

MySQL 5.6.20 リリースノート（日本語翻訳）

セキュリティノート

- **セキュリティ修正:** MySQL 5.6 Commercial Server のリンクされた OpenSSL ライブラリがバージョン 1.0.1g からバージョン 1.0.1h に更新された。1.0.1g およびそれ以前の OpenSSL のバージョンは、[CVE-2014-0224](#) に対して脆弱であると報告されている。

この変更は、代わりに **yaSSL** ライブラリを使用する MySQL Server 5.6 のオラクル社製 MySQL Community ビルドに影響しない (CVE-2014-0224)。

DTrace サポート

- 今回、MySQL には UEK カーネルを使用した Oracle Linux 6 以上での DTrace サポートが含まれるようになった。DTrace が存在する場合、特殊な **CMake** オプションを必要とせず、サーバのビルドがこれを検出する。MySQL での DTrace の使用については、「[Tracing mysqld Using DTrace](#)」を参照のこと。

InnoDB ノート

- **重要な変更:** 大規模な外部に保存される **BLOB** フィールドの REDO ログ書き込みは、最新のチェックポイントを上書きする可能性があった。5.6.20 パッチは、REDO ログ **BLOB** 書き込みを REDO ログファイルサイズの 10% に制限する。5.7.5 パッチは、制限を強制することなくバグに対処する。MySQL 5.5 の場合、バグは既知の制限のままである。

MySQL 5.6 に導入された REDO ログ **BLOB** 書き込み制限の結果として、[innodb log file size](#) は、テーブルの行にある最大の **BLOB** データサイズおよび他の変数の長さフィールド ([VARCHAR](#)、[VARBINARY](#)、および [TEXT](#) タイプのフィールド) の長さの 10 倍よりも大きな値に設定する必要がある。このように設定しない場合、“Row size too large” (行サイズが大きすぎる) エラーが発生する可能性があった。
[innodb log file size](#) 設定がすでに十分に大きい場合やテーブルに **BLOB** データが含まれていない場合、対応は必要ない (Bug #16963396、Bug #19030353、Bug #69477)。

追加または変更された機能

- **レプリケーション:** 新しいシステム変数 `binlog_impossible_mode` は、たとえばファイルエラーなどが原因でサーバがバイナリログに書き込みできない場合の動作をコントロールする。後方互換性については、`binlog_impossible_mode` のデフォルトは `IGNORE_ERROR` であり、これはサーバがエラーのログを記録して、ログを停止し、データベースへの更新を継続することを意味する。この変数を `ABORT_SERVER` に設定すると、サーバがバイナリログに書き込みできない場合はサーバがログを停止し、シャットダウンする (Bug #51014、Bug #11758766)。
- **CMake** サポートが更新され、**CMake** バージョン 3 を処理するようになった (Bug #19001781)。
- 5.6.17 で導入された新しい Debian7、Ubuntu12.04、および Ubuntu14.04 ディストリビューションサポートは、今回、`deb-precise`、`deb-wheezy`、および `deb-trusty` ディレクトリ内の `packaging` ディレクトリ直下に配置されたプラットフォーム固有のパッケージングソースとともに出荷されるようになった (Bug #19020385)。
- LinuxThreads のサポートはソースコードから削除された。Linux 2.6 において、LinuxThreads は NPTL に置き換えられた (Bug #17007529、Bug #72888、Bug #18913935)。
- デフォルトでは、`mysql_install_db` がテンプレートを使用して `my.cnf` ファイルをインストールベースディレクトリに作成する。一部のデプロイメントでは、これが望ましくない場合がある。この動作を抑制できるように、今回、`mysql_install_db` は `--keep-my-cnf` オプションをサポートするようになり、任意の既存の `my.cnf` ファイルを保持して、新規の `my.cnf` ファイルを作成しない (Bug #71600、Bug #18205019)。
- 今回、`mysqlhotcopy` ユーティリティは非推奨となり、MySQL の将来のバージョンから削除される。この理由の一部としては、これが `MyISAM` および `ARCHIVE` ストレージエンジンとのみ機能すること、また、Unix では機能するが Windows では機能しないことが挙げられる。代替には、`mysqldump` および MySQL Enterprise Backup などがある。
- `timed_mutexes` システム変数には影響なく、非推奨である。

