
MySQL 8.0 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.5. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 8.0 documentation, see the [MySQL 8.0 Reference Manual](#), which includes an overview of features added in MySQL 8.0 ([What Is New in MySQL 8.0](#)), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.7 to MySQL 8.0 ([Changes Affecting Upgrades to MySQL 8.0](#)).

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (<http://dev.mysql.com/downloads/>), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit either the [MySQL Forums](#) or [MySQL Mailing Lists](#), where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the [MySQL Documentation Library](#).

Document generated on: 2017-12-12 (revision: 13482)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL 8.0.5 (Not yet released, Release Candidate)	3
Changes in MySQL 8.0.4 (Not yet released, Release Candidate)	3
Changes in MySQL 8.0.3 (2017-09-21, Release Candidate)	3
Changes in MySQL 8.0.2 (2017-07-17, Development Milestone)	24
Changes in MySQL 8.0.1 (2017-04-10, Development Milestone)	48
Changes in MySQL 8.0.0 (2016-09-12, Development Milestone)	90
Index	121

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.5.

Legal Notices

Copyright © 1997, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any

means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Changes in MySQL 8.0.5 (Not yet released, Release Candidate)



Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.

Version 8.0.5 has no changelog entries, or they have not been published because the product version has not been released.

Changes in MySQL 8.0.4 (Not yet released, Release Candidate)



Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.

Version 8.0.4 has no changelog entries, or they have not been published because the product version has not been released.

Changes in MySQL 8.0.3 (2017-09-21, Release Candidate)



Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.

- [Account Management Notes](#)

- [Atomic DDL Notes](#)
- [C API Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Configuration Notes](#)
- [Data Dictionary Notes](#)
- [Deprecation and Removal Notes](#)
- [InnoDB Notes](#)
- [Optimizer Notes](#)
- [Packaging Notes](#)
- [Performance Schema Notes](#)
- [Security Notes](#)
- [Server Administration](#)
- [Spatial Data Support](#)
- [SQL Syntax Notes](#)
- [X Plugin Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Account Management Notes

- `CREATE USER` now permits a `DEFAULT ROLE` clause enabling the account default roles to be specified.
`SHOW CREATE USER` now displays the account default roles if the default is not `NONE`. (Bug #24670738, Bug #82987)
- MySQL now maintains information about password history, which makes it possible to enable restrictions on reuse of previous passwords. DBAs can require that new passwords not be selected from previous passwords for some number of password changes or period of time. It is possible to establish password-reuse policy globally using the `password_history` and `password_reuse_interval` system variables, as well as on a per-account basis using the `CREATE USER` and `ALTER USER` statements. Together with existing password-expiration capabilities to require that passwords be changed periodically, the new reuse-restriction capabilities provide DBAs more complete control over password management. For more information, see [Password Management](#).



Important

MySQL implements password-reuse restrictions by means of new columns in the `mysql.user` system table and a new `mysql.password_history` system table. If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these system database changes. Until this is done, *password changes are not possible*.

Atomic DDL Notes

- MySQL now supports atomic data definition statements (*atomic DDL*). An atomic DDL statement combines the data dictionary updates, storage engine operations, and binary log writes associated with a DDL operation into a single, crash-safe, transaction that is either fully committed or rolled back.

Both table and non-table DDL statements are supported. Table-related DDL operations require storage engine support, whereas non-table DDL operations do not. Currently, the [InnoDB](#) storage engine supports atomic DDL.

- Supported table DDL statements include [CREATE](#), [ALTER](#), and [DROP](#) statements for databases, tablespaces, tables, and indexes, and the [TRUNCATE TABLE](#) statement.
- Supported non-table DDL statements include:
 - [CREATE](#) and [DROP](#) statements, and, if applicable, [ALTER](#) statements for stored programs, triggers, views, and user-defined functions (UDFs). Atomic DDL support for [CREATE TRIGGER](#) and [DROP TRIGGER](#) was added in MySQL 8.0.0.
 - Account management statements: [CREATE](#), [ALTER](#), [DROP](#), and, if applicable, [RENAME](#) statements for users and roles, as well as [GRANT](#) and [REVOKE](#) statements. Atomic DDL support for account management statements was added in MySQL 8.0.1.

For table-related DDL operations, [InnoDB](#) writes DDL logs to the `mysql.innodb_ddl_log` data dictionary table. Enabling the `innodb_print_ddl_logs` configuration option prints DDL recovery logs to `stderr`.

The atomic DDL feature changes the behavior of some statements:

- [DROP VIEW](#) fails with an error if a named view does not exist, and no changes are made. Previously, the statement returned an error indicating which views did not exist, but also dropped the views that did exist.
- [DROP TABLE](#) fails with an error if a named table does not exist, and no changes are made. Previously, the statement returned an error indicating which tables did not exist, but also dropped the tables that did exist.
- [DROP TABLE](#) is fully atomic if all named tables use an atomic DDL-supported storage engine.
- [DROP DATABASE](#) is atomic if all tables use an atomic DDL-supported storage engine. However, removal of the database directory from the file system occurs last and is not part of the atomic transaction. If removal of the database directory fails due to a file system error or server halt, the [DROP DATABASE](#) transaction is not rolled back.
- Interrupted DDL operations on tables that use an atomic DDL-supported storage engine no longer introduce discrepancies between the storage engine, data dictionary, and binary log, or leave behind orphan files.
- Partial execution of account management statements is no longer permitted. Account management statements either succeed for all named users or roll back and have no effect if an error occurs.

Changes to [DROP TABLE](#), [DROP VIEW](#), and account management statement behavior have implications for cross-version replication configurations.

For more information, see [Atomic Data Definition Statement Support](#). (Bug #24620918)

C API Notes

- The MySQL C API now enables clients to specify that metadata transfer for result sets is optional. Suppression of metadata transfer can improve performance, particularly for sessions that execute many queries that return few rows each. For more information, see [C API Optional Result Set Metadata](#).

Character Set Support

- MySQL now supports Russian collations for the `utf8mb4` Unicode character set:
 - `utf8mb4_ru_0900_ai_ci` is accent insensitive and case insensitive.
 - `utf8mb4_ru_0900_as_cs` is accent sensitive and case sensitive.

Compilation Notes

- For debug builds, the `SAFE_MUTEX` compilation flag was disabled if the `memcached` plugin was included in the build. This no longer occurs; `SAFE_MUTEX` is always enabled for debug builds. Some code issues found as a result of this change were corrected. (Bug #26442367, Bug #87068)
- Binary packages on EL6 and EL7 now are compiled using Devtoolset 6 rather than Devtoolset3 and GCC 6.2.1 rather than 4.9.2. (Bug #26436968, Bug #87061)
- MySQL now compiles for SPARC on Oracle Linux. (Bug #26306331, Bug #86745)
- MySQL compilation on macOS using Clang now requires a Clang version different from 8.0, which has problems with certain inline constructs. (Bug #26279510, Bug #86711)
- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

Configuration Notes

- The `performance_schema_max_mutex_classes` system variable default value was increased from 220 to 250. The `performance_schema_max_thread_classes` system variable default value was increased from 50 to 100. (Bug #26193630)
- The new `cte_max_recursion_depth` system variable implements a common table expression (CTE) maximum recursion depth. The server terminates execution of any CTE that recurses more levels than the value of this variable. For more information, see [Limiting Common Table Expression Recursion](#). (Bug #26136509, Bug #86444)
- The `back_log` system variable default value is now the value of `max_connections`, which enables the permitted backlog to adjust to the maximum permitted number of connections.
- To enable the Event Scheduler by default, the `event_scheduler` system variable default value was changed from `OFF` to `ON`.
- The `max_allowed_packet` system variable default value was increased from 4194304 (4M) to 67108864 (64M).
- The `max_error_count` system variable default variable was increased from 64 to 1024.

Data Dictionary Notes

- These `INFORMATION_SCHEMA` tables have been reimplemented as views on data dictionary tables:
-

FILES
PARTITIONS
REFERENTIAL_CONSTRAINTS

Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. For example, the server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.

Also as a result of this change, foreign key information is retrieved from the data dictionary, not from InnoDB.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes. (Bug #25583288)

Deprecation and Removal Notes

- **Replication:** The deprecated global scope for the `sql_log_bin` system variable has been removed. `sql_log_bin` now has session scope only. Applications that rely on accessing `@global.sql_log_bin` should be adjusted.
- These encryption-related deprecated items have been removed:
 - The `ENCODE()` and `DECODE()` functions.
 - The `ENCRYPT()` function.
 - The `DES_ENCRYPT()`, and `DES_DECRYPT()` functions, the `--des-key-file` option, the `have_crypt` system variable, the `DES_KEY_FILE` option for the `FLUSH` statement, and the `HAVE_CRYPT CMake` option.

In place of the removed encryption functions: For `ENCRYPT()`, consider using `SHA2()` instead for one-way hashing. For the others, consider using `AES_ENCRYPT()` and `AES_DECRYPT()` instead. (Bug #26493987)

- The deprecated `tx_isolation` and `tx_read_only` system variables have been removed. Use `transaction_isolation` and `transaction_read_only` instead.
- The deprecated query cache has been removed. Removal includes these items:
 - The `FLUSH QUERY CACHE` and `RESET QUERY CACHE` statements.
 - These system variables: `query_cache_limit`, `query_cache_min_res_unit`, `query_cache_size`, `query_cache_type`, `query_cache_wlock_invalidate`.
 - These status variables: `Qcache_free_blocks`, `Qcache_free_memory`, `Qcache_hits`, `Qcache_inserts`, `Qcache_lowmem_prunes`, `Qcache_not_cached`, `Qcache_queries_in_cache`, `Qcache_total_blocks`.
 - These thread states: `checking privileges on cached query`, `checking query cache for query`, `invalidating query cache entries`, `sending cached result to client`, `storing result in query cache`, `Waiting for query cache lock`.

These deprecated query cache items remain deprecated, but have no effect, and will be removed in a future MySQL release:

- The `SQL_CACHE` and `SQL_NO_CACHE SELECT` modifiers.
- The `ndb_cache_check_time` system variable.

The `have_query_cache` system variable remains deprecated, always has a value of `NO`, and will be removed in a future MySQL release.

- The deprecated `EXTENDED` and `PARTITIONS` keywords for the `EXPLAIN` statement have been removed. These keywords are unnecessary because their effect is always enabled.
- The unused `date_format`, `datetime_format`, `time_format`, and `max_tmp_tables` system variables have been removed.
- The deprecated `multi_range_count` system variable has been removed.
- The deprecated `log_warnings` system variable and `--log-warnings` server option have been removed. Use the `log_error_verbosity` system variable instead.
- The deprecated `secure_auth` system variable and `--secure-auth` client option have been removed. The `MYSQL_SECURE_AUTH` option for the `mysql_options()` C API function was removed.
- The deprecated `--ignore-builtin-innodb` server option and `ignore_builtin_innodb` system variable have been removed.

InnoDB Notes

- Renaming of columns in a parent foreign key is temporarily disabled due to ongoing work on foreign key locking. This restriction will be lifted in MySQL 8.0.4. (Bug #26334071)

References: See also: Bug #26659110.

Optimizer Notes

- The optimizer now supports a `SET_VAR` hint that sets the session value of a system variable for the duration of a single statement. Examples:

```
SELECT /*+ SET_VAR(sort_buffer_size = 16M) */ name FROM people ORDER BY name;  
INSERT /*+ SET_VAR(foreign_key_checks=OFF) */ INTO t2 VALUES(2);
```

For more information, see [Optimizer Hints](#). (Bug #22906815)

- The optimizer now uses column-value histogram statistics stored in the `column_statistics` data dictionary table to construct query execution plans. Histogram use applies to predicates involving comparison of a column to a constant. See [Optimizer Statistics](#).
- Previously, there was no way of skipping the use of index dives to estimate index usefulness, except by using the `eq_range_index_dive_limit` system variable. Now index dive skipping is possible for single-table queries under certain query conditions (see [Range Optimization](#)).
- The `optimizer_switch` system variable has a new flag named `use_invisible_indexes` to control whether the optimizer uses invisible indexes for query execution plan construction. If the flag is `off` (the default), the optimizer ignores invisible indexes (the same behavior as prior to the introduction of this flag). If the flag is `on`, invisible indexes remain invisible but the optimizer takes them into account for execution plan construction.

Packaging Notes

- `mysqlcheck` was missing in the MySQL Server Docker image, which prevented `mysql_upgrade` from running. (Bug #26400146, Bug #86968)
- For Debian, non-debug binaries were moved from the `mysql-server` package to the `mysql-server-core` package. (Bug #26382333, Bug #86899)

- The Debian/Ubuntu `mysql-community-source` package is no longer produced because the MySQL source tarball it contained is provided by other packages at dev.mysql.com. (Bug #26201482)

Performance Schema Notes

- As of MySQL 8.0.2, Performance Schema table definitions are maintained internally to the server. In consequence of that change, `CREATE TABLE` and `DROP TABLE` are no longer possible for Performance Schema tables. (Bug #26136994)
- The `events_statements_summary_by_digest` table now provides, for each row, a sample statement that produces the digest value in the row. Applications can use this information as a more efficient means of capturing statement samples than alternatives such as probing the `xxx_history_long` tables. The latter approach requires enabling the corresponding `xxx_history_long` consumers, which is additional overhead for applications that do not otherwise need those tables. For more information, see [Performance Schema Statement Digests and Sampling](#), and [Statement Summary Tables](#).

Additionally, the `FIRST_SEEN` and `LAST_SEEN` timestamp columns of the `events_statements_summary_by_digest` table now have a fractional seconds part.

- The Performance Schema `setup_instruments` table now has columns for instrument metadata: Instrument properties, instrument volatility, and a documentation string describing the the instrument purpose. Also, the `TIMED` column now can be `NULL`, indicating that the instrument does not support timing. See [The setup_instruments Table](#).

The new Performance Schema `setup_threads` table exposes instrumented thread class names and attributes. See [The setup_threads Table](#).

Security Notes

- A new `caching_sha2_password` authentication plugin is available. Like the `sha256_password` plugin, `caching_sha2_password` implements SHA-256 password hashing, but uses caching to address latency issues at connect time. It also supports more connection protocols and does not require linking against OpenSSL for RSA password-exchange capabilities. See [SHA-2 Pluggable Authentication](#).
- If the server is started with the `--skip-grant-tables` option to disable authentication checks, the server now enables `--skip-networking` to prevent remote connections.

Server Administration

- MySQL now supports creation and management of resource groups, and permits assigning threads running within the server to particular groups so that threads execute according to the resources available to the group. Group attributes enable control over its resources, to enable or restrict resource consumption by threads in the group. DBAs can modify these attributes as appropriate for different workloads. Currently, CPU time is a manageable resource, represented by the concept of “virtual CPU” as a term that includes CPU cores, hyperthreads, hardware threads, and so forth. The server determines at startup how many virtual CPUs are available, and database administrators with appropriate privileges can associate these CPUs with resource groups and assign threads to groups. For more information, see [Resource Groups](#).

Spatial Data Support

- **Incompatible Change:** Previously, these functions that test geometry relationships supported only Cartesian spatial reference systems (SRs): `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`, `MBRContains()`, `MBRCoveredBy()`, `MBRCovers()`, `MBRDisjoint()`, `MBREquals()`, `MBRIntersects()`, `MBROverlaps()`, `MBRTouches()`, `MBRWithin()`.

These functions now detect geometry arguments in a geographic (ellipsoidal) SRS and return geographic results. Calculations for projected SRSs and SRID 0 remain the same. For more information, see [Spatial Relation Functions That Use Object Shapes](#), and [Spatial Relation Functions That Use Minimum Bounding Rectangles](#).



Note

If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- **InnoDB:** Spatial reference identifier (SRID) support was added for [InnoDB](#) spatial indexes.
- Spatial data types now permit an [SRID](#) attribute, to explicitly indicate the spatial reference system (SRS) for values stored in the column. See [Spatial Data Types](#).

To indicate each column's [SRID](#) attribute value, if there is one, the [INFORMATION_SCHEMA.COLUMNS](#) table now has an [SRS_ID](#) column.

A spatial column with an explicit [SRID](#) attribute is SRID-restricted: The column takes only values with that ID, and [SPATIAL](#) indexes on the column become subject to use by the optimizer. The optimizer ignores [SPATIAL](#) indexes on spatial columns with no [SRID](#) attribute. See [SPATIAL Index Optimization](#).

SQL Syntax Notes

- [ALTER TABLE](#) now supports easier column renaming using [RENAME COLUMN *old_name* TO *new_name*](#) syntax. See [ALTER TABLE Syntax](#). For changing a column name but not its definition, [RENAME COLUMN](#) is more convenient than [CHANGE](#), which requires respecifying the current column definition. With [CHANGE](#), you must look up the definition if you do not know it, and if you do not respecify it exactly, there is a possibility of data change or loss. (Bug #11746522, Bug #26949, Bug #11747473, Bug #32497, Bug #11765084, Bug #58006, Bug #14031617)

X Plugin Notes

- The X Plugin could not be installed when the server was started with the [--skip-grant-tables](#) option. (Bug #26516678)
- X Plugin socket connections were not working correctly. (Bug #26427112, Bug #87019)
- When compiling MySQL from source, certain infrequently used [CMake](#) arguments caused issues for the X Plugin build. (Bug #26141933)
- A [Mysqlx.Connection.CapabilitiesGet](#) request using MySQL X Protocol did not return the complete list of available authentication mechanisms. (Bug #26044113)
- For mixed case or uppercase schema names, the statement [list_objects](#) could incorrectly report a collection as a table. (Bug #25769683)
- The X Plugin was omitted from the list of plugins to include for testing data directory permissions. (Bug #24823999)

Functionality Added or Changed

- **InnoDB:** The new [innodb_dedicated_server](#) configuration option, which is disabled by default, causes [InnoDB](#) to automatically configure the following options according to the amount of memory detected on the server:
 - [innodb_buffer_pool_size](#)

- `innodb_log_file_size`
- `innodb_flush_method`

This option is intended for MySQL server instances that run on a dedicated server. For more information, see [Enabling Automatic Configuration for a Dedicated MySQL Server](#).

- **InnoDB:** Renaming a general tablespace is now supported by `ALTER TABLESPACE ... RENAME TO` syntax.

The `ALTER TABLESPACE` and `DROP TABLESPACE ENGINE` clause is deprecated and will be removed in a future release.

- **InnoDB:** Code related to obsoleted InnoDB system tables was removed. `INFORMATION_SCHEMA` views based on InnoDB system tables were replaced by internal system views on data dictionary tables. Affected InnoDB `INFORMATION_SCHEMA` views were renamed:

Table 1 Renamed InnoDB Information Schema Views

Old Name	New Name
<code>INNODB_SYS_COLUMNS</code>	<code>INNODB_COLUMNS</code>
<code>INNODB_SYS_DATAFILES</code>	<code>INNODB_DATAFILES</code>
<code>INNODB_SYS_FIELDS</code>	<code>INNODB_FIELDS</code>
<code>INNODB_SYS_FOREIGN</code>	<code>INNODB_FOREIGN</code>
<code>INNODB_SYS_FOREIGN_COLS</code>	<code>INNODB_FOREIGN_COLS</code>
<code>INNODB_SYS_INDEXES</code>	<code>INNODB_INDEXES</code>
<code>INNODB_SYS_TABLES</code>	<code>INNODB_TABLES</code>
<code>INNODB_SYS_TABLESPACES</code>	<code>INNODB_TABLESPACES</code>
<code>INNODB_SYS_TABLESTATS</code>	<code>INNODB_TABLESTATS</code>
<code>INNODB_SYS_VIRTUAL</code>	<code>INNODB_VIRTUAL</code>

After upgrading to MySQL 8.0.3 or later, update any scripts that reference previous InnoDB `INFORMATION_SCHEMA` view names.

`INNODB_TABLESPACES_BRIEF`, a new `INFORMATION_SCHEMA` view, provides space, name, path, flag, and space type data for InnoDB tablespaces.

- **InnoDB:** When InnoDB was integrated with the global data dictionary, file-per-table tablespace names in the data dictionary were created in the form of `innodb_file_per_table_x`, where `x` is the InnoDB tablespace ID. For ease of use, file-per-table tablespace names in the data dictionary are once again the same as the table name.

Upgrade from MySQL 5.7 to MySQL 8.0 appends MySQL 5.7 `innodb_table_stats` and `innodb_index_stats` tablespace names in the data dictionary with “`_backup57`” to differentiate them from their MySQL 8.0 counterparts.

- **InnoDB:** The default `innodb_autoinc_lock_mode` setting was changed from 1 (consecutive) to 2 (interleaved). Interleaved lock mode permits the execution of multi-row inserts in parallel, which improves concurrency and scalability. The new `innodb_autoinc_lock_mode` default setting reflects the change from statement-based replication to row based replication as the default replication type in MySQL 5.7. Statement-based replication requires the consecutive auto-increment lock mode to ensure that auto-increment values are assigned in a predictable and repeatable order for a given sequence of SQL

statements, whereas row-based replication is not sensitive to the execution order of SQL statements. For more information, see [InnoDB AUTO_INCREMENT Lock Modes](#).

For systems that use statement-based replication, the new `innodb_autoinc_lock_mode` default setting may break applications that depend on sequential auto-increment values. To restore the previous default, set `innodb_autoinc_lock_mode` to 1.

- **InnoDB:** Serialized Dictionary Information (SDI) is now present in all [InnoDB](#) tablespace files except for temporary tablespace and undo tablespace files. SDI is serialized metadata for schema, table, and tablespace objects. The presence of SDI data provides metadata redundancy. For example, dictionary object metadata may be extracted from tablespace files if the data dictionary becomes unavailable. SDI extraction is performed using the `ibd2sdi` tool. SDI data is stored in [JSON](#) format.

The inclusion of SDI data in tablespace files increases tablespace file size. An SDI record requires a single index page, which is 16k in size by default. However, SDI data is compressed when it is stored to reduce the storage footprint.

- **InnoDB:** The `innodb_flush_neighbors` default value was changed from 1 to 0, which disables flushing of neighboring pages from the buffer pool. A setting of 0 is optimal for non-rotational storage (SSD) devices where seek time is not a significant factor. For systems that use rotational storage (HDD), it is recommended to change the setting back to the previous default value of 1.
- **InnoDB:** Default values for configuration options that affect buffer pool preflushing and flushing behavior were modified:
 - The `innodb_max_dirty_pages_pct_lwm` default value was changed to 10. The previous default value of 0 disables buffer pool preflushing. A value of 10 enables preflushing when the percentage of dirty pages in the buffer pool exceeds 10%. Enabling preflushing improves performance consistency.
 - The `innodb_max_dirty_pages_pct` default value was changed from 75 to 90. [InnoDB](#) attempts to flush data from the buffer pool so that the percentage of dirty pages does not exceed this value. The increased default value permits a greater percentage of dirty pages in the buffer pool.
- **InnoDB:** The minimum `innodb_undo_tablespaces` value changed from 0 to 2. In previous releases, the system tablespace is used for rollback segments if `innodb_undo_tablespaces` is set to 0. A minimum value of 2 ensures that rollback segments are created in undo tablespaces instead of the system tablespace. For more information, see [Configuring Undo Tablespaces](#).
- **Replication; JSON:** Added the `binlog_row_value_options` system variable. Currently this variable can be unset, or set to the value `PARTIAL_JSON`. This causes MySQL's row-based replication to use a compact binary log format for each update modifying only a small portion of a JSON document and using any combination of `JSON_SET()`, `JSON_REPLACE()`, and `JSON_REMOVE()`. The compact format includes only the modified parts of the JSON document, not the full document, in the after-image used for the update in the binary log. If the modification requires more space than the full document, or if it is not possible to generate a partial update, the full document is used instead.

See the description of the variable as well as [Partial Updates of JSON Values](#), for more information.

- **Replication:** The `IGNORE_SERVER_IDS` option of the `CHANGE MASTER TO` statement is now deprecated when using GTID-based replication (`gtid_mode=ON`). With GTIDs, transactions that have already been applied are automatically ignored, so this function is not needed.

Before starting GTID-based replication, check for and clear all ignored server ID lists that have previously been set on the servers involved. The `SHOW SLAVE STATUS` statement, which can be issued for individual channels, displays the list of ignored server IDs if there is one. If there is no list, the `Replicate_Ignore_Server_Ids` field is blank.

If `gtid_mode=ON` is set for the server, a deprecation warning is now issued if you include the `IGNORE_SERVER_IDS` option in a `CHANGE MASTER TO` statement. A deprecation warning is also issued if you issue a `SET GTID_MODE=ON` statement when any channel has existing server IDs set with `IGNORE_SERVER_IDS`. If you do receive the deprecation warning, you can still clear a list after `gtid_mode=ON` is set by issuing a `CHANGE MASTER TO` statement containing the `IGNORE_SERVER_IDS` option with an empty list.

- **Replication:** The following obsolete `mysqlbinlog` options are now deprecated and will be removed in a future release:

- `--short-form`, which could be used for testing to limit the output to statements alone.
- `--stop-never-slave-server-id`, which provided a server ID for connections using the `--stop-never` option. If you require this function, use the `--connection-server-id` option instead.

The deprecation warnings for these options are sent to standard error, rather than to standard output, so that they do not interfere with the operation of tools that use the output of `mysqlbinlog`.

- **Replication:** The `log_slave_updates` system variable is now set to ON by default, so you do not need to specify `--log-slave-updates` explicitly when you start a replication slave.

The `log_slave_updates` system variable is read-only. If you need to prevent a replication slave from logging the updates performed by its SQL thread to its own binary log, specify `--skip-log-slave-updates` when you start the slave, or specify `log_slave_updates=OFF` in the configuration file for the slave.

- **Replication:** The Group Replication thread states are now shown in the Performance Schema tables.
- **Replication:** The `group_replication_communication_debug_options` variable has been added which enables you to dynamically filter out debugging and tracing messages per Group Replication component, such as GCS, XCOM, and so on. (Bug #10200)
- **JSON:** The `JSON_MERGE()` function is renamed to `JSON_MERGE_PRESERVE()`.

This release also adds the `JSON_MERGE_PATCH()` function, an RFC 7396 compliant version of `JSON_MERGE_PRESERVE()`; its behavior is the same as that of `JSON_MERGE_PRESERVE()`, with the following two exceptions:

- `JSON_MERGE_PATCH()` removes any member in the first object with a matching key in the second object, provided that the value associated with the key in the second object is not JSON `null`.
- If the second object has a member with a key matching a member in the first object, `JSON_MERGE_PATCH()` *replaces* the value in the first object with the value in the second object, whereas `JSON_MERGE_PRESERVE()` *appends* the second value to the first value.

This example compares the results of merging the same 3 JSON objects, each having a matching key "a", with each of these functions:

```
mysql> SET @x = '{ "a": 1, "b": 2 }',
>         @y = '{ "a": 3, "c": 4 }',
>         @z = '{ "a": 5, "d": 6 }';

mysql> SELECT  JSON_MERGE_PATCH(@x, @y, @z)    AS Patch,
->             JSON_MERGE_PRESERVE(@x, @y, @z) AS Preserve\G
***** 1. row *****
Patch: { "a": 5, "b": 2, "c": 4, "d": 6 }
Preserve: { "a": [1, 3, 5], "b": 2, "c": 4, "d": 6 }
```

`JSON_MERGE()` is still supported as an alias of `JSON_MERGE_PRESERVE()`, but is now deprecated and subject to removal in a future MySQL release.

See [Functions That Modify JSON Values](#), for more information. (Bug #81283, Bug #23255346)

- **InnoDB** now uses Variance-Aware Transaction Scheduling (VATS) for scheduling the release of transaction locks when the system is highly loaded, which helps reduce lock sys wait mutex contention. Lock scheduling uses VATS when ≥ 32 threads are suspended in the lock wait queue.

For more information about Variance-Aware Transaction Scheduling (VATS), see [Identifying the Major Sources of Variance in Transaction Latencies: Towards More Predictable Databases](#).

Thanks to Jiamin Huang for the contribution. (Bug #25290971, Bug #84266)

- MySQL can now extend metadata locks, as necessary, to tables that are related by a foreign key constraint. The ability to extend metadata locks prevents conflicting DML and DDL operations from being executed concurrently on related tables. This feature also enables updates to foreign key metadata when a parent table is modified. Previously, the foreign key metadata, which is owned by the child table, could not be updated safely.
- The binary log is now enabled by default at server startup. The `log_bin` system variable is set to ON by default, instead of OFF, even if the `--log-bin` option has not been specified. Binary logging is standard practice for production installations, so enabling it by default removes configuration and planning steps that were usually required.

If you want to disable binary logging, you can specify the `--skip-log-bin` or `--disable-log-bin` option at startup.

The `server_id` system variable is now set to 1 by default, instead of 0. For servers in a replication topology, you must still change this setting to specify a unique server ID for each replication server. Previously, the server could not start with `log_bin = ON` if no server ID was specified. Now, the server can start, but a warning message is issued if you did not set an explicit server ID. (From MySQL 8.0.4, the message is only informational.)

With binary logging enabled for a server, all statements that change data are logged to the server's binary log, which is a sequence of files with a base name and numeric extension. By default, the server creates binary log files and an index file in the data directory. In MySQL 8.0.3, the default base name of these files is `host_name-bin`, using the name of the host machine. From MySQL 8.0.4, the default base name is `binlog` if you do not supply the `--log-bin` option, and `host_name-bin` if you supply the option with no string or an empty string.

You can choose the names and locations of the binary log files and index file by specifying the `--log-bin` and `--log-bin-index` options. You are recommended to specify a base name explicitly, so that if the host name changes, you can easily continue to use the same binary log file names. The `log_bin_basename` system variable holds the base name and any specified path for the binary log files.

The relay log and relay log index on a replication slave, whose names are specified by the `--relay-log` and `--relay-log-index` options, cannot be given the same names as the binary log and binary log index. From MySQL 8.0.3, the server issues an error message and does not start if the binary log and relay log file base names would be the same.

The server creates a new binary log file in the series each time it starts or flushes the logs. The server also creates a new binary log file automatically after the current file's size reaches `max_binlog_size`, which defaults to the maximum permitted value of 1GB. In MySQL 8.0.3, binary log files expire by default

after 30 days, and can then be automatically removed at startup or when the binary log is flushed. You can purge binary log files manually using the `PURGE BINARY LOGS` statement, or specify a different binary log expiration period using the `binlog_expire_logs_seconds` system variable.

Many other options are available to modify the behavior of binary logging. For more information, see [The Binary Log](#) and [Binary Logging Options and Variables](#).

References: See also: Bug #26730000.

- The `expire_logs_days` system variable, which specifies the binary log expiration period in days, is now deprecated and will be removed in a future release. `expire_logs_days` does not provide sufficient flexibility for defining the binary log expiration period.

`binlog_expire_logs_seconds` can be used to set the binary log expiration period in seconds. In MySQL 8.0.3, the effects of the two variables are currently cumulative. For example, if `expire_logs_days` is 1 and `binlog_expire_logs_seconds` is 43200, then the binary log is purged every 1.5 days. This produces the same result as setting `binlog_expire_logs_seconds` to 129600 and `expire_logs_days` to 0. Note that the default `expire_logs_days` setting of 30 days is currently added to the binary log expiration period if `expire_logs_days` is not specified. To use `binlog_expire_logs_seconds` alone, set `expire_logs_days=0` explicitly.

To disable automatic purging of the binary log, you must set both `expire_logs_days` and `binlog_expire_logs_seconds` explicitly to 0.

References: See also: Bug #26483363.

- A new type of backup lock permits DML during an online backup while preventing operations that could result in an inconsistent snapshot. The new backup lock is supported by `LOCK INSTANCE FOR BACKUP` and `UNLOCK INSTANCE` syntax. The `BACKUP_ADMIN` privilege is required to use these statements.
- The new `get_sysvar_source` plugin service enables plugins to retrieve the source of system variable settings.
- The zlib library version bundled with MySQL was raised from version 1.2.3 to version 1.2.11. MySQL implements compression with the help of the zlib library.

The zlib `compressBound()` function in zlib 1.2.11 returns a slightly higher estimate of the buffer size required to compress a given length of bytes than it did in zlib version 1.2.3. The `compressBound()` function is called by `InnoDB` functions that determine the maximum row size permitted when creating compressed `InnoDB` tables or inserting rows into compressed `InnoDB` tables. As a result, `CREATE TABLE ... ROW_FORMAT=COMPRESSED` or `INSERT` operations with row sizes very close to the maximum row size that were successful in earlier releases could now fail.

If you have compressed `InnoDB` tables with large rows, it is recommended that you test compressed table `CREATE TABLE` statements on a MySQL 8.0.3 test instance prior to upgrading.

- In MySQL 8.0.2, the system variables for the slave status logs, `master_info_repository` and `relay_log_info_repository`, were set to `TABLE` instead of `FILE` by default. In MySQL 8.0.3, the `FILE` setting for both these system variables is deprecated, and a warning is issued if it is used. The `FILE` setting will be removed in a future release.

The `TABLE` setting ensures that replication repository information is stored in `InnoDB` tables, rather than in files in the data directory. The use of tables makes replication resilient to unexpected halts.

The default names for the slave status logs when stored as files were `master.info` and `relay-log.info`. The names could be changed using the `--master-info-file` and `--relay-`

`log-info-file` options, respectively. As InnoDB tables, the slave status logs are named `mysql.slave_master_info` and `mysql.slave_relay_log_info`.

To modify an existing replication slave that is using a `FILE` repository for the slave status logs to use `TABLE` repositories, convert the existing replication repositories dynamically by running the following commands:

```
STOP SLAVE;
SET GLOBAL master_info_repository = 'TABLE';
SET GLOBAL relay_log_info_repository = 'TABLE';
```

The master info log table `mysql.slave_master_info` should be protected because it contains the password for connecting to the master. When you back up the replication slave's data, ensure that you back up the `mysql.slave_master_info` and `mysql.slave_relay_log_info` tables containing the slave status logs, because they are needed to resume replication after you restore the data from the slave.

Bugs Fixed

- **Incompatible Change; JSON:** If a JSON object contained multiple members with the same key name, MySQL kept the first member and discarded the remainder. This contradicts RFC 7159, which suggests that duplicate key names can be handled in one of the ways listed here:
 - Report an error (or otherwise fail to parse the object)
 - Report all of the name-value pairs, including duplicates
 - Report the last name-value pair only

When a JSON text is evaluated in JavaScript, the last name-value pair is kept if multiple pairs with the same name are specified. MySQL now does likewise, and implements the last of the three options just listed, as shown here:

```
mysql> CREATE TABLE t1 (c1 JSON);
mysql> INSERT INTO t1 VALUES ('{"x": 17, "x": "red", "x": [3, 5, 7]}');
mysql> SELECT c1 FROM t1;
+-----+
| c1          |
+-----+
| {"x": [3, 5, 7]} |
+-----+
```

The fix for this issue also corrects a failure in the MySQL 8.0 server to handle insertion into a JSON column of data containing JSON arrays as the values for multiple identical keys. (Bug #86620, Bug #86866, Bug #26238736, Bug #26369555)

- **Performance; JSON:** Creating a representation of a JSON string now optimizes for the most common case—that the string to be processed contains no special characters that need to be escaped—scanning for the first special character in the string, and copying each sequence of characters which do not require escaping in a single `memcpy()` call, rather than checking each character in turn to determine whether it needed to be escaped, escaping it if so, and then copying it, one by one, as was done previously.

This fix also corrects a failure to escape the control character `\u001f`, or unit separator character. (Bug #86898, Bug #26388690, Bug #87722, Bug #26780307)

References: See also: Bug #25977595.

- **InnoDB:** A long semaphore wait occurred when executing `ALTER TABLE`, `DROP TABLE`, and `DROP DATABASE` operations. (Bug #26779650)
- **InnoDB:** Invalid error handling code was removed from a function related to tablespace import. (Bug #26595476)
- **InnoDB:** File-per-table tablespaces created prior to MySQL 5.6 caused a failure during an in-place upgrade to MySQL 8.0.2. The tablespaces were not registered with the `InnoDB SYS_TABLESPACES` system table, as required.

Tables with decimal columns created prior to MySQL 5.5 also caused a failure during an in-place upgrade to MySQL 8.0.2, due to a precision type mismatch. (Bug #26542296, Bug #87229)

- **InnoDB:** A query was interrupted during concurrent `ALTER TABLE` operations due to a secondary index entry count mismatch. (Bug #26381213)
- **InnoDB:** A segmentation fault occurred when attempting to open a table that was altered while strict mode was disabled to include conflicting `TABLESPACE` and `COMPRESSION` attributes. (Bug #26375851)
- **InnoDB:** A segmentation fault occurred during a DML operation that used the TempTable storage engine. (Bug #26363837)
- **InnoDB:** Implicit row format conversion during an `ALTER TABLE ... REORGANIZE PARTITION` operation raised an invalid assertion. (Bug #26326611)
- **InnoDB:** An `ALTER TABLE` operation that rebuilt an encrypted table did not set the encryption attribute properly. (Bug #26243264)
- **InnoDB:** A memory leak was encountered on Windows when using the TempTable storage engine for in-memory internal temporary tables. (Bug #26237680)
- **InnoDB:** Misleading errors were produced when running Valgrind tests on a server build that was not enabled for Valgrind testing. (Bug #26037206)
- **InnoDB:** Problematic code related to dropping orphan full-text search tables caused an invalid object ID assertion failure on startup. (Bug #25998362)
- **InnoDB:** A check for discarded partitions during a DML operation only checked the first partition. Failure to check for other discarded partitions caused an assertion failure. (Bug #25942592)
- **InnoDB:** Allocated memory was not initialized before it was written to a file, resulting in a Valgrind error. (Bug #25913151, Bug #85986)
- **InnoDB:** Adding a virtual column and index in the same statement caused an error. (Bug #25899959)
- **InnoDB:** Replication lag occurred on slave instances during large update operations on tables with many partitions. (Bug #25687813, Bug #85352)
- **InnoDB:** A failure occurred during an end range comparison. (Bug #25669686)
- **InnoDB:** Enabling the `innodb_buffer_pool_load_now` setting failed in read-only mode. The event that signals the buffer pool load thread was not initialized. (Bug #25586766)
- **InnoDB:** A cursor position check by a multiversion concurrency control row search function raised an assertion. (Bug #25377592)

- **InnoDB:** The wrong variable was passed to the `row_mysql_handle_errors` routine causing an assertion failure. (Bug #25183130)
- **InnoDB:** A long wait for a dictionary operation lock held by a full-text search synchronization operation caused a server exit. (Bug #24938374, Bug #26376681, Bug #26376239)
- **InnoDB:** Assertion code was modified to account for the possibility of a transaction attempting to acquire an explicit lock on a record while another transaction converts an implicit lock to an explicit lock on the same record prior to a commit operation. (Bug #24344131)
- **InnoDB:** A `FLUSH TABLES ... FOR EXPORT` operation on an encrypted or page-compressed table raised an assertion. (Bug #22916982)
- **InnoDB:** A `DROP TABLE` operation was not permitted with an `innodb_force_recovery` setting greater than 0. (Bug #22392152)
- **InnoDB:** A `CREATE TABLE ... SELECT` operation raised an assertion failure when the newly created table was dropped before the transaction was committed. (Bug #22154768)
- **Partitioning:** In certain cases when fetching heap records a partition ID could be set to zero. (Bug #86255, Bug #26034430)
- **Partitioning:** It was possible for a `CREATE TABLE` statement that failed to create a partitioned `InnoDB` table not to be rolled back correctly. This was due to an extraneous commit made while performing a check of foreign key information. Since partitioned tables do not presently support foreign keys, this check is unnecessary, and so is no longer made in such cases. (Bug #85299, Bug #25667278)
- **Partitioning:** Queries involving `NULL` were not always handled correctly on tables that were partitioned by `LIST`. (Bug #76418, Bug #20748521)

References: See also: Bug #86255, Bug #26034430.

