MySQL 5.5 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 5.5, up through MySQL 5.5.37. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 5.5 documentation, see the MySQL 5.5 Reference Manual, which includes an overview of features added in MySQL 5.5 (What Is New in MySQL 5.5), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.1 to MySQL 5.5 (Upgrading from MySQL 5.1 to 5.5).

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.

Previously, MySQL development proceeded by including a large set of features and moving them over many versions within a release series through several stages of maturity (Alpha, Beta, and so forth). This development model had a disadvantage in that problems with only part of the code could hinder timely release of the whole.

MySQL development now uses a milestone model. The move to this model provides for more frequent milestone releases, with each milestone proceeding through a small number of releases having a focus on a specific subset of thoroughly tested features. Following the releases for one milestone, development proceeds with the next milestone; that is, another small number of releases that focuses on the next small set of features, also thoroughly tested.

MySQL 5.5.0-m2 is the first release for Milestone 2. The new features of this milestone may be considered to be initially of beta quality. For subsequent Milestone 2 releases, we plan to use increasing version numbers (5.5.1 and higher) while continuing to employ the “-m2” suffix. For Milestone 3, we plan to change the suffix to “-m3”. Version designators with “-alpha” or “-beta” suffixes are no used.

You may notice that the MySQL 5.5.0 release is designated as Milestone 2 rather than Milestone 1. This is because MySQL 5.4 was actually designated as Milestone 1, although we had not yet begun referring to milestone numbers as part of version numbers at the time.

Document generated on: 2014-01-23 (revision: 2637)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 5.5, up through MySQL 5.5.37.

Legal Notices

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 5.5.37 (Not yet released)

Version 5.5.37 has no changelog entries.

Changes in MySQL 5.5.36 (Not yet released)

Version 5.5.36 has no changelog entries.

Changes in MySQL 5.5.35 (2013-12-03)

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

- A new `CMake` option, `WITH_ASAN`, permits enabling address sanitization for compilers that support it. (Bug #17435338)
- Attempts to use the `thread_concurrency` system variable (which has an effect only for Solaris 8 and earlier) now indicate that it has no effect when that is the case. (Bug #67944, Bug #16032946)

Bugs Fixed

- **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:** `CHECK TABLE` would ignore the `QUICK` option. (Bug #17513737)
- **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:** 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:** 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:** 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:** 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:** 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:** The value of `LAST_INSERT_ID()` was not correctly replicated when filtering rules were used on the slave. (Bug #17234370, Bug #69861)
- Enabling Index Merge optimizer switches and setting a small `sort_buffer_size` value could lead to a server exit. (Bug #17617945)
- Some license and documentation files were missing from Windows MSI packages. (Bug #17584523)
- 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)
- `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)
- 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)
- The usual failed-login attempt accounting was not applied to failed `COM_CHANGE_USER` commands. (Bug #16241992, Bug #17357535)
- 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)
- The `my_b_vprintf()` function could produce incorrect results for long integers on 64-bit systems. (Bug #67386, Bug #16978278)
- `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)
- 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)
- 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 5.5.34 (2013-09-20)

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 5.5 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.

Bugs Fixed

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

- 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.

- 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)

- Within a stored procedure, repeated execution of a prepared `CREATE TABLE` statement for a table with partitions could cause a server exit. (Bug #16614004)

- 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)

- 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)

- 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)

- `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)

- 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 5.5.33 (2013-07-31)

A known limitation 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.

Functionality Added or Changed

- 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)

- `comp_err` now checks to make sure that new errors are not being added to MySQL 5.1 or 5.5 because the set of errors for these series is frozen. (Bug #16807394)

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:** 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)

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

- **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:** Some expressions employing variables were not handled correctly by `LOAD DATA`. (Bug #16753869)

- **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)

- Installation of Solaris PKG packages could fail to execute `mysql_install_db` because it was invoked with the `--random-passwords` option (which does not exist until MySQL 5.6). (Bug #17160741)

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

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

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

- Upgrading from community SLES RPM packages to commercial packages for the same MySQL version failed with conflict errors. (Bug #16545296)

- 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)
- Out-of-bounds reads could occur within `filename_to_tablename()`. (Bug #14834378)
- 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)
- With the thread pool plugin in use, normal connection termination caused the `Aborted_clients` status variable to be incremented. (Bug #14081240)
- 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)
- 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)
- `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)
- 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)
- RPM source packages did not list `libaio-devel` as a dependency, causing builds to fail. (Bug #69158, Bug #16785036)
- Comparison of a `DATETIME` value and a string did not work correctly for the `utf8_unicode_ci` collation. (Bug #68795, Bug #16567381)

Changes in MySQL 5.5.32 (2013-06-03)

A known limitation 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.

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

- **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:** When calling the `lock_rec_block_validate()` function after releasing the kernel mutex, there is a chance the lock might be invalid and result in a Valgrind error due to an invalid read on `lock->index`. This fix copies the `lock->index` when the kernel mutex is being held and passes the `lock->index` to `lock_rec_block_validate()`. (Bug #17022398, Bug #69413, Bug #16268289, Bug #68244)

- **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:** The `page_zip_available` function would count some fields twice. (Bug #16463505)

- **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:** 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:** 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:** 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:** For `UPDATE` statements in which an error occurred, it was possible for a temporary file opened during the update not to be closed. (Bug #15978766)
- **InnoDB:** An overflow would occur for `innodb_row_lock_time_max` and `innodb_row_lock_current_waits`. This fix modifies code logic in `storage/innobase/srv/srv0srv.c`. (Bug #16005310)
- **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:** 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:** 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:** 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:** 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:** 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)
- The WKB reader for spatial operations could fail and cause a server exit. (Bug #16451878)
- `EXPORT_SET()` or `MAKE_SET()` with many `COUNT(*)` arguments could cause a server exit. (Bug #16359402)
- 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)
- `thread_pool_high_priority_connection` could not be set at server startup. (Bug #16310373)

- 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)
- A `GROUP_CONCAT()` invocation containing subquery having an outer reference caused the server to exit. (Bug #16347343)
- For debug builds, `GROUP_CONCAT(... ORDER BY)` within an `ORDER BY` clause could cause a server exit. (Bug #16347426)
- 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)
- A prepared statement that used `GROUP_CONCAT()` and an `ORDER BY` clause that named multiple columns could cause the server to exit. (Bug #16075310)
- `ORDER BY MATCH ... AGAINST` could cause a server exit. (Bug #16073689)
- The `mysql.server` script exited with an error if the `status` command was executed with multiple servers running. (Bug #15852074)
- A query with a union and a join could crash the parser. (Bug #14786792, Bug #16076289)
- Installation using Solaris packages ran `mysql_install_db` during upgrade operations (this should occur only for new installations). (Bug #14747671, Bug #16534721)
- When processing row-based-replication events in the old binary log format from prior to MySQL 5.1 GA builds, `mysqlbinlog` could result in out-of-bounds heap buffer reads and undefined behaviour. (Bug #14771299)
- The `mysql` client allocated but did not free a string after reading each line in interactive mode, resulting in a memory leak. (Bug #14685362)
- `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)
- On 64-bit Mac OS X systems, `CMake` used `x86` rather than `x86_64` when determining the machine type. (Bug #58462, Bug #11765489)
- 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)
- The `url` columns in the `mysql` datatbase help tables were too short to hold some of the URLs in the help content. For new installations, these columns are now created as type `TEXT` to accommodate longer URLs.

  For upgrades, `mysql_upgrade` does *not* update the columns. Modify them manually using these statements:

```
ALTER TABLE mysql.help_category MODIFY url TEXT NOT NULL;
```

```
ALTER TABLE mysql.help_topic MODIFY url TEXT NOT NULL;
```

(Bug #61520, Bug #12671635)

- 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)
- 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)

- Using range access with an index prefix could produce incorrect results. (Bug #68750, Bug #16540042)
- `MD5()` code did not properly initialize one of its data structures. (Bug #68909, Bug #16626742)
- When specified in an option file, the `plugin-dir` client option was ignored. (Bug #68800, Bug #16680313)
- 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)
- MySQL Configuration Wizard did not anticipate existing files from a previous MySQL install operation, causing it to fail starting the MySQL service. (Workaround: Manually delete MySQL data in the `ProgramData` folder.) (Bug #62106, Bug #16777237)

Changes in MySQL 5.5.31 (2013-04-18)

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 5.5.31, 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 5.5.31, it is possible to install using `yum`:

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

  For upgrades to MySQL 5.5.31, 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 5.5.*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

- MySQL no longer uses the default OpenSSL compression. (Bug #16235681)

Bugs Fixed

- **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)

- **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 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:** For `InnoDB` tables, if a `PRIMARY KEY` on a `VARCHAR` column (or prefix) was empty, index page compression could fail. (Bug #16400920)
- **InnoDB:** Crash recovery would fail with a `!recv_no_log_write` assertion when reading a page. (Bug #16405422)
- **InnoDB:** For debug builds, `InnoDB` status exporting was subject to a race condition that could cause a server exit. (Bug #16292043)
- **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:** InnoDB now aborts execution on Windows by calling the `abort()` function directly, as it does on other platforms. (Bug #16263506)
- **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:** Arithmetic underflow during page compression for `CREATE TABLE` on an `InnoDB` table could cause a server exit. (Bug #16089381)

- **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:** 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:** 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:** The length of internally generated foreign key names was not checked. If internally generated foreign key names were over the 64 character limit, this resulted in invalid DDL from SHOW CREATE TABLE. This fix checks the length of internally generated foreign key names and reports an error message if the limit is exceeded. (Bug #44541, Bug #11753153)

- **Partitioning:** A query on a table partitioned by range and using `TO_DAYS()` as a partitioing function always included the first partition of the table when pruning. This happened regardless of the range employed in the `BETWEEN` clause of such a query. (Bug #15843818, Bug #49754)

- **Partitioning:** Execution of `ALTER TABLE ... DROP PARTITION` against a view caused the server to crash, rather than fail with an error as expected. (Bug #14653504)

- **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:** 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:** 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.

- `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` could report incorrect memory-allocation values when the correct values exceeded 4GB. (Bug #16414644)
- 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)
- 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 upgrade operations, RPM packages produced unnecessary errors about being unable to access `.err` files. (Bug #16235828)
- Invocation of the range optimizer for a `NULL` select caused the server to exit. (Bug #16192219)
- 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)
- 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 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)
- Issuing a `PREPARE` statement using certain combinations of stored functions and user variables caused the server to exit. (Bug #16056537)
- 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)
- Contention in the thread pool during kill processing could lead to a Valgrind panic. (Bug #15921866)
- 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)
- 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)
- `SHOW COLUMNS` on a view defined as a `UNION` of `Geometry` columns could cause the server to exit. (Bug #14362617)
- A `LIKE` pattern with too many `'%'` wildcards could cause a segmentation fault. (Bug #14303860)

- `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)
- Subqueries with `OUTER JOIN` could return incorrect results if the subquery referred to a column from another `SELECT`. (Bug #13068506)
- On Microsoft Windows, the MSI package would now allow a license switch (community to or from the commercial edition) when the switched MySQL Server versions were identical. (Bug #13071597)
- `mysql_install_db` did not escape `'_'` in the host name for statements written to the grant tables. (Bug #11746817)
- 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 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)
- Setting `max_connections` to a value less than the current number of open connections caused the server to exit. (Bug #44100, Bug #11752803)
- Incorrect metadata could be produced for columns returned from some views. (Bug #65379, Bug #14096619)
- For debug builds, some queries with `SELECT ... FROM DUAL` nested subqueries raised an assertion. (Bug #60305, Bug #11827369)
- 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)
- `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)

Changes in MySQL 5.5.30 (2013-02-05)

Known limitations of this release:

On Microsoft Windows, when using the MySQL Installer to install MySQL Server 5.5.30 on a host with an existing MySQL Server of a different version (such as 5.6.10), 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 5.5.30.

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)

- **InnoDB:** The `innodb_print_all_deadlocks` configuration option from MySQL 5.6 was backported to MySQL 5.5. This option records each deadlock condition in the MySQL error log, allowing easier troubleshooting if frequent deadlocks point to application coding issues. (Bug #14515889)

- In RPM packages built for Unbreakable Linux Network, `libmysqld.so` now has a version number. (Bug #15972480)

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)

- **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:** 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:** 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:** The server could halt with an assertion error when creating an index on a column prefix for a column using a multi-byte 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:** 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:** A regression introduced by the fix for Bug#14100254 would result in a "!BPAGE->FILE_PAGE_WAS_FREED" assertion. (Bug #14676249)
- **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:** 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:** 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:** 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:** 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)

- **Partitioning:** Concurrent `ALTER TABLE ... REBUILD PARTITION` operations could interfere with one another, even when not running against the same table, because they both used global memory for storage. Now each partition rebuild operation stores intermediate data in memory that is local to that process. (Bug #14589559, Bug #66645)

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

- **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:** 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)

- Directory name manipulation could result in stack overflow on Mac OS X and Windows. (Bug #16066243)

- Joins of exactly 32 tables and containing a `HAVING` clause returned an empty result. (Bug #15972635)

- A buffer-handling problem in yaSSL was fixed. (Bug #15965288)

- A `mysys` library string-formatting routine could mishandle width specifiers. (Bug #15960005)

- In certain cases, `UpdateXML()` could return `NULL` incorrectly. (Bug #15948580)

  References: See also Bug #13007062.

- Metadata locking and table definition cache routines did not always check length of names passed to them. (Bug #15954872)
- `XA START` had a race condition that could cause a server crash. (Bug #14729757)
- Enabling the query cache during high client contention could cause the server to exit. (Bug #14727815)
- 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.
- The server sometimes failed to respect `MAX_CONNECTIONS_PER_HOUR` limits on user connections. (Bug #14627287)
- Output generated with `mysqldump --routines` could produce syntax errors when reloaded. (Bug #14463669)
- With the thread pool plugin installed, a workload consisting of concurrent `KILL` statements and ping queries caused the server to exit. (Bug #14458232, Bug #14458002)
- `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)
- Passing an unknown time zone specification to `CONVERT_TZ()` resulted in a memory leak. (Bug #12347040)
- For dumps of the `mysql` database, `mysqldump` skips the `event` table unless the `--events` option is given. `mysqldump` now prints a warning if invoked without `--events` that the `mysql.event` table is not dumped without that option. (Bug #55587, Bug #11762933)
- For `MEMORY` tables with `HASH` indexes, `DELETE` sometimes failed to delete all applicable rows. (Bug #51763, Bug #11759445)
- 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)
- `mysqld_safe` used the nonportable `-e` test construct. (Bug #67976, Bug #16046140)
- Configuring the server with `performance_schema_events_waits_history_size=0` and `performance_schema_events_waits_history_long_size=0` could cause a Performance Schema segmentation fault. (Bug #68008, Bug #16060864)
- `DECIMAL` multiplication operations could produce significant inaccuracy. (Bug #45860, Bug #11754279)
- For subqueries executing using a `filesort`, the optimizer could produce an incorrect result containing wrong rows. (Bug #66845, Bug #14636211)

  References: See also Bug #12667154.
- `UNION` type conversion could incorrectly turn unsigned values into signed values. (Bug #49003, Bug #11757005)
- 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.

Changes in MySQL 5.5.29 (2012-12-21)

Beginning with MySQL 5.5.29, Oracle no longer provides binaries for Mac OS X 10.5, Debian 5, RHEL/OL 4, FreeBSD 7, Windows XP, or Windows 2003.

Functionality Added or Changed

- The `SHOW AUTHORS` and `SHOW CONTRIBUTORS` statements are now deprecated in MySQL 5.5 and have been removed in MySQL 5.6.

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

- **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:** 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)

- 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)

- Attempting to create an auto-increment column in an `InnoDB` table with a `NULL` type attribute could cause a serious error. (Bug #14758479)
- 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)
- Repeated execution of a query containing a subquery that used `MAX()` could result in increasing memory consumption. (Bug #14683676)
- `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)
- `SHOW PROFILE` could be used to cause excessive server memory consumption. (Bug #14629232)
- 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)
- Improper memory cleanup could cause the server to exit. (Bug #14536113)
- 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)
- Access to `INFORMATION_SCHEMA` tables through a view could leak memory. (Bug #13734987)
- A memory leak could occur for queries containing a subquery that used `GROUP BY` on an outer column. (Bug #13724099)
- 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)
- `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)
- Improper memory cleanup could cause the server to exit. (Bug #13340270)
- A memory leak occurred due to failure to clean up after `QUICK_INDEX_MERGE_SELECT/Unique`. (Bug #12694872, Bug #14542543)
- 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.

- 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)

- 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)

- In debug builds, an `InnoDB` assertion was overly aggressive about prohibiting an open range. (Bug #66513, Bug #14547952)

- 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)

- `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)

- RHEL RPM packages had a conflict between `mysql-libs` and `mysql-shared`. (Bug #67965, Bug #16041010)

Changes in MySQL 5.5.28 (2012-09-28)

Audit Log Plugin Notes

- MySQL Enterprise Edition now includes MySQL Enterprise Audit, implemented using a server plugin named `audit_log`. MySQL Enterprise Audit uses the open MySQL Audit API to enable standard, policy-based monitoring and logging of connection and query activity executed on specific MySQL servers. Designed to meet the Oracle audit specification, MySQL Enterprise Audit provides an out of box, easy to use auditing and compliance solution for applications that are governed by both internal and external regulatory guidelines.

  When installed, the audit plugin enables MySQL Server to produce a log file containing an audit record of server activity. The log contents include when clients connect and disconnect, and what actions they perform while connected, such as which databases and tables they access.

  For more information, see MySQL Enterprise Audit Log Plugin.

Functionality Added or Changed

- The internal interface of the Thread Pool plugin has changed. Old versions of the plugin will work with current versions of the server, but versions of the server older than 5.5.28 will not work with current versions of the plugin.

Bugs Fixed

- **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:** 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:** 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:** Certain `INFORMATION_SCHEMA` tables originally introduced in MySQL 5.6 are now also available in MySQL 5.5 and MySQL 5.1: `INNODB_BUFFER_PAGE`, `INNODB_BUFFER_PAGE_LRU`, and `INNODB_BUFFER_POOL_STATS`. (Bug #13113026)

- **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)

- **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)

- **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:** In master-master replication with `--log-slave-updates` enabled, setting a user variable and then performing inserts using this variable caused the `Exec_master_log_position` column in the output of `SHOW SLAVE STATUS` not to be updated. (Bug #13596613)
- The RPM spec file now also runs the test suite on the new binaries, before packaging them. (Bug #14318456)
- 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)
- The Thread Pool plugin did not respect the `wait_timeout` timeout for client sessions. (Bug #13699303)
- `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)
- 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)
- 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)
- `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)

Changes in MySQL 5.5.27 (2012-08-02)

Functionality Added or Changed

- **Important Change:** The `YEAR(2)` data type is now deprecated because it is problematic. Support for `YEAR(2)` will be removed in a future release of MySQL. For more information, see `YEAR(2)` Limitations and Migrating to `YEAR(4)`.
- The `mysql_clear_password` cleartext client-side authentication plugin is intended for authentication schemes that require the server to receive the password as entered on the client side, without hashing. Because the password is sent in the clear, this plugin should be used within the context of a secure connection, such as an SSL connection, to avoid exposing the password over the network. To make inadvertent use of this plugin less likely, it is now required that clients explicitly enable it. This can be done several ways:
  - Set the `LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN` environment variable to a value that begins with `1`, `Y`, or `y`. This enables the plugin for all client connections.

- The `mysql`, `mysqladmin`, and `mysqlslap` client programs support an `--enable-cleartext-plugin` option that enables the plugin on a per-invocation basis.
- The `mysql_options()` C API function supports a `MYSQL_ENABLE_CLEARTEXT_PLUGIN` option that enables the plugin on a per-connection basis. Also, any program that uses `libmysqlclient` and reads option files can enable the plugin by including an `enable-cleartext-plugin` option in an option group read by the client library.

Bugs Fixed

- **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:** 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:** 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:** 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:** 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)
- **Partitioning:** Insertion of an out-of-range value into a partitioned table was not handled correctly in all cases. This is a regression that first appeared in MySQL 5.5.23. (Bug #14005441, Bug #65587)
- **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 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.

- **Replication:** After upgrading a replication slave to MySQL 5.5.18 or later, enabling the query cache eventually caused the slave to fail. (Bug #64624, Bug #14005409)
- The server did not build with `gcc` 4.7. (Bug #14238406)
- 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)
- Certain arguments to `RPAD()` could lead to “uninitialized variable” warnings. (Bug #14039955)
- `COUNT(DISTINCT(SELECT 1))` could be evaluated incorrectly if the optimizer used Loose Index Scan. (Bug #13444084)

  References: See also Bug #13813126.
- The presence of a file named `.empty` in the `test` database prevented that database from being dropped. (Bug #12845091)
- 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. (Bug #12667154)
- `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)
- MySQL builds failed with CMake 2.8.8. (Bug #65050, Bug #14017376)
- `COUNT(DISTINCT(IF ...))` could be evaluated incorrectly if the optimizer used Loose Index Scan. (Bug #64445, Bug #13813126)

  References: See also Bug #13444084.
- File access by the `ARCHIVE` storage engine was not instrumented and thus not shown in Performance Schema tables. (Bug #63340, Bug #13417440)
- `mysqlbinlog` exited with no error code if file write errors occurred. (Bug #55289, Bug #11762667)
- Using `CONCAT()` to construct a pattern for a `LIKE` pattern match could result in memory corrupting and match failure. (Bug #59140, Bug #11766101)
- yaSSL rejected valid SSL certificates that OpenSSL accepts. (Bug #54348, Bug #11761822)
- Sessions could end up deadlocked when executing a combination of `SELECT`, `DROP TABLE`, `KILL`, and `SHOW ENGINE INNODB STATUS`. (Bug #60682, Bug #12636001)
- If an account had a nonzero `MAX_USER_CONNECTIONS` value, that value was not always respected. (Bug #65104, Bug #14003080)

Changes in MySQL 5.5.26 (Not released)

MySQL 5.5.26 was not released, and has been replaced by MySQL 5.5.27. All changes that should have appeared in MySQL 5.5.26 appear instead in MySQL 5.5.27.

Users of MySQL 5.5.25a and previous MySQL 5.5 releases should upgrade to MySQL 5.5.27.

For a complete list of fixes, improvements, and other changes made in MySQL 5.5.27, see Changes in MySQL 5.5.27 (2012-08-02).

Changes in MySQL 5.5.25a (2012-07-05)

Note

Due to MSI restrictions, the MSI packages of MySQL 5.5.25a will treat the version as 5.5.26 internally; for example, as displayed by the Installation Wizard. MySQL itself reports the version as 5.5.25a; for example, if you check the value of the `VERSION()` SQL function or the `version` system variable.

Bugs Fixed

- 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)

Changes in MySQL 5.5.25 (2012-05-30)

Note

MySQL 5.5.25 is superseded by MySQL 5.5.25a due to a regression bug that can cause excessive disk usage (for details, see Bug #65745). Current users of 5.5.25: Upgrade to 5.5.25a. Users contemplating an upgrade to 5.5.25: Upgrade to 5.5.25a instead.

Functionality Added or Changed

- **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.)
- The `--safe-mode` server option now is deprecated and will be removed in MySQL 5.6.

Bugs Fixed

- **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)
- **InnoDB:** The `Innodb_buffer_pool_pages_flushed` status variable was incorrectly set to twice the value it should be. Its value should never exceed the value of `Innodb_pages_written`. (Bug #14000361, Bug #65030)
- **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:** 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:** 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)

- **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:** 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.

- For queries with `ORDER BY COUNT(*)` and `LIMIT`, the optimizer could choose an execution plan that produced incorrect results. (Bug #12713907)

- `SHOW TABLES` was very slow unless the required information was already in the disk cache. (Bug #60961, Bug #12427262)

- 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)

Changes in MySQL 5.5.24 (2012-05-07)

Functionality Added or Changed

- **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)

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #64884)
- **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:** 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)
- 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)
- 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)
- `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.
- On Windows, `mysqlslap` crashed for attempts to connect using shared memory. (Bug #31173, Bug #11747181, Bug #59107, Bug #11766072)

Changes in MySQL 5.5.23 (2012-04-12)

Functionality Added or Changed

- 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)

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #59533)

- **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:** 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)

- **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)

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

- **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:** 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:** When shutting down the MySQL server, the cleanup operations of the `InnoDB` shutdown could take a long time with no output, making the server appear to be hung.

  ```
  [Note] mysqld: Normal shutdown
  InnoDB: Starting shutdown...
  ```

```
InnoDB: Shutdown completed; log sequence number ...
```

Now additional progress messages are displayed between the “starting” and “completed” messages:

```
InnoDB: Waiting for srv_monitor_thread (srv_lock_timeout_thread/ srv_error_monitor_thread) to exit
InnoDB: Waiting for %lu active transactions to exit
InnoDB: Waiting for master thread (worker threads) to be suspended
InnoDB: Pending checkpoint_writes: %lu
InnoDB: Pending log flush writes: %lu
InnoDB: Waiting for %lu buffer page I/Os to complete
InnoDB: Waiting for dirty buffer pages to be flushed
```

For both fast shutdown and slow shutdown, a progress messages is printed every 60 seconds:

```
InnoDB: Waiting for %lu tables to be dropped
```

During a slow shutdown, two additional messages are printed if certain phases take longer than normal:

```
InnoDB: Waiting for %lu undo logs to be purged
InnoDB: number of pages just purged: %lu

InnoDB: Waiting for change buffer merge to complete\n
InnoDB: number of bytes of change buffer just merged:  %lu
```

(Bug #11755873, Bug #47707)

- **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:** 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.

- An infinite thread loop could develop within Performance Schema, causing the server to become unresponsive. (Bug #13898343)

- 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)

- Mishandling of `NO_BACKSLASH_ESCAPES` SQL mode within stored procedures on slave servers could cause replication failures. (Bug #12601974)

- `SAVEPOINT` statements were incorrectly disallowed within XA transactions. (Bug #64374, Bug #13737343)

  References: See also Bug #11766752.

- 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)
- On Mac OS X 10.5, the MySQL Preference Pane did not run on Intel-based systems. (Bug #60712, Bug #13788147)

Changes in MySQL 5.5.22 (2012-03-21)

Functionality Added or Changed

- **InnoDB:** A deprecation warning is now issued when `--ignore-builtin-innodb` is used. (Bug #13586262)
- yaSSL was upgraded from version 1.7.2 to 2.2.0. (Bug #13706828)

  References: See also Bug #13713205.

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #63775)
- **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)
- **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:** 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:** 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)

- **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)

- 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)

- `myisam_sort_buffer_size` could not be set larger than 4GB on 64-bit systems. (Bug #45702, Bug #11754145)

- 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)

- 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.

Changes in MySQL 5.5.21 (2012-02-17)

Functionality Added or Changed

- 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)

- 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)

Bugs Fixed

- **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:** An earlier change (in MySQL 5.1.59 and 5.5.16) was found to modify date-handling behavior in General Availability-status series (MySQL 5.1 and 5.5). This change has been reverted.

The change was that several functions became more strict when passed a `DATE()` function value as their argument, thus they rejected incomplete dates with a day part of zero. These functions were affected: `CONVERT_TZ()`, `DATE_ADD()`, `DATE_SUB()`, `DAYOFYEAR()`, `LAST_DAY()`, `TIMESTAMPDIFF()`, `TO_DAYS()`, `TO_SECONDS()`, `WEEK()`, `WEEKDAY()`, `WEEKOFYEAR()`, `YEARWEEK()`. The previous behavior has been restored. (Bug #13458237)

- **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:** 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:** 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:** 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:** When copying a partitioned `InnoDB` table from a Linux system to a Windows system, you could encounter this error:

```
101115 14:19:53 [ERROR] Table .\test\d has no primary key in InnoDB data
dictionary, but has one in MySQL!
```

  Normally, the solution to copy `InnoDB` tables from Linux to Windows is to create the tables on Linux with the `lower_case_table_names` option enabled. Partitioned tables, with `#P#` appended to the filename, were not covered by that solution. (Bug #11765438, Bug #58406)

- **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:** On Windows replication slave hosts, `STOP SLAVE` took an excessive length of time to complete when the master was down. (Bug #11752315, Bug #43460)
- 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)
- 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)
- When the optimizer performed conversion of `DECIMAL` values while evaluating range conditions, it could produce incorrect results. (Bug #13453382)
- 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)
- 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.
- When used with the `--xml` option, `mysqldump --routines` failed to dump any stored routines, triggers, or events. (Bug #11760384, Bug #52792)
- 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 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)

- Under some circumstances, the result of `SUBSTRING_INDEX()` incorrectly depended on the contents of the previous row. (Bug #42404, Bug #11751514)
- 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)

Changes in MySQL 5.5.20 (2012-01-10)

Functionality Added or Changed

- 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)

Bugs Fixed

- **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)

- **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)

- **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:** 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:** 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)

- 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)
- `LOAD INDEX INTO CACHE` could cause a server exit if the index cache was too small. (Bug #12361113)
- Enabling `myisam_use_mmap` could cause the server to crash. (Bug #48726, Bug #11756764)
- Concurrent access to `ARCHIVE` tables could cause corruption. (Bug #42784, Bug #11751793)
- The `handle_segfault()` signal-handler code in `mysqld` could itself crash due to calling unsafe functions. (Bug #54082, Bug #11761576)
- Locale information for `FORMAT()` function instances was lost in view definitions. (Bug #63020, Bug #13344643)

Changes in MySQL 5.5.19 (2011-12-08)

Functionality Added or Changed

- **Replication:** Previously, replication slaves could connect to the master server through master accounts that use nonnative authentication, except Windows native authentication. This is now also true for Windows native authentication.
- 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)

Bugs Fixed

- **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)

- **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)
- **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)

- Rounding `DBL_MAX` returned `DBL_MAX`, not 'inf'. (Bug #13261955)
- 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)
- 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)
- `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)

Changes in MySQL 5.5.18 (2011-11-16)

Beginning with MySQL 5.5.18, Debian packages for MySQL are available.

Functionality Added or Changed

- 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)

Bugs Fixed

- **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)

- **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)

- **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:** 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)

- **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.

- 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)

- `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)

- 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)

- `decimal_round()` could cause a server exit when processing long numeric strings. (Bug #12563865)

- For `FEDERATED` tables, loss of connection to the remote table during some insert operations could cause a server crash. (Bug #34660, Bug #11747970)

- 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)

- `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)

Changes in MySQL 5.5.17 (2011-10-19)

Functionality Added or Changed

- **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 (except Windows native authentication) if the required client-side plugin is installed on the slave side in the directory named by the slave `plugin_dir` system variable. The exception for Windows is lifted in MySQL 5.5.19. (Bug #12897501)
- `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)
- The `make_win_bin_dist` script is no longer used and has been removed from MySQL distributions and the manual. (Bug #58241)

Bugs Fixed

- **Performance; InnoDB:** This fix improves the performance of instrumentation code for `InnoDB` buffer pool operations. (Bug #12950803, Bug #62294)
- **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:** 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)
- Internal conversion of zero to binary and back could yield a result with incorrect precision. (Bug #12911710)
- Valgrind warnings generated by `filesort` operations were fixed. (Bug #12856915)
- `mysqld_safe` did not properly check for an already running instance of `mysqld`. (Bug #11878394)
- With Valgrind enabled, `InnoDB` semaphore wait timeouts were too low and could expire. (Bug #11765460)
- 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)
- An assertion designed to detect zero-length sort keys also was raised when the entire key set fit in memory. (Bug #58200, Bug #11765254)
- A linking problem prevented the `FEDERATED` storage engine plugin from loading. (Bug #40942, Bug #11750417)

Changes in MySQL 5.5.16 (2011-09-15)

External Authentication

- Commercial distributions of MySQL now include two plugins that enable MySQL Server to use external authentication methods to authenticate MySQL users:

  - PAM (Pluggable Authentication Modules) enables a system to access various kinds of authentication methods through a standard interface. A PAM authentication plugin enables MySQL Server to use PAM to authenticate MySQL users.

    The PAM plugin uses the information passed to it by the MySQL server (such as user name, host name, password, and authentication string), plus whatever is available for PAM lookup (such as Unix passwords or an LDAP directory). The plugin checks the user credentials against PAM and returns success or failure.

    The PAM authentication plugin has been tested on Linux and Mac OS X.

    

    Note

    The PAM plugin works with a client-side plugin that simply sends the password to the server in clear text so it can be passed to PAM. This may be a security problem in some configurations, but is necessary to use the server-side PAM library. To avoid problems if there is any possibility that the password would be intercepted, clients should connect to MySQL Server using SSL. See The Cleartext Client-Side Authentication Plugin.

  - Distributions of MySQL for Windows include an authentication plugin that enables MySQL Server to use native Windows services to authenticate client connections. Users who have logged in to Windows can connect from MySQL client programs to the server based on the information in their environment without specifying an additional password.

    The client and server exchange data packets in the authentication handshake. As a result of this exchange, the server creates a security context object that represents the identity of the client in the Windows OS. This identity includes the name of the client account. The Windows authentication plugin uses the identity of the client to check whether it is a given account or a member of a group. By default, negotiation uses Kerberos to authenticate, then NTLM if Kerberos is unavailable.

    The Windows authentication plugin should work on Windows 2000 Professional

  These authentication plugins enable MySQL Server to accept connections from users defined outside the MySQL grant tables. They also support the MySQL proxy-user capability. Each plugin can return to MySQL a user name different from the login user, which means that the plugin can return the MySQL user that defines the privileges the externally authenticated user should have. For example, an external user named `joe` can connect and have the privileges of the MySQL user named `developer`.

  The server-side PAM and Windows authentication plugins are included only in commercial distributions. They are not included in MySQL community distributions. The client-side plugins with which they communicate are included in all distributions, including community distributions. This permits clients from any release to connect to a server that has the server-side plugin loaded.

  For more information about these plugins, see The PAM Authentication Plugin, and The Windows Native Authentication Plugin. For general information about pluggable authentication in MySQL, see Pluggable Authentication. For proxy user information, see Proxy Users.

Thread Pool Plugin Notes

- The default thread-handling model in MySQL Server executes statements using one thread per client connection. As more clients connect to the server and execute statements, overall performance degrades. Commercial distributions of MySQL now include a thread pool plugin that provides an alternative thread-handling model designed to reduce overhead and improve performance. The plugin implements a thread pool that increases server performance by efficiently managing statement execution threads for large numbers of client connections.

  The thread pool addresses several problems of the one thread per connection model:

  - Too many thread stacks make CPU caches almost useless in highly parallel execution workloads. The thread pool promotes thread stack reuse to minimize the CPU cache footprint.
  - With too many threads executing in parallel, context switching overhead is high. This also presents a challenging task to the operating system scheduler. The thread pool controls the number of active threads to keep the parallelism within the MySQL server at a level that it can handle and that is appropriate for the server host on which MySQL is executing.
  - Too many transactions executing in parallel increases resource contention. In `InnoDB`, this increases the time spent holding central mutexes. The thread pool controls when transactions start to ensure that not too many execute in parallel.

  On Windows, the thread pool plugin requires Windows Vista or newer. On Linux, the plugin requires kernel 2.6.9 or newer.

  For more information, see The Thread Pool Plugin.

Functionality Added or Changed

- **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)

- 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.

- 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.

- The thread pool plugin should be loaded at server startup, and not loaded or unloaded at runtime. An error now occurs for attempts to load or unload it with the `INSTALL PLUGIN` or `UNINSTALL PLUGIN` statement.

Bugs Fixed

- **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 “random read-ahead” feature that was removed from the `InnoDB` Plugin is now available again. Because it is only helpful for certain workloads, it is turned off by default. To turn it on, enable the `innodb_random_read_ahead` configuration option. Because this feature can improve performance in some cases and reduce performance in others, before relying on this setting, benchmark both with and without the setting enabled. (Bug #12356373)

- **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.

  However, 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()`. Because this changes date-handling behavior in General Availability-status series (MySQL 5.1 and 5.5), it was reverted in 5.1.62 and 5.5.21. The change is retained in MySQL 5.6.

  References: See also Bug #13458237.

- **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:** 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:** Unused functions were removed from the internal `InnoDB` code related to mini-transactions, to clarify the logic. (Bug #12626794, Bug #61240)

- **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.

- 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)
- A `DBUG_ASSERT` added by Bug #11792200 was overly aggressive in raising assertions. (Bug #12537160)
- `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)
- 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)
- Upgrades using an RPM package recreated the `test` database, which is undesirable when the DBA had removed it. (Bug #45415, Bug #11753896)
- The embedded server crashed when `argc = 0`. (Bug #57931, Bug #12561297)
- `(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)

Changes in MySQL 5.5.15 (2011-07-28)

Functionality Added or Changed

- The undocumented `--all` option for `perror` is deprecated and will be removed in MySQL 5.6.

Bugs Fixed

- **Performance; InnoDB:** The `DROP TABLE` command for an `InnoDB` table could be very slow, in a configuration with a combination of table compression, partitioning, and a large buffer pool. (Bug #12635227, Bug #61188)
- **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)
- **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:** The `UNIX_TIMESTAMP()` function was not treated as a monotonic function for purposes of partition pruning. (Bug #11746819, Bug #28928)
- **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.

- Compiling the server with maintainer mode enabled failed for `gcc` 4.6 or higher. (Bug #12727287)
- The option-parsing code for empty strings leaked memory. (Bug #12589928)
- `mysql_list_fields()` returned incorrect character set information for character columns of views. (Bug #12337762)
- 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.

- In debug builds, `Field_new_decimal::store_value()` was subject to buffer overflows. (Bug #55436, Bug #11762799)
- `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)

- For `MyISAM` tables, attempts to insert incorrect data into an indexed `GEOMETRY` column could result in table corruption. (Bug #57323, Bug #11764487)
- `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)

- For unknown users, the native password plugin reported incorrectly that no password had been specified even when it had. (Bug #59792, Bug #11766641)

Changes in MySQL 5.5.14 (2011-07-05)

Functionality Added or Changed

- **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.

- **InnoDB:** `InnoDB` now permits concurrent reads while creating a secondary index. (Bug #11853126)

  References: See also Bug #11751388, Bug #11784056, Bug #11815600.

- `CMake` configuration support on Linux now provides a boolean `ENABLE_GCOV` option to control whether to include support for `gcov`. (Bug #12549572)

- Client programs now display more information for SSL errors to aid in diagnosis and debugging of connection problems. (Bug #21287, Bug #11745920)

Bugs Fixed

- **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 mistake in thread cleanup could cause a replication master to crash. (Bug #12578441)

- 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)

- Subsequent to `Prepared statement needs to be re-prepared` errors, inserts into `DECIMAL` columns caused a server exit. (Bug #12608543)

- In some cases, memory allocated for `Query_tables_list::sroutines()` was not freed properly. (Bug #12429877)

- 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 during two-phase commits if the binary log was used as the transaction coordinator log. (Bug #12346411)

- `Field_geom::reset()` failed to reset its base `Field_blob`. The range optimizer used the uninitialized field during optimization and execution, causing the server to exit. (Bug #11908153)

- An attempt to install nonexistent files during installation was corrected. (Bug #43247, Bug #11752142)

- 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)
- 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)
- 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)
- 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)
- `ALTER EVENT` could change the event status. (Bug #57156, Bug #11764334)

Changes in MySQL 5.5.13 (2011-05-31)

Note

Very old (MySQL 4.0) clients are not working temporarily due to a problem discovered after the release of MySQL 5.5.12. We are looking at fixing the problem. Update: This is fixed in MySQL 5.5.14.

Functionality Added or Changed

- The client-side plugin that accompanies the server-side Windows authentication plugin is now included in all MySQL distributions. This will permit clients from any release, whether commercial or community, to connect to a server that has the server-side plugin loaded. See The Windows Native Authentication Plugin (Bug #59780, Bug #11766631)

Bugs Fixed

- **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:** 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)

- **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:** Typographical errors appeared in the text of several replication error messages. (The word “position” was misspelled as “postion”.) (Bug #11762616, Bug #55229)

- 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)

- `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)

- Internal Performance Schema header files were unnecessarily installed publicly. (Bug #53281)

- 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 optimizer sometimes incorrectly processed `HAVING` clauses for queries that did not also have an `ORDER BY` clause. (Bug #48916, Bug #11756928)

- `LOAD DATA INFILE` errors could leak I/O cache memory. (Bug #58072, Bug #11765141)

- `PROCEDURE ANALYSE()` could leak memory for `NULL` results, and could return incorrect results if used with a `LIMIT` clause. (Bug #48137, Bug #11756242)

- 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)

- 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)

- On Linux, the `mysql` client built using the bundled `libedit` did not read `~/.editrc`. (Bug #49967, Bug #11757855)

- For `LOAD DATA INFILE`, multi-byte 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.

