点型	否	是
整型	定点型	否	是
整型	字符串类型	否	是
整型	二进制类型	否	是
整型	带日期的时间类型	否	是
整型	TIME类型	否	是
定点型	浮点型	否	是
定点型	字符串类型	否	是
定点型	二进制类型	否	是
定点型	带日期的时间类型	否	是
定点型	TIME类型	否	是
字符串类型	二进制类型	否	是
带日期的时间类型	整型（不可转为带日期的时间类型）	否	是
带日期的时间类型	浮点型（不可转为带日期的时间类型）	否	是
带日期的时间类型	定点型（不可转为带日期的时间类型）	否	是
TIME类型	整型（不可转为TIME类型）	否	是
TIME类型	字符串类型（不可转为TIME类型）	否	是
TIME类型	二进制类型（不可转为TIME类型）	否	是
TIME类型	带日期的时间类型	否	是

索引字段类型	常量类型	GaussDB是否支持	MySQL是否支持
YEAR类型	YEAR类型	是	是
YEAR类型	可转为YEAR类型的常量（如2034等整型）	是	是
BIT类型	BIT类型	否	是

表 3-22 是否支持走索引

索引字段类型	常量类型	GaussDB是否走索引	MySQL是否走索引
字符串类型	整型	否	否
字符串类型	浮点型	否	否
字符串类型	定点型	否	否
二进制类型	整型	否	否
二进制类型	浮点型	否	否
二进制类型	定点型	否	否
带日期的时间类型	字符串类型（不可转为带日期的时间类型）	否	否
带日期的时间类型	二进制类型（不可转为带日期的时间类型）	否	否
TIME类型	浮点型（不可转为TIME类型）	否	否
TIME类型	定点型（不可转为TIME类型）	否	否
BIT类型	字符串类型	否	否

3.2.4 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集，支持的范围如下。

表 3-23 字符集列表

MySQL数据库	GaussDB数据库
utf8mb4	支持

MySQL数据库	GaussDB数据库
utf8mb4_0900_ai_ci	支持
utf8_general_ci	支持
utf8_bin	支持
utf8_unicode_ci	支持

3.2.6 事务

GaussDB数据库兼容MySQL的事务，但存在部分差异。本章节介绍GaussDB的M-Compatibility数据库中事务相关的差异。

事务默认隔离级别

M-Compatibility默认隔离级别为READ COMMITTED，MySQL默认隔离级别为REPEATABLE READ。

```
-- 查看当前事务隔离级别。  
m_db=# SHOW transaction_isolation;
```

子事务

M-Compatibility中，通过SAVEPOINT用于在当前事务里建立一个新的保存点（子事务），使用ROLLBACK TO SAVEPOINT回滚到一个保存点（子事务），子事务回滚后父事务可以继续运行，子事务的回滚不影响父事务的事务状态。

MySQL不存在创建保存点（子事务）。

嵌套事务

嵌套事务指在事务块中开启新事务。

M-Compatibility中，正常事务块中开启新事务会警告存在一个进行中的事务，忽略开启命令；异常事务块中开启新事务将报错，必须在执行ROLLBACK/COMMIT之后才可以执行，执行ROLLBACK/COMMIT会回滚之前语句。

MySQL中，正常事务块中开启新事务会先把之前事务提交，然后开启新事务；异常事务块中开启新事务会忽略错误，提交之前无错误的语句并开启新事务。

```
-- M-Compatibility正常事务块中，开启新事务会警告并忽略。  
m_db=# DROP TABLE IF EXISTS test_t;  
m_db=# CREATE TABLE test_t(a int, b int);  
m_db=# BEGIN;  
m_db=# INSERT INTO test_t values(1, 2);  
m_db=# BEGIN; -- 会警告there is already a transaction in progress。  
m_db=# SELECT * FROM test_t ORDER BY 1;  
m_db=# COMMIT;  
  
-- M-Compatibility异常事务块中，开启新事务会报错，必须ROLLBACK/COMMIT之后才可以执行。  
m_db=# BEGIN;  
m_db=# ERROR sql; -- 错误语句。  
m_db=# BEGIN; -- 报错。  
m_db=# COMMIT; -- ROLLBACK/COMMIT之后才可以执行。
```

隐式提交的语句

M-Compatibility使用GaussDB存储，继承GaussDB事务机制，事务中执行DDL、DCL不会自动提交。

MySQL在DDL、DCL、管理类语句，锁相关语句会自动提交。

```
-- M-Compatibility创建表和设置GUC参数可以回滚掉。  
m_db=# DROP TABLE IF EXISTS test_table_rollback;  
m_db=# BEGIN;  
m_db=# CREATE TABLE test_table_rollback(a int, b int);  
m_db=# \d test_table_rollback;  
m_db=# ROLLBACK;  
m_db=# \d test_table_rollback; -- 不存在该表。
```

SET TRANSACTION 差异

M-Compatibility中，SET TRANSACTION同时设置多次隔离级别/事务访问模式时，只有最后一个会生效；多个事务特性支持使用空格和逗号分隔。

MySQL中SET TRANSACTION不允许设置多次隔离级别/事务访问模式；多个事务特性只支持使用逗号分隔。

表 3-25 SET TRANSACTION 差异

语法	功能	差异
SET TRANSACTION	设置事务特性。	M-Compatibility中，不开启m_format_dev_version='s2'参数时，SET TRANSACTION在会话级别生效，功能与SET SESSION TRANSACTION一致；开启m_format_dev_version='s2'参数时，SET TRANSACTION是设置下一个事务特性；MySQL中SET TRANSACTION在下一个事务生效。
SET SESSION TRANSACTION	设置会话级事务特性。	-
SET GLOBAL TRANSACTION	设置全局会话级事务特性，该特性适用于后续会话，对当前会话无影响。	M-Compatibility中，GLOBAL是全局会话级别生效，只针对当前数据库实例，其它数据库不影响。 MySQL中，会使所有数据库生效。

```
-- SET TRANSACTION会话级生效。  
m_db=# SET TRANSACTION ISOLATION LEVEL READ COMMITTED READ WRITE;  
m_db=# SHOW transaction_isolation;  
m_db=# SHOW transaction_read_only;  
-- M-Compatibility同时设置多次隔离级别/事务访问模式，最后一个生效。  
m_db=# SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED, ISOLATION LEVEL REPEATABLE READ, READ WRITE, READ ONLY;  
m_db=# SHOW transaction_isolation; -- repeatable read  
m_db=# SHOW transaction_read_only; -- on
```

START TRANSACTION 差异

M-Compatibility中，START TRANSACTION开启事务时，同时支持设置隔离级别；同时设置多次隔离级别/事务访问模式时，只有最后一个会生效；当前版本不支持立即开启一致性快照；多个事务特性支持空格和逗号分隔。

MySQL的start transaction 开启事务时，不支持设置隔离级别，不支持设置多次事务访问模式；多个事务特性只支持逗号分隔。

```
-- 开启事务设置隔离级别。  
m_db=# START TRANSACTION ISOLATION LEVEL READ COMMITTED;  
m_db=# COMMIT;  
-- 多次设置访问模式。  
m_db=# START TRANSACTION READ ONLY, READ WRITE;  
m_db=# COMMIT;
```

事务相关的 GUC 参数

表 3-26 事务相关的 GUC 参数差异

GUC参数	功能	差异
autocommit	设置事务自动提交模式。	-
transaction_isolation	在M-Compatibility中是设置当前事务的隔离级别。 在MySQL中是设置会话级事务的隔离级别。	- GaussDB中，通过使用SET transaction_isolation = value命令，只能改变当前事务的隔离级别。如果想要改变会话级的隔离级别，可以使用default_transaction_isolation。在MySQL中，通过使用SET命令，可以改变会话级的事务隔离级别。 - 支持范围差异。 MySQL中，支持以下隔离级别设置，对大小写不敏感，对空格敏感： - READ-COMMITTED - READ-UNCOMMITTED - REPEATABLE READ - SERIALIZABLE GaussDB中，支持以下隔离级别设置，对大小写和空格敏感： - read committed - read uncommitted - repeatable read - serializable - default（设置和会话中默认隔离级别一样） - 设置m_format_dev_version = 's2'时，支持MySQL的隔离级别设置。 - 在GaussDB中，新事务的transaction_isolation值将被初始化为default_transaction_isolation的值。

GUC参数	功能	差异
tx_isolation	设置事务的隔离级别； tx_isolation 和 transaction_isolation是同义词。	M-Compatibility中不支持查询，不支持修改，建议直接查询transaction_isolation。
default_transaction_isolation	设置事务的隔离级别。	M-Compatibility中通过SET设置会改变会话级事务隔离级别。 MySQL中不支持该系统参数。
transaction_read_only	设置事务的访问模式。	- 在M-Compatibility中，通过使用SET命令，只能改变当前事务的访问模式。如果想要改变会话级的访问模式，可以使用default_transaction_read_only。 - 在MySQL中，通过使用SET命令，可以改变会话级的事务隔离级别。 - 在GaussDB中，新事务的transaction_read_only值将被初始化为default_transaction_read_only的值。
tx_read_only	设置事务的访问模式。 tx_read_only 和 transaction_read_only是同义词。	M-Compatibility中不支持查询，不支持修改，建议直接查询transaction_read_only。
default_transaction_read_only	设置事务的访问模式。	M-Compatibility中通过SET设置会改变会话级事务访问模式；MySQL中不支持该系统参数。

3.2.7 SQL

3.2.7.1 关键字

约束差异：

- 当关键字在M-Compatibility模式下为保留关键字，在MySQL中为非保留关键字，其差异为：在M-Compatibility模式下不可作为表名、列名、列别名、AS列别名、AS表别名、表别名、函数名和变量名，在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字，在MySQL中为保留关键字，其差异为：在M-Compatibility模式下可作为表名、列名、列别名、AS列别名、AS表别名、表别名、函数名和变量名，在MySQL中不支持。

- 当关键字在M-Compatibility模式下为保留关键字（可以是函数或类型），在MySQL中为保留关键字，其差异为：在M-Compatibility模式下可作为列别名、AS列别名、函数名和变量名，在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字（可以是函数或类型），在MySQL中为非保留关键字，其差异为：在M-Compatibility模式下不可作为表名、列名、AS表别名和表别名，在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字（不能是函数或类型），在MySQL中为保留关键字，其差异为：在M-Compatibility模式下可作为表名、列名、列别名、AS列别名、AS表别名、表别名和变量名，在MySQL中不支持。
- 当关键字在M-Compatibility模式下为非保留关键字（不能是函数或类型），在MySQL中为非保留关键字，其差异为：在M-Compatibility模式下不可作为函数名，在MySQL中支持。

📖 说明

在M-Compatibility模式下的非保留关键字、保留关键字（可以是函数或类型）以及非保留关键字（不能是函数或类型）之中，以下关键字不能作为列别名进行使用：

BETWEEN、BIGINT、BLOB、CHAR、CHARACTER、CROSS、DEC、DECIMAL、DIV、DOUBLE、EXISTS、FLOAT、FLOAT4、FLOAT8、GROUPING、INNER、INOUT、INT、INT1、INT2、INT3、INT4、INT8、INTEGER、JOIN、LEFT、LIKE、LONGBLOB、LONGTEXT、MEDIUMBLOB、MEDIUMINT、MEDIUMTEXT、MOD、NATURAL、NUMERIC、OUT、OUTER、PRECISION、REAL、RIGHT、ROW、ROW_NUMBER、SIGNED、SMALLINT、SOUNDS、TINYBLOB、TINYINT、TINYTEXT、VALUES、VARCHAR、VARYING、WITHOUT

其中，SIGNED和WITHOUT在MySQL中可以作为列别名进行使用。

3.2.7.2 标识符

M-Compatibility模式下标识符存在以下差异：

- GaussDB无引号标识符中不支持以美元符号（\$）开头，MySQL无引号标识符中支持。
- GaussDB无引号标识符中的支持大小写敏感的数据库对象。
- GaussDB标识符支持U+0080~U+00FF扩展字符，MySQL标识符支持U+0080~U+FFFF的扩展字符。
- 无引号标识符中，GaussDB不支持创建以数字开头包含一个e或E结尾作为标识符的表，例如：

```
-- GaussDB报错不支持，MySQL支持
m_db=# CREATE TABLE 23e(c1 int);
ERROR: syntax error at or near "23"
LINE 1: CREATE TABLE 23e(c1 int);
                        ^
m_db=# CREATE TABLE t1(23E int);
ERROR: syntax error at or near "23"
LINE 1: CREATE TABLE t1(23E int);
                        ^
```

- 有引号标识符中，GaussDB对于创建了列名为纯数字或科学计算法的表，不支持直接使用，需要在引号中使用；对于点操作符（.）场景，列名为纯数字或科学计算法的表也需要在引号中使用。例如：

```
-- 创建列名为纯数字或科学计算法的表
m_db=# CREATE TABLE t1(`123` int, `1e3` int, `1e` int);
CREATE TABLE

-- 向表中插入数据
m_db=# INSERT INTO t1 VALUES(7, 8, 9);
INSERT 0 1
```

```
-- 结果非预期，但与MySQL结果一致
m_db=# SELECT 123 FROM t1;
?column?
-----
    123
(1 row)

-- 结果非预期，但与MySQL结果一致
m_db=# SELECT 1e3 FROM t1;
?column?
-----
    1000
(1 row)

-- 结果非预期，并且与MySQL结果不一致
m_db=# SELECT 1e FROM t1;
e
---
    1
(1 row)

-- 正确用法
m_db=# SELECT `123` FROM t1;
123
-----
    7
(1 row)

m_db=# SELECT `1e3` FROM t1;
1e3
-----
    8
(1 row)

m_db=# SELECT `1e` FROM t1;
1e
-----
    9
(1 row)

-- 点操作符的场景，GaussDB不支持，MySQL支持
m_db=# SELECT t1.123 FROM t1;
ERROR: syntax error at or near ".123"
LINE 1: SELECT t1.123 FROM t1;
              ^

m_db=# SELECT t1.1e3 FROM t1;
ERROR: syntax error at or near "1e3"
LINE 1: SELECT t1.1e3 FROM t1;
              ^

m_db=# SELECT t1.1e FROM t1;
ERROR: syntax error at or near "1"
LINE 1: SELECT t1.1e FROM t1;
              ^

-- 点操作符的场景，正确用法：
m_db=# SELECT t1.`123` FROM t1;
123
-----
    7
(1 row)

m_db=# SELECT t1.`1e3` FROM t1;
1e3
-----
    8
(1 row)

m_db=# SELECT t1.`1e` FROM t1;
1e
```

```
-----
          9
      (1 row)

m_db=# DROP TABLE t1;
DROP TABLE
```

- GaussDB分区名使用双引号（需要设置SQL_MODE为ANSI_QUOTES）或反引号时区分大小写，MySQL不区分。
- MySQL标识符长度限制为64字符，而GaussDB标识符长度限制为63字节。超过标识符的长度限制后，MySQL报错，GaussDB会对标识符截断并告警。

3.2.7.3 DDL

表 3-27 DDL 语法兼容介绍

概述	详细语法说明	差异
建表和修改表时支持创建主键、UNIQUE索引、外键约束	ALTER TABLE、CREATE TABLE	• 在GaussDB中，当约束关联的表为ustore，且SQL语句中指定为using btree时，底层会建立为ubtree。 • 在GaussDB中，允许将外键作为分区键。 • 索引名、约束名、key名 GaussDB是SCHEMA下唯一，MySQL是表下唯一。

概述	详细语法说明	差异
支持自增列	ALTER TABLE、CREATE TABLE	- GaussDB的自动增长列建议为索引的第一个字段，否则建表时产生警告，MySQL自动增长列必须为索引第一个字段，否则建表时会报错。GaussDB含有自动增长列的表进行某些操作时会产生错误，例如：ALTER TABLE EXCHANGE PARTITION。 - GaussDB的 AUTO_INCREMENT = value语法，value必须为小于2^{127}的正数。MySQL可以为0，GaussDB不可以。 - GaussDB中当自增值已经达到字段数据类型的最大值时，继续自增将产生错误。MySQL有些场景产生错误或警告，有些场景仍自增为最大值。 - 不支持 innodb_autoinc_lock_mode系统变量，GaussDB的 GUC参数 auto_increment_cache=0 时，批量插入自动增长列的行为与MySQL系统变量 innodb_autoinc_lock_mode=1相似。 - GaussDB的自动增长列在导入数据或者进行Batch Insert执行计划的插入操作时，对于混合0、NULL和确定值的场景，如果产生错误，后续插入自增值不一定与MySQL完全一致。可以使用GUC参数 auto_increment_cache控制预留自增值的数量。 - GaussDB的并行导入或插入自动增长列触发自增时，每个并行线程预留的缓存值也只在其线程中使用，未完全使用完毕的话，也会出现表中自动增长列的值不连续的情况。并行插入产生的自增值结果无法保证与MySQL完全一致。

概述	详细语法说明	差异
		• GaussDB的本地临时表中的自动增长列批量插入时不会预留自增值，正常场景不会产生不连续的自增值。MySQL临时表与普通表中的自动增长列自增结果一致。 • GaussDB的SERIAL数据类型为原有的自增列，与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。 • GaussDB不允许auto_increment_offset的值大于auto_increment_increment的值，会产生错误。MySQL允许，并说明auto_increment_offset会被忽略。 • 在表有主键或索引的情况下，ALTER TABLE命令重写表数据的顺序与MySQL不一定相同，GaussDB按表数据存储顺序重写，MySQL会按主键或索引顺序重写，导致自增值的顺序可能不同。 • GaussDB使用ALTER TABLE命令添加或修改自增列时，第一次预留自增值的数量是表统计信息中的行数，统计信息的行数不一定与MySQL一致。 • GaussDB的last_insert_id函数返回值为128位的整型。 • GaussDB在触发器或用户自定义函数中自增时，刷新last_insert_id返回值。MySQL不刷新。 • GaussDB的对GUC参数auto_increment_offset和auto_increment_increment设置超出范围的值会产生错误。MySQL会自动改为边界值。

概述	详细语法说明	差异
支持前缀索引	CREATE INDEX、ALTER TABLE、CREATE TABLE	- GaussDB中前缀长度不得超过2676，键值的实际长度受内部页面限制，若字段中含有多字节字符或者一个索引上有多个键，索引行长度可能会超限报错。 - GaussDB中主键索引中不支持前缀键，创建或添加主键时不支持指定前缀长度。
支持指定字符集与排序规则	ALTER SCHEMA、ALTER TABLE、CREATE SCHEMA、CREATE TABLE	- 指定库级字符集时，除BINARY字符集外，暂不支持创建新库/模式的字符集与数据库的server_encoding不同。 - 指定表级、列级字符集和字符序时，MySQL支持指定与库级字符集、字符序不同的字符集和字符序。在GaussDB中，表级、列级字符集和字符序仅支持BINARY字符集、字符序或者与库级字符集、字符序相同的字符集、字符序。
修改表时支持在表第一列前面或者在指定列后面添加列	ALTER TABLE	-
修改列名称/定义语法	ALTER TABLE	暂不支持DROP INDEX DROP KEY ORDER BY子项。

概述	详细语法说明	差异
创建分区表语法	CREATE TABLE PARTITION	- MySQL在以下场景支持表达式，不支持多个分区键： - 使用LIST分区/RANGE分区策略，不指定COLUMNS关键字。 - 使用HASH分区策略。 - MySQL在以下场景不支持表达式，支持多个分区键： - 使用LIST分区/RANGE分区策略，指定COLUMNS关键字。 - 使用KEY分区策略。 - GaussDB不支持使用表达式作为分区键，不支持指定二级分区。 - GaussDB仅在以下场景支持使用多个分区键：使用LIST分区/RANGE分区策略。 - GaussDB在使用LIST分区时，需要保证list_value都是对应分区键类型的合法有效值，如超出有效值范围则会导致创建失败，MySQL允许无效值创建成功。 - GaussDB分区表不支持用生成列作为分区键。
建表和修改表时支持指定表级和列级comment	CREATE TABLE、ALTER TABLE	-
创建索引时支持指定索引级comment	CREATE INDEX	-

概述	详细语法说明	差异
交换普通表和分区表分区的数据	ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点： - 对于自增列，MySQL执行ALTER TABLE EXCHANGE PARTITION后，自增列会被重置；GaussDB则不会被重置，自增列则按照旧的自增值递增。 - MySQL表或分区使用tablespace时，则无法进行分区和普通表数据的交换；GaussDB表或分区使用不同的tablespace时，仍可进行分区和普通表数据的交换。 - 对于列默认值，MySQL不会校验默认值，因此默认值不同时也可进行分区和普通表数据的交换；GaussDB会校验默认值，如果默认值不同，则无法进行分区和普通表数据的交换。 - MySQL在分区表或普通表上进行DROP列操作后，表结构仍然一致，则可进行分区和普通表数据的交换；GaussDB需要保证普通表和分区表的被删除列严格对齐才能进行分区和普通表数据的交换。 - MySQL和GaussDB的哈希算法不同，所以两者在相同的hash分区存储的数据可能不一致，导致最后交换的数据也可能不一致。 - MySQL的分区表不支持外键，普通表包含外键或其他表引用普通表的外键，则无法进行分区和普通表数据的交换；GaussDB的分区表支持外键，在两个表的外键约束一致时，则可进行分区和普通表数据的交换，GaussDB的分区表不带外键，普通表有其他表引用，如果分区表和普通表表一致，则可进行分区和普通表数据的交换。