- **Replication:** On a multi-threaded slave, it was possible for a deadlock state to occur due to the timing of updates to the record of disk space used by the relay log. The timing of the update has now been changed so that the deadlock cannot occur. (Bug #26729635)
- **Replication:** With `slave_preserve_commit_order=1` set, a deadlock could occur between a transaction holding a shared write lock on a table, and a transaction earlier in the commit order that also required a shared write lock. (Bug #26666609)
- **Replication:** The Group Replication flow control variables now correctly permit you to have some members in a group that do not affect the minimum throughput of the flow-control mechanism, effectively ignoring those members in case they become blocked. (Bug #26537497)
- **Replication:** The unused variable `opt_reckless_slave` was removed. (Bug #26500285)
- **Replication:** When the Group Replication plugin `Delayed_initialization_thread` failed to start due to unavailable resources, a locked mutex was being kept behind which would cause issues on `Delayed_initialization_thread` destructor. The fix ensures that the mutex is unlocked when the thread fails to start. (Bug #26394678)
- **Replication:** In the case of delayed initialization of the Group Replication plugin, deployed in single-primary mode, secondaries were able to get writes through an asynchronous replication channel, which is not allowed in normal initialization of the Group Replication plugin. (Bug #26314756)
- **Replication:** If the options file contained Group Replication related settings the server could stop unexpectedly on start up. (Bug #26314472)

- **Replication:** `FLUSH LOGS` attempted to send an OK message after having already sent an error response during the commit phase. (Bug #26272158)
- **Replication:** With GTIDs generated for incident log events, MySQL error code 1590 (`ER_SLAVE_INCIDENT`) could not be skipped using the `--slave-skip-errors=1590` startup option on a replication slave. (Bug #26266758)
- **Replication:** `COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE` was set to an incorrect value when `group_replication_recovery_complete_at="transactions_certified"` on a recovering member. (Bug #26180350)
- **Replication:** The values of the `group_replication_recovery_use_ssl` and `group_replication_recovery_ssl_verify_server_cert` variables were not being updated when configured for the Group Replication recovery channel. (Bug #26142801)
- **Replication:** When replicating a partitioned table with an index, on a replication slave where `HASH_SCAN` was specified as part of the `slave_rows_search_algorithms` setting, the slave I/O thread sometimes stopped with an error `HA_ERR_KEY_NOT_FOUND`. (Bug #26137159)
- **Replication:** It was possible to set `server_uuid` to the same value as `group_replication_group_name`. Doing so could result in unexpected behavior because GTIDs are identified by a UUID. Now it is not possible to set `server_uuid` to the same value as `group_replication_group_name`. (Bug #26035931)
- **Replication:** The system variable `pseudo_slave_mode`, which is for internal server use, sometimes raised an assertion when it was changed inside a transaction. The server no longer changes this variable inside a transaction. (Bug #26034192, Bug #86250)
- **Replication:** When write sets are used for parallelization by a replication slave (as specified by the `binlog_transaction_dependency_tracking` system variable), empty transactions are now ignored, and the handling of relay log rotation has been optimized. (Bug #25982097)
- **Replication:** The Performance Schema `replication_applier_status_by_worker` table sometimes incorrectly displayed a value for `APPLYING_TRANSACTION` for an inactive worker, because the table was being populated before the worker thread stopped. (Bug #25896166, Bug #85951)
- **Replication:** Joining a member running a lower version to a group running a higher version resulted in the members running the higher version becoming unreachable. (Bug #25568493)
- **Replication:** Attempting to uninstall the plugin while `START GROUP_REPLICATION` executed could result in unexpected behavior. (Bug #25423650)
- **Replication:** In case of a failure while creating multiple slave applier worker threads, some threads would be left orphaned and their resources would not be collected. Thanks to Laurynas Biveinis for his contribution to fixing this bug. (Bug #24679056, Bug #82980)
- **Replication:** The `binlog_checksum` option cannot be changed within a transaction. MySQL cannot log this statement, as would be required inside a transaction, while the requested function is being performed on the binary log. (Bug #22914463)
- **Replication:** It was possible to start the server with invalid values for the Group Replication flow control options. Now, the `--group-replication-flow-control-min-quota`, `--group-replication-flow-control-max-quota`, and `--group-replication-flow-control-min-recovery-quota` options are validated on server startup. (Bug #87206, Bug #26531899)
- **Replication:** If hostname resolution was not working for a member in Group Replication, the error returned when attempting to connect was referring to credentials. The error message has been improved to describe the problem with hostname resolution. (Bug #86858, Bug #26368004)

- **Replication:** The Group Replication plugin no longer sets `auto_increment_increment` and `auto_increment_offset` variables when single primary mode is active. (Bug #86669, Bug #26263155)
- **Replication:** Group Replication partition threads were not visible in the Performance Schema tables. (Bug #86626, Bug #26241008)
- **Replication:** `group_replication_force_members` could be used in situations where the group was working properly, in other words a majority was reachable. This incorrect use could cause instability in the group. Therefore, its use has been restricted to the scenario for which it was created, for forming a new membership from a subset of a previous group's membership when a majority of the members are unreachable. (Bug #86359, Bug #26093967)
- **Replication:** On a member which had both Group Replication and asynchronous replication running simultaneously, asynchronous replication was not respecting the restrictions required by Group Replication, such as only using InnoDB storage engine, tables requiring primary keys and so on. This could also be encountered when running `mysqlbinlog` against the member. Now, members that are running Group Replication and asynchronous replication do not allow Group Replication's requirements to be broken. (Bug #85164, Bug #25609945)

References: See also: Bug #85781, Bug #25828806.

- **JSON:** Containers in the internal representations of JSON objects and arrays (`Json_object` and `Json_array`) have been changed to use smart pointers rather than raw pointers to `Json_dom`, so that orphaned DOM objects are now automatically destroyed. (Bug #26161264)
- **JSON:** ASCII character 31 (`\u001f`, the unit separator) in a string literal within a JSON document was not quoted when the JSON document was formatted as a string (for example, by `CAST(' '\u001f' ' AS JSON)`). (Bug #25977959)
- **JSON:** When a `path_expression` identified a nonarray value, the `JSON_INSERT()` and `JSON_ARRAY_INSERT()` functions failed to evaluate `path_expression[0]` as equal to `path_expression`. (Bug #86213, Bug #26022576)
- **JSON:** Searches with `JSON_EXTRACT()` that used wildcards took an inordinate amount of time. (Bug #84523, Bug #25418534)

References: See also: Bug #83959, Bug #25151440.

- Under heavy load, an infinite loop occurred in Performance Schema buffer container code. (Bug #26666274)
- MySQL-specific typedefs such as `uchar` and `my_bool` were inadvertently reintroduced into the client namespace if the `mysql.h` header file was included. (Bug #26588846, Bug #26582752, Bug #87337)
- `uint8korr()` and related macros were fixed so that they explicitly do unaligned accesses, even on x86. (Bug #26568748, Bug #87298)
- The `main.mysql_upgrade_grant`, `main.roles-upgrade`, and `auth_sec.secure_file_priv_warnings`, test cases mishandled the error log. The `sys_vars.innodb_redo_log_encrypt_basic` test case output was unstable. Thanks to Laurynas Biveinis for the patches. (Bug #26562401, Bug #87279, Bug #26575150, Bug #87313, Bug #26575142, Bug #87314, Bug #26582158, Bug #87303)
- For debug builds, with `sql_buffer_result` enabled, recursive common table expressions caused a server exit. (Bug #26556025)

- Incorrect resolution of a window function as a constant function could result in a server exit. (Bug #26500442)
- For window functions, use of `GROUP BY ... WITH ROLLUP` could cause a server exit. (Bug #26497353, Bug #26497247)
- For window functions, `JSON` columns raised an assertion. (Bug #26496733)
- For window functions, a value less than the argument for a `RANGE` frame caused a server exit. (Bug #26496645)
- Compiling with `-DWITHOUT_SERVER=1` resulted in `my_symlink.c` compilation failure due to missing `#include` for `my_dir.h`. Thanks to Christian Hesse for the patch. (Bug #26495816, Bug #87137)
- Compiling with `-DWITH_SSL=system -DWITH_ZLIB=system` assumed that the system `openssl zlib` command was available, which might not be the case. Now availability of that command is checked, and if unavailable, the `zlib_decompress` utility is built. (Bug #26494495, Bug #87123)
- yaSSL could incorrectly perform TLS cipher negotiation. (Bug #26482173)
- Some `thread_stack` settings could result in a server exit. (Bug #26438067)
- For window functions, a `ROW` frame accepted noninteger arguments for the row count. (Bug #26411055, Bug #86990)
- `REPLACE(UUID(), ...)` expressions could be cached (improperly) and return the same value for each row of a result set. (Bug #26395601)
- When building MySQL within the source tree, `make install` installed some `CMake` files into the `mysql-test` directory within the tree. (Bug #26385175, Bug #86905)
- The `PROCESS_ID` column in the Performance Schema `session_connect_attrs` and `session_account_connect_attrs` tables was changed from `INT` to `BIGINT UNSIGNED` to accommodate larger process ID values. Thanks to Daniël van Eeden for the patch. (Bug #26357806, Bug #86835)
- The index on a generated column, whose value was generated from `JSON_EXTRACT()`, was sometimes not used by the optimizer when it should have been. (Bug #26352119)
- Setting the `log_error_services` system variable to `NULL` caused a server exit. (Bug #26331795)
- For `tar` file packages, some test suite shared libraries were installed in the server package rather than the test package. (Bug #26329850)
- `SHOW COLUMNS` for a valid view could fail. (Bug #26322203, Bug #86778)
- An operation that caused renaming or removal of histogram statistics could cause a server exit. (Bug #26303972)
- For a `VARCHAR` column, sorting using an explicit collation (`ORDER BY col_name COLLATE collation_name`) was much slower than with an implicit collation (no `COLLATE` clause), even if the explicit collation was the same as the implicit collation. (Bug #26286790, Bug #86710)
- `SET binlog_format = ROW` produced a syntax error because `ROW` is now a reserved word. This syntax is now recognized specially to preserve backward compatibility. (Bug #26269280)
- `SET PERSIST_ONLY`, should be permitted only to users who have the `PERSIST_RO_VARIABLES_ADMIN` and `SYSTEM_VARIABLES_ADMIN` privileges, but was incorrectly also permitted to users with the `SUPER` privilege. (Bug #26247864)

- The `information_schema_stats` configuration option, introduced in MySQL 8.0.0, was removed and replaced by `information_schema_stats_expiry`.

`information_schema_stats_expiry` defines an expiration setting for cached `INFORMATION_SCHEMA` table statistics. For more information, see [Optimizing INFORMATION_SCHEMA Queries](#).

The `TABLES_DYNAMIC`, `STATISTICS_DYNAMIC`, and `SHOW STATISTICS_DYNAMIC` internal system views were removed. (Bug #26203731, Bug #83957)

- Source packages for Debian platforms contained prebuilt debug binaries, causing build failures on any architectures other than the one on which those binaries were built. (Bug #26186911)
- The UDF registration service did not work if used during server startup. (Bug #26173244)
- Disabling table instrumentation by changes to the Performance Schema `setup_objects` table could cause incorrect index names in index statistics. (Bug #26162562)
- A misleading error message was returned when attempting to drop a nonexistent tablespace file. (Bug #26133507, Bug #86438)
- When running `mysqlbinlog` with the `--read-from-remote-server` option, rewrite rules specified using the `--rewrite-db` option were ignored, so data was not written to the target database. (Bug #26117735, Bug #86288)
- Timestamp data copied from the data dictionary cache during a DDL operation was converted using a `time_zone` value that was no longer valid. The resulting timestamp data was incorrect, causing an error in release builds and an assertion failure in debug builds. (Bug #26091333, Bug #86290)
- Successful data dictionary updates but failure to write the binary log event could result in an inconsistent state. (Bug #26037355)
- `mysqlbinlog` now prints the full metadata for the event type `Table_map_log_event`. (Bug #26020990)
- Some string functions in an `ALTER EVENT` statement could cause a server exit. (Bug #25942505)
- Uninstalling the `daemon_memcached` plugin caused a serious error. (Bug #25909540)
- Prepared statements that used a common table expression and many `?` parameters could be slow. (Bug #25903274, Bug #85933)
- For `UPDATE` or `DELETE` statements with an `ORDER BY ... LIMIT` clause, the optimizer sometimes failed to identify a cheaper ordering method than `filesort`. (Bug #25899921)
- The `rpl_diff.inc` test case file did not find the data difference between servers. Thanks to Yura Sorokin for the patch. (Bug #25860138, Bug #85838)
- An ngram fulltext parser search query returned incorrect results and raised an assertion. (Bug #25851975)
- The combination of an index virtual generated column, a foreign key, and a trigger could cause an assertion to be raised. (Bug #25817660, Bug #85757)
- Selecting from a view could yield different results with materialization enabled versus materialization disabled. (Bug #25782811, Bug #85622)
- After using `SET PERSIST` to set the `event_scheduler` system variable, the server failed to restart if started with the `--skip-grant-tables` option. (Bug #25776940)

- For debug builds, adding an index to a table that had a foreign key relationship could raise an assertion. (Bug #25739983)
- An assertion could be raised for `MIN()/MAX()` access to system tables. (Bug #25738624)
- A failed `DROP VIEW` could be written to the binary log. (Bug #25680097)
- `SHOW CREATE VIEW` sometimes added a database name prefix to table names that was not present in the original view definition. (Bug #25634576, Bug #85176)
- The Performance Schema `variables_info` table displayed incorrect `VARIABLE_SOURCE` and `VARIABLE_PATH` values for variables set within option files specified by `!include` or `!includedir` directives. (Bug #25563891)
- Constant string propagation could fail for UCA-based collations. (Bug #25503965, Bug #84837)
- `mysqldump` displayed incorrect progress information about the number of tables dumped. (Bug #25432850)
- Calculations for UCA 9.0.0 collations were inefficient for tailoring rules containing contraction characters. (Bug #25426632, Bug #84577, Bug #25426632, Bug #84577)
- `GROUP BY DESC` on `DECIMAL` values could incorrectly group `NULL` with non-`NULL` values. (Bug #25407964, Bug #84537)
- Some `mysqldump` warnings went to the standard output rather than the standard error output and consequently were written to the dump file. (Bug #25380000, Bug #82992)
- `NULL` values generated as a result of `WITH ROLLUP` were replaced with the previous row's value when executing a prepared statement on a view having the `GROUP BY .. WITH ROLLUP` clause. (Bug #25174118)
- A server error occurred when a full text search result exceeded the `innodb_ft_result_cache_limit` setting. The patch for this bug also backports a related patch (Bug #21140111). (Bug #25033538)
- A parser refactoring in MySQL 8.0.1 resulted in incorrect handling of some `INSERT ... ON DUPLICATE KEY UPDATE` statements. These problems have been corrected. (Bug #24716127, Bug #25526439, Bug #25071305)
- For debug builds, a `CREATE TABLE` statement with a `VARBINARY` or `BINARY` column having a default value in hexadecimal format caused a server exit. (Bug #24679166, Bug #83020)
- If a stored function was considered a constant by the optimizer, calling it from a subquery in a `NOT IN` condition in the `WHERE` clause could cause a server exit. (Bug #23577867)
- A `mysqldump` memory leak was fixed. Thanks to Yura Sorokin for the patch. (Bug #23531150, Bug #81714)
- Incorrect results or a server exit could result when a query used Batched Key Access optimization and a virtual generated column was part of the join buffer. (Bug #23169112)
- If a session rolled back to a savepoint and then was killed, the statements up to the point of the savepoint could be committed. (Bug #22350047, Bug #79596)
- MySQL accepted a reference to an alias of an aggregated expression defined in an outer query block even when the reference occurred within a `GROUP BY` subquery where the reference was meaningless. (Bug #21974346, Bug #78785)

- For clients that used Connector/Python and authenticated using the [sha256_password](#) plugin, the server could handle connections incorrectly. (Bug #21421642)
- Some [SELECT DISTINCT](#) queries with [GROUP BY](#) could return incorrect results. (Bug #20692219, Bug #76283)

Changes in MySQL 8.0.2 (2017-07-17, Development Milestone)



Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running [mysql_upgrade](#). For example, you may find it necessary to dump your data with [mysqldump](#) before the upgrade and reload it afterward.

- [Account Management Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Data Dictionary Notes](#)
- [Deprecation and Removal Notes](#)
- [Logging Notes](#)
- [Optimizer Notes](#)
- [Packaging Notes](#)
- [Parser Notes](#)
- [Performance Schema Notes](#)
- [Security Notes](#)
- [Spatial Data Support](#)
- [Test Suite Notes](#)
- [X Plugin Notes](#)
- [Platform-Specific Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Account Management Notes

- During data directory initialization or upgrade, MySQL now creates a `'mysql.session'@'localhost'` reserved account. This account is used internally by plugins to access the server. It is locked so that it cannot be used for client connections. (Bug #25642343)
- These system variables now are available to define mandatory roles and to enable granted roles to be automatically activated at client connection time:
 - `mandatory_roles` takes a value listing roles the server should treat as automatically granted to all users.
 - `activate_all_roles_on_login` enables control over automatic activation of all granted roles when users log in to the server.

For more information, see [Using Roles](#).

Character Set Support

- For Unicode data that uses NO PAD collations, sorting of multibyte and variable-length values has been improved:
 - NO PAD collations are those based on UCA 9.0.0 and higher, such as `utf8mb4_0900_ai_ci` or `utf8mb4_ja_0900_as_cs`.
 - The performance improvement is greatest for key values with short weight strings; that is, strings where the weight strings do not fill their entire permitted length. For a `VARCHAR(10)` column that uses the `utf8mb4_0900_ai_ci` collation, values may take up to 160 bytes. The string `'a'` uses only 2 bytes out of a possible 160 and is more sparse than `'abcdefghij'`. But even `'abcdefghij'` uses only 20 bytes of a possible 160 and is more sparse than a string of, for example, ten copies of U+337F SQUARE CORPORATION (株式会社), whose weight string requires the entire 160 bytes available.

(Bug #25750527, Bug #85546)

- MySQL now supports a new collation, `utf8mb4_0900_as_ci`, for the `utf8mb4` Unicode character set. This collation is accent sensitive and case insensitive. It is similar to the default `utf8mb4` collation (`utf8mb4_0900_ai_ci`) except that the default collation is accent insensitive.

MySQL also now supports a new Japanese collation, `utf8mb4_ja_0900_as_cs_ks`, for the `utf8mb4` Unicode character set. This collation is like `utf8mb4_ja_0900_as_cs` in that it is accent sensitive and case sensitive, but `utf8mb4_ja_0900_as_cs_ks` is also kana sensitive and distinguishes Katakana characters from Hiragana characters. `utf8mb4_ja_0900_as_cs` treats Katakana and Hiragana characters as equal for sorting. Applications that require a Japanese collation but not kana sensitivity may use `utf8mb4_ja_0900_as_cs` for better sort performance. `utf8mb4_ja_0900_as_cs` uses three weight levels for sorting; `utf8mb4_ja_0900_as_cs_ks` uses four.

`utf8mb4_ja_0900_as_cs_ks` is the first collation to use the `_ks` collation suffix. Japanese collations without this suffix are not kana sensitive.

For more information, see [Unicode Character Sets](#).

- These character set changes were made to MySQL client support:
 - These clients use a default character set of `utf8mb4` rather than `latin1:mysql`, `mysql_upgrade`, `mysqladmin`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqlslap`, `mysqlshow`, and `mysqltest`. (This change does not affect character set autodetection performed by `mysql`, `mysqladmin`, `mysqlcheck`, `mysqlimport`, and `mysqlshow`, as described at [Connection Character Sets and Collations](#)).

- Client connections created using the `libmysqlclient` library use a default character set of `utf8mb4` rather than `latin1`.
- The `mysqltest` program supports a `--default-character-set` option for specifying the character set explicitly.

Compilation Notes

- **Performance:** The strict aliasing optimization for GCC is no longer disabled, which results in a minor improvement for MySQL single-threaded performance.
- **Linux:** MySQL now compiles on Alpine Linux. (Bug #25945568, Bug #80322)
- **Solaris:** On Solaris, server builds now use `std::atomic`, so they must be linked against `libstatomic`. (Bug #25957991)
- MySQL now compiles using Visual Studio 2017. (Bug #25788406)
- The required version of the Boost library for server builds has been raised to 1.64.0. (Bug #25772329, Bug #85593)
- These changes were made with respect to client program development:
 - Client programs should only need to `#include` the `<mysql.h>` header file. In particular, `<my_config.h>` should not be needed, and is no longer installed.
 - The `my_init()` function is no longer included in the list of symbols exported from `libmysqlclient`. It need not be called explicitly by client programs because it is called implicitly by other C API initialization functions.
(Bug #25732787)
- For the Xcode IDE, header files are added to MySQL project sources so they can be searched. (Bug #25636986)
- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

Component Notes

- New services are available to enable server components to register and unregister system variables (`component_sys_variable_register`, `component_sys_variable_unregister`) and to register status variables (`status_variable_registration`).
- Two new services are available to enable server components and plugins to register and unregister user-defined-functions (UDFs): `mysql_service_udf_registration` and `mysql_service_udf_registration_aggregate` provide registration services for scalar and aggregate UDFs, respectively. These services enable components and plugins to manage UDFs for themselves, without the need for `CREATE FUNCTION` and `DROP FUNCTION` statements.

UDFs registered using either these services or `CREATE FUNCTION` now are listed in the new Performance Schema `user_defined_functions` table. See [The user_defined_functions Table](#).
- The `mysql_string` string-manipulation service has been converted from a plugin service to a component service: a set of string service APIs for use by server components. The password validation

component has been updated to use the revised service implementation. For information about services, see [MySQL Services for Components and Plugins](#).

Configuration Notes

- Previously, `LOCAL` capability for `LOAD DATA` operations was enabled by default in some contexts: The client library in MySQL binary distributions was compiled with client-side `LOCAL` capability enabled, and the `local_infile` system variable was enabled on the server side. `LOCAL` capability is now disabled by default in all contexts. Client programs must be configured explicitly to enable `LOCAL`, and the server must be run with `local_infile` enabled. See [Security Issues with LOAD DATA LOCAL](#). (Bug #24511108)
- `mysqld` startup behavior has been modified as follows. In general, these changes enable more consistent or easier server startup when not using `mysqld_safe`.
 - The server executable determines its own full path name at startup and uses the parent of the directory in which it is located as the default `basedir` value. This in turn enables the server to use that `basedir` when searching for server-related information such as the `share` directory containing error messages.
 - If error log output cannot be redirected to the `--log-error` option value, error output could be lost. This is now less likely.
 - If the server is started using the `--daemonize` option and is not connected to a tty device, a default error logging option of `--log-error=""` is used in the absence of an explicit logging option, to direct error output to the default log file. Previously, an error could occur under these circumstances.
 - The `-D` and `-I` options now are synonyms for `--daemonize` and `--initialize`, respectively.(Bug #20398088, Bug #75343)
- MySQL now supports a `SET PERSIST_ONLY` variant of `SET` statement syntax, for making configuration changes at runtime that also persist across server restarts. Like `SET PERSIST`, `SET PERSIST_ONLY` writes the variable setting to an option file named `mysqld-auto.cnf` in the data directory. However, unlike `PERSIST`, `PERSIST_ONLY` does not modify the runtime global system variable value. This makes `PERSIST_ONLY` suitable for configuring read-only system variables that only be set can at server startup. For more information, see [Using Option Files](#), and [SET Syntax for Variable Assignment](#).

Use of `SET PERSIST_ONLY` requires the new `PERSIST_RO_VARIABLES_ADMIN` privilege.

- The `explicit_defaults_for_timestamp` system variable is now enabled by default (previously disabled by default), and a warning occurs if you disable it. This means that the nonstandard (and deprecated) behaviors for default values and `NULL`-value handling in `TIMESTAMP` columns are now disabled by default.

Data Dictionary Notes

- The `column_stats` system table has been removed and replaced by the `column_statistics` data dictionary table.
- The `version` data dictionary table was renamed to `dd_properties`.

Metadata for `INFORMATION_SCHEMA` tables created by dynamic plugins now is recorded in the data dictionary and visible through the `INFORMATION_SCHEMA TABLES` table.

These tables are no longer visible in `INFORMATION_SCHEMA: SHOW_STATISTICS`, `SHOW_STATISTICS_DYNAMIC`, `STATISTICS_BASE`, `STATISTICS_DYNAMIC`, `TABLES_DYNAMIC`. User impact is minimal because these tables are for internal use by server components.

- The [InnoDB](#) storage engine now uses the global MySQL data dictionary rather than its own storage engine-specific data dictionary. For information about the data dictionary, see [MySQL Data Dictionary](#).

The following list briefly describes the main implications of this change:

- Upgrade and downgrade implications:
 - To upgrade from MySQL 5.7 to MySQL 8.0, you must perform the upgrade procedure described at [Upgrading MySQL](#).
 - Downgrading from MySQL 8.0 to MySQL 5.7 is only supported using the logical downgrade method (a `mysqldump` downgrade). In-place downgrades are not supported.
- Metadata updates associated with exporting and importing tablespaces using the transportable tablespace feature are now performed on global data dictionary tables instead of [InnoDB](#) data dictionary tables.
- [InnoDB](#) in-memory metadata is now instantiated from global data dictionary objects. This metadata was previously read from [InnoDB](#) system tables.
- Table options that signify tablespace encryption and transparent page compression are now retrieved from the global data dictionary.
- Data dictionary support was added for [InnoDB FULLTEXT](#) indexes. Auxiliary index table names were changed to lowercase.
- [InnoDB](#) metadata created or modified during DDL operations is now written to the global data dictionary.
- `mysql` system tables and data dictionary tables are now created in a single [InnoDB](#) tablespace file named `mysql.ibd` in the MySQL data directory. Previously, these tables were created in individual [InnoDB](#) tablespace files in the `mysql` database directory. Associated changes include:
 - The [InnoDB](#) data dictionary buffer table, which stores fast changing [InnoDB](#) metadata, was renamed to `innodb_dynamic_metadata` and moved from the [InnoDB](#) system tablespace to the data dictionary tablespace.
 - Undo tablespace metadata now resides in the data dictionary tablespace.
 - Temporary tablespace metadata now resides in the data dictionary tablespace.
 - Table definitions for tables created in the `mysql` tablespace, the [InnoDB](#) system tablespace (`innodb_system`), or general tablespaces now include a `TABLESPACE` attribute. This change has the following implications:
 - You cannot use `CREATE TEMPORARY TABLE ... LIKE` to create an empty table based on the definition of a table that resides in any of the aforementioned tablespaces, as these tablespaces do not support temporary tables.
 - `CREATE TABLE ... LIKE` preserves the `TABLESPACE` attribute of the original table and creates a new table in the defined tablespace regardless of the `innodb_file_per_table` setting. This is a temporary regression.

For more information and workarounds, see [CREATE TABLE ... LIKE Syntax](#), and [CREATE TEMPORARY TABLE Syntax](#).

Deprecation and Removal Notes

- The `mysql` client by default strips comments in statements sent to the server, and this behavior is controlled using `--skip-comments` (strip comments), and `--comments` (preserve comments).

Comment stripping is now deprecated. This feature and the options to control it will be removed in a future MySQL release.

- Support for these deprecated syntax constructs for table and column references has been removed and their use now results in an error. Instances of these constructs should be changed to remove the leading period.
 - `.col_name`
 - `.tbl_name`
 - `.tbl_name.col_name`
- Symbolic link support as described at [Using Symbolic Links for MyISAM Tables on Unix](#), along with the `--symbolic-links` option that controls it, is now deprecated and will be removed in a future version of MySQL. In addition, the option is now disabled by default. The related `have_symlink` system variable also is deprecated and will be removed in a future version of MySQL.

Logging Notes

- **Incompatible Change:** These error-logging changes have been made:
 - The server is more forgiving if it cannot find the configured error-message file (specified using the `lc_messages_dir` and `lc_messages` system variables). Previously, the server wrote a message to the error log to indicate the problem, then aborted the startup process and exited. Now the server writes a message, but continues startup and defaults to built-in English messages. This applies to messages the server writes to the error log and sends to clients. See [Setting the Error Message Language](#).
 - Error logging was rewritten to use the MySQL component architecture. Traditional error logging is implemented using built-in components, and logging using the system log is implemented as a loadable component. In addition, a loadable JSON log writer is available. To control which log components to enable, use the `log_error_services` system variable. For more information, see [The Error Log](#).
 - Incompatibility: To enable logging to the system log, you must load the `log_sink_syseventlog` log component and list it in the `log_error_services` value (see [Error Logging to the System Log](#)). This differs from MySQL 5.7 and earlier, for which logging to the system log is enabled by default on Windows, and on all platforms requires no component loading.

A consequence of this configuration change is that the `log_syslog` system variable previously used to control logging to the system log is obsolete and changes to its value have no effect. `log_syslog` is now deprecated and will be removed in a future MySQL release.

Optimizer Notes

- MySQL now supports window functions that, for each row from a query, perform a calculation using rows related to that row. These include functions such as `RANK()`, `LAG()`, and `NTILE()`. In addition, several existing aggregate functions now can be used as window functions; for example, `SUM()` and `AVG()`. For more information, see [Window Functions](#).

**Note**

Each of the following words now is a reserved word and cannot be used as an identifier without identifier quoting: `CUME_DIST`, `DENSE_RANK`, `FIRST_VALUE`, `GROUPS`, `LAG`, `LAST_VALUE`, `LEAD`, `NTH_VALUE`, `NTILE`, `OVER`, `PERCENT_RANK`, `RANK`, `ROW_NUMBER`, `WINDOW`.

- MySQL now enables management of histogram statistics for table column values:
 - The `ANALYZE TABLE` statement supports `UPDATE HISTOGRAM` and `DROP HISTOGRAM` clauses for generating and removing column histogram statistics.
 - The server stores histogram information in the `column_statistics` data dictionary table. Histograms are viewable using the `INFORMATION_SCHEMA.COLUMN_STATISTICS` table.
 - The `histogram_generation_max_mem_size` system variable controls the amount of memory available for histogram generation.
 - The Performance Schema has a `memory/sql/histograms` instrument for monitoring memory allocations performed for histogram generation.
 - `mysqldump` and `mysqldump` have a `--column-statistics` option to add `ANALYZE TABLE` statements to the output to generate histogram statistics for dumped tables when the dump file is reloaded.

For more information, see [ANALYZE TABLE Syntax](#), [Optimizer Statistics](#), and [The INFORMATION_SCHEMA COLUMN_STATISTICS Table](#).

Packaging Notes

- `mysqladmin` was added to Docker/Minimal packages because it is needed by InnoDB Cluster. (Bug #25998285)
- Debian/Ubuntu packages now support multiple MySQL instances with `systemd`. See [Managing MySQL Server with systemd](#). (Bug #24559588, Bug #82785)

Parser Notes

- The parser rules for `ALTER TABLE` were refactored to be context independent and improve maintainability and extensibility. A resulting effect is that some previously accepted undocumented syntax variants are no longer accepted. For example, `CREATE TABLE` statements were permitted with column names qualified by the table name, or by the current database and table name. Such statements now produce an error.

Performance Schema Notes

- The Performance Schema now supports versioning, and maintains the current definitions for its tables internally. At startup, the server compares its supported Performance Schema version with the Performance Schema version stored in the data dictionary. If the versions differ, the server drops any old Performance Schema tables and recreates them using the current definitions. In consequence of this change:
 - For MySQL upgrades, it is no longer necessary to run `mysql_upgrade` to incorporate changes to Performance Schema tables because they are recreated automatically as necessary at server startup.
 - The `mysql_system_tables.sql` support script no longer includes SQL statements for Performance Schema table creation because these table definitions are maintained internally.

To support dynamic Performance Schema table manipulation, a new component service named `pfs_table_service` is now available.

- The Performance Schema default instrumentation settings have changed:
 - The `transaction` instrument and the `events_transactions_current` and `events_transactions_history` consumers are now enabled by default. See [Performance Schema Transaction Tables](#).
 - All `memory/%` memory instruments are now enabled by default. See [Memory Summary Tables](#).
 - The `wait/lock/metadata/sql/mdl` metadata lock instrument is now enabled by default. See [The metadata_locks Table](#).

Security Notes

- **Security Fix:** The linked OpenSSL library for the MySQL Commercial Server has been updated to version 1.0.2l. Issues fixed in the new OpenSSL version are described at <http://www.openssl.org/news/vulnerabilities.html>.

This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (Bug #26160962)

- Previously, any user could execute the `XA RECOVER` statement to discover the XID values for outstanding prepared XA transactions, possibly leading to commit or rollback of an XA transaction by a user other than the one who started it. Now `XA RECOVER` is permitted only to users who have the new `XA_RECOVER_ADMIN` privilege, which is expected to be granted only to administrative users who have need for it. This might be the case, for example, for administrators of an XA application if it has crashed and it is necessary to find outstanding transactions started by the application so they can be rolled back. This privilege requirement does not affect normal commit or rollback of an XA transaction because the user who started it knows its XID.

For MySQL upgrades, `mysql_upgrade` grants `XA_RECOVER_ADMIN` to users who have the `SUPER` privilege, unless some account is found that already has `XA_RECOVER_ADMIN`. (Bug #17188129)

Spatial Data Support

- Argument suitability checking was improved for these spatial functions that test geometry relationships: `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`, `MBRContains()`, `MBRCoveredBy()`, `MBRCovers()`, `MBRDisjoint()`, `MBREquals()`, `MBRIntersects()`, `MBROverlaps()`, `MBRTouches()`, `MBRWithin()`. For more information, see [Spatial Relation Functions That Use Object Shapes](#), and [Spatial Relation Functions That Use Minimum Bounding Rectangles](#).

As a consequence of this work, performance of spatial relation functions was improved for cases where both arguments contain geometries of one dimensionality; that is, where the `GeometryCollection` arguments (or one `GeometryCollection` and another type of geometry) passed to a spatial relation function can be reduced to a `MultiPoint`, `MultiLineString`, or `MultiPolygon`. In those cases, the more complex general `GeometryCollection` can be avoided. (Bug #22285402, Bug #79406)

Test Suite Notes

- The `--do-test-list` option for `mysql-test-run.pl` accepts an argument for a file containing tests one per line. Lines in the file are now accepted in any of the following formats:

```
test_name
```

```
test_name.test
suite_name.test_name
suite_name.test_name.test
path/to/test/file
```

(Bug #25700362)

- `mysql-test-run.pl` and `mysqltest` are now more restrictive about permitted test case and result file names. Names must consist only of alphanumeric characters (A-Z, a-z, 0-9), dash (-), or underscore (_), and cannot start with dash or underscore. An error occurs for nonconforming file names. (Bug #25487471)
- These changes were made to the `--xml-report` option for `mysql-test-run.pl`:
 - A `<failure>` tag identifies any test failing on a retry attempt.
 - Separate statistics and fields are included for skipped and disabled tests.
 - The XML report is created in the build directory if no absolute path is given for out-of-source builds.
 - For combination runs, a field named `variation` is included in the `<testcase>` tag.
 - Results for all tests belonging to a suite are aggregated within a single `<testsuite>` tag.
 - Information regarding failures is reported with a brief reason in an attribute named `message`, along with other details within the tag.

(Bug #25349924)

- `mysql-test-run.pl` now supports a `--report-unstable-tests` option that has these effects:
 - Reports any test that has passed using at least one retry attempt in a separate category called “Unstable tests” in the summary.
 - If all failures are due to unstable tests, `mysql-test-run.pl` produces a warning but exits successfully.
 - Adds a new XML tag to report unstable tests, if the `--xml-report` option is also specified.

(Bug #24473420, Bug #25984429)

- `mysqltest` now supports a `replace_numeric_round` command that takes an argument value from 0 to 16 indicating the number of decimals to round numeric values to. This can be used to help prevent result content mismatch errors for tests in which slightly different results are obtained across platforms due to precision differences. Thanks to Daniel Black for the patch. (Bug #23280117, Bug #81399)
- `mysql-test-run.pl` now looks for a `testname-client.opt` file, which is analogous to the `testname-master.opt` file but for specifying test-specific client options. (Bug #17084918)

X Plugin Notes

- X Plugin now handles expired SSL certificates correctly. (Bug #25835833)
- The output of clauses such as `GROUP BY` and `HAVING` has been improved by moving the `grouping_criteria` expression to the derived query. (Bug #25549637, Bug #24497007)
- During install of the X Plugin a blank rule was being generated in the Firewall white list. (Bug #24488234)

Platform-Specific Notes

- **Linux:** The generic Linux build for MySQL 8.0 now supports Non-Uniform Memory Access (NUMA) on its host system. Systems that use the build need to have `libnuma` installed on them. See [Installing MySQL on Unix/Linux Using Generic Binaries](#) for more details. (Bug #26005558)

Functionality Added or Changed

- **Incompatible Change; InnoDB:** `ADD PARTITION`, `DROP PARTITION`, `COALESCE PARTITION`, `REORGANIZE PARTITION`, and `REBUILD PARTITION ALTER TABLE` options are now supported by native partitioning in-place APIs and may be used with `ALGORITHM={COPY|INPLACE}` and `LOCK` clauses.

`EXCHANGE PARTITION`, which previously supported `ALGORITHM` and `LOCK` clauses, is performed by a new server layer API for compatibility with the MySQL data dictionary. Partition and table exchange is performed internally by the native partition handler.

`DROP PARTITION` with `ALGORITHM=INPLACE` deletes data stored in the partition and drops the partition. However, `DROP PARTITION` with `ALGORITHM=COPY` or `old_alter_table=ON` rebuilds the partitioned table and attempts to move data from the dropped partition to another partition with a compatible `PARTITION ... VALUES` definition. Data that cannot be moved to another partition is deleted.

- **InnoDB:** InnoDB now uses tablespace map files during recovery to identify tablespaces that require redo log application. This is a change from redo log tablespace discovery that was used previously.

If tablespace map files are lost or corrupted, the `innodb_scan_directories` startup option may be used to specify tablespace directories when starting MySQL after a server outage.

For more information, see [Tablespace Discovery During Crash Recovery](#). (Bug #24793413)

- **InnoDB:** In previous releases, InnoDB stores `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes specified in `CREATE TABLE` or `ALTER TABLE` statements even if those attributes are silently ignored when creating or altering a table. In such cases, `SHOW CREATE TABLE` and the `Create_options` column reported by `SHOW TABLE STATUS` show the specified `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes rather than the actual attributes of the table. InnoDB now stores and reports the actual `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes of the table.
- **Replication:** In previous versions issuing `STOP GROUP_REPLICATION` stopped the plugin but the server still accepted transactions. This meant the transactions were not transmitted to the group. To make `STOP GROUP_REPLICATION` safer, now `super_read_only` is set to `ON` immediately upon issuing `STOP GROUP_REPLICATION`, which ensures no transactions are accepted. (Bug #25495067, Bug #84795)
- **Replication:** When there was a network partition and a member was in a minority all queries to that member blocked. To improve this situation, the `group_replication_unreachable_majority_timeout` variable has been added which enables you to configure how long members in a minority wait to regain contact with a member in the majority before leaving the group. (Bug #25473794)
- **Replication:** The `group_replication_transaction_size_limit` variable was added to enable you to protect a group against large transactions causing a failure. (Bug #84785, Bug #25510757)
- **Replication:** Support for binary log files created by versions earlier than MySQL 5.0 has been removed, and binary log Version 1 and Version 3 formats are no longer supported by slaves or `mysqlbinlog`.
- **Replication:** The `replication_group_members` and `replication_group_member_stats` Performance Schema tables have been extended to provide more monitoring information for Group Replication. The `replication_group_members` table now includes information about member's roles,

and the versions of MySQL running on a member. The `replication_group_member_stats` table now includes information about all members in the group, the applier and local queues, and rolled back transactions.

- **Replication:** The receiver thread has been improved to no longer block other thread's activities when waiting for disk space. This improves the monitoring of replication ensuring that it reports correctly when the receiver thread is waiting for disk space. If you are not able to free disk space to allow the receiver thread to continue its activity, it can be forcefully stopped without side effects in most cases.
- **Replication:** A new transaction length field has been added to the `Gtid_log_event` which stores the transaction length in bytes.
- **Replication:** The following variables have had their defaults changed to ensure replication is as robust and efficient as possible by default:
 - `master_info_repository=TABLE` and `relay_log_info_repository=TABLE` ensure that replication repository information is stored in InnoDB tables, making replication resilient to unexpected halts.
 - `transaction_write_set_extraction=XXHASH64` enables the optimal method for generating write set hashes.
 - `slave_rows_search_algorithms='INDEX_SCAN,HASH_SCAN'` enables the use of a hash table to avoid repeated table scans when no Primary Key or Primary Key Equivalent (non-null unique secondary index) exists for a table
 - `expire_logs_days=30` causes unused binary logs that are older than 30 days to be purged, ensuring storage space is used efficiently.
- **Replication:** The Group Replication plugin now notifies other components in the server that some relevant events have happened. Upon view changes, recovery state updates, network partitioning and primary election, the plugin informs listeners registered in the service registry and notifies them that an event has occurred. These listeners, which could even be other plugins, can then react to these events.
- **Replication:** The `group_replication_member_weight` variable has been added which enables you to control the election of new primaries in single-primary mode. In previous versions primary election was based on the member's UUID, with the lowest UUID elected as the new primary in the event of fail over. Use this variable to assign numeric weights to members to ensure that specific members are elected, for example during scheduled maintenance of the primary or to ensure certain hardware is prioritized.
- **Replication:** The following new system variables have been added to configure Group Replication:
 - `group_replication_flow_control_hold_percent`
 - `group_replication_flow_control_max_commit_quota`
 - `group_replication_flow_control_member_quota_percent`
 - `group_replication_flow_control_min_quota`
 - `group_replication_flow_control_min_recovery_quota`
 - `group_replication_flow_control_period`
 - `group_replication_flow_control_release_percent`

These variables enable you to fine tune flow control of individual group members, adjusting the quota depending on the task the member performs in the group.

- **JSON:** Added support for ranges in the XPath expressions used with many MySQL JSON functions, including `JSON_EXTRACT()` and `JSON_REMOVE()`. Such a range is specified using the syntax `start to end`, where `start` and `end` are, respectively, the first and last indexes of a range of elements from a JSON array (always numbered starting with 0). For example, `$$[1 to 3]` includes the second, third, and fourth elements, as shown here:

```
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[1 to 3]');
+-----+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[1 to 3]') |
+-----+
| [2, 3, 4] |
+-----+
1 row in set (0.00 sec)
```

This work also provides support in such expressions for the `last` keyword, which you can use to represent the index of the last (rightmost) element in the current array, like this:

```
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last]');
+-----+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last]') |
+-----+
| 5 |
+-----+
1 row in set (0.00 sec)
```

Indexes relative to the end of the array are also supported, as shown here:

```
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last-4 to last-2]');
+-----+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last-4 to last-2]') |
+-----+
| [1, 2, 3] |
+-----+
1 row in set (0.00 sec)
```

For further information and examples, see [Searching and Modifying JSON Values](#). (Bug #79052, Bug #22285926)

- **JSON:** The optimizer now supports partial (in-place) updates of JSON documents when using `JSON_SET()`, `JSON_REPLACE()`, or `JSON_REMOVE()`. (Previously, all updates of JSON column values were done by completely removing the previous document and writing the new one in its place.) In general, this optimization can be performed when the replacement value is less than or equal in size to the column's original value, and no new elements are added to the original value. It *cannot* be performed for a simple replacement of the column value of the form `UPDATE tbl SET json_col = json_val`.

[The JSON Data Type](#) provides more information about the requirements for an update to be optimized in this way.

This work includes the addition of two JSON utility functions, `JSON_STORAGE_SIZE()` and `JSON_STORAGE_FREE()`, in the MySQL Server. `JSON_STORAGE_SIZE()` returns the number of bytes used to store the binary representation of a JSON document, whether the document is presented as a column value in a table, as the value of a user variable, or as a JSON literal. In the case of a JSON column, this is the space used to store the JSON document as it was inserted into the column, prior to any partial updates that may have been performed on it since then. `JSON_STORAGE_FREE()` shows the number of bytes in the binary representation of a JSON column value that were freed by the most recent partial update of the column. For a user variable storing a JSON document, `JSON_STORAGE_FREE()` always returns 0; it also returns 0 if the argument is a JSON literal.

