GOLDILOCKS 26c Release Note (en)

SUNJESOFT Inc.

Table of Contentsv

1. Overview 1
2. New Features and Enhancements 3
2.1 Database Administration and Information Views 3
Parallel Recovery 3
Parallel Backup 3
Parallel Restore 4
DICTIONARY_SCHEMA Role Information Views 4
DICTIONARY_SCHEMA Histogram Views 5
DICTIONARY_SCHEMA Library Views 5
DICTIONARY_SCHEMA Trigger Views 6
INFORMATION_SCHEMA Role Information Views 6
CHECK_CONSTRAINTS Views 6
INFORMATION_SCHEMA Trigger Information Views 7
Performance and Internal Status Views 7
RECOVERY_SLAVES 8
Histogram Generation Control Properties 8
BUFFER_LRU_SCAN_PERCENT 8
FULL_TABLE_SCAN_CACHING_THRESHOLD 9
INST_HASH_TABLE_BUCKET_MAX_COUNT 9
GLOBAL_TRANSACTION_LOG_BLOCK_SIZE 9
SHARED_SESSION_MEMORY_INIT_SIZE 9
Instant Index Reference Column Threshold 10
REDO_LOGGING_THROTTLING 10
LOG_FLUSHER_HOT_POLICY_INTERVAL 10
ARCHIVE_LOG_THROTTLING 10
2.2 SQL and PSM 11
CURRENT_ROLE and LOCAL_MEMBER_POSITION 11
GROUPING and GROUPING_ID 11
Regular Expression Functions and REGEXP_LIKE Condition 11

PRETTY Option for JSON Output 12
Controlling Duplicate JSON Object Keys 12
ORDER BY in JSON Array Aggregation 12
APPROX_COUNT_DISTINCT 12
Object-Specific Page Fragmentation Statistics Functions 13
ROLE Object DDL 13
Histogram Construction in ANALYZE TABLE 13
FOR COLUMN GROUPS in ANALYZE TABLE 14
Improved Performance and Accuracy of ANALYZE Sampling 14
Role Auditing in Audit Policies 14
Renaming a Change Tracking File 14
CHECK Constraints 15
FOREIGN KEY Constraints 15
Enabling and Disabling Constraints 15
Enabling and Disabling Indexes 16
Specifying Table Trigger Execution Order 16
ALTER TABLE name REORGANIZE 16
MERGE 17
APPEND INSERT and Hints 17
Local Shard Limit in DELETE FROM 17
ROLLUP·CUBE·GROUPING SETS in GROUP BY 18
Referencing SELECT Aliases in GROUP BY 18
PIVOT Clause 18
UNPIVOT Clause 18
TABLESAMPLE Clause 19
FILTER for Aggregate Functions 19
FILTER for JSON Aggregate Functions 19
Execution Location Hints for Window Functions and UNION ALL 19
ALTER SYSTEM CANCEL SESSION 20
SET ROLE 20
External Routine 20
Trigger Objects 21
Call Specification 21
Library Object DDL 21
Trigger Object DDL 21
2.3 API and Development Environment 23
Added the commit_write_mode JDBC Connection Property 23
Added the --parse Option to gpec 23
Support for Embedded SQL Dynamic SQL Method 4 23

Python 3 Type Stub Support for PyDBC . . . 23
Support for Character Codec Configuration in PyDBC Python 3 . . . 24
Support for Connection Diagnostics, Character Set Retrieval, and Output Converters in PyDBC Python 3 . . . 24
Support for connection.maxwrite in PyDBC . . . 24
Handling Multiple Result Sets with cursor.nextset() in PyDBC . . . 25
Added Procedure Metadata Retrieval with procedureColumns() in PyDBC . . . 25
Support for fetchval(), Cursor Iteration, and Statement Cancellation in PyDBC . . . 25
2.4 Utility . . . 26
Role DDL Output in gsql/gsqlnet . . . 26
Trigger DDL Output in gsql/gsqlnet . . . 26
APPEND INSERT Options in gloader/gloadernet . . . 26
2.5 Replication . . . 27
Data Migration to Other DBMSs Using CYCLONE . . . 27
Separation of the CYCLONE Heartbeat Protocol . . . 27
3. Compatibility and Operational Changes . . . 29
3.1 Platform, Transaction, and Server Configuration . . . 29
Changes to Release Platforms . . . 29
SERIALIZABLE Isolation Level No Longer Supported in Cluster Systems . . . 29
Default Value of DEFAULT_INDEX_PCTFREE . . . 30
Default Value of DEFAULT_MAXTRANS . . . 30
Name and Default Value of BUFFER_DIRTY_PAGE_LIMIT . . . 30
Changes to Cluster Communication Buffer Properties . . . 30
Consolidation of Plan Cache Control Properties . . . 31
Online Modification of PLAN_CACHE_SIZE . . . 31
Removal of REF_COUNT from V$SQL_CACHE . . . 31
Default Value of AGING_PLAN_INTERVAL . . . 31
Default Value of SESSION_MEMORY_INIT_SIZE . . . 32
Relaxed TRANSACTION_TABLE_SIZE Constraints in Cluster Systems . . . 32
Changes to Cluster Deadlock Detection . . . 32
Changes to Cluster Asynchronous Replica Configuration . . . 32
Default Value of SHARED_MEMORY_STATIC_SIZE . . . 32
Rename of INST_TABLE_PAGE_SIZE . . . 33
Separation of Cluster Commit Communication . . . 33
Changes to Control File Temporary File Configuration . . . 33
Renaming Online DDL Properties . . . 33
3.2 SQL and PSM . . . 35
Restriction on Concurrent Execution of ALTER INDEX COALESCE . . . 35
Removal of the ADD CONSTRAINT ON SCHEMA Privilege . . . 35
Default Schema Path Change for New Users . . . 35

Renaming the Shard Divisor Option in Online DDL . . . 36
Removal of MINSIZE Support in STORAGE . . . 36
Removal of MAXSIZE Support in Index STORAGE . . . 37
Restriction on Concurrent Execution of GSI COALESCE . . . 37
Changes to Built-in Package Installation . . . 37
3.3 API and Development Environment . . . 39
Renaming the ODBC ALTERNATE_LOCATOR Connection Attribute . . . 39
Separation of Field Roles in SQL_LONG_VARIABLE_LENGTH_STRUCT for ODBC . . . 39
Renaming the JDBC ALTERNATE_LOCATOR Connection Property . . . 39
Renaming the Embedded SQL GET CLUSTER_GROUP_ID Statement . . . 40
Separation of Field Roles in the Embedded SQL LONG VARCHAR Structure . . . 40
Changes to Negative Length Error Handling for Embedded SQL VARCHAR and VARBINARY . . . 40
Changes to PyDBC Python 3 Build and Packaging . . . 41
Clarification of the Unsupported Value of cursor.lastrowid in PyDBC . . . 41
Changes to Time Zone Type Return Behavior in PyDBC Python 3 . . . 41
Time Zone Types Returned as Strings in PyDBC Python 2 . . . 41
Improved Fractional Second Conversion for PyDBC TIME and TIMESTAMP Types . . . 42
Improved Decimal Precision and Scale Handling and Restrictions on Non-Finite Values in PyDBC . . . 42
SQLAlchemy 1.4 and 2.0 Support . . . 42
Hibernate 6, 7, and 8 Support . . . 42
3.4 Utility . . . 43
Deprecation of the directio-size Option in gloader/gloadernet . . . 43
redo_log File Type Name in gdump . . . 43
datafile File Type Name in gdump . . . 43
ALTERNATE_LOCATOR Configuration Property Name in glocator . . . 43
glocator Replication Configuration . . . 44
Removal of the Argument from the glocator sync Option . . . 44
ALTERNATE_LOCATOR Configuration Property Name in gagent . . . 44

1.

Overview

This chapter describes the features added or changed in GOLDILOCKS 26c.1. New features and enhance ments are described by product area. Changes to supported environments, settings, and syntax that may affect existing environments are summarized in the **Compatibility and Operational Changes section**.

Each item includes descriptions and reference links to the relevant sections of the User Manual to help us ers understand and use the changes.

Area	Description
Database administration	Parallel backup, restore and recovery, information views, server property changes
SQL and PSM	Roles, constraints, statistics, aggregate functions, JSON, regular expressions, MERGE, external routines, triggers
API	ODBC, JDBC, embedded SQL changes, PyDBC features and data type handling, frame work support
Utility	DDL output in gsql, APPEND INSERT in gloader, changes to gdump, glocator, and gag ent
Replication	CYCLONE integration with other DBMSs, separation of the heartbeat protocol

2.

New Features and Enhancements

2.1 Database Administration and Information Views

Parallel Recovery

Parallel recovery has been added. The number of slave threads participating in recovery can be specified using PARALLEL in ALTER DATABASE RECOVER. This feature can be used for database, data file, and table space recovery, as well as incomplete recovery. Data files are recovered by applying online and archived logs.

The number of slave threads can be set to a value between 0 and 64. If 0 or NOPARALLEL is specified, only the master thread is used. If the parallel recovery clause or the number of threads in PARALLEL is omitted, the value of RECOVERY_SLAVES is used. Complete recovery of the entire database must be performed in MOUNT phase and ARCHIVELOG mode.

For more information, refer to **ALTER DATABASE RECOVER**, **recovery slaves clause**.

