MySQL 5.6 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 5.6, up through MySQL 5.6.21. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 5.6 documentation, see the MySQL 5.6 Reference Manual, which includes an overview of features added in MySQL 5.6 (What Is New in MySQL 5.6), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.5 to MySQL 5.6 (Upgrading from MySQL 5.5 to 5.6).

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/), it means that the version has not yet been released.

The date mentioned with a release version is the date of the last revision control system changeset on which the release was based, not necessarily the date when the distribution packages were made available. The binaries are usually made available a few days after the date of the tagged changeset because building and testing all packages takes some time.

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.

Document generated on: 2014-08-01 (revision: 3848)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 5.6, up through MySQL 5.6.21.

Legal Notices

For more information on the terms of this license, or for details on how the MySQL documentation is built and produced, please visit MySQL Contact & Questions.

For additional licensing information, including licenses for third-party libraries used by MySQL products, see Preface and 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.

Changes in MySQL 5.6.21 (Not yet released)

InnoDB Notes

- The `--skip-innodb` option is now deprecated 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).

Changes in MySQL 5.6.20 (2014-07-31)

Security Notes

- **Security Fix:** The linked OpenSSL library for the MySQL 5.6 Commercial Server has been updated from version 1.0.1g to version 1.0.1h. Versions of OpenSSL prior to and including 1.0.1g are reported to be vulnerable to CVE-2014-0224.

  This change does not affect the Oracle-produced MySQL Community build of MySQL Server 5.6, which uses the `yaSSL` library instead. (CVE-2014-0224)

InnoDB Notes

- **Important Change:** 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.

  As a result of the redo log `BLOB` write limit introduced for MySQL 5.6, `innodb_log_file_size` should be set to a value greater than 10 times the largest `BLOB` data size found in the rows of your tables plus the length of other variable length fields (VARCHAR, VARBINARY, and TEXT type fields). Failing to do so could result in “Row size too large” errors. No action is required if your innodb_log_file_size setting is already sufficiently large or your tables contain no BLOB data. (Bug #16963396, Bug #19030353, Bug #69477)

Functionality Added or Changed

- **Replication:** The new system variable binlog_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 binlog_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)

- CMake support was updated to handle CMake version 3. (Bug #19001781)

- 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)

- 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)

- By default, mysql_install_db creates a my.cnf file in the installation base directory using a template. This may be undesireable for some deployments. To enable this behavior to be suppressed, mysql_install_db now supports a --keep-my-cnf option to preserve any existing my.cnf file and not create a new my.cnf file. (Bug #71600, Bug #18205019)

- The mysqlhotcopy utility is now deprecated and will be removed in a future version of MySQL. Among the reasons for this: It works only for the MyISAM and ARCHIVE storage engines; it works on Unix but not Windows. Alternatives include mysqldump and MySQL Enterprise Backup.

- The timed_mutexes system variable has no effect and is deprecated.

Bugs Fixed

- **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)

- **InnoDB:** Opening a parent table that has thousands of child tables could result in a long semaphore wait condition. (Bug #18806829)

- **InnoDB:** For single item full-text searches, deleted documents were included in inverse document frequency (IDF) calculations. (Bug #18711306, Bug #72548)

- **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:** A DELETE operation on a table with full-text search indexes raised an assertion. (Bug #18683832)

  References: See also Bug #14639605.

- **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 would fail to set CONN_DATA->CRSR_TRX to NULL, resulting in a serious error. (Bug #18652854)

- **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:** 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:** 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 fix for Bug#16418661 added superfluous buf_flush_list() logic to InnoDB startup code. (Bug #17798076, Bug #70899)

- **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:** 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)

- **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:** 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 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:** 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:** Quotation marks were not always handled correctly by LOAD DATA INFILE when written into the binary log. (Bug #18207212, Bug #71603)

- **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:** 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 #17283409, Bug #69954)

- **Replication:** On Windows, mysqldump failed if the error log file was deleted (missing) from the active MySQL server. (Bug #17076131)
- **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)
- 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 bug is a regression of Bug #11760197.
- 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)
- Compiler flags were not passed to DTrace, causing problems for 32-bit builds cross-compiled on 64-bit platforms. (Bug #18593044)
- 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)
- 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)
- 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)
- mysqladmin password masked the old password given on the command line, but not the new password. (Bug #18163964)
- 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)
- For full-text queries on InnoDB tables, attempts to access deleted document IDs could lead to a server exit. (Bug #18079671)
- MyISAM temporary files could be used to mount a code-execution attack. (Bug #18045646)
- For queries that selected from the Performance Schema events_statements_current table, adding an ORDER BY clause could produce incorrect results. (Bug #17729044)

- 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)

- Clients could determine based on connection error message content whether an account existed. (Bug #16513435, Bug #17357528)

- 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)

- Use of a nonmultibyte algorithm for skipping leading spaces in multibyte strings could cause a server exit. (Bug #12368495, Bug #18315770)

- For a view defined on a `UNION`, the server could create an invalid view definition. (Bug #65388, Bug #14117018, Bug #72018, Bug #18405221)

- Configuring with `cmake -DWITHOUT_SERVER` to build clients without the server failed for builds outside of the source tree. (Bug #66000, Bug #14367046)

- 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.

- Deadlock could occur if three threads simultaneously performed `INSTALL PLUGIN`, `SHOW VARIABLES`, and `mysql_change_user()`. (Bug #71236, Bug #18008907, Bug #72870, Bug #18903155)

- `mysql_config_editor` exited when given an empty argument to the `--login-path` option. (Bug #71837, Bug #18311024, Bug #18830493)

- MySQL did not compile with Bison 3. A workaround is to downgrade to Bison 2. (Bug #71250, Bug #18017820, Bug #18978946)

- 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)

- 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)

- Client auto-reconnect did not work for clients linked against `libmysqlclient`, even with `MYSQL_OPT_RECONNECT` enabled. (Bug #70026, Bug #17309863)

- Upgrades using RPM packages could change the ownership of an installation directory. (Bug #71715, Bug #18281535)

- Proxy users were unable to execute statements if the proxied user password had expired. (Bug #71337, Bug #18057562)

- 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)

- 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)

Changes in MySQL 5.6.19 (2014-05-30)

There is no MySQL Community Server 5.6.18. That version number was used for an out-of-schedule release of the Enterprise Edition to address the OpenSSL “Heartbleed” issue. This issue did not affect the Community Edition because it uses yaSSL, not OpenSSL, so a new release of the Community Server was not needed, and 5.6.17 is followed by 5.6.19.

Functionality Added or Changed

- The obsolete and unmaintained `charset2html` utility has been removed from MySQL distributions. (Bug #71897, Bug #18352347)
- The `mysqlbug`, `mysql_waitpid`, and `mysql_zap` utilities have been deprecated and will be removed in MySQL 5.7.

Bugs Fixed

- **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:** 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:** 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)
- **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:** Enabling the `InnoDB` Table Monitor would result in a `ib_table->stat_initialized` assertion failure. (Bug #17039528, Bug #69641)
- **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)
- **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 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 #18364070)

References: See also Bug #18236612.

- **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:** 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:** The server did not always handle the auto.cnf file correctly in cases where this file's permissions were incorrect. (Bug #17786581, Bug #70891)
- **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:** 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)
- The server could fail to properly reprepare triggers that referred to another table after that table was truncated. (Bug #18596756, Bug #72446, Bug #18665853)
- Certain INFORMATION_SCHEMA queries could cause a server exit. (Bug #18319790)
- 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)
- Solaris-specific scripts were included in and installed by non-Solaris packages. (Bug #18305641)
- innobase_strnxfrm() wrote one byte too many. (Bug #18277082)
- Concurrent execution of a FLUSH TABLE operation and a stored program that used a cursor could cause a server exit. (Bug #18158639)
- 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)
- 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)
- 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)

- `mysqldump` could create table definitions in the dump file that resulted in `Too many columns` errors when reloading the dump file. (Bug #17477959)
- 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)
- The optimizer trace could cause a server exit in cases where a subquery was transformed away. (Bug #17458054)
- The Debug Sync facility could lose a signal, leading to a spurious `ER_DEBUG_SYNC_TIMEOUT` error. (Bug #14765080, Bug #18221750)
- Compilation problems were fixed for errors reported by `Clang` and `gcc` when compiling in C++11 mode. (Bug #66803, Bug #14631159)
- `CMake` produced not-useful warnings about `INTERFACE_LINK_LIBRARIES` policy. (Bug #71089, Bug #17905155, Bug #17894997)
- `LOAD DATA LOCAL INFILE` could use all CPU if import errors occurred when there were no line delimiters. (Bug #51840, Bug #11759519)
- On Windows, `REPAIR TABLE` and `OPTIMIZE TABLE` failed for `MyISAM` tables with `.MYD` files larger than 4GB. (Bug #69683, Bug #17235179)
- 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)
- Updates could fail to update all applicable rows in cases where multiple key values were identical except for trailing spaces. (Bug #69684, Bug #17156940)

Changes in MySQL 5.6.18 (2014-04-11)

A known limitation of this release:

> **Note**
>
> If you have `InnoDB` tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

Heartbleed Bug

- **Security Fix:** The MySQL 5.6 Commercial Server has been updated to use `OpenSSL` version 1.0.1g, which has been publicly reported as not vulnerable to CVE-2014-0160. Please see Oracle Note #1645479.1 for further details.

  The MySQL 5.6 Community Server is built using the `yaSSL` cryptographic software library instead of `OpenSSL`. Oracle-produced MySQL 5.6 Community Server binaries use `YaSSL` libraries which have been reported as not affected by CVE-2014-0160. Users of MySQL Server binaries produced by parties other than Oracle should seek a vulnerability assessment from their respective binary providers.

  Since the only change in MySQL Server 5.6.18 is the inclusion of `OpenSSL` libraries publicly reported as unaffected by CVE-2014-0160, and since Oracle-produced MySQL Community builds do not include `OpenSSL` libraries known to be affected by CVE-2014-0160, Oracle is not producing builds for MySQL Community Server for version 5.6.18. This means that MySQL Community Server is skipping version 5.6.18. (Bug #18533200, CVE-2014-0160)

Bugs Fixed

- Executing a correlated subquery on an `ARCHIVE` table which has an `AUTO_INCREMENT` column caused the server to hang. (Bug #18065452)

Changes in MySQL 5.6.17 (2014-03-27)

A known limitation of this release:

> **Note**
>
> If you have `InnoDB` tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

Functionality Added or Changed

- **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.

- **Incompatible Change:** The `ERROR_FOR_DIVISION_BY_ZERO`, `NO_ZERO_DATE`, and `NO_ZERO_IN_DATE` SQL modes now are deprecated and setting the `sql_mode` value to include any of them generates a warning. In MySQL 5.7, these modes do nothing. Instead, their effects are included in the effects of strict SQL mode (`STRICT_ALL_TABLES` or `STRICT_TRANS_TABLES`). The motivation for the change in MySQL 5.7 is to reduce the number of SQL modes with an effect dependent on strict mode and make them part of strict mode itself.

  To make advance preparation for an upgrade to MySQL 5.7, see SQL Mode Changes in MySQL 5.7. That discussion provides guidelines to assess whether your applications will be affected by the SQL mode changes in MySQL 5.7.

- **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)

- On Solaris, mysql_config --libs now includes -R*/path/to/library* so that libraries can be found at runtime. (Bug #18235669)

- mysql_install_db provides a more informative diagnostic message when required Perl modules are missing. (Bug #69844, Bug #18187451)

- The IGNORE clause for ALTER TABLE is now deprecated and will be removed in a future version of MySQL. ALTER IGNORE TABLE causes problems for replication, prevents online ALTER TABLE for unique index creation, and causes problems with foreign keys (rows removed in the parent table).

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. 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:** Running a SELECT on a partitioned table caused a memory access violation in memcpy(). (Bug #18383840)

  References: See also Bug #18167648.

- **InnoDB:** An invalid memmove in fts_query_fetch_document would cause a serious error. (Bug #18229433)

- **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:** 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:** InnoDB would fail 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:** 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:** 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)

- **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:** A full-text tokenizer thread would terminate with an incorrect error message. (Bug #18021306)

- **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:** 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:** 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:** 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:** 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` would fail to start when `innodb_data_file_path` specified the data file size in kilobytes by appending `K` to the size value. (Bug #16287752)

- **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)

- **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.

- **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:** 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:** SHOW SLAVE STATUS used incorrect values when reporting MASTER_SSL_CRL and MASTER_SSL_CRLPATH. (Bug #17772911, Bug #70866)

  References: This bug was introduced by Bug #11747191.
- **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:** A number of possible state messages used as values for the PROCESSLIST_STATE column of the Performance Schema threads table were longer than the width of the column (64 characters).

  The long state messages are now silently truncated in order to avoid errors. This fix applies in MySQL 5.6 only; a permanent fix for the issue is made in MySQL 5.7 and later. (Bug #17319380)
- **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:** 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:** The server checks to determine whether semisynchronous replication has been enabled without a lock, and if this is the case, it takes the lock and checks again. If semisynchronous replication was disabled after the first but prior to the second one, this could cause the server to fail. (Bug #14511533, Bug #66411)

  References: See also Bug #17920923.

- **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)

- mysql_install_db could hang while reading /dev/random to generate a random root password. (Bug #18395378)

- Compilation failed if MySQL was configured with CFLAGS set to include a -Werror option with an argument. (Bug #18173037)

- 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)

- While printing the server version, the mysql client did not check for buffer overflow in a string variable. (Bug #18186103)

- Building MySQL from source on Windows using Visual Studio 2008 would fail with an identifier not found error due to a regression introduced by the patch for Bug#16249481. (Bug #18057449)

- 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, a table handle was unnecessarily instrumented. (Bug #18047865)

- 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)
- The SUM_SORT_MERGE_PASSES column value in the Performance Schema events_statements_summary_by_digest table was calculated incorrectly. (Bug #17938255)
- If the Performance Schema events_statements_summary_by_digest 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 optimizer could push down a condition when the index did not have the key part present in the condition. (Bug #17814492)
- 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)
- DROP TRIGGER succeeded even with the read_only system variable enabled. (Bug #17503460)
- 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)
- When run by root, mysqld --help --verbose exited with a nonzero error code after displaying the help message. (Bug #70058, Bug #17324415)
- 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)
- 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)
- A deadlock error occurring during subquery execution could cause an assertion to be raised. (Bug #69969, Bug #17307201)
- 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)
- Messages written by the server to the error log for LDML collation definition problems were missing the collation name. (Bug #68144, Bug #16204175)
- Aggregating the results of a subquery in the FROM clause could produce incorrect results. (Bug #71244, Bug #18014565)
- 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)
- 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)
- On Windows, mysql_install_db.pl could be run only from within the bin directory under the installation directory. (Bug #42421, Bug #11751526)

- The `msql2mysql`, `mysql_convert_table_format`, `mysql_find_rows`, `mysql_fix_extensions`, `mysql_setpermission`, and `mysqlaccess` utilities are now deprecated and will be removed in MySQL 5.7. (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 5.6.16 (2014-01-31)

Known limitations of this release:

Note

Building MySQL from source on Windows using Visual Studio 2008 fails with an `identifier not found` error. Later versions of Visual Studio are unaffected. The workaround is to set the `CMake` build option, `INNODB_PAGE_ATOMIC_REF_COUNT`, to `OFF`. This option is `ON` by default.

Note

If you have `InnoDB` tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

Functionality Added or Changed

- **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. Enabling and disabling monitors for periodic output by creating and dropping specially named tables is deprecated and may be removed in a future release. For additional information, see `InnoDB` Monitors.

- 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)

- `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)

Bugs Fixed

- **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:** Using the InnoDB memcached plugin (see InnoDB Integration with memcached) with innodb_api_enable_binlog set to 1 caused the server to leak memory. (Bug #70757, Bug #17675622)
- **InnoDB:** A boolean mode full-text search query would result in a memory access violation during parsing. (Bug #17978763)
- **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:** Due to a parser error, full-text search queries that include a sub-expression could return the wrong result. (Bug #17840768)
- **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:** InnoDB would report an incorrect operating system error code after failing to initialize. (Bug #17788055, Bug #70867)
- **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:** 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:** 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:** In debug builds, a merge insert buffer during a page read would cause a memory access violation. (Bug #17561188)
- **InnoDB:** Truncating a memcached InnoDB table while memcached is performing DML operations would result in a serious error. (Bug #17468031)
- **InnoDB:** In sync0rw.ic, rw_lock_x_lock_func_nowait would needlessly call os_thread_get_curr_id. (Bug #17509710, Bug #70417)
- **InnoDB:** Attempting to rename a table to a missing database would result in a serious error. (Bug #17447500)
- **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:** 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 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)

- **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:** The ALTER TABLE INPLACE algorithm would fail 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:** The function os_file_get_status would not work with raw devices. (Bug #17023438, Bug #69424)

- **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:** Renaming a column while also adding or dropping columns in the same ALTER TABLE operation would cause an error. (Bug #16864981)

- **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:** On Windows, the full-text search (FTS) object ID was not in the expected hexadecimal format. (Bug #16559254)

  References: See also Bug #16559119.

- **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:** Table renaming errors would appear in the LATEST FOREIGN KEY ERROR section of the SHOW ENGINE INNODB STATUS output. (Bug #12762390, Bug #61746)

- **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 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:** Creating and dropping large numbers of temporary tables could lead to increased memory consumption. (Bug #17806014)

- **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:** 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 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 #17544169)

  References: See also Bug #14309019.

- **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:** 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)

- 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)

- With the compressed client/server protocol enabled, Performance Schema statement instrumentation could raise an assertion. (Bug #17794846)

- In some cases, UNIX_TIMESTAMP() could return NULL when it should return 0. (Bug #17728371)

- An assertion could be raised if a filesort failed to resize its main buffer when record properties changed. (Bug #17757914)

- 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)

- Using the mysqldump --set-gtid-purged option with no value caused mysqldump to crash. (Bug #17650245)

- A race condition between Performance Schema statement event threads led to a server exit. (Bug #17637970)
- In a view definition requireing resolution of aggregrate expressions within a subquery to an outer query, selecting from the view could cause a server exit. (Bug #17547804)

  References: This bug is a regression of Bug #16436383.
- An addressing error in accessing the join buffer could produce invalid results or a server exit. (Bug #17513341)
- `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)
- 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 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)
- 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)
- Several `-W` warning flags were turned off for compilation in maintainer mode if MySQL was configured with `-DWITH_INNODB_MEMCACHED=1`. (Bug #13898319)
- 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)
- For the `utf8_bin` collation, `ORDER BY LOWER(`*`col_name`*`)` could produce incorrect ordering. (Bug #69005, Bug #16691598)
- Several issues identified by the Coverity static analysis tool were fixed. Thanks to Honza Horak for the patch. (Bug #70830, Bug #17760511)
- On Windows, the `--local-service` server option did not work, and was not displayed in the `--help` message. (Bug #69637, Bug #17049656)
- 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)

- 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)
- 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)
- Complex updates of Performance Schema tables involving joins or subqueries failed to update every row. (Bug #70025, Bug #17309657)
- 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)
- COUNT(DISTINCT) sometimes produced an incorrect result when the last read row contained a `NULL` value. (Bug #68749, Bug #16539979, Bug #71028, Bug #17867117)
- `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)
- 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)
- 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)
- Some scripts displayed out-of-date information regarding where to report bugs. (Bug #68742, Bug #16530527)
- Some files in the Performance Schema `file_instances` table were not being removed because the file-removal operation was not instrumented. (Bug #69782, Bug #17209750)
- `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)
- 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)
- `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)

Changes in MySQL 5.6.15 (2013-12-03)

A known limitation of this release:

Note

If you have `InnoDB` tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

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/.

Functionality Added or Changed

- **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)

- 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)

- A new `CMake` option, `WITH_ASAN`, permits enabling AddressSanitizer for compilers that support it. (Bug #17435338)

- The hash function used for metadata locking was modified to reduce overhead. (Bug #68487, Bug #16396598)

Bugs Fixed

- **InnoDB; Replication:** The InnoDB mecached plugin would update a record before inserting to the binlog, which would cause slave server replication to stop. The insert should occur before the update. (Bug #17358875)
- **InnoDB:** A regression introduced by the fix for Bug#17371537 resulted a memory leak for memcached insert operations. (Bug #17738935)
- **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:** 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:** 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:** 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:** 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:** CHECK TABLE would ignore the QUICK option. (Bug #17513737)
- **InnoDB:** The information_schema.innodb_metrics index_merge counter was not incremented in btr0btr.cc. This patch also introduces new counters (index_page_reorg_attempts, index_page_reorg_successful and index_page_discards) and renames the index_merges counter to “index_page_merge_attempts” to distinguish it from the index_page_merge_successful counter. (Bug #17409657, Bug #70241)
- **InnoDB:** Setting the O_DIRECT flag on a file on tmpfs on some operating systems would result in an error printed to the error log. Creating multiple temporary tables on tmpfs would cause the error to be printed repeatedly. The error message has been changed to a warning that is only printed once when running CREATE TABLE multiple times. (Bug #17441867)
- **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 would fail 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:** 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:** Full-text index creation on a large table would fail due to insufficient temporary table space and result in a misleading “incorrect key file” error. (Bug #17339606)

- **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:** 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:** `trx_create` and `trx_free` would be called for every `memcached` `get` request. (Bug #17371537, Bug #70172)

- **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:** In `btr_validate_level` there are checks to ensure that all B-tree pages are marked when allocated. The checks would fail on the change buffer because the allocation of change buffer pages is handled differently than other B-tree pages. (Bug #16884217)

- **InnoDB:** When the change buffer is enabled, `InnoDB` would fail 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:** 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:** 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:** Due to a regression in MySQL 5.6, creating or dropping tables with `innodb_force_recovery` set to `3` (`SRV_FORCE_NO_TRX_UNDO`) would fail. 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:** 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:** 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:** 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:** A replication master did not handle correctly the disabling of the semisync plugin on the master and the slave, with a subsequent stopping of the slave. (Bug #17460821, Bug #70349)
- **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:** 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:** 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:** Trying to set CHANGE MASTER TO ... MASTER_AUTO_POSITION = 0 failed with error 1777 (**ER_AUTO_POSITION_REQUIRES_GTID_MODE_ON**). (Bug #17277744)
- **Replication:** The value of LAST_INSERT_ID() was not correctly replicated when filtering rules were used on the slave. (Bug #17234370, Bug #69861)
- **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:** 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)

- `libmysqlclient` version 18 files were removed from `MySQL-shared-compat` RPM packages to avoid a conflict between the `MySQL-shared` and `MySQL-shared-compat` RPM packages. (Bug #17749617)
- Enabling Index Merge optimizer switches and setting a small `sort_buffer_size` value could lead to a server exit. (Bug #17617945)
- 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)
- Some license and documentation files were missing from Windows MSI packages. (Bug #17584523)
- Selecting from the Performance Schema `session_connect_attrs` table under high load could cause a server exit. (Bug #17542370)
- Using MySQL Installer to install MySQL on Windows, then using Uninstall a Program in the Control Panel to uninstall MySQL, resulted in the MySQL service entry not being removed. (Bug #17550741)
- Compilation failures under Visual Studio 2012 were corrected. (Bug #17430236)
- 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)
- 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)
- 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)
- The `filesort` implementation sometimes failed to allocate enough buffer space, leading to a server exit. (Bug #17326567)
- `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)
- An internal `InnoDB` string routine could write past the end of a buffer. (Bug #16765410)
- For uninstall operations, the MySQL Server MSI installer failed to remove some files if the server was running. This could lead to problems for a subsequent installation of a lower MySQL version: The installer reported a successful operation but not some of the files from the original installation were not replaced by their lower-versioned counterparts. (Bug #16685125)
- 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)
- 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)
- For upgrades using the Windows MSI package installer, the upgrade dialog message was missing the “from” version. (Bug #16053094)
- Very long database names in queries could cause the server to exit. (Bug #15912213, Bug #16900358)

- 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)

- Recursion by functions called as arguments to XPath expressions was not handled correctly, sometimes causing such expressions to fail. (Bug #14040071)

- 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)

- Some possible cases of memory use after being freed were fixed. Thanks to Jan Staněk for the patch. (Bug #68918, Bug #16725945)

- In debug builds, static initialization code could call DBUG functions before the DBUG subsystem was initialized. (Bug #69653, Bug #17063675)

- With the thread pool plugin enabled, the PROCESSLIST_USER and PROCESSLIST_HOST columns of the Performance Schema threads 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)

- COUNT(DISTINCT) should not count NULL values, but they were counted when the optimizer used Loose Index Scan. (Bug #69841, Bug #17222452)

- 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)

- 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)

- 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)

- Performance Schema instrumentation overhead was reduced for frequent connect/disconnect operations. (Bug #70018, Bug #17310878)

- Setting host_cache_size at startup had no effect. (Bug #70552, Bug #17576516)

- InnoDB full-text searches failed to find records within transactions that included savepoints. (Bug #70333, Bug #17458835)

- 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)

- 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)

- 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)

- 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)

- 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)

- Missing va_end() calls were added to logging and UCS2 code. Thanks to Jan Staněk for the patch. (Bug #68896, Bug #16725769)

- MySQL did not compile on Mac OS X 10.9 (Mavericks). (Bug #70542, Bug #17647863)

- 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 5.6.14 (2013-09-20)

A known limitation of this release:

Note

If you have InnoDB tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

Audit Log Plugin Notes

- MySQL 5.7 changed audit log file output to a new format that has better compatibility with Oracle Audit Vault. This format has been backported to MySQL 5.6 and it is 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 OLD). For details about each format, see The Audit Log File.

  In addition, 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.

  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.

  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

- **Incompatible Change:** In MySQL 5.6.13, the statement/com/ abstract statement instrument in the Performance Schema setup_instruments table was renamed to statement/com/new_packet. That change has been reverted.

Applications that refer to the old instrument name must be updated with the new name. For more information about the use of abstract instruments in statement classification, see Performance Schema Statement Event Tables. (Bug #16750433, Bug #17271055)

- **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)

Bugs Fixed

- **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)
- **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:** 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:** 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:** An assertion would be raised in fil_node_open_file due to a missing .ibd file. Instead of asserting, InnoDB should return false and the caller of fil_node_open_file should handle the return message. (Bug #17305626, Bug #70007)
- **InnoDB:** The assertion ut_ad(oldest_lsn <= cur_lsn) in file buf0flu.cc would fail 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:** 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:** 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:** 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:** When dropping all indexes on a column with multiple indexes, InnoDB would fail to block a DROP INDEX operation when a foreign key constraint requires an index. (Bug #16896810)

- **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:** 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)

- **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:** For the Barracuda file format and beyond, the externally stored prefix would be read even though the prefix is already stored locally in memory. (Bug #16569640)

- **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:** 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:** 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 would fail to check if the server is in a shutdown state before sleeping. (Bug #13417564, Bug #63276)

- **InnoDB:** An infinite loop could occur in buf_page_get_gen when handling compressed-only pages. (Bug #12560151, Bug #61132)

- **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)

- **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:** 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 binlog 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:** 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:** 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:** 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:** 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 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:** 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)
- The mysql_real_connect() C API function could leak memory if it failed. (Bug #17337684)
- 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 bug is a regression of Bug #14188793.
- 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)
- SELECT * from performance_schema.events_statements_current could raise an assertion due to a race condition under load. (Bug #17164720)
- 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)
- 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)
- Under load, truncating the accounts Performance Schema table could cause a server exit. (Bug #17084615)
- 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)
- 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)
- 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)
- The my_strtoll10() function could incorrectly convert some long string-format numbers to numeric values and fail to set the overflow flag. (Bug #16997513)
- 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)
- 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.
- For some statements, memory leaks could result when the optimizer removed unneeded subquery clauses. (Bug #16807641)

  References: This bug is a regression of Bug #15875919.
- Password rewriting in the general query log now also applies to prepared statements. (Bug #16732621)
- Within a stored procedure, repeated execution of a prepared CREATE TABLE statement for a table with partitions could cause a server exit. (Bug #16614004)
- For debug builds, when the optimizer removed an Item_ref pointing to a subquery, it caused a server exit. (Bug #16509874)

  References: This bug is a regression of Bug #16318585.
- 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)
- 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)

- Metadata returned for a prepared SELECT statement that had outer joins could indicate that columns containing NULL values were NOT NULL. (Bug #12818811)

- 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)

- 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)

- 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 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)

- 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 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)

- Full-text search on InnoDB tables failed on searches for words containing apostrophes when using boolean operators. (Bug #69932, Bug #17276125)

- 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)

Changes in MySQL 5.6.13 (2013-07-31)

Known limitations of this release:

Note

On Microsoft Windows, MySQL Installer does not upgrade MySQL Enterprise Backup (MEB) 3.8.1 to 3.8.2 (latest version). A workaround is to uninstall MEB 3.8.1 and then install MEB 3.8.2 (latest version) with MySQL Installer.

Note

If you have InnoDB tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

Functionality Added or Changed

- **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.

  In addition, the statement/com/ abstract statement instrument was renamed to statement/com/new_packet.

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)

- **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)

- 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 now are deprecated. They can cause problems when new options are implemented for programs. A prefix that is currently unambiguous might become ambiguous in the future. If an unambiguous prefix is given, a warning now occurs to provide feedback. For example:

```
Warning: Using unique option prefix compr instead of compress is
deprecated and will be removed in a future release. Please use the
full name instead.
```

  Option prefixes are no longer supported in MySQL 5.7; only full options are accepted. (Bug #16996656)

- 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)

- 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)

Bugs Fixed

- **Performance; Important Change; InnoDB:** InnoDB would fail 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:** 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, 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)

- **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:** 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:** A full-text search using the IN BOOLEAN MODE modifier would result in an assertion failure. (Bug #16927092)

  References: This bug is a regression of Bug #16516193.

- **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:** Removed invalid compilation warning messages that appeared when compiling the InnoDB memcached plugin. (Bug #16816824)

- **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:** 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:** A memory leak would occur when inserting or replacing a row in a full-text search index on a table with more than 96 columns. (Bug #16809167)

- **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:** Some characters in the identifier for a foreign key constraint are modified during table exports. (Bug #16722314, Bug #69062)

- **InnoDB:** The page_zip_validate() consistency check would fail after compressing a page, in page_zip_compress(). This problem was caused by page_zip_decompress(), which would fail 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:** Stale InnoDB memcached connections would result in a memory leak. (Bug #16707516, Bug #68530)

- **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:** 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:** Restarting InnoDB in read-only mode and running a workload would occasionally return a global_segment < os_aio_n_segments assertion. (Bug #16362046)

- **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:** 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:** 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:** 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:** 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:** 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:** 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` would fail with the same result.

  The bug fix temporarily disables `foreign_key_checks` while the previous table definition is restored. (Bug #65701, Bug #14227431)

- **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:** The `pthread_mutex`, `commit_threads_m`, which was initialized but never used, has been removed from the code base. (Bug #60225, Bug #11829813)

- **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:** 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)

- **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 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.

- **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:** 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:** 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:** 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 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:** 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:** 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)

- 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)

- `Sql_condition::set_subclass_origin()` could perform an out-of-bounds read. (Bug #16969091)

- Initialization of `keycache_*` variables (see Multiple Key Caches) during server startup could write to incorrect memory. (Bug #16945503)

- 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)

- 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)

- The Performance Schema could spawn a thread using incorrect instrumentation information. (Bug #16939689)

- `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 bug is a regression of Bug #14789787.

- The code base was modified to account for new warning checks introduced by `gcc` 4.8. (Bug #16729109)

- 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)

- 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)
- 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)
- Some rows for a session could be missing sporadically from the Performance Schema session_connect_attrs table while the session was executing a workload. (Bug #16576980)
- The server could make the wrong decision about whether an account password was expired. (Bug #16604641)
- Upgrading from community SLES RPM packages to commercial packages for the same MySQL version failed with conflict errors. (Bug #16545296)
- A full-text search syntax error failed to print to standard output. (Bug #16429688, Bug #16765397)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- The server could exit in do_copy_not_null() due to an improper NULL-value check. (Bug #16316564)
- 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)
- The usual failed-login attempt accounting was not applied to failed COM_CHANGE_USER commands. (Bug #16241992, Bug #17357535)
- 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)
- The optimizer trace could print ranges for key parts that were not usable for range access. (Bug #14615536)

- 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)
- Passwords in statements were not obfuscated before being written to the audit log. (Bug #14536456)
- With the thread pool plugin in use, normal connection termination caused the Aborted_clients status variable to be incremented. (Bug #14081240)
- 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)
- MySQL Installer, if run in custom install or change mode, offered installation options that had no effect. (Bug #12928601)
- 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)
- 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)
- 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)
- 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)
- Unoptimized versions of the *xxx*korr() macros in my_global.h were used on 64-bit x86 processors. (Bug #61179, Bug #12565703)
- sql-common/client_plugin.c contained a nonportable use of a va_list parameter. (Bug #62769, Bug #13252623)
- 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)
- Full-text search on InnoDB tables failed on searches for words containing apostrophes. (Bug #69216, Bug #16801781)
- Full-text search on InnoDB tables failed on searches for literal phrases combined with + or - operators. (Bug #68720, Bug #16516193)
- 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)
- 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.
- 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)

- 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)

- 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)

- A typo in cmake/dtrace.cmake prevented DTrace support from being enabled by -DENABLE_DTRACE-on. (Bug #60743, Bug #12325449)

- 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)

- Some LEFT JOIN queries with GROUP BY could return incorrect results. (Bug #68897, Bug #16620047)

  References: This bug is a regression of Bug #11760517.

- For queries with ORDER BY ... LIMIT, the optimizer could choose a nonordering index for table access. (Bug #69410, Bug #16916596)

- 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 bug is a regression of Bug #15848521.

- Comparison of a DATETIME value and a string did not work correctly for the utf8_unicode_ci collation. (Bug #68795, Bug #16567381)

- 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)

Changes in MySQL 5.6.12 (2013-06-03)

Known limitations of this release:

> **Important**
>
> InnoDB may fail to open a tablespace that has multiple data files due to newly introduced corruption checking functionality. It is recommended that you do not upgrade to this version if you have more than one file for your shared InnoDB tablespace. If you have upgraded to an affected version and the server no longer starts, you can upgrade to a later version when it becomes available or downgrade to an earlier version.

> **Note**
>
> If you have InnoDB tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

Functionality Added or Changed

- 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)

Bugs Fixed

- **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)

- **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:** 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:** 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 missing comma in SHOW STATUS output would break MySQL Enterprise Monitor parsing. (Bug #16723686)

- **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 memory leak would occur in dict_check_tablespaces_and_store_max_id() when space_id is equal to zero. (Bug #16737332)

- **InnoDB:** DML operations on compressed temporary tables would result in a Valgrind error in the buffer manager stack. (Bug #16593331)

- **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:** 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:** Under certain circumstances, LRU flushing would take a long time possibly affecting all flushing activity and causing a shutdown timeout. (Bug #16500209)
- **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:** The `InnoDB` memcached `test.demo_test` table would fail to work when defined as a `utf8` charset table. (Bug #16499038)
- **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:** The `page_zip_available` function would count some fields twice. (Bug #16463505)
- **InnoDB:** This fix removes most calls to `OS_THREAD_SLEEP` from InnoDB. (Bug #16472953, Bug #68588)
- **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:** `FLUSH TABLES FOR EXPORT` would sleep too often while flushing pages from buffer pools. (Bug #16471701)
- **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:** In debug builds, an insert would fail with an invalid assertion: `sync_thread_levels_g(array, level - 1, TRUE)`. (Bug #16409715)
- **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:** 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:** `RENAME TABLE` would result in a hang due to a MySQL mutex acquisition deadlock. (Bug #16305265)
- **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:** 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:** 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 a compressed table, the “page reorganize” function would ignore the innodb_log_compressed_pages option and always log the entire compressed page, which increased the size of the redo log. The “page reorganize” function now adheres to the innodb_log_compressed_pages option and does not log compressed page images to the redo log when innodb_log_compressed_pages is set to “OFF”. (Bug #16267120)
- **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:** 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 processing read-write workloads, InnoDB would scan more pages than are required for flushing, unnecessarily consuming CPU resource. (Bug #16037180)
- **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:** 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:** 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)
- **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:** 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)
- **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:** 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:** 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 bug was introduced by Bug #16566658.
- **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:** 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:** 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 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:** 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:** 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:** 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:** 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:** 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:** 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:** 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:** `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:** 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)

- Removing a server RPM package did not shut down the existing server if it was running. (Bug #16798868)

- Overhead for setting `PROCESSLIST_STATE` values in the Performance Schema `THREADS` table has been reduced. (Bug #16633515)

- The Windows authentication plugin failed to free a context buffer for each connection. (Bug #16591288)

- The `DBUG_PRINT()` macro unnecessarily evaluated arguments when debugging was not enabled. (Bug #16556597)

- 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 server could attempt a `filesort` operation for a zero-size sort length, causing it to exit. (Bug #16503160)
- `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)
- Opening a cursor on a `SELECT` within a stored procedure could cause a segmentation fault. (Bug #16499751)

  References: This bug is a regression of Bug #14740889.
- `SET PASSWORD` treated *`user`*`@'%'` and *`user`*`@''` as referring to the same `mysql.user` table row. (Bug #16488043)
- 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)
- Some `INFORMATION_SCHEMA` queries that used `ORDER BY` did not use a `filesort` optimization as they did in MySQL 5.5. (Bug #16423536)
- 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)
- The WKB reader for spatial operations could fail and cause a server exit. (Bug #16451878)
- 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)
- Manually-created accounts (using `INSERT`) with a malformed password effectively had no password. (Bug #16414396)
- 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, `DBUG_EXPLAIN` resulted in a buffer overflow when the `debug` system variable value was more than 255 characters. (Bug #16402143)
- Within an XA transaction in ACTIVE state, statements causing an implicit commit could result in metadata locks being released too early. (Bug #16362832)
- Installing Debian packages on Ubuntu 12.10 succeeded using `dpkg`, but not with Software Center 5.4.1.4. (Bug #16387513)
- 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)
- `thread_pool_high_priority_connection` could not be set at server startup. (Bug #16310373)
- Re-execution of a stored procedure could cause a server exit in `Item_field::fix_outer_field`. (Bug #16317443)
- A `GROUP_CONCAT()` invocation containing subquery having an outer reference caused the server to exit. (Bug #16347343)
- With `secure_auth` enabled, a user with a password that used the pre-4.1 (old) hashing could not update it to use the 4.1 (new) hashing. (Bug #16304018)
- For debug builds, `GROUP_CONCAT(... ORDER BY)` within an `ORDER BY` clause could cause a server exit. (Bug #16347426)

- The validate_password plugin did not always enforce appropriate constraints against assigning empty passwords. (Bug #16346443)
- 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 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)
- If loose index scan was used on a query that used MIN(), a segmentation fault could occur. (Bug #16222245)
- If multiple statements were sent in a single request, the audit log plugin logged only the last one. Now it logs each statement separately. (Bug #16169063)
- For debug builds, an assertion was incorrectly raised for queries executed using eq_ref access and filesort. (Bug #16164885)
- An outer join between a regular table and a derived table that is implicitly groups could cause a server exit. (Bug #16177639)
- A prepared statement that used GROUP_CONCAT() and an ORDER BY clause that named multiple columns could cause the server to exit. (Bug #16075310)
- Creating a FEDERATED table without specifying a connection string caused a server exit. (Bug #16048546)
- ORDER BY MATCH ... AGAINST could cause a server exit. (Bug #16073689)
- 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)
- 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)
- In some cases, REVOKE could fail to revoke the GRANT OPTION privilege. (Bug #14799187)
- The mysql client allocated but did not free a string after reading each line in interactive mode, resulting in a memory leak. (Bug #14685362)
- 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)
- INSERT ... ON DUPLICATE KEY UPDATE on a view could cause a server exit. (Bug #14261010)
- Grouping by an outer BLOB column in a subquery caused a server exit. (Bug #13966809, Bug #14700180)
- The server could exit due to improper handling of the error from an invalid comparison. (Bug #13009341)
- The CMake check for unsigned time_t failed on all platforms. (Bug #11766815)
- 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)

- On 64-bit Mac OS X systems, CMake used x86 rather than x86_64 when determining the machine type. (Bug #58462, Bug #11765489)
- 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)
- 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)
- mysqld --help and mysqld --verbose --help performed unnecessary logging. (Bug #68578, Bug #16442113)
- 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)
- 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)
- IF() function evaluations could produce different results when executed in a prepared versus nonprepared statement. (Bug #45370, Bug #11753852)
- 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)
- 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)

- 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)

- 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)

- For debug builds, metadata locking for `CREATE TABLE ... SELECT` could raise an assertion. (Bug #68695, Bug #16503173)

- 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)

- When specified in an option file, the `plugin-dir` client option was ignored. (Bug #68800, Bug #16680313)

- 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)

- 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 optimizer could choose a poor execution plan for queries with `ORDER BY ... LIMIT`. (Bug #69013, Bug #16697792)

- `FOUND_ROWS()` could return an incorrect value if the preceding query used `filesort`. (Bug #69119, Bug #16760474)

  References: This bug is a regression of Bug #68458.

- 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)

Changes in MySQL 5.6.11 (2013-04-18)

A known limitation of this release:

Note

If you have InnoDB tables with full-text search indexes and you are upgrading from MySQL 5.6.10 to a MySQL version up to and including MySQL 5.6.18, the server will fail to start after the upgrade (Bug#72079). This bug is fixed in MySQL 5.6.19. As a workaround, remove full-text search indexes prior to upgrading and rebuild full-text search indexes after the upgrade is completed.

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 5.6.11, 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 5.6.11, it is possible to install using yum:

  ```
  shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
  ```

  For upgrades to MySQL 5.6.11, 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 5.6.*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)

Functionality Added or Changed

- **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.

- 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)

- `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` uses this option to upgrade the system tables in the `mysql` database before tables in other databases: It upgrade 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)

- The only supported value for the `innodb_mirrored_log_groups` system variable is 1, so this variable is now deprecated. Setting it to 1 at startup results in a warning. Setting it to a value other than 1 at startup results in an error and the server exits. This variable will be removed in a future release.

Bugs Fixed

- **Performance; InnoDB:** Switching the MySQL table used by the InnoDB memcached interface (using the `@@` notation), was made more efficient, by reading cached information about the cache policy to use for each table. This optimization lets you frequently switch between tables during a session that uses the memcached interface, without incurring I/O overhead from examining table metadata each time. (Bug #16206654)

- **Performance; InnoDB:** Performance was improved for operations on tables with many rows that were deleted but not yet purged. The speedup applies mainly to workloads that perform bulk deletes, or updates to the primary key columns, and where the system is busy enough to experience purge lag. (Bug #16138582, Bug #68069)

- **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)

- **Performance; InnoDB:** The I/O routines used when the AIO subsystem were made more efficient, to merge consecutive I/O requests into a single operation. This fix solves a performance issue introduced during the 5.6 development cycle. (Bug #16043841, Bug #67973)

- **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.

- **Important Change; Replication**

  > **Important**
  >
  > This fix was reverted in MySQL 5.6.12. See Changes in MySQL 5.6.12 (2013-06-03).

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 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)

- **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 would fail to find rows. (Bug #16586355)

- **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:** 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:** 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:** 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:** Crash recovery would fail with a !recv_no_log_write assertion when reading a page. (Bug #16405422)

- **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:** Improper testing of compatibility between the referencing and referenced during ALTER TABLE ... ADD FOREIGN key could cause a server exit. (Bug #16330036)

- **InnoDB:** Rollback did not include changes made to temporary tables by read-only transactions. (Bug #16310467)

- **InnoDB:** The InnoDB page-splitting algorithm could recurse excessively. (Bug #16345265)

- **InnoDB:** For debug builds, InnoDB status exporting was subject to a race condition that could cause a server exit. (Bug #16292043)

- **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:** 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:** RENAME TABLE would result in a hang due to a MySQL mutex acquisition deadlock. (Bug #16305265)

- **InnoDB:** Internal read operations could be misclassified as synchronous when they were actually asynchronous. When the I/O requests returned sooner than expected, threads could be scheduled inefficiently. This issue mainly affected read-ahead requests, and thus had relatively little impact on I/O performed by user queries. (Bug #16249505, Bug #68197)

- **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:** InnoDB now aborts execution on Windows by calling the abort() function directly, as it does on other platforms. (Bug #16263506)
- **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:** 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:** 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:** The InnoDB memcached plugin could encounter a serious error under a heavy load, such as produced by benchmark runs. (Bug #16182660, Bug #68096)
- **InnoDB:** If the MySQL server halted at a precise moment when a purge operation was being applied from the change buffer, the operation could be incorrectly performed again during the next restart. A workaround was to set the configuration option innodb_change_buffering=changes, to turn off change buffering for purge operations. (Bug #16183892, Bug #14636528)
- **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:** In some cases, deadlock detection did not work, resulting in sessions hanging waiting for a lock-wait timeout. (Bug #16169638)
- **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:** 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:** 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:** On systems that cannot handle unaligned memory access, depending on the stack frame alignment, a SIGBUS error could occur during startup. This issue was observed on Solaris 64-bit systems. (Bug #16021177)
- **InnoDB:** ALTER TABLE for InnoDB tables was not fully atomic. (Bug #15989081)
- **InnoDB:** When innodb_mirrored_log_groups was set to a value other than the default 1, the MySQL server encountered a serious error during startup while loading the InnoDB memcached

plugin. In earlier releases, the server would refuse to start (but not display an error) when this setting was changed. This fix cleans up the error handling for unsupported values of this configuration option. (Bug #15907954, Bug #67670)

- **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 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:** The server could exit during an attempt by InnoDB to reorganize or compress a compressed secondary index page. (Bug #14606334)

- **InnoDB:** A RENAME TABLE statement could stall for several minutes before timing out. This issue could occurred for a table using compression, with change buffering enabled. (Bug #14556349)

- **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:** If the server was started with the skip-innodb option, or InnoDB otherwise failed to start, query any of these Information Schema tables would cause a severe error:

  - INNODB_BUFFER_PAGE
  - INNODB_BUFFER_PAGE_LRU
  - INNODB_BUFFER_POOL_STATS

  (Bug #14144290)

- **InnoDB:** Online DDL had a restriction that prevented renaming a column and adding a foreign key involving that column in a single ALTER TABLE statement. Now, this combination of operations is allowed in a single statement. (Bug #14105491)

- **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:** The value of the innodb_version variable was not updated consistently for all server releases for the InnoDB Plugin in MySQL 5.1, and the integrated InnoDB component in MySQL 5.5, 5.6, and higher. Since InnoDB and MySQL Server development cycles are fully integrated and synchronized, now the value returned by the innodb_version variable is the same as for the version variable. (Bug #13463493, Bug #63435)

- **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)

- **InnoDB:** This fix removes left-over prototype code for `srv_parse_log_group_home_dirs`, and related header comments. (Bug #68133, Bug #16198764)

- **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)

- **Partitioning:** `ALGORITHM = INPLACE`, which was disallowed in MySQL 5.6.10 for DDL statements operating on partitioned tables, can once again be used with such statements. (Bug #16216513)

  References: See also Bug #14760210.

- **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:** 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:** 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:** 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:** 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:** `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:** 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:** 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 bug is a regression of Bug #14548159, Bug #66550.

- **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:** 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:** 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:** 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:** 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:** Internal objects used for relay log information were only partially deleted before freeing their memory. (Bug #14677824)
- **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:** 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)
- **Microsoft Windows:** In Shared Memory mode, the MySQL Server could crash when receiving requests from multiple threads. (Bug #13934876)
- Failure to handle a full-text search wildcard properly could cause the server to exit. (Bug #16446108)
- `InnoDB` now reports row and table locks to the thread pool plugin. Deadlocks within a thread group could occur otherwise. (Bug #16448639)
- `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` could report incorrect memory-allocation values when the correct values exceeded 4GB. (Bug #16414644)
- 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)
- On Linux, a race condition involving `epoll()` could cause the thread pool plugin to miss events. This was most likely on systems with greater than 16 cores. (Bug #16367483)
- 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)
- For debug builds, checking of password constraints could raise an assertion for statements that updated passwords. (Bug #16289303)
- The `BUILD-CMAKE` file in MySQL distributions was updated with the correct URL for `CMake` information. (Bug #16328024)
- 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)
- 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 bug is a regression of Bug #15875919.
- Very small `join_buffer_size` values could cause an assertion to be raised. (Bug #16328373)
- Some aggregate queries attempted to allocate excessive memory. (Bug #16343992)
- 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)
- 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)
- Conversion of numeric values to `BIT` could yield unexpected results. (Bug #16271540)
- Certain legal `HAVING` clauses were rejected as invalid. (Bug #16221433)

- 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)
- Queries using range predicates that were evaluated using the LooseScan semi-join strategy could return duplicate rows. (Bug #16221623)

  References: This bug is a regression of Bug #14728469.
- For upgrade operations, RPM packages produced unnecessary errors about being unable to access `.err` files. (Bug #16235828)
- In the range optimizer, an index merge failure could cause a server exit. (Bug #16241773)
- A full-text query using Boolean mode could return zero results in some cases where the search term was a quoted phrase:
  - If the quoted phrase was preceded by a + sign. For example, this combination of a Boolean + operator and a phrase would return zero results:

    ```
    where match(content) against('+"required term due to plus sign"' in boolean mode)
    ```
  - If the quoted phrase contained any stopwords. For example, the stopword "the" inside the phrase caused the query to return zero results:

    ```
    where match(content) against('"stopword inside the phrase"' in boolean mode)
    ```

  (Bug #16206253, Bug #68150)
- For debug builds, the server could exit due to incorrect calculation of applicable indexes for a join that involved `const` tables. (Bug #16165832)
- A bug in range optimization sometimes led to incorrect condition calculation for index merge union. This could lead to missing rows. (Bug #16164031, Bug #68194, Bug #16229746)
- The Performance Schema could return incorrect values for the `PROCESSLIST_INFO` column of the `threads` table. (Bug #16215165)
- Invocation of the range optimizer for a `NULL` select caused the server to exit. (Bug #16192219)
- `mysql_config --libs` displayed incorrect output. (Bug #16200717)
- 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)
- yaSSL did not perform proper padding checks, but instead examined only the last byte of plaintext and used it to determine how many bytes to remove. (Bug #16218104)
- Using `GROUP BY WITH ROLLUP` in a prepared statement could cause the server to exit. (Bug #16163596)
- If, in a `SELECT`, the `HAVING` clause contained a function call which itself contained an alias to a selected expression, the server could sometimes exit. (Bug #16165981)
- Setting the `slave_rows_search_algorithms` system variable to an inappropriate value could cause the server to exit. (Bug #16074161)
- Directory name manipulation could result in stack overflow on Mac OS X and Windows. (Bug #16066243)

- With statement-based binary logging, dropping a TEMPORARY InnoDB table could cause a segmentation fault. (Bug #16076275)
- 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)
- 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)
- The server executed EXPLAIN FORMAT=JSON for some malformed queries improperly. (Bug #16078557)
- Performance Schema instrumentation was missing for slave worker threads. (Bug #16083949)
- 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)
- Setting a system variable to DEFAULT could cause the server to exit. (Bug #16044655)
- 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)
- For debug builds, creating an InnoDB table in strict SQL mode that violated the maximum key length limit caused the server to exit. (Bug #16035659)
- Issuing a PREPARE statement using certain combinations of stored functions and user variables caused the server to exit. (Bug #16056537)
- Joins of exactly 32 tables and containing a HAVING clause returned an empty result. (Bug #15972635)
- A mysys library string-formatting routine could mishandle width specifiers. (Bug #15960005)
- The --character-set-server option could set connection character set system variables to values such as ucs2 that are not permitted. (Bug #15985752)
- 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)
- Under some circumstances, mysql --secure-auth permitted passwords to be sent to the server using the old (pre-4.1) hashing format. (Bug #15977433)
- 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)
- With index condition pushdown enabled, queries for which the pushed-down condition contained no columns in the used index could be slow. (Bug #15896009)

- Table creation operations added entries to the Performance Schema file_instances table, but these were not always removed for table drop operations. (Bug #15927620)
- 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)
- 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.
- Creating an InnoDB table with a FULLTEXT index could encounter a serious error if the table name contained nonalphanumeric characters. (Bug #14835178, Bug #16036699)
- Enabling the query cache during high client contention could cause the server to exit. (Bug #14727815)
- The MSI Installer installed MySQL in “per-user” mode, which could result in conflicts or failure to detect an existing installation if two users installed MySQL on the same machine. Now the MSI Installer uses “per-machine” installation mode. (Bug #14711808)
- The server sometimes failed to respect MAX_CONNECTIONS_PER_HOUR limits on user connections. (Bug #14627287)
- The optimizer could return incorrect results after transforming an IN subquery with aggregate functions to an EXISTS subquery. (Bug #14586710)
- 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)
- SET PASSWORD for anonymous users did not work correctly. (Bug #14561102)
- 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)
- Invoking the FORMAT() function with a locale and a very large number could cause the server to exit. (Bug #14040155)
- yaSSL rejected some valid server SSL certificates. (Bug #13777928)
- 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)

- mysql_install_db did not escape '_' in the host name for statements written to the grant tables. (Bug #11746817)

- mysqld_safe used the nonportable -e test construct. (Bug #67976, Bug #16046140)

- 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 optimizer used loose index scan for some queries for which this access method is inapplicable. (Bug #42785, Bug #11751794)

- 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 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)

- UNION ALL on BLOB columns could produce incorrect results. (Bug #50136, Bug #11758009)

- View access in low memory conditions could raise a debugging assertion. (Bug #39307, Bug #11749556)

- 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.