概述	详细语法说明	差异
修改分区表的分区键信息	ALTER TABLE	MySQL支持修改分区表的分区键信息，GaussDB中不支持。
支持CREATE TABLE ... LIKE语法	CREATE TABLE ... LIKE	- 在MySQL 8.0.16 之前的版本中，CHECK约束会被语法解析但功能会被忽略，表现为不复制CHECK约束，GaussDB支持复制CHECK约束。 - 对于主键约束名称，在建表时，MySQL所有主键约束名称固定为PRIMARY KEY，GaussDB不支持复制。 - 对于唯一键约束名称，在建表时，MySQL支持复制，GaussDB不支持复制。 - 对于CHECK约束名称，在建表时，MySQL 8.0.16 之前的版本无CHECK约束信息，GaussDB支持复制。 - 对于索引名称，在建表时，MySQL支持复制，GaussDB不支持复制。 - 在跨sql_mode模式建表时，MySQL受宽松模式和严格模式控制，GaussDB可能存在严格模式失效的情况。 例如：源表存在默认值“0000-00-00”，在“no_zero_date”严格模式下，GaussDB建表成功，且包含默认值“0000-00-00”，严格模式失效；而MySQL建表失败，受严格模式控制。
TRUNCATE子分区语法	ALTER TABLE [IF EXISTS] table_name truncate_clause;	支持子项有差异，对于truncate_clause： - M-Compatibility模式： TRUNCATE PARTITION { { ALL partition_name [, ...] } FOR (partition_value [, ...]) } [UPDATE GLOBAL INDEX] - MySQL支持： TRUNCATE PARTITION {partition_names ALL}

概述	详细语法说明	差异
主键索引名	CREATE TABLE table_name (col_definitine ,PRIMARY KEY [index_name] [USING method] ({ column_name (expression) }[ASC DESC] } [, ...]) index_parameters [USING method COMMENT 'string']	GaussDB中的主键指定索引名后创建的索引名为用户所指定的索引名，MySQL索引名为PRIMARY。
删除有依赖的对象	DROP drop_type name CASCADE;	在GaussDB中，删除有依赖的对象需要加CASCADE，MySQL不需要。
NOT NULL约束不允许插入NULL值	CREATE TABLE t1(id int NOT NULL DEFAULT 8); INSERT INTO t1 VALUES(NULL); INSERT INTO t1 VALUES(1), (NULL),(2);	在MySQL宽松模式下，会将NULL进行类型转换，并成功插入数据；在MySQL严格模式下不允许插入NULL值。在GaussDB不支持此特性，在宽松模式和严格模式下均不允许插入NULL值。
CHECK约束生效	CREATE TABLE	- CREATE TABLE带CHECK约束的时候，MySQL 8.0会生效，MySQL 5.7只解析语法但不生效。GaussDB在此功能上同步MySQL 8.0版本，且GaussDB CHECK约束可以引用其他列，而MySQL不能。 - GaussDB一个表中最多只能加32767个CHECK约束。
索引的algorithm和lock选项不起作用	CREATE INDEX ... DROP INDEX ...	M-Compatibility模式的CREATE/DROP INDEX语句中INDEX选项algorithm_option和lock_option目前只在语法上支持，创建时不报错，但实际不起作用。
CREATE TABLE hash分区和二级分区的存储与MySQL不同	CREATE TABLE	GaussDB的CREATE TABLE语句中hash分区表和二级分区表所使用的hash函数与MySQL不一致，因此hash分区表和二级分区表的存储与MySQL有区别。

概述	详细语法说明	差异
分区表索引	CREATE INDEX	- GaussDB的分区表索引分为LOCAL和GLOBAL两种。LOCAL索引与某个具体分区绑定，而GLOBAL索引则对应整个分区表。 - LOCAL和GLOBAL索引的创建方法和默认规则具体说明参见《开发指南》中“SQL语法 > SQL语句 > C > CREATE INDEX”章节，例如：在非分区键上创建唯一索引，会默认创建为GLOBAL索引。 - MySQL无GLOBAL索引的概念。在GaussDB中，当分区表索引为GLOBAL索引时，对表分区进行DROP、TRUNCATE、EXCHANGE等操作不会默认更新GLOBAL索引，进而导致GLOBAL索引失效，导致后续语句无法选中该索引。为了避免这种场景，建议用户在使用分区操作语法时在最后显指定UPDATE GLOBAL INDEX子句，或配置全局GUC参数enable_gpi_auto_update为true（推荐），使得在进行分区操作时自动更新GLOBAL索引。
CREATE/ALTER TABLE语句中分区表为KEY分区，不支持指定algorithm。部分分区定义入参不支持表达式。	CREATE TABLE、ALTER TABLE	GaussDB的CREATE/ALTER TABLE语句中分区表为KEY分区，不支持指定algorithm。不支持表达式入参的语法： - PARTITION BY HASH() - PARTITION BY KEY() - VALUES LESS THAN()
分区表不支持LINEAR/KEY hash	CREATE TABLE ... PARTITION ...	GaussDB分区表不支持LINEAR/KEY hash。
check和auto_increment语法不能作用在同一字段	CREATE TABLE	由于MySQL 5.7的check字段不生效，check和auto_increment同时作用于同一字段只有auto_increment生效，但GaussDB报错。

概述	详细语法说明	差异
删除存在依赖关系的表	DROP TABLE	GaussDB删除存在依赖的表必须加上CASCADE才能成功，MySQL不需要。
表定义相关选项	CREATE TABLE... 、 ALTER TABLE ...	- GaussDB不支持以下选项： AVG_ROW_LENGTH、 CHECKSUM、 COMPRESSION、 CONNECTION、DATA DIRECTORY、INDEX DIRECTORY、 DELAY_KEY_WRITE、 ENCRYPTION、 INSERT_METHOD、 KEY_BLOCK_SIZE、 MAX_ROWS、 MIN_ROWS、 PACK_KEYS、 PASSWORD、 STATS_AUTO_RECALC、 STATS_PERSISTENT、 STATS_SAMPLE_PAGES。 - 以下选项在GaussDB中不报错，但实际上也不生效： ENGINE、 ROW_FORMAT。
CMK密钥轮转，轮换加密COLUMN ENCRYPTION KEY的CLIENT MASTER KEY，对COLUMN ENCRYPTION KEY明文进行重加密。	ALTER COLUMN ENCRYPTION KEY	M-Compatibility模式不支持全密态。因此不支持该语法。
密态等值查询特性使用多级加密模型，主密钥加密列密钥，列密钥加密数据。本语法用于创建主密钥对象。	CREATE CLIENT MASTER KEY	M-Compatibility模式不支持全密态。因此不支持该语法。
创建一个列加密密钥，该密钥可用于加密表中的指定列。	CREATE COLUMN ENCRYPTION KEY	M-Compatibility模式不支持全密态。因此不支持该语法。
全密态功能，传输密钥到服务端缓存，只在开启内存解密逃生通道的情况下使用。	\send_token	M-Compatibility模式不支持全密态。因此不支持该语法。

概述	详细语法说明	差异
全密态功能，传输密钥到服务端缓存，只在开启内存解密逃生通道的情况下使用。	\st	M-Compatibility模式不支持全密态。因此不支持该语法。
全密态功能，销毁服务端缓存的密钥，只在开启内存解密逃生通道的情况下使用。	\clear_token	M-Compatibility模式不支持全密态。因此不支持该语法。
全密态功能，销毁服务端缓存的密钥，只在开启内存解密逃生通道的情况下使用。	\ct	M-Compatibility模式不支持全密态。因此不支持该语法。
在全密态数据库特性中,用于设置访问外部密钥管理者的参数。	\key_info KEY_INFO	M-Compatibility模式不支持全密态。因此不支持该语法。
全密态功能，用于开启三方动态库功能与加载三方动态库时的参数设置。	\crypto_module_info MODULE_INFO	M-Compatibility模式不支持全密态。因此不支持该语法。
全密态功能，用于开启三方动态库功能与加载三方动态库时的参数设置。	\cmi MODULE_INFO	M-Compatibility模式不支持全密态。因此不支持该语法。
generated always as 语句不能再引用由 generated always as 生成的列。	Generated Always AS	GaussDB的generated always as语句不能再引用由generated always as生成的列，MySQL 可以。
支持更改表名语法	ALTER TABLE tbl_name RENAME [TO AS =] new_tbl_name;	- GaussDB的ALTER RENAME 语法仅支持修改表名称功能操作，不能耦合其它功能操作。 - GaussDB中，仅旧表名字段支持使用 schema.table_name格式，且新表名与旧表名将属于同一个Schema。 - GaussDB不支持新旧表跨 Schema重命名操作；但如有权限，则可在当前 Schema下修改其他Schema下表名称。
禁用GUC参数 enable_expr_fusion	SET enable_expr_fusion= ON	M-Compatibility模式暂不支持 GUC参数enable_expr_fusion 打开。

概述	详细语法说明	差异
支持CREATE VIEW AS SELECT语法	CREATE VIEW table_name AS query;	- 针对以下类型，不支持 CREATE VIEW view_name AS query语法中query包含计算操作（如函数调用、使用操作符计算）。 - BINARY[(n)] - BOOLEAN/BOOL - VARBINARY(n) - CHAR[(n)] - VARCHAR(n) - TIME[(p)] - DATETIME[(p)] - TIMESTAMP[(p)] - BIT[(n)] - NUMERIC[(p[,s])] - DECIMAL[(p[,s])] - DEC[(p[,s])] - FIXED[(p[,s])] - FLOAT4[(p, s)] - FLOAT8[(p,s)] - FLOAT[(p)] - REAL[(p, s)] - FLOAT[(p, s)] - DOUBLE[(p,s)] - DOUBLE PRECISION[(p,s)] - TEXT - TINYTEXT - MEDIUMTEXT - LONGTEXT - BLOB - TINYBLOB - MEDIUMBLOB - LOBLOB - 在query为简单查询场景下，M-Compatibility模式针对上述类型的计算操作进行报错提示，例如： <pre>m_db=# CREATE TABLE TEST (salary int(10)); CREATE TABLE</pre>

概述	详细语法说明	差异
		m_db=# INSERT INTO TEST VALUES(8000); INSERT 0 1 m_db=# CREATE VIEW view1 AS SELECT salary/10 as te FROM TEST; ERROR: Unsupported type numeric used with expression in CREATE VIEW statement. m_db=# CREATE VIEW view2 AS SELECT sec_to_time(salary) as te FROM TEST; ERROR: Unsupported type time used with expression in CREATE VIEW statement. 在query为复合查询，子查询等非简单查询场景下，M-Compatibility模式针对上述类型的计算操作与MySQL存在差异，M-Compatibility模式下新创建表的数据类型列精度属性不保留。
索引名可重名范围	CREATE TABLE、 CREATE INDEX	MySQL中索引名在一个表下唯一，在不同的表下可以有相同的索引名。M-Compatibility模式中的索引名在同一个SCHEMA下唯一，在同一的SCHEMA下不可用相同的索引名。在M-Compatibility模式下，针对会自动创建索引的约束和key，也会有相同的规则。

概述	详细语法说明	差异
视图依赖差异	CREATE VIEW、ALTER TABLE	MySQL中视图存储，只记录目标表的表名、列名、数据库名信息，不记录目标表的唯一标识；GaussDB会将创建视图时的SQL解析，并存储目标表的唯一标识。因此存在如下差异： - 修改存在视图依赖的列的数据类型，MySQL中对应的视图不感知目标表的修改，因此可以修改成功；GaussDB视图中的列禁止修改数据类型，因此无法修改该列的数据类型。 - 重命名存在视图依赖的列，MySQL中对应的视图不感知目标表的修改，因此可以修改成功，但是后续无法查询该视图；GaussDB视图中，每个列精确存储了其对应的表和列的唯一标识，因此表中的列名可以修改成功，视图中的列名不被修改，且后续可以查询该视图。
外键差异	CREATE TABLE	- GaussDB外键约束对类型不敏感，如果主表和从表对应的字段数据类型存在隐式类型转换就可以建成。MySQL外键类型敏感。如果两个表对应的列类型不同外键无法建成。 - MySQL不支持通过MODIFY COLUMN或CHANGE COLUMN方式修改表列外键所在列的数据类型或列名等，GaussDB可以。
索引升降序差异	CREATE INDEX	在MySQL 5.7中，ASC DESC被解析但是被忽略，默认行为为ASC；在MySQL 8.0及GaussDB中，ASC DESC被解析且生效。

概述	详细语法说明	差异
指定字段的默认值	CREATE TABLE、ALTER TABLE	- MySQL 5.7指定默认值时，仅支持不带括号形式的默认值。MySQL 8.0与GaussDB支持带括号形式的默认值。 -- GaussDB m_db=# DROP TABLE IF EXISTS t1, t2; DROP TABLE m_db=# CREATE TABLE t1(a DATETIME DEFAULT NOW()); CREATE TABLE m_db=# CREATE TABLE t2(a DATETIME DEFAULT (NOW())); CREATE TABLE -- MySQL 5.7 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.04 sec) mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW()); Query OK, 0 rows affected (0.04 sec) mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW())); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(NOW())' at line 1 -- MySQL 8.0 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.17 sec) mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW()); Query OK, 0 rows affected (0.19 sec) mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW())); Query OK, 0 rows affected (0.20 sec) - MySQL给BLOB、TEXT、JSON数据类型指定默认值时必须给默认值添加括号，GaussDB给上述数据类型指定默认值时可以不添加括号。 - GaussDB指定默认值时不会校验默认值是否溢出；MySQL指定不带括号形式的默认值时会校验是否溢

概述	详细语法说明	差异
		出，指定带括号形式的默认值时不会校验是否溢出。 GaussDB支持使用DATE/TIME/TIMESTAMP开头的时间常量给字段指定默认值，MySQL使用DATE/TIME/TIMESTAMP开头的时间常量给字段指定默认值时必须要在默认值外添加括号。 <pre>-- GaussDB m_db=# DROP TABLE IF EXISTS t1, t2; DROP TABLE m_db=# CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); CREATE TABLE m_db=# CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); CREATE TABLE -- MySQL 5.7 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.02 sec) mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); ERROR 1067 (42000): Invalid default value for 'a' mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(TIMESTAMP '2000-01-01 00:00:00'))' at line 1 -- MySQL 8.0 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.14 sec) mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); ERROR 1067 (42000): Invalid default value for 'a' mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); Query OK, 0 rows affected (0.19 sec)</pre>

3.2.7.4 DML

表 3-28 DML 语法兼容介绍

概述	详细语法说明	差异
DELETE支持从指定分区（或子分区）删除数据	DELETE	-
UPDATE支持ORDER BY和LIMIT	UPDATE	-
SELECT INTO语法	SELECT	- GaussDB可以使用SELECT INTO根据查询结果创建一个新表，MySQL不支持。 - GaussDB的SELECT INTO语法不支持将多个查询进行集合运算后的结果作为查询结果。

概述	详细语法说明	差异
REPLACE INTO语法	REPLACE	时间类型初始值的差异。例如： - MySQL不受严格模式和宽松模式的影响，可向表中插入时间0值，即： mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, f2, f3); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; +-----+ +-----+-----+ f1 f2 f3 +-----+ +-----+-----+ 0000-00-00 00:00:00 0000-00-00 00:00:00 0000-00-00 +-----+ +-----+-----+ 1 row in set (0.00 sec) - GaussDB在宽松模式下才可以成功插入时间0值，即 gaussdb=# SET sql_mode = ''; SET gaussdb=# CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); CREATE TABLE gaussdb=# REPLACE INTO test VALUES(f1, f2, f3); REPLACE 0 1 gaussdb=# SELECT * FROM test; f1 f2 f3 ----- +-----+-----+ 0000-00-00 00:00:00 0000-00-00 00:00:00 0000-00-00 (1 row) 在严格模式下，则报错 Incorrect Date/Time/ Datetime/Timestamp/Year value。
SELECT支持指定多分区查询	SELECT	-
UPDATE支持指定多分区更新	UPDATE	-

概述	详细语法说明	差异
LOAD DATA导入数据功能	LOAD DATA	在使用LOAD DATA导入数据功能时，GaussDB与MySQL相比有如下差异： - LOAD DATA语法执行结果与M*严格模式一致，宽松模式暂未适配。 - IGNORE与LOCAL参数功能仅为当导入数据与表中数据存在冲突时，忽略当前冲突行数据功能和当文件中字段数小于指定表中列数时自动为其余列填充默认值功能，其余功能暂未适配。 [(col_name_or_user_var [, col_name_or_user_var] ...)]指定列参数不支持重复指定列。 [FIELDS TERMINATED BY 'string']指定换行符不能与[LINES TERMINATED BY 'string']分隔符相同。 - 执行LOAD DATA语法写入表中的数据若无法转换为表中数据类型格式时报错。 - LOAD DATA SET表达式中不支持指定列名计算。 - LOAD DATA只能用于表，不能用于视图。 - Windows下的文件与Linux环境下文件默认换行符存在差异，LOAD DATA无法识别此场景会报错，建议用户导入时检查导入文件行尾换行符。 - GaussDB不设置GUC参数m_format_behavior_compat_options值时，LOAD DATA无论是否指定LOCAL参数都仅支持从服务端导入数据；MySQL在指定LOCAL参数时支持从客户端环境导入数据，不指定LOCAL时则从服务端环境导入数据；GaussDB设置GUC参数值包含enable_load_data_remote_transmission后，LOAD

概述	详细语法说明	差异
		DATA LOCAL参数行为与MySQL一致。
INSERT支持VALUES引用列语法	INSERT INTO tabname VALUES(1,2,3) ON DUPLICATE KEY UPDATE b = VALUES(column_name)	GaussDB的ON DUPLICATE KEY UPDATE子句中的VALUES()不支持表名.列名格式，MySQL支持。
LIMIT限制差异	DELETE、SELECT、 UPDATE	各个语句的limit子项与MySQL的limit项当前存在差异。 GaussDB中limit参数最大值为BIG INT类型限制（超过9223372036854775807报错）。在MySQL中，limit最大值为unsigned LONGLONG类型限制（超过18446744073709551615 报错）。 limit可以设置小数值，实际执行时四舍五入。MySQL不能取小数。
反斜杠(\)用法差异	INSERT	反斜杠(\)的用法在GaussDB和MySQL中都可以由参数控制但当前默认用法不同： MySQL中使用参数NO_BACKSLASH_ESCAPES控制字符串和标识符中的反斜杠(\)被解析为普通字符还是转义字符，默认反斜杠字符(\)作为字符串和标识符中的转义字符。使用“SET sql_mode='NO_BACKSLASH_ESCAPES;”语句可以禁用反斜杠字符(\)作为字符串和标识符中的转义字符。 GaussDB中使用参数standard_conforming_strings控制字符串和标识符中的反斜杠(\)被解析为普通字符还是转义字符。默认值为on，在普通字符串文本中按照SQL标准把反斜杠(\)当普通文本。使用“SET standard_conforming_strings=off;”语句将反斜杠字符(\)作为字符串和标识符中的转义字符。

概述	详细语法说明	差异
插入值少于字段数目时，MySQL报错，GaussDB补充空值。	INSERT	GaussDB不指定列的列表时，如果插入值少于字段数目，默认按建表时的字段顺序赋值。字段上有非空约束时报错，没有非空约束时，如果指定了默认值则缺省部分补充默认值，若未指定默认值则补充空。
ORDER BY中排序的列必须包括在结果集的列中。	SELECT	在GaussDB中，在与GROUP BY子句一起使用的情况下，ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。在与DISTINCT关键字一起使用的情况下，ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。
INSERT ON DUPLICATE KEY UPDATE修改约束字段。	INSERT	GaussDB不允许对约束字段用ON DUPLICATE KEY UPDATE进行修改。MySQL允许。
SELECT结果允许存在重复列名。	SELECT	-
NATURAL JOIN与MySQL有差异。	SELECT	在GaussDB中，NATURAL [[LEFT RIGHT] OUTER] JOIN允许不指定LEFT RIGHT，不指定时NATURAL OUTER JOIN为NATURAL JOIN。允许连续使用多次JOIN。
外键数据类型是timestamp/datetime时，UPDATE/DELETE外表报错。	UPDATE/DELETE	外键数据类型是timestamp/datetime时，UPDATE/DELETE外表报错，MySQL成功。

概述	详细语法说明	差异
nature join和using兼容。	SELECT	- GaussDB join的顺序严格按照从左往右，MySQL可能会调整顺序。 - GaussDB和MySQL在natural join与using时均不允许左表或右表参与join的字段出现歧义（一般由左或右临时表中重名字段造成）。因为两者join的顺序有差别，故行为上可能有差别。 - GaussDB的行为： <pre>m_regression=# CREATE TABLE t1(a int,b int); CREATE TABLE m_regression=# CREATE TABLE t2(a int,b int); CREATE TABLE m_regression=# CREATE TABLE t3(a int,b int); CREATE TABLE m_regression=# SELECT * FROM t1 JOIN t2; a b a b ----+----+----+---- (0 rows) m_regression=# SELECT * FROM t1 JOIN t2 natural join t3; -- failed, 因为:列a,b在t1 join t2 得到的临时表中存在重 复, 故nature join存在歧义。 ERROR: common column name "a" appears more than once in left table</pre> - MySQL的行为： <pre>mysql> SELECT * FROM t1 JOIN t2 NATURAL JOIN t3; Empty set (0.00 sec) mysql> SELECT * FROM (t1 join t2) NATURAL JOIN t3; ERROR 1052 (23000): Column 'a' in from clause is ambiguous</pre>
with clause兼容MySQL 8.0版本	SELECT、INSERT、UPDATE、DELETE	-
join兼容	SELECT	GaussDB join不支持使用逗号“,”的连接方式，MySQL支持。 GaussDB不支持use index for join。
SELECT语句中列表达式为函数表达式、算数表达式等情形时，查询结果显示的列名为？column？	SELECT	GaussDB SELECT语句中列表达式为函数表达式、算数表达式等情形时，查询结果显示的列名为？column？，MySQL为对应表达式。

概述	详细语法说明	差异
SELECT导出文件（into outfile）	SELECT ... INTO OUFILe ...	SELECT INTO OUTFILE语法，导出文件中FLOAT、DOUBLE、REAL类型的值显示精度和MySQL存在差异，不影响COPY导入和导入后的值。