Parallel Backup

Parallel backup has been added. The number of backup threads can be specified using the PARALLEL clause in ALTER DATABASE BACKUP INCREMENTAL. The allowed range is 1 to 64. If the number of threads in PARALLEL is omitted, it defaults to 1. If the PARALLEL clause itself is omitted, the operation runs in NO PARALLEL mode.

The actual number of threads used for parallel processing is limited by the number of target data files and PIECEs. If either the number of data files or the number of PIECEs is less than the specified number of parallel threads, parallel processing is performed using the smaller number.

For more information, refer to **ALTER DATABASE BACKUP**, **〈parallel clause〉**.

Parallel Restore

Parallel restore has been added. The PARALLEL clause can be used in ALTER DATABASE RESTORE and ta blespace restore statements to specify the number of threads and restore data files from incremental bac kups. A full database restore must be performed in MOUNT phase.

The number of parallel threads can be set to a value between 1 and 64. If the number of threads in PARA LLEL is omitted, it defaults to 1. If the PARALLEL clause itself is omitted, the operation runs in NOPARALL EL mode..

For more information, refer to **ALTER DATABASE RESTORE**, **parallel clause**.

DICTIONARY_SCHEMA Role Information Views

The following DICTIONARY_SCHEMA views have been added to provide information about roles and the privileges granted to them. These views allow users to check the list of roles, role grants to users and othe r roles, privileges granted to roles, and the active roles in the current session.

Views	Description
DBA_ROLES DBA_ROLE_PRIVS USER_ROLE_PRIVS	Retrieve all roles, roles granted to all users and roles, and roles granted to the current user.
ROLE_COL_PRIVS ROLE_DB_PRIVS ROLE_PACKAGE_PRIVS ROLE_PROC_PRIVS	Retrieve column, database, package, and routine privileges granted to roles.
ROLE_ROLE_PRIVS ROLE_SCHEMA_PRIVS ROLE_SEQ_PRIVS	Retrieve role grants between roles and schema and sequence privileges.
ROLE_SYS_PRIVS ROLE_TAB_PRIVS ROLE_TBS_PRIVS	Retrieve system privileges (database · tablespace · schema) table privileges, and tables pace privileges granted to roles.
SESSION_ROLES	Retrieve the roles that are active in the current session.

The ROLE_*_PRIVS views provide information about privileges granted to roles that the user can access.

For more information, refer to **DBA_ROLES**, **DBA_ROLE_PRIVS**, **ROLE_COL_PRIVS**, **SESSION_ROLES**.

DICTIONARY_SCHEMA Histogram Views

The following DICTIONARY_SCHEMA views have been added to provide information about histogram bu ckets. The BALANCE views retrieve bucket information for height-balanced histograms, while the FREQU ENCY views retrieve bucket information for frequency histograms.

Views	Description
ALL_HISTOGRAM_BALANCE, ALL_HISTOGRAM_FREQUENCY	Retrieve histogram buckets accessible to the current user.
DBA_HISTOGRAM_BALANCE, DBA_HISTOGRAM_FREQUENCY	Retrieve all histogram buckets.
USER_HISTOGRAM_BALANCE, USER_HISTOGRAM_FREQUENCY	Retrieve histogram buckets owned by the current user.

For more information, refer to **ALL_HISTOGRAM_BALANCE**, **ALL_HISTOGRAM_FREQUENCY**, **DBA_HISTO GRAM_BALANCE**, **USER_HISTOGRAM_BALANCE**.

DICTIONARY_SCHEMA Library Views

The following DICTIONARY_SCHEMA views have been added to allow users to query library objects and t heir privileges. These views help users identify the scope of library objects and distinguish between privile ges granted and privileges received.

Views	Description
ALL_LIBRARIES, DBA_LIBRARIES, USER_LIBRARIES	Displays libraries that the current user can access, all libraries in the database, and libraries owned by the current user.
ALL_LIBRARY_PRIVS, ALL_LIBRARY_PRIVS_MADE, ALL_LIBRARY_PRIVS_RECD	ALL_LIBRARY_PRIVS displays privileges associated with the current user's own ership, granting, and receiving relationships, as well as privileges for which an active role or PUBLIC is the grantee. ALL_LIBRARY_PRIVS_MADE displays privi leges for which the user is the owner or grantor. ALL_LIBRARY_PRIVS_RECD displays privileges for which the user, an active role, or PUBLIC is the grantee.
DBA_LIBRARY_PRIVS	Displays all library privilege grant information in the database.
USER_LIBRARY_PRIVS, USER_LIBRARY_PRIVS_MADE, USER_LIBRARY_PRIVS_RECD	USER_LIBRARY_PRIVS displays privileges for which the current user is the ow ner, grantor, or grantee. USER_LIBRARY_PRIVS_MADE displays privileges for which the current user is the owner. USER_LIBRARY_PRIVS_RECD displays pri vileges for which the current user is the grantee.
ROLE_LIBRARY_PRIVS	Displays library privileges granted to roles that the current user can access.

For more information, refer to **ALL_LIBRARIES**, **ALL_LIBRARY_PRIVS**, **USER_LIBRARY_PRIVS**, **ROLE_LIBRA**

RY_PRIVS를 참조한다.

DICTIONARY_SCHEMA Trigger Views

ALL_TRIGGERS, DBA_TRIGGERS, and USER_TRIGGERS have been added to query trigger objects. ALL_TRIGGERS displays triggers on tables that the current user can access, DBA_TRIGGERS displays all triggers in the database, and USER_TRIGGERS displays triggers owned by the current user.

For more information, refer to **ALL_TRIGGERS**, **DBA_TRIGGERS**, **USER_TRIGGERS**.

INFORMATION_SCHEMA Role Information Views

The following INFORMATION_SCHEMA views have been added to query roles and the privileges of enabled roles.

Views	Description
ADMINISTRABLE_ROLE_AUTHORIZATIONS	Displays role grant information for which the current user or role has WITH ADMIN OPTION
APPLICABLE_ROLES, ENABLED_ROLES	Display the roles applicable to the current SQL session and the roles that are currently enabled, respectively.
ROLE_COLUMN_GRANTS, ROLE_MODULE_GRANTS, ROLE_ROUTINE_GRANTS, ROLE_TABLE_GRANTS, ROLE_USAGE_GRANTS	Display column, module, routine, and table privileges, as well as object USAGE privileges, that the currently enabled roles can use or have granted.

For more information, refer to **ADMINISTRABLE_ROLE_AUTHORIZATIONS**, **APPLICABLE_ROLES**, **ENABLED_ROLES**, **ROLE_COLUMN_GRANTS**.

CHECK_CONSTRAINTS Views

The INFORMATION_SCHEMA.CHECK_CONSTRAINTS view has been added to query CHECK constraints. It provides information about the owner, schema, table, name, and check condition of constraints owned by a user or role. CHECK_CLAUSE provides the check condition expression and returns NULL for NOT NULL constraints.

For more information, reefer to **CHECK_CONSTRAINTS**.

INFORMATION_SCHEMA Trigger Information Views

The following INFORMATION_SCHEMA views have been added to query trigger definitions, execution order, and dependent objects.

Views	Description
TRIGGERED_UPDATE_COLUMNS	Displays columns specified as targets of UPDATE events in triggers that the current user can access.
TRIGGERS	Displays triggers defined on tables that the current user or role can access.
TRIGGER_EVENT_ORDER	Displays the execution order of triggers that share the same table, timing, row or statement level, and event.
TRIGGER_MODULE_USAGE, TRIGGER_ROUTINE_USAGE, TRIGGER_SEQUENCE_USAGE, TRIGGER_TABLE_USAGE	Display the modules, routines, sequences, and tables on which triggers depend.

For more information, refer to **TRIGGERED_UPDATE_COLUMNS**, **TRIGGERS**, **TRIGGER_EVENT_ORDER**, **TRIGGER_TABLE_USAGE**.

Performance and Internal Status Views

The following PERFORMANCE_VIEW_SCHEMA views have been added to query operational status.

Views	Description
V$DB_PROPERTY	Displays permanent properties set when the database is created. These properties are read-only after database creation.
V$RELATION	Displays the types and availability of table and index relationships, the number of allocated pages, and other related information.
V$TCL_LOGFILE	Displays the paths, group status, sizes, and other information about transaction commit log files.
V$ALLOCATOR	Displays the types and descriptions of memory allocators.
V$CLUSTER_CONNECTION, V$CLUSTER_QUEUE	Display cluster connection and queue information. These views are available only in a cluster environment.
V$UNDO_SEGMENT	Displays the list of undo segments and their allocated page information.

For more information, refer to **V$DB_PROPERTY**, **V$RELATION**, **V$TCL_LOGFILE**, **V$ALLOCATOR**, **V$CLUSTER_CONNECTION**, **V$CLUSTER_QUEUE**, **V$UNDO_SEGMENT**.

RECOVERY_SLAVES

The RECOVERY_SLAVES property has been added to control the number of slave threads used in parallel recovery. The default value is 8. If set to 0, recovery is performed using only the master thread, without a ny slave threads. If the parallel recovery clause or the number of PARALLEL is omitted from ALTER DATAB ASE RECOVER, the value set in RECOVERY_SLAVES is used.

For more information, refer to **RECOVERY_SLAVES**, **<recovery slaves clause>**.

