
MySQL 8.0 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.2. 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: 2017-05-22 (revision: 11711)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL 8.0.2 (Not yet released, Development Milestone)	2
Changes in MySQL 8.0.1 (2017-04-10, Development Milestone)	3
Changes in MySQL 8.0.0 (2016-09-12, Development Milestone)	43
Index	74

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.2.

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.

Changes in MySQL 8.0.2 (Not yet released, Development Milestone)



Note

This is a milestone release, for use at your own risk. 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 8.0.2 has no changelog entries, or they have not been published because the product version has not been released.

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



Note

This is a milestone release, for use at your own risk. 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](#)
- [DTrace Support](#)
- [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. (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 1 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>
Icelandic	<code>utf8mb4_is_0900_as_cs</code>
Japanese	<code>utf8mb4_ja_0900_as_cs</code>
Latvian	<code>utf8mb4_lv_0900_as_cs</code>
Lithuanian	<code>utf8mb4_lt_0900_as_cs</code>
Polish	<code>utf8mb4_pl_0900_as_cs</code>
Classical Latin	<code>utf8mb4_la_0900_as_cs</code>
Romanian	<code>utf8mb4_ro_0900_as_cs</code>
Slovak	<code>utf8mb4_sk_0900_as_cs</code>
Slovenian	<code>utf8mb4_sl_0900_as_cs</code>
Modern Spanish	<code>utf8mb4_es_0900_as_cs</code>
Traditional Spanish	<code>utf8mb4_es_trad_0900_as_cs</code>
Swedish	<code>utf8mb4_sv_0900_as_cs</code>
Turkish	<code>utf8mb4_tr_0900_as_cs</code>
Vietnamese	<code>utf8mb4_vi_0900_as_cs</code>

`utf8mb4_0900_as_cs` also works as an accent-sensitive, case-sensitive collation for the languages in the following table.

Table 2 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

Language Name	Language Code
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 3 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 `mysql.h` header file now requires a C++ or C99 compiler to compile.
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. (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)
- 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.
- The `--temp-pool` server option has been removed.

Data Dictionary Notes

- Crash safety of `DROP DATABASE` and multiple-table `DROP TABLE` statements is improved such that, in common scenarios, the discrepancy between storage engines, the data dictionary, and the binary log is now limited to one table at most. For more information, see [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.

DTrace Support

- 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` is now 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` is now 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 deprecated `PROCEDURE ANALYSE()` syntax has been removed.
- 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 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 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 will return different results, compared to previous MySQL versions.

- 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 9.0. The table contents are 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](#).
- 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_GeomCollFromTtxt()`
 - `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](#).

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)
- `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)

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

- **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` is now 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)
- **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 SQL transaction `SAVEPOINT`.
- **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 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_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)
- 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` is now 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 `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 `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.
 - 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)
- **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 will made Group Replication not stop throttling when the cause of 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:** As 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 LOG` 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 #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 secure 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_INNOODB_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 of 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)

- MySQL did not compile with GCC 7. (Bug #25643811)
- 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)
- 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)
- 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)
- 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)
- 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 `/etc/sysconfig/mysqld`, the `MYSQLD_OPTS` value can be set in that file 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. (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)
- 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)
- 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). (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. 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.)

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` is now 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 4 utf8mb4 UCA 9.0.0 Language-Specific Collations

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

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

Table 5 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
Zulu	zu

`utf8mb4_da_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_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)
- 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.