- Setting max_connections to a value less than the current number of open connections caused the server to exit. (Bug #44100, Bug #11752803)

- 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)

- For debug builds, some queries with SELECT ... FROM DUAL nested subqueries raised an assertion. (Bug #60305, Bug #11827369)

- 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)

- 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)

- 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)

- For arguments with fractional seconds greater than six decimals, SEC_TO_TIME() truncated, rather than rounding as it should have. (Bug #68061, Bug #16093024)

- MySQL failed to build if configured with WITH_LIBWRAP enabled. (Bug #67018, Bug #16342793)

- 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)

- Configuring with -DWITH_SSL=*/path/to/openssl* resulted in link errors due to selection of the incorrect libcrypto. (Bug #68277, Bug #16284051)

- 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)

- If the server was started with --skip-grant-tables, ALTER USER ... PASSWORD EXPIRE caused the server to exit. (Bug #68300, Bug #16295905)

- 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)

- mysql_install_db did not work in Solaris 10 sparse root zones. (Bug #68117, Bug #16197860)

- For EXPLAIN DELETE and EXPLAIN UPDATE the possible_keys column listed all indexes, not just the applicable indexes. (Bug #67830, Bug #15972078)

- The Perl version of mysql_install_db mishandled some error messages. (Bug #68118, Bug #16197542)

- 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)

- Adding an ORDER BY clause following an IN subquery could cause duplicate rows to be returned. (Bug #68330, Bug #16308085)

- 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)

- 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)

Changes in MySQL 5.6.10 (2013-02-05, General Availability)

Beginning with MySQL 5.6.10, MySQL Enterprise Edition is available for MySQL 5.6. Specifically, MySQL Enterprise 5.6.10 includes these components previously available only in MySQL 5.5: MySQL Enterprise Security (PAM and Windows authentication plugins), MySQL Enterprise Audit, and MySQL Thread Pool. For information about these features, see MySQL Enterprise Edition. To learn more about commercial products, see http://www.mysql.com/products/.

Known limitations of this release:

On Microsoft Windows, when using the MySQL Installer to install MySQL Server 5.6.10 on a host with an existing MySQL Server of a different version (such as 5.5.30), that also has a different license (community versus commercial), you must first update the license type of the existing MySQL Server. Otherwise, MySQL Installer will remove MySQL Server(s) with different licenses from the one you chose with MySQL Server 5.6.10.

On Microsoft Windows 8, updating a community release to a commercial release requires you to manually restart the MySQL service after the update.

Functionality Added or Changed

- **InnoDB:** When compressed tables were used, the calculation to compute memory usage within the buffer pool was complex because the compressed pages could be smaller than 16KB or the user-specified page size. Although this information can be retrieved from the INFORMATION_SCHEMA.INNODB_BUFFER_PAGE table, that operation is expensive. The following new status variables help to simplify calculations involving buffer pool memory usage:

  - Innodb_buffer_pool_bytes_data, to supplement Innodb_buffer_pool_pages_data.
  - Innodb_buffer_pool_bytes_dirty, to supplement Innodb_buffer_pool_pages_dirty.

  (Bug #15842637)

- **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)

- 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)

- The validate_password_policy_number system variable was renamed to validate_password_policy. (Bug #14588121)

- 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)

Bugs Fixed

- **Performance; InnoDB:** Some data structures related to undo logging could be initialized unnecessarily during a query, although they were only needed under specific conditions. (Bug #14676084)

- **Performance; InnoDB:** Optimized read operations for compressed tables by skipping redundant tests. The check for whether any related changes needed to be merged from the insert buffer was being called more often than necessary. (Bug #14329288, Bug #65886)

- **Performance; InnoDB:** Immediately after a table was created, a query against it would not use a loose index scan. The same query might use a loose index scan following an ALTER TABLE on the table. The fix improves the accuracy of the cost estimate for queries involving the grouping functions min() and max(), and prevents the query plan from being changed by the ALTER TABLE statement. (The more stable query plan might or might not use a loose index scan.) (Bug #14200010)

- **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)

- **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:** In online DDL operations, a DROP FOREIGN KEY clause was not allowed in an ALTER TABLE statement that also performed any of the following:

  - Adding or dropping a column.

  - Adding or dropping a primary key index.

  - Making a column NULL or NOT NULL.

  - Reordering columns.

  - Changing the ROW_FORMAT or KEY_BLOCK_SIZE properties.

  (Bug #16095573, Bug #68019)

- **InnoDB:** During an online DDL operation, changing a column from nullable to NOT NULL could succeed or fail differently depending on whether the ALTER TABLE statement used ALGORITHM=INPLACE or ALGORITHM=COPY. An operation with ALGORITHM=COPY would succeed even if the column contained NULL values, while an operation with ALGORITHM=INPLACE would fail because of the possibility that the column contained NULL values. Now, making a column NOT NULL in combination with the ALGORITHM=INPLACE clause is allowed, but only if the sql_mode configuration option includes the STRICT_TRANS_TABLES or STRICT_ALL_TABLES setting. If the ALGORITHM clause is not specified with the ALTER TABLE statement, the online DDL operation will use ALGORITHM=INPLACE if possible, or ALGORITHM=COPY if not. (Bug #15961327)

- **InnoDB:** Under certain circumstances, an InnoDB table was reported as corrupted after import using ALTER TABLE ... IMPORT TABLESPACE. The problem was accompanied by one of these messages:

  ```
  Warning  : InnoDB: The B-tree of index "PRIMARY" is corrupted.
  error    : Corrupt
  ```

or:

```
Warning   : InnoDB: The B-tree of index "GEN_CLUST_INDEX" is corrupted.
error     : Corrupt
```

This issue occurred intermittently, and primarily affected large tables. The `REPAIR TABLE` statement would fix the problem reported by the error message. (Bug #15960850, Bug #67807)

- **InnoDB:** Some Valgrind warnings were issued during shutdown, while cleaning up a background thread that handles optimization of tables containing `FULLTEXT` indexes. (Bug #15994393)
- **InnoDB:** During an online DDL operation to add a unique index, DML operations that created duplicate values could fail with an `ER_DUP_KEY` error even though the index had not been made visible yet. (There was a brief time window when this condition could occur.) The fix causes the index creation operation to fail instead, if the index would be invalid because of duplicate entries produced by concurrent DML. (Bug #15920445)
- **InnoDB:** If an online DDL operation to add a unique index failed, because duplicate items were created by concurrent DML during the online DDL operation, the `ALTER TABLE` operation failed with the wrong error type. It returned `ER_INDEX_CORRUPT`; now it returns the new error code `ER_DUP_UNKNOWN_IN_INDEX`. (It does not return `ER_DUP_KEY`, because the duplicate key value is not available to be reported when this condition occurs.) (Bug #15920713)
- **InnoDB:** `ALTER TABLE` statements using the online DDL feature could cause Valgrind warnings. (Bug #15933178)
- **InnoDB:** Names of indexes being created by an online DDL operation were being displayed incorrectly in `INFORMATION_SCHEMA` tables while the operation was in progress. This fix ensures the table names have the leading `0xff` byte stripped off for `INFORMATION_SCHEMA` queries. This change affects the columns:

  - `innodb_buffer_page.index_name`
  - `innodb_buffer_page_lru.index_name`
  - `innodb_cmp_per_index.index_name`
  - `innodb_cmp_per_index_reset.index_name`
  - `innodb_locks.lock_index`
  - `innodb_sys_indexes.name`

  (Bug #15946256)
- **InnoDB:** The status variable `Innodb_buffer_pool_read_ahead_evicted` could show an inaccurate value, higher than expected, because some pages in the buffer pool were incorrectly considered as being brought in by read-ahead requests. (Bug #15859402, Bug #67476)
- **InnoDB:** Creating an index on a `CHAR` column could fail for a table with a character set with varying length, such as `UTF-8`, if the table was created with the `ROW_FORMAT=REDUNDANT` clause. (Bug #15874001)
- **InnoDB:** If the server crashed near the end of an online DDL `ALTER TABLE` statement, a subsequent `CHECK TABLE` statement using the `EXTENDED` clause could cause a serious error. (Bug #15878013)
- **InnoDB:** Specifying an `innodb_log_file_size` value of 4GB or larger was not possible on 64-bit Windows systems. This issue only affected debug builds. (Bug #15882860)

- **InnoDB:** This fix ensures that in case of a serious unhandled error during an ALTER TABLE operation that copies the original table, any data that could be needed for data recovery is preserved, in tables using names of the form #sql-ib-*table_id* or #mysql50##sql-ib-*table_id*. (Bug #15866623)

- **InnoDB:** An online DDL operation to add a primary key to a table could encounter a serious error if the table also had an index on a column prefix of a BLOB column.

  This fix suspends the background purge operation while a table is being rebuilt by an ALTER TABLE statement, if any rows containing off-page columns would be removed. Currently, to avoid excessive space usage during the online DDL operation, avoid these types of concurrent DML operations until the ALTER TABLE is finished:

  - DELETE of rows that contain off-page columns.

  - UPDATE of primary key columns in rows that contain off-page columns.

  - UPDATE of off-page columns.

  (Bug #14827736)

- **InnoDB:** The server could halt with an assertion error when creating an index on a column prefix for a column using a multibyte character set:

  ```
  InnoDB: Assertion failure in thread thread_num in file row0merge.cc line 465
  InnoDB: Failing assertion: len == ifield-<fixed_len
  ```

  (Bug #14753402)

- **InnoDB:** The server could halt with an assertion error while creating an index:

  ```
  InnoDB: Assertion failure in thread thread_num in file row0merge.cc line 465
  ```

  This issue affected tables with a combination of ROW_FORMAT=REDUNDANT off-page columns, and an index on a column prefix. (Bug #14753402)

- **InnoDB:** A regression introduced by the fix for Bug#14100254 would result in a “!BPAGE->FILE_PAGE_WAS_FREED” assertion. (Bug #14676249)

- **InnoDB:** INFORMATION_SCHEMA tables with InnoDB metadata, such as innodb_sys_tablestats, displayed nonalphanumeric characters in the names of tables using an encoded format, for example with @0024 instead of $. (Bug #14550145)

- **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:** With a large value for innodb_buffer_pool_size, and innodb_buffer_pool_instances set greater than 1, pages were being incorrectly evicted from the buffer pool. (Bug #14125092)

- **InnoDB:** Corruption of the innodb_ft_user_stopword_table table could cause a server exit. (Bug #67960, Bug #16038656)

- **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.

- **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:** Issuing START SLAVE UNTIL SQL_BEFORE_GTIDS = *gtid_set*, where *gtid_set* covered a large number (tens or hundreds of millions) of transactions, could cause the server to hang. (Bug #15968413)

- **Replication:** When a slave was started using --skip-innodb and replication info file repositories (FILE being the default for both --relay-log-info-repository and --master-info-repository), replication was incorrectly stopped. However, if the slave is using file repositories and not currently migrating between info repositories, replication should be able to work without issues. Now the server ignores errors raised when trying to open table info repositories in such conditions.

  In addition, binary log initialization was not performed correctly when starting the slave with --skip-innodb, which caused the --log-bin option to be ignored. (Bug #15956714, Bug #67798, Bug #15971607)

- **Replication:** When temporary and persistent tables, or temporary tables using different storage engines, are dropped in a single statement, this statement is actually written as two statements to the binary log, each represented by its own log event. When gtid_mode is ON, each DDL event must have a GTID; however, in such cases, the statement dropping the temporary table was uncommitted, which meant that it was not given its own GTID.

  Now, when a DDL statement dropping a temporary table and a table that is persistent, or that uses a different storage engine, is separated in the manner just described, and the resulting logged statement affecting only the temporary table does not implicitly commit, a commit is forced so that the corresponding log event has own unique GTID. (Bug #15947962)

- **Replication:** When used on a binary log that had been written by a GTID-enabled server, mysqlbinlog did not correctly handle transactions left unclosed by the omission of statements that were ignored when the --database option was employed.

  Now, whenever mysqlbinlog --database reads a GTID log event, it checks to see whether there is an unclosed transaction, and if so, issues a commit. (Bug #15912728)

- **Replication:** Semisynchronous replication did not work correctly with GTIDs enabled. (Bug #15927032)

  References: See also Bug #14737388.

- **Replication:** When GTIDs were enabled, the automatic dropping of a temporary table when a client disconnected did not always generate a GTID. Now each logged DROP TABLE statement, including any generated by the server, is guaranteed to have its own GTID. (Bug #15907504)

- **Replication:** After dropping a column from the slave's version of a table, then altering the same column of this table on the master (so that a type conversion would have been required had the column not been droppped on the slave), inserts into this table caused replication to fail. (Bug #15888454)

- **Replication:** Use of sql_slave_skip_counter is not compatible with GTID-based replication. Setting this variable to a nonzero value is now disallowed whenever --gtid-mode = ON, and attempting to do so fails with an error. (Bug #15833516)
- **Replication:** When a binary log is replayed on a server (for example, by executing a command like mysqlbinlog binlog.000001 | mysql), it sets a pseudo-slave mode on the client connection used, so that the server can read binlog and apply binary log events correctly. However, the pseudo-slave mode was not disabled after the binary log dump was read, which caused unexpected filtering rules to be applied to SQL statements subsequently executed on the same connection. (Bug #15891524)
- **Replication:** During mysqld shutdown, global GTID variables were released before it was made certain that all plugins had stopped using them. (Bug #14798275)
- **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 #14737388)

  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:** Names of databases in binary log query log events were not properly checked for length. (Bug #14636219)
- **Replication:** Issuing START SLAVE concurrently with setting sql_slave_skip_counter or slave_net_timeout could cause a deadlock. (Bug #14236151)
- **Replication:** When using statement-based replication, and where the master and the slave used table schemas having different AUTO_INCREMENT columns, inserts generating AUTO_INCREMENT values logged for a given table on the master could be applied to the wrong table on the slave. (Bug #12669186)
- **Replication:** Repeated execution of CHANGE MASTER TO statements using invalid MASTER_LOG_POS values could lead to errors and possibly a crash on the slave. Now in such cases, the statement fails with a clear error message. (Bug #11764602, Bug #57454)
- **Microsoft Windows:** Dynamic file names (with colons) are no longer allowed. Static file names using the Alternate Data Stream (ADS) NTFS functionality of Microsoft Windows may continue to be used. (Bug #11761752)
- 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)
- During client connection processing, the server now performs password-expiration checking after SSL checks. (Bug #16103348)
- A buffer-handling problem in yaSSL was fixed. (Bug #15965288)
- The plugin logging routine mishandled its argument, resulting in undefined behavior. (Bug #16002890)
- An online DDL operation that dropped an index could proceed despite not having sufficient locks on the table. This issue could cause a serious error, although the error was only observed in debug builds. (Bug #15936065)
- An ALTER TABLE with the ADD PRIMARY KEY or ADD UNIQUE INDEX clause could encounter a serious error if the columns for the primary key or unique index contained duplicate entries. This error occurred intermittently, depending on how the rows were physically distributed across index blocks. (Bug #15908291)
- The ALTER TABLE statement can now use the LOCK=NONE clause, allowing online DDL with concurrent DML, for child tables containing foreign key constraints. (Bug #15912214)

- In certain rare cases, a query using UpdateXML() could cause the server to crash. (Bug #15948580)

  References: See also Bug #13007062.

- AES_DECRYPT() and AES_ENCRYPT() had memory leaks when MySQL was compiled using OpenSSL. (Bug #15909183)
- Several OpenSSL-related Valgrind warnings were corrected. (Bug #15908967)
- Several OpenSSL-related memory leaks were fixed. (Bug #15921729)
- Very long database names in queries could cause the server to exit. (Bug #15912213, Bug #16900358)
- Within a stored procedure, executing a multiple-table DELETE statement that used a very long table alias could cause the server to exit. (Bug #15954896)
- Very long table aliases in queries could cause the server to exit. (Bug #15948123)
- Metadata locking and table definition cache routines did not always check length of names passed to them. (Bug #15954872)
- A comment added to mysqldump output for the --set-gtid-purged option was malformed and caused a syntax error when the dump file was reloaded. (Bug #15922502)

  References: See also Bug #14832472.

- Contention in the thread pool during kill processing could lead to a Valgrind panic. (Bug #15921866)
- In the absence of a FULLTEXT index on an InnoDB table, a full-text query with COUNT(*) could raise an assertion. (Bug #15950531)
- If an error occurred during the final phase of an online DDL operation, some cached metadata about the table might not be restored to its original state. This issue typically affected operations that renamed a column, and also dropped and re-created an index on that column, in the same ALTER TABLE statement. This issue did not affect operations that reorganize the clustered index of the table, such as adding a new primary key. (Bug #15866734)
- In debug builds, the server could not start on 64-bit Windows systems when a value of 16 GB or higher was specified for innodb_buffer_pool_size. Non-debug builds would likely have subtler issues, such as memory being allocated for the buffer pool but not used, or read requests overlooking pages already cached in the buffer pool.

  On 32-bit Windows systems, the value of innodb_buffer_pool_instances is increased if necessary so that no buffer pool instance is larger then 1.3 GB, due to system limitations on memory allocation. This automatic adjustment needed for 32-bit Windows systems was incorrectly applied to 64-bit systems also; for systems with 16 GB or larger buffer pools, the adjusted value of innodb_buffer_pool_instances would exceed the upper limit of 64, causing an assertion error in debug builds. (Bug #15883071)

- A heavy workload of online DDL and concurrent DML on a table on a master server could cause errors as the changes were replicated to slave servers. For example, processing a DROP COLUMN operation at the same time as queries referring to the dropped column could cause errors on slave servers if the statements finished in a different order than on the master. (Bug #15878880)
- If the server shut down unexpectedly, the presence of an InnoDB table with 1018 columns (very close to the upper limit of 1020 columns) could cause an assertion error during server restart:

  ```
  InnoDB: Failing assertion: table->n_def == table->n_cols - 3
  ```

  (Bug #15834685)

- The Performance Schema normally ignores temporary table events. User-defined temporary tables are truncated by being re-created, but the Performance Schema did not recognize re-created temporary tables as being temporary and raised an assertion. (Bug #15884836)

- The Performance Schema `session_connect_attrs` table displayed extraneous information. (Bug #15864703)

- Subqueries with `COUNT(DISTINCT ...))` could cause the server to exit. (Bug #15832620)

  References: See also Bug #11750963.

- `Rows_log_event` allocated one too few bytes for the row buffer. (Bug #15890178)

- For the LooseScan semi-join strategy, the optimizer could rely on an uninitialized variable. (Bug #15849654)

- For debug builds, an assertion could be raised when: 1) A view was based on a `MEMORY` table; 2) The table was altered to drop some column in use by the view; 3) A `SELECT` was done on the view with binary logging disabled. (Bug #15847447)

- If loose index scan was used on a query with descending order, the result set contained `NULL` values instead of the correct values. (Bug #15848665)

- The optimizer's cost-based choice between `IN -> EXISTS` subquery transformation and subquery materialization was sometimes incorrect if the `IN` predicate was `OR`-ed with some other predicate. (Bug #15866339)

  References: See also Bug #13111584.

- In some cases, a cost value was printed to Optimizer Trace output without being initialized, resulting in incorrect output. (Bug #15877453)

- Several code issues identified by Fortify were corrected. (Bug #15884324)

- Some queries, if used as prepared statements, caused the server to exit if an error occurred. (Bug #15877062)

- Complex `IN` subqueries could cause the server to exit. (Bug #15877738)

- It was possible to expire the password for an account even if the account is authenticated by an authentication plugin that does not support password expiration. (Bug #15849009)

- When the server reads the `mysql.user` table, it now checks for invalid native and old-native password hashes and ignores accounts with invalid hashes. (Bug #14845445)

- The `validate_password` plugin did not check certain passwords. (Bug #14843970)

- `GRANT ... IDENTIFIED BY` could fail to flush the privileges. (Bug #14849959)

- Setting the `validate_password_length` system variable did not take into account that the minimum value is a function of several other related system variables. Now the server will not set the value less than the value of this expression:

  ```
  validate_password_number_count
  + validate_password_special_char_count
  + (2 * validate_password_mixed_case_count)
  ```

  (Bug #14850601)

- An issue with locking order for the system tables and the InnoDB data dictionary could lead to an internal deadlock within MySQL. (Bug #14805484)

- When used with an XPath expression that contained the output of a stored function, ExtractValue() failed with the error Only constant XPATH queries are supported. (Bug #14798445, Bug #67313)

- MySQL could encounter an error during shutdown on Windows XP or earlier systems. This issue did not affect systems running Windows Vista or higher, which use atomic condition variables to represent Windows Events. (Bug #14822849)

- Temporary table creation during execution of INFORMATION_SCHEMA queries could result in Valgrind warnings. (Bug #14801497)

- mysqladmin did not properly process commands for users with expired passwords. (Bug #14833621)

- XA START had a race condition that could cause a server crash. (Bug #14729757)

- The clause ALGORITHM=INPLACE clause in an ALTER TABLE statement for a partitioned table could lead to consistency issues if a crash occurred while changes were applied to some of the underlying tables but not all. This fix prohibits the ALGORITHM=INPLACE clause for DDL operations on partitioned tables. (Bug #14760210)

- The server could halt with an assertion error due to a recently added error code:

  ```
  InnoDB: unknown error code 1502
  InnoDB: Assertion failure in thread thread_num in file row0mysql.cc line 683
  mysqld got signal 6 ;
  ```

  Now, the server returns the error code DB_DICT_CHANGED to the client in this case. (Bug #14764015)

- Queries that used grouping failed when executed using a cursor if the optimizer processed the grouping using a temporary table. (Bug #14740889)

- The server could exit when the MyISAM storage engine (rather than MEMORY) was used to materialize a derived table. (Bug #14728469)

- The sha256_password authentication plugin requires that the client connect either using SSL or have RSA enabled. When neither condition was met, an uninformative error message was produced. Now the error message is more informative. (Bug #14751925)

- The server now logs warnings at startup if the file specified for the validate_password_dictionary_file system variable violates constraints on valid password file contents. (Bug #14588148)

- At startup, some InnoDB boolean system variables could be set to 1 or 0, but not ON or OFF. These included innodb_file_per_table, innodb_force_load_corrupted, and innodb_large_prefix. (Bug #14494893)

- Output generated with mysqldump --routines could produce syntax errors when reloaded. (Bug #14463669)

- Calculations involving self-intersecting polygons caused an assertion to be raised. (Bug #14503584)

- If ALTER TABLE was killed, the server could report ER_QUERY_INTERRUPTED even if the alterations had been made successfully. This is misleading to the user. Also, the statement would not be written to the binary log, leading to incorrect replication (Bug #14382643)

- The parser failed to return an error for some invalid UNION constructs. (Bug #13992148)

- 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.

In addition, for explicit client plugin loading, the mysql_load_plugin() and mysql_load_plugin_v() C API functions have been modified to use the LIBMYSQL_PLUGIN_DIR value if it exists and the --plugin-dir option was not given. If --plugin-dir is given, mysql_load_plugin() and mysql_load_plugin_v() ignore LIBMYSQL_PLUGIN_DIR. (Bug #13994567, Bug #18110355)

- With the ONLY_FULL_GROUP_BY SQL mode enabled, executing a stored function twice that contains a SQL query that is not valid with that mode enabled caused the server to exit. (Bug #13996639)
- Autosizing of Performance Schema parameters could result in settings that caused excessive CPU use. (Bug #67736, Bug #15927744)
- The optimizer sometimes chose a nonoptimimal range scan strategy when a query included a LIMIT clause. (Bug #67432, Bug #15829358)
- Full-text searches in InnoDB tables could return incorrect results. (Bug #67257, Bug #14771282)
- The mysql client could mishandle the delimiter command if it occurred on a line during which mysql was looking for the end of a quoted string. (Bug #64135, Bug #13639125)
- The Performance Schema normally ignores temporary table events, but sometimes failed to properly identify a table as temporary and consequently recorded events for the table. (Bug #67098, Bug #14756887)
- Some messages written by the server to the error log referred to the deprecated --log-slow-queries option rather than the --slow-query-log option. Similarly, the server referred to the deprecated --log option rather than the --general-log-file and --log-output options. (Bug #67892, Bug #15996571)
- Attempting to perform an in-place upgrade from MySQL 5.1 to 5.6 causes the server to exit due to a mismatch between the privilege structures in the two series. (This is not a supported operation, but the server should not exit ungracefully.) (Bug #67319, Bug #14826854)
- DECIMAL multiplication operations could produce significant inaccuracy. (Bug #45860, Bug #11754279)
- Due to a thread race condition, the server could exit while attempting to read the Performance Schema threads.PROCESSLIST_INFO column. (Bug #68127, Bug #16196158)
- The optimizer could choose an IN-to-EXISTS transformation for subquery execution in some cases when subquery materialization would be cheaper. (Bug #67511, Bug #15848521)
- It is not permitted to use CREATE TABLE to create an NDB table with user-defined partitioning and a foreign key. However, it was possible to create an NDB table with a foreign key, then add partitioning to it using ALTER TABLE, thus creating a table which was impossible to back up or restore using mysqldump. Now the prohibition is enforced consistently. (Bug #67492, Bug #15844519)
- For single-table DELETE or UPDATE statements, EXPLAIN displayed a type value of ALL (full-table scan access method) even if the optimizer chose to scan the table by an index access method. Now the type value is displayed as index. (Bug #67637, Bug #15892875)
- mysqldump could fail to dump all tables in the mysql database. (Bug #67261, Bug #14771252)

Changes in MySQL 5.6.9 (2012-12-11)

Installation Notes

- The --random-passwords option for mysql_install_db is now supported for MySQL install operations (not upgrades) using Solaris PKG packages.

Functionality Added or Changed

- **Incompatible Change; Replication:** A number of variable and other names relating to GTID-based replication have been changed, with a view to making these names more appropriate and meaningful. *The old names are no longer supported*.

  The features so renamed are shown in the following list:

  - The `--disable-gtid-unsafe-statements` server option has been renamed `--enforce-gtid-consistency`; the `disable_gtid_unsafe_statements` system variable has been renamed `enforce_gtid_consistency`.
  - The `gtid_done` server system variable has been renamed `gtid_executed`.

    The `gtid_lost` server system variable has been renamed `gtid_purged`; in addition, this variable is no longer read-only.
  - The `SQL_THREAD_WAIT_AFTER_GTIDS()` function has been renamed `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`.

  For more information, see Replication with Global Transaction Identifiers, and Global Transaction ID Options and Variables. (Bug #14775984)

- **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 deprecated and will be removed in a future MySQL release. See Using Symbolic Links for Databases on Windows.

- The server now provides thread information (for `SHOW PROCESSLIST`) to indicate the progress of in-place `ALTER TABLE` operations:

  - `preparing for alter table`

    The server is preparing to execute an in-place `ALTER TABLE`.
  - `altering table`

    The server is in the process of executing an in-place `ALTER TABLE`.
  - `committing alter table to storage engine`

    The server has finished an in-place `ALTER TABLE` and is committing the result.

  (Bug #14790408)

- For client connections restrictd by the server because the client account password is expired, the server now permits `SET PASSWORD` only if the account named in the statement matches the account used by the client. (Bug #14807074)

  References: See also Bug #14698309.

- `mysqld` now writes dates to the error log in ISO (`YYYY-MM-DD hh:mm:ss`) format. It also includes its process ID following the date. Thanks to Davi Arnaut for the patch. (Bug #56240, Bug #11763523)

- `InnoDB` automatically extends each secondary index by appending the primary key columns to it. Previously, the optimizer did not take into account the primary key columns of the extended secondary index when determining how and whether to use that index. Now the optimizer takes the primary key columns into account, which can result in more efficient query execution plans and better performance.

  The optimizer can use extended secondary keys for `ref`, `range`, and `index_merge` index access, for loose index scans, for join and sorting optimization, and for `MIN()`/`MAX()` optimization.

The new `use_index_extensions` flag of the `optimizer_switch` system variable permits control over whether the optimizer takes the primary key columns into account when determining how to use an `InnoDB` table's secondary indexes. By default, `use_index_extensions` is enabled. To check whether disabling use of index extensions will improve performance, use this statement:

```
SET optimizer_switch = 'use_index_extensions=off';
```

For more information, see Use of Index Extensions. (Bug #62025, Bug #12814559, Bug #56714, Bug #11763940)

Bugs Fixed

- **Performance; InnoDB:** The timing values for low-level `InnoDB` read operations were adjusted for better performance with fast storage devices, such as SSD. This enhancement primarily affects read operations for `BLOB` columns in compressed tables. (Bug #13702112, Bug #64258)

- **Incompatible Change:** The `THREAD_ID` column in Performance Schema tables was widened from `INT` to `BIGINT` to accommodate 64-bit values.

  Note

  As a consequence of this change, the `PROCESSLIST_ID` column of the `threads` table is now `NULL` for background threads. Previously, the value was 0 for background threads.

  If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this change to the `performance_schema` database. (Bug #14664453)

- **Incompatible Change:** Connection ID (thread ID) values greater than 32 bits can occur on some systems (such as busy or long-running 64-bit systems), causing these problems:

  - Connection IDs written to the general query log and slow query log were incorrect. This was true for logging to both files and tables.

  - The `CONNECTION_ID()` function could return a value with a data type too small for values larger than 32 bits.

  - The `mysql_thread_id()` and `mysql_kill()` C API functions did not handle ID values larger than 32 bits. This could result in killing the wrong thread; for example, if you invoked `mysql_kill(mysql_thread_id())`.

  Connection IDs now are permitted to be 64-bit values when the server supports them (when built with 64-bit data types), which has these effects:

  - Connection IDs are logged correctly to the general query log and slow query log.

    Note

    This change involves a modification to the log tables, so after upgrading to this release, you must run mysql_upgrade and restart the server.

  - `CONNECTION_ID()` returns a data type appropriate for values larger than 32 bits.

  - `mysql_thread_id()` is unchanged; the client/server protocal has only 4 bytes for the ID value. This function returns an incorrect (truncated) value for connection IDs larger than 32 bits and should be avoided.

    `mysql_kill()` still cannot handle values larger than 32 bits, but to guard against killing the wrong thread now returns an error in these cases:

    - If given an ID larger than 32 bits, `mysql_kill()` returns a `CR_INVALID_CONN_HANDLE` error.

    - After the server's internal thread ID counter reaches a value larger than 32 bits, it returns an `ER_DATA_OUT_OF_RANGE` error for any `mysql_kill()` invocation and `mysql_kill()` fails.

    To avoid problems with `mysql_thread_id()` and `mysql_kill()`, do not use them. To get the connection ID, execute a `SELECT CONNECTION_ID()` query and retrieve the result. To kill a thread, execute a `KILL` statement.

  (Bug #19806, Bug #11745768, Bug #65715, Bug #14236124, Bug #44728, Bug #11753308)

- **Incompatible Change:** `LAST_INSERT_ID(*expr*)` did not work for *`expr`* values greater than the largest signed `BIGINT` value. Such arguments now are accepted, with some consequences for compatibility with previous versions:

  - `LAST_INSERT_ID()` now returns a `BIGINT UNSIGNED` value, not a `BIGINT` (signed) value.

  - `LAST_INSERT_ID(*expr*)` now returns an unsigned integer value, not a signed integer value.

  - For `AUTO_INCREMENT` columns, negative values are no longer supported.

  (Bug #20964, Bug #11745891)

- **Important Change; InnoDB:** A DML statement using the index merge access method could lock many rows from the table, even when those rows were not part of the final result set. This fix reduces the excessive locking by releasing the locks of unmatched rows. This optimization affects only transactions with isolation level equal to or less strict than `READ COMMITTED`; it does not apply to transactions using `REPEATABLE READ` or `SERIALIZABLE` isolation level. (Bug #14226171)

- **Important Change; Replication:** Because running the server with GTIDs enabled prevented changes to nontransactional tables, programs such as `mysql_upgrade` and `mysql_install_db` were unable to operate on system tables that used the MyISAM storage engine and therefore could not function correctly. Now, when running with `--enforce-gtid-consistency` (required whenever `--gtid-mode=ON`), the server allows single statements on nontransactional tables. (Bug #14722659)

- **Important Change; Replication:** Formerly, the value of the `Seconds_Behind_Master` column in the output of `SHOW SLAVE STATUS` was always set to `NULL` whenever the SQL thread or the I/O thread was stopped. Now, this column is set to `NULL` only if the SQL thread is not running, or if the I/O thread is not running following a check to determine whether or not the SQL thread has processed all of the relay log. (If the SQL thread has finished processing and the I/O thread is running, `Seconds_Behind_Master` is 0.) (Bug #12946333)

- **InnoDB; Partitioning:** Previously, when attempting to optimize one or more partitions of a partitioned table that used a storage engine that does not support partition-level `OPTIMIZE`, such as `InnoDB`, MySQL reported `Table does not support optimize, doing recreate + analyze instead`, then re-created the entire table, but did not actually analyze it. Now in such cases, the warning message is, `Table does not support optimize on partitions. All partitions will be rebuilt and analyzed.` In addition, the entire table is analyzed after first being rebuilt. (Bug #11751825, Bug #42822)

- **InnoDB:** If the server crashed while rows were inserted into a table with a `FULLTEXT` index but before the transaction was committed, an error could occur during the next startup:

  ```
  InnoDB: Assertion failure in thread thread_num in file dict0dict.cc line 1019
  ```

  (Bug #14826779)

- **InnoDB:** The server could halt with an error when accessing an `InnoDB` table containing a `FULLTEXT` index through the `HANDLER` statement. (Bug #14788710)

- **InnoDB:** The server could halt with an error when two kinds of operations happened simultaneously:
  - A ROLLBACK of an inserted row that contained off-page columns.
  - An online DDL operation involving a table of ROW_FORMAT=DYNAMIC or ROW_FORMAT=COMPRESSED (that is, using the Barracuda file format) that rebuilt the table. For example, ADD/DROP COLUMN, ADD PRIMARY KEY, change ROW_FORMAT.

  (Bug #14842014)

- **InnoDB:** The server could halt with an assertion error for an ANALYZE TABLE operation, depending on the structure of the table and its indexes:

  ```
  InnoDB: Assertion failure in thread thread_num in file dict0dict.ic line 447
  InnoDB: Failing assertion: pos < table->n_def
  ```

  (Bug #14755452)

- **InnoDB:** A timeout error could occur on Windows systems when doing ALTER TABLE statements with the DISCARD TABLESPACE or IMPORT TABLESPACE clauses, due to a temporary tablespace file remaining in the file system. (Bug #14776799)

- **InnoDB:** An online DDL operation for an InnoDB table incorrectly reported an empty value ('') instead of the correct key value when it reported a duplicate key error for a unique index using an index prefix. (Bug #14729221)

- **InnoDB:** InnoDB tables with FULLTEXT indexes could allocate memory for thread handles that was never released, possibly leading to resource issues on Windows systems. (Bug #14759163)

- **InnoDB:** During an online DDL operation that copies the table, the secondary index of the table could become corrupted. (Bug #14753701)

- **InnoDB:** This fix makes MySQL more responsive to KILL QUERY statements when the query is accessing an InnoDB table. (Bug #14704286)

- **InnoDB:** If the server crashed after an online DDL CREATE INDEX operation, an error could occur while rolling back incomplete transactions on the next startup:

  ```
  InnoDB: error in sec index entry del undo in
  ...
  InnoDB: Assertion failure in thread thread_num in file row0umod.cc line 559
  ```

  (Bug #14707452)

- **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:** This fix improves the error handling when an ALTER TABLE operation adds a column beyond the maximum number allowed for an InnoDB table. It also raises the maximum number of columns for an InnoDB table from 1000 to 1020. (Bug #14705287)

- **InnoDB:** During an online DDL operation that rebuilt the table, a CHECK TABLE statement could report a count mismatch for all secondary indexes. (Bug #14606472)

- **InnoDB:** After a FULLTEXT index was created and dropped from an InnoDB table, further ALTER TABLE operations to add, drop, and rename columns could cause a serious error. Regression of bug #13972248. (Bug #14504337)

- **InnoDB:** If an ALTER TABLE statement failed while attempting to create a FULLTEXT index for an InnoDB table, the server could halt with an assertion error while dropping the incomplete index. (Bug #14504174)

- **InnoDB:** During shutdown, with the innodb_purge_threads configuration option set greater than 1, the server could halt prematurely with this error:

  ```
  mysqld got signal 11
  ```

  A workaround was to increase innodb_log_file_size and set innodb_purge_threads=1. The fix was backported to MySQL 5.5 and 5.1, although those versions do not have the innodb_purge_threads configuration option so the error was unlikely to occur. (Bug #14234028)

- **InnoDB:** The server could halt with an error under some combinations of concurrent operations:

  ```
  InnoDB: unknown error code 20
  ```

  This issue originated during the 5.6 development cycle. It affected only transactions using the READ COMMITTED andREAD UNCOMMITTED isolation levels. (Bug #13641662, Bug #12424846)

- **InnoDB:** This fix improves the error message when a foreign key constraint cannot be created. Instead of referring to an inability to create a table with an auto-generated name, the message clearly states the error:

  ```
  ERROR 1215 (HY000): Cannot add foreign key constraint
  ```

  Issuing a subsequent SHOW WARNINGS statement provides additional detail about any secondary indexes that are required. (Bug #11745444, Bug #15324)

- **Replication:** If a table to be replicated had a FULLTEXT index, this index was not ruled out when selecting the type of scan to be used in finding the next row, even though it cannot be used to find the correct one. The row applier subsequently tried unsuccessfully to employ an index scan, causing replication to fail. Now in such cases, indexes which do not provide for sequential access (such as FULLTEXT) are not considered when determining whether to use a table, index, or hash scan for this purpose. (Bug #14843764)

- **Replication:** When using the GTID-aware master-slave protocol, the slave I/O thread used the wrong position. When using GTIDs, the position is not normally used, but as a special case, the position was used in addition to the GTID when the slave reconnected to the same master (even though this was not necessary). This problem is fixed by making the GTID-aware master-slave protocol not use positions at all any longer. (Bug #14828028)

- **Replication:** Given a stored routine R in which the GTID_SUBTRACT() function was invoked: Once GTID_SUBTRACT() returned NULL when called inside R, it continued to return NULL every time it was called within R, for the remainder of the client session. (Bug #14838575)

- **Replication:** MySQL Enterprise Backup, mysqldump, and mysqlhotcopy could not be used with a GTID-enabled MySQL Server, because they were unable to restore the server's GTID state and so could not restore from any point in the binary log other than the very beginning.

  As part of the fix for this problem, the gtid_purged system variable (formerly named gtid_lost) is no longer read-only; now it is possible to add GTIDs to it when gtid_executed (formerly gtid_done) is empty. (Bug #14787808)

- **Replication:** Restarting replication after the first binary log file was purged resulted in the error Got fatal error 1236 from master when reading data from binary log: 'The slave is connecting using CHANGE MASTER TO MASTER_AUTO_POSITION = 1, but the master has purged binary logs containing GTIDs that the slave requires.' This led GTID-based replication to fail. (Bug #14756691)

- mysql_install_db failed to honor the --user option. (Bug #15866735)
- The optimizer could allocate insufficient memory when determining subquery execution strategies, causing the server to exit. (Bug #14846866)
- Creating an InnoDB table with a FULLTEXT index could encounter a serious error if the table name contained nonalphanumeric characters. (Bug #14835178, Bug #16036699)
- Invalid memory reads could occur for queries that selected from a zero-length table name. (Bug #14780820)
- With LOCK TABLES in effect, CREATE TABLE IF NOT EXISTS ... LIKE could raise an assertion. (Bug #14788976)
- The automatic key generation part of derived table handling did not handle properly columns specified as part of the VALUES() clause and caused an assertion to be raised. (Bug #14786324)
- Attempting to read a utf16 file with LOAD DATA INFILE raised an assertion. (Bug #14786470)
- init_io_cache() used memset() to clear a mutex but passed the wrong mutex size. (Bug #14838882)
- The optimizer could raise an assertion when evaluating a range test against an IS NOT NULL condition. (Bug #14843705)
- An assertion could be raised executing INSERT, UPDATE, or DELETE after implicitly starting a READ ONLY transaction in LOCK TABLES mode. (Bug #14788540)
- When a backup is taken using mysqldump on a server with global transaction IDs (GTIDs) enabled, the dump file did not contain any GTID information. This eventually results in replicating the transactions from the beginning of history when the backup is used to bring up a slave.

  To enable control over GTID information written to the dump file, mysqldump now has a --set-gtid-purged option that indicates whether to add a SET @@global.gtid_purged statement to the output.

  The following table shows the permitted option values. The default value is AUTO.

Value	Meaning
OFF	Add no SET statement to the output.
ON	Add a SET statement to the output. An error occurs if GTIDs are not enabled on the server.
AUTO	Add a SET statement to the output if GTIDs are enabled on the server.

  (Bug #14832472)
- A query with a union and a join could crash the parser. (Bug #14786792, Bug #16076289)
- Out-of-bounds reads could occur within filename_to_tablename(). (Bug #14834378)
- After issuing ALTER TABLE ... DISCARD TABLESPACE, an online DDL operation for the same table could fail on Windows systems with an error: Got error 11 from storage engine. An ALTER TABLE statement with the ALGORITHM=INPLACE clause could also create an empty .ibd file, making the tablespace no longer “discarded”. (Bug #14735917)
- Attempting to create an auto-increment column in an InnoDB table with a NULL type attribute could cause a serious error. (Bug #14758479)
- An assertion was raised if ALTER TABLE was used to rename a column to same name as an existing column while also reordering the renamed column using AFTER or FIRST. (Bug #14756089)

- An assertion could be raised if semi-join materialization was used to evaluate a `NOT IN` subquery. (Bug #14751858)

- For some continuation handler nestings, continuation could occur at the wrong location. (Bug #14724836)

- `SHOW PROCESSLIST` output was not sorted in `Id` order. (Bug #14771006)

- For some `SELECT` statements, `EXPLAIN` could cause the server to exit. (Bug #14761894)

- A memory leak occurred for attempts to use `ALTER TABLE` to set a default value for a tiny, medium, or long `BLOB` or `TEXT` column. (Bug #14756206)

- Installation using Solaris packages ran `mysql_install_db` during upgrade operations (this should occur only for new installations). (Bug #14747671, Bug #16534721)

- For `UPDATE` statements, `EXPLAIN` showed the total key length in the `key_len` column rather than the length of the used key parts. (Bug #14682438)

- Starting the server with `--bind-address` and then setting `host_cache_size` to 0 could result in the server stopping for certain kinds of client connections. (Bug #14689561)

- With index condition pushdown enabled, the optimizer could produce incorrect results for derived tables. (Bug #14640176)

- The optimizer could incorrectly use a nonspatial index to optimize spatial operations, causing an assertion to be raised. (Bug #14600994)

- `SHOW PROFILE` could be used to cause excessive server memory consumption. (Bug #14629232)

- Several problems with `mysql_config_editor` were fixed:

  - There was no error message for write errors to the configuration file.

  - The `--all` option is not supported for the `remove` command, but there was no warning message for attempts to use `remove --all`.

  - The `--all` option is not supported for the `set` command, but there was no warning message for attempts to use `set --all`.

  In addition, the `--user`, `--password`, and `--host` options now are supported for the `remove` command. When present, the `remove` command removes only the requested values from the login path. If none of them is given, `remove` removes the entire `client` login path. For example, this command removes only the `user` value from the `client` login path rather than the entire `client` login path:

  ```
  mysql_config_editor remove --login-path=client --user
  ```

  (Bug #14505672, Bug #14545989, Bug #14545999)

- A `LIKE` pattern with too many `'%'` wildcards could cause a segmentation fault. (Bug #14303860)

- Query rewriting to scrub passwords for logging was done even if none of the associated logs were enabled. Also, `CREATE SERVER` and `ALTER SERVER` are now rewritten as necessary. (Bug #14073554)

- Previously, the Performance Schema `events_statements_summary_by_digest` table was a summary grouped by the `DIGEST` column alone. Now this table contains a `SCHEMA_NAME` column and the digest summary is grouped by the `SCHEMA_NAME` and `DIGEST` columns.

  If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this change into the `performance_schema` database. (Bug #14075527)

- CHECK TABLE and REPAIR TABLE could crash if a MyISAM table had a corrupt key (.MYI) file. Now the server produces an error. (Bug #13556441)

- CHECK TABLE and REPAIR TABLE could crash if a MyISAM table had a corrupt key (.MYI) file. Now the server produces an error. (Bug #13556107, Bug #13556000)

- A “buffer too small” error message from the myisamchk command referred to the myisam_sort_buffer_size configuration option, when it should have referred to sort_buffer_size.

  myisamchk now has a myisam_sort_buffer_size variable available as an alternative name to sort_buffer_size. myisam_sort_buffer_size is preferable to sort_buffer_size because its name corresponds to the myisam_sort_buffer_size server system variable that has a similar meaning. sort_buffer_size should be considered deprecated. (Bug #11754894, Bug #46578)

- For dumps of the mysql database, mysqldump skipped the event table unless the --events option was given. This no longer occurs. To skip the event table if that is desired, use the --ignore-table option instead (Bug #55587, Bug #11762933)

- mysqld_safe ignored the value of the UMASK environment variable, leading to behavior different from mysqld with respect to the access mode of created files. Now mysqld_safe (and mysqld_multi) attempt to approximate the same behavior as mysqld. (Bug #57406, Bug #11764559)

- On Mac OS X, reinitializing the query cache could cause the server to exit. Thanks to Davi Arnaut for the patch. (Bug #67156, Bug #14741880)

- For MEMORY tables with HASH indexes, DELETE sometimes failed to delete all applicable rows. (Bug #51763, Bug #11759445)

- The server failed to use the query cache for queries in which a database or table name contained special characters and the table storage engine was InnoDB. (Bug #64821, Bug #13919851)

- On Mac OS X, KILL could sometimes be unreliable. (Bug #37780, Bug #11748945)

- The Performance Schema host_cache table displayed some lines multiple times. This was not an issue with the host cache itself, only with the table that provides information about the cache contents. (Bug #67236, Bug #14764890)

Changes in MySQL 5.6.8 (2012-11-07)

Configuration Notes

- This release continues the process begun in MySQL 5.6.6 of making changes to the default values of server parameters. The motivation for these changes is to provide better out-of-box performance and to reduce the need for database administrators to change settings manually. These changes are subject to revision in future releases as we gain feedback.

  In some cases, a parameter has a different fixed default value. In other cases, the server autosizes a parameter at startup using a formula based on other related parameters or server host configuration, rather than using a fixed value. For example, the setting for host_cache_size is its previous default of 128, adjusted up by an amount proportional to the value of max_connections. The idea behind autosizing is that when the server has information available to make a decision about a parameter setting likely to be better than a fixed default, it will.

  The following table summarizes changes to defaults. For variables that are autosized, the main variable description provides additional detail about the sizing algorithm. See Server System Variables, and InnoDB Startup Options and System Variables. Any of these default settings can be overridden by specifying an explicit value at server startup.

Parameter	Old Default	New Default
host_cache_size	128	Autosized using max_connections
innodb_log_file_size	5MB	48MB
open_files_limit	0	Autosized using max_connections
performance_schema	OFF	ON
performance_schema_events_waits_history_long_size	10000	Autosized
performance_schema_events_waits_history_size	10	Autosize
performance_schema_max_cond_instances	1000	Autosize
performance_schema_max_file_instances	10000	Autosize
performance_schema_max_mutex_instances	1000000	Autosize
performance_schema_max_rwlock_instances	1000000	Autosize
performance_schema_max_table_handles	100000	Autosize
performance_schema_max_table_instances	50000	Autosize
performance_schema_max_thread_instances	1000	Autosize
query_cache_size	0	1M
query_cache_type	ON	OFF
table_definition_cache	400	Autosized using table_open_cache
table_open_cache	400	2000
thread_cache_size	0	Autosized using max_connections

Installation Notes

- On Unix platforms, mysql_install_db supports a new option, --random-passwords, that provides for more secure MySQL installation. Invoking mysql_install_db with --random-passwords causes it to perform the following actions in addition to its normal operation:

  - The installation process creates a random password, assigns it to the initial MySQL root accounts, and sets the “password expired” flag for those accounts.

  - The initial random root password is written to the .mysql_secret file in the directory named by the HOME environment variable. Depending on operating system, using a command such as sudo may cause the value of HOME to refer to the home directory of the root system user. .mysql_secret is created with mode 600 to be accessible only to the system user for whom it is created.

    If .mysql_secret already exists, the new password information is appended to it. Each password entry includes a timestamp so that in the event of multiple install operations it is possible to determine the password associated with each one.

  - No anonymous-user MySQL accounts are created.

  As a result of these actions, it is necessary after installation to start the server, connect as root using the password written to the .mysql_secret file, and to select a new root password. Until this is done, root cannot do anything else. This must be done for each root account you intend to use. To change the password, you can use the SET PASSWORD statement (for example, with the mysql client). You can also use mysqladmin or mysql_secure_installation.

  New RPM install operations (not upgrades) invoke mysql_install_db with the --random-passwords option. As a consequence, RPM installs from this version onward will have their root accounts secured, and will have no anonymous-user accounts. (Install operations using RPMs for Unbreakable Linux Network are unaffected because they do not use mysql_install_db.)

For install operations using a binary .tar.gz distribution or a source distribution, you can invoke mysql_install_db with the --random-passwords option manually to make your MySQL installation more secure. This is recommended, particularly for sites with sensitive data.

- On Unix platforms, mysql_install_db now creates a default option file named my.cnf in the base installation directory. This file is created from a template included in the distribution package named my-default.cnf. You can find the template in or under the base installation directory. When started using mysqld_safe, the server uses my.cnf file by default. If my.cnf already exists, mysql_install_db assumes it to be in use and writes a new file named my-new.cnf instead.

  With one exception, the settings in the default option file are commented and have no effect. The exception is that the file changes the sql_mode system variable from its default of NO_ENGINE_SUBSTITUTION to also include STRICT_TRANS_TABLES:

  ```
  sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
  ```

  This setting produces a server configuration that results in errors rather than warnings for bad data in operations that modify transactional tables. See Server SQL Modes.

  The my-default.cnf template replaces the older sample option files (my-small.cnf, my-medium.cnf, and so forth), which are no longer distributed.

- mysql_install_db is now a Perl script and can be used on any system with Perl installed. Previously, it was a shell script and available only on Unix platforms.

  In addition, mysql_install_db is more strict about the --datadir option value. Only the last component of the path name is created if it does not exist; the parent directory must already exist or an error occurs. Previously, it created any nonexistent directories in the path name.

Functionality Added or Changed

- **InnoDB:** The InnoDB transportable tablespace feature was enhanced to allow ALTER TABLE ... IMPORT TABLESPACE to succeed in some cases where the corresponding .cfg file was not available. This enhancement allows recovery of data even in some cases where the system tablespace is corrupted or deleted. (Bug #14589582, Bug #66715)

- The number of atomic operations performed by the Performance Schema was reduced. (Bug #14658739)

- ALTER USER now can be used as a prepared statement. (Bug #66874, Bug #14646014)

- On Unix systems, the mysql client writes statements executed interactively to a history file (see mysql Logging). mysql now ignores for logging purposes statements that match any pattern in the “ignore” list. By default, the pattern list is "*IDENTIFIED*:*PASSWORD*", to ignore statements that refer to passwords. Pattern matching is not case sensitive. Within patterns, two characters are special:

  - ? matches any single character.

  - * matches any sequence of zero or more characters.

  To specify additional patterns, use the --histignore command option or set the MYSQL_HISTIGNORE environment variable. (If both are specified, the option value takes precedence.) The value should be a colon-separated list of one or more patterns, which are appended to the default pattern list.

  Patterns specified on the command line might need to be quoted or escaped to prevent your command interpreter from treating them specially. For example, to suppress logging for UPDATE and DELETE statements in addition to statements that refer to passwords, invoke mysql like this:

```
shell> mysql --histignore="*UPDATE*:*DELETE*"
```

(Bug #48287, Bug #11756377)

- On Windows, many MySQL executables depend on the libeay32.dll and ssleay32.dll SSL libraries at runtime. To ensure that the proper versions of these libraries are found, the install process copies them into the same directory as the executables.