- Valgrind warnings caused by comparing index values to an uninitialized field were corrected. (Bug #58705, Bug #11765713)

- For repeated invocation of some stored procedures, the server consumed memory that it did not release until the connection terminated. (Bug #60025, Bug #11848763)
- 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 5.5.12 (2011-05-05)

Functionality Added or Changed

- 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)

Bugs Fixed

- **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 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)
- **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)
- On Windows, the server rejected client connections if no DNS server was available. (Bug #12325375)
- `mysql_upgrade` did not properly upgrade the `authentication_string` column of the `mysql.user` table. (Bug #11936829)
- `InnoDB` invoked some `zlib` functions without proper initialization. (Bug #11849231)
- `CREATE TABLE` permitted a `TABLESPACE` table option but did not write the option value to the `.frm` file. (Bug #11769356)
- `SHOW EVENTS` did not always show events from the correct database. (Bug #41907, Bug #11751148)
- 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)
- With prepared statements, the server could attempt to send result set metadata after the table had been closed. (Bug #56115, Bug #11763413)
- When the server was started with the `--skip-innodb` option, it initialized the `have_innodb` system variable to `YES` rather than `DISABLED`. (Bug #59393, Bug #11766306)
- 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)
- 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)
- `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)
- 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)

Changes in MySQL 5.5.11 (2011-04-07)

Functionality Added or Changed

- `InnoDB` now permits concurrent reads on a table while creating nonprimary unique indexes. (This was found to create problems and was reverted in 5.5.12.) (Bug #11784056)

- 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)

- 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)

- 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.

- For the Windows installer, debug information files and the embedded MySQL server have been removed from the standard MSI distribution file to reduce the download size for the majority of users.

  If these files are needed, the Zip distribution must be downloaded separately and be extracted in the installation directory, which is most probably `C:\Program Files\MySQL\MySQL Server 5.5` on English systems.

  Please note that upon product de-installation, these extracted files from the Zip distribution must be removed from the system manually.

- A new system variable, `max_long_data_size`, now controls the maximum size of parameter values that can be sent with the `mysql_stmt_send_long_data()` C API function. If not set at server startup, the default is the value of the `max_allowed_packet` system variable. This variable is deprecated. In MySQL 5.6, it is removed and the maximum parameter size is controlled by `max_allowed_packet`.

Bugs Fixed

- **Important Change:** The length of the `plugin` column of the `mysql.user` system table is increased to 64 characters. This is now the same size as the `name` column of the `mysql.plugin` table. (Bug #11766610, Bug #59752)

- **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:** A failed `DROP DATABASE` statement could break statement-based replication. (Bug #58381, Bug #11765416)

- 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.

- Division of large numbers could cause stack corruption. (Bug #11792200)

- The server read one byte too many when trying to process an XML string lacking a closing single quote (`'`) or double quote (`"`) character used as an argument for `UpdateXML()` or `ExtractValue()`. (Bug #59901, Bug #11766725)

  References: See also Bug #44332, Bug #11752979.

- Aggregation followed by a subquery could produce an incorrect result. (Bug #59839, Bug #11766675)

- `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)

- 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)

- `SHOW WARNINGS` output following `EXPLAIN EXTENDED` could include unprintable characters. (Bug #57341, Bug #11764503)
- 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)
- 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)
- 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)
- 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)

- `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)
- `FIND_IN_SET()` could work differently in MySQL 5.5 than in 5.1. (Bug #59405, Bug #11766317)
- In some cases, `SHOW WARNINGS` returned an empty result when the previous statement failed. (Bug #55847, Bug #11763166)
- 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)

Changes in MySQL 5.5.10 (2011-03-15)

C API Notes

- **Incompatible Change:** The shared library version of the client library was increased to 18 to reflect ABI changes, and avoid compatibility problems with the client library in MySQL 5.1. Note that this is an incompatible change between 5.5.10 and earlier 5.5 versions, so client programs that use the 5.5 client library should be recompiled against the 5.5.10 client library. (Bug #60061, Bug #11827366)

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.

Functionality Added or Changed

- 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)

- `mysqldump --xml` now displays comments from column definitions. (Bug #13618, Bug #11745324)

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #36544)

- **Important Change; InnoDB:** The `libaio` library, which has been used on Linux systems since MySQL 5.5.4, is now linked into `mysqld` dynamically rather than statically. If the library is not already on your Linux system, install it using the appropriate package manager for your distribution. The `libaio-dev` library is not sufficient; you must have the `libaio` library. (Bug #11893055, Bug #60544)

- **InnoDB:** Raised the number of I/O requests that each AIO helper thread could process, from 32 to 256. The new limit applies to Linux and Unix platforms; the limit on Windows remains 32. (Bug #59472)

- **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)

- **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)

- 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)

- `--autocommit=ON` did not work (it set the global `autocommit` value to 0, not 1). (Bug #59432, Bug #11766339)

- 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)

- `DELETE` or `UPDATE` statements could fail if they used `DATE` or `DATETIME` values with a year, month, or day part of zero. (Bug #59173)

- In debug builds, `SUBSTRING_INDEX(FORMAT(...), FORMAT(...))` could cause a server crash. (Bug #58371, Bug #11765406)

- 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)

- Outer joins with an empty table could produce incorrect results. (Bug #58422, Bug #11765451)

- 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 `mysql` client went into an infinite loop if the standard input was a directory. (Bug #57450, Bug #11764598)

- The expression *`const1`* `BETWEEN` *`const2`* `AND` *`field`* was optimized incorrectly and produced incorrect results. (Bug #57030, Bug #11764215)

- 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)

- Sorting using `ORDER BY AVG(DISTINCT decimal_col)` caused a server crash or incorrect results. (Bug #52123, Bug #11759784)

- `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)

- `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)

- 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)

- There was an erroneous restriction on file attributes for `LOAD DATA INFILE`. (Bug #59085, Bug #11766052)

- Setting the `optimizer_switch` system variable to an invalid value caused a server crash. (Bug #59894, Bug #11766719)

- `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 5.5.9 (2011-02-07)

Functionality Added or Changed

- The `mysqladmin` and `mysqldump` clients now have `--default-auth` and `--plugin-dir` options for specifying which authentication plugin and plugin directory to use. (Bug #58139, Bug #11765201)

- `sql_priv.h` now includes an `OPTION_ALLOW_BATCH` flag for the `transaction_allow_batching` feature of MySQL Cluster. (Bug #57604)

- 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`. Any other nonnumeric value is interpreted as `OFF`. (Bug #46393 improves on this such that `ON`, `TRUE`, `OFF`, and `FALSE` are recognized, and other values are invalid.) (Bug #51631, Bug #11759326)

  References: See also Bug #11754743.

- In the audit plugin interface, the `MYSQL_AUDIT_CONNECTION_CLASS` event class was added, and the `MYSQL_AUDIT_GENERAL_STATUS` subclass was added to the `MYSQL_AUDIT_GENERAL_CLASS` event class. The new symbol definitions can be found in the `plugin_audit.h` header file.

Bugs Fixed

- **Security Fix:** A security bug was fixed. (Bug #57952)

- **Performance; InnoDB:** `InnoDB` now uses the `PAUSE` instruction on all platforms where it is available. Previously, `InnoDB` used the `PAUSE` instruction only on Windows systems. (Bug #58666)

- **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)

- **Performance:** Queries involving `InnoDB` tables in the `INFORMATION_SCHEMA` tables `TABLE_CONSTRAINTS`, `KEY_COLUMN_USAGE`, or `REFERENTIAL_CONSTRAINTS` were slower than necessary because statistics were rechecked more often than required, even more so when many foreign keys were present. The improvement to this may be of particular benefit to users of MySQL Enterprise Monitor with many monitored servers or tens of thousands of tables. (Bug #43818, Bug #11752585)

- **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.

- **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:** The server could crash with an assertion error, if a stored procedure, stored function, or trigger modified one `InnoDB` table containing an auto-increment column, and dropped another `InnoDB` table containing an auto-increment column. (Bug #56228)

- **InnoDB:** In `InnoDB` status output, the value for `I/O sum[]` could be incorrect, displayed as a very large number. (Bug #57600)

- **InnoDB:** The command to create a debug build (`cmake -DWITH_DEBUG ...`) now automatically sets the `InnoDB` debugging flag `UNIV_DEBUG` on all platforms. Formerly, the `UNIV_DEBUG` flag might not be set for Windows platforms with Visual Studio and not on OS X with Xcode. (Bug #58279)

- **InnoDB:** When multiple `InnoDB` buffer pools were enabled, `SHOW ENGINE INNODB` statements displayed information about each one, but not summary information combining statistics for the entire buffer pool subsystem. Now, the aggregated information is displayed in the `BUFFER POOL AND MEMORY` section, and information about individual buffer pool instances is displayed in a new `INDIVIDUAL BUFFER POOL INFO` section. (Bug #58461)

- **InnoDB:** When the `lower_case_table_names` variable was set to 2, `InnoDB` could fail to restore a `mysqldump` dump of a table with foreign key constraints involving case-sensitive names. (Bug #55222)

- **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:** The `OPTIMIZE TABLE` statement reset the auto-increment counter for an `InnoDB` table. Now the auto-increment value is preserved across this operation. (Bug #18274)

- **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.

- **Partitioning:** Failed `ALTER TABLE ... TRUNCATE PARTITION` statements were written to the binary log. (Bug #58147)

- **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)

- **Replication:** While an `INSERT DELAYED` statement with a single inserted value does not return any visible warnings, such a warning could be still written into the error log. (Bug #57666, Bug #11764793)

  References: See also Bug #49567.

- **Replication:** When closing a session that used temporary tables, binary logging could sometimes fail with a spurious `Failed to write the DROP statement for temporary tables to binary log`. (Bug #57288)

- **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 the binary log, and Server Status Variables. (Bug #57275, Bug #11764443)

- **Replication:** By default, a value is generated for an `AUTO_INCREMENT` column by inserting either `NULL` or 0 into the column. Setting the `NO_AUTO_VALUE_ON_ZERO` server SQL mode suppresses this behavior for 0, so that it occurs only when `NULL` is inserted into the column.

  This behavior is also followed on a replication slave (by the slave SQL thread) when applying events that have been logged on the master using the statement-based format. However, when applying events that had been logged using the row-based format, `NO_AUTO_VALUE_ON_ZERO` was ignored, which could lead to an assertion.

  To fix this issue, the value of an `AUTO_INCREMENT` column is no longer generated when applying an event that was logged using the row-based row format, as this value is already contained in the changes applied on the slave. (Bug #56662)

- **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:** The `BINLOG` statement modified the values of session variables, which could lead to problems with operations such as point-in-time recovery. One such case occurred when replaying a row-based binary log which relied on setting `foreign_key_checks = OFF` at the session level to create and populate a set of `InnoDB` tables having foreign key constraints. (Bug #54903)

- **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:** Previously, when a statement failed with a different error on the slave than on the master, the slave SQL thread displayed a message containing:
  - The error message for the master error code
  - The master error code
  - The error message for the slaves error code
  - The slave error code

  However, the slave has no information with which to fill in any print format specifiers for the master message, so it actually displayed the message format string. To make it clearer that the slave is not displaying the actual message as it appears on the master, the slave now indicates that the master part of the output is the message format, not the actual message. For example, previously the slave displayed information like this:

```
Error: "Query caused different errors on master and slave. Error
on master: 'Duplicate entry '%-.192s' for key %d' (1062), Error on
slave: 'no error' (0). Default database: 'test'. Query: 'insert
into t1 values(1),(2)'" (expected different error codes on master
and slave)
```

  Now the slave displays this:

```
Error: "Query caused different errors on master and slave. Error
on master: message format='Duplicate entry '%-.192s' for key %d'
error code=1062 ; Error on slave: actual message='no error', error
code=0. Default database: 'test'. Query: 'insert into t1 values(1),(2)'"
(expected different error codes on master and slave)
```

  (Bug #46697)

- A Valgrind failure occurred in `fn_format` when called from `archive_discover`. (Bug #58205, Bug #11765259)
- Issuing `EXPLAIN EXTENDED` for a query that would use condition pushdown could cause `mysqld` to crash. (Bug #58553, Bug #11765570)
- It was possible to compile `mysqld` with Performance Schema support but with a dummy atomic-operations implementation, which caused a server crash. This problem does not affect binary distributions. It is helpful as a safety measure for users who build MySQL from source. (Bug #56769)
- `CMake` polluted the source tree by writing installation-related temporary files there. (Bug #58372)
- `CMake` did not add `LINK_LIBRARIES` for `MYSQL_ADD_PLUGIN` for `libmysqld`. (Bug #58158)
- `CREATE DATABASE` and `DROP DATABASE` caused `mysql --one-database` to lose track of the statement-filtering context. (Bug #54899)
- Configuration with maintainer mode enabled resulted in errors when compiling with `icc`. (Bug #57991, Bug #58871)
- Configuring MySQL with `-DWITHOUT_PERFSCHEMA_STORAGE_ENGINE=1` caused build failures. (Bug #58953)

- Security context references in `sp_head.cc` were rewritten for improved DTrace compatibility. (Bug #58350)
- The `cp1251` character set did not properly support the Euro sign (`0x88`). For example, converting a string containing this character to `utf8` resulted in `'?'` rather than the `utf8` Euro sign. (Bug #56639)
- The `ucs2` character set does not support characters outside the Basic Multilingual Plane (BMP), but converting to `ucs2` a string containing such characters did not produce a conversion-failure warning. (Bug #58321)
- On Solaris, the MySQL build failed if it was configured with debugging enabled. (Bug #58699)
- Several Valgrind warnings were fixed. (Bug #58948, Bug #59021)
- View creation could produce Valgrind warnings. (Bug #57352)
- During assignment of values to system variables, legality checks on the value range occurred too late, preventing proper error checking. (Bug #43233)
- Some unsigned system variables could be displayed with negative values. (Bug #55794)
- The `BIT_AND()` function could return incorrect results when a join returned no matching rows. (Bug #57954)
- If MySQL was built with Visual Studio Express, the project wixca was not built. (Bug #58411)
- An assertion could be raised during concurrent execution of `DROP DATABASE` and `REPAIR TABLE` if the drop deleted a table's `.TMD` file at the same time the repair tried to read details from the old file that was just removed.

  A problem could also occur when `DROP TABLE` tried to remove all files belonging to a table at the same time `REPAIR TABLE` had just deleted the table's `.TMD` file. (Bug #54486)
- If a client supplied a user name longer than the maximum 16 characters permitted for names stored in the MySQL grant tables, all characters were being considered significant when checking for a match. Historically, only the first 16 characters were used for matching; this behavior was restored. (Bug #49752)
- If the set of values aggregated with `AVG(DISTINCT)` contained a `NULL` value, the function result could be incorrect. (Bug #57932)
- 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)
- When `mysqld` printed crash dump information, it incorrectly indicated that some valid pointers were invalid. (Bug #51817)
- `EXPLAIN` could crash for queries that used `GROUP_CONCAT()`. (Bug #58396)
- On FreeBSD, if `mysqld` was killed with a `SIGHUP` signal, it could corrupt `InnoDB .ibd` files. (Bug #51023, Bug #11758773)
- An assertion could be raised for queries for which the optimizer could choose between Index Merge range access or const ref access methods. (Bug #58456)
- Unnecessary subquery evaluation in contexts such as statement preparation or view creation could cause a server crash. (Bug #57703)
- An assertion could be raised if the server was closing a session at the same time the session was being killed by another thread. (Bug #58136)

- `EXPLAIN` could crash for queries that accessed two derived tables. (Bug #58730)
- If the remote server for a `FEDERATED` table could not be accessed, queries for the `INFORMATION_SCHEMA.TABLES` table failed. (Bug #35333)
- The `my_seek()` and `my_tell()` functions ignored the `MY_WME` flag when they returned an error, which could cause client programs to hang. (Bug #48451)
- In rare cases, `LIKE` expressions failed for an indexed column that used a collation containing contractions. (Bug #57737)
- An `ORDER BY` clause was bound to the incorrect substatement when used in `UNION` context. (Bug #57986)
- 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)
- On Solaris, time-related functions such as `NOW()` or `SYSDATE()` could return a constant value. (Bug #42054)
- 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)
- `NULL` geometry values could cause a crash in `Item_func_spatial_collection::fix_length_and_dec`. (Bug #57321)
- After compilation from source, all header files were installed in the same directory, even those that should be installed into subdirectories of the installation include directory. (Bug #51925)
- On Mac OS X, a configuration error caused the preference pane to fail. (Bug #51264)
- `mysqlslap` failed to check for a `NULL` return from `mysql_store_result()` and crashed trying to process the result set. (Bug #59109, Bug #11766074)
- For `DIV` expressions, assignment of the result to multiple variables could cause a server crash. (Bug #59241, Bug #11766191)

  References: See also Bug #8457.
- 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)
- Condition pushdown optimization could push down conditions with incorrect column references. (Bug #58134, Bug #11765196)
- `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)

Changes in MySQL 5.5.8 (2010-12-03, General Availability)

Configuration Notes

- MySQL releases are now built on all platforms using `CMake` rather than the GNU autotools, so autotools support has been removed. For instructions on building MySQL with `CMake`, see Installing MySQL from Source. If you are familiar with autotools but not `CMake`, you might find this transition document helpful: Autotools to CMake Transition Guide. Third-party tools that need to extract the MySQL version number formerly found in `configure.in` can use the `VERSION` file. See MySQL Configuration and Third-Party Tools.

Functionality Added or Changed

- The time zone tables available at http://dev.mysql.com/downloads/timezones.html have been updated. These tables can be used on systems such as Windows or HP-UX that do not include zoneinfo files. (Bug #40230)

- Support for the `IBMDB2I` storage engine has been removed. (Bug #58079)

- For an upgrade to MySQL 5.5.7 from a previous release, the server exited if the `mysql.proxies_priv` table did not exist, making upgrades inconvenient. Now the server treats a missing `proxies_priv` table as equivalent to an empty table. However, after starting the server, you should still run `mysql_upgrade` to create the table. (Bug #57551)

- The client/server protocol now includes a `SERVER_QUERY_WAS_SLOW` flag to indicate when a query is slow; that is, when query execution exceeds the value of the `long_query_time` system variable. (Bug #57058)

- The following words are no longer reserved words the way they are in earlier MySQL 5.5 releases: `SLOW`, `GENERAL`, `IGNORE_SERVER_IDS`, `MASTER_HEARTBEAT_PERIOD` (Bug #57899)

- The `autocommit` system variable is enabled by default for all user connections, and the session value can be set for each new connection by setting the `init_connect` system variable to `SET autocommit=0`. However, this has no effect for users who have the `SUPER` privilege.

  Now the global `autocommit` value can be set at server startup, and this value is used to initialize the session value for all new connections, including those for users with the `SUPER` privilege. The variable is treated as a boolean value so it can be enabled with `--autocommit`, `--autocommit=1`, or `--enable-autocommit`. It can be disabled with `--autocommit=0`, `--skip-autocommit`, or `--disable-autocommit`. (Bug #57316)

- 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).

- Changes to replication in MySQL 5.6 make `mysqlbinlog` output generated by the `--base64-output=ALWAYS` option unusable, so `ALWAYS` is now deprecated and will be an invalid option value in MySQL 5.6. This should not be a significant problem because `--base64-output` values other than `AUTO` are supposed to be used only for debugging, not for production environments.

  References: See also Bug #28760.

Bugs Fixed

- **Security Fix; InnoDB:** A failed `CREATE TABLE` statement for an `InnoDB` table could allocate memory that was never freed. (Bug #56947)

- **Security Fix:** A security bug was fixed. (Bug #55146, Bug #56287)
- **Security Fix:** A security bug was fixed. (Bug #56814)
- **Security Fix:** A security bug was fixed. (Bug #54484)
- **Security Fix:** A security bug was fixed. (Bug #57130)
- **Security Fix:** A security bug was fixed. (Bug #57272)
- **Security Fix:** A security bug was fixed. (Bug #57477)
- **Security Fix:** A security bug was fixed. (Bug #57687)
- **Security Fix:** A security bug was fixed. (Bug #57659)
- **Security Fix:** A security bug was fixed. (Bug #58005)
- **Performance; InnoDB:** Improved concurrency when several `ANALYZE TABLE` or `SHOW TABLE STATUS` statements are run simultaneously for `InnoDB` tables. (Bug #53046)
- **Incompatible Change:** The following changes were made to the `performance_schema.threads` table for conformance with the implementation in MySQL 5.6:
  - `ID` column: Renamed to `PROCESSLIST_ID`, removed `NOT NULL` from definition.
  - `NAME` column: Changed from `VARCHAR(64)` to `VARCHAR(128)`.

  (Bug #57154)
- **Incompatible Change:** Previously, tables in the `performance_schema` database had uppercase names. This was incompatible with the `lower_case_table_names` system variable, and caused issues when the variable value was changed *after* installing or upgrading.

  Now `performance_schema` table names are lowercase, so they appear in uniform lettercase regardless of the `lower_case_table_names` setting. References to these tables in SQL statements should be given in lowercase. This is an incompatible change, but provides compatible behavior across different values of `lower_case_table_names`.

  If you upgrade to MySQL 5.5.8 from an earlier version of MySQL 5.5, be sure to run `mysql_upgrade` (and restart the server) to change the names of existing `performance_schema` tables from uppercase to lowercase. If `mysql_upgrade` does not work, use this procedure:

  1. Stop `mysqld`.
  2. Remove the `performance_schema/*.frm` files from the data directory.
  3. Create a separate “dummy” MySQL 5.5.8 installation.
  4. Copy the `performance_schema/*.frm` files from the dummy installation to the installation you are upgrading.
  5. Restart `mysqld` and run `mysqld_upgrade --force` and check that it does not produce errors.
  6. Remove the dummy installation.

  (Bug #57609)
- **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:** 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:** The `InnoDB` system tablespace could grow continually for a server under heavy load. (Bug #57611)
- **InnoDB:** Values could be truncated in certain `INFORMATION_SCHEMA` columns, such as `REFERENTIAL_CONSTRAINTS.REFERENCED_TABLE_NAME` and `KEY_COLUMN_USAGE.REFERENCED_TABLE_NAME`. (Bug #57960)
- **InnoDB:** A followup fix to bug #54678. `TRUNCATE TABLE` could still cause a crash (assertion error) in the debug version of the server. (Bug #57700)
- **InnoDB:** A large number of foreign key declarations could cause the output of the `SHOW CREATE STATEMENT` statement to be truncated. (Bug #56143)

- **InnoDB:** For an `InnoDB` table created with `ROW_FORMAT=COMPRESSED` or `ROW_FORMAT=DYNAMIC`, a query using the `READ UNCOMMITTED` isolation level could cause the server to stop with an assertion error, if `BLOB` or other large columns that use off-page storage were being inserted at the same time. (Bug #57799)

- **InnoDB:** Heavy concurrent updates of a `BLOB` column in an `InnoDB` table could cause a hang. (Bug #57579)

- **InnoDB:** An existing `InnoDB` table could be switched to `ROW_FORMAT=COMPRESSED` implicitly by a `KEY_BLOCK_SIZE` clause in an `ALTER TABLE` statement. Now, the row format is only switched to compressed if there is an explicit `ROW_FORMAT=COMPRESSED` clause. on the `ALTER TABLE` statement.

  Any valid, nondefault `ROW_FORMAT` parameter takes precedence over `KEY_BLOCK_SIZE` when both are specified. `KEY_BLOCK_SIZE` only enables `ROW_FORMAT=COMPRESSED` if `ROW_FORMAT` is not specified on either the `CREATE TABLE` or `ALTER TABLE` statement, or is specified as `DEFAULT`. In case of a conflict between `KEY_BLOCK_SIZE` and `ROW_FORMAT` clauses, the `KEY_BLOCK_SIZE` is ignored if `innodb_strict_mode` is off, and the statement causes an error if `innodb_strict_mode` is on. (Bug #56632)

- **InnoDB:** The clause `KEY_BLOCK_SIZE=0` is now permitted on `CREATE TABLE` and `ALTER TABLE` statements for `InnoDB` tables, regardless of the setting of `innodb_strict_mode`. The zero value has the effect of resetting the `KEY_BLOCK_SIZE` table parameter to its default value, depending on the `ROW_FORMAT` parameter, as if it had not been specified. That default is 8 if `ROW_FORMAT=COMPRESSED`. Otherwise, `KEY_BLOCK_SIZE` is not used or stored with the table parameters.

  As a consequence of this fix, `ROW_FORMAT=FIXED` is not permitted when `innodb_strict_mode` is enabled. (Bug #56628)

- **InnoDB:** Turning off the `innodb_stats_on_metadata` option could prevent the `ANALYZE TABLE` statement from updating the cardinality statistics of `InnoDB` tables. (Bug #57252)

- **InnoDB:** The server could stop with an assertion error on Windows Vista and Windows 7 systems. (Bug #57720)

- **InnoDB:** A query for an `InnoDB` table could return the wrong value if a column value was changed to a different case, and the column had a case-insensitive index. (Bug #56680, Bug #11763909)

- **InnoDB:** Clarified the message when a `CREATE TABLE` statement fails because a foreign key constraint does not have the required indexes. (Bug #16290)

- **Partitioning:** In debug builds, an `INSERT ... ON DUPLICATE KEY UPDATE col_name = 0` statement on an `AUTO_INCREMENT` column caused the server to crash. (Bug #57890)

- **Partitioning:** Issuing `ALTER TABLE ... ADD PRIMARY KEY` on a partitioned `InnoDB` table could cause the MySQL Server to crash. (Bug #57778)

- **Partitioning:** In-place `ALTER TABLE` operations (that do not involve a table copy) on a partitioned table could leave the table in an unusable state. (Bug #57985)

- **Replication:** Concurrent statements using a stored function and a `DROP DATABASE` statement that caused the same stored function to be dropped could cause statement-based replication to fail. This problem is resolved by making sure that `DROP DATABASE` takes an exclusive metadata lock on all stored functions and stored procedures that it causes to be dropped. (Bug #57663)

  References: See also Bug #30977.

- **Replication:** If there exist both a temporary table and a nontemporary table having the same name, updates normally apply only to the temporary table, with the exception of a `CREATE TABLE ...`

SELECT statement that creates a nontemporary table having the same name as an existing temporary table. When such a statement was replicated using the MIXED logging format, and the statement was unsafe for row-based logging, updates were misapplied to the temporary table.

Updates were also applied wrongly when a temporary table that used a transactional storage engine was dropped inside a transaction, followed by updates within the same transaction to a nontemporary table having the same name. (Bug #55478)

References: See also Bug #47899, Bug #55709.

- **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)

- **Replication:** When making changes to relay log settings using CHANGE MASTER TO, the I/O cache was not cleared. This could result in replication failure when the slave attempted to read stale data from the cache and then stopped with an assertion. (Bug #55263)

- **Replication:** Trying to read from a binary log containing a log event of an invalid type caused the slave to crash. (Bug #38718)

- **Replication:** Replication of SET and ENUM columns represented using more than 1 byte (that is, SET columns with more than 8 members and ENUM columns with more than 256 constants) between platforms using different endianness failed when using the row-based format. This was because columns of these types are represented internally using integers, but the internal functions used by MySQL to handle them treated them as strings. (Bug #52131)

  References: See also Bug #53528.

- **Replication:** When replicating the mysql.tables_priv table, the Grantor column was not replicated, and was thus left empty on the slave. (Bug #27606)

- Passing a string that was not null-terminated to UpdateXML() or ExtractValue() caused the server to fail with an assertion. (Bug #57279, Bug #11764447)

- The find_files() function used by SHOW statements performed redundant and unnecessary memory allocation. (Bug #51208)

- The server could crash inside memcpy() when reading certain Performance Schema tables. (Bug #56761, Bug #58003)

- Warnings raised by a trigger were not cleared upon successful completion. Now warnings are cleared if the trigger completes successfully, per the SQL standard. (Bug #55850)

- Handling of host name lettercase in GRANT statements was inconsistent. (Bug #36742)

- A user with no privileges on a stored routine or the mysql.proc table could discover the routine's existence. (Bug #57061)

- Several compilation problems were fixed. (Bug #57992, Bug #57993, Bug #57994, Bug #57995, Bug #57996, Bug #57997, Bug #58057)
- On file systems with case insensitive file names, and `lower_case_table_names=2`, the server could crash due to a table definition cache inconsistency. (Bug #46941)
- Clients using a client library older than MySQL 5.5.7 suffered loss of connection after executing `mysql_change_user()` while connected to a 5.5.7 server. (Bug #57689)
- If a `STOP SLAVE` statement was issued while the slave SQL thread was executing a statement that invoked the `SLEEP()` function, both statements hung. (Bug #56096)
- For `CMake` builds, some parts of the source were unnecessarily compiled twice if the embedded server was built. (Bug #55647)
- `OPTIMIZE TABLE` for `InnoDB` tables could raise an assertion. (Bug #55930)
- Boolean command options caused an error if given with an option value and the `loose-` option prefix. (Bug #54569)
- The server could crash as a result of accessing freed memory when populating `INFORMATION_SCHEMA.VIEWS` if a view could not be opened properly. (Bug #56540)
- `SET GLOBAL debug` could cause a crash on Solaris if the server failed to open the trace file. (Bug #57274)
- Setting the `read_only` system variable at server startup did not work. (Bug #58669)
- `mysql_upgrade` failed after an upgrade from MySQL 5.1. (Bug #58514)
- The Performance Schema did not count I/O for the binary log file. (Bug #58052)
- `BETWEEN` did not use indexes for `DATE` or `DATETIME` columns. (Bug #58190)
- `SHOW PROCESSLIST` displayed non-ASCII characters improperly. (Bug #57306)
- The `CMake` “wrapper” for `configure` (`configure.pl`) did not handle the `--with-comment` option properly. (Bug #52275)
- In debug builds, a missing `DBUG_RETURN` macro in `sql/client.c` caused `mysql` to be unable to connect to the server. (Bug #57744)
- In debug builds, an assertion could be raised during conversion of strings to floating-point values. (Bug #57203)
- When configuring the build with `-DBUILD_CONFIG=mysql_release` and building with Visual Studio Express, the build failed if `signtool.exe` was not present. (Bug #58313)
- When configuring the build with `-DBUILD_CONFIG=mysql_release` on Linux, `libaio` is required, but the error message if it was missing was uninformative. (Bug #58227)
- Incorrect error handling raised an assertion if character set conversion wrapped an item that failed. (Bug #57882)
- In debug builds, inserting a `FLOAT` value into a `CHAR(0)` column could cause a server crash. (Bug #58137)
- Queries executed using the Index Merge access method and a temporary file could return incorrect results. (Bug #56862)

- Grouping by a `TIME_TO_SEC()` function result could cause a server crash or incorrect results. Grouping by a function returning a `BLOB` could cause an unexpected “Duplicate entry” error and incorrect result. (Bug #52160)
- The Windows sample option files contained values more appropriate for Linux. (Bug #50021)
- After creation of a table with two foreign key constraints, the `INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS` table displayed only one of them. (Bug #57904)
- An error in a stored procedure could leave the session in a different default database. (Bug #54375)
- A failed `RENAME TABLE` operation could prevent a `FLUSH TABLES WITH READ LOCK` from completing. (Bug #47924)
- Failure to create a thread to handle a user connection could cause a server crash. (Bug #58080)
- Memory was allocated in `fn_expand()` for storing path names, but not freed anywhere. (Bug #58173)
- Valgrind warnings about overlapping memory when double-assigning the same variable were corrected. (Bug #56138)
- Use of `NAME_CONST()` in a `HAVING` clause caused a server crash. (Bug #58199)
- During configuration, `ADD_VERSION_INFO` in `cmake/mysql_version.cmake` failed if `LINK_FLAGS` was modified. (Bug #58074)
- The `MySQL-shared` RPM package failed to provide the lowercase virtual identifier `'mysql-shared'` in the RPM `'Provides'` tags (usually used for backward compatibility). (Bug #57596)
- Error messages for several delegate-related initialization error conditions that should not occur were changed to help identify the area of failure and to instruct the user to file a bug report if they do occur. A delegate is a set of internal data structures and algorithms. (Bug #47027)
- If the *`file_name`* argument to the `--defaults-file` or `--defaults-extra-file` option was not a full path name, it could be interpreted incorrectly in some contexts and cause a server crash. Now the *`file_name`* argument is interpreted as relative to the current working directory if given as a relative path name rather than as a full path name. (Bug #57108)
- With `CMake` 2.8.3, the `-DBUILD_CONFIG=mysql_release` option did not work. (Bug #58272)
- In debug builds, an assertion could be raised if a `send_eof()` method was called after an error occurred. (Bug #54812)
- The `ARCHIVE` storage engine could not be loaded with DTrace enabled on Solaris. (Bug #47739, Bug #11755901)
- `DELETE` with `FORCE INDEX` did not always force the index. (Bug #42209, Bug #11751370)
- `SET NAMES utf8 COLLATE utf8_sinhala_ci` did not work. (Bug #26474)
- The `utf16_bin` collation uses code-point order, not byte-by-byte order, as described at Unicode Character Sets. (The order was byte-by-byte in MySQL 5.5.7.)

Changes in MySQL 5.5.7 (2010-10-14)

Authentication Notes

- MySQL now supports pluggable authentication, such that the server uses plugins to authenticate incoming client connections. Client programs load an authentication plugin that interacts properly with the corresponding server plugin.

  Pluggable authentication enables two important capabilities, external authentication and proxy users:

  - Pluggable authentication makes it possible for clients to connect to the MySQL server with credentials that are appropriate for authentication methods other than native authentication based on passwords stored in the `mysql.user` table. For example, plugins can be created to use external authentication methods such as PAM, Windows login IDs, LDAP, or Kerberos.

  - If a user is permitted to connect, an authentication plugin can return to the server a user name different from the name of the connecting user, to indicate that the connecting user is a proxy for another user. While the connection lasts, the proxy user is treated, for purposes of access control, as having the privileges of a different user. In effect, one user impersonates another.

  Pluggable authentication entails these changes:

  - For user specifications in the `CREATE USER` and `GRANT` statements, there is a new `IDENTIFIED WITH` clause for specifying the authentication plugin.

  - For the `mysql.user` table, there are new columns that specify plugin information. The `plugin` column, if nonempty, indicates which plugin authenticates connections for an account. The `authentication_string` column is a string that the server passes to the plugin for connections by clients that authenticate using the plugin.

  - For the `mysql_options()` C API function, there are new `MYSQL_DEFAULT_AUTH` and `MYSQL_PLUGIN_DIR` options that enable client programs to load authentication plugins.

  - For the `mysql` client, there are new `--default-auth` and `--plugin-dir` options for specifying which authentication plugin and plugin directory to use. These options will be added to other clients in future releases.

  - For the `mysqltest` client, there is a new `--plugin-dir` option for specifying which plugin directory to use, and a new `connect()` command argument to specify an authentication plugin.

  - For the server plugin API, there is a new `MYSQL_AUTHENTICATION_PLUGIN` plugin type.

  - A new client plugin API enables client programs to manage plugins.

  - The native authentication methods previously supported in MySQL have been reimplemented as plugins. These methods provide against the current password format and pre-MySQL 4.1.1 format that uses shorter password hash values. This change reimplements the native methods as plugins that cannot be unloaded. Existing clients authenticate as before with no changes needed. In particular, starting the server with the `--secure-auth` option still prevents clients that have pre-4.1.1 password hashes from connecting, and `--skip-grant-tables` still disables all password checking.

  Proxy user capability entails these changes:

  - There is a new `PROXY` privilege that can be managed with the `GRANT` and `REVOKE` statements.

  - The new `proxy_user` and `external_user` system variables indicate whether the current session uses proxying.

  - A new `mysql.proxies_priv` grant table records proxy information for MySQL accounts.

Due to these changes, the server requires that a new grant table, `proxies_priv`, be present in the `mysql` database. If you are upgrading to MySQL 5.5.7 from a previous MySQL release rather than performing a new installation, the server will find that this table is missing and exit during startup with the following message:

```
Table 'mysql.proxies_priv' doesn't exist
```

To create the `proxies_priv` table, start the server with the `--skip-grant-tables` option to cause it to skip the normal grant table checks, then run `mysql_upgrade`. For example:

```
shell> mysqld --skip-grant-tables &
shell> mysql_upgrade
```

Then stop the server and restart it normally.

You can specify other options on the `mysqld` command line if necessary. Alternatively, if your installation is configured so that the server normally reads options from an option file, use the `--defaults-file` option to specify the file (enter each command on a single line):

```
shell> mysqld --defaults-file=/usr/local/mysql/etc/my.cnf
         --skip-grant-tables &
shell> mysql_upgrade
```

With the `--skip-grant-tables` option, the server does no password or privilege checking, so any client can connect and effectively have all privileges. For additional security, use the `--skip-networking` option as well to prevent remote clients from connecting.

Note

The upgrade problem just described is fixed in MySQL 5.5.8. The server treats a missing `proxies_priv` table as equivalent to an empty table.

For additional information, consult these references:

- Information about pluggable authentication, including installation and usage instructions: Pluggable Authentication.
- Information about proxy users: Proxy Users.
- Information about the server and client plugin API: Writing Plugins.
- Information about the C API functions for managing client plugins: See C API Client Plugin Functions.
- Information about current restrictions on the use of pluggable authentication, including which connectors support which plugins: See Restrictions on Pluggable Authentication. Third-party connector developers should read that section to determine the extent to which a connector can take advantage of pluggable authentication capabilities and what steps to take to become more compliant.

Configuration Notes

- MySQL releases now are built using `CMake` rather than the GNU autotools. Accordingly, the instructions for installing MySQL from source have been updated to discuss how to build MySQL using `CMake`. See Installing MySQL from Source. If you are familiar with autotools but not `CMake`, you might find these transition instructions helpful: Autotools to CMake Transition Guide

The build process is now similar enough on all platforms, including Windows, that there are no longer sections dedicated to notes for specific platforms.

The default installation layout when compiling from source now matches that used for binary distributions. You will notice these differences for installations from source distributions:

- `mysqld` is installed in `bin`, not `libexec`.
- `mysql_install_db` is installed in `scripts`, not `bin`.
- The data directory is `data`, not `var`.

The `make_binary_distribution` and `make_win_bin_dist` scripts are now obsolete. To create a binary distribution, use `make package`.

Functionality Added or Changed

- **Incompatible Change:** Previously, if you flushed the logs using `FLUSH LOGS` or `mysqladmin flush-logs` and `mysqld` was writing the error log to a file (for example, if it was started with the `--log-error` option), it renamed the current log file with the suffix `-old`, then created a new empty log file. This had the problem that a second log-flushing operation thus caused the original error log file to be lost unless you saved it under a different name. For example, you could use the following commands to save the file:

```
shell> mysqladmin flush-logs
shell> mv host_name.err-old backup-directory
```

To avoid the preceding file-loss problem, renaming no longer occurs. The server merely closes and reopens the log file. To rename the file, you can do so manually before flushing. Then flushing the logs reopens a new file with the original file name. For example, you can rename the file and create a new one using the following commands:

```
shell> mv host_name.err host_name.err-old
shell> mysqladmin flush-logs
shell> mv host_name.err-old backup-directory
```

(Bug #29751)

References: See also Bug #56821.

- A new status variable, `Handler_read_last`, displays the number of requests to read the last key in an index. With `ORDER BY`, the server issues a first-key request followed by several next-key requests, whereas with `ORDER BY DESC`, the server issues a last-key request followed by several previous-key requests. (Bug #52312)
- The unused and undocumented `thread_pool_size` system variable was removed. (Bug #57338)
- Added a new `SHOW PROCESSLIST` state, `Waiting for query cache lock`. This indicates that a session is waiting to take the query cache lock while it performs some query cache operation. (Bug #56822)
- The `pstack` library was nonfunctional and has been removed, along with the `--with-pstack` option for `configure` and the `--enable-pstack` option for `mysqld`. (Bug #57210)
- Previously, the server supported values of `OFF`, `ON`, and `FORCE` for the `--plugin_name=value` option format for controlling plugin loading using an option named after the plugin. Such options now support

a FORCE_PLUS_PERMANENT value. This value is like FORCE, but in addition prevents the plugin from being unloaded at runtime. If a user attempts to do so with UNINSTALL PLUGIN, an error occurs. See Installing and Uninstalling Plugins.

In addition, the INFORMATION_SCHEMA.PLUGINS table now has a LOAD_OPTION column that indicates the plugin loading value (OFF, ON, FORCE, or FORCE_PLUS_PERMANENT). See The INFORMATION_SCHEMA PLUGINS Table.

Bugs Fixed

- **Security Fix; Incompatible Change; InnoDB:** Issuing TRUNCATE TABLE and examining the same table's information in the INFORMATION_SCHEMA database at the same time could cause a crash in the debug version of the server.

