

MySQL 5.1.39 リリースノート（日本語翻訳）

- **パフォーマンス**: `bulk_insert_buffer_size` 値が 256KB を超える MyISAM テーブルでは、最大 100 行までの同時挿入時に、複数行の INSERT および INSERT ... SELECT 処理などの一括挿入操作のパフォーマンスが大幅に向上している ([Bug#44723](#))。
- **パーティショニング**: パーティションドテーブルの空パーティションに対して INSERT ... SELECT ステートメントを実行すると、`ERROR 1030 (HY000): Got error 124 from storage engine` で失敗していた。この問題は、LOAD DATA CONCURRENT INFILE ステートメントの進行中にパーティションテーブルに対して実行したクエリが同じエラーで失敗する原因にもなっていた ([Bug#46639](#))。

[Bug#35845](#)、[Bug#44657](#)、[Bug#45840](#) も参照。

- **パーティショニング**: パーティションドテーブルの TIMESTAMP カラムが CURRENT_TIMESTAMP のデフォルト値で、このカラムが ON UPDATE オプションで定義されていない場合、このテーブルに対して ALTER TABLE ... REORGANIZE PARTITION ステートメントを実行すると、TIMESTAMP カラム値が CURRENT_TIMESTAMP に設定されていた ([Bug#46478](#))。
- **パーティショニング**: テーブルのパーティショニングキーで TO_DAYS() 関数が使用されている場合、パーティションの解除が必ずしも正しく機能しなかった ([Bug#46362](#))。
- **パーティショニング**: パーティショニングサポート付きでコンパイルされた MySQL Server のバイナリでパーティショニングサポートが無効になっている場合、パーティションドテーブルにアクセスしようとする、サーバがクラッシュしていた ([Bug#39893](#))。
- **パーティショニング**: 日付値のカラムに無効な日付が保持されている場合に、パーティショニング式で TO_DAYS() を使用すると、選択違反の原因になった。これは、このような状況でこの関数が NULL を返し、NULL 値を保持しているパーティションは取り除かれるためである。たとえば、このようなテーブルの DATE カラムに '2001-02-00' が挿入され、このテーブルに対する以降のクエリで WHERE `date_col < '2001-02-00'` が使用された場合、'2001-01-01' は '2001-02-00' より小さいことから TO_DAYS('2001-02-00') は NULL として評価され、'2001-01-01' が保持されている行は返されないため、この問題が発生していた。今回のリリースでは、テーブルで RANGE または LIST パーティショニングが使用され、パーティショニング式に

TO_DAYS () が含まれている場合、NULL パーティションは無視されずにスキャンされるようになった。

この問題の修正により、RANGE または LIST によるパーティションドテーブルに対する SELECT * FROM *table* WHERE *date_col* < *date_val* 形式のクエリが、サーバ SQL モードに ALLOW_INVALID_DATES が含まれているかのように、実際にはクエリの発行時にこれがサーバ SQL の一部でない場合にも処理されるなどの不正動作も修正されている ([Bug#20577](#))。

[Bug#18198](#)、[Bug#32021](#)、[Bug#46362](#) も参照。

- **レプリケーション:** トランザクショナルテーブルに、非トランザクショナルテーブルを更新するトリガが存在する場合、トランザクショナルテーブルの AUTO_INCREMENT カラムの複数行更新を実行すると、マスタとスレーブが矛盾する可能性があった。マスタでこのような更新に失敗した場合、マスタでは行がまったく更新されない一方、スレーブでは一部の行が（誤って）更新されることがあった ([Bug#46864](#))。
- **レプリケーション:** --replicate-rewrite-db オプションを使用する場合、スレーブとの接続の切断時に、マスタでこのオプションによって参照されるデータベースがカレントデータベースであると、このデータベースに存在するテンポラリテーブルが正しくドロップされなかった ([Bug#46861](#))。
- **レプリケーション:** トランザクショナルテーブルと非トランザクショナルテーブルの両方を変更するステートメントに失敗した場合、マスタではトランザクショナルな変更が自動的にロールバックする一方、スレーブではエラーが無視され、変更がロールバックしないため、マスタとスレーブが矛盾する原因となっていた。

