MySQL 5.7 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 5.7, up through MySQL 5.7.14. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 5.7 documentation, see the MySQL 5.7 Reference Manual, which includes an overview of features added in MySQL 5.7 (What Is New in MySQL 5.7), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.6 to MySQL 5.7 (Changes Affecting Upgrades to MySQL 5.7).

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: 2016-06-03 (revision: 9034)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 5.7, up through MySQL 5.7.14.

Legal Notices

Changes in MySQL 5.7.14 (Not yet released)

Version 5.7.14 has no changelog entries, or they have not yet been published because the product version has not yet been released.

Changes in MySQL 5.7.13 (2016-06-02)

Account Management Notes

- In MySQL 5.7.8, the maximum length of MySQL user names was increased from 16 to 32 characters, but some applicable contexts for this increase were overlooked. Additional changes in maximum user name length now have been applied:

  - The size of these `mysql` system table columns is increased:

    - The `definer` column of the `event` and `proc` tables

    - The `grantor` column of the `procs_priv` and `tables_priv` tables

    In each case, the column previously was defined as `CHAR(77)`, where 77 was chosen to permit a *`user_name@host_name`* string containing a user name up to 16 characters, a `@` character, and a host name up to 60 characters. Each column now is defined as `CHAR(93)`, reflecting an increase in permitted user name length from 16 to 32 characters.

  - A similar change from `CHAR(77)` to `CHAR(93)`applies to the `DEFINER` column of these `INFORMATION_SCHEMA` tables: `EVENTS`, `ROUTINES`, `TRIGGERS`, `VIEWS`. Along with the changes

to `INFORMATION_SCHEMA` tables, output from any corresponding `SHOW` statements now displays `DEFINER` values up to 93 characters. Examples of affected statements: `SHOW EVENTS`, `SHOW TRIGGERS`, `SHOW PROCEDURE STATUS`.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate the changes to the `mysql` system database. A server from MySQL 5.7.13 or higher for which `mysql_upgrade` has not been run continues to permit a maximum of 77 characters in the system tables mentioned previously, and an `ER_USER_COLUMN_OLD_LENGTH` error will occur in those system table contexts where a *`user_name@host_name`* value from 78 to 93 characters long is given.

Audit Log Notes

- The `audit_log` plugin that forms the basis for MySQL Enterprise Audit now supports fine-grained filtering of events. This enables a reduction in the number of log entries written to the audit log file, which increases overall performance due to fewer write operations during log recording. It also simplifies subsequent audit log processing in terms of readability and processing time.

  Fine-grained audit log filtering is rule based, implemented using tables that store filter definitions and a set of user-defined functions (UDFs) that enable filter manipulation. To simplify installing the tables and UDFs along with the `audit_log` plugin, an installation script is now provided. For more information, see Installing or Uninstalling MySQL Enterprise Audit, and Audit Log Filtering.

  By default, audit log filtering now logs no auditable events for any users. This differs from the pre-MySQL 5.7.13 legacy audit log, which logs all auditable events for all users. To produce log-everything behavior, see the installation instructions.

Configuration Notes

- On platforms for which systemd support is installed, systemd has the capability of managing multiple MySQL instances. For details, see Configuring Multiple MySQL Instances Using systemd. Consequently, `mysqld_multi` and `mysqld_multi.server` are not installed because they are unnecessary. (Bug #81093, Bug #23134620)

Security Notes

- The linked OpenSSL library for the MySQL Commercial Server has been updated to version 1.0.1t. Issues fixed in the new version are described at http://www.openssl.org/news/vulnerabilities.html.

  This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (Bug #23229564)

- MySQL Server now includes an SQL interface for keyring key management, implemented as a set of user-defined functions (UDFs) that access the functions provided by the internal keyring service. For more information, see Keyring Key Management Functions. For information about the keyring service functions invoked by the UDFs, see The Keyring Service. For general keyring information, see The MySQL Keyring.

Functionality Added or Changed

- For better separation of output from multiple statements, `mysqltest` now flushes output sent to `stdout` when the output destination is not a file. Previously, flushing occurred only for file output. (Bug #21435906)

- `support-files/MacOSX/ReadMe.txt` is no longer included in MySQL distributions. (Bug #81038, Bug #23088916)

- The version of the `tcmalloc` library included in MySQL distributions was very old. It has been removed and is no longer included with MySQL. (Bug #80994, Bug #23068660)

- The `my_make_scrambled_password()` function in the C client library was restricted earlier in MySQL 5.7 (not exported to client programs). The function has once again been made visible to client programs. (Bug #80974, Bug #23061746)

- It is possible to use `ALTER TABLE` to change the default value of a column *`col_name`*, which may change the value of a generated column expression that refers to the column using `DEFAULT(col_name)`. For this reason, `ALTER TABLE` operations that change the definition of a column now cause a table rebuild if any generated column expression uses `DEFAULT()`. (Bug #80299, Bug #22680839)

- 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)

Bugs Fixed

- **InnoDB:** A rollback operation run concurrently with an operation involving a generated virtual column caused a server exit. (Bug #23313102)

  References: This issue is a regression of: Bug #21869656.

- **InnoDB:** Potential buffer overflow issues were corrected for the `InnoDB memcached` plugin. (Bug #23187607)

- **InnoDB:** An `ALTER TABLE` operation that added an `AUTO_INCREMENT` column on a table with virtual columns raised an assertion. (Bug #23052231)

- **InnoDB:** Statements executed in a transaction that was rolled back asynchronously by a higher priority transaction caused a deadlock error and subsequent replication failure. (Bug #23021168, Bug #80898)

- **InnoDB:** An `ALTER TABLE` operation that attempted to add a generated virtual column and a full-text index raised an assertion. (Bug #23014521)

- **InnoDB:** The full-text index cache was freed during a background index cache synchronization. (Bug #22996488)

- **InnoDB:** A full-text index operation raised an assertion. (Bug #22963169)

- **InnoDB:** Memory was allocated to I/O slots unnecessarily, causing an apparent memory leak. (Bug #22956469, Bug #80772)

- **InnoDB:** An `ALTER TABLE` operation raised an assertion when attempting to create a key containing a generated column. (Bug #22951879)
- **InnoDB:** A startup failure due to an invalid option resulted in a server exit after a subsequent restart. An error returned by the `innobase_start_or_create_for_mysql` function was not checked. (Bug #22939581, Bug #80761)
- **InnoDB:** An `ALTER TABLE ... IMPORT TABLESPACE` operation on file-per-table tablespace containing an encrypted table failed when run in a different session than the preceding `ALTER TABLE ... DISCARD TABLESPACE` operation. (Bug #22918999, Bug #80708)
- **InnoDB:** A `FLUSH TABLES` operation on a table with a discarded tablespace raised an assertion. (Bug #22899690, Bug #80669)
- **InnoDB:** A DML operation involving a table with a virtual column raised an assertion. (Bug #22899305)
- **InnoDB:** An invalid read at `innobase_get_computed_value()` raised a Valgrind error. (Bug #22898168, Bug #80667)
- **InnoDB:** Online DDL operations like `ALTER TABLE ... ADD INDEX` were not permitted for tables created with the `ENCRYPTION` attribute. (Bug #22897921)
- **InnoDB:** `InnoDB memcached` code assumed the nonexistence of the `htonll()` function on OS X, resulting in a build failure on OS X 10.10 and later. (Bug #22865112)
- **InnoDB:** Starting the server in read-only mode failed when encrypted tables were present. The call to `fil_encryption_rotate()` was not skipped when the server was started in read-only mode. (Bug #22723797)
- **InnoDB:** An R-tree purge operation raised an assertion. (Bug #22698076, Bug #80327)
- **InnoDB:** An `INSERT` operation on a table with a `FULLTEXT` index and `FTS_DOC_ID` column failed because the inserted `FTS_DOC_ID` value exceeded the permitted gap between consecutive `FTS_DOC_ID` values. To avoid this problem, the permitted gap between the largest used `FTS_DOC_ID` value and new `FTS_DOC_ID` value was raised from 10000 to 65535. (Bug #22679185)
- **InnoDB:** Validation code for transparent page compression incorrectly permitted `innodb_strict_mode=OFF`, which allowed the `COMPRESSION` attribute to be applied to a general tablespace. Page compression is only supported with file-per-table tablespaces. (Bug #22615096, Bug #80182)
- **InnoDB:** An memory order issue related to atomic operations caused assertion failures on ARM64 and POWER platforms. (Bug #22608616)
- **InnoDB:** `DROP TABLESPACE` returned an error if the remote general tablespace data file was missing. (Bug #22232892, Bug #79330)
- **InnoDB:** An `ALTER TABLE` operation that changed table row format from `COMPACT` to `COMPRESSED` raised an assertion. A function involved in the operation passed incorrect page size information. (Bug #22046353)
- **InnoDB:** With `innodb_autoinc_lock_mode=0`, multiple threads waiting for a table-level lock caused an unexpected deadlock. (Bug #21983865, Bug #78761)
- **InnoDB:** A race condition in `trx_kill_blocking()` raised an assertion. (Bug #21508537)
- **InnoDB:** An `OPTIMIZE TABLE` operation on a table with a full-text index raised an assertion. (Bug #21378944)

- **InnoDB:** A buffer pool load operation resulted in a “Cannot allocate 0 bytes” error. (Bug #21371070)
- **InnoDB:** A `FLUSH TABLES ... FOR EXPORT` operation appeared to stall. A loop in the `ibuf_contract_in_background` function failed to exit. (Bug #21133329, Bug #77011)
- **InnoDB:** A full-text query raised an assertion. Under certain circumstances, DDL operations such as `ALTER TABLE ... RENAME` caused full-text auxiliary tables to be removed on server restart. (Bug #13651665)
- **Replication:** With `gtid_mode=ON`, executing an empty query before setting `gtid_next` made the latter action fail. It was because only statements that do not change the data can be executed before one can successfully change `gtid_next`, and an empty query was not considered one of those “safe” statements. This fix allows the setting of `gtid_next` after an empty query. (Bug #22811150)
- **Replication:** Slaves running MySQL 5.7 could not connect to a MySQL 5.5 master due to an error retrieving the `server_uuid`, which is not part of MySQL 5.5. This was caused by changes in the method of retrieving the `server_uuid`. (Bug #22748612)

  References: This issue is a regression of: Bug #21455603.
- **Replication:** Setting certain semisynchronous-replication configurations on the master server when semisynchronous replication was not enabled on it might cause the server to exit. This fix prevents the unexpected exits to occur in the situation. (Bug #22602324)
- **Replication:** The variable `explicit_defaults_ts` was not initialized during the construction of a `Query_event` object. That caused Valgrind warnings for dependency on an uninitialized variable. This fix makes sure the variable is initialized. (Bug #22110916, Bug #78999)

  References: This issue is a regression of: Bug #18885916, Bug #72794.
- **Replication:** In the `next_event()` function, which is called by a slave's SQL thread to read the next even from the relay log, the SQL thread did not release the `relaylog.log_lock` it acquired when it ran into an error (for example, due to a closed relay log), causing all other threads waiting to acquire a lock on the relay log to hang. With this fix, the lock is released before the SQL thread leaves the function under the situation. (Bug #21697821)

  References: See also: Bug #20492319.
- **Replication:** A partially failed statement was not correctly consuming an auto-generated or specified GTID when binary logging was disabled. The fix ensures that a partially failed `DROP TABLE`, a partially failed `DROP USER` or a partially failed `DROP VIEW` consume respectively the relevant GTID and save it into `@@GLOBAL.GTID_EXECUTED` and `mysql.gtid_executed` table when binary logging is disabled. (Bug #21686749)
- **Replication:** An intermittent ASan error was being reported on the `rpl.rpl_checksum_cache` test. The error reported was related to the binary log sender doing a heap-use-after-free on a given memory address. (Bug #78995, Bug #22109863)
- **Replication:** `mysqldump` has been updated to make it compatible with multi-source replication. Now when replication channels other than the default channel are found, `mysqldump --dump-slave` outputs a `CHANGE MASTER TO` statement for each replication channel. (Bug #78467, Bug #21855705)
- **Replication:** If a multi-threaded replication slave running with `relay_log_recovery=1` stopped unexpectedly, during restart the relay log recovery process could fail. This was due to transaction inconsistencies not being filled, see Handling an Unexpected Halt of a Replication Slave. Prior to this fix, to recover from this situation required manually setting `relay_log_recovery=0`, starting the slave with `START SLAVE UNTIL SQL_AFTER_MTS_GAPS` to fix any transaction inconsistencies and then

restarting the slave with `relay_log_recovery=1`. This process has now been automated, enabling relay log recovery of a multi-threaded slave upon restart automatically. (Bug #77496, Bug #21507981)

- Fedora builds now are configured using `-DMYSQL_MAINTAINER_MODE=0` to silence GCC 6 warnings. (Bug #23274249)
- Allocation of a large number (2^20) of Performance Schema index statistic objects could cause a server exit. (Bug #23188107)
- If the `keyring_okv` plugin configuration directory was missing, attempts to rotate the `InnoDB` master key could cause a server exit. (Bug #23149683)
- `INSERT` with `ON DUPLICATE KEY UPDATE` and `REPLACE` on a table with a foreign key constraint defined failed with an incorrect “duplicate entry” error rather than a foreign key constraint violation error. (Bug #23135731)

  References: This issue is a regression of: Bug #78853, Bug #22037930.
- Contention in Performance Schema mutex instrumentation creation and destruction has been reduced, such that mutexes for which instruments are frequently created and destroyed are maintained in separate pages from those for which instruments are rarely created and destroyed. (Bug #22965826)
- With certain build options, an uninitialized variable in `get_key_scans_params()` could produce a compilation error. (Bug #22916059)
- Adding new tokens to the parser caused query digest values to change. (Bug #22906606)
- For debug builds, `CONCAT_WS()` could raise an assertion if there was nothing to append. (Bug #22888420)
- Fixed Valgrind warnings for failed `LEAST()` evaluations. (Bug #22883278)
- `INET_NTOA()` could cause a server exit when producing an error message. (Bug #22881810)

  References: This issue is a regression of: Bug #22042027.
- The `my_write()` call could cause a server exit if it attempted to check the current connection and the connection had been killed. (Bug #22867809)

  References: This issue is a regression of: Bug #21688407.
- Invoking Enterprise Encryption functions in multiple threads simultaneously could cause a server exit. (Bug #22839278)
- Setting `log_syslog_tag` to `NULL` could cause a server exit. (Bug #22834781)

  References: This issue is a regression of: Bug #22180046.
- If the expression for an indexed generated column contained an `AND` or `OR` operator, the optimizer could choose that index too often and create execution plans that produced incorrect results. (Bug #22810883)
- `CREATE TABLE` statements in `mysqlpump` output could be missing `KEY` clauses and would not load. (Bug #22726732)
- Attempting to use Enterprise Encryption functions after creating and dropping them could cause a server exit. (Bug #22669012)
- Setting `sort_buffer_size` to a very large value could cause some operations to fail and result in a server exit. (Bug #22594514)

- For an `InnoDB` table containing generated columns, using the table in a join could result in a server exit due to improper error checking. (Bug #22561845)
- `REPLACE` on a table with an indexed generated column could cause a server exit if the index prefix length was calculated incorrectly. (Bug #22445211)
- An `UPDATE` operation affecting a generated virtual `BLOB` column could cause a server exit. (Bug #22444212)
- `SHOW CREATE TABLE` for a table containing a generated column could cause a server exit or produce an `Illegal mix of collations` error. (Bug #22392268)
- On a slave server, replication of an `UPDATE` statement that updated an indexed `BLOB` virtual generated column of an `InnoDB` table could cause a server exit. (Bug #22241015)
- An assertion could be raised when a deadlock occurred due to a `SELECT ... GROUP BY ... FOR UPDATE` query executed using a Loose Index Scan. (Bug #22187476)
- `mysqlpump` could exit due to improper handling of error conditions in a dump thread. (Bug #22017120)
- Several potential buffer overflow issues were corrected. (Bug #21977380, Bug #23187436, Bug #23202778, Bug #23195370, Bug #23202699)
- If the CA certificate as given to the `--ssl-ca` option had an invalid path, yaSSL returned an error message different from OpenSSL. Now both return `SSL connection error: SSL_CTX_set_default_verify_paths failed`. (Bug #21920657)
- Installing MySQL from a `yum` or `zypper` repository resulted in `/var/log/mysqld.log` being created with incorrect user and group permissions. (Bug #21879694, Bug #78512)
- With `show_compatibility_56=OFF`, `SHOW VARIABLES` and `SHOW STATUS` statements having a `WHERE` clause did not work. (Bug #21783883)
- The `audit_log` plugin failed to abort the current operation when told to do so by a plugin handler for a `MYSQL_AUDIT_PARSE_PREPARSE` event. (Bug #21457956)
- Memory leaks reported by Valgrind for `mysqlpump` were fixed. (Bug #21237667)
- Some string functions returned one or a combination of their parameters as their result. If one of the parameters had a non-ASCII character set, the result string had the same character set, resulting in incorrect behavior when an ASCII string was expected. (Bug #18740222)
- On Fedora 24, upgrades using a Community MySQL Server RPM failed to replace an installed MariaDB Galera server due to a change in the MariaDB package. (Bug #81390, Bug #23273818)
- The `mysql_read_defaults_options()` function was missing a break statement, causing any option value for the `--ssl-cipher` option also to be applied to the `--tls-version` option, with unpredictable results. (Bug #81139, Bug #23129821)
- A `DELETE` from joined tables using a derived table in the `WHERE` clause failed if one of the joined tables was used in the derived table. (Bug #81014, Bug #23074801)
- Compiling of clients that used the MySQL C API could fail if they used an obsolete path to `mysql.h` or `<mysql/mysql.>` and did not set the include path. This was due to use of `#include <file_name>` notation rather than `#include "file_name"` in internal MySQL header files, which have been adjusted. (Bug #80935, Bug #23047194)
- MySQL did not compile under Solaris 12 using Sun Studio. To correct this, instances of `__attribute__` were changed to `MY_ATTRIBUTE`. (Bug #80748, Bug #22932576)

- The `service_mysql_keyring.h` and `services.h` header files misspelled `mysql_keyring_service_st` as `mysql_keyring_file_service_st`. (Bug #80688, Bug #22908232)

- The fix for Bug #79194 did not cover the `eq_ref` access method, with the result that left joins could return incorrect results.

  

  Note

  This bug fix has a very small negative performance effect such that it fails to cache an `eq_ref`-accessed row that is on the inner side of an outer join. Regular inner joins are not affected.

  (Bug #80526, Bug #22833364)

  References: This issue is a regression of: Bug #79194, Bug #22176604.

- The `sanity()` macro in `strings/decimal.c` produced Valgrind warnings due to reading uninitialized buffer contents. (Bug #80461, Bug #22782203, Bug #22839915)

- Starting the server with `--initialize` failed if the `keyring_file_data` system variable was also set at startup. To handle this, with `--initialize` the server no longer skips registration of plugins loaded with the `--early-plugin-load` option. (Bug #80451, Bug #22777039)

- Loading the `Rewriter` query rewrite plugin when there was no `query_rewrite` database resulted in Valgrind warnings. (Bug #80333, Bug #22710312)

- The optimizer transformed `EXISTS (SELECT * ...)` constructs to `EXISTS (SELECT 1 ...)` before all columns in the inner subquery had been resolved, which could result in rejection of valid queries that included a `HAVING` clause without `GROUP BY` in the subquery. (Bug #80231, Bug #22655856)

- For a server compiled with `-DWITH_PERFSCHEMA_STORAGE_ENGINE=0`, a memory leak could occur for buffered log messages used during server startup. (Bug #80089, Bug #22578574)

- A query could return incorrect results under these conditions: A column with a default value contained `NULL`; `SELECT DISTINCT` or a `GROUP BY` clause was used and the column containing the `NULL` value was part of the select list; an `InnoDB` temporary table was used during query processing. (Bug #79591, Bug #22343910)

- A null pointer dereference of a parser structure could occur during stored procedure name validation. (Bug #79396, Bug #22286421)

- Database initialization failed during installation using Ubuntu 15.10 packages. (Bug #79377, Bug #22252900)

- `mysql_upgrade` failed to widen the `User` and `Proxied_user` columns in the `mysql.proxies_priv` system table from 16 to 32 characters. (Bug #78254, Bug #21762656)

- Failure of `UNINSTALL PLUGIN` could lead to inaccurate or confusing errors for subsequent `INSTALL PLUGIN` operations. (Bug #74977, Bug #20085672)

- `mysqld_multi` displayed misleading error messages when it was unable to execute `my_print_defaults`. (Bug #74636, Bug #19920049)

- Previously, upgrading the server using an RPM package (including installation using `yum`) required upgrading the client package to the same MySQL version, which may be undesirable for some

installations. This rule has been relaxed so that upgrading to a General Availability (GA) server version requires only that some GA client version be installed, which is less likely to require a client upgrade. (Bug #72230, Bug #18518216)

- `mysqldump` failed silently with no error message when it encountered an error while executing `FLUSH LOGS`. (Bug #71783, Bug #18284273)

Changes in MySQL 5.7.12 (2016-04-11)

Starting with MySQL Server 5.7.12, we are introducing the concept of Rapid Plugins. Out of the box, 5.7.12 follows the same stability promise as our current release process, but it also allows for plugins with additional functionality to be installed very easily. The first plugin to be released with MySQL Server 5.7.12 following this concept is the new X Plugin, which exposes a new communications protocol called the X Protocol.

The expanded capabilities of the X Protocol enable us to provide the new X DevAPI in our MySQL Connectors and Client applications, like the new MySQL Shell. The goal of the X DevAPI is to support a new schema object type called document collections as well as relational and combined document store/relational capabilities. Now developers, designers and DBAs can deploy MySQL databases that implement document store, relational, or hybrid document/relation models. For documentation about how to get started using MySQL as a document store, see Using MySQL as a Document Store.

- Configuration Notes
- Security Notes
- Functionality Added or Changed
- Bugs Fixed

Configuration Notes

- **Incompatible Change:** To load a keyring plugin, the `--early-plugin-load` option is used. Previously, the default `--early-plugin-load` option value was the name of the `keyring_file` plugin library file. Now the default value is empty.

  

  Important

  `InnoDB` tablespace encryption requires the `keyring_file` plugin to be loaded prior to `InnoDB` initialization, so this change of default `--early-plugin-load` value introduces an incompatibility for upgrades from 5.7.11 to 5.7.12 or higher. Administrators who have encrypted `InnoDB` tablespaces must take explicit action to ensure continued loading of the `keyring_file` plugin: Start the server with an `--early-plugin-load` option that names the plugin library file. For example, on platforms where the plugin library file suffix is `.so`, use these lines in the server `my.cnf` file:

  ```
  [mysqld]
  early-plugin-load=keyring_file.so
  ```

  On other platforms, adjust the file name suffix as necessary. For more information, see The MySQL Keyring.

  (Bug #80413, Bug #22748738)

Security Notes

- **InnoDB:** The `InnoDB` tablespace encryption feature, which provides at-rest data encryption for `InnoDB` tables, now supports Oracle Key Vault for encryption key management. Oracle Key Vault support relies on the `keyring_okv` plugin which is available in MySQL Enterprise Edition. A secure and robust encryption key management solution such as OKV is critical for security and for compliance with various security standards. For more information, see InnoDB Tablespace Encryption.

- The linked OpenSSL library for the MySQL Commercial Server has been updated to version 1.0.1s. Issues fixed in the new version are described at http://www.openssl.org/news/vulnerabilities.html.

  This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (Bug #22685885, Bug #22923458)

- MySQL Enterprise Edition now includes a `keyring_okv` plugin that uses Oracle Key Vault for keyring backend storage. For more information, see The MySQL Keyring.

Functionality Added or Changed

- For queries with many `OR` conditions, the optimizer now is more memory-efficient and less likely to exceed the memory limit imposed by the `range_optimizer_max_mem_size` system variable. In addition, the default value for that variable has been raised from 1536000 to 8388608. (Bug #79450, Bug #22283790)

Bugs Fixed

- **InnoDB; Partitioning:** `INSERT` and `SELECT` statements against a partitioned `InnoDB` table having generated columns were not always handled correctly. (Bug #22444530)

  References: See also: Bug #21776494, Bug #21824564, Bug #21864838.

- **InnoDB:** An error was returned on startup when a replication slave attempted to access an encrypted table. The server UUID used to compose the master key name was not persisted to the tablespace data file, resulting in the use of an incorrect master key. (Bug #22912582)

- **InnoDB:** In debug builds, an unnecessary `buf_block_align()` call could cause a latching order violation. A DML operation resulted in a page mismatch assertion due to the same `buf_block_align()` call. (Bug #22709463, Bug #21321238)

- **InnoDB:** An `ALTER TABLE ... ADD COLUMN` operation on a table with virtual columns raised an assertion. (Bug #22650296)

- **InnoDB:** An unnecessary comparison in tablespace encryption code caused compiler warnings. (Bug #22645816)

- **InnoDB:** `INNODB_SYS_TABLESPACES` could report incorrect `ALLOCATED_SIZE` and `FILE_SIZE` values for a general tablespace created outside of the data directory. The `i_s_dict_fill_sys_tablespaces()` function generated an incorrect remote file path. (Bug #22590095, Bug #80070)

- **InnoDB:** In debug builds, an update operation on a table with virtual columns raised an assertion. (Bug #22572997)

- **InnoDB:** Modifications were made to `InnoDB` code to handle warnings when compiling with Microsoft Visual Studio 2015. (Bug #22542547, Bug #79964)

- **InnoDB:** Running `REPLACE` operations on multiple connections resulted in a hang. (Bug #22530768, Bug #79185)

- **InnoDB:** Operations relating to tabespace encryption resulted in a hang on FreeBSD. (Bug #22520464, Bug #79901)

- **InnoDB:** MySQL stalled when synchronizing the `InnoDB` full-text index cache. (Bug #22516559, Bug #16510576, Bug #73816)

- **InnoDB:** A `CREATE TABLE ... DATA DIRECTORY` operation failed to create a table while `innodb_flush_method` was set to `O_DIRECT`. (Bug #22180006, Bug #79200)

  References: This issue is a regression of: Bug #21113036.

- **InnoDB:** In debug builds, a buffer pool resizing operation resulted in a segmentation violation. The `buf_pool_resizing` variable was not protected. (Bug #22179317)

- **InnoDB:** The `innodb_open_files` setting could exceed the open files limit. (Bug #22111472)

- **InnoDB:** `InnoDB` attempted `crc32` checksum validation instead of `innodb` after a `crc32` checksum validation failure, causing repeated `crc32` checksum validation attempts. Checksum validation order is now optimized dynamically.

  Thanks to Daniel Black for the patch. (Bug #79725, Bug #22452597)

- **Replication:** When a slave with no binary log connected to a group replication master, it failed with `Error running query`, but no information about the failure was available from the output of `SHOW SLAVE STATUS` or in the `replication_applier_status_by_worker` table. (Bug #22699395)

- **Replication:** Setting `relay_log_recovery=1` generated an error during recovery, due to repositories not being initialized. (Bug #22523554)

- **Replication:** Issuing `STOP SLAVE` caused a spurious `Error reading packet from server: Lost connection to MySQL server during query` message to be written to the error log. With this fix, when connection to the master is lost, the `abort_slave` flag is checked and the error message is printed only if the flag is not set. (Bug #22305605, Bug #79504)

  References: See also: Bug #12977988, Bug #22290309.

- **Replication:** When the binary log was disabled, the GTID specified by `gtid_next` when committing an empty transaction caused by a `BEGIN` statement was saved in `gtid_executed` when it should not have been. This did not occur when the binary log was enabled.

  Now, when `gtid_next` is set for the current session, `BEGIN` raises an **`ER_CANT_DO_IMPLICIT_COMMIT_IN_TRX_WHEN_GTID_NEXT_IS_SET`** error inside any transaction regardless of whether binary logging is enabled, since this statement causes an implicit commit. In this case, the GTID specified by `gtid_next` is not saved in `gtid_executed`. (Bug #22130929)

- **Replication:** When replication was configured but not started on a slave, the variable `currently_executing_gtid` was not properly initialized, but it would be used if the performance_schema table replication_applier_status_by_worker was queried, causing assertion failures and pointer issues. With this fix, the variable is now properly initialized at the construction of the `Relay_log_info` object. (Bug #21953132, Bug #78722)

- **Replication:** `RESET SLAVE ALL` could delete a channel even when `master_pos_wait` and `wait_until_sql_thread_after_gtid` were still waiting for transactions to be applied. This could cause a MySQL server exit when the functions tried to access the channel that was deleted. Now, a channel reference counter was added that is increased if the channel should not be deleted when the functions are running. `RESET SLAVE ALL` will wait for no reference, and then it will delete the channel. (Bug #21842399, Bug #78445)

- **Replication:** When `relay_log_recovery=1`, a slave server, at its initialization, still tried to scan the relay log files in order to update `Retrieved_Gtid_Set` and the transaction parser state, which was an

unnecessary waste of resources because the slave I/O thread would just be initialized to the SQL thread position in the new relay log file. With this fix, the slave server skips scanning the relay log files when `relay_log_recovery=1`. (Bug #21798355, Bug #78352)

- **Replication:** XA transactions were not handled correctly when `--gtid-mode=ON` and the binary log was disabled. It was because on both master and slave servers, the GTID state was sometimes not saved and the GTID ownership was sometimes not cleaned up. This fix makes sure those steps are now properly performed. (Bug #21618727)

  References: See also: Bug #22165138, Bug #77740, Bug #21452916.

- **Replication:** When a multi-threaded slave stopped with an error, the same error message was printed three times. Now, the SQL thread's kill acceptance status is saved, and only printed once. (Bug #21198611, Bug #77237)

- **Replication:** When using a multi-threaded slave with `relay_log_info_repository` set to `TABLE`, the slave applier thread failed to write XA transactions to the worker configuration. (Bug #20988291)

- **Replication:** `mysqlbinlog --verbose` displayed BINARY and VARBINARY data as ordinary strings, causing any single quote (“”) or backslash (“\”) among the data to be printed as such, which was confusing to the users and, in the case of a backslash, caused the next character to be skipped. This fix makes `mysqlbinlog` print the characters' hexadecimal values (“\x27” for single quote and “\x5c” for backslash) instead. (Bug #20836250)

- **Replication:** The test case `main.merge` failed when the variables `binlog_format` was set to “`ROW,`” as the server tried to get information for table creation for a child table before it was opened. With this fix, the server skips getting information for the table in the situation. (Bug #20574550, Bug #75976)

- **Replication:** When a server was run with `relay_log_info_repository=TABLE` and the `--super-read-only` option enabled, a statement which caused an update to the slave info tables, such as `STOP SLAVE` or `CHANGE MASTER TO`, resulted in a `'STOP SLAVE' failed: 1290: The MySQL server is running` error, preventing the statement being executed. The fix ensures that replication operations are permitted regardless of the setting of `read_only` and `super_read_only`. (Bug #78963, Bug #22097534)

- **Replication:** Valgrind tests of `mysqlbinlog` revealed some memory leaks. (Bug #78223, Bug #21697461, Bug #78966)

- **Replication:** If a query on a master generated an error and partial results were written to the binary log, for example due to a `DROP TABLE IF EXISTS` statement applying to multiple tables that would break foreign key constraints, when a slave configured with replication filters encountered the query it could be incorrectly binary logged. This caused errors such as:

  ```
  Last_SQL_Error: Query caused different errors on master and slave. Error on
  master: message (format)='Cannot delete or update a parent row: a foreign
  key constraint fails' error code=1217 ; Error on slave: actual message='no
  error', error code=0. Default database: 'db1'. Query: 'DROP TABLE IF EXISTS
  `table1` /* generated by server */'
  ```

  There were two fixes required for this bug.

  - If a `DROP TABLE` statement used to drop a single table fails, to avoid partial results causing this bug the query is not written to the binary log. If a `DROP TABLE` statement used to drop a list of tables fails, any partial results it generates are written to the binary log with an error.

  - When a query that generates an error as expected was received by a slave but it was skipped due to replication filters, the slave was incorrectly checking the error. The fix for Bug #76493 ensures that

this comparison of the expected error from the master with the actual error from the slave does not happen.

(Bug #77684, Bug #21435502)

References: See also: Bug #20797764, Bug #76493.

- `RTRIM()` on large strings could be very slow. (Bug #22884187)

  References: This issue is a regression of: Bug #18315770, Bug #12368495.

- Integer overflow could occur during client handshake processing, leading to a server exit. (Bug #22722946)
- Missing initializers in the query plan constructor resulted in Valgrind warnings. (Bug #22573117)
- The obsolete `support-files/MySQL-shared-compat.spec.sh` file was removed from distributions. (Bug #22525609)
- `mysqlpump` failed (syntax error) when a view name contained a space character. View names are now quoted. (Bug #22505474)
- `UNHEX()` with an invalid argument could result in garbage characters in the warning message. (Bug #22358933)
- Improper host name checking in X509 certificates could permit man-in-the-middle attacks. (Bug #22295186, Bug #22738607)
- For debug builds, when an indexed nonnullable generated column with an empty string generated expression was updated during an insert for a duplicated key, there was an optimization that resulted in the server failing to find the duplicated index, causing an assertion to be raised. (Bug #22195364)
- A boolean mode full-text search caused a segmentation fault. (Bug #22176795)
- Queries on generated columns that used `WITH ROLLUP` could raise an assertion. (Bug #22131343)
- For a prepared statement that used a derived table, an assertion could be raised at execute time when checking statement privileges. (Bug #22108567)
- Concurrent selecting and flushing of a `FEDERATED` table while killing connections accessing it could result in a server exit. (Bug #21918190)
- For debug builds, when a query using join buffering and one of the tables inserted into the join buffer was accessed using a dynamic range scan on an index containing a virtual column, a Valgrind error occurred when writing columns to the join buffer. (Bug #21872184)
- After iterations of uninstalling and installing the `audit_log` plugin, the server could hang. (Bug #21796658)
- With a `LOCK TABLES` statement in effect, access to Performance Schema tables could fail, as could `SHOW STATUS` with `show_compatibility_56=OFF`. (Bug #21789221)
- Executing `GRANT PROXY` statements after altering the definition of the `mysql.user` system table could result in a server exit. (Bug #21699037)
- Certain error messages included part of the SQL statement that produced them, possibly exposing data. (Bug #21682356)
- Although it is possible to create nontemporary tables using the prefix `#sql`, Performance Schema assumed that tables named using this prefix were temporary and could be ignored. Performance

Schema now uses table attributes other than the name to identify temporary tables. (Bug #21105475, Bug #22532368, Bug #79934)

- Account filtering performed by the `audit_log` plugin incorrectly used the account named by the `USER()` function rather than the `CURRENT_USER()` function (the latter being the account used for authentication). (Bug #19509471)

- `NAME_CONST()` can return null if incorrect arguments are given. In some cases, this was not handled and could cause a server exit. (Bug #19143243)

- Character set conversion operations on `NULL` parameters to prepared statements could cause a server exit. (Bug #18823979)

- Loose Index Scan was not chosen for queries that had an equality condition. (Bug #18109609)

- Long or complex SQL statements could cause the parser to run out of memory. The new `parser_max_mem_size` system variable now enables control over the maximum amount of memory available to the parser. The default value places no limit on memory available, but the value can be reduced to protect against out-of-memory situations. (Bug #14744160)

- A Valgrind warning for `memory_free_noop()` was silenced. (Bug #80457, Bug #22782197)

- The MySQL server failed to start if built with the `-m32` option on Solaris/Sparc, due to improper static data alignment in `init_delegates()`. (Bug #80445, Bug #22763880)

- The `plugin_keyring.h` header file misspelled `st_mysql_keyring` as `st_mysql_keyring_file`. (Bug #80414, Bug #22748867)

- For shared-memory connections on Windows, the client library opened a handle on a mutex but did not close it. Subsequent attempts to restart the server on the other end of the connection failed if the client still had the mutex handle open. (Bug #80234, Bug #22646779)

- `mysqld` attempted to initialize plugins specified using the `--early-plugin-load` option when the `--help` option was specified. (Bug #80077, Bug #22573767)

- `ALTER USER IDENTIFIED WITH ...` expired the account password, even if the authentication plugin did not support password expiration. (Bug #79999, Bug #22551523)

- Some Performance Schema global instruments, if not enabled at server startup, could be in a state where it was not possible to enable them at runtime. This restriction has been lifted. Affected instruments include `mutex`, `rwlock`, `cond`, and `socket` instances.

  Thanks to Zhang Yingqiang for the patch. (Bug #79784, Bug #22517278)

- Setting the `super_read_only` system variable at server startup had no effect. (Bug #79328, Bug #22233503)

- These `audit_log` plugin issues were corrected:

  - Calling `my_message()` from the `MYSQL_AUDIT_GENERAL_CLASS` handler resulted in infinite recursion.

  - Diagnostic messages were improved for the case when the `MYSQL_AUDIT_GENERAL_CLASS` handler returned nonzero.

  - Calling `my_message()` from the `MYSQL_AUDIT_SERVER_STARTUP_CLASS` handler did not abort server startup as it should have.

  - `SHOW GLOBAL VARIABLES` produced different output for the `null_audit_abort_value` system variable with `show_compatibility_56` enabled and disabled.

  (Bug #79079, Bug #22136709, Bug #79091, Bug #22142166, Bug #79092, Bug #22142209, Bug #21783798)

- For `INSERT` and `UPDATE` operations that caused `FOREIGN KEY` constraint violations, errors were reported rather than warnings when the `IGNORE` keyword was used. (Bug #78853, Bug #22037930)

  References: See also: Bug #23135731.

- Using the server session service, executing an SQL statement from a thread with no VIO context could raise an assertion. (Bug #78734, Bug #21959409)

- Error messages raised by `JSON_KEYS()` and `JSON_EXTRACT()` contained the wrong data. (Bug #78699, Bug #22026278)

- `CREATE VIEW` statements that used the `TIMESTAMPDIFF()` function with `MICROSECOND` as the unit resulted in incorrect view definitions. (Bug #78506, Bug #21877062)

- For an existing user, `CREATE USER IF NOT EXISTS` produced an error rather than a warning. Similarly, for an nonexisting user, `ALTER USER IF EXISTS` produced an error rather than a warning. (Bug #78374, Bug #21807286)

- For some queries, an Index Merge access plan was chosen over a range scan when the cost for the range scan was the same or less. (Bug #77209, Bug #21178196)

- `UNHEX()` could attempt a left shift of a negative number. (Bug #73964, Bug #19642015)

- `EXPLAIN` for `SELECT ... FOR UPDATE` statements acquired locks. (Bug #72858, Bug #18899860)

- Processlist state information was not updated correctly for `LOAD DATA INFILE` and could show a state different from `executing`. (Bug #69375, Bug #16912362)

Changes in MySQL 5.7.11 (2016-02-05)

- Compilation Notes
- Data Type Notes
- Installation Notes
- Packaging Notes
- Security Notes
- Functionality Added or Changed
- Bugs Fixed

Compilation Notes

- A value of `system` is now permitted for the `WITH_BOOST` `CMake` option. If this option is not set or is set to `system`, it is assumed that the correct version of Boost is installed on the compilation host in the standard location. In this case, the installed version of Boost is used rather than any version included with a MySQL source distribution. (Bug #22224313)

- In addition to the `mysql-5.7.11.tar.gz` source tarball, another tarball named `mysql-boost-5.7.11.tar.gz` is provided. The new tarball contains everything in the first tarball, but also contains all the required Boost header files in a subdirectory named `boost`. This is for the benefit of those who do not have the correct Boost version installed and do not wish to or cannot download it. To build from this source distribution, add `-DWITH_BOOST=boost` to the `CMake` command line.

Data Type Notes

- Bit functions and operators comprise `BIT_COUNT()`, `BIT_AND()`, `BIT_OR()`, `BIT_XOR()`, `&`, `|`, `^`, `~`, `<<`, and `>>`. Currently, bit functions and operators require `BIGINT` (64-bit integer) arguments and return `BIGINT` values, so they have a maximum range of 64 bits. Arguments of other types are converted to `BIGINT` and truncation might occur.

  A planned extension for MySQL 5.8 is to permit bit functions and operators to take binary string type arguments (`BINARY`, `VARBINARY`, and the `BLOB` types), enabling them to take arguments and produce return values larger than 64 bits. Consequently, bit operations on binary arguments in MySQL 5.7 might produce different results in MySQL 5.8. To provide advance notice about this potential change in behavior, the server now produces warnings for bit operations for which binary arguments will not be converted to integer in MySQL 5.8. These warnings afford an opportunity to rewrite affected statements. To explicitly produce MySQL 5.7 behavior in a way that will not change after an upgrade to 5.8, cast bit-operation binary arguments to convert them to integer. For more information and examples, see Bit Functions and Operators.

Installation Notes

- Previously, `mysqld --initialize` required the data directory to not exist or, if it existed, to be empty. Now an existing data directory is permitted to be nonempty if every entry either has a name that begins with a period (`.`) or is named using an `--ignore-db-dir` option. (Bug #79250, Bug #22213873)

Packaging Notes

- Packaging support for Ubuntu 15.10 was added. (Bug #79104, Bug #22147191)

Security Notes

- The linked OpenSSL library for the MySQL Commercial Server has been updated from version 1.0.1p to version 1.0.1q. Issues fixed in the new version are described at http://www.openssl.org/news/vulnerabilities.html.

  This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (Bug #22348181)

- The default value of the `default_password_lifetime` system variable that controls the global password expiration policy has been changed from 360 (360 days) to 0 (no password expiration). The default of 360 sometimes took people by surprise when account passwords expired a year after upgrading to MySQL 5.7. To continue to use a value other than 0 as the password expiration, start the server with an explicit setting for `default_password_lifetime`. For example, use these lines in an option file:

  ```
  [mysqld]
  default_password_lifetime=360
  ```

  (Bug #77277, Bug #21284761)

- MySQL Server now supports a keyring service that enables internal server components and plugins to securely store sensitive information for later retrieval. The implementation includes a `keyring_file`

plugin that stores keyring data in a file local to the server host. For more information, see The MySQL Keyring, Writing Keyring Plugins, and The Keyring Service.

- MySQL client programs now support an `--ssl-mode` option that enables you to specify the security state of the connection to the server. Permitted option values are `PREFERRED` (establish a secure encrypted connection if the server supports the capability, falling back to an unencrypted connection otherwise), `DISABLED` (establish an unencrypted connection), `REQUIRED` (establish a secure connection, or fail), `VERFIFY_CA` (like `REQUIRED`, but additionally verify the server certificate), `VERIFY_IDENTITY` (like `VERIFY_CA`, but additionally verify that the server certificate matches the host name to which the connection is attempted). For backward compatibility, the default is `PREFERRED` if `--ssl-mode` is not specified.

  These clients support `--ssl-mode`: `mysql`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlshow`, `mysqlpump`, `mysqlslap`, `mysqltest`, `mysql_upgrade`.

  The `--ssl-mode` option comprises the capabilities of the client-side `--ssl` and `--ssl-verify-server-cert` options. Consequently, both of those options are now deprecated and will be removed in a future MySQL release. 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 is *not* deprecated.)

  For the C API, the new `MYSQL_OPT_SSL_MODE` option for `mysql_options()` corresponds to the `--ssl-mode` option. The `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. They are now deprecated and will be removed in a future MySQL release. Use `MYSQL_OPT_SSL_MODE` with an option value of `SSL_MODE_REQUIRED` or `SSL_MODE_VERIFY_IDENTITY` instead.

  For more information, see Command Options for Secure Connections, and mysql_options().

  In consequence of this change, the minor C API version number was incremented.

Functionality Added or Changed

- **InnoDB:** A new `InnoDB` configuration option, `innodb_tmpdir`, allows you to configure a separate directory for temporary files created during online `ALTER TABLE` operations that rebuild the table. This option was introduced to help avoid MySQL temporary directory overflows that could occur as a result of large temporary files created during online `ALTER TABLE` operations. `innodb_tmpdir` can be configured dynamically using a `SET` statement.

  Online `ALTER TABLE` operations that rebuild a table also create an *intermediate* table file in the same directory as the original table. The `innodb_tmpdir` option is not applicable to intermediate table files. (Bug #19183565)

- **InnoDB:** `InnoDB` now supports at-rest data encryption for `InnoDB` tables stored in file-per-table tablespaces. Encryption is enabled by specifying the `ENCRYPTION` option when creating or altering an `InnoDB` table. For more information, see InnoDB Tablespace Encryption.

- **Replication:** The `log_statements_unsafe_for_binlog` variable was added to provide control over whether the warnings generated by error 1592 are added to the binary log or not.

- yaSSL was upgraded to version 2.3.9. This upgrade corrects an issue in which yaSSL handled only cases of zero or one leading zeros for the key agreement instead of potentially any number, which in rare cases could cause connections to fail when using DHE cipher suites. (Bug #22361038)

- The Valgrind function signature in `mysql-test/valgrind.supp` was upgraded for Valgrind 3.11. (Bug #22214867)
- The `audit_log` plugin now produces events in the `MYSQL_AUDIT_TABLE_ACCESS_CLASS` class. These events are abortable. (Bug #21458192)
- The format of log output produced by `mysqld_safe` now is configurable using the `--mysqld-safe-log-timestamps` option. This option can be used to produce log timestamps in formats compatible with the server or in formats used by `mysqld_safe` in older versions of MySQL. For more information, see `mysqld_safe` — MySQL Server Startup Script. (Bug #78475, Bug #21862951)
- The server now supports an `--early-plugin-load` option that indicates which plugins to load before loading mandatory built-in plugins and before storage engine initialization. One use for this option is to load the `keyring_file` plugin: The `InnoDB` storage engine uses the keyring tablespace encryption, so the `keyring_file` plugin must be loaded early to ensure that it is available prior to `InnoDB` initialization.
- Storage engines now can request notification about acquisition and release of exclusive metadata locks. As result, the `LOCK_STATUS` column of the `metadata_locks` Performance Schema table has two new status values. The `PRE_ACQUIRE_NOTIFY` and `POST_RELEASE_NOTIFY` status values are brief and signify that the metadata locking subsubsystem is notifying interested storage engines while entering lock acquisition or leaving lock release operations.
- The `mysql_plugin` utility is deprecated and will be removed in a future version of MySQL. 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 `mysql_kill()`, `mysql_list_fields()`, `mysql_list_processes()`, and `mysql_refresh()` C API functions are deprecated and will be removed in a future version of MySQL. The same is true of the corresponding `COM_PROCESS_KILL`, `COM_FIELD_LIST`, `COM_PROCESS_INFO`, and `COM_REFRESH` client/server protocol commands. Instead, use `mysql_query()` to execute a `KILL`, `SHOW COLUMNS`, `SHOW PROCESSLIST`, or `FLUSH` statement, respectively.

Bugs Fixed

- **InnoDB; Partitioning:** When `OPTIMIZE TABLE` rebuilt a partitioned `InnoDB` table, it placed the resulting partition tablespace files (`*.ibd` files) in the default data directory instead of the directory specified using the `DATA DIRECTORY` option. (Bug #75112, Bug #20160327)
- **InnoDB:** `InnoDB` failed to update index statistics when adding or dropping a virtual column. (Bug #22469660, Bug #79775)
- **InnoDB:** A small `InnoDB` buffer pool size with a large `innodb_stats_persistent_sample_pages` setting resulted in a `Difficult to find free blocks in the buffer pool` warning. (Bug #22385442)
- **InnoDB:** `memcached` connections are blocked from accessing tables that contain an indexed virtual column. Accessing an indexed virtual column requires a callback to the server, but a `memcached` connection does not have access to the server code. (Bug #22384503, Bug #79691)
- **InnoDB:** `InnoDB` did not return an informative message when refusing an online `ALTER TABLE` operation that attempted to add an index and a virtual column. (Bug #22374827)
- **InnoDB:** An invalid `innodb_saved_page_number_debug` setting caused a server exit. (Bug #22311319, Bug #79516)
- **InnoDB:** `InnoDB` failed to free a table object when a compressed table and temporary compressed table were created in the same shared tablespace. (Bug #22306581)

- **InnoDB:** In NUMA-related code, the size information passed to the `mbind()` call in the `buf_chunk_init()` function was incorrect. (Bug #22293530, Bug #79434)
- **InnoDB:** NUMA support was incomplete for online buffer pool resizing operations. (Bug #22293511, Bug #79354)
- **InnoDB:** A `SELECT COUNT(*)` query that counted the results of a full-text subquery raised an assertion. (Bug #22270139)
- **InnoDB:** `InnoDB` passed a buffer with an incorrect `TINYBLOB` data length for a virtual column, resulting in a purge thread failure. (Bug #22256752)
- **InnoDB:** A purge failure occurred while deleting data from a table that contained a spatial index. (Bug #22230442, Bug #22366370)
- **InnoDB:** An assertion was raised when purge accessed a freed page while attempting to rebuild virtual column data from the clustered index. (Bug #22204260)
- **InnoDB:** Only prefix bytes were logged for an indexed virtual column, resulting in an a `Clustered record for sec rec not found` error. (Bug #22202788)
- **InnoDB:** A small buffer pool with an `innodb_page_size` setting of `64K` could cause startup, bootstrap, and recovery failures. (Bug #22179133, Bug #79201)
- **InnoDB:** Unreachable code that checks for 32-bit file offsets was removed. (Bug #22163880, Bug #79150)
- **InnoDB:** A slow shutdown failure was caused by background threads adding undo records to the purge history list during or after purge thread exit. (Bug #22154730)
- **InnoDB:** The `InnoDB` purge thread died attempting to purge a virtual column index record that was not delete-marked. (Bug #22141031)
- **InnoDB:** In debug builds, an `ALTER TABLE` operation that increased the column length of a virtual column raised an assertion. (Bug #22139917)
- **InnoDB:** `ut_allocator` prepended the allocation payload with a 12-byte header on 32-bit systems, causing unaligned memory access. On 32-bit SPARC systems, the unaligned memory access caused a crash during bootstrap. (Bug #22131684)
- **InnoDB:** In debug builds, an `ALTER TABLE` operation that added a new virtual column before an existing virtual column raised an assertion. (Bug #22123674, Bug #22111464)
- **InnoDB:** `InnoDB` startup messages related to dumping and loading of the buffer pool were improved. (Bug #22096661, Bug #78960)
- **InnoDB:** Support was enabled for `ALGORITHM=INPLACE` operations that add an index on an existing virtual column while dropping another virtual column. Support was also enabled for `ALGORITHM=INPLACE` operations that add a virtual column and an index. When adding an index on a newly-added virtual column, purge now skips the uncommitted virtual index. (Bug #22082762)
- **InnoDB:** The wrong table object was used to compute virtual column values for a query that accessed multiple instances of the same table. (Bug #22070021)
- **InnoDB:** A purge thread failure occurred when inserting and deleting spatial data. The child page number field was not stored during the R-tree search stage. (Bug #22027053)
- **InnoDB:** Starting the server with an empty `innodb_data_home_dir` entry in the configuration file caused `InnoDB` to look for the buffer pool file in the root directory, resulting in a startup error. (Bug #22016556, Bug #78831)

- **InnoDB:** A failure to compute virtual column values caused an excessive number of error messages. (Bug #21968375)
- **InnoDB:** An `INFORMATION_SCHEMA.INNODB_CMP_PER_INDEX` query raised an assertion. A dictionary mutex was taken while `InnoDB` populated an in-memory heap table. The mutex was not released before `InnoDB` attempted to convert the in-memory heap table to an optimized internal temporary table. (Bug #21950756, Bug #78714)
- **InnoDB:** To avoid a potential hang and redo log overwrite, the `innodb_log_file_size` minimum value was increased from 1MB to 4MB, and the length calculation in `log_margin_checkpoint_age()` was revised. (Bug #21924224, Bug #78647)
- **InnoDB:** A full-text query run under high concurrency caused a server exit due to an invalid memory access. (Bug #21922532)
- **InnoDB:** An `ALTER TABLE` operation on a table partitioned across multiple tablespaces moved existing partitions to the table's default tablespace, resulting in an assertion on `SHOW CREATE TABLE`. Likewise, `ALTER TABLE tbl_name TABLESPACE tablespace_name` moved existing partitions to the named tablespace. Only `ALTER TABLE ... REORGANIZE PARTITION` should move existing partitions to the table's default tablespace or to a named tablespace. Running `ALTER TABLE tbl_name TABLESPACE tablespace_name` on a table partitioned across multiple tablespaces should only change the table's default tablespace. (Bug #21914047, Bug #22124042, Bug #79030)
- **InnoDB:** With a large `innodb_sort_buffer_size` setting, adding an index on an empty table performed more slowly than expected. (Bug #21762319, Bug #78262)
- **InnoDB:** A race condition occurred between `fil_names_write()` and `file_rename_tablespace_in_mem()`. (Bug #21549928)
- **InnoDB:** Purge attempted to access undo pages that were freed by a preceding undo log truncate operation, resulting in an assertion. (Bug #21508627)
- **InnoDB:** `InnoDB` did not return an informative message when refusing an online `ALTER TABLE` operation on a table with a spatial index. (Bug #20111575)
- **InnoDB:** A compiler barrier was added to `ut_relax_cpu()`. The `ut_always_false` dummy global variable was removed from `ut_delay()`. (Bug #20045167, Bug #74832)
- **InnoDB:** Incorrect index values were returned while dropping a virtual column. The altered table object was used to evaluate virtual column values. (Bug #79773, Bug #22469459)

  References: This issue is a regression of: Bug #22082762.
- **Partitioning:** Subquery scans on partitioned tables with virtual columns could leak memory. (Bug #79145, Bug #22162200)
- **Replication:** The behavior of `SET GTID_PURGED` was not consistent between version 5.6 and 5.7. The fix ensures that version 5.7 does not initiate a transaction for `SET GTID_PURGED` statements. (Bug #21472492)
- **Replication:** When DML invokes a trigger or a stored function that inserts into an `AUTO_INCREMENT` column, that DML has to be marked as an unsafe statement. If the tables are locked in the transaction prior to the DML statement (for example by using `LOCK TABLES`), then the DML statement was not being marked as an unsafe statement. The fix ensures that such DML statements are marked correctly as unsafe. (Bug #17047208)
- **Replication:** If `pseudo_slave_mode` was set to 1 while an XA transaction was in the prepare stage, an assert was generated. The fix ensures that changes from 0 to 1 can be made during XA transactions.

Note that this variable is solely for internal use by the server. (Bug #79416, Bug #22273964, Bug #78695, Bug #21942487)

- **Replication:** When using GTIDs, a GTID-violating statement generates an error in the following cases:

  - `enforce_gtid_consistency=ON`
  - `gtid_mode` is either `ON` or `ON_PERMISSIVE` and `gtid_next=AUTOMATIC`
  - `GTID_NEXT=UUID:NUMBER`

  The error was preventing the implicit pre-commit from happening, which caused a race condition. This has been fixed by making the error not prevent the implicit pre-commit from happening. This matches the expected behavior for GTID-violating DDL statements to implicitly commit the previous transaction before executing it. (Bug #78543, Bug #21895421)

- **Replication:** When a slave was configured with `log_bin=OFF`, the applier (SQL) thread was failing to correctly roll back partial transactions left in the relay log. The fix ensures that on reconnection, the applier thread correctly rolls back a partial transaction and starts applying it again from the next relay log file. (Bug #78211, Bug #21691396)

- **Replication:** If the server stopped unexpectedly immediately before committing an XA transaction which had been prepared, and the transaction modified the `mysql.gtid_executed` table, then the subsequent recovery aborted due to an `innodb_lock_wait_timeout` error when it was reading the `mysql.gtid_executed` table. To fix the problem, XA transactions are no longer permitted to modify the `mysql.gtid_executed` table. (Bug #77740, Bug #21452916)

- **Replication:** As part of the fix for Bug #16290902, when writing a `DROP TEMPORARY TABLE IF EXISTS` query into the binary log, the query is no longer preceded by a `USE `db`` statement. Instead the query uses a fully qualified table name, for example `DROP TEMPORARY TABLE IF EXISTS `db`.`t1`;`. This changed the application of `replicate-rewrite-db` filter rules, as they work only on the default database specified in a `USE` statement. This caused slaves to fail when the resulting `CREATE TEMPORARY TABLE` was applied. The fix ensures that at the time of writing a `DROP TEMPORARY TABLE IF EXISTS` query into the binary log, a check is made for the default database. If it exists then the query is written as `USE default_db` in the binary log. If a default database is not present then the query is logged with the qualified table name. (Bug #77417, Bug #21317739)

- **Replication:** If generating a GTID for a transaction fails, the transaction is not written to the binary log but still gets committed. Although running out of GTIDs is a rare situation, if it did occur an error was written to the binary log as a sync stage error. With `binlog_error_action=ABORT_SERVER`, the server aborts on such an error, avoiding data inconsistency. When `binlog_error_action=IGNORE_ERROR`, the server continues binary logging after such an error, potentially leading to data inconsistency between the master and the slave. The fix changes the error to be correctly logged as a flush stage error. (Bug #77393, Bug #21276561)

- **Replication:** When using `--gtid-mode=on` , `--enforce-gtid-consistency` , and `--binlog-format=row`, if a user defined function with multiple `DROP TEMPORARY TABLE` statements was executed on a master, the resulting binary log caused an error on slaves. The fix ensures that stored functions and triggers are also considered multi-statement transactions, and that when `--enforce-gtid-consistency` is enabled, functions with `CREATE TEMPORARY TABLE` or `DROP TEMPORARY TABLE` statements generate an `ER_GTID_UNSAFE_CREATE_DROP_TEMPORARY_TABLE_IN_TRANSACTION` error. (Bug #77354, Bug #21253415)

- **Replication:** Stored procedure local variables that were used in an `ALTER EVENT` statement were not being replicated correctly. This was related to the fact that `CALL` statements are not written into the binary log. Instead each statement executed in a stored procedure is binary logged separately,

with the exception that the query string is modified so that uses of stored procedure local variables are replaced with `NAME_CONST('spvar_name', 'spvar-value')` calls. DDL statements (which are always binary logged in statement binary log mode irrespective of the current binary log format) can also use stored procedure local variables and a clash could cause them to not be replicated correctly. The fix ensures that any stored procedure local variables used in a query are replaced with `NAME_CONST(...)`, except for the case when it is a DML query and the binary log format is `ROW`. (Bug #77288, Bug #21229951)

- **Replication:** `DROP TABLE` statements are regenerated by the server before being written to the binary log. If a table or database name contained a non-regular character, such as non-latin characters, the regenerated statement was using the wrong name, breaking replication. The fix ensures that in such a case the regenerated name is correctly converted back to the original character set. Also during work on this bug, it was discovered that in the rare case that a table or database name contained 64 characters, the server was throwing an `assert(M_TBLLEN < 128)` assertion. The assertion has been corrected to be less than or equal 128. (Bug #77249, Bug #21205695)

  References: See also: Bug #78036, Bug #22261585, Bug #21619371.

- **Replication:** Irrespective of the current `binlog_format` setting, DDL that changes metadata on a master is always identified and written to the binary log in `STATEMENT` format. Such DDL could occur from event based SQL statements, such as `CREATE EVENT` or `DROP EVENT`, or transactions that had unsafe functions such as `sysdate()`. When `binlog_format=MIXED` and attempting to replicate such DDL, it was not being correctly identified and therefore was not being correctly replicated. (Bug #71859, Bug #19286708)

- The System-V initialization script for RHEL6 or older failed to enable the `mysqld` service by default. (Bug #22600974)

- Some activations of triggers that referenced a `NEW` value inside a query might cause a server exit. (Bug #22377554)

- Parsing the output of `ST_GeometryType()` as a `DATETIME` value with a default character set of `utf32` caused a server exit. (Bug #22340858)

- For a character set loaded from an XML file, the server could fail to properly initialize its state map, leading to a server exit. (Bug #22338946)

- Inserting a token of 84 4-byte characters into a full-text index raised an assertion. The maximum token length was 84 characters up to a maximum of 252 bytes, which did not account for 4-byte characters. The maximum byte length is now 336 bytes. (Bug #22291765, Bug #79475)

- For some combination of consumers, the Performance Schema prepared statement instrumentation could cause a server exit. (Bug #22291560)

- If a client attempted to use an unsupported client character set (`ucs2`, `utf16`, `utf32`), the error message reported to the client differed for SSL and non-SSL connections. (Bug #22216715)

- Data corruption or a server exit could occur if a stored procedure had a variable declared as `TEXT` or `BLOB` and data was copied to that variable using SELECT ... INTO syntax from a `TEXT` or `BLOB` column. (Bug #22203532, Bug #22232332, Bug #21941152)

- For debug builds, with the `ONLY_FULL_GROUP_BY` SQL mode disabled, the optimizer could attempt to sort on outer references, causing an assertion to be raised. (Bug #22200984)

- Different handling of `YEAR` values for `INSERT` and `SELECT` could raise an assertion when such values were used in a generated-column expression. (Bug #22195458)

- For debug builds, for queries involving `MIN()` or `MAX()` on an indexed column and a reference to an unindexed datetime column, the optimizer could attempt to access unread values, causing an assertion to be raised. (Bug #22186926)
- Geohash decoding (for example, for `ST_LongFromGeoHash()`, `ST_LatFromGeoHash()`, and `ST_PointFromGeoHash()`) could yield incorrect results due to the rounding algorithm being too aggressive. (Bug #22165582)
- In debug builds, with `READ UNCOMMITTED` transaction isolation level, a `SELECT` reading a generated column using an index could raise an assertion. (Bug #22133710)
- For generated columns, the optimizer could fail to establish the proper table reference, resulting in a server exit. (Bug #22132822)
- For some combination of consumers, the Performance Schema file instrumentation could fail due to an attempt to use a `NULL` pointer while instrumenting temporary file I/O. (Bug #22130453)
- The Performance Schema could raise an assertion based on the (incorrect) assumption that instrumenting a temporary file open operation always resulted in an instrumented file. (Bug #22118669)
- An `ALTER TABLE` statement that added an index on a virtual generated column using the `INPLACE` algorithm did not properly report warnings (or errors in strict SQL mode) for problems with virtual column values. Any subsequent `ALTER TABLE` on the same table using the `COPY` algorithm produced such warnings (or failures in strict SQL mode) due to evaluating the problematic value, but left the connection in a state that made further `INPLACE` alterations on the table fail for the same reason. (Bug #22095783)
- If the left expression of an `IN` expression was a row subquery that accesses no tables, an assertion could be raised (in debug builds), or incorrect results could be returned (in release builds). (Bug #22089623)
- Expressions that match an indexed generated column may be replaced with the generated column by the optimizer to enable use of the associated index. However, this optimization was not performed for single-table update and delete statements. The optimizer now extends this replacement optimization to such statements. (Bug #22077611)
- `ANSI` SQL mode could cause inconsistencies in processing of generated column expressions. (Bug #22018979)
- Removal of server session plugins was faulty and could cause a server exit. (Bug #21983102)
- For some queries, if the optimizer used Disk-Sweep Multi-Range Read optimization on generated columns, the server could exit. (Bug #21980430)
- `mysqlpump` tries to do as much work in parallel as possible, but the dump threads lacked a synchronization point before backing up the data, resulting in inconsistent backup. `mysqlpump` now locks the server and flushes all the tables using `FLUSH TABLES WITH READ LOCK` to ensure that any further connections view the same state of all the databases.

  This change lifts the restriction aginst the `--single-transaction` option being mutually exclusive with parallelism. When using `--single-transaction`, it is no longer necessary to disable parallelism by setting `--default-parallelism` to 0 and not using any instances of `--parallel-schemas`. (Bug #21980284)
- A fault in `pthread_rwlock_unlock()` wherein it decremented the lock counter even for already unlocked objects could result in deadlock. (Bug #21966621)
- The Performance Schema could acquire a double lock on session system variables, causing a server hang or (in debug builds) an assertion to be raised. (Bug #21935106)

- Certain queries containing `WHERE 0` of the following form could cause a server exit due to uninitialized reads: `SELECT (SELECT col AND constant FROM t WHERE 0) IN (SELECT constant FROM t1)`. (Bug #21922202)
- `CREATE TEMPORARY TABLE .. SELECT` statements involving `BIT` columns that resulted in a column type redefinition could cause a server exit or an improperly created table. (Bug #21902059)
- For `UPDATE` operations on `InnoDB` tables, there could be a mismatch between the value of a virtual generated column in the index and the value in the “before” buffer, resulting in a server exit. (Bug #21875520)
- With `character_set_server=utf16le`, some values of `ft_boolean_syntax` could cause a server exit for full-text searches. (Bug #21631855)
- With `gtid_mode=ON`, concurrent execution of `SHOW TABLE STATUS` and `REVOKE ALL PRIVILEGES` could lead to deadlock in there was a view in the database and `REVOKE ALL PRIVILEGES` failed for some but not all of the named users. (Bug #21463167)
- `mysqlpump` could exit due to incorrect synchronization of view-handling threads during dump processing. (Bug #21399236, Bug #21447753)
- With `LOCK TABLES` in force, an attempt to open a temporary `MERGE` table consisting of a view in its list of tables (not the last table in the list) caused a server exit. (Bug #20691429)
- For certain prepared statements, the optimizer could transform join conditions such that it used a pointer to a temporary table field that was no longer available after the initial execution. Subsequent executions caused a server exit. (Bug #19941403)
- Repeated execution of `ALTER TABLE v1 CHECK PARTITION` as a prepared statement, where `v1` is a view, led to a server exit.

  In addition, output for some administrative operations, when they are attempted on a view, changes from “Corrupt” to “Operation failed”. These include `ANALYZE TABLE`, `OPTIMIZE TABLE`, and `REPAIR TABLE`, and `ALTER TABLE` statements that perform `ANALYZE PARTITION`, `CHECK PARTITION`, `OPTIMIZE PARTITION`, and `REPAIR PARTITION` operations. (Bug #19817021)
- Valgrind detected some possibly unsafe use of string functions in code used for asymmetric encryption. (Bug #19688135)
- An out-of-memory failure in join buffer allocation could lead to incorrect results for multiple-table queries. (Bug #19031409)
- SSL connections ignored any change made by passing the `MYSQL_OPT_READ_TIMEOUT` option to the `mysql_options()` C API function. (Bug #17618162)
- For `ALTER TABLE` statements, the parser did not support the `ALGORITHM` clause for some operations involving tablespaces or partitions. (Bug #17400320)
- Debian packages create the `root` user account using the `auth_socket` authentication plugin to achieve secure-by-default installation if installation was done with a blank `root` password. However, `auth_socket` was being used even if the password was not blank. (Bug #80137, Bug #22594846, Bug #23321113)
- Solaris packages failed to note the dependency of the MySQL client library on the `libstlport` library. (Bug #79778, Bug #22504264)
- Thread handle resource leakage could occur when creating threads for handling connections on Windows, which could lead to Windows servers eventually running out of handles. (Bug #79714, Bug #22455198)

- Using systemd to start `mysqld` failed if configuration files contained multiple `datadir` lines. Now the last `datadir` line is used. (Bug #79613, Bug #22361702)
- A derived table contained in the `SET` clause of an `UPDATE` statement should be materialized to avoid an error about updating a table that is also read in the same statement. Materialization did not occur for some statements, leading to that error. (Bug #79590, Bug #22343301)
- MySQL 5.7.8 prohibited references to select list columns of the outer query from the `HAVING` clause of a correlated subquery in the inner query because they are not permitted by standard SQL. However, because this is a frequently used extension, it is once again permitted. (Bug #79549, Bug #22328395)

  References: This issue is a regression of: Bug #19823076.
- Installing just shared libraries, clients, and development support files failed to install everything needed to build client applications because the `binary_log_types.h` header file was not installed. (Bug #79531, Bug #22321338)
- `SHOW CREATE TRIGGER` could fail to display all applicable SQL modes. (Bug #79526, Bug #22313133)

  References: This issue is a regression of: Bug #18311187.
- On SELinux, `mysqld --initialize` with an `--init-file` option could fail to initialize the data directory. (Bug #79442, Bug #22314098, Bug #79458, Bug #22286481)
- Syntax checks were not always performed when an `ALTER TABLE` statement changed a column's type from `TEXT` to `JSON`. This could lead to `JSON` columns containing invalid JSON data. This issue was observed when the original `TEXT` column used the `utf8mb4_bin` collation. (Bug #79432, Bug #22278524)
- Hexadecimal and bit literals written to saved view definitions could be truncated. This could also affect extended `EXPLAIN` output. (Bug #79398, Bug #22268110)
- `ST_Buffer()` returned an error for geometries with an SRID different from 0. Nonzero SRID values now are permitted but ignored (calculations are still done using Cartesian coordinates). (Bug #79394, Bug #22306745)
- A regression caused failure of the workaround at Restrictions on Subqueries for avoiding `ER_UPDATE_TABLE_USED` errors when referencing the same table in a subquery as was used as target for an `UPDATE` or `DELETE` statement. (Bug #79333, Bug #22239474)
- Statements causing multiple parse errors could cause an assertion to be raised. (Bug #79303, Bug #22222013)
- Some queries with derived tables perform better with materialization than when merged into the outer query. The optimizer no longer uses merging by default for derived tables containing dependent subqueries in the select list. (Bug #79294, Bug #22223202)
- `ALTER USER` and `SET PASSWORD` did not work at server startup when invoked in the file named using the `--init-file` option. (Bug #79277, Bug #22205360)
- When not in strict SQL mode, attempts to implicitly insert `NULL` into a `NOT NULL` column resulted in different behavior depending on whether the table had a trigger. (Bug #79266, Bug #22202665)
- Some replication-only code was not protected with `#ifdef` and failed to compile with the `WITH_UBSAN CMake` option enabled. (Bug #79236, Bug #22190632)
- Configuring MySQL with the `-DWITH_UBSAN=ON CMake` option resulted in spurious runtime warnings from `comp_err`. These are now suppressed. Also, a `CMake` warning was added that undefined behavior address sanitizer support is currently experimental. (Bug #79230, Bug #22190656)

- `INSERT INTO ... SELECT` statements could insert values other than `DEFAULT` into generated columns. (Bug #79204, Bug #22179637)
- With the `derived_merge` flag of the `optimizer_switch` system variable enabled, queries that used a derived table on the inner side of an outer join could return incorrect results. (Bug #79194, Bug #22176604)

  References: See also: Bug #80526, Bug #22833364.
- Memory leaks in `libmysqld` were corrected. (Bug #79187, Bug #22174219)
- `FOUND_ROWS()` could return a negative value if the preceding query was a `UNION` involving `SQL_CALC_FOUND_ROWS` and `LIMIT ... OFFSET`. (Bug #79131, Bug #22155786)
- `IN`-to-`EXISTS` subquery transformation could cause `SELECT NULL IN (subquery)` to return 0 rather than `NULL`.

  `IN`-to-`EXISTS` subquery transformation could yield incorrect results for queries for which the table was nonempty, the subquery on the left side of the `IN` predicate produced an empty result, and semi-join optimization was disabled. (Bug #78946, Bug #22090717, Bug #74403, Bug #19822406)
- The result from `WEIGHT_STRING()` could be incorrect when used in a view. (Bug #78783, Bug #21974321)
- For connections made using a Unix socket file, specifying the `--ssl-ca` option caused the connection to fail. (Bug #78509, Bug #21878661)
- If server was started with `--thread-handling=no-threads`, no foreground thread was created for a client connection. The Performance Schema did not account for the possibility of no foreground threads for queries on the `session_connect_attrs` table, causing an assertion to be raised. (Bug #78292, Bug #21765843)
- `mysqlpump` generated incorrect `INSERT` statements for tables that had generated columns. (Bug #78082, Bug #21650559)
- `ALTER TABLE ... CONVERT TO CHARACTER SET` operations that used the `INPLACE` algorithm were ineffective if the table contained only numeric data types. Also, such operations failed to clean up their temporary `.frm` file. (Bug #77554, Bug #21345391)
- Heavy `SHOW PROCESSLIST` or `SELECT ... FROM INFORMATION_SCHEMA.PROCESSLIST` activity could result in the server accepting more than `max_connections` connections. (Bug #75155, Bug #20201006)
- When used with the `libmysqld` embedded server, the `mysql_stmt_execute()` C API function failed with a `malformed communication packet` error, even for simple prepared statements. (Bug #70664, Bug #17883203)
- Queries using `SUM(DISTINCT)` could produce incorrect results when there were many distinct values. (Bug #56927, Bug #11764126, Bug #79648, Bug #22370382)

Changes in MySQL 5.7.10 (2015-12-07)

- Security Notes
- Functionality Added or Changed
- Bugs Fixed

Security Notes

- Previously, MySQL supported only the TLSv1 protocol for encrypting secure connections. TLS support is now extended to enable a higher level of encrypted connection security:

  - When compiled using OpenSSL 1.0.1 or higher, MySQL supports the TLSv1, TLSv1.1, and TLSv1.2 protocols.

  - When compiled using the bundled version of yaSSL, MySQL supports the TLSv1 and TLSv1.1 protocols.

  Because TLSv1.2 requires OpenSSL, support for this protocol is available in binary distributions only for MySQL Commercial Server, and not for MySQL Community Server (which is compiled using yaSSL). To enable TLSv1.2 support if you build from source, you must set the `WITH_SSL` `CMake` option to use OpenSSL.

  The `tls_version` system variable enables specifying at startup the TLS protocols permitted by the server. On the client side, the `--tls-version` option enables specifying the TLS protocols permitted per client invocation.

  The new `MASTER_TLS_VERSION` option for the `CHANGE MASTER TO` statement specifies the encryption protocols permitted by the master for slave connections.

  The `mysql_options()` C API function has a new `MYSQL_OPT_TLS_VERSION` option that enables specifying from within the client library the TLS protocols permitted by a client program.

  By default, MySQL attempts to use the highest TLS protocol version available, depending on which SSL library was used to compile the server and client, which key size is used, and whether the server or client are restricted from using some protocols; for example, by means of `tls_version`/`--tls-version`.

  For more information, see Secure Connection Protocols and Ciphers.

Functionality Added or Changed

- **InnoDB:** Enabling the new `innodb_background_drop_list_empty` debug configuration option helps avoid test case failures by delaying table creation until the background drop list is empty. (Bug #21891560)

- **InnoDB:** The `innodb_support_xa` system variable, which enables support for two-phase commit in XA transactions, is deprecated. `InnoDB` support for two-phase commit in XA transactions is always enabled as of MySQL 5.7.10. Disabling `innodb_support_xa` is no longer permitted as it makes replication unsafe and prevents performance gains associated with binary log group commit.

- These client programs now support the `--enable-cleartext-plugin` option: `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlshow`. This option enables the `mysql_clear_password` cleartext authentication plugin. (See The Cleartext Client-Side Authentication Plugin.) (Bug #21235226)

- These functions now produce warnings for invalid input: `UNHEX()`, `INET_NTOA()`, `INET_ATON()`, `INET6_NTOA()`, `INET6_ATON()`. Also, `INET_ATON()` now returns `NULL` and produces a warning for arguments that contain more than four groups of digits. (Bug #78856, Bug #22042027, Bug #78884, Bug #22071558)

- MySQL distributions now include `lz4_decompress` and `zlib_decompress` utilities that can be used to decompress `mysqlpump` output that was compressed using the `--compress-output=LZ4` and `--compress-output=ZLIB` options. For more information, see `lz4_decompress` — Decompress mysqlpump LZ4-Compressed Output, and `zlib_decompress` — Decompress mysqlpump ZLIB-Compressed Output. (Bug #78108, Bug #21644479)

- `mysql_upgrade` now attempts to print more informative errors than `FATAL ERROR: Upgrade failed`. (Bug #77803, Bug #21489398)

Bugs Fixed

- **InnoDB:** When attempting to create a cascading foreign key constraint on a primary key column used in a virtual index, the error message that was returned did not include information from `dict_foreign_has_col_in_v_index()`. (Bug #22050059)
- **InnoDB:** An in-place operation that rebuilt a table with multiple indexed virtual columns raised an assertion. (Bug #22018532)
- **InnoDB:** A `SELECT ... FOR UPDATE` operation on a table that only contained virtual columns and a virtual column index raised an invalid assertion. (Bug #21981164, Bug #21880930)
- **InnoDB:** The calculation in `dict_index_node_ptr_max_size()` that determines the maximum index record size treated fixed fields that exceeded the maximum column length as fixed-size fields instead of variable length fields. (Bug #21968191)
- **InnoDB:** A server exit during a `TRUNCATE TABLE` operation on a table with a full-text index caused startup to fail. (Bug #21959479)
- **InnoDB:** An `ALTER TABLE ... DISCARD TABLESPACE` operation raised an invalid assertion. (Bug #21957001, Bug #78728)
- **InnoDB:** Compared to previous releases, small tablespaces containing tables with `BLOB` values had larger data files due to regression introduced in MySQL 5.7.5. (Bug #21950389, Bug #78623)

  References: This issue is a regression of: Bug #18756233.
- **InnoDB:** The `ord_part` flag was not reset after a failed operation to create an index on a virtual column, causing `InnoDB` to assert on a subsequent insert operation. (Bug #21941320, Bug #21979969)
- **InnoDB:** An invalid search tuple was created for a table with virtual columns. (Bug #21922176)
- **InnoDB:** `InnoDB` attempted to purge a virtual column index record that was not marked for deletion. (Bug #21901389)
- **InnoDB:** Successive open table operations on tables with virtual columns caused a memory access violation. (Bug #21894654)
- **InnoDB:** Update vector generation for a full-text search `DOC_ID` did not initialize a new `ufield`, resulting in an assertion. (Bug #21891185)
- **InnoDB:** `InnoDB` failed to prevent foreign key `SET NULL` or `CASCADE` constraints on virtual column base columns and virtual index columns. (Bug #21890816)
- **InnoDB:** An invalid comparison raised an assertion under a 64k `innodb_page_size` setting. (Bug #21882024, Bug #78516)
- **InnoDB:** When estimating the maximum record size of a b-tree index page, `InnoDB` incorrectly treated long-length fixed fields (greater than 767 bytes) as fixed-size fields instead of variable length fields. (Bug #21880445)
- **InnoDB:** In debug builds, `dtuple_get_n_fields` attempted to read freed memory that was previously allocated for a virtual column tuple, resulting in a server exit. (Bug #21875974)
- **InnoDB:** Selecting from `INNODB_CMPMEM` with option `big-tables=1` raised a buffer pool mutex assertion. (Bug #21871451, Bug #78494)

- **InnoDB:** `InnoDB` returned an invalid corruption-related error message during an `IMPORT TABLESPACE` operation. (Bug #21838158, Bug #77321)
- **InnoDB:** After a crash on Windows, copying the data directory to a non-Windows platform to perform the restore caused a crash recovery failure on startup. The code did not convert file path separators from "\" to "/" in the redo log. (Bug #21825127, Bug #78410)
- **InnoDB:** A `FLUSH TABLE ... FOR EXPORT` operation asserted in `row_quiesce_table_start()` when run on a partitioned table with partitions residing in a system or general tablespace. (Bug #21796845)
- **InnoDB:** A `DROP TABLE` operation resulted in a server exit. The return value of a function call was not checked, which lead to dereferencing of a null pointer. The patch for this bug also addresses a potential race condition. (Bug #21794102, Bug #78336)
- **InnoDB:** In debug builds, an ordered scan across multiple partitions did not use a priority queue, resulting in an assertion. (Bug #21753477)
- **InnoDB:** After restarting the server, the `COMPRESSION` column of the `INNODB_SYS_TABLESPACES` table displayed incorrect data. The `COMPRESSION` column was removed from `INNODB_SYS_TABLESPACES`. To view the current setting for page compression, use `SHOW CREATE TABLE`. (Bug #21687636, Bug #78197)
- **InnoDB:** An invalid table flags value raised an assertion. The `SYS_TABLES.MIX_LEN(flags2)` value was not read for tables that use `ROW_FORMAT=REDUNDANT`. (Bug #21644827)
- **InnoDB:** An assertion was raised when crash recovery handling of an `MLOG_TRUNCATE` redo log record treated a shared tablespace as a file-per-table tablespace. Redo was skipped for the shared tablespace. (Bug #21606676)
- **InnoDB:** Transaction rollback after recovery failed due to an invalid assertion. (Bug #21575121)
- **InnoDB:** Insufficient information in the undo log about spatial columns raised an assertion and could result in a upgrade failure. Consequently, a slow shutdown is required prior to performing an in-place upgrade from MySQL 5.7.8 or 5.7.9 to 5.7.10 or higher. For more information, refer to Changes Affecting Upgrades to MySQL 5.7. (Bug #21508582)

  References: This issue is a regression of: Bug #21340268.
- **InnoDB:** A secondary index is not permitted on a virtual column that is based on a foreign key-referenced column that uses `ON DELETE CASCADE`, `ON DELETE SET NULL`, `ON UPDATE CASCADE`, or `ON UPDATE SET NULL`. The restriction was not enforced. (Bug #21508402, Bug #77843)
- **InnoDB:** A duplicate key error that occurred during an online DDL operation reported an incorrect key name. (Bug #21364096, Bug #77572)
- **InnoDB:** An `ALTER TABLE` operation caused the server to exit on disk full. (Bug #21326304, Bug #77497)
- **InnoDB:** The system tablespace data file did not extend automatically when reaching the file size limit, causing startup to fail with a size mismatch error and preventing the addition of another system tablespace data file. (Bug #21287796, Bug #77128)
- **InnoDB:** Altering the letter case of a column introduced an inconsistency between the `frm` file and data dictionary resulting in a failed `CREATE INDEX` operation on the altered column. (Bug #20755615)
- **InnoDB:** An `ALTER TABLE` operation that converted a table to an `InnoDB` file-per-table tablespace did not check for unknown files with the same name as the destination `.idb` file, permitting an unknown file of the same name to be overwritten. (Bug #19218794, Bug #73225)

- **InnoDB:** `row_merge_read_clustered_index()` did not handle a bulk load error correctly. (Bug #19163625)
- **Partitioning:** Partition scans did not evaluate virtual generated columns properly. This could cause issues with partitioned tables having an index on a virtual `BLOB` column. (Bug #21864838, Bug #21881155)
- **Partitioning:** While executing `CHECK TABLE`, when checking whether rows were in the correct partition, the partition engine missed updates for virtual generated columns. (Bug #21779554)
- **Partitioning:** Performing an in-place `ALTER TABLE` on a partitioned `InnoDB` table having one or more partitions which used a separate tablespace could cause the server to fail. (Bug #21755994)
- **Partitioning:** When all partitions were pruned, they were not initialized for scanning during initialization of indexes. This involved two related issues, one being that the active index was not set back to the maxmimum key value when the index was closed. In addition, when this occurred as part of a multi-range read, there were attempts to access unset variables. (Bug #78260, Bug #21754608, Bug #21620577)
- **Replication:** On a multi-threaded slave configured with `master_info_repository=TABLE` and `relay_log_info_repository=TABLE` which had previously been run with `autocommit=1`, if the slave was stopped and `autocommit` changed to 0, executing `START SLAVE` caused the session to appear to hang. After the lock wait timeout, when `START SLAVE` proceeded the server would stop unexpectedly. The fix ensures that when `master_info_repository=TABLE`, `relay_log_info_repository=TABLE`, and `autocommit=0` a new transaction is generated for start and commit to avoid deadlocks. (Bug #21440793)
- **Replication:** Fatal errors encountered during flushing or synchronizing the binary log were being ignored. Such errors are now caught and handled depending on the setting of `binlog_error_action`. (Bug #76795, Bug #68953, Bug #20938915, Bug #16666407)
- If the server was started with `--performance_schema_accounts_size=0`, querying the Performance Schema status variable tables caused a server exit. (Bug #22131713)
- For debug builds, using `ALTER TABLE` to add a generated column to a table could cause a deadlock. (Bug #22083048)
- The systemd unit file did not specify any `--pid-file` option for `mysqld`, with the result that server startup could fail. The unit file now includes a default `--pid-file` option in the `ExecStart` value. This default can be overridden in the `override.conf` file by changing both `PIDFile` and `ExecStart` to specify the PID file path name. (Bug #22066787)
- A query with nested derived tables and scalar subqueries in the select list of the derived tables might in some cases cause a server exit. (Bug #22062023)
- When `mysqld` was run with `--initialize`, it used `chown()` to set the data directory owner, even if ownership was already correct. This caused problems for AppArmor and SELinux. The server now checks whether the data directory owner is correct and skips the `chown()` call if so. (Bug #22041387)
- Failed evaluation of a generated column expression for `CREATE TABLE` or `ALTER TABLE` could cause a server exit. Now if expression evaluation causes truncation or provides incorrect input to a function, the statement terminates with an error and the DDL operation is rejected. (Bug #22018999)
- Creating a unique index on a virtual `POINT` column could result in an error or assertion for later table accesses. (Bug #22017616)
- Sending a load spike to a newly started server could cause the Performance Schema to allocate a large amount of memory, possibly leading to out-of-memory failure. (Bug #22006088)

- A missing error check could result in a server exit for `DELETE` statements that referred to user-defined variables. (Bug #21982313)
- Possible buffer overflow from incorrect use of `strcpy()` and `sprintf()` was corrected. (Bug #21973610)
- MySQL RPM packages for RHEL5 failed to create the `mysql` system user. (Bug #21950975)
- MySQL does not support columns of `ROW` type, but the server did not prevent generated columns from being created that used `ROW` expressions. These are now prohibited. (Bug #21940542)
- The `version_tokens` plugin called the locking service using a timeout value of only one second. The timeout is now taken from the default value of the `lock_wait_timeout` system variable (that is, one year). (Bug #21928198)
- Spatial functions could return invalid results if given a polygon or multipolygon argument containing holes such that a hole vertex touched the exterior ring at a point lying in the interior of an exterior ring segment. This could manifest itself as: `ST_UNION()` producing an invalid polygon; `ST_SymDifference()` producing an invalid multipolygon; `ST_Intersection()` producing an invalid self-intersecting polygon; `ST_Difference()` producing an invalid geometry. (Bug #21927733, Bug #21927639, Bug #21927558, Bug #21977775)
- With the `STRICT_TRANS_TABLES` SQL mode enabled, it was not possible to insert data into a `VIRTUAL` generated column defined with the `NOT NULL` attribute. (Bug #21927469)
- Problems leading to Valgrind warnings for OpenSSL random number generation were corrected. (Bug #21927436)
- Querying views on Windows could lead to memory leaks. (Bug #21908206)

  References: This issue is a regression of: Bug #13901905.
- Generated column definitions specified with the `NULL` attribute resulted in a syntax error. (Bug #21900170)
- A stored procedure that used `ST_Area()` could return different numbers of rows for the first and second executions. (Bug #21889842)
- For polygon values with an interior ring that touches an exterior ring, `ST_Buffer()` could return invalid polygon values. (Bug #21871856)
- Two rows in the `threads` Performance Schema table could have the same `THREAD_OS_ID` value. (Bug #21865330)
- For debug builds, using `ALTER TABLE` to change the expression for a generated column could cause a server exit. (Bug #21854004)
- `ALTER USER` failed if the server was started with `--skip-grant-tables`. (Bug #21847825)
- Performance Schema reads of a session's `THD` structure while the session was running could create race conditions and result in a server exit. (Bug #21841412)
- A prepared statement that computes `ST_IsSimple()` or `ST_Buffer_Strategy()` on a nullable column in an outer join could return different numbers of rows for the first and second statement executions. (Bug #21841051)
- Queries on the `variables_by_thread` Performance Schema table could cause a server exit when examining the system variables of a new connection. (Bug #21840950)

- Spatial functions could read already freed memory. (Bug #21823135)
- For Debian package control files, `libnuma-dev` was added to `Build-Depends` to enable NUMA support. (Bug #21822631)
- Selecting `DECIMAL` values into user-defined variables could cause a server exit. (Bug #21819304)
- Re-evaluation of a generated column expression could cause access to previously freed memory and a server exit. (Bug #21810529)
- `ST_SymDifference()` could raise an assertion for polygons with self-intersection points. (Bug #21767301, Bug #79031, Bug #22124757)
- `USER` field output from the audit log plugin was malformed. (Bug #21766380)
- A server exit could occur for queries for which a) a `GROUP BY` included primary key and secondary key columns; and b) the `WHERE` clause included an equality predicate on the first primary key column where that column was constant. (Bug #21761044)
- Building MySQL using parallel compilation sometimes failed with an attempt to compile `sql_yacc.yy` before `lex_token.h` had been created. (Bug #21680733)
- With binary logging enabled, issuing `DROP TEMPORARY TABLE` when in XA_IDLE state caused an assertion to be raised. Now an `ER_XAER_RMFAIL` error is returned.

  In consquence of the fix for this issue, statements that previously succeeded in XA_IDLE state now fail with an `ER_XAER_RMFAIL` error. When running with `--gtid-mode=ON`, an explicit `DROP` continues to fail with `ER_GTID_UNSAFE_CREATE_DROP_TEMPORARY_TABLE_IN_TRANSACTION`. (Bug #21638823)
- A query with a subquery in the left-hand part of an `IN` subquery that was transformed into a semi-join might cause a server exit. (Bug #21606400)
- Concurrent `FLUSH PRIVILEGES` and `REVOKE` or `GRANT` statements could produce a small time window during which invalid memory access to proxy user information could occur, leading to a server exit. (Bug #21602056)
- Using `WITH ROLLUP` within a subquery could cause a server exit. (Bug #21575790)
- For debug builds, a call to `MAKE_SET()` with a subquery as argument might be evaluated before tables were locked, causing an assertion to be raised. (Bug #21547779)
- Starting the server with the `query_alloc_block_size` system variable set to certain negative values on a machine without enough memory could result in out-of-memory errors. (Bug #21503595)
- Using `UNINSTALL PLUGIN` to uninstall the `daemon_example` plugin could cause a server exit. (Bug #21467458)
- `FLUSH DES_KEY_FILE` failed to reload the DES key file. (Bug #21370329)
- If an error occurred during the setup phase of subquery materialization used to compute an `IN` predicate, cleanup of the temporary table did not happen, leading to Valgrind errors. (Bug #21346081)
- On Windows, the sysbench benchmark tool's "run" command would hang when attempting to create multiple tables for the OLTP test when using shared memory connections. (Bug #21319192, Bug #77481)
- Queries rejected by MySQL Enterprise Firewall were truncated to 512 characters when written to the error log. (Bug #20948270)

- A server exit could occur for the second execution of a prepared statement for which an `ORDER BY` clause referred to a column position. (Bug #20755389)
- Repeated execution of a prepared statement could cause a server exit if the default database was changed. (Bug #20447262)
- `mysql_plugin` could exit due to improper checking of string operation operands. (Bug #20376670)
- After failure to create a temporary table during join processing and releasing the table descriptor, an attempt to access the now-invalid descriptor could cause a server exit. (Bug #19918299)
- Type conversion failure for `DECIMAL` values could cause a server exit. (Bug #19912326, Bug #20013538)
- `INSERT DELAYED` could cause a server exit for tables partitioned with a character column as the key and for which the expression required a character set conversion. (Bug #19894161)
- A server exit could occur when updating a view using an `ALL` comparison operator on a subquery that selects from an indexed column in the main table. (Bug #19434916)
- With AddressSanitizer (ASAN) enabled, triggers that contained null or invalid characters could cause an ASAN server exit. (Bug #18831513)
- Incorrect error checking for the `NAME_CONST()` function could lead to a server exit. (Bug #17733850)
- On SELinux, `mysqld --initialize` with an `--init-file` option could fail to initialize the data directory. Although fixed in 5.7.11, the Fedora 23 and EL6 5.7.10 RPM's were also updated with the fix; as a *-5.7.10-2-*.rpm release to the Yum repository. (Bug #79442, Bug #22314098, Bug #22286481)
- `INSERT ... ON DUPLICATE KEY UPDATE` could result in a memory leak when executed as a prepared statement. (Bug #79122, Bug #22151233)

  References: This issue is a regression of: Bug #21908206.
- Queries that needed to store the result of `ST_AsWKB()` in a temporary table could fail with an error message. (Bug #79060, Bug #22131961)

  References: This issue is a regression of: Bug #21614368.
- If `mysqld` was started with the `--help` option, it created a binary log index file. If that file was located in the data directory and the command preceded data directory initialization, initialization then failed due to a nonempty data directory. (Bug #78986, Bug #22107047)
- Some of the source files for spatial functions in the `sql` directory took excessive compile time and required too much compiler memory allocation. (Bug #78900, Bug #22078874)
- Internal buffer sizes in `resolve_stack_dump` were increased to accommodate larger symbol space requirements for C++ code. (Bug #78885, Bug #22071592)
- Problems leading to Valgrind warnings for `libmysqld` were corrected. (Bug #78819, Bug #22007587)
- MySQL development RPM packages could fail to install if MySQL Connector/C development RPM packages were installed. (Bug #78815, Bug #22005375)
- `mysqladmin --help` displayed the `old-password` command, even though the command itself was removed in MySQL 5.7.5. (Bug #78774, Bug #21972941)
- The `filename` character set is intended for internal use, but references to it in SQL statements did not produce an error. Now they do. (Bug #78732, Bug #21958734)

- If a generated column used an expression that is affected by the SQL mode, the expression could produce different results for the same input values, depending on the current SQL mode. (For example, interpretation of the `||` operator depends on the `PIPES_AS_CONCAT` SQL mode.) Now expression evaluation uses the SQL mode in effect at the time the column is defined. (Bug #78665, Bug #21929967)

- Casting large hexadecimal values could produce an incorrect result and no truncation warning. (Bug #78641, Bug #21922414)

- `mysqlpump` generated incorrect `ALTER TABLE` statements for adding foreign keys. (Bug #78593, Bug #21907297)

- The error message returned when trying to define a `BLOB`, `TEXT`, `JSON`, or `GEOMETRY` column with a default value (Error 1101, `ER_BLOB_CANT_HAVE_DEFAULT`) referred to `BLOB` and `TEXT` columns only. The same error applies to any of these four types when trying to use the `DEFAULT` option with it in a column definition; the corresponding error message now makes this clear by referring to `JSON` and `GEOMETRY` columns as well. (Bug #78527, Bug #21887035)

- A query using `JSON_EXTRACT()` returned the wrong result after a virtual index was added to the table. (Bug #78464, Bug #21854241)

- Executing `HELP` statements or statements involving the `CONVERT_TZ()` function could lead to a memory leak and to `MyISAM` reference-count errors at server shutdown. (Bug #78443, Bug #21840241)

- MySQL did not recognize functional dependencies from base columns in a generated column expression to the generated column. (Bug #78377, Bug #21807579)

- It was possible to store non-ASCII data in columns intended to store data of character set `ascii`. (Bug #78276, Bug #21774967)

- Adding a `SPATIAL` index to a `MyISAM` table could cause the cardinality of other indexes to become incorrect. (Bug #78213, Bug #21789000)

- MySQL could fail to compile on Solaris 11.3 when `/usr/gnu/bin/as` was used as the linker. (Bug #77797, Bug #21484716)

- Some punctuation characters in the `armscii8` character set are represented by two encodings, with the result that a character stored using one encoding would not be found using a search with the other encoding. For such characters, MySQL now selects the encoding with the lowest value to consistently map instances onto the same encoding. (Bug #77713, Bug #21441405)

- `Item_copy_decimal::copy()` did not take the `div_precision_increment` system variable value into account, resulting in `DECIMAL` values being returned with incorrect precision from some queries. (Bug #77634, Bug #21462523)

- For queries with implicit grouping; an index with a string column as its first part; and a `WHERE` clause with an equality comparison comparing the column to a string with trailing characters in addition to the column value, an aggregate function that should return `NULL` returned non-`NULL`. (Bug #77480, Bug #21318711)

- `LOAD_FILE()` could cause a server exit for some pathnames if the character set was `cp932`. (Bug #76555, Bug #20819220)

  References: See also: Bug #51893.

- For constructs such as `ORDER BY` *`numeric_expr`* `COLLATE` *`collation_name`*, the character set of the expression was treated as `latin1`, which resulted in an error if the collation specified after `COLLATE` is incompatible with `latin1`. Now when a numeric expression is implicitly cast to a character expression

in the presence of `COLLATE`, the character set used is the one associated with the named collation. (Bug #73858, Bug #20425399)

Changes in MySQL 5.7.9 (2015-10-21, General Availability)

- Audit Log Notes
- C API Notes
- Compilation Notes
- Packaging Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- sys Schema Notes
- Functionality Added or Changed
- Bugs Fixed

Audit Log Notes

- Some events cannot be terminated. Previously, if an audit plugin returned nonzero status for a nonterminable event, the server ignored the status and continued processing the event. However, if an audit plugin used the `my_message()` function to terminate a nonterminable event, a server exit occurred. Now the server correctly handles termination of nonterminable events using `my_message()`. (Bug #21458066)

C API Notes

- **Incompatible Change:** The `mysql_options()` C API function has two new options, `MYSQL_OPT_MAX_ALLOWED_PACKET` and `MYSQL_OPT_NET_BUFFER_LENGTH`, that set the `max_allowed_packet` and `net_buffer_length` system variables, respectively. Each option name also now can be passed to the `mysql_get_option()` C API function to retrieve its value. For more information, see mysql_options(), and mysql_get_option().

  The (undocumented) `mysql_get_parameters()` function has been removed. Applications that attempt to use it will get link errors and should be modified to use `mysql_options()` and `mysql_get_option()` instead.

  One affected application is DBD::mysql, the MySQL driver for the Perl DBI. Upgrade to DBD::mysql 4.033 or higher, which includes a fix for the C API change just described. (Bug #20821550)

  References: See also: Bug #20686665.

- Previously, it was necessary to call `mysql_thread_end()` for each `mysql_thread_init()` call to avoid a memory leak. C API internals have been reimplemented to reduce the amount of information allocated by `mysql_thread_init()` that must be freed by `mysql_thread_end()`:

  - For release/production builds without debugging support enabled, `mysql_thread_end()` need not be called.

- For debug builds, `mysql_thread_init()` allocates debugging information for the DBUG package (see The DBUG Package). `mysql_thread_end()` must be called for each `mysql_thread_init()` call to avoid a memory leak.

(Bug #20621281, Bug #21802367)

Compilation Notes

- Support for building using Microsoft Visual Studio 2015 was added. Changes include using the native (added in VS 2015) timespec library if it exists, renamed lfind/lsearch and timezone/tzname to avoid redefinition problems, set `TMPDIR` to `""` by default as `P_tmpdir` no longer exists, deprecated `std::hash_map` in favor of `std::unordered_map`, and added Wix Toolset 3.10 support. (Bug #21657078)

- MySQL distributions now contain a `mysqlclient.pc` file that provides information about MySQL configuration for use by the `pkg-config` command. This enables `pkg-config` to be used as an alternative to `mysql_config` for obtaining information such as compiler flags or link libraries required to compile MySQL applications. For more information, see Building C API Client Programs Using pkg-config.

  A new `INSTALL_PKGCONFIGDIR` `CMake` option is available to specify the directory in which to install the `mysqlclient.pc` file. The default value is `INSTALL_LIBDIR/pkgconfig`, unless `INSTALL_LIBDIR` ends with `/mysql`, in which case that is removed first. (Bug #76131, Bug #20637746)

Packaging Notes

- The shell and Perl versions of `mysql_install_db` have been removed from MySQL distributions. The executable C++ version of `mysql_install_db` implemented in MySQL 5.7.5 is still present, but remains deprecated (use `mysqld --initialize` instead) and will be removed in a future MySQL release. (Bug #21625471)

- The deprecated `_r` versions of the `libmysqlclient` libraries are no longer installed. (Bug #21311067)

Performance Schema Notes

- Two changes were made regarding the effect of `show_compatibility_56`:

  - Previously, when `show_compatibility_56=OFF`, selecting from the following `INFORMATION_SCHEMA` system and status variable tables returned an empty result and a deprecation warning:

    ```
    INFORMATION_SCHEMA.GLOBAL_VARIABLES
    INFORMATION_SCHEMA.SESSION_VARIABLES
    INFORMATION_SCHEMA.GLOBAL_STATUS
    INFORMATION_SCHEMA.SESSION_STATUS
    ```

    This caused confusion for applications that were not aware that such selects could be empty: An empty result and a warning was not sufficient notice to signal the need to migrate to the corresponding Performance Schema system and status variable tables.

    To address this issue, selecting from the `INFORMATION_SCHEMA` system and status tables now produces an error, to make it more evident that an application is operating under conditions that require modification, as well as where the problem lies. The error code is `ER_FEATURE_DISABLED_SEE_DOC`. The error message indicates which table is disabled and that the `show_compatibility_56` documentation should be consulted.

- Previously, when `show_compatibility_56=ON`, selecting from the following Performance Schema status variable tables returned an empty result:

  ```
  performance_schema.global_status
  performance_schema.session_status
  ```

  This made it more difficult to migrate applications from the `INFORMATION_SCHEMA` status variable tables to the corresponding Performance Schema tables: Successfully selecting from the Performance Schema tables required knowing both that the server is from MySQL 5.7 and that `show_compatibility_56=OFF`.

  To address this issue, selecting from the Performance Schema status variable tables now produces the same result regardless of the value of `show_compatibility_56`. Thus, it is necessary to know only that the server is from MySQL 5.7.9 or higher. (If so, select from the Performance Schema tables. Otherwise, select from the `INFORMATION_SCHEMA` tables.)

  For additional information about the effects of `show_compatibility_56` and migration issues, see Server System Variables, and Migrating to Performance Schema System and Status Variable Tables. (Bug #21606701)

- These Performance Schema tables now are world readable and accessible without the `SELECT` privilege: `global_variables`, `session_variables`, `global_status`, and `session_status`. An implication of this change is that `SHOW VARIABLES` and `SHOW STATUS` no longer require privileges on the underlying Performance Schema tables from which their output is produced when `show_compatibility_56=OFF`. (Bug #21251297)

- With the `show_compatibility_56` system variable enabled, the reported values of the `Last_query_cost` and `Last_query_partial_plans` status variables were incorrect.

  With the `show_compatibility_56` system variable disabled, the reported values of the `Created_tmp_tables`, `Handler_external_lock`, and `Table_open_cache_%` status variables were incorrect. (Bug #20483278, Bug #21788549, Bug #21788887)

- Previously, the `transaction` instrument in the `setup_instruments` table was disabled by default, and the `events_transactions_current` and `events_transactions_history` consumers in the `setup_consumers` table were enabled by default. This setup is inconsistent, and having the consumers enabled could lead to the impression that transactions were instrumented by default. Now, the consumers are also disabled by default. To monitor transactions, enable the instrument and the applicable consumers. (Bug #78311, Bug #21780891)

- With the `show_compatibility_56` system variable disabled, `SHOW VARIABLES` and `SHOW STATUS` statements failed if MySQL was compiled without Performance Schema support. Consequently, it is no longer possible to compile without the Performance Schema. If it is desired to compile without particular types of instrumentation, that can be done with the following `CMake` options:

  ```
  DISABLE_PSI_COND
  DISABLE_PSI_FILE
  DISABLE_PSI_IDLE
  DISABLE_PSI_MEMORY
  DISABLE_PSI_METADATA
  DISABLE_PSI_MUTEX
  DISABLE_PSI_PS
  DISABLE_PSI_RWLOCK
  DISABLE_PSI_SOCKET
  DISABLE_PSI_SP
  DISABLE_PSI_STAGE
  ```

```
DISABLE_PSI_STATEMENT
DISABLE_PSI_STATEMENT_DIGEST
DISABLE_PSI_TABLE
DISABLE_PSI_THREAD
DISABLE_PSI_TRANSACTION
```

For example, to compile without mutex instrumentation, configure MySQL using the `-DDISABLE_PSI_MUTEX=1` option. (Bug #78159, Bug #21669500)

- The `session_account_connect_attrs` Performance Schema table had mistakenly been changed to require the `SELECT` privilege. It requires no special privileges again. (Bug #77702, Bug #21436364)

  References: This issue is a regression of: Bug #14569746.

- In the `setup_timers` table, a `CYCLE` timer for ARM64 platforms is now available. (Bug #77620, Bug #21374923)

- The Performance Schema now includes these instruments for monitoring I/O on binary log and relay log cache files:

  ```
  wait/io/file/sql/binlog_cache
  wait/io/file/sql/binlog_index_cache
  wait/io/file/sql/relaylog_cache
  wait/io/file/sql/relaylog_index_cache
  ```

  In addition, the default value of `performance_schema_max_file_classes` has been increased from 50 to 80. (Bug #76225, Bug #20675180)

- The Performance Schema `threads` table now contains a `THREAD_OS_ID` column that indicates the thread or task identifier as defined by the underlying operating system. For example, the column value corresponds to the Process Explorer thread ID on Windows and the `gettid()` value on Linux. For more information, see The threads 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` database.

Plugin Notes

- The audit plugin API has been extensively revised to support a finer breakdown of the general event type (`MYSQL_AUDIT_GENERAL_CLASS`) into more specific events. This enables audit plugins to more precisely indicate the types of events in which they are interested and reduces overhead for plugins that have use for only a few event types. The API also now permits early termination of event execution. For more information, see Writing Audit Plugins. The general event type is still available but is deprecated and will be removed in a future MySQL release.