  As a result of this change, InnoDB always uses the fast truncation technique, equivalent to DROP TABLE and CREATE TABLE. It no longer performs a row-by-row delete for tables with parent-child foreign key relationships. TRUNCATE TABLE returns an error for such tables. Modify your SQL to issue DELETE FROM *table_name* for such tables instead. (Bug #54678)

- **Security Fix:** The PolyFromWKB() function could crash the server when improper WKB data was passed to the function. (Bug #51875, Bug #11759554, CVE-2010-3840)

- **Security Fix:** In prepared-statement mode, EXPLAIN for a SELECT from a derived table caused a server crash. (Bug #54488)

- **Security Fix:** The CONVERT_TZ() function crashed the server when the timezone argument was an empty SET column value. (Bug #55424)

- **Security Fix:** EXPLAIN EXTENDED caused a server crash with some prepared statements. (Bug #54494)

- **Security Fix:** The server crashed for assignment of values of types other than Geometry to items of type GeometryCollection (MultiPoint, MultiCurve, MultiSurface). Now the server checks the value type and fails with bad geometry value if it detects incorrect parameters. (Bug #55531)

- **Performance; InnoDB:** The master InnoDB background thread could sometimes cause transient performance drops due to excessive flushing of modified pages. (Bug #56933)

- **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:** `HANDLER ... READ` statements that invoke stored functions can cause replication errors. Such statements are now disallowed and result in an `ER_NOT_SUPPORTED_YET` error. (Bug #54920)

- **Important Change; InnoDB:** The server could crash with an assertion, possibly leading to data corruption, while updating the primary key of an `InnoDB` table containing `BLOB` or other columns requiring off-page storage. This fix applies to the `InnoDB` Plugin in MySQL 5.1, and to `InnoDB` 1.1 in MySQL 5.5. (Bug #55543)

- **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 server crashed during a `RENAME TABLE` operation on an `InnoDB` table, subsequent crash recovery could fail. This problem could also affect an `ALTER TABLE` statement that caused a rename operation internally. (Bug #55027)

- **InnoDB:** The output from the `SHOW ENGINE INNODB STATUS` command can now be up to 1MB. Formerly, it was truncated at 64KB. Monitoring applications that parse that output can check whether it exceeds this new, larger limit by testing the `Innodb_truncated_status_writes` status variable. (Bug #56922)

- **InnoDB:** Changed the locking mechanism for the `InnoDB` data dictionary during `ROLLBACK` operations, to improve concurrency for `REPLACE` statements. (Bug #54538)

- **InnoDB:** Improved the performance of `UPDATE` operations on `InnoDB` tables, when only non-indexed columns are changed. (Bug #56340)

- **InnoDB:** The server could crash with a high volume of concurrent `LOCK TABLES` and `UNLOCK TABLES` statements. (Bug #57345)

- **InnoDB:** The server could crash on shutdown, if started with `--innodb-use-system-malloc=0`. (Bug #55627)

- **InnoDB:** A heavy workload with a large number of threads could cause a crash in the debug version of the server. (Bug #55699)

- **InnoDB:** When MySQL was restarted after a crash with the option `innodb_force_recovery=6`, certain queries against `InnoDB` tables could fail, depending on `WHERE` or `ORDER BY` clauses.

Usually in such a disaster recovery situation, you dump the entire table using a query without these clauses. During advanced troubleshooting, you might use queries with these clauses to diagnose the position of the corrupted data, or to recover data following the corrupted part. (Bug #55832)

- **InnoDB:** For debug builds, a `SELECT ... FOR UPDATE` statement affecting a range of rows in an `InnoDB` table could cause a server crash. (Bug #56716)
- **InnoDB:** If the server crashed during an `ALTER TABLE` operation on an `InnoDB` table, examining the table through `SHOW CREATE TABLE` or querying the `INFORMATION_SCHEMA` tables could cause the server to stop with an assertion error. (Bug #56982)
- **InnoDB:** `InnoDB` incorrectly reported an error when a cascading foreign key constraint deleted more than 250 rows. (Bug #57255)
- **InnoDB:** `InnoDB` startup messages now include the start and end times for buffer pool initialization, and the total buffer pool size. (Bug #48026)
- **InnoDB:** With multiple buffer pools enabled, `InnoDB` could flush more data from the buffer pool than necessary, causing extra I/O overhead. (Bug #54346)
- **InnoDB:** The `CHECK TABLE` command could cause a time-consuming verification of the `InnoDB` adaptive hash index memory structure. Now this extra checking is only performed in binaries built for debugging. (Bug #55716)
- **InnoDB:** `InnoDB` transactions could be incorrectly committed during recovery, rather than rolled back, if the server crashed and was restarted after performing `ALTER TABLE ... ADD PRIMARY KEY` on an `InnoDB` table, or some other operation that involves copying the entire table. (Bug #53756)
- **InnoDB:** Setting the `PACK_KEYS=0` table option for an `InnoDB` table prevented new indexes from being added to the table. (Bug #54606)
- **InnoDB:** The server could crash when opening an `InnoDB` table linked through foreign keys to a long chain of child tables. (Bug #54582, Bug #11762038)
- **Partitioning:** An `ALTER TABLE` statement acting on table partitions that failed while the affected table was locked could cause the server to crash. (Bug #56172)
- **Partitioning:** Multiple-table `UPDATE` statements involving a partitioned `MyISAM` table could cause this table to become corrupted. Not all tables affected by the `UPDATE` needed to be partitioned for this issue to be observed. (Bug #55458)
- **Partitioning:** `EXPLAIN PARTITIONS` returned bad estimates for range queries on partitioned `MyISAM` tables. In addition, values in the `rows` column of `EXPLAIN PARTITIONS` output did not take partition pruning into account. (Bug #53806, Bug #46754)
- **Replication:** `SET PASSWORD` caused failure of row-based replication between a MySQL 5.1 master and a MySQL 5.5 slave.

  This fix makes it possible to replicate `SET PASSWORD` correctly, using row-based replication between a master running MySQL 5.1.53 or a later MySQL 5.1 release to a slave running MySQL 5.5.7 or a later MySQL 5.5 release. (Bug #57098)

  References: See also Bug #55452, Bug #57357.
- **Replication:** When a slave tried to execute a transaction larger than the slave's value for `max_binlog_cache_size`, it crashed. This was caused by an assertion that the server should roll back only the statement but not the entire transaction when the error `ER_TRANS_CACHE_FULL` occurred.

However, the slave SQL thread always rolled back the entire transaction whenever any error occurred, regardless of the type of error. (Bug #55375)

- **Replication:** The error message for `ER_SLAVE_HEARTBEAT_VALUE_OUT_OF_RANGE` was hard coded in English in `sql_yacc.yy`, so that it could not be translated in `errmsg.txt` for other languages.

  Additionally, this same error message was used for three separate error conditions:

  - When the heartbeat period exceeded the value of `slave_net_timeout`.
  - When the heartbeat period was nonnegative but shorter than 1 millisecond.
  - When the value for the heartbeat period was either negative or greater than the maximum permitted.

  These issues have been addressed as follows:

  - By using three distinct error messages for each of the conditions listed previously.
  - By moving the sources for these error messages into the `errmsg-utf8.txt` file to facilitate translations into languages other than English.

  (Bug #54144)

- **Replication:** Backticks used to enclose identifiers for savepoints were not preserved in the binary log, which could lead to replication failure when the identifier, stripped of backticks, could be misinterpreted, causing a syntax or other error.

  This could cause problems with MySQL application programs making use of generated savepoint IDs. If, for instance, `java.sql.Connection.setSavepoint()` is called without any parameters, Connector/J automatically generates a savepoint identifier consisting of a string of hexadecimal digits `0`-`F` encased in backtick (`` ` ``) characters. If such an ID took the form `` `NeN` `` (where *`N`* represents a string of the decimal digits `0`-`9`, and `e` is a literal uppercase or lowercase “E” character). Removing the backticks when writing the identifier into the binary log left behind a substring which the slave MySQL server tried to interpret as a floating point number, rather than as an identifier. The resulting syntax error caused loss of replication. (Bug #55961)

  References: See also Bug #55962.

- **Replication:** Prepared multiple-row `INSERT DELAYED` statements were written to the binary log without `DELAYED`. (Bug #56678, Bug #11763907)

  References: This bug is a regression of Bug #54579, Bug #11762035.

- Multiple-statement execution could fail. (Bug #40877)

- The server entered an infinite loop with high CPU utilization after an error occurred during flushing of the I/O cache. (Bug #55629)

- Performance for certain read-only queries, in particular `point_select`, had deteriorated compared to previous versions. (Bug #56585)

- Using `REPAIR TABLE` *`tbl_name`* `USE_FRM` on a `MERGE` table caused the server to crash. (Bug #46339)

- In `LOAD DATA INFILE`, using a `SET` clause to set a column equal to itself caused a server crash. (Bug #51850)

- For the Performance Schema, the default number of rwlock classes was increased to 30, and the default number of rwlock and mutex instances was increased to 1 million. These changes were made to account for the volume of data instrumented when the `InnoDB` storage engine is used (because of the `InnoDB` buffer pool). (Bug #55576)
- Trailing space removal for `utf32` strings was done with non-multi-byte-safe code, leading to incorrect result length and assertion failure. (Bug #42511)
- The `plugin_ftparser.h` and `plugin_audit.h` include files are part of the public API/ABI, but were not tested by the ABI check. (Bug #52821)
- `MIN()` or `MAX()` with a subquery argument could raise a debug assertion for debug builds or return incorrect data for nondebug builds. (Bug #54465)
- The fix for Bug #30234 caused the server to reject the `DELETE tbl_name.* ...` Access compatibility syntax for multiple-table `DELETE` statements. (Bug #53034)
- `INFORMATION_SCHEMA` plugins with no `deinit()` method resulted in a memory leak. (Bug #54253)
- If a view was named as the destination table for `CREATE TABLE ... SELECT`, the server produced a warning whether or not `IF NOT EXISTS` was used. Now it produces a warning only when `IF NOT EXISTS` is used, and an error otherwise. (Bug #55777)
- Setting `SETUP_INSTRUMENTS.TIMER = 'NO'` caused `TIMER_WAIT` values for aggregations to be `NULL` rather than 0. (Bug #53874)
- Short (single-letter) command-line options did not work. (Bug #55873)
- Queries involving predicates of the form *`const`* `NOT BETWEEN` *`not_indexed_column`* `AND` *`indexed_column`* could return incorrect data due to incorrect handling by the range optimizer. (Bug #54802)
- A negative `TIME` argument to `MIN()` or `MAX()` could raise an assertion. (Bug #56120)
- After the fix for Bug #39653, the shortest available secondary index was used for full table scans. The primary clustered key was used only if no secondary index could be used. However, when the chosen secondary index includes all columns of the table being scanned, it is better to use the primary index because the amount of data to scan is the same but the primary index is clustered. This is now taken into account. (Bug #55656)
- If one session attempted to drop a database containing a table which another session had opened with `HANDLER`, any instance of `ALTER DATABASE`, `CREATE DATABASE`, or `DROP DATABASE` issued by the latter session produced a deadlock. (Bug #54360)
- In debug builds, the server raised an assertion for `DROP DATABASE` in installations that had an outdated or corrupted `mysql.proc` table. For example, this affected `mysql_upgrade` when run as part of a MySQL 5.1 to 5.5 upgrade. (Bug #56137)
- The ordering for supplementary characters in the `utf8mb4_bin`, `utf16_bin`, and `utf32_bin` collations was incorrect. (Bug #55980)
- It was possible to compile `mysqld` with Performance Schema support but with a dummy atomic-operations implementation, which caused a server crash. This problem does not affect binary distributions. It is helpful as a safety measure for users who build MySQL from source. (Bug #56521)
- An atomic “compare and swap” operation using x86 assembly code (32 bit) could access incorrect data, which would make it work incorrectly and lose the intended atomicity. This in turn caused the MySQL

server to work on inconsistent data structures and return incorrect data. That code affected only 32-bit builds; the effect has been observed when `icc` was used to build binaries. With `gcc`, no incorrect results have been observed during tests, so this fix is a proactive one. Other compilers do not use this assembly code. (Bug #52419)

- Row subqueries producing no rows were not handled as `UNKNOWN` values in row-comparison expressions. (Bug #54190)
- In some cases, when the left part of a `NOT IN` subquery predicate was a row and contained `NULL` values, the query result was incorrect. (Bug #51070)
- `EXPLAIN` produced an incorrect `rows` value for queries evaluated using an index scan and that included `LIMIT`, `GROUP BY`, and `ORDER BY` on a computed column. (Bug #50394)
- A malformed packet sent by the server when the query cache was in use resulted in lost-connection errors. (Bug #42503)
- For some queries, the optimizer produced incorrect results using the Index Merge access method with `InnoDB` tables. (Bug #50402)
- With an `UPDATE IGNORE` statement including a subquery that was evaluated using a temporary table, an error transferring the data from the temporary was ignored, causing an assertion to be raised. (Bug #54543)
- `CREATE TABLE` failed if a column referred to in an index definition and foreign key definition had different lettercases in the two definitions. (Bug #39932)
- If a query specified a `DATE` or `DATETIME` value in a format different from `'YYYY-MM-DD HH:MM:SS'`, a greater-than-or-equal (`>=`) condition matched only greater-than values in an indexed `TIMESTAMP` column. (Bug #55779, Bug #50774, Bug #11758558)
- If there was an active `SELECT` statement, an error arising during trigger execution could cause a server crash. (Bug #55421)
- `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. (Bug #47485)
- A buffer overrun could occur when formatting `DBL_MAX` numbers. (Bug #57209)
- The `tcmalloc` library was missing from binary MySQL packages for Linux. (Bug #56267)
- If an application using the embedded server called `mysql_library_init()` a second time after calling `mysql_library_init()` and `mysql_library_end()` to start and stop the server, the application crashed when reading option files. (Bug #53251)
- Assignment of `InnoDB` scalar subquery results to a variable resulted in unexpected `S` locks in `READ COMMITTED` transaction isolation level. (Bug #55382)
- The `max_length` metadata value of `MEDIUMBLOB` types was reported as 1 byte greater than the correct value. (Bug #53296)
- The server could crash during shutdown due to a race condition relating to Performance Schema cleanup. (Bug #55105, Bug #56324)
- The server could crash inside `memcpy()` when reading certain Performance Schema tables. (Bug #56761, Bug #58003)

- `CHECKSUM TABLE` for Performance Schema tables could cause a server crash due to uninitialized memory reads. (Bug #50557)
- Memory leaks detected by Valgrind were corrected. (Bug #56709)
- If the global and session `debug` system variables had the same value, the debug trace file could be closed twice, leading to freeing already freed memory and a server crash. (Bug #46165)
- `ALTER TABLE` on a `MERGE` table could result in deadlock with other connections. (Bug #56292, Bug #57002)
- An `INSERT DELAYED` statement for a `MERGE` table could cause deadlock if it occurred as part of a transaction or under `LOCK TABLES`, and there was a concurrent DDL or `LOCK TABLES ... WRITE` statement that tried to lock one of its underlying tables. (Bug #56251)
- The server crashed if a table maintenance statement such as `ANALYZE TABLE` or `REPAIR TABLE` was executed on a `MERGE` table and opening and locking a child table failed. For example, this could happen if a child table did not exist or if a lock timeout happened while waiting for a conflicting metadata lock to disappear.

  As a consequence of this bug fix, it is now possible to use `CHECK TABLE` for log tables without producing an error. (Bug #56422, Bug #56494)
- In debug builds, `FLUSH TABLE` *`table_list`* `WITH READ LOCK` for a `MERGE` table led to an assertion failure if one of the table's children was not present in the list of tables to be flushed. (Bug #55273)
- Deadlock could occur for heavily concurrent workloads consisting of a mix of DML, DDL, and `FLUSH TABLES` statements affecting the same set of tables. (Bug #56715, Bug #56404, Bug #56405)
- A bad `DBUG_PRINT` statement in `fill_schema_schemata()` caused server crashes on Solaris. (Bug #54478)
- Executing `XA END` after an XA transaction was already ended raised an assertion. (Bug #56448)
- Comparison of one `STR_TO_DATE()` result with another could return incorrect results. (Bug #56271)
- On Solaris with `gcc` 3.4.6, `ha_example.so` was built with DTrace support even if the server was not, causing plugin loading problems. (Bug #55966)
- On Mac OS X, `RENAME TABLE` raised an assertion if the `lower_case_table_names` system variable was 2 and the old table name was specified in uppercase. (Bug #56595)
- If `ALTER EVENT` failed to load an event after altering it, an assertion could be raised. This could occur, for example, if `ALTER EVENT` was killed with `KILL QUERY`. (Bug #44171)
- When `mysqld` was started as a service on Windows and `mysqld` was writing the error log to a file (for example, if it was started with the `--log-error` option), the server reassigned the file descriptors of the `stdout` and `stderr` streams to the file descriptor of the log file. On Windows, if `stdout` or `stderr` is not associated with an output stream, the file descriptor returns a negative value. Previously, this caused the file descriptor reassignment to fail and the server to abort. To avoid this problem on Windows, the server now first assigns the `stdout` and `stderr` streams to the log file stream by opening this file. This causes the `stdout` and `stderr` file descriptors to be nonzero and the server can successfully reassign them to the file descriptor of the log file. (Bug #56821)

  References: This bug is a regression of Bug #29751.
- `mysqlcheck` behaved differently depending on the order in which options were given on the command line. (Bug #35269)

- `CASE` expressions with a mix of operands in different character sets sometimes returned incorrect results. (Bug #55744)
- Constant `SUBTIME()` expressions could return incorrect results. (Bug #57039)
- An assertion could be raised by `DELETE` on a view that referenced another view which in turn (directly or indirectly) referenced more than one table. (Bug #51099)
- `COALESCE()` in MySQL 5.5 could return a result different from MySQL 5.1 for some arguments. (Bug #57095)
- A `SELECT` statement could produce a number of rows different from a `CREATE TABLE ... SELECT` that was supposed to select the same rows. (Bug #56423)

  References: This bug is a regression of Bug #38999.
- When invoked to display a help message, `mysqld` also displayed spurious warning or error messages. (Bug #30025)

Changes in MySQL 5.5.6 (2010-09-13, Release Candidate)

Functionality Added or Changed

- **Incompatible Change:** The `SHA2()` function now returns a character string with the connection character set and collation. Previously, it returned a binary string. This is the same change made for several other encryption functions in MySQL 5.5.3. (Bug #54661)
- **InnoDB:** The mechanism that checks if there is enough space for redo logs was improved, reducing the chance of encountering this message: `ERROR: the age of the last checkpoint is x, which exceeds the log group capacity y`. (Bug #39168)
- **InnoDB:** Improved performance and scalability on Windows systems, especially for Windows Vista and higher. Re-enabled the use of atomic instructions on Windows systems. For Windows Vista and higher, reduced the number of event handles used. To compile on Windows systems now requires Windows SDK v6.0 or later; either upgrade to Visual Studio 2008 or 2010, or for Visual Studio 2005, install Windows SDK Update for Windows Vista. (Bug #22268)
- Overhead for the Performance Schema interface was reduced. (Bug #55087)
- Previously, `MySQL-shared-compat` RPMs for Linux contained both the current and previous client library versions for the target platform. Thus, the package contents overlapped with `MySQL-shared` RPMs, which contain only the current client library version. This can result in problems in two cases:

  - When the `MySQL-shared` RPM is installed but later it is determined that the `MySQL-shared-compat` RPM is needed (an application is installed that was linked against an older client library). Installing the `MySQL-shared-compat` RPM results in a conflict because both include the current library version. This can be overcome by using the `--force` option to RPM, or by first uninstalling the `MySQL-shared` RPM (which breaks dependencies).
  - When the `MySQL-shared-compat` RPM is installed, but old applications that require it are removed or upgraded to the current library version. In this case, `MySQL-shared-compat` cannot be replaced with `MySQL-shared` as long as current applications are installed. This can be overcome by using the `--force` option to RPM, which incurs the risk of breaking dependencies.

  Now the `MySQL-shared-compat` RPMs include only older client library versions and no longer include the current version, so that the `MySQL-shared` and `MySQL-shared-compat` RPM contents no longer overlap. The `MySQL-shared-compat` RPM can be installed even if the `MySQL-shared` RPM is installed, without producing conflicts related to the current library version. The `MySQL-shared-compat`

RPM can be uninstalled when old applications are removed or upgraded to the current library version, without breaking applications that already use the current library version.

If you previously installed the `MySQL-shared-compat` RPM because you needed both the current and previous libraries, you should install both the `MySQL-shared` and `MySQL-shared-compat` RPMs now. (Bug #56150)

References: See also Bug #12368215.

- `REPAIR TABLE` and `OPTIMIZE TABLE` table now catch and throw any errors that occur while copying table statistics from the old corrupted file to newly created file. For example. if the user ID of the owner of the `.frm`, `.MYD`, or `.MYI` file is different from the user ID of the `mysqld` process, `REPAIR TABLE` and `OPTIMIZE TABLE` generate a "cannot change ownership of the file" error unless `mysqld` is started by the `root` user. (Bug #61598, Bug #13600058)
- Within stored programs, `LIMIT` clauses now permit integer-valued routine parameters or local variables as parameters. (Bug #11918)
- Code was removed for the following no-longer-supported platforms: NetWare, MS-DOS, VMS, QNX, and 32-bit SPARC.

Bugs Fixed

- **Security Fix; InnoDB:** After changing the values of the `innodb_file_format` or `innodb_file_per_table` configuration parameters, DDL statements could cause a server crash. (Bug #55039, CVE-2010-3676)
- **Security Fix:** Queries could cause a server crash if the `GREATEST()` or `LEAST()` function had a mixed list of numeric and `LONGBLOB` arguments, and the result of such a function was processed using an intermediate temporary table. (Bug #54461, CVE-2010-3838)
- **Security Fix:** During evaluation of arguments to extreme-value functions such as `LEAST()` and `GREATEST()`, type errors did not propagate properly, causing the server to crash. (Bug #55826, CVE-2010-3833)
- **Security Fix:** The server could crash after materializing a derived table that required a temporary table for grouping. (Bug #55568, CVE-2010-3834)
- **Security Fix:** Queries with nested joins could cause an infinite loop in the server when used from stored procedures and prepared statements. (Bug #53544, CVE-2010-3839)
- **Security Fix:** `GROUP_CONCAT()` and `WITH ROLLUP` together could cause a server crash. (Bug #54476, CVE-2010-3837)
- **Security Fix:** Pre-evaluation of `LIKE` predicates during view preparation could cause a server crash. (Bug #54568, Bug #11762026, CVE-2010-3836)
- **Security Fix:** A user-variable assignment expression that is evaluated in a logical expression context can be precalculated in a temporary table for `GROUP BY`. However, when the expression value is used after creation of the temporary table, it was re-evaluated, not read from the table, and a server crash resulted. (Bug #55564, CVE-2010-3835)
- **Security Fix:** After `ALTER TABLE` was used on a temporary transactional table locked by `LOCK TABLES`, any later attempts to execute `LOCK TABLES` or `UNLOCK TABLES` caused a server crash. (Bug #54117)
- **Security Fix:** Using `EXPLAIN` with queries of the form `SELECT ... UNION ... ORDER BY (SELECT ... WHERE ...)` could cause a server crash. (Bug #52711, CVE-2010-3682)

- **Security Fix:** Use of `TEMPORARY InnoDB` tables with nullable columns could cause a server crash. (Bug #54044, CVE-2010-3680)
- **Security Fix:** A malformed argument to the `BINLOG` statement could result in Valgrind warnings or a server crash. (Bug #54393, CVE-2010-3679)
- **Security Fix:** Incorrect handling of `NULL` arguments could lead to a crash for `IN()` or `CASE` operations when `NULL` arguments were either passed explicitly as arguments (for `IN()`) or implicitly generated by the `WITH ROLLUP` modifier (for `IN()` and `CASE`). (Bug #54477, CVE-2010-3678)
- **Security Fix:** Joins involving a table with a unique `SET` column could cause a server crash. (Bug #54575, CVE-2010-3677)
- **Security Fix:** A security bug was fixed. (Bug #49124)
- **Performance; InnoDB:** The setting `innodb_change_buffering=all` could produce slower performance for some operations than the previous default, `innodb_change_buffering=inserts`. (Bug #54914)
- **Performance; InnoDB:** An `EXPLAIN` plan for an `InnoDB` table could vary greatly in the estimated cost for a `BETWEEN` clause. (Bug #53761)
- **Incompatible Change; Replication:** As of MySQL 5.5.6, handling of `CREATE TABLE IF NOT EXISTS ... SELECT` statements has been changed for the case that the destination table already exists:

  - Previously, for `CREATE TABLE IF NOT EXISTS ... SELECT`, MySQL produced a warning that the table exists, but inserted the rows and wrote the statement to the binary log anyway. By contrast, `CREATE TABLE ... SELECT` (without `IF NOT EXISTS`) failed with an error, but MySQL inserted no rows and did not write the statement to the binary log.
  - MySQL now handles both statements the same way when the destination table exists, in that neither statement inserts rows or is written to the binary log. The difference between them is that MySQL produces a warning when `IF NOT EXISTS` is present and an error when it is not.

  This change in handling of `IF NOT EXISTS` results in an incompatibility for statement-based replication from a MySQL 5.1 master with the original behavior and a MySQL 5.5 slave with the new behavior. Suppose that `CREATE TABLE IF NOT EXISTS ... SELECT` is executed on the master and the destination table exists. The result is that rows are inserted on the master but not on the slave. (Row-based replication does not have this problem.)

  To address this issue, statement-based binary logging for `CREATE TABLE IF NOT EXISTS ... SELECT` is changed in MySQL 5.1 as of 5.1.51:

  - If the destination table does not exist, there is no change: The statement is logged as is.
  - If the destination table does exist, the statement is logged as the equivalent pair of `CREATE TABLE IF NOT EXISTS` and `INSERT ... SELECT` statements. (If the `SELECT` in the original statement is preceded by `IGNORE` or `REPLACE`, the `INSERT` becomes `INSERT IGNORE` or `REPLACE`, respectively.)

  This change provides forward compatibility for statement-based replication from MySQL 5.1 to 5.5 because when the destination table exists, the rows will be inserted on both the master and slave. To take advantage of this compatibility measure, the 5.1 server must be at least 5.1.51 and the 5.5 server must be at least 5.5.6.

  To upgrade an existing 5.1-to-5.5 replication scenario, upgrade the master first to 5.1.51 or higher. Note that this differs from the usual replication upgrade advice of upgrading the slave first.

A workaround for applications that wish to achieve the original effect (rows inserted regardless of whether the destination table exists) is to use `CREATE TABLE IF NOT EXISTS` and `INSERT ... SELECT` statements rather than `CREATE TABLE IF NOT EXISTS ... SELECT` statements.

Along with the change just described, the following related change was made: Previously, if an existing view was named as the destination table for `CREATE TABLE IF NOT EXISTS ... SELECT`, rows were inserted into the underlying base table and the statement was written to the binary log. As of MySQL 5.1.51 and 5.5.6, nothing is inserted or logged. (Bug #47442, Bug #47132, Bug #48814, Bug #49494)

- **Incompatible Change:** Handling of warnings and errors during stored program execution was problematic:
  - If one statement generated several warnings or errors, only the handler for the first was activated, even if another might be more appropriate. Now the server chooses the more appropriate handler.
  - Warning or error information could be lost.

  (Bug #36185, Bug #5889, Bug #9857, Bug #23032)
- **Incompatible Change:** Several changes were made to Performance Schema tables:
  - The `SETUP_OBJECTS` table was removed.
  - The `PROCESSLIST` table was renamed to `THREADS`.
  - The `EVENTS_WAITS_SUMMARY_BY_EVENT_NAME` table was renamed to `EVENTS_WAITS_SUMMARY_GLOBAL_BY_EVENT_NAME`.

  (Bug #55416)
- **Incompatible Change:** If the server was started with `character_set_server` set to `utf16`, it crashed during full-text stopword initialization. Now the stopword file is loaded and searched using `latin1` if `character_set_server` is `ucs2`, `utf16`, or `utf32`.

  If any table was created with `FULLTEXT` indexes while the server character set was `ucs2`, `utf16`, or `utf32`, it should be repaired using this statement:

  ```
  REPAIR TABLE tbl_name QUICK;
  ```

  (Bug #32391)
- **Important Change; Replication:** The `LOAD DATA INFILE` statement is now considered unsafe for statement-based replication. When using statement-based logging mode, the statement now produces a warning; when using mixed-format logging, the statement is made using the row-based format. (Bug #34283)
- **InnoDB:** The `Lock_time` field in the slow query log now reports a larger value, including the time for `InnoDB` lock waits at the statement level. (Bug #53496)
- **InnoDB:** The `InnoDB` storage engine was not included in the default installation when using the `configure` script. (Bug #55547)
- **InnoDB:** The `mysql_config` tool did not output the requirement for the `aio` library for `mysqld-libs`. (Bug #55215)
- **InnoDB:** Some memory used for `InnoDB` asynchronous I/O was not freed at shutdown. (Bug #54764)

- **InnoDB:** For an `InnoDB` table with an auto-increment column, the server could crash if the first statement that references the table after a server restart is a `SHOW CREATE TABLE` statement. (Bug #55277)
- **InnoDB:** The server could crash during the recovery phase of startup, if it previously crashed while inserting `BLOB` or other large columns that use off-page storage into an `InnoDB` table created with `ROW_FORMAT=REDUNDANT` or `ROW_FORMAT=COMPACT`. (Bug #54408)
- **InnoDB:** For an `InnoDB` table created with `ROW_FORMAT=COMPRESSED` or `ROW_FORMAT=DYNAMIC`, a query using the `READ UNCOMMITTED` isolation level could cause the server to stop with an assertion error, if `BLOB` or other large columns that use off-page storage were being inserted at the same time. (Bug #54358)
- **InnoDB:** For debug builds, the database server could crash when renaming a table that had active transactions. (Bug #54453)
- **InnoDB:** Performing large numbers of `RENAME TABLE` statements caused excessive memory use. (Bug #47991)
- **InnoDB:** Fast index creation in the `InnoDB Plugin` could fail, leaving the new secondary index corrupted. (Bug #54330)
- **InnoDB:** `InnoDB` could issue an incorrect message on startup, if tables were created under the setting `innodb_file_per_table=ON`. The message was of the form `InnoDB: Warning: allocated tablespace n, old maximum was 0`. If you encounter this message after upgrading, create an `InnoDB` table with `innodb_file_per_table = ON` and restart the server. The message should not be displayed any more. If you continue to encounter this message, or if you get it and haven't used a version without this fix, you might have corruption in your shared tablespace. If so, back up and reload your data. (Bug #54658)
- **InnoDB:** An `ALTER TABLE` statement could convert an `InnoDB` compressed table (with `row_format=compressed`) back to an uncompressed table (with `row_format=compact`). (Bug #54679)
- **InnoDB:** The output from the `SHOW ENGINE INNODB STATUS` command now includes information about “spin rounds” for RW-locks (both shared and exclusive locks). (Bug #54726)
- **InnoDB:** If a session executing `TRUNCATE TABLE` on an `InnoDB` table was killed during `open_tables()`, an assertion could be raised. (Bug #53757)
- **InnoDB:** `InnoDB` could not create tables that used the `utf32` character set. (Bug #52199)
- **InnoDB:** Misimplementation of the `os_fast_mutex_trylock()` function in `InnoDB` resulted in unnecessary blocking and reduced performance. (Bug #53204)
- **InnoDB:** Implementation of the 64-bit `dulint` structure in `InnoDB` was not optimized for 64-bit processors, resulting in excessive storage and reduced performance. (Bug #54728)
- **InnoDB:** An assertion was raised if (1) an `InnoDB` table was created using `CREATE TABLE ... SELECT` where the query used an `INFORMATION_SCHEMA` table and a view existed in the database; or (2) any statement that modified an `InnoDB` table had a subquery referencing an `INFORMATION_SCHEMA` table. (Bug #55973)
- **Partitioning:** An `ALTER TABLE ... ADD PARTITION` statement run concurrently with a read lock caused spurious `ER_TABLE_EXISTS_ERROR` and `ER_NO_SUCH_TABLE` errors on subsequent attempts. (Bug #53676)

  References: See also Bug #53770.

- **Partitioning:** `ALTER TABLE ... TRUNCATE PARTITION`, when called concurrently with transactional DML on the table, was executed immediately and did not wait for the concurrent transaction to release locks. As a result, the `ALTER TABLE` statement was written into the binary log before the DML statement, which led to replication failures when using row-based logging. (Bug #49907)

  References: See also Bug #42643.

- **Partitioning:** When the storage engine used to create a partitioned table was disabled, attempting to drop the table caused the server to crash. (Bug #46086)

- **Partitioning:** `INSERT ON DUPLICATE KEY UPDATE` statements performed poorly on tables having many partitions. The handler function for reading a row from a specific index was not optimized in the partitioning handler. (Bug #52455)

- **Partitioning:** `UPDATE` and `INSERT` statements affecting partitioned tables performed poorly when using row-based replication. (Bug #52517)

  References: This bug is a regression of Bug #39084.

- **Partitioning:** With `innodb_thread_concurrency = 1`, `ALTER TABLE ... REORGANIZE PARTITION` and `SELECT` could deadlock. There were unreleased latches in the `ALTER TABLE ... REORGANIZE PARTITION` thread which were needed by the `SELECT` thread to be able to continue. (Bug #54747)

- **Replication:** If the SQL thread was started while the I/O thread was performing rotation of the relay log, the two threads could begin to race for the same I/O cache, leading to a server crash. (Bug #54509)

  References: See also Bug #50364.

- **Replication:** The value of `binlog_direct_non_transactional_updates` had no effect on statements mixing transactional tables and nontransactional tables, or mixing temporary tables and nontransactional tables.

  As part of the fix for this issue, updates to temporary tables are now handled as transactional or nontransactional according to their storage engine types. (In effect, the current fix reverts a change made previously as part of the fix for Bug #53259.)

  In addition, unsafe mixed statements (that is, statements which access transactional table as well nontransactional or temporary tables, and write to any of them) are now handled as transactional when the statement-based logging format is in use. (Bug #53452)

  References: See also Bug #51894.

- **Replication:** A number of statements generated unnecessary warnings as potentially unsafe statements. (Due to the fix for Bug #51894, a temporary table is treated in this context as a transactional table, so that any mixed statement such as `t_innodb` + `t_myisam` or `t_temp` + `t_myisam` is flagged as unsafe.)

  To reduce the number of spurious warnings produced when this happened, some of the criteria used to classify a statements as safe or unsafe have been changed. For more information about handling of mixed statements, see Transactional, nontransactional, and mixed statements. (Bug #53259)

  References: See also Bug #53452, Bug #54872.

- **Replication:** When `binlog_format=STATEMENT`, any statement that is flagged as being unsafe, possibly causing the slave to go out of sync, generates a warning. This warning is written to the server log, the warning count is returned to the client in the server's response, and the warnings are accessible through `SHOW WARNINGS`.

The current bug affects only the counts for warnings to the client and that are visible through `SHOW WARNINGS`; it does not affect which warnings are written to the log. The current issue came about because the fix for an earlier issue caused warnings for substatements to be cleared whenever a new substatement was started. However, this suppressed warnings for unsafe statements in some cases. Now, such warnings are no longer cleared. (Bug #50312)

References: This bug was introduced by Bug #36649.

- **Replication:** When using the row-based logging format, a failed `CREATE TABLE ... SELECT` statement was written to the binary log, causing replication to break if the failed statement was later re-run on the master. In such cases, a `DROP TABLE ... IF EXIST` statement is now logged in the event that a `CREATE TABLE ... SELECT` fails. (Bug #55625)
- **Replication:** When using the row-based logging format, a `SET PASSWORD` statement was written to the binary log twice. (Bug #55452)
- **Replication:** When closing temporary tables, after the session connection was already closed, if the writing of the implicit `DROP TABLE` statement into the binary log failed, it was possible for the resulting error to be mishandled, triggering an assertion. (Bug #55387)
- **Replication:** Executing `SHOW BINLOG EVENTS` increased the value of `max_allowed_packet` applying to the session that executed the statement. (Bug #55322)
- **Replication:** Setting `binlog_format` to `ROW`, then creating and dropping a temporary table led to an assertion. (Bug #54925)
- **Replication:** When using mixed-format replication, changes made to a nontransactional temporary table within a transaction were not written into the binary log when the transaction was rolled back. This could lead to a failure in replication if the temporary table was used again afterwards. (Bug #54872)

  References: See also Bug #53259.
- **Replication:** If `binlog_format` was explicitly switched from `STATEMENT` to `ROW` following the creation of a temporary table, then on disconnection the master failed to write the expected `DROP TEMPORARY TABLE` statement into the binary log. As a consequence, temporary tables (and their corresponding files) accumulated as this scenario was repeated. (Bug #54842)

  References: See also Bug #52616.
- **Replication:** Two related issues involving temporary tables and transactions were introduced by a fix made in MySQL 5.1.37:

  1. When a temporary table was created or dropped within a transaction, any failed statement that following the `CREATE TEMPORARY TABLE` or `DROP TEMPORARY TABLE` statement triggered a rollback, which caused the slave to diverge from the master.
  2. When a `CREATE TEMPORARY TABLE ... SELECT * FROM ...` statement was executed within a transaction in which only tables using transactional storage engines were used and the transaction was rolled back at the end, the changes—including the creation of the temporary table—were not written to the binary log.

  The current fix restores the correct behavior in both of these cases. (Bug #53560)

  References: This bug was introduced by Bug #43929.
- **Replication:** When `CURRENT_USER()` or `CURRENT_USER` was used to supply the name and host of the affected user or of the definer in any of the statements `DROP USER`, `RENAME USER`, `GRANT`, `REVOKE`,

and `ALTER EVENT`, the reference to `CURRENT_USER()` or `CURRENT_USER` was not expanded when written to the binary log. This resulted in `CURRENT_USER()` or `CURRENT_USER` being expanded to the user and host of the slave SQL thread on the slave, thus breaking replication. Now `CURRENT_USER()` and `CURRENT_USER` are expanded prior to being written to the binary log in such cases, so that the correct user and host are referenced on both the master and the slave. (Bug #48321)

- **Replication:** Replication could break if a transaction involving both transactional and nontransactional tables was rolled back to a savepoint. It broke if a concurrent connection tried to drop a transactional table which was locked after the savepoint was set. This `DROP TABLE` completed when `ROLLBACK TO SAVEPOINT` was executed because the lock on the table was dropped by the transaction. When the slave later tried to apply the binary log events, it failed because the table had already been dropped. (Bug #50124)

- For the general query log and slow query log, logging to tables incurred excessive overhead beginning with MySQL 5.1.21. This overhead has been eliminated. (Bug #11747038, Bug #30414)

  References: See also Bug #29129.

- The `PERFORMANCE_SCHEMA` database was not correctly created and populated on Windows. (Bug #52809)

- After an RPM installation, `mysqld` would be started with the `root` user, rather than the `mysql` user. (Bug #56574)

- When upgrading an existing install with an RPM on Linux, the MySQL server might not have been restarted properly. This was due to a naming conflict when upgrading from a `community` named RPM. Previous installations are now correctly removed, the MySQL initialization script is recreated, and the MySQL server is restarted as normal. (Bug #55015)

- `CMake` produced bad dependencies for the `sql/lex_hash.h` file during configuration. (Bug #55842)

- The default compiler options for Mac OS X 10.5 were set incorrectly. (Bug #55601)

- The Windows MSI installer failed during installation to preserve custom settings, such as the configured data directory. (Bug #55169)

- The optimization method of the `ARCHIVE` storage engine did not preserve the `.frm` file embedded in the `.ARZ` file when rewriting the `.ARZ` file for optimization. This meant an `ARCHIVE` table that had been optimized could not be discovered.

