Data Warehouse Service

Product Bulletin

Issue 03

Date 2025-09-19





Copyright © Huawei Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks 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 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.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base

Bantian, Longgang Shenzhen 518129

People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:

https://www.huawei.com/en/psirt/vul-response-process

For vulnerability information, enterprise customers can visit the following web page:

https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Product Notice	1
1.1 Checking the GaussDB(DWS) Version	1
1.2 New Features in 9.1.0.x	4
1.3 Version 8.3.0	23
1.3.1 New Features in 8.3.0.100	23
1.3.2 New Features and Resolved Issues in 8.3.0.x	39
1.4 Version 8.1.3	
1.4.1 New Features in 8.1.3	
1.4.2 New Features and Resolved Issues in 8.1.3.x	71
1.5 Version 8.1.1	84
1.6 Deleted Database Interfaces	104
2 Version Support Notes	108

1 Product Notice

- 1.1 Checking the GaussDB(DWS) Version
- 1.2 New Features in 9.1.0.x
- 1.3 Version 8.3.0
- 1.4 Version 8.1.3
- 1.5 Version 8.1.1
- 1.6 Deleted Database Interfaces

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

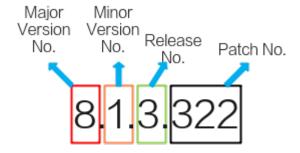


Table 1-1 GaussDB(DWS) cluster version mapping

Released On	Cluster Version	Database Kernel Version	Description
2025-01-17	9.1.0.212	9.1.0	1.2 New
2024-12-13	9.1.0.211		Features in 9.1.0.x
2024-11-25	9.1.0.210		
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	1.3 Version
2024-07-22	8.3.0.108		8.3.0
2024-04-17	8.3.0.100		
2023-11-02	8.1.3.325 (recommended version)	8.1.3	1.4.1 New Features in 8.1.3
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	1.5 Version
2022-04-15	8.1.1.300		8.1.1
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		

Released On	Cluster Version	Database Kernel Version	Description
2021-07-30	8.1.1.100		

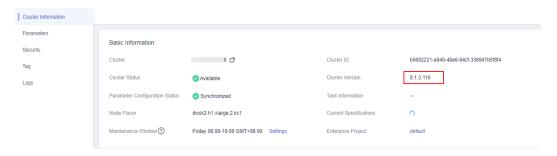
Querying the Cluster Version

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 Table 1-1.
- Method 2: Connect to the GaussDB(DWS) database. Run the following SQL statement:

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 9.1.0.218 (May 2025)

Table 1-2 New features/Resolved issues in patch 9.1.0.218

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
New features	None	-	-	-
Resolved issues	There are write locks in high-concurrency scenarios once LLVM is enabled.	After LLVM is enabled, an operating system (OS) will reclaim memory when an application repeatedly allocates and releases memory in high-concurrency scenarios. During the memory reclamation, holding the mmap write lock can indeed block other threads from allocating new memory, affecting memory access performance.	Versions earlier than 9.1.0.218	Up gra de the vers ion to 9.1. 0.2 18 or late r.
	Hstore and time series tables are no longer used.	The time series tables, column-store delta tables, and historical HStore tables are no longer used in clusters of version 9.1.0.218 or later. They are replaced by the HStore Opt table.	-	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	The error "invalid memory alloc reuqest size" is displayed during presort execution	During the runtime filter presorting, the ID of another column is incorrectly obtained. As a result, the obtained attlen is incorrect, and the function for reading the data length is incorrect. When attlen is set to -1, the system incorrectly uses the single-byte processing function to read the Chinese characters encoded in GBK. As a result, a negative value for character length is calculated, and memory fails to be allocated.	9.1.0.210	
	After enable_hstore_binlog_t able is enabled and services are running for a long time, the pg_csnlog file on the standby node is stacked. After the file is cleared, the space is still not reclaimed.	To prevent unwanted recycling of CSN logs, at least one VACUUM operation is required. During this process, pg_binlog_slots on the primary node collects oldestxmin, but the standby node does not execute the collection. As a result, oldestxmin for reclaiming CSN logs is always 0 and CKP does not reclaim CSN logs.	9.1.0.201	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	The memory estimated by ANALYZE deviates significantly from the actual memory, so CCN is queued abnormally.	If the defined column width exceeds 1024 and the stored data is short or empty, the estimated memory width using ANALYZE is large.	8.1.3	
	When UPSERT operations are executed in batches, the binlogs on the source and target tables are not synchronized.	When UPSERT operations are performed on all columns, the old binary log is deleted and the new binary log is not recorded. The binlogs on the source and target tables are not synchronized.	Versions earlier than 9.1.0.218	
	Fixed the issue where after a switchover is performed, the values of autovacuum and autoanalyze of the original primary cluster are disabled.	The autovacuum parameter of the primary cluster is originally on. After a switchover is performed, the value is changed to off. After the switchover is performed again, the values of autovacuum and autoanaly are not as expected.	9.1.0	
	Fixed the issue where the value of enable_orc_cache is automatically modified in upgrades.	After a cluster is upgraded to 9.1.0.1 or later, the value of enable_orc_cache is changed from on to off.	9.1.0.210	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	The "stream plan check failed" is displayed after the upgrade.	After the value partition plan changes, the value partition plan generated by WindowAgg is inconsistent between the upper node and the lower node.	8.2.1.225	
	After a DR switchover, the error message "xlog flush request" is displayed in the primary cluster.	Only the primary CN in the DR cluster is backed up. The full or incremental backup type is recorded in the metadata file of the backup set. When the DR cluster is restored, the metadata file information is read to determine whether to clear the CN directory (cleared during full restoration). CN IDs in the primary and DR clusters may be different. If the backup type is queried based on the ID, the incorrect information may be read. As a result, necessary files are not cleared during full restoration, and residual files affect services.	9.1.0	
	Turbo problem hardening	Some unconventional scenarios (for example, multiple UNION ALL and inconsistent data types) are not considered. As a result, an exception occurs during a query.	9.1.0	

