Data Warehouse Service

Product Bulletin

Issue	04
Date	2025-01-08

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Product Notice

1.1 Checking the GaussDB(DWS) Version

The cluster version of GaussDB(DWS) displayed on the management console is in the format of x.y.z.p, which is explained in **Figure 2-1**.

Generally, new functions and features of GaussDB(DWS) are released based on the release number. For example, 8.1.3 and 8.2.0 are iteration versions. New features are added to new versions. For details about the version lifecycle, see **Version Lifecycle**.

After each iteration version is released, a patch is released to fix problems, for example, patch 8.1.3.322. A patch only fixes problems and does not add new functions or features.

Figure 1-1 GaussDB(DWS) version description

Table 1-1 GaussDB(DWS) cluster version mapping

Released On	Cluster Version	Database Kernel Version	Description
2024-12-13	9.1.0.211	9.1.0	**New Features in 9.1.0.x**
2024-11-25	9.1.0.210		

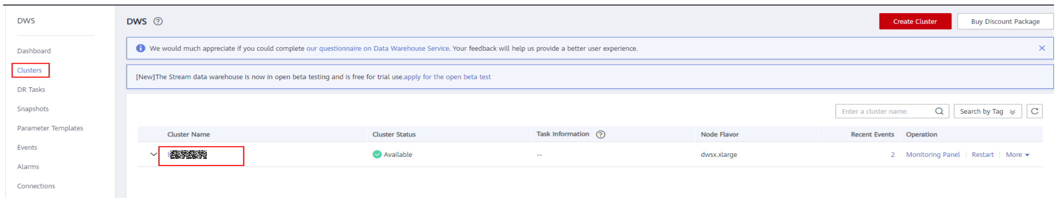
Released On	Cluster Version	Database Kernel Version	Description
2024-10-23	9.1.0.105		
2024-09-25	9.1.0.102		
2024-08-12	9.1.0.100		
2024-10-26	8.3.0.110	8.3.0	**Version 8.3.0**
2024-07-22	8.3.0.108		
2024-07-10	8.3.0.105 (available only to whitelisted users)		
2024-06-10	8.3.0.103 (available only to whitelisted users)		
2024-04-25	8.3.0.101 (available only to whitelisted users)		
2024-04-17	8.3.0.100		
2024-10-15	8.2.1.230 (available only to whitelisted users)	8.2.1	**Version 8.2.1**
2024-07-08	8.2.1.225 (available only to whitelisted users)		
2024-06-10	8.2.1.223 (available only to whitelisted users)		
2024-04-30	8.2.1.220 (available only to whitelisted users)		
2024-02-28	8.2.1.119 (available only to whitelisted users)		
2023-05-30	8.2.1.100 (available only to whitelisted users)		
2023-09-25	8.2.0.107	8.2.0	**Version 8.2.0**
2023-07-17	8.2.0.106		
2023-05-25	8.2.0.103		

Released On	Cluster Version	Database Kernel Version	Description
2023-04-15	8.2.0.102		
2023-03-13	8.2.0.101		
2023-01-14	8.2.0.100		
2022-11-28	8.2.0		
2024-10-15	**8.1.3.336 (recommended version)**	8.1.3	**New Features in 8.1.3**
2024-06-30	8.1.3.333		
2024-03-16	8.1.3.330		
2023-11-02	8.1.3.325		
2023-10-16	8.1.3.323		
2023-08-04	8.1.3.322		
2023-06-26	8.1.3.321		
2023-05-19	8.1.3.320		
2023-03-09	8.1.3.310		
2022-12-19	8.1.3.300		
2022-10-31	8.1.3.200		
2022-08-23	8.1.3.110		
2022-06-20	8.1.3.100		
2022-04-15	8.1.3		
2022-06-20	8.1.1.500	8.1.1	**Version 8.1.1**
2022-04-15	8.1.1.300		
2022-03-30	8.1.1.205		
2022-03-18	8.1.1.203		
2022-02-24	8.1.1.202		
2022-01-25	8.1.1.201		
2021-12-09	8.1.1.200		
2021-07-30	8.1.1.100		
2021-05-15	8.1.0.100	8.1.0	**Version 8.1.0**
2021-02-08	8.0.1.500	8.0.1	**Version 8.0.1**

Released On	Cluster Version	Database Kernel Version	Description
2020-07-31	8.0.1.100		
2020-04-28	1.7.2	8.0.0	-
2020-03-20	1.7.1		
2019-08-08	1.5.200	R8C10	-

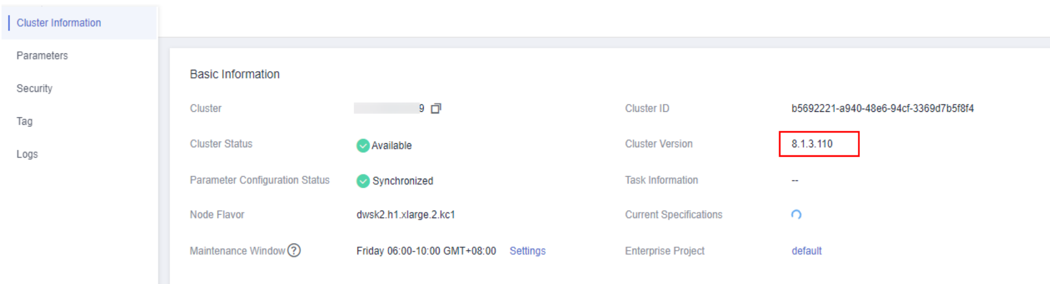
Querying the Cluster Version

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, Choose **Clusters**.



Step 2 In the cluster list, find the target cluster and click the cluster name. The **Basic Information** tab is displayed.

The cluster version is 8.1.3.110.

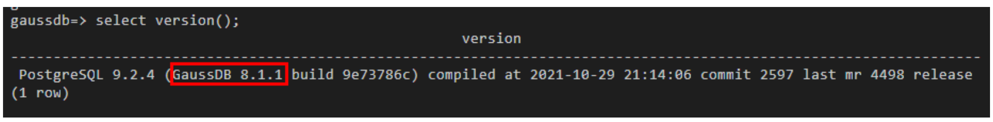


----End

Querying the Database Kernel Version

- Method 1: Query the cluster version by referring to **Querying the Cluster Version** and find the corresponding database kernel version in .
- Method 2: Connect to the GaussDB(DWS) database. Run the following SQL statement:

```
select version();
```



1.2 New Features in 9.1.0.x

NOTE

The beta features discussed below are not available for commercial use. Search for technical support before utilizing these features.

Patch 9.1.0.211 (December 13, 2024)

This is a patch version that fixes known issues.

Version 9.1.0.210 (November 25, 2024)

Storage-compute decoupling

1. You can use the **explain warmup** command to preload data into the local disk cache, either at the cold or hot end.
2. The enhanced elastic VW function offers more flexible ways to distribute services. Services can be distributed to either the primary VW or the elastic VW by CN.
3. Storage-compute decoupled tables support parallel insert operations, which improves data loading performance.
4. The storage-compute decoupled table has a recycle bin feature. This allows you to quickly recover from misoperations such as dropping or truncating a table or partition.
5. Both hot and cold tables can utilize disk cache and asynchronous I/Os to improve performance.

Real-time data warehouse

1. The performance for **limit...offset** page turning and **inlist** operations has been significantly improved.
2. The Binlog feature is now available for commercial use.
3. Automatic partitioning now supports time columns of both integer and variable-length types.

Lakehouse

1. Parquet/ORC read and write now support the zstd compression format.
2. The **create table like** command now allows using a table from an external schema as the source table.
3. Foreign tables can be exported in parallel.

High availability

1. Storage-compute decoupled tables and hot and cold tables support incremental backup and restoration.
2. In storage-compute decoupling scenarios, parallel copy is used to increase backup speed.

Ecosystem compatibility

1. The system is compatible with the replace into syntax of MySQL and the interval time type.
2. The **pg_get_tabledef** export function now displays comments.

O&M and stability improvement

1. When disk usage is high, data can be dumped from the standby node to OBS.
2. When the database is about to become read-only, certain statements that write to disks and generate new tables and physical files are intercepted to quickly reclaim disk space and ensure the execution of other statements.
3. Audit logs can be dumped to OBS.
4. The lightweight lock view **pgxc_lwlocks** is added.
5. The common lock view now includes lock acquisition and wait time stamps.
6. The global deadlock detection function is now enabled by default.
7. A lock function is added between VACUUM FULL and SELECT.
8. The expiration time has been added to **gs_view_invalid** to assist O&M personnel in clearing invalid objects.

Constraints

1. The maximum number of VWs supported is 256, with each VW supporting a maximum of 1,024 DNs. It is best to have no more than 32 VWs, with each VW containing no more than 128 DNs.
2. OBS storage-compute decoupled tables do not support DR or fine-grained backup and restoration.

Behavior changes

1. Enabling the **max_process_memory** adaptation during the upgrade and using the active/standby mode will increase the available memory of DNs.
2. By default, data consistency check is enabled for data redistribution during scale-out, which increases the scale-out time by 10%.
3. Create an **Hstore_opt** table with the Turbo engine enabled and retain the default value **middle** for the compression level.
4. By default, the OBS path of a storage-compute decoupled table is displayed as a relative path.
5. To use the disk cache, enable the asynchronous I/O parameter.
6. The interval for clearing indexes of column-store tables has been changed from 1 hour to 10 minutes to quickly clear the occupied index space.
7. **CREATE TABLE** and **ALTER TABLE** do not support columns with the **ON UPDATE** expression as distribution columns.
8. During Parquet data query, the timestamp data saved in INT96 format is not adjusted for 8 hours.
9. **max_stream_pool** is used to control the number of threads cached in the stream thread pool. The default value is changed from **65525** to **1024** to prevent idle threads from using too much memory.

10. The **track_activity_query_size** parameter takes effect upon restart instead of dynamically.
11. The logical replication function is no longer supported, and an error will be reported when related APIs are called.

Patch 9.1.0.105 (October 23, 2024)

This is a patch version that fixes known issues.

Patch 9.1.0.102 (September 25, 2024)

This is a patch version that fixes known issues.

Upgrade

1. Upgrade from 9.0.3 to 9.1.0 is supported.

Fixed known issues

1. Supported **alter database xxx rename to yyy** in the storage-compute decoupling version.
2. Fixed the problem of incorrect display of storage-compute decoupling table's \d+ space size.
3. Fixed the problem of asynchronous sorting not running post backup and restoration.
4. Fixed the problem of inability to use **Create Table Like** syntax after deleting the **bitmap index** column.
5. Fixed the performance rollback problem in Turbo engine's **group by** scenario caused by **hash** algorithm conflicts.
6. Maintained the scheduler processes' handling of failed tasks in the same manner as version 8.3.0.
7. Fixed the problem of **pg_stat_object** space expansion in fault scenario.
8. Fixed the problem of DataArts Studio reporting an error when delivering a **Vacuum Full** job after upgrading from 8.3.0 to 9.1.0.
9. Fixed the problem of high CPU and memory usage during **JSON** field calculation.

Enhanced functions

1. ORC foreign tables support the ZSTD compression format.
2. GIS supports the **st_asmvtgeom**, **st_asmvt**, and **st_squaregrid** functions.

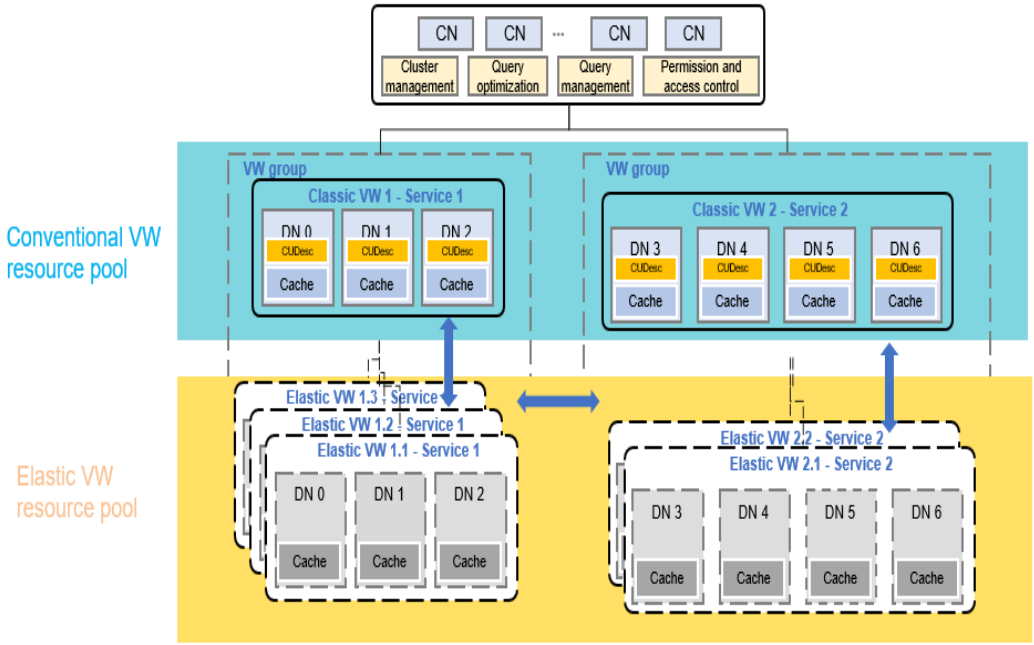
Version 9.1.0.100 (August 12, 2024)

Elastic architecture

1. Architecture upgrade: The storage-compute decoupling architecture 3.0, based on OBS, introduces layered and elastic computing and storage, with on-demand storage charging to reduce costs and improve efficiency. Multiple virtual warehouses (VWs) can be deployed to enhance service isolation and resolve resource contention.

2. The elastic VW feature, which is stateless and supports read/write acceleration, addresses issues like insufficient concurrent processing, unbalanced peak and off-peak hours, and resource contention for data loading and analytics. For details, see **Elastically Adding or Deleting a Logical Cluster**.
3. Both auto scale-out and classic scale-out are supported when adding or deleting DNs. Auto scale-out does not redistribute data on OBS, while classic scale-out redistributes all data. The system automatically selects the scale-out mode based on the total number of buckets and DNs.
4. The storage-compute decoupling architecture (DWS 3.0) enhances performance with disk cache and asynchronous I/O read/write. When the disk cache is fully utilized, performance matches that of the storage-compute integration architecture (DWS 2.0).

Figure 1-2 Decoupled storage and compute



Real-time processing

1. Launched the vectorized Turbo acceleration engine, doubling the performance of tpch 1000x.
2. Upgraded version of hstore, called hstore_opt, offers a higher compression ratio and works in conjunction with the Turbo engine to reduce storage space by 40% when compared to column storage.
3. With Flink, you can connect directly to DNs to import data into the database. This results in linear performance improvement in batch data import scenarios. For details, see **Real-Time Binlog Consumption by Flink**.
4. GaussDB(DWS) supports Binlog (currently in beta) and can be used in conjunction with Flink to enable incremental computing. For details, see **Subscribing to Hybrid Data Warehouse Binlog**.
5. This update significantly improves full-column performance while reducing resource consumption.

6. GaussDB(DWS) supports materialized views (currently in beta). For details, see **CREATE MATERIALIZED VIEW**.
7. To improve coarse filtering, the **Varchar/text** column now supports **bitmap index** and **bloom filter**. When creating a table, you must specify them explicitly. For details, see **CREATE TABLE**.
8. To enhance performance in topK and join scenarios, the **runtime filter** feature is now supported. You can learn more about GUC parameters **runtime_filter_type** and **runtime_filter_ratiox** in **Other Optimizer Options**.
9. GaussDB(DWS) supports asynchronous sorting to enhance the **min-max** coarse filtering effect of **PCK** columns.
10. The performance in the **IN** scenario is greatly improved.
11. **ANALYZE** supports incremental merging of partition statistics, collecting only statistics on changed partitions and reusing historical data, which improves execution efficiency. It collects statistics only on predicate columns.
    - The **CREATE TABLE** syntax now includes the **incremental_analyze** parameter to control whether to enable incremental ANALYZE mode for partitioned tables. For details, see **CREATE TABLE**.
    - The **enable_analyze_partition** GUC parameter determines whether to collect statistics on a partition of a table. For details, see **Other Optimizer Options**.
    - The **enable_expr_skew_optimization** GUC parameter controls whether to use expression statistics in the skew optimization policy. For details, see **Optimizer Method Configuration**.
    - **ANALYZE | ANALYSE**
12. GaussDB(DWS) supports large and wide tables, with a maximum of 5,000 columns.
13. **Create index/reindex** supports parallel processing.
14. The **pgxc_get_cstore_dirty_ratio** function is added to obtain the dirty page rate of CU, Delta, and CUDesc in the target table (only **hstore_opt** is supported).

[Convergence and unification]

1. One-click lakehouse: You can use **create external schema** to connect to the HiveMetaStore metadata, avoiding complex **create foreign table** operations and reducing maintenance costs. For details, see **Accessing HiveMetaStore Across Clusters**.
2. GaussDB(DWS) allows for reading and writing in Parquet/ORC format, as well as overwriting, appending, and multi-level partition read and write.
3. GaussDB(DWS) allows for reading in Hudi format.
4. Foreign tables support concurrent execution of **ANALYZE**, significantly improving the precision and speed of statistics collection. However, foreign tables do not support AutoAnalyze capabilities, so it is recommended to manually perform **ANALYZE** after data import.
5. Foreign tables can use the local disk cache for read acceleration.
6. Predicates such as **IN** and **NOT IN** can be pushed down for foreign tables to enhance partition pruning.

7. Foreign tables now support complex types such as map, struct, and array, as well as bytea and blob types.
8. Foreign tables support data masking and row-level access control.
9. GDS now supports the fault tolerance parameter **compatible_illegal_char** for exporting foreign tables.
10. The **read_foreign_table_file** function is added to parse ORC and Parquet files, facilitating fault demarcation.

High availability

1. The fault recovery speed of the unlogged table is greatly improved.
2. Backup sets support cross-version restoration. Fine-grained table-level restoration supports restoration of backup sets generated by clusters of earlier versions (8.1.3 and later versions).
3. Fine-grained table-level restoration supports restoration to a heterogeneous cluster (the number of nodes, DNs, and CNs can be different).
4. Fine-grained restoration supports permissions and comments. Cluster-level and schema-level physical fine-grained backups support backup permissions and comments, as do table-level restorations and schema-level DR.

Space saving

1. Column storage now supports JSONB and JSON types, allowing JSON tables to be created as column-store tables, unlike earlier versions which only supported row-store tables.
2. Hot and cold tables support partition-level **index unusable**, saving local index space for cold partitions.
3. The upgraded **hstore_opt** provides a higher compression ratio and, when used with the Turbo engine, saves 40% more space compared to column storage.

O&M and stability improvement

1. The query filter is enhanced to support interception by SQL feature, type, source, and processed data volume. For details, see **CREATE BLOCK RULE**.
2. GaussDB(DWS) now automatically frees up memory resources by reclaiming idle connections in a timely manner. You can specify the **syscache_clean_policy** parameter to set the policy for clearing the memory and number of idle DN connections. For details, see **Connection Pool Parameters**.
3. The **gs_switch_respool** function is added for dynamic switching of the resource pool used by **queryid** and **threadid**. This enables dynamic adjustment of the resources used by SQL. For details, see **Resource Management Functions**.
4. The **pg_sequences** view is added to display the attributes of sequences accessible to the current user.
5. The following functions are added to allow you to query information about all chunks requested by the memory in a specified shared memory:
   - **pg_shared_chunk_detail**
   - **pv_session_chunk_detail**

  - pg_shared_chunk_dump
  - pv_session_chunk_dump

6. The **pgxc_query_resource_info** function is added to display the resource usage of the SQL statement corresponding to a specified query ID on all DNs. For details, see **pgxc_query_resource_info**.
7. The **pgxc_stat_get_last_data_access_timestamp** function is added to return the last access time of a table. This helps the service to identify and clear tables that have not been accessed for a long time. For details, see **pgxc_stat_get_last_data_access_timestamp**.
8. SQL hints support more hints that provide better control over the generation of execution plans. For details, see **Configuration Parameter Hints**.
9. Performance fields are added to top SQL statements that are related to syntax parsing and disk cache. This makes it easier to identify performance issues. For details, see **Real-time Top SQL**.
10. The preset data masking administrator has the authority to create, modify, and delete data masking policies.
11. Audit logs can record objects that are deleted in cascading mode.
12. Audit logs can be dumped to OBS.

Ecosystem compatibility

1. **if not exists** can be included in the **create schema**, **create index**, and **create sequence** statements.
2. The **merge into** statement now allows for specified partitions to be merged. For details, see **MERGE INTO**.
3. In Teradata-compatible mode, trailing spaces in strings can be ignored when comparing them.
4. GUC parameters can be used to determine if the **n** in **varchar(n)** will be automatically converted to **nvarchar2**.
5. PostGIS has been upgraded to version 3.2.2.

Restrictions

1. A maximum of 256 VWs are supported, each with 1,024 DNs. It is recommended to have no more than 32 VWs and 128 DNs.
2. DR is not supported by OBS tables that have decoupled storage and compute. Only full backup and restoration are available.

Behavior changes

1. **VACUUM FULL**, **ANALYZE**, and **CLUSTER** are only supported for individual tables, not the entire database. Even though there are no syntax errors, the commands will not be executed.
2. OBS tables with decoupled storage and compute do not support delta tables. If **enable_delta** is set to **on**, no error is reported, but delta tables do not take effect. If a delta table is required, use the hstore-opt table instead.
3. By default, NUMA core binding is enabled and can be turned off dynamically using the **enable_numa_bind** parameter.

4. Upgrading from version 8.3.0 Turbo to version 910 changes the numeric(38) data type in Turbo tables to numeric(39), without affecting display width. Rolling back to the previous version will not reverse this change.
5. Due to the decoupling of storage and compute, the EVS storage space in DWS 3.0 is half that of DWS 2.0 by default. For example, purchasing 1 TB of EVS storage provides 500 GB in DWS 3.0 for active/standby mode, compared to 1 TB in DWS 2.0. When migrating data from DWS 2.0 to DWS 3.0, the EVS storage space required in DWS 3.0 is twice that of DWS 2.0.

1.3 Version 9.0.x (cloud native 3.0)

[V9.0.3 Release Date] February 2024

[Release date of V9.0.2]: November 2023

Version 9.0.3

[Added new features]

Table 1-2 New features in 9.0.3

Scenario	Feature	Description	Reference
Data lakehouse	Readable and writable foreign tables	Readable and writable foreign tables are supported, allowing for read and write on a single foreign table.	-
	Foreign table query performance optimization	Predicate pushdown for **in** and **not in** enhances partition pruning.	-
	Complex data type support	Complex data types such as map, struct, array, and bytea/blob are supported.	-
	Parallel foreign table export	Parallel exporting of foreign tables is supported with **enable_insert_ft_dop** (default: **disabled**).	-
	INSERT OVERWRITE	The insert overwrite operation can be performed on either the entire table or individual partitions.	INSERT

Scenario	Feature	Description	Reference
	HiveMetaStore statistics accessibility	HiveMetaStore statistics of external schema tables can be read.	-
Decoupled storage and compute	Asynchronous read optimization	Asynchronous read is optimized.	-
	Data can be imported in parallel to tables with decoupled storage and compute.	Data can be imported to in parallel to OBS tables with decoupled storage and compute (**enable_insert_dop** enabled, which is disabled by default). Exporting data in parallel consumes more CPU and memory resources.	-
	INSERT OVERWRITE for OBS tables with decoupled storage and compute	OBS tables with decoupled storage and compute support INSERT OVERWRITE.	INSERT
	Hot and cold local disk cache	The local disk cache supports hot and cold settings. Different cache policies can be configured for different partitions.	-
Enhanced engine	jsonb/json for column-storage	Column storage supports the jsonb/json type.	-

Version 9.0.2

[Added new features]

Table 1-3 New features in version 9.0.2

Scenario	Feature	Description	Reference
Data lakehouse	One-click lakehouse	**create external schema** is used to directly connect to HiveMetaStore metadata, avoiding complex **create foreign table** operations.	-
	Parallel foreign table ANALYZE	This function is controlled by the **enable_parallel_analyze** parameter. By default, this function is enabled, which significantly improves the speed of collecting ANALYZE statistics. Note: Foreign tables do not have the AUTO ANALYZE capability. You are advised to manually perform ANALYZE to collect statistics after data import.	-
	parquet/orc read and write for foreign tables	Overwriting, appending, and multi-level partition read/write are supported.	CREATE FOREIGN TABLE (SQL on OBS or Hadoop)
	Foreign table read acceleration by disk cache	Foreign tables can use the local disk cache for read acceleration.	-
	Foreign table data masking and row-level access control.	Foreign tables support datamasking and row-level access control.	-
Decoupled storage and compute	Decoupled storage and compute	You can specify **COLVERSION=3.0** to create a column-store table as an OBS table with decoupled storage and compute. In this way, user data is stored in OBS and is charged on demand.	CREATE TABLE

Scenario	Feature	Description	Reference
	Asynchronous read and write for OBS tables with decoupled storage	OBS tables that use decoupled storage can perform asynchronous reads and writes.	-
	Parallel ANALYZE for OBS tables with decoupled storage	OBS tables with decoupled storage support parallel ANALYZE, which is controlled by the GUC parameter **enable_parallel_analyze**. This function is enabled by default.	-
	Backup and restoration	OBS tables with separated storage and compute support only full backup and restoration.	-
	Elastic VW	Elastic Virtual Warehouse (VW) supports read/write acceleration. It takes 12 minutes to start an elastic VW, which is irrelevant to the current node scale, number of objects, and data volume. Integrating additional elastic VWs can bolster the system's read/write throughput, concurrent processing, and resource isolation.	**Periodically Adding or Deleting DWS 3.0 Logical Clusters**

Scenario	Feature	Description	Reference
	Elastic scale-out and ordinary scale-out.	Elastic scale-out does not redistribute data on OBS. Classic scaling redistributes all data. The system automatically determines the scaling mode based on whether the value of **table_buckets** divided by the total number of DNs exceeds the [**min_table_buckets_per_dn**, **max_table_buckets_per_dn**] range.	**Impact on the System**
	OBS monitoring view	Added the OBS monitoring view **PGXC_OBS_IO_SCHEDULER_PERIODIC_STATS**.	-
	Disk cache view	Added the disk cache views **PGXC_DISK_CACHE_STATS** and **PGXC_DISK_CACHE_ALL_STATS.**	-
	Clearing residual files	Added functions **pgxc_scan_residualfiles** and **pgxc_rm_scan_residualfiles_archive** to detect and clear residual files.	-
Performance optimization	Memory sorting performance optimization	Optimized the memory sorting performance.	-

[Behavior Change]

1. VACUUM FULL/ANALYZE/CLUSTER are not supported for the entire database. No error is reported for the syntax, but the syntax is not executed actually. They are supported only for a single table.
2. OBS tables with decoupled storage and compute do not support delta tables. If **enable_delta** is set to **on**, no error is reported, but delta tables do not take effect.

3. Only scale-out is supported. During scale-out, tables that are being scaled out can only be queried.
4. The default sampling ratio is adjusted to one ten-thousandth by setting **default_statistics_target** to **-0.01**.

1.4 Version 8.3.0

1.4.1 New Features in 8.3.0.100

For details about new features in 8.3.0.x, see **New features**. For details about resolved issues in each patch, see **New Features and Resolved Issues in 8.3.0.x**.

[V8.3.0.110 Release Date] 2024-10-26

[V8.3.0.108 Release Date] 2024-07-22

[V8.3.0.105 release date] 2024-07-10 (whitelisted users)

[V8.3.0.103 release date] 2024-06-10 (whitelisted users)

[V8.3.0.101 release date] 2024-04-25 (whitelisted users)

[V8.3.0.100 Release Date] 2024-04-17

New features

Table 1-4 8.3.0 feature changes

Category	Feature	Description	Reference
SQL	Last access time of a table object can be recorded.	• Added the **enable_save_dataaccess_timestamp** parameter to specify whether to record the last access time of a table. • Added the **last_reference_timestamp** field to the **PGXC_STAT_OBJECT** view. It can be used to query the last access time of a table. • Added the **pg_stat_get_all(regclass)** and **pgxc_stat_get_all(regclass)** functions to return table tuple information on CNs recorded in **pg_stat_object**.	• "GUC Parameters" > "Statistics During the Database Running" > "Query and Index Statistics Collector" in the *Developer Guide* • "System Catalogs and Views" > "System Catalogs" > "PGXC_STAT_OBJECT" in the *Developer Guide* • "Functions and Operators > Statistics Information Functions" in *SQL Syntax Reference*
Security	OneAccess authentication	• Added **OneAccess authentication** to the **rolauthinfo** field of **PG_AUTHID**. • Added the following description to the **AUTHINFO** parameter: When the OneAccess authentication mode is supported, **authinf** must contain **oneaccessClientId** and **domain** information in the format of 'oneaccessClien-tId=xxxx, domain=xxxx'.	• "System Catalogs" > "PG_AUTHID" in the *Developer Guide* • "DDL Syntax" > "CREATE ROLE" in *SQL Syntax Reference*
Usability	**pg_job** can record error information.	Added the **PG_JOB_INFO** system catalog to record the execution results of scheduled tasks.	"System Catalogs and Views" > "System Catalogs" > "PG_JOB_INFO" in the *Developer Guide*

Category	Feature	Description	Reference
High Performance	Pooler can connect to the standby node.	The GUC parameter **enable_connect_standby** is added to set the CN to connect to the standby DN.	"GUC Parameters" > "Connection and Authentication" > "Communication Library Parameters" in the *Developer Guide*
CMC	Sunset of historical system catalog and system function interfaces	Users can learn about the product evolution through the sunset plan. This improves product experience, and improves product reliability and usability.	Added section "Metadata Sunset Description" to summarize sunsetted views and functions.