- The SHOW AUTHORS and SHOW CONTRIBUTORS statements have been removed.

Bugs Fixed

- **Important Change; Replication:** When running the slave with the --slave-skip-errors option, successive skipped events (errors logged as warnings) were found to contain information from previous warnings, which caused an excessive amount of redundant information to be written to the error log. This problem could occur when using row-based or mixed-format binary logging.

  The fix for this issue is to clear these warnings prior to processing the next skipped event. In addition, the skipped events are now handled in the same way regardless of the value of binlog_format, and a skipped error always causes a warning to be written to the error log, as long as the value of the log_warnings system variable is greater than 1. (Bug #12776842)

- **Important Change:** The server system variables profiling, have_profiling, and profiling_history_size are now deprecated, and are subject to removal in a future release of the MySQL Server. (Bug #14658683)

- **InnoDB:** The auxiliary tables for FULLTEXT indexes were being created in the system tablespace, regardless of the setting for the innodb_file_per_table configuration option. (Bug #14723291)

- **InnoDB:** The thread gathering persistent statistics for an InnoDB table could cause a serious error if it accessed the table while a TRUNCATE TABLE operation was in progress:

  ```
  InnoDB: Assertion failure in thread thread_num in file fsp0fsp.cc line 1882
  ```

  (Bug #14765035)

- **InnoDB:** When a CREATE INDEX operation failed for an InnoDB FULLTEXT index due to a duplicate key error, some allocated memory was not freed. (Bug #14759111)

- **InnoDB:** An online DDL operation to create a unique index could fail to detect duplicate index values, when the duplicate values were caused by DML operations while the index was being created. (Bug #14733674)

- **InnoDB:** When using the transportable tablespace feature, the ALTER TABLE ... IMPORT TABLESPACE statement could crash if the InnoDB table being flushed contained a FULLTEXT index. With this fix, the table data can be imported, although you must drop and re-create the FULLTEXT index after the import operation. (Bug #14712962, Bug #67081)

- **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:** During a brief time window while creating an InnoDB unique index, MySQL could print a spurious warning message:

  ```
  WARNING: CANNOT FIND INDEX ?index_name IN INNODB INDEX TRANSLATION TABLE
  ```

  The cause was that MySQL started enforcing the uniqueness constraint before the existence of the index was fully registered. The fix suppresses the incorrect message during this final stage of index creation. (Bug #14735988)

- **InnoDB:** If a CREATE TABLE statement failed due to a disk full error, some memory allocated during the operation was not freed properly. (Bug #14708715)

- **InnoDB:** If creation of a FULLTEXT index failed because of a “row too large” condition, a subsequent ALTER TABLE operation could cause the server to halt with an error. (Bug #14668777)

- **InnoDB:** An assertion failure occurred when a bogus duplicate key error was flagged during online ALTER TABLE. This issue only occurred for a table that lacked a primary key and any secondary indexes. This patch fixes the assertion failure, but not the bogus duplicate key error, which is reported as Bug#14723456. (Bug #14712710)

- **InnoDB:** The InnoDB memcached plugin can now work with tables where the underlying character set is multibyte. (Bug #14711015, Bug #67076)

- **InnoDB:** An ALTER TABLE operation on an InnoDB table containing a FULLTEXT index could cause make the server halt with an assertion error. The fix causes all ALTER TABLE operations for such tables to use the table-copying behavior of the ALGORITHM=COPY clause. (Bug #14681198)

- **InnoDB:** If an InnoDB table containing a FULLTEXT index was being modified by a TRUNCATE TABLE statement and on online DDL operation simultaneously, the server could end up with inconsistent internal locks or could crash. (Bug #14676329)

- **InnoDB:** If the server crashed while executing TRUNCATE TABLE for an InnoDB table containing a FULLTEXT index, further errors could occur during crash recovery, preventing the server from restarting. (Bug #14676345)

- **InnoDB:** If the MySQL server crashed while XA transactions were in PREPARED state, inconsistent data could be produced during crash recovery if the query cache was enabled. The fix allows MySQL to disable the query cache during crash recovery if required. (Bug #14658648)

- **InnoDB:** With the innodb_file_per_table setting enabled, a DROP TABLE operation could cause a crash, due to a race condition that depended on the timing of pending I/O requests. (Bug #14594600, Bug #66718)

- **InnoDB:** If an online DDL operation failed due to a duplicate key error, caused by DML changes being made concurrently to the table, the server could crash with an assertion error. (Bug #14591797)

- **InnoDB:** If a FULLTEXT index was dropped from an InnoDB table, and the server crashed later for an unrelated reason, an additional error could occur while attempting to access nonexistent FULLTEXT data structures. (Bug #14586855)

- **InnoDB:** MySQL could crash while creating an InnoDB table if the disk became full at a specific moment: after the .frm file was created but before the corresponding .ibd file was created. (Bug #14645935)

- **InnoDB:** If the server crashed at the specific point when a change buffer entry was being merged into a buffer pool page, the transaction log and the change buffer were left in an inconsistent state. After a restart, MySQL could crash after reading the corresponding secondary index page. The problem was more likely to occur in MySQL 5.5 or later, where the original insert buffering mechanism was generalized to cover other operations. (Bug #14636528, Bug #66819, Bug #58571, Bug #61104, Bug #65443)

- **InnoDB:** If a crash occurred during a CREATE TABLE operation, the InnoDB data dictionary could be left in an inconsistent state, causing a crash if the partially created table was accessed later. (Bug #14601290)

- **InnoDB:** On startup, MySQL would not start if there was a mismatch between the value of the innodb_log_file_size configuration option and the actual size of the ib_logfile* files that make up the redo log. This behavior required manually removing the redo log files after changing the value of innodb_log_file_size. The fix causes MySQL to write all dirty pages to disk and re-create the redo log files during startup if it detects a size mismatch. (Bug #14596550)

- **InnoDB:** A query against an InnoDB table with a FULLTEXT index could crash, if the AGAINST clause contained a character sequence that was encoded incorrectly for the character set of the table. (Bug #14588091)
- **InnoDB:** The server could crash with a confusing message if it ran out of space for temporary files during index creation.

  ```
  InnoDB: Assertion failure in thread thread_num in file mtr0mtr.cc line 306
  InnoDB: Failing assertion: mtr->state == 12231
  ```

  (Bug #14586256)
- **InnoDB:** An ALTER TABLE on an InnoDB table that dropped the primary key and then re-created it with columns in a different order could cause an error. The issue affected tables where the swapped columns referenced each other in a single-table foreign key relationship. The data dictionary could be left in an inconsistent state, where the table was listed in SHOW TABLES output but could not be queried or dropped. For example, if the table was declared with primary key columns (c1,c2) and a foreign key with c1 REFERENCES c2:

  ```
  ALTER TABLE t2 DROP PRIMARY KEY, ADD PRIMARY KEY (c2, c1);
  ERROR 1030 (HY000): Got error 38 from storage engine
  ```

  (Bug #14548753)
- **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:** During an online DDL operation, a ROLLBACK affecting the same table could cause an assertion error if the table formerly contained a FULLTEXT index. Some bookkeeping information related to FULLTEXT indexes for InnoDB tables is preserved even after such an index is dropped. (Bug #14503700)
- **InnoDB:** Table names containing non-ASCII characters were displayed incorrectly when the MYSQL.INNODB_TABLE_STATS.TABLE_NAME column was queried. (Bug #14404879)
- **InnoDB:** A race condition could cause a crash during an online CREATE INDEX statement for an InnoDB table. This bug only affected very small tables. It required a DML operation to be in progress for the table, affecting the primary key columns, at the same time the CREATE INDEX statement was issued. (Bug #14117641)
- **InnoDB:** If a transaction was started with a consistent snapshot, then new indexes were added to the table while the transaction was in progress, a subsequent UPDATE statement could incorrectly encounter the error:

  ```
  ER_TABLE_DEF_CHANGED: insufficient history for index
  ```

  This issue could cause an assertion error in debug builds. (Bug #14036214)
- **InnoDB:** The server could crash with an assertion error during operations on tables with ROW_FORMAT=COMPRESSED. (Bug #14001972)
- **InnoDB:** In rare circumstances, during operations on an InnoDB table containing foreign keys, pages in the buffer pool could be evicted but not written to disk, leading to data inconsistency. (Bug #13688491)
- **InnoDB:** In rare circumstances, MySQL could apply InnoDB undo records out of order during a ROLLBACK of an operation that modified a BLOB column. This issue could cause an assertion error in debug builds:

  ```
  !bpage->file_page_was_freed
  ```

(Bug #13249921)

- **InnoDB:** In debug builds, a mismatch in the InnoDB PAGE_FREE list would cause an assertion. (Bug #12701488)

- **Partitioning:** The server now skips pruning of tables (see Partition Pruning) that use a storage engine which handles its own partitioning internally. The server now also explicitly rejects attempts to use explicit partitioning for such tables. (Bug #14672885)

- **Partitioning:** When used with a table having multiple columns in its primary key, but partitioned by KEY using a column that was not part of the primary key as the partitioning column, a query using an aggregate function and DISTINCT such as SELECT SUM(DISTINCT *pk_column_1*) FROM *table* WHERE *pk_column_2* = *constant* was not handled correctly. (Bug #14495351)

- **Replication:** When using a multi-threaded slave, if all worker threads were kept busy, it was possible for cleanup of an internal MTS circular buffer to fail, resulting in a full buffer and failure of the slave. (Bug #14710881)

- **Replication:** When invoked while gtid_mode was set to OFF, the SQL_THREAD_WAIT_AFTER_GTIDS() function waited indefinitely, unless a timeout was specified. In the latter case, the function could return incorrect values. Now, when gtid_mode is OFF, SQL_THREAD_WAIT_AFTER_GTIDS() always returns NULL, as expected. (Bug #14640065)

- **Replication:** Partially-failed GRANT and REVOKE statements were not always handled the same way on the master and the slave. We now log an incident event whenever an error occurs, even if it is only a partial error, with a message stating that manual reconciliation is required. (Bug #14598585)

- **Replication:** Executing FLUSH LOGS in parallel with COMMIT could cause the server to hang. (Bug #14640486)

- **Replication:** There existed a gap in time between the appending of the current GTID to the server's list of logged GTIDs and the commit of the transaction by the storage engine. On slow platforms, or when using profiling, this could cause SELECT SQL_THREAD_WAIT_AFTER_GTIDS(*gtid*) to return before the data actually reached the database.

  Now the current GTID is appended to the logged GTIDs following the commit, which removes this gap and so eliminates a possible source of inconsistency. (Bug #14116526)

- **Replication:** The error shown when a relay log file was missing from the relay log index file informed the user only that the log file was not found, but did not specify the exact reason. Now in such cases, the error message returned is Could not find target log file mentioned in relay log info in the index file '*index_file_name*' during relay log initialization. (Bug #11758505)

- **Replication:** Following an insert into a nontransactional table that failed due to insufficient disk space, the server did not properly clean up all pending events, leading to an assert or possibly to other errors. (Bug #11750014)

- **Replication:** Backtick (`) characters were not always handled correctly in internally generated SQL statements, which could sometimes lead to errors on replication slaves or cause failure of restore operations from binary log files. (Bug #66550, Bug #14548159, Bug #29422, Bug #11746883)

- A DELETE statement for an InnoDB table could write incorrect transaction metadata into a record, causing the server to halt with an error. To work around this issue, reduce the specified length of the primary key to less than 1K bytes. (Bug #14731482)

- For an in-place ALTER TABLE operation on an InnoDB table that produced a duplicate-key error for NULL values, the error message displayed the column default value rather than NULL. (Bug #14723364)

- mysql_secure_installation could not change the password for an account that had password_expired='Y' in the mysql.user table row for that account. (Bug #14726722)

- In debug builds, the server could crash because db_suicide() failed to handle SIGABRT signals. (Bug #14649493)

- With the optimizer tracing enabled, the INFORMATION_SCHEMA.OPTIMIZER_TRACE table can be queried to find tracing information about the last statements. However, for queries for which the results were retrieved from the query cache, this information was not available. (Bug #14665052)

- Patches for materialized semi-joins caused failures of the query plan interface used by NDBCLUSTER. (Bug #14704659)

- Queries that used a nested join with a subquery in the FROM clause and an ORDER BY ... DESC clause could return too few rows. (Bug #14678404)

- There was a performance regression for queries using SELECT ... INTO user variables and a WHERE condition on one or more of the variables in the INTO list. (Bug #14664077)

  References: This bug was introduced by Bug #12408412.

- mysqladmin password did not work for accounts with an expired password. (The fix for this problem is limited to accounts with passwords that use native or “old” native hashing. It still does not handle accounts that use SHA-256 password hashing.)

  As a consequence of this patch, the restricted mode of operation enforced by the server on operations permitted to clients with expired passwords now includes SET statements in addition to SET PASSWORD. This is useful if the account uses a password hashing format that requires old_passwords to be set to a value different from its default. (Bug #14698309)

- Repeated execution of a query containing a subquery that used MAX() could result in increasing memory consumption. (Bug #14683676)

- EXPLAIN DELETE ... WHERE *impossible_condition* could function incorrectly when it was used in a stored routine. (Bug #14601802)

  References: This bug was introduced by Bug #11752097.

- An incomplete result could be stored in the query cache when a query failed with an error (providing that the query cache was enabled, and was set to a nonzero size). This fix ensures that it is no longer possible for queries that finish with an error to be cached. (Bug #14621700)

  References: This bug was introduced by Bug #40264.

- USE *dbname* could fail with Unknown database when *dbname* contained multiple backtick (`) characters. (Bug #14645196)

- The configure.pl script that converts GNU configure options to CMake equivalents generated erroneous output for the --with-client-ldflags and --with-mysqld-ldflags options. It now ignores those options. (Bug #14593123)

- Attempts to insert, update, delete from, or lock unknown Performance Schema tables failed with an ER_TABLEACCESS_DENIED_ERROR error rather than ER_NO_SUCH_TABLE. (Bug #14633008)

- Outer joins could execute inefficiently and return incorrect results if joins were pushed down to the storage engine. (Bug #14644936)

- Index condition pushdown in conjunction with descending index range scan could return incorrect results if there were multiple ranges in the range scan. (Bug #14604223)

- The server could crash when registering tables in the query cache for queries that selected from views. (Bug #14619935)

- Small values of max_sort_length could produce incorrect results for integer, decimal, floating-point, or temporal data types. Now max_sort_length is ignored for those data types. (Bug #14596888)

- With semi-join and materialization optimizations enabled, a query that materialized a `const` table returned incorrect results when `STRAIGHT_JOIN` was added. (Bug #14609394)
- A prepared statement that referenced views in an `IN` subquery could return different results for different executions. (Bug #14641759)

  References: See also Bug #13773979.
- The thread cache implementation worked in LIFO rather than FIFO fashion and could result in a thread being denied service (although this was a remote possibility). (Bug #14621627)
- Within a stored program, memory allocated to hold condition information was not released until program exit, leading to excessive memory use. (Bug #14640599)
- In-source builds modified the source file `sql/share/dictionary.txt`. (Bug #14562699)
- The server printed excessive `Got error 159 when reading table` messages to the error log when one transaction attempted to access a table that had been modified by another. (Bug #14579877)
- Materialization of a subquery in the `FROM` clause could return the wrong number of rows if the subquery included a `LIMIT` clause. (Bug #14576727)
- The optimizer could choose an incorrect execution plan for updates to `InnoDB` tables based on indexes that use column prefixes. (Bug #14578060)
- A query with a subquery and `ORDER BY` and `LIMIT` clauses returned fewer rows than expected when executed using semi-join materialization. (Bug #14580874)
- Improper memory cleanup could cause the server to exit. (Bug #14536113)
- On Windows, `mysql_plugin` could not find `my_print_defaults`. (Bug #14471052)
- A query with a subquery in the `JOIN ... ON` clause with an outer reference to a field that was out of scope could cause the server to crash. (Bug #14498914)
- When used in `GRANT` statements, quoted user name or host name values containing leading or trailing spaces caused privileges to be assigned incorrectly until a `FLUSH PRIVILEGES` statement was issued.

  Now, as a result of this fix, quoted name and host identifiers used in a `GRANT` statement are automatically trimmed of any leading and trailing spaces, before privileges are assigned. (Bug #14328259)
- Granting or revoking the `PROXY` privilege caused the server to exit if the server was started with `--skip-name-resolve`. (Bug #14211140)
- `CREATE USER` and `DROP USER` could fail to flush the privileges, requiring `FLUSH PRIVILEGES` to be used explicitly. (Bug #13864642)
- On Mac OS X, the `version_compile_machine` system variable did not include the value `64` for server binaries compiled on a 64-bit system. (Bug #13859866)
- Access to `INFORMATION_SCHEMA` tables through a view could leak memory. (Bug #13734987)
- On Microsoft Windows with CMake 2.6, the build process would not stop if the `create_initial_db` step failed. (Bug #13713525)
- The test in `mysqld_safe` for the presence of the `--plugin_dir` option and assignment of a default value to it were performed before the actual argument parsing took place. (Bug #13548161)
- The number of connection errors from a given host as counted by the server was periodically reset, with the result that `max_connect_errors` was never reached and invalid hosts were never blocked from trying to connect. (Bug #11753779)

References: See also Bug #38247, Bug #43006, Bug #45584, Bug #45606.

- The `Range checked for each record` optimization is now used for conditions with outer query references. (Bug #11750963)

- A cached query result was not empty at the end of statement execution as expected. This could occur when executing queries (with the query cache enabled and set to a nonzero size) where the result was not sent to the client such as those executed by the Event Scheduler, or when executing stored routines containing queries while the server was running in bootstrap mode. (Bug #11755580, Bug #14609893)

- Metadata locking resulted in excessive contention in read-only workloads involving `InnoDB` tables and a low number of connections.

  Now the set of metadata locks can be partitioned into separate hashes to permit connections accessing different objects to use different locking hashes and reduce contention. The new `metadata_locks_hash_instances` system variable can be used to specify the number of hashes. (Bug #66473, Bug #14569140)

- `ST_Contains()` and `ST_Within()` incorrectly reported that a polygon did not contain itself. `ST_Equals()` incorrectly returned 0 for polygons that differed only in shifted vertices. (Bug #64653, Bug #13864679)

- `ST_Difference()` could incorrectly produce empty polygons in the result. (Bug #64649, Bug #13865773)

- For some queries involving `ORDER BY`, the optimizer chose the wrong index for accessing the table. (Bug #45969, Bug #11754370, Bug #14338686)

- On Windows, the Perl version of `mysql_install_db` created system tables in the `mysql` database that were not populated properly. (Bug #65584, Bug #14181049)

- Random number generation during client authentication consumed excessive CPU. (Bug #66567, Bug #14555434)

- `libmysqlclient` did not use symbol versioning. Thanks to Nicholas Bamber for the patch. (Bug #64386, Bug #13788218)

- The parser rejected legal queries that involved a `UNION` where the right hand side query term has a table in parenthese. (Bug #54382, Bug #11761854)

- In debug builds, `vio_read()` printed `errno` rather than `socket_error` to the debug trace. (Bug #28775, Bug #11746795)

Changes in MySQL 5.6.7 (2012-09-29, Release Candidate)

Beginning with MySQL 5.6.7, Oracle no longer provides binaries for Mac OS X 10.5, Debian 5, RHEL/OL 4, SLES 10, FreeBSD 7, Windows XP, or Windows 2003.

Configuration Notes

- This release continues the process begun in MySQL 5.6.6 of making changes to the default values of server parameters. The motivation for these changes is to provide better out-of-box performance and to reduce the need for database administrators to change settings manually. These changes are subject to revision in future releases as we gain feedback.

  The following table summarizes changes to defaults. Any of these default settings can be overridden by specifying an explicit value at server startup.

Parameter	Old Default	New Default
`innodb_data_file_path`	ibdata1:10M:autoextend	ibdata1:12M:autoextend

Functionality Added or Changed

- **Performance; InnoDB:** A new setting O_DIRECT_NO_FSYNC was added to the innodb_flush_method configuration option. This setting is similar to O_DIRECT, but omits the subsequent fsync() call. Suitable for some filesystems but not others. (Bug #11754304, Bug #45892)

- **Important Change; Partitioning:** The maximum number of partitions for a user-partitioned table is increased from 1024 to 8192. (Bug #11755685)

- **Important Change:** In MySQL 5.6.6, INSERT DELAYED was deprecated, to be removed in a future release. The same is now also true for DELAYED-related system variables delayed_insert_limit, delayed_insert_timeout, delayed_queue_size, max_delayed_threads, and max_insert_delayed_threads, and DELAYED-related status variables Delayed_errors, Delayed_insert_threads, Delayed_writes, and Not_flushed_delayed_rows.

- **InnoDB:** The --innodb-read-only option lets you run a MySQL server in read-only mode. You can access InnoDB tables on read-only media such as a DVD or CD, or set up a data warehouse with multiple instances all sharing the same data directory. See Configuring InnoDB for Read-Only Operation for usage details. (Bug #14143600)

- **InnoDB:** You can now select the compression level for InnoDB compressed tables, from the familiar range of 0-9 used by zlib. The compression level is controlled by the innodb_compression_level configuration option, with a default value of 6:

  - Increasing the compression level increases CPU overhead, possibly reducing the amount of storage needed for any particular row, reducing the possibility of a compression failure and subsequent page split.

  - Decreasing the compression level reduces CPU overhead, possibly increasing the amount of storage needed for any particular row, increasing the possibility of a compression failure and subsequent page split.

  You can also control whether compressed pages in the buffer pool are stored in the redo log when an update operation causes pages to be compressed again. This behavior is controlled by the innodb_log_compressed_pages configuration option. Turning off logging for compressed pages reduces the amount of redo data that is generated, possibly improving throughput. If the compressed page is required during crash recovery, it is compressed again at that time.

- **InnoDB:** Each data block in an InnoDB compressed table contains a certain amount of empty space (padding) to allow DML operations to modify the row data without re-compressing the new values. Too much padding can increase the chance of a compression failure, requiring a page split, when the data does need to be re-compressed after extensive changes. The amount of padding can now be adjusted dynamically, so that DBAs can reduce the rate of compression failures without re-creating the entire table with new parameters, or re-creating the entire instance with a different page size. The associated new configuration options are innodb_compression_failure_threshold_pct, innodb_compression_pad_pct_max

- **InnoDB:** New INFORMATION_SCHEMA tables, innodb_cmp_per_index and innodb_cmp_per_index_reset, provide statistics on InnoDB tables that use compression. The statistics at the index level let DBAs measure whether the proportion of successful or failed compression operations is acceptable for a particular combination of table, index, page size, and workload. Typically, the compression failure rate should be less than 10%, particularly when using a compressed table to handle an OLTP-style workload with frequent INSERT, UPDATE, or DELETE operations.

  Because gathering those statistics could be very time consuming and would hurt performance negatively, the new tables are enabled only when the new configuration option innodb_cmp_per_index_enabled is set to ON. (It is OFF by default.)

- When MySQL is configured with `-DWITH_SSL=system` to build with OpenSSL, CMake now produces an error if OpenSSL is older than version 1.0.1 (Bug #14167227)
- The default has changed from false to true for the `--secure-auth` option for `mysql` and the `MYSQL_SECURE_AUTH` option for the `mysql_options()` C API function. (Bug #13789417)
- The server now issues a `Note` diagnostic if an index is created that duplicates an existing index. (Bug #37520, Bug #11748842)
- The `WITH_SSL` option for CMake now accepts a *`path_name`* value that indicates the path name to the OpenSSL installation to use. This can be useful instead of a value of `system` when the CMake code detects an older or incorrect installed OpenSSL version. (Another permitted way to do the same thing is to set the `CMAKE_PREFIX_PATH` option to *`path_name`*.) (Bug #61619, Bug #12762891)
- 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.
- The MySQL client library now includes SSL support built in. When linking MySQL client programs, you should no longer specify either `-lssl` or `-lcrypto`.

  References: See also Bug #12762891, Bug #14167227.
- The unused `multi_range_count` system variable is now deprecated, and will be removed in a future release.
- 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 `SHOW PROFILE` and `SHOW PROFILES` statements. Use the Performance Schema instead; see MySQL Performance Schema.
  - The unused `date_format` `datetime_format` `time_format`, and `max_tmp_tables` system variables.
  - The obsolete `mysql.host` table. New MySQL 5.6 installations will no longer create this table. For upgrades, `mysql_upgrade` will check for this table and issue a warning about it being deprecated if it is nonempty.
  - The (undocumented) `--plugin-`*`xxx`* syntax for controlling plugin option *`xxx`*.

Bugs Fixed

- **Performance; InnoDB:** This fix improves the performance of the `InnoDB` `memcached` plugin in several ways:
  - A background thread periodically commits changes made to the database through `memcached` API calls. This commit interval based on time rather than number of

operations lets you increase the value of daemon_memcached_w_batch_size and daemon_memcached_r_batch_size without the risk of some changes remaining uncommitted when DML activity is infrequent. You can control the frequency of these automatic commits through the innodb_api_bk_commit_interval configuration option.

- When binary log support is enabled through the innodb_api_enable_binlog configuration option, you can increase the value of daemon_memcached_w_batch_size higher than the default of 1, allowing several DML operations to be committed together rather than a separate commit for each one.

- Internally, the efficiency of mutexes and table opening/closing was improved for operations involving the memcached plugin.

(Bug #14252821)

- **Performance; InnoDB:** The OPTIMIZE TABLE statement now updates the InnoDB persistent statistics for that table when appropriate. (Bug #14238097)

- **Performance; InnoDB:** This fix removes redundant checksum validation on InnoDB pages. The checksum was being verified both when a compressed page was read from disk and when it was uncompressed. Now the verification is only performed when the page is read from disk. (Bug #14212892, Bug #64170)

- **Performance; Replication:** On Solaris systems, enabling slave_parallel_workers could lead to a slowdown in event executions on the slave. (Bug #14641110)

  References: See also Bug #13897025.

- **Performance; Replication:** When slave_parallel_workers was enabled, an internal multiplier representing the number of events above a certain overrun level in the worker queue was never reset to zero, even when the excess had been taken care of; this caused the multiplier to grow without interruption over time, leading to a slowdown in event executions on the slave. (Bug #13897025)

- **Performance:** View definitions (in .frm files) were not cached and thus every access to a view involved a file read. Definitions now are cached for better performance. (Bug #13819275)

- **Performance:** Certain instances of subquery materialization could lead to poor performance. Subquery materialization now is chosen only if it is less costly than the EXISTS transformation. (See Optimizing Subqueries with Subquery Materialization, and Optimizing Subqueries with EXISTS Strategy.)

  This fix introduces a new flag for the optimizer_switch system variable named subquery_materialization_cost_based. If the flag is on (the default), the optimizer performs a cost-based choice between subquery materialization and IN -> EXISTS subquery transformation if either method could be used. If the flag is off, the optimizer chooses subquery materialization over IN -> EXISTS subquery transformation, which was the previous behavior. (Bug #13111584)

- **Incompatible Change:** Using ALTER TABLE to change the definition of a foreign key column could cause a loss of referential integrity. For example, changing a foreign key column that contained NULL values to be NOT NULL caused the NULL values to be the empty string. Similarly, an ALTER TABLE IGNORE that removed rows in a parent table could break referential integrity.

  The server now prohibits changes to foreign key columns with the potential to cause loss of referential integrity. A workaround is to use ALTER TABLE ... DROP FOREIGN KEY before changing the column definition and ALTER TABLE ... ADD FOREIGN KEY afterward. (Bug #46599, Bug #11754911)

- **Important Change; Replication:** When issued during an ongoing transaction, any of the following statements that are used to control MySQL Replication now cause the transaction to be committed:

- CHANGE MASTER TO
- START SLAVE
- STOP SLAVE
- RESET SLAVE

  For more information, see Statements That Cause an Implicit Commit. (Bug #13858841)

  References: See also Bug #14298750, Bug #13627921.

- **Important Change:** The ALTER USER statement cleared the user password in the mysql.user table. It no longer does this. (Bug #14226518)
- **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; Partitioning:** A SELECT from a partitioned InnoDB table having no primary key sometimes failed to return any rows where a nonempty result was expected. In such cases the server also returned the error Can't find record in *table_name* or Incorrect key file for table *table_name*. (Bug #13947868)
- **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:** On Windows systems, a file access error due to an incorrect value for MYSQL_DATADIR could cause an InnoDB assertion error. The error could persist after restarting MySQL. (Bug #14558324)
- **InnoDB:** The default for the innodb_checksum_algorithm, which was briefly changed to crc32 during the MySQL 5.6 development cycle, was switched back to innodb for improved compatibility of InnoDB data files during a downgrade to an earlier MySQL version. (Bug #14525151)
- **InnoDB:** When configuring the InnoDB memached plugin system table, INNODB_MEMCACHE.CONTAINERS, a comma (“,”) and empty space are used as a delimiter for mapping multiple columns to a memcached value. This fix allows the pipe character, (“|”), to also be used as a delimiter. (Bug #14560228)
- **InnoDB:** In an ALTER TABLE that rebuilds a table, and in particular, ADD COLUMN, DROP COLUMN, there were some assertion failures related to FULLTEXT indexes, particularly for tables containing more than one FULLTEXT index. The fix makes the ALTER TABLE correctly use or not use online DDL depending on the presence of FULLTEXT indexes. If a table had a FULLTEXT index that was dropped, any restrictions on online DDL for that table remain, due to the hidden FTS_DOC_ID column. (Bug #14488218)
- **InnoDB:** Under certain conditions, the innodb_io_capacity_max configuration option now uses the following formula to calculate a default value:

  ```
  innodb_io_capacity_max = max(2000, innodb_io_capacity * 2)
  ```

  The formula only takes affect when you specify a value for innodb_io_capacity at server startup and do not specify a value for innodb_io_capacity_max. The formula is not used when setting a value for innodb_io_capacity dynamically using a SET statement. (Bug #14469086)
- **InnoDB:** The syntax ALTER TABLE ... DROP FOREIGN KEY ... ALGORITHM=COPY incorrectly considered the names of foreign keys to be case-sensitive. (Bug #14394071)

- **InnoDB:** Under heavy load of concurrent DML and queries, an InnoDB table with a unique index could return nonexistent duplicate rows to a query. (Bug #14399148, Bug #66134)
- **InnoDB:** When an error (such as a duplicate key error) was detected during an online DDL operation, while applying changes made to the table while an index was being built, MySQL could encounter an assertion error if the same ALTER TABLE statement also contained any DROP INDEX clauses. (Bug #14392805)
- **InnoDB:** ALTER TABLE statements for partitioned tables could cause unnecessary locking and undo information. As part of the new online DDL feature, MySQL minimizes this overhead when practical, or you can specify the ALGORITHM=INPLACE clause on the ALTER TABLE statement. (Bug #14322667)
- **InnoDB:** A heavy query workload against an InnoDB table with a FULLTEXT index could cause a crash. The issue only occurred with some number of queries per second and some number of concurrent connections. (Bug #14347352)
- **InnoDB:** When an InnoDB table had a system-chosen primary key, based on a unique index on non-nullable columns, an error was issued if one of the primary key columns was altered to be nullable. The message was:

  ```
  Warning 1082 InnoDB: Table table_name has a primary key in InnoDB data
  dictionary, but not in MySQL!
  ```

  This issue only affected ALTER TABLE statements using the online DDL mechanism, that is, with the ALGORITHM=INPLACE clause specified or implied. (Bug #14353985)
- **InnoDB:** If an online CREATE INDEX operation failed, there was a brief period of time when concurrent DML operations could fail because the table was considered to be in an error state. (Bug #14341099)
- **InnoDB:** The mysql_install_db command could crash with an assertion error:

  ```
  InnoDB: Assertion failure in thread thread_num in file trx0rseg.cc line 326
  ```

  The size of the InnoDB system tablespace was being capped at 10MB, but during the 5.6 development cycle, the minimum size of a system tablespace became slightly larger than 10MB. (Bug #14315223)
- **InnoDB:** The server could crash if a read-only transaction was killed in a session that contained an InnoDB temporary table. (Bug #14213784)
- **InnoDB:** When more than one InnoDB temporary table was created and accessed within the same transaction, queries on those temporary tables could fail with an ER_TABLE_DEF_CHANGED error. (Bug #14234581)
- **InnoDB:** This fix addresses several issues regarding AUTO_INCREMENT columns when adding a column using online DDL (that is, with ALGORITHM=INPLACE). Now the AUTO_INCREMENT_OFFSET value is used properly, the calculation for the next value is corrected, FLOAT, DOUBLE, and unsigned INTEGER auto-increment values are handled correctly, and overflow conditions are detected. (Bug #14219624)
- **InnoDB:** A SHOW ENGINE...STATUS command could crash if an XA transaction was created using the statement START TRANSACTION READ ONLY. (Bug #14218867)
- **InnoDB:** This fix prevents online DDL operations from conflicting with foreign key operations happening simultaneously on the same table. Updates or deletes based on CASCADE or SET NULL clauses in the foreign key definition are blocked while the online DDL is in progress, because the information needed in case of a ROLLBACK would not be available after the ALTER TABLE statement completes. (Bug #14219233)

- **InnoDB:** A race condition could cause assertion errors during a DROP TABLE statement for an InnoDB table. Some internal InnoDB functions did not correctly determine if a tablespace was missing; other functions did not handle the error code correctly if a tablespace was missing. (Bug #14251529)
- **InnoDB:** With the MySQL 5.6 online DDL feature, an ALTER TABLE statement to add a primary key to an InnoDB table could succeed, even though the primary key columns contained duplicate values. (Bug #14219515)
- **InnoDB:** The server could crash with a combination of a transaction with SERIALIZABLE isolation level, FLUSH TABLES ... WITH READ LOCK, and a subsequent query. The error message was:

  ```
  InnoDB: Failing assertion: prebuilt->stored_select_lock_type != LOCK_NONE_UNSET
  ```

  (Bug #14222066)
- **InnoDB:** When an auto-increment column used a FLOAT or DOUBLE data type, if the auto-increment value became very large (larger than the maximum unsigned long long value), subsequent inserts could fail or cause the server to halt. (Bug #14145950, Bug #55071)
- **InnoDB:** An online DDL operations to add a foreign key could incorrectly leave some memory allocated if the DDL encountered an error. (Bug #14156259)
- **InnoDB:** An INSERT into a table after a failed online DDL operation could cause an erroneous assertion error:

  ```
  InnoDB: Failing assertion: prebuilt->trx_id == 0 || prebuilt->trx_id <=
  last_index->trx_id
  ```

  (Bug #14176821)
- **InnoDB:** The server could hang at startup, during crash recovery, if the rollback of previously active transactions conflicted with the dropping of temporary tables. With this fix, persistent statistics do not apply to InnoDB temporary tables. (Bug #14175080)
- **InnoDB:** The configuration option innodb_max_io_capacity was renamed to innodb_io_capacity_max, to emphasize its relationship to the existing innodb_io_capacity option. (Bug #14175020)
- **InnoDB:** The server could crash with a signal 8 (division by zero error) due to a race condition while computing index statistics. (Bug #14150372)
- **InnoDB:** Deleting from an InnoDB table containing a prefix index, and subsequently dropping the index, could cause a crash with an assertion error. (Bug #13807811)
- **InnoDB:** The value of the NUMBER_PAGES_CREATED and NUMBER_PAGES_WRITTEN columns of the INFORMATION_SCHEMA.INNODB_BUFFER_POOL_STATS table were set to incorrect values, and the NUMBER_PAGES_GET column was not being set at all. (Bug #13639187)
- **InnoDB:** The error message was improved for the case where an UPDATE failed because the row included several BLOB values greater than 768 bytes each, causing the size of a row to exceed half the page size. The old message, was misleading; it suggested using BLOBs, when the 768-byte prefix for each BLOB column was the cause of the limit error:

  ```
  Error Code 1118: Row size too large. The maximum row size for the used table
  type, not counting BLOBs, is 8126. You have to change some columns to TEXT or
  BLOBs
  ```

A workaround for the problem was to create the table with the ROW_FORMAT=DYNAMIC or ROW_FORMAT=COMPRESSED clause, which is now suggested in the message. (Bug #13453036, Bug #63507)

- **InnoDB:** The server could crash when updating very large BLOB values, typically 16MB or more. (Bug #13450566)

- **InnoDB:** A problem in the locking mechanism could cause a serious error with queries using the HANDLER statement. (Bug #11766271, Bug #59344)

- **InnoDB:** When a SELECT ... FOR UPDATE, UPDATE, or other SQL statement scanned rows in an InnoDB table using a < or <= operator in a WHERE clause, the next row after the affected range could also be locked. This issue could cause a lock wait timeout for a row that was not expected to be locked. The issue occurred under various isolation levels, such as READ COMMITTED and REPEATABLE READ. (Bug #11765218)

- **InnoDB:** The new online DDL feature addressed long-standing bugs where ALTER TABLE statements caused table rebuilds unnecessarily. This particular bug applied to changing default values for TIMESTAMP columns. (Bug #11753646, Bug #45124)

- **InnoDB:** Various inconsistent behaviors, including tables becoming inaccessible, were cleaned up for ALTER TABLE statements involving InnoDB tables involved in foreign key relationships. (Bug #11744929, Bug #5670)

- **Partitioning:** For tables using PARTITION BY HASH or PARTITION BY KEY, when the partition pruning mechanism encountered a multi-range list or inequality using a column from the partitioning key, it continued with the next partitioning column and tried to use it for pruning, even if the previous column could not be used. This caused partitions which possibly matched one or more of the previous partitioning columns to be pruned away, leaving partitions that matched only the last column of the partitioning key.

  This issue was triggered when both of the following conditions were met:

  1. The columns making up the table's partitioning key were used in the same order as in the partitioning key definition by a SELECT statement's WHERE clause as in the column definitions;

  2. The WHERE condition used with the last column of the partitioning key was satisfied only by a single value, while the condition testing some previous column from the partitioning key was satisfied by a range of values.

  An example of a statement creating a partitioned table and a query against this for which the issue described above occurred is shown here:

```
CREATE TABLE t1 (
  c1 INT,
  c2 INT,
  PRIMARY KEY(c2, c1)
) PARTITION BY KEY()  # Use primary key as partitioning key
  PARTITIONS 2;

SELECT * FROM t1 WHERE c2 = 2 AND c1 <> 2;
```

  This issue is resolved by ensuring that partition pruning skips any remaining partitioning key columns once a partition key column that cannot be used in pruning is encountered. (Bug #14342883)

- **Partitioning:** The buffer for the row currently read from each partition used for sorted reads was allocated on open and freed only when the partitioning handler was closed or destroyed. For SELECT statements on tables with many partitions and large rows, this could cause the server to use excessive amounts of memory.

This issue has been addressed by allocating buffers for reads from partitioned tables only when they are needed and freeing them immediately once they are no longer needed. As part of this fix, memory is now allocated for reading from rows only in partitions that have not been pruned (see Partition Pruning). (Bug #13025132)

References: See also Bug #11764622, Bug #14537277.

- **Replication; Microsoft Windows:** On 64-bit Windows platforms, values greater than 4G for the max_binlog_cache_size and max_binlog_stmt_cache_size system variables were truncated to 4G. This caused LOAD DATA INFILE to fail when trying to load a file larger than 4G in size, even when max_binlog_cache_size was set to a value greater than this. (Bug #13961678)
- **Replication:** Updates writing user variables whose values were never set on a slave while using --replicate-ignore-table could cause the slave to fail. (Bug #14597605)

  References: This bug was introduced by Bug #14275000.
- **Replication:** When the number of multi-threaded slave workers (as determined by setting the slave_parallel_workers server system variable) was changed when using relay_log_info_repository=TABLE, the mysql.slave_worker_info table did not reflect the change. (Bug #14550905)

  References: See also Bug #13804728, Bug #14550945, Bug #14578740.
- **Replication:** With relay_log_info_repository=FILE and slave_parallel_workers greater than 0, changing the relay log info repository type to TABLE and restarting the slave mysqld caused a subsequent START SLAVE statement to crash the slave. (Bug #14550945)

  References: See also Bug #13804728, Bug #14550905, Bug #14578740.
- **Replication:** A manually created file named slave_worker_info in the MySQL Server's data directory could be mistaken for the actual relay log info file. In addition, when the number of workers (slave_parallel_workers server system variable) was decreased, the corresponding info files were not removed as expected. (Bug #14578740)

  References: See also Bug #13804728, Bug #14550905, Bug #14550945.
- **Replication:** Executing the SQL_THREAD_WAIT_AFTER_GTIDS() function without binary logging enabled could cause the server to crash. (Bug #14457883)
- **Replication:** Using COM_BINLOG_DUMP_GTID with incorrect data could cause the server to crash. (Bug #14509140)
- **Replication:** An internal routine in the MySQL Replication code removed elements from a hash used to store a mapping between databases and worker threads at the same time that the hash was being iterated over. This could cause an unintended reordering of the has elements and thus possibly to incorrect results from routines using this hash. (Bug #14381701)

  References: See also Bug #13864642.
- **Replication:** The names of the binary log and relay log Performance Schema mutexes were mistakenly changed to names that differed from the MySQL 5.5 names. The names have been reverted to those used in MySQL 5.5. (Bug #14366314)
- **Replication:** When setting up replication between a master and a slave which was using --master-info-repository=TABLE, the mysql.slave_master_info table was not updated the first time that START SLAVE was issued. (Bug #14298750)

  References: See also Bug #13858841.
- **Replication:** The --disable-gtid-unsafe-statements option caused any nontransactional DML statement involving temporary tables to be rejected with an error even with binlog_format

set explicitly to ROW, in spite of the fact that they are not written to the binary log in this case. Now, such statements are allowed when using row-based logging, as long as any nontransactional tables affected by the statements are also temporary tables. (Bug #14272627)

- **Replication:** When using multi-threaded slaves, --replicate-rewrite-db rules were not honored while assigning databases to slave worker threads, which could cause statements to be executed out of order when this option was used. This could result in a slave that was inconsistent with the master. (Bug #14232958)
- **Replication:** mysql_upgrade failed when the server was running with gtid_mode=ON and --disable-gtid-unsafe-statements because the MySQL system tables are stored using MyISAM. This problem is fixed by changing the default logging behavior for mysql_upgrade; logging is now disabled by default. (Actions taken by mysql_upgrade depend on the server version, and thus should not be replicated to slaves.) To enable logging, you can execute mysql_upgrade using the --write-binlog option. (Bug #14221043, Bug #13833710)
- **Replication:** The initialization and usage of a number of internal programming objects relating to GTIDs did not work properly with PERFORMANCE_SCHEMA. (Bug #14152637)
- **Replication:** The scheduler for multi-threaded slaves did not take into account databases implicitly involved in operations through foreign key dependencies, which could lead to a temporary loss of consistency on the slave. To avoid this problem, replication events on the master that invoke foreign key relationships between table is different databases are now marked in such a way that they can be scheduled sequentially to avoid race conditions and thereby inconsistency. However, this can adversely affect performance. (Bug #14092635)
- **Replication:** When using a multi-threaded slave, the repository type employed for the relay log info log was not always used automatically for worker repositories as expected. (Bug #13804728)

  References: See also Bug #14550905, Bug #14550945, Bug #14578740.
- **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:** The Master_id column of the mysql.slave_master_info and mysql.slave_relay_log_info tables showed the slave's server ID instead of the master's server ID. (Bug #12344346)
- **Replication:** Statements such as UPDATE ... WHERE *primary_key_column* = *constant* LIMIT 1 are flagged as unsafe for statement-based logging, despite the fact that such statements are actually safe. In cases where a great many such statements were run, this could lead to disk space becoming exhausted do to the number of such false warnings being logged. To prevent this from happening, a warning suppression mechanism is introduced. This warning suppression acts as follows: Whenever the 50 most recent ER_BINLOG_UNSAFE_STATEMENT warnings have been generated more than 50 times in any 50-second period, warning suppression is enabled. When activated, this causes such warnings not to be written to the error log; instead, for each 50 warnings of this type, a note is written to the error log stating The last warning was repeated *N* times in last *S* seconds. This continues as long as the 50 most recent such warnings were issued in 50 seconds or less; once the number of warnings has decreased below this threshold, the warnings are once again logged normally.

  The fix for this issue does not affect how these warnings are reported to MySQL clients; a warning is still sent to the client for each statement that generates the warning. This fix also does not make any changes in how the safety of any statement for statement-based logging is determined. (Bug #11759333, Bug #51638)

  References: See also Bug #11751521, Bug #42415.
- In-place ALTER TABLE operations for InnoDB tables could raise an assertion attempting to acquire a lock. (Bug #14516798)

- ALTER TABLE ... DROP FOREIGN KEY that did not name the foreign key to be dropped caused a server crash. Now the foreign key name is required. (Bug #14530380)

- In mysql_com.h, the CLIENT_CONNECT_ATTRS and CLIENT_PLUGIN_AUTH_LENENC_CLIENT_DATA symbols incorrectly were defined as the same value. (Bug #14482472)

- With a password policy of STRONG and a password of 100 characters or more, VALIDATE_PASSWORD_STRENGTH() could cause a server crash. (Bug #14458293)

- PASSWORD(NULL) and OLD_PASSWORD(NULL) could cause a server crash. (Bug #14458217)

- Polygons with holes could cause a server crash for spatial operations. (Bug #14497827)

- The Threads_running status variable was not updated properly. (Bug #14471011)

- mysql_secure_installation did not work if old_passwords was set to 2 (use the sha256_password authentication plugin). (Bug #14506073)

- GROUP_CONCAT() with DISTINCT or ORDER BY on GEOMETRY values caused a server crash. (Bug #14468106)

- For complex conditions, the optimizer could produce an incorrect range construction and return incorrect query results. (Bug #14497598)

- Item_cache_str::save_in_field() dereferenced a null pointer if the cached value was NULL. (Bug #14501403)

- The optimizer could raise an assertion when grouping and sorting in descending order on an indexed column. (Bug #14498999)

- A query with GROUP BY ... WITH ROLLUP comparing a grouping column using the IN operator caused an assertion to be raised. (Bug #14500792)

- In debug builds, with semi-join enabled, GROUP BY ... WITH ROLLUP that did not use a temporary table could cause a server crash. (Bug #14499409)

- An assertion was raised when using the join cache for a query that contained an IN subquery query with a subquery that is expected to return a single row but returned more than one. (Bug #14499331)

- Index condition pushdown in conjunction with descending index range scan caused a performance regression. (Bug #14503142)

- The explicit_defaults_for_timestamp system variable was not visible (for example, with SHOW VARIABLES), so it was not possible to make runtime decisions based on its value. (Bug #14409088)

- An ALTER TABLE for an InnoDB table that attempted to add an index and also change the nullability of a column participating in that index raised an assertion. (Bug #14404635)

- When resolving outer fields, Item_field::fix_outer_fields() creates new Item_refs 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)

- The --server-public-key option for mysql and mysqltest has been renamed to --server-public-key-path to reflect that it refers to a file and for consistency with related server-side variable naming. Also, this option now is available only if MySQL was built with OpenSSL (not yaSSL) because yaSSL does not support the necessary RSA encryption. (Bug #14348721)

- The result set could contain extra rows for queries on MyISAM tables that used the SQL_BUFFER_RESULT modifier and a subquery. (Bug #14348858)

- For debug builds, if one session used a DDL statement to alter an `InnoDB` table, another session could raise an assertion failure if it had a pre-alter consistent snapshot of the table. (Bug #14365043)
- The Performance Schema used listed the nanosecond timer by default for stages and statements in the `setup_timers` table. But if this timer was not available on a given platform (such as Windows), timing for stages and statements failed to work. Now the idle, stage, and statement timers used the preferred timers if they are available, but alternate timers if not. (Bug #14298586)
- The Performance Schema generated different digests for a statement before and after selecting a database. (Bug #14256311)
- The RPM spec file now also runs the test suite on the new binaries, before packaging them. (Bug #14318456)
- Inside a stored program, references to stored program variables in XML functions such as `ExtractValue()` failed after the first execution of the stored program. (Bug #14317442)
- A LooseScan semi-join could return duplicate rows from the outer table. (Bug #14271594)
- Some queries for which the optimizer used index condition pushdown in conjunction with `ref` access could be very slow if the index was read in descending order. (Bug #14287654, Bug #14503142)
- Queries executed using MaterializeScan semi-join strategy and a materialized subquery could return too many rows. (Bug #14272788)
- `ALTER SERVER`, `CREATE SERVER`, and `DROP SERVER` with an empty server name caused a server crash. (Bug #14220942)
- The server did not build with `gcc` 4.7. (Bug #14238406)
- If a call to `socket()` failed, the Performance Schema created instrumentation for it anyway. (Bug #14209598)
- `REQUIRE ISSUER` clauses for `GRANT` statements were not rewritten properly for logging and caused a server crash. (Bug #14211069)
- For queries that used range access, the optimizer could read uninitialized data, resulting in Valgrind warnings. (Bug #14200538)
- The Performance Schema digest-generation code could fail with a race condition. (Bug #14250296)
- With semi-join optimization enabled, subqueries in the `WITH CHECK OPTION` clause of view definitions were evaluated incorrectly. (Bug #14230177)
- `WEIGHT_STRING()` could crash if given a bad flags argument. (Bug #14211236)
- An optimizer trace could crash attempting to print freed subquery items. (Bug #14238404)
- `mysql_upgrade` did not set the `STATS_PERSISTENT=0` table option for `InnoDB` tables in the `mysql` database. (Bug #14195056)
- `ALTER TABLE` with `DISCARD TABLESPACE` or `IMPORT TABLESPACE` did not acquire a sufficiently strong metadata lock to prevent a concurrent `ALTER TABLE` statement with `ADD` or `DROP` from modifying the tablespace. This could result in warnings or raise an assertion. (Bug #14213236)
- Some queries with a `HAVING` clause with a function that referred to a function in the `WHERE` list with a subquery as parameter caused an assertion to be raised. (Bug #14209318)
- String allocation could cause Valgrind warnings. (Bug #14201818)
- For JSON-format `EXPLAIN` statements, derived tables were not handled properly and caused a server crash. (Bug #14167499)
- Incorrect internal conversion of string-format dates could cause a server crash. (Bug #14167911)

- In debug builds, comparisons for strings that had the `ucs2_unicode_520_ci` collation could raise an assertion. (Bug #14161973)
- Join processing could attempt to clean up a temporary table that had not been instantiated, causing a server crash. (Bug #14168270)
- In debug builds, the optimizer raised an unnecessary (too strict) assertion about `MyISAM` key lengths. (Bug #14179461)
- In-place `ALTER TABLE` did not work for a table with a `GEOMETRY` column, even if the alteration did not involve that column. (Bug #14140927)
- Negative values could be erroneously reported for some columns in the `buffer_pool_pages_in_flush` row in the `information_schema.innodb_metrics` table. (Bug #14090287)
- Improper error handling for `CREATE SERVER`, `DROP SERVER`, and `ALTER SERVER` could raise an assertion. (Bug #14061851)
- `RELEASE SAVEPOINT` did not have sufficient checks for the XA transaction state to prevent a savepoint from being released while the transaction was in a prepared state. (Bug #14062726)
- The FirstMatch strategy for semi-joins produced incorrect results for some queries with multiple inner tables. (Bug #14081638)
- With materialization and semi-joins enabled, some queries with an OR condition could produce incorrect results. (Bug #14075016)
- In-place `ALTER TABLE` did not handle autopartitioning storage engines such as `NDB`. (Bug #14063233)
- `JSON`-format `EXPLAIN` statements could raise an assertion or cause the server to hang for statements with an impossible-`WHERE` clause and subqueries in `ORDER BY` or `GROUP BY` clauses. (Bug #14084642)
- For nonexistent files, the Performance Schema file I/O instrumentation sometimes did extra work or was subject to instrumentation leaks. (Bug #14113704)
- Within a trigger, references to a temporary table used during the query execution process could end up pointing to nonexisting fields on subsequent executions, causing a server crash. (Bug #14105951)
- Small `sort_buffer_size` values could result in a server crash. (Bug #14111180)
- Improper initialization by spatial functions could cause a server crash the first time they were invoked following server startup. (Bug #14015762)
- Polygon sorting by spatial functions could be done incorrectly and cause a server crash. (Bug #13938850)
- For JSON-format `EXPLAIN` statements, improper handling of subqueries could cause an assertion to be raised. (Bug #13956275)
- `SELECT` on a partitioned table that used a join buffer could cause a server crash. (Bug #13949549)
- The argument for `LIMIT` must be an integer, but if the argument was given by a placeholder in a prepared statement, the server did not reject noninteger values such as `'5'`. (Bug #13868860)
- For `DELETE` statements, `WHERE` clause row retrieval that should access only the index tree could raise an assertion. (Bug #13919180)
- Some arguments could cause `ST_Buffer()` to crash. (Bug #13832749, Bug #13833019)
- Queries that used the `ST_Contains` and `Within()` functions yielded incorrect results when argument columns had a spatial index. (Bug #13813064)

- CHECK TABLE and REPAIR TABLE could crash if a key definition differed in the .frm and .MYI files of a MyISAM table. Now the server produces an error. (Bug #13555854)
- The optimizer used a full index scan for cases for which a loose index scan was preferable. (Bug #13464493)

  References: This bug is a regression of Bug #12540545.
- COUNT(DISTINCT(SELECT 1)) could be evaluated incorrectly if the optimizer used Loose Index Scan. (Bug #13444084)

  References: See also Bug #13813126.
- A query for a FEDERATED table could return incorrect results when the underlying table had a compound index on two columns and the query included an AND condition on the columns. (Bug #12876932)
- mysqlhotcopy failed for databases containing views. (Bug #62472, Bug #13006947, Bug #12992993)
- “Illegal mix of collation” errors were returned for some operations between strings that should have been legal. (Bug #64555, Bug #13812875)
- mysqldump could dump views and the tables on which they depend in such an order that errors occurred when the dump file was reloaded. (Bug #44939, Bug #11753490)
- The ST_Contains() and Within() functions yielded an incorrect result when used on a column with a SPATIAL index. (Bug #65348, Bug #14096685)
- Adding a LIMIT clause to a query containing GROUP BY and ORDER BY could cause the optimizer to choose an incorrect index for processing the query, and return more rows than required. (Bug #54599, Bug #11762052)
- If the server was started with secure_auth disabled, it did not produce a warning that this is a deprecated setting. (Bug #65462, Bug #14136937)
- The GeomFromWKB() function did not return NULL if the SRID argument was NULL, and non-NULL SRID values were not included in the converted result. (Bug #65094, Bug #13998446)
- With statement-based binary logging, stored routines that accessed but did not modify tables took too strong a lock for the tables, unnecessarily blocking other statements that also accessed those tables. (Bug #62540, Bug #13036505)
- For some queries, the optimizer used index_merge access method when this was more work than ref access. (Bug #65274, Bug #14120360)
- In prepared statements, MYSQL_TYPE_DATE parameters when converted to an integer were handled as MYSQL_TYPE_DATETIME values and the conversion produced incorrect results. (Bug #64667, Bug #13904869)
- mysqlbinlog did not accept input on the standard input when the standard input was a pipe. (Bug #49336, Bug #11757312)
- The argument to the --ssl-key option was not verified to exist and be a valid key. The resulting connection used SSL, but the key was not used. (Bug #62743, Bug #13115401)
- In-place ALTER TABLE incorrectly handled indexes using key prefixes by using a copy algorithm. (Bug #65865, Bug #14304973)
- COUNT(DISTINCT(IF ...)) could be evaluated incorrectly if the optimizer used Loose Index Scan. (Bug #64445, Bug #13813126)

  References: See also Bug #13444084.

