

MySQL 5.6.26	パラメータ	MySQL 5.7.8	パラメータ	概要	関連URL
binlog_error_action	IGNORE_ERROR	binlog_error_action	ABORT_SERVER	IGNORE_ERROR で、サーバーがエラーのログを記録し、ロギングを停止してから、更新の実行を継続することを意味します。これは、古いバージョンの MySQL Server との下位互換性を提供するためです。この変数を ABORT_SERVER に設定すると、サーバーがバイナリログに書き込まないときはロギングを停止し、シャットダウンします。	https://dev.mysql.com/doc/refman/5.6/ja/replication-options-binary-log.html#sysvar_binlog_error_action
binlog_format	STATEMENT	binlog_format	ROW	行ベース、ステートメントベース、または複合型のレプリケーションのいずれれを使用するか指定します。ステートメントベースは MySQL 5.6 のデフォルトです。レプリケーション形式を参照してください。	https://dev.mysql.com/doc/refman/5.6/ja/server-options.html#option_mysql_binlog_format
binlog_gtid_simple_recovery	OFF	binlog_gtid_simple_recovery	ON	MySQL バージョン 5.6.21 では、この変数は simplified_binlog_gtid_recovery として追加され、MySQL バージョン 5.6.23 では、その名前が binlog_gtid_recovery_simplified に変わりました。デフォルトでは、MySQL はクラッシュから立ち上がる時に、バイナリログファイルを回復して一番古いファイルから始めて GTID イベントを検索するため、大量のバイナリログファイルがある場合はこれに時間がかかることがあります。このオプションを有効にすることで、代わりに一番新しいバイナリログファイルから GTID イベントを検索されます。	http://dev.mysql.com/doc/refman/5.6/ja/replication-options-binary-log.html#sysvar_binlog_gtid_recovery_simplified
eq_range_index_dive_limit	10	eq_range_index_dive_limit	200	最適化が限定する行数を推定するときに、インデックスタイプの使用からインデックス統計の使用に切り換える場合の等価比較条件内の等価範囲の数を指定します。これは次に示す同等のいずれかの形式を持つ式の評価に適用され、このとき最適化は一意でないインデックスを使用して col_name 値を参照します。統計の使用を無効にして、常にインデックスタイプを使用するには、eq_range_index_dive_limit を 0 に設定します。最適な推定を行うためにテーブルインデックス統計を更新するには、ANALYZE TABLE を使用します。	https://dev.mysql.com/doc/refman/5.6/ja/server-system-variables.html#sysvar_eq_range_index_dive_limit
innodb_buffer_pool_dump_at_shutdown	OFF	innodb_buffer_pool_dump_at_shutdown	ON	次回再起動時のウォームアッププロセスの時間を短縮するために、MySQL サーバーのシャットダウン時に、InnoDB のバッファプールにキャッシュされるページを記録するかどうかを指定します。一般に、innodb_buffer_pool_load_at_startup と組み合わせて使用されます。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_buffer_pool_dump_at_shutdown
innodb_buffer_pool_instances	8	innodb_buffer_pool_instances	1	InnoDB のバッファプールが分割される領域の数です。バッファプールが数 G バイトの範囲にあるシステムでは、バッファプールを個別のインスタンスに分割すると、キャッシュされたページに対して異なるスレッドが読み取りおよび書き込みを行うときの競合が減るため、並列性が向上する場合があります。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_buffer_pool_instances
innodb_buffer_pool_load_at_startup	OFF	innodb_buffer_pool_load_at_startup	ON	MySQL サーバーの起動時に、以前に保持されたときと同じページをロードすることで、InnoDB のバッファプールが自動的にウォームアップされるように指定します。一般に、innodb_buffer_pool_dump_at_shutdown と組み合わせて使用されます。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_buffer_pool_load_at_startup
innodb_checksum_algorithm	innodb	innodb_checksum_algorithm	crc32	値 innodb は、すべての MySQL バージョンとの下位互換性があります。値 crc32 では、より高速に、変更されたすべてのブロックのチェックサムを計算し、ディスク読み取りごとにチェックサムをチェックするアルゴリズムが使用されます。テーブルスペース内のブロックが crc32 アルゴリズムを使用するように変更されたあとは、関連付けられたテーブルを以前のバージョンの MySQL で読み取ることができません。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_checksum_algorithm
innodb_file_format	Antelope	innodb_file_format	Barracuda	新しい InnoDB テーブルで使用されるファイル形式です。現在は、Antelope および Barracuda がサポートされています。これは、独自のテーブルスペースを持つテーブルにのみ適用されるため、これを有効にするには、innodb_file_per_table が有効になっている必要があります。テーブルの冗雑などの特定の InnoDB 機能を使用するには、Barracuda ファイル形式が必要です。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_file_format
innodb_file_format_max	Antelope	innodb_file_format_max	Barracuda	サーバーの起動時に InnoDB によって、この変数の値がシステムテーブルスペースのファイル形式タグ (Antelope や Barracuda など) に設定されます。サーバーで「大きい」ファイル形式のテーブルが作成されたり、開かれたりすると、innodb_file_format_max の値がそのファイル形式に設定されます。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_file_format_max