Category	Feature	Description	Reference
HA	Enhanced O&M views	Added functions: 1. **pgxc_get_xlog_stats()**, which runs on CNs and the types and numbers of Xlogs on DNs from the DN startup time. 2. **pgxc_get_wal_speed()**, which runs on CNs and obtains the WAL generation rate of each DN and the receive, write, flush, and redo rates of the standby DN. 3. **pg_xlog_display_one_lsn(start_lsn)**, which runs on CNs or DNs and parses the Xlog in the current location based on the start LSN. 4. **pg_xlogdump (tablename)**, which runs on CNs or DNs and parses and filters Xlog files based on table names. 5. **pg_xlogdump (xid)**, which runs on CNs or DNs and parses and filters Xlog files based on transaction IDs. 6. **pg_xlogdump (start_lsn, end_lsn)**, which runs on CNs or DNs and parses Xlog files based on the start and end LSNs.	"Functions and Operators" > "Statistics Information Functions" in *SQL Syntax Reference*

Category	Feature	Description	Reference
HA	Clearing residual files	Added the following functions to scan residual files: • pg_scan_residualfiles() • pgxc_scan_residualfiles(query_flag) • pg_get_scan_residualfiles() • pgxc_get_scan_residualfiles(query_flag) • pg_archive_scan_residualfiles() • pgxc_archive_scan_residualfiles(query_flag) • pg_rm_scan_residualfiles_archive() • pgxc_rm_scan_residualfiles_archive(query_flag)	"Functions and Operators" > "Residual File Management Functions" > "Functions for Scanning Residual Files" in *SQL Syntax Reference*
High Performance	Column-store CU level-2 partitions	Added the **secondary_part_column** and **secondary_part_num** parameters to specify the column names and number of level-2 partition columns in a column-store table.	"Hybrid Data Warehouse" > "Hybrid Data Warehouse Syntax" > "CREATE TABLE" in the *Developer Guide*
High Performance	Table-level oldestxmin	Added the GUC parameter **enable_table_level_oldestxmin** to control whether to enable the table-level oldestxmin feature. Added the GUC parameter **old_txn_threshold**. When the table-level oldestxmin is calculated, transactions whose running duration exceeds the value of this parameter are regarded as long transactions.	"GUC Parameters" > "Automatic Cleanup" in the *Developer Guide*

Category	Feature	Description	Reference
Foreign tables	sql on hudi	Added the Hudi system function **hudi_set_sync_commit** to set the start timestamp of the Hudi automatic synchronization task. Added the **cow_improve** parameter is added to optimize the **COPY_ON_WRITE** table. This parameter can be specified only when **format** is set to **hudi**.	"Functions and Operators" > "Hudi System Functions" in *SQL Syntax Reference* "DDL Syntax" > "CREATE FOREIGN TABLE (SQL on OBS or Hadoop)" in *SQL Syntax Reference*
Lock	Distributed deadlock detection	Added the GUC parameters **enable_global_deadlock_detector** and **global_deadlock_detector_period** to specify whether to enable the distributed deadlock detection function and the detection interval.	"GUC Parameters" > "Lock Management" in the *Developer Guide*
Real-time queries	Approximate computing: column-store vectorized functions based on the HLL algorithm	Added the **approx_count_distinct(col_name)** function. Added the **approx_count_distinct_precision** parameter, which indicates the number of buckets in the HyperLogLog++ (HLL++) algorithm. This parameter can be used to adjust the error rate of the **approx_count_distinct** aggregate function.	"Functions and Operators" > "Aggregate Functions" in *SQL Syntax Reference* "GUC Parameters" > "Developer Options" in the *Developer Guide*
Real-time queries	Subplan vectorization	Added the GUC parameter **vector_engine_strategy** to control the vectorization enhancement policy.	"GUC Parameters" > "Miscellaneous Parameters" in the *Developer Guide*

Category	Feature	Description	Reference
SQL	Funnel functions and retention functions	Added the funnel functions **window_funnel**, **retention**, **range_retention_count**, and **range_retention_sum**.	"Functions and Operators" > "Funnel and Retention Functions" in *SQL Syntax Reference*
SQL	Job backpressure	Added the GUC parameter **max_queue_statements** to set the maximum queue length of queuing jobs.	"GUC Parameters" > "Resource Management" in the *Developer Guide*
SQL	Releasing locks at query end	Added the GUC parameter **enable_release_scan_lock** to control whether to release the level-1 lock after the SELECT statement is executed.	"GUC Parameters" > "Lock Management" in the *Developer Guide*
SQL	Replication tables	Added **disable_update_returning_check** to the GUC parameter **behavior_compat_options.**	"GUC Parameters" > "Miscellaneous Parameters" in the *Developer Guide*
Replicability	Optimized the rounding rule of the cast function.	Added options for the GUC parameter **behavior_compat_options**: **enable_banker_round**: specifies how numeric types round their values, using either the rounding or the banker method.	"GUC Parameters" > "Miscellaneous Parameters" in the *Developer Guide*
SQL	**string_agg** behavior compatibility	Added the **enable_full_string_agg** option to the GUC parameter **behavior_compat_options**.	"GUC Parameters" > "Miscellaneous Parameters" in the *Developer Guide*
Import and export	Upgraded the ORC third-party library and optimized its performance.	Added the GUC parameter **dfs_max_memory** to specify the maximum memory that can be occupied during ORC export.	"GUC Parameters" > "Resource Consumption" > "Memory" in the *Developer Guide*

Category	Feature	Description	Reference
Partition management	The COPY statement spilling behavior is optimized.	Added the GUC parameter **default_partition_cache_strategy** to control the default policy of partition cache.	"GUC Parameters" > "Resource Management" in the *Developer Guide*
Hybrid data warehouse	Batch upsert is optimized for hstore tables.	Added the GUC parameter **enable_hstore_keyby_upsert**, which controls the optimization of batch upsert in the hstore table.	"Hybrid Data Warehouse" > "Hybrid Data Warehouse GUC Parameters" in the *Developer Guide*
Partition management	The automatic partition reducing operation of automatic partition management blocks real-time data import to the database.	Added the **pg_partition_management_time** function to modify the invoking time of the automatic partition increasing tasks.	"Functions and Operators" > "Database Object Functions" > "Partition Management Functions" in *SQL Syntax Reference*
Compatibility	Integer division result compatible with PG (added in 8.3.0.100)	The **enable_int_division_by_truncate** option is added to the GUC parameter **behavior_compat_options** to control whether the result set is an integer or a floating point number. The behavior is compatible with PostgreSQL or ORA.	"GUC Parameters" > "Version and Platform Compatibility" in the *Developer Guide*
Compatibility	New parameter that controls whether the **case when** condition can contain functions that return multiple result sets.	Added the **unsupported_set_function_case** option to the GUC parameter **behavior_compat_options**. Specifies whether the **case when** condition can contain functions that return multiple result sets.	"GUC Parameters" > "Version and Platform Compatibility" in the *Developer Guide*

Category	Feature	Description	Reference
Compatibility	The window function **last_value** supports the **ignore nulls** feature.	The **LAST_VALUE** function supports the **IGNORE NULLS** syntax.	"Functions and Operators" > "Window Functions" in *SQL Syntax Reference*
Performance	LEFT JOIN estimation (added in 8.3.0.100)	The GUC parameter **left_join_estimation_enhancement** is added to determine whether to use the optimized estimated number of rows for **left join**. The default value is **off**.	"GUC Parameters" > "Query Planning" > "Optimizer Method Configuration" in the *Developer Guide*
SQL syntax enhancement	Clearing and rebuilding indexes in the column-store index area (added in 8.3.0.100)	The **WITHOUT UNUSABLE** syntax option is added. • ALTER INDEX index_name REBUILD [PARTITION partition_name] [WITHOUT UNUSABLE] • ALTER TABLE REBUILD PARTITION partition_name [WITHOUT UNUSABLE] • REINDEX { INDEX | [INTERNAL] TABLE } name [FORCE | WITHOUT UNUSABLE] • REINDEX { INDEX | [INTERNAL] TABLE } name PARTITION partition_name [FORCE | WITHOUT UNUSABLE]	"ALTER INDEX/ALTER TABLE/REINDEX" in *SQL Syntax Reference*

Behavior Changes

Table 1-5 Behavior changes

Change Type	No.	Item	Description
Deleted	1	policy_oid	Previously, the value of the **policy_oid** field was NULL post-upgrade to version 8.3.0. Now, a script automatically populates this field during the upgrade process.
Modified	2	gs_dump	**gs_dump** includes the **unusable** status of indexes.
	3	pg_get_tabledef	**pg_get_tabledef** includes the **unusable** status of indexes.
	4	pg_get_indexdef	**pg_get_indexdef** carries the **unusable** status of the index.
	5	create table like	When copying an index using the **INCLUDE** clause, the new index inherits the unusable status from the source index.

SQL Syntax Changes

Table 1-6 SQL syntax changes

Change Type	No.	Item	Description
Modified	1	CREATE TABLE	New support scenario: **create <***common table* **>like <***partition table* **>INCLUDING INDEXES**. The index creation logic has been updated to set the index type based on the actual table type, not the 'like' table type.
Added	2	last_value ignore nulls	The **last_value ignore nulls** syntax is introduced to exclude NULL values during calculations.
	3	ALTER INDEX index_name REBUILD [PARTITION partition_name] [WITHOUT UNUSABLE]	Enhanced syntax to rebuild unusable indexes.

Change Type	No.	Item	Description
	4	ALTER TABLE REBUILD PARTITION partition_name [WITHOUT UNUSABLE]	Enhanced syntax to rebuild unusable indexes.
	5	REINDEX { INDEX | [INTERNAL] TABLE } name [FORCE | WITHOUT UNUSABLE]	Enhanced syntax to rebuild unusable indexes.
	6	REINDEX { INDEX | [INTERNAL] TABLE } name PARTITION partition_name [FORCE | WITHOUT UNUSABLE]	Enhanced syntax to rebuild unusable indexes.
	7	CREATE INDEX ... [UNUSABLE] ... PARTITION index_partition_name [UNUSABLE]	Added support for creating indexes in the unusable.

System Catalogs

Table 1-7 System catalogs

Change Type	No.	Item	Description
Modified	1	pg_stat_object	Renamed the reserved field **extra1** to **last_autovacuum_csn**.
	2	gs_wlm_session_info	Added a new field **parse_time**.
	3	pg_job_info	Added a new column **jobdb** to store the database information of jobs.

System Functions

Table 1-8 System functions

Change Type	No.	Item	Description
Added	1	dbms_job.submit_node	Added a new function **dbms_job.submit_node** to designate the execution node, which is defaulted to null, representing the current CN node.
	2	dbms_job.change_node	Added a new function **dbms_job.change_node** to designate the execution node, which is defaulted to null, representing the current CN node.
	3	Funnel function	Added the funnel function **windowfunnel**.
	4	Retention function	Added retention functions.
	5	Retention extension function	Added the retention extension function **range_retention_count**.
	6	Retention extension function	Added the retention extension function **range_retention_sum**.
	7	approx_count_distinct	Added an approximate calculation aggregate function.
	8	pg_scan_residualfiles	Scans all residual file records in the database where the current node resides
	9	pgxc_scan_residualfiles	Scans all nodes for the residual files of the current database
	10	pg_get_scan_residualfiles	Obtains all residual file records of the current node.
	11	pgxc_get_scan_residualfiles	Obtains residual file records on all nodes.
	12	pg_archive_scan_residualfiles	Archives all residual file records of the current node.
	13	pgxc_archive_scan_residualfiles	Archives residual file records on all nodes.
	14	pg_rm_scan_residualfiles_archive	Deletes files from the archived file list on the current node.

Change Type	No.	Item	Description
	15	pgxc_rm_scan_residualfiles_archive	Deletes files in the archive directory from all nodes.
	16	pg_partition_management_time	Modifies the invoking time of the auto-increment partition task.
	17	uniq	Assists in UV calculation of data, such as precise deduplication.
	18	reload_cold_partition	Changes a cold partition into a hot partition.
Modified	19	pgxc_get_small_cu_info	Added the feature of querying the number of CUs in each level-2 partition.
	20	pgxc_get_wlm_session_info_bytime	Added the **parse_time** field.
Deleted	21	pg_sync_cstore_delta(text)	Deprecated the system function.
	22	pg_sync_cstore_delta	Deprecated the system function.
	23	pgxc_pool_check	Deprecated the system function.
	24	pg_delete_audit	Deprecated the system function.
	25	pg_log_comm_status	Deprecated the system function.
	26	pgxc_log_comm_status	Deprecated the system function.
	27	signal_backend	Deprecated the system function.
	28	pg_stat_get_realtime_info_internal	Deprecated the system function.
	29	pg_stat_get_wlm_session_info_internal	Deprecated the system function.
	30	pg_stat_get_wlm_session_info	Deprecated the system function.
	31	pg_stat_get_wlm_statistics	Deprecated the system function.
	32	pg_user_iostat	Deprecated the system function.
	33	pg_stat_get_wlm_session_iostat_info	Deprecated the system function.

System Views

Table 1-9 System views

Change Type	No.	Item	Description
Added	1	pgxc_memory_debug_info	This view displays the memory error information of each node in the current cluster during job execution, helping locate memory errors.
Modified	2	pgxc_stat_object	Changed with **pg_stat_object**. The reserved field **extra1** is renamed **last_autovacuum_csn**.
	3	gs_wlm_session_statistics	Added the **parse_time** field.
	4	pgxc_wlm_session_statistics	Added the **parse_time** field.
	5	gs_wlm_session_history	Added the **parse_time** field.
	6	pgxc_wlm_session_history	Added the **parse_time** field.
	7	gs_wlm_session_info	Added the **parse_time** field.
	8	information_schema.tables	Modified the view definition and changed the filter criteria so that the views whose **reloptions** is empty can be queried.
	9	gs_table_stat	Optimized the query performance of the **gs_table_stat** view.
	10	gs_row_table_io_stat	Optimized the query performance of equivalence logic rewriting.
	11	gs_column_table_io_stat	Optimized the query performance of equivalence logic rewriting.
Deleted	12	gs_wlm_session_info_all	Deprecated the system view.
	13	pg_wlm_statistics	Deprecated the system view.
	14	pg_session_iostat	Deprecated the system view.

GUC Parameters

Table 1-10 GUC parameters

Change Type	No.	Item	Description
Added	1	max_queue_statements	Default: **-1** (unlimited queue length). Triggers an error and exits the job when the queue length exceeds this value. This parameter does not take effect for newly installed and upgraded clusters.
	2	job_retention_time	Maximum days to store **pg_job** execution results. Default: 30 days.
	3	vector_engine_strategy	Controls vectorization of the operator. Default: **improve** (maximize vectorization). Alternate: **force** (roll back to row storage plan).
	4	enable_release_scan_lock	Determines if the **SELECT** statements release the level-1 lock post-execution. It is disabled by default.
	5	job_queue_naptime	Interval for scheduling task checks and starting task threads. Default: **1** second.
	6	approx_count_distinct_precision	Number of buckets in HLL++ algorithm, affecting the error rate of the **approx_count_distinct** function. The number of buckets affects the precision of estimating the distinct value. More buckets make the estimation more accurate. Default: 17 buckets.
	7	llvm_compile_expr_limit	Limits the number of compiled LLVM expressions.
	8	llvm_compile_time_limit	Sets a threshold for LLVM compilation time as a percentage of executor running time, triggering an alarm if exceeded.
	9	max_opt_sort_rows	Maximum rows for optimized limit +offset in an **ORDER BY** statement. If the number of rows exceeds the value of this parameter, the original logic is used. If the number of lines is less than the value of this parameter, the optimized logic is used. Default: **0** (use original logic).

Change Type	No.	Item	Description
	10	dfs_max_memory	Maximum memory for ORC export. Default: **262144** KB.
	11	default_partition_cache_strategy	Default policy for partition cache control.
	12	enable_connect_standby	Allows CN to connect to a standby DN for O&M operations. This parameter applies only to O&M operations.
	13	enable_stream_sync_quit	Determines synchronous exit of stream thread after plan execution. This parameter is disabled by default so that sync quit is not blocked.
	14	full_group_by_mode	Behavior after enabling **disable_full_group_by_mysql**: • **nullpadding** indicates that NULL values in a non-aggregation column are replaced with values and non-null values in the column are used. The result set may contain different rows. • **notpadding** indicates that NULL values are not processed for non-aggregation columns and the entire row of data is obtained. The result set of non-aggregation columns is a random row.
Modified	15	behavior_compat_options	Added the **alter_distribute_key_by_partition** option to control **INSERT INTO** execution by partition post **ALTER TABLE**.
	16	behavior_compat_options	Added the **enable_full_string_agg** option for full or incremental aggregation logic in **string_agg(a, delimiter) over (partition by b order by c)**.
	17	behavior_compat_options	Added the **unsupported_set_function_case** option to control **case when** conditions with multi-result set functions.

Change Type	No.	Item	Description
	18	behavior_compat_options	Added the **enable_unknown_datatype** option, disallowing creation of tables with **unknown** type columns if unset.
	19	behavior_compat_options	Added the **enable_whole_row_var** option, intercepting scenarios where a table name is used as an expression (**SELECT T FROM T**, or **SELECT .... FROM T GROUP BY T**) if unset.
	20	time_track_strategy	Parameter level changed from **USERSET** to **SIGHUP**. The default value is **timer** at the kernel level, which can be set to tsc on the management console.
	21	rewrite_rule	Controls whether to use **case when** rewriting.

1.4.2 New Features and Resolved Issues in 8.3.0.x

8.3.0.110

Table 1-11 New features/Resolved issues in version 8.3.0.110

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	By default, the global deadlock detection function is enabled for newly installed clusters. In earlier versions, this is disabled by default.	-	-	-

Category	Feature or Resolved Issue	Cause	Version	Handling Method
Resolved issues	When a common table expression (CTE) is referenced only by another CTE for multiple times, null pointer access may be triggered, causing a fault.	When a CTE is referenced multiple times exclusively by another CTE, it can result in incorrect recursive traversal logic. This may trigger a null pointer access and cause a fault in the cluster.	8.3.0.101	Upgrade the version to 8.3.0.110 or later.
	If a query statement for a column-store table contains multiple result set functions, the error message " Set-returning function is not supported in vector engine" is displayed when the statement is executed.	The vectorized executor on the non-table scan operator does not support multi-result-set functions.	8.3.0.101	
	When the cluster is upgraded to 8.3.0, the thread pool function becomes invalid.	When upgrading the cluster to version 8.3.0, adjust the **max_stream_pool** setting based on the current hardware configuration. Note that **max_stream_pool** is set to **0** due to an abnormal default unit identification.	8.3.0.101	
	The error message "The parameter destMax is too small or parameter count is larger than macro parameter" is displayed when a Parquet foreign table is used for query.	The buffer space for string-type fields in the Parquet foreign table is not properly managed, causing insufficient memory allocation for fields larger than 1 KB. This, in turn, leads to service errors.	8.3.0.101	

Category	Feature or Resolved Issue	Cause	Version	Handling Method
	When performing a partition exchange operation on redistributed cold and hot tables, the error message "Multi-temperature table only supports column partition table" is displayed.	The temporary table cannot be created during redistribution because it references the attributes of the original cold and hot tables.	8.3.0.101	

8.3.0.108

Table 1-12 New features/Resolved issues in version 8.3.0.108

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-
Resolved issues	Lock Timeout Occurs During Concurrent Copy to the Database	The lock API is being misused in the release of the lock by RecoverFromDelta, causing it to remain locked.	8.3.0.105	Upgrade the version to 8.3.0.108 or later.

8.3.0.105

Table 1-13 New features/Resolved issues in version 8.3.0.105

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-
Resolved issues	A result set error can occur during multiple **count distinct** operations with **SHARE SCAN**.	This issue arises from the incorrect use of upper-layer equivalence class information in the CTE subquery, which results in misidentifying the distribution column and omitting the necessary data redistribution operator.	Versions earlier than 8.3.0.105	Upgrade the version to 8.3.0.105 or later.
	System catalog **autovacuum** is not functioning properly after upgrading to version 830.	Prior to the upgrade, a DR migration was carried out, which involved setting **autovacuum_max_workers** to **0** and then changing it to the default value of **3**. However, the value was not set to the expected value of **6** after the upgrade.	Versions earlier than 8.3.0.105	
	If a UDF receives data in **recv** mode, an error might not be captured even if **error** is **EAGAIN** after a timeout.	This occurs when **cn_retry** is enabled and the **socket recv** times out for 20 seconds, triggering error reporting logic instead of capturing the **EAGAIN** error code.	Versions earlier than 8.3.0.105	

8.3.0.103

Table 1-14 New features/Resolved issues in version 8.3.0.103

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	json/jsonb supports vectorization.	-	-	-
Resolved issues	Core dumps occur occasionally with ORC foreign tables.	During **foreign scan** query pushdowns on ORC foreign tables, if the query constraint is a Boolean expression with implicit conversion and uses the **is null** condition, abnormal internal type conversion and memory overwriting may occur (RelabelType is forcibly converted to var) (RelabelType is forcibly converted to var).	Versions earlier than 8.3.0.103	Upgrade the version to 8.3.0.103 or later.
	Cstore buffer error during 10 million batch upserts.	The cstore buffer's cache elimination mechanism is Clock Swap, which operates a maximum of three cycles every three seconds. This limitation prevents the elimination of a large number of CUs (Usage Count > 3).	Versions earlier than 8.3.0.103	
	Delta table space not released after import tasks stop.	Deleted records in the **hstore opt** table may not be cleared, preventing space from being released.	Versions earlier than 8.3.0.103	

8.3.0.101

Table 1-15 New features/Resolved issues in version 8.3.0.101

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	**VACUUM FULL** adapts to binlog tables.	-	-	-
Resolved issues	Executing the analyze statement results in an error message stating "Could not open file 'pg_clog/000000000075'."	Performing **VACUUM FULL** on a column-store table can prematurely reclaim clog files, causing access issues during **ANALYZE** after an active/standby switchover.	8.3.0.100	Upgrade the version to 8.3.0.101 or later.
	Error "duplicate key value violates unique constraint 'pg_jobs_id_index', Key(job_id)=() already exist" during high concurrency job creation with **\parallel on**.	The system failed to acquire the necessary lock for assigning unique job_ids. Consequently, concurrent processes attempting to obtain the same job_id led to error reports.	8.3.0.100	
	Redistribution fails due to inability to skip specified tables.	In multiple node group scenarios, redistribution cannot skip specified tables.	8.3.0.100	
	Hstore test case failure due to faulty network adapter.	Declaring a temporary variable without assigning a value leads to unexpected parameter values and excessively negative memory usage, causing test case execution failure when the network adapter is faulty.	8.3.0.100	

1.5 Version 8.2.1

1.5.1 New Features in 8.2.1

For details about the new features of version 8.2.1.*x* (released only for whitelisted users), see **New Features in 8.2.1**. For details about the resolved issues in each patch, see **New Features and Resolved Issues in 8.2.1.x**.

[V8.2.1.230 Release Date] October 2024

[V8.2.1.225 Release Date] July 2024

[V8.2.1.223 Release Date] June 2024

[V8.2.1.220 Release Date] April 2024

[V8.2.1.119 Release Date] February 2024

[V8.2.1.100 Release Date] May 2023

New features

Table 1-16 8.2.1 Feature changes

Feature	Description	Reference
Improved usability of the resource management function: Added the global memory control and adjustment mechanism, added top SQL statements and views to query the skews of disk space occupation of different DNs.	• The GUC parameter **enable_global_memctl** is added to control whether to enable the global memory management function. • **stmt_type** can be recorded to facilitate aggregate analysis on SQL statements. • The **PGXC_WLM_TABLE_DISTRIBUTION_SKEWNESS** view is added to display data skews of tables in the current database.	• "GUC Parameters > Load Management" in *Developer Guide* • "System Catalogs and System Views > System Views >GS_WLM_SESSION_HISTORY/ GS_WLM_SESSION_STATISTICS" in *Developer Guide*. The **stmt_type** field is added to display the query types of statements. • "System Catalogs and System Views > System Views > PGXC_WLM_TABLE_DISTRIBUTION_SKEWNESS" in *Developer Guide*

Feature	Description	Reference
Network I/O management and control based on resource pools	• The **enable_libcomm_schedule** parameter is added to control whether to enable network management and control. The **low_priority_bandwidth** parameter is added to specify the upper limit of the network bandwidth that can be occupied by low-priority queues. This parameter is used to control the network flows of low-priority queues. • The **send_speed** and **recv_speed** fields are added to the following system catalogs and views to indicate the average sending/receiving rate in the monitoring period: System catalog: GS_RESPOOL_RESOURCE_HISTORY GS_WLM_USER_RESOURCE_HISTORY System Views: GS_RESPOOL_RESOURCE_INFO PG_TOTAL_USER_RESOURCE_INFO PGXC_RESPOOL_RESOURCE_INFO PGXC_RESPOOL_RESOURCE_HISTORY PGXC_TOTAL_USER_RESOURCE_INFO PGXC_WLM_USER_RESOURCE_HISTORY • The | **BANDWIDTH = VALUE,| ACTION = ['abort' | 'penalty']** syntax is added to the **ALTER EXCEPT RULE/CREATE EXCEPT RULE** syntax. The	• "GUC Parameters > Resource Management" in *Developer Guide* • "System Catalogs and System Views" in *Developer Guide* • "DDL Syntax > ALTER EXCEPT RULE/CREATE EXCEPT RULE" in *SQL Syntax Reference* • "DDL Syntax > ALTER RESOURCE POOL/ CREATE RESOURCE POOL" in *SQL Syntax Reference*

Feature	Description	Reference
	parameter **bandwidth** sets the maximum network bandwidth that can be used for job execution. • The | **weight=bandwidth_weight** syntax is added to the **ALTER RESOURCE POOL/ CREATE RESOURCE POOL** syntax. The parameter **weight** specifies the network bandwidth weight of a resource pool.	
Exception rules (resource limit for a single SQL statement)	• The **short_acc**, **except_rule**, and **weight** fields are added to **PG_RESOURCE_POOL**. • The parameter **except_info** is added to **GS_WLM_SESSION_HISTORY**/ **GS_WLM_SESSION_STATISTICS** to display information about exception rules triggered by statements.	• "System Catalogs and Views > System Catalogs > PG_RESOURCE_POOL" in *Developer Guide* • "System Catalogs and System Views > System Views > GS_WLM_SESSION_HISTORY/ GS_WLM_SESSION_STATISTICS" in *Developer Guide*
By default, **&level** is set to **perf** for TopSQL clauses to monitor clauses of stored procedures, optimize internal statements of stored procedures, and locate faults.	The GUC parameter **resource_track_subsql_duration** is added to filter the minimum execution time of clauses in a stored procedure.	• "GUC Parameters > Resource Management" in *Developer Guide* • "Resource Monitoring > Real-Time Top SQL/ Historical Top SQL" in *Developer Guide*
Enhanced the communication capability. Added the GUC parameter **conn_recycle_timeout** to automatically reclaim idle connections.	The GUC parameter **conn_recycle_timeout** is added to specify the interval for reclaiming idle connections between CNs and DNs to the connection pool.	"GUC Parameters > Connection and Authentication > Communication Library Parameters" in *Developer Guide*

Feature	Description	Reference
Administrators can use SQL functions to query process stack information for service problem demarcation and analysis.	The **gs_stack()** function is added to obtain the stack information of CN or DN processes.	"Functions and Operators > Statistics Information Functions" in *SQL Syntax Reference*
Path pruning optimization in complex scenarios	• The GUC parameter **prefer_hashjoin_path** is added to control whether to preferentially generate the hash join path. • The **prefer_hashjoin_path** parameter is added to the parameters supported by GUC hints.	• "GUC Parameters> Query Planning> Optimizer Method Configuration" in *Developer Guide* • "Query Performance Optimization > Query Improvement > Hint-based Tuning > Configuration Parameter Hints" in *Developer Guide*
Top SQL optimization at the perf level	• The GUC parameter **time_track_strategy** is added to set the policy for collecting the execution time of the operators of the current session. • The views **show_tsc_info**, **show_all_tsc_info**, **get_tsc_info** and **get_all_tsc_info** are added to query the TSC information of the current node and all nodes. • The functions **show_tsc_info(), get_tsc_info()** and **test_tsc_info** are added to query the TSC information of the current node and all nodes.	• "GUC Parameters > Resource Management > time_track_strategy" in *Developer Guide* • "System Catalogs and System Views > System Views" in *Developer Guide* • "Functions and Operators > System Administration Functions > Other Functions" in *SQL Syntax Reference*

Feature	Description	Reference
share scan for multiple **count (distinct)** expressions	• The GUC parameter **stream_ctescan_pred_threshold** is added to control the minimum number of filter criteria contained in a CTE. • The **stream_ctescan_max_estimate_mem** parameter is added to control the maximum estimated memory value of the CTE. • The GUC parameter **stream_ctescan_refcount_threshold** is added to control the maximum number of times that the CTE can be referenced.	"GUC Parameters > Query Planning > Other Optimizer Options" in the *Developer Guide*
All non-aggregation function query fields do not need to be displayed after GROUP BY.	The **disable_full_group_by_mysql** option is added to the GUC parameter **behavior_compat_options** to control whether the non-aggregation function query fields can be partially displayed after **GROUP BY** in the query.	Developer Guide > GUC Parameters > Version and Platform Compatibility > Platform and Client Compatibility
Optimized **DISCARD**	The keyword **{ ALL | TEMP | TEMPORARY | PLANS | SEQUENCES }** is added to the syntax to release all temporary resources related to the current session, resources related to all temporary tables in the current session, cached query plans, and status related to all cached sequences.	"DDL Syntax > DISCARD" in *SQL Syntax Reference*
The interface function **read_global_var** is added.	The **read_global_var** interface function is added to read the current value of a global variable.	"Functions and Operators > System Management Functions > Configuration Setting Functions" in *SQL Syntax Reference*

Feature	Description	Reference
Optimized skew value comparison	When **Detail** is enabled, the execution plan displays the time required for skew value comparison.	"DML Syntax>EXPLAIN" in *SQL Syntax Reference*
postgis upgrade	• Added the method of installing PostGIS-3.2.2. • PostGIS3.2.2 does not support raster-related functions. • PostGIS 3.2.2 depends on the following third-party open-source software: Geos-3.11.0, Proj-6.0.0, Json 0.12.1, Libxml2 2.7.1, Sqlite3 • Added the list of operators and functions supported by PostGIS 3.2.2.	• "PostGIS Extension > PostGIS Overview" in *Developer Guide* • "PostGIS Extension > PostGIS Installation" in *Developer Guide* • "PostGIS Extension > PostGIS Support and Restrictions" in *Developer Guide*
pgxc_wlm_session_statistics and **explain SQL** support **unique_sql_id**.	The **unique_sql_id** field is added to the **GS_WLM_SESSION_HISTORY** view to indicate the normalized unique SQL IDs.	"System View > GS_WLM_SESSION_HISTORY" in *Developer Guide*
The arrays generated by the IN/ANY/ALL condition can be split into common expressions for execution.	The GUC parameter **enable_array_optimization** is added to determine whether to split the arrays generated by the **IN**, **ANY**, and **ALL** conditions into common expressions for execution.	"GUC Parameters > Query Planning > Other Optimizer Options" in *Developer Guide*
Optimized the rewriting of multi-column count(distinct) to support rewriting of volatile functions.	A value is added to GUC parameter **volatile_shipping_version**. The default value is changed from **2** to **3**.	"GUC Parameters > Query Planning > Other Optimizer Options" in *Developer Guide*
Removed the restriction that a maximum of 10 skew optimization values are allowed.	The GUC parameter **max_skew_num** is added to control the number of skew values allowed by the optimizer for redistribution optimization.	"GUC Parameters > Query Planning > Other Optimizer Options" in *Developer Guide*

Feature	Description	Reference
Optimized pushdown of some conditions in a foreign table for collaborative analysis.	The option **disable_gc_fdw_filter_partial_pushdown** is added to the GUC parameter **behavior_compat_options** to control the pushdown of filter criteria used to query data in the collaborative analysis foreign table (type: gc_fdw).	Developer Guide > GUC Parameters > Version and Platform Compatibility > Platform and Client Compatibility
Added data set verification for the collaborative analysis operators.	The **gc_fdw_verify_option** parameter is added to control whether to enable the logic for verifying the number of rows in the result set in the collaborative analysis feature.	"GUC Parameters > Developer Options" in *Developer Guide*
Concurrent truncate, exchange, and select operations	The options of the GUC parameter **ddl_select_concurrent_mode** is updated. The default value is changed to **none**. The **exchange** option is added. The parameter type is changed from single-enumerated type to multi-enumerated type. Multiple values can be set at the same time. Different values are separated by commas (,).	"GUC Parameters > Lock Management" in *Developer Guide*
Optimized the GTM thread pool to improve the GTM's capability of handling high concurrency.	Added the following GUC parameters: **gtm_option**, **defer_xid_cleanup_time**, **gtm_enable_threadpool**, and **gtm_num_threads**.	"GUC Parameters > Cluster Transaction Parameters" in *Developer Guide* "GUC Parameters > GTM Parameters" in *Developer Guide*
Empty files can be generated when empty tables are exported from GaussDB(DWS) to xsky S3.	The **obs_null_file** parameter is added to import and export empty files between GaussDB(DWS) and OBS.	"DDL Syntax > CREATE FOREIGN TABLE (OBS Import and Export)" in *SQL Syntax Reference*

Feature	Description	Reference
Improved autovacuum usability.	• The GUC parameter **enable_pg_stat_object** is added to control whether the autovacuum updates the system catalogs. • The **PG_STAT_OBJECT** system catalog is added to store table statistics and autovacuum efficiency information of the current instance. • The system view **PGXC_STAT_OBJECT** displays the table statistics and autovacuum efficiency information of all instances in the cluster.	• "GUC Parameters > Automatic Cleanup" in *Developer Guide* • "System Catalogs and System Views > System Catalogs > PG_STAT_OBJECTS" in *Developer Guide* • "System Catalogs and System Views > System Views > PGXC_STAT_OBJECT" in *Developer Guide*
Optimize the cost of scanning column-store table indexes. (8.2.1.100)	The GUC parameter **index_selectivity_cost** is added to control the cost calculation of cbtree during index scanning of column-store tables.	"GUC Parameters > Query Planning > Optimizer Method Configuration" in *Developer Guide*
The autovacuum lock of the column-store delta table is upgraded to resolve the error of adms import.	The GUC parameter **enable_mergelock_upgrade** is added to control whether to upgrade the lock level from 4 to 7 during merge of column-store delta tables, preventing errors caused by concurrent service operations.	"GUC Parameters > Lock Management" in *Developer Guide*
If no index type is specified for a column-store table, a B-tree index is created by default.	The GUC parameter **default_table_behavior** is added to specify the default table behavior.	"GUC Parameters > Developer Options" in *Developer Guide*