- Starting the server with `--bind-address=*` is supposed to cause the server to accept TCP/IP connections on all server host IPv6 and IPv4 interfaces if the server host supports IPv6, or TCP/IP connections on all IPv4 addresses otherwise. But the server sometimes did not correctly detect when IPv6 was not supported, and failed to start. (Bug #66303, Bug #14483430)
- Internal temporary `MyISAM` tables were unnecessarily registered in an open-table list protected by a global mutex, causing excessive mutex contention. (Bug #65077, Bug #14000697)
- Queries with `ALL` over a `UNION` could return an incorrect result if the `UNION` result contained `NULL`. (Bug #65902, Bug #14329235)
- There was a performance regression for queries that used `GROUP BY` and `COUNT(DISTINCT)`. (Bug #49111, Bug #11757108)
- In debug builds, an `InnoDB` assertion was overly aggressive about prohibiting an open range. (Bug #66513, Bug #14547952)

Changes in MySQL 5.6.6 (2012-08-07, Milestone 9)

> **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.

Binary Logging

- **Performance:** The server now implements group commit for the binary log: Multiple commits are grouped in memory, then written and flushed to disk as a group rather than individually. This reduces the number of writes and flushes, improving performance of binary logging. Group commit works for all storage engines. `InnoDB` implements some optimizations to take advantage of group commit capability.

  These system variables were added in conjunction with group commit:

  - `binlog_order_commits`: Whether to commit transactions in the same order they are written to the binary log or permit them to be committed in parallel.
  - `binlog_max_flush_queue_time`: How long in microseconds to keep reading transactions from the flush queue before proceeding with the group commit.
  - `innodb_flush_log_at_timeout`: Write and flush logs every *`N`* seconds.

Configuration Notes

- This release of MySQL implements changes to the default values of several server parameters. The motivation for these changes is to provide better out-of-box performance and to reduce the need for database administrators to change settings manually. These changes are subject to revision in future releases as we gain feedback. (See Changes to Server Defaults.)

  In some cases, a parameter has a different fixed default value. In other cases, the server autosizes a parameter at startup using a formula based on other related parameters or server host configuration, rather than using a fixed value. For example, the setting for `back_log` is its previous default of 50, adjusted up by an amount proportional to the value of `max_connections`. The idea behind autosizing is that when the server has information available to make a decision about a parameter setting likely to be better than a fixed default, it will.

  The following table summarizes changes to defaults. For variables that are autosized, the main variable description provides additional detail about the sizing algorithm. See Server System

Variables, and InnoDB Startup Options and System Variables. Any of these default settings can be overridden by specifying an explicit value at server startup.

Parameter	Old Default	New Default
back_log	50	Autosized using max_connections
binlog_checksum	NONE	CRC32
--binlog-row-event-max-size	1024	8192
flush_time	1800 (on Windows)	0
innodb_autoextend_increment	8	64
innodb_buffer_pool_instances	1	8 (platform dependent)
innodb_checksum_algorithm	INNODB	CRC32
innodb_concurrency_tickets	500	5000
innodb_file_per_table	0	1
innodb_old_blocks_time	0	1000
innodb_open_files	300	Autosized using innodb_file_per_table, table_open_cache
innodb_stats_on_metadata	ON	OFF
join_buffer_size	128KB	256KB
max_allowed_packet	1MB	4MB
max_connect_errors	10	100
performance_schema	OFF	ON
performance_schema_events_waits_history_long_size	10000	Autosized
performance_schema_events_waits_history_size	10	Autosized
performance_schema_max_cond_instances	1000	Autosized
performance_schema_max_file_instances	10000	Autosized
performance_schema_max_mutex_instances	1000000	Autosized
performance_schema_max_rwlock_instances	1000000	Autosized
performance_schema_max_table_handles	100000	Autosized
performance_schema_max_table_instances	50000	Autosized
performance_schema_max_thread_instances	1000	Autosized
sql_mode	'' (empty string)	NO_ENGINE_SUBSTITUTION
sync_master_info	0	10000
sync_relay_log	0	10000
sync_relay_log_info	0	10000

With regard to compatibility with previous releases, the most important changes are:

- innodb_file_per_table is enabled (previously disabled)
- innodb_checksum_algorithm is CRC32 (previously INNODB)
- binlog_checksum is CRC32 (previously NONE)

Therefore, if you are upgrading an existing MySQL installation, have not already changed the values of these parameters from their previous defaults, and backward compatibility is a concern, you may want to explicitly set these parameters to their previous defaults. For example, put these lines in the server option file:

```
[mysqld]
innodb_file_per_table=0
innodb_checksum_algorithm=INNODB
binlog_checksum=NONE
```

Those settings preserve compatibility as follows:

- With the new default of `innodb_file_per_table` enabled, `ALTER TABLE` operations following an upgrade will move `InnoDB` tables that are in the system tablespace to individual `.ibd` files. Using `innodb_file_per_table=0` will prevent this from happening.

- Setting `innodb_checksum_algorithm=INNODB` permits binary downgrades after upgrading to this release. With a setting of `CRC32`, InnoDB would use checksumming that older MySQL versions cannot use.

- With `binlog_checksum=NONE`, the server can be used as a replication master without causing failure of older slaves that do not understand binary log checksums.

Performance Schema Notes

- The Performance Schema is now enabled by default (the `performance_schema` system variable is enabled by default). To disable it, set `performance_schema=off` at server startup.

  In addition, the Performance Schema now automatically sizes the values of several of its parameters at server startup if they are not set explicitly. For example, it sizes the parameters that control the sizes of the events waits tables this way. To see which parameters are sized under this policy, use `mysqld --verbose --help` and look for those with a default value of –1, or see Performance Schema System Variables.

  For each autosized parameter that is not set at server startup (or is set to –1), the Performance Schema determines how to set its value based on the value of the following system values, which are considered as “hints” about how you have configured your MySQL server:

  ```
  max_connections
  open_files_limit
  table_definition_cache
  table_open_cache
  ```

  To override autosizing for a given parameter, set it a value other than –1 at startup. In this case, the Performance Schema assigns it the specified value.

  At runtime, `SHOW VARIABLES` displays the actual values that autosized parameters were set to.

  If the Performance Schema is disabled, its autosized parameters remain set to –1 and `SHOW VARIABLES` displays –1.

Security Notes

- These security improvements were implemented:

  - MySQL now provides a method for storing authentication credentials encrypted in an option file named `.mylogin.cnf`. To create the file, use the `mysql_config_editor` utility. The file can be read later by MySQL client programs to obtain authentication credentials for connecting to a MySQL server. `mysql_config_editor` writes the `.mylogin.cnf` file using encryption so the credentials are not stored as clear text, and its contents when decrypted by client programs are used only in memory. In this way, passwords can be stored in a file in non-cleartext format and used later without ever needing to be exposed on the command line or in an environment variable. For more information, see mysql_config_editor — MySQL Configuration Utility.

The .mylogin.cnf file can contain multiple sets of options, known as “login paths.” This makes it easy to set up multiple “personalities” for connecting to different MySQL servers. Any of these can be selected by name later using the --login-path option when you invoke a client program. See Command-Line Options that Affect Option-File Handling.

- MySQL now supports stronger encryption for user account passwords, available through an authentication plugin named sha256_password that implements SHA-256 password hashing. This plugin is built in, so it is always available and need not be loaded explicitly. For more information, including instructions for creating accounts that use SHA-256 passwords, see The SHA-256 Authentication Plugin.

  Other changes associated with the introduction of the sha256_password plugin:

  - The old_passwords system variable previously permitted values of 1 or 0 to control whether “old” or “new” MySQL native password hashing was used by the CREATE USER and GRANT statements and the PASSWORD() function. Now old_passwords permits a value of 2 to select use of SHA-256 password hashing.

    Note

    Previously, old_passwords permitted values of OFF or ON as synonyms for 0 or 1. That is no longer true.

  - SHA-256 password hashing (old_passwords=2) uses a random salt value, which makes the result from PASSWORD() nondeterministic. Consequently, statements that use this function are no longer safe for statement-based replication and cannot be stored in the query cache.

  - If MySQL is built with OpenSSL, RSA encryption can be used to transmit passwords during the client connection process. The sha256_password_private_key_path and sha256_password_public_key_path system variables permit the private and public key files to be named on the server side. The Rsa_public_key status variable displays the public key value. The mysql and mysqltest clients support a --server-public-key option permitting the public key file to be specified explicitly when connecting to the server. (This option is implemented through a new MYSQL_SERVER_PUBLIC_KEY option to the mysql_options() C API function.)

  MySQL Connector support: Connectors that use the C client library should work with sha256_password with no changes. Connectors that implement the authentication process for themselves must be updated to account for changes in the client/server protocol.

- The server now has a --default-authentication-plugin option to specify the default plugin to associate with new accounts for which no plugin is named explicitly. Permitted values are mysql_native_password (use MySQL native passwords; this is the default value) and sha256_password (use SHA-256 passwords). This option also changes the initial old_passwords value to be consistent with the password hashing method required by the default plugin, if necessary.

  Note

  If you use this option to change the default authentication method to a value other than mysql_native_password, clients older than MySQL 5.5.7 will no longer be able to connect because they will not understand the change to the authentication protocol.

- The mysql.user table now has a password_expired column to enable DBAs to expire account passwords and require users to reset their password. The default password_expired value is 'N', but can be set to 'Y' with the new ALTER USER statement. After an account's password has been expired, all operations performed by the account in subsequent connections to the server result in an error until the user issues a SET PASSWORD statement to establish a new account password. For more information, see ALTER USER Syntax, and Password Expiration and Sandbox Mode.

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

> ⚠ **Caution**
>
> **Update:** ALTER USER also set the Password column to the empty string, so do not use this statement in 5.6.6. This problem has been fixed in MySQL 5.6.7.

- MySQL now has provision for checking password security:

  - In statements that assign a password supplied as a cleartext value, the value is checked against the current password policy and rejected if it is weak (the statement returns an ER_NOT_VALID_PASSWORD error). This affects the CREATE USER, GRANT, and SET PASSWORD statements. Passwords given as arguments to the PASSWORD() and OLD_PASSWORD() functions are checked as well.

  - The strength of potential passwords can be assessed using the new VALIDATE_PASSWORD_STRENGTH() SQL function, which takes a password argument and returns an integer from 0 (weak) to 100 (strong).

  Both capabilities are implemented by the validate_password plugin. If the plugin is not installed, the affected statements and PASSWORD() and OLD_PASSWORD() work as before (no password checking), and VALIDATE_PASSWORD_STRENGTH() always returns 0.

  The validate_password plugin also implements a set of system variables corresponding to the parameters that control password checking. If the plugin is installed, you can modify these variables to configure the password policy.

  The validate_password plugin is written using the MySQL plugin API, which has been extended to support writing password-validation plugins.

  For more information, see The Password Validation Plugin. For information about writing password-checking plugins, see Writing Password-Validation Plugins.

- mysql_upgrade now produces a warning if it finds user accounts with passwords hashed with the older pre-4.1 hashing method. Such accounts should be updated to use more secure password hashing. See Password Hashing in MySQL

(Bug #65461, Bug #14136939)

Functionality Added or Changed

- **Performance; InnoDB:** Many DDL operations on InnoDB tables can now be performed online, without making the tables unavailable for queries. Some operations, such as creating or dropping indexes, even allow DML statements (INSERT, UPDATE, DELETE) on the table while the operation is in progress. A single online DDL operation can also take the place of a sequence of statements, such as several DROP INDEX statements, ALTER TABLE ... ADD COLUMN, and then several CREATE INDEX statements. See InnoDB and Online DDL for full details.

  An additional effect of this change occurs for consistent-read transactions that try to reread data from a table which was changed by ALTER TABLE in another session. Instead of receiving an empty set, the transaction will receive an error (ER_TABLE_DEF_CHANGED, “Table definition has changed, please retry transaction”). (Bug #58368, Bug #11765404, Bug #11872643, Bug #12325508, Bug #11765266, Bug #60689)

- **Performance; InnoDB:** The MySQL server now includes the widely used `memcached` in-memory caching system, and a plugin that allows fast NoSQL-style access to `InnoDB` tables through the `memcached` protocol. This access method avoids the overhead of SQL parsing and constructing a query optimization plan. You can store the underlying data in a single `InnoDB` table, or spread it across multiple tables. You can read and write data through both `memcached` and SQL. For example, you can do fast single-key lookups through `memcached get` calls, and do statistical reports across all the data through SQL.

  Several configuration options let you fine-tune this system, in particular to balance raw performance against durability and consistency of data. The main new configuration options are `daemon_memcached_option`, `daemon_memcached_r_batch_size`, `daemon_memcached_w_batch_size`, `innodb_api_trx_level`, `innodb_api_enable_mdl`, and `innodb_api_enable_binlog`.

  See `InnoDB` Integration with memcached for full details.

- **Performance; InnoDB:** The persistent statistics feature for `InnoDB` tables is now enabled by default, and can be controlled at the level of individual tables. This feature involves the configuration options `innodb_stats_persistent`, `innodb_stats_auto_recalc`, and `innodb_stats_persistent_sample_pages`, and the clauses `STATS_PERSISTENT`, `STATS_AUTO_RECALC`, and `STATS_SAMPLE_PAGES` of the `CREATE TABLE` and `ALTER TABLE` statements. See Configuring Persistent Optimizer Statistics Parameters for usage details.

- **Incompatible Change:** The `--safe-mode` server option has been removed.

- **Incompatible Change:** It is now explicitly disallowed to assign the value `DEFAULT` to stored procedure or function parameters or stored program local variables (for example with a `SET` *`var_name`* `= DEFAULT` statement). This was not previously supported, or documented as permitted, but is flagged as an incompatible change in case existing code inadvertantly used this construct. It remains permissible to assign `DEFAULT` to system variables, as before, but assigning `DEFAULT` to parameters or local variables now results in a syntax error.

  After an upgrade to MySQL 5.6.6 or later, existing stored programs that use this construct produce a syntax error when invoked. If a `mysqldump` file from 5.6.5 or earlier is loaded into 5.6.6 or later, the load operation fails and affected stored program definitions must be changed.

- **Important Change; Partitioning:** MySQL nows supports *partition lock pruning*, which allows for many DDL and DML statements against partitioned tables using `MyISAM` (or another storage engine that employs table-level locking) to lock only those partitions directly affected by the statement. These statements include (but are not limited to) many `SELECT`, `SELECT ... PARTITION`, `UPDATE`, `REPLACE`, `INSERT`, and other statements. This enhancement improves especially the performance of many such statements when used with tables having many (32 or more) partitions. For a complete list of affected statements with particulars, and other information, see Partitioning and Locking. (Bug #37252, Bug #11748732)

- **Important Change; Replication:** It is now possible, in the event that a multi-threaded slave fails while running with the `--relay-log-recovery` option, to switch it safely to single-threaded mode despite the presence of any gaps with unprocessed transactions in the relay log. To accomplish this, you can now use `START SLAVE [SQL_THREAD] UNTIL SQL_AFTER_MTS_GAPS` to cause the slave SQL threads to run until no more such gaps are found in the relay log. Once this statement has completed, you can change the `slave_parallel_workers` system variable, and (if necessary) issue a `CHANGE MASTER TO` statement before restarting the slave. (Bug #13893363)

  References: See also Bug #13893310.

- **Important Change; Replication:** `INSERT ON DUPLICATE KEY UPDATE` is now marked as unsafe for statement-based replication if the target table has more than one primary or unique key. For more information, see Determination of Safe and Unsafe Statements in Binary Logging. (Bug #58637, Bug #11765650, Bug #13038678)

- **Important Change; Replication:** The `SHOW BINARY LOGS` statement (and its equivalent `SHOW MASTER LOGS`) may now be executed by a user with the `REPLICATION CLIENT` privilege. (Formerly, the `SUPER` privilege was necessary to use either form of this statement.)

- **Important Change:** `INSERT DELAYED` is now deprecated, and will be removed in a future release. Use `INSERT` (without `DELAYED`) instead. (Bug #13985071)

- **Important Change:** In MySQL, the `TIMESTAMP` data type differs in nonstandard ways from other data types:

  - `TIMESTAMP` columns not explicitly declared with the `NULL` attribute are assigned the `NOT NULL` attribute. (Columns of other data types, if not explicitly declared as `NOT NULL`, permit `NULL` values.) Setting such a column to `NULL` sets it to the current timestamp.

  - The first `TIMESTAMP` column in a table, if not declared with the `NULL` attribute or an explicit `DEFAULT` or `ON UPDATE` clause, is automatically assigned the `DEFAULT CURRENT_TIMESTAMP` and `ON UPDATE CURRENT_TIMESTAMP` attributes.

  - `TIMESTAMP` columns following the first one, if not declared with the `NULL` attribute or an explicit `DEFAULT` clause, are automatically assigned `DEFAULT '0000-00-00 00:00:00'` (the “zero” timestamp). For inserted rows that specify no explicit value for such a column, the column is assigned `'0000-00-00 00:00:00'` and no warning occurs.

  Those nonstandard behaviors remain the default for `TIMESTAMP` but now are deprecated and this warning appears at startup:

  ```
  [Warning] TIMESTAMP with implicit DEFAULT value is deprecated.
  Please use --explicit_defaults_for_timestamp server option (see
  documentation for more details).
  ```

  As indicated by the warning, to turn off the nonstandard behaviors, enable the new `explicit_defaults_for_timestamp` system variable at server startup. With this variable enabled, the server handles `TIMESTAMP` as follows instead:

  - `TIMESTAMP` columns not explicitly declared as `NOT NULL` permit `NULL` values. Setting such a column to `NULL` sets it to `NULL`, not the current timestamp.

  - No `TIMESTAMP` column is assigned the `DEFAULT CURRENT_TIMESTAMP` or `ON UPDATE CURRENT_TIMESTAMP` attributes automatically. Those attributes must be explicitly specified.

  - `TIMESTAMP` columns declared as `NOT NULL` and without an explicit `DEFAULT` clause are treated as having no default value. For inserted rows that specify no explicit value for such a column, the result depends on the SQL mode. If strict SQL mode is enabled, an error occurs. If strict SQL mode is not enabled, the column is assigned the implicit default of `'0000-00-00 00:00:00'` and a warning occurs. This is similar to how MySQL treats other temporal types such as `DATETIME`.

  To upgrade servers used for replication, upgrade the slaves first, then the master. Replication between the master and its slaves should work provided that all use the same value of `explicit_defaults_for_timestamp`:

  1. Bring down the slaves, upgrade them, configure them with the desired value of `explicit_defaults_for_timestamp`, and bring them back up.

     The slaves will recognize from the format of the binary logs received from the master that the master is older (predates the introduction of `explicit_defaults_for_timestamp`) and that operations on `TIMESTAMP` columns coming from the master use the old `TIMESTAMP` behavior.

2. Bring down the master, upgrade it, and configure it with the same `explicit_defaults_for_timestamp` value used on the slaves, and bring it back up.

(Bug #63034, Bug #13344629, Bug #55131, Bug #11762529)

- **Important Change:** The `YEAR(2)` data type is now deprecated because it is problematic. `YEAR(2)` columns in existing tables are treated as before, but `YEAR(2)` in new or altered tables are converted to `YEAR(4)`. Support for `YEAR(2)` will be removed entirely in a future release of MySQL. For more information, see `YEAR(2)` Limitations and Migrating to `YEAR(4)`.

- **InnoDB:** `InnoDB` now supports the `DATA DIRECTORY='directory'` clause of the `CREATE TABLE` statement, which allows you to create `InnoDB` file-per-table tablespaces (`.ibd` files) in a location outside the MySQL data directory.

  For additional information, see Specifying the Location of a Tablespace.

- **InnoDB:** `InnoDB` tables now support the notion of “transportable tablespaces”, allowing `.ibd` files to be exported from a running MySQL instance and imported into another running instance. The `FOR EXPORT` clause of the `FLUSH TABLE` command writes any unsaved changes from `InnoDB` memory buffers to the `.ibd` file. After copying the `.ibd` file and a separate metadata file to the other server, you can use the `DISCARD TABLESPACE` and `IMPORT TABLESPACE` clauses of the `ALTER TABLE` statement to bring the table data into a different MySQL instance.

  For more information, see Copying Tablespaces to Another Server (Transportable Tablespaces).

- **InnoDB:** For systems with constant heavy workloads, or workloads that fluctuate widely, several new configuration options let you fine-tune the flushing behavior for `InnoDB` tables: `innodb_adaptive_flushing_lwm`, `innodb_max_dirty_pages_pct_lwm`, `innodb_max_io_capacity` (changed in subsequent point releases to `innodb_io_capacity_max`), and `innodb_flushing_avg_loops`. These options feed into an improved formula used by the `innodb_adaptive_flushing` option. For full details about improvements to flushing algorithms and options, see Tuning `InnoDB` Buffer Pool Flushing.

- **Replication:** The `STOP SLAVE` option `SQL_BEFORE_GTIDS` did not function correctly, and the `SQL_AFTER_GTIDS` option for the same statement did not function at all. (Bug #13810456)

- **Replication:** Added the `--slave-rows-search-algorithms` option for `mysqld`, which determines the search algorithms used for finding matches for slave updates when `slave_allow_batching` is enabled, including whether or not table or index hashing is used with searches employing a primary or unique key, some other key, or no key.

- The Performance Schema has a new system variable, `performance_schema_session_connect_attrs_size`, and new status variable, `Performance_schema_session_connect_attrs_lost`. The system variable is the amount of preallocated memory per thread used to hold connection attribute strings. If the connection attribute strings are larger than the reserved storage, the status variable is incremented. (Bug #14076427)

- yaSSL was upgraded from version 1.7.2 to 2.1.4. (Bug #13713205)

  References: See also Bug #13706828.

- The generic “procedure API” has been removed from the server. This was formerly present as a means of writing server procedures, but went unused except for `PROCEDURE ANALYSE()`. Removing the interface simplifies aspects of the internal procedure representation that were related to code no longer in the server but had a negative effect on its operation, in the sense that these aspects hindered the ability of the optimizer to perform better on more common query types. In addition, this code hindered future optimizer development and its removal will have benefit that development.

  `PROCEDURE ANALYSE()` remains available, but is no longer implemented using a public interface. (For information, see Using `PROCEDURE ANALYSE`.) One consequence of removing the procedure interface is that EXPLAIN SELECT ... PROCEDURE ANALYSE() now works where previously it produced an error.

- To improve scalability by reducing contention among sessions for the global lock on the open tables cache, the cache now can be partitioned into several smaller cache instances. A session now need lock only one instance to access it for DML statements. This segments cache access among instances, permitting higher performance for operations that need to use the cache when many there are many sessions accessing tables. (DDL statements still require a lock on the entire cache, but such statements are much less frequent than DML statements.)

  A new system variable, table_open_cache_instances, permits control over the number of cache instances. Each instance has a size of table_open_cache / table_open_cache_instances. By default, the number of instances is 1.

  Three new status variables provide information about the operation of the open tables cache. Table_open_cache_hits and Table_open_cache_misses indicate the number of hits and misses or lookups in the cache. Table_open_cache_overflows indicates how many times, after a table is opened or closed, an instance has an unused entry and the size of the instance is larger than table_open_cache / table_open_cache_instances.

- Previously, for semi-join processing the outer query specification was limited to simple table scans or inner joins using comma syntax, and view references were not possible. Now outer join and inner join syntax is permitted in the outer query specification, and the restriction that table references must be base tables has been lifted.

- It is now possible for client programs to pass connection attributes to the server in the form of key/value pairs. Attributes are manipulated using the MYSQL_OPT_CONNECT_ATTR_RESET and MYSQL_OPT_CONNECT_ATTR_DELETE options for the mysql_options() C API function, and the MYSQL_OPT_CONNECT_ATTR_ADD option for the new mysql_options4() function. Connection attributes are exposed through the session_connect_attrs and session_account_connect_attrs Performance Schema tables.

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

  For more information, see C API Function Descriptions, and MySQL Performance Schema.

- Previously, the default value for the --bind-address option was 0.0.0.0, which causes the server to accept TCP/IP connections on all server host IPv4 interfaces. To make it easier to use IPv6 connections without special configuration, the default --bind-address value now is *. This is similar to 0.0.0.0, but causes the server to also accept TCP/IP connections on all IPv6 interfaces if the server host supports IPv6. (Another way to accept IPv4 and IPv6 connections is by using --bind-address=::, but in this case an error occurs if the server host does not support IPv6.)

- The optimizer's cost model for disk-sweep Multi-Read Range (DS-MRR) has been improved. The improved cost model makes it more likely that DSMRR will be used for queries that read much data from disk.

- For the WITH_SSL CMake option, no is no longer a permitted value or the default value. The default is now bundled. Consequently, MySQL now is always built with SSL support.

Bugs Fixed

- **Performance; InnoDB; Partitioning:** The statistics used by the optimizer for queries against partitioned InnoDB tables were based only on the first partition of each such table, leading to use of the wrong execution plan. (Bug #13694811)

  References: This bug was introduced by Bug #11756867.

- **Performance; InnoDB:** Improved the efficiency of the system calls to get the system time to record the start time for a transaction. This fix reduces potential cache coherency issues that affected performance. (Bug #13993661)
- **Performance; InnoDB:** Improved the efficiency of InnoDB code with regard to CPU cache coherency. (Bug #14034087)
- **Performance; InnoDB:** Improved the algorithm related to adaptive flushing. This fix increases the rate of flushing in cases where compression is used and the data set is larger than the buffer pool, leading to eviction. (Bug #13990648, Bug #65061)
- **Performance; InnoDB:** Improved the efficiency of the COMMIT operation for InnoDB tables, by reducing the potential for context switching and acquiring/re-acquiring mutexes while the operation is in progress. (Bug #13989037)
- **Performance; InnoDB:** The order in which flushes are performed when the innodb_flush_neighbors configuration option is enabled was improved. The algorithm makes the neighbor-flushing technique faster on HDD storage, while reducing the performance overhead on SSD storage. (innodb_flush_neighbors typically is not needed for SSD hardware.) (Bug #13798956)
- **Performance; InnoDB:** This fix improves the speed of DROP TABLE for InnoDB tables by removing a scan of the buffer pool to remove entries for the adaptive hash index. This improvement is most noticeable on systems with very large buffer pools and the innodb_adaptive_hash_index option enabled. (Bug #13704145, Bug #64284)
- **Performance; Replication:** All changes made as part of a given transaction are cached; when the transaction is committed, the contents of this cache are written to the binary log. When using global transaction identifiers, the GTID identifying this transaction must be the first event among all events in the cache belonging to the transaction.

  Previously, a portion of the cache was preallocated as a buffer when the transaction began; upon commit it was completed with a valid GTID. However, because it was not possible to perform a seek in the cache, it was necessary to flush it to a temporary file, and then seek within this file. When the cache buffer is not big enough to accommodate all changes comprising a given transaction, it swapped the data to disk, then reinitialized the cache to have the buffer properly filled with the correct data again. The buffer was actually flushed and the cache reinitialized every time a GTID event was written, even in those cases in which all events making up a given transaction fit within the cache buffer, which could negatively impact the performance of binary logging (and thus replication) when using GTIDs.

  Now the cache is reinitialized only when it is actually necessary—in other words, only when the cache is in fact swapped to disk.

  In addition, the fix for this issue addresses a missing unlock operation when the server failed to write an empty transaction group and reduces the amount of code needed for prepending the GTID to the contents of the cache before flushing the cache to disk. (Bug #13877432)

  References: See also Bug #13738296.
- **Performance:** Within stored programs, the overhead of making statements log friendly was incurred even when the corresponding log was not enabled. (Bug #12884336)
- **Performance:** The MD5() and SHA1() functions had excessive overhead for short strings. (Bug #49491, Bug #11757443, Bug #60227, Bug #14134662)
- **Incompatible Change:** Metadata was handled incorrectly for objects such as tables or views that were used in a stored program. Metadata for each such object was gathered at the beginning of program execution, but not updated if DDL statements modified the object during program execution (or modified it between executions of the program if the program remained in the stored program cache). This resulted in mismatches between the actual object structure and the structure the stored

program believed the object to have during execution, and caused problems such as data errors or server crashes.

Now metadata changes to objects used in a stored program are detected during execution and affected statements within the program are reparsed so that they use the updated metadata.

Example: Suppose that a stored program executes this statement in a loop and that the columns in the table t1 are altered during loop execution:

```
SELECT * FROM t1;
```

Previously, errors occurred because program execution did not detect that SELECT * evaluates to a different set of columns after the change. Now the table change is detected and the SELECT is reparsed to determine the new set of columns.

Reparsing occurs for other cases as well, such as t1 being changed from a base table to a view or a TEMPORARY table. For more information, see Caching of Prepared Statements and Stored Programs.

There is a possible incompatibility regarding the new behavior: Application code that assumed the previous behavior and implemented a workaround may need to be changed.

Other instances of corrected problems:

- SELECT * within a stored program could fail for TEMPORARY tables created within the program using prepared statements.
- “Unknown column” errors or bad data could result from changing the set of columns in a table used within a stored program between executions of the program or while the table was used within a program loop. Errors could also occur under similar circumstances for a view if the view definition was changed, for a TEMPORARY table if the table was dropped.
- Failure of triggers to notice metadata changes in objects accessed within the program could cause trigger malfunction.
- Failure of a stored program to notice metadata changes in objects accessed within the program could cause replication to fail.

(Bug #61434, Bug #12652835, Bug #55678, Bug #11763018, Bug #64574, Bug #13840615, Bug #33843, Bug #11747732, Bug #33289, Bug #11747626, Bug #33255, Bug #11747619, Bug #33000, Bug #11747566, Bug #27011, Bug #11746530, Bug #33083, Bug #11747581, Bug #32868, Bug #11747537, Bug #12257, Bug #11745236)

- **Important Change; MySQL Cluster:** mysqld_safe now traps Signal 13 (SIGPIPE) so that this signal no longer kills the MySQL server process. (Bug #33984)
- **InnoDB; Replication:** When binary log statements were replayed on the slave, the Com_insert, Com_update, and Com_delete counters were incremented by BEGIN statements initiating transactions affecting InnoDB tables but not by COMMIT statements ending such transactions. This affected these statements whether they were replicated or they were run using mysqlbinlog. (Bug #12662190)
- **InnoDB:** If MySQL crashed during an ALTER TABLE t DISCARD TABLESPACE operation, it could leave InnoDB in a state where it crashes at the next startup. The error message was:

  ```
  InnoDB: Error: a record lock wait happens in a dictionary operation!
  ```

  (Bug #14146981)
- **InnoDB:** When importing an InnoDB tablespace representing a compressed table, unnecessary checksum calculations were being performed. (Bug #14161424)

- **InnoDB:** Dropping an InnoDB temporary table could leave behind the .ibd file if the table was created with the innodb_file_per_table setting enabled. On Windows systems, this could cause an additional problem: repeated attempts to drop the file for 2000 seconds. In addition to resolving the incorrect path name used to drop the file, this fix also limits the retry loop to 10 seconds, for example if the file cannot be removed because it is locked by a backup process. (Bug #14169459)

- **InnoDB:** A race condition could cause a crash during an online CREATE INDEX statement for an InnoDB table. This bug only affected very small tables. It required a DML operation to be in progress for the table, affecting the primary key columns, at the same time the CREATE INDEX statement was issued. (Bug #14117641)

- **InnoDB:** If a row was deleted from an InnoDB table, then another row was re-inserted with the same primary key value, an attempt by a concurrent transaction to lock the row could succeed when it should have waited. This issue occurred if the locking select used a WHERE clause that performed an index scan using a secondary index. (Bug #14100254, Bug #65389)

- **InnoDB:** An assertion error could occur if an XA transaction was created within a session designated as read-only. (Bug #14108709)

- **InnoDB:** This fix improves the accuracy of the data in the INFORMATION_SCHEMA table innodb_metrics for systems with innodb_buffer_pool_instances set to greater than 1. The improved information applies to the number of pages flushed from the buffer pool, specifically these entries in the table:

  ```
  buffer_flush_batch_total_pages
  buffer_flush_neighbor_total_pages
  buffer_flush_adaptive_total_pages
  buffer_flush_sync_total_pages
  buffer_flush_background_total_pages
  buffer_LRU_batch_total_pages
  ```

  (Bug #14037167)

- **InnoDB:** In a transaction using the REPEATABLE READ isolation level, an UPDATE or DELETE statement for an InnoDB table could sometimes overlook rows recently committed by other transactions. As explained in Consistent Nonlocking Reads, DML statements within a REPEATABLE READ transaction apply to rows committed by other transactions, even if a query could not see those rows. (Bug #14007649, Bug #65111)

- **InnoDB:** During an ANALYZE TABLE statement for an InnoDB table, the server could hang (in non-debug builds), or an assertion error could occur, indicating recursive acquisition of a lock (in debug builds). (Bug #14007109)

- **InnoDB:** Using the KILL statement to terminate a query could cause an unnecessary message in the error log:

  ```
  [ERROR] Got error -1 when reading table table_name
  ```

  (Bug #13933132)

- **InnoDB:** Querying the INFORMATION_SCHEMA.INNODB_TRX or related tables while the server was running a heavy InnoDB workload could cause a crash, with messages in the error log referring to the function fetch_data_into_cache_low. This issue arose during new feature work and only affected MySQL 5.6. (Bug #13966453)

- **InnoDB:** Fixes a recently introduced issue with InnoDB persistent statistics, that could cause a crash (non-debug builds) or assertion error (debug builds). (Bug #13946118)

- **InnoDB:** Including a % character in a query using an InnoDB FULLTEXT index could cause a crash. (FULLTEXT indexes for InnoDB tables are a new feature, still under development.) (Bug #13940669, Bug #64901)

- **InnoDB:** An assertion could be raised if an InnoDB table was moved to a different database using ALTER TABLE ... RENAME while the database was being dropped by DROP DATABASE. (Bug #13982017)

- **InnoDB:** If the server crashed while dropping an InnoDB temporary table or an index on a temporary table, further errors could occur during crash recovery, preventing the server from restarting. (Bug #13913670)

- **InnoDB:** A FULLTEXT query for an InnoDB table could filter the search terms incorrectly if a term using the minus operator was followed by another term using the plus operator. (Bug #13907075)

- **InnoDB:** When a table was renamed, the InnoDB persistent statistics were not associated with the new table name. (Bug #13920437)

- **InnoDB:** With the innodb_force_recovery configuration option set to 2 or greater, a shutdown could hang after the message:

  ```
  InnoDB: Waiting for purge thread to be suspended
  ```

  This issue was introduced during recent changes within the MySQL 5.6 development cycle. (Bug #13830371)

- **InnoDB:** Deleting a huge amount of data from InnoDB tables within a short time could cause the purge operation that removes delete-marked records to stall. This issue could result in unnecessary disk space use, but does not cause any problems with data integrity. If this issue causes a disk space shortage, restart the server to work around it. This issue is only likely to occur on 32-bit platforms. (Bug #13847885)

- **InnoDB:** The server could crash when using the SAVEPOINT statement in conjunction with InnoDB tables containing FULLTEXT indexes. (Bug #13831840)

- **InnoDB:** Running concurrent bulk inserts on a server with auto_increment_offset=1, auto_increment_increment greater than 1, and innodb_autoinc_lock_mode=1 could result in intermittent errors like the following, even with the primary key set to auto_increment and omitted from the INSERT statement:

  ```
  Duplicate entry 'value' for key 'PRIMARY'
  ```

  The workaround was to set auto_increment_offset=1 or innodb_autoinc_lock_mode=0 (“traditional”). (Bug #13817703, Bug #61209)

- **InnoDB:** A slave server in a replication configuration could exit while creating an InnoDB temporary table. (Bug #13838761)

- **InnoDB:** The performance_schema counters for InnoDB RW-locks did not record some cases where mini-transactions acquired locks. (Bug #13860722)

- **InnoDB:** During an ALTER TABLE statement to create a primary key for an InnoDB table, some column characteristics could be set incorrectly, leading to errors during subsequent queries. The incorrect data could be the maximum length for a column prefix, or the state of the NOT NULL flag.

  In MySQL 5.1, this fix applies to the InnoDB Plugin, but not the built-in InnoDB storage engine. (Bug #13641275)

- **InnoDB:** The server could halt with an assertion error when DDL and DML operations were run on the same InnoDB table simultaneously:

```
InnoDB: Error: a record lock wait happens in a dictionary operation!
```

This fix stems from the online DDL feature in MySQL 5.6. (Bug #13641926)

- **InnoDB:** If the innodb_undo_tablespaces and innodb_undo_logs configuration options were specified to refer to separate undo tablespaces, and the associated tablespaces did not exist, that error condition was not being correctly detected during startup. (Bug #13016100)

- **InnoDB:** An ALTER TABLE statement for an InnoDB table that dropped one index and create another could fail with an error code 1280, and displaying the wrong index name in the message. (Bug #13029445, Bug #62544)

- **InnoDB:** The error handling and message was improved for attempting to create a foreign key with a column referencing itself. The message suggested a potential problem with the data dictionary, when no such problem existed. (Bug #12902967)

- **InnoDB:** For an InnoDB table with a trigger, under the setting innodb_autoinc_lock_mode=1, sometimes auto-increment values could be interleaved when inserting into the table from two sessions concurrently. The sequence of auto-increment values could vary depending on timing, leading to data inconsistency in systems using replication. (Bug #12752572, Bug #61579)

- **InnoDB:** An ALTER TABLE with both IGNORE and ADD UNIQUE KEY clauses produced an error if duplicates were found, rather than removing all duplicate rows after the first one. With this fix, the ALTER TABLE IGNORE syntax automatically enables the ALGORITHM=COPY clause if the ALTER TABLE statement creates an index. (Bug #12622150)

- **InnoDB:** When data was removed from an InnoDB table, newly inserted data might not reuse the freed disk blocks, leading to an unexpected size increase for the system tablespace or .ibd file (depending on the setting of innodb_file_per_table. The OPTIMIZE TABLE could compact a .ibd file in some cases but not others. The freed disk blocks would eventually be reused as additional data was inserted. (Bug #11766634, Bug #59783)

- **InnoDB:** The CHECK TABLE statement could fail for a large InnoDB table due to a timeout value of 2 hours. For typical storage devices, the issue could occur for tables that exceeded approximately 200 or 350 GB, depending on I/O speed. The fix relaxes the locking performed on the table being checked, which makes the timeout less likely. It also makes InnoDB recognize the syntax CHECK TABLE QUICK, which avoids the possibility of the timeout entirely. (Bug #11758510, Bug #50723)

- **InnoDB:** Full-text search in InnoDB tried to follow foreign key references without keeping track of which ones it had already seen. With circular and other complex setups, this could loop forever or a very long time, leading to the appearance of the query thread hanging. (Bug #64274, Bug #13701973)

- **Partitioning:** If a partitioned table t1 was created using the ROW_FORMAT option, attempting to perform ALTER TABLE t1 EXCHANGE PARTITION ... WITH TABLE t2 failed with the error Tables have different definitions even if the definition for table t2 was identical to that for t1. This occurred because a check was made for an explicit ROW_FORMAT setting in the table definition, and if this was set, the operation was rejected.

  Now in such cases the row format actually used for each table is checked explicitly and the EXCHANGE PARTITION operation is permitted to execute if both row formats are the same. (Bug #11894100)

- **Partitioning:** The PARTITION_COMMENT column of the INFORMATION_SCHEMA.PARTITIONS table truncated partition comments, displaying only the first 80 characters.

  As part of the fix for this issue, the maximum length for a partition comment is now set at 1024 characters, and this width is honored by INFORMATION_SCHEMA.PARTITIONS.PARTITION_COMMENT. (Bug #11748924, Bug #37728)

- **Replication:** It was possible in some cases when using semisynchronous replication for log rotation to take place before an ongoing transaction was committed or rolled back. (Bug #14123372)

- **Replication:** When a complete global transaction spanned relay logs such that only its GTID appeared in a given relay log while the body of the transaction (including `BEGIN` and `COMMIT` statements) appeared in the next relay log, the GTID was interpreted incorrectly as belonging to an empty group. (Bug #14136654)
- **Replication:** If the relay logs were removed after the server was stopped, without stopping replication first, the server could not be started correctly. (Bug #14029212, Bug #65152)

  References: See also Bug #13971348.
- **Replication:** The `--bootstrap` option for `mysqld` is used by `mysql_install_db` when it initializes the system tables. Now, whenever this option is used, GTIDs (see Replication with Global Transaction Identifiers) and replication are automatically disabled. (Bug #13992602)
- **Replication:** It was theoretically possible for concurrent execution of more than one instance of `SHOW BINLOG EVENTS` to crash the MySQL Server. (Bug #13979418)
- **Replication:** The text for the error **`ER_AUTO_POSITION_REQUIRES_GTID_MODE_ON`** referred to `AUTO_POSITION = 1` although this should be `MASTER_AUTO_POSITION = 1`. The text has been corrected. (Bug #13868465)
- **Replication:** If errors were encountered while trying to initialize the `mysql.slave_master_info` or `mysql.slave_relay_log_info` tables, the server refused to start. Now in such cases, the warning message `Error while checking replication metadata. This might also happen when doing a live upgrade from a version that did not make use of the replication metadata tables` is issued to advise the user that this has happened, but the server is permitted to continue starting. (Bug #13893363)
- **Replication:** Queries that were more than 255 characters in length were truncated when viewed in the output of `SHOW BINLOG EVENTS` or `mysqlbinlog`. This was due to the length of the query being stored in `Rows_query_log_events` using a single byte. (Bug #13799489)
- **Replication:** A `CHANGE MASTER TO` statement could alter the effective value of `relay_log_purge`. In addition, the `relay_log_recovery` system variable is now read-only, and can be changed only by starting the server with `--relay-log-recovery`. (Bug #13840948)
- **Replication:** When `binlog_rows_query_log_events` = 1 and a statement is written to the binary log using the row-based logging format, the server generates a an additional log event containing the text of the original statement. If `mysqlbinlog` is executed on this log using the `--verbose --verbose`, the original statement is printed. To prevent the statement from being executed in addition to the row event (which would in effect cause the statement to be excuted twice), it is commented out with a leading `#` character.

  This was implemented with the assumption that such a statement would consist of a single line, which meant that a statement covering multiple lines was handled incorrectly, in that only the first line of the statement actually commented out. Now in such cases, every line of the statement is commented out with a leading `#`. (Bug #13799555)
- **Replication:** Replication locks and some of the protocols controlling the use of these locks were not well implemented or enforced. In particular, this fix improves lock handling for statements such as `CHANGE MASTER TO`, `SHOW SLAVE STATUS`, and `FLUSH LOGS`. (Bug #13779291)
- **Replication:** When logging transactions that affected both transactional and nontransactional tables, the following statements could sometimes be written into the binary log in the wrong order or on the wrong side of a transaction boundary:
  - `SET`,
  - `CREATE FUNCTION`,
  - `SHOW BINLOG EVENTS`,

  - REVOKE ALL PRIVILEGES.

  (Bug #13627921)

- **Replication:** To provide a crash-safe slave, it was previously necessary to change the storage engine for the slave_master_info, slave_relay_log_info, and slave_worker_info tables from MyISAM to InnoDB manually, by issuing ALTER TABLE. To simplify the setup of replication using these slave log tables, they are now created using the InnoDB storage engine. (Bug #13538891)

- **Replication:** Setting binlog_checksum on the master to a value that was unknown on the slave caused replication to fail. Now in such cases, replication checksums are disabled on the slave and replication stops with an appropriate error message. (Bug #13553750, Bug #61096)

- **Replication:** When the slave had been set using CHANGE MASTER TO with the MASTER_DELAY option equal to any permitted value greater than zero, then stopped using STOP SLAVE, pointed at the current relay log position (as shown by SHOW SLAVE STATUS), and started again, START SLAVE failed with the error Could not initialize master info structure. (Bug #12995174)

- **Replication:** The --relay-log-space-limit option was sometimes ignored.

  More specifically, when the SQL thread went to sleep, it allowed the I/O thread to queue additional events in such a way that the relay log space limit was bypassed, and the number of events in the queue could grow well past the point where the relay logs needed to be rotated. Now in such cases, the SQL thread checks to see whether the I/O thread should rotate and provide the SQL thread a chance to purge the logs (thus freeing space).

  Note that, when the SQL thread is in the middle of a transaction, it cannot purge the logs; it can only ask for more events until the transaction is complete. Once the transaction is finished, the SQL thread can immediately instruct the I/O thread to rotate. (Bug #12400313, Bug #64503)

  References: See also Bug #13806492.

- **Replication:** An event whose length exceeded the size of the master dump thread's max_allowed_packet caused replication to fail. This could occur when updating many large rows and using row-based replication.

  As part of this fix, a new server option --slave-max-allowed-packet is added, which permits max_allowed_packet to be exceeded by the slave SQL and I/O threads. Now the size of a packet transmitted from the master to the slave is checked only against this value (available as the value of the slave_max_allowed_packet server system variable), and not against the value of max_allowed_packet. (Bug #12400221, Bug #60926)

- **Replication:** Statements using AUTO_INCREMENT, LAST_INSERT_ID(), RAND(), or user variables could be applied in the wrong context on the slave when using statement-based replication and replication filtering server options (see How Servers Evaluate Replication Filtering Rules). (Bug #11761686, Bug #54201)

  References: See also Bug #11754117, Bug #45670, Bug #11746146, Bug #23894.

- **Replication:** An INSERT into a table that has a composite primary key that includes an AUTO_INCREMENT column that is not the first column of this composite key is not safe for statement-based binary logging or replication. Such statements are now marked as unsafe and fail with an error when using the STATEMENT binary logging format. For more information, see Determination of Safe and Unsafe Statements in Binary Logging, as well as Replication and AUTO_INCREMENT.

> **Note**
>
> This issue does not affect tables using the InnoDB storage engine, since an InnoDB table with an AUTO_INCREMENT column requires at least one key where the auto-increment column is the only or leftmost column.

(Bug #11754117, Bug #45670)

References: See also Bug #11761686, Bug #54201, Bug #11746146, Bug #23894.

- **Replication:** After upgrading a replication slave to MySQL 5.6.2 or later, enabling the query cache eventually caused the slave to fail. (Bug #64624, Bug #14005409)

- **Microsoft Windows:** For Microsoft Windows, the deprecated MySQL Configuration Wizard is no longer distributed, and instead the newer MySQL Installer is available and preferred.