Note

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

- 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.
- 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 new `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 metadata in serialized form that describes objects such as tables, columns, and indexes. Along with the disappearance of `.frm` and trigger metadata files, mentioned previously, you might notice the appearance of `.SDI` files. These are serialized data 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 [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.
- 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;
```

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. (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](#).
- 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 has a `--test-progress` option to cause display of the percentage of tests remaining. (Bug #25601131, Bug #20755059, Bug #76455)
- `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 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](#).

- **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.

- **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 `innchecksum` 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 `innchecksum` 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 TABLE` 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 sent `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..piddmp`; 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

--network-timeout, 3
--ssl, 43
--ssl-mode, 43
--ssl-verify-server-cert, 43
32-bit, 43

A

adaptive hash index, 3, 43
aggregation, 3
ALGORITHM=COPY, 43
ALTER TABLE, 3, 43
atomic operations, 43
audit_log plugin, 3
AUTO_INCREMENT, 43

B

Barracuda, 43
binary log, 3, 43
binlog_expire_logs_seconds, 3
binlog_transaction_dependency_size, 3
binlog_transaction_dependency_tracking, 3
BIN_TO_UUID(), 43
BIT, 43
bit operations, 43
BLOB, 43
bool, 3
Boost, 3, 43
buffer pool, 3, 43
buffer pool mutex, 43
buffer pool scan, 43

C

C API, 3, 43
C++11, 43
CASE, 3
CAST(), 43
CEIL(), 43
change buffer, 43

- character sets, 3, 43
- character_set_database, 3
- character_set_server, 3
- checksum, 43
- CHECKSUM TABLE, 43
- check_stack_overrun(), 3
- Clang, 43
- collations, 3, 43
- collation_database, 3
- collation_server, 3
- columns_priv table, 43
- common table expressions, 3
- compiling, 3, 43
- components, 3, 43
- compression, 3, 43
- COM_SHUTDOWN, 43
- configuration, 3, 43
- connection handling, 43
- CONNECTION_CONTROL plugin, 3
- CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS plugin, 3
- CONVERT(), 43
- corruption, 43
- cost model, 3
- cost tables, 3
- crc32, 43
- CREATE MASTER, 43
- CREATE TABLE, 3, 43
- CREATE TEMPORARY TABLE, 43

D

- data dictionary, 3, 43
- data file, 43
- data types, 43
- db table, 43
- deadlock, 43
- debug, 43
- Debug Sync, 3
- derived tables, 3
- descending indexes, 3
- dictionary object cache, 43
- DML, 43
- DOM, 3
- double, 3
- doxygen, 43
- DROP DATABASE, 3
- DROP TABLE, 3
- DTrace, 3

E

- embedded server, 3
- empty string, 43
- encryption, 3
- errors, 3, 43

events table, 43
extent, 43

F

FEDERATED, 3
FLOOR(), 43
flush method, 43
foreign keys, 3, 43
fragment page, 43
freebsd, 3
full-text, 43
full-text search, 3
FULLTEXT index, 3
func table, 43
functions, 3
futex, 3

G

gap lock, 3
generated columns, 3, 43
geometry, 3
GIS, 3, 43
glibc, 3
GREATEST(), 43
group, 3
group commit, 3, 43
GROUPING(), 3
GROUP_CONCAT(), 3
GTID, 3
gtid, 43

H

HEX(), 43

I

ibd2sdi, 43
ibttmp1, 43
import, 43
IMPORT TABLE, 3
Important Change, 3, 43
Incompatible Change, 3, 43
index, 3
index statistics, 43
index_id, 43
INFORMATION_SCHEMA, 3, 43
INFORMATION_SCHEMA.FILES, 3
innochecksum, 43
InnoDB, 3, 43
INNODB_CACHED_INDEXES, 43
innodb_file_format, 43
innodb_file_format_check, 43
innodb_file_format_max, 43
innodb_large_prefix, 43

innodb_numa_interleave, 3
INSTALL COMPONENT, 43