Histogram Generation Control Properties

The following properties have been added to control the number of histogram buckets and samples whe n ANALYZE TABLE is executed.

Views	Description
HISTOGRAM_BALANCE_BUCKET_COUNT	Specifies the number of buckets in a height-balanced histog ram.
HISTOGRAM_BALANCE_MAX_SAMPLE_COUNT	Specifies the maximum number of data samples to use whe n building a height-balanced histogram.
HISTOGRAM_FREQUENCY_BUCKET_COUNT	Specifies the number of buckets used as the criterion for cre ating a frequency histogram. If the required number of buck ets exceeds the configured value, the histogram is not creat ed.

The default value for both BUCKET_COUNT properties is 0. If either value is 5 or less, the corresponding h istogram is not created. The recommended value in the manual is 20.

For more information, refer to **HISTOGRAM_BALANCE_BUCKET_COUNT**, **HISTOGRAM_BALANCE_MAX_ SAMPLE_COUNT**, **HISTOGRAM_FREQUENCY_BUCKET_COUNT**.

BUFFER_LRU_SCAN_PERCENT

The BUFFER_LRU_SCAN_PERCENT property has been added to manage buffers for disk tablespaces. Whe n no free buffers are immediately available, this property specifies the percentage of pages to scan in the LRU list to find buffers that can be reused, based on the number of buffer pages configured in BUFFER_C ACHE_SIZE. The default value is 40%.

For more information, refer to **BUFFER_LRU_SCAN_PERCENT**.

FULL_TABLE_SCAN_CACHING_THRESHOLD

The FULL_TABLE_SCAN_CACHING_THRESHOLD property has been added to determine whether to use t he buffer cache during a full scan of a disk table. Caching is determined by comparing the number of pag es used by the table with a threshold calculated based on BUFFER_CACHE_SIZE.

The default value is 20, which enables caching for tables that use pages equal to or less than 2.0% of BU FFER_CACHE_SIZE. A value of 1000 represents 100%, enabling caching for all tables scanned in full.

For more information, refer to **FULL_TABLE_SCAN_CACHING_THRESHOLD**.

INST_HASH_TABLE_BUCKET_MAX_COUNT

The INST_HASH_TABLE_BUCKET_MAX_COUNT property has been added to specify the upper limit for th e estimated bucket count of a hash instant table. This property can be changed at the session level. Chan ges made using ALTER SYSTEM are applied according to the DEFERRED mode.

For more information, refer to **INST_HASH_TABLE_BUCKET_MAX_COUNT**.

GLOBAL_TRANSACTION_LOG_BLOCK_SIZE

The GLOBAL_TRANSACTION_LOG_BLOCK_SIZE property has been added to specify the block size of glob al transaction log files. The unit is bytes, and the value must be one of 512, 1024, 2048, or 4096. The def ault value is 512, and this property cannot be changed.

For more information, refer to **GLOBAL_TRANSACTION_LOG_BLOCK_SIZE**.

SHARED_SESSION_MEMORY_INIT_SIZE

The SHARED_SESSION_MEMORY_INIT_SIZE property has been added to specify the initial memory size fo r shared sessions. The default value is 786432 (768K). This property is distinct from SESSION_MEMORY_I NIT_SIZE, which specifies the initial memory size for dedicated sessions.

For more information, refer to **SHARED_SESSION_MEMORY_INIT_SIZE**.

Instant Index Reference Column Threshold

The INSTANT_INDEX_REFERENCE_COLUMN_THRESHOLD property has been added to reduce the storage space used by instant indexes. If the size of a column stored in an instant index is greater than or equal to the configured threshold, the column is stored by reference.

If an instant index key exceeds the 16,000-byte limit, this property can be adjusted to reduce the key size. Using reference-based storage can reduce index space usage, but may degrade performance.

For more information, refer to **INSTANT_INDEX_REFERENCE_COLUMN_THRESHOLD**.

REDO_LOGGING_THROTTLING

The REDO_LOGGING_THROTTLING property has been added to control the redo logging rate during inde x creation and rebuilding. It is intended to reduce the impact of large volumes of log writes on the system and online services. If the number of dirty blocks in the log buffer exceeds the configured value, the oper ation waits until the blocks are flushed to disk.

For more information, refer to **REDO_LOGGING_THROTTLING**.

LOG_FLUSHER_HOT_POLICY_INTERVAL

The LOG_FLUSHER_HOT_POLICY_INTERVAL property has been added to control the busy-waiting duratio n of the log flusher thread in gmaster while waiting for an event. The unit is microseconds, and the defau lt value is 0, which disables busy waiting. Increasing this value can speed up flushing the log buffer to dis k, but it also increases CPU usage.

For more information, refer to **LOG_FLUSHER_HOT_POLICY_INTERVAL**.

ARCHIVE_LOG_THROTTLING

The ARCHIVE_LOG_THROTTLING property has been added to control disk I/O during redo log archiving. Each time the amount of data copied to the archive destination file exceeds the configured value, the pro cess sleeps once to throttle I/O during the copy operation.

For more information, refer to **ARCHIVE_LOG_THROTTLING**.

2.2 SQL and PSM

CURRENT_ROLE and LOCAL_MEMBER_POSITION

The system information functions CURRENT_ROLE and LOCAL_MEMBER_POSITION() have been added. CURRENT_ROLE returns the role set for the current session, while LOCAL_MEMBER_POSITION() returns t he cluster member position of the server processing the user query. LOCAL_MEMBER_POSITION() can be used to identify the executing member in a cluster system.

For more information, refer to **CURRENT_ROLE**, **LOCAL_MEMBER_POSITION**.

GROUPING and GROUPING_ID

The GROUPING and GROUPING_ID functions have been added. They are used with GROUP BY and retur n an integer representing a bit value in which each argument is set to 0 if it is a current grouping key, or 1 otherwise. This allows the grouping status of subtotal and grand total rows to be distinguished. GROU PING_ID behaves the same as GROUPING and accepts up to 31 arguments

For more information, refer to **GROUPING**, **GROUPING_ID**.

Regular Expression Functions and REGEXP_LIKE Condition

The following functions and condition that use regular expressions have been added. A target string and a pattern must be specified. The pattern can be up to 512 bytes long.

Name	Description
REGEXP_COUNT	Returns the number of matches for the pattern.
REGEXP_INSTR	Returns the starting position of a matching string or the position of the character imm ediately following it.
REGEXP_LIKE Condition	Checks whether a string matches the pattern.
REGEXP_REPLACE	Replaces a matching string with the specified string.
REGEXP_SUBSTR	Returns a substring that matches the pattern.

For more information, refer to **REGEXP_COUNT**, **REGEXP_INSTR**, **REGEXP_LIKE**, **REGEXP_REPLACE**, **REG EXP_SUBSTR**.

PRETTY Option for JSON Output

The PRETTY option has been added to the JSON Output Clause of the JSON String Constructor. To chang e the output format, specify PRETTY after the RETURNING clause, which defines the result type of the JS ON string. The return type can be specified as CHAR(n), VARCHAR(n), or LONG VARCHAR.

For more information, refer to **JSON Output Clause**.

Controlling Duplicate JSON Object Keys

A JSON key uniqueness constraint option has been added to JSON_OBJECT, JSON_OBJECTAGG, and JSO N_OBJECTAGG() OVER. WITH UNIQUE KEYS returns an error if a JSON object contains duplicate keys, wh ile WITHOUT UNIQUE KEYS allows duplicate keys. This option allows the duplicate key handling policy to be specified within the function used to generate the JSON object.

For more information, refer to **JSON Key Uniqueness Constraint**, **JSON_OBJECT**, **JSON_OBJECTAGG**, **JSO N_OBJECTAGG() OVER**.

ORDER BY in JSON Array Aggregation

A JSON array aggregate ORDER BY clause option has been added to JSON_ARRAYAGG and JSON_ARRA YAGG() OVER. The function generates a JSON array after sorting values using the ORDER BY clause speci fied within the function. ASC or DESC and NULLS FIRST or NULLS LAST can be specified. The defaults are ASC and NULLS LAST.

For more information, refer to **JSON Array Aggregate Order By Clause**, **JSON_ARRAYAGG**, **JSON_ARRAY AGG() OVER**.

APPROX_COUNT_DISTINCT

The approximate aggregate function APPROX_COUNT_DISTINCT has been added. It estimates the numb er of distinct non-NULL values using the HyperLogLog algorithm. It is useful for estimating the number of distinct values in large datasets. The returned value may differ from the exact count returned by COUNT (DISTINCT ...). When FILTER is specified, only values that satisfy the filter condition are included in the agg regation.

For more information, refer to **APPROX_COUNT_DISTINCT**.

Object-Specific Page Fragmentation Statistics Functions

The TABLE_PHYSICAL_STATS, INDEX_PHYSICAL_STATS, and GSI_PHYSICAL_STATS functions have been added to retrieve page fragmentation statistics. TABLE_PHYSICAL_STATS analyzes regular tables, INDEX_PHYSICAL_STATS analyzes indexes, and GSI_PHYSICAL_STATS analyzes global secondary indexes (GSIs) of clustered tables. The target name must be specified as an identifier, not as a string.