  The `ARCHIVE` engine stores the `.frm` file in the `.ARZ` file so it can be transferred from machine to machine without also needing to copy the `.frm` file. The engine subsequently restores the embedded `.frm` during discovery. (Bug #45377)

- The `-features=no%except` option was missing from the build for Solaris/x86. (Bug #55250)

- Building MySQL on Solaris 8 x86 failed when using Sun Studio due to `gcc` inline assembly code. (Bug #55061)

- The ABI check for MySQL failed to compile with `gcc` 4.5. (Bug #52514)

- Builds of MySQL generated a large number of warnings. (Bug #53445)

- The `mysql-debug.pdb` supplied with releases did not match the corresponding `mysqld.exe`. (Bug #52850)

- Reading from a temporary `MERGE` table, with two nontemporary child `MyISAM` tables, resulted in the error:

```
ERROR 1168 (HY000): Unable to open underlying table which is differently
defined or of non-MyISAM type or doesn't exist
```

(Bug #36171)

- An assertion occurred in `ha_myisammrg.cc` line 1137:

```
  DBUG_ASSERT(this->file->children_attached);
```

  The problem was found while running RQG tests and the assertion occurred during `REPAIR`, `OPTIMIZE`, and `ANALYZE` operations. (Bug #47633)

- Performance Schema header files were not installed in the correct directory. (Bug #53255)
- Attempts to access a nonexistent Performance Schema table resulted in a misleading error message. (Bug #52586)
- The Performance Schema could enter an infinite loop if required to create a large number of mutex instances. (Bug #52502)
- The server could crash when processing subqueries with empty results. (Bug #53236)
- The `large_pages` system variable was tied to the `--large-files` command-line option, not the `--large-pages` option. (Bug #52716)
- Missing Performance Schema tables were not reported in the error log at server startup. (Bug #53617)
- `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` underreported the amount of memory allocated by the Performance Schema. (Bug #53566)
- A client with automatic reconnection enabled saw the error message `Lost connection to MySQL server during query` if the connection was lost between the `mysql_stmt_prepare()` and `mysql_stmt_execute()` C API functions. However, `mysql_stmt_errno()` returned 0, not the corresponding error number 2013. (Bug #53899)
- A client could supply data in chunks to a prepared statement parameter other than of type `TEXT` or `BLOB` using the `mysql_stmt_send_long_data()` C API function (or `COM_STMT_SEND_LONG_DATA` command). This led to a crash because other data types are not valid for long data. (Bug #54041)
- The server failed to disregard sort order for some zero-length tuples, leading to an assertion failure. (Bug #54459)
- If a session tried to drop a database containing a table opened with `HANDLER` in another session, any `DATABASE` statement (`CREATE`, `DROP`, `ALTER`) executed by that session produced a deadlock. (Bug #54360)
- Sort-`index_merge` for join tables other than the first table used excessive memory. (Bug #41660)
- The results of some `ORDER BY ... DESC` queries were sorted incorrectly. (Bug #51431)
- `Index Merge` between three indexes could return incorrect results. (Bug #50389)
- A signal-handler redefinition for `SIGUSR1` was removed. The redefinition could cause the server to encounter a kernel deadlock on Solaris when there are many active threads. Other POSIX platforms might also be affected. (Bug #54667)

- Assignments of the `PASSWORD()` or `OLD_PASSWORD()` function to a user variable did not preserve the character set of the function return value. (Bug #54668)
- Use of `uint` in `typelib.h` caused compilation problems in Windows. This was changed to `unsigned int`. (Bug #52959)
- `mysql_secure_installation` did not properly identify local accounts and could incorrectly remove nonlocal `root` accounts. (Bug #54004)
- In a slave SQL thread or Event Scheduler thread, the `SLEEP()` function could not sleep more than five seconds. (Bug #54729)
- `mysql_secure_installation` sometimes failed to locate the `mysql` client. (Bug #52274)
- `MIN()` and `MAX()` returned incorrect results for `DATE` columns if the set of values included `'0000-00-00'`. (Bug #49771)
- If a symbolic link was used in a file path name, the Performance Schema did not resolve all file I/O events to the same name. (Bug #52134)
- Performance Schema event collection for a thread could “leak” from one connection to another if the thread was used for one connection, then cached, then reused for another connection. (Bug #54782)
- The Performance Schema displayed spurious startup error messages when the server was run in bootstrap mode. (Bug #54467)
- Statements of the form `UPDATE ... WHERE ... ORDER BY` used a `filesort` even when not required. (Bug #36569)

  References: See also Bug #53737, Bug #53742.
- The patch for Bug #36569 caused performance regressions and incorrect execution of some `UPDATE` statements. (Bug #53737, Bug #53742)
- `INFORMATION_SCHEMA.ENGINES` and `SHOW ENGINES` described `MyISAM` as the default storage engine, but this is not true as of MySQL 5.5.5. (Bug #54832)
- `INFORMATION_SCHEMA.COLUMNS` reported incorrect precision for `BIGINT UNSIGNED` columns. (Bug #53814)
- Reading a `ucs2` data file with `LOAD DATA INFILE` was subject to three problems. 1) Incorrect parsing of the file as `ucs2` data, resulting in incorrect length of the parsed string. This is fixed by truncating the invalid trailing bytes (incomplete multi-byte characters) when reading from the file. 2) Reads from a proper `ucs2` file did not recognize newline characters. This is fixed by first checking whether a byte is a newline (or any other special character) before reading it as a part of a multi-byte character. 3) When using user variables to hold column data, the character set of the user variable was set incorrectly to the database charset. This is fixed by setting it to the character set specified in the `LOAD DATA INFILE` statement, if any. (Bug #51876)
- Incorrect error handling could result in an `OPTIMIZE TABLE` crash. (Bug #54783)
- With `lower_case_table_names` set to a nonzero value, searches for table or database names in `INFORMATION_SCHEMA` tables could produce incorrect results. (Bug #53095)
- In debug builds, an assertion could be raised when the server tried to send an OK packet to the client after having failed to detect errors during processing of the `WHERE` condition of an `UPDATE` statement. (Bug #54734)

- Some queries involving `GROUP BY` and a function that returned `DATE` raised a debug assertion. (Bug #52159)
- `SET sql_select_limit = 0` did not work. (Bug #54682)
- `mysql_upgrade` could incorrectly remove `TRIGGER` privileges. (Bug #53613)
- `safemalloc` was excessively slow under certain conditions and has been removed. The `--skip-safemalloc` server option has also been removed, and the `--with-debug=full` configuration option is no different from `--with-debug`. (Bug #34043)
- Queries that named view columns in a `GROUP BY` clause could cause a server crash. (Bug #54515)
- Portability problems in `SHOW STATUS` could lead to incorrect results on some platforms. (Bug #53493)
- A debugging assertion could be raised after a write failure to a closed socket. (Bug #42496)
- Searches in `INFORMATION_SCHEMA` tables for rows matching a nonexistent database produced an error instead of an empty query result. (Bug #49542)
- `GROUP BY` operations used `max_sort_length` inconsistently. (Bug #55188)
- The `my_like_range_xxx()` functions returned badly formed maximum strings for Asian character sets, which caused problems for storage engines. (Bug #45012)
- Rows inserted in a table by one session were not immediately visible to another session that queried the table, even if the insert had committed. (Bug #37521)
- `mysqld_safe` contained a syntax error that prevented it from restarting the server. (Bug #54991)
- The `MERGE` storage engine tried to use memory mapping on the underlying `MyISAM` tables even on platforms that do not support it and even when `myisam_use_mmap` was disabled. This led to a hang for `INSERT INTO ... SELECT FROM` statements that selected from a `MyISAM` table into a `MERGE` table that contained the same `MyISAM` table. (Bug #54811, Bug #50788)
- `DROP DATABASE` failed if there was a `TEMPORARY` table with the same name as a non-`TEMPORARY` table in the database. (Bug #48067)
- `SHOW CREATE TRIGGER` took a stronger metadata lock than required. This caused the statement to be blocked unnecessarily. For example, `LOCK TABLES ... WRITE` in one session blocked `SHOW CREATE TRIGGER` in another session.

  Also, a `SHOW CREATE TRIGGER` statement issued inside a transaction did not release its metadata locks at the end of statement execution. Consequently, `SHOW CREATE TRIGGER` was able to block other sessions from accessing the table (for example, using `ALTER TABLE`). (Bug #55498)
- A `SHOW CREATE TABLE` statement issued inside a transaction did not release its metadata locks at the end of statement execution. Consequently, `SHOW CREATE TABLE` was able to block other sessions from accessing the table (for example, using `ALTER TABLE`). (Bug #54905)
- `XA START` had a race condition that could cause a server crash. (Bug #51855)
- A single-table `DELETE` ordered by a column that had a hash-type index could raise an assertion or cause a server crash. (Bug #55472)
- Errors during processing of `WHERE` conditions in `HANDLER ... READ` statements were not detected, so the handler code still tried to send EOF to the client, raising an assertion. (Bug #54401)

- A statement that was aborted by `KILL QUERY` while it waited on a metadata lock could raise an assertion in debug builds, or send OK to the client instead of `ER_QUERY_INTERRUPTED` in regular builds. (Bug #55223)
- `mysql_upgrade` did not handle the `--ssl` option properly. (Bug #55672)
- Using `MIN()` or `MAX()` on a column containing the maximum `TIME` value caused a server crash. (Bug #55648)
- Enumeration plugin variables were subject to a type-casting error, causing inconsistent results between different platforms. (Bug #42144)
- `INSERT IGNORE INTO ... SELECT` statements could raise a debug assertion. (Bug #54106)
- The server was not checking for errors generated during the execution of `Item::val_xxx()` methods when copying data to a group, order, or distinct temp table's row. (Bug #55580)
- `ORDER BY` clauses that included user-variable expressions could raise a debug assertion. (Bug #55565)
- For distributions built with `CMake` rather than the GNU autotools, `mysql` lacked `pager` support, and some scripts were built without the execute bit set. (Bug #54466, Bug #54129)
- The `thread_concurrency` system variable was unavailable on non-Solaris systems. (Bug #55001)
- A call to `mysql_library_init()` following a call to `mysql_library_end()` caused a client crash. (Bug #55345)
- If audit plugins were installed that were interested in `MYSQL_AUDIT_GENERAL_CLASS` events and the general query log was disabled, failed `INSTALL PLUGIN` or `UNINSTALL PLUGIN` statements caused a server crash. (Bug #54989)
- Incorrect handling of user variable assignments as subexpressions could lead to incorrect results or server crashes. (Bug #55615)
- `PARTITION BY KEY` on a `utf32` `ENUM` column raised a debugging assertion. (Bug #52121, Bug #11759782)
- With `character_set_connection` set to `utf16` or `utf32`, `CREATE TABLE t1 AS SELECT HEX() ...` caused a server crash. (Bug #45263)
- `FLUSH TABLES WITH READ LOCK` in one session and `FLUSH TABLES tbl_list WITH READ LOCK` in another session were mutually exclusive.

  This bug fix involved several changes to the states displayed by `SHOW PROCESSLIST`:

  - `Table lock` was replaced with `Waiting for table level lock`.
  - `Waiting for table` was replaced with `Waiting for table flush`.
  - These states are new: `Waiting for global metadata lock`, `Waiting for schema metadata lock`, `Waiting for stored function metadata lock`, `Waiting for stored procedure metadata lock`, `Waiting for table metadata lock`.

  (Bug #52044)
- A pending `FLUSH TABLES tbl_list WITH READ LOCK` statement unnecessarily aborted transactions. (Bug #52117)
- `IF()` with a subquery argument could raise a debug assertion for debug builds under some circumstances. (Bug #55077)

- Deadlocks involving `INSERT DELAYED` statements were not detected. The server could crash if the delayed handler thread was killed due to a conflicting shared metadata lock. (Bug #54332)
- `SHOW CREATE EVENT` released all metadata locks held by the current transaction. This invalidated any existing savepoints and raised an assertion if `ROLLBACK TO SAVEPOINT` was executed. (Bug #54105)
- Some functions did not calculate their `max_length` metadata value correctly. (Bug #54916)
- The embedded server raised an assertion when it attempted to load plugins. (Bug #56085)
- `FORMAT()` did not respect the decimal point character if the locale was changed and always returned an ASCII value. (Bug #55912)
- `DROP TABLE` held a lock during `unlink()` file system operations, causing performance problems if `unlink()` took a long time. (Bug #41158)
- After `ALTER TABLE` was used on a temporary transactional table locked by `LOCK TABLES`, any later attempts to execute `LOCK TABLES` or `UNLOCK TABLES` caused a server crash. (Bug #54117)
- `InnoDB` produced no warning at startup about illegal `innodb_file_format_check` values. (Bug #55095)
- A join with an aggregated function and impossible `WHERE` condition returned an extra row. (Bug #54416)
- Threads that were calculating the estimated number of records for a range scan did not respond to the `KILL` statement. That is, if a `range` join type is possible (even if not selected by the optimizer as a join type of choice and thus not shown by `EXPLAIN`), the query in the `statistics` state (shown by the `SHOW PROCESSLIST`) did not respond to the `KILL` statement. (Bug #25421)
- Problems in the atomic operations implementation could lead to server crashes. (Bug #22320, Bug #52261)

Changes in MySQL 5.5.5 (2010-07-06)

icc Notes

- This is the final release of MySQL 5.5 for which Generic Linux MySQL binary packages built with the `icc` compiler on x86 and x86_64 will be offered. These were previously produced as an alternative to our main packages built using `gcc`, as they provided noticeable performance benefits. In recent times the performance differences have diminished and build and runtime problems have surfaced, thus it is no longer viable to continue producing them.

  We continue to use the `icc` compiler to produce our distribution-specific RPM packages on ia64.

InnoDB Notes

- `InnoDB` has been upgraded to version 1.1.1. This version is considered of “early adopter” quality.

  `InnoDB` is now the default storage engine, rather than `MyISAM`, in the regular and enterprise versions of MySQL. This change has the following consequences:

  - Existing tables are not affected by this change, only new tables that are created.
  - Some of the `InnoDB` option settings also change, so that the default configuration represents the best practices for `InnoDB` functionality, reliability, and file management: `innodb_file_format=Barracuda` rather than `Antelope`, `innodb_strict_mode=ON` rather than `OFF`, and `innodb_file_per_table=ON` rather than `OFF`.

- The system tables remain in MyISAM format.
- MyISAM remains the default storage engine for the embedded version of MySQL.

Follow these steps to ensure a smooth transition when upgrading:

- Familiarize yourself with the new default setting for the InnoDB file-per-table option, which creates a separate .ibd file for each user table. Adapt any backup procedure to include these files. For details, see InnoDB File-Per-Table Mode.
- Test the installation and operation for any applications that you run on the database server, to determine if they use any features specific to MyISAM that cause problems during installation (when the tables are created) or at runtime (when MyISAM-specific features might fail, or reliance on MyISAM settings for performance might become apparent). The InnoDB "strict" mode might also alert you to problems while setting up tables for an application.
- As a preliminary test for individual tables rather than an entire application, you can use the statement ALTER TABLE *table_name* ENGINE=INNODB; to convert an existing table to use the InnoDB storage engine, and then run compatibility and performance tests.
- Where necessary, add ENGINE=MYISAM clauses to CREATE TABLE statements, for tables that require features specific to MyISAM, such as full-text search.
- Benchmark the most important queries, to check whether you need to make changes to the table indexes.
- Measure the performance of applications under typical load, to check whether you need to change any additional InnoDB configuration settings.
- As a last resort, if a database server is devoted entirely to applications that can only run with MyISAM tables, you could add a default-storage-engine line in the configuration file, or a --default-storage-engine option in the database server startup command, to re-enable MyISAM as the default storage engine for that server. For details about setting the default storage engine, see Setting the Storage Engine.

Functionality Added or Changed

- **Incompatible Change:** All numeric operators and functions on integer, floating-point and DECIMAL values now throw an "out of range" error (ER_DATA_OUT_OF_RANGE) rather than returning an incorrect value or NULL, when the result is out of the supported range for the corresponding data type. See Out-of-Range and Overflow Handling. (Bug #8433)
- **InnoDB:** The INFORMATION_SCHEMA.INNODB_TRX table now includes a number of fields that duplicate information from the SHOW ENGINE INNODB STATUS output. You no longer need to parse that output to get complete transaction information. (Bug #53336)
- **InnoDB:** InnoDB stores redo log records in a hash table during recovery. On 64-bit systems, this hash table was 1/8 of the buffer pool size. To reduce memory usage, the dimension of the hash table was reduced to 1/64 of the buffer pool size (or 1/128 on 32-bit systems). (Bug #53122)
- The deprecated mysql_fix_privilege_tables script has been removed. (Bug #42589)
- A new system variable, skip_name_resolve, is set from the value of the --skip-name-resolve server option. This provides a way to determine at runtime whether the server uses name resolution for client connections. (Bug #37168)

- For events of MYSQL_AUDIT_GENERAL_CLASS, the event subclass was not passed to audit plugins even though the server passed the subclass to the plugin handler. The subclass is now available through the following changes:

  - The struct mysql_event_general structure has a new event_subclass member.

  - The new member changes the interface, so the audit plugin interface version, MYSQL_AUDIT_INTERFACE_VERSION, has been incremented from 0x0100 to 0x0200. Plugins that require access to the new member must be recompiled to use version 0x0200 or higher.

  The NULL_AUDIT example plugin in the plugin/audit_null directory has been modified to count events of each subclass, based on the event_subclass value. See Writing Audit Plugins. (Bug #47059)

- The Rows_examined value in slow query log rows now is nonzero for UPDATE and DELETE statements that modify rows. (Bug #49756)

- Previously, the innodb_file_format_check system variable served a dual purpose. Setting it at server startup would keep InnoDB from starting if any tables used a more recent file format than supported by the current level of InnoDB. If InnoDB could start, the same system variable was set to the "highest" file format value used by any InnoDB table in the database. Thus, its value could change from the value you specified.

  Now, checking and recording the file format tag are handled using separate variables. innodb_file_format_check can be set to 1 or 0 at server startup to enable or disable whether InnoDB checks the file format tag in the system tablespace. If the tag is checked and is higher than that supported by the current version of InnoDB, an error occurs and InnoDB does not start. If the tag is not higher, InnoDB sets the value of innodb_file_format_max to the file format tag.

  For background information about InnoDB file-format management, see InnoDB File-Format Management. (Bug #49792, Bug #53654)

- Added the SHA2() function, which calculates the SHA-2 family of hash functions (SHA-224, SHA-256, SHA-384, and SHA-512). (Contributed by Bill Karwin) (Bug #13174)

- It is now possible to build MySQL on all platforms using CMake instead of the GNU autotools. (Prior to MySQL 5.5.5, CMake support was limited to Windows.) For instructions on using CMake to build MySQL, see Installing MySQL from Source.

- Windows MSI package installers create and set up the data directory that the installed server will use, but now also create a pristine "template" data directory named data under the installation directory. This directory can be useful when the machine will be used to run multiple instances of MySQL: After an installation has been performed using an MSI package, the template data directory can be copied to set up additional MySQL instances. See Running Multiple MySQL Instances on One Machine.

Bugs Fixed

- **Security Fix:** Privilege checking for UNINSTALL PLUGIN was incorrect. (Bug #51770, CVE-2010-1621)

- **Security Fix:** The server could be tricked into reading packets indefinitely if it received a packet larger than the maximum size of one packet. (Bug #50974, CVE-2010-1849)

- **Security Fix:** The server failed to check the table name argument of a COM_FIELD_LIST command packet for validity and compliance to acceptable table name standards. This could be exploited to bypass almost all forms of checks for privileges and table-level grants by providing a specially crafted table name argument to COM_FIELD_LIST.

In MySQL 5.0 and above, this permitted an authenticated user with `SELECT` privileges on one table to obtain the field definitions of any table in all other databases and potentially of other MySQL instances accessible from the server's file system.

Additionally, for MySQL version 5.1 and above, an authenticated user with `DELETE` or `SELECT` privileges on one table could delete or read content from any other table in all databases on this server, and potentially of other MySQL instances accessible from the server's file system. (Bug #53371, CVE-2010-1848)

- **Security Fix:** The server was susceptible to a buffer-overflow attack due to a failure to perform bounds checking on the table name argument of a `COM_FIELD_LIST` command packet. By sending long data for the table name, a buffer is overflown, which could be exploited by an authenticated user to inject malicious code. (Bug #53237, CVE-2010-1850)
- **Security Fix:** `LOAD DATA INFILE` did not check for SQL errors and sent an OK packet even when errors were already reported. Also, an assert related to client/server protocol checking in debug servers sometimes was raised when it should not have been. (Bug #52512, CVE-2010-3683)
- **Security Fix:** The server could crash if there were alternate reads from two indexes on a table using the `HANDLER` interface. (Bug #54007, CVE-2010-3681)
- **Security Fix:** A security bug was fixed. (Bug #52315)
- **Security Fix:** A security bug was fixed. (Bug #53933)
- **Security Fix:** A security bug was fixed. (Bug #53907)
- **Security Fix:** A security bug was fixed. (Bug #48157)
- **Security Fix:** A security bug was fixed. (Bug #52357)
- **Performance; InnoDB:** Deadlock detection could be a bottleneck in `InnoDB` processing, if many transactions attempted to update the same row simultaneously. The algorithm has been improved to enhance performance and scalability, in the InnoDB Plugin for MySQL 5.1, and in InnoDB 1.1 for MySQL 5.5. (Bug #49047)
- **Performance:** While looking for the shortest index for a covering index scan, the optimizer did not consider the full row length for a clustered primary key, as in `InnoDB`. Secondary covering indexes are now preferred, making full table scans less likely. (Bug #39653)

  References: See also Bug #55656.
- **Incompatible Change:** `TRUNCATE TABLE` did not take an exclusive lock on a table if truncation was done by deleting all rows in the table. For `InnoDB` tables, this could break proper isolation because `InnoDB` ended up aborting some granted locks when truncating a table. Now an exclusive metadata lock is taken before `TRUNCATE TABLE` can proceed. This guarantees that no other transaction is using the table.

  Incompatible change: Truncation using delete no longer fails if `sql_safe_updates` is enabled (this was an undocumented side effect). (Bug #42643)
- **Incompatible Change:** After `SET TRANSACTION ISOLATION LEVEL` to set the isolation level for the next transaction, the session value of the `tx_isolation` system variable could appear to change to the transaction isolation level after completion of statements within the transaction. Now the current transaction isolation level is now established at transaction start. If there was a `SET TRANSACTION ISOLATION LEVEL statement`, the value is taken from it. Otherwise, the session `tx_isolation` value is used. A change in the session value while a transaction is active is still permitted, but no longer

affects the current transaction isolation level. This is an incompatible change. A change in the session isolation level made while there is no active transaction overrides a `SET TRANSACTION ISOLATION LEVEL` statement, if there was any. (Bug #20837)

- **Important Change; Replication:** It was possible to set `sql_log_bin` with session scope inside a transaction or subquery. (Bug #53437)
- **Important Change; Replication:** When invoked, `CHANGE MASTER TO` and `SET GLOBAL sql_slave_skip_counter` now cause information to be written to the error log about the slave's state prior to execution of the statement. For `CHANGE MASTER TO`, this information includes the previous values of `MASTER_HOST`, `MASTER_PORT`, `MASTER_LOG_FILE`, and `MASTER_LOG_POS`. For `SET GLOBAL sql_slave_skip_counter`, this information includes the previous values of `RELAY_LOG_FILE`, `RELAY_LOG_POS`, and `sql_slave_skip_counter`. (Bug #43406, Bug #43407)
- **Important Change; Replication:** When changing `binlog_format` or `binlog_direct_non_transactional_updates`, permissions were not checked prior to checking the scope and context of the variable being changed.

  As a result of this fix, an error is no longer reported when—in the context of a transaction or a stored function—you try to set a value for a session variable that is the same as its previous value, or for a variable whose scope is global only. (Bug #51277)
- **Important Change:** When using fast `ALTER TABLE`, different internal ordering of indexes in the MySQL optimizer and the `InnoDB` storage engine could cause error messages about possibly mixed up `.frm` files and incorrect index use. (Bug #47622)
- **InnoDB; Replication:** Reading from a table that used a self-logging storage engine and updating a table that used a transactional engine (such as `InnoDB`) generated changes that were written to the binary log using statement format which could make slaves diverge. However, when using mixed logging format, such changes should be written to the binary log using row format. (This issue did not occur when reading from tables using a self-logging engine and updating `MyISAM` tables, as this was already handled by checking for combinations of nontransactional and transactional engines.) Now such statements are classified as unsafe, and in mixed mode, cause a switch to row-based logging. (Bug #49019)
- **InnoDB; Replication:** `TRUNCATE TABLE` performed on a temporary table using the `InnoDB` storage engine was logged even when using row-based mode. (Bug #51251)
- **InnoDB:** The server could crash during shutdown, if started with the option `--innodb_use_sys_malloc=0`. (Bug #52546)
- **InnoDB:** Some combinations of `SELECT` and `SELECT FOR UPDATE` statements could fail with errors about locks, or incorrectly release a row lock during a semi-consistent read operation. (Bug #53674)
- **InnoDB:** The server could crash with a message `InnoDB: Assertion failure in thread` *nnnn*, typically during shutdown on a Windows system. (Bug #53947)
- **InnoDB:** The values of `innodb_buffer_pool_pages_total` and `innodb_buffer_pool_pages_misc` in the `information_schema.global_status` table could be computed incorrectly. (Bug #52983)
- **InnoDB:** Multi-statement execution could fail with an error about foreign key constraints. This problem could affect calls to `mysql_query()` and `mysql_real_query()`, and `CALL` statements that invoke stored procedures. (Bug #48024)
- **InnoDB:** Fixed a checksum error reported for compressed tables when the `--innodb_checksums` option is enabled. Although the message stated that the table was corrupted, the table is actually fine. (Bug #53248)

- **InnoDB:** `InnoDB` checks to see whether a row could possibly exceed the maximum size if all columns are fully used. This produced `Row size too large` errors for some tables that could be created with the built-in `InnoDB` from older MySQL versions. Now the check is only done when `innodb_strict_mode` is enabled or if the table is dynamic or compressed. (Bug #50495)

- **InnoDB:** Adding a unique key on multiple columns, where one of the columns is `NULL`, could mistakenly report duplicate key errors. (Bug #53290)

- **InnoDB:** A mismatch between index information maintained within the `.frm` files and the corresponding information in the InnoDB system tablespace could produce this error: `[ERROR] Index index of table has n columns unique inside InnoDB, but MySQL is asking statistics for m columns. Have you mixed up .frm files from different installations?` (Bug #44571)

- **InnoDB:** Connections waiting for an `InnoDB` row lock ignored `KILL` until the row lock wait ended. Now, `KILL` during lock wait results in “query interrupted” instead of “lock wait timeout exceeded”. The corresponding transaction is rolled back. (Bug #51920)

- **InnoDB:** An overly strict assertion could fail during the purge of delete-marked records in `DYNAMIC` or `COMPRESSED` `InnoDB` tables that contain column prefix indexes. (Bug #52746)

- **InnoDB:** `InnoDB` attempted to choose off-page storage without ensuring that there was an “off-page storage” flag in the record header. To correct this, in `DYNAMIC` and `COMPRESSED` formats, `InnoDB` stores locally any non-`BLOB` columns having a maximum length not exceeding 256 bytes. This is because there is no room for the “external storage” flag when the maximum length is 255 bytes or less. This restriction trivially holds in `REDUNDANT` and `COMPACT` formats, because there `InnoDB` always stores locally columns having a length up to `local_len` = 788 bytes. (Bug #52745)

- **InnoDB:** `InnoDB` page splitting could enter an infinite loop for compressed tables. (Bug #52964)

- **InnoDB:** When reporting a foreign key constraint violation during `INSERT`, `InnoDB` could display uninitialized data for the `DB_TRX_ID` and `DB_ROLL_PTR` system columns. (Bug #53202)

- **Partitioning; Replication:** Attempting to execute `LOAD DATA` on a partitioned `MyISAM` table while using statement-based logging mode caused the master to hang or crash. (Bug #51851)

- **Partitioning; Replication:** The `NO_DIR_IN_CREATE` server SQL mode was not enforced when defining subpartitions. In certain cases, this could lead to failures on replication slaves. (Bug #42954)

- **Partitioning:** Rows inserted into a table created using a `PARTITION BY LIST COLUMNS` option referencing multiple columns could be inserted into the wrong partition. (Bug #52815)

- **Partitioning:** When attempting to perform DDL on a partitioned table and the table's `.par` file could not be found, the server returned the inaccurate error message `Out of memory; restart server and try again (needed 2 bytes)`. Now in such cases, the server returns the error `Failed to initialize partitions from .par file`. (Bug #49161)

- **Partitioning:** `ALTER TABLE` statements that cause table partitions to be renamed or dropped (such as `ALTER TABLE ... ADD PARTITION`, `ALTER TABLE ... DROP PARTITION`, and `ALTER TABLE ... REORGANIZE PARTITION`) — when run concurrently with queries against the `INFORMATION_SCHEMA.PARTITIONS` table — could fail, cause the affected partitioned tables to become unusable, or both. This was due to the fact that the `INFORMATION_SCHEMA` database ignored the name lock imposed by the `ALTER TABLE` statement on the partitions affected. In particular, this led to problems with `InnoDB` tables, because `InnoDB` would accept the rename operation, but put it in a background queue, so that subsequent rename operations failed when `InnoDB` was unable to find the correct partition. Now, `INFORMATION_SCHEMA` honors name locks imposed by ongoing `ALTER TABLE` statements that cause partitions to be renamed or dropped. (Bug #50561)

References: See also Bug #47343, Bug #45808.

- **Partitioning:** It was possible to execute a `CREATE TEMPORARY TABLE tmp LIKE pt` statement, where `pt` is a partitioned table, even though partitioned temporary tables are not permitted. This caused the server to crash. Now a check is performed to prevent such statements from being executed. (Bug #49477)

- **Partitioning:** Partition pruning on `RANGE` partitioned tables did not always work correctly; the last partition was not excluded if the range was beyond it (when not using `MAXVALUE`). Now the last partition is not included if the partitioning function value is not within the range. (Bug #51830)

- **Partitioning:** Attempting to partition a table using a `DECIMAL` column caused the server to crash; this was not supported and is now specifically not permitted. (Bug #51347)

- **Partitioning:** The `insert_id` server system variable was not reset following an insert that failed on a partitioned `MyISAM` table having an `AUTO_INCREMENT` column. (Bug #50392)

- **Partitioning:** Foreign keys are not supported on partitioned tables. However, it was possible using an `ALTER TABLE` statement to set a foreign key on a partitioned table; it was also possible to partition a table with a single foreign key. (Bug #50104)

- **Partitioning:** `GROUP BY` queries performed poorly for some partitioned tables. This was due to the block size not being set for partitioned tables, thus the keys per block was not correct, which could cause such queries to be optimized incorrectly. (Bug #48229)

  References: See also Bug #37252.

- **Partitioning:** `REPAIR TABLE` failed for partitioned `ARCHIVE` tables. (Bug #46565)

- **Replication:** When using unique keys on `NULL` columns in row-based replication, the slave sometimes chose the wrong row when performing an update. This happened because a table having a unique key on such a column could have multiple rows containing `NULL` for the column used by the unique key, and the slave merely picked the first row containing `NULL` in that column. (Bug #53893)

- **Replication:** When a `CREATE TEMPORARY TABLE ... SELECT` statement was executed within a transaction that updated only transactional engines and was later rolled back (for example, due to a deadlock) the changes—including the creation of the temporary table—were not written to the binary log, which caused subsequent updates to this table to fail on the slave. (Bug #53421)

- **Replication:** DDL statements that lock tables (such as `ALTER TABLE`, `CREATE INDEX`, and `CREATE TRIGGER`) caused spurious `ER_BINLOG_ROW_MODE_AND_STMT_ENGINE` or `ER_BINLOG_STMT_MODE_AND_ROW_ENGINE` errors, even though they did not insert rows into any tables.

  

  Note

  The error `ER_BINLOG_ROW_MODE_AND_STMT_ENGINE` is generated when `binlog_format=ROW` and a statement modifies a table restricted to statement-based logging; `ER_BINLOG_STMT_MODE_AND_ROW_ENGINE` is generated when `binlog_format=STATEMENT` and a statement modifies a table restricted to row-based logging.

  (Bug #50479)

  References: This bug was introduced by Bug #39934, Bug #11749859.

- **Replication:** In some cases, attempting to update a column with a value of an incompatible type resulted in a mismatch between master and slave because the column value was set to its implicit default value on the master (as expected), but the same column on the slave was set to `NULL`. (Bug #52868)

- **Replication:** When using a nontransactional table on the master with autocommit disabled, no `COMMIT` was recorded in the binary log following a statement affecting this table. If the slave's copy of the table used a transactional storage engine, the result on the slave was as though a transaction had been started, but never completed. (Bug #49522)

  References: See also Bug #29288.

- **Replication:** The failure of a `REVOKE` statement was logged with the wrong error code, causing replication slaves to stop even when the failure was expected on the master. (Bug #51987)

- **Replication:** A buffer overrun in the handling of `DATE` column values could cause `mysqlbinlog` to fail when reading logs containing certain combinations of DML statements on a table having a `DATE` column followed by dropping the table. (Bug #52202)

- **Replication:** When using the statement-based logging format, statements that used `CONNECTION_ID()` were always kept in the transaction cache; consequently, nontransactional changes that should have been flushed before the transaction were kept in the transaction cache. (Bug #53075)

  References: This bug was introduced by Bug #51894.

- **Replication:** When temporary tables were in use, switching the binary logging format from `STATEMENT` to `ROW` did not take effect until all temporary tables were dropped. (The existence of temporary tables should prevent switching the format only from `ROW` to `STATEMENT` from taking effect, not the reverse.) (Bug #52616)

- **Replication:** Issuing any DML on a temporary table `temp` followed by `DROP TEMPORARY TABLE temp`, both within the same transaction, caused replication to fail.

  The fix introduces a change to statement-based binary logging with respect to temporary tables. Within a transaction, changes to temporary tables are saved to the transaction cache and written to the binary log when the transaction commits. Otherwise, out-of-order logging of events could occur. This means that temporary tables are treated similar to transactional tables for purposes of caching and logging. This affects assessment of statements as safe or unsafe and the associated error message was changed from:

  ```
  Unsafe statement written to the binary log using statement format
  since BINLOG_FORMAT = STATEMENT. Statements that read from both
  transactional and non-transactional tables and write to any of them
  are unsafe.
  ```

  To:

  ```
  Unsafe statement written to the binary log using statement format
  since BINLOG_FORMAT = STATEMENT. Statements that read from both
  transactional (or a temporary table of any engine type) and
  non-transactional tables and write to any of them are unsafe.
  ```

  (Bug #51894)

  References: See also Bug #51291, Bug #53075, Bug #53259, Bug #53452, Bug #54872. This bug was introduced by Bug #46364.

- **Replication:** Enabling `binlog_direct_non_transactional_updates` causes nontransactional changes to be written to the binary log upon committing the statement. However, even when not enabled, the addition of this variable introduced a number of undesired changes in behavior:

  1. When using `ROW` or `MIXED` logging mode: Nontransactional changes executed within a transaction prior to any transactional changes were written to the statement cache, but those following any transactional changes were written to the transactional cache instead, causing these (later) nontransactional changes to be lost.
  2. When using `ROW` or `MIXED` logging mode: When rolling back a transaction, any nontransactional changes that might be in the transaction cache were disregarded and truncated along with the transactional changes.
  3. When using `STATEMENT` logging mode: A statement that combined transactional and nontransactional changes prior to any other transactional changes within the transaction, but failed, was kept in the transactional cache until the transaction ended, rather than being written to the binary log at the instant of failure (and not deferred to the end of the transaction).

  These problems have been handled as follows:

  - The setting for `binlog_direct_non_transactional_updates` no longer has any effect when the value of `binlog_format` is either `ROW` or `MIXED`. This addresses the first two issues previously listed.
  - When using statement-based logging with `binlog_direct_non_transactional_updates` set to `ON`, any statement combining transactional and nontransactional changes within the same transaction is now stored in the transaction cache, whether or not it succeeds, and regardless of its order of execution among any transactional statements within that transaction. This means that such a statement is now written to the binary log only on transaction commit or rollback.

  (Bug #51291)

  References: This bug was introduced by Bug #46364.

- **Replication:** When run with the `--database` option, `mysqlbinlog` printed `ROLLBACK` statements but did not print any corresponding `SAVEPOINT` statements. (Bug #50407)

- **Replication:** When a `CREATE EVENT` statement was followed by an additional statement and the statements were executed together as a single statement, the `CREATE EVENT` statement was padded with “garbage” characters when written to the binary log. This led to a syntax error when the event was read from the log. (Bug #50095)

- **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.

- **Replication:** When using temporary tables, the binary log needs to insert a pseudo-thread ID for threads that are using temporary tables, each time a switch happens between two threads, both of which are using temporary tables. However, if a thread issued a failing statement before exit, its ID was not recorded in the binary log, and this in turn caused the ID for the next thread that tried to do something with a temporary table not to be logged as well. Subsequent replays of the binary log failed with the error `Table ... doesn't exist`. (Bug #51226)

  References: This bug is a regression of Bug #35583.

- **Replication:** If the master was using `sql_mode='TRADITIONAL'`, duplicate key errors were not sent to the slave, which received `0` rather than the expected error code. This caused replication to fail even when such an error was expected. (Bug #51055)
- **Replication:** ACK packets in semisynchronous replication were not checked for length and malformed packets could cause a server crash. (Bug #52748)
- For updates to `InnoDB` tables, `TIMESTAMP` columns could be updated even when no values actually changed. (Bug #47453)
- A problem with equality propagation optimization for prepared statements and stored procedures caused a server crash upon re-execution of the prepared statement or stored procedure. (Bug #51650)

  References: See also Bug #8115, Bug #8849.
- Accessing a `MERGE` table with an empty underlying table list incorrectly resulted in a “wrong index” error message rather than “end of file.” (Bug #35274)
- Setting the session value of the `debug` system variable also set the global value. (Bug #38054)
- `ALTER TABLE ... ADD COLUMN` for a table with multiple foreign keys caused a server crash. (Bug #45052)
- Using `REPLACE` to update a previously inserted negative value in an `AUTO_INCREMENT` column of an `InnoDB` table caused the table auto-increment value to be updated to 2147483647. (Bug #47720)
- During MySQL server installation using the MSI package on Windows, the `default-character-set` option would be included in the default configuration template file. This caused the MySQL server to fail to start properly. (Bug #52380)
- The `make_binary_distribution` target to `make` could fail on some platforms because the lines generated were too long for the shell. (Bug #54590)
- `mysqld` could fail during execution when using SSL. (Bug #34236)
- Builds of the embedded `mysqld` failed due to a missing element of the `struct NET`. (Bug #53908, Bug #53912)
- Executing a `LOAD XML INFILE` statement could sometimes lead to a server crash. (Bug #51571)
- When using `UNINSTALL PLUGIN` to remove a loaded plugin, open tables and connections caused `mysqld` to hang until the open connections had been closed. (Bug #39053)
- A query that read from a derived table (of the form `SELECT ... FROM (SELECT ...)`) produced incorrect results when the following conditions were present:
  - The table subquery contained a derived query (`(SELECT ...) AS column`).
  - The derived query could potentially produce zero rows or a single `NULL` (that is, no rows matched, or the query used an aggregate function such as `SUM()` running over zero rows).
  - The table subquery joined at least two tables.
  - The join condition involved an index.

  (Bug #47904)
- A `HAVING` clause on a joined table in some cases failed to eliminate rows which should have been excluded from the result set. (Bug #51242)

- Performing a single in-place `ALTER TABLE` containing `ADD INDEX` and `DROP INDEX` options that used the same index name could result in a corrupt table definition file. Now such `ALTER TABLE` statements are no longer performed in place. (Bug #49838)
- The type inference used for view columns caused some columns in views to be handled as the wrong type, as compared to the same columns in base tables. `DATE` columns in base tables were treated as `TIME` columns in views, and base table `TIME` columns as view `DATETIME` columns. (Bug #50918)
- The `MERGE` engine failed to open a child table from a different database if the child table or database name contained characters that were subject to table name to file name encoding.