修正されたバグ

- **重要な変更; レプリケーション:** `DROP TABLE` ステートメントは、これに通常の(テンポラリーではない)テーブルおよびテンポラリーテーブルが含まれている場合、あるいはこれにトランザクショナルと非トランザクショナルの両方のストレージエンジンを使用したテンポラリーテーブルが含まれている場合、バイナリログへ送信される前に複数のステートメントに分割されることがある。今回、GTID を使用することで、これらのテーブルの組み合わせに影響する `DROP TABLE` ステートメントは、`gtid_next` システム変数の値が `AUTOMATIC` でない限り許可されなくなった。これは、GTID がサーバ上で有効である場合に、各ステートメントに関連付けられた 1 つの GTID のみを含みながら、上述の状況で `DROP TABLE` を発行すると(SQL スレッドは `SET gtid_next='uuid: number'` に続いてこれを実行する)、元の `DROP TABLE` の分割に続いて結果として発生するすべてのステートメントへの割り当てに十分な GTID が存在しない場合、問題が発生することが理由である。

`DROP TABLE` ステートメントは、以下のようなテーブルの特性に従って、現在の変化するトランザクションに関するステートメントの動作が原因で分割されることがある。

- 通常の(テンポラリーではない)テーブルの `DROP TABLE` が即時にコミットされる。
- トランザクショナルストレージエンジンを使用したテンポラリーテーブルの `DROP TABLE` が現在のトランザクションにコミットされる(`COMMIT` の後)。
- 非トランザクショナルストレージエンジンを使用するテンポラリーテーブルの `DROP TABLE` が即時にコミットされる。

単一の `DROP TABLE` ステートメント内における、このようなテーブルタイプ 3 つすべてのネーミングは、MySQL サーバが元のステートメントをバイナリログ内で 3 つの個別の `DROP TABLE` ステートメントに分割する原因となる。GTID が有効になっていても `gtid_next` の値が `AUTOMATIC` ではない場合、上述のテーブルタイプのいずれかを混合する `DROP TABLE` ステートメントの発行は、サーバが結果として発生する全ステートメントを使用してバイナリログに書き込みを行うための GTID の個数が不足する原因となる。また、`DROP TABLE IF EXISTS` については、テーブルの一部またはすべてが存在していない場合であっても、ステートメント内で指定されたすべてのテーブルのバイナリログ内で常に書き込みが行われる。

トランザクショナルまたは非トランザクショナルのストレージエンジンを使用しているかどうかに従って、テンポラリテーブルは **DROP TABLE** によって異なる方法で処理されるため、存在しない **DROP TEMPORARY TABLE** ステートメントによって名前が付けられたテーブルは、すべてトランザクショナルであると仮定される。このことは、2 つの非トランザクショナルテンポラリテーブルが含まれた **DROP TEMPORARY TABLE** がマスタ上で発行される場合、両方のテーブルに名前を付ける 1 つの **DROP TABLE** ステートメントのみを書き込むことを意味する。テンポラリテーブルのいずれかがスレーブ上に存在しなくなった場合、SQL スレッドがステートメントを実行する際には、非トランザクショナル(ただし既存の)テンポラリテーブルおよび非既存のトランザクショナルテンポラリテーブルに影響するため、これを複数のステートメントに分割しようとする。このことは、SQL スレッドに元の **DROP TABLE** ステートメントの 1 つの GTID のみが含まれていても、バイナリログ内に 2 つの **DROP TABLE** ステートメントを書き込む必要があるため、問題につながる。

また、マスタが再起動したことを検出した後にスレーブがテンポラリテーブルをドロップした場合、1 つの疑似スレッドあたり、および 1 つのデータベースあたり(ただし、単一の **DROP TABLE** ステートメント内のトランザクショナルおよび非トランザクショナルのストレージエンジンを使用して組み合わされたテンポラリテーブルの)、1 つの **DROP TABLE** ステートメントのログを記録する。

今回、**gtid_next** が *uuid.number* 値に設定され、上述の任意のテーブルタイプを混合して **DROP TABLE** ステートメントが発行される場合、クライアントセッション内でエラーをスローするようになった。