All three functions return page, used, and fragmented information as VARCHAR values. The optional sampling_ratio_value argument specifies the percentage of pages to analyze. It accepts values from 1 to 100 and defaults to 100. When sampling is used, used and fragmented are calculated based on the sampled pages, while page returns the total number of allocated pages.

For more information, refer to **TABLE_PHYSICAL_STATS**, **INDEX_PHYSICAL_STATS**, **GSI_PHYSICAL_STATS**.

ROLE Object DDL

DDL statements for creating and dropping roles and granting and revoking roles to and from users or other roles have been added. No privileges are automatically granted to a newly created role, so the required privileges must be granted separately.

Syntax	Description
CREATE ROLE	Defines a role. The CREATE ROLE ON DATABASE privilege is required.
DROP ROLE	Drops a role.
GRANT role TO	Grants a role to a user or another role.
REVOKE role FROM	Revokes a previously granted role.

For more information, refer to **CREATE ROLE**, **DROP ROLE**, **GRANT role TO**, **REVOKE role FROM**.

Histogram Construction in ANALYZE TABLE

ANALYZE TABLE can now build height-balanced and frequency histograms. Setting HISTOGRAM_BALANCE_BUCKET_COUNT enables the construction of a height-balanced histogram, while setting HISTOGRAM_FREQUENCY_BUCKET_COUNT enables the construction of a frequency histogram. Both properties default to 0, and the corresponding histogram is not constructed if the property value is 5 or less. The recommended value is 20. A frequency histogram is not created if the number of required buckets exceeds the configured value.

For more information, refer to **ANALYZE TABLE**, **HISTOGRAM_BALANCE_BUCKET_COUNT**, **HISTOGRAM_FREQUENCY_BUCKET_COUNT**.

FOR COLUMN GROUPS in ANALYZE TABLE

The FOR COLUMN GROUPS clause has been added to ANALYZE TABLE. It collects NUM_DISTINCT statistics for specified column combinations, which can be used for statistical analysis of conditions involving multiple columns. Each column group must contain 2 to 4 columns, and the same column cannot be specified more than once. Column groups that differ only in column order are treated as the same column group.

For more information, refer to **FOR COLUMN GROUPS**, **ANALYZE TABLE**.

Improved Performance and Accuracy of ANALYZE Sampling

ANALYZE sampling has been enhanced with repeatable random sampling, approximate NUM_DISTINCT estimation (hyper log algorithm), and NUM_DISTINCT correction using an exponential curve. For example, using 10% sampling can produce the same execution plans as a full ANALYZE for all 22 TPC-H queries.

The ESTIMATE STATISTICS clause of ANALYZE TABLE allows the number or percentage of rows to be sampled to be specified. The percentage specified by SAMPLE percentage PERCENT must be an integer between 1 and 99. If the number of sampled rows is less than MIN_SAMPLE_ROW_COUNT, the value of that property is used.

For more information, refer to **ESTIMATE STATISTICS**.

Role Auditing in Audit Policies

The role_audit_clause has been added to CREATE AUDIT POLICY and ALTER AUDIT POLICY. The ROLES role_name clause specifies that SQL statements executed using a particular role are subject to auditing. The created policy must be enabled using AUDIT POLICY. The AUDIT SYSTEM ON DATABASE privilege is required to create or modify audit policies. Even for SQL statements executed by the same user, audit results may differ depending on whether the operations are performed using the object's owner privileges or role privileges.

For more information, refer to **role_audit_clause**, **ALTER AUDIT POLICY**.

Renaming a Change Tracking File

The ALTER DATABASE RENAME CHANGE TRACKING FILE statement has been added. A new name can be specified for a change tracking file using this statement. The database must be in the MOUNT phase, a

nd change tracking must be enabled. The ALTER DATABASE ON DATABASE privilege is required to execu te this statement. If an absolute path is not specified for the file name, the path is constructed by combini ng SYSTEM_TABLESPACE_DIR with the file name.

For more information, refer to **ALTER DATABASE RENAME CHANGE TRACKING FILE**.

CHECK Constraints

CHECK constraints have been added. A logical expression can be defined in CREATE TABLE, ALTER TABL E name ADD COLUMN, and ALTER TABLE name ADD CONSTRAINT to specify the condition that each ro w must satisfy. A condition is considered violated if it evaluates to FALSE, while TRUE or UNKNOWN is co nsidered to satisfy the condition.

Subqueries, references to other objects, nondeterministic expressions, and pseudo-columns are not allow ed in the condition. The maximum number of referenced columns is 32. To restrict NULL values as well, t he handling of UNKNOWN results must be taken into account.

For more information, refer to **CHECK constraint**, **CREATE TABLE**, **ALTER TABLE name ADD COLUMN**, **AL TER TABLE name ADD CONSTRAINT**.

FOREIGN KEY Constraints

FOREIGN KEY constraints have been added to define referential relationships in CREATE TABLE, ALTER T ABLE name ADD COLUMN, and ALTER TABLE name ADD CONSTRAINT. Foreign key values must exist in t he referenced table's PRIMARY KEY or UNIQUE constraint. However, foreign key values containing NULL are allowed as an exception.

For column constraints, the REFERENCES clause is used. For table constraints, the FOREIGN KEY (...) REFE RENCES clause is used. Foreign keys can be defined only on base tables, not on temporary tables or views.

For more information, refer to **FOREIGN KEY constraint**, **CREATE TABLE**, **ALTER TABLE name ADD COLU MN**, **ALTER TABLE name ADD CONSTRAINT**.

Enabling and Disabling Constraints

The constraint enforcement option is now available in ALTER TABLE name ALTER CONSTRAINT to change the enforcement status of a constraint. ENFORCED enables the constraint, while NOT ENFORCED disables it. The constraint to be modified can be specified by its name or by the corresponding UNIQUE constraint associated with a PRIMARY KEY or a column list.

For more information, refer to **ALTER TABLE name ALTER CONSTRAINT**.

Enabling and Disabling Indexes

The ALTER INDEX name ENABLE/DISABLE statement has been added to control index enforcement. ENA BLE builds and enables the index, while DISABLE disables the index and releases all extents used by it. Dis abled indexes are not used for DML or SELECT operations.

ENABLE and ENFORCED perform the same operation, as do DISABLE and NOT ENFORCED. Indexes creat ed for key constraints must be managed using ALTER CONSTRAINT.

For more information, refer to **ALTER INDEX name ENABLE/DISABLE**.

Specifying Table Trigger Execution Order

The ALTER TABLE name SET TRIGGER ORDER statement has been added to change the execution order of triggers by rearranging the names of triggers defined on a table. If only some triggers are specified, the y are executed first in the specified order, while the remaining triggers retain their existing order. All liste d triggers must belong to the same table and have the same timing (BEFORE or AFTER) and execution gr anularity (ROW or STATEMENT).

For more information, refer to **ALTER TABLE name SET TRIGGER ORDER**.

ALTER TABLE name REORGANIZE

The ALTER TABLE name REORGANIZE statement has been added to physically reorganize a table. The ON LINE option allows INSERT, UPDATE, and DELETE operations during reorganization, while the OFFLINE op tion does not. The default is ONLINE. The SCAN PARTITION and PARALLEL options can be used to specify partition scanning and parallel processing.

The default redo logging mode is LOGGING. When NOLOGGING is specified, the table may become unus able if the server terminates abnormally after reorganization. Therefore, when using NOLOGGING, a CHE CKPOINT must be performed after reorganization, as described in the manual.

For more information, refer to **ALTER TABLE name REORGANIZE**.

MERGE

The MERGE statement has been added. It compares a source specified by USING with a target table base d on the ON condition. If a match is found, UPDATE, DELETE, or DO NOTHING is performed; otherwise, I NSERT or DO NOTHING is performed. Multiple data modification operations can be specified in a single st atement, each with its own condition.

For each target row, the WHEN clauses are evaluated in the order in which they appear, and only the first matching branch is executed. No separate MERGE privilege is required. SELECT privilege on the reference d columns and the privileges required for the INSERT, UPDATE, and DELETE operations specified in the st atement are required.

For more information, refer to **MERGE**.

APPEND INSERT and Hints

The APPEND INSERT data loading method and the hint used to specify it have been added. This method a ppends data to unused or newly allocated space, making it suitable for bulk data loading. For disk tables, data is written directly to the data files without passing through the buffer.

APPEND INSERT can be used with the APPEND hint in INSERT statements, the corresponding options in gl oader, or certain bulk synchronization DDL statements. PARALLEL allows multiple sessions to perform AP PEND INSERT operations concurrently, while NOLOGGING, when used with this option, reduces the amo unt of log data generated. STATEMENT_NOFORCE specifies whether the data pages used by a statement are written to disk when the statement completes.
Depending on the specified options, referential integrity checks and index key updates may be handled di fferently. Therefore, the appropriate options should be selected based on the data loading method.

For more information, refer to **Adding Data Using the APPEND INSERT Method**.

Local Shard Limit in DELETE FROM

A local shard limit clause has been added to DELETE FROM to limit the number of records deleted. For a s harded table in a cluster, this clause deletes up to limit_row_count records from the local group. It can be used only with sharded tables, and limit_row_count must be a positive value. This clause cannot be used with the result offset clause or the fetch limit clause.

For more information, refer to **local shard limit clause**.