この問題は、失敗したステートメントをスレーブ上で自動的にロールバックさせることにより修正されている。ただし、対応する ROLLBACK ステートメントがリレーログファイルで見つからない場合、トランザクションはロールバックしない ([Bug#46130](#))。

[Bug#33864](#) も参照。

- **レプリケーション:** slave_transaction_retries が設定されている場合、複製後、スレーブでのデッドロックのためロールバックしたステートメントは、再試行されるはずである。ただし、特定の状況では、この変数が設定されていても、エラー1213

(Deadlock found when trying to get lock; try restarting transaction) でレプリケーションが停止していた ([Bug#45694](#))。

- レプリケーション: CREATE DATABASE IF NOT EXISTS、CREATE TABLE IF NOT EXISTS、および CREATE EVENT IF NOT EXISTS のバイナリログ動作 (およびそれによるレプリケーション動作) は、これらのステートメント間で、また DROP DATABASE IF EXISTS、DROP TABLE IF EXISTS、および DROP EVENT IF EXISTS と一貫していなかった。DROP ... IF EXISTS ステートメントは、ステートメントで指定されたデータベースオブジェクトが存在しない場合でも、常にログに記録される。一方、CREATE ... IF NOT EXISTS ステートメントでは、ステートメントで指定されたデータベースオブジェクトが存在する場合、CREATE EVENT IF NOT EXISTS のみ記録されていた。

今回のリリースでは、ステートメントで指定されたオブジェクトが存在するかどうかに関わらず、すべての CREATE ... IF NOT EXISTS ステートメントがバイナリログに記録されるようになった (これによって複製)。詳細については、[16.3.1.3「Replication of CREATE ... IF NOT EXISTS Statements」](#) を参照。

例外: CREATE TABLE IF NOT EXISTS ... SELECT のレプリケーションおよびログは、引き続き既存のルールに従って処理される。詳細については、[16.3.1.4「Replication of CREATE TABLE ... SELECT Statements」](#) を参照。

([Bug#45574](#))

- レプリケーション: ステートメントベースのレプリケーションを使用する場合、データベースレベルの文字セットが必ずしもレプリケーション SQL スレッドで尊重されなかった。これにより、マスタで LOAD DATA によって挿入されたデータが、不正な文字セットで複製される可能性があった。

注意

行ベースのレプリケーションを使用している場合、これは問題にならなかった。

([Bug#45516](#))

- レプリケーション: 場合によって、STOP SLAVE ステートメントにより、レプリケーションスレーブがクラッシュする可能性があった。これは、Windows または Macintosh プラットフォーム上の MySQL に固有の問題であった ([Bug#45238](#)、[Bug#45242](#)、[Bug#45243](#)、[Bug#46013](#)、[Bug#46014](#)、[Bug#46030](#))。

[Bug#40796](#)、[Bug#40796](#) も参照。

- **レプリケーション:** DEFINER 句が CURRENT_USER に設定されているか、または明示的に設定されていないスケジュールドイベントを作成すると、マスタとスレーブが矛盾する原因となっていた。どちらの場合も、この問題は、DEFINER がカレントスレッドの CURRENT_USER に設定されていることに起因する（マスタでは、CURRENT_USER は `mysql/d` ユーザであるのに対し、スレーブでは、CURRENT_USER は空）。

この動作は以下のように修正されている。

- DEFINER として CURRENT_USER が使用されている場合、CREATE EVENT ステートメントがバイナリログに書き込まれる前に、CURRENT_USER の値に置換される。
- DEFINER が明示的に設定されていない場合、バイナリログへの書き込み前に、CURRENT_USER の値を使用する DEFINER 句が CREATE EVENT ステートメントに追加される。

([Bug#44331](#))

[Bug#42217](#) も参照。

- **レプリケーション:** ステートメントベースログ形式を使用している場合、同じトランザクション内では、最初に非トランザクショナルテーブル (MyISAM テーブルなど) に対して更新を実行し、その後トランザクショナルテーブル (InnoDB ストレージエンジンを使用するテーブルなど) を更新するトランザクショナルステートメントと非トランザクショナルステートメントの組み合わせのみがセーフである。これは、非トランザクショナルテーブルに対する修正が即座に他の接続で認識できるのに対し、更新はすぐにはバイナリログに書き込まれないため、マスタとスレーブが矛盾する可能性があるためである（他の組み合わせでは、原因となる依存関係が隠されるため、非トランザクショナルテーブルを更新するステートメントを正しい順序でバイナリログに書き込むことができない）。