Each of these functions, like many other MySQL functions that act on JSON values, also accepts a string that can be successfully parsed as a JSON document. For more information and examples, see [JSON Utility Functions](#).

- RPM `.spec` files now include support for running unit tests. (Bug #25814143, Bug #85743)
- For Windows, MSI installer packages now include a check for the required Visual Studio redistributable package, and produce a message asking the user to install it if it is missing. (Bug #25658832)
- The `mysql` client now supports a `--binary-as-hex` option that causes display of binary data using hexadecimal notation (`0xvalue`). Thanks to Daniël van Eeden for the patch. (Bug #25340722, Bug #84391)
- The `SHOW TABLES` statement now supports an optional `EXTENDED` keyword that causes statement output to list hidden tables created by failed `ALTER TABLE` statements. (These temporary tables have names beginning with `#sql`.) For more information, see [SHOW TABLES Syntax](#). (Bug #24786075, Bug #83241)
- Serialized Dictionary Information (SDI) files now have a lowercase extension (`.sdi`) instead of uppercase.
- The `TempTable` storage engine replaces the `MEMORY` storage engine as the default engine for in-memory internal temporary tables. The `TempTable` storage engine provides efficient storage for `VARCHAR` and `VARBINARY` columns. The `internal_tmp_mem_storage_engine` session variable defines the storage engine for in-memory internal temporary tables. Permitted values are `TempTable` (the default) and `MEMORY`. The `temptable_max_ram` configuration option defines the maximum amount of memory that can be occupied by the `TempTable` storage engine before it starts storing data on disk. The default value is 1073741824 bytes (1GiB).

The `memory/temptable/physical_ram` and `memory/temptable/physical_disk` Performance Schema instruments may be used to monitor `TempTable` memory allocation and disk storage.

- The MySQL 8.0.2 release introduces a number of undo related changes:
 - The number of undo tablespaces may be modified at runtime or when the server is restarted using the existing `innodb_undo_tablespaces` configuration option. Previously, the number of undo tablespaces could only be configured when initializing the MySQL instance and could not be changed afterward. This change permits the addition of undo tablespaces and rollback segments as a database installation grows.
 - `innodb_undo_log_truncate` is enabled by default. When enabled, any undo tablespace that exceeds the threshold value defined by `innodb_max_undo_log_size` is marked for truncation. See [Truncating Undo Tablespaces](#).
 - The `innodb_undo_tablespaces` default value was changed from 0 to 2, which means that rollback segments are created in two separate undo tablespaces instead of the `InnoDB` system tablespace by default. A minimum of two undo tablespaces is required to permit truncation of undo logs.

Setting `innodb_undo_tablespaces` to 0 is deprecated and will not be supported in a future release.

- The naming convention used for undo tablespace files is changed from `undoNNN` to `undo_NNN`, where `NNN` is the undo space number.
- The `innodb_rollback_segments` configuration option defines the number of rollback segments per undo tablespace. Previously, `innodb_rollback_segments` was a global setting that specified the total number of rollback segments for the MySQL instance. This change increases the number of

rollback segments available for concurrent transactions, and more rollback segments increases the likelihood that concurrent transactions use separate rollback segments for undo logs resulting in less resource contention.

- The `innodb_undo_logs` configuration option is removed. The `innodb_rollback_segments` configuration option performs the same function and should be used instead.
- The `InnoDB_available_undo_logs` status variable is removed. The number of available rollback segments per tablespace may be retrieved using `SHOW VARIABLES LIKE 'innodb_rollback_segments';`
- The `SHOW COLUMNS` and `SHOW INDEX` statements now support an optional `EXTENDED` keyword that causes statement output to include information about hidden columns and indexes that MySQL uses internally and are not accessible by users. For more information, see [SHOW COLUMNS Syntax](#), and [SHOW INDEX Syntax](#).

Bugs Fixed

- **Incompatible Change:** Plugins such as Group Replication and X Plugin now use the `mysql.session` account added in this version. If you are upgrading from a previous version which did not include the `mysql.session` account you must run `mysql_upgrade` to ensure the account is created. If `mysql_upgrade` is not run, plugins fail to start with the error message `There was an error when trying to access the server with user: mysql.session@localhost`. Make sure the user is present in the server and that `mysql_upgrade` was run after a server update. (Bug #26042764)

References: See also: Bug #24311527, Bug #25642343, Bug #25750822, Bug #25103980, Bug #83841.

- **InnoDB:** Queries run on `INFORMATION_SCHEMA.TABLES` performed more slowly with `information_schema_stats` set to `latest`. (Bug #26197113)
- **InnoDB:** An `ALTER TABLE ... ADD PARTITION` operation that specified a `DATA DIRECTORY` clause failed to ignore the `TABLESPACE` attribute of the table. (Bug #26113652)
- **InnoDB:** When `foreign_key_checks` is disabled, a child table with a foreign key constraint can be created before the parent table, which can result in a foreign key constraint failure, as the parent table is unaware of the constraint. When a table is created, there is now a call to load foreign key constraints for the table and check for child tables. (Bug #25976199)
- **InnoDB:** A parsing error occurred while optimizing a table with a full-text index. (Bug #25914332)
- **InnoDB:** Compiling MySQL on Windows with Microsoft Visual C++ 2015 or macOS 10.12.4 with GCC 4.2.1 or Apple LLVM version 8.0.0 (clang-800.0.38) returned warnings. (Bug #25910531)
- **InnoDB:** In debug builds, an assertion was raised during bootstrap when the system tablespace file (`ibdata1`) ran out of space during creation of doublewrite pages. (Bug #25872368)
- **InnoDB:** Incorrect locking order caused a deadlock when `InnoDB` attempted to persist an auto-increment counter value to disk. (Bug #25833228)
- **InnoDB:** Internal methods for accessing dictionary table object data did not account for virtual columns. (Bug #25822154)

References: This issue is a regression of: Bug #23748128.

- **InnoDB:** The length of a virtual column field in a virtual index record was less than the expected template column length. (Bug #25793677)

- **InnoDB:** In debug builds, shutting down the server with `--innodb-fast-shutdown=0` raised an assertion. (Bug #25756224)
- **InnoDB:** The `ibd2sdi` utility exited when run on an unsupported file type. (Bug #25738491)
- **InnoDB:** `InnoDB` did not set the compression algorithm when opening a partitioned table. (Bug #25685868)
- **InnoDB:** An in-place `ALTER TABLE` operation failed to set the encryption type, causing a `FLUSH TABLES ... FOR EXPORT` operation to assert. (Bug #25672779)
- **InnoDB:** A latch that was held while registering a file close caused a hang condition. (Bug #25658467)
- **InnoDB:** During recovery, prepared transactions were rolled back if the `innodb_force_recovery` setting was greater than 0. (Bug #25651042)
- **InnoDB:** A `CREATE TABLE` operation that defined a unique key with an eight character prefix on a NOT NULL `TEXT` field would raise an assertion if a primary key was not defined. (Bug #25647413)
- **InnoDB:** Updates to data dictionary tables combined with updates to `InnoDB` system tables for full-text search auxiliary tables raised a lock-related assertion. (Bug #25610353)
- **InnoDB:** The server allocated memory unnecessarily for an operation that rebuilt the table. (Bug #25573565, Bug #85043)
- **InnoDB:** Test-related code intended to simulate a random read on a non-existent page raised an invalid assertion. (Bug #25479538)

References: This issue is a regression of: Bug #25053705.

- **InnoDB:** With `foreign_key_checks` disabled, `InnoDB` incorrectly printed messages to the error log when operations were performed on a table that referenced a non-existent foreign key. (Bug #25365223)
- **InnoDB:** A `DROP TABLE` or `TRUNCATE TABLE` operation could raise an assertion following a failed online `ALTER TABLE ... ADD INDEX` operation. The index root page was dropped prematurely. (Bug #25357789)
- **InnoDB:** During a `TRUNCATE TABLE` operation on a file-per-table tablespace, a dictionary operation lock was released before eviction of dirty pages from the buffer pool, causing a latch order violation. (Bug #25357789)
- **InnoDB:** When using an index merge optimizer switch, a `SELECT COUNT(*)` operation sometimes returned 0. Partitioning code incorrectly performed a `memcpy` instead of a column copy of columns read by the index, causing the wrong records to be copied. (Bug #25332330, Bug #81031)
- **InnoDB:** After a `RENAME TABLE` operation that moved a table to a different schema, `InnoDB` returned an error on restart indicating that it could not locate the tablespace data file. `InnoDB` failed to update `INNODB_SYS_DATAFILES` data dictionary table during the `RENAME TABLE` operation. (Bug #25189192, Bug #84038)
- **InnoDB:** During an `ALTER TABLE` operation that rebuilt a table containing a virtual column, `InnoDB` failed to apply a concurrent insert log record. (Bug #24961167)
- **InnoDB:** `InnoDB` failed to apply the concurrent delete log for an in-place `ALTER TABLE` operation due to a virtual column validation issue. (Bug #24960450)
- **Partitioning:** Following execution of an `ALTER TABLE ... OPTIMIZE PARTITION` statement, names of tablespaces used by the table were not preserved, and subsequent accesses to the table eventually caused the server to fail. (Bug #25512556)

- **Replication:** When replicating a partitioned table with an index, on a replication slave where `HASH_SCAN` was specified as part of the `slave_rows_search_algorithms` setting, the slave I/O thread sometimes stopped with an error `HA_ERR_KEY_NOT_FOUND`. (Bug #26137159)
- **Replication:** A `USE` statement that followed a `SET GTID_NEXT` statement sometimes had no effect. (Bug #26128931)
- **Replication:** A misleading warning was issued when the command `FLUSH LOGS` or `PURGE LOGS BEFORE` was used on a binary log file with an expiry time set, and the binary log file was in use. The warning related to the file being in use, and implied that a purge attempt had taken place, even if the expiry time had not yet been reached. Now, MySQL checks the expiry time of the binary log file first, and only then checks whether the file is in use. So the warning is only issued for an in-use binary log file that is old enough to be purged. (Bug #25973525)
- **Replication:** Groups can now contain members running different server versions to enable you to do online upgrades of a replication group. The rules for combining members in a group with different versions are:
 - If you have a group with 8.0 members, you cannot add a 5.7 member
 - If you have a group with 5.7 members you can add a 8.0 member, but it remains in read-only mode. Writing to this member is dangerous while the group contains multiple server versions and should be avoided.

In a single-primary group, if the current primary leaves the group and a new primary must be elected, the primary is first chosen from the lower version members. If no lower version member is found, the primary is chosen from newer version members. (Bug #25876807)

- **Replication:** When `binlog_checksum=NONE` was set on a MySQL server after startup, and then Group Replication was started, if an error occurred, the server remained in `RECOVERING` state and could not be shut down. (Bug #25793366, Bug #85667)
- **Replication:** Now that XA transactions are prepared and committed in two parts, an issue with statement-based replication has been identified. If two XA transactions committed on the master are being prepared on the slave in the inverse order, locking dependencies can occur that cannot be safely resolved. The issue is not present with row-based replication.

XA transactions are therefore now considered unsafe for statement-based replication.

- When `binlog_format = STATEMENT`, a warning is issued for DML statements inside XA transactions, and replication might fail with deadlock on slaves.
- When `binlog_format = MIXED`, DML statements inside XA transactions are logged using row-based replication.
- When `binlog_format = ROW`, DML statements inside XA transactions are logged as before.

(Bug #25786490, Bug #85639)

- **Replication:** The following Performance Schema replication tables now continue to be populated when the Performance Schema is disabled:
 - `replication_connection_configuration`
 - `replication_connection_status`
 - `replication_applier_configuration`

- `replication_applier_status`
- `replication_applier_status_by_coordinator`
- `replication_applier_status_by_worker`

The exception is local timing information (start and end timestamps for transactions) in the replication tables `replication_connection_status`, `replication_applier_status_by_coordinator`, and `replication_applier_status_by_worker`. This information is not collected when the Performance Schema is disabled. (Bug #25694813)

- **Replication:** In a Group Replication setup where circular asynchronous replication was implemented between members of different replication groups, view change log events were repeatedly replicated between the groups with new generated GTIDs each time. The fix ensures that view change log events are ignored outside the named replication group where they occur, and never generate new GTIDs. (Bug #25674926)

References: See also: Bug #26049695, Bug #25928854, Bug #25721175.

- **Replication:** When first starting the MySQL server following an installation from RPM, password validation plugin is activated by default (true only for RPM installations). If binary logging was already enabled at this time, the activation was logged, even though plugin activations should not be recorded in the binary log. (Bug #25672750)
- **Replication:** The `XA START`, `XA END`, `XA COMMIT`, and `XA ROLLBACK` statements, used to perform XA transactions, were incrementing the counter for a `replicate-do-db` filter that named the relevant database. The counter is no longer incremented for these statements. (Bug #25669344)
- **Replication:** In a setup where single-primary Group Replication was combined with asynchronous replication, for example with S1 and S2 forming a group and with S2 and S3 functioning as master and slave, secondaries such as S2 were accepting transactions and these could then enter the group. The fix prevents secondaries creating an asynchronous replication channel when belonging to a single-primary group, and Group Replication cannot be started when asynchronous replication is running. (Bug #25574200, Bug #85047)

References: See also: Bug #86325, Bug #26078602.

- **Replication:** Group Replication failed to start if the `super_read_only` system variable was enabled. (Bug #25481287, Bug #84733)
- **Replication:** MySQL 8.0 servers were not able to join a replication group that contained MySQL 5.7 servers, and MySQL 5.7 servers in a replication group could not be upgraded to MySQL 8.0. (Bug #25477979)
- **Replication:** In the event that a member failed to join a group the member was not stopping and continued to accept transactions. To avoid this set your members to have `super_read_only=1` in the `my.cnf` file. Group Replication now checks for this setting upon successful start up and sets `super_read_only=0`. This ensures that members which do not successfully join a group cannot accept transactions. (Bug #25474736, Bug #84728)
- **Replication:** When `mysqlbinlog` output from a MySQL 5.7 server was applied on a MySQL 8.0 server, the MySQL 8.0 server generated its own timestamp for the `original_commit_timestamp`. The fix ensures that the value of `original_commit_timestamp` is correctly set to 0, meaning that the timestamp is not known. (Bug #25316086)

- **Replication:** If the server was started with the `--log-bin` option, discovery of one Performance Schema table with invalid structure caused all subsequently checked Performance Schema tables to be marked invalid as well. (Bug #25041396)
- **Replication:** Interleaved transactions could sometimes deadlock the slave applier when the transaction isolation level was set to `REPEATABLE-READ`. (Bug #25040331)
- **Replication:** The `slave_skip_errors` system variable did not permit error numbers larger than 3000. Thanks to Tsubasa Tanaka for the patch. (Bug #24748639, Bug #83184)
- **Replication:** The SQL state of `ER_TRANSACTION_ROLLBACK_DURING_COMMIT` was `HY00`, which is generic and not suitable for rollback. The SQL state has been modified to 40000. (Bug #24658431)
- **Replication:** `mysqlbinlog`, if invoked with the `--raw` option, does not flush the output file until the process terminates. But if also invoked with the `--stop-never` option, the process never terminates, thus nothing is ever written to the output file. Now the output is flushed after each event. (Bug #24609402)
- **Replication:** A memory leak in `mysqlbinlog` was fixed. The leak happened when processing fake rotate events, or when using `--raw` and the destination log file could not be created. The leak only occurred when processing events from a remote server. Thanks to Laurynas Biveinis for his contribution to fixing this bug. (Bug #24323288, Bug #82283)
- **Replication:** Loading and initialization of the Group Replication plugin failed if no `root` account was present. (Bug #24311527)

References: See also: Bug #25750822, Bug #25103980, Bug #83841.

- **Replication:** A slave server could lose events not yet applied when `MASTER_AUTO_POSITION=0`, both replication threads were stopped, and the applier delay was changed using `CHANGE MASTER TO MASTER_DELAY=N`. (Bug #23203678, Bug #81232)

References: See also: Bug #25340185, Bug #84375.

- **Replication:** Transmission of large GCS messages could take so long the sender appeared to have died. (Bug #22671846)
- **Replication:** Multi-threaded slaves could not be configured with small queue sizes using `slave_pending_jobs_size_max` if they ever needed to process transactions larger than that size. Any packet larger than `slave_pending_jobs_size_max` was rejected with the error `ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX`, even if the packet was smaller than the limit set by `slave_max_allowed_packet`.

With this fix, `slave_pending_jobs_size_max` becomes a soft limit rather than a hard limit. If the size of a packet exceeds `slave_pending_jobs_size_max` but is less than `slave_max_allowed_packet`, the transaction is held until all the slave workers have empty queues, and then processed. All subsequent transactions are held until the large transaction has been completed. The queue size for slave workers can therefore be limited while still allowing occasional larger transactions. (Bug #21280753, Bug #77406)

- **Replication:** An incident event that broke replication was not written to the binary log with a GTID, so that it was not possible to skip the event using `SET gtid_next=value`. Instead, it was necessary to set the relay log file and relay log positions directly; this meant that, when autopositioning was enabled, it was necessary first to disable it, then to set the relay log file and position, and finally to re-enable autopositioning.

Now in such cases MySQL writes the incident event into the statement cache, so that a GTID is generated and written for it prior to flushing, and that the slave applier works with the change. Then

users can skip the event using the SQL statement `SET gtid_next=value`, followed by `BEGIN` and `COMMIT`. (Bug #19594845)

- **Replication:** Issuing `SHOW SLAVE STATUS FOR CHANNEL 'group_replication_recovery'` following a restart of a server using group replication led to an unplanned shutdown. (Bug #85739, Bug #25813258)
- **Replication:** Setting an empty filter rule using `CHANGE REPLICATION FILTER` caused issues when running tests with UBSan. (Bug #85405, Bug #25702297)
- **Replication:** When the receiver thread held a lock on the relay log while queuing an event, clients executing `SHOW SLAVE STATUS` or `SHOW RELAYLOG EVENTS` were blocked. (Bug #85084, Bug #25584734)
- **Replication:** Indexes for the Performance Schema `replication_applier_status_by_worker` table worked correctly only for single threaded replication. This was due to the fact that the indexing relied exclusively on the SQL thread ID regardless of the existence of any workers. Now in such cases, the index also considers worker thread ID when multithreading is enabled on the slave. (Bug #84646, Bug #25444610)
- **Replication:** In certain cases, the master could write to the binary log a `last_committed` value which was smaller than it should have been. This could cause the slave to execute in parallel transactions which should not have been, leading to inconsistencies or other errors. (Bug #84471, Bug #25379659)
- **Replication:** When using `group_replication_ip_whitelist=AUTOMATIC`, IPs in the private network are permitted automatically, but some class C IP addresses were not being permitted correctly. (Bug #84329, Bug #25503458)
- **Replication:** When an existing `GTID_NEXT` transaction was assigned a conflicting GTID by the server, Group Replication generated an assert upon detecting two transactions with same GTID. This was because Group Replication generates the GTID after conflict detection, which is later than with master/slave replication. The fix relaxes some conditions to only be called when commit is done and a message has been added to alert you when a GTID has already been used. (Bug #84153, Bug #25232042)
- **Replication:** The replication applier thread returns Error 3002 `ER_INCONSISTENT_ERROR` when there is a difference between an expected error number and the actual error number. It is now possible to ignore this error by using 3002 with `slave_skip_errors`. (Bug #83186, Bug #24753281)
- **Replication:** MySQL lost its GTID position following a restart when a dump from `mysqldump` had been used to load data.

To keep this problem from occurring, the `mysql.gtid_executed` table is now excluded automatically from dumps made by `mysqldump`. (Bug #82848, Bug #24590891)

- **Replication:** It was possible to set `binlog_format` without causing an error when there were open temporary tables. (Bug #82467, Bug #24411680)
- **Replication:** Corruption of relay logs for one channel in multi-source replication caused good channels not to be initialized during a server restart. In addition, when run with `--skip-slave-start=false`, the server also failed to start slave threads for those channels which were in good condition, despite the fact that it should have started the slave threads for all good channels.

Now, regardless of any errors on other channels, the server attempts to create and initialize channels that are in good condition, and starts slave threads for the good channels if `--skip-slave-start` is disabled. As part of this fix, `START SLAVE` and `STOP SLAVE`, which are intended to operate on all channels, are also modified such that they continue executing on all good channels even if they find bad channels among them. (Bug #82209, Bug #24285104)

- **Replication:** The SQL thread was unable to GTID skip a partial transaction. (Bug #81119, Bug #25800025)
- **Replication:** It was possible for `FLUSH LOGS` to write a `Rotate_log_event` into an uninitialized log file. (Bug #80368, Bug #22732184)
References: See also: Bug #23531998, Bug #81734.
- **Microsoft Windows:** On Windows, the `Docs/INFO_SRC` file was missing the `build-date` entry. (Bug #25799855)
- **JSON:** `NULLIF()` failed with an assertion error in debug builds if the result from this function was used in a JSON context. This result can now be used safely as a JSON value.
This fix includes using `DEBUG_ASSERT()` for handling this error instead of `DEBUG_ABORT()`, which caused debugging problems on some platforms. (Bug #25818544)
References: See also: Bug #21383497, Bug #21383530.
- **JSON:** The internal `Json_array` now uses a `std::vector` instead of a `Preallocated_array` to store its elements, which reduces the amount of heap space required by them. (Bug #85877, Bug #25867454)
- The `mysqld_pre_systemd` script in RPM packages found the error log setting in option files if specified as `log-error` but not as `log_error`, though both are permitted. (Bug #26148391, Bug #86466)
- `ALTER TABLE ... ADD PARTITION` statements could cause a server exit. (Bug #26132947)
- `SET PERSIST` did not work for X Plugin system variables. (Bug #26115672)
- With the `gtid_mode` system variable not set in a `.cnf` option file or on the command-line, the server failed to restart after using `SET PERSIST` or `SET GLOBAL` to set `gtid_mode` to `ON_PERMISSIVE`. (Bug #26085712)
- During prepared statement execution, too many bytes of a buffer could be read. (Bug #26042934, Bug #85937)
- Dynamic privileges were not loaded when the server was started using the `--initialize` option. (Bug #26005645)
- Debian client packages were missing information about conflicts with `akonadi-backend-mysql` packages. (Bug #26002288)
- `mysqldump` could write database names in `USE` statements incorrectly. (Bug #25998635)
- If the `mysql_stmt_close()` C API function was called, it freed memory that later could be accessed if `mysql_stmt_error()`, `mysql_stmt_errno()`, or `mysql_stmt_sqlstate()` was called. To obtain error information after a call to `mysql_stmt_close()`, call `mysql_error()`, `mysql_errno()`, or `mysql_sqlstate()` instead. (Bug #25988681)
- Upgrading from MySQL 5.7 to MySQL 8.0 failed during data dictionary creation due to table names that exceeded the 64 character length limit. (Bug #25973237, Bug #86120)
- Queries could be cached incorrectly, leading to incorrect query results, under these circumstances: `InnoDB` table; rows are being inserted but have not yet been committed; a query uses the table as a base table in a derived table; the optimizer chooses to materialize the derived table. (Bug #25943038, Bug #86047)
- `mysqld` misbehaved if compiled with `AddressSanitizer` enabled and invoked with `--basedir=` (that is, with an empty option value). (Bug #25914296, Bug #85994)

- The `ST_LatFromGeohash()`, `ST_LongFromGeohash()`, and `ST_PointFromGeohash()` functions did not allow data that originated from a `CHAR` column. (Bug #25912557, Bug #85981)
- Attempting to drop a data file from a tablespace with multiple data files returned a duplicate file name error. (Bug #25858461)
- An attribute was added to data dictionary client method declarations to generate compilation warnings when return values are ignored. (Bug #25840927)
- The data dictionary class hierarchy was simplified. (Bug #25835968, Bug #85811)
- A restriction that prevented the data dictionary object update function (`Dictionary_client::update`) from being called twice on the same object was removed. (Bug #25833932, Bug #85800)
- On Ubuntu platforms, the MySQL service script did not terminate correctly if the user running the script did not have permission to access the data directory or PID file. (Bug #25825833)
- String comparison queries on the `INFORMATION_SCHEMA.CHARACTER_SETS.DESCRPTION` field were case sensitive, which is a regression from earlier releases. (Bug #25824297)
- Man pages for a few utilities were missing from Debian/Ubuntu packages. (Bug #25811814)
- An `INFORMATION_SCHEMA` view executed in `LOCK TABLES` mode, using the `SERIALIZABLE` isolation level, and with `autocommit` disabled, failed to use non-locking reads, which could block DDL operations or cause deadlocks. (Bug #25811413)
- Table locking failures could occur if tables were used by a trigger for which a trigger of the same name existed in another database and the database names differed only in lettercase. (Bug #25807393)
- An in-place MySQL upgrade failed if a `.TRG` file was missing the `created` line for the trigger definition. (This problem is unlikely to occur for upgrades to MySQL 8.0 from 5.7 because the `created` line is present for MySQL 5.7.2 and higher.) (Bug #25805260, Bug #85704)
- The `field-t` unit test failed to run with AddressSanitizer enabled. Thanks to Laurynas Biveinis for the patch. (Bug #25803823, Bug #85678)
- Debian client packages were missing information about conflicts with native packages. (Bug #25799475)
- The `CREATE_OPTIONS` column of the `INFORMATION_SCHEMA.TABLES` table was treated as having the `binary` collation, so functions such as `UPPER()` and `LOWER()` did not have the intended result. (Bug #25793429)
- After importing a table with `DECIMAL` column, accessing the table raised an assertion. (Bug #25792649)
- `ALTER TABLE` could fail when the default character set changed to `utf8mb4` due to incorrect column length calculations. (Bug #25779239, Bug #85614)
- If a server component `deinit()` method failed, the component was still unloaded. (Bug #25764325)
- Stored program execution could fail for `DEFINER` accounts with expired passwords even if they were locked and could not be used to connect to the server. Now `DEFINER` accounts are not checked for expired passwords if they are locked. (Bug #25741966)
- `DISTINCT` operations on temporary tables could produce incorrect results due to allocation of too-small comparison keys. (Bug #25740550, Bug #85518)
- In optimizer trace output, `num_tmp_files` did not actually indicate number of files. It has been renamed to `num_initial_chunks_spilled_to_disk` and indicates the number of chunks before any merging has occurred. (Bug #25733784, Bug #85487)

- The Perl path in `#!` lines at the beginning of Perl scripts has been adjusted to `/usr/local/bin/perl` for FreeBSD 11. (Bug #25719975)
- With the `IGNORE_SPACE` SQL mode enabled, syntax error messages always reported line 1 as the line number. (Bug #25717617)
- `mysqldump` failed to properly quote certain identifiers in SQL statements written to the dump output. (Bug #25717383)
- Dropping a tablespace without that associated tablespace file failed with an error stating that the tablespace dictionary object is invalid. (Bug #25717019)
- Client preauthorization by the server was missing a length check for a length-encoded string. (Bug #25714674)
- In debug builds, a `CREATE TABLESPACE` operation raised an invalid assertion when using the NDB storage engine. A validation function that checked for zero-length data files did not apply to NDB tablespaces and was removed. (Bug #25700242)
- For debug builds, `EXPORT_SET()` operations could raise an assertion for some arguments. (Bug #25688192)
- For debug builds, the assertion added for Bug#59686 was too strict and could be raised when it should not have been. (Bug #25685958)

References: See also: Bug #59686.

- A code refactoring in MySQL 8.0.1 caused several assertions to be raised in debug builds. (Bug #25669590, Bug #25669606, Bug #25669580, Bug #25688504)
- References: This issue is a regression of: Bug #25221172, Bug #84103.
- For some double-precision calculations, overflow could occur when calculating the exponent part. (Bug #25664323, Bug #85290)
- `SHOW INDEX` output was inconsistent for a `FULLTEXT` index defined on multiple columns. (Bug #25659276)
- During a group commit, the stack could incorrectly be reported exhausted, leading to stack overflow. (Bug #25656875)
- When the character set of one string comparison operand was a superset of the character set of the other operand, some comparisons were disallowed that should be permitted by converting the operand with the “smaller” character set to the “larger” character set. `utf8mb4` and `utf32` are considered to be a superset of any other encoding. (Bug #25642319, Bug #85224)
- An in-place (binary) upgrade could change the default `character_set_client` and `character_set_connection` variables for stored routines, possibly resulting in different routine behavior. (Bug #25633041)
- For debug builds, the partitioning handler could inappropriately evaluate generated column expressions in `UPDATE` statements, resulting in a raised assertion. (Bug #25615803, Bug #85179)
- In strict SQL mode, an `ER_TRUNCATED_WRONG_VALUE` error could be converted from a warning to an error but then ignored, leading to a raised assertion. (Bug #25586959, Bug #25586673)
- Failure occurred for `ALTER TABLE` on an `ARCHIVE` table containing a `NOT NULL` column having a geometry data type. For debug builds, an assertion was raised. For non-debug builds, an error occurred. (Bug #25582178, Bug #85059)

- For debug builds, an assertion could be raised for `DROP TRIGGER` of a trigger for a table used by a view. (Bug #25581925)
- For upgrades from MySQL 5.7 to MySQL 8.0, the server wrote unnecessary parsing warnings to the error log. These messages are now suppressed. (Bug #25518436, Bug #84889)
- The server exited abnormally attempting to access invalid memory. (Bug #25501659)
- The maximum length of the `name` field in the `column_type_elements` table in the data dictionary was extended from 255 bytes to 1020 bytes to accommodate long `ENUM` and `SET` values that could result from using a multibyte character set.

When using a multibyte character set, a single `ENUM` or `SET` element occupies a maximum of $M \times w$ bytes in the column type definition in the data dictionary, where M is the element literal length and w is the number of bytes required for the maximum-length character in the character set.

In prior releases, the maximum supported length of an individual `ENUM` or `SET` element depended on the number of elements in the type. Thus, there could be a single element with $(M \times w) = 64K$, or 64K elements with $(M \times w) = 1$.

The maximum supported length of an individual `ENUM` or `SET` element is now $M \leq 255$ and $(M \times w) \leq 1020$, regardless of the number of elements in the type. (Bug #25481355)

- With `mysqld` secured by TCP wrappers and the `hosts.allow` and `hosts.deny` files configured to restrict access from an IP address, connection attempts from that address resulted in too many messages to the error log. (Bug #25476479, Bug #84708)
- `mysqldump` no longer includes the `slave_master_info` and `slave_relay_log_info` tables in dumps of the `mysql` system database. Restoring a dump file containing these tables caused problems by changing the replication state improperly. (Bug #25469190)
- Changes made by calling `mysql_options()` to set `MYSQL_OPT_SSL_MODE` could be affected by later `mysql_options()` calls. Now setting `MYSQL_OPT_SSL_MODE` is unaffected by later `mysql_options()` calls. (Bug #25452210)
- A race condition could occur for `CREATE TABLE` statements with `DATA DIRECTORY` or `INDEX DIRECTORY` clauses. (Bug #25451091)
- With AddressSanitizer enabled, compiling the `keyring_file` plugin produced One-Definition Rule violations. (Bug #25448205)
- `ALTER TABLE ... MODIFY` on a `DATETIME NOT NULL` column using an `AFTER` clause resulted in an `ER_INVALID_USE_OF_NULL` error. (Bug #25385334)
- The range optimizer could create an incorrect query tree, resulting in a server exit. (Bug #25369742, Bug #25586531)
- `mysqld_failed` to start the server if the `--datadir` option was specified with a relative path name. (Bug #25364806)
- `XA PREPARE`, `XA ROLLBACK`, and `XA COMMIT` for a transaction from a disconnected session did not take a global commit lock and modified the binary log and InnoDB redo log even when `FLUSH TABLES WITH READ LOCK` was in effect. This could lead to inconsistent backups when backup tools assumed that the server was in a read-only state. (Bug #25364178, Bug #84442)
- `GROUP_CONCAT(DISTINCT)` returned nonunique values if the data size was greater than the value of the `tmp_table_size` system variable. (Bug #25331425, Bug #84320)

- The fix for Bug #78777 had different effects depending on whether the Performance Schema is enabled. (Bug #25309017, Bug #84305)

References: This issue is a regression of: Bug #78777.

- An aggregate function in some nested queries could cause a server exit. (Bug #25303711)
- Virtual generated column expressions that used the `BIN_TO_UUID()`, `CONV()`, or `HEX()` functions could cause problems if the connection character set was changed. In this context, the table character set is now used for these functions regardless of connection character set. (Bug #25287633)
- The `Rewriter` plugin did not perform locking properly if the `read_only` system variable was enabled. (Bug #25264253)
- With `read_only` enabled, creation of non-`TEMPORARY` tables by non-`SUPER` users was permitted under certain conditions. (Bug #25250768)
- For a table having a `TIMESTAMP` or `DATETIME` column having a default of `CURRENT_TIMESTAMP`, the column could be initialized to `0000-00-00 00:00:00` if the table had a `BEFORE INSERT` trigger. (Bug #25209512, Bug #84077)
- On Windows, `Time` values in `SHOW PROCESSLIST` output drifted higher over time. (Bug #25101724, Bug #83019)
- An assertion was raised when a create function at server bootstrap attempted to evaluate non-resolved expressions. (Bug #24961932)
- The number of data dictionary cache lookups was reduced, and `ALTER TABLE` handling of triggers was improved help ensure trigger persistence in case of a server exit during `ALTER TABLE` processing. (Bug #24930129, Bug #83473)
- Certain queries against `InnoDB` tables that used a primary key and a subquery could return incorrect results if the `index_merge_intersection` flag of the `optimizer_switch` system variable was enabled. (Bug #24829050, Bug #79675)
- On x86 machines, the `uint3korr()` macro read 4 bytes of data instead of the intended 3 bytes. (Bug #24807826, Bug #83264)
- An assertion was raised during a fetch operation by the memcached plugin. (Bug #24605783)
- Queries that contained `UNION` in a subquery and `GROUP BY` could return incorrect results. (Bug #24595639)
- Some syntactically incorrect `CREATE INDEX` statements could cause a server exit rather than a syntax error. (Bug #24593992)
- An in-place MySQL upgrade failed if the `innodb_table_stats` or `innodb_index_stats` tables were not present in the `mysql` system database. (This problem should not occur for upgrades to MySQL 8.0 from 5.7 because those tables are present in MySQL 5.7.) (Bug #24557143)
- Incorrect behavior could occur for `INSERT` statements executed in stored-program or prepared-statement context, if the `VALUES` part of an `ON DUPLICATE KEY UPDATE` clause referred to a `BLOB` value in the `INSERT` column list. (Bug #24538207, Bug #25361251, Bug #25530880, Bug #25684790)
- `systemd` support scripts in Debian packages contained hardcoded references to the data directory, making it difficult to change the data directory using `--datadir`. (Bug #24398446, Bug #82417)
- MySQL failed to compile under macOS 10.10.5 using Clang. (Bug #24352163, Bug #82340)

- If a `REPLACE` statement tried to update a row in a table containing a virtual generated column of type `BLOB`, subsequent DML statements could behave incorrectly. (Bug #23573575)
- Some `PROXY` grants were not replicated to slaves, causing incorrect replication. (Bug #23289541, Bug #81424)
- `EXPLAIN` for single-table `UPDATE` or `DELETE` statements could raise an assertion attempting to optimize away subqueries. (Bug #23209903)
- The help output from `mysqlxtest` has been improved. (Bug #23107137, Bug #81086)
- For builds with AddressSanitizer enabled, the `ST_Simplify()` function could attempt to use already freed memory. (Bug #23023817)
- Compiler flags were adjusted to eliminate numerous warnings that occurred when compiling the `keyring_file` plugin using Clang. (Bug #22834591, Bug #80524)
- If enabling the Event Scheduler caused an event defined as `ON COMPLETION NOT PRESERVE` to be dropped because its execution time had passed, the drop event was not written to the binary log, causing slaves not to replicate it and replication failure if an event of the same name was created later. (Bug #22150112)
- For some `CREATE TABLE ... SELECT` statements, adding an `ORDER BY` clause changed column data types, or caused an assertion to be raised for debug builds. (Bug #16833464)
- `LOAD XML INFILE` performance became noticeably slower when the XML file being read contained a great many spaces, such as those introduced by indenting or pretty-printing. Now all leading whitespace is trimmed from each such value before reading it into memory. (Bug #16212207)

Changes in MySQL 8.0.1 (2017-04-10, Development Milestone)



Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.



Note

This release makes several important changes in Unicode character set support. In particular, the default character set has changed from `latin1` to `utf8mb4`.

- [C API Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Configuration Notes](#)
- [Data Dictionary Notes](#)

- [Deprecation and Removal Notes](#)
- [Optimizer Notes](#)
- [Packaging Notes](#)
- [Parser Notes](#)
- [Performance Schema Notes](#)
- [Plugin Service Notes](#)
- [Security Notes](#)
- [Spatial Data Support](#)
- [SQL Syntax Notes](#)
- [Test Suite Notes](#)
- [X Plugin Notes](#)
- [Platform-Specific Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

C API Notes

- The C API implemented by `libmysqlclient` now includes a client interface for reading a stream of replication events from a MySQL server binary log. For more information, see [C API Binary Log Interface](#)

Character Set Support

- **Important Change:** The default character set has changed from `latin1` to `utf8mb4`. These system variables are affected:
 - The default value of the `character_set_server` and `character_set_database` system variables has changed from `latin1` to `utf8mb4`.
 - The default value of the `collation_server` and `collation_database` system variables has changed from `latin1_swedish_ci` to `utf8mb4_0900_ai_ci`.

As a result, the default character set and collation for new objects differ from previously unless an explicit character set and collation are specified. This includes databases and objects within them, such as tables, views, and stored programs. One way to preserve the previous defaults is to start the server with these lines in the `my.cnf` file:

```
[mysqld]
character_set_server=latin1
collation_server=latin1_swedish_ci
```

- Performance of UCA 9.0.0-based collations (for example, `utf8mb4_0900_ai_ci`) was improved. These collations are now faster than any other UCA collations.

Additionally, the `max_length_for_sort_data` system variable default value was increased from 1024 to 4096. (Bug #24823885, Bug #83319)

- The default collation for the `utf8mb4` character set has changed from `utf8mb4_general_ci` to `utf8mb4_0900_ai_ci`; `utf8mb4_general_ci` does not handle characters outside the Basic Multilingual Plane (BMP) correctly. (Bug #24742157)
- The pad attribute for Unicode 9.0.0 collations was changed from PAD SPACE to NO PAD. Consequently, these collations now treat spaces at the end of strings like any other character. The affected collations have names that contain the string `_0900_`.

Comparisons of `VARCHAR` columns that have a 9.0.0 collation differ from other collations with respect to trailing spaces. For example, `'a'` and `'a '` compare as different strings, not the same string. Example:

```
mysql> SET NAMES 'latin1' COLLATE 'latin1_swedish_ci';
mysql> SELECT 'a' = 'a ';
+-----+
| 'a' = 'a ' |
+-----+
|          1 |
+-----+
mysql> SET NAMES 'utf8mb4' COLLATE 'utf8mb4_0900_ai_ci';
mysql> SELECT 'a' = 'a ';
+-----+
| 'a' = 'a ' |
+-----+
|          0 |
+-----+
```

The `INFORMATION_SCHEMA.COLLATIONS` table now has a `PAD_ATTRIBUTE` column that indicates the pad attribute for each collation.

A problem with the `latin1_de` collation involving early weight string truncation has been corrected. The only likely effect is for `WEIGHT_STRING()` function results.

- Complementing earlier work in MySQL 8.0.0 to add case-insensitive and accent-insensitive collations for the `utf8mb4` Unicode character set, new case-sensitive and accent-sensitive collations have been added. The general collation is named `utf8mb4_0900_as_cs`, and there are language-specific collations with characteristics similar to `utf8mb4_0900_as_cs` except that language-specific rules take precedence where applicable. The language-specific collations are indicated by ISO 639-1 language codes in the collation name, as shown in the following table. In two cases the language code has an additional item that denotes a variant (German phone book order, Traditional Spanish).



Note

`utf8mb4_ja_0900_as_cs` is the first Japanese language-specific collation available for Unicode in MySQL.

Table 2 utf8mb4 UCA 9.0.0 Language-Specific Collations

Language	Collation
Croatian	<code>utf8mb4_hr_0900_as_cs</code>
Czech	<code>utf8mb4_cs_0900_as_cs</code>
Danish	<code>utf8mb4_da_0900_as_cs</code>
Esperanto	<code>utf8mb4_eo_0900_as_cs</code>
Estonian	<code>utf8mb4_et_0900_as_cs</code>
German phone book order	<code>utf8mb4_de_pb_0900_as_cs</code>
Hungarian	<code>utf8mb4_hu_0900_as_cs</code>

Language	Collation
Icelandic	utf8mb4_is_0900_as_cs
Japanese	utf8mb4_ja_0900_as_cs
Latvian	utf8mb4_lv_0900_as_cs
Lithuanian	utf8mb4_lt_0900_as_cs
Polish	utf8mb4_pl_0900_as_cs
Classical Latin	utf8mb4_la_0900_as_cs
Romanian	utf8mb4_ro_0900_as_cs
Slovak	utf8mb4_sk_0900_as_cs
Slovenian	utf8mb4_sl_0900_as_cs
Modern Spanish	utf8mb4_es_0900_as_cs
Traditional Spanish	utf8mb4_es_trad_0900_as_cs
Swedish	utf8mb4_sv_0900_as_cs
Turkish	utf8mb4_tr_0900_as_cs
Vietnamese	utf8mb4_vi_0900_as_cs

[utf8mb4_0900_as_cs](#) also works as an accent-sensitive, case-sensitive collation for the languages in the following table.

Table 3 Languages for Which [utf8mb4_0900_as_cs](#) is Suitable

Language Name	Language Code
German (dictionary order)	de
English	en
French (locale fr_FR)	fr
Irish Gaelic	ga
Indonesian	id
Italian	it
Luxembourgian	lb
Malay	ms
Dutch	nl
Portuguese	pt
Swahili	sw
Zulu	zu

[utf8mb4_0900_as_cs](#) is suitable for French French (locale [fr_FR](#)) but not for Canadian French (locale [fr_CA](#)). For Canadian French, [utf8mb4_0900_ai_ci](#) is suitable. The reason for the difference between [ai_ci](#) and [as_cs](#) collations is that Canadian French has a different order of accented characters than French French.