また、今回、少なくとも 1 つのトランザクショナルテンポラリテーブルがステートメントによってドロップされる場合にのみ、非既存のテンポラリテーブルをグループ化し、これらがトランザクショナルであると仮定するようになった。どのトランザクショナルテンポラリテーブルもドロップされない場合、非既存のテンポラリテーブルは、すべて非トランザクショナルテンポラリテーブルであると仮定される。

今回、マスタによる再起動の場合、スレーブもテンポラリテーブルのドロップを適切に処理するようになった(Bug #17620053)。

InnoDB: 何千もの子テーブルが含まれた親テーブルを開くことが、long semaphore wait 条件の原因となる可能性があった (Bug #18806829)。

InnoDB: 単一アイテムの全文検索の場合、削除されたドキュメントが 逆文書出現頻度 inverse document frequency (IDF) 計算に含まれ ていたていなかった (Bug #18711306、Bug #72548)。

InnoDB: `mysqld` MySQL 起動時開始時、innodb data file path オプションを使用して複数のデータファイルを指定すると、2 番目のファイルにデータが書き込まれた後、`Space id in fsp header(fsp ヘッダ内のスペース id)` エラーを返すことがあった (Bug #18767811)。

InnoDB: 全文検索インデックスを使用したテーブル上での **DELETE** 操作は、アサーションを発生させた (Bug #18683832)。

参照先: Bug #14639605 も参照のこと。

InnoDB: `memcached flush_all` コマンドを呼び出す際、**InnoDB** が接続およびトランザクションを初期化しようとする。トランザクションが **TRX_STATE_NOT_STARTED** 状態の場合、**InnoDB** は **CONN_DATA->CRSR_TRX** の NULL への設定に失敗することがあり、深刻なエラーの原因となった (Bug #18652854)。

InnoDB: **InnoDB** が共有ライブラリとしてビルドされる際、**InnoDB** 全文検索 (FTS) の **INFORMATION_SCHEMA** プラグインが `Can't open shared library 'ha_innodb.so' (共有ライブラリ 'ha_innodb.so' を開けません)` エラーとともに失敗することがあった (Bug #18655281、Bug #70178)。

InnoDB: MySQL 5.6.5 で導入されたリグレーションは、innodb file per table が有効になっても、システムテーブルスペース (スペース 0) 内に全文検索インデックステーブルが作成される原因となった (Bug #18635485)。

InnoDB: **InnoDB memcached** プラグインは、`lock_plugin` ミューテックスを取得せずに `plugin_del` を呼び出すことがあった。このバグ修正は、`ib_cursor_delete_row` 内での競合状態にも対処する (Bug #18409840)。

InnoDB: Bug#16418661 の修正が、必要以上の `buf_flush_list()` ロジックを **InnoDB** 起動コードに追加した (Bug #17798076、Bug #70899)。

InnoDB: Duplicate FTS_DOC_ID(重複 FTS_DOC_ID) エラーおよび Cannot find index FTS_DOC_ID_INDEX in InnoDB index translation table(インデックス FTS_DOC_ID_INDEX が InnoDB インデックス解釈テーブル内で見つからない) エラーの結果である `fts_get_next_doc_id` 内での競合状態 (Bug #17447086、Bug #70311)。

参照先: Bug #16469399 も参照のこと。

InnoDB: 異なるプロセッサタイプでのメモリ順序付けにおける差異により、一部のミューテックスおよび読み取り-書き込みロックフラグが一貫性をもって読み取られなかった (Bug #11755438、Bug #47213)。

パーティショニング: プライマリキー内に複数のカラムが含まれ、`LIST COLUMNS (R)` (この場合、`R` はプライマリキー定義内に一覧で掲載された最後の (右端の) カラム) によってパーティショニングされたテーブルが、誤った結果を返した (Bug #17909699、Bug #71095)。

レプリケーション: `mysqlbinlog --raw` は、失敗した書き込みが原因で発生したエラーをチェックせず、これはバイナリログのサイレントでの破損の結果となる可能性があった。今回、このような場合はエラーによって停止するようになった (Bug #18742916、Bug #72597)。