ただし、場合によっては、この状況が正しく処理されず、ステートメントがセーフかどうか必ずしも正しく判断されなかった。特に、トランザクショナルテーブルと非トランザクショナルテーブルの両方に影響するマルチテーブル更新や、トランザクショナルテーブルで動作するトリガを保持する非トランザクショナルテーブルのデータを修正する（またはその逆）ステートメントが、アンセーフであってもそのように判断されなかった。

この修正により、ステートメントベースログモードの同じトランザクション内でトランザクショナルテーブルと非トランザクショナルテーブルの更新を組み合わせた場合に、レプリケーションがセーフかどうかについて以下のように判断されるようになった。

1. 特定のトランザクションで非トランザクショナルテーブル内のデータを修正するステートメントは、同じトランザクション内でトランザクショナルテーブルにアクセスするデータ修正ステートメントより先に発行されていれば、セーフと見なされる。
2. トランザクショナルテーブルのみを更新するステートメントは、常にセーフと見なされる。
3. 同じトランザクション内でトランザクショナルテーブルと非トランザクショナルテーブルの両方に影響するステートメントは、常にアンセーフと見なされる。両方のテーブルを修正しない場合も、これに該当する。たとえば、`INSERT INTO innodb_table SELECT * FROM myisam_table`などのステートメントもアンセーフと見なされる。

注意

現時点での修正は、ステートメントベースログモードを使用している場合のみ有効である。将来的な MySQL のリリースでは、MIXED または ROW 形式を使用している場合に発生する同様の問題が解決される予定。

([Bug#28976](#))

- スタックオーバーフローチェックで、ヒープに格納されている構造のサイズが考慮されなかった ([Bug#46807](#)) 。
- 以下の要素を使用してクエリを実行すると、サーバがクラッシュする可能性があった。1. 最も外側の SELECT 内の「不可能な where」、2. 最も外側の SELECT 内の集計、3. トップレベルの WHERE sargable 述語として外部フィールドを含む WHERE 句を使用する、相関関係のあるサブクエリ ([Bug#46749](#)) 。
- 同名のテーブルがすでに存在し、そのテーブルに AUTO_INCREMENT カラムが含まれている場合、CREATE TABLE ... SELECT により表明違反が発生する可能性があった ([Bug#46616](#)) 。
- MERGE テーブルトリガに対して SHOW CREATE TRIGGER を実行すると、表明違反が発生していた ([Bug#46614](#)) 。

- ルーズなインデックススキャンアクセスメソッドが選択されたクエリで、同等の `col_name < 0` ではなく `col_name` 形式の条件を使用すると、表明違反が発生していた ([Bug#46607](#))。
- HANDLER で開いたテーブルに対して TRUNCATE TABLE を実行しても、ハンドラが閉じずに矛盾した状態で残され、これがサーバクラッシュの原因となる可能性があった ([Bug#46456](#))。
- FROM 句および PROCEDURE ANALYSE () にサブクエリを含むクエリを実行すると、サーバクラッシュが発生していた ([Bug#46184](#))。
- ソート実行中のクエリを強制終了すると、メモリリークが発生する可能性があった ([Bug#45962](#))。
- たとえば、リテラル DECIMAL 値からテーブルカラムのタイプを推測する場合などに、DECIMAL 値を切り捨てると、表明違反の原因となる可能性があった ([Bug#45261](#))。
- IS NULL 範囲の処理中にバッファオーバーフローが発生する可能性があった ([Bug#37044](#))。
- Windows システムで `mysqladmin --wait ping` がクラッシュしていた ([Bug#35132](#))。

※本翻訳は、理解のための便宜的な訳文として、オラクルが著作権等を保有する英語原文を NRI の責任において翻訳したものであり、変更情報の正本は英語文です。また、翻訳に誤訳等があったとしても、オラクルには一切の責任はありません。