  In addition, a `security_context` plugin service is now available. Audit plugins can use this service to examine or modify the security context of threads associated with audited events. See MySQL Services for Plugins.

Security Notes

- For RPM-based packages, the permissions used to create the data directory (`/var/lib/mysql`) have been changed from 755 to 751. This tightens the data directory permissions while still permitting world access to the `mysql.sock` file in that directory. (Bug #21066592)

Spatial Data Support

- The required version of the Boost library for server builds has been raised from 1.58.0 to 1.59.0. (Bug #77960, Bug #21567456)

- Spatial functions such as `ST_MPointFromText()` and `ST_GeomFromText()` that accept WKT-format representations of `MultiPoint` values now permit individual points within values to be surrounded by parentheses. For example, both of the following function calls are valid, whereas previously the second one produced an error:

```
ST_MPointFromText('MULTIPOINT (1 1, 2 2, 3 3)')
ST_MPointFromText('MULTIPOINT ((1 1), (2 2), (3 3))')
```

  In addition, functions such as `ST_AsText()` and `ST_AsWKT()` that produce WKT-format results now display `MultiPoint` values with parentheses surrounding each point. (Bug #54065, Bug #11761559)

sys Schema Notes

- The `sys` schema included in MySQL distributions was updated to version 1.5.0. This version includes new features:

  - A new `diagnostics()` stored procedure enables DBAs and other support personnel to collect diagnostic information for investigating MySQL instance performance. A new `metrics` view and `statement_performance_analyzer()` stored procedure provide supporting infrastructure for the `diagnostics()` procedure.

  - The following `sys` schema views now provide progress reporting for long-running transactions:

```
processlist
session
x$processlist
x$session
```

    The `progress` column of these views shows the percentage of work completed for stages that support progress reporting. For more information, see sys Schema Progress Reporting.

  - `sys` schema objects now have a `DEFINER` of `'mysql.sys'@'localhost'`. (Previously, the `DEFINER` was `'root'@'localhost'`.) Use of the dedicated `mysql.sys` account avoids problems that occur if a DBA renames or removes the `root` account.

  `sys` schema 1.5.0 also includes fixes for several issues:

  - The `sys` schema `ps_is_instrument_default_enabled()` and `ps_is_instrument_default_timed()` stored functions returned incorrect results in some cases.

  - The `ENABLED` and `HISTORY` columns that were added to the `setup_actors` Performance Schema table in earlier MySQL 5.7 releases caused the `sys` schema `ps_setup_reset_to_default()` stored procedure not to work.

  - Handing of event-timing information in the `sys` schema was updated to handle changes to Performance Schema event-timing columns in MySQL 5.7.8.

  - `mysql_upgrade` previously checked for an exact object-count value in the `sys` schema to determine whether an upgrade was needed. If local objects had been added, the resulting reinstallation removed those objects. Now it checks for at least the expected number of objects.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` to incorporate these changes into the `sys` schema.

Thanks to Daniël van Eeden, Jesper Wisborg Krogh, Shlomi Noach, and Morgan Tocker for their contributions to this update. (Bug #78115, Bug #21647101, Bug #77927, Bug #21550271, Bug #78720, Bug #21966366)

Functionality Added or Changed

- **Incompatible Change; InnoDB:** To better manage redo log format changes, the redo log header of the first redo log file (`ib_logfile0`) now includes a format version identifier and a text string that identifies the MySQL version that created the redo log files.

  A new boolean configuration option, `innodb_log_checksums`, replaces the `innodb_log_checksum_algorithm` option. `innodb_log_checksums=ON` enables a `CRC-32C` checksum, making it the only supported checksum for redo log pages.

  This patch also removes unused fields from the redo log header and checkpoint pages.

  Due to redo log format changes introduced by this patch, upgrading to or downgrading from MySQL 5.7.9 and higher requires a clean shutdown and, in some cases, removal of existing redo log files. For instructions related to this change, see Changes Affecting Upgrades to MySQL 5.7, and Changes Affecting Downgrades from MySQL 5.7. (Bug #21759424, Bug #78275)

- **Important Change; InnoDB:** `DYNAMIC` replaces `COMPACT` as the implicit default row format for `InnoDB` tables. A new configuration option, `innodb_default_row_format`, specifies the default `InnoDB` row format. Permitted values include `DYNAMIC` (the default), `COMPACT`, and `REDUNDANT`.

  The `COMPACT` row format remained the default row format until this release to ensure compatibility with older versions of `InnoDB` in MySQL 5.1 and earlier. Now that MySQL 5.1 has reached the end of its product lifecycle, the newer `DYNAMIC` row format becomes the default. For information about advantages of the `DYNAMIC` row format, see DYNAMIC and COMPRESSED Row Formats.

  Newly created tables use the row format defined by `innodb_default_row_format` when a `ROW_FORMAT` option is not specified explicitly or when `ROW_FORMAT=DEFAULT` is used.

  Existing tables retain their current row format if a `ROW_FORMAT` option was specified explicitly. If a `ROW_FORMAT` option was not specified explicitly or if `ROW_FORMAT=DEFAULT` was used, any operation that rebuilds a table also silently changes the row format of the table to the format defined by `innodb_default_row_format`. For more information, see Specifying the Row Format for a Table.

- **Important Change:** Introduced the `->` JSON column-path operator. *`column->path`* is now supported as a synonym of `JSON_EXTRACT(`*`column, path`*`)`, where *`column`* is a `JSON` column, and *`path`* is a valid JSON path.

  An expression with `->`, like its equivalent that uses `JSON_EXTRACT()` instead, can be used in place of a column identifier wherever the latter can occur within a valid SQL statement. For example, the following `CREATE TABLE` and `SELECT` statements are valid:

```
CREATE TABLE t1 (
  a JSON,
  b INT,
  g INT GENERATED ALWAYS AS (a->"$.id"),
  h INT GENERATED ALWAYS AS (a->"$.storeid"),
  INDEX i (g),
  INDEX j (h)
);

SELECT
  CONCAT(a->"$.fname", ' ', a->"$.lname") AS name,
```

```
a->"$.id" AS id,
a->"$.storeid" AS store
FROM t1
WHERE g > 500
ORDER BY a->"$.storeid", a->"$.lname";
```

A column-path expression can be used for any column value that is read in a `SELECT` column list, or in a `WHERE`, `ORDER BY`, or `GROUP BY` clause in any SQL statement; such expressions cannot be used to set values.

When an SQL statement contains one or more expressions using `->` notation, each of these is translated into an equivalent expression that employs the `JSON_EXTRACT()` function instead. This can be seen in the output from `EXPLAIN` when used on such a statement.

Like `JSON_EXTRACT()`, the `->` operator returns as `NULL` if no matching value for an otherwise valid path is found.

For more information about `->` and `JSON_EXTRACT()`, see Functions That Search JSON Values. See Searching and Modifying JSON Values, for information about JSON path support. See also Secondary Indexes and Generated Virtual Columns, for additional information and examples.

- **InnoDB:** A new `INNODB_METRICS` server operations counter (`innodb_dict_lru_count`) counts the number of tables evicted from the table cache LRU list. Thanks to Daniël van Eeden for the patch. (Bug #21682332, Bug #78190)
- **InnoDB:** The new `innodb_numa_interleave` read-only configuration option allows you to enable the NUMA interleave memory policy for allocation of the `InnoDB` buffer pool. When `innodb_numa_interleave` is enabled, the NUMA memory policy is set to `MPOL_INTERLEAVE` for the `mysqld` process. After the `InnoDB` buffer pool is allocated, the `NUMA` memory policy is set back to `MPOL_DEFAULT`. This option is only available on NUMA-enabled systems.

  Thanks to Stewart Smith for the patch. (Bug #18871046, Bug #72811)
- MySQL distributions now include these header files because `my_sys.h` depends on them: `my_thread_local.h`, `thr_cond.h`, `thr_mutex.h`, `thr_rwlock.h`. (Bug #21909332)
- MySQL Server RPM packages now obsolete MySQL Connector C. Installing MySQL Server causes older `libmysqlclient` from any MySQL Connector C packages to be removed and replaces them with the current `libmysqlclient`. (Bug #21900800)
- RPM `.spec` files were updated so that MySQL Server builds from source RPM packages will include the proper files to take advantage of operating system NUMA capabilities. This introduces a runtime dependency on `libnuma.so.1`. RPM and `yum` detect this and refuse to install if that library is not installed. (Bug #21775221)
- The `JSON_APPEND()` function was renamed to `JSON_ARRAY_APPEND()`. (Bug #21560934)
- The server now generates a warning when the `default_storage_engine` or `default_tmp_storage_engine` system variable is set to a disabled storage engine named in the `disabled_storage_engines` system variable. (Bug #21405865)
- Metadata locking for tablespaces has been extended so that, for DDL statements that refer to multiple tablespaces, a metadata lock is acquired on all used tablespaces. (Bug #21376265)
- Support for building with Solaris Studio 5.13 was added. (Bug #21185883)
- `mysql_ssl_rsa_setup` now is less noisy by default. Output from `openssl` commands is displayed only if `--verbose` is given. (Bug #21024979)

- yaSSL was upgraded to version 2.3.8.

  Upgrading from older versions fixes a connection-failure issue when used with the thread pool plugin. (Bug #20774956, Bug #21888925)

- Insert overhead for the `MEMORY` storage table was reduced by caching computed hash values. (Bug #78480, Bug #21866029)

- The initial-password message written by `mysqld --initialize` to the error log has been reduced from a `[Warning]` to a `[Note]`. (Bug #78182, Bug #21680457)

- `mysqlpump` now supports a `--version` option. (Bug #77894, Bug #21534277)

- Unneeded scripts and test files were removed from the `tests` directory and that directory was renamed to `testclients`. (Bug #77807, Bug #21490075)

- The JSON value-updating functions `JSON_APPEND()`, `JSON_SET()`, `JSON_REPLACE()`, and `JSON_INSERT()` now treat SQL `NULL` values as JSON null literals, which is consistent with `JSON_OBJECT()` and `JSON_ARRAY()`. (Bug #77733, Bug #21450922)

- Performance Schema digests in `DIGEST_TEXT` columns have `...` appended to the end to indicate when statements exceed the maximum statement size and were truncated. This is also now done for statement text values in `SQL_TEXT` columns. (Bug #75861, Bug #20519832)

- Output from `mysql_upgrade` is now less noisy and more informative. (Bug #59077, Bug #11766046)

- A new `SHUTDOWN` SQL statement is available. This provides an SQL-level interface to the same functionality previously available using the `mysqladmin shutdown` command or the `mysql_shutdown()` C API function. See SHUTDOWN Syntax.

  The `mysql_shutdown()` function and corresponding `COM_SHUTDOWN` client/server protocol command are deprecated and will be removed in a future version of MySQL. Instead, use `mysql_query()` to execute a `SHUTDOWN` statement.

- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.

Bugs Fixed

- **Important Change; InnoDB; Partitioning:** There was no way to upgrade existing partitioned tables to use the native partitioning implemented for `InnoDB` tables in MySQL 5.7.6. This fix adds support to both `mysql_upgrade` and the `mysql` client for upgrading partitioned `InnoDB` tables created in previous releases, which used the `ha_partition` handler, to use `InnoDB` native partitioning instead.

  - `mysql_upgrade` now checks for all `InnoDB` tables that were created using the generic `ha_partition` handler and attempts to upgrade them to `InnoDB` native partitioning.

  - In the `mysql` client, pre-5.7.6 partitioned `InnoDB` tables can be upgraded one by one to native partitioning using the `ALTER TABLE ... UPGRADE PARTITIONING` statement that is implemented in this release.

  (Bug #20727344, Bug #76374)

- **Important Change; Replication:** The `START SLAVE` and `STOP SLAVE` statements can no longer be used for the `group_replication_recovery` channel. See Replication Channels, for more information. (Bug #21680074)

- **Important Change; Replication:** In MySQL 5.7, the `binlog_max_flush_queue_time` system variable introduced in MySQL 5.6 no longer has any effect. This variable is now deprecated, and has been marked for eventual removal in a future MySQL release. (Bug #21347087)

- **Important Change; Replication:** When using a single-threaded slave, the status of the applier thread is now reported as part of the `replication_applier_status_by_worker` table, instead of `replication_applier_status_by_coordinator`. This means that `replication_applier_status_by_coordinator` is now empty when using a single-threaded slave; it should be noted that such reporting for multi-threaded slaves has not been changed, and continues to be shown in that table. (Bug #74765, Bug #20001173)

- **InnoDB:** The undo log contained insufficient information about virtual columns and virtual column indexes, which could cause a server exit when adding or dropping virtual columns. As a result, a slow shutdown (using `innodb_fast_shutdown=0`) is required prior to performing an in-place upgrade or downgrade from MySQL 5.7.8. For more information, refer to Upgrading or Downgrading MySQL. (Bug #21869656, Bug #78489)

- **InnoDB:** A table-rebuilding `ALTER TABLE` operation that created an index on a virtual column raised an assertion. (Bug #21847170, Bug #78452)

- **InnoDB:** A `SELECT ... FOR UPDATE` operation on a table with virtual generated columns raised an assertion. `InnoDB` unnecessarily retrieved a non-indexed virtual column for a covered secondary index scan. (Bug #21827963)

- **InnoDB:** Creating a table with large rows failed when using a `ROW_FORMAT=DYNAMIC` or `ROW_FORMAT=COMPRESSED` due to an incorrect undo log record size calculation. The same operation succeeded with `ROW_FORMAT=COMPACT` or `ROW_FORMAT=REDUNDANT`.

  `ROW_FORMAT=DYNAMIC` and `ROW_FORMAT=COMPRESSED` now permit a row length violation at DDL time if `innodb_strict_mode` is disabled. (Bug #21816041, Bug #78392)

- **InnoDB:** Adding an index on a generated virtual column with an index prefix length that exceeded the maximum length caused a server exit. (Bug #21812026)

- **InnoDB:** A memory leak occurred after crash recovery. Memory allocated in `fil_space_read_name_and_filepath()` was not freed. (Bug #21811321)

- **InnoDB:** Altering the data type or computation method of a virtual generated column caused an error. (Bug #21810004)

- **InnoDB:** An assertion was raised when creating a spatial index. `InnoDB` failed to count virtual columns that preceded the spatial index column. (Bug #21807340)

- **InnoDB:** Tablespace discovery modifications in MySQL 5.7.5 included the removal of code related to `MLOG_FILE_CREATE2` redo log records. As a result, the redo log did not contain sufficient information about file creation. (Bug #21801423, Bug #78363)

- **InnoDB:** The redo log provided no indication that redo logging is disabled for index page writes during concurrent DDL operations. As a result, external hot backup tools could produce corrupt backups. (Bug #21796691, Bug #78351)

- **InnoDB:** In debug builds, recovery asserted during a transparent page compression test. A torn page from the doublewrite buffer caused an LSN debug check failure. (Bug #21796092)

- **InnoDB:** Creating a virtual generated column on a partitioned table caused a server exit. (Bug #21790751, Bug #78326)

- **InnoDB:** An old version of `numactl` headers on the build host caused a compilation error when building a MySQL version that includes NUMA memory policy support. (Bug #21785074)
- **InnoDB:** A `SELECT .. FOR UPDATE` operation on an indexed virtual generated column raised an assertion. (Bug #21775459)
- **InnoDB:** The tablespace identifier (`space_id`) was logged twice for an `MLOG_TRUNCATE` redo log record. (Bug #21744589)
- **InnoDB:** Crash recovery issued an invalid error message indicating that an `isl` file could not be opened or is not correct. (Bug #21691438)
- **InnoDB:** The `InnoDB` Monitor displayed incorrect mutex creation information for a mutex with a long semaphore wait. (Bug #21682997, Bug #78179)
- **InnoDB:** Error message formatting was corrected in the `os_file_write_page()` function. (Bug #21681433, Bug #78184)
- **InnoDB:** The `i_s_dict_fill_sys_tablespaces()` function could free memory associated with a file name before printing an error message. (Bug #21680518, Bug #78180)
- **InnoDB:** In debug builds, scanned and applied redo log records are now printed with a string identifier instead of a numerical identifier when `--debug=d,ib_log` is passed to `mysqld`. For example, `rec MLOG_2BYTES` is now printed instead of `rec 2`. (Bug #21664268, Bug #78148)
- **InnoDB:** The number of system calls made by the `InnoDB` page compression feature was reduced. (Bug #21654695)
- **InnoDB:** The `recv_parse_log_rec` function returned the length of the redo log record instead of 0 when encountering an incomplete `MLOG_CHECKPOINT` record. (Bug #21640085, Bug #78058)

  References: This issue is a regression of: Bug #17798076.
- **InnoDB:** In debug builds, a parenthesis mismatch in a `MATCH() ... AGAINST` clause raised a full-text parser assertion. (Bug #21638907)
- **InnoDB:** Debug code was added to avoid a doublewrite buffer assertion that was raised during Valgrind testing. (Bug #21631197)
- **InnoDB:** Log messages were improved to help identify out-of-space errors that could occur when adding rollback segments. (Bug #21629618)
- **InnoDB:** In debug builds, enabling the `innodb_log_checkpoint_now` debug option while a table-rebuilding `ALTER TABLE` operation is running could result in an infinite loop. (Bug #21628087, Bug #78056)
- **InnoDB:** `UNIV_INLINE` was not defined for the `dict_table_has_indexed_v_cols` function. (Bug #21628058, Bug #78055)
- **InnoDB:** Altering a virtual column data type is not supported as an in-place operation. (Bug #21617377)
- **InnoDB:** Sorting was skipped by an `ALTER TABLE` statement that changed the primary key and dropped the last column of the previous primary key. (Bug #21612714, Bug #78020)
- **InnoDB:** During recovery, an invalid `isl` file was treated as a missing `isl` file, resulting in the tablespace being opened using the file location defined in an `MLOG_FILE_*` record in the redo log. Recovery no longer opens the tablespace if the `isl` file is invalid. (Bug #21577278, Bug #77986)

- **InnoDB:** Code that provided a timeout mechanism intended to reduce adaptive hash index search latch (`btr_search_latch`) contention was removed. The code became obsolete after the introduction of adaptive hash index search system partitioning in MySQL 5.7.8. (Bug #21569876, Bug #77957)
- **InnoDB:** The `COMPRESSION` option was displayed incorrectly in `SHOW CREATE TABLE` output. (Bug #21557723, Bug #77940)
- **InnoDB:** An `innodb_data_file_path` mismatch raised an assertion, as did initializing the database with a data file size that was too small for the specified page size. (Bug #21551464)
- **InnoDB:** A check was added to prevent accessing full-text index tables that are in an inconsistent state. (Bug #21529012)
- **InnoDB:** A schema mismatch error occurred when importing a tablespace that was altered by `DROP INDEX` operation on the source server. (Bug #21514135, Bug #77659)
- **InnoDB:** Creating an index on a virtual generated column after adding a full-text index on a preceding column raised an assertion. (Bug #21478389)
- **InnoDB:** For tables with a `DYNAMIC` or `COMPRESSED` row format, more data than necessary was undo logged for virtual columns. (Bug #21477535)
- **InnoDB:** An `INSERT` operation raised a `btr_search_enabled` assertion. Assertion code was too restrictive. (Bug #21457373)
- **InnoDB:** Reloading a table that was evicted while empty caused an `AUTO_INCREMENT` value to be reset. (Bug #21454472, Bug #77743)
- **InnoDB:** A crash during a `TRUNCATE TABLE` operation caused the server to exit on startup. (Bug #21451922)
- **InnoDB:** The `.isl` was not removed when dropping a general tablespace that was created outside of the MySQL data directory. (Bug #21446772, Bug #77724)
- **InnoDB:** An inconsistent read occurred under the `REPEATABLE READ` transaction isolation level. Transactions that operated on the same row were removed from transaction ID list in the incorrect order. (Bug #21433768, Bug #77699)

  References: This issue is a regression of: Bug #17320977.
- **InnoDB:** Attempting to create a general tablespace data file on a Windows root drive caused an error. (Bug #21419888, Bug #77676)
- **InnoDB:** After disabling the adaptive hash index feature, an adaptive hash index latch was unnecessarily obtained and released. (Bug #21407023)
- **InnoDB:** An `ALTER TABLE` operation on a table with an index defined on a virtual column incorrectly modified the data of the virtual column. (Bug #21376546, Bug #77628)
- **InnoDB:** A virtual column-related purge operation raised an assertion. (Bug #21374258)
- **InnoDB:** On a partitioned table, creating an index on a generated column raised an assertion. (Bug #21372331)
- **InnoDB:** Resizing the buffer pool online raised an assertion due to a memory full condition. (Bug #21348684, Bug #77564)
- **InnoDB:** An `ALTER TABLE ... IMPORT TABLESPACE` operation with `innodb_page_size=4K` and `ROW_FORMAT=DYNAMIC` raised an assertion. (Bug #21341030, Bug #77540)

- **InnoDB:** For spatial indexes, `InnoDB` unnecessarily stored a 3072-byte prefix in undo log records instead of just the maximum bounding rectangle (MBR). For columns with externally stored data, both the prefix and MBR are logged. (Bug #21340268, Bug #77537)
- **InnoDB:** Invalid `init_ftfuncs()` assertion code was removed. (Bug #21300774)
- **InnoDB:** Memory allocation sanity checks were added to the `memcached` code. (Bug #21288106)
- **InnoDB:** An incorrect reference count caused a hang in the `TrxInInnoDB` constructor. `innobase_close_connection()` released the transaction object before destroying the `TrxInInnoDB` object where the reference count is adjusted. (Bug #21280816)
- **InnoDB:** A MySQL 5.7.8 patch that reintroduced `SHOW ENGINE INNODB MUTEX` functionality caused a performance regression. (Bug #21266784)

  References: See also: Bug #77314, Bug #21238953.
- **InnoDB:** A `memcached flush_all` command raised an assertion. A function that starts a transaction was called from within assertion code. (Bug #21239299, Bug #75199)
- **InnoDB:** A shutdown hang occurred when a high priority transaction waited for a victim transaction to exit while the victim transaction waited for an asynchronous rollback to complete. (Bug #21143276)
- **InnoDB:** A data corruption occurred on ARM64. GCC builtins did not issue the correct fences when setting or unsetting the lock word. (Bug #21102971, Bug #76135)
- **InnoDB:** Server shutdown was delayed waiting for the purge thread to exit. To avoid this problem, the number of calls to `trx_purge()` was reduced, and the `trx_purge()` batch size was reduced to 20. (Bug #21040050)
- **InnoDB:** In `READ COMMITTED` mode, a `REPLACE` operation on a unique secondary index resulted in a constraint violation. Thanks to Alexey Kopytov for the patch. (Bug #21025880, Bug #76927)
- **InnoDB:** The `IBUF_BITMAP_FREE` bit indicated that there was more free space in the leaf page than was actually available. (Bug #20796566)
- **InnoDB:** Moving the data directory before recovering a crashed database caused tablespace discovery to fail for file-per-table tablespaces created outside of the MySQL data directory. (Bug #20698468, Bug #76308)
- **InnoDB:** The `innodb_buf_flush_list_now` debug setting failed to flush all dirty pages to disk. (Bug #20582189)
- **InnoDB:** An `ALTER TABLE ... ADD FULLTEXT INDEX` operation raised an assertion. A thread attempted to use a lower priority transaction that was being rolled back before the rollback operation completed. (Bug #20481175)
- **InnoDB:** Running an `ALTER TABLE` operation on a referencing table with a cascading foreign key constraint during a concurrent DML operation on the referenced table caused a loss of referential integrity. (Bug #20367116)
- **InnoDB:** Setting `lower_case_table_names=0` on a case-insensitive file system could result in a hang condition when running an `INSERT INTO ... SELECT ... FROM` *`tbl_name`* operation with the wrong *`tbl_name`* letter case. An error message is now printed and the server exits when attempting to start the server with `--lower_case_table_names=0` on a case-insensitive file system. (Bug #20198490, Bug #75185)
- **InnoDB:** The server failed to start with an `innodb_force_recovery` setting greater than 3. `InnoDB` was set to read-only mode before redo logs were applied.

`DROP TABLE` is now supported with an `innodb_force_recovery` setting greater than 3. (Bug #19779113)

- **InnoDB:** The `trx_sys_read_pertable_file_format_id()` function reported the wrong file format. (Bug #19206671)
- **InnoDB:** The `mysql_system_tables_fix_for_downgrade.sql` script, provided to facilitate system table alterations when downgrading from MySQL 5.7.6 or higher, was removed from the MySQL installation directory. The script was no longer valid for all downgrade paths. For more information, see Changes Affecting Downgrades from MySQL 5.7. (Bug #78259, Bug #21753832)
- **InnoDB:** A virtual generated column on a table that uses index condition pushdown (ICP) caused an assertion. (Bug #77842, Bug #21507796, Bug #21478287)
- **Packaging; OS X:** Using `user=mysql` during installation on OS X did not allow the `mysql` database to be installed. To fix this problem, OS X packages now use the `--no-defaults` option when creating this database. This also means that having a `my.cnf` file on the system no longer affects the installation. (Bug #21364902)
- **Partitioning:** Error handling for failed partitioning-related `ALTER TABLE` operations against non-partitioned tables was not performed correctly (Bug #20284744)
- **Partitioning:** `ALTER TABLE` when executed from a stored procedure did not always work correctly with tables partitioned by `RANGE`. (Bug #77333, Bug #16613004, Bug #21246891)
- **Replication:** The group replication applier channel does not support `DATABASE` as the `slave_parallel_type`; when group replication is started, this is checked for explicitly, and handled correctly. However, it remained possible to change this value indirectly at a later point in time by increasing the value of `slave_parallel_workers` while the slave SQL thread was stopped, which caused the applier to fail with an error. To fix this problem, the `slave_parallel_type` for the `group_replication_applier` is now checked to make sure that it is set to `LOGICAL_CLOCK` whenever the number of `slave_parallel_workers` is set greater then 0, and not merely when group replication is first started. (Bug #21798804)
- **Replication:** As `binlog_error_action=ABORT_SERVER` is the default in MySQL 5.7.7 and later it is being used for more error situations. The behavior has been adjusted to generate a core dump to improve troubleshooting possibilities. (Bug #21486161, Bug #77738)
- **Replication:** At runtime, some `Gtid_set` objects could be instrumented with a performance schema mutex key equal to 0 (which is invalid), due to its use as the effective default value when the mutex key was not actually supplied. This allowed these objects to be created without a valid key, which led to further issues when using them. (Bug #21485997)
- **Replication:** When running the server with `gtid_mode=ON`, a `DELETE` from a `MEMORY` table following a restart was not written to the binary log correctly. (Bug #21045848)
- **Replication:** The locking behavior of replication administration statements has changed to make `SHOW SLAVE STATUS` more concurrent. This makes the `NONBLOCKING` clause redundant for `SHOW SLAVE STATUS` and it has been removed. (Bug #20593028)
- **Replication:** **`ER_CANT_USE_AUTO_POSITION_WITH_GTID_MODE_OFF`** errors were not reported using the correct format. (Bug #20545943)
- **Replication:** When the dump thread was killed while dumping an inactive binary log, some events in this log could be skipped and thus not replicated. (Bug #78337, Bug #21816399)

  References: See also: Bug #74607, Bug #19975697.

- **Replication:** XA transactions could cause an assert condition on `XA COMMIT`; this was happening because the internal transaction state was not reset between `XA PREPARE` and `XA COMMIT` or `XA ROLLBACK`, due to the fact that these operations constitute separate transactions under XA. In addition, `XA ROLLBACK` statements were not handled properly in some cases. (Bug #78264, Bug #21755890)

- **Replication:** The interface between the Group Replication plugin and the Performance Schema engine made use of a type of memory allocation which was passed to the server, and was a potential source of problems when passing information between the plugin and `performance_schema` tables. The implementation for this interface has been reworked so as to avoid performing this type of memory allocation when sharing data. (Bug #78263, Bug #21755699)

- **Replication:** The MTS submode set for each channel was ignored by the worker threads, which continued to read and use the global flag set for all slave channels. This could lead to errors when the coordinator was of one type and its workers of another. (Bug #77763, Bug #21464737)

- **Replication:** Replication slaves could fail for having insufficient privileges when they had been granted only the `REPLICATION SLAVE` privilege. (Bug #77732, Bug #21455603)

- **Replication:** The status variable `Slave_open_temp_tables` keeps track of the number of temporary tables that are opened by the replication slave. If multi-source replication is enabled, it is the total number of temporary tables for all channels. This fix addresses the following issues relating to this variable:

  - `RESET SLAVE FOR CHANNEL` *`channel`* forced the value of `Slave_open_temp_tables` to 0; in the event that some other replication channel had open temporary tables which were later dropped, the value wrapped around to a large negative value (`1 - 2`^{32}). This also caused spurious or missed warnings when issuing a `STOP SLAVE` or `CHANGE MASTER TO` statement.

  - The internal function that modifies `Slave_open_temp_tables` in such cases relied on two incorrect assumptions:

    1. That the variable is updated by only one thread when multi-threaded slaves are not enabled, which is not true in the case of multi-source replication.

       That non-atomic operations are safe with a single writer and multiple readers, which is not necessarily true for some platforms supported by MySQL.

  (Bug #77585, Bug #21357008)

- **Replication:** The warning `'@@session.gtid_executed' is deprecated and will be removed in a future release.` was printed even when the session variable `gtid_executed` was not included in the result of a query. In addition, the result of `SELECT @@session.gtid_executed` included a duplicate warning. Both issues occurred because the warning was printed whenever the value of `gtid_executed` was accessed by a statement, such access occurring as a matter of course, whether or not a given variable is actually included in the result.

  To fix this issue, we make handling of `@@session.gtid_executed` consistent with how the also-deprecated variable `@@global.sql_log_bin` is treated in such cases, by making the following changes:

  - `gtid_executed` is no longer included in the `performance_schema.session_variables` table.

  - `gtid_executed` is still included in the `information_schema.session_variables` table, but when `show_compatibility_56 = ON`, the warning is not issued when querying the session_variables table, or when issuing `SHOW VARIABLES` or `SHOW SESSION VARIABLES`, even when using a matching `LIKE` clause with either of the `SHOW` statements.

The warning is still issued by a statement such as `SELECT @@session.gtid_executed` which accesses the value of the variable directly. (Bug #77574, Bug #21354712)

References: See also: Bug #75980, Bug #20575529, Bug #76626, Bug #20854952.

- **Replication:** When a transaction consisting of a single statement with a specified GTID failed in autocommit mode, its GTID was not released when rolling it back when binary logging was disabled. (Bug #77521, Bug #21338147)

- **Replication:** The slave group event parser did not properly register an `XA_ROLLBACK` event as a transaction boundary. (Bug #77392, Bug #21273010)

  References: See also: Bug #20920851.

- **Replication:** `mysqlbinlog` printed a `ROLLBACK` statement at the end of the binary log file, which when played back failed with error 1782 `@@SESSION.GTID_NEXT cannot be set to ANONYMOUS when @@GLOBAL.GTID_MODE = ON`. This occurred when the binary log file did not include any data related events, or when the relay log file included a `Format_description_log_event` that had been generated on the master at server startup.

  The fix for this issue causes a relay log's `Format_description_log_event` to do nothing if it is applied by a `BINLOG` statement, and stops a `ROLLBACK` from setting `gtid_next` to `ANONYMOUS` when the state of `gtid_next` has not yet been determined by a subsequent event. (Bug #76887, Bug #20980932)

- **Replication:** `SAVEPOINT` and `ROLLBACK TO SAVEPOINT` within a trigger led to an assertion. (Bug #76727, Bug #20901025)

- **Replication:** While a `SHOW BINLOG EVENTS` statement was executing, any parallel transaction was blocked. The fix ensures that the `SHOW BINLOG EVENTS` process now only acquires a lock for the duration of calculating the file's end position, therefore parallel transactions are not blocked for long durations. (Bug #76618, Bug #20928790)

- **Replication:** If a `CREATE VIEW` statement failed, it was being incorrectly written to the binary log even though it did not result in the creation of a partial view. The fix ensures that such statements are not recorded in the binary log. Additionally it was found that when a statement which had failed on a master was received by a slave with an expected error, if the statement was skipped on the slave, for example due to a replication filter, the expected error was being compared with the actual error that happened on the slave. The fix ensures that if a statement with an expected error is received by a slave, if the statement has not been filtered, only then is it compared with the actual error that happened on the slave. (Bug #76493, Bug #20797764)

- **Replication:** The action specified for `binlog_error_action` was not always honored correctly after a hardware failure occurred during log rotation. (Bug #76379, Bug #20805298)

- **Replication:** When using MySQL 5.7.6 and later with `binlog_format=row` and `gtid_mode=off`, if `CREATE ... SELECT` was killed during execution it could lead to an inconsistent state, breaking replication. The cause was that in MySQL 5.7.6 the way `CREATE ... SELECT` was logged was changed, so that a commit was introduced between the `CREATE TABLE` and `SELECT` steps. The fix ensures that `CREATE ... SELECT` does not commit in the middle of the transaction when `binlog_format=row`. (Bug #76320, Bug #77098, Bug #20742519, Bug #21114464)

- **Replication:** Modifying the `master_info_repository` or `relay_log_info_repository` inside a transaction and later rolling back that transaction left the repository in an unusable state. We fix this by preventing any modification of these repositories inside a transaction. (Bug #74950, Bug #20074353)

- **Replication:** Transactions added to `gtid_executed` using `SET gtid_purged` were not taken into account by `WAIT_FOR_EXECUTED_GTID_SET()` until a subsequent transaction was committed by a client or slave thread. (Bug #73838, Bug #19579811)

- **Replication:** When using `--relay-log-info-repository=TABLE`, the `mysql.slave_relay_log_info` table is updated when a transaction is committed or when a flush is performed explicitly, such as during relay log rotation. If a transaction that uses any nontransactional tables (for example `MyISAM` tables) is split across multiple relay logs, it is partially committed on relay log flush. When `gtid_mode=ON`, this caused the same GTID to be used for the remaining portion of the transaction, which raised an **`ER_GTID_NEXT_TYPE_UNDEFINED_GROUP`** error.

  We fix this issue by postponing in such cases the update of the relay log information repository that normally occurs on relay log rotation until the commit for the transaction in question has been executed.

  This issue did not affect tables using transactional storage engines such as `InnoDB`. (Bug #68525, Bug #16418100)

  References: See also: Bug #21630907, Bug #76974.

- RHEL RPM packages had incorrect dependency information. (Bug #22218841)

- For an index-only scan over an indexed generated column, the server could do random calculations; the random results were not exposed to the user, but Valgrind warnings could occur, and the server could exit when calculations involved functions which did not expect such incorrect data. (Bug #21833760)

- Although the use of `JSON` values with `GREATEST()` or `LEAST()` is not currently supported, the server did not handle attempts to do so correctly, leading to an assert (Linux) or `exit()` call (Windows) in debug builds. Now when you try to use JSON values with either of these functions, the server emits a suitable warning (`ER_NOT_SUPPORTED_YET`). (Bug #21828321)

  References: See also: Bug #21383497.

- A table that included a generated column referencing a `JSON` column in some cases become corrupted, so that a subsequent access of the table using a different connection caused the server to fail. (Bug #21808680)

  References: See also: Bug #21824519, Bug #78408.

- For tables with `VIRTUAL` generated columns, an `INSERT` with an empty values list could cause a server exit. (Bug #21807818)

- `CMake` configuration was adjusted to handle new warnings reported by Clang 3.7. (Bug #21803314)

- For plugins of type `PROTOCOL_PLUGIN`, execution of `INSTALL PLUGIN`, `UNINSTALL PLUGIN`, or `SHUTDOWN` could cause a server exit. Such plugins are no longer permitted to execute these statements. (Bug #21797816)

- Using a materialized view defined over a table containing generated columns could cause a server exit. (Bug #21797776)

- For partitioned `InnoDB` tables containing a virtual generated column, reads from the table could return random data for the column. (Bug #21779011)

- The `CMake` checks for NUMA availability could cause compilation problems on platforms without NUMA support. (Bug #21774859)

- The optimizer did not consider nonfunctional expressions such as `(a AND b) = 1` when looking for indexed generated columns to substitute for the `(a AND b)` expression. Now expressions using the `AND` and `OR` logical operators are considered. (Bug #21770798)
- For debug builds, when the optimizer tried to clone certain types of keys for a range optimization, an assertion was raised. (Bug #21761867)
- For debug builds, the server could exit when the optimizer attempted to estimate the cost for processing unique values when there were no keys. (Bug #21697002)
- An `INSERT` into a view with a subquery could fail if executed as a prepared statement. (Bug #21696206)
- For queries on `InnoDB` tables for which the optimizer used `SPATIAL` indexes for full index scans, the result was empty because such indexes do not support a full scan. The optimizer no longer considers `SPATIAL` indexes as candidates for full index scans. (Bug #21663612)
- For some inputs, `ST_Intersection()` could return an invalid polygon. (Bug #21658453)
- Spatial functions could simplify geometry values in contexts where the value might be used elsewhere in a query, producing incorrect results. (Bug #21652012)
- If `ST_ConvexHull()` or `ST_SRID()` were used in a view definition, the resulting definition contained `ST_Convex_Hull()` (misspelled) or `SRID()` (deprecated). (Bug #21651588)
- `JSON_TYPE()` returned `OPAQUE` for some binary values that it should have identified as `BLOB`. (Bug #21649073)
- For debug builds, enabling the `PAD_CHAR_TO_FULL_LENGTH` SQL mode could cause `SHOW FUNCTION STATUS` to raise an assertion. (Bug #21632425)
- `mysqlpump` did not exit with a message for some combinations of incompatible options. (Bug #21628662)
- An assertion could be raised if the optimizer tried to create a temporary table based on a prepared statement parameter. (Bug #21625929)
- Executing a prepared statement with multiple nested subqueries could raise an assertion. (Bug #21624851)
- For debug builds, failure of subquery optimization could cause an assertion to be raised due to improper error handling. (Bug #21621313)
- Some table and index optimizer hints were lost early in statement processing, so query rewrite plugins did not have access to them. This could cause incorrect matching between incoming statements and statement pattern templates. (Bug #21619780)
- Queries containing nested subqueries combining grouping and outer references might cause a server exit. (Bug #21619634)
- Passing `NULL` as the second or third argument to `ST_AsGeoJSON()` could cause the server to stop responding to the session or (in debug builds) to raise an assertion.

  Giving input to `HANDLER READ` that could not be converted to the correct type could cause the server to stop responding to the session or (in debug builds) to raise an assertion. (Bug #21616810, Bug #21650603)
- For debug builds, `ST_IsValid(NULL)` could raise an assertion. (Bug #21616647)
- For debug builds, an assertion could be raised for negative zero values when converting time values to decimal. (Bug #21616585)

- `ST_AsWKB()` could cause a server exit if given invalid data. (Bug #21614368)

  References: See also: Bug #22131961.

- If an aggregate function was used over a generated column that was itself part of a multiple-column index, the server could exit. (Bug #21613615)

- A missing error check during column reference resolution could result in an incorrect error message or (in debug builds) an assertion being raised. (Bug #21613422)

- For debug builds, an assertion could be raised in `Filesort::make_sortorder()` for attempts to sort `Item_ref` objects. (Bug #21611270)

- JSON functions could return incorrect values if a path argument was passed as a user-defined variable that changed values between result set rows. (Bug #21602361)

- For debug builds, an assertion was raised for some queries that have a semi-join and use the materialization strategy, if a key length or number of key parts was zero. (Bug #21566735)

- Compilation using `gcc` 4.9 or 5.1 failed on ARM64 platforms. (Bug #21552524)

- If a multiple-column `UPDATE` statement failed to update a `JSON` column that was then referenced in a later update, the server could exit. (Bug #21547877)

- For debug builds, invalid geometry byte strings could cause spatial functions to raise an assertion rather than return an error to the caller. (Bug #21546656)

- For debug builds, a too-strict assertion could be raised by invalid characters for `LOAD DATA`. (Bug #21542698)

- The server could exit when `InnoDB` tried to update a secondary index on a `VIRTUAL` generated column of type `BLOB`. (Bug #21530366)

- For debug builds, incorrect caching of JSON values could cause an assertion to be raised. (Bug #21491442)

- An empty string (which is not a valid JSON value) normally is parsed and returned as a JSON null literal, but in some cases could raise an assertion for debug builds. (Bug #21487833)

- For temporary tables created to handle `UNION` statements that selected `CHAR` or `SET` columns, the maximum column width could be too long for `InnoDB` to handle. Now such columns are created as variable-length columns. (Bug #21480999)

- For builds configured with `MAX_INDEXES` greater than 64, certain queries for which the server used temporary tables could cause a server exit. (Bug #21466850)

- Adding or dropping a `VIRTUAL` generated column could cause a server exit. (Bug #21465626)

- For plugins that use the audit plugin API, `MYSQL_AUDIT_GLOBAL_VARIABLE_SET` events passed to the notification function did not include the new variable value. (Bug #21457699)

- Queries on a table containing an indexed generated column could fail if the table name contained special characters. (Bug #21454155)

- When a view was the inner table of an outer join, a `JSON` column could produce a non-`NULL` value when `NULL` was expected. (Bug #21448719)

- If `JSON_CONTAINS_PATH()` was called with a *`one_or_all`* argument of `all` and a path argument contained a wildcard, the function found all matches per path, even though in this case one match is sufficient. (Bug #21442775)

- `JSON_SET()` and `JSON_REPLACE()` sometimes produced an incorrect result if a path expression identified a nonarray value. (Bug #21442624)
- Suppression of JSON conversion errors using non-strict SQL mode or `INSERT IGNORE` could then cause an assertion to be raised if an empty value inserted into a `JSON NOT NULL` column was copied to another `JSON` column. (Bug #21437989)
- For debug builds, some spatial functions that accept raw byte data for spatial arguments (for example, specified as hex values) could raise an assertion if such an argument contained extra garbage following valid data. (Bug #21397107)
- For deeply nested JSON input, `ST_GeomFromGeoJSON()` or `JSON_VALID()` could produce stack overflow. (Bug #21389101, Bug #21377136)
- Failure to parse a JSON string that contained a floating-point number with a large, negative exponent could cause a server exit. (Bug #21384048)
- For debug builds, an incorrect assertion could be raised during subquery execution. (Bug #21383882)
- For debug builds, a missing error check in `Item_sum_hybrid::fix_fields()` caused an assertion to be raised. (Bug #21383714)
- For debug builds, invoking `ST_AsGeoJSON()` within `GROUP BY ... WITH ROLLUP` could raise an assertion. (Bug #21383497)
- `JSON_SEARCH()` could return incorrect results if an invalid escape expression was specified. (Bug #21383284)
- For debug builds, a `NULL` first argument to `JSON_SET()` could raise an assertion. (Bug #21381806)
- For expressions of the form `(subquery) IN (subquery)`, where a subquery could return a JSON value, failure to handle a row result could cause a server exit.` (Bug #21376088)
- Failure of `JSON_APPEND()` to handle a legal condition could cause a server exit. (Bug #21373874)
- Calls to `ST_Buffer()` could hang or raise an assertion. (Bug #21372946)
- The server could exit in unclean fashion if configured to listen on a TCP/IP port number already in use by another server instance. (Bug #21368299)
- Certain subqueries as arguments to `PROCEDURE ANALYSE()` could cause a server exit. (Bug #21350175)
- A query with a right outer join inside a derived table might return wrong data. (Bug #21350125)
- Starting the server with `--skip-grant-tables` (or with options such as `--initialize` for which `--skip-grant-tables` is implicit) prevented the `INSTALL PLUGIN` and `UNINSTALL PLUGIN` statements from working. (Bug #21335821)
- `mysql_ssl_rsa_setup` could create an unwanted `.rnd` file in the data directory. (The file is actually created by `openssl`, which `mysql_ssl_ras_setup` invokes. `mysql_ssl_rsa_setup` now cleans up the file.) (Bug #21335818)
- Some `INFORMATION_SCHEMA` queries consumed excessive memory due to suboptimal query plans and insufficient materialization. (Bug #21299665)
- With the server configured to send error messages to `syslog` or a log file, messages generated prior to error log setup were sent to `stderr` or `stdout`. These messages are now buffered until error log setup has completed, then logged to the proper destination. (Bug #21296553)

- Executing a prepared statement using a derived table and an aggregate function in a subquery in the `SELECT` list could cause a server exit. (Bug #21277074)
- `GRANT` created the account for nonexistent accounts even if the `NO_AUTO_CREATE_USER` SQL mode was enabled. (Bug #21271571)
- A query with a `NOT IN` subquery that had `COUNT(DISTINCT)` could return incorrect results. (Bug #21243772)
- When started using a very old data directory (from MySQL 5.0), the server could exit due to failure to properly read the old grant tables. (Bug #21216433)
- A mulitple-table update involving generated columns that updated used a temporary table could cause a server exit or raise an assertion. (Bug #21216067)
- Unit testing now uses Google Mock 1.7 rather than 1.6. (Bug #21215389)
- For queries containing an expression of the form `(x IN (subquery)) IN (subquery2)`, a combination of semi-join and subquery materialization strategies could cause a server exit. (Bug #21205577)
- If a query contained an outer join such as `LEFT JOIN (t1,t2,...)` and a hint was used to disable join buffering on a right-side table but not on the others, a server exit occurred. (Bug #21205282)
- For a cursor type of `CURSOR_TYPE_READ_ONLY`, retrieving the result set for the first execution of a prepared `CALL` staement could be missing the first result set row if the data was numeric; raise an assertion for debug builds if the data was string; cause loss of the server connection when calling `mysql_stmt_fetch()`. (Bug #21199582)
- Dangling blob pointers could remain when closing an `InnoDB` table, resulting in a subsequent read of invalid memory and a server exit. (Bug #21153489)
- For some operations where sorting or grouping required a temporary table, the table could have zero columns and raise an assertion. (Bug #21143151)
- Queries containing an expression of the form `(x IS NULL) IN (subquery)` could cause a server exit. (Bug #21139402)
- During server SSL file autogeneration, `ca.pem` briefly had insecure file permissions. (Bug #21138119)
- An assertion could be raised due to incorrect error handling if a `SELECT ... FOR UPDATE` subquery resulted in deadlock and caused a rollback. (Bug #21096444)
- Selecting the result of an `INSERT()` function call to which input was passed as a hexidecimal string could expose more information than was passed to the function. (Bug #21056907)
- Subqueries having `COUNT()` with `GROUP BY` could yield incorrect results. (Bug #21055139, Bug #78029, Bug #21615020)
- The updatable property of a view is set during view creation. If the underlying table was dropped and re-created as a nonupdatable one, the updatable property of the original view was not revised accordingly. This could cause a server exit for attempts to insert or replace into the view is made. (This problem was specific to views with multiple tables/views and did not occur with update statements.) (Bug #21039264)
- The locking functions provided by the `version_token` plugin were renamed: `vtoken_get_read_locks()`, `vtoken_get_write_locks()`, and `vtoken_release_locks()` are now named `version_tokens_lock_shared()`, `version_tokens_lock_exclusive()`, and `version_tokens_unlock()`, respectively.

These functions also failed to have any effect because they were implicitly unlocked at the end of the statement in which they were set. (Bug #21034322, Bug #21280801)

- Servers linked against yaSSL and compiled with GCC 4.8.2 could fail to respond correctly to connection attempts until several seconds after startup. (Bug #21025377)
- When upgrading an old data directory (MySQL 5.0 or 5.1), `mysql_upgrade` could fail to properly read the `mysql.proc` table. (Bug #20968596)
- For tables with subpartitions, the server could exit due to incorrect error handling during partition pruning if the partition could be identified but not the subpartition. (Bug #20909518)
- `mysql_upgrade` could fail to look for checked tables in the wrong database during the repair phase. (Bug #20868496)
- `DELETE` could check privileges for the wrong database when table aliases were used. (Bug #20777016)
- `mysqldump` used incorrect syntax for generated column definitions. (Bug #20769542)
- Within a trigger, use of a cursor that accessed `OLD` or `NEW` values from a row could cause a server exit. (Bug #20760261)
- Failure during execution of an `XA PREPARE` statement could result in an invalid XA transaction state. Subsequent attempts to start another XA transaction led to an `ER_XAER_OUTSIDE` error. (Bug #20538956)
- The audit log plugin could audit accounts named in the `audit_log_exclude_accounts` system variable. (Bug #20408206)
- If a generated foreign key index was renamed by the same `ALTER TABLE` statement that added a new foreign key with the same name, the server could exit. (Bug #20146455)
- `ALTER TABLE` operations that dropped and added the same `FULLTEXT` index were not performed as in-place (fast) operations that avoid using a temporary copy of the table. (Bug #20106837)
- When the number of days calculated by `DATE_FORMAT()` function was negative, the server could exit. (Bug #19985318)
- `ALTER TABLE` operations to add or modify columns could create geometry columns containing invalid data due to missing validation. (Bug #19880316)
- If range optimization was attempted on an index with a string column as its first part and values used for comparison were fully truncated, comparisons would be incorrect and produce incorrect results. (Bug #19333852)

  References: This issue is a regression of: Bug #16407965.
- The server could hang due to incorrect cleanup of aggregate functions used in a query. (Bug #18979515)
- The server could exit while checking for appropriate indexes to use for certain queries that used aggregate function in the `WHERE` clause. (Bug #18706592)
- On Windows, the `validate_password` plugin could cause a server exit during the dictionary check. (Bug #18636874)
- Invoking a stored program without qualifying it with the database name could lead to stored program compilation errors. (Bug #18599181)

- `EXPLAIN` of statements containing `GROUP_CONCAT()` could cause a server exit. (Bug #17865675)
- The value of the `FOUND_ROWS()` function that returns the number of rows found in the previous query could change during execution of the next query. Now the value of `FOUND_ROWS()` for the previous query remains constant during execution of the next query. (Bug #17846246)
- Failure to check for error conditions could cause some updates or deletes to result in a server exit. (Bug #17763238)
- On Windows, heap corruption in the audit log plugin caused server startup failure. (Bug #14700102)
- If the `UPDATE` part of `INSERT ... ON DUPLICATE KEY UPDATE` swapped two column values, the server could read incorrect data and exit. (Bug #13901905)
- For debug builds, merging a derived table into an outer query block could raise an assertion. (Bug #79502, Bug #22305361, Bug #21139722)
- `mysqlpump` failed to compile with Clang. (Bug #78637, Bug #21924096)
- For debug builds, a `DROP TRIGGER` statement could raise an assertion if the trigger was defined on a table that contained a generated column. (Bug #78408, Bug #21824519)
- Some stress test files in the `mysql-test/suite/innodb_stress` directory had the executable file mode set although they were not script files. (Bug #78403, Bug #21822413)
- Subqueries that used a derived table and contained a set function referring to a column from that derived table might be aggregated in the wrong query block. (Bug #78250, Bug #21753180)
- For some inputs, `ST_Union()` could return an invalid geometry collection. (Bug #78206, Bug #21689998)
- On non-Windows systems, setting `range_alloc_block_size` or `query_alloc_block_size` to a value larger than 32 bits at startup could cause a server exit. The maximum value for these system variables (on all platforms) is now limited to $2^{32} - 1$, rounded down to the nearest multiple of 1024. (Bug #78188, Bug #21682231)
- `mysql-test-run.pl` now has an `--valgrind-clients` option that causes all clients started by `.test` files to be run with `valgrind`. This option requires `valgrind` 3.9 or later.

  In addition, several client memory leak issues were fixed. (Bug #78165, Bug #21672747)
- The `mysql` client parser incorrectly interpreted optimizer hint comments that contained `;`, `"`, `'`, or `` ` `` characters. (Bug #78114, Bug #21646026)
- These Version Tokens issues were resolved:
  - `version_tokens_delete()` now strips whitespace surrounding token names in its argument, similar to `version_tokens_set()` and `version_tokens_edit()`.
  - Passing `NULL` to `version_tokens_delete()` caused a server exit.
  - Passing an argument with an empty token name to to `version_tokens_set()` or `version_tokens_edit()` caused a server exit.
  - Passing `NULL` as the timeout value to `version_tokens_lock_exclusive()` or `version_tokens_lock_shared()` caused a server exit.

  (Bug #78111, Bug #21645001, Bug #21646106, Bug #21645944, Bug #21646017)

- Columns selected from the right-hand table of a left join, which was also a derived table, might produce incorrect *`NULL`* value information when used in an `IN` subquery. (Bug #77980, Bug #21574933)

  References: This issue is a regression of: Bug #14358878.

- On Windows, if the MySQL server was started as a service, logging to the error log file was disabled. (Bug #77977, Bug #21574096)

  References: This issue is a regression of: Bug #21328041.

- In the `setup_instruments` Performance Schema table, it was possible to set memory instrument to `TIMED='YES'`, although memory operations are never timed. Now such attempts are ignored. It was possible to set built-in memory instruments (with names of the form `memory/performance_schema/%`) to `ENABLED='YES'`, although built-in memory instruments cannot be disabled. Now such attempts are ignored. (Bug #77944, Bug #21562212)

- `mysqldump` and `mysqlpump` output included `sys` schema stored programs even when the `sys` schema was not dumped. (Bug #77926, Bug #21549860)

- RPM installation scripts failed if configuration files contained multiple `datadir` lines. Now the last `datadir` line is used. (Bug #77878, Bug #21527467)

- An unnecessary `memset()` call invoked during Performance Schema digest operations has been removed, which improves performance by reducing overhead. (Bug #77863, Bug #21528683)

- A potential race condition for the safe mutex implementation was corrected. This implementation is enabled by default only for debug builds. (Bug #77862, Bug #21522888)

- Binary logging of `CREATE USER` and statements could log the hash of the password hash (rather than the hash itself) when `log_backward_compatible_user_definitions` was enabled. Binary logging of `ALTER USER` statements could include attributes not present in the original statements.

  In consequence of the fix for these issues, `log_backward_compatible_user_definitions` has been replaced by `log_builtin_as_identified_by_password`. If this variable is enabled, binary logging for `CREATE USER` statements involving built-in authentication plugins rewrites the statements to include an `IDENTIFIED BY PASSWORD` clause, and `SET PASSWORD` statements are logged as `SET PASSWORD` statements, rather than being rewritten to `ALTER USER` statements. (Bug #77860, Bug #21516392, Bug #20535561)

- `mysqld --initialize` produced warnings about missing SSL files, which is unnecessary because initialization does not require SSL. (Bug #77825, Bug #21498544)

- Certain JSON functions could return incorrect results when used in prepared statements which had path expression constants. (Bug #77785, Bug #21472872)

- Valgrind errors could occur during partition pruning for tables containing generated columns. (Bug #77782, Bug #21469535)

- When `mysqlpump` was invoked with the `--defer-table-indexes` option, it could generate incorrect `CREATE TABLE` statements for tables with a foreign key and a primary key but not a secondary index. (Bug #77759, Bug #21462732)

- An attempt to use a previously unused time zone with `CONVERT_TZ()` could produce warnings or errors or (in debug builds) raise an assertion if GTIDs were enabled but the binary log was not enabled.

  In debug builds, an attempt to use a previously unused time zone with `CONVERT_TZ()` or as the value of the `time_zone` system variable inside a stored program could raise an assertion. (Bug #77753, Bug #21459999, Bug #77748, Bug #21459795)

- The server initialization script used for the `service mysql status` command on Linux sometimes incorrectly reported that the server was stopped. (Bug #77696, Bug #21768876)

- `ALTER TABLE` could raise an assertion for a table with an indexed virtual column having a column position greater than 64. (Bug #77656, Bug #21391781)

- Evaluation of virtual generated columns could fail to evaluate all base columns and result in invalid memory reads. (Bug #77653, Bug #21390605)

- For statements of the form `CREATE TABLE ... SELECT`, where the table was defined to contain a `DECIMAL UNSIGNED`, calculation of the row size was incorrect, leading to incorrect values in the table. (Bug #77636, Bug #21383896)

- A `WHERE` predicate containing both `TRIM(LEADING ..)` and `TRIM(TRAILING ...)` could be incorrectly optimized away. (Bug #77631, Bug #21447969)

- For wait events, the Performance Schema uses the `CYCLE` timer by default, but failed to fall back to a different timer if `CYCLE` was unavailable. (Bug #77577, Bug #21374104)

- A disk-full condition during execution of a `CREATE TABLESPACE` statement caused a server exit. (Bug #77556, Bug #21347001)

- A privilege precheck for derived tables could fail and cause a server exit. (Bug #77525, Bug #21338077)

- For spatial functions, input polygons were automatically closed if open. However, the Open Geospatial Consortium guidelines require that input polygons already be closed. Unclosed polygons are now rejected as invalid rather than being closed. (Bug #77505, Bug #21327888)

- When a `VIRTUAL` generated column was added to a table, it was not ensured that data being calculated by the generated column expression would not be out of range for the column. This could lead to inconsistent data being returned and unexpectedly failed statements.

  `ALTER TABLE` now supports `WITHOUT VALIDATION` and `WITH VALIDATION` clauses to control whether `ALTER TABLE` validates the data for a `VIRTUAL` generated column:

  - With `WITHOUT VALIDATION` (the default if neither clause is specified), an in-place operation is performed (if possible), data integrity is not checked, and the statement finishes more quickly. However, later reads from the table might report warnings or errors for the column if values are out of range.

  - With `WITH VALIDATION`, `ALTER TABLE` copies the table. If an out-of-range or any other error occurs, the statement fails. Because a table copy is performed, the statement takes longer.

  `WITHOUT VALIDATION` and `WITH VALIDATION` are permitted only with `ADD COLUMN`, `CHANGE COLUMN`, and `MODIFY COLUMN` operations. (Bug #77478, Bug #21317507)

- For some string functions, data was truncated when evaluated in subqueries due to incorrect space calculations when creating temporary tables to hold intermediate results. (Bug #77473, Bug #21317406)

- Statement digests did not include information about optimizer hint comments, causing statements with and without hints to be aggregated. (Bug #77414, Bug #21286261)

- A predicate of the form `WHERE ROUND(X,Y) > 0`, where *X* is a column name and `Y` is a program local variable, could return false when it should return true. (Bug #77391, Bug #21279005)

- `ST_SymDifference()` with multipolygon arguments could return incorrect results. (Bug #77372, Bug #21263152)

- Updating `VARCHAR` and `TEXT` columns in the same `UPDATE` statement could produce incorrect results. When a `VARCHAR` column was assigned to a `TEXT` column and the `VARCHAR` column was then set to a different value, the `TEXT` column's result contained the `VARCHAR` column's new value. (Bug #77135, Bug #21143080)

- A subquery in a `HAVING` clause that returned more than 1 row could cause a server exit.

  Additional to the bug fix, `EXPLAIN` now displays `Zero limit` rather than `Impossible WHERE` when optimizing a query with `LIMIT 0`. (Bug #76998, Bug #21067109)

- If an `INFORMATION_SCHEMA` query that performed a table-open operation encountered a corrupt table and attempted to repair it, a deadlock could occur, resulting in an aborted transaction without an appropriate error being reported. Such queries now do not attempt table repair. (Bug #76912, Bug #21021848)

- `mysqladmin -u root -p` could exit with a segmentation fault. (Bug #76538, Bug #20802751)

- The optimizer sometimes generates an index for a derived table (subquery in the `FROM` clause). If this occurred for a statement executed within a stored program, a memory leak could occur. (Bug #76349, Bug #20728894)

- Optimizer estimates for filtering conditions could lead to suboptimal execution plans if the expected number of rows selected from a table was between 0 and 1. The estimate is now made to be at least 1. (Bug #76314, Bug #20701585)

- If a file was specified using an `--init-file` option, `mysqld --initialize` produced errors for statements in the file such as `GRANT` that affect user accounts. (Bug #75918, Bug #20546898)

- The optimizer could incorrectly assume an out-of-memory condition while optimizing a range scan for the `OR` operator, resulting in overestimation of the number of qualifying rows. (Bug #75248, Bug #20229614)

- The `events_statements_history` Performance Schema table could have an `ERRORS` column value of 0 when other columns indicated there were errors. (Bug #74614, Bug #19929832)

- View creation from a `UNION` failed with a duplicate-column error if a `SELECT` statement in the `UNION` other than the first used the same column name multiple times. (Bug #74539, Bug #19886430)

- Timestamp values written to the slow query log could be incorrect. (Bug #73974, Bug #19646918)

- When the `mysql` client was used to connect to the server in batch mode using an account with an expired password, the error message was not meaningful. `mysql` now reports “Please use --connect-expired-password option or invoke mysql in interactive mode” in this case. (Bug #72696, Bug #21464621)

- For `UPDATE` statements with `ORDER BY`, the optimizer could perform an unnecessary `filesort` on a key that was used for scanning as well as being updated. (Bug #72518, Bug #18698556)

- For a query with many range conditions, the optimizer would estimate that too much memory would be required for a range scan and fall back to a less optimal plan, such as a full table scan.

  A new `range_optimizer_max_mem_size` system variable now controls the limit on memory consumption for the range optimizer. A value of 0 means “no limit.” If an execution plan considered by the optimizer uses the range access method but the optimizer estimates that the amount of memory needed for this method would exceed the limit, it abandons the plan and considers other plans. (Bug #70247, Bug #17413040, Bug #17769777)

- Empty XML elements having the form `<element/>` were not handled correctly by the `LOAD XML` statement. (Bug #67542, Bug #16171518)

- As the number of open `MyISAM` tables increased, lookups to check whether a table was open became expensive, particularly when the table was not open. Lookup performance has been improved, with the overhead reduction especially beneficial for selects on large number of tables with large values of `table_open_cache` and `table_definition_cache`. (Bug #49177, Bug #11757169)

Changes in MySQL 5.7.8 (2015-08-03)

This release adds support for Debian 8 and Ubuntu 15.04.

- Account Management Notes
- Backup Notes
- Configuration Notes
- JSON Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Plugin Service Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Mode Notes
- Functionality Added or Changed
- Bugs Fixed

Account Management Notes

- The `CREATE USER` statement now supports an `IF NOT EXISTS` clause that causes the statement to produce a warning for each named account that already exists, rather than an error. The `ALTER USER` and `DROP USER` statements now support an `IF EXISTS` clause that cause the statements to produce a warning for each named account that does not exist, rather than an error. For details, see CREATE USER Syntax, ALTER USER Syntax, and DROP USER Syntax.

  These statement variants can be useful in replication scenarios when the set of accounts differs between master and slave. They also permit scripting account-management operations that otherwise would terminate for statement errors.

- The maximum length of MySQL user names has been increased from 16 to 32 characters, which provides greater flexibility in choosing the user name part of MySQL account names. The change affects permitted user names in these contexts:

  - Account-management statements, such as `CREATE USER`, `GRANT`, `REVOKE`, and `SHOW GRANTS`.
  - Statements that support a `DEFINER` clause, such as `CREATE PROCEDURE` and `CREATE VIEW`.
  - Other statements with clauses that contain user names, such as `CHANGE MASTER TO` and `CREATE SERVER`.

- Columns that store user names in `mysql` system database, `INFORMATION_SCHEMA`, and Performance Schema tables have been widened to accommodate 32 characters.

There are no changes in the client/server protocol, which exchanges user names as null-terminated strings. However, third-party programs that use this protocol to communicate may need to be modified if they use or store user names based on the assumption of 16 characters maximum.

The increase in maximum user name length has implications for MySQL administration:

- Replication implication: Replication of user names longer than 16 characters to a slave that supports only shorter user names will fail. However, this should occur only when replicating from a newer master to an older slave, which is not a recommended configuration.
- Downgrade implication: If a newer server supports any accounts with a user name longer than 16 characters, downgrades to an older version of MySQL that supports only shorter names is not possible.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this change in user name length.

Backup Notes

- A new client program, `mysqlpump`, provides an alternative to `mysqldump`. Its features include:
  - Parallel processing of databases, and of objects within databases, to speed up the dump process
  - Better control over which databases and database objects (tables, stored programs, user accounts) to dump
  - Dumping of user accounts as account-management statements (`CREATE USER`, `GRANT`) rather than as inserts into the `mysql` system database
  - Capability of creating compressed output
  - Progress indicator (the values are estimates)
  - For dump file reloading, faster secondary index creation for `InnoDB` tables by adding indexes after rows are inserted

  For more information, see `mysqlpump` — A Database Backup Program.

  There are some notable differences between `mysqlpump` and `mysqldump`:

  - With no options, `mysqlpump` dumps everything, whereas `mysqldump` dumps nothing.
  - For `mysqlpump`, the `--routines` and `--events` are enabled by default, whereas for `mysqldump`, they are disabled by default.

Configuration Notes

- The default configuration for systemd now sets `LimitNOFILE` to 5000 to increase the number of file descriptors available to the MySQL server. This change applies to Linux systems on which MySQL installation is performed using RPM packages. On such systems, the number of descriptors available is often set by the operating system to 1024. The change causes the number of descriptors to match the `--open-files-limit` option default value of 5000. To configure a different number of descriptors, set `LimitNOFILE` as described at Managing MySQL Server with systemd. (Bug #21073014)

JSON Notes

- MySQL now supports a native `JSON` data type that enables efficient access to data in JSON (JavaScript Object Notation) documents. The `JSON` data type provides these advantages over storing JSON-format strings in a string column:

  - Automatic validation of JSON documents stored in `JSON` columns. Invalid documents produce an error.

  - Optimized storage format. JSON documents stored in `JSON` columns are converted to an internal format that permits efficient access to document elements.

  Along with the `JSON` data type, a set of SQL functions is available to enable operations on JSON values, such as creation, manipulation, and searching. In addition, the `CONVERT()` and `CAST()` functions can convert values between `JSON` and other types.

  For more information, see The JSON Data Type, and JSON Functions.

Optimizer Notes

- The optimizer now is able to use indexes on generated columns, even when queries do not refer to such columns directly by name. The optimizer recognizes query expressions that match definitions of generated columns and uses indexes from those columns as appropriate during query execution. For details, see Optimizer Use of Generated Column Indexes.

- The optimizer cost model has a new `memory_block_read_cost` parameter in the `mysql.engine_cost` table representing the cost of reading an index or data block from an in-memory database buffer.

  Together with the existing `io_block_read_cost` parameter representing the cost of reading a block from disk, this change enables cost models for data access methods to take into account the costs of reading information from different sources; that is, the cost of reading information from disk versus reading information already in a memory buffer. For the initial implementation, the default value of `memory_block_read_cost` is the same as `io_block_read_cost`. Tuning the values remains as future work, although you can change the values to see how that affects query performance. 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 this change into the `mysql` database.

- The optimizer hint capability introduced in MySQL 5.7.7 has been expanded to subquery execution strategies. Subquery hints affect whether to use semi-join transformations and which semi-join strategies to permit, and, when semi-joins are not used, whether to use subquery materialization or `IN`-to-`EXISTS` transformations. Examples:

  ```
  SELECT /*+ SEMIJOIN(FIRSTMATCH, LOOSESCAN) */ * FROM t1 ...;
  SELECT id, a IN (SELECT /*+ SUBQUERY(MATERIALIZATION) */ a FROM t1) FROM t2;
  SELECT * FROM t2 WHERE t2.a IN (SELECT /*+ SUBQUERY(INTOEXISTS) */ a FROM t1);
  ```

  For more information, see Subquery Optimizer Hints.

  There is also a new `duplicateweedout` flag for the `optimizer_switch` system variable. This flag enables use of `optimizer_switch` to specify whether to use the Duplicate Weedout semi-join strategy, which was not previously possible.

Packaging Notes

- For Windows, the MSI installer package no longer includes debugging binaries/information components (including PDB files). These are available in a separate Zip archive named `mysql-VERSION-winx64-debug-test.zip` for 64-bit and `mysql-VERSION-win32-debug-test.zip` for 32-bit. (Bug #18296012)

Performance Schema Notes

- Current-event timing now provides more information. Previously, while a wait, stage, statement, or transaction event was executing, the respective tables displayed the event with `TIMER_START` populated, but with `TIMER_END` and `TIMER_WAIT` as `NULL`:

```
events_waits_current
events_stages_current
events_statements_current
events_transactions_current
```

  To make it possible to determine how how long a not-yet-completed event has been running, the timer columns now are set as follows:

  - `TIMER_START` is populated (unchanged from previous behavior)
  - `TIMER_END` is populated with the current timer value
  - `TIMER_WAIT` is populated with the time elapsed so far (`TIMER_END` − `TIMER_START`)

  To find events that have not yet completed (that is, have no `END_EVENT_ID`) and have taken longer than *`N`* picoseconds thus far, monitoring applications can use this expression in queries:

```
WHERE END_EVENT_ID IS NULL AND TIMER_WAIT > N
```

  (Bug #75156, Bug #20889406)

- The Performance Schema incorporates these changes:

  - The `show_compatibility_56` system variable default value, previously `ON`, has been changed to `OFF`. Applications that require 5.6 behavior should set this variable to `ON` until such time as they have been migrated to the new behavior for system variables and status variables. See Migrating to Performance Schema System and Status Variable Tables
  - When the Performance Schema session variable tables produced output, they included no rows for global-only variables and thus did not fully reflect all variable values in effect for the current session. This has been corrected so that each table has a row for each session variable, and a row for each global variable that has no session counterpart. This change applies to the `session_variables` and `session_status` tables.
  - It is no longer required that the `show_compatibility_56` system variable be `OFF` for the Performance Schema system variable tables to produce output. The tables now produce output regardless of the variable value. This change applies to the `global_variables`, `session_variables`, and `variables_by_thread` tables.
  - `WHERE` clauses for `SHOW VARIABLES` and `SHOW STATUS` were deprecated in MySQL 5.7.6. This restriction has been lifted so that `WHERE` is supported as before 5.7.6.
  - The `metadata_locks` table now displays tablespace locks. Rows for these locks have an `OBJECT_TYPE` value of `TABLESPACE`.

- The Performance Schema logs wait, stage, statement, and transaction events in these history tables:

```
events_waits_history
events_waits_history_long
events_stages_history
events_stages_history_long
events_statements_history
events_statements_history_long
events_transactions_history
events_transactions_history_long
```

  Previously, historical event logging was controlled entirely by enabling or disabling history-related consumers in the `setup_consumers` table. These flags are global to the server, with the result that historical data was collected either for all threads or no threads.

  The Performance Schema now uses history consumers in conjunction with the `setup_actors` table to make it possible to control collection of historical events per host, user, or account (combination of host and user). This table has a new `HISTORY` column that indicates whether to collect historical events (subject also to which history consumers are enabled), and each new foreground thread is matched against rows in the table. If a matching row is found, its `HISTORY` value is recorded in the row for the thread in the `threads` table, which also now has a `HISTORY` column.

  Enabling historical event logging for a given session can be done independent of enabling instrumentation for it. Consequently, you can control more precisely what events are logged in history tables, with these advantages:

  - A decrease in runtime overhead when historical data is needed only for a subset of the instrumented sessions.
  - A reduction of noise in the history tables, facilitating troubleshooting on busy servers that generate a large number of events.

  For more information, see Pre-Filtering by Thread, The setup_actors Table, and The threads Table.

- The `threads` table now contains a `CONNECTION_TYPE` column that indicates the connection protocol. It can be used to determine how the connection was made. Permitted values are `TCP/IP` (TCP/IP connection established without SSL), `SSL/TLS` (TCP/IP connection established with SSL), `Socket` (Unix socket file connection), `Named Pipe` (Windows named pipe connection), and `Shared Memory` (Windows shared memory connection).

  Connection-type information is also written to the general query log for new connections, and the audit log interface was revised to incorporate the connection type.

  For more information, see The threads Table, The General Query Log, and Writing Audit Plugins.

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.

References: See also: Bug #76167, Bug #20652173, Bug #20684424, Bug #20811494.

Plugin Service Notes

- MySQL server plugins have access to server “services,” as described in MySQL Services for Plugins. MySQL distributions now include plugins that demonstrate how to test plugin service APIs. The `test_framework` plugin is a bare bones plugin that shows the minimum required framework for service testing. The `test_services` and `test_services_threaded` plugins demonstrate how to test the

`my_snprintf` and `my_plugin_log_service` services in unthreaded and threaded contexts. For more information, see Plugins for Testing Plugin Services, in The MySQL Test Framework, Version 2.0.

- MySQL distributions now provide a locking interface that implements locks with three attributes: Lock namespace, lock name, and lock mode. The namespace enables different applications to use the same lock names without colliding by creating locks in separate namespaces. Locks can be created with a mode of either read (shared) or write (exclusive).

  This locking interface is available at two levels: 1) As a C language interface, callable as a plugin service from server plugins or user-defined functions; 2) At the SQL level, as a set of user-defined functions that map onto calls to the service routines. For more information, see The Locking Service.

  The interface provided by the locking service is distinct from that provided by `GET_LOCK()` and related SQL functions (see Miscellaneous Functions). For example, `GET_LOCK()` does not implement namespaces and provides only exclusive locks, not distinct read and write locks.

Plugin Notes

- The initial implementation for query rewrite plugins used its own API. This API has been reimplemented to use the audit plugin API. For more information, see Writing Audit Plugins. One effect of the query rewrite plugin reimplementation is reduced overhead.

- MySQL distributions now include Version Tokens, a feature that enables creation of and synchronization around server tokens that applications can use to prevent accessing incorrect or out-of-date data. Version Tokens is based on a plugin library that implements a `version_tokens` plugin and a set of user-defined functions. For more information, see Version Tokens.

- These changes were made for the `Rewriter` query rewrite plugin (see The Rewriter Query Rewrite Plugin):

  - There is now a single installation script, `install_rewriter.sql`. Previously, there were two installation scripts, `install_rewriter.sql` and `install_rewriter_with_optional_columns.sql`, which differed in whether they created the `pattern_digest` and `normalized_columns` columns of the `rewrite_rules` table. `install_rewriter.sql` now always creates those columns, so there is no need for `install_rewriter_with_optional_columns.sql`.

  - The `enabled` column of the `rewrite_rules` table is now defined as `ENUM('YES','NO')` rather than as `CHAR(1)`. Correspondingly, to enable a rule, set this column to `YES` rather than `Y`.

  To upgrade if you have previously installed the `Rewriter` plugin, uninstall it by running the uninstallation script first, then run the installation script. After reinstalling, load your rewrite rules again (this is necessary because uninstalling drops the rules table). For instructions, see Installing or uninstalling the Rewriter Query Rewrite Plugin.

Security Notes

- Community Edition RPM packages now invoke `mysql_ssl_rsa_setup` during installation to create default SSL and RSA key and certificate files. (Bug #20855737)

- A new system variable, `require_secure_transport`, enables administrators to require all client connections to the server to be made using some form of secure transport. Qualifying connections are TCP/IP connections that use SSL, or connections that use a socket file (on Unix) or shared memory (on Windows). When this variable is enabled, the server rejects nonsecure connection attempts, which fail with an `ER_SECURE_TRANSPORT_REQUIRED` error.

This capability supplements per-account SSL requirements, which take precedence. For exmaple, if an account is defined with `REQUIRE SSL`, enabling `require_secure_transport` does not make it possible to use the account to connect using a Unix socket file.

Spatial Data Support

- `ST_NumInteriorRing()` was added as more a standard-compliant alias of `ST_NumInteriorRings()`. (Bug #77598, Bug #21362781)
- All spatial computations now are done using Boost.Geometry functions. All older non-Boost-based algorithms have been removed. (Bug #77444, Bug #21300713)
- Geometry constructor functions that take WKT or WKB values (such as `ST_GeomFromText()` and `ST_GeomFromWKB()`) did not check for trailing garbage bytes. They now reject trailing nonwhitespace characters and produce an error. (Bug #77244, Bug #21198064)
- The required version of the Boost library for server builds has been raised from 1.57.0 to 1.58.0. (Bug #76354, Bug #20721087)
- Geometry object constructor functions such as `Point()` and `MultiPolygon()` now are stricter about rejecting invalid arguments. (Bug #76337, Bug #20712775)

SQL Mode Notes

- In MySQL 5.7.4, the `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` SQL modes were changed so that they did nothing when named explicitly. Instead, their effects were included in the effects of strict SQL mode (`STRICT_ALL_TABLES` or `STRICT_TRANS_TABLES`). The intent was to reduce the number of SQL modes with an effect dependent on strict mode and make them part of strict mode itself.

  However, the change to make strict mode more strict by including `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` caused some problems. For example, in MySQL 5.6 with strict mode but not `NO_ZERO_DATE` enabled, `TIMESTAMP` columns can be defined with `DEFAULT '0000-00-00 00:00:00'`. In MySQL 5.7.4 with the same mode settings, strict mode includes the effect of `NO_ZERO_DATE` and `TIMESTAMP` columns cannot be defined with `DEFAULT '0000-00-00 00:00:00'`. This causes replication of `CREATE TABLE` statements from 5.6 to 5.7.4 to fail if they contain such `TIMESTAMP` columns.

  The long term plan is still to have the three affected modes be included in strict SQL mode and to remove them as explicit modes in a future MySQL release. But to restore compatibility in MySQL 5.7 with MySQL 5.6 strict mode and to provide additional time for affected applications to be modified, the following changes have been made:

  - `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` again have an effect when named explicitly. This reverts a change made in MySQL 5.7.4.
  - `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` are no longer part of strict SQL mode. This reverts a change made in MySQL 5.7.4.
  - `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` are now included in the default `sql_mode` value, which as a result includes these modes: `ONLY_FULL_GROUP_BY`, `STRICT_TRANS_TABLES`, `NO_ZERO_IN_DATE`, `NO_ZERO_DATE`, `ERROR_FOR_DIVISION_BY_ZERO`, `NO_AUTO_CREATE_USER`, and `NO_ENGINE_SUBSTITUTION`.

  With the preceding changes, stricter data checking is still enabled by default, but the individual modes can be disabled in environments where it is currently desirable or necessary to do so.

Although ERROR_FOR_DIVISION_BY_ZERO, NO_ZERO_DATE, and NO_ZERO_IN_DATE again can be used separately from strict mode, it is intended that they be used together. As a reminder, a warning now occurs if they are enabled without also enabling strict mode or vice versa.

References: See also: Bug #75439, Bug #20367829.

Functionality Added or Changed

- **InnoDB:** The adaptive hash index search system is now partitioned, with each index bound to a specific partition, and each partition protected by a separate latch. Partitioning is controlled by the innodb_adaptive_hash_index_parts configuration option.

  Prior to MySQL 5.7.8, the adaptive hash index search system was protected by a single latch (btr_search_latch) which could become a point of contention. To reduce contention, innodb_adaptive_hash_index_parts is set to 8 by default. The maximum setting is 512. (Bug #20985298)

- **InnoDB:** The new innodb_log_checksum_algorithm option specifies how to generate and verify the checksum stored in redo log disk blocks. innodb_log_checksum_algorithm supports same algorithms as innodb_checksum_algorithm, which include innodb, crc32, none, and their associated strict forms. Previously, only the innodb algorithm was supported for redo log disk blocks. innodb_log_checksum_algorithm=innodb is the default setting. Thanks to Alexey Kopytov for the patch. (Bug #20531208, Bug #75595)

- **InnoDB:** InnoDB now supports secondary indexes on virtual generated columns. For more information, see Secondary Indexes and Generated Virtual Columns.

- **InnoDB:** Internal server-layer functions were added to allow InnoDB purge threads to construct and destroy thread handle objects, and to compute virtual generated column index values when a table object is not present. This enhancement was required to support secondary indexes on virtual generated columns.

- **InnoDB:** Virtual generated column values no longer occupy space in database rows. With this change, a table rebuild is no longer required when adding or dropping virtual generated columns. Only a system table update is necessary, to register the new metadata.

  Virtual generated columns are still represented in InnoDB metadata. The N_COLS field of INNODB_SYS_TABLES still counts virtual generated columns, and INNODB_SYS_COLUMNS still includes virtual generated column metadata.

  A new INFORMATION_SCHEMA table, INNODB_SYS_VIRTUAL, provides metadata about columns upon which virtual generated columns are based.

- **InnoDB:** InnoDB now supports page-level compression for file-per-table tablespaces. Page compression is enabled by specifying the COMPRESSION attribute when creating or altering a table. Supported compression algorithms include Zlib and LZ4. This feature, which is referred to as transparent page compression, relies on sparse file and hole punching support. It is supported on Windows with NTFS, and a subset of MySQL-supported Linux platforms where the kernel level provides hole punching support.

  For more information about this feature, see InnoDB Page Compression.

- **InnoDB:** The new innodb_flush_sync configuration option, which is enabled by default, causes the innodb_io_capacity setting to be ignored for bursts of I/O activity that occur at checkpoints. To adhere to the limit on InnoDB background I/O activity defined by the innodb_io_capacity setting, disable innodb_flush_sync.

- **InnoDB:** The default value for `innodb_purge_threads` and `innodb_page_cleaners` was changed from 1 to 4. If the number of page cleaner threads exceeds the number of buffer pool instances, `innodb_page_cleaners` is automatically set to the same value as `innodb_buffer_pool_instances`.

- **Replication:** The behavior of `SET GTID_PURGED` has been changed so that it does not add any GTIDs to `Previous_gtids_log_event` and does not rotate the binary log. Instead the GTIDs are added to the `mysql.gtid_executed` table. This fix ensures that it is safe in all cases to use `binlog_gtid_simple_recovery=1` for a server using MySQL 5.7.8 or later, where all binary logs were generated by servers using MySQL 5.7.8 or later. (Bug #75767, Bug #20470724)

- **Replication:** When using a multi-threaded slave, each worker thread has its own queue of transactions to process. In previous MySQL versions, `STOP SLAVE` waited for all workers to process their entire queue. This logic has been changed so that `STOP SLAVE` first finds the newest transaction that was committed by any worker thread. Then, it waits for all workers to complete transactions older than that. Newer transactions are not processed. The new logic allows `STOP SLAVE` to complete faster in case some worker queues contain multiple transactions. (Bug #75525, Bug #20369401)

- Solaris tarball and PKG distributions no longer have `-gcc` in the distribution file names. (Bug #21047137)

- Previously, the `max_digest_length` system variable controlled the maximum digest length for all server functions that computed statement digests. However, whereas the Performance Schema may need to maintain many digest values, other server functions such as query rewrite plugins need only one digest per session. Increasing the `max_digest_length` value has little impact on total memory requirements for those functions, but can increase Performance Schema memory requirements significantly. To enable configuring digest length separately for the Performance Schema, its digest length is now controlled by the new `performance_schema_max_digest_length` system variable. (Bug #20963147)

- The server now prints more descriptive diagnostic messages for bad values of `secure_file_priv`. (Bug #20771331)

- The `libmysqld` embedded server took its default `secure_file_priv` value from the `INSTALL_SECURE_FILE_PRIVDIR` `CMake` option, but cannot share the same directory with a non-embedded server. The new `INSTALL_SECURE_FILE_PRIV_EMBEDDEDDIR` option enables a separate directory to be specified for `libmysqld`. The default value is `NULL`. (Bug #20770671)

- `my_print_defaults` now masks passwords. To display passwords in cleartext, use the new `--show` option. In addition, The output for client programs invoked with the `--print-defaults` option now masks passwords. (Bug #19953365, Bug #20903330)

- For attempts to create a multiple-column `SPATIAL` index, the server previously returned an “Incorrect arguments to SPATIAL INDEX” error. Now it returns `ER_TOO_MANY_KEY_PARTS` (“Too many key parts specified; max 1 parts allowed”). (Bug #18320371)

- For tables that contain object information, the Performance Schema now uses lowercase stored program names. (Bug #17818062)

- To make the effect of password-change operations more clear, `mysql_secure_installation` now displays the user whose password is being changed. (Bug #17343687)

- The patch number of the C client library is now increased for each patch version of the server. This number has the format *`major.minor.patch`*. (Bug #77544, Bug #21341481)

- The `max_statement_time` system variable was renamed to `max_execution_time`. The `Max_statement_time_exceeded`, `Max_statement_time_set`,

and `Max_statement_time_set_failed` status variables were renamed to `Max_execution_time_exceeded`, `Max_execution_time_set`, and `Max_execution_time_set_failed`.

The `MAX_STATEMENT_TIME` option for `SELECT` statements was removed because its functionality is now available using the more general optimizer hint syntax (see Optimizer Hints). Statements that begin like this:

```
SELECT MAX_STATEMENT_TIME = N ...
```

Should be rewritten to begin like this:

```
SELECT /*+ MAX_EXECUTION_TIME(N) */ ...
```

There are some minor implementation differences between the two. `MAX_STATEMENT_TIME` was not permitted in non-top-level `SELECT` statements such as subqueries, or in stored programs, and produced an error. `MAX_EXECUTION_TIME()` is permitted in those contexts, but is ignored. (Bug #77461, Bug #21306646, Bug #77460, Bug #21306392, Bug #77459, Bug #21306319)

- `GeometryCollection()` with no arguments is now permitted as a way to create an empty geometry. (Bug #77114, Bug #21127270)
- The shutdown timeout value in `/etc/init.d/mysqld` was too short for some environments. The value has been increased from 60 seconds to 600 seconds. (Bug #76900, Bug #20987568)
- Use of the optimizer cost model was extended to estimating index scan costs within `test_if_cheaper_ordering()` for the I/O cost of accessing table blocks. (Bug #76804, Bug #20947871)
- For MySQL install operations on OS X from DMG packages, if a a random `root` account password is generated, it now is displayed in a dialog box. (Bug #76792, Bug #20930305)
- `mysqldump` no longer dumps the `sys` schema by default. It is still possible to dump it by naming it explicitly on the command line (for example, `mysqldump --databases sys`). (Bug #76735, Bug #20902791)
- For non-TCP/IP connections, these changes were made when `--ssl` was specified to force SSL to be used:
  - For named pipe and shared memory connections, attempts to use SSL now produce an error because these connections use non-network protocols.
  - For Unix socket file connections, SSL does not add any security. The connection is permitted, but the `mysql` client now produces a warning that SSL does not add anything for this connection protocol.

  (Bug #76508, Bug #20785409, Bug #21025587)
- `mysql_ssl_rsa_setup` now has a `--uid=name` option that enables specifying the owner for any files created by the program (if the program is executed as `root`). (Bug #76369, Bug #20726413)
- MySQL distributions now include an `innodb_stress` suite of test cases. Thanks to Mark Callaghan for the contribution. (Bug #76347, Bug #20717127)
- The data type for generated columns now permits the `COLLATE` attribute. (Bug #76329, Bug #20709487)
- Connections for the `FEDERATED` storage engine now set the `program_name` session connection attribute to `federated` to permit identification of the connection source. (Bug #68781, Bug #16555730)

- Previously, changes to the `validate_password` plugin dictionary file (named by the `validate_password_dictionary_file` system variable) while the server was running required a restart for the server to recognize the changes. Now `validate_password_dictionary_file` can be set at runtime and assigning a value causes the named file to be read without a restart.

  In addition, two new status variables are available. `validate_password_dictionary_file_last_parsed` indicates when the dictionary file was last read, and `validate_password_dictionary_file_words_count` indicates how many words it contains. (Bug #66697, Bug #14588145)

- The error produced for a `COM_FIELD_LIST` command with too much data was changed from `ER_UNKNOWN_COM_ERROR` to the more informative `ER_MALFORMED_PACKET`. (Bug #53699, Bug #11761229)

- A new system variable, `disabled_storage_engines`, enables administrators to designate storage engines that cannot be used to create new tables or tablespaces. By default, this variable is empty (no engines disabled), but it can be set to a comma-separated list of one or more engines. Any engine named in the value cannot be used to create tables or tablespaces with `CREATE TABLE` or `CREATE TABLESPACE`, and cannot be used with `ALTER TABLE ... ENGINE` or `ALTER TABLESPACE ... ENGINE` to change the storage engine of existing tables or tablespaces. Attempts to do so result in an `ER_DISABLED_STORAGE_ENGINE` error.

  `disabled_storage_engines` does not restrict other DDL statements for existing tables, such as `CREATE INDEX`, `TRUNCATE TABLE`, `ANALYZE TABLE`, `DROP TABLE`, or `DROP TABLESPACE`. This permits a smooth transition so that existing tables or tablespaces that use a disabled engine can be migrated to a permitted engine by means such as `ALTER TABLE ... ENGINE` *`permitted_engine`*.

- MySQL distributions no longer include the `sql-bench` directory. The `INSTALL_SQLBENCHDIR` `CMake` option has also been removed.

  References: See also: Bug #21303289.

- The default value of the `table_open_cache_instances` system variable has been increased from 1 to 16.

- The `read_only` system variable enables the server to be put into read-only mode, in which the server permits client updates only from users who have the `SUPER` privilege. A new `super_read_only` system variable, if enabled, prohibits client updates even from users who have `SUPER`. “Super” read-only mode can be useful for operations such as preparing a server for a move or upgrade because it prevents all client users from modifying data, even administrators.

- The so-called “fast mutex” code has been removed from the server sources. It provides no measurable benefit, complicates the code, and is problematic for certain architectures such as POWER8. The (undocumented) `WITH_FAST_MUTEXES` `CMake` option has also been removed.

  References: See also: Bug #37703, Bug #11748914, Bug #72806, Bug #18871517, Bug #72807, Bug #18871138, Bug #72805, Bug #18870931.

- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.

- `mysqld_safe` no longer uses the data directory as a possible location for setting `MYSQL_HOME`. (This has been deprecated since MySQL 5.0.)

Bugs Fixed

- **Incompatible Change:** The `mysql_parser` plugin service interface defined in the `service_parser.h` header file was incorrect for the case that the `MYSQL_DYNAMIC_PLUGIN` macro was defined. The test for this symbol used the wrong name, so the interface did not enable the proper code. The test has been corrected, and some adjustments made to the API for function pointer members within the `mysql_parser_service_st` structure:

  - Some function pointer names began with `mysql_parser_`, others with `mysql_`. For consistency, function pointer member names that began with `mysql_parser_` were changed to begin with `mysql_`.
  - The missing `mysql_get_statement_digest` function pointer member was added.

  These modifications change the service API. Any plugin to be used with this version of MySQL that relies on the service must be recompiled. (Bug #20856729)

- **Incompatible Change:** Internal storage format for `VIRTUAL` generated columns was modified. For `MyISAM` tables with such columns, this is an incompatible change; for upgrades, use `ALTER TABLE` to drop the columns before the upgrade and add them again after the upgrade. (Bug #77312, Bug #21237637)

- **InnoDB; Partitioning:** In certain rare cases the optimizer pruned all partitions for an `InnoDB` table but failed to remove the table from consideration as a source for obtaining matches, instead calling for the table to be initialized and prepared for fetching records. This occurred when the active index was not set during initialization. To fix this problem, we now set the active index ID during initialization even if there are no partitions to select from. This behavior also matches the way the same case is already handled in MySQL 5.6. (Bug #21211524)

- **InnoDB; Partitioning:** Sorted index reads on partitioned `InnoDB` tables added rows to the prefetch cache, which could cause rows from the wrong partition since the prefetch cache does not support partitioned tables. Now the the prefetch cache is disabled in such cases. (Bug #20584754)

- **InnoDB; Partitioning:** `ALTER TABLE ADD UNIQUE INDEX` failed when run concurrently with an `INSERT` on the same partitioned `InnoDB` table. (Bug #20510811, Bug #75834)

- **InnoDB; Partitioning:** The `CREATE_TIME` column of the `INFORMATION_SCHEMA.TABLES` table now shows the correct table creation time for partitioned `InnoDB` tables. The `CREATE_TIME` column of the `INFORMATION_SCHEMA.PARTITIONS` table now shows the correct partition creation time for a partition of partitioned `InnoDB` tables.

  The `UPDATE_TIME` column of the `INFORMATION_SCHEMA.TABLES` table now shows when a partitioned `InnoDB` table was last updated by an `INSERT`, `DELETE`, or `UPDATE`. The `UPDATE_TIME` column of the `INFORMATION_SCHEMA.PARTITIONS` table now shows when a partition of a partitioned `InnoDB` table was last updated. (Bug #69990, Bug #17299181)

- **InnoDB:** With `innodb_strict_mode=OFF`, a `CREATE TEMPORARY TABLE ... ROW_FORMAT=Compressed TABLESPACE=innodb_file_per_table DATA DIRECTORY ...` statement raised an assertion. The `DATA DIRECTORY` clause is not supported with temporary tables and should be ignored. (Bug #21324507, Bug #77495)

- **InnoDB:** The `btr_search_drop_page_hash_index` function dereferenced the adaptive hash index block before acquiring a latch, which could result in a race condition. (Bug #21310520)

- **InnoDB:** A regression introduced in MySQL 5.7.2 caused an `innochecksum`-related memory leak. (Bug #21255718)

- **InnoDB:** The `records_in_range` function returned a constant value for spatial indexes. (Bug #21245805, Bug #77332)

- **InnoDB:** In some cases, memory was not properly allocated for `rw_lock_t` instances. (Bug #21242541)
- **InnoDB:** `SHOW ENGINE INNODB STATUS` no longer reports mutex metrics. Mutex metrics are now reported by `SHOW ENGINE INNODB MUTEX`. (Bug #21238953, Bug #77314)

  References: See also: Bug #21052754, Bug #21266784.
- **InnoDB:** Functionality required to build adaptive hash indexes on field prefixes was reintroduced to improve sequential insert performance. The functionality was removed in MySQL 5.7.2 by the fix for Bug #21198396. (Bug #21198396, Bug #77246)
- **InnoDB:** When defining `buf_block_t`, a lock and a mutex were often accessed in the same vicinity, which could cause unintended cache line sharing. (Bug #21153684)
- **InnoDB:** The `ib_cursor_moveto` function did not accept a search tuple with fewer fields than are defined for the index. (Bug #21121197, Bug #77083)
- **InnoDB:** The `ib_table_truncate` function failed to release a transaction, resulting in a hang on server shutdown. (Bug #21121164, Bug #77084)
- **InnoDB:** The `ib_open_table_by_id` function passed an incorrect argument to `dict_table_open_on_id`. (Bug #21121084, Bug #77100)
- **InnoDB:** On Unix-like platforms, `os_file_create_simple_no_error_handling_func` and `os_file_create_func` opened files in different modes when `innodb_flush_method` was set to `O_DIRECT`. (Bug #21113036, Bug #76627)
- **InnoDB:** A cascade operation resulted in a duplicate entry error in `FTS_DOC_ID_INDEX`. The same document ID was used by two cascade operations. (Bug #21111301, Bug #77087)
- **InnoDB:** Starting the server with an invalid `innodb_data_file_path` setting did not produce a sufficiently informative error message. (Bug #21103446, Bug #77056)
- **InnoDB:** A regression introduced in MySQL 5.7.5 caused sorting to be skipped when rebuilding a table after dropping a single-column primary key. (Bug #21103101)
- **InnoDB:** Opening a foreign key-referenced table with `foreign_key_checks` enabled resulted in an error when the table or database name contained special characters. (Bug #21094069, Bug #77043)
- **InnoDB:** The `page_zip_verify_checksum` function returned false for a valid compressed page. (Bug #21086723)
- **InnoDB:** DDL operations for tablespaces could fail to implicitly commit the current transaction. (Bug #21081898)
- **InnoDB:** The rollback of a partially completed transaction containing more than one update to a spatial index raised an assertion in `row_ins_sec_index_entry_by_modify()`. (Bug #21076238)
- **InnoDB:** In the case of a lock conflict, shutdown could hang waiting for asynchronous rollback to finish. (Bug #21075892)
- **InnoDB:** To avoid conflicts with implicitly created file-per-table tablespaces, `CREATE TABLESPACE ... ADD DATAFILE` no longer supports creation of tablespace data files in subdirectories under the MySQL data directory (`datadir`). Additionally, the data file path specified in a `CREATE TABLESPACE ... ADD DATAFILE` statement must be an existing directory. `InnoDB` no longer creates missing directories for `CREATE TABLESPACE ... ADD DATAFILE` statements. (Bug #21068487, Bug #77002)

- **InnoDB:** `SHOW ENGINE INNODB MUTEX` functionality, which was removed in MySQL 5.7.2, was revised and added back. Mutex statistics collection can now be configured dynamically using the following options:

  - To enable the collection of mutex statistics, run:

    ```
    SET GLOBAL innodb_monitor_enable='latch';
    ```

  - To reset mutex statistics, run:

    ```
    SET GLOBAL innodb_monitor_reset='latch';
    ```

  - To disable the collection of mutex statistics, run:

    ```
    SET GLOBAL innodb_monitor_disable='latch';
    ```

  For more information, see SHOW ENGINE Syntax.

  Sync debug checking for the `InnoDB` storage engine, previously defined under `UNIV_SYNC_DEBUG`, is now defined under `UNIV_DEBUG` and is available when debugging support is compiled in using the `WITH_DEBUG CMake` option. When debugging support is compiled in, `InnoDB` sync debug checking is controlled by the `innodb_sync_debug` configuration option. (Bug #21052754)

- **InnoDB:** A tablespace opened locally by a truncate routine was not closed in protected mode. (Bug #21046968)

- **InnoDB:** An assertion was raised when truncation logic identified inactive undo tablespaces as candidates for undo log truncation. Some undo tablespaces were left inactive when the number of available undo tablespaces exceeded the number of undo logs. (Bug #21046781)

- **InnoDB:** At startup, `InnoDB` updated the `SYS_DATAFILES` internal system table with the `space_id` and path of each system tablespace file even though a record was already present and `SYS_DATAFILES.PATH` matched the current value. (Bug #21044191)

- **InnoDB:** In Windows debug builds, an `innodb_flush_method` setting of `normal` or `unbuffered` raised an assertion. (Bug #20981684)

- **InnoDB:** Transactions could be subjected to rollback while performing DDL operations. The transactions were not marked as DDL transactions, and `TRX_FORCE_ROLLBACK_DISABLE` was not set. (Bug #20979020)

- **InnoDB:** An `ALTER TABLE ... IMPORT TABLESPACE` operation on a table with prefix index failed with a schema mismatch error. (Bug #20977779, Bug #76877)

- **InnoDB:** Initializing the database with an `innodb_page_size` setting of 64K and a system tablespace data file size less than 12MB raised an assertion. With an `InnoDB` page size of 64KB, the first system tablespace data file (`ibdata1`) was not large enough to contain the doublewrite buffer blocks (block1 and block2). To ensure that the doubwrite buffer blocks fit within the first system tablespace data file, a minimum data file size is now enforced. If `innodb_page_size` is less than or equal to 16KB, the minimum data file size for the first system tablespace data file (`ibdata1`) is 3MB. For `innodb_page_size=32k`, the minimum data file size is 6MB. For `innodb_page_size=64k`, the minimum data file size is 12MB. (Bug #20972309)

- **InnoDB:** Full-text search operations between tables related by a foreign key constraint were not properly cascaded during iterative DML cascading operations. (Bug #20953265, Bug #76817)

- **InnoDB:** In MySQL 5.7.2, 32 of 128 undo logs (rollback segments) were reserved as non-redo undo logs for temporary table transactions. With one of the remaining undo logs always allocated to the system

tablespace, 95 undo logs remained available for allocation to the system tablespace or separate undo tablespaces. This change effectively reduced the `innodb_undo_tablespaces` maximum limit from 126 to 95. In other words, a limit of 95 available undo logs also limited the maximum number of undo tablespaces to 95. In MySQL 5.7.8, the `innodb_undo_tablespaces` maximum value is officially reduced to 95. (Bug #20938115)

- **InnoDB:** A memory leak occurred when a foreign key constraint object was loaded with the parent table while the child table failed to load. The foreign key constraint object should only be loaded with the child table. (Bug #20926253, Bug #21041449)
- **InnoDB:** Debug only code set `m_prebuilt`, which may affect in-place `ALTER TABLE` behaviour. `m_prebuilt` to should not be used to store an intermediate value in debug-only code. (Bug #20921940, Bug #76774)
- **InnoDB:** The definition of the `fil_node_t` data structure was moved from `fil0fil.cc` to `fil0fil.h` so that diagnostic code outside that module can access information about files that belong to a tablespace. (Bug #20886222, Bug #76694)
- **InnoDB:** Assertion code was revised to avoid compiler warnings that occurred when compiling MySQL on Mac OS X 10.10.2. (Bug #20883256, Bug #76690)
- **InnoDB:** After upgrading to MySQL 5.7.6 or later, restarting the server after truncating a table that was originally created in MySQL 5.7.5 or earlier raised an “incorrect MERGE_THRESHOLD length in SYS_INDEXES” error. A `MERGE_THRESHOLD` column was added to the internal `SYS_INDEXES` table in MySQL 5.7.6. `SYS_INDEXES` records for tables that were not rebuilt or imported after upgrading did not include the new column. The `TRUNCATE TABLE` operation updated the table's `SYS_INDEXES` records to include the new column but set the `MERGE_THRESHOLD` value to NULL. (Bug #20882432)
- **InnoDB:** A failure to load a change buffer bitmap page during a concurrent delete tablespace operation caused a server exit. (Bug #20878735)
- **InnoDB:** A shutdown hang occurred when an `innodb_force_recovery` setting of 3 or higher prevented the rollback of transactions that were in an `ACTIVE` state. `ACTIVE` transactions are now placed in `XA PREPARE` state in the main-memory data structure to allow shutdown to proceed normally. The transactions are recovered as `ACTIVE` on the next restart and are rolled back unless `innodb_force_recovery` is again set to 3 or higher. (Bug #20874411)
- **InnoDB:** If a server exit occurred during an `XA ROLLBACK`, the transaction was incorrectly recovered in `XA PREPARE` state. As a result, subsequent `XA COMMIT` transactions were possible, which would break ACID compliance and potentially cause corruption between indexes of a table. (Bug #20872655, Bug #76672)
- **InnoDB:** A `CREATE TABLESPACE` operation raised a Valgrind error due to a memory leak in the `os_create_subdirs_if_needed` function. (Bug #20865674)
- **InnoDB:** Calls to `buf_page_print()` were removed to avoid filling `mysql-test-run` logs with `InnoDB` page dumps. Page dumps related to file I/O are still printed. (Bug #20863042)
- **InnoDB:** `CREATE TABLESPACE` failed to move internal tablespace files to a reserved name space that starts with an `innodb_` prefix, permitting internal tablespace files to be dropped. (Bug #20840368, Bug #76603)
- **InnoDB:** A `TRUNCATE TABLE` operation on a general tablespace table with a full-text search index raised an assertion. (Bug #20834483)
- **InnoDB:** An assertion was raised on shutdown due to XA PREPARE transactions holding explicit locks. (Bug #20816223, Bug #76567)

- **InnoDB:** The `fts_print_doc_id` function printed too much debug information in debug builds. `fts_enable_diag_print` is now used instead. (Bug #20811125)
- **InnoDB:** After a failed `DROP TABLE` operation, the purge background thread asserted while attempting to access an index page of the table. Purge should not attempt to clean a table that is marked as corrupt. Purge now checks for a corrupt primary index. (Bug #20789078, Bug #75913)
- **InnoDB:** A checksum mismatch error on a `crc32` checksum was encountered when restarting the server on a data file copied from a machine with a different endianness. The `crc32` checksum should be recognized regardless of the native byte order of the system where the checksum was generated. (Bug #20783098, Bug #76375)
- **InnoDB:** An `ALTER TABLE ... DROP INDEX` operation on a table with foreign key dependencies raised an assertion. (Bug #20768847)
- **InnoDB:** An assertion was raised when `InnoDB` attempted to dereference a NULL foreign key object. (Bug #20762798)
- **InnoDB:** An `ALTER TABLE` operation raised an assertion due a regression introduced in MySQL 5.7.6 with the introduction of native partitioning support for general tablespaces. (Bug #20759613)
- **InnoDB:** In some instances, functions that call `DBUG_ENTER` did not call `DBUG_RETURN`. (Bug #20753620, Bug #76447)
- **InnoDB:** A regression of the `trx_is_started` function in MySQL 5.7.6 caused a shutdown hang. (Bug #20744155)
- **InnoDB:** The MeCab full-text plugin parser failed to handle an out-of-memory exception. (Bug #20742590)
- **InnoDB:** An assertion was raised during a rollback operation due to a record the was incorrectly undelete-marked. (Bug #20734998)
- **InnoDB:** An assertion was raised in a debug build when an `ALTER TABLE` operation invoked obsolete foreign key code while attempting to create an optimized temporary table as part of an optimizer plan. Temporary tables do not support foreign keys. Invocation of foreign key code is now blocked for optimized temporary tables. (Bug #20730289)
- **InnoDB:** An `INSERT` operation raised an assertion. The calculation that determines the number of extents to reserve when storing a `BLOB` did account for compressed pages. (Bug #20713559)
- **InnoDB:** Missing brackets in the `fsp_flags_is_valid` function could result in a failure to recognize a corrupted data file. (Bug #20671465)
- **InnoDB:** A query that used a percentage character '%' as the last character in a query token raised a full-text parser plugin assertion. Full-text parser plugins created using the full-text parser plugin framework now ignore the '%' character if specified as the first or last character in a query token. Using the '%' character as the first character in a query token is not permitted by the internal SQL parser. Using the '%' character as last character in a query token is reserved for prefix matching. (Bug #20668156)
- **InnoDB:** The `INFORMATION_SCHEMA.FILES` table now reports metadata for all `InnoDB` tablespace types including file-per-table tablespaces, general tablespaces, the system tablespace, temporary table tablespaces, and undo tablespaces (if present). System tablespace and temporary table tablespace metadata is no longer reported by the `INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES` and `INFORMATION_SCHEMA.INNODB_SYS_DATAFILES` tables. However, these tables continue to provide metadata for file-per-table and general tablespaces. (Bug #20660744, Bug #21086257, Bug #77032, Bug #76182)

- **InnoDB:** The `InnoDB` full-text search feature with the mecab parser plugin would print an empty error message. (Bug #20651493, Bug #76164)
- **InnoDB:** Importing a tablespace with a full-text index resulted in an assertion when attempting to rebuild the index. (Bug #20637494)
- **InnoDB:** A DML operation raised an assertion in `btr_estimate_n_rows_in_range()`. The assertion code was too strict. (Bug #20618309)
- **InnoDB:** Defining a user-created `FTS_DOC_ID` column as a primary key produced incorrect full-text search relevancy rankings. (Bug #20597981)
- **InnoDB:** During a table import operation, an `INSERT` failed with a duplicate key error on an `AUTO_INCREMENT` column due to an incorrectly initialized `AUTO_INCREMENT` value. (Bug #20597821, Bug #76037)
- **InnoDB:** After dropping a full-text search index, the hidden `FTS_DOC_ID` and `FTS_DOC_ID_INDEX` columns prevented online DDL operations. (Bug #20590013, Bug #76012)
- **InnoDB:** An assertion was raised on server startup when `InnoDB` tried to create a temporary file in a non-existent temporary directory (`tmpdir`) while in read-only mode. (Bug #20578834)
- **InnoDB:** The `innodb_checksum_algorithm` `strict_*` settings (`strict_none`, `strict_innodb`, and `strict_crc32`) caused the server to halt when `InnoDB` encountered a valid but non-matching checksum. For example, with `innodb_checksum_algorithm=strict_crc32`, a valid `innodb` checksum would cause the server to halt. Now, instead of halting the server, `InnoDB` only prints an error message. (Bug #20568464)
- **InnoDB:** After moving the MySQL data directory and modifying the `datadir` configuration parameter to point to the new location, tables stored in general tablespaces failed to open because the tablespace data file could not be found. To address this problem, `CREATE TABLESPACE ... ADD DATAFILE` now creates an `isl` file in the MySQL data directory when a general tablespace data file is created outside of the MySQL data directory. Also, the `fil_ibd_open` function now searches for general tablespaces in the same way that it searches for file-per-table tablespaces. (Bug #20563954)
- **InnoDB:** General tablespaces created on Windows using a relative data file path could not be opened on Unix-like systems. `InnoDB` failed to convert the backslash (“\”) directory separator that is used in the Windows version of the relative data file path. (Bug #20555168)
- **InnoDB:** General tablespaces now support partitioned `InnoDB` tables, and individual partitions and subpartitions can now be assigned to a general tablespace. `SHOW CREATE TABLE` output was revised to include quotes around the tablespace identifier. (Bug #20554858, Bug #20588947)
- **InnoDB:** An `ALTER TABLE` operation that added a spatial index caused the server to exit. (Bug #20547644)
- **InnoDB:** Assertion code which checks for the lowest possible page number for a tablespace did not account for general tablespaces. (Bug #20544581, Bug #20810627)
- **InnoDB:** The `InnoDB` `memcached` plugin handled unsigned NOT NULL integer columns incorrectly. Thanks to Piotr Jurkiewicz for the patch. (Bug #20535517, Bug #75864)
- **InnoDB:** The following changes were implemented for full-text index auxiliary tables:
  - If the primary table is assigned to a general tablespace, full-text auxiliary tables are created in the same general tablespace.
  - Full-text auxiliary tables are created with the same row format as the primary table.