レプリケーション: スレーブのワークスレッドが大きすぎるステートメントを実行しようとした場合に、結果として発生するエラーがクラッシュの原因となった。今回、このような場合には、バッファのサイズに適合するためにエラーが切り捨てられるようになった (Bug #18563480)。

レプリケーション: 行ベースのレプリケーションを使用する際、スレーブ上に存在しなかったマスタ上で行を更新または削除すると、変更を処理しようとする場合にスレーブの失敗につながった。この問題は、プライマリキーが欠けている **InnoDB** テーブルで発生した (Bug #18432495、Bug #72085)。

レプリケーション: 引用符は、バイナリログへの書き込み時、`LOAD DATA INFILE` によって常に正しく処理されていたわけではなかった (Bug #18207212、Bug #71603)。

レプリケーション: MySQL 5.6.20 以降、現在の `AUTO_INCREMENT` 値と、現在および影響を受ける値の行の数の合計の間の範囲外にユーザが `AUTO_INCREMENT` 値を指定した場合、正しくレプリケーションが行われるようになった。以前のバージョンでは、ユーザが `AUTO_INCREMENT` 値を範囲外に指定した場合でも、スレーブによってエラーが生成された (Bug #17588419、Bug #70583)。

レプリケーション: ロックの取得に関与したスレッドのグループは、以下のイベントの発生時にデッドロックする可能性があった。

- a. ダンプスレッドがスレーブから再接続する。マスタ上では新しいダンプスレッドがゾンビダンプスレッドを強制終了しようとする。スレッドの `LOCK_thd_data` は取得済みであり、`LOCK_log` を取得するところである。
- b. バイナリログの表示を実行しているアプリケーションスレッドは、`LOCK_log` を取得済みであり、`LOCK_index` を取得するところである。
- c. `PURGE BINARY LOGS` を実行しているアプリケーションスレッドは、`LOCK_index` を取得済みであり、`LOCK_thread_count` を取得するところである。
- d. `SHOW PROCESSLIST` (または `SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST`) を実行しているアプリケーションスレッドは、`LOCK_thread_count` を取得済みであり、ゾンビダンプスレッドの `LOCK_thd_data` を取得するところである。

これは、ここにスレッドを一覧で示したのと同じ順序で 4 つのスレッドデッドロックにつながる。

この問題は、`LOCK_log` および `LOCK_index` の順序ルール、さらには `LOCK_thread_count` および `LOCK_thd_data` を順序付けるためのルールが存在するために発生するが、これら 2 つのロックのセット間における順序付けのルールは存在しない。これは、`SHOW PROCESSLIST` によって呼び出された内部 `mysqld_list_processes()` 関数が、関数の完全なライフタイムにおいて `LOCK_thread_count` を取得したこと、さらには各スレッドの `LOCK_thd_data` を取得およびリリースしたことが理由である。今回、この関数はグローバルスレッドリストからスレッドのコピーを取得するようになり、これらにおいて走査を実行するようになったが、これは

`LOCK_thread_count` のリリース後にのみ行われる。この走査中、`LOCK_thd_remove` を使用してグローバルスレッドリストの削除がブロックされるため、そうでなければ削除によって破壊されるはずであったコピーが走査中に有効なままであり続ける。この修正に従ったロック順序をここに示す。

```
LOCK_thd_remove -> LOCK_thd_data -> LOCK_log -> LOCK_index ->
LOCK_thread_count
```

(Bug #17283409、Bug #69954)

- **レプリケーション:** Windows では、アクティブな MySQL サーバからエラーログファイルが削除された(欠落していた)場合、**mysqldump** が失敗した(Bug #17076131)。
- **レプリケーション:** 異なる MySQL バージョンからのテンポラルタイプのフィールド (TIMESTAMP, DATETIME, TIME のような)を含むテーブルのレプリケーションは、互換性のない TIMESTAMP 型が原因で失敗した。これは、以前のバージョンではサポートされていない MySQL 5.6.4 に追加された小数の TIMESTAMP 形式によって引き起こされてた。現在ではどちらの形式でも TIMESTAMP をレプリケーションできる。(Bug #70124, Bug #17532932)
- **レプリケーション:** クライアントアプリケーションは、`BINLOG_DUMP_NON_BLOCK` フラグを初期のハンドシェイクパケット(`COM_BINLOG_DUMP`)内に設定できる必要がある。フラグを設定していない `COM_BINLOG_DUMP` を発行しているサーバへ接続しているクライアントは、サーバがバイナリログ内の最後のイベントを送信した際に EOF を取得せず、これは接続がブロックする原因となる。MySQL 5.6.5 ではエラーで削除されたこのフラグは、今回、現在のリリースにおいて復元された。