Feature	Description	Reference
DFX view for resource monitoring (8.2.1.100)	Added the following views: • **GS_QUERY_MONITOR** displays the running/ queuing information and resource usage of ongoing queries. Only queuing and running jobs are displayed. • **GS_RESPOOL_MONITOR** displays the running information and resource usage of all resource pool jobs. • **GS_USER_MONITOR** displays the job running information and resource usage of all users.	"System Catalogs and System Views > System Views" in *Developer Guide*
The bind monitoring function is added for top SQL statements when BEs are used in batches. (8.2.1.100)	• For a main statement that is not spilled to disks, its record in the top SQL history table is displayed only when the next job is delivered. • Added a restriction to the **query_plan** column in the **GS_WLM_SESSION_HISTORY** and **GS_WLM_SESSION_STATISTICS** views: Execution plans are shown for DML statements but not for DDL statements. When a user issues a Parse Bind Execute (PBE) batch statement, the execution plan for the PBE statement should include the number of data bindings and be displayed as a percentage under **PBE bind times**.	• "Resource Monitoring" > "Real-Time Top SQL" in the *Developer Guide* • "System Catalogs and System Views > System Views > GS_WLM_SESSION_HISTORY/ GS_WLM_SESSION_STATISTICS" in *Developer Guide*
comm_max_stream can be dynamically set. (8.2.1.100)	The type of the GUC parameter **comm_max_stream** is changed from POSTMASTER to SIGHUP. The value range is changed from 1-60000 to 1-65535. The default value is changed to 1024.	"GUC Parameters > Connection and Authentication > Communication Library Parameters" in *Developer Guide*

Feature	Description	Reference
Enhanced audit log functions (8.2.1.100). The object name, field name, operation type, number of returned records, and error code columns are added.	• The GUC parameter **audit_object_details** is added to control whether to record the **object_details** field in audit logs. This field records the table name, column name, and column type in the audit statement. • The **object_details** field is added to the **pg_query_audit()** function to record the columns involved in the statement and their usage types. The **result_rows** field indicates the number of records returned after the statement is executed. The **error_code** field indicates the error code when an error occurs during statement execution. The **pg_query_audit_details()** function is added for viewing audit logs and parsing the **object_name** and **object_details** fields in audit logs from the JSON format.	• "GUC Parameters> Auditing> Audit Switch" in *Developer Guide* • "Functions and Operators> Security Functions" in *SQL Syntax Reference*

Feature	Description	Reference
Multiple masking policies for a single table (8.2.1.100)	• The **policy_name** (indicating the masking policy name) is added to the **REDACTION_COLUMNS** view. The **policy_oid** field is added to the **PG_REDACTION_COLUMN** system catalog to indicate the OID of the masking policy. This field is used to search for masking column information from the metadata in the system catalog. The **policy_order** field is added to the **PG_REDACTION_POLICY** system catalog to indicate the masking policy sequence. • The **AFTER | BEFORE** parameter is added to **CREATE REDACTION POLICY** to specify the relative position of the current policy.	• "System Catalogs and System Views > REDACTION_COLUMNS / PG_REDACTION_COLUMN/ PG_REDACTION_POLICY" in *Developer Guide* • "DDL Syntax > CREATE REDACTION POLICY" in *SQL Syntax Reference*
Optimized the 1 GB limit for column-store autovacuum. (8.2.1.100)	The GUC parameter **col_min_file_size** is added to resolve the following issue: A large amount of dirty data is generated when a column-store table is imported to the database. Space reclamation cannot be triggered if the file size does not exceed 1 GB.	"GUC Parameters > Automatic Cleanup" in *Developer Guide*
GB18030 code upgrade (8.2.1.100)	The description of common encoding formats is added to the description of the encoding parameter. The encoding format GB18030_2022 is added to Dataencoding to solve the error of exporting data in the GB18030 (earlier version) format.	• "DDL Syntax > CREATE FOREIGN TABLE (for GDS Import and Export)" in *SQL Syntax Reference* • "DML Syntax > COPY" in *SQL Syntax Reference*

Feature	Description	Reference
Core last word tool (8.2.1.100)	The GUC parameter **enable_ffic_log** is added, which provides a lightweight log solution to quickly and efficiently record core dump, providing necessary information for core dump fault locating and demarcation, without relying on the gdb tool.	"GUC Parameters > Resource Management" in *Developer Guide*
Delta table information view for HStore tables (8.2.1.100)	The **pgxc_get_hstore_delta_info(rel_name text)** function is added to quickly query delt table information (such as the number of records of each type and the size of the delta table) to help locate faults.	"Hybrid Data Warehouse > Hybrid Data Warehouse Functions" in *Developer Guide*

Feature	Description	Reference
Modified the copy error tolerance table (8.2.1.100)	• The **columnname**, **errcode**, and **queryid** columns are added to the **public.pgxc_copy_error_log** table. • Canceled the restriction on the **LOG ERRORS DATA** parameter: Only users with the **supper** permission can use the **LOG ERRORS DATA** parameter. • The **max_copy_data_display** parameter is added. The following modifications are made to the copy fault tolerance table of GaussDB(DWS): 1. The restriction is removed: Only the administrator can modify the **rawrecord** field in the original data of the error table. 2. If the **not null** constraint is violated, an error is reported and recorded in the error table. 3. Error fields and error codes are displayed to facilitate fault locating when wide tables are imported to the database.	• "System Management Functions > Other Functions" in *SQL Syntax Reference* • "DML Syntax > COPY" in *SQL Syntax Reference* • "GUC Parameters > Parallel Import" in *Developer Guide*
Enhanced hot and cold tables (8.2.1.100)	The user functions **refresh_hot_storage(text)** and **refresh_hot_storage(text)** are added for cold and hot tables. **refresh_hot_storage(text, text)** is used to refresh the partition data of a specified multi-temprature table to OBS.	"Functions and Operators > Database Object Functions > Hot and Cold Table User Functions" in *SQL Syntax Reference*

Feature	Description	Reference
Index scan (8.2.1.100)	The GUC parameter **index_cost_limit** is added to repair the forward compatibility of column-store indexcost and avoid the default plan change.	"GUC Parameters > Optimizer Method Configuration" in *Developer Guide*
Cache size of a sequence can be modified (8.2.1.100).	The **CACHE** keyword is added to the **ALTER SEQUENCE** syntax to specify the number of sequence numbers to be allocated and store the numbers in the memory for faster sequence access.	"DDL Syntax > ALTER TABLE" in *SQL Syntax Reference*
Removed the restriction on the number of merge partitions (8.2.1.100).	Remove the restriction on the maximum number of source partitions in the **merge_clause** clause.	"DDL Syntax > ALTER TABLE PARTITION" in *SQL Syntax Reference*
GDS fault tolerance compatibility (8.2.1.230)	Added GUC parameter **gds_fill_multi_missing_fields** This parameter controls the behavior when the GDS foreign table fault tolerance parameter **fill_missing_fields** is set to **true** or **on**. If enabled, the GDS foreign table allows for the loss of multiple last columns in a single row of the source data file. Otherwise, the GDS foreign table still permits the loss of multiple last columns in a single row of the source data file. This parameter compatible with historical behavior.	Developer Guide > GUC Parameters > Version and Platform Compatibility > Platform and Client Compatibility
The database can preferentially select **tsc** based on the node status to collect the operator time. (8.2.1.230)	The value **opt** is added to the **time_track_strategy** parameter. The database preferentially selects **tsc** to collect the operator time based on the node status.	"GUC Parameters > Resource Management" in *Developer Guide*

Feature	Description	Reference
Fine-grained table-level restoration to heterogeneous clusters (8.2.1.230)	Restoring one or more tables to a heterogeneous cluster is possible with OBS or XBSA as the backup media. You can add the **db-options** parameter to rebuild the target database and enable table-level restoration from the cluster-level backup set to the new cluster.	Tool Guide > Server Tool > GaussRoach.py > Functions Tool Guide > Server Tool > GaussRoach.py > Command Reference > restore
Physical fine-grained cross-version recovery (8.2.1.230)	• Physical fine-grained backup sets at the cluster level, backed up in version 8.1.3 or later, can be restored with precision for a single table or multiple tables. • Physical fine-grained backup sets at the schema level, backed up in version 8.1.3 or later, can be restored with precision for single or multiple tables and schema-level disaster recovery.	Tool Guide > Server Tool > GaussRoach.py > Functions Tool Guide > Server Tool > GaussRoach.py > Constraints
Physical fine-grained backup supports restoration of permissions and comments (8.2.1.230)	• If fine-grained table permission restoration fails, the failure information is recorded in the backup set directory, but it will not impact the restoration process. • Schema comments cannot be restored during fine-grained restoration of a single table or multiple tables. • The **dump-options** parameter is added and CLI parameters are configured to specify the backup permission and comment.	Tool Guide > Server Tool > GaussRoach.py > Constraints Tool Guide > Server Tool > GaussRoach.py > Command Reference > backup/restore

SQL Syntax Changes

Table 1-17 SQL syntax changes

Change Type	No.	Item	Description
Added	1	obs_null_file	The table-level parameter **obs_null_file** is added for management and control. When this parameter is set to on, an error is reported when a non-existent file or an invalid path is detected during the import process into GaussDB(DWS).
	2	replace_illegal_chars	The GDS foreign table option **replace_illegal_chars** is added to replace 0x00 import.
	3	DISCARD GLOBAL TEMP [TABLE]	Clears global temporary tables within the current session. When executed without specifying a table name, it will remove all global temporary tables associated with the session. If a specific table name is provided, only that particular global temporary table will be cleared.
	4	ALTER SEQUENCE [IF EXISTS] name CACHE cache	The **cache** value of a sequence can be changed.
	5	CREATE STATISTICS	Creates an extended statistics object for a table.
	6	DROP STATISTICS	Deletes an extended statistics record.
	7	ALTER STATISTICS	The ALTER syntax for extended statistics is added.
	8	The keyword **MASKED** is added to **ALTER FUNCTION**.	Added the syntax for setting whether a function supports non-masking.
	9	last_value ignore nulls	The **last_value ignore nulls** syntax is added to exclude NULL values during calculations.

Change Type	No.	Item	Description
	10	ALTER MATERIALIZED VIEW qualified_name OWNER TO RoleId	Changes the owner of the materialized view.
	11	CREATE MATERIALIZED VIEW	Creates a materialized view.
	12	ALTER MATERIALIZED VIEW	Modifies the properties of a materialized view.
	13	DROP MATERIALIZED VIEW	Deletes a materialized view.
	14	REFRESH MATERIALIZED VIEW	Refreshes a materialized view.
Modified	15	UNIQUE NULLS NOT DISTINCE and UNIQUE NULLS IGNORE	**UNIQUE NULLS NOT DISTINCE** and **UNIQUE NULLS IGNORE** are prohibited for use when defining a foreign table. The unique constraints applied to foreign tables serve only as informational constraints and do not result in the creation of an underlying index. Therefore, these settings, which pertain to the treatment of NULL values in unique constraints, are inapplicable in this context.
	16	**MERGE INTO** is prohibited for views containing triggers	When a view contains a trigger and a **MERGE INTO** operation is performed, the view remains unaltered. It is advisable to avoid using **MERGE INTO** in such cases.
	17	discard all, discard temp, discard sequences, and discard plans	These parameters are used to release temporary resources related to the current session and reset them to the initial state.
	18	CREATE/ALTER RESOURCE POOL	The **weight** option is added to set the network weight.
	19	CREATE/ALTER EXCEPT RULE	Added the bandwidth exception identification and penalty behaviors for exception handling.

Change Type	No.	Item	Description
	20	LOCK [TABLE] {[ONLY] name [, ...]| {name [*]} [, ...]} [IN <lockmode> MODE] [NOWAIT] [LOCAL COORDINATOR ONLY];	The 8.2.0 version introduces syntax specifically for locking the local CN. Initially, it supports only the ACCESS SHARE lock mode. Following this update, the system accommodates all eight levels of lock modes.
	21	[BEFORE | AFTER] old_policy_name for CREATE REDACTION POLICY	The new **[BEFORE|AFTER] old_policy_name** option has been added. If not specified, the default behavior is to create the new policy subsequent to the one with the highest policy_order for the given table object.
	22	timestampdiff()	The problem that the result is incorrect when the time difference is negative is resolved.
	23	In full join scenarios, the check scope of volatile function rewriting is narrowed down.	In scenarios involving a full join across multiple tables, the current version streamlines the process by examining only the **volatile** attribute of functions following the **ON** condition. Additionally, previous limitations on the rewriting of certain volatile functions have been lifted.
	24	merge into	Partitions can be specified.

Keywords

Table 1-18 Keywords

Change Type	No.	Item	Description
Added	1	FACT	The keyword **FACT** is added to the HINT syntax. Consequently, the **FACT** cannot be used as an alias within HINT.

Change Type	No.	Item	Description
	2	LIGHT	A non-reserved keyword **LIGHT** has been added to the **Analyze (light) table** statement to indicate that dynamic sampling is initiated manually.

System Catalogs

Table 1-19 System catalogs

Change Type	No.	Item	Description
Added	1	PG_STAT_OBJECT	Thread statistics and autovacuum effect are recorded to the **pg_stat_object** system catalog.
	2	PG_PLAN_BASELINE	The binding relationships between SQL statements and plans are save in this catalog
	3	PG_PROC_REDACT	A system catalog is added to record the manual non-masking functions.
Modified	4	GS_WLM_SESSION_INFO	The **stmt_type** field is added to identify the query statement type.
	5	GS_RESPOOL_RESOURCE_HISTORY	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	6	GS_WLM_USER_RESOURCE_HISTORY	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	7	PG_REDACTION_POLICY	The **policy_order** field is added to record the priority of masking policies associated with the same table object. A larger sequence number indicates a later creation and a higher priority.

Change Type	No.	Item	Description
	8	PG_REDACTION_COLUMN	The **policy_oid** field is added to record the masking policy corresponding to the masking column details. Starting with version 8.2.1, the relationship between masking policies and masking columns is one-to-many, while the relationship between masking policies and table objects is many-to-one.
	9	PG_STAT_OBJECT	Renamed the reserved field **extra1** to **last_autovacuum_csn**.
	10	GS_WLM_SESSION_INFO	The **unique_plan_id**, **sql_hash**, **plan_hash** and **use_plan_baseline** fields are added.

System Functions

Table 1-20 System functions

Change Type	No.	Item	Description
Added	1	SHOW_TSC_INFO()	Queries the TSC conversion information of the current node.
	2	GET_TSC_INFO()	Re-acquires stable TSC conversion data for the current node without updating memory.
	3	TEST_TSC_INFO(time double, loops int)	Tests time conversion accuracy with constraints: time <= 60s, 1 <= loops <= 10.
	4	GET_FIRST_VALUE	Returns the first row's value from the current column.
	5	READ_GLOBAL_VAR	Accesses global session-level variables (e.g., **my.var**).
	6	GS_QUERY_PENALTY(query_id)	Implements a manual query downgrade using **query_id**.

Change Type	No.	Item	Description
	7	CURRENT_TEMP_SCHEMA	Displays the temporary schema of the current session.
	8	GS_WAIT_CURRENT_ACTIVE_DDL_COMPLETE	Awaits the application of GUC parameters across all threads.
	9	PGXC_WAIT_CURRENT_ACTIVE_DDL_COMPLETE	Awaits the application of GUC parameters across all threads.
	10	GS_SWITCH_PART_RELFILENODE	Swaps filenodes between two specified partitions.
	11	REFRESH_HOT_STORAGE(text)	Transfers all data from a specified hot-cold table to OBS. The return result is the count of cold partitions on the DN.
	12	REFRESH_HOT_STORAGE(text, text)	Transfers specified partition data of a hot-cold table to OBS. The return result is the count of cold partitions on the DN.
	13	PG_QUERY_AUDIT_DETAILS	Invokes **pgxc_query_audit** or **pg_query_audit** to parse the **object_details** and **object_name** fields.
	14	PGXC_HSTORE_DELTA_INFO	Queries HStore Delta table records and table sizes from the CN for troubleshooting.
	15	COPY_PARTITION_STATS	Copies statistics from one table's partition to another without stats.
	16	UPDATE_PARTITION_RELSTATS	Updates **relpages** and **reltuples** stats for a partitioned table.
	17	PG_GET_STAT_EXPRESSIONS	Retrieves textual expressions from created expression statistics.

Change Type	No.	Item	Description
	18	GS_HSTORE_COMPACTION	Initiates manual compaction of an hstore table, independent of **autovacuum_compaction_rows_limit**. The first parameter specifies the target table name, while the second parameter defines the small CU rows limit, defaulting to **100** if unspecified.
	19	PG_SCAN_RESIDUALFILES	Scans residual files.
	20	PG_GET_SCAN_RESIDUALFILES	Retrieves the list of scanned residual files.
	21	PGXC_LOCK_WAIT_STATUS()	Queries cluster lock wait relationships.
	22	PG_CANCEL_BACKEND (pid, msg)	Sends a custom error message when a PID session is interrupted.
	23	PG_GET_STATISTICSOBJDEF	Obtains the statements for creating expression statistics.
	24	PG_STAT_GET_ALL	Returns table-related tuples from **pg_stat_object** on the current CN. The input parameters are **namespace** and **relname**.
	25	PGXC_STAT_GET_ALL	Returns table-related tuples from **pg_stat_object** across all CNs. The input parameters are **namespace** and **relname**.
	26	PGXC_STAT_OBJECT_BYNAME	Quickly returns the relname records from **pgxc_stat_object**. The input parameters are **namespace** and **relname**.
	27	PG_DUMP_PLANMGMT_INFO	Updates saved plans for subsequent binding, unbinding, and dropping operations.

Change Type	No.	Item	Description
	28	PG_FOREIGN_INSERT_COMMIT	Used by the CN to deliver a rename plan to DNs when an insert operation is performed on an object in the external schema table.
	29	PG_SYSTEM_WITH_TOAST_CREATE	Creates a toast table for the **pg_external_namespace** system catalog.
	30	PG_GET_EXTERNAL_SCHEMA_TABLE_OPTIONS	Obtains the table object options of an external schema.
	31	PG_GET_EXTERNAL_SCHEMA_TABLE_COL	Obtains the table column metadata of external schemas.
Modified	32	GS_WLM_REBUILD_SCHEMA_HASH(oid)	The input parameter is changed from **int** to **OID**.
	33	GS_WLM_ALL_USER_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	34	GS_GET_RESPOOL_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	35	GS_WLM_USER_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed**, and the input parameter **username** are added.
	36	PG_STAT_GET_WLM_REALTIME_SESSION_INFO	The **except_info** column is added to display information about exception rules triggered by statements.

Change Type	No.	Item	Description
	37	GS_TABLE_DISTRIBUTION	• The **relpersistence** field is added. – **t**: local temporary table – **g**: global temporary table – **u**: unlogged table – **p**: common table • The **sessionid** field is added to indicate the session thread IDs of the global temporary table.
	38	PG_FAST_TABLE_SIZE	• The **relpersistence** field is added. – **t**: local temporary table – **g**: global temporary table – **u**: unlogged table – **p**: common table • The **sessionid** field is added to indicate the session thread IDs of the global temporary table.
	39	PG_LIFECYCLE_TABLE_DATA_DISTRIBUTE	Updated to allow access for non-administrator users.
	40	PG_QUERY_AUDIT	The **object_details**, **result_rows**, and **error_code** columns are added to audit logs. The **pg_query_audit output** parameter is modified, and three columns are added.
	41	PGXC_QUERY_AUDIT	The **object_details**, **result_rows**, and **error_code** columns are added to audit logs. The **pgxc_query_audit** parameter is modified, and three columns are added.

Change Type	No.	Item	Description
	42	GET_COL_CU_INFO	• The **dirty_percent** parameter is added. The value ranges from **1** to **100**. The default value is **70**. • New column **dirty_cu_count** returns the count of CUs with a deletion rate above **dirty_percent**.
	43	PG_STAT_GET_WLM_REALTIME_OPERATOR_INFO	The following fields are added: **parent node ID**, **execution times**, **progress**, **network**, and **disk read/write**.
	44	PG_STAT_GET_WLM_REALTIME_OPERATOR_INFO	The input parameter **queryid** is added to query the statement information of a specified query ID.
	45	TRUNC(timestamp with time zone)	The stability type is modified to **stable** from **immutable** to improve performance in certain scenarios.
	46	PG_STAT_GET_WLM_REALTIME_SESSION_INFO	The **stmt_type** column is added to identify the statement type.
	47	PG_STAT_GET_WLM_REALTIME_SESSION_INFO	The **unique_sql_id** field is added.

System Views

Table 1-21 System views

Change Type	No.	Item	Description
Added	1	SHOW_TSC_INFO	Records the TSC conversion information of the current node.
	2	SHOW_ALL_TSC_INFO	Records the TSC conversion information of all nodes.

Change Type	No.	Item	Description
	3	GET_TSC_INFO	Re-acquires stable TSC conversion data for the current node without updating memory.
	4	GET_ALL_TSC_INFO	Re-acquires stable TSC conversion data for all nodes without updating memory.
	5	PGXC_WLM_TABLE_DISTRIBUTION_SKEWNESS	Records disk space skew of physical tables on DNs.
	6	PGXC_STAT_OBJECT	Records the **pg_stat_object** information of all nodes in the cluster:
	7	PG_GLOBAL_TEMP_ATTACHED_PIDS	Records information about sessions of resources occupied by global temporary tables on the current node.
	8	PGXC_GLOBAL_TEMP_ATTACHED_PIDS	Records information about sessions of resources occupied by global temporary tables on all nodes.
	9	PG_STATS_EXT_EXPRS	Records expression statistics stored in the **pg_statistic_data** system catalog.
	10	PV_RUNTIME_EXPRSTATS	It is created based on the **pg_stat_get_runtime_exprstats** function and is used to query the expression statistics generated by dynamic sampling.
	11	PG_PLAN_BASELINE	Records the associations between SQL statements and plans.
	12	PG_STAT_OBJECT_EXT	Accelerates the return of the query conditions in **pgxc_stat_object** by adding query conditions to the inner layers of **pgxc_parallel_query**.
Modified	13	GS_WLM_SESSION_STATISTICS	The **stmt_type** column is added to identify the statement type.

Change Type	No.	Item	Description
	14	PGXC_WLM_SESSION_STATISTICS	The **stmt_type** column is added to identify the statement type.
	15	GS_WLM_SESSION_HISTORY	The **stmt_type** column is added to identify the statement type.
	16	GS_WLM_SESSION_INFO	The **stmt_type** column is added to identify the statement type.
	17	PGXC_WLM_SESSION_HISTORY	The **stmt_type** column is added to identify the statement type.
	18	PGXC_WLM_SESSION_INFO	The **stmt_type** column is added to identify the statement type.
	19	GS_RESPOOL_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	20	PGXC_RESPOOL_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	21	PG_TOTAL_USER_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	22	PGXC_TOTAL_USER_RESOURCE_INFO	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	23	PGXC_RESPOOL_RESOURCE_HISTORY	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.
	24	PGXC_WLM_USER_RESOURCE_HISTORY	The network sending and receiving rate monitoring metrics **send_speed** and **recv_speed** are added.

Change Type	No.	Item	Description
	25	GS_WLM_SESSION_STATISTICS/ PGXC_WLM_SESSION_STATISTICS	The **except_info** column is added to display information about exception rules triggered by statements.
	26	GS_WLM_SESSION_INFO/ PGXC_WLM_SESSION_INFO	The **except_info** column is added to display information about exception rules triggered by statements.
	27	GS_WLM_SESSION_HISTORY/ PGXC_WLM_SESSION_HISTORY	The **except_info** column is added to display information about exception rules triggered by statements.
	28	REDACTION_COLUMNS	The view definition of **redaction_columns** and the **policy_name** field is added.
	29	PG_STATS	The **partname** field is added to indicate partitions.
	30	PV_RUNTIME_ATTSTATS	The **partname** field is added to the **pg_catalog.pv_runtime_attstats** view to indicate partitions.
	31	GS_WLM_OPERATOR_STATISTICS	The following fields are added: **parent node ID**, **execution times**, **progress**, **network**, and **disk read/write**.
	32	PGXC_WLM_OPERATOR_STATISTICS	The following fields are added: **parent node ID**, **execution times**, **progress**, **network**, and **disk read/write**.
	33	PGXC_STAT_OBJECT	Changed with **pg_stat_object**. The reserved field **extra1** is renamed **last_autovacuum_csn**.

Behavior Changes

Table 1-22 Behavior changes

Change Type	No.	Item	Description
Added	1	Partition view dependency changes	Before 8.2.1: No view dependency for **SELECT PARTITION()** or **PARTITION FOR()** views, causing query errors. 8.2.1 and later: Partition OID dependency added. Views are rebuilt if partitions change in view decoupling scenarios and errors are reported in non-view dependency scenarios.
Modified	2	gtm_max_trans	Increased upper limit to accommodate higher thread counts for emergency streams, addressing service concurrency issues in large clusters.
	3	max_process_memory	Reduced the number of CNs to half of DNs to optimize memory usage.
	4	Catch-up TV process	Removed data page transaction lock from Catchup, distinguishing it from DDL. Added LwLock to serialize operations between Catchup and DDL services, preventing access to empty files.
	5	Standby DN checkpoint time	Checkpoints are no longer executed after a 15 minute period. Now they are executed immediately after redo operations reach the checkpoint to reduce RTO during large Xlog imports.
	6	Standby DN file closure	The home page on standby DN is forcibly closed after executing heap_xlog_newpage and seq_redo, ensuring data integrity.

Change Type	No.	Item	Description
	7	Indexscan hints	Expanded to include indexonlyscan. When both indexscan and indexonlyscan hints are present, indexonlyscan is prioritized.
	8	Case sensitivity in expressions	Discontinued support for roughcheck due to CU minmax calculations based on C sorting, which could lead to result set discrepancies.
	9	Multiple count(distinct) rewriting	Changed rewriting behavior. In non-share scan scenarios, subsequent expansion is unaffected. In share scan scenarios, a stream CTE plan is generated.
	10	Top SQL substatement monitoring	Monitoring approach altered from recursive to recording only the first-layer substatements.
	11	ArrayLockFreeQueue	Initial memory allocation for lock-free queues capped at 1 GB.
	12	Explain	The **unique SQL ID** is displayed in the Explain result set.
	13	Explain	The **unique SQL ID** of "Explain + SQL statement" is the same as that of the SQL statement.
	14	GS/ PGXC_WLM_SESSION_STATISTICS view	Added **unique SQL ID** field.

Change Type	No.	Item	Description
	15	GROUP BY query fields	• In version 8.1.3, the query field for non-aggregate functions is supplemented to the **min(xx)** format. From version 8.2.0 onwards, a random value is used in this column. • In version 8.2.0, query fields use non-null values for multi-column non-aggregation functions, leading to misaligned rows in the result set. In version 8.2.1, rows in the result set are aligned.
	16	alter function owner to superuser	Restricted **sysadmin** users from changing function or stored procedure owners to a system user, except when changing from a system user to another system user.
	17	Plus sign (+) connection sequence	The connection sequence of the plus sign (+) is aligned with Oracle. The original connection sequence of the plus sign (+) is related to the join association sequence. After the modification, there will be change in the join condition sequence and result set of related expressions, such as nvl expressions.
	18	Concurrent updates in hstore tables	When the same row is concurrently updated in an hstore table, an error is immediately reported. The previous behaviors are as follows: 1). If a line is concurrently updated, an error is reported after waiting. 2). If a line is concurrently delete, no errors are reported after waiting. 3). If a line is concurrently updated and deleted, an error is reported after waiting. The behaviors are changed to the same.

Change Type	No.	Item	Description
	19	Catch-up lock level	Catchup no longer holds a level 1 lock.
	20	Continuous failover logic	In consecutive failovers, data is synchronized from the secondary DN to the standby DN only for the first time.
	21	Unique SQL IDs for temporary tables	Prior to version 8.2.1, temporary tables sharing the same name across different sessions had distinct schema names, resulting in distinct unique SQL IDs for each table. With the introduction of version 8.2.1, schema names in such temporary tables are standardized using fixed strings, ensuring uniform unique SQL IDs for identical SQL statements executed in different sessions.
	22	Fine-grained DR in **degraded** state	Fine-grained DR supports backup in the degraded state of the primary cluster, expect scenarios where the primary node lacks a normal CN.
	23	View decoupling	When an invalid view is accessed, the automatic rebuilding action is not triggered. The system catalog is not updated but expanded locally.

GUC Parameters

Table 1-23 GUC parameters

Change Type	No.	Item	Description
Added	1	time_track_strategy	Specifies the time statistics method of non-vectorized operators, including **tsc**, **vector**, **timer**, and **frequency** (debug mode)

Change Type	No.	Item	Description
	2	max_skew_num	Controls the number of skew values allowed by the optimizer for redistribution optimization.
	3	disable_full_group_by_mysql	In MySQL compatibility mode, this parameter permits non-aggregate function query fields post **GROUP BY** when enabled.
	4	enable_col_index_vacuum	Controls autovacuum of column-store indexes, defaulted to **false**.
	5	enable_pg_stat_object	Records the **pg_stat_object system** catalog when enabled; remains unrecorded when disabled.
	6	conn_recycle_timeout	Determines the interval for connection reclamation.
	7	index_selectivity_cost	Controls the cost calculation for cbtree during index scanning of column-store tables, applicable when the selection rate is ≥ 0.001.
	8	default_table_behavior	Controls default behavior of a table. In versions earlier than 8.2.1, only **column_btree_index** is supported (the default index created for a column-store table is btree).
	9	resource_track_subsql_duration	For TopSQL monitoring, it records substatements in stored procedures exceeding the specified threshold.
	10	max_files_per_node	Limits the number of files a single query can open per node.
	11	max_copy_data_display	Specifies the maximum character display for the **rawrecord** field in the COPY error table.
	12	max_process_memory_balanced	Sets the upper limit for **max_process_memory** in primary/standby load balancing mode.
	13	index_cost_limit	Controls the cost calculation of cbtree during column-store table index scanning (the selection rate is greater than or equal to 0.001). Hints can be set.

Change Type	No.	Item	Description
	14	audit_object_details	Determines whether to log the **object_details** column in audit records.
	15	enable_fd_check	Checks for incorrect FD shutdowns, enabled by default.
	16	idle_in_transaction_timeout	Previously, after session_timeout was set to 0, there might be long idle connections holding table locks. To solve this problem, the transaction-level parameter **idle_in_transaction_timeout** is added to set the idle timeout interval for idle transaction connections. 1. It is set by users. The unit is second. The default value is 0, indicating that this function is disabled. The value ranges from 0 to 86400 (one day). 2. This parameter is valid only for client connections that are directly connected to CNs. It does not applicable to direct connections to DNs or internal connections. 3. After the value of this parameter is changed to a non-zero value, a FATAL error is reported when the transaction is in the idle state for a period longer than the specified value.
	17	enable_save_data access_timestamp	Records the last access time at the table level when enabled. It is disabled by default.

Change Type	No.	Item	Description
	18	behavior_compat_options	Added **check_function_shippable** to the **behavior_compat_options** parameter. If **check_function_shippable** is enabled, the function **ship** is checked. 1. An error occurs when attempting to push down the outer layer containing DML statements. 2. If the outer layer can be shipped and the inner layer is immutable, no errors will be reported. 3. No errors will be reported if both the outer and inner layers are shippable. 4. An error will be reported if the outer layer is shippable and the inner layer is neither immutable nor shippable.
	19	enable_release_scan_lock	The USERSET parameter **enable_release_scan_lock** is added. When this parameter is enabled, the level-1 lock is released after the statement is executed.
	20	job_queue_naptime	The **job_queue_naptime parameter** is added to set the interval for triggering a task and the timeout interval. The default value is 1s.
	21	enable_stream_sync_quit	The GUC parameter is added. By default, it is disabled to ensure that the sync quit command is not blocked.
	22	**enable_full_string_agg** option of **behavior_compat_options**	The **enable_full_string_agg** option is added to **behavior_compat_options**. Enabling this parameter allows the **string_agg()** function to execute full aggregation within a window when used with **OVER (PARTITION BY xx ORDER BY xxx)**. Conversely, if this parameter is disabled, **string_agg()** will perform incremental aggregation in the same context. By default, this parameter is set to disabled.