  - If the primary table was created in a location outside of the data directory using the `DATA DIRECTORY` clause, full-text auxiliary tables are created in the same location as the primary table.

  (Bug #20527217, Bug #75869)

- **InnoDB:** The `memcached set` command permitted a negative expire time value. Expire time is stored internally as an unsigned integer. A negative value would be converted to a large number and accepted. The maximum expire time value is now restricted to `INT_MAX32` to prevent negative expire time values. (Bug #20478242, Bug #75790)

- **InnoDB:** An interrupted `ALTER TABLE` operation that rendered a child table unavailable caused an error and debug assertion after crash recovery, when the `ALTER TABLE` operation on the parent table detected that the foreign keys of the parent table could not be loaded. The debug assertion was removed and the error was replaced by a warning. (Bug #20476395)

  References: This issue is a regression of: Bug #19267051.

- **InnoDB:** In debug builds, enabling the `btr_cur_limit_optimistic_insert_debug` flag raised a deadlock exception in the change buffer clustered index. (Bug #20459905, Bug #75736)

- **InnoDB:** A warning message is now printed if `DB_TRX_ID` stored in a record is found to be greater than `max_trx_id`. In debug builds, an assertion is raised. (Bug #20445525)

- **InnoDB:** Estimates for the number of records in a range for a given dataset could differ depending on the page size. (Bug #20427694)

- **InnoDB:** During shutdown on Windows, the listener object in `handle_shutdown()` was freed while the listener was running, resulting in an exception in `buf_pool_from_bpage()`. (Bug #20421223)

- **InnoDB:** `SHOW ENGINE INNODB STATUS` output showed negative reservation and signal count values due to a counter overflow error. (Bug #20417397)

- **InnoDB:** Failure to check the status of a cursor transaction read-only option before reusing the cursor transaction for a write operation resulted in a server exit during a `memcached` workload. (Bug #20391552)

- **InnoDB:** An assertion was raised in a debug build when populating a spatial index during an `ALTER TABLE` operation. The size of data tuples for compressed rows is calculated in the `rec_get_converted_size_comp_prefix_low` function. Debug code within the function did not account for the spatial index or the `DATA_SYS_CHILD` data type in the node pointers. (Bug #20372749)

- **InnoDB:** `CHECK TABLE` returned a “wrong count” error for tables with spatial indexes. (Bug #20313067)

- **InnoDB:** MDL locks taken by `memcached` clients caused a MySQL Enterprise Backup `FLUSH TABLES WITH READ LOCK` operation to hang. (Bug #20275612)

- **InnoDB:** An embedded MySQL server failed to start with `innodb_undo_tablespaces=2`. The server was unable to locate undo tablespaces that were created when the MySQL instance was initialized. For embedded MySQL installations, the `innodb_undo_directory` default value of “`.`” may not be the same directory as the MySQL data directory. To address this problem, `innodb_undo_directory` is now NULL by default, requiring that a path be specified. If a path is not specified, undo tablespaces are created in the MySQL data directory, as defined by `datadir`. A workaround for pre-MySQL 5.7.8 embedded installations is to define an absolute path for `innodb_undo_directory`. (Bug #20023425)

- **InnoDB:** A DML operation raised an assertion in file `lock0lock.cc`. A session holding an exclusive row lock on a clustered index page initiated a page reorganization while another session waited for

a lock on the same row. The page reorganization changed the lock order, causing an assertion in `lock_rec_add_to_queue()`. (Bug #20005279)

- **InnoDB:** A `DROP DATABASE` operation raised an assertion. (Bug #19929435)
- **InnoDB:** `InnoDB` failed to open a tablespace after the data directory location of the tablespace was changed from a relative path to a full path. `InnoDB` failed to recognize that the relative path, which remained embedded in the data dictionary, pointed to the same data file as the full path. (Bug #19896685)
- **InnoDB:** A `TRUNCATE TABLE` operation appeared to hang when run in parallel with a read-write workload. (Bug #19873470, Bug #74312)
- **InnoDB:** The sorted index build feature introduced in MySQL 5.7.5 caused a performance regression when adding an index to a small table. The regression was due to excessive flushing triggered by a forced checkpoint that occurs after the sorted index build. (Bug #19865673, Bug #74472)
- **InnoDB:** Updates to indexed columns could be slower in MySQL 5.7.5 and higher. In pre-MySQL 5.7.5 releases, `InnoDB` reserves 1/16 of the space in clustered index pages for future inserts and updates. This behaviour changed in MySQL 5.7.5 with the introduction of the `innodb_fill_factor` option. With `innodb_fill_factor=100`, B-tree index pages were completely filled during sorted index builds, and subsequent updates to index pages resulted in page splitting. To restore pre-MySQL 5.7.5 behavior, the default setting of `innodb_fill_factor=100` now leaves 1/16 of the space in clustered index pages free for future index growth. (Bug #19821087, Bug #74325)
- **InnoDB:** The `ha_innobase::index_flags` function returned invalid flags for spatial indexes. (Bug #19473391)
- **InnoDB:** On Windows, asynchronous I/O requests remained waiting after `InnoDB` initialization was aborted. (Bug #19363615)
- **InnoDB:** An index record was not found on rollback due to inconsistencies in the `purge_node_t` structure. The inconsistency resulted in warnings and error messages such as “error in sec index entry update”, “unable to purge a record”, and “tried to purge sec index entry not marked for deletion”. (Bug #19138298, Bug #70214, Bug #21126772, Bug #21065746)
- **InnoDB:** The `ut_when_dtor` struct, added in MySQL 5.7 to address a Valgrind issue, was removed to reduce code complexity. (Bug #18309926)
- **InnoDB:** An `INSERT` operation raised an assertion when the transaction mode was modified after the transaction started. (Bug #15866285)
- **InnoDB:** Queries that use both `UNION` and `UNION ALL` and disable the index would cause an assertion due to duplicate B-tree values. (Bug #76439, Bug #20752543)
- **InnoDB:** In debug builds, attempting to create a spatial index after dropping the `mysql.innodb_table_stats` table raised an assertion in the `btr_cur_open_at_rnd_pos_func` function. (Bug #76437, Bug #20753642)
- **InnoDB:** Transaction objects were passed to optimized temporary table APIs, causing an assertion. Optimized temporary tables, which do not support rollback and are not shared across connections, should ignore the transaction objects. (Bug #76415, Bug #20748479)
- **InnoDB:** When `innodb_thread_concurrency=1`, queries on optimized temporary tables caused other sessions to hang. Queries on optimized temporary tables should not increment the number of active threads. (Bug #76346, Bug #20762059)
- **Partitioning:** During execution of correlated subqueries, the server reinitialized a scan executed on the same table without ending the previous scan. (Bug #20949314, Bug #76810)

- **Partitioning:** `CREATE TABLE` statements that used an invalid function in a subpartitioning expression did not always fail gracefully as expected. (Bug #20310212)
- **Partitioning:** For an ordered index scan over multiple partitions, MySQL performs a merge sort across them using a priority queue whose entries hold pointers to buffers containing fetched rows. When all rows from all partitions are fetched, this queue is now empty. When this occurred, subsequent attempts to fetch rows were done by passing a null buffer pointer, which caused the server to fail. This could manifest itself when executing `HANDLER ... READ ... PREV` against a partitioned table. Now in such cases this pointer holds `NO_CURRENT_PART_ID` so that the partitioning handler is aware that the queue is empty. (Bug #20270687)
- **Partitioning:** `REPAIR TABLE ... QUICK` could fail when used with multiple partitioned tables. (Bug #76154, Bug #20647894)
- **Partitioning:** In certain cases, `ALTER TABLE ... REBUILD PARTITION` was not handled correctly when executed on a locked table. (Bug #75677, Bug #20437706)
- **Replication:** When using multiple replication channels, issuing `RESET SLAVE` on a non-default replication channel removes the channel, whereas issuing `RESET SLAVE` on the default replication channel does not remove the channel, as it always exists. In previous versions, this meant that the default replication channel did not correctly reset some configuration and status parameters. The fix ensures that issuing `RESET SLAVE` on the default replication channel resets all parameters. (Bug #21107331, Bug #21111229, Bug #77086)
- **Replication:** Repeatedly checking for **`ERR_LOCK_WAIT_TIMEOUT`** (as done, for example by repeatedly executing `SHOW SLAVE STATUS`) during a prolonged write lock on a table led to an assert. (Bug #21095969)
- **Replication:** `SHOW BINLOG EVENTS` was not showing the correct statement for `XA COMMIT ... ONE PHASE`. Although the event was logged and replicated correctly, `SHOW BINLOG EVENTS` was showing an incorrect statement when handling the event. The fix ensures that the statement is correctly displayed. (Bug #21053526)
- **Replication:** When changing `gtid_mode` online, if `autocommit` was set to 0 and a `set gtid_next=UUID:NUMBER` statement had been issued, then changing `gtid_mode` was not being blocked. The fix ensures that variables which can only be set outside transaction context can now only be set if the thread does not own a GTID and does not hold anonymous ownership. This changes the behavior of these variables:

  - `enforce_gtid_consistency`
  - `gtid_mode`
  - `gtid_purged`
  - `session_track_gtids`

  (Bug #20865683)
- **Replication:** `mysqlbinlog` would apply any rewrite rules before applying the database filter. This meant that in cases when statement-based replication transactions were mixed with row-based replication transactions only one or the other type of transaction would be output. The fix changes the behavior so that the rewrite rules also apply to the `USE` *`db_name`* clause, rewriting the database specified by *`db_name`* according to the setting of the `--rewrite-db` parameter. This makes it possible to use the `--database` option on the query and row events. In addition, it removes the suppression of the `USE` *`db_name`* statement and ensures that the rewrite is done before the database filter. (Bug #20810442)

- **Replication:** Row unpacking did not function correctly in some cases when running the server with `binlog_row_image` set to `minimal`. (Bug #20468712)
- **Replication:** When slaves, and especially semisynchronous replication slaves, connected to a master there was a chance they could encounter a `SLAVE HAS MORE GTIDS THAN THE MASTER HAS` error. During connection the slave sends all replicated GTIDs to the master, and the master checks if all the GTIDs matching its `server_uuid` are included in its `gtid_executed` GTID set. There was a chance that a GTID was already in the slave's `gtid_executed` GTID set, but not in the master's `gtid_executed` GTID set. This was due to the GTID being added into `gtid_executed` after it was added to the binary log, meaning it was possible that a transaction had been replicated and applied on the slave, but not committed on the master yet. The fix ensures that the master checks if all GTIDs are in the union of `gtid_executed` and `gtid_owned`. (Bug #20464737)
- **Replication:** A replication slave running with a `gtid_mode` other than `OFF`, `log_bin=OFF` and `relay_log_info_repository=TABLE` was consuming the GTID of a transaction prematurely when applying a transaction that spanned across distinct relay log files. This was caused when the slave SQL thread was flushing the relay log information to the `mysql.slave_relay_log_info` table while in the middle of a transaction, or immediately after the GTID of a transaction, because of the rotation of the relay log. The fix skips saving the `gtid_state` for operations that save the relay log information to the `mysql.slave_relay_log_info` table. (Bug #20451386)
- **Replication:** When `gtid_mode=ON`, issuing a `SET gtid_next='UUID:NUMBER'` statement and then issuing a statement that caused an error, such as `CREATE..SELECT`, led to an assertion failure on `COMMIT`. This was due to `gtid_next` being incorrectly set to undefined for implicitly committing statements, even if the statement failed with an error before the implicit commit happened. (Bug #20343644, Bug #20444828)
- **Replication:** If a slave was restarted with `--relay-log-recovery` enabled and the initialization of the default replication channel had failed, a slave could refuse to start or cause an assert in debug mode. (Bug #20191813, Bug #20236305)
- **Replication:** When using `RESET MASTER`, the GTID state (`gtid_executed` and `gtid_purged`) is reset. On a server with `log_bin=OFF`, using `RESET MASTER` fails because the binary log is not enabled. However, since MySQL 5.7.5, GTIDs can be enabled even when the binary log is disabled. So in this case there was no way to reset the GTID state. The fix ensures that `RESET MASTER` can be executed on a server with GTIDs enabled and `log_bin=OFF`, enabling you to reset the GTID state. (Bug #19706455)
- **Replication:** If statement based logging was in use, when updating multiple tables in a single statement, a single transaction could be logged as two different transactions. This was due to the binary logging process not properly identifying statements which were operating over transactional tables. The fix ensures that they are correctly identified, even if such statements do not change the contents of the tables. (Bug #16621582, Bug #21349028)
- **Replication:** When starting more than one instance of `mysqld` on the same host at the same time, the instances could get the same `server_uuid`. The fix uses settings unique to each started `mysqld` instance to ensure that each gets a unique UUID. (Bug #16459136)
- **Replication:** Some replication thread statuses were not being shown in the `PROCESSLIST_INFO` column of the `threads` table. (Bug #77115, Bug #21127308)
- **Replication:** In MySQL 5.7.7 and earlier, GTIDs were automatically disabled whenever `--initialize` or `--bootstrap` were enabled. In MySQL 5.7.8 and later GTIDs are not disabled when `--initialize` or `--bootstrap` are enabled. (Bug #76884, Bug #20980271)
- **Replication:** When using `mysql-test-run.pl` with the `--ps-protocol` option to run a test against a server with `log_bin=OFF`, setting `gtid_next` and then executing a DDL statement caused an error.

This was due to the DDL statement not being correctly logged to consume the GTID specified when setting `gtid_next`. (Bug #76820, Bug #20954452)

- **Replication:** When `log_bin=ON`, if a read-only XA transaction was prepared but had an empty body, a subsequent `XA ROLLBACK` caused an assertion. (Bug #76734, Bug #20902763)

- **Replication:** `explicit_defaults_for_timestamp` has been changed from a global variable to a global and session variable, and the session variable is now replicated. This means that you can change the variable and still be guaranteed that every statement uses the same value for the variable on master and slave, even if the variable is not changed synchronously on master and slave. (Bug #76657, Bug #20866059)

- **Replication:** When a slave was stopped, `replication_applier_status_by_worker` showed `worker_id` as 1 and did not show the correct number. (Bug #76637, Bug #20857660)

- **Replication:** Setting `SESSION.GTID_NEXT=default` immediately after setting `SESSION.GTID_NEXT='ANONYMOUS'` causes an `ER_CANT_SET_GTID_NEXT_WHEN_OWNING_GTID` error, but this also changes the `gtid_next` type from `ANONYMOUS` to `AUTOMATIC`, meaning that the next transaction could potentially cause an `GTID_NEXT->TYPE != AUTOMATIC_GROUP || THD->OWNED_GTID.IS_EMPTY()` assertion. To avoid this possibility, the fix ensures that setting `SESSION.GTID_NEXT=default` does not change the `gtid_next` type if it could cause an error. (Bug #76434, Bug #20753378)

- **Replication:** Additional replication channels could not be added if the server had been started with `server_id=0`, the default. The fix ensures that a replication slave checks the `server_id` is greater than 0 when `CHANGE MASTER TO` has been issued before continuing with the initialization of a channel. (Bug #76432, Bug #20753463, Bug #20712720)

- **Replication:** When a server was configured with `gtid_mode=ON`, `log-bin=OFF`, and with `autocommit` enabled, during commit it was possible to encounter an `ASSERTION `IS_STARTED()' FAILED` error. This was possible when an applier thread committed a transaction's GTID into the `gtid_executed` table before transaction prepare when binary logging was disabled, or binary logging was enabled and `log_slave_updates` was disabled. The cause was that when the server was saving a transaction's GTID into the `gtid_executed` table, the calculated transaction context could be committed during the save phase when `autocommit` was enabled. The fix ensures that the transaction context is calculated after saving the GTID's state, and then commit is executed on any remaining transactions. (Bug #76425, Bug #20748570)

- **Replication:** When using row-based logging with `autocommit` disabled and `GTID_MODE=OFF_PERMISSIVE`, if a transaction started with a `CREATE TEMPORARY TABLE` statement, then regardless of whether the table was transactional or non-transactional, the transaction began an automatic GTID violating transaction. However, if `GTID_NEXT='UUID:NUMBER'` was issued immediately after executing the `CREATE TEMPORARY TABLE` statement, which sets `gtid_next` type to `GTID_GROUP`, upon committing the transaction, a check for possible violation of GTID consistency was causing an assertion failure because the gtid_next type had been changed from `AUTOMATIC_GROUP` to `GTID_GROUP`.

  The fix ensures that transactions with an empty owned GTID correctly check if they break GTID consistency. As part of this fix, it was found that when `autocommit` was disabled, the statement `CREATE TEMPORARY TABLE` did not start a transaction, so immediately setting `GTID_NEXT='UUID:NUMBER'` could not cause an error. The fix ensures that when `autocommit` is disabled, executing `CREATE TEMPORARY TABLE` or `DROP TEMPORARY TABLE` starts a transaction, regardless of the state of `log_bin`, `binlog_format` and whether a transactional or non-transactional storage engine is in use. This makes the behavior consistent, but is a change in logging when `log-bin=OFF`. (Bug #76416, Bug #20748502)

- **Replication:** When using GTIDs with `log-bin=OFF`, a combination of statements which included a `BINLOG` statement executing a `Format_description_log_event` would cause an assertion. The reason was that the server would run the routine to handle the end of a GTID violating transaction, which was incorrect because when `--log-bin=OFF` the transaction should not have been handled as an offending transaction. The fix ensures that before the compatibility testing, the state of `log_bin` is checked. If `log_bin=OFF`, then the compatibility checks are skipped. This makes the server not activate flags that would then trigger the incorrect run of the routine to handle GTID violating transactions that ultimately resulted in an assertion. (Bug #76406, Bug #20743468)

- **Replication:** When binary logging was enabled, using stored functions and triggers resulting in a long running procedure that inserted many records caused the memory use to increase rapidly. This was due to memory being allocated per variable. The fix ensures that in such a situation, memory is allocated once and the same memory is reused. (Bug #75879, Bug #20531812)

- **Replication:** If an error occurred when using a multi-threaded slave, issuing a `CHANGE MASTER TO` statement which resulted in an `ER_MTS_CHANGE_MASTER_CANT_RUN_WITH_GAPS` error, and then issuing `RESET SLAVE`, made it impossible to change master due to repeated `ER_MTS_CHANGE_MASTER_CANT_RUN_WITH_GAPS` errors. Running the debug version of `mysqld` caused an unexpected exit in this case. The fix ensures that the recovery process for multi-threaded slaves avoids this. (Bug #75574, Bug #20411374)

- **Replication:** When using semisynchronous replication performance was degrading when the number of threads increased beyond a certain threshold. To improve performance, now only the thread which is committing is responsible for deleting the active transaction node. All other operations do not touch this active transaction list. (Bug #75570, Bug #20574628)

- **Replication:** When `gtid_executed_compression_period` is set to a number greater than 0, there is a thread that wakes up after every number of transactions specified by `gtid_executed_compression_period` to perform range compression on the `mysql.gtid_executed` table. There was a small chance that the thread would miss a signal and not wake up, so that one pass of the compression algorithm would be missed and the table left uncompressed. The fix ensures that the thread wakes up consistently. (Bug #75014, Bug #20104307)

- **Replication:** Using `mysqlbinlog` to process log events greater than 1.6GB failed with an out of memory error. This was caused by an internal error converting the length variable. The fix upgrades the length variable to avoid overflow in both encoding and decoding functions. (Bug #74734, Bug #20350989)

- **Replication:** Setting `gtid_next` inside a stored procedure and committing an empty transaction caused an error. This was due to the empty transaction not being correctly logged to consume the GTID specified by setting `gtid_next`. (Bug #74253, Bug #19774317)

- **Replication:** Some messages that were meant to be printed when `log_warnings` was greater than 1 started appearing in the error log. The information regarding the multi-threaded applier in particular was too verbose and much of this information was very technical and development oriented. The fix ensures that these messages are not logged. (Bug #74203, Bug #19729278)

- **Replication:** When using a multi-threaded slave with `slave_preserve_commit_order=1`, certain combinations of transactions being applied in parallel could cause a deadlock and stop the slave responding. The fix introduces a check for such deadlocks, so that when a transaction needs to wait for another transaction to release a row lock, `InnoDB` checks if there is a deadlock caused by the commit order. If it finds a deadlock caused by the commit order, it sets a deadlock flag for the slave worker which is holding the row lock. Then the worker rolls back its transaction and tries again. (Bug #74177, Bug #20136704)

- **Replication:** When `relay_log_recovery` is set, the error log entry that reports the new recovery positions has been extended to also report the old relay log positions. (Bug #74089, Bug #21305976)

- **Replication:** When a master with `--binlog_checksum=none` and `--gtid-mode=ON` was replicating to a slave with `--binlog_checksum=crc32`, restarting the slave's SQL thread caused an `Event crc check` error. This was due to the `Format_description_log_event` from the master not being correctly found in existing relay logs after restarting the slave's SQL thread. The fix ensures that the `Previous_gtids_log_event` is correctly skipped and that the correct `Format_description_log_event` is found in existing relay logs after restarting the slave's SQL thread. (Bug #73806, Bug #20644100, Bug #76746, Bug #20909880)

- **Replication:** When `gtid_mode=on`, GTIDs are automatically added to the `mysql.gtid_executed` table. If a GTID was manually inserted into the `mysql.gtid_executed` table and then automatic update inserted the same GTID, the server crashed. Manually inserting GTIDs into `mysql.gtid_executed` is an unsupported operation, but this fix ensures that the server does not crash in such a situation. (Bug #73601, Bug #19451053)

- **Replication:** When using GTIDs, a multi-threaded slave which had `relay_log_recovery=1` and that stopped unexpectedly could encounter a `relay-log-recovery cannot be executed when the slave was stopped with an error or killed in MTS mode` error upon restart. The fix ensures that the relay log recovery process checks if GTIDs are in use or not. If GTIDs are in use, the multi-threaded slave recovery process uses the GTID protocol to fill any unprocessed transactions. (Bug #73397, Bug #19316063)

- **Replication:** When `master_info_repository=TABLE` the receiver thread stores received event information in a table. The memory used in the process of updating the table was not being freed correctly and this could lead to an out of memory error. The fix ensures that after an event is flushed to the relay log file by a receiver thread, the memory used is freed. (Bug #72885, Bug #19390463, Bug #69848, Bug #20124342)

- **Replication:** The status variables `Rpl_semi_sync_master_net_wait_time` and `Rpl_semi_sync_master_net_avg_wait_time` were always `0` in MySQL 5.7.4 and later. These variables are now deprecated and will be removed in a future version. (Bug #72627, Bug #18750614)

- **Replication:** When two slaves with the same `server_uuid` were configured to replicate from a single master, the I/O thread of the slaves kept reconnecting and generating new relay log files without new content. In such a situation, the master now generates an error which is sent to the slave. By receiving this error from the master, the slave I/O thread does not try to reconnect, avoiding this problem. (Bug #72581, Bug #18731252)

- **Replication:** If a slave encountered a `ER_NET_READ_INTERRUPTED` or `ER_NET_WRITE_INTERRUPTED` error while getting a timestamp or server ID from the master, setting `MASTER_HEARTBEAT_PERIOD` and so on, the slave's receiver thread stopped. The fix ensures that these errors are treated as transient network errors, and the slave receiver thread attempts to automatically reconnect to the master in such a situation. (Bug #71374, Bug #18091217)

- **Replication:** Using `mysqlbinlog` to replay a relay log which ended with `GTID_LOG_EVENT` could cause the following error:

  ```
  ERROR 1790 (HY000) @@SESSION.GTID_NEXT cannot be changed by a client that
  owns a GTID. The client owns UUID:GTID. Ownership is released on COMMIT or
  ROLLBACK.
  ```

  If a relay log rotate happens (either through a receiver thread restart or after issuing the `ROTATE` command) exactly after writing a `GTID_LOG_EVENT`, when replaying such a relay log's end `ROTATE_EVENT`, it was mistakenly identified as being inside a transaction, whereas the transaction was actually started after `GTID_LOG_EVENT`. This caused `mysqlbinlog` to append `SET @@SESSION.GTID_NEXT='AUTOMATIC'`, resulting in two `GTID_NEXT` statements one after the other.

The fix ensures that `mysqlbinlog` generates `SET @@SESSION.GTID_NEXT='AUTOMATIC'` only outside of a transaction and when there has not been a previous `GTID_LOG_EVENT`.

Similarly, using `mysqlbinlog` to concatenate and replay a relay log which contained a partial GTID transaction caused the above error. A relay log can contain a partial GTID transaction when `AUTO_POSITION` is enabled if a receiver thread is restarted when it is in the middle of transferring a transaction from a master. On restart the slave retrieves the full transaction again. In this case, the first relay log contains a partial GTID transaction and the second relay log contains the full GTID transaction again. When using `mysqlbinlog` to concatenate such a relay log, the partial transaction was not being correctly detected and therefore a `ROLLBACK` was not being correctly generated. The fix identifies partial GTID transactions using the format description event of the second relay log, ensuring that a `ROLLBACK` is correctly added. (Bug #70711, Bug #17650326)

- **Replication:** The replication connection now sends the `program_name` attribute, “mysqld”, in line with the behavior of other client connections. In addition, a `_client_role` attribute has been added and is set to “binary_log_listener”, to clarify the replication connection's role, as well as the `_client_replication_channel_name` attribute, which is set to the replication channel's name. Similarly, `mysqlbinlog` now sets `_client_role` to “binary_log_listener”. These changes are exposed through the `session_connect_attrs` Performance Schema table. (Bug #68782, Bug #16555723)

- On platforms where `char` is unsigned, `mysql_config_editor` could fail to detect failed operations. Affected platforms include ARM and PowerPC. (Bug #21355630)

- The `Rewriter` plugin linked against the `mysys` library, which is already linked into the server and thus available at load time when the plugin is installed. (Bug #21255496)

- Memory leaks found by enabling AddressSanitizer were corrected in `mysql`, `mysqlcheck`, `mysqldump`, `mysqlshow`, `mysqlslap`, `mysqltest`, `mysql_client_test`, `mysql_upgrade`, and `mysql_install_db`. (Bug #21246627, Bug #21246842, Bug #21246964, Bug #21247377, Bug #21250562, Bug #21250584, Bug #21250644, Bug #21250876, Bug #21250947, Bug #21253535, Bug #21253653, Bug #21254060, Bug #21255860)

- Multiple definitions of `key_memory_KEY_CACHE` caused compilation failure when ASAN was enabled. (Bug #21245718)

- For debug builds, failure of the range optimizer to properly propagate errors occurring during partition pruning could raise an assertion. (Bug #21211492)

- A `SET PASSWORD` statement that failed with `ER_MUST_CHANGE_PASSWORD` could still change the `password_last_changed` column for a row in the `mysql.user` table. (Bug #21192879)

- An optimizer hint assertion could be raised when a table was used in both parts of an `INSERT INTO ... SELECT` statement. (Bug #21192857)

- The `create_tmp_table()` return value was not checked, which could lead to a server exit. (Bug #21190532)

- Incorrect cost calculation for the semi-join Duplicate Weedout strategy could result in a server exit. (Bug #21184091)

- Some Valgrind warnings in `Item_type_holder::join_types()` were spurious and have been silenced. (Bug #21156155)

  References: This issue is a regression of: Bug #19471564.

- The optimizer hint parser could read freed memory. (Bug #21148405)

- For debug builds, `VIRTUAL` generated columns could be marked writable during read operations and cause an assertion to be raised for partitioned tables. (Bug #21142905)
- For debug builds, a missing error test for full-text searches could cause an assertion to be raised. (Bug #21140111)
- Outer references do not work as arguments to `MATCH()`, but the server did not properly detect them. Now it does and raises an error. (Bug #21140088)

  References: See also: Bug #20007383.
- `EXPLAIN` could raise an assertion trying to display very large full-text search rank values. (Bug #21140067)
- For debug builds, full-text searches could raise an assertion if the optimizer tried to use a covering index when that was not appropriate. (Bug #21140039)
- `SHOW STATUS` and `SHOW VARIABLES` failed to produce output if the server was started with the Performance Schema disabled. (Bug #21139458)
- `ST_Intersection()` could produce areal and point intersection results, but was not able to produce linear intersection results. (Bug #21109896)
- `ALTER TABLE` statements that defined a generated column using `MATCH ... AGAINST` in its expression could raise an assertion. (Bug #21098119)
- The server could raise an assertion or produce an incorrect error message for inserts into a view if a single table for insertion could not be identified. (Bug #21097485)
- `open_files_limit` could be set higher than permitted by the operating system. (Bug #21074643)
- `systemd` timeout logic could be triggered if `InnoDB` log rebuilding or recovery took too long. Because the time this may take is unknown, `systemd` timeout during service start or stop is now disabled. (Bug #21071740)
- Comparisons of table names in optimizer hints did not respect the value of the `lower_case_table_names` system variable. (Bug #21056644)
- `CMake` configuration was adjusted to disable unnecessary warnings reported by Clang and display them only if `-DMYSQL_MAINTAINER_MODE=1` is used. (Bug #21041451)
- Multiple executions of a prepared `SET` statement that used a subquery could result in a server exit. (Bug #20982756)
- With `auto_generate_certs` enabled, the server automatically created SSL files if any of `ca.pem`, `server-cert.pem`, and `server-key.pem` were missing from the data directory. Now it creates the files only if all of them are missing (the same test used by `mysql_ssl_rsa_setup`). (Bug #20963082)
- The server compiled with Performance Schema support could not be started with `performance_schema=OFF` due to a dependency on it for the `sys` schema, which expected to find Performance Schema tables. The Performance Schema now creates its tables during startup even if disabled. (Bug #20956599)
- For `CREATE TABLE ... SELECT`, it was possible to assign values to generated columns in the destination table. (Bug #20949226)
- The server could exit when the Performance Schema read thread status variables under load. (Bug #20927157, Bug #20922218, Bug #21103103)

- CMake support was adjusted for the change of the -Wno-unused-local-typedefs option to -Wno-unused-local-typedef in Clang 3.6. (Bug #20921370)
- For debug builds, XA PREPARE raised an assertion if a transaction contained at least one update and none were InnoDB updates. (Bug #20920851)
- Using ST_Centroid() with a geometry collection containing an invalid polygon could cause a server exit. (Bug #20918881)
- For small values of the read_rnd_buffer_size system variable, internal caching of temporary results could fail and cause query execution failure. (Bug #20895852)
- Invalid memory pointer access could occur during access to the events_statements_history Performance Schema table, resulting in a server exit. (Bug #20878306)
- For debug builds, passing EXPORT_SET() to VALIDATE_PASSWORD_STRENGTH() could raise an assertion. (Bug #20863229)
- A failed FLUSH PRIVILEGES statement followed by statements to create or drop accounts could cause a server exit. (Bug #20857652)
- Large values of the *points_per_circle* argument to the ST_Buffer_Strategy() function could cause large amounts of memory to be used. To avoid inadvertent excessive memory use, the maximum value of this argument is now constrained to be the value of the new max_points_in_geometry system variable. This variable has default, minimum, and maximum values of 65,536, 3, and 1,048,576, respectively. (Bug #20842030, Bug #21212788)
- For certain inputs, ST_Buffer() could raise an assertion. (Bug #20841874)
- An assertion could be raised if the server used a string column as the key of a temporary table. (Bug #20835095)

  References: This issue is a regression of: Bug #19695490.
- SHOW VARIABLES mutexes were being locked twice, resulting in a server exit. (Bug #20788853)
- ull2dec() was modified to avoid a problem with GCC 5 in optimized mode. (Bug #20768820)
- Using GCC 5, debug builds failed due to compiler warnings. (Bug #20768717)
- DDL operations on a server configured with InnoDB as read only caused a server exit due to invalid memory access during error reporting. (Bug #20763179)
- ALTER TABLE could fail to prevent subqueries in the definition of generated columns, resulting in a server exit. (Bug #20757211)
- Invalid use of the THD structure with generated columns could cause an assertion to be raised. (Bug #20746926)
- Parser state was initialized incorrectly for parsing generated column expressions. (Bug #20745142)
- For large values of max_digest_length, the Performance Schema could encounter an overflow error when computing memory requirements, resulting in a server exit. (Bug #20738072)
- Columns specified through JOIN ... USING or NATURAL JOIN that were resolved from a derived table could raise an assertion. (Bug #20733540)
- MySQL 5.7.6 restricted the list of symbols exported by the C client library. One of these was mysql_get_parameters, but that is used by the DBD::mysql Perl module. mysql_get_parameters is now exported. (Bug #20686665)

References: See also: Bug #18427840, Bug #20476596, Bug #20821550.

- Cleanup after a `MATCH()` operation could write to freed memory. (Bug #20685427)
- `NDB` could raise an assertion for failure to get the tablespace name when attempting to acquire a metadata lock. (Bug #20676000)
- `mysqlslap` and `mysql_client_test` failed to use an SSL connection by default. (Bug #20654023)
- The Spencer `regex` library used for the `REGEXP` operator could be subject to heap overflow in some circumstances. (Bug #20642505)
- A missing error check after a call to `find_field_in_tables()` within the optimizer could cause an assertion to be raised. (Bug #20615597)
- Optimization of `x IN (SELECT y FROM DUAL WHERE ...)` was treated the same as `x IN (SELECT y FROM DUAL)`, losing the `WHERE` clause and resulting in a server exit. (Bug #20615023)
- A buffer-overflow error could occur for `mysqlslap` during option parsing. (Bug #20605441)
- For debug builds, `DROP DATABASE` raised an assertion if there were non-database files in the database directory. (Bug #20573701)
- For `CREATE TABLE ... SELECT`, an error occurred if a selected column was a generated column that depended on a nonselected column. To handle this, the destination table does not preserve information about whether selected columns are generated columns. (Bug #20566243)
- A user with an expired password could execute `ALTER USER` statements other than to assign a new password. (Bug #20553132)
- An OpenSSL error queue associated with each thread was not freed on thread release, resulting in a Valgrind error. (Bug #20551271)
- The property of whether a view is updatable was calculated when it was created. If the view referred to another view that was dropped and recreated and the new definition of the referenced view had different updatability than the original definition, that could affect the updatability of the referring view. Not taking into account this change in updatability could cause an assertion to be raised. To avoid this problem, the server now assesses updatability when reading a view definition rather than at view creation time. (Bug #20515155)
- Built-in SQL functions could raise an assertion or cause a server exit if the wrong thread pointer was used to produce an error or warning message. (Bug #20454979)
- Incorrect calculation of the length of strings written to the binary log could raise an assertion or cause a server exit. (Bug #20444737)

  References: This issue is a regression of: Bug #16066637.
- The range optimizer interpreted a hidden key part (`InnoDB` primary key) as a minimum bounding rectangle (MBR) index. Such primary keys cannot be used as MBRs, and a server exit resulted.. (Bug #20430526)
- The `WITH CHECK OPTION` of a view was sometimes ignored if the view was included in another view. For discussion of the implications of this fix, see The View WITH CHECK OPTION Clause. (Bug #20407961)
- Calculation of “within” or “contains” relationships failed for some types of geometry collections. (Bug #20379981)

- Long path name values for some options could lead to stack overflow. (Bug #20376760)
- Setting the password for an account not using a built-in authentication plugin could cause the account to become unusable.

  The fix for this problem involves a change to the authentication plugin API to add a new authentication_flags member to the server-side plugin descriptor. See Writing the Server-Side Authentication Plugin (Bug #20364862)
- Spatial WKT export functions produced too-long string representations of coordinate values instead of switching to exponential notation. (Bug #20363531)
- An off-by-one error in string-copying code could result in a buffer overflow. (Bug #20359808)
- The `events_waits_summary_by_instance` Performance Schema table could fail to return rows for socket instruments. (Bug #20348824)
- Under certain conditions, the `libedit` command-line library could write outside an array boundary and cause a client program crash. (Bug #20318154)
- Invalid linestring values with a single point and unclosed polygons with fewer than four points could cause a server exit. Such invalid values now are rejected. (Bug #20316779)
- `mysql_config_editor` could exit abnormally while encrypting passwords. (Bug #20294225)
- A deadlock error reported by InnoDB could cause rollback inside InnoDB while the transaction continued at the SQL layer. (Bug #20262654)
- MySQL sometimes produced no warning when it was unable to interpret a character in a given character set. (Bug #20238729)
- Host value matching for the grant tables could fail to use the most specific of values that contained wildcard characters. (Bug #20181776)
- For MySQL distributions linked against yaSSL, a corrupt client key file could cause clients to exit. (Bug #20168526)
- Use of `SELECT COUNT(DISTINCT)` in a subquery in the `FROM` clause could produce incorrect results. (Bug #20145024)

  References: This issue is a regression of: Bug #18766378.
- For join queries with a large number of tables, the server could exit converting the join to a semi-join. (Bug #20109861)
- `ALTER TABLE` operations that changed only an index comment were not being treated as a fast/in-place alteration. (Bug #20106553)
- Following execution of a `GRANT ... WITH GRANT OPTION` statement, execution of a prepared statement with a view could cause a server exit. (Bug #20030284)
- `ADDTIME()` could produce an out-of-range result with a year >= 10,000. (Bug #19900900)
- Within a stored procedure, access to view columns after DDL or `FLUSH TABLES` statements in the procedure could cause a server exit. (Bug #19897405)
- References to select list columns of the outer query from the `HAVING` clause of a correlated subquery in the inner query should, but did not, return an error, resulting in a server exit. (Bug #19823076)
- Several `ST_Envelope()` problems were corrected:

  - If the mimimum bounding rectangle (MBR) of a geometry degrades to a `Point` or horizontal or vertical `LineString`, `ST_Envelope()` returns that value rather than an invalid polygon.
  - The return value for an empty geometry collection now is an empty geometry rather than `NULL`.
  - If a geometry is geometrically invalid but has a valid WKB string, return a valid MBR rather than `NULL`.

  (Bug #19811953, Bug #20196720)

- For debug builds, an assertion could be raised when a top-level query had a `HAVING` clause that contained a subquery referencing a column from the top-level query. (Bug #19811896)
- `GROUP BY` or `ORDER BY` on a `CHAR(0) NOT NULL` column could lead to a server exit. (Bug #19660891)
- The server could exit if a grouped query had a nongrouped subquery that contained a reference to an aggregate function. (Bug #19585938)
- Loading corrupt spatial data into a `MyISAM` table could cause the server to exit during index building. (Bug #19573096)
- Some spatial functions converted -0 to 0. This no longer occurs. (Bug #19504183)
- For debug builds, certain `UPDATE` statements could raise an assertion. (Bug #19055268)
- The LooseScan execution strategy for semi-joins failed to evaluate the `WHERE` condition on rows coming from the first inner table of an outer join. (Bug #18892055)
- An internal procedure that creates temporary tables and expected a flat list of expressions to map onto table columns sometimes received a list that was not flat, causing an assertion to be raised. (Bug #18745214)
- For `MyISAM` or `MEMORY` tables, a nested join with a subquery could product a result set with missing rows when the `optimizer_switch` `condition_fanout_filter` flag was enabled. (Bug #18717059)
- For some status variables that should monotonically increase, `SHOW GLOBAL STATUS` in one session could show them as decreasing when other concurrent sessions changed user or disconnected. (Bug #18591145)
- On Windows, setting `query_cache_min_res_unit` to too large a value could result in a value of 0 and a subsequent server exit. (Bug #18487951)
- For debug builds, `SET` statements that assigned a subquery value to a variable could raise an assertion due to improper cleanup related to `GROUP BY` or `ORDER BY` clauses. (Bug #18486509)
- The `validate_password` plugin was not installed by RPM packages for platorms using systemd or SysV-style initialization scripts. (Bug #18438833)
- Some queries involving spatial relation checks would produce correct results for `MyISAM` tables but not `InnoDB` tables. (Bug #18422162)
- Deleting rows from an empty `MyISAM` table with a spatial index resulted in a spurious error message about a corrupt index. (Bug #18412756)
- Boolean full-text searches for `MyISAM` tables could fail. (Bug #18279587)
- A client that attempted to establish SSL connections from a large number of threads simultaneously could exit with a segmentation fault. (Bug #18052165)

- Deletes from `CSV` tables could cause a server exit. (Bug #17902624)
- For `HANDLER` read statements that scanned a spatial index, type conversion errors of values read from the index could cause a server exit. (Bug #17846865)
- A query with an `IN` subquery where the left-hand side was a scalar subquery might cause a server exit. (Bug #17832047)
- The server could exit under conditions when a query contained the following construct but produced an empty result:

  ```
  literal-valued row constructor <=> (subquery containing UNION)
  ```

  (Bug #17668844)
- If ownership of memory allocation was transferred between threads, Performance Schema memory instrumentation could report memory use of the threads incorrectly. (Bug #17473077)
- For debug builds, statements including table-less subqueries could raise an assertion when executed within scheduled events. (Bug #17435114)
- The `--help` message displayed by `mysql_secure_installation` did not show options related to option-file processing, such as `--defaults-file`. (Bug #17339009)
- Memory usage values in the `memory_summary_global_by_event_name` Performance Schema table could be negative. (Bug #17243619)
- For debug builds, an assertion could be raised in character-set conversion code due to an overly strict condition. (Bug #13740934)
- An event scheduler thread could be freed improperly, potentially leading to a server exit. (Bug #77593, Bug #21145277, Bug #21053167)
- `mysql-systemd-start` failed if `datadir` was set in `/etc/my.cnf`. (Bug #77357, Bug #21262883)
- The unused and unmaintained `BUILD/build_mccge.sh` script has been removed from the source tree. (Bug #77336, Bug #21246941)
- `ST_IsValid()` could return false for some valid `MultiPolygon` arguments. (Bug #77317, Bug #21238969)
- `ST_Buffer()` with a `LineString` argument could produce a `Polygon` that self-intersected. (Bug #77316, Bug #21238614)
- Compilation failed when building MySQL without the Performance Schema. (Bug #77292, Bug #21229433)
- Updating the `setup_consumers` table to set history or long-history consumers had no affect on historical event logging for existing threads. (Bug #77278, Bug #21223458)
- `SHOW GLOBAL STATUS` `Com_`*`xxx`* counters did not reflect `SELECT` statements. (Bug #77231, Bug #21186946)
- The outdated and not-maintained `plugin/daemon_example/ChangeLog` file was removed. (Bug #77188, Bug #21168681)
- `ST_ConvexHull()` could return incorrect results for `MultiLineString` arguments. (Bug #77167, Bug #21153716)

- Executing a prepared `EXPLAIN` statement could cause the server to hang. (Bug #77144, Bug #21139522)
- If the server was started with the `--ssl-cipher` option, autogeneration and autodetection of SSL certificates did not work. (Bug #77078, Bug #21108296)
- Optimizer hint query block names are identifiers, but the parser did not recognize valid identifer names such as `123a` when used in *`@query_block_name`* syntax. (Bug #77047, Bug #21095608)
- For `mysqldump`, the `-T` option is supposed to be the short form of the `--tab` option, but was mistakenly associated with `--debug-info` instead. (Bug #77037, Bug #21088793)

  References: This issue is a regression of: Bug #66854.
- SSL certificates autogenerated by the server could have CN values that exceeded 64 characters. In that case, the server now omits the *`_server_version`* part of the CN values so the length falls within 64 characters. (Bug #77036, Bug #21087159)
- The Common Name value written by `mysql_ssl_rsa_setup` to the `client-cert.pem` client certificate file was `MySQL_Server_`*`suffix`*`_Auto_Generated_Server_Certificate` rather than `MySQL_Server_`*`suffix`*`_Auto_Generated_Client_Certificate`. (Bug #77035, Bug #21087116)
- Deallocation of Debug Sync structures within the `InnoDB` handlerton close connection method could raise an assertion. (Bug #77005, Bug #21069721)
- Queries on a geometry column returned an error instead of a result if there existed a `UNIQUE` index on the column. (Bug #77000, Bug #21067378)
- An assertion could be raised if a multiple-table `UPDATE` of a view, where the same column was used in the `SET` and `JOIN` clauses, was used as a prepared statement. (Bug #76962, Bug #21045724)
- With row-based binary logging, automatic dropping of a scheduled event that had reached the end of its lifetime could raise an assertion. (Bug #76958, Bug #21041908)
- If a single-table subquery had identical `GROUP BY` and `ORDER BY` clauses on a `UNIQUE NOT NULL` column, the results could be incorrectly ordered. (Bug #76947, Bug #21038929)
- The `PARSE_GCOL_EXPR` keyword used internally by the parser was treated as a reserved word and thus could not be used as an identifier without quoting it. (Bug #76943, Bug #21035515)
- When the directory specified for the `secure_file_priv` system variable did not exist, the server produced a `Failed to normalize the argument` error message. It now produces a message indicating that the directory did not exist. The same problem occurred for the `--datadir` option to `mysql_ssl_rsa_setup` and was fixed the same way. (Bug #76918, Bug #21021894)
- For logging to the binary log, the server could rewrite `CREATE USER` and `ALTER USER` statements, adding an `ACCOUNT UNLOCK` clause not present in the original statement. This could unlock locked accounts and cause differences between master and slave servers. The clause is no longer written unless present in the original statement. (Bug #76911, Bug #20996273)
- The `INDEX_NAME` column of the Performance Schema `table_io_waits_summary_by_index_usage` table could sometimes show incorrect index names for tables until they had been in use for some time. (Bug #76882, Bug #20980217)
- Compilation could fail in the query rewrite plugin code for some `CMake` options. (Bug #76800, Bug #20937654)
- `DO` statements containing multiple expressions could result in a memory leak.

A consequence of the bug fix is that `DO` statement errors previously converted to warnings now are returned as errors. (Bug #76779, Bug #20924241, Bug #17479887)

- Previously, SSL files created automatically by the server were valid for one year. The validity period has been extended to ten years (the same as SSL files created by `mysql_ssl_rsa_setup`). (Bug #76778, Bug #20923066)
- `mysql_upgrade` failed if the `show_compatibility_56` system variable was enabled. (Bug #76757, Bug #20914786)
- Unaligned memory access could cause spatial operations to fail. (Bug #76748, Bug #20911624)
- Identifiers in normalized statements were sometimes quoted and sometimes not, an inconsistency that caused matching failure for statement digests and digest texts. This caused problems for Performance Schema aggregation by digest. Identifiers now are quoted consistently. (Bug #76723, Bug #20896539)
- Ubuntu packages were missing dependencies for `killall` and `psmisc`. (Bug #76716, Bug #20893836)
- `SHOW GLOBAL VARIABLES` and selecting from the `INFORMATION_SCHEMA GLOBAL_VARIABLES` table resulted in a spurious warning about the `sql_log_bin` system variable. (Bug #76626, Bug #20854952)
- `mysqld --help --verbose` was slow if the `InnoDB` buffer pool was configured to a large size. Now with those options, buffer pool allocation is not performed. (Bug #76625, Bug #20856397)
- An assertion could be raised for queries with a `GROUP BY` clause and a table for which the optimizer identified multiple candidate indexes. (Bug #76576, Bug #20819199)
- `CREATE USER` events written to the binary log included the new `ACCOUNT` syntax even with `log_backward_compatible_user_definitions` enabled. (Bug #76560, Bug #20814051)
- The server rejected empty `COM_SHUTDOWN` packets. (Bug #76552, Bug #20810928)

  References: This issue is a regression of: Bug #14525642.
- For some startup errors, the server could call `exit()` before shutting down plugins and thus failed to invoke their `atexit()` handlers. (Bug #76532, Bug #20798617)
- In `sql/handler.h`, `HA_ATTACHABLE_TRX_COMPATIABLE` and `HA_GENERATED_COLUMNS` were defined with the same value. (Bug #76503, Bug #20783191)
- `mysqlimport --use-threads` did not actually use multiple threads. (Bug #76480, Bug #20772273)
- The mutex used for the optimizer cost model cost-constant cache was not instrumented by the Performance Schema. This instrument is now available as `wait/synch/mutex/sql/Cost_constant_cache::LOCK_cost_const`. (Bug #76460, Bug #20755430)
- These statement-timeout problems were corrected:
  - An assertion could be raised with `max_statement_time` set greater than zero and multiple concurrent sessions executing certain `EXPLAIN` statements.
  - The error message indicating that statement execution was interrupted referred to the `max_statement_time` system variable, even if the relevant timeout was a per-statement value. The error message is now more generic.
  - It was not possible to set the `max_statement_time` system variable at server startup.
  - Setting `max_statement_time` to set a statement timeout could cause memory leaks or assertion failures on Windows.

- Attempting to kill statements that use attachable transactions caused subsequent statements to function improperly, resulting in assertion failures. A `max_statement_time` timeout on such statements could produce a similar outcome.

  

  Note

  Subsequent to these changes, the `max_statement_time` system variable was renamed to `max_execution_time`.

  (Bug #76446, Bug #20788811, Bug #76915, Bug #21021670, Bug #76916, Bug #21021754, Bug #20705648, Bug #20705642, Bug #75782, Bug #20507804)

  References: See also: Bug #77461, Bug #21306646.

- Attempts to create a foreign key matching a `FULLTEXT` index failed. For debug builds, attempts to create a foreign key matching a `SPATIAL` index raised an assertion. (Bug #76445, Bug #20752436)
- The `ORDER BY` clause of a derived table was appended to an `INSERT` statement, but ordering for a table being inserted into is irrelevant and caused a server exit. (Bug #76436, Bug #20753569)
- A failing `ALTER TABLE` tablespace operation (`DISCARD TABLESPACE` or `IMPORT TABLESPACE` could produce an incorrect internal tablespace state, causing a succeeding statement to fail. (Bug #76424, Bug #20748660)
- Enabling the `sql_buffer_result` system variable could cause a server exit for multiple-table `UPDATE` statements. (Bug #76419, Bug #20748537)
- The value of `secure_file_priv` displayed as `NULL` for both `--secure_file_priv=NULL` (correct) and `--secure_file_priv=""` (incorrect). (Bug #76401, Bug #20741572)
- A `Provides` rule in RPM `.spec` files misspelled “mysql-embedded” as “mysql-emdedded”. (Bug #76385, Bug #20734434)
- Compiling using Clang with AddressSanitizer (ASAN) enabled caused the `gen_lex_hash` utility to abort with memory leak check failures. (Bug #76351, Bug #20720615)
- `SHOW CREATE TABLE` did not correctly display generated columns that had a character set defined. (Bug #76328, Bug #20709462)
- The `Com_stmt_reprepare` status variable was missing from the `global_status` and `session_status` Performance Schema tables. (Bug #76305, Bug #20697446)
- Attempts to establish SSL connections to a Community Edition server failed if the client had a password and the server did not have the general query log enabled. (Bug #76286, Bug #20693153)
- Prepared statement execution statistics were not correctly tracked in the `prepared_statements_instances` Performance Schema table. (Bug #76284, Bug #20692556)
- Some server warnings referred to the now-deprecated `mysql_install_db` command. (Bug #76251, Bug #20681412)
- For multibyte character sets, `LOAD DATA` could fail to allocate space correctly and ignore input rows as a result. (Bug #76237, Bug #20683959)

  References: This issue is a regression of: Bug #14653594.

- Compilation could fail due to a missing dependency on `lex_token.h` for `sql_yacc.cc.o`. (Bug #76235, Bug #20678411)

- `EXPLAIN` for a query containing an uncorrelated subquery could attempt to materialize the subquery twice, raising an assertion. (Bug #76205, Bug #20665051)
- Debian and Ubuntu package installers neglected to run `mysql_ssl_rsa_setup` during installation. (Bug #76163, Bug #20650118)
- Global status variables related to SSL certificate metadata were available only within sessions established using SSL. (Bug #76157, Bug #20648276)
- The `mysql_real_escape_string_quote()` C API function failed to escape backtick (`) characters when the `NO_BACKSLASH_ESCAPES` SQL mode was disabled. (Bug #76146, Bug #20645725)
- The message displayed to indicate that a password was expired and must be reset referred to the deprecated `SET PASSWORD` statement. It now refers to `ALTER USER`. (Bug #76053, Bug #20602572)
- `ALTER USER` statements that named an authentication plugin did not check whether the plugin is valid. (Bug #76052, Bug #20602525)
- If a proxy user expired the password of the proxied user, the current proxy user session was affected (the server considered its password expired). To execute SQL statements again, it was necessary for the proxy user to disconnect and reconnect again. (Bug #76043, Bug #20599280)
- `mysqld --help --verbose` tried to perform actions that have nothing to do with displaying a help message: Locking files, initializing system files, and checking for a `plugin` table. (Bug #75995, Bug #20581228)
- For a `SET = (subquery)` statement within a stored procedure, the server could exit if a subquery transformation was performed. (Bug #75994, Bug #20583321)
- Code for reading and writing the grant tables assumed that these were `MyISAM` tables and did not handle errors that can be thrown if the tables are handled by a different storage engine. (Bug #75955, Bug #20561087)
- Several spatial function issues were resolved by use of Boost.Geometry for GIS algorithms:
  - `ST_Centroid()` with a `MultiPolygon` argument could produce incorrect results.
  - Multiple calls to `ST_Intersection()` could return inconsistent results for some arguments.
  - `ST_Within()` and `ST_Touches()` could return incorrect results for some arguments.

  (Bug #75829, Bug #20508769, Bug #69425, Bug #19270344, Bug #69538, Bug #19270334)
- Nonoptimal cost estimates for key lookups could cause some queries to be executed with a table scan rather than key lookups. (Bug #75695, Bug #20443863)
- Operations on a string exceeding `max_allowed_packet` bytes could return `NULL` and incorrectly replace an existing value in `UPDATE` statements with `NULL` rather than failing. (Bug #75539, Bug #20376498)
- The MeCab full-text parser plugin was omitted from RPM and Debian packages. (Bug #75429, Bug #20315007)
- `EXPLAIN` for `INSERT ... SELECT` statements into a multiple-table view always displayed the first table of the view as the table being inserted into, even if it was not. (Bug #75424, Bug #20310257)
- The parser could dereference a null pointer after an out-of-memory error. (Bug #75372, Bug #20294206)
- Some queries could return different results depending on whether the `semijoin` flag of the `optimizer_switch` system variable was enabled or disabled. (Bug #75270, Bug #20239912)

- With a small thread stack, queries with many expressions could produce a thread stack overrun error. (Bug #74985, Bug #20087571)
- On platforms where the `char` is unsigned, the server was unable to parse collation definitions that included non-7-bit ASCII characters. Affected platforms include ARM and PowerPC. Thanks to Alexey Kopytov for the patch. (Bug #74891, Bug #20928289, Bug #21682439)
- If the server was started with the `explicit_defaults_for_timestamp` system variable enabled, `CREATE TABLE` statements that defined a column as `TIMESTAMP NOT NULL` failed. (Bug #74529, Bug #19881933)
- In the `threads` Performance Schema table, the `PROCESSLIST_STATE` and `PROCESSLIST_INFO` values did not change for the `thread/sql/main` main thread instrument as the thread state changed. (Bug #74517, Bug #19887143)
- On OS X 10.10 (Yosemite), `mysqld` failed to start automatically. The startup item has been replaced with a `launchd` job, which enables the preference pane checkbox for automatic startup to work again. (Bug #74434, Bug #19858350)
- Specifying a bad `--init-file` option value could cause the server to hang at startup. (Bug #74402, Bug #19822257)
- `mysql_install_db` did not write a date to the `.mysql_secret` file. (Bug #74006, Bug #19659004)
- Incorrect results could be produced tor views and derived tables on the inner side of an outer join and from which non-nullable expressions such as literals were selected. (Bug #73953, Bug #20841369, Bug #67014, Bug #15967464, Bug #65936, Bug #14358878, Bug #67300, Bug #15936817, Bug #76327, Bug #20708288)
- If a spatial column contained invalid spatial data, creating a `SPATIAL` index on the column failed to produce an error. (Bug #73871, Bug #19593342)
- Certain queries for the `INFORMATION_SCHEMA TABLES` and `COLUMNS` tables could lead to excessive memory use when there were large numbers of empty `InnoDB` tables. (Bug #72322, Bug #18592390)
- Large integer literals converted to floats for comparison with decimal data could lose precision and produce incorrect results. (Bug #72056, Bug #18411494, Bug #21139707)
- When choosing join order, the optimizer could incorrectly calculate the cost of a table scan and choose a table scan over a more efficient `eq_ref` join. (Bug #71584, Bug #18194196)
- The server interpreted `--tc-heuristic-recover` option values incorrectly due to an off-by-one error. Thanks to Laurynas Biveinis for the patch. (Bug #70860, Bug #19771769)
- On OS X, the `vio_io_wait()` call could cause stack corruption for a large number of file descriptors (more than `FD_SETSIZE`). (Bug #69903, Bug #17259750)
- Queries that included a `HAVING` clause based on nondeterministic functions could produce incorrect results. (Bug #69638, Bug #17055185)
- MySQL failed to compile using OpenSSL 0.9.8e. (Bug #68999, Bug #16861371)
- For `mysqlslap`, the combination of `--auto-generate-sql-secondary-indexes` and `--auto-generate-sql` failed because it tried to insert 36-digit UUID values into a `VARCHAR(32)` column. Thanks to Tsubasa Tanaka for the patch. (Bug #55265, Bug #11762644)

Changes in MySQL 5.7.7 (2015-04-08, Release Candidate)

- Installation Notes

Installation Notes

- Installers for more binary distribution types provide secure deployment. This includes installers for SLES, Solaris, OS X, FreeBSD, and Linux generic binary compressed `tar` distributions. These installers create a single `'root'@'localhost'` account without other `root` or anonymous-user accounts, and do not create a `test` database accessible by any user. (Some installers may provide options to create additional accounts or a test database, but only if selected by the user.) Installers that run interactively and can ask the installing user for the initial `root` password do so. Noninteractive installers generate a random `root` password that the administrator can use to connect to the server the first time and choose a new password.

Optimizer Notes

- It is now possible to provide hints to the optimizer within individual SQL statements, which enables finer control over statement execution plans than can be achieved using the `optimizer_switch` system variable. Optimizer hints are specified as `/*+ ... */` comments following the `SELECT`, `INSERT`, `REPLACE`, `UPDATE`, or `DELETE` keyword of statements or query blocks. Hints are also permitted in statements used with `EXPLAIN`, enabling you to see how hints affect execution plans. Examples:

  ```
  SELECT /*+ NO_RANGE_OPTIMIZATION(t3 PRIMARY, f2_idx) */ f1
    FROM t3 WHERE f1 > 30 AND f1 < 33;
  SELECT /*+ BKA(t1) NO_BKA(t2) */ * FROM t1 INNER JOIN t2 WHERE ...;
  SELECT /*+ NO_ICP(t1, t2) */ * FROM t1 INNER JOIN t2 WHERE ...;
  EXPLAIN SELECT /*+ NO_ICP(t1) */ * FROM t1 WHERE ...;
  ```

  For more information, see Optimizer Hints.

Packaging Notes

- Several binary distribution types have been made more modular, to split out test components into a separate distribution file. This reduces the size of the main download. In addition to the previously available test/debug distributions already available for Windows Zip archives, RPM packages, and Debian packages, the current release makes separate test distributions available for Solaris PKG files, and generic binary Linux and OS X compressed `tar` packages. These separate distributions have “test” in the distribution file name.

  Generally, use of a test distribution requires that the main distribution is also installed. Additionally, for Solaris, the main and test distributions must be for the same version of MySQL.

  References: See also: Bug #20613327, Bug #20546298.

Performance Schema Notes

- The `events_statements_history` and `events_transactions_history` consumers now are enabled by default.

  References: See also: Bug #71207, Bug #18376132.

Security Notes

- Previously, proxy user mapping was available only for authentication plugins that implemented that capability for themselves. The MySQL server itself now can map proxy users according to granted proxy privileges. If the new `check_proxy_users` system variable is enabled, the server performs proxy user mapping for any authentication plugin that requests it. By default, `check_proxy_users` is disabled, so the server performs no proxy user mapping even for authentication plugins that request it.

  In addition, the `mysql_native_password` and `sha256_password` built-in authentication plugins have been modified to take advantage of this server capability, and thus now are able to support proxy users. The new `mysql_native_password_proxy_users` and `sha256_password_proxy_users` system variables control whether each plugin requests proxy user mapping. By default, both variables are disabled, which produces behavior that is backward compatible with previous releases.

  For information about user proxying, see Proxy Users.

- The C client library now attempts to establish a secure connection by default whenever the server supports secure connections. This affects client programs as follows:

  - In the absence of an `--ssl` option, the client falls back to an unencrypted connection if a secure connection cannot be established.
  - To require a secure connection and fail if one cannot be established, invoke the client with `--ssl` or a synonym (`--ssl=1`, `--enable-ssl`).
  - To use an unencrypted connection, invoke the client with `--ssl=0` or a synonym (`--skip-ssl`, `--disable-ssl`).

  For more information, see Command Options for Secure Connections.

  This change affects these standard MySQL client programs: `mysql`, `mysql_config_editor`, `mysql_install_db`, `mysql_plugin`, `mysql_secure_installation`, `mysql_upgrade`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlshow`, and `mysqlslap`. It will also affect new releases of MySQL Connectors that are based on the C client library: Connector/C, Connector/C++, and Connector/ODBC.

Spatial Data Support

- The `ST_Buffer()`, `ST_Difference()`, `ST_Distance()`, `ST_Intersection()`, `ST_IsSimple()`, `ST_SymDifference()`, and `ST_Union()` functions have been reimplemented to use the functionality available in Boost.Geometry. The functions may raise an exception for invalid geometry argument values when the previous implementation may not have.

  In addition, `ST_Buffer()` now takes up to three optional arguments to specify point, join, and end strategies that influence buffer computation. Values for strategy arguments are produced using the new `ST_Buffer_Strategy()` function. See Spatial Operator Functions.

sys Schema Notes

- MySQL distributions now include the `sys` schema, a set of objects that helps DBAs and developers interpret data collected by the Performance Schema. `sys` schema objects can be used for typical tuning and diagnosis use cases.

For new installations, the `sys` schema is installed by default during data directory initialization if you use `mysqld` with the `--initialize` or `--initialize-insecure` option, or if you use `mysql_install_db`. To permit this behavior to be suppressed, `mysql_install_db` now has a `--skip-sys-schema` option. `mysqld` has no such option, but if you initialize the data directory using `mysqld --initialize` (or `--initialize-insecure`) rather than `mysql_install_db`, you can drop the `sys` schema manually after initialization if it is unneeded.

For upgrades, `mysql_upgrade` installs the `sys` schema if it is not installed, and upgrades it to the current version otherwise. To permit this behavior to be suppressed, `mysql_upgrade` now has a `--skip-sys-schema` option.

`mysql_upgrade` returns an error if a `sys` schema exists but has no `version` view, on the assumption that absence of this view indicates a user-created `sys` schema. To upgrade in this case, remove or rename the existing `sys` schema first.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` to install the `sys` schema.

For more information, see MySQL sys Schema.

Functionality Added or Changed

- **Important Change; InnoDB:** The following changes were made to `InnoDB` configuration parameter default values:

  - The `innodb_file_format` default value was changed to `Barracuda`. The previous default value was `Antelope`. This change allows tables to use Compressed or Dynamic row formats.

  - The `innodb_large_prefix` default value was changed to `ON`. The previous default was `OFF`. When `innodb_file_format` is set to `Barracuda`, `innodb_large_prefix=ON` allows index key prefixes longer than 767 bytes (up to 3072 bytes) for tables that use a Compressed or Dynamic row format.

  - The `innodb_strict_mode` default value was changed to `ON`. The previous default was `OFF`. When `innodb_strict_mode` is enabled, `InnoDB` raises error conditions in certain cases, rather than issuing a warning and processing the specified statement (perhaps with unintended behavior).

  The configuration parameter default changes described above may affect replication and `mysqldump` operations. Consider the following recommendations when using the new default settings:

  - When replicating or replaying `mysqldump` data from older MySQL versions to MySQL 5.7.7 or higher, consider setting `innodb_strict_mode` to `OFF` to avoid errors. Target settings should not be more strict than source settings.

  - When replicating from MySQL 5.7.7 or higher to older slaves, consider setting `innodb_file_format=Barracuda` and `innodb_large_prefix=ON` on the slave so that the target and source have the same settings.

  The following file format related configuration parameters are deprecated and will be removed in a future release:

  - `innodb_file_format`

  - `innodb_file_format_check`

  - `innodb_file_format_max`

- `innodb_large_prefix`

These four configuration parameters were provided 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. Also, the file format scheme, by which named file formats would be introduced as new features were added, was not used after introduction of the `Barracuda` file format. `InnoDB` formats have changed since the introduction of the `Barracuda` file format, but new named file formats have not been added.

If non-default values are used for any of the four deprecated parameters, `InnoDB` prints a deprecation and removal warning to the server error log. The same warning is issued to the client if the parameters are set dynamically using a `SET` statement.

- **Important Change; InnoDB:** The `innodb_buffer_pool_dump_at_shutdown` and `innodb_buffer_pool_load_at_startup` configuration options are now enabled by default. With this change, a percentage of most-recently-used buffer pool pages is dumped at server shutdown and restored at server startup. This behavior helps avoid a lengthy buffer pool warmup period after restarting the server, particularly for instances with large buffer pools. The `innodb_buffer_pool_dump_pct` option defines the percentage of buffer pool pages that are dumped. The default value for `innodb_buffer_pool_dump_pct` is reduced from 100 to 25.

  These configuration option default value changes represent a change in behaviour at server shutdown and startup. If you prefer the previous default values, it is recommended that you configure the options explicitly in your MySQL configuration file after upgrading to MySQL 5.7.7 or later, and before restarting the server.

  For more information, see Saving and Restoring the Buffer Pool State.

- **Important Change; InnoDB:** The `innodb_checksum_algorithm` default value is now `crc32`. The previous default setting was `innodb`. This change also means that `innodb_checksums=ON` is now equivalent to `innodb_checksum_algorithm=crc32` instead of `innodb_checksum_algorithm=innodb`.

- **InnoDB:** The `InnoDB` MeCab full-text parser plugin now supports the `eucjpms`, `cp932`, and `utf8mb4` character sets. (Bug #20534096)

- **InnoDB:** To address a scalability bottleneck for some workloads where `LOCK_grant` is locked in read-mode, `LOCK_grant` locks are now partitioned. Read lock requests on `LOCK_grant` now acquire one of multiple `LOCK_grant` partitions. Write locks must acquire all partitions.

  To address another scalability bottleneck, the server no longer performs unnecessary lock acquisitions when creating interal temporary tables.

  References: See also: Bug #72829, Bug #20023139.

- **Replication:** The defaults of some replication related variables have been modified. The following changes have been made:

  - `binlog_gtid_simple_recovery=TRUE`
  - `binlog-format=ROW`
  - `binlog_error_action=ABORT_SERVER`
  - `sync_binlog=1`

- `slave_net_timeout=60`

Additionally, the session scope of `gtid_executed` has been deprecated. The global scope of `gtid_executed` remains supported.

- **Replication:** The XA implementation in MySQL has been made much more compatible with the XA specification. A prepared XA transaction is no longer rolled back at disconnect. Now when replicating using the binary log, an XA transaction in `PREPARED` state persists in the binary log until an explicit `XA COMMIT` or `XA ROLLBACK` statement is issued. In prior versions, an XA transaction that was in `PREPARED` state would be rolled back on clean server shutdown or client disconnect. Similarly, an XA transaction that was in `PREPARED` state would still exist in `PREPARED` state in case the server was shutdown abnormally and then started again, but the contents of the transaction could not be written to the binary log. As part of this feature a new event, `XA_prepare_log_event`, has been added to track XA transactions in the `PREPARED` state and enable them to be replicated. To finalize a two-phase XA transaction, the `XA COMMIT` or `XA ROLLBACK` is recorded separately in the binary log, possibly interleaving with other transactions. XA transactions committed with `XA COMMIT ONE PHASE` are logged as one part using `XA_prepare_log_event`.

  References: See also: Bug #12161, Bug #11745231, Bug #75204, Bug #20214365.

- Performance Schema stage event instruments that provide statement progress information now are enabled and timed by default. The affected instruments are those displayed by this statement:

```
SELECT * FROM performance_schema.setup_instruments WHERE
ENABLED='YES' AND NAME LIKE "stage/%";
```

  (Bug #20685859)

- The `XID` column of Performance Schema transaction tables (for example, `events_transactions_current`) has been split into three columns to permit access to the component parts of XID values. The new columns are `XID_FORMAT_ID`, `XID_GTRID`, and `XID_BQUAL`. (Bug #18320361)

- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.

- In MySQL 5.7.6, the `NO_AUTO_CREATE_USER` was deprecated. (It is preferable to create MySQL accounts with `CREATE USER` rather than `GRANT`.) Now the default SQL mode includes `NO_AUTO_CREATE_USER` and assignments to `sql_mode` that change the `NO_AUTO_CREATE_USER` mode state produce a warning, except assignments that set `sql_mode` to `DEFAULT`. `NO_AUTO_CREATE_USER` will be removed in a future MySQL release, at which point its effect will be enabled at all times (`GRANT` will not create accounts).

- Previously, view definitions were not permitted to contain derived tables (subqueries) in the `FROM` clause. This restriction has now been lifted.

  Previously, derived tables (subqueries) or views in the `FROM` clause could not be merged into the outer query if they contained subqueries in the `SELECT` list. Instead, they were processed using materialization. This restriction has now been lifted so that queries previously executed using materialization can be executed more quickly using merging. (Bug #12755, Bug #11745276, Bug #60417, Bug #11865600)

Bugs Fixed

- **InnoDB:** `FLUSH TABLE ... FOR EXPORT`, which is an unsupported operation for tables residing in a general tablespace, failed to report a warning message. (Bug #20631305)
- **InnoDB:** The MeCab parser accessed the byte beyond the length of the document resulting in an `Invalid read of size 1` Valgrind error. Document allocation size is now length + 1 byte. (Bug #20589432)
- **InnoDB:** A debug assertion should not be raised for static mutexes. (Bug #20588765)
- **InnoDB:** Tablespace and file path data updates to internal system tables during startup caused undo log corruption. The updates were made before calling `truncate::fixup_tables()`, which does not expect to encounter changed pages. (Bug #20534616)
- **InnoDB:** A `TRUNCATE TABLE` operation on a temporary table raised an assertion. The temporary table object was incompletely constructed when reloaded from `SYS_TABLES`. (Bug #20527363, Bug #72080)
- **InnoDB:** `Return value 16 when calling pthread_mutex_destroy()` warnings were returned during `atexit()` processing. (Bug #20513522)
- **InnoDB:** A debug variable used to modify the first page of a tablespace raised an assertion when set to a nonexistent tablespace ID. (Bug #20511314, Bug #75833, Bug #19865274, Bug #74481)
- **InnoDB:** A buffer pool load operation raised an assertion when attempting to read pages that were placed out of tablespace bounds by a preceding `TRUNCATE TABLE` operation. (Bug #20474542)
- **InnoDB:** Removal of a foreign key object from the data dictionary cache during error handling caused the server to exit. (Bug #20442523)
- **InnoDB:** An assertion was raised during a redo log resize operation that was triggered by a file size mismatch encountered during recovery. Code introduced in MySQL 5.6.8 to automatically handle redo log file size mismatches failed to ensure that the buffer pool was clean prior to recreating redo log files. (Bug #20425387)
- **InnoDB:** An `InnoDB memcached extra_col_value[]` array was freed without checking the allocated flag, causing a server exit. (Bug #20400373)
- **InnoDB:** A DML operation performed while a flushing operation was in progress raised a memcached-related assertion. (Bug #20390277)
- **InnoDB:** A `CHECK TABLE` operation on a table with a spatial index raised an assertion. The `row_sel_sec_rec_is_for_clust_rec` function failed to handle an externally stored field. (Bug #20311344)
- **InnoDB:** `CHECK TABLE` reported misplaced rows after an in-place `ALTER TABLE` operation on a “partitioned by key” table. An in-place operation that drops and adds the primary key should not be permitted for tables that are partitioned by key. (Bug #20190520)
- **InnoDB:** Estimates that were too low for the size of merge chunks in the result sorting algorithm caused a server exit. (Bug #20049521)
- **InnoDB:** An `ALTER TABLE ... RENAME` operation raised an invalid assertion. The assertion code used an incorrect transaction object. (Bug #18523599)

  References: This issue is a regression of: Bug #17447500.
- **InnoDB:** Running `mysql_upgrade` after a binary upgrade to MySQL 5.7.6 caused a server exit on a system with tablespace data files that were created in MySQL 5.1 or earlier. The fix for Bug #17345513 in MySQL 5.7.6 failed to address all instances of garbage `FIL_PAGE_TYPE` values in tablespace data files created in MySQL 5.1 or earlier.

With this patch, the manual process described in the MySQL 5.7.6 release notes entry for Bug #17345513 for repairing non-index pages that contain invalid `FIL_PAGE_TYPE` values is no longer necessary.

The patch for Bug #17345513 also failed to recompute page checksums after resetting invalid `FIL_PAGE_TYPE` values. Upon restarting the server, a failure would occur due to an apparent page corruption. The page checksum is now recomputed before the new `FIL_PAGE_TYPE` value is written to the data file. (Bug #76262, Bug #20691930)

References: This issue is a regression of: Bug #17345513.

- **InnoDB:** For full-text searches, the optimizer could choose an index that does not produce correct relevancy rankings. (Bug #74686, Bug #19950568)
- **Partitioning:** The MySQL Server unnecessarily requested the default number of partitions for a table whenever it opened a partitioned table. This was unnecessary since the server already has this information about the table to be opened. Now the server requests this information only when needed—that is, only if it is creating or altering a partitioned table. (Bug #76007, Bug #20585753)
- **Partitioning:** A number of `ALTER TABLE` statements that attempted to add partitions, columns, or indexes to a partitioned table while a write lock was in effect for this table were not handled correctly. (Bug #74451, Bug #74478, Bug #74491, Bug #74560, Bug #74746, Bug #74841, Bug #74860, Bug #74869, Bug #19856162, Bug #19864284, Bug #19873019, Bug #19891663, Bug #19990815, Bug #20026661, Bug #20031966, Bug #20033503, Bug #19827845)
- **Partitioning:** Executing an `ALTER TABLE` on a partitioned table on which a write lock was in effect could cause subsequent SQL statements on this table to fail. (Bug #74288, Bug #74634, Bug #19784790, Bug #19918805)

  References: See also: Bug #19856162, Bug #74451.
- **Replication:** Some memory copy operations being performed on the `replication_connection_status` Performance Schema table were using an incorrect length, which could lead to a buffer overflow error or truncated output. The fix ensures that the correct length is used. (Bug #20535692)
- **Replication:** When using multi-source replication on a multi-threaded slave (where `slave_parallel_workers` is greater than 1), and `slave_transaction_retries` was greater than 1, the slave would fail to open the relay log file. This was due to the slave worker incorrectly constructing the relay log file path for its replication channel. (Bug #20448413)
- **Replication:** When the `automatic_sp_privileges` variable is set, the server automatically grants the `EXECUTE` and `ALTER ROUTINE` privileges to the creator of a stored routine, if the user does not already have these privileges. When a privileged user creates a procedure with `DEFINER` as a non privileged user on a master, the current user is considered to be a privileged user and the `mysql.procs_priv` table is not updated. When such a statement was replicated to slave, the non-privileged `DEFINER` was considered as the current user on the slave and privileges were being allocated. This caused a difference in the privileges that were being allocated on the master and the slave. The fix ensures that creater of the stored routine is added to the binary log, and the slave now checks first if the user exists before granting privileges. To maintain compatibility with previous versions, the `DEFINER` is used when the `INVOKER` is not available. As part of this fix, anonymous users can be used to replicate from master to slave. (Bug #20049894)
- **Replication:** If the I/O thread on a replication slave failed while it was in the initialization phase, it was not providing this information in the `Last_IO_Error` field after issuing `SHOW SLAVE STATUS`. The fix ensures that such errors are reported correctly in the `Last_IO_Error` field. (Bug #18909984)

- **Replication:** After issuing `RESET SLAVE`, the `RECEIVED_TRANSACTION_SET` field in the `performance_schema.replication_connection_status` table showed incorrect values. This could cause an incorrect string value error. (Bug #18751585, Bug #19840342)
- **Replication:** When `gtid_mode=ON` and `slave_net_timeout` was set to a low value, the slave I/O thread could appear to hang. This was due to the slave heartbeat not being sent regularly enough when the dump thread found many events that could be skipped. The fix ensures that the heartbeat is sent correctly in such a situation. (Bug #74607, Bug #19975697)
- `CMake` failed to detect the OpenSSL version properly for recent versions of OpenSSL (the format of the version string changed). (Bug #20756770)
- `GRANT` and `ALTER USER` could clear the password-expiration flag for operations not related to resetting the password. (Bug #20634154)
- For upgrades from MySQL 5.6 to 5.7 that involve moving `mysql.user` table passwords from the `Password` column to the `authentication_string` column, `mysql_upgrade` neglected to handle rows with an empty `plugin` value and a pre-4.1 password hash. (Bug #20614545)
- For table-modifying statements, the parser could dereference the parse tree without checking for out-of-memory conditions or null pointers. (Bug #20607407)
- `mysql_stmt_prepare()` could leak memory allocated to metadata. (Bug #20598261)
- Debian packages were missing some dependencies. (Bug #20561621)
- A server exit could be caused by a query that contained a `HAVING` clause, which itself contained an `IN()` subquery predicate, where the subquery referenced a column of the query. (Bug #20558891)
- In MySQL 5.7.6, the `PASSWORD()` function was deprecated, but no warning was produced when it was invoked. Similarly, the `old_passwords` system variable was deprecated, but no warning was produced when it was set. (Bug #20545464)
- The server could exit if a client using the cleartext authentication plugin attempted to connect with an empty password. (Bug #20537246)
- A query cache invalidation function used a too-small buffer for holding encoded database names, which could result in a server exit. (Bug #20528928)
- Valgrind warnings were silenced for display of GTID-related debug information. (Bug #20506672)
- Some queries that had a derived table (subquery) in the `FROM` clause could raise an assertion. (Bug #20487336)
- A table-modifying statement that followed a failed table-modifying could result in a server exit. (Bug #20460208)
- Union queries over views containing `ENUM` or `SET` values were not handled properly. (Bug #20456178)
- A natural left join between between a derived table and a regular table, joined with another natural left join to another regular table could cause a server exit. (Bug #20455184)
- The optimizer could try to create an index of the wrong data type on internal temporary tables. (Bug #20454833)
- A multiple-table `UPDATE` statement where one of the specified tables was a derived table could cause a server exit. (Bug #20454533)

- Mishandling of SRID values within `ST_GeomFromGeoJSON()` could cause an assertion to be raised. (Bug #20416705)
- Under certain conditions, `LCASE()`, `DECODE()`, and `ENCODE()` could have source and destination overlap in memory-copying operations. (Bug #20315088, Bug #75931, Bug #20554017)
- `ST_Distance()` could return incorrect results on 32-bit platforms. (Bug #20259578)
- If a view was processed using the MERGE algorithm and had an ORDER BY clause, an error occurred if the view was queried using `GROUP BY` with the `ONLY_FULL_GROUP_BY` SQL mode enabled, unless the query selected all view columns. (Bug #20210742)
- For debug builds, the optimizer could reject use of LooseScan for `eq_ref` access joins and raise an assertion. The optimizer now permits this combination for query execution. (Bug #20119743)
- An out-of-range error in a subquery could raise an assertion. (Bug #20035071)
- Renaming the `mysql.procs_priv` table and executing `SHOW GRANTS` resulted in a server exit. (Bug #20006361)
- Ordering by a `GROUP_CONCAT()` result could cause a server exit. (Bug #19880368, Bug #20730220)
- The server could exit due to an inappropriate full-text lookup using a full-text predicate within a subquery that contained an outer reference. (Bug #19828320)
- For a prepared statement with an `ORDER BY` that refers by column number to a `GROUP_CONCAT()` expression that has an outer reference, repeated statement execution could cause a server exit. (Bug #19814337)
- For a materialized internal temporary table used with semi-joins, the optimizer could add an index to it but then use an inappropriate lookup strategy, causing a server exit. (Bug #19695490, Bug #21782943)
- The optimizer could raise an assertion due to incorrectly associating an incorrect field with a temporary table. (Bug #19612819, Bug #20730129)
- Specifying `--general_log_file=` (with an empty value) at server startup caused the server to fail and exit. (Bug #19392264)
- Improper propagation of `ORDER BY` for a derived table or view used within a multiple-table `UPDATE` could raise an assertion. (Bug #18439019)
- The `thd_proc_info()` function defined in `plugin.h` was not actually implemented. This has been changed to `set_thd_proc_info()`. (Bug #11844974)
- For debug builds, an assertion was raised when calculating the symmetric difference between a `MultiLineString` and a `MultiPoint`. (Bug #77580, Bug #21355906)
- The query rewrite framework introduced in MySQL 5.7.6 produced excesssive mutex acquisition that caused performance degradation under some conditions. (Bug #76509, Bug #20785598)
- `mysql_install_db` started `mysqld` in bootstrap mode, but failed to wait for it to finish, causing premature shutdown and the need for crash recovery. (Bug #76344, Bug #20728488)
- `SHOW CREATE USER` did not work for clients older than MySQL 5.7. (Bug #76093, Bug #20627890)
- The deprecated `IDENTIFIED BY PASSWORD` syntax is supported only for `CREATE USER` and `GRANT`, but `ALTER USER` failed to reject it. `ALTER USER` now produces a syntax error. (Bug #76048, Bug #20600865)

- Inappropriate `-Werror` options could appear in `mysql_config --cflags` output. (Bug #76019, Bug #20590904)
- With `InnoDB` as the default temporary table storage engine, `InnoDB` sometimes made incorrect assumptions about temporary table key part lengths. (Bug #76016, Bug #20590162)
- Selecting from the `global_variables` Performance Schema table resulted in a spurious warning about the `sql_log_bin` system variable. (Bug #75980, Bug #20575529)
- For the embedded server, proper deprecation warning were not produced for `SHOW VARIABLES` and `SHOW STATUS` statements that included a `WHERE` clause. (Bug #75951, Bug #20559828)
- A subquery that contained a user-defined variable could cause an assertion to be raised. (Bug #75934, Bug #20554585)
- `SET PASSWORD ... = PASSWORD('auth_string')` syntax was to be deprecated in MySQL 5.7.6, but was made illegal. This syntax is now available again, but generates a warning due to its deprecated status. These alternatives remain available, the first of which now should be considered the preferred form:

  ```
  ALTER USER ... IDENTIFIED BY 'auth_string';
  SET PASSWORD ... = 'auth_string';
  ```

  (Bug #75927, Bug #20552143)
- AddressSanitizer compilation errors were silenced. (Bug #75739, Bug #20459338, Bug #75740, Bug #20459363)
- Corrections were made for a number of code issues that resulted in compiler warnings about array bounds, possibly uninitialized variables, and variables being set but not used. (Bug #75735, Bug #20458574)
- The `mysql` client could exit prematurely when invoked with the `--quick` option. (Bug #74182, Bug #19723750)
- `CHECK TABLE ... FOR UPGRADE` did not report temporal columns that use the old datetime format (from before MySQL 5.6.4). Consequently, `mysql_upgrade` did not know to issue `REPAIR TABLE` statements to rebuild tables that contain such columns, and subsequent `ALTER TABLE` statements were unable to perform fast alterations to the extent possible had the tables been repaired. Now, if the `avoid_temporal_upgrade` system variable is disabled, `CHECK TABLE` reports old temporal columns and `REPAIR TABLE` upgrades tables from old temporal format to the new format. (Bug #73008, Bug #18985579)
- With the `offline_mode` system variable enabled, the server sometimes failed to accept connection from a user with the `SUPER` privilege due to a race condition. (Bug #72760, Bug #18842228)
- Information written to the slow query log for `HANDLER ... READ` statements always had `rows_sent` and `rows_examined` values of 0. (Bug #71892, Bug #18335504)
- `mysql_real_connect()` could close a file descriptor twice if the server was not running. (Bug #69423, Bug #19226740)
- Some key descriptors used by the optimizer were uninitialized. Thanks to Sergei Glushchenko for the patch. (Bug #68713, Bug #16512701)
- `EXPLAIN` could show incorrect `filtered` values for queries that included a `LIMIT` clause. (Bug #34124, Bug #11747810)

Changes in MySQL 5.7.6 (2015-03-09, Milestone 16)

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

- **Incompatible Change:** The `CREATE USER` and `ALTER USER` statements have additional account-management capabilities. Together, they now can be used to fully establish or modify authentication, SSL, and resource-limit properties, as well as manage password expiration and account locking and unlocking. For example, `ALTER USER` can assign passwords, and it can modify the authentication plugin for users, with no need for direct manipulation of the `mysql.user` table. For details, see CREATE USER Syntax, and ALTER USER Syntax.

  Account locking control is a new feature that permits administrators to completely disable an account from being used to connect to the server. Account locking state is recorded in the `account_locked` column of the `mysql.user` table. See User Account Locking.

  A new statement, `SHOW CREATE USER`, shows the `CREATE USER` statement that creates the named user. The accompanying `Com_show_create_user` status variable indicates how many times the statement has been executed.

  A new system variable, `log_backward_compatible_user_definitions`, if enabled, causes the server to log `CREATE USER`, `ALTER USER`, and `GRANT` statements in backward-compatible (pre-5.7.6) fashion. Enabling this variable promotes compatibility for cross-version replication.

  The `authentication_string` column in the `mysql.user` table now stores credential information for all accounts. The `Password` column, previously used to store password hash values for accounts authenticated with the `mysql_native_password` and `mysql_old_password` plugins, is removed.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate the changes to the `mysql` database. `mysql_upgrade` moves `Password` column values to the `authentication_string` column and removes the `Password` column. For nonupgraded installations that have no `account_locked` column, the server treats all accounts as unlocked, and attempts to lock or unlock and account produce an error.

The preceding changes make the following features obsolete. They are now deprecated and support for them will be removed in a future MySQL release:

- Using `GRANT` to create users. Instead, use `CREATE USER`. Following this practice makes the `NO_AUTO_CREATE_USER` SQL mode immaterial for `GRANT` statements, so it too is deprecated.
- Using `GRANT` to modify account properties other than privilege assignments. This includes authentication, SSL, and resource-limit properties. Instead, establish such properties at account-creation time with `CREATE USER` or modify them afterward with `ALTER USER`.
- `IDENTIFIED BY PASSWORD 'hash_string'` syntax for `CREATE USER` and `GRANT`. Instead, use `IDENTIFIED WITH auth_plugin AS 'hash_string'` for `CREATE USER` and `ALTER USER`, where the `'hash_string'` value is in a format compatible with the named plugin.
- The `PASSWORD()` function is deprecated and should be avoided in any context. Thus, `SET PASSWORD ... = PASSWORD('auth_string')` syntax is also deprecated. `SET PASSWORD ... = 'auth_string'` syntax is not deprecated; nevertheless, `ALTER USER` is now the preferred statement for assigning passwords.

Warning

The changes in this release result in a semantic incompatibility for one `SET PASSWORD` syntax:

```
SET PASSWORD ... = 'literal string';
```

Previously, `SET PASSWORD` interpreted the string as a password hash value to be stored directly. Now, `SET PASSWORD` interprets the string as a cleartext string and hashes it appropriately for the account authentication plugin before storing it.

Note

Any application that uses `PASSWORD()` to create hash values (a practice that has been discouraged for some time) should be modified to use a different hash-generation method. For suggestions, see the description of `PASSWORD()` in Encryption and Compression Functions.

- The `old_passwords` system variable. Account authentication plugins can no longer be left unspecified in the `mysql.user` table, so any statement that assigns a password from a cleartext string can unambiguously determine the hashing method to use on the string before storing it in the `mysql.user` table. This renders `old_passwords` superflous.

Note

It is a known issue in this release that the following `SET PASSWORD` syntax produces an error:

```
SET PASSWORD ... = PASSWORD('auth_string');
```

That syntax was to be deprecated, not removed. It will be restored in the next release, but generate a warning due to its deprecated status. These alternatives are available, the first of which is the preferred form:

```
ALTER USER ... IDENTIFIED BY 'auth_string';
SET PASSWORD ... = 'auth_string';
```

The change in `mysql.user` table structure has compatibility implications for upgrading and downgrading:

- You can perform a binary (in-place) upgrade to MySQL 5.7.6 or later and run `mysql_upgrade` to migrate the `Password` column contents to the `authentication_string` column.
- If you plan to upgrade by loading a `mysqldump` dump file from an older (pre-5.7.6) MySQL installation, you must observe these conditions for the `mysqldump` command used to generate the file:
  - You must include the `--add-drop-table` option
  - You must not include the `--flush-privileges` option

  Load the pre-5.7.6 dump file into the 5.7.6 server before running `mysql_upgrade`.
- Because the `Password` column is gone in 5.7.6 and up, downgrading to a version older than 5.7.6 requires a special procedure. See Changes Affecting Downgrades from MySQL 5.7.

References: See also: Bug #67449, Bug #14845612.

Configuration Notes

- `mysqld` now supports a `--daemonize` option that causes it to run as a traditional, forking daemon. This permits the server to work with operating systems that use systemd for process control. Advantages include automatic restarts after failure, handling of the user and group used to run the daemon, resource control, and temporary-file cleanup.

  The new `WITH_SYSTEMD` `CMake` option, when enabled, causes installation of systemd support files. In addition, scripts such as `mysqld_safe` and the System V initialization script are not installed. On platforms where systemd is not available, enabling `WITH_SYSTEMD` results in an error from `CMake`. When `WITH_SYSTEMD` is enabled, the new `SYSTEMD_SERVICE_NAME` and `SYSTEMD_PID_DIR` options may also be used to specify the MySQL service name and directory in which the server creates the PID file, respectively.

  Support files for systemd are installed when you install MySQL using an RPM distribution for these Linux platforms:

  - Red Hat Enterprise Linux 7; Oracle Linux 7; CentOS 7
  - SUSE Linux Enterprise Server 12
  - Fedora 20, 21

  You also obtain systemd support by installing from a source distribution that is configured with the `-DWITH_SYSTEMD=1` `CMake` option.

To provide better information to management processes, the server now returns one of the exit codes described in the following list. The phrase in parentheses indicates the action taken by systemd in response to the code.

- 0 = successful termination (no restart done)
- 1 = unsuccessful termination (no restart done)
- 2 = unsuccessful termination (restart done)

Note

Any management script written for older servers should be revised to handle three exit values if it checks only for 1 as a failure exit value.

For more information, see Managing MySQL Server with systemd. That section also includes information about specifying options previously specified in `[mysqld_safe]` option groups. Because `mysqld_safe` is not installed when systemd is used, such options must be specified another way.

Generated Columns

- MySQL now supports the specification of generated columns in `CREATE TABLE` and `ALTER TABLE` statements. Values of a generated column are computed from an expression included in the column definition. Generated columns can be virtual (computed “on the fly” when rows are read) or stored (computed when rows are inserted or updated). The `INFORMATION_SCHEMA.COLUMNS` table shows information about generated columns.

  Uses for generated columns include simplifying queries when applications select from a table using a complex expression, simulating functional indexes, or substituting for views. For more information, see CREATE TABLE and Generated Columns.

Installation Notes

- The `mysqld` server and `mysql_upgrade` utility have been modified to make binary (in-place) upgrades from MySQL 5.6 easier without requiring the server to be started with special options. The server checks whether the system tables are from a MySQL version older than 5.7 (that is, whether the `mysql.user` table has a `Password` column). If so, it permits connections by users who have an empty authentication plugin in their `mysql.user` account row, as long as they have a `Password` value that is empty (no password) or a valid native (41-character) password hash.

  This means that you can connect as `root` and upgrade your system tables even with an older `mysql.user` table for which `root` has no authentication plugin named. In particular, you can run `mysql_upgrade`, connecting as `root`, with no need to start the server with any special options. Previously, if the `root` account had an empty plugin value, a procedure involving starting the server with `--skip-grant-tables` and multiple restarts was required. The procedure now is simpler.

  1. Stop the old (MySQL 5.6) server
  2. Upgrade the MySQL binaries in place (replace the old binaries with the new ones)
  3. Start the MySQL 5.7 server normally (no special options)
  4. Run `mysql_upgrade` to upgrade the system tables
  5. Restart the MySQL 5.7 server

- Previously, for a new MySQL installation on Unix and Unix-like systems, initialization of the data directory (including the tables in the `mysql` system database) was done using `mysql_install_db`. On Windows, MySQL distributions included a data directory with prebuilt tables in the `mysql` database.

  `mysql_install_db` functionality now has been integrated into the MySQL server, `mysqld`. To use this capability to initialize a MySQL installation, if you previously invoked `mysql_install_db` manually, invoke `mysqld` with the `--initialize` or `--initialize-insecure` option, depending on whether you want the server to generate a random password for the initial `'root'@'localhost'` account.

  As a result of this change, `mysql_install_db` is deprecated, as is the special `--bootstrap` option that `mysql_install_db` passes to `mysqld`. These will be removed in a future MySQL release. Also, the `$HOME/.mysql_secret` file written by `mysql_install_db` is no longer needed. If it is present on your system, you can remove it.

  Initializing a MySQL installation using `mysqld` works on all platforms, including Windows. In particular, it is possible to initialize a Windows installation without the set of prebuilt tables for the `mysql` database. (However, it is unnecessary to do so for this release because Windows distributions still include the pre-built tables.)

  For more information, see Initializing the Data Directory Manually Using mysqld.

Optimizer Notes

- To handle a derived table (subquery in the `FROM` clause) or view reference, the optimizer can materialize the derived table or view reference to an internal temporary table or merge it into the outer query block. Previously, derived tables were always materialized, whereas equivalent view references were sometimes materialized and sometimes merged. This inconsistent treatment of equivalent queries could lead to performance problems: Unnecessary derived table materialization takes time and prevents the optimizer from pushing down conditions to derived tables.

  The optimizer now handles derived tables in consistent fashion; that is, the same as view references. This better avoids unnecessary materialization and enables use of pushed-down conditions that produce more efficient execution plans. For an example, see Optimizing Subqueries with Subquery Materialization.

  The optimizer also better handles propagation of an `ORDER BY` clause in a derived table or view reference to the outer query block, doing so only when this makes sense. Previously, the optimizer always propagated `ORDER BY`, even if it was irrelevant or resulted in an invalid query.

  For statements such as `DELETE` or `UPDATE` that modify tables, using the merge strategy for a derived table that previously was materialized can result in an `ER_UPDATE_TABLE_USED` error:

```
mysql> DELETE FROM t1
    -> WHERE id IN (SELECT id
    ->              FROM (SELECT t1.id
    ->                    FROM t1 INNER JOIN t2 USING (id)
    ->                    WHERE t2.status = 0) AS t);
ERROR 1093 (HY000): You can't specify target table 't1'
for update in FROM clause
```

  The error occurs when merging a derived table into the outer query block results in a statement that both selects from and modifies a table. (Materialization does not cause the problem because, in effect, it converts the derived table to a separate table.) To avoid this error, disable the `derived_merge` flag of the `optimizer_switch` system variable before executing the statement:

```
mysql> SET optimizer_switch = 'derived_merge=off';
```

The `derived_merge` flag controls whether the optimizer attempts to merge derived tables and view references into the outer query block, assuming that no other rule prevents merging. By default, the flag is `on` to enable merging. Setting the flag to `off` prevents merging and avoids the error just described. (Other workarounds include using `SELECT DISTINCT` or `LIMIT` in the subquery, although these are not as explicit in their effect on materialization.) If an `ER_UPDATE_TABLE_USED` error occurs for a view reference that uses an expression equivalent to the subquery, adding `ALGORITHM=TEMPTABLE` to the view definition prevents merging and takes precedence over the current `derived_merge` value.

For more information, see Optimizing Derived Tables and View References.

References: See also: Bug #20073366, Bug #59203, Bug #11766159.

Packaging Notes

- Packaging scripts such as those included in RPM or Debian packages have been modified per the principle that files installed from MySQL distributions should have the most restrictive permissions possible. In the following description, assume that the account used to administer MySQL has owner (user) = `mysql`, group = `mysql`.
  - Installers that create the `mysql` account do so with a shell of `/bin/false` to prevent direct login to the account.
  - The data directory and its contents are owned by and accessible only to owner/group `mysql`/`mysql`, with permissions of 750 for directories, 755 for executable files, 640 for other files.
  - Others files (including executables and libraries) have owner/group of `root`/`root`, with these permissions:
    - Executables: 755
    - Man pages, character set files, header files, test suite files: 644
    - Library files: Conventions appropriate for the host system

  Packaging scripts that perform the preceding actions for installation also perform them for upgrades, with the exceptions that if the `mysql` account exists, it is left unchanged, and if the data directory exists, its permissions and ownership are left unchanged.
- For Windows, the noinstall Zip archive was split into two separate Zip archives. The MySQL test suite, MySQL benchmark suite, and debugging binaries/information components (including PDB files) were moved into their own Zip archive named `mysql-`*`VERSION`*`-winx64-debug-test.zip` for 64-bit and `mysql-`*`VERSION`*`-win32-debug-test.zip` for 32-bit. This change was made to reduce the file size of the more common download.

Performance Schema Notes

- The Performance Schema incorporates these changes:
  - The Performance Schema now allocates memory incrementally, scaling its memory use to actual server load, instead of allocating all the memory it needs during server startup. Consequently, configuration of the Performance Schema is easier; most sizing parameters need not be set at all. A server that handles a very low load will consume less memory without requiring explicit configuration to do so.

    These system variables are autoscaled:

```
performance_schema_accounts_size
performance_schema_hosts_size
performance_schema_max_cond_instances
performance_schema_max_file_instances
performance_schema_max_index_stat
performance_schema_max_metadata_locks
performance_schema_max_mutex_instances
performance_schema_max_prepared_statements_instances
performance_schema_max_program_instances
performance_schema_max_rwlock_instances
performance_schema_max_socket_instances
performance_schema_max_table_handles
performance_schema_max_table_instances
performance_schema_max_table_lock_stat
performance_schema_max_thread_instances
performance_schema_users_size
```

There are new instruments named with the prefix `memory/performance_schema/` that expose how much memory is allocated for internal buffers in the Performance Schema. These instruments are displayed in the `memory_summary_global_by_event_name` table.

For more information about how Performance Schema allocates memory and how to assess the amount currently in use, see The Performance Schema Memory-Allocation Model.

- Instrumentation for table indexes and table locks is more flexible and less memory intensive. For a table for which index and table lock instrumentation is disabled (as specified in the `setup_objects` table), the Performance Schema allocates no memory for statistics collection. For a table for which index and table lock instrumentation is enabled, memory allocation for statistics collection is deferred until the table begins to be used.

  Configuration for instrumentation of table indexes and table locks now is exposed explicitly:

  - The `performance_schema_max_table_lock_stat` and `performance_schema_max_index_stat` system variables configure how many indexes per table and how many table locks are subject to statistics collection. These variables are autoscaling by default, reducing memory allocation for MySQL installations where the default allocation was greater than necessary for a server's workload (for example, when databases contain large numbers of tables). They can be set at startup to place explicit limits on memory allocation.

  - The `Performance_schema_table_lock_stat_lost` and `Performance_schema_index_stat_lost` status variables enable assessing whether the corresponding system variable settings are so low as to result in loss of instrumentation.

- System and status variable information is now available in Performance Schema tables:

  - System variables: `global_variables`, `session_variables`, and `variables_by_thread` contain individual system variable values.

  - Status variables: `global_status`, `session_status`, and `status_by_thread`, contain individual status variable values. `status_by_account`, `status_by_host`, and `status_by_user` contain session status variable values aggregated per account, host name, and user name.

  These Performance Schema tables contain information similar to that available from the `SHOW VARIABLES` and `SHOW STATUS` statements and the `GLOBAL_VARIABLES`, `SESSION_VARIABLES`, `GLOBAL_STATUS`, and `SESSION_STATUS INFORMATION_SCHEMA` tables.

  The Performance Schema tables offer these advantages:

- By using the `variables_by_thread` and `status_by_thread` tables, it is possible to obtain session variables for any session, not just the current session. It is also possible to obtain *only* session variables, rather than a mix of session and global variables such as returned by `SHOW SESSION VARIABLES` and `SHOW SESSION STATUS`.
- Requests for global variables produce only global variables, whereas for the `SHOW GLOBAL STATUS` statement and the `GLOBAL_STATUS INFORMATION_SCHEMA` table, requests produce not only global variables, but also session variables that have no global counterpart.
- Access to the Performance Schema tables requires the `SELECT` privilege, whereas the `SHOW` statements and `INFORMATION_SCHEMA` tables do not.

For more information, see Performance Schema System Variable Tables, Performance Schema Status Variable Tables, and Performance Schema Status Variable Summary Tables.

The value of the new `show_compatibility_56` system variable affects the output produced from and privileges required for system and status variable statements and tables. For details, see the description of that variable in Server System Variables.

The `INFORMATION_SCHEMA` tables now are deprecated in preference to the Performance Schema tables and will be removed in a future MySQL release. For advice on migrating away from the `INFORMATION_SCHEMA` tables to the Performance Schema tables, see Migrating to Performance Schema System and Status Variable Tables.

Note

It is a known issue in this release that the `session_variables` and `session_status` tables do not fully reflect all variable values in effect for the current session; they include no rows for global variables that have no session counterpart. This is corrected in MySQL 5.7.8.