[utf8mb4_da_0900_as_cs](#) also works as an accent-sensitive, case-sensitive collation for the languages in the following table.

Table 4 Languages for Which utf8mb4_da_0900_as_cs is Suitable

Language Name	Language Code
Norwegian	no
Norwegian Bokmål	nb
Norwegian Nynorsk	nn

The nonlanguage-specific `utf8mb4_0900_as_cs` and language-specific `utf8mb4_LANG_0900_as_cs` Unicode collations each have these characteristics:

- The collation is based on Unicode Collation Algorithm (UCA) 9.0.0 and Common Locale Data Repository (CLDR) v30, is accent sensitive, and case sensitive. These characteristics are indicated by `_0900`, `_as`, and `_cs` in the collation name. Exception: `utf8mb4_la_0900_as_cs` is not based on CLDR because Classical Latin is not defined in CLDR.
- The collation works for all characters in the range [U+0, U+10FFFF].
- If the collation is not language specific, it sorts all characters, including supplemental characters, in default order (described following). If the collation is language specific, it sorts characters of the language correctly according to language-specific rules, and characters not in the language in default order.
- By default, the collation sorts characters having a code point listed in the DUCET table (Default Unicode Collation Element Table) according to the weight value assigned in the table. The collation sorts characters not having a code point listed in the DUCET table using their implicit weight value, which is constructed according to the UCA.

The collation sorts on all three DUCET weight levels, including the tertiary level. This compares with accent-insensitive, case-insensitive collations, which sort only on the primary and secondary levels.

- For non-language-specific collations, characters in contraction sequences are treated as separate characters. For language-specific collations, contractions might change character sorting order.

For more information, see [Unicode Character Sets](#).

Compilation Notes

- **Incompatible Change:** The `my_bool` type is no longer used in MySQL source code. Any third-party code that used this type to represent C boolean variables should use the `bool` or `int` C type instead.



Note

The change from `my_bool` to `bool` means that the `mysql.h` header file now requires a C++ or C99 compiler to compile.

(Bug #25597667)

- **InnoDB:** A LOB page deletion function failed to delete LOB pages. (Bug #24480254)
- Windows builds now use the default runtime libraries (builds use the `/MD` flag). (Bug #25611609)
- **CMake** support was added for compiling with Developer Studio 12.6. (Bug #25384295)
- The rapidjson library included in MySQL distributions was upgraded to version 1.1.0. (Bug #24947436, Bug #83515)

- The required version of the Boost library for server builds has been raised to 1.63.0. (Bug #24579061, Bug #82834, Bug #25126144, Bug #83905)
- For GCC versions higher than 4.4, `-fno-expensive-optimizations` was replaced with `-ffp-contract=off`, which has the effect of enabling more optimizations. Thanks to Alexey Kopytov for the patch. (Bug #24571672, Bug #82760)
- For building MySQL 8.0, the minimum required version of CMake is now 3.2.3 on Windows, 3.4.0 on Solaris, and 2.8.12 otherwise. (Bug #24481181, Bug #82628)

References: See also: Bug #24687701.

- Work was done to clean up the source code base, including: Removing unneeded CMake checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

Configuration Notes

- In source distributions, several utilities previously in the `extra` directory have been moved to the new `utilities` directory. (Bug #25416084)
- Two new CMake options are available for debugging. `LINK_RANDOMIZE` indicates whether to randomize the order of symbols in the `mysqld` binary (default `OFF`), and `LINK_RANDOMIZE_SEED` specifies a seed value for `LINK_RANDOMIZE`. (Bug #25336715)
- MySQL failed to compile if `-DENABLE_DEBUG_SYNC=OFF` AND `-DWITH_DEBUG=ON` were both given. The `ENABLE_DEBUG_SYNC` option has been removed and enabling `WITH_DEBUG` enables Debug Sync. (Bug #18374703)
- These changes were made with respect to persisted system variables:
 - A new statement, `RESET PERSIST`, enables removal of variable settings that were persisted with `SET PERSIST` to the `mysqld-auto.cnf` file that stores persisted global system variable settings.
 - Using `SET PERSIST` (or `@@persist.`) to set a global variable to `DEFAULT` or to the variable literal default value previously also added a setting for the variable to the `mysqld-auto.cnf` file if it was not present, or removed it from `mysqld-auto.cnf` if it was present. Now the assignment always adds a setting for the variable to the `mysqld-auto.cnf` file; to remove the setting from the file, use `RESET PERSIST`.
 - A new Performance Schema table, `persisted_variables`, provides an SQL interface to the `mysqld-auto.cnf` file, enabling the file contents to be inspected at runtime using `SELECT` statements.
 - The Performance Schema `variables_info` table has new columns showing when and by which user each system variable was most recently set.

For more information, see [Using System Variables](#), and [RESET PERSIST Syntax](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema.

References: See also: Bug #24522064.

- The default value of the `query_cache_size` system variable has been reduced from 1M to 0. Consequently no query cache buffer is allocated by default.

Data Dictionary Notes

- SQL-layer code and the storage engine API were extended to provide data dictionary support for:
 - Storing information about auxiliary columns and keys that [InnoDB](#) adds to tables implicitly, storage engine-private data and object identifiers for tables and tablespaces, and [InnoDB](#) full-text search auxiliary tables and other similar, implicitly created objects.
 - Combining data dictionary updates, storage engine changes, and binary log writes for DDL operations into atomic transactions.

A behavior change for [DROP TABLE](#) occurs if any named tables do not exist. Previously, the statement returned an error indicating which tables did not exist and it was unable to drop, but also dropped the tables that did exist. Now the statement still indicates which tables did not exist, but fails with an error and no changes are made.

For more information, see [Atomic Data Definition Statement Support](#), and [WL#7743](#).

- These [INFORMATION_SCHEMA](#) tables have been reimplemented as views on data dictionary tables:

```
EVENTS
PARAMETERS
ROUTINES
TRIGGERS
```

Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. For example, the server no longer must create a temporary table for each query of the [INFORMATION_SCHEMA](#) table.

Improvements for those tables also apply to [SHOW](#) statements that display information corresponding to the [INFORMATION_SCHEMA](#) tables. For example, [SHOW TRIGGERS](#) displays the same information as the [TRIGGERS](#) table.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes.

Deprecation and Removal Notes

- The deprecated [PROCEDURE ANALYSE\(\)](#) syntax has been removed.
- The deprecated `libmysqld` embedded server library has been removed, along with:
 - The `mysql_config --libmysqld-libs`, `--embedded-libs`, and `--embedded` options
 - The CMake `WITH_EMBEDDED_SERVER`, `WITH_EMBEDDED_SHARED_LIBRARY`, and `INSTALL_SECURE_FILE_PRIV_EMBEDDED` options
 - The (undocumented) `mysql --server-arg` option
 - The `mysqltest --embedded-server`, `--server-arg`, and `--server-file` options
 - The `mysqltest_embedded` and `mysql_client_test_embedded` test programs
- The deprecated `--temp-pool` server option has been removed.
- Support for DTrace has been removed.

Optimizer Notes

- **InnoDB:** MySQL now supports descending indexes: `DESC` in an index definition is no longer ignored but causes storage of key values in descending order. Previously, indexes could be scanned in reverse order but at a performance penalty. A descending index can be scanned in forward order, which is more efficient. Descending indexes also make it possible for the optimizer to use multiple-column indexes when the most efficient scan order mixes ascending order for some columns and descending order for others. For more information, see [Descending Indexes](#).



Note

Previously, relying on implicit `GROUP BY` sorting was deprecated but `GROUP BY` did sort. `GROUP BY` no longer sorts by default, so query results may differ from previous MySQL versions. To produce a given sort order, use explicit `ASC` or `DESC` designators for `GROUP BY` columns or provide an `ORDER BY` clause.

- Optimizer trace output now includes more information about `filesort` operations, such as key and payload size and why add-on fields are not packed. (Bug #25246184, Bug #84180)
- Previously, invisible indexes were supported only for the `InnoDB` storage engine. Invisible indexes are now storage engine neutral (supported for any engine). (Bug #23541244)
- A `GROUPING()` function has been introduced for use in `GROUP BY` queries that include a `WITH ROLLUP` modifier. This function enables you to distinguish `NULL` values for super-aggregate rows in the query result from `NULL` values in regular grouped rows. For more information, see [Miscellaneous Functions](#), and [GROUP BY Modifiers](#). Thanks to Zhe Dong for a patch that was partially used to implement this feature.



Note

`GROUPING` now is a reserved word and cannot be used as an identifier without identifier quoting.

(Bug #11754449, Bug #46053)

- The optimizer now supports `INDEX_MERGE` and `NO_INDEX_MERGE` hints to affect use of the Index Merge access method. Examples:

```
SELECT /*+ INDEX_MERGE(t1 f1, f2, f3) */ f2 FROM t1
  WHERE f1 = 'o' AND f2 = f3 AND f3 <= 4;
SELECT /*+ NO_INDEX_MERGE(t1 f2, f3) */ f1 FROM t1
  WHERE (f2 = 5 OR f3 = 'c') AND (f1 = 4 OR f4 = 'f');
```

For more information, see [Index Merge Optimization](#), and [Optimizer Hints](#).

- The `server_cost` and `engine_cost` optimizer cost model tables in the `mysql` system database now include a `default_value` column that contains the default value for each cost table estimate. This column is a read-only generated column that retains its value even if the associated cost estimate is changed. For more information, see [The Optimizer Cost Model](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database.

- In the optimizer cost model `engine_cost` table, the cost for memory access was decreased relative to the cost for disk access. An implication of this change is that the execution plan may change between two runs of the same query. For example, at server startup before data has been read into the buffer pool, you may get a different plan than after the query has been run because then the data will be in memory.

- MySQL now supports common table expressions, both nonrecursive and recursive. Common table expressions enable use of named temporary result sets, implemented by permitting a `WITH` clause preceding `SELECT` statements and certain other statements. For more information, see [WITH Syntax \(Common Table Expressions\)](#).

**Note**

`RECURSIVE` now is a reserved word and cannot be used as an identifier without identifier quoting.

- The optimizer now supports hints that enable specifying the order in which to join tables. For more information, see [Optimizer Hints](#).

Packaging Notes

- **Microsoft Windows:** *Reminder:* MySQL 8.0 requires the Microsoft Visual C++ 2015 Redistributable Package to run on Windows platforms. Users should make sure the package has been installed on the system before starting the server. The package is available at the [Microsoft Download Center](#).
- Changes in RPM package structure require a larger set of packages to be removed to install MySQL Server cleanly. (Bug #25603087)
- To avoid potential race conditions, Debian packages now use the GNU `install` utility rather than a combination of `mkdir`, `touch`, and `chown`. (Bug #25258829)
- New Debian/Ubuntu packages named `server-core` and `client-core` are now available. These packages contain binaries only, without configuration or service scripts, directory setup, man pages, and so forth. They are installed automatically with the standard packages, but when installed by themselves do not result in a functioning MySQL setup. They may be of interest for user which wish to customize their own installation setup. (Bug #25146364)
- `scripts/mysql_security_commands.sql` and `support-files/mysql.server-sys5.sh` are no longer used and have been removed from MySQL distributions. (Bug #24756400, Bug #24756442)
- RPM packages now are built with `-DWITH_NUMA=ON` for platforms with NUMA support: OEL higher than EL5, Fedora, SLES, Docker. (Bug #24689078)
- The `my-default.cnf.sh` file (used to produce a default `my-default.cnf` or `my-default.ini` file) is no longer included in source distributions and `my-default.cnf` and `my-default.ini` are no longer included in or installed by distribution packages. (Bug #22525354)
- The unused and undocumented `archive_reader` and `archive_test` utilities have been removed from MySQL source distributions. (Bug #12818207, Bug #62014)

Parser Notes

- The parser no longer considers `\N` as a synonym for `NULL` in SQL statements. Use `NULL` instead.

This change does not affect text file import or export operations performed with `LOAD DATA INFILE` or `SELECT ... INTO OUTFILE`, for which `NULL` continues to be represented by `\N`. See [LOAD DATA INFILE Syntax](#).

Performance Schema Notes

- The Performance Schema maintains statement event summary tables that contain information about minimum, maximum, and average statement latency. Those tables permit high-level assessment of system performance. To permit assessment at a more fine-grained level, the Performance Schema now

also collects histogram data for statement latencies. These histograms provide additional insight into latency distributions and are available in these tables:

- `events_statements_histogram_by_digest`: Statement histograms per schema and digest value
- `events_statements_histogram_global`: Statement histogram summarized globally

For more information, see [Statement Histogram Summary Tables](#).

In addition, the `events_statements_summary_by_digest` table now has columns that provide percentile information about statement latencies. For more information, see [Statement Summary Tables](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema.

- System and status variable information is no longer maintained in the `INFORMATION_SCHEMA`. These tables have been removed: `GLOBAL_VARIABLES`, `SESSION_VARIABLES`, `GLOBAL_STATUS`, `SESSION_STATUS`. Use the corresponding Performance Schema tables instead. See [Performance Schema System Variable Tables](#), and [Performance Schema Status Variable Tables](#). In addition, the `show_compatibility_56` system variable has been removed. It was used in the transition period during which system and status variable information in `INFORMATION_SCHEMA` tables was moved to Performance Schema tables, and is no longer needed. These status variables have been removed: `Slave_heartbeat_period`, `Slave_last_heartbeat`, `Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`. The information they provided is available in Performance Schema tables; see [Migrating to Performance Schema System and Status Variable Tables](#).
- The deprecated `INFORMATION_SCHEMA INNODB_LOCKS` and `INNODB_LOCK_WAITS` tables have been removed. To obtain InnoDB data lock information, use the Performance Schema `data_locks` and `data_lock_waits` tables instead. Alternatively, use the `sys` schema `innodb_lock_waits` and `x$innodb_lock_waits` views, which have been reimplemented to use the new Performance Schema tables rather than the removed `INFORMATION_SCHEMA` tables.

The Performance Schema tables expose what data locks exist, which transaction holds each lock, and which transaction lock requests are blocked by other locks. For InnoDB, the `data_locks` table reports all data locks, in contrast to the `INNODB_LOCKS` table, which reported only waited-for data locks. For more information, see [The data_locks Table](#), and [The data_lock_waits Table](#). For information about mapping old table columns to new table columns, see [The INFORMATION_SCHEMA INNODB_LOCKS Table](#), and [The INFORMATION_SCHEMA INNODB_LOCK_WAITS Table](#).

The tables differ in the privileges required: The `INFORMATION_SCHEMA` tables require the global `PROCESS` privilege. The Performance Schema tables require the usual Performance Schema privilege of `SELECT` on the table to be selected from.

The `sys` schema `innodb_lock_waits` and `x$innodb_lock_waits` views can be used without change, with the exception that the `locked_table` column containing combined schema/table name values has been removed and replaced with `locked_table_schema` and `locked_table_name` columns that contain separate schema and table name values. In addition, these views now contain `locked_table_partition` and `locked_table_subpartition` columns. For more information, see [The innodb_lock_waits and x\\$innodb_lock_waits Views](#),

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema and the `sys` schema.

Plugin Service Notes

- MySQL Server 8.0 includes a component-based infrastructure for improving server extensibility; see [MySQL Server Components](#). However, MySQL plugins use an interface that predates the component interface. A new plugin service, `plugin_registry_service`, enables plugins to access the component registry and its services.

Security Notes

- **Security Fix:** OpenSSL is ending support for version 1.0.1 in December 2016; see <https://www.openssl.org/policies/releasestrat.html>. Consequently, MySQL Commercial Server builds now use version 1.0.2 rather than version 1.0.1, and the linked OpenSSL library for the MySQL Commercial Server has been updated from version 1.0.1 to version 1.0.2j. For a description of issues fixed in this version, see <https://www.openssl.org/news/vulnerabilities.html>.

This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead.

- **Incompatible Change:** For `STANDALONE` and `WIN` builds, the default `secure_file_priv` value has changed from the empty string to `NULL`. This is a secure-by-default setting because it disables import and export operations. To permit those operations, set `secure_file_priv` to the path name of the directory to use for those operations. (Bug #24679907, Bug #24695274, Bug #24707666)
- **Incompatible Change:** These changes were made to `mysqld_safe`:
 - Unsafe use of `rm` and `chown` in `mysqld_safe` could result in privilege escalation. `chown` now can be used only when the target directory is `/var/log`. An incompatible change is that if the directory for the Unix socket file is missing, it is no longer created; instead, an error occurs. Due to these changes, `/bin/bash` is required to run `mysqld_safe` on Solaris. `/bin/sh` is still used on other Unix/Linux platforms.
 - The `--ledir` option now is accepted only on the command line, not in option files.
 - `mysqld_safe` ignores the current working directory.

Other related changes:

- Initialization scripts that invoke `mysqld_safe` pass `--basedir` explicitly.
- Initialization scripts create the error log file only if the base directory is `/var/log` or `/var/lib`.
- Unused systemd files for SLES were removed.

(Bug #24483092, Bug #25088048, Bug #25378439, Bug #25378565)

References: See also: Bug #24464380, Bug #24388753, Bug #24619033, Bug #82920.

- **Incompatible Change:** Server components and plugins now can define privileges dynamically (at runtime), which enables them to create privileges specifically associated with the capabilities they implement.

Implementation of dynamic privileges enables DBAs to begin migrating away from the `SUPER` privilege. In the past, `SUPER` has been associated with a wide and growing variety of operations, not all related to each other. Many operations covered by `SUPER` now are also associated with a dynamic privilege of more limited scope, and that privilege is registered by the server component or plugin that implements the operation. Each such operation that previously required the `SUPER` privilege can be permitted to an account by granting the associated dynamic privilege rather than `SUPER`. For example, a user who must be able to modify global system variables can be granted `SYSTEM_VARIABLES_ADMIN` rather than `SUPER`.

This change improves security by enabling DBAs to avoid granting `SUPER` and tailor user privileges more closely to the operations permitted. `SUPER` is now deprecated and will be removed in a future version of MySQL.

Dynamic privileges apply only at the global level. The server stores information about current assignments of dynamic privileges to user accounts in `mysql.global_grants`, a new table in the `mysql` system database.

For more information, see [Static Versus Dynamic Privileges](#). That discussion includes instructions for migrating accounts away from `SUPER` to dynamic privileges.

Incompatibility: `SHOW GRANTS` no longer displays `ALL PRIVILEGES` in its global-privileges output because the meaning of `ALL PRIVILEGES` at the global level varies depending on which dynamic privileges are defined. Instead, `SHOW GRANTS` explicitly lists each granted global privilege. For details, see [SHOW GRANTS Syntax](#). Applications that process `SHOW GRANTS` output should be adjusted accordingly.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database.

- yaSSL was upgraded to version 2.4.2. This upgrade corrects issues with: Potential AES side channel leaks; DSA padding for unusual sizes; the `SSL_CTX_load_verify_locations()` OpenSSL compatibility function failing to handle long path directory names. (Bug #24512715, Bug #24740291)
- MySQL Server now includes a plugin library that enables administrators to introduce an increasing delay in server response to clients after a certain number of consecutive failed connection attempts. This capability provides a deterrent that slows down brute force attacks that attempt to access MySQL user accounts. For more information, see [The Connection-Control Plugins](#).

Spatial Data Support

- **Incompatible Change:** Previously, `ST_Distance()` supported only Cartesian spatial reference systems (SRSS). `ST_Distance()` now detects geometry arguments in a geographic (ellipsoidal) SRS and returns the geodetic distance on the ellipsoid in meters. The only permitted geographic argument types are `Point` and `Point`, or `Point` and `MultiPoint` (in any argument order). Calculations for projected SRSs and SRID 0 remain the same. For more information, see [Spatial Relation Functions That Use Object Shapes](#).



Note

If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- **Incompatible Change:** These functions previously accepted either WKB strings or geometry arguments. Geometry arguments are no longer permitted and produce an error.
 - `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()`
 - `ST_GeomFromWKB()`, `ST_GeometryFromWKB()`
 - `ST_LineFromWKB()`, `ST_LinestringFromWKB()`
 - `ST_MLineFromWKB()`, `ST_MultiLinestringFromWKB()`
 - `ST_MPointFromWKB()`, `ST_MultiPointFromWKB()`

- `ST_MPolyFromWKB()`, `ST_MultiPolygonFromWKB()`
- `ST_PointFromWKB()`
- `ST_PolyFromWKB()`, `ST_PolygonFromWKB()`

For information about migrating queries that refer to those functions away from using geometry arguments to using WKB arguments, see [Functions That Create Geometry Values from WKB Values](#).

- **Incompatible Change:** These functions now interpret latitude and longitude coordinates as in the order specified by the spatial reference system. The functions also accept an optional argument to override the default axis order.

- `ST_GeomCollFromText()`, `ST_GeometryCollectionFromText()`, `ST_GeomCollFromText()`
- `ST_GeomFromText()`, `ST_GeometryFromText()`
- `ST_LineFromText()`, `ST_LinestringFromText()`
- `ST_MLineFromText()`, `ST_MultiLinestringFromText()`
- `ST_MPointFromText()`, `ST_MultiPointFromText()`
- `ST_MPolyFromText()`, `ST_MultiPolygonFromText()`
- `ST_PointFromText()`
- `ST_PolyFromText()`, `ST_PolygonFromText()`
- `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()`
- `ST_GeomFromWKB()`, `ST_GeometryFromWKB()`
- `ST_LineFromWKB()`, `ST_LinestringFromWKB()`
- `ST_MLineFromWKB()`, `ST_MultiLinestringFromWKB()`
- `ST_MPointFromWKB()`, `ST_MultiPointFromWKB()`
- `ST_MPolyFromWKB()`, `ST_MultiPolygonFromWKB()`
- `ST_PointFromWKB()`
- `ST_PolyFromWKB()`, `ST_PolygonFromWKB()`

For more information, see [Functions That Create Geometry Values from WKT Values](#), and [Functions That Create Geometry Values from WKB Values](#).



Note

If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- **Incompatible Change:** These functions now return latitude and longitude coordinates in the order specified by the spatial reference system that applies to the geometry value argument. The functions also accept an optional argument to override the default axis order.

- `ST_AsBinary()`, `ST_AsWKB()`
- `ST_AsText()`, `ST_AsWKT()`

For more information, see [Geometry Format Conversion Functions](#).



Note

If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- The `st_spatial_reference_systems` data dictionary table that stores information about spatial reference systems other than SRID 0 has been upgraded from EPSG Dataset 8.7 to 9.0. Also, the table contents are now exposed through the `INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS` table, which is implemented as a view on the data dictionary. For more information, see [The INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS Table](#).

MySQL now has an `INFORMATION_SCHEMA ST_GEOMETRY_COLUMNS` table that provides information about table columns that store spatial data. The new table is implemented as a view on the existing `INFORMATION_SCHEMA COLUMNS` table. For more information, see [The INFORMATION_SCHEMA ST_GEOMETRY_COLUMNS Table](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these `INFORMATION_SCHEMA` changes. (Bug #25337054, Bug #84384)

- A new spatial function, `ST_SwapXY()`, accepts a geometry argument and swaps the X and Y values of each coordinate pair within the geometry. For more information, see [Geometry Format Conversion Functions](#).

SQL Syntax Notes

- Derived table syntax now permits a list of explicit column names following the table name, to override the default column names taken from the select list. For example, the column names for this derived table come from its select list:

```
mysql> SELECT * FROM (SELECT 1, 2, 3, 4) AS dt;
+----+-----+-----+
| 1 | 2 | 3 | 4 |
+----+-----+-----+
| 1 | 2 | 3 | 4 |
+----+-----+-----+
```

To provide column names, follow the derived table name with a parenthesized list of column names:

```
mysql> SELECT * FROM (SELECT 1, 2, 3, 4) AS dt (a, b, c, d);
+----+-----+-----+
| a | b | c | d |
+----+-----+-----+
| 1 | 2 | 3 | 4 |
+----+-----+-----+
```

Test Suite Notes

- `mysql-test-run.pl` now has a `--charset-for-testdb=charset_name` option for specifying the default character set for the `test` database. The default value is `latin1`. (Bug #25494068, Bug #84806)

- `mysql-test-run.pl` now has an `--only-big-tests` option, which causes only big tests to run. Normal (non-big) tests are skipped. If both `--big-test` and `--only-big-tests` are given, `--only-big-tests` is ignored. (Bug #25182306)
- `mysql-test-run.pl` has a new `--discover` option. When given, `mysql-test-run.pl` attempts to preload `discover`, the Developer Studio Memory Error Discovery Tool when starting `mysqld`. Reports from `discover` may be found in `log/mysqld.%p.txt` under the directory given by `--vardir`. This option is supported only on SPARC-M7 systems. (Bug #25048971)
- `mysqltest` now supports an `expr` command that enables an expression to be evaluated and assigned to a variable. For details, see the command description in http://dev.mysql.com/doc/dev/mysql-server/PAGE_MYSQL_TEST_COMMANDS.html (Bug #24806741)
- `mysqltest` now supports a `force-cpdir` command that copies a source directory to a destination directory recursively (that is, it copies subdirectories). (Bug #24806681)
- `mysql-test-run.pl` has a new `--summary-report` option. When given, `mysql-test-run.pl` generates a plain text version of the test summary only and writes it to the file named as the option argument. The file is suitable for sending by email. (Bug #24512357, Bug #82708)
- `mysql-test-run.pl` could not be run with `--valgrind-option=--tool=custom_tool`, for values of `custom_tool` such as `massif` or `helgrind`, because it added the options for `memcheck` that might not be understood by other tools. Also, the `mysql-test-run.pl --callgrind` option did not work because it supplied an invalid `--base` option to `callgrind`. Thanks to Daniel Black for the patch on which the fixes were based. (Bug #23713613, Bug #82039)
- The `mysqltest connect()` function now permits `SOCKET` and `TCP` values for the `options` argument, to specify using the socket-file and TCP/IP connection protocols.

In addition, connection protocols now must match the current system. Previously, passing `PIPE` or `SHM` on non-Windows systems caused the default (socket-file) connection protocol to be used. Now this causes an error, and, similarly, passing `SOCKET` on Windows systems causes an error. (Bug #21046241)

- `mysql-test-run.pl` now supports a `--bootstrap` option that enables `mysqld` options to be designated as bootstrap options. When `mysql-test-run.pl` finds `--bootstrap` options in a `master.opt` file, it removes and reinitializes the data directory and restarts the server with the new option settings. (Bug #18184868)

X Plugin Notes

- The `MYSQLX_TCP_PORT` CMake option was ignored. (Bug #25493867, Bug #84804)
- The protobuf compiler emitted “unused import” warnings for `mysqlx.proto`. (Bug #25444009, Bug #84641)
- Connections were released which were not initialized when validation occurred. (Bug #25392280)
- The `Mysqlx_notice_warning_sent` and `Mysqlx_notice_other_sent` status variable values did not increment as expected. (Bug #25289949, Bug #84258, Bug #25290001, Bug #84260)
- The manual definition of the `MYSQL_DYNAMIC_PLUGIN` macro was redundant. (Bug #25162590, Bug #83988)
- IPv6 connectivity support defined a new system variable `Mysqlx-bind-address` with the default value of '0.0.0.0', but the default value should have been '*' so that connections to X Plugin may be made using an IPv6 address, an IPv4 address, or a hostname. (Bug #25047909, Bug #83688)
- The `stmt` field was marked as `required` in the message:

```

message Mysqlx.Crud.ModifyView {
    required Collection collection = 1;

    optional string definer = 2;
    optional ViewAlgorithm algorithm = 3;
    optional ViewSqlSecurity security = 4;
    optional ViewCheckOption check = 5;

    repeated string column = 6;
    required Mysqlx.Crud.Find stmt = 7;
}

```

(Bug #24968735, Bug #83595)

- For a table created with `CREATE TABLE foo (doc JSON)`, the statement `list_objects` incorrectly reported the table as a collection. (Bug #24963952)
- Sending `Expect.Open` to a connected socket before authentication led to the following error messages:

```

... Plugin mysqlx reported: '10: Invalid message 24 received during client initialization'
... Plugin mysqlx reported: '10: ERROR reading from socket Bad file descriptor (9) 2'

```

(Bug #24940144, Bug #83494)

- On Linux, MySQL Shell in JavaScript and Python interactive mode hid collections if the letter case in the collection name was not uniform. (Bug #24848125)
- If multiple user accounts exist that match the authenticating user, the X Plugin would attempt to authenticate all matching accounts instead of the best matching account. (Bug #24847537)
- Boost classes with corresponding functionality in C++ 11 were moved or wrapped. (Bug #24680856)
- When two X Plugin instances were started on the same port, the resulting error message was split over two lines and contained duplicate information. (Bug #24679018, Bug #83008)
- The `performance_schema.threads` table did not show `processlist` statistics for the X Plugin. (Bug #24638038)
- If the client sent an unknown message type to the server, the server would send back an error message and then disconnect the client. (Bug #24611754, Bug #82868)
- When the client sent a zero length message, the X Plugin did not send a response. For a subsequent message the client received an error `E_X_BAD_MESSAGE` and was disconnected, depending on the header content. (Bug #24595459, Bug #82862)
- X Plugin displayed its version to any connected user, including those not authenticated. (Bug #24562707, Bug #82784)
- X Plugin generated an incorrect query when a duplicate field name existed in an `Update` statement. (Bug #24510083)
- The statement `list_objects` returned incorrect information for some views in `sys` database. (Bug #24472325)
- X Plugin was trying to send result sets to a client, even where the connection had been closed. The plugin now stops execution of a stored procedure if the connection is reset. (Bug #24440344)
- Attempting to list objects using the `list_objects` statement without explicitly setting the database name resulted in the collection being incorrectly classified as a table. (Bug #23205895, Bug #81248)

- X Plugin now supports the `sha256_password` plugin.

Platform-Specific Notes

- **Solaris:** The minimum required version of Solaris is now Solaris 11 update 3, due to a dependency on system runtime libraries.
- **Solaris:** On Solaris, MySQL is now built with Developer Studio 12.5 instead of `gcc`. The binaries require the Developer Studio C/C++ runtime libraries to be installed. See here for how to install only the libraries:

https://docs.oracle.com/cd/E60778_01/html/E60743/gozsu.html

Functionality Added or Changed

- **Incompatible Change:** Previously, for `INSERT ... SELECT ... ON DUPLICATE KEY UPDATE` statements for which the `UPDATE` referred to `SELECT` columns, the results could be incorrect if the `SELECT` was a `UNION`. Now such statements are prohibited and produce an error. To work around this restriction, write the `UNION` as a derived table so that its rows can be treated as a single-table result set, which is permitted. See [INSERT ... ON DUPLICATE KEY UPDATE Syntax](#).
- **InnoDB:** By default, `InnoDB` reads uncommitted data when calculating statistics. In the case of an uncommitted transaction that deletes rows from a table, `InnoDB` excludes records that are delete-marked when calculating row estimates and index statistics, which can lead to non-optimal execution plans for other transactions that are operating on the table concurrently using a transaction isolation level other than `READ UNCOMMITTED`. To avoid this scenario, a new configuration option, `innodb_stats_include_delete_marked`, can be enabled to ensure that `InnoDB` includes delete-marked records when calculating persistent optimizer statistics. (Bug #23333990)
- **InnoDB:** Geometry parsing and bounding box computational code for R-trees was moved from `InnoDB` to the server.
- **InnoDB:** `InnoDB` now supports `NOWAIT` and `SKIP LOCKED` options with `SELECT ... FOR SHARE` and `SELECT ... FOR UPDATE` locking read statements. `NOWAIT` causes the statement to return immediately if a requested row is locked by another transaction. `SKIP LOCKED` removes locked rows from the result set. See [Locking Read Concurrency with NOWAIT and SKIP LOCKED](#).

`SELECT ... FOR SHARE` replaces `SELECT ... LOCK IN SHARE MODE`, but `LOCK IN SHARE MODE` remains available for backward compatibility. The statements are equivalent. However, `FOR SHARE` supports `NOWAIT`, `SKIP LOCKED`, and `OF tbl_name` options. See [SELECT Syntax](#).

`OF tbl_name` applies locking queries to named tables.



Note

`OF` now is a reserved word and cannot be used as an identifier without identifier quoting.

- **InnoDB:** The `InnoDB` tablespace encryption feature now supports encryption of redo log and undo log data, controlled by the `innodb_redo_log_encrypt` and `innodb_undo_log_encrypt` configuration options. See [Redo Log Data Encryption](#), and [Undo Log Data Encryption](#).
- **InnoDB:** `InnoDB` internal temporary tables that are stored on disk now support multiple cursor positions, permitting single writer and multiple reader access within the same thread. The purpose of this enhancement is to provide support for recursive and non-recursive common table expressions (CTEs).

Additionally, consistent-read access to [InnoDB](#) on-disk internal temporary tables is replaced by a dirty-read scheme, and row counts for [InnoDB](#) on-disk internal temporary tables now use row count statistics instead of slower table scans.

- **InnoDB:** [InnoDB](#) now compresses large objects into a sequence of smaller [zlib](#) streams for tables that use `ROW_FORMAT=COMPRESSED`. Previously, large object data was compressed into a single [zlib](#) stream.
- **Replication:** View change events from a Group Replication group can now be replicated to an external multithreaded slave (MTS) of type `DATABASE`. (Bug #25170698, Bug #84008)
- **Replication:** When a negative or fractional timeout parameter was supplied to `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`, the server behaved in unexpected ways. With this fix:
 - A fractional timeout value is read as-is, with no round-off.
 - A negative timeout value is rejected with an error if the server is on a strict SQL mode; if the server is not on a strict SQL mode, the value makes the function return NULL immediately without any waiting and then issue a warning.

(Bug #24976304, Bug #83537)

- **Replication:** Added the `binlog_expire_logs_seconds` system variable, which sets an interval in seconds for purging of the binary log. The effects of this variable and `expire_logs_days` are cumulative, making it possible to set a period such as 1.5 days. To completely disable automatic binary log purging, set both variables equal to 0, which is the default value for both of them. (Bug #71697, Bug #18260088)
- **Replication:** Performance schema tables have been added to monitor replication lags and queues. The `replication_connection_status` table has updated information on the last transaction queued in the relay log, as well as the transaction currently being queued in the relay log. The `replication_applier_status_by_coordinator` table has updated information on the last transaction written to the buffer of a worker, as well as the transaction currently being processed by the coordinator. The `replication_applier_status_by_worker` table has updated information on the last transaction applied by the worker, as well as on the transaction currently being applied. The information presented in each of these tables contains the transaction's GTID, commit timestamps, processing stage timestamp, and the timestamp of the completion.

In the `replication_applier_status_by_worker` table, the `LAST_SEEN_TRANSACTION` column is replaced by the `APPLYING_TRANSACTION` column.

- **Replication:** The new `binlog_expire_logs_seconds` variable is a global server variable in addition to the existing `expire_logs_days` variable. The purpose is to facilitate finer grained retention policies of binary logs than the current day granularity.
- **Replication:** Work has been done to reduce contention between receiver (I/O) and applier (SQL) replication threads. This ensures slaves keep up with a higher insert load on their master.
- **Replication:** The `binlog-transaction-dependency-tracking` variable has been added to improve writeset-based multi-threaded slave dependency tracking on masters. This enables you to choose whether to put information in the binary log that allows the slave to parallelize based on commit timestamps or transaction write sets. Parallelizing on write sets has potentially much more parallelism since it does not depend on the commit history, and as such, applying binary logs on a slave may explore better the underlying computing hardware (cpu cores) and ultimately this means that replication can become a lot faster. Our testing shows that we can apply binlog on the slave about 4 times faster, even for a single-threaded workload, which is the most difficult workloads for multi-threaded slaves.

- **Replication:** Correct recovery of DDL statements or transactions by binary log implements support for correct recovery of DDL transactions based on the binary log data. In the context of the new data-dictionary it becomes possible to make metadata changes and write them to the binary log as a single transaction by using the 2-phase commit protocol.
- **Replication:** Multi-source replication now supports per-channel replication filters. This makes it possible to filter out the execution of selected replicated data from a specific channel. Until now replication filters have been global, and only applicable to all channels.
- **Replication:** Row-based replication now uses extended table metadata in the binary log. The extended metadata serves two major purposes: slaves use the metadata to transfer data smoothly when its table structure is different from master's, and external software can use the metadata to decode row events and store the data into external databases.
- **Replication:** Group Replication now supports `SAVEPOINT` SQL transactions.
- **Replication:** An infrastructure for GTID based delayed replication and replication lag monitoring has been added to enable you to properly monitor replication lag. Two new timestamps that are associated with each transaction (not each event or statement) in the binary log have been introduced. The `original_commit_timestamp` is in microseconds since the epoch when the transaction was committed on the original master, and the `immediate_commit_timestamp` is in microseconds since the epoch when the transaction was committed on the immediate master.
- **Replication:** The `RESET MASTER` statement has been extended to allow specification of a binary log file number. The `RESET MASTER TO` statement accepts an argument which specifies the index of the new binary log file to use. The purpose is to simplify failover procedures by replacing the `FLUSH BINARY LOGS` and `PURGE BINARY LOGS TO` statements with a single `RESET MASTER` statement.
- **Replication:** The process used for how delay is defined and calculated for delayed replication have been improved. This relies on a new timestamp in the binary log, the `immediate_commit_timestamp`, which is the number of microseconds since epoch when the transaction was written to the binary log of the immediate master. This means that the delay is no longer computed per-event, it is now applied per-transaction.
- **Replication:** It is now possible to specify whether information written into the binary log enables replication slaves to parallelize based on commit timestamps, or on transaction write sets.

Using write sets has a the potential for greater parallelism than using commit timestamps since it does not depend on the commit history. When applying binary logs in this fashion on a replication slave, it may be able to leverage capabilities of the underlying computing hardware (such as CPU cores) and thus speed up this process.

The interface for choosing the source of parallelization is implemented as a new server system variable `binlog_transaction_dependency_tracking` which can take any one of the values `COMMIT_ORDER`, `WRITESET`, or `WRITESET_SESSION`. `COMMIT_ORDER` (the default) causes parallelization information to be logged using commit timestamps; `WRITESET` causes this information to be logged using write sets in such a way that any transactions not updating the same row can be parallelized; and `WRITESET_SESSION` acts in the same fashion as `WRITESET`, except that updates originating with the same session cannot be reordered. The size of the row hash history that is kept in memory for tracking transaction dependencies can be set using `binlog_transaction_dependency_history_size`, also introduced in this release.

- **JSON:** Added the JSON utility function `JSON_PRETTY()`, which prints an existing `JSON` value, or any string that can successfully be parsed as a JSON document, in a format that can be easily read by humans. Each JSON object member or array value is displayed on a separate line of the output; each child object or array is intended 2 spaces with respect to its parent.

Examples:

```
mysql> SELECT JSON_PRETTY('123');
+-----+
| JSON_PRETTY('123') |
+-----+
| 123                |
+-----+

mysql> SELECT JSON_PRETTY("[1,3,5]");
+-----+
| JSON_PRETTY("[1,3,5]") |
+-----+
| [
| 1,
| 3,
| 5
| ]                |
+-----+

mysql> SELECT JSON_PRETTY('{ "a": "10", "b": "15", "x": "25" }');
+-----+
| JSON_PRETTY('{ "a": "10", "b": "15", "x": "25" }') |
+-----+
| {
|   "a": "10",
|   "b": "15",
|   "x": "25"
| }                |
+-----+
```

- **JSON:** Previously, when sorting JSON values in a query using `ORDER BY`, each value was represented by a part having a fixed length in the sort key, each such part requiring 1K of memory. In many cases this usage was excessive—for example, an integer scalar value requires only a relatively very few bytes, so that the remainder of the 1K (90% or more) was taken up by padding.

Variable length parts for JSON sort keys have been introduced to address this issue, with each key part now taking up only the space that is needed to store the value. This has the following benefits for performance:

- Because sort buffer space is used more effectively, filesorts need not flush to disk as early or often, which means that more data can be sorted in memory.
- Shorter keys can be compared more quickly than longer ones, providing a noticeable improvement in performance for in-memory sorts as well as sorts requiring disk usage.
- The `WEIGHT_STRING()` debugging function no longer supports the `LEVEL` clause. (Bug #25469683, Bug #84723)
- For Developer Studio 12.5, 32-bit builds are now disabled unless the `FORCE_UNSUPPORTED_COMPILER` option is given. (Bug #25267157, Bug #84230)
- Several foreign key constraint checks that were storage engine agnostic were being done inside `InnoDB`. These are now done at the SQL layer. (Bug #25252847)
- Some GIS out-of-bounds checking was simplified. Thanks to Daniel Black for the contribution. (Bug #25202470, Bug #84062)
- The set of compiler flags used for Developer Studio 12.5 was improved. (Bug #25148549, Bug #83955)

- `CMake` now uses `CMAKE_POLICY` of `CMP0022 NEW` rather than `CMP0022 OLD`. (Bug #25090147, Bug #83804)
- If MySQL was configured to build with the `-Wno-error` option, `mysql_config` produced incorrect output for its `--cflags` option.

The set of compiler options that `mysql_config` and `pkg-config` produce now is determined by whitelisting rather than blacklisting. (Bug #25040566, Bug #22898475, Bug #80662)

- The performance of UTF-8 binary collations was improved. (Bug #24788778, Bug #83247, Bug #25076862)
- The systemd service file for `mysqld` now includes a `Documentation` value in the `[Unit]` section to provide a link to the systemd documentation in the MySQL Reference Manual. (Bug #24735762)
- Unit testing now uses Google Mock 1.8. (Bug #24572381, Bug #82823)
- The unimplemented and nonstandard `WITH CUBE` clause for `GROUP BY` is no longer supported.



Note

`CUBE` now is a reserved word and cannot be used as an identifier without identifier quoting.