Change Type	No.	Item	Description
	23	**enable_cast_hashjoin** option of behavior_compat_options	The **enable_cast_hashjoin** parameter is added to **behavior_compat_options**. When enabled, this parameter facilitates type conversion in JOIN conditions, such as between timestamp and timestampz, to enable efficient hash joins. By default, this parameter is disabled.
Modified	24	behavior_compat_options	The parameter **DISABLE_SET_GLOBAL_VAR_ON_DATANODE** has been added to the **behavior_compat_options** to ensure that global variables cannot be set on DNs.
	25	hashjoin_spill_strategy	Policies 5 and 6 have been implemented, mirroring the functionalities of policies 0 and 1, respectively. However, there have been modifications to the latter: now, if an internal table exceeds memory capacity, it will be subdivided recursively until further division is not possible. Subsequently, the system will initiate a swap between the internal and external tables.
	26	behavior_compat_options	The **behavior_compat_options parameter** now includes the **disable_gc_fdw_filter_partial_pushdown option**, which governs the filter conditions applied to foreign tables during collaborative analysis. Enabling this option means that if any condition is non-pushable, none will be pushed down. Conversely, disabling it allows for partial pushdown of conditions.
	27	prefer_hashjoin_path	When the **prefer_hashjoin_pathda** parameter is enabled, the system prioritizes the creation of a hash join path. This may cause execution plans with the same cost to change.

Change Type	No.	Item	Description
	28	volatile_shipping_version	The configuration now includes the value 3 to regulate the pushdown behavior of volatile functions. Meanwhile, value 2 has been updated to facilitate the pushdown of volatile functions within replicated Common Table Expressions (CTEs). Furthermore, the pushdown of volatile functions in CTEs is now disabled in scenarios where share scan is not used.
	29	behavior_compat_options	The **ignore_unshipped_concurrent_update** option is added to control whether to ignore new tuple detection in concurrent update scenarios.
	30	max_connections	The minimum value is changed from 1 to 100.
	31	max_process_memory	The setting takes effect directly without depending on whether **max_process_memory_auto_adjust** is enabled.
	32	index_selectivity_cost	The forward compatibility of column-store **indexcost** is rectified to prevent the default plan from changing. The default value of this parameter is changed to **-1**, and hints can be set.
	33	behavior_compat_options	The option **enable_pushdown_groupingset_subquery** has been added. By default, if a subquery includes a grouping set, the outer condition is not eligible for pushdown into the subquery. However, when this parameter is enabled, it allows the outer condition to be pushed down into the subquery. It is important to verify the accuracy of the results after the pushdown is performed.
	34	comm_max_stream	**comm_max_stream** supports reload.
	35	enable_tsdb_multi_temperature	This parameter specifies whether cold and hot tables can be created for time series tables. The default value is changed to **off**.

Change Type	No.	Item	Description
	36	enable_col_index_vacuum	The default value of **enable_col_index_vacuum** is changed to **true**. By default, **autovacuum** is allowed to clear column-store indexes.
	37	enable_redistribute	This parameter controls the query optimizer's use of data transmission in **local redistribute** and **split redistribute** redistribution modes. It is not used in versions earlier than 8.2.1.
	38	time_track_strategy	• The default value of the kernel is changed to **timer**, and the guc parameter level is set to **sighup**. • Parameter policies are set on the OM side. After the default timer, is upgraded, if TSC is supported during, set this parameter to **TSC**. Otherwise, set this parameter to **vector**.
	39	behavior_compat_options	The value **enable_banker_round** is added to control the use of the banker algorithm.
	40	behavior_compat_options	The **orderby_null_first** parameter is added to determine if null values should be treated as the lowest values when performing **order by** sorting operations. It is important to note that this parameter is only applicable to TD-compatible databases.
	41	behavior_compat_options	The parameter **alter_distribute_key_by_partition** is added to enable the distribution of data across partitions during an **ALTER TABLE** operation that uses the **DISTRIBUTE BY** clause. When enabled, this parameter ensures that the **INSERT INTO** command targets specific partitions. Conversely, disabling this parameter will maintain the default behavior, where **INSERT INTO** affects the entire partitioned table.

Change Type	No.	Item	Description
	42	behavior_compat_options	The parameter **enable_use_syscol_in_replicate_table** is added. If this parameter is not configured, utilizing system columns such as **oid**, **ctid**, **tableoid**, or **xc_node_id** as filters, join conditions, or within the **HAVING** clause for operations like **INSERT**, **UPDATE**, **MERGE INTO**, or **DELETE** will trigger an error message.
	43	behavior_compat_options	The parameter **enable_force_add_batch** is added to enhance control over batch processing. When **support_batch_bind** is set to **on**, and both **enable_fast_query_shipping** and **enable_light_proxy** are set to **off**, GaussDB(DWS) receives U packets in addbatch mode. It is important to be aware that this mode may lead to slower packet importation into the database, potentially resulting in memory shortages. Therefore, careful consideration is advised when configuring this parameter.
	44	behavior_compat_options	The **disable_update_returning_check** option is added to the **behavior_compat_options parameter**. This addition provides control over whether updates to the replication table should include the **RETURNING** clause.
	45	cost_model_version	The value 3 is added, which signifies an enhancement to the broadcast cost estimation for large cluster environments. This optimization builds upon the existing value 2, enabling the optimizer to more effectively select superior execution plans.
	46	enable_track_record_subsql	The default value of this parameter is changed to on.

Change Type	No.	Item	Description
Deleted	47	enable_grant_public	This parameter specifies that the **grant to public** syntax is not supported in security mode and is deleted in versions later than 8.2.1.
	48	enable_grant_option	This parameter specifies that the **grant with grant option** syntax is not supported in security mode and is deleted in versions later than 8.2.1.

1.5.2 New Features and Resolved Issues in 8.2.1.x

8.2.1.230

Table 1-24 New features/Resolved issues in version 8.2.1.230

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	Fine-grained backup and restoration support online DDL operations, allowing tables to be modified while performing backups.	-	-	-
	Fine-grained table-level restoration allows tables to be restored to a heterogeneous cluster, regardless of any topology consistency between the target and restored clusters.	-	-	-
	Fine-grained backup and restoration support cross-version restoration.	-	-	-

Category	Feature or Resolved Issue	Cause	Version	Handling Method
	Cluster-level and schema-level physical fine-grained backup support the backup of permissions and comments.	-	-	-
	During the redistribution phase of online scale-out, the priority can be adjusted dynamically.	-	-	-
Resolved issues	Catchup conflicts with the DDL service lock and can last a long time.	If DDL operations are performed in a stored procedure, the catchup operation might not end before the transaction is submitted, resulting in a lock timeout error.	8.0.x	Upgrade to 8.2.1.230.
	The **WITH RECURSIVE** statement runs indefinitely.	In the Arm environment, if thread information synchronization is disrupted, variables may not be updated at the same time.	8.1.3.322	
	The memory usage of **gs_wlm_readjust_relfilenode_size_table** is high.	The **pg_relfilenode_size** table is completely loaded to the memory, occupying too much memory.	8.1.3.323	
	As the cluster runs for an extended period, the memory usage of **TopMemoryContext** rises.	After the stream thread returns to the thread pool, there is a delay in releasing the memory, resulting in a gradual increase in the **TopMemoryContext** memory usage.	8.2.1.22	

Category	Feature or Resolved Issue	Cause	Version	Handling Method
	The SQL statement execution stops unexpectedly, and the error message " canceling statement due to coordinator request" is displayed.	An error occurs when executing a statement with the stream operator. The cancel message is sent to the substream thread, which has already returned to the stream thread pool. The next query reuses the stream thread. Due to the lack of strong consistency verification on the query ID, the residual signal from the previous statement is responded to, causing the statement to terminate abnormally.	8.1.3.110	
	During the schema space query, the value of **usedspace** exceeds that of **permspace**.	When determining the maximum limit of the schema space, the system compares the used space with the limit. Consequently, the actual usage surpasses the limit.	Versions earlier than 8.2.1.230	
	The service fails to be executed when the **max_files_per_node** parameter is set to **-1**.	During SQL execution, when a stream thread is created, the system reads the **max_files_per_node** parameter, which defaults to **50,000**. An error message indicates that the number of handles exceeds the limit. So even if **guc** is set to **-1**, it does not take effect.	8.1.3.321	

Category	Feature or Resolved Issue	Cause	Version	Handling Method
	The error message "Stream plan check failed." is displayed during SQL statement execution. Execution datanodes list of stream node mismatch in parent node.	In the process of generating a plan, any changes made to a plan node will affect all nodes, as the lower-layer plan node relies on the upper-layer plan node.	Versions earlier than 8.2.1.230	
	Service statements cannot be terminated when **xc_maintenance_mode** is disabled during redistribution after scale-out.	During the scale-out redistribution phase, functions like **pg_cancel_query** and **pg_cancel_backend** can only be used when **xc_maintenance_mode** is enabled, meaning user service statements cannot be terminated during redistribution.	Versions earlier than 8.2.1.230	

8.2.1.225

Table 1-25 New features/Resolved issues in version 8.2.1.225

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-

Category	Feature or Resolved Issue	Cause	Version	Handling Method
Resolved issues	Replacement of GDS invalid characters fails.	When invalid GDS characters are replaced with special characters (), an exception occurs. This is because the replacement changes the string length, but the original length is still used in subsequent processing. As a result, some characters are truncated and cannot be replaced correctly.	Versions earlier than 8.2.1.225	Upgrade to 8.2.1.225.
	During concurrent pressure tests, **gather** performance occasionally deteriorates.	If a statement includes the stream operator, multiple stream threads are generated on DNs. The topConsumer thread, which integrates substream thread data and sends it to the CN, can only clear the stream thread group after all substream threads exit.	Versions earlier than 8.2.1.225	

8.2.1.223

Table 1-26 New features/Resolved issues in version 8.2.1.223

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-

Type	Feature or Resolved Issue	Cause	Version	Handling Method
Resolved issues	Cluster hang detection triggers a switchover.	Before signal reconstruction, the unreliable SIGUSR2 was used for IPC. After reconstruction, reliable signals 34 and 35 are used. However, sending too many signals increases the likelihood of timer creation failures.	8.2.1.220	Upgrade to 8.2.1.223.
	A core dump (GsCgroupIsClass) occurs when pgxc_cgroup_reload_conf messages are sent concurrently.	This happens because of unlocked pointer access. When the reload function modifies the pointer, it results in a wild pointer access and a core dump.	8.2.1.220	
	The table size reported by the **gs_table_distribution** function differs significantly from the actual size.	This happens when data in the **pg_refilenode_size** system catalog is read in batches and calculated, causing repeated accumulation of the current batch's table size.	8.2.1.220	
	Executing an SQL statement may result in the error "Could not open file 'pg_clog/000000000075'".	After **VACUUM FULL** on a column-store table, clogs may be prematurely reclaimed, making them inaccessible during **ANALYZE** after an active/standby switchover.	8.2.1.119	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	The issue of freememory showing a large negative value due to uninitialized temporary variables has been resolved.	Declaring a temporary variable without assigning a value led to unexpected parameter values and excessively negative memory usage, causing test case failures when the network adapter was faulty.	8.2.1.220	
	After configuring **VACUUM FULL** for intelligent O&M, the actual execution time can exceed the configured range.	When the scheduler kills the **VACUUM FULL** task, a new task is inserted, preventing complete execution of the kill task.	8.1.3.x	

8.2.1.220

Table 1-27 New features/Resolved issues in version 8.2.1.220

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	• **MERGE INTO** allows for specified partitions. • Plan management is supported. • GDS now supports the fault tolerance parameter **compatible_illegal_chars** for exporting foreign tables. • The window function **last_value** supports the **ignore nulls** feature.	-	-	-
Resolved issues	The SQL statement's execution is unstable and slow, with **pgxc_thread_wait_status** showing **HashJoin - nestloop** for extended periods.	Each partition group has about 10,000 rows, causing prolonged nestloop execution due to data variations.	8.1.3.300	Upgrade to 8.2.1.220.
	The database's large number of objects leads to slow performance and high memory usage during queries.	This is mainly due to numerous tables with the **internal_mask** option in column storage mode, causing inefficient permission verification.	Versions earlier than 8.2.1.119	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Excessive expressions during LLVM compilation result in high CPU usage.	Enabling LLVM and having a large number of expressions can result in prolonged execution times. Disabling LLVM reduces execution time from several hours to just over 10 minutes when expressions exceed 1,000.	8.1.3.320	
	The cursor fetches 2,000 records each time, with memory usage exceeding estimates by 24 MB per fetch. As a result, when the total number of data records reaches 20,000,000, the query execution fails.	In PBE scenarios, the previously generated plan is reused, and the estimated memory increases by a fixed value each time. This can lead to memory overestimation and CCN queuing.	8.1.3.323	
	Memory leakage during JSON-type queries causes high memory usage.	This happens when there is unreleased memory in the jsonb **out** function.	8.1.3.x	
	Executing **SELECT * FROM WITH** clause in customer service SQL statements causes a CN core dump.	ProjectionPushdown updates the **rte** but fails to update the **var** based on the new **rte**, leading to a core dump during **quals** processing.	8.1.3.323	
	Overflow occurs when the number of **WHERE** conditions in a **DELETE** statement exceeds the upper limit.	The number of **WHERE** conditions in the **DELETE** statement exceeds the int16 limit of 32,767. This causes an overflow and results in a core dump.	8.1.2.x	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	During scale-out, the redistribution process restarts, causing a suspension of over an hour when generating a table list.	To generate a table list, the system catalog is queried on the CN and **INSERT INTO** needs to be executed for each record inserted into the distributed table **pgxc_redistb**. However, if there are numerous tables, using the **VALUES** statement can be time-consuming.	8.1.3.110	
	CN memory leakage occurs during transaction rollback caused by primary key conflicts.	1. In JDBC, the PBE protocol inserts data through CN lightweight, but primary key conflicts cause errors. 2. A transaction with multiple cross-CN lightweight queries or unnamed statements sent by JDBC saves the global LightProxy object to the portal before execution. If not released after the transaction, it causes memory accumulation. 3. Numerous CachedPlanQuery memory contexts appear in the **pv_session_memory_detail** view of the CN.	8.2.0.103	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Actual memory usage exceeds estimates in **COUNT DISTINCT** and **UNION ALL** scenarios.	In multi-branch serial execution, only the memory when the lower-layer operator returns data is considered, not during its actual execution, leading to underestimated memory usage.	8.1.3.321	
	After a cluster restart, only the first session connection can use SQL debugging, with subsequent connections failing. Breakpoints do not work when debugging SQL statements in DataStudio.	The scheduling entry variable is released after database disconnection, resulting in a null pointer and failed debugging logic.	• 8.1.1.x • 8.1.3.x	
	The **WITH RECURSIVE** statement runs indefinitely in Arm environments.	Abnormal thread information synchronization in the Arm environment can result in variables not being updated synchronously.	8.1.3.323	
	Executing **INSERT OVERWRITE** for a specific partition overwrites the entire table.	In PBE logic, **INSERT OVERWRITE** does not copy partition information, causing **FILENODE** exchange on the entire table.	Versions earlier than 8.2.1.220	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	The subquery result set containing WindowAgg is abnormal.	The subquery result set containing WindowAgg is abnormal. WindowAgg is not considered when generating a bloom filter. If the join association column is not a WindowAgg grouping column, grouping data is reduced, affecting the window function's grouping result.	8.1.3.x	
	Memory insufficiency errors occur and the view shows that the SQL statements with high memory usage are **VACUUM FULL** operations.	Performing **VACUUM FULL** on every partition of a partitioned table prevents memory from being released, causing memory usage to keep growing until an error is triggered.	8.1.3.x	
	Restarting the logical cluster times out.	This happens because of the default use of 10 IP addresses by the CM, requiring dynamic adaptation.	8.2.1.200	
	After a version update, numerous "Wait poll time out" errors occur.	The LibcommCheckWait-Poll function behaves unexpectedly when passed **-1**.	8.2.1.200	

8.2.1.119

Table 1-28 New features/Resolved issues in version 8.2.1.119

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	The **last_value** window function now supports **ignore nulls** and is compatible with Redshift.	-	-	-
Resolved issues	Error when using **try_cast** function on column-store tables.	The **try_cast** function is incompatible with the vectorized executor, causing errors during execution.	8.2.1.100 and earlier versions	Upgrade the version to 8.2.1.119.
	Slow insert operations post-cluster restart.	Indexes are required to scan full data during insert operations after a restart, impacting performance.	8.2.1.100 and earlier versions	
	CCN count abnormalities not triggering calibration, leading to queuing issues.	A bug in code processing prevents the calibration mechanism from activating when CCN counts are abnormal.	8.2.1.100 and earlier versions	
	Primary key conflict error and CN memory leak when inserting data via JDBC using PBE protocol.	The CN lightweight process does not release the lightweight object post-transaction, leading to memory accumulation.	8.2.1.100 and earlier versions	
	Incorrect plan **generation when enable_stream_ctescan** GUC hint is set.	The rollback to a non-ShareScan plan is incomplete when CTE memory usage estimates exceed thresholds, resulting in execution failure.	8.2.1.100 and earlier versions	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Metadata restoration failure when backup size exceeds 64 MB on OBS.	Code vulnerabilities discard the last buffer segment during segmented download, corrupting the metadata.	8.2.1.100 and earlier versions	
	High memory usage during hstore delta table ANALYZE sampling.	The process of delta table combining I records consumes excessive memory. Toast data and delta data deserialization space must be released promptly.	8.2.1.100 and earlier versions	
	Inability to push down volatile functions in single-reference CTE queries.	Version 821 adds constraints against pushing down volatile functions in CTEs, which should be removed for single-reference scenarios to enable pushdown.	8.2.1.100 and earlier versions	
	Excessive space occupation by temporary files in XFS system, causing read-only cluster state.	Each temporary file spilled to disks occupies 16 MB in XFS. Too many spilled files cause cluster read-only state, necessitating a reduction in disk space usage by these files.	8.2.1.100 and earlier versions	

1.6 Version 8.2.0

[Release date of V8.2.0.107] September 25, 2023

[Release date of V8.2.0.106] July 17, 2023

[Release date of V8.2.0.103] May 25, 2023

[Release date of V8.2.0.102] April 15, 2023

[Release date of V8.2.0.101] March 13, 2023

[Release date of V8.2.0.100] January 14, 2023

[Release date of V8.2.0]: November 28, 2022

New features

Table 1-29 New Functions in 8.2.0

Feature	Description	Reference
The pgcrypto encryption extension plug-in is added.	You can use pgcrypto to encrypt data.	Developer Guide > Database Security Management> Sensitive Data Management> Use pgcrypto to Encrypt Data
The **object_name** column in the audit log records all read and write objects in the SQL statement.	• The description about querying and auditing the records of multiple object names is added. • Added the GUC parameter **audit_object_name_format**.	• Developer Guide > Database Security Management > Database Audit > Querying Audit Results • User Guide > Audit Logs > Database Audit Logs > Viewing Database Audit Logs • Developer Guide > GUC Parameters> Auditing> Audit Switch
Certain user operations can be allowed in security mode.	The GUC parameter **security_enable_options** is added to control whether the **grant_to_public**, **grant_with_grant_option**, and **foreign_table_options** functions can be used in security mode.	Developer Guide > GUC Parameters> Auditing> Operation Audit
Session IDs can be recorded in audit logs.	• The configuration item **normal_session_id** is added to determine whether to generate session IDs in normal format. • Modified the **session_id** field in **pg_query_audit()**.	• Developer Guide > GUC Parameters > Miscellaneous Parameters > **behavior_compat_options** • SQL Syntax Reference > Functions and Operators> Security Functions

Feature	Description	Reference
MIXED AGG	The GUC parameter **enable_mixedagg** is added to control how the optimizer uses the Mixed Agg type.	Developer Guide > GUC Parameters> Query Planning> Optimizer Method Configuration

Feature	Description	Reference
The circuit breaker mechanism based on exception rules is optimized to prevent a slow SQL statement from affecting the entire cluster or resource pool.	• Added the system catalog **GS_BLOCKLIST_QUERY** and system view **GS_BLOCKLIST_QUERY** for querying job blocklist information and exception information. • The **query_exception_count_limit** parameter is added to specify the maximum number of times that an exception rule can be triggered for a job. If this upper limit is exceeded, the job will be automatically added to the blocklist and cannot be executed. The job can be resumed only after it is removed from the blocklist. • Functions are added to record job exception information. gs_increase_except_num(unique_sql_id int8) gs_increase_except_num(unique_sql_id int8, except_num int4) gs_increase_except_num(unique_sql_id int8, except_num int4, except_time int8) gs_update_blocklist_hash_info(unique_sql_id int8, is_remove boolean) gs_update_blocklist_hash_info() gs_append_blocklist(unique_sql_id int8)	• Developer Guide >System Catalogs and Views >System Catalogs/System Views • Developer Guide > GUC Parameters > Load Management • SQL Syntax Reference > Functions and Operators > System Administration Functions > Resource Management Functions

Feature	Description	Reference
	gs_remove_blocklist(unique_sql_id int8) gs_wlm_rebuild_except_rule_hash()	

Feature	Description	Reference
Full utilization of memory resources	• The memory negative feedback mechanism is added to the **PGXC_RESPOOL_RESOURCE_INFO** view. • Changed the type of **max_process_memory** from postmaster to sighup. If a single DN is deployed on a server, **max_process_memory** = (Physical memory – **vm.min_free_kbytes**) x 0.6 Added the **max_process_memory_auto_adjust** parameter to control whether to enable the automatic adjustment function of the **max_process_memory** parameter. • Added the **enable_wlm_internal_memory_limit** parameter to specify whether to enable the built-in limit on estimated statement memory usage in load management. Added the **enable_strict_memory_expansion** parameter to determine whether to strictly control the increase of statement memory usage. Added the **allow_zero_estimate_memory** parameter to control whether the estimated statement memory usage can be 0.	• Developer Guide > System Catalogs and System Views > System Views > PGXC_RESPOOL_RESOURCE_INFO • Developer Guide > GUC Parameters > Resource Consumption > Memory • Developer Guide > GUC Parameters > Resource Management

Feature	Description	Reference
The exception rule interface is implemented using the kernel syntax.	• The **PG_EXCEPT_RULE** system catalog is added to store information about exception rules. • Added syntaxes **ALTER EXCEPT RULE**, **CREATE EXCEPT RULE**, and **DROP EXCEPT RULE**.	• Developer Guide > System Catalogs and Views > System Catalogs > PG_EXCEPT_RULESQL • SQL Syntax Reference > DDL Syntax > ALTER/CREATE/DROP EXCEPT RULE
Reconstructed resource management	• The system catalog **GS_WLM_USER_RESOURCE_HISTORY** contains data on both CNs and DNs. • The **PGXC_TOTAL_USER_RESOURCE_INFO** view is added to display real-time resource consumption information of users on all instances. The **PGXC_WLM_USER_RESOURCE_HISTORY** view is added to display historical information about resource consumption of all users on the corresponding instances.	• Developer Guide > System Catalogs and Views > System Catalogs > GS_WLM_USER_RESOURCE_HISTORY • Developer Guide > System Catalogs and System Views > System Views > PGXC_TOTAL_USER_RESOURCE_INFO/ PGXC_WLM_USER_RESOURCE_HISTORY
A parameter is added in the optimizer phase. An error will be reported for the SQL statement run with too many stream threads.	The GUC parameter **max_streams_per_query** is added to control the number of stream nodes in a query plan.	Developer Guide > GUC Parameters > Other Optimizer Options
If a GaussDB(DWS) cluster becomes read-only, you can perform TRUNCATE and DROP to quickly free up disk space and automatically cancel the read-only state.	In 8.2.0 and later versions, you can free up disk space by using **DROP** or **TRUNCATE TABLE** in a read-only cluster.	User Guide > Cluster O&M > Removing the Read-only Status

Feature	Description	Reference
Hints can take effect in subqueries.	• Hint parameters can be used in DML statements, including **INSERT**, **UPDATE**, **MERGE**, and **DELETE**. • Hint enhancement in multi-table scenarios: Hints are enhanced for scenarios where the number of items in the **FROM** list exceeds the threshold, preventing hint failures and improving performance. • The **enable_from_collapse_hint** parameter is added to control whether to preferentially rewrite the **FROM** list with hints in effect.	• Developer Guide > Performance Optimization > Hint-based Tuning > Plan Hint Optimization > Hint That Disables Subquery Pull-up • Developer Guide > GUC Parameters > Query Planning > Other Optimization Options

Feature	Description	Reference
Dynamic sampling	• The **autoanalyze_mode** parameter is added to control whether **autoanalyze** or **autovacuum** is enabled. • The **autoanalyze_cache_num** parameter is added to set the maximum number of tables whose statistics can be cached by the lightweight **autoanalyze**. • The comparison between common **autoanalyze** and lightweight **autoanalyze** is added. • The **PV_RUNTIME_RELSTATS** view is added to display table-level statistics generated by **autoanalyze** in the memory. • The **PV_RUNTIME_ATTSTATS** view is added to display column-level statistics generated by **autoanalyze** in the memory. • The **pg_stat_get_runtime_relstats** function is added to return table-level statistics generated by lightweight **autoanalyze** in the memory. • The **pg_stat_get_runtime_attstats** function is added to return	• Developer Guide > GUC Parameters > Automatic Cleanup • Developer Guide > Data Migration > Other Operations > Analyzing a Table • Developer Guide > System Catalogs and System Views > System Views > PV_RUNTIME_RELSTATS • Developer Guide > System Catalogs and System Views > System Views > PV_RUNTIME_ATTSTATS • SQL Syntax Reference > Functions and Operators > Statistics Information Functions

Feature	Description	Reference
	column-level statistics generated by lightweight **autoanalyze** in the memory. • The **pg_stat_set_last_data_changed_num** function is added to set the number of historical modifications on the current node in the table.	
The **on** condition in the **MERGE INTO** statement supports system columns.	Added the syntax for modifying the status flag of an index.	SQL Syntax Reference > DDL Syntax > ALTER INDEX
Sequence pushdown	Added the **volatile_shipping_version** parameter to control the pushdown scope of the **volatile** functions.	Developer Guide > GUC Parameters > Query Planning> Optimizer Method Configuration

Feature	Description	Reference
Volatile temporary tables are supported.	• The **get_volatile_pg_class** function is added to obtain the basic information about **pg_class** corresponding to the volatile temporary table, including table name parameters and the table list. • The **get_volatile_pg_attribute** function is added to obtain the basic information about **pg_attribute** corresponding to the volatile temporary table, including column parameters and the column list. • The **VOLATILE** keyword is added to **CREATE TABLE** and **CREATE TABLE AS** in the DDL syntax; and to **SELECT INTO** in the DQL syntax. • The **DISCARD** syntax is added to release internal resources related to database sessions in the current session. • **VACUUM FULL** does not support operations on volatile temporary tables. • The **max_volatile_memory** parameter is added to specify the maximum total memory occupied by contexts related to volatile temporary tables in all sessions.	• SQL Syntax Reference > Functions and Operators > System Administration Functions > Other Functions • SQL Syntax Reference >DDL Syntax > CREATE TABLE • SQL Syntax Reference >DDL Syntax > CREATE TABLE AS • SQL Syntax Reference > DQL Syntax > SELECT INTO • SQL Syntax Reference >DDL Syntax > DISCARD • SQL Syntax Reference > DDL Syntax > VACUUM • Developer Guide > GUC Parameters > Resource Consumption > Memory • Developer Guide > Performance Optimization > SQL Execution Troubleshooting > Automatic Retry upon SQL Statement Errors

Feature	Description	Reference
	• The **max_volatile_tables** parameter is added to specify the maximum number of volatile temporary tables created for each session. • The constraints on the volatile temporary table are added.	

Feature	Description	Reference
Enhanced MySQL compatibility	• The **TRY_CAST** keyword is added. • The CONV function is added to convert the given value or string into the value of a specific number system, and to output the result as a string. • The HEX function is added to return a specified value or a hexadecimal string. • The UNHEX function is added to perform the reverse operation of HEX(n). • The SPACE function is added to return a string consisting of a specified number of spaces. • The STRCMP function is added to compare the sizes of two strings. • The BIN function is added to convert the bigint type from decimal to binary, and to return the result as a string. • The **substring_index** function is added to perform a case-sensitive search for delimiters and return the substring that appears before the delimiter for the specified **count**. • The **rand** function is added to return a random number in the range 0.0 to 1.0. • The **truncate** function is added to truncate a	• SQL Syntax Reference > Keyword • SQL Syntax Reference > Functions and Operators> Character Processing Functions and Operators • SQL Syntax Reference > Functions and Operators > Binary String Functions and Operators • SQL Syntax Reference >Functions and Operators > Mathematical Functions and Operators • SQL Syntax Reference >Functions and Operators > Date and Time Processing Functions and Operators • SQL Syntax Reference > Functions and Operators > Type Conversion Functions • SQL Syntax Reference > Functions and Operators> UUID Functions • SQL Syntax Reference > Functions and Operators > Array Functions and Operators • SQL Syntax Reference >DDL Syntax > CREATE TABLE • SQL Syntax Reference >DDL Syntax > ALTER TABLE • Developer Guide > System Catalogs and System Views > System Catalogs > PG_ATTRDEF

Feature	Description	Reference
	number to specified decimal places. • The **addtime** function is added to return the result of a given datetime plus a time interval of a specified unit. • The **subtime** function is added to return the result of a given datetime subtracted by a time interval of a specified unit. • The **timediff** function is added to subtract one date parameter from another. • The **curdate** function is added to return the current date. • The **curtime** function is added to return the current time. • The **convert_tz** function is added to convert the given date and time to those in the specified time zone. • The **try_cast** function is added to convert a parameter to the value of a specified type. • The **cast(x, y)** function is added to convert data types. • The **uuid** function is added to generate a sequence number of the UUID type. • The interval function is added to search for the last array index that is less than or equal to the target	• Developer Guide > Syntax Compatibility Differences Among Oracle, Teradata, and MySQL

Feature	Description	Reference
	parameter **n** from the input integer array. • The **split** function is added to separate strings by delimiter and return an array. • The **ON UPDATE on_update_expr** syntax is added to the **CREATE TABLE** syntax. • The **MODIFY [COLUMN] ON UPDATE on_update_expr** syntax is added to the **ALTER TABLE** syntax. • The **adbin_on_update** and **adsrc_on_update** columns are added to the **PG_ATTRDEF** system catalog. • Improved MySQL compatibility with the **CURDATE**, **CURTIME(p)**, and backquote syntaxes. • In the **ALTER FUNCTION**, **ALTER INDEX**, **ALTER TABLE** and **ALTER VIEW** syntaxes, the new name can be prefixed with the schema name of the original table. The schema name cannot be changed at the same time.	