innodb_large_prefix	OFF	innodb_large_prefix	ON	このオプションを有効にすると、DYNAMIC および COMPRESSED 行フォーマットを使用する InnoDB テーブルで、767 バイトよりも長い (最大で 3072 バイトの) インデックスキープリフィクスが許可されます。(このようなテーブルの作成には、innodb_file_format=barracuda および innodb_file_per_table=true のオプション値も必要になります。)	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_large_prefix
innodb_log_buffer_size	8388608	innodb_log_buffer_size	16777216	ディスク上のログファイルに書き込む際に InnoDB で使用されるバッファのサイズ (バイト単位) です。デフォルトの値は 8M バイトです。ログバッファを大きくすると、トランザクションがコミットする前にディスクにログを書き込まなくても、大規模なトランザクションを実行できます。したがって、多数の行を更新、挿入、または削除するトランザクションの場合、ログバッファを大きくすると、ディスク I/O を節約できます。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_log_buffer_size
innodb_purge_threads	1	innodb_purge_threads	4	InnoDB のページ操作専用のバックグラウンドスレッドの数です。MySQL 5.6.5 での新しいデフォルトの最小値である 1 は、ページ操作がマスタースレッドの一部としてではなく、常にバックグラウンドスレッドで実行されることを表します。ゼロ以外の値にすると、1 つ以上のバックグラウンドスレッドでページ操作が実行されるため、InnoDB 内の内部競合を削減でき、拡張性が改善されます。この値を 1 よりも大きくすると、数多くの個別のページスレッドが作成されるため、複数のテーブル上で DML 操作が実行されるシステムの効率性を改善できます。最大値は 32 です。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_purge_threads
innodb_strict_mode	OFF	innodb_strict_mode	ON	innodb_strict_mode を ON にすると、InnoDB は特定の条件に対応した警告ではなく、エラーを返します。厳密モードは、SQL 内の無視できる誤字や構文エラー、または操作モードと SQL ステートメントのさまざまな組み合わせによる意図しないその他の結果から保護する際に役立ちます。innodb_strict_mode を ON にすると、InnoDB は警告を発行して、指定されたステートメントを処理する (意図しない動作が伴う可能性があります) のではなく、特定のケースでエラー状況が発生します。これは、MySQL で受け入れられる SQL 構文を制御し、警告なしでエラーを無視するのかが、入力構文とデータ値を検証するのを決定する MySQL の sql_mode と類似しています。	http://dev.mysql.com/doc/refman/5.6/ja/innodb-parameters.html#sysvar_innodb_strict_mode
performance_schema_setup_actors_size	100	performance_schema_setup_actors_size	-1	performancece_schema関連	http://dev.mysql.com/doc/refman/5.6/ja/performance-schema-system-variables.html
performance_schema_setup_objects_size	100	performance_schema_setup_objects_size	-1	performancece_schema関連	http://dev.mysql.com/doc/refman/5.6/ja/performance-schema-system-variables.html
sql_mode	NO_ENGINE_SUBSTITUTION	sql_mode	ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, NO_ZERO_IN_DATE, NO_ZERO_DATE, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENGINE_SUBSTITUTION	モードは MySQL がサポートする SQL 構文と、MySQL が実行するデータ検証に影響します。これにより、MySQL をさまざまな環境で使用したり、MySQL をほかのデータベースサーバーと一緒に使用したりすることが、さらに容易になります。	http://dev.mysql.com/doc/refman/5.6/ja/sql-mode.html
sync_binlog	0	sync_binlog	1	この変数の値が 0 より大きい場合は、sync_binlog コミットグループがバイナリログに書き込まれたあとに、MySQL サーバーはそのバイナリログをディスクに同期します (fdatasync() を使用)。sync_binlog のデフォルト値は 0 で、これはディスクに同期しません。この場合、サーバーはオペレーティングシステムに依存して、ほかのファイルに関してバイナリログの内容をときどきフラッシュします。値 1 が一番安全な選択です (クラッシュの場合にバイナリログから失われるコミットグループが最大で 1 つです)。しかし、一番遅い選択でもあります (ディスクにバッチ付きキャッシュがある場合を除きます。その場合は同期が非常に速くなります)。	https://dev.mysql.com/doc/refman/5.6/ja/replication-options-binary-log.html#sysvar_sync_binlog
table_open_cache_instances	1	table_open_cache_instances	16	開いているテーブルキャッシュインスタンスの数 (デフォルトは 1)。セッション間の競合を減少させることでスケラビリティを改善するために、開いているテーブルキャッシュを、サイズが table_open_cache / table_open_cache_instances の 1/10 以下の小さいキャッシュインスタンスにパーティション化できます。DML ステートメントでは、セッションはインスタンスにアクセスするために、1 つのインスタンスのみをアクセスする必要があります。このセッションは複数インスタンスにわたってアクセスし、多くのセッションがテーブルにアクセスする場合はキャッシュを使用する演算の高いパフォーマンスが可能になります。(DDL ステートメントでは引き続きキャッシュ全体のログが必要ですが、そのようなステートメントは DML ステートメントよりも頻度がずっと低くなります。)通常 16 以上のコアを使用するシステムでは、8 または 16 の値が推奨されます。	https://dev.mysql.com/doc/refman/5.6/ja/server-system-variables.html#sysvar_table_open_cache_instances
warning_count	0	warning_count	1	メッセージを生成した最後のステートメントから得られたエラー、警告、および注意の数。この変数は読み取り専用です。	https://dev.mysql.com/doc/refman/5.6/ja/server-system-variables.html#sysvar_warning_count