ROLLUP·CUBE·GROUPING SETS in GROUP BY

Support for ROLLUP, CUBE, and GROUPING SETS has been added to the grouping elements of GROUP BY. ROLLUP progressively reduces the listed grouping columns to calculate subtotals and a grand total. CUBE calculates all possible grouping combinations, while GROUPING SETS explicitly specifies only the required combinations. These features allow subtotal configurations to be selected according to aggregation requirements.

For more information, refer to **group by clause**.

Referencing SELECT Aliases in GROUP BY

GROUP BY grouping column references can now refer to aliases defined in the SELECT list. An alias assigned to a column in the SELECT list can be used as a grouping key. If multiple aliases have the same name, the reference is ambiguous. If an alias has the same name as an actual column, the actual column reference takes precedence.

```
SELECT COUNT(*), t1.c1 AS A1 FROM t1 GROUP BY A1;
```

For more information, refer to **group by clause**.

PIVOT Clause

The PIVOT clause has been added to convert row values into columns. A cross table can be created by specifying an aggregate function, a pivot column in the FOR clause, and a list of values in the IN clause of a table reference in the FROM clause. Aggregate functions cannot be nested, and the number of columns specified in the FOR clause must match the number of elements in each value combination listed in the IN clause.

For more information, refer to **pivot clause**.

UNPIVOT Clause

The UNPIVOT clause has been added to convert column values into rows. The IN clause specifies the names of the source columns to be unpivoted. The names of the columns that store the converted values and identify the source columns are defined using the value column and the FOR column, respectively. To include NULL values in the result, specify INCLUDE NULLS. If omitted, EXCLUDE NULLS is applied by default.

For more information, refer to **unpivot clause**.

TABLESAMPLE Clause

The SAMPLE clause has been added to randomly sample a subset of rows from a table. TABLESAMPLE specifies a sampling percentage greater than 0 and less than or equal to 100. The sampling unit can be selected using PERCENT ROWS or PERCENT PAGES. If the sampling unit is omitted, PERCENT ROWS is applied by default. A seed value can be specified using REPEATABLE(seed_value). If no seed value is specified, the sampling results may vary between executions.

For more information, refer to **sample clause**.

FILTER for Aggregate Functions

The FILTER feature has been added to aggregate functions. By specifying FILTER (WHERE condition) after an aggregate function call, only values that satisfy the condition are included in the aggregation. Different conditions can be specified for individual aggregate functions within a single SELECT statement.

```
SELECT COUNT(*), COUNT(*) FILTER (WHERE c1 > 1) FROM t1;
```

For more information, refer to **Aggregate Function**, **COUNT(*) FILTER**.

FILTER for JSON Aggregate Functions

The FILTER feature has been added to JSON aggregate constructors. By specifying FILTER (WHERE condition) after JSON_ARRAYAGG or JSON_OBJECTAGG, only values that satisfy the condition are included in the resulting array or object. The aggregation result is returned as a single row containing a JSON string.

For more information, refer to **JSON aggregate Constructor**, **JSON_ARRAYAGG**, **JSON_OBJECTAGG**.

Execution Location Hints for Window Functions and UNION ALL

Window hints and UNION ALL driver hints have been added to SQL hints. In a cluster query, these hints specify whether window functions or UNION ALL operations are processed on the current server or on the

servers of each group. These driver hints are ignored in standalone environments.

Hint Category	Current Server	Each Group's Server
window hints	LOCAL_WINDOW	REMOTE_WINDOW
union all driver hints	LOCAL_UNION_ALL	REMOTE_UNION_ALL

For more information, refer to **window hints**, **union all driver hints**.

ALTER SYSTEM CANCEL SESSION

The ALTER SYSTEM CANCEL SESSION statement has been added to cancel an operation running in a session. The session ID and serial number of the target session must be specified. In a cluster database, the target member position can also be specified. The ALTER SYSTEM ON DATABASE privilege is required to execute this statement.

For more information, refer to **ALTER SYSTEM CANCEL SESSION**.

SET ROLE

The SET ROLE role_name statement has been added to change the role of the current session. A role that can be applied to the current user can be selected, or a role can be set using the ACCESS CONTROL ON DATABASE privilege. Specifying SET ROLE NONE clears the current role setting. Roles cannot be changed during an active transaction. After the role is changed, statements are executed based on the configured current role.

For more information, refer to **SET ROLE role_name**.

External Routine

External routines have been added to execute user-written C programs. Functions stored in a shared library can be called from PSM routines. The external routine process, gextproc, loads and executes the library and returns the results to PSM. By defining a library object and a call specification, external functions can be used within the DBMS.

For more information, refer to **External Routine**, **Call Specification**.

Trigger Objects

Trigger objects have been added to define actions that execute automatically when DML operations are p erformed on a specific table. Actions written in PSM can be associated with INSERT, UPDATE, and DELET E events and configured to execute BEFORE or AFTER an event, at either the ROW or STATEMENT level. E nabled triggers execute automatically when the associated event occurs and cannot be invoked explicitly. When the target table is dropped, its dependent triggers are also dropped.

For more information, refer to **Trigger Overview**, **CREATE TRIGGER**.

Call Specification

Call specifications have been added to map external C functions to PSM routines. The LANGUAGE C, LIB RARY, NAME, and PARAMETER clauses are used to map the external function name and the data types o f its arguments and return value, allowing the function to be called from PSM. The EXECUTE privilege on the target library, or an equivalent privilege specified in the manual, is required to use a call specification.

For more information, refer to **Call Specification**.

Library Object DDL

The CREATE LIBRARY and DROP LIBRARY statements have been added to manage library objects. CREAT E LIBRARY associates a shared library file path for a C program with a schema object name. Specifying OR REPLACE replaces an existing object. DROP LIBRARY removes a library object, and specifying IF EXISTS en sures that the object is dropped only if it exists.

For more information, refer to **CREATE LIBRARY**, **DROP LIBRARY**.

Trigger Object DDL

The following DDL statements have been added to create, drop, recompile, enable or disable, and renam e trigger objects.

Syntax	Description
CREATE TRIGGER	Defines DML events and the actions to be executed.
DROP TRIGGER	Drops a trigger.
ALTER TRIGGER name COMPILE	Recompiles a trigger.

Syntax	Description
ALTER TRIGGER name ENABLE/DISABLE	Enables or disables a trigger.
ALTER TRIGGER name RENAME TO	Renames a trigger.

For more information, refer to **CREATE TRIGGER**, **DROP TRIGGER**, **ALTER TRIGGER name COMPILE**, **ALTER TRIGGER name ENABLE/DISABLE**, **ALTER TRIGGER name RENAME TO**.

2.3 API and Development Environment

Added the commit_write_mode JDBC Connection Property

The commit_write_mode connection property has been added to JDBC. This property specifies how logs generated when a transaction completes are flushed to the log files on disk. A value of 0 uses the server c onfiguration, 1 specifies wait, and 2 specifies no wait. The default value is 0.

For more information, refer to **JDBC: connection property**.

Added the --parse Option to gpec

The --parse option has been added to gpec, the Embedded SQL precompiler. This option specifies the par sing level for the input source code. none parses only the minimum structure required for execution, whil e partial parses the source to the extent required for preprocessing. The default value is partial.
For example, it can be used as *gpec --parse=none sample.gc*.

For more information, refer to **Embedded SQL: --parse**.

Support for Embedded SQL Dynamic SQL Method 4

Embedded SQL now supports dynamic SQL method 4. This method is used for SQL statements whose nu mber or types of columns to be retrieved, or number of parameters are unknown at precompile time. It p rovides the ANSI SQL descriptor and SQLDA approaches. These approaches use descriptors to specify inp ut and output information at runtime for PREPARE, EXECUTE, or cursor operations. Neither approach su pports array operations using struct, the SQL_CURSOR type, or arrays of the VARCHAR type.

For more information, refer to **Embedded SQL: Method 4**.

Python 3 Type Stub Support for PyDBC

The PyDBC Python 3 package now includes the pygoldilocks.pyi type stub, which can be used with IDEs a nd static type checking tools.

For more information, refer to **PyDBC: API Reference**.

Support for Character Codec Configuration in PyDBC Python 3

PyDBC Python 3 supports connection.setencoding() and setdecoding().
setencoding(encoding, ctype=SQL_CHAR) specifies the codec used to convert SQL statements and string parameters to bytes.
setdecoding(sqltype, encoding, ctype=SQL_CHAR) specifies the codec used to convert character results to Python strings.
Only SQL_CHAR is supported as the argument; wide character types are not supported. These settings persist for the lifetime of the connection and are not reset by executing SQL statements or calling commit() or rollback().

For more information, refer to **PyDBC: Character Codec Configuration**.

Support for Connection Diagnostics, Character Set Retrieval, and Output Converters in PyDBC Python 3

PyDBC Python 3 supports connection.messages, character_set_name(), and SQL-type-specific output converters. The messages attribute stores DB-API warnings generated during connection operations. This list is cleared when the next connection operation begins, so it should be checked immediately after each operation. The character_set_name() method returns the name of the GOLDILOCKS character set currently configured for the connection.

When an output converter is registered using add_output_converter(sqltype, func), it is applied to all cursors associated with the connection. The conversion function receives raw bytes or None representing SQL NULL, and its return value becomes the corresponding value in the query result row. Converter registrations and modifications take effect starting with the next call to execute().

For more information, refer to **PyDBC: Output Converter**.