  Further, the `MERGE` engine did not properly open a child table from the same database if the child table name contained characters such as `'/'`, or `'#'`. (Bug #48265)
- `CHECKSUM TABLE` could compute the checksum for `BIT` columns incorrectly. (Bug #51304)
- `mysql_upgrade` did not detect when `CSV` log tables incorrectly contained columns that could be `NULL`. Now these columns are altered to be `NOT NULL`. (Bug #49823)
- The query shown by `EXPLAIN EXTENDED` plus `SHOW WARNINGS` could produce results different from the original query. (Bug #47669)
- Two sessions trying to set the global `event_scheduler` system variable to different values could deadlock. (Bug #51160)
- For LDML-defined collations, some data structures were not initialized properly to enable `UPPER()` and `LOWER()` to work correctly. (Bug #51976)
- Invalid memory reads occurred for `HANDLER ... READ NEXT` after a failed `HANDLER ... READ FIRST`. (Bug #51877)
- The `YEAR` values `2000` and `0000` could be treated as equal. (Bug #49910)
- The optimizer performed an incorrect join type when `COALESCE()` appeared within an `IN()` operation. (Bug #51598)
- Following a bulk insert into a `MyISAM` table, if `MyISAM` failed to build indexes using repair by sort, data file corruption could occur. (Bug #51307)
- With an XA transaction active, `SET autocommit = 1` could cause side effects such as memory corruption or a server crash. (Bug #51342)
- The optimization to read `MIN()` or `MAX()` values from an index did not properly handle comparisons with `NULL` values. This could produce incorrect results for `MIN()` or `MAX()`when the `WHERE` clause tested a `NOT NULL` column for `NULL`. (Bug #47762)
- Killing a query during the optimization phase of a subquery could cause a server crash. (Bug #47761)
- Setting `myisam_repair_threads` larger than 1 could result in the cardinality for all indexes of a `MyISAM` table being set to 1 after parallel index repair. (Bug #47444)
- Use of `HANDLER` statements with tables that had spatial indexes caused a server crash. (Bug #51357)
- The server crashed when it could not determine the best execution plan for queries involving outer joins with nondeterministic `ON` clauses such as the ones containing the `RAND()` function, a user-defined function, or a `NOT DETERMINISTIC` stored function. (Bug #48483)
- `mysql_upgrade` did not create temporary files properly. (Bug #41057)

- On Windows, `LOAD_FILE()` could cause a crash for some pathnames. (Bug #51893)
- It was possible for `DROP TABLE` of one `MyISAM` table to remove the data and index files of a different `MyISAM` table. (Bug #40980)
- If `myisam_sort_buffer_size` was set to a small value, table repair for `MyISAM` tables with `FULLTEXT` indexes could crash the server. (Bug #51866)
- `EXPLAIN EXTENDED` crashed trying to resolve references to freed temporary table columns for `GROUP_CONCAT() ORDER BY` arguments. (Bug #52397)
- In debug builds, if the listed columns in the view definition of the table used in an `INSERT ... SELECT` statement mismatched, an assertion was raised in the query cache invalidation code following the failing statement. (Bug #46615)
- The optimizer could attempt to evaluate the `WHERE` clause before any rows had been read, resulting in a server crash. (Bug #52177)
- If the arguments to a `CONCAT()` call included a local routine variable, selecting the return value into a user variable could produce an incorrect result. (Bug #40625)
- The test for `readline` during configuration failed when trying to build MySQL in a directory other than the source tree root. (Bug #35250)
- `MyISAM` could write uninitialized data to new index pages. Now zeros are written to unused bytes in the pages. (Bug #47598)
- The server crashed when the optimizer attempted to determine constant tables but a table storage engine did not support exact record count. (Bug #51494)
- For a query that selected from a view and used an alias for the view, the metadata used the alias name rather than the view name in the `MYSQL_FIELD.table` member. (Bug #41788)
- The server could crash populating the `INFORMATION_SCHEMA.PROCESSLIST` table due to lack of mutex protection. (Bug #51377)
- After `TRUNCATE TABLE` of a `MyISAM` table, subsequent queries could crash the server if `myisam_use_mmap` was enabled. (Bug #51868)
- Setting `@@GLOBAL.debug` to an empty string failed to clear the current debug settings. (Bug #52629)
- There was a race condition between flags used for signaling that a query was killed, which led to error-reporting and lock-acquisition problems. (Bug #52356)
- User-defined variables of type `REAL` that contained `NULL` were handled improperly when assigned to a column of another type. (Bug #50511)
- Stored routine DDL statements were written to the binary log using statement-based format regardless of the current logging format. (Bug #51839)
- With a non-`latin1` ASCII-based current character set, the server inappropriately converted `DATETIME` values to strings. This resulted in the optimizer not using indexes on such columns. (Bug #52849)
- `SHOW CREATE TABLE` was blocked if the table was write locked by another session. (Bug #52593)
- `mysql_upgrade` attempted to work with stored routines before they were available. (Bug #52444)
- A `COUNT(DISTINCT)` query on a view could cause a server crash. (Bug #51980)

- Corrupt `MyISAM` tables were automatically repaired even when `myisam_recover_options` was set to `OFF`. (Bug #51327)
- Two sessions trying to set the global `event_scheduler` system variable to `OFF` resulted in one of them hanging waiting for the event scheduler to stop. (Bug #52367)
- An assertion was raised as a result of a `NULL` string being passed to the `dtoa` code. (Bug #52165)
- `OPTIMIZE TABLE` for an `InnoDB` table could raise an assertion if another session issued a concurrent `DROP TABLE`. (Bug #47459)
- Certain path names passed to `LOAD_FILE()` could cause a server crash. (Bug #53417)
- For debug builds, creating a view containing a subquery that might require collation adjustment caused an assertion to be raised. For example, this could occur if some items had different collations but the result collation could be adjusted to the one of them. (Bug #52120)
- Locking involving the `LOCK_plugin`, `LOCK_global_system_variables`, and `LOCK_status` mutexes could deadlock. (Bug #51591)
- `InnoDB` fast index creation could incorrectly use a table copy in some cases. (Bug #50946)
- If the server is started with `--skip-grant-tables`, plugin loading and unloading should be prohibited, but the server failed to reject `INSTALL PLUGIN` and `UNINSTALL PLUGIN` statements. (Bug #46261)
- On Windows, the server failed to find a description for Event ID 100. (Bug #48042)
- A syntactically invalid trigger could cause the server to crash when trying to list triggers. (Bug #50755)
- A memory leak occurred due to missing deallocation of the `comparators` array (a member of the `Arg_comparator` class). (Bug #52124)
- `EXPLAIN` could cause a server crash for some queries with subqueries. (Bug #48419)
- `mysqld_safe` set `plugin_dir` using a hardcoded default path name rather than a path depending on `basedir`. (Bug #52737)
- In MySQL 5.1, `READ COMMITTED` was changed to use less locking due to the availability of row-based binary logging (see the Note under `READ COMMITTED` at `SET TRANSACTION` Syntax). However, `READ UNCOMMITTED` did not have the same change, so it was using more locks than the higher isolation level, which is unexpected. This was changed so that `READ UNCOMMITTED` now also uses the lesser amount of locking and has the same restrictions for binary logging. (Bug #48607)
- Semi-consistent read was implemented for `InnoDB` to address Bug #3300. Semi-consistent reads do not block when a nonmatching record is already locked by some other transaction. If the record is not locked, a lock is acquired, but is released if the record does not match the `WHERE` condition. However, semi-consistent read was attempted even for `UPDATE` statements having a `WHERE` condition of the form `pk_col1=constant1, ..., pk_colN=constantN`. Some code failed that was designed with the assumption that semi-consistent read would be only attempted on table scans. (Bug #52663)
- `mysqld_safe` did not always pass `--open-files-limit` through to `mysqld`. `mysqld_safe` did not treat dashes and underscores as equivalent in option names. (Bug #47095)
- `InnoDB` could fail to create a unique index on `NULL` columns. (Bug #41904)
- Using an initial command with `mysql_options(..., MYSQL_INIT_COMMAND, ...)` that generated multiple result sets (such as a stored procedure or a multi-statement command) left the connection unusable. (Bug #42373)

- `mysqldump` and `SELECT ... INTO OUTFILE` truncated long `BLOB` and `TEXT` values to 766 bytes. (Bug #53088)
- Setting `--secure-file-priv` to the empty string left the value unaffected. (Bug #50373)
- `UPDATE` on an `InnoDB` table modifying the same index that was used to satisfy the `WHERE` condition could trigger a debug assertion under some circumstances. (Bug #53830)
- In the debug version of the server, the `FreeState()` function could in some circumstances be called twice, leading to an assertion failure. (Bug #52884)
- Valgrind warnings resulting from passing incomplete `DATETIME` values to the `TIMESTAMP()` function were corrected. (Bug #53942)
- Calculation of intervals for Event Scheduler events was not portable. (Bug #50087)
- A trigger could change the behavior of assigning `NULL` to a `NOT NULL` column. (Bug #48525)
- Selecting from `INFORMATION_SCHEMA.ROUTINES` or `INFORMATION_SCHEMA.PARAMETERS` resulted in a memory leak. (Bug #48729)
- Aggregate functions could incorrectly return `NULL` in outer join queries. (Bug #52051)
- The Loose Index Scan optimization method assumed that it could depend on the partitioning engine to maintain interval endpoint information, as if it were a storage engine. (Bug #50939)
- Incorrect results could be returned for `LEFT JOIN` of `InnoDB` tables with an impossible `WHERE` condition. (Bug #53334)
- When the transaction isolation level was `REPEATABLE READ` and binary logging used statement or mixed format, `SELECT` statements with subqueries referencing `InnoDB` tables unnecessarily acquired shared locks on rows in these tables. (Bug #46947)
- For single-table `DELETE` statements that used quick select and index scan simultaneously caused a server crash or assertion failure. (Bug #53450)
- MySQL incorrectly processed `ALTER DATABASE` `` `#mysql50#`*`special`*`` `` `UPGRADE DATA DIRECTORY NAME` where *`special`* was `.`, `..`, or a sequence starting with `./` or `../`. It used the server data directory (which contains other regular databases) as the database directory. (Bug #53804, CVE-2010-2008)
- `InnoDB` crashed when replacing duplicates in a table after a fast `ALTER TABLE` added a unique index. (Bug #53592)
- For `InnoDB` tables, the error handler for a fast `CREATE INDEX` did not reset the error state of the transaction before attempting to undo a failed operation, resulting in a crash. (Bug #53591)
- The definition of the `MY_INIT` macro in `my_sys.h` included an extraneous semicolon, which could cause compilation failure. (Bug #53906)
- On some systems, such as Mac OS X, the `sockaddr_in` and `sockaddr_in6` structures contain a non-standard field (`sin_len` / `sin6_len`) that must be set but was not. This resulted in host name lookup failure. (Bug #52923)
- `gcc` 4.4.0 could fail to compile `dtoa.c`. (Bug #45882)
- Spurious duplicate-key errors occurred for multiple-column indexes on `BINARY` columns. (Bug #52430)
- Cast operations on `NULL` `DECIMAL` values could cause server crashes or Valgrind warnings. (Bug #52168)

- `ALTER TABLE` for views is not legal but did not produce an error. (If you need to rename a view, use `RENAME TABLE`.) (Bug #53976)
- Valgrind warnings in the `InnoDB` `compare_record()` function were corrected. (Bug #38999)
- Operations on geometry data types failed on some systems for builds compiled with Sun Studio. (Bug #52208)
- Column names displayed from the `PARTITION_EXPRESSION` column of the `INFORMATION_SCHEMA.PARTITIONS` table did not include escape characters as necessary. (Bug #39338)
- Concurrent `SHOW COLUMNS` statements could cause a server crash. (Bug #52856)
- Previously, the server held a global mutex while performing file operations such as deleting an `.frm` or data file, or reading index statistics from a data file. Now the mutex is not held for these operations. Instead, the server uses metadata locks. (Bug #50589, Bug #51557, Bug #49463)
- A crash occurred if a table that was locked with `LOCK TABLES` was listed twice in a `DROP TABLE` statement. (Bug #54282)
- The server could crash with an out of memory error when trying to parse a query that was too long to fit in memory. Now the parser rejects such queries with an `ER_OUT_OF_RESOURCES` error. (Bug #42064)
- `OPTIMIZE TABLE` could be run on a table in use by a transaction in a different session, causing repeatable read to break. (Bug #53798)
- The `Lock_time` value in the slow query log was negative for stored routines. (Bug #53191)
- For a concurrent load of 16 or more connections containing many `LOCK TABLES WRITE` statements for the same table, server throughput was significantly lower for MySQL 5.5.3 and 5.5.4 than for earlier versions (10%–40% lower depending on concurrency). (Bug #52289)
- On Intel x86 machines, the optimizer could choose different execution plans for a query depending on the compiler version and optimization flags used to build the server binary. (Bug #48537)
- For outer joins, the optimizer could fail to properly calculate table dependencies. (Bug #52005)
- Queries that used `MIN()` or `MAX()` on indexed columns could be optimized incorrectly. (Bug #53859)
- The optimizer sometimes used `filesort` for `ORDER BY` when it should have used an index. (Bug #38745)
- `ALTER TABLE` on `InnoDB` tables (including partitioned tables) acquired exclusive locks on rows of the table being altered. If there was a concurrent transaction that did locking reads from this table, this sometimes led to a deadlock that was not detected by the metadata lock subsystem or by InnoDB (and was reported only after exceeding `innodb_lock_wait_timeout`). (Bug #51263)
- The `length` and `max_length` metadata values were incorrect for columns with the `TEXT` family of data types that used multi-byte character sets. This bug was introduced in MySQL 5.5.3. (Bug #52520)
- When `SET TRANSACTION ISOLATION LEVEL` was used to set the isolation level for the next transaction, the level could persist for subsequent transactions. (Bug #39170)
- The `AND CHAIN` option for `COMMIT` and `ROLLBACK` failed to preserve the current transaction isolation level. Setting `completion_type` to 1 also failed to do so. (Bug #53343)
- Setting the `innodb_change_buffering` system variable to `DEFAULT` produced an incorrect result. (Bug #53165)

- Inconsistent checking of the relationship between `SHOW` statements and `INFORMATION_SCHEMA` queries caused such queries to fail sometimes. (Bug #54422)
- For the `COMMIT` and `ROLLBACK` statements, the `AND CHAIN` and `RELEASE` modifiers should be mutually exclusive, but the parser permitted both to be specified. (Bug #46527)
- If the `completion_type` session variable was changed after a stored procedure or prepared statement had been cached, the change had no effect on subsequent executions of the procedure or statement. (Bug #53346)
- The `check_table_is_closed()` debugging function did not protect access to the `MyISAM` open tables list, with the result that server crashes could occur during table drop or rename operations. (Bug #52432)
- Manual pages for a few little-used programs were missing from RPM packages. (Bug #44370)
- `support-files/mysql.spec.sh` had unnecessary Perl dependencies. (Bug #49723)
- The behavior of the RPM installation for both new installations and upgrade installations has changed.

  During a new installation, the server boot scripts are installed, but the MySQL server is not started at the end of the installation, since the status of the system during an unattended installation is not known.

  During an upgrade installation using the RPM packages, if the server is running when the upgrade occurs, the server is stopped, the upgrade occurs, and server is restarted. If the server is not already running when the RPM upgrade occurs, the server is not started at the end of the installation.

  The boot scripts for MySQL are installed in the appropriate directories in `/etc`, so the MySQL server will be restarted automatically at the next machine reboot. (Bug #27072)
- A query on a `FEDERATED` table in which the data was ordered by a `TEXT` column returned incorrect results. For example, a query such as the following produced incorrect results if column `column1` was a `TEXT` column:

  ```
  SELECT * FROM table1 ORDER BY column1;
  ```

  (Bug #32426)
- The parser allocated too much memory for a query string containing multiple statements. (Bug #27863)
- MySQL Makefiles relied on GNU extensions. (Bug #30708)
- `ROW_COUNT()` returned a meaningful value only for some DML statements. Now it returns a value as follows:

  - DDL statements: 0. This applies to statements such as `CREATE TABLE` or `DROP TABLE`.
  - DML statements other than `SELECT`: The number of affected rows. This applies to statements such as `UPDATE`, `INSERT`, or `DELETE` (as before), but now also to statements such as `ALTER TABLE` and `LOAD DATA INFILE`.
  - `SELECT`: -1 if the statement returns a result set, or the number of rows “affected” if it does not. For example, for `SELECT * FROM t1`, `ROW_COUNT()` returns -1. For `SELECT * FROM t1 INTO OUTFILE 'file_name'`, `ROW_COUNT()` returns the number of rows written to the file.
  - `SIGNAL` statements: 0.

  (Bug #21818)

Changes in MySQL 5.5.4 (2010-04-09)

InnoDB Notes

- `InnoDB` has been upgraded to version 1.1. This version is considered of Beta quality.

RPM Notes

- `debuginfo` RPM packages are no longer being built or published.

Functionality Added or Changed

- **InnoDB:** `InnoDB` now has a `innodb_use_native_aio` system variable that can be disabled at startup if there is a problem with the asynchronous I/O subsystem in the OS. This variable applies to Linux systems only, where the MySQL server now has a dependency on the `libaio` library.

Bugs Fixed

- **Performance; InnoDB:** The redo scan during `InnoDB` recovery used excessive CPU. The efficiency of this scan was improved, significantly speeding up crash recovery. (Bug #49535, Bug #29847)
- **Performance; InnoDB:** `InnoDB` page-freeing operations were made faster for compressed blocks, speeding up `ALTER TABLE`, `DROP TABLE`, and other operations on compressed tables that free compressed blocks. One symptom of the older behavior could be 100% CPU use during these operations. (Bug #35077)
- **InnoDB:** The AIX implementation of `readdir_r()` caused `InnoDB` errors. (Bug #50691)
- **InnoDB:** The limit of 1023 concurrent data-modifying transactions has been raised. The limit is now 128 × 1023 concurrent transactions that generate undo records. You can remove any workarounds that require changing the proper structure of your transactions, such as committing more frequently or delaying DML operations to the end of a transaction. See Better Scalability with Multiple Rollback Segments. (Bug #26590)
- `InnoDB` did not reset table `AUTO_INCREMENT` values to the last used values after a server restart. (Bug #49032)
- A unique index on a column prefix could not be upgraded to a primary index even if there was no primary index already defined. (Bug #51378)
- When using the `EXAMPLE` storage engine, when the engine had been built as a plugin (instead of built in), and DTrace probes had been enabled during the build, loading the storage engine library failed due to a missing object table entry.

Changes in MySQL 5.5.3 (2010-03-24, Milestone 3)

InnoDB Notes

- This release includes `InnoDB` 1.0.6. This version is considered of Release Candidate (RC) quality.

IPv6 Support

- MySQL Server now can accept TCP/IP connections from clients connecting over IPv6. See IPv6 Support. For example, this command connects over IPv6 to the MySQL server on the local host:

```
shell> mysql -h ::1
```

  To use this capability, two things must be true:

- Your system must be configured to support IPv6.
- The default MySQL server configuration permits only IPv4 connections, so the server must be configured for IPv6 connections. To permit IPv6 connections in addition to or instead of IPv4 connections, start the server with an appropriate `--bind-address` option.

MySQL account names permit IPv6 addresses to enable DBAs to specify privileges for clients that connect to the server over IPv6. See Specifying Account Names. IPv6 addresses can be specified in account names in statements such as `CREATE USER`, `GRANT`, and `REVOKE`. For example:

```
mysql> CREATE USER 'bill'@'::1' IDENTIFIED BY 'secret';
mysql> GRANT SELECT ON mydb.* TO 'bill'@'::1';
```

The default set of accounts created during MySQL installation now include an account for `'root'@'::1'`. See Securing the Initial MySQL Accounts. This account can be used to make connections as `root` if the server is bound to `::1` and accepts only local IPv6 connections. (Bug #8836)

Performance Schema Notes

- MySQL Server now includes the Performance Schema, a feature for monitoring server execution at a low level. The implementation uses the `PERFORMANCE_SCHEMA` storage engine and the `performance_schema` database. The Performance Schema focuses primarily on performance data. This differs from `INFORMATION_SCHEMA`, which serves for inspection of metadata. For more information, see MySQL Performance Schema.

  Performance Schema support is included in binary MySQL distributions but is disabled by default. To enable it, start the server with the `--performance_schema` option.

  To create the `performance_schema` database if you are upgrading from an earlier release, run `mysql_upgrade` and restart the server. See `mysql_upgrade` — Check and Upgrade MySQL Tables.

Functionality Added or Changed

- **Performance:** The performance of internal functions that trim multiple spaces from strings when comparing them has been improved. (Bug #14637)
- **Incompatible Change:** The following obsolete constructs have been removed. Where alternatives are shown, applications should be updated to use them.
  - The `log_bin_trust_routine_creators` system variable (use `log_bin_trust_function_creators`).
  - The `myisam_max_extra_sort_file_size` system variable.
  - The `record_buffer` system variable (use `read_buffer_size`).
  - The `sql_log_update` system variable.
  - The `table_lock_wait_timeout` system variable.
  - The `table_type` system variable (use `storage_engine`).
  - The `FRAC_SECOND` modifier for the `TIMESTAMPADD()` function (use `MICROSECOND`).
  - The `TYPE` table option to specify the storage engine for `CREATE TABLE` or `ALTER TABLE` (use `ENGINE`).

- The `SHOW TABLE TYPES` SQL statement (use `SHOW ENGINES`).
- The `SHOW INNODB STATUS` and `SHOW MUTEX STATUS` SQL statements (use `SHOW ENGINE INNODB STATUS` `SHOW ENGINE INNODB MUTEX`).
- The `SHOW PLUGIN` SQL statement (use `SHOW PLUGINS`).
- The `LOAD TABLE ... FROM MASTER` and `LOAD DATA FROM MASTER` SQL statements (use `mysqldump` or `mysqlhotcopy` to dump tables and `mysql` to reload dump files).
- The `BACKUP TABLE` and `RESTORE TABLE` SQL statements (use `mysqldump` or `mysqlhotcopy` to dump tables and `mysql` to reload dump files).
- `TIMESTAMP(N)` data type: The ability to specify a display width of *N* (use without *N*).
- The `--default-character-set` and `--default-collation` server options (use `--character-set-server` and `--collation-server`).
- The `--delay-key-write-for-all-tables` server option (use `--delay-key-write=ALL`).
- The `--enable-locking` and `--skip-locking` server options (use `--external-locking` and `--skip-external-locking`).
- The `--log-bin-trust-routine-creators` server option (use `--log-bin-trust-function-creators`).
- The `--log-long-format` server option.
- The `--log-update` server option.
- The `--master-`*`xxx`* server options to set replication parameters (use the `CHANGE MASTER TO` statement instead): `--master-host`, `--master-user`, `--master-password` , `--master-port`, `--master-connect-retry`, `--master-ssl`, `--master-ssl-ca`, `--master-ssl-capath`, `--master-ssl-cert`, `--master-ssl-cipher`, `--master-ssl-key`.
- The `--safe-show-database` server option.
- The `--skip-symlink` and `--use-symbolic-links` server options (use `--skip-symbolic-links` and `--symbolic-links`).
- The `--sql-bin-update-same` server option.
- The `--warnings` server option (use `--log-warnings`).
- The `--no-named-commands` option for `mysql` (use `--skip-named-commands`).
- The `--no-pager` option for `mysql` (use `--skip-pager`).
- The `--no-tee` option for `mysql` (use `--skip-tee`).
- The `--position` option for `mysqlbinlog` (use `--start-position`).
- The `--all` option for `mysqldump` (use `--create-options`).
- The `--first-slave` option for `mysqldump` (use `--lock-all-tables`).
- The `--config-file` option for `mysqld_multi` (use `--defaults-extra-file`).

- The `--set-variable=var_name=value` and `-O var_name=value` general-purpose options for setting program variables (use `--var_name=value`).

(Bug #48048)

References: See also Bug #47974, Bug #56408.

- **Incompatible Change:** `CREATE VIEW` and `DROP VIEW` now are prohibited while a `LOCK TABLES` statement is in effect. (Bug #56571)

- **Incompatible Change:** Several columns were added to the `INFORMATION_SCHEMA.ROUTINES` table to provide information about the `RETURNS` clause data type for stored functions: `DATA_TYPE`, `CHARACTER_MAXIMUM_LENGTH`, `CHARACTER_OCTET_LENGTH`, `NUMERIC_PRECISION`, `NUMERIC_SCALE`, `CHARACTER_SET_NAME`, and `COLLATION_NAME`.

  This change produces an incompatibility for applications that depend on column order in the `ROUTINES` table because the new columns appear between the `ROUTINE_TYPE` and `DTD_IDENTIFIER` columns. Such applications may need to be adjusted to account for the new columns.

- **Incompatible Change:** Aliases for wildcards (as in `SELECT t.* AS 'alias' FROM t`) are no longer accepted and result in an error. Previously, such aliases were ignored silently. (Bug #27249)

- **Incompatible Change:** The Unicode implementation has been extended to provide support for supplementary characters that lie outside the Basic Multilingual Plane (BMP). Noteworthy features:

  - `utf16` and `utf32` character sets have been added. These correspond to the UTF-16 and UTF-32 encodings of the Unicode character set, and they both support supplementary characters.

  - The `utf8mb4` character set has been added. This is similar to `utf8`, but its encoding allows up to four bytes per character to enable support for supplementary characters.

  - The `ucs2` character set is essentially unchanged except for the inclusion of some newer BMP characters.

  In most respects, upgrading to MySQL 5.5 should present few problems with regard to Unicode usage, although there are some potential areas of incompatibility. These are the primary areas of concern:

  - For the variable-length character data types (`VARCHAR` and the `TEXT` types), the maximum length in characters is less for `utf8mb4` columns than for `utf8` columns.

  - For all character data types (`CHAR`, `VARCHAR`, and the `TEXT` types), the maximum number of characters that can be indexed is less for `utf8mb4` columns than for `utf8` columns.

  Consequently, if you want to upgrade tables from `utf8` to `utf8mb4` to take advantage of supplementary-character support, it may be necessary to change some column or index definitions.

  For additional details about the new Unicode character sets and potential incompatibilities, see Unicode Support, and Upgrading from Previous to Current Unicode Support.

- **Incompatible Change:** The server now includes `dtoa`, a library for conversion between strings and numbers by David M. Gay. In MySQL, this library provides the basis for improved conversion between string or `DECIMAL` values and approximate-value (`FLOAT`/`DOUBLE`) numbers:

  - Consistent conversion results across platforms, which eliminates, for example, Unix versus Windows conversion differences.

- Accurate representation of values in cases where results previously did not provide sufficient precision, such as for values close to IEEE limits.
- Conversion of numbers to string format with the best possible precision. The precision of `dtoa` is always the same or better than that of the standard C library functions.

Because the conversions produced by this library differ in some cases from previous results, the potential exists for incompatibilities in applications that rely on previous results. For example, applications that depend on a specific exact result from previous conversions might need adjustment to accommodate additional precision.

For additional information about the properties of `dtoa` conversions, see Type Conversion in Expression Evaluation.

References: See also Bug #12860, Bug #21497, Bug #26788, Bug #24541, Bug #34015.

- **Incompatible Change:** Several changes were made to processing of server system variables and command-line options to make their treatment more consistent.

  General changes:

  - The help message text displayed by `mysqld --verbose --help` now consistently uses dashes to show the names of options and system variables that can be set at server startup. Previously, the message used both dashes and underscores (generally with dashes for options and underscores for system variables). For example, the help message now displays `--log-output` and `--general-log`, whereas previously it displayed `--log-output` and `--general_log`.

    This is a display-only change. The permissible syntax for setting options and variables remains unchanged:

    - At server startup, you can specify options and variables on the command line or in option files using either dashes or underscores.
    - For those system variables that can be set at runtime (for example, using `SET`), you must specify them using underscores.

  - There are fewer session-only system variables. These variables now have a global value: `autocommit`, `foreign_key_checks`, `profiling`, `sql_auto_is_null`, `sql_big_selects`, `sql_buffer_result`, `sql_log_bin`, `sql_log_off`, `sql_notes`, `sql_quote_show_create`, `sql_safe_updates`, `sql_warnings`, `unique_checks`.

    For those variables, you can now set the global value to change the value from which the session value is initialized for new sessions.

    The following list shows the variables that remain session-only. They apply only in the context of a specific session so that a global value is of no use: `debug_sync`, `error_count`, `identity`, `insert_id`, `last_insert_id`, `pseudo_thread_id`, `rand_seed1`, `rand_seed2`, `timestamp`, `warning_count`.

  - All system variables are accessible at runtime using `@@` syntax (`@@GLOBAL.`*`var_name`*, `@@SESSION.`*`var_name`*, `@@`*`var_name`*). Previously, this syntax produced an error for some variables.
  - All system variables are included as appropriate in the output from `SHOW {GLOBAL, SESSION} VARIABLES` and the `INFORMATION_SCHEMA.GLOBAL_VARIABLES` and

`INFORMATION_SCHEMA.SESSION_VARIABLES` tables. Previously, some variables were not displayed.

- "As appropriate" in the preceding item means that `SHOW GLOBAL VARIABLES` and `INFORMATION_SCHEMA.GLOBAL_VARIABLES` no longer include session-only system variables. Previously, these included the global value of a variable if it had one, and the session value if not. (`SHOW SESSION VARIABLES` still includes global-only variables.)
- The server now enforces type checking for assignments to system variables, so it is more consistent and strict about rejecting invalid values.
- For attempts to assign a negative value to an unsigned system variable, the server truncates the value to the minimum permitted value. Previously, there was sometimes wraparound to a large positive value.
- Some system variables (typically those that control memory or disk allocation) are permitted to take only values that are a multiple of a given block size, and assigning a value not a block size multiple causes truncation to the nearest multiple. (For example, `net_buffer_length` must be a multiple of 1024. Assigning 16384 results in a value of 16384, whereas assigning 16383 results in a value of 15360.) A warning now occurs when adjustment of the specified value takes place. Previously, adjustment was silent.
- More system variables can be assigned the value `DEFAULT` to set them to their default value. Previously, this syntax produced an error in some cases.
- All variables that have a `SET` data type value can be set to an integer value that is treated like a bit mask. Previously, this did not work for some SET-type variables.
- The default value for several system variables no longer differs between 32-bit and 64-bit builds. Previously, the values differed by about 100 bytes for some variables.
- There are no longer any write-only system variables. For example, `SELECT @@rand_seed1` returns 0, not `Variable 'rand_seed1' can only be set, not read`.

Variable-specific changes:

- The `concurrent_insert` system variable now is handled as an enumeration with the permissible values `NEVER`, `AUTO`, and `ALWAYS`. The corresponding integer values 0, 1, and 2 are still recognized.
- The `completion_type` system variable now is handled as an enumeration with the permissible values `NO_CHAIN`, `CHAIN`, and `RELEASE`. The corresponding integer values 0, 1, and 2 are still recognized.
- For `concurrent_insert` and `completion_type`, the string form of the value is displayed by `SHOW VARIABLES` and `SELECT @@var_name`.
- The unused `rpl_recovery_rank` system variable is deprecated.
- The `storage_engine` system variable is deprecated in favor of the new system variable `default_storage_engine`. This enables pairing of the `--default-storage-engine` command-line option with a system variable of a more closely corresponding name.
- The `--myisam-recover` option is renamed to `--myisam-recover-options` to pair better with the name of the `myisam_recover_options` system variable. The old option name still works because it is recognized as an unambiguous prefix of the new name. (Option prefix recognition occurs as described in Specifying Program Options.)

- `--myisam-recover-options` has a new permissible value `OFF`.
- Attempts to drop the default key cache produce an error. Previously, it produced only a warning and status of success even though the attempt failed.

References: See also Bug #34437, Bug #34635, Bug #11747961, Bug #34829, Bug #34878, Bug #25430.

- **Incompatible Change:** Implicit conversion of a number or temporal value to string now produces a value that has a character set and collation determined by the `character_set_connection` and `collation_connection` system variables. (These variables commonly are set with `SET NAMES`. For information about connection character sets, see Connection Character Sets and Collations.)

  This change means that such a conversion results in a character (nonbinary) string (a `CHAR`, `VARCHAR`, or `LONGTEXT` value), except when the connection character set is set to `binary`. In that case, the conversion result is a binary string (a `BINARY`, `VARBINARY`, or `LONGBLOB` value).

  Previously, an implicit conversion always produced a binary string, regardless of the connection character set. Such implicit conversions to string typically occur for functions that are passed numeric or temporal values when string values are more usual, and thus could have effects beyond the type of the converted value. Consider the expression `CONCAT(1, 'abc')`. The numeric argument `1` was converted to the binary string `'1'` and the concatenation of that value with the nonbinary string `'abc'` produced the binary string `'1abc'`.

  This change in conversion behavior affects several functions that expect string arguments because a numeric or temporal argument converted to a string now results in a character rather than binary string argument:

  - String functions: `CONCAT()`, `CONCAT_WS()`, `ELT()`, `EXPORT_SET()`, `INSERT()`, `LCASE()`, `LEFT()`, `LOWER()`, `LPAD()`, `LTRIM()`, `MID()`, `QUOTE()`, `REPEAT()`, `REPLACE()`, `REVERSE()`, `RIGHT()`, `RPAD()`, `RTRIM()`, `SOUNDEX()`, `SUBSTRING()`, `TRIM()`, `UCASE()`, `UPPER()`.
  - Date and time functions: `ADDDATE()`, `ADDTIME()`, `DATE_ADD()`, `DATE_SUB()`, `DAYNAME()`, `GET_FORMAT()`, `MONTHNAME()`, `SUBDATE()`, `SUBTIME()`, `TIMESTAMPADD()`.

  These functions remain unaffected:

  - `CHAR()` without a `USING` clause still returns `VARBINARY`.
  - Functions that previously returned `utf8` strings still do so. Examples include `CHARSET()` and `COLLATION()`.

  Encryption and compression functions that expect string arguments and previously returned binary strings are affected depending on the content of the return value:

  - If the return value contains only ASCII characters, the function now returns a character string with the connection character set and collation: `MD5()`, `OLD_PASSWORD()`, `PASSWORD()`, `SHA()`, `SHA1()`. The `ASTEXT()` and `ASWKT()` spatial functions also fall into this category.
  - If the return value can contain non-ASCII characters, the function still returns a binary string: `AES_ENCRYPT()`, `COMPRESS()`, `DES_ENCRYPT()`, `ENCODE()`, `ENCRYPT()`.

  The `INET_NTOA()` return value contains only ASCII characters, and this function now returns a character string with the connection character set and collation rather than a binary string.

- **Important Change; Replication:** `RESET MASTER` and `RESET SLAVE` now reset the values shown for `Last_IO_Error`, `Last_IO_Errno`, `Last_SQL_Error`, and `Last_SQL_Errno` in the output of `SHOW SLAVE STATUS`. (Bug #34654)

  References: See also Bug #44270.

- **Important Change:** The `--skip-thread-priority` option is now deprecated such that the server will not change the thread priorities by default. Giving threads different priorities might yield marginal improvements in some platforms (where it actually works), but it might instead cause significant degradation depending on the thread count and number of processors. Meddling with the thread priorities is a not a safe bet as it is very dependent on the behavior of the CPU scheduler and system where MySQL is being run. (Bug #35164, Bug #37536)

- **Replication; Cluster Replication:** MySQL Replication now supports attribute promotion and demotion for row-based replication between columns of different but similar types on the master and the slave. For example, it is possible to promote an `INT` column on the master to a `BIGINT` column on the slave, and to demote a `TEXT` column to a `VARCHAR` column.

  The implementation of type demotion distinguishes between lossy and non-lossy type conversions, and their use on the slave can be controlled by setting the `slave_type_conversions` global server system variable.

  For more information, see Row-based replication: attribute promotion and demotion. (Bug #47163, Bug #46584)

- **Replication:** For replication based on row-based and mix-format binary logging, it is now safe to mix transactional and nontransactional statements within a transaction. The nontransactional statements are logged immediately rather than waiting until the transaction ends, ensuring that their results are logged and replicated correctly regardless of the result of the transaction.

- For boolean options, the option-processing library now prints additional information in the `--help` message: If the option is enabled by default, the message says so and indicates that the `--skip` form of the option disables the option. This affects all compiled MySQL programs that use the library. (Bug #35224)

- The use of the `SQL_CACHE` and `SQL_NO_CACHE` options in `SELECT` statements now is checked more restrictively: 1) Previously, both options could be given in the same statement. This is no longer true; only one can be given. 2) Previously, these options could be given in `SELECT` statements that were not at the top-level. This is no longer true; the options are not permitted in subqueries (including subqueries in the `FROM` clause, and `SELECT` statements in unions other than the first `SELECT`. (Bug #35020)

- MySQL support for adding collations using LDML specifications did not support the `<i>` identity rule that indicates one character sorts identically to another. The `<i>` rule now is supported. See LDML Syntax Supported in MySQL. (Bug #37129)

- `SHOW PROFILE CPU` has been ported to Windows. Thanks to Alex Budovski for the patch. (Bug #50057)

- `mysqltest` has a new `--max-connections` option to set a higher number of maximum permitted server connections than the default 128. This option can also be passed using `mysql-test-run.pl`. (Bug #51135)

- `mysql-test-run.pl` has a new `--portbase` option and a corresponding `MTR_PORT_BASE` environment variable for setting the port range, as an alternative to the existing `--build-thread` option. (Bug #50182)

- `mysql-test-run.pl` now has a `--gprof` option that runs the server through the `gprof` profiler, much the same way the currently supported `--gcov` option runs it through `gcov`. (Bug #49345)
- `mysqltest` now has a `lowercase_result` command that converts the output of the next statement to lowercase. This is useful for test cases where the lettercase may vary between platforms. (Bug #48863)
- `mysqltest` now has a `remove_files_wildcard` command that removes files matching a pattern from a directory. (Bug #39774)
- The maximum length of table comments was extended from 60 to 2048 characters. The maximum length of column comments was extended from 255 to 1024 characters. Index definitions now can include a comment of up to 1024 characters.
- Added the `PARAMETERS` table to `INFORMATION_SCHEMA`. The `PARAMETERS` table provides information about stored procedure and function parameters, and about return values for stored functions.
- The `TABLESPACES` table has been added to `INFORMATION_SCHEMA` for tracking tablespace details.
- When the server detects `MyISAM` table corruption, it now writes additional information to the error log, such as the name and line number of the source file, and the list of threads accessing the table. Example: `Got an error from thread_id=1, mi_dynrec.c:368`. This is useful information to include in bug reports.
- Three options were added to `mysqldump` make it easier to generate a dump from a slave server:
  - `--dump-slave` is similar to `--master-data`, but the `CHANGE MASTER TO` statement contains binary log coordinates for the slave's master host, not the slave itself.
  - `--apply-slave-statements` causes `STOP SLAVE` and `START SLAVE` statements to be added before the `CHANGE MASTER TO` statement and at the end of the output, respectively.
  - `--include-master-host-port` causes the `CHANGE MASTER TO` statement to include `MASTER_PORT` and `MASTER_HOST` options for the slave's master.

  (Bug #8368)
- The `mysql` client now has an `--auto-vertical-output` option, which causes result sets to be displayed vertically if they are too wide for the current window, and uses normal tabular format otherwise. (This applies to statements terminated by `;` or `\G`.) (Bug #26780)
- Previously, prepared `CALL` statements could be used through the C API only for stored procedures that produce at most one result set, and applications could not use placeholders for `OUT` or `INOUT` parameters. For prepared `CALL` statements used using `PREPARE` and `EXECUTE`, placeholders could not be used for `OUT` or `INOUT` parameters.

  For the C API, prepared `CALL` support now is expanded in the following ways:
  - A stored procedure can produce any number of result sets. The number of columns and the data types of the columns need not be the same for all result sets.
  - The final values of `OUT` and `INOUT` parameters are available to the calling application after the procedure returns. These parameters are returned as an extra single-row result set following any result sets produced by the procedure itself. The row contains the values of the `OUT` and `INOUT` parameters in the order in which they are declared in the procedure parameter list.
  - A new C API function, `mysql_stmt_next_result()`, is available for processing stored procedure results. See C API Support for Prepared `CALL` Statements.

  - The `CLIENT_MULTI_RESULTS` flag now is enabled by default. It no longer needs to be enabled when you call `mysql_real_connect()`. (This flag is necessary for executing stored procedures because they can produce multiple result sets.)

  For `PREPARE` and `EXECUTE`, placeholder support for `OUT` and `INOUT` parameters is now available. See `CALL` Syntax. (Bug #11638, Bug #17898)

- `mysqladmin` now permits the password value to be omitted following the `password` command. In this case, `mysqladmin` prompts for the password value, which enables you to avoid specifying the password on the command line. Omitting the password value should be done only if `password` is the final command on the `mysqladmin` command line. Otherwise, the next argument is taken as the password. (Bug #5724)

- `TRUNCATE TABLE` now is permitted for a table for which a `WRITE` lock has been acquired with `LOCK TABLES`. (Bug #20667)

  References: See also Bug #46452.

- Some conversions between Japanese character sets are more efficient.

- `FLUSH LOGS` now takes an optional *`log_type`* value so that `FLUSH` *`log_type`* `LOGS` can be used to flush only a specified log type. These *`log_type`* options are permitted:

  - `BINARY` closes and reopens the binary log files.
  - `ENGINE` closes and reopens any flushable logs for installed storage engines.
  - `ERROR` closes and reopens the error log file.
  - `GENERAL` closes and reopens the general query log file.
  - `RELAY` closes and reopens the relay log files.
  - `SLOW` closes and reopens the slow query log file.

  Thanks to Eric Bergen for the patch to implement this feature. (Bug #14104)

- The server now provides a pluggable audit interface that enables information about server operations to be reported to interested parties. Audit plugins may register with the audit interface to receive notification about server operations. When an auditable event occurs within the server, the server determines whether notification is needed. For each registered audit plugin, the server checks the event against those event classes in which the plugin is interested and passes the event to the plugin if there is a match. For more information, see Audit Plugins.

- The `optimizer_switch` system variable now has an `engine_condition_pushdown` flag to control whether storage engine condition pushdown optimization is used. As a consequence, the `engine_condition_pushdown` system variable now is deprecated.

- Code that produces query IDs and updates the value of the `Threads_running` status variable no longer acquires a global lock that also protects the list of all connections. Instead, it relies on atomic increment and decrement instructions. This improves scalability and to a certain extent alleviates the problem described in Bug #11751904.

  References: See also Bug #42930.

Bugs Fixed

- **Security Fix:** The server crashed if an account with the `CREATE ROUTINE` privilege but not the `EXECUTE` privilege attempted to create a stored procedure. (Bug #44798)

- **Security Enhancement:** When the `DATA DIRECTORY` or `INDEX DIRECTORY` clause of a `CREATE TABLE` statement referred to a subdirectory of the data directory through a symlinked component of the data directory path, it was accepted, when for security reasons it should be rejected. (Bug #39277)

- **Performance; Replication:** When writing events to the binary log, transactional events (that is, events that operate on transactional tables) are written to a thread-specific transaction cache, which is then written to the binary log on commit. To handle nontransactional events, there was a lock taken on the binary log (when entering the function `MYSQL_BIN_LOG::write()`), even when the event was written to the transaction cache instead of the binary log, causing a major bottleneck in replication performance. (Bug #42757)

- **Incompatible Change; Replication:** The `binlog_format` system variable can no longer be set inside a transaction. In other words, the binary logging format can no longer be changed while a transaction is in progress. (Bug #47863)

- **Incompatible Change; Replication:** Concurrent statements using a stored function and `DROP FUNCTION` for that function could break statement-based replication.

  DDL statements for stored procedures and functions are now prohibited while a `LOCK TABLES` statement is in effect. (Bug #30977)

  References: See also Bug #57663.

- **Incompatible Change:** The parser accepted illegal syntax in a `FOREIGN KEY` clause:

  - Multiple `MATCH` clauses.
  - Multiple `ON DELETE` clauses.
  - Multiple `ON UPDATE` clauses.
  - `MATCH` clauses specified after `ON UPDATE` or `ON DELETE`. In case of multiple redundant clauses, this leads to confusion, and implementation-dependent results.