この修正の一部として、新しい `--connection-server-id` オプションが **mysqlbinlog** に追加された。このオプションは、この問題の有無の検証を目的としてクライアントが MySQL サーバをテストするために使用できる (Bug #71178、Bug #18000079)。

- **SELECT** がその **FROM** リスト内に結合で導出されたテーブルを含んでいて、**SELECT** リストが **COUNT (DISTINCT)** を含んでいた場合、基礎となる結果のセットが空であっても **COUNT ()** が 1 を返した (Bug #18853696)。

参照先: このバグは Bug #11760197 のリグレッションである。

- オプティマイザトレースを有効にすると、**HAVING** 句内のサブクエリを使用したクエリをサーバが終了する原因となる可能性があった (Bug #18791851)。
- SHA および MD5 の関数は、内部 **filename** 文字セットを使用した操作に失敗し、サーバ終了の原因となる可能性があった (Bug #18786138)。
- **mysqldump** に渡された大きな引数は、バッファオーバーフローおよびプログラム終了につながる可能性があった (Bug #18779944)。
- コンパイラフラグが DTrace に渡されず、64 ビットプラットフォーム上での 32 ビットビルドのクロスコンパイルでの問題の原因となった (Bug #18593044)。
- パーティションされたテーブル上の **ALTER TABLE** は、誤ったストレージエンジンがテーブルの **.frm** ファイルに書き込まれ、**SHOW CREATE TABLE** に表示される原因となる可能性があった (Bug #18618561)。
- **max heap table size** システム変数が大きな値 (20GB) に設定されている場合、テンポラリテーブルまたは **MEMORY** ストレージエンジンを使用したテーブルの作成は、サーバ終了の原因となった (Bug #18463911)。
- MySQL が **-DINSTALL_LIBDIR=lib64** オプションを指定してビルドされた場合や、MySQL パッケージが異なるインストールプレフィックスにアンパックされた場合、**mysql_config** は機能しなかった。また、RPM ビルドでも間違ったインストールプレフィックスを使用するものがあったため、**mysql_config** は機能しなかった。(Bug #18382225)
- デバッグビルドの場合、**ujis_japanese_ci**、**utf8mb4_turkish_ci**、または **eucjpms_bin** 照合を使用した全文クエリ文字列における **0x00** 文字は、アサーションを発生させる可能性があった (Bug #18277305)。
- **mysqladmin password** は、コマンドラインに与えられた古いパスワードをマスクしたが、新しいパスワードはマスクしなかった (Bug #18163964)。
- yaSSL コードでは、バッファオーバーフローの原因となる可能性がある、Off-by-one エラーが証明書デコーディングで発生した。

yaSSL コードには、対応する **closedir()** のない **opendir()** が含まれていた (Bug #18178997、Bug #17201924)。