- Previously, the Performance Schema enabled instrumentation for new foreground threads if there was a row in the `setup_actors` table that matched the thread user and host. Now, the `setup_actors` table has an `ENABLED` column that indicates whether or not to enable instrumentation for matching foreground threads. This permits instrumentation for matching threads to be disabled explicitly. For more information, see Pre-Filtering by Thread, The setup_actors Table, and The threads Table.
- Two previously hardcoded limits on SQL statement handling are now configurable:
  - The maximum number of bytes from SQL statements to display in the `SQL_TEXT` column of statement event tables, such as `events_statements_current`.
  - The number of bytes available for computing statement digests. Statement digests appear in the `DIGEST_TEXT` column of statement event tables.

  Previously, both values were fixed at 1024. It is now possible to change them at server startup using the `performance_schema_max_sql_text_length` and `max_digest_length` system variables. (The name `max_digest_length` does not begin with `performance_schema_` because statement digesting is now done at the SQL level even if the Performance Schema is disabled and is available to other aspects of server operation that could benefit from it. For example, query rewrite plugins now make use of statement digests, even if the Performance Schema is disabled.)

  The defaults remain at 1024, but the values can be reduced to use less memory or increased to permit longer statements to be distinguished for display and digesting purposes. Each variable has a range from 0 to 1024 × 1024.

Any bytes in excess of `performance_schema_max_sql_text_length` are discarded and do not appear in the `SQL_TEXT` column. Statements differing only after that many initial bytes are indistinguishable in this column.

Any bytes in excess of `max_digest_length` during digest computation do not factor into digest values. Statements differing only after that many bytes of parsed statement tokens produce the same digest and are aggregated for digest statistics.

For applications that generate very long statements that differ only at the end, the ability to change `max_digest_length` variables enables computation of digests that distinguish statements that previously were aggregated to the same digest. Conversely, administrators can devote less server memory to digest storage by reducing the values of this variable. Administrators should keep in mind that larger values result in correspondingly increased memory requirements, particularly for workloads that involve large numbers of simultaneous sessions. (`max_digest_length` bytes are allocated per session.)

For more information, see Performance Schema Statement Digests.

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.

References: See also: Bug #71057, Bug #68514, Bug #16414081.

Plugin Notes

- MySQL Server now supports query rewrite plugins:
  - A preparse query rewrite plugin enables rewriting of SQL statements arriving at the server before the server processes them. The plugin receives a statement string and may return a different string.
  - A postparse query rewrite plugin enables statement rewriting based on parse trees. The server parses each statement and passes its parse tree to the plugin, which may traverse the tree. The plugin can return the original tree to the server for further processing, or construct a different tree and return that instead.

  MySQL distributions now include a postparse query rewrite plugin named `Rewriter`. This plugin is rule based. You can add rows to its rules table to cause `SELECT` statement rewriting.

  One application of query rewrite plugins is to enable use of newer server capabilities with applications that might not be subject to modification. For example, the plugin can add optimizer hints to statements produced by an older application that cannot be changed.

  For more information, see Query Rewrite Plugins, and The Rewriter Query Rewrite Plugin.

Security Notes

- **Security Fix:** Due to the LogJam issue (https://weakdh.org/), OpenSSL has changed the Diffie-Hellman key length parameters for openssl-1.0.1n and up. OpenSSL has provided a detailed explanation at http://openssl.org/news/secadv_20150611.txt. To adopt this change in MySQL, the key length used in `vio/viosslfactories.c` for creating Diffie-Hellman keys has been increased from 512 to 2,048 bits. (Bug #77275, Bug #21221862, Bug #18367167, Bug #21307471, Bug #21449838)
- The `secure_file_priv` system variable is used to limit the effect of data import and export operations. The following changes have been made to how the server handles this variable:

- The server checks the value of `secure_file_priv` at startup and writes a warning to the error log if the value is insecure. The setting is considered insecure if `secure_file_priv` has an empty value, or the value is the data directory or a subdirectory of it, or a directory that is accessible by all users. If `secure_file_priv` is set to a nonexistent path, the server writes an error message to the error log and exits.
- `secure_file_priv` can be set to `NULL` to disable all import and export operations.
- Previously, the `secure_file_priv` system variable was empty by default. Now the default value is platform specific and depends on the value of the `INSTALL_LAYOUT` `CMake` option, as shown in the following table.

`INSTALL_LAYOUT` Value	**Default `secure_file_priv` Value**
`STANDALONE`, `WIN`	empty
`DEB`, `RPM`, `SLES`, `SVR4`	`/var/lib/mysql-files`
Otherwise	`mysql-files` under the `CMAKE_INSTALL_PREFIX` value

- To specify the default `secure_file_priv` value explicitly if you are building from source, use the new `INSTALL_SECURE_FILE_PRIVDIR` `CMake` option.

References: See also: Bug #18140348.

- To make it easier to produce the files required to support secure connections using SSL and secure password exchange using RSA over unencrypted connections, MySQL distributions now include the `mysql_ssl_rsa_setup` utility. This utiliy uses the `openssl` command, so its use is contingent on having OpenSSL installed on your machine. When invoked, `mysql_ssl_rsa_setup` checks the data directory for SSL and RSA files and uses `openssl` to create them if they are missing. For more information, see `mysql_ssl_rsa_setup` — Create SSL/RSA Files.

  Autodiscovery of key and certificate files in the data directory at startup now applies to servers compiled using yaSSL. Previously, this applied only to servers compiled using OpenSSL. See Configuring MySQL to Use Secure Connections.

  If the server automatically enables secure connections, it writes a message to the error log. If the server finds that the CA certificate is self-signed, it writes a warning to the error log. (The certificate will be self-signed if created automatically by the server or manually using `mysql_ssl_rsa_setup`.)

Spatial Data Support

- Spatial functions now allocate memory in larger chunks to reduce number of allocation calls and reduce overhead. (Bug #20073459, Bug #74949)
- A new set of spatial convenience functions is available:
  - `ST_Distance_Sphere()`: Returns the mimimum spherical distance in meters between two points and/or multipoints on a sphere.
  - `ST_IsValid()`: Checks whether a geometry is valid.
  - `ST_MakeEnvelope()`: Returns the rectangle that forms the envelope around two points.
  - `ST_Simplify()`: Returns a simplified geometry.
  - `ST_Validate()`: Returns a validated geometry, or `NULL` if it is invalid.

For details, see Spatial Convenience Functions.

- The spatial function namespace is being made more consistent, with the ultimate goal that each spatial function name begins with `ST_` if it performs an exact operation, or with `MBR` if it performs an operation based on minimum bounding rectangles.

  Currently, some functions have two implementations and up to three related names: A name with an `ST_` prefix, and a name with an `MBR` prefix, and a name with no prefix:

  - The name with an `ST_` prefix performs an exact operation.
  - The name with an `MBR` prefix performs an operation based on minimum bounding rectangles.
  - The name with neither prefix sometimes is an alias for the `ST_` name (as with `Area()` and `ST_Area()`), sometimes an alias for the `MBR` name (as with `Contains()` and `MBRContains()`),

  This release implements the following changes in spatial function naming:

  - A function with an `ST_` prefix is added for each non-`MBR` function that has no `ST_` name.
  - Each function that does not begin with `ST_` or `MBR` is deprecated.
  - The exceptions are the geometry object construction functions, which remain unchanged: `Point()`, `LineString()`, `Polygon()`, `MultiPoint()`, `MultiLineString()`, `MultiPolygon()`, and `GeometryCollection()`.
  - These functions are deprecated in favor of the `MBR` names: `Contains()`, `Disjoint()`, `Equals()`, `Intersects()`, `Overlaps()`, `Within()`.
  - These functions are deprecated 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()`.
  - These `ST_` names are added and are preferred over the corresponding non-`ST_` names, which now are deprecated: `ST_MLineFromText()`, `ST_MLineFromWKB()`, `ST_MPointFromText()`, `ST_MPointFromWKB()`, `ST_MPolyFromText()`, `ST_MPolyFromWKB()`, `ST_MultiLineStringFromText()`, `ST_MultiLineStringFromWKB()`, `ST_MultiPointFromText()`, `ST_MultiPointFromWKB()`, `ST_MultiPolygonFromText()`, `ST_MultiPolygonFromWKB()`.
  - `ST_GeomCollFromTxt()` is added as a synonym for `ST_GeomCollFromText()`.

  - `ST_Length()` is added to implement the same operation as the now-deprecated `GLength()`, which has a nonstandard name because a function named `Length()` already existed (to calculate string lengths).

  Use of deprecated functions produces a warning. The deprecated functions will be removed in a future MySQL release.

- Functions for checking spatial relations now use functionality available in Boost.Geometry 1.56 and up: `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`. Specifically, these functions now:

  - Are able to test the relationship between all pairs of argument types handled by Boost.Geometry.

  - Raise an exception for invalid argument types when the previous implementation may not have.

  This work also corrected issues that `ST_Overlaps()` returned 1 and `ST_Intersects()` returned 0 for two polygons that shared only a boundary, and that `ST_Intersects()` sometimes incorrectly calculated the result for intersections of `LineString` and `Polygon`.

  References: See also: Bug #68091, Bug #16174580, Bug #71076, Bug #17894858.

- GIS code now uses `Boost.Geometry.Rtree` to improve handling of geometry collection arguments in binary GIS functions. The minimum bounding rectangles (MBRs) of geometry collection components are used to set up an rtree index, which is used to search for possible matching components using each MBR of the components of the other geometry collection. The results from the rtree index search are provided to precise computation algorithms to avoid unnecessary (and much more expensive) precise computation. As a result, the time complexity of handling geometry collection arguments in GIS algorithms is reduced from O(N^2) to O(NlogN).

- These changes have been made for spatial functions that operate on minimum bounding rectangles (MBRs) of geometry values:

  - Two new functions test the covering relationship of two geometries using their MBRs. `MBRCovers()` indicates whether the MBR of one geometry covers that of another. `MBRCoveredBy()` tests the opposite, indicating whether the MBR of one geometry is covered by that of another.

  - The spatial function `MBREquals()` has been implemented. It should be used in preference to `MBREqual()`, which is now deprecated. (The new name is more consistent with the similar `Equals()` function.)

  - `MBRTouches()` now correctly uses the geometry MBRs, not the geometries themselves.

  - `MBRTouches()` and `MBRWithin()` better conform to the Open Geospatial Consortium specification: When a point lies on the boundary of a line segment or polygon, or when a line segment lies totally on the boundary of a polygon, it is not considered “within” but instead “touches.” Also, two identical points are not considered “touches.”

Functionality Added or Changed

- **Incompatible Change; InnoDB:** The `MERGE_THRESHOLD` value for index pages is now configurable using a `COMMENT` clause with `CREATE TABLE`, `ALTER TABLE`, and `CREATE INDEX` statements. If the page-full percentage for an index page falls below the `MERGE_THRESHOLD` value when a row is deleted or when a row is shortened by an `UPDATE` operation, `InnoDB` attempts to merge the index page with a neighboring index page. The default `MERGE_THRESHOLD` value is 50, which is the previously hard-coded value.

This feature adds a `MERGE_THRESHOLD` column to the internal `SYS_INDEXES` table. `SYS_INDEXES` records for tables that were originally created in 5.7.5 or earlier do not include this column. Rebuilding or importing these tables after upgrading to MySQL 5.7.6 or later properly updates `SYS_INDEXES` records with the new `MERGE_THRESHOLD` column. Otherwise, `TRUNCATE TABLE` or `ALTER TABLE ... RENAME INDEX` operations followed by a server restart or a table reload causes errors. This bug is fixed in MySQL 5.7.8 (Bug #20882432).

For more information, see Configuring the Merge Threshold for Index Pages.

- **Incompatible Change:** A new C API function, `mysql_real_escape_string_quote()`, has been implemented as a replacement for `mysql_real_escape_string()` because the latter function can fail to properly encode characters when the `NO_BACKSLASH_ESCAPES` SQL mode is enabled. In this case, `mysql_real_escape_string()` cannot escape quote characters except by doubling them, and to do this properly, it must know more information about the quoting context than is available. `mysql_real_escape_string_quote()` takes an extra argument for specifying the quoting context. For usage details, see mysql_real_escape_string_quote().

  

  Note

  Applications should be modified to use `mysql_real_escape_string_quote()`, instead of `mysql_real_escape_string()`, which now fails and produces an `CR_INSECURE_API_ERR` error if `NO_BACKSLASH_ESCAPES` is enabled.

  References: See also: Bug #19211994.

- **InnoDB:** All remaining code related to the `innodb_file_io_threads` system variable, which was removed in MySQL 5.5, was removed from the source code. (Bug #19843885)

- **InnoDB:** `InnoDB` system tablespace data is now exposed in the `INNODB_SYS_TABLESPACES` and `INNODB_SYS_DATAFILES` Information Schema tables.

- **InnoDB:** To modularize and decouple the partitioning engine from the server code base, partitioning operations in the storage engine handler class were moved to a new `partition_handler` base class, which is now the interface for partitioning-specific storage engine functionality.

- **InnoDB:** `InnoDB` now supports the creation of general tablespaces using `CREATE TABLESPACE` syntax.

  ```
  CREATE TABLESPACE `tablespace_name`
    ADD DATAFILE 'file_name.ibd'
    [FILE_BLOCK_SIZE = n]
  ```

  General tablespaces can be created outside of the MySQL data directory, are capable of holding multiple tables, and support tables of all row formats.

  Tables are added to a general tablespace using `CREATE TABLE` *`tbl_name`* `... TABLESPACE [=]` *`tablespace_name`* or `ALTER TABLE` *`tbl_name`* `TABLESPACE [=]` *`tablespace_name`* syntax.

  For more information, see CREATE TABLESPACE Syntax.

- **InnoDB:** `InnoDB` now supports native partitioning. Previously, `InnoDB` relied on the `ha_partition` handler, which creates a handler object for each partition. With native partitioning, a partitioned `InnoDB` table uses a single partition-aware handler object. This enhancement reduces the amount of memory required for partitioned `InnoDB` tables.

  The following changes accompany `InnoDB` native partitioning support:

- Partition definition (`.par`) files are no longer created. Partition definitions are stored in the internal data dictionary.
- For partitioned `InnoDB` tables, `FLUSH TABLES` does not reset the “next” `AUTO_INCREMENT` value. Instead, the next `AUTO_INCREMENT` value is kept and used after the `FLUSH TABLES` operation. If the highest `AUTO_INCREMENT` value is deleted before a `FLUSH TABLES` operation, it is not reused afterwards.
- Minor changes to statistics could result in changed execution plans.
- The minimum number of rows estimated for a partitioned `InnoDB` table is 1 instead of 2.
- The minimum number of rows estimated for range read on a partitioned `InnoDB` table index is 0 per partition instead of 1.
- Instead of only including the largest partitions when calculating matching rows in an index range, all partitions in the read set (after pruning is completed) are included. As a result, statistics for matching index rows are more accurate, but time spent during the Optimizer phase may increase for tables with numerous partitions.

- **InnoDB:** The following buffer pool flushing-related enhancements are included in MySQL 5.7.6:
  - The adaptive flushing algorithm flushes all pages at the end of the flush list if there is a high distribution of pages associated with the oldest LSN.
  - Once redo space reaches 30% full, a pre-scan on buffer pool instances determines the oldest modified pages in each buffer pool instance. Based on this information, the adaptive flushing algorithm determines the number of pages to flush from each buffer pool instance during a single flush pass. This approach helps ensure that the oldest modified pages are flushed first.
  - On Linux platforms where it is possible and where the `mysqld` execution user is authorized, the `setpriority()` system call is used to give `page_cleaner` threads priority over other MySQL/InnoDB threads to help page flushing keep pace with the current workload. `mysqld` execution user authorization can be configured in `/etc/security/limits.conf`. Refer to your Linux operating system documentation for more information.
  - When the oldest modification LSN is close to the defined maximum (`max_modified_age_sync`), a synchronous preflush of buffer pool pages is initiated which may result in a “flush wait” scenario for user threads. To smooth throughput, user threads are only required to wait for a target LSN to be reached instead of waiting for an entire flushing batch to finish. User thread waits are reported as sync flush waits by the `buffer_flush_sync_waits` metric of the `INFORMATION_SCHEMA.INNODB_METRICS` table.
  - A block was added to prevent the log write mechanism from overwriting last checkpoint LSN.
  - A message is printed to the server error log if the `innodb_io_capacity_max` setting is too high.
  - New metrics for monitoring `page_cleaner` thread activity were added to the `INNODB_METRICS` table:
    - `buffer_flush_adaptive_avg_pass`: Number of adaptive flushes passed during the recent Avg period.
    - `buffer_flush_adaptive_avg_time_est`: Estimated time (ms) spent for adaptive flushing recently.

- `buffer_flush_adaptive_avg_time_slot`: Avg time (ms) spent for adaptive flushing recently per slot.
- `buffer_flush_adaptive_avg_time_thread`: Avg time (ms) spent for adaptive flushing recently per thread.
- `buffer_flush_avg_pass`: Number of flushes passed during the recent Avg period.
- `buffer_flush_avg_time`: Avg time (ms) spent for flushing recently.
- `buffer_flush_n_to_flush_by_age`: Number of pages targeted by LSN Age for flushing.
- `buffer_LRU_batch_flush_avg_pass`: Number of LRU batch flushes passed during the recent Avg period.
- `buffer_LRU_batch_flush_avg_time_est`: Estimated time (ms) spent for LRU batch flushing recently.
- `buffer_LRU_batch_flush_avg_time_slot`: Avg time (ms) spent for LRU batch flushing recently per slot.
- `buffer_LRU_batch_flush_avg_time_thread`: Avg time (ms) spent for LRU batch flushing recently per thread.
- `buffer_LRU_get_free_loops`: Total loops in LRU get free.
- `buffer_LRU_get_free_waits`: Total sleep waits in LRU get free.

- **InnoDB:** The Performance Schema now instruments stage events for monitoring `InnoDB` `ALTER TABLE` and buffer pool load operations. The new stage events include:
  - `stage/innodb/alter table (read PK and internal sort)`
  - `stage/innodb/alter table (merge sort)`
  - `stage/innodb/alter table (insert)`
  - `stage/innodb/alter table (flush)`
  - `stage/innodb/alter table (log apply index)`
  - `stage/innodb/alter table (log apply table)`
  - `stage/innodb/alter table (end)`
  - `stage/innodb/buffer pool load`

  For more information, see InnoDB Integration with MySQL Performance Schema, and Saving and Restoring the Buffer Pool State.

- **InnoDB:** Replication-related support was added to `InnoDB` which enables prioritization of slave applier transactions over other transactions in deadlock scenarios. This transaction prioritization mechanism is reserved for future use.

- **InnoDB:** `CHECK TABLE` functionality was enhanced for `InnoDB` `SPATIAL` indexes. Previously, `CHECK TABLE` only performed minimal checks on `InnoDB` `SPATIAL` indexes. Enhanced functionality includes an R-tree validity check and a check to ensure that the R-tree row count matches the clustered index.

- **InnoDB:** The default setting for the `internal_tmp_disk_storage_engine` option, which defines the storage engine the server uses for on-disk internal temporary tables, is now `INNODB`. With this change, the Optimizer uses the `InnoDB` storage engine instead of `MyISAM` for internal temporary tables. For related information, see Internal Temporary Table Use in MySQL.

- **InnoDB:** `InnoDB` now supports 32KB and 64KB page sizes. For both page sizes, the maximum record size is 16KB. `ROW_FORMAT=COMPRESSED` is not supported when `innodb_page_size` is set to 32k or 64k. For `innodb_page_size=32k`, extent size is 2MB. For `innodb_page_size=64k`, extent size is 4MB. The `innodb_log_buffer_size` default value was increased from 8M to 16M to support the new page sizes. `innodb_log_buffer_size` should be at least 16M when using a 32k or 64k page size.

- **InnoDB:** To support future development, the code that initializes, validates and handles tablespace and table flags was refactored. Also, the `fil_create_ibd_tablespace` function was refactored, and some functions and variables related to single tablespaces were renamed.

- **Replication:** The variable `binlogging_impossible_mode` has been renamed `binlog_error_action`. `binlogging_impossible_mode` is now deprecated. (Bug #19507567)

- **Replication:** When using InnoDB with binary logging enabled, concurrent transactions written in the InnoDB redo log are now grouped together before synchronizing to disk when `innodb_flush_log_at_trx_commit` is set to 1, which reduces the amount of synchronization operations. This can lead to improved performance. (Bug #19424075)

- **Replication:** Added a number of features and enhancements relating to Performance Schema and group replication. These additions are listed here:

  - The `replication_group_members` and `replication_group_member_stats` Performance Schema tables.

  - The `START GROUP_REPLICATION` and `STOP GROUP_REPLICATION` SQL statements.

  - A `GROUP_NAME` column to the `replication_connection_status` table.

  - The `transaction_write_set_extraction` system variable.

  - A number of errors and error mesages.

- **Replication:** There is now a `Previous_gtids` event in every binary log, regardless of the value of `gtid_mode`. In previous versions, it was only generated when `gtid_mode=on`. Similarly, there is now an `Anonymous_gtid` event before every transaction when `gtid_mode=off`. These changes ensure that similar per-transaction events are generated regardless of the type of binary logging in use. As well as enabling the newly added ability to change `gtid_mode` online, this also has a positive impact on the recovery of `gtid_purged` and `gtid_executed`.

- **Replication:** A new more general purpose parallelization algorithm is now used when `slave_parallel_type=LOGICAL_CLOCK`, replacing the previous algorithm that was limited to transactions on different databases. This improves throughput when transactions on the master do not depend on each other. Now even two concurrent transactions on a master can execute in parallel on a slave, if they hold all of their locks on the master. Additionally, transaction dependency is now tracked on the slave through extra fields added to replication transactions in the binary log.

- **Replication:** It is now possible to change replication mode without having to shut down the server or synchronize the topology. As part of this feature, the following changes have been made:

  - The variable `gtid_mode` is now dynamic. It can be set by `SUPER` from a top-level statement. The states `OFF_PERMISSIVE` and `ON_PERMISSIVE` have been added.

- The variable `enforce_gtid_consistency` is now dynamic. It can be set by `SUPER` from a top-level statement.
- The status variable `Ongoing_anonymous_transaction_count` has been introduced. This shows the number of ongoing transactions which have been marked as anonymous.
- The status variables `Ongoing_anonymous_gtid_violating_transaction_count` and `Ongoing_automatic_gtid_violating_transaction_count` have been introduced in debug-enabled builds. They are not available in non-debug builds. These variables count the number of ongoing transactions that violate GTID consistency, which use `gtid_next=ANONYMOUS` and `gtid_next=AUTOMATIC`, respectively.

For more information, see Changing Replication Modes on Online Servers

- **Replication:** MySQL Multi-Source Replication adds the ability to replicate from multiple masters to a slave. MySQL Multi-Source Replication topologies can be used to back up multiple servers to a single server, to merge table shards, and consolidate data from multiple servers to a single server. See MySQL Multi-Source Replication.

  As part of MySQL Multi-Source Replication, replication channels have been added. Replication channels enable a slave to open multiple connections to replicate from, with each channel being a connection to a master. See Replication Channels.

- Undocumented functions in the C client library are now hidden. This helps minimize namespace pollution, and permits linking for applications that require functions both from yaSSL (in the client library) and from OpenSSL. (Bug #20476596, Bug #18427840)
- `CMake` support was updated to handle `CMake` version 3.1. (Bug #20344207)
- Previously, debug builds on Windows were built with `/Ob0`, which disables function inlining. Builds now use `/Ob1` to enable inlining. The new `WIN_DEBUG_NO_INLINE` `CMake` option can be used to control inlining. The default value is `OFF` (inlining enabled); if set to `ON`, inlining is disabled. (Bug #20316320)
- yaSSL was upgraded to version 2.3.7. (Bug #19695101, Bug #20201864)
- The new `-DWITH_UBSAN=ON` CMake option enables the Undefined Behavior Sanitizer. This feature is supported by GCC 4.9 and up, and Clang 3.4 and up. (Bug #19587393)
- The valid date range of the SSL certificates in `mysql-test/std_data` has been extended to the year 2029. (Bug #18366947)
- Overhead was reduced for queries such as tested by the sysbench “order-by-range” test. (Bug #75390, Bug #20296891)
- The `mysql` client program now supports `\C` in the `prompt` command to signify the current connection identifier. Thanks to Tsubasa Tanaka for the patch. (Bug #75242, Bug #20227145)
- The server now includes its version number when it writes the initial “starting” message to the error log, to make it easier to tell which server instance error log output applies to. This value is the same as that available from the `version` system variable. (Bug #74917, Bug #20052694)
- The required version of the Boost library for server builds has been raised from 1.55.0 to 1.57.0. (Bug #74666, Bug #19940297, Bug #73432, Bug #19320102)
- Previously, the `auth_socket` authentication plugin checked the socket user name only against the MySQL user name specified by the client program to the server. Now, if those names do not match, the plugin also checks whether the socket user name matches the name specified in the

`authentication_string` column of the `mysql.user` table row. The plugin permits the connection for a match in either case. Thanks to Daniël van Eeden for the patch. (Bug #74586, Bug #20041925)

- The `libmysqlclient` version number has been incremented to 20.0.0. (Bug #74206, Bug #19729266)
- A new `CMake` option, `WITH_MSCRT_DEBUG`, is available to control Visual Studio CRT memory leak tracing. The default is `OFF`. (Bug #73064, Bug #19031370)
- Beginning with MySQL 5.7.2, the server disables at startup any account that has no authentication plugin. The server now writes a more extensive message to the error log in this case to indicate how to reenable such accounts. (Bug #73026, Bug #19011337)
- `ALTER TABLE` did not take advantage of fast alterations that might otherwise apply to the operation to be performed, if the table contained temporal columns found to be in pre-5.6.4 format (`TIME`, `DATETIME`, and `TIMESTAMP` columns without support for fractional seconds precision). Instead, it upgraded the table by rebuilding it. Two new system variables enable control over upgrading such columns and provide information about them:

  - `avoid_temporal_upgrade` controls whether `ALTER TABLE` implicitly upgrades temporal columns found to be in pre-5.6.4 format. This variable is disabled by default. Enabling it causes `ALTER TABLE` not to rebuild temporal columns and thereby be able to take advantage of possible fast alterations.
  - `show_old_temporals` controls whether `SHOW CREATE TABLE` output includes comments to flag temporal columns found to be in pre-5.6.4 format. Output for the `COLUMN_TYPE` column of the `INFORMATION_SCHEMA.COLUMNS` table is affected similarly. This variable is disabled by default.

  Both variables are deprecated and will be removed in a future MySQL release. (Bug #72997, Bug #18985760)
- The minimum value of the `stored_program_cache` system variable has been changed from 256 to 16, to enable configuration of a smaller amount of memory devoted to the stored program cache. (Bug #72451, Bug #18661573)
- The code in `my_strnxfrm_simple()` was suboptimal and was improved. Thanks to Alexey Kopytov for the patch. (Bug #68476, Bug #16403708)
- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.
- The metadata locking subsystem (see Metadata Locking) has been extended to cover concurrent access to tablespaces. This includes DDL statements that explicitly affect tablespaces: `ALTER TABLESPACE`, `CREATE TABLESPACE`, and `DROP TABLESPACE`. It also includes DDL statements that affect tablespace contents: `ALTER TABLE`, `CREATE INDEX`, `CREATE TABLE`, `DROP INDEX`, `DROP TABLE`, `LOCK TABLES`, `RENAME TABLE`, and `TRUNCATE TABLE`.
- For queries that combine `ORDER BY` with `LIMIT`, the optimizer may switch to an index that applies to the `ORDER BY`. In some cases, the decision to switch was based on a heuristic rather than on cost. The optimizer now uniformly makes the decision whether to switch on a cost basis. This should result in better performanance when switching would cause a query to read an entire index or a large part of it to find qualifying rows.

  References: See also: Bug #78993, Bug #22108385, Bug #73837, Bug #19579507, Bug #16522053.
- Server and client errors are numbered in ranges beginning from 1000 and 2000, respectively. However, server error numbers are approaching 2000, leading to a potential conflict with client error numbers. To deal with this, server error numbers for MySQL 5.7 now have a range beginning with 3000. This

is implemented by permitting multiple `start-error-number N` lines in `sql/share/errmsg-utf8.txt`, with each such line resetting the numbering to *`N`*.

- Refactoring within the optimizer resulted in the following improvements to `EXPLAIN` output:

  - Output that showed `ORDER BY col_name` for implicitly grouped queries no longer does so.

  - Output for `INSERT` statements involving partition pruning now shows only the partitions actually used, not all partitions in the table.

  - Output for `UPDATE`, `INSERT`, or `DELETE` statements no longer shows “Using join buffer” in cases when join buffering was not used.

  In addition, for killed queries where the previously returned error was “Unknown error”, the error is now “Query execution was interrupted”.

  References: See also: Bug #70553, Bug #17575172.

- MySQL now provides a built-in ngram full-text parser plugin that supports Chinese, Japanese, and Korean (CJK), and an installable MeCab full-text parser plugin for Japanese. The parser plugins can be used with `InnoDB` and `MyISAM` tables.

  The built-in MySQL full-text parser uses the white space between words as a delimiter to determine where words begin and end, which is a limitation of the built-in MySQL full-text parser for ideographic languages that do not use word delimiters. The addition of ngram and MeCab full-text parser plugins address this limitation.

  For more information see ngram Full-Text Parser, and MeCab Full-Text Parser Plugin.

- The `plugin` and `servers` tables in the `mysql` system database now are `InnoDB` (transactional) tables. Previously, these were `MyISAM` (nontransactional) tables.

  In consequence of this change, `INSTALL PLUGIN` and `UNINSTALL PLUGIN` are now included among the statements that cause an implicit commit (see Statements That Cause an Implicit Commit).

  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` database.

- The following items are deprecated and will be removed in a future MySQL release. Where alternatives are shown, applications should be updated to use them.

  - The `ENCRYPT()`, `DES_ENCRYPT()`, and `DES_DECRYPT()` functions. Consider using `AES_ENCRYPT()` and `AES_DECRYPT()` instead.

  - The `sync_frm` system variable. This variable will be removed when `.frm` files become obsolete.

  - The global `character_set_database` and `collation_database` system variables are deprecated and will be removed in a future version of MySQL.

    Assigning a value to the session `character_set_database` and `collation_database` system variables is deprecated and assignments produce a warning. The session variables will become read only in a future version of MySQL and assignments will produce an error. It will remain possible to access the session variables to determine the database character set and collation for the default database.

  - Conversion of pre-MySQL 5.1 database names containing special characters to 5.1 format with the addition of a `#mysql50#` prefix. (For information about these conversions, see Mapping of Identifiers

to File Names.) Because such conversions now are deprecated, the `--fix-db-names` and `--fix-table-names` options for `mysqlcheck` and the `UPGRADE DATA DIRECTORY NAME` clause for the `ALTER DATABASE` statement are also deprecated.

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.

- The variable `session_track_gtids` was added, which enables a tracker that captures GTIDs and appends them to the OK packet.

Bugs Fixed

- **Incompatible Change; InnoDB:** If your system contains tablespace data files created prior to MySQL 5.1, read this note carefully before upgrading to MySQL 5.7.6.

  Tablespace data files created prior to MySQL 5.1 sometimes contain garbage `FIL_PAGE_TYPE` values in some pages. To address this issue, a new file page type constant (`FIL_PAGE_TYPE_UNKNOWN`) was added, and unknown `FIL_PAGE_TYPE` values are now reset to `FIL_PAGE_TYPE_UNKNOWN`.

  However, you may still encounter a failure when a non-index page that contains an invalid `FIL_PAGE_INDEX` or `FIL_PAGE_RTREE` value in the `FIL_PAGE_TYPE` field is written to disk. You can address the failure by modifying the data file to replace the invalid `FIL_PAGE_TYPE` values with `FIL_PAGE_UNKNOWN`. The error log message provides the tablespace ID and page number of the page with the invalid `FIL_PAGE_TYPE` value.

  Before MySQL 5.6, `InnoDB` page size is always 16384 bytes. `FIL_PAGE_TYPE` is defined as 24. So, if page number P of a tablespace data file is affected, the data at byte offset 16384*P+24 should contain the bytes `0x45 0xbf` (`FIL_PAGE_INDEX`) or `0x45 0xbe` (`FIL_PAGE_RTREE`). Replace these bytes with `0x00 0x0d` (`FIL_PAGE_UNKNOWN`).

  If the page contains the strings `infimum` and `supremum` at byte offset 99 or 101 (75 or 77 bytes after the start of the `FIL_PAGE_TYPE`), the page could be an index page, and there may be actual corruption.

  

  Warning

  Do not attempt to modify data files directly unless you know exactly what you are doing and fully understand the implications. Manually modifying the data file is no longer required in MySQL 5.7.7 with the fix for Bug #20691930.

  If `FIL_PAGE_TYPE` is reset to `FIL_PAGE_TYPE_UNKNOWN` in your pre-MySQL 5.5 data file, you may encounter a page corruption error when restarting the server. The error is due to a `FIL_PAGE_TYPE` field in the `InnoDB` page checksum that is still set to the previous `FIL_PAGE_TYPE` value. To address the error, rewrite the page checksum using the `innochecksum` tool. The patch for Bug #20691930 in MySQL 5.7.7 addresses this problem by recomputing the page checksum after resetting the `FIL_PAGE_TYPE` value. (Bug #17345513, Bug #17332603, Bug #19658698)

- **InnoDB:** Opening and closing of optimized temporary tables caused a negative table reference count in `InnoDB` Monitor output. (Bug #20608113)

- **InnoDB:** A duplicate database page corruption error message was removed from `buf0buf.cc`. (Bug #20605167)

- **InnoDB:** The `NAME` column of the `INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO` table was incorrectly declared as 192 characters wide. The correct length is 64 characters. (Bug #20512578)

- **InnoDB:** A full-text phrase search returned an incorrect result. An empty string was handled incorrectly when tokenizing a newly inserted row. (Bug #20465273, Bug #75755)
- **InnoDB:** A workaround introduced in MySQL 5.7.0, in the patch for Bug #14658648, was removed. The workaround allowed MySQL to disable the query cache during crash recovery. Inconsistent data could be produced during crash recovery if MySQL crashed while XA transactions were in a PREPARED state with the query cache enabled. The bug was fixed in MySQL 5.7.2 by the patch for Bug #16593427. (Bug #20461632)
- **InnoDB:** In debug builds, assertion code related to buffer pool resizing caused a significant increase in Valgrind testing time. (Bug #20461123)
- **InnoDB:** The use of change buffering for a spatial index raised an assertion. Change buffer flags should not be set for spatial indexes. (Bug #20452564, Bug #75718)
- **InnoDB:** On `ALTER TABLE ... IMPORT TABLESPACE`, there was a missing dictionary unlock call on Out-Of-Memory(OOM) that could result in a failure when allocating memory for a `.ibd` file path string. (Bug #20430105)
- **InnoDB:** An undefined reference error occurred when building MySQL with `DWITH_EXTRA_CHARSETS=none`. (Bug #20429800)
- **InnoDB:** Optimizing a `FULLTEXT` index raised an assertion. The last optimized word of a `FULLTEXT` index is stored in the `CONFIG` table `value` column which is defined as CHAR(50). An assertion was raised when the last optimized word was greater than 50 characters in length. The `CONFIG` table `value` column is defined as CHAR(200) as of MySQL 5.6.24 and MySQL 5.7.6.

  If your `innodb_ft_max_token_size` setting is greater than 50, it is recommended that you recreate existing `InnoDB` `FULLTEXT` indexes after upgrading to MySQL 5.6.24 or MySQL 5.7.6 to avoid this issue. `FULLTEXT` indexes created after upgrading to MySQL 5.6.24 or MySQL 5.7.6 are unaffected. (Bug #20418326)
- **InnoDB:** The `innodb_optimize_point_storage` option and related internal data types (`DATA_POINT` and `DATA_VAR_POINT`) were removed. (Bug #20415831)
- **InnoDB:** `fts_optimize_thread()` set a NULL `exit_event` when the server started to shut down, and before `fts_optimize_thread` was started. (Bug #20389745)
- **InnoDB:** The `memcached` `process_arithmetic_command` raised an assertion. The wrong error code was returned for a nonexistent `decr` key. (Bug #20386835)
- **InnoDB:** The expiration time (`exptime`) defined using the `memcached` `set` command was ignored. `InnoDB` `memcached` set the expiration time to an interval value instead of a system time value. (Bug #20381342, Bug #70055)
- **InnoDB:** A NaN value in the GIS-related `mbr_join_square` function raised an assertion. (Bug #20379160)
- **InnoDB:** The `innobase_close_thd` function and related wrapper functions and pointers were removed. The functions and pointers were introduced with the `InnoDB` `memcached` plugin but never used. (Bug #20369370)
- **InnoDB:** An assertion was raised when the full-text search `fts_savepoint_release()` function released a named transaction savepoint and all subsequent savepoints. Only the initial savepoint should be released. (Bug #20341916)
- **InnoDB:** Table names were displayed inconsistently in diagnostic output for `InnoDB` tables that store persistent statistics. (Bug #20330831)

- **InnoDB:** The `INFORMATION_SCHEMA.TABLES` `UPDATE_TIME` field, enabled for `InnoDB` tables in MySQL 5.7.2, was not updated for `XA COMMIT` of recovered transactions that were in `XA PREPARE` state. (Bug #20303205)
- **InnoDB:** An incorrect expression was used in `/storage/innobase/trx/trx0trx.cc`. `trx->lock.rec_pool.empty()` was used instead of `trx->lock.table_pool.empty()`. (Bug #20294158, Bug #75373)
- **InnoDB:** In `/storage/innobase/handler/ha_innodb.cc`, a `va_end()` was missing in returns that were added in MySQL 5.7.5. (Bug #20285744, Bug #75323)
- **InnoDB:** A full-text search optimization operation raised an assertion. (Bug #20281800)
- **InnoDB:** A tablespace export operation set the purge state to `PURGE_STATE_STOP`, but the purge thread did not check the purge state until the current purge operation was completed. In the case of a large history list, the tablespace export operation was delayed, waiting for the current purge operation to finish. The purge state is now checked with every purge batch. (Bug #20266847, Bug #75298)
- **InnoDB:** When a page is read from disk, there is a check for pending insert buffer entries which involves acquiring a latch on the insert buffer page. If pending entries are found, they are merged. Because the change buffer is only applicable to B-tree secondary leaf pages in non-temporary tablespaces, insert buffer merge is not necessary for all page types. Using page_type, page_level, and tablespace type information from the page that is read from disk, insert buffer merge is now skipped for non-applicable page types. (Bug #20220909)
- **InnoDB:** The name of the internal pseudo-tablespace that is created for the `InnoDB` redo log was changed from `ib_logfile101` to `innodb_redo_log`. The new name aligns with other MySQL 5.7 internal tablespace names that use an “innodb_” prefix. (Bug #20204978)
- **InnoDB:** An `ALTER TABLE ... ADD INDEX` operation raised an assertion due to assertion code that did not allow an online index status of `ONLINE_INDEX_ABORTED_DROPPED`. The assertion code was relaxed. (Bug #20198726)
- **InnoDB:** Attempting to access the table name for a table that was NULL due to a prior inconsistency caused a crash in `innobase_update_foreign_cache()` when printing an error. (Bug #20146176)
- **InnoDB:** An error occurred when the `push_warning_printf` function was invoked during server recovery. This function was previously used to print a warning message to the client. Also, `current_thd` was NULL when the server was restarted. (Bug #20144839)
- **InnoDB:** The last flushing loop on shutdown did not call `buf_flush_wait_LRU_batch_end()`, resulting in an assertion failure. (Bug #20137435)
- **InnoDB:** The `dict_index_t::auto_gen_clust_index` flag, which was used inconsistently and redundant, was removed. (Bug #20136192)
- **InnoDB:** A memory access violation in `fts_optimize_thread` caused the server to halt. A table was freed but not removed from the full-text search optimize queue. (Bug #20125560)
- **InnoDB:** An assertion was raised while updating statistics for referenced tables after a cascade update. (Bug #20125466)
- **InnoDB:** A crash occurred in `btr_cur_latch_leaves` while performing a load operation. Checking the page state without latching the page caused an inconsistency. The page state should only be checked after the page is latched. (Bug #20111105, Bug #74596)
- **InnoDB:** The `INNODB_METRICS` `adaptive_hash_searches_btree` counter failed to report counter data. (Bug #20080942, Bug #74511)

- **InnoDB:** Due to a regression introduced in MySQL 5.6.20, `mysqld stop` did not stop the `mysqld` server process while the `InnoDB memcached` plugin was active. (Bug #20078646, Bug #74956)

  References: This issue is a regression of: Bug #18409840.

- **InnoDB:** The `commit_node` member from the `tab_node_t` and `ind_node_t` query graph objects, used during table and index creation, were removed. The `commit_node` member was initialized but never used. Unused `TABLE_COMMIT_WORK` and `INDEX_COMMIT_WORK` execution steps were also removed. (Bug #20060218)

- **InnoDB:** An `ALTER TABLE ... RENAME` failure on a table with a `FULLTEXT` index raised an assertion. (Bug #20043707)

- **InnoDB:** A duplicate key error encountered during a `REPLACE` operation on a temporary table raised an assertion. (Bug #20040791)

- **InnoDB:** An `ALTER TABLE` operation that changed the name of a foreign key column resulted in a failure when reloading the foreign key constraint. The previous column name remained in the data dictionary cache instead of being evicted. (Bug #20031243)

- **InnoDB:** `ALTER TABLE` failed to check if the table is corrupted. An `ALTER TABLE` operation that affects `InnoDB` metadata should be refused if the clustered index is corrupted or the table is marked as corrupted. An `ALTER TABLE` operation should also be refused if the table is not rebuilt and a corrupted secondary index would remain after the `ALTER TABLE` operation. (Bug #20015132, Bug #74810)

- **InnoDB:** On Windows, renaming a `FULLTEXT` search file name raised an assertion. (Bug #20001827)

- **InnoDB:** A row update operation raised an assertion in `row_upd_sec_index_entry()`. In `row_merge_read_clustered_index()`, the cached spatial index was not inserted prior to the mini-transaction commit. Once the mini-transaction was committed, the clustered index page was updated or freed, resulting in the primary key fields for cached spatial index entries pointing to invalid addresses. (Bug #19999469)

- **InnoDB:** An assertion was raised in the `btr_cur_search_to_nth_level` function. Both shared locks (s-locks) and shared-exclusive locks (sx-locks) should be permitted for all latch modes. (Bug #19984494)

- **InnoDB:** An `ALTER TABLE` operation on a table with a `FULLTEXT` index raised an assertion. The table was already present in the cache and the `FULLTEXT` indexes were already initialized. When the table was reloaded, the `FULLTEXT` indexes were initialized again, causing the assertion. (Bug #19978288)

- **InnoDB:** An `UPDATE` operation on a compressed temporary table raised an assertion. Shared temporary tablespace attributes were used when extending the tablespace for a compressed temporary table. (Bug #19976331)

- **InnoDB:** Error messages regarding a size limitation on `BLOB` or `TEXT` data inserted in a single transaction were revised. (Bug #19975322)

- **InnoDB:** Server logs reported a `vector subscript out of range` error. (Bug #19955501)

- **InnoDB:** `CHECK TABLE` failed to check if the table is in a corrupt state before performing validation, resulting in an assertion. (Bug #19954054)

- **InnoDB:** To avoid I/O on tablespaces that are rarely written to, the `fsp_get_available_space_in_free_extents` function now accesses metadata from cached fields instead of the tablespace header page in the buffer pool.

  This patch also includes the following optimizations:

  - To avoid lookups, `fsp_fill_free_list()` and some other functions now take a `fil_space_t` pointer instead of a numeric tablespace identifier.
  - The `fil_extend_space_to_desired_size` function was renamed to `fil_space_extend` and its API was simplified.
  - A new method, `undo::Truncate::was_tablespace_truncated`, was added to avoid a consistency check before flushing of truncated undo tablespace files.

  (Bug #19949683)

- **InnoDB:** A failed `DROP TABLE` operation could leave a table in an inconsistent state without marking the table as corrupted. (Bug #19946781, Bug #74676)
- **InnoDB:** A wrapper class was added to improve printing of quoted SQL identifiers, such as index, column and tablespace names. (Bug #19933607)
- **InnoDB:** An `ALTER TABLE` operation raised an assertion. When a foreign key object was removed from the dictionary cache, an incorrect foreign key object was removed from the rb-tree. (Bug #19908343)

  References: This issue is a regression of: Bug #18806829.
- **InnoDB:** DML operations on a table with full-text search indexes raised an invalid assertion. (Bug #19905246)

  References: This issue is a regression of: Bug #19314480.
- **InnoDB:** A missing `DBUG_RETURN()` in `ha_innobase::update_row` raised an assertion. (Bug #19904800)
- **InnoDB:** In debug builds, setting the `innodb_limit_optimistic_insert_debug` debug configuration option to 1 caused an infinite B-tree page split. (Bug #19904003, Bug #74605)
- **InnoDB:** An `ALTER TABLE ... DROP PRIMARY KEY, ADD PRIMARY KEY` operation that changed the prefix length of the primary key field raised an assertion in the bulk insert code. (Bug #19896922)
- **InnoDB:** Some `InnoDB` diagnostic output to stderr included unnecessary line breaks and lines without a preceding timestamp. Output from multiple threads could become interleaved due to messages being written out in several non-atomic steps. (Bug #19895222)
- **InnoDB:** The `innodb_create_intrinsic` option, introduced in MySQL 5.7.5, was removed. (Bug #19893327)
- **InnoDB:** As of MySQL 5.7.5, MySQL builds depend on atomic memory access primitives being present on the target platform. To simplify the code, `HAVE_ATOMIC_BUILTINS` was removed from the `InnoDB` source in MySQL 5.7.6. `InnoDB` now depends on Microsoft atomics on Windows, and on GCC-style atomics on other platforms. (Bug #19856411)
- **InnoDB:** A severe error occurred during the log apply phase of an online `ALTER TABLE` operation that was converting a table with a UTF-8 charset to `ROW_FORMAT=REDUNDANT`. (Bug #19843246, Bug #19895661, Bug #20219871)
- **InnoDB:** A multiple-table delete operation caused the server to halt. (Bug #19815702)
- **InnoDB:** A buffer pool dump referred to a non-existing tablespace ID. (Bug #19814155)

  References: This issue is a regression of: Bug #19149177.

- **InnoDB:** The logic used to select native AIO on Windows was simplified. All Windows versions supported by MySQL 5.7 now support native AIO. The logic required to handle older Windows versions that do not support native AIO was no longer necessary. (Bug #19803939)
- **InnoDB:** In debug builds, `buf_block_align()` could be called from debug assertion code while the buffer pool is being resized, resulting in a race condition. (Bug #19803497)
- **InnoDB:** A `FLUSH TABLES` operation raised an assertion. (Bug #19803418)
- **InnoDB:** The `dict_boot()` function did not set the maximum length of columns used for index fields, resulting in `dict_index_node_ptr_max_size()` returning incorrect values. (Bug #19791849)
- **InnoDB:** When dummy tables are created, the `autoinc_mutex` member of the of the `dict_table_t` object was created unnecessarily. Similarly, the `zip_pad.mutex` object of `dict_index_t` object was created unnecessarily for dummy indexes. To avoid unnecessary mutex contention, `autoinc_mutex` and `zip_pad.mutex` objects are now allocated and initialized on the first lock attempt. (Bug #19788198, Bug #73361)
- **InnoDB:** `log_sys->mutex` was not held when reading the `fil_space_t::max_lsn` field, causing a race condition. (Bug #19729855)

  References: This issue is a regression of: Bug #18645050.
- **InnoDB:** `btr_insert_into_right_sibling()` could delete node pointers at the parent page. To avoid latch order violations and deadlocks with other threads, lock intention is now checked for leaf pages as well as upper non-leaf pages. (Bug #19729316)
- **InnoDB:** `InnoDB` performed unnecessary table lookups in the change buffer during tablespace export operations. (Bug #19724300)
- **InnoDB:** The `fil_tablespace_deleted_or_being_deleted_in_mem()` function, added in MySQL 4.1, was longer necessary and has been removed. There is a fallback check in `fil_io()` that returns `DB_TABLESPACE_DELETED`. (Bug #19719727)
- **InnoDB:** To ease future development, the `ha_innobase::create` function was refactored. (Bug #19718568)
- **InnoDB:** Redundant conditional branching and a redundant a check for `srv_read_only_mode` were removed from `ha_innobase::create()`. Redundant conditional branching and an unused local variable were removed from `ha_innobase::delete_table()`. (Bug #19712822)
- **InnoDB:** Unused code related to UTF-8 handling for `InnoDB` `FULLTEXT` indexes was removed. (Bug #19712059)
- **InnoDB:** The `fil_index_tree_is_freed()` function, which returned a false negative when the index root page was reallocated, was replaced by improved logic for freeing index trees. This patch also removed a redundant parameter that was passed to `dict_drop_index_tree()`. (Bug #19710798)
- **InnoDB:** The `InnoDB` change buffer tree, which was created inside the `InnoDB` data dictionary cache unnecessarily, is now created directly, bypassing the cache. This patch also removes the `DICT_UNIVERSAL` flag, which was set in in connection with `DICT_IBUF`. Neither of the flags is used for persistent data structures, which makes `DICT_UNIVERSAL` unnecessary. (Bug #19710650)
- **InnoDB:** The `fil_space_t::tablespace_version` field, introduced to keep track of `ALTER TABLE...DISCARD TABLESPACE` followed by `ALTER TABLE IMPORT TABLESPACE` operations, was removed. The `tablespace_version` field ensured that a change buffer merge would not occur for old buffered entries while a tablespace with the same `space_id` was imported. The field was redundant and no longer required. (Bug #19710564)

- **InnoDB:** Removed unused code related to index name lookup, and replaced a function that permitted duplicate index names. (Bug #19710348)
- **InnoDB:** Column and index names were unnecessarily escaped in `InnoDB` diagnostic messages and interfaces. This patch also adds a new function, `innobase_quote_identifier`, for quoting `FOREIGN KEY` constraints and column names in `SHOW CREATE TABLE` output. (Bug #19704286)
- **InnoDB:** When using the MySQL thread pool, connections encountered long semaphore waits during load testing. (Bug #19703758, Bug #19887285)
- **InnoDB:** Since the introduction of fast index creation in MySQL 5.1, index objects have been added to the `SYS_INDEXES` internal data dictionary table before being committed. Uncommitted entries were identified by a prefix (defined as `TEMP_INDEX_PREFIX`). `TEMP_INDEX_PREFIX` was also used in the `InnoDB` data dictionary cache, resulting in complications when displaying or comparing index names. To address this problem, a new `dict_index_t::uncommitted` flag was introduced along with accessor methods `is_committed()` and `set_committed()`. Before this change, some `InnoDB INFORMATION_SCHEMA` tables displayed uncommitted index names with a preceding question mark. The question mark prefix is now omitted. (Bug #19702328)
- **InnoDB:** `InnoDB` displayed tables names inconsistently in diagnostic messages. Some messages displayed table names using an internal representation while other messages displayed table names in a translated form. (Bug #19694618)
- **InnoDB:** For `FULLTEXT` indexes, a lookup for the `FTS_DOC_ID_INDEX` was performed during DML operations. To avoid the costly lookups, a pointer to `FTS_DOC_ID_INDEX` is now cached at DDL time. (Bug #19693488)
- **InnoDB:** To simplify code, the `is_redo_skipped` flag, introduced in MySQL 5.7.5 with the `CREATE INDEX` bulk load feature, was removed. The flag caused redo logging for page allocation to be skipped. Redo logs are now generated for page allocation, even when creating a new tablespace. (Bug #19693192)
- **InnoDB:** An `MLOG_FILE_NAME` redo log record, which provides the information necessary to identify tablespace files that changed since the last checkpoint, were emitted on log checkpoint even though there were no changes to tablespace files. If a tablespace file is missing or unreadable on crash recovery, the inconsistency should be ignored if there are no redo logs to apply. For related information, see Tablespace Discovery During Crash Recovery. (Bug #19685095)
- **InnoDB:** An unused parameter, `archive_space_id`, that was passed and ignored in the `log_group_init` function, was removed. (Bug #19669129)

  References: See also: Bug #16296837.
- **InnoDB:** In read-only mode, a GIS data search using the `MBRCONTAINS()` function raised an assertion. (Bug #19664678)
- **InnoDB:** Page reservation for the index tree was not performed before calling `btr_page_alloc()`. (Bug #19660261)
- **InnoDB:** Building MySQL 5.7.5 on a Debian 7 32-bit system with GCC resulted in a MySQL server failure. The problem was due to a GCC bug (Debian Bug Report #764220) that causes incorrect code to be emitted when a function that takes a pointer or reference as a parameter is declared as `attribute((const))` or `attribute((pure))`. The problem is known to occur on Debian Wheezy 7.6 x86 with g++-4.6 (Debian 4.6.3-14) 4.6.3 or g++ (Debian 4.7.2-5) 4.7.2, and on Debian Jessie/Sid amd64 with gcc (Debian 4.9.1-15) 4.9.1 or g++ (Debian 4.9.1-15) 4.9.1. The bug may exist in other gcc-4.x versions as well any GCC version that accepts the `attribute((const))` or `attribute((pure))` code.

To avoid the bug, problematic attributes have been removed from MySQL functions that take pointers or references that they are dereferencing.

This patch also removed instances of `attribute((nonnull))`, which do not always generate a warning when NULL is passed, and may not emit code for handling the NULL case. (Bug #19632776)

- **InnoDB:** A rollback operation raised an assertion in `lock_rec_free_all_from_discard_page_low()` due to stale records locks on empty pages that were being removed from an index tree. (Bug #19628598)
- **InnoDB:** The `modify_clock` value is now stored to allow the `buf_page_optimistic_get()` function, used to get optimistic access to a database page, to succeed in most cases. An unnecessary `PAGE_HEAP_TOP` (record heap top pointer) and `FIL_PAGE_TYPE` (file page type) set was removed from `btr0bulk.cc`. (Bug #19611367)
- **InnoDB:** The `dict_set_corrupted()` function attempted to update the clustered index of the `SYS_INDEXES` data dictionary table incorrectly. (Bug #19584379)
- **InnoDB:** Compiling with the new Clang 3.5 release resulted in a number of `InnoDB` compilation warnings. (Bug #19579603)
- **InnoDB:** Removed unused API definitions from `api0api.h` and `api0api.cc` source files. (Bug #19579149)
- **InnoDB:** The `DICT_TF2_USE_FILE_PER_TABLE` flag should be tested by the `dict_table_use_file_per_table` function to verify that the table uses a file-per-table tablespace. (Bug #19578222)
- **InnoDB:** With `innodb_create_intrinsic` enabled, temporary tables created during `ALTER TABLE` operations were marked as optimized temporary tables, resulting in an assertion. Enabling `innodb_create_intrinsic` should only affect `CREATE TABLE` and `CREATE INDEX` operations. (Bug #19565749)
- **InnoDB:** Valgrind testing returned a `Conditional jump or move depends on uninitialised value(s) at buf_page_is_zeroes` error. The unread portion of the page contained garbage values. (Bug #19536534)
- **InnoDB:** An `INSERT` operation on a spatial index resulted in a crash in `split_rtree_node()`. The `mbr_join_square` function failed to check for infinity and NaN (not a number) values. (Bug #19533996, Bug #73776)
- **InnoDB:** With change buffering enabled, a buffered sequence of operations that should not have been buffered resulted in an `Unable to purge a record` error. (Bug #19528825, Bug #73767)
- **InnoDB:** Pages with a checksum value of zero were incorrectly treated as empty pages. A page should only be considered empty if its checksum value and LSN field values are zero. (Bug #19500258, Bug #73689)

  References: This issue is a regression of: Bug #17335427.
- **InnoDB:** The C-style function, `ib_logf()`, used for writing log messages, has been removed in favor of C++ style classes (`ib::info`, `ib::warn`, `ib::error`, and `ib::fatal`). (Bug #19495721)
- **InnoDB:** The `InnoDB` data dictionary was not updated when a `ALTER TABLE ... CHANGE COLUMN` operation changed the case of the column name. (Bug #19465984)
- **InnoDB:** `InnoDB` returned a `table not found` error for a missing tablespace file. (Bug #19419026)

- **InnoDB:** `InnoDB` shutdown stalled due to a user thread that was in a waiting state. (Bug #19386426)
- **InnoDB:** After upgrading to MySQL 5.7, an `ALTER TABLE` operation on a tables created in MySQL 5.6 and containing GIS data would cause a serious error. (Bug #19368904)
- **InnoDB:** After an online `ALTER TABLE ... ADD INDEX` operation, crash recovery failed due to a regression introduced with the `CREATE INDEX` bulk insert enhancement introduced in MySQL 5.7.5. (Bug #19316315, Bug #19308426)
- **InnoDB:** On non-Windows platforms, `os-file_pread` and `os_file_pwrite` functions return -1 when an error occurs. This value was printed in an error message as the number of bytes read or written. Instead of printing the -1 value in the error message, a separate error message indicating a system call failure is now printed. Thanks to David Bennett for the patch. (Bug #19315210, Bug #73365)
- **InnoDB:** A memory access violation caused `fts_optimize_thread` and `mysqld` to terminate. (Bug #19314480)
- **InnoDB:** A procedure, called from a function to perform an operation on a temporary table, caused the server to halt. (Bug #19306524)
- **InnoDB:** Attempting to shut down the server after starting the server with `innodb_force_recovery=6` resulted in a hang. (Bug #19265668, Bug #73341)
- **InnoDB:** The `fil_inc_pending_ops()` and `fil_decr_pending_ops()` functions have been replaced by `fil_space_acquire()` and `fil_space_release()`. This change removes a space ID lookup. The new functions are implemented in `buf_load()`, `fsp_get_available_space_in_free_extents()`, and `lock_rec_block_validate()`, which is a debug function. The patch for this bug also removed `fil_tablespace_is_being_deleted()`, which was an orphaned function. (Bug #19149177)
- **InnoDB:** If the log sequence number (LSN) has not increased, the `log_write_up_to()` function should not initiate redo log writing. (Bug #19068569, Bug #73109)
- **InnoDB:** A `CREATE TABLE` operation failed with a `table is full` error when running a MySQL server with `innodb_flush_method=O_DIRECT` on a Linux system with an ext3 file system. The error was due to an internal `posix_fallocate()` failure that occurs when `O_DIRECT` is specified. To allow the file operation to proceed, the internal `posix_fallocate()` failure now prints an error message to the error log. (Bug #18903979)
- **InnoDB:** As part of a cleanup of `InnoDB` `INSERT` code paths, assertion code was added to `ha_innobase::end_stmt()` and other places at the start of DDL. Debug code was added to `row_log_table_apply()`. Assertion code was added to optimized temporary table-related functions, and unused parameters were removed. (Bug #18894337)

  References: This issue is a regression of: Bug #11758237.
- **InnoDB:** A full-text search operation caused a segmentation fault. (Bug #18778259)
- **InnoDB:** Enhancements introduced in MySQL 5.7.5 related to tablespace discovery during crash recovery caused a performance regression. (Bug #18645050)
- **InnoDB:** If a database is named using uppercase letters on a MySQL server with `lower_case_table_names=2` (which is default on Mac OS X), `InnoDB` stores the database name as specified in the `InnoDB` internal system table (`SYS_TABLES`) but stores the name in lowercase on disk. During crash recovery, the case mismatch resulted in a conflict that marked the tablespace `.ibd` file as missing. The patch for this bug converts database names to lowercase on crash recovery. (Bug #18412598, Bug #72043)

- **InnoDB:** A full-text query expansion search using a search phrase plus wildcard operator resulted in `InnoDB: Did not find word ... for query expansion search` errors. This patch also addressed an issue related to full-text indexes being “unsynced” by DDL rollback. (Bug #18229097, Bug #19831736)

  References: This issue is a regression of: Bug #17373659.

- **InnoDB:** In debug builds, the `InnoDB` Lock Monitor asserted after a `DROP TABLE` operation, and the `InnoDB` Monitor encountered an assertion in `buf_page_get_gen`. (Bug #18062698, Bug #71343, Bug #18173184, Bug #68116)

- **InnoDB:** A `CREATE TABLE` operation that failed when `innodb_strict_mode` was enabled succeeded without printing a warning when `innodb_strict_mode` was disabled. (Bug #17852083)

- **InnoDB:** `buf_LRU_free_page()` would call `buf_page_set_sticky(bpage)`, needlessly making removed pages sticky in some cases. (Bug #17407091, Bug #70228)

- **InnoDB:** A slow shutdown (`innodb_fast_shutdown=0`) after crash recovery raised an assertion. Slow shutdown did not wait for background rollback operations to finish before proceeding. (Bug #16862810)

- **InnoDB:** The criteria used to define a small tablespace was inconsistent. Thanks to Laurynas Biveinis for the patch. (Bug #16696906, Bug #68970)

- **InnoDB:** For explicit cache coherency, a write barrier was added to the head of `os_thread_create_func()`, and a read barrier was added to assertion code in `rw_lock_free_func()`. (Bug #13364876, Bug #62692, Bug #18870970, Bug #72809)

- **InnoDB:** A `memcached append` operation on an `INT` column caused a segmentation fault. `append` operations on `INT` columns are not supported and are now blocked. (Bug #75200, Bug #20209756)

- **InnoDB:** The integer column value was handled incorrectly for the `memcached incr` and `decr` commands. (Bug #69415, Bug #20083106, Bug #74874, Bug #20044123)

- **Partitioning:** When multiple columns are used in `KEY` partitioning, their order may help determine the partition in which the row is placed. Changing this order by means of an `ALTER TABLE` that uses `ALGORITHM=INPLACE` can lead to inconsistency when placing rows in partitions; in other words, a row inserted before such an operation operation is placed in one partition, but the same row inserted afterwards is placed in a different one. For this reason, altering the order of a multicolumn index online is no longer allowed when that index is also used as the base for partitioning the table by `KEY`; instead, you must use a copying `ALTER TABLE` to perform the change. (Bug #17896265)

- **Replication:** When `enforce_gtid_consistency` was set to `WARN`, if a second GTID consistency violating statement within a transaction was encountered, it was not raising a warning. This was due to the fact that by design the transaction context was marked as GTID violating, hence no other warnings were being issued until the transaction committed. The fix ensures that a warning is raised for all statements inside a transaction correctly. (Bug #20414559)

- **Replication:** After restarting a slave, the first relay log was missing the `Previous_gtids` log event. Since MySQL version 5.7.6, a `Previous_gtids` log event is added to every log. This fix ensures that a `Previous_gtids` log event is correctly added to the first relay log. (Bug #20106390)

- **Replication:** When purging binary logs and the first left binary log contained only a `Previous_gtids` log event, a `lost_gtids->is_empty()` assertion was caused. This was related to the fix for Bug#16741603 and has now been corrected. (Bug #20075721)

- **Replication:** When using a slave configured to use a special character set such as UTF-16, UTF-32, or UCS-2, the receiver (I/O) thread failed to connect. The fix ensures that in such a situation, if a slave's character set is not supported then default to using the `latin1` character set. (Bug #19855907)

- **Replication:** If a client thread on a slave executed `FLUSH TABLES WITH READ LOCK` while the master executed a DML, executing `SHOW SLAVE STATUS` in the same client became blocked, causing a deadlock. The fix ensures that the read lock is only held during the period that the relay log is being updated and the deadlock is avoided. (Bug #19843808)

- **Replication:** When using multi-source replication with multiple channels and with a multi-threaded slave enabled, resetting the slave and then executing `RESET SLAVE ALL`, `START SLAVE` or `STOP SLAVE` resulted in a crash. This has now been fixed and the multi-threaded slave can be restarted in a multi-source replication setup. (Bug #19784641)

- **Replication:** The `CHANGE REPLICATION FILTER` statement can be used to create an empty filter, for example when clearing previously configured replication filters. This caused a crash in previous versions when creating an empty filter for `REPLICATE_DO_TABLE`, `REPLICATE_IGNORE_TABLE`, `REPLICATE_WILD_DO_TABLE`, or `REPLICATE_WILD_IGNORE_TABLE`. This fix ensures that these replication filters can be safely cleared by setting the filter to be empty. (Bug #19711674)

- **Replication:** When using a MySQL version that had been compiled with the `WITH_DEBUG` option enabled, using `expire_logs_days` to purge binary logs caused a restart to crash the server. This problem arose after the fix for Bug #17283409. The fix ensures that `current_thd` is checked before calling `DEBUG_SYNC()`. (Bug #19553099)

- **Replication:** When using a multi-threaded slave, the slave receiver (SQL) thread stopped with an `ER_MTS_CANT_PARALLEL` error when issuing a `LOAD DATA INFILE` statement that tried to load data into a non-transactional table on the master but failed, for example due to a primary key violation. This was caused by the multi-threaded slave applier incorrectly handling `DELETE_FILE` events. The fix ensures that a multi-threaded slave handles `DELETE_FILE` events correctly. (Bug #19552923)

- **Replication:** Sometimes the slave I/O thread leaves a partial group in the current relay log, for example when it is killed or stopped. After it is restarted, a new relay log is created on rotation and a pair of `ROTATE_EVENT` and `FORMAT_DESCRIPTION_EVENT` is replicated from master and written into the new relay log. When using a multi-threaded slave, problems such as error 1755 were encountered when applying the remaining part of the group in the relay log. This fix ensures that if `MASTER_AUTO_POSITION` is enabled, then the worker rolls back the partial group, finishes its work, and then applies the new complete copy of the group. If `MASTER_AUTO_POSITION` is disabled, the worker does not roll back the partial group. (Bug #19545298)

- **Replication:** Start log events were not checked by slaves for minimum size. (Bug #19145698)

- **Replication:** When using row-based replication with `slave_type_conversions` enabled, a binary log with more than one `Rows_log_event` in succession caused a crash. This was due to the temporary tables generated as part of the `slave_type_conversions` process being released too early. This fix ensures that the temporary tables are not released too early, and also ensures that long transactions do not cause an out of memory error. (Bug #18770469, Bug #19704825)

- **Replication:** When using binary log files that had been manually copied from the master, for example to avoid I/O thread reading delay, a multi-threaded slave generated error 1755. Because the `Previous_gtids` log event is logged using the master's `server_id` and not the slave's `server_id,` the previous events were not being skipped correctly. This fix ensures that the events in `Previous_gtids` log event are always skipped, regardless of whether they are from the relay log (generated on the slave) or from the binary log (generated on the master and manually copied to the slave as the relay log). (Bug #17812024)

- **Replication:** When replicating from an earlier version MySQL master, such as version 4.1, checksums are not used for events. Replicating to a slave running a newer version of MySQL, such as version 5.6, which has `slave_sql_verify_checksum` enabled by default meant that the last 4 bytes of events from the older master were being incorrectly interpreted as the checksum. A warning is now generated

and to avoid such a situation, set `slave_sql_verify_checksum=0` to disable checksums on the slave. (Bug #17276183)

- **Replication:** When using multi-source replication and a multi-threaded slave in a situation that required recovery of a channel, such as after a slave applier thread error, or after a crash, the channel was not being recovered correctly. This meant there was no attempt to fix gaps in transaction execution left by the stopped session, which led to some transactions being applied repeatedly. The fix ensures that in such a situation, the correct channel is passed through to multi-threaded slave recovery. (Bug #74906, Bug #20046222)

- **Replication:** Ignorable log events were introduced in MySQL 5.6, but were found to not be functioning correctly. This has now been fixed. (Bug #74683, Bug #19949915)

- **Replication:** When an XA transaction was active, executing an internal rollback, for example using the `BINLOG` statement, resulted in an assertion. The fix ensures that a rollback happens only for a slave when a transaction spans multiple binary log files. Rollback does not happen now if the Format_description comes from the `BINLOG` statement being executed in the MySQL client. (Bug #74597, Bug #19928622)

- **Replication:** The GTIDs of transactions committed in a group were not added to `gtid_executed` in order and this sometimes caused temporary gaps in `gtid_executed`. When these gaps occurred, the server would have to add and remove intervals from the GTID set, and this requires a mutex, which would cause contention and could reduce performance. The fix ensures that GTIDs are added to `gtid_executed` in the same commit order without gaps. (Bug #74328, Bug #19982543)

- **Replication:** When `gtid_mode=ON` and `log-bin=OFF`, committed transaction GTIDs are added to the `mysql.gtid_executed` table. In such a configuration, the committed transaction GTID was not being correctly added to `gtid_purged` until the next time the server was restarted. The fix ensures that committed transaction's GTIDs are added to `gtid_purged` at the time of commit. (Bug #74279, Bug #19781336)

- **Replication:** In a replication topology where:

  - the slave had `GTID_MODE=ON` and `MASTER_AUTO_POSITION=1`

  - the master had `GTID_MODE=ON` and had not executed any transactions since it was started

  if the slave used the `MASTER_POS_WAIT` function to wait until it had received the full binary log from the master while the master had not executed any transactions, then the `MASTER_POS_WAIT` function would never finish, or would time out. This was caused because after a server restart, the master's binary log ends with a `Previous_gtids` log event but this event was not being replicated, so the slave was not made aware of the master's binary log position. The fix ensures that the `Previous_gtids` log event is replicated correctly, so that the slave becomes aware of the correct binary log position on the master, ensuring that the `MASTER_POS_WAIT` function can finish. (Bug #73727, Bug #19507923)

- **Replication:** When restarting MySQL with `relay_log_recovery` enabled to recover from a crash, if the SQL thread had never been started, the position from which to start recovery was not correctly initialized because `Relay_Master_Log_File` was missing. This fix ensures that in such a situation each of the relay logs, starting from the first relay log file, is searched for a rotate event from the master, which specifies where replication started from. This rotate event is then used to set the SQL thread's `Relay_Master_Log_File` and `Relay_Log_Pos` and recovery continues as normal. (Bug #73039, Bug #19021091)

- **Replication:** When using GTIDs for replication and with `MASTER_AUTO_POSITION` enabled, if a slave requested GTIDs which had been already been purged by the master, the master was sending all available GTIDs. This happened because the master reads all available binary logs and searches

for a binary log which contains a GTID that is not contained in the union of `gtid_executed` and `gtid_retrieved`. If such a GTID is found, the master starts sending the information starting from that location. In a situation where the union of the slave's `gtid_executed` and `gtid_retreived` set did not contain the master's `gtid_purged` set, the slave would expect GTIDs which had already been purged by the master. This fix ensures that in such a situation, the slave's I/O thread is aborted with an error "Master has purged binary logs containing GTIDs that the slave requires.". (Bug #73032, Bug #19012085)

- **Replication:** When using a multi-threaded slave with GTID based replication, enabling `--replicate-same-server-id` caused the slave thread to stop with an error and replication could not be started. This was caused by a `Previous_gtids` log event not being correctly filtered in such a setup and reaching the worker thread. The fix ensures that `Previous_gtids` log event is correctly processed by the coordinator thread. (Bug #72988, Bug #18967791)

- **Replication:** A kernel mutex contention was being caused because `mysqlbinlog` was calling localtime() for every event read, which in turn called stat(/etc/localtime). This fix ensures that `mysqlbinlog` uses localtime_r(), which is optimized to store the read only timezone internal structure. This also means that `mysqlbinlog` now establishes the time zone at the beginning of processing and you can not change it during processing. This is the same behavior as MySQL server. (Bug #72701, Bug #18808072)

- **Replication:** In normal usage, it is not possible for a slave to have more GTIDs than the master. But in certain situations, such as after a hardware failure or incorrectly cleared `gtid_purged`, the master's binary log could be truncated. This fix ensures that in such a situation, the master now detects that the slave has transactions with GTIDs which are not on the master. An error is now generated on the slave and the I/O thread is stopped with an error. The master's dump thread is also stopped. This prevents data inconsistencies during replication. (Bug #72635, Bug #18789758)

- **Replication:** When using a GTID based replication slave with auto positioning enabled, there was a possibility that the last fully received transaction could be requested again by the slave I/O thread when the GTID of the transaction was not in the slave's `gtid_executed` set. This situation could occur for example if the SQL thread had not applied the transaction or a `RESET MASTER` statement was issued on the slave to clean up its `gtid_executed` set. The fix ensures that a GTID based replication slave using auto positioning does not ask for a fully received transaction twice, regardless of the slave's `gtid_executed` set, and it now only adds a GTID to the `Retrieved_Gtid_Set` when the whole transaction has been received. (Bug #72392, Bug #18629623, Bug #17943188)

- **Replication:** When using `SHOW SLAVE STATUS` to monitor replication performance, `Seconds_Behind_Master` sometimes displayed unexpected lag behind the master. This was caused by `Previous_gtids` log events being written to the slave's relay log with a timestamp behind the master, and then being used to calculate the `Seconds_Behind_Master`. This fix ensures that events generated on the slave that are added to the relay log and are not used when calculating `Seconds_Behind_Master`. (Bug #72376, Bug #18622657)

- **Replication:** The global scope for the `sql_log_bin` system variable has been deprecated, and this variable can now be set with session scope only. The statement `SET GLOBAL SQL_LOG_BIN` now produces an error. It remains possible for now to read the global value of `sql_log_bin`, but doing so produces a warning. You should act now to remove from your applications any dependencies on reading this value, as the ability to do so will be removed in the next major MySQL release following MySQL 5.7. (Bug #67433, Bug #15868071)

- For debug builds, an assertion could be raised during index selection if a spatial index used a column that was also part of the primary index. (Bug #20451454)

- On Linux, trying to install a `.dll` plugin (intended for Windows) resulted in a memory leak. (Bug #20439894)

- On 32-bit platforms, byte-count calculations for `utf8` arguments for `RPAD()` could overflow and cause a server exit. (Bug #20316028)
- `mysqltest` had a memory leak if another process shut down the server. (Bug #20221262)
- The `mysql_session_track_get_first()` C API function returned 1 instead of 0 even after a valid query was executed to change the session state. (Bug #20126551)
- On Ubuntu 14.10, MySQL install operations could fail to reload AppArmor. (Bug #20092641)
- For debug builds, the server could raise an assertion during `DELETE` processing due to failure to handle a subquery that was required to be a scalar subquery but returned more than 1 row. (Bug #20086791)
- `mysql_list_fields()` and `mysql_stmt_prepare()` could leak memory. This problem was introduced in MySQL 5.7.5 as a result of the change to EOF packet handling. (Bug #20065461, Bug #20065517)
- Some queries with argumentless functions, `GROUP BY`, and `ROLLUP` caused an assertion to be raised. (Bug #20034943)
- A user with a name of `event_scheduler` could view the Event Scheduler process list without the `PROCESS` privilege. (Bug #20007583, Bug #20754369)
- The `mysql` client could exit prematurely when invoked with the `--xml` option. (Bug #19974879)
- `InnoDB` table checksum calculation could yield an incorrect result if the value of the `innodb_checksum_algorithm` system variable was modified during the operation. (Bug #19931177)
- The `LIKE` operator could produce unreliable results of the `ESCAPE` clause contained an expression that was constant at execution time but unknown prior to that. (Bug #19931126)
- Execution of certain `BINLOG` statements while temporary tables were open by `HANDLER` statements could cause a server exit. (Bug #19894987, Bug #20449914)
- A malformed `mysql.proc` table row could result in a server exit for `DROP DATABASE` of the database associated with the `proc` row. (Bug #19875331)
- Binary distributions for Solaris built with Sun Studio now ship with the `stlport` library due to a dependency of client programs on that library. (Bug #19845068)
- `SHOW GRANTS` after connecting using a proxy user could display the password hash of the proxied user. (Bug #19817663)
- For debug builds, the optimizer could produce a bad index scan cost when creating a temporary table for a derived table, and raise an assertion as a result. (Bug #19793998)
- Unlocking a temporary table after locking and truncating it could cause a server exit. (Bug #19786309)
- `IN` predicates could be incorrectly flagged as candidates for semi-join flattening, causing an assertion to be raised when flattening was attempted. (Bug #19779600, Bug #18932813)
- Large values of the `transaction_prealloc_size` system variable could cause the server to allocate excessive amounts of memory. The maximum value has been adjusted down to 128K. A similar change was made for `transaction_alloc_block_size`. Transactions can still allocate more than 128K if necessary; this change reduces the amount that can be preallocated, as well as the maximum size of the incremental allocation blocks. (Bug #19770858, Bug #20730053)
- RPM and DEB packages set the default `sql_mode` value incorrectly (they did not set `ONLY_FULL_GROUP_BY`, which is now included in the default value). (Bug #19766800)

- Source RPM packages were missing the proper dependency on the Boost library. (Bug #19714453)
- A server exit could occur for queries that compared two rows using the `<=>` operator and the rows belonged to different character sets. (Bug #19699237, Bug #20730155)
- The Enterprise Encryption plugin could mishandle string arguments. (Bug #19688008, Bug #20730103)
- The optimizer detected functional dependency for equality expressions of the form *`col_name = expr`*, but not for expressions of the form *`(col_name, ...) = (col_name, ...)`*. Now it handles the latter as well. (Bug #19687724)
- Certain `InnoDB` errors caused stored function and trigger condition handlers to be ignored. (Bug #19683834, Bug #20094067)
- On some 32-bit platforms, `GET_LOCK(`*`lock_name, -1`*`)` returned immediately due to timeout rather than waiting for the lock. (Bug #19674349)
- If a DML statement containing a subquery caused a deadlock inside `InnoDB`, `InnoDB` would roll back the transaction. This would not be noticed in the SQL layer, with the result that execution continued, eventually leading to an assertion being raised inside `InnoDB`. (Bug #19670163)
- With `default_authentication_plugin` set to `sha256_password`, password hashes written to the binary log were in the wrong format. (Bug #19660998)
- `GROUP BY` or `ORDER BY` on a `CHAR(0) NOT NULL` column could lead to a server exit. (Bug #19660891)
- Under load, the server could exit while attempting to populate the `OBJECT_TYPE` column for selects from the `events_waits_current` Performance Schema table. (Bug #19658933)
- `ST_AsGeoJson()` could fail when given an illegal `max_dec_digits` or `options` argument. (Bug #19657747)
- Geohash spatial functions failed when given a geohash argument having a collation other than the default collation. (Bug #19657725)
- Checks enforced by `ONLY_FULL_GROUP_BY` cannot reliably be run if a grouped query is part of `CREATE VIEW`. The MySQL server tried to run those checks and could exit; now only statements that actually use the view run the checks. (Bug #19636980)
- For debug builds, an assertion could be incorrectly raised when a grouped query referred to a view. (Bug #19636409)
- For debug builds: Adding a unique index to a `POINT NOT NULL` column triggered a warning and the key was not promoted to a primary key. Creating a unique index on a different non-`NULL` column in the same table then raised an assertion. (Bug #19635706)
- When there is no change in session state, the OK packet sent from server to the client contained an unneeded byte at the end of the packet. (Bug #19625718)
- Debug builds of `mysql_install_db` did not compile on Solaris 11 U2 due to use of the deprecated `vfork()` function. (Bug #19603400)
- An assertion could be raised for either of these conditions: 1) A conversion to semi-join intended for scalar subqueries was applied to multiple-row subqueries. 2) An `IN` predicate for which the left-hand side was a scalar subquery converted to a semi-join was checked to see whether it could use materialization. (Bug #19586047)

- For debug builds, if an intermediate or final result produced NaN or a negative number, `ST_Distance()` caused a server exit. This function now produces a `ER_GIS_INVALID_DATA` error instead. (Bug #19584716)

- `CMake` configuration was adjusted to handle new warnings reported by Clang 3.5, using the `-Wpointer-bool-conversion` and `-Wundefined-bool-conversion` compiler options. (Bug #19584183)

- If a `CREATE TABLE` or `ALTER TABLE` partitioning statement was executed in strict SQL mode and an `ER_WRONG_TYPE_COLUMN_VALUE_ERROR` error occurred, the `sql_mode` was reset to `''` and the stack of error handlers was corrupted, leading to a server exit. (Bug #19584181)

- Attempting to start the server on a port that was already in use produced Valgrind errors. (Bug #19566148)

- Session state was not included with the results of queries saved in the query cache. (Bug #19550875)

- Illegal `CREATE TABLE` statements could fail to create the table (as expected), but still generate table statistics in the Performance Schema. (Bug #19535945)

- Setting `session_track_system_variables` to `NULL` could lead to an eventual server exit. (Bug #19514067)

- The client protocol tracing plugin did not account for the removal of the EOF packet from the client/server protocol in MySQL 5.7.5. (Bug #19512199)

- The default value for the condition filtering effect for equality conditions on nonindexed columns was adjusted from 0.005 to 0.1. The original value caused too-high estimates for the condition filtering effect for columns with low cardinality. (Bug #19505175)

- A `UNION` statement for which the first query block returned a `POINT` column and the second returned a geometric column with a non-`POINT` value failed if the query used `InnoDB` temporary tables or stored the result in an `InnoDB` table. (Bug #19471564)

- An assertion could be raised for queries evaluated using a semi-join LooseScan if an index scan was used on one index and a range scan on another index. (Bug #19465034)

- For client programs, `--secure-auth` is now deprecated and `--skip-secure=auth` is illegal, but use of `--skip-secure-auth` resulted in a warning followed by the help message rather than an error. (Bug #19438612)

- In strict SQL mode, some `SELECT` statements could execute differently within and without stored procedures. (Bug #19418619)

- If the `audit_log` plugin encountered a disk-full error, the server would exit.

  Now, if the file system to which the audit log is being written fills up, a “disk full” error is written to the error log. Audit logging continues until the audit log buffer is full. If free disk space has not been made available by the time the buffer fills, client sessions will hang, and stopping the server at the time of client sessions hanging will result in audit log corruption. To avoid this if client sessions are hung, ensure that free space is available on the audit logging file system before stopping the server. (Bug #19411485)

- With the `validate_password` plugin activated and dictionary lookups enabled, passing a user-defined variable to `PASSWORD()` could cause a server exit. (Bug #19388163)

- Statements that used Geohash spatial functions could not be prepared. (Bug #19383904)

- The XPath `number()` function failed when invoked with no argument. Now MySQL treats `number()` as if it had been invoked for the current context node (in other words, as if `number(.)` had been used instead), which is the behavior called for in the XPath specification for this case. (Bug #19323016)

  References: This issue is a regression of: Bug #19056196.

- With the query cache enabled, certain queries that began with comment sequences could cause invalid memory read errors. (Bug #19322795)

- Certain `GRANT PROXY` statements affected only in-memory privileges and were reverted by `FLUSH PRIVILEGES` or a server restart. (Bug #19309652)

- `STR_TO_DATE()` could mishandle conversion of numeric input to date, resulting in a server exit. (Bug #19047644)

- Fixed a Valgrind warning for an out-of-bounds read while parsing `'0E+'`. (Bug #19047527)

- Under certain conditions, `DATE_FORMAT()` could use the same buffer for its format argument and the function result, resulting in invalid memory reads. (Bug #19047488)

- Conversion of a string to an IPv6 address could raise a Valgrind warning. (Bug #19047425)

- For failure to create a temporary table due to being out of file descriptors, the server exited rather than returning an error. (Bug #18948649)

- `mysqldump` failed to report a disk-full error if the dump destination was located on an NFS mount. (Bug #18817867)

- Under certain conditions, a proxy user could expire the password of the proxied user. (Bug #18815349)

- Previously, `InnoDB` permitted a foreign key to be created which referenced a parent table for which the user did not have sufficient privileges. Now, the user must have the `REFERENCES` privileges for the parent table to create a foreign key. (Bug #18790730)

- The server could exit due to an optimizer failure to allocate enough memory for resolving outer references. (Bug #18782905, Bug #19892803)

- The `mysql_session_track_get_first()` and `mysql_session_track_get_next()` C API functions could cause a client crash if passed invalid arguments. (Bug #18769620)

- If two internal temporary tables were created based on the same aggregate function, the server could exit. (Bug #18766378)

- For some queries that contained a derived table (subquery in the `FROM` clause), delay of materialization resulted in a suboptimal execution plan due to a less accurate row-count estimate. (Bug #18607971)

- For some multiple-table `UPDATE` statements, the join order of the tables could incorrectly influence the result. (Bug #18449085)

- `ST_Touches()` could cause a server exit for some inputs. (Bug #18304448)

- Copying `InnoDB` tables containing full-text columns from Windows to Linux caused a server exit on Linux during full-text index initialization. (Bug #18285007, Bug #19864963, Bug #73155)

- A server running with `--default-authentication-plugin=sha256_password` rejected connection attempts by MySQL 5.1 clients requiring a password. (Bug #18160400)

- The `validate_password` plugin did not properly enforce password constraints for accounts authenticated by the `sha256_password` authentication plugin. (Bug #18140348)

- For `UPDATE` and `DELETE` statements, the server could exit after attempting to access an uninitialized data structure. (Bug #18036143)
- Execution of a prepared statement with a nested `IN` subquery and a view could cause a server exit. (Bug #17973601)
- Starting the server with `start service` or `mysqld_safe` could result in failure to use the correct plugin directory. (Bug #17619241)
- `FLUSH TABLES` on a `FEDERATED` table failed if the table had been idle longer than the `wait_timeout` time plus the TCP keepalive time. (Bug #17599258)
- For `FEDERATED` tables, `IGNORE` handling for `DELETE IGNORE` statements was ignored. (Bug #17564775)
- For debug builds, an assertion was raised for `ALTER TABLE` when accessing an indexed column for which the operation modified the column length, if the length was 767 and was being increased. (Bug #16886196)
- Selecting all columns from `INFORMATION_SCHEMA.TABLES` did not reopen tables if they were in the table cache, but selecting a subset of those columns under the same conditions did reopen tables. (Bug #16869534)
- On Windows, the `replace` utility did not work. (Bug #16581605)
- Creating a `FEDERATED` table with an `AUTO_INCREMENT` column using a `LIKE` clause results in a server exit. (Bug #12671631)
- For debug builds, a missing error check permitted certain `ALTER TABLE` statements that should fail to continue processing. (Bug #76515, Bug #20788817)
- For RPM-based installation operations, no information was produced to indicate that `mysql_install_db` wrote the initial `root` password to `$HOME/.mysql_secret`. These operations now use `mysqld --initialize`, which writes the password to the standard error output. (Bug #75859, Bug #20518217)
- For some full-text queries, incomplete optimizer cleanup regarding index use could affect subsequent queries against the same table. (Bug #75688, Bug #20442572, Bug #20261601)
- For JSON-format `EXPLAIN` output, the `filtered` value was displayed to an unwarranted number of digits precsion. This value is now limited to two digits following the decimal point. (Bug #75663, Bug #20429156)
- Pushed joins were not working for `NDB` tables. (Bug #75256, Bug #20234994)
- For a slow network connection, the timeout for downloading Boost (600 seconds) could be too short. A new `CMake` option, `DOWNLOAD_BOOST_TIMEOUT`, is now available to configure the timeout. (Bug #75238, Bug #20223893)
- For some queries with `LIMIT`, `EXPLAIN` could indicate that execution would be done using `filesort`, but execution actually was done using an index read. (Bug #75233, Bug #20219846)
- Several spelling errors in error messages and the source code were corrected. Thanks to Otto Kekäläinen for the patch. (Bug #75084, Bug #20135835)
- A bulk `INSERT` followed by other statements followed by `LOAD DATA` could produce incorrect `AUTO_INCREMENT` values. (Bug #75068, Bug #20126635)
- When `CMake` did not find the required version of Boost, the error message did not indicate the required version. Now it does. (Bug #75026, Bug #20108908)

- Enabling the `log_timestamps` system variable incorrectly required binary logging to be enabled. (Bug #75025, Bug #20108866)
- During token processing, the parser check whether a token contained 7-bit data could be applied to the wrong token. (Bug #74984, Bug #20086997)
- For a privilege error on a table underlying a view, a more general error should be supplied for attempts to access the view, so as not to provide information about the view contents. This did not happen in strict SQL mode. (Bug #74868, Bug #20032855)
- For subqueries that used `GET_LOCK()` or `RELEASE_LOCK()` in decimal context, the server could create ill-defined temporary tables, resulting in a raised assertion. (Bug #74859, Bug #20031761)
- `default_password_lifetime` was marked `volatile`, unnecessarily because it is protected with a mutex. Thanks to Stewart Smith for the patch. (Bug #74849, Bug #20029439)
- Removed the unused `grant_option` global variable from `mysqld.cc`. Thanks to Stewart Smith for the patch. (Bug #74847, Bug #20029398)
- `InnoDB` boolean full-text searches incorrectly handled `+` combined with parentheses; for example, `+word1 +(>word2 <word3)`. (Bug #74845, Bug #20028323)
- `NULL` as an expression was not recognized as a literal for calculation of Performance Schema statement digests. (Bug #74813, Bug #20015246)
- MySQL failed to compile with GCC 4.9.1 in debug mode. (Bug #74710, Bug #19974500)
- An optimizer cost model constructor allocated but did not destroy a cost constant object, resulting in a memory leak. (Bug #74590, Bug #19895764)
- Certain queries could raise an assertion when a internal string operation produced a `NULL` pointer rather than an empty string. (Bug #74500, Bug #19875294, Bug #13358486, Bug #79988, Bug #22551116)
- For `mysql_install_db`, the `--no-defaults` option was not passed to `mysqld`. (Bug #74477, Bug #19863782)
- For debug builds, the server could exit due to an optimizer failure to allocate enough memory for group references. (Bug #74447, Bug #19855522)
- For 32-bit Solaris builds, alignment problems resulting from improper use of varargs function arguments caused core dumps and incorrect output. (Bug #74395, Bug #19821617)
- For the `table_io_waits_summary_by_table` Performance Schema table, there was an off-by-one error for the `COUNT_FETCH` and `COUNT_READ` values. (Bug #74379, Bug #19814559)
- Depending on contents, geometry collection objects were not properly destroyed, resulting in a memory leak. (Bug #74371, Bug #19813931)
- Using `(row subquery1) NOT IN (row subquery2)` with `NULL` values in the left argument could cause an assertion failure. (Bug #74357, Bug #19805761)
- Any index comment specified for `ALTER TABLE ... ADD INDEX` was ignored. (Bug #74263, Bug #19779365)
- Reading a system variable with a `NULL` value inside a stored program caused any subsequent reads to return a `NULL` value even though the variable value might change across invocations of the stored program. (Bug #74244, Bug #19770958)
- Storage engine API code and functions in the `handler.h` and `handler.cc` files that are never called or referenced were removed. (Bug #74207, Bug #19729286)

- The `-DENABLED_PROFILING=0` CMake option resulted in compilation errors. (Bug #74166, Bug #19730970)
- With the change in MySQL 5.7.5 to `InnoDB` for the help tables in the `mysql` database, `mysql_install_db` became much slower for loading the help-table content. This was due to the `INSERT` statements loading with autocommit enabled. Now all the statements execute as a single transaction, not one transaction per statement. (Bug #74132, Bug #19703580)
- On CentOS 6, specifying a relative path name for the `--socket` option caused MySQL startup script failure. (Bug #74111, Bug #19775856)
- The `group_concat_max_len` system variable could be set to its maximum value at runtime, but not in an option file. (Bug #74037, Bug #19670915)
- The server incorrectly wrote client-side error messages to the error log: Deadlock found when trying to get lock; try restarting transaction. (Bug #73988, Bug #19656296)
- The client part of the `sha256_password` plugin could not be specified as a default client plugin (`--default-auth=sha256_password`) for users authenticating with other server plugins. (Bug #73981, Bug #19651687, Bug #17675203)
- Miscalculation of memory requirements for qsort operations could result in stack overflow errors in situations with a large number of concurrent server connections. (Bug #73979, Bug #19678930)
- `REPEAT()` wasted time concatenating empty strings. (Bug #73973, Bug #19646643)
- The capabililty of using `InnoDB` for temporary tables in MySQL 5.7.5 resulted in certain queries failing: Some queries involving multiple-table `UPDATE`, queries involving long `PRIMARY KEY` values, and queries involving `DISTINCT SUM()`. (Bug #73927, Bug #19627741, Bug #73932, Bug #19628808, Bug #73702, Bug #19497209)
- On Windows, setting the `max_statement_time` session variable greater than 0 resulted in a memory leak. (Bug #73897, Bug #19605472)
- In Solaris 11.2, `dtrace -V` output changed from `Sun D` to `Oracle D`, causing detection of DTrace availability to fail during MySQL configuration. (Bug #73826, Bug #19586917)
- `DROP DATABASE` failed if the database directory contained `.cfg` files (such as created by `FLUSH TABLES FOR EXPORT`). (Bug #73820, Bug #19573998)
- On 32-bit systems, `GLength()` returned a non-INF value for `LineString` values of infinite length. (Bug #73811, Bug #19566186)
- `mysql_config --libs_r` produces output containing link flags for `libmysqlclient_r`, even though that library was removed in MySQL 5.5 and replaced with a symlink to the underlying `libmysqlclient` library. The output now refers directly to `libmysqlclient`. (The implication is that it is no longer necessary to maintain the symlink for the sake of being able to use `mysql_config --libs_r`.) (Bug #73724, Bug #19506315)
- For statement digest calculation, the Performance Schema failed to recognize signed literal numbers as values representable by `?` and created multiple digests for statements that should have had the same signature. Now all instances of unary plus and unary minus followed by a number reduce to `?` in digests. (Bug #73504, Bug #19389709)
- Compilation on Windows using Visual Studio 2013 resulted in “unresolved external symbol” errors. (Bug #73461, Bug #19351573)
- `OLD_PASSWORD()` is deprecated, but no warning was produced when it was invoked. (Bug #73376, Bug #19285177)

- A server warning error message referred to the obsolete `table_cache` system variable rather than to `table_open_cache`. Thanks to Daniël van Eeden for the patch to fix some of the instances. (Bug #73373, Bug #19285052, Bug #75081, Bug #20135780)
- Certain queries for which subquery materialization or `UNION DISTINCT` was used together with a hash index on a temporary table could produce incorrect results or cause a server exit. (Bug #73368, Bug #19297190)
- If a table had a `NOT NULL` column, for an `INSERT` statement on the table for which the column value was not specified, the server produced `ERROR 1048 "Column cannot be null"` rather than `Warning 1364 "Field doesn't have a default value"` if there was a `BEFORE` trigger with an action type different from `ON INSERT`. (Bug #73207, Bug #19182009)
- The `IS_FREE_LOCK()` and `IS_USED_LOCK()` function implementations contained a race condition due to which they could access freed memory when a user lock was concurrently checked and freed. Accessing freed memory could result in an incorrect function return value or server exit. (Bug #73123, Bug #19070633)
- `SHOW EVENTS` in the `performance_schema` database returned an access-denied error, rather than an empty result as is done for `INFORMATION_SCHEMA`. Now an empty result is returned. (Bug #73082, Bug #19050141)
- `LOCK TABLES` sometimes acquired an insufficiently strong lock for implicitly locked tables. (Bug #72887, Bug #18913551)
- Sort order of output from a view could be incorrect when the view definition includes an `ORDER BY` clause but the view is selected from using a `WHERE` clause. (Bug #72734, Bug #18838002, Bug #81235, Bug #23207758)
- The server no longer logs the following warnings because they are uninformative: Client failed to provide its character set. '*`charset`*' will be used as client character set. (Bug #72543, Bug #18708334)
- The `ENABLED_LOCAL_INFILE` `CMake` option incorrectly was enabled by default. (Bug #72106, Bug #18448743)
- The server could fail to parse inserted strings for `SET` columns for which the column definition had exactly 64 elements. (Bug #71259, Bug #18020499)
- Use of ODBC-format date literals could produce incorrect query results. (Bug #69233, Bug #16812821)
- `mysql_setpermission` failed to properly quote user names in SQL statements that it generated. (Bug #66317, Bug #14486004)
- For `FEDERATED` tables, `DELETE FROM` *`tbl_name`* statements were sent to the remote server as `TRUNCATE TABLE` *`tbl_name`* statements, with possible side effects on transaction handling and `AUTO_INCREMENT` processing. (Bug #42878, Bug #11751864)
- A file created for an internal temporary table could cause problems if the file was orphaned for some reason and the file name was reused for later queries. (Bug #32917, Bug #11747548)
- `mysql_tzinfo_to_sql` failed in `STRICT_ALL_TABLES` SQL mode if time zone tables contained malformed information. (Bug #20545, Bug #11745851)

Changes in MySQL 5.7.5 (2014-09-25, Milestone 15)

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.

Compilation Notes

- **Important Change:** MySQL builds on Windows using Visual Studio now require Visual Studio 2013 or later. The previous requirement was Visual Studio 2010 or later. (Bug #18404381)

- **Important Change:** The atomic-operations API was simplified to use only the existing GCC built-in implementation or platform-provided implementations (for Windows, Solaris), and to remove the custom mutex-based fallback implementation. The retained implementations are those able to use CPU-native atomics. This simplifies the atomics APIs and related code and deals with bugs resulting from the fallback implementation.

  As part of this work, the (undocumented) `WITH_ATOMIC_LOCKS` and `MY_ATOMIC_MODE_RWLOCKS` `CMake` options were removed.

  On platforms where native atomics are supported, this change introduces no issues. For other platforms, here are potential MySQL compilation issues, and solutions:

  - 32-bit Linux variants that use GCC 4.1 will no longer work. This includes Red Hat 5, which is a supported platform. The solution to this problem is to use a new GCC or set the `-march` compiler option. For example, use GCC 4.4, which is available on Red Hat 5. For information about specifying compiler options, see Compiler Flags.

  - There may be issues on unsupported platforms. For example, 64-bit PowerPC, 32-bit ARM, and 64-bit ARM will not compile with older compilers. The solution for these cases is to use GCC 4.7 or later.

- `CMake` now checks for minimum versions of supported compilers: `gcc` 4.4 (Linux, Solaris); Sun Studio 12u2 (Solaris client library); Clang 3.3 (Mac OS X, FreeBSD). This check can be disabled with the `-DFORCE_UNSUPPORTED_COMPILER=ON` option. (Bug #19187034)

- Noisy compiler warnings on FreeBSD 10 were silenced. (Bug #18790490)

- `CMake` workarounds for older Mac OS X and XCode versions were removed. On Mac OS X, compilation always uses Clang, even for 32-bit builds.

  Compilation on Mac OS X is now supported for Mac OS 10.8 and up, using XCode 5 and up. Compilation on older versions may work but is unsupported. (Bug #18510941)

- Previously, the `MYSQL_MAINTAINER_MODE` `CMake` option was turned on by default for debug builds and off for release builds, and `MYSQL_MAINTAINER_MODE` caused `-Werror` to be enabled when building with GCC. This made it cumbersome to enable `-Werror` under certain conditions, such as when compiling with Clang.

  Now, `MYSQL_MAINTAINER_MODE` is on by default when compiling debug builds with GCC, and `MYSQL_MAINTAINER_MODE` enbles `-Werror` regardless of whether GCC or Clang is used. Enabling `-Werror` with Clang can be done simply by explicitly setting `-DMYSQL_MAINTAINER_MODE=1` when running `CMake`. In addition, some compilation warnings reported by Clang 3.4 were fixed, making it possible to build the default debug build with `-Werror`. (Bug #18313717)

- Build support was modified to produce the same warnings for Clang as for `gcc`. (Bug #17959689)

Configuration Notes

- **Incompatible Change:** `mysql_install_db` has been rewritten from Perl into C++. This enables it to be provided as an executable binary and eliminates its dependency on having Perl installed.

  The new implementation involves several other differences as well. The following items list some of the most significant changes. For more information, see `mysql_install_db` — Initialize MySQL Data Directory.

  - The executable binary version is located in the `bin` installation directory, whereas the Perl version is located in the `scripts` installation directory. For upgrades from an older version of MySQL, you may find a version in both directories. To avoid confusion, remove the version in the `scripts` directory. Applications that expect to find `mysql_install_db` in the `scripts` directory should be updated to look in the `bin` directory instead.

  - Some options are handled differently. For example, the `--datadir` option is mandatory.

  - There are several new options. For example, there are options that afford explicit control over the administrative account that is created. By default, this is `'root'@'localhost'`, but you can use `--admin-user` and `--admin-host` to change the user and host parts of the account name.

  - Several options have been removed or replaced. For example, `--skip-random-passwords` has been replaced by `--insecure`.

  - `mysql_install_db` always overwrites the `.mysql_secret` file, rather than appending to it if it exists. It is assumed that immediately after installation, you will connect to the server using the file contents and reset the administrative password before proceeding to another deployment.

  - `mysql_install_db` no longer passes unrecognized options to `mysqld`. (But you can use `--defaults-extra-file` to specify an option file to be added to the `mysqld` bootstrapping command.)

- `mysql_install_db` no longer creates a default `my.cnf` file.

DTrace Support

- MySQL now includes DTrace support on Oracle Linux 6 or higher with UEK kernel. If DTrace is present, server builds will detect it with no special `CMake` options required. For information about using DTrace on MySQL, see Tracing mysqld Using DTrace.

Error Handling

- The server was made more consistent and resilient with regard to handling of statements for which the `IGNORE` keyword is specified.

  - The server failed to report warnings for `INSERT IGNORE` statements.
  - The server could fail to report warnings for multiple-table `DELETE IGNORE` statements.
  - `UPDATE` triggers for a table were invoked even for `UPDATE IGNORE` statements for which a unique index caused the update to be ignored.
  - For debug builds, an assertion could be raised for errors occurring in `DELETE IGNORE` statements.
  - For debug builds, an assertion could be raised for deadlocks resulting from `DELETE IGNORE` statements.
  - For `DELETE IGNORE` executed on the parent table in a foreign key relationship, foreign key violation errors were treated as warnings (correct), but rows that did not produce foreign key violations were not deleted.

  The server was made more consistent and resilient with regard to handling of statements in strict SQL mode.

  - In strict SQL mode, triggers could permit operations not permitted in strict mode.
  - In strict SQL mode, deprecation warnings about duplicate indexes were incorrectly promoted to errors.
  - Strict SQL mode was not applied to multiple-table `DELETE` statements.

  For more information about `IGNORE` and strict SQL mode, see Comparison of the IGNORE Keyword and Strict SQL Mode. (Bug #6196, Bug #11744960, Bug #43895, Bug #11752648, Bug #68726, Bug #16522924, Bug #16860715, Bug #16860829, Bug #14786621, Bug #17550423, Bug #42910, Bug #11751889, Bug #16976939, Bug #18526888)

InnoDB Notes

- **Incompatible Change:** The `InnoDB` storage engine can no longer be disabled. The `--skip-innodb` option is deprecated and has no effect, and its use results in a warning. It will be removed in a future MySQL release. This also applies to its synonyms (`--innodb=OFF`, `--disable-innodb`, and so forth).

  A new `innodb_lock_no_retry` flag for the `--debug` option is now available. `--debug='d,innodb_lock_no_retry'` causes `InnoDB` to fail immediately during startup if locks cannot be acquired, rather than making 100 attempts before failing. This may be useful during testing or debugging to produce faster server exit when `InnoDB` cannot acquire its locks.

  One reason for disabling `InnoDB` is to enable starting a server instance using the same data directory as an existing instance. (`MyISAM` permits that, but `InnoDB` does not.) Because `InnoDB` can no longer be

disabled, the workaround is to stop the existing instance before starting another so that there is only one active instance using a data directory at a time.

- **Incompatible Change:** A new log record type (`MLOG_FILE_NAME`) is used to identify file-per-table tablespaces that have been modified since the last checkpoint. This enhancement simplifies tablespace discovery during crash recovery and eliminates scans on the file system prior to redo log application. For more information about the benefits of this enhancement, see Tablespace Discovery During Crash Recovery.

  This enhancement changes the redo log format, requiring that MySQL be shut down cleanly before upgrading to or downgrading from MySQL 5.7.5.

Optimizer Notes

- The optimizer computes more accurate costs for semi-join materialization. (Bug #18558561)
- Optimizer trace output for range access in the `considered_access_path` section has been improved: Instead of always printing `"access_type": "ref"` for index lookup types, `"eq_ref"`, `"ref"`, or `"fulltext"` is now printed. (Bug #18195373)
- During query execution plan construction, the optimizer now uses condition filtering to make better use of all conditions on a table in determining the estimate of qualifying rows that will be joined to the next table. For example, even though there might be an index that can be used to select rows, there might also be additional conditions in the `WHERE` clause that can further restrict the estimate for qualifying rows.

  Use of additional conditions is controlled by the `condition_fanout_filter` flag of the `optimizer_switch` system variable. This flag is on by default but can be disabled to suppress use of condition filtering (for example, for a query that is found to perform better without it).
- The optimizer now uses more exact index statistics. Currently, the improved values are used by `InnoDB`, with these effects:
  - In many cases, better execution plans result for queries for which previously a less optimal join index or table join order was chosen.
  - The row estimates in `EXPLAIN` output are more accurate, as well as the filter values in some cases.
  - Cardinality estimates in the index statistics displayed by `SHOW INDEX` are more accurate for `InnoDB` tables.
- To generate execution plans, the optimizer uses a cost model that is based on estimates of the cost of various operations that occur during query execution. The optimizer has a set of compiled-in default “cost constants” available to it to make decisions regarding execution plans.

  The optimizer now has in addition a database of cost estimates to use during execution plan construction. These estimates are stored in the `server_cost` and `engine_cost` tables in the `mysql` system database and are configurable at any time: Any non-`NULL` cost estimate stored in the cost model tables overrides the corresponding compiled-in default estimate. Any `NULL` estimate indicates to the optimizer to use the compiled-in default.