- After running `ALTER TABLE *tbl* DISCARD TABLESPACE` for an `InnoDB` table, certain other `ALTER TABLE` operations such as renaming the table or rebuilding the primary key could cause a crash. (Bug #14213568)

- For conditions of the form `WHERE p1 AND (p2 OR p3)`, the optimizer now uses the index merge access method on `(p2,p3)` if it is more efficient than a range scan on `p1`. Previously, index merge was not considered when a range scan was possible. (Bug #14208922)

- With logging of the general query log to a table, logging was disabled within a read-only transaction because write lock acquisition on the log table was blocked. (Bug #14136866)

- Error messages that should have said "YEAR(2)" said "YEAR(0)" instead. (Bug #14167585)

- For debug builds, `INSERT IGNORE INTO ... SELECT` that selected more than `max_join_size` rows could raise an assertion. (Bug #14145442)

- If a nonexistent page was requested to be loaded into the `InnoDB` buffer pool by the `innodb_buffer_pool_load_at_startup` configuration option, a subsequent shutdown operation could hang. (Bug #14106082)

- The Performance Schema did not generate consistent digest values for `CALL` statements. (Bug #14069132)

- The `ARCHIVE` storage engine could not be built unless the Performance Schema was also built. (Bug #14116252)

- In debug builds, warnings occurring during creation of an `InnoDB` table with `ROW_FORMAT=DYNAMIC` and `innodb_file_per_table` disabled could raise an assertion. (Bug #14101563)

- Derived tables and tables created with `CREATE TABLE ... SELECT` using the output from single-row queries with `NULL` in the first column could change the value to 0. (Bug #14069831)

- Incorrect assessment of column nullability for a subquery result within a trigger could cause “column cannot be null” errors. (Bug #14069810, Bug #14005353)

- In debug builds, the server failed to check for error status from the storage engine and raised an assertion. (Bug #14101852)

- When the index enforcing a foreign key constraint was dropped while `foreign_key_checks=0`, further operations involving the foreign key column could cause a serious error after the `foreign_key_checks` option was re-enabled. (Bug #14025221)

- For debug builds compiled with `gcov`, tests that used `DBUG_SUICIDE` lost `gcov` data. (Bug #14028421)

- The LooseScan semi-join strategy could fail to remove duplicates from the result set. (Bug #14053325)

- Certain arguments to RPAD() could lead to "uninitialized variable" warnings. (Bug #14039955)
- Mishandling of failed internal commits in administrative statements such as ANALYZE TABLE could cause an assertion to be raised. (Bug #14001091)
- Queries containing references to user variables were not written to the general query log with some rewriting, not as received. (Bug #13958454)
- For debug builds, conversion of a double-precision value to the lldiv_t type could raise an assertion. (Bug #13976233)
- Some arguments to MAKETIME() could cause a buffer overflow. (Bug #13982125)
- Selecting MIN() or MAX() from a left or right join involving an INFORMATION_SCHEMA table could cause a server crash. (Bug #13966514)
- For debug builds, the optimizer could change the query plan when checking sort order and return incorrect results. (Bug #13949068)
- Improper calculation of decimals for TIME values given as arguments to IF() or IFNULL() could cause a server crash. (Bug #13988413, Bug #14042545)
- Mishandling of failure during multiple-table UPDATE IGNORE statements could cause an assertion to be raised. (Bug #13974815)
- Queries that grouped by an outer BLOB column in a subquery caused a server crash. (Bug #13966809)
- Overhead for Performance Schema table aggregation operations was excessive. (Bug #13862186)
- An infinite thread loop could develop within Performance Schema, causing the server to become unresponsive. (Bug #13898343)
- When the InnoDB persistent statistics feature was turned on, an ALTER TABLE statement on an InnoDB table with delete-marked records could cause a crash (non-debug builds) or assertion error (debug builds). (Bug #13838962, Bug #13867915)
- viosslfactories did not compile on Oracle Linux 6.0 with CMake options -DWITH_SSL=system and -DWITH_DEBUG=1. (Bug #13799126)
- If KILL QUERY interrupted an INSERT or UPDATE that had the IGNORE modifier, OK was incorrectly returned to the client rather than an error code. Now an error ("Query execution was interrupted") is returned instead. (Bug #13822652)
- If KILL QUERY interrupted a statement during derived table materialization, the server crashed later trying to read the nonexistent materialized table. (Bug #13820776)
- For comparison of a temporal value to and indexed character column, the optimizer could apply the range access method and thus perform an indexed search that found only literal matches. This is incorrect because MySQL permits a variety of delimiters in temporal values represented as strings. (Bug #13803810)
- Some errors in MySQL 5.6 had different numbers than in MySQL 5.5. (Bug #13833438)
- Incorrect stored program caching could cause statements within a stored program that included a GROUP BY clause to return different results across multiple program invocations. (Bug #13805127)
- With subquery materialization enabled, some queries with a subquery in the HAVING clause caused a server crash. (Bug #13848789)
- Several clarifications were made to optimizer trace output. (Bug #13799348)
- The version_compile_machine system variable sometimes did not include the value 64 for server binaries compiled on a 64-bit system. (Bug #13859866)

- The Performance Schema stored identifiers in digest tables as `utf8` without converting them from the original character set first. (Bug #13809293)
- In bootstrap mode, the server signal handler thread did not shut down if the server aborted early. (Bug #13837221)
- Incorrect cost calculations for two-table joins could lead to incorrect join order. (Bug #13810048)

  References: This bug is a regression of Bug #26106.
- In debug builds, a race condition in a signal handler during shutdown caused a server crash. (Bug #13793813)
- For open ranges that effectively resulted in a full index scan, the optimizer did not discard the range predicate as unneeded. (Bug #13731380)
- A prepared statement that referenced views and were executed using semi-join transformation could return different results for different executions. (Bug #13773979)

  References: See also Bug #14641759.
- `(a,b) IN (SELECT c,d FROM t1 WHERE ...)` could produce incorrect results if `t1` had an index on `(c, d)` and `c` or `d` contained `NULL` values. (Bug #13731417)
- Outer join queries with `ALL` could return incorrect results because the optimizer incorrectly rewrote them to use inner join. (Bug #13735712)
- The range optimizer sometimes did not treat equivalent expressions the same, depending on the order of the operands. For example, it could treat `a <= b` and `b >= a` differently. (Bug #13701206)
- With semi-join optimization enabled, an assertion was raised for queries for which the number of tables was greater than the search depth. (Bug #13685026)
- A query executed with literal values in the `WHERE` clause could return results different from the same query written to select the same literal values from a separate table using a `SELECT` statement in the `WHERE` clause. (Bug #13468414)
- Truncating a table partition did not invalidate queries in the query cache that used the table. (Bug #13485448)
- Setting `max_sort_length` to small values could cause a server crash. (Bug #13485416)
- Condition handler code could assume that after handler execution, control would pass up a single level to the parent, sometimes leading to a server crash. (Bug #13431226)
- If a `GROUP_CONCAT()` result was calculated using intermediate results (for example, if `ORDER BY` or `DISTINCT` was present), individual intermediate results were each truncated to a maximum of 64K, even if the `group_concat_max_len` system variable was set to a larger value. Now the length of any intermediate result and the final result are controlled by the `group_concat_max_len` value. (Bug #13387020)
- Queries with `ALL` subquery predicates could return incorrect results due to a faulty query transformation. (Bug #13330886)
- Switching between index scans and random scans using the `HANDLER` interface could result in failure of the interface to properly reinitialize scans. (Bug #13008220)
- The presence of a file named `.empty` in the `test` database prevented that database from being dropped. (Bug #12845091)
- For queries with `ORDER BY COUNT(*)` and `LIMIT`, the optimizer could choose an execution plan that produced incorrect results. (Bug #12713907)

- IPv6 functions such as `IS_IPV6()` produced Valgrind warnings with arguments that used a multibyte character set. (Bug #12635232, Bug #14040277)

- For some subqueries that should be executed using a range scan on a nonprimary index and required use of filesort, only the first execution of the subquery was done as a range scan. All following executions were done as full table scans, resulting in poor performance. In addition, if index condition pushdown was used, incorrect results could be returned. (Bug #12667154)

- Queries that used `STRAIGHT_JOIN` and were executed using Multi-Range Read optimization could result in a memory leak. (Bug #12365385)

- Overhead for the Performance Schema was reduced. (Bug #12346211)

- `IN` subqueries that used a variance or standard deviation aggregate function could return a different result depending on whether the `optimizer_switch` `materialization` flag was enabled.

  > **Note**
  >
  > Those aggregate functions may now return a result with a different number of decimals from previously.

  (Bug #11766758)

- On Windows, initial database creation failed during bootstrapping. (Bug #11766342)

- `SAVEPOINT` statements were incorrectly disallowed within `XA` transactions. (Bug #64374, Bug #13737343)

  References: See also Bug #11766752.

- If the `--bind-address` option was given a host name value and the host name resolved to more than one IP address, the server failed to start. For example, with `--bind-address=localhost`, if `localhost` resolved to both `127.0.0.1` and `::1`, startup failed. Now the server prefers the IPv4 address in such cases. (Bug #61713, Bug #12762885)

- Under some conditions, the effect of `RENAME USER` was not recognized until `FLUSH PRIVILEGES` was used (which should not be necessary). (Bug #61865, Bug #12766319)

- The Performance Schema incorrectly displayed some backslashes in Windows file names (by doubling them). (Bug #63339, Bug #13417446)

- `SHOW` statements treated stored procedure, stored function, and event names as case sensitive. (Bug #56224, Bug #11763507)

- With `lower_case_table_names=2` on systems with case-insensitive file systems such as Windows or Mac OS X, `CREATE TABLE ... LIKE` did not preserve lettercase of the destination table name as given in the statement. (Bug #64211, Bug #13702397)

- MySQL was overly agressive in enforcing the `NO_ZERO_DATE` and `NO_ZERO_IN_DATE` SQL modes for default values in column definitions for `CREATE TABLE` and `ALTER TABLE` statements. Previously, default dates that were invalid with those SQL modes enabled produced an error, even when strict mode was not enabled. Now with `NO_ZERO_DATE` or `NO_ZERO_IN_DATE` enabled, invalid default dates produce a warning if strict SQL mode is not enabled, and an error if strict mode is enabled. (Bug #34280, Bug #11747847)

- `mysql_store_result()` and `mysql_use_result()` are not for use with prepared statements and are not intended to be called following `mysql_stmt_execute()`, but failed to return an error when invoked that way in `libmysqld`. (Bug #62136, Bug #13738989)

  References: See also Bug #47485.

- For `ALTER TABLE` with the `IGNORE` keyword, `IGNORE` is now part of the information provided to the storage engine. It is up to the storage engine whether to use this when choosing between the inplace or copy algorithm for altering the table. For InnoDB index operations, IGNORE is not used if the index is unique, so the copy algorithm is used. (Bug #40344, Bug #11750045)

- Some Czech error messages contained invalid characters. (Bug #64310, Bug #13726075)
- A multiple-table UPDATE with the IGNORE keyword resulted in an inappropriate and not meaningful Got error 0 from storage engine message. (Bug #49539, Bug #11757486)
- SHOW TABLES was very slow unless the required information was already in the disk cache. (Bug #60961, Bug #12427262)
- File access by the ARCHIVE storage engine was not instrumented and thus not shown in Performance Schema tables. (Bug #63340, Bug #13417440)
- An inappropriate mutex was used to protect random number generation, causing contention during connect operations. (Bug #62282, Bug #12951609)
- mysqlbinlog exited with no error code if file write errors occurred. (Bug #55289, Bug #11762667)
- On Windows, the mysql client crashed when invoked using its full path name. (Bug #60858, Bug #12402882)
- Due to a race condition, it was possible for two threads to end up with the same query ID for different queries. (Bug #58785, Bug #11765785)
- The server crashed at shutdown if the slow query log file was a named pipe. (Bug #64345, Bug #13733221)
- Using CONCAT() to construct a pattern for a LIKE pattern match could result in memory corrupting and match failure. (Bug #59140, Bug #11766101)
- For debug builds, errors occurring during processing of INSERT DELAYED statements could crash the server. (Bug #60114, Bug #11827404)
- 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)
- yaSSL rejected valid SSL certificates that OpenSSL accepts. (Bug #54348, Bug #11761822)
- A regression bug in the optimizer could cause excessive disk usage for UPDATE statements on InnoDB tables. For tables created with innodb_file_per_table enabled, OPTIMIZE TABLE can be used to recover excessive space used. For tables created in the InnoDB system tablespace,it is necessary to perform a dump and restore into a new instance of the system tablespace. (Bug #65745, Bug #14248833)
- Sessions could end up deadlocked when executing a combination of SELECT, DROP TABLE, KILL, and SHOW ENGINE INNODB STATUS. (Bug #60682, Bug #12636001)
- For queries with range predicates, the optimizer could miscalculate the number of key parts used, possibly leading to a server crash. (Bug #58731, Bug #11765737)
- Parse errors that occurred while loading UCA or LDML collation descriptions were not written to the error log. (Bug #65593, Bug #14197426)
- The optimizer could chose a worse execution plan for a condition that used a quoted number compared to the unquoted number. (Bug #43319, Bug #11752201)
- When a query was killed, the error code was not always properly propagated up through the server code. (Bug #43353, Bug #11752226)
- LEFT JOIN on derived tables was very slow. This is now addressed through the use of subquery materialization. (Bug #34364, Bug #11747876)

- Queries that used `WHERE (col1, col2) IN ((const, const))` were optimized for `SELECT`, but not for `DELETE` or `UPDATE`. (Bug #43187, Bug #11752097)

- If an account had a nonzero `MAX_USER_CONNECTIONS` value, that value was not always respected. (Bug #65104, Bug #14003080)

- If the server held a global mutex while doing network I/O, client disconnections could be slow. (Bug #53096, Bug #11760669)

- Incorrect metadata could be produced for columns returned from some views. (Bug #65379, Bug #14096619)

- When dumping the `mysql` database, `mysqldump` did not include the `general_log` and `slow_query_log` tables because they cannot be locked. This caused a problem after reloading the dump file if that file contained a `DROP DATABASE` statement for the `mysql` database: The database no longer contained the log tables and attempts to log to them failed. Now `mysqldump` includes statements to re-create the `general_log` and `slow_query_log` tables so that they exist after loading the dump file. Log table contents still are not dumped. (Bug #45740, Bug #11754178)

- On Windows, `mysqlslap` crashed for attempts to connect using shared memory. (Bug #31173, Bug #11747181, Bug #59107, Bug #11766072)

- During the startup process, `mysqld` could incorrectly remove the PID file of an already running `mysqld`. (Bug #23790, Bug #11746142)

  References: See also Bug #14726272.

- Using `ALTER TABLE` to add a `TIMESTAMP` column containing `DEFAULT CURRENT_TIMESTAMP` in the definition resulted in a column containing `'0000-00-00 00:00:00'`, not the current timestamp. (Bug #17392, Bug #11745578)

- For table or database names that are longer than 64 characters, the error “Incorrect table name” was returned rather than “Identifier too long”. (Bug #25168, Bug #11746295)

- Redundant “Specified key was too long” messages could be produced by index-creation operations. (Bug #31149, Bug #11747177)

- Code for the storage engine API did not check the return value from the `ha_rnd_init()`, `ha_index_init()`, and `index_init()` functions. (Bug #26040, Bug #11746399, Bug #54166, Bug #11761652)

Changes in MySQL 5.6.5 (2012-04-10, Milestone 8)

> **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.

Beginning with MySQL 5.6.5, Oracle no longer provides binaries for Mac OS X 10.5. This aligns with Apple no longer providing updates or support for this platform.

Data Type Notes

- Previously, at most one `TIMESTAMP` column per table could be automatically initialized or updated to the current date and time. This restriction has been lifted. Any `TIMESTAMP` column definition can have any combination of `DEFAULT CURRENT_TIMESTAMP` and `ON UPDATE CURRENT_TIMESTAMP` clauses. In addition, these clauses now can be used with `DATETIME` column definitions. For more information, see Automatic Initialization and Updating for `TIMESTAMP` and `DATETIME`.

Replication with GTIDs

- **Important Change; Replication:** This release introduces *global transaction identifiers* (GTIDs) for MySQL Replication. A GTID is a unique identifier that is assigned to each transaction as it is committed; this identifier is unique on the MySQL Server where the transaction originated, as well as across all MySQL Servers in a given replication setup. Because GTID-based replication depends on tracking transactions, it cannot be employed with tables that employ a nontransactional storage engine such as `MyISAM`; thus, it is currently supported only with `InnoDB` tables.

  Because each transaction is uniquely identified, it is not necessary when using GTIDs to specify positions in the master's binary log when starting a new slave or failing over to a new master. This is reflected in the addition of a new `MASTER_AUTO_POSITION` option for the `CHANGE MASTER TO` statement which takes the place of the `MASTER_LOG_FILE` and `MASTER_LOG_POS` options when executing this statement to prepare a MySQL Server to act as a replication slave.

  To enable GTIDs on a MySQL Server, the server must be started with the options `--gtid-mode=ON` `--disable-gtid-unsafe-statements` `--log-bin` `--log-slave-updates`. These options are needed whether the server acts as a replication master or as a replication slave; the `--gtid-mode` and `--disable-gtid-unsafe-statements` options are new in this release. Once the master and slave have each been started with these options, it is necessary only to issue a `CHANGE MASTER TO ... MASTER_AUTO_POSITION=1` followed by `START SLAVE` on the slave to start replication.

  A number of new server system variables have also been added for monitoring GTID usage. For more information about these options and variables, see Global Transaction ID Options and Variables.

  As part of these changes, three new `mysqlbinlog` options—`--include-gtids`, `--exclude-gtids`, and `--skip-gtids`—have been added for reading binary logs produced when the server participates in replication with GTIDs.

  > **Important**
  >
  > Due to an issue discovered just prior to release, you cannot import a dump made using `mysqldump` from a MySQL 5.5 server to a MySQL 5.6.5 server and then use `mysqlupgrade` on the MySQL 5.6.5 server while GTIDs are enabled; doing so makes it impossible to connect to the server normally following the upgrade. Instead, you should import the dump and run `mysqlupgrade` while the MySQL 5.6.5 server is running with `--gtid-mode=OFF`, then restart it with `--gtid-mode=ON`. (Bug #13833710) (`mysqlupgrade` can be executed when the server is running with `--gtid-mode` set either to `OFF`, or to `ON`.)

  For additional information about GTIDs and setting up GTID-based replication, see Replication with Global Transaction Identifiers.

Host Cache Notes

- MySQL now provides more information about the causes of errors that occur when clients connect to the server, as well as improved access to the host cache, which contains client IP address and host name information and is used to avoid DNS lookups. These changes have been implemented:

  - New `Connection_errors_`*`xxx`* status variables provide information about connection errors that do not apply to specific client IP addresses.

  - The host cache has additional counters to track errors that do apply to specific IP addresses.

  - A new `host_cache` Performance Schema table exposes the contents of the host cache so that it can be examined using `SELECT` statements. Access to host cache contents makes it possible to answer questions such as how many hosts are cached, what kinds of connection errors are occurring for which hosts, or how close host error counts are to reaching the max_connect_errors system variable limit. The Performance Schema must be enabled or this table is empty.

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

- The host cache size now is configurable using the host_cache_size system variable. Setting the size to 0 disables the host cache.This is similar to disabling the cache by starting the server with --skip-host-cache, but using host_cache_size is more flexible because it can also be used to resize, enable, or disable the host cache at runtime, not just at server startup. If you start the server with --skip-host-cache, the host cache cannot be re-enabled at runtime.

For more information, see DNS Lookup Optimization and the Host Cache, and The host_cache Table. (Bug #22821, Bug #24906, Bug #45817, Bug #59404, Bug #11746048, Bug #11746269, Bug #11754244, Bug #11766316)

Optimizer Notes

- These query optimizer improvements were implemented:
  - The EXPLAIN statement now can produce output in JSON format. To select this, use EXPLAIN FORMAT = JSON *explainable_stmt* syntax. With FORMAT = JSON, the output includes regular EXPLAIN information, as well as extended and partition information.

    Traditional EXPLAIN output has also changed so that empty columns contain NULL rather the empty string. In addition, UNION RESULT rows have Using filesort in the Extra column because a temporary table is used to buffer UNION results.

    To work for both Optimizer Trace and JSON-format EXPLAIN output, the end_marker parameter for the optimizer_trace system variable has been moved to a separate end_markers_in_json system variable. This is an incompatible change to the optimizer_trace variable. For more information, see MySQL Internals: Tracing the Optimizer.
  - The optimizer tries to find the best query execution plan by beginning with the most promising table and recursively adding to the plan the most promising of the remaining tables. Partial execution plans with a higher cost than an already found plan are pruned. The optimizer now attempts to improve the order in which it adds tables to the plan, resulting in a reduction of the number of partial plans considered.

    Queries that are likely to have improved performance are joins of many tables, where most tables use eq_ref or ref join types (as indicated by EXPLAIN output).

    A new status variable, Last_query_partial_plans, counts the number of iterations the optimizer makes in execution plan construction for the previous query.
  - The optimizer uses semi-join and materialization strategies to optimize subquery execution. See Optimizing Subqueries with Semi-Join Transformations, and Optimizing Subqueries with Subquery Materialization. In addition, the Batched Key Access (BKA) Join and Block Nested-Loop (BNL) Join algorithms used for inner join and outer join operations have been extended to support semi-join operations. For more information, see Block Nested-Loop and Batched Key Access Joins.

    Several flags have been added to the optimizer_switch system variable to enable control over semi-join and subquery materialization strategies. The semijoin flag controls whether semi-joins are used. If it is set to on, the firstmatch and loosescan flags enable finer control over the permitted semi-join strategies. The materialization flag controls whether subquery materialization is used. If semijoin and materialization are both on, semi-joins also use materialization where applicable. These flags are on by default. See Controlling Switchable Optimizations.

  - For expressions such as *col_name* IN(*values*) that compare a column to a list of values, the optimizer previously made row estimates using index dives for each value in the list. This becomes inefficient as the number of values becomes large. The optimizer now can make row estimates for such expressions using index statistics instead, which is less accurate but quicker for a large number of values. The point at which the optimizer switches from index dives to index statistics is configurable using the new eq_range_index_dive_limit system variable. For more information, see Equality Range Optimization of Many-Valued Comparisons.

Performance Schema Notes

- The Performance Schema has these additions:

  - The Performance Schema now has a host_cache table that exposes the contents of the host cache so that it can be examined using SELECT statements. See Host Cache Notes elsewhere in this changelog.

  - The Performance Schema now maintains statement digest information. This normalizes and groups statements with the same “signature” and permits questions to be answered about the types of statements the server is executing and how often they occur.

    - A statement_digest consumer in the setup_consumers table controls whether the Performance Schema maintains digest information.

    - The statement event tables (events_statements_current, events_statements_history, and events_statements_history_long) have DIGEST and DIGEST_TEXT columns that contain digest MD5 values and the corresponding normalized statement text strings.

    - A events_statements_summary_by_digest table provides aggregated statement digest information.

  For more information, see The host_cache Table, Performance Schema Statement Event Tables, and Statement Summary Tables.

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

Security Notes

- Passwords stored in the older hash format used before MySQL 4.1 are less secure than passwords that use the native password hashing method and should be avoided. Pre-4.1 passwords and the mysql_old_password authentication plugin are now deprecated. To prevent connections using accounts that have pre-4.1 password hashes, the secure_auth system variable is now enabled by default. (To permit connections for accounts that have such password hashes, start the server with --secure_auth=0.) (Bug #13586336)

- MySQL client programs now issue a warning if a password is given on the command line that this can be insecure.

Functionality Added or Changed

- **Incompatible Change:** The obsolete OPTION modifier for the SET statement has been removed.

- **InnoDB:** --ignore-builtin-innodb is now ignored if used. (Bug #13586262)

- The MySQL-shared-compat RPM package enables users of Red Hat-provided mysql-*-5.1 RPM packages to migrate to Oracle-provided MySQL-*-5.5 packages. MySQL-shared-compat now replaces the Red Hat mysql-libs package by replacing libmysqlclient.so files of the latter package, thus satisfying dependencies of other packages on mysql-libs. This change affects only users of Red Hat (or Red Hat-compatible) RPM packages. Nothing is different for users of Oracle RPM packages. (Bug #13867506)

- Temporary tables for INFORMATION_SCHEMA queries now use dynamic MyISAM row format if they contain sufficiently large VARCHAR columns, resulting in space savings. (Bug #13627632)
- A new CMake option, MYSQL_PROJECT_NAME, can be set on Windows or Mac OS X to be used in the project name. (Bug #13551687)
- As of MySQL 5.5.3, the LOW_PRIORITY modifier for LOCK TABLES ... LOW_PRIORITY WRITE has no effect. This modifier is now deprecated. Its use should be avoided and now produces a warning. Use LOCK TABLES ... WRITE instead. (Bug #13586314)
- If the log_queries_not_using_indexes system variable is enabled, slow queries that do not use indexes are written to the slow query log. In this case, it is now possible to put a logging rate limit on these queries by setting the new log_throttle_queries_not_using_indexes system variable, so that the slow query log does not grow too quickly. By default, this variable is 0, which means there is no limit. Positive values impose a per-minute limit on logging of queries that do not use indexes. The first such query opens a 60-second window within which the server logs queries up to the given limit, then suppresses additional queries. If there are suppressed queries when the window ends, the server logs a summary that indicates how many there were and the aggregate time spent in them. The next 60-second window begins when the server logs the next query that does not use indexes. (Bug #55323, Bug #11762697)
- A new server option, --slow-start-timeout, controls the Windows service control manager's service start timeout. The value is the maximum number of milliseconds that the service control manager waits before trying to kill the MySQL service during startup. The default value is 15000 (15 seconds). If the MySQL service takes too long to start, you may need to increase this value. A value of 0 means there is no timeout. (Bug #45546, Bug #11754011)
- The mysql client now supports an --init-command=*str* option. The option value is an SQL statement to execute after connecting to the server. If auto-reconnect is enabled, the statement is executed again after reconnection occurs. (Bug #45634, Bug #11754087)
- Several subquery performance issues were resolved through the implementation of semi-join subquery optimization strategies. See Optimizing Subqueries with Semi-Join Transformations. (Bug #47914, Bug #11756048, Bug #58660, Bug #11765671, Bug #10815, Bug #11745162, Bug #9021, Bug #13519134, Bug #48763, Bug #11756798, Bug #25130, Bug #11746289)
- New utf8_general_mysql500_ci and ucs2_general_mysql500_ci collations have been added that preserve the behavior of utf8_general_ci and ucs2_general_ci from versions of MySQL previous to 5.1.24. Bug #27877 corrected an error in the original collations but introduced an incompatibility for columns that contain German 'ß' LATIN SMALL LETTER SHARP S. (As a result of the fix, that character compares equal to characters with which it previously compared different.) A symptom of the problem after upgrading to MySQL 5.1.24 or newer from a version older than 5.1.24 is that CHECK TABLE produces this error:

  ```
  Table upgrade required.
  Please do "REPAIR TABLE `t`" or dump/reload to fix it!
  ```

  Unfortunately, REPAIR TABLE could not fix the problem. The new collations permit older tables created before MySQL 5.1.24 to be upgraded to current versions of MySQL.

  To convert an affected table after a binary upgrade that leaves the table files in place, alter the table to use the new collation. Suppose that the table t1 contains one or more problematic utf8 columns. To convert the table at the table level, use a statement like this:

  ```
  ALTER TABLE t1
  CONVERT TO CHARACTER SET utf8 COLLATE utf8_general_mysql500_ci;
  ```

  To apply the change on a column-specific basis, use a statement like this (be sure to repeat the column definition as originally specified except for the COLLATE clause):

```
ALTER TABLE t1
MODIFY c1 CHAR(N) CHARACTER SET utf8 COLLATE utf8_general_mysql500_ci;
```

To upgrade the table using a dump and reload procedure, dump the table using mysqldump, modify the CREATE TABLE statement in the dump file to use the new collation, and reload the table.

After making the appropriate changes, CHECK TABLE should report no error.

For more information, see Checking Whether Tables or Indexes Must Be Rebuilt, and Rebuilding or Repairing Tables or Indexes. (Bug #43593, Bug #11752408)

- MySQL distributions no longer include the GPL readline input-editing library. This results in simpler maintenance and support, and simplifies licensing considerations.

- The SET TRANSACTION and START TRANSACTION statements now support READ WRITE and READ ONLY modifiers to set the transaction access mode for tables used in transactions. The default mode is read/write, which is the same mode as previously. Read/write mode now may be specified explicitly with the READ WRITE modifier. Using READ ONLY prohibits table changes and may enable storage engines to make performance improvements that are possible when changes are not permitted.

  In addition, the new --transaction-read-only option and tx_read_only system variable permit the default transaction access mode to be set at server startup and runtime.

  For more information, see SET TRANSACTION Syntax, and START TRANSACTION, COMMIT, and ROLLBACK Syntax.

Bugs Fixed

- **Performance; InnoDB:** The optimizer now takes into account InnoDB page sizes other than 16KB, which can be configured with the innodb_page_size option when creating a MySQL instance. This change improves the estimates of I/O costs for queries on systems with non-default InnoDB page sizes. (Bug #13623078)

- **Performance; InnoDB:** Memory allocation for InnoDB tables was reorganized to reduce the memory overhead for large numbers of tables or partitions, avoiding situations where the “resident set size” could grow regardless of FLUSH TABLES statements. The problem was most evident for tables with large row size. Some of the memory that was formerly allocated for every open table is now allocated only when the table is modified for the first time. (Bug #11764622, Bug #57480)

- **Incompatible Change; Replication:** CHANGE MASTER TO statements were written into the error log using quoted numeric values, although the syntax for this statement does not allow such option values to be quoted. This meant that such statements could not be copied from the error log and re-run verbatim. Now CHANGE MASTER TO statements are written to the error log without the extraneous quotation marks, and so are syntactically correct as logged.

- **Incompatible Change:** A change in MySQL 5.6.3 caused LAST_DAY() to be more strict and reject incomplete dates with a day part of zero. For this function, a nonzero day part is not necessary, so the change has been reverted. (Bug #13458237)

- **Important Change; InnoDB:** When a row grew in size due to an UPDATE operation, other (non-updated) columns could be moved to off-page storage so that information about the row still fit within the constraints of the InnoDB page size. The pointer to the new allocated off-page data was not set up until the pages were allocated and written, potentially leading to lost data if the system crashed while the column was being moved out of the page. The problem was more common with tables using ROW_FORMAT=DYNAMIC or ROW_FORMAT=COMPRESSED along with the Barracuda file format, particularly with the innodb_file_per_table setting enabled, because page allocation operations are more common as the .ibd tablespace files are extended. Still, the problem could occur with any combination of InnoDB version, file format, and row format.

A related issue was that during such an UPDATE operation, or an INSERT operation that reused a delete-marked record, other transactions could see invalid data for the affected column, regardless of isolation level.

The fix corrects the order of operations for moving the column data off the original page and replacing it with a pointer. Now if a crash occurs at the precise moment when the column data is being transferred, the transfer will not be re-run during crash recovery.

In MySQL 5.1, this fix applies to the InnoDB Plugin, but not the built-in InnoDB storage engine. (Bug #13721257, Bug #12612184, Bug #12704861)

- **Important Change; Partitioning:** The query cache did not always function correctly with partitioned tables in a transactional context. For this reason, the query cache is now disabled for any queries using partitioned tables, and such queries can no longer be cached. For more information, see Restrictions and Limitations on Partitioning. (Bug #11761296, Bug #53775)

- **Important Change; Replication:** The CHANGE MASTER TO statement was not checked for invalid characters in values for options such as MASTER_HOST and MASTER_USER. In addition, when the server was restarted, a value containing certain characters was trimmed, causing the loss of its original value. Now such values are validated, and in cases where the value contains invalid characters, including linefeed (\n or 0x0A) characters, the statement fails with an error (**ER_MASTER_INFO**). (Bug #11758581, Bug #50801)

- **Important Change; Replication:** Moving the binary log file, relay log file, or both files to a new location, then restarting the server with a new value for --log-bin, --relay-log, or both, caused the server to abort on start. This was because the entries in the index file overrode the new location. In addition, paths were calculated relative to datadir (rather than to the --log-bin or --relay-log values).

  The fix for this problem means that, when the server reads an entry from the index file, it now checks whether the entry contains a relative path. If it does, the relative part of the path is replaced with the absolute path set using the --log-bin or --relay-log option. An absolute path remains unchanged; in such a case, the index must be edited manually to enable the new path or paths to be used. (Bug #11745230, Bug #12133)

- **InnoDB:** If InnoDB was started with innodb_force_recovery set to a value of 3 or 4, and there are transactions to roll back, normal shutdown would hang waiting for those transactions to complete. Now the shutdown happens immediately, without rolling back any transactions, because non-zero values for innodb_force_recovery are only appropriate for troubleshooting and diagnostic purposes. (Bug #13628420)

- **InnoDB:** The MySQL server could hang in some cases if the configuration option innodb_use_native_aio was turned off. (Bug #13619598)

- **InnoDB:** An erroneous assertion could occur, in debug builds only, when creating an index on a column containing zero-length values (that is, ''). (Bug #13654923)

- **InnoDB:** A DDL operation such as ALTER TABLE ... ADD COLUMN could stall, eventually timing out with an Error 1005: Can't create table message referring to fil_rename_tablespace. (Bug #13636122, Bug #62100, Bug #63553)

- **InnoDB:** The configuration option innodb_sort_buf_size was renamed to innodb_sort_buffer_size for consistency. This work area is used while creating an InnoDB index. (Bug #13610358)

- **InnoDB:** A Valgrind error was fixed in the function os_aio_init(). (Bug #13612811)

- **InnoDB:** The server could crash when creating an InnoDB temporary table under Linux, if the $TMPDIR setting points to a tmpfs filesystem and innodb_use_native_aio is enabled, as it is by default in MySQL 5.5.4 and higher. The entry in the error log looked like:

```
101123  2:10:59  InnoDB: Operating system error number 22 in a file operation.
InnoDB: Error number 22 means 'Invalid argument'.
```

The crash occurred because asynchronous I/O is not supported on tmpfs in some Linux kernel versions. The workaround was to turn off the innodb_use_native_aio setting or use a different temporary directory. The fix causes InnoDB to turn off the innodb_use_native_aio setting automatically if it detects that the temporary file directory does not support asynchronous I/O. (Bug #13593888, Bug #11765450, Bug #58421)

- **InnoDB:** During startup, the status variable Innodb_buffer_pool_dump_status could be empty for a brief time before being initialized to the correct value not started. (Bug #13513676)

- **InnoDB:** Valgrind errors when referencing the internal function buf_LRU_scan_and_free_block() were fixed. (Bug #13491704)

- **InnoDB:** The MySQL server could halt with an assertion error:

  ```
  InnoDB: Failing assertion: page_get_n_recs(page) > 1
  ```

  Subsequent restarts could fail with the same error. The error occurred during a purge operation involving the InnoDB change buffer. The workaround was to set the configuration option innodb_change_buffering=inserts. (Bug #13413535, Bug #61104)

- **InnoDB:** The MySQL error log could contain messages like:

  ```
  InnoDB: Ignoring strange row from mysql.innodb_index_stats WHERE ...
  ```

  The fix makes the contents of the innodb_index_stats and innodb_table_stats tables case-sensitive, to properly distinguish the statistics for tables whose names differ only in letter case. Other cases were fixed where the wrong name could be selected for an index while retrieving persistent statistics. (Bug #13432465)

- **InnoDB:** References to C preprocessor symbols and macros HAVE_purify, UNIV_INIT_MEM_TO_ZERO, and UNIV_SET_MEM_TO_ZERO were removed from the InnoDB source code. They were only used in debug builds instrumented for Valgrind. They are replaced by calls to the UNIV_MEM_INVALID() macro. (Bug #13418934)

- **InnoDB:** A discrepancy could arise between the number of available InnoDB undo logs and the number of undo logs that were currently active. Now the innodb_undo_logs system variable reports the number of active undo logs, and the new Innodb_available_undo_logs status variable reports the total number of undo logs. (Bug #13255225)

- **InnoDB:** Certain CREATE TABLE statements could fail for InnoDB child tables containing foreign key definitions. This problem affected Windows systems only, with the setting lower_case_table_names=0. It was a regression from MySQL bug #55222. (Bug #13083023, Bug #60229)

- **InnoDB:** When doing a live downgrade from MySQL 5.6.4 or later, with innodb_page_size set to a value other than 16384, now the earlier MySQL version reports that the page size is incompatible with the older version, rather than crashing or displaying a “corruption” error. (Bug #13116225)

- **InnoDB:** If the server crashed during a TRUNCATE TABLE or CREATE INDEX statement for an InnoDB table, or a DROP DATABASE statement for a database containing InnoDB tables, an index could be corrupted, causing an error message when accessing the table after restart:

  ```
  InnoDB: Error: trying to load index index_name for table table_name
  InnoDB: but the index tree has been freed!
  ```

In MySQL 5.1, this fix applies to the InnoDB Plugin, but not the built-in InnoDB storage engine. (Bug #12861864, Bug #11766019)

- **InnoDB:** A DDL operation for an InnoDB table could cause a busy MySQL server to halt with an assertion error:

  ```
  InnoDB: Failing assertion: trx->error_state == DB_SUCCESS
  ```

  The error occurred if the DDL operation was run while all 1023 undo slots were in use by concurrent transactions. This error was less likely to occur in MySQL 5.5 and 5.6, because raising the number of InnoDB undo slots increased the number of simultaneous transactions (corresponding to the number of undo slots) from 1K to 128K. (Bug #12739098, Bug #62401)

- **InnoDB:** InnoDB persistent statistics gave less accurate estimates for date columns than for columns of other data types. The fix changes the way cardinality is estimated for nonunique keys, and avoids situations where identical values could be counted twice if they occurred on different index pages. (Bug #12429443)

- **InnoDB:** With 1024 concurrent InnoDB transactions running concurrently and the innodb_file_per_table setting enabled, a CREATE TABLE operation for an InnoDB table could fail. The .ibd file from the failed CREATE TABLE was left behind, preventing the table from being created later, after the load had dropped.

  The fix adds error handling to delete the erroneous .ibd file. This error was less likely to occur in MySQL 5.5 and 5.6, because raising the number of InnoDB undo slots increased the number of simultaneous transactions needed to trigger the bug, from 1K to 128K. (Bug #12400341)

- **InnoDB:** The innodb_max_purge_lag variable controls how to delay DML operations when purge operations are lagging. Previously, if an old consistent read view was detected, DML operations would not be delayed even though the purge lag exceeded the innodb_max_purge_lag setting.

  Additionally, if the innodb_max_purge_lag setting was used, situations could arise in which the DML delay time would continue to increase but not be applied right away due to the presence an old consistent read view. This could result in a lengthy DML delay when the accumulated DML delay time is eventually applied.

  This fix caps the DML delay at a maximum value, removes the consistent read check, and revises the DML delay calculation. (Bug #12407434, Bug #60776)

- **InnoDB:** Improved the accuracy of persistent InnoDB statistics for large tables. The estimate of distinct records could be inaccurate if the index tree was more than 3 levels deep. (Bug #12316365)

- **InnoDB:** Shutdown could hang with messages like this in the log:

  ```
  Waiting for purge thread  to be suspended
  ```

  After 1 hour, the shutdown times out and mysqld quits. This problem is most likely to occur with a high value for innodb_purge_threads. (Bug #11765863, Bug #58868, Bug #60939)

- **InnoDB:** When DROP TABLE failed due to all undo slots being in use, the error returned was Unknown table '...' rather than the expected Too many active concurrent transactions. (Bug #11764724, Bug #57586)

  References: See also Bug #11764668, Bug #57529.

- **InnoDB:** Server startup could produce an error for temporary tables using the InnoDB storage engine, if the path in the $TMPDIR variable ended with a / character. The error log would look like:

  ```
  120202 19:21:26  InnoDB: Operating system error number 2 in a file operation.
  ```

```
InnoDB: The error means the system cannot find the path specified.
InnoDB: If you are installing InnoDB, remember that you must create
InnoDB: directories yourself, InnoDB does not create them.
120202 19:21:26  InnoDB: Error: trying to open a table, but could not
InnoDB: open the tablespace file './t/#sql7750_1_0.ibd'!
InnoDB: Have you moved InnoDB .ibd files around without using the
InnoDB: commands DISCARD TABLESPACE and IMPORT TABLESPACE?
InnoDB: It is also possible that this is a temporary table #sql...,
InnoDB: and MySQL removed the .ibd file for this.
```

The workaround for the problem was to create a similar temporary table again, copy its .frm file to tmpdir under the name mentioned in the error message (for example, #sql123.frm) and restart mysqld with tmpdir set to its normal value without a trailing slash, for example /var/tmp. On startup, MySQL would see the .frm file and issue DROP TABLE for the orphaned temporary table. (Bug #11754376, Bug #45976)

- **Partitioning:** When creating a view from a SELECT statement that used explicit partition selection, the partition selection portion of the query was ignored. (Bug #13559657)

- **Partitioning:** Adding a partition to an already existing LIST-partitioned table did not work correctly if the number of items in the new partition was greater than 16. This could happen when trying to add a partition using an ALTER TABLE ... ADD PARTITION statement, or an ALTER TABLE ... REORGANIZE PARTITION statement.

  This 16-item limit was not apparent when using either CREATE TABLE ... PARTITION BY LIST or ALTER TABLE ... PARTITION BY LIST. (Bug #13029508, Bug #62505)

- **Partitioning:** A function internal to the code for finding matching subpartitions represented an unsigned number as signed, with the result that matching subpartitions were sometimes missed in results of queries. (Bug #12725206, Bug #61765)

  References: See also Bug #20257.

- **Partitioning:** An ALTER TABLE ... ADD PARTITION statement subsequent to ALTER TABLE ... REORGANIZE PARTITION failed on a table partitioned by HASH or KEY. (Bug #11764110, Bug #56909)

- **Replication:** Executing mysqlbinlog with the --start-position=*N* option, where *N* was equal either to 0 or to a value greater than the length of the dump file, caused it to crash.

  This issue was introduced in MySQL 5.5.18 by the fix for Bug #32228 and Bug #11747416. (Bug #13593869, Bug #64035)

- **Replication:** When starting the server, replication repositories were checked even when the --server-id was equal to 0 (the default), in spite of the fact that a valid nonzero value for --server-id must be supplied for a server that acts as either a master or a slave in MySQL replication.

  This could cause problems when trying to perform a live upgrade from MySQL 5.5, although it was possible to work around the issue by starting the server with --skip-slave-start (in addition to any other required options).

  To avoid this problem, replication repositories are now checked only when the server is started with --server-id using a nonzero value. (Bug #13427444, Bug #13504821)

- **Replication:** Formerly, the default value shown for the Port column in the output of SHOW SLAVE HOSTS was 3306 whether the port had been set incorrectly or not set at all. Now, when the slave port is not set, the actual port used by the slave is shown. This change also affects the default shown for the --report-port server option. (Bug #13333431)

- **Replication:** A race condition could occur when running multiple instances of mysqld on a single machine, when more than slave thread was started at the same time, and each such thread tried to use the same temporary file concurrently. (Bug #12844302, Bug #62055)

- **Replication:** mysqlbinlog --database=*dbname* included all SET INSERT_ID=*n* assignments from the binary log in its output, even if database *dbname* was never referenced in the binary log. This was due to the fact that COMMIT statements were not associated with any database in the binary log. Now in such cases, the current database is tracked so that only those SET INSERT_ID assignments that are made in the context of changes to tables in database *dbname* are actually printed in the mysqlbinlog output. (Bug #11746146, Bug #23894)

  References: See also Bug #23890, Bug #46998, Bug #11761686, Bug #54201, Bug #11754117, Bug #45670.

- **Replication:** Statements that wrote to tables with AUTO_INCREMENT columns based on an unordered SELECT from another table could lead to the master and the slave going out of sync, as the order in which the rows are retrieved from the table may differ between them. Such statements include any INSERT ... SELECT, REPLACE ... SELECT, or CREATE TABLE ... SELECT statement. Such statements are now marked as unsafe for statement-based replication, which causes the execution of one to throw a warning, and forces the statement to be logged using the row-based format if the logging format is MIXED. (Bug #11758263, Bug #50440)

- **Replication:** It was possible on replication slaves where FEDERATED tables were in use to get timeouts on long-running operations, such as Error 1160 Got an error writing communication packets. The FEDERATED tables did not need to be replicated for the issue to occur. (Bug #11758931, Bug #51196)

  References: See also Bug #12896628, Bug #61790.

- **Replication:** On Windows replication slave hosts, STOP SLAVE took an excessive length of time to complete when the master was down. (Bug #11752315, Bug #43460)

- mysqldump tried to execute SET statements as SET OPTION, which failed when used against 5.6 or higher servers because the deprecated OPTION keyword has been removed from SET syntax. (Bug #13813473)

- The optimizer did not perform constant propagation for views, so a query containing views resulted in a less efficient execution plan than the corresponding query using only base tables. (Bug #13783777)

- After using an ALTER TABLE statement to change the KEY_BLOCK_SIZE property for an InnoDB table, for example when switching from an uncompressed to a compressed table, subsequent server restarts could fail with a message like:

  ```
  InnoDB: Error: data file path/ibdata2 uses page size 1024,
  InnoDB: but the only supported page size in this release is=16384
  ```

  This issue is a regression introduced in MySQL 5.5.20. (Bug #13698765, Bug #64160)

- _mi_print_key() iterated one time too many when there was a NULL bit, resulting in Valgrind warnings. (Bug #13686970)

- In debug builds, a Debug Sync timeout warning was treated as an error, causing an assertion to be raised. (Bug #13688248)

- A memory leak could occur for queries containing a subquery that used GROUP BY on an outer column. (Bug #13724099)

- If during server startup a signal such as SIGHUP was caught prior to full server initialization, the server could crash. This was due to a race condition between the signal handler thread and the main thread performing server initialization. To prevent this from happening, signal processing is now suspended until full initialization of all server components has been completed successfully. (Bug #13608371, Bug #62311)

- A SELECT from a subquery that returned an empty result could itself fail to return an empty result as expected. (Bug #13651009, Bug #13650418)

- An aggregated expression of type `MIN()` or `MAX()` should return `NULL` but could instead return the empty set if the query was implicitly grouped and there was no `HAVING` clause that evaluates to `FALSE`. (Bug #13599013)
- The shared version of `libmysqlclient` did not export these functions for linking by client programs: `get_tty_password()`, `handle_options()`, `my_print_help()`. (Bug #13604121)
- For debug builds, negative values with a zero integer part and nonzero fractional part (such as -0.1111) were not detected, so the negative fractional part was later cast to a large unsigned number and raised an assertion. (Bug #13616434)
- Pushing down to `InnoDB` an index condition that called a stored function resulted in a server crash. This kind of condition is no longer pushed down. (Bug #13655397)
- Date-handling code could raise an assertion attempting to calculate the number of seconds since the epoch. (Bug #13545236)
- Left join queries could be incorrectly converted to inner joins and return erroneous result sets. (Bug #13595212)
- Converting a string ending with a decimal point (such as `'1.'`) to a floating-point number raised a data truncation warning. (Bug #13500371)
- The Performance Schema instrumentation for stages did not fully honor the `ENABLED` column in the `schema.setup_instruments` table. (Bug #13509513)
- `SELECT` statements failed for the `EXAMPLE` storage engine. (Bug #13511529)

  References: This bug was introduced by Bug #11746275.
- For queries that used a join type of `ref_or_null`, the optimizer could skip the filesort operation and sort the results incorrectly. (Bug #13531865)
- For some queries, a filesort operation was done even when the result contained only a single row and needed no sorting. (Bug #13529048)
- The optimizer could return an incorrect select limit in some cases when a query included no explicit `LIMIT` clause. (Bug #13528826)
- In some cases, the optimizer failed to use a covering index when that was possible and read data rows instead. (Bug #13514959)
- Use of an uninitialized `TABLE_SHARE` member could cause a server crash. (Bug #13489996)
- A query that used an index on a `CHAR` column referenced in a `BETWEEN` clause could return invalid results. (Bug #13463488, Bug #63437)
- Expressions that compared a `BIGINT` column with any non-integer constant were performed using integers rather than decimal or float values, with the result that the constant could be truncated. This could lead to any such comparison that used `<`, `>`, `<=`, `>=`, `=`, `!=`/`<>`, `IN`, or `BETWEEN` yielding false positive or negative results. (Bug #13463415, Bug #11758543, Bug #63502, Bug #50756)
- Enabling index condition pushdown could cause performance degradation. (Bug #13430436)
- For debug builds, enabling `optimizer_trace` could cause an assertion to be raised. (Bug #13430443)
- When the optimizer performed conversion of `DECIMAL` values while evaluating range conditions, it could produce incorrect results. (Bug #13453382)
- On Windows, rebuilds in a source distribution failed to create the initial database due to insufficient cleanup from the previous run or failure to find the proper server executable. (Bug #13431251)

- An application linked against libmysqld could crash in debug mode with a stack smashing detected error if it tried to connect without specifying the user name. (Bug #13460909)
- Implicitly grouped queries with a const table and no matching rows could return incorrect results. (Bug #13430588)
- Some outer joins that used views as inner tables did not evaluate conditions correctly. (Bug #13464334)
- Instantiating a derived table for a query with an empty result caused a server crash. (Bug #13457552)
- Temporary MyISAM tables (unlike normal MyISAM tables) did not use the dynamic row format when they contained VARCHAR columns, resulting in larger temporary files (and more file I/O) than necessary. (Bug #13350136)
- When a fixed-width row was inserted into a MyISAM temporary table, the entire content of the record buffer was written to the table, including any trailing space contained in VARCHAR columns, the issue being that this trailing space could be uninitialized. This problem has been resolved by insuring that only the bytes actually used to store the VARCHAR (and none extra) are copied and inserted in such cases. (Bug #13389854)
- When merging ranges that effectively resulted in a full index scan, the optimizer did not discard the range predicate as unneeded. (Bug #13354910)
- Fractional seconds parts were lost for certain UNION ALL queries. (Bug #13375823)
- When executing EXPLAIN, it was assumed that only the default multi-range read implementation could produce an ordered result; this meant that when a query on a table that used a storage engine providing its own sorted MRR, it was ignored, so that EXPLAIN failed to report Using MRR even when a multi-range read was used. (Bug #13330645)
- Some multiple-table updates could update a row twice. (Bug #13095459)
- Performance Schema idle event timings were not normalized to the same units as wait timings. (Bug #13018537)
- In MySQL 5.6.3, a number of status variables were changed to longlong types so that they would roll over much later. However, the format string used by mysqladmin status to print Queries per second values did not reflect this, causing such values to be misreported. (Bug #12990746)

  References: See also Bug #42698. This bug was introduced by Bug #11751727.
- For debug builds, two assertions could be raised erroneously for UPDATE statements. (Bug #12912171)
- When the result of a stored function returning a non-integer type was evaluated for NULL, an incorrect type warning (Warning 1292 Truncated incorrect INTEGER value) is generated, although such a test for NULL should work with any type. This could cause stored routines not handling the warning correctly to fail.

  The issue could be worked around by wrapping the result in an expression, using a function such as CONCAT(). (Bug #12872824, Bug #62125)
- A query that used an aggregate function such as MAX() or MIN() of an index with NOT BETWEEN in the WHERE clause could fail to match rows, thus returning an invalid result. (Bug #12773464, Bug #61925)
- When running mysqldump with both the --single-transaction and --flush-logs options, the flushing of the log performed an implicit COMMIT (see Statements That Cause an Implicit Commit), causing more than one transaction to be used and thus breaking consistency. (Bug #12809202, Bug #61854)

- With ONLY_FULL_GROUP_BY SQL mode enabled, columns that were not aggregated in the select list or named in a GROUP BY were incorrectly permitted in ORDER BY. (Bug #12626418)
- Mishandling of NO_BACKSLASH_ESCAPES SQL mode within stored procedures on slave servers could cause replication failures. (Bug #12601974)
- LOAD INDEX INTO CACHE could cause a server exit if the index cache was too small. (Bug #12361113)
- Passing a user variable as an argument to GROUP_CONCAT() could cause a server exit if the variable value changed during query execution. (Bug #12408412)
- 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)
- Attempting to execute ALTER TABLE on a temporary MERGE table having an underlying temporary table rendered the MERGE table unusable, unless the ALTER TABLE specified a new list of underlying tables. (Bug #11764786, Bug #57657)
- When used with the --xml option, mysqldump --routines failed to dump any stored routines, triggers, or events. (Bug #11760384, Bug #52792)
- A HAVING clause in a query using MIN() or MAX() was sometimes ignored. (Bug #11760517, Bug #52935)

  References: See also Bug #11758970, Bug #51242, Bug #11759718, Bug #52051.
- Previously, .OLD files were not included among the files deleted by DROP DATABASE. Files with this extension are now also deleted by the statement. (Bug #11751736, Bug #42708)
- A prepared statement using a view whose definition changed between preparation and execution continued to use the old definition, which could cause the prepared statement to return incorrect results. (Bug #11748352, Bug #36002)
- If an attempt to initiate a statement failed, the issue could not be reported to the client because it was not prepared to receive any error messages prior to the execution of any statement. Since the user could not execute any queries, they were simply disconnected without providing a clear error.

  After the fix for this issue, the client is prepared for an error as soon as it attempts to initiate a statement, so that the error can be reported prior to disconnecting the user. (Bug #11755281, Bug #47032)
- It was possible in the event of successive failures for mysqld_safe to restart quickly enough to consume excessive amounts of CPU. Now, on systems that support the sleep and date system utilities, mysqld_safe checks to see whether it has restarted more than 5 times in the current second, and if so, waits 1 second before attempting another restart. (Bug #11761530, Bug #54035)
- The embedded server crashed when argc = 0. (Bug #57931, Bug #12561297)
- Enabling myisam_use_mmap could cause the server to crash. (Bug #48726, Bug #11756764)
- The handle_segfault() signal-handler code in mysqld could itself crash due to calling unsafe functions. (Bug #54082, Bug #11761576)
- Stored functions could produce an error message that referred to ORDER BY even though the offending statement within the function had no such clause. (Bug #35410, Bug #11748187)
- UPDATE IGNORE returned an incorrect count for number of rows updated when there were duplicate-key conflicts in a multiple-table update. (Bug #59715, Bug #11766576)
- Locale information for FORMAT() function instances was lost in view definitions. (Bug #63020, Bug #13344643)

- The stored routine cache was subject to a small memory leak that over time or with many routines being used could result in out-of-memory errors.

  The fix for this issue also introduces a new global server system variable stored_program_cache which can be used for controlling the size of the stored routine cache. (Bug #44585, Bug #11753187)

- On Windows, the server incorrectly constructed the full path name of the plugin binary for INSTALL PLUGIN and CREATE FUNCTION ... SONAME. (Bug #45549, Bug #11754014)

- Using myisamchk with the sort recover method to repair a table having fixed-width row format could cause the row pointer size to be reduced, effectively resulting in a smaller maximum data file size. (Bug #48848, Bug #11756869)

- myisam_sort_buffer_size could not be set larger than 4GB on 64-bit systems. (Bug #45702, Bug #11754145)

- The optimizer mishandled STRAIGHT_JOIN used with nested joins; for example, by not evaluating tables in the specified order. (Bug #59487, Bug #11766384, Bug #43368, Bug #11752239, Bug #60080, Bug #11766858)

- A subquery involved in a comparison requiring a character set conversion caused an error that resulted in a server crash. (Bug #59185, Bug #11766143)

- mysqlhotcopy failed for databases containing views. (Bug #62472, Bug #13006947, Bug #12992993)

- Assigning the result of a subquery to a user variable raised an assertion when the outer query included DISTINCT and GROUP BY. (Bug #57196, Bug #11764371)

- On Windows, pasting multiple-line input including a CRLF terminator on the last line into the mysql client resulted in the first character of the last line being changed, resulting in erroneous statements. Handling of newlines in pasted input was also incorrect. (Bug #60901, Bug #12589167, Bug #64104, Bug #13639107)

- Invalid memory reads could occur when cmp_item_sort_string::store_value() tried to refer to a temporary value that could be changed or deleted by other functions. (Bug #57510, Bug #11764651)

- Due to improper locking, concurrent inserts into an ARCHIVE table at the same time as repair and check operations on the table resulted in table corruption. (Bug #37280, Bug #11748748)

- For MEMORY tables, a scan of a HASH index on a VARCHAR column could fail to find some rows if the index was on a prefix of the column. (Bug #47704, Bug #11755870)

- If tables were locked by LOCK TABLES ... READ in another session, SET GLOBAL read_only = 1 failed to complete. (Bug #57612, Bug #11764747)

- The VIO description string was initialized even for connections where it was unneeded. (Bug #62285, Bug #12951586)

- The contents of the shared and shared-compat RPM packages had been changed in versions 5.5.6 and 5.6.1 to avoid the overlap which they traditionally had (and still have in MySQL 5.0 and 5.1). However, the RPM meta information had not been changed in accordance, and so RPM still assumed a conflict between shared and shared-compat RPM packages. This has been fixed. (Bug #60855, Bug #12368215)

  References: See also Bug #56150.