- **InnoDB** テーブルでの全文クエリの場合、削除されたドキュメント ID へのアクセスの試行がサーバ終了につながる可能性があった (Bug #18079671)。
- **MyISAM** テンポラリファイルをコード実行攻撃のマウントに使用することができた (Bug #18045646)。
- パフォーマンススキーマ **events statements current** テーブルから選択されたクエリの場合、**ORDER BY** 句の追加は不正確な結果を生成する原因となる可能性があった (Bug #17729044)。
- クエリに **MIN()** / **MAX()** および *aggregate_function* (**DISTINCT**) (たとえば **SUM(DISTINCT)**) の両方が含まれていて、ルーズなインデックススキャンを使用して実行された場合、**MIN()** / **MAX()** の結果値が不適切に設定された (Bug #17217128)。
- **UNION** ステートメントの場合、調査済み行の値が不正確に計算されていた。これは、パフォーマンススキーマステートメントのテーブル (**events statements current** など) の **ROWS_EXAMINED** カラムに対して大きすぎる値であるためマニフェストであった (Bug #17059925)。
- クライアントはアカウントが存在したかどうか、コネクションエラーメッセージの内容に基づいて決定することができた。 (Bug #16513435, Bug #17357528, Bug #19273967)
- クライアントは、アカウントが存在しているかどうかをエラーメッセージの内容に基づいて判断する可能性があった (Bug #16513435, Bug #17357528)。
- **InnoDB** カラム内で **TINYBLOB** または **GEOMETRY** カラムのプリフィックスにインデックスを作成する際、アサーションが発生する可能性があった (Bug #16368875, Bug #18776592, Bug #17665767)。
- マルチバイト文字列内で先頭のスペースをスキップするために非マルチバイトアルゴリズムを使用することは、サーバ終了の原因となる可能性があった (Bug #12368495, Bug #18315770)。
- **UNION** で定義されたビューの場合、サーバが無効なビュー定義を作成する可能性があった (Bug #65388, Bug #14117018, Bug #72018, Bug #18405221)。
- サーバなしのクライアントをビルドするために **cmake -DWITHOUT_SERVER** を構成することが、ソースツリー外部のビルドに対して失敗した (Bug #66000, Bug #14367046)。

- **big_tables** が有効になっている場合、非重複キーで定数の等価条件が含まれた単純な結合において **COUNT (DISTINCT)** を使用したクエリは、不正確な結果を返した (Bug #52582、Bug #11760197)。

参照先: Bug #18853696 も参照のこと。

- 3 つのスレッドが同時に **INSTALL PLUGIN**、**SHOW VARIABLES**、および **mysql_change_user()** を実行した場合、デッドロックが発生する可能性があった (Bug #71236、Bug #18008907、Bug #72870、Bug #18903155)。
- 空の引数を **--login-path** オプションに提供する際、**mysql_config_editor** が終了した (Bug #71837、Bug #18311024、Bug #18830493)。
- Bison 3 では MySQL がコンパイルしなかった。回避策は、Bison 2 にダウングレードすることである (Bug #71250、Bug #18017820、Bug #18978946)。
- 準同期レプリケーションがアクティブである際の準同期レプリケーションプラグインのアンインストールおよび再インストールは、レプリケーションの失敗の原因となった。今回、プラグインはアンインストール可能であるかどうかをチェックするようになり、準同期レプリケーションがアクティブな場合はエラーを生成するようになった。マスタ側のプラグインをアンインストールするには、準同期スレーブが存在してはならない。スレーブ側のプラグインをアンインストールするには、実行中の準同期 I/O スレッドが存在してはならない (Bug #70391、Bug #17638477)。
- **MIN()** または **MAX()** によって参照されるカラム内に述語が存在していた場合で、先行する重文インデックス内のキー部分におけるすべての論理和内にその述語が存在していなかった場合、ルーズなインデックススキャンは不正確な結果を返した (Bug #71097、Bug #17909656)。
- **MYSQL_OPT_RECONNECT** が有効になっていたとしても、クライアントの自動再接続は、**libmysqlclient** に対してリンクされたクライアントには機能しなかった (Bug #70026、Bug #17309863)。
- RPM パッケージを使用したアップグレードは、インストールディレクトリの所有権を変更する可能性があった (Bug #71715、Bug #18281535)。
- プロキシユーザは、プロキシユーザパスワードの期限が切れた場合、ステートメントを実行できなかった (Bug #71337、Bug #18057562)。

- 新しい CMake オプションである `SUNPRO_CXX_LIBRARY` は、Solaris 10 以降、`stlport4` ではなく `libCstd` に対するリンク付けを有効にする。これは、サーバが C++98 に依存するため、クライアントコードに対してのみ機能する。使用例:

```
cmake -DWITHOUT_SERVER=1 -DSUNPRO_CXX_LIBRARY=Cstd
```

(Bug #72352、Bug #18605389)

- いくつかのテストおよび構成ファイルのファイルパーミッションと行末はより一貫性を持つようになり、パッケージチェック機能による警告を回避するようになった (Bug #68521、Bug #16415173、Bug #16395459、Bug #68517、Bug #16415032、Bug #71112、Bug #17919313、Bug #71113、Bug #17919422)。

※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。