  Implementation and testing is ongoing to make it safe for DBAs to change these values. Currently, changing them should be considered at your own risk.

  There is also a new `FLUSH` variant, `FLUSH OPTIMIZER_COSTS`, that causes the server to reread the cost tables and apply any changed estimates to new sessions.

  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` database.

Performance Schema Notes

- **Incompatible Change:** The Performance Schema now provides a `user_variables_by_thread` table that exposes user-defined variables. For more information, see Performance Schema User Variable Tables.

  In consequence of this change, the server now limits user-defined variable names to a maximum of 64 characters, the length of the `VARIABLE_NAME` column in the table. Previously, the server did not enforce a limit. The new limit is similar to the limit on the lengths of many other identifiers in MYSQL (see User-Defined Variables). Queries that use very long user-defined variable names must be rewritten to user shorter names.

  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` database.

- Previously, for the wait event tables (such as `events_waits_current`), the `NUMBER_OF_BYTES` column was `NULL` for table I/O waits; that is, for events for the `wait/io/table/sql/handler` instrument. For table I/O waits, this value now indicates the number of rows processed.

  In addition, for batch I/O operations (such as row fetches for table or index scans), the Performance Schema now can report a single event for *`N`* rows, rather than reporting a single-row event *`N`* times. This change significantly reduces Performance Schema overhead for table batch I/O by reducing the number of reporting calls. The tradeoff is lesser accuracy for event timing. Rather than time for an individual row operation as in per-row reporting, timing for batch I/O includes time spent for operations such as join buffering, aggregation, and returning rows to the client.

  For more information on the conditions under which batch I/O reporting occurs, see the description of the `NUMBER_OF_BYTES` column in The events_waits_current Table.

- The Performance Schema stage event tables (`events_stages_current`, `events_stages_history`, and `events_stages_history_long`) contain two new columns that, taken together, provide a stage progress indicator for each row:

  - `WORK_COMPLETED`: The number of work units completed for the stage
  - `WORK_ESTIMATED`: The number of work units expected for the stage

  Each column is `NULL` if no progress information is provided for an instrument. Interpretation of the information, if it is available, is entirely up to the instrument implementation. Initially, to demonstrate the concept, the `stage/sql/copy to tmp table` instrument provides progress information if it is enabled. In this case, the unit for interpretation of the columns is number of rows copied.

  For more information, see Performance Schema Stage Event Tables.

  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` database.

Security Notes

- **Incompatible Change:** MySQL 5.6 deprecated passwords that used the older pre-4.1 password hashing format. Support for these passwords is now removed, which involves the following changes. Applications that use any feature no longer supported must be modified.

- The server-side `mysql_old_password` authentication plugin is removed. Accounts that use this plugin are disabled at startup and the server writes an “unknown plugin” message to the error log. For instructions on upgrading accounts that use this plugin, see Migrating Away from Pre-4.1 Password Hashing and the mysql_old_password Plugin.

  The client-side `mysql_old_password` authentication plugin is removed from the C client library.

- The `--secure-auth` option to the server and client programs is the default, but is now a no-op. It is deprecated and will be removed in a future MySQL release.

- The `--skip-secure-auth` option to the server and client programs is no longer supported and using it produces an error.

- The `secure_auth` system variable permits only a value of 1; a value of 0 is no longer permitted.

- For the `old_passwords` system variable, a value of 1 (produce pre-4.1 hashes) is no longer permitted.

- The `OLD_PASSWORD()` function is removed.

- The `mysqladmin old-password` command is removed.

- MySQL now includes a server-side authentication plugin named `mysql_no_login` for setting up accounts that accept no client connections. This plugin enables DBAs to implement the following use cases:

  - Stored program and view objects that perform sensitive or adminstrative operations must run with elevated privileges. Less-privileged users must be able to execute these objects but not be able to directly log in as the account that has the privileges. To implement this, create a no-login account using `mysql_no_login`, grant it the required privileges, define objects with a `DEFINER` of that account, and include `SQL SECURITY DEFINER` in the definitions.

  - Access to a proxy account must always be by the usual proxy mechanism, never by users logging in directly to the proxy account. To implement this, assign `mysql_no_login` as the authentication plugin when you create the proxy account.

  For more information, see The No-Login Authentication Plugin.

- MySQL distributions now attempt to deploy with SSL and RSA capabilities enabled by default.

  To make it easier to support secure connections, MySQL servers compiled using OpenSSL now can automatically generate SSL and RSA files at startup if they are missing:

  - The server automatically generates server-side and client-side SSL certificate and key files in the data directory if the new `auto_generate_certs` system variable is enabled, no SSL options other than `--ssl` are specified, and the server-side SSL files are missing from the data directory. These files enable secure client connections using SSL.

  - The server automatically generates RSA private/public key-pair files in the data directory if the new `sha256_password_auto_generate_rsa_keys` system variable is enabled, no RSA options are specified, and the RSA files are missing from the data directory. These files enable secure password exchange using RSA over unencrypted connections for accounts authenticated by the `sha256_password` plugin.

The server-side `--ssl` option value now is enabled by default for all servers. For servers compiled using OpenSSL, if `--ssl` is enabled and other SSL options are not given to configure SSL explicitly, the server attempts to enable SSL automatically at startup:

- If the server finds valid SSL files named `ca.pem`, `server-cert.pem`, and `server-key.pem` in the data directory, it enables SSL to permit SSL connections by clients. (These files need not have been autogenerated; what matters is that they have the indicated names and are valid.)
- If the server does not find valid SSL files in the data directory, it continues executing but does not enable SSL.

For any SSL and RSA files that the server finds and uses automatically, it uses the file names to set the corresponding system variables (`ssl_ca`, `ssl_cert`, `ssl_key`, `sha256_password_private_key_path`, `sha256_password_public_key_path`).

For more information, see Configuring MySQL to Use Secure Connections, and Creating SSL and RSA Certificates and Keys using MySQL.

Spatial Data Support

- **InnoDB:** `SPATIAL` indexes can now be used for `InnoDB` tables. `InnoDB` supports indexing of spatial data types, including use of `ALTER TABLE ... ALGORITHM=INPLACE` for online operations (`ADD SPATIAL INDEX`). To support transaction isolation properties, `InnoDB` uses predicate locking. A predicate lock locks the minimum bounding rectangle (MBR) used for a query so that other transactions cannot insert or modify a row that would match the query condition.

  For more information, see Optimizing Spatial Analysis. and Predicate Locks for Spatial Indexes. (Bug #18674219)

- The Open Geospatial Consortium guidelines document the use of open polygons (polygons where the start point is not equal to the end point) but the MySQL GIS implementation did not support them. Now MySQL supports open polygons: An open polygon is converted to a closed one by appending the starting point to the point sequence. Before:

```
mysql> SELECT AsText(PolygonFromText('POLYGON((10 10,20 10,20 20,10 20))'));
+-------------------------------------------------------------+
| AsText(PolygonFromText('POLYGON((10 10,20 10,20 20,10 20))')) |
+-------------------------------------------------------------+
| NULL                                                        |
+-------------------------------------------------------------+
```

  After:

```
mysql> SELECT AsText(PolygonFromText('POLYGON((10 10,20 10,20 20,10 20))'));
+-------------------------------------------------------------+
| AsText(PolygonFromText('POLYGON((10 10,20 10,20 20,10 20))')) |
+-------------------------------------------------------------+
| POLYGON((10 10,20 10,20 20,10 20,10 10))                    |
+-------------------------------------------------------------+
```

  (Bug #17168699)

- `GeometryCollection()` returned `NULL` if the argument contained nonsupported geometries. Now `GeometryCollection()` returns all the proper geometries contained in the argument even if a nonsupported geometry is present. (Bug #17168643)

- This MySQL release makes increased use of the Boost.Geometry library to provide better reliability and increased functionality for spatial functions. As a result, several previously unimplemented functions have been implemented, and several previously existing functions now accept a wider range of geometry argument types (or argument type combinations for functions that take multiple geometries):

  - These previously unimplemented spatial operator functions are now available: `ST_ConvexHull`, `ST_Difference()`, `ST_Intersection()`, `ST_SymDifference()`, and `ST_Union()`.

  - These previously existing functions are more robust and return non-`NULL` values for more geometry argument types: `ST_Area()`, `ST_Centroid()`, `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Distance()`, `ST_Envelope()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, and `ST_Within()`. These functions also exist as non-`ST_` synomyms; for example, `ST_Area()` and `Area()` are synonyms.

- MySQL now includes functions for converting between GeoJSON documents and spatial values: `ST_AsGeoJSON()` and `ST_GeomFromGeoJSON()`. For more information, see Spatial GeoJSON Functions.

- MySQL now includes functions that enable manipulation of geohash values, which provides applications the capabilities of importing and exporting geohash data, and of indexing and searching geohash values:

  - `ST_GeoHash()` returns a geohash string given a return value length and either longitude and latitude values or a `POINT` value.

  - `ST_LongFromGeoHash()` and `ST_LatFromGeoHash()` return the longitude or latitude value, respectively, given a geohash string argument.

  - `ST_PointFromGeoHash()` produces a `POINT` value from a geohash string argument.

SQL Mode Notes

- **Incompatible Change:** These SQL mode changes were made:

  - Strict SQL mode for transactional storage engines (`STRICT_TRANS_TABLES`) is now enabled by default.

  - Implementation of the `ONLY_FULL_GROUP_BY` SQL mode has been made more sophisticated, to no longer reject deterministic queries that previously were rejected.

    - MySQL now recognizes when a nonaggregated selected column is functionally dependent on (uniquely determined by) `GROUP BY` columns.

    - MySQL has an extension to standard SQL that permits references in the `HAVING` clause to aliased expressions in the select list. Previously, enabling `ONLY_FULL_GROUP_BY` disables this extension, thus requiring the `HAVING` clause to be written using unaliased expressions. This restriction has been lifted so that the `HAVING` clause can refer to aliases regardless of whether `ONLY_FULL_GROUP_BY` is enabled.

    In consequence, `ONLY_FULL_GROUP_BY` is now enabled by default, to prohibit nondeterministic queries containing expressions not guaranteed to be uniquely determined within a group.

  - The changes to the default SQL mode result in a default `sql_mode` system variable value with these modes enabled: `ONLY_FULL_GROUP_BY`, `STRICT_TRANS_TABLES`, `NO_ENGINE_SUBSTITUTION`.

  - The `ONLY_FULL_GROUP_BY` mode is now included in the modes comprised by the `ANSI` SQL mode.

- A new function, `ANY_VALUE()`, is available that can be used to force MySQL to accept queries that it thinks should be rejected with `ONLY_FULL_GROUP_BY` enabled. The function return value and type are the same as the return value and type of its argument, but the function result is not checked for the `ONLY_FULL_GROUP_BY` SQL mode.

If you find that having `ONLY_FULL_GROUP_BY` enabled causes queries for existing applications to be rejected, either of these actions should restore operation:

- If it is possible to modify an offending query, do so, either so that nondeterministic nonaggregated columns are functionally dependent on `GROUP BY` columns, or by referring to nonaggregated columns using `ANY_VALUE()`.
- If it is not possible to modify an offending query (for example, if it is generated by a third-party application), set the `sql_mode` system variable at server startup to not enable `ONLY_FULL_GROUP_BY`.

For more information about SQL modes and `GROUP BY` queries, see Server SQL Modes, and MySQL Handling of GROUP BY. (Bug #18486310)

Functionality Added or Changed

- **Incompatible Change:** The `GET_LOCK()` function has been reimplemented using the metadata locking (MDL) subsystem and its capabilities have been extended:
  - Previously, `GET_LOCK()` permitted acquisition of only one named lock at a time, and a second `GET_LOCK()` call released any existing lock. Now `GET_LOCK()` permits acquisition of more than one simultaneous named lock and does not release existing locks.

    Applications that rely on the behavior of `GET_LOCK()` releasing any previous lock must be modified for the new behavior.
  - The capability of acquiring multiple locks introduces the possibility of deadlock among clients. The MDL subsystem detects deadlock and returns an `ER_USER_LOCK_DEADLOCK` error when this occurs.
  - The MDL subsystem imposes a limit of 64 characters on lock names, so this limit now also applies to named locks. Previously, no length limit was enforced.
  - Locks acquired with `GET_LOCK()` now appear in the `metadata_locks` Performance Schema table. The `OBJECT_TYPE` column says `USER LEVEL LOCK` and the `OBJECT_NAME` column indicates the lock name.
  - A new function, `RELEASE_ALL_LOCKS()` permits release of all acquired named locks at once.

  For more information, see Miscellaneous Functions.
- **Incompatible Change:** In MySQL 5.6.6, the `YEAR(2)` data type was deprecated. Support for `YEAR(2)` has now been removed. Once you upgrade to MySQL 5.7.5 or newer, any remaining `YEAR(2)` columns must be converted to `YEAR(4)` to become usable again. For conversion strategies, see YEAR(2) Limitations and Migrating to YEAR(4). For example, run `mysql_upgrade` after upgrading.
- **Incompatible Change:** Previously, `mysql_upgrade` performed an upgrade by invoking the `mysql` and `mysqlcheck` clients. `mysql_upgrade` has been reimplemented to generate the required SQL statements itself and execute them by communicating directly with server.

  In consequence of this change, `mysql_upgrade` now supports the `--bind-address` option enabling the network interface for connecting to the server to be chosen. It also supports `--net-buffer-`

`length` and `--max-allowed-packet` options enabling the initial and maximum communication packet size to be specified.

Also in consequence of this change, `mysql_upgrade` no longer supports the `--tmpdir` option. This option specified the location of temporary files used to supply input to `mysql`, but it no longer has any purpose because `mysql_upgrade` no longer invokes `mysql`. Any upgrade scripts that invoke `mysql_upgrade` and use `--tmpdir` must be modified to remove that option.

- **InnoDB:** For optimal shutdown and recovery performance, shutdown and recovery phases are now supported by the multi-threaded page cleaner feature (`innodb_page_cleaners`) that was introduced in MySQL 5.7.4. (Bug #18805275)

- **InnoDB:** Work was done to introduce the notion of attachable transactions in `InnoDB` (for AutoCommit / ReadOnly / ReadCommitted / NonLocking transactions). This is used to read from `InnoDB` Data Dictionary tables. Along with this, attachable transactions were exposed to the server. Data Dictionary access code will use them to read Data Dictionary data.

- **InnoDB:** You can now truncate undo logs that reside in undo tablespaces. This feature is enabled using the `innodb_undo_log_truncate` configuration option. For more information, see Truncating Undo Logs That Reside in Undo Tablespaces.

- **InnoDB:** `InnoDB` memory allocations now are instrumented for the Performance Schema and will appear in the memory summary tables.

- **InnoDB:** Instead of inserting one index record at a time, `InnoDB` now performs a bulk load when creating or rebuilding indexes. This method of index creation is also known as a sorted index build. This enhancement, which improves the efficiency of index creation, also applies to full-text indexes. Is is not supported with spatial indexes.

  A new global configuration option, `innodb_fill_factor`, defines the percentage of space on each page that is filled with data during a sorted index build, with the remaining space reserved for future index growth. For more information, see Sorted Index Builds.

- **InnoDB:** The `FIL_PAGE_FLUSH_LSN` field, written to the first page of each `InnoDB` system tablespace file and to `InnoDB` undo tablespace files, is now only written to the first file of the `InnoDB` system tablespace (page number 0:0).

  As a result of this patch, if you have a multiple-file system tablespace and decide to downgrade from MySQL 5.7 to MySQL 5.6, you may encounter an invalid message on MySQL 5.6 startup stating that `the log sequence numbers x and y in ibdata files do not match the log sequence number y in the ib_logfiles`. If you encounter this message, restart MySQL 5.6 to ensure that startup has run properly. The invalid message should no longer appear.

- **InnoDB:** The `innodb_buffer_pool_size` parameter is now dynamic, allowing you to resize the buffer pool without restarting the server. The resizing operation, which involves moving pages to a new location in memory, is performed chunks. Chunk size is configurable using the new `innodb_buffer_pool_chunk_size` configuration option. You can monitor resizing progress using the new `Innodb_buffer_pool_resize_status` status variable. For more information, see Configuring InnoDB Buffer Pool Size Online.

  See Configuring InnoDB Buffer Pool Size Online for more information.

- **Replication:** When replicating from a master running a version earlier than MySQL 5.6.0 to a slave running MySQL 5.6.0 or later, the slave requires the `master_uuid` value, which is the `server_uuid` value from the master. The `master_uuid` value is unsupported on the older master, and in such a replication situation could become invalid on the newer slave. A check for empty `master_uuid` now ensures that the slave uses an empty value for `master_uuid`. (Bug #18338203)

- **Replication:** Retrying of transactions is now supported when multi-threading is enabled on a slave. In previous versions, `slave_transaction_retries` was treated as equal to 0 when using multi-threaded slaves. (Bug #16390504, Bug #68465)

- **Replication:** Global transaction identifiers (GTIDs) are now logged in a MySQL system table whenever they are enabled on the server, which lifts a previous requirement to use binary logging when replicating with GTIDs. If binary logging is disabled, the server stores the GTID for each transaction in the `mysql.gtid_executed` table as the transaction is executed. If binary logging is enabled, then, whenever the binary log is rotated or the server is shut down, the server also writes into the new binary log the GTIDs for all transactions from the previous binary log.

  Because the `mysql.gtid_executed` table can become filled with many rows with single-transaction GTIDs having the same originating server and sequential transaction IDs, the server compresses this table periodically whenever GTIDs are enabled. You can control the frequency with which the table is compressed by setting the `executed_gtids_compression_period` system variable. This variable's default value is 1000, which means that compression of the table is applied following each 1000 transactions. You can set the `executed_gtids_compression_period` to 0 to disable the compression altogether, but you should be aware that doing this may cause the space required by this table to increase significantly. (See mysql.gtid_executed Table Compression.)

  Compression of the mysql.gtid_executed table is performed by a dedicated thread. You can obtain information about the state of this thread in the `threads` Performance Schema table. (Bug #14730192)

- **Replication:** The new variable `simplified_binlog_gtid_recovery` can be used to change the way binary log files are searched for previous GTIDs during recovery, speeding up the process when a large number of binary log files exist. (Bug #69097, Bug #16741603, Bug #74071, Bug #19686914)

- **Replication:** The new system variable `binlogging_impossible_mode` controls what happens if the server cannot write to the binary log, for example, due to a file error. For backward compatibility, the default for `binlogging_impossible_mode` is `IGNORE_ERROR`, meaning the server logs the error, halts logging, and continues updates to the database. Setting this variable to `ABORT_SERVER` makes the server halt logging and shut down if it cannot write to the binary log. (Bug #51014, Bug #11758766)

- **Replication:** To make monitoring of a replication setup easier, various replication related variables have been moved to the `performance_schema` tables. This is particularly helpful for monitoring multi-source replication.

- **Replication:** The new SQL function `WAIT_FOR_EXECUTED_GTID_SET()` makes the current syncing option for the slave with master independent of the slave threads and improves the return value.

- **Replication:** The new options `binlog_group_commit_sync_delay` and `binlog_group_commit_sync_no_delay_count` provide a way to configure the synchronization of the binary log. This enables more transactions to be synchronized together to disk at once, reducing the overall time to commit a group of transactions because the larger groups require fewer time units per group.

- **Replication:** Multi-threaded slaves can use the new `slave_preserve_commit_order` variable to ensure that the order which transactions were committed on the master is preserved on the slave. This prevents the slave from entering a state that the master was not in and is well suited to using multi-threaded slaves for replication read scale-out.

- New Debian7, Ubuntu12.04, and Ubuntu14.04 distribution support that was introduced with 5.6.17 now comes with the platform-specific packaging source placed under the `packaging` directory, in the `deb-precise`, `deb-wheezy`, and `deb-trusty` directories. (Bug #19020385)

- `CMake` support was updated to handle `CMake` version 3. (Bug #19001781)

- The rwlock used for the `SAFE_HASH` implementation is now instrumented for the Performance Schema. The instrument name is `wait/synch/rwlock/mysys/SAFE_HASH::lock`. (Bug #18991366)
- The (undocumented) `binary-configure.sh` script has been removed from MySQL distributions. (Bug #18694238)
- RHEL 4 is not supported for 5.7, so the `support-files/RHEL4-SElinux` file was removed. (Bug #18651087)
- The (undocumented) `FEATURE_SET CMake` option was removed. (Bug #18521389)
- Unused private fields reported by Clang's `-Wunused-private-field` compiler warning option were removed. (Bug #18489724)
- `thr_alarm.h` and `thr_alarm.c` were removed because they contain dead code almost exclusively. The remaining live code was moved to `mysqld.cc`. `my_alarm.h` and `my_alarm.c` were also removed, and the code from them that is actually used was moved to `my_lock.c`. (Bug #18411456)
- The deprecated `timed_mutexes` system variable has been removed. (Bug #18277305)
- `CMake` support was updated to handle the new directory layout for Sun C++ 5.13. (Bug #73034, Bug #19010286)
- The obsolete and unmaintained `charset2html` utility has been removed from MySQL distributions. (Bug #71897, Bug #18352347)
- `mysqld` help text for `--general_log` was clarified. Thanks to Andrew Gaul for the patch. (Bug #71463, Bug #18127243)
- The `fill_help_tables.sql` file that is used to load server-side help table content now contains the following statement to suppress binary logging and prevent table contents from replicating to slaves:

  ```
  SET sql_log_bin=0;
  ```

  Because help table content is specific to the a particular server version, this prevents loading incorrect content into the slaves, which do not necessarily run the same version of MySQL as the master. For more information, see Replication of Server-Side Help Tables. (Bug #69564, Bug #17015822)
- The empty string provided for numeric or enumeration options (for example, `--port=""`) produced inconsistent or confusing behavior. Such empty option values now are rejected with an error. (Bug #68055, Bug #16102788)
- The `mysqladmin flush-logs` command now permits optional log types to be given, to specify which logs to flush. Following the `flush-logs` command, you can provide a space-separated list of one or more of the following log types: `binary`, `engine`, `error`, `general`, `relay`, `slow`. These correspond to the log types that can be specified for the `FLUSH LOGS` SQL statement. Thanks to Daniël van Eeden for the patch. (Bug #60878, Bug #12368203)
- A new status variable, `Max_used_connections_time`, indicates the time at which `Max_used_connections` reached its current value. Thanks to Jordi Prats for the patch. (Bug #59738, Bug #11766596)
- Previously, on Unix and Unix-like systems, MySQL support for sending the server error log to `syslog` was implemented by having `mysqld_safe` capture server error output and pass it to `syslog`. The server now includes native `syslog` support, which has been extended to include Windows. Server error logging to `syslog`, Event Log, or a file, should be controlled using `mysqld` options. Doing so using `mysqld_safe` options is now deprecated. For more information about sending server error output to `syslog`, see The Error Log. (Bug #55370, Bug #11762739)

- If connection IDs went beyond the 32-bit limit and started over at 1, the server now ensures that IDs still in use will not be reissued. (Bug #44167, Bug #11752851)

- Internally, spatial data types such as `Geometry` are represented as `BLOB` values, so when invoked with the `--hex-blob` option, `mysqldump` now displays spatial values in hex. (Bug #43544, Bug #11752369)

- Scalability for `InnoDB` tables was improved by avoiding `THR_LOCK` locks. As a result of this change, DML statements for `InnoDB` tables that previously waited for a `THR_LOCK` lock will wait for a metadata lock:

  - Explicitly or implicitly started transactions that update any table (transactional or nontransactional) will block and be blocked by `LOCK TABLES ... READ` for that table. This is similar to how `LOCK TABLES ... WRITE` works.

  - Tables that are implicitly locked by `LOCK TABLES` now will be locked using metadata locks rather than `THR_LOCK` locks (for `InnoDB` tables), and locked using metadata locks in addition to `THR_LOCK` locks (for all other storage engines). Implicit locks occur for underlying tables of a locked view, tables used by triggers for a locked table, or tables used by stored programs called from such views and triggers.

    Multiple-table updates now will block and be blocked by concurrent `LOCK TABLES ... READ` statements on any table in the update, even if the table is used only for reading.

  - `HANDLER ... READ` for any storage engine will block and be blocked by a concurrent `LOCK TABLES ... WRITE`, but now using a metadata lock rather than a `THR_LOCK` lock.

  The preceding changes are visible several ways. For example, when a DML statement such as `INSERT INTO t1` in one session is blocked by `LOCK TABLES t1 READ` in another session:

  - In the Performance Schema, `THR_LOCK` acquisitions and waits will be registered in the `metadata_locks` table and for `wait/lock/metadata/sql/mdl` events rather than registered in the `table_handles` table and for `wait/lock/table/sql/handler` events.

  - In the process list (`SHOW PROCESSLIST` or `INFORMATION_SCHEMA.PROCESSLIST`), the state value will be `Waiting for table metadata lock` rather than `Waiting for table level lock`.

  - The `Table_locks_immediate` and `Table_locks_waited` status variables will no longer be incremented.

  Issues that went away as a result of these locking changes:

  - For debug builds, concurrent execution of `LOCK TABLES ... READ` and a DML statement affecting the same `InnoDB` table might lead to `Found lock of type 6 that is write and read locked` warnings in the error log.

  - Execution of DDL statements under `LOCK TABLES` might have led to deadlock if 1) this `LOCK TABLES` statement, in addition to the table to be changed by DDL, also had some tables read-locked or locked implicitly (for example, through triggers) and 2) there was some concurrent DML which was blocked on a table-level lock held by `LOCK TABLES` and 3) there was a concurrent `PREPARE` (or corresponding connector API call) which prepared a statement using tables to be affected by the first DDL and some other table which was affected by another DDL statement which had to wait for the DML statement.

  (Bug #42147, Bug #11751331)

- The `mysql` client now indicates whether `USE` statements produced warnings. (Bug #29965, Bug #11746951)

- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.

- In the MySQL client/server protocol, EOF and OK packets serve the same purpose, to mark the end of a query execution result. Due to recent changes in the OK packet (such as session state tracking), and to avoid repeating the changes in the EOF packet, the EOF packet is now deprecated.

- The server-side help tables and time zone tables in the `mysql` system database now are `InnoDB` (transactional) tables. Previously, these were `MyISAM` (nontransactional) tables. The affected tables are:

  ```
  help_category
  help_keyword
  help_relation
  help_topic
  time_zone
  time_zone_leap_second
  time_zone_name
  time_zone_transition
  time_zone_transition_type
  ```

  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` database.

  `START TRANSACTION` and `COMMIT` statements have been added to the output from `mysql_tzinfo_to_sql` used to populate the time zone tables, to ensure that reload operations are permanent.

- MySQL Server now supports an “offline mode” with these characteristics:

  - Connected client users who do not have the `SUPER` privilege are disconnected on the next request, with an appropriate error. Disconnection includes terminating running statements and releasing locks. Such clients also cannot initiate new connections, and receive an appropriate error.

  - Connected client users who have the `SUPER` privilege are not disconnected, and can initiate new connections to manage the server.

  - Replication slave threads are permitted to keep applying data to the server.

  Only users who have the `SUPER` privilege can control offline mode. To put a server in offline mode, change the value of the new `offline_mode` system variable from `OFF` to `ON`. To resume normal operations, change `offline_mode` from `ON` to `OFF`. In offline mode, clients that are refused access receive an `ER_SERVER_OFFLINE_MODE` error.

- `ALTER TABLE ... EXCHANGE PARTITION` syntax now includes an optional `{WITH|WITHOUT} VALIDATION` clause. When `WITHOUT VALIDATION` is specified, `ALTER TABLE ... EXCHANGE PARTITION` does not perform row-by-row validation when exchanging a populated table with the partition, permitting database administrators to assume responsibility for ensuring that rows are within the boundaries of the partition definition. `WITH VALIDATION` is the default behaviour and need not be specified explicitly. For more information, see Exchanging Partitions and Subpartitions with Tables.

- `mysqlslap` now has a `--sql-mode` option that enables the SQL mode to be set for the client session.

- It is now possible to specify the storage engine the server uses for on-disk internal temporary tables (see Internal Temporary Table Use in MySQL), by setting the new `internal_tmp_disk_storage_engine` system variable. Permitted values are `MYISAM` (the default) and `INNODB`.

- The custom rwlock implementation for Windows was replaced with standard Windows API calls. As a result of this change, Windows binaries require Windows 7 / Windows Server 2008 R2 or newer. In particular, Windows binaries no longer work on Windows Vista or Windows Server 2008 (plain, not R2).
- The Boost.Geometry library now is required to build MySQL. Two new `CMake` options enable control over the library source location, and whether to download it automatically:
  - `-DWITH_BOOST=path_name` specifies the Boost library directory location. It is also possible to specify the Boost location by setting the `BOOST_ROOT` or `WITH_BOOST` environment variable.
  - `-DDOWNLOAD_BOOST=bool` specifies whether to download the Boost source if it is not present in the specified location. The default is `OFF`.

  For example, if you normally build MySQL placing the object output in the `bld` subdirectory of your MySQL source tree, you can build with Boost like this:

  ```
  mkdir bld
  cd bld
  cmake .. -DDOWNLOAD_BOOST=ON -DWITH_BOOST=$HOME/my_boost
  ```

  This causes Boost to be downloaded into the `my_boost` directory under your home directory. If the required Boost version is already there, no download is done. If the required Boost version changes, the newer version is downloaded.

  If Boost is already installed locally and your compiler finds the Boost header files on its own, it may not be necessary to specify the preceding `CMake` options. However, if the version of Boost required by MySQL changes and the locally installed version has not been upgraded, you may have build problems. Using the `CMake` options should give you a successful build.
- The deprecated `mysqlhotcopy` utility has been removed from MySQL distributions. Alternatives include `mysqldump` and MySQL Enterprise Backup.
- The deprecated `mysqlbug`, `mysql_waitpid`, and `mysql_zap` utilities have been removed from MySQL distributions.
- The deprecated `storage_engine` system variable has been removed. Use `default_storage_engine` instead.

Bugs Fixed

- **Performance; Replication:** When processing the dump thread, a semisynchronous replication master checked whether or not the dump thread came from a semisynchronous slave by checking the value of `rpl_semi_sync_slave_enabled`, but did so for every operation performed on this thread, which had significant negative impact on performance. Now this check is made only once, when the dump thread is started, which should noticeably improve the performance of semisynchronous replication in most cases. (Bug #17932935)
- **Important Change; Partitioning:** In an `ALTER TABLE` statement, the server accepted `REBUILD` with the name of a subpartition as valid syntax even though the `REBUILD` keyword in this case did nothing. Now `REBUILD` is rejected in such cases, and causes the statement to fail with an error. (Bug #19075411, Bug #73130)

  References: This issue is a regression of: Bug #14028340, Bug #65184.
- **Important Change; Replication:** A `DROP TABLE` statement may be divided into multiple statements before it is sent to the binary log if it contains regular (not temporary) tables and temporary tables, or if it contains temporary tables using both transactional and non-transactional storage engines. Now,

when using GTIDs, DROP TABLE statements affecting these combinations of tables are no longer allowed unless the value of the gtid_next system variable is AUTOMATIC. This is because, with GTIDs enabled on the server, issuing a DROP TABLE in the cases just described while having only one GTID associated with each statement (the SQL thread does this following SET gtid_next='*uuid*:*number*') causes problems when there are not enough GTIDs for assignment to all the resulting statements following the division of the original DROP TABLE.

A DROP TABLE statement might be split due to the behavior of the statement with respect to the current transaction varying, depending on table characteristics, as follows:

- DROP TABLE of a regular (not temporary) table is committed immediately
- DROP TABLE of a temporary table using a transactional storage engine is committed with the current transaction (following COMMIT)
- DROP TABLE of a temporary table that uses a nontransactional storage engine is committed immediately

Naming all three of these types of tables in a single DROP TABLE statement causes the MySQL server to divide the original statement into three separate DROP TABLE statements in the binary log. If GTIDs are enabled but the value of gtid_next is not AUTOMATIC, issuing a DROP TABLE statement that mixes any of the table types described previously causes the server to have an insufficient number of GTIDs to write with all of the resulting statements into the binary log. In addition, DROP TABLE IF EXISTS is always written in the binary log for all tables specified in the statement, even if some or all of the tables do not exist.

Because temporary tables are handled differently by DROP TABLE depending on whether they use a transactional or nontransactional storage engine, any tables named by a DROP TEMPORARY TABLE statement that do not exist are assumed to be transactional. This means that, if a DROP TEMPORARY TABLE with two nontransactional temporary tables is issued on the master, it would writes only one DROP TABLE statement naming both tables. If one of the temporary tables no longer exists on the slave, then, when the SQL thread executes the statement, it tries to divide it into multiple statements due to it affecting a nontransactional (but existing) temporary table and a nonexistent transactional temporary table; this leads to problems because the SQL thread has only one GTID for the original DROP TABLE statement but must write two DROP TABLE statements in the binary log.

In addition, when the slave dropped temporary tables after detecting that the master had restarted, it logged one DROP TABLE statement per pseudo-thread and per database, but combined temporary tables using transactional and nontransactional storage engines in a single DROP TABLE statement.

Now, we throw an error in the client session if gtid_next is set to a *uuid*:*number* value and a DROP TABLE statement is issued mixing any of the table types described previously.

In addition, we now group the nonexistent temporary tables and assume them to be transactional only if at least one transactional temporary table is dropped by the statement. If no transactional temporary tables are dropped, any nonexistent temporary tables are assumed to be nontransactional temporary tables.

The slave now also handles dropping of temporary tables correctly in the event of the restart by the master. (Bug #17620053)

- **Important Change; Replication:** The maximum length that can be used for the password in a CHANGE MASTER TO statement is 32 characters. Previously, when a longer password was employed, it was accepted, but any excess length was silently truncated by the server. Following this fix, when the password's length exceeds 32 characters, CHANGE MASTER TO fails with an error. (Bug #11752299, Bug #43439)

- **InnoDB; Partitioning:** Large numbers of partitioned `InnoDB` tables could consume much more memory when used in MySQL 5.6 or 5.7 than the memory used by the same tables used in previous releases of the MySQL Server. (Bug #17780517, Bug #70641)

  References: This issue is a regression of: Bug #11764622, Bug #57480.

- **InnoDB:** The `TTASFutexMutex`, the mutex implementation that uses the Linux futex, failed to request `m_lock_word` alignment. (Bug #19525395, Bug #73760)

- **InnoDB:** If a crash occurs after a drop index action during a `TRUNCATE TABLE` operation, the root page could be left in a free state. On crash recovery, an LSN check failed to check free pages when attempting to pin the root page, resulting in an assertion. (Bug #19520482)

- **InnoDB:** Replaced a `goto` statement with an `if` statement in the `fil_mutex_enter_and_prepare_for_io` function in `fil0fil.cc`. (Bug #19488149)

- **InnoDB:** An `INSERT` operation on a table with spatial data columns raised an assertion. (Bug #19474851)

- **InnoDB:** Removed unused code related to binary log information stored in the `InnoDB` `trx_sys` page. (Bug #19471743)

- **InnoDB:** An `ALTER TABLE ... ADD FOREIGN KEY` operation could cause a serious error. (Bug #19471516, Bug #73650)

- **InnoDB:** Reduced `mem_heap_zalloc` calls in `upd_create`. Only a single call is necessary to allocate memory for `upd_t`. (Bug #19297656, Bug #73272)

- **InnoDB:** In debug builds, an `INSERT` operation affecting compressed tables would raise a sync-related assertion. (Bug #19295893)

- **InnoDB:** An `ALTER TABLE` operation raised an `fk_tables.empty()` assertion. After calling `dict_load_foreigns()`, all associated tables that are linked by a foreign key should be loaded to ensure that foreign keys are cached. (Bug #19267051)

  References: This issue is a regression of: Bug #16244691.

- **InnoDB:** A transaction returned from the transaction pool was not in a clean state. (Bug #19244969)

- **InnoDB:** An `MLOG_CHECKPOINT` marker was incorrectly omitted when performing a log checkpoint. (Bug #19233510, Bug #73304)

- **InnoDB:** On crash recovery, `InnoDB` would call `exit()` when encountering a corruption or inconsistency. Some of the `exit()` calls have been removed to allow `InnoDB` to shut down properly. (Bug #19229231, Bug #73300)

- **InnoDB:** When multiple `daemon_memcached_option` options are defined, the `INFORMATION_SCHEMA.GLOBAL_VARIABLES` table and `SHOW VARIABLES` statement should only display the first `daemon_memcached_option` option that is defined in the command line string or in the MySQL configuration file. (Bug #19204759, Bug #73287)

- **InnoDB:** Running `SHOW ENGINE INNODB STATUS` repeatedly under performance testing conditions could result in a serious error. (Bug #19196052)

- **InnoDB:** Removed unused `one_flush` variable from `storage/innobase/include/log0log.h`. (Bug #19192364, Bug #73269)

- **InnoDB:** Retrieval of multiple values with a single `get` command would return incorrect results instead of an error message. The `InnoDB` `memcached` plugin does not currently support retrieval of multiple values with a single `get` command. (Bug #19172212, Bug #72453)

- **InnoDB:** Attempting to perform operations on a timed out key would cause the `memcached` daemon to crash and restart. (Bug #19172013, Bug #72586)

- **InnoDB:** An `ALTER TABLE` operation that does not perform a sort on the clustered index could result in a duplicate record. (Bug #19163915)

  References: See also: Bug #17657223.

- **InnoDB:** Improved error handling, diagnostics, and test coverage related to crash recovery error handling. (Bug #19145637, Bug #73179)

- **InnoDB:** Improved error handling for calls to `handler::records()`. (Bug #19142753)

  References: This issue is a regression of: Bug #16802288.

- **InnoDB:** With a transaction isolation level less than or equal to `READ COMMITTED`, gap locks were not taken when scanning a unique secondary index to check for duplicates. As a result, duplicate check logic failed allowing duplicate key values in the unique secondary index. (Bug #19140907)

  References: This issue is a regression of: Bug #16133801.

- **InnoDB:** A race condition that occurred when dynamically disabling `innodb_adaptive_hash_index` caused the purge thread to assert. (Bug #19069698)

- **InnoDB:** The `INNODB_PAGE_ATOMIC_REF_COUNT` `CMake` option is removed in MySQL 5.7.5. This option was enabled by default but could be disabled for systems that do not support atomics. As of MySQL 5.7.5, support for atomics is required to build MySQL, making the `INNODB_PAGE_ATOMIC_REF_COUNT` option obsolete. (Bug #19061440)

- **InnoDB:** In debug builds, an invalid `rw_latch == RW_NO_LATCH` assertion would cause the server to halt. (Bug #18977128)

- **InnoDB:** Added debug assertions to the adaptive hash index code to check that the tablespace ID in buffer blocks match the index space. (Bug #18965518, Bug #72986)

- **InnoDB:** During recovery, a segmentation fault would occur when marking a table as corrupt. (Bug #18942294)

  References: This issue is a regression of: Bug #11830883.

- **InnoDB:** A code comment for the `os_event_is_set` function in `storage/innobase/os/os0event.cc` was incorrect. (Bug #18940008, Bug #72919)

- **InnoDB:** A latching order violation would occur while inserting `BLOB` data. (Bug #18883885)

- **InnoDB:** For PowerPC, `InnoDB` now uses special PowerPC instructions for setting priority of hardware threads in `InnoDB` mutex spin loops. Thanks to Stewart Smith for the contribution. (Bug #18842925, Bug #72754)

- **InnoDB:** The `innodb_memcached_config.sql` configuration script failed after running the `mysql_secure_installation` script, which removes the MySQL `test` database. The `innodb_memcached_config.sql` script now creates the `test` database if it does not exist. (Bug #18816381, Bug #72678)

- **InnoDB:** Removed unused function definitions and declarations from the `InnoDB memcached` API. (Bug #18815992, Bug #72723)
- **InnoDB:** `CACHE_LINE_SIZE` for PowerPC was changed from 64 bytes to 128 bytes. (Bug #18814859, Bug #72718)
- **InnoDB:** Opening a parent table that has thousands of child tables could result in a long semaphore wait condition. (Bug #18806829)
- **InnoDB:** `trx_cleanup_at_db_startup` failed to reset `trx->rsegs->m-redo` content in debug code. (Bug #18795594)
- **InnoDB:** On `mysqld` start, specifying multiple data files using the `innodb_data_file_path` option would return a `Space id in fsp header` error after data is written to the second file. (Bug #18767811)
- **InnoDB:** When storing `BLOB` data, `InnoDB` failed to reserve the required tablespace pages prior to allocating the pages, which raised an assertion on `INSERT`. (Bug #18756233)
- **InnoDB:** A failed in-place `ALTER TABLE` operation would leave behind non-unique temporary file names in the data dictionary preventing future `ALTER TABLE` operations on the same table due to temporary file name conflicts. To avoid this problem, temporary file names are made unique by appending a static global number that is initialized to a random distributed 32-bit number using `ut_time()` and `ut_crc32()`. The number is then incremented atomically for each assigned temporary file name. Previously, temporary files were named using the format `#sql-ibtid`, where `tid` is the table ID. Temporary files are now named using the format `#sql-ibtid-inc`, where `tid` is the table ID and `inc` is the incremented number. (Bug #18734396, Bug #72594)
- **InnoDB:** A regression introduced by the fix for Bug #11758237 resulted in a `cursor->index->name == TEMP_INDEX_PREFIX` assertion. (Bug #18723872)

  References: This issue is a regression of: Bug #11758237.
- **InnoDB:** For single item full-text searches, deleted documents were included in inverse document frequency (IDF) calculations. (Bug #18711306, Bug #72548)
- **InnoDB:** The `page_create` function has been optimized to use simpler functions to initialize pages. (Bug #18704384)
- **InnoDB:** A `DELETE` operation on a table with full-text search indexes raised an assertion. (Bug #18683832)

  References: See also: Bug #14639605.
- **InnoDB:** To-be-imported tablespace files (`FIL_TYPE_IMPORT`) were not flushed after being converted to normal tablespace files (`FIL_TYPE_TABLESPACE`), resulting in too many open files. (Bug #18663997)

  References: This issue is a regression of: Bug #18236692.
- **InnoDB:** When `InnoDB` is built as a shared library, attempting to load the `InnoDB` full-text search (FTS) `INFORMATION_SCHEMA` plugin would fail with a `Can't open shared library 'ha_innodb.so'` error. (Bug #18655281, Bug #70178)
- **InnoDB:** When calling the `memcached flush_all` command, `InnoDB` attempts to initialize a connection and a transaction. If the transaction is in `TRX_STATE_NOT_STARTED` state, `InnoDB` failed to set `CONN_DATA->CRSR_TRX` to NULL, resulting in a serious error. (Bug #18652854)
- **InnoDB:** An `INSERT` operation on a table with `BLOB` columns raised an assertion. (Bug #18646430)

References: This issue is a regression of: Bug #16963396.

- **InnoDB:** An `INSERT` operation on a table with `GEOMETRY` columns raised an assertion in `rtr_page_split_and_insert()`. (Bug #18644435)
- **InnoDB:** The temporary tablespace file (`ibtmp1`) was held open by the `page_cleaner` thread and could not be removed on startup, resulting in a hang. (Bug #18642372)
- **InnoDB:** A regression introduced in MySQL 5.6.5 would cause full-text search index tables to be created in the system tablespace (space 0) even though `innodb_file_per_table` was enabled. (Bug #18635485)
- **InnoDB:** After upgrading from 5.6.10 to MySQL versions up to and including MySQL 5.6.18, `InnoDB` would attempt to rename obsolete full-text search auxiliary tables on server startup, resulting in an assertion failure. (Bug #18634201, Bug #72079)
- **InnoDB:** In rare cases, the purge process would attempt to delete a secondary index record that was not marked for deletion, resulting in an inconsistent secondary index. (Bug #18631496)
- **InnoDB:** After running `OPTIMIZE TABLE` on an `InnoDB` table with a spatial index, running a `SELECT` statement that uses the spatial index could result in a crash. (Bug #18619945, Bug #72361)

  References: This issue is a regression of: Bug #13975225.
- **InnoDB:** On startup, with `innodb_file_per_table=ON`, the page cleaner thread would raise a `srv_get_active_thread_type() == SRV_NONE` debug assertion when encountering an active master thread. (Bug #18598813)
- **InnoDB:** `InnoDB` would try to merge a b-tree change buffer for a dedicated undo tablespace. (Bug #18593561)
- **InnoDB:** Included `unistd.h` in `innodb.cmake` to enable futexes in debug builds. (Bug #18522549, Bug #72225)
- **InnoDB:** `TRUNCATE TABLE` would write page-level redo logs during the `DROP TABLE` step of a `TRUNCATE TABLE` operation. (Bug #18511398)
- **InnoDB:** A `COMMIT` operation related to full-text search resulted in a segmentation fault. (Bug #18503734)
- **InnoDB:** `srv_active_wake_master_thread()` was called directly in `innobase_commit` and `innobase_prepare`, waking up the master thread and incrementing `srv_activity_count`. `srv_active_wake_master_thread()` should only be called after committing write transactions, not after read-only transactions or rollbacks. This patch also replaces some calls to `srv_active_wake_master_thread()` with calls to `ib_wake_master_thread()`. (Bug #18477009, Bug #72137)
- **InnoDB:** A `DB_LOCK_WAIT` during a foreign key check caused redundant delete marking, resulting in a failing assertion. (Bug #18451287)
- **InnoDB:** With `UNIV_SYNC_DEBUG` enabled, a late call to `sync_check_enable()` would result in an `m_enabled` assertion failure. (Bug #18433658)
- **InnoDB:** `InnoDB` would write to the redo log for an `IMPORT TABLESPACE` operation before the tablespace import was complete. (Bug #18424134)
- **InnoDB:** The `InnoDB memcached` plugin would call `plugin_del` without acquiring the `lock_plugin` mutex. This bug fix also addresses a race condition in `ib_cursor_delete_row`. (Bug #18409840)

- **InnoDB:** The `os_event_wait_time_low` function would sometimes return `OS_SYNC_TIME_EXCEEDED` before the sync time has elapsed. (Bug #18386498)
- **InnoDB:** With persistent statistics enabled, `SHOW TABLE STATUS` output and the `TABLE_ROWS` column of `INFORMATION_SCHEMA.TABLES` could report an incorrect number of table rows for tables with externally stored pages. (Bug #18384390)
- **InnoDB:** Running a `SELECT` on a partitioned table caused a memory access violation in `memcpy()`. (Bug #18383840)

  References: See also: Bug #18167648.
- **InnoDB:** A regression introduced by the fix for Bug#18069105 could result in a table corruption and failing assertions. (Bug #18368345)

  References: This issue is a regression of: Bug #18069105.
- **InnoDB:** The data file was not opened prior to calling `fil_fusionio_enable_atomic_write()`, resulting in an assertion failure. (Bug #18368241)
- **InnoDB:** The fix for Bug#17699331 caused a high rate of read/write lock creation and destruction which resulted in a performance regression. (Bug #18345645, Bug #71708)

  References: This issue is a regression of: Bug #17699331.
- **InnoDB:** Code quality improvements for the redo log subsystem. (Bug #18345004)
- **InnoDB:** Added the C++ `ostream` mechanism for error logging. (Bug #18320915)
- **InnoDB:** Removed the `recv_max_parsed_page_no` code variable, which was only used in a diagnostic error message. (Bug #18312967)
- **InnoDB:** `buf_pool->flush_rbt`, which is only intended to be used for recovery, would be allocated for database creation and never freed. (Bug #18253089)
- **InnoDB:** Calls to `sched_getcpu` would cause page faults. (Bug #18225489)
- **InnoDB:** `ib_heap_resize` failed to verify that `new_size` is greater than or equal to `old_size` before calling `memcpy`. (Bug #18178915)
- **InnoDB:** After crash recovery and with `UNIV_DEBUG` enabled, purge failed with a `buf_pool_from_bpage(bpage) == buf_pool` assertion failure. (Bug #18154145)
- **InnoDB:** Assertion code in `buf_page_release_latch()` in `buf0buf.ic` was too restrictive. (Bug #17869571)
- **InnoDB:** For each insert, `memset` would be called three times to allocate memory for system fields. To reduce CPU usage, the three `memset` calls are now combined into a single call. (Bug #17858679, Bug #71014)
- **InnoDB:** The fix for Bug#16418661 added superfluous `buf_flush_list()` logic to `InnoDB` startup code. (Bug #17798076, Bug #70899)

  References: This issue is a regression of: Bug #16418661.
- **InnoDB:** A problem renaming temporary tables during an `ALTER TABLE` operation would raise an assertion and print a warning to the error log. Temporary table names were not sufficiently unique. (Bug #17713871)

- **InnoDB:** In-place `ALTER TABLE` operations requiring a table rebuild would sort the clustered index even though the primary key order remained unchanged. This behavior caused unnecessary temporary table usage and I/O. (Bug #17657223)

- **InnoDB:** Under certain circumstances, adaptive hash index latches (`btr_search_latch`) were persistent. With atomics-based rw-locks (read-write locks), persistent adaptive hash index latches are unnecessary and may block other adaptive hash index updates. (Bug #17554489, Bug #70216)

- **InnoDB:** A race condition in `fts_get_next_doc_id` resulted in `Duplicate FTS_DOC_ID` and `Cannot find index FTS_DOC_ID_INDEX in InnoDB index translation table` errors. (Bug #17447086, Bug #70311)

  References: See also: Bug #16469399.

- **InnoDB:** Enabling the `InnoDB` Table Monitor would result in a `ib_table->stat_initialized` assertion failure. (Bug #17039528, Bug #69641)

- **InnoDB:** Redo log writes for large, externally stored `BLOB` fields could overwrite the most recent checkpoint. The 5.6.20 patch limits the size of redo log `BLOB` writes to 10% of the redo log file size. The 5.7.5 patch addresses the bug without imposing a limitation. For MySQL 5.5, the bug remains a known limitation. (Bug #16963396, Bug #19030353, Bug #69477)

- **InnoDB:** The error log message that is printed on `CREATE TABLE` when the number of `BLOB` or `TEXT` fields exceed the row size limit did not provide sufficient information. The error message now provides the maximum row size, current row size, and the field that causes the maximum row size to be exceeded. (Bug #16874873, Bug #69336)

- **InnoDB:** An in-place `ALTER TABLE` operation on a table with a broken foreign key constraint could raise an assertion. (Bug #16869435)

- **InnoDB:** The `lock_number_of_rows_locked` function used a bit vector to track the number of record locks held by a transaction. To optimize reporting, the bit vector was replaced by a simple counter. (Bug #16479309, Bug #68647)

- **InnoDB:** Inserting a record into an `InnoDB` table with a key that falls between the maximum key of a full page and the minimum key of the “next” page could result in unnecessary page splits and under-filled pages. If the insert point is at the end of a page, `InnoDB` now attempts to insert to the next page before splitting the page. (Bug #15923864, Bug #67718)

- **InnoDB:** With `innodb_max_dirty_pages_pct=0` buffer pool flushing would not be initiated until the percentage of dirty pages reached at least 1%, which would leave up to 1% of dirty pages unflushed. (Bug #13029450, Bug #62534)

- **InnoDB:** Due to differences in memory ordering on different processor types, some mutex and read-write lock flags were not read consistently. (Bug #11755438, Bug #47213)

- **InnoDB:** Debug assertion code in file `row0ins.cc` did not account for rw-lock shared-exclusive (sx-lock) mode, resulting in an assertion failure. (Bug #73534, Bug #19420253)

- **InnoDB:** When `foreign_key_checks` is disabled, `InnoDB` would allow an index required by a foreign key constraint to be dropped, thereby placing the table into an inconsistent state. Dropping an index required by a foreign key constraint should not be permitted. (Bug #70260, Bug #17449901)

- **Partitioning:** When the `index_merge_intersection` flag (enabled by default) or the `index_merge_union` flag was enabled by the setting of the `optimizer_switch` system variable, queries returned incorrect results when executed against partitoned tables that used the `MyISAM` storage engine, as well as partitioned `InnoDB` tables that lacked a primary key. (Bug #18167648)

References: See also: Bug #16862316, Bug #17588348, Bug #17648468.

- **Partitioning:** Selecting from a table having multiple columns in its primary key and partitioned by `LIST COLUMNS(R)`, where *`R`* was the last (rightmost) column listed in the primary key definition, returned an incorrect result. (Bug #17909699, Bug #71095)

- **Replication:** Misleading or confusing error messages have been revised, which were produced in the following 2 cases when using `mysqlbinlog` with the `--raw` option:

  - When this option was used together with `--include-gtids`.

  - When this option was used together with `--exclude-gtids` and `--read-from-remote-master=BINLOG-DUMP-NON-GTIDS`.

  (Bug #19459836, Bug #73625)

- **Replication:** After the fix for Bug #16861624, killing a multi-threaded slave worker which was waiting for a commit lock caused a debug assertion to fail. This fix ensures that such a situation can not occur. (Bug #19311260)

- **Replication:** Temporary errors related to a slave worker thread which was retrying a transaction were visible in `SHOW SLAVE STATUS`. This fix ensures that slave worker threads do not store temporary errors in the slave status. (Bug #19282301)

- **Replication:** A corrupted header length in `FORMAT_DESCRIPTION_LOG_EVENT` could cause the server to stop unexpectedly. This was due to `FORMAT_DESCRIPTION_LOG_EVENT` being considered invalid if the header length was too short. (Bug #19145712)

- **Replication:** Removed an unnecessary write lock that was taken by an internal function while adding a GTID to a GTID set, which should improve the performance of the function and the code dependent on it during such operations. (Bug #18963555, Bug #72977)

- **Replication:** Semisynchronous replication did not work as expected when the `rpl_semi_sync_master_wait_no_slave` and `rpl_semi_sync_master_wait_for_slave_count` variables were set. The values of the variables were changed, but the related internal status was not updated during initialization. (Bug #18835117, Bug #18466390)

- **Replication:** `RESET SLAVE ALL` did not clear `IGNORE_SERVER_IDS`, although this statement should clear any values that are set by `CHANGE MASTER TO`. Now `RESET SLAVE ALL` always empties the list of server IDs to ignore, whenever it is executed. (Bug #18816897)

- **Replication:** The same internal function had effects which caused three similar problems when resetting or starting slaves. These three issues are listed here:

  - `RESET SLAVE` automatically set the heartbeat period (`Slave_heartbeat_period`) to its default value.

  - `RESET SLAVE` automatically set `SSL_VERIFY_SERVER_CERT` to the default.

  - When a server was not configured as a slave (that is, when no `CHANGE MASTER TO` statement had yet been executed), the subsequent failure of `START SLAVE` was expected but had the unintended side effect of resetting the heartbeat period to the default.

  The function has been rewritten such that code affecting heartbeat or SSL certificate usage has been eliminated or moved to a more appropriate location, eleminating the side effects formerly seen with `RESET SLAVE` or a failed `START SLAVE`.

As part of this fix, in order to be able to keep heartbeats enabled by default when changing the master, if host and port are given but the heartbeat period is not specified in a `CHANGE MASTER TO` statement, we force it to the default value. (Bug #18791604, Bug #18778485, Bug #18777899)

- **Replication:** `mysqlbinlog --raw` did not check for errors caused by failed writes, which could result in silent corruption of binary logs. Now in such cases it stops with an error. (Bug #18742916, Bug #72597)
- **Replication:** When committing a transaction, a flag is now used to check whether a thread has been created, rather than checking the thread itself, which uses more resources, particularly when running the server with `master_info_repository=TABLE`. (Bug #18684222)

  References: See also: Bug #17967378.
- **Replication:** When a slave worker thread tried to execute a statement that was too large, the resulting error caused a crash. Now in such cases, the error is truncated to fit the size of the buffer. (Bug #18563480)
- **Replication:** Log rotation events could cause `group_relay_log_pos` to be moved forward incorrectly within a group. This meant that, when the transaction was retried, or if the SQL thread was stopped in the middle of a transaction following one or more log rotations (such that the transaction or group spanned multiple relay log files), part or all of the group was silently skipped.

  This issue has been addressed by correcting a problem in the logic used to avoid touching the coordinates of the SQL thread when updating the log position as part of a relay log rotation whereby it was possible to update the SQL thread's coordinates when not using a multi-threaded slave, even in the middle of a group. (Bug #18482854)
- **Replication:** When using GTIDs with `MASTER_AUTO_POSITION` enabled, if an I/O thread was restarted it failed with an `ER_GTID_NEXT_TYPE_UNDEFINED_GROUP` error due to a partial transaction not being correctly rolled back before resuming the I/O thread. This fix ensures that the partial transaction is correctly rolled back. (Bug #18472603)
- **Replication:** When using row-based replication, updating or deleting a row on the master that did not exist on the slave led to failure of the slave when it tried to process the change. This problem occurred with `InnoDB` tables lacking a primary key. (Bug #18432495, Bug #72085)
- **Replication:** A multi-threaded slave now checks that a free worker is available to dispatch the next event. In previous versions, a multi-threaded slave crashes with a worker error if the thread coordinator can not find a free worker. The crash was caused when the coordinator did not return a free worker, for example if the coordinator was aborted at the same time using `STOP SLAVE`. (Bug #18363515)
- **Replication:** When replicating from a MySQL 5.5 or earlier master to a MySQL 5.6 or later slave, the `SOURCE_UUID` column of the slave's `performance_schema.replication_connection_status` table contained random data. Now in such cases, `SOURCE_UUID` is left blank. (Bug #18338203)
- **Replication:** During relay log initialization, the thread context was used as a flag for the reconstruction of the retrieved GTID set, an operation that does not depend on this parameter. This could be problematic if relay log initialization was called in another context other than the legacy replication scenario; if the invocation was made in a context where the thread context was always present, this prevented the set's reconstruction. The opposite could also happen when the thread context was not present, which cause the initialization to be performed twice.

  To avoid such issues, the thread context flag is replaced with a new flag that allows the reconstruction in all contexts but prevents multiple invocations. (Bug #18337036)
- **Replication:** When `mysqlbinlog` processed multiple binary log files into a single output file, this file was not in a useful state for point-in-time recovery, when it failed with the error, `When`

`@@SESSION.GTID_NEXT is set to a GTID, you must explicitly set it to a different value after a COMMIT or ROLLBACK. Please check GTID_NEXT variable manual page for detailed explanation. Current @@SESSION.GTID_NEXT is 'xyz'`. When `mysqlbinlog` processes a binary log containing GTIDs, it outputs `SET gtid_next` statements, but `gtid_next` is set to undefined whenever a commit occurs; this left `gtid_next` undefined when the server had finished processing the output from `mysqlbinlog`. When the next binary log file started with one or more anonymous statements or transactions, the combination of gtid_next being left undefined at the end of the first binary log and the second binary log containing anonymous transactions to the error described previously (Error 1837, **ER_GTID_NEXT_TYPE_UNDEFINED_GROUP**).

To fix this issue, now, whenever `mysqlbinlog` encounters this situation, it inserts `SET gtid_next = AUTOMATIC` if required to avoid leaving the previous binary log with `gtid_next` undefined.

In addition, as a result of this fix, `mysqlbinlog` no longer outputs session variable information for every binary log; now, this value is printed only once unless it changes. (Bug #18258933, Bug #71695)

- **Replication:** Quotation marks were not always handled correctly by `LOAD DATA INFILE` when written into the binary log. (Bug #18207212, Bug #71603)

- **Replication:** Changing `master_info_repository` between `TABLE` and `FILE` could produce duplicate values in the `Replicate_Ignore_Server_Ids` column of the output from `SHOW SLAVE STATUS` and the `Ignored_server_ids` column of the `mysql.slave_master_info` table. This could arise because the list of IDs stored internally by the server was not always sorted before checking for duplicates when inserting new values into the list. Now the list of server IDs to be ignored is always sorted, prior to performing this check. (Bug #18192817, Bug #18593479, Bug #18920203)

- **Replication:** In certain cases, the server mishandled triggers and stored procedures that tried to modify other tables when called by `CREATE TABLE ... SELECT`. This is now handled correctly as an error. (Bug #18137535)

- **Replication:** The `REPLICATE_WILD_DO_TABLE` and `REPLICATE_WILD_IGNORE_TABLE` options for `CHANGE REPLICATION FILTER` could be employed with values which contained no period (`.`) character, even though values for these options must be in the format *`db_name.tbl_name`*.. (Bug #18095449)

- **Replication:** When used on a table employing a transactional storage engine, a failed `TRUNCATE TABLE` was still written to the binary log and thus replayed on the slave. This could lead to inconsistency when the master retained data that was removed on the slave.

  Now in such cases `TRUNCATE TABLE` is logged only when it executes successfully. (Bug #17942050, Bug #71070)

- **Replication:** Beginning in MySQL 5.6.20, when a user specified `AUTO_INCREMENT` value falls outside of the range between the current `AUTO_INCREMENT` value and the sum of the current and number of rows affected values it is replicated correctly. In previous versions, an error was generated by the slave even if the user specified `AUTO_INCREMENT` value fell outside of the range. (Bug #17588419, Bug #70583)

- **Replication:** When the I/O thread reconnected to a master using GTIDs and multithreaded slaves while in the middle of a transaction, it failed to abort the transaction, leaving a partial transaction in the relay log, and then retrieving the same transaction again. This occurred when performing a rotation of the relay log. Now when reconnecting, the server checks before rotating the log in such cases, and waits first for any ongoing transaction to complete. (Bug #17326020)

- **Replication:** On Windows, `mysqldump` failed if the error log file was deleted (missing) from the active MySQL server. (Bug #17076131)

- **Replication:** When the binary log was rotated due to receipt of a `SIGHUP` signal, the new binary log did not contain the `Previous_gtid_event` required for subsequent processing of that binary log's GTID events. Now when `SIGHUP` is received, steps are taken to insure that the server writes the necessary `Previous_gtid_event` to the new log before writing any GTID events to the new log. (Bug #17026898)

- **Replication:** The `CLIENT_REMEMBER_OPTIONS` flag for compressed slave connections is no longer reset and all options are retained. This restores functionality of all options to compressed slave connections. (Bug #72901, Bug #18923691, Bug #73324, Bug #19244772)

- **Replication:** When using row-based replication, setting a slave's `slave_rows_search_algorithms` variable to `HASH_SCAN` caused an `ER_KEY_NOT_FOUND` error even though that record existed in the storage layer. This fix ensures that the unique key for each record is correctly maintained and such a situation does not occur. (Bug #72788, Bug #18860225)

- **Replication:** When using row-based replication, running a long transaction involving a large number of events could trigger an Out of Memory (OOM) error if the slave's table structure was not compatible with the master's table structure. Such an incompatible situation could occur if the table on the slave had been manually changed, or when replicating between different MySQL versions that have different data types. This OOM error was caused because the virtual temporary tables created for the row conversion were not being freed until the end of the transaction, which was a problem when replicating large numbers of events.

  Starting with this version, such virtual tables are correctly freed during the conversion process. (Bug #72610, Bug #18770469)

  References: See also: Bug #19692387.

- **Replication:** The error messages generated when a duplicate server UUID causes issues during replication have been improved. The slave error now identifies the duplicate server UUID and the master error identifies the zombie thread that has been killed. (Bug #72578, Bug #18731211)

- **Replication:** When an event group was spanned across multiple relay log files, a slave could incorrectly identify GTID-header group boundaries. This meant that when a transaction was retried, or if the SQL thread was stopped in the middle of a transaction after some rotates, the `Gtid_log_event` was being silently skipped on the slave, and the transaction was logged with the slave's GTID. This problem also impacted on using `START SLAVE UNTIL MASTER_LOG_POS = log_pos;` with GTIDs enabled. If `log_pos` was in the middle of a transaction, the `Gtid_log_event` was not correctly detected as the beginning of the transaction and replication stopped before this event. With this fix, threads correctly detect that they are part of a group, and this is used to check if a `Gtid_log_event` is part of a transaction. (Bug #72313, Bug #18652178, Bug #18306199)

- **Replication:** When `gtid_mode=ON`, and a transaction is filtered out on the slave, the GTID of the transaction is still logged on the slave as an “empty” transaction (consisting of a GTID followed immediately by `BEGIN` and then `COMMIT`). This is necessary to prevent the transaction from being retransmitted the next time the slave reconnects or is involved in a failover. The current fix addresses two issues relating to such “empty” transactions:

  - No empty transaction was generated for `CREATE TEMPORARY TABLE` or `DROP TEMPORARY TABLE` statements.

  - If the slave used a database filter (`--replicate-do-db` or `--replicate-ignore-db` option), no empty transaction was generated.

  (Bug #71376, Bug #18095502, Bug #18145032)

- **Replication:** Client applications should be able to set the `BINLOG_DUMP_NON_BLOCK` flag in the initial handshake packet (`COM_BINLOG_DUMP`). Clients connecting to a server issuing a `COM_BINLOG_DUMP` with the flag unset do not get an EOF when the server has sent the last event in the binary log, which causes the connection to block. This flag, which was removed in error in MySQL 5.6.5, is now restored in the current release.

  As part of this fix, a new `--connection-server-id` option is added to `mysqlbinlog`. This option can be used by the client to test a MySQL server for the presence of this issue. (Bug #71178, Bug #18000079)

- **Replication:** On a master that is using semisynchronous replication, where `rpl_semi_sync_master_wait_no_slave` is enabled and `rpl_semi_sync_master_timeout` is set to long timeout, killing the I/O thread could cause the server to hang on shutdown. This fix ensures that if the dump thread finds that there no semisynchronous slaves connected to the master, the setting of `rpl_semi_sync_master_wait_no_slave` is ignored and the shutdown proceeds correctly. (Bug #71047, Bug #17879675)

- **Replication:** When using semisynchronous replication, if the binary log position was changed to a future position on a slave then an assertion error was generated on the master. This fix ensures that in such a situation the future position is correctly acknowledged and an error is instead generated on the slave. (Bug #70327, Bug #17453826)

- **Replication:** Replication of tables that contained temporal type fields (such as `TIMESTAMP`, `DATETIME`, and `TIME`) from different MySQL versions failed due to incompatible `TIMESTAMP` types. The fractional `TIMESTAMP` format added in MySQL 5.6.4 was not being correctly converted. You can now replicate a `TIMESTAMP` in either format correctly according to the `slave_type_conversions` variable. (Bug #70124, Bug #17532932)

- **Replication:** A group of threads involved in acquiring locks could deadlock when the following events occurred:

  1. Dump thread reconnects from slave; on master, a new dump thread tries to kill zombie dump threads; having acquired the thread's `LOCK_thd_data`, it is about to acquire `LOCK_log`.

  2. Application thread executing show binary logs, having acquired `LOCK_log` and about to acquire `LOCK_index`.

  3. Application thread executing `PURGE BINARY LOGS`; having acquired `LOCK_index`, it is about to acquire `LOCK_thread_count`.

  4. Application thread executing `SHOW PROCESSLIST` (or `SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST`), having acquired `LOCK_thread_count` and about to acquire the zombie dump thread's `LOCK_thd_data`.

  This leads to the 4 threads deadlocking in the same order which the threads have been listed here.

  This problem arises because there are ordering rules for `LOCK_log` and `LOCK_index`, as well as rules for ordering `LOCK_thread_count` and `LOCK_thd_data`, but there are no rules for ordering across these two sets of locks. This was because the internal `mysqld_list_processes()` function invoked by `SHOW PROCESSLIST` acquired `LOCK_thread_count` for the complete lifetime of the function as well as acquiring and releasing each thread's `LOCK_thd_data`. Now this function takes a copy of the threads from the global thread list and performs its traversal on these, and only after releasing `LOCK_thread_count`. During this traversal, removal from the global thread list is blocked using `LOCK_thd_remove` such that the copies that would otherwise be destroyed by the removal remain valid during traversal. The locking order following this fix is shown here:

```
LOCK_thd_remove -> LOCK_thd_data -> LOCK_log -> LOCK_index -> LOCK_thread_count
```

(Bug #69954, Bug #17283409)

References: See also: Bug #73475, Bug #19364731, Bug #19365180.

- **Replication:** When an SQL thread which was waiting for a commit lock was killed and restarted it caused a transaction to be skipped on slave. This fix ensures that thread positions are correctly persisted and transactions resume at the correct position. (Bug #69873, Bug #17450876)
- With DTrace support enabled, certain other compilation options could cause the build to fail. (Bug #19506247)
- yaSSL client code did not validate the encryption size or session ID length, which could cause the client to exit. (Bug #19463277, Bug #19463565)
- Before fsync operations, the server reports a wait, but it reported a table-lock wait rather than a sync wait. This is a minor problem, possibly resulting in come incorrect thread pool plugin statistics. (Bug #19428231)
- The optimizer could raise an assertion due to incorrectly handling the table map while sorting an internal temporary table. (Bug #19416826)
- yaSSL could fail preauthorization if the client supplied inaccurate buffer lengths. (Bug #19370676, Bug #19355577)
- Using Multi-Range Read (MRR) for table access could leak Performance Schema instrumented table handles. (Bug #19301539)
- For `InnoDB` full-text searches, invalid multibyte `gb10830` characters could cause a server exit. (Bug #19233075)
- Competition between threads could lead to timeout failure trying to rotate the audit log file. (Bug #19184973)
- On Windows, `sql_yacc.cc` and `sql_yacc.h` were generated twice during the build process. This can lead to compilation failure if parallel processes are permitted. (Bug #19060850)
- Repetitive functions called as arguments to `ExtractValue()` were not always handled correctly. (Bug #19051306)
- For large sets, `EXPORT_SET()` could produce undefined results or a cause a server exit. (Bug #19048609)
- Invalid `utf16` data could cause the server to become unresponsive. (Bug #18937504)
- `LPAD()` and `RPAD()` could cause a server exit if the pad string argument was not well formed. (Bug #18935421)
- The optimizer could create a zero-length column for a temporary table, causing a server exit. (Bug #18928848)
- In debug builds, certain range queries could cause an assertion failure. (Bug #18921626)
- Improper copying of query strings between the `Prepared_statement` and `THD` data structures could result in a server exit. (Bug #18920075)

- Performance Schema memory instrumentation did not honor the `ENABLED` flag in the `setup_instruments` table or the consumers in the `setup_consumers` table. This has been corrected, with the result that unnecessary statistics are not collected and overhead is reduced. (Bug #18900309)
- When a `SELECT` included a derived table in a join in its `FROM` list and the `SELECT` list included `COUNT(DISTINCT)`, the `COUNT()` returned 1 even if the underlying result set was empty. (Bug #18853696)

  References: This issue is a regression of: Bug #11760197.
- Conversion failure of “zero” dates in strict SQL mode could cause a server exit. (Bug #18840123)
- Modulo operations on `DECIMAL` values in some cases could overflow and cause a server exit. (Bug #18839617)
- Enabling optimizer trace could cause a server exit for queries with a subquery in a `HAVING` clause. (Bug #18791851)
- SHA and MD5 functions failed for operations using the internal `filename` character set and could cause a server exit. (Bug #18786138)
- Large arguments passed to `mysqldump` could lead to buffer overflow and program exit. (Bug #18779944)
- If a materialized subquery read from a view, and contained an inner subquery having an outer reference to a column of the view, results could be incorrect. (Bug #18770217)
- Passing bad arguments to `SHA2()` could cause a server exit. (Bug #18767104)
- Spatial operations on `InnoDB` tables could fail attempting to access nonexistent index statistics. (Bug #18743725)
- `ORDER BY` of a GIS function that was given invalid arguments could cause a server exit. (Bug #18701868)
- The code for processing the `gb18030` character set had a too-strict assertion for single-character invalid characters. (Bug #18700668)
- After a metadata change, a reprepared trigger could cause a server exit or prune an incorrect partition. (Bug #18684393)
- `mysql_secure_installation` ignored options defined after an unrecognized option. (Bug #18659533)
- `VALIDATE_PASSWORD_STRENGTH()` could enter an infinite loop for some arguments. (Bug #18636291)
- `ALTER TABLE` on a partitioned table could result in the wrong storage engine being written into the table's `.frm` file and displayed in `SHOW CREATE TABLE`. (Bug #18618561)
- The server could fail to properly reprepare triggers that referred to another table after that table was truncated. (Bug #18596756, Bug #72446, Bug #18665853)
- Compiler flags were not passed to DTrace, causing problems for 32-bit builds cross-compiled on 64-bit platforms. (Bug #18593044)
- For conditions on the form `t.key NOT IN (c1, c2, ...)`, if one or more of the `c1`, `c2` was `NULL`, the optimizer generated incorrect range predicates, possibly yielding incorrect results.

The range optimizer would build predicates for empty in-lists (because `NULL` values are removed from `NOT IN (in-list)`). (Bug #18556403, Bug #18715670)

- After conversion of an in-memory internal temporary table to disk, the server could use a pointer to the old table, resulting in a server exit. (Bug #18535226)
- For queries executed using Loose Index Scan, incorrect cost estimates could be produced if index statistics were unavailable. (Bug #18497308)
- Setting `default_storage_engine` to a bad value could cause server failure later. (Bug #18487724)
- In debug builds, `MAKE_SET()` within a `GROUP BY` clause raised an assertion. (Bug #18487060)
- In debug builds, subquery optimization could be overly aggressive about raising an assertion. (Bug #18486607)
- In debug builds, a qsort operation on decimal values could raise an assertion. (Bug #18486249)
- For debug builds, an assertion was raised for attempts using a cursor within a stored routine to fetch a large value (`INT`) which cannot fit into a variable (`TINYINT`). (Bug #18484649)
- `MOD` for very small decimal right-hand arguments could cause a server exit. (Bug #18469276)
- With the `max_heap_table_size` system variable set to a large value (20GB), creation of a temporary table or a table using the `MEMORY` storage engine caused a server exit. (Bug #18463911)
- `mysql_session_track_get_first()` and `mysql_session_track_get_next()` returned information only when the `type` argument was `SESSION_TRACK_SYSTEM_VARIABLES`, and returned information for all types, not just `SESSION_TRACK_SYSTEM_VARIABLES`. Now they return information of the type requested and only that type. (Bug #18463441)
- For `mysql_upgrade`, specifying the `--defaults-extra-file` with a nonexisting file caused a segmentation fault on some platforms. (Bug #18443096)
- Assigning some values to the `session_track_system_variables` system variable could cause a server exit. (Bug #18405677)
- `mysql_install_db` could hang while reading `/dev/random` to generate a random `root` password. (Bug #18395378)
- `EXPLAIN FOR CONNECTION` showed an incorrect filtering value for dynamic range queries. (Bug #18395059)
- The client library now includes a call to `X509_verify_cert_error_string()` in the SSL certificate verification code, to be more robust in detecting invalid certificates. (Bug #18384260)
- Queries that use `AES_ENCRYPT()`, `AES_DECRYPT()`, or `RANDOM_BYTES()` cannot be cached but were permitted to enter the query cache. (Bug #18383169)
- If MySQL was built with the `-DINSTALL_LIBDIR=lib64` option, `mysql_config` did not work if the MySQL package was unpacked into a location with a different installation prefix. Also, `mysql_config` did not work for some RPM builds because it used an incorrect installation prefix. (Bug #18382225)
- Plugin registration code in the embedded server (compiled without the Performance Schema) failed for plugins compiled with the Performance Schema. (Bug #18363910)
- For indexes on prefixes or character string columns, index corruption could occur for assignment of binary data to the column due to improper character counting. (Bug #18359924)

- The `HAVE_IB_LINUX_FUTEX` `CMake` check failed due to a missing include file. (Bug #18353271)
- After a code reorganization in MySQL 5.7.4, `ORDER BY` for multiple-table `UPDATE` statements was ignored. (Bug #18352634)
- The addition in MySQL 5.7.4 of session state information to the OK packet of the client/server protocol caused the `mysql->info` member to be missing a terminating null terminator. (Bug #18349102)
- Improper linking of join caches by the optimizer could lead to a server exit. (Bug #18335908)
- In debug builds, lack of proper object initialization of decimal objects caused an assertion to be raised. (Bug #18335446)
- `UNCOMPRESSED_LENGTH()` could return `NULL` when it should not have. (Bug #18335269)
- On Windows, some test cases ran too slowly due to `mysqltest` not testing properly for server termination. (Bug #18330694)
- The `COM_RESET_CONNECTION` command did not reset some session system variables: `rand_seed1`, `rand_seed2`, `timestamp`. Also, it did not clear warnings, and, although it reset the `profiling` variable, it did not reset profiling information. (Bug #18329348, Bug #18329560, Bug #18328396, Bug #18329452)
- Certain `INFORMATION_SCHEMA` queries could cause a server exit. (Bug #18319790)
- `EXPLAIN` for some full-text queries could raise an assertion. (Bug #18313651)
- Solaris-specific scripts were included in and installed by non-Solaris packages. (Bug #18305641)
- `DELETE` is not allowed on views defined as joins on the base tables. However, MySQL incorrectly permitted `REPLACE` (which is like a combination of `DELETE` and `INSERT`) on such views. (Bug #18286777)
- For debug builds, a `0x00` character in a full-text query string that used the `ujis_japanese_ci`, `utf8mb4_turkish_ci`, or `eucjpms_bin` collation could raise an assertion. (Bug #18277305)
- `innobase_strnxfrm()` wrote one byte too many. (Bug #18277082)
- On Windows, use of the `gb18030_unicode_520_ci` caused a server exit. (Bug #18271429)
- The internal `likely()` and `unlikely()` macros conflicted with Boost symbols. To resolve this, the macros were converted to inline functions. (Bug #18242233)
- If the left-hand-side of an `IN` predicate was a scalar subquery but returned no row, the server could exit. (Bug #18223655, Bug #18447874)
- For queries involving an `AND` of two geometry ranges, the optimizer could decide no index was usable but try to use it anyway, resulting in a server exit. (Bug #18220153)
- Argument checking for a `memmove()` call in `my_net_read()` was improved. (Bug #18184793)
- `mysql_upgrade` invoked the `mysql` and `mysqlcheck` clients and the command arguments could be visible to other processes. This issue is addressed by the reimplementation of `mysql_upgrade` to communicate directly with the server, a change described elsewhere in these release notes. (Bug #18180398)
- yaSSL code had an off-by-one error in certificate decoding that could cause buffer overflow.

  yaSSL code had an `opendir()` without a corresponding `closedir()`. (Bug #18178997, Bug #17201924)

- `EXPLAIN` on a query with an `EXISTS` subquery containing a `UNION` could cause a server exit. Multiple executions of a prepared `EXPLAIN` on a `UNION` of subqueries could cause a server exit. (Bug #18167356)
- `mysqladmin password` masked the old password given on the command line, but not the new password. (Bug #18163964)
- For `InnoDB` tables, boolean full-text queries for terms ending with `*` could return incorrect results. (Bug #18128757)
- For XA transactions, -1 could be assigned as the format ID part of an XID value, resulting in mishandling (server hang or exit) of concurrent XA statements. (Bug #18107853)
- The client library could cause clients to exit due to incorrectly mapping the client error number to the corresponding message, if reallocation of packet buffer memory occurred. (Bug #18080920)
- For full-text queries on `InnoDB` tables, attempts to access deleted document IDs could lead to a server exit. (Bug #18079671)
- The optimizer could compute a negative cost value when constructing an execution plan for `MERGE` table queries. (Bug #18066518)
- Executing a correlated subquery on an `ARCHIVE` table which has an `AUTO_INCREMENT` column caused the server to hang. (Bug #18065452)
- Calling `mysql_get_server_version()` with an invalid connection handler argument caused the client to exit. Now it returns 0 and reports a `CR_COMMANDS_OUT_OF_SYNC` error. (Bug #18053212)
- `MyISAM` temporary files could be used to mount a code-execution attack. (Bug #18045646)
- If the optimizer chose to perform an index scan, in some cases it could choose a noncovering rather than a covering index. (Bug #18035906)
- The `CMake -DWITHOUT_PARTITION_STORAGE_ENGINE=1` option did not work. As part of fixing this problem, a preferred syntax for disabling storage engines was implmented. The syntax `-DWITH_`*`engine`*`_STORAGE_ENGINE=0` is now preferred to `-DWITHOUT_`*`engine`*`_STORAGE_ENGINE=1`. For example, use:

  ```
  -DWITH_EXAMPLE_STORAGE_ENGINE=0
  -DWITH_FEDERATED_STORAGE_ENGINE=0
  -DWITH_PARTITION_STORAGE_ENGINE=0
  ```

  Rather than:

  ```
  -DWITHOUT_EXAMPLE_STORAGE_ENGINE=1
  -DWITHOUT_FEDERATED_STORAGE_ENGINE=1
  -DWITHOUT_PARTITION_STORAGE_ENGINE=1
  ```

  (Bug #17947926)
- The C client library could leak memory when client plugins were used. (Bug #17933308)
- For debug builds, `DROP FUNCTION` with a too-long function name raised an assertion. (Bug #17903490)
- A `BEFORE UPDATE` trigger could insert `NULL` into a `NOT NULL` column. (Bug #17864349)
- A `UNION ALL` query with `SQL_CALC_FOUND_ROWS` and a `LIMIT` with an offset for one query block reported an incorrect number of found rows. (Bug #17833261)

- For debug builds, a `DELETE` statement with a subquery that returned multiple rows in a context requiring a scalar subquery could raise a diagnostic area assertion. (Bug #17787664)
- Using an outer reference in a `GROUP BY` or `ORDER BY` clause in a subquery could cause a server exit. (Bug #17748273)
- For queries that selected from the `events_statements_current` Performance Schema table, adding an `ORDER BY` clause could produce incorrect results. (Bug #17729044)
- A (rare) deadlock could occur between `LOCK_thd_data` and the `InnoDB` `trx_sys` mutex. One thread could read a query string while it was being removed by another thread. (Bug #17606098)
- On Windows, calling `mysql_thread_init()` call without `mysql_init()` caused the client to exit. windows. Now it returns a nonzero result because it is an error to call `mysql_thread_init()` before the client library is initialized with `mysql_library_init()`. (Bug #17514920)
- For `REPLACE` on a view, a row being replaced in a view might conflict with one or more rows in the base table, some of which might not be accessible by the view. In some cases, rows inaccessible by the view could be replaced. (Bug #17487701)
- `mysqldump` could create table definitions in the dump file that resulted in `Too many columns` errors when reloading the dump file. (Bug #17477959)
- Using `LOAD DATA INFILE` to load fixed-length data into a view could cause a server exit. (Bug #17458965)
- The optimizer trace could cause a server exit in cases where a subquery was transformed away. (Bug #17458054)
- `UPDATE` statements that modified full-text indexes could cause a server exit. (Bug #17457755)
- Sending a `SIGQUIT` or `SIGINT` signal to `mysql` could result in a `glibc` double free or corruption error. (Bug #17297324)
- A server could have its socket file taken over by a second server listening on different TCP/IP port but the same socket file. The socket file also would be unlinked by normal shutdown of the second server. To avoid this, the server now uses a lock file with the same name as the socket file and a `.lock` suffix. (For example, `/tmp/mysql.sock` has a lock file of `/tmp/mysql.sock.lock`.) The lock file contains the process ID of the server process that created the socket file. (Bug #17286856)
- If a query had both `MIN()/MAX()` and *`aggregate_function`*`(DISTINCT)` (for example, `SUM(DISTINCT)`) and was executed using Loose Index Scan, the result values of `MIN()/MAX()` were set improperly. (Bug #17217128)
- For `UNION` statements, the rows-examined value was calculated incorrectly. This was manifest as too-large values for the `ROWS_EXAMINED` column of Performance Schema statement tables (such as `events_statements_current`). (Bug #17059925)
- When joining one large table without indexes to a number of smaller tables with indexes, the optimizer chose to join on the large table as the last table, causing a large number of large table scans. (Bug #16838146)
- Row constructor arguments to `INTERVAL()` could cause a server exit. (Bug #16439419)
- An assertion could be raised when creating a index on a prefix of a `TINYBLOB` or `GEOMETRY` column in an `InnoDB` column. (Bug #16368875, Bug #18776592, Bug #17665767)
- `mysql_config_editor` left some files open when they were no longer needed, resulting in Valgrind warnings. (Bug #16368498)

- Several issues related to the `ONLY_FULL_GROUP_BY` SQL mode were corrected:

  - With `ONLY_FULL_GROUP_BY` enabled, some valid queries were rejected if the accessed table was replaced by a view.

  - Queries of the form `SELECT DISTINCT col1 ... ORDER BY col2` qualify as forbidden by SQL2003 (hidden `ORDER BY` columns combined with `DISTINCT`), but were not rejected with the `ONLY_FULL_GROUP_BY` SQL mode enabled.

  (Bug #16021396, Bug #18993257, Bug #13581713)

- The change made in MySQL 5.7.0 to display the XID value in hexadecimal for `XA RECOVER` if it contained nonprintable characters was reverted because it caused problems for some clients. Now the statement takes an optional `CONVERT XID` keyword so that clients can request the XID value in hexadecimal on demand. (Bug #14670465)

- To help with crash analysis on Windows, more information has been added to the core file that is written on `mysqld` abort when the `core-file` option is enabled. (Bug #14041454)

- Use of a nonmultibyte algorithm for skipping leading spaces in multibyte strings could cause a server exit. (Bug #12368495, Bug #18315770)

- Executing a `DELETE` statement on a table with no key in safe-update mode resulted in an `ER_UPDATE_WITHOUT_KEY_IN_SAFE_MODE` error. For debug builds, using `IGNORE` in the the statement resulted in an assertion being raised. (Bug #74493, Bug #19873291)

- Binary MySQL distributions for OS X 10.8 and up now bundle the `MySQL.prefPane` and `MySQLStartupItem.pkg` tools into the main package as configurable options instead of separate packages. (Bug #74123, Bug #19701502)

- For debug builds, a `SELECT` under load from the `events_statements_current` Performance Schema table could raise an assertion. (Bug #73530, Bug #19419463)

- `mysql_secure_installation` failed to run properly if the `root` account authentication plugin was `sha256_password`. (Bug #73148, Bug #19127636)

- The server did not take the `lower_case_table_names` value into account in determining the database directory from which to read the `db.opts` file, and thus could read the file from an incorrect directory. (Bug #72900, Bug #18923685)

- `SHA2()` failed to return `NULL` if the hash-length argument was `NULL` or not one of the permitted values. (Bug #72856, Bug #18899869)

- `mysql_install_db` failed to create the initial `root` account if `autocommit` was disabled. (Bug #72724, Bug #18911807)

- The `mysql` client displayed `gb18030` data incorrectly. (Bug #72573, Bug #18726196)

- The `gb18030_chinese_ci` collation treated `'Y'` equal to `'~'`. (Bug #72565, Bug #18729428)

- The query cache was not invalidated for a table when a `CASCADE DELETE` or `CASCADE UPDATE` referential constraint was specified and the database name or table name contained special characters. (Bug #72547, Bug #18710853)

- `NOW(N)` in a view definition was stored as `NOW()`, thus losing the fractional seconds part. (Bug #72472, Bug #18675237)

- If a prepared statement being executed produced an error, the server failed to write the statement to the general query log. (Bug #72375, Bug #18616826)

- A new `CMake` option, `SUNPRO_CXX_LIBRARY`, enables linking against `libCstd` instead of `stlport4` on Solaris 10 or later. This works only for client code because the server depends on C++98. Example usage:

```
cmake -DWITHOUT_SERVER=1 -DSUNPRO_CXX_LIBRARY=Cstd
```

  (Bug #72352, Bug #18605389)

- A `SELECT` statement using a subquery with `UNION` and `ORDER BY` did not permit use of an alias in the outer statement. (Bug #72189, Bug #18498344)

- `UNION` queries with an aggregate function in an `ORDER BY` clause were not rejected as they should be. Now such queries are rejected with an `ER_AGGREGATE_ORDER_FOR_UNION` error. Example:

```
SELECT 1 AS foo UNION SELECT 2 ORDER BY MAX(1);
```

  A nonaggregated query with an `ORDER BY` applied to it cannot contain aggregate functions, but was not rejected as it should be. Now such queries are rejected with an `ER_AGGREGATE_ORDER_NON_AGG_QUERY` error. Example:

```
SELECT a FROM t1 ORDER BY COUNT(*);
```

  (Bug #72174, Bug #18503515, Bug #72512, Bug #18694751)

- `MOD` operations on a `DECIMAL` value with leading zeros could produce incorrect results. (Bug #72100, Bug #18509896)

- `mysqlslap` accepted an `--iterations` option value of 0, which resulted in a divide-by-zero error. The minimum option value now is 1. Thanks to Tsubasa Tanaka for the patch. (Bug #72082, Bug #18430704)

- `mysql_upgrade` could fail if the `mysql.user` table contained multiple accounts with the same user name and host name where the host name differed in lettercase. This is still not permitted, but now `mysql_upgrade` prints a more informative error message to indicate the nature of the problem:

```
ERROR 1644 (45000): Multiple accounts exist for user_name, host_name
that differ only in Host lettercase; remove all except one of them
```

  (Bug #72066, Bug #18415196)

- `ER_CANT_CREATE_GEOMETRY_OBJECT` was treated as a fatal error, thus not catchable with condition handlers. (Bug #72064, Bug #18413646)

- Some comparisons between `BIGINT` signed and unsigned values could yield incorrect results. (Bug #72046, Bug #18408499)

- For `IN()` predicates with values different from the key data value, the optimizer sometimes used a table scan when it could do a range scan. (Bug #71962, Bug #18364815)

- `mysql_config_editor` exited when given an empty argument to the `--login-path` option. (Bug #71837, Bug #18311024, Bug #18830493)

- Upgrades using RPM packages could change the ownership of an installation directory. (Bug #71715, Bug #18281535)

- The `threads` Performance Schema table displayed a `PROCESS_ID` value of `NULL` for replication threads. Now it displays the same ID as `SHOW PROCESSLIST` and the `INFORMATION_SCHEMA.PROCESSLIST` table. (Bug #71682, Bug #18259356)
- In the `DIGEST_TEXT` column of Performance Schema statement events tables, references to system variables of the form `@@var_name` were stored as `@ @ var_name`. (Bug #71634, Bug #18304086)
- For `mysqldump`, dump and restore operations failed for database names that contained backslash (`'\'`). Thanks for Xiaobin Lin for the patch. (Bug #71437, Bug #18109728)
- A simultaneous `OPTIMIZE TABLE` and online `ALTER TABLE` on the same `InnoDB` table could result in deadlock. (Bug #71433, Bug #18110156)
- `XA START` after a server restart with the exising XID followed by `XA COMMIT` failed to commit. (Bug #71352, Bug #18068253)
- Proxy users were unable to execute statements if the proxied user password had expired. (Bug #71337, Bug #18057562)
- MySQL did not compile with Bison 3. (Bug #71250, Bug #18017820, Bug #18978946)
- Deadlock could occur if three threads simultaneously performed `INSTALL PLUGIN`, `SHOW VARIABLES`, and `mysql_change_user()`. (Bug #71236, Bug #18008907, Bug #72870, Bug #18903155)
- A statement of the following form broke row-based replication because it created a table having a field of data type `BIGINT` with a display width of 3000, which is beyond the maximum acceptable value of 255:

  ```
  CREATE TABLE t1 AS SELECT REPEAT('A',1000) DIV 1 AS a;
  ```

  (Bug #71179, Bug #17994219)
- When MySQL runs as service on Windows, `NTService.Stop()` initiates shutdown and exit events during shutdown. After a code reorganization in MySQL 5.7.3, a call to `clean_up()` was missed, resulting in initiation of crash recovery. (Bug #71104, Bug #17980260)
- If there was a predicate on a column referenced by `MIN()` or `MAX()` and that predicate was not present in all the disjunctions on key parts earlier in the compound index, Loose Index Scan returned an incorrect result. (Bug #71097, Bug #17909656)
- Invalid memory access could occur when using prepared statements if a `mysql` client connection was lost after statement preparation was complete and there was at least one statement that was in initialized state but not prepared yet. (Bug #70429, Bug #17512527)
- Uninstalling and reinstalling semisynchronous replication plugins while semisynchronous replication was active caused replication failures. The plugins now check whether they can be uninstalled and produce an error if semisynchronous replication is active. To uninstall the master-side plugin, there must be no semisynchronous slaves. To uninstall the slave-side plugin, there must be no semisynchronous I/O threads running. (Bug #70391, Bug #17638477)
- Client auto-reconnect did not work for clients linked against `libmysqlclient`, even with `MYSQL_OPT_RECONNECT` enabled.

  Also, if a `FEDERATED` table was accessed after `wait_timeout` expired, a `Lost connection to MySQL server` error occurred without an attempt to re-establish the connection. (Bug #70026, Bug #17309863, Bug #14874, Bug #11745408)
- Full-text queries on `MyISAM` tables that included a `LIMIT` clause but no `WHERE` clause could return too few rows. (Bug #69908, Bug #17261347)

- Updates could fail to update all applicable rows in cases where multiple key values were identical except for trailing spaces. (Bug #69684, Bug #17156940)
- On Windows, `REPAIR TABLE` and `OPTIMIZE TABLE` failed for `MyISAM` tables with `.MYD` files larger than 4GB. (Bug #69683, Bug #17235179)
- For logging of prepared statements to the general query log, the `Execute` line was logged after statement execution, not before. (Bug #69453, Bug #16953758, Bug #20536590)
- Calls to `UNCOMPRESS()` produced Valgrind warnings during verification of the zip header of the compressed data. (Bug #69202, Bug #18693654)
- `mysql_tzinfo_to_sql` mishandled some values from the abbreviation list (read from the timezone information file) due to failure to account for the null character appended to the end of the char array. (Bug #68861, Bug #16664462)
- Some statements could be written to the slow query log twice. (Bug #68626, Bug #16467055)
- File permissions and line endings of several test and configuration files were made more consistent to avoid warnings from package checkers. (Bug #68521, Bug #16415173, Bug #16395459, Bug #68517, Bug #16415032, Bug #71112, Bug #17919313, Bug #71113, Bug #17919422)
- In some cases, a successful `CREATE VIEW` could add invalid parentheses to expressions in the view definition. (Bug #67791, Bug #15948263)
- If the general query log or slow query log file was set to a FIFO or socket file, and the file reader went away, the server stopped executing statements. Now the server detects such files, logs an error message, and continues with the appropriate log disabled. (Bug #67088, Bug #14757009)
- For non-debug builds of several client programs, the `--help` message did not correctly indicate that the `--debug`, `--debug-check`, and `--debug-info` apply only for debug builds. (Bug #66854, Bug #16272328)
- Notification of events for the general log were received by the audit log plugin only if the general query log was enabled. Now notifications are posted regardless of whether the general query log is enabled. (Bug #60782, Bug #12368204, Bug #20536590, Bug #75796, Bug #20479643)
- Queries that used `GROUP BY INSERT()` could produce spurious duplicate-key errors. (Bug #58081, Bug #11765149)
- `mysql_install_db` could fail if not invoked in the MySQL installation base directory. (Bug #54034, Bug #11761529)
- With `big_tables` enabled, queries that used `COUNT(DISTINCT)` on a simple join with a constant equality condition on a non-duplicate key returned incorrect results. (Bug #52582, Bug #11760197)

  References: See also: Bug #18853696.
- `LOAD DATA LOCAL INFILE` could use all CPU if import errors occurred when there were no line delimiters. (Bug #51840, Bug #11759519)
- For an existing nondynamic (built-in) plugin, the error message for an attempted `UNINSTALL PLUGIN` was misleading (the plugin does not exist). Now the message indicates that built-in plugins cannot be uninstalled. (Bug #51771, Bug #11759453)
- `LIKE` matches failed for code points of HALF WIDTH KATAKANA in the `sjis` and `cp932` character sets. (Bug #47641, Bug #11755818)

- The server failed to produce an error for `INSERT` statements that provided no column names but did provide column values. (Bug #20943, Bug #11745889, Bug #18064775)

  References: This issue is a regression of: Bug #16820562.

- Connection IDs are 32-bit unsigned integers, beginning at 1. When the server assigns connection IDs and reaches the top of the 32-bit range, it rolls the value over to begin at 1 again. It was possible that the server would assign a connection ID to a new thread while that ID was still in use by an existing thread, if the old thread was particularly long running. For such cases, reference to the ID becomes ambiguous. For example, it cannot reliably be determined for `KILL connection_id` which thread to kill, which could lead to undefined behavior. This behavior has been corrected so that in-use IDs are not reused.

Changes in MySQL 5.7.4 (2014-03-31, Milestone 14)

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.

In Memoriam

This release is dedicated to the memory of two young engineers of the MySQL Engineering family, Astha and Akhila, whom we lost while they were in their early twenties. This is a small remembrance and a way to recognize your contribution to the 5.7 release. You will be missed.

Performance Schema Notes

- Performance Schema performance was improved in the following ways:
  - When a thread connects, reset of all per-thread statistics is now delayed until a statistic is actually collected. This lazy initialization benefits workloads with very short-lived sessions, for which instrumentation is disabled.
  - When a thread disconnects, the per-thread statistics are aggregated to a parent only for statistics that actually collected data. This optimization benefits workloads with very short-lived sessions, for which instrumentation is disabled.
  - For statement instrumentation, reset of an individual `EVENT_NAME` statistic is also now delayed until a statistic is actually collected. This benefits all workloads that contain only a few types of statements (`SELECT`, `INSERT`, `UPDATE`, `DELETE`, and so forth) from the very large set statements supported in MySQL. Only statements for event names actually executed are aggregated on disconnect.
  - The memory footprint of internal memory buffers is reduced, by removing some attributes reserved for future use, that were in fact not used. In particular, statistics for mutexes, rwlocks and conditions now need less memory.

The Performance Schema now instruments prepared statements (for both the binary and text protocols):

- Information about prepared statements is available in the `prepared_statements_instances` table. This table enables inspection of prepared statements used in the server and provides aggregated statistics about them.
- The `performance_schema_max_prepared_statements_instances` system variable controls the size of the table.
- The `Performance_schema_prepared_statements_lost` status variable indicates how many prepared statements could not be instrumented.

For more information, see The prepared_statements_instances 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 into the `performance_schema` database.

References: See also: Bug #18324285.

Security Notes

- **Incompatible Change:** MySQL now enables database administrators to establish a policy for automatic password expiration: Any user who connects to the server using an account for which the password is past its permitted lifetime must change the password. The implementation has these components:
  - The `default_password_lifetime` system variable defines global password expiration policy. A value of *`N`* greater than zero means that passwords have a lifetime of *`N`* days. A value of 0 disables automatic password expiration. The default is 360; passwords must be changed approximately once per year.
  - The `mysql.user` table has new columns that store expiration policy information for individual accounts:
    - `password_last_changed` indicates when the password was last changed. The server uses this column when clients connect to determine whether the password is past its lifetime and must be changed per the expiration policy in effect.
    - `password_lifetime` indicates the account password lifetime. A value of *`N`* greater than zero means that the password has a lifetime of *`N`* days. 0 disables automatic password expiration. `NULL` (the default) means that the global expiration policy applies.
  - The `ALTER USER` statement has new options to set password expiration policy for individual accounts.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate the changes to the `mysql.user` table. For each account, `mysql_upgrade` uses the time at which it executes to set the `password_last_changed` column and sets `password_lifetime` to `NULL`. This causes the default global expiration policy to go into effect (passwords have a lifetime of 360 days).

  For more information, see Password Expiration Policy, and ALTER USER Syntax.

- **Incompatible Change:** MySQL deployments installed using `mysql_install_db` now are secure by default. The following changes have been implemented as the default deployment characteristics:
  - The installation process creates only a single `root` account, `'root'@'localhost'`, automatically generates a random password for this account, and marks the password expired. The MySQL

administrator must connect as `root` using the random password and use `SET PASSWORD` to select a new password. (The random password is found in the `.mysql_secret` file in the home directory of the effective user running the script.)

- Installation creates no anonymous-user accounts.
- Installation creates no `test` database.

Those changes are implemented by modifying the default mode of operation for `mysql_install_db`, which is invoked automatically during RPM installation operations. Therefore, the changes also affect non-RPM installation methods for which you invoke `mysql_install_db` manually.

Because `mysql_install_db` deployment now is secure by default, the `--random-passwords` option is unnecessary and has been removed. It has been replaced by the `--skip-random-passwords` option. You can use this option to explicitly produce a deployment that is *not* secure by default:

- No random password is generated for the `'root'@'localhost'` account.
- A `test` database is created that is accessible by any user.

Note

`mysql_install_db` no longer creates anonymous-user accounts, even with `--skip-random-passwords`.

The `mysql_secure_installation` program now has a `--use-default` option, which causes the program to execute noninteractively. It can be used for unattended install operations.

SQL Mode Notes

- **Incompatible Change**

Note

The change described here to make `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` do nothing when named explicitly and include them in strict mode was reverted in MySQL 5.7.8 (see Changes in MySQL 5.7.8 (2015-08-03)). The intent is still that they be used in conjunction with strict mode, so a warning occurs as of 5.7.8 if they are enabled without also enabling strict mode or vice versa.

The `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` SQL modes now are deprecated and do nothing when named explicitly. Instead, their effects are included in the effects of strict SQL mode (`STRICT_ALL_TABLES` or `STRICT_TRANS_TABLES`). In other words, strict mode now means the same thing as the previous meaning of strict mode plus the `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` modes. This change reduces the number of SQL modes with an effect dependent on strict mode and makes them part of strict mode itself.

To prepare for the SQL mode changes in this version of MySQL, it is advisable *before* upgrading to read SQL Mode Changes in MySQL 5.7. That discussion provides guidelines to assess whether your applications will be affected by these changes.

The `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` SQL mode names are still recognized (so that statements that refer to them do not produce an error), but they will be removed in a future MySQL release. To make advance preparation for versions of MySQL in which these mode names do not exist, applications should be modified not to refer to them.

Functionality Added or Changed

- **Performance; InnoDB:** `InnoDB` now supports multiple page_cleaner threads for flushing dirty pages from buffer pool instances. A new system variable, `innodb_page_cleaners`, is used to specify the number of page_cleaner threads. The default value of `1` maintains the pre-MySQL 5.7.4 configuration in which there is only a single page_cleaner thread. This enhancement builds on work completed in MySQL 5.6.2, which introduced a single page cleaner thread to offload buffer pool flushing work from the `InnoDB` master thread.

- **Incompatible Change:** The `AES_ENCRYPT()` and `AES_DECRYPT()` functions now permit control of the block encryption mode and take an optional initialization vector argument:

  - The new `block_encryption_mode` system variable controls the mode for block-based encryption algorithms. Its default value is `aes-128-ecb`, which signifies encryption using a key length of 128 bits and ECB mode.

  - An optional *`init_vector`* argument provides an initialization vector for encryption modes that require it:

    ```
    AES_ENCRYPT(str,key_str[,init_vector])
    AES_DECRYPT(crypt_str,key_str[,init_vector])
    ```

  - A random string of bytes to use for the initialization vector can be produced by calling the new `RANDOM_BYTES()` function.

  For more information, see Encryption and Compression Functions.

  These changes make statements that use `AES_ENCRYPT()` or `AES_DECRYPT()` unsafe for statement-based replication and they cannot be stored in the query cache. Queries that use `RANDOM_BYTES()` are unsafe for statement-based replication and cannot be stored in the query cache.

- **InnoDB:** `InnoDB` now supports the Transportable Tablespace feature for partitioned `InnoDB` tables and individual `InnoDB` table partitions. This enhancement eases backup procedures for partitioned tables and enables copying of partitioned tables and individual table partitions between MySQL instances. For additional information, see Copying File-Per-Table Tablespaces to Another Server. (Bug #18121824, Bug #70196, Bug #18304194, Bug #71784)

- **InnoDB:** Parameters used to identify an `InnoDB` pages are replaced by two new classes, and `fold` value and `zip_size` value calculations have been optimized. These changes simplify code by reducing the number of page identifier parameters passed to functions. (Bug #18073495)

- **InnoDB:** If system tablespace files (“ibdata files”) are located on Fusion-io devices that support atomic writes, doublewrite buffering is automatically disabled and Fusion-io atomic writes are used for all data files. Because the doublewrite buffer setting is global, doublewrite buffering is also disabled for data files residing on non-Fusion-io hardware. (Bug #18069105)

- **InnoDB:** Reverse index leaf page scan has been optimized. `btr_pcur_restore_position_func()` can now perform optimistic restoration for reverse cursors, which reduces `block->mutex` contention on the root page, especially for concurrent reverse scans. (Bug #17666170)

- **InnoDB:** A new system variable, `innodb_log_write_ahead_size`, allows you to configure the write-ahead block size for redo logs to a value that matches the operating system or file system cache block size in order to avoid “read-on-write” for redo log writes. Read-on-write occurs when redo log blocks are not entirely cached to the operating system or file system due to a mismatch between write-ahead block size for redo logs and operating system or file system cache block size. Avoiding read-on-write improves throughput stability for redo log writes. (Bug #17571371)

- **InnoDB:** MySQL now supports rebuilding regular and partitioned `InnoDB` tables using online DDL (`ALGORITHM=INPLACE`) for the following operations:

  - `OPTIMIZE TABLE`
  - `ALTER TABLE ... FORCE`
  - `ALTER TABLE ... ENGINE=INNODB` (when run on an `InnoDB` table)

    Online DDL support reduces table rebuild time and permits concurrent DML, which helps reduce user application downtime. For additional information, see Overview of Online DDL.