Feature	Description	Reference
Value-based redistribution	• New GUC hint parameters are added. The **enable_stream_ctescan** parameter cannot be set at the subquery level, and the **enable_value_redistribute** parameter can be set at the subquery level. • The **enable_value_redistribute** parameter is added to specify whether to generate value-based redistribution plans.	• Developer Guide > Query Performance Optimization > Query Improvement > Hint-based Tuning > Configuration Parameter Hints • Developer Guide > GUC Parameters > Query Planning> Optimizer Method Configuration
Magic set enhancement	• The **eager_magicset** value is added to the **rewrite_rule** parameter. You can use **eager_magicset** to query rewriting rules (to push conditions from the main query to subqueries). • The **windowagg_pushdown_enhancement** parameter is added to specify whether to enable enhanced predicate pushdown for the window functions in aggregation scenarios.	• Developer Guide > GUC Parameters > Developer Options • Developer Guide > GUC Parameters > Query Planning > Other Optimizer Options
SetOp supports two-level computing.	The **setop_optmode** parameter is added to control whether to perform deduplication on the query branch statements of a set operation (**UNION**/ **EXCEPT**/**INTERSECT**) without the **ALL** option.	Developer Guide > GUC Parameters > Query Planning> Optimizer Method Configuration

Feature	Description	Reference
The performance of OBS foreign table import is optimized.	The **file_split_threshold** parameter is added, which is valid only for read only foreign tables in TEXT format. It is applicable to the import scenario where the number of files is smaller than the number of DNs.	SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (for OBS Import and Export)
GaussDB(DWS) can connect to OBS through an agency.	The **security_token** parameter is added to form a temporary security credential together with the temporary AK and SK.	• SQL Syntax Reference > DDL Syntax > CREATE SERVER • SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (for OBS Import and Export)
The AK and SK can be specified using **server** when OBS foreign tables are exported in text format.	• A description is added for the **fdw_name** parameter of **FOREIGN DATA WRAPPER**. GDS foreign table uses **gsmpp_server** and **fdw_name** is **dist_fdw**. • A constraint is added for the **access_key**, **secret_access_key**, and **security_token** parameters. **FOREIGN DATA WRAPPER** is set to **dist_fdw**. • The **gsmpp_server** parameter is changed to **server_name**. You can use the **gsmpp_server** created by the initial database by default, or use a custom server.	• SQL Syntax Reference > DDL Syntax > CREATE SERVER • SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (for OBS Import and Export)
OBS read and write performance monitoring	The **GS_OBS_LATENCY** view is added to record the average latency of OBS.	Developer Guide > System Catalogs and Views > System Views > GS_OBS_LATENCY

Feature	Description	Reference
OBS read/write bandwidth views	The read/write bandwidth views **GS_OBS_READ_TRAFFIC** and **GS_OBS_WRITE_TRAFFIC** are added. Data is aggregated at an interval of 10 minutes. The **logtime** column is added to display time groups.	Developer Guide > System Catalogs and System Views > System Views
Column storage supports ring buffer.	The GUC parameter **enable_cstore_ring_buffer** is added to control the column-store ring buffer.	Developer Guide > GUC Parameters > Resource Consumption > Memory > enable_cstore_ring_buffer
Adaptation to GaussDB(DWS) hstore column-store vacuum	System functions are added for column-store vacuum. get_col_file_info(table_name) get_all_col_file_info() col_rebuild_file_relation(table_name)	SQL Syntax Reference > Functions and Operators > System Administration Functions > Other Functions
Compressed text can be imported to OBS foreign tables **dfs_fdw**.	The **compression** parameter is added to specify the file compression mode for import and export.	SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (SQL on OBS or Hadoop)

Feature	Description	Reference
The import with indexes is optimized to avoid generating too many Xlogs.	• The **PGXC_STAT_WAL** view is added to show the WAL logs and data page traffic information of the current query. • The GUC parameter **enable_wal_decelerate** is added to set the WAL log rate limit. • The GUC parameter **wal_decelerate_policy** is added to control the action after rate limiting is triggered. • The GUC parameter **wal_write_speed** is added to specify the maximum WAL write speed allowed by each query on a single DN per second. • The GUC **parameter wal_decelerate_trigger_threshold** is added to specify the threshold for a query to trigger WAL write rate limiting on a single DN. • Two functions are dded to the SQL syntax. • The **pg_stat_wal_write()** function is added to record the thread information of the current instance and collect statistics on the import volume and rate of WAL logs and data pages. • The **pgxc_stat_wal_write()** function is added to record information about threads	• Developer Guide > System Catalogs and Views > System Views > PGXC_STAT_WAL • Developer Guide > GUC Parameters > Write Ahead Logs > Settings • SQL Syntax Reference > Functions and Operators > Statistics Information Functions

Feature	Description	Reference
	interacting with the CN on each DN and collect statistics on the import volume and rate of WAL logs and data pages.	
The **dist fdw** foreign table in the **LATIN1** database supports the **dataencoding** parameter.	The **dataencoding** parameter is added to convert between **dataencoding** and **encoding** during GDS import and export.	• SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (for GDS Import and Export) • SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (SQL on other GaussDB(DWS)
Bandwidth compression for GDS communication	The **gds_compress** is added to reduce the bandwidth usage for interconnection between clusters when users use GDS pipe files.	• SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (for GDS Import and Export) • SQL Syntax Reference > DDL Syntax > CREATE FOREIGN TABLE (SQL on other GaussDB(DWS)
DDL lock timeout configuration	The GUC parameter **ddl_lock_timeout** is added to specify the DDL lock timeout interval.	Developer Guide > GUC Parameters > Lock Management > ddl_lock_timeout
PG_LOCKS view enhancement	• Added the views **PGXC_WAIT_DETAIL** and **PGXC_LOCKWAIT_DETAIL**. • Columns including **wait_on_pid** and **query_id** are added.	Developer Guide > System Catalogs and System Views > System Views > PGXC_WAIT_DETAIL Developer Guide > System Catalogs and System Views > System Views > PGXC_LOCKWAIT_DETAIL

Feature	Description	Reference
The original pushdown plan is modified.	The **implied_quality_optmode** parameter is added to specify how to pass conditions for the equivalent columns in a statement.	Developer Guide > GUC Parameters > Query Planning > Other Optimizer Options > implied_quality_optmode
ANALYZE sampling adaptation (added in 8.2.0.100)	• Added GUC parameters **analyze_stats_mode** and **analyze_sample_mode**. • The default value of **random_function_version** is changed to **1**.	• Developer Guide > GUC Parameters > Automatic Cleanup • Developer Guide > GUC Parameters > Query Planning > Other Optimizer Options
Cost estimation enhancement (added in 8.2.0.100)	The default value of **enable_extrapolation_stats** is changed to **on**.	Developer Guide > GUC Parameters > Query Planning > Other Optimizer Options
Column-store small CU view (added in 8.2.0.100)	Add the CU information function for column-store tables. • get_col_cu_info • get_col_file_vacuum_info • get_all_col_cu_info	SQL Syntax Reference > Functions and Operators > System Administration Functions > Other Functions
Optimized dirty page monitoring view (added in 8.2.0.100)	Added view: **PGXC_STAT_TABLE_DIRTY** Added functions: **pg_stat_get_tuple()** and **pgxc_stat_single_table**	• Developer Guide > System Catalogs and System Views > System Views > PGXC_STAT_TABLE_DIRTY • "Functions and Operators > Statistics Information Functions" in the *SQL Syntax Reference*
SELECT does not block **TRUNCATE** (add in 8.2.0.100)	Added the GUC parameter **ddl_select_concurrent_mode** to control the concurrency of DDL and **SELECT** statements.	Developer Guide > GUC Parameters > Lock Management

Feature	Description	Reference
Support for HStore tables (added in 8.2.0.100)	Added the **enable_hstore** parameter to the **CREATE TABLE** syntax in the standalone hybrid data warehouse scenario.	• Developer Guide > Hybrid Data Warehouse • SQL Syntax Reference >DDL Syntax > CREATE TABLE
Fine-grained DR (added in 8.2.0.100)	• Added the function of backing up DR system catalogs and system views. • Added the **enable_disaster_cstore** and **fine_disaster_table_role** parameters to the **CREATE TABLE** syntax. • Added the **local coordinator only** parameter to the **LOCK** syntax. • Added the release and subscription SQL syntax.	• Developer Guide > System Catalogs and System Views > System Catalogs >PG_FINE_DR_INFO/ PG_STAT_LAST_OPERATION/ PG_PUBLICATION/ PG_PUBLICATION_REL/ PG_PUBLICATION_NAMESPACE • Developer Guide > System Catalogs and System Views > System Views > PG_PUBLICATION_TABLES • SQL Syntax Reference >DDL Syntax > CREATE TABLE • SQL Syntax > DML Syntax > LOCK • SQL Syntax > DDL Syntax > ALTER PUBLICATION/ CREATE PUBLICATION/DROP PUBLICATION/ALTER SUBSCRIPTION/ CREATE SUBSCRIPTION/DROP SUBSCRIPTION

Feature	Description	Reference
Display none of the non-aggregation function query columns after **GROUP BY** (added in 8.2.0.101)	The **disable_full_group_by_mysql** option is added to the GUC parameter **behavior_compat_options**. Users can set this parameter to display none of the non-aggregation function query columns after **GROUP BY** in a query.	Developer Guide > GUC Parameters > Miscellaneous Parameters > **behavior_compat_options**
Optimization of the returned number of limits on each DN in the **distinct limit** scenario (added in 8.2.0.101)	The **enable_agg_limit_opt** parameter is added to specify whether to optimize **select distinct col from table limit N**.	Developer Guide > GUC Parameters > Query Planning > Other Optimizer Options
The query performance of a single table is improved, and the restriction on **inlist roughcheck** is removed. (Added in 8.2.0.101)	The **inlist_rough_check_threshold** parameter is added to control the maximum number of values in the **IN** condition.	Developer Guide > GUC Parameters > Query Planning > Other Optimizer Options
Optimized hstore hot and cold table features (added in 8.2.0.101)	Hot and cold data can be managed in HStore tables.	Developer Guide > Hybrid Data Warehouse > Introduction to Hybrid Data Warehouse
Optimized DN Parallel Import in the PBE Scenario (added in 8.2.0.102)	Added the GUC parameter **enable_parallel_batch_insert** to control whether to enable concurrent data import to row-store and column-store tables.	Developer Guide > GUC Parameters > Parallel Import

Behavior changes

Table 1-30 Behavior changes

Change Type	No.	Change	Description
Added	1	Added WindowAgg support for value partition execution plans	The window function is executed at the outermost layer of statements. If **ORDER BY** is not used, the data behaviors are in a default order. In version 8.2.0, after **enable_value_redistribute** is enabled by default, the data becomes unordered. You need to use the ORDER BY clause to specify the order.
Modified	2	Changed the exception rule configuration mode from tool configuration to syntax configuration.	In earlier versions, exception rules are configured for Cgroups using gs_cgroup. In 8.2.0, exception rules are configured using the **CREATE**/**ALTER**/**DROP EXCEPT RULE** syntax.
	3	Replaced the exception rule **qualificationtime** with **elapsedtime**.	In earlier versions, **qualificationtime** is used as a rule, but it can only be used together with **cpuskewpercent**. Their effect is the same as that of **elapsedtime**. Therefore, this configuration item can be replaced with **elapsedtime**.
	4	Tid column name	Row storage supports the tid column names, which are not supported before.
	5	max_process_memory	Changed the number of CNs to half of that of DNs.

SQL Syntax

Table 1-31 SQL syntax changes

Change Type	No.	Feature	Change Description
Added	1	Table parameter **analyze_mode**	Added the parameter **analyze_mode**, which specifies the ANALYZE modes supported by a table. If this parameter is not set, the previous mode remains unchanged.
	2	Hints for disabling subqueries	Added the hints for disabling subqueries. Format: **no_merge (***current query***)** and **no_merge(***relid***)** (specified object).
	3	DML hints	Added support for hints in INSERT, UPDATE, MERGE, and DELETE statements.
	4	AGG redistribution hints support specified column names.	Column names can be specified for AGG redistribution hints. The format is **/+ redistribute ((*) (**a b c d**)) */**.
	5	**Count(distinct)** hints	Added hints for specifying the count(distinct) column, effective only in the double Hash Agg scenario.
	6	**invisible**	Added index OPTIONS to specify whether the index scan is visible in the execution plan.
	7	Backquotes	Backquotes can be used to distinguish common characters from special characters.
	8	**ON UPDATE** in column definition	When the ON UPDATE attribute is specified in the column definition, the update timestamp is automatically filled when data is updated.
	9	**CREATE**/**ALTER**/**DROP EXCEPT RULE**	Added the syntax to add, create, and delete kernel exception rules.

Change Type	No.	Feature	Change Description
	10	**OPTIONS security_token** in **CREATE SERVER** statements	Added **OPTIONS security_token** to transfer security tokens for using temporary AKs and SKs to access OBS.
	11	**OPTIONS security_token** in **CREATE FOREIGN TABLE** statements	Added **OPTIONS security_token** to transfer security tokens for using temporary AKs and SKs to access OBS.
	12	**file_split_ threshold** in **(dist_fdw) option** in OBS foreign tables	Added the **file_split_ threshold** to the OBS foreign table parameter **(dist_fdw) option**.
	13	**gds_compress** in GDS foreign tables	Added the **gds_compress** parameter, which is used for compressed transmission of data between DNs and GDSs during GDS foreign table interconnection. Currently, only the Snappy compression algorithm is supported.
	14	**dataencoding** in GDS foreign tables	Added the **dataencoding** parameter, which specifies the character set of the data stored in the **latin1** database.
	15	**gds_compress** in **gc_fdw** foreign tables (**sql on other gaussdb** foreign tables)	Added the **gds_compress** parameter, which is used for compressed transmission of data between DNs and GDSs during GDS foreign table interconnection. Currently, only the Snappy compression algorithm is supported.
	16	**dataencoding** in **gc_fdw** foreign tables (**sql on other gaussdb**).	Added the **dataencoding** parameter, which specifies the character set of the data stored in the **latin1** database.

Change Type	No.	Feature	Change Description
Modified	17	**CREATE/ALTER REDACTION POLICY**	The default value of **deltarow_threshold** is set to **10000**, which specifies the upper limit of rows when column-store tables are imported to the Delta table. This parameter is valid only when the table-level parameter **enable_delta** is enabled.
	18	Syntax of association between resource pools exception rules	Added the **INHERIT** option, which specifies whether a policy is inherited from other policies.
	19	**single_node** support for OBS foreign table syntax	In earlier versions, memory exception rules can be configured for resource pools. The exception rules in 8.2.0 are configured via the unified interface: **CREATE/ALTER RESOURCE POOL 'pool' with (EXCEPT_RULE='rule1,2');**
	20	Use **server** to specify the AK and SK of an OBS foreign table (**dist_fdw**).	**Single_node** supports OBS foreign tables (**dfs_fdw**).

Keywords

Table 1-32 Keywords

Change Type	No.	Keyword	Change Description
Added	1	TRY_CAST	Added the non-reserved keyword **TRY_CAST** for the compatibility with the **TRY_CAST** function in MySQL. If the type conversion fails, NULL is returned.

System Catalogs

Table 1-33 System catalogs

Change Type	No.	Name	Change Description
Added	1	gs_blocklist_query	Added the system catalog **gs_blocklist_query**, which is used to store blocklist statements and exception rule triggering information.
	2	pg_except_rule	Added the system catalog **pg_except_rule**, which is used to store information about exception rules created in the cluster, including the rule name, rule threshold, and action triggered by the rule.
Modified	3	pg_redaction_policy, pg_redaction_column	Added the **inherited** column, which specifies whether a policy is manually created or inherited.

System Functions

Table 1-34 System functions

Change Type	No.	Function	Description
Added	1	pg_stat_get_runtime_relstats	Queries table-level statistics generated by AUTOANALYZE.
	2	pg_stat_get_runtime_attstats	Queries column-level statistics generated by AUTOANALYZE.
	3	pg_stat_set_last_data_changed_num	Interface for manually setting the total size of historical modification for the remedy of the **pgstat** content loss.

Change Type	No.	Function	Description
	4	strcmp()	Compares two strings. If the first string equals the second string, 0 is returned. If the first string is less than the second string, -1 is returned. If the second string is less than the first string, 1 is returned. If the input parameters contain null, null is returned.
	5	hex()	Converts each character or decimal integer in the input parameter into a hexadecimal number.
	6	unhex()	Performs the reverse operation of **hex()**. This function interprets each pair of hexadecimal digits (in the argument) as a number and converts it into a character. The result character is returned in bytea format.
	7	space(n)	Returns a string consisting of **n** spaces.
	8	rand()	Obtains a random number ranging from 0.0 to 1.0.
	9	truncate(x,d)	Truncates **x** to **d** decimal places. If **d** is a negative number, then the number is truncated to the left side of the decimal point.
	10	addtime(expr1,expr2)	Adds time. expr1 is of the time or timestamp type, and expr2 is of the interval type.
	11	subtime(expr1,expr2)	Subtracts a time period **expr2** from the time **expr1**. **expr1** is of the time or timestamp type. **expr2** is of the interval type. The return type is the same as that of **expr1**.

Change Type	No.	Function	Description
	12	timediff(expr1,expr2)	Calculates the time difference between two **expr1** and **expr2** (expr1 - expr2). Returns the time type. The arguments must be both of the time or timestamp type.
	13	curdate()	Returns the current date (date type).
	14	curtime([fsp])	Returns the current time (time type). **fsp** indicates the precision of the returned time.
	15	uuid()	Returns a universally unique identifier (UUID).
	16	convert_tz()	Convert the time zone.
	17	cast(expr,typename)	Converts **expr** to a specified type.
	18	digest()	Generates binary hash values of the given data based on different algorithms.
	19	hmac()	Calculates the MAC value for the data with the key.
	20	crypt () and gen_salt ()	The **crypt()** and **gen_salt()** functions are used for password hashing. **crypt()** executes hashes to encrypt data, and **gen_salt()** generates salted hashes.
	21	pgp_sym_encrypt()	Used for symmetric key encryption.
	22	pgp_sym_decrypt()	Decrypts a message encrypted using a PGP symmetric key.
	23	pgp_pub_encrypt()	Used for public key encryption.
	24	pgp_pub_decrypt()	Decrypts a message encrypted using a PGP public key.
	25	pgp_key_id()	Extracts the key ID of the PGP public or private key. If an encrypted message is used as the input, the ID of the key used to encrypt the message will be returned.

Change Type	No.	Function	Description
	26	armor()	Converts binary data into PGP ASCII-armor format by the CRC calculation and formatting of a Base64 string.
	27	dearmor()	Performs the conversion opposite to the **armor()** function.
	28	pgp_armor_headers()	Returns the armor header in data.
	29	encrypt	Original encryption function, which does not support any advanced functions of PGP encryption. The IV is 0.
	30	decrypt	Original decryption function, which does not support any advanced functions of PGP encryption. The IV is 0.
	31	encrypt_iv	Original decryption function, which does not support any advanced functions of PGP encryption. The IV can be set by users.
	32	decrypt_iv	Original decryption function, which does not support any advanced functions of PGP encryption. The IV can be set by users.
	33	gen_random_bytes	Generates cryptographically strong random bytes.
	34	gen_random_uuid()	Returns a random UUID of version 4.
	35	get_volatile_pg_class	Obtains the **pg_class** metadata related to all volatile temporary tables in memory (metadata in volatile temporary tables is not stored in system catalogs).
	36	get_volatile_pg_attribute	Obtains the pg_attribute metadata related to all volatile temporary tables in memory.

Change Type	No.	Function	Description
	37	gs_increase_except_num	Increases the number of query exceptions.
	38	gs_update_blocklist_hash_info	Updates the blocklist information in memory.
	39	gs_append_blocklist	Adds blocklist statements.
	40	gs_remove_blocklist	Removes statements from a blocklist.
	41	gs_wlm_rebuild_except_rule_hash	Triggers the building of an exception rule kernel hash table.
	42	pg_stat_wal_write	Queries the thread information on the current CN or DN, and the import volume and rate of WAL and data pages.
	43	pgxc_stat_wal_write	Queries the thread information about the interaction between CNs and DNs, and the import volume and rate of WALs and data pages.
	44	get_col_file_info	Obtains the number of empty CU files and the total number of CU files of a specified column-store table.
	45	get_all_col_file_info	Obtains the number of empty CU files and the total number of CU files of all column-store tables.
	46	col_rebuild_file_relation	Replaces the Relfilenode and reorganize the CU files of the specified column-store table. After the reorganization, the CU files become consecutive 1 GB files.
Modified	47	pg_query_audit/ pgxc_query_audit	Changed the **thread_id** field to the **session id** field to record the session where the statement is executed.
	48	login_audit_messages/ login_audit_messages_pid	Added the session id field to record the session where the statement is executed.

System Views

Table 1-35 System views

Change Type	No.	View	Change Description
Added	1	gs_obs_latency	Monitors OBS latency.
	2	pgxc_stat_wal	Queries the traffic information about WAL logs and data pages of the current query.
	3	pgxc_lockwait_detail	Displays detailed information about the lock wait chain on each node.
	4	pgxc_wait_detail	Displays details about the SQL waiting chains of all nodes, including the wait nodes and queries.
Modified	5	gs_obs_read_traffic	Data aggregation interval is changed to 10 minutes, and the **logtime** column is added to display time groups.
	6	gs_obs_write_traffic	Data aggregation interval is changed to 10 minutes, and the **logtime** column is added to display time groups.

GUC Parameters

Table 1-36 GUC parameters

Change Type	No.	Parameter	Change Description
Added	1	enable_wlm_internal_memory_limit	Controls whether to enable the internal restriction of the WLM. For example, the estimated memory of a job cannot exceed 80%, 90%, or 40%.
	2	enable_strict_memory_expansion	Controls whether job memory expansion is performed on DNs.

Change Type	No.	Parameter	Change Description
	3	allow_zero_estimate_memory	Controls whether the estimated memory of a job can be set to 0 when there is no statistics for the job.
	4	max_process_memory_auto_adjust	Controls whether to dynamically adjust **max_process_memory** during a primary/standby CM switchover.
	5	wlm_memory_feedback_adjust	Specifies whether to enable the memory negative feedback function in dynamic load management. The value can be **on**, **off**, or **on(50,40)**.
	6	autoanalyze_mode	A system-level AUTOANALYZE control parameter. By default, the **normal** mode is the same as before. If the mode is set to **light**, dynamic sampling is enabled.
	7	enable_redactcol_computable	This parameter specifies whether to enable the computable data redaction. By default, this function is disabled.
	8	windowagg_pushdown_enhancement	Controls whether to push down the intersection of **PARTITION BY** and **GROUP BY** when the window function and **GROUP BY** appear at the same time.
	9	enable_from_collapse_hint	Specifies whether the plans with hints take precedence in execution. By default, the parameter is enabled and statement-level hints are supported.
	10	enable_value_redistribute	Specifies whether to use value partition streams to optimize WindowAgg.

Change Type	No.	Parameter	Change Description
	11	setop_optmode	Specifies whether to perform deduplication on subqueries of the setop operation. The default value is **disable**, indicating that deduplication is not performed, which is the same as that in versions earlier than 8.2.0. The value **force** indicates that deduplication is forcibly performed. The value **cost** indicates that deduplication is performed based on the cost. (Note: Deduplication is performed only when the bottom layer of the setop branch is the stream operator.)
	12	security_enable_options	Indicates that some operations are enabled in security mode. By default, this parameter is left blank.
	13	audit_object_name_format	Specifies the displaying mode of objects in the **object_name** column in audit logs. The default value is **single**, indicating that a single object is displayed.
	14	enable_redactcol_equal_const	Restricts the constant obtaining behaviors in the new version. By default, this function is disabled, and equivalent comparison between anonymized columns and constants is prohibited.
	15	enable_mixedagg	Indicates whether to generate a mixed aggregate execution plan. This parameter is disabled by default.
	16	max_streams_per_query	Controls whether to report an error in the optimizer phase for a plan with a large number of streams to prevent streams from occupying too many connections. The default value is **10000**.

Change Type	No.	Parameter	Change Description
	17	agg_max_mem	Controls the maximum estimated memory when the number of aggregation columns exceeds **5**. This prevents CCN queuing caused by high estimated memory.
	18	max_volatile_memory	Maximum number of memory contexts related to the volatile temporary table. The unit is **KB**. The value ranges from **1024** to **INT_MAX**. The default value is **1 GB**.
	19	max_volatile_tables	Maximum number of volatile temporary tables (including auxiliary tables such as TOAST and CUDESC) in a single session. The default value is **300**.
	20	enable_hashfilter	Determines whether to generate hash filters.
	21	volatile_shipping_version	Controls the sequence pushdown enhancing in 8.2.0. The default value is **1**.
	22	smp_thread_cost	Control the generated query_dop value. The default value is **1000**.
	23	enable_wal_decelerate	Specifies whether to limit the flow control rate when data is imported with indexes.
	24	wal_decelerate_policy	Specifies the behavior policy after rate limiting is triggered.
	25	wal_write_speed	Indicates the maximum WAL write speed (byte/s) of each query on a single DN per second.
	26	wal_decelerate_trigger_threshold	Specifies the WAL write rate limit triggered by each query on a single DN.
	27	enable_cstore_ring_buffer	Controls whether to enable the column-store ring buffer.

Change Type	No.	Parameter	Change Description
	28	enable_tsdb_multi_temperature	Specifies whether to enable the OBS cold and hot storage for time series tables. By default, the OBS cold and hot storage is disabled.
	29	enable_tsdb_multi_temperature	Specifies whether to enable the OBS cold and hot storage for time series tables. By default, the OBS cold and hot storage is disabled.
	30	ddl_lock_timeout	Specifies the lockwait time for DDL statements. If the lockwait times out, DDL statements are terminated. This parameter is valid only for level-8 locks. The default value is 0, indicating that this parameter does not take effect.
	31	build_backup_param	Specifies the minimum specifications for disk backup during incremental build.
Modified	32	comm_tcp_mode	The parameter type is changed from POSTMASTER to SIGHUP.
	33	max_process_memory	Indicates the maximum memory usage of a process. The type of the parameter is changed from POSTMASTER to SIGHUP. The minimum threshold is 2 GB. You are not advised to set it to the minimum threshold.
	34	rewrite_rule	The enumerated value **eager_magicset** is added to control the magic set enhancement.

Change Type	No.	Parameter	Change Description
	35	behavior_compat_options	The **normal_session_id** parameter is added to **behavior_compat_options** to determine whether to concatenate the session ID with **node_name**. By default, **normal_session_id** is disabled, and the session ID is concatenated with **node_name**.
	36	behavior_compat_options	Added **behavior_compat_options** to **merge_into_with_trigger**. After the parameter is enabled, **MERGE INTO** can be executed on tables with triggers. This parameter is disabled by default.
	37	behavior_compat_options	Added the option **disable_jsonb_exact_match** to control the operator matching. By default, this option is disabled. When this option is disabled, only exact match of JSONB types is supported. If exact match fails, the non-JSONB types are matched. When this option is enabled, and the operator matching does not distinguish types.
	38	hashjoin_spill_strategy	Policies 5 and 6 are added. The behavior is the same as that of policies 0 and 1. The original policies 0 and 1 are modified. If the internal table is too large to fit in memory and the external table is small, keep dividing the internal table until it can no longer be divided. If the issue persists, try swapping the internal and external tables.

Change Type	No.	Parameter	Change Description
	39	sql_use_spacelimit	Modified the initial value of **sql_use_spacelimit** to 10% (upper limit) of the disk space where the current instance resides. If the value is less than 10 GB (lower limit), set the value to 10 GB.
	40	temp_file_limit	Modified the initial value of **temp_file_limit** to 10% (upper limit) of the disk space where the current instance resides. If the value is less than 10 GB (lower limit), set the value to 10 GB.
Deleted	41	enable_grant_public	Indicates that the **grant to public** syntax is not supported in security mode.
	42	enable_grant_option	Indicates that the **grant with grant option** syntax is not supported in security mode.

1.7 Version 8.1.3

1.7.1 New Features in 8.1.3

For details about new features in 8.1.3.*x*, see **New features**. For details about resolved issues in each patch, see **New Features and Resolved Issues in 8.1.3.x**.

[Release date of V8.1.3.336]: October 15, 2024

[Release date of V8.1.3.333]: June 30, 2024

[Release date of V8.1.3.330]: March 16, 2024

[Release date of V8.1.3.325]: November 2, 2023

[Release date of V8.1.3.323]: October 16, 2023

[Release date of V8.1.3.322]: August 4, 2023

[Release date of V8.1.3.321]: June 26, 2023

[Release date of V8.1.3.320]: May 19, 2023

[Release date of V8.1.3.310]: March 9, 2023

[Release date of V8.1.3.300]: December 19, 2022

[Release date of V8.1.3.200]: October 31, 2022

[Release date of V8.1.3.110]: August 23, 2022

[Release date of V8.1.3.100]: June 20, 2022

[Release date of V8.1.3]: April 15, 2022

New features

Table 1-37 New functions in version 8.1.3

Feature	Description	Reference
Partition addition and partition table management are supported.	PERIOD (interval type) and TTL (interval type) are now available in partition management.	**CREATE TABLE** **ALTER TABLE** **CREATE TABLE PARTITION**
Support for foreign tables in JSON format	1. READ ONLY HDFS foreign tables in JSON format are supported. 2. READ ONLY OBS foreign tables in JSON file format are supported.	**CREATE FOREIGN TABLE (SQL on Hadoop or OBS)**

Feature	Description	Reference
Cross-Cluster interconnection	1. Constraints are removed: a. The hang constraint of GDS is removed. b. The constraints on common aggregation functions, such as **sum**, **count**, **min**, **max**, and **avg** are removed. c. The constraints on scalar functions and analytical functions are removed. d. The strong verification of the column names and column types of foreign tables and source tables is added. e. The constraint that the column name cannot contain **WHERE** is removed. f. The limit constraint is removed. g. The temporary GDS foreign table policy is improved. 2. If the local cluster is the destination cluster, you can run the following statements:	**GDS-based Cross-Cluster Interconnection**
Enhanced cold and hot table functions	Automatic migration of cold and hot tables	**CREATE TABLE**
The row-store ring buffer is controlled by a switch.	The ring buffer is controlled by a switch and its threshold is adjustable. Batch update of large tables no longer uses the ring buffer.	**buffer_ring_ratio**

Feature	Description	Reference
GDS supports the euro sign.	The GBK character set supports the euro sign.	**CREATE DATABASE**
OBS foreign tables support XSKY S3.	OBS foreign tables are adapted to XSKY and the parameter **eol** is added.	**CREATE FOREIGN TABLE (SQL on Hadoop or OBS)**
SQL supports Roaring Bitmaps.	The RoaringBitmap function is supported to better support common customer profile services of Internet services.	**Bitmap Functions and Operators** **RoaringBitmap**
Case-insensitive is supported to improve Teradata compatibility.	When migrating data from Teradata to GaussDB(DWS), there is a lot of duplicate data in the source data with different letter cases. If GaussDB(DWS) removes duplicates using the **distinct** syntax, there will still be a significant amount of duplicate data remaining. Using the **upper** function to deduplicate the data will change the source data and require complex procedures and logic. To make things easier and reduce workload, GaussDB(DWS) now has a case-insensitive feature.	**Collation Support** **Collation Version Function**

Feature	Description	Reference
List partitioning	List partitioned tables are supported.	• Syntax: **CREATE TABLE PARTITION** **ALTER TABLE PARTITION** • System catalog: **PG_PARTITION** • Views: **DBA_PART_INDEXES** **DBA_PART_TABLES** **DBA_TAB_PARTITIONS** **USER_IND_PARTITIONS** **USER_PART_INDEXES** **USER_TAB_PARTITIONS**
Enhanced MySQL compatibility	**ALTER TABLE** can modify multiple columns at a time. **ALTER TABLE** supports **COMMENT**. **ALTER TABLE** supports **CHANGE**, **DROP KEY**, **DROP PARTITION**, and **FIRST**/**AFTER**.	**ALTER INDEX** and **CREATE INDEX** **ALTER TABLE** and **CREATE TABLE** **ALTER TABLE PARTITION**
Projection pushdown	Unnecessary columns in a multi-column view are automatically deleted during reference to avoid redundant calculation.	Added the description for **projection pushdown** to **rewrite_rule**
Share scan	Share scan	Stream plan CTEs support share scan. Added **enable_stream_ctescan**. The value **STREAM_CTESCAN** is added to **logging_module**. The optional **[NOT] MATERIALIZED** is added to the **WITH** clause in the **SELECT** syntax.