概述	详细语法说明	差异
SELECT/UPDATE/ INSERT/REPLACE指定 模式名、表名	SELECT/UPDATE/ INSERT/REPLACE	- SELECT语句指定投影列时，MySQL支持“模式名.表别名.列名”的三段式用法，GaussDB不支持。 m_db=# CREATE SCHEMA test; CREATE SCHEMA m_db=# CREATE TABLE test.t1(a int); CREATE TABLE m_db=# SELECT test.alias1.a FROM t1 alias1; ERROR: invalid reference to FROM-clause entry for table "alias1" LINE 1: SELECT test.alias1.a FROM t1 alias1; ^ HINT: There is an entry for table "alias1", but it cannot be referenced from this part of the query. CONTEXT: referenced column: a - UPDATE/REPLACE SET中，MySQL的三段式用法为 database.table.column；GaussDB的三段式用法为 table.column.field，其中 field为指定复合类型中的属性。 - INSERT ... SET中，MySQL支持使用column、table.column和 database.table.column；GaussDB只支持使用 column，不支持使用 table.column和 database.table.column。 - INSERT ... SET中，MySQL支持SET子句中等式右边引用列名以及包含列名的表达式，GaussDB不支持： - GaussDB的行为： m_db=# CREATE TABLE t2 (a int default 3, b int default 5); CREATE TABLE m_db=# INSERT INTO t2 SET a = b + 1; ERROR: Column "b" does not exist. LINE 1: INSERT INTO t2 SET a = b + 1; ^ HINT: There is a column named "b" in table "t2", but it cannot be referenced from

概述	详细语法说明	差异
		this part of the query. <pre>m_db=# INSERT INTO t2 SET a = b + 1, b = 0; ERROR: Column "b" does not exist. LINE 1: INSERT INTO t2 SET a = b + 1, b = 0; ^ HINT: There is a column named "b" in table "t2", but it cannot be referenced from this part of the query.</pre><pre>m_db=# INSERT INTO t2 SET b = 0, a = b + 1; ERROR: Column "b" does not exist. LINE 1: INSERT INTO t2 SET b = 0, a = b + 1; ^ HINT: There is a column named "b" in table "t2", but it cannot be referenced from this part of the query.</pre><pre>m_db=# INSERT INTO t2 SET a = a + 1; ERROR: Column "a" does not exist. LINE 1: INSERT INTO t2 SET a = a + 1; ^ HINT: There is a column named "a" in table "t2", but it cannot be referenced from this part of the query.</pre><pre>m_db=# DROP TABLE t2; DROP TABLE</pre> – MySQL的行为： <pre>mysql> CREATE TABLE t2 (a int default 3, b int default 5); Query OK, 0 rows affected (0.07 sec) mysql> INSERT INTO t2 SET a = b + 1; Query OK, 1 row affected (0.02 sec) mysql> SELECT * FROM t2; +-----+-----+ a b +-----+-----+ 6 5 +-----+-----+ 1 row in set (0.00 sec) mysql> INSERT INTO t2 SET a = b + 1, b = 0; Query OK, 1 row affected (0.00 sec)</pre>

概述	详细语法说明	差异
		<pre>mysql> SELECT * FROM t2; +-----+-----+ a b +-----+-----+ 6 5 6 0 +-----+-----+ 2 rows in set (0.00 sec) mysql> INSERT INTO t2 SET b = 0, a = b + 1; Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM t2; +-----+-----+ a b +-----+-----+ 6 5 6 0 1 0 +-----+-----+ 3 rows in set (0.00 sec) mysql> INSERT INTO t2 SET a = a + 1; Query OK, 1 row affected (0.02 sec) mysql> SELECT * FROM t2; +-----+-----+ a b +-----+-----+ 6 5 6 0 1 0 4 5 +-----+-----+ 4 rows in set (0.00 sec) mysql> DROP TABLE t2; Query OK, 4 rows affected (0.40 sec)</pre>
UPDATE SET执行顺序与MySQL存在差异	UPDATE ... SET	MySQL中，UPDATE SET的顺序是从前往后依次UPDATE，前面UPDATE的结果会影响后面的结果，且允许多次设置同一列；GaussDB中为先取出原来的所有相关的数据，再一次性UPDATE，且不允许多次设置同一列，二者存在差异。
IGNORE特性	UPDATE/DELETE/INSERT	MySQL数据库和GaussDB执行过程的差异，因此产生的WARNING条数和WARNING信息可能存在不同。

概述	详细语法说明	差异
SHOW COLUMNS语法	SHOW	● 用户权限验证与MySQL存在差异。 - GaussDB中需要拥有指定表所在Schema的USAGE权限，同时还需要拥有指定表的任意表级权限或列级权限，仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 - MySQL中需要拥有指定表的任意表级权限或列级权限，仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 ● LIKE和WHERE子句中涉及到字符串比较操作时，排序规则与MySQL存在差异。 - MySQL中使用utf8_general_ci。 - GaussDB中排序规则使用当前客户端的collation_connection。GaussDB中建议用户在WHERE子句中，不要对返回字段以外的列进行选择，否则可能会出现非预期的报错。 -- 预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE `b`='pri'; ERROR: Column "b" does not exist. LINE 1: SHOW FULL COLUMNS FROM t02 WHERE `b`='pri'; ^ -- 非预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE `c`='pri'; ERROR: input of anonymous composite types is not implemented LINE 1: SHOW FULL COLUMNS FROM t02 WHERE `c`='pri';

概述	详细语法说明	差异
		^
SHOW CREATE DATABASE语法	SHOW	用户权限验证与MySQL存在差异。 - GaussDB中需要拥有指定Schema的USAGE权限。 - MySQL中需要拥有任意库级权限（除GRANT OPTION和USAGE）、任意表级权限（除GRANT OPTION）或任意列级权限。

概述	详细语法说明	差异
SHOW CREATE TABLE 语法	SHOW	● 用户权限验证与MySQL存在差异。 – GaussDB中需要拥有指定表所在Schema的USAGE权限和指定表的任意表级权限。 – MySQL中需要拥有指定表的任意表级权限（除GRANT OPTION）。 ● 返回的建表语句与MySQL存在差异。 – GaussDB中索引以CREATE INDEX语句的形式返回。MySQL中表的索引在CREATE TABLE语句中返回。主要因为GaussDB中CREATE INDEX语法支持的可选参数范围与CREATE TABLE语法中创建索引不同，因此某些索引无法在CREATE TABLE语句中创建。 – GaussDB中CREATE TABLE语法的ENGINE和ROW_FORMAT选项仅做了语法适配，实际不生效，因此在返回的建表语句中不予显示。 ● 设置兼容性参数 m_format_dev_version为's2'后，返回的建表语句才兼容MySQL。兼容的内容包括：列注释位置变更、表注释位置变更、全局临时表ON COMMIT选项位置变更、主键与唯一约束位置变更、主键与唯一约束中的USING INDEX TABLESPACE选项不再显示以及索引注释位置变更。

概述	详细语法说明	差异
SHOW CREATE VIEW 语法	SHOW	● 用户权限验证与MySQL存在差异。 - GaussDB中需要拥有指定视图所在Schema的USAGE权限和指定视图的任意表级权限。 - MySQL中需要拥有指定视图的表级SELECT和表级SHOW VIEW权限。 ● 返回的视图创建语句与MySQL存在差异。以SELECT * FROM tbl_name形式创建的视图，GaussDB中*不会被展开，而MySQL中会展开。 ● 返回结果中的character_set_client和collation_connection字段与MySQL存在差异。 - MySQL中显示视图创建时系统变量character_set_client和collation_connection的会话值 - GaussDB中未记录相关元数据，显示为NULL。
SHOW PROCESSLIST 语法	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.processlist视图内字段内容与大小写保持一致，MySQL中可能存在差异。 ● GaussDB中用户只能访问自己的线程信息，拥有SYSADMIN权限的用户可以访问所有用户的线程信息。 ● MySQL中用户只能访问自己的线程信息，拥有PROCESS权限的用户可以访问所有用户的线程信息。

概述	详细语法说明	差异
SHOW [STORAGE] ENGINES	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.engines视图内字段内容与大小写保持一致，MySQL中可能存在差异。因为MySQL与GaussDB的存储引擎不同，所以该指令查询的结果不同。
SHOW [SESSION] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.session_status视图内字段内容与大小写保持一致，MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。
SHOW [GLOBAL] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.global_status视图内字段内容与大小写保持一致，MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。
SHOW INDEX	SHOW	● 用户权限验证与MySQL存在差异。 – GaussDB中需要拥有指定SCHEMA的USAGE权限和指定表的任意表级权限或者任意列级权限。 – MySQL中需要拥有指定表的任意表级权限（除GRANT OPTION）或者任意列级权限。 ● GaussDB中临时表存储于独立的临时Schema中，在使用FROM或IN db_name条件来展示指定临时表索引信息时，须指明db_name为临时表所在的Schema才能展示临时表索引信息，否则会提示不存在该临时表，这一点和MySQL在部分情况下存在差异。

概述	详细语法说明	差异
SHOW SESSION VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.session_variables视图内字段内容及大小写保持一致，与MySQL可能存在差异。
SHOW GLOBAL VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.global_variables视图内字段内容及大小写保持一致，与MySQL可能存在差异。
SHOW CHARACTER SET	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.character_sets视图内字段内容及大小写保持一致，与MySQL可能存在差异。
SHOW COLLATION	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.collations视图内字段内容及大小写保持一致，与MySQL可能存在差异。
EXCEPT语法	SELECT	-
SELECT支持STRAIGHT_JOIN语法	SELECT	GaussDB中多表关联JOIN场景下生成的执行计划，与MySQL可能存在差异。

概述	详细语法说明	差异
SHOW TABLES	SHOW	- LIKE行为存在差异，具体请参见操作符节的“LIKE”。 - WHERE表达式行为存在差异，具体行为请参见GaussDB数据库的“WHERE表达式”。 - GaussDB中：表和数据库的权限需要分开赋予用户，查询的数据库必须是用户在SHOW SCHEMAS上可以查询到，不能仅仅有表的权限，必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。 - GaussDB中：校验逻辑中优先校验Schema是否存在，后校验当前用户是否对Schema具有权限，与MySQL存在差异。
SHOW TABLE STATUS	SHOW	- GaussDB中：该语法展示数据依赖information_schema下的tables视图。MySQL中tables指定的是表。 - GaussDB中：表和数据库的权限需要分开赋予用户，查询的数据库必须是用户在SHOW SCHEMAS上可以查询到，不能仅仅有表的权限，必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。 - GaussDB中：校验逻辑中优先校验Schema是否存在，后校验当前用户是否对Schema具有权限，与MySQL存在差异。
HAVING语法	SELECT	GaussDB的HAVING必须且只能引用GROUP BY子句中的列或聚合函数中使用的列。MySQL支持对此行为的扩展，并允许HAVING引用列表中的SELECT列和外部子查询中的列。

概述	详细语法说明	差异
SELECT后跟行表达式	SELECT	MySQL不支持SELECT后跟行表达式，GaussDB支持SELECT后跟行表达式。 MySQL的行为： mysql> SELECT row(1,2); ERROR 1241 (21000): Operand should contain 1 column(s) GaussDB的行为： m_db=# SELECT row(1,2); row(1,2) ----- (1,2) (1 row)

概述	详细语法说明	差异
SELECT FOR SHARE/FOR UPDATE/ LOCK IN SHRAE MODE	SELECT	- GaussDB不支持FOR SHARE/FOR UPDATE/ LOCK IN SHARE MODE子句和UNION/EXCEPT/DISTINCT/GROUP BY/HAVING子句一起使用，MySQL 5.7部分支持（FOR SHARE/EXCEPT语法不支持），MySQL 8.0均支持。 - 当将锁子句与LEFT/RIGHT [OUTER] JOIN子句连用时，LEFT JOIN不支持给右表上锁，RIGHT JOIN不支持给左表上锁；MySQL可以给JOIN两侧的表同时上锁。 - MySQL不支持给同一张表指定多次锁子句；GaussDB支持给同一张表指定多次锁子句，实际生效按照最强的锁处理。 <pre>-- GaussDB m_db=# DROP TABLE IF EXISTS t1; DROP TABLE m_db=# CREATE TABLE t1(a INT, b INT); CREATE TABLE m_db=# INSERT INTO t1 VALUES(1,2); INSERT 0 1 m_db=# SELECT * FROM t1 FOR UPDATE OF t1 LOCK IN SHARE MODE; a b ---+--- 1 2 (1 row) m_db=# DROP TABLE t1; DROP TABLE -- MySQL mysql> DROP TABLE IF EXISTS t1; Query OK, 0 rows affected (0.05 sec) mysql> CREATE TABLE t1(a INT, b INT); Query OK, 0 rows affected (0.09 sec) mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.01 sec)</pre>

概述	详细语法说明	差异
		<pre>mysql> SELECT * FROM t1 FOR UPDATE OF t1 LOCK IN SHARE MODE; ERROR 3569 (HY000): Table t1 appears in multiple locking clauses. mysql> DROP TABLE t1; Query OK, 0 rows affected (0.05 sec)</pre>

概述	详细语法说明	差异
SELECT语法支持范围	SELECT	- GaussDB的HAVING子句必须与GROUP BY子句或聚合函数一同使用，MySQL支持在查询语句中仅指定HAVING子句。 - GaussDB的HAVING必须且只能引用GROUP BY子句中的列或聚合函数中使用的列。MySQL支持对此行为的扩展，并允许HAVING引用列表中的SELECT列和外部子查询中的列。 - GaussDB以及MySQL5.7使用GROUP BY WITH ROLLUP时，不支持与DISTINCT、ORDER BY子句共同使用，MySQL8.0支持。 - 当查询表为空表时，GaussDB在使用指定WITH ROLLUP语句进行查询时，查询结果为一条空行数据，而MySQL查询结果为空。 - GaussDB指定FROM子句中的表别名时，支持带字段名称。MySQL 5.7不支持指定表别名时带字段名称，MySQL 8.0仅支持给子查询指定表别名时带字段名称。 <pre> -- GaussDB m_db=# DROP TABLE IF EXISTS t1; DROP TABLE m_db=# CREATE TABLE t1(a INT, b INT); CREATE TABLE m_db=# INSERT INTO t1 VALUES(1,2); INSERT 0 1 m_db=# SELECT * FROM t1 t2(a, b); a b ---+--- 1 2 (1 row) m_db=# SELECT * FROM (SELECT * FROM t1) t2(a, b); a b ---+--- 1 2 (1 row) -- MySQL 5.7 mysql> DROP TABLE IF EXISTS t1; Query OK, 0 rows affected, 1 </pre>

概述	详细语法说明	差异
		<pre>warning (0.00 sec) mysql> CREATE TABLE t1(a INT, b INT); Query OK, 0 rows affected (0.03 sec) mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.01 sec) mysql> SELECT * FROM t1 t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1 mysql> SELECT * FROM (SELECT * FROM t1) t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1 -- MySQL 8.0 mysql> DROP TABLE IF EXISTS t1; Query OK, 0 rows affected (0.10 sec) mysql> CREATE TABLE t1(a INT, b INT); Query OK, 0 rows affected (0.18 sec) mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.03 sec) mysql> SELECT * FROM t1 t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1 mysql> SELECT * FROM (SELECT * FROM t1) t2(a, b); +-----+-----+ a b +-----+-----+ 1 2 +-----+-----+ 1 row in set (0.00 sec)</pre> 当查询语句不带FROM子句时，GaussDB支持带WHERE子句，与MySQL 8.0保持一致，MySQL 5.7不支持。

概述	详细语法说明	差异
		-- GaussDB m_db=# SELECT 1 WHERE true; 1 --- 1 (1 row) -- MySQL 5.7 mysql> SELECT 1 WHERE true; ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'where true' at line 1 -- MySQL 8.0 mysql> SELECT 1 WHERE true; +----+ 1 +----+ 1 +----+ 1 row in set (0.00 sec)

概述	详细语法说明	差异
INSERT ... ON DUPLICATE KEY UPDATE语法	INSERT	- GaussDB的ON DUPLICATE KEY UPDATE子句中的VALUES()不支持表名.列名格式，MySQL支持。 - GaussDB不允许对约束字段用ON DUPLICATE KEY UPDATE进行修改，MySQL允许。 - INSERT ... query ON DUPLICATE KEY UPDATE语句中，query为子查询时，GaussDB的ON DUPLICATE KEY UPDATE子句不支持引用子查询中的列名；MySQL支持。 - ON DUPLICATE KEY UPDATE子句更新多列时，MySQL前面UPDATE的结果会影响后面的结果，且允许同一列设置多次。GaussDB中，前面UPDATE的结果不会影响后面的结果，且不支持同一列设置多次，二者存在差异。 - 插入数据违反唯一约束执行update操作时，GaussDB和MySQL返回的受影响行数存在差异。更新一条数据时，GaussDB返回1，MySQL返回2；如果将现有行更新为其当前值，GaussDB返回1，MySQL返回0。

概述	详细语法说明	差异
GROUP BY HAVING字段为查询投影列别名且与表字段名称重合，语法可能存在歧义，与MySQL结果可能存在差异	SELECT	- 查询表达式别名与表字段重名的情况：GaussDB中无论GROUP BY后面是数字还是名称，HAVING条件中名称指代表字段。 - MySQL中GROUP BY后面是数字，HAVING条件中名称指代投影列中的表达式别名，GROUP BY后面是名称，HAVING条件中名称指代表字段。 <pre>m_db=# CREATE TABLE t1(col_int int); CREATE TABLE m_db=# INSERT INTO t1 VALUES(1), (2); INSERT 0 2 m_db=# SELECT abs(col_int) + 2 AS col_int FROM t1 GROUP BY col_int HAVING col_int > 2; col_int ----- (0 rows) m_db=# SET sql_mode = ''; SET m_db=# SELECT abs(col_int) + 2 AS col_int FROM t1 GROUP BY 1 HAVING col_int > 2; col_int ----- (0 rows) mysql> CREATE TABLE t1(col_int int); Query OK, 0 rows affected (0.00 sec) mysql> INSERT INTO t1 VALUES(1), (2); Query OK, 2 rows affected (0.00 sec) Records: 2 Duplicates: 0 Warnings: 0 mysql> SELECT abs(col_int) + 2 AS col_int FROM t1 GROUP BY col_int HAVING col_int > 2; Empty set, 2 warnings (0.00 sec) mysql> SELECT abs(col_int) + 2 AS col_int FROM t1 GROUP BY 1 HAVING col_int > 2; +-----+ col_int +-----+ 3 4 +-----+ 2 rows in set (0.00 sec)</pre>

3.2.7.5 DCL

表 3-29 DCL 语法兼容介绍

概述	详细语法说明	差异
SET NAMES指定 COLLATE子句	SET [SESSION LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] DEFAULT};	GaussDB中暂不支持指定 charset_name与数据库字符集不同。具体请参考《M-Compatibility开发指南》中“SQL参考 > SQL语法 > SQL语句 > S > SET”章节。
USE切换当前模式	USE schema_name	使用USE语句指定模式，当用户没有对应模式的USAGE权限时，MySQL产生报错，GaussDB会将当前模式指定为空。 -- MySQL mysql> USE test; ERROR 1044 (42000): Access denied for user 'u1'@'%' to database 'test' -- GaussDB m_db=> USE test; SET m_db=> SELECT database(); ERROR: function returned NULL CONTEXT: referenced column: database

3.2.7.6 其他语句

表 3-30 其他语法兼容介绍

概述	详细语法说明	差异
事务相关语法	数据库默认隔离级别	M-Compatibility默认隔离级别为READ COMMITTED，MySQL默认隔离级别为REPEATABLE READ。 M-Compatibility隔离级别只有READ COMMITTED、REPEATABLE READ生效。
事务相关语法	事务嵌套	M-Compatibility中嵌套事务不会自动提交，MySQL会自动提交。

概述	详细语法说明	差异
事务相关语法	自动提交	M-Compatibility使用GaussDB存储，继承GaussDB事务机制，事务中执行DDL，DCL不会自动提交。MySQL在DDL、DCL、管理类语句，锁相关语句会自动提交。
事务相关语法	报错后需rollback	M-Compatibility事务中报错，需要执行rollback，MySQL无限制。
事务相关语法	锁机制	M-Compatibility锁机制只能在事务块中使用，MySQL无限制。
锁机制	锁机制	- MySQL获取read锁后，当前会话无法进行写操作，M-Compatibility获取read锁后，当前会话可以进行写操作。 - MySQL给表上锁后，读取其他表报错，M-Compatibility无限制。 - MySQL同一会话中获取同一个表的锁，会自动释放上一个锁，并提交事务，M-Compatibility无该机制。 - M-Compatibility中LOCK TABLE只能在一个事务块的内部有用，且无UNLOCK TABLE命令，锁总是在事务结束时释放。
PBE	PBE	- 重复创建同名的PREPARE语句，M-Compatibility会报已经存在的错误，需要先删除已有statement，MySQL会覆盖旧的statement。 - M-Compatibility和MySQL在SQL语句执行过程中对异常场景的报错阶段不同，例如解析层、执行层等；而PREPARE语句对预备语句只处理到解析层。因此PBE下对于异常场景，报错位置在PREPARE阶段还是EXECUTE阶段，M-Compatibility和MySQL存在可能差异。

概述	详细语法说明	差异
单行注释语法	单行注释语法	单行注释语法仅在设置参数 m_format_behavior_compat_ options='forbid_none_space_c omment'后注释行为与MySQL 一致。

3.2.7.7 用户与权限

概述

在M-Compatibility中，用户与权限管控相关的行为、语法整体沿用GaussDB的机制，暂不同步MySQL。

用户与权限的行为与GaussDB保持一致，具体行为说明请参见《开发指南》中的“数据库安全 > 用户及权限”章节。

用户与权限的语法在原有GaussDB的基础上，裁剪了部分语法，具体语法说明请参见《M-Compatibility开发指南》中的“SQL参考 > SQL语法 > SQL语句”章节。M-Compatibility与GaussDB的语法差异请参见[表3-31](#)。