  (Bug #13975225)

- **InnoDB:** The `innodb_use_sys_malloc` and `innodb_additional_mem_pool_size` system variables, which were deprecated in MySQL 5.6.3, are removed in MySQL 5.7.4.

- **InnoDB:** MySQL now includes a `gb18030` character set that supports the China National Standard GB18030 character set. For more information about MySQL character set support, see Character Set Support.

- **InnoDB:** The `InnoDB` Tablespace Monitor and `InnoDB` Table Monitor were removed in MySQL 5.7.4. Table and tablespace metadata can be obtained from `INFORMATION_SCHEMA` tables.

- **InnoDB:** New global configuration parameters, `innodb_status_output` and `innodb_status_output_locks`, allow you to dynamically enable and disable the standard `InnoDB` Monitor and `InnoDB` Lock Monitor for periodic output. Previously, enabling and disabling these monitors for periodic output required creating and dropping specially named tables (`innodb_monitor` and `innodb_lock_monitor`). For additional information, see InnoDB Monitors.

- **Replication:** The binary log dump thread has been optimized by removing unnecessary reallocation of the send buffer. Previously, memory was allocated then freed for every event sent to the slave, even when this was not strictly necessary. Following this optimization, the MySQL Server can make better use of hardware resources by having the dump thread employ adaptative memory allocation, which can also result in less CPU usage. (Bug #11747349, Bug #31932, Bug #11752288, Bug #43426, Bug #13727951)

- **Replication:** It is now possible in many cases to execute `CHANGE MASTER TO` without first issuing `STOP SLAVE`. This capability is added by implementing the following changes in the behavior of the `CHANGE MASTER TO` statement, which now depends only on whether the slave SQL thread or slave I/O thread is stopped, as described here:

  - If the SQL thread is stopped, you can execute `CHANGE MASTER TO` using the `RELAY_LOG_FILE`, `RELAY_LOG_POS`, and `MASTER_DELAY` options, even if the slave I/O thread is running. No other options may be used with this statement when the I/O thread is running.
  - If the I/O thread is stopped, you can execute `CHANGE MASTER TO` using any of the options for this statement *except* `RELAY_LOG_FILE`, `RELAY_LOG_POS`, or `MASTER_DELAY`, even when the SQL thread is running. These three options cannot be used when the I/O thread is running.
  - Both the SQL thread and the I/O thread must be stopped before issuing `CHANGE MASTER TO` with `MASTER_AUTO_POSITION = 1`.

  If you are using statement-based replication and temporary tables, it is possible for a `CHANGE MASTER TO` statement following a `STOP SLAVE` statement to leave behind temporary tables on the slave. As part of this set of improvements, a warning is now issued whenever this occurs. You can avoid this in such cases by making sure that `Slave_open_temp_tables` is equal to 0 prior to executing these statements.

For more information, see CHANGE MASTER TO Syntax, and Switching Masters During Failover.

- **Replication:** Implemented separate threads for sending and receiving semisynchronous replication acknowledgement signals, so that event streams and ACK streams can be sent and received simultaneously. This should reduce many common delays and thus improve performance with semisynchronous replication in a number of settings.

- On Solaris, `mysql_config --libs` now includes `-R/path/to/library` so that libraries can be found at runtime. (Bug #18235669)

- On Windows, `NOMINMAX` is set using the `ADD_DEFINITIONS()` `CMake` macro rather than in `config.h.cmake` so that it is set even without including `my_config.h`. (Bug #18192896)

- `CMake` support for compiling MySQL with `gcc` on Solaris was improved. Binary distributions for Solaris now are built using `gcc` rather than Sun Studio, to enable compilation of code not handled by Sun Studio. The client programs and the client libraries except the embedded library are still built using Sun Studio.

  A consequence of this change is that on Solaris, `mysql_config` no longer provides arguments for linking with the embedded library, since this is now built using `gcc` instead of Sun Studio. To get linking arguments for the embedded library, use the alternative script `mysql_server_config` instead. (Bug #18146422, Bug #17826757)

- The `CHECK_FUNCTION_REPLACEMENT()` `CMake` macro was removed from `Windows.cmake` and replacement functions are set explicitly instead since the result of the check was already hard coded in `WindowsCache.cmake`. (Bug #18116661)

- MySQL now compiles using Clang 3.4. (Bug #18047020)

- In MySQL 5.7.1, the MySQL test suite `mysql-test-run.sh` program was modified to start the server with `InnoDB` rather than `MyISAM` as the default storage engine. All tests in the MySQL test suite were modified to include a `force_default_myisam.inc` file. This had to be done because most legacy test results were recorded with the `MyISAM` engine and failed with a result difference if run with `InnoDB`. A project is underway to migrate these tests and remove `force_default_myisam.inc` for tests that do not need `MyISAM`. In 5.7.4, the `rpl` and `binlog` suites and parts of the main suite were migrated. (Bug #17902011)

- Performance Schema instrumentation was added to capture GTIDs for transaction events. (Bug #17799147)

- Performance Schema overhead was reduced for the `pfs_lock` implementation and the uses of atomic operations in general. (Bug #17766582)

- `CMake` now aborts the configuration process on Windows if a Visual Studio version earlier than 2010 is used. (Bug #17730320)

- A new `CMake` option, `WITH_MSAN`, permits enabling MemorySanitizer for compilers that support it. (Bug #17632319)

- Previously, `ALTER TABLE` in MySQL 5.6 could alter a table such that the result had temporal columns in both 5.5 and 5.6 format. Now `ALTER TABLE` upgrades old temporal columns to 5.6 format for `ADD COLUMN`, `CHANGE COLUMN`, `MODIFY COLUMN`, `ADD INDEX`, and `FORCE` operations. This conversion cannot be done using the `INPLACE` algorithm because the table must be rebuilt, so specifying `ALGORITHM=INPLACE` in these cases results in an error. Specify `ALGORITHM=COPY` if necessary.

When `ALTER TABLE` does produce a temporal-format conversion, it generates a message that can be displayed with `SHOW WARNINGS`: `TIME/TIMESTAMP/DATETIME columns of old format have been upgraded to the new format`. (Bug #17246318)

- The `mysql_version.h` file defines two new macros, `LIBMYSQL_VERSION` and `LIBMYSQL_VERSION_ID`, that indicate the string and numeric forms of the client library version.

  - In the client library included with MySQL Server distributions, these macros have the same values as `MYSQL_SERVER_VERSION` and `MYSQL_VERSION_ID`. For example, in MySQL 5.7.4, `MYSQL_SERVER_VERSION` and `LIBMYSQL_VERSION` are `"5.7.4-m14"`, and `MYSQL_VERSION_ID` and `LIBMYSQL_VERSION_ID` are `50704`.

  - In the client library included with Connector/C distributions, `MYSQL_SERVER_VERSION` and `MYSQL_VERSION_ID` have the values of the MySQL version on which the Connector/C distribution is based, whereas `LIBMYSQL_VERSION` and `LIBMYSQL_VERSION_ID` indicate the Connector/C version. For example, Connector/C 6.1.3 is based on MySQL 5.7.4, so `MYSQL_SERVER_VERSION` and `MYSQL_VERSION_ID` have values of `"5.7.4-m14"` and `50704`, whereas `LIBMYSQL_VERSION` and `LIBMYSQL_VERSION_ID` have values of `"6.1.3"` and `60103`.

  In addition, the `mysql_get_client_info()` and `mysql_get_client_version()` C API functions in the client library now return values that reflect the type of distribution that provides the client library:

  - In MySQL distributions, `mysql_get_client_info()` returns `MYSQL_SERVER_VERSION` and `mysql_get_client_version()` returns `MYSQL_VERSION_ID`. This is the same as before.

  - In Connector/C distributions, `mysql_get_client_info()` returns `LIBMYSQL_VERSION` and `mysql_get_client_version()` returns `LIBMYSQL_VERSION_ID`. Previously, these functions returned the MySQL version, the same as in MySQL distributions.

  (Bug #17171724)

- Overhead was reduced for `filesort` comparison operations. (Bug #14635144)

- Based on community feedback, the default value of 10 for the `eq_range_index_dive_limit` system variable has proven to be too low. The default has been raised to 200. (Bug #70586, Bug #17587952)

- `mysql_install_db` provides a more informative diagnostic message when required Perl modules are missing. (Bug #69844, Bug #18187451)

- `CMake` now supports a `-DTMPDIR=`*`dir_name`* option to specify the default `tmpdir` value. If unspecified, the value defaults to `P_tmpdir` in `<stdio.h>`. Thanks to Honza Horak for the patch. (Bug #68338, Bug #16316074)

- MySQL now supports server-side timeouts for execution of `SELECT` statements:

  - `SELECT` supports a `MAX_STATEMENT_TIME` option to specify a timeout for individual queries. For example:

    ```
    SELECT MAX_STATEMENT_TIME = 5000 id, name FROM my_table WHERE ...
    ```

    The server terminates the statement if its execution exceeds the timeout value.

  - The `max_statement_time` system variable specifies the timeout value for `SELECT` statements executed within the session that include no `MAX_STATEMENT_TIME` option. If the value is 0, timeouts are not enabled.

- The `Max_statement_time_exceeded`, `Max_statement_time_set`, and `Max_statement_time_set_failed` status variables provide information about `SELECT` statements affected by timeouts.

Timeout values are in milliseconds.

For more information, see SELECT Syntax, and Server System Variables.

Thanks to Davi Arnaut for the patch on which this feature is based. (Bug #68252, Bug #16271666)

- Overhead was reduced for metadata lock acquisition for DML statements. (Bug #58627, Bug #11765641)
- Logarithmic functions return `NULL` if the argument is less than or equal to 0.0E0. They now also report a warning “Invalid argument for logarithm”. (Bug #50507, Bug #11758319)
- Code instrumented with Valgrind did not preallocate memory in `alloc_root()`, to help find bugs. This behavior is now also enabled if ASAN (address sanitizer) is used. (Bug #44582, Bug #11753184)
- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.
- `mysqlbug`, an obsolete program for generating bug reports, has been removed from the source code.
- The server can now report session-state changes to client programs. Reportable session state consists of these values:
  - The default schema (database)
  - Session-specific values for system variables
  - User-defined variables
  - Temporary tables
  - Prepared statements

  The MySQL client/server protocol now includes tracker information so that session state changes can be detected. One use for the tracker mechanism is that it provides a means for MySQL Connectors, Fabric, and client applications to determine whether any session context is available to ensure session migration from one server to another. (To change connections in a load-balanced environment, it is necessary to detect whether there is session state to take into consideration when deciding whether a switch can be made.)

  The user interface to control the tracker and retrieve state-change information from it has the following components, which enable implementation of state-change tracking on the client side:

  - Clients can request notification when there is a change to any of the session state-related values just listed, in the form of a flag that is set in the OK packet received from the server after the change occurs. To control notification, enable or disable the `session_track_state_change` system variable. This variable is disabled by default.
  - Clients can request notification of changed values for certain specific types of session state information:

- The default schema name. To control notification, enable or disable the `session_track_schema` system variable. This variable is enabled by default.
- The session values of system variables. Notification occurs for the system variables named by the `session_track_system_variables` system variable. By default, notification is enabled for `time_zone`, `autocommit`, `character_set_client`, `character_set_results`, and `character_set_connection`. (The latter three variables are those affected by `SET NAMES`.)

- To enable applications to extract the state-change information returned by the server, the MySQL C API includes a pair of functions:
  - `mysql_session_track_get_first()` fetches the first state-change information received from the server.
  - `mysql_session_track_get_next()` fetches any remaining state-change information received from the server. Following a successful call to `mysql_session_track_get_first()`, call this function repeatedly as long as it returns success.
  - Because there are new API functions, the client library ABI version is now 18.3. Shared library names now include 18.3 where appropriate.
- The `mysqltest` program has `enable_session_track_info` and `disable_session_track_info` commands to enable and disable tracking of session state-change information.

  For more information, see Server System Variables and mysql_session_track_get_first(). For information about the structure of the OK packet used to convey state-change information, see OK_Packet.

- The metadata locking subsystem now uses a lock-free algorithm for acquiring and releasing locks typical for DML statements. This gives better performance/scalability on multi-core machines in workloads involving lots of small read-only transactions.
- The deprecated `IGNORE` clause for `ALTER TABLE` has been removed and its use produces an error.
- The metadata locking subsystem implementation now uses lock-free hashing rather than a hash protected by a mutex. An implication of this change is that the `metadata_locks_cache_size` and `metadata_locks_hash_instances` system variables no longer have any effect, so they are deprecated and will be removed in a future MySQL release.

Bugs Fixed

- **Incompatible Change:** Old clients (older than MySQL 5.5.7) failed to parse authentication data correctly if the server was started with the `--default-authentication-plugin=sha256_password` option.

  

  Note

  As a result of this bug fix, MySQL 5.6.16 clients cannot connect to a 5.6.17 server using an account that authenticates with the `sha256_password` plugin, nor can 5.6.17 clients connect to a 5.6.16 server. Similarly, MySQL 5.7.3 clients cannot connect to a 5.7.4 server using an account that authenticates with the `sha256_password` plugin.

  (Bug #17495562)

- **Important Change; InnoDB; Partitioning:** The `FLUSH TABLES` statement's `FOR EXPORT` option is now supported for partitioned `InnoDB` tables. (Bug #16943907)

- **InnoDB; Replication:** Attempting to reset a replication slave while `innodb_force_recovery` is greater than `0` would return a cryptic error message: `ERROR(1030) HY000: Got error -1 from storage engine`. The error message has been changed to: `ERROR HY000: Operation not allowed when innodb_force_recovery > 0`. Replication options such as `--relay-log-info-repository=TABLE` and `--master-info-repository=TABLE` store information in tables in `InnoDB`. When `innodb_force_recovery` is greater than 0, replication tables cannot be updated which may cause replication administration commands to fail. (Bug #17287443, Bug #69907)

- **InnoDB; Replication:** An `INSERT ...ON DUPLICATE KEY UPDATE` statement run on a table with multiple unique indexes would sometimes cause events to be incorrectly written to the binary log. (Bug #11758237, Bug #50413)

- **InnoDB; Replication:** Using the `InnoDB memcached` plugin (see InnoDB memcached Plugin) with `innodb_api_enable_binlog` set to 1 caused the server to leak memory. (Bug #70757, Bug #17675622)

- **InnoDB:** A `!sync_check_iterate(check)` assertion occurred in `fts_create_doc_id()` during an `INSERT` operation. (Bug #18253210)

- **InnoDB:** `trx_undo_truncate_start` would write to the redo log for temporary undo log segments, resulting in a purge thread assertion. (Bug #18252937)

- **InnoDB:** Unused parameters and variables along with disabled functionality has been removed from redo log code. This patch also includes redo log code improvements including test enablement, addition of Valgrind instrumentation, and minor code refactoring. (Bug #18251675)

- **InnoDB:** The user defined type, `xid_t`, was defined multiple times. (Bug #18251254)

- **InnoDB:** Doublewrite buffer error messages referenced page numbers in an inconsistent manner. (Bug #18242594)

- **InnoDB:** `InnoDB` would perform unnecessary redo log writes and flushing for temporary tablespaces. (Bug #18236692)

- **InnoDB:** The `truncate_t::drop_indexes` and `truncate_t::create_indexes` methods would disable redo logging when modifying the system tablespace. (Bug #18236000)

- **InnoDB:** For full-text queries, a failure to check that `num_token` is less than `max_proximity_item` could result in an assertion. (Bug #18233051)

- **InnoDB:** An invalid `memmove` in `fts_query_fetch_document` would cause a serious error. (Bug #18229433)

- **InnoDB:** `InnoDB` would write to the redo log for some operations on temporary tablespaces. (Bug #18226934)

- **InnoDB:** `log_mutex_exit` should be called before `log_buffer_extend` when the log buffer mutex is held. (Bug #18202904)

- **InnoDB:** The GCC 4.4 compiler would emit a bogus warnings about `InnoDB` parsing functions, indicating that output is uninitialized. (Bug #18192536)

- **InnoDB:** To simplify code and reduce memory usage, `InnoDB` redo log scan records for file-level operations, previously stored in a hash table, are now processed immediately. (Bug #18180875)

- **InnoDB:** `innodb_ft_result_cache_limit` now has a hardcoded maximum value of 4294967295 bytes or (2**32 -1). The maximum value was previously defined as the maximum value of `ulong`. (Bug #18180057, Bug #71554)

- **InnoDB:** `TRUNCATE TABLE` on Windows would report multiple `'chsize' returned OS error 71` errors. (Bug #18176071, Bug #71173)
- **InnoDB:** An `UPDATE` resulted in a memory access error in `lock_rec_other_trx_holds_expl`. The transaction list (`trx_sys->rw_trx_list`) was traversed without acquiring the transaction subsystem mutex (`trx_sys->mutex`). (Bug #18161853)
- **InnoDB:** `InnoDB` failed to restore a corrupt first page of a system tablespace data file from the doublewrite buffer, resulting in a startup failure. (Bug #18144349, Bug #18058884)
- **InnoDB:** Temporary tablespace file size did not match the file size specified by `--innodb-temp-data-file-path` due to an error in file size allocation logic. (Bug #18141070)
- **InnoDB:** A regression introduced by Bug #14329288 would result in a performance degradation when a compressed table does not fit into memory. (Bug #18124788, Bug #71436)

  References: This issue is a regression of: Bug #14329288.
- **InnoDB:** The loader in some Solaris versions would refuse to start a GCC-compiled binary (such as the `mysqld` binary) that uses the `PAUSE` assembler instruction. (Bug #18122171)
- **InnoDB:** The maximum value for `innodb_thread_sleep_delay` is now 1000000 microseconds. The previous maximum value (4294967295 microseconds on 32-bit and 18446744073709551615 microseconds on 64-bit) was unnecessarily large. Because the maximum value of `innodb_thread_sleep_delay` is limited by the value set for `innodb_adaptive_max_sleep_delay` (when set to a non-zero value), the maximum value for `innodb_thread_sleep_delay` is now the same as the maximum value for `innodb_adaptive_max_sleep_delay`. (Bug #18117322)
- **InnoDB:** The `fil_node_create` function would perform an unnecessary hash table lookup. (Bug #18116588)
- **InnoDB:** `INFORMATION_SCHEMA.INNODB_TRX` contained a bogus transaction ID that did not match transaction ID values printed elsewhere. The method used to retrieve transaction IDs was inconsistent. (Bug #18111007)
- **InnoDB:** When starting the server, unnecessary “checking space” log messages would be printed when processing the doublewrite buffer. (Bug #18101380)
- **InnoDB:** A compilation error (`unable to find string literal operator`) was returned when building in `c++11` mode. (Bug #18082139)
- **InnoDB:** The `UNIV_LOG_DEBUG` debug flag, which is no longer fully functional, has been removed. (Bug #18080537)
- **InnoDB:** In the case of a corrupted clustered index on a temporary table, the server would crash on an `INSERT` instead of returning an error. In the case of a corrupted clustered index on a normal table, an error was not returned and the `INSERT` would succeed. (Bug #18064548)
- **InnoDB:** Specifying an alternate directory for `InnoDB` tables using `datadir` and then moving .ibd files to the default MySQL `data` directory would result in a serious error when attempting a `DROP TABLE` operation. (Bug #18063570)
- **InnoDB:** Attempting to uninstall the `InnoDB memcached` plugin while the `InnoDB memcached` plugin is still initializing would kill the `InnoDB memcached` daemon thread. Uninstall should wait until initialization is complete. (Bug #18038948)
- **InnoDB:** A full-text tokenizer thread would terminate with an incorrect error message. (Bug #18021306)

- **InnoDB:** In debug builds, creating a unique index on a binary column, with input data containing duplicate keys, would cause an assertion. (Bug #18010711)
- **InnoDB:** The `srv_monitor_thread` would crash in the `lock_print_info_summary()` function due to a race condition between the `srv_monitor_thread` and purge coordinator thread. (Bug #17980590, Bug #70430)
- **InnoDB:** A boolean mode full-text search query would result in a memory access violation during parsing. (Bug #17978763)
- **InnoDB:** Logging functions were not used consistently. The bug fix replaces occurrences of `fprintf(stderr, message)` and `fputs()` with `ib_logf()`. Also, because `ib_logf()` emits a timestamp with each message, the bug fix removes unnecessary occurrences of `ut_print_timestamp()`. (Bug #17935793, Bug #17534737)
- **InnoDB:** Due to a parser error, full-text search queries that include a sub-expression could return the wrong result. (Bug #17840768)
- **InnoDB:** On Windows, a regression introduced in 5.7.3 would allow log writes during sync operations, which should not be allowed due to an issue in some Windows environments. (Bug #17824101)
- **InnoDB:** The `innochecksum` tool did not use a Windows-specific API to retrieve file size information, which resulted in an incorrect error message (`Error: ibdata1 cannot be found`) when the MySQL 5.6 `innochecksum` 2GB file size limit was exceeded. `innochecksum` now provides support for files larger than 2GB in both MySQL 5.6 and MySQL 5.7. (Bug #17810862, Bug #70936)
- **InnoDB:** Due to a regression introduced by the fix for Bug#17371537, memory was not allocated for the default memcached engine when using the default memcached engine as the backstore for data instead of `InnoDB`. (Bug #17800829)
- **InnoDB:** A page allocation for an undo log due failed with a “table is full” error message instead of an “undo log is full” error message. (Bug #17779822)
- **InnoDB:** If a crash occurred while temporary tables are active, `InnoDB` would report an invalid error message on restart indicating that a temporary table does not exist in the `InnoDB` internal data dictionary. (Bug #17779729)
- **InnoDB:** `ut_free` could be called more than once in succession. (Bug #17763472)
- **InnoDB:** An index tree modification could result in a deadlock. (Bug #17754767)
- **InnoDB:** A race condition in `DebugPolicy::enter()` would cause a segmentation fault in `sync_array_cell_print`. (Bug #17713784)
- **InnoDB:** Manipulating a table after discarding its tablespace using `ALTER TABLE ... DISCARD TABLESPACE` could result in a serious error. (Bug #17700280)
- **InnoDB:** Persistent optimizer statistics would cause stalls due to latch contention. (Bug #17699331, Bug #70768)
- **InnoDB:** Attempting to add an invalid foreign key when foreign key checking is disabled (`foreign_key_checks=0`) would cause a serious error. (Bug #17666774)
- **InnoDB:** For debug builds, the table rebuilding variant of online `ALTER TABLE`, when run on tables with BLOB columns, would cause an assertion in the `row_log_table_apply_update` function. For normal builds, a `DB_PRODUCTION` error would be returned. (Bug #17661919)
- **InnoDB:** An `InnoDB` full-text search failure would occur due to an “unended” token. The string and string length should be passed for string comparison. (Bug #17659310)

- **InnoDB:** `MATCH() ... AGAINST` queries that use a long string as an argument for `AGAINST()` could result in an error when run on an `InnoDB` table with a full-text search index. (Bug #17640261)

- **InnoDB:** Databases names beginning with a digit or special character would cause a full-text search (FTS) parser error. (Bug #17607687)

  References: See also: Bug #17607956.

- **InnoDB:** Under certain conditions, a regression introduced by the fix for Bug #11758237 would cause an assertion error when `INSERT ... ON DUPLICATE KEY UPDATE` or `REPLACE` statements encounter a `DB_DUPLICATE_KEY` error. (Bug #17604730)

  References: This issue is a regression of: Bug #11758237.

- **InnoDB:** In debug builds, a merge insert buffer during a page read would cause a memory access violation. (Bug #17561188)

- **InnoDB:** The patch for Bug #16852278, which simplifies and optimizes comparison functions in `InnoDB`, caused a query performance regression. (Bug #17543588)

  References: See also: Bug #16852278.

- **InnoDB:** In `sync0rw.ic`, `rw_lock_x_lock_func_nowait` would needlessly call `os_thread_get_curr_id`. (Bug #17509710, Bug #70417)

- **InnoDB:** Truncating a `memcached` `InnoDB` table while `memcached` is performing DML operations would result in a serious error. (Bug #17468031)

- **InnoDB:** The server could fail to restart if a crash occurred immediately following a `RENAME TABLE` in an `ALTER TABLE`, `RENAME TABLE` sequence. (Bug #17463290)

- **InnoDB:** If a tablespace data file path is updated in a `.isl` file and then a crash recovery is performed, the updated tablespace data file path is read from the `.isl` file but the `SYS_DATAFILES` table would not be not updated. The `SYS_DATAFILES` table is now updated with the new data file path after crash recovery. (Bug #17448389)

- **InnoDB:** Attempting to rename a table to a missing database would result in a serious error. (Bug #17447500)

- **InnoDB:** If the first page (page 0) of file-per-table tablespace data file was corrupt, recovery would be halted even though the doublewrite buffer contained a clean copy of the page. (Bug #17335427, Bug #70087, Bug #17341780)

- **InnoDB:** The `InnoDB` `memcached` Readme file (`README-innodb_memcached`) incorrectly stated that libevent 1.6.0 is linked statically into daemon `memcached`. The bundled version of libevent is 1.4.12, not 1.6.0. (Bug #17324419, Bug #70034)

- **InnoDB:** When creating a table there are a minimum of three separate inserts on the `mysql.innodb_index_stats` table. To improve `CREATE TABLE` performance, there is now a single `COMMIT` operation instead of one for each insert. (Bug #17323202, Bug #70063)

- **InnoDB:** The server would halt with an assertion in `lock_rec_has_to_wait_in_queue(lock)` due to a locking-related issue and a transaction being prematurely removed from `trx_sys->rw_trx_set`. (Bug #17320977)

- **InnoDB:** The `ALTER TABLE` `INPLACE` algorithm failed to decrease the auto-increment value. (Bug #17250787, Bug #69882)

- **InnoDB:** Comments in `btr0cur.cc` incorrectly stated that `btr_cur_pessimistic_update()` and `btr_cur_optimistic_update()` would accept a NULL value. (Bug #17231743, Bug #69847)
- **InnoDB:** `dict_table_schema_check` would call `dtype_sql_name` needlessly. (Bug #17193801, Bug #69802)
- **InnoDB:** `fil_check_first_page()` failed to check if `fsp_flags_get_zip_size()` returned a valid value, which resulted in a segmentation fault when starting `mysqld`. (Bug #17033182)
- **InnoDB:** The function `os_file_get_status` would not work with raw devices. (Bug #17023438, Bug #69424)
- **InnoDB:** The `lock_rec_other_has_expl_req` function in `lock0lock.cc` would perform unnecessary work. (Bug #17016214, Bug #69576)
- **InnoDB:** Valgrind would report uninitialized values while running a rollback debug test. The Valgrind warnings should only appear in Valgrind-instrumented builds. (Bug #16969876)
- **InnoDB:** During crash recovery, an incorrect transaction active time would result in rolling back an uncommitted transaction. (Bug #16936961, Bug #69438)
- **InnoDB:** Heap block debugging information (`file_name`, `lineno`), used for logging diagnostics, would appear in release builds. This information should only appear in debug builds. (Bug #16924719, Bug #69422)
- **InnoDB:** An online `ALTER TABLE` operation would consume more memory than expected. During an online `ALTER TABLE` operation, an online log buffer containing a head and tail buffer is created for each index that is created or rebuilt. The tail buffer is the writer context and is only required for concurrent write operations on an index while the `ALTER TABLE` operation is in progress. The head buffer is the reader context and is only required during the log apply phase. To reduce memory consumption, the tail buffer is now allocated when the first DML statement is run on the index, and the head buffer is only allocated in the log apply phase and freed afterwards. (Bug #16868967, Bug #69325, Bug #17911720)
- **InnoDB:** Renaming a column while also adding or dropping columns in the same `ALTER TABLE` operation would cause an error. (Bug #16864981)
- **InnoDB:** A type name (`srv_shutdown_state`) was the same as a variable name. The `srv_shutdown_state` type name has been changed to `srv_shutdown_t`. (Bug #16735398)
- **InnoDB:** The `buf_buddy_relocate` function would perform an unnecessary hash lookup. (Bug #16596057)
- **InnoDB:** On Windows, the full-text search (FTS) object ID was not in the expected hexadecimal format. (Bug #16559254)

  References: See also: Bug #16559119.
- **InnoDB:** Server shutdown would result in a hang with the following message written to the error log: "`[NOTE] InnoDB: Waiting for purge thread to be suspended`." (Bug #16495065)
- **InnoDB:** `InnoDB` failed to start when `innodb_data_file_path` specified the data file size in kilobytes by appending `K` to the size value. (Bug #16287752)
- **InnoDB:** Fetching and releasing pages from the buffer pool and tracking the page state are expensive and complex operations. Prior to the bug fix, these operations were performed using a page mutex. Using a page mutex to track several things is expensive and does not scale well. The bug fix separates fetch and release tracking (in-use state) of a page from page I/O state tracking. Fetch and release is now tracked using atomics where available.

For portability, a new `CMake` build option, `INNODB_PAGE_ATOMIC_REF_COUNT` (default `ON`), can be used to disable atomic page reference counting on platforms where atomics support is not available. When atomic page reference counting is enabled (default), "`[Note] InnoDB: Using atomics to ref count buffer pool pages`" is printed to the error log at server startup. If atomic page reference counting is disabled, "`[Note] InnoDB: Using mutexes to ref count buffer pool pages`" is printed instead. (Bug #16249481, Bug #68079)

- **InnoDB:** An insert buffer merge would cause an assertion error due to incorrectly handled ownership information for externally stored BLOBs.

```
InnoDB: Assertion failure in thread thread_num in file ibuf0ibuf.cc line 4080
InnoDB: Failing assertion: rec_get_deleted_flag(rec, page_is_comp(page))
```

  (Bug #14668683)

- **InnoDB:** Decreasing the `auto_increment_increment` value would have no affect on the next auto-increment value. (Bug #14049391, Bug #65225)

- **InnoDB:** Table renaming errors would appear in the `LATEST FOREIGN KEY ERROR` section of the `SHOW ENGINE INNODB STATUS` output. (Bug #12762390, Bug #61746)

- **InnoDB:** The page latching algorithm for b-trees would lock sibling leaf pages, prolonging dictionary locks. The bug fix implements prefectching of sibling leaf pages to reduce index lock holding time. (Bug #12734249, Bug #61736)

- **InnoDB:** `BUF_READ_AHEAD_AREA` would frequently call `ut_2_power_up` for workloads with a high I/O rate. The calculation is now performed once and the result is stored in the `buf_pool_t` structure. (Bug #11762242, Bug #54814)

- **InnoDB:** `UNIV_SYNC_DEBUG`, which was disabled in `univ.i` with the fix for Bug#16720368, is now enabled. (Bug #69617, Bug #17033591)

- **Partitioning:** Queries using the `index_merge` optimization (see Index Merge Optimization) could return invalid results when run against tables that were partitioned by `HASH`. (Bug #17588348, Bug #70588)

  References: See also: Bug #16862316, Bug #17648468, Bug #18167648.

- **Partitioning:** When no partition had returned a row since the last **`HA_ERR_KEY_NOT_FOUND`** error, the use of uninitialized memory in the priority queue used for returning rows in sorted order could lead to a crash of the server. (Bug #17401628)

- **Replication:** When running the server with `--gtid-mode=ON`, `STOP SLAVE` followed by `START SLAVE` resulted in a mismatch between the information provided by `INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO` and the `Slave_open_temp_tables` status variable: the `INNODB_TEMP_TABLE_INFO` table showed that no temporary tables existed, but `Slave_open_temp_tables` had a nonzero value. (Bug #18236612)

- **Replication:** Attempting to use semisynchronous replication concurrently with SSH connections caused the server to fail. (Bug #18219471)

- **Replication:** When `MASTER_HEARTBEAT_PERIOD` was not included in `CHANGE MASTER TO`, the statement reset `Slave_heartbeat_period` to its default value and `Slave_received_heartbeats` to 0. Now the heartbeat period is not changed by `CHANGE MASTER TO` unless explicitly set using `MASTER_HEARTBEAT_PERIOD`. In addition, the statement no longer resets `Slave_received_heartbeats`. (Bug #18185490)

- **Replication:** After setting `MASTER_SSL_CRLPATH` using a `CHANGE MASTER TO` statement, the option value was not displayed properly in the `SSL_CRL_PATH` column of the `replication_connection_configuration` Performance Schema table. (Bug #18174719)

- **Replication:** The `MASTER_SSL_CRL` and `MASTER_SSL_CRLPATH` options are not available when using yaSSL; MySQL Replication now sets these to `NULL` automatically whenever yaSSL is enabled. (Bug #18165937)

- **Replication:** `mysqlbinlog` did not free up memory used by its event buffer when using the `--rewrite-db` option. (Bug #18164998)

- **Replication:** Setting `--slave-parallel-workers` to 1 or greater and starting the slave caused the slave SQL thread to use but not release memory until the slave was restarted with `STOP SLAVE` and `START SLAVE`. (Bug #18001777, Bug #71197)

- **Replication:** When a slave was configured with replication filters and `--log-warnings=2`, every statement which was filtered caused an entry to be written in the error log. For busy servers which generated many statements to be filtered, the result was that the error log could quickly grow to many gigabytes in size. Now a throttle is used for such errors, so that an error message is printed only once in a given interval, saying that this particular error occurred a specific number of times during that interval. (Bug #17986385)

- **Replication:** When the binary log I/O cache grew to exactly 32768 bytes and the current transaction was preceded by a transaction whose size was greater than 32768 bytes, events could be corrupted when written into the binary log. (Bug #17842137)

- **Replication:** When the master and the slave both had `gtid_mode=ON` set initially, and the slave SQL thread was stopped while there remained GTID transactions in the relay log, if the slave was then restarted with `gtid_mode=OFF`, then the slave SQL thread executed any anonymous transaction it encountered without writing its GTID to the binary log, with the result that the GTID was lost. This could cause problems when the slave was later promoted to a master, as the transaction would be played again on the promoted master's slaves, leading quickly to inconsistencies on those slaves. (Bug #17827018)

  References: See also: Bug #17813449.

- **Replication:** When the master and the slave both had `gtid_mode=OFF` set initially, and the slave SQL thread was stopped while there remained anonymous transactions in the relay log, if the slave was then restarted with `gtid_mode=ON`, then the slave assigned GTIDs such transactions. This could cause problems when the slave was later promoted to a master, as the transactions would be played again on the promoted master's slaves, leading quickly to inconsistencies on those slaves. (Bug #17813449)

  References: See also: Bug #17827018.

- **Replication:** Creating and dropping large numbers of temporary tables could lead to increased memory consumption. (Bug #17806014)

- **Replication:** `SHOW SLAVE STATUS` used incorrect values when reporting `MASTER_SSL_CRL` and `MASTER_SSL_CRLPATH`. (Bug #17772911, Bug #70866)

  References: This issue is a regression of: Bug #11747191.

- **Replication:** When `log_warnings` is greater than 1, the master prints binary log dump thread information—containing the slave server ID, binary log file name, and binary log position—in `mysqld.1.err`. A slave server ID greater than 2 billion was printed with a negative value in such cases. (Bug #17641586, Bug #70685)

- **Replication:** `mysqlbinlog --verbose` failed when it encountered a corrupt row event in the binary log. Such a row event could also cause the slave to fail. (Bug #17632978)

  References: See also: Bug #16960133.

- **Replication:** Binary log events could be sent to slaves before they were flushed to disk on the master, even when `sync_binlog` was set to 1. This could lead to either of those of the following two issues when the master was restarted following a crash of the operating system:

  - Replication cannot continue because one or more slaves are requesting replicate events that do not exist on the master.
  - Data exists on one or more slaves, but not on the master.

  Such problems are expected on less durable settings (`sync_binlog` not equal to 1), but it should not happen when `sync_binlog` is 1. To fix this issue, a lock (`LOCK_log`) is now held during synchronization, and is released only after the binary events are actually written to disk. (Bug #17632285, Bug #70669)

- **Replication:** When running the slave with `--slave-parallel-workers` at 1 or greater, setting `--slave-skip-errors=all` caused the error log to be filled with instances of the warning `Slave SQL: Could not execute Query event. Detailed error: ;, Error_code: 0`. (Bug #17581990, Bug #68429)

  References: See also: Bug #17986385.

- **Replication:** When semi-synchronous replication was configured on an independent server with no slaves and `rpl_semi_sync_master_wait_no_slave` was set to `OFF`, the master still waited for an `ACK` from the slave. When `rpl_semi_sync_master_wait_no_slave` is set to `OFF`, the master should revert to normal replication when the number of slaves reaches zero during the specified timeout period. Now in such cases the server checks whether semi-synchronous replication is switched on, and, if so, goes on to check whether any slaves are connected. If none are connected, semi-synchronous replication is disabled until such time that the user sets the value of `rpl_semi_sync_master_wait_no_slave` to `ON`. (Bug #17510411, Bug #70360)

- **Replication:** A number of possible state messages used as values for the `PROCESSLIST_STATE` column of the `threads` Performance Schema table were longer than the width of the column (64 characters).

  The long state messages have now been rewritten, and shortened accordingly. This fix applies in MySQL 5.7 and later. (Bug #17319380)

- **Replication:** `Seconds_Behind_Master` in the output of `SHOW SLAVE STATUS` could under some conditions be reported as 0 when it should have had a value greater than zero. (Bug #17233214)

  References: See also: Bug #16579028.

- **Replication:** The server did not handle correctly the insertion of a row larger than 4 GB when using row-based replication. (Bug #17081415)

- **Replication:** When using row-based replication, an additional auto-increment column on the slave version of a table was not updated correctly; a zero was inserted instead. (Bug #17066269, Bug #69680)

- **Replication:** Statements involving the Performance Schema tables should not be written to the binary log, because the content of these tables is applicable only to a given MySQL Server instance, and may differ greatly between different servers in a replication topology. The database administrator should be able to configure (`INSERT`, `UPDATE`, or `DELETE`) or flush (`TRUNCATE TABLE`) performance schema

tables on a single server without affecting others. However, when replicating from a MySQL 5.5 master to a MySQL 5.5 or later slave, warnings about unsafe statements updating Performance Schema tables were elevated to errors. For MySQL 5.6 and later slaves, this prevented the simultaneous use of `performance_schema` and GTIDs (see Replication with Global Transaction Identifiers).

This fix causes all updates on tables in the `performance_schema` database to be filtered on the master and not replicated, regardless of the type of logging that is in effect. Prior to this fix, statements using were handled by being marked as unsafe for replication, which caused warnings during execution; the statements were nonetheless written to the binary log, regardless of the logging format in effect.

Existing replication behavior for tables in the `INFORMATION_SCHEMA` database is not changed by this fix.

For more information, see MySQL Performance Schema. (Bug #16814264)

References: See also: Bug #14741537, Bug #18259193.

- **Replication:** Invalid event offsets in the binary log were not always handled correctly, which could lead to replication failure. (Bug #16736412, Bug #69087)
- **Replication:** The semisynchronous replication plugin was called twice for a DDL statement, incrementing `Rpl_semi_sync_master_yes_tx` by 2 instead of 1 each time such a statement was executed. (Bug #70410, Bug #17509011)
- **Replication:** Semisynchronous replication became very slow if there were many dump threads (such as from `mysqlbinlog` or slave I/O connections) working at the same time. It was also found that semisynchronous master plugin functions were called even when the dump connections did not support semisynchronous replication, which led to locking of the plugin lock as well as wasting time on necessary code.

  After this fix, non-semisynchronous dump threads no longer call semisynchronous master functions to observe binary events. (Bug #70218, Bug #17434690)
- During compilation, attempts to create `sql_yacc.h` could be made from multiple directories simultaneously. (Bug #18319335)
- MySQL distributions for Solaris now include a source tarball for `gcc` under the `share` directory, to comply with GPL conditions resulting from inclusion of the C++ runtime library. (Bug #18306484)
- `mysql_secure_installation` attempted to free memory incorrectly and exited abnormally after a failed attempt to read an option file. (Bug #18255657)
- While printing the server version, the `mysql` client did not check for buffer overflow in a string variable. (Bug #18186103)
- `mysql_secure_installation` exited if `mysql_install_db` had been run with the `--skip-random-passwords` option. (Bug #18181665)
- Compilation failed if MySQL was configured with `CFLAGS` set to include a `-Werror` option with an argument. (Bug #18173037)
- When the optimizer attempted to use MRR or DS-MRR to read an internally created temporary table, the server could exit or raise an assertion. (Bug #18172819)
- The default compiler flags are picked up from `cmake/build_configurations/compiler_options.cmake`. This can be switched off by the `CMake -DWITH_DEFAULT_COMPILER_OPTIONS=0` option. However, it could also be switched off for the C or C++ compilers if the `CFLAGS` or `CXXFLAGS` environment variables were set.

Those environment variables now have no such effect. To specify compiler flags, use `-DWITH_DEFAULT_COMPILER_OPTIONS=0` option, or the `-DCMAKE_C_FLAGS=flags` and `-DCMAKE_C_FLAGS=flags` options can be used. (Bug #18158812)

- A bug in the range optimizer code that handles index merge could lead to a server exit or missing rows in the result set. (Bug #18136628)
- A shared `libmysqld` embedded server library was not built on Linux. A new `WITH_EMBEDDED_SHARED_LIBRARY` `CMake` option now makes this possible. (Bug #18123048, Bug #16430656, Bug #68559)
- Type casting during `LIKE` pattern match operations could cause a server exit. (Bug #18114294)
- `mysql_config` improperly produced nonempty output when invoked with the `--libmysqld-libs` (or a synonym) if MySQL was configured with the `WITHOUT_SERVER` option. (Bug #18102839)
- Repeated rebuilds in the same source tree resulted in `libmysqld.a` increasing in size each time. (Bug #18082702)
- `SHOW GRANTS` could be used to view the password hash for a proxied user. Password hash display now requires the `SUPER` privilege. (Bug #18057514)
- Building MySQL from source on Windows using Visual Studio 2008 failed with an `identifier not found` error due to a regression introduced by the patch for Bug#16249481. (Bug #18057449)

  References: This issue is a regression of: Bug #16249481.
- On Microsoft Windows, the rw-lock backup implementation for the `my_atomic_*` functions was always used. Now, the native Microsoft Windows implementation is used, where available. (Bug #18054042)
- When tables are reopened from the table cache and the current thread is not instrumented for the Performance Schema, the server exited attempting to populate `OWNER_THREAD_ID` in the `table_handles` table. (Bug #18047865)
- Link failures were fixed on Solaris SPARC and Linux 64-bit platforms. (Bug #18004599)
- A memory leak occurred within the Performance Schema during server startup. (Bug #18003651)
- Building `libevent` was incorrectly dependent on MySQL being configured with the `-DWITH_INNODB_MEMCACHED=1` option. (Bug #17964544)
- During shutdown, a mutex that was still locked could be removed, causing a server exit. (Bug #17959898)
- Compilation used different warning flags for Clang and GCC, producing different warning output depending on which compiler you use. Warning output is now consistent for the two compilers. (Bug #17959689)
- On Solaris, configuration failed if no `STL_LIBRARY_NAME` was found. (Bug #17954277)
- `storage/ndb/CMakeLists.txt` reset the `CMake` cache for some compiler flags for which the result should have been saved. (Bug #17949504)
- The `SUM_SORT_MERGE_PASSES` column value in the `events_statements_summary_by_digest` Performance Schema table was calculated incorrectly. (Bug #17938255)
- If the `events_statements_summary_by_digest` Performance Schema table was full when a statement with a new digest was found, the `Performance_schema_digest_lost` status variable was not incremented. (Bug #17935314)

- The audit log plugin could cause a server exit during log file rotation operations when there were many operations happening for multiple connections. (Bug #17930339)
- `DECIMAL NOT NULL` items could return `NULL` in subqueries. (Bug #17921777)
- `FORCE INDEX [FOR ORDER BY] (index_name)` did not work for joins.

  The fix for this bug also changes the warning created for `EXPLAIN`. Instead of printing only `{IGNORE|USE|FORCE} INDEX` it now also prints `FOR {GROUP BY|ORDER BY|JOIN}` if that was specified in the query. (Bug #17889511)
- `mysql_secure_installation` exited if it connected using SSL and the user had an expired password. (Bug #17880395)
- Shutdown of open connection threads could fail to occur cleanly during server shutdown. (Bug #17863690)
- For debug builds, inserts into a multiple-table view could raise an assertion. (Bug #17834434)
- The optimizer could push down a condition when the index did not have the key part present in the condition. (Bug #17814492)
- With the compressed client/server protocol enabled, Performance Schema statement instrumentation could raise an assertion. (Bug #17794846)
- The `resetconnection` command for `mysql` did not report proper errors if the server was down or the user password had expired. (Bug #17772561)
- Contraction information in a collation could be mishandled, resulting in incorrect decisions about whether a character is part of a contraction, and miscalculation of contraction weights. (Bug #17760379)
- An assertion could be raised if a `filesort` failed to resize its main buffer when record properties changed. (Bug #17757914)
- Valgrind errors were produced during row comparator setup. (Bug #17755540)
- The patch for Bug #16041903 introduced an incorrect `DBUG_ASSERT` that in debug builds raised a spurious assertion. (Bug #17746721)

  References: This issue is a regression of: Bug #16041903.
- Build and execution problems were fixed for builds made with `gcc` 4.8.1 in 32-bit mode on SPARC. (Bug #17740390)
- Compilation failed if MySQL was configured using `-DWITH_LIBWRAP=1`. (Bug #17738805)
- For debug builds, the `filesort` algorithm could raise a spurious assertion. (Bug #17734642)
- The `mysql_get_option` symbol was missing from `libmysql.dll`. (Bug #17733103)
- In some cases, `UNIX_TIMESTAMP()` could return `NULL` when it should return 0. (Bug #17728371)
- The server could exit when executing an `INSERT ... SELECT` with `UNION`, `ROLLUP`, and `ON DUPLICATE KEY UPDATE` with a subquery. (Bug #17727506)

  References: This issue is a regression of: Bug #16967281.
- The cache used for the Index Merge access method was freed only after successful retrieval of all rows. Interruption or failure of the operation led to a file descriptor leak. (Bug #17708621)

- The optimizer calculated the cost for joined buffer scans incorrectly, evaluating rows filtered out by attached conditions not once, but once per join buffer. (Bug #17694707)
- Using the `mysqldump --set-gtid-purged` option with no value caused `mysqldump` to crash. (Bug #17650245)
- If `SAFE_MUTEX` was enabled (true only for debug builds), `THR_LOCK_mutex` was used before being initialized. (Bug #17641055, Bug #70639)
- A race condition between Performance Schema statement event threads led to a server exit. (Bug #17637970)
- Incorrect reference counting in the range optimizer module resulted in potential for missing or duplicate rows in the query result set. (Bug #17619119)
- For debug builds, an aggregate function in a subquery join condition could raise an assertion. (Bug #17602807)
- After the fix for Bug #16409270, it was not possible to `#include <mysql.h>` following `#include <windows.h>`. (Bug #17514554)

  References: See also: Bug #16409270.
- An addressing error in accessing the join buffer could produce invalid results or a server exit. (Bug #17513341)
- The parser permitted some queries with multiple `ORDER BY` clauses, which then failed during execution and caused a server exit. (Bug #17473479)
- For debug builds, the server could exit for statements that inserted into a `BLOB` column declared as `NOT NULL` using a subquery that retrieved from a `BLOB` column and included `GROUP BY NULL`. (Bug #17458917)
- Within a `CASE` expression, use of a subquery referencing the `VALUES()` function could cause a server exit. (Bug #17458914)

  References: This issue is a regression of: Bug #14789787.
- `SET PASSWORD` combined with assignment of a variable from a subquery result could raise an assertion. (Bug #17458750)
- Insufficient cleanup after removal of a `SELECT_LEX` structure caused dereferencing of a NULL pointer and a server exit. (Bug #17458169)
- The parser silently accepted duplicate `ORDER BY` clauses and/or `LIMIT` clauses before `ORDER BY` clauses in subqueries. These caused failures during query execution. Fixing this problem results in some changes in parser behavior. The parser no longer accepts:
  - A `LIMIT` clause before an `ORDER BY` clause
  - A `LIMIT` clause in a parentheses-less `SELECT` statement before a `UNION` keyword
  - An `INTO` clause after a `PROCEDURE ANALYSE()` clause

  (Bug #17426017, Bug #17703542, Bug #17727401)
- On Windows, `mysql_secure_installation` exited if the `root` password was expired. (Bug #17415203)

- `mysql_config` incorrectly included some flags to generate compiler warning output. (Bug #17400967)
- With semi-join optimization enabled, queries with nested subqueries could cause a server exit due to incorrect resolution of references to columns in the middle query block. (Bug #17398972)
- If accepting a connection attempt failed due to an out-of-memory error, the server could access a stale thread structure for a previously disconnected connection, resulting in Valgrind errors. (Bug #17398792)
- The SHA256 password authentication algorithm allocated a buffer one byte too short. (Bug #17397073)
- For `CASE` expressions involving floating-point numbers, the `max_length` and `decimal` values could be computed incorrectly. The logic for `CASE` was corrected to be the same as for `COALESCE()`, which performs a similar operation. (Bug #17388045)
- A client crash occurred if `mysql_set_server_option()` or several other C API functions were called before `mysql_real_connect()`. (Bug #17338958)
- In some cases, the optimizer wrote fixed-length temporary `MyISAM` tables to disk rather than variable-length temporary tables. (Bug #17231940)
- Enabling the `validate_password` plugin could result in incorrect password hashes being stored in the `mysql.user` table. (Bug #17065383)
- For debug builds, the second execution of a prepared statement processed using a semi-join could cause a server exit. (Bug #16988465)
- A spurious assertion was raised for queries processed using a semi-join LooseScan optimization that required rows to be returned in order. (Bug #16977389)
- A circular dependency problem involving `sql/sql_builtin.cc` was resolved. (Bug #16877045)
- For accounts authenticated using the `sha256_password` plugin, setting the password after the password had been expired did not clear the password-expired flag. (Bug #16872181)
- During server shutdown, file information was freed before calling `query_logger.cleanup()`, leading to a memory leak. (Bug #16859266)
- For prepared `INSERT INTO ... SELECT` statements, nonexistent column names were not reported during statement preparation, but only later at statement execution. (Bug #16820562)
- Multiple-table updates failed to update under certain conditions. (Bug #16767011)
- Crash recovery of temporary tables used uninitialized memory. (Bug #16754540)
- On Mac OS X 10.7, a race condition involving `vio_shutdown()` and the select-based implementation of `vio_io_wait()` could cause a server exit. (Bug #16354789, Bug #17733393)
- Host names in example URLs used within the source code were replaced by names in the example.com domain, the domain intended by IANA for this purpose. (Bug #15890092)
- For `utf8` and `utf8mb4` strings, handler functions unnecessarily called a Unicode conversion function. (Bug #14057034)
- On Mac OS X, preloading of client plugins specified with the `LIBMYSQL_PLUGINS` environment variable could fail unless the plugins were located in the hardwired default plugin directory. The C API now checks during plugin preloading for a `LIBMYSQL_PLUGIN_DIR` environment variable which can be set to the path name of the directory in which to look for client plugins. (Bug #13994567, Bug #18110355)
- Certain `(... NULL ...) IN (...)` expressions returned `NULL` when they should return 0, such as `SELECT (NULL, 1) IN ((0, 0), (0, 0))`. (Bug #13944462)

- Several `-W` warning flags were turned off for compilation in maintainer mode if MySQL was configured with `-DWITH_INNODB_MEMCACHED=1`. (Bug #13898319)
- The optimizer set up for dynamic range access in some cases where range access cannot be used, resulting in fallback to a table scan. (Bug #13814468)
- Executing `mysqladmin shutdown` for a server running with the thread pool plugin enabled and servicing a large number of concurrent connections caused the server to exit abnormally. (Bug #13788920)
- Calling the `ExtractValue()` function with an invalid XPath expression could in some cases lead to a failure of the server. (Bug #12428404, Bug #61065)
- Use of a nonmultibyte algorithm for skipping leading spaces in multibyte strings could cause a server exit. (Bug #12368495, Bug #18315770)
- With `ONLY_FULL_GROUP_BY` SQL mode enabled, a query that uses `GROUP BY` on a column derived from a subquery in the `FROM` clause failed with a `column isn't in GROUP BY` error, if the query was in a view. (Bug #11923239)
- The `stage/sql/Waiting to get readlock` Performance Schema instrument is no longer used and has been removed. (Bug #71298, Bug #18035404)
- `mysqlbinlog` leaked memory in relation to `--rewrite-db` processing. (Bug #71283, Bug #18027692)
- Previously, for `EXPLAIN` output, the rows-examined estimate for Performance Schema tables always displayed as 1000. Now a more accurate estimate is displayed based on sizing parameters used when allocating memory for each table. This results in no change of behavior because Performance Schema tables have no indexes. (Bug #71278, Bug #18024455)
- Optimizer trace output from the range optimizer could include raw binary data and generate unprintable characters. Now binary data is printed in hex format. (Bug #71273, Bug #18023222)
- During configuration, `CMake` improperly checked for the C++ header file `cxxabi.h`. (Bug #71268, Bug #18147458)
- Aggregating the results of a subquery in the `FROM` clause could produce incorrect results. (Bug #71244, Bug #18014565)
- Previously, the first stage executed within a statement was `stage/sql/init`. This collided with a different stage named `init` and was incompatible with the `starting` stage for `SHOW PROFILE`. The first stage executed within a statement is now named `stage/sql/starting`. (Bug #71201, Bug #17993294)
- `CMake` produced a warning in `ssl.cmake` due to malformed syntax. (Bug #71094, Bug #17905144)
- `CMake` produced not-useful warnings about `INTERFACE_LINK_LIBRARIES` policy. (Bug #71089, Bug #17905155, Bug #17894997)
- `mysqldump --single-transaction` acquired metadata locks for each dumped table but did not release them until the dump operation finished. Consequently, other DDL operations on a dumped table blocked even after the table itself had been dumped. `mysqldump` now attempts to release metadata locks earlier. (Bug #71017, Bug #17862905)
- `sql_resolver.cc` referred to partitioning code that should have been protected by an `#ifdef`, even when MySQL was configured with `-DWITH_PARTITION_STORAGE_ENGINE=OFF`. (Bug #71010, Bug #17876794)

- The `wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_sync` mutex was not properly instrumented for the Performance Schema. (Bug #70939, Bug #17813333)
- The `-DWITH_EXAMPLE_STORAGE_ENGINE=1` `CMake` option was ignored but should not have been. If `-DWITH_EXAMPLE_STORAGE_ENGINE=0` is given, the `EXAMPLE` storage engine is built as a plugin. (Bug #70859, Bug #17772560)

  References: See also: Bug #18324650.
- `FLUSH STATUS` cleared a variable that could result a subsequent implicit commit of an XA transaction causing a server exit. (Bug #70854, Bug #17911445)
- Overhead was reduced within critical sections of the `my_fopen()` and `my_register_filename()` `mysys` functions. Thanks to Po-Chun Chang for the patch. (Bug #70848)
- Several issues identified by the Coverity static analysis tool were fixed. Thanks to Honza Horak for the patch. (Bug #70830, Bug #17760511)
- A query that creates a temporary table to find distinct values and has a constant value in the projected list could produce incorrect results. (Bug #70657, Bug #17634335)
- Configuring with `-DWITH_DEBUG=1` did not have the same effect as configuring with `-DCMAKE_BUILD_TYPE=Debug`. (Bug #70647, Bug #17632854)
- The prototype of the Performance Schema instrumentation API `mysql_cond_timedwait()` call was fixed to be drop-in compatible with `pthread_cond_timedwait()`. This fix affects only implementers of third-party plugins. (Bug #70628, Bug #17702677)
- Some `BETWEEN` expressions on unsigned values were evaluated using signed arithmetic. Thanks to Xiaobin Lin for the patch. (Bug #70622, Bug #17606942)
- An incorrect result could be returned for a query with an `IF()` predicate in the `WHERE` clause combined with `OUTER JOIN` in a subquery that is transformed to a semi-join. (A workaround is to disable semi-join using `SET optimizer_switch='semijoin=off';`) (Bug #70608, Bug #17600176)
- The server wrote an excessive number of “Sort aborted” messages to the error log. (Bug #70173, Bug #17372396)
- When run by `root`, `mysqld --help --verbose` exited with a nonzero error code after displaying the help message. (Bug #70058, Bug #17324415)
- Complex updates of Performance Schema tables involving joins or subqueries failed to update every row. (Bug #70025, Bug #17309657)
- For debug builds, JSON-format `EXPLAIN` statements for queries that involve semi-join materialization could cause a server exit. (Bug #70014, Bug #17305943)
- A deadlock error occurring during subquery execution could cause an assertion to be raised. (Bug #69969, Bug #17307201)
- For an existing user, `GRANT` with an empty password (`IDENTIFIED BY [PASSWORD] ''`) did not change the password. (Bug #69899, Bug #17256161)
- Downloading of the Google Mock library could fail during configuration. This is fixed by requiring `CMake` 2.8.2 or higher. (Bug #69854, Bug #17231722)
- Some files in the `file_instances` Performance Schema table were not being removed because the file-removal operation was not instrumented. (Bug #69782, Bug #17209750)

- For the path specified with the `--basedir` option, `mysql_plugin` attempted to unlink the path rather than free the memory in which the path was stored. (Bug #69752, Bug #17168602)
- A temporal literal string without delimiters and more than 14 digits was validated as a `TIMESTAMP/DATETIME` value with a two-digit precision fractional seconds part. But fractional seconds should always be separated from other parts of a time by a decimal point. (Bug #69714, Bug #17080703)
- For system variables that take a string value, `SET` statements permitted an unquoted value, but values that contained dots were parsed incorrectly and only part of the value was assigned. For example, `SET GLOBAL slow_query_log_file = my_slow.log` assigned the value `my_slow`. Now such values must be quoted or an error occurs. (Bug #69703, Bug #17075846)
- It was not possible to query a view with an `ORDER BY` clause that referenced an alias in the `SELECT` clause of the view definition, unless all columns in the view were named in the select list.

  To handle this problem, the server now writes a view differently into the `.frm` file that stores the view definition. If you experience view-evaluation errors such as just described, drop and recreate the view so that the `.frm` file contains the updated view representation. (Bug #69678, Bug #17077305)
- On Windows, the `--local-service` server option did not work, and was not displayed in the `--help` message. (Bug #69637, Bug #17049656)
- The `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlslap`, and `mysqlshow` programs now support a `--secure-auth` option that prevents sending passwords to the server in old (pre-4.1) format. This option is enabled by default; use `--skip-secure-auth` to disable it. (Bug #69051, Bug #16723046)
- For the `utf8_bin` collation, `ORDER BY LOWER(col_name)` could produce incorrect ordering. (Bug #69005, Bug #16691598)
- MySQL client programs from a Community Edition distribution could not connect using SSL to a MySQL server from an Enterprise Edition. This was due to a difference in certificate handling by yaSSL and OpenSSL (used for Community and Enterprise, respectively). OpenSSL expected a blank certificate to be sent when not all of the `--ssl-ca`, `--ssl-cert`, and `--ssl-key` options were specified, and yaSSL did not do so. To resolve this, yaSSL has been modified to send a blank certificate when an option is missing. (Bug #68788, Bug #16715064)
- A full-text search combined with derived tables (subqueries in the `FROM` clause) caused a server exit.

  Now if a full-text operation depends on a derived table, the server produces an error indicating that a full-text search cannot be done on a materialized table. (Bug #68751, Bug #16539903)
- COUNT(DISTINCT) sometimes produced an incorrect result when the last read row contained a `NULL` value. (Bug #68749, Bug #16539979, Bug #71028, Bug #17867117)
- Some scripts displayed out-of-date information regarding where to report bugs. (Bug #68742, Bug #16530527)
- Updating a `FEDERATED` table with `UPDATE... JOIN` caused a server exit when the local table contained a single row and that row could be joined to a row in the `FEDERATED` table. (Bug #68354, Bug #16324629)
- Messages written by the server to the error log for LDML collation definition problems were missing the collation name. (Bug #68144, Bug #16204175)
- `mysqlcheck` did not correctly handle table names containing dots. (Bug #68015, Bug #16064833)
- Compilation problems were fixed for errors reported by `Clang` and `gcc` when compiling in C++11 mode. (Bug #66803, Bug #14631159)

- `cmake/configure.pl` listed instances of `WITH_COMMENT` rather than the correct option `COMPILATION_COMMENT`. (Bug #65834, Bug #14298560)
- The `make_atomic_cas_body64` implementation on IA32 with `gcc` but without `gcc` builtins could be miscompiled due to an incorrect constraint. The patch also causes MySQL to use builtin atomics when compiled using `Clang`. (Bug #63451, Bug #17242996)
- On Mac OS X, the `libmysqlclient` dylib file linked to itself. (Bug #61699, Bug #13890998, Bug #61243, Bug #12590037)
- The optimizer could choose `ref` access over `eq_ref` access when cost of a nonunique access was evaluated before cost of a unique index. (Bug #54808, Bug #11762236)
- On Windows, `mysql_install_db.pl` could be run only from within the `bin` directory under the installation directory. (Bug #42421, Bug #11751526)
- `gcov` printed warnings without file names. (Bug #33269, Bug #11747622)
- `mysql_install_db` referred to the obsolete `mysqlbug` script for reporting problems. It now refers to http://bugs.mysql.com/ instead. (Bug #29716, Bug #11746921)
- The deprecated `msql2mysql`, `mysql_convert_table_format`, `mysql_find_rows`, `mysql_fix_extensions`, `mysql_setpermission`, and `mysqlaccess` utilities were removed. (Bug #27482, Bug #69012, Bug #69014, Bug #69015, Bug #69016, Bug #69017, Bug #11746603, Bug #16699248, Bug #16699279, Bug #16699284, Bug #16699317, Bug #18179576)

Changes in MySQL 5.7.3 (2013-12-03, Milestone 13)

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.

Audit Log Notes

- MySQL 5.7 changed audit log file output to a new format that has better compatibility with Oracle Audit Vault. It is now possible to select either the old or new format using the new `audit_log_format` system variable, which has permitted values of `OLD` and `NEW` (default `NEW`). For details about each format, see The Audit Log File.

If you change the value of `audit_log_format`, use this procedure to avoid writing log entries in one format to an existing log file that contains entries in a different format:

1. Stop the server.
2. Rename the current audit log file manually.
3. Restart the server with the new value of `audit_log_format`. The audit log plugin will create a new log file, which will contain log entries in the selected format.

Compilation Notes

- `CMake` configuration for the `Clang` compiler sets more appropriate flags for building on Linux. Specifically, `-g -fno-omit-frame-pointer -fno-strict-aliasing` is now added. (Bug #17633291)

Full-Text Search Notes

- **Important Change; InnoDB:** `InnoDB` now supports external full-text parser plugins. In order to support `InnoDB` full-text parser plugins that are called in boolean mode, a new "`position`" member has been added to the `MYSQL_FTPARSER__BOOLEAN_INFO` structure. If you plan to use an existing full-text parser plugin that is called in boolean mode with MySQL 5.7.3 or later, you must add support for the new "`position`" member, which is described in Writing Full-Text Parser Plugins. Altering a `MyISAM` table with a full-text parser plugin to use `InnoDB` is also supported. For additional information about full-text parser plugins, see Full-Text Parser Plugins.

Optimizer Notes

- The server no longer uses a temporary table for `UNION` statements that meet certain qualifications. Instead, it retains from temporary table creation only the data structures necessary to perform result column typecasting. The table is not fully instantiated and no rows are written to or read from it; rows are sent directly to the client. The result is reduced memory and disk requirements, and smaller delay before the first row is sent to the client because the server need not wait until the last query block is executed. `EXPLAIN` and optimizer trace output will change: The `UNION RESULT` query block will not be present because that block is the part that reads from the temporary table.

  The conditions that qualify a `UNION` for evaluation without a temporary table are:

  - The union is `UNION ALL`, not `UNION` or `UNION DISTINCT`.
  - There is no global `ORDER BY` clause.
  - The union is not the top-level query block of an `{INSERT | REPLACE} ... SELECT ...` statement.

  (Bug #50674, Bug #11758470)

- The optimizer now is able to apply the range scan access method to queries of this form:

```
SELECT ... FROM t1 WHERE ( col_1, col_2 ) IN (( 'a', 'b' ), ( 'c', 'd' ));
```

  Previously, for range scans to be used it was necessary for the query to be written as:

```
SELECT ... FROM t1 WHERE ( col_1 = 'a' AND col_2 = 'b' )
OR ( col_1 = 'c' AND col_2 = 'd' );
```

For the optimizer to use a range scan, queries must satisfy these conditions:

- Only `IN()` predicates are used, not `NOT IN()`.
- On the left side of the `IN()` predicate, the row constructor contains only column references.
- On the right side of the `IN()` predicate, row constructors contain only runtime constants, which are either literals or local column references that are bound to constants during execution.
- On the right side of the `IN()` predicate, there is more than one row constructor.

`EXPLAIN` output for applicable queries changes from full table scan or index scan to range scan. Changes are also visible by checking the values of the `Handler_read_first`, `Handler_read_key`, and `Handler_read_next` status variables. (Bug #31188, Bug #11747186)

- The modified `filesort` algorithm now includes an additional optimization designed to enable more tuples to fit into the sort buffer: For additional columns of type `CHAR` or `VARCHAR`, or any nullable fixed-size data type, the values are packed. For example, without packing, a `VARCHAR(255)` column value containing only 3 characters takes 255 characters in the sort buffer. With packing, the value requires only 3 characters plus a two-byte length indicator.

  For data containing packable strings shorter than the maximum column length or many `NULL` values, more records fit into the sort buffer. This improves in-memory sorting of the sort buffer and performance of disk-based merge sorting of the temporary file.

  In edge cases, packing may be disadvantageous: If packable strings are the maximum column length or there are few `NULL` values, the space required for the length indicators reduces the number of records that fit into the sort buffer and sorting is slower in memory and on disk.

  Packing is not applicable if the filesort uses a priority queue for sorting, as is the case when an `ORDER BY ... LIMIT` optimization is applied (see LIMIT Query Optimization).

  If a `filesort` is done, optimizer trace output includes a `filesort_summary` block. For example:

```
"filesort_summary": {
  "rows": 100,
  "examined_rows": 100,
  "number_of_tmp_files": 0,
  "sort_buffer_size": 25192,
  "sort_mode": "<sort_key, packed_additional_fields>"
}
```

  The `sort_mode` value provides information about the algorithm used and the contents of the sort buffer:

```
<sort_key, rowid>: sort using row pointers
<sort_key, additional_fields>: sort using additional fields
<sort_key, packed_additional_fields>: sort using packed additional fields
```

  For additional information about the `filesort` algorithm, see ORDER BY Optimization. For information about the optimizer trace, see MySQL Internals: Tracing the Optimizer.

Packaging Notes

- Previously, MySQL Server distributions included the MySQL Reference Manual in Info format (the Docs/mysql.info file). Because the license for the manual restricts redistribution, its inclusion in Community packages caused problems for downstream redistributors, such as those who create Linux distributions.

Community distributions of MySQL Server no longer include the mysql.info file, to make the repackaging and redistribution process easier (for example, the source tarball and its checksum can be used directly). This change applies to all source and binary Community packaging formats. Commercial (Enterprise) distributions are unchanged.

For those who wish to continue using the MySQL Reference Manual in Info format, we have made it available at http://dev.mysql.com/doc/.

Performance Schema Notes

- The Performance Schema now instruments transactions. The information collected includes quantitative and qualitative data including transaction duration, transaction counts, and frequency of various transaction attributes such as isolation level and access modes. This information is collected in tables that contain current and recent transaction events, and is aggregated in summary tables across several dimensions, including user, account, and thread (client connection).

  These new tables store transaction events:

  - `events_transactions_current`: Current transaction events
  - `events_transactions_history`: The most recent transaction events for each thread
  - `events_transactions_history_long`: The most recent transaction events overall

  There are also summary tables that provide aggregated transaction information.

  Within the event hierarchy, wait events nest within stage events, which nest within statement events, which nest within transactions. To reflect this, the `NESTING_EVENT_TYPE` column, in those tables that have it, permits a new value, `TRANSACTION`, in addition to the existing values `STATEMENT`, `STAGE`, and `WAIT`.

  To permit control over configuration of transaction event collection, these changes were made to Performance Schema setup tables:

  - The `setup_instruments` table contains a new instrument named `transaction`. This instrument is disabled by default.
  - The `setup_consumers` table contains new consumer values with names corresponding to the current and recent transaction event table names. These consumers may be used to filter collection of transaction events. Only `events_transactions_current` is enabled by default.
  - The `setup_timers` table contains a new row with a `NAME` value of `transaction` that indicates the unit for transaction event timing. The default unit is `NANOSECOND`.

  For more information, see Performance Schema Transaction Tables, and Transaction 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.

- The Performance Schema now exposes metadata lock information:

  - Locks that have been granted (shows which sessions own which current metadata locks)
  - Locks that have been requested but not yet granted (shows which sessions are waiting for which metadata locks).

- Lock requests that have been killed by the deadlock detector or timed out and are waiting for the requesting session's lock request to be discarded

This information enables you to understand metadata lock dependencies between sessions. You can see not only which lock a session is waiting for, but which session currently holds that lock.

The Performance Schema now also exposes table lock information that shows which table handles the server has open, how they are locked, and by which sessions.

These specific changes were implemented:

- The `metadata_locks` and `table_handles` tables list current locks and lock requests for metadata locks and table locks.
- The `setup_instruments` table now has a `wait/lock/metadata/sql/mdl` instrument for metadata locks. This instrument is disabled by default.
- The `performance_schema_max_metadata_locks` system variable configures the maximum number of metadata locks tracked in the `metadata_locks` table. For `table_handles`, the size is configured by the existing `performance_schema_max_table_handles` system variable.
- The `Performance_schema_metadata_lock_lost` status variable indicates the number of times a metadata lock could not be recorded. For `table_handles`, tables that are opened but cannot be instrumented are counted by the existing `Performance_schema_table_handles_lost` status variable.

For more information, see Performance Schema Lock 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.

Security Notes

- **Incompatible Change:** Previously, the `--ssl` option has been treated as advisory: When given, a secure connection was permitted but not required. Also, several other `--ssl-`*`xxx`* options implied `--ssl`. Because of this, the option was usually not used explicitly as `--ssl`, but in its negated form as `--ssl=0`, which prevents use of encryption. This was true on both the client and server sides, and true for any synonyms of `--ssl` (`--ssl=1`, `--enable-ssl`) or `--ssl=0` (`--skip-ssl`, `--disable-ssl`).

  Now the meaning of `--ssl` has changed *on the client-side only*. (There are no secure-connection changes on the server side.)

  When given on the client side as `--ssl` (or a synonym), the option is no longer advisory but prescriptive. Given a server enabled to support secure connections, a client program can require a secure conection by specifying only the `--ssl` option. The connection attempt fails if a secure connection cannot be established. In addition, other `--ssl-`*`xxx`* options no longer imply `--ssl`. This is an incompatible change in the sense that MySQL client commands that use `--ssl` now will fail unless a secure connection can be established. On the other hand, for a successful connection attempt, the connection is guaranteed to be secure. Previously, there was no such guarantee.

  If other `--ssl-`*`xxx`* options are given in the absence of `--ssl`, the client attempts to establish a secure connection. If the server is not configured to support secure connections, the client falls back to an unencrypted connection.

  There is no change in the meaning of `--ssl=0` (and its synonyms) to prevent use of encryption and override other `--ssl-`*`xxx`* options.

A new `MYSQL_OPT_SSL_ENFORCE` option is available for the `mysql_options()` C API function to indicate whether to require the connection to use encryption. If enabled, it has the same effect as specifying `--ssl` on the command line: If an encrypted connection cannot be established, the connection attempt fails.

For more information, see Command Options for Secure Connections, and mysql_options().

The `MASTER_SSL=1` option for the `CHANGE MASTER TO` statement has changed as well, analogous to the change in the meaning of `--ssl`. That is, when given, the slave connection to the master must use encryption or the connection attempt fails. (Bug #11744828)

Functionality Added or Changed

- **Performance; InnoDB:** The `log_write_up_to` function, which writes to redo log files up to a certain log sequence number (LSN) and optionally flushes writes to disk, has been refactored to improve performance for workloads with heavy `log_sys::mutex` contention and where `innodb_flush_log_at_trx_commit=2`.

- **Performance:** The `LOCK_thread_count` mutex protected several independent internal server structures and variables, and was a bottleneck, particularly affecting server performance in the circumstance when many clients were connecting and disconnecting at once. This mutex was decomposed into more specific mutexes and atomic operations to alleviate the bottleneck and improve performance.

  As part of this work, the following status variables are no longer visible in the embedded server because for that server they were not updated and were not meaningful: `Aborted_connects`, `Connection_errors_accept`, `Connection_errors_internal`, `Connection_errors_max_connections`, `Connection_errors_peer_addr`, `Connection_errors_select`, `Connection_errors_tcpwrap`.

- **Incompatible Change:** Several statement instruments in the `setup_instruments` table are used by the Performance Schema during the early stages of statement classification before the exact statement type is known. These instruments were renamed to more clearly reflect their “abstract” nature:

Old Instrument Name	New Instrument Name
`statement/com/`	`statement/abstract/new_packet`
`statement/com/Query`	`statement/abstract/Query`
`statement/rpl/relay_log`	`statement/abstract/relay_log`

  In addition, statistics for abstract instruments are no longer collected in the following tables, because no such instrument is ever used as the final classification for a statement:

```
events_statements_summary_by_thread_by_event_name
events_statements_summary_by_account_by_event_name
events_statements_summary_by_user_by_event_name
events_statements_summary_by_host_by_event_name
events_statements_summary_global_by_event_name
```

  Applications that refer to the old instrument names must be updated with the new names. For more information about the use of abstract instruments in statement classification, see Performance Schema Statement Event Tables. (Bug #16750433, Bug #17271055)

- **Incompatible Change:** The `EXPLAIN` statement has been changed so that the effects of the `EXTENDED` and `PARTITIONS` keywords are always enabled. `EXTENDED` and `PARTITIONS` are still recognized,

but are superfluous and have been deprecated. They will be removed from `EXPLAIN` syntax in a future MySQL release.

`EXPLAIN` output differs as follows as a result of this change:

- The filtered and partitions columns appear in `EXPLAIN` output regardless of whether the `EXTENDED` and `PARTITIONS` keywords are specified. This is an incompatible change for applications that expect to identify column information by position rather than by name, and such applications will need adjustment.
- `SHOW WARNINGS` immediately following `EXPLAIN` shows additional execution plan information regardless of whether the `EXTENDED` keyword is specified. (An additional deprecation warning is included if the statement includes the `EXTENDED` or `PARTITIONS` keyword.)

- **InnoDB:** The `InnoDB memcached` plugin now supports inserts and reads on mapped `InnoDB` tables that have an `INTEGER` defined as the primary key. (Bug #17315083, Bug #17203937)

- **Replication:** Replication filtering rules can now be set dynamically on the slave using the SQL statement `CHANGE REPLICATION FILTER` introduced in this release. This statement has the same effect as starting the slave `mysqld` with one or more of the options `--replicate-do-db`, `--replicate-ignore-db`, `--replicate-do-table`, `--replicate-ignore-table`, `--replicate-wild-do-table`, `--replicate-wild-ignore-table`, and `--replicate-rewrite-db`.

  For example, issuing the statement `CHANGE REPLICATION FILTER REPLICATE_DO_TABLE = (d1.t2)` is equivalent to starting the slave `mysqld` with `--replicate-do-table='d1.t2'`.

  `CHANGE REPLICATION FILTER` differs from the server options in that, to take effect, the statement requires only that the slave SQL thread be stopped beforehand and restarted afterwards, using `STOP SLAVE SQL_THREAD` and `START SLAVE SQL_THREAD`, respectively.

  This statement leaves any existing replication filtering rules unchanged; to unset all filters of a given type, set the filter to an empty list, as shown in this example:

```
CHANGE REPLICATION FILTER REPLICATE_DO_DB = ();
```

  You can list multiple replication filtering rules in the same statement, separated by commas. When multiple instances of the *same* rule are found, only the last instance is used.

  For more information, see CHANGE REPLICATION FILTER Syntax; see also How Servers Evaluate Replication Filtering Rules. (Bug #15877941, Bug #11752237, Bug #67362, Bug #43366)

- **Replication:** Previously, with semisynchronous replication enabled, the master waited for a single slave acknowledgment per transaction before proceeding. A new system variable, `rpl_semi_sync_master_wait_for_slave_count`, enables the number of slave acknowledgments required per transaction to be configured. The minimum (and default) value is 1. The maximum is 65,536. Performance is best for small values of this variable.

- The Performance Schema now instruments the read/write lock `Delegate::lock`, which is used for the following classes:

```
Trans_delegate
Binlog_storage_delegate
Binlog_transmit_delegate
Binlog_relay_IO_delegate
```

A different instrument name is used for each subclass, to have distinct statistics for distinct uses. The instruments are visible in the `schema.setup_instruments` table and have these names:

```
wait/synch/rwlock/sql/Trans_delegate::lock
wait/synch/rwlock/sql/Binlog_storage_delegate::lock
wait/synch/rwlock/sql/Binlog_transmit_delegate::lock
wait/synch/rwlock/sql/Binlog_relay_IO_delegate::lock
```

(Bug #17590161, Bug #70577)

- Some dependencies between client-side plugin header files were removed:
  - The `MYSQL_PLUGIN_EXPORT` macro required by plugin declarations is now declared directly in `mysql/client_plugin.h` instead of getting the definition from `mysql/plugin.h`. That macro was the only thing required by client-side plugins and declared in server-side header `mysql/plugin.h`, so including `mysql/client_plugin.h` in an application no longer requires the application to also include `mysql/plugin.h`.
  - `mysql/plugin_trace.h` no longer uses `C_MODE_START` or `C_MODE_END`. Consequently, including `mysql/plugin_trace.h` in an application no longer requires the application to also include `my_global.h`.

  Applications might require `mysql/plugin.h` or `my_global.h` for other reasons, of course. (Bug #17582168)
- Overhead for Performance Schema instrumentation associated with thread creation was reduced. (Bug #17539520)
- It is now possible to enable the Performance Schema but exclude certain parts of the instrumentation. For example, to enable the Performance Schema but exclude stage and statement instrumentation, do this:

```
shell> cmake . -DWITH_PERFSCHEMA_STORAGE_ENGINE=1 \
        -DDISABLE_PSI_STAGE=1 \
        -DDISABLE_PSI_STATEMENT=1
```

  For more information, see the descriptions of the `DISABLE_PSI_`*`XXX`* `CMake` options in MySQL Source-Configuration Options. (Bug #17478068)
- A new `CMake` option, `WITH_ASAN`, permits enabling AddressSanitizer for compilers that support it. (Bug #17435338)
- Several compilation warnings were fixed that occurred when compiling without debugging enabled. (Bug #17332094)
- The implementation of condition variables specific to Windows XP and Windows Server 2003 was removed from the source code because MySQL is not supported on those platforms as of MySQL 5.6. (Bug #17332056)
- A new `ER_ENGINE_OUT_OF_MEMORY` error code is available for use by storage engines to report out-of-memory conditions. (Bug #16807964)
- Overhead for deprecation warnings was reduced. (Bug #70402, Bug #17497869)
- For `GRANT` statements, `ER_SP_DOES_NOT_EXIST` errors for nonexistent stored procedures and functions now specify `PROCEDURE does not exist` or `FUNCTION does not exist` rather than the less-specific `PROCEDURE or FUNCTION does not exist`. (Bug #69628, Bug #17036976)

- The hash function used for metadata locking was modified to reduce overhead. (Bug #68487, Bug #16396598)
- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.
- Because there are new API functions (`mysql_get_option()`, `mysql_reset_connection()`), the library ABI version is now 18.2. Shared library names now include 18.2 where appropriate.
- A new `mysql_get_option()` C API function is available that returns the current value of applicable `mysql_options()` options. See mysql_get_option().
- When a connection is returned to the thread pool plugin, the connection thread context must be cleaned up. Previously, this was done using `COM_CHANGE_USER` (which is like the `mysql_change_user()` C API function). However, that operation reauthenticates, which is unnecessary network roundtrip overhead in this context.

  Now it is possible for client connection state to be reset in a more lightweight manner without causing reauthentication. The API is exposed publicly through these changes:

  - A new `COM_RESET_CONNECTION` protocol command (defined in `mysql_com.h`)
  - A new `mysql_reset_connection()` C API function
  - A new `resetconnection` command for the `mysql` client

  Resetting a connection has effects similar to `mysql_change_user()` or an auto-reconnect except that the connection is not closed and reopened, and reauthentication is not done. See mysql_change_user()) and see Controlling Automatic Reconnection Behavior).

  For more information, see mysql_reset_connection() and `mysql` — The MySQL Command-Line Tool.

Bugs Fixed

- **Incompatible Change:** For logging to the `general_log` and `slow_log` tables in the `mysql` database, log lines containing multiple character sets were not always handled correctly. The `general_log.argument` and `slow_log.sql_text` columns now have been changed from `MEDIUMTEXT` to `MEDIUMBLOB`. Consequently, no character set transformation is done for logging to tables now, which aligns it with logging to files.

  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` database. (Bug #14575847)
- **Important Change; Replication:** `START SLAVE UNTIL SQL_AFTER_GTIDS` did not cause the slave to stop until the next GTID event was received following execution of the transaction having the indicated GTID, which could cause issues in the case when the next GTID event is delayed, or does not exist. Now the slave stops after completing the transaction with that GTID. (Bug #14767986)
- **Important Change; Replication:** It was possible to start the server with binary logging enabled but no server ID specified; in such cases, the server would set `server_id` to 1 (rather than 0) while slaves remained unable to connect.

  Now `--server-id` must be used when starting the server with binary logging enabled, otherwise the server is unable to start. If `--server-id=0` is used, this value is no longer changed by the server; in this case, updates are written to the binary log, but slaves are unable to connect. Using `--server-id` without specifying a value has the same effect as setting it explicitly to 0. (Bug #11763963, Bug #56739)

- **InnoDB; Replication:** The `InnoDB` `mecached` plugin would update a record before inserting to the binary log, which would cause slave server replication to stop. The insert should occur before the update. (Bug #17358875)
- **InnoDB:** When new indexes are added by an `ALTER TABLE` operation, instead of only saving table-level statistics and statistics for the new indexes, `InnoDB` would save statistics for the entire table, including the table's other indexes. This behavior slowed `ALTER TABLE` performance. (Bug #17848838, Bug #16511145)
- **InnoDB:** A regression introduced by the fix for Bug#17371537 resulted a memory leak for `memcached` insert operations. (Bug #17738935)

  References: See also: Bug #17371537.
- **InnoDB:** Fault-tolerant code found in the log apply code for `InnoDB` `ALTER TABLE ... IN PLACE` could result in data corruption. (Bug #17625063, Bug #17512497)
- **InnoDB:** The `trx->error_key_num` field was not initialized in the error injection code found in `storage/innobase/handler/handler0alter.cc`. The `error_key_num` field is usually 0 but can be a non zero value if the memory buffer of a DDL transaction object is reused. (Bug #17624926)
- **InnoDB:** Databases names beginning with a digit would cause a full-text search (FTS) parser error. (Bug #17607956)

  References: See also: Bug #17161372.
- **InnoDB:** An `ALTER TABLE ... CHANGE [COLUMN]` operation would result in an `rbt_empty(index_cache->words)` assertion. (Bug #17536995)
- **InnoDB:** `buf_flush_event` would be created at flush thread startup instead of server startup. Also, `buf_flush_event` would be signaled when `InnoDB` is started in read-only mode. (Bug #17516062)
- **InnoDB:** `CHECK TABLE` would ignore the `QUICK` option. (Bug #17513737)
- **InnoDB:** An excessive amount of memory would be consumed when querying `INFORMATION_SCHEMA.INNODB_FT_INDEX_TABLE`. The problem would occur for very large full-text search indexes. (Bug #17483582, Bug #70329)
- **InnoDB:** Running `SHOW ENGINE INNODB STATUS` on one connection thread and killing that thread by running a `KILL CONNECTION` statement from a different connection thread would result in a severe error. (Bug #17474166)
- **InnoDB:** In debug builds, test case failures would occur due to `ibuf_contract_ext` performing merges and `dict_stats_update` returning evicted pages back into the buffer pool while `ibuf_change_buffering_debug` is enabled. (Bug #17446090)
- **InnoDB:** `InnoDB` failed to return an error when attempting to run a query after discarding the tablespace. (Bug #17431533)
- **InnoDB:** A severe error would occur after discarding a tablespace. (Bug #17430207)
- **InnoDB:** Data in the `OPERATION` column of `performance_schema.events_waits_current` table was incorrect due to a code regression introduced in MySQL 5.7.2. (Bug #17429480)
- **InnoDB:** During a `TRUNCATE TABLE` operation, `InnoDB: Trying to TRUNCATE a missing index of table ...` warnings would be printed to the error log. These warnings should not be printed when the index is a full-text search (FTS) index. (Bug #17402002, Bug #70226)

  References: See also: Bug #12429565.

- **InnoDB:** During parallel full-text search (FTS) index creation, a scanner thread reads in documents and passes them to the tokenizer. The tokenizer frees documents from memory when tokenization is complete. When tokenizing documents with a large amount of text, the tokenizer thread would not keep pace with the scanner thread. As a result, memory would not be freed fast enough and the “tokenization pending list” would grow in size. (Bug #17384979)
- **InnoDB:** `row_scan_index_for_mysql` would allocate a buffer size of `UNIV_PAGE_SIZE` for `row_search_for_mysql`. When the record length was greater than `UNIV_PAGE_SIZE`, a Valgrind error would occur. (Bug #17378106)
- **InnoDB:** `trx_create` and `trx_free` would be called for every `memcached get` request. (Bug #17371537, Bug #70172)
- **InnoDB:** A full-text search (FTS) `BOOLEAN MODE` query with an invalid character in the query string could result in a memory access violation failure. (Bug #17350055)
- **InnoDB:** An assertion would be raised when the database initialization thread encountered other threads performing buffer pool flushing. (Bug #17349975)
- **InnoDB:** Full-text index creation on a large table failed due to insufficient temporary table space and result in a misleading “incorrect key file” error. (Bug #17339606)
- **InnoDB:** The `UNIV_BLOB_DEBUG` compile-time debug flag, which is not often used and is limited by its inability to work across crash recovery, has been removed in order to simplify code refactoring. (Bug #17338452)
- **InnoDB:** The `UNIV_SEARCH_DEBUG` compile-time debug check, which has never reported a failure, has been removed along with all references to it. (Bug #17338432)
- **InnoDB:** `trx_sys_t::ro_trx_list` has been removed. Adding and removing transactions from the `ro_trx_list` can be costly, and placing transactions on the list by default is no longer necessary after mutex related optimizations implemented in MySQL 5.7.2. User transactions are still placed on the `trx_t::mysql_trx_list`. Currently, background read-only transactions are not placed on any list. (Bug #17332300)
- **InnoDB:** When `innodb_file_per_table` is set to `OFF`, replication failed with error code 1880 when truncating tables. For tables that reside in the shared tablespace, the truncate log file name used during the truncate action was not unique. The error is due to a MySQL 5.7.2 code regression related to Bug #14174004. (Bug #17327409)
- **InnoDB:** When `InnoDB` is retrieving rows and a `KILL QUERY` statement is issued, `InnoDB` would return false errors. (Bug #16950658)
- **InnoDB:** In `btr_validate_level` there are checks to ensure that all B-tree pages are marked when allocated. The checks failed on the change buffer because the allocation of change buffer pages is handled differently than other B-tree pages. (Bug #16884217)
- **InnoDB:** The hardcoded size for the `srv_max_n_threads` variable was insufficient. The variable setting is now configured based on the maximum number of connection threads and `InnoDB` background threads. (Bug #16884077)
- **InnoDB:** InnoDB would set `UNIV_WORD_SIZE` to 4 for both Windows 32-bit and 64-bit systems. With this patch, `UNIV_WORD_SIZE` is set to 8 on 64-bit Windows systems. This patch also removes `UNIV_WORD_ALIGNMENT`, which is no longer used. (Bug #16774645)
- **InnoDB:** A `SELECT COUNT(*)` query would take a long time to complete when run concurrently with a `LOAD DATA` operation. The `mtr_memo_contains` function, which determines if an object is part

of a memo in a mini transaction, contained a nested loop that caused the query to run slowly. (Bug #16764240, Bug #69141)

- **InnoDB:** When the change buffer is enabled, `InnoDB` failed to write a transaction log record when merging a record from the insert buffer to a secondary index page if the insert was performed as an "update-in-place". (Bug #16752251, Bug #69122)

- **InnoDB:** Due to a regression in MySQL 5.6, creating or dropping tables with `innodb_force_recovery` set to `3` (`SRV_FORCE_NO_TRX_UNDO`) failed. Additionally, this bug fix includes a code modification that sets `InnoDB` to read-only when `innodb_force_recovery` is set to a value greater than `3` (`SRV_FORCE_NO_TRX_UNDO`). (Bug #16631778, Bug #69892)

- **InnoDB:** An `InnoDB memcached` configuration error message contained an incorrect file name. The error message stated, `Please create config table containers in database innodb_memcache by running innodb_config.sql. error 31`. The correct file name is `innodb_memcached_config.sql`. Also, the "error 31" portion of the error message has been translated to its text equivalent, which is "Table not found". (Bug #16498810, Bug #68684)

- **InnoDB:** In `mutex_spin_wait()`, the `sync_array_reserve_cell` function could fail to find an empty slot on systems with sync wait arrays that are small in size, resulting in an error. (Bug #16245498)

- **InnoDB:** When `index_read_map` is called for an exact search and fails to return a record due to non-matching search criteria, the cursor would be positioned on the next record after the searched key. A subsequent call to `index_next` would return the next record instead of returning the previous non-matching row, thereby skipping a record. (Bug #14621190, Bug #15965874, Bug #17314241, Bug #70038, Bug #17413093, Bug #12860669, Bug #60220, Bug #17565888)

- **InnoDB:** An implicit rollback caused the server to halt when restarting with an `innodb_force_recovery` value of 3 or greater. This bug was addressed by the combination of fixes for Bug #16310467 and Bug #17253499. (Bug #14178835)

  References: See also: Bug #16310467, Bug #17253499.

- **InnoDB:** An infinite loop could occur in `buf_page_get_gen` when handling compressed-only pages. (Bug #12560151, Bug #61132)

- **InnoDB:** `InnoDB` would attempt to free `BLOB` pages already marked as free. (Bug #11762662, Bug #55284)

- **InnoDB:** Converting a table with a large number of columns from `MyISAM` to `InnoDB` would cause an assertion due to insufficient log buffer space. Instead of asserting, `InnoDB` now attempts to increase log buffer size automatically if the redo log size is too large. (Bug #11758196, Bug #50366)

- **Partitioning:** The storage engine was set incorrectly during a rebuild of a partition; the table storage engine was ignored and the default storage engine used instead. Thus, in MySQL 5.1, it was possible for `REBUILD PARTITION` to change the partition storage engine from `InnoDB` to `MyISAM`, and for the reverse (rebuilding partitions of `MyISAM` tables causing the partitions to use `InnoDB`) to occurin MySQL 5.5 and later. Now, when rebuilding partitions, the storage engine actually used by the table is checked and used by the handler for the rebuild operation, so that the partition storage engine is not inadvertently changed. (Bug #17559867)

- **Partitioning:** Index condition pushdown did not work with partitioned tables. (Bug #17306882, Bug #70001)

- **Partitioning:** After disabling the parent table's indexes with `ALTER TABLE ... DISABLE KEYS`, rebuilding any of its partitions enabled the indexes on those partitions, leading `MyISAM` to fail with an error when the optimizer tried to use one of the affected indexes.

Now in such cases, we check for disabled indexes on the table before rebuilding any of its partitions. If the indexes have been disabled, then we disable them on the partition following the rebuild. (Bug #16051817)

- **Replication:** When GTID-based replication was used with an intra-schema multi-threaded slave, transactions were assigned to the first worker thread only. (Bug #17590616, Bug #70536)
- **Replication:** The `WORKER_ID` column of the `replication_execute_status_by_worker` table did not use the internal `id` column values from the `mysql.slave_worker_info` table, as expected. (Bug #17514406, Bug #70426)
- **Replication:** The `THREAD_ID` column values shown in the `performance_schema.replication_execute_status_by_worker` table used the same thread IDs shown in the output from `SHOW PROCESSLIST`, rather than those used by other `performance_schema` tables. (Bug #17440991, Bug #70423)

  References: See also: Bug #17473308, Bug #17526982.
- **Replication:** The final argument in the `SET` clause of a `LOAD DATA ... SET` statement was repeated in the binary log. (Bug #17429677, Bug #70277)
- **Replication:** When an error encountered by the dump thread while reading events from the active binary log file was a temporary error, so that the dump thread tried to read the event, it was possible for the dump thread to seek the wrong position, which could cause one or more events to be resent. To prevent this, the thread's position is obtained after each correct read of an event.

  In addition, with this fix, only binary logs that are not closed normally are marked as possibly being corrupted.

  Finally, two warnings are added; these are now returned when a dump thread encounters a temporary error. (Bug #17402313)
- **Replication:** Setting `rpl_semi_sync_master_enabled` while the master was waiting for a reply from the slave could in some cases cause the master to fail. (Bug #17327454, Bug #70045)
- **Replication:** When stopping the I/O thread, it was possible with a very large transaction (equivalent to a binary log size greater than 100MB) that the thread did not receive the transaction to the end. When reconnecting with `MASTER_AUTO_POSITION=1` it then tried to fetch changes from the next transaction, which could lead to loss of the incomplete transaction and its data. (Bug #17280176, Bug #69943)
- **Replication:** Trying to set `CHANGE MASTER TO ... MASTER_AUTO_POSITION = 0` failed with error 1777 (**`ER_AUTO_POSITION_REQUIRES_GTID_MODE_ON`**). (Bug #17277744)
- **Replication:** No error was written to the log file when writing an incident event to the binary log. This meant that the user was not alerted that a slave server will later be stopped by the incident event. To prevent this from happening, error messages are now written in the log file for all incidents written to the binary log. (Bug #17258782)
- **Replication:** The value of `LAST_INSERT_ID()` was not correctly replicated when filtering rules were used on the slave. (Bug #17234370, Bug #69861)
- **Replication:** `RESET SLAVE ALL` reset only the host, port, user, password, and log positions. Now it resets all connection parameters. (Bug #17185647)
- **Replication:** An internal function used for storing GTID values could sometimes try to handle them as strings of the wrong length. (Bug #17032712, Bug #69618)

- **Replication:** During row-based replication with `binlog_row_image` set to `MINIMAL`, updating only some columns of a table having 9 or more columns caused `mysqlbinlog` to fail when it was used with the `--verbose` option. (Bug #16960133)
- **Replication:** When a master with semisynchronous replication enabled was shut down, the master failed to wait for either a semisyncnronous `ACK` or timeout before completing the shutdown. This prevented semisynchronous replication from reverting to asynchronous replication and allowed open transactions to complete on the master, which resulted in missing events on the slave.

  To fix this problem, dump threads are now stopped last during shutdown, after the client is told to stop, so that, if the dump thread has pending events from active clients, they can be sent to the slave. (Bug #16775543)
- **Replication:** Issuing a `GRANT` statement with invalid parameters caused the master to write `LOST_EVENTS` events into its binary logs, causing replication to stop. Now such cases, if one or more grants or revocations of privileges are successful, an incident is written to the log; otherwise, only a warning is logged. (Bug #16629195, Bug #68892)
- **Replication:** `START SLAVE` failed when the server was started with the options `--master-info-repository=TABLE` `relay-log-info-repository=TABLE` and with `autocommit` set to 0, together with `--skip-slave-start`.

  A workaround for previous versions of MySQL is to restart the slave `mysqld` without the `--skip-slave-start` option. (Bug #16533802)
- **Replication:** A number of unneeded initializations of objects that were used but not actually needed for reading the relay info log were removed. (Bug #16291602)
- **Replication:** `mysqlbinlog` now supports the same command-line options relating to SSL as `mysql`, `mysqladmin`, and other MySQL client programs. See Command Options for Secure Connections, for more information. (Bug #11751199, Bug #41975)
- **Replication:** Previous versions of `mysqlbinlog` did not correctly accept the `ssl-ca` option in an option file. This fix ensures that this option can be correctly used. In earlier versions a work around is to use the `loose-ssl-ca` option. (Bug #74864, Bug #20032654)
- For debug builds, an error occurring during `DELETE IGNORE` could raise an assertion. (Bug #17720294)
- `UNION ALL` statements for which one `SELECT` returned zero rows could result in an incorrect `FOUND_ROWS()` value. (Bug #17708480)
- Enabling Index Merge optimizer switches and setting a small `sort_buffer_size` value could lead to a server exit. (Bug #17617945)
- Some license and documentation files were missing from Windows MSI packages. (Bug #17584523)
- `UNION ALL` queries for which the last `SELECT` selected an aggregate value from an empty table resulted in an incorrect `FOUND_ROWS()` value. (Bug #17580869)
- Semi-join materialization strategy was not used for `VARCHAR` columns longer than 512 bytes, resulting in use of a less-efficient strategy and worse query performance. (The limit in characters rather than bytes depends on the column character set; 170 characters for `utf8`, for example.) (Bug #17566396)
- Disconnect processing overhead was reduced for sessions that have no outstanding prepared statements. Previously, a global mutex was acquired to handle these, but there is no need to do so in the absence of such statements. (Bug #17560986)
- The optimizer did not consider a clustered primary key as a covering index unless all columns required for a query were in the primary key definition. This incorrectly raised the calculated cost of using the

index and caused it not to be used in some cases when it would be more efficient than the index otherwise chosen. (Bug #17560636)

- Selecting from the `session_connect_attrs` Performance Schema table under high load could cause a server exit. (Bug #17542370)
- The Performance Schema had a race condition adding and deleting stored procedure entries that could raise an assertion. (Bug #17529279)
- `DROP TRIGGER` succeeded even with the `read_only` system variable enabled. (Bug #17503460)
- An incorrect result could be produced by a simple `COUNT(DISTINCT)` query on a table that contains a large number of distinct values. This was more likely when `tmp_table_size` or `max_heap_table_size` were set to small values. (Bug #17500866)
- Performance Schema memory instrumentation overhead was reduced.

  Memory allocated internally by the Performance Schema was not reported by the memory instrumentation. This was corrected, and there is now a `memory/performance_schema/internal_buffers` instrument (disabled by default) that can be enabled to obtain this information. (Bug #17493868)
- Stored programs were not listed in the `objects_summary_global_by_type` Performance Schema table. (Bug #17472833)
- `my_print_stacktrace()` printed a Reference Manual URL that pointed to the 5.1 manual. It now is parameterized for the current server series. (Bug #17465503)
- In debug builds, `SHOW PROCEDURE CODE` raised an assertion. (Bug #17434385)
- Compilation failures under Visual Studio 2012 were corrected. (Bug #17430236)
- For JSON-format `EXPLAIN` statements, materialized views were incorrectly labeled as `optimized_away_subqueries` rather than as `materialized_from_subquery`. (Bug #17428655)
- `KILL` with an invalid thread ID value could raise an assertion. (Bug #17420682)
- An assertion was raised if a statement tried to set an exception condition in a diagnostics area which already contained an exception or completion condition. This could occur if a failed stored program tried to transfer its exception condition to the diagnostics area of its caller that already contained a condition. (Bug #17400687)
- In the Performance Schema memory instrumentation, statistics collected for memory-free operations could be under-evaluated, leading to the appearance of a memory leak. (Bug #17400029)
- Compiling without the Performance Schema resulted in unresolved symbols. (Bug #17399658)
- Some warnings produced by `mysql_install_db` referred to the now-deprecated `log_warnings` system variable. These have been updated to refer to `log_error_verbosity` instead. (Bug #17363350)
- An assertion was raised if `SET PASSWORD` was used for an account that has been manually deleted from the `mysql.user` table but still present in memory. (Bug #17359329)
- Savepoints could not be used successfully following an `ER_LOCK_DEADLOCK` error (or `ER_LOCK_WAIT_TIMEOUT` error, if `innodb_rollback_on_timeout` was enabled). (Bug #17356954)

  References: This issue is a regression of: Bug #14188793.

- The `CLIENT_CONNECT_WITH_DB` flag was improperly handled in the C client library. This could lead to a malformed packet sent to the server. (Bug #17351732)
- Upgrades using RPM packages failed if the MySQL server was running due to failure to ignore the `mysqld_safe.pid` file created by `mysqld_safe`. (Bug #17343851)
- The `mysql_real_connect()` C API function could leak memory if it failed. (Bug #17337684)
- The `filesort` implementation sometimes failed to allocate enough buffer space, leading to a server exit. (Bug #17326567)
- Cost comparison for `ref` access could be inaccurate. (Bug #17303649)
- The `mysql_options()` C API function could leak memory if called more than once with the `MYSQL_SET_CLIENT_IP` option. (Bug #17297012)
- The `CONV()` function could call `abs(INT_MIN)`, which is undefined, and cause a server exit. (Bug #17296644)
- An error array in the SSL code was missing a comma, leading to implicit concatenation of adjacent messages and a resulting off-by-one error in the relationship between error numbers and messages. (Bug #17294150)
- Full-text search on `InnoDB` tables failed on searches that used the `+` boolean operator. (Bug #17280122)
- For single-threaded workloads, the optimizer recognizes some special cases for which it can avoid function calls and enhance performance. (Bug #17234723)
- The `my_load_path()` function could in some cases calculate a path value incorrectly. (Bug #17204851)
- Temporary table columns were marked as temporarily nullable without taking into account the presence of triggers for the table. This could lead to `NOT NULL` columns being updated to `NULL` by a multiple-table `UPDATE` statement. (Bug #17055378)
- In debug builds, an assert could be raised if a statement was killed while executing a trigger. (Bug #17049537)
- Within a stored program, comparison of the value of a scalar subquery with an `IN` clause resulted in an error for the first execution and raised an assertion for the second execution. (Bug #17029399)
- Information was not transferred between two optimizer modules because there were duplicate variables for the same information. This could lead to suboptimal query execution plans and incorrect query results. (Bug #16982071, Bug #70021, Bug #17310161)
- JSON-format `EXPLAIN` statements could leak memory. (Bug #16970785)
- Some queries with `ROLLUP` and an inner subquery with a reference to an outer field could raise an assertion. (Bug #16967281)
- `GRANT` without an `IDENTIFIED BY` clause resulted in an error even for existing users. (Bug #16938568)
- `GROUP_CONCAT()` with an invalid separator could cause a server exit. (Bug #16870783)
- For partitioned tables, queries could return different results depending on whether Index Merge was used. (Bug #16862316)

  References: See also: Bug #17648468, Bug #176588348, Bug #18167648.

- `mysqltest_embedded` does not work with the `--ps-protocol` option, so it now issues a warning if that option is given. (Bug #16817580)
- An internal `InnoDB` string routine could write past the end of a buffer. (Bug #16765410)
- GIS intersection-related code was missing a return value check, leading to a loop in nondebug builds and a raised assertion in debug builds. (Bug #16659166)
- For debug builds, when the optimizer removed an `Item_ref` pointing to a subquery, it caused a server exit. (Bug #16509874)

  References: This issue is a regression of: Bug #16318585.
- It is no longer possible to expire the password for anonymous-user accounts because an anonymous user cannot execute `SET PASSWORD` to reset the account password. (Bug #16483619)
- On Windows, a MySQL client program that simply used `#include <mysql.h>` could not be compiled due to missing Windows declarations in that file. The same program would compile on other platforms. (Bug #16409270)

  References: See also: Bug #17514554.
- `HANDLER READ` could cause a server exit due to wrongly considering columns as constant. (Bug #16386136)
- Using the binary client/server protocol, the second execution of a prepared statement for a query with parameters in the `LIMIT` clause raised an assertion. (Bug #16346241)
- Very long database names in queries could cause the server to exit. (Bug #15912213, Bug #16900358)
- Memory allocated for the Performance Schema was not freed at server shutdown. (Bug #14771682)
- If an `INSTALL PLUGIN` statement contained invalid UTF-8 characters in the shared library name, it caused the server to hang (or to raise an assertion in debug builds). (Bug #14653594)
- Standalone Windows MSI packages did not have the `ALLUSERS` property set. They now set `ALLUSERS=1`. For earlier MSI packages in this MySQL series, a workaround is to use the following command:

  ```
  C:\> msiexec /i msi_installer_name ALLUSERS=1
  ```

  (Bug #14647206)
- Deadlocks involving metadata locks and `InnoDB` deadlocks were both reported as an `ER_LOCK_DEADLOCK` error, but only `InnoDB` deadlocks rolled back the transaction. Now both deadlocks roll back the transaction. (Bug #14188793)
- Columns in a `PRIMARY KEY` must be `NOT NULL`, but if declared explicitly as `NULL` produced no error. Now an error occurs. For example, a statement such as `CREATE TABLE t (i INT NULL PRIMARY KEY)` is rejected. The same occurs for similar `ALTER TABLE` statements. (Bug #13995622, Bug #66987, Bug #15967545, Bug #16545198)
- Some `.pdb` files were missing from Windows Zip archive distributions. (Bug #13878021)
- Several issues identified by the Coverity static analysis tool were fixed. Thanks to Jan Staněk and Honza Horak for the patches. (Bug #70591, Bug #17590095)
- Setting `host_cache_size` at startup had no effect. (Bug #70552, Bug #17576516)

- MySQL did not compile on Mac OS X 10.9 (Mavericks). (Bug #70542, Bug #17647863)
- For `EXPLAIN FOR CONNECTION connection_id`, the parser did not permit *`connection_id`* values larger than the maximum signed `BIGINT` value. The maximum unsigned value now is permitted. (Bug #70533, Bug #17564492)
- `EXPLAIN FOR CONNECTION` reported no error if the connection ID belonged to no connection thread. (Bug #70532, Bug #17564493)
- In some cases, range conditions over indexes defined on column prefixes returned incomplete result sets. (For example, `SELECT ... WHERE 'abcdef1' < col_name AND col_name < 'abcdef9'`, where the index on *`col_name`* indexed only the first 6 characters.) (Bug #70341, Bug #17458273)
- `InnoDB` full-text searches failed to find records within transactions that included savepoints. (Bug #70333, Bug #17458835)
- Incorrect reference counting in the range optimizer module resulted in potential for missing or duplicate rows in the query result set. (Bug #70236, Bug #17405466)
- If asked to upgrade a server that was running without `InnoDB` enabled, `mysql_upgrade` issued complaints about `InnoDB` tables not existing (tables that will not exist unless `InnoDB` is available). (Bug #70152, Bug #17361912)
- With the thread pool plugin enabled, the `PROCESSLIST_USER` and `PROCESSLIST_HOST` columns of the `threads` Performance Schema table were always `NULL` for client sessions. Also, for the main thread, those columns were not `NULL` but set to a user account.