Support for connection.maxwrite in PyDBC

PyDBC supports connection.maxwrite for handling large character and binary parameters. This attribute specifies the threshold size, in bytes, that determines whether parameters are bound directly or sent using SQLPutData.
A value of 0 uses the default threshold determined by the driver and data type. Custom values must be at least 255.

This attribute affects only how input data is transmitted. It does not truncate data or limit the size of quer y results.

For more information, refer to **PyDBC: maxwrite**.

Handling Multiple Result Sets with cursor.nextset() in PyDBC

Multiple result sets returned by a procedure can be processed sequentially using cursor.nextset(). Rows in the current result set can be retrieved using fetch methods such as fetchone(). Calling nextset() advances to the next result set. It returns True if another result set is available, or None if no more result sets remai n.

For more information, refer to **PyDBC: nextset()**.

Added Procedure Metadata Retrieval with procedureColumns () in PyDBC

PyDBC now provides cursor.procedureColumns(procedure=None, catalog=None, schema=None). This m ethod retrieves metadata for a procedure's IN, OUT, and INOUT parameters, return values, and result colu mns through the ODBC SQLProcedureColumns function.
The returned metadata includes information such as column_name, column_type, data_type, type_name, column_size, and ordinal_position.

For more information, refer to **PyDBC: procedureColumns()**.

Support for fetchval(), Cursor Iteration, and Statement Cancel lation in PyDBC

PyDBC supports fetchval(), cursor iteration, and statement cancellation.
The fetchval() method returns the first column value of the next row, or None if no more rows are availab le.
Cursor iteration retrieves and processes result rows one at a time.
The cursor.cancel() method requests cancellation of the statement currently executing on that cursor. Exc eptions may occur during cancellation, so after handling any exceptions, verify whether the cursor can be reused.

For more information, refer to PyDBC: **fetchval()** and **cancel()**.

2.4 Utility

Role DDL Output in gsql/gsqlnet

The ₩ddl_role command has been added to output the DDL for a role object. Running ₩ddl_role name outputs all statements related to creating the role, granting privileges, and adding comments.
To output only specific statements, specify the CREATE, GRANT, or COMMENT option.

For more information, refer to **Role DDL Output** in gsql/gsqlnet.

Trigger DDL Output in gsql/gsqlnet

The ₩ddl_trigger command has been added to output the DDL for a trigger object. Running ₩ddl_trigge r name outputs the statements for creating the trigger and adding comments.
To output only specific statements, specify the CREATE or COMMENT option. Objects stored in the recycl e bin are excluded from the output.

For more information, refer to **Trigger DDL Output** in gsql/gsqlnet.

APPEND INSERT Options in gloader/gloadernet

The APPEND INSERT method for loading data and the --merge, --skip_index_maintenance, and --nologgi ng options have been added. During import operations, --merge EXTENT loads data using serial APPEND INSERT, while --merge SEGMENT loads data using PARALLEL APPEND INSERT.

Option	Usage
--skip_index_maintenance	Used with --merge. Skips index maintenance and sets indexes to the UNUSABLE state. If omitted, the operation uses DEFERRED_INDEX_MAINTENANCE.
--nologging	Used with --merge SEGMENT to load data using NOLOGGING APPEND INSERT. If omitted, data is loaded using LOGGING.

The existing --append option specifies whether existing data is retained, deleted, or truncated before loa ding. It serves a different purpose from the newly added --merge option.

For more information, refer to **APPEND INSERT Options** in gloader/gloadernet.

2.5 Replication

Data Migration to Other DBMSs Using CYCLONE

The feature for migrating data to Oracle, DB2, MySQL, and Tibero using CYCLONE has been added. The CYCLONE master analyzes changes in the source GOLDILOCKS redo log and transmits them to the slave, which applies the changes to the target DBMS using ODBC.

Depending on the target DBMS, specify the path to the corresponding ODBC driver file using the ORACLE_DRIVER, DB2_DRIVER, MYSQL_DRIVER, or TIBERO_DRIVER setting on the slave. For MySQL and DB2, specify the target database using MYSQL_DATABASE or DB2_DATABASE, respectively. For SYNC operations, use the corresponding SYNC_ORACLE_DRIVER, SYNC_DB2_DRIVER, SYNC_MYSQL_DRIVER, or SYNC_TIBERO_DRIVER setting on the master.

The column data types in GOLDILOCKS and the target DBMS must match. When integrating with another DBMS, DDL processing during CYCLONE replication is not supported.

For more information, refer to **Data Migration to Other DBMSs Using CYCLONE**.

Separation of the CYCLONE Heartbeat Protocol

The CYCLONE heartbeat protocol has been separated from the data protocol. The connection timeout is configured using the existing HEARTBEAT_TIMEOUT setting. This setting is available on both the master and slave and specifies the maximum time, in seconds, to maintain a connection when communication is disrupted due to network disconnections or system failures. The default value is 30 seconds, and the minimum value is 10 seconds.

For more information, refer to **Separation of the CYCLONE Heartbeat Protocol**.

3.

Compatibility and Operational Changes

3.1 Platform, Transaction, and Server Configuration

Changes to Release Platforms

HP, AIX, and PPC64 have been removed from the supported release platforms. The current release platform table in the manual lists linux-x86_64 for servers and Linux- and Windows-based client platforms. The appropriate package must be selected for the target installation environment.

For more information, refer to **Release Platform**.

SERIALIZABLE Isolation Level No Longer Supported in Cluster Systems

The SERIALIZABLE transaction isolation level is no longer supported in cluster systems. In a cluster environment, READ COMMITTED must be used when setting the isolation level with SET TRANSACTION or SET SESSION CHARACTERISTICS AS TRANSACTION.
The isolation level set using SET TRANSACTION applies to the next transaction and is restored to the default setting when that transaction ends.

For more information, refer to the isolation level sections in **SET TRANSACTION** and **SET SESSION CHARACTERISTICS**.

Default Value of DEFAULT_INDEX_PCTFREE

The default value of DEFAULT_INDEX_PCTFREE has been changed to 10. When PCTFREE is not explicitly specified during index creation, the value of this property is applied. Therefore, the updated default value of 10 also applies when PCTFREE is omitted from existing creation scripts.

For more information, refer to **DEFAULT_INDEX_PCTFREE**.

Default Value of DEFAULT_MAXTRANS

The default value of DEFAULT_MAXTRANS has been changed to 32. When MAXTRANS is not explicitly s pecified during table or index creation, the value of this property is applied.

For more information, refer to **DEFAULT_MAXTRANS**.

Name and Default Value of BUFFER_DIRTY_PAGE_LIMIT

INCREMENTAL_CHECKPOINT_CRITERIA has been renamed to BUFFER_DIRTY_PAGE_LIMIT, and its defaul t value has been changed to 0. The previous name is documented as an alias in the manual.

This property determines when to perform an incremental checkpoint based on the number of modified pages in the buffer cache. The default value of 0 represents infinity, meaning that an incremental checkp oint is not triggered by this criterion, even if all pages in the buffer cache have been modified.

For more information, refer to **BUFFER_DIRTY_PAGE_LIMIT**.

Changes to Cluster Communication Buffer Properties

CLUSTER_CM_BUFFER_COUNT has been deprecated, and the following properties have been added to c onfigure the number of communication buffers for each dispatcher.

Property	Description
LOCKABLE_DISPATCHER_CM_BUFFER_COUNT	Specifies the number of communication buffers for the l ockable dispatcher.
LOCKLESS_DISPATCHER_CM_BUFFER_COUNT	Specifies the number of communication buffers for the l ockless dispatcher.
SYNC_DISPATCHER_CM_BUFFER_COUNT	Specifies the number of communication buffers for the s ynchronization dispatcher.

The default value for each property is 1. These properties cannot be modified using ALTER SESSION or AL TER SYSTEM.

For more information, refer to **LOCKABLE_DISPATCHER_CM_BUFFER_COUNT**, **LOCKLESS_DISPATCHER_ CM_BUFFER_COUNT**, **SYNC_DISPATCHER_CM_BUFFER_COUNT**.

Consolidation of Plan Cache Control Properties

MAXIMUM_FLANGE_COUNT has been deprecated, and the plan cache is now controlled solely by PLAN_ CACHE_SIZE. PLAN_CACHE_SIZE specifies the amount of memory allocated to the plan cache in bytes. T his value must not be interpreted as the number of SQL statements.

For more information, refer to **PLAN_CACHE_SIZE**.

Online Modification of PLAN_CACHE_SIZE

PLAN_CACHE_SIZE can now be modified online without restarting the server. This property specifies the amount of memory allocated to the SQL execution plan cache. Session-level modifications are not allowe d; only system-level modifications are permitted.

For more information, refer to **PLAN_CACHE_SIZE**.

Removal of REF_COUNT from V$SQL_CACHE

The REF_COUNT column has been removed from V$SQL_CACHE. This view provides statistics on shared SQL execution plans, including SQL text, the number of plans, plan size, and plan text. Monitoring querie s that explicitly reference REF_COUNT must be updated to reflect the revised column list.

For more information, refer to **V$SQL_CACHE**.

Default Value of AGING_PLAN_INTERVAL

The default value of AGING_PLAN_INTERVAL has changed from 0 to 3. The unit is seconds. SQL executio n plans older than the specified interval become eligible for aging.