  These illegal syntaxes are now properly rejected. Existing applications that used them will require adjustment. (Bug #34455)

- **Incompatible Change:** The parser accepted an `INTO` clause in nested `SELECT` statements, which is invalid because such statements must return their results to the outer context. This syntax is no longer permitted. (Bug #33204)

- **Incompatible Change:** For application compatibility reasons, when `sql_auto_is_null` is 1, MySQL converts *`auto_inc_col`* `IS NULL` to *`auto_inc_col`* `= LAST_INSERT_ID()`. However, this was being done regardless of whether the predicate was alone or at the top level. Now it occurs only when it is a single top-level predicate.

  In conjunction with this bug fix, the default value of the `sql_auto_is_null` system variable has been changed from 1 to 0, which may cause incompatibilities with existing applications. (Bug #41371)

- **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:** A deadlock occurred for this sequence of events: Session 1 locked a table using `LOCK TABLES`; Session 2 dropped the database containing the table; Session 1 created any database.

  As a consequence of this bug fix, `CREATE DATABASE` is not permitted within a session that has an active `LOCK TABLES` statement. (Bug #49988)

- **Incompatible Change:** For debug builds, attempts to execute `RESET` statements within a transaction that had acquired metadata locks led to an assertion failure.

  As a result of this bug fix, `RESET` statements now cause an implicit commit. (Bug #51336)

- **Incompatible Change:** Due to work done for Bug #989, `FLUSH TABLES` is not permitted when there is an active `LOCK TABLES ... READ`. This caused a problem with `mysqlhotcopy`, which used that sequence of statements. `mysqlhotcopy` now uses `FLUSH TABLES tbl_list WITH READ LOCK` to flush and lock tables. If `mysqlhotcopy` is used with a server older than MySQL 5.5.3 that does not support this statement, it has a new option `--old_server` that causes it to use the previous statement sequence.

  To provide a workaround for the restriction that `FLUSH TABLES` is no longer permitted when there is an active `LOCK TABLES ... READ`, `FLUSH TABLES` has a new variant, `FLUSH TABLES tbl_list WITH READ LOCK`, that enables tables to be flushed and locked in a single operation. As a result of this change, applications that previously used this statement sequence to lock and flush tables will fail:

  ```
  LOCK TABLES tbl_list READ;
  FLUSH TABLES tbl_list;
  ```

  Such applications should now use this statement instead:

  ```
  FLUSH TABLES tbl_list WITH READ LOCK;
  ```

  (Bug #42465)

- **Incompatible Change:** Several changes were made to alias resolution in multiple-table `DELETE` statements so that it is no longer possible to have inconsistent or ambiguous table aliases.

  - In MySQL 5.1.23, alias declarations outside the *`table_references`* part of the statement were disallowed for the `USING` variant of multiple-table `DELETE` syntax, to reduce the possibility of ambiguous aliases that could lead to ambiguous statements that have unexpected results such as deleting rows from the wrong table.

    Now alias declarations outside *`table_references`* are disallowed for all multiple-table `DELETE` statements. Alias declarations are permitted only in the *`table_references`* part.

    Incorrect:

    ```
    DELETE FROM t1 AS a2 USING t1 AS a1 INNER JOIN t2 AS a2;
    DELETE t1 AS a2 FROM t1 AS a1 INNER JOIN t2 AS a2;
    ```

    Correct:

    ```
    DELETE FROM t1 USING t1 AS a1 INNER JOIN t2 AS a2;
    DELETE t1 FROM t1 AS a1 INNER JOIN t2 AS a2;
    ```

- Previously, for alias references in the list of tables from which to delete rows in a multiple-table delete, the default database is used unless one is specified explicitly. For example, if the default database is `db1`, the following statement does not work because the unqualified alias reference `a2` is interpreted as having a database of `db1`:

  ```
  DELETE a1, a2 FROM db1.t1 AS a1 INNER JOIN db2.t2 AS a2
  WHERE a1.id=a2.id;
  ```

  To correctly match an alias that refers to a table outside the default database, you must explicitly qualify the reference with the name of the proper database:

  ```
  DELETE a1, db2.a2 FROM db1.t1 AS a1 INNER JOIN db2.t2 AS a2
  WHERE a1.id=a2.id;
  ```

  Now alias resolution does not require qualification and alias references should not be qualified with the database name. Qualified names are interpreted as referring to tables, not aliases.

Statements containing alias constructs that are no longer permitted must be rewritten. (Bug #27525)

References: See also Bug #30234.

- **Incompatible Change:** `DROP TABLE` now is permitted only if you have acquired a `WRITE` lock with `LOCK TABLES`, or if you hold no locks, or if the table is a `TEMPORARY` table.

  Previously, if other tables were locked, you could drop a table with a read lock or no lock, which could lead to deadlocks between clients. The new stricter behavior means that some usage scenarios will fail when previously they did not. (Bug #25858)

- **Incompatible Change:** If a data definition language (DDL) statement occurred for a table that was being used by another session in an active transaction, statements could be written to the binary log in the wrong order. For example, this could happen if `DROP TABLE` occurred for a table being used in a transaction. This is now prevented by deferring release of metadata locks on tables used within a transaction until the transaction ends.

  This bug fix results in some incompatibilities with previous versions:

  - A table that is being used by a transaction within one session cannot be used in DDL statements by other sessions until the transaction ends.
  - `FLUSH TABLES` is not permitted when there is an active `LOCK TABLES ... READ`. Use `FLUSH TABLES tbl_list WITH READ LOCK` instead. This causes a problem with `mysqlhotcopy`, fixed in Bug #42465.

  (Bug #989, Bug #39675)

- **Incompatible Change:** The `Locked` thread state was equivalent to the `Table lock` state and has been removed. It no longer appears in `SHOW PROCESSLIST` output. (Bug #28870)

- **Important Change; Replication:** For an engine that supported only row-based replication, replication stopped with an error when executing row events.

  For information about changes in how the binary logging format is determined in relation to statement type and storage engine logging capabilities, see Mixed Binary Logging Format.

As part of the fix for this issue, the `EXAMPLE` storage engine is now changed so that it supports statement-based logging only. Previously, it supported row-based logging only. (Bug #39934, Bug #11749859)

- **Important Change:** The IPv6 loopback address `::1` was interpreted as a hostname rather than a numeric IP address.

  In addition, the IPv6-enabled server on Windows interpreted the hostname `localhost` as `::1` only, which failed to match the default `'root'@'127.0.0.1'` account in the `mysql.user` privilege table.

  

  Note

  As a result of this fix, a `'root'@'::1'` account is added to the `mysql.user` table as one of the default accounts created during MySQL installation.

  (Bug #43006)

  References: See also Bug #38247, Bug #11753779, Bug #45584, Bug #45606.

- **InnoDB; Replication:** Column length information generated by `InnoDB` did not match that generated by `MyISAM`, which caused invalid metadata to be written to the binary log when trying to replicate `BIT` columns. (Bug #49618)

- **InnoDB:** `SHOW INNODB STATUS` could display incorrect information about deadlocks, when the deadlock detection routine stops early (to avoid excessive CPU usage). (Bug #49001)

- **InnoDB:** Concurrent execution of `ALTER TABLE` for `InnoDB` table and a transaction that tried to read and then update the table could result in a deadlock between table-level locks and `InnoDB` row locks, which was detected only after the `innodb_lock_wait_timeout` timeout occurred. (Bug #37346)

- **Partitioning:** When used on a partitioned table, `ALTER TABLE` produced the wrong error message when the name of a nonexistent storage engine was used in the `ENGINE` clause. (Bug #35765)

- **Partitioning:** The first time that a query against the `INFORMATION_SCHEMA.TABLES` table for partitioned tables using the `ARCHIVE` engine was run, it returned invalid data. If the server had been restarted since such a table had been created, or if the table had never actually been opened, its `DATA_LENGTH` was reported as 0 bytes. (The second and subsequent attempts to issue the same query returned the expected result.) (Bug #44622)

- **Partitioning:** `ALTER TABLE` on a partitioned table caused unnecessary deadlocks. (Bug #43867)

  References: See also Bug #46654. This bug is a regression of Bug #40181.

- **Partitioning:** After attempting to create a duplicate index on a partitioned table (and having the attempt fail as expected), a subsequent attempt to create a new index on the table caused the server to hang. (Bug #40181)

- **Partitioning:** When using a debug build of MySQL, if a query against a partitioned table having an index on one or more `DOUBLE` columns used that index, the server failed with an assertion. (Bug #45816)

- **Partitioning:** When one user was in the midst of a transaction on a partitioned table, a second user performing an `ALTER TABLE` on this table caused the server to hang. (Bug #34604)

- **Partitioning:** Attempting to drop a partitioned table from one connection while waiting for the completion of an `ALTER TABLE` that had been issued from a different connection, and that changed the storage engine used by the table, could cause the server to crash. (Bug #42438)

- **Partitioning:** Portions of the partitioning code were refactored in response to potential regression issues uncovered while working on the fix for Bug #31210. (Bug #32115)

  References: See also Bug #40281.

- **Replication:** Statements that updated `AUTO_INCREMENT` columns in multiple tables were logged using the row-based format when `--binlog_format` was set to `MIXED`, but did not cause an `Unsafe statement` warning to be generated when `--binlog_format` was set to `STATEMENT`. (Bug #45827)

  References: See also Bug #39934, Bug #11749859.

- **Replication:** When using statement-based replication, database-level character sets were not always honored by the replication SQL thread. This could cause data inserted on the master using `LOAD DATA` to be replicated using the wrong character set.

  

  Note

  This was not an issue when using row-based replication.

  (Bug #45516)

- **Replication:** When using row-based replication, the incomplete logging of a group of events involving both transaction and nontransactional tables could cause `STOP SLAVE` to hang. (Bug #45940)

  References: See also Bug #319, Bug #38205.

- **Replication:** Large transactions and statements could corrupt the binary log if the size of the cache (as set by `max_binlog_cache_size`) was not large enough to store the changes.

  Now, for transactions that do not fit into the cache, the statement is not logged, and the statement generates an error instead.

  For nontransactional changes that do not fit into the cache, the statement is also not logged—an incident event is logged after committing or rolling back any pending transaction, and the statement then raises an error.

  

  Note

  If a failure occurs before the incident event is written the binary log, the slave does not stop, and the master does not report any errors.

  (Bug #43929, Bug #11752675)

  References: See also Bug #37148, Bug #11748696, Bug #46166, Bug #11754544.

- **Replication:** Executing the sequence of statements `RESET SLAVE`, `RESET MASTER`, and `FLUSH LOGS`, when binary log or relay log files listed in the index file could not be found, could cause the server to crash. This could happen, for example, when these files had been moved or deleted manually. (Bug #41902)

- **Replication:** MySQL creates binary logs in a numbered sequence, with a maximum possible 4294967295 concurrent log files, 4294967295 being the maximum value for an unsigned long integer. However, binary log file extensions were turned into negative numbers once the variable used to hold the value reached the maximum value for a signed long integer (2147483647). Consequently, when the sequence value was incremented to the next (negative) number, MySQL tried to create the file using a `.000000` extension, causing the server to fail since this file already existed.

Negative file extensions are no longer permitted, and an error is returned when the limit is reached. In addition, `FLUSH LOGS` now also reports warnings to the user, if the extension number has reached the limit, and warnings are printed to the error log when the limit is approaching. (Bug #40611)

- **Replication:** Issuing concurrent `STOP SLAVE`, `START SLAVE`, and `RESET SLAVE` statements using different connections caused the replication slave to crash. (Bug #38716)

  References: See also Bug #38715, Bug #44312.

- **Replication:** On Windows, `RESET MASTER` failed in the event of a missing binlog file rather than issuing a warning and completing the rest of the statement. (Bug #42150, Bug #42218)

- **Replication:** `mysqlbinlog` sometimes failed when trying to create temporary files; this was because it ignored the specified temp file directory and tried to use the system `/tmp` directory instead. (Bug #35546)

  References: See also Bug #35543.

- **Replication:** A slave compiled using `--with-libevent` and run with `--thread-handling=pool-of-threads` could sometimes crash. (Bug #36929)

- **Replication:** A `CHANGE MASTER TO` statement with no `MASTER_HEARTBEAT_PERIOD` option failed to reset the heartbeat period to its default value. (Bug #34686)

- **Replication:** When using the row-based or mixed replication format with a debug build of the MySQL server, inserts into columns using the `utf32` character set on the master caused the slave to crash. (Bug #51787)

  References: See also Bug #51716.

- **Replication:** When using the row-based or mixed replication format, column values using the `utf16` character set on the master were padded incorrectly on the slave. (Bug #51716)

  References: See also Bug #51787.

- **Replication:** When using the semisynchronous replication plugin on Windows, the wait time calculated when the master was waiting for reply from the slave was incorrect. In addition, when the wait time was less than the current time, the master did not wait for a reply at all.

  This issue was caused by the fact that a different internal function was used to get current time by the plugin on Windows as opposed to other platforms, and this function was not correctly implemented. Now the Windows version of the plugin uses the same function as other platforms for this purpose. (Bug #49557)

- **Replication:** If a `CHANGE MASTER TO` statement set `MASTER_HEARTBEAT_PERIOD` to 30 or higher, `Slave_received_heartbeats` did not increase on the slave. This caused the slave to reconnect before the time indicated by `slave_net_timeout` had elapsed.

  This issue affected big-endian 64-bit platforms such as Solaris/SPARC. (Bug #50296)

- **Replication:** Adding an index to a table on the master caused the slave to stop logging slow queries to the slow query log. (Bug #50620)

- **Replication:** An issue internal to the code, first seen in Bug #49132 but not completely resolved in the fix for that bug, was removed. This should prevent similar issues to those in the previous bug with `binlog_format` changes following DDL statements.

For developers working with the MySQL Server code: the public class variable `THD::current_stmt_binlog_row_based` was supposed to have been removed as part of the fix for Bug #39934, but was still present in the code. If a developer later tried to use this variable, it could cause the previous issues to re-occur, and possibly new ones to arise. The variable has now been removed; the previously added class functions `THD::is_current_stmt_binlog_format_row()`, `THD::set_current_stmt_binlog_format_row()`, and `THD::clear_current_stmt_binlog_format_row()` should be used instead. (Bug #51021)

References: See also Bug #11749859.

- **Replication:** The error message given when trying to replicate (using statement-based mode) insertions into an `AUTO_INCREMENT` column by a stored function or a trigger was improved. (Bug #50192)

- **Replication:** Statement-based replication of user variables having numeric data types did not always work correctly. (Bug #49562, Bug #11757508)

- **Replication:** There were two related issues concerning handling of unsafe statements and setting of the binary logging format when there were open temporary tables on the master, and the existing replication format was row-based or mixed:

  1. When using `binlog_format=ROW`, and an unsafe statement was executed while there were open temporary tables on the master, the statement `SET @@session.binlog_format = MIXED` failed with the error `Cannot switch out of the row-based binary log format when the session has open temporary tables`.

  2. When using `binlog_format=MIXED`, and an unsafe statement was executed while there were open temporary tables on the master, the statement `SET @@session.binlog_format = STATEMENT` caused any subsequent DML statements to be written to the binary log using the row-based format instead of the statement-based format.

  (Bug #45855, Bug #45856)

- **Replication:** The server could deadlock when `FLUSH LOGS` was executed concurrently with DML statements. To fix this problem, nontransactional changes are now always flushed before transactional changes. (Bug #50038)

- **Replication:** Metadata for `GEOMETRY` fields was not properly stored by the slave in its definitions of tables. (Bug #49836)

  References: See also Bug #48776.

- **Replication:** Due to a change in the format of the information used by the slave to connect to the master, which could cause to reject connection attempts to older masters by newer slaves. (Bug #49259)

  References: This bug was introduced by Bug #13963.

- **Replication:** When using row-based logging, a failing `INSERT ... SELECT` statement on a nontransactional table was not flagged correctly, such that, if a rollback was requested and no other nontransactional table had been updated, nothing was written to the binary log. (Bug #47175)

  References: See also Bug #40278.

- **Replication:** Even though `INSERT DELAYED` statements are unsafe for statement-based replication, they caused the statement only to be logged in row format when the binary logging format was `MIXED`, but did not cause a warning to be generated when the binary logging format was `STATEMENT`. (Bug #45825)

- **Replication:** When using `MIXED` binary logging format, statements containing a `LIMIT` clause and occurring in stored routines were not written to the log as row events. (Bug #45785)

- **Replication:** `STOP SLAVE` did not flush the relay log or the `master.info` or `relay-log.info` files, which could lead to corruption if the server crashed. (Bug #44188)

- **Replication:** Formerly, only slaves that had been started with the `--report-hosts` option were visible in the output of `SHOW SLAVE HOSTS`. Now, all slaves that are registered with the master appear in `SHOW SLAVE HOSTS` output.

  As part of the fix for this issue, the `Rpl_recovery_rank` column, which had appeared in the output of `SHOW SLAVE HOSTS` in some MySQL releases, was removed because the corresponding server variable `rpl_recovery_rank` (now deprecated) was never actually used. (Bug #13963)

  References: See also Bug #21132, Bug #21869.

- For an IPv6-enabled MySQL server, privileges specified using standard IPv4 addresses for hosts were not matched (only IPv4-mapped addresses were handled correctly).

  As part of the fix for this bug, a new build option `--disable-ipv6` has been introduced. Compiling MySQL with this option causes all IPv6-specific code in the server to be ignored.

  

  Important

  If the server has been compiled using `--disable-ipv6`, it is not able to resolve hostnames correctly when run in an IPv6 environment.

  (Bug #11754062, Bug #45606)

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

- The hostname cache failed to work correctly. (Bug #45584)

  References: See also Bug #38247, Bug #43006, Bug #11753779, Bug #45606.

- An IPv6-enabled MySQL server did not resolve the IP addresses of incoming connections correctly, with the result that a connection that attempted to match any privilege table entries using fully qualified domain names for hostnames or hostnames using wildcards were dropped. (Bug #38247)

  References: See also Bug #43006, Bug #11753779, Bug #45584, Bug #45606.

- A Windows Installation using the GUI installer would fail with:

  ```
  MySQL Server 5.1 Setup Wizard ended prematurely

  The wizard was interrupted before MySQL Server 5.1. could be completely installed.

  Your system has not been modified. To complete installation at another time, please run
  setup again.

  Click Finish to exit the wizard
  ```

  This was due to a step in the MSI installer that could fail to execute correctly on some environments. (Bug #45418)

- MySQL Server permitted the creation of a merge table based on views but crashed when attempts were made to read from that table. The following example demonstrates this:

  ```
  #Create a test table
  ```

```
CREATE TABLE tmp (id int, c char(2));

#Create two VIEWs upon it
CREATE VIEW v1 AS SELECT * FROM tmp;
CREATE VIEW v2 AS SELECT * FROM tmp;

#Finally create a MERGE table upon the VIEWs
CREATE TABLE merge (id int, c char(2))
ENGINE=MERGE UNION(v1, v2);

#Reading from the merge table lead to a crash
SELECT * FROM merge;
```

The final statement generated the crash. (Bug #44040)

- When archive tables were joined on their primary keys, a query returned no result if the optimizer chose to use this index. (Bug #40677)
- The `CSV` storage engine did not parse `'\X'` characters when they occurred in unquoted fields. (Bug #40814)
- Dropping a locked `Maria` table leads to an assertion failure. (Bug #39395)
- `mysqlbinlog` left temporary files on the disk after shutdown, leading to the pollution of the temporary directory, which eventually caused `mysqlbinlog` to fail. This caused problems in testing and other situations where `mysqlbinlog` might be invoked many times in a relatively short period of time. (Bug #35543)
- Compiling MySQL on FreeBSD failed due to missing definitions for certain network constants. (Bug #34292)
- The parser incorrectly permitted MySQL error code 0 to be specified for a condition handler. (This is incorrect because the condition must be a failure condition and 0 indicates success.) (Bug #36510)
- `mysql_stmt_prepare()` did not reset the list of messages (those messages available using `SHOW WARNINGS`). (Bug #36004)
- Host name lookup failure could lead to a server crash. (Bug #39153)
- Previously, statements inside a stored program did not clear the warning list. For example, warnings or errors generated by statements within a trigger or stored function would be accumulated and added to the message list for the statement that activated the trigger or invoked the function, “polluting” the output of `SHOW WARNINGS` or `SHOW ERRORS` for the outer statement. Normally, messages for a statement that can generate messages replace messages from the previous such statement. The effect was that a statement could have a different effect on the message list depending on whether it executed inside or outside of a stored program.

  Now within a stored program, successive statements that can generate messages update the message list and replace messages from the previous such statement. Only messages from the last of these statements is copied to the message list for the outer statement. (Bug #36649)
- When parsing or formatting interval values of `DAY_MICROSECOND` type, fractional seconds were not handled correctly when more-significant fields were implied or omitted. (Bug #36466)
- Threads were set to the `Table lock` state in such a way that use of this state by other threads to check for a lock wait was subject to a race condition. (Bug #39897)
- `myisampack --join` did not create the destination table `.frm` file. (Bug #36573)
- `mysql_install_db` failed if run as `root` and the root directory (`/`) was not writable. (Bug #36462)

- Sign loss could occur in several contexts:
  - `SEC_TO_TIME()` could lose the sign of negative arguments.
  - `MAKETIME()` could lose the sign of negative arguments.
  - Comparison of `TIME` values could lose the sign of operands.

  (Bug #42661, Bug #42662, Bug #42664)
- The state of a thread for the embedded server was always displayed as `Writing to net`, which is incorrect because there is no network connection for the embedded server. (Bug #41971)
- `mysqld_safe` did not treat dashes and underscores as equivalent in option names. Thanks to Erik Ljungstrom for the patch to fix this bug. (Bug #40368)
- Valgrind warnings that occurred for `SHOW TABLE STATUS` with `InnoDB` tables were silenced. (Bug #38479)
- A natural join of `INFORMATION_SCHEMA` tables could cause an assertion failure. (Bug #43834)
- Execution of `FLUSH TABLES` or `FLUSH TABLES WITH READ LOCK` concurrently with `LOCK TABLES` resulted in deadlock. (Bug #45066)
- Plugin shutdown could lead to an assertion failure caused by using an already destroyed mutex in the metadata locking subsystem. (Bug #39674)
- Killing a delayed-insert thread could cause a server crash. (Bug #45067)
- The patch for Bug #10374 broke named-pipe and shared-memory connections on Windows. (Bug #41860)
- The `mysql_real_connect()` C API function only attempted to connect to the first IP address returned for a hostname. This could be a problem if a hostname mapped to multiple IP address and the server was not bound to the first one returned. Now `mysql_real_connect()` attempts to connect to all IPv4 or IPv6 addresses that a domain name maps to. (Bug #45017)

  References: See also Bug #47757.
- The server could crash if an attempt to open a `MERGE` table child `MyISAM` table failed. (Bug #42862)
- The server could crash attempting to flush privileges after receipt of a `SIGHUP` signal. (Bug #46495)
- Improperly closing tables when `INSERT DELAYED` needed to reopen tables could cause an assertion failure. (Bug #45949)

  References: See also Bug #18484.
- `HANDLER` statements are now not permitted if a table lock has been acquired with `LOCK TABLES`. (Bug #43272)
- An assertion failure could occur if `OPTIMIZE TABLE` was started on an `InnoDB` table and the table was altered to a different storage engine during the optimization operation. (Bug #42074)
- Deadlock occurred if one session was running a multiple-statement transaction that involved a single partitioned table and another session attempted to alter the table. (Bug #46654)
- If `INSERT INTO` *`tbl_name`* invoked a stored function that modified *`tbl_name`*, the server crashed. (Bug #46374)

- `CREATE VIEW` raised an assertion if a temporary table existed with the same name as the view. (Bug #47635)
- Selecting from the process list in the embedded server caused a crash. (Bug #47304)

  References: See also Bug #43733.
- If a temporary table was created with the same name as a view referenced in a stored routine, routine execution could raise an assertion. (Bug #47313)
- Programs did not exit if the option file specified by `--defaults-file` was not found. (Bug #47216)
- For queries that used `GROUP_CONCAT(DISTINCT ...)`, the value of `max_heap_table_size` was used for memory allocation, which could be excessive. Now the minimum of `max_heap_table_size` and `tmp_table_size` is used. (Bug #46018)
- Creation of a temporary `BLOB` or `TEXT` column could create a column with the wrong maximum length. (Bug #33969)
- Valgrind warnings for several logging messages were corrected. (Bug #49130)
- Constant expressions in `WHERE`, `HAVING`, or `ON` clauses were not cached, but were evaluated for each row. This caused a slowdown of query execution, especially if constant user-defined functions or stored functions were used. (Bug #33546)
- The `mysql` could default to the `ascii` character set, which is not a valid character set choice for MySQL. The `latin1` character set will now be used when an ASCII environment has been identified. (Bug #51166)
- `SHOW CREATE VIEW` returned invalid SQL if the definition contained a `SELECT 'string'` statement where the *`string`* was longer than the maximum length of a column name, due to the fact that this text was also used as an alias (in the `AS` clause).

  Because not all names retrieved from arbitrary `SELECT` statements can be used as view column names due to length and format restrictions, the server now checks the conformity of automatically generated column names and rewrites according to a predefined format any names that are not acceptable as view column names before storing the final view definition on disk.

  In such cases, the name is now rewritten as `Name_exp_`*`pos`*, where *`pos`* is the position of the column. To avoid this conversion scheme, define explicit, valid names for view columns using the *`column_list`* clause of the `CREATE VIEW` statement.

  As part of this fix, aliases are now generated only for top-level statements. (Bug #40277)
- Running `SHOW CREATE TABLE` on a view `v1` that contained a function which accessed another view `v2` could trigger a infinite loop if the view referenced within the function (`v2`) caused a warning to be raised while being opened. (Bug #48449)
- If a prepared statement used both a `MERGE` table and a stored function or trigger, execution sometimes failed with a `No such table` error. (Bug #47648)
- An `.ARZ` file missing from the database directory caused the server to crash. (Bug #48757)
- When building MySQL when using a different target directory (for example using the `VPATH` environment variable), the build of the embedded `readline` component failed. (Bug #35250)
- On some Unix/Linux platforms, an error during build from source could be produced, referring to a missing `LT_INIT` program. This is due to versions of `libtool` 2.1 and earlier. (Bug #51009)

- Corrected a potential problem of unintended file overwriting when the `MY_DONT_OVERWRITE_FILE` flag was used. (Bug #47126)
- If a stored function contained a `RETURN` statement with an `ENUM` value in the `ucs2` character set, `SHOW CREATE FUNCTION` and `SELECT DTD_IDENTIFIER FROM INFORMATION_SCHEMA.ROUTINES` returned incorrect values. (Bug #48766)
- An aliasing violation in the C API could lead to a crash. (Bug #48284)
- A dependent subquery containing `COUNT(DISTINCT col_name))` could be evaluated incorrectly. (Bug #48920)
- `mysqltest` no longer permits you to execute an SQL statement on a connection after doing a `send` command, unless you do a `reap` first. This was previously accepted but could produce unpredictable results. (Bug #49269)
- `InnoDB` took a shared row lock when executing `SELECT` statements inside a stored function as a part of a transaction using `REPEATABLE READ`. This prevented other transactions from updating the row. (Bug #44613)
- The return values for calls to put information into the stored routine cache were not consistently checked, raising an assertion. (Bug #50412)
- Purging the stored-routine cache could take a long time and render the server unresponsive. (Bug #41804)
- When used in conjunction with `LOCK TABLES`, `FLUSH TABLE tbl_list` waited for all tables with old versions to clear from the table definition list, rather than only the named tables. (Bug #43685)
- `HANDLER` statements within a transaction that already holds metadata locks could lead to deadlocks.

  Before this fix, all handlers for `TEMPORARY` tables were reset whenever any base table was opened. (Bug #46224)
- There was no timeout for attempts to acquire metadata locks (for example, a `DROP TABLE` attempt for a table that was open in another transaction would not time out).

  To handle such situations, there is now a `lock_wait_timeout` system variable that specifies the timeout in seconds for attempts to acquire metadata locks. The permitted values range from 1 to 31536000 (1 year). The default is 31536000.

  This timeout applies to all statements that use metadata locks. These include DML and DDL operations on tables, views, stored procedures, and stored functions, as well as `LOCK TABLES`, `FLUSH TABLES WITH READ LOCK`, and `HANDLER` statements.

  The timeout value applies separately for each metadata lock attempt. A given statement can require more than one lock, so it is possible for the statement to block for longer than the `lock_wait_timeout` value before reporting a timeout error. When lock timeout occurs, `ER_LOCK_WAIT_TIMEOUT` is reported.

  `lock_wait_timeout` does not apply to delayed inserts, which always execute with a timeout of 1 year. This is done to avoid unnecessary timeouts because a session that issues a delayed insert receives no notification of delayed insert timeouts.

  In addition: The unused `table_lock_wait_timeout` system variable was removed. The `LOW_PRIORITY` modifier for `LOCK TABLES ... WRITE` locks now has no effect. The meaning of `LOW_PRIORITY` remains as before in other contexts, such as for `INSERT` or `DELETE` statements. `innodb_table_locks=0` no longer has an effect for tables locked explicitly with `LOCK TABLES ...`

WRITE. It still has an effect for tables locked for read or write by LOCK TABLES ... WRITE implicitly (for example, through triggers) or by LOCK TABLES ... READ. (Bug #45225, Bug #56272)

- mysqld_multi failed due to a syntax error in the script. (Bug #51468)
- On POSIX systems, calls to select() with a file descriptor set larger than FD_SETSIZE resulted in unpredictable I/O errors; for example, when a large number of tables required repair. (Bug #48929)
- For debug builds, SHOW BINARY LOGS raised an assertion if binary logging was not enabled. (Bug #50780)
- Full-text queries that used the truncation operator (*) could enter an infinite loop. (Bug #50351)
- For debug builds on Windows, warnings about incorrect use of debugging directives were written to the error log. The directives were rewritten to eliminate these messages. (Bug #49025)
- Slow CALL statements were not always logged to the slow query log because execution time for multiple-statement stored procedures was assessed incorrectly. (Bug #47905)
- The optimizer normally prefers use of filesort plus the join cache to a full index scan. But this combination was used even if the index is clustered, in which case, the clustered index scan can be faster. (Bug #50843)
- Incorrect handling of BIT columns in temporary tables could lead to spurious duplicate-key errors. (Bug #50591)
- Referring to a subquery result in a HAVING clause could produce incorrect results. (Bug #50995)
- mysql --show-warnings crashed if the server connection was lost. (Bug #49646)
- When read_only was enabled, the server incorrectly prevented data modifications to TEMPORARY tables belonging to transactional storage engines such as InnoDB. (Bug #33669)
- ALTER TABLE on a MERGE table that has been locked using LOCK TABLES ... WRITE incorrectly produced an ER_TABLE_NOT_LOCKED_FOR_WRITE error. (Bug #51240)
- Setting key_buffer_size to a negative value could lead to very large allocations. Now an error occurs. (Bug #42103)
- In the embedded server, stack overflow checks for recursive stored procedure calls did not work and stack overflow could occur. (Bug #43201)
- Some plugins configured as mandatory could be disabled at server startup. (Bug #44691)
- Attempts to print octal numbers with my_vsnprintf() could cause a crash. (Bug #47212)
- Setting binlog_format to DEFAULT assigned a value different from the default. (Bug #49540)
- SHOW VARIABLES did not correctly display string-valued system variables that contained \0 characters. (Bug #49644)
- MySQL program option-processing code incorrectly displayed some options when printing ambiguous-option errors. (Bug #49640)
- For string-valued system variables containing multi-byte characters, the byte length was used in contexts where the character length was more appropriate. (Bug #49645)
- Failure to open a view with a nonexistent DEFINER was improperly handled and the server crashed later attempting to lock the view. (Bug #47734)

- `flush_cache_records()` did not correctly check for errors that should cause statement execution to stop, leading to a server crash. (Bug #39022)
- `INSERT INTO ... VALUES(DEFAULT)` failed to insert the correct value for `ENUM` columns. For `MyISAM` tables, an empty value was inserted. For `CSV` tables, the table became corrupt. (Bug #33717)
- The `sql_mode` system variable could be assigned the illegal value of `'?'`. (Bug #34834)
- String-valued system variables could be assigned literal values, but could not be assigned values using expressions. Now expressions are legal. (Bug #34883, Bug #46314)
- Valgrind warnings about uninitialized variables in optimizer code were corrected. (Bug #45195)
- Propagation of a large unsigned numeric constant in `WHERE` expressions could lead to incorrect results. This also affected `EXPLAIN EXTENDED`, which printed incorrect numeric constants in such transformed `WHERE` expressions. (Bug #45360)
- Grouping by a subquery in a query with a `DISTINCT` aggregate function led to incorrect and unordered grouping values. (Bug #45640)
- Valgrind warnings about memory allocation overruns for handling `CREATE FUNCTION` statements for UDFs were corrected. (Bug #46570)
- The server did not recognize that the stored procedure cache became invalid if a view was created or modified within a procedure, resulting in a crash. (Bug #50624)
- The second or subsequent invocation of a stored procedure containing `DROP TRIGGER` could cause a server crash. (Bug #50423)
- For debug builds, an assertion was incorrectly raised in the optimizer when matching `ORDER BY` expressions. (Bug #50335)
- Queries optimized with `GROUP_MIN_MAX` did not clean up `KEYREAD` optimizations properly, causing subsequent queries to return incomplete rows. (Bug #49902)
- For dynamic format `MyISAM` tables containing `LONGTEXT` columns, a bulk `INSERT ... ON DUPLICATE KEY UPDATE` or bulk `REPLACE` could cause corruption. (Bug #49628)
- For debug builds, with `sql_safe_updates` enabled, a multiple-table `UPDATE` with the `IGNORE` modifier could raise an assertion. (Bug #49534)
- For debug builds, killing a `SELECT` retrieving from a view that was processing a function raised an assertion. (Bug #47736)
- For debug builds, creating a view containing a row constructor raised an assertion. (Bug #48294)
- Invalid memory reads could occur following a query that referenced a `MyISAM` table multiple times with a write lock. (Bug #48438)
- `EXPLAIN EXTENDED` crashed trying to print column names for a subquery in the `FROM` clause when the table had gone out of scope. (Bug #49487)
- Renaming a column of an `InnoDB` table caused the server to go out of sync with the `InnoDB` data dictionary. To avoid this issue, renaming a column uses the older technique of copying all the table data rather than updating the table in-place. (Bug #47621)
- `InnoDB` logged an error repeatedly trying to load a page into the buffer pool, filling the error log and using excessive disk space. Now the number of attempts is limited to 100, after which the operation aborts with a message. (Bug #38901)

- Command-line options for enumeration-type plugin variables were not honored. (Bug #41010)
- For plugins that did not have command-line options other than the ones to select the plugin itself, those options were not displayed in the `mysqld` help message. (Bug #44797)
- System variables could be set to invalid values. (Bug #40988)
- Plugins could find the unqualified form of their system variables but not the qualified form. For example, a plugin `p` with a system variable `sv` could find `sv` but not `p_sv`. (Bug #32902)
- A global read lock obtained with `FLUSH TABLES WITH READ LOCK` did not prevent sessions from creating tables. (Bug #35935)
- For `InnoDB` tables, the test for using an index for `ORDER BY` sorting did not distinguish between primary keys and secondary indexes and expected primary key values to be concatenated to index values the way they are to secondary key values. (Bug #49324)
- For `CREATE TABLE ... LIKE` with a `MERGE` source table that included a `UNION` clause, that clause was omitted from the definition of the destination table. (Bug #37371)
- `mysqld_safe` did not pass the correct default value of `plugin_dir` to `mysqld`. (Bug #51938)
- Plugins in a binary release could not be installed into a debug version of the server. (Bug #49022)
- Aggregate functions on `TIMESTAMP` columns could yield incorrect or undefined results. (Bug #50888)
- If an operation had an `InnoDB` table, and two triggers, `AFTER UPDATE` and `AFTER INSERT`, competing for different resources (such as two distinct `MyISAM` tables), the triggers were unable to execute concurrently. In addition, `INSERT` and `UPDATE` statements for the `InnoDB` table were unable to run concurrently. (Bug #26141)
- The character set was not being properly initialized for `CAST()` with a type such as `CHAR(2) BINARY`, which resulted in incorrect results or a server crash. (Bug #17903)
- Labels in stored routines did not work if the character set was not `latin1`. (Bug #7088)
- `mysqld` sometimes miscalculated the number of digits required when storing a floating-point number in a `CHAR` column. This caused the value to be truncated, or (when using a debug build) caused the server to crash. (Bug #26788)

  References: See also Bug #12860.
- When inserting an extraordinarly large value into a `DOUBLE` column, the value could be truncated in such a way that the new value cannot be reloaded manually or from the output of `mysqldump`. (Bug #21497)
- Zero-padding of exponent values was not the same across platforms. (Bug #12860)
- `Data truncated for column col_num at row row_num` warnings were generated for some (constant) values that did not have too high precision. (Bug #24541)
- The grammar for `GROUP BY`, when used with `WITH CUBE` or `WITH ROLLUP`, caused a conflict with the grammar for view definitions that included `WITH CHECK OPTION`. (Bug #9801)
- Stored procedure exception handlers were catching fatal errors (such as out of memory errors), which could cause execution not to stop to due a continue handler. Now fatal errors are not caught by exception handlers and a fatal error is returned to the client. (Bug #15192)
- Delayed-insert threads were counted as connected but not as created, incorrectly leading to a `Threads_connected` value greater than the `Threads_created` value. (Bug #17954)

- If a connection was waiting for a `GET_LOCK()` lock or a `SLEEP()` call, and the connection aborted, the server did not detect this and thus did not close the connection. This caused a waste of system resources allocated to dead connections. Now the server checks such a connection every five seconds to see whether it has been aborted. If so, the connection is killed (and any lock request is aborted). (Bug #10374)
- Killing a statement that invoked a stored function could return an incorrect error message indicating table corruption rather than that the statement had been interrupted. (Bug #32140)
- Occurrence of an error within a stored routine did not always cause immediate statement termination. (Bug #31881)
- Statements to create, alter, or drop a view were not waiting for completion of statements that were using the view, which led to incorrect sequences of statements in the binary log when statement-based logging was enabled. (Bug #25144)
- For `DROP FUNCTION db_name.func_name` (that is, when the function name is qualified with the database name), the statement should apply only to a stored function named `func_name` in the given database. However, if a UDF with the same name existed, the statement dropped the UDF instead. (Bug #31767)
- `perror` did not work for errors described in the `sql/share/errmsg.txt` file. (Bug #10143)
- For `CREATE TABLE`, the parser did not enforce that parentheses were present in a `CHECK (expr)` clause; now it does. The parser did not enforce that `CONSTRAINT [symbol]` without a following `CHECK` clause was illegal; now it does. (Bug #11714, Bug #35578, Bug #38696)
- `ALTER TABLE` could not be used to add columns to a table if the table had an index on a `utf8` column with a `TEXT` data type. (Bug #26180)
- For the `DIV` operator, incorrect results could occur for noninteger operands that exceed `BIGINT` range. Now, if either operand has a noninteger type, the operands are converted to `DECIMAL` and divided using `DECIMAL` arithmetic before converting the result to `BIGINT`. If the result exceeds `BIGINT` range, an error occurs. (Bug #8457, Bug #11745058)

  References: See also Bug #59241.
- Several data-modification statements were not being counted toward the `MAX_UPDATES_PER_HOUR` user resource limit. (Bug #21793)
- Previously, the server handled character data types for a stored routine parameter, local routine variable created with `DECLARE`, or stored function return value as follows: If the `CHARACTER SET` attribute was present, the `COLLATE` attribute was not supported, so the character set's default collation was used. (This includes use of `BINARY`, which in this context specifies the binary collation of the character set.) If there was no `CHARACTER SET` attribute, the database character set and its default collation were used.

  Now for character data types, if there is a `CHARACTER SET` attribute in the declaration, the specified character set and its default collation is used. If the `COLLATE` is also present, that collation is used rather than the default collation. If there is no `CHARACTER SET` attribute, the database character set and collation in effect at routine creation time are used. (The database character set and collation are given by the value of the `character_set_database` and `collation_database` system variables.) (Bug #24690)
- A statement that caused a circular wait among statements did not return a deadlock error. Now the server detects deadlock and returns `ER_LOCK_DEADLOCK`. (Bug #22876)
- For `INSERT DELAYED` statements issued for a table while an `ALTER TABLE` operation on the table was in progress, the server could return a spurious `Server shutdown in progress` error. (Bug #18484)

References: See also Bug #45949.

- The value of `sql_slave_skip_counter` was empty when displayed by `SHOW VARIABLES` or `INFORMATION_SCHEMA.GLOBAL_VARIABLES`. (Bug #20413, Bug #37187)

- Previously, for some Asian CJK character sets, the `UPPER()` and `LOWER()` functions worked only for basic Latin letters (`A-Z`, `a-z`). The affected character sets are `ujis`, `sjis`, `gb2312`, `cp932`, `eucjpms`, `big5`, `euckr`, and `gbk`.

  Now `UPPER()` and `LOWER()` perform case conversion correctly for all characters in these character sets, with the exception that if a character set contains a character in only one lettercase, conversion to the other lettercase cannot be done.

- `CREATE TABLE ... LIKE` did not always produce an error is the source table column defaults were illegal for the current version of MySQL. (This could occur if the table was created using an older server that was less restrictive about legal default values.) (Bug #22090)

Changes in MySQL 5.5.2 (2010-02-12)

InnoDB Notes

- This release includes `InnoDB` 1.0.6. This version is considered of Release Candidate (RC) quality.