  

  Note

  As part of the bug fix implementation, Performance Schema instrumentation for the thread pool plugin was changed to use `thread_pool`, not `sql`.

  (Bug #70028, Bug #17310065, Bug #17049691)
- Performance Schema instrumentation overhead was reduced for frequent connect/disconnect operations. (Bug #70018, Bug #17310878)
- Full-text search on `InnoDB` tables failed on searches for words containing apostrophes when using boolean operators.

  The `innodb_ft_max_token_size` maximum value was incorrectly defined as 252, which is the maximum byte length. The maximum `innodb_ft_max_token_size` value is now 84, which is the maximum character length. (Bug #69932, Bug #17276125)
- `COUNT(DISTINCT)` should not count `NULL` values, but they were counted when the optimizer used Loose Index Scan. (Bug #69841, Bug #17222452)
- `InnoDB` deadlock caused transaction rollback but did not release metadata locks, blocking concurrent DDL on the transaction tables until the connection that got the deadlock issued an explicit `COMMIT` or `ROLLBACK`. (Bug #69668, Bug #17054007)
- In debug builds, static initialization code could call DBUG functions before the DBUG subsystem was initialized. (Bug #69653, Bug #17063675)
- Reads of `Geometry` values within a stored program could read already-freed memory and produce incorrect results. (Bug #69517, Bug #16985214)
- Specifying the same directory with multiple instances of `--ignore-db-dir` caused a server exit. (Bug #69441, Bug #16944177)

- For queries that qualify to be written to the slow query log, the check whether the log is enabled now occurs earlier, reducing overhead.

  Also, when `log_throttle_queries_not_using_indexes` is specified at server startup, a value is now required. Previously, it was incorrectly treated as optional. (Bug #69420, Bug #16924125)

- Some `INSERT INTO ... SELECT ... FROM` statements were slow unless the `tmp_table_size` and `max_heap_table_size` system variables were set large enough to permit the temporary table used for query processing to be stored in the `MEMORY` storage engine. (Bug #69368, Bug #16894092)

- Missing `va_end()` calls were added to logging and UCS2 code. Thanks to Jan Staněk for the patch. (Bug #68896, Bug #16725769)

- `ER_TRUNCATED_WRONG_VALUE` (truncated value) errors for `DECIMAL` values failed to show the erroneous input value. (Bug #68745, Bug #16552002)

- For queries of the form `UPDATE ... WHERE unique_key ORDER BY ... LIMIT ...`, incorrect rows could be updated. Unique keys permit multiple `NULL` values, but the optimizer did not always consider all of them. (Bug #68656, Bug #16482467)

- Within a stored function or trigger, occurrence of an `ER_LOCK_WAIT_TIMEOUT` error caused failure to execute a condition handler. (Bug #67947, Bug #16041903)

- With a `NULL` literal in a particular parameter position, `IFNULL()`, `COALESCE()`, and `CASE` returned a signed value when they should return an unsigned value. (Bug #65976, Bug #14359340)

- The server uses the ethernet hardware address for UUID generation, but made assumptions about the names of ethernet devices rather than querying the system for their names. Thanks to Honza Horak for the patch. (Bug #63055, Bug #13548252)

- Host names in grant tables are stored in lowercase, but `mysql_install_db` could fail to observe this convention, leading to accounts that could not be dropped with `DROP USER`. (Bug #62255, Bug #12917164, Bug #62254, Bug #12917151)

- If one connection changed its default database and simultaneously another connection executed `SHOW PROCESSLIST`, the second connection could access invalid memory when attempting to display the first connection's default database. memory. (Bug #58198, Bug #11765252)

- At server shutdown, a race condition between the main thread and the shutdown thread could cause server failure. (Bug #56666, Bug #11763896)

- Fixed a potential problem with the MySQL string function `strmov()`. Its behavior with respect to overlap of source and destination previously depended on native support for `stpcpy()`. (Bug #48864, Bug #17429539)

- For a column declared as a `PRIMARY KEY`, the MySQL sometimes unnecessarily added a `DEFAULT` clause. For example, for `CREATE TABLE t (a INT, PRIMARY KEY(a))`, a `DEFAULT 0` clause was added, whereas with `CREATE TABLE t (a INT PRIMARY KEY)`, it was not. (Bug #36723, Bug #11748566)

- Views containing `ORDER BY integer` could result in errors at view evaluation time. Consider these view definitions, which use `ORDER BY` with an ordinal number:

```
CREATE VIEW v1 AS SELECT x, y, z FROM t ORDER BY 2;
CREATE VIEW v2 AS SELECT x, 1, z FROM t ORDER BY 2;
```

In the first case, `ORDER BY 2` refers to a named column `y`. In the second case, it refers to a constant 1. For queries that select from either view fewer than 2 columns (the number named in the `ORDER BY` clause), an error occurred if the server evaluated the view using the MERGE algorithm. Examples:

```
mysql> SELECT x FROM v1;
ERROR 1054 (42S22): Unknown column '2' in 'order clause'
mysql> SELECT x FROM v2;
ERROR 1054 (42S22): Unknown column '2' in 'order clause'
```

To handle view definitions like this, the server now writes them differently into the `.frm` file that stores the view definition. This difference is visible with `SHOW CREATE VIEW`. Previously, the `.frm` file contained this for the `ORDER BY 2` clause:

```
For v1: ORDER BY 2
For v2: ORDER BY 2
```

Now, the `.frm` file contains this:

```
For v1: ORDER BY `t`.`y`
For v2: ORDER BY ''
```

That is, for `v1`, 2 is replaced by a reference to the name of the column referred to. For `v2`, 2 is replaced by a constant string expression (ordering by a constant has no effect, so ordering by any constant will do).

If you experience view-evaluation errors such as just described, drop and recreate the view so that the `.frm` file contains the updated view representation. Alternatively, for views like `v2` that order by a constant value, drop and recreate the view with no `ORDER BY` clause. (Bug #28695, Bug #11746789)

- Killing a query that is performing a `filesort` operation resulted in an `ER_SERVER_SHUTDOWN` (Server shutdown in progess) error. (Bug #18256, Bug #11745656)

Changes in MySQL 5.7.2 (2013-09-21, Milestone 12)

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.

- Authentication Notes
- Diagnostics Notes
- Logging Notes
- Performance Schema Notes
- RPM Notes
- Security Notes
- Semisynchronous Replication Notes

- Trigger Notes
- Functionality Added or Changed
- Bugs Fixed

Authentication Notes

- **Incompatible Change:** Previously, account rows in the `mysql.user` table could have an empty `plugin` column value. In this case, the server authenticated such an account using either the `mysql_native_password` or `mysql_old_password` plugin, depending on whether the password hash value in the `Password` column used native hashing or the older pre-4.1 hashing method. With the deprecation of old-format password hashes in MySQL 5.6.5, this heuristic for deciding which authentication plugin to use is unnecessary and it is desirable that `user` table rows always specify explicitly which authentication plugin applies.

  To that end, the `plugin` column is now defined to be non-`NULL` with a default value of `'mysql_native_password'`, and associated server operations require the column to be nonempty. In conjunction with this `plugin` column definition modification, several other changes have been made:

  - The `--default-authentication-plugin` command-line option is reimplemented as the `default_authentication_plugin` system variable. Its use at server startup is unchanged, but now the default plugin value can be examined at runtime using `SHOW VARIABLES` or `SELECT @@default_authentication_plugin`. The variable is read only and cannot be changed at runtime.
  - When `mysql_install_db` is run, it invokes the server to initialize the `mysql` database. The server now assigns every `user` table row a nonempty `plugin` column value. The value is `'mysql_native_password'` unless the `default_authentication_plugin` system variable is set otherwise at server startup.
  - `mysql_upgrade` checks `user` table rows and, for any row with an empty `plugin` column, sets that column to `'mysql_native_password'` or `'mysql_old_password'` depending on the hash format of the `Password` column value.
  - At startup, and at runtime when `FLUSH PRIVILEGES` is executed, the server checks `user` table rows. For any row with an empty `plugin` column, the server writes a warning to the error log of this form:

    ```
    [Warning] User entry 'user_name'@'host_name' has an empty plugin
    value. The user will be ignored and no one can login with this user
    anymore.
    ```

    To address this issue, execute `mysql_upgrade`.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate the `plugin` column change into the `mysql` database and assign the appropriate nonempty plugin value to any empty `plugin` column values. However, because the server now checks for and disables accounts with empty `plugin` column values, it is necessary to upgrade as follows.

  If you plan to to upgrade using the data directory from your existing MySQL installation:

  1. Stop the old server
  2. Upgrade the MySQL binaries in place (replace the old binaries with the new ones)
  3. Restart the server with the `--skip-grant-tables` option to disable privilege checking

4. Run `mysql_upgrade` to upgrade the system tables
5. Restart the server normally (without `--skip-grant-tables`)

If you plan to upgrade by reloading a dump file generated from your existing MySQL installation:

1. To generate the dump file, run `mysqldump` without the `--flush-privileges` option
2. Stop the old server
3. Upgrade the MySQL binaries in place (replace the old binaries with the new ones)
4. Restart the server with the `--skip-grant-tables` option to disable privilege checking
5. Reload the dump file (`mysql < dump_file`)
6. Run `mysql_upgrade` to upgrade the system tables
7. Restart the server normally (without `--skip-grant-tables`)

`mysql_upgrade` runs by default as the MySQL `root` user. For the preceding procedures, if the `root` password is expired when you run `mysql_upgrade`, you will see a message that your password is expired and that `mysql_upgrade` failed as a result. To correct this, reset the `root` password to unexpire it and run `mysql_upgrade` again:

```
shell> mysql -u root -p
Enter password: ****  <- enter root password here
mysql> SET PASSWORD = PASSWORD('root-password');
mysql> quit

shell> mysql_upgrade -p
Enter password: ****  <- enter root password here
```

The password-resetting statement normally does not work if the server is started with `--skip-grant-tables`, but the first invocation of `mysql_upgrade` flushes the privileges, so when you run `mysql`, the statement is accepted.

Diagnostics Notes

- **Incompatible Change:** Per the SQL standard, nondiagnostic statements should clear the diagnostics area when they begin executing. Previously, MySQL differed from this in that some nondiagnostic statements did not do this. MySQL now follows the SQL standard, which affects the content of the diagnostics area for some statements. Consequently, the result from statements such as `SHOW WARNINGS` that display the diagnostics area now differs somewhat:

  - The previous behavior: `SHOW WARNINGS` displays information about the conditions (errors, warnings, and notes) resulting from the most recent statement in the current session that generated messages. It shows nothing if the most recent statement used a table and generated no messages. (That is, statements that use a table but generate no messages clear the message list.) Statements that do not use tables and do not generate messages have no effect on the message list.
  - The new behavior: `SHOW WARNINGS` displays information about the conditions resulting from execution of the most recent nondiagnostic statement in the current session.

  The result from other diagnostic statements is affected similarly (`SHOW ERRORS`, `GET DIAGNOSTICS`).

  The following example demonstrates the difference in behavior.

Previously:

```
mysql> DROP TABLE test.no_such_table;
ERROR 1051 (42S02): Unknown table 'test.no_such_table'
mysql> SELECT @@warning_count;
Query OK, 0 rows affected (0.00 sec)
mysql> SHOW WARNINGS;
+-------+------+-----------------------------------+
| Level | Code | Message                           |
+-------+------+-----------------------------------+
| Error | 1051 | Unknown table 'test.no_such_table' |
+-------+------+-----------------------------------+
1 row in set (0.00 sec)
```

Here, the `SELECT` statement does not use tables and does not generate messages, so it does not change the diagnostics area. Consequently, `SHOW WARNINGS` output pertains to the `DROP TABLE` statement.

Now:

```
mysql> DROP TABLE test.no_such_table;
ERROR 1051 (42S02): Unknown table 'test.no_such_table'
mysql> SELECT @@warning_count;
Query OK, 0 rows affected (0.00 sec)
mysql> SHOW WARNINGS;
Empty set (0.00 sec)
```

Here, the `SELECT` statement clears the diagnostics area because it is a nondiagnostic statement. Consequently, `SHOW WARNINGS` output pertains to the `SELECT` statement (and is empty because the `SELECT` produces no messages).

An implication of this change in diagnostics area handling is that if you expect to display the warning count as well as the list of messages, you should list the messages first because selecting the `warning_count` value clears the message list. Alternatively, use `SHOW COUNT(*) WARNINGS` to display the count; this is recognized as a diagnostic statement and does not clear the diagnostics area. Similar considerations apply to use of `error_count`.

For compliance with the SQL standard, which states that diagnostics statements are not preparable, MySQL no longer supports the following as prepared statements:

- `SHOW WARNINGS`, `SHOW COUNT(*) WARNINGS`
- `SHOW ERRORS`, `SHOW COUNT(*) ERRORS`
- Statements containing any reference to the `warning_count` or `error_count` system variable.

In other words, those statements are now treated, in terms of preparability, the same as `GET DIAGNOSTICS`, which was already not preparable.

Logging Notes

- **Incompatible Change:** Several changes have been made to provide more logging control and more informative log messages:
  - The `log_error_verbosity` system variable now controls verbosity of the server in writing error, warning, and note messages to the error log. Permitted values are 1 (errors only), 2 (errors and warnings), 3 (errors, warnings, and notes), with a default of 3.

`log_error_verbosity` is preferred over, and should be used instead of, the older `log_warnings` system variable. See the description of `log_warnings` for information about how that variable relates to `log_error_verbosity` (Server System Variables). The `log_warnings` system variable and `--log-warnings` command-line option now are deprecated and will be removed in a future MySQL release.

Note

The effective default verbosity is different now. The previous default (`log_warnings=1`) corresponds to `log_error_verbosity=2`, but the default `log_error_verbosity` is 3. To achieve a logging level similar to the previous default, set `log_error_verbosity=2`.

- Default server verbosity is less when invoked with the `--bootstrap` option (such as is done by `mysql_install_db`): Only errors are written during the installation process so that they are less likely to be overlooked by the installer.

- The `log_timestamps` system variable has been introduced for control of the timestamp time zone of messages written to the error log, and of general query log and slow query log messages written to files. (It does not affect the time zone of general query log and slow query log messages written to log tables, but rows retrieved from those tables can be converted from the local system time zone to any desired time zone with `CONVERT_TZ()` or by setting the session `time_zone` system variable.)

Note

The default timestamp time zone is different now (`UTC` rather than the local system time zone). To restore the previous default, set `log_timestamps=SYSTEM`.

- The format of timestamps has changed for messages written to the error log, and for general query log and slow query log messages written to files. Timestamps are written using ISO 8601 / RFC 3339 format: `YYYY-MM-DDThh:mm:ss.uuuuuu` plus a tail value of `Z` signifying Zulu time (UTC) or `±hh:mm` (an offset from UTC). In addition, for the general query log file, timestamps are included in every message, not just when the second changes.

  The format of timestamps has also changed for messages written to the general query log and slow query log tables (`mysql.general_log`, `mysql.slow_log`), which now include fractional seconds. (The column type for timestamps has changed from `TIMESTAMP` to `TIMESTAMP(6)`.)

- Previously, the ID included in error log messages was the `mysqld` process ID. Now the ID is that of the thread within `mysqld` responsible for writing the message. This is more informative with respect to which part of the server produced the message. It is also more consistent with general query log and slow query log messages, which include the connection thread ID.

For information about log output destinations, see Selecting General Query and Slow Query Log Output Destinations. For information about specific logs, see The Error Log, The General Query Log, and The Slow Query Log.

Performance Schema Notes

- The Performance Schema now provides tables that expose replication information. This is similar to the information available from the `SHOW SLAVE STATUS` statement, but representation in table form is more accessible and has usability benefits:

- `SHOW SLAVE STATUS` output is useful for visual inspection, but not so much for programmatic use. By contrast, using the Performance Schema tables, information about slave status can be searched using general `SELECT` queries, including complex `WHERE` conditions, joins, and so forth.
- Query results can be saved in tables for further analysis, or assigned to variables and thus used in stored procedures.
- The replication tables provide better diagnostic information. For multi-threaded slave operation, `SHOW SLAVE STATUS` reports all coordinator and worker thread errors using the `Last_SQL_Errno` and `Last_SQL_Error` fields, so only the most recent of those errors is visible and information can be lost. The replication tables store errors on a per-thread basis without loss of information.
- The last seen transaction is visible in the replication tables on a per-worker basis. This is information not avilable from `SHOW SLAVE STATUS`.
- Developers familiar with the Performance Schema interface can extend the replication tables to provide additional information by adding rows to the tables.

These tables provide replication information:

- `replication_connection_configuration` and `replication_connection_status` indicate the configuration parameters used by the slave for connecting to the master and the status of the connection.
- `replication_execute_configuration` and `replication_execute_status` indicate, for aspects of transaction execution on the slave not specific to any given thread, the configuration parameters and the current execution status.
- `replication_execute_status_by_coordinator` and `replication_execute_status_by_worker` contain thread-specific transaction execution information, either about the SQL thread (for a single-threaded slave), or about the coordinator and worker threads (for a multi-threaded slave).

If the slave is multi-threaded, the SQL thread is the coordinator for worker threads. In this case, the `Last_SQL_Error` field of `SHOW SLAVE STATUS` output now shows exactly what the `Last_Error_Message` column in the `replication_execute_status_by_coordinator` Performance Schema table shows. The field value is modified to suggest that there may be more failures in the other worker threads which can be seen in the `replication_execute_status_by_worker` table that shows each worker thread's status.

For more information, see Performance Schema Replication 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.

- The Performance Schema now instruments stored program execution and aggregates statistics for them. This includes stored procedures, stored functions, triggers, and Event Scheduler events.

  These specific changes were implemented:

  - The `setup_instruments` table has new instruments. The `statement/scheduler/event` instrument tracks all events executed by the Event Scheduler. Instruments with names of the form `statement/sp/`*`program_instruction`* track internal instructions executed by stored programs.

- The `setup_objects` table `OBJECT_TYPE` column now permits values of `'EVENT'`, `'FUNCTION'`, `'PROCEDURE'`, `'TABLE'`, or `'TRIGGER'`, not just `'TABLE'`.
- Statement event tables (`events_statements_current`, `events_statements_history`, and `events_statements_history_long`) now have a `NESTING_LEVEL` column that indicates the event nesting level.
- The `performance_schema_max_program_instances` and `performance_schema_max_statement_stack` system variables configure the maximum number of stored programs and the maximum depth of nested stored program calls for which the Performance Schema maintains statistics.
- The `Performance_schema_program_lost` and `Performance_schema_nested_statement_lost` status variables indicate the number of stored programs for which statistics were lost, and the number of stored program statements for which statistics were lost.
- The `events_statements_summary_by_program` summary table aggregates statement events per stored program.

For more information, see Event Pre-Filtering, and 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` database.

- The Performance Schema now instruments memory usage and aggregates memory usage statistics, detailed by these factors:
  - Type of memory used (various caches, internal buffers, and so forth)
  - Thread, account, user, host indirectly performing the memory operation

  The Performance Schema instruments the following aspects of memory use
  - Memory sizes used
  - Operation counts
  - Low and high water marks

  Memory sizes help to understand or tune the memory consumption of a server.

  Operation counts help to understand or tune the overall pressure the server is putting on the memory allocator, which has an impact on performance. Allocating a single byte one million times is not the same as allocating one million bytes a single time; tracking both sizes and counts can expose the difference.

  Low and high water marks are critical to detect workload spikes, overall workload stability, and possible memory leaks.

  These specific changes were implemented:
  - The `setup_instruments` table now has memory instruments. These have names of the form `memory/`*`component/instrument_name`*. Memory instrumentation is disabled by default.
  - The `performance_schema_max_memory_classes` system variable configures the maximum number of memory instruments.

- The `Performance_schema_memory_classes_lost` status variable indicates the number of times a memory instrument could not be loaded.
- Several summary tables aggregate memory-related events.

For more information, see Memory 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.

RPM Notes

- It was not possible to upgrade a community RPM to a commercial RPM using `rpm -uvh` or `yum localupdate`. To deal with this, the RPM spec file has been updated in MySQL 5.7.2, which has the following consequences:
  - For a non-upgrade installation (no existing MySQL version installed), it possible to install MySQL using `yum`.
  - For upgrades, it is necessary to clean up any earlier MySQL installations. In effect, the update is performed by removing the old installations and installing the new one.

  Additional details follow.

  For a non-upgrade installation of MySQL 5.7.2, it is possible to install using `yum`:

```
shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
```

  For upgrades to MySQL 5.7.2, the upgrade is performed by removing the old installation and installing the new one. To do this, use the following procedure:

  1. Remove the existing 5.7.*X* installation. *`OLDVERSION`* is the version to remove.

```
shell> rpm -e MySQL-server-OLDVERSION.glibc23.i386.rpm
```

     Repeat this step for all installed MySQL RPMs.

  2. Install the new version. *`NEWVERSION`* is the version to install.

```
shell> rpm -ivh MySQL-server-NEWVERSION.glibc23.i386.rpm
```

  Alternatively, the removal and installation can be done using `yum`:

```
shell> yum remove MySQL-server-OLDVERSION.glibc23.i386.rpm
shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
```

  (Bug #16445097, Bug #16445125, Bug #16587285)

Security Notes

- Platform availability, usability, and security of `mysql_secure_installation` has been improved. Previously, this program was a shell script available for Unix and Unix-like systems. It has been converted to a binary executable program (written in C++) that is available on all platforms. Implementation as a C++ program permits `mysql_secure_installation` to connect directly

to the MySQL server using the client/server protocol, rather than by invoking `mysql` to do so and communicating with `mysql` using temporary files.

This reimplementation of `mysql_secure_installation` is feature-compatible with previous versions, but the following usability improvements have been made:

- The `validate_password` plugin can be used for password strength checking.
- Standard MySQL options such as `--host` and `--port` are supported on the command line and in option files.

For more information, see `mysql_secure_installation` — Improve MySQL Installation Security. For more information about `validate_password`, see The Password Validation Plugin.

Semisynchronous Replication Notes

- **Replication:** Semisynchronous replication master servers now use a different wait point by default in communicating wih slaves. This is the point at which the master waits for acknowledgment of transaction receipt by a slave before returning a status to the client that committed the transaction. The wait point is controlled by the new `rpl_semi_sync_master_wait_point` system variable. These values are permitted:
  - `AFTER_SYNC` (the default): The master writes each transaction to its binary log and the slave, and syncs the binary log to disk. The master waits for slave acknowledgment of transaction receipt after the sync. Upon receiving acknowledgment, the master commits the transaction to the storage engine and returns a result to the client, which then can proceed.
  - `AFTER_COMMIT`: The master writes each transaction to its binary log and the slave, syncs the binary log, and commits the transaction to the storage engine. The master waits for slave acknowledgment of transaction receipt after the commit. Upon receiving acknowledgment, the master returns a result to the client, which then can proceed.

    For older versions of MySQL, semisynchronous master behavior is equivalent to a setting of `AFTER_COMMIT`.

  The replication characteristics of these settings differ as follows:

  - With `AFTER_SYNC`, all clients see the committed transaction at the same time: After it has been acknowledged by the slave and committed to the storage engine on the master. Thus, all clients see the same data on the master.

    In the event of master failure, all transactions committed on the master have been replicated to the slave (saved to its relay log). A crash of the master and failover to the slave is lossless because the slave is up to date.
  - With `AFTER_COMMIT`, the client issuing the transaction gets a return status only after the server commits to the storage engine and receives slave acknowledgment. After the commit and before slave acknowledgment, other clients can see the committed transaction before the committing client.

    If something goes wrong such that the slave does not process the transaction, then in the event of a master crash and failover to the slave, it is possible that such clients will see a loss of data relative to what they saw on the master.

  The new wait point is a behavior change, but requires no reconfiguration. The change does introduce a version compatibility constraint because it increments the semisynchronous interface version: Servers

for MySQL 5.7.2 and up do not work with semisynchronous replication plugins from older versions, nor do servers from older versions work with semisynchronous replication plugins for MySQL 5.7.2 and up.

Trigger Notes

- Previously, a table could have at most one trigger for each combination of trigger event (`INSERT`, `UPDATE`, `DELETE`) and action time (`BEFORE`, `AFTER`). This limitation has been lifted and multiple triggers are permitted. Along with that change, several additional modifications were made:

  - By default, triggers for each combination of trigger event and action time execute in the order they were created. To make it possible to specify trigger activation order, `CREATE TRIGGER` now supports `FOLLOWS` and `PRECEDES` clauses. Each clause takes the name of an existing trigger that has the same trigger event and action time.

  - The `ACTION_ORDER` column in the `INFORMATION_SCHEMA.TRIGGERS` table is no longer 0 but an integer greater than zero that indicates the order in which triggers activate.

  - Creation time for triggers is now maintained, as a `TIMESTAMP(2)` value (with a fractional part in hundredths of seconds):

    - The `CREATED` column in the `TRIGGERS` table is no longer `NULL`, for triggers created as of MySQL 5.7.2.

    - The same is true for the `Created` column of `SHOW TRIGGERS` output, and for the (new) `Created` column of `SHOW CREATE TRIGGER` output.

    - The *`tbl_name`*`.TRG` file that stores trigger information for table *`tbl_name`* now contains a `created` line with trigger creation times.

    For additional information, see Using Triggers, CREATE TRIGGER Syntax, SHOW CREATE TRIGGER Syntax, SHOW TRIGGERS Syntax, The INFORMATION_SCHEMA TRIGGERS Table, and Table Trigger Storage.

  - If run against a table that has triggers, `mysql_upgrade` and `CHECK TABLE ... FOR UPGRADE` display this warning for each trigger created before MySQL 5.7.2:

    ```
    Trigger db_name.tbl_name.trigger_name does not have CREATED attribute.
    ```

    The warning is informational only. No change is made to the trigger.

  These changes have implications for backups, upgrades, and downgrades, as described following. For brevity, “multiple triggers” here is shorthand for “multiple triggers that have the same trigger event and action time.”

  Backup and restore. `mysqldump` dumps triggers in activation order so that when the dump file is reloaded, triggers are re-created in the same activation order. However, if a `mysqldump` dump file contains multiple triggers for a table that have the same trigger event and action time, an error occurs for attempts to load the dump file into an older server that does not support multiple triggers. (See the downgrading notes for a workaround; you can convert triggers to be compatible with older servers.)

  Upgrades. Suppose that you upgrade an old server that does not support multiple triggers to MySQL 5.7.2 or newer. If the new server is a replication master and has old slaves that do not support multiple triggers, an error occurs on those slaves if a trigger is created on the master for a table that already has a trigger with the same trigger event and action time. To avoid this problem, upgrade the slaves first, then upgrade the master.

Downgrades. If you downgrade a server that supports multiple triggers to an older version that does not, the downgrade has these effects:

- For each table that has triggers, all trigger definitions remain in the `.TRG` file for the table. However, if there are multiple triggers with the same trigger event and action time, the server executes only one of them when the trigger event occurs. For information about `.TRG` files, see Table Trigger Storage.
- If triggers for the table are added or dropped subsequent to the downgrade, the server rewrites the table's `.TRG` file. The rewritten file retains only one trigger per combination of trigger event and action time; the others are lost.

To avoid these problems, modify your triggers before downgrading. For each table that has multiple triggers per combination of trigger event and action time, convert each such set of triggers to a single trigger as follows:

1. For each trigger, create a stored routine that contains all the code in the trigger. Values accessed using `NEW` and `OLD` can be passed to the routine using parameters. If the trigger needs a single result value from the code, you can put the code in a stored function and have the function return the value. If the trigger needs multiple result values from the code, you can put the code in a stored procedure and return the values using `OUT` parameters.
2. Drop all triggers for the table.
3. Create one new trigger for the table that invokes the stored routines just created. The effect for this trigger is thus the same as the multiple triggers it replaces.

Functionality Added or Changed

- **Performance; InnoDB:** Multi-version concurrency control (MVCC) in `InnoDB` requires that each transaction using MVCC be assigned a read view. To improve `InnoDB` read-only and read-write performance, read view creation has been optimized by reducing mutex contention.
- **Performance; InnoDB:** An `sx-lock`, a new type of rw-lock, optimizes concurrency and improves scalability for read-write workloads. `sx-locks` reduce locking contention for B-tree index operations by providing write access while permitting inconsistent reads by other threads.
- **Performance; InnoDB:** Memory for transaction instances (`trx_t`) is now allocated in configurable sized blocks that are a multiple of transaction instance size. Transaction instances are also placed in a priority queue and ordered by their address in memory so that when instances are allocated from the pool, they are close together. This enhancement reduces the cost incurred by iterating over transactions instances when allocating instances from the pool.
- **Incompatible Change:** Previously, the Performance Schema statement instrumentation did not include statements executed on a slave replication server. To address this, a new abstract instrument, `statement/rpl/relay_log`, has been added to the `setup_instruments` table. This instrument is used during the early stages of replicated statement classification before the exact statement type is known. (Bug #16750433, Bug #17271055)
- **Incompatible Change:** Previously, the main loop responsible for accepting client connections also performed initialization of data structures related to each connection. These initialization tasks now are delegated to worker threads to minimize work done by the accept loop and maximize connection acceptance rate.

  As a result of this change, the `bind_address`, `thread_cache_size`, and `thread_handling` system variables are no longer visible to the embedded server (`libmysqld`). Similarly, the `Slow_launch_threads` and `Threads_cached` status variables are not meaningful in the embedded

server. These variables are no longer visible within the embedded server and embedded applications that rely on these variables should be modified accordingly. (Bug #62288, Bug #12951536, Bug #62284, Bug #12951595, Bug #62283, Bug #12951605)

- **Incompatible Change:** The unused `--basedir` and `--datadir` options for `mysql_upgrade` have been removed.

- **Important Change; Partitioning:** It is now possible to check and to repair partitions which contain duplicate key violations. This is implemented by allowing the `IGNORE` keyword in `ALTER TABLE` statements using the `CHECK PARTITION` or `REPAIR PARTITION` option. The keyword has the following effects on the behavior of these statements:

  - `ALTER IGNORE TABLE ... REPAIR PARITION` removes all rows that cannot be moved due to the presence of duplicate keys.

  - `ALTER IGNORE TABLE ... CHECK PARTITION` reports the contents of all columns in the partitioning expression for each row.

  

  Note

  Support for the `IGNORE` keyword is removed in MySQL 5.7.4.

  (Bug #16900947)

- **Important Change; Replication:** By default, when promoting integers from a smaller type on the master to a larger type on the slave (for example, from a `SMALLINT` column on the master to a `BIGINT` column on the slave), the promoted values are treated as though they are signed. Now in such cases it is possible to modify or override this behavior using one or both of `ALL_SIGNED`, `ALL_UNSIGNED` in the set of values specified for the `slave_type_conversions` server system variable. For more information, see Row-based replication: attribute promotion and demotion, as well as the description of the variable. (Bug #15831300)

- **InnoDB:** `innochecksum` functionality has been enhanced with new options and extended capabilities. See `innochecksum` — Offline InnoDB File Checksum Utility. (Bug #16945722)

- **InnoDB:** A new `CMake` option, `WITH_INNODB_EXTRA_DEBUG`, has been added that enables additional InnoDB debug checks. `WITH_INNODB_EXTRA_DEBUG` can only be enabled when the `WITH_DEBUG` option is also enabled. (Bug #16821155)

- **InnoDB:** When building MySQL from source, you can now define the type of mutex used by `InnoDB` using the new `MUTEX_TYPE` `CMake` option.

- **InnoDB:** Refactored mutex code makes selecting the appropriate mutex easier and allows multiple mutex types to be combined in the same instance. The refactored code also removes the distinction between `fast_mutex_t` and home brew `ib_mutex_t` types, implements a common interface for both mutex types, and allows new mutex types to be added in the future. Additionally, mutex code is decoupled from `InnoDB` code so that it can be used as a library, and a “PolicyMutex” interface has been introduced. The new interface uses static inheritance (templates) for mutex implementation making it easier to define policies and customize mutex behavior.

- **InnoDB:** Buffer pool list scans and related batch processing have been optimized to reduce scan complexity and the number of pages scanned.

- **InnoDB:** `InnoDB` buffer pool dump and load operations have been enhanced. A new system variable, `innodb_buffer_pool_dump_pct`, allows you to specify the percentage of most recently used pages in each buffer pool to read out and dump. When there is other I/O activity being performed by `InnoDB`

background tasks, InnoDB attempts to limit the number of buffer pool load operations per second using the innodb_io_capacity setting.

- **InnoDB:** DML operations (INSERT, UPDATE, DELETE) for temporary tables have been optimized by turning off redo logging, locking, and change buffering that is not required for temporary tables. Turning off these functions optimizes temporary table DML operations by reducing associated I/O.

- **InnoDB:** The limit on concurrent data-modifying transactions is now 96 * 1023 transactions that generate undo records. As of MySQL 5.7.2, 32 of 128 rollback segments are allocated to non-redo logs for transactions that modify temporary tables and related objects. This reduces the maximum number of concurrent data-modifying transactions from 128K to 96K. The 96K limit assumes that transactions do not modify temporary tables. If all data-modifying transactions also modify temporary tables, the limit would be 32K concurrent transactions.

- **InnoDB:** MySQL 5.7.2 introduces a new type of undo log for both normal and compressed temporary tables and related objects. The new type of undo log is not a redo log, as temporary tables are not recovered during crash recovery and do not require redo logs. The new undo log resides in the temporary tablespace. The default temporary tablespace file, ibtmp1, is located in the data directory by default and is always recreated on server startup. A user defined location for the temporary tablespace file can be specified by setting innodb_temp_data_file_path. For more information, see InnoDB Temporary Table Undo Logs.

- **InnoDB:** Read-only transactions will no longer be assigned a transaction ID. Conversely, an ID will only be assigned if a transaction is explicitly tagged as “read-write”, if a transaction has acquired an X or IX lock on a table, or if a transaction is a read-only transaction writing to a temporary table. All other transactions are considered “read-only” and are not assigned an ID. Furthermore, read-only transactions are not tagged as “read-only” unless they are explicitly started with START TRANSACTION READ ONLY. For transactions without transaction IDs, SHOW ENGINE INNODB STATUS prints an identifier that is unique but only within the context of the SHOW ENGINE INNODB STATUS invocation.

- **InnoDB:** For SELECT COUNT (*) queries, where a table's committed record count is changed by transaction deltas, there is now a single handler call to the storage engine to return the record count to the optimizer instead of one call for each record. This change generally improves SELECT COUNT (*) query performance and reduces in-memory table scan cost, as each record is no longer returned to the optimizer.

  In some instances, however, where there is a large clustered index and a very small secondary index, performance may not be improved. Previously, the optimizer would choose to traverse the smaller secondary index instead of the larger clustered index. The smaller secondary index could, in this case, offer better performance than a clustered index with a single handler call to the storage engine. However, there may be no performance benefit if the secondary index is often updated. When a secondary index page is modified by a transaction that is more recent than the COUNT(*) transaction, InnoDB must read the clustered index to determine if the record is visible. In this case, InnoDB would read both the secondary and clustered index, which is costlier than reading only the clustered index.

- **InnoDB:** Beginning with MySQL 5.7.2, UPDATE_TIME displays a timestamp value for the last UPDATE, INSERT, or DELETE performed on InnoDB tables. Previously, UPDATE_TIME displayed a NULL value for InnoDB tables. For MVCC, the timestamp value reflects the COMMIT time, which is considered the last update time. Timestamps are not persisted when the server is restarted or when the table is evicted from the InnoDB data dictionary cache.

- **InnoDB:** The process for converting a transaction's implicit lock to an explicit lock has been optimized to improve performance. The optimization reduces lock_sys_t::mutex contention.

- **InnoDB:** A number of internal debug flags in the InnoDB code could only be set at compilation time or from a debugger. As a result, a significant amount of diagnostic information was unused. This

enhancement replaces internal debug flags with DBUG labels so that the DBUG package can be used and printouts from various `InnoDB` subsystems can be enabled using the `mysqld --debug` command line option. See the Debugging a MySQL Server section for information about configuring MySQL for debugging, creating trace files, and using the `mysqld --debug` option.

- **Partitioning:** The following operations are now supported for individual subpartitions as well as partitions: `ANALYZE`, `CHECK`, `OPTIMIZE`, `REPAIR`, and `TRUNCATE` (see ALTER TABLE Partition Operations).

  

  Note

  This fix also allows the use of `REBUILD` with individual subpartitions, even though this is not actually supported by MySQL, and has no effect. This issue is addressed in MySQL 5.7.5 and later by disallowing `REBUILD` with subpartitions in `ALTER TABLE` statements.

  (Bug #14028340, Bug #65184)

  References: See also: Bug #19075411, Bug #73130.

- **Replication:** Previously, transactions could be applied in parallel only if they did not touch the same database. However, the MySQL Server uses a lock-based scheduler, which means that it should be possible to execute in parallel all uncommitted replication threads already in the prepare phase, without violating consistency. Such parallel execution can now be enabled on the slave by starting the slave `mysqld` with `--slave-parallel-type=LOGICAL_CLOCK` or, if `mysqld` is already started, by setting the value of the global system variable `slave_parallel_type` to `'LOGICAL_CLOCK'` on a stopped slave.

  When this feature is enabled, each transaction is marked with a logical timestamp. This timestamp identifies the last transaction committed at the time that the current transaction entered the prepare stage, and all transactions having the same timestamp can execute in parallel.

  To disable this feature without restarting, stop the slave using `STOP SLAVE` (if it is running as a slave), issue `SET @global-slave_parallel_type='DATABASE'`, then issue `START SLAVE` when you want the slave to resume. You can also disable the feature by restarting the slave `mysqld` without the `--slave-parallel-type` option, or by setting it explicitly to `DATABASE`. When parallel execution of preapred transactions is disabled, the slave follows the old behavior and applies in parallel only those transactions that do not cause changes in the same database.

- **Replication:** The master dump thread was refactored to reduce lock contention and improve master throughput. Previously, the dump thread took a lock on the binary log whenever reading an event; now the lock is held only while reading the position at the end of the last successfully written event. This means that multiple dump threads can now read concurrently from the binary log file, and that dump threads can read while clients write to the binary log.

- Support for LinuxThreads has been removed from the source code. LinuxThreads was superseded by NPTL in Linux 2.6. (Bug #17007529, Bug #72888, Bug #18913935)

- Previously, program options could be specified in full or as any unambiguous prefix. For example, the `--compress` option could be given to `mysqldump` as `--compr`, but not as `--comp` because the latter is ambiguous. Option prefixes are no longer supported; only full options are accepted. This is because prefixes can cause problems when new options are implemented for programs and a prefix that is currently unambiguous might become ambiguous in the future. Some implications of this change:

  - The `--key-buffer` option must now be specified as `--key-buffer-size`.
  - The `--skip-grant` option must now be specified as `--skip-grant-tables`.

(Bug #16996656)

- Support for building Apple universal binaries to support PowerPC has been removed from the source code. (Bug #16959103)
- `CMake` no longer checks for `memmove()` or `memcpy()` because they are standard C library functions. Also, implementation of the `bmove_upp()` function was replaced with calls to `memmove()`, which may have positive performance implications. (Bug #16839824)
- The C API `libmysqlclient` shared-library `.so` files now have version 18.1.0 (up from version 18.0.0 used in MySQL 5.5). 18.1.0 can be used as a replacement for 18.0.0. (Bug #16809055, Bug #59106, Bug #12407476)
- Use of `DYNAMIC_ARRAY` was reduced, which improves performance of certain range queries by 3-4%. (Bug #16736776, Bug #17030235)
- `mysqladmin` now supports a `--show-warnings` option to display warnings resulting from execution of statements sent to the server. (Bug #16517756)
- `mysql_upgrade` now verifies that the server version matches the version against which it was compiled, and exits if there is a mismatch. In addiion, a `--version-check` option permits specifying whether to enable version checking (the default), or disable checking if given as `--skip-version-checking`. (Bug #16500013)
- The following items are deprecated and will be removed in a future MySQL release. Where alternatives are shown, applications should be updated to use them.
  - The `ENCODE()` and `DECODE()` functions. Consider using `AES_ENCRYPT()` and `AES_DECRYPT()` instead.
  - The `INFORMATION_SCHEMA.PROFILING` table. Use the Performance Schema instead; see MySQL Performance Schema.

  (Bug #16463921)
- Invoking `CMake` with `-DWITH_AUTHENTICATION_PAM=1` now causes the build to fail (rather than issue only a warning) if the PAM plugin cannot be built. (Bug #14554639)
- In batch mode, `mysql` formatted result status messages such as ""Query OK, 1 row affected"" but did not print them. Now these messages are not formatted. (Bug #69486, Bug #16971432)
- Several inefficiencies were corrected:
  - A loop in `Item_in_subselect::single_value_transformer()` could execute too many times.
  - The `myisamchk()`, `my_test_if_sort_rep()`, and `recreate_table()` functions in `MyISAM` code could execute too many times.

  Thanks to Po-Chun Chang for the patches to correct these issues. (Bug #69138, Bug #16764131, Bug #69117, Bug #16751784, Bug #69561, Bug #17007268, Bug #69553, Bug #17001703)
- Plugins can now define and expose floating-point system variables of type double using the `MYSQL_SYSVAR_DOUBLE()` and `MYSQL_THDVAR_DOUBLE()` accessor macros. See Client Plugin Descriptors. (Bug #68121, Bug #16194302)
- Work was done to clean up the source code base, including: Removal of unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removal of function declarations without definitions.

- MySQL now supports the use of protocol trace plugins: client-side plugins that implement tracing of communication between a client and the server that takes place using the client/server protocol. Protocol trace plugins use the client plugin API.

  In MySQL source distributions, a test protocol trace plugin is implemented in the `test_trace_plugin.cc` file in the `libmysql` directory. This can be examined as a guide to writing other protocol trace plugins.

  For more information, see Writing Plugins; in particular, Writing Protocol Trace Plugins.

- To make it easier to see the difference between good and bad execution plans, JSON-format `EXPLAIN` output now includes this additional cost information:

  - `query_cost`: The total cost of a query block, whether a top-level query or subquery. For a top-level `SELECT`, this should be equal to the `Last_query_cost` status variable.
  - `sort_cost`: The cost of the first sorting operation (`GROUP BY` or `ORDER BY`) where and if `filesort` is used.
  - `read_cost`: The cost of reading data from each table used in the query block (that is, access method cost).
  - `eval_cost`: The cost of condition evaluation for each table in the query block.
  - `prefix_cost`: The cost of executing prefix join in the query block; that is, the cost of joining tables of the query block from the first one to the one (and including it) for which the value is given.
  - `data_read_per_join`: The estimated amount of data processed by the handler interface per query or subquery execution. This is essentially record width * number of read records.
  - `rows_produced_per_join`/ `rows_examined_per_join`: The estimated number of records from the table (per table from the query block) produced or examined per single query block execution.
  - `used_columns`: The list of columns from the table (per each table in the query block) used for either read or write in the query.

  This cost information is not displayed for `INFORMATION_SCHEMA` tables.

- `EXPLAIN` can now be used to obtain the execution plan for an explainable statement executing in a named connection:

  ```
  EXPLAIN [options] FOR CONNECTION connection_id;
  ```

  For example, if you are running a statement in one session that is taking a long time to complete, using `EXPLAIN FOR CONNECTION` in another session may yield useful information about the cause of the delay and help you optimize the statement.

  `connection_id` is the connection identifier, as obtained from the `INFORMATION_SCHEMA PROCESSLIST` table or the `SHOW PROCESSLIST` statement. If you have the `PROCESS` privilege, you can specify the identifier for any connection. Otherwise, you can specify the identifier only for your own connections.

  Changes in `EXPLAIN` output:

- In the output from `EXPLAIN FOR CONNECTION`, an `Extra` value of `Plan isn't ready yet` means that the optimizer has not finished creating the execution plan for the statement executing in the named connection. (For JSON-format output, this is indicated by `planned: false`.)

- In the output from any `EXPLAIN` used to obtain the execution plan for non-`SELECT` statements, the `select_type` value displays the statement type for affected tables. For example, `select_type` is `DELETE` for `DELETE` statements.

A new status variable, `Com_explain_other`, indicates how many `EXPLAIN FOR CONNECTION` statements have been executed.

For more information, see EXPLAIN Syntax, and Obtaining Execution Plan Information for a Named Connection.

- Semi-join LooseScan strategy now can use `ref` access and applies to a wider range of queries.

Bugs Fixed

- **Performance; Important Change; InnoDB:** `InnoDB` failed to open a tablespace that has multiple data files. This removes the known limitation that was in MySQL Server 5.6.12. (Bug #17033706, Bug #69623)

- **Performance; InnoDB:** A code regression introduced in MySQL 5.6 negatively impacted `DROP TABLE` and `ALTER TABLE` performance. This could cause a performance drop between MySQL Server 5.5.x and 5.6.x. (Bug #16864741, Bug #69316)

- **Performance; InnoDB:** When `innodb_thread_concurrency` is set to a non-zero value, there was a possibility that all `innodb_concurrency_tickets` would be released after each row was read, resulting in a concurrency check after each read. This could impact performance of all queries. One symptom could be higher system CPU usage. We strongly recommend that you upgrade to MySQL Server 5.6.13 if you use this setting. This could cause a performance drop between MySQL Server 5.5.x and 5.6.x. (Bug #68869, Bug #16622478)

- **Incompatible Change:** When used for an existing MySQL account, the `GRANT` statement could produce unexpected results if it included an `IDENTIFIED WITH` clause that named an authentication plug differing from the plugin named in the corresponding `mysql.user` table row.

  Because `IDENTIFIED WITH` is intended only for `GRANT` statements that create a new user, it is now prohibited if the named account already exists. (Bug #16083276)

- **Incompatible Change:** It is possible for a column `DEFAULT` value to be valid for the `sql_mode` value at table-creation time but invalid for the `sql_mode` value when rows are inserted or updated. Example:

```
SET sql_mode = '';
CREATE TABLE t (d DATE DEFAULT 0);
SET sql_mode = 'NO_ZERO_DATE,STRICT_ALL_TABLES';
INSERT INTO t (d) VALUES(DEFAULT);
```

  In this case, 0 should be accepted for the `CREATE TABLE` but rejected for the `INSERT`. However, previously the server did not evaluate `DEFAULT` values used for inserts or updates against the current `sql_mode`. In the example, the `INSERT` succeeds and inserts `'0000-00-00'` into the `DATE` column.

  The server now applies the proper `sql_mode` checks to generate a warning or error at insert or update time.

A resulting incompatibility for replication if you use statement-based logging (`binlog_format=STATEMENT`) is that if a slave is upgraded, a nonupgraded master will execute the preceding example without error, whereas the `INSERT` will fail on the slave and replication will stop.

To deal with this, stop all new statements on the master and wait until the slaves catch up. Then upgrade the slaves followed by the master. Alternatively, if you cannot stop new statements, temporarily change to row-based logging on the master (`binlog_format=ROW`) and wait until all slaves have processed all binary logs produced up to the point of this change. Then upgrade the slaves followed by the master and change the master back to statement-based logging. (Bug #68041, Bug #16078943)

- **Important Change; Replication:** When the server was running with `--binlog-ignore-db` and `SELECT DATABASE()` returned `NULL` (that is, there was no currently selected database), statements using fully qualified table names in *`dbname`*.*`tblname`* format were not written to the binary log. This was because the lack of a currently selected database in such cases was treated as a match for any possible ignore option rather than for no such option; this meant that these statements were always ignored.

  Now, if there is no current database, a statement using fully qualified table names is always written to the binary log. (Bug #11829838, Bug #60188)

- **InnoDB; Partitioning:** Joins involving partitioned `InnoDB` tables having one or more `BLOB` columns were not always handled correctly. The `BLOB` column or columns were not required to be join columns, or otherwise to be named or referenced in the statement containing the join, for this issue to occur. (Bug #16367691)

- **InnoDB; Partitioning:** Following any query on the `INFORMATION_SCHEMA.PARTITIONS` table, `InnoDB` index statistics as shown in the output of statements such as `SELECT * FROM INFORMATION_SCHEMA.STATISTICS` were read from the last partition, instead of from the partition containing the greatest number of rows. (Bug #11766851, Bug #60071)

  References: See also: Bug #16882435, Bug #69179.

- **InnoDB; Replication:** Trying to update a column, previously set to `NULL`, of an `InnoDB` table with no primary key caused replication to fail on the slave with `Can't find record in '`*`table`*`'`.

  

  Note

  This issue was inadvertently reintroduced in MySQL 5.6.6, and fixed again in MySQL 5.6.12.

  (Bug #11766865, Bug #60091)

  References: See also: Bug #16566658.

- **InnoDB:** In Windows 64-bit debug builds, read view `COPY_TRX_IDS` would report a "`vector subscript out of range`" error to standard error output. (Bug #17320056)

- **InnoDB:** When logging the delete-marking of a record during online `ALTER TABLE...ADD PRIMARY KEY`, `InnoDB` writes the transaction ID to the log as it was before the deletion or delete-marking of the record. When doing this, `InnoDB` would overwrite the `DB_TRX_ID` field in the original table, which could result in locking issues. (Bug #17316731)

- **InnoDB:** An assertion failure would occur while searching an index tree and traversing multiple levels where a block is accessed or pinned at each level. (Bug #17315967)

- **InnoDB:** The `row_sel_sec_rec_is_for_clust_rec` function would incorrectly prepare to compare a NULL column prefix in a secondary index with a non-NULL column in a clustered index. (Bug #17312846)

- **InnoDB:** An incorrect purge would occur when rolling back an update to a delete-marked record. (Bug #17302896)
- **InnoDB:** The assertion `ut_ad(oldest_lsn <= cur_lsn)` in file `buf0flu.cc` failed because the current max LSN would be retrieved from the buffer pool before the oldest LSN. (Bug #17252421)
- **InnoDB:** `InnoDB` `memcached` `add` and `set` operations would perform more slowly than SQL `INSERT` operations. (Bug #17214191)
- **InnoDB:** As commented in `log0log.h`, `old_lsn` and `old_buf_free` should only be compiled when `UNIV_LOG_DEBUG` is enabled. (Bug #17160270, Bug #69724)
- **InnoDB:** When started in ready-only mode, `InnoDB` would assert on a `SAVEPOINT`. (Bug #17086428)
- **InnoDB:** Before dropping an index, a check is performed to ensure the index root page is free. If the index root page is free, dropping activity is avoided. A transaction would be initialized before the check is performed. If the check evaluated to true, the initialized transaction would be left in a dangling state. (Bug #17076822)
- **InnoDB:** `InnoDB` would rename a user-defined foreign key constraint containing the string "_ibfk_" in its name, resulting in a duplicate constraint. (Bug #17076737, Bug #69693, Bug #17076718, Bug #69707)
- **InnoDB:** An `InnoDB` monitor test would raise an assertion in `ha_innodb.cc` due to a mutex conflict. (Bug #17027249)
- **InnoDB:** In debug builds, the `trx_sys->rw_max_trx_id` variable would sometimes be reversed resulting in an inconsistent `CLUST_INDEX_SIZE` value. (Bug #17026780)
- **InnoDB:** The `ha_innobase::clone` function would incorrectly assert that a thread cannot clone a table handler that is used by another thread, and that the original table handler and the cloned table handler must belong to the same transaction. The incorrect assertions have been removed. (Bug #17001980)
- **InnoDB:** A regression introduced in the fix for Bug #14606334 would cause crashes on startup during crash recovery. (Bug #16996584)
- **InnoDB:** Rolling back an `INSERT` after a failed `BLOB` write would result in an assertion failure. The assertion has been modified to allow NULL `BLOB` pointers if an error occurs during a `BLOB` write. (Bug #16971045)
- **InnoDB:** `SHOW ENGINE INNODB STATUS` output referenced a thread in hex format (example: `thread handle 0x880`), whereas the same thread was referenced in the `SHOW ENGINE INNODB STATUS` transaction list in decimal format (example: `thread id 2176`). (Bug #16934269, Bug #69437)
- **InnoDB:** A full-text search using the IN BOOLEAN MODE modifier would result in an assertion failure. (Bug #16927092)

  References: This issue is a regression of: Bug #16516193.
- **InnoDB:** When `CHECK TABLE` found a secondary index that contained the wrong number of entries, it would report an error but not mark the index as corrupt. `CHECK TABLE` now marks the index as corrupt when this error is encountered, but only the index is marked as corrupt, not the table. As a result, only the index becomes unusable until it is dropped and rebuilt. The table is unaffected. (Bug #16914007)
- **InnoDB:** `InnoDB` would attempt to gather statistics on partially created indexes. (Bug #16907783)
- **InnoDB:** To avoid namespace clashes, usage of `'using namespace std'` has been removed from `InnoDB`. (Bug #16899560)

- **InnoDB:** When dropping all indexes on a column with multiple indexes, `InnoDB` failed to block a `DROP INDEX` operation when a foreign key constraint requires an index. (Bug #16896810)
- **InnoDB:** Optimized explicit record locking routines. (Bug #16880127)
- **InnoDB:** The server would crash during a `memcached` set operation. The failure was due to a padded length value for a utf8 char column. During a `memcached` update operation, a field from an old tuple would be copied with a data length that was less than the padded utf8 char column value. This fix ensures that old tuples are not copied. Instead, a new tuple is created each time. (Bug #16875543)
- **InnoDB:** `innochecksum` would ignore the return value of `fwrite` which could result in an error or generate warnings and compilation errors when `WITH_INNODB_EXTRA_DEBUG` `CMake` is enabled. (Bug #16872677)
- **InnoDB:** An assertion failure would occur in file `row0log.cc` on `ROW_FORMAT=REDUNDANT` tables that contained an unexpected but valid data directory flag. (Bug #16863098)
- **InnoDB:** An assertion in `row0mysql.cc`, which ensures that the dictionary operation lock is not taken recursively, failed. (Bug #16862290)
- **InnoDB:** The two `INFORMATION_SCHEMA` tables for the InnoDB buffer pool could show an invalid page type for read-fixed blocks. This fix will show the unknown page type for blocks that are I/O-fixed for reading. (Bug #16859867)
- **InnoDB:** `InnoDB` record comparison functions have been simplified and optimized. (Bug #16852278)
- **InnoDB:** Removed invalid compilation warning messages that appeared when compiling the `InnoDB` memcached plugin. (Bug #16816824)
- **InnoDB:** During an insert buffer merge, InnoDB would invoke `lock_rec_restore_from_page_infimum()` on a potentially invalid record pointer. (Bug #16806366)
- **InnoDB:** The `innodb_rwlock_x_spin_waits` item in the `INFORMATION_SCHEMA.INNODB_METRICS` table would show the same value as the `innodb_rwlock_x_os_waits` item. (Bug #16798175)
- **InnoDB:** The `trx_tables_locked` counter in `INFORMATION_SCHEMA.INNODB_TRX` would not account for all tables with locks. (Bug #16793724)
- **InnoDB:** This patch removes the `UNIV_INTERN` function, which was introduced in MySQL 5.1 to help replace static linking in InnoDB with the shared object plugin. `UNIV_INTERN` is no longer required. (Bug #16781511)
- **InnoDB:** In debug builds, an online `ALTER TABLE` operation that performed a full table copy would raise an assertion. The assertion was due to a race condition that would occur during BLOB retrieval, when applying the table modification log to any log block except for the very last one. This fix modifies `row_log_table_apply_convert_mrec()` to ensure that an index B-tree lock is acquired to protect the access to `log->blobs` and the BLOB page. (Bug #16774118)
- **InnoDB:** In debug builds, an assertion could occur in `OPT_CHECK_ORDER_BY` when using binary directly in a search string, as binary may include `NULL` bytes and other non-meaningful characters. This fix will remove non-meaningful characters before the search is run. (Bug #16766016)
- **InnoDB:** The `page_zip_validate()` debug function, which is enabled when `UNIV_ZIP_DEBUG` is defined at compilation time, invokes `page_zip_decompress()`, which in turn would update some compression statistics. This would cause some mysql-test-run tests to fail. (Bug #16759605)

- **InnoDB:** Valgrind testing returned memory leak errors which resulted from a regression introduced by the fix for Bug #11753153. The `dict_create_add_foreign_to_dictionary` function would call `pars_info_create` but failed to call `pars_info_free`. (Bug #16754901)

- **InnoDB:** When the function `trx_rollback_or_clean_recovered()` rolls back or cleans up transactions during a crash recovery, it removes the trx objects from the `trx_sys` list without freeing up the memory used by those objects. To prevent a memory leak, this fix adds `trx_free_for_background()` calls to `trx_rollback_resurrected()`, the function that removes the trx objects. (Bug #16754776)

- **InnoDB:** A memory leak would occur in `dict_check_tablespaces_and_store_max_id()` when `space_id` is equal to zero. (Bug #16737332)

- **InnoDB:** The `page_zip_validate()` consistency check failed after compressing a page, in `page_zip_compress()`. This problem was caused by `page_zip_decompress()`, which failed to set `heap_no` correctly when a record contained no user data bytes. A record with no user data bytes occurs when, for example, a primary key is an empty string and all secondary index fields are NULL or an empty string. (Bug #16736929)

- **InnoDB:** A missing comma in `SHOW STATUS` output would break MySQL Enterprise Monitor parsing. (Bug #16723686)

- **InnoDB:** This patch is a code cleanup which may provide a minor performance improvement when keys are not used on columns and when using the default `latin1_swedish_ci` collation. (Bug #16723431)

- **InnoDB:** Some characters in the identifier for a foreign key constraint are modified during table exports. (Bug #16722314, Bug #69062)

- **InnoDB:** After a clean shutdown, `InnoDB` does not check `.ibd` file headers at startup. As a result, in a crash recovery scenario, `InnoDB` could load a corrupted tablespace file. This fix implements consistency and status checks to avoid loading corrupted files. (Bug #16720368)

- **InnoDB:** A regression introduced with the fix for Bug #11762038 would cause `InnoDB` to raise an incorrect error message. The message stated that, “InnoDB cannot delete/update rows with cascading foreign key constraints that exceed max depth of 20”. The error message would occur when killing connections reading from `InnoDB` tables that did not have foreign key constraints. (Bug #16710923)

  References: This issue is a regression of: Bug #11762038.

- **InnoDB:** Stale `InnoDB` memcached connections would result in a memory leak. (Bug #16707516, Bug #68530)

- **InnoDB:** In debug builds, an assertion failure would occur if `innodb_log_group_home_dir` does not exist. Instead of an assertion, `InnoDB` now aborts with an error message if `innodb_log_group_home_dir` does not exist. (Bug #16691130, Bug #69000)

- **InnoDB:** An existing full-text index would become invalid after running `ALTER TABLE ADD FULLTEXT` due to an unsynchronized full-text cache. (Bug #16662990, Bug #17373659)

- **InnoDB:** An `INSERT` into a temporary table resulted in the following assert: `ASSERT ID > 0 IN TRX_WRITE_TRX_ID()`. This fix corrects conditions for moving a transaction from a read-only list to a read-write list when the server is running in read-only mode. (Bug #16660575)

- **InnoDB:** Shutting down and restarting InnoDB with `--innodb-force-recovery` set to 3 or greater (4, 5, or 6) and attempting to drop a table would result in a crash. With `innodb_force_recovery` mode set to 3 or greater DML operations should be blocked and DDL operations allowed. This fix ensures that DDL operations are allowed. (Bug #16631778)

- **InnoDB:** A race condition would occur between `ALTER TABLE ... ADD KEY` and `INSERT` statements, resulting in an “Unable to Purge a Record” error. (Bug #16628233)

- **InnoDB:** Very large `InnoDB` full-text search (FTS) results could consume an excessive amount of memory. This bug fix reduces memory consumption for FTS results and introduces a new configuration parameter, `innodb_ft_result_cache_limit`, which places a default size limit of 2000000000 bytes on the `InnoDB` FTS query result cache. `innodb_ft_result_cache_limit` has an unlimited maximum value and can be set dynamically. (Bug #16625973)

- **InnoDB:** This fix addresses a race condition that would occur between the rollback of a recovered transaction and creation of a secondary index in a locked operation. The race condition would corrupt the secondary index. (Bug #16593427)

- **InnoDB:** DML operations on compressed temporary tables would result in a Valgrind error in the buffer manager stack. (Bug #16593331)

- **InnoDB:** When `ADD PRIMARY KEY` columns are reordered in an `ALTER TABLE` statement (for example: `ALTER TABLE t1 ADD PRIMARY KEY(a,b), CHANGE a a INT AFTER b`), the log apply for `UPDATE` operations failed to find rows. (Bug #16586355)

- **InnoDB:** A code regression resulted in a record lock wait in a dictionary operation. A code modification made to avoid starting a transaction on a temporary table failed to reset the state back to `init` upon completion of the operation. If a transaction is started, the state is usually reset by `trx_commit`. To catch similar problems in the future, this fix adds asserts to `innobase_commit()`, `innobase_rollback()`, and `ha_innobase::update_thd()` that trigger when `trx->dict_operation` and `trx->dict_operation_lock_mode` are not set. (Bug #16575799)

- **InnoDB:** In debug builds, the `assert_trx_in_list()` assert failed, causing a race condition. This fix removes the assert. The same assert is verified in the caller and existing checks are sufficient. (Bug #16567258)

- **InnoDB:** The MySQL printf facility (`my_vsnprintf`) did not understand the Microsoft I32 and I64 integer format width specifiers, such as `%I64u` for printing a 64-bit unsigned integer. As a result, `DBUG_PRINT` could not be used with the InnoDB UINT64PF format, which is defined as `%I64u` on Windows. This fix replaces the non-standard “I64” and “I32” length modifiers on Windows with “ll” and “l” so that they will be understood by both `my_snprintf()` and `ut_snprintf()`. (Bug #16559119)

- **InnoDB:** `ALTER TABLE` operations on `InnoDB` tables that added a `PRIMARY KEY` using a column prefix could produce an incorrect result. (Bug #16544336)

- **InnoDB:** For `ALTER TABLE` operations on `InnoDB` tables that required a table-copying operation, other transactions on the table might fail during the copy. However, if such a transaction issued a partial rollback, the rollback was treated as a full rollback. (Bug #16544143)

- **InnoDB:** The `row0purge.h` include file contained a self-referencing inclusion. (Bug #16521741)

- **InnoDB:** During a transaction commit, `prepare_commit_mutex` is acquired to preserve the commit order. If the commit operation failed, the transaction would be rolled back but the mutex would not be released. Subsequent insert operations would not be able to acquire the same mutex. This fix frees `prepare_commit_mutex` during `innobase_rollback`. (Bug #16513588)

- **InnoDB:** The `recv_writer` thread would only start after all redo log scans finished. In the case of multiple redo log scans, accumulated redo records would be applied after each scan and before processing the next scan. The absence of the `recv_writer` thread to help with flushing would slow recovery or result in a server startup timeout. This fix ensures that the `recv_writer` thread starts before the first scan batch is processed. (Bug #16501172)

- **InnoDB:** Under certain circumstances, LRU flushing would take a long time possibly affecting all flushing activity and causing a shutdown timeout. (Bug #16500209)
- **InnoDB:** The `InnoDB` memcached `test.demo_test` table failed to work when defined as a `utf8` charset table. (Bug #16499038)
- **InnoDB:** In cases where threads are forced to do single page flushing, `fsync()` would be triggered for all data files. This fix allows for synchronous single page flushing. (Bug #16477781)
- **InnoDB:** This fix removes most calls to `OS_THREAD_SLEEP` from InnoDB. (Bug #16472953, Bug #68588)
- **InnoDB:** `FLUSH TABLES FOR EXPORT` would sleep too often while flushing pages from buffer pools. (Bug #16471701)
- **InnoDB:** Concurrently inserting into a full-text table would cause some inserts to fail. Duplicate values would be generated for full-text search document IDs when performing inserts into a hidden full-text search document ID column. (Bug #16469399)
- **InnoDB:** An `InnoDB` memcached file descriptor leak would cause a serious error. (Bug #16466664)
- **InnoDB:** The `page_zip_available` function would count some fields twice. (Bug #16463505)
- **InnoDB:** This fix replaces the `IB_ULONGLONG_MAX` constant with `LSN_MAX` where the code refers to log sequence numbers, or with `TRX_ID_MAX` where `trx->no` is initialized to an undefined value. This change does not alter the value of the constant. (Bug #16458660)
- **InnoDB:** This fix corrects the text for InnoDB error 6025, which stated, “InnoDB: read can't be opened in ./ib_logfile0 mode.”. The corrected message states, “InnoDB: ./ib_logfile0 can't be opened in read mode.” The variable and mode in the message construct were transposed. (Bug #16434398)
- **InnoDB:** When changing the shared tablespace file name using `innodb_data_file_path` and leaving the current log files in place, `InnoDB` would create a new tablespace file and overwrite the log files resulting in a mismatch between the data dictionary and tables on disk. This bug fix ensures that `InnoDB` does not create a new tablespace if there are inconsistent system tablespaces, undo tablespaces, or redo log files. (Bug #16418661)
- **InnoDB:** Creating a foreign key constraint using the `ALTER TABLE INPLACE` algorithm requires `foreign_key_checks` to be set to 0 (`SET foreign_key_checks = 0;`). As a result, an appropriate duplicate ID check would not be performed. (Bug #16413976)
- **InnoDB:** When the InnoDB shutdown mode (`innodb_fast_shutdown`) is set to 2 and the master thread enters the flush loop, the thread would not be able to exit under some circumstances. This could lead to a shutdown hang. (Bug #16411457)
- **InnoDB:** In debug builds, an insert failed with an invalid assertion: `sync_thread_levels_g(array, level - 1, TRUE)`. (Bug #16409715)
- **InnoDB:** Crash recovery failed with a `!recv_no_log_write` assertion when reading a page. (Bug #16405422)
- **InnoDB:** An `ALTER TABLE` operation that performed a table copy failed because a temporary tablespace with the same name already existed. This fix makes temporary tables and tablespace names more unique by adding the current log sequence number (LSN) to the end of the previous table or file name. For example, table name “`test/#sql-ib21`” becomes “`test/#sql-ib21-1701208`,” where `1701208` is the current LSN. Both the LSN and the table ID are needed to ensure that the name is unique because it is theoretically possible for multiple threads to have the same LSN. Including the table ID allows the temporary name to be associated with the table. (Bug #16403420)

- **InnoDB:** Multiple concurrent calls to `dict_update_statistics()` would result in unnecessary server load. (Bug #16400412)
- **InnoDB:** On 64-bit Windows builds, `INNODB_BUFFER_POOL_SIZE` would not accept an allocation of more than 32GB. This limitation was due to a bug that truncated the internal value for the InnoDB buffer pool size to 32 bits on 64-bit Windows builds. (Bug #16391722, Bug #68470)
- **InnoDB:** Restarting InnoDB in read-only mode and running a workload would occasionally return a `global_segment < os_aio_n_segments` assertion. (Bug #16362046)
- **InnoDB:** `DROP DATABASE` failed if the database contained an InnoDB table that had a data file in an external data directory. The external data file had an “InnoDB Symbolic Link” file type (`.isl`) that was not recognized by MySQL. This fix adds `.isl` as a known InnoDB file type. (Bug #16338667)
- **InnoDB:** `RENAME TABLE` would result in a hang due to a MySQL mutex acquisition deadlock. (Bug #16305265)
- **InnoDB:** This fix removes dated and incomplete code that is protected by the `UNIV_LOG_ARCHIVE` macro. (Bug #16296837)
- **InnoDB:** Under testing, a `FLUSH TABLE` operation resulted in a timeout due to a missing acknowledgment that the purge thread had stopped. (Bug #16277387)
- **InnoDB:** For compressed tables, a page reorganize operation would always write an `MLOG_ZIP_PAGE_REORGANIZE` record to the redo log, which is only correct if `innodb_log_compressed_pages=OFF`. When `innodb_log_compressed_pages=ON`, the page reorganize operation should log the compressed page image. (Bug #16267120)
- **InnoDB:** When tables are linked by foreign key constraints, loading one table would open other linked tables recursively. When numerous tables are linked by foreign key constraints, this would sometimes lead to a thread stack overflow causing the server to exit. Tables linked by foreign key constraints are now loaded iteratively. Cascade operations, which were also performed in a recursive manner, are now performed iteratively using an explicit stack. (Bug #16244691, Bug #65384)
- **InnoDB:** After disabling foreign key checks with `SET foreign_key_checks=0` and performing a `DROP INDEX`, the table was no longer accessible after restarting the server. This fix allows the table with missing foreign key indexes to be accessed when `SET foreign_key_checks=0`. When the table is accessible, the user must recreate the missing indexes to fulfill the foreign key constraints. (Bug #16208542, Bug #68148)
- **InnoDB:** When a transaction is in `READ COMMITTED` isolation level, gap locks are still taken in the secondary index when a row is inserted. This occurs when the secondary index is scanned for duplicates. The function `row_ins_scan_sec_index_for_duplicate()` always calls the function `row_ins_set_shared_rec_lock()` with `LOCK_ORDINARY` irrespective of the transaction isolation level. This fix modifies the `row_ins_scan_sec_index_for_duplicate()` function to call `row_ins_set_shared_rec_lock()` with `LOCK_ORDINARY` or `LOCK_REC_NOT_GAP`, based on the transaction isolation level. (Bug #16133801, Bug #68021)
- **InnoDB:** Persistent stats would be disabled unnecessarily when running in read-only mode. When running in read-only mode, fetching stats from disk does not involve any modification of on-disk data except for when `ANALYZE TABLE` is run. This fix enables persistent stats for read-only mode. (Bug #16083211)
- **InnoDB:** Starting `mysqld` with `--innodb_log_buffer_size=50GB` failed to allocate memory and returned NULL. For non-debug builds there was no check in place and a segmentation fault occurred. This fix adds a log message stating that memory failed to be allocated, and adds an assertion. (Bug #16069598, Bug #68025)

- **InnoDB:** When `UNIV_DEBUG` is enabled in debug builds, `buf_validate()` is often called which sometimes results in false alarms in tests on semaphore wait timeout. This fix increases counter values to reduce false alarms. (Bug #16068056)
- **InnoDB:** While printing a UTF-8 table name, `InnoDB` would truncate the table name, resulting in an incomplete buffer and subsequent Valgrind error. This bug fix also addresses an incorrect debugging error message. (Bug #16066351)
- **InnoDB:** The `explain_filename` function, which provides information about a partition by parsing the file name, would return an error when attempting to parse a file name with no partition information. (Bug #16051728)
- **InnoDB:** Stopping the server, removing a database table (d1.t1) `.frm` file from the data directory, restarting the server, and dropping the database (d1), would cause an assertion. (Bug #16043216)
- **InnoDB:** While processing read-write workloads, InnoDB would scan more pages than are required for flushing, unnecessarily consuming CPU resource. (Bug #16037180)
- **InnoDB:** `TRUNCATE TABLE` failed to handle the return value from `btr_create` when `btr_create` is invoked by `TRUNCATE TABLE` for creation of a new index. (Bug #16026889)
- **InnoDB:** An overflow would occur for `innodb_row_lock_time_max` and `innodb_row_lock_current_waits`. This fix modifies code logic in `storage/innobase/srv/srv0srv.c`. (Bug #16005310)
- **InnoDB:** Attempting to create a table while in `innodb_read_only` mode would result in the following error: `ERROR 1015 (HY000): Can't lock file (errno: 165 - Table is read only)`. (Bug #15963619)
- **InnoDB:** An active `FLUSH TABLE FOR EXPORT` thread would cause a hang during shutdown. The fix ensures that `trx_is_interrupted()` is checked during `ibuf_merge`. (Bug #15953255)
- **InnoDB:** `innochecksum` would return an error when run on compressed tables. (Bug #14612872, Bug #66779)
- **InnoDB:** A multi-row `INSERT ... ON DUPLICATE KEY UPDATE` insert failure, caused by a duplicate key error, would result in duplicate auto-increment values. (Bug #14483484, Bug #66301)
- **InnoDB:** A mismatch between `.ibd` files and the `InnoDB` data dictionary could occur if `TRUNCATE TABLE` is interrupted by a crash. The mismatch would be encountered after recovery. To avoid this problem, truncate table information is written to a truncate log file that resides temporarily in the log directory. The truncate log file has the following naming convention: `ib_`*`space_id`*`_trunc.log`. If the `TRUNCATE` operation is successful, the truncate log file is removed. If the `TRUNCATE` operation is interrupted by a crash, information is read from the truncate log file during recovery, the log records are applied, and the truncate log file is removed. (Bug #14174004, Bug #13997329, Bug #17227149, Bug #17238361)
- **InnoDB:** The documentation incorrectly stated that `START TRANSACTION WITH CONSISTENT SNAPSHOT` provides a consistent snapshot only if the current isolation level is `REPEATABLE READ` or `SERIALIZABLE`. `START TRANSACTION WITH CONSISTENT SNAPSHOT` only works with `REPEATABLE READ`. All other isolation levels are ignored. The documentation has been revised and a warning is now generated whenever the `WITH CONSISTENT SNAPSHOT` clause is ignored. (Bug #14017206, Bug #65146)
- **InnoDB:** The `srv_master_thread` background thread, which monitors server activity and performs activities such as page flushing when the server is inactive or in a shutdown state, runs on a one second delay loop. `srv_master_thread` failed to check if the server is in a shutdown state before sleeping. (Bug #13417564, Bug #63276)

- **InnoDB:** In the error log, a full-text search index would be reported missing from the data dictionary during a `TRUNCATE TABLE` operation. After restarting `mysqld`, the following `InnoDB` error would be reported: "`InnoDB: Error: trying to load index idx13 for table test/g1 but the index tree has been freed.`." (Bug #12429565)

  References: See also: Bug #17402002.

- **InnoDB:** When the value provided for `innodb_buffer_pool_size` on 32-bit systems is too large, an error message would incorrectly reference the internal variable, `innobase_buffer_pool_size`, instead of `innodb_buffer_pool_size`. (Bug #11759578, Bug #51901)

- **InnoDB:** Compiling `InnoDB` on Windows Vista 64-bit with VS2005 would result in compilation errors. (Bug #11752731, Bug #44004)

- **InnoDB:** The `row_check_index_for_mysql` method, which checks for NULL fields during an index scan or `CHECK TABLE` operation, would iterate unnecessarily. Thanks to Po-Chun Chang for the patch to correct this issue. (Bug #69377, Bug #16896647)

- **InnoDB:** When running an `InnoDB` full-text search in boolean mode, prefixing an asterisk (`*`) to a search string (`'*string'`) would result in an error whereas for `MyISAM`, a prefixed asterisk would be ignored. To ensure compatibility between `InnoDB` and `MyISAM`, `InnoDB` now handles a prefixed asterisk in the same way as `MyISAM`. (Bug #68948, Bug #16660607)

- **InnoDB:** Successive deletes in descending key order would lead to under-filled `InnoDB` index pages. When an `InnoDB` index page is under-filled, it is merged with the left or right sibling node. The check performed to determine if a sibling node is available for merging was not functioning correctly. (Bug #68501, Bug #16417635)

- **InnoDB:** Setting `foreign_key_checks=0` and running `ALTER TABLE` to change the character set of foreign key columns for a database with multiple tables with foreign key constraints would leave the database in an inconsistent state. Subsequent `ALTER TABLE` operations (using the `COPY` algorithm) with `foreign_key_checks=1` would fail due to the detected inconsistency. Reversion of the partially executed `ALTER TABLE` operation would also fail, resulting in the loss of the table being altered. When running the same `ALTER TABLE` operation with a `RENAME` clause, the inconsistency would not be detected but if the `ALTER TABLE` operation failed for some other reason, reversion of the partially executed `ALTER TABLE` failed with the same result.

  The bug fix temporarily disables `foreign_key_checks` while the previous table definition is restored. (Bug #65701, Bug #14227431)

- **InnoDB:** Creating a table with a comment or default textual value containing an apostrophe that is escaped with a backslash would sometimes cause the InnoDB storage engine to omit foreign key definitions. (Bug #61656, Bug #12762377)

- **InnoDB:** The `pthread_mutex`, `commit_threads_m`, which was initialized but never used, has been removed from the code base. (Bug #60225, Bug #11829813)

- **InnoDB:** In many cases InnoDB calls `exit(1)` when it encounters a fatal error. An `exit(1)` call does not produce a crash dump or provide information about the process state. Additionally, on Windows, an `exit(1)` call does not report a crashed process in the Event Viewer. This fix replaces `exit(1)` calls with `ut_error` calls in a number of places. (Bug #56400, Bug #11763660)

- **Partitioning:** Creating a table `t1` using `CREATE TABLE ... PARTITION BY LIST ... PARTITION ... VALUES IN (NULL)`, then attempting to execute `CREATE TABLE ... LIKE t1` caused the server to fail. (Bug #16860588)

- **Partitioning:** When upgrading to MySQL 5.5.31 or higher, a message is written into the output of `mysql_upgrade` when encountering a partitioned table for which the `ALGORITHM` option is required

to maintain binary compatibility with the original; the message includes the `ALTER TABLE` statement required to make the change. For such a table having a sufficiently large number of partitions, the message was truncated with an error before the complete `ALTER TABLE` statement could be written. (Bug #16589511)

- **Partitioning:** When a range specified in the `WHERE` condition of a query against a table partitioned by `RANGE` entirely within that of one of the partitions, the next partition was also checked for rows although it should have been pruned away.

  Suppose we have a range-partitioned table `t` created using the following SQL statement:

  ```
  CREATE TABLE t  (
      id INT AUTO_INCREMENT,
      dt DATETIME,
      PRIMARY KEY (dt,id),
      UNIQUE KEY (id,dt)
  )
      PARTITION BY RANGE  COLUMNS(dt) (
          PARTITION p0 VALUES LESS THAN ('2013-01-01'),
          PARTITION p1 VALUES LESS THAN ('2013-01-15'),
          PARTITION p2 VALUES LESS THAN ('2013-02-01'),
          PARTITION p3 VALUES LESS THAN ('2013-02-15'),
          PARTITION pmax VALUES LESS THAN (MAXVALUE)
  );
  ```

  An example of a query that exhibited this issue when run against `t` is shown here:

  ```
  SELECT COUNT(*) FROM t
      WHERE dt >= '2013-02-01' AND dt < '2013-02-15';
  ```

  In this case, partition `pmax` was checked, even though the range given in the `WHERE` clause lay entirely within partition `p3`. (Bug #16447483)

- **Partitioning:** When dropping a partitioned table, the table's `.par` file was deleted first, before the table definition or data. This meant that, if the server failed during the drop operation, the table could be left in an inconsistent state in which it could neither be accessed nor dropped.

  The fix for this problem makes the following changes:

  - Now, when dropping a partitioned table, the table's `.par` file is not removed until all table data has been deleted.
  - When executing `DROP TABLE` of a partitioned table, in the event that its `.par` file is determined to be missing, the table's `.frm` file is now immediately deleted, in effect forcing the drop to complete.

  (Bug #13548704, Bug #63884)

- **Replication:** The server attempted to perform an internal truncatation of the `slave_worker_info` table while resetting it, even though this is a DDL operation and should not be used conccurrently with DML operations. To prevent this from happening, the reset now performs sequential row deletion in place of the truncation operation. (Bug #17286858, Bug #69898)

- **Replication:** The data size for a table map event created during execution was calculated, but not when the event was created from a network packet. This could later cause problems when the data fields of such events were treated as if they had a length equal to 0 when trying to write the events to a cache, or to the binary log.

To avoid future problems of this nature, the table map's data size is now calculated in both cases. (Bug #17164074)

- **Replication:** When the `--relay-log-info-file` option was used together with `--slave-parallel-workers` set to a value greater than 1, `mysqld` failed to start. (Bug #17160671)
- **Replication:** The commit error caused by the failure of binary log rotation failure generated an incident event in the binary log file and interrupted the user session with error messages which did not mention that the slave server would be stopped later when the incident event was replayed.

  Now, when encountering binary log rotation failure, a more helpful error message is instead written to the log, alerting the user to investigate in a timely manner. (Bug #17016017)
- **Replication:** The condition leading to the issue fixed in Bug #16579083 continued to raise an error even though the condition itself no longer cause the issue to occur. (Bug #16931177, Bug #69369)

  References: See also: Bug #16271657, Bug #16491597, Bug #68251, Bug #68569. This issue is a regression of: Bug #16579083.
- **Replication:** The `mysqlbinlog` option `--rewrite-db` caused `USE` statements to be ignored, even for databases that were not referenced by the option. (Bug #16914535)
- **Replication:** When `rpl_semi_sync_master_timeout` was set to an extremely large value, semisynchronous replication became very slow, especially when many sessions were working in parallel. It was discovered that the code to calculate this timeout was inside the wait loop itself, with the result that an increase in the value of `rpl_semi_sync_master_timeout` caused repeated iterations. This fix improves the method used to calculate wakeup times, and moves it outside of the wait loop, so that it is executed one time only. (Bug #16878043, Bug #69341)
- **Replication:** It was possible to cause a deadlock after issuing `FLUSH TABLES WITH READ LOCK` by issuing `STOP SLAVE` in a new connection to the slave, then issuing `SHOW SLAVE STATUS` using the original connection.

  The fix for this includes the addition of the `rpl_stop_slave_timeout` system variable, to control the time in seconds to wait for slave to stop after issuing `STOP SLAVE` before returning a warning. (Bug #16856735)
- **Replication:** It was possible in `CHANGE MASTER TO` statements to set the `MASTER_DELAY` option greater than the supported maximum value ($2^{31} - 1$). In addition, the error resulting from setting `MASTER_DELAY` to a value greater than 2^{32} was not handled correctly. (Bug #16820156, Bug #16960315, Bug #69249, Bug #69469)
- **Replication:** Some expressions employing variables were not handled correctly by `LOAD DATA`. (Bug #16753869)
- **Replication:** In some circumstances, the message in the `Last_Error` column from the output of `SHOW SLAVE STATUS` referred to `GTID_NEXT_LIST` although this variable is not currently implemented (the name is reserved for possible future use). Now in such cases the error message no longer refers to this variable. (Bug #16742886, Bug #69096)

  References: See also: Bug #16715809, Bug #69045.
- **Replication:** `mysqlbinlog --rewrite-db` failed when the name of the destination database contained any underscore (_) characters. (Bug #16737279)
- **Replication:** Issuing a `FLUSH TABLES` statement on a GTID-enabled master caused replication to fail. It was found that this misbehavior was introduced by the fix for Bug #16062608, which disallowed

statements that perform an implicit commit but whose changes are not logged when `gtid_next` is set to any value other than `AUTOMATIC`. The changes made in that fix have been reverted, and such statements are (again) allowed without regard to the value of this variable. (Bug #16715809, Bug #69045)

References: Reverted patches: Bug #16062608.

- **Replication:** Point-in-time recovery could fail when trying to restore a single database from a binary log in row-based format using `mysqlbinlog` with the `--database` option. (Bug #16698172)

- **Replication:** A crash-on-commit error caused `InnoDB` to lose the previous transaction following execution of a `RESET MASTER` statement. This occurred because the prepare phase caused a flush to disk, while the commit phase did not perform a corresponding flush within `InnoDB`.

  To fix this problem, `RESET MASTER` now causes storage engine logs to be flushed on commit. (Bug #16666456, Bug #68932)

- **Replication:** On Windows platforms, issuing `SHOW SLAVE STATUS` while the slave I/O thread was being terminated due to an error caused the slave to fail. (Bug #16662771)

- **Replication:** When processing an `Update_rows_log_event` or `Delete_rows_log_event` from the binary log, the before image is hashed and stored in a hash table. Following this, the original table is scanned for the desired records; subsequent processing hashes each record fetched from the original table and performs a lookup for it in the hash table. However, columns read from the image that had originally been set to `NULL` could instead contain random or “garbage” data, causing the lookup (and thus replication) to fail with an error such as `Could not execute Update_rows event on table...`. (Bug #16621923)

  References: See also: Bug #11766865. This issue is a regression of: Bug #16566658.

- **Replication:** When used with the options `--dump-slave --include-master-host-port`, `mysqldump` printed the port number within quotation marks, as if it were a string value rather than an integer. (Bug #16615117)

- **Replication:** Linker errors occurred if the header file `log_event.h` was included in an application containing multiple source files, because the file `rpl_tblmap.cc` was included in `log_event.h`. This fix moves the inclusion of `rpl_tblmap.cc` into the source files that use `log_event.h`. (Bug #16607258)

- **Replication:** The error displayed by `SHOW SLAVE STATUS` when a worker thread fails to apply an event contained no event coordinate information. The GTID for the event's group was also not shown. Now in such cases, the text shown for `Last_SQL_Error` is prefixed with the (physical) master binary log coordinates, as well as the value of `gtid_next` when this has been set. (Bug #16594095)

- **Replication:** A session attachment error during group commit causes the rollback of the transaction (as intended), but the transaction in which this happened was still written to the binary log and replicated to the slave. Thus, such an error could lead to a mismatched master and slave.

  Now when this error occurs, an incident event is written in the binary log which causes replication to stop, and notifies the user that redundant events may exist in the binary log. An additional error is also now reported to the client, indicating that the ongoing transaction has been rolled back. (Bug #16579083)

- **Replication:** Due to time resolution issues on some systems, the time to be taken by the dump thread for a reply from the slave could be calculated to be less than zero, leading to `Semi-sync master wait for reply fail to get wait time` errors. Since this condition does not have a

negative impact on replication, errors caused by these conditions have been reduced to warnings. (Bug #16579028)

- **Replication:** Running the server with `--log-slave-updates` together with `--replicate-wild-ignore-table` or `--replicate-ignore-table` in some cases caused updates to user variables not to be logged. (Bug #16541422)
- **Replication:** When using `mysqlbinlog` and the `mysql` client to roll forward two or more binary logs on a server having GTIDs enabled, the `gtid_next` variable was not properly reset when switching from the first to the second binary log, causing processing to halt with an error at that point. (Bug #16532543)
- **Replication:** The `mysqlbinlog` options `--include-gtids`, `--exclude-gtids`, and `--skip-gtids` did not work correctly when trying to process multiple files. (Bug #16517775)
- **Replication:** When one or more GTID log events but no previous GTIDs log events were found in the binary log, the resulting error was mishandled and led to a failure of the server. (This is an extremely rare condition that should never occur under normal circumstances, and likely indicates that the binary log file has somehow been corrupted.) Now in such cases, an appropriate error is issued, and is handled correctly. (Bug #16502579, Bug #68638)
- **Replication:** Attempting to execute `START SLAVE` after importing new `slave_master_info` and `slave_relay_log_info` tables failed with an empty error message. Now an appropriate error and message are issued in such cases. (Bug #16475866, Bug #68605)
- **Replication:** Restarting the server after the `slave_relay_log_info` table had been emptied caused `mysqld` to fail while trying to return an error. (Bug #16460978, Bug #68604)
- **Replication:** The warning issued when specifying `MASTER_USER` or `MASTER_PASSWORD` with `CHANGE MASTER TO` was unclear for a number of reasons, and has been changed to read, `Storing MySQL user name or password information in the master info repository is not secure and is therefore not recommended. Please consider using the USER and PASSWORD connection options for START SLAVE; see 'START SLAVE Syntax' in the MySQL Manual for more information.` (Bug #16460123, Bug #16461303, Bug #68602, Bug #68599)
- **Replication:** Extra binary log rotations were performed due to concurrent attempts at rotation when the binary log became full, which were allowed to succeed. This could lead to the unnecessary creation of many small binary log files. (Bug #16443676, Bug #68575)
- **Replication:** When the size of an execution event exceeded the maximum set for the buffer (`slave_pending_jobs_size_max`), row-based replication could hang with `Waiting for slave workers to free pending events`. (Bug #16439245, Bug #68462)
- **Replication:** Following disconnection from the master, the slave could under certain conditions report erroneously on reconnection that it had received a packet that was larger than `slave_max_allowed_packet`, causing replication to fail. (Bug #16438800, Bug #68490)
- **Replication:** A slave using row-based replication was unable to read the rows containing columns of type `MYSQL_TYPE_DECIMAL` properly (old-style decimal, used prior to MySQL 5.0.3). Now the slave throws an error if it receives this type of data. You can convert the old-style `DECIMAL` format to the binary format used in current MySQL releases with `ALTER TABLE`; see Upgrading from MySQL 4.1 to 5.0, for more information. (Bug #16416302)
- **Replication:** An SQL thread error during MTS slave recovery caused the slave to fail. (Bug #16407467, Bug #68506)
- **Replication:** When using the options `--read-from-remote-server --stop-never --base64-output=decode-rows --verbose`, `mysqlbinlog` failed to reset the counter used to store the current position within the file when the binary log on the server was rotated. (Bug #16316123, Bug #68347)

- **Replication:** When using `mysqldump` to back up a database created with MySQL 5.6.4 or an earlier version, setting `--set-gtid-purged=AUTO` caused the backup to fail, because pre-5.6.5 versions of MySQL did not support GTIDs, and it could not be determined if GTIDs were enabled for the database. This fix makes sure `mysqldump` does not attempt to output a `SET @@global.gtid_purged` statement when backing up any pre-5.6.5 databases. (Bug #16303363, Bug #68314)

- **Replication:** `DROP TEMP TABLE IF EXISTS` statements could lead to failures in applying the binary log during point-in-time recovery operations. This is due to the fact that, when using row-based replication, the server appends `IF EXISTS` to any `DROP TEMPORARY TABLE` statements written to the binary log, and that the slave SQL thread does not check * wildcard filter rules for `DROP TEMPORARY TABLE IF EXISTS`. If `--log-slave-updates` was also enabled on the slave, such a statement was preceded by a `USE` statement. If the database referred by the `USE` statement did not exist, the statement failed, and stopped replication.

  Now, when writing `DROP TEMPORARY TABLE IF EXISTS` into the binary log, no `USE` statement is written, and the table name in the `DROP TEMPORARY TABLE` statement is a fully qualified table name. (Bug #16290902)

- **Replication:** Deadlocks could sometimes occur on group commits with a high number of concurrent updates, as well as when one client held a lock from a commit while another client imposed a lock while rotating the binary log. (Bug #16271657, Bug #16491597, Bug #68251, Bug #68569)

- **Replication:** After a transaction was skipped due to its GTID already having been logged, all remaining executed transactions were incorrectly skipped until `gtid_next` was pointed to a different GTID.

  To avoid this incorrect behavior, all transactions—even those that have been skipped—are marked as undefined when they are commited or rolled back, so that an error is thrown whenever a second transaction is executed following the same `SET @@session.gtid_next` statement. (Bug #16223835)

- **Replication:** When semisynchronous replication was enabled, the automatic dropping on the master of an event created using `ON COMPLETION NOT PRESERVE` caused the master to fail. (Bug #15948818, Bug #67276)

- **Replication:** Modifying large amounts of data within a transaction can cause the creation of temporary files. Such files are created when the size of the data modified exceeds the size of the binary log cache (`max_binlog_cache_size`). Previously, such files persisted until the client connection was closed, which could allow them to grow until they exhausted all available disk space in `tmpdir`. To prevent this from occurring, the size of a temporary file created in this way in a given transaction is now reset to 0 when the transaction is committed or rolled back. (Bug #15909788, Bug #18021493, Bug #66237)

- **Replication:** When the master had more than one table with an auto-increment column, *and* the slave ignored at least one of these tables due to `--replicate-ignore-table` rules, *but* at least one them was replicated, even so—the replicated table or tables having at least one trigger updating one or more tables existing only on the slave—updates to any of the auto-increment tables on the master caused replication to fail. (Bug #15850951, Bug #67504)

- **Replication:** Setting a `SET` column to `NULL` inside a stored procedure caused replication to fail. (Bug #14593883, Bug #66637)

- **Replication:** The binary log contents got corrupted sometimes, because the function `MYSQL_BIN_LOG::write_cache` always thought it had reached the end-of-cache when the function `my_b_fill()` reported a '0,' while that could also mean an error had occurred. This fix makes sure that whenever `my_b_fill()` returns a '0,' an error check is performed on `info->error`. (Bug #14324766, Bug #60173)

- **Replication:** The internal function `MYSQL_BIN_LOG::open_binlog()` contained an unneeded variable, which has been removed. (Bug #14134590, Bug #60188)

- **Replication:** `PURGE BINARY LOGS` by design does not remove binary log files that are in use or active, but did not provide any notice when this occurred. Now, when log files are not removed under such conditions, a warning is issued; this warning includes information about the file or files were not removed when the statement was issued. (Bug #13727933, Bug #63138)
- **Replication:** It was possible for the multi-threaded slave coordinator to leak memory when the slave was stopped while waiting for the next successful job to be added to the worker queue. (Bug #13635612)
- **Replication:** When replicating to a `BLACKHOLE` table using the binary logging format, updates and deletes cannot be applied and so are skipped. Now a warning is generated for this whenever it occurs.

  

  Note

  `binlog_format=STATEMENT` is recommended when replicating to tables that use the `BLACKHOLE` storage engine.

  (Bug #13004581)
- **Replication:** Temporary files created by `LOAD DATA INFILE` were not removed if the statement failed. (Bug #11763934, Bug #56708)
- **Replication:** After the client thread on a slave performed a `FLUSH TABLES WITH READ LOCK` and was followed by some updates on the master, the slave hung when executing `SHOW SLAVE STATUS`. (Bug #68460, Bug #16387720)
- **Microsoft Windows:** On Microsoft Windows, passing in `--local-service` to `mysqld.exe` when also passing in a service name could cause a crash at startup. (Bug #16999777, Bug #69549)
- The contents of SQL condition items such as `TABLE_NAME`, `CONSTRAINT_NAME`, an so forth were lost if resignaled by a stored routine condition handler. (Bug #17280703)
- `AES_ENCRYPT()` and `AES_DECRYPT()` failed to work correctly when MySQL was built with an `AES_KEY_LENGTH` value of 192 or 256. (Bug #17170207)
- `SELECT * from performance_schema.events_statements_current` could raise an assertion due to a race condition under load. (Bug #17164720)
- `InnoDB` full-text searches failed in databases whose names began with a digit. (Bug #17161372)
- A successful connection failed to reset the per-IP address counter used to count successive connection failures. This could possibly cause a host to be blocked, when the `max_connect_errors` limit was reached. (Bug #17156507)
- With the thread pool plugin enabled and SSL in use, an error in one connection might affect other connections, causing them to experience a lost connection. (Bug #17087862)
- Under load, truncating the `accounts` Performance Schema table could cause a server exit. (Bug #17084615)
- `my_pthread.h` unconditionally included `pfs_thread_provider.h`, a noninstalled header file, resulting in compilation failure when compiling MySQL applications against the installed header files. (Bug #17061480)
- Indexed lookups on `POINT` columns was slower for `InnoDB` tables in MySQL 5.7 compared to 5.6. (Bug #17057168)
- The Performance Schema was built for embedded server builds. This no longer occurs. (Bug #17041705)

- Reads from message buffers for closed connections could occur. (Bug #17003702)
- The server could exit while using a cursor to fetch rows from a `UNION` query. (Bug #16983143)
- The range optimizer incorrectly assumed that any geometry function on a spatial index returned rows in ROWID order, which could result in incorrect query results. (Bug #16960800)
- `mysql_secure_installation` did not properly clean up the `mysql.proxies_privs` table for removed accounts. (Bug #16959850)
- A race condition in the thread pool plugin could cause status variables such as `Aborted_connects` not to be incremented and permitting concurrent kills to happen for the same thread ID. (Bug #16959022)
- At server startup, it was possible to set the `validate_password_length` system variable to a value smaller than permitted by the values of other password-length variables related to it. (Bug #16957721)
- Initialization of `keycache_*` variables (see Multiple Key Caches) during server startup could write to incorrect memory. (Bug #16945503)
- For debug builds, improper use of `SAFE_MUTEX` within `dbug.c` caused different code areas to have different ideas about size and contents of a mutex. This could result in out-of-bounds memory writes. (Bug #16945343)
- The Performance Schema could spawn a thread using incorrect instrumentation information. (Bug #16939689)
- The server did excessive locking on the `LOCK_active_mi` and `active_mi->rli->data_lock` mutexes for any `SHOW STATUS LIKE 'pattern'` statement, even when the pattern did not match status variables that use those mutexes (`Slave_heartbeat_period`, `Slave_last_heartbeat`, `Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`). Now attempts to show those variables do not lock those mutexes. This might result is slightly stale data, but better performance. (Bug #16904035)
- Full-text phrase search in `InnoDB` tables could read incorrect memory. (Bug #16885178)
- It was not possible to keep several major versions of MySQL in the same `yum` repository. (Bug #16878042)
- Excessive memory consumption was observed for multiple execution of a stored procedure under these circumstances: 1) The stored procedure had an SQL statement that failed during validation. 2) The stored procedure had an SQL statement that required repreparation. (Bug #16857395)
- The Batched Key Access method could return incorrect results on big-endian machines if a table buffered in the BKA join cache used a storage engine such as `InnoDB` or `MyISAM` with little-endian storage format, and then the next table used a storage engine such as `NDB` with native-endian storage format. (Bug #16853897)
- The error string for `ER_COL_COUNT_DOESNT_MATCH_PLEASE_UPDATE` string contained a hardcoded database name (`'mysql.%s'`), which is incorrect when the error referred to a table in a different database. (Bug #16813605)
- An assertion could be raised when the optimizer considered pushing down an index condition containing an updatable user variable and did not contain fields from the index. (Bug #16804581)
- If a `SET` statement containing a subquery caused a deadlock inside `InnoDB`, `InnoDB` rolled back the transaction. However, the SQL layer did not notice this and continued execution, resulting eventually in an assertion being raised inside `InnoDB`. (Bug #16802288)

- Removing a server RPM package did not shut down the existing server if it was running. (Bug #16798868)
- Some errors in MySQL 5.7 had different numbers than in MySQL 5.6. (Bug #16780120)
- A race condition in the server could cause issues with the `mysqld` process ID file when startup was aborted. As part of the fix for this issue, `mysqld_safe` now creates its own PID file `mysqld_safe.pid` in the server's data directory. (Bug #16776528)
- `HAVE_REPLICATION` now is set from `CMake` rather than in `my_global.h` so that it is not dependent on `my_global.h` having been included. (Bug #16768511)
- `INSERT ... ON DUPLICATE KEY UPDATE` could cause a server exit if a column with no default value was set to `DEFAULT`. (Bug #16756402)

  References: This issue is a regression of: Bug #14789787.
- `CMake` now assumes the existence of standard C header files such as `stdlib.h` and `stdarg.h`. (Bug #16748528)
- In a prepared statement or stored routine, if the `HAVING` clause of a subquery referenced some column of the GROUP BY of the parent query, the server could exit. (Bug #16739050)
- Compiling failed with `-DMY_ATOMIC_MODE_RWLOCKS=1` or on platforms on which MySQL did not support lockless atomic operations (such as ARM). (Bug #16736461)
- Password rewriting in the general query log now also applies to prepared statements. (Bug #16732621)
- The code base was modified to account for new warning checks introduced by `gcc` 4.8. (Bug #16729109)
- The function `fill_locks_row()`, which is responsible for providing data for the `INFORMATION_SCHEMA.INNODB_LOCKS` table, would try to look up the B-tree page in the buffer pool for `INFIMUM` and `SUPREMUM` records, both of which have a predefined `heap_no`. This generated unnecessary buffer pool contention and caused information to be omitted when a page was not available in the buffer pool. This fix removes the buffer pool lookup for `PAGE_HEAP_NO_INFIMUM` (`heap_no=0`) and `PAGE_HEAP_NO_SUPREMUM` (`heap_no=1`) from `fill_locks_row()`. (Bug #16684523)
- The deprecated `thread_concurrency` system variable has been removed. (Bug #16661195)
- The read-only `open_files_limit` system variable did not show the maximum number of open files the `mysqld` process could have, but instead the number that was requested after adjusting the `--open-files-limit` command-line option. (Bug #16657588)
- Kill handling in the thread pool plugin was subject to timeout problems and Valgrind warnings. (Bug #16633880)
- Overhead for setting `PROCESSLIST_STATE` values in the `THREADS` Performance Schema table has been reduced. (Bug #16633515)
- Within a stored procedure, repeated execution of a prepared `CREATE TABLE` statement for a table with partitions could cause a server exit. (Bug #16614004)
- The server could make the wrong decision about whether an account password was expired. (Bug #16604641)
- The Windows authentication plugin failed to free a context buffer for each connection. (Bug #16591288)
- Some rows for a session could be missing sporadically from the `session_connect_attrs` Performance Schema table while the session was executing a workload. (Bug #16576980)

- The `DBUG_PRINT()` macro unnecessarily evaluated arguments when debugging was not enabled. (Bug #16556597)
- Some problems compiling on Solaris in 64-bit mode with `gcc` and `g++` were corrected. (Bug #16555106)
- `SHOW WARNINGS` and `SHOW ERRORS` did not properly reset the warning count. (Bug #16522662)
- Clients could determine based on connection error message content whether an account existed. (Bug #16513435, Bug #17357528, Bug #19273967)
- Geometry methods that worked with WKB data performed insufficient input data validation, which could cause Valgrind errors or a server exit. (Bug #16510712, Bug #12772601)
- The server could attempt a `filesort` operation for a zero-size sort length, causing it to exit. (Bug #16503160)
- Opening a cursor on a `SELECT` within a stored procedure could cause a segmentation fault. (Bug #16499751)

  References: This issue is a regression of: Bug #14740889.
- `CREATE TABLE` or `ALTER TABLE` can fail if the statement specified unsupported options or something was missing. Previously, such errors were returned as `ER_ILLEGAL_HA`. Now they are returned as the new `ER_MISSING_HA_CREATE_OPTION` error. (Bug #16498740)
- Enabling the query cache could cause repeatable-read transactions to return incorrect results. (Bug #16497925)
- `my_load_defaults()` was modified to accommodate some problems under compilation with `gcc` 4.7.2 that could cause a client crash during option processing. (Bug #16497125)
- Missing variable initialization led to incorrectly returning an error from `st_select_lex_unit::explain` and led to a failed assertion. (Bug #16484966)
- When index condition pushdown was used on a descending range scan and the first range interval did not contain any qualifying records, the result of the range scan could be empty even if other range intervals contained qualifying records. (Bug #16483273)
- The WKB reader for spatial operations could fail and cause a server exit. (Bug #16451878)
- The `ER_OUTOFMEMORY` error was used in some places without the proper parameter being passed, resulting in incorrect diagnostic output. (Bug #16449659)
- Failure to handle a full-text search wildcard properly could cause the server to exit. (Bug #16446108)
- Optimizer heuristics inappropriately preferred range access over `ref` access in cases when the `ref` access referred to a column of a table earlier in the join seqence. (Bug #16437940)
- For queries using `ref` access on `CHAR` and `VARCHAR` data types, the `ref` access condition could be evaluated again as part of the query condition or pushed down as an index condition to the storage engine. (Bug #16437630)
- If the optimizer was using a loose index scan, the server could exit while attempting to create a temporary table. (Bug #16436567)
- Incorrect results or a server exit could be caused by a reference to an aggregated expression inside a nested subquery, where the aggregated expression was evaluated in a query block more than two levels outer to the reference. (Bug #16436383)

- If a table has been marked as containing only `NULL` values for all columns if it is a `NULL`-complemented row of an outer join, then rollup on the column which cannot be nullable results in a server exit. (Bug #16436014)
- Unlike `MyISAM`, `InnoDB` does not support boolean full-text searches on nonindexed columns, but this restriction was not enforced, resulting in queries that returned incorrect results. (Bug #16434374)
- Performance Schema parameter autosizing at startup did not take into account later autosizing changes to other startup parameters on which the Performance Schema parameters depended. (Bug #16430532)
- A full-text search syntax error failed to print to standard output. (Bug #16429688, Bug #16765397)
- Some `INFORMATION_SCHEMA` queries that used `ORDER BY` did not use a `filesort` optimization as they did in MySQL 5.5. (Bug #16423536)
- Debugging flags used to set the `debug` system variable were ignored if they were a prefix of keywords already in the debugging list. (Bug #16415978)
- Manually-created accounts (using `INSERT`) with a malformed password effectively had no password. (Bug #16414396)
- For debug builds, `DBUG_EXPLAIN` resulted in a buffer overflow when the `debug` system variable value was more than 255 characters. (Bug #16402143)
- Several scripts in the `sql-bench` directory that were supposed to be executable did not have the executable access bit set. (Bug #16395606)
- For debug builds, with an XA transaction in IDLE or PREPARED status, execution of a query with the query cache enabled could cause a server exit. (Bug #16388996)
- If the primary key for the `mysql.proc` system table was removed (an unsupported and not-recommended operation), the server exited for subsequent stored procedure invocation. Similar problems could occur for other system tables. Now an error occurs instead. (Bug #16373054)
- A server exit could occur for queries of the form `SELECT (SELECT 1 FROM t1) IN (SELECT a FROM t1)` when attempting to evaluate the constant left-hand argument to the `IN` subquery predicate. (Bug #16369522)
- An assertion could be raised when creating a index on a prefix of a `TINYBLOB` or `GEOMETRY` column in an `InnoDB` column. (Bug #16368875, Bug #18776592, Bug #17665767)
- If a lock timeout resulted from an `UPDATE` with a nested `SELECT` being unable to access rows being accessed by another thread, the error could go unchecked and cause an assertion to be raised later. (Bug #16367039)
- In debug builds, failure in the range optimizer for an `ER_LOCK_DEADLOCK` or `ER_LOCK_WAIT_TIMEOUT` error could go undetected and cause an assertion to be raised when a response was sent to the client. In release builds, this problem manifested as clients receiving an `OK` for a statement that had failed. (Bug #16366994, Bug #16247110)
- In debug builds, failure in the range optimizer for an `ER_LOCK_DEADLOCK` or `ER_LOCK_WAIT_TIMEOUT` error could go undetected and cause the `filesort` code to raise an assertion. In release builds, this problem manifested as clients receiving an `ER_FILSORT_ABORT` rather than the correct error code. (Bug #16366881)
- For debug builds, `set_field_to_null()` could raise an assertion for attempts to insert `NULL` into a `NOT NULL` column. (Bug #16362246)

- An assertion could be raised if, in greedy search mode, the optimizer identified join orders but was unable to choose one. (Bug #16361170)
- A race condition in `vio_shutdown()` could cause a server exit. (Bug #16354789)
- For debug builds, `GROUP_CONCAT(... ORDER BY)` within an `ORDER BY` clause could cause a server exit. (Bug #16347426)
- A `GROUP_CONCAT()` invocation containing subquery having an outer reference caused the server to exit. (Bug #16347343)
- The `validate_password` plugin did not always enforce appropriate constraints against assigning empty passwords. (Bug #16346443)
- Transforming some subqueries that select temporal or `BIGINT` types or to a semijoin caused a server exit on the second execution of prepared statements or stored programs. (Bug #16319671)
- Re-execution of a stored procedure could cause a server exit in `Item_field::fix_outer_field`. (Bug #16317443)
- For debug builds, the server could exit for queries involving a nested subquery, a subquery transformed into a semi-join and using a view. (Bug #16317076)
- The server could exit in `do_copy_not_null()` due to an improper `NULL`-value check. (Bug #16316564)
- No warning was generated if a duplicate index existed after dropping a column associated with a multiple-column index. (Bug #16315351)
- `SELECT DISTINCT` with `WITH ROLLUP` could result in a `Duplicate entry 'NULL' for key '<auto_key>'` error. (Bug #16314835)
- Oracle RPM packages were unusable by `yum` due to issues with the `obsoletes` line in the `.spec` file causing `yum` to interpret the package as obsoleting itself. (Bug #16298542)
- The range optimizer could set up incorrect ranges for queries that used `XOR` operations. (Bug #16272562)
- `mysql_secure_installation` could not connect to the server if the account used had an expired password. It invoked `mysql` noninteractively, resulting in that program failing to connect. Now `mysql` supports a `--connect-expired-password` option that indicates to the server that it can handle sandbox mode for expired-password accounts even if invoked noninteractively, and `mysql_secure_installation` invokes `mysql` with this option. (Bug #16248315)
- The usual failed-login attempt accounting was not applied to failed `COM_CHANGE_USER` commands. (Bug #16241992, Bug #17357535)
- For debug builds, an assertion could be raised if a failed `LOAD DATA INFILE` statement will followed by an `INSERT` for the same table within the same session. (Bug #16240526)
- If loose index scan was used on a query that used `MIN()`, a segmentation fault could occur. (Bug #16222245)
- For debug builds, an assertion was incorrectly raised for queries executed using `eq_ref` access and `filesort`. (Bug #16164885)
- A user variable referenced during execution of a prepared statement is set to memory that is freed at the end of execution. A second execution of the statement could result in Valgrind warnings when accessing this memory. (Bug #16119355)

- Misoptimization of left expressions in prepared statements could cause a server exit. (Bug #16095534)
- If `my_write()` encountered a disk-full condition, it could return an incorrect error value. (Bug #16078792, Bug #19984788)
- The server could exit the second time a stored routine was invoked that performed an `UPDATE` or `DELETE` using an invalid column in the join clause. (Bug #16078466)
- Certain queries containing `ORDER BY` or `SQL_CALC_FOUND_ROWS` could cause a server exit for JSON-format `EXPLAIN` statements. (Bug #16077396, Bug #16078113)
- A prepared statement that used `GROUP_CONCAT()` and an `ORDER BY` clause that named multiple columns could cause the server to exit. (Bug #16075310)
- `ORDER BY MATCH ... AGAINST` could cause a server exit. (Bug #16073689)
- Creating a `FEDERATED` table without specifying a connection string caused a server exit. (Bug #16048546)
- When a partition is missing, code in `ha_innodb.cc` would retry 10 times and sleep for a microsecond each time while holding `LOCK_open`. The retry logic for partitioned tables was introduced as a fix for Bug#33349 but did not include a test case to validate it. This fix removes the retry logic for partitioned tables. If the problem reported in Bug#33349 reappears, a different solution will be explored. (Bug #15973904)
- Client programs from MySQL 5.6.4 and up could confuse older servers during the connection process by using newer protocol features not understood by older servers. (Bug #15965409)
- The `mysql.server` script exited with an error if the `status` command was executed with multiple servers running. (Bug #15852074)
- In some cases, `REVOKE` could fail to revoke the `GRANT OPTION` privilege. (Bug #14799187)
- Use of the `VALUES()` function in the `VALUES()` clause of an `INSERT` statement could result in Valgrind warnings or an unstable server, possibly leading to a server exit. (Bug #14789787)
- The Debug Sync facility could lose a signal, leading to a spurious `ER_DEBUG_SYNC_TIMEOUT` error. (Bug #14765080, Bug #18221750)
- The `mysql` client allocated but did not free a string after reading each line in interactive mode, resulting in a memory leak. (Bug #14685362)
- The optimizer trace could print ranges for key parts that were not usable for range access. (Bug #14615536)
- Killing a connection while it was in the process of disconnecting could lead to an assertion being raised, Valgrind warnings, and general unstability. (Bug #14560522)
- Passwords in statements were not obfuscated before being written to the audit log. (Bug #14536456)
- When running a query on `INFORMATION_SCHEMA.INNODB_BUFFER_PAGE` that requested `table_name` and `index_name` values, query results would include index pages without `table_name` or `index_name` values. (Bug #14529666)
- Several `COM_xxx` commands in the client-server protocol did not have length checks for incoming network packets, which could result in various problems for malformed input. (Bug #14525642)
- If used to process a prepared `CALL` statement for a stored procedure with `OUT` or `INOUT` parameters, `mysql_stmt_store_result()` did not properly set the flags required to retrieve all the result sets. (Bug #14492429, Bug #17849978)

- `INSERT ... ON DUPLICATE KEY UPDATE` on a view could cause a server exit. (Bug #14261010)
- With the thread pool plugin in use, normal connection termination caused the `Aborted_clients` status variable to be incremented. (Bug #14081240)
- A build failure occurred if `HAVE_CRYPT` was 0. (Bug #14036425)
- Grouping by an outer `BLOB` column in a subquery caused a server exit. (Bug #13966809, Bug #14700180)
- On Windows, command-line options of the form `--opt_name="opt_value"` worked but `--opt_name='opt_value'` did not.