- The result of SUBSTRING_INDEX() could be missing characters when used as an argument to conversion functions such as LOWER(). (Bug #60166, Bug #11829861)

- Some debugging information was written to the buffer after a flush, resulting in the information not appearing until the next flush. (Bug #64048, Bug #13608112)

- A confusing CREATE TABLE error message was improved. (Bug #54963, Bug #11762377)
- Under some circumstances, the result of SUBSTRING_INDEX() incorrectly depended on the contents of the previous row. (Bug #42404, Bug #11751514)
- Setting an event to DISABLED status and with the ON COMPLETION NOT PRESERVE attribute caused it to be dropped at the next server restart. (Bug #37666, Bug #11748899)
- For comparisons containing out-of-range constants, the optimizer permitted warnings to leak through to the client, even though it accounted for the range issue internally. (Bug #56962, Bug #11764155)
- The decision about how to sort a result set could be reported incorrectly by EXPLAIN for some statements, causing Using filesort or Using temporary to be reported when they should not have been or vice versa. This could occur for statements that included index hints, that had the form SELECT SQL_BIG_RESULT ... GROUP BY, that used SQL_CALC_FOUND_ROWS with LIMIT, or that used GROUP BY, ORDER BY, and LIMIT.

Changes in MySQL 5.6.4 (2011-12-20, Milestone 7)

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.

Condition Handler Changes

- **Incompatible Change:** MySQL now supports the GET DIAGNOSTICS statement. GET DIAGNOSTICS provides applications a standardized way to obtain information from the diagnostics area, such as whether the previous SQL statement produced an exception and what it was. For more information, see GET DIAGNOSTICS Syntax.

  In addition several deficiencies in condition handler processing rules were corrected so that MySQL behavior is more like standard SQL:

  - Block scope is used in determining which handler to select. Previously, a stored program was treated as having a single scope for handler selection.
  - Condition precedence is more accurately resolved.
  - Diagnostics area clearing has changed. Bug #55843 caused handled conditions to be cleared from the diagnostics area before activating the handler. This made condition information unavailable within the handler. Now condition information is available to the handler, which can inspect it with the GET DIAGNOSTICS statement. The condition information is cleared when the handler exits, if it has not already been cleared during handler execution.
  - Previously, handlers were activated as soon as a condition occurred. Now they are not activated until the statement in which the condition occurred finishes execution, at which point the most appropriate handler is chosen. This can make a difference for statements that raise multiple conditions, if a condition raised later during statement execution has higher precedence than an earlier condition and there are handlers in the same scope for both conditions. Previously, the handler for the first condition raised would be chosen, even if it had a lower precedence than other handlers. Now the handler for the condition with highest precedence is chosen, even if it is not the first condition raised by the statement.
  - Issues that caused server crashes resulting from incorrect handler call stack processing were fixed.

The work just described involved several condition-handler bug fixes:

- The RETURN statement did not clear the diagnostics area as it should have. Now the diagnostics area is cleared before executing RETURN. This prevents a condition in a nested function call from incorrectly propagating to an outer scope. It also means there is no way to return an SQL warning from a stored function. This change is not backward compatible, but the resulting behavior is more like standard SQL.
- When an SQL HANDLER was activated, the handled condition was immediately removed from the diagnostics area. Consequently, any SQL diagnostic statement executed in the handler was unable to examine the condition that triggered the handler.
- If multiple handlers existed at the same level within a stored program, the wrong one could be chosen.
- If an error occurred in a context where different handlers were present at different levels of nesting, an outer handler could be chosen rather than the innermost one.

For more information, see Scope Rules for Handlers. (Bug #12951117, Bug #38806, Bug #11749343, Bug #55852, Bug #11763171, Bug #61392, Bug #12652873, Bug #11660, Bug #11745196, Bug #48637, Bug #11756690)

Fractional Seconds Handling

- **Incompatible Change:** MySQL now permits fractional seconds for TIME, DATETIME, and TIMESTAMP values, with up to microseconds (6 digits) precision. To define a column that includes a fractional seconds part, use the syntax *type_name*(*fsp*), where *type_name* is TIME, DATETIME, or TIMESTAMP, and *fsp* is the fractional seconds precision. For example:

  ```
  CREATE TABLE t1 (t TIME(3), dt DATETIME(6));
  ```

  The *fsp* value, if given, must be in the range 0 to 6. A value of 0 signifies that there is no fractional part. If omitted, the default precision is 0. (This differs from the standard SQL default of 6, for compatibility with previous MySQL versions.)

  The following items summarize the implications of this change. See also Fractional Seconds in Time Values.

  - For TIME, DATETIME, and TIMESTAMP columns, the encoding and storage requirements in new tables differ from such columns in tables created previously because these types now include a fractional seconds part.
  - Due to changes to encoding and storage requirements for DATETIME and TIMESTAMP types, importing pre-MySQL 5.6.4 InnoDB tables that contain DATETIME and TIMESTAMP types into MySQL 5.6.4 (or later) requires a workaround procedure which is described in the “Server Changes” section of Upgrading from MySQL 5.5 to 5.6 .
  - Syntax for temporal literals now produces temporal values: DATE '*str*', TIME '*str*', and TIMESTAMP '*str*', and the ODBC-syntax equivalents. The resulting value includes a trailing fractional seconds part if specified. Previously, the temporal type keyword was ignored and these constructs produced the string value. See Standard SQL and ODBC Date and Time Literals
  - Functions that take temporal arguments accept values with fractional seconds. Return values from temporal functions include fractional seconds as appropriate.
  - Three INFORMATION_SCHEMA tables, COLUMNS, PARAMETERS, and ROUTINES, now have a DATETIME_PRECISION column. Its value is the fractional seconds precision for TIME, DATETIME, and TIMESTAMP columns, and NULL for other data types.

- The C API accommodates fractional seconds as follows:
  - In the `MYSQL_FIELD` column metadata structure, the `decimals` member indicates the fractional seconds precision for `TIME`, `DATETIME`, and `TIMESTAMP` columns. Clients can determine whether a result set temporal column has a fractional seconds part by checking for a nonzero `decimals` value in the corresponding `MYSQL_FIELD` structure. Previously, the `decimals` member indicated the precision for numeric columns and was zero otherwise.

  - In the `MYSQL_TIME` structure used for the binary protocol, the `second_part` member indicates the microseconds part for `TIME`, `DATETIME`, and `TIMESTAMP` columns. Previously, the `second_part` member was unused.

In some cases, previously accepted syntax may produce different results. The following items indicate where existing code may need to be changed to avoid problems:

- Some expressions produce results that differ from previous results. Examples: The `timestamp` system variable returns a value that includes a microseconds fractional part rather than an integer value. Functions that return a result that includes the current time (such as `CURTIME()`, `SYSDATE()`, or `UTC_TIMESTAMP()`) interpret an argument as an *`fsp`* value and the return value includes a fractional seconds part of that many digits. Previously, these functions permitted an argument but ignored it.

- `TIME` values are converted to `DATETIME` by adding the time to the current date. (This means that the date part of the result differs from the current date if the time value is outside the range from `'00:00:00'` to `'23:59:59'`.) Previously, conversion of `TIME` values to `DATETIME` was unreliable. See Conversion Between Date and Time Types.

- `TIMESTAMP(`*`N`*`)` was permitted in old MySQL versions, but *`N`* was a display width rather than fractional seconds precision. Support for this behavior was removed in MySQL 5.5.3, so applications that are reasonably up to date should not be subject to this issue. Otherwise, code must be rewritten.

> **Note**
>
> There may be problems replicating from a master server that understands fractional seconds to an older slave that does not:
>
> - For `CREATE TABLE` statements containing columns that have an *`fsp`* value greater than 0, replication will fail due to parser errors.
>
> - Statements that use temporal data types with an *`fsp`* value of 0 will work for with statement-based logging but not row-based logging. In the latter case, the data types have binary formats and type codes on the master that differ from those on the slave.
>
> - Some expression results will differ on master and slave. For example, expressions that involve the `timestamp` system variable or functions that return the current time have different results, as described earlier.

(Bug #8523, Bug #11745064)

InnoDB Notes

- MySQL now supports `FULLTEXT` indexes for `InnoDB` tables. The core syntax is very similar to the `FULLTEXT` capability from earlier releases, with the `CREATE TABLE` and `CREATE INDEX` statements, and `MATCH() ... AGAINST()` clause in the `SELECT` statement. The new `@` operator allows proximity searches for terms that are near each other in the document. The detailed search processing is controlled by a new set of configuration options: `innodb_ft_enable_stopword`, `innodb_ft_server_stopword_table`, `innodb_ft_user_stopword_table`, innodb_ft_cache_size, innodb_ft_min_token_size, and innodb_ft_max_token_size. You can monitor the workings of the InnoDB full-text search system by querying new INFORMATION_SCHEMA tables: innodb_ft_default_stopword, innodb_ft_index_table, innodb_ft_index_cache, innodb_ft_config, innodb_ft_deleted, and innodb_ft_being_deleted.

Optimizer Notes

- These query optimizer improvements were implemented:

  - The optimizer detects and optimizes away these useless query parts within IN/ALL/SOME/EXISTS subqueries:

    - DISTINCT

    - GROUP BY, if there is no HAVING clause and no aggregate functions

    - ORDER BY, which has no effect because LIMIT is not supported in these subqueries

Performance Schema Notes

- The Performance Schema has these additions:

  - The Performance Schema now permits instrument and consumer configuration at server startup, which previously was possible only at runtime using UPDATE statements for the setup_instruments and setup_consumers tables. This change was made because configuration at runtime is too late to disable instruments that have already been initialized during server startup. For example, the wait/sync/mutex/sql/LOCK_open mutex is initialized once during server startup, so attempts to disable the corresponding instrument at runtime have no effect.

    To control an instrument at server startup, use an option of this form:

    ```
    --performance-schema-instrument='instrument_name=value'
    ```

    Here, *instrument_name* is an instrument name such as wait/sync/mutex/sql/LOCK_open, and *value* is one of these values:

    - off, false, or 0: Disable the instrument

    - on, true, or 1: Enable and time the instrument

    - counted: Enable and count (rather than time) the instrument

    Each --performance-schema-instrument option can specify only one instrument name, but multiple instances of the option can be given to configure multiple instruments. In addition, patterns are permitted in instrument names to configure instruments that match the pattern. To configure all condition synchronization instruments as enabled and counted, use this option:

    ```
    --performance-schema-instrument='wait/synch/cond/%=counted'
    ```

    To disable all instruments, use this option:

    ```
    --performance-schema-instrument='%=off'
    ```

    Longer instrument name strings take precedence over shorter pattern names, regardless of order. For information about specifying patterns to select instruments, see Naming Instruments or Consumers for Filtering Operations.

An unrecognized instrument name is ignored. It is possible that a plugin installed later may create the instrument, at which time the name is recognized and configured.

To control a consumer at server startup, use an option of this form:

```
--performance-schema-consumer_consumer_name=value
```

Here, *consumer_name* is a consumer name such as events_waits_history, and *value* is one of these values:

- off, false, or 0: Do not collect events for the consumer
- on, true, or 1: Collect events for the consumer

For example, to enable the events_waits_history consumer, use this option:

```
--performance-schema-consumer-events-waits-history=on
```

The permitted consumer names can be found by examining the setup_consumers table. Patterns are not permitted.

Along with the preceding changes to permit configuration at server startup, the default instrument and consumer configuration has changed. Previously, all instruments and consumers were enabled by default. Now, instruments are disabled except the statement, I/O, and idle instruments. Consumers are disabled except the global, thread, and current-statement consumers. These changes produce a default configuration with a low overhead.

- Tables that have an EVENT_ID column now also have an END_EVENT_ID column to support determination of nested event relationships:

  - events_waits_current, events_waits_history, events_waits_history_long
  - events_stages_current, events_stages_history, events_stages_history_long
  - events_statements_current, events_statements_history, events_statements_history_long

  As before, EVENT_ID is populated with the thread current event counter when an event starts. In addition, END_EVENT_ID is NULL until the event ends, at which point it is set to the new thread current event counter. This permits the relationship “event B is included in event A” to be determined using the following expression, without having to follow each inclusion relationship using NESTING_EVENT_ID:

  ```
  A.EVENT_ID <= B.EVENT_ID AND B.END_EVENT_ID <= A.END_EVENT_ID
  ```

- The Performance Schema aggregates file I/O operations in two places, the events_waits_summary_*xxx* tables and the file_summary_*xxx* tables. It was possible to join the events_waits_summary_global_by_event_name table to the file_summary_by_event_name by using the EVENT_NAME column. However, it was not possible to do the same with the events_waits_summary_by_instance and file_summary_by_instance tables because the former uses OBJECT_INSTANCE_BEGIN as the instance identifier and the latter uses FILE_NAME. This means that it was possible to obtain both file I/O latency and usage per file, but not to correctly correlate latency to usage when there was more than one form of file (such as multiple redo logs, table files, and so forth).

  To address this issue, the file_summary_by_instance table now has an OBJECT_INSTANCE_BEGIN column. In addition, both file_summary_by_instance and file_summary_by_event_name have additional aggregation columns (such as timer wait information), which in many cases makes it possible to obtain the desired summary information without need for a join at all.

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

For more information, see MySQL Performance Schema.

Functionality Added or Changed

- **Performance; InnoDB:** You can now set the InnoDB page size for uncompressed tables to 8KB or 4KB, as an alternative to the default 16KB. This setting is controlled by the innodb_page_size configuration option. You specify the size when creating the MySQL instance. All InnoDB tablespaces within an instance share the same page size. Smaller page sizes can help to avoid redundant or inefficient I/O for certain combinations of workload and storage devices, particularly SSD devices with small block sizes.

- **Performance; InnoDB:** New optimizations apply to read-only InnoDB transactions. See Optimizing InnoDB Read-Only Transactions for details. The new optimizations make autocommit more applicable to InnoDB queries than before, as a way to signal that a transaction is read-only because it is a single-statement SELECT.

- **Replication:** Previously, replication slaves could connect to the master server only through master accounts that use native authentication. Now replication slaves can also connect through master accounts that use nonnative authentication if the required client-side plugin is installed on the slave side in the directory named by the slave plugin_dir system variable. (Bug #12897501)

- The optimizer trace capability now tracks temporary tables created by the server during statement execution. (Bug #13400713)

- Performance of metadata locking operations on Windows XP systems was improved by instituting a cache for metadata lock objects. This permits the server to avoid expensive operations for creation and destruction of synchronization objects on XP. A new system variable, metadata_locks_cache_size, permits control over the size of the cache. The default size is 1024. (Bug #12695572)

- Upgrading from an Advanced GPL RPM package to an Advanced RPM package did not work. Now on Linux it is possible to use rpm -U to replace any installed MySQL product by any other of the same release family. It is not necessary to remove the old produce with rpm -e first. (Bug #11886309)

- MEMORY table creation time is now available in the CREATE_TIME column of the INFORMATION_SCHEMA.TABLES table and the Create_time column of SHOW TABLE STATUS output. (Bug #51655, Bug #11759349)

- Previously, MySQL servers from 5.1 and up refused to open ARCHIVE tables created in 5.0 because opening them caused a server crash. The server now can open 5.0 ARCHIVE tables, and REPAIR TABLE updates them to the format used in 5.6. However, the recommended upgrade procedure is still to dump 5.0 ARCHIVE tables before upgrading and reload them after upgrading. (Bug #48633, Bug #11756687)

- The make_win_bin_dist script is no longer used and has been removed from MySQL distributions and the manual. (Bug #58241)

- Error messages that referred only to an error code now also include the corresponding error description. (Bug #48348, Bug #11756433)

- The MySQL code base was changed to permit use of the C++ Standard Library and to enable exceptions and runtime type information (RTTI). This change has the following implications:

  - Libraries and executables depend on some C++ standard library. On Linux, this has not been the case previously. On Solaris, the default dependency has changed from the default library to libstlport, which is now included with binary distributions for users whose system does not have it.

- The -fno-rtti and -fno-exceptions options are no longer used to build plugins, such as storage engines. Users who write their own plugins should omit these options if they were using tem.
- C++ users who compile from source should set CXX to a C++ compiler rather than a C compiler. For example, use g++ rather than gcc. This includes the server as well as client programs.
- mysql_config now has a --cxxflags option. This is like the --cflags option, but produces flags appropriate for a C++ compiler rather than a C compiler.
- User-defined functions can be written in C++ using standard library features.

Bugs Fixed

- **Security Enhancement; Replication:** The START SLAVE statement now accepts USER and PASSWORD options. By default, MySQL native authentication is used, and the user name and password are stored in the master.info repository. This behavior can be overridden by additionally specifying the name (DEFAULT_AUTH) and location (PLUGIN_DIR) of an authentication plugin when issuing START SLAVE. .

  As part of this change, warnings are now issued in the following cases:

  1. If START SLAVE USER="..." PASSWORD="..." or CHANGE MASTER TO MASTER_USER="..." MASTER_PASSWORD="..." is executed using an unencrypted connection, the warning message Sending passwords in plain text without SSL/TLS is extremely insecure is generated (**ER_INSECURE_PLAIN_TEXT**).
  2. If the user name and password are stored in or read from the master.info repository in the course of executing CHANGE MASTER TO, a warning message is printed out to the error log: Storing MySQL user name or password information in the master.info repository is not secure and is therefore not recommended (**ER_INSECURE_CHANGE_MASTER**).

  Note

  The text of a running START SLAVE statement, including values for USER and PASSWORD, can be seen in the output of a concurrent SHOW PROCESSLIST statement. The complete text of a CHANGE MASTER TO statement is also visible to SHOW PROCESSLIST.

  See also Pluggable Authentication. (Bug #13083642)

- **Performance; InnoDB:** The process of deallocating the InnoDB Adaptive Hash Index was made faster, during shutdown or when turning off the AHI with the statement:

  ```
  SET GLOBAL innodb_adaptive_hash_index=OFF;
  ```

  (Bug #13006367, Bug #62487)

- **Performance; InnoDB:** This fix improves the performance of instrumentation code for InnoDB buffer pool operations. (Bug #12950803, Bug #62294)
- **Performance; InnoDB:** This fix improved the efficiency and concurrency of freeing pages in the InnoDB buffer pool when performing a DROP TABLE for an InnoDB table when the innodb_file_per_table option is enabled.

  This change is most noticeable for systems with large buffer pools. During the drop operation, one traversal of the buffer pool memory structure is changed from the LRU list (the entire buffer pool) to the flush list (a much smaller structure). The LRU scanning is reduced, but not entirely eliminated.

The buffer pool mutex is also released periodically, so that if the drop operation takes significant time, other threads can proceed concurrently. (Bug #11759044, Bug #51325)

- **Incompatible Change; Replication:** The statements in the following list are now marked as unsafe for statement-based replication. This is due to the fact that each of these statements depends on the results of a SELECT statement whose order cannot always be determined. When using STATEMENT logging mode, a warning is issued in the binary log for any of these statements; when using MIXED logging mode, the statement is logged using the row-based format.

  - INSERT ... SELECT ... ON DUPLICATE KEY UPDATE
  - REPLACE ... SELECT
  - CREATE TABLE ... IGNORE SELECT
  - CREATE TABLE ... REPLACE SELECT
  - INSERT IGNORE ... SELECT
  - UPDATE IGNORE

  When upgrading, you should note the use of these statements in your applications, keeping in mind that a statement that inserts or replaces rows obtained from a SELECT can take up many times as much space in the binary log when logged using row-based format than when only the statement itself is logged. Depending on the number and size of the rows selected and inserted (or replaced) by any such statements, the difference in size of the binary log after the logging of these statements is switched from statement-based to row-based can potentially be several orders of magnitude. See Advantages and Disadvantages of Statement-Based and Row-Based Replication. (Bug #11758262, Bug #50439)

- **Incompatible Change:** Previously, “Aborted connection” errors were written to the error log based on the session value of log_warnings, which permitted users with minimal privileges to cause many messages to be written to the log unless restricted by the MAX_CONNECTIONS_PER_HOUR resource limit. Now this logging is based on the global log_warnings variable. There are no remaining uses of the session log_warnings variable, so it has been removed and the variable now has only a global value. (Bug #53466, Bug #11761014)

- **Important Change; InnoDB:** If an ALTER TABLE statement failed for an InnoDB table due to an error code from an underlying file-renaming system call, InnoDB could lose track of the .ibd file for the table. This issue only occurred when the innodb_file_per_table configuration option was enabled, and when the low-level error persisted through thousands of retry attempts. In MySQL 5.1, this issue applied to the InnoDB Plugin but not the built-in InnoDB storage engine.

  For example, if you encounter an error like the following:

  ```
  mysql> alter table sb2 add column d2 int;
  ERROR 1025 (HY000): Error on rename of './sbtest/#sql-1eb9_1' to
  './sbtest/sb2' (errno: -1)
  ```

  you might be able to access the #sql* table by copying a .frm file from a table with an identical schema. The table name to use for the .frm filewould be `sbtest.#mysql50##sql-1eb9_1` in the preceding example. (Bug #12884631, Bug #62146)

- **Important Change; Replication:** Setting an empty user in a CHANGE MASTER TO statement caused an invalid internal result and is no longer permitted. Trying to use MASTER_USER='' or setting MASTER_PASSWORD while leaving MASTER_USER unset causes the statement to fail with an error. (Bug #13427949)

- **InnoDB:** An internal deadlock could occur within InnoDB, on a server doing a substantial amount of change buffering for DML operations, particularly DELETE statements. (Bug #13340047)

- **InnoDB:** Fixed a compilation problem that affected the InnoDB source code with gcc 4.6.1. The affected InnoDB source file was btr/btr0cur.c. (Bug #13116045)

- **InnoDB:** Querying the INFORMATION_SCHEMA.INNODB_SYS_TABLESTATS table could cause the server to halt with an assertion error, in debug builds only. (Bug #12960058)

- **InnoDB:** Valgrind errors when building with the settings innodb_checksum_algorithm=innodb and innodb_checksum_algorithm=crc32 were fixed. (Bug #12939557)

- **InnoDB:** Unused functions were removed from the internal InnoDB code related to mini-transactions, to clarify the logic. (Bug #12626794, Bug #61240)

- **InnoDB:** Lookups using secondary indexes could give incorrect matches under a specific set of conditions. The conditions involve an index defined on a column prefix, for a BLOB or other long column stored outside the index page, with a table using the Barracuda file format. (Bug #12601439, Bug #12543666)

- **InnoDB:** An UPDATE statement for an InnoDB table could hang. The issue affects tables using the Barracuda file format and having multiple indexes on column prefixes. The size of an undo log record could exceed the page size, even though the total size of the column prefixes was less than the page size (usually 16KB). In MySQL 5.5 and higher, this error is now reported using the new code ER_UNDO_RECORD_TOO_BIG. In MySQL 5.1 with the InnoDB Plugin, this error is reported using the existing code ER_TOO_BIG_ROWSIZE. (Bug #12547647)

- **InnoDB:** This fix corrects cases where the MySQL server could hang or abort with a long semaphore wait message. (This is a different issue than when these symptoms occurred during a CHECK TABLE statement.) (Bug #11766591, Bug #59733)

- **InnoDB:** Issuing INSERT...ON DUPLICATE KEY statements for InnoDB tables from concurrent threads could cause a deadlock, particularly with the INSERT...ON DUPLICATE KEY UPDATE form. The problem could also be triggered by issuing multiple INSERT IGNORE statements. The fix avoids deadlocks caused by the same row being accessed by more than one transaction. Deadlocks could still occur when multiple rows are inserted and updated simultaneously by different transactions in inconsistent order; those types of deadlocks require the standard error handling on the application side, of re-trying the transaction. (Bug #11759688, Bug #52020, Bug #12842206)

- **Partitioning:** CHECKSUM TABLE returned 0 for a partitioned table unless the statement was used with the EXTENDED option. (Bug #11933226, Bug #60681)

- **Partitioning:** Error 1214 (ER_TABLE_CANT_HANDLE_FT), given when trying to use a FULLTEXT index with a partitioned table, displayed the misleading text The used table type doesn't support FULLTEXT indexes was misleading and has been replaced with Error 1752 (ER_FULLTEXT_NOT_SUPPORTED_WITH_PARTITIONING) which shows the more accurate FULLTEXT index is not supported for partitioned tables. (Bug #11763825, Bug #56590)

- **Partitioning:** Using ALTER TABLE to remove partitioning from a valid MyISAM table could corrupt it. (Bug #52599, Bug #11760213)

- **Replication:** The value set for the slave_parallel_workers system variable (or the corresponding --slave-parallel-workers server option) was not always honored correctly; in such cases a random value was used. (Bug #13334470)

- **Replication:** Execution of LOAD DATA on a MyISAM table having an after-insert trigger which wrote into an InnoDB table caused multi-threaded statement-based replication to abort with error 1742 (Cannot execute the current event group in the parallel mode). (Bug #12982188)

- **Replication:** Several warnings and informational messages were revised for typographic errors and clarity. (Bug #12947248, Bug #12978113)

- **Replication:** When a statement containing a large number of rows to be applied on a slave table that does not contain a primary key, a considerable amount of time can be needed to find and change all the rows that are to be changed. The current fix helps diagnose this issue by printing a message to the error log if the execution time for a given statement replicated using row-based replication takes more than 60 seconds. `log_warnings` must be greater than 1 for this message to be printed to the error log. (Bug #11760927, Bug #53375)

- **Replication:** `mysqlbinlog --hexdump` printed the last row of the hex dump incorrectly, in two ways:

  1. If the length of the last row was eight bytes, the end of the previous row was copied to the end of the last row, padding the last row to full length.

  2. If the length of the last row was less than sixteen bytes, its textual representation was not aligned with that of previous rows.

  (Bug #11747887, Bug #34386)

- **Replication:** A replication master could send damaged events to slaves after the binary log disk on the master became full. To correct this issue, only complete events are now pushed by the master dump thread to the slave I/O thread. In addition, the error text that the master sends to the slave when an incomplete event is found now states that the incomplete event may have been caused by running out of disk space on the master, and provides coordinates of the first and the last event bytes read. (Bug #11747416, Bug #32228)

  References: See also Bug #64035, Bug #13593869.

- **Replication:** `--replicate-rewrite-db=`*`from_name`*`->`*`to_name`* did not work correctly when the name of the source database (*`from_name`*) consisted of only a single character. (Bug #34332, Bug #11747866)

- With `InnoDB` change buffering enabled and `innodb_page_size` set to an 8K or 4K page size, an `UPDATE` statement could fail if a column being updated contained a value longer than 1/8th of the page size. (Bug #13336585)

- An incorrect `InnoDB` assertion could cause the server to halt. This issue only affected debug builds. The assertion referenced the source file `btr0pcur.ic` and the variable `cursor->pos_state`. (Bug #13358468)

- A derived table with more than 64 columns caused a server crash. (Bug #13354889)

- Writes to the slow log involved a call to `thd->current_utime()` even if no log entries ended up being written, unnecessarily reducing performance. (Bug #13326965)

- Access privileges were checked for each stored program instruction, even if the instruction used no tables, resulting in reduced performance. (Bug #13251277)

- The error message for `ER_EVENT_CANNOT_ALTER_IN_THE_PAST` was incorrect. (Bug #13247871)

- Rounding `DBL_MAX` returned `DBL_MAX`, not 'inf'. (Bug #13261955)

- For materialized temporary tables, a missing key length check could cause incorrect query results. (Bug #13261277)

- During the table-opening process, memory was allocated and later freed that was needed view loading, even for statements that did not use views. These unnecessary allocation and free operations are no longer done. (Bug #13116518)

- Subqueries with `OUTER JOIN` could return incorrect results if the subquery referred to a column from another `SELECT`. (Bug #13068506)

- `mysql_plugin` returned the wrong error code from failed server bootstrap execution. (Bug #12968567)

- mysql_plugin mishandled the --plugin-ini, --mysqld, and --my-print-defaults options under some circumstances. (Bug #12968815)
- The Performance Schema nested some network I/O events within the wrong statement. (Bug #12981100)
- Internal conversion of zero to binary and back could yield a result with incorrect precision. (Bug #12911710)
- With index condition pushdown enabled, STRAIGHT_JOIN queries could produce incorrect results. (Bug #12822678, Bug #12724899)
- Valgrind warnings generated by filesort operations were fixed. (Bug #12856915)
- An IN-to-EXISTS subquery transformation could yield incorrect results if the outer value list contained NULL. (Bug #12838171)
- The result of ROUND() was incorrect for certain numbers. (Bug #12744991)
- A warning resulting from use of SPACE() referred to REPEAT() in the warning message. (Bug #12735829)
- IN and EXISTS subqueries with DISTINCT and ORDER BY could return incorrect results. (Bug #12709738)
- A memory leak occurred due to failure to clean up after QUICK_INDEX_MERGE_SELECT/Unique. (Bug #12694872, Bug #14542543)
- Several improvements were made to the libedit library bundled with MySQL distributions, and that is available for all platforms that MySQL supports except Windows.

  - Navigation keys did not work for UTF-8 input.
  - Word navigation and delete operations did not work for UTF-8 input with Cyrillic characters.
  - Nonlatin characters were corrupted in overwrite mode for UTF-8 input.
  - Long queries caused the statement history file to become corrupted.
  - The Alt key caused history operations to fail.

  (Bug #12605400, Bug #12613725, Bug #12618092, Bug #12624155, Bug #12617651, Bug #12605388)
- SELECT SQL_BUFFER_RESULT query results included too many rows if a GROUP BY clause was optimized away. (Bug #12578908)
- decimal_round() could cause a server exit when processing long numeric strings. (Bug #12563865)
- mysqldump --all-databases did not dump the replication log tables. (They could be dumped only by naming them explicitly when invoking mysqldump, and using the --master-data option.)

  As a result of the fix for this problem, it is now possible to execute statements requiring read locks on the replication log tables at any time, while any statements requiring a write lock on either or both of these tables are disallowed whenever replication is in progress. For more information, see Replication Relay and Status Logs. (Bug #12402875, Bug #60902)
- mysqld_safe did not properly check for an already running instance of mysqld. (Bug #11878394)
- The client-server protocol now has the client send authentication data as length-encoded strings so that data longer than 256 bytes can be sent. This is done using the CLIENT_PLUGIN_AUTH_LENENC_CLIENT_DATA client capability. (Bug #11878962)

- For a lower_case_table_names value of 1 or 2 and a database having a mixed-case name, calling a stored function using a fully qualified name including the database name failed. (Bug #60347, Bug #11840395)

- The help message for mysql_install_db did not indicate that it supports the --defaults-file, --defaults-extra-file and --no-defaults options. (Bug #58898, Bug #11765888)

- myisampack could create corrupt FULLTEXT indexes when compressing tables. (Bug #53646, Bug #11761180)

- For debug builds, an assertion could be raised for ALTER statements that performed a RENAME operation. This occurred for storage engine handlertons that exposed the HTON_FLUSH_AFTER_RENAME flag. (Bug #38028, Bug #11749050)

- An assertion designed to detect zero-length sort keys also was raised when the entire key set fit in memory. (Bug #58200, Bug #11765254)

- An ALTER TABLE that included an ADD ... AFTER operation to add a new column after a column that had been modified earlier in the statement failed to find the existing column. (Bug #34972, Bug #11748057)

- A linking problem prevented the FEDERATED storage engine plugin from loading. (Bug #40942, Bug #11750417)

- When a join operation contained a view, the optimizer sometimes failed to associate the view's WHERE clause with the first table or view in a join when it was possible to do so, resulting in a less efficient query. (Bug #59696, Bug #11766559)

- A multiple-table UPDATE statement required the UPDATE privilege on a view which was only read if the view was processed using the merge algorithm. (Bug #59957, Bug #11766767)

- If index condition pushdown access was chosen and then abandoned, some variables were not cleared, leading to incorrect query results. (Bug #62533)

- For FEDERATED tables, loss of connection to the remote table during some insert operations could cause a server crash. (Bug #34660, Bug #11747970)

- mysql_install_db printed the --skip-grant-tables server option as --skip-grant in one of its error messages. (Bug #58534, Bug #11765553)

- A query that selected a GROUP_CONCAT() function result could return different values depending on whether an ORDER BY of the function result was present. (Bug #41090, Bug #11750518)

- Subqueries could return incorrect results when materialization was enabled. (Bug #40037, Bug #11749901, Bug #12705660, Bug #12908058)

- InnoDB used incorrect identifier quoting style in an error message that resulted in an error if a user followed the suggestion in the message. (Bug #49556, Bug #11757503)

- OPTIMIZE TABLE could corrupt MyISAM tables if myisam_use_mmap was enabled. (Bug #49030, Bug #11757032)

- ARCHIVE tables with NULL columns could cause server crashes or become corrupt under concurrent load. (Bug #51252, Bug #11758979)

- During optimization, ZEROFILL values may be converted to string constants. However, CASE expressions did not handle switching data types after the planning stage, leading to CASE finding a null pointer instead of its argument. (Bug #57135, Bug #11764313)

- Deadlock could occur when these four things happened at the same time: 1) An old dump thread was waiting for the binary log to grow. 2) The slave server that replicates from the old dump thread tried to reconnect. During reconnection, the new dump thread tried to kill the old dump thread. 3)

A `KILL` statement tried to kill the old dump thread. 4) An `INSERT` statement caused a binary log rotation. (Bug #56299, Bug #11763573)

- The estimate of space required for `filesort` operations could be too high, resulting in inefficient initialization. (Bug #37359, Bug #11748783)
- If a plugin was uninstalled, thread local variables for plugin variables of string type with wth `PLUGIN_VAR_MEMALLOC` flag were not freed. (Bug #56652, Bug #11763882)
- An assertion was raised when selecting from a view that selects from a view that used a user-defined function that had been deleted. (Bug #59546, Bug #11766440)
- `mysql_upgrade` did not upgrade the system tables or create the `mysql_upgrade_info` file when run with the `--write-binlog` or `--skip-write-binlog` option. (Bug #60223, Bug #11827359)
- Concurrent access to `ARCHIVE` tables could cause corruption. (Bug #42784, Bug #11751793)
- The `SQL_BIG_RESULT` modifier could change the results for queries that included a `GROUP BY` clause. (Bug #53534, Bug #11761078)
- The `CMake` configuration checks did not properly test whether the C compiler supports the `inline` keyword. (Bug #61708, Bug #12711108)
- For some queries, the `index_merge` access method was used even when more expensive then `ref` access. (Bug #32254, Bug #11747423)
- Collation for the `SPACE()` function was determined by the parse time value of the `collation_connection` system variable (instead of the runtime value), which could give unexpected results from prepared statements, triggers, and stored procedures. (Bug #23637, Bug #11746123)

Changes in MySQL 5.6.3 (2011-10-03, Milestone 6)

> **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.

Parallel Event Execution (multi-threaded slave)

- **Replication:** MySQL replication now supports a multi-threaded slave executing replication events from the master across different databases in parallel, which can result in significant improvements in application throughput when certain conditions are met. The optimum case is that the data be partitioned per database, and that updates within a given database occur in the same order relative to one another as they do on the master. However, transactions do not need to be coordinated between different databases.

  The `slave_parallel_workers` server system variable (added in this release) sets the number of slave worker threads for executing replication events in parallel. When parallel execution is enabled, the slave SQL thread acts as the coordinator for the slave worker threads, among which transactions are distributed on a per-database basis. This means that a worker thread on the slave can process successive transactions on a given database without waiting for updates on other databases to complete.

  Due to the fact that transactions on different databases can occur in a different order on the slave than on the master, checking for the most recently executed transaction does not guarantee that all previous transactions from the master have been executed on the slave. This has implications for logging and recovery when using a multi-threaded slave. For information about how to interpret binary logging information when using multi-threading on the slave, see SHOW SLAVE STATUS Syntax.

Optimizer Notes

- These query optimizer improvements were implemented:
  - The EXPLAIN statement now provides execution plan information for DELETE, INSERT, REPLACE, and UPDATE statements. Previously, EXPLAIN provided information only for SELECT statements.
  - The optimizer more efficiently handles subqueries in the FROM clause (that is, derived tables):
    - Materialization of subqueries in the FROM clause is postponed until their contents are needed during query execution, which improves performance. Previously, subqueries in the FROM clause were materialized for EXPLAIN SELECT statements. This resulted in partial SELECT execution, even though the purpose of EXPLAIN is to obtain query plan information, not to execute the query. This materialization no longer occurs, so EXPLAIN is faster for such queries. For non-EXPLAIN queries, delay of materialization may result in not having to do it at all. Consider a query that joins the result of a subquery in the FROM clause to another table: If the optimizer processes that other table first and finds that it returns no rows, the join need not be carried out further and the optimizer can completely skip materializing the subquery.
    - During query execution, the optimizer may add an index to a derived table to speed up row retrieval from it.

    For more information, see Optimizing Subqueries in the FROM Clause (Derived Tables).
  - A Batched Key Access (BKA) join algorithm is now available that uses both index access to the joined table and a join buffer. The BKA algorithm supports inner join and outer join operations, including nested outer joins. Benefits of BKA include improved join performance due to more efficient table scanning.

    Two flags have been added to the optimizer_switch system variable (block_nested_loop and batched_key_access). These flags control how the optimizer uses the Block Nested-Loop and Batched Key Access join algorithms. Previously, the optimizer_join_cache_level system variable was used for join buffer control; this variable has been removed.

    For more information, see Block Nested-Loop and Batched Key Access Joins.
  - The optimizer now has a tracing capability. This will be of use to optimizer developers, and also to users who file bugs against the optimizer and want to provide more information that will help resolve the bug. The interface is provided by a set of optimizer_trace_*xxx* system variables and the INFORMATION_SCHEMA.OPTIMIZER_TRACE table, but is subject to change. For details, see MySQL Internals: Tracing the Optimizer.

  (Bug #44802, Bug #11753371, Bug #14295, Bug #11745379, Bug #27975, Bug #11746677)

Performance Schema Notes

- The Performance Schema has these additions:
  - The Performance Schema now instruments stages and statements. Stages are steps during the statement-execution process, such as parsing a statement, opening a table, or performing a filesort operation. Stages correspond to the thread states displayed by SHOW PROCESSLIST or that are visible in the INFORMATION_SCHEMA.PROCESSLIST table. Stages begin and end when state values change.

    Within the event hierarchy, wait events nest within stage events, which nest within statement events. To reflect this nesting in wait-event tables such as events_waits_current, the NESTING_EVENT_ID column now can be non-NULL to indicate the EVENT_ID value of the event within which an event is nested, and NESTING_EVENT_TYPE is a new column indicating the type of the nesting event.

The setup_instruments table now contains instruments with names that begin with stage and statement. Corresponding to these instruments, the setup_timers table now contains rows with NAME values of stage and statement that indicate the unit for stage and statement event timing. The default unit for each is NANOSECOND.

These new tables store stage and statement events:

- events_stages_current: Current stage events
- events_stages_history: The most recent stage events for each thread
- events_stages_history_long: The most recent stage events overall
- events_statements_current: Current statement events
- events_statements_history: The most recent statement events for each thread
- events_statements_history_long: The most recent statement events overall

The setup_consumers table now contains consumer values with names corresponding to those table names. These consumers may be used to filter collection of stage and statement events.

There are also summary tables that provide aggregated stage and statement information.

Application developers can use statement instrumentation to see in detail the statements generated by an application, and how these statements are executed by the server. Stage instrumentation can be used to focus on particular parts of statements. This information may be useful to change how an application issues queries against the database, to minimize the application footprint on the server, and to improve application performance and scalability.

For more information, see Performance Schema Stage Event Tables, Stage Summary Tables, Performance Schema Statement Event Tables, and Statement Summary Tables.

- The Performance Schema now provides statistics about connections to the server. When a client connects, it does so under a particular user name and from a particular host. The Performance Schema tracks connections per account (user name plus host name) and separately per user name and per host name, using these tables:

  - accounts: Connection statistics per client account
  - hosts: Connection statistics per client host name
  - users: Connection statistics per client user name

  There are also summary tables that provide aggregated connection information.

  It is good security practice to define a dedicated account per application, so that an application is given privileges to perform only those actions that it needs during its operation. This also facilitates monitoring because the information in the connection tables can be used by application developers to see load statistics per application when deploying several applications against a given database server.

  For more information, see Performance Schema Connection Tables, and Connection Summary Tables.

- Previously, the setup_objects table could be used only to specify by inclusion which objects to instrument. There was no way to explicitly disable object instrumentation, such as to configure instrumention for all tables except those in a particular database. Now the setup_objects table includes an ENABLED column that indicates whether to instrument matching objects. This feature improves the setup_objects table usability because it permits exclusion patterns.

  The default table contents now include a row that disables instrumentation for tables in the mysql database, which is a change from the previous default object instrumentation. This change is chosen assuming that end users want to instrument application objects, not internal server tables. The change reduces the default Performance Schema overhead because I/O and locks on mysql tables are not instrumented.

  The table also includes rows that disable instrumentation for tables in the INFORMATION_SCHEMA and performance_schema databases. This is not a change in behavior because those tables were not instrumented before, but these rows make the full object instrumentation defaults explicit.

- The Performance Schema now instruments sockets. This enables monitoring of network communication to and from the server. Information collected includes network activity such as socket instances, socket operations, and number of bytes transmitted and received.

  The setup_instruments table now contains instruments with names that begin with wait/io/socket. There is also an idle instrument used for idle events when a socket is waiting for the next request from the client. Corresponding to the latter instrument, the setup_timers table now contains a row with a NAME value of idle that indicates the unit for idle event timing. The default unit is MICROSECOND.

  These new tables contain socket information:

  - socket_instances: A real-time snapshot of the active connections to the MySQL server

  - socket_summary_by_instance: Aggregate timer and byte count statistics generated by the wait/io/socket/* instruments for all socket I/O operations, per socket instance

  - socket_summary_by_event_name: Aggregate timer and byte count statistics generated by the wait/io/socket/* instruments for all socket I/O operations, per socket instrument

  The information in the socket tables can be used by application developers, particularly those developing web-based applications, to assess the volume of network traffic directly attributable to queries generated by their application. This can be particularly useful during development of applications intended for large-scale implementations.

  For more information, see The socket_instances Table, and Socket Summary Tables.

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

Security Notes

- Statement logging has been modified so that passwords do not appear in plain text for the following statements:

  ```
  CREATE USER ... IDENTIFIED BY ...
  GRANT ... IDENTIFIED BY ...
  SET PASSWORD ...
  ```

  Passwords in those statements are rewritten not to appear literally in statement text, for the general query log, slow query log, and binary log. Rewriting does not apply to other statements.

  For the general query log, password rewriting can be suppressed by starting the server with the --log-raw option. This option may be useful for diagnostic purposes, to see the exact text of statements as received by the server, but for security reasons is not recommended for production use.

One change you will notice is that statements that cannot be parsed (due, for example, to syntax errors) are no longer written to the general query log because they cannot be known to be password free. Use cases that require logging of all statements including those with errors should use the `--log-raw` option, bearing in mind that this also bypasses password writing.

Functionality Added or Changed

- **Performance; InnoDB:** At shutdown, MySQL can record the pages that are cached in the `InnoDB` buffer pool, then reload those same pages upon restart. This technique can help to quickly reach consistent throughput after a restart, without a lengthy warmup period. This preload capability uses a compact save format and background I/O to minimize overhead on the MySQL server. The basic dump/restore capability is enabled through the configuration options `innodb_buffer_pool_dump_at_shutdown` and `innodb_buffer_pool_load_at_startup`. Related configuration options such as `innodb_buffer_pool_dump_now` and `innodb_buffer_pool_load_now` offer extra flexibility for advanced users to configure the MySQL server for different workloads. See Preloading the `InnoDB` Buffer Pool for Faster Restart for details. (Bug #11765816, Bug #58819)

- **Performance; InnoDB:** When `innodb_file_per_table` is enabled, each `InnoDB` table is created in its own tablespace file (`.ibd` file). As data inside the table grows, the `.ibd` file is extended, which is an I/O operation that may create a bottleneck for busy systems with many `InnoDB` tables. For `InnoDB` tables that are stored inside the system tablespace, the extension operation happens less frequently, as space freed by `DELETE` or `TRUNCATE` operations within one table can be reused by another table.

  MySQL 5.6 improves concurrency for extending `InnoDB` tablespace files (`.ibd` files), so that multiple `.ibd` files can be extended simultaneously without blocking read or write operations performed by other threads. (Bug #11763692, Bug #56433)

- **Performance; InnoDB:** You can improve the efficiency of the `InnoDB` checksum feature by specifying the configuration option `innodb_checksum_algorithm=crc32`, which turns on a faster checksum algorithm. This option replaces the `innodb_checksums` option. Data written using the old checksum algorithm (option value `innodb`) is fully upward-compatible; tablespaces modified using the new checksum algorithm (option value `crc32`) cannot be downgraded to an earlier version of MySQL that does not support the `innodb_checksum_algorithm` option. See Using the CRC32 Checksum Algorithm for Faster Checksums for details. (Bug #11757757, Bug #49852)

- **Performance; InnoDB:** This feature optionally moves the `InnoDB` undo log out of the system tablespace into one or more separate tablespaces. The I/O patterns for the undo log make these new tablespaces good candidates to move to SSD storage, while keeping the system tablespace on hard disk storage. This feature is controlled by the configuration options `innodb_undo_directory`, `innodb_undo_tablespaces`, and `innodb_undo_logs` (formerly known as `innodb_rollback_segments`). Users cannot drop the separate tablespaces created to hold `InnoDB` undo logs, or the individual segments inside those tablespaces.

  MySQL instances configured this way are not downward-compatible; older versions of MySQL cannot access the undo logs that reside in their own tablespace.

  For details, see Storing `InnoDB` Undo Logs in Separate Tablespaces.

- **Performance; InnoDB:** Work continues to offload flush operations from the `InnoDB` main thread, doing them in the `page_cleaner` thread instead. The latest changes to the **buffer pool** flushing algorithms can improve performance for some I/O-bound workloads, particularly in configurations with multiple buffer pool instances. You control this feature by adjusting the settings for the `innodb_lru_scan_depth` and `innodb_flush_neighbors` configuration options. To find the optimal settings, test each combination of the above settings with both the Adaptive Hash Index and the Doublewrite Buffer turned on and off. See Tuning `InnoDB` Buffer Pool Flushing for more details.

- **Performance; InnoDB:** The `InnoDB` thread-scheduling code has been enhanced to work better with greater than 16 threads. Where possible, atomic instructions are used. You control this feature by setting the configuration option innodb_thread_concurrency to a nonzero value, and adjusting the value of innodb_adaptive_max_sleep_delay. See Configuring Thread Concurrency for InnoDB for details.

- **Performance; InnoDB:** The code that detects deadlocks in InnoDB transactions has been modified to use a fixed-size work area rather than a recursive algorithm. The resulting detection operation is faster as a result. You do not need to do anything to take advantage of this enhancement.

  Under both the old and new detection mechanisms, you might encounter a search too deep error that is not a true deadlock, but requires you to re-try the transaction the same way as with a deadlock.

- **Incompatible Change:** In the audit plugin interface, the event_class member was removed from the mysql_event_general structure and the calling sequence for the notification function was changed. Originally, the second argument was a pointer to the event structure. The function now receives this information as two arguments: an event class number and a pointer to the event. Corresponding to these changes, MYSQL_AUDIT_INTERFACE_VERSION was increased to 0x0300.

  The plugin_audit.h header file, and the NULL_AUDIT example plugin in the plugin/audit_null directory were modified per these changes. See Writing Audit Plugins.

- **Important Change; Replication:** The RESET SLAVE statement has been extended with an ALL keyword. In addition to deleting the master.info, relay-log.info, and all relay log files, RESET SLAVE ALL also clears all connection information otherwise held in memory following execution of RESET SLAVE. (Bug #11809016, Bug #11763210)

- **InnoDB:** InnoDB now permits concurrent reads while creating a secondary index. (Bug #11853126)

  References: See also Bug #11751388, Bug #11784056, Bug #11815600.

- **InnoDB:** The InnoDB redo log files now have a maximum combined size of 512GB, increased from 4GB. You can specify the larger values through the innodb_log_file_size option.

  There is no special upgrade process or file format to take advantage of this enhancement. The bytes that record the extra size information were already reserved in the InnoDB system tablespace. However, if you develop applications that interact with the InnoDB logical sequence number (LSN) value, change your code to use guaranteed 64-bit variables to store and compare LSN values, rather than 32-bit variables. (Bug #11765780, Bug #58779)

- **InnoDB:** InnoDB tables can now be created with character sets whose collation ID is greater than 255. For example, the following InnoDB table can now be created, where formerly the collation ID of 359 was beyond the range supported by InnoDB.

  ```
  sql> show collation like 'ucs2_vn_ci';
  +------------+---------+-----+---------+----------+---------+
  | Collation  | Charset | Id  | Default | Compiled | Sortlen |
  +------------+---------+-----+---------+----------+---------+
  | ucs2_vn_ci | ucs2    | 359 |         |          |       8 |
  +------------+---------+-----+---------+----------+---------+
  1 row in set (0.00 sec)

  mysql> create table two_byte_collation (c1 char(1) character set ucs2 collate ucs2_vn_ci)
    -> engine = InnoDB;
  Query OK, 0 rows affected (0.16 sec)
  ```

  This capability opens up InnoDB tables for use with a range of user-defined character sets. MySQL's predefined character sets have previously been limited to a maximum of 255, and now that restriction is lifted. See Choosing a Collation ID for more information.

- **Replication:** MySQL 5.6.1 added timestamps to the error messages shown in the Last_IO_Error and Last_SQL_Error columns of the output of SHOW SLAVE STATUS. Now these timestamps are shown in separate columns of their own, named Last_IO_Error_Timestamp and Last_SQL_Error_Timestamp, respectively. (Bug #11765599, Bug #58584)

References: See also Bug #43535, Bug #11752361.

- **Replication:** BEGIN, COMMIT, and ROLLBACK statements are now cached along with the statements instead of being written when the cache is flushed to the binary log. This change does not affect DDL statements—which are written into the statement cache, then immediately flushed—or Incident events (which, along with Rotate events, are still written directly to the binary log).

  References: See also Bug #57275, Bug #11764443.

- Following EXPLAIN EXTENDED, a change has been made to the transformed query displayed by SHOW WARNINGS. Each SELECT part now is preceded by the id value from the associated EXPLAIN output row. This makes it easier to see the correspondence between those rows and parts of the transformed query. For example, this query:

  ```
  EXPLAIN EXTENDED SELECT 36 FROM DUAL
  ```

  results in:

  ```
  /* select#1 */ select 36 from dual
  ```

  And this query:

  ```
  EXPLAIN EXTENDED SELECT a FROM t
  WHERE a IN (SELECT b FROM u UNION SELECT c from v)
  ```

  results in:

  ```
  /* select#1 */ select a from t where a in (/* select#2 */
  select b from u union /* select#3 */ select c from v);
  ```

  (Bug #13035597)

- Several memory allocation calls were eliminated, resulting in improved performance. (Bug #12552221)

- CMake configuration support on Linux now provides a boolean ENABLE_GCOV option to control whether to include support for gcov. (Bug #12549572)

- Previously, Performance Schema instrumentation for both the binary log and the relay log used these instruments:

  ```
  wait/io/file/sql/binlog
  wait/io/file/sql/binlog_index
  wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_index
  wait/synch/cond/sql/MYSQL_BIN_LOG::update_cond
  ```

  Now instrumentation for the relay log uses these instruments, which makes it possible to distinguish binary log and relay log events:

  ```
  wait/io/file/sql/relaylog
  wait/io/file/sql/relaylog_index
  wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_index
  wait/synch/cond/sql/MYSQL_RELAY_LOG::update_cond
  ```

  (Bug #59658, Bug #11766528)

- The server now exposes SSL certificate expiration dates through the Ssl_server_not_before and Ssl_server_not_after status variables. Both variables have values in ANSI time format (for example, Sep 12 16:22:06 2013 GMT), or are blank for non-SSL connections. (Bug #57648, Bug #11764778)

- When invoked with the --auto-generate-sql option, mysqlslap dropped the schema specified with the --create-schema option at the end of the test run, which may have been unexpected by the user. mysqlslap now has a --no-drop option that prevents any schema created during the test run from being dropped. (Bug #58090, Bug #11765157)

- Previously, TEMPORARY tables created with CREATE TEMPORARY TABLES had the default storage engine unless the definition included an explicit ENGINE option. (The default engine is the value of the default_storage_engine system variable.) Since MySQL 5.5.5, when the default storage engine was changed from the nontransactional MyISAM engine to the transactional InnoDB engine, TEMPORARY tables have incurred the overhead of transactional processing.