Functionality Added or Changed

- **Replication:** Introduced the `binlog_direct_non_transactional_updates` system variable. Enabling this variable causes updates using the statement-based logging format to tables using nontransactional engines to be written directly to the binary log, rather than to the transaction cache.

  Before enabling this variable, be certain that you have no dependencies between transactional and nontransactional tables. A statement that both selects from an `InnoDB` table and inserts into a `MyISAM` table is an example of such a dependency. For more information, see Binary Log Options and Variables. (Bug #46364)

  References: See also Bug #28976, Bug #40116.

Bugs Fixed

- **Security Fix:** For servers built with yaSSL, a preauthorization buffer overflow could cause memory corruption or a server crash. We thank Evgeny Legerov from Intevydis for providing us with a proof-of-concept script that permitted us to reproduce this bug. (Bug #50227, CVE-2009-4484)

- **Performance; Partitioning:** When used on partitioned tables, the `records_in_range` handler call checked more partitions than necessary. The fix for this issue reduces the number of unpruned partitions checked for statistics in partition range checking, which has resulted in some partition operations being performed up to 2-10 times faster than before this change was made, when testing with tables having 1024 partitions. (Bug #48846)

  References: See also Bug #37252, Bug #47261.

- **Performance:** The `MD5()` and `SHA1()` functions had excessive overhead for short strings. (Bug #49491, Bug #11757443, Bug #60227, Bug #14134662)

- **Performance:** The method for comparing `INFORMATION_SCHEMA` names and database names was nonoptimal and an improvement was made: When the database name length is already known, a length check is made first and content comparison skipped if the lengths are unequal. (Bug #49501)

- **Incompatible Change:** In `plugin.h`, the `MYSQL_REPLICATION_PLUGIN` symbol was out of synchrony with its value in MySQL 6.0 because the lower-valued `MYSQL_AUDIT_PLUGIN` was not present. To correct this, `MYSQL_AUDIT_PLUGIN` has been added in MySQL 5.5, changing the value of `MYSQL_REPLICATION_PLUGIN` from 5 to 6. Attempts to load the audit plugin produce an error occurs because only the `MYSQL_AUDIT_PLUGIN` symbol was added, not the audit plugin itself. This error will go away when the audit plugin is added to MySQL 5.5 (in 5.5.3). Replication plugins from earlier 5.5.x releases must be recompiled against the current release before they will work with the current release. (Bug #49894)

- **Important Change; Replication:** The `RAND()` function is now marked as unsafe for statement-based replication. Using this function now generates a warning when `binlog_format=STATEMENT` and causes the format to switch to row-based logging when `binlog_format=MIXED`.

  This change is being introduced because, when `RAND()` was logged in statement mode, the seed was also written to the binary log, so the replication slave generated the same sequence of random numbers as was generated on the master. While this could make replication work in some cases, the order of affected rows was still not guaranteed when this function was used in statements that could update multiple rows, such as `UPDATE` or `INSERT ... SELECT`; if the master and the slave retrieved rows in different order, they began to diverge. (Bug #49222)

- **InnoDB; Partitioning:** When an `ALTER TABLE ... REORGANIZE PARTITION` statement on an `InnoDB` table failed due to `innodb_lock_wait_timeout` expiring while waiting for a lock, `InnoDB` did not clean up any temporary files or tables which it had created. Attempting to reissue the `ALTER TABLE` statement following the timeout could lead to storage engine errors, or possibly a crash of the server. (Bug #47343)

- **InnoDB:** Creating or dropping a table with 1023 transactions active caused an assertion failure. (Bug #49238)

- **InnoDB:** If `innodb_force_recovery` was set to 4 or higher, the server could crash when opening an `InnoDB` table containing an auto-increment column. MySQL versions 5.1.31 and later were affected. (Bug #46193)

- **Replication:** `FLUSH LOGS` could in some circumstances crash the server. This occurred because the I/O thread could concurrently access the relay log I/O cache while another thread was performing the `FLUSH LOGS`, which closes and reopens the relay log and, while doing so, initializes (or re-initializes) its I/O cache. This could cause problems if some other thread (in this case, the I/O thread) is accessing it at the same time.

  Now the thread performing the `FLUSH LOGS` operation takes a lock on the relay log before actually flushing it. (Bug #50364)

  References: See also Bug #53657.

- **Replication:** In some cases, inserting into a table with many columns could cause the binary log to become corrupted. (Bug #50018)

  References: See also Bug #42749.

- **Replication:** When using row-based replication, setting a `BIT` or `CHAR` column of a `MyISAM` table to `NULL`, then trying to delete from the table, caused the slave to fail with the error `Can't find record in table`. (Bug #49481, Bug #49482)

- **Replication:** When logging in row-based mode, DDL statements are actually logged as statements; however, statements that affected temporary tables and followed DDL statements failed to reset the binary log format to `ROW`, with the result that these statements were logged using the statement-based

format. Now the state of `binlog_format` is restored after a DDL statement has been written to the binary log. (Bug #49132)

- **Replication:** When using row-based logging, the statement `CREATE TABLE t IF NOT EXIST ... SELECT` was logged as `CREATE TEMPORARY TABLE t IF NOT EXIST ... SELECT` when `t` already existed as a temporary table. This was caused by the fact that the temporary table was opened and the results of the `SELECT` were inserted into it when a temporary table existed and had the same name.

  Now, when this statement is executed, `t` is created as a base table, the results of the `SELECT` are inserted into it—even if there already exists a temporary table having the same name—and the statement is logged correctly. (Bug #47418)

  References: See also Bug #47442.

- **Replication:** Due to a change in the size of event representations in the binary log, when replicating from a MySQL 4.1 master to a slave running MySQL 5.0.60 or later, the `START SLAVE UNTIL` statement did not function correctly, stopping at the wrong position in the log. Now the slave detects that the master is using the older version of the binary log format, and corrects for the difference in event size, so that the slave stops in the correct position. (Bug #47142)

- **Replication:** A `LOAD DATA INFILE` statement that loaded data into a table having a column name that had to be quoted (such as `` `key` INT``) caused replication to fail when logging in mixed or statement mode. In such cases, the master wrote the `LOAD DATA` event into the binary log without quoting the column name. (Bug #49479)

  References: See also Bug #47927. This bug is a regression of Bug #43746.

- **Replication:** Spatial data types caused row-based replication to crash. (Bug #48776)

- **Replication:** With semisynchronous replication, memory allocated for handling transactions could be freed while still in use, resulting in a server crash. (Bug #50157)

- **Replication:** Manually removing entries from the binary log index file on a replication master could cause the server to repeatedly send the same binary log file to slaves. (Bug #28421)

- A race condition on the privilege hash tables permitted one thread to try to delete elements that had already been deleted by another thread. A consequence was that `SET PASSWORD` or `FLUSH PRIVILEGES` could cause a crash. (Bug #35589, Bug #35591)

- The `printstack` function does not exist on Solaris 8 or earlier, which led to a compilation failure. (Bug #50409)

- If an invocation of a stored procedure failed in the table-open stage, subsequent invocations that did not fail in that stage could cause a crash. (Bug #47649)

- When the `mysql` client was invoked with the `--vertical` option, it ignored the `--skip-column-names` option. (Bug #47147)

- On Fedora 12, `strmov()` did not guarantee correct operation for overlapping source and destination buffer. Calls were fixed to use an overlap-safe version instead. (Bug #48866)

- Within a stored routine, selecting the result of `CONCAT_WS()` with a routine parameter argument into a user variable could return incorrect results. (Bug #50096)

- Several `strmake()` calls had an incorrect length argument (too large by one). (Bug #48983)

- Some prepared statements could raise an assertion when re-executed. (Bug #49570)

- `EXPLAIN EXTENDED UNION ... ORDER BY` caused a crash when the `ORDER BY` referred to a nonconstant or full-text function or a subquery. (Bug #49734)
- With one thread waiting for a lock on a table, if another thread dropped the table and created a new table with the same name and structure, the first thread did not notice that the table had been re-created and tried to used cached metadata that belonged to the old table but had been freed. (Bug #48157)
- The SSL certificates in the test suite were about to expire. They have been updated with expiration dates in the year 2015. (Bug #50642)
- Debug output for join structures was garbled. (Bug #50271)
- A user could see tables in `INFORMATION_SCHEMA.TABLES` without appropriate privileges for them. (Bug #50276)
- `mysql-test-run.pl` now recognizes the `MTR_TESTCASE_TIMEOUT`, `MTR_SUITE_TIMEOUT`, `MTR_SHUTDOWN_TIMEOUT`, and `MTR_START_TIMEOUT` environment variables. If they are set, their values are used to set the `--testcase-timeout`, `--suite-timeout`, `--shutdown-timeout`, and `--start-timeout` options, respectively. (Bug #49210)
- If `EXPLAIN` encountered an error in the query, a memory leak occurred. (Bug #45989)
- 1) In rare cases, if a thread was interrupted during a `FLUSH PRIVILEGES` operation, a debug assertion occurred later due to improper diagnostics area setup. 2) A `KILL` operation could cause a console error message referring to a diagnostic area state without first ensuring that the state existed. (Bug #33982)
- Mixing full-text searches and row expressions caused a crash. (Bug #49445)
- A crash occurred when a user variable that was assigned to a subquery result was used as a result field in a `SELECT` statement with aggregate functions. (Bug #47371)
- The `filesort` sorting method applied to a `CHAR(0)` column could lead to a server crash. (Bug #49897)
- `sql_buffer_result` had an effect on non-`SELECT` statements, contrary to the documentation. (Bug #49552)
- In some cases a subquery need not be evaluated because it returns only aggregate values that can be calculated from table metadata. This sometimes was not handled by the enclosing subquery, resulting in a server crash. (Bug #49512)
- `SPATIAL` indexes were permitted on columns with nonspatial data types, resulting in a server crash for subsequent table inserts. (Bug #50574)
- Index prefixes could be specified with a length greater than the associated column, resulting in a server crash for subsequent table inserts. (Bug #50542)
- Use of loose index scan optimization for an aggregate function with `DISTINCT` (for example, `COUNT(DISTINCT)`) could produce incorrect results. (Bug #50539)
- The server crashed when an `InnoDB` background thread attempted to write a message containing a partitioned table name to the error log. (Bug #50201)
- The optimizer could continue to execute a query after a storage engine reported an error, leading to a server crash. (Bug #46175)
- The `Table_locks_waited` waited variable was not incremented in the cases that a lock had to be waited for but the waiting thread was killed or the request was aborted. (Bug #30331)

- `ALTER TABLE` with both `DROP COLUMN` and `ADD COLUMN` clauses could crash or lock up the server. (Bug #31145)

Changes in MySQL 5.5.1 (2010-01-04)

When the publishing process for MySQL 5.5.1 was already running, the MySQL team was informed about a security problem in the SSL connect area (a possibility to crash the server). The problem is caused by a buffer overflow in the yaSSL library. MySQL Servers using OpenSSL are not affected; it can occur only when SSL (using yaSSL) is enabled.

This problem is under detailed investigation with the various versions, configurations, and platforms. When that has finished, the problem will be fixed as soon as possible, and new binaries for the affected versions will be released. However, building and testing these binaries in the various configurations on the various platforms will take some time. The bug is tracked with CVE ID CVE-2009-4484. We repeat the general security hint: If it is not *absolutely* necessary that external machines can connect to your database instance, we recommend that the server's connection port be blocked by a firewall to prevent any such illegitimate accesses.

Update: This bug is fixed in MySQL 5.5.2.

InnoDB Notes

- `InnoDB` has been upgraded to version 1.0.6. This version is considered of Release Candidate (RC) quality. InnoDB Storage Engine Change History, may contain information in addition to those changes reported here.

RPM Notes

- The version information in RPM package files has been changed:
  - The “level” field of a MySQL version number is now also included in the RPM version and in the package file name.
  - The RPM “release” value now counts from 1, not 0.

  For example, the generic x86 server RPM file of 5.5.1-m2 is named `MySQL-server-5.5.1_m2-1.glibc23.i386.rpm`. This improves consistency with other formats that also include the level in the file name (for this version: “m2”). For example, the `tar.gz` file name is `mysql-5.5.1-m2-linux-i686-glibc23.tar.gz`. The different separator, underscore `'_'` for RPM, is required by the syntax of RPM.

Functionality Added or Changed

- **Partitioning:** The `UNIX_TIMESTAMP()` function is now supported in partitioning expressions using `TIMESTAMP` columns. For example, it now possible to create a partitioned table such as this one:

```
CREATE TABLE t (c TIMESTAMP)
PARTITION BY RANGE ( UNIX_TIMESTAMP(c) ) (
    PARTITION p0 VALUES LESS THAN (631148400),
    PARTITION p1 VALUES LESS THAN (946681200),
    PARTITION p2 VALUES LESS THAN (MAXVALUE)
);
```

  All other expressions involving `TIMESTAMP` values are now rejected with an error for attempts to create a new partitioned table or to alter an existing partitioned table.

  When accessing an existing partitioned table having a timezone-dependent partitioning function (where the table was using a previous version of MySQL), a warning rather than an error is issued. In such

cases, you should fix the table. One way of doing this is to alter the table's partitioning expression so that it uses `UNIX_TIMESTAMP()`. (Bug #42849)

Bugs Fixed

- **Performance:** When the query cache is fragmented, the size of the free block lists in the memory bins grows, which causes query cache invalidation to become slow. There is now a 50ms timeout for a `SELECT` statement waiting for the query cache lock. If the timeout expires, the statement executes without using the query cache. (Bug #39253)

  References: See also Bug #21074.

- **Incompatible Change; Replication:** The file names for the semisynchronous plugins were prefixed with `lib`, unlike file names for other plugins. The file names no longer have a `lib` prefix.

  This change introduces an incompatibility if the plugins had been installed using the previous names. To handle this, uninstall the older version before installing the newer version. For example, use these statements for the master side plugins on Unix:

  ```
  mysql> UNINSTALL PLUGIN rpl_semi_sync_master;
  mysql> INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';
  ```

  If you do not uninstall the older version first, attempting to install the newer version results in an error:

  ```
  mysql> INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';
  ERROR 1125 (HY000): Function 'rpl_semi_sync_master' already exists
  ```

  For the slave side, similar statements apply:

  ```
  mysql> UNINSTALL PLUGIN rpl_semi_sync_slave;
  mysql> INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';
  ```

- **Important Change; Replication:** The following functions have been marked unsafe for statement-based replication:

  - `GET_LOCK()`
  - `IS_FREE_LOCK()`
  - `IS_USED_LOCK()`
  - `MASTER_POS_WAIT()`
  - `RELEASE_LOCK()`
  - `SLEEP()`
  - `SYSDATE()`
  - `VERSION()`

  None of the functions just listed are guaranteed to replicate correctly when using the statement-based format because they can produce different results on the master and the slave. The use of any of these functions while `binlog_format` is set to `STATEMENT` is logged with the warning, `Statement is not safe to log in statement format`. When `binlog_format` is set to `MIXED`, the binary logging

format is automatically switched to the row-based format whenever one of these functions is used. (Bug #47995)

- **Important Change:** After a binary upgrade to MySQL 5.1 from a MySQL 5.0 installation that contains `ARCHIVE` tables:

  - Before MySQL 5.1.42, accessing those tables will cause the server to crash, even if you have run `mysql_upgrade` or `CHECK TABLE ... FOR UPGRADE`.

  - As of MySQL 5.1.42, the server will not open 5.0 `ARCHIVE` tables at all.

  In either case, the solution is to use `mysqldump` to dump all 5.0 `ARCHIVE` tables before upgrading, and reload them into MySQL 5.1 after upgrading. The same problem occurs for binary downgrades from MySQL 5.1 to 5.0. (Bug #47012)

- **InnoDB:** When compiling on Windows, an error in the `CMake` definitions for `InnoDB` caused the engine to be built incorrectly. (Bug #49502)

- **Partitioning:** When `SHOW CREATE TABLE` was invoked for a table that had been created using the `COLUMNS` keyword or the `TO_SECONDS()` function, the output contained the wrong MySQL version number in the conditional comments. (Bug #49591)

- **Partitioning:** A query that searched on a `ucs2` column failed if the table was partitioned. (Bug #48737)

- **Partitioning:** In some cases, it was not possible to add a new column to a table that had subpartitions. (Bug #48276)

- **Partitioning:** `SELECT COUNT(*)` from a partitioned table failed when using the `ONLY_FULL_GROUP_BY` SQL mode. (Bug #46923)

  References: This bug was introduced by Bug #45807.

- **Partitioning:** `SUBPARTITION BY KEY` failed with `DEFAULT CHARSET=utf8`. (Bug #45904)

- **Replication:** When using row-based logging, `TRUNCATE TABLE` was written to the binary log even if the affected table was temporary, causing replication to fail. (Bug #48350)

- **Replication:** When using the `STATEMENT` or `MIXED` logging format, the statements `LOAD DATA CONCURRENT LOCAL INFILE` and `LOAD DATA CONCURRENT INFILE` were logged as `LOAD DATA LOCAL INFILE` and `LOAD DATA LOCAL INFILE`, respectively (in other words, the `CONCURRENT` keyword was omitted). As a result, when using replication with either of these logging modes, queries on the slaves were blocked by the replication SQL thread while trying to execute the affected statements. (Bug #34628)

- **Replication:** A flaw in the implementation of the purging of binary logs could result in orphaned files being left behind in the following circumstances:

  - If the server failed or was killed while purging binary logs.

    If the server failed or was killed after creating of a new binary log when the new log file was opened for the first time.

  In addition, if the slave was not connected during the purge operation, it was possible for a log file that was in use to be removed; this could lead data loss and possible inconsistencies between the master and slave. (Bug #45292)

- **Cluster Replication:** When `expire_logs_days` was set, the thread performing the purge of the log files could deadlock, causing all binary log operations to stop. (Bug #49536)

- Building MySQL on Fedora Core 12 64-bit failed, due to errors in `comp_err`. (Bug #48864)
- When running `mysql_secure_installation`, the command failed if the `root` password contained multiple space, `'\'`, `'#'`, or quote characters. (Bug #48031)
- On Windows, the `mysql_secure_installation` command failed to load the `Term::ReadKey` module, which was required for correct operation. (Bug #35106)
- Use of `InnoDB` monitoring (`SHOW ENGINE INNODB STATUS` or one of the `InnoDB` Monitor tables) could cause a server crash due to invalid access to a shared variable in a concurrent environment. (Bug #38883)
- `GROUP BY` on a `constant` (single-row) `InnoDB` table joined to other tables caused a server crash. (Bug #44886)
- On Windows, `InnoDB` could not be built as a statically linked library. (Bug #48317)
- Valgrind errors for `InnoDB` were corrected. (Bug #45992, Bug #46656)
- For `YEAR(2)` values, `MIN()`, `MAX()`, and comparisons could yield incorrect results. (Bug #43668)
- Assignment of a system variable sharing the same base name as a declared stored program variable in the same context could lead to a crash. (Bug #47627)
- The server could crash when attempting to access a non-conformant `mysql.proc` system table. For example, the server could crash when invoking stored procedure-related statements after an upgrade from MySQL 5.0 to 5.1 without running `mysql_upgrade`. (Bug #41726)
- Privileges for stored routines were ignored for mixed-case routine names. (Bug #48872)

  References: See also Bug #41049.
- `DISTINCT` was ignored for queries with `GROUP BY WITH ROLLUP` and only `const` tables. (Bug #48475)
- Loose index scan was inappropriately chosen for some `WHERE` conditions. (Bug #48472)
- The `innodb_file_format_check` system variable could not be set at runtime to `DEFAULT` or to the value of a user-defined variable. (Bug #47167)
- If the `--log-bin` server option was set to a directory name with a trailing component separator character, the basename of the binary log files was empty, so that the created files were named `.000001` and `.index`. The same thing occurred with the `--log-bin-index`, `--relay-log`, and `--relay-log-index` options. Now the server reports and error and exits. (Bug #34739)
- Concurrent `ALTER TABLE` operations on an `InnoDB` table could raise an assertion. (Bug #48782)
- During query execution, ranges could be merged incorrectly for `OR` operations and return an incorrect result. (Bug #48665)
- The `InnoDB` Table Monitor reported the `FLOAT` and `DOUBLE` data types incorrectly. (Bug #48526)
- With row-based binary logging, the server crashed for statements of the form `CREATE TABLE IF NOT EXISTS` *`existing_view`* `LIKE` *`temporary_table`*. This occurred because the server handled the existing view as a table when logging the statement. (Bug #48506)
- A bad typecast could cause query execution to allocate large amounts of memory. (Bug #48458)
- `MATCH IN BOOLEAN MODE` searches could return too many results inside a subquery. (Bug #47930)

- Connecting to a 4.1.x server from a 5.1.x or higher `mysql` client resulted in a memory-free error when disconnecting. (Bug #47655)
- `mysql_secure_installation` did not work on Solaris. (Bug #48086)
- Parts of the range optimizer could be initialized incorrectly, resulting in Valgrind errors. (Bug #48459)
- The server could crash and corrupt the tablespace if the `InnoDB` tablespace was configured with too small a value, or if `innodb_file_per_table` was enabled and many `CREATE TEMPORARY TABLE` statements were executed and the temporary file directory filled up. (Bug #48469)
- The `mysql` client `status` command displayed an incorrect value for the server character set. (Bug #47671)
- The `IGNORE` clause on a `DELETE` statement masked an SQL statement error that occurred during trigger processing. (Bug #46425)
- If a session acquired a global read lock with `FLUSH TABLES WITH READ LOCK`, acquired a lock for one table with `LOCK TABLES`, and issued an `INSERT DELAYED` statement for another table, deadlock could occur. (Bug #47682)
- The return value was not checked for some `my_hash_insert()` calls. (Bug #45613)
- If a comparison involved a constant value that required type conversion, the converted value might not be cached, resulting in repeated conversion and poorer performance. (Bug #34384)
- User-defined collations with an ID less than 256 were not initialized correctly when loaded and caused a server crash. (Bug #47756)
- Comparison with `NULL` values sometimes did not produce a correct result. (Bug #42760)
- The result of comparison between nullable `BIGINT` and `INT` columns was inconsistent. (Bug #49517)
- Incorrect cache initialization prevented storage of converted constant values and could produce incorrect comparison results. (Bug #49489)
- Comparisons involving `YEAR` values could produce incorrect results. (Bug #49480)

  References: See also Bug #43668.
- If a query involving a table was terminated with `KILL`, a subsequent `SHOW CREATE TABLE` for that table caused a server crash. (Bug #48985)
- A Valgrind error in `make_cond_for_table_from_pred()` was corrected. Thanks to Sergey Petrunya for the patch to fix this bug. (Bug #49506)
- The `mysql_upgrade` command added three columns to the `mysql.proc` table (`character_set_client`, `collation_connection`, and `db_collation`), but did not populate the columns with correct values. This led to error messages reported during stored procedure execution. (Bug #41569)
- Valgrind warnings for `CHECKSUM TABLE` were corrected. (Bug #49465)
- The `mysql.server` script had incorrect shutdown logic. (Bug #49772)
- On Solaris, the server printed no stack trace to the error log after a crash. (Bug #47391)
- The error message for `ER_UPDATE_INFO` was subject to buffer overflow or truncation. (Bug #48500)

- Specifying an index algorithm (such as `BTREE`) for `SPATIAL` or `FULLTEXT` indexes caused a server crash. These index types do not support algorithm specification, and it is not longer permitted to do so. (Bug #49250)
- The optimizer sometimes incorrectly handled conditions of the form `WHERE col_name='const1' AND col_name='const2'`. (Bug #49199)
- The `LIKE` operator did not work correctly when using an index for a `ucs2` column. (Bug #49028)
- Execution of `DECODE()` and `ENCODE()` could be inefficient because multiple executions within a single statement reinitialized the random generator multiple times even with constant parameters. (Bug #49141)
- `check_key_in_view()` was missing a `DBUG_RETURN` in one code branch, causing a crash in debug builds. (Bug #48995)
- When compressed `MyISAM` files were opened, they were always memory mapped, sometimes causing memory-swapping problems. To deal with this, a new system variable, `myisam_mmap_size`, was added to permit limiting the amount of memory used for memory mapping of `MyISAM` files. (Bug #37408)
- For debug builds on Windows, `SAFEMALLOC` was defined inconsistently, leading to mismatches when using `my_malloc()` and `my_free()`. (Bug #49811)
- In debug builds, killing a `LOAD XML INFILE` statement raised an assertion.

  Implemented in the course of fixing this bug, `mysqltest` has a new `send_eval` command that combines the functionality of the existing `send` and `eval` commands. (Bug #42520)
- It was possible for `init_available_charsets()` not to initialize correctly. (Bug #45058)
- With binary logging enabled, `REVOKE ... ON {PROCEDURE|FUNCTION} FROM ...` could cause a crash. (Bug #49119)
- Re-execution of a prepared statement could cause a server crash. (Bug #48508)
- `SHOW BINLOG EVENTS` could fail with a error: `Wrong offset or I/O error`. (Bug #48357)
- Valgrind warnings related to binary logging of `LOAD DATA INFILE` statements were corrected. (Bug #48340)
- The first execution of `STOP SLAVE UNTIL` stopped too early. (Bug #47210)
- Incomplete reset of internal `TABLE` structures could cause a crash with `eq_ref` table access in subqueries. (Bug #48709)
- The `push_warning_printf()` function was being called with an invalid error level, `MYSQL_ERROR::WARN_LEVEL_ERROR`, causing an assertion failure. To fix the problem, `MYSQL_ERROR::WARN_LEVEL_ERROR` has been replaced by `MYSQL_ERROR::WARN_LEVEL_WARN`. (Bug #49638)
- Queries containing `GROUP BY ... WITH ROLLUP` that did not use indexes could return incorrect results. (Bug #47650)
- For a `VARCHAR(N)` column, `ORDER BY BINARY(col_name)` sorted using only the first `N` bytes of the column, even though column values could be longer than `N` bytes if they contained multi-byte characters. (Bug #44131)
- Under heavy load with a large query cache, invalidating part of the cache could cause the server to freeze (that is, to be unable to service other operations until the invalidation was complete). (Bug #21074)

References: See also Bug #39253.

- Output from `mysql --html` did not encode the `'<'`, `'>'`, or `'&'` characters. (Bug #27884)
- On some Windows systems, `InnoDB` could report `Operating system error number 995 in a file operation` due to transient driver or hardware problems. `InnoDB` now retries the operation and adds `Retry attempt is made` to the error message. (Bug #3139)
- Using the `SHOW ENGINE INNODB STATUS` statement when using partitions in `InnoDB` tables caused `Invalid (old?) table or database name` errors to be logged. (Bug #32430)

Changes in MySQL 5.5.0 (2009-12-07, Milestone 2)

Previously, MySQL development proceeded by including a large set of features and moving them over many versions within a release series through several stages of maturity (Alpha, Beta, and so forth). This development model had a disadvantage in that problems with only part of the code could hinder timely release of the whole. As you might have found when testing MySQL Server 6.0, alpha quality code could jeopardize the stability of the entire release. (One consequence of this was that MySQL Server 6.0 has been withdrawn.)

MySQL development now uses a milestone model. The move to this model provides for more frequent milestone releases, with each milestone proceeding through a small number of releases having a focus on a specific subset of thoroughly tested features. Following the releases for one milestone, development proceeds with the next milestone; that is, another small number of releases that focuses on the next small set of features, also thoroughly tested.

MySQL 5.5.0-m2 is the first release for Milestone 2. The new features of this milestone may be considered to be initially of beta quality. For subsequent Milestone 2 releases, we plan to use increasing version numbers (5.5.1 and higher) while continuing to employ the “-m2” suffix. For Milestone 3, we plan to change the suffix to “-m3”. Version designators with “-alpha” or “-beta” suffixes are no used.

You may notice that the MySQL 5.5.0 release is designated as Milestone 2 rather than Milestone 1. This is because MySQL 5.4 was actually designated as Milestone 1, although we had not yet begun referring to milestone numbers as part of version numbers at the time.

InnoDB Notes

- The `InnoDB Plugin` is included in MySQL 5.5 releases as the built-in version of `InnoDB`. The version of the `InnoDB` in this release is 1.0.5 and is considered of Release Candidate (RC) quality.

  This version of `InnoDB` offers new features, improved performance and scalability, enhanced reliability and new capabilities for flexibility and ease of use. Among the features are “Fast index creation,” table and index compression, file format management, new `INFORMATION_SCHEMA` tables, capacity tuning, multiple background I/O threads, and group commit.

  In this version of `InnoDB`, the `innodb_file_io_threads` system variable has been removed and replaced with `innodb_read_io_threads` and `innodb_write_io_threads`. If you upgrade from MySQL 5.1 to MySQL 5.5 and previously explicitly set `innodb_file_io_threads` at server startup, you must change your configuration. Either remove any reference to `innodb_file_io_threads` or replace it with references to `innodb_read_io_threads` and `innodb_write_io_threads`.

Functionality Added or Changed

- **Incompatible Change:** The deprecated `--default-table-type` server option has been removed (use `--default-storage-engine`). (Bug #34818)

- **Incompatible Change:** Two status variables have been added to `SHOW STATUS` output. `Innodb_buffer_pool_read_ahead` and `Innodb_buffer_pool_read_ahead_evicted` indicate the number of pages read in by the `InnoDB` read-ahead background thread, and the number of such pages evicted without ever being accessed, respectively. Also, the status variables `Innodb_buffer_pool_read_ahead_rnd` and `Innodb_buffer_pool_read_ahead_seq` status variables have been removed. (Bug #42885)

- **Incompatible Change:** MySQL Server now includes a plugin services interface that complements the plugin API. The services interface enables server functionality to be exposed as a "service" that plugins can access using function calls. The `libmysqlservices` library provides access to the available services and dynamic plugins now must be linked against this library (use the `-lmysqlservices` flag). See MySQL Services for Plugins. (Bug #48461)

- **Incompatible Change:** The `TRADITIONAL` SQL mode now includes `NO_ENGINE_SUBSTITUTION`. (Bug #21099)

- **Incompatible Change:** Several changes have been made regarding the language and character set of error messages:

  - The `language` system variable has been removed and replaced with the new `lc_messages_dir` and `lc_messages` system variables. `lc_messages_dir` has only a global value and is read only. `lc_messages` has global and session values and can be modified at runtime, so the error message language can be changed while the server is running, and individual clients each can have a different error message language by changing their session `lc_messages` value to a different locale name.

  - The `--language` option for specifying the directory for the error message file is now deprecated. The new `lc_messages_dir` and `lc_messages` system variables should be used instead, and the server treats `--language` as an alias for `lc_messages_dir`.

  - Error messages previously were constructed in a mix of character sets. This issue is resolved by constructing error messages internally within the server using UTF-8 and returning them to the client in the character set specified by the `character_set_results` system variable. The content of error messages therefore may in some cases differ from the messages returned previously.

  See Setting the Error Message Language, and Character Set for Error Messages.

  References: See also Bug #46218, Bug #46236.

- **Partitioning:** It is now possible to assign indexes on partitioned `MyISAM` tables to key caches using the `CACHE INDEX` and to preload such indexes into the cache using `LOAD INDEX INTO CACHE` statements. Cache assignment and preloading of indexes for such tables can be performed for one, several, or all partitions of the table.

  This functionality is supported for only those partitioned tables that employ the `MyISAM` storage engine.

- **Partitioning:** A new `ALTER TABLE` option, `TRUNCATE PARTITION`, makes it possible to delete rows from one or more selected partitions only. Unlike the case with `ALTER TABLE ... DROP PARTITION`, `ALTER TABLE ... TRUNCATE PARTITION` merely deletes all rows from the specified partition or partitions, and does not change the definition of the table.

- **Partitioning:** New `PARTITION BY RANGE COLUMNS(column_list)` and `PARTITION BY LIST COLUMNS(column_list)` options are added for the `CREATE TABLE` and `ALTER TABLE` statements.

  A major benefit of `RANGE COLUMNS` and `LIST COLUMNS` partitioning is that they make it possible to define ranges or lists based on column values that use string, date, or datetime values.

These new extensions also broaden the scope of partition pruning to provide better coverage for queries using comparisons on multiple columns in the `WHERE` clause, some examples being `WHERE a = 1 AND b < 10` and `WHERE a = 1 AND b = 10 AND c < 10`.

See `RANGE` Partitioning, `LIST` Partitioning, and Partition Pruning.

- **Replication:** A replication heartbeat mechanism has been added to facilitate monitoring. This provides an alternative to checking log files, making it possible to detect in real time when a slave has failed.

  Configuration of heartbeats is done using a new `MASTER_HEARTBEAT_PERIOD = interval` clause for the `CHANGE MASTER TO` statement (see `CHANGE MASTER TO` Syntax); monitoring can be done by checking the values of the status variables `Slave_heartbeat_period` and `Slave_received_heartbeats` (see Server Status Variables).

  The addition of replication heartbeats addresses a number of issues:

  - Relay logs were rotated every `slave_net_timeout` seconds even if no statements were being replicated.
  - `SHOW SLAVE STATUS` displayed an incorrect value for `Seconds_Behind_Master` following a `FLUSH LOGS` statement.
  - Replication master-slave connections used `slave_net_timeout` for connection timeouts.

  (Bug #20435, Bug #29309, Bug #30932)

- **Replication:** Because `SHOW BINLOG EVENTS` cannot be used to read events from relay log files, a new `SHOW RELAYLOG EVENTS` statement has been added for this purpose. (Bug #28777)

- **Replication:** The global server variable `sync_relay_log` is introduced for use on replication slaves. Setting this variable to a nonzero integer value *`N`* causes the slave to synchronize the relay log to disk after every *`N`* events. Setting its value to 0 permits the operating system to handle synchronization of the file. The action of this variable, when enabled, is analogous to how the `sync_binlog` variable works with regard to binary logs on a replication master.

  The global server variables `sync_master_info` and `sync_relay_log_info` are introduced for use on replication slaves to control synchronization of, respectively, the `master.info` and `relay.info` files.

  In each case, setting the variable to a nonzero integer value *`N`* causes the slave to synchronize the corresponding file to disk after every *`N`* events. Setting its value to 0 permits the operating system to handle synchronization of the file instead.

  The actions of these variables, when enabled, are analogous to how the `sync_binlog` variable works with regard to binary logs on a replication master.

  An additional system variable `relay_log_recovery` is also now available. When enabled, this variable causes a replication slave to discard relay log files obtained from the replication master following a crash.

  These variables can also be set in `my.cnf`, or by using the `--sync-relay-log`, `--sync-master-info`, `--sync-relay-log-info`, and `--relay-log-recovery` server options.

  For more information, see Replication Slave Options and Variables. (Bug #31665, Bug #35542, Bug #40337)

- **Replication:** In circular replication, it was sometimes possible for an event to propagate such that it would be reapplied on all servers. This could occur when the originating server was removed from the replication circle and so could no longer act as the terminator of its own events, as normally happens in circular replication.

  To prevent this from occurring, a new `IGNORE_SERVER_IDS` option is introduced for the `CHANGE MASTER TO` statement. This option takes a list of replication server IDs; events having a server ID which appears in this list are ignored and not applied. For more information, see `CHANGE MASTER TO` Syntax.

  In conjunction with the introduction of `IGNORE_SERVER_IDS`, `SHOW SLAVE STATUS` has two new fields. `Replicate_Ignore_Server_Ids` displays information about ignored servers. `Master_Server_Id` displays the `server_id` value from the master. (Bug #25998)

  References: See also Bug #27808.

- **Replication:** MySQL now supports an interface for semisynchronous replication: A commit performed on the master side blocks before returning to the session that performed the transaction until at least one slave acknowledges that it has received and logged the events for the transaction. Semisynchronous replication is implemented through an optional plugin component. See Semisynchronous Replication.

- Columns that provide a catalog value in `INFORMATION_SCHEMA` tables (for example, `TABLES.TABLE_CATALOG`) now have a value of `def` rather than `NULL`. (Bug #35427)

- The `have_community_features` system variable was renamed to `have_profiling`.

  Previously, to enable profiling, it was necessary to run `configure` with the `--enable-community-features` and `--enable-profiling` options. Now only `--enable-profiling` is needed. (Bug #44651)

- On Linux (and perhaps other systems), the performance of MySQL Server can be improved by using a different `malloc()` implementation, developed by Google and called `tcmalloc`. The gain is noticeable with a higher number of simultaneous users. To support use of this library, the following changes have been made:

  - The server is linked against the default `malloc()` provided by the respective platform.

  - Binary distributions for Linux include `libtcmalloc_minimal.so` (a shared library that can be linked against at runtime) in `pkglibdir` (that is, the same directory within the package where server plugins and similar object files are located). The version of `tcmalloc` included with MySQL comes from `google-perftools` 1.4.

    If you want to try `tcmalloc` but are using a binary distribution for a non-Linux platform or a source distribution, you can install Google's `tcmalloc`. Some distributions provide it in a `google-perftools` package or with a similar name, or you can download it from Google at http://code.google.com/p/google-perftools/ and compile it yourself.

  - `mysqld_safe` now supports a `--malloc-lib` option that enables administrators to specify that `mysqld` should use `tcmalloc`.

  The `--malloc-lib` option works by modifying the `LD_PRELOAD` environment value to affect dynamic linking to enable the loader to find the memory-allocation library when `mysqld` runs:

  - If the option is not given, or is given without a value (`--malloc-lib=`), `LD_PRELOAD` is not modified and no attempt is made to use `tcmalloc`.

  - If the option is given as `--malloc-lib=tcmalloc`, `mysqld_safe` looks for a `tcmalloc` library in `/usr/lib` and then in the MySQL `pkglibdir` location (for example, `/usr/local/mysql/`

lib or whatever is appropriate). If tmalloc is found, its path name is added to the beginning of the LD_PRELOAD value for mysqld. If tcmalloc is not found, mysqld_safe aborts with an error.

- If the option is given as --malloc-lib=*/path/to/some/library*, that full path is added to the beginning of the LD_PRELOAD value. If the full path points to a nonexistent or unreadable file, mysqld_safe aborts with an error.
- For cases where mysqld_safe adds a path name to LD_PRELOAD, it adds the path to the beginning of any existing value the variable already has.

As a result of the preceding changes, Linux users can use the libtcmalloc_minimal.so now included in binary packages by adding these lines to the my.cnf file:

```
[mysqld_safe]
malloc-lib=tcmalloc
```

Those lines also suffice for users on any platform who have installed a tcmalloc package in /usr/lib. To use a specific tcmalloc library, specify its full path name. Example:

```
[mysqld_safe]
malloc-lib=/opt/lib/libtcmalloc_minimal.so
```

(Bug #47549)

- The InnoDB buffer pool is divided into two sublists: A new sublist containing blocks that are heavily used by queries, and an old sublist containing less-used blocks and from which candidates for eviction are taken. In the default operation of the buffer pool, a block when read in is loaded at the midpoint and then moved immediately to the head of the new sublist as soon as an access occurs. In the case of a table scan (such as performed for a mysqldump operation), each block read by the scan ends up moving to the head of the new sublist because multiple rows are accessed from each block. This occurs even for a one-time scan, where the blocks are not otherwise used by other queries. Blocks may also be loaded by the read-ahead background thread and then moved to the head of the new sublist by a single access. These effects can be disadvantageous because they push blocks that are in heavy use by other queries out of the new sublist to the old sublist where they become subject to eviction.

  InnoDB now provides two system variables that enable LRU algorithm tuning:

  - innodb_old_blocks_pct

    Specifies the approximate percentage of the buffer pool used for the old block sublist. The range of values is 5 to 95. The default value is 37 (that is, 3/8 of the pool).

  - innodb_old_blocks_time

    Specifies how long in milliseconds (ms) a block inserted into the old sublist must stay there after its first access before it can be moved to the new sublist. The default value is 0: A block inserted into the old sublist moves immediately to the new sublist the first time it is accessed, no matter how soon after insertion the access occurs. If the value is greater than 0, blocks remain in the old sublist until an access occurs at least that many ms after the first access. For example, a value of 1000 causes blocks to stay in the old sublist for 1 second after the first access before they become eligible to move to the new sublist.