Patch 9.1.0.215 (March 2025)

Table 1-3 New features/Resolved issues in patch 9.1.0.215

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
New features	None.	-	-	-
Resolved issues	Fixed the problem where the intelligent O&M scheduler triggered data flushes to disks.	The intelligent O&M scheduler sends the pgxc_parallel_query function to each DN to check the table size. The DN query results include auxiliary tables like CUDesc and Delta for column-store tables. Each partition also has its own CUDesc and Delta tables. This makes the final result set large. The DN sends a summary of the query results to the CN. If the result set is too large, this triggers a disk flush. This can slow down the query or fill up the disk.	8.3.0.100	Up gra de the vers ion to 9.1. 0.2 15 or late r.
	Fixed the issue where the storage-compute decoupled V3 table would sometimes crash if DiskCache was turned off.	If the storage-compute decoupled V3 table is used, DiskCache is disabled, and the service thread fails to get space while reading OBS objects, the transaction rolls back. This can cause errors.	9.1.0	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	Fixed the issue where the CM component could not restart the cluster if a node was subhealthy (hung).	 cm_ctl starts a cluster in two phases: 1. Check each node's status by connecting with PSSH. 2. Delete each nodes' start and stop files by connecting with PSSH. The SSH command misreads its parameters. This causes a single faulty node check to take over 300 seconds. If multiple nodes are faulty, the time needed grows quickly. Eventually, the command fails to deliver. 	8.1.3	
	Fixed the issue where, in data lake scenarios, too many concurrent requests (over 3,000) slowed connections between CNs and DNs, resulting in low overall performance.	One file is allocated to only one DN. However, in the execution plan, the CN still establishes connections with all DNs. In data lakes, a single cluster needs to support more than 3,000 concurrent requests, which increases the overhead.	9.1.0	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	Fixed the issue where a long write process field in a data lake sometimes caused the DN process to malfunction.	If the write process field is too long in a data lake, an error occurs. During rollback, the system calls the relevant logic. But some memory is already freed, leading to a function error and shutting down the DN process.	9.1.0	
	Fixed the SQL injection vulnerability (CVE-2025-1094).	Your product may have the PostgreSQL SQL injection vulnerability (CVE-2025-1094).	9.1.0	
	Fixed the issue where query errors sometimes happened in row-column mixed scenarios when using the NestLoop and Stream operators.	The NestLoop operator includes the stream operator. First, run the materialization operator on the inner side. In a mixed rowstore and columnstore setup, the Row Adapter operator appears with the materialized operator. This prevents the materialized operator from running, leading to a hang.	8.3.0.108	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	Fixed the issue where the DN connection was checked during the communication thread's idle time, causing occasional slow database responses.	The communication thread checks all connections to other DNs during idle periods. Each check takes 1 ms. In large clusters with over 100 connections, this can take more than 100 ms, delaying the next packet send. This can cause occasional slowdowns in services that need fast response times.	9.1.0	

Patch 9.1.0.213 (February 2025)

Table 1-4 New features/Resolved issues in patch 9.1.0.213

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
New features	After migrating Hive data to GaussDB(DWS), you can choose whether to automatically convert an empty string to 0 in MySQL compatibility mode.	-	-	-