For more information, refer to **AGING_PLAN_INTERVAL**.

Default Value of SESSION_MEMORY_INIT_SIZE

The default value of SESSION_MEMORY_INIT_SIZE has changed from 131072 (128 KB) to 524288 (512 KB). This property specifies the initial memory size allocated for a dedicated session.

For more information, refer to **SESSION_MEMORY_INIT_SIZE**.

Relaxed TRANSACTION_TABLE_SIZE Constraints in Cluster Systems

The requirement for all cluster members to have the same TRANSACTION_TABLE_SIZE value has been removed. The database must be restarted to change the transaction table size of an individual member.

When reducing the configured value, it must be set to a value greater than the maximum slot identifier used by prepared transactions after restart recovery. Setting it to a value less than or equal to this maximum may cause the restart to fail.

For more information, refer to **TRANSACTION_TABLE_SIZE**.

Changes to Cluster Deadlock Detection

With the introduction of built-in cluster deadlock detection, the existing CLUSTER_DEADLOCK_TIMEOUT property has been deprecated.

Changes to Cluster Asynchronous Replica Configuration

Cluster replica updates have been improved so that async replica is used by default. If asynchronous replication is not supported in a given environment, the system handles this automatically. The existing CLUSTER_ASYNC_REPLICATION property has been deprecated.

Default Value of SHARED_MEMORY_STATIC_SIZE

The default value of the SHARED_MEMORY_STATIC_SIZE property has changed to 800M (838,860,800 bytes). This property specifies the size of the Shared Static Area (SSA) and cannot be modified using the ALTER SESSION or ALTER SYSTEM statements.

For more information, refer to **SHARED_MEMORY_STATIC_SIZE.**

Rename of INST_TABLE_PAGE_SIZE

The INST_TABLE_BLOCK_SIZE property has been renamed to INST_TABLE_PAGE_SIZE. The previous nam e remains supported as an alias. This property determines the page size of instant tables. If the fixed porti on of an instant record exceeds the specified page size, a maximum record length error may occur.

For more information, refer to **INST_TABLE_PAGE_SIZE**.

Separation of Cluster Commit Communication

The cluster commit protocol has been separated to use a dedicated dispatcher and queue. As a result, the existing CLUSTER_COMMIT_STREAM_ISOLATION property has been deprecated..

The operational status can be monitored through the V$CLUSTER_DISPATCHER view, which provides dis patcher types and send/receive statistics. The TYPE column includes a COMMIT entry. The list of cluster q ueues and their queued item counts can be viewed in V$CLUSTER_QUEUE.

For more information, refer to **V$CLUSTER_DISPATCHER**, **V$CLUSTER_QUEUE**.

Changes to Control File Temporary File Configuration

The control file storage mechanism has been changed so that temporary files are no longer used. As a res ult, the CONTROL_FILE_TEMP_NAME property has been deprecated. This property was previously used to specify the temporary directory and file name for storing a copy of the control file.

For more information, refer to **CONTROL_FILE_TEMP_NAME**.

Renaming Online DDL Properties

The names of properties related to online DDL have been changed as follows. The previous property nam es remain supported as aliases for backward compatibility.

Previous Name	New Name	Description
REBALANCE_BLO CK_READ_COUNT	ONLINE_DDL_BLOCK_R EAD_COUNT	Specifies the number of records to include in each block, the unit of remote transfer.

Previous Name	New Name	Description
REBALANCE_SHARD_DIVISOR	ONLINE_DDL_SCAN_PARTITION	Specifies the number of shards to divide for synchronization during online DDL execution.
MAXIMUM_JOURNAL_REPLAY_COUNT	ONLINE_DDL_MAXIMUM_JOURNAL_REPLAY_COUNT	Specifies the maximum number of replay iterations for journal logs generated by DML operations.
ONLINE_JOURNAL_REPLAY_THRESHOLD	ONLINE_DDL_JOURNAL_REPLAY_THRESHOLD	Specifies the threshold for acquiring an EXCLUSIVE lock and transitioning to the final replay phase when the amount of remaining journal logs is small.

For more information, refer to **ONLINE_DDL_BLOCK_READ_COUNT**, **ONLINE_DDL_SCAN_PARTITION**, **ONLINE_DDL_MAXIMUM_JOURNAL_REPLAY_COUNT**, **ONLINE_DDL_JOURNAL_REPLAY_THRESHOLD**.

3.2 SQL and PSM

Restriction on Concurrent Execution of ALTER INDEX COALESCE

ALTER INDEX COALESCE can no longer be executed concurrently with other ALTER statements. If administrative tasks were previously executed concurrently in version 22c.1, their execution order must be adjusted.

COALESCE reclaims space by merging adjacent index leaf pages. Concurrent execution with DML operations continues to be supported as before. This restriction applies only to concurrent execution with other ALTER statements.

For more information, refer to **ALTER INDEX name COALESCE**.

Removal of the ADD CONSTRAINT ON SCHEMA Privilege

The ADD CONSTRAINT ON SCHEMA privilege has been removed. Table owners can add constraints using the ALTER ON TABLE privilege, which is automatically granted when CREATE TABLE is executed. Non-owners must have either the ALTER ON TABLE privilege on the target table or the ALTER TABLE ON SCHEMA privilege.

The owner of an added constraint is the table owner.

For more information, refer to **ALTER TABLE name ADD CONSTRAINT**.

Default Schema Path Change for New Users

PUBLIC has been excluded from the default user schema path applied when a user is created using the CREATE USER statement. The default search path now includes only the user's own schema. To use the same search path as in version 22c.1 and earlier, PUBLIC must be explicitly specified after the user's schema.

```
ALTER USER u1 SCHEMA PATH (u1, public);
```

Object references that omit the schema name are resolved based on the schema path. However, the default schema path for users created with the WITHOUT SCHEMA option is set to PUBLIC.

For more information, refer to **Specifying a Schema in CREATE USER** and **Changing the Schema Path**.

Renaming the Shard Divisor Option in Online DDL

The shard divisor option used in online DDL has been renamed to scan partition. Existing DDL scripts mus t be updated to use the current syntax, SCAN PARTITION integer. This option specifies the number of part itions into which shards are divided for relocation and synchronization. If omitted, the value of ONLINE_D DL_SCAN_PARTITION is used.
The DDL statements affected are as follows.

- ALTER DATABASE MOVE SHARD
- ALTER DATABASE REBALANCE
- ALTER DATABASE REBALANCE EXCLUDE CLUSTER GROUP
- ALTER DATABASE SYNCHRONIZE
- ALTER TABLE name MOVE SHARD
- ALTER TABLE name REBALANCE
- ALTER TABLE name REBALANCE EXCLUDE CLUSTER GROUP cluster_group_list
- ALTER TABLE name SYNCHRONIZE

Removal of MINSIZE Support in STORAGE

MINSIZE is no longer supported in the STORAGE clause of the following DDL statements. When upgradin g existing DDL scripts, this option must be removed.
The supported segment attributes for tables are INITIAL, NEXT, and MAXSIZE, while those for indexes are INITIAL and NEXT.
The affected DDL statements are as follows.

- CREATE TABLE
- CREATE TABLE AS SELECT
- CREATE INDEX
- ALTER INDEX name STORAGE
- ALTER INDEX name REBUILD
- ALTER TABLE name ADD GLOBAL SECONDARY INDEX
- ALTER TABLE name ALTER GLOBAL SECONDARY INDEX
- ALTER TABLE name ALTER GLOBAL SECONDARY INDEX REBUILD

Removal of MAXSIZE Support in Index STORAGE

MAXSIZE is no longer supported in the STORAGE clause of DDL statements. Accordingly, this option mus t be removed from existing scripts that create, modify, or rebuild indexes and global secondary indexes.
The supported segment attributes are INITIAL and NEXT.
The affected DDL statements are as follows.

- **CREATE INDEX**
- **ALTER INDEX name STORAGE**
- **ALTER INDEX name REBUILD**
- **ALTER TABLE name ADD GLOBAL SECONDARY INDEX**
- **ALTER TABLE name ALTER GLOBAL SECONDARY INDEX**
- **ALTER TABLE name ALTER GLOBAL SECONDARY INDEX REBUILD**

Restriction on Concurrent Execution of GSI COALESCE

The ALTER TABLE ALTER GLOBAL SECONDARY INDEX COALESCE statement can no longer be executed c oncurrently with other ALTER statements. Accordingly, if administrative tasks were previously executed c oncurrently in version 22c.1, their execution order must be adjusted.

This statement performs a COALESCE operation on the global secondary index of the specified table to el iminate index fragmentation. The target table name must be specified when executing this operation. Th e complete syntax is ALTER TABLE name ALTER GLOBAL SECONDARY INDEX COALESCE.

For more information, refer to **ALTER TABLE name ALTER GLOBAL SECONDARY INDEX COALESCE**.

Changes to Built-in Package Installation

The installation method for some built-in packages provided in version 22c.1 and earlier has changed. SQ L scripts are now provided so that users can install the required packages manually.
To continue using packages from previous versions, the SQL scripts for the required packages must be ins talled.
Each installation file is located in $GOLDILOCKS_HOME/admin/packages/. Connect to gsql with SYSDBA privileges and execute the script using the --import option.