  See The InnoDB Buffer Pool. (Bug #45015)

- Previously, mysqldump would not dump the INFORMATION_SCHEMA database and ignored it if it was named on the command line. Now, mysqldump will dump INFORMATION_SCHEMA if it is named on the

command line. Currently, this requires that the `--skip-lock-tables` (or `--skip-opt`) option be given. (Bug #33762)

- 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)
- The `LOAD XML INFILE` statement was added. This statement makes it possible to read data directly from XML files into database tables. For more information, see `LOAD XML` Syntax.
- Parser performance was improved for identifier scanning and conversion of ASCII string literals.
- If the value of the `--log-warnings` option is greater than 1, the server now writes access-denied errors for new connection attempts to the error log (for example, if a client user name or password is incorrect). (Bug #25822)
- Sinhala collations `utf8_sinhala_ci` and `ucs2_sinhala_ci` were added for the `utf8` and `ucs2` character sets. (Bug #26474)
- Added the `TO_SECONDS()` function, which converts a date or datetime value to a number of seconds since the year 0. This is a general-purpose function, but is useful for partitioning. You may use this function in partitioning expressions, and partition pruning is supported for tables defined using such expressions.
- Several undocumented C API functions were removed: `mysql_manager_close()`, `mysql_manager_command()`, `mysql_manager_connect()`, `mysql_manager_fetch_line()`, `mysql_manager_init()`, `mysql_disable_reads_from_master()`, `mysql_disable_rpl_parse()`, `mysql_enable_reads_from_master()`, `mysql_enable_rpl_parse()`, `mysql_master_query()`, `mysql_master_send_query()`, `mysql_reads_from_master_enabled()`, `mysql_rpl_parse_enabled()`, `mysql_rpl_probe()`, `mysql_rpl_query_type()`, `mysql_set_master()`, `mysql_slave_query()`, and `mysql_slave_send_query()`. (Bug #31952, Bug #31954)
- The undocumented, deprecated, and not useful `SHOW COLUMN TYPES` statement has been removed. (Bug #5299)
- The server now supports a Debug Sync facility for thread synchronization during testing and debugging. To compile in this facility, configure MySQL with the `--enable-debug-sync` option. The `debug_sync` system variable provides the user interface Debug Sync. `mysqld` and `mysql-test-run.pl` support a `--debug-sync-timeout` option to enable the facility and set the default synchronization point timeout.
- The `CREATE TABLESPACE` privilege has been introduced. This privilege exists at the global (superuser) level and enables you to create, alter, and drop tablespaces and logfile groups.
- On Windows, use of POSIX I/O interfaces in `mysys` was replaced with Win32 API calls (`CreateFile()`, `WriteFile()`, and so forth) and the default maximum number of open files has been increased to 16384. The maximum can be increased further by using the `--open-files-limit=N` option at server startup. (Bug #24509)
- MySQL now implements the SQL standard `SIGNAL` and `RESIGNAL` statements. See Condition Handling. (Bug #11661)
- `mysql_upgrade` now has an `--upgrade-system-tables` option that causes only the system tables to be upgraded. With this option, data upgrades are not performed.
- Previously, in the absence of other information, the MySQL client programs `mysql`, `mysqladmin`, `mysqlcheck`, `mysqlimport`, and `mysqlshow` used the compiled-in default character set, usually `latin1`.

Now these clients can autodetect which character set to use based on the operating system setting, such as the value of the `LANG` or `LC_ALL` locale environment language on Unix system or the code page setting on Windows systems. For systems on which the locale is available from the OS, the client uses it to set the default character set rather than using the compiled-in default. Thus, users can configure the locale in their environment for use by MySQL clients. For example, setting `LANG` to `ru_RU.KOI8-R` causes the `koi8r` character set to be used. The OS character set is mapped to the closest MySQL character set if there is no exact match. If the client does not support the matching character set, it uses the compiled-in default. (For example, `ucs2` is not supported as a connection character set.)

An implication of this change is that if your environment is configured to use a non-`latin1` locale, MySQL client programs will use a different connection character set than previously, as though you had issued an implicit `SET NAMES` statement. If the previous behavior is required, start the client with the `--default-character-set=latin1` option.

Third-party applications that wish to use character set autodetection based on the OS setting can use the following `mysql_options()` call before connecting to the server:

```
mysql_options(mysql,
              MYSQL_SET_CHARSET_NAME,
              MYSQL_AUTODETECT_CHARSET_NAME);
```

See Connection Character Sets and Collations.

- The Greek locale `'el_GR'` is now a permissible value for the `lc_time_names` system variable.
- The `FORMAT()` function now supports an optional third parameter that enables a locale to be specified to be used for the result number's decimal point, thousands separator, and grouping between separators. Permissible locale values are the same as the legal values for the `lc_time_names` system variable (see MySQL Server Locale Support). For example, the result from `FORMAT(1234567.89,2,'de_DE')` is `1.234.567,89`. If no locale is specified, the default is `'en_US'`.
- The `libmysqlclient` client library is now built as a thread-safe library. The `libmysqlclient_r` client library is still present for compatibility, but is just a symlink to `libmysqlclient`.

Bugs Fixed

- **Security Fix; Important Change:** It was possible to circumvent privileges through the creation of `MyISAM` tables employing the `DATA DIRECTORY` and `INDEX DIRECTORY` options to overwrite existing table files in the MySQL data directory. Use of the MySQL data directory in `DATA DIRECTORY` and `INDEX DIRECTORY` is no longer permitted. This is now also true of these options when used with partitioned tables and individual partitions of such tables. (Bug #32167, CVE-2008-2079)

  References: See also Bug #39277.

- **Security Fix:** The server crashed if an account without the proper privileges attempted to create a stored procedure. (Bug #44658)
- **Security Fix:** MySQL clients linked against OpenSSL could be tricked not to check server certificates. (Bug #47320, CVE-2009-4028)
- **Performance:** The server unnecessarily acquired a query cache mutex even with the query cache disabled, resulting in a small performance decrement which could show up as threads often in the “invalidating query cache entries (table)” state, particularly on a replication slave with row-based replication. Now if the server is started with `query_cache_type` set to 0, it does not acquire the query cache mutex. This has the implication that the query cache cannot be enabled at runtime. (Bug #38551)

- **Performance:** The `InnoDB` adaptive hash latch is released (if held) for several potentially long-running operations. This improves throughput for other queries if the current query is removing a temporary table, changing a temporary table from memory to disk, using `CREATE TABLE ... SELECT`, or performing a `MyISAM` repair on a table used within a transaction. (Bug #32149)

- **Incompatible Change; Replication:** Concurrent transactions that inserted rows into a table with an `AUTO_INCREMENT` column could break statement-based or mixed-format replication error 1062 `Duplicate entry '...' for key 'PRIMARY'` on the slave. This was especially likely to happen when one of the transactions activated a trigger that inserted rows into the table with the `AUTO_INCREMENT` column, although other conditions could also cause the issue to manifest.

  As part of the fix for this issue, any statement that causes a trigger or function to update an `AUTO_INCREMENT` column is now considered unsafe for statement-based replication. For more information, see Replication and `AUTO_INCREMENT`. (Bug #45677)

  References: See also Bug #42415, Bug #48608, Bug #50440, Bug #53079.

- **Incompatible Change:** For system variables that take values of `ON` or `OFF`, `OF` was accepted as a legal variable. Now system variables that take “enumeration” values must be assigned the full value. This affects some other variables that previously could be assigned using unambiguous prefixes of permissible values, such as `tx_isolation`. (Bug #34828)

- **Incompatible Change:** In binary installations of MySQL, the supplied `binary-configure` script would start and configure MySQL, even when command help was requested with the `--help` command-line option. The `--help` option, if provided, no longer starts and installs the server. (Bug #30954)

- **Incompatible Change:** Access privileges for several statements are more accurately checked:

  - `CHECK TABLE` requires some privilege for the table.
  - `CHECKSUM TABLE` requires `SELECT` for the table.
  - `CREATE TABLE ... LIKE` requires `SELECT` for the source table and `CREATE` for the destination table.
  - `SHOW COLUMNS` displays information only for those columns for which you have some privilege.
  - `SHOW CREATE TABLE` requires some privilege for the table (previously required `SELECT`).
  - `SHOW CREATE VIEW` requires `SHOW VIEW` and `SELECT` for the view.
  - `SHOW INDEX` requires some privilege for any column.
  - `SHOW OPEN TABLES` displays only tables for which you have some privilege on any column.

  (Bug #27145)

- **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)

- **Important Change; Replication:** `MyISAM` transactions replicated to a transactional slave left the slave in an unstable condition. This was due to the fact that, when replicating from a nontransactional storage engine to a transactional engine with `autocommit` disabled, no `BEGIN` and `COMMIT` statements were written to the binary log; thus, on the slave, a never-ending transaction was started.

The fix for this issue includes enforcing `autocommit` mode on the slave by replicating all `autocommit=1` statements from the master. (Bug #29288)

- **Important Change:** An option that requires a value, when specified in an option file without a value, was assigned the text of the next line in the file as the value. Now, if you fail to specify a required value in an option file, the server aborts with an error.

  This change does not effect how options are handled by the server when they are used on the command line. For example, starting the server using `mysqld_safe --relay-log --relay-log-index &` causes the server to create relay log files named `--relay-log-index.000001`, `--relay-log-index.000002`, and so on, because the `--relay-log` option expects an argument. (Bug #25192)

- **Partitioning:** An `ALTER TABLE ... ADD PARTITION` statement that caused `open_files_limit` to be exceeded led to a MySQL server crash. (Bug #46922)

  References: See also Bug #47343.

- **Partitioning:** When performing an `INSERT ... SELECT` into a partitioned table, `read_buffer_size` bytes of memory were allocated for every partition in the target table, resulting in consumption of large amounts of memory when the table had many partitions (more than 100).

  This fix changes the method used to estimate the buffer size required for each partition and limits the total buffer size to a maximum of approximately 10 times `read_buffer_size`. (Bug #45840)

- **Partitioning:** The cardinality of indexes on partitioned tables was calculated using the first partition in the table, which could result in suboptimal query execution plans being chosen. Now the partition having the most records is used instead, which should result in better use of indexes and thus improved performance of queries against partitioned tables in many if not most cases. (Bug #44059)

- **Partitioning:** For partitioned tables with more than ten partitions, a full table scan was used in some cases when only a subset of the partitions were needed. (Bug #33730)

- **Partitioning:** Truncating a partitioned `MyISAM` table did not reset the `AUTO_INCREMENT` value. (Bug #35111)

- **Replication:** When using statement-based replication and the transaction isolation level was set to `READ COMMITTED` or a less strict level, `InnoDB` returned an error even if the statement in question was filtered out according to the `--binlog-do-db` or `--binlog-ignore-db` rules in effect at the time. (Bug #42829)

- **Replication:** When using the row-based format to replicate a transaction involving both transactional and nontransactional engines, which contained a DML statement affecting multiple rows, the statement failed. If this transaction was followed by a `COMMIT`, the master and the slave could diverge, because the statement was correctly rolled back on the master, but was applied on the slave. (Bug #47287)

  References: See also Bug #46864.

- **Replication:** A problem with the `BINLOG` statement in the output of `mysqlbinlog` could break replication; statements could be logged with the server ID stored within events by the `BINLOG` statement rather than the ID of the running server. With this fix, the server ID of the server executing the statements can no longer be overridden by the server ID stored in the binary log's `format description` statement. (Bug #46640)

  References: This bug was introduced by Bug #32407.

- **Replication:** When a session was closed on the master, temporary tables belonging to that session were logged with the wrong database names when either of the following conditions was true:

1. The length of the name of the database to which the temporary table belonged was greater than the length of the current database name.

2. The current database was not set.

(Bug #48216)

References: See also Bug #46861, Bug #48297.

- **Replication:** `FLUSH LOGS` did not close and reopen the binary log index file. (Bug #34582)

  References: See also Bug #48738.

- **Replication:** When using statement-based or mixed-format replication, the database name was not written to the binary log when executing a `LOAD DATA INFILE` statement. This caused problems when the table being loaded belonged to a database other than the current database; data could be loaded into the wrong table (if a table having the same name existed in the current database) or replication could fail (if no table having that name existed in the current database). Now a table referenced in a `LOAD DATA INFILE` statement is always logged using its fully qualified name when the database to which it belongs is not the current database. (Bug #48297)

- **Replication:** When `mysqlbinlog --verbose` was used to read a binary log that had been written using row-based format, the output for events that updated some but not all columns of tables was not correct. (Bug #47323)

- **Replication:** When using row-based replication, `DROP TEMPORARY TABLE IF EXISTS` was written to the binary log if the table named in the statement did not exist, even though a `DROP TEMPORARY TABLE` statement should never be logged in row-based logging mode, whether the table exists or not. (Bug #46572)

- **Replication:** When using row-based replication, changes to nontransactional tables that occurred early in a transaction were not immediately flushed upon committing a statement. This behavior could break consistency since changes made to nontransactional tables become immediately visible to other connections. (Bug #47678)

  References: See also Bug #47287, Bug #46864, Bug #43929, Bug #11752675, Bug #46129. This bug is a regression of Bug #40116.

- **Replication:** Performing `ALTER TABLE ... DISABLE KEYS` on a slave table caused row-based replication to fail. (Bug #47312)

- **Replication:** `BEGIN` statements were not included in the output of `mysqlbinlog`. (Bug #46998)

- **Replication:** By default, all statements executed by the `mysql_upgrade` program on the master are written to the binary log, then replicated to the slave. In some cases, this can result in problems; for example, it attempted to alter log tables on replicated databases (this failed due to logging being enabled).

  As part of this fix, a `mysql_upgrade` option, `--write-binlog`, is added. Its inverse, `--skip-write-binlog`, can be used to disable binary logging while the upgrade is in progress. (Bug #43579)

- **Replication:** On the master, if a binary log event is larger than `max_allowed_packet`, the error message `ER_MASTER_FATAL_ERROR_READING_BINLOG` is sent to a slave when it requests a dump from the master, thus leading the I/O thread to stop. On a slave, the I/O thread stops when receiving a packet larger than `max_allowed_packet`.

In both cases, however, there was no `Last_IO_Error` reported, which made it difficult to determine why the slave had stopped in such cases. Now, `Last_IO_Error` is reported when `max_allowed_packet` is exceeded, and provides the reason for which the slave I/O thread stopped. (Bug #42914)

References: See also Bug #14068, Bug #47200, Bug #47303.

- **Replication:** The internal function `get_master_version_and_clock()` (defined in `sql/slave.cc`) ignored errors and passed directly when queries failed, or when queries succeeded but the result retrieved was empty. Now this function tries to reconnect the master if a query fails due to transient network problems, and to fail otherwise. The I/O thread now prints a warning if the same system variables do not exist on master (in the event the master is a very old version of MySQL, compared to the slave.) (Bug #45214)

- **Replication:** When the logging format was set without binary logging being enabled, the server failed to start. Now in such cases, the server starts successfully, `binlog_format` is set, and a warning is logged instead of an error. (Bug #42928)

- **Replication:** An error message relating to permissions required for `SHOW SLAVE STATUS` was confusing. (Bug #34227)

- **Replication:** When using row-based format, replication failed with the error `Could not execute Write_rows event on table ...; Field '...' doesn't have a default value` when an `INSERT` was made on the master without specifying a value for a column having no default, even if strict server SQL mode was not in use and the statement would otherwise have succeeded on the master. Now the SQL mode is checked, and the statement is replicated unless strict mode is in effect. For more information, see Server SQL Modes. (Bug #38173)

  References: See also Bug #38262, Bug #43992.

- **Replication:** Replicating `TEXT` or `VARCHAR` columns declared as `NULL` on the master but `NOT NULL` on the slave caused the slave to crash. (Bug #43789)

  References: See also Bug #38850, Bug #43783, Bug #43785, Bug #47741, Bug #48091.

- **Replication:** Two issues encountered on replication slaves during startup were fixed:

  1. A failure while allocating the master info structure caused the slave to crash.

  2. A failure during recovery caused the relay log file not to be properly initialized which led to a crash on the slave.

  (Bug #43075)

- **Replication:** When `autocommit` was set equal to `1` after starting a transaction, the binary log did not commit the outstanding transaction. This happened because the binary log commit function saw only the values of the new settings, and decided that there was nothing to commit.

  This issue was first observed when using the `Falcon` storage engine, but it is possible that it affected other storage engines as well. (Bug #37221)

- **Replication:** Valgrind revealed an issue with `mysqld` that was corrected: memory corruption in replication slaves when switching databases. (Bug #19022)

- **Replication:** The value of `Slave_IO_running` in the output of `SHOW SLAVE STATUS` did not distinguish between all 3 possible states of the slave I/O thread (not running; running but not connected;

connected). Now the value `Connecting` (rather than `No`) is shown when the slave I/O thread is running but the slave is not connected to a replication master.

The server system variable `Slave_running` also reflects this change, and is now consistent with what is shown for `Slave_IO_running`. (Bug #30703, Bug #41613, Bug #51089)

- **Replication:** Queries written to the slow query log on the master were not written to the slow query log on the slave. (Bug #23300)

  References: See also Bug #48632.

- **Replication:** The `--base64-output` option for `mysqlbinlog` was not honored for all types of events. This interfered in some cases with performing point-in-time recovery. (Bug #32407)

  References: See also Bug #46640, Bug #34777.

- **API:** The fix for Bug #24507 could lead in some cases to client application failures due to a race condition. Now the server waits for the “dummy” thread to return before exiting, thus making sure that only one thread can initialize the POSIX threads library. (Bug #42850)

- If `InnoDB` reached its limit on the number of concurrent transactions (1023), it wrote a descriptive message to the error log but returned a misleading error message to the client, or an assertion failure occurred. (Bug #46672)

  References: See also Bug #18828.

- An infinite hang and 100% CPU usage occurred after a handler tried to open a merge table.

  If the command `mysqladmin shutdown` was executed during the hang, the debug server generated the following assert:

  ```
  mysqld: table.cc:407: void free_table_share(TABLE_SHARE*): Assertion `share->ref_count ==
  0' failed.
  090610 14:54:04 - mysqld got signal 6 ;
  ```

  (Bug #45781)

- When building a pluggable storage engine on Windows, the engine name could be based on the directory name where the engine was located, rather than the configured storage engine name. (Bug #47795)

- The `mysys/mf_strip.c` file, which defines the `strip_sp()` function, has been removed from the MySQL source. The function was no longer used within the main build, and the supplied function was causing symbol errors on Windows builds. (Bug #47857)

- When building storage engines on Windows it was not possible to specify additional libraries within the `CMake` file required for the build. An `${engine}_LIBS` macro has been included in the files to support these additional storage-engine specific libraries. (Bug #47797)

- The `mysql-stress-test.pl` test script was missing from the `noinstall` packages on Windows. (Bug #41546)

- When creating a new instance on Windows using `mysqld-nt` and the `--install` parameter, the value of the service would be set incorrectly, resulting in a failure to start the configured service. (Bug #46917)

- If the server was started with an option that had a missing or invalid value, a subsequent error that normally would cause the server to shut down could cause it to crash instead. (Bug #42244)

- There was a race condition between the event scheduler and the server shutdown thread. (Bug #32771)
- Concurrent execution of `FLUSH TABLES` along with `SHOW FUNCTION STATUS` or `SHOW PROCEDURE STATUS` could cause a server crash. (Bug #34895)
- Different invocations of `CHECKSUM TABLE` could return different results for a table containing columns with spatial data types. (Bug #35570)
- The `SHOW FUNCTION CODE` and `SHOW PROCEDURE CODE` statements are not present in nondebug builds, but attempting to use them resulted in a “syntax error” message. Now the error message indicates that the statements are disabled and that you must use a debug build. (Bug #33637)
- Some warnings were being reported as errors. (Bug #36777)
- `mysql_real_connect()` did not check whether the `MYSQL` connection handler was already connected and connected again even if so. Now a `CR_ALREADY_CONNECTED` error occurs. (Bug #33831)
- There were spurious warnings about `"Truncated incorrect DOUBLE value"` in queries with `MATCH ... AGAINST` and `>` or `<` with a constant (which was reported as an incorrect `DOUBLE` value) in the `WHERE` condition. (Bug #34374)
- A Valgrind warning in `open_tables()` was corrected. (Bug #41759)
- `INFORMATION_SCHEMA` access optimizations did not work properly in some cases. (Bug #39270)
- Result set metadata for columns retrieved from `INFORMATION_SCHEMA` tables did not have the `db` or `org_table` members of the `MYSQL_FIELD` structure set. (Bug #35428)
- When a storage engine plugin failed to initialize before allocating a slot number, it would acidentally unplug the engine installed in slot 0. (Bug #41013)
- The `mysql_stmt_close()` C API function did not flush all pending data associated with the prepared statement. (Bug #39519)
- A workload consisting of `CREATE TABLE ... SELECT` and DML operations could cause deadlock. (Bug #37433)
- Optimized builds of `mysqld` crashed when built with Sun Studio on SPARC platforms. (Bug #40244)
- `SHOW CREATE EVENT` output did not include the `DEFINER` clause. (Bug #35297)
- The `ALTER ROUTINE` privilege incorrectly permitted `SHOW CREATE TABLE`. (Bug #38347)
- `INSTALL PLUGIN` and `UNINSTALL PLUGIN` did not handle plugin identifiers consistently with respect to lettercase. (Bug #33731)
- `mysqlbinlog` option-processing code had a memory leak. (Bug #38468)
- For settings of `lower_case_table_names` greater than 0, some queries for `INFORMATION_SCHEMA` tables left entries with incorrect lettercase in the table definition cache. (Bug #44738)
- Setting the `general_log_file` or `slow_query_log_file` system variable to a nonconstant expression caused the variable to become unset. (Bug #38124)
- `DROP DATABASE` did not clear the message list. (Bug #43012, Bug #43138)
- Previously, `InnoDB` performed `REPLACE INTO T SELECT ... FROM S WHERE ...` by setting shared next-key locks on rows from `S`. Now `InnoDB` selects rows from `S` with shared locks or as a consistent read, as for `INSERT ... SELECT`. This reduces lock contention between sessions. (Bug #37232)

- `mysqld_safe` could fail to find the `logger` program. (Bug #44736)
- The default values for the general query log and slow query log file are documented to be based on the server host name and located in the data directory. However, they were in fact being based on the basename and location of the process ID (PID) file. The name and location defaults for the PID file are based on the server host name and data directory, so if it was not assigned a different name explicitly, its defaults were used and the general query log and slow query log file defaults were as documented. But if the PID file was assigned a value with the `--pid-file` option, the defaults for the general query log and slow query log file were incorrect. This has been rectified so that the defaults for all three files are based on the server host name and data directory.

  A remaining problem is that the binary log and relay log *`.NNNNNN`* and `.index` basename defaults are based on the PID file basename, contrary to the documentation. This issue is to be addressed as Bug #45359. (Bug #33693)
- A `COMMENT` longer than 64 characters caused `CREATE PROCEDURE` to fail. (Bug #34197)
- The `LAST_DAY()` and `MAKEDATE()` functions could return `NULL`, but the result metadata indicated `NOT NULL`. Thanks to Hiromichi Watari for the patch to fix this bug. (Bug #33629)
- The combination of `MIN()` or `MAX()` in the select list with `WHERE` and `GROUP BY` clauses could lead to incorrect results. (Bug #45386)
- Redefining a trigger could cause an assertion failure. (Bug #43054)
- `ALTER TABLE` neglected to preserve `ROW_FORMAT` information from the original table, which could cause subsequent `ALTER TABLE` and `OPTIMIZE TABLE` statements to lose the row format for `InnoDB` tables. (Bug #39200)
- Instance Manager (`mysqlmanager`) has been removed, but a reference to it still appeared in the `mysql.server` script. (Bug #33472)
- `CREATE TABLE` failed if a column name in a `FOREIGN KEY` clause was given in a lettercase different from the corresponding index definition. (Bug #39932)
- After an error such as a table-full condition, `INSERT IGNORE` could cause an assertion failure for debug builds. (Bug #46075)
- For its warning count, the `mysql_info()` C API function could print the number of truncated data items rather than the number of warnings. (Bug #34898)
- For debug builds, executing a stored procedure as a prepared statement could sometimes cause an assertion failure. (Bug #44521)
- The server crashed if a shutdown occurred while a connection was idle. This happened because of a `NULL` pointer dereference while logging to the error log. (Bug #46267)
- The test suite was missing from RPM packages. (Bug #46834)
- `CREATE TABLE ... SELECT` could cause a server crash if no default database was selected. (Bug #45998)
- Using `mysql_stmt_execute()` to call a stored procedure could cause a server crash. (Bug #44495)
- Truncation of `DECIMAL` values could lead to assertion failures; for example, when deducing the type of a table column from a literal `DECIMAL` value. (Bug #45261)

  References: See also Bug #48370.

- Some Perl scripts in AIX packages contained an incorrect path to the `perl` executable. (Bug #44643)
- `SHOW ERRORS` returned an empty result set after an attempt to drop a nonexistent table. (Bug #42364)
- `SHOW CREATE TRIGGER` for a `MERGE` table trigger caused an assertion failure. (Bug #46614)
- A query containing a subquery in the `FROM` clause and `PROCEDURE ANALYSE()` caused a server crash. (Bug #46184)

  References: See also Bug #48293.
- A query containing a view using temporary tables and multiple tables in the `FROM` clause and `PROCEDURE ANALYSE()` caused a server crash.

  As a result of this bug fix, `PROCEDURE ANALYSE()` is legal only in a top-level `SELECT`. (Bug #48293)

  References: See also Bug #46184.
- `TRUNCATE TABLE` for a table that was opened with `HANDLER` did not close the handler and left it in an inconsistent state that could lead to a server crash. Now `TRUNCATE TABLE` for a table closes all open handlers for the table. (Bug #46456)
- The server could crash for queries with the following elements: 1. An “impossible where” in the outermost `SELECT`; 2. An aggregate in the outermost `SELECT`; 3. A correlated subquery with a `WHERE` clause that includes an outer field reference as a top-level `WHERE` sargable predicate; (Bug #46749)
- Concurrent `INSERT INTO ... SELECT` statements for an `InnoDB` table could cause an `AUTO_INCREMENT` assertion failure. (Bug #46650)
- The `mysql_config` script contained a reference to `@innodb_system_libs@` that was not replaced with the corresponding library flags during the build process and ended up in the output of `mysql_config --libs`. (Bug #47007)
- For `InnoDB` tables, an unnecessary table rebuild for `ALTER TABLE` could sometimes occur for metadata-only changes. (Bug #46760)
- Attempts to enable `large_pages` with a shared memory segment larger than 4GB caused a server crash. (Bug #43606)
- `DIV` operations that are out of range generated an error `Error (Code 1264): Out of range value` (correct), but also an error: `Error (Code 1041): Out of memory` (incorrect). (Bug #46606)
- An assertion could be raised for `CREATE TABLE` if there was a pending `INSERT DELAYED` or `REPLACE DELAYED` for the same table. (Bug #47274)
- Queries of the form `SELECT SUM(DISTINCT varchar_key) FROM tbl_name` caused a server crash. (Bug #47421)
- Selecting from the process list in the embedded server caused a crash. (Bug #43733)

  References: See also Bug #47304.
- Client flags were incorrectly initialized for the embedded server, causing several tests in the `jp` test suite to fail. (Bug #45159)
- Failed multiple-table `DELETE` statements could raise an assertion. (Bug #46958)
- The `socket` system variable was unavailable on Windows. (Bug #45498)

- Privileges for `SHOW CREATE VIEW` were not being checked correctly. (Bug #35996)
- `LOAD DATA INFILE` statements were written to the binary log in such a way that parsing problems could occur when re-executing the statement from the log. (Bug #43746)
- `SELECT ... WHERE ... IN (NULL, ...)` was executed using a full table scan, even if the same query without the `NULL` used an efficient range scan. (Bug #44139)

  References: See also Bug #18360.
- `EXPLAIN` caused a server crash for certain valid queries. (Bug #47106)
- The `configure` option `--without-server` did not work. (Bug #46980)
- Solaris binary packages now are compiled with `-g0` rather than `-g`. (Bug #47137)
- The `pthread_cond_wait()` implementations for Windows could deadlock in some rare circumstances. (Bug #47768)
- The MySQL client library mishandled `EINPROGRESS` errors for connections in nonblocking mode. This could lead to replication failures on hosts capable of resolving both IPv4 and IPv6 network addresses, when trying to resolve `localhost`. (Bug #37267)

  References: See also Bug #44344.
- Appending values to an `ENUM` or `SET` definition is a metadata change for which `ALTER TABLE` need not rebuild the table, but it was being rebuilt anyway. (Bug #45567)
- Debug builds could not be compiled with the Sun Studio compiler. (Bug #47474)
- On Mac OS X or Windows, sending a `SIGHUP` signal to the server or an asynchronous flush (triggered by `flush_time`) caused the server to crash. (Bug #47525)
- A multiple-table `UPDATE` involving a natural join and a mergeable view raised an assertion. (Bug #47150)
- `InnoDB` did not disallow creation of an index with the name `GEN_CLUST_INDEX`, which is used internally. (Bug #46000)
- Dropping an `InnoDB` table that used an unknown collation (created on a different server, for example) caused a server crash. (Bug #46256)
- `InnoDB` did not always disallow creating tables containing columns with names that match the names of internal columns, such as `DB_ROW_ID`, `DB_TRX_ID`, `DB_ROLL_PTR`, and `DB_MIX_ID`. (Bug #44369)
- The server used the wrong lock type (always `TL_READ` instead of `TL_READ_NO_INSERT` when appropriate) for tables used in subqueries of `UPDATE` statements. This led in some cases to replication failure because statements were written in the wrong order to the binary log. (Bug #42108)
- `InnoDB` use of `SELECT MAX(autoinc_column)` could cause a crash when MySQL data dictionaries went out of sync. (Bug #44030)
- The `NUM_FLAG` bit of the `MYSQL_FIELD.flags` member now is set for columns of type `MYSQL_TYPE_NEWDECIMAL`. (Bug #42980)
- Incorrect handling of predicates involving `NULL` by the range optimizer could lead to an infinite loop during query execution. (Bug #47123)
- A simple `SELECT` with implicit grouping could return many rows rather than a single row if the query was ordered by the aggregated column in the select list. (Bug #47280)

- Failure to treat `BIT` values as unsigned could lead to unpredictable results. (Bug #42803)
- Incorrect handling of range predicates combined with `OR` operators could yield incorrect results. (Bug #42846)
- The `FORCE INDEX FOR ORDER BY` index hint was ignored when join buffering was used. (Bug #43029)
- During the build of the Red Hat IA64 MySQL server RPM, the system library link order was incorrect. This made the resulting Red Hat IA64 RPM depend on "libc.so.6.1(GLIBC_PRIVATE)(64bit)", thus preventing installation of the package. (Bug #45706)
- A build configured using the `--without-server` option did not compile the yaSSL code, so if `--with-ssl` was also used, the build failed. (Bug #47957)
- `mysys/mf_keycache.c` requires threading, but no test was made for thread support. (Bug #47923)
- With `InnoDB`, renaming a table column and then creating an index on the renamed column caused a server crash to the `.frm` file and the `InnoDB` data directory going out of sync. Now `InnoDB` 1.0.5 returns an error instead: `ERROR 1034 (HY000): Incorrect key file for table 'tbl_name'; try to repair it`. To work around the problem, create another table with the same structure and copy the original table to it. (Bug #44571)
- The GPL and commercial license headers had different sizes, so that error log, backtrace, core dump, and cluster trace file line numbers could be off by one if they were not checked against the version of the source used for the build. (For example, checking a GPL build backtrace against commercial sources.) (Bug #46216)
- An `InnoDB` error message incorrectly referred to the nonexistent `innodb_max_files_open` variable rather than to `innodb_open_files`. (Bug #44338)
- Repair by sort or parallel repair of `MyISAM` tables might not fail over to repair with key cache. (Bug #47073)
- Trailing spaces were not ignored for user-defined collations that mapped spaces to a character other than 0x20. (Bug #46448)

  References: See also Bug #29468.
- Server shutdown failed on Windows. (Bug #48047)
- Executing `SHOW MASTER LOGS` as a prepared statement without binary logging enabled caused a crash for debug builds. (Bug #34741)
- Concurrent execution of statements requiring a table-level lock and statements requiring a non-table-level write lock for a table could deadlock. (Bug #45143)
- For certain `SELECT` statements using `ref` access, MySQL estimated an incorrect number of rows, which could lead to inefficient query plans. (Bug #38049)
- On Windows, when an idle named pipe connection was forcibly closed with a `KILL` statement or because the server was being shut down, the thread that was closing the connection would hang infinitely.

  As a result of the work done for this bug, the `net_read_timeout`, `net_write_timeout`, and `wait_timeout`, system variables now apply to connections over all transports, not just to TCP/IP. (Bug #47571, Bug #31621)

- A function call could end without throwing an error or setting the return value. For example, this could happen when an error occurred while calculating the return value. This is fixed by setting the value to `NULL` when an error occurs during evaluation of an expression. (Bug #47412)
- If a transaction was rolled back inside `InnoDB` due to a deadlock or lock wait timeout, and a statement in the transaction had an `IGNORE` clause, the server could crash at the end of the statement or on shutdown. (Bug #46539)
- During cleanup of a stored procedure's internal structures, the flag to ignore the errors for `INSERT IGNORE` or `UPDATE IGNORE` was not cleaned up, which could result in a server crash. (Bug #47788)
- Error handling was missing for `SELECT` statements containing subqueries in the `WHERE` clause and that assigned a `SELECT` result to a user variable. The server could crash as a result. (Bug #48291)
- `InnoDB` could crash when updating spatial values. (Bug #47777)
- If the first argument to `GeomFromWKB()` function was a geometry value, the function just returned its value. However, it failed to preserve the argument's `null_value` flag, which caused an unexpected `NULL` value to be returned to the caller, resulting in a server crash. (Bug #47780)
- An assertion could fail if the optimizer used a `SPATIAL` index. (Bug #48258, Bug #47019)
- For debug builds, an assertion could fail during the next statement executed for a temporary table after a multiple-table `UPDATE` involving that table modified an `AUTO_INCREMENT` column with a user-supplied value. (Bug #47919)
- The Serbian locale name `'sr_YU'` is obsolete. It is still recognized for backward compatibility, but `'sr_RS'` now should be used instead. (Bug #46633)
- The weekday names for the Romanian `lc_time_names` locale `'ro_RO'` were incorrect. Thanks to Andrei Boros for the patch to fix this bug. (Bug #43207)
- The `caseinfo` member of the `CHARSET_INFO` structure was not initialized for user-defined Unicode collations, leading to a server crash. (Bug #45645)
- On Solaris and HP-UX systems with the environment set to the default `C` locale, MySQL client programs issued an `Unknown OS character set` error. (Bug #46619)
- For `ALTER TABLE`, renaming a `DATETIME` or `TIMESTAMP` column unnecessarily caused a table copy operation. (Bug #43508)
- For the embedded server on Windows, `InnoDB` crashed when `innodb_file_per_table` was enabled and a table name was in full path format. (Bug #42383)
- On Windows, when a failed I/O operation occurred with return code of `ERROR_WORKING_SET_QUOTA`, `InnoDB` intentionally crashed the server. Now `InnoDB` sleeps for 100ms and retries the failed operation. (Bug #47055)
- Following a literal, the `COLLATE` clause was mishandled such that different results could be produced depending on whether an index was used. (Bug #48447)
- A combination of `GROUP BY WITH ROLLUP`, `DISTINCT` and the `const` join type in a query caused a server crash when the optimizer used a temporary table to resolve `DISTINCT`. (Bug #48131)
- Simultaneous `ANALYZE TABLE` operations for an `InnoDB` tables could be subject to a race condition. (Bug #38996)
- With a nonstandard `InnoDB` page size, some error messages became inaccurate.

Note

Changing the page size is not a supported operation and there is no guarantee that `InnoDB` will function normally with a page size other than 16KB. Problems compiling or running InnoDB may occur. In particular, `ROW_FORMAT=COMPRESSED` in `InnoDB` assumes that the page size is at most 16KB and uses 14-bit pointers.

A version of `InnoDB` built for one page size cannot use data files or log files from a version built for a different page size.

(Bug #41490)

- In a replication scenario with `innodb_locks_unsafe_for_binlog` enabled on the slave, where rows were changed only on the slave (not through replication), in some rare cases, many messages of the following form were written to the slave error log: `InnoDB: Error: unlock row could not find a 4 mode lock on the record`. (Bug #41756)
- Searches using a nondefault collation could return different results for a table depending on whether partitioning was used. (Bug #48161)
- The encoding of values for `SET system_variable = identifier` statements was incorrect, resulting in incorrect error messages. (Bug #47597)
- When MySQL crashed (or a snapshot was taken that simulates a crash), it was possible that internal XA transactions (used to synchronize the binary log and `InnoDB`) could be left in a `PREPARED` state, whereas they should be rolled back. This occurred when the `server_id` value changed before the restart, because that value was used to construct XID values.

  Now the restriction is relaxed that the `server_id` value be consistent for XID values to be considered valid. The rollback phase should then be able to clean up all pending XA transactions. (Bug #46944)
- On FreeBSD, memory mapping for `MERGE` tables could fail if underlying tables were empty. (Bug #47139)
- In some cases, using a null microsecond part in a `WHERE` condition (for example, `WHERE date_time_field <= 'YYYY-MM-DD HH:MM:SS.0000'`) could lead to incorrect results due to improper `DATETIME` comparison. (Bug #47963)
- `XA START` could cause an assertion failure or server crash when it is called after a unilateral rollback issued by the Resource Manager (both in a regular transaction and after an XA transaction). (Bug #43171)
- The subquery optimizer had a memory leak. (Bug #48060)
- Configuring MySQL for DTrace support resulted in a build failure on Solaris if the directory for the `dtrace` executable was not in `PATH`. (Bug #45810)
- `SUM()` artificially increased the precision of a `DECIMAL` argument, which was truncated when a temporary table was created to hold the results. (Bug #48370)

  References: See also Bug #45261.
- `InnoDB` mishandled memory-allocation failures in the `os_mem_alloc_large()` function. (Bug #48237)
- `WHERE` clauses with *`outer_value_list`* `NOT IN` *`subquery`* were handled incorrectly if the outer value list contained multiple items at least one of which could be `NULL`. (Bug #48177)

- If an outer query was invalid, a subquery might not be set up. `EXPLAIN EXTENDED` did not expect this and caused a crash by trying to dereference improperly set up information. (Bug #48295)
- When a query used a `DATE` or `DATETIME` value formatted using any separator characters other than hyphen (`'-'`) and a `>=` condition matching only the greatest value in an indexed column, the result was empty if an index range scan was employed. (Bug #47925)
- If an `InnoDB` table was created with the `AUTO_INCREMENT` table option to specify an initial auto-increment value, and an index was added in a separate operation later, the auto-increment value was lost (subsequent inserts began at 1 rather than the specified value). (Bug #47125)
- `InnoDB` did not compile on some Solaris systems. (Bug #47058)
- `InnoDB` did not compile using `gcc` 4.1 on PowerPC systems. (Bug #46718)
- `InnoDB` raised errors in some cases in a manner not compatible with `SIGNAL` and `RESIGNAL`. (Bug #47233)
- Some queries with nested outer joins could lead to crashes or incorrect results because an internal data structure was handled improperly. (Bug #42116)
- `InnoDB` now ignores negative values supplied by a user for an `AUTO_INCREMENT` column when calculating the next value to store in the data dictionary. Setting `AUTO_INCREMENT` columns to negative values is undefined behavior and this change should bring the behavior of `InnoDB` closer to what users expect. (Bug #46965)
- After renaming a user, granting that user privileges could result in the user having privileges additional to those granted. (Bug #41597)
- In some cases, the server did not recognize lettercase differences between `GRANT` attributes such as table name or user name. For example, a user was able to perform operations on a table with privileges of another user with the same user name but in a different lettercase.

  In consequence of this bug fix, the collation for the `Routine_name` column of the `mysql.proc` table is changed from `utf8_bin` to `utf8_general_ci`. (Bug #41049)

  References: See also Bug #48872.
- A Valgrind error during index creation by `InnoDB` was corrected. (Bug #46657)
- On 64-bit systems, `--skip-innodb` did not skip `InnoDB` startup. (Bug #46043)
- `mysqladmin debug` could crash on 64-bit systems. (Bug #47382)
- Certain `INTERVAL` expressions could cause a crash on 64-bit systems. (Bug #48739)
- `GRANT` and `REVOKE` crashed if a user name was specified as `CURRENT_USER()`. (Bug #48319)
- Referring to a stored function qualified with the name of one database and tables in another database caused a “table doesn't exist” error. (Bug #18444)
- A `Table ... doesn't exist` error could occur for statements that called a function defined in another database. (Bug #17199)
- When building MySQL on Windows from source, the `WITH_BERKELEY_STORAGE_ENGINE` option would fail to configure `BDB` support correctly. (Bug #27693)
- The default database is no longer changed to `NULL` (“no database”) if `DROP DATABASE` for that database failed. (Bug #26704)

- There were cases where string-to-number conversions would produce warnings for `CHAR` values but not for `VARCHAR` values. (Bug #28299)
- `DROP TABLE` for `INFORMATION_SCHEMA` tables produced an `Unknown table` error rather than the more appropriate `Access denied`. (Bug #24062)
- `ALTER TABLE` statements that added a column and added a nonpartial index on the column failed to add the index. (Bug #31031)
- When an `InnoDB` tablespace filled up, an error was logged to the client, but not to the error log. Also, the error message was misleading and did not indicate the real source of the problem. (Bug #31183)
- For `const` tables that were optimized away, `EXPLAIN EXTENDED` displayed them in the `FROM` clause. Now they are not displayed. If all tables are optimized away, `FROM DUAL` is displayed. (Bug #30302)
- In `mysql`, using **Control+C** to kill the current query resulted in a `ERROR 1053 (08S01): Server shutdown in progress" message` if the query was waiting for a lock. (Bug #28141)
- `SELECT COUNT(DISTINCT)` was slow compared with `SELECT DISTINCT`. Now the server can use loose index scan for certain forms of aggregate functions that use `DISTINCT`. See Loose Index Scan. (Bug #21849, Bug #38213)