表 3-31 M-Compatibility 与 GaussDB 的语法差异

语法说明	概述	M-Compatibility与GaussDB 的差异
CREATE ROLE	创建一个角色。	在M-Compatibility中，不支持指定涉及以下关键字的选项：ENCRYPTED、UNENCRYPTED、RESOURCE POOL、PERM SPACE、TEMP SPACE、SPILL SPACE。 在M-Compatibility中，创建USER时会自动创建与USER同名的Schema，MySQL不创建。
CREATE USER	创建一个用户。	
CREATE GROUP	创建一个新用户组。 CREATE GROUP是CREATE ROLE的别名，不推荐使用。	
ALTER ROLE	修改角色属性。	
ALTER USER	修改用户属性。	
ALTER GROUP	修改一个用户组的属性。	-
DROP ROLE	删除角色。	-
DROP USER	删除用户。	-
DROP GROUP	删除用户组。	-
DROP OWNED	删除一个数据库角色所拥有的数据库对象。	-
REASSIGN OWNED	修改数据库对象的属主。	M-Compatibility中不支持该语法。

语法说明	概述	M-Compatibility与GaussDB的差异
GRANT	对角色和用户进行授权操作。	M-Compatibility中不支持授予或回收函数、存储过程、表空间、DATABASE LINK等对象的权限。
REVOKE	用于撤销一个或多个角色的权限。	
ALTER DEFAULT PRIVILEGES	设置应用于将来创建的对象权限（这不会影响分配到已有对象中的权限）。	M-Compatibility中不支持该语法。

差异说明

- 语法格式差异

M-Compatibility的授权语法请参见《M-Compatibility开发指南》中的“SQL参考 > SQL语法 > SQL语句 > G > GRANT”章节，MySQL中的授权语法如下：

```
-- 全局级、数据库级、表级、存储过程级赋权语法
GRANT
    priv_type [(column_list)]
        [, priv_type [(column_list)]] ...
    ON [object_type] priv_level
    TO user [auth_option] [, user [auth_option]] ...
    [REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
    [WITH {GRANT OPTION | resource_option} ...]

-- 用户代理赋权语法
GRANT PROXY ON user
    TO user [, user] ...
    [WITH GRANT OPTION]

object_type: {
    TABLE
  | FUNCTION
  | PROCEDURE
}

priv_level: {
    *
  | *.*
  | db_name.*
  | db_name.tbl_name
  | tbl_name
  | db_name.routine_name
}

user:
    'user_name'@'host_name'

auth_option: {
    IDENTIFIED BY 'auth_string'
  | IDENTIFIED WITH auth_plugin
  | IDENTIFIED WITH auth_plugin BY 'auth_string'
  | IDENTIFIED WITH auth_plugin AS 'auth_string'
  | IDENTIFIED BY PASSWORD 'auth_string'
}

tls_option: {
    SSL
  | X509
```

```
| CIPHER 'cipher'  
| ISSUER 'issuer'  
| SUBJECT 'subject'  
}  
  
resource_option: {  
| MAX_QUERIES_PER_HOUR count  
| MAX_UPDATES_PER_HOUR count  
| MAX_CONNECTIONS_PER_HOUR count  
| MAX_USER_CONNECTIONS count  
}
```

- 赋权类型差异
MySQL支持的赋权类型如下：

表 3-32 MySQL 支持的赋权类型

权限类型	释义及权限级别
ALL [PRIVILEGES]	授予指定访问级别的所有权限，除了 GRANT OPTION和 PROXY。
ALTER	启用ALTER TABLE。级别：全局、数据库、表。
ALTER ROUTINE	允许更改或删除存储过程。级别：全局、数据库、例程。
CREATE	启用数据库和表创建。级别：全局、数据库、表。
CREATE ROUTINE	启用存储过程创建。级别：全局、数据库。
CREATE TABLESPACE	允许创建、更改或删除表空间和日志文件组。级别：全局。
CREATE TEMPORARY TABLES	启用CREATE TEMPORARY TABLE。级别：全局、数据库。
CREATE USER	启用CREATE USER、DROP USER、RENAME USER和REVOKE ALL PRIVILEGES。级别：全局。
CREATE VIEW	允许创建或更改视图。级别：全局、数据库、表。
DELETE	启用DELETE。级别：全局、数据库、表。
DROP	允许删除数据库、表和视图。级别：全局、数据库、表。
EVENT	启用定时任务。级别：全局、数据库。
EXECUTE	使用户能够执行存储过程。级别：全局、数据库、存储过程。
FILE	使用户能够使服务器读取或写入文件。级别：全局。

权限类型	释义及权限级别
GRANT OPTION	允许向其他账户授予权限或从其他账户删除权限。级别：全局、数据库、表、存储过程、代理。
INDEX	允许创建或删除索引。级别：全局、数据库、表。
INSERT	启用 INSERT 。级别：全局、数据库、表、列。
LOCK TABLES	在具有SELECT权限的表上启用LOCK TABLES 。级别：全局、数据库。
PROCESS	使用户能够通过 SHOW PROCESSLIST 查看所有正在运行的线程。级别：全局。
PROXY	启用用户代理。级别：从用户到用户。
REFERENCES	启用外键创建。级别：全局、数据库、表、列。
RELOAD	启用 FLUSH 操作的使用。级别：全局。
REPLICATION CLIENT	使用户能够查询源服务器或副本服务器的位置。级别：全局。
REPLICATION SLAVE	允许副本从源读取二进制日志。级别：全局。
SELECT	启用使用 SELECT 。级别：全局、数据库、表、列。
SHOW DATABASES	启用 SHOW DATABASES 以显示所有数据库。级别：全局。
SHOW VIEW	启用 SHOW CREATE VIEW 。级别：全局、数据库、表。
SHUTDOWN	启用 mysqladmin shutdown 的使用。级别：全局。
SUPER	启用其他管理操作，例如 CHANGE MASTER TO 、 KILL 、 PURGE BINARY LOGS 、 SET GLOBAL 和 mysqladmin debug 命令。级别：全局。
TRIGGER	启用触发器操作。级别：全局、数据库、表。
UPDATE	启用 UPDATE 。级别：全局、数据库、表、列。
USAGE	等价于“没有特权”。

M-Compatibility以级别划分支持以下权限：

表 3-33 M-Compatibility 支持的赋权类型

授权对象	支持授予的权限
数据库	CREATE、CONNECT、TEMPORARY、TEMP、ALTER、DROP、COMMENT
模式	CREATE、USAGE、ALTER、DROP、COMMENT
表、视图	SELECT、INSERT、UPDATE、DELETE、TRUNCATE、REFERENCES、TRIGGER、ALTER、DROP、COMMENT、INDEX、VACUUM
列	SELECT、INSERT、UPDATE、REFERENCES、COMMENT
序列	SELECT、USAGE、UPDATE、ALTER、DROP、COMMENT

- MySQL通过'*.*'表示全局层级的授权对象；GaussDB通过'{DATABASE} db_name'表示数据库层级的授权对象。GaussDB的数据库层级对应MySQL的全局层级。
- MySQL通过'schema_name.*'表示数据库/模式层级的授权对象；GaussDB通过'{SCHEMA} schema_name'表示模式层级的授权对象。GaussDB的模式层级对应MySQL的数据库/模式层级。
- MySQL中用户名为两部分：用户名@主机名；M-Compatibility当前仅支持用户名。
- MySQL支持在GRANT赋权语法中修改用户验证，安全连接，资源参数属性，即auth_option、tls_option和resource option；M-Compatibility赋权语法中不支持以上特性，需使用CREATE USER、ALTER USER设置用户相关属性。
- MySQL支持用户代理赋权，GRANT PROXY ON主要用于对多个用户进行统一的权限管理。MySQL 5.7未提供角色机制，而在MySQL 8.0和M-Compatibility中都提供了角色机制。角色能满足用户对于多个用户权限统一管控的目标，可以替代GRANT PROXY ON。
- M-Compatibility拥有public的概念，所用用户都拥有public的权限，部分系统表、系统视图可供所有用户查询。用户可以对public所拥有的权限进行grant和revoke；MySQL中，新创建的用户只拥有全局的usage权限，这个权限很小，几乎为0，只有连接数据库和查询information_schema 数据库的权限。
- M-Compatibility中，对象的所有者缺省具有该对象上的所有权限，出于安全考虑所有者可以舍弃部分权限，但ALTER、DROP、COMMENT、INDEX、VACUUM以及对对象的可再授予权限属于所有者固有的权限，隐式拥有；MySQL中，没有owner的概念，即使用户创建了表，如果没赋予用户对应权限，那么用户也不能对其创建的表进行IUD等操作。
- 在MySQL中，USAGE实际上表示无权限，所用用户都拥有该权限，当执行revoke或grant usage时，实际上不会进行任何修改；在M-Compatibility中，USAGE权限如下：
 - 对于模式，USAGE允许访问包含在指定模式中的对象，若没有该权限，则只能看到这些对象的名称。
 - 对于序列，USAGE允许使用nextval函数。
- 在M-Compatibility中，支持给用户设置管理员角色，包括系统管理员（SYSADMIN）、安全管理员（CREATEROLE）、审计管理员

（AUDITADMIN）、监控管理员（MONADMIN）、运维管理员（OPRADMIN）、安全策略管理员（POLADMIN）。默认情况下拥有SYSADMIN属性的系统管理员，具备系统最高权限。三权分立后，系统管理员将不再具有CREATEROLE属性（安全管理员）和AUDITADMIN属性（审计管理员）能力，即不再拥有创建角色和用户的权限，也不再拥有查看和维护数据库审计日志的权限；在MySQL中，不支持该用户设置管理员角色，也没有三权分立相关设计。

- 在M-Compatibility中，可以给用户赋予ANY权限，表示用户能够在非系统模式下拥有对应的权限，包括CREATE ANY TABLE、SELECT ANY TABLE、CREATE ANY INDEX等；在MySQL中，不支持ANY权限的赋予。
- MySQL中提供SHOW GRANTS查询用户权限；M-Compatibility中，可以通过gsq客户端元命令'\l+'、'\dn+'、'\dp'查询权限信息，也可以通过查询pg_namespace、pg_class、pg_attribute等系统表的权限相关字段查询权限信息。
- MySQL中数据库、表、列被删除时，相关的授权信息在系统表中依然保留，如果重新创建同名对象用户依然拥有权限；M-Compatibility中当数据库、表、列被删除时，相关的授权信息会被删除，在重新创建同名对象后需要重新授权。
- MySQL在授予数据库层级的权限时，支持‘_’和‘%’对数据库名进行模糊匹配；M-Compatibility不支持对象名模糊匹配，‘_’或‘%’等特殊字符被识别为普通字符。
- MySQL中，GRANT语句中指定用户不存在时默认会创建该账户（此特性已在MySQL 8.0中移除）；M-Compatibility不支持给未创建用户赋权。

3.2.7.8 系统表和系统视图

表 3-34 M-Compatibility 与 GaussDB 的系统表或系统视图的差异

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_schema.columns	generation_expression	该字段输出结果因涉及M-Compatibility与MySQL的表达式的字符串拼接逻辑的不同而存在差异。
information_schema.columns	data_type	该字段输出结果因涉及M-Compatibility的数据类型format_type输出，目前未修改，与MySQL存在差异。
information_schema.columns	column_type	该字段输出结果因涉及M-Compatibility的数据类型format_type输出，目前未修改，与MySQL存在差异。
information_schema.tables	engine	M-Compatibility中： - ENGINE对齐information_schema.engines数据。 - 部分系统表ENGINE为空。 - 在默认为ASTORE表且未指定STORAGE_TYPE时ENGINE为空。
information_schema.tables	version	M-Compatibility中不支持该字段。

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_s chema.tables	row_format	M-Compatibility中不支持该字段。
information_s chema.tables	avg_row_len gth	M-Compatibility下表示使用数据文件除以所有元组数（包括活元组和死元组）的结果。表中没有元组，值为null。
information_s chema.tables	max_data_le ngth	M-Compatibility中不支持该字段。
information_s chema.tables	data_free	M-Compatibility中表示死元组在总元组中的比例乘以数据文件大小。如果表中没有元组，则为null。
information_s chema.tables	check_time	M-Compatibility中不支持该字段。
information_s chema.tables	create_time	在M-Compatibility下，此字段与MySQL行为表现有差异，对于创建视图的情形MySQL中该字段置null，M-Compatibility则显示实际的创建表时间。数据库自带的表，视图设置null。
information_s chema.tables	update_time	M-Compatibility数据库自带的表，视图设置null。
information_s chema.statisti cs	collation	M-Compatibility只有值A、D，不会是NULL。
information_s chema.statisti cs	packed	M-Compatibility中不支持该字段。
information_s chema.statisti cs	sub_part	M-Compatibility中不支持该字段。
information_s chema.statisti cs	comment	M-Compatibility中不支持该字段。
information_s chema.partiti ons	subpartition_ name	M-Compatibility中，如果分区不是子分区，则为null。
information_s chema.partiti ons	subpartition_ ordinal_positi on	M-Compatibility中，如果分区不是子分区，则为null。

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_schema.partitions	partition_method	M-Compatibility中，分区策略。如果分区不是一级分区，则为null。 • 'r': 范围分区。 • 'i': 间隔分区。 • 'l': list分区。 • 'h': hash分区。
information_schema.partitions	subpartition_method	M-Compatibility中，子分区策略。如果分区不是二级分区，则为null。 • 'r': 范围分区。 • 'i': 间隔分区。 • 'l': list分区。 • 'h': hash分区。
information_schema.partitions	partition_description	M-Compatibility中，是区分一级分区和二级分区的。
information_schema.partitions	partition_expression	M-Compatibility中不支持该字段。
information_schema.partitions	subpartition_expression	M-Compatibility中不支持该字段。
information_schema.partitions	data_length	M-Compatibility中不支持该字段。
information_schema.partitions	max_data_length	M-Compatibility中不支持该字段。
information_schema.partitions	index_length	M-Compatibility中不支持该字段。
information_schema.partitions	data_free	M-Compatibility中不支持该字段。
information_schema.partitions	create_time	M-Compatibility中不支持该字段。
information_schema.partitions	update_time	M-Compatibility中不支持该字段。

系统表或系统视图	差异列	M-Compatibility与MySQL的差异
information_s chema.partiti ons	check_time	M-Compatibility中不支持该字段。
information_s chema.partiti ons	checksum	M-Compatibility中不支持该字段。
information_s chema.partiti ons	partition_co mment	M-Compatibility中不支持该字段。
information_s chema.partiti ons	nodegroup	M-Compatibility中不支持该字段。

说明

- 视图中对于整型的类型回显，不支持指定精度范围。如MySQL的bigint(1)，M-Compatibility下对应的是bigint类型，MySQL中bigint(21) unsigned，在M-Compatibility下对应的是bigint unsigned类型。
- MySQL中int类型，在M-Compatibility中是integer类型。
- MySQL中的set、enum类型的字段，M-Compatibility中都不支持。
m_schema.columns_priv视图的Column_priv字段、m_schema.tables_priv视图的Table_priv、Column_priv字段、m_schema.procs_priv视图的Routine_type、Proc_priv字段、m_schema.proc视图的type、language、sql_data_access、is_deterministic、security_type、sql_mode字段、m_schema.func视图的type字段均不支持，在此版本中不予显示。
- 由于information_schema.tables中table_rows、avg_row_length、data_length、data_free、index_length及information_schema.statistics中cardinality基于统计信息获取，查看前请先执行ANALYZE，更新统计信息后再查看（如果数据库中更新数据，建议延迟执行ANALYZE）。
- information_schema.statistics包含的索引列需要是创建索引中索引列是完整的表列，如果索引列是表达式，不在这个视图中。
- information_schema.partitions 中table_row基于统计信息获取，查看前请先执行ANALYZE，更新统计信息后再查看（如果数据库中更新数据，建议延迟执行ANALYZE）。
- 对于支持的grantee字段，MySQL的格式是'*user_name*@'*host_name*'，在M-Compatibility数据库，是被授予权限的用户或角色的名称。
- 对于支持的host字段，在M-Compatibility数据库，返回当前节点的hostname。
- m_schema.tables_priv、information_schema.user_privileges、information_schema.schema_privileges、information_schema.table_privileges、information_schema.column_privileges、m_schema.columns_priv、m_schema.func、m_schema.procs_priv 在MySQL下需要授权后才能查看视图内容，M-Compatibility数据库可以根据默认权限查看到对应的内容。如对于表t1，在MySQL下需要先对t1给对应的用户授权，才能在权限视图中看到对应的权限信息，M-Compatibility数据库下则可以直接在视图中看到t1表相关的权限信息。
- m_schema中的系统视图，但在MySQL是系统表。
- information_schema.views中的VIEW_DEFINITION以及information_schema.routines中的ROUTINE_DEFINITION不做字符序控制。
- 《M-Compatibility开发指南》中“Schema”章节所列举的字符类型的视图字段，字符集使用utf8mb4，字符序使用utf8mb4_bin或utf8mb4_general_ci，字符序优先级为《M-Compatibility开发指南》中“SQL参考 > 字符集与字符序 > 字符集和字符序合并规则”所描述的“支持字符序的数据类型的列”的优先级，和MySQL存在差异。

3.2.8 驱动

3.2.8.1 ODBC

3.2.8.1.1 ODBC 接口参考

获取参数描述信息

SQLDescribeParam接口是ODBC API中的一个函数，用于获取与预处理SQL语句（如调用SQLPrepare）相关参数的描述信息。它可以返回参数的类型、大小、是否允许NULL值等元数据，这对于动态构建SQL语句和绑定参数非常有用。

原型

```
SQLRETURN SQLDescribeParam(  
    SQLHSTMT      StatementHandle,  
    SQLUSMALLINT  ParameterNumber,  
    SQLSMALLINT   *DataTypePtr,  
    SQLULEN       *ParameterSizePtr,  
    SQLSMALLINT   *DecimalDigitsPtr,  
    SQLSMALLINT   *NullablePtr);
```

表 3-35 SQLDescribeParam 参数说明

参数名	参数说明	差异
StatementHandle	语句句柄。	-
ParameterNumber	参数序号，起始为1，依次递增。	-
DataTypePtr	指向返回参数数据类型的指针。	MySQL ODBC对于任意类型均返回SQL_VARCHAR。 GaussDB ODBC的会根据内核返回的不同类型判断返回给应用相应的DataType类型。
ParameterSizePtr	指向返回参数大小的指针。	MySQL ODBC若允许ODBC驱动程序使用更大的数据包进行数据传输，则返回24M，否则返回255。 GaussDB ODBC根据实际类型返回参数大小。
DecimalDigitsPtr	指向返回参数十进制位数的指针。	-
NullablePtr	指向返回参数是否允许NULL值的指针。	MySQL ODBC直接返回SQL_NULLABLE_UNKNOWN。 GaussDB ODBC直接返回SQL_NULLABLE。

3.2.8.2 JDBC

JDBC可以将数据库实际支持的数据类型和JDBC标准数据类型做转换，支持的数据类型参考《开发指南》中“应用程序开发教程 > 基于JDBC开发 > JDBC接口参考”章节中相关的set和get接口。

说明

在JDBC使用time()类型需要保持精度时，例如time(6)，会正确保留精度，行为与MySQL8.0保持一致。

3.3 MySQL 兼容性 MYSQL 模式

3.3.1 数据类型

3.3.1.1 数值数据类型

整数类型

除特别说明外，MySQL兼容性MySQL模式中的数据类型精度、标度、位数大小等默认不支持用浮点型数值定义，建议使用合法的整型数值定义。

整数类型公共差异说明：

- 输入格式：
 - MySQL
对于类似“asbd”、“12dd”、“12 12”等字符场景的输入，会采取截断或返回0值并上报WARNING处理，在严格模式时插表会失败。
 - GaussDB
 - 整数类型（TINYINT、SMALLINT、MEDIUMINT、INT、INTEGER、BIGINT）的输入，当非法字符串部分被截断时，如“12@3”，会直接截断并无提示信息，插表成功。
 - 当整数类型全部被截断（如“@123”）或字符串为空时，返回0，且插表成功。
- 操作符：
 - +、-、*操作符
GaussDB：INT/INTEGER/SMALLINT/BIGINT在进行运算时，返回值为类型本身，不会向上提升类型，当返回值超范围时报错。
MySQL：支持提升类型到BIGINT后计算。
 - |、&、^、~运算符
GaussDB：在类型所占用BIT位中计算；GaussDB中^表示指数运算，如需使用异或运算符，使用#替换。
MySQL：提升类型计算。
- 负数显示类型转换：
GaussDB：宽松模式结果为0，严格模式报错。
MySQL：依据其对应的二进制将最高位替换成数值位计算结果，例如(-1)::uint4 = 4294967295。
- 其他差异：
INT[(M)]精度，MySQL控制格式化输出，GaussDB仅语法支持，不支持功能。
- 聚集函数：
 - variance：GaussDB表示样本方差，MySQL表示总体方差。
 - stddev：GaussDB表示样本标准差，MySQL表示总体标准差。
- 显示宽度：
 - 在为整型数字列指明宽度信息时，如果不同时指定ZEROFILL，则宽度信息在表结构描述中不显示。
 - INSERT语句插入字符类型字段，GaussDB统一补齐0后插入。

- JOIN USING语句，涉及类型推导，MySQL默认第一张表列，GaussDB若结果为有符号类型则宽度信息失效，否则为第一张表字段宽度。
- greatest/least、ifnull/if、case when/decode，MySQL不补齐0，GaussDB在类型及宽度信息一致时补齐0。
- 作为函数/存储过程出入参、返回值时，MySQL支持功能、GaussDB语法不报错功能不支持。