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
Resolved issues	Temporary tables are not cleared when external tables are involved in INSERT OVERWRITE INTO during JDBC connection.	The access process of the JDBC connection involves multiple phases (parser, bind, and exec). In the parser phase, the SQL statement is rewritten, creating a temporary table for INSERT OVERWRITE. If an external foreign table is used, the rewrite operation is repeated in the exec phase, meaning the temporary table from the parser phase is not deleted. This results in two temporary tables being created, with only the one from the later phase being deleted, causing temporary table residue.	9.1.0.211	Up gra de the vers ion to 9.1. 0.2 13 or late r.
	The service receives a 100% skew alarm. The skew content is OVERWRITE temporary table skew.	When INSERT OVERWRITE is executed or the distribution key is changed, a temporary common table is generated to store the data from the source table. If there is data skew, an alarm is triggered for the temporary table name, leading to a failure in the analysis process.	9.1.0.211	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	Resolved the result set problem caused by data precision alignment in the Turbo engine.	If certain data nodes scan zero rows in the base table, the result set sent back to the CN is NULL with a default of 0 decimal places. When the result set is combined with the sum result returned by DNs containing data, the precision of the aligned data is inaccurately calculated. Consequently, the final sum result is stored as int64 type, leading to an unexpected result set.	9.1.0.212	
	Fixed the issue that the system does not check whether the partition path is empty when an external foreign table is used to check the partition path.	When the external table of the HDFS server checks whether the partition path corresponds to the partition field definition, the system does not check whether the partition path is empty.	9.1.0.211	
	Fixed the issue where the null pointer return value was not handled when using an external foreign table to retrieve partition information from Hive MetaStore.	The partition values obtained from Hive MetaStore are null.	9.1.0.212	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	The time zone conversion result of the convert_tz function does not meet the expectation in a certain scenario.	Compatibility with MySQL is not considered when the convert_tz function is used. As a result, the result is not as expected.	9.1.0.210	
	Fixed the issue that the substr result set was incorrect when LLVM was enabled.	GBK data can be imported into the ASCII code database. When LLVM is utilized, the bottom layer of LLVM does not verify the invocation of substr(a, start_index, len) on GBK data columns. This results in miscalculating the character width of GBK as 4 instead of 2 due to reusing UTF-8 character width logic.	8.1.3.x	
	Resolved the result set problem caused by incorrect string processing when using the inlist-to-hash feature in the Turbo engine.	Changing the character string from attlen16 to attlen-1 in the uniq hash table in the Turbo engine incorrectly employs the strlen interface to determine the string length. In the hstoreopt delta table, if attlen is modified from -1 to a fixed length, preliminary batch conversion is necessary.	9.1.0.212	

Patch 9.1.0.212 (January 2025)

Real-time data warehouse

- 1. Resolved the issue that the result set of the date type query is pushed down in MySQL-compatible mode.
- 2. Fixed the issue that the result set is incorrect when limit is set to null or all.
- 3. Fixed the issue of incorrect statistics resetting, inability to trigger **auto vacuum**, and delayed space reclamation.
- 4. Resolved the deadlock problem related to refreshing materialized views and concurrent DDL operations.
- 5. Resolved the issue where data in the original table was mistakenly deleted when a temporary table was manually removed after an error occurred during the redistribution of a cold or hot table during scale-out.
- 6. Resolved the problem of local disk space increase during cold and hot table scale-out.

Lakehouse

- 1. Supported the special character ';' in the path for a foreign table to access OBS.
- 2. Optimized the task allocation for querying Parquet foreign tables to enhance the disk cache hit ratio.

Backup and restoration

- 1. Fixed the problem of intermediate status files remaining during backup and restoration, which would occupy disk space.
- 2. Resolved the backup failure issue when the elastic VW is present.
- 3. Supported backup and restoration for cold and hot tables, prolonging the backup and restoration time.

Ecosystem compatibility

1. Fixed the issue that the PostGIS plug-in may fail to be created.

O&M improvement

- 1. Resolved the issue that SQL monitoring metrics are incompletely collected.
- 2. Resolved the memory leak problem of the secondary node.
- 3. Fixed the issue that intelligent O&M is not started on time.
- 4. Fixed the problem where the scheduler could not be properly scheduled due to residual data caused by a failed **database drop**.
- 5. Fixed the issue of high communication memory usage during high concurrency.
- 6. Resolved the performance problem caused by residual sequences on the GTM when an exception occurs.

Behavior changes

1. To prevent errors during complex SQL execution, we disabled the **ANALYZE** feature of the predicate column during upgrades or new installations.

- 2. In the previous version, if there was an abnormal network connection to the GTM during a drop table operation in a scenario where the table definition contained a sequence column, a warning would be reported. Although the drop table operation could be successfully executed, the sequences might remain on the GTM. However, in the new version, an error is reported when dropping the table, requiring a retry. GaussDB(DWS) now supports dropping tables within transaction blocks. If a drop table statement is successfully executed but the transaction is rolled back, the sequences will be deleted from the GTM, but the table will still exist on the CN. In this case, the table needs to be dropped again to avoid errors indicating that the sequences do not exist.
- 3. The **truncate** operation can proactively terminate **SELECT** operations in case of lock conflicts, and this feature is disabled by default. In the previous version, if the session executing the **SELECT** statement was terminated, it would cause an error, but the connection would remain open. However, in the new version, the session executing the **SELECT** statement is automatically closed, and the service requires reconnection.