(Bug #24572048, Bug #82821)

- These outdated build scripts were removed from MySQL source distributions: `BUILD/SETUP.sh`, `BUILD/autorun.sh`, `BUILD/cmake_configure.sh`, `BUILD/compile-dist`, and `cmake/configure.pl`. (Bug #24512933)
- Consistency and maintainability of Debian/Ubuntu packaging maintainer scripts was improved. (Bug #23588977)
- In-place `ALTER TABLE` operations are now possible for column modifications of tables containing generated columns that depend on columns with a `DEFAULT` value if the modified columns are not involved in the generated column expressions. For example, changing the `NULL` property of a separate column can be done in place without a table rebuild. (Bug #22987899, Bug #80832)
- For dumps of the `mysql` system database, `mysqldump` no longer generates DDL statements for the `innodb_index_stats` and `innodb_table_stats` tables. Such statements fail at dump reload time because those tables are not user accessible. (Bug #22655287)
- If `mysqld` is invoked with `--daemonize`, `stdout` and `stderr` are redirected to `/dev/null` if connected to a terminal type device, so that `mysqld` can behave as a true daemon. (Bug #21627629)
- `mysql_secure_installation` is more strict about what it considers valid yes and no responses. (Bug #13344753, Bug #62925)
- MySQL client and utility programs reported their own program-specific version number when invoked with the `--version` option, and the version strings did not follow a consistent format. Now these programs print the MySQL version number (same as `mysqld`) in a consistent format. (Bug #11763201, Bug #55885)
- A new `mysqldump` option, `--network-timeout`, enables large tables to be dumped by setting `max_allowed_packet` to its maximum value and network read and write timeouts to a large value. This option is enabled by default. To disable it, use `--skip-network-timeout`. (Bug #11754493, Bug #46103)

- Added two JSON aggregation functions `JSON_ARRAYAGG()` and `JSON_OBJECTAGG()`. The `JSON_ARRAYAGG()` function takes a column or column expression as an argument, and aggregates the result set as a single JSON array, as shown here:

```
mysql> SELECT col FROM t1;
+-----+
| col                                     |
+-----+
| {"key1": "value1", "key2": "value2"} |
| {"keyA": "valueA", "keyB": "valueB"} |
+-----+
2 rows in set (0.00 sec)

mysql> SELECT JSON_ARRAYAGG(col) FROM t1;
+-----+
| JSON_ARRAYAGG(col)                     |
+-----+
| [{"key1": "value1", "key2": "value2"}, {"keyA": "valueA", "keyB": "valueB"}] |
+-----+
1 row in set (0.00 sec)
```

The order of the array elements is unspecified.

`JSON_OBJECTAGG()` takes two columns or expressions which it interprets as a key and a value, respectively; it returns the result as a single JSON object, as shown here:

```
mysql> SELECT id, col FROM t1;
+----+-----+
| id | col                                     |
+----+-----+
|  1 | {"key1": "value1", "key2": "value2"} |
|  2 | {"keyA": "valueA", "keyB": "valueB"} |
+----+-----+
2 rows in set (0.00 sec)

mysql> SELECT JSON_OBJECTAGG(id, col) FROM t1;
+-----+
| JSON_OBJECTAGG(id, col)                |
+-----+
| {"1": {"key1": "value1", "key2": "value2"}, "2": {"keyA": "valueA", "keyB": "valueB"}} |
+-----+
1 row in set (0.00 sec)
```

A `NULL` key causes an error; duplicate keys are ignored.

For more information, see [Aggregate \(GROUP BY\) Functions](#). (Bug #78117, Bug #21647417)

- MySQL now supports an `IMPORT TABLE` statement that imports `MyISAM` tables based on information contained in `.SDI` (Serialized Dictionary Information) metadata files. This statement is useful for exporting “raw” table files from one server and importing them into another, and provides a faster alternative to dumping tables as a file of SQL statements using `mysqldump` and processing the dump file using `mysql`. For more information, see [IMPORT TABLE Syntax](#).
- The deprecated `replace` utility has been removed. If you wish to continue using this utility, be sure to retain a copy from an installed version of MySQL before upgrading to the current version.
- The removal of the `mysql_shutdown()` C API function in MySQL 8.0.0 has been reverted. It is still to be removed in a future MySQL release, but that removal has been deferred.

Bugs Fixed

- **Incompatible Change:** The stored definition of a view for which an explicit column name list was provided could be invalid. For example, for this statement:

```
CREATE VIEW v1 (name2) AS SELECT 1 AS name1 UNION SELECT 2 ORDER BY name1;
```

The stored definition looked like this, which is invalid:

```
CREATE VIEW v1 AS SELECT 1 AS name2 UNION SELECT 2 AS 2 ORDER BY name1;
```

Now column names within the stored definition are not replaced by the column name list:

```
CREATE VIEW v1 (name2) AS SELECT 1 AS name1 UNION SELECT 2 AS 2 ORDER BY name1;
```

An incompatibility resulting from this change is that the `CREATE VIEW` statement can no longer be recreated solely from the `INFORMATION_SCHEMA VIEWS` table, because the `VIEW_DEFINITION` value does not show the column name list. Instead, you must also consult the `COLUMNS` table to get the column name list. Alternatively (and more simply), use `SHOW CREATE VIEW`. (Bug #23265335, Bug #81377)

- **InnoDB; Microsoft Windows:** On a MySQL 64-bit build on Windows, a file I/O retry result was misinterpreted due to a missing cast necessary for the correct operation of the retry path, resulting in a failing assertion and operating system error. (Bug #24711351)
- **InnoDB:** In debug builds, a call that initialized latch debugging raised an assertion due to a race condition. (Bug #25700405)
- **InnoDB:** The restriction that required the first undo tablespace to use space ID 1 was removed. The first undo tablespace may now be assigned a space ID other than 1. Space ID values for undo tablespaces are still assigned in a consecutive sequence. (Bug #25551311)
- **InnoDB:** A `DROP TABLE` operation raised an assertion on a server with an `innodb_force_recovery` setting of 5 or 6. `DROP TABLE` is no longer permitted with these `innodb_force_recovery` settings. (Bug #25385590)

References: This issue is a regression of: Bug #19779113.

- **InnoDB:** Compiling the server without the Performance Schema caused a build failure. (Bug #25348787)
- **InnoDB:** During read-ahead, the wrong page size was used to calculate the tablespace size. (Bug #25330449)
- **InnoDB:** Compiling on Fedora 25 using `DWITH_LZ4=system` resulted in a build failure due to a deprecated `LZ4_COMPRESS_LIMITEDOUTPUT` function. (Bug #25297593)
- **InnoDB:** Disabling macros such as `UNIV_PFS_MUTEX`, `UNIV_PFS_RWLOCK`, and `UNIV_PFS_THREAD` caused compilation errors. (Bug #25251082)
- **InnoDB:** A NULL virtual column field name in a virtual index caused a server exit during a field name comparison that occurs while populating virtual columns affected by a foreign key constraint. (Bug #25222337)
- **InnoDB:** The file handle type name for InnoDB file I/O Performance Schema instrumentation was changed from `os_pfs_file_t` to `pfs_os_file_t`. (Bug #25220118)
- **InnoDB:** Transactions with subqueries on `INFORMATION_SCHEMA` tables could block concurrent DDL operations. (Bug #25200952)

- **InnoDB:** During a range comparison, a secondary index field number was passed instead of clustered index field number, eventually causing the retrieval of an incorrect field. (Bug #25175249)
- **InnoDB:** A server exit on restart was caused by missing `my_thread_init()` and `my_thread_exit()` functions for background threads that initialize the `st_my_thread_var` structure. (Bug #25167032)
- **InnoDB:** A `memcached` read operation with a non-default read batch size configuration resulted in a server exit. (Bug #25147515)
- **InnoDB:** The `INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS` table reported `NULL` for a foreign key constraint name (`UNIQUE_CONSTRAINT_NAME`) after restarting the server. (Bug #25126722)
- **InnoDB:** A gap lock was taken unnecessarily during foreign key validation while using the `READ COMMITTED` isolation level. (Bug #25082593)
- **InnoDB:** Loading `InnoDB` tables required more memory in MySQL 5.7 due primarily to the addition of in-memory structure members introduced with temporary table optimizations. The in-memory structure members, only used for temporary tables, are now only allocated as needed. (Bug #25080442)
- **InnoDB:** After a `TRUNCATE TABLE` operation on a table with a `FULLTEXT` index, space size was incorrectly calculated resulting in an invalid read. (Bug #25053705)
- **InnoDB:** A prepared XA transaction was rolled back by a high priority transaction. The high priority transaction should wait if the blocking transaction is in a prepared state. (Bug #25032066)
- **InnoDB:** `InnoDB` passed an invalid argument to `syscall(SYS_futex)`. (Bug #24923840, Bug #83375)
- **InnoDB:** After redo log recovery, the node size of an undo tablespace object could sporadically be zero, eventually causing an error. Additionally, the undo tablespace object was incorrectly placed on the LRU list, and the header page of the undo tablespace could be present the buffer cache when the tablespace object is freed, causing a file-open failure on the first page read. (Bug #24916359)
- **InnoDB:** During a checkpoint, all `MLOG_FILE_NAME` redo log records were written in a single mini-transaction (mtr), causing a log parsing buffer overflow. (Bug #24793413, Bug #83245)
- **InnoDB:** The GCC `mach_parse_compressed` function should load one to five bytes depending on the value of the first byte. Due to a GCC bug, GCC 5 and 6 emit code to load four bytes before the first byte value is checked (GCC Bug #77673). A workaround prevents this behavior. Thanks to Laurynas Biveinis for the patch. (Bug #24707869, Bug #83073)
- **InnoDB:** Code related to tablespace type functions was improved to address issues with naming consistency, function distribution, and function usage. (Bug #24706739, Bug #83092)
- **InnoDB:** A mechanism was added to debug builds to ensure that keys for `InnoDB` Performance Schema instrumentation are registered with Performance Schema. The mechanism causes startup to fail on debug builds if the number of Performance Schema keys does not match the number of registered Performance Schema keys. (Bug #24686908)
- **InnoDB:** A race condition while updating table statistics could result in an estimated row count of 1 and an incorrect query execution plan. (Bug #24666839, Bug #82968)
- **InnoDB:** Due to a `glibc` bug, short-lived detached threads could exit before the caller had returned from `pthread_create()`, causing a server exit. Thanks to Laurynas Biveinis for the patch. (Bug #24605956, Bug #82886)
- **InnoDB:** An error in code related to table statistics raised an assertion in the `dict0stats.cc` source file. (Bug #24585978)

- **InnoDB:** The list of module base names representing modules in the [InnoDB](#) code base that allocate memory using `ut_malloc` or `ut_new` was incomplete. The list is used by the Performance Schema for reporting of [InnoDB](#) memory allocation event data. (Bug #24571816)
- **InnoDB:** Some [InnoDB](#) rw-lock and mutex keys were not registered for use with the Performance Schema. (Bug #24571597)
- **InnoDB:** After increasing the value of `innodb_undo_logs` and restarting the server, the number of active undo tablespaces was not increased when assigning undo tablespaces to newly allocated rollback segments. (Bug #24488141)
- **InnoDB:** The unused `MLOG_UNDO_HDR_DISCARD` redo log record type and related functions were removed. (Bug #24482001)
- **InnoDB:** An assertion was raised when the purge thread started due to the server not recognizing a tablespace ID as an undo tablespace ID. (Bug #24479773)

References: This issue is a regression of: Bug #23517560.

- **InnoDB:** [InnoDB](#) incorrectly reported an error about missing encryption when restoring pages from the doublewrite buffer during recovery. (Bug #24471076)
- **InnoDB:** [InnoDB](#) reported an incorrect estimate for the number of pages in the buffer pool for a table. The number of pages was not decremented correctly when pages were removed. (Bug #24464147)

References: This issue is a regression of: Bug #21747906.

- **InnoDB:** The following code changes related to undo tablespace management were implemented:

- An in-memory array for temporary tablespace rollback segments was added, freeing 32 slots in the `TRX_SYS` page for use by redo-enabled rollback segments.

As a result of this change, you no longer need to consider temporary tablespace rollback segments when setting the `innodb_rollback_segments` configuration option during configuration of separate undo tablespaces.

- Code comments related to undo logs and rollback segments were revised.
- Fixed arrays that list rollback segments and undo tablespaces were replaced by dynamic size arrays.
- The function that initializes undo tablespaces was refactored.
- A class was added for managing just-in-time creation of undo and undo file names.

(Bug #24462978)

- **InnoDB:** A cached undo segment was not removed from the rollback segment history during a slow shutdown. (Bug #24450908)
- **InnoDB:** An error during a table-rebuilding operation on a table with only a generated clustered index (`GEN_CLUST_INDEX`) raised an assertion due to an error called with an invalid key name. (Bug #24444831)
- **InnoDB:** MySQL did not build with GCC 6.1.1. (Bug #24438752)
- **InnoDB:** Rotating the tablespace encryption master key while the server is in read-only mode raised an assertion instead of displaying an error message. (Bug #24404091)

- **InnoDB:** On a table without an explicitly defined primary key, **InnoDB** did not replace the implicit clustered index (`GEN_CLUST_INDEX`) when a unique key was defined on a `NOT NULL` column. (Bug #24397406)
- **InnoDB:** `handler::keys_to_use_for_scanning()` was removed from the handler API. The function was no longer used. The `HA_READ_ORDER` index flag provides the same information for each index. (Bug #24364448)
- **InnoDB:** Page cleaner threads asserted due to a regression related to the adaptive hash index feature. (Bug #24346574)

References: This issue is a regression of: Bug #21407023.

- **InnoDB:** **InnoDB** failed to free memory used by the full-text optimizer thread. (Bug #24331265)
- **InnoDB:** During recovery, **InnoDB** attempted to fetch LOB pages using a null reference. (Bug #23615208)
- **InnoDB:** When adding a new index, the server dropped an internally defined foreign key index and attempted to use a secondary index defined on a virtual generated column as the foreign key index, causing a server exit. **InnoDB** now permits a foreign key constraint to reference a secondary index defined on a virtual generated column. (Bug #23533396)
- **InnoDB:** An `INFORMATION_SCHEMA.FILES` query resulted in a server exit due to a race condition with a concurrent tablespace creation operation. (Bug #23477214)
- **InnoDB:** A concurrent DML operation during an in-place `ALTER TABLE` operation that rebuilt the table did not update a virtual index, resulting in a mismatch between the virtual index and clustered index. (Bug #23219499)
- **InnoDB:** A `TRUNCATE TABLE` operation held the `dict_sys` mutex while scanning for and removing pages from the buffer pool, causing concurrent DDL operations to stall. The mutex is now released during the scan and acquired again when the scan is completed. (Bug #23070734, Bug #80060)
- **InnoDB:** Inserting GIS data into an R-tree raised an assertion due to a missing page number field that was encountered when storing the B-tree cursor. (Bug #23044098, Bug #80939)
- **InnoDB:** Writing to an **InnoDB** internal temporary table did not increment the `Handler_write` counter. (Bug #23024178)
- **InnoDB:** Changes to the **InnoDB** recovery process in MySQL 5.7 could require up to three scans of the redo log during recovery. To reduce the number of scans, the first and second scans were merged. With this change, there is only one scan unless the redo log record hash table that is populated by the scan reaches its memory threshold. In this case, a second scan is initiated that performs a simultaneous scan and apply. (Bug #22963951, Bug #80788)
- **InnoDB:** A table-copying online `ALTER TABLE` operation on a `ROW_FORMAT=REDUNDANT` table with indexed virtual columns raised an assertion. (Bug #22018745)
- **InnoDB:** After a server restart, concurrent `INSERT` operations a table with an auto-increment primary key resulted in a duplicate entry error. The current auto-increment value was not changed after `auto_increment_increment` and `auto_increment_offset` settings were modified. (Bug #20989615, Bug #76872)
- **InnoDB:** Performance Schema instrumentation for **InnoDB** file I/O was disabled on Windows. (Bug #14025581)

- **InnoDB:** The `row_search_mvcc()` function unnecessarily traversed the entire table for a range query, which occurred when the record was not in the transaction read view. (Bug #84202, Bug #23481444, Bug #25251375)
- **Packaging:** The `my_create_minidump` function in the `mysys` library called the Windows API function `MiniDumpWriteDump` with an incorrect value (NULL) for the `ExceptionParam` parameter. This resulted in exception information being omitted from the minidump file. (Bug #24505650, Bug #82695)
- **Partitioning:** For a partitioned table, dropping a partition dropped triggers for the table as well. (Bug #24449174)
- **Partitioning:** Updating a row of a table that had partitioning on a generated column could raise an assertion failure for debug builds, and return incorrect results in nondebug builds. (Bug #22574695, Bug #80080)
- **Replication:** MySQL 8.0.1 adds the original commit timestamp related statements, but `mysqlbinlog` was wrongly reporting that these statements have been present since MySQL 8.0.0 because they were written to the log with the incorrect version number `80000`. The fix changes the version to `80001` in `mysqlbinlog` dumps. (Bug #25710507)
- **Replication:** `MEMBER_STATE` of a group replication member did not go from `ERROR` to `OFFLINE` when the `STOP GROUP_REPLICATION` command was executed if the error state was due to ER3092. (Bug #25674926)
- **Replication:** With flow control enabled, reaching a minimum flow control quota of 1 causes Group Replication to not stop throttling when the reason for throttling was no longer in effect. (Bug #25461354)
- **Replication:** Using an unresolvable host name in `group_replication_group_seeds` caused `START GROUP_REPLICATION` to fail. The fix ensures that host names in `group_replication_group_seeds` are validated when starting Group Replication and the list must contain at least one valid address. Invalid addresses are ignored. (Bug #25460324, Bug #84674)
- **Replication:** Lock contention impeded binary relay performance during processing of the relay log file on the slave. (Bug #25321231, Bug #77778)
- **Replication:** A failed node that was removed from the cluster could be expelled upon rejoining the cluster. (Bug #25311008)
- **Replication:** The `_gr_user` account created by Group Replication plugin installation was not reliably removed when the plugin was uninstalled. (Bug #25298987)
- **Replication:** When starting Group Replication on an offline node, the node could be configured for replication, but fail for recovery. (Bug #25256910)
- **Replication:** When using a multi-threaded slave, applier errors displayed worker ID data that was inconsistent with data externalized in Performance Schema replication tables. (Bug #25231367)
- **Replication:** Not all Group Replication GCS debug and trace messages were enabled in debug mode. (Bug #25209109, Bug #84079)
- **Replication:** Compiling MySQL 5.7.17 failed with a variable length array error. (Bug #25163241, Bug #83994)
- **Replication:** In row-based replication, a message that incorrectly displayed field lengths was returned when replicating from a table with a `utf8mb3` column to a table of the same definition where the column was defined with a `utf8mb4` character set. (Bug #25135304, Bug #83918)
- **Replication:** Group Replication GCS was not discarding messages when a member within the group was inactive. (Bug #25134074)

- **Replication:** Some unnecessary warnings were given when the Group Replication plugin was compiled on Windows platforms. (Bug #25119288)
 - **Replication:** If the binary log on a master server was rotated and a full disk condition occurred on the partition where the binary log file was being stored, the server could stop unexpectedly. The fix adds a check for the existence of the binary log when the dump thread switches to next binary log file. If the binary log is disabled, all binary logs up to the current active log are transmitted to slave and an error is returned to the receiver thread. (Bug #25076007)
 - **Replication:** An assertion could be raised if the Group Replication plugin attempted to contact the server when that was no longer possible. (Bug #25071492)
 - **Replication:** The GTID transaction skipping mechanism that silently skips a GTID transaction that was previously executed did not work properly for XA transactions. (Bug #25041920)
 - **Replication:** After executing restarts on the group replication applier SQL thread, the plugin could no longer detect failure of the thread. (Bug #24969065)
 - **Replication:** Building Group Replication on Windows requires a minimum CMAKE version of 2.8.12. (Bug #24964522)
 - **Replication:** If a relay log index file named relay log files that did not exist, `RESET SLAVE ALL` sometimes did not fully clean up properly. (Bug #24901077)
 - **Replication:** When the MTS slave applier stopped because of an (injected) error, it reported no useful information for troubleshooting. (Bug #24822686)
 - **Replication:** `FLUSH BINARY LOGS` could become slow with data replicated from many servers. (Bug #24806259, Bug #83270)
 - **Replication:** When using XA transactions, if a lock wait timeout or deadlock occurred for the applier (SQL) thread on a replication slave, the automatic retry did not work. The cause was that while the SQL thread would do a rollback, it would not roll the XA transaction back. This meant that when the transaction was retried, the first event was `XA START` which was invalid as the XA transaction was already in progress, leading to an `XAER_RMFAIL` error. (Bug #24764800)
- References: See also: Bug #83588, Bug #24923091, Bug #24966941.

- **Replication:** Enabling the group replication plugin caused the `performance_schema_max_mutex_classes` default value of 200 to be exceeded. As a result, some group replication mutex instruments did not appear in the `performance_schema.setup_instruments` table. (Bug #24746530)
- **Replication:** A partially failed `CREATE USER`, `RENAME USER`, or `ALTER USER` statement was not correctly consuming an auto-generated or specified GTID when binary logging was disabled. (Bug #24693798)
- **Replication:** `Binlog_sender`, which writes events from the binary log to a packet buffer and then sends the packet to the slave, did not reduce the size of the send buffer as expected. (Bug #24643036)
- **Replication:** The group commit update of GTIDs has been refactored to improve performance on workloads with many small transactions. (Bug #24398760)
- **Replication:** If the `relay_log` option was not specified in a configuration file, the `relay_log_basename` variable was being internally constructed on the fly using `hostname` but the `relay_log_basename` variable was not set. When a slave tried to access this uninitialized variable it resulted in an unexpected halt of the server. (Bug #24352667)

- **Replication:** For servers built with yaSSL, using group replication with encrypted connections could result in timeout failures waiting for view delivery. (Bug #23592214)
- **Replication:** When using a multi-threaded slave (`slave_parallel_workers` greater than 0) the value of `Seconds_Behind_Master` was incorrect when rotating a relay log. (Bug #23532304)
- **Replication:** An `XA PREPARE` statement that failed during the intermediate steps could lead to an inconsistent `XA` transaction state, where `ID = -1` but the `binlogged` flag was set to `true`. This caused asserts while executing `XA COMMIT` and `XA ROLLBACK` queries. (Bug #22915670)
- **Replication:** The server prevented several replication-related administrative statements from working if the `read_only` system variable was enabled. (Bug #22857926)
- **Replication:** `CHANGE MASTER TO` for a channel that did not exist could raise an assertion. (Bug #22255698)
- **Replication:** The delay specified by the `binlog_group_commit_sync_delay` system variable was applied to too many binary log commit groups. (Bug #21420180)
- **Replication:** The number of generated unwanted fseeks into the binary log file being replicated to a slave has been reduced. (Bug #83226, Bug #24763579)
- **Replication:** The fix for Bug #81657 was not correctly merged into MySQL 8.0. Thanks to Laurynas Biveinis for alerting us. (Bug #83124, Bug #24715790)
- **Replication:** The `rpl.rpl_binlog_errors` test was failing sporadically on Windows. (Bug #82302, Bug #24330138)
- **Replication:** When `binlog_group_commit_sync_delay` was set to a value between 1 and 9, if `binlog_group_commit_sync_no_delay_count` was set to a value greater than 1, and the number of transaction commits was less than `binlog_group_commit_sync_no_delay_count`, these commits hung forever if no more commits were received; and if `binlog_group_commit_sync_no_delay_count` was set to 0, all transaction commits hung forever. (Bug #80652, Bug #22891628)
- **Replication:** Concurrent `CREATE TRIGGER` and `DROP TRIGGER` statements were not being binary logged in the correct order, causing slaves to fail. (Bug #77095, Bug #21114768)
- **Microsoft Windows:** On Windows, `SHOW TABLES FROM db_name` hung if `db_name` was given in uppercase. (Bug #24800048, Bug #83262)
- **Microsoft Windows:** 32-bit builds are no longer supported on Windows, but `CMake` failed to detect when a 32-bit build would be attempted, resulting in compilation errors later. Now `CMake` detects 32-bit build attempts and produces an appropriate error message. (Bug #24487483, Bug #82645)
- **Solaris:** When the `WITH_INNODB_MEMCACHED CMake` option is enabled, `memcached` now can be built on Solaris 11 and 12 using the Developer Studio 12.5 compiler. (Bug #24504155, Bug #82692)
- **Solaris:** Library search path handling on Solaris was incorrect. (Bug #24487934, Bug #82646)
- **JSON:** When a `JSON` value consisted of a large sub-document wrapped in many levels of `JSON` arrays, objects, or both, serialization of the `JSON` value sometimes required an excessive amount time to complete. (Bug #23031146)
- **JSON:** When a `NULL` value existed in a `JSON` column, the result from a query using `GROUP_CONCAT ()` together with the `ORDER BY` clause was not always correct. (Bug #22992666)
- **JSON:** The internal `rapid_json_handler` used its own data structures to represent a partially-built DOM; these had to be converted into a `Json_dom` graph before returning the result. Now this handler

builds the graph directly, which reduces the amount of work required to build it, and thus to parse a JSON document. (Bug #22900110)

- **JSON:** The internal function `Item_func_case::val_json()` did not always set the null value flag as expected when a `CASE` expression evaluated to `NULL`, leading to an assertion in debug builds of the server. (Bug #22887227)
- **JSON:** The `SUM()` function truncated decimal values extracted from JSON documents, producing an integer result. (Bug #84935, Bug #25530204)
- **JSON:** A JSON document that contained a double value slightly greater than the maximum value that can be represented by a double silently replaced it with zero instead of rejecting the value and raising an error. Such values are now handled correctly in MySQL JSON documents.

The underlying issue was traced to a problem with RapidJSON, which has been reported to that library's developers as [Issue #849](#). (Bug #84891, Bug #25518504)

- **JSON:** The `JSON_SEARCH()` and `JSON_CONTAINS_PATH()` functions did not work when the `one_or_all` argument was specified using UTF-16 encoding. For both of these functions, this argument is now converted to `utfmb4` if need be before its value is checked. (Bug #84880, Bug #22516960)
- **JSON:** The `JSON_UNQUOTE()` function did not work with strings that used UTF-16 encoding. Now these strings are converted to `utfmb4` internally before being processed. (Bug #84878, Bug #25516881)
- **JSON:** Updating the same `JSON` column in a single statement could cause incorrect values to be written into the table. This occurred when the second update overwrote the column value with a subset of itself. An example of such a statement is shown here:

```
UPDATE t SET col = JSON_ARRAY(value), col = col->'${0}';
```

(Bug #84694, Bug #25461627)

- **JSON:** The functions `JSON_QUOTE()` and `JSON_UNQUOTE()` did not work correctly with multibyte character sets such as `utf8mb4`. (Bug #84680, Bug #25455065)

References: See also: Bug #77234, Bug #21193273.

- **JSON:** The error message for Error 3152 `ER_JSON_USED_AS_KEY` has been changed from `JSON column '%s' cannot be used in key specification` to the less confusing and more accurate `JSON column '%s' supports indexing only via generated columns on a specified JSON path`. (Bug #81364, Bug #23274244)
- **JSON:** Internal tests for MySQL JSON functionality ran out of stack space on some platforms when run against a debug-enabled server. Because timely checks were not made for stack usage, the server did not detect this situation, leading to a server exit.

The fix for this issue is twofold:

- Stack overrun checks are now made before attempting to serialize a nested array or object, so that the operation fails gracefully when processing deeply nested JSON documents, rather than causing an exit.
- Serialization of JSON documents has been reorganized so that it requires less use of the stack when compiled without optimization.

(Bug #81083, Bug #23106330, Bug #26399306)

- MySQL did not compile with GCC 7. (Bug #25643811, Bug #26825211)
- The (undocumented) `WINDOWS_RUNTIME_MD` CMake option has been removed. (Bug #25611359)
- If `--skip-innodb` or one of its variants was used, a spurious warning about `avoid_temporal_upgrade` was generated. (Bug #25573578)
- `mysqld_safe` failed to restart the server if a `PID_FILE.shutdown` file was present. (Bug #25572504)
References: This issue is a regression of: Bug #11751149.
- For Debian/Ubuntu packages, user-defined collation files could be overwritten during MySQL upgrades. Charset files are now marked as conffiles so that user customizations generate a prompt during upgrades whether to overwrite them. (Bug #25525628, Bug #84761)
- For `CREATE TABLE` statements that specified the table name with a database qualifier and included a `DATA DIRECTORY` or `INDEX DIRECTORY` option, an error occurred if there was no default database. (Bug #25514146, Bug #84861)
- `referenced_table_schema` and `referenced_table_name` field values in the `mysql.foreign_keys` data dictionary table were not stored in lowercase when `lower_case_table_names` was enabled. (Bug #25495714)
- Starting the server with `performance_schema_digests_size=1` caused an abnormal exit. (Bug #25492129, Bug #84786)
- For clients linked against yaSSL, connections became invalid when a read timeout occurred, rather than retrying the read. (Bug #25444075)
- MySQL compilation in different directories produced different builds to leakage of absolute paths into debug information and `__FILE__`. (Bug #25436469, Bug #84608)
- Calculations for UCA 9.0.0 collations were inefficient for tailoring rules containing contraction characters. (Bug #25426632, Bug #84577, Bug #25426632, Bug #84577)
- A negative internal connection timeout value caused the connection to go idle and abort. A negative value now causes the connection to block indefinitely in the absence of I/O. (Bug #25408557)
- A mutex Performance Schema name was too long and produced a warning at server startup. (Bug #25406915)
- MySQL failed to compile on some platforms with `-DWITH_LIBWRAP=ON`. CMake support now checks whether `tcpd.h` has proper function prototypes. (Bug #25395543, Bug #84495)
- An index defined on a data dictionary table column exceeded the maximum index key length when the instance was initiated with `innodb_page_size=4k`. (Bug #25384527)
- `mysqld_safe` did not check whether the directory named by the `--basedir` option existed. (Bug #25365194)
- Configuring CMake with `-G ninja` resulted in build output that was inappropriate for build platforms other than Xcode or Visual Studio. (Bug #25358460)
- `mysqld_safe` failed if the error log file named by the `--log-error` option was a FIFO. (Bug #25356221, Bug #84427)
- For prepared statements, an alias within a subquery or derived table might cause incorrect behavior during statement execution if another alias depended on it. (Bug #25343335, Bug #84398, Bug #25171608)

- `mysqld_safe` could fail if the `--datadir` option value ended with a `/` character. (Bug #25319457)
- A recent change to `mysqld_safe` caused the `mysql.server` script to be unable to start it if the base directory was specified as an absolute path that differed from the compiled-in default absolute path. (Bug #25319392, Bug #84263)
- The `CONNECTION_CONTROL` plugin failed to compile if the Performance Schema was disabled. (Bug #25308357, Bug #84304)
- Passwords did not expire correctly for accounts created using MySQL Workbench. (Bug #25299309)
- For System V init scripts for RPMs, the `[mysqld]` option-file section was being ignored for some options, such as `pid-file`. (Bug #25287707, Bug #84172)
- Init scripts failed to launch `mysqld_safe` if a non-default base directory was used. (Bug #25261472, Bug #84219)
- `CMake` now detects whether a GCC 5.3.0 loop optimization bug occurs and attempts a workaround if so. (Bug #25253540)
- `mysqld_safe --no-defaults` did not work (inadvertent consequence of an earlier bug fix). (Bug #25244898, Bug #84173)
- Semicolon (`;`) characters within or between statements could cause distinct digests to be generated from identical statements. (Bug #25244533, Bug #83253)
- Components could not register services without referring to their private implementation. A `SERVICE_IMPLEMENTATION(component, service)` macro now enables this to be avoided. (Bug #25238906)
- Certain stored functions, if used in a query `WHERE` clause, could be handled using Index Condition Pushdown (which should not happen), resulting in a server exit. (Bug #25196653, Bug #25174454)
- For `ai_ci` collations based on Unicode Collation Algorithm 9.0.0, accented characters that compare equal were treated as different by `LIKE` comparisons. (Bug #25167284, Bug #83999)
- For a client linked against `libmysqlclient`, invalid memory access could occur during use of prepared statements. (Bug #25164932)
- Some Linux startup scripts did not process the `datadir` setting correctly. (Bug #25159791)
- If a character set is specified for a column of a partitioned table, a segmentation fault could occur while upgrading. (Bug #25153261)
- `LOAD DATA` failed to accept multibyte characters that followed an escape sequence. (Bug #25147988, Bug #83950, Bug #25865525)
- The fix for Bug #25088048 caused the command used by `mysqld_safe` to start the MySQL server to no longer include the `mysqld` path. (Bug #25144379)

References: This issue is a regression of: Bug #25088048.

- For UCA collations, `LIKE` comparisons against a pattern that ended with the escape character returned incorrect results. (Bug #25140629, Bug #83930)
- The default character set and collation were used instead of the character set and collation defined in `db.opt` file when upgrading the schema from MySQL 5.7 to MySQL 8.0. (Bug #25139901)
- Instead of updating the data dictionary cache at the end of DDL statements using separate function calls, the data dictionary cache is now updated as part of transaction commit. (Bug #25095798, Bug #83818)

- `CREATE TABLE` now requires the `FILE` privilege if `DATA DIRECTORY` or `INDEX DIRECTORY` is specified explicitly as a table or partition option. `ALTER TABLE` requires the `FILE` privilege if either option is specified explicitly as a partition option (it ignores them if specified as table options). (Bug #25092566)
- There were some differences between ICU order and MySQL Unicode collations for Hungarian contractions and ligatures. MySQL now follows ICU order. (Bug #25090543)
- The data dictionary failed to initialize and start when using a binary collation. The query string generated to create the data dictionary schema did not add quotes to the binary collation name. (Bug #25054104, Bug #83706)
- Executing a stored procedure containing a query that accessed a view could allocate memory that was not freed until the session ended. (Bug #25053286)
- Compilation on FreeBSD 11 failed attempting to check `MAP_NORESERVE`, which is no longer defined. (Bug #25048128, Bug #83689)
- `mysql-test-run.pl` now checks whether the `TSAN_OPTIONS` environment variable is set. If so, the value is taken as the path name of a file containing ThreadSanitizer suppressions (errors to be ignored during test runs). Additionally, the `--sanitize` option now causes `mysql-test-run.pl` to scan the server error logs for ThreadSanitizer messages. (Bug #24970905, Bug #83601)
- After starting the server with a non-zero `--lower-case-table-names` setting, an assertion was raised when `USE INFORMATION_SCHEMA;` was the first instruction from the first client. (Bug #24963580)
- For case-insensitive Unicode collations, the various space characters did not hash to the same value, resulting in incorrect comparisons between them. (Bug #24956750, Bug #83549)
- Dictionary clients now track uncommitted dictionary objects that are being modified by DDL statements, making uncommitted changes by the same session visible to a dictionary client without affecting the dictionary object cache prior to commit time. (Bug #24956365, Bug #83548)
- Simultaneous dictionary object cache misses were not handled correctly. An object retrieved by one thread could be evicted from the cache before a waiting thread attempted to access it. (Bug #24949179)
- `CMake` now avoids configuring the `-fexpensive-optimizations` option for GCC versions for which the option triggers faulty shift-or optimizations. (Bug #24947597, Bug #83517)
- `CMake` support was added for compiling using the `-std=c++03` option under Developer Studio 12.5. This is now used rather than `stlport` by default. (Bug #24947136, Bug #83512, Bug #25229424)
- `NCHAR` and `NATIONAL CHAR` are synonyms, but `CAST(expr AS NCHAR)` succeeded, whereas `CAST(expr AS NATIONAL CHAR)` did not. Now both work. (Bug #24934161)
- OEL RPM packages now better detect which platforms have multilib support (for which 32-bit and 64-bit libraries can be installed). Thanks to Alexey Kopytov for the patch. (Bug #24925181, Bug #83457)
- OEL RPM packages now better detect which platforms do not have multilib support (for which 32-bit and 64-bit libraries can be installed). Thanks to Alexey Kopytov for the patch. (Bug #24916428, Bug #83428)
- The `LOCATE()` function returned `NULL` if the `substr` or `str` argument was `NULL`, but not if the `pos` argument was `NULL`. Now it returns `NULL` if any argument is `NULL`. (Bug #24911350, Bug #83427)
- Bit operations could cause a server exit to occur if argument nullability was mishandled. (Bug #24910958, Bug #24930038, Bug #24930829)

- The Block Nested Loop algorithm could allocate too much memory during query execution. (Bug #24909223)
- Information about building MySQL 5.6 compatibility libraries in the MySQL 5.7 and higher `.spec` file is needed only for building `libmysqlclient` and `libmysqld`. Information about building the InnoDB memcached plugin was removed. (Bug #24908345, Bug #83409)
- Incorrect updating of view metadata could raise an assertion. (Bug #24834622)
- Initialization of the `keyring_okv` plugin failed if the `STANDBY_SERVER` setting was missing from the `okvclient.ora` configuration file, effectively making this a mandatory setting. `STANDBY_SERVER` is now optional. (Bug #24816271)
- Data dictionary objects acquired by `Dictionary_client::acquire_uncached()` are now owned by the current auto-releaser instead of the caller. Also, acquisition of `TABLE_SHARE` view objects is performed by `acquire()` and `clone()` instead of `acquire_uncached()`. (Bug #24813358, Bug #83296)
- Privilege checking could be incorrect for a derived table used within a multiple-table `UPDATE` invoked within a stored procedure or view object, for the second or subsequent execution of the object, if the derived table was merged into the outer query. (Bug #24810564)
- The `Created_tmp_tables` status variable was incremented in some cases when no temporary table was created. (Bug #24808970, Bug #83287)
- An in-place upgrade from MySQL 5.7 to MySQL 8.0 failed if parsing of a stored routine body failed while migrating the routine. Now a warning is reported and the routine is created without parsing its body. Also, warnings are now reported when dependency resolution fails for a view during view migration. (Bug #24805140, Bug #83275)
- For compilation, `mysql_upgrade` is dependent on the dynamically generated `sql_commands_system_tables_data_fix.h` file, but a missing dependency could cause that file not to be generated. (Bug #24802377, Bug #83272)
- A race condition between transactions accessing the access-control list (ACL) cache and use of the ACL cache to populate `INFORMATION_SCHEMA` tables could cause a server exit. (Bug #24786029)
- For debug builds, `EXPLAIN` or `DESCRIBE` for a table with a database or table name longer than the maximum permitted length raised an assertion rather than displaying an appropriate error. (Bug #24751177, Bug #83114)
- The implementation of several `INFORMATION_SCHEMA` tables as views on data dictionary tables introduced a number of native SQL functions intended only for internal use by the server, but they could be invoked by users. Those functions now produce an error if invoked by users. (Bug #24749248, Bug #83189)
- `CASE`, `COALESCE()`, `IF()` and `IFNULL()` could merge a mix of signed and unsigned arguments incorrectly and produce an incorrect result type. (Bug #24733658, Bug #83148)
- Connections from a client to a server with SSL enabled succeeded even if `--ssl-mode` had a value of `VERIFY_CA` or `VERIFY_IDENTITY` and the client did not provide a CA certificate. (Bug #24732452, Bug #23189252, Bug #25397416, Bug #84508)
- Manual creation of a directory in the data directory resulted in `USE dir_name` succeeding even though the directory was not registered as a database in the data dictionary. (Bug #24732194, Bug #83140)
- If InnoDB statistics were incorrect, `FOUND_ROWS()` could return 1 even when the previous `SELECT` returned no rows. (Bug #24714857, Bug #83110)

- `ALTER TABLE` on a temporary table could raise an assertion if a nontemporary table with the same name existed. (Bug #24713918, Bug #83117)
- `CMake` now sets `-DWITH_NUMA=ON` for Debian platforms where possible. (Bug #24689101)
- To better provide atomic file creation, Debian packaging scripts now use the coreutils `install` command rather than `touch`, `chmod`, and `chown`. (Bug #24688682)
- Enabling the `DISABLE_SHARED CMake` option caused compilation failure. (Bug #24687701, Bug #83039)

References: This issue is a regression of: Bug #24481181.

- The `BIN()`, `OCT()`, and `HEX()` functions could mishandle values of the `BIT` data type. (Bug #24686658, Bug #83031)
- For debug builds, an invalid `utf8` character in the comment of a `CREATE EVENT` or `ALTER EVENT` statement raised an assertion. This now produces an `ER_INVALID_CHARACTER_STRING` error. (Bug #24679962)
- The encoding for the `utf8` character set permitted characters between U+D800 and U+DFFF as valid, though they are reserved for surrogate pairs and do not directly represent characters. They are now considered invalid. (Bug #24672415)
- A query could produce incorrect results if the `WHERE` clause contained a dependent subquery, the table had a secondary index on the columns in the select list followed by the columns in the subquery, and `GROUP BY` or `DISTINCT` permitted the query to use a Loose Index Scan. (Bug #24671968, Bug #83005)
- After a `RENAME TABLE` operation on a table with a foreign key, the generated constraint name was not updated in `TABLE_CONSTRAINTS`. (Bug #24666169)
- On macOS, `CMake` AddressSanitizer support did not work. (Bug #24661626, Bug #82976)

References: This issue is a regression of: Bug #23759968.

- Compilation failed on macOS 10.11 with Xcode 8.0. (Bug #24661523, Bug #82975)
- Index hints applied to invisible indexes produced no error. (Bug #24660093, Bug #82960)
- In some cases, `INFORMATION_SCHEMA.KEY_COLUMN_USAGE` did not report data for all foreign key constraints. (Bug #24655803, Bug #82961)
- `REPLACE()` on large strings could be slow and unkillable. (Bug #24652792)
- The `DebugPrintTest` and `DebugPrintDeathTest` unit tests did not handle divide-by-zero testing properly on the Aarch64 platform. Thanks to Alexey Kopytov for the patch. (Bug #24624555, Bug #82889)
- Compilation on FreeBSD using GCC 6 did not work. (Bug #24619561, Bug #82922)
- Changes made to `mysqld_safe` in recent MySQL releases require the `--ledir`, `--mysqld`, `--mysqld-version` options to be specified on the command line; they can no longer be specified in option files. This could cause failure of init scripts that invoke `mysqld_safe`. Such scripts now pass the value of the `MYSQLD_OPTS` environment variable as the first command-line argument to `mysqld_safe`, with the value set to such command line-only `mysqld_safe` option values as may be required. On platforms that use `systemd`, the `MYSQLD_OPTS` value can be set in `/etc/sysconfig/mysqld` with a line such as this:

```
MYSQLD_OPTS=" --ledir=/mysqld_ledir --mysqld=my_wrapper "
```

The value of `MYSQLD_OPTS` can also include `mysqld` options for `mysqld_safe` to pass to `mysqld`. (Bug #24619033, Bug #82920)

References: This issue is a regression of: Bug #24464380, Bug #24483092, Bug #25088048, Bug #25378439, Bug #25378565.