GaussDB数据库和MySQL数据库整数类型具体差异请参见[表3-36](#)。

表 3-36 整数类型

MySQL数据库	GaussDB数据库	差异
BOOL	支持，存在差异	MySQL：BOOL/BOOLEAN类型实际映射为TINYINT类型。 GaussDB：支持BOOL，其中： “真”值的有效文本值是：TRUE、't'、'true'、'y'、'yes'、'1'、'TRUE'、true、'on'以及所有非0数值。 “假”值的有效文本值是：FALSE、'f'、'false'、'n'、'no'、'0'、0、'FALSE'、false、'off'。 使用TRUE和FALSE是比较规范的做法（也是SQL兼容的做法）。
BOOLEAN	支持，存在差异	
TINYINT[(M)] [UNSIGNED]	支持，存在差异	详情请参见 整数类型公共差异说明 。
SMALLINT[(M)] [UNSIGNED]	支持，存在差异	详情请参见 整数类型公共差异说明 。
MEDIUMINT[(M)] [UNSIGNED]	支持，存在差异	MySQL存储MEDIUMINT数据需要3字节。 - 带符号的范围是-8,388,608 ~ +8,388,607。 - 无符号的范围是0 ~ +16,777,215。 GaussDB映射为INT类型，存储需要4字节。 - 带符号的范围是-2,147,483,648 ~ +2,147,483,647。 - 无符号的范围是0 ~ +4,294,967,295。 其他差异请参见 整数类型公共差异说明 。
INT[(M)] [UNSIGNED]	支持，存在差异	详情请参见 整数类型公共差异说明 。
INTEGER[(M)] [UNSIGNED]	支持，存在差异	详情请参见 整数类型公共差异说明 。
BIGINT[(M)] [UNSIGNED]	支持，存在差异	详情请参见 整数类型公共差异说明 。

任意精度类型

表 3-37 任意精度类型

MySQL数据库	GaussDB数据库	差异
DECIMAL[(M[,D])]	支持，存在差异	- 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 - 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 - 输入格式：当字符串入参全部被截断时不会报错，如“@123”；只有被部分截断时才会报错，如“12@3”。
NUMERIC[(M[,D])]	支持，存在差异	
DEC[(M[,D])]	支持，存在差异	
FIXED[(M[,D])]	不支持	-

浮点类型

表 3-38 浮点类型

MySQL数据库	GaussDB数据库	差异
FLOAT[(M,D)]	支持，存在差异	- 分区表支持：FLOAT数据类型不支持KEY键值分区策略分区表。 - 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 - 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 - 输出格式：对于非法入参一律报错ERROR，不会在sql_mode="的宽松模式下报WARNING。
FLOAT(p)	支持，存在差异	- 分区表支持：FLOAT数据类型不支持KEY键值分区策略分区表。 - 操作符：数值类型使用^操作符，与MySQL不一致，GaussDB中^操作符为取指数运算。 - 取值范围：定义精度p时，仅支持使用合法的整型数据类型。 - 输出格式：对于非法入参一律报错ERROR，不会在sql_mode="的宽松模式下报WARNING。

MySQL数据库	GaussDB数据库	差异
DOUBLE[(M,D)]	支持，存在差异	- 分区表支持：DOUBLE数据类型不支持KEY键值分区策略分区表。 - 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 - 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 - 输出格式：对于非法入参一律报错ERROR，不会在sql_mode="的宽松模式下报WARNING。
DOUBLE PRECISION[(M,D)]	支持，存在差异	- 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 - 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 - 输出格式：对于非法入参一律报错ERROR，不会在sql_mode="的宽松模式下报WARNING。
REAL[(M,D)]	支持，存在差异	- 分区表支持：REAL数据类型不支持KEY值分区策略分区表。 - 操作符：GaussDB中“^”表示指数运算，如需使用异或运算符，使用“#”替换；MySQL中“^”表示异或。 - 取值范围：精度M，标度D不支持浮点型数值输入，只支持整型数值输入。 - 输出格式：对于非法入参一律报错ERROR，不会在sql_mode="的宽松模式下报WARNING。

序列整数

表 3-39 序列整数

MySQL数据库	GaussDB数据库	差异
SERIAL	支持，存在差异	GaussDB中SERIAL具体介绍请参见《开发指南》手册中的“SQL参考 > 数据类型 > 数值类型”章节。 规格上与MySQL的差异如下： <pre>CREATE TABLE test(f1 serial, f2 CHAR(20));</pre> - 类型定义差异，MySQL的serial是映射到BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE，GaussDB的serial是映射到INTEGER NOT NULL DEFAULT nextval('test_f1_seq'::regclass)。如： -- MySQL serial的定义： mysql> SHOW CREATE TABLE test\G ***** 1. row ***** Table: test Create Table: CREATE TABLE `test` (`f1` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `f2` char(20) DEFAULT NULL, UNIQUE KEY `f1` (`f1`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 1 row in set (0.00 sec) -- GaussDB serial的定义 gaussdb=# \d+ test Table "public.test" Column Type Modifiers Storage Stats target Description -----+----- +-----+-----+ f1 integer not null default nextval('test_f1_seq'::regclass) plain f2 character(20) extended Has OIDs: no Options: orientation=row, compression=no, storage_type=USTORE - INSERT场景下serial类型DEFAULT值的差异。 如： -- MySQL插入serial的DEFAULT值 mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa'); Query OK, 1 row affected (0.00 sec) mysql> INSERT INTO test VALUES(10, 'aaaa'); Query OK, 1 row affected (0.00 sec) mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa'); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; +----+-----+ f1 f2 +----+-----+ 1 aaaa 10 aaaa

MySQL数据库	GaussDB数据库	差异
		<pre> 11 aaaa +----+-----+ 3 rows in set (0.00 sec) -- GaussDB插入serial的DEFAULT值 gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa'); INSERT 0 1 gaussdb=# INSERT INTO test VALUES(10, 'aaaa'); INSERT 0 1 gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa'); INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2 -----+----- 1 aaaa 2 aaaa 10 aaaa (3 rows)</pre> REPLACE场景下serial类型引用列的差异，GaussDB引用列的介绍请参见《开发指南》手册中的“SQL参考 > SQL语法 > R > REPLACE”章节。如：<pre>-- MySQL插入serial引用列的值 mysql> REPLACE INTO test VALUES(f1, 'aaaa'); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, 'bbbb'); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; +----+-----+ f1 f2 +----+-----+ 1 aaaa 2 bbbb +----+-----+ 2 rows in set (0.00 sec) -- GaussDB插入serial引用列的值 gaussdb=# REPLACE INTO test VALUES(f1, 'aaaa'); REPLACE 0 1 gaussdb=# REPLACE INTO test VALUES(f1, 'bbbb'); REPLACE 0 1 gaussdb=# SELECT * FROM test; f1 f2 -----+----- 0 aaaa 0 bbbb (2 rows)</pre>

3.3.1.2 日期与时间数据类型

表 3-40 日期与时间数据类型

MySQL数据库	GaussDB数据库	差异
DATE	支持，存在差异	GaussDB支持date数据类型，与MySQL相比规格上存在如下差异： ● 输入格式 - GaussDB只支持字符类型，不支持数值类型。如支持'2020-01-01'或'20200101'字符串格式，不支持20200101数值输入。MySQL支持数值输入转换为date类型。 - 分隔符：GaussDB不支持加号“+”、冒号“:”作为年、月、日之间的分隔符，其他符号都支持。MySQL所有符号均可作为分隔符。分隔符混合使用的某些场景也不支持，与MySQL也有差异，如'2020-01>01'，'2020/01+01'等，不建议混合使用分隔符，建议使用最常用的“-”、“/”作为分隔符。 - 无分隔符：推荐使用完整格式，如'YYYYMMDD'或者'YYMMDD'。其他不完整的格式（包括超长格式）解析的规则与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 ● 输出格式 GaussDB在sql_mode参数不包含'strict_trans_tables'选项（定义为宽松模式，否则为严格模式）时，允许年、月、日的值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值得，如date '0000-00-10'转换为：0002-12-10 BC。非法输入或者超过范围时，会报warning信息，并返回0000-00-00值。MySQL对于包含0值年、月、日的date值会原样输出。 ● 取值范围 GaussDB的范围是4713-01-01 BC ~ 5874897-12-31 AD，支持公元前的日期，宽松模式下超过范围时，返回的是0值：0000-00-00，严格模式下会报错。MySQL的范围是 0000-00-00 ~ 9999-12-31，宽松模式下超过范围后，各个场景下的表现并不一致，可能报错（如SELECT查询语句中），也可能返回0000-00-00值（如INSERT时）。此差异会导致date类型作为函数入参时，函数返回的结果存在差异。 ● 操作符

MySQL数据库	GaussDB数据库	差异
		- GaussDB仅支持date类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；date与interval类型的加法运算，返回结果为date类型；date与interval类型的减法运算，返回结果为date类型；date类型之间的减法运算，返回结果为interval类型。 - MySQL date类型和其他数值类型运算时，会先将date转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如： <pre> -- MySQL: date + 数值, 先将date类型转换为数值 20200101,再与1相加, 结果为数值类型20200102 mysql> SELECT date'2020-01-01' + 1; +-----+ date'2020-01-01' + 1 +-----+ 20200102 +-----+ 1 row in set (0.00 sec) -- GaussDB: date + 数值, 数值类型会转换为interval类型 1 day, 然后相加得到新的日期。 gaussdb=# SELECT date'2020-01-01' + 1; ?column? ----- 2020-01-02 (1 row) </pre> ● 类型转换 相比较MySQL，GaussDB仅支持date类型与char(n)、nchar(n)、datetime、timestamp类型之间的相互转换，不支持与binary、decimal、json、integer、unsigned integer、time 类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考数据类型转换章节的描述。

MySQL数据库	GaussDB数据库	差异
DATETIME[(fsp)]	支持，存在差异	GaussDB支持datetime数据类型，与MySQL相比规格上存在如下差异： ● 输入格式 - GaussDB只支持字符类型，不支持数值类型。如支持'2020-01-01 10:20:30.123456'或'20200101102030.123456'字符串格式，不支持如20200101102030.123456的数值类型输入。MySQL支持数值输入转换为datetime类型。 - 分隔符：GaussDB不支持加号“+”、冒号“:”作为年、月、日之间的分隔符，其他的符号都支持。仅支持冒号“:”作为时、分、秒之间的分隔符，其他的符号都不支持。分隔符混合使用的某些场景也不支持，与MySQL也有差异，不推荐使用。MySQL支持所有符号作为分隔符。 - 无分隔符：GaussDB推荐使用完整格式'YYYYMMDDhhmiss.ffffff'。其他不完整的格式（包括超长格式）解析的规则可能与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 ● 输出格式： - 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的格式，格式与MySQL无差异，且不受DateStyle参数的影响。但是对于精度部分，如果最后几位为0，GaussDB不显示，MySQL会显示。 - GaussDB在sql_mode参数不包含'strict_trans_tables'选项（定义为宽松模式，否则为严格模式）时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值得，如datetime '0000-00-10 00:00:00' 转换为：0002-12-10 00:00:00 BC。非法输入或者超过范围时，会报warning信息，并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的datetime值会原样输出。 ● 取值范围 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。 294277-01-09 04:00:54.775807 AD 返回的是infinity。对于超过范围的值，严格模式下GaussDB会报错，MySQL是否报错取决于使用场景。一般查询场景不报错，而执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00值，MySQL根据使用场景可能报错，也可能返回

MySQL数据库	GaussDB数据库	差异
		0000-00-00 00:00:00值或者null值。这个差异会导致以datetime类型为入参的函数执行结果与MySQL也存在差异。 ● 精度 范围0~6，作为表列的类型时缺省为0，与MySQL一致。对于 datetime[(p)] 'str' 表达式场景，GaussDB将(p)作为精度解析，缺省为6，将'str'按照p指定的精度格式化成datetime类型。MySQL不支持datetime[(p)] 'str'表达式。 ● 操作符 - GaussDB仅支持datetime类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；datetime与interval类型的加法运算，返回结果为datetime类型；datetime与interval类型的减法运算，返回结果为datetime类型；datetime类型之间的减法运算，返回结果为interval类型。 - MySQL datetime类型和其他数值类型运算时，会先将datetime转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如： <pre>-- MySQL: datetime + 数值, 先将datetime类型转换为数值 20201010123456,再与1相加, 结果为数值类型 20201010123457 mysql> SELECT cast('2020-10-10 12:34:56.123456' AS datetime) + 1; +-----+ cast('2020-10-10 12:34:56.123456' as datetime) + 1 +-----+ 20201010123457 +-----+ 1 row in set (0.00 sec)</pre> <pre>-- GaussDB: datetime + 数值, 数值类型会转换为interval类型 1 day, 然后相加得到新的datetime。 gaussdb=# SELECT cast('2020-10-10 12:34:56.123456' AS datetime) + 1; ?column? ----- 2020-10-11 12:34:56 (1 row)</pre> 将datetime类型与数值的运算结果作为函数的入参，可能导致函数的结果与MySQL也存在差异。 ● 类型转换 相比较MySQL，GaussDB仅支持datetime类型与char(n)、varchar(n)、timestamp类型之间的相互转换、datetime到date、time类型的转换（仅赋值和显式转换）。不支持与binary、decimal、json、integer、unsigned integer类

MySQL数据库	GaussDB数据库	差异
		型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考数据类型转换章节的描述。 ● 时区 GaussDB支持datetime值中携带时区信息（时区偏移或者时区名），如'2020-01-01 12:34:56.123456 +01:00' 或者 '2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。MySQL不支持（5.7版本不支持，8.0及之后的版本支持）。 ● GaussDB的datetime数据类型的表字段实际上会被转换为timestamp(p) without time zone 类型，查询表信息或者使用工具导出的表结构，其字段的数据类型显示的是timestamp(p) without time zone，而不是datetime。MySQL显示的是datetime(p)。

MySQL数据库	GaussDB数据库	差异
TIMESTAMP[(fsp)]	支持，存在差异	GaussDB支持timestamp数据类型，与MySQL相比规格上存在如下差异： ● 输入格式： - 只支持字符类型，不支持数值类型。如支持 '2020-01-01 10:20:30.123456' 或 '20200101102030.123456' 字符串格式，不支持如 20200101102030.123456 的数值类型输入。MySQL支持数值输入转换为 timestamp 类型。 - 分隔符：不支持加号 “+”、冒号 “:” 作为年、月、日之间的分隔符，其他的符号都支持。仅支持冒号 “:” 作为时、分、秒之间的分隔符，其他的符号都不支持。分隔符混合使用的某些场景也不支持，与MySQL也有差异，不推荐使用。MySQL支持所有符号作为分隔符。 - 无分隔符：推荐使用完整格式 'YYYYMMDDhhmiss.ffffff'。其他不完整的格式（包括超长格式）解析的规则可能与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 ● 输出格式： - 统一为 'YYYY-MM-DD hh:mi:ss.ffffff' 的格式，格式与MySQL无差异，且不受 DateStyle 参数的影响。但是对于精度部分，如果最后几位为0，GaussDB不显示，MySQL会显示。 - GaussDB在sql_mode参数不包含 'strict_trans_tables' 选项（定义为宽松模式，否则为严格模式）时，允许年、月、日值是0，但是输出时会按照年、月、日的顺序依次转换为合法的值，如timestamp '0000-00-10 00:00:00' 转换为：0002-12-10 00:00:00 BC。非法输入或者超过范围时，会报warning信息，并返回 0000-00-00 00:00:00 值。MySQL对于包含0值年、月、日的timestamp值会原样输出。 ● 取值范围： 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。 294277-01-09 04:00:54.775807 AD 返回的是 infinity。对于超过范围的值，严格模式下 GaussDB会报错，MySQL是否报错取决于使用场景。一般查询场景不报错，而执行DML SQL语句更改表属性的值时报错。宽松模式下 GaussDB返回0000-00-00 00:00:00值，MySQL根据使用场景可能报错，也可能返回

MySQL数据库	GaussDB数据库	差异
		0000-00-00 00:00:00值或者null值。这个差异会导致以timestamp类型为入参的函数执行结果与MySQL也存在差异。 - 精度： 范围0~6，作为表列的类型时缺省为0，与MySQL一致。对于 timestamp[(p)] 'str' 表达式场景： - GaussDB将(p)作为精度解析，缺省为6，将'str'按照p指定的精度格式化timestamp类型。 - MySQL将timestamp 'str'的含义与GaussDB一致，缺省精度也为6。但是将timestamp(p) 'str'解析为函数调用，p作为timestamp函数的入参，结果返回一个timestamp类型的值，'str'作为投影列的别名。 - 操作符： - GaussDB仅支持timestamp类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；timestamp与interval类型的加法运算，返回结果为timestamp类型；timestamp与interval类型的减法运算，返回结果为timestamp类型；timestamp类型之间的减法运算，返回结果为interval类型。 - MySQL timestamp类型和其他数值类型运算时，会先将timestamp转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如： <pre>-- MySQL: timestamp + 数值，先将timestamp类型转换为数值20201010123456.123456,再与1相加，结果为数值类型20201010123457.123456 mysql> SELECT timestamp '2020-10-10 12:34:56.123456' + 1; +-----+ timestamp '2020-10-10 12:34:56.123456' + 1 +-----+ 20201010123457.123456 +-----+ 1 row in set (0.00 sec)</pre> <pre>-- GaussDB: timestamp + 数值，数值类型会转换为interval类型 1 day，然后相加得到新的timestamp。 gaussdb=# SELECT timestamp '2020-10-10 12:34:56.123456' + 1; ?column? ----- 2020-10-11 12:34:56.123456 (1 row)</pre> 将timestamp类型与数值的运算结果作为函数的入参，可能导致函数的结果与MySQL也存在差异。

MySQL数据库	GaussDB数据库	差异
		- 类型转换： 相比较MySQL，GaussDB仅支持timestamp类型与char(n)、varchar(n)、datetime类型之间的相互转换、timestamp到date、time类型的转换（仅赋值和显式转换）。不支持与binary、decimal、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考数据类型转换章节的描述。 - 时区： GaussDB支持timestamp值中携带时区信息（时区偏移或者时区名），如'2020-01-01 12:34:56.123456 +01:00' 或者 '2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。如果更改服务器时区，timestamp类型的值输出时会转换为更改后时区的时间戳。MySQL不支持（5.7版本不支持，8.0及之后的版本支持）。 - GaussDB的timestamp数据类型的表字段实际上会被转换为timestamp(p) with time zone类型，查询表信息或者使用工具导出的表结构，其字段的数据类型显示的是timestamp(p) with time zone，而不是timestamp。MySQL显示的是timestamp(p)。

MySQL数据库	GaussDB数据库	差异
TIME[(fsp)]	支持，存在差异	GaussDB支持time数据类型，与MySQL相比规格上存在如下差异： ● 输入格式： - 只支持字符类型，不支持数值类型。如支持 '1 10:20:30'或'102030'字符串格式，不支持 102030数值输入。MySQL支持数值输入转换为time类型。 - 分隔符：GaussDB仅支持冒号“:”作为时、分、秒之间的分隔符，其他的符号都不支持。MySQL支持所有的符号作为分隔符。 - 无分隔符：推荐使用完整格式，如 'hhmiss.ffffff'。其他不完整的格式（包括超长格式）解析的规则可能与MySQL存在差异，可能报错或者解析的结果与MySQL不一致，不推荐使用。 - 分、秒、精度输入负数时，GaussDB数据库可能会忽略第一个负数开始的部分，涉及的部分解析为0，如：'00:00:-10' 解析结果为 '00:00:00'。也可能报错，如：'00:00:-10000' 会解析报错。取决于输入值的范围。而MySQL数据库统一报错。 ● 输出格式： - 统一为hh:mi:ss.ffffff的格式，格式与MySQL无差异。但是对于精度部分，如果最后几位为0，GaussDB不显示，MySQL会显示。 ● 取值范围： -838:59:59.000000 ~ 838:59:59.000000，与MySQL一致。对于超过范围的值，GaussDB在宽松模式下执行SELECT、INSERT、UPDATE等DML操作时，返回的值都是就近的边界值：-838:59:59或838:59:59。MySQL是查询时报错，DML操作返回的值才是就近边界值，场景上存在差异。此差异会导致time类型作为函数入参时，函数返回的结果也存在差异。 ● 精度： - 范围0~6，作为表列的类型时缺省为0，与MySQL一致。对于 time(p) 'str' 表达式场景，GaussDB将(p)作为精度解析，缺省为6，将'str'按照p指定的精度格式化成time类型。MySQL是解析为time函数，p是入参，'str'是投影列的别名。 ● 操作符： - GaussDB仅支持time类型之间的比较操作符“=”、“!=”、“<”、“<=”、“>”、“>=”，返回true或者false；time与

MySQL数据库	GaussDB数据 库	差异
		interval类型的加法运算，返回结果为time类型；time与interval类型的减法运算，返回结果为time类型；time类型之间的减法运算，返回结果为interval类型。 – MySQL time类型和其他数值类型运算时，会先将time转换为数值类型，然后按照数值类型运算，结果也为数值类型。与GaussDB存在差异。如： <pre>-- MySQL： time + 数值，先将time类型转换为数值123456，再与1相加，结果为数值类型123457 mysql> SELECT time '12:34:56' + 1; +-----+ time '12:34:56' + 1 +-----+ 123457 +-----+ 1 row in set (0.00 sec)</pre> -- GaussDB： time + 数值，数值类型会转换为interval类型1 day，然后相加得到新的time，由于是加了24小时，得到的仍然是12:34:56。 <pre>gaussdb=# SELECT time '12:34:56' + 1; ?column? ----- 12:34:56 (1 row)</pre> 将time类型与数值的运算结果作为函数的入参，可能导致函数的结果与MySQL也存在差异。 ● 类型转换： 相比较MySQL，GaussDB仅支持time类型与char(n)、nchar(n)类型之间的相互转换、datetime、timestamp到time类型的转换。不支持与binary、decimal、date、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考数据类型转换章节的描述。