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, improving 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 pq 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

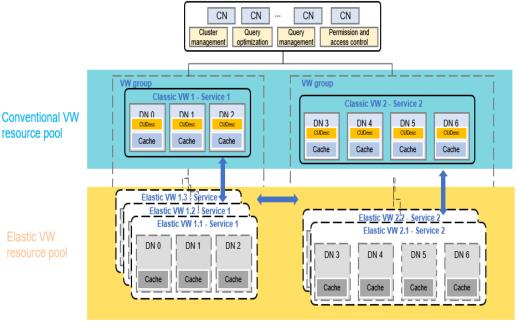
- 1. ORC foreign tables support the ZSTD compression format.
- 2. GIS supports the **st_asmvtgeom**, **st_asmvt**, and **st_squaregrid** functions.

Version 9.1.0.100 (August 12, 2024)

Elastic architecture

- Architecture upgrade: The storage-compute decoupling architecture 3.0, based on OBS, introduces layered and elastic computing and storage, with ondemand storage charging to reduce costs and improve efficiency. Multiple virtual warehouses (VWs) can be deployed to enhance service isolation and resolve resource contention.
- 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. This update significantly improves full-column performance while reducing resource consumption.
- 4. GaussDB(DWS) supports materialized views (currently in beta). For details, see CREATE MATERIALIZED VIEW.
- 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.
- 6. 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**.
- 7. GaussDB(DWS) supports asynchronous sorting to enhance the **min-max** coarse filtering effect of **PCK** columns.
- 8. The performance in the **IN** scenario is greatly improved.
- 9. **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
- 10. GaussDB(DWS) supports large and wide tables, with a maximum of 5,000 columns.
- 11. Create index/reindex supports parallel processing.
- 12. 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]

- 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

- The query filter is enhanced to support interception by SQL feature, type, source, and processed data volume. For details, see CREATE BLOCK RULE.
- 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.
- 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. The keyword **status** is added. Avoid using **status** as a database object name. If changing the column alias to **status** causes a service error, add **AS** to the service statement to fix it. Here is an example:

SELECT c1, min (c2) status, c3 from t1; //Error SQL ERROR: syntax error at or near "status"

SELECT c1, min(c2) AS status, c3 from t1; //SQL statement for avoiding the problem. Add AS.

- 2. **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.
- 3. 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.
- 4. By default, NUMA core binding is enabled and can be turned off dynamically using the **enable_numa_bind** parameter.
- 5. 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.
- 6. 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 8.3.0

1.3.1 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 **1.3.2 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 (OBT users)

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

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

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

New features

Table 1-5 8.3.0 feature changes

Category	Feature	Description	Reference
SQL	Last access time of a table object can be recorded.	 Added the enable_save_dataacce ss_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 	 "GUC Parameters" "Statistics During the Database Running" > "Query and Index Statistics Collector" in the Developer Guide "System Catalogs and Views" > "System Catalogs" > "PGXC_STAT_OBJEC T" in the Developer Guide "Functions and Operators > Statistics Information Functions" in SQL Syntax Reference
Security	OneAccess authenticatio n	 pg_stat_object. 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 'oneaccessClientId=xxxx, domain=xxxx'. 	 "System Catalogs" "PG_AUTHID" in the Developer Guide "DDL Syntax" >
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 Performa nce	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 <i>Developer Guide</i>
СМС	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
НА	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.	"Functions and Operators" > "Statistics Information Functions" in SQL Syntax Reference
		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.	

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() 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 Performa nce	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 Performa nce	Table-level oldestxmin	Added the GUC parameter enable_table_level_oldes txmin 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 <i>Developer Guide</i>