installing, 3
INSTALL_STATIC_LIBRARIES, 43
internal temporary table, 3
isl, 43
isl file, 43
IS_UUID(), 43

J

Japanese, 3
JSON, 3, 43
JSON_ARRAYAGG(), 3
JSON_CONTAINS(), 43
JSON_CONTAINS_PATH(), 3
JSON_EXTRACT(), 43
JSON_MERGE(), 43
Json_object::consume(), 43
JSON_OBJECTAGG(), 3
JSON_PRETTY(), 3
JSON_QUOTE(), 3
JSON_SEARCH, 3
JSON_SEARCH(), 43
JSON_SET(), 43
JSON_TYPE(), 43
JSON_UNQUOTE, 3
JSON_UNQUOTE(), 3, 43

K

keyring_file plugin, 3
keyring_okv plugin, 3

L

latch debugging, 3
latin1, 3
latin1_swedish_ci, 3
LEAST(), 43
LEFT(), 43
libevent, 43
libmysqld, 3, 43
literal values, 43
LOAD DATA, 3
LOB, 3
LOCATE(), 3
locking, 3, 43
locking read, 3
log buffer, 43
logging, 3, 43
logrotate, 3
lz4_decompress, 43

M

memcached, 3, 43

- memory, 3
- metadata, 43
- Microsoft Windows, 3, 43
- MTS, 3
- multi-threaded slave, 3
- mutex, 43
- mysql-test-run.pl, 3, 43
- mysql-test.run.pl, 3
- mysql.innodb_table_stats.table_name, 43
- mysqldump, 3, 43
- mysqld_safe, 3
- mysqld_safe.pid, 3
- mysqlpump, 3
- mysqltest, 3, 43
- mysql_config, 3
- mysql_config_editor, 3
- mysql_insert_id(), 3
- mysql_install_db, 43
- mysql_plugin, 43
- mysql_secure_installation, 3
- mysql_shutdown(), 3, 43
- mysys, 43
- my_bool, 3

N

- NDB Cluster, 43
- ndb_binlog_index, 43
- NO PAD collations, 3
- NO WAIT, 3
- NULL, 3, 43
- NUMA, 43

O

- one_or_all, 3
- ONLY_FULL_GROUP_BY, 3
- OpenSSL, 3
- operators, 43
- optimizer, 3, 43
- optimizer hints, 3, 43
- options, 3, 43
- ORDER BY, 3

P

- Packaging, 3
- packaging, 3, 43
- PAD SPACE collations, 3
- page, 43
- page cleaner, 43
- page cleaner thread, 3
- parallelization, 3
- parser, 3, 43
- partition, 43
- Partitioning, 3, 43

- partitioning, 3
- paths, 43
- performance, 43
- Performance Schema, 3, 43
- pkg-config, 3
- plugins, 3
- plugin_registry_service, 3
- prepared statements, 43
- privileges, 3, 43
- proc table, 43
- procedure, 43
- PROCEDURE ANALYSE(), 3
- proxies_priv table, 43
- pruning, 43
- purge, 43

Q

- query cache, 3
- query_cache_size, 3
- QUOTE(), 3

R

- r-tree, 3
- race condition, 43
- RapidJSON, 3
- rapidjson, 3
- read-ahead, 3
- recovery, 3, 43
- redo log, 3
- register, 43
- relay log, 3
- RENAME TABLE, 3
- REPEAT(), 43
- REPLACE(), 3, 43
- Replication, 3, 43
- replication, 43
- reserved words, 3, 43
- RESET PERSIST, 3
- resource usage, 43
- RIGHT(), 43
- roles, 3, 43
- rollback, 43
- rollback segment, 3
- ROLLUP, 3
- row format, 43
- row-based, 3
- ROW_FORMAT, 3

S

- secure_file_priv, 3
- Security Fix, 3
- SELECT, 3
- SET PERSIST, 43

SHOW CREATE TABLE, 43
SHOW ENGINE INNODB STATUS, 43
SHUTDOWN, 43
SKIP LOCKED, 3
slave, 43
slow shutdown, 43
Solaris, 3, 43
sort, 3
space_id, 43
SQL syntax, 43
SQL_CACHE, 43
sql_mode, 3
SQL_NO_CACHE, 43
SSL, 3
stack size, 3
startup, 43
statistics, 3, 43
storage engines, 43
stored programs, 43
ST_AsBinary(), 3
ST_AsGeoJSON(), 43
ST_AsText(), 3
ST_AsWKB(), 3
ST_AsWKT(), 3
ST_GeomFromGeoJSON(), 43
ST_SRID(), 43
ST_SwapXY(), 3
ST_X(), 43
ST_Y(), 43
SUBSTR(), 43
SUBSTRING_INDEX(), 43
SUM(), 3

T

tablespace, 3, 43
tablespace encryption, 3
tables_priv table, 43
temporary table, 43
temporary tables, 3
test suite, 3, 43
thread management, 43
TIME_TRUNCATE_FRACTIONAL, 3
transactions, 3
transportable tablespace, 43
triggers, 3
TRUNCATE PARTITION, 43
TRUNCATE TABLE, 3, 43
truncation, 3
two-phase commit, 43
type conversion, 43

U

UBSAN, 3

undo log, 3
undo tablespace, 3, 43
Unicode, 3, 43
UNINSTALL COMPONENT, 43
UNION, 43
UNIV_INNOCHECKSUM, 43
UPDATE, 3
upgrade, 3
upgrades, 3
usability, 3
user table, 43
UTF-16, 3
utf8mb4, 3
utf8mb4_0900_ai_ci, 3, 43
utf8mb4_general_ci, 3
UUIDs, 43
UUID_TO_BIN(), 43

V

Valgrind, 43
validate_password plugin, 43
version_tokens plugin, 3
views, 3, 43
virtual columns, 3
virtual generated column, 43
virtual indexes, 3, 43
Visual C++ Redistributable, 3

W

WEIGHT_STRING(), 3
WITH, 3

X

X Protocol, 3, 43
XA, 3, 43
XA transactions, 3, 43
XML, 3

Y

yaSSL, 3

Z

zlib_decompress, 43