- `SET PERSIST innodb_buffer_pool_size = value` stored the original value of `innodb_buffer_pool_size` to `mysqld-auto.cnf`, not the new value. (Bug #24613005, Bug #82905)
- For SLES packages, a typo in the installation script postamble prevented some cleanup from occurring. (Bug #24605300, Bug #82389)
- Some messages written by the server to the error log while upgrading the data directory were missing the standard timestamp and process ID information. (Bug #24600054, Bug #82874)
- It was possible for the session values of the `Last_query_cost` and `Last_query_partial_plans` status variables to be accessed before they had been initialized. (Bug #24596263)
- Warnings occurring during `CREATE TABLE ... SELECT` could cause a server exit. (Bug #24595992)
- For `LOAD DATA` used to insert data into an updateable view, the check to verify whether a column is actually updatable was missing. (Bug #24595937)
- The server could dereference a null pointer when a deterministic function returning `LONGTEXT` was used in a subquery. (Bug #24595581)
- A view altered with `ALTER VIEW` might be dropped if the statement failed with an error. (Bug #24594140)
- Conversion of `JSON` documents to string could be slow if the document was large and contained many signed integers. (Bug #24586888)
- The data dictionary can contain entries for temporary tables (names beginning with `#sql`). These tables were exposed to `INFORMATION_SCHEMA` queries and through `SHOW` statements. This could cause `mysqldump` and `mysqlpump` to fail when they attempted to dump such tables. Temporary tables are now hidden to `INFORMATION_SCHEMA` queries and `SHOW` statements. (Bug #24580599, Bug #24571427)
- When a view was defined, the character set of the definition was not considered. If this differed from the default character set, table names used in the view might not be recognized, causing the view to be marked invalid.

For debug builds, an assertion could be raised if an error occurred while parsing an `ALTER VIEW` statement and the diagnostics area was empty. (Bug #24580586)

- For `LOAD DATA` statements, input data with too many column values produced only a warning, rather than an error as in MySQL 5.6. An error now occurs. (Bug #24577194, Bug #82830)
- Using `SET PERSIST` with the `global_log` system variable was ineffective. (Bug #24569624, Bug #82807)
- In the Performance Schema `variables_info` table, the `VARIABLE_SOURCE` column was not set properly for some boolean options specified in option files without a variable. (Bug #24567960)
- The `.mylogin.cnf` option file is intended for use by client programs, but the server was reading it as well. The server no longer reads it. (Bug #24557925)

- Use of `boost::chrono` was replaced with `std::chrono` because the former was causing link errors. (Bug #24556808, Bug #82781)
- The X Plugin was built with compilation options different from other plugins. (Bug #24555770, Bug #82777)
- When populating the `variables_by_thread` table, the Performance Schema could attempt to access session variables of other threads that were being deinitialized. (Bug #24555658)
- Users without proper privileges could load and unload server components. (Bug #24528148)
- Concurrent execution of `INSTALL COMPONENT` and `UNINSTALL COMPONENT` statements could cause a server exit. (Bug #24527148)
- On Debian/Ubuntu platforms, the systemd startup script for MySQL ignored `datadir` settings in `/etc/mysql/my.cnf`. (Bug #24517024, Bug #82709)
- With a `LOCK TABLES` statement in effect, DML statements on a table that had triggers could cause a server exit. (Bug #24506766)
- Parallel slave threads running account-management statements could fail due to a race condition in handling privilege cache locks. (Bug #24503606)
- Executing `ALTER TABLE` on a table that has triggers concurrently with other DDL operations could result in a corrupted data dictionary. (Bug #24497803)
- If `mysqladmin shutdown` encountered an error determining the server process ID file, it displayed an error message that did not clearly indicate the error was nonfatal. It now indicates that execution continues. (Bug #24496214)
- For the `null_audit` plugin, setting the `null_audit_event_record` system variable improperly could cause a server exit. This variable should be set only from within the `null_audit` plugin, so it is now read only. (Bug #24493829, Bug #82670)
- In certain contexts, an expression such as `IF(col_name > 5000, (1 / col_name), 5000)` could get a type of `DECIMAL(6,4)`, which would truncate results. (Bug #24492965, Bug #82668)
- The data structure used for `ZEROFILL` columns could experience memory corruption, leading eventually to a server exit. (Bug #24489302)
- Operation of the `mysql-multi.server.sh` script was based on `my.cnf` in the data directory. That option file is no longer used, so `mysql-multi.server.sh` has been removed. (Bug #24487870)
- A query could produce incorrect results if `MIN()` or `MAX()` in a subquery referred to an indexed column. (Bug #24484060, Bug #82638, Bug #24657798, Bug #82965)
- `SHOW TRIGGERS` output order could differ on Linux and Windows. (Bug #24482919, Bug #82637)
- `mysqld_safe` attempted to read `my.cnf` in the data directory, although that is no longer a standard option file location. (Bug #24482156)
- After running `mysql_upgrade`, executing an `INSTALL COMPONENT` statement could cause a server exit. (Bug #24453571)
- A regular expression pattern match into a large string could result in a server exit due to memory allocation failure or integer overflow. (Bug #24449076, Bug #24449090)
- Starting the server with a MySQL 5.7 data directory resulted in failure due to absence of the `default_roles` and `role_edges` system tables. (Bug #24447771)

- An incorrect error was reported for `CREATE TABLE` statements with a large value for the `CONNECTION` table option. The value is now limited to 1024 bytes. (Bug #24437124)
- An assertion could be raised if an `ER_LOCK_WAIT_TIMEOUT` error occurred during execution of `SHOW CREATE TRIGGER`. (Bug #24420809, Bug #82483)
- Constant folding could produce incorrect results for large unsigned integers. (Bug #24401273, Bug #82425)
- Use of very long subpartition names could result in a server exit. Now partition or subpartition names larger than 64 characters produce an `ER_TOO_LONG_IDENT` error. (Bug #24400628, Bug #82429)
- The `Gis_wkb_vector<Gis_point>` copy constructor was not explicitly instantiated, causing build problems for the Intel compiler. (Bug #24397833, Bug #82358)
- Privilege escalation was possible by exploiting the way `REPAIR TABLE` used temporary files. (Bug #24388746)
- A race condition between `UNINSTALL PLUGIN` and `SHOW PLUGINS` could result in a server exit. (Bug #24344026)
- With `ROW` mode binary logging, component installation raised an assertion. (Bug #24343582)
- A potential memory leak related to roles was fixed. (Bug #24337928)
- Subqueries that were converted to semi-joins and programmatically generated an `INFORMATION_SCHEMA` table could incorrectly treat the `INFORMATION_SCHEMA` table as empty.
A workaround for this problem prior to the bug fix: `SET optimizer_switch='semijoin=off';` (Bug #24287772, Bug #82214)
- `mysqldumpslow` failed to parse timestamps in the slow query log; it had not been updated to track a change in log timestamp format. (Bug #24007040)
- The AppArmor profile installed by Ubuntu packages was missing an entry permitting `libnuma` to read a `/sys` hierarchy path, resulting in server startup failure. (Bug #23854929)
- The optimizer could incorrectly treat `RAND()` as a constant for queries subjected to semi-join transformation. (Bug #23854015)
- For debug builds, unequal-length binary operands for bit operators could raise an assertion. (Bug #23853628)
- If a transaction rollback request rolled back only a statement and not the whole transaction, any attempt at attaching another operation to the transaction raised an assertion. (Bug #23753319, Bug #82143)
- `STRCMP()` for arguments with the `utf8mb4_unicode_ci` collation could return results different from `ORDER BY`. `STRCMP()` has been corrected. (Bug #23752284, Bug #82132)
- Complete logical backups made with `mysqlpump` could not be restored if GTIDs were enabled.
To enable control over GTID information written to the dump file, `mysqlpump` now has a `--set-gtid-purged` option that indicates whether to add a `SET @@global.gtid_purged` statement to the output. (Bug #23748432)
- Infinite recursion could occur if the `audit_log` plugin signalled an error while handling an error. (Bug #23717558, Bug #82052)
- Messages written by the `audit_log` plugin to the error log regarding `MYSQL_AUDIT_CONNECT` event failures now print the underlying error cause as well to aid debugging. (Bug #23710632)

- MySQL now uses `readdir()` rather than `readdir_r()`. The latter has been deprecated since `glibc 2.24` and caused debug builds of MySQL and builds using GCC 6.1 to fail.

Additionally, several problems resulting in GCC 6.1 compiler warnings were corrected. (Bug #23708395, Bug #24437737, Bug #82515, Bug #24459890, Bug #25103242)

- `FORCE INDEX` was ineffective for `SELECT COUNT(*)` queries. (Bug #23596760, Bug #81854)
- During startup, the server creates a lock file for the Unix socket file (for example, `mysql.sock.lock` as a lock file for `mysql.sock`). If the server failed to write the process ID to the lock file, it failed to remove that file, which could cause subsequent server startups to fail until the file was removed manually. (Bug #23582603, Bug #81838)
- For debug builds, queries executed using `Item_func_spatial_collection::val_str()` could raise an assertion. (Bug #23573720)
- For audit log events in the connection class, the `connection_type` value was available only for connect events. The value is now available in connect, disconnect, and change-user events. (Bug #23541550)
- The `audit_log` plugin `audit_log_filter_remove_filter()` function caused a server exit if given a `NULL` argument. (Bug #23522793)
- Attempts to configure MySQL without the `FEDERATED` storage engine failed (the `CMake -DWITH_FEDERATED_STORAGE_ENGINE=0` option did not work). (Bug #23508203, Bug #81665)
- On Solaris, `gettimeofday()` could return an invalid value and cause a server shutdown. (Bug #23499695)
- The `keyring_file` plugin could attempt to write keys to its storage file when the file did not exist. To ensure that keys are flushed only when the correct storage file exists, `keyring_file` now stores a SHA-256 checksum of the keyring in the file. Before updating the file, the plugin verifies that it contains the expected checksum. (Bug #23498254)
- `START GROUP REPLICATION` uses stacked `Srv_session` and did not return to the correct thread. `START GROUP REPLICATION` and `STOP GROUP REPLICATION` are now removed from the list of permitted commands. (Bug #23337984)
- A union query resulting in tuples larger than `max_join_size` could result in a server exit. (Bug #23303485)
- Grant tables with incorrect structure may cause problems in user management operations. As a consequence of the fix for this, for any operation that modifies a grant table, the server now checks whether the table has the expected structure and produces an error if not. `mysql_upgrade` must be run to update the tables to the expected structure. (Bug #23295423, Bug #25095876, Bug #25448037)
- `ST_ExteriorRing()` could cause a server exit due to being passed an invalid WKB string believed to be valid. (Bug #23280574)
- For debug builds, failure to prepare a branch of an XA transaction could lead to a server exit. (Bug #23264552, Bug #81375)
- The optimizer could choose `ref` access on a secondary index rather than `range` access on the primary key, even when the cost was higher. (Bug #23259872, Bug #81341)
- An instance of the disk-full error message contained the wrong error code. (Bug #23247332, Bug #81346)

- For a query with `ORDER BY` and `LIMIT`, an optimizer trace did not record the optimizer's switch to a different index. (Bug #23227428, Bug #81250)
- Improper handling of a lock used by the `version_tokens` plugin and user-defined functions could result in a server exit if a UDF was called while `version_tokens` was being uninstalled. (Bug #23210850)
- Certain errors in DML statements executed within stored programs could be mishandled and result in a server exit. (Bug #23209989)
- The `QUOTE()` function could allocate excessive memory. A limit of `max_allowed_packet` bytes is now imposed and returns `NULL` with a warning for attempts to allocate more. (Bug #23195404)
- For some deeply nested expressions, the optimizer failed to detect stack overflow, resulting in a server exit. (Bug #23135667)
- For sessions created through the X Plugin, incorrect thread attachment/detachment could cause a server exit. (Bug #23057045)
- When attempting to locate the data directory, `mysqld_safe` incorrectly considered `$MY_BASEDIR_VERSION/var` as one of the possible locations. (Bug #23013510, Bug #80866)
- For some generated columns, character set conversion of the column definition for table rebuilds could change column values. (Bug #22991924)
- The OS X DMG installer did not properly set up keyring plugin installation. (Bug #22991650)
- If a query performed a `GROUP BY` on a column of a derived table and the select list contained an expression mixing an aggregate function and the group column, an error was raised if the `ONLY_FULL_GROUP_BY` SQL mode was enabled. (Bug #22924183, Bug #80726)
- A binary (in-place) upgrade from MySQL 5.6 to 5.7 followed by a data export performed using `mysqlpump` resulted in an `Invalid default value for date_column` error for attempts to reload the dump file. (Bug #22919028, Bug #80706)
- A failed cast of a long float to integer could cause a server exit. (Bug #22907691)
- On Unix and Unix-like systems, the error log file resulting from specifying `--log-error` without an option value was incorrectly created in the directory of the PID file if the `--pid-file` option was also given. (Bug #22900354)
- The `main.log_tables-big` test case could be unstable on highly loaded hosts. Thanks to Laurynas Biveinis for the patch. (Bug #22874167, Bug #80607)
- SQL statements executed through the X Plugin were not instrumented in the Performance Schema. (Bug #22859462)
- The `rpl.rpl_key_rotation` test case did not synchronize properly with the master server. Thanks to Laurynas Biveinis for the patch. (Bug #22838596, Bug #80531)
- `DROP INDEX` operations could fail due to inconsistent handling of index prefix lengths for `TEXT`-type columns (`TINYTEXT` and so forth).

A consequence of this fix is more restrictive behavior for `CREATE TABLE` and `CREATE INDEX` statements for which a specified index prefix exceeds the maximum column data type size:

- For a nonunique index, either an error occurs (if strict SQL mode is enabled), or the index length is reduced to lie within the maximum column data type size and a warning is produced (if strict mode is not enabled).

- For a unique index, an error occurs regardless of SQL mode because reducing the index length might enable insertion of nonunique entries that do not meet the specified uniqueness requirement.

(Bug #22740093, Bug #80392)

- Metadata locking on stored routine names was performed on a case-sensitive basis, but routine names are not case sensitive. (Bug #22700385)
- Queries that used an aggregate function with `DISTINCT` could produce incorrect results. (Bug #22686994, Bug #80310)
- The `innodb_numa_interleave` system variable was erroneously available on some systems that were not NUMA-enabled. Thanks to Tomislav Plavcic for the patch.

`CMake` now sets the default `WITH_NUMA` value based on whether the current platform has `NUMA` support. For platforms without `NUMA` support, `CMake` behaves as follows:

- With no `NUMA` option (the normal case), `CMake` continues normally, producing only this warning: NUMA library missing or required version not available
- With `-DWITH_NUMA=ON`, `CMake` aborts with this error: NUMA library missing or required version not available

(Bug #22678436, Bug #80288)

- Certain SQL queries involving complex `WHERE` conditions could cause warnings, memory corruption, or a server exit. (Bug #22671573)
- When taking the server offline, a race condition within the Performance Schema could lead to a server exit. (Bug #22551677)
- On macOS, if a table with an associated trigger was renamed to a new name containing both lowercase and uppercase characters, `DROP TRIGGER` for the trigger resulted in an `ER_NO_SUCH_TABLE` error for the table. (Bug #22512899, Bug #79873)
- In the `MYSQL_FIELD` C API structure, the `org_table` value for derived tables was `*`, which could cause failure for queries that depend on this value. The `org_table` value for views and derived tables now is set as follows: If the column is selected from a view, `org_table` names the view. If the column is selected from a derived table, `org_table` names the base table. If a derived table wraps a view, `org_table` still names the base table. If the column is an expression, `org_table` is the empty string. (Bug #22364401, Bug #79641)
- For RPM packages, the default `error-log` location in the deployed `/etc/my.cnf` file differed from the location in the installed `logrotate` script, causing `logrotate` to fail. (Bug #22322685)
- The Performance Schema `events_statements_summary_by_digest` table could contain multiple rows for the same statement digest and schema combination, rather than the expected single (unique) row. (Bug #22320066, Bug #79533)
- For Performance Schema system and status variable tables, variable values expressed in a character set different from `utf8` could be truncated or incorrect. (Bug #22313205)
- For debug builds, altering a table partitioning expression using an expression attribute with an invalid UTF-8 name caused an assertion to be raised. (Bug #22152229)
- After performing inserts in a table containing an `AUTO_INCREMENT` column and then performing a `SELECT` operation, the `LAST_INSERT_ID()` returns the correct value, but the value of the `mysql_insert_id()` C API function was being reset to 0. (Bug #22028117, Bug #78778)

- On Ubuntu, error messages were displayed during upgrades from Community to Commercial packages that made it appear as though `mysqld` and `my_print_defaults` had not been installed. Those messages were spurious and have been silenced. (Bug #21807248)
- With the `use_index_extensions` flag of the `optimizer_switch` system variable disabled, some `SELECT DISTINCT` queries could return incorrect results. (Bug #21749123, Bug #78244)
- An invalid string value in the `WHERE` clause of an `UPDATE` statement, caused an index scan rather than a range scan to be used. For values not present in the index, this could be much slower. Now the optimizer determines this to be an “impossible `WHERE`” condition. (Bug #21032418, Bug #76933)
- Debian packages were missing an AppArmor-related include file and incorrectly were marked dependent on AppArmor (making it impossible to disable AppArmor by uninstalling it). (Bug #20768958)
- When the `automatic_sp_privileges` system variable was enabled, it did not have the expected effect for anonymous users. (Bug #20266641)
- In a replication environment, `SET PASSWORD` or `ALTER USER` could fail to execute on the slave due to failure to parse the hash string correctly. (Bug #20228478)
- The optimizer `resolve_const_item()` function called `Item_decimal()` with the last two arguments in the wrong order. (Bug #19062566)
- An in-place `ALTER TABLE` operation failed to report an error when adding a `DATE` or `DATETIME` column under these conditions: a) the column was `NOT NULL` and no default value was supplied; b) strict and `NO_ZERO_DATE` SQL modes were enabled; c) the table was not empty.

An `ALTER TABLE` operation failed with an error rather than a warning when adding a `DATE` or `DATETIME` column under these conditions: a) the column was `NOT NULL` and no default value was supplied; b) strict SQL mode was enabled and `NO_ZERO_DATE` SQL mode was not enabled; c) the table was not empty. (Bug #16888677)

- Inserting a `TIME`, `DATE`, or `TIMESTAMP` value with a fractional seconds part into a column having the same type but fewer fractional digits resulted in rounding. This differs from MySQL 5.5, which used truncation rather than rounding. To enable control over this behavior, a new `TIME_TRUNCATE_FRACTIONAL` SQL mode is available. The default is to use rounding. If this mode is enabled, truncation occurs instead. (Bug #16583910, Bug #68760)
- `NOT IN` subqueries could produce incorrect results when an index prefix of an inner table was used. (Bug #13915291)
- On non-Linux Unix systems, the `mysql.server` startup script used the Linux command `pidof` rather than `pgrep`. (Bug #13788154, Bug #64342)
- Starting multiple instances of `mysqld_safe` after an abnormal server exit could result in one `mysqld_safe` instance killing another. As a consequence of the bug fix, the `mysqld_safe.pid` file is no longer used. (Bug #11751149, Bug #41908)
- The `--help` message for `mysqld_safe` was corrected to mention that the `--no-defaults`, `--defaults-file`, and `--defaults-extra-file` options, if given, must be the first argument. (Bug #11745176, Bug #11192)
- The bounds check for the XML parser position stack for each level (which has a fixed depth) used the size of the array as the upper limit, and so was off by one. This is fixed by decreasing the allowable depth by one, which actually matches the maximum number of elements in the position stack. (Bug #83871, Bug #25111907)

References: See also: Bug #14040071, Bug #15948580.

Changes in MySQL 8.0.0 (2016-09-12, Development Milestone)



Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.

- [Account Management Notes](#)
- [C API Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Data Dictionary Notes](#)
- [Data Type Notes](#)
- [Doxygen Notes](#)
- [Optimizer Notes](#)
- [Packaging Notes](#)
- [Parser Notes](#)
- [Performance Schema Notes](#)
- [Security Notes](#)
- [Spatial Data Support](#)
- [Test Suite Notes](#)
- [X Plugin Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Account Management Notes

- **Incompatible Change:** The grant tables in the `mysql` system database are now `InnoDB` (transactional) tables. Previously, these were `MyISAM` (nontransactional) tables. This change applies to these tables: `user`, `db`, `tables_priv`, `columns_priv`, `procs_priv`, `proxies_priv`.

The change of grant table storage engine underlies an accompanying change to the behavior of account-management statements. Previously, an account-management statement that named multiple

users could succeed for some users and fail for others. Now, each statement is transactional and either succeeds for all named users or rolls back and has no effect if any error occurs. The statement is written to the binary log if it succeeds, but not if it fails; in that case, rollback occurs and no changes are made. The preceding behavior applies to these statements: `ALTER USER`, `CREATE ROLE`, `CREATE USER`, `DROP ROLE`, `DROP USER`, `GRANT`, `RENAME USER`, `REVOKE`. (`SET PASSWORD` is not listed because it applies to at most one user and is effectively transactional already.) A side effect of this change in behavior is that partially completed account management statements on a MySQL 5.7 master fail when replicated on a MySQL 8.0 slave. For more information, see [Atomic Data Definition Statement Support](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database.



Note

If MySQL is upgraded from an older version but the grant tables have not been upgraded from `MyISAM` to `InnoDB`, the server considers them read only and account-management statements produce an error.

Due to the change of storage engine from `MyISAM` to `InnoDB`, `SELECT` without `ORDER BY` on grant tables can produce different row orders than previously. If a query result must have specific row ordering characteristics, include an `ORDER BY` clause.

- MySQL now supports roles, which are named collections of privileges. Roles enable assignment of sets of privileges to accounts and provide a convenient alternative to granting individual privileges, both for conceptualizing desired privilege assignments and implementing them:
 - Roles can be created and dropped.
 - Roles can have privileges granted to and revoked from them.
 - Roles can be granted to and revoked from user accounts.
 - The active roles for an account can be selected from among those granted to the account, and can be changed during sessions for that account.

For more information, see [Using Roles](#).



Note

`ROLE` now is a reserved word and cannot be used as an identifier without identifier quoting.

C API Notes

- The `libmysqlclient` shared library major version number is increased from 20 (used in MySQL 5.7) to 21 for MySQL 8.0. (Bug #77600, Bug #21363863)

Character Set Support

- The `utf8mb4` Unicode character set has a new general collation named `utf8mb4_0900_ai_ci`. `utf8mb4` also has several new language-specific collations with characteristics similar to `utf8mb4_0900_ai_ci` except that language-specific rules take precedence where applicable. The language-specific collations are indicated by ISO 639-1 language codes in the collation name, as shown in the following table. In two cases the language code has an additional item that denotes a variant (German phone book order, Traditional Spanish).

Table 5 utf8mb4 UCA 9.0.0 Language-Specific Collations

Language	Collation
Croatian	utf8mb4_hr_0900_ai_ci
Czech	utf8mb4_cs_0900_ai_ci
Danish	utf8mb4_da_0900_ai_ci
Esperanto	utf8mb4_eo_0900_ai_ci
Estonian	utf8mb4_et_0900_ai_ci
German phone book order	utf8mb4_de_pb_0900_ai_ci
Hungarian	utf8mb4_hu_0900_ai_ci
Icelandic	utf8mb4_is_0900_ai_ci
Latvian	utf8mb4_lv_0900_ai_ci
Lithuanian	utf8mb4_lt_0900_ai_ci
Polish	utf8mb4_pl_0900_ai_ci
Classical Latin	utf8mb4_la_0900_ai_ci
Romanian	utf8mb4_ro_0900_ai_ci
Slovak	utf8mb4_sk_0900_ai_ci
Slovenian	utf8mb4_sl_0900_ai_ci
Modern Spanish	utf8mb4_es_0900_ai_ci
Traditional Spanish	utf8mb4_es_trad_0900_ai_ci
Swedish	utf8mb4_sv_0900_ai_ci
Turkish	utf8mb4_tr_0900_ai_ci
Vietnamese	utf8mb4_vi_0900_ai_ci

`utf8mb4_0900_ai_ci` also works as an accent- insensitive, case-insensitive collation for the languages in the following table.

Table 6 Languages for Which utf8mb4_0900_ai_ci is Suitable

Language Name	Language Code
German (dictionary order)	de
English	en
Canadian French (locale fr_CA)	fr
Irish Gaelic	ga
Indonesian	id
Italian	it
Luxembourgian	lb
Malay	ms
Dutch	nl
Portuguese	pt
Swahili	sw

Language Name	Language Code
Zulu	zu

`utf8mb4_da_0900_ai_ci` also works as an accent-insensitive, case-insensitive collation for the languages in the following table.

Table 7 Languages for Which `utf8mb4_da_0900_ai_ci` is Suitable

Language Name	Language Code
Norwegian	no
Norwegian Bokmål	nb
Norwegian Nynorsk	nn

The nonlanguage-specific `utf8mb4_0900_ai_ci` and language-specific `utf8mb4_LANG_0900_ai_ci` Unicode collations each have these characteristics:

- The collation is based on Unicode Collation Algorithm (UCA) 9.0.0 and Common Locale Data Repository (CLDR) v30, is accent insensitive, and case insensitive. These characteristics are indicated by `_0900`, `_ai`, and `_ci` in the collation name. Exception: `utf8mb4_la_0900_ai_ci` is not based on CLDR because Classical Latin is not defined in CLDR.
- The collation works for all characters in the range [U+0, U+10FFFF].
- If the collation is not language specific, it sorts all characters, including supplemental characters, in default order (described following). If the collation is language specific, it sorts characters of the language correctly according to language-specific rules, and characters not in the language in default order.
- By default, the collation sorts characters having a code point listed in the DUCET table (Default Unicode Collation Element Table) according to the weight value assigned in the table. The collation sorts characters not having a code point listed in the DUCET table using their implicit weight value, which is constructed according to the UCA.
- For non-language-specific collations, characters in contraction sequences are treated as separate characters. For language-specific collations, contractions might change character sorting order.

For more information, see [Unicode Character Sets](#).

Compilation Notes

- **Microsoft Windows:** For building MySQL on Windows, the toolchain now prefers 64-bit tools when possible (previously 32-bit). This speeds up linking and avoids issues related to limited address space with the 32-bit linker. (Bug #80675, Bug #22900585)
- `CMake` now causes the build process to link with the GNU `gold` linker if it is available. To suppress use of this linker, specify the `-DUSE_LD_GOLD=0` `CMake` option. (Bug #23759968, Bug #82163)
- The `WITH_EXTRA_CHARSETS` `CMake` option has been removed. MySQL builds are configured with all character sets by default now. Users who want fewer character sets can edit `cmake/character_sets.cmake` directly and recompile the server. (Bug #80005, Bug #22552125)
- The required version of the Boost library for server builds has been raised from 1.59.0 to 1.60.0. (Bug #79380, Bug #22253921)

- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.
- MySQL source code now permits and uses C++11 features. To enable a good level of C++11 support across all supported platforms, the following minimum compiler versions now apply:
 - GCC: 4.8 or higher
 - Clang: 3.4 or higher (Xcode 7 on OS X)
 - Solaris Studio: 12.4 or higher (Solaris client build only)
 - Visual Studio: 2015
 - CMake: On Windows, the required Visual Studio version results in a required CMake version of 3.2.3 or higher

On Solaris, the `stlport` library is no longer used. This makes the `SUNPRO_CXX_LIBRARY` CMake option obsolete, so it has been removed.

Component Notes

- MySQL Server now includes a component-based infrastructure for improving server extensibility:
 - A component provides services that are available to the server and other components. (With respect to service use, the server is a component, equal to other components.) Components interact with each other only through the services they provide.
 - The `INSTALL COMPONENT` and `UNINSTALL COMPONENT` statements provide an SQL interface for component manipulation at runtime.
 - A loader service registers installed components in the `mysql.component` system table, and installs registered components during the startup sequence for subsequent server restarts.

For general information about the component infrastructure and its SQL-level interface, see [MySQL Server Components](#). For information about the internal implementation of components, see <http://dev.mysql.com/doc/dev/mysql-server/latest/>.

Configuration Notes

- **Incompatible Change; InnoDB:** Previously, enabling the `innodb_read_only` system variable prevented creating and dropping tables only for the `InnoDB` storage. As of MySQL 8.0. enabling `innodb_read_only` prevents these operations for all storage engines. Table creation and drop operations modify data dictionary tables in the `mysql` system database, but those tables use the `InnoDB` storage engine and cannot be modified when `innodb_read_only` is enabled. The same principle applies to other table operations that require modifying data dictionary tables, and to operations that modify other tables in the `mysql` database that use the `InnoDB` storage engine, such as the grant tables and the `func` and `plugin` tables. (Bug #21611899)
- The hardcoded memory page size of 8KB for the memory-mapped transaction coordinator was too small for platforms such as ARM64 and PowerPC where the page size is much larger. The server now invokes a system call to get the page size of the current platform rather than using a hardcoded value. A consequence for the `--log-tc-size` option is that the minimum and default values are now 6 times the page size. Also, the value must be a multiple of the page size. Thanks to Alexey Kopytov for the patch. (Bug #23014086, Bug #80818, Bug #26931470, Bug #87995)

- MySQL now supports a `SET PERSIST` variant of `SET` statement syntax, for making configuration changes at runtime that also persist across server restarts. Like `SET GLOBAL`, `SET PERSIST` is permitted for any global system variable that is dynamic (settable at runtime). The statement changes the runtime variable value, but also writes the variable setting to an option file named `mysqld-auto.cnf` in the data directory. At startup, the server processes this file after all other option files. For more information, see [Using Option Files](#), and [SET Syntax for Variable Assignment](#).

To provide information showing how each system variable was most recently set, the Performance Schema now has a `variables_info` table that lists each system variable and the source from which it got its value. See [Performance Schema variables_info Table](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this change into the Performance Schema.

- The deprecated `mysql_install_db` program has been removed from MySQL distributions. Data directory initialization should be performed by invoking `mysqld` with the `--initialize` or `--initialize-insecure` option instead. In addition, the deprecated `--bootstrap` option for `mysqld` that was used by `mysql_install_db` has been removed, and the `INSTALL_SCRIPTDIR` CMake option that controlled the installation location for `mysql_install_db` has been removed.

Version 1 test suite code previously was located in the `mysql-test/lib/v1` directory of MySQL source distributions. This code used `mysql_install_db` and has been removed. The `MYSQL_INSTALL_DB` environment variable and a value of 1 for the `MTR_VERSION` environment variable are no longer supported.

Data Dictionary Notes

- Incompatible Change:** MySQL Server now incorporates a global data dictionary containing information about database objects in transactional tables. In previous MySQL releases, dictionary data was stored in metadata files and nontransactional system tables.



Important

A data dictionary-enabled server entails some general operational differences compared to a server that does not have a data dictionary; see [Data Dictionary Usage Differences](#). Also, for upgrades to MySQL 8.0, the upgrade procedure differs somewhat from previous MySQL releases and requires that you verify the upgrade readiness of your installation by checking specific prerequisites. For more information, see [Upgrading MySQL](#), particularly [Verifying Upgrade Prerequisites for Your MySQL 5.7 Installation](#).

`InnoDB` continues to use its own data dictionary in the MySQL 8.0.0 release.

The following list briefly describes the main implications of this change:

- The `.frm` metadata files previously associated with base tables and views no longer exist. Metadata previously stored in `.frm` files is now stored in data dictionary tables.

Similarly, trigger metadata previously stored in `.TRG` and `.TRN` files is stored in a data dictionary table and those files no longer exist.

- With the removal of `.frm` files, the 64KB table definition size limit imposed by the `.frm` file structure is removed.
- With the removal of `.frm` files, the `INFORMATION_SCHEMA.TABLES VERSION` field now reports a hardcoded value of 10, which is the last `.frm` file version used in MySQL 5.7.

- A new dictionary object cache that serves the MySQL data dictionary stores previously accessed data dictionary objects in memory to enable object reuse and minimize disk I/O. An LRU-based eviction strategy is used to evict least recently used objects from memory. The cache comprises several partitions that store different object types. For more information, see [Dictionary Object Cache](#).
- New internal data dictionary APIs enable the server, internal storage engines, and plugins to access and store data in the MySQL data dictionary. Internal data dictionary APIs are introduced for handling of schemas, tablespaces, tablespace files, tables, partitioned tables, table partition data, triggers, stored routines, events, table objects, views, character sets, and collations.

With this change, data dictionary updates and binary log writes for `CREATE TRIGGER` and `DROP TRIGGER` operations are combined into a single, atomic transaction.

- Data dictionary tables are invisible, but in most cases there are corresponding `INFORMATION_SCHEMA` tables that can be queried instead. This enables the underlying data dictionary tables to be changed as server development proceeds, while maintaining a stable `INFORMATION_SCHEMA` interface for application use.

Some `INFORMATION_SCHEMA` tables have been reimplemented entirely as views on data dictionary tables:

```
CHARACTER_SETS
COLLATIONS
COLLATION_CHARACTER_SET_APPLICABILITY
COLUMNS
KEY_COLUMN_USAGE
SCHEMATA
STATISTICS
TABLES
TABLE_CONSTRAINTS
VIEWS
```

Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. In particular, for each `INFORMATION_SCHEMA` table that is a view on data dictionary tables:

- The server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.
- When the underlying data dictionary tables store values previously obtained by directory scans (for example, to enumerate database names or table names within databases) or file-opening operations (for example, to read information from `.frm` files), `INFORMATION_SCHEMA` queries for those values now use table lookups instead. (Additionally, even for a non-view `INFORMATION_SCHEMA` table, values such as database and table names are retrieved by lookups from the data dictionary and do not require directory or file scans.)
- Indexes on the underlying data dictionary tables permit the optimizer to construct efficient query execution plans, something not true for the previous implementation that processed the `INFORMATION_SCHEMA` table using a temporary table per query.

The preceding improvements also apply to `SHOW` statements that display information corresponding to the `INFORMATION_SCHEMA` tables that are views on data dictionary tables. For example, `SHOW DATABASES` displays the same information as the `SCHEMATA` table.

For `INFORMATION_SCHEMA` queries that retrieve table statistics, the server now can use statistics cached in `INFORMATION_SCHEMA` tables, or obtain the latest statistics directly from storage engines. The `information_schema_stats` system variable controls which statistics source the server uses.

- When `information_schema_stats` is `CACHED` (the default), the server uses cached statistics stored in the `STATISTICS` and `TABLES` tables.
- When `information_schema_stats` is `LATEST`, the server obtains statistics directly from storage engines. In this case, the server treats queries on `STATISTICS` and `TABLES` as queries for the latest statistics stored in the `STATISTICS_DYNAMIC` and `TABLES_DYNAMIC` tables.

Affected `INFORMATION_SCHEMA` table statistic columns include:

```
STATISTICS.CARDINALITY
TABLES.AUTO_INCREMENT
TABLES.AVG_ROW_LENGTH
TABLES.CHECKSUM
TABLES.CHECK_TIME
TABLES.CREATE_TIME
TABLES.DATA_FREE
TABLES.DATA_LENGTH
TABLES.INDEX_LENGTH
TABLES.MAX_DATA_LENGTH
TABLES.TABLE_ROWS
TABLES.UPDATE_TIME
```

For more information, see [Optimizing INFORMATION_SCHEMA Queries](#).

- The `foreign_keys` and `foreign_key_column_usage` tables now store foreign key information. The standard SQL way to obtain foreign key information is by using the `INFORMATION_SCHEMA REFERENTIAL_CONSTRAINTS` and `KEY_COLUMN_USAGE` tables; these tables are now implemented as views on the `foreign_keys`, `foreign_key_column_usage`, and other data dictionary tables.

For some foreign key errors, the server now produces more appropriate and more informative error messages.



Note

Incompatibility: Previously, MySQL supported foreign key names longer than 64 characters. Foreign key names as stored in the `foreign_keys` and `foreign_key_column_usage` tables are a maximum of 64 characters, per the SQL standard, so longer foreign key names are no longer permitted.

- Because the data dictionary provides information about database objects, the server no longer checks directory names in the data directory to find databases. Consequently, the `--ignore-db-dir` option and `ignore_db_dirs` system variable are extraneous and have been removed. Update system configurations and application programs accordingly.
- System table changes:
 - Many system tables have been converted from `MyISAM` (nontransactional) tables to `InnoDB` (transactional) tables. For example, as discussed elsewhere in these release notes, the grant tables are now `InnoDB` tables. Other examples follow.
 - The `func` table that stores user-defined function information in the `mysql` system database now is an `InnoDB` (transactional) table. Previously, it was a `MyISAM` (nontransactional) table.

In consequence of this change, `CREATE FUNCTION` and `DROP FUNCTION` statements cause an implicit commit, even when used for user-defined functions (see [Statements That Cause an Implicit Commit](#)). Previously, they caused an implicit commit when used for stored functions, but not for user-defined functions.

- Previously, information about stored routines and events was stored in the `proc` and `event` tables of the `mysql` system database. Those tables are no longer used. Instead, information about stored routines and events is stored in the `routines`, `events`, and `parameters` data dictionary tables in the `mysql` system database. The old tables used the `MyISAM` (nontransactional) storage engine. The new tables use the `InnoDB` (transactional) engine.

Previously, creating a stored routine that contained illegal characters produced a warning. This is now an error.

- To permit access to system tables (for example, time zone or log tables) to be distinguished from access to nonsystem tables, the server uses the `Locking system tables` and `Opening system tables` thread states rather than the `System lock` and `Opening tables` thread states. See [General Thread States](#).
- `InnoDB` changes:
 - Persistent `InnoDB` tablespaces now include transactional storage for *Serialized Dictionary Information (SDI)*, which is dictionary object data in serialized form. Along with the disappearance of `.frm` and trigger metadata files, mentioned previously, you might notice the appearance of `.SDI` files. These are serialized dictionary information files. SDI transactional storage is reserved for an in-progress feature not yet fully implemented.
 - A new command-line utility, `ibd2sdi`, is used to extract serialized dictionary information (SDI) from persistent `InnoDB` tablespaces. SDI data is not present in persistent `InnoDB` tablespaces in this release. The `ibd2sdi` utility is reserved for future use.
 - `InnoDB` startup code was refactored to support MySQL initialization changes related to the MySQL data dictionary feature.
- Upgrade and downgrade implications:
 - To upgrade to MySQL 8.0 from MySQL 5.7, you must perform the upgrade procedure described at [Upgrading MySQL](#).
 - Downgrading from MySQL 8.0 to MySQL 5.7 is only supported using the logical downgrade method (a `mysqldump` downgrade). In-place downgrades are not supported.

(Bug #80481, Bug #22811659)

Data Type Notes

- Bit functions and operators comprise `BIT_COUNT()`, `BIT_AND()`, `BIT_OR()`, `BIT_XOR()`, `&`, `|`, `^`, `~`, `<<`, and `>>`. Prior to MySQL 8.0, bit functions and operators required `BIGINT` (64-bit integer) arguments and returned `BIGINT` values, so they had a maximum range of 64 bits. Non-`BIGINT` arguments were converted to `BIGINT` prior to performing the operation and truncation could occur. Now bit functions and operators permit binary string type arguments (`BINARY`, `VARBINARY`, and the `BLOB` types) and return a value of like type, which enables them to take arguments and produce return values larger than 64 bits. Nonbinary string arguments are converted to `BIGINT` and processed as such, as before.

Permitting binary string arguments for bit functions and operators makes it easier not only to manipulate larger values, but to perform bit operations not easily done previously on certain types of data, such as UUID and IPv6 values. For examples, see [Bit Functions and Operators](#).

An implication of this change in behavior is that bit operations on binary string arguments might produce a different result in MySQL 8.0 than in 5.7. For information about how to prepare in MySQL 5.7 for potential incompatibilities between MySQL 5.7 and 8.0, see [Bit Functions and Operators](#), in [MySQL 5.7 Reference Manual](#).

Doxygen Notes

- The MySQL source code has been updated to use Doxygen for the internal documentation. The generated content for this milestone is available at <http://dev.mysql.com/doc/dev/mysql-server/8.0.0/>. This is a work in progress. As new MySQL versions are distributed, the Doxygen documentation will be updated, with the latest version always available at <http://dev.mysql.com/doc/dev/mysql-server/latest/>.

It is also possible to generate the Doxygen content locally from a MySQL source distribution using the instructions at [Generating MySQL Doxygen Documentation Content](#).

Optimizer Notes

- **InnoDB**: The storage engine interface now enables the optimizer to provide information about the size of the record buffer to be used for scans that the optimizer estimates will read multiple rows. The buffer size can vary based on the size of the estimate. **InnoDB** uses this variable-size buffering capability to take advantage of row prefetching, and to reduce the overhead of latching and B-tree navigation. Previously, **InnoDB** used a small, fixed-size buffer.
- The optimizer now supports table-level **MERGE** and **NO_MERGE** hints for specifying whether derived tables or views should be merged into the outer query block or materialized using an internal temporary table. Examples:

```
SELECT /*+ MERGE(dt) */ * FROM (SELECT * FROM t1) AS dt;  
SELECT /*+ NO_MERGE(dt) */ * FROM (SELECT * FROM t1) AS dt;
```

For more information, see [Optimizer Hints](#). (Bug #79554, Bug #22328100)

- MySQL now supports invisible indexes. An invisible index is not used by the optimizer at all, but is otherwise maintained normally. Indexes are visible by default. Invisible indexes make it possible to test the effect of removing an index on query performance, without making a destructive change that must be undone should the index turn out to be required. This feature applies to **InnoDB** tables, for indexes other than primary keys.