  On all platforms, for Performance Schema options of the form `--performance_schema_instrument="instrument=value"`, invalid instrument names now are rejected. (Bug #13955232)
- The server could exit after failing to handle an out-of-memory condition in `open_normal_and_derived_tables()`. (Bug #13553905)
- The server could exit due to improper handling of the error from an invalid comparison. (Bug #13009341)
- MySQL Installer, if run in custom install or change mode, offered installation options that had no effect. (Bug #12928601)
- Metadata returned for a prepared `SELECT` statement that had outer joins could indicate that columns containing `NULL` values were `NOT NULL`. (Bug #12818811)
- The thread pool plugin produced an error message containing an incorrect maximum `thread_pool_prio_kickup_timer` value. (Bug #12817590)
- For debug builds, the server could exit as a result of a series of statements that used a user variable such that its character set/collation changed from statement to statement. (Bug #12368577)
- Incorrect results could be returned from queries that used several *`aggr_func`*`(DISTINCT)` functions (where *`aggr_func`*`()` is an aggregate function such as `COUNT()`) when these referred to different columns of the same composite key. (Bug #12328597)
- Queries of the form `SELECT ... UNION SELECT ... ORDER BY` were parsed incorrectly, with the `ORDER BY` applied to the final `SELECT` rather than to the statement as a whole. (Bug #11886060)
- An identifier containing special characters could become too long after conversion of such characters to encoded format, resulting in SQL errors or failure to find files. (Bug #11766880)
- The `CMake` check for `unsigned time_t` failed on all platforms. (Bug #11766815)
- `mysql_convert_table_format` ignored `--engine` or `-e` as a synonym for the `--type` option. (Bug #11756950)
- `mysqladmin debug` causes the server to write debug information to the error log. On systems that supported `mallinfo()`, the memory-status part of this output was incorrect in 64-bit environments when `mysqld` consumed more than 4GB memory.

  Now the server uses `malloc_info()` to obtain memory-status information. `malloc_info()` does not report the memory that the `glibc` `malloc()` implementation internally allocates using `mmap()`. However, it does provide the memory usage information in all the memory arenas.

  This bug fix also involves a change of output format. The server now writes memory information in XML format rather than as plain text. Example:

```
Memory status:
<malloc version="1">
<heap nr="0">
<sizes>
<size from="33" to="33" total="1056" count="32"/>
<size from="65" to="65" total="65" count="1"/>
<size from="113" to="113" total="226" count="2"/>
<size from="129" to="129" total="2451" count="19"/>
<size from="145" to="145" total="290" count="2"/>
<size from="161" to="161" total="1288" count="8"/>
<size from="209" to="209" total="418" count="2"/>
</sizes>
<total type="fast" count="0" size="0"/>
<total type="rest" count="66" size="5794"/>
<system type="current" size="10833920"/>
<system type="max" size="10833920"/>
<aspace type="total" size="10833920"/>
<aspace type="mprotect" size="10833920"/>
</heap>
<total type="fast" count="0" size="0"/>
<total type="rest" count="66" size="5794"/>
<system type="current" size="10833920"/>
<system type="max" size="10833920"/>
<aspace type="total" size="10833920"/>
<aspace type="mprotect" size="10833920"/>
</malloc>
```

(Bug #11746658)

- The `DBUG_ENTER` string for the `THD::increment_questions_counter()` function incorrectly named the `THD::increment_updates_counter()` function. (Bug #69989, Bug #17297266)
- RPM packages did not provide lowercase tags for their contents. For example, a server RPM indicated that it provided `MySQL-server`, but not `mysql-server`. (Bug #69830, Bug #17211588)
- If the `WITH_SSL` `CMake` option was specified with an incorrect path to the SSL installation or the path to an unsupported (too old) SSL installation, the option was implicitly changed to the `bundled` value and yaSSL was used instead. Now `CMake` exits with an error so the user knows that the option value must be changed. (Bug #69744, Bug #17162055)
- When selecting a union of an empty result set (created with `WHERE 1=0` or `WHERE FALSE`) with a derived table, incorrect filtering was applied to the derived table. (Bug #69471, Bug #16961803)

  References: This issue is a regression of: Bug #15848521.
- For queries with `ORDER BY ... LIMIT`, the optimizer could choose a nonordering index for table access. (Bug #69410, Bug #16916596)
- If `query_cache_type` was disabled at startup to prevent the query cache from being enabled at runtime, disabling `query_cache_type` at runtime generated a warning even though it was already disabled. (Bug #69396, Bug #16906827)
- When an internal buffer was too small for the workload, the Performance Schema could spend a lot of time in an internal spin loop attempting to allocate a memory buffer, and fail. (Bug #69382, Bug #16945618)
- In the absence of `SQL_CALC_FOUND_ROWS` in the preceding query, `FOUND_ROWS()` should return the number of rows in the result set, but this did not always happen if the query contained `ORDER BY`. (Bug #69271, Bug #16827872)

- Full-text search on `InnoDB` tables failed on searches for words containing apostrophes. (Bug #69216, Bug #16801781)
- The `libmysql.dll` library was missing several symbols: `my_init`, `mysql_client_find_plugin`, `mysql_client_register_plugin`, `mysql_load_plugin`, `mysql_load_plugin_v`, `mysql_options4`, and `mysql_plugin_options`. (Bug #69204, Bug #16797982, Bug #62394)
- If an `UPDATE` containing a subquery caused a deadlock inside `InnoDB`, the deadlock was not properly handled by the SQL layer. The SQL layer then tried to unlock the row after `InnoDB` rolled back the transaction, raising an assertion inside `InnoDB`. (Bug #69127, Bug #16757869)
- `FOUND_ROWS()` could return an incorrect value if the preceding query used `filesort`. (Bug #69119, Bug #16760474)

  References: This issue is a regression of: Bug #68458.
- The optimizer could choose a poor execution plan for queries with `ORDER BY ... LIMIT`. (Bug #69013, Bug #16697792)
- Some possible cases of memory use after being freed were fixed. Thanks to Jan Staněk for the patch. (Bug #68918, Bug #16725945)
- Some `LEFT JOIN` queries with `GROUP BY` could return incorrect results. (Bug #68897, Bug #16620047)

  References: This issue is a regression of: Bug #11760517.
- Some errors could be handled by condition handlers only if they were raised by particular statements, such as `INSERT`, but not if they were raised by other statements, like `UPDATE`. An example would be the foreign-key error `ER_NO_REFERENCED_ROW_2` which could be treated differently, depending on which statement raised it. (Bug #68831, Bug #16587369)
- When specified in an option file, the `plugin-dir` client option was ignored. (Bug #68800, Bug #16680313)
- Comparison of a `DATETIME` value and a string did not work correctly for the `utf8_unicode_ci` collation. (Bug #68795, Bug #16567381)
- When only counting events but not timing them, Performance Schema would report `MIN_TIMER_WAIT` values as a large number instead of 0. (Bug #68768, Bug #16552425)
- Using range access with an index prefix could produce incorrect results. (Bug #68750, Bug #16540042)
- Full-text search on `InnoDB` tables failed on searches for literal phrases combined with `+` or `-` operators. (Bug #68720, Bug #16516193)
- For debug builds, metadata locking for `CREATE TABLE ... SELECT` could raise an assertion. (Bug #68695, Bug #16503173)
- Compilation on Solaris using `gcc` produced incorrect builds for 32-bit systems. (Bug #68675)
- `mysqld --help` and `mysqld --verbose --help` performed unnecessary logging. (Bug #68578, Bug #16442113)
- A new `CMake` option, `WITH_EDITLINE`, is provided to indicate whether to use the bundled or system `libedit`/`editline` library. The permitted values are `bundled` (the default) and `system`.

  `WITH_EDITLINE` replaces `WITH_LIBEDIT`, which has been removed. (Bug #68558, Bug #16430208)
- Overhead for the `skip_trailing_space()` function was reduced. (Bug #68477, Bug #16395778)

- If Loose Index Scan was used to evaluate a query that compared an integer column to an integer specified as a quoted string (for example, `col_name = '1'`), the query could return incorrect results. (Bug #68473, Bug #16394084)
- Optimizations that used extended secondary keys (see Use of Index Extensions) worked only for `InnoDB`, even for storage engines with the requisite underlying capabilities. (Bug #68469, Bug #16391678)
- `mysql_install_db` incorrectly tried to create the `mysql.innodb_table_stats` and `mysql.innodb_index_stats` tables if `InnoDB` was not available. (Bug #68438, Bug #16369955)
- `BIT(0)` is not a valid data type specification but was silently converted to `BIT(1)`. Now an `ER_INVALID_FIELD_SIZE` error occurs and the specification is rejected. (Bug #68419, Bug #16358989)
- In a MySQL server newer than MySQL 5.5 using a nonupgraded `mysql.user` table (for which `mysql_upgrade` had not been run), statements to set passwords caused a server exit due to a faulty check for the `password_expired` column. (Bug #68385, Bug #16339767)
- Indexes on derived tables that were used during the first invocation of a stored procedure were not used in subsequent invocations. (Bug #68350, Bug #16346367)
- If a function such as `AES_DECRYPT()` that requires SSL support failed, the error could affect later calls to functions that require SSL support. (Bug #68340, Bug #16315767)
- For `DELETE` and `UPDATE` statements, `EXPLAIN` displayed `NULL` in the `ref` column for some cases where `const` is more appropriate. (Bug #68299, Bug #16296268)
- The `mysql` client incorrectly used `latin1` for certain comparisons even if started with a multibyte default character set, resulting in a client crash. (Bug #68107, Bug #16182919)
- In option files, the server could misinterpret option settings if the value was given after the option name with no `=` sign in between. (Bug #67740, Bug #15930031)
- Performance of prepared DML statements containing `?` parameter substitution markers was improved under row-based logging format: Since the binary log in this case need not include the statement text, and since the statement will not be forced to statement-based logging as some DDL statements might be, there is no need to substitute `?` markers to produce a statement suitable for logging. (Bug #67676, Bug #16038776)
- `ELT(LEAST(...),..)` could return a non-`NULL` value even if `LEAST()` returned `NULL`. (Bug #67578, Bug #16171537)
- If the server could not find the `errmsg.sys` file at startup, the resulting error message did not indicate which configuration parameter to check. (Bug #67576, Bug #15880018)
- `mysqldump` wrote `SET` statements as `SET OPTION`, which failed when reloaded because the deprecated `OPTION` keyword has been removed from `SET` syntax. (Bug #67507, Bug #15844882)
- For failure to create a new thread for the event scheduler, event execution, or new connection, no message was written to the error log. This could lead to the impression that the event scheduler was running normally when it was not. (Bug #67191, Bug #14749800, Bug #16865959)
- Configuring with `cmake -DWITHOUT_SERVER` to build clients without the server failed for builds outside of the source tree. (Bug #66000, Bug #14367046)
- `mysqldump` assumed the existence of the `general_log` and `slow_log` tables in the `mysql` database. It failed if invoked to dump tables from an older server where these tables do not exist. (Bug #65670, Bug #14236170)

- If an account had a nonzero `MAX_USER_CONNECTIONS` value, that value was not always respected. (Bug #65104, Bug #14003080)
- Attempts to build from a source RPM package could fail because the build process attempted to refer to a `pb2user` that might not exist. (Bug #64641, Bug #13865797, Bug #69339, Bug #16874980)
- When an `ALTER TABLE` operation was performed with an invalid foreign key constraint, the error reported was `ER_CANT_CREATE_TABLE` rather than `ER_CANNOT_ADD_FOREIGN`. (Bug #64617, Bug #13840553)
- If one session had any metadata lock on a table, another session attempting `CREATE TABLE [IF NOT EXISTS]` for the same table would hang. This occurred due to an attempt in the second session to acquire an exclusive metadata lock on the table before checking whether the table already existed. An exclusive metadata lock is not compatible with any other metadata locks, so the session hung for the lock timeout period if another session had the table locked.

  Now the server attempts to acquire a shared metadata lock on the table first to check whether it exists, then upgrade to an exclusive lock if it does not. If the table does exist, an error occurs for `CREATE TABLE` and a warning for `CREATE TABLE IF NOT EXISTS`. (Bug #63144, Bug #13418638)
- `sql-common/client_plugin.c` contained a nonportable use of a `va_list` parameter. (Bug #62769, Bug #13252623)
- `InnoDB` does not support full-text parser plugins, but failed to report an error if they were specified. Now an `ER_INNODB_NO_FT_USES_PARSER` error is returned. (Bug #62004, Bug #12843070)
- The `url` columns in the `mysql` datatbase help tables were too short to hold some of the URLs in the help content. These columns are now created as type `TEXT` to accommodate longer URLs. (Bug #61520, Bug #12671635)
- A typo in `cmake/dtrace.cmake` prevented DTrace support from being enabled by `-DENABLE_DTRACE-on`. (Bug #60743, Bug #12325449)
- The Turbo Boyer-Moore algorithm used for `LIKE` pattern matches failed to handle some patterns. The server now uses the original Boyer-Moore algorithm. (Bug #59973, Bug #11766777)
- Boolean plugin system variables did not behave well on machines where `char` is unsigned; some code attempted to assign a negative value to these. (Bug #59905, Bug #11864205)
- Some subquery transformations were not visible in `EXPLAIN` output. (Bug #59852, Bug #11766685)
- Configuring MySQL with `-DWITH_EXTRA_CHARSETS=none` caused a build failure. (Bug #58672, Bug #11765682)
- Two problems adding or subtracting keyword from the current `debug` system variable setting were corrected:
  - A `debug` value of `'d'` means “all debug macros enabled”. The following sequence left the value in an incorrect state:

```
mysql> SET debug = 'd';SELECT @@debug;
+---------+
| @@debug |
+---------+
| d       |
+---------+


mysql> SET debug = '+d,M1';SELECT @@debug;
```

```
+---------+
| @@debug |
+---------+
| d,M1    |
+---------+
```

The first `SET` statement enables all debug macros. The second `SET` should add the `M1` macro to the current set, which should result in no change because the current set is already “all macros”. Instead, the second `SET` reset the current set to only the `M1` macro, effectively disabling all others. The server now correctly leaves `debug` set to `'d'`.

- A `debug` value of `''` means “no debug macros enabled”. The following sequence left the value in an incorrect state:

```
mysql> SET debug = 'd,M1';SELECT @@debug;
+---------+
| @@debug |
+---------+
| d,M1    |
+---------+


mysql> SET debug = '-d,M1';SELECT @@debug;
+---------+
| @@debug |
+---------+
| d       |
+---------+
```

The first `SET` statement sets `debug` to the `M1*` macro. The second `SET` should subtract the `M1` macro from the current set, leaving no debug macros enabled. Instead, the second `SET` reset the current set to `'d'` (all macros enabled). The server now correctly sets `debug` to `''`.

(Bug #58630, Bug #11765644)

- It is now possible to suppress installation of the `mysql-test` directory after compiling MySQL from source by invoking `CMake` with the `INSTALL_MYSQLTESTDIR` option explicitly set to empty:

```
cmake . -DINSTALL_MYSQLTESTDIR=
```

Previously, attempts to do this resulted in an error. (Bug #58615, Bug #11765629)

- On 64-bit Mac OS X systems, `CMake` used `x86` rather than `x86_64` when determining the machine type. (Bug #58462, Bug #11765489)
- Long table or column names could cause `mysqlshow` to exit. (Bug #53956, Bug #11761458)
- With `big_tables` enabled, queries that used `COUNT(DISTINCT)` on a simple join with a constant equality condition on a non-duplicate key returned incorrect results. (Bug #52582, Bug #11760197)

  References: See also: Bug #18853696.

- The `!includedir` directive in option files did not read `.cnf` or `.ini` files that included a dot in the file name preceding the extension. (Bug #51609, Bug #11759306)
- Successful queries served from the query cache did not clear warnings. (Bug #49634, Bug #11757567)

- If `ALTER TABLE` was used to set the default value for a `TIMESTAMP` or `DATETIME` column that had `CURRENT_TIMESTAMP` as its default when it was created, the new default was not shown by `SHOW CREATE TABLE`, and incorrect values could be inserted into the column. (Bug #45669, Bug #11754116)
- `IF()` function evaluations could produce different results when executed in a prepared versus nonprepared statement. (Bug #45370, Bug #11753852)
- The range optimizer used the wrong prerequisite for concluding that a table is the inner table of an outer join. This led to incorrect cost estimates and choice of the wrong index for query processing. (Bug #37333, Bug #11748775)
- For better robustness against stack overflow, the server now accounts for the size of the guard area when making thread stack size requests. (Bug #35019, Bug #11748074)
- If `mysqld` crashed during a shutdown initiated by `/etc/init.d/mysql stop`, `mysqld_safe` restarted `mysqld` when it should not have. (Bug #34084, Bug #13864548)
- `mysql.h` no longer defines `__WIN__` on Windows, and the MySQL sources have been changed to test for `_WIN32` instead. (Bug #20338, Bug #11745828)

Changes in MySQL 5.7.1 (2013-04-23, Milestone 11)

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.

Audit Log Notes

- Several changes were made to the audit log plugin for better compatibility with Oracle Audit Vault.

  The format of the audit log file has changed:

  - Information within `<AUDIT_RECORD>` elements written in the old format using attributes is written in the new format using subelements.
  - The new format includes more information in `<AUDIT_RECORD>` elements. Every element includes a `RECORD_ID` value providing a unique identifier. The `TIMESTAMP` value includes time zone information. Query records include `HOST`, `IP`, `OS_LOGIN`, and `USER` information, as well as `COMMAND_CLASS` and `STATUS_CODE` values.

    The `STATUS_CODE` value differs from the existing `STATUS` value: `STATUS_CODE` is 0 for success and 1 for error, which is compatible with the EZ_collector consumer for Audit Vault. `STATUS` is the value of the `mysql_errno()` C API function. This is 0 for success and nonzero for error, and thus is not necessarily 1 for error.

  Example of old `<AUDIT_RECORD>` format:

```
<AUDIT_RECORD
 TIMESTAMP="2013-04-15T15:27:27" NAME="Query" CONNECTION_ID="3" STATUS="0" SQLTEXT="SELECT 1" />
```

Example of new `<AUDIT_RECORD>` format:

```
<AUDIT_RECORD>
 <TIMESTAMP>2013-04-15T15:27:27 UTC</TIMESTAMP>
 <RECORD_ID>3998_2013-04-15T15:27:27</RECORD_ID>
 <NAME>Query</NAME>
 <CONNECTION_ID>3</CONNECTION_ID>
 <STATUS>0</STATUS>
 <STATUS_CODE>0</STATUS_CODE>
 <USER>root[root] @ localhost [127.0.0.1]</USER>
 <OS_LOGIN></OS_LOGIN>
 <HOST>localhost</HOST>
 <IP>127.0.0.1</IP>
 <COMMAND_CLASS>select</COMMAND_CLASS>
 <SQLTEXT>SELECT 1</SQLTEXT>
</AUDIT_RECORD>
```

When the audit log plugin rotates the audit log file, it uses a different file name format. For a log file named `audit.log`, the plugin previously renamed the file to `audit.log.TIMESTAMP`. The plugin now renames the file to `audit.log.TIMESTAMP.xml` to indicate that it is an XML file.

For information about the audit log plugin, see MySQL Enterprise Audit.

If you previously used an older version of the audit log plugin, use this procedure to avoid writing new-format log entries to an existing log file that contains old-format entries:

1. Stop the server.
2. Rename the current audit log file manually. This file will contain only old-format log entries.
3. Update the server and restart it. The audit log plugin will create a new log file, which will contain only new-format log entries.

The API for writing audit plugins has also changed. The `mysql_event_general` structure has new members to represent client host name and IP address, command class, and external user. For more information, see Writing Audit Plugins.

Functionality Added or Changed

- **Performance:** String hashing overhead was reduced. This also improves performance for metadata locking, the table definition cache, and Performance Schema table I/O and file I/O instrumentation. (Bug #13944392)
- **Incompatible Change:** `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` output used a mix of `row_count` and `count` attributes. These are now all `count`. Similarly, the output used a mix of `row_size` and `size` attributes. These are now all `size`. (Bug #16165468)
- **InnoDB:** Added a separate tablespace for all non-compressed `InnoDB` temporary tables. The new tablespace is always recreated on server startup.

  The new tablespace, `ibtmp1`, is located in the MySQL data directory (`datadir`) by default. A newly added configuration file option, `innodb_temp_data_file_path`, allows for a user-defined temporary data file path. For related information, see InnoDB Temporary Table Undo Logs.
- **InnoDB:** Prior to this release, `InnoDB` stored spatial data types as binary `BLOB` data, mapped to the internal `DATA_BLOB` data type. `BLOB` remains the underlying data type but spatial data types are now

mapped to a new internal data type, DATA_GEOMETRY. With BLOB as the underlying data type, a prefix index can still be used on all GEOMETRY data type columns.

- **InnoDB:** InnoDB temporary table metadata is no longer stored in InnoDB system tables. Instead, a new INFORMATION_SCHEMA table, INNODB_TEMP_TABLE_INFO, provides users with a snapshot of active temporary tables. The table contains metadata and reports on all user and system-created temporary tables that are active within a given InnoDB instance.
- **InnoDB:** DDL performance for InnoDB temporary tables is improved through optimization of CREATE TABLE, DROP TABLE, TRUNCATE TABLE, and ALTER TABLE statements. Optimizations were achieved by limiting actions performed by DDL statements to only those necessary for temporary tables.
- **InnoDB:** VARCHAR size may be increased using an in-place ALTER TABLE, as in this example:

```
ALTER TABLE t1 ALGORITHM=INPLACE, CHANGE COLUMN c1 c1 VARCHAR(255);
```

  This is true as long as the number of length bytes required by a VARCHAR column remains the same. For VARCHAR values of 0 to 255, one length byte is required to encode the value. For VARCHAR values of 256 bytes or more, two length bytes are required. As a result, in-place ALTER TABLE only supports increasing VARCHAR size from 0 to 255 bytes or increasing VARCHAR size from a value equal to or greater than 256 bytes. In-place ALTER TABLE does not support increasing VARCHAR size from less than 256 bytes to a value equal to or greater than 256 bytes. In this case, the number of required length bytes would change from 1 to 2, which is only supported by a table copy (ALGORITHM=COPY).

  Decreasing VARCHAR size using in-place ALTER TABLE is not supported. Decreasing VARCHAR size requires a table copy (ALGORITHM=COPY).

  For additional details, refer to the “Column Properties” information in Overview of Online DDL.

- **InnoDB:** Online index renaming is supported by ALTER TABLE, which now includes a RENAME INDEX clause, as shown in the following example: “ALTER TABLE t RENAME INDEX i1 TO i2”, where i1 is the current name of the index and i2 is the new name.

  The result of “ALTER TABLE t RENAME INDEX i1 TO i2” would be a table with contents and structure that is identical to the old version of “t1” except for the index name, which is now “i2” instead of “i1”.

- **Partitioning:** HANDLER statements are now supported with partitioned tables.
- **Replication:** An Auto_Position column has been added to the output generated by SHOW SLAVE STATUS. The value of this column shows whether replication autopositioning is in use. If autopositioning is enabled—that is, if MASTER_AUTO_POSITION = 1 was set by the last successful CHANGE MASTER TO statement that was executed on the slave—then the column's value is 1; if not, then the value is 0. (Bug #15992220)
- **Replication:** The functions GTID_SUBTRACT() and GTID_SUBSET() were formerly available in libmysqld only when it was built with replication support. Now these functions are always available when using this library, regardless of how it was built.
- **Replication:** Added the --rewrite-db option for mysqlbinlog, which allows mysqlbinlog to rewrite the names of databases when playing back binary logs written using the row-based logging format. Multiple rewrite rules can be created by specifying the option multiple times.
- MySQL no longer uses the default OpenSSL compression. (Bug #16235681)
- There is now a distinct error code (ER_MUST_CHANGE_PASSWORD_LOGIN) for the error sent by the server to a client authenticating with an expired password. (Bug #16102943)

- In RPM packages built for Unbreakable Linux Network, `libmysqld.so` now has a version number. (Bug #15972480)
- Error messages for `ALTER TABLE` statement using a `LOCK` or `ALGORITHM` value not supported for the given operation were very generic. The server now produces more informative messages. (Bug #15902911)
- If a client with an expired password connected but `old_passwords` was not the value required to select the password hashing format appropriate for the client account, there was no way for the client to determine the proper value. Now the server automatically sets the session `old_passwords` value appropriately for the account authentication method. For example, if the account uses the `sha256_password` authentication plugin, the server sets `old_passwords=2`. (Bug #15892194)
- `mysqldump` now supports an `--ignore-error` option. The option value is a comma-separated list of error numbers specifying the errors to ignore during `mysqldump` execution. If the `--force` option is also given to ignore all errors, `--force` takes precedence. (Bug #15855723)
- `mysql_config_editor` now supports `--port` and `--socket` options for specifying TCP/IP port number and Unix socket file name. (Bug #15851247)
- `mysqlcheck` has a new `--skip-database` option. The option value is the name of a database (case sensitive) for which checks should be skipped.

  `mysql_upgrade` adds this option to `mysqlcheck` commands that it generates to upgrade the system tables in the `mysql` database before tables in other databases: It upgrades the `mysql` database, then all databases except the `mysql` database. This avoids problems that can occur if user tables are upgraded before the system tables. (Bug #14697538, Bug #68163, Bug #16216384)
- The `validate_password_policy_number` system variable was renamed to `validate_password_policy`. (Bug #14588121)
- Previously, on Linux the server failed to perform stack backtrace attempts for versions of `glibc` older than the current minimum supported version (2.3). Now on such attempts the server displays a message that the `glibc` version is too old to support backtracing. (Bug #14475946)
- In JSON-format `EXPLAIN` output, the `attached_condition` information for subqueries now includes `select#` to indicate the relative order of subquery execution. (Bug #13897507)
- The following changes were made to the sandbox mode that the server uses to handle client connections for accounts with expired passwords:

  - There is a new `disconnect_on_expired_password` system variable (default: enabled). This controls how the server treats expired-password accounts.
  - Two flags were added to the C API client library: `MYSQL_OPT_CAN_HANDLE_EXPIRED_PASSWORDS` for `mysql_options()` and `CLIENT_CAN_HANDLE_EXPIRED_PASSWORDS` for `mysql_real_connect()`. Each flag enables a client program to indicate whether it can handle sandbox mode for accounts with expired passwords.

    `MYSQL_OPT_CAN_HANDLE_EXPIRED_PASSWORDS` is enabled for `mysqltest` unconditionally, for `mysql` in interactive mode, and for `mysqladmin` if the first command is `password`.

  For more information about how the client-side flags interact with `disconnect_on_expired_password`, see Password Expiration and Sandbox Mode. (Bug #67568, Bug #15874023)
- If a user attempted to access a nonexistent column for which the user had no access, the server returned an error indicating that the column did not exist. Now the server returns an error indicating that

the user does not have privileges for the column, which provides no information about column existence. (Bug #19947, Bug #11745788)

- The MySQL test suite `mysql-test-run.sh` program now starts the server with `InnoDB` rather than `MyISAM` as the default storage engine. To maintain compatibility of test results with existing result files, test cases were modified to add a line that includes the `force_default_myisam.inc` file as necessary. In a future release, for those test cases not specifically requiring `MyISAM`, that line will be removed (so they run with `InnoDB`) and test results will be updated.

- `ALTER TABLE` now supports a `RENAME INDEX` clause that renames an index. The change is made in place without a table-copy operation.

- The `mysql` client now has a `--syslog` option that causes interactive statements to be sent to the system `syslog` facility. Logging is suppressed for statements that match the default "ignore" pattern list (`"*IDENTIFIED*:*PASSWORD*"`), as well as statements that match any patterns specified using the `--histignore` option. For more information, see mysql Logging.

- The deprecated `innodb_mirrored_log_groups` system variable has been removed.

Bugs Fixed

- **Performance; InnoDB:** The `DROP TABLE` statement for a table using compression could be slower than necessary, causing a stall for several seconds. MySQL was unnecessarily decompressing pages in the buffer pool related to the table as part of the `DROP` operation. (Bug #16067973)

- **Incompatible Change; Partitioning:** Changes in the `KEY` partitioning hashing functions used with numeric, date and time, `ENUM`, and `SET` columns in MySQL 5.5 makes tables using partitioning or subpartitioning by `KEY` on any of the affected column types and created on a MySQL 5.5 or later server incompatible with a MySQL 5.1 server. This is because the partition IDs as calculated by a MySQL 5.5 or later server almost certainly differ from those calculated by a MySQL 5.1 server for the same table definition and data as a result of the changes in these functions.

  The principal changes in the `KEY` partitioning implementation in MySQL 5.5 resulting in this issue were as follows: 1. The hash function used for numeric and date and time columns changed from binary to character-based. 2. The base used for hashing of `ENUM` and `SET` columns changed from `latin1 ci` characters to binary.

  The fix involves adding the capability in MySQL 5.5 and later to choose which type of hashing to use for `KEY` partitioning, which is implemented with a new `ALGORITHM` extension to the `PARTITION BY KEY` option for `CREATE TABLE` and `ALTER TABLE`. Specifying `PARTITION BY KEY ALGORITHM=1 ([columns])` causes the server to use the hashing functions as implemented in MySQL 5.1; using `ALGORITHM=2` causes the server to use the hashing functions from MySQL 5.5 and later. `ALGORITHM=2` is the default. Using the appropriate value for `ALGORITHM`, you can perform any of the following tasks:

  - Create `KEY` partitioned tables in MySQL 5.5 and later that are compatible with MySQL 5.1, using `CREATE TABLE ... PARTITION BY KEY ALGORITHM=1 (...)`.

  - Downgrade `KEY` partitioned tables that were created in MySQL 5.5 or later to become compatible with MySQL 5.1, using `ALTER TABLE ... PARTITION BY KEY ALGORITHM=1 (...)`.

  - Upgrade `KEY` partitioned tables originally created in MySQL 5.1 to use hashing as in MySQL 5.5 and later, using `ALTER TABLE ... PARTITION BY KEY ALGORITHM=2 (...)`.

    Important: After such tables are upgraded, they cannot be used any longer with MySQL 5.1 unless they are first downgraded again using `ALTER TABLE ... PARTITION BY KEY ALGORITHM=1 (...)` on a MySQL server supporting this option.

This syntax is not backward compatible, and causes errors in older versions of the MySQL server. When generating `CREATE TABLE ... PARTITION BY KEY` statements, `mysqldump` brackets any occurrence of `ALGORITHM=1` or `ALGORITHM=2` in conditional comments such that it is ignored by a MySQL server whose version is not at least 5.5.31. An additional consideration for upgrades is that MySQL 5.6 servers prior to MySQL 5.6.11 do not ignore the `ALGORITHM` option in such statements when generated by a MySQL 5.5 server, due to the that the conditional comments refer to version 5.5.31; in this case, you must edit the dump manually and remove or comment out the option wherever it occurs before attempting to load it into a MySQL 5.6.10 or earlier MySQL 5.6 server. This is not an issue for dumps generated by MySQL 5.6.11 or later version of `mysqldump`, where the version used in such comments is 5.6.11. For more information, see ALTER TABLE Partition Operations.

As part of this fix, a spurious assertion by `InnoDB` that a deleted row had previously been read, causing the server to assert on delete of a row that the row was in the wrong partition, was also removed. (Bug #14521864, Bug #66462, Bug #16093958, Bug #16274455)

References: See also: Bug #11759782.

- **Incompatible Change:** For debug builds, creating an InnoDB table in strict SQL mode that violated the maximum key length limit caused the server to exit.

  A behavior change in consequence of this bug fix: In strict SQL mode, a key length limit violation now results in a error (and the table is not created), rather than a warning and truncation of the key to the maximum key length. This applies to all storage engines. (Bug #16035659)

- **Important Change; Plugin API; Replication:** Because the behavior of the fulltext plugin may vary between MySQL servers, it is not possible to guarantee that statements using this plugin produce the same results on masters and slaves. For this reason, statements depending on the fulltext plugin are now marked as unsafe for statement-based logging. This means that such statements are logged using row format when `binlog_format=MIXED`, and cause a warning to be generated when `binlog_format=STATEMENT`. (Bug #11756280, Bug #48183)

- **Important Change; Replication**

  

  Important

  This fix was reverted in MySQL 5.7.2. See Changes in MySQL 5.7.2 (2013-09-21, Milestone 12).

  Executing a statement that performs an implicit commit but whose changes are not logged when `gtid_next` is set to any value other than `AUTOMATIC` is not permitted. Now in such cases, the statement fails with an error. This includes the statements in the following list:

  - `CHANGE MASTER TO`
  - `START SLAVE`
  - `STOP SLAVE`
  - `REPAIR TABLE`
  - `OPTIMIZE TABLE`
  - `ANALYZE TABLE`
  - `CHECK TABLE`

- `CREATE SERVER`
- `ALTER SERVER`
- `DROP SERVER`
- `CACHE INDEX`
- `LOAD INDEX INTO CACHE`
- `FLUSH`
- `RESET`

(Bug #16062608)

References: See also: Bug #16484323.

- **Important Change; Replication:** The version number reported by `mysqlbinlog --version` has been increased to 3.4. (Bug #15894381, Bug #67643)
- **Important Change; Replication:** The lettercasing used for displaying UUIDs in global transaction identifiers was inconsistent. Now, all GTID values use lowercase, including those shown in the `Retrieved_Gtid_Set` and `Executed_Gtid_Set` columns from the output of `SHOW SLAVE STATUS`. (Bug #15869441)
- **Important Note; Replication:** Using row-based logging to replicate from a table to a same-named view led to a failure on the slave. Now, when using row-based logging, the target object type is checked prior to performing any DML, and an error is given if the target on the slave is not actually a table.

  

  Note

  It remains possible to replicate from a table to a same-named view using statement-based logging.

  (Bug #11752707, Bug #43975)
- **MySQL Cluster:** The setting for the `DefaultOperationRedoProblemAction` API node configuration parameter was ignored, and the default value used instead. (Bug #15855588)
- **MySQL Cluster:** Job buffers act as the internal queues for work requests (signals) between block threads in `ndbmtd` and could be exhausted if too many signals are sent to a block thread.

  Performing pushed joins in the `DBSPJ` kernel block can execute multiple branches of the query tree in parallel, which means that the number of signals being sent can increase as more branches are executed. If `DBSPJ` execution cannot be completed before the job buffers are filled, the data node can fail.

  This problem could be identified by multiple instances of the message `sleeploop 10!!` in the cluster out log, possibly followed by `job buffer full`. If the job buffers overflowed more gradually, there could also be failures due to error 1205 (`Lock wait timeout exceeded`), shutdowns initiated by the watchdog timer, or other timeout related errors. These were due to the slowdown caused by the 'sleeploop'.

  Normally up to a 1:4 fanout ratio between consumed and produced signals is permitted. However, since there can be a potentially unlimited number of rows returned from the scan (and multiple scans of this

type executing in parallel), any ratio greater 1:1 in such cases makes it possible to overflow the job buffers.

The fix for this issue defers any lookup child which otherwise would have been executed in parallel with another is deferred, to resume when its parallel child completes one of its own requests. This restricts the fanout ratio for bushy scan-lookup joins to 1:1. (Bug #14709490)

References: See also: Bug #14648712.

- **MySQL Cluster:** The recently added LCP fragment scan watchdog occasionally reported problems with LCP fragment scans having very high table id, fragment id, and row count values.

  This was due to the watchdog not accounting for the time spent draining the backup buffer used to buffer rows before writing to the fragment checkpoint file.

  Now, in the final stage of an LCP fragment scan, the watchdog switches from monitoring rows scanned to monitoring the buffer size in bytes. The buffer size should decrease as data is written to the file, after which the file should be promptly closed. (Bug #14680057)

- **InnoDB:** When parsing a delimited search string such as “abc-def” in a full-text search, `InnoDB` now uses the same word delimiters as `MyISAM`. (Bug #16419661)

- **InnoDB:** This fix improves code readability by addressing naming inconsistencies for InnoDB `PERFORMANCE_SCHEMA` key declarations. (Bug #16414044)

- **InnoDB:** Status values in the `innodb_ft_config` table would not update. The `innodb_ft_config` is intended for internal configuration and should not be used for statistical information purposes. To avoid confusion, column values that are intended for internal use have been removed from the `innodb_ft_config` table. This fix also removes the `innodb_ft_config` table and other internal full text search-related tables that were unintentionally exposed. (Bug #16409494, Bug #68502)

- **InnoDB:** This fix disables a condition for extra splitting of clustered index leaf pages, on compressed tables. Extra page splitting was only done to reserve space for future updates, so that future page splits could be avoided. (Bug #16401801)

- **InnoDB:** For `InnoDB` tables, if a `PRIMARY KEY` on a `VARCHAR` column (or prefix) was empty, index page compression could fail. (Bug #16400920)

- **InnoDB:** The `InnoDB` page-splitting algorithm could recurse excessively. (Bug #16345265)

- **InnoDB:** Improper testing of compatibility between the referencing and referenced during `ALTER TABLE ... ADD FOREIGN` key could cause a server exit. (Bug #16330036)

- **InnoDB:** Importing a tablespace with the configuration file present would not import the data file. This problem would occur when all pages are not flushed from the buffer pool after a table is altered using the copy and rename approach. This fix ensures that all pages are flushed from the buffer pool when a table is altered using the copy and rename approach. (Bug #16318052)

- **InnoDB:** Rollback did not include changes made to temporary tables by read-only transactions. (Bug #16310467)

- **InnoDB:** When using `ALTER TABLE` to set an `AUTO_INCREMENT` column value to a user-specified value, InnoDB would set the `AUTO_INCREMENT` value to the user-specified value even when the `AUTO_INCREMENT` value is greater than the user-specified value. This fix ensures that the `AUTO_INCREMENT` value is set to the maximum of the user-specified value and MAX(auto_increment_column)+1, which is the expected behaviour. (Bug #16310273)

- **InnoDB:** For debug builds, `InnoDB` status exporting was subject to a race condition that could cause a server exit. (Bug #16292043)
- **InnoDB:** With `innodb_api_enable_mdl=OFF`, an `ALTER TABLE` operation on an `InnoDB` table that required a table copy could cause a server exit. (Bug #16287411)
- **InnoDB:** An assertion failure would occur in `heap->magic_n == MEM_BLOCK_MAGIC_N` due to a race condition that appeared when `row_merge_read_clustered_index()` returned an error. (Bug #16275237)
- **InnoDB:** InnoDB now aborts execution on Windows by calling the `abort()` function directly, as it does on other platforms. (Bug #16263506)
- **InnoDB:** This fix removes an unnecessary debug assertion related to page_hash locks which only affects debug builds. The debug assertion is no longer valid and should have been removed when hash_lock array was introduced in MySQL 5.6. (Bug #16263167)
- **InnoDB:** Without warning, InnoDB would silently set `innodb-buffer-pool-instances` to 1 if the buffer pool size is less than 1GB. For example, if `innodb-buffer-pool-size` is set to 200M and `innodb-buffer-pool-instances` is set to 4, InnoDB would silently set `innodb-buffer-pool-instances` to 1. This fix implements a warning message and new logic for `innodb-buffer-pool-size` and `innodb-buffer-pool-instances`. (Bug #16249500, Bug #61239)
- **InnoDB:** The `lock_validate` function, which is only present in debug builds, acquired and released mutexes to avoid hogging them. This behavior introduced a window wherein changes to the hash table could occur while code traversed the same set of data. This fix updates `lock_validate` logic to collect all records for which locks must be validated, releases mutexes, and runs a loop to validate record locks. (Bug #16235056)
- **InnoDB:** `ALTER TABLE` functions would perform a check to see if InnoDB is in read-only mode (`srv_read_only_mode=true`). If InnoDB was in read-only mode, the check would return a successful status and do nothing else. This fix replaces `srv_read_only_mode` check conditions with debug assertions. (Bug #16227539)
- **InnoDB:** When the InnoDB buffer pool is almost filled with 4KB compressed pages, inserting into 16KB compact tables would cause 8KB `pages_free` to increase, which could potentially slow or stall inserts. (Bug #16223169)
- **InnoDB:** This fix updates InnoDB code in `ha_innodb.cc` and `handler0alter.cc` to use `TABLE::key_info` instead of both `TABLE::key_info` and `TABLE_SHARE::key_info`. (Bug #16215361)
- **InnoDB:** When InnoDB locking code was revised, a call to register lock waits was inadvertently removed. This fix adds the call back to the InnoDB locking code. (Bug #16208201)
- **InnoDB:** A direct call to the `trx_start_if_not_started_xa_low()` function would cause a debug assertion. (Bug #16178995)
- **InnoDB:** In the case of LOCK WAIT for an insert in a foreign key table, `InnoDB` could report a false dictionary-changed error and cause the insert to fail rather than being retried. (Bug #16174255)
- **InnoDB:** An in-place `ALTER TABLE` on an `InnoDB` table could fail to delete the statistics for the old primary key from the `mysql.innodb_index_stats` table. (Bug #16170451)
- **InnoDB:** In some cases, deadlock detection did not work, resulting in sessions hanging waiting for a lock-wait timeout. (Bug #16169638)

- **InnoDB:** When the primary key of a table includes a column prefix, and a full-text index is defined on the table, a full-text search resulted in an unnecessary warning being written to the error log. This fix suppresses the unnecessary warning. (Bug #16169411)
- **InnoDB:** `LOCK_TIME` would not be logged correctly in the slow query log. `LOCK_TIME` did not account for InnoDB row lock wait time. (Bug #16097753)
- **InnoDB:** Arithmetic underflow during page compression for `CREATE TABLE` on an `InnoDB` table could cause a server exit. (Bug #16089381)
- **InnoDB:** For debug builds, online `ALTER TABLE` operations for `InnoDB` tables could cause a server exit during table rebuilding. (Bug #16063835)
- **InnoDB:** In some cases, the `InnoDB` purge coordinator did not use all available purge threads, resulting in suboptimal purge activity. (Bug #16037372)
- **InnoDB:** `ALTER TABLE` for `InnoDB` tables was not fully atomic. (Bug #15989081)
- **InnoDB:** This fix replaces most uses of `UT_SORT_FUNCTION_BODY`, an InnoDB recursive merge sort, with the `std::sort()` function from the C++ Standard Template Library (STL). The `std::sort()` function requires less memory and is faster due to in-line execution. (Bug #15920744)
- **InnoDB:** This fix addresses unnecessary buffer pool lookups that would occur while freeing blob pages, and implements a debug status instrument, `innodb_ahi_drop_lookups`, for testing purposes. (Bug #15866009)
- **InnoDB:** This fix implements a 256-byte boundary for extending a `VARCHAR` column instead of 256-character boundary. This change allows for in-place extension of a `VARCHAR` column through an update of the data dictionary. (Bug #15863023)
- **InnoDB:** Creating numerous tables, each with a full-text search index, could result in excessive memory consumption. This bug fix adds a new configuration parameter, `innodb_ft_total_cache_size`, which defines a global memory limit for full-text search indexes. If the global limit is reached by an index operation, a force sync is triggered. (Bug #14834698, Bug #16817453)
- **InnoDB:** This fix modifies `InnoDB` code to ensure that unused thread handles are closed when the thread exits, instead of leaving thread handles open until shutdown of `mysqld` on Windows. (Bug #14762796)
- **InnoDB:** This fix removes unnecessary overhead by removing table locking and disabling read view creation and MVCC when InnoDB is started in read-only mode (`--innodb-read-only=true`). (Bug #14729365)
- **InnoDB:** A regression introduced by the fix for Bug#14100254 would result in a “!BPAGE->FILE_PAGE_WAS_FREED” assertion. (Bug #14676249)
- **InnoDB:** Full-text search (FTS) index savepoint information would not be set resulting in a severe error when attempting to rollback to the savepoint. (Bug #14639605, Bug #17456092)
- **InnoDB:** The `innodb_sync_array_size` variable was incorrectly allowed to be configured at runtime. As documented, `innodb_sync_array_size` must be configured when the MySQL instance is starting up, and cannot be changed afterward. This fix changes `innodb_sync_array_size` to a non-dynamic variable, as intended. (Bug #14629979)
- **InnoDB:** An error at the filesystem level, such as too many open files, could cause an unhandled error during an `ALTER TABLE` operation. The error could be accompanied by Valgrind warnings, and by this assertion message:

```
Assertion `! is_set()' failed.
mysqld got signal 6 ;
```

(Bug #14628410, Bug #16000909)

- **InnoDB:** The server could exit during an attempt by `InnoDB` to reorganize or compress a compressed secondary index page. (Bug #14606334)
- **InnoDB:** A DML operation performed while a `RENAME TABLE` operation waits for pending I/O operations on the tablespace to complete would result in a deadlock. (Bug #14556349)
- **InnoDB:** Attempting to unninstall the InnoDB memcached Plugin while the plugin is still installing caused the Mysql server to terminate. While the plugin deamon thread was still initializing, plugin variables were not yet set and the uninstall process could not cleanup resources. This fix adds a variable to indicate initialization status. If initialization is incomplete, the uninstall process will wait. (Bug #14279541)
- **InnoDB:** If the value of `innodb_force_recovery` was less than 6, opening a corrupted table might loop forever if a corrupted page was read when calculating statistics for the table. Information about the corrupted page was written repeatedly to the error log, possibly causing a disk space issue. The fix causes the server to halt after a fixed number of failed attempts to read the page. To troubleshoot such a corruption issue, set `innodb_force_recovery=6` and restart. (Bug #14147491, Bug #65469)
- **InnoDB:** When printing out long semaphore wait diagnostics, `sync_array_cell_print()` ran into a segmentation violation (SEGV) caused by a race condition. This fix addresses the race condition by allowing the cell to be freed while it is being printed. (Bug #13997024)
- **InnoDB:** Attempting to replace the default `InnoDB` full-text search (FTS) stopword list by creating an `InnoDB` table with the same structure as `INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD` would result in an error. `SHOW CREATE TABLE` revealed that the new `InnoDB` table was created with `CHARSET=utf8`. The `InnoDB` FTS stopword table validity check only supported latin1. This fix extends the validity check for all supported character sets. (Bug #68450, Bug #16373868)
- **InnoDB:** This fix removes left-over prototype code for `srv_parse_log_group_home_dirs`, and related header comments. (Bug #68133, Bug #16198764)
- **InnoDB:** Killing a query caused an InnoDB assertion failure when the same table (cursor) instance was used again. This is the result of a regression error introduced by the fix for Bug#14704286. The fix introduced a check to handle kill signals for long running queries but the cursor was not restored to the proper state. (Bug #68051, Bug #16088883)
- **InnoDB:** On startup, InnoDB reported a message on 64-bit Linux and 64-bit Windows systems stating that the CPU does not support crc32 instructions. On Windows, InnoDB does not use crc32 instructions even if supported by the CPU. This fix revises the wording of the message and implements a check for availability of crc32 instructions. (Bug #68035, Bug #16075806)
- **InnoDB:** The length of internally generated foreign key names was not checked. If internally generated foreign key names were over the 64 character limit, this resulted in invalid DDL from SHOW CREATE TABLE. This fix checks the length of internally generated foreign key names and reports an error message if the limit is exceeded. (Bug #44541, Bug #11753153)
- **Partitioning:** A query on a table partitioned by range and using `TO_DAYS()` as a partitioing function always included the first partition of the table when pruning. This happened regardless of the range employed in the `BETWEEN` clause of such a query. (Bug #15843818, Bug #49754)
- **Partitioning:** Partition pruning is now enabled for tables using a storage engine that provides automatic partitioning, such as the `NDB` storage engine, but which are explicitly partitioned. Previously, pruning

was disabled for all tables using such a storage engine, whether or not the tables had explicitly defined partitions.

In addition, as part of this fix, explicit partition selection is now disabled for tables using a storage engine (such as `NDB`) that provides automatic partitioning. (Bug #14827952)

References: See also: Bug #14672885.

- **Partitioning:** Execution of `ALTER TABLE ... DROP PARTITION` against a view caused the server to crash, rather than fail with an error as expected. (Bug #14653504)
- **Partitioning:** A query result was not sorted if both `DISTINCT` and `ORDER BY` were used and the underlying table was partitioned. (Bug #14058167)
- **Partitioning:** Inserting any number of rows into an `ARCHIVE` table that used more than 1000 partitions and then attempting to drop the table caused the MySQL Server to fail. (Bug #13819630, Bug #64580)
- **Replication; Linux; Microsoft Windows:** Replication failed between a Linux master using `lower_case_table_names` set to 0 and a Windows slave having `lower_case_table_names` set to 2, after a replicated table was opened on the slave; in addition, `FLUSH TABLES` was required afterwards to see which updates had actually been applied on the slave. This was because `lower_case_table_names` was checked only to see whether it was equal to 1 prior to forcing a conversion of replicated database object names to lower case for checking the table cache. Now in such cases, `lower_case_table_names` is checked to see whether it is set to a nonzero value. (Bug #16061982)
- **Replication; Microsoft Windows:** When the `binlog.index` file ended with `\r\n` (CR+LF), MySQL wrongly included the `\r` character in the name of the file it tried to open, causing replication to fail. This could cause problems with restarting the server after editing this file on a Windows system. (Bug #11757413, Bug #49455)
- **Replication:** When using GTIDs and binary log auto-positioning, the master had to scan all binary logs whenever the slave reconnected (due to reasons such as I/O thread failure or a change of master) before it could send any events to slave. Now, the master starts from the oldest binary log that contains any GTID not found on the slave. (Bug #16340322, Bug #68386)
- **Replication:** When the server version of the master was greater than or equal to 10, replication to a slave having a lower server version failed. (Bug #16237051, Bug #68187)
- **Replication:** When replicating to a MySQL 5.6 master to an older slave, Error 1193 (**`ER_UNKNOWN_SYSTEM_VARIABLE`**) was logged with a message such as `Unknown system variable 'SERVER_UUID' on master, maybe it is a *VERY OLD MASTER*`. This message has been improved to include more information, similar to this one: `Unknown system variable 'SERVER_UUID' on master. A probable cause is that the variable is not supported on the master (version: 5.5.31), even though it is on the slave (version: 5.6.11)`. (Bug #16216404, Bug #68164)
- **Replication:** The print format specifier for the `server_id` was incorrectly defined as a signed 32-bit integer with a range of -2144783647 to 2144783648. This fix changes the `server_id` integer type to an unsigned 32-bit integer type, with a range of 0 to 4294967295, which is the documented range for the `--server-id` option. (Bug #16210894)
- **Replication:** When MTS is on and transactions are being applied, the slave coordinator would hang when encountering a checksum error on a transaction event. This was due to a deadlock situation in which the coordinator assumed a normal stop while a worker waited for the coordinator to dispatch more events. For debug builds, the problem appeared as an assertion failure, which was due to the coordinator not setting `thd->is_error()` when encountering an error. (Bug #16210351)

- **Replication:** A zero-length name for a user variable (such as ``@``) was incorrectly considered to be a sign of data or network corruption when reading from the binary log. (Bug #16200555, Bug #68135)

- **Replication:** Running `SHOW RELAYLOG EVENTS` at a slave where no relay log file is present returned the following incorrect error message: "`Error when executing command SHOW BINLOG EVENTS: Could not find target log.`" The error message text has been changed to: "`Error when executing command SHOW RELAYLOG EVENTS: Could not find target log.`" (Bug #16191895)

- **Replication:** `mysqlbinlog` can connect to a remote server and read its binary logs. In MySQL 5.6 and later, this tool can also wait for the server to generate and send additional events, in practice behaving like a slave connecting to a master. In cases where the server sent a heartbeat, `mysqlbinlog` was unable to handle it properly. As a consequence, `mysqlbinlog` failed at this point, without reading any more events from the server. To fix this problem, `mysqlbinlog` now ignores any binary log events of type `HEARTBEAT_LOG_EVENT` that it receives. (Bug #16104206)

- **Replication:** `STOP SLAVE` could cause a deadlock when issued concurrently with a statement such as `SHOW STATUS` that retrieved the values for one or more of the status variables `Slave_retried_transactions`, `Slave_heartbeat_period`, `Slave_received_heartbeats`, `Slave_last_heartbeat`, or `Slave_running`. (Bug #16088188, Bug #67545)

  References: See also: Bug #16088114.

- **Replication:** Backtick (`) characters were not always handled correctly in internally generated SQL statements, which could sometimes lead to errors on the slave. (Bug #16084594, Bug #68045)

  References: This issue is a regression of: Bug #14548159, Bug #66550.

- **Replication:** In order to provision or to restore a server using GTIDs, it is possible to set `gtid_purged` to a given GTID set listing the transactions that were imported. This operation requires that the global `gtid_executed` and `gtid_purged` server system variables are empty. (This is done in order to avoid the possibility of overriding server-generated GTIDs.)

  The error message `GTID_PURGED can only be set when GTID_EXECUTED is empty` that was raised when this requirement was not met could be confusing or misleading because it did not specify the scope of the affected variables. To prevent this from happening, error messages that refer to variables relating to GTIDs now specify the scope of any such variables when they do so. (Bug #16084426, Bug #68038)

- **Replication:** The session-level value for `gtid_next` was incorrectly reset on the slave for all rollbacks, which meant that GTIDs could be lost for multi-statement transactions, causing the slave to stop with an **`ER_GTID_NEXT_TYPE_UNDEFINED_GROUP`** error. Now this is done only when a complete transaction is being rolled back, or when `autocommit` is enabled. (Bug #16084206)

- **Replication:** Dropping a table that includes non-regular ASCII characters in the table name caused a replication failure. The parser converted the table name into standard charset characters and stored the table name in the table_name variable. When the drop table query was regenerated using the table_name variable, the table name was not converted back to the original charset.

  Additionally, table and database names with 64 characters caused an assert failure. The assert required the table or database name to be less than 128 characters. Latin characters require two-bytes each, which requires an assert condition of less than or equal to 128 bits.

  The fix includes a new function to convert tables names back to the original charset, and a correction to the assert condition allowing table and database names be less than or equal to 128 bits. (Bug #16066637)

- **Replication:** Using the `--replicate-*` options (see Replication Slave Options and Variables) could in some cases lead to a memory leak on the slave. (Bug #16056813, Bug #67983)
- **Replication:** In some cases, when the slave could not recognize the server version of the master, this could cause the slave to fail. (Bug #16056365)
- **Replication:** In certain cases, the dump thread could send a heartbeat out of synchronisation with format description events. One of the effects of this issue what that, after provisioning a new server from a backup data directory and setting `--gtid-mode=ON` and enabling autopositioning (see CHANGE MASTER TO Syntax), replication failed to start, with the error `Read invalid event from master...`. The same problem could also cause GTID-based replication to fail due to skipped events following a unplanned shutdown of the master. (Bug #16051857)
- **Replication:** Replication failed when a replicated `LOAD DATA` statement inserted rows into a view. (Bug #15993712, Bug #67878)
- **Replication:** When using GTID-based replication, and whenever a transaction was executed on the master but was not sent to the slave because the slave already had a transaction with that ID, semisynchrononous replication timed out. One case in which this could happen was during a failover operation where the new master started behind the new slave. (Bug #15985893)
- **Replication:** An unnecessary flush to disk performed after every transaction when using `FILE` as the replication info repository type could degrade performance. Now this is done only when both data and relay log info is stored in (transactional) tables. (Bug #15980626)
- **Replication:** When a slave read a table map event from the binary log, it assumed that the metadata size was always less than twice the column count of the table in use, which failed when the event contained the wrong value for this field. (Bug #15830022)
- **Replication:** When reading row log events from the binary log, the slave assumed that these events were always valid; because of this, an event having an invalid binary log offset could cause the slave to crash. Now in such cases, the slave fails gracefully, and an error is reported, if any of the fields in a given row event are invalid. (Bug #15829568)
- **Replication:** Table IDs used in replication were defined as type `ulong` on the master and `uint` on the slave. In addition, the maximum value for table IDs in binary log events is 6 bytes (281474976710655). This combination of factors led to the following issues:

  - Data could be lost on the slave when a table was assigned an ID greater than `uint`.
  - Table IDs greater than 281474976710655 were written to the binary log as 281474976710655.
  - This led to a stopped slave when the slave encountered two tables having the same table ID.

  To fix these problems, IDs are now defined by both master and slave as type `ulonglong` but constrained to a range of 0 to 281474976710655, restarting from 0 when it exceeds this value. (Bug #14801955, Bug #67352)
- **Replication:** `MASTER_POS_WAIT()` could hang or return -1 due to invalid updates by the slave SQL thread when transactions were skipped by the GTID protocol. (Bug #14775893)

  References: See also: Bug #15927032.
- **Replication:** Trying to execute a Stop event on a multi-threaded slave could cause unwanted updates to the relay log, leading the slave to lose synchronization with the master. (Bug #14737388)
- **Replication:** Internal objects used for relay log information were only partially deleted before freeing their memory. (Bug #14677824)

- **Replication:** When the server starts, it checks whether semisynchronous replication has been enabled without a lock, and, if so, it takes the lock, then tests again. Disabling semisynchronous replication following the first of the these tests, but prior to the second one, could lead to a crash of the server. (Bug #14511533, Bug #66411)

- **Replication:** It was possible in certain cases—immediately after detecting an EOF in the dump thread read event loop, and before deciding whether to change to a new binary log file—for new events to be written to the binary log before this decision was made. If log rotation occurred at this time, any events that occurred following EOF detection were dropped, resulting in loss of data. Now in such cases, steps are taken to make sure that all events are processed before allowing the log rotation to take place. (Bug #13545447, Bug #67929)

  References: See also: Bug #16016886.

- **Replication:** It was possible for the `MASTER_POS_WAIT()` function to return prematurely following a `CHANGE MASTER TO` statement that updated the `RELAY_LOG_POS` or `RELAY_LOG_NAME`. This could happen because `CHANGE MASTER TO` did not update the master log position in such cases, causing `MASTER_POS_WAIT()` to read an invalid log position and to return immediately.

  To fix this problem, the master log position is flagged as invalid until the position is set to a valid value when the SQL thread reads the first event, after which it is flagged as valid. Functions such as `MASTER_POS_WAIT()` now defer any comparison with the master log position until a valid value can be obtained (that is, after the first event following the `CHANGE MASTER TO` statement has been applied). (Bug #11766010, Bug #59037)

- **Replication:** If the disk becomes full while writing to the binary log, the server hangs until space is freed up manually. It was possible after this was done for the MySQL server to fail, due to an internal status value being set when not needed. Now in such cases, rather than trying to set this status, a warning is written in the error log instead. (Bug #11753923, Bug #45449)

- **Replication:** The binary log and relay log files used the name of the PID file instead of the host name as the basename. (Bug #11753843, Bug #45359)

- **Microsoft Windows:** In Shared Memory mode, the MySQL Server could crash when receiving requests from multiple threads. (Bug #13934876)

- RPM packages were missing the `innodb_engine.so` and `libmemcached.so` plugins. (Bug #17001088)

- Windows MSI installers for MySQL 5.7 had a 5.6 upgrade code, not a 5.7 upgrade code. (Bug #16445344)

- `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` could report incorrect memory-allocation values when the correct values exceeded 4GB. (Bug #16414644)

- The server could exit if a prepared statement attempted to create a table using the name of an existing view while an SQL handler was opened. (Bug #16385711)

- Performance Schema statement tokenization overhead was reduced. (Bug #16382260)

- A long database name in a `GRANT` statement could cause the server to exit. (Bug #16372927)

- Some aggregate queries attempted to allocate excessive memory. (Bug #16343992)

- For debug builds, an assertion could be raised if a statement failed with autocommit enabled just before an `XA START` statement was issued. (Bug #16341673)

- Very small `join_buffer_size` values could cause an assertion to be raised. (Bug #16328373)

- The `BUILD-CMAKE` file in MySQL distributions was updated with the correct URL for `CMake` information. (Bug #16328024)
- The optimizer's attempt to remove redundant subquery clauses raised an assertion when executing a prepared statement with a subquery in the `ON` clause of a join in a subquery. (Bug #16318585)

  References: This issue is a regression of: Bug #15875919.
- Incorrect results were returned if a query contained a subquery in an `IN` clause which contained an `XOR` operation in the `WHERE` clause. (Bug #16311231)
- A Valgrind failure could occur if a `CREATE USER` statement was logged to the general query log and the `old_passwords` system variable was set to 2. (Bug #16300620)
- For debug builds, checking of password constraints could raise an assertion for statements that updated passwords. (Bug #16289303)
- Conversion of numeric values to `BIT` could yield unexpected results. (Bug #16271540)
- Fixed warnings when compiling with XCode 4.6. Fixed warnings when compiling when the `_XOPEN_SOURCE` or `isoctal` macro was already defined in the environment. (Bug #16265300, Bug #60911, Bug #12407384)
- In the range optimizer, an index merge failure could cause a server exit. (Bug #16241773)
- For upgrade operations, RPM packages produced unnecessary errors about being unable to access `.err` files. (Bug #16235828)
- Queries using range predicates that were evaluated using the LooseScan semi-join strategy could return duplicate rows. (Bug #16221623)

  References: This issue is a regression of: Bug #14728469.
- Certain legal `HAVING` clauses were rejected as invalid. (Bug #16221433)
- yaSSL did not perform proper padding checks, but instead examined only the last byte of cleartext and used it to determine how many bytes to remove. (Bug #16218104)
- The Performance Schema could return incorrect values for the `PROCESSLIST_INFO` column of the `threads` table. (Bug #16215165)
- `mysql_config --libs` displayed incorrect output. (Bug #16200717)
- Invocation of the range optimizer for a `NULL` select caused the server to exit. (Bug #16192219)
- For debug builds, the server could exit due to incorrect calculation of applicable indexes for a join that involved `const` tables. (Bug #16165832)
- For a `CREATE TABLE (... col_name TIMESTAMP DEFAULT CURRENT_TIMESTAMP ...) ... SELECT` statement for which the `SELECT` did not provide a value for the `TIMESTAMP` column, that column was set to '0000-00-00 00:00:00', not the current timestamp. (Bug #16163936)
- Using `GROUP BY WITH ROLLUP` in a prepared statement could cause the server to exit. (Bug #16163596)
- With the thread pool plugin enabled, large numbers of connections could lead to a Valgrind panic or failure of clients to be able to connect. (Bug #16088658, Bug #16196591)
- Performance Schema instrumentation was missing for slave worker threads. (Bug #16083949)

- The server executed `EXPLAIN FORMAT=JSON` for some malformed queries improperly. (Bug #16078557)
- If the error for a failed `CACHE INDEX` statement index within a stored program was processed by a condition handler, a malformed packet and “Command out of sync” error occurred. (Bug #16076180)
- Setting the `slave_rows_search_algorithms` system variable to an inappropriate value could cause the server to exit. (Bug #16074161)
- `SET PASSWORD` and `GRANT ... IDENTIFIED BY` have no effect on the password of a user who is authenticated using an authentication plugin that accesses passwords stored externally to the mysql.user table. But attempts to change the password of such a user produced no warning, leading to the impression that the password had been changed when it was not. Now MySQL issues an `ER_SET_PASSWORD_AUTH_PLUGIN` warning to indicate that the attempt was ignored. (Bug #16072004)
- Directory name manipulation could result in stack overflow on Mac OS X and Windows. (Bug #16066243)
- References to the unused `SIGNAL_WITH_VIO_SHUTDOWN` macro in the `CMake` files were removed. (Bug #16066150)
- The initial `test` database contained a `dummy.bak` file that prevented `DROP DATABASE` from working. This file is no longer included. Also, a `db.opt` file is now included that contains these lines:

  ```
  default-character-set=latin1
  default-collation=latin1_swedish_ci
  ```

  (Bug #16062056)
- Issuing a `PREPARE` statement using certain combinations of stored functions and user variables caused the server to exit. (Bug #16056537)
- Setting a system variable to `DEFAULT` could cause the server to exit. (Bug #16044655)
- For debug builds, if the server was started with binary logging disabled, executing `SHOW RELAYLOG EVENTS` from within a stored procedure raised an assertion. (Bug #16043173)
- The query parser leaked memory for some syntax errors. (Bug #16040022)
- During shutdown, the server could attempt to lock an uninitialized mutex. (Bug #16016493)
- The `--default-authentication-plugin` option permitted invalid plugin values, and did not always set the `old_passwords` system variable to a value appropriate for the named plugin. (Bug #16014394)
- Instances of `#ifdef WITH_MYISAMMRG_STORAGE_ENGINE` and `#ifdef WITH_CSV_STORAGE_ENGINE` in the server source code were removed because the `CSV` and `MERGE` storage engine plugins are mandatory. (Bug #15997345)
- The `--character-set-server` option could set connection character set system variables to values such as `ucs2` that are not permitted. (Bug #15985752)
- For debug builds, executing a statement within a trigger or stored function that caused an implicit commit raised an assertion. (Bug #15985318)
- Under some circumstances, `mysql --secure-auth` permitted passwords to be sent to the server using the old (pre-4.1) hashing format. (Bug #15977433)
- A `mysys` library string-formatting routine could mishandle width specifiers. (Bug #15960005)

- Table creation operations added entries to the `file_instances` Performance Schema table, but these were not always removed for table drop operations. (Bug #15927620)
- With index condition pushdown enabled, queries for which the pushed-down condition contained no columns in the used index could be slow. (Bug #15896009)
- A query with an `EXISTS/IN/ALL/ANY` subquery with an `ORDER BY` clause ordering by an outer column of type `BLOB` that is not in the select list caused an assertion to fire. (Bug #15875919)

  References: See also: Bug #14728142.
- In special cases, the optimizer did not consider indexes that were applicable to query processing, resulting in potentially suboptimal execution and incorrect `EXPLAIN` output. (Bug #15849135, Bug #16094171)
- Queries in the query cache involving a given table were incorrectly invalidated if a `TEMPORARY` table of the same name was dropped. (Bug #14839743)
- The optimizer could return nonmatching records for queries that used `ref` access on string data types. (Bug #14826522)

  References: See also: Bug #14682735.
- Failure of `CREATE SERVER` due to a missing or read-only `mysql.servers` table resulted in a memory leak. (Bug #14781478)
- Table names can be up to 64 characters, but the message string for the `ER_TABLE_NEEDS_REBUILD` and `ER_TABLE_NEEDS_UPGRADE` errors were truncating names longer than 32 characters. (Bug #14753226)
- Enabling the query cache during high client contention could cause the server to exit. (Bug #14727815)
- Enabling the slow query log at runtime when access permissions on the log file prevented the server from writing to it caused the server to exit. (Bug #14711347)
- If the optimizer calculated a row count of zero for the inner table of an outer join, it could not determine proper ordering for the following tables. (Bug #14628746)
- The server sometimes failed to respect `MAX_CONNECTIONS_PER_HOUR` limits on user connections. (Bug #14627287)
- The server could access the `DEBUG_SYNC` facility while closing temporary tables during connection shutdown, after the facility had been cleaned up, leading to an assertion being raised. (Bug #14626800)
- The optimizer could return incorrect results after transforming an `IN` subquery with aggregate functions to an `EXISTS` subquery. (Bug #14586710)
- Table removal could fail and cause the server to exit for very long file names. (Bug #14581920)
- When a client program loses the connection to the MySQL server or if the server begins a shutdown after the client has executed `mysql_stmt_prepare()`, the next `mysql_stmt_prepare()` returns an error (as expected) but subsequent `mysql_stmt_execute()` calls crash the client. (Bug #14553380)
- Previously, if multiple `--login-path` options were given, `mysql_config_editor` ignored all but the last one. Now multiple `--login-path` options result in an error. (Bug #14551712)
- If MySQL server was started with options to enable the general query log or slow query log, but access permissions on the log file prevented the server from writing to it, the server started with an error message indicating that logging was off and that the server needed to be restarted after the problem was corrected. This was incorrect because it is also possible to set the logging variables again at runtime

(without a restart) after correcting the problem. The error message now indicates this possibility. (Bug #14512467)

- For debug builds, creating a `TEMPORARY` table inside a trigger caused the server to exit. (Bug #14493938)
- `SHOW COLUMNS` on a view defined as a `UNION` of `Geometry` columns could cause the server to exit. (Bug #14362617)
- The `sha256_password_private_key_path` and `sha256_password_public_key_path` system variables indicate key files for the `sha256_password` authentication plugin, but the server failed to properly check whether the key files were valid. Now in the event that either key file is invalid, the server logs an error and exits. (Bug #14360513)
- `SET var_name = VALUES(col_name)` could cause the server to exit. This syntax is now prohibited because in `SET` context there is no column name and the statement returns `ER_BAD_FIELD_ERROR`. (Bug #14211565)
- The `COM_CHANGE_USER` command in the client/server protocol did not properly use the character set number in the command packet, leading to incorrect character set conversion of other values in the packet. (Bug #14163155)
- If the server was started with `--skip-grant-tables`, the `CREATE EVENT` and `ALTER EVENT` statements resulted in a memory leak. (Bug #14059662)
- Invoking the `FORMAT()` function with a locale and a very large number could cause the server to exit. (Bug #14040155)
- For debug builds, improper handling for `AUTO_INCREMENT` value overflow could cause the server to exit. (Bug #13875572)
- Certain plugin-related conditions can make a user account unusable:
  - The account requires an authentication plugin that is not loaded.
  - The account requires the `sha256_password` authentication plugin but the server was started with neither SSL nor RSA enabled as required by this plugin.

  The server now checks those conditions by default and produces warnings for unusable accounts. This checking slows down server initialization and `FLUSH PRIVILEGES`, so it is made optional by means of the new `validate_user_plugins` system variable. This variable is enabled by default, but if you do not require the additional checking, you can disable it at startup to avoid the performance decrement. (Bug #13010061, Bug #14506305)
- Passing an unknown time zone specification to `CONVERT_TZ()` resulted in a memory leak. (Bug #12347040)
- The obsolete `linuxthreads.txt` and `glibc-2.2.5.patch` files in the `Docs` directory of MySQL distributions have been removed. (Bug #11766326)
- The server could exit if built to permit a maximum number of indexes per table larger than 64.

  In the course of fixing this problem, a `-DMAX_INDEXES=N` `CMake` option was added to permit building the server to support a larger maximum number of indexes per table. The default is 64. The maximum is 255. Values smaller than 64 are ignored and the default of 64 is used. (Bug #11761614)
- `mysql_install_db` did not escape `'_'` in the host name for statements written to the grant tables. (Bug #11746817)

- With `explicit_defaults_for_timestamp` enabled, inserting `NULL` into a `TIMESTAMP NOT NULL` column now produces an error (as it already did for other `NOT NULL` data types), instead of inserting the current timestamp. (Bug #68472, Bug #16394472)
- Handling of `SQL_CALC_FOUND_ROWS` in combination with `ORDER BY` and `LIMIT` could lead to incorrect results for `FOUND_ROWS()`. (Bug #68458, Bug #16383173)
- If `INET6_NTOA()` or `INET6_ATON()` returned `NULL` for a row in a result set, following rows also returned `NULL`. (Bug #68454, Bug #16373973)
- A statement with an aggregated, nongrouped outer query and an aggregated, nongrouped subquery in the `SELECT` list could return incorrect results. (Bug #68372, Bug #16325175)
- Adding an `ORDER BY` clause following an `IN` subquery could cause duplicate rows to be returned. (Bug #68330, Bug #16308085)
- If the server was started with `--skip-grant-tables`, `ALTER USER ... PASSWORD EXPIRE` caused the server to exit. (Bug #68300, Bug #16295905)
- Configuring with `-DWITH_SSL=/path/to/openssl` resulted in link errors due to selection of the incorrect `libcrypto`. (Bug #68277, Bug #16284051)
- If `mysql` is built with the bundled `libedit` library, the library is built as static code, to avoid linking to a different dynamic version at runtime. Dynamic linking could result in use of a different, incompatible version and a segmentation fault. (Bug #68231, Bug #16296509)
- Some table I/O performed by the server when calling a storage engine were missing from the statistics collected by the Performance Schema. (Bug #68180, Bug #16222630)
- The Perl version of `mysql_install_db` mishandled some error messages. (Bug #68118, Bug #16197542)
- `mysql_install_db` did not work in Solaris 10 sparse root zones. (Bug #68117, Bug #16197860)
- For arguments with fractional seconds greater than six decimals, `SEC_TO_TIME()` truncated, rather than rounding as it should have. (Bug #68061, Bug #16093024)
- Queries with many values in a `IN()` clause were slow due to inclusion of debugging code in non-debugging builds. (Bug #68046, Bug #16078212)

  References: See also: Bug #58731, Bug #11765737.
- `ALTER TABLE *tbl_name* ADD COLUMN col_name TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP` inserted `0000-00-00 00:00:00` rather than the current timestamp if the alteration was done in place rather than by making a table copy. (Bug #68040, Bug #16076089)
- `mysqld_safe` used the nonportable `-e` test construct. (Bug #67976, Bug #16046140)
- The server did not enforce the `port` or `report_port` upper limit of 65,535 and truncated larger values. (Bug #67956, Bug #16035522)
- Nonspatial indexes only support exact-match lookups for spatial columns, but the optimizer incorrectly used `range` access in some cases, leading to incorrect results. (Bug #67889, Bug #15993693)
- For `EXPLAIN DELETE` and `EXPLAIN UPDATE` the `possible_keys` column listed all indexes, not just the applicable indexes. (Bug #67830, Bug #15972078)
- `SLEEP()` produced no warning or error for `NULL` or negative arguments. Now it produces a warning, or an error in strict SQL mode. (Bug #67548, Bug #15859462)

- Attempts to create a trigger for which a trigger with the same action time and event already existed resulted in an `ER_NOT_SUPPORTED_YET` error rather than an `ER_TRG_ALREADY_EXISTS` error. (Bug #67357, Bug #14801721)
- If a table had rows in the `INFORMATION_SCHEMA.INNODB_CMP_PER_INDEX` table, dropping the table did not remove those rows. (Bug #67283, Bug #14779330)
- MySQL failed to build if configured with `WITH_LIBWRAP` enabled. (Bug #67018, Bug #16342793)
- If one thread was rebuilding a result for the query cache, other threads in the middle of using the previous result could fail to discard the old result properly. For debug builds, this raised an assertion. (Bug #66781, Bug #14631798)
- `CMake` did not check whether the system `zlib` had certain functions required for MySQL, resulting in build errors. Now it checks and falls back to the bundled `zlib` if the functions are missing. (Bug #65856, Bug #14300733)
- If a dump file contained a view with one character set and collation defined on a view with a different character set and collation, attempts to restore the dump file failed with an “illegal mix of collations” error. (Bug #65382, Bug #14117025)
- The `SQL_NO_CACHE` keyword is supposed to prevent the server from checking the query cache to see whether the query result is already cached, and to prevent it from caching the query result. However, the query cache check was suppressed only if `SQL_NO_CACHE` was preceded and followed by space characters. (For example, the server checked the cache if the keyword was followed by a newline.) Now the parser requires that the preceding and following characters be whitespace characters, not spaces. (Bug #64164, Bug #13641256)
- If the server was started without a `--datadir` option, `SHOW VARIABLES` could show an empty value for the `datadir` system variable. (Bug #60995, Bug #12546953)
- When a view definition contained a special character in the `SEPARATOR` clause of the `GROUP_CONCAT()` aggregate function, `mysqldump` created an invalid view definition that produced an error when the dump file was reloaded. (Bug #60920, Bug #12395512)
- For debug builds, some queries with `SELECT ... FROM DUAL` nested subqueries raised an assertion. (Bug #60305, Bug #11827369)
- The `--log-slow-admin-statements` and `--log-slow-slave-statements` command options now are exposed at runtime as the `log_slow_admin_statements` and `log_slow_slave_statements` system variables. Their values can be examined using `SHOW VARIABLES`. The variables are dynamic, so their values can be set at runtime. (The options were actually *replaced* by the system variables, but as system variables can be set at server startup, no option functionality is lost.) (Bug #59860, Bug #11766693)
- Source code in the `mysys` library for the `my_malloc_lock` and `my_free_lock` memory-locking APIs was never used and has been removed. (Bug #54662, Bug #11762107)
- If the server failed to read `errmsg.sys`, it could exit with a segmentation fault. (Bug #53393, Bug #11760944)
- `UNION ALL` on `BLOB` columns could produce incorrect results. (Bug #50136, Bug #11758009)
- An out-of-memory condition could occur while handling an out-of-memory error, leading to recursion in error handling. (Bug #49514, Bug #11757464)
- The `REPLACE()` function produced incorrect results when a user variable was supplied as an argument and the operation was performed on multiple rows. (Bug #49271, Bug #11757250)

- `UNION` type conversion could incorrectly turn unsigned values into signed values. (Bug #49003, Bug #11757005)
- If XA support was activated by multiple storage engines, the server would exit. (Bug #47134, Bug #11755370)
- Use of `KILL` to kill a statement in another session could in some cases cause that session to return an incorrect error code. (Bug #45679, Bug #11754124)
- Setting `max_connections` to a value less than the current number of open connections caused the server to exit. (Bug #44100, Bug #11752803)
- The optimizer used loose index scan for some queries for which this access method is inapplicable. (Bug #42785, Bug #11751794)
- View access in low memory conditions could raise a debugging assertion. (Bug #39307, Bug #11749556)
- The output for `SHOW CREATE VIEW` could vary depending on the `DEFINER` account privileges. (Bug #34553, Bug #11747931)
- On Windows, the `log_error` system variable did not accurately reflect the error log file name in some cases. For example, if the server was started without `--console` or `--log-error`, the default is to log to *`host_name`*`.err` in the data directory, but `log_error` remained blank.

  Now `log_error` should be nonblank and reflect the log file name in all cases, on all platforms. The value is `stderr` if the server does not write error messages to a file and sends them to the console (standard error output) instead. In particular, on Windows, `--console` overrides use of an error log and sends error messages to the console, so `log_error` will be set to `stderr`. (Bug #8307, Bug #11745049)
- If a column is declared as `NOT NULL`, it is not permitted to insert `NULL` into the column or update it to `NULL`. However, this constraint was enforced even if there was a `BEFORE INSERT` (or `BEFORE UPDATE` trigger) that set the column to a non-`NULL` value. Now the constraint is checked at the end of the statement, per the SQL standard. (Bug #6295, Bug #11744964)

Changes in MySQL 5.7.0 (Not released, Milestone 10)

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.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- **Important Change; Replication:** `SHOW SLAVE STATUS` when run concurrently with `STOP SLAVE` can take a long time to execute if the slave SQL thread was in the midst of applying a large update. To fix this problem, a new `NONBLOCKING` option has been added to the `SHOW SLAVE STATUS` statement. When this option is used, `SHOW SLAVE STATUS` does not wait on the SQL or I/O threads but returns immediately. This means that the reported states of these threads may not be completely up to date when the option is used. `NONBLOCKING` is intended primarily for use by monitoring tools in which

obtaining an immediate response is more important than having the most timely data. (Bug #15993588, Bug #67879)

- **Important Change; Replication:** Added the `--idempotent` option for `mysqlbinlog`, which causes the MySQL Server to employ idempotent mode. This causes suppression of all duplicate-key and key-not-found errors when processing updates from the binary log. The mode is in effect for the current `mysqlbinlog` client and client session only.

- **Important Change:** `INSERT DELAYED` is no longer supported. The server recognizes but ignores the `DELAYED` keyword, handles the insert as a nondelayed insert, and generates an `ER_WARN_LEGACY_SYNTAX_CONVERTED` warning. ("INSERT DELAYED is no longer supported. The statement was converted to INSERT."). Similarly, `REPLACE DELAYED` is handled as a nondelayed replace. The `DELAYED` keyword will be removed in a future release.

  In addition, several `DELAYED`-related options or features were removed:

  - The `--delayed-insert` option for `mysqldump`.
  - The `COUNT_WRITE_DELAYED`, `SUM_TIMER_WRITE_DELAYED`, `MIN_TIMER_WRITE_DELAYED`, `AVG_TIMER_WRITE_DELAYED`, and `MAX_TIMER_WRITE_DELAYED` columns of the `table_lock_waits_summary_by_table` Performance Schema 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 into the `performance_schema` database.
  - `mysqlbinlog` no longer writes comments mentioning `INSERT DELAYED`.

- **Microsoft Windows:** Windows Vista, Windows Server 2008, and newer support native symlinking using the `mklink` command. This makes the MySQL Server implementation of database symbolic links using `.sym` files redundant, so that mechanism is now removed. This change has the following implications:

  - Existing `.sym` files are now ignored. Database symlinks should be recreated using `mklink`. See Using Symbolic Links for Databases on Windows.
  - The `--symbolic-links` and `--skip-symbolic-links` options and the `have_symlink` system variable now are meaningful only for Unix systems, and not for Windows.

- Previously, **Control+C** in `mysql` interrupted the current statement if there was one, or exited `mysql` if not. Now **Control+C** interrupts the current statement if there was one, or cancels any partial input line otherwise, but does not exit. (Bug #66583, Bug #14554568)

- The server now issues a warning if an index is created that duplicates an existing index, or an error in strict SQL mode. (Bug #37520, Bug #11748842)

- MySQL now supports stacked diagnostics areas. When a push to the diagnostics area stack occurs, the first (current) diagnostics area becomes the second (stacked) diagnostics area and a new current diagnostics area is created as a copy of it. Within a condition handler, executed statements modify the new current diagnostics area, but `GET STACKED DIAGNOSTICS` can be used to inspect the stacked diagnostics area to obtain information about the condition that caused the handler to activate, independent of current conditions within the handler itself. (Previously, there was a single diagnostics area. To inspect handler-activating conditions within a handler, it was necessary to check this diagnostics area before executing any statements that could change it.) See GET DIAGNOSTICS Syntax, and The MySQL Diagnostics Area.

- The `mysql_clear_password` cleartext client-side authentication plugin is intended for authentication schemes that require the server to receive the password as entered on the client side, without hashing.

Because the password is sent in the clear, this plugin should be used within the context of a secure connection, such as an SSL connection, to avoid exposing the password over the network. To make inadvertent use of this plugin less likely, it is now required that clients explicitly enable it. This can be done several ways:

- Set the `LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN` environment variable to a value that begins with `1`, `Y`, or `y`. This enables the plugin for all client connections.
- The `mysql`, `mysqladmin`, and `mysqlslap` client programs support an `--enable-cleartext-plugin` option that enables the plugin on a per-invocation basis.
- The `mysql_options()` C API function supports a `MYSQL_ENABLE_CLEARTEXT_PLUGIN` option that enables the plugin on a per-connection basis. Also, any program that uses `libmysqlclient` and reads option files can enable the plugin by including an `enable-cleartext-plugin` option in an option group read by the client library.

Bugs Fixed

- **Important Change; Replication:** Statements involving the Performance Schema tables should not be written to the binary log, because the content of these tables is applicable only to a given MySQL Server instance, and may differ greatly between different servers in a replication topology. The database administrator should be able to configure (`INSERT`, `UPDATE`, or `DELETE`) or flush (`TRUNCATE TABLE`) performance schema tables on a single server without affecting others. However, when using replication with GTIDs enabled (see Replication with Global Transaction Identifiers), warnings about unsafe statements updating Performance Schema tables were elevated to errors, preventing the use of `performance_schema` and GTIDs together.

  Similar problems were encountered with replication and system logging tables when GTIDs were enabled.

  This fix introduces the concept of a *nonreplicated* or *local table*. Now when MySQL replication encounters a table that is marked as local, updates to this table are ignored.

  This fix defines as local the following tables, which are no longer replicated:

  - All tables in the `performance_schema` database
  - `mysql.general_log`
  - `mysql.slow_log`
  - `mysql.slave_relay_log_info`
  - `mysql.slave_master_info`
  - `mysql.slave_worker_info`

  Before this fix, statements using the `performance_schema` and other tables just listed were handled by being marked as unsafe for replication, which caused warnings during execution; the statements were nonetheless written to the binary log, regardless of the logging format in effect.

  Existing replication behavior for tables in the `INFORMATION_SCHEMA` database is not changed by this fix.

  For more information, see MySQL Performance Schema. See also MySQL Server Logs, and Slave Status Logs. For information about general and slow query log tables, see Selecting General Query and Slow Query Log Output Destinations. (Bug #14741537)

- **Important Change:** Formerly, the `ExtractValue()` and `UpdateXML()` functions supported a maximum length of 127 characters for XPath expressions supplied to them as arguments. This limitation has now been removed. (Bug #13007062, Bug #62429)

- **InnoDB:** Creating and altering tables repeatedly would result in a memory leak that was due to a duplicate key error. The duplicate key error occurred because the `row_merge_build_indexes` function did not call `row_fts_psort_info_destroy` often enough. As full-text search indexes were created with a unique index, the unique index failed due to the duplicate key error, and full-text search build resource would not be released. (Bug #14759111)

- **InnoDB:** During an online DDL operation, a duplicate key error could be incorrectly issued if a record was inserted and subsequently updated while the table was being rebuilt. (Bug #14723456)

- **InnoDB:** InnoDB IO threads within Performance Schema were exposed with the following name: "`io_handler_thread`". This fix implements specific keys such as `io_read_handler_thread`, `io_write_handler_thread`, `io_ibuf_handler_thread` to differentiate InnoDB IO threads within Performance Schema. (Bug #14670810)

- **InnoDB:** If the server crashed at a precise moment during an `ALTER TABLE` operation that rebuilt the clustered index for an `InnoDB` table, the original table could be inaccessible afterward. An example of such an operation is `ALTER TABLE ... ADD PRIMARY KEY` The fix preserves the original table if the server halts during this operation. You might still need to rename the `.ibd` file manually to restore the original table contents: in MySQL 5.6 and higher, rename from `#sql-ib$`*`new_table_id`*`.ibd` to *`table_name`*`.ibd` within the database directory; prior to MySQL 5.6, the temporary file to rename is *`table_name`*`#1` or `#2`. (Bug #14669848)

- **InnoDB:** Inserting data of varying record lengths into an `InnoDB` table that used compression could cause the server to halt with an error. (Bug #14554000, Bug #13523839, Bug #63815, Bug #12845774, Bug #61456, Bug #12595091, Bug #61208)

- **InnoDB:** This fix addresses an assert condition that would occur when inserting large BLOBs into tablespaces with a 4KB physical page size or into some compressed tables. Extents would not be allocated soon enough for tablespaces with smaller physical page sizes. (Bug #14520559)

- **InnoDB:** If a table was defined with an index key length very close to the upper length limit of 3072, a query against that table could cause a serious error. (Bug #14500557, Bug #14537695)

- **InnoDB:** In debug builds, a mismatch in the `InnoDB` `PAGE_FREE` list would cause an assertion. (Bug #12701488)

- **InnoDB:** On Linux systems, certain I/O requests that read or wrote fewer than the requested number of bytes could cause the server to crash. This issue could happen more frequently with asynchronous I/O requests. The messages did not clearly identify what type of error occurred:

  ```
  InnoDB: Operating system error number 0 in a file operation.
  InnoDB: Error number 0 means 'Success'.
  ```

  With this fix, MySQL retries the operation several times before giving up. (The number of retries is defined by the constant `NUM_RETRIES_ON_PARTIAL_IO` in the source code, default value 10.) (Bug #11761646, Bug #54160)

- **Partitioning:** When the server is started with `--skip-partition`, it should reject DDL or DML statements on partitioned tables. However, for `DROP TABLE`, the server dropped the `.frm` file, and for `RENAME TABLE`, the server renamed the `.frm` file. (Bug #11763795)

- **Replication:** `mysqlbinlog` did not properly decode `DECIMAL` values in a row-based binary log. This could cause invalid values to be printed out for `DECIMAL` columns. (Bug #14309019)

References: See also: Bug #17544169.

- **Replication:** `mysqlbinlog -v -v` prints in verbose mode, with comments on data column types, from a binary log file. When `mysqlbinlog -v -v` encountered a column data value which was `NULL`, the column's data type was not updated; as a result, the data type of the previous column was printed instead, or—in the case where this was the table's first column—the type was shown as `<an integer>`. Now in such cases, the data type is shown correctly. (Bug #14171756)

- **Replication:** When using `mysqlbinlog` with the `--verbose` option to read a binary log written by a MySQL server using row-based or mixed-format logging, invalid SQL could be produced when comments appeared inside `BINLOG` statements. One way in which this could happen was when a function that updated data was used within an `INSERT ... SELECT` statement. (Bug #12889121)

- **Replication:** `mysql_upgrade` on the master broke replication when the slave was run with `--log-output` equal to `FILE` or `NONE`. (Bug #11763447)

- **Replication:** Issuing `STOP SLAVE` caused a spurious `Error reading packet from server: Lost connection to MySQL server during query` message to be written to the error log. (Bug #11761457, Bug #12977988, Bug #53955)

- **Replication:** When an error occurs in the slave SQL thread, this causes the `Slave_SQL_Error` and `Slave_SQL_Errno` columns from `SHOW SLAVE STATUS` to display the reason for the error. The error number should be one of the usual constants `ER_*` defined in `sql/share/errmsg.txt`, and the error message should be the corresponding string. However, in some cases, `Slave_SQL_Errno` was set to something other than an `ER_*` number, and `Slave_SQL_Error` to a hard-coded error message rather than a translatable string from `sql/share/errmsg.txt`. Now all errors shown by `SHOW SLAVE STATUS` originate in `sql/share/errmsg.txt`, as expected. (Bug #11760365, Bug #52768)

- **Microsoft Windows:** On Microsoft Windows, `CMake` entries for POSIX API's not found on Microsoft Windows were added to the CMake cache. This decreases the number of expected "Not found" errors while compiling MySQL. (Bug #14790333)

- **Microsoft Windows:** On Microsoft Windows, queries referring to a table with invalid characters would search the system for invalid file names. The generated system error code (`ERROR_INVALID_NAME`) was not recognized by MySQL, so this unknown error would be reported to the server log as "ERRNO: 22 - INVALID ARGUMENT". MySQL now recognizes these errors and reports them as the table does not existing, and it no longer logs them to the server error log. (Bug #14642248)

- **Microsoft Windows:** On Windows, starting the server with `--log-error` and `--console` caused the server to write to the log file but not the console. Before MySQL 5.5.3, this occurred only if `--log-error` was specified after `--console`. Now, `--console` overrides `--log-error` no matter the option order so that `--console` produces console output in all cases. (Bug #14207773, Bug #65592)

- **Microsoft Windows:** It was possible to specify a Named Pipe that was already in use. This is no longer allowed, as an error is now emitted and the process is aborted. After `mysqld.exe` was started in Named Pipe mode with a pipe name that was already used by a different instance, neither instance was able to shut down properly when a shutdown command was received from a `TCP` socket in any of the processes. Therefore, `mysqld.exe` was not terminated. (Bug #13891058, Bug #61885)

- **Microsoft Windows:** On Microsoft Windows, a failed API or function call in `mysqld.exe` could sometimes report the error code 22, instead of the proper error code. (Bug #11763004)

- **Cluster Replication:** Transactions originating on a replication master are applied on slaves as if using `AO_AbortError`, but transactions replayed from a binary log were not. Now transactions being replayed from a log are handled in the same way as those coming from a “live” replication master.

  See The NdbOperation::AbortOption Type, for more information. (Bug #14615095)

- Joins of exactly 32 tables and containing a `HAVING` clause returned an empty result. (Bug #15972635)
- The parser rejected some legal `UNION` statements. (Bug #14730856)
- Setting `thread_cache_size` to a negative value at server startup resulted in a value of 16384 rather than 0. (Bug #14683107)
- `XA RECOVER` displayed nonprintable characters in the XID data. Now such characters are hex encoded. (Bug #14670465)
- There was no warning at startup if the server was started with an invalid `query_cache_size` value. (Bug #14576423)
- The return value from `IS_USED_LOCK()` was reported using the wrong data type. (Bug #14575699)
- When resolving outer fields, `Item_field::fix_outer_fields()` creates new `Item_ref`s for each execution of a prepared statement, so these must be allocated in the runtime memroot. The memroot switching before resolving `JOIN::having` caused these to be allocated in the statement root, leaking memory for each prepared statement execution. (Bug #14409015)
- Activation of a stored program handler did not preserve the current diagnostics stack. (Bug #14342913)
- In debug builds, killing a `HELP` statement caused an assertion to be raised. (Bug #14221840)
- If an error occurred during evaluation of the `BEFORE` expression of a `PURGE BINARY LOGS BEFORE` statement, the statement did not abort as it should have and later raised an assertion. (Bug #14215847)
- For the index merge access method, the optimizer could make a suboptimal choice of indexes to use. (Bug #14095506)
- An assertion could be raised if the attempt to open the `.frm` file for a temporary table failed. (Bug #13359247)
- If the state of the Event Scheduler was changed during server shutdown, the server could crash. (Bug #13002460)
- When storing the definition for a view that used the `UPPER()` or `LOWER()` function, the function call was replaced by `UCASE()` or `LCASE()`, respectively (as shown in the output of `SHOW CREATE VIEW`). This was in spite of the fact that `UPPER()` and `LOWER()` are standard, with `UCASE()` and `LCASE()` being MySQL synonyms for these. This made it more difficult to move databases between MySQL and other database systems.

  With this fix, calls to `UPPER()` and `LOWER()` within views are no longer rewritten when storing their definitions; instead, `UCASE()` is now rewritten as `UPPER()` in stored view definitions, and `LCASE()` as `LOWER()`, which increases the portability of the views. (Bug #12844279)
- For queries that accessed an `INFORMATION_SCHEMA` table in a subquery, an attempt to lock a mutex that had already been locked could cause a server crash. (Bug #11765744)
- A view using INTERVAL() could be created, but it was not possible to select from the view, nor could it be shown with SHOW CREATE VIEW. (Bug #11753832)
- The `Range checked for each record` optimization is now used for conditions with outer query references. (Bug #11750963)
- For an `ALTER TABLE` statement that renamed or changed the default value of a `BINARY` column, the alteration was done using a table copy and not in place. (Bug #67141, Bug #14735373, Bug #69580, Bug #17024290)

- For queries using `ref` access on string data types, the `ref` access condition could be evaluated again as part of the query condition or pushed down as an index condition to the storage engine. (Bug #66983, Bug #14682735)
- Concurrent execution of `DROP DATABASE` and any of `CREATE FUNCTION`, `CREATE PROCEDURE`, or `CREATE EVENT` could be written to the binary log in the wrong order, causing replication failure. (Bug #65428, Bug #14127220)
- For a view defined on a `UNION`, the server could create an invalid view definition. (Bug #65388, Bug #14117018, Bug #72018, Bug #18405221)
- If `read_only` is enabled, it is still permitted to create `TEMPORARY` tables. But in this case, a non-`TEMPORARY` table with the same name could also be created, which should not be permitted. (Bug #64992, Bug #13969578)
- Enabling the session value of `low_priority_updates` had no effect for `INSERT` statements. (Bug #64892, Bug #13939940)
- References to a stored function without a database name qualifier while there was no default database resulted in an `ER_SP_DOES_NOT_EXIST` error rather than `ER_NO_DB_ERROR`. (Bug #64692, Bug #13864485)
- The server refused client connections while executing `FLUSH PRIVILEGES`. (Bug #63178, Bug #13418619)
- A view was created with an incorrect definition if the `WHERE` clause contained string literals and `character_set_client` and `character_set_connection` were set to different character sets. (Bug #63094, Bug #13520710)
- `SHOW CREATE VIEW` failed if the tables underlying the view were changed. (Bug #61718, Bug #12762393)
- Concurrent inserts were blocked by selects if the inserts were generated from within a stored procedure. (Bug #58689, Bug #11765698)
- An `INSERT INTO ... SELECT` statement that inserted no rows unnecessarily invalidated statements in the query cache that used the target table. (Bug #50065, Bug #11757947)
- Using `ALTER TABLE` to rename a table to `.` resulted in a table with no name. (Bug #49636, Bug #11757569)
- `SHOW GLOBAL STATUS` caused performance problems on busy servers due to lock contention. (Bug #42930, Bug #11751904)
- `INSERT INTO ... SELECT ... ON DUPLICATE KEY UPDATE` and `LOAD DATA CONCURRENT REPLACE` took too weak a lock, leading to the possibility of concurrent `SELECT` statements returning inconsistent results. (Bug #38046, Bug #11749055)
- An event was not dropped from the `mysql.event` table under these circumstances: The event was created while the event scheduler was enabled; the scheduler was disabled and re-enabled; the event expiration time was reached. (Bug #34804, Bug #11748012)