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_detecto r_period to specify whether to enable the distributed deadlock detection function and the detection interval.	"GUC Parameters" > "Lock Management" in the <i>Developer Guide</i>
Real-time queries	Approximate computing: column-store vectorized functions based on the HLL algorithm	Added the approx_count_distinct(co l_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 windowfunnel, 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 <i>Developer Guide</i>
SQL	Replication tables	Added disable_update_returnin g_check to the GUC parameter behavior_compat_option s.	"GUC Parameters" > "Miscellaneous Parameters" in the Developer Guide
Replicabili ty	Optimized the rounding rule of the cast function.	Added options for the GUC parameter behavior_compat_option s: 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_option s.	"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 managem ent	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 warehous e	Batch upsert is optimized for hstore tables.	Added the GUC parameter enable_hstore_keyby_ups ert, which controls the optimization of batch upsert in the hstore table.	"Hybrid Data Warehouse" > "Hybrid Data Warehouse GUC Parameters" in the Developer Guide
Partition managem ent	The automatic partition reducing operation of automatic partition management blocks realtime data import to the database.	Added the pg_partition_manageme nt_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_tr uncate option is added to the GUC parameter behavior_compat_option s 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
Compatib ility	New parameter that controls whether the case when condition can contain functions that return multiple result sets.	Added the unsupported_set_functio n_case option to the GUC parameter behavior_compat_option s. 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
Compatib ility	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
Performa nce	LEFT JOIN estimation (added in 8.3.0.100)	The GUC parameter left_join_estimation_enh ancement 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 enhance ment	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 1-6 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 1-7 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.</partition></common>
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 1-8 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_in fo	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 1-9 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_dist inct	Added an approximate calculation aggregate function.
	8	pg_scan_residualfi les	Scans all residual file records in the database where the current node resides
	9	pgxc_scan_residua lfiles	Scans all nodes for the residual files of the current database
	10	pg_get_scan_resid ualfiles	Obtains all residual file records of the current node.
	11	pgxc_get_scan_res idualfiles	Obtains residual file records on all nodes.
	12	pg_archive_scan_r esidualfiles	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_resid ualfiles_archive	Deletes files from the archived file list on the current node.

Change Type	No.	Item	Description
	15	pgxc_rm_scan_resi dualfiles_archive	Deletes files in the archive directory from all nodes.
	16	pg_partition_man agement_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_partiti on	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_ses sion_info_bytime	Added the parse_time field.
Deleted	21	pg_sync_cstore_de lta(text)	Deprecated the system function.
	22	pg_sync_cstore_de lta	Deprecated the system function.
	23	pgxc_pool_check	Deprecated the system function.
	24	pg_delete_audit	Deprecated the system function.
	25	pg_log_comm_sta tus	Deprecated the system function.
	26	pgxc_log_comm_s tatus	Deprecated the system function.
	27	signal_backend	Deprecated the system function.
	28	pg_stat_get_realti me_info_internal	Deprecated the system function.
	29	pg_stat_get_wlm_ session_info_inter nal	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 1-10 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_statist ics	Added the parse_time field.
	4	pgxc_wlm_session_sta tistics	Added the parse_time field.
	5	gs_wlm_session_histor y	Added the parse_time field.
	6	pgxc_wlm_session_hist ory	Added the parse_time field.
	7	gs_wlm_session_info	Added the parse_time field.
	8	information_schema.t ables	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_st at	Optimized the query performance of equivalence logic rewriting.
Deleted	12	gs_wlm_session_info_a ll	Deprecated the system view.
	13	pg_wlm_statistics	Deprecated the system view.
	14	pg_session_iostat	Deprecated the system view.

GUC Parameters

Table 1-11 GUC parameters

Change Type	No.	Item	Description
Added	1	max_queue_state ments	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_tim e	Maximum days to store pg_job execution results. Default: 30 days.
	3	vector_engine_str ategy	Controls vectorization of the operator. Default: improve (maximize vectorization). Alternate: force (roll back to row storage plan).
	4	enable_release_sc an_lock	Determines if the SELECT statements release the level-1 lock post-execution. It is disabled by default.
	5	job_queue_naptim e	Interval for scheduling task checks and starting task threads. Default: 1 second.
	6	approx_count_dist inct_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_tim e_limit	Sets a threshold for LLVM compilation time as a percentage of executor running time, triggering an alarm if exceeded.
	9	max_opt_sort_row s	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_st andby	Allows CN to connect to a standby DN for O&M operations. This parameter applies only to O&M operations.
	13	enable_stream_sy nc_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_mo de	 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_strateg y	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.

1.3.2 New Features and Resolved Issues in 8.3.0.x

Table 1-12 New features/Resolved issues in version 8.3.0.110

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
New features	By default, the global deadlock detection function is enabled for newly installed clusters. In earlier versions, this is disabled by default.	-	-	-

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.3. 0.1
	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	or late r.
	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	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	When performing a partition exchange operation on redistributed cold and hot tables, the error message "Multitemperature 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	

Table 1-13 New features/Resolved issues in version 8.3.0.108

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.3. 0.1 08 or late r.