MySQL数据库	GaussDB数据库	差异
YEAR[(4)]	支持，存在差异	GaussDB支持year数据类型，与MySQL相比规格上存在如下差异： - 操作符： - GaussDB仅支持year类型之间的比较操作符：=、!=、<、<=、>、>=。返回true或者false。 - GaussDB仅支持year类型与int4类型之间的算术操作符：+、-。返回整型值，MySQL是返回无符号整型值。 - 类型转换： 相比较MySQL，GaussDB仅支持year类型与int4类型的转换，仅支持int4、varchar、numeric、date、time、timestamp、timestampz类型到year类型的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致，参考数据类型转换章节的描述。
INTERVAL	支持，存在差异	GaussDB支持INTERVAL数据类型，但INTERVAL在MySQL中为表达式，同时存在以下差异： - 不支持字符串类型的日期输入作为运算，如：SELECT '2023-01-01' + interval 1 day。 - 不支持interval expr unit语法中，expr为负整数或浮点数的输入，如：SELECT date'2023-01-01' + interval -1 day。 - 不支持interval expr unit语法中，expr为运算表达式的输入，如：SELECT date'2023-01-01' + interval 4/2 day。 - interval表达式参与运算时，返回值固定为datetime类型，MySQL为datetime或date类型。运算的逻辑与原有GaussDB保持一致，与MySQL有差异。 - interval expr unit语法中，expr数值支持的范围会根据unit单位的不同有所差异，最大可支持的范围为[-2147483648, 2147483647]。超过范围时，严格模式报error，宽松模式报warning并返回0值。 - interval expr unit语法中，expr指定的字段数量大于unit预期的字段数量时，在严格模式，报error；在宽松模式，报warning并返回0值。如unit取值为DAY_HOUR，预期的字段数量为2，expr取值为'1-2-3'，字段数量为3。

3.3.1.3 字符串数据类型

表 3-41 字符串数据类型

MySQL数据库	GaussDB数据库	差异
CHAR[(M)]	支持，存在差异	● 输入格式 - GaussDB自定义函数参数和返回值不支持长度校验，存储过程参数不支持长度校验，同时也不支持在PAD_CHAR_TO_FULL_LENGTH打开时补齐正确的空格，MySQL支持。 - GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 语法 GaussDB的 Cast (expr as char) 语法无法根据输入的字符串长度转成对应的类型，只支持转成varchar类型。不支持cast(" as char) 和 cast(" as char(0))将空串转成char(0)类型。MySQL支持按长度转成对应的类型。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 - GaussDB除以0会报错，MySQL返回null。 - “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 - “^”：GaussDB表示次方幂，MySQL表示按位异或。

MySQL数据库	GaussDB数据库	差异
VARCHAR(M)	支持，存在差异	● 输入格式 - GaussDB的自定义函数参数和返回值不支持长度校验，存储过程参数不支持长度校验，MySQL支持。 - GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警，MySQL不支持。 - GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 - GaussDB除以0会报错，MySQL返回null。 - “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 - “^”：GaussDB表示次方幂，MySQL表示按位异或。
TINYTEXT	支持，存在差异	● 输入格式 - GaussDB中该类型的长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过255字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 - GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 - GaussDB除以0会报错，MySQL返回null。 - “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 - “^”：GaussDB表示次方幂，MySQL表示按位异或。

MySQL数据库	GaussDB数据库	差异
TEXT	支持，存在差异	● 输入格式 - GaussDB中该类型的长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过65535字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 - GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 - GaussDB除以0会报错，MySQL返回null。 - “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 - “^”：GaussDB表示次方幂，MySQL表示按位异或。
MEDIUMTEXT	支持，存在差异	● 输入格式 - GaussDB中该类型的长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过16777215字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 - GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 - GaussDB除以0会报错，MySQL返回null。 - “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 - “^”：GaussDB表示次方幂，MySQL表示按位异或。

MySQL数据库	GaussDB数据库	差异
LONGTEXT	支持，存在差异	● 输入格式 - GaussDB只支持不超过1GB的长度，MySQL支持4GB-1字节的长度。 - GaussDB不支持转义字符输入，不支持""双引号输入，MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值进行加、减、乘、除、求余计算，返回值是整型值，MySQL是返回浮点型。 - GaussDB除以0会报错，MySQL返回null。 - “~”：GaussDB返回负数，MySQL返回8字节无符号整数。 - “^”：GaussDB表示次方幂，MySQL表示按位异或。
ENUM('value 1','value2',...)	不支持	-
SET('value1','value2',...)	不支持	-

3.3.1.4 二进制数据类型

表 3-42 二进制数据类型

MySQL数据库	GaussDB数据库	差异
BINARY[(M)]	不支持	-
VARBINARY(M)	不支持	-

MySQL数据库	GaussDB数据库	差异
TINYBLOB	支持，存在差异	- 取值范围：GaussDB中该类型由BYTEA类型映射得来，长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过255字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 - 输入格式：不支持转义字符输入，不支持""双引号输入。 - 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 - 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“ ”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“ ”、“^”。
BLOB	支持，存在差异	- 取值范围：GaussDB中该类型由BYTEA类型映射得来，长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过65535字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 - 输入格式：不支持转义字符输入，不支持""双引号输入。 - 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 - 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“ ”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“ ”、“^”。

MySQL数据库	GaussDB数据库	差异
MEDIUMBLOB	支持，存在差异	- 取值范围：GaussDB中该类型由BYTEA类型映射得来，长度不能超过1GB，超过长度限制后会报错。MySQL中该类型不能超过16777215字节，超过长度限制后，在严格模式下会报错，在宽松模式下会对数据进行截断并告警。 - 输入格式：不支持转义字符输入，不支持""双引号输入。 - 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 - 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“ ”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“ ”、“^”。
LONGBLOB	支持，存在差异	- 取值范围：GaussDB中该类型由BYTEA类型映射得来，只支持不超过1GB的长度，具体范围参照BYTEA数据类型集中式和分布式规格。 - 输入格式：不支持转义字符输入，不支持""双引号输入。 - 输出格式：对于'\0'字符，查询结果表现为“\000”，使用JDBC驱动的getBytes接口获取表现为'\0'字符。 - 操作符：不支持算数运算符“+”、“-”、“*”、“/”、“%”；不支持常用逻辑运算符或、与、非（“ ”、“&&”、“!”）；不支持常用位运算符“~”、“&”、“ ”、“^”。
BIT[(M)]	不支持	-

3.3.1.5 JSON 数据类型

表 3-43 JSON 数据类型

MySQL数据库	GaussDB数据库	差异
JSON	支持，存在差异	- GaussDB数据库MySQL兼容性MYSQL模式中的JSON类型与GaussDB数据库原生的JSON类型行为一致，与MySQL行为差异较大，此处不再逐个列出。 - GaussDB数据库MySQL兼容性MYSQL模式中的JSON类型具体行为请参见《开发指南》中的“SQL参考 > 数据类型 > JSON/JSONB类型”章节。

3.3.1.6 数据类型支持的属性

表 3-44 数据类型支持的属性

MySQL数据库	GaussDB数据库
NULL	支持
NOT NULL	支持
DEFAULT	支持
ON UPDATE	支持
PRIMARY KEY	支持
CHARACTER SET name	支持
COLLATE name	支持

3.3.1.7 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换：

- 操作符（比较操作符、运算操作符等）的操作数的数据类型不一致。常见于查询条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新（包括INSERT、UPDATE、MERGE、REPLACE等）的目标列，数据的类型和列的定义类型不一致。
- 显式的类型转换：cast(expr AS datatype)，将expr表达式类型转换为datatype类型。
- 集合运算（UNION、MINUS、EXCEPT、INTERSECT）确定最终投影列的目标数据类型后，各个SELECT查询的投影列的类型和目标数据类型不一致。

- 其他表达式计算场景，根据不同表达式的数据类型，来决定用于比较或者最终结果的目标数据类型。
 - DECODE
 - CASE WHEN
 - `lexpr [ NOT ] IN (expr_list)`
 - BETWEEN AND
 - JOIN USING(a,b)
 - GREATEST和LEAST
 - NVL 和 COALESCE

GaussDB和MySQL数据库对于数据类型转换、转换的目标数据类型有着完全不同的规则。如下示例体现了两者处理的差异：

```
-- MySQL: in 执行结果为0，表示false。根据规则，会将'1970-01-01'与列表中的表达式依次比较，结果都为0，因此最终结果为0。
mysql> SELECT '1970-01-01' IN ('1970-01-02', 1, '1970-01-02');
+-----+
| '1970-01-01' in ('1970-01-02', 1, '1970-01-02') |
+-----+
| 0 |
+-----+

-- GaussDB: in 执行结果为true，与MySQL结果相反：根据规则选定的公共类型为int类型，因此将左表达式'1970-01-01'转换为int类型与列表中的表达式转换为int类型后的值依次比较。
-- '1970-01-01'和'1970-01-02'转换为int类型时都为1970（MySQL兼容性下，转换时遇到非法字符后忽略，将前面部分转换为int类型），比较结果为相等，因此返回结果为true。
gaussdb=# SELECT '1970-01-01' IN ('1970-01-02', 1::int, '1970-01-02') AS result;
result
-----
t
(1 row)
```

数据类型转换规则的差异：

- GaussDB数据库对于不同数据类型之间的转换规则有明确的定义：
 - 是否支持转换：pg_cast系统表中是否定义两种类型的转换路径，没有定义则不支持。
 - 支持转换的场景：支持任意场景转换、仅支持显式（cast表达式）转换、仅支持赋值时转换。不支持的场景下即使定义了转换路径，也不能进行数据类型转换。
- MySQL数据库支持任意两种数据类型之间做转换。

由于存在以上差异，基于MySQL数据库的应用程序向GaussDB数据库迁移时，SQL语句可能由于不支持不同数据类型之间的转换而报错。或者支持转换的场景下，转换的规则有差异导致SQL语句执行的结果不同。

推荐的做法是：SQL语句中尽量使用相同的数据类型做比较或者赋值等操作，避免因数据类型转换导致非预期结果或者性能损耗。

选择目标数据类型的规则差异：

对于有些场景，比较的数据类型或者返回的数据类型需要综合考虑多个表达式的类型才能确定。比如UNION运算中，不同SELECT语句中相同位置的投影列具有不同的数据类型，查询结果的最终数据类型，需要由各个SELECT语句投影列的数据类型共同确定。

确定目标数据类型的规则，GaussDB数据库和MySQL数据库存在体系上的差异。

- GaussDB数据库规则：
 - 操作符的操作数类型不一致时，并不是将操作数的类型统一转换为目标类型再计算。而是直接注册两个数据类型的操作符，操作符处理中定义两个不同类型的处理规则。此方式不存在类型隐式转换，但自定义的处理规则隐含了转换的操作。
 - 集合运算和表达式场景，确定目标数据类型的规则：
 - 如果所有类型都相同，则此类型即为目标类型。
 - 两个数据类型如果不同，检查数据类型是否属于同一种类的数据类型，如数值类型、字符类型、日期时间类型等。不属于同一种类的数据类型，无法确定目标类型，此时SQL语句执行会报错。
 - 对于category属性（在pg_type系统表中定义）相同的数据类型，具有preferred属性（在pg_type系统表中定义）的数据类型会被选为目标类型。或者操作数1能转换为操作数2（没有转换路径），而操作数2无法转换为操作数1或数值类型优先级小于操作数2，则选择操作数2作为目标类型。
 - 如果涉及到3个及以上的数据类型，确定目标类型的规则为：
`common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3)`，依次迭代处理，得到最终的结果。
 - 对于IN和NOT IN表达式，如果根据以上规则无法确认目标类型，会将lexpr与expr_list中每一个表达式单独按照等值操作符（=）逐个比较。
 - 精度的确定：以最终选定的表达式的精度作为最终结果。
- MySQL数据库规则：
 - 操作符的操作数类型不一致时，先按照如下规则确定目标类型。确定后将类型不一致的操作数转换成目标类型后再做处理。
 - 两个参数都是string类型，则都按照string类型比较。
 - 两个参数都是integer类型，则都按照integer类型比较。
 - 十六进制数值如果不与数值比较，则当做二进制字符串比较。
 - 一个参数是datetime/timestamp类型，另一个参数是常量，将常量转换为时间戳类型然后比较。
 - 如果其中一个参数是decimal类型，比较时使用的数据类型取决于另外一个参数。另外一个为decimal或者integer类型时，按照decimal类型；另外一个为其他类型，按照real类型比较。
 - 其他场景都转换为 real 类型后比较。
 - 集合运算和表达式场景，确定目标数据类型的规则如下：
 - 建立任意两个类型之间的目标类型矩阵。给定两个类型，通过矩阵即可以确定目标类型。
 - 如果涉及到3个及以上的数据类型，确定目标类型的规则为：
`common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3)`，依次迭代处理，得到最终的结果。

- 如果目标类型是integer类型，且各个表达式类型包含有符号和无符号的混合场景，则会将类型提升到更高精度的integer类型。符号的确定：所有表达式都是无符号时，结果才为无符号，否则结果为有符号。
- 精度确定：以表达式中的最大精度作为最终结果。

从以上规则可知：GaussDB和MySQL数据库在数据类型的转换规则上有很大差异，不能直接对比。在上述场景下，SQL语句的执行结果可能和MySQL数据库不一致。当前版本推荐各个表达式使用相同的类型，或提前使用cast转换成需要的类型来规避差异。

3.3.2 系统函数

3.3.2.1 流量控制函数

表 3-45 流量控制函数列表

MySQL数据库	GaussDB数据库	差异
IF()	支持，存在差异	• expr1入参仅支持bool类型。非bool类型入参若不能转换为bool类型则报错。 • 若expr2、expr3两个入参类型不同且两类型间不存在隐式转换函数则报错。 • 两个入参类型相同时，返回该入参类型。 • 若expr2、expr3两个入参类型分别为NUMERIC、STRING或TIME其中一个，GaussDB输出为TEXT类型，MySQL输出为VARCHAR类型。
IFNULL()	支持，存在差异	• 若expr1、expr2两个入参类型不同且两类型间不存在隐式转换函数则报错。 • 两个入参类型相同时，返回该入参类型。 • 若expr1、expr2两个入参类型范畴分别为NUMERIC、STRING或TIME其中一个，GaussDB输出为TEXT类型，MySQL输出为VARCHAR类型。 • 两个入参类型中一个为FLOAT4类型，另一个为NUMERIC范畴中任一类型，返回值为DOUBLE类型。MySQL其中一入参为FLOAT4，另一入参为TINYINT、UNSIGNED TINYINT、SMALLINT、UNSIGNED SMALLINT、MEDIUMINT、UNSIGNED MEDIUMINT、BOOL任一类型时，返回FLOAT4类型，第一个入参为FLOAT4，第二个入参为BIGINT或UNSIGNED BIGINT时，返回FLOAT类型。

MySQL数据库	GaussDB数据库	差异
NULLIF()	支持，存在差异	- GaussDB中NULLIF()类型推导遵从以下逻辑： - 如果两个参数的数据类型不同，且两个入参类型存在等值比较操作符，则返回对应等值操作符对应的左值类型，否则会对两个入参类型进行强制类型兼容。 - 若强制类型兼容后，存在等值比较操作符，则返回强制类型兼容后对应等值操作符的左值类型。 - 若强制类型兼容后，仍找不到对应等值操作符，则报错。 --两个入参类型存在等值比较操作符 gaussdb=# SELECT pg_typeof(nullif(1::int2, 2::int8)); pg_typeof ----- smallint (1 row) --两个入参类型不存在等值比较操作符，但在强制类型兼容后可以找到等值比较操作符 gaussdb=# SELECT pg_typeof(nullif(1::int1, 2::int2)); pg_typeof ----- bigint (1 row) --两个入参类型不存在等值比较操作符，且强制类型兼容后也不存在等值比较操作符 gaussdb=# SELECT nullif(1::bit, '1'::MONEY); ERROR: operator does not exist: bit = money LINE 1: SELECT nullif(1::bit, '1'::MONEY); ^ HINT: No operator matches the given name and argument type(s). You might need to add explicit type casts. CONTEXT: referenced column: nullif - MySQL输出类型仅与第一个入参类型有关： - 第一个入参为tinyint、smallint、mediumint、int、bool时，输出为int类型。 - 第一个入参为bigint时，输出为bigint类型。 - 第一个入参为unsigned tinyint、unsigned smallint、unsigned mediumint、unsigned int、bit时，输出为unsigned int类型。 - 第一个入参为unsigned bigint时，输出为unsigned bigint。 - 第一个入参为浮点型即float、double、real时，输出为double类型。 - 第一个入参类型为decimal或numeric类型时，输出为decimal类型。 - 第一个入参类型为时间类型或字符串类型即date、time、date、datetime、timestamp、char、varchar以及tinytext、enum、set时，输出为varchar类型。

MySQL数据库	GaussDB数据库	差异
		- 第一个入参类型为text、mediumtext、longtext时，输出为longtext类型。 - 第一个入参类型为tinyblob时，输出为varbinary类型。 - 第一个入参类型为mediumblob或longblob时，输出为longblob类型。 - 第一个入参为blob时，输出为blob类型。
ISNULL()	支持，存在差异	GaussDB中返回值为BOOLEAN类型的t或f，MySQL中返回值为INT类型的1或0。

3.3.2.2 日期和时间函数

以下为GaussDB数据库MySQL兼容性MYSQL模式中日期时间函数的公共说明，与MySQL行为一致。

- 函数入参为时间类型表达式的情况：
时间类型表达式主要包括text、datetime、date或time，但所有可以隐式转换为时间表达式的类型都可以作为入参，比如数字类型可以通过先隐式转化为text，再作为时间类型表达式生效。
但是，不同函数的具体生效情况会有所不同。例如：datediff函数仅计算日期之间的差值，因此时间表达式会被解析为日期；而timestampdiff函数在计算时间差值时，会根据unit参数来决定将时间表达式解析为date、time或datetime。
- 函数入参为无效日期的情况：
一般而言，日期时间函数支持date、datetime的范围和MySQL保持一致。date支持的范围为'0000-01-01'到'9999-12-31'，datetime支持的范围为'0000-01-01 00:00:00'到'9999-12-31 23:59:59'。虽然GaussDB支持的date、datetime范围大于MySQL，但是越界仍然算无效日期。
大部分时间函数对于入参为无效时间时，会告警并返回NULL，只有能通过cast正常转换的日期，才是正常合理的日期。
- 函数入参的分隔符场景：
对于时间函数，处理入参时会把所有非数字字符视作分隔符，然后根据数字所处的位置进行计算，推荐使用标准写法，年月日之间使用-分隔符，时分秒之间使用:分隔符，毫秒之前通过.来进行分隔。
易错场景：在MySQL兼容性MYSQL模式数据库中执行“SELECT timestampdiff(hour, '2020-03-01 00:00:00', '2020-02-28 00:00:00+08');”时，时间函数不会自动计算时区，所以此处+08并未识别为时区，而是+作为分隔符当作秒来进行计算。

GaussDB的日期时间函数的大部分功能场景与MySQL一致，但仍有差异，部分差异如下：

- 函数入参为NULL时，函数返回NULL，无WARNING或ERROR告警。这些函数包括：
from_days、date_format、str_to_date、datediff、timestampdiff、date_add、subtime、month、time_to_sec、to_days、to_seconds、dayname、

monthname、convert_tz、sec_to_time、addtime、adddate、date_sub、timediff、last_day、weekday、from_unixtime、unix_timestamp、subdate、day、year、weekofyear、dayofmonth、dayofyear、week、yearweek、dayofweek、time_format、hour、minute、second、microsecond、quarter、get_format、extract、makedate、period_add、timestampadd、period_diff、utc_time、utc_timestamp、maketime、curtime

示例：

```
gaussdb=# SELECT day(null);
day
-----
(1 row)
```

- 纯数字入参个别函数与MySQL有差异，不带引号的数字入参统一转成text入参来处理。

示例：

```
gaussdb=# SELECT day(19231221.123141);
WARNING: Incorrect datetime value: "19231221.123141"
CONTEXT: referenced column: day
day
-----
(1 row)
```

- 时间日期运算函数：adddate、subdate、date_add、date_sub。当运算后的结果为日期时，支持的范围为[0000-01-01, 9999-12-31]，当运算后的结果为日期时间时，支持的范围为[0000-01-01 00:00:00.000000, 9999-12-31 23:59:59.999999]，当运算后的结果超过支持的范围时，在严格模式下报ERROR，在宽松模式下报WARNING。另外，当运算后的日期结果在范围[0000-01-01, 0001-01-01]中时，GaussDB正常返回结果，MySQL返回'0000-00-00'。

示例：

```
gaussdb=# SELECT subdate('0000-01-01', interval 1 hour);
ERROR: Datetime function: datetime field overflow
CONTEXT: referenced column: subdate

gaussdb=# SELECT subdate('0001-01-01', interval 1 day);
subdate
-----
0000-12-31
(1 row)
```

- 对于日期和时间函数的date或datetime类型入参，含有0月或0日则为非法值，在严格模式下报ERROR；在宽松模式，当输入为字符串或数字时报WARNING，输入为date或datetime类型时视为上一年12月或上一月最后一日处理。

对于cast函数，转换为date、datetime时，严格模式下会报ERROR；宽松模式下不会报WARNING，而是视为上一年12月或上一月最后一日处理，需要注意此区别。MySQL对于包含0年、0月或0日的情况会原样输出。

示例：

```
gaussdb=# SELECT adddate('2023-01-00', 1);-- 严格模式
ERROR: Incorrect datetime value: "2023-01-00"
CONTEXT: referenced column: adddate

gaussdb=# SELECT adddate('2023-01-00', 1);-- 宽松模式
WARNING: Incorrect datetime value: "2023-01-00"
CONTEXT: referenced column: adddate
adddate
-----
```

```
(1 row)

gaussdb=# SELECT adddate(date'2023-00-00', 1);-- 宽松模式
          adddate
-----
2022-12-01
(1 row)
```

```
gaussdb=# SELECT cast('2023/00/00' AS date);-- 宽松模式
          date
-----
2022-11-30
(1 row)
```

```
gaussdb=# SELECT cast('0000-00-00' AS datetime);-- 宽松模式
          timestamp
-----
0000-00-00 00:00:00
(1 row)
```