To control whether an index is invisible explicitly for a new index, use a **VISIBLE** or **INVISIBLE** keyword as part of the index definition for **CREATE TABLE**, **CREATE INDEX**, or **ALTER TABLE**. To alter the invisibility of an existing index, use a **VISIBLE** or **INVISIBLE** keyword with the **ALTER TABLE ... ALTER INDEX** operation. For more information, see [Invisible Indexes](#).

- The **mysql** system database now contains a **column_stats** table designed to store statistics about column values. For more information, see [Optimizer Statistics](#).

Packaging Notes

- Development milestone releases in previous MySQL series were numbered using a suffix of **-mN**, to indicate development milestone **N**. In MySQL 8.0, development releases use the suffix **-dmr**. For example, this release of MySQL is numbered **8.0.0-dmr**. (Bug #80408, Bug #22748154)

- As a consequence of the use of C++11 features described elsewhere in these release notes, the following packaging changes have been made:
 - Support for Red Hat Enterprise Linux 5 and Oracle Linux 5 RPMs has been dropped
 - Generic binary tarball builds have been moved to Red Hat Enterprise Linux 6

Parser Notes

- The parser rules for `SELECT` and `UNION` were refactored to be more consistent (the same `SELECT` syntax applies uniformly in each such context) and reduce duplication. Several user-visible effects resulted from this work:
 - `NATURAL JOIN` permits an optional `INNER` keyword (`NATURAL INNER JOIN`), in compliance with standard SQL.
 - Right-deep joins without parentheses are permitted (for example, `... JOIN ... JOIN ... ON ... ON`), in compliance with standard SQL.
 - The parser accepts parentheses around query expressions. For example, `(SELECT ... UNION SELECT ...)` is permitted.
 - The parser better conforms to the documented permitted placement of the `SQL_CACHE` and `SQL_NO_CACHE` query modifiers.
 - Left-hand nesting of unions, previously permitted only in subqueries, is now permitted in top-level statements. For example, this statement is now accepted as valid:

```
(SELECT 1 UNION SELECT 1) UNION SELECT 1;
```

(Bug #11746363, Bug #25734)

- The parser rules for `CREATE TABLE` were refactored to be context independent and improve maintainability and extensibility. Several user-visible effects resulted from this work:
 - For generated columns, including `NOT NULL NULL` resulted in a column that included the `NOT NULL` attribute, which differed from nongenerated columns. Such definitions now use the final attribute `NULL`, resulting in a nullable column (consistent with nongenerated columns).
 - `CREATE TEMPORARY TABLE` no longer permits multiple instances of `TEMPORARY`.
 - Previously, `PARSE_GCOL_EXPR` was a keyword and could not be used as a label in stored programs. It is no longer a keyword and can be used as a label.
 - Messages for some syntax errors are more precise with respect to the location of the error within the statement.

Performance Schema Notes

- **Incompatible Change:** The Performance Schema now instruments server errors (and warnings), and exposes statistical information about them through a set of summary tables:
 - The `error` instrument controls whether error information is collected (enabled by default).
 - Several tables contain error information, summarized in various ways:
`events_errors_summary_global_by_error`,
`events_errors_summary_by_account_by_error`,

```
events_errors_summary_by_host_by_error,
events_errors_summary_by_thread_by_error,
events_errors_summary_by_user_by_error.
```

- The `performance_schema_error_size` system variable controls the number of instrumented errors.

For more information, see [Error Summary Tables](#)

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

In consequence of the preceding changes, two server error symbols were renamed:

`ER_CANT_SET_ENFORCE_GTID_CONSISTENCY_ON_WITH_ONGOING_GTID_VIOLATING_TRANSACTIONS` is now `ER_CANT_ENFORCE_GTID_CONSISTENCY_WITH_ONGOING_GTID_VIOLATING_TX` and `ER_SET_ENFORCE_GTID_CONSISTENCY_WARN_WITH_ONGOING_GTID_VIOLATING_TRANSACTIONS` is now `ER_ENFORCE_GTID_CONSISTENCY_WARN_WITH_ONGOING_GTID_VIOLATING_TX`.

Also, several server error codes were found to be no longer used in the server and have been removed as obsolete. Applications that test specifically for any of these errors should be updated:

```
ER_BINLOG_READ_EVENT_CHECKSUM_FAILURE
ER_BINLOG_ROW_RBR_TO_SBR
ER_BINLOG_ROW_WRONG_TABLE_DEF
ER_CANT_ACTIVATE_LOG
ER_CANT_CHANGE_GTID_NEXT_IN_TRANSACTION
ER_CANT_CREATE_FEDERATED_TABLE
ER_CANT_CREATE_SROUTINE
ER_CANT_DELETE_FILE
ER_CANT_GET_WD
ER_CANT_SET_GTID_PURGED_WHEN_GTID_MODE_IS_OFF
ER_CANT_SET_WD
ER_CANT_WRITE_LOCK_LOG_TABLE
ER_CREATE_DB_WITH_READ_LOCK
ER_CYCLIC_REFERENCE
ER_DB_DROP_DELETE
ER_DELAYED_NOT_SUPPORTED
ER_DIFF_GROUPS_PROC
ER_DISK_FULL
ER_DROP_DB_WITH_READ_LOCK
ER_DROP_USER
ER_DUMP_NOT_IMPLEMENTED
ER_ERROR_DURING_CHECKPOINT
ER_ERROR_ON_CLOSE
ER_EVENTS_DB_ERROR
ER_EVENT_CANNOT_DELETE
ER_EVENT_CANT_ALTER
ER_EVENT_COMPILE_ERROR
ER_EVENT_DATA_TOO_LONG
ER_EVENT_DROP_FAILED
ER_EVENT_MODIFY_QUEUE_ERROR
ER_EVENT_NEITHER_M_EXPR_NOR_M_AT
ER_EVENT_OPEN_TABLE_FAILED
ER_EVENT_STORE_FAILED
ER_EXEC_STMT_WITH_OPEN_CURSOR
ER_FAILED_ROUTINE_BREAK_BINLOG
ER_FLUSH_MASTER_BINLOG_CLOSED
ER_FORM_NOT_FOUND
ER_FOUND_GTID_EVENT_WHEN_GTID_MODE_IS_OFF__UNUSED
ER_FRM_UNKNOWN_TYPE
ER_GOT_SIGNAL
ER_GRANT_PLUGIN_USER_EXISTS
```

```

ER_GTID_MODE_REQUIRES_BINLOG
ER_GTID_NEXT_IS_NOT_IN_GTID_NEXT_LIST
ER_HASHCHK
ER_INDEX_REBUILD
ER_INNODB_NO_FT_USES_PARSER
ER_LIST_OF_FIELDS_ONLY_IN_HASH_ERROR
ER_LOAD_DATA_INVALID_COLUMN_UNUSED
ER_LOGGING_PROHIBIT_CHANGING_OF
ER_MALFORMED_DEFINER
ER_MASTER_KEY_ROTATION_ERROR_BY_SE
ER_NDB_CANT_SWITCH_BINLOG_FORMAT
ER_NEVER_USED
ER_NISAMCHK
ER_NO_CONST_EXPR_IN_RANGE_OR_LIST_ERROR
ER_NO_FILE_MAPPING
ER_NO_GROUP_FOR_PROC
ER_NO_RAID_COMPILED
ER_NO_SUCH_KEY_VALUE
ER_NO_SUCH_PARTITION__UNUSED
ER_OBSOLETE_CANNOT_LOAD_FROM_TABLE
ER_OBSOLETE_COL_COUNT_DOESNT_MATCH_CORRUPTED
ER_ORDER_WITH_PROC
ER_PARTITION_SUBPARTITION_ERROR
ER_PARTITION_SUBPART_MIX_ERROR
ER_PART_STATE_ERROR
ER_PASSWD_LENGTH
ER_QUERY_ON_MASTER
ER_RBR_NOT_AVAILABLE
ER_SKIPPING_LOGGED_TRANSACTION
ER_SLAVE_CHANNEL_DELETE
ER_SLAVE_MULTIPLE_CHANNELS_HOST_PORT
ER_SLAVE_MUST_STOP
ER_SLAVE_WAS_NOT_RUNNING
ER_SLAVE_WAS_RUNNING
ER_SP_GOTO_IN_HNDLR
ER_SP_PROC_TABLE_CORRUPT
ER_SQL_MODE_NO_EFFECT
ER_SR_INVALID_CREATION_CTX
ER_TABLE_NEEDS_UPG_PART
ER_TOO_MUCH_AUTO_TIMESTAMP_COLS
ER_UNEXPECTED_EOF
ER_UNION_TABLES_IN_DIFFERENT_DIR
ER_UNSUPPORTED_BY_REPLICATION_THREAD
ER_UNUSED1
ER_UNUSED2
ER_UNUSED3
ER_UNUSED4
ER_UNUSED5
ER_UNUSED6
ER_VIEW_SELECT_DERIVED_UNUSED
ER_WRONG_MAGIC
ER_WSAS_FAILED

```

- Previously, the `DIGEST` and `DIGEST_TEXT` columns in the Performance Schema `events_statements_current` table were populated only after statement execution ended. Now, the columns are populated just after parsing and before statement execution begins. This enables monitoring applications to access statement digest information during statement execution. (Bug #23336542)
- Previously, Performance Schema optimizations focused on reducing the overhead involved in collecting monitoring data. Complementing that earlier work, overhead now is also reduced for Performance Schema queries that retrieve that data. This is achieved by the addition of indexes to most Performance Schema tables, which gives the optimizer access to execution plans other than full table scans. These

indexes also improve performance for related objects, such as `sys` schema views that use those tables. For more information, see [Optimizing Performance Schema Queries](#).

- The size of the `ROLE` column of the `setup_actors` Performance Schema table was increased from 16 to 32 characters.

Security Notes

- The `validate_password_check_user_name` system variable is now enabled by default rather than disabled. This means that when the `validate_password` plugin is enabled, by default it now rejects passwords that match the current session user name.
- The client-side `--ssl` and `--ssl-verify-server-cert` options have been removed. Use `--ssl-mode=REQUIRED` instead of `--ssl=1` or `--enable-ssl`. Use `--ssl-mode=DISABLED` instead of `--ssl=0`, `--skip-ssl`, or `--disable-ssl`. Use `--ssl-mode=VERIFY_IDENTITY` instead of `--ssl-verify-server-cert` options. (The server-side `--ssl` option remains unchanged.)

For the C API, `MYSQL_OPT_SSL_ENFORCE` and `MYSQL_OPT_SSL_VERIFY_SERVER_CERT` options for `mysql_options()` correspond to the client-side `--ssl` and `--ssl-verify-server-cert` options and have been removed. Use `MYSQL_OPT_SSL_MODE` with an option value of `SSL_MODE_REQUIRED` or `SSL_MODE_VERIFY_IDENTITY` instead.

Spatial Data Support

- Spatial functions for import and export of Well-Known Text (WKT) values used MySQL `'GEOMETRYCOLLECTION()'` nonstandard syntax rather than OpenGIS `'GEOMETRYCOLLECTION EMPTY'` standard syntax. Now both syntaxes are understood for import and the standard syntax is used for export. See [Functions That Create Geometry Values from WKT Values](#). (Bug #23632147, Bug #81964)
- The `ST_X()` and `ST_Y()` spatial functions now permit an optional second argument that specifies an X or Y coordinate value, respectively. With two arguments, the function result is the point value from the first argument with the appropriate coordinate modified. In addition, `ST_X()` and `ST_Y()` with a single argument now are stricter and produce an `ER_UNEXPECTED_GEOMETRY_TYPE` error rather than returning `NULL` if the argument is a valid geometry but not a point. For more information, see [Point Property Functions](#).
- The `ST_SRID()` spatial function now permits an optional second argument that specifies a SRID value. With two arguments, the function result is the geometry value from the first argument with its SRID modified according to the second argument. For more information, see [General Geometry Property Functions](#).
- MySQL now stores information about spatial reference systems other than SRID 0, for use with spatial data. This information is stored in the `st_spatial_reference_systems` data dictionary table and is based on EPSG Dataset 8.7. For information about spatial reference systems, see [Spatial Reference System Support](#).

Previously, the `ST_IsValid()`, `ST_MakeEnvelope()`, and `ST_Validate()` functions required geometry arguments with SRID 0. They now accept geometry arguments with an SRID for a projected spatial reference system.

- In MySQL 5.7, several spatial functions available under multiple names were deprecated to move in the direction of making the spatial function namespace more consistent, the goal being that each spatial function name begin with `ST_` if it performs an exact operation, or with `MBR` if it performs an operation based on minimum bounding rectangles. The deprecated functions have now been removed to leave only the corresponding `ST_` and `MBR` functions:

- These functions are removed in favor of the MBR names: `Contains()`, `Disjoint()`, `Equals()`, `Intersects()`, `Overlaps()`, `Within()`.
- These functions are removed in favor of the ST_ names: `Area()`, `AsBinary()`, `AsText()`, `AsWKB()`, `AsWKT()`, `Buffer()`, `Centroid()`, `ConvexHull()`, `Crosses()`, `Dimension()`, `Distance()`, `EndPoint()`, `Envelope()`, `ExteriorRing()`, `GeomCollFromText()`, `GeomCollFromWKB()`, `GeomFromText()`, `GeomFromWKB()`, `GeometryCollectionFromText()`, `GeometryCollectionFromWKB()`, `GeometryFromText()`, `GeometryFromWKB()`, `GeometryN()`, `GeometryType()`, `InteriorRingN()`, `IsClosed()`, `IsEmpty()`, `IsSimple()`, `LineFromText()`, `LineFromWKB()`, `LineStringFromText()`, `LineStringFromWKB()`, `MLineFromText()`, `MLineFromWKB()`, `MPointFromText()`, `MPointFromWKB()`, `MPolyFromText()`, `MPolyFromWKB()`, `MultiLineStringFromText()`, `MultiLineStringFromWKB()`, `MultiPointFromText()`, `MultiPointFromWKB()`, `MultiPolygonFromText()`, `MultiPolygonFromWKB()`, `NumGeometries()`, `NumInteriorRings()`, `NumPoints()`, `PointFromText()`, `PointFromWKB()`, `PointN()`, `PolyFromText()`, `PolyFromWKB()`, `PolygonFromText()`, `PolygonFromWKB()`, `SRID()`, `StartPoint()`, `Touches()`, `X()`, `Y()`.
- `GLength()` is removed in favor of `ST_Length()`.

Test Suite Notes

- `mysql-test-run.pl` now supports a `--do-suite` option, which is similar to `--do-test` but permits specifying entire suites of tests to run. (Bug #24350345)
- The `mysqltest rmdir` command fails if the directory to be removed contains any files or directories. To enable recursive removal of a directory as well as its contents, if any, `mysqltest` now supports a `force-rmdir` command. (Bug #24316799)
- Two new test suite options make it easier to debug test cases:
 - `mysql-test-run.pl` supports a `--mysqltest=options` option that enables options to be passed to `mysqltest`.
 - `mysqltest` supports a `--trace-exec` option that causes it to immediately print output from executed programs to `stdout`.

`mysql-test-run.pl` now recognizes the `MTR_CTEST_TIMEOUT` environment variable. If set, the value is a timeout in seconds to pass to `ctest` unit test commands. (Bug #21821049, Bug #21278845)

- For test cases in the MySQL test suite, it was previously possible to use symbolic error names for the `--error` command only for server errors. This is now also possible for client errors. For example:

```
--error CR_SERVER_GONE_ERROR
```

(Bug #21048973, Bug #76972)

- The `mysqltest` program now has a `copy_files_wildcard` command that copies all files that match a pattern from a source directory to a destination directory. See http://dev.mysql.com/doc/dev/mysql-server/PAGE_MYSQL_TEST_COMMANDS.html (Bug #82111, Bug #23743035)

X Plugin Notes

- The `Protobuf` decoder class limited the number of nested objects to 50 (the default value). (Bug #23707238, Bug #82025)

- The statement `list_objects` incorrectly reported a table as a collection. (Bug #23631240)
- The `create_collection` statement created a collection table with a unique key index on the `'_id'` column instead of on the primary key. (Bug #23284569)

Functionality Added or Changed

- **Incompatible Change; Partitioning:** The generic partitioning handler has been removed from the MySQL server. As part of this change, `mysqld` no longer supports the `--partition` and `--skip-partition` options, and the server can no longer be built using `-DWITH_PARTITION_STORAGE_ENGINE`. `partition` is also no longer displayed in the output of `SHOW PLUGINS`, or shown in the `INFORMATION_SCHEMA.PLUGINS` table.

In order to support partitioning of a given table, the storage engine used for the table must now provide its own (“native”) partitioning handler. `InnoDB` is the only storage engine supported in MySQL 8.0 which includes a native partitioning handler. An attempt to create partitioned tables in MySQL 8.0 using any other storage engine fails. (The `NDB` storage engine used by MySQL NDB Cluster also provides its own partitioning handler, but is currently not supported by MySQL 8.0.)

Effects on upgrades. The direct upgrade of a partitioned table using a storage engine other than `InnoDB` (such as `MyISAM`) from MySQL 5.7 (or earlier) to MySQL 8.0 is not supported. There are two options for upgrading such a table to be compatible with MySQL 8.0, listed here:

- Remove the table's partitioning; you can do this without any data loss by executing an `ALTER TABLE ... REMOVE PARTITIONING` statement.
- Change the storage engine used for the table to `InnoDB`, using `ALTER TABLE ... ENGINE=INNODB`; this leaves the table's partitioning in place. At least one of these operations must be performed for any partitioned non-`InnoDB` table, prior to upgrading the server to MySQL 8.0. Otherwise, such a table cannot be used following the upgrade.

For information about converting `MyISAM` tables to `InnoDB`, see [Converting Tables from MyISAM to InnoDB](#).

An analogous situation is met when importing databases from a dump file that was created in MySQL 5.7 or earlier using `mysqldump` into a MySQL 8.0 server, due to the fact that table creation statements that would result in a partitioned table using a storage engine without such support fail with an error in MySQL 8.0. For this reason you must ensure that any statements in the dump file creating partitioned tables do not also specify an unsupported storage engine. You can do this either by removing any references to partitioning from `CREATE TABLE` statements that use a value for the `STORAGE ENGINE` option other than `InnoDB`, or by specifying the storage engine as `InnoDB` (or allowing `InnoDB` to be used by default).

For more information, see [Partitioning Limitations Relating to Storage Engines](#).

- **Important Change; InnoDB:** The following `InnoDB` file format configuration parameters were deprecated in MySQL 5.7.7 and are now removed:
 - `innodb_file_format`
 - `innodb_file_format_check`
 - `innodb_file_format_max`
 - `innodb_large_prefix`

File format configuration parameters were necessary for creating tables compatible with earlier versions of InnoDB in MySQL 5.1. Now that MySQL 5.1 has reached the end of its product lifecycle, the parameters are no longer required.

The `FILE_FORMAT` column was removed from the `INNODB_SYS_TABLES` and `INNODB_SYS_TABLESPACES` Information Schema tables.

- **InnoDB:** The `innodb_buffer_pool_debug` option permits multiple buffer pool instances when the buffer pool is less than 1GB in size, ignoring the 1GB minimum buffer pool size constraint imposed on `innodb_buffer_pool_instances`. (Bug #24287290)
- **InnoDB:** A new dynamic configuration option, `innodb_deadlock_detect`, can be used to disable deadlock detection. On high concurrency systems, deadlock detection can cause a slowdown when numerous threads wait for the same lock. At times, it may be more efficient to disable deadlock detection and rely on the `innodb_lock_wait_timeout` setting for transaction rollback when a deadlock occurs. (Bug #23477773)
- **InnoDB:** The `libinnodb_zipdecompress.a` library allows external tools to use the `page_zip_decompress_low()` function to decompress InnoDB pages. (Bug #21405300, Bug #77664)
- **InnoDB:** To address contention that could occur under some workloads, the buffer pool mutex was removed and replaced by several list and hash protecting mutexes. Also, several buffer pool related variables no longer require buffer pool mutex protection. Thanks to Yasufumi Kinoshita and Laurynas Biveinis for the patch. (Bug #20381905, Bug #75534)
- **InnoDB:** InnoDB now avoids intermediate commits that would occur every 10000 rows during `ALTER TABLE ALGORITHM=COPY` operations. The purpose of intermediate commits was to speed up recovery in the case of an aborted `ALTER TABLE ALGORITHM=COPY` operation. If an `ALTER TABLE ALGORITHM=COPY` operation is aborted, the new, uncommitted table is now dropped during DDL log recovery before the undo log is rolled back, thereby avoiding time-consuming data rollback for the uncommitted table. Undo logging is now suppressed for `ALTER TABLE ALGORITHM=COPY` operations unless there is an `IGNORE` clause or something else that requires rollback capability.

If there is full-text index on the table being altered, full-text data is inserted into full-text auxiliary tables as the `ALTER TABLE ALGORITHM=COPY` operation inserts rows into the new, uncommitted table. Previously, full-text data was only processed on transaction commit. (Bug #17479594)

- **InnoDB:** To reduce read-write lock contention that can result from multiple purge threads purging rows from the same table, undo records are now grouped and assigned to different purge threads by table ID.
- **InnoDB:** InnoDB code now uses the C++ `std::thread` library for thread management.
- **InnoDB:** BLOB code was refactored to provide an internal C++ interface for operations on compressed and uncompressed BLOB data.
- **InnoDB:** The InnoDB `memcached` plugin now supports multiple `get` operations (fetching multiple key/value pairs in a single `memcached` query) and range queries. See [InnoDB memcached Multiple get and Range Query Support](#).
- **InnoDB:** The `innodb_stats_sample_pages` system variable was removed. `innodb_stats_sample_pages` was deprecated in MySQL 5.6.3 and replaced by `innodb_stats_transient_sample_pages`.
- **InnoDB:** When encountering index tree corruption, InnoDB writes a corruption flag to the redo log, which makes the corruption flag crash safe. InnoDB also writes in-memory corruption flag data to an

engine-private system table on each checkpoint. During recovery, InnoDB reads corruption flags from both locations and merges results before marking in-memory table and index objects as corrupt.

- **InnoDB:** The `innodb_locks_unsafe_for_binlog` system variable was removed. `innodb_locks_unsafe_for_binlog` was deprecated in MySQL 5.6.3. The `READ COMMITTED` isolation level provides similar functionality.
- **InnoDB:** InnoDB no longer creates `.isl` files (InnoDB Symbolic Link files) when creating tablespace data files outside of the MySQL data directory. Redo log records are now used to locate remote tablespaces.

With this change, moving a remote tablespace while the server is offline by manually modifying an `.isl` file is no longer supported.

- **InnoDB:** InnoDB no longer supports compressed temporary tables. When `innodb_strict_mode` is enabled (the default), `CREATE TEMPORARY TABLE` returns an error if `ROW_FORMAT=COMPRESSED` or `KEY_BLOCK_SIZE` is specified. If `innodb_strict_mode` is disabled, warnings are issued and the temporary table is created using a non-compressed row format.

With this change, all temporary tables are created in the shared temporary tablespace, `ibtmp1`.

The `PER_TABLE_TABLESPACE` and `IS_COMPRESSED` columns were removed from the `INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO`.

- **InnoDB:** A new `INFORMATION_SCHEMA` table, `INNODB_CACHED_INDEXES`, reports the number of index pages cached in the InnoDB buffer pool for each index.
- **InnoDB:** The `innodb_checksums` system variable was removed. `innodb_checksums` was replaced by `innodb_checksum_algorithm` in MySQL 5.6.3.
- **InnoDB:** InnoDB startup code was refactored.
- **InnoDB:** The `innodb_flush_method` default value is no longer `NULL`. On Unix-like systems, the default value is `fsync`. On Windows systems, the default value is `unbuffered`.

On Windows, the `innodb_flush_method` setting no longer affects the `innodb_use_native_aio` setting. There are now two possible settings for `innodb_flush_method` on Windows, `unbuffered` (unbuffered I/O) and `normal` (buffered I/O). With this change, you can enable asynchronous I/O with buffered I/O, which is a new combination (`innodb_use_native_aio=ON` and `innodb_flush_method=normal`). The `async_unbuffered` setting was removed.

You can now set `innodb_flush_method` and `innodb_change_buffering` configuration options using numeric values.

- **InnoDB:** The `innodb_support_xa` system variable, which enables support for two-phase commit in XA transactions, was removed. As of MySQL 5.7.10, InnoDB support for two-phase commit in XA transactions is always enabled.
- **InnoDB:** The new `INFORMATION_SCHEMA.INNODB_CACHED_INDEXES` table reports the number of index pages cached in the InnoDB buffer pool for each index.
- **InnoDB:** InnoDB no longer creates an `.isl` file (InnoDB Symbolic Link files) when creating a file-per-table tablespace data file outside of the MySQL data directory. InnoDB now uses the redo log to locate remote tablespace data files.

Offline relocation of a file-per-table tablespace data file created outside of the MySQL data directory by modifying the `.isl` file is no longer supported.

- **InnoDB:** The current maximum auto-increment counter value is now written to the redo log each time the value changes, and it is saved to an engine-private system table on each checkpoint. These changes make the current maximum auto-increment counter value persistent across server restarts. Additionally:
 - A server restart no longer cancels the effect of the `AUTO_INCREMENT = N` table option. If you initialize the auto-increment counter to a specific value, or if you alter the auto-increment counter value to a larger value, the new value is persisted across server restarts.
 - A server restart immediately following a `ROLLBACK` operation no longer results in the reuse of auto-increment values that were allocated to the rolled-back transaction.
 - If you modify an `AUTO_INCREMENT` column value to a value larger than the current maximum auto-increment value (in an `UPDATE` operation, for example), the new value is persisted, and subsequent `INSERT` operations allocate auto-increment values starting from the new, larger value.

For more information, see [AUTO_INCREMENT Handling in InnoDB](#), and [InnoDB AUTO_INCREMENT Counter Initialization](#). (Bug #199, Bug #13726455)

- **Replication:** There are two improvements to how a `CHANGE MASTER TO` statement is written into the error log (`mysqld.log`):
 - Before, no commas were put between the option specifications (for example `MASTER_USER =` and `MASTER_PASSWORD =`), so users who wanted to use the statement by copy and paste had to insert the commas manually. Commas are now inserted when the statement is written to the error log.
 - When the literal “<secret>” is inserted as a placeholder for the `MASTER_PASSWORD` value, no quotes are used now, so users who forget to replace the literal with the real password before a copy and paste gets a syntax error immediately, instead of running into other issues.

(Bug #18194384)

- **Replication:** It is now possible to restore a backup of a GTID-based replication server because you can add GTIDs to `gtid_purged`, regardless of whether `gtid_executed` is empty or not. This enables you to restore backups from GTID-based replication servers without losing existing GTID information and binary logs. The GTIDs to add are those which existed in `gtid_executed` at the time of taking the backup. The syntax for `SET GTID_PURGED` has been extended so that `SET GTID_PURGED = "+gtid_set"` adds `gtid_set` to the existing `gtid_purged` GTID set.
- **Replication:** New Performance Schema stages have been added to show the progress of row-based replication. You can use these stages to check the progress of slow operations in row-based replication. Additionally you can find out which database the changes are being applied to. This assists in troubleshooting row-based replication issues and provides more information for performance tuning. For more information see [Monitoring Row-based Replication](#)
- **JSON:** This release adds an unquoting extraction operator `->>`, sometimes also referred to as an inline path operator, for use with `JSON` documents stored in MySQL. The new operator is similar to the `->` operator, but performs `JSON` unquoting of the value as well. For a `JSON` column `mycol` and `JSON` path expression `mypath`, the following three expressions are equivalent:
 - `JSON_UNQUOTE(JSON_EXTRACT(mycol, "$.mypath"))`
 - `JSON_UNQUOTE(mycol->"$.mypath")`
 - `mycol->>"$.mypath"`

The `->>` operator can be used in SQL statements wherever `JSON_UNQUOTE(JSON_EXTRACT())` would be allowed. This includes (but is not limited to) `SELECT` lists, `WHERE` and `HAVING` clauses, and `ORDER BY` and `GROUP BY` clauses.

For more information, see [Functions That Search JSON Values](#), and [JSON Path Syntax](#). (Bug #78736, Bug #21980346)

- To produce more accurate estimates, the `MEMORY` storage engine now calculates index statistics (records per key estimates) using floating-point rather than integer arithmetic. (Bug #23024059)
- A new CMake option, `INSTALL_STATIC_LIBRARIES`, enables control over whether to install static libraries. The default is `ON`. If set to `OFF`, these libraries are not installed: `libmysqlclient.a`, `libmysqld.a`, `libmysqlservices.a`. (Bug #22891432)
- The internal `mysql_prepare_create_table()` server function has been refactored for improved code maintainability and clarity. This code revision results in the following minor changes of behavior for `CREATE TABLE` and `ALTER TABLE`:
 - Attempts to create a second primary key based on a `NULL` column now produce an `ER_MULTIPLE_PRI_KEY` error rather than `ER_PRIMARY_CANT_HAVE_NULL`.
 - Attempts to create a second primary key based on a generated column now produce an `ER_MULTIPLE_PRI_KEY` error rather than `ER_UNSUPPORTED_ACTION_ON_GENERATED_COLUMN`.
 - Attempts to create a full-text key on a `JSON` column now produce an `ER_JSON_USED_AS_KEY` error rather than `ER_BAD_FT_COLUMN`.
 - Attempts to create a key in a storage engine that does not support keys (for example, `EXAMPLE`) now produce an `ER_TOO_MANY_KEYS` error rather than `ER_TOO_MANY_KEY_PARTS`.

(Bug #22884886)

- Previously, the Performance Schema was not built for `libmysqld`, the embedded server. This prevented use of the `SHOW STATUS` and `SHOW VARIABLES` statements with `show_compatibility_56=OFF` because, with that setting, those statements take their results from Performance Schema tables. Now for `libmysqld`, the required Performance Schema tables are built (with no instrumentation collected), so that those `SHOW` statements can be supported with `show_compatibility_56=OFF`. (Bug #22809694)
- Several internal functions used by `JSON_CONTAINS()`, `JSON_SEARCH()`, and other MySQL JSON functions created excessive numbers of local copies of keys, values, or both, when performing inspections of JSON objects. Such copying has been eliminated or reduced in many cases. In addition, the lifetimes of temporary objects used by some of these functions have been reduced. These changes should make these and related JSON functions perform more efficiently than previously, and with fewer resources required. (Bug #22602142)
- If the system `lz4` and `openssl zlib` commands are available, the `lz4_decompress` and `zlib_decompress` utilities are unneeded. Two changes enable those utilities not to be built: If the new `WITH_LZ4` CMake option is set to `system`, `lz4_decompress` is not built or installed. If the `WITH_ZLIB` CMake option is set to `system`, `zlib_decompress` is not built or installed. (Bug #22329851)
- Source files for the MySQL strings library have been converted from C (`.c` suffix) to C++ (`.cc` suffix). This enables stricter compilation checks and use of C++ features in the library code. (Bug #22124719)
- Source code for the `mysys` library now uses C++ rather than C to take advantage of stricter compilation checks and permit use of C++ features. (Bug #21881278)

- A new CMake option, `WITH_TSAN`, permits enabling ThreadSanitizer for compilers that support it. (Bug #80409, Bug #23171902)
- The global list of connections, previously protected by a single mutex, has been partitioned into eight parts, each protected by its own instance of the mutex. The result is a reduction of overhead and improved performance for connection processing. An implication of this change for monitoring purposes is that the Performance Schema now exposes eight different instances each of the `LOCK_thd_list` mutex, `LOCK_thd_remove` mutex, and `COND_thd_list` condition variable.
- MySQL now provides functions to manipulate UUID values and make them easier to work with:
 - `UUID_TO_BIN()` and `BIN_TO_UUID()` convert between UUID values in string and binary formats (represented as hexadecimal characters and `VARBINARY(16)`, respectively). This permits conversion of string UUID values to binary values that take less storage space. UUID values converted to binary can be represented in a way that permits improved indexing efficiency.
 - `IS_UUID()` returns 1 or 0 to indicate whether its argument is a valid string-format UUID value.

For more information about these functions, see [Miscellaneous Functions](#)

- The `mysql_plugin` utility has been removed. Alternatives include loading plugins at server startup using the `--plugin-load` or `--plugin-load-add` option, or at runtime using the `INSTALL PLUGIN` statement.
- The server now relies on storage engines to clean up temporary tables left from previous server runs. `InnoDB` does this by discarding the temporary tablespace on restart. `MyISAM` and other similar storage engines still rely on scanning the temporary directory to detect leftover tables, by looking for files belonging to these engines with a certain name pattern.
- The deprecated `mysql_shutdown()` C API function and corresponding `COM_SHUTDOWN` client/server protocol command have been removed. Instead, use `mysql_query()` to execute a `SHUTDOWN` statement.
- The server no longer performs conversion of pre-MySQL 5.1 database names containing special characters to 5.1 format with the addition of a `#mysql150#` prefix. Because these conversions are no longer performed, the `--fix-db-names` and `--fix-table-names` options for `mysqlcheck`, the `UPGRADE DATA DIRECTORY NAME` clause for the `ALTER DATABASE` statement, and the `Com_alter_db_upgrade` status variable have been removed.

Upgrades are supported only from one major version to another (for example, 5.0 to 5.1, or 5.1 to 5.5), so there should be little remaining need for conversion of older 5.0 database names to current versions of MySQL. As a workaround, upgrade a MySQL 5.0 installation to MySQL 5.1 before upgrading to a more recent release.

Bugs Fixed

- **Incompatible Change:** Concatenation of spatial values makes little sense, so the `CONCAT()` and `CONCAT_WS()` functions now produce an error for spatial arguments. (Bug #22893669)
- **Important Change; JSON:** The empty string value is now accepted as a key when used with `JSON` functions such as `JSON_EXTRACT()`. In such cases, it must be quoted. (Bug #79643, Bug #22366102)
- **NDB Cluster:** Previously, the `mysql.ndb_binlog_index` table was created even if the server was built without `NDB`. Now the table is created only if the server is built with `NDB`. (Bug #22874872)
- **InnoDB; Microsoft Windows:** An unspecified block size resulted in an empty `INFORMATION_SCHEMA.TABLESPACES` table on Windows NTFS with a cluster page size greater than or equal to 8K. (Bug #23598872)

- **InnoDB; Microsoft Windows:** Compilation of [InnoDB](#) with Visual Studio 2015 Update 2 returned warnings. (Bug #23056963)
- **InnoDB:** Unnecessary checks were removed from the `ut_cpuid()` function which is used to fetch information about the CPU. (Bug #24405292)
- **InnoDB:** An asynchronous read operation on a deleted tablespace raised an error. (Bug #24388498)
- **InnoDB:** `dict_col_t` accessors were added to the [InnoDB](#) code. (Bug #24363566)
- **InnoDB:** `dict_col` functions in the [InnoDB](#) code were replaced by accessors. (Bug #24361098)
- **InnoDB:** `dict_index_t` functions in the [InnoDB](#) code were replaced by accessors. (Bug #24361023)
- **InnoDB:** Unnecessary code that checked for and released reserved adaptive hash index search latches was removed. (Bug #24300175)
- **InnoDB:** A system tablespace data file size greater than 4G on a 32-bit operating system could result in an overflow condition. (Bug #23753625)
- **InnoDB:** Internal methods for accessing table object data were added to `dict_table_t`. (Bug #23748128)
- **InnoDB:** The restriction that required the first undo tablespace to use `space_id` 1 was removed to avoid `space_id` conflicts with existing tablespaces during upgrade. The first undo tablespace can now use a `space_id` other than 1. `space_id` values for undo tablespaces are still assigned in a consecutive sequence. (Bug #23517560)
- **InnoDB:** Internal accessor functions for iterating the indexes of a table were replaced with accessor methods. Dead code was removed. (Bug #23336108)
- **InnoDB:** The `mysql.innodb_index_stats` and `mysql.innodb_table_stats` table definitions, which were previously created by an SQL script, are now hard-coded. As a result, the `dict_table_schema_check` function is longer required and was removed. (Bug #23336079)
- **InnoDB:** The `ut_snprint` function was replaced by the C++11 `snprintf` function. (Bug #23329353)
- **InnoDB:** For consistency, instances of `uint` in [InnoDB](#) code were replaced with `space_id_t` and `page_no_t` data types. (Bug #23297169)
- **InnoDB:** Use of `boost::atomic` in [InnoDB](#) code was replaced with `std::atomic`. (Bug #23280649)
- **InnoDB:** MySQL binaries were not built with the NUMA feature. (Bug #23259754)
- **InnoDB:** References to `UNIV_NONINL` and `UNIV_MUST_NOT_INLINE` were removed. The `fut0fut.cc` and `ut0byte.cc` files, which were only necessary when `UNIV_NONINL` was defined, were also removed. (Bug #23150562)
- **InnoDB:** The `mutex_own()` mapping caused warnings when compiling with Clang or newer GCC compilers. (Bug #23090278)
- **InnoDB:** Querying the Performance Schema for [InnoDB](#) memory allocation event data incorrectly reported values of 0. (Bug #23020280)
- **InnoDB:** `DEBUG_OFF` compile-time flags were replaced by `UNIV_DEBUG` flags. To improve error log output, `ut_dbg_assertion_failed()` now uses `sql_print_error()` to display the file name, line number, and message in a single line. The thread ID is displayed in a subsequent line. (Bug #22996442, Bug #23028144)

- **InnoDB:** `SHOW ENGINE INNODB STATUS` output displayed negative spin rounds per wait values. Thanks to Laurynas Biveinis for the patch. (Bug #22844987, Bug #79703)
- **InnoDB:** The `innodb_disable_resize_buffer_pool_debug` option was removed. The patch for this change also removed a code variable and simplified the `buf_pool_resize()` function. (Bug #22755053)
- **InnoDB:** After a successful `ALTER TABLE ... ALGORITHM=COPY` operation, an assertion was raised while building a previous version of a clustered index record. (Bug #22707367)
- **InnoDB:** A DML operation that updated a counter in a table with a virtual index raised on assertion in `row_parse_int()`. (Bug #22650195)
- **InnoDB:** The `InnoDB memcached` plugin would not load when compiled with `libevent 2.0`. (Bug #22646919)
- **InnoDB:** Unused calculations for integer-based `rec_per_key` values were removed from `InnoDB`. Integer-based `rec_per_key` information was replaced by floating point index statistics in an earlier release. (Bug #22625348)
- **InnoDB:** On slow shutdown, purge thread shutdown was initiated before the background rollback thread exited, resulting in an assertion failure. (Bug #22561332)
- **InnoDB:** Blocks were lost in `row_vers_old_has_index_entry()` due to unfreed heaps. (Bug #22543834, Bug #79973)
- **InnoDB:** A transportable tablespace debug test raised an assertion that was due to a race condition. (Bug #22453668)
- **InnoDB:** The server failed to start due to missing undo tablespaces. (Bug #22452992)
- **InnoDB:** `InnoDB` recovery asserted while attempting to close an undo tablespace due to buffered undo tablespace changes introduced by the recovery process. (Bug #22361764)
- **InnoDB:** Reallocation of `memcached`-referenced memory raised an assertion. (Bug #22304250, Bug #79500)
- **InnoDB:** Building `InnoDB` with C++11 returned “register” deprecation warnings. Handling of “register” deprecation warnings remained in the code after the deprecated “register” keyword was removed. Also, an unused declaration of `yyset_extra()` was removed. (Bug #22292704)
- **InnoDB:** `SHOW CREATE TABLE` output for partitioned tables did not accurately display tablespace assignment information for table partitions. (Bug #22245554)
- **InnoDB:** An `ALTER TABLE...TRUNCATE PARTITION` operation ignored the table's `KEY_BLOCK_SIZE` attribute and used the default value instead, which is half of the `innodb_page_size` value. (Bug #22186558, Bug #79223)
- **InnoDB:** Memory leaks in `innochecksum` were corrected. (Bug #22179518)
- **InnoDB:** A `SPACE_ID` column was added to the `INNODB_CACHED_INDEXES` table. The `INDEX_ID` value is no longer a global unique identifier. (Bug #22172026)
- **InnoDB:** A purge thread open table callback for virtual columns raised an assertion due to an unexpected data dictionary table latch. As a temporary workaround, purge is temporarily disabled for virtual generated columns. This temporary workaround may cause b-tree expansion due to unpurged delete-marked records for indexes on virtual columns. (Bug #22153217)
- **InnoDB:** Creating a table with a full-text index and a foreign key constraint failed when `foreign_key_checks` was disabled. (Bug #22094601, Bug #78955)

References: This issue is a regression of: Bug #16845421.

- **InnoDB:** The `ha_innobase::m_primary_key` field was removed. It was redundant. A boolean predicate, `TABLE_SHARE::is_missing_primary_key()`, was added. (Bug #21928734, Bug #78662)
- **InnoDB:** A buffer pool load operation that attempted to load an uninitialized page caused a Valgrind failure. (Bug #21747906)
- **InnoDB:** Unused functions introduced by the `InnoDB memcached` plugin were removed. (Bug #21625760)
- **InnoDB:** An internal global variable used by the `innodb_buffer_pool_size` configuration option was removed. (Bug #21512749)
- **InnoDB:** An `InnoDB` page cleaner thread asserted during a buffer pool resize operation. (Bug #21473497)
- **InnoDB:** Log buffer contention was reduced with the addition of a second buffer, allowing for concurrent log buffer writing and flushing. A new mutex was added to protect log buffer flushing. Thanks to Zhai Weixiang for the patch. (Bug #21352937, Bug #77094)
- **InnoDB:** Unused `InnoDB` and `libsql` functions and variables were removed, and global symbols were converted to static keywords, where possible. (Bug #21153166, Bug #21141390, Bug #77146, Bug #21178589)
- **InnoDB:** The `SysTablespace::parse_units()` function now returns the number of pages in a file instead of the number of megabytes. The `SysTablespace::normalize_size()` function was removed. Error messages in `SysTablespace::parse_params()` were revised. (Bug #21040199, Bug #76949)
- **InnoDB:** For persistent tables, the internal unique identifier for `InnoDB` indexes (`index_id`) now includes a tablespace identifier (`space_id,index_id`). This change makes index identifiers unique at the tablespace level as well as the `InnoDB` instance level, and supports future work related to index identifier allocation. (Bug #20737524, Bug #76392)
- **InnoDB:** Code related to `innochecksum` was cleaned up and reorganized. Checksum functionality is now located in `buf0checksum.cc`. (Bug #20518099)
- **InnoDB:** `__attribute__((nonnull))` was removed from `InnoDB` code. The attribute is no longer permitted by `InnoDB` coding guidelines. (Bug #20468234)
- **InnoDB:** A new struct was added to provide a logical interface for handling and manipulating external `BLOB` field references. (Bug #18195972)
- **InnoDB:** `TRUNCATE TABLE` is now mapped to `DROP TABLE` and `CREATE TABLE`. This change has the following implications:
 - On systems with a large buffer pool and `innodb_adaptive_hash_index` enabled, `TRUNCATE TABLE` operations previously caused a temporary drop in system performance due to an LRU scan that occurred when removing the table's adaptive hash index entries. The remapping of `TRUNCATE TABLE` to `DROP TABLE` and `CREATE TABLE` avoids the problematic LRU scan.
 - `TRUNCATE TABLE` is temporarily non-atomic. A server exit during a `TRUNCATE TABLE` operation can result in a dropped table and orphaned foreign key constraints in the `InnoDB SYS_FOREIGN` and `SYS_FOREIGN_COLS` system tables.