Feature	Description	Reference
Bloom Filter for column-store tables	In a HASH JOIN, if the thread of the foreign table contains HDFS tables or column-store tables, the Bloom filter is triggered.	**enable_bloom_filter**
Column-store replication tables support **UPDATE**.	Replicated tables support **UPDATE**, **DELETE**, and **MERGE INTO** operations.	**MERGE INTO** **UPDATE** **UPSERT**
Fine-grained permission management	1. Added the **ALTER**, **DROP** and **VACUUM** permissions at table level. 2. Added the **ALTER** and **DROP** permissions at schema level. 3. Add the preset roles **role_signal_backend** and **role_read_all_stats**.	• **Preset Roles** • Added table-level permissions: **VACUUM** **ALTER TABLE** **DROP TABLE** • Added schema-level permissions: **ALTER SCHEMA** **DROP SCHEMA** **ALTER DEFAULT PRIVILEGES** • **GRANT** • **REVOKE**
The problem of the **INSERT OVERWRITE** deadlock on multiple CNs is fixed.	No deadlock occurs when **INSERT OVERWRITE** is triggered on multiple CNs at the same time.	-

Feature	Description	Reference
Execution plan optimization	• If the join conditions contain **OR** and the execution plan uses nestloop, the performance is poor. Therefore, the statement is changed to **UNION ALL**. • To join column-store tables, if the join or filter conditions contain **OR**, the **OR** conditions are pruned before join. • When large and small tables are joined across many DNs, the small table is broadcast instead of being redistributed.	**Tuning Optimizer Parameters** Added **projection_pushdown** and **or_conversion** to **rewrite_rule**.
Relfile-based space statistics and space control	1. The query results of the **pg_table_size**, **pg_relation_size** and **table_skewness** views can be returned in seconds. 2. Schema space control can collect statistics on the full space and the constraints are removed. 3. The user/schema space calibration functions can be quickly executed, and the schema space calibration function can be automatically executed. 4. The SQL interface allocates space quota based on the total schema space instead of the DN level.	• **Workspace Management** • Added GUC parameters for load management: **space_once_adjust_num** **space_readjust_schedule** • Added the system catalog: **PG_RELFILENODE_SIZE** • Added functions: **gs_table_distribution(schemaname text, tablename text)** **gs_table_distribution()** **pgxc_wlm_readjust_relfilenode_size_table()** **pgxc_wlm_readjust_relfilenode_size_table(integer)**

Feature	Description	Reference
Enhanced performance fault locating and demarcation	1. Enhanced communication exception monitoring logs 2. Enhanced **pgxc_stat_activity** view 3. Top SQL statements can be associated with unique SQL statements. 4. Added the distributed view **pg_session_wlmstat**.	• Added the system view **PG_COMM_QUERY_SPEED**. • Added the description for the **lwtid and stmt_type** column of **PG_STAT_ACTIVITY**. • Added the **unique_sql_id** column information to **GS_WLM_SESSION_HISTORY**. • Added the system view **PGXC_SESSION_WLMSTAT**.
Resource load management optimization and reconstruction	1. Enhanced user resource monitoring: The user resource monitoring logic is modified. 2. Enhanced resource monitoring: Job and user resource monitoring is optimized, and queue-level resource monitoring is added.	• **User Resource Monitoring** • Enhanced resource monitoring: – Added: **Resource Pool Monitoring**. – Added system views: **PGXC_RESPOOL_RESOURCE_INFO** **PGXC_RESPOOL_RUNTIME_INFO** **PGXC_RESPOOL_RESOURCE_HISTORY** **GS_RESPOOL_RESOURCE_INFO** **GS_RESPOOL_RUNTIME_INFO** – Added the system catalog: **GS_RESPOOL_RESOURCE_HISTORY**

Feature	Description	Reference
Modified partition table features	1. Changed the limit on the number of boundary values: The number of partitions in a partition table cannot exceed 32767, and the number of boundary values of all partitions cannot exceed 32767. 2. The default partition creation policy is modified. When a partition table is created, two default partitions are created. The partition time range of the two default partitions is **PERIOD**.	**CREATE TABLE PARTITION**
Redistribution column hints are supported in the AGG process.	During data migration from Oracle data warehouses, you can manually specify the distribution column using enhanced hints.	**Plan Hint Optimization** **Stream Operation Hints**
DDL lock timeout configuration (added in 8.1.3.110)	To specify whether to report an error when the divisor is 0 in MySQL compatibility mode, the compatibility configuration item **enable_division_by_zero_mysql** is added to the **behavior_compat_options** parameter.	**behavior_compat_options**
DDL lock timeout configuration (added in 8.1.3.200)	Added the GUC parameter **ddl_lock_timeout**, which specifies the DDL lock timeout interval.	**ddl_lock_timeout**

Feature	Description	Reference
Enhanced the view **PG_LOCKS**. (added in 8.1.3.200)	• Added the views **PGXC_WAIT_DETAIL** and **PGXC_LOCKWAIT_DETAIL**. • Add columns such as **wait_on_pid** and **query_id**.	• **PGXC_WAIT_DETAIL** • **PGXC_LOCKWAIT_DETAIL**
Data masking compatibility configuration (added in 8.1.3.310)	The GUC parameter **redact_compat_options** is added to configure the compatibility of the data masking techniques.	• **redact_compat_options**

Feature	Description	Reference
Concurrent truncate, exchange, and select operations (added in 8.1.3.320)	Added the GUC parameter **ddl_select_concurrent_mode**. This parameter is used to solve the problem that the query statement lasts for a long time and blocks DDL in the scenario where the data volume is large or the query is complex. The effect is the same as that of the Oracle database. Application scenarios: • Concurrent truncate and select operations • Concurrent exchange and select operations Scenario constraints: • Concurrency is not supported when there are conflicts with locks of higher levels (higher than level 1). For example, **autoanalyze** is triggered by **select** when **autoanalyze_mode** is set to **normal**. • Concurrency is not supported when there are conflicts with locks in transaction blocks. Connection constraints: • When the JDBC, ODBC, or Python driver is used, if **autocommit** is set to **false**, this feature is not supported. • This feature is not supported by query statements delivered by Data Studio and	**ddl_select_concurrent_mode**

Feature	Description	Reference
	auto-committed transactions.	
Enabling of the early stop optimization for Limit statements (added in 8.1.3.320)	Added the GUC parameter **enable_limit_stop** to specify whether to enable the early stop optimization for LIMIT statements.	**enable_limit_stop**
Early stop behavior change (added in 8.1.3.322)	The default value of the GUC parameter **enable_limit_stop** is changed to **on**.	**enable_limit_stop**
normalize_negative_zero (added in 8.1.3.333)	The **behavior_compat_options** parameter now includes the **normalize_negative_zero** option, which determines whether the **ceil()** and **round()** functions will return **-0** for a given float value.	**behavior_compat_options** ("Developer Guide > GUC Parameters > Miscellaneous Parameters")
internal_compat_options (added in 8.1.3.333)	A new GUC parameter **internal_compat_options** has been added to allow for configuration of the database compatibility behavior. The **light_proxy_permission_compat** option is used to configure the nested query permission in the light proxy scenario.	"GUC Parameters" > "Miscellaneous Parameters" in the *Developer Guide*
disable_client_detection_commit (added in 8.1.3.333)	The **behavior_compat_options** parameter now includes the **disable_client_detection_commit** option, which determines whether the connection to the client is checked before committing each transaction.	**behavior_compat_options** ("Developer Guide > GUC Parameters > Miscellaneous Parameters")

Feature	Description	Reference
enable_stream_ctescan (Disabled by default for new installation) (modified in 8.1.3.333)	Modified the **enable_stream_ctescan** parameter. This parameter is forward-compatible after an upgrade and is disabled by default upon new installation.	"GUC Parameters" > "Optimizer Method Configuration" in *Developer Guide*
enable_trunc_orc_string (added in 8.1.3.336)	Controls the foreign table query behavior when the foreign table field is in ORC format and the data type is varchar(n), but the field type in the ORC file is string and the length of the string exceeds n.	**behavior_compat_options** ("Developer Guide > GUC Parameters > Miscellaneous Parameters")
gds_fill_multi_missing_fields (added in 8.1.3.336)	gds_fill_multi_missing_fields Controls the behavior when the GDS foreign table fault tolerance parameter **fill_missing_fields** is set to **true** or **on**.	**behavior_compat_options** ("Developer Guide > GUC Parameters > Miscellaneous Parameters")

SQL Syntax

Table 1-38 SQL syntax

Change Type	No.	Name	Change Description
Added	1	CREATE FOREIGN TABLE (SQL on Hadoop or OBS)	Added **json** for the parameter **Format** of **OPTION**.
	2	CREATE FOREIGN TABLE (SQL on Hadoop or OBS)	Added **force_mapping** for **OPTION**.
	3	Column type **kvtype**	Specifies the column type of time series tables in the **CREATE TABLE** and **ALTER COLUMN** syntax.
	4	Table-level parameter: **sub_partition_count**	Configures the number of level-2 partitions in a time sequence table.

Change Type	No.	Name	Change Description
	5	Table-level parameter **ttl**	Specifies the expiration time of auto-added partitions.
	6	Table-level parameter **period**	Specifies the interval for triggering partition adding and dropping tasks and the time range of a single partition.
	7	List partitioning support	The list partition operation syntax is added, including the **CREATE** and **ALTER** operations. The **CREATE** syntax is a distinct branch of syntax. In the **AT** syntax, all syntax, except for **SPLIT**, is similar to range partitioning.
	8	ALTER TABLE ADD INDEX/ DROP INDEX	The **ALTER TABLE** syntax branch is added to add and delete indexes.
	9	CREATE TABLE ... LIKE	The **CREATE TABLE... LIKE** syntax without brackets is added.
	10	ALTER TABLE ... ADD/ MODIFY COLUMN	The statements for adding and modifying columns are added to **ALTER TABLE**. **COMMENT** and **DEFAULT** can be set.
	11	Added the **[NOT] MATERIALIZED** syntax after **with cte as**.	Added the **[NOT] MATERIALIZED** syntax after **with cte as**.
	12	alter index ... comment	The **COMMENT** clause is now available in the **ALTER INDEX** syntax.
	13	ALTER TABLE ... add index comment	The **COMMENT** clause is now available in the **alter table add index** syntax.
	14	create index index_name on table_name comment	The **COMMENT** clause is now available in the **CREATE INDEX** syntax.
	15	copy to	Added **server**, **bom**, **fileprefix**, and **maxrow** to the **OPTION** of the **COPY TO** statement.

Change Type	No.	Name	Change Description
	16	CREATE FOREIGN TABLE (OBS import and export)	Added **bom** to the **OPTION** of the **CREATE FOREIGN TABLE** statement (OBS import and export).
	17	Operator =	If two Roaring Bitmaps are equal, **true** is returned. Otherwise, **false** is returned.
	18	Operator <>	If two Roaring Bitmaps are not equal, **true** is returned. Otherwise, **false** is returned.
	19	Operator &	Calculates the intersection of two Roaring Bitmaps.
	20	Operator |	Calculates the union of two Roaring Bitmaps.
	21	Operator |	Calculates the result of adding an ID to a Roaring Bitmap.
	22	Operator #	Calculates the XOR result of two Roaring Bitmaps.
	23	Operator -	Obtains the set of elements that are in a Roaring Bitmap but not in another Roaring Bitmap.
	24	Operator -	Removes a specified ID from a Roaring Bitmap.
	25	Operator @>	If the Roaring Bitmap before the operator contains the elements after the operator, **true** is returned. Otherwise, **false** is returned.
	26	Operator <@	If the Roaring Bitmap after the operator contains the elements before the operator, **true** is returned. Otherwise, **false** is returned.
	27	Operator &&	If two Roaring Bitmaps overlap, **true** is returned. Otherwise, **false** is returned.
	28	ALTER TABLE DROP KEY	Added the syntax for deleting indexes. The usage of this syntax is the same as that of **ALTER TABLE DROP INDEX**.

Change Type	No.	Name	Change Description
	29	ALTER TABLE CHANGE	Compatible with the **CHANGE** syntax in MySQL. Column names and column attributes can be modified at the same time.
	30	ALTER TABLE ... FIRST/ AFTER colname	The syntax layer supports the **FIRST**/**AFTER** syntax, but does not implement the actual semantics. It is applicable to the **ALTER TABLE ADD**/ **MODIFY**/**CHANGE COLUMN** scenario and is controlled by the parameter **skip_first_after_mysql**.
	31	AGG HINT	Added Agg hints for performance optimization.
	32	ALTER TABLE ... ALTER COLUMN cstore_cu_sample_ratio	To reduce the work of CU decompression, samples are concentrated in randomly selected CUs. Therefore, it is difficult to obtain data features. This field attribute is used to select more CUs without increasing the sampling ratio to facilitate data feature obtaining.
Modified	33	Table-level parameter **deltarow_threshold**	The default value of **deltarow_threshold** is set to **10000**, which specifies the upper limit of rows when column-store tables are imported to the Delta table. This parameter is valid only when the table-level parameter **enable_delta** is enabled.

Change Type	No.	Name	Change Description
	34	More data types are supported by Btree indexes.	The data types supported by the B-tree indexes are extended to support row store tables, column-store tables, and time series tables. btree_gin provides data types int2, int4, int8, float4, float8, timestamp with time zone, timestamp without time zone, time with time zone, time without time zone, date, interval, oid, money, "char", varchar, text, bytea, bit, varbit, macaddr, macaddr8, inet, cidr, uuid, name, bool, bpchar, and all enum types.
	35	**GRANT**/**REVOKE**/**ALTER DEFAULT PRIVILEGES**	The ALTER/DROP/VACUUM permissions are added.
	36	**ALTER TABLE ... DROP PARTITION**	The **DROP PARTITION** syntax can be used to delete multiple partitions.
	37	**DEFAULT** clause in the **CREATE**/**ALTER TABLE** statement	The **DEFAULT** clause in the **CREATE**/**ALTER TABLE** statement does not support suffix operators (currently, only the factorial suffix operator "!" is used). For example, create table t (a int default 3!) is not allowed.

Keywords

Table 1-39 Keywords

Change Type	No.	Name	Change Description
Added	1	MATERIALIZED	Added the **[NOT] MATERIALIZED** syntax after **with cte as**. As a non-reserved keyword, it can be still used as an object name. When it is used as a column alias, **AS** must be added.

Change Type	No.	Name	Change Description
	2	time_fill	It is used as the keyword of a time filling expression to output the **time_fill** column. It cannot be used as a function name or user-defined data type name.
	3	fill_first/fill_last/fill_avg	It is used as the keyword of a time filling expression to output the filled column. It cannot be used as a function name or user-defined data type name.
	4	list	Specifies the type of a partitioned table. As a non-reserved keyword, it can still be used as an object name. When it is used as a column alias, **AS** must be added.
	5	tsfield/tstag/tstime	Specifies the type of a partitioned table. As a non-reserved keyword, it can still be used as an object name. When it is used as a column alias, **AS** must be added.

System Catalogs

Table 1-40 System catalogs

Change Type	No.	Name	Change Description
Added	1	rb_added	A value is added to RoaringBitmap.
	2	pg_partition	Added the **boundexprs** column to **pg_partition**.
	3	pg_relfilenode_size	New system catalog
	4	pg_attribute	The **attkvtype** column is added to **pg_attribute** to record the **kvtype** of columns.

Change Type	No.	Name	Change Description
	5	pg_collation	The **case_insensitive** record is added to support case-insensitive behaviors.

System Functions

Table 1-41 System functions

Change Type	No.	Name	Change Description
Added	1	rb_build	Converts an int array into a bitmap.
	2	rb_to_array	Reverse operation of **rb_build**. It converts a Roaring Bitmap into an int array.
	3	rb_and	Obtains the intersection of two Roaring Bitmaps.
	4	rb_or	Obtains the union of two Roaring Bitmaps.
	5	rb_xor	Obtains the XOR result of two Roaring Bitmaps.
	6	rb_andnot	Performs AND then negation operation on two Roaring Bitmaps.
	7	rb_cardinality	Calculates the cardinality of a Roaring Bitmap.
	8	rb_and_cardinality	Calculates the cardinality of the And result of two Roaring Bitmaps.
	9	rb_or_cardinality	Calculates the cardinality of the OR result of two Roaring Bitmaps.
	10	rb_xor_cardinality	Calculates the cardinality of the XOR result of two Roaring Bitmaps.
	11	rb_andnot_cardinality	Calculates the cardinality of the andNot result of two Roaring Bitmaps.

Change Type	No.	Name	Change Description
	12	rb_is_empty	Determines whether a Roaring Bitmap is empty.
	13	rb_equals	Determines whether two Roaring Bitmaps are equal.
	14	rb_intersect	Determines whether two Roaring Bitmaps intersect.
	15	rb_contain	Determines whether the first Roaring Bitmap contains a specified value.
	16	rb_add	Adds a value to a Roaring Bitmap.
	17	rb_remove	Deletes a value from a Roaring Bitmap.
	18	rb_flip	Reverses the Roaring Bitmap within a specified range.
	19	rb_min	Obtains the minimum value of a Roaring Bitmap.
	20	rb_max	Obtains the maximum value of a Roaring Bitmap.
	21	rb_rank	Returns the cardinality of values in a bitmap that are less than or equal to the specified Offset.
	22	rb_contain_rb	Check whether the first Roaring Bitmap contains the second Roaring Bitmap.
	23	rb_containedby_rb	Check whether the second Roaring Bitmap contains the first Roaring Bitmap.
	24	rb_containedby	Determines whether a specified value is included in a specified Roaring Bitmap.
	25	rb_iterate	Returns the ints corresponding to a Roaring Bitmap.
	26	rb_and_agg	Aggregates a Roaring Bitmap column based on the AND logic.
	27	rb_or_agg	Aggregates a Roaring Bitmap column based on the OR logic.

Change Type	No.	Name	Change Description
	28	rb_xor_agg	Aggregates a Roaring Bitmap column based on the XOR logic.
	29	rb_and_cardinality_agg	Obtains the cardinality of the AND result of a Roaring Bitmap column.
	30	rb_or_cardinality_agg	Obtains the cardinality of the OR result of a Roaring Bitmap column.
	31	rb_xor_cardinality_agg	Obtains the cardinality of the XOR result of a Roaring Bitmap column.
	32	rb_build_agg	Aggregates the int column into RoaringBitmap data.
	33	pgxc_wlm_readjust_relfilenode_size_table()	Space statistics calibration function. It does not recreate the **PG_RELFILENODE_SIZE** system catalog but re-calibrates the user and schema space.
	34	gs_table_distribution()	Quickly queries a table size.
	35	pg_obs_cold_refresh_time	Modifies the time for automatic switchover of OBS multi-temperature tables.
	36	gs_clean_tag_relation	Deletes the useless data in the tagid row of the tag table. The input parameter of the function is the OID of the time series table. Each partition traverses the minimum value of the tagid column in the **cudesc** table to obtain the minimum tagid value in the entire time series table.
	37	proc_drop_partition	Drops a partition whose boundary time exceeds the TTL.
	38	proc_add_partition	Creates partitions for a partitioned table.
	39	pg_collation_actual_version	Returns the actual version number of an ICU collation.

Change Type	No.	Name	Change Description
	40	first	Indicates the first element in a group.
	41	last	Indicates the last element in a group.
	42	mode	Indicates the value with the highest occurrence frequency in a group.
	43	delta	Indicates the difference between two adjacent rows.
	44	percentile_of_value	Indicates the approximate percentile value.
	45	value_of_percentile	Indicates the approximate percentile.
	46	spread	Indicates the difference between the maximum value and minimum value in a group.
	47	pg_flush_buffers	Flushes all row-store dirty pages.
Modified	48	pg_stat_activity series views	The **stmt_type** and **lwtid** columns are added to the **pg_stat_get_activity_with_conninfo**, **pg_stat_activity**, **pgxc_stat_activity** and **pg_stat_get_activity** views.
	49	Added two rows to the **pg_authid** system catalog.	Added the preset roles **pg_role_signal_backend** and **pg_role_read_all_stats**.
	50	vac_fileclear_relation	Returns the number of files to be cleared in a specified table. If no files are to be cleared, 0 is returned.
	51	vac_fileclear_all_relation	Returns the number of files to be cleared in all column store tables. If no files are to be cleared, 0 is returned.

System Views

Table 1-42 System views

Change Type	No.	Name	Change Description
Added	1	pgxc_session_wlmstat	New view
	2	pg_comm_query_speed	New view. It is used to query sending information based on query_ids.
	3	pgxc_respool_resource_info pgxc_respool_runtime_info pgxc_respool_resource_history gs_respool_resource_info gs_respool_runtime_info	New views related to resource monitoring.
	4	pgxc_wait_detail pgxc_lockwait_detail	Added the enhanced view **pg_locks**.
Modified	5	DBA_PART_INDEXES DBA_PART_TABLES DBA_TAB_PARTITIONS USER_IND_PARTITIONS USER_PART_INDEXES USER_TAB_PARTITIONS	Added description for list partitions.
	6	pgxc_wlm_session_statistics	The logic for querying real-time information about Top SQL statements from all CNs is changed from serial to parallel to improve performance. The function remains unchanged.
	7	all_indexes	The definition of the view **all_indexes** in **sys** and **pg_catalog** is incorrect. If objects with the same name exist in different schemas, the result set expands.

Behavior Changes

Table 1-43 Behavior changes

Change Type	No.	Name	Change Description
Modified	1	In **CREATE INDEX**, the target table is a time series table.	Any index created in the time series table is converted to a double index in the tag table. The index column of the double index is the specified column that the index is created for.
	2	Secondary load management	Added secondary management to load management. Functions and multi-statements that contain complex queries may trigger multiple controls. You can set **enable_transaction_parctl** to **off** to disable secondary management. If you do so, transaction block statements and multi-statement management will also be disabled.
	3	Load management (autoanalyze management)	The control logic for queries triggering autoanalyze is changed from **no control** to **control**. You can set **enable_transaction_parctl=off** to disable the autoanalyze control.
	4	User monitoring view **pg_total_user_resource_info**	1. Changed the CPU and memory resource usage and limit to the resource usage and limit in the cluster. 2. The monitored CPU, I/O, and memory objects are changed from complex jobs to all jobs. 3. The CPU monitoring logic is changed from Cgroup monitoring to job CPU monitoring summary.

Change Type	No.	Name	Change Description
	5	Audit logs	1. Statements in a transaction are audited when the statement type is set to be audited, even if the transaction is not set to be audited. 2. The **DECLARE CURSOR** statement is audited when the GUC parameter **audit_operation_exec** is set to **select**.

GUC Parameters

Table 1-44 GUC parameters

Change Type	No.	Name	Change Description
Added	1	space_readjust_schedule	Indicates whether to enable automatic calibration for users and schema spaces. The options are **off**, **auto**, and **auto** (*xxx* K/M/G). The default value is **auto**.
	2	space_once_adjust_num	Threshold of the number of files processed each time during slow build and fine-grained calibration in the space management and statistics functions. The default value is **300**.
	3	tag_cache_max_number	Indicates the maximum number of items cached in the tag cache of a global hash table. The default value is **10000000**.
	4	enable_tagbucket_auto_adapt	Indicates whether to enable tagbucket adaptive adjustment. The default value is **on**.
	5	cache_tag_value_num	Number of cached tag tuples during tag column lateread. The default value is **60000**.

Change Type	No.	Name	Change Description
	6	buffer_ring_ratio	Controls the threshold and switch of the ring buffer in a row-store table. The default value is **250**, that is, 1/4 (250/1000), which is the same as the previous logic.
	7	enable_stream_ctescan	This parameter is added. By default, this parameter is set to **OFF** after an upgrade and is set to **ON** after a new installation. Specifies whether to enable the **share scan** function for the stream plan.
	8	behavior_compat_options	The **disable_including_all_mysql** option is added to control whether the **CREATE TABLE LIKE** syntax is in **INCLUDING ALL** mode. By default, this option is not set and the **CREATE TABLE LIKE** syntax is in **INCLUDING ALL** mode.
	9	profile_logging_module	The GUC parameter **profile_logging_module** is added to configure the type of performance logs to be recorded. By default, OBS, HADOOP, and REMOTE_DATANODE are enabled, and MD is disabled for both upgrade and new installation. The method of setting this parameter is the same as that of setting **logging_module**.
	10	object_mtime_record_mode	Controls the behaviors of **pg_object** recording modification time. By default, the behavior of the old version is retained. The new option can be set to not record the TRUNCATE/GRANT/REVOKE operations on partitions.
	11	skew_option	Specifies whether to enable an optimization policy.

Change Type	No.	Name	Change Description
	12	ddl_lock_timeout	DDL lock timeout interval.
Modified	13	rewrite_rule	• The **orconversion** option is added and is enabled by default. Indicates the OR optimization rules for equi-join. Change: The query plan converts nestloop to hashjoin. • The **projection_pushdown** option is added and is enabled by default. Change: Unused columns in subqueries, CTEs, and views are deleted.
	14	max_process_memory	It is an OM setting item configured during initial installation. The original formula is: Available memory x 0.7/(1 + Number of DNs). The coefficient 0.7 is changed to 0.8. It remains unchanged in scenarios such as upgrade and scale-out of the old cluster.
	15	enable_bloom_filter	Reused parameter. It is now used to control BloomFilter in a column-store table. The default value remains unchanged (**Enabled**).
	16	retry_ecode_list	Added the default value **45003** to solve the problem that wrong partitions may be queried when partitions are added and queried concurrently in a list partitioned table.
	17	auth_iteration_count	The default value is reduced from **50000** to **10000**. The change applies to both installation and upgrade. Indicates the number of hash iterations of the passwords of the client and server.

1.7.2 New Features and Resolved Issues in 8.1.3.x

8.1.3.336

Table 1-45 New features/Resolved issues in version 8.1.3.336

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-
Resolved issues	The error message " value too long for type character varying(512)" is displayed during foreign table import.	If the type of the ORC foreign table does not match the corresponding field type in the file (e.g., varchar (y) vs. string), an error will be reported during the import process due to the field length exceeding the limit.	8.1.1.100	Upgrade the version to 8.1.3.336 or later.
	Changes in OpenSSL affect the scheduler, causing errors during installation, scale-out, and scheduler functions.	OpenSSL changes the sequence of **LD_LIBRARY_PATH**, placing the system directory before the GAUSS directory, leading to incorrect **.so** files being found and startup failures.	Versions earlier than 8.1.3.336	

8.1.3.333

Table 1-46 New features/Resolved issues in version 8.1.3.333

Category	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-
Resolved issues	Permissions are lost after a cluster upgrade.	Upon upgrading to version 8.1.3.330, an error message appears indicating insufficient permission for certain link services. An earlier version contains a vulnerability that allows improper permission bypass. Upgrading to a version that addresses this issue results in a change in behavior.	8.1.3.330	Upgrade the version to 8.1.3.333 or later.
	The **ceil** function's output of **-0** affects service judgment.	The **ceil** function may return **-0** when rounding a float value. Use the **normalize_negative_zero** parameter to control whether **-0** is returned.	Versions earlier than 8.1.3.333	
	The sqlbuil fuzzing test causes a core dump "(ctePlan != __null && ctePlan->isCtePlan&&InitStreamFlow" in complex queries.	During the **finalize_node_id** phase, **initplan** is copied, and the sequence of **glob->subplans** is adjusted. However, the position of the **CTE** pointed to by **referencedCtePlanIds** is not adjusted, causing the **CTE** to point to an incorrect subplan and be set to NULL.	Versions earlier than 8.1.3.333	

Category	Feature or Resolved Issue	Cause	Version	Handling Method
	In GaussDB(DWS), queries using the decimal data type in Hive display null values.	GaussDB(DWS) and Hive process decimal data differently.	Versions earlier than 8.1.3.333	
	The **WITH RECURSIVE** statement runs indefinitely.	In the Arm environment, thread information synchronization is abnormal during the **WITH RECURSIVE** statement execution, causing variables to not update synchronously.	8.1.3.323	

8.1.3.330

Table 1-47 New features/Resolved issues in version 8.1.3.330

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	None	-	-	-
Resolved issues	Core dump during plan generation for **insert into select** statement.	Duplicate sublinks in ORCLAUSE conversion lead to double adjustment in OffsetVarNodes.	Versions earlier than 8.1.3.330	Upgrade the version to 8.1.3.330 or later.

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Accidental core dump "core:CStorePartitionInsert::findBiggestPartition" during column storage lightweight test.	During statistics collection for **bulkloadUsedMemSize** in a partitioned table using column-store, both **UPDATE** and **INSERT** operators are gathered by the system. However, the adaptive memory size is calculated separately for each operator when data is flushed to disks, which can result in **g_bulkloadUsedMemSize** being higher than expected. If the memory usage of the **UPDATE** operator reaches the estimated threshold, it could prevent an **INSERT** operator data record from being inserted.	Versions earlier than 8.1.3.330	
	Wait ccn queuing without reaching threshold.	Inability to calibrate global memory during CCN queuing blocks jobs.	Versions earlier than 8.1.3.330	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Unauthorized table queries possible through nesting.	The light proxy's permission check relies on the **ExecCheckRTPerms** function within the CN's checkLightQuery procedure. This check is not extended to DNs. For **ExecCheckRTPerms** to operate correctly, the **rangeTables** parameter should be set to **RTE_RELATION**. If not, the function defaults to returning **true**. However, this mechanism fails for nested SQL statements, where the type changes to **RTE_SUBQUERY**, skipping the necessary permission validation.	Versions earlier than 8.1.3.330	
	Occasional "canceling statement due to statement timeout" error with deletion statement.	The transaction time is not reset for the w packet, leading to thread reuse.	Versions earlier than 8.1.3.330	
	Memory leak when querying JSON type data.	Memory is not released in the jsonb **out** function, causing high usage.	Versions earlier than 8.1.3.330	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	"Tuple concurrently updated" error during concurrent job execution.	The OID of a partition is identical to that of its corresponding table. During the execution of the ANALYZE command, the **pg_object** catalog is updated with the OID of the partition. This update process also applies to the **pg_object** record of the table itself. Concurrently, DDL operations may be executed. For instance, if an ALTER TABLE command is executed, it will modify the **pg_object** record of the table, which could potentially result in an error.	Versions earlier than 8.1.3.330	

8.1.3.325

Table 1-48 New features/Resolved issues in version 8.1.3.325

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	• The GB18030_2022 character encoding is supported. • The subquery hint function can be disabled. • The GDS foreign table is compatible with invalid UTF-8 characters.	-	-	-
Resolved issues	Logs fail and the cluster is hung after DN restart.	When the stream thread error logs are printed, the stream thread responds to the cancel signal. When the logs are printed again, the stream thread is hung up.	8.1.3.320	Upgrade the version to 8.1.3.325.
	• When a service table is created, the following error message is occasionally displayed: **relation "xxx" already exists**. • The error **could not read block xxx in file "xxx"** is occasionally reported.	When the OID usage surpasses 4.2 billion, the system reallocates OIDs. During this reallocation process, intermittent errors may occur.	8.1.3.323 and earlier versions	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Checkpoint failures on the standby DN prevent the reclamation of xlogs, leading to their continuous accumulation.	During the **redo checkpoint** operation on the standby DN, the detection of incomplete btree splitting triggers an RM 11 error. This may indicate a btree splitting failure or that the process is still ongoing.	8.1.3.323 and earlier versions	
	If a lightweight update occurs on a column-store table while autovacuum runs in the background, the system may report an error **Unsupported to update different rows with the same cuid and col in light update**.	In lightweight update, the same cuid and col cannot be used to update different rows.	8.1.3.323 and earlier versions	
	The **no_merge** hint operation does not take effect in some scenarios.	The subquery hint function cannot be disabled.	8.1.3.323 and earlier versions	
	When a long-time query statement is executed on the background, if you press Ctrl+C to end the statement and then press Ctrl+Z to exit the gsql client, residual statements exist.	Upon receiving the CTRL+C signal, the CN thread enters the error jump process. In this case, the CN thread invokes EmitErrorReport, but EmitErrorReport does not respond to the signal.	8.1.3.323 and earlier versions	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	When a wide table is used with a real-time table for associated update, the memory usage is too high, affecting the performance.	Associating a wide table with a real-time table for updates introduces superfluous columns. This leads to excessive memory consumption during the optimizer's execution plan estimation, thereby deteriorating performance	8.1.3.323 and earlier versions	
	In the ShareScan plan, the memory of the tuplestore and batchstore for storing CTEs cannot be expanded. As a result, data may be written to disks.	cteMaxMem in the ShareScan execution plan is not copied, read, or written. As a result, the DN fails to obtain the value from the CTE execution plan, and the created tuplestore and batchstore for storing the CTE result set cannot be expanded in memory. As a result, data spills to disks.	8.1.3.323 and earlier versions	
	The error message **could not determine data type of parameter** is reported when a lightweight column-store update is performed.	In the remote query of the execution plan, the **parame** parameter is specified as its position in the base table, and the system column cannot match this parameter.	8.1.3.323 and earlier versions	

8.1.3.322

Table 1-49 New features/Resolved issues in version 8.1.3.322

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	The default value of the GUC parameter **enable_limit_stop** is changed to **on**.	-	-	-
Resolved issues	None	-	-	-

8.1.3.320

Table 1-50 New features/Resolved issues in version 8.1.3.320

Type	Feature or Resolved Issue	Cause	Version	Handling Method
New features	Concurrent truncate, exchange, and select operations Added the GUC parameter **ddl_select_concurrent_mode**. This parameter is used to solve the problem that the query statement lasts for a long time and blocks DDL in the scenario where the data volume is large or the query is complex. The effect is the same as that of the Oracle database. Application scenarios: ● Concurrent truncate and select operations ● Concurrent exchange and select operations Scenario constraints ● Concurrency is not supported when there are conflicts with locks of higher levels (more than one level). For example, **autoanalyze** is triggered by **select** when **autoanalyze_mode** is set to **normal**. ● Concurrency is not supported when there are conflicts with locks in transaction blocks.	-	-	-

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Connection constraints: • When the JDBC, ODBC, or Python driver is used, if **autocommit** is set to **false**, this feature is not supported. • This feature is not supported by query statements delivered by Data Studio and auto-committed transactions.			
	Whether to enable early stop optimization for LIMIT statements can be controlled. Added the GUC parameter **enable_limit_stop** to specify whether to enable the early stop optimization for LIMIT statements.	-	-	-
Resolved issues	The statement-level estimated memory exceeds the value of **max_process_memory** and causes ccn queuing.	The statement-level estimated memory exceeds the value of **max_process_memory**.	8.1.3.310 and earlier versions	Upgrade the version to 8.1.3.320 or later.
	When the MERGE operation is performed, an error message**value xxx is out of range for type integer** is displayed.	The **pg_toast_get_baseid()** function uses an integer interface for OID processing.	8.1.3.310 and earlier versions	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	Querying the dirty page rate view of a large table containing 200,000 data rows results in significant memory consumption.	In the process of data querying, each node in the statistics-querying linked list requests memory to store tupStatus. Post-query, while the nodes' memory is freed, the allocated memory within the nodes for tupStatus remains unreleased.	8.1.1.x	
	When a Vacuum operation times out and is terminated, some DNs retain leftover threads, which then fail to respond to signals and obstruct the execution of subsequent statements.	The btvacuumscan thread does not respond to signals. As a result, after the Vacuum execution times out and is killed, some DNs have residual threads and cannot respond to signals.	8.1.1.x	
	Executing percentage sampling on partitioned and time series tables within the MySQL compatibility library triggers an error: "unsupported feature with temporary/ unlogged table for partitioned table".	In 8.1.3, the default inheritance mode is INCLUDING ALL to match the CREATE TABLE LIKE syntax in MySQL. However, this causes an error during ANALYZE percentage sampling because the command for automatically creating temporary replication tables is included the sampling.	8.1.3.310 and earlier versions	

Type	Feature or Resolved Issue	Cause	Version	Handling Method
	The SELECT query of the cold and hot tables triggers **runtime autoanalyze**, deteriorating the query performance.	If the select operation is performed After the TRUNCATE PARTITION, EXCHANGE PARTITION, DROP PARTITION, ALTER DISTRIBUTE, or ALTER COLUMN TYPE operation, AUTOANALYZE is triggered.	8.1.3.310 and earlier versions	
	The value of **elapsed_time** in **pg_session_wlmstat** is inconsistent with the actual execution time of the top SQL statement.	When querying the **pg_session_wlmstat** view, the **elapsed_time** value appears excessively high for statements in the RUNNING state.	8.0.x	
	When the SELECT permission is granted to a foreign table, the error message "has no distribute type" is displayed.	The default distribution mode is not specified for the OBS dfs server write-only foreign tables.	8.1.3.310 and earlier versions	

1.8 Version 8.1.1

V8.1.1.500 release date: June 20, 2022

[Release date of V8.1.1.300]: April 15, 2022

[Release date of V8.1.1.205]: March 30, 2022

[Release date of V8.1.1.203]: March 18, 2022

[Release date of V8.1.1.202]: February 24, 2022

[Release date of V8.1.1.201]: January 25, 2022

[Release date of V8.1.1.200]: December 9, 2021

[Release date of V8.1.1.100]: July 30, 2021

NOTE

The database kernel functions of clusters 8.1.1.200, 8.1.1.201, 8.1.1.202, 8.1.1.203, 8.1.1.205, 8.1.1.300, 8.1.1.500, and 8.1.1.100 are the same. Some functions are adapted to the management console. For details, see **What's New**.