Package	Installation File
DBMS_LOCK	DBMS_LOCK.sql
DBMS_OUTPUT	DBMS_OUTPUT.sql

Package	Installation File
DBMS_SQL	DBMS_SQL.sql
DBMS_STANDARD	DBMS_STANDARD.sql

For more information, refer to **Installing DBMS_LOCK**, **Installing DBMS_OUTPUT**, **Installing DBMS_SQL**, a nd **Installing DBMS_STANDARD**.

3.3 API and Development Environment

Renaming the ODBC ALTERNATE_LOCATOR Connection Attribute

The ODBC connection attribute has been renamed from ALTERNATE_LOCATORS to ALTERNATE_LOCATOR. This attribute specifies a locator that provides alternative connection information when the glocator does not respond.
If the existing configuration is used, the attribute name in the locator DSN section of the odbc.ini file must be changed to the following format: ALTERNATE_LOCATOR=HOST=address:PORT=port

For more information, refer to **ODBC: locator DSN Keyword**.

Separation of Field Roles in SQL_LONG_VARIABLE_LENGTH_STRUCT for ODBC

The field layout of SQL_LONG_VARIABLE_LENGTH_STRUCT, which is used with non-standard ODBC data types, has been changed. buf_len specifies the capacity of the arr buffer, while len specifies the actual data length. Both values are expressed in bytes.
This structure is used with SQL_C_LONGVARCHAR and SQL_C_LONGVARBINARY. The application must allocate a buffer, set arr and buf_len, populate the buffer with input data, and then set len to the actual data length before binding.

For more information, refer to **ODBC: SQL_C_LONGVARCHAR Binding Example**.

Renaming the JDBC ALTERNATE_LOCATOR Connection Property

The JDBC connection property ALTERNATE_LOCATORS has been renamed to ALTERNATE_LOCATOR. When configuring JDBC properties, the lowercase form alternate_locator must be used, and the address of the alternative glocator must be specified in IP:PORT format.
For example, the property can be set in the Properties object passed when establishing a connection as follows: prop.setProperty("alternate_locator", "127.0.0.1:42582");

For more information, refer to **JDBC: Example of Setting the Locator Property**.

Renaming the Embedded SQL GET CLUSTER_GROUP_ID Statement

The Embedded SQL statement GET GROUPID has been renamed to GET CLUSTER_GROUP_ID. In a cluster environment that uses a global connection, the group ID of a target SQL statement can be obtained by specifying the target SQL statement after EXEC SQL GET CLUSTER_GROUP_ID INTO :group_id.
The target SQL statement must be a DELETE, INSERT, SELECT, or UPDATE statement on a table with a shard key defined. The :group_id variable must use a signed numeric type.
The target SQL statement is not actually executed; instead, it is cached in a prepared state. An invalid ID value of -1 may be returned.

For more information, refer to **Embedded SQL: EXEC SQL GET CLUSTER_GROUP_ID INTO**.

Separation of Field Roles in the Embedded SQL LONG VARCHAR Structure

The field layout of SQL_LONG_VARIABLE_LENGTH_STRUCT, which represents LONG VARCHAR data in Embedded SQL, has changed. The buf_len field specifies the capacity of the arr buffer, while the len field specifies the actual data length. Both values are measured in bytes.
The application must allocate memory for arr and set buf_len, then set len to the actual length of the data. The application is responsible for freeing the allocated memory after use.

For more information, refer to **Embedded SQL: LONG VARCHAR**.

Changes to Negative Length Error Handling for Embedded SQL VARCHAR and VARBINARY

Changes to Negative Length Error Handling for Embedded SQL VARCHAR and VARBINARY
When an Embedded SQL VARCHAR or VARBINARY variable is used as an IN or IN_OUT parameter, an error is now raised if its len value is negative. Therefore, data must be stored in arr before input, and len must be set to the exact length of the actual data.
In the manual examples, strlen(varstr.arr) is used to set the length of string data, while the number of bytes copied is assigned to len for binary data.

For more information, refer to Embedded SQL: **VARCHAR Constraints** and **VARBINARY Constraints**.

Changes to PyDBC Python 3 Build and Packaging

The PyDBC Python 3 package now uses a pyproject.toml-based build system and supports wheel and sdis t packaging.
PyDBC is a C extension module that uses the GOLDILOCKS ODBC library, gdlcs. On Linux, the package ca n be installed by running *python3 -m pip install .* in the *$GOLDILOCKS_HOME/app_dev/pygoldilocks/ver3* directory. The major and minor versions of Python used for installation must match those used to run PyD BC.

For more information, refer to **PyDBC: Python 3 Installation**.

Clarification of the Unsupported Value of cursor.lastrowid in PyDBC

The cursor.lastrowid attribute in PyDBC is not supported and always returns None. Therefore, the identifi er of an inserted row cannot be retrieved using this attribute.

For more information, refer to **PyDBC: Cursor**.

Changes to Time Zone Type Return Behavior in PyDBC Python 3

In PyDBC Python 3, the TIME WITH TIME ZONE and TIMESTAMP WITH TIME ZONE data types are now ret urned as timezone-aware objects that preserve the UTC offset. The corresponding Python types are dateti me.time and datetime.datetime, respectively.

For more information, refer to **PyDBC: SQL Value Conversion in Python 3**.

Time Zone Types Returned as Strings in PyDBC Python 2

In PyDBC Python 2, the TIME WITH TIME ZONE and TIMESTAMP WITH TIME ZONE data types are returne d as strings. This differs from Python 3, which returns these types as datetime.time and datetime.datetim e objects, respectively. Therefore, code that supports both Python versions must handle the return types differently.

For more information, refer to **PyDBC: SQL Value Conversion in Python 2**.

Improved Fractional Second Conversion for PyDBC TIME and TIMESTAMP Types

PyDBC has improved the handling of fractional second and microsecond conversions for the TIME and TIMESTAMP data types. These SQL types are returned as datetime.time and datetime.datetime objects in Python, respectively. The fractional second precision for both SQL types ranges from 0 to 6 digits, with a default precision of 6 digits.

For more information, refer to PyDBC: **SQL Value Conversion**, **TIME Precision**, and **TIMESTAMP Precision**.

Improved Decimal Precision and Scale Handling and Restrictions on Non-Finite Values in PyDBC

PyDBC has improved the handling of precision and scale for the NUMBER, NUMERIC, and DECIMAL data types. Decimal values, such as NaN and Infinity, are not supported. PyDBC passes Python decimal input values to ODBC as SQL_NUMERIC structures and returns retrieved SQL_NUMERIC values as decimal.Decimal objects.

For more information, refer to PyDBC: **Parameter Passing** and **SQL Value Conversion**.

SQLAlchemy 1.4 and 2.0 Support

SQLAlchemy versions 1.4 and 2.0 are supported. To use the GOLDILOCKS dialect, Python 3.8 or later, alembic, and the pygoldilocks package are required. After installing the dialect, connect by passing a URL in the format goldilocks://username:password@host:port to the create_engine() function.

For more information, refer to **SQLAlchemy: Requirements**.

Hibernate 6, 7, and 8 Support

Hibernate versions 6, 7, and 8 are supported.

For more information, refer to **Hibernate: Interworking with GoldilocksDialect Class**.

3.4 Utility

Deprecation of the directio-size Option in gloader/gloadernet

The --directio-size option has been deprecated. Check whether this option is used in existing execution commands and scripts.

redo_log File Type Name in gdump

The log file type name in gdump has been changed to redo_log. The file type is specified before the path of the file to be dumped. Use redo_log to output online or archived redo log files in text format.

For more information, refer to **redo_log File Type Name** in gdump.

datafile File Type Name in gdump

The data file type name in gdump has been changed to datafile. When dumping a data file, specify it in the format gdump datafile file path. The existing options are still available: --header outputs header information, while --number and --fetch specify the starting page number and the number of pages to output, respectively.

For more information, refer to **datafile File Type Name in gdump**.

ALTERNATE_LOCATOR Configuration Property Name in glocator

The glocator configuration property name has been changed from ALTERNATE_LOCATORS to ALTERNATE_LOCATOR. Specify the name of the peer locator used for replication, and define HOST and PORT in the corresponding name section of the same configuration file.

For more information, refer to **ALTERNATE_LOCATOR Configuration Property Name** in glocator.

glocator Replication Configuration

The glocator configuration has changed from a multi-instance configuration to a replication configuratio n. The two glocator instances are classified as master and sub (substitute) according to their startup order. Use the --sync option to connect the sub instance, which starts later, to the master instance, which starts first. During synchronization, the data from both glocator instances is merged. If duplicate entries have di fferent values, the most recently updated value overwrites the other.

For more information, refer to glocator **Replication Configuration**.

Removal of the Argument from the glocator sync Option

The argument previously specified for the --sync option in glocator has been removed. glocator is now sta rted using the glocator --start --sync command. The ALTERNATE_LOCATOR property must be configured in the configuration file in advance.

For more information, refer to **Removal of the Argument from the glocator sync Option**.

ALTERNATE_LOCATOR Configuration Property Name in gage nt

The gagent configuration property name has been changed from ALTERNATE_LOCATORS to ALTERNATE _LOCATOR. Specify the name of the alternate locator in the [AGENT] section, and define HOST and PORT in the corresponding name section.

For more information, refer to **ALTERNATE_LOCATOR Configuration Property Name** in gagent.