  To permit the default storage engine for TEMPORARY tables to be set independently of the default engine for permanent tables, the server now supports a default_tmp_storage_engine system variable. For example, to create TEMPORARY tables as nontransactional tables by default, start the server with --default_tmp_storage_engine=MyISAM. The storage engine for TEMPORARY tables can still be specified on an individual basis by including an ENGINE option in table definitions. (Bug #49232, Bug #11757216)

- A new server option, --plugin-load-add, complements the --plugin-load option. --plugin-load-add adds a plugin or plugins to the set of plugins to be loaded at startup. The argument format is the same as for --plugin-load. --plugin-load-add can be used to avoid specifying a large set of plugins as a single long unwieldy --plugin-load argument.

  --plugin-load-add can be given in the absence of --plugin-load, but any instance of --plugin-load-add that appears before --plugin-load. has no effect because --plugin-load resets the set of plugins to load.

  This change affects the output of mysqld --verbose --help in that a value for plugin-load is no longer printed. (Bug #59026, Bug #11766001)

- The mysql client program now has a --binary-mode option that helps when processing mysqlbinlog output that may contain BLOB values. By default, mysql translates \r\n in statement strings to \n and interprets \0 as the statement terminator. --binary-mode disables both features. It also disables all mysql commands except charset and delimiter in non-interactive mode (for input piped to mysql or loaded using the source command). (Bug #33048, Bug #11747577)

- Previously, for MySQL binaries linked against OpenSSL, if an SSL key file supplied to the MySQL server or a MySQL client program (using the --ssl-key option) was protected by a passphrase, the program would prompt the user for the passphrase. This is now also the case for MySQL binaries linked against yaSSL. (Bug #44559, Bug #11753167)

- The NULL_AUDIT example plugin in the plugin/audit_null directory has been updated to count instances of events in the MYSQL_AUDIT_CONNECTION_CLASS event class. See Writing Audit Plugins.

- The max_allowed_packet system variable now controls the maximum size of parameter values that can be sent with the mysql_stmt_send_long_data() C API function.

- For temporary tables created with the CREATE TEMPORARY TABLE statement, the privilege model has changed.

  Previously, the CREATE TEMPORARY TABLES privilege enabled users to create temporary tables with the CREATE TEMPORARY TABLE statement. However, other operations on a temporary table, such as INSERT, UPDATE, or SELECT, required additional privileges for those operations for the database containing the temporary table, or for the nontemporary table of the same name.

To keep privileges for temporary and nontemporary tables separate, a common workaround for this situation was to create a database dedicated to the use of temporary tables. Then for that database, a user could be granted the CREATE TEMPORARY TABLES privilege, along with any other privileges required for temporary table operations done by that user.

Now, the CREATE TEMPORARY TABLES privilege enables users to create temporary tables with CREATE TEMPORARY TABLE, as before. However, after a session has created a temporary table, the server performs no further privilege checks on the table. The creating session can perform any operation on the table, such as DROP TABLE, INSERT, UPDATE, or SELECT.

One implication of this change is that a session can manipulate its temporary tables even if the current user has no privilege to create them. Suppose that the current user does not have the CREATE TEMPORARY TABLES privilege but is able to execute a DEFINER-context stored procedure that executes with the privileges of a user who does have CREATE TEMPORARY TABLES and that creates a temporary table. While the procedure executes, the session uses the privileges of the defining user. After the procedure returns, the effective privileges revert to those of the current user, which can still see the temporary table and perform any operation on it. (Bug #27480, Bug #11746602)

- The Windows installer now creates an item in the MySQL menu named MySQL command line client - Unicode. This item invokes the mysql client with properties set to communicate through the console to the MySQL server using Unicode. It passes the --default-character-set=utf8 option to mysql and sets the font to the Lucida Console Unicode-compatible font. See Unicode Support on Windows.

- MySQL now includes support for manipulating IPv6 network addresses and for validating IPv4 and IPv6 addresses:

  - The INET6_ATON() and INET6_NTOA() functions convert between string and numeric forms of IPv6 addresses. Because numeric-format IPv6 addresses require more bytes than the largest integer type, the representation uses the VARBINARY data type.

  - The IS_IPV4() and IS_IPV6() functions test whether a string value represents a valid IPv4 or IPv6 address. The IS_IPV4_COMPAT() and IS_IPV4_MAPPED() functions test whether a numeric-format value represents a valid IPv4-compatible or IPv4-mapped address.

  - No changes were made to the INET_ATON() or INET_NTOA() functions that manipulate IPv4 addresses.

  IS_IPV4() is more strict than INET_ATON() about what constitutes a valid IPv4 address, so it may be useful for applications that need to perform strong checks against invalid values. Alternatively, use INET6_ATON() to convert IPv4 addresses to internal form and check for a NULL result (which indicates an invalid address). INET6_ATON() is equally strong as IS_IPV4() about checking IPv4 addresses.

- Client programs now display more information for SSL errors to aid in diagnosis and debugging of connection problems. (Bug #21287, Bug #11745920)

- The undocumented --all option for perror has been removed. Also, perror no longer displays messages for BDB error codes.

- 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 innodb_table_monitor table. Similar information can be obtained from InnoDB INFORMATION_SCHEMA tables. See INFORMATION_SCHEMA Tables for InnoDB.

  - The innodb_locks_unsafe_for_binlog system variable.

  - The innodb_stats_sample_pages system variable. Use innodb_stats_transient_sample_pages instead.
  - The innodb_use_sys_malloc and the innodb_additional_mem_pool_size system variables.

- A new utility, mysql_plugin, enables MySQL administrators to manage which plugins a MySQL server loads. It provides an alternative to manually specifying the --plugin-load option at server startup or using the INSTALL PLUGIN and UNINSTALL PLUGIN statements at runtime. See mysql_plugin — Configure MySQL Server Plugins.

- mysqld now has an --ignore-db-dir option that tells the server to ignore a given name for purposes of the SHOW DATABASES statement or INFORMATION_SCHEMA tables. For example, if a MySQL configuration locates the data directory at the root of a file system on Unix, the system might create a lost+found directory there that the server should ignore. Starting the server with --ignore-db-dir=lost+found causes that name not to be listed as a database.

  To specify more than one name, use this option multiple times, once for each name. Specifying the option with an empty value (that is, as --ignore-db-dir=) resets the directory list to the empty list.

  Instances of this option given at server startup are used to set the ignore_db_dirs system variable.

  In addition to directories named by --ignore-db-dir, directories having a name that begins with a period are ignored. (Bug #22615, Bug #11746029)

- MySQL binaries linked against OpenSSL (but not yaSSL) now support certificate revocation lists for SSL connections:

  - The MySQL server and MySQL client programs that support SSL recognize --ssl-crl and --ssl-crlpath options for specifying a revocation list file or directory containing such files.
  - The ssl_crl and ssl_crlpath system variables indicate the values of the --ssl-crl and --ssl-crlpath options with which the server was started.
  - The CHANGE MASTER TO statement has MASTER_SSL_CRL and MASTER_SSL_CRLPATH options for specifying revocation list information to use when the slave connects to the master. The mysql.slave_master_info file has two more rows to store the values of these options. The SHOW SLAVE STATUS statement has two more columns to display the values of these options.

    The mysql_options() C API function has MYSQL_OPT_SSL_CRL and MYSQL_OPT_SSL_CRLPATH options for specifying revocation list information to use when the client connects to the master. In addition, mysql_options() now also supports MYSQL_OPT_SSL_CA, MYSQL_OPT_SSL_CAPATH, MYSQL_OPT_SSL_CERT, MYSQL_OPT_SSL_CIPHER, and MYSQL_OPT_SSL_KEY options for specifying other SSL parameters.

  (Bug #31224, Bug #11747191)

- Some plugins operate in such a matter that they should be loaded at server startup, and not loaded or unloaded at runtime. The plugin API now supports marking plugins this way. The st_mysql_plugin structure now has a flags member, which can be set to the OR of the applicable flags. The PLUGIN_OPT_NO_INSTALL flag indicates that the plugin cannot be loaded at runtime with the INSTALL PLUGIN statement. This is appropriate for plugins that must be loaded at server startup with the --plugin-load option. The PLUGIN_OPT_NO_UNINSTALL flag indicates that the plugin cannot be unloaded at runtime with the UNINSTALL PLUGIN statement.

The new member changes the interface, so the plugin interface version, MYSQL_PLUGIN_INTERFACE_VERSION, has been incremented from 0x0102 to 0x0103. Plugins that require access to the new member must be recompiled to use version 0x0103 or higher.

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #59533)
- **Performance; InnoDB:** This fix improves the performance of operations on VARCHAR(*N*) columns in InnoDB tables, where *N* is declared as a large value but the actual string values in the table are short. (Bug #12835650)
- **Performance; InnoDB:** The mechanism that InnoDB uses to detect if the MySQL server is idle was made more accurate, to avoid slowdown due to flush operations that normally occur when no other activity is taking place. The mechanism now considers that the server is not idle if there are pending read requests for the InnoDB buffer pool. (Bug #11766123, Bug #59163)
- **Incompatible Change:** For socket I/O, an optimization for the case when the server used alarms for timeouts could cause a slowdown when socket timeouts were used instead.

  The fix for this issue results in several changes:

  - Previously, timeouts applied to entire packet-level send or receive operations. Now timeouts apply to individual I/O operations at a finer level, such as sending 10 bytes of a given packet.
  - The handling of packets larger than max_allowed_packet has changed. Previously, if an application sent a packet bigger than the maximum permitted size, or if the server failed to allocate a buffer sufficiently large to hold the packet, the server kept reading the packet until its end, then skipped it and returned an ER_NET_PACKET_TOO_LARGE error. Now the server disconnects the session if it cannot handle such large packets.
  - On Windows, the default value for the MYSQL_OPT_CONNECT_TIMEOUT option to mysql_options() is no longer 20 seconds. Now the default is no timeout (infinite), the same as on other platforms.
  - Building and running MySQL on POSIX systems now requires support for poll() and O_NONBLOCK. These should be available on any modern POSIX system.

  (Bug #54790, Bug #36225, Bug #11762221, Bug #51244, Bug #11758972)
- **Incompatible Change:** The mysql_affected_rows() C API function returned 3 (instead of 2) for INSERT ... ON DUPLICATE KEY UPDATE statements where there was a duplicated key value.

  Now the affected-rows value per row is 1 if the row is inserted as a new row, 2 if an existing row is updated, and 0 if an existing row is set to its current values. If you specify the CLIENT_FOUND_ROWS flag to mysql_real_connect() when connecting to mysqld, the affected-rows value is 1 (not 0) if an existing row is set to its current values. (Bug #46675, Bug #11754979)
- **Incompatible Change:** Handling of a date-related assertion was modified. A consequence of this change is that several functions become more strict when passed a DATE() function value as their argument and reject incomplete dates with a day part of zero. These functions are affected: CONVERT_TZ(), DATE_ADD(), DATE_SUB(), DAYOFYEAR(), LAST_DAY(), TIMESTAMPDIFF(), TO_DAYS(), TO_SECONDS(), WEEK(), WEEKDAY(), WEEKOFYEAR(), YEARWEEK().

  It was later determined that stricter handling is unnecessary for LAST_DAY(), which was reverted to permit a zero day part in MySQL 5.6.5.

  References: See also Bug #13458237.
- **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:** The `DATA_LENGTH` column in the `INFORMATION_SCHEMA.TABLES` table now correctly reports the on-disk sizes of tablespaces for `InnoDB` compressed tables. (Bug #12770537)

- **InnoDB:** A failed `CREATE INDEX` operation for an `InnoDB` table could result in some memory being allocated but not freed. This memory leak could affect tables created with the `ROW_FORMAT=DYNAMIC` or `ROW_FORMAT=COMPRESSED` setting. (Bug #12699505)

- **InnoDB:** With the configuration settings `innodb_file_per_table=1` and `innodb_file_format=Barracuda`, inserting a column value greater than half the page size, and including that column in a secondary index, could cause a crash when that column value was updated. (Bug #12637786)

- **InnoDB:** The underlying tables to support the `InnoDB` persistent statistics feature were renamed and moved into the `mysql` database. `innodb.table_stats` became `mysql.innodb_table_stats`, and `innodb.index_stats` became `mysql.innodb_index_stats`. (Bug #12604399)

- **InnoDB:** The server could halt if `InnoDB` interpreted a very heavy I/O load for 15 minutes or more as an indication that the server was hung. This change fixes the logic that measures how long `InnoDB` threads were waiting, which formerly could produce false positives. (Bug #11877216, Bug #11755413, Bug #47183)

- **InnoDB:** With the setting `lower_case_table_names=2`, inserts into `InnoDB` tables covered by foreign key constraints could fail after a server restart. (Bug #11831040, Bug #60196, Bug #60909)

- **InnoDB:** If the MySQL Server crashed immediately after creating an `InnoDB` table, attempting to access the table after restart could cause another crash. The issue could occur if the server halted after `InnoDB` created the primary index for the table, but before the index definition was recorded in the MySQL metadata. (Bug #11766824, Bug #60042)

- **InnoDB:** If the server crashed while an XA transaction was prepared but not yet committed, the transaction could remain in the system after restart, and cause a subsequent shutdown to hang. (Bug #11766513, Bug #59641)

- **InnoDB:** The MySQL server could hang during `CREATE TABLE`, `OPTIMIZE TABLE`, or `ALTER TABLE` or other DDL operation that performs a table copy for an `InnoDB` table, if such operations were performed by multiple sessions simultaneously. The error was reported as:

  ```
  InnoDB: Error: semaphore wait has lasted > 600 seconds
  ```

  (Bug #11760042, Bug #52409)

- **InnoDB:** With the setting `lower_case_table_names=2`, inserts into `InnoDB` tables covered by foreign key constraints could fail after a server restart. This is a similar problem to the foreign key error in Bug #11831040 / Bug #60196 / Bug #60909, but with a different root cause and occurring on Mac OS X.

- **Partitioning:** The internal `get_partition_set()` function did not take into account the possibility that a key specification could be `NULL` in some cases. (Bug #12380149)

- **Partitioning:** Auto-increment columns of partitioned tables were checked even when they were not being written to. In debug builds, this could lead to a server crash. (Bug #11765667, Bug #58655)

- **Partitioning:** When executing a row-ordered retrieval index merge, the partitioning handler used memory from that allocated for the table, rather than that allocated to the query, causing table object memory not to be freed until the table was closed. (Bug #11766249, Bug #59316)

- **Partitioning:** Attempting to use `ALTER TABLE ... EXCHANGE PARTITION` to exchange a view with a (nonexistent) partition of a table that was not partitioned caused the server to crash. (Bug #11766232, Bug #60039)

- **Partitioning:** The `UNIX_TIMESTAMP()` function was not treated as a monotonic function for purposes of partition pruning. (Bug #11746819, Bug #28928)

- **Partitioning:** A problem with a previous fix for poor performance of `INSERT ON DUPLICATE KEY UPDATE` statements on tables having many partitions caused the handler function for reading a row from a specific index to fail to store the ID of the partition last used. This caused some statements to fail with `Can't find record` errors. (Bug #59297, Bug #11766232)

  References: This bug is a regression of Bug #52455.

- **Replication:** When using row-based replication and attribute promotion or demotion (see Replication of Columns Having Different Data Types), memory allocated internally for conversion of `BLOB` columns was not freed afterwards. (Bug #12558519)

- **Replication:** A memory leak could occur when re-creating a missing master info repository, because a new I/O cache used for a reference to the repository was re-created when the repository was re-created, but the previous cache was never removed. (Bug #12557307)

- **Replication:** A mistake in thread cleanup could cause a replication master to crash. (Bug #12578441)

- **Replication:** A race condition could occur between a user thread and the SQL thread when both tried to read the same memory before its value was safely set. This issue has now been corrected.

  In addition, internal functions relating to creation of and appending to log events, when storing data, used memory local to the functions which was freed when the functions returned. As part of the fix for this problem, the output of `SHOW SLAVE STATUS` has been modified such that it no longer refers to files or file names in the accompanying status message, but rather contains one of the messages `Making temporary file (append) before replaying LOAD DATA INFILE` or `Making temporary file (create) before replaying LOAD DATA INFILE`. (Bug #12416611)

- **Replication:** The name of the `Ssl_verify_server_cert` column in the `mysql.slave_master_info` table was misspelled as `Ssl_verify_servert_cert`. (Bug #12407446, Bug #60988)

- **Replication:** When `mysqlbinlog` was invoked using `--base64-output=decode-row` and `--start-position=pos`, (where *pos* is a point in the binary log past the format description log event), a spurious error of the type shown here was generated:

  ```
  malformed binlog: it does not contain any Format_description_log_event...
  ```

  However, since there is nothing unsafe about not printing the format description log event, the error has been removed for this case. (Bug #12354268)

- **Replication:** It is no longer possible to change the storage engine used by the `mysql.slave_master_info` and `mysql.slave_relay_log_info` tables while replication is running. This means that, to make replication crash-safe, you must make sure that both of these tables use a transactional storage engine before starting replication.

  For more information, see Replication Relay and Status Logs, and Options for Logging Slave Status to Tables. (Bug #11765887, Bug #58897)

- **Replication:** A failed `CREATE USER` statement was mistakenly written to the binary log. (Bug #11827392, Bug #60082)

- **Replication:** When a slave requested a binary log file which did not exist on the master, the slave continued to request the file regardless. This caused the slave's error log to be flooded with low-level **EE_FILENOTFOUND** errors (error code 29) from the master. (Bug #11745939, Bug #21437)
- **Replication:** Retrying a transaction on the slave could insert extra data into nontransactional tables. (Bug #11763126, Bug #55789)

  References: See also Bug #11763471, Bug #56184.
- **Replication:** A transaction was written to the binary log even when it did not update any nontransactional tables. (Bug #11763471, Bug #56184)

  References: See also Bug #11763126, Bug #55789.
- **Replication:** Typographical errors appeared in the text of several replication error messages. (The word “position” was misspelled as “postion”.) (Bug #11762616, Bug #55229)
- **Replication:** mysqlbinlog using the --raw option did not function correctly with binary logs from MySQL Server versions 5.0.3 and earlier. (Bug #11763265, Bug #55956)
- **Replication:** Temporary deadlocks in the slave SQL thread could cause unnecessary Deadlock found when trying to get lock; try restarting transaction error messages to be logged on the slave.

  Now in such cases, only a warning is logged unless slave_transaction_retries has been exceeded by the number of such warnings for a given transaction. (Bug #11748510, Bug #36524)
- **Replication:** Processing of corrupted table map events could cause the server to crash. This was especially likely if the events mapped different tables to the same identifier, such as could happen due to Bug #56226.

  Now, before applying a table map event, the server checks whether the table has already been mapped with different settings, and if so, an error is raised and the slave SQL thread stops. If it has been mapped with the same settings, or if the table is set to be ignored by filtering rules, there is no change in behavior: the event is skipped and IDs are not checked. (Bug #44360, Bug #11753004)

  References: See also Bug #11763509.
- **Replication:** If a LOAD DATA INFILE statement—replicated using statement-based replication—featured a SET clause, the name-value pairs were regenerated using a method (Item::print()) intended primarily for generating output for statements such as EXPLAIN EXTENDED, and which cannot be relied on to return valid SQL. This could in certain cases lead to a crash on the slave.

  To fix this problem, the server now names each value in its original, user-supplied form, and uses that to create LOAD DATA INFILE statements for statement-based replication. (Bug #60580, Bug #11902767)

  References: See also Bug #34283, Bug #11752526, Bug #43746.
- **Replication:** Using the --server-id option with mysqlbinlog could cause format description log events to be filtered from the binary log, leaving mysqlbinlog unable to read the remainder of the log. Now such events are always read without regard to the value of this option.

  As part of the fix for this problem, mysqlbinlog now also reads rotate log events without regard to the value of --server-id. (Bug #59530, Bug #11766427)
- **Replication:** Error 1590 (ER_SLAVE_INCIDENT) caused the slave to stop even when it was started with --slave-skip-errors=1590. (Bug #59889, Bug #11768580, Bug #11799671)
- **Replication:** A failed DROP DATABASE statement could break statement-based replication. (Bug #58381, Bug #11765416)
- The Performance Schema caused a bottleneck for LOCK_open. (Bug #12993572)

- mysqld_safe ignored any value of plugin_dir specified in my.cnf files. (Bug #12925024)
- The metadata locking subsystem added too much overhead for INFORMATION_SCHEMA queries that were processed by opening only .frm or .TRG files and had to scan many tables. For example, SELECT COUNT(*) FROM INFORMATION_SCHEMA.TRIGGERS was affected. (Bug #12828477)
- With profiling disabled or not compiled in, set_thd_proc_info() unnecessarily checked file name lengths. (Bug #12756017)

  References: This bug is a regression of Bug #59273.
- Compilation failed on Mac OS X 10.7 (Lion) with a warning: Implicit declaration of function 'pthread_init' (Bug #12779790)
- The result for ANY subqueries with nested joins could be missing rows. (Bug #12795555)
- Compiling the server with maintainer mode enabled failed for gcc 4.6 or higher. (Bug #12727287)
- For prepared statements, an OK could be sent to the client if the prepare failed due to being killed. (Bug #12661349)
- Some Valgrind warnings were corrected:
  - For the SUBSTRING(), LEFT(), RIGHT(), LPAD(), RPAD(), and REPEAT() functions, a missing NULL value check was corrected.
  - For the LIKE operator, an attempt to use an uninitialized string buffer in the case of an empty wildcard was corrected.

  (Bug #12634989, Bug #59851, Bug #11766684)
- For debug builds, a field-type check raised an assertion if the type was MYSQL_TYPE_NULL. (Bug #12620084)
- Adding support for Windows authentication to libmysqlclient introduced a link dependency on the system Secur32 library. The Microsoft Visual C++ link information now pulls in this library automatically. (Bug #12612143)
- The option-parsing code for empty strings leaked memory. (Bug #12589928)
- With index condition pushdown enabled, a crash could occur due to an invalid end-of-range value. (Bug #12601961)
- The server could fail to free allocated memory when INSERT DELAYED was used with binary logging enabled. (Bug #12538873)
- A DBUG_ASSERT added by Bug #11792200 was overly aggressive in raising assertions. (Bug #12537160)
- In some cases, memory allocated for Query_tables_list::sroutines() was not freed properly. (Bug #12429877)
- Assignments to NEW.*var_name* within triggers, where *var_name* had a BLOB or TEXT type, were not properly handled and produced incorrect results. (Bug #12362125)
- After the fix for Bug #11889186, MAKEDATE() arguments with a year part greater than 9999 raised an assertion. (Bug #12403504)
- An assertion could be raised due to a missing NULL value check in Item_func_round::fix_length_and_dec(). (Bug #12392636)
- An assertion could be raised if Index Condition Pushdown code pushed down an index condition containing a subquery. (Bug #12355958)

- InnoDB could add temporary index information to INFORMATION_SCHEMA, which could raise an assertion. (Bug #12340873)
- On Windows, the server rejected client connections if no DNS server was available. (Bug #12325375)
- XA COMMIT could fail to clean up the error state if it discovered that the current XA transaction had to be rolled back. Consequently, the next XA transaction could raise an assertion when it checked for proper cleanup of the previous transaction. (Bug #12352846)
- An assertion could be raised during two-phase commits if the binary log was used as the transaction coordinator log. (Bug #12346411)
- A too-strict assertion could cause a server crash. (Bug #12321461)
- mysql_list_fields() returned incorrect character set information for character columns of views. (Bug #12337762)
- mysql_upgrade did not properly upgrade the authentication_string column of the mysql.user table. (Bug #11936829)
- The optimizer sometimes chose a forward index scan followed by a filesort to reverse the order rather than scanning the index in reverse order. (Bug #11882131)
- InnoDB invoked some zlib functions without proper initialization. (Bug #11849231)
- With index condition pushdown enabled, queries that used STRAIGHT_JOIN on data that included NULL values could return incorrect results. (Bug #11873324)
- Previously, an inappropriate error message was produced if a multiple-table update for an InnoDB table with a clustered primary key would update a table through multiple aliases, and perform an update that may physically move the row in at least one of these aliases. Now the error message is: Primary key/partition key update is not permitted since the table is updated both as '*tbl_name1*' and '*tbl_name2*' (Bug #11882110)

  References: See also Bug #11764529.
- Corrected a condition that produced an InnoDB message in the error log, unlock row could not find a 3 mode lock on the record. This situation could occur with a combination of a subquery and a FOR UPDATE clause under the READ UNCOMMITTED isolation level. The fix also improves the debuggability of such messages by including the original SQL statements that caused them. (Bug #11766322, Bug #59410)
- Division of large numbers could cause stack corruption. (Bug #11792200)
- With Valgrind enabled, InnoDB semaphore wait timeouts were too low and could expire. (Bug #11765460)
- CHECK TABLE and REPAIR TABLE failed to find problems with MERGE tables that had underlying tables missing or with the wrong storage engine. Issues were reported only for the first underlying table. (Bug #11754210)
- SHOW EVENTS did not always show events from the correct database. (Bug #41907, Bug #11751148)
- An assertion was raised if a statement tried to upgrade a metadata lock while there was an active FLUSH TABLE *tbl_list* WITH READ LOCK statement. Now if a statement tries to upgrade a metadata lock in this situation, the server returns an ER_TABLE_NOT_LOCKED_FOR_WRITE error to the client. (Bug #57649, Bug #11764779)
- With the conversion from GNU autotools to CMake for configuring MySQL, the USE_SYMDIR preprocessor symbol was omitted. This caused failure of symbolic links (described at Using Symbolic Links). (Bug #59408, Bug #11766320)

- The code for `PROCEDURE ANALYSE()` had a missing `DBUG_RETURN` statement, which could cause a server crash in debug builds. (Bug #58140, Bug #11765202)
- For a client connected using SSL, the `Ssl_cipher_list` status variable was empty and did not show the possible cipher types. (Bug #52596, Bug #11760210)
- In `Item::get_date`, a Valgrind warning for a missing `NULL` value check was corrected. (Bug #59164, Bug #11766124)
- In `Item_func_month::val_str()`, a Valgrind warning for a too-late `NULL` value check was corrected. (Bug #59166, Bug #11766126)
- In `Item_func::val_decimal`, a Valgrind warning for a missing `NULL` value check was corrected. (Bug #59125, Bug #11766087)
- In `Item_func_str_to_date::val_str`, a Valgrind warning for an uninitialized variable was corrected. (Bug #58154, Bug #11765216)
- In `extract_date_time()`, a Valgrind warning for a missing end-of-string check was corrected. (Bug #59151, Bug #11766112)
- A missing variable initialization for `Item_func_set_user_var` objects could raise an assertion. (Bug #59527, Bug #11766424)
- In string context, the `MIN()` and `MAX()` functions did not take into account the unsignedness of a `BIGINT UNSIGNED` argument. (Bug #59132, Bug #11766094)
- When used to upgrade tables, `mysqlcheck` (and `mysql_upgrade`, which invokes `mysqlcheck`) did not upgrade some tables for which table repair was found to be necessary. In particular, it failed to upgrade `InnoDB` tables that needed repair, leaving them in a nonupgraded state. This occurred because:

  - `mysqlcheck --check-upgrade ---auto-repair` checks for tables that are incompatible with the current version of MySQL. It does this by issuing the `CHECK TABLE ... FOR UPGRADE` statement and examining the result.
  - For any table found to be incompatible, `mysqlcheck` issues a `REPAIR TABLE` statement. But this fails for storage engines such as `InnoDB` that do not support the repair operation. Consequently, the table remained unchanged.

  To fix the problem, the following changes were made to `CHECK TABLE ... FOR UPGRADE` and `mysqlcheck`. Because `mysql_upgrade` invokes `mysqlcheck`, these changes also fix the problem for `mysql_upgrade`.

  - `CHECK TABLE ... FOR UPGRADE` returns a different error if a table needs repair but its storage engine does not support `REPAIR TABLE`:

    Previous:

    ```
    Error: ER_TABLE_NEEDS_UPGRADE
    Table upgrade required. Please do "REPAIR TABLE `tbl_name`" or
    dump/reload to fix it!
    ```

    Now:

    ```
    Error: ER_TABLE_NEEDS_REBUILD
    Table rebuild required. Please do "ALTER TABLE `tbl_name` FORCE" or
    dump/reload to fix it!
    ```

  - `mysqlcheck` recognizes the new error and issues an `ALTER TABLE ... FORCE` statement. The `FORCE` option for `ALTER TABLE` was recognized but did nothing; now it is implemented and acts as a “null” alter operation that rebuilds the table.

(Bug #47205, Bug #11755431)

- With prepared statements, the server could attempt to send result set metadata after the table had been closed. (Bug #56115, Bug #11763413)
- CREATE TRIGGER and DROP TRIGGER can change the prelocking list of stored routines, but the routine cache did not detect such changes, resulting in routine execution with an inaccurate locking list. (Bug #58674, Bug #11765684)
- On Windows, the authentication_string column recently added to the mysql.user table caused the Configuration Wizard to fail. (Bug #59038, Bug #11766011)
- The mysql_load_plugin() C API function did not clear the previous error. (Bug #60075, Bug #11766854)
- An invalid pathname argument for the --defaults-extra-file option of MySQL programs caused a program crash. (Bug #59234, Bug #11766184)
- The optimizer sometimes requested ordered access from a storage engine when ordered access was not required. (Bug #57601, Bug #11764737)
- Attempts to grant the EXECUTE or ALTER ROUTINE privilege for a nonexistent stored procedure returned success instead of an error. (Bug #51401, Bug #11759114)
- The optimizer sometimes incorrectly processed HAVING clauses for queries that did not also have an ORDER BY clause. (Bug #48916, Bug #11756928)
- Table I/O for the Performance Schema table_io_waits_summary_by_index_usage table was counted as using no index for UPDATE and DELETE statements, even when an index was used. (Bug #60905, Bug #12370950)
- An assertion could be raised in Item_func_int_val::fix_num_length_and_dec() due to overflow for geometry functions. (Bug #57900, Bug #11764994)
- With lower_case_table_names=2, resolution of objects qualified by database names could fail. (Bug #50924, Bug #11758687)
- The server permitted max_allowed_packet to be set lower than net_buffer_length, which does not make sense because max_allowed_packet is the upper limit on net_buffer_length values. Now a warning occurs and the value remains unchanged. (Bug #59959, Bug #11766769)
- LOAD DATA INFILE errors could leak I/O cache memory. (Bug #58072, Bug #11765141)
- Setting optimizer_join_cache_level to 3 or greater raised an assertion for some queries. (Bug #59651, Bug #11766522)
- PROCEDURE ANALYSE() could leak memory for NULL results, and could return incorrect results if used with a LIMIT clause. (Bug #48137, Bug #11756242)
- Selecting from a view for which the definition included a HAVING clause failed with an error:

  ```
  1356: View '...' references invalid table(s) or column(s)
  or function(s) or definer/invoker of view lack rights to use them
  ```

  (Bug #60295, Bug #11829681)
- The server did not check for certain invalid out of order sequences of XA statements, and these sequences raised an assertion. (Bug #59936, Bug #11766752, Bug #12348348)
- CREATE TABLE syntax permits specification of a STORAGE {DEFAULT|DISK|MEMORY} option. However, this value was not written to the .frm file, so that a subsequent CREATE TABLE ... LIKE for the table did not include that option.

Also, ALTER TABLE of a table that had a tablespace incorrectly destroyed the tablespace. (Bug #60111, Bug #11766883, Bug #34047, Bug #11747789)

- In Item_func_in::fix_length_and_dec(), a Valgrind warning for uninitialized values was corrected. (Bug #59270, Bug #11766212)
- An incorrect max_length value for YEAR values could be used in temporary result tables for UNION, leading to incorrect results. (Bug #59343, Bug #11766270)
- An internal client macro reference was removed from the client_plugin.h header file. This reference made the file unusable. (Bug #60746, Bug #12325444)
- In some cases, SHOW WARNINGS returned an empty result when the previous statement failed. (Bug #55847, Bug #11763166)
- Comparison of a DATETIME stored program variable and NOW() resulted in “Illegal mix of collations error” when character_set_connection was set to utf8. (Bug #60625, Bug #11926811)
- On Linux, the mysql client built using the bundled libedit did not read ~/.editrc. (Bug #49967, Bug #11757855)
- For some queries, the optimizer performed range analysis too many times for the same index. (Bug #59415, Bug #11766327)
- For LOAD DATA INFILE, multibyte character sequences could be pushed onto a stack too small to accommodate them. (Bug #58069, Bug #11765139)
- In ROUND() calculations, a Valgrind warning for uninitialized memory was corrected. (Bug #58937, Bug #11765923)

  References: This bug is a regression of Bug #33143.
- Table objects associated with one session's optimizer structures could be closed after being passed to another session, prematurely ending the second session's table or index scan. (Bug #56080, Bug #11763382)
- Valgrind warnings caused by comparing index values to an uninitialized field were corrected. (Bug #58705, Bug #11765713)
- The mysql client sometimes did not properly close sessions terminated by the user with **Control+C**. (Bug #52515, Bug #11760134)
- In debug builds, Field_new_decimal::store_value() was subject to buffer overflows. (Bug #55436, Bug #11762799)
- An attempt to install nonexistent files during installation was corrected. (Bug #43247, Bug #11752142)
- For repeated invocation of some stored procedures, the server consumed memory that it did not release until the connection terminated. (Bug #60025, Bug #11848763)
- Some tables were not instrumented by the Performance Schema even though they were listed in the setup_objects table. (Bug #59150, Bug #11766111)
- On some platforms, the Incorrect value: xxx for column yyy at row zzz error produced by LOAD DATA INFILE could have an incorrect value of *zzz*. (Bug #46895, Bug #11755168)
- For an InnoDB table, dropping and adding an index in a single ALTER TABLE statement could fail. (Bug #54927, Bug #11762345)

- An embedded client aborted rather than issuing an error message if it issued a `TEE` command (`\T file_name`) and the directory containing the file did not exist. This occurred because the wrong error handler was called. (Bug #57491, Bug #11764633)

- For an outer join with a `NOT IN` subquery in the `WHERE` clause, a null left operand to the `NOT IN` returned was treated differently than a literal `NULL` operand. (Bug #56881, Bug #11764086)

- As a side effect of optimizing `condition AND TRUE` or `condition OR FALSE`, MySQL for certain subqueries forgot that the columns used by the condition needed to be read, which raised an assertion in debug builds. (Bug #58690, Bug #11765699)

- Using `CREATE EVENT IF NOT EXISTS` for an event that already existed and was enabled caused multiple instances of the event to run. (Bug #61005, Bug #12546938)

- The range created by the optimizer when OR-ing two conditions could be incorrect, causing incorrect query results. (Bug #58834, Bug #11765831)

- A problem introduced in MySQL 5.5.11 caused very old (MySQL 4.0) clients to be unable to connect to the server. (Bug #61222, Bug #12563279)

- On FreeBSD 64-bit builds of the embedded server, exceptions were not prevented from propagating into the embedded application. (Bug #38965, Bug #11749418)

- In MySQL 5.1 and up, if a table had triggers that used syntax supported in 5.0 but not 5.1, the table became unavailable. Now the table is marked as having broken triggers. These should be dropped and recreated manually. (Bug #45235, Bug #11753738)

- `LOAD DATA INFILE` incorrectly parsed relative data file path names that ascended more than three levels in the file system and as a consequence was unable to find the file. (Bug #60987, Bug #12403662)

- Incorrect handling of metadata locking for `FLUSH TABLES WITH READ LOCK` for statements requiring prelocking caused two problems:

  - Execution of any data-changing statement that required prelocking (that is, involved a stored function or trigger) as part of a transaction slowed down somewhat all subsequent statements in the transaction. Performance in a transaction that periodically involved such statements gradually degraded over time.

  - Execution of any data-changing statement that required prelocking as part of a transaction prevented a concurrent `FLUSH TABLES WITH READ LOCK` from proceeding until the end of the transaction rather than at the end of the particular statement.

  (Bug #61401, Bug #12641342)

- `CREATE TABLE ... LIKE` for a `MyISAM` table definition that included an `DATA DIRECTORY` or `INDEX DIRECTORY` table option failed, instead of creating a table with those options omitted as documented. (Bug #52354, Bug #11759990)

- `ALTER TABLE {MODIFY|CHANGE} ... FIRST` did nothing except rename columns if the old and new versions of the table had exactly the same structure with respect to column data types. As a result, the mapping of column name to column data was incorrect. The same thing happened for `ALTER TABLE DROP COLUMN ... ADD COLUMN` statements intended to produce a new version of the table with exactly the same structure as the old version. (Bug #61493, Bug #12652385)

- A race condition between loading a stored routine using the name qualified by the database name and dropping that database resulted in a spurious error message: `The table mysql.proc is missing, corrupt, or contains bad data` (Bug #47870, Bug #11756013)

- The fractional part of the “Queries per second” value could be displayed incorrectly in MySQL status output (for example, in the output from `mysqladmin status` or the `mysql` `STATUS` command). (Bug #61205, Bug #12565712)

- Threads blocked in the `waiting for table metadata` state were not visible in `performance_schema.THREADS` or `SHOW PROFILE`. (Bug #56475, Bug #11763728)
- Previously, Performance Schema table columns that held byte counts were `BIGINT UNSIGNED`. These were changed to `BIGINT` (signed). This makes it easier to perform calculations that compute differences between columns. (Bug #59631, Bug #11766504)
- The server failed to compile if partitioning support was disabled. (Bug #61625, Bug #12694147)
- Upgrades using an RPM package recreated the `test` database, which is undesirable when the DBA had removed it. (Bug #45415, Bug #11753896)
- The `mysql-log-rotate` script was updated because it referred to deprecated MySQL options. (Bug #61038, Bug #12546842)
- For some statements such as `DESCRIBE` or `SHOW`, views with too many columns produced errors. (Bug #49437, Bug #11757397)
- For unknown users, the native password plugin reported incorrectly that no password had been specified even when it had. (Bug #59792, Bug #11766641)
- For queries with many `eq_ref` joins, the optimizer took excessive time to develop an execution plan. (Bug #41740, Bug #11751026, Bug #58225, Bug #11765274)
- The embedded server crashed when `argc = 0`. (Bug #57931, Bug #12561297)
- If a statement ended with mismatched quotes, the server accepted the statement and interpreted whatever was after the initial quote as a text string. (Bug #60993, Bug #12546960)
- Some status variables rolled over to zero after reaching the maximum 32-bit value. They have been changed to 64-bit values. (Bug #42698, Bug #11751727)
- `ALTER EVENT` could change the event status. (Bug #57156, Bug #11764334)
- A handled condition (error or warning) could be shown as not handled at the end of the statement. (Bug #55843, Bug #11763162)

  References: This bug is a regression of Bug #23032.
- `(5 DIV 2)` and `(5.0 DIV 2)` produced different results (2 versus 3) because the result of the latter expression was not truncated before conversion to integer. This differed from the behavior in MySQL 5.0 and 5.1. Now both expressions produce 2. (Bug #61676, Bug #12711164)
- `CREATE TABLE` without an `ENGINE` option determined the default engine at parse rather than execution time. This led to incorrect results if the statement was executed within a stored program and the default engine had been changed in the meantime. (Bug #50614, Bug #11758414)
- `SELECT DISTINCT` with a deterministic stored function in the `WHERE` clause could produce incorrect results. (Bug #59736, Bug #11766594)
- Spatial operations on certain corner cases could cause a server crash: Polygons with zero-point linerings; polygons with touching linerings. (Bug #51979, Bug #11759650, Bug #47429, Bug #11755628)
- Index condition pushdown code accessed an uninitialized variable. (Bug #59843, Bug #11766678)
- Fixed “shift count greater than width of type” compilation warnings. (Bug #60908, Bug #12402772)

  References: See also Bug #12561303.
- With `DISTINCT`, `CONCAT(`*`col_name,...`*`)` returned incorrect results when the arguments to `CONCAT()` were columns with an integer data type declared with a display width narrower than the values in the column. (For example, if an `INT(1)` column contained `1111`.) (Bug #4082)

Changes in MySQL 5.6.2 (2011-04-11)

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.

INFORMATION_SCHEMA Tables for InnoDB Buffer Pool Information

- **InnoDB:** The new INFORMATION_SCHEMA tables INNODB_BUFFER_PAGE, INNODB_BUFFER_PAGE_LRU, and INNODB_BUFFER_POOL_STATS display InnoDB buffer pool information for tuning on large-memory or highly loaded systems.

INFORMATION_SCHEMA Table for InnoDB Metrics

- **InnoDB:** A new INFORMATION_SCHEMA table, INNODB_METRICS, lets you query low-level InnoDB performance information, getting cumulative counts, averages, and minimum/maximum values for internal aspects of the storage engine operation. You can start, stop, and reset the metrics counters using the innodb_monitor_enable, innodb_monitor_disable, innodb_monitor_reset, and innodb_monitor_reset_all system variables.

Persistent InnoDB Optimizer Statistics

- **Performance; InnoDB:** The optimizer statistics for InnoDB tables can now persist across server restarts, producing more stable query performance. You can also control the amount of sampling done to estimate cardinality for each index, resulting in more accurate optimizer statistics. This feature involves the configuration options innodb_analyze_is_persistent (later replaced by innodb_stats_persistent), innodb_stats_persistent_sample_pages, and innodb_stats_transient_sample_pages, and the ANALYZE TABLE statement. See Configuring Persistent Optimizer Statistics Parameters for details.

INFORMATION_SCHEMA Tables for InnoDB Data Dictionary

- **InnoDB:** The InnoDB data dictionary, containing metadata about InnoDB tables, columns, indexes, and foreign keys, is available for SQL queries through a set of INFORMATION_SCHEMA tables.

InnoDB Configurable Data Dictionary Cache

- **InnoDB:** To ease the memory load on systems with huge numbers of tables, InnoDB now frees up the memory associated with an opened table using an LRU algorithm to select tables that have gone the longest without being accessed. To reserve more memory to hold metadata for open InnoDB tables, increase the value of the table_definition_cache configuration option. InnoDB treats this value as a “soft limit” for the number of open table instances in the InnoDB data dictionary cache. The actual number of tables with cached metadata could be higher than the value specified for table_definition_cache, because metadata for InnoDB system tables, and parent and child tables in foreign key relationships, is never evicted from memory. For additional information, refer to the table_definition_cache documentation. (Bug #20877, Bug #11745884)

Optimizer Notes

- The optimizer implements Index Condition Pushdown (ICP), an optimization for the case where MySQL retrieves rows from a table using an index. Without ICP, the storage engine traverses the index to locate rows in the base table and returns them to the MySQL server which evaluates the WHERE condition for the rows. With ICP enabled, and if parts of the WHERE condition can be evaluated by using only fields from the index, the MySQL server pushes this part of the WHERE condition down to the storage engine. The storage engine then evaluates the pushed index condition by using the index entry and only if this is satisfied is base row be read. ICP can reduce the number of accesses the storage engine has to do against the base table and the number of accesses the MySQL server has to do against the storage engine. For more information, see Index Condition Pushdown Optimization.

- The optimizer implements Disk-Sweep Multi-Range Read. Reading rows using a range scan on a secondary index can result in many random disk accesses to the base table when the table is large and not stored in the storage engine's cache. With the Disk-Sweep Multi-Range Read (MRR) optimization, MySQL tries to reduce the number of random disk access for range scans by first scanning the index only and collecting the keys for the relevant rows. Then the keys are sorted and finally the rows are retrieved from the base table using the order of the primary key. The motivation for Disk-sweep MRR is to reduce the number of random disk accesses and instead achieve a more sequential scan of the base table data. For more information, see Multi-Range Read Optimization.
- The optimizer now more efficiently handles queries (and subqueries) of the following form:

  ```
  SELECT ... FROM single_table ... ORDER BY non_index_column [DESC] LIMIT [M,]N;
  ```

  That type of query is common in web applications that display only a few rows from a larger result set. For example:

  ```
  SELECT col1, ... FROM t1 ... ORDER BY name LIMIT 10;
  SELECT col1, ... FROM t1 ... ORDER BY RAND() LIMIT 15;
  ```

  The sort buffer has a size of `sort_buffer_size`. If the sort elements for *`N`* rows are small enough to fit in the sort buffer (*`M`*+*`N`* rows if *`M`* was specified), the server can avoid using a merge file and perform the sort entirely in memory. For details, see Optimizing `LIMIT` Queries.

Explicit Partition Selection

- **Partitioning:** It is now possible to select one or more partitions or subpartitions when querying a partitioned table. In addition, many data modification statements (`DELETE`, `INSERT`, `REPLACE`, `UPDATE`, `LOAD DATA`, and `LOAD XML`) that act on partitioned tables also now support explicit partition selection. For example, assume we have a table named `t` with some integer column named `c`, and `t` has 4 partitions named `p0`, `p1`, `p2`, and `p3`. Then the query `SELECT * FROM t PARTITION (p0, p1) WHERE c < 5` returns rows only in partitions `p0` and `p1` that match the `WHERE` condition, whereas partitions `p2` and `p3` are not checked.

  For additional information and examples, see Partition Selection, as well as the descriptions of the statements just listed.

Performance Schema Notes

- The Performance Schema has these additions:
  - The Performance Schema now has tables that contain summaries for table and index I/O wait events, as generated by the `wait/io/table/sql/handler` instrument:
    - `table_io_waits_summary_by_table`: Aggregates table I/O wait events. The grouping is by table.
    - `table_io_waits_summary_by_index_usage`: Aggregates table index I/O wait events. The grouping is by table index.

    The information in these tables can be used to assess the impact of table I/O performed by applications. For example, it is possible to see which tables are used and which indexes are used (or not used), or to identify bottlenecks on a table when multiple applications access it. The results may be useful to change how applications issue queries against a database, to minimize application footprint on the server and to improve application performance and scalability.

A change that accompanies the new tables is that the events_waits_current table now has an INDEX_NAME column to identify which index was used for the operation that generated the event. The same is true of the event-history tables, events_waits_history, and events_waits_history_long.

- The Performance Schema now has an instrument named wait/lock/table/sql/handler in the setup_instruments table for instrumenting table lock wait events. It differs from wait/io/table/sql/handler, which instruments table I/O. This enables independent instrumentation of table I/O and table locks.

  Accompanying the new instrument, the Performance Schema has a table named table_lock_waits_summary_by_table that aggregates table lock wait events, as generated by the new instrument. The grouping is by table.

  The information in this table may be used to assess the impact of table locking performed by applications. The results may be useful to change how applications issue queries against the database and use table locks, to minimize the application footprint on the server and to improve application performance and scalability. For example, an application locking tables for a long time may negatively affect other applications; the instrumentation makes this visible.

- To selectively control which tables to instrument for I/O and locking, use the setup_objects table. See Pre-Filtering by Object.

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

For more information, see MySQL Performance Schema.

Pluggable Authentication

- MySQL distributions now include auth_socket, a server-side authentication plugin that authenticates clients that connect from the local host through the Unix socket file. The plugin uses the SO_PEERCRED socket option to obtain information about the user running the client program (and thus can be built only on systems that support this option). For a connection to succeed, the plugin requires a match between the login name of the connecting client user and the MySQL user name presented by the client program. For more information, see The Socket Peer-Credential Authentication Plugin. (Bug #59017, Bug #11765993, Bug #9411, Bug #11745104)

- MySQL distributions now include mysql_clear_password, a client-side authentication plugin that sends the password to the server without hashing or encryption. Although this is insecure, and thus appropriate precautions should be taken (such as using an SSL connection), the plugin is useful in conjunction with server-side plugins that must have access to the original password in clear text. For more information, see The Cleartext Client-Side Authentication Plugin.

Crash-Safe Binary Log

- **Replication:** The MySQL Server now records and reads back only complete events or transactions to and from the binary log. By default, the server now logs the length of the event as well as the event itself and uses this information to verify that the event was written correctly to the log. A master also uses by default this value to verify events when reading from the binary log.

  If you enable writing of checksums (using the binlog_checksum system variable), the master can use these instead by enabling the master_verify_checksum system variable. The slave I/O thread also verifies events received from the master. You can cause the slave SQL thread to use checksums (if available) as well, when reading from the relay log, by enabling the slave_sql_verify_checksum system variable on the slave.

- **Replication:** Support for checksums when writing and reading the binary log is added to the MySQL Server. Writing checksums into the binary log is disabled by default; it can be enabled by starting the server with the --binlog-checksum option. To cause the server to read checksums from the binary log, start the server with the `--master-verify-checksum` option. The `--slave-sql-verify-checksum` option causes the slave to read checksums from the relay log.

Slave Log Tables

- **Replication:** It is now possible to write information about the slave connection to the master and about the slave's execution point within the relay log to tables rather than files. Logging of master connection information and of slave relay log information to tables can be done independently of one another; this is controlled by the `--master-info-repository` and `--relay-log-info-repository` server options. When `--master-info-repository` is set to `TABLE`, connection information is logged in the `slave_master_info` table in the `mysql` system database. When `--relay-log-info-repository` is set to `TABLE`, relay log information is logged to the `slave_relay_log_info` table, also in the `mysql` database.

Row Image Control

- **Replication:** Added the `binlog_row_image` server system variable, which can be used to enable row image control for row-based replication. This means that you can potentially save disk space, network resources, and memory usage by the MySQL Server by logging only those columns that are required for uniquely identifying rows, or which are actually changed on each row, as opposed to logging all columns for each and every row change event. In addition, you can use a “noblob” mode where all columns, except for unneeded `BLOB` or `TEXT` columns, are logged.

  For more information, see System Variables Used with Binary Logging. (Bug #47200, Bug #11755426, Bug #47303, Bug #56917, Bug #11755426, Bug #11755513, Bug #11764116)

Functionality Added or Changed

- **Performance; InnoDB:** A separate `InnoDB` thread (`page_cleaner`) now handles the flushing of dirty pages that was formerly done by the `InnoDB` master thread. (Bug #11762412, Bug #55004)

- **Performance; InnoDB:** The `InnoDB` kernel mutex, which controls concurrent access to the `InnoDB` kernel, has been split into several mutexes and rw-locks, for improved concurrency.

- **Performance; InnoDB:** The `innodb_purge_threads` system variable can now be set to a value higher than 1.

- **Incompatible Change:** The following obsolete constructs have been removed. Where alternatives are shown, applications should be updated to use them.

  - The `FLUSH MASTER` and `FLUSH SLAVE` statements. Use the `RESET MASTER` and `RESET SLAVE` statements instead.

- **Important Change; Replication:** Added the `--binlog-rows-query-log-events` option for `mysqld`. Using this option causes a server logging in row-based mode to write informational *rows query log* events (SQL statements, for debugging and other purposes) to the binary log. MySQL server and MySQL programs from MySQL 5.6.2 and later normally ignore such events, so that they do not pose an issue when reading the binary log. `mysqld` and `mysqlbinlog` from previous MySQL releases cannot read such events in the binary log, and fail if they attempt to do so. For this reason, you should never prepare logs for a MySQL 5.6.1 or earlier replication slave server (or other reader such as `mysqlbinlog`) with this option enabled on the master. (Bug #11758695, Bug #50935, Bug #11758695)

- **InnoDB:** `InnoDB` can optionally log details about all deadlocks that occur, to assist with troubleshooting and diagnosis. This feature is controlled by the `innodb_print_all_deadlocks` system variable. (Bug #1784, Bug #17572)

- **Replication:** On MySQL replication slaves having multiple network interfaces, it is now possible to set which interface to use for connecting to the master using the `MASTER_BIND='interface'` option in a `CHANGE MASTER TO` statement.

The value set by this option can be seen in the Master_Bind column of the output from SHOW SLAVE STATUS or the Bind column of the mysql.slave_master_info table. (Bug #25939, Bug #11746389)

- **Replication:** Added the log_bin_basename system variable, which contains the complete file name and path to the binary log file. (The log_bin system variable shows only whether or not binary logging is enabled; log_bin_basename, however, reflects the name set with the --log-bin server option.) Also added relay_log_basename system variable, which shows the file name and complete path to the relay log file.

  References: See also Bug #19614, Bug #11745759.