- The InnoDB memcached plugin `flush_all` command invokes `DELETE` instead of `TRUNCATE TABLE`. `DELETE` has a higher overhead cost than `FLUSH TABLES` since it involves undo-logging, delete-marking, and eventually purging each deleted row.
- A log checkpoint that occurred for internal truncate table operations on file-per-table tablespaces was replaced by a log flush.

(Bug #16834993, Bug #68184, Bug #16207919)

- **InnoDB:** InnoDB wasted most pages in extents used for fragment pages. (Bug #16204823, Bug #67963)
- **Partitioning:** In some cases, an issue with partition pruning being attempted a second time during optimization after all partitions had already been pruned at parsing time led to an assert. (Bug #23194259)
- **Partitioning:** A partitioned table whose table name and any partition name had a combined length in excess of 61 characters could not be imported from a backup created using `mysqldump`. When the table also employed subpartitioning, then the combined length of the table name, any partition name, and the name of any subpartition of this partition could not exceed 57 characters without triggering the same issue.

This was due to the fact that the internal `mysql.innodb_table_stats` table allowed a maximum of 64 characters for the column used to store the table name, even though InnoDB stores, for a partitioned or subpartitioned table, a row in `innodb_table_stats` for each partition or subpartition wherein the value actually used to represent the table name follows the pattern `table_name#P#partition_name` or `table_name#P#partition_name#SP#subpartition_name`, respectively. This issue is fixed by changing the definition of the `innodb_table_stats` to accommodate the maximum combined length of these attributes plus `#P#` and `#SP#` (199 characters). (Bug #72061, Bug #18416479)

- **Replication:** In `Slave_worker::write_info()`, `DEBUG_ENTER()` had “`Master_info::write_info`” as its argument instead of “`Slave_worker::write_info`”. This fix corrects the argument. Thanks to Stewart Smith for the patch. (Bug #21658067, Bug #78133)
- **Replication:** When using `START SLAVE UNTIL position` statements with a multi-threaded slave the only `UNTIL` clause available was `SQL_BEFORE_GTIDS`. Now multi-threaded slaves are compatible with the `START SLAVE UNTIL MASTER_LOG_FILE = 'log_name', MASTER_LOG_POS = log_position` and `START SLAVE UNTIL RELAY_LOG_FILE = 'log_name', RELAY_LOG_POS = log_position` statements. (Bug #75843, Bug #20513547)
- **Microsoft Windows:** On Windows, setting the global `log_syslog` system variable in multiple threads could cause a server exit. (Bug #22180046)
- **Solaris:** The client library failed to build on Solaris using the `Cstd` library. (Bug #24353920, Bug #82347)
- **JSON:** `CHECKSUM TABLE` calculated the checksums for `JSON` values using the memory addresses of the values rather than the values themselves, which made the checksum vary. Now in such cases the calculation is based on the actual `JSON` value, and not on that value's address. (Bug #23535703)
- **JSON:** Passing `NULL` to a stored procedure expecting a `JSON` parameter led to an assertion failure in debug builds. (Bug #23209914)
- **JSON:** Parsing of `JSON` path arguments failed to distinguish between a `NULL` path and one that was syntactically invalid.

This has been changed so that parsing of these paths now clearly distinguishes between valid non-NULL paths, NULL paths, and invalid paths. (Bug #22816576)

- **JSON:** For debug builds, an assertion could be raised when the server created a temporary table to hold `JSON` objects. (Bug #22782948)
- **JSON:** Queries that executed a `JSON` function that raised an error could cause a server exit. (Bug #22253965)
- Renaming a table to be part of a nonexistent database failed (correctly), but with an `Unknown error` message. A proper error message is now produced; this was corrected as part of the data dictionary implementation. (Bug #25167507, Bug #84000)
- For segmentation faults on FreeBSD, the server did not generate a stack trace. (Bug #24566529, Bug #23575445, Bug #81827)
- On macOS, stack trace demangling now occurs for builds compiled using Clang, just as for GCC. (Bug #23606094, Bug #81908)
- `libevent` was built on macOS even when not needed. (Bug #23228287, Bug #81311)
- A function that returns a `JSON` value could cause a server exit if called as part of a `CASE` statement in a stored procedure. (Bug #23212765)
- Previously, different values were reported by `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` and `SELECT * FROM performance_schema.memory_summary_global_by_event_name` for total memory used in the Performance Schema. The memory for scalable buffer pages, instrumented as `memory/performance_schema/scalable_buffer`, was missing from the `SHOW ENGINE STATUS` output. That statement now includes the missing memory, displayed as `(pfs_buffer_scalable_container).memory`. (Bug #23104498)
- The `-fexpensive-optimizations` option to GCC caused ARM64 and PowerPC builds to compute floating-point operations slightly differently from other platforms. This option was enabled by `-O2` and higher optimization levels. The option now is disabled on platforms negatively affected by it. (Bug #23046775)
- After a failed administrative operation such as `ALTER TABLE ... OPTIMIZE PARTITION`, selecting from the Performance Schema in lock-tables mode could hang. (Bug #23044286)
- In builds with AddressSanitizer enabled, `CAST(... AS BINARY)` could cause a server exit. (Bug #22900560)
- Some spatial functions were reported using a different spatial function name in error messages. (Bug #22883056, Bug #80627)
- Fixed Valgrind warnings with Clang in optimized mode for the `my_strtod_int()` function. (Bug #22839888)
- Precision math operations on values with 64 decimals could produce a 0 result. (Bug #22828692)
- `ST_GeomFromGeoJSON()` could return an error with valid arguments. (Bug #22804853)
- For debug builds with `STRICT_TRANS_TABLES` SQL mode enabled, an assertion could be raised by `INSERT` or `REPLACE` statements that had made changes that could not be rolled back and that subsequently generated an `ER_NO_DEFAULT_FOR_FIELD` error. (Bug #22635253)
- The optimizer contained a `memcpy()` call that did not check for overlapping source and destination. (Bug #22537196)

- For debug builds, a missing error check on the result of a subquery that accessed a `JSON` value could raise an assertion. (Bug #22522073)
- Preparing a `CREATE TABLE ... SELECT` statement, then flushing tables (thus closing the table) and executing the prepared statement could cause a server exit. (Bug #22393309)
- A prepared statement that used a parameter in the select list of a derived table that was part of a join could cause a server exit. (Bug #22392374, Bug #24380263)
- Some grant tables did not account for the increase in maximum user name length from 16 to 32 characters in MySQL 5.7.8. (Bug #22379607, Bug #79680)
- Re-evaluation of a generated column expression could cause access to previously freed memory and a server exit. (Bug #22346120)
- `HANDLER` read statements that searched an index when the target index value was not stored into the row buffer successfully could cause a server exit. (Bug #22321965)
- Improper handling of numeric-to-`ZEROFILL` conversion for `NULL` values could lead to a server exit. (Bug #22281205)
- Using a subquery containing a row constructor to set a variable in a `SET` statement could cause a server exit. (Bug #22276843)
- If the SQL mode did not include `ALLOW_INVALID_DATES`, a query that contained `invalid_date IN (subquery)` and was handled by subquery materialization could cause a server exit. (Bug #22262843)
- For the embedded server, the code following the check for invalid arguments was invoked with missing or incorrect arguments, which could lead to an improper exit. (Bug #22262706)
- On OS X, `vio_io_wait()` used `select()`, limiting the number of file descriptors to 1024. Now `kqueue` event notification is used instead to avoid this limit. FreeBSD was changed to use `kqueue` as well. (Bug #22244911)
- Memory leaks could result if stored routine loading involved temporarily changing the default database and errors occurred restoring the original default database. (Bug #22179795)
- `CMake` configuration was adjusted to check for `-Wxxx` compiler options instead of `-Wno-xxx` because the latter produce false positives for GCC. (Bug #21881753)
- There could be discrepancies between the values of `INFORMATION_SCHEMA.EVENTS.LAST_EXECUTED` and `mysql.event.last_executed`. This no longer occurs. Event information is stored in the `mysql.events` data dictionary table, which is invisible, so that `INFORMATION_SCHEMA.EVENTS` is the sole interface to event metadata. (Bug #21374010)
- Views could evaluate user-defined or SQL functions before evaluating restrictions from the view definition. (Bug #20933307)
- With `-DENABLE_DTRACE=ON`, `CMake` did not check whether a working DTrace installation was present. Now it checks and aborts if DTrace cannot be found. (Bug #20671056)
- If given a relative path name for the `--log-error` option, `mysqld` could send `stdout` and `stderr` to the wrong location. (Bug #20609063)
- Evaluation of `LEAST()` and `GREATEST()` could use too small a sort buffer for datetime and string literals, causing an assertion to be raised. (Bug #20565160)
- The range of error numbers for errors that are new in MySQL 8.0 has been designated to begin with 3500. (Bug #20538173)

- Debian packaging was updated not to set the `sql_mode` system variable in `my.cnf`. (Bug #20535729)
- Event loading from the `mysql.event` system table could fail if the `PAD_CHAR_TO_FULL_LENGTH` SQL mode was enabled. (Bug #20073523, Bug #74947)
- Statements such as `INSERT` and `LOAD DATA` that use the `REPLACE` or `IGNORE` keyword to handle duplicate records could affect subsequent operations. (Bug #20017428)
- `CREATE TABLE ... SELECT` where non-`BIT` data was selected from the source table into a `BIT` column in the destination table could cause a server exit. (Bug #19930894)
- Compilation failed on OS X when MySQL was configured with `-DMYSQL_MAINTAINER_MODE=1` and compiled with clang/Xcode 6.0. (Bug #19694515, Bug #74100)
- For `CHANGE MASTER TO` statements rewritten to filter the password before being written to the general query log, any `MASTER_AUTO_POSITION` clause was lost. (Bug #19622609)
- Use of the `VALUES()` function in a `SELECT` clause could result in a server exit. (Bug #19601973)
- A potential null-pointer dereference and memory leak in table-rename code were corrected. (Bug #18194270)
- Using `MATCH ... AGAINST` to compare a character column and an aggregate function could cause a server exit. (Bug #17865492)
- Slightly different values for the number of connections could be reported in various information sources, such as the `Connections` status variable, Performance Schema `threads` and `global_status` tables, and `SHOW PROCESSLIST` statement. (Bug #17666696)
- A query with a subquery containing a set operation with an outer reference might cause a server exit. (Bug #17270896)
- Using `GRANT` to change a password for an invalid user produced an error, but also updated the `mysql.user` system table. (Bug #17180985)
- The parser for spatial WKT data accepted numbers such as `0.23` but not `.23`, the equivalent value without the leading zero. Now both formats are accepted. (Bug #17167633)
- Previously, if a client attempted to send connection attribute key/value pairs that in aggregate had a size larger than the value of the `performance_schema_session_connect_attrs_size` system variable, the Performance Schema truncated the attribute data. In addition, the Performance Schema wrote this message to the error log if the `log_warnings` system variable was greater than zero:

```
[Warning] Connection attributes of length N were truncated
```

This message was not helpful to a DBA attempting to determine the problematic client, so several changes have been made to connection attribute handling:

- Truncation of connection attributes still occurs for excessive data, but the log message is more informative. It includes the number of bytes lost, the connection identifier, and information about the client user. The additional information should enable DBAs to more easily identify clients for which attribute truncation occurred.
- When truncation occurs, a `_truncated` attribute is added to the session attributes with a value indicating how many bytes were lost, if the attribute buffer has sufficient space. This enables the Performance Schema to expose per-connection truncation information in the connection attribute tables.

- A new status variable, `Performance_schema_session_connect_attrs_longest_seen`, indicates the longest connection attribute buffer smaller than 64KB seen by the server. If this value is larger than `performance_schema_session_connect_attrs_size`, attribute truncation has occurred, and DBAs may wish to increase the latter value, or, alternatively, investigate which clients are sending large amounts of attribute data.

For more information, see [Performance Schema Connection Attribute Tables](#). (Bug #16576959)

- Introduction of the data dictionary enabled several `INFORMATION_SCHEMA` problems to be addressed:
 - Queries on `INFORMATION_SCHEMA.STATISTICS` could return different results depending on the order of columns in the select list.
 - Some `INFORMATION_SCHEMA` tables had suboptimal column types and sizes. Such tables that are now views on data dictionary tables in the `mysql` system database have more appropriate column definitions.
 - Queries on `INFORMATION_SCHEMA` tables that resulted in directory scans to determine database or file names no longer do so, but instead read database and table names from the data dictionary.
 - Queries on `INFORMATION_SCHEMA` tables that opened `.frm` files to obtain table metadata no longer do so, but instead read this information from the data dictionary.
 - For comparisons of database or table names in `INFORMATION_SCHEMA` queries, using `COLLATE` to force a given collation worked only if applied to the `INFORMATION_SCHEMA` table column, but not if applied to the comparison value.

For additional information about performing such comparisons, see [Using Collation in INFORMATION_SCHEMA Searches](#).

(Bug #14017351, Bug #65121, Bug #17559183, Bug #70462, Bug #23259470, Bug #81347, Bug #20372562, Bug #75532, Bug #13878164, Bug #11756519, Bug #48445)

- For abnormal server exit on Windows, the server previously created a minidump file named `module_name.dmp`, where `module_name` is the name of the server executable file. To prevent earlier minidump files from being overwritten, minidump file names now include the process ID and have the form `module_name.pid.dmp`; for example, `mysqld.exe.7296.dmp`. (Bug #12779463)
- For queries on `INFORMATION_SCHEMA` tables, comparisons of schema and table names could be case sensitive or insensitive, depending on the characteristics of the underlying file system and the `lower_case_table_names` system variable value. Furthermore, it was ineffective to provide a `COLLATE` clause to change the comparison properties because that clause was ignored. This has been changed so that `COLLATE` is no longer ignored and can be used to obtain the desired comparison properties. (Bug #11748044, Bug #34921)
- `FLOOR(CEIL())` truncated large `BIGINT UNSIGNED` arguments. (Bug #80873, Bug #23013359)
- Manipulation of a value returned by the `JSON_MERGE()` function using `JSON_SET()` sometimes produced an invalid result. (Bug #80787, Bug #22961128)
- `ST_AsGeoJSON()` failed when geometry arguments were supplied using user-defined variables. (Bug #80697, Bug #22912800)
- `CAST(expr AS BINARY(N))` unexpectedly returned `NULL` for some valid values of `N`. (Bug #80630, Bug #22885819)

- Geometry import functions that took an SRID parameter cast it to an unsigned 32-bit integer without warning or error, so negative values or values larger than unsigned 32-bit integer range were silently converted to a number within the range. Now, all geometry functions that take the SRID as a parameter check that it is within unsigned 32-bit integer range and produce an `ER_DATA_OUT_OF_RANGE` error if not. This also applies to GeoJSON and GeoHash functions that previously checked that the parameter was within range but returned a different error code. (Bug #80499, Bug #22819614)
- If rounding occurred while storing a predicate value, the range optimizer might not return correct results for the `<` and `<=` operators. (Bug #80244, Bug #22661012)
- For the `mf_iocache` unit test, add a missing `va_end()`, fix a memory leak by calling `my_end()`, and add a target for the test. Thanks to Daniel Black for the patch on which these changes were based. (Bug #80085, Bug #22578670)
- `SELECT DISTINCT SUBSTR()` could incorrectly discard values as duplicates for large position or length arguments. The same issue also affected `LEFT()` and `RIGHT()`. (Bug #80047, Bug #22565155)
- `SUBSTRING_INDEX(str, delim, count)` did not properly handle `count` values larger than 32 bits. (Bug #79978, Bug #22545429)
- For calls to `CONVERT()`, literal string arguments could be modified during execution, producing incorrect results. (Bug #79924, Bug #22531111)
- `REPLACE('a', BINARY 'b', NULL)` returned 'a' rather than `NULL`. (Bug #79912, Bug #22523836)
- Transaction state tracking now avoids a function that is not 8-bit safe, for enhanced compatibility with nonstandard character sets. (Bug #79905, Bug #22523383)
- `GREATEST()` and `LEAST()` treated all integer input as signed. (Bug #79902, Bug #22523685)
- Lines in the general query log were missing a tab between the timestamp and the thread ID. Thanks to Tsubasa Tanaka for the patch. (Bug #79868, Bug #22508563)
- Some comparisons between unsigned values and negative upper limits could return incorrect results (for example, `CAST(100 AS UNSIGNED) BETWEEN 1 AND -1`). (Bug #79857, Bug #22501606)
- The `REPEAT()` function did not properly handle output from the `SUBSTR()` function. (Bug #79695, Bug #22391186)
- The `JSON_TYPE()` function now shows the type of `BIT` literals cast to JSON as `BLOB`, rather than `BIT`. (Bug #79308, Bug #22297987)
- Configuring MySQL with the `-DWITH_UBSAN=ON CMake` option produced a server that was not fully functional. (Bug #79238, Bug #22194071)
- `sql_common.h`, a header file included in MySQL distributions, included and was therefore dependent on `hash.h`, a header file not included in MySQL distributions. This resulted in compilation failures. To eliminate this dependency, `sql_common.h` was modified to no longer include `hash.h`. (Bug #79237, Bug #22187997, Bug #70672, Bug #17633467)
- Timers used for checking maximum statement execution time were initialized even when the server was started with the `--help` option. If `--help` is given, this is no longer done. (Bug #79182, Bug #22172389)
- The optimizer failed when trying to optimize away expressions of the form `IF(true, '2015-01-01', '2015-01-01') IS NOT NULL`. (Bug #79114, Bug #22148586)
- Subtraction of an unsigned decimal could return a negative value, but with metadata type information of `UNSIGNED BINARY`. Subtraction for unsigned decimal subtraction now is handled the same way as

for unsigned integer: Produce an `ER_DATA_OUT_OF_RANGE` error if the result is negative, unless the `NO_UNSIGNED_SUBTRACTION` SQL mode is enabled. (Bug #78914, Bug #22083757)

- Handling by the `HEX()` function of numbers larger than 2^{64} was improved. (Bug #78828, Bug #22297983)

References: This issue is a regression of: Bug #9854.

- The client-side plugin deinitialization function signature was changed from `int (*deinit)()` to `int (*deinit)(void)` to avoid warnings when compiling with `-Wstrict-prototypes`. (Bug #78177, Bug #21680094, Bug #81419, Bug #23282498)
- `CREATE TABLE` reported an incorrect error if a very long or incorrect path name was specified for the `DATA DIRECTORY` or `INDEX DIRECTORY` table option. Now `ER_PATH_LENGTH` or `ER_WRONG_VALUE` are reported for those cases. (Bug #76635, Bug #20857556)
- The server now tries to provide more informative messages for these error codes: `ER_CANT_CREATE_DB`, `ER_CANT_CREATE_TABLE`, `ER_DB_DROP_DELETE`, `ER_DB_DROP_RMDIR`, `ER_ERROR_DURING_COMMIT`, `ER_ERROR_DURING_ROLLBACK`, `ER_GET_ERRNO`. (Bug #76298, Bug #20694494)
- For some instances of failure to prepare an XA transaction, incomplete transaction cleanup could raise an assertion. (Bug #75809, Bug #20488921)
- `mysqld` could attempt to close an invalid socket file descriptor. Thanks to Zhai Weixiang for the patch. (Bug #75778, Bug #20504513)
- A statement of the following form converted the table data to `latin1`, but also changed the table default character set to `latin1` and ignored the `utf8` clause:

```
ALTER TABLE tbl_name CHARACTER SET utf8, CONVERT TO CHARACTER SET latin1;
```

Thanks to Daniel Black for the patch. (Bug #75320, Bug #20279241)

- In `mysqld.cc`, the `abort_loop` variable was quantified with `volatile`, which on some platforms could result in changes not being seen immediately in threads running on different cores. Thanks to Stewart Smith for the patch. (Bug #74846, Bug #20134637)
- Calling a procedure which created a view from a trigger, or creating a function that called a procedure that executed `RENAME TABLE` could, under certain circumstances, raise an assertion. (Bug #74740, Bug #19988193, Bug #21198646)
- Timestamps for server-side prepared statements could be written to the binary log up to a second behind timestamps for the corresponding nonprepared statements, leading to time value differences between master and slave servers. (Bug #74550, Bug #19894382, Bug #25187670)
- For dynamic storage engine plugins, `DROP TABLE`, `TRUNCATE TABLE`, and `RENAME TABLE` did not work due to incorrectly determining the engine from the `.frm` file. (Bug #74277, Bug #19902868)
- Executed prepared statements are logged with `?` parameter markers replaced by data values. Construction of the logged string was inefficient and has been improved. (Bug #73056, Bug #20955496)
- Assignment by a plugin to its thread variables of string type could leak memory. (Bug #71759, Bug #19917521)
- Grouping with a view could produce an `ER_INVALID_GROUP_FUNC_USE` error ("Invalid use of group function") when selecting from the base table did not. (Bug #70220, Bug #17406425)

- Test cases that were intended to be storage engine-agnostic but were actually using a specific engine were corrected.

Index

Symbols

--basedir, 24
--binary-as-hex, 24
--column-statistics, 24
--comments, 24
--daemonize, 24
--des-key-file, 3
--ignore-builtin-innodb, 3
--initialize, 24
--log-warnings, 3
--network-timeout, 48
--secure-auth, 3
--skip-comments, 24
--skip-grant-tables, 3
--skip-networking, 3
--ssl, 90
--ssl-mode, 90
--ssl-verify-server-cert, 90
--symbolic-links, 24
--temp-pool, 48
--transaction-isolation, 3
--transaction-read_only, 3
1F, 3
32-bit, 90
\u001f, 3

A

account management, 24, 90
adaptive hash index, 48, 90
aggregation, 48
ALGORITHM=COPY, 90
Alpine Linux, 24
ALTER TABLE, 3, 24, 48, 90
ANALYZE TABLE, 24
ASCII 31, 3
atomic DDL, 3, 48, 90
atomic operations, 90
audit_log plugin, 48
authentication plugins, 3
auto-increment, 3, 24
autopositioning, 24
autoscale, 3
autowrapping, 3
AUTO_INCREMENT, 90

B

backup, 3
back_log, 3

- Barracuda, 90
- basedir, 24
- binary log, 3, 24, 48, 90
- binlog_expire_logs_seconds, 48
- binlog_format, 24
- binlog_row_value_options, 3
- binlog_transaction_dependency_history_size, 48
- binlog_transaction_dependency_tracking, 48
- BIN_TO_UUID(), 90
- BIT, 90
- bit operations, 90
- BLOB, 90
- bool, 48
- Boost, 24, 48, 90
- bootstrap, 24
- buffer pool, 3, 48, 90
- buffer pool mutex, 90
- buffer pool scan, 90

C

- C API, 3, 24, 48, 90
- C++11, 90
- caching_sha2_password, 3
- CASE, 48
- CAST(), 90
- CEIL(), 90
- change buffer, 90
- CHANGE REPLICATION FILTER, 24
- channel initialization, 24
- character escapes, 3
- character sets, 3, 24, 48, 90
- character_set_database, 48
- character_set_server, 48
- checksum, 90
- CHECKSUM TABLE, 90
- check_stack_overrun(), 48
- Clang, 24, 90
- COLLATE, 3
- collations, 3, 24, 48, 90
- collation_database, 48
- collation_server, 48
- columns_priv table, 90
- column_statistics table, 3, 24
- COLUMN_STATISTICS table, 24
- column_stats table, 24
- comments, 24
- common table expressions, 3, 48
- compiling, 3, 24, 48, 90
- components, 24, 48, 90
- component_sys_variable_register service, 24
- component_sys_variable_unregister service, 24
- compression, 3, 48, 90
- COM_SHUTDOWN, 90

- configuration, 3, 24, 48, 90
- connection handling, 90
- CONNECTION_CONTROL plugin, 48
- CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS plugin, 48
- CONVERT(), 90
- corruption, 90
- cost model, 48
- cost tables, 48
- COUNT(), 24
- crc32, 90
- CREATE FUNCTION, 24
- CREATE INDEX, 48
- CREATE MASTER, 90
- CREATE TABLE, 3, 24, 48, 90
- CREATE TABLESPACE, 24
- CREATE TEMPORARY TABLE, 90
- CREATE USER, 3
- cte_max_recursion_depth, 3

D

- data dictionary, 3, 24, 48, 90
- data dictionary cache, 24
- data file, 90
- data types, 3, 90
- datetime_format, 3
- date_format, 3
- db table, 90
- DEBUG_ABORT(), 24
- DDL, 3, 48
- dd_properties table, 24
- deadlock, 90
- debug, 90
- Debug Sync, 48
- DECODE(), 3
- derived tables, 48
- descending indexes, 48
- DES_DECRYPT(), 3
- DES_ENCRYPT(), 3
- dictionary object cache, 90
- DISTINCT, 24
- DML, 90
- Docker, 3
- DOM, 48
- double, 48
- doxygen, 90
- DROP DATABASE, 3, 48
- DROP DATAFILE, 24
- DROP FUNCTION, 24
- DROP INDEX, 48
- DROP TABLE, 3, 24, 48
- DROP TABLESPACE, 24
- DROP VIEW, 3
- DTrace, 48

duplicate keys, 3

E

embedded server, 48
empty string, 90
ENCODE(), 3
ENCRYPT(), 3
encryption, 3, 24, 48
ENUM, 24
error logging, 24
errors, 3, 24, 48, 90
ER_INCONSISTENT_ERROR, 24
Event Scheduler, 24
events table, 90
events_statements_summary_by_digest, 3
event_scheduler, 3
EXPLAIN EXTENDED, 3
EXPLAIN PARTITIONS, 3
explicit_defaults_for_timestamp, 24
EXTENDED, 24
extent, 90

F

FEDERATED, 48
FLOOR(), 90
FLUSH DES_KEY_FILE, 3
FLUSH LOGS, 24
flush method, 90
FLUSH QUERY CACHE, 3
FLUSH TABLES, 3, 24
flushing, 3
FORCE INDEX, 3
foreign keys, 3, 24, 48, 90
formatting, 24
fragment page, 90
FreeBSD, 48
full-text, 90
full-text search, 3, 24, 48
FULLTEXT index, 48
FULLTEXT indexes, 24
func table, 90
functions, 48
futex, 48

G

gap lock, 48
general tablespace, 3
generated columns, 24, 48, 90
geometry, 48
GIS, 3, 24, 48, 90
glibc, 48
GREATEST(), 90
Group, 3

group, 3, 24, 48
group commit, 48, 90
group replication, 24
GROUPING(), 48
GROUP_CONCAT(), 24, 48
GTID, 24, 48, 90
GTID skip, 24
gtid_mode, 3, 24

H

have_crypt, 3
have_query_cache, 3
have_symlink, 24
heap size, 24
HEX(), 90
histogram statistics, 3, 24
histogram_generation_max_mem_size, 24

I

ibd2sdi, 3, 24, 90
ibttmp1, 90
ignore_builtin_innodb, 3
IGNORE_SPACE, 24
import, 90
IMPORT TABLE, 48
Important Change, 48, 90
in-place update, 24
Incident event, 24
Incompatible Change, 3, 24, 48, 90
index, 48
index dives, 3
index statistics, 90
index_id, 90
INDEX_MERGE, 48
INFORMATION_SCEHMA, 24
INFORMATION_SCHEMA, 3, 24, 48, 90
INFORMATION_SCHEMA.COLUMNS, 3
INFORMATION_SCHEMA.FILES, 48
innochecksum, 90
InnoDB, 3, 24, 48, 90
INNODB_CACHED_INDEXES, 90
innodb_file_format, 90
innodb_file_format_check, 90
innodb_file_format_max, 90
innodb_large_prefix, 90
innodb_numa_interleave, 48
INSERT, 48
INSTALL COMPONENT, 90
installing, 24, 48
INSTALL_STATIC_LIBRARIES, 90
internal temporary table, 48
isl file, 90
IS_UUID(), 90

J

Japanese, 24, 48
JSON, 3, 24, 48, 90
Json_array, 3, 24
JSON_ARRAYAGG(), 48
JSON_ARRAY_INSERT(), 3
JSON_CONTAINS(), 90
JSON_CONTAINS_PATH(), 48
Json_dom, 3
JSON_EXTRACT(), 3, 90
JSON_INSERT(), 3
JSON_MERGE(), 3, 90
JSON_MERGE_PATCH(), 3
JSON_MERGE_PRESERVE(), 3
Json_object, 3
Json_object::consume(), 90
JSON_OBJECTAGG(), 48
JSON_PRETTY(), 48
JSON_QUOTE(), 48
JSON_REMOVE(), 24
JSON_REPLACE(), 24
JSON_SEARCH, 48
JSON_SEARCH(), 90
JSON_SET(), 24, 90
JSON_STORAGE_FREE(), 24
JSON_STORAGE_SIZE(), 24
JSON_TYPE(), 90
JSON_UNQUOTE, 48
JSON_UNQUOTE(), 48, 90

K

kana sensitivity, 24
keyring_file plugin, 48
keyring_okv plugin, 48
KEY_BLOCK_SIZE, 24

L

last (keyword), 24
last_committed, 24
latch, 24
latch debugging, 48
latin1, 24, 48
latin1_swedish_ci, 48
lc_messages, 24
lc_messages_dir, 24
LEAST(), 90
LEFT(), 90
libevent, 90
libmysqld, 48, 90
Linux, 24
LIST, 3
literal values, 90
LOAD DATA, 24, 48

LOAD XML INFILE, 24
LOB, 48
local_infile, 24
LOCATE(), 48
locking, 3, 24, 48, 90
locking read, 48
locks, 24
log buffer, 90
log rotation, 48
logging, 3, 24, 48, 90
log_error_services, 3, 24
log_error_verbosity, 3
log_warnings, 3
lz4_decompress, 90

M

max_allowed_packet, 3
max_connections, 3
max_error_count, 3
max_length_for_sort_data, 48
max_tmp_tables, 3
MBRContains(), 3, 24
MBRCoveredBy(), 3, 24
MBRCovers(), 3, 24
MBRDisjoint(), 3, 24
MBREquals(), 3, 24
MBRIntersects(), 3, 24
MBROverlaps(), 3, 24
MBRTouches(), 3, 24
MBRWithin(), 3, 24
memcached, 3, 24, 48, 90
memory, 24, 48
MERGE, 90
metadata, 24, 90
Microsoft Windows, 24, 48, 90
MTS, 3, 24, 48
multi-source, 24
multi-threaded slave, 3, 24, 48
multi_range_count, 3
mutex, 90
MVCC, 3
mysql-test-run.pl, 24, 48, 90
mysql-test.run.pl, 48
mysql.gtid_executed, 24
mysql.innodb_table_stats.table_name, 90
mysql.session account, 24
mysqlbinlog, 3, 24
mysqldump, 3, 24, 48, 90
mysqld_safe, 48
mysqld_safe.pid, 48
mysqlpump, 3, 24, 48
mysqltest, 24, 48, 90
mysqlxtest, 24

mysql_config, 48
 mysql_config_editor, 48
 mysql_insert_id(), 48
 mysql_install_db, 90
 mysql_options(), 24
 MYSQL_OPT_SSL_MODE, 24
 mysql_plugin, 90
 mysql_secure_installation, 48
 mysql_service_udf_registration service, 24
 mysql_service_udf_registration_aggregate service, 24
 mysql_shutdown(), 48, 90
 mysql_stmt_close(), 24
 mysql_stmt_errno(), 24
 mysql_stmt_error(), 24
 mysql_stmt_sqlstate(), 24
 mysql_string component, 24
 mysql_upgrade, 24
 mysys, 90
 my_bool, 48
 my_init(), 24

N

NDB Cluster, 90
 ndb_binlog_index, 90
 ndb_cache_check_time, 3
 ngram, 3
 NO PAD collations, 24, 48
 NO WAIT, 48
 NO_INDEX_MERGE, 48
 NO_MERGE, 90
 NULL, 3, 48, 90
 NULLIF(), 24
 NUMA, 24, 90

O

ON DUPLICATE KEY UPDATE, 48
 one_or_all, 48
 ONLY_FULL_GROUP_BY, 48
 OpenSSL, 24, 48
 operators, 90
 OPTIMIZE PARTITION, 24
 optimizer, 3, 24, 48, 90
 optimizer hints, 3, 48, 90
 optimizer_switch, 3
 options, 3, 24, 48, 90
 ORDER BY, 48
 original_commit_timestamp, 24

P

packaging, 3, 24, 48, 90
 Packaging, 48
 PAD SPACE collations, 48
 page, 90

- page cleaner, 90
- page cleaner thread, 48
- page compression, 3, 24
- parallelization, 48
- parser, 3, 24, 48, 90
- partial update, 24
- partial updates, 3
- partition, 3, 90
- partition ID, 3
- Partitioning, 3, 24, 48, 90
- partitioning, 3, 24, 48
- password history, 3
- passwords, 3
- path expressions, 3
- paths, 90
- Performance, 3, 24
- performance, 90
- Performance Schema, 3, 24, 48, 90
- performance_schema_max_mutex_classes, 3
- PERSIST_RO_VARIABLES_ADMIN, 24
- pkg-config, 48
- pluggable authentication, 3, 24
- plugin, 24
- plugin service, 3
- plugins, 3, 24, 48
- plugin_registry_service, 48
- pointers, 3
- prepared statements, 24, 90
- privileges, 3, 24, 48, 90
- proc table, 90
- procedure, 90
- PROCEDURE ANALYSE(), 48
- proxies_priv table, 90
- PROXY, 24
- pruning, 90
- purge, 90

Q

- Qcache_free_blocks, 3
- Qcache_free_memory, 3
- Qcache_hits, 3
- Qcache_inserts, 3
- Qcache_lowmem_prunes, 3
- Qcache_not_cached, 3
- Qcache_queries_in_cache, 3
- Qcache_total_blocks, 3
- query cache, 3, 24, 48
- query rewrite plugins, 24
- query_cache_limit, 3
- query_cache_min_res_unit, 3
- query_cache_size, 3, 48
- query_cache_type, 3
- query_cache_wlock_invalidate, 3

QUOTE(), 48
quoting, 3

R

R-tree, 48
race condition, 90
ranges, 24
RapidJSON, 48
rapidjson, 48
read-ahead, 48
read_only, 24
receiver thread, 24
recovery, 3, 24, 48, 90
redo log, 24, 48
register, 90
relay log, 24, 48
RENAME COLUMN, 3
RENAME TABLE, 24, 48
REPEAT(), 90
REPEATABLE-READ, 24
REPLACE(), 3, 48, 90
replace_numeric_round, 24
Replication, 3, 24, 48, 90
replication, 3, 90
replication_applier_status_by_worker, 3, 24
reserved user accounts, 24
reserved words, 48, 90
RESET PERSIST, 48
RESET QUERY CACHE, 3
resource groups, 3
resource usage, 90
restart, 24
RFC 7159, 3
RIGHT(), 90
roles, 3, 24, 48, 90
rollback, 90
rollback segment, 48
rollback segments, 24
ROLLUP, 48
Rotate_log_event, 24
ROW FORMAT, 24
row format, 90
row-based, 48
ROW_FORMAT, 48
row_table_add_foreign_constraints(), 3
RPM, 24
Russian, 3

S

SAFE_MUTEX, 3
scope, 3
SDI, 3
SDI file, 24

secure_auth, 3
secure_file_priv, 48
security, 24
Security Fix, 24, 48
SELECT, 48
SET, 24
SET GLOBAL, 24
SET PERSIST, 3, 24, 90
SET PERSIST_ONLY, 24
setup_instruments, 3
setup_threads, 3
SET_VAR, 3
sha256_password, 3
SHOW COLUMNS, 3, 24
SHOW CREATE TABLE, 24, 90
SHOW CREATE USER, 3
SHOW CREATE VIEW, 3
SHOW ENGINE INNODB STATUS, 90
SHOW INDEX, 24
SHOW RELAYLOG EVENTS, 24
SHOW SLAVE STATUS, 24
SHOW TABLES, 24
shutdown, 24
SHUTDOWN, 90
SKIP LOCKED, 48
skip-slave-start, 24
slave, 90
slave applier, 24
slave_pending_jobs_size_max, 24
slave_skip_errors, 24
slow shutdown, 90
Solaris, 24, 48, 90
sort, 48
space_id, 90
spatial index, 3
SQL, 24
SQL syntax, 90
sql_buffer_result, 3
SQL_CACHE, 3, 90
sql_log_bin, 3
sql_mode, 24, 48
SQL_NO_CACHE, 3, 90
SRID attribute, 3
SSL, 24, 48
stack size, 48
startup, 24, 90
statistics, 48, 90
status_variable_registration service, 24
std::vector, 24
storage engines, 90
stored programs, 3, 24, 90
ST_AsBinary(), 48
ST_AsGeoJSON(), 90
ST_AsText(), 48

ST_AsWKB(), 48
ST_AsWKT(), 48
ST_Contains(), 3, 24
ST_Crosses(), 3, 24
ST_Disjoint(), 3, 24
ST_Equals(), 3, 24
ST_GeomCollFromText(), 48
ST_GeomCollFromTxt(), 48
ST_GeomCollFromWKB(), 48
ST_GeometryCollectionFromText(), 48
ST_GeometryCollectionFromWKB(), 48
ST_GeometryFromText(), 48
ST_GeometryFromWKB(), 48
ST_GeomFromGeoJSON(), 90
ST_GeomFromText(), 48
ST_GeomFromWKB(), 48
ST_Intersects(), 3, 24
ST_LineFromText(), 48
ST_LineFromWKB(), 48
ST_LinestringFromText(), 48
ST_LinestringFromWKB(), 48
ST_MLineFromText(), 48
ST_MLineFromWKB(), 48
ST_MPointFromText(), 48
ST_MPointFromWKB(), 48
ST_MPolyFromText(), 48
ST_MPolyFromWKB(), 48
ST_MultiLinestringFromText(), 48
ST_MultiLinestringFromWKB(), 48
ST_MultiPointFromText(), 48
ST_MultiPointFromWKB(), 48
ST_MultiPolygonFromText(), 48
ST_MultiPolygonFromWKB(), 48
ST_Overlaps(), 3, 24
ST_PointFromText(), 48
ST_PointFromWKB(), 48
ST_PolyFromText(), 48
ST_PolyFromWKB(), 48
ST_PolygonFromText(), 48
ST_PolygonFromWKB(), 48
ST_SRID(), 90
ST_SwapXY(), 48
ST_Touches(), 3, 24
ST_Within(), 3, 24
ST_X(), 90
ST_Y(), 90
SUBSTR(), 90
SUBSTRING_INDEX(), 90
SUM(), 48
SUPER, 3

T

table statistics, 3

- tablespace, 3, 48, 90
- tablespace encryption, 24, 48
- tablespaces, 24
- tables_priv table, 90
- temporary table, 24, 90
- temporary tables, 24, 48
- TempTable, 3
- test suite, 3, 24, 48, 90
- thread management, 90
- TIMESTAMP, 24
- time_format, 3
- TIME_TRUNCATE_FRACTIONAL, 48
- to (keyword), 24
- transaction, 3
- transaction locks, 3
- transactions, 3, 24, 48
- transaction_isolation, 3
- transaction_read_only, 3
- transportable tablespace, 24, 90
- triggers, 24, 48
- TRUNCATE PARTITION, 90
- TRUNCATE TABLE, 24, 48, 90
- truncation, 48
- tuning, 3
- two-phase commit, 90
- tx_isolation, 3
- tx_read_only, 3
- type conversion, 90

U

- UBSan, 24
- UBSAN, 48
- UDFs, 3
- undo, 24
- undo log, 48
- undo tablespace, 3, 48, 90
- Unicode, 3, 24, 48, 90
- UNINSTALL COMPONENT, 90
- UNION, 48, 90
- UNIV_INNOCHECKSUM, 90
- UPDATE, 48
- upgrades, 3, 24, 48
- usability, 48
- user table, 90
- user_defined_functions table, 24
- use_invisible_indexes, 3
- UTF-16, 48
- utf8mb4, 3, 24, 48
- utf8mb4_0900_ai_ci, 48, 90
- utf8mb4_0900_as_ci, 24
- utf8mb4_general_ci, 48
- utf8mb4_ja_0900_as_cs_ks, 24
- utf8mb4_ru_0900_ai_ci, 3

utf8mb4_ru_0900_as_cs, 3
UUID(), 3
UUIDs, 90
UUID_TO_BIN(), 90

V

Valgrind, 3, 90
validate_password plugin, 90
variables_info, 3
version table, 24
version_tokens plugin, 48
views, 48, 90
virtual column, 24
virtual columns, 3, 24, 48
virtual generated column, 90
virtual indexes, 48, 90
Visual C++ Redistributable, 48

W

WEIGHT_STRING(), 48
whitespace, 24
wildcards, 3
window functions, 3, 24
WITH, 48

X

X Protocol, 3, 48, 90
XA, 24, 48, 90
XA RECOVER, 24
XA transactions, 24, 48, 90
XA_RECOVER_ADMIN, 24
XML, 48
XPath syntax, 24

Y

yaSSL, 3, 48

Z

zlib, 3
zlib_decompress, 90