Table 1-14 New features/Resolved issues in version 8.3.0.105

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.3. 0.1 05 or late
	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_wo rkers 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	r.
	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	

Table 1-15 New features/Resolved issues in version 8.3.0.103

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.3. 0.1 03 or late r.
	Cstore buffer error during 10 million batch upserts.	cstore_buffers use a clock sweep to clear the cache. The sweep can run up to three times every three seconds. CUs with a usage count above three are not removed. Therefore, you might get an error about cstore_buffers being too small, even if there is free memory.	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	

Table 1-16 New features/Resolved issues in version 8.3.0.101

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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/ 00000000000075'."	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	Up gra de the vers ion to 8.3.
	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	01 or late r.
	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	

1.4 Version 8.1.3

1.4.1 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 **1.4.2 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 1-17 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	READ ONLY HDFS foreign tables in JSON format are supported.	CREATE FOREIGN TABLE (SQL on Hadoop or OBS)
	2. READ ONLY OBS foreign tables in JSON file format are supported.	

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.	GDS-based Cross- Cluster Interconnection
	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 temporaryGDS foreign tablepolicy is improved.	
	2. If the local cluster is the destination cluster, you can run the following statements:	
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 DWS, there is a lot of duplicate data in the source data with different letter cases. If 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, 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_PARTITION S USER_IND_PARTITION NS USER_PART_INDEXES USER_PART_INDEXES USER_PART_INDEXES USER_PART_INDEXES USER_TAB_PARTITIO NS
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	 Added the ALTER, DROP and VACUUM permissions at table level. Added the ALTER and DROP permissions at schema level. Add the preset roles role_signal_backend and role_read_all_stats. 	 Preset Roles Added permissions to tables: VACUUM ALTER TABLE DROP TABLE Added permissions to schemas: 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	 The query results of the pg_table_size, pg_relation_size and table_skewness views can be returned in seconds. Schema space control can collect statistics on the full space and the constraints are removed. The user/schema space calibration functions can be quickly executed, and the schema space calibration function can be automatically executed. The SQL interface allocates space quota based on the total schema space instead of the DN level. 	 Workspace Management Added the following GUC parameters for load management: space_once_adjust_n um space_readjust_sched ule Added the system catalog: PG_RELFILENODE_SIZ

Feature	Description	Reference
Enhanced performance fault locating and demarcation	 Enhanced communication exception monitoring logs Enhanced pgxc_stat_activity view Top SQL statements can be associated with unique SQL statements. Added the distributed view pg_session_wlmstat. 	 Added the system view PG_COMM_QUERY_S PEED. Added the description for the lwtid and stmt_type columns of PG_STAT_ACTIVITY. Added the unique_sql_id column information to GS_WLM_SESSION_H ISTORY. Added the system view PGXC_SESSION_WLM STAT.
Resource load management optimization and reconstruction	 Enhanced user resource monitoring: The user resource monitoring logic is modified. 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_R ESOURCE_INFO PGXC_RESPOOL_R UNTIME_INFO PGXC_RESPOOL_R SOURCE_HISTOR Y GS_RESPOOL_RES OURCE_INFO GS_RESPOOL_RES OURCE_INFO Added the system catalog: GS_RESPOOL_RES OURCE_HISTORY

Feature	Description	Reference
Modified partition table features	 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. 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_optio ns 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_DE TAIL. Add columns such as wait_on_pid and query_id. 	PGXC_WAIT_DETAILPGXC_LOCKWAIT_DE TAIL
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 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.	ddl_select_concurrent_ mode
	 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	

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_optio ns parameter now includes the normalize_negative_zer o option, which determines whether the ceil() and round() functions will return -0 for a given float value.	behavior_compat_optio ns ("Developer Guide > GUC Parameters > Miscellaneous Parameters")
internal_compat_options (added in 8.1.3.333)	A new GUC parameter internal_compat_option s 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_optio ns parameter now includes the disable_client_detectio n_commit option, which determines whether the connection to the client is checked before committing each transaction.	behavior_compat_optio ns ("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_optio ns ("Developer Guide > GUC Parameters > Miscellaneous Parameters")
gds_fill_multi_missing_fie lds (added in 8.1.3.336)	gds_fill_multi_missing_fie lds Controls the behavior when the GDS foreign table fault tolerance parameter fill_missing_fields is set to true or on.	behavior_compat_optio ns ("Developer Guide > GUC Parameters > Miscellaneous Parameters")

SQL Syntax

Table 1-18 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 B-tree 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 1-19 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 1-20 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 1-21 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_relfilen ode_size_table()	Space statistics calibration function. It does not recreate the PG_RELFILENODE_SIZE system catalog but recalibrates 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_versio n	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_coninfo, 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 1-22 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_inf o	New views related to resource monitoring.
		pgxc_respool_runtime_info	
		pgxc_respool_resource_hist ory	
		gs_respool_resource_info	
		gs_respool_runtime_info	
	4	pgxc_wait_detail	Added the enhanced view pg_locks.
		pgxc_lockwait_detail	pg_tocks.
Modified	5	DBA_PART_INDEXES	Added description for list partitions.
		DBA_PART_TABLES DBA_TAB_PARTITIONS	partitions.
		USER_IND_PARTITIONS	
		USER_PART_INDEXES	
		USER_TAB_PARTITIONS	
	6	pgxc_wlm_session_statistic s	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 1-23 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=of f to disable the autoanalyze control.
	4	User monitoring view pg_total_user_resource_i nfo	 Changed the CPU and memory resource usage and limit to the resource usage and limit in the cluster. The monitored CPU, I/O, and memory objects are changed from complex jobs to all jobs. 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 1-24 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 finegrained 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_ad apt	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_mod e	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 equijoin. 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.