- The mysql_upgrade, mysqlbinlog, mysqlcheck, mysqlimport, mysqlshow, and mysqlslap clients now have --default-auth and --plugin-dir options for specifying which authentication plugin and plugin directory to use. (Bug #58139)

- Boolean system variables can be enabled at run time by setting them to the value ON or OFF, but previously this did not work at server startup. Now at startup such variables can be enabled by setting them to ON or TRUE, or disabled by setting them to OFF or FALSE. Any other nonnumeric value is invalid. (Bug #46393)

  References: See also Bug #11754743, Bug #51631.

- Previously, for queries that were aborted due to a sort problem, the server wrote the message Sort aborted to the error log. Now the server writes more information to provide a more specific message, such as:

  ```
  Sort aborted: Out of memory (Needed 24 bytes)
  Out of sort memory, consider increasing server sort buffer size
  Sort aborted: Out of sort memory, consider increasing server sort
    buffer size
  Sort aborted: Incorrect number of arguments for FUNCTION test.f1;
    expected 0, got 1
  ```

  In addition, if the server was started with --log-warnings=2, the server writes information about the host, user, and query. (Bug #36022, Bug #11748358)

- MySQL distributions now include an INFO_SRC file that contains information about the source distribution, such as the MySQL version from which it was created. MySQL binary distributions additionally include an INFO_BIN file that contains information about how the distribution was built, such as compiler options and feature flags. In RPM packages, these files are located in the /usr/share/doc/packages/MySQL-server directory. In tar.gz and derived packages, they are located in the Docs directory under the location where the distribution is unpacked. (Bug #42969, Bug #11751935)

- If the --init-file option is given, the server now writes messages indicating the beginning and end of file execution to the error log. (Bug #48387, Bug #11756463)

- The server now writes thread shutdown messages to the error log during the shutdown procedure. (Bug #48388, Bug #11756464)

- The server now includes the thread ID in rows written to the slow query log. In the slow query log file, the thread ID is the last value in the line. In the mysql.slow_log log table, there is a new thread_id column.

  To update the slow_log table if you are upgrading from an earlier release, run mysql_upgrade and restart the server. See mysql_upgrade — Check and Upgrade MySQL Tables. (Bug #53630, Bug #11761166)

- Multi-read range access is now based on cost estimates and no longer used for simple queries for which it is not beneficial. (Bug #37576, Bug #11748865)

- mysqldump --xml now displays comments from column definitions. (Bug #13618, Bug #11745324)
- A new plugin service, my_plugin_log_service, enables plugins to report errors and specify error messages. The server writes the messages to the error log. See MySQL Services for Plugins.
- Previously, for queries that were aborted due to a sort problem or terminated with KILL in the middle of a sort, the server wrote the message Sort aborted to the error log. Now the server writes more information about the cause of the error. These causes include:
  - Insufficient disk space in the temporary file directory prevented a temp file from being created
  - Insufficient memory for sort_buffer_size to be allocated
  - Somebody ran KILL *id* in the middle of a filesort operation
  - The server was shut down while some queries were sorting
  - A transaction was rolled back or aborted due to a lock wait timeout or deadlock
  - Unexpected errors, such as a source table or even temp table was corrupt
  - Processing of a subquery failed which was also sorting

  (Bug #30771, Bug #11747102)
- The undocumented SHOW NEW MASTER statement has been removed.
- Windows provides APIs based on UTF-16LE for reading from and writing to the console. MySQL now supports a utf16le character set for UTF-16LE, and the mysql client for Windows has been modified to provide improved Unicode support by using these APIs.

  To take advantage of this change, you must run mysql within a console that uses a compatible Unicode font and set the default character set to a Unicode character set that is supported for communication with the server. For instructions, see Unicode Support on Windows.

Bugs Fixed

- **Security Fix:** Pre-evaluation of LIKE predicates during view preparation could cause a server crash. (Bug #54568, Bug #11762026, CVE-2010-3836)
- **Performance; InnoDB:** An UPDATE statement for an InnoDB table could be slower than necessary if it changed a column covered by a prefix index, but did not change the prefix portion of the value. The fix improves performance for InnoDB 1.1 in MySQL 5.5 and higher, and the InnoDB Plugin for MySQL 5.1. (Bug #58912, Bug #11765900)
- **Incompatible Change; Replication:** It is no longer possible to issue a CREATE TABLE ... SELECT statement which changes any tables other than the table being created. Any such statement is not executed and instead fails with an error.

  One consequence of this change is that FOR UPDATE may no longer be used at all with the SELECT portion of a CREATE TABLE ... SELECT.

  This means that, prior to upgrading from a previous release, you should rewrite any CREATE TABLE ... SELECT statements that cause changes in other tables so that the statements no longer do so.

  This change also has implications for statement-based replication between a MySQL 5.6 (or later slave) and a master running a previous version of MySQL. In such a case, if a CREATE TABLE ... SELECT statement on the master that causes changes in other tables succeeds on the master, the statement nonetheless fails on the slave, causing replication to stop. To keep this from happening, you should either use row-based replication, or rewrite the offending statement before running it on the master. (Bug #11749792, Bug #11745361, Bug #39804, Bug #55876)

References: See also Bug #47899.

- **Incompatible Change:** When auto_increment_increment is greater than one, values generated by a bulk insert that reaches the maximum column value could wrap around rather producing an overflow error.

  As a consequence of the fix, it is no longer possible for an auto-generated value to be equal to the maximum BIGINT UNSIGNED value. It is still possible to store that value manually, if the column can accept it. (Bug #39828, Bug #11749800)

- **Important Change; Partitioning:** Date and time functions used as partitioning functions now have the types of their operands checked; use of a value of the wrong type is now disallowed in such cases. In addition, EXTRACT(WEEK FROM *col_name*), where *col_name* is a DATE or DATETIME column, is now disallowed altogether because its return value depends on the value of the default_week_format system variable. (Bug #54483, Bug #11761948)

  References: See also Bug #57071, Bug #11764255.

- **Important Change; Replication:** The CHANGE MASTER TO statement required the value for RELAY_LOG_FILE to be an absolute path, whereas the MASTER_LOG_FILE path could be relative.

  The inconsistent behavior is resolved by permitting relative paths for RELAY_LOG_FILE, in which case the path is assumed to be relative to the slave's data directory. (Bug #12190, Bug #11745232)

- **InnoDB; Partitioning:** The partitioning handler did not pass locking information to a table's storage engine handler. This caused high contention and thus slower performance when working with partitioned InnoDB tables. (Bug #59013)

- **InnoDB:** This fix introduces a new configuration option, innodb_change_buffer_max_size, which defines the size of the change buffer as a percentage of the size of the buffer pool. Because the change buffer shares memory space with the buffer pool, a workload with a high rate of DML operations could cause pages accessed by queries to age out of the buffer pool sooner than desirable. This fix also devotes more I/O capacity to flushing entries from the change buffer when it exceeds 1/2 of its maximum size. (Bug #11766168, Bug #59214)

- **InnoDB:** It was not possible to query the information_schema.INNODB_TRX table while other connections were running queries involving BLOB types. (Bug #55397, Bug #11762763)

- **InnoDB:** The presence of a double quotation mark inside the COMMENT field for a column could prevent a foreign key constraint from being created properly. (Bug #59197, Bug #11766154)

- **InnoDB:** InnoDB returned values for “rows examined” in the query plan that were higher than expected. NULL values were treated in an inconsistent way. The inaccurate statistics could trigger “false positives” in combination with the max_join_size setting, because the queries did not really examine as many rows as reported.

  A new configuration option innodb_stats_method lets you specify how NULL values are treated when calculating index statistics. Allowed values are nulls_equal (the default), nulls_unequal and null_ignored. The meanings of these values are similar to those of the myisam_stats_method option. (Bug #30423)

- **Partitioning:** Failed ALTER TABLE ... PARTITION statements could cause memory leaks. (Bug #56380, Bug #11763641)

  References: See also Bug #46949, Bug #11755209, Bug #56996, Bug #11764187.

- **Replication:** When using the statement-based logging format, INSERT ON DUPLICATE KEY UPDATE and INSERT IGNORE statements affecting transactional tables that did not fail were not written to the binary log if they did not insert any rows. (With statement-based logging, all successful statements should be logged, whether they do or do not cause any rows to be changed.) (Bug #59338, Bug #11766266)

- **Replication:** Formerly, STOP SLAVE stopped the slave I/O thread first and then stopped the slave SQL thread; thus, it was possible for the I/O thread to stop after replicating only part of a transaction which the SQL thread was executing, in which case—if the transaction could not be rolled back safely—the SQL thread could hang.

  Now, STOP SLAVE stops the slave SQL thread first and then stops the I/O thread; this guarantees that the I/O thread can fetch any remaining events in the transaction that the SQL thread is executing, so that the SQL thread can finish the transaction if it cannot be rolled back safely. (Bug #58546, Bug #11765563)

- **Replication:** The --help text for mysqlbinlog now indicates that the --verbose (-v) option outputs pseudo-SQL that is not necessarily valid SQL and cannot be guaranteed to work verbatim in MySQL clients. (Bug #47557, Bug #11755743)

- **Replication:** mysqlbinlog printed USE statements to its output only when the default database changed between events. To illustrate how this could cause problems, suppose that a user issued the following sequence of statements:

  ```
  CREATE DATABASE mydb;
  USE mydb;
  CREATE TABLE mytable (column_definitions);
  DROP DATABASE mydb;
  CREATE DATABASE mydb;
  USE mydb;
  CREATE TABLE mytable (column_definitions);
  ```

  When played back using mysqlbinlog, the second CREATE TABLE statement failed with Error: No Database Selected because the second USE statement was not played back, due to the fact that a database other than mydb was never selected.

  This fix ensures that mysqlbinlog outputs a USE statement whenever it reads one from the binary log. (Bug #50914, Bug #11758677)

- Two unused test files in storage/ndb/test/sql contained incorrect versions of the GNU Lesser General Public License. The files and the directory containing them have been removed. (Bug #11810224)

  References: See also Bug #11810156.

- An OUTER JOIN query using WHERE *col_name* IS NULL could return an incorrect result. (Bug #58490, Bug #11765513)

- When using ExtractValue() or UpdateXML(), if the XML to be read contained an incomplete XML comment, MySQL read beyond the end of the XML string when processing, leading to a crash of the server. (Bug #44332, Bug #11752979)

- Issuing EXPLAIN EXTENDED for a query that would use condition pushdown could cause mysqld to crash. (Bug #58553, Bug #11765570)

- An assertion could be raised if –1 was inserted into an AUTO_INCREMENT column by a statement writing more than one row. (Bug #50619, Bug #11758417)

- On FreeBSD, if mysqld was killed with a SIGHUP signal, it could corrupt InnoDB .ibd files. (Bug #51023, Bug #11758773)

- An uninitialized variable for the index condition pushdown access method could result in a server crash or Valgrind warnings. (Bug #58837, Bug #11765834)

- If filesort fell back to an ordinary sort/merge, it could fail to handle memory correctly. (Bug #59331, Bug #11766260)

- In a subquery, a UNION with no referenced tables (or only a reference to the DUAL virtual table) did not permit an ORDER BY clause. (Bug #58970, Bug #11765950)

- `MIN(year_col)` could return an incorrect result in some cases. (Bug #59211, Bug #11766165)
- Comparisons of aggregate values with `TIMESTAMP` values were incorrect. (Bug #59330, Bug #11766259)
- If `max_allowed_packet` was set larger than 16MB, the server failed to reject too-large packets with “Packet too large” errors. (Bug #58887, Bug #11765878)
- `OPTIMIZE TABLE` for an `InnoDB` table could raise an assertion if the operation failed because it had been killed. (Bug #58933, Bug #11765920)
- `--autocommit=ON` did not work (it set the global `autocommit` value to 0, not 1). (Bug #59432, Bug #11766339)
- Valgrind warnings about uninitialized variables were corrected. (Bug #59145, Bug #11766106)
- The fix for Bug #25192 caused `load_defaults()` to add an argument separator to distinguish options loaded from option files from those provided on the command line, whether or not the application needed it. (Bug #57953, Bug #11765041)

  References: See also Bug #11746296.
- The `DEFAULT_CHARSET` and `DEFAULT_COLLATION` `CMake` options did not work. (Bug #58991, Bug #11765967)
- When `mysqladmin` was run with the `--sleep` and `--count` options, it went into an infinite loop executing the specified command. (Bug #58221, Bug #11765270)
- `mysqlslap` failed to check for a `NULL` return from `mysql_store_result()` and crashed trying to process the result set. (Bug #59109, Bug #11766074)
- `DELETE` or `UPDATE` statements could fail if they used `DATE` or `DATETIME` values with a year, month, or day part of zero. (Bug #59173)
- Running a query against an `InnoDB` table twice, first with index condition pushdown enabled and then with it disabled, could produce different results. (Bug #58816, Bug #11765813)
- For `DIV` expressions, assignment of the result to multiple variables could cause a server crash. (Bug #59241, Bug #11766191)

  References: See also Bug #8457.
- If one connection locked the `mysql.func` table using either `FLUSH TABLES WITH READ LOCK` or `LOCK TABLE mysql.func WRITE` and a second connection tried to either create or drop a UDF function, a deadlock occurred when the first connection tried to use a UDF function. (Bug #53322, Bug #11760878)
- The parser failed to initialize some internal objects properly, which could cause a server crash in the cleanup phase after statement execution. (Bug #47511, Bug #11755703)
- In debug builds, `SUBSTRING_INDEX(FORMAT(...), FORMAT(...))` could cause a server crash. (Bug #58371, Bug #11765406)
- A `NOT IN` predicate with a subquery containing a `HAVING` clause could retrieve too many rows, when the subquery itself returned `NULL`. (Bug #58818, Bug #11765815)
- Outer joins on a unique key could return incorrect results. (Bug #57034, Bug #11764219)
- The `ESCAPE` clause for the `LIKE` operator permits only expressions that evaluate to a constant at execution time, but aggregate functions were not being rejected. (Bug #59149, Bug #11766110)
- `WHERE` conditions of the following forms were evaluated incorrectly and could return incorrect results:

```
WHERE null-valued-const-expression NOT IN (subquery)
WHERE null-valued-const-expression IN (subquery) IS UNKNOWN
```

(Bug #58628, Bug #11765642)

- Condition pushdown optimization could push down conditions with incorrect column references. (Bug #58134, Bug #11765196)

- Outer joins with an empty table could produce incorrect results. (Bug #58422, Bug #11765451)

- A query that contained an aggregate function but no `GROUP BY` clause was implicitly grouped. But implicitly grouped queries return zero or one row, so ordering does not make sense. (Bug #47853)

- The server and client did not always properly negotiate authentication plugin names. (Bug #59453, Bug #11766356)

- `SHOW PRIVILEGES` did not display a row for the `PROXY` privilege. (Bug #59275, Bug #11766216)

- The Performance Schema did not update status handler status variables, so `SHOW STATUS LIKE '%handler%'` produced undercounted values. (Bug #59799, Bug #11766645)

- The `mysql` client went into an infinite loop if the standard input was a directory. (Bug #57450, Bug #11764598)

- Aggregation followed by a subquery could produce an incorrect result. (Bug #59839, Bug #11766675)

- A query of the following form returned an incorrect result, where the values for *`col_name`* in the result set were entirely replaced with `NULL` values:

  ```
  SELECT DISTINCT col_name ... ORDER BY col_name DESC;
  ```

  (Bug #59308, Bug #11766241)

- `mysqldump` did not quote database names in `ALTER DATABASE` statements in its output, which could cause an error at reload time for database names containing a dash. (Bug #59398, Bug #11766310)

- `SHOW PROFILE` could truncate source file names or fail to show function names. (Bug #59273, Bug #11766214)

- Memory leaks detected by Valgrind, some of which could cause incorrect query results, were corrected. (Bug #59110, Bug #11766075)

- Some string-manipulating SQL functions use a shared string object intended to contain an immutable empty string. This object was used by the SQL function `SUBSTRING_INDEX()` to return an empty string when one argument was of the wrong data type. If the string object was then modified by the SQL function `INSERT()`, undefined behavior ensued. (Bug #58165, Bug #11765225)

- Some RPM installation scripts used a hardcoded value for the data directory, which could result in a failed installation for users who have a nonstandard data directory location. The same was true for other configuration values such as the PID file name. (Bug #56581, Bug #11763817)

- On some systems, debug builds of `comp_err` could fail due to an uninitialized variable. (Bug #59906, Bug #11766729)

- If a multiple-table update updated a row through two aliases and the first update physically moved the row, the second update failed to locate the row. This resulted in different errors depending on the storage engine, although these errors did not accurately describe the problem:

  - `MyISAM`: `Got error 134 from storage engine`

  - `InnoDB`: `Can't find record in 'tbl'`

For MyISAM, which is nontransactional, the update executed first was performed but the second was not. In addition, for two equal multiple-table update statements, one could succeed and the other fail depending on whether the record actually moved, which is inconsistent.

Now such an update returns an error if it will update a table through multiple aliases, and perform an update that may physically move the row in at least one of these aliases. (Bug #57373, Bug #11764529, Bug #55385, Bug #11762751)

- DES_DECRYPT() could crash if the argument was not produced by DES_ENCRYPT(). (Bug #59632, Bug #11766505)
- On FreeBSD and OpenBSD, the server incorrectly checked the range of the system date, causing legal values to be rejected. (Bug #55755, Bug #11763089)
- Parsing nested regular expressions could lead to recursion resulting in a stack overflow crash. (Bug #58026, Bug #11765099)
- With index condition pushdown enabled, a join could produce an extra row due to parts of the select condition for the second table in the join not being evaluated. (Bug #59186, Bug #11766144)
- injector::transaction did not have support for rollback. (Bug #58082, Bug #11765150)
- With index condition pushdown enabled, incorrect results were returned for queries on MyISAM tables involving HAVING and LIMIT, when the column in the WHERE condition contained NULL. (Bug #58838, Bug #11765835)
- Starting the server with the --defaults-file=*file_name* option, where the file name had no extension, caused a server crash. (Bug #58455, Bug #11765482)
- Internally, XOR items partially behaved like functions and partially as conditions. This resulted in inconsistent handling and crashes. The issue is fixed by consistently treating XOR items as functions. (Bug #59793, Bug #11766642)
- SHOW WARNINGS output following EXPLAIN EXTENDED could include unprintable characters. (Bug #57341, Bug #11764503)
- An assertion was raised if a stored routine had a DELETE IGNORE statement that failed but due to the IGNORE had not reported any error. (Bug #58709, Bug #11765717)
- For a query that used a subquery that included GROUP BY inside a < ANY() construct, no rows were returned when there should have been. (Bug #56690, Bug #11763918)
- There was an erroneous restriction on file attributes for LOAD DATA INFILE. (Bug #59085, Bug #11766052)
- SHOW CREATE TRIGGER failed if there was a temporary table with the same name as the trigger subject table. (Bug #58996, Bug #11765972)
- DISTINCT aggregates on DECIMAL UNSIGNED fields could trigger an assertion. (Bug #52171, Bug #11759827)
- Bitmap functions used in one thread could change bitmaps used by other threads, raising an assertion. (Bug #43152, Bug #11752069)
- Attempting to create a spatial index on a CHAR column longer than 31 bytes led to an assertion failure if the server was compiled with safemutex support. (Bug #59888, Bug #11766714)
- An assertion was raised if an XA COMMIT was issued when an XA transaction had already encountered an error (such as a deadlock) that required the transaction to be rolled back. (Bug #59986, Bug #11766788)
- The MYSQL_HOME environment variable was being ignored. (Bug #59280, Bug #11766219)

- When CASE ... WHEN arguments had different character sets, 8-bit values could be referenced as utf16 or utf32 values, raising an assertion. (Bug #44793, Bug #11753363)
- An incorrect character set pointer passed to my_strtoll10_mb2() caused an assertion to be raised. (Bug #59648, Bug #11766519)
- The “greedy” query plan optimizer failed to consider the size of intermediate query results when calculating the cost of a query. This could result in slowly executing queries when there are much faster execution plans available. (Bug #59326, Bug #11766256)
- FIND_IN_SET() could work differently in MySQL 5.5 than in 5.1. (Bug #59405, Bug #11766317)
- Setting the optimizer_switch system variable to an invalid value caused a server crash. (Bug #59894, Bug #11766719)
- On Windows, an object in thread local storage could be used before the object was created. (Bug #55730, Bug #11763065)
- Queries that used COALESCE() with cp1251 strings could result in an “illegal mix of collations” error. (Bug #60101, Bug #11766874)
- DATE_ADD() and DATE_SUB() return a string if the first argument is a string, but incorrectly returned a binary string. Now they return a character string with a collation of connection_collation. (Bug #31384, Bug #11747221)

Changes in MySQL 5.6.1 (Not released, Milestone 5)

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.

Performance Schema Notes

- The Performance Schema has these additions:
  - The setup_consumers table contents have changed. Previously, the table used a “flat” structure with a one-to-one correspondence between consumer name and destination table. This has been replaced with a hierarchy of consumer settings that enable progressively finer control of which destinations receive events. The previous *xxx*_summary_*xxx* consumers no longer exist. Instead, the Performance Schema maintains appropriate summaries automatically for the levels for which settings in the consumer hierarchy are enabled. For example, if only the top-level (global) consumer is enabled, only global summaries are maintained. Others, such as thread-level summaries, are not. See Pre-Filtering by Consumer. In addition, optimizations have been added to reduce Performance Schema overhead.
  - It is now possible to filter events by object using the new setup_objects table. Currently, this table can be used to selectively instrument tables, based on schema names and/or table names. See Pre-Filtering by Object. A new table, objects_summary_global_by_type, summarizes events for objects.
  - It is now possible to filter events by thread, and the Performance Schema collects more information for each thread. A new table, setup_actors, can be used to selectively instrument user connections, based on the user name and/or host name of each connecting session. The threads table, which contains a row for each active server thread, was extended with several new columns. With these additions, the information available in threads is like that available from the INFORMATION_SCHEMA.PROCESSLIST table or the output from SHOW PROCESSLIST. Thus,

all three serve to provide information for thread-monitoring purposes. Use of threads differs from use of the other two thread information sources in these ways:

- Access to threads does not require a mutex and has minimal impact on server performance. INFORMATION_SCHEMA.PROCESSLIST and SHOW PROCESSLIST have negative performance consequences because they require a mutex.
- threads provides additional information for each thread, such as whether it is a foreground or background thread, and the location within the server associated with the thread.
- threads provides information about background threads. This means that threads can be used to monitor activity the other thread information sources cannot.
- You can control which threads are monitored by setting the INSTRUMENTED column or by using the setup_actors table.

For these reasons, DBAs who perform server monitoring using INFORMATION_SCHEMA.PROCESSLIST or SHOW PROCESSLIST may wish to monitor using threads instead.

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

For more information, see MySQL Performance Schema.

Functionality Added or Changed

- **Incompatible Change:** The following obsolete constructs have been removed. Where alternatives are shown, applications should be updated to use them.
  - The --log server option and the log system variable. Instead, use the --general_log option to enable the general query log and the --general_log_file=*file_name* option to set the general query log file name.
  - The --log-slow-queries server option and the log_slow_queries system variable. Instead, use the --slow_query_log option to enable the slow query log and the --slow_query_log_file=*file_name* option to set the slow query log file name.
  - The --one-thread server option. Use --thread_handling=no-threads instead.
  - The --skip-thread-priority server option.
  - The engine_condition_pushdown system variable. Use the engine_condition_pushdown flag of the optimizer_switch variable instead.
  - The have_csv, have_innodb, have_ndbcluster, and have_partitioning system variables. Use SHOW ENGINES instead.
  - The sql_big_tables system variable. Use big_tables instead.
  - The sql_low_priority_updates system variable. Use low_priority_updates instead.
  - The sql_max_join_size system variable. Use max_join_size instead.
  - The SLAVE START and SLAVE STOP statements. Use the START SLAVE and STOP SLAVE statements instead.
  - The ONE_SHOT modifier for the SET statement.
- **Important Change; Replication:** Replication filtering options such as --replicate-do-db, --replicate-rewrite-db, and --replicate-do-table were not consistent with one another in regard to case sensitivity. Now all --replicate-* options follow the same rules for case sensitivity

applying to names of databases and tables elsewhere in the MySQL server, including the effects of the `lower_case_table_names` system variable. (Bug #51639, Bug #11759334)

- **Important Change; Replication:** Added the `MASTER_RETRY_COUNT` option to the `CHANGE MASTER TO` statement, and a corresponding `Master_Retry_Count` column to the output of `SHOW SLAVE STATUS`. The option sets the value shown in this column. `MASTER_RETRY_COUNT` is intended eventually to replace the older (and now deprecated) `--master-retry-count` server option, and is now the preferred method for setting the maximum number of times that the slave may attempt to reconnect after losing its connection to the master. (Bug #44209, Bug #11752887, Bug #44486, Bug #11753110)

- **InnoDB:** Setting `innodb_read_ahead_threshold` to `0` disables read-ahead. Prior to 5.6.1, a value of `0` would trigger a read-ahead upon reading the boundary page of a 64 page extent. (Bug #11763876, Bug #56646)

- **InnoDB:** `InnoDB` can now report the total size of the rollback segment, measured in pages. The value is reported through the `information_schema.innodb_metrics` table, using the counter `trx_rseg_curent_size`. You enable and query the counter as follows:

```
mysql (information_schema) > set global innodb_monitor_enable = 'trx_rseg_curent_size';

mysql (information_schema) > select name, count, max_count, comment from innodb_metrics where name =
+----------------------+-------+-----------+----------------------------------------+
| name                 | count | max_count | comment                                |
+----------------------+-------+-----------+----------------------------------------+
| trx_rseg_curent_size |   346 |       346 | Current rollback segment size in pages |
+----------------------+-------+-----------+----------------------------------------+
```

  (Bug #57584)

- **Replication:** Added the `Slave_last_heartbeat` status variable, which shows when a replication slave last received a heartbeat signal. The value is displayed using `TIMESTAMP` format. (Bug #45441)

- **Replication:** `SHOW SLAVE STATUS` now displays the actual number of retries for each connection attempt made by the I/O thread. (Bug #56416, Bug #11763675)

- **Replication:** Timestamps have been added to the output of `SHOW SLAVE STATUS` to show when the most recent I/O and SQL thread errors occurred. The `Last_IO_Error` column is now prefixed with the timestamp for the most recent I/O error, and `Last_SQL_Error` shows the timestamp for the most recent SQL thread error. The timestamp values use the format `YYMMDD HH:MM:SS` in both of these columns. For more information, see `SHOW SLAVE STATUS` Syntax. (Bug #43535, Bug #11752361, Bug #64255, Bug #13726435)

- “Unknown table” error messages that included only the table name now include the database name as well. (Bug #34750, Bug #11747993)

- There is now a `bind_address` system variable containing the value of the `--bind-address` option. This enables the address to be accessed at runtime. (Bug #44355, Bug #11752999)

- A `--bind-address` option has been added to a number of MySQL client programs: `mysql`, `mysqldump`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqlimport`, and `mysqlshow`. This is for use on a computer having multiple network interfaces, and enables you to choose which interface is used to connect to the MySQL server.

  A corresponding change was made to the `mysql_options()` C API function, which now has a `MYSQL_OPT_BIND` option for specifying the interface. The argument is a host name or IP address (specified as a string).

- The Block Nested-Loop (BNL) Join algorithm previously used only for inner joins has been extended and can be employed for outer join operations, including nested outer joins. For more information, see Block Nested-Loop and Batched Key Access Joins.

In conjunction with this work, a new system variable, optimizer_join_cache_level, controls how join buffering is done.

- Several changes were made to optimizer-related system variables:

  - The optimizer_switch system variable has new engine_condition_pushdown and index_condition_pushdown flags to control whether storage engine condition pushdown and index condition pushdown optimizations are used. The engine_condition_pushdown system variable now is deprecated. For information about condition pushdown, see Engine Condition Pushdown Optimization, and Index Condition Pushdown Optimization.

  - The optimizer_switch system variable has new mrr and mrr_cost_based flags to control use of the Multi-Range Read optimization. The optimizer_use_mrr system variable has been removed. For information about Multi-Range Read, see Multi-Range Read Optimization.

  - The join_cache_level system variable has been renamed to optimizer_join_cache_level. This enables a single SHOW VARIABLES LIKE 'optimizer%' statement to show more optimizer-related settings.

- Croatian collations were added for Unicode character sets: utf8_croatian_ci, ucs2_croatian_ci, utf8mb4_croatian_ci, utf16_croatian_ci, and utf32_croatian_ci. Thee collations have tailoring for Croatian letters: Č, Ć, Dž, Đ, Lj, Nj, Š, Ž. They are based on Unicode 4.0.

- Previously, EXPLAIN output for a large union truncated the UNION RESULT row at the end of the list as follows if the string became too large:

  ```
  <union1,2,3,4,...>
  ```

  To make it easier to understand the union boundaries, truncation now occurs in the middle of the string:

  ```
  <union1,2,3,...,9>
  ```

  (Bug #30597, Bug #11747073)

- Changes to replication in MySQL 5.6 make mysqlbinlog output generated by the --base64-output=ALWAYS option unusable. ALWAYS is now an invalid value for this option. If the option is given without a value, the effect is now the same as --base64-output=AUTO rather than --base64-output=ALWAYS.

  References: See also Bug #28760.

- The OpenGIS specification defines functions that test the relationship between two geometry values. MySQL originally implemented these functions such that they used object bounding rectangles and returned the same result as the corresponding MBR-based functions. Corresponding versions are now available that use precise object shapes. These versions are named with an ST_ prefix. For example, Contains() uses object bounding rectangles, whereas ST_Contains() uses object shapes. For more information, see Functions That Test Spatial Relations Between Geometry Objects.

  There are also now ST_ aliases for existing spatial functions that were already exact. For example, ST_IsEmpty() is an alias for IsEmpty()

  In addition, the IsSimple() and ST_Distance() spatial functions are now implemented. (Bug #4249, Bug #11744883)

- The Unicode implementation has been extended to include a utf16le character set, which corresponds to the UTF-16LE encoding of the Unicode character set. This is similar to utf16 (UTF-16) but is little-endian rather than big-endian.

  Two utf16le collations are available:

  - utf16le_general_ci: The default collation, case sensitive (similar to utf16_general_ci).
  - utf16le_bin: Case sensitive, with by-codepoint comparison that provides the same order as utf16_bin.

  There are some limitations on the use of utf16le. With the exception of the item regarding user-defined collations, these are the same as the limitations on ucs2, utf16, and utf32.

  - utf16le cannot be used as a client character set, which means that it also does not work for SET NAMES or SET CHARACTER SET.
  - It is not possible to use LOAD DATA INFILE to load data files that use utf16le.
  - FULLTEXT indexes cannot be created on a column that uses utf16le. However, you can perform IN BOOLEAN MODE searches on the column without an index.
  - The use of ENCRYPT() with utf16le is not recommended because the underlying system call expects a string terminated by a zero byte.
  - It is not possible to create user-defined UCA collations for utf16le because there is no utf16le_unicode_ci collation, which would serve as the basis for such collations.

- TO_BASE64() and FROM_BASE64() functions are now available to perform encoding to and from base-64 strings.

- Support for adding Unicode collations that are based on the Unicode Collation Algorithm (UCA) has been improved:

  - MySQL now recognizes a larger subset of the LDML syntax that is used to write collation descriptions. In many cases, it is possible to download a collation definition from the Unicode Common Locale Data Repository and paste the relevant part (that is, the part between the <rules> and </rules> tags) into the MySQL Index.xml file.
  - Character representation in LDML rules is more flexible. Any character can be written literally, not just basic Latin letters. For collations based on UCA 5.2.0, hexadecimal notation can be used for any character, not just BMP characters.
  - When problems are found while parsing Index.xml, better diagnostics are produced.
  - For collations that require tailoring rules, there is no longer a fixed size limit on the tailoring information.

  For more information, see LDML Syntax Supported in MySQL, and Diagnostics During Index.xml Parsing.

- 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 thread_concurrency system variable.
  - The --language server option. Use the lc_messages_dir and lc_messages system variables instead.
  - The --master-retry-count server option. Use the MASTER_RETRY_COUNT option the CHANGE MASTER TO statement instead.

Bugs Fixed

- **Incompatible Change; Replication:** When determining whether to replicate a CREATE DATABASE, DROP DATABASE, or ALTER DATABASE statement, database-level options now take precedence over any --replicate-wild-do-table options. In other words, when trying to replicate one of these statements, --replicate-wild-do-table options are now checked if and only if there are no database-level options that apply to the statement. (Bug #46110, Bug #11754498)

- **Incompatible Change; Replication:** The behavior of INSERT DELAYED statements when using statement-based replication has changed as follows:

  Previously, when using binlog_format=STATEMENT, a warning was issued in the client when executing INSERT DELAYED; now, no warning is issued in such cases.

  Previously, when using binlog_format=STATEMENT, INSERT DELAYED was logged as INSERT DELAYED; now, it is logged as an INSERT, without the DELAYED option.

  However, when binlog_format=STATEMENT, INSERT DELAYED continues to be executed as INSERT (without the DELAYED option). The behavior of INSERT DELAYED remains unchanged when using binlog_format=ROW: INSERT DELAYED generates no warnings, is executed as INSERT DELAYED, and is logged using the row-based format.

  This change also affects binlog_format=MIXED, because INSERT DELAYED is no longer considered unsafe. Now, when the logging format is MIXED, no switch to row-based logging occurs. This means that the statement is logged as a simple INSERT (that is, without the DELAYED option), using the statement-based logging format. (Bug #54579, Bug #11762035)

  References: See also Bug #56678, Bug #11763907, Bug #57666. This bug was introduced by Bug #39934, Bug #11749859.

- **Incompatible Change:** CREATE TABLE statements (including CREATE TABLE ... LIKE) are now prohibited whenever a LOCK TABLES statement is in effect.

  One consequence of this change is that CREATE TABLE ... LIKE makes the same checks as CREATE TABLE and does not just copy the .frm file. This means that if the current SQL mode is different from the mode in effect when the original table was created, the table definition might be considered invalid for the new mode and the statement will fail. (Bug #42546, Bug #11751609)

- **Incompatible Change:** Starvation of FLUSH TABLES WITH READ LOCK statements occurred when there was a constant load of concurrent DML statements in two or more connections. Deadlock occurred when a connection that had some table open through a HANDLER statement tried to update data through a DML statement while another connection tried to execute FLUSH TABLES WITH READ LOCK concurrently.

  These problems resulted from the global read lock implementation, which was reimplemented with the following consequences:

  - To solve deadlock in event-handling code that was exposed by this patch, the LOCK_event_metadata mutex was replaced with metadata locks on events. As a result, DDL operations on events are now prohibited under LOCK TABLES. This is an incompatible change.

  - The global read lock (FLUSH TABLES WITH READ LOCK) no longer blocks DML and DDL on temporary tables. Before this patch, server behavior was not consistent in this respect: In some cases, DML/DDL statements on temporary tables were blocked; in others, they were not. Since the main use cases for FLUSH TABLES WITH READ LOCK are various forms of backups and temporary tables are not preserved during backups, the server now consistently permits DML/DDL on temporary tables under the global read lock.

  - The set of thread states has changed:

  - `Waiting for global metadata lock` is replaced by `Waiting for global read lock`.
  - Previously, `Waiting for release of readlock` was used to indicate that DML/DDL statements were waiting for release of a read lock and `Waiting to get readlock` was used to indicate that `FLUSH TABLES WITH READ LOCK` was waiting to acquire a global read lock. Now `Waiting for global read lock` is used for both cases.
  - Previously, `Waiting for release of readlock` was used for all statements that caused an explicit or implicit commit to indicate that they were waiting for release of a read lock and `Waiting for all running commits to finish` was used by `FLUSH TABLES WITH READ LOCK`. Now `Waiting for commit lock` is used for both cases.
  - There are two other new states, `Waiting for trigger metadata lock` and `Waiting for event metadata lock`.

  (Bug #57006, Bug #11764195, Bug #54673, Bug #11762116)

- **InnoDB; Replication:** If the master had `innodb_file_per_table=OFF`, `innodb_file_format=Antelope` (and `innodb_strict_mode=OFF`), or both, certain `CREATE TABLE` options, such as `KEY_BLOCK_SIZE`, were ignored. This could permit the master to avoid raising `ER_TOO_BIG_ROWSIZE` errors.

  However, the ignored `CREATE TABLE` options were still written into the binary log, so that, if the slave had `innodb_file_per_table=ON` and `innodb_file_format=Barracuda`, it could encounter an `ER_TOO_BIG_ROWSIZE` error while executing the record from the log, causing the slave SQL thread to abort and replication to fail.

  In the case where the master was running MySQL 5.1 and the slave was MySQL 5.5 (or later), the failure occurred when both master and slave were running with default values for `innodb_file_per_table` and `innodb_file_format`. This could cause problems during upgrades.

  To address this issue, the default values for `innodb_file_per_table` and `innodb_file_format` are reverted to the MySQL 5.1 default values—that is, `OFF` and `Antelope`, respectively. (Bug #56318, Bug #11763590)

- **InnoDB:** If the MySQL Server crashed immediately after creating an `InnoDB` table, the server could quit with a `signal 11` during the subsequent restart. The issue could occur if the server halted after `InnoDB` created the primary index for the table, but before the index definition was recorded in the MySQL metadata. (Bug #57616)

  References: This bug is a regression of Bug #54582.

- **InnoDB:** With binary logging enabled, `InnoDB` could halt during crash recovery with a message referring to a transaction ID of 0. (Bug #54901, Bug #11762323)

- **Replication:** Due to changes made in MySQL 5.5.3, settings made in the `binlog_cache_size` and `max_binlog_cache_size` server system variables affected both the binary log statement cache (also introduced in that version) and the binary log transactional cache (formerly known simply as the binary log cache). This meant that the resources used as a result of setting either or both of these variables were double the amount expected. To rectify this problem, these variables now affect only the transactional cache. The fix for this issue also introduces two new system variables `binlog_stmt_cache_size` and `max_binlog_stmt_cache_size`, which affect only the binary log statement cache.

  In addition, the `Binlog_cache_use` status variable was incremented whenever either cache was used, and `Binlog_cache_disk_use` was incremented whenever the disk space from either cache was used, which caused problems with performance tuning of the statement and transactional caches, because it was not possible to determine which of these was being exceeded when attempting to troubleshoot excessive disk seeks and related problems. This issue is solved by changing the behavior of these two status variables such that they are incremented only in response to usage of the binary log transactional cache, as well as by introducing two new status variables `Binlog_stmt_cache_use` and `Binlog_stmt_cache_disk_use`, which are incremented only by usage of the binary log statement cache.

The behavior of the `max_binlog_cache_size` system variable with regard to active sessions has also been changed to match that of the `binlog_cache_size` system variable: Previously, a change in `max_binlog_cache_size` took effect in existing sessions; now, as with a change in `binlog_cache_size`, a change in `max_binlog_cache_size` takes effect only in sessions begun after the value was changed.

For more information, see System Variables Used with Binary Logging, and Server Status Variables. (Bug #57275, Bug #11764443)

- **Replication:** The `Binlog_cache_use` and `Binlog_cache_disk_use status` variables were incremented twice by a change to a table using a transactional storage engine. (Bug #56343, Bug #11763611)

  References: This bug is a regression of Bug #50038.

- **Replication:** When an error occurred in the generation of the name for a new binary log file, the error was logged but not shown to the user. (Bug #46166)

  References: See also Bug #37148, Bug #11748696, Bug #40611, Bug #11750196, Bug #43929, Bug #51019.

- **Replication:** When `lower_case_table_names` was set to 1 on the slave, but not on the master, names of databases in replicated statements were not converted, causing replication to fail on slaves using case-sensitive file systems. This occurred for both statement-based and row-based replication.

  In addition, when using row-based replication with `lower_case_table_names` set to 1 on the slave only, names of tables were also not converted, also causing replication failure on slaves using case-sensitive file systems. (Bug #37656)

- **Replication:** When `STOP SLAVE` is issued, the slave SQL thread rolls back the current transaction and stops immediately if the transaction updates only tables which use transactional storage engines. Previously, this occurred even when the transaction contained `CREATE TEMPORARY TABLE` statements, `DROP TEMPORARY TABLE` statements, or both, although these statements cannot be rolled back. Because temporary tables persist for the lifetime of a user session (in the case, the replication user), they remain until the slave is stopped or reset. When the transaction is restarted following a subsequent `START SLAVE` statement, the SQL thread aborts with an error that a temporary table to be created (or dropped) already exists (or does not exist, in the latter case).

  Following this fix, if an ongoing transaction contains `CREATE TEMPORARY TABLE` statements, `DROP TEMPORARY TABLE` statements, or both, the SQL thread now waits until the transaction ends, then stops. (Bug #56118, Bug #11763416)

- The range optimizer ignored conditions on inner tables in semi-join `IN` subqueries, causing the optimizer to miss good query execution plans. (Bug #35674, Bug #11748263)

- A server crash or memory overrun could occur with a dependent subquery and joins. (Bug #34799, Bug #11748009)

- Selecting from a view that referenced the same table in the `FROM` clause and an `IN` clause caused a server crash. (Bug #33245)

- The server returned incorrect results for `WHERE ... OR ... GROUP BY` queries against `InnoDB` tables. (Bug #37977, Bug #11749031)

- An incorrectly checked `XOR` subquery optimization resulted in an assertion failure. (Bug #37899, Bug #11748998)

- A query that could use one index to produce the desired ordering and another index for range access with index condition pushdown could cause a server crash. (Bug #37851, Bug #11748981)
- The optimizer could underestimate the memory required for column descriptors during join processing and cause memory corruption or a server crash. (Bug #42744, Bug #11751763)
- `NULL` values were not grouped properly for some joins containing `GROUP BY`. (Bug #45267, Bug #11753766)
- A `HAVING` clause could be lost if an index for `ORDER BY` was available, incorrectly permitting additional rows to be returned. (Bug #45227, Bug #11753730)
- A Valgrind failure occurred in `fn_format` when called from `archive_discover`. (Bug #58205, Bug #11765259)
- Passing a string that was not null-terminated to `UpdateXML()` or `ExtractValue()` caused the server to fail with an assertion. (Bug #57279, Bug #11764447)
- With index condition pushdown enabled, `InnoDB` could crash due to a mismatch between what pushdown code expected to be in a record versus what was actually there. (Bug #36981, Bug #11748647)
- In bootstrap mode, the server could not execute statements longer than 10,000 characters. (Bug #55817, Bug #11763139)
- After setting `collation_connection` to one of the collations for the `ucs2` or `utf16` character sets, it was not possible to change the collation thereafter. (Bug #65000, Bug #13970475)
- `cmake -DBUILD_CONFIG=mysql_release` on Linux previously required `libaio` to be linked in. Now it is possible to specify `-DIGNORE_AIO_CHECK` to build without `libaio`. (Bug #58955, Bug #11765940)
- The server crashed on optimizations that used the `range checked for each record` access method. (Bug #32229, Bug #11747417)
- The server crashed on optimization of queries that compared an indexed `DECIMAL` column with a string value. (Bug #32262, Bug #11747426)
- Compared to MySQL 5.1, the optimizer failed to use join buffering for certain queries, resulting in slower performance for those queries. (Bug #30363, Bug #11747028)
- If the optimizer used a Multi-Range Read access method for index lookups, incorrect results could occur for rows that contained any `BLOB` or `TEXT` data types. (Bug #30622, Bug #11747076)
- For Multi-Range Read scans used to resolve `LIMIT` queries, failure to close the scan caused file descriptor leaks for `MyISAM` tables. (Bug #30221, Bug #11746994)
- Deeply nested subqueries could cause stack overflow or a server crash. (Bug #32680, Bug #11747503)
- `SHOW CREATE DATABASE` did not account for the value of the `lower_case_table_names` system variable. (Bug #21317, Bug #11745926)
- `Contains()` failed for multipolygon geometries. (Bug #32032, Bug #11747370)

Changes in MySQL 5.6.0 (Not released, Milestone 4)

> **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.

Performance Schema Notes

- The Performance Schema now includes instrumentation for table input and output. Instrumented operations include row-level accesses to persistent base tables or temporary tables. Operations that affect rows are fetch, insert, update, and delete. For a view, waits are associated with base tables referenced by the view.

Globally Unique Server IDs

- **Replication:** Globally unique IDs for MySQL servers were implemented. A UUID is now obtained automatically when the MySQL server starts. The server first checks for a UUID written in the auto.cnf file (in the server's data directory), and uses this UUID if found. Otherwise, the server generates a new UUID and saves it to this file (and creates the file if it does not already exist). This UUID is available as the server_uuid system variable.

  MySQL replication masters and slaves know each other's UUIDs. The value of a slave's UUID can be read in the output of SHOW SLAVE HOSTS. After a slave is started using START SLAVE, the value of the master's UUID is available on the slave in the output of SHOW SLAVE STATUS. (Bug #33815, Bug #11747723)

  References: See also Bug #16927, Bug #11745543.

Functionality Added or Changed

- **Partitioning:** It is now possible to exchange a partition of a partitioned table or a subpartition of a subpartitioned table with a nonpartitioned table that otherwise has the same structure using the ALTER TABLE ... EXCHANGE PARTITION statement. This can be used, for example, for importing and exporting partitions.

  For more information and examples, see Exchanging Partitions and Subpartitions with Tables.

- **Replication:** These unused and deprecated items have been removed: the --init-rpl-role and --rpl-recovery-rank options, the rpl_recovery_rank system variable, and the Rpl_status status variable. (Bug #54649, Bug #11762095)

  References: See also Bug #34437, Bug #11747900, Bug #34635, Bug #11747961.

- **Replication:** The SHOW SLAVE STATUS statement now has a Master_Info_File field indicating the location of the master.info file. (Bug #50316, Bug #11758151)

- **Replication:** MySQL now supports delayed replication such that a slave server deliberately lags behind the master by at least a specified amount of time. The default delay is 0 seconds. Use the new MASTER_DELAY option for CHANGE MASTER TO to set the delay to *N* seconds:

  ```
  CHANGE MASTER TO MASTER_DELAY = N;
  ```

  An event received from the master is not executed until at least *N* seconds later than its execution on the master.

  START SLAVE and STOP SLAVE take effect immediately and ignore any delay. RESET SLAVE resets the delay to 0.

  SHOW SLAVE STATUS has three new fields that provide information about the delay:

  - SQL_Delay: The number of seconds that the slave must lag the master.

  - SQL_Remaining_Delay: When Slave_SQL_Running_State is Waiting until MASTER_DELAY seconds after master executed event, this field contains the number of seconds left of the delay. At other times, this field is NULL.

  - Slave_SQL_Running_State: The state of the SQL thread (analogous to Slave_IO_State). The value is identical to the State value of the SQL thread as displayed by SHOW PROCESSLIST.

  When the slave SQL thread is waiting for the delay to elapse before executing an event, SHOW PROCESSLIST displays its State value as Waiting until MASTER_DELAY seconds after master executed event.

  The relay-log.info file now contains the delay value, so the file format has changed. See Slave Status Logs. In particular, the first line of the file now indicates how many lines are in the file. If you downgrade a slave server to a version older than MySQL 5.6, the older server will not read the file correctly. To address this, modify the file in a text editor to delete the initial line containing the number of lines.

  The introduction of delayed replication entails these restrictions:

  - Previously the BINLOG statement could execute all types of events. Now it can execute only format description events and row events.
  - The output from mysqlbinlog --base64-output=ALWAYS cannot be parsed. ALWAYS becomes an invalid value for this option in 5.6.1.

  For additional information, see Delayed Replication. (Bug #28760, Bug #11746794)

- The Romansh locale 'rm_CH' is now a permissible value for the lc_time_names system variable. (Bug #50915, Bug #11758678)
- mysqlbinlog now has a --binlog-row-event-max-size option to enable large row events to be read from binary log files. (Bug #49932)
- mysqldump now has an --add-drop-trigger option which adds a DROP TRIGGER IF EXISTS statement before each dumped trigger definition. (Bug #34325, Bug #11747863)
- A new SQL function, WEIGHT_STRING(), returns the weight string for an input string. The weight string represents the sorting and comparison value of the input string. See String Functions.
- mysqlbinlog now has the capability to back up a binary log in its original binary format. When invoked with the --read-from-remote-server and --raw options, mysqlbinlog connects to a server, requests the log files, and writes output files in the same format as the originals. See Using mysqlbinlog to Back Up Binary Log Files.
- The Unicode character sets now have a *xxx*_german2_ci collation that provides DIN-2 (phone book) ordering (for example, utf8_german2_ci). See Unicode Character Sets.
- In MySQL 5.5, setting optimizer_search_depth to the deprecated value of 63 switched to the algorithm used in MySQL 5.0.0 (and previous versions) for performing searches. The value of 63 is now treated as invalid.
- Unicode collation names now may include a version number to indicate the Unicode Collation Algorithm (UCA) version on which the collation is based. Initial collations thus created use version UCA 5.2.0. For example, utf8_unicode_520_ci is based on UCA 5.2.0. UCA-based Unicode collation names that do not include a version number are based on version 4.0.0.

  LOWER() and UPPER() perform case folding according to the collation of their argument. A character that has uppercase and lowercase versions only in a Unicode version more recent than 4.0.0 will be converted by these functions only if the argument has a collation that uses a recent enough UCA version.

  The LDML rules for creating user-defined collations are extended to permit an optional version attribute in <collation> tags to indicate the UCA version on which the collation is based. If the version attribute is omitted, its default value is 4.0.0. See Adding a UCA Collation to a Unicode Character Set.

- Vietnamese collations were added for the Unicode character sets. Those based on Unicode Collation Algorithm 5.2.0 have names of the form *xxx*_vietnamese_520_ci (for example, utf8_vietnamese_520_ci). Those based on Unicode Collation Algorithm 4.0.0 have names of the form *xxx*_vietnamese_ci (for example, utf8_vietnamese_ci). These collations are the same as the corresponding *xxx*_unicode_520_ci and *xxx*_unicode_ci collations except for precomposed characters which are accented versions of “A”, “D”, “E”, “O”, and “U”. There is no change to ideographic characters derived from Chinese. There are no digraphs.

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #49124)
- **InnoDB:** The server could crash on shutdown, if started with --innodb-use-system-malloc=0. (Bug #55581, Bug #11762927)
- **Replication:** The internal flag indicating whether a user value was signed or unsigned (unsigned_flag) could sometimes change between the time that the user value was recorded for logging purposes and the time that the value was actually written to the binary log, which could lead to inconsistency. Now unsigned_flag is copied when the user variable value is copied, and the copy of unsigned_flag is then used for logging. (Bug #51426, Bug #11759138)

  References: See also Bug #49562, Bug #11757508.
- The BLACKHOLE storage engine failed to load on Solaris and OpenSolaris if DTrace probes had been enabled. (Bug #47748, Bug #11755909)
- Passwords for CREATE USER statements were written to the binary log in plaintext rather than in ciphertext. (Bug #50172)
- Performance Schema code was subject to a buffer overflow. (Bug #53363)
- Some error messages included a literal mysql database name rather than a parameter for the database name. (Bug #46792, Bug #11755079)
- On Windows, an IPv6 connection to the server could not be made using an IPv4 address or host name. (Bug #52381, Bug #11760016)
- In the ER_TABLEACCESS_DENIED_ERROR error message, the command name parameter could be truncated. (Bug #45355, Bug #11753840)
- On Windows, the my_rename() function failed to check whether the source file existed. (Bug #51861, Bug #11759540)
- The embedded server could crash when determining which directories to search for option files. (Bug #55062, Bug #11762465)
- Subquery execution for EXPLAIN could be done incorrectly and raise an assertion. (Bug #52317, Bug #11759957)
- The ref column of EXPLAIN output for subquery lines could be missing information. (Bug #50257, Bug #11758106)
- There was a mixup between GROUP BY and ORDER BY concerning which indexes should be considered or permitted during query optimization. (Bug #52081, Bug #11759746)
- Searches for data on a partial index for a column using the utf8 character set would fail. (Bug #24858)
- To forestall the occurrence of possible relocation errors in the future, libmysys, libmystrings, and libdbug have been changed from normal libraries to “noinst” libtool helper libraries, and are no longer installed as separate libraries. (Bug #29791, Bug #11746931)
- On Windows, mysqlslap crashed for attempts to connect using shared memory. (Bug #31173, Bug #11747181, Bug #59107, Bug #11766072)

- For queries with `GROUP BY`, `FORCE INDEX` was not ignored as it should have been when it would result in a more expensive query execution plan. (Bug #18144, Bug #11745649)

- A suboptimal query execution plan could be chosen when there were several possible `range` and `ref` accesses. Now preference is given to the keys that match the most parts and choosing the best one among them. (Bug #26106, Bug #11746406)