- 若函数入参为numeric数据类型，在非法输入的情况下不会产生报错，会把入参当做0值处理。

示例：

```
gaussdb=# SELECT from_unixtime('aa');
          from_unixtime
-----
1970-01-01 08:00:00
(1 row)
```

- 最多保留6位小数，不保留后置都为0的小数。

示例：

```
gaussdb=# SELECT from_unixtime('1234567899.000000');
          from_unixtime
-----
2009-02-14 07:31:39
(1 row)
```

- 时间函数参数为字符串时，只保证年月日之间使用“-”分隔，时分秒之间使用“:”分隔时结果正确。

示例：

```
gaussdb=# SELECT adddate('20-12-12',interval 1 day);
          adddate
-----
2020-12-13
(1 row)
```

- 在MySQL中，当函数的返回值为varchar时，在GaussDB中，函数对应的返回值为text。

-- GaussDB中函数的返回值。

```
gaussdb=# SELECT pg_typeof(adddate('2023-01-01', 1));
          pg_typeof
-----
text
(1 row)
```

-- MySQL中函数的返回值。

```
mysql> CREATE VIEW v1 AS SELECT adddate('2023-01-01', 1);
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DESC v1;
```

Field	Type	Null	Key	Default	Extra
adddate('2023-01-01', 1)	varchar(29)	YES		NULL	

```
1 row in set (0.00 sec)
```

表 3-46 日期与和时间函数列表

MySQL数据库	GaussDB数据 库	差异
ADDDATE()	支持，存在差 异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见 INTERVAL差异说明 。

MySQL数据库	GaussDB数据库	差异
ADDTIME()	支持，存在差异	- MySQL对第二入参为DATETIME样式字符串返回NULL，GaussDB可以计算。 - 入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。 - MySQL中ADDTIME函数如果第一个参数是动态参数（例如在预准备语句中），则返回类型为TIME。否则，函数的解析类型派生自第一个参数的解析类型。GaussDB中ADDTIME函数的返回值规则如下： - 第一个入参为date，第二个入参为date，返回值为time。 - 第一个入参为date，第二个入参为text，返回值为text。 - 第一个入参为date，第二个入参为datetime，返回值为time。 - 第一个入参为date，第二个入参为time，返回值为time。 - 第一个入参为text，第二个入参为date，返回值为text。 - 第一个入参为text，第二个入参为text，返回值为text。 - 第一个入参为text，第二个入参为datetime，返回值为text。 - 第一个入参为text，第二个入参为time，返回值为text。 - 第一个入参为datetime，第二个入参为date，返回值为datetime。 - 第一个入参为datetime，第二个入参为text，返回值为text。 - 第一个入参为datetime，第二个入参为datetime，返回值为datetime。 - 第一个入参为datetime，第二个入参为time，返回值为datetime。 - 第一个入参为time，第二个入参为date，返回值为time。 - 第一个入参为time，第二个入参为text，返回值为text。 - 第一个入参为time，第二个入参为datetime，返回值为time。 - 第一个入参为time，第二个入参为time，返回值为time。
CONVERT_TZ()	支持	-

MySQL数据库	GaussDB数据库	差异
CURDATE()	支持	-
CURRENT_DATE(), CURRENT_DATE	支持	-
CURRENT_TIME(), CURRENT_TIME	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，其他均报错；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，超过6的整型值，会告警并按照精度6输出时间值；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
CURTIME()	支持，存在差异	GaussDB此函数输入字符串或者非整型值，会被隐式转成整型，然后再校验精度，[0,6]范围之外的会报错，范围之内会正常输出时间值；MySQL直接报错。GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，其他均报错；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
YEARWEEK()	支持	-
DATE_ADD()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见 INTERVAL差异说明 。
DATE_FORMAT()	支持	-
DATE_SUB()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见 INTERVAL差异说明 。
DATEDIFF()	支持	-
DAY()	支持	-
DAYNAME()	支持	-

MySQL数据库	GaussDB数据库	差异
DAYOFMONTH()	支持	-
DAYOFWEEK() ()	支持	-
DAYOFYEAR()	支持	-
EXTRACT()	支持	-
FROM_DAYS() ()	支持	-
FROM_UNIXTIME() ()	支持	-
GET_FORMAT() (,)	支持	-
HOUR()	支持	-
LAST_DAY() (,)	支持	-
LOCALTIME() (,) LOCALTIME	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，其他整型值直接报错；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
LOCALTIMESTAMP() (,) LOCALTIMESTAMP	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，超过6的整型值，会告警并按照精度6输出时间值；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
MAKEDATE() (,)	支持	-
MAKETIME() (,)	支持，存在差异	与MySQL相比，入参为NULL时，GaussDB不支持maketime函数自嵌套，MySQL支持。
MICROSECOND() (,)	支持	-
MINUTE() (,)	支持	-
MONTH() (,)	支持	-

MySQL数据库	GaussDB数据库	差异
MONTHNAME()	支持	-
NOW()	支持，存在差异	GaussDB的按精度输出的时间值（小数点后的值）是四舍五入的；MySQL是直接截断的。GaussDB按精度输出的时间值（小数点后的值）末尾0都不显示；MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值，作为返回时间的精度，超过6的整型值，会告警并按照精度6输出时间值；MySQL的精度值有效值是[0,6]，但是输入的整型值内部会对256求余（例257，会返回精度1的时间值）。
PERIOD_ADD()	支持，存在差异	当入参period或结果小于0时，GaussDB参考MySQL 8.0.x版本的表现，报错处理。MySQL 5.7会发生整数回绕，导致计算结果异常。
PERIOD_DIFF()	支持，存在差异	当入参或结果小于0时，GaussDB参考MySQL 8.0.x版本的表现，报错处理。MySQL 5.7会发生整数回绕，导致计算结果异常。
QUARTER()	支持	-
SEC_TO_TIME()	支持	-
SECOND()	支持	-
STR_TO_DATE()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是text，MySQL返回的是datetime、date。
SUBDATE()	支持，存在差异	此函数的表现会因为interval表达式的差异与MySQL有差异，具体可见 INTERVAL差异说明 。

MySQL数据库	GaussDB数据库	差异
SUBTIME()	支持，存在差异	- MySQL对第二入参为DATETIME样式字符串返回NULL，GaussDB可以计算。 - 入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。 - MySQL中SUBTIME函数如果第一个参数是动态参数（例如在预准备语句中），则返回类型为TIME。否则，函数的解析类型派生自第一个参数的解析类型。GaussDB中SUBTIME函数的返回值规则如下： - 第一个入参为date，第二个入参为date，返回值为time。 - 第一个入参为date，第二个入参为text，返回值为text。 - 第一个入参为date，第二个入参为datetime，返回值为time。 - 第一个入参为date，第二个入参为time，返回值为time。 - 第一个入参为text，第二个入参为date，返回值为text。 - 第一个入参为text，第二个入参为text，返回值为text。 - 第一个入参为text，第二个入参为datetime，返回值为text。 - 第一个入参为text，第二个入参为time，返回值为text。 - 第一个入参为datetime，第二个入参为date，返回值为datetime。 - 第一个入参为datetime，第二个入参为text，返回值为text。 - 第一个入参为datetime，第二个入参为datetime，返回值为datetime。 - 第一个入参为datetime，第二个入参为time，返回值为datetime。 - 第一个入参为time，第二个入参为date，返回值为time。 - 第一个入参为time，第二个入参为text，返回值为text。 - 第一个入参为time，第二个入参为datetime，返回值为time。 - 第一个入参为time，第二个入参为time，返回值为time。
SYSDATE()	支持，存在差异	MySQL入参整型值会按照一字节最大值255整数回绕，GaussDB不回绕。

MySQL数据库	GaussDB数据库	差异
YEAR()	支持	-
TIME_FORMAT()	支持	-
TIME_TO_SEC()	支持	-
TIMEDIFF()	支持	-
WEEKOFYEAR()	支持	-
TIMESTAMPADD()	支持	-
TIMESTAMPDIFF()	支持	-
TO_DAYS()	支持	-
TO_SECONDS()	支持	-
UNIX_TIMESTAMP()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是numeric，MySQL返回的是int。
UTC_DATE()	支持，存在差异	- MySQL支持无括号调用，GaussDB不支持。MySQL入参整型值会按照一字节最大值255整数回绕。 - MySQL入参只支持0-6整数，GaussDB支持可以隐式转换为0-6的输入。
UTC_TIME()	支持，存在差异	
UTC_TIMESTAMP()	支持，存在差异	
WEEK()	支持	-
WEEKDAY()	支持	-

3.3.2.3 字符串函数

表 3-47 字符串函数列表

MySQL数据库	GaussDB数据库	差异
BIN()	支持，存在差异	函数入参支持类型存在差异，GaussDB入参支持类型如下： • 整数类型：tinyint、smallint、mediumint、int、bigint • 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned • 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 • 浮点类型：float、real、double • 定点类型：numeric、decimal、dec • 布尔类型：bool
CONCAT()	支持，存在差异	无论参数的数据类型如何，concat返回值的数据类型始终为text；MySQL的concat在含有二进制类型参数时，返回值为二进制类型。
CONCAT_WS()	支持，存在差异	无论参数的数据类型如何，concat_ws返回值的数据类型始终为text；MySQL的concat_ws在含有二进制类型参数时，返回值为二进制类型，其他情况返回值为字符串类型。

MySQL数据库	GaussDB数据库	差异
ELT()	支持，存在差异	- 函数入参1支持类型存在差异，GaussDB入参1支持类型如下： - 整数类型：tinyint、smallint、mediumint、int、bigint - 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned - 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 - 浮点类型：float、real、double - 定点类型：numeric、decimal、dec - 布尔类型：bool - 函数入参2支持类型存在差异，GaussDB入参2支持类型如下： - 整数类型：tinyint、smallint、mediumint、int、bigint - 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned、bigint unsigned - 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext - 浮点类型：float、real、double - 定点类型：numeric、decimal、dec - 布尔类型：bool - 大对象类型：tinyblob、blob、mediumblob、longblob - 日期类型：datetime、timestamp、date、time
FIELD()	支持，存在差异	- 函数入参为在bigint最大值 ~ bigint unsigned 最大值范围内的数字，存在不兼容。 - 函数入参为浮点型float(m, d)、double(m, d)、real(m, d)时精度更高，存在不兼容。
FIND_IN_SET()	支持，存在差异	当数据库encoding = 'SQL_ASCII'时，不支持默认的大小写判断规则，即在用户不指定字符集规则的情况下，大写与小写区分判断。

MySQL数据库	GaussDB数据库	差异
INSERT()	支持，存在差异	- Int64类型传参有范围限制，一旦超出-9223372036854775808~9223372036854775807范围会直接报错，MySQL对数值类型传参范围无限制，异常会告警按照上限或下限数值处理。 - 字符串传参有限制，入参text类型字符串长度最大为2^30-5字节，入参bytea类型字符串长度最大为2^30-512字节。 - s1和s2任意参数为bytea类型时，涉及到结果出现非法字符的情况可能展示结果与MySQL有差异但是字符编码与MySQL是一致的。
LOCATE()	支持，存在差异	入参1为bytea类型，入参2为text类型时，GaussDB与MySQL行为存在差异。
MAKE_SET()	支持，存在差异	- bits参数为整型时，最大范围支持到int128，低于MySQL范围。 - bits参数为日期类型datetime、timestamp、date、time，由于时间类型转整型与MySQL存在差异，目前均未做支持。 - bit类型或bool类型由于此类数据类型GaussDB与MySQL存在差异，返回结果导致的差异为GaussDB与MySQL固有差异。bits入参为bool类型，str入参为bit类型与bool类型均不做支持。 - bits入参为字符串或文本类型时，仅支持纯整型数字形式，其他形式存在差异。且纯整型数字范围限制在bigint范围。 - str入参整型数值超过正负81个9，返回值与MySQL有差异。 - str入参当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，以科学计数法打印，此为固有差异。

MySQL数据库	GaussDB数据库	差异
QUOTE()	支持，存在差异	- 已知str字符串中含有“\z”，“\r”，“\%”，“_”，GaussDB未进行转义，与MySQL存在差异。斜线后跟部分数字也会引起差异，如“\563”。由转义字符引起的本函数与MySQL的差异，此为GaussDB与MySQL的转义字符差异。 - str字符串中的“\b”，输出结果表现形式与MySQL有差异。此为GaussDB与MySQL的固有差异 - str字符串中含有“\0”时，GaussDB由于UTF-8字符集不识别该字符，输入不成功。此为GaussDB与MySQL的固有差异 - str为bit或bool类型时，由于GaussDB与MySQL此类型目前有差异，暂不支持此类类型。 - GaussDB最大支持1GB数据传输，str入参长度最大支持536870908字节，函数返回结果字符串最大支持1GB。 - str入参整型数值超过正负81个9，返回值与MySQL有差异。 - str入参当以科学计数法表示时，GaussDB末尾0值会显示，MySQL不显示，以科学计数法打印，此为固有差异。
SPACE()	支持，存在差异	- GaussDB入参最大支持1073741818字节，超出返回空字符串。MySQL的入参默认最大支持4194304字节，超出告警。 - 函数入参支持类型存在差异，GaussDB入参支持类型如下： - 整数类型：tinyint、smallint、mediumint、int、bigint - 无符号整数类型：tinyint unsigned、smallint unsigned、int unsigned - 字符和文本类型：char、varchar、tinytext、text、mediumtext、longtext，仅支持纯数字整数字符串，且整数范围在bigint范围内。 - 浮点类型：float、real、double - 定点类型：numeric、decimal、dec - 布尔类型：bool
SUBSTR()	支持	-
SUBSTRING()	支持	-
SUBSTRING_INDEX()	支持	-

MySQL数据库	GaussDB数据库	差异
STRCMP()	支持，存在差异	- 函数入参支持类型存在差异，GaussDB入参支持类型如下： - 字符类型：char、varchar、nvarchar2、text - 二进制类型：bytea - 数值类型：tinyint [unsigned]、smallint [unsigned]、integer [unsigned]、bigint [unsigned]、float4、float8、numeric - 日期时间类型：date、time without time zone、datetime、timestampz - 对于数值类型中的浮点类型，由于连接参数设置不同，精度可能与M有差异，不建议使用该场景，或使用NUMERIC类型代替。
SHA()/SHA1()	支持	-
SHA2()	支持	-

3.3.2.4 强制转换函数

表 3-48 强制转换函数列表

MySQL数据库	GaussDB数据库	差异
CAST()	支持，存在差异	数据类型转换规则和支持的转换类型均以GaussDB支持的转换范围和规则为准。
CONVERT()	支持，存在差异	数据类型转换规则和支持的转换类型均以GaussDB支持的转换范围和规则为准。

3.3.2.5 加密函数

表 3-49 加密函数列表

MySQL数据库	GaussDB数据库	差异
AES_DECRYPT()	支持	-
AES_ENCRYPT()	支持	-

MySQL数据库	GaussDB数据库	差异
JSON_KEYS()	支持	-
JSON_LENGTH()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是int，MySQL返回的是bigint。
JSON_MERGE()	支持	-
JSON_OBJECT()	支持	-
JSON_QUOTE()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是JSON，MySQL返回的是varchar或者text。
JSON_REMOVE()	支持	-
JSON_REPLACE()	支持	-
JSON_SEARCH()	支持，存在差异	返回值与MySQL有差异，GaussDB返回的是text，MySQL返回的是JSON。
JSON_SET()	支持	-
JSON_TYPE()	支持，存在差异	数值类型的JSON值统一识别为number，与MySQL有差异。
JSON_UNQUOTE()	支持	-
JSON_VALID()	支持	-

3.3.2.7 聚合函数

表 3-51 聚合函数列表

MySQL数据库	GaussDB数据库	差异
GROUP_CONCAT()	支持，存在差异	- 当group_concat参数中同时有DISTINCT和ORDER BY语法时，所有ORDER BY后的表达式必须也在DISTINCT的表达式之中。 - group_concat(... order by 数字)不代表按照几个参数的顺序，数字只是一个常量表达式，相当于不排序。 - 无论入参的数据类型是什么，group_concat返回值的数据类型始终为text；MySQL的group_concat在含有二进制类型参数时，返回值为二进制类型，其他情况返回值为字符串类型，并且返回值长度大于512时，其数据类型为字符串大对象或二进制大对象。 - GUC参数group_concat_max_len有效范围是0-1073741823，最大值比MySQL小。
DEFAULT()	支持，存在差异	- 字段默认值为数组形式，GaussDB返回数组形式，MySQL不支持数组类型。 - GaussDB字段是隐藏列（比如xmin、cmin），default函数返回空值。 - GaussDB支持分区表、临时表、多表连接查询默认值。 - GaussDB支持查询列名包含字符串值节点（表示名称）和A_Star节点（表示出现“*”），如default(tt.t4.id)和default(tt.t4.*）。不合法的查询列名和A_Star节点，GaussDB和MySQL报错信息有差异。 - GaussDB创建字段默认值，没有检验字段类型的范围，使用default函数可能报错。 - 字段的默认值是函数表达式时，GaussDB的default函数返回建表时字段的default表达式的计算值。MySQL的default函数返回NULL。

3.3.2.8 数字操作函数

表 3-52 数字操作函数列表

MySQL数据库	GaussDB数据库	差异
log2()	支持，存在差异	• 小数位显示与MySQL存在差异，受GaussDB浮点数据类型限制，可通过参数 extra_float_digits控制小数位个数显示。 • 由于输入精度内部处理差异，GaussDB与MySQL会存在结果计算差异。 • 支持数据类型有： - 整数类型：bigint、int16、int、smallint、tinyint。 - 无符号整数类型：bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned。 - 浮点数类型：numeric、real。 - 字符串类型：character、character varying、clob、text，仅支持纯数字整数字符串。 - set类型。 - NULL空类型。
log10()	支持，存在差异	• 小数位显示与MySQL存在差异，受GaussDB浮点数据类型限制，可通过参数 extra_float_digits控制小数位个数显示； • 由于输入精度内部处理差异，GaussDB与MySQL会存在结果计算差异； • 支持数据类型有： - 整数类型：bigint、int16、int、smallint、tinyint。 - 无符号整数类型：bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned。 - 浮点数类型：numeric、real。 - 字符串类型：character、character varying、clob、text，仅支持纯数字整数字符串。 - set类型。 - NULL空类型。
RAND([seed])	支持，存在差异	受函数内部使用的随机数生成算法的影响，函数返回的随机数与MYSQL存在差异。

3.3.2.9 其他函数

表 3-53 其他函数列表

MySQL数据库	GaussDB数据库	差异
UUID()	支持	-
UUID_SHORT()	支持	-

3.3.3 操作符

GaussDB数据库兼容绝大多数MySQL的操作符，但存在部分差异。除特别说明外，MySQL兼容性MYSQL模式中的操作符行为默认为GaussDB原生行为。

表 3-54 操作符

MySQL数据库	GaussDB数据库	差异
安全等于 (<=>)	支持	-

MySQL数据库	GaussDB数据库	差异
[NOT] REGEXP	支持，存在差异	- 当设置GUC参数b_format_dev_version='s2'时，模式字符串pattern中有“\\a”、“\\d”、“\\e”、“\\n”、“\\Z”、“\\u”等转义字符时，匹配源字符串“\a”、“\d”、“\e”、“\n”、“\Z”、“\u”时，GaussDB行为与MySQL 5.7不一致，与MySQL 8.0一致。 - 当设置GUC参数b_format_dev_version='s2'时，GaussDB中“\b”可以与“\\b”匹配，MySQL会匹配失败。 - 模式字符串pattern非法入参，只存在右单括号“)”时，GaussDB会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 - 在de abc匹配序列de或abc的匹配规则，当 左右存在空值时，GaussDB不会报错，MySQL 5.7会报错，MySQL 8.0不会报错。 - 制表符“\t”正则匹配字符类[:blank:]，GaussDB可匹配，MySQL 5.7不能匹配，MySQL 8.0可匹配。 - GaussDB支持非贪婪模式匹配，即尽可能少的匹配字符，在部分特殊字符后加“?”问号字符，例如：“??, *, +?, {n}?, {n,}?, {n,m}?”。MySQL 5.7版本不支持非贪婪模式匹配，并报错：Got error 'repetition-operator operand invalid' from regexp。MySQL 8.0版本已经支持。 - 在binary字符集下，text类型、blob类型均会转换成bytea类型，由于REGEXP操作符不支持bytea类型，因此无法匹配。
[NOT] RLIKE	支持，存在差异	同[NOT] REGEXP。

3.3.4 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集，支持的范围如下。

表 3-55 字符集列表

MySQL数据库	GaussDB数据库
utf8mb4	支持
gbk	支持
gb18030	支持

 说明

目前GaussDB对不属于当前字符集的非法律字符未执行严格的编码逻辑校验，可能导致此类非法字符成功输入。而MySQL会校验报错。

3.3.5 排序规则

GaussDB数据库支持指定模式、表或列的排序规则，支持的范围如下。

 说明

- 排序规则差异说明：
- 当前仅有字符串类型、部分二进制类型支持指定排序规则，其他类型不支持指定排序规则，可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。MySQL中所有类型可以指定字符序，但除字符串、二进制类型其他排序规则无实际意义。
 - 当前排序规则仅支持在其对应字符集与库级字符集一致时可以指定。
 - utf8mb4字符集下默认字符序为utf8mb4_general_ci，与MySQL 5.7保持一致，utf8mb4_0900_ai_ci为utf8mb4的非默认字符序。
 - GaussDB中utf8和utf8mb4为同一个字符集。

表 3-56 排序规则列表

MySQL数据库	GaussDB数据库
utf8mb4_general_ci	支持
utf8mb4_unicode_ci	支持
utf8mb4_bin	支持
gbk_chinese_ci	支持
gbk_bin	支持
gb18030_chinese_ci	支持
gb18030_bin	支持
binary	支持
utf8mb4_0900_ai_ci	支持
utf8_general_ci	支持
utf8_bin	支持