New features

Table 1-51 New Functions in 8.1.1

Feature	Description	Reference
2048 nodes in a cluster	Supported up to 2048 nodes in a cluster.	Technical Specifications
Resource monitoring performance tuning	1. Added functions **pgxc_wlm_get_schema_space(cstring)** and **pgxc_wlm_analyze_schema_space(cstring)**. 2. Added system views **PGXC_TOTAL_SCHEMA_INFO**, **PGXC_TOTAL_SCHEMA_INFO_ANALYZE**, and **GS_WLM_SQL_ALLOW**.	• Functions pgxc_wlm_get_schema_space(cstring) and pgxc_wlm_analyze_schema_space(cstring) • System views PGXC_TOTAL_SCHEMA_INFO, PGXC_TOTAL_SCHEMA_INFO_ANALYZE, and GS_WLM_SQL_ALLOW
Lightweight **UPDATE**	1. Added descriptions about column-store tables and the lightweight UPDATE operation on column-store tables to "Precautions". 2. Added the GUC parameter **enable_light_colupdate**.	• UPDATE • enable_light_colupdate
Primary key CU of column-store tables	Supported primary key constraints of column-store tables.	• Constraint Design • ALTER TABLE • CREATE INDEX
Space management	Added the GUC parameter **bi_page_reuse_factor**.	bi_page_reuse_factor

Feature	Description	Reference
GDS productization	Added system views **PGXC_BULKLOAD_PROGRESS**, **PGXC_BULKLOAD_STATISTICS**, and **PG_BULKLOAD_STATISTICS**.	• PGXC_BULKLOAD_PROGRESS • PGXC_BULKLOAD_STATISTICS • PG_BULKLOAD_STATISTICS
Hot and cold data storage	1. Added the description of **REFRESH STORAGE** to "ALTER TABLE". 2. Added the functions and descriptions of **COLVERSION** and **STORAGE_POLICY** to "CREATE TABLE". 3. Added the function and description of the OBS tablespace to "CREATE TABLESPACE".	• ALTER TABLE • CREATE TABLE
C function in SM4 CBC mode	Added security functions **gs_encrypt(encryptstr, keystr, cryptotype, cryptomode, hashmethod)**, **gs_decrypt(decryptstr, keystr, cryptotype, cryptomode, hashmethod)**, and **gs_hash(hashstr, hashmethod)**. Modified the description of **gs_encrypt_aes128(encryptstr, keystr)** and **gs_decrypt_aes128(decryptstr, keystr)**.	Security Functions
Support for the built-in **MEDIAN** function	Added the function **median(expression)**.	Aggregate Functions
Adjusting the cascade delete for tables with views	1. Added the description of rebuilding a view. 2. Added the following description to the **REBUILD** parameter of **ALTER VIEW**: Only columns of the character, number, and time types in the base table can be modified. When **view_independent** is set to **on**, views can be automatically rebuilt.	• Creating and Managing Views • ALTER VIEW

Feature	Description	Reference
Custom data redaction	1. Optimized data redaction. 2. Provided data redaction functions in different forms. 3. Updated the columns in the system catalog **PG_REDACTION_COLUMN** and system view **REDACTION_COLUMNS**. 4. Added the syntax **ALTER REDACTION POLICY**, **CREATE REDACTION POLICY**, and **DROP REDACTION POLICY**.	• Data Redaction • Data Redaction Functions • PG_REDACTION_COLUMN and REDACTION_COLUMNS • ALTER REDACTION POLICY • CREATE REDACTION POLICY • DROP REDACTION POLICY
After the password expires, a user can log in to the database only after changing the password.	1. "Keyword" in the *SQL Syntax* Added **EXPIRATION** (non-reserved). 2. Added the security function **gs_password_expiration**, and the description that the system prompts users to change their passwords after the **gs_password_deadline** password expires. This is related to the GUC parameter **password_effect_time**. 3. Added **PASSWORD EXPIRATION period** to the syntax **ALTER ROLE**, **ALTER USER**, and **CREATE USER**. 4. Added the following description to **CREATE ROLE**: number of days before the login password of a role expires. A user needs to change the password before it expires. If the login password expires, the user cannot log in to the system. In this case, the user needs to ask the administrator to set a new login password and use it to log in to the system. 5. Added the columns **rolauthinfo**, **rolpwdexpire**, and **rolpwdtim** to the system catalog **PG_AUTHID**.	• Keywords • Security Functions • ALTER ROLE, ALTER USER, and CREATE USER • CREATE ROLE • PG_AUTHID
Increasing the hash table size	Added the GUC parameter **expand_hashtable_ratio**.	expand_hashtable_ratio

Feature	Description	Reference
SMP adaptation enabled	Added "Suggestions for SMP Parameter Settings".	Suggestions for SMP Parameter Settings
PRETTY as the default value of **EXPLAIN**	Changed the default display format of **EXPLAIN** to **PRETTY**. Added the statistics of filtering and projection time to the new operator. The statistics and the operator execution time are displayed in the same row.	SQL Execution Plan Details
Enhanced concurrent SMP	1. Added the constraint of SMP: the short query scenario where the plan generation time accounts for a large proportion. 2. Added the description of setting **query_dop** to **1** in the short query scenario. 3. Added the GUC parameter **query_dop_ratio**.	• SMP Application Scenarios and Restrictions • SMP Manual Optimization Suggestions • query_dop_ratio
row2vec optimization	Added the GUC parameter **enable_row_fast_numeric**.	enable_row_fast_numeric

Feature	Description	Reference
MySQL compatibility	1. Added the negative processing of **concat**, **log(x)**, **left**, and **right**, **last_day** and **next_day** return types of **int + unknown** operations, and compatibility differences of the operator ^. 2. Added the keywords **IFNULL** and **TIMESTAMPDIFF**. 3. Added the description of processing **CASE**, **COALESCE**, **IF**, and **IFNULL** in MySQL-compatible mode. 4. Added the MySQL-compatible schema and example of the **behavior_compat_options** option. 5. **[OFFSET start [ROW | ROWS]] | LIMIT start, { count | ALL } }]** supported by **SELECT**, **SELECT INTO**, and **VALUES**. 6. Added MySQL compatibility to the **DBCOMPATIBILITY** of **CREATE DATABASE**. 7. Added the following content: compatibility differences between **concat(str1, str2)**, **left(str text, n int)**, and **right(str text, n int)** for character processing functions and operators; compatibility differences between **log(x)** and ^ for numeric operation functions and operators; time and date processing functions and operators, **timestampdiff(field, timestamp1, timestamp2) / to_days(timestamp) / data_format**; conditional expression functions, including **if(bool_expr, expr1, expr2)**, **ifnull(expr1, expr2)**, and **isnull(expr)**.	• Syntax Compatibility Differences Among Oracle, Teradata, and MySQL • Keywords • UNION, CASE, and Related Constructs • behavior_compat_options • SELECT, SELECT INTO, and VALUES • CREATE DATABASE • Functions and Operators
Support for **UPSERT**	Added the **UPSERT** syntax. Added **IGNORE**, **AS**, **ON DUPLICATE KEY**, and **ON CONFLICT** to the **INSERT** syntax.	UPSERT

Feature	Description	Reference
Support for the **INSERT**, **UPDATE**, and **DELETE** operations of views	Added **Updatable Views** to "CREATE VIEW".	CREATE VIEW
One-click onsite information collection	Added the **STATS boolean** parameter to the **EXPLAIN** syntax.	EXPLAIN
One-click relationship analysis of a lock wait	1. Added the lock information function **pgxc_get_lock_conflicts()**. 2. Added the system views **PGXC_DEADLOCK** and **PGXC_LOCK_CONFLICTS**.	• System Information Functions • System views PGXC_DEADLOCK and PGXC_LOCK_CONFLICTS
Combination with an empty string of the **CHAR** type	Added the configuration item **bpchar_text_without_rtrim** to the parameter **behavior_compat_option**.	behavior_compat_options
Support for Python 3.*x*	Supported Python 3.	Using GDS to Import Data from a Remote Server
When **to_date** and **to_timestamp** process an empty string, **0001-01-01** is returned, and **null** is returned for TD.	1. Added "Syntax Compatibility Differences Among Oracle, Teradata, and MySQL". 2. Added the configuration item **convert_empty_str_to_null_td** to **behavior_compat_options**.	• Syntax Compatibility Differences Among Oracle, Teradata, and MySQL • behavior_compat_options
Support for **INSERT OVERWRITE**	Supported **INSERT OVERWRITE**.	INSERT
Obtaining all views queried by the current user	Added the view **GS_VIEW_DEPENDENCY_PATH**.	GS_VIEW_DEPENDENCY_PATH
Support for **percentile_disc** and **percentile_cont**	Added the following aggregate functions: **percentile_disc(const) within group(order by expression)** **percentile_cont(const) within group(order by expression)**	Aggregate Functions

SQL Syntax

Table 1-52 SQL syntax

Change Type	No.	Name	Change Description
Add	1	LIMIT offset, count	The **LIMIT offset,count** syntax is supported.
	2	EXPLAIN(STATS ON)...	The **EXPLAIN(STATS ON)** syntax is supported. It is used to export information for regenerating a plan.
	3	CREATE/ALTER/DROP REDACTION POLICY	Added the syntax for masking DDL statements.
	4	INSERT IGNORE INTO INSERT INTO ON DUPLICATE KEY UPDATE INSERT INTO ON CONFLICT DO UPDATE INSERT INTO ON CONFLICT DO NOTHING	**UPSERT** is supported.
	5	INSERT OVERWRITE INTO	**INSERT OVERWRITE** is supported.
	6	ALTER TABLE REFRESH STORAGE	Users can change hot data to cold data in tables.
	7	ALTER VIEW ONLY view_name REBUILD	**ALTER VIEW ONLY** *view_name* **REBUILD** is supported.
	8	ALTER SERVER REFRESH	You can update the configuration file of the HDFS server if the HDFS configuration was modified.
Modify	9	Operator	In MySQL compatibility mode, XOR is used. In ORA or TD mode, exponentiation is used.
	10	MERGE INTO	An error will be reported if the target table of the **MERGE INTO** statement contains triggers.
	11	CREATE/ALTER Table	Added options about **cold_tablespace** and **storage_policy** in **Reloptions**.

Keywords

Table 1-53 Keywords

Change Type	No.	Name	Change Description
Add	1	TIMESTAMPDIFF	The **COL_NAME_KEYWORD** keyword is added.
	2	IFNULL	The **COL_NAME_KEYWORD** keyword is added.
	3	REFRESH	Non-reserved keyword
Modify	4	IF	**UNRESERVED_KEYWORD** is changed to **COL_NAME_KEYWORD**.
	5	ISNULL	**UNRESERVED_KEYWORD** is changed to **COL_NAME_KEYWORD**.
	6	VERIFY	This keyword is changed from reserved to non-reserved.

System Catalogs

Table 1-54 System catalogs

Change Type	No.	Name	Change Description
Modify	1	PG_REDACTION_COLUMN	The **func_expr** column is added.
	2	PG_AUTHID	The **rolpwdexpire** and **rolpwdtime** columns are added.

System Functions

Table 1-55 System functions

Change Type	No.	Name	Change Description
Add	1	pgxc_wlm_readjust_schema_space()	You can perform parallel calibration in schema space.

Change Type	No.	Name	Change Description
	2	pgxc_wlm_get_schema_space()	You can query the schema space information of the cluster.
	3	pgxc_wlm_analyze_schema_space()	You can analyze the schema space information of the cluster.
	4	gs_roach_enable_delay_ddl_recycle	You can enable DDL delay by specifying a backup set name.
	5	gs_roach_disable_delay_ddl_recycle	You can disable DDL delay by specifying a backup set name.
	6	gs_roach_stop_backup	You can disable row-store backup by specifying a backup set name.
	7	gs_roach_switch_xlog	You can configure whether to perform the checkpoint operation for xlog switch.
	8	pgxc_get_lock_conflicts()	This function returns information about conflicting locks on nodes.
	9	mask_none/mask_full/mask_partial	A built-in masking function is added.
	10	median	The median agg function is added.
	11	pgxc_node_stat_reset_time()	You can query the reset time of each node.
	12	pgxc_rel_iostat()	You can query the I/O statistics of each node.
	13	pgxc_redo_stat()	You can query the redo statistics of each node.
	14	pgxc_instance_time()	You can query the time statistics of each instance.
	15	pgxc_settings()	You can query the GUC settings of each node.
	16	pgxc_replication_slots()	You can query the replication slot statistics of each node.
	17	pgxc_stat_replication()	You can query the replication statistics of each node.

Change Type	No.	Name	Change Description
	18	pgxc_stat_bgwriter()	You can query the statistics of the bgwriter process on each node.
	19	pgxc_wait_events()	You can query statistics on the wait events of each node.
	20	pgxc_os_run_info()	You can query the OS performance statistics of the servers where instances are deployed.
	21	get_node_stat_reset_time()	You can query the time when the current node status is reset.
	22	get_local_rel_iostat()	You can query the I/O statistics of the current node.
	23	get_instr_wait_event()	You can query statistics on the wait events of the current node.
	24	pg_stat_get_db_total_blk_write_time()	You can query the total block write time.
	25	pg_stat_get_db_total_blk_read_time()	You can query the total block read time.
	26	pg_stat_get_db_total_temp_bytes()	You can query the total size of temporary files.
	27	pg_stat_get_db_total_temp_files()	You can query the total number of temporary files.
	28	pg_stat_get_db_total_deadlocks()	You can query the total number of deadlocks.
	29	pg_stat_get_db_total_conflict_all()	You can query the total number of conflicts.
	30	pg_stat_get_db_total_tuples_deleted()	You can query the total number of deleted tuples.
	31	pg_stat_get_db_total_tuples_updated()	You can query the total number of updated tuples.
	32	pg_stat_get_db_total_tuples_inserted()	You can query the total number of inserted tuples.
	33	pg_stat_get_db_total_tuples_fetched()	You can query the total number of fetched tuples.

Change Type	No.	Name	Change Description
	34	pg_stat_get_db_total_tuples_returned()	You can query the total number of returned tuples.
	35	pg_stat_get_db_total_blocks_hit()	You can query the total number of hit data blocks in the memory.
	36	pg_stat_get_db_total_blocks_fetched()	You can query the total number of read data blocks.
	37	pg_stat_get_db_total_xact_rollback()	You can query the total number of rollback transactions.
	38	pg_stat_get_db_total_xact_commit()	You can query the total number of commit transactions.
	39	pg_stat_get_db_total_numbackends()	You can query the total number of backends.
	40	gs_encrypt()	Encryption function
	41	gs_decrypt()	Decryption function
	42	gs_hash()	Hash function
	43	gs_password_expiration()	You can query the remaining password validity period (**rolpwdtime** and **rolpwdexpire** columns in the **pg_authid system** catalog) of the current user.
	44	to_char(date)	The to_char(date) function is added to be compatible with the return format of the DATE timestamp processed by to_char in TD mode.
	45	gs_wlm_set_queryband_action	You can configure the query_band load behavior.
	46	gs_wlm_set_queryband_order	You can configure the search priority of query_band.
	47	gs_wlm_get_queryband_action	You can query the **query_band** load behavior.
	48	CONCAT	A MYSQL compatibility rule is added. If the input parameter contains NULL, NULL will be returned.

Change Type	No.	Name	Change Description
	49	TIMESTAMPDIFF	The TIMESTAMPDIFF function can be used to return the time difference in a specified unit.
	50	TO_DAYS	The TO_DAYS function can return the number of days between the input time and 0.
	51	DATE_FORMAT	The DATE_FORMAT function can convert an input date and time into a string in a specified format. The string is in MYSQL format and starts with %.
	52	IF	The IF function can return the corresponding value based on the first input.
	53	IFNULL	The IFNULL function can return the first non-NULL value.
	54	ISNULL	The ISNULL function can check whether the input is NULL.
Modify	55	pg_resume_bkp_flag	You can obtain the delay DDL flag during backup and restoration.
	56	pg_query_audit	The **begintime**, **operation_type**, **command_text**, **transaction_xid** and **query_id** fields are added to audit logs. The **time** field is changed to **endtime**, and the **type** field is changed to **audit_type**. In addition, **detail_info** is used to store the command execution results instead of commands, which have been moved to **command_text**.
	57	pg_delete_audit	For security purposes, the interface for deleting audit logs cannot be provided. If this function is called, an error will be reported.

Change Type	No.	Name	Change Description
	58	log()	Logarithmic function. In ORA or TD mode, it represents a logarithm to base 10. In MySQL mode, it represents a natural logarithm.
	59	to_number	The to_number function instead of the numeric_in function is called. In TD mode, if the input is an empty string, null will be returned.
	60	left()	You can truncate a string. If the input parameter is a negative number, **-n** for example, all characters except the last */n/* character will be returned in ORA or TD mode, and an empty string will be returned in MySQL mode.
	61	right()	You can truncate a string. If the input parameter is a negative number, **-n** for example, all characters except the first */n/* character will be returned in ORA or TD mode, and an empty string will be returned in MySQL mode.
	62	last_day	Input parameters support the timestamptz type. In MySQL mode, the return type is date. In ORA or TD mode, the return type is timestamp.
	63	next_day	Input parameters support the timestamptz type. In MySQL mode, the return type is date. In ORA or TD mode, the return type is timestamp.
	64	add_months	Input parameters support the **timestamptz** type. In MySQL mode, the return type is **timestamptz**, while in ORA or TD mode, it is **timestamp**.

Change Type	No.	Name	Change Description
	65	add_months	The **date** type is added to the input parameter. In MySQL compatibility mode, the return type is **date**, whereas in ORA or TD mode, it is **timestamp**.
	66	pg_cbm_recycle_file	This parameter is added so that the concurrent backup and disaster recovery features can recycle CBM.
	67	pgxc_query_audit	The **begintime**, **operation_type**, **command_text**, **transaction_xid** and **query_id** fields are added to audit logs. The **time** field is changed to **endtime**, and the **type** field is changed to **audit_type**. In addition, **detail_info** is used to store the command execution results instead of commands, which have been moved to **command_text**.
	68	login_audit_messages	Enhanced the audit log feature.
Delete	69	add_policy/drop_policy/ alter_policy/enable_policy/ disable_policy	The old data making interface was deleted.

System Views

Table 1-56 System views

Change Type	No.	Name	Change Description
Add	1	GS_WLM_SQL_ALLOW	You can query the existing whitelist.
	2	PG_TOTAL_SCHEMA_INFO	You can query the schema space information of a node.
	3	PGXC_TOTAL_SCHEMA_INFO	You can query the schema space information of a cluster.

Change Type	No.	Name	Change Description
	4	PGXC_TOTAL_SCHEMA_INFO_ANALYZE	You can analyze the schema space information of the cluster.
	5	PGXC_LOCK_CONFLICTS	You can query the information about conflicting locks on each node.
	6	PGXC_DEADLOCK	You can query the information about lock waits in a distributed deadlock (including information about lock objects and lock holders).
	7	PGXC_NODE_STAT_RESET_TIME	You can query the node reset time.
	8	GS_NODE_STAT_RESET_TIME	You can check the reset time of the current node.
	9	GLOBAL_STAT_DATABASE	You can query global database statistics.
	10	GLOBAL_REL_IOSTAT	You can query global I/O statistics.
	11	PGXC_REL_IOSTAT	You can query node I/O statistics.
	12	GS_REL_IOSTAT	You can query the I/O statistics of the current node.
	13	GLOBAL_REDO_STAT	You can query global redo statistics.
	14	PGXC_REDO_STAT	You can query node redo statistics.
	15	GLOBAL_WORKLOAD_TRANSACTION	You can query global workload transaction statistics.
	16	PGXC_INSTANCE_TIME	You can query instance time statistics.
	17	PGXC_SETTINGS	You can query node GUC settings.
	18	PGXC_REPLICATION_SLOTS	You can query the replication slot statistics of the nodes.
	19	PGXC_STAT_REPLICATION	You can query the replication statistics of the nodes.

Change Type	No.	Name	Change Description
	20	PGXC_STAT_BGWRITER	You can query node bgwriter statistics.
	21	PGXC_WAIT_EVENTS	You can query wait events on the nodes.
	22	GS_WAIT_EVENTS	You can query wait event statistics of the current node.
	23	PGXC_OS_RUN_INFO	You can query the OS performance statistics of the servers where instances are deployed.
	24	PG_LIFECYCLE_DATA_DISTRIBUTE	You can query the data distribution statistics of OBS cold and hot tables.
	25	PG_BULKLOAD_STATISTICS	You can encapsulate the pg_stat_get_node_bulkload_statistics function.
	26	PGXC_BULKLOAD_STATISTICS	You can encapsulate the pgxc_stat_get_node_bulkload_statistics function.
	27	PGXC_BULKLOAD_PROGRESS	This function can aggregate the query results of the **pgxc_bulkload_statistics** view and calculate the GDS service progress percentage.
	28	PG_QUERYBAND_ACTION	You can query all the query_band load behaviors.
	29	GS_VIEW_DEPENDENCY_PATH	You can query the dependency between cascading query views.
Modify	30	REDACTION_COLUMNS	The **function_infos** column is added to the system view definition.
	31	PGXC_GET_TABLE_SKEWNESS	Fixed the problem that the round-robin table is not displayed in the **PGXC_GET_TABLE_SKEWNESS** view.
	32	PGXC_STAT_DATABASE	You can query database statistics on each instance.

Behavior Changes

Table 1-57 Behavior changes

Change Type	No.	Name	Change Description
Modify	1	If the Not-null and CHECK constraints conflict, the details of the entire row are not printed.	If the Not-null and CHECK constraints conflict, the details of the entire row are not printed.
	2	Compatible with C80 behavior. By default, the implicit type conversion from interval to text is not matched.	If **behavior_compat_options** is set to **enable_interval_to_text**, this implicit type conversion is supported.
	3	Date type	In ORA mode, the date type is reversely parsed as **pg_catalog.date**.
	4	Agg plan	Required columns are generated when agg generates a stream plan.
	5	Median is used with collate.	Collate cannot be used with median, percentile_cont, or percentile_disc.
	6	Modifications on multi-column partitioning	The rule to check the boundary values of multiple columns during partition modification. If the boundary value of a column is the maximum value, the boundary values of other columns are ignored.

Change Type	No.	Name	Change Description
	7	Generate a plan with **enable_nestloop** is set to **off** and no association conditions.	Assume there are two tables that do not have equivalent join conditions with each other, but have equivalent join conditions with other tables. If the number of joined rows increases, the nestloop plan can be executed. After the rectification, the nestloop plan cannot be executed. For equivalent join columns that do not support hashjoin, if **mergejoin** is set to **off**, the nestloop cost will change. The execution plan will change from Nestloop + Indexscan to Nestloop + Materialize.

GUC Parameters

Table 1-58 GUC parameters

Change Type	No.	Name	Change Description
Add	1	behavior_compat_options	• The **convert_empty_str_to_null_td** option is added to be compatible with the to_date, to_timestamp, and to_number functions that return null after processing an empty string in TD mode. • The **enable_interval_to_text** option is added to determine whether to support the implicit conversion from interval to text.
	2	debug_group_dop	You can configure the DOP of each stream group for statement-level optimization during SMP adaptation. This value is left empty by default.

Change Type	No.	Name	Change Description
	3	enable_row_fast_numeric	Numeric data in row-store tables is flushed to disks in bigint format.
	4	expand_hashtable_ratio	Dynamic extension of the hash table
	5	query_dop_ratio	A logic DOP makes decisions based on the system resources and cost. If the decisions are inaccurate, a parameter is added for escape. The default value is **1**.
	6	show_unshippable_warning	A parameter is added to determine whether to print logs that have not been pushed down to the client. By default, the function is disabled.
	7	assign_abort_xid	Session-level parameters are added. If data is deleted by mistake, incomplete recovery can be performed quickly. This parameter is left blank by default. The GUC parameter indicates rolling back the transaction submitted by a specific xid.
	8	bi_page_reuse_factor	Idle FSM factor during batch insert reuse (**0** indicates returning to the old logic)
	9	check_cu_size_threshold	Userset level. When data is inserted to a column-store table, if the amount of data inserted to a CU is greater than **check_cu_size_threshold**, row size will be checked to avoid generating CUs greater than 1 GB.
	10	enable_light_colupdate	Userset level, indicating whether lightweight update is enabled.

Change Type	No.	Name	Change Description
Modify	11	cost_model_version	The parameter value option **2** is added. The default value **1** remains unchanged. If the value is less than 2, the original function will be used for fixed value sampling. If the value is greater than 2, the function improved to enhance randomness is used for fixed value sampling.
	12	expected_computing_node group	After an in-place upgrade, set **expected_computing_nodegroup** to **bind**.
	13	query_dop	For a newly installed cluster, the default value is changed to **1**. For an upgraded cluster, the default value remains unchanged.

1.9 Version 8.1.0

Release date: May 15, 2021

Cluster version: 8.1.0.100

Feature Changes

Table 1-59 8.1.0 feature changes

Feature	Description
Enhanced **ANALYZE** for temporary tables and single-transaction operations.	Transactions and stored procedures support the ANALYZE command. No error is reported when ANALYZE and ALTER TABLE are executed concurrently. The automatic analyze function supports temporary tables.
Improved performance of the vectorized executor.	• Enhanced vectorized execution (HASH FULL JOIN) • string_agg/listagg supports vectorization.
Partition pruning with non-constant partition column filter criteria	Partition pruning is optimized. Currently, partition pruning can be performed only for constants. Variables that need to be calculated cannot be pruned and need to be optimized.

Feature	Description
Schema-level full backup + Table- and schema-level fine-grained restoration	Schema-level full backup + Table- and schema-level fine-grained restoration
Supported third-party user-defined data conversion of GDS.	Supported third-party user-defined data conversion.
Optimized RTO.	RPO = 0 and RTO < 60s in intra-cluster HA scenarios; RTO < 30s in POC scenarios
SQL on HDFS	Supported ORC data export using OBS foreign tables.
Supported XML function parsing.	Supported XML function parsing.
Modified distribution columns.	Modified distribution columns.
Supported automatic job migration when a CN fails.	If the current CN is faulty, its jobs can be automatically migrated.
Supported smooth upgrade of C functions.	Supported smooth upgrade of C functions.
Supported Teradata DSQL compatibility of gsql.	1. Supported equivalent comparison of dynamic variable strings (excluding other logical operations, AND/NOR logic, and nesting). 2. GOTO label 3. The THEN module supports only SQL and GOTO statements.
Synchronized the time zone with the IANA Time Zone Database (tz).	Synchronized the time zone with the IANA Time Zone Database (tz).

SQL Syntax

Table 1-60 SQL Syntax

Change Type	No.	Name	Change Description
Add	1	Column-store partitioned tables support the split operation.	Column-store partitioned tables support the split operation.
	2	CREATE TABLE tbl(LIKE tbl2 INCLUDING DROPCOLUMNS)	The **LIKE DROPCOLUMNS** syntax is supported.

Change Type	No.	Name	Change Description
	3	ALTER TABLE tbl DISTRIBUTE BY	Added the syntax for modifying the distribution column of a table.
	4	ALTER TABLE set version	You can switch between the old and new formats of column-store tables.
	5	CREATE/ALTER FOREIGN TABLE Write Only	Data can be written in OBS ORC format.
	6	CREATE/ALTER FOREIGN TABLE Write Only	Data can be written in OBS HDFS format.
	7	XML	The XML data type can be used in tables and functions.
	8	ALTER VIEW	alter view viewname REBUILD;
Modify	9	VACUUM FULL	A deadlock occurs when concurrent VACUUM FULL operations are performed on the same table.
	10	DROP TEXT SEARCH CONFIGURATION	If the text search configuration to be deleted is the current **default_text_search_config**, an error is reported and the deletion fails.
	11	CREATE/ALTER ROLE/ USER AUTHINFO 'ldap'	The syntax for specifying the user authentication type as LDAP is modified to remove the 64 character length limit.
	12	CREATE USER/ROLE with authinfo 'ldap' password disable	Modified the syntax for creating an LDAP user.
	13	ALTER USER/ROLE with authinfo 'ldap' password disable	Modified the syntax for creating an LDAP user.
Delete	14	ALTER TABLE*Partitioned table*exchange with unlogg*Table*	Forbid partitioned table exchange with unlogged tables.
	15	Inheritance can be modified using **ALTER TABLE ... inherit**.	The optimizer is not adapted to table inheritance. The alter table inherit interface needs to be disabled.

Keywords

Table 1-61 Keywords

Change Type	No.	Name	Change Description
Add	1	xmltable	The **COL_NAME_KEYWORD** keyword has been adapted for inspection.
	2	xmlnamespaces	The **COL_NAME_KEYWORD** keyword has been adapted for inspection.

System Catalogs

Table 1-62 System catalogs

Change Type	No.	Name	Change Description
Add	1	dbms_om.gs_wlm_session_info	New system catalog
	2	dbms_om.gs_wlm_operator_info	New system catalog
Modify	3	pg_jobs	Same as the old system catalog **pg_job**, the metadata relationship is associated using OIDs. If the shared system catalog is full, and you cannot add a transaction after deleting one, **pg_job** will be renamed as **pg_job_proc**.
	4	pg_proc	The textanycat and anytextcat functions are changed to non-STRICT functions.
	5	pg_proc	The database_to_xml, database_to_xmlschema, and database_to_xml_and_xmlschema functions are changed to non-STRICT functions.
	6	pg_catalog.gs_wlm_session_info	It is changed to a view.

Change Type	No.	Name	Change Description
	7	pg_catalog.gs_wlm_operator_info	1. It is changed to a view. 2. The **nodename** column is added.
	8	pg_catalog.gs_wlm_ec_operator_info	It is changed to a view.
	9	pg_catalog.gs_wlm_ec_operator_info	1. It is changed to a view. 2. The **nodename**, **plan_node_name**, **ec_operator**, and **ec_fetch_count** columns are added.