1.4.2 New Features and Resolved Issues in 8.1.3.x

Table 1-25 New features/Resolved issues in version 8.1.3.336

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.1. 3.3
	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	late r.

Table 1-26 New features/Resolved issues in version 8.1.3.333

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.1. 3.3 or late r.
	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_z ero 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&&InitStreamF low" 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	

Categor y	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	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	

Table 1-27 New features/Resolved issues in version 8.1.3.330

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
New features	None	-	1	-
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	Up gra de the vers ion to 8.1. 3.3 or late r.

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	Accidental core dump "core:CStorePartitionIn- sert::findBiggestPartition" during column storage lightweight test.	During statistics collection for bulkloadUsedMem-Size 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_bulkloadUsedMem Size 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	

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	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	

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	"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	

Table 1-28 New features/Resolved issues in version 8.1.3.325

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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	Up gra de the vers ion to 8.1.
	 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	3.3 25.

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	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	

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	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	

Table 1-29 New features/Resolved issues in version 8.1.3.322

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
New features	The default value of the GUC parameter enable_limit_stop is changed to on.	-	-	-
Resolved issues	None	-	-	-

Table 1-30 New features/Resolved issues in version 8.1.3.320

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
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 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.			

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	 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.	-	1	
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	Up gra de the vers
	When the MERGE operation is performed, an error messagevalue 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	ion to 8.1. 3.3 20 or late r.

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	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	

Туре	Feature or Resolved Issue	Cause	Version	Ha ndl ing Me tho d
	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	

1.5 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 .

New features

Table 1-31 New Functions in 8.1.1

Feature	Description	Reference
2048 nodes in a cluster	Supported up to 2048 nodes in a cluster.	
Resource monitoring performance tuning	Added functions pgxc_wlm_get_schema_space(cs tring) and pgxc_wlm_analyze_schema_spac e(cstring).	
	 Added system views PGXC_TOTAL_SCHEMA_INFO, PGXC_TOTAL_SCHEMA_INFO_A NALYZE, and GS_WLM_SQL_ALLOW. 	
Lightweight UPDATE	Added descriptions about column-store tables and the lightweight UPDATE operation on column-store tables to "Precautions". Added the GUC parameter	
Driman Lloy Cl. of	enable_light_colupdate.	
Primary key CU of column-store tables	Supported primary key constraints of column-store tables.	
Space management	Added the GUC parameter bi_page_reuse_factor .	
GDS productization	Added system views PGXC_BULKLOAD_PROGRESS, PGXC_BULKLOAD_STATISTICS, and PG_BULKLOAD_STATISTICS.	

Feature	Description	Reference
Hot and cold data storage	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".	
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).	
Support for the built-in MEDIAN function	Added the function median(expression).	
Adjusting the cascade delete for tables with views	 Added the description of rebuilding a view. 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. 	

Feature	Description	Reference
Custom data redaction	 Optimized data redaction. Provided data redaction functions in different forms. Updated the columns in the system catalog PG_REDACTION_COLUMN and system view REDACTION_COLUMNS. Added the syntax ALTER REDACTION POLICY, CREATE REDACTION POLICY, and 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.	
Increasing the hash table size	Added the GUC parameter expand_hashtable_ratio.	
SMP adaptation enabled	Added "Suggestions for SMP Parameter Settings".	

Feature	Description	Reference
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.	
Enhanced concurrent SMP	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.	
row2vec optimization	Added the GUC parameter 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).	
Support for UPSERT	Added the UPSERT syntax. Added IGNORE , AS , ON DUPLICATE	
	KEY, and ON CONFLICT to the INSERT syntax.	