3.3.6 SQL

3.3.6.1 DDL

表 3-57 DDL 语法兼容介绍

概述	详细语法说明	差异
建表和修改表时支持创建主键、UNIQUE索引	ALTER TABLE、CREATE TABLE	- GaussDB当前不支持 UNIQUE INDEX KEY index_name语法，使用 UNIQUE INDEX KEY index_name语法时会报错。MySQL支持。 - 当约束被建立为全局二级索引，SQL语句中指定using btree时，底层会建立为 ubtree。 - 当约束关联的表为ustore，且SQL语句中指定为using btree时，底层会建立为 ubtree。
支持前缀索引	CREATE INDEX	- 前缀长度不得超过2676，键值的实际长度受内部页面限制，若字段中含有多字节字符或者一个索引上有多个键，索引行长度可能会超限报错。 - CREATE INDEX语法中，不支持以下关键字作为前缀键的字段名称：COALESCE、EXTRACT、GREATEST、LEAST、LNNVL、NULLIF、NVL、NVL2、OVERLAY、POSITION、REGEXP_LIKE、SUBSTRING、TIMESTAMPDIFF、TREAT、TRIM、XMLCONCAT、XMLELEMENT、XMLEXISTS、XMLFOREST、XMLPARSE、XMLPI、XMLROOT、XMLSERIALIZE。 - 主键和唯一键索引中不支持前缀键。
支持指定字符集与排序规则	ALTER SCHEMA、ALTER TABLE、CREATE SCHEMA、CREATE TABLE	-

概述	详细语法说明	差异
创建分区表语法兼容	CREATE TABLE PARTITION	-
建表和修改表时支持指定表级和列级 comment	CREATE TABLE、ALTER TABLE	-
创建索引时支持指定索引级comment	CREATE INDEX	-

概述	详细语法说明	差异
交换普通表和分区表分区的数据	ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点： - MySQL表或分区使用 tablespace时，则无法进行分区和普通表数据的交换；GaussDB表或分区使用不同的tablespace时，仍可进行分区和普通表数据的交换。 - 对于列默认值，MySQL不会校验默认值，因此默认值不同时也可进行分区和普通表数据的交换；GaussDB会校验默认值，如果默认值不同，则无法进行分区和普通表数据的交换。 - MySQL在分区表或普通表上进行DROP列操作后，表结构仍然一致，则可进行分区和普通表数据的交换；GaussDB需要保证普通表和分区表的被删除列严格对齐才能进行分区和普通表数据的交换。 - MySQL和GaussDB的哈希算法不同，所以两者在相同的hash分区存储的数据可能不一致，导致最后交换的数据也可能不一致。 - MySQL的分区表不支持外键，普通表包含外键或其他表引用普通表的外键，则无法进行分区和普通表数据的交换；GaussDB的分区表支持外键，在两个表的外键约束一致时，则可进行分区和普通表数据的交换；GaussDB的分区表不带外键，普通表有其他表引用，如果分区表和普通表表一致，则可进行分区和普通表数据的交换。

概述	详细语法说明	差异
支持自增列	ALTER TABLE、CREATE TABLE	- 目前仅支持各个DN独立自增的本地自动增长列。 - 自动增长列建议为索引（非全局二级索引）的第一个字段，否则建表时产生警告，含有自动增长列的表进行某些操作时会产生错误，例如：ALTER TABLE EXCHANGE PARTITION。MySQL自动增长列必须为索引第一个字段。 - AUTO_INCREMENT = value语法，value必须为小于2^{127}的正数。MySQL不校验value。 - 当自增值已经达到字段数据类型的最大值时，继续自增将产生错误。MySQL有些场景产生错误或警告，有些场景仍自增为最大值。 - 不支持 innodb_autoinc_lock_mode系统变量，GaussDB的GUC参数 auto_increment_cache=0 时，批量插入自动增长列的行为与MySQL系统变量 innodb_autoinc_lock_mode=1相似。 - 自动增长列在导入数据或者进行Batch Insert执行计划的插入操作时，对于混合0、NULL和确定值的场景，如果产生错误，后续插入自增值不一定与MySQL完全一致。提供 auto_increment_cache参数，可以控制预留自增值的数量。 - 批量插入在不同执行计划下，自增的顺序、自增值预留数量可能与MySQL不完全相同。例如：“INSERT INTO table VALUES(...), (...),...” 由于要分布到不同DN，在某些执行计划中，DN获取不到即将插入的行数。提供

概述	详细语法说明	差异
		auto_increment_cache参数，可以控制预留自增值的数量。 • 并行导入或插入自动增长列触发自增时，每个并行线程预留的缓存值也只在其线程中使用，未完全使用完毕的话，也会出现表中自动增长列的值不连续的情况。并行插入产生的自增值结果无法保证与MySQL完全一致。 • SERIAL数据类型为原有的自增列，与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。 • 不允许auto_increment_offset的值大于auto_increment_increment的值，会产生错误。MySQL允许，并说明auto_increment_offset会被忽略。 • 在表有主键或索引的情况下，ALTER TABLE命令重写表数据的顺序与MySQL不一定相同，GaussDB按表数据存储顺序重写，MySQL会按主键或索引顺序重写，导致自增值的顺序可能不同。 • ALTER TABLE命令添加或修改自增列时，第一次预留自增值的数量是表统计信息中的行数，统计信息的行数不一定与MySQL一致。 • 在触发器或用户自定义函数中自增时，刷新last_insert_id返回值。MySQL不刷新。 • 对GUC参数auto_increment_offset和auto_increment_increment设置超出范围的值会产生错

概述	详细语法说明	差异
		误。MySQL会自动改为边界值。 ● 暂不支持last_insert_id函数。 ● 本地临时表暂不支持自动增长列。 ● sql_mode设置no_auto_value_on_zero参数，表定义的自动增长列为非NOT NULL约束，向表中插入数据不指定自动增长列的值时，GaussDB中自动增长列插入NULL值，且不触发自增；MySQL中自动增长列插入NULL值，触发自增。
支持删除表的主键约束	ALTER TABLE	-

概述	详细语法说明	差异
支持CREATE TABLE ... LIKE语法兼容	CREATE TABLE ... LIKE	- 在MySQL 8.0.16 之前的版本中，CHECK约束会被语法解析但功能会被忽略，表现为不复制CHECK约束，GaussDB支持复制CHECK约束。 - 对于set数据类型，在建表时，MySQL支持复制，GaussDB不支持复制。 - 对于主键约束名称，在建表时，MySQL所有主键约束名称固定为PRIMARY KEY，GaussDB不支持复制。 - 对于唯一键约束名称，在建表时，MySQL支持复制，GaussDB不支持复制。 - 对于CHECK约束名称，在建表时，MySQL 8.0.16 之前的版本无CHECK约束信息，GaussDB支持复制。 - 对于索引名称，在建表时，MySQL支持复制，GaussDB不支持复制。 - 在跨sql_mode模式建表时，MySQL受宽松模式和严格模式控制，GaussDB可能存在严格模式失效的情况。 例如：源表存在默认值“0000-00-00”，在“no_zero_date”严格模式下，GaussDB建表成功，且包含默认值“0000-00-00”，严格模式失效；而MySQL建表失败，受严格模式控制。 - 针对跨数据库创建表，MySQL支持，GaussDB不支持。 - 针对源表为临时表，创建非临时表，MySQL支持，GaussDB不支持。

概述	详细语法说明	差异
支持更改表名兼容语法	ALTER TABLE[IF EXISTS] tbl_name RENAME [TO AS =] new_tbl_name; RENAME {TABLE TABLES} tbl_name TO new_tbl_name [, tbl_name2 TO new_tbl_name2, ...];	● GaussDB的alter rename语法仅支持修改表名称功能操作，不能耦合其它功能操作。 ● GaussDB中，仅旧表名字段支持使用 schema.table_name格式，且新表名与旧表名将属于同一个Schema。 ● GaussDB的不支持新旧表跨Schema重命名操作；但如有权限，则可在当前Schema下修改其它Schema下表名称。 ● GaussDB的rename多组表的语法支持全为本地临时表的重命名，不支持本地临时表和非本地临时表组合的场景。

概述	详细语法说明	差异
支持增加子分区语法兼容	ALTER TABLE [IF EXISTS] { table_name [*] ONLY (table_name) } action [, ...]; action: move_clause exchange_clause row_clause merge_clause modify_clause split_clause add_clause drop_clause ilm_clause add_clause: ADD {{partition_less_than_item partition_start_end_item partition_list_item} PARTITION({partition_less_than_item partition_start_end_item partition_list_item}}}	- 不支持ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition1,...);语法添加多分区。 - 仅支持原有添加多分区语法：ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2[y1]), ...;。

3.3.6.2 DML

表 3-58 DML 语法兼容介绍

概述	详细语法说明	差异
DELETE支持ORDER BY和LIMIT	DELETE	-
UPDATE支持ORDER BY和LIMIT	UPDATE	-

概述	详细语法说明	差异
REPLACE INTO语法兼容	REPLACE	- 时间类型初始值的差异。例如: - MySQL不受严格模式和宽松模式的影响，可向表中插入时间0值，即：<pre>mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, f2, f3); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; +-----+-----+ +-----+-----+ + f1 f2 f3 +-----+-----+ +-----+-----+ + 0000-00-00 00:00:00 0000-00-00 00:00:00 0000-00-00 +-----+-----+ +-----+-----+ + 1 row in set (0.00 sec)</pre> - GaussDB在宽松模式下才可以成功插入时间0值，即<pre>gaussdb=# SET b_format_version = '5.7'; SET gaussdb=# SET b_format_dev_version = 's1'; SET gaussdb=# SET sql_mode = ''; SET gaussdb=# CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL) DISTRIBUTE BY HASH(f1); CREATE TABLE gaussdb=# REPLACE INTO test VALUES(f1, f2, f3); REPLACE 0 1 gaussdb=# SELECT * FROM test; f1 f2 f3 +-----+-----+ +-----+-----+ 0000-00-00 00:00:00 0000-00-00 00:00:00 0000-00-00 (1 row)</pre>

概述	详细语法说明	差异
		在严格模式下，则报错 date/time field value out of range: "0000-00-00 00:00:00"。 • BIT类型在有NOT NULL时初始值的差异。例如： – MySQL BIT类型的初始值为空串"，即：<pre>mysql> CREATE TABLE test(f1 BIT(3) NOT NULL); Query OK, 0 rows affected (0.01 sec) mysql> REPLACE INTO test VALUES(f1); Query OK, 1 row affected (0.00 sec) mysql> SELECT f1, f1 IS NULL FROM test; +-----+ f1 f1 is null +-----+ 0 0 +-----+ 2 rows in set (0.00 sec)</pre> – GaussDB BIT类型的初始值为NULL，则报错。<pre>gaussdb=# CREATE TABLE test(f1 int, f2 BIT(3) NOT NULL) DISTRIBUTE BY HASH(f1); CREATE TABLE gaussdb=# REPLACE INTO test VALUES(1, f2); ERROR: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (1, null).</pre>
SELECT支持指定多分区查询	SELECT	-
UPDATE支持指定多分区更新	UPDATE	-

概述	详细语法说明	差异
LOAD DATA导入数据功能	LOAD DATA	- LOAD DATA语法执行结果与MySQL严格模式一致，宽松模式暂未适配。 - IGNORE与LOCAL参数功能仅为当导入数据与表中数据存在冲突时，忽略当前冲突行数据功能和当文件中字段数小于指定表中列数时自动为其余列填充默认值功能，其余功能暂未适配。 - 指定LOCAL关键字，且文件路径为相对路径时，文件从二进制目录下搜索；不指定LOCAL关键字，且文件路径为相对路径时，文件从数据目录下搜索。 - 语法中指定分隔符，转义字符，分行符等符号时，若指定为单引号，将导致词法解析错误。 [(col_name_or_user_var [, col_name_or_user_var] ...)]指定列参数不支持重复指定列。 [FIELDS TERMINATED BY 'string']指定换行符不能与[LINES TERMINATED BY 'string']分隔符相同。 - 执行LOAD DATA语法写入表中的数据若无法转换为表中数据类型格式时报错。 - 指定列时，仅支持指定为列名，不支持指定为用户变量。 - LOAD DATA SET表达式中不支持指定列名计算。 - 若set表达式返回值类型与对应列类型之间不存在隐式转换函数则报错。 - LOAD DATA不支持INSERT、DELETE触发器。 - LOAD DATA只能用于表，不能用于视图。 - Windows下的文件与Linux环境下文件默认换行符存在差异，LOAD DATA无法识

概述	详细语法说明	差异
		别此场景会报错，建议用户导入时检查导入文件行结尾的换行符。

概述	详细语法说明	差异
INSERT IGNORE兼容	INSERT IGNORE	- GaussDB会返回降级后的错误信息，MySQL则会将降级后的错误信息记录到错误堆栈中，然后调用show warnings;命令查看。例如： - 时间类型的差异。例如： - GaussDB中date、datetime、timestamp默认零值。<pre>gaussdb=# CREATE TABLE test(f1 DATE NOT NULL, f2 DATETIME NOT NULL, f3 TIMESTAMP NOT NULL); gaussdb=# INSERT IGNORE INTO test VALUES(NULL, NULL, NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2 -----+----- 1970-01-01 1970-01-01 00:00:00 (1 row)</pre> - MySQL中date、datetime、timestamp默认零值。<pre>mysql> CREATE TABLE test(f1 DATE NOT NULL, f2 DATETIME NOT NULL, f3 TIMESTAMP NOT NULL); Query OK, 0 rows affected (0.00 sec) mysql> INSERT IGNORE INTO test VALUES(NULL, NULL, NULL); Query OK, 1 row affected, 3 warnings (0.00 sec)</pre>

概述	详细语法说明	差异
		<pre>mysql> show warnings; +-----+-----+ Level Code Message +-----+-----+ Warning 1048 Column 'f1' cannot be null Warning 1048 Column 'f2' cannot be null Warning 1048 Column 'f3' cannot be null +-----+-----+ 3 rows in set (0.00 sec)</pre> <pre>mysql> SELECT * FROM test; +-----+-----+ f1 f2 f3 +-----+-----+ 0000-00-00 0000-00-00 00:00:00 0000-00-00 00:00:00 +-----+-----+ 1 row in set (0.00 sec)</pre> - 由于GaussDB不支持MySQL的bit类型，因此忽略bit类型NOT NULL约束和插入的bit类型长度与定义不同的场景下不支持INSERT IGNORE错误降级。 - GaussDB中bit类型<pre>gaussdb=# CREATE TABLE test(f1 BIT(10) NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL); ERROR: Un-support feature DETAIL: ignore null for insert statement is not supported in column f1. gaussdb=# INSERT IGNORE INTO test VALUES('1010'); ERROR: bit string length 4 does not match type bit(10) CONTEXT: referenced column: f1</pre> - MySQL中bit类型<pre>mysql> CREATE TABLE test(f1 BIT(10) NOT NULL); Query OK, 0 rows affected (0.00 sec) mysql> INSERT IGNORE INTO test VALUES(NULL); Query OK, 1 row affected, 1</pre>

概述	详细语法说明	差异
		warning (0.00 sec) mysql> INSERT IGNORE INTO test VALUES('1010'); Query OK, 1 row affected, 1 warning (0.01 sec) • MySQL数据库时间类型指定精度时，插入时间零值会显示精度，GaussDB则不显示，例如： – GaussDB指定时间精度 gaussdb=# CREATE TABLE test(f1 TIME(3) NOT NULL, f2 DATETIME(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL,NULL,NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2 f3 -----+----- +-----+ 00:00:00 1970-01-01 00:00:00 1970-01-01 00:00:00 (1 row) – MySQL指定时间精度 mysql> CREATE TABLE test(f1 TIME(3) NOT NULL, f2 DATETIME(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL); Query OK, 0 rows affected (0.00 sec) mysql> INSERT IGNORE INTO test VALUES(NULL,NULL,NULL); Query OK, 1 row affected, 3 warnings (0.00 sec) mysql> SELECT * FROM test; +-----+ +-----+ +-----+

概述	详细语法说明	差异
		f1 f2 f3 +-----+ +-----+ +-----+ 00:00:00.000 0000-00-00 00:00:00.000 0000-00-00 00:00:00.000 +-----+ +-----+ +-----+ 1 row in set (0.00 sec) - 由于MySQL数据库和GaussDB执行过程的差异，因此，产生的warnings条数可能不同，例如： - GaussDB产生的warnings条数<pre>gaussdb=# CREATE TABLE test(f1 INT, f2 INT not null); CREATE TABLE gaussdb=# INSERT INTO test VALUES(1,0),(3,0),(5,0); INSERT 0 3 gaussdb=# INSERT IGNORE INTO test SELECT f1+1, f1/f2 FROM test; WARNING: division by zero CONTEXT: referenced column: f2 WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (2, null). WARNING: division by zero CONTEXT: referenced column: f2 WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (4, null). WARNING: division by zero CONTEXT: referenced column: f2 WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (6, null). INSERT 0 3</pre> - MySQL产生的warnings条数<pre>mysql> CREATE TABLE test(f1 INT, f2 INT not null); Query OK, 0 rows affected (0.01 sec) mysql> INSERT INTO test VALUES(1,0),(3,0),(5,0);</pre>

概述	详细语法说明	差异
		Query OK, 3 rows affected (0.00 sec) Records: 3 Duplicates: 0 Warnings: 0 mysql> INSERT IGNORE INTO test SELECT f1+1, f1/f2 FROM test; Query OK, 3 rows affected, 4 warnings (0.00 sec) Records: 3 Duplicates: 0 Warnings: 4 ● MySQL数据库和GaussDB INSERT IGNORE在触发器 中的差异，例如： – GaussDB触发器中使用 INSERT IGNORE gaussdb=# CREATE TABLE test1(f1 INT NOT NULL); CREATE TABLE gaussdb=# CREATE TABLE test2(f1 INT); CREATE TABLE gaussdb=# CREATE OR REPLACE FUNCTION trig_test() RETURNS TRIGGER AS \$\$ gaussdb\$\$ BEGIN gaussdb\$\$ INSERT IGNORE INTO test1 VALUES(NULL); gaussdb\$\$ RETURN NEW; gaussdb\$\$ END; gaussdb\$\$ \$\$ LANGUAGE plpgsql; CREATE FUNCTION gaussdb=# CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW EXECUTE PROCEDURE trig_test(); CREATE TRIGGER gaussdb=# INSERT INTO test2 VALUES(NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null). CONTEXT: SQL statement "INSERT IGNORE INTO test1 VALUES(NULL)" PL/pgSQL function trig_test() line 3 at SQL statement INSERT 0 1 gaussdb=# SELECT * FROM test1; f1 ---- 0 (1 rows) gaussdb=# SELECT * FROM test2;

概述	详细语法说明	差异
		f1 ----- (1 rows) - MySQL触发器中使用 INSERT IGNORE mysql> CREATE TABLE test1(f1 INT NOT NULL); Query OK, 0 rows affected (0.01 sec) mysql> CREATE TABLE test2(f1 INT); Query OK, 0 rows affected (0.00 sec) mysql> DELIMITER mysql> CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW -> BEGIN -> INSERT IGNORE into test1 values(NULL); -> END Query OK, 0 rows affected (0.01 sec) mysql> DELIMITER ; mysql> INSERT INTO test2 VALUES(NULL); ERROR 1048 (23000): Column 'f1' cannot be null mysql> INSERT IGNORE INTO test2 VALUES(NULL); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test1; +-----+ f1 +-----+ 0 +-----+ 1 row in set (0.00 sec) mysql> SELECT * FROM test2; +-----+ f1 +-----+ NULL +-----+ 1 row in set (0.00 sec) • GaussDB的bool、serial的实现机制与MySQL不同，因此其默认零值与MySQL不同，例如： - GaussDB的行为 gaussdb=# CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL); NOTICE: CREATE TABLE will create implicit sequence

概述	详细语法说明	差异
		"test_f1_seq" for serial column "test.f1" CREATE TABLE gaussdb=# INSERT IGNORE INTO test values(NULL,NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2 ----+---- 0 f (1 row) - MySQL的行为 mysql> CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL); Query OK, 0 rows affected (0.00 sec) mysql> INSERT IGNORE INTO test values(NULL,NULL); Query OK, 1 row affected, 1 warning (0.00 sec) mysql> SELECT * FROM test; +-----+ f1 f2 +-----+ 1 0 +-----+ 1 row in set (0.00 sec)

3.3.6.3 DCL

表 3-59 DCL 语法兼容介绍

概述	详细语法说明	差异
SET NAMES指定 COLLATE子句	SET [SESSION LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] DEFAULT};	GaussDB中暂不支持指定 charset_name与数据库字符集 不同。具体请参见《开发指 南》中“SQL参考 > SQL语法 > S > SET ” 章节。

3.3.7 驱动

3.3.7.1 JDBC

3.3.7.1.1 JDBC 接口参考

GaussDB与MySQL的JDBC接口定义一致，均遵循业界规范，本章节主要介绍GaussDB数据库的MySQL兼容性MYSQL模式与MySQL数据库JDBC接口的行为差异。

获取结果集中的数据

ResultSet对象提供了丰富的方法，以获取结果集中的数据。获取数据常用的方法如表3-60所示，其他方法请参考JDK官方文档。

表 3-60 ResultSet 对象的常用方法

方法	描述	差异
int getInt(int columnIndex)	按列标获取int型数据。	-
int getInt(String columnLabel)	按列名获取int型数据。	-
String getString(int columnIndex)	按列标获取String型数据。	字段类型为整型且带有ZEROFILL属性时，GaussDB按照ZEROFILL属性要求的宽度信息用0进行补位后输出结果，MySQL直接输出结果。
String getString(String columnLabel)	按列名获取String型数据。	字段类型为整型且带有ZEROFILL属性时，GaussDB按照ZEROFILL属性要求的宽度信息用0进行补位后输出结果，MySQL直接输出结果。
Date getDate(int columnIndex)	按列标获取Date型数据。	-
Date getDate(String columnLabel)	按列名获取Date型数据。	-