System Functions

Table 1-63 System functions

Change Type	No.	Name	Change Description
Add	1	pgxc_get_residualfiles	Unified CN query function of pg_get_residualfiles.
	2	pgxc_rm_residualfiles	Unified CN query function of pg_rm_residualfiles.
	3	pgxc_verify_residualfiles	Unified CN query function of pg_verify_residualfiles.
	4	query_to_xmlschema	It maps the query content to an XML schema document.
	5	query_to_xml	Maps the query result to an XML file.
	6	query_to_xml_and_xmlschema	It maps the contents of a query to XML values and schema documents.
	7	table_to_xmlschema	It maps the contents of a relational table to an XML schema document.
	8	table_to_xml	It maps the contents of the relation table to XML values.
	9	table_to_xml_and_xmlschema	It maps the contents of a relational table to XML values and schema documents.

Change Type	No.	Name	Change Description
	10	cursor_to_xmlschema	It maps the contents of a cursor query to an XML schema document.
	11	cursor_to_xml	It maps the contents of a cursor query to an XML document.
	12	schema_to_xmlschema	It maps the contents of the entire schema into an XML schema document.
	13	schema_to_xml	It maps the contents of the entire schema to an XML document.
	14	schema_to_xml_and_xmlschema	It maps the contents of the entire schema to XML values and schema documents.
	15	database_to_xmlschema	It maps the contents of the entire database into XML schema documents.
	16	database_to_xml	It maps the contents of the entire database to XML documents.
	17	database_to_xml_and_xmlschema	It maps the contents of the entire database to XML values and schema documents.
	18	xmlpi	It creates an XML processing instruction.
	19	xmlcomment	It creates an XML comment with the specified text as the content.
	20	xmlserialize	It generates a string from a value of the xml type.
	21	xmlparse	It generates a value of the xml type from character data.
	22	xpath	It returns an array of XML values corresponding to the node set generated by the XPath expression.
	23	get_instr_unique_sql_remote_cns	It obtains the unique sql data on all CNs except the current CN.

Change Type	No.	Name	Change Description
	24	xml_is_well_formed	It checks whether the text string is a well-formed XML value.
	25	xml_is_well_formed_content	It checks whether the text string is well-formed.
	26	xml_is_well_formed_document	It checks whether the text string is well-formed.
	27	xmlconcat	It concatenates a list of XML values into a single value.
	28	xmlagg	It combines a function that joins input values together.
	29	IS DOCUMENT	It determines whether the parameter value is a correct XML file.
	30	IS NOT DOCUMENT	It determines whether the parameter value is an incorrect XML file.
	31	xmlexists	It determines whether the XPath expression in the parameter returns any node.
	32	xpath_exists	It determines whether the XPath expression in the parameter returns any node.
	33	xmlelement	It produces an XML element with the given name, attributes, and content.
	34	xmlforest	It produces an XML forest (sequence) of an element using the given name and content.
	35	pg_xlog_replay_completion	You can query the Xlog redo progress of the current DN.
	36	pg_data_sync_from_dummy_completion	It displays the progress of transferring the current DN data page from the dummystandby.
	37	pg_stat_xlog_space	It displays the Xlog space usage on the current DN.
	38	pgxc_stat_xlog_space	It displays the Xlog space usage on all primary DNs.

Change Type	No.	Name	Change Description
	39	xmlroot	It modifies the attributes of the root node of an XML value.
	40	pg_get_residualfiles	It queries the list of all residual files on the current node.
	41	pg_get_running_jobs	It queries jobs that are running on the current node.
	42	pg_is_residualfiles	It can be used to check whether a specified **relfilenode** is a residual file in the current database.
	43	pg_rm_residualfiles	It deletes files from a specified residual file list.
	44	pg_verify_residualfiles	It can verify if the list of remaining files recorded in a specific file are actually residual files.
	45	table_distribution	It adds a **table_distribution** function whose parameter type is regclass.
	46	XMLTABLE	The **XMLTABLE** function can be used to parse XML data.
Modify	47	pgxc_get_senders_catchup_time	The **catchup** process information field is added.
	48	pg_stat_get_data_senders	The **catchup** process information field is added.
	49	pgxc_get_thread_wait_status	The **num_node_display** parameter is added.
	50	pg_stat_get_status	The **num_node_display** parameter is added.
	51	pg_catalog.pgxc_get_wlm_session_info_bytime	The first parameter is changed from case-sensitive to case-insensitive.

System Views

Table 1-64 System views

Change Type	No.	Name	Change Description
Add	1	pg_job_single	Job information of a single node.
	2	pg_job	Used for forward compatibility with the pg_job system catalog and distributed pg_job_single collection.
	3	pgxc_get_stat_all_partitions	Obtain the number of insert/update/delete/live/dead tuples and dirty page rate of each partition in all partitioned tables.
	4	gs_view_dependency	The dependency of the cascading query view is the union of the preceding two functions.
Modify	5	pgxc_get_instr_unique_sql	Run the get_instr_unique_sql_remote_cns statement to obtain the unique SQL data on other CNs.
	6	pg_stat_replication	Debugged the function.
	7	pg_get_senders_catchup_time	The catchup process information field is added.
	8	pg_catalog.gs_wlm_operator_history	The nodename field is added.
	9	pg_catalog.pgxc_wlm_operator_history	The nodename field is added.
	10	pg_catalog.pgxc_wlm_operator_info	The nodename field is added.

Behavior Changes

Table 1-65 Behavior changes

Change Type	No.	Name	Change Description
Modify	1	CN retry supports the stored procedure that affects real-time printing.	The output of the stored procedure is printed in real time. If cn retry occurs, a notice is displayed, indicating that the output message is repeated. If cn retry occurs before printing, no notice is output.

GUC Parameters

Table 1-66 GUC parameters

Change Type	No.	Name	Change Description
Add	1	wal_compression_level	Zlib compression level of the PFI log compression function. The default value is **9**.
	2	wal_compression	Whether to enable the PFI log compression function. This function is disabled by default.
	3	max_xlog_backup_size	Size of the Xlog backup file. When the size of a backup Xlog file exceeds the value of this parameter, the earliest backup Xlog file is automatically deleted until the size of the backup Xlog file is less than 90% of the value of this parameter.
	4	max_cache_partition_num	Maximum number of partitions reserved in the memory during redistribution. If the number of partitions exceeds the value of this parameter, the earliest partition will be written to disks in CU format.

Change Type	No.	Name	Change Description
	5	password_encryption_type	Encryption type of user passwords. The value **2** is added, indicating that the password is encrypted using SHA256. If the current cluster is upgraded from 8.0.0 or an earlier version, the default value of this parameter is **1** for forward compatibility. The default value is **2** for a newly installed cluster.
	6	join_num_distinct	Default **distinct** value of the **join** column. The default value is **-20**.
	7	cost_model_version	Whether the cost estimation optimization takes effect. The default value is **1**, indicating that the optimization takes effect.
	8	qual_num_distinct	Default **distinct** value of the filter column. The default value is **200**.
	9	behavior_compat_options	The varray_verification option is added to roll back the new verification.
	10	behavior_compat_options	The **check_function_conflicts** option is added to check whether the **IMMUTABLE** function has non-IMMUTABLE behavior.
	11	auto_process_residualfile	Switch for automatically recording residual files. The default value is **true**, indicating that the function is enabled.
	12	default_colversion	Default format for creating a column-store table. The default format is **1.0**.
	13	enable_partition_dynamic_pruning	Whether to support dynamic pruning during partition table scanning. Dynamic pruning is enabled by default.

Change Type	No.	Name	Change Description
	14	enable_join_pseudoconst	Whether to generate a join expression on the join expression that is equal to a constant. Similar to the t1 inner join t2 on t1.a=t2.a where t1.a = 1 scenario, the join expression can be generated based on t1.a=t2.a. (In the earlier version, such join conditions cannot be used for join cond.)
	15	view_independent	Whether to enable the view dependency decoupling function. Default value: **off**
	16	enable_view_update	Whether to enable single view update.
Modify	17	enable_index_nestloop	If the system is upgraded from C80 or an earlier version to the latest patch version, set **enable_index_nestloop** to **off**. Enabling the upgrade of **enable_index_nestloop** from 6.5.0 or 6.5.0 will maintain forward compatibility. The default value of **enable_index_nestloop** is **on**.
	18	archive_mode	The default value of the xlog archive switch is changed from off to on. In some scenarios, the performance is slightly affected. In POC scenarios such as performance comparison, you are advised to manually avoid this problem.
	19	cost_param	The default value is changed to **16**, which corresponds to the optimized cost estimation.
	20	rewrite_rule	The partialpush option is removed.

Change Type	No.	Name	Change Description
	21	behavior_compat_options	Added the strict_concat_functions parameter for forward compatibility of the textanycat and anytextcat functions.
	22	behavior_compat_options	The 'strict_text_concat_td' option has been added to change textcat, textanycat, and anytextcat functions to strict functions in TD mode.
	23	behavior_compat_options	The strict_text_concat_td option is added to be compatible with the null stitching behavior in T/D mode.
	24	behavior_compat_options	The bpchar_text_without_rtrim option is added to be compatible with the style of processing spaces at the end of the bpchar string in TD mode.

Time Zone

Table 1-67 Time zone

Change Type	No.	Name	Change Description
Modify	1	Asia/Beijing: The definition of the Beijing time zone is changed to be the same as that of Asia/Shanghai.	1. The time zone offset before 1901 is changed from 08:00:00 to 08:05:43. 2. The DST rules from 1940 to 1949 are added. • The DST is used from June 1 to October 12, 1940. • The DST is used from March 15, 1941 to November 1, 1941. • The daylight saving time (DST) was used from January 31, 1942 to September 1, 1945. The DST was used in the three years because the government did not adjust the time zone rules nationwide during World War II. • The DST is used from May 15, 1946 to September 31, 1946. • The DST is used from April 15, 1947 to October 31, 1947. • The DST is used from May 1, 1948 to September 30, 1948. • DST is used from May 1, 1949 to September 30, 1949. 3. DST rules were also adjusted between 1986 and 1991. • From 1986 to 1991, the DST was changed from 00:00 to 02:00. • From 1987 to 1991, the DST was adjusted to the

Change Type	No.	Name	Change Description
			standard time zone, and the first Sunday after September 10 was changed to the first Sunday after September 11.

Change Type	No.	Name	Change Description
	2	Modified the definition of Asia/Shanghai.	1. The time zone offset before 1901 is changed from 08:05:57 to 08:05:43. 2. Adjust the DST rules from 1940 to 1941. • The DST time in 1940 is changed from June 3 to October 1 to June 1 to October 12. • The DST time in 1941 is changed from March 16 to October 1 to March 15 to November 1. 3. The DST rules from 1942 to 1949 are added. • Daylight saving time (DST) was used from January 31, 1942 to September 1, 1945, as the government did not modify the time zone rules during World War II. • The DST is used from May 15, 1946 to September 31, 1946. • The DST is used from April 15, 1947 to October 31, 1947. • The DST is used from May 1, 1948 to September 30, 1948. • DST is used from May 1, 1949 to September 30, 1949. 4. DST rules were also adjusted between 1986 and 1991. • From 1986 to 1991, the DST was changed from 00:00 to 02:00. • From 1987 to 1991, the DST was adjusted to the standard time zone, and the first Sunday after

Change Type	No.	Name	Change Description
			September 10 was changed to the first Sunday after September 11.
	3	Asia/Urumqi	After May 1980, the time zone offset is changed from GMT+8 to GMT+6.
	4	The five time zones defined based on the physical time zone are changed to two time zones.	1. The time zone names Asia/Harbin (long white time zone), Asia/Chongqing (longshu time zone), and Asia/Shanghai (original standard time zone) are reserved. All the time zone names point to the newly defined Asia/Shanghai (standard time zone of China). It is used in the whole country. 2. The Asia/Kashgar (Kunlun time zone) and Asia/Urumqi (New Tibet time zone) time zone names are reserved, and the definitions point to the newly defined Asia/Urumqi (Urumqi time zone). Two time zone definitions are used in the local area.
	5	The initial default time zone may change.	During database initialization, a time zone that best matches the operating system time zone is used as the default time zone. The time zone definition supported by the database is changed. As a result, the initial default time zone may be changed.

1.10 Version 8.0.1

V8.0.1.500 release date: February 8, 2021

V8.0.1.100 release date: July 31, 2020

SQL Syntax

Table 1-68 SQL syntax

Change Type	No.	Name	Change Description
Add	1	Setting the temporary tablespace quota	The CREATE USER/ROLE user1... TEMP SPACE '5GB' syntax is added to set the temporary space quota.
	2	Changing the temporary tablespace quota	The ALTER USER user1 TEMP SPACE '5GB'; syntax is added to modify the temporary space quota.
	3	Setting the space limit for the intermediate calculation result set to be flushed to disks	The CREATE USER/ROLE user1... SPILL SPACE'unlimited; syntax is added to set the temporary space quota of the intermediate result set.
	4	Modifying the upper limit of the space for storing intermediate calculation results	The ALTER USER user1 SPILL SPACE '5GB'; syntax is added to set the temporary space quota for the intermediate result set.
	5	ALTER TABLE	The alter table modify not null [enable] / alter table modify null syntax is added.
	6	CREATE INDEX	The index name can contain schema when an index is created.
	7	ALTER SEQUENCE	The ALTER SEQUENCE syntax is added.
	8	CREATE SYNONYM	The CREATE [OR REPLACE] SYNONYM syn_name FOR obj_name syntax is added.
	9	DROP SYNONYM	The DROP SYNONYM IF EXISTS syn_name syntax is added.
	10	ALTER SYNONYM	The ALTER SYNONYM syn_name OWNER TO newowner syntax is added.
	11	CREATE USER/ROLE with ldap password disable	AD authentication is supported. When creating a user, set the user authentication type to LDAP.

Change Type	No.	Name	Change Description
	12	ALTER USER/ROLEwith ldap password disable	AD authentication is supported, and the user authentication type is changed to LDAP.
	13	Adding parameter options to the copy fault tolerance table	The log errors data option is added to support user-defined selection. Fill in the rawrecord field.
Modify	14	GIN index	The GIN index is disabled in the 300 scenario.
Delete	15	ALTER TABLE *Partitioned table*exchange with unlogg *Table*	Do not exchange partitioned tables with unlogged tables.

Keywords

Table 1-69 Keywords

Change Type	No.	Name	Change Description
Add	1	SYNONYM	The non-reserved keyword SYNONYM is added.
	2	VERIFY	The reserved keyword VERIFY is added.
	3	FAST	The non-reserved keyword FAST is added.
	4	COMPLETE	The non-reserved keyword COMPLETE was added.
Modify	5	FUNCTION	The keyword FUNCTION is changed to a non-reserved keyword U.
	6	OVER	The keyword OVER is changed to a non-reserved keyword U.
	7	SPLIT	The SPLIT keyword is changed to a non-reserved keyword U.
	8	RETURN	The type of keyword RETURN is changed to a non-reserved keyword U.

Change Type	No.	Name	Change Description
	9	ISNULL	The ISNULL type is changed to a non-reserved keyword U.

System Catalogs

Table 1-70 System catalogs

Change Type	No.	Name	Change Description
Add	1	PG_SYNONYM	Added the pg_synonym system catalog for storing the mapping between synonym objects and associated object names.
	2	PG_REDACTION_POLICY PG_REDACTION_COLUMN	pg_redaction_policy stores object masking policies and status expressions. pg_redaction_column stores object column information, including which columns need to be anonymized and the anonymization mode.
	3	PG_OBJECT	Stores the creation time, creation user, and last modification time of a table/foreign table/view/index/sequence/function.
Modify	4	PG_AUTHID	The authinfo column is added to identify the authentication type of a user. The default value is empty. If the user is an AD user, the value is ldap.

System Functions

Table 1-71 System functions

Change Type	No.	Name	Change Description
Add	1	dbms_redact.add_policy dbms_redact.alter_policy dbms_redact.enable_policy dbms_redact.disable_policy dbms_redact.drop_policy	• Add a data masking policy. • Modify the masking policy information, including adding a masking column to a table object and modifying the masking mode of the column. • Enable or disable a data masking policy. • Deleting a data masking policy
	2	pgxc_get_workload_sql_count	Provides statistics on the number of **SELECT**/**UPDATE**/**INSERT**/**DELETE** statements executed in all workload Cgroup on all CNs of the current cluster and the number of DDL, DML, and DCL statements.
	3	pgxc_get_workload_sql_elapse_time	Provides statistics on response time of **SELECT**/**UPDATE**/**INSERT**/**DELETE** statements executed in all workload Cgroup on all CNs of the current cluster.
	4	get_instr_unique_sql	Provides unique SQL information collected on the current node.
	5	reset_instr_unique_sql(cstring, cstring, INT8)	Deletes the collected Unique SQL information.
	6	pgxc_get_instr_unique_sql	Provides complete Unique SQL information collected on all CNs in a cluster.
	7	pg_check_authid	Checks whether the user exists based on the user OID.

Change Type	No.	Name	Change Description
Modify	8	pg_stat_get_sql_count	Provides statistics on the results of **SELECT**/**UPDATE**/**INSERT**/**DELETE**/**MERGE INTO** statements executed by all users on all the nodes, their response time, and the number of DDL, DML, and DCL statements.
	9	pgxc_get_sql_count	Provides statistics on the results of **SELECT**/**UPDATE**/**INSERT**/**DELETE**/**MERGE INTO** statements executed by all users on the current node, their response time, and the number of DDL, DML, and DCL statements.

System Views

Table 1-72 System views

Change Type	No.	Name	Change Description
Add	1	ALL_SYNONYMS DBA_SYNONYMS USER_SYNONYMS	• ALL_SYNONYMS displays all synonyms accessible to the current user. • DBA_SYNONYMS displays all synonyms in the database. Only users with system administrator permissions can access this view. • USER_SYNONYMS displays synonyms accessible to the current user.
	2	REDACTION_POLICIES REDACTION_COLUMNS	The two tables correspond to two system catalogs. The OID is converted to the name to display the policy information.

Change Type	No.	Name	Change Description
	3	GS_WORKLOAD_SQL_COUNT	Displays statistics on the number of SQL statements executed in workload Cgroups on the current node, including the number of **SELECT**, **UPDATE**, **INSERT**, and **DELETE** statements and the number of DDL, DML, and DCL statements.
	4	PGXC_WORKLOAD_SQL_COUNT	Displays statistics on the number of SQL statements executed in workload Cgroups on all CNs in a cluster, including the number of **SELECT**, **UPDATE**, **INSERT**, and **DELETE** statements and the number of DDL, DML, and DCL statements.
	5	GLOBAL_WORKLOAD_SQL_COUNT	Displays statistics on the number of SQL statements executed in all workload Cgroups in a cluster, including the number of **SELECT**, **UPDATE**, **INSERT**, and **DELETE** statements and the number of DDL, DML, and DCL statements.
	6	GS_WORKLOAD_SQL_ELAPSE_TIME	Displays statistics on the response time of SQL statements in workload Cgroups on the current node, including the maximum, minimum, average, and total response time of **SELECT**, **UPDATE**, **INSERT**, and **DELETE** statements. The unit is microsecond.
	7	PGXC_WORKLOAD_SQL_ELAPSE_TIME	Displays statistics on the response time of SQL statements in workload Cgroups on all CNs in a cluster, including the maximum, minimum, average, and total response time of **SELECT**, **UPDATE**, **INSERT**, and **DELETE** statements. The unit is microsecond.

Change Type	No.	Name	Change Description
	8	GLOBAL_WORKLOAD_SQL_ELAPSE_TIME	Displays statistics on the response time of SQL statements in all workload Cgroups in a cluster, including the maximum, minimum, average, and total response time of **SELECT**, **UPDATE**, **INSERT**, and **DELETE** statements. The unit is microsecond.
	9	PGXC_INSTR_UNIQUE_SQL	Displays the complete Unique SQL statistics of all CN nodes in the cluster.
	10	GS_INSTR_UNIQUE_SQL	You can query the Unique SQL execution information collected by the current node.
Modify	11	ALL_OBJECTS DBA_OBJECTS USER_OBJECTS	Modify the view definition and extend the SYNONYM object.
	12	PGXC_THREAD_WAIT_STATUS	Modify the view definition.
	13	GS_SQL_COUNT	Provides statistics on the number of **SELECT**/**UPDATE**/**INSERT**/**DELETE**/**MERGE INTO** statements executed by all users on the current node, response time, and the number of DDL, DML, and DCL statements.
	14	PGXC_SQL_COUNT	Provides statistics on the number of **SELECT**/**UPDATE**/**INSERT**/**DELETE**/**MERGE INTO** statements executed by all users in the cluster, response time, and the number of DDL, DML, and DCL statements.
	15	ALL_OBJECTS DBA_OBJECTS USER_OBJECTS	Modify the view definition and add the creation time and last modification time of the object.

Change Type	No.	Name	Change Description
	16	PG_TABLES	Modify the view definition and add the creation user, creation time, and last modification time of the table.

Behavior Changes

Table 1-73 Behavior changes

Change Type	No.	Name	Change Description
Add	1	CN Retry Disables Some Copy FROM Scenarios	• The COPY FROM STDIN statement is not supported. • The gsql \copy from meta-command is not supported. • JDBC CopyManager copyIn cannot be used to import data.

GUC Parameters

Table 1-74 GUC Parameters

Change Type	No.	Name	Change Description
Add	1	sql_use_spacelimit	Specifies the maximum storage space for a single SQL statement on a single DN.
	2	remote_read_mode	Sets the remote read mode.
	3	enable_incremental_catchup	Sets whether to enable incremental catchup.
	4	wait_dummy_time	Maximum time for waiting for the secondary server to start or send back the scan list during incremental catchup.
	5	enable_xlog_group_insert	It specifies whether the xlog enables the group insertion mode on the ARM platform.

Change Type	No.	Name	Change Description
	6	topsql_retention_time	Number of days for storing data in the gs_wlm_session_info and gs_wlm_operator_info tables.
	7	user_metric_retention_time	Indicates the number of days for storing historical resource monitoring data.
	8	instance_metric_retention_time	Indicates the number of days for storing historical resource monitoring data of an instance.
	9	enable_user_metric_persistent	Indicates whether to enable the function of monitoring and dumping historical user resources.
	10	enable_instance_metric_persistent	Indicates whether to enable the function of dumping historical instance resource monitoring data.
	11	COMM_IPC	Switch for locating communication performance problems. This parameter specifies whether to print the information about the packets sent and received by each communication node.
	12	COMM_PARAM	Communication performance problem locating switch. This parameter specifies whether to print the session parameter settings during node communication.
	13	plan_cache_mode	Controls the behavior of the cached plan. The custom plan, generic plan, or auto mode can be forcibly used (the default value is auto).
Modify	14	audit_system_object	The 18th digit is added to indicate whether to audit SYNONYM objects. The maximum value is changed to 524287.

Change Type	No.	Name	Change Description
	15	comm_max_stream	In the FusionCube environment, the default value of this parameter is changed to 10000 after the cluster is installed.
	16	enable_dynamic_workload	The default value is changed from off to on.
	17	behavior_compat_options	The convert_interval_to_text member is added to control whether the implicit type conversion from interval to text is supported. By default, the implicit type conversion is not supported.

1.11 Deleted Database Interfaces

To facilitate the ongoing development and maintenance of GaussDB(DWS), a metadata sunset feature is implemented to phase out outdated GUC parameters, system functions, and system views. For the sake of preserving historical compatibility, deprecated functions and views are archived within the user-created SUNSET EXTENSION, ensuring that they do not interfere with future version upgrades.

CREATE EXTENSION SUNSET

When there are attempts to access deprecated system functions and views, an error will be displayed, indicating their non-existence. GaussDB(DWS) facilitates the creation of these legacy system functions and views using the CREATE EXTENSION SUNSET statement. This extension encapsulates the sunset functions and views, preserving their original definitions prior to sunset.

Here is how to create it:

The administrator executes the **CREATE EXTENSION SUNSET** command to establish the extension. This extension needs to be created only once by the user. The access permissions for the sunset functions and views remain consistent with those from the pre-sunset version.

```
CREATE EXTENSION sunset;
CREATE EXTENSION

 \dx+ sunset
                                                  Objects in extension "sunset"
                                                       Object
Description
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------
```



```
 function pg_delete_audit(timestamp with time zone,timestamp with time zone)
 function pg_log_comm_status()
 function pg_stat_get_realtime_info_internal(oid,oid,bigint,cstring,oid)
 function pg_stat_get_wlm_session_info_internal(oid,oid,bigint,oid)
 function pg_stat_get_wlm_session_info(oid)
 function pg_stat_get_wlm_session_iostat_info(integer)
 function pg_stat_get_wlm_statistics(integer)
 function pg_sync_cstore_delta()
 function pg_sync_cstore_delta(text)
 function pg_user_iostat(cstring)
 function pgxc_log_comm_status()
 function pgxc_pool_check()
 function signal_backend(bigint,integer)
 function update_pgjob(bigint,"char",bigint,timestamp without time zone,timestamp without time
zone,timestamp without time zone,timestamp without time zone,timestamp without time zone,smallint)
 view gs_wlm_session_info_all
 view pg_session_iostat
 view pg_wlm_statistics
(17 rows)
```

Deleted Database Interfaces of the 8.3.0 Cluster Version

Table 1-75 Sunset system functions and views

Type	Item	Description
Functions	pg_sync_cstore_delta(text)	Merges delta table data to the primary table in column-store mode. This function has been deprecated. It is replaced by the **VACUUM DELTAMERGE xxx** syntax.
	pg_sync_cstore_delta	Merges delta table data to the primary table in column-store mode. This function has been deprecated. It is replaced by the **VACUUM DELTAMERGE xxx** syntax.
	pgxc_pool_check	Checks whether the connection data cached in the connection pool is consistent with **pgxc_node**. This function has been deprecated.
	pg_delete_audit	Deletes audit logs. This function has been deprecated. Manual deletion of audit logs is not allowed.
	pg_log_comm_status	Indicates the log recording status of the internal structure of the SCTP communication layer on the current node. This function has been deprecated.
	pgxc_log_comm_status	Indicates the log recording status of the internal structure of the SCTP communication layer on all nodes. This function has been deprecated.
	signal_backend	This function has been deprecated.

Type	Item	Description
	pg_stat_get_realtime_info_internal	Obtains real-time top SQL information at the query level. This function has been deprecated. It is replaced by the **PGXC_WLM_SESSION_STATISTICS** view.
	pg_stat_get_wlm_session_info_internal	Obtains top SQL information at the query level. This function has been deprecated. It is replaced by the **pg_stat_get_wlm_realtime_session_info** function.
	pg_stat_get_wlm_session_info	Obtains session information. This function has been deprecated. It is replaced by the **PGXC_WLM_SESSION_INFO** view.
	pg_stat_get_wlm_statistics	Obtains session information. This function has been deprecated. It is replaced by the **PGXC_WLM_SESSION_INFO** view.
	pg_user_iostat	Obtains the I/O information of a user. This function has been deprecated. It is replaced by the **PGXC_TOTAL_USER_RESOURCE_INFO** view.
	pg_stat_get_wlm_session_iostat_info	Obtains the I/O information of a query. This function has been deprecated. It is replaced by the **PGXC_WLM_SESSION_STATISTICS** or **PGXC_WLM_SESSION_INFO** view.
System views	gs_wlm_session_info_all	Obtains session information. This view has been deprecated. It is replaced by the **PGXC_WLM_SESSION_INFO** view.
	pg_wlm_statistics	Displays load management information after a job is complete or an exception has been handled. This view has been deprecated. It is replaced by the **PGXC_WLM_SESSION_INFO** view.
	pg_session_iostat	Obtains the I/O information of a session. This view has been deprecated. It is replaced by the **PGXC_WLM_SESSION_STATISTICS** view.

Deleted Database Interfaces of the 9.1.0 Cluster Version

Table 1-76 Sunset database interfaces

Type	Item	Description
Extension	file_fdw	Local file access, which is deprecated.

Type	Item	Description
External server	gsmpp_errorinfo_server	This interface is used for reading data from the import error table, but it has been discarded. Accessing the import error table no longer requires **gsmpp_errorinfo_server**.
Logical replication	pg_create_logical_replication_slot	Creates a logical replication slot. This has been discarded in 9.1.0.200. The logical replication feature has been discarded.
Logical replication	pg_logical_slot_peek_changes	Decodes changes without pushing them to the replication slot. This has been discarded in 9.1.0.200. The logical replication feature has been discarded.
Logical replication	pg_logical_slot_get_changes	Decodes and pushes changes to the stream replication slot. This has been deprecated in 9.1.0.200. The logical replication feature has been discarded.

2 Version Support Notes

The cluster version of GaussDB(DWS) displayed on the management console is in the format of x.y.z.p, which is explained in **Figure 2-1**.

Generally, new functions and features of GaussDB(DWS) are released by release number, for example, 8.1.3 and 8.2.0.

After each iteration version is released, a patch is released to fix problems, for example, patch 8.1.3.322. A patch only fixes problems and does not add new functions or features.

Figure 2-1 GaussDB(DWS) version description

Version Lifecycle

Table 2-1 provides the lifecycle of GaussDB(DWS) versions (release number), helping you plan your version update.

For versions that have reached EOS or are not **Recommended Version**, you are advised to upgrade them to the latest stable version by referring to **Version Upgrade Policies**.

Table 2-1 GaussDB(DWS) version lifecycle

Version	Running status	Released On	EOM Date (End of Marketing)	EOS Date (End of Service)
9.0.x	Released	2023-09-30	2025-06-30	2026-06-30
8.3.0.x	Released	2024-03-30	2026-03-30	2027-03-30
8.2.1.x	Released	2023-05	2025-09-30	2027-03-30
8.2.0.x	Released	2022-11-30	2025-09-30	2027-03-30
8.1.3.x	Released	2022-04-15	2024-06-30	2025-12-31
8.1.1.x	EOM	2021-07-30	2023-06-30	2025-12-31
8.1.0.x	EOM	2021-05-15	2023-06-30	2025-12-31
8.0.1.x	EOM	2020-07-31	2023-06-30	2024-12-31

Version Upgrade Policies

Table 2-2 Version incorporation

Source Version	EOS	Suggestion	Upgrade Plan
1.5.x/1.6.x	Yes	Direct upgrade is not supported. You are advised to migrate data to **8.1.3.336** (recommended version).	You are advised to contact technical support engineers for evaluation before performing this operation.
1.7.x	Yes	You are advised to upgrade the version to **8.1.3.336** (recommended version).	You cannot directly upgrade to version 8.1.3.336 from your current version. Instead, a multi-hop upgrade is required. Contact technical support for evaluation before the upgrade.

Source Version	EOS	Suggestion	Upgrade Plan
8.0.1.x	No	You are advised to upgrade the version to **8.1.3.336** (recommended version).	You cannot directly upgrade to version 8.1.3.336 from your current version. Instead, a multi-hop upgrade is required. Contact technical support for evaluation before the upgrade.
8.1.0.x	No	You are advised to upgrade the version to **8.1.3.336** (recommended version).	You cannot directly upgrade to version 8.1.3.336 from your current version. Instead, a multi-hop upgrade is required. Contact technical support for evaluation before the upgrade.
8.1.1.x~8.1.3.323	No	You are advised to upgrade the version to **8.1.3.336** (recommended version).	It can be directly upgraded to 8.1.3.336. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.
8.2.0.x (whitelist user version)	No	You are advised to upgrade the version to **8.3.0.110**.	It can be directly upgraded to 8.3.0.110. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.

Source Version	EOS	Suggestion	Upgrade Plan
8.2.1.1xx (whitelist user version)	No	You are advised to upgrade the version to **8.3.0.110**.	It can be directly upgraded to 8.3.0.110. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.
8.2.1.2xx (whitelist user version)	No	You are advised to upgrade the version to **8.3.0.110**.	It can be directly upgraded to 8.3.0.110. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.
8.2.1.3xx (whitelisted user version)	No	You are advised to upgrade the version to **8.3.0.110**.	It can be directly upgraded to 8.3.0.110. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.
8.3.0.x	No	You are advised to upgrade the version to **8.3.0.110**.	It can be directly upgraded to 8.3.0.110. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.

Source Version	EOS	Suggestion	Upgrade Plan
9.0.1/9.0.2	No	You are advised to upgrade the version to **9.1.0.105**.	You cannot directly upgrade to version **9.1.0.105** from your current version. Instead, upgrade to 9.0.3 and then to 9.1.0.105. You can upgrade the cluster on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.
9.0.3/9.1.0.x	No	You are advised to upgrade the version to **9.1.0.105**.	It can be directly upgraded to 9.1.0.105. You are advised to contact technical support for evaluation before the upgrade. It can be upgraded on the console. For details, see **Upgrading a GaussDB(DWS) Cluster**.