Feature	Description	Reference
Support for the INSERT, UPDATE, and DELETE operations of views	Added Updatable Views to "CREATE VIEW".	
One-click onsite information collection	Added the STATS boolean parameter to the EXPLAIN syntax.	
One-click relationship analysis of a lock wait	 Added the lock information function pgxc_get_lock_conflicts(). Added the 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.	
Support for Python 3.x	Supported Python 3.	
When to_date and to_timestamp process an empty string, 0001-01-01 is returned, and null is returned for TD.	 Added "Syntax Compatibility Differences Among Oracle, Teradata, and MySQL". Added the configuration item convert_empty_str_to_null_td to behavior_compat_options. 	
Support for INSERT OVERWRITE	Supported INSERT OVERWRITE.	
Obtaining all views queried by the current user	Added the view 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)	

SQL Syntax

Table 1-32 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 1-33 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 1-34 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 1-35 System functions

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

Change Type	No.	Name	Change Description
	2	pgxc_wlm_get_schema_sp ace()	You can query the schema space information of the cluster.
	3	pgxc_wlm_analyze_schem a_space()	You can analyze the schema space information of the cluster.
	4	gs_roach_enable_delay_dd l_recycle	You can enable DDL delay by specifying a backup set name.
	5	gs_roach_disable_delay_dd l_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_tim e()	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_r ead_time()	You can query the total block read time.
	26	pg_stat_get_db_total_tem p_bytes()	You can query the total size of temporary files.
	27	pg_stat_get_db_total_tem p_files()	You can query the total number of temporary files.
	28	pg_stat_get_db_total_dead locks()	You can query the total number of deadlocks.
	29	pg_stat_get_db_total_confl ict_all()	You can query the total number of conflicts.
	30	pg_stat_get_db_total_tupl es_deleted()	You can query the total number of deleted tuples.
	31	pg_stat_get_db_total_tupl es_updated()	You can query the total number of updated tuples.
	32	pg_stat_get_db_total_tupl es_inserted()	You can query the total number of inserted tuples.
	33	pg_stat_get_db_total_tupl es_fetched()	You can query the total number of fetched tuples.

Change Type	No.	Name	Change Description
	34	pg_stat_get_db_total_tupl es_returned()	You can query the total number of returned tuples.
	35	pg_stat_get_db_total_bloc ks_hit()	You can query the total number of hit data blocks in the memory.
	36	pg_stat_get_db_total_bloc ks_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_num backends()	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_ac tion	You can configure the query_band load behavior.
	46	gs_wlm_set_queryband_or der	You can configure the search priority of query_band.
	47	gs_wlm_get_queryband_ac tion	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 1-36 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_IN FO	You can query the schema space information of a cluster.

Change Type	No.	Name	Change Description
	4	PGXC_TOTAL_SCHEMA_IN FO_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_TI ME	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_TRA NSACTION	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_SLOT S	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_DIST RIBUTE	You can query the data distribution statistics of OBS cold and hot tables.
	25	PG_BULKLOAD_STATISTIC S	You can encapsulate the pg_stat_get_node_bulkload_st atistics function.
	26	PGXC_BULKLOAD_STATIST ICS	You can encapsulate the pgxc_stat_get_node_bulkload_statistics function.
	27	PGXC_BULKLOAD_PROGR ESS	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_P ATH	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_SKEWN ESS	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 1-37 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 1-38 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_warnin g	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_nodegr oup 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.

1.6 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 1-39 Sunset system functions and views

Туре	Item	Description
Functions	pg_sync_cstore_de lta(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_de lta	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_sta tus	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_s tatus	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.

Туре	Item	Description
	pg_stat_get_realti me_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_inter nal	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_in fo_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 1-40 Sunset database interfaces

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

Туре	Item	Description
External server	gsmpp_errorinfo_s erver	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 replicatio n	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 replicatio n	pg_logical_slot_pe ek_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 replicatio n	pg_logical_slot_ge t_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.

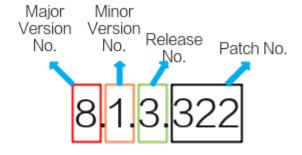
2 Version Support Notes

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	Runn ing statu s	Released On	EOM Date (End of Marketing)	EOS Date (End of Service)
9.1.0.x	Relea sed	2024-07-30	2027-06-30	2028-12-30
8.1.3.x	Relea sed	2022-03-30	2025-03-30	The EOS date is extended.
8.1.1.x	EOM	2021-07-30	2023-06-30	2025-12-31

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.
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.

Source Version	EOS	Suggestion	Upgrade Plan
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~8.1.3.3 23	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.
8.3.0.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.
9.0.1/9.0.2	No	You are advised to upgrade the version to 9.1.0.218.	You cannot directly upgrade to version 9.1.0.105 from your current version. Instead, upgrade to 9.0.3 and then to 9.1.0.218. You can upgrade the cluster on the console. For details, see Upgrading a GaussDB(DWS) Cluster.

Source Version	EOS	Suggestion	Upgrade Plan
9.0.3/9.1.0.x	No	You are advised to upgrade the version to 9.1.0.218.	It can be directly upgraded to 9.1.0.218 on the console. You are advised to contact technical support for evaluation before the upgrade. For details, see Upgrading a GaussDB(DWS) Cluster.