

■PHPのMySQLi 関数一覧

※「chk」欄は、PHPのMySQL連携プログラミングで、利用頻度と処理上の必要度から判断した優先度です。

作成: 2015.12.21 / 修正 2016.06.08 yoshi

| No. | 関数:オブジェクト指向型 | 関数:手続き型 | 機能 | chk | サンプル(オブジェクト指向型)他 |
|-------------------|------------------------------|-----------------------------|--|-----|---|
| mysqli クラス | | | | | |
| 1 | mysqli::\$affected_rows | mysqli_affected_rows | 直近の INSERT、UPDATE、REPLACE、DELETE クエリにより変更された行の数を返します。SELECTは、mysqli_num_rows() と同様に動作 返り値の“-1” は、クエリエラーを示す | ○ | <code>\$db->query("INSERT INTO persons (name,code) VALUES('理香','K001')"); echo \$db->affected_rows;</code> |
| 2 | mysqli::autocommit | mysqli_autocommit | データベース更新の自動コミットをオン(TRUE)またはオフ(FALSE)にする | ○ | <code>\$db->autocommit (FALSE);</code> |
| 3 | mysqli::begin_transaction | mysqli_begin_transaction | トランザクションを開始する(InnoDB,NDBエンジンのみ、 PHP5.5以降) <code>\$db->begin_transaction(\$flags);</code> // \$flagsは、以下を選択可能 MYSQL_TRANS_START_READ_ONLY MYSQL_TRANS_START_READ_WRITE MYSQL_TRANS_START_WITH_CONSISTENT_SNAPSHOT | △ | <code>\$db->begin_transaction (MYSQLI_TRANS_START_READ_ONLY); \$db->query("SELECT * FROM persons"); \$db->commit(); \$db->close();</code> |
| 4 | mysqli::change_user | mysqli_change_user | 指定されたデータベース接続のユーザー名を変更する | △ | |
| 5 | mysqli::character_set_name | mysqli_character_set_name | データベース接続のデフォルトの文字コードセットを返す | | |
| 6 | mysqli::\$client_info | mysqli_get_client_info | MySQL クライアント情報を取得する | △ | |
| 7 | mysqli::\$client_version | mysqli_get_client_version | MySQL クライアントのバージョンを文字列で返す | △ | |
| 8 | mysqli::close | mysqli_close | 事前にオープンしているデータベース接続を閉じる 持続的でない MySQL 接続や結果セットは、PHP スクリプトの実行が終了する時点で自動的に破棄されますが、持続的な接続を閉じません。 | ◎ | <code>\$db->close();</code> |
| 9 | mysqli::commit | mysqli_commit | 現在のトランザクションをコミットする | ○ | <code>\$db->commit();</code> |
| 10 | mysqli::\$connect_errno | mysqli_connect_errno | 直近の接続コールに関するエラーコードを返す | ○ | <code>echo mysqli_connect_errno();</code> |
| 11 | mysqli::\$connect_error | mysqli_connect_error | 直近の接続エラーの内容を文字列で返す | ◎ | <code>echo mysqli_connect_error();</code> |
| 12 | mysqli::__construct | mysqli_connect | 新規に MySQL サーバーへの接続をオープンする | ◎ | <code>public function __construct(\$host, \$user, \$pass, \$db) { ... }</code> |
| 13 | mysqli::debug | mysqli_debug | デバッグ操作を行う 実行するデバッグ操作を表す文字列をパラメータに設定 | △ | |
| 14 | mysqli::dump_debug_info | mysqli_dump_debug_info | デバッグ情報をログに出力する | △ | |
| 15 | mysqli::\$errno | mysqli_errno | 直近の関数コールによるエラーコードを返す | ○ | <code>echo \$db->errno;</code> |
| 16 | mysqli::\$error_list | mysqli_error_list | 直近で実行したコマンドからのエラーの一覧を返す | ○ | <code>print_r(\$db->error_list);</code> |
| 17 | mysqli::\$error | mysqli_error | 直近のエラーの内容を文字列で返す | ○ | <code>echo \$db->error;</code> |
| 18 | mysqli::\$field_count | mysqli_field_count | 直近のクエリのカラムの数を返す | △ | |
| 19 | mysqli::get_charset | mysqli_get_charset | 文字セットオブジェクトを返す | △ | |
| 20 | mysqli::get_client_info | mysqli_get_client_info | MySQL クライアント情報を取得する | △ | |
| 21 | mysqli_get_client_stats | mysqli_get_client_stats | クライアントのプロセス単位の統計情報を返す | △ | |
| 22 | mysqli::\$client_version | mysqli_get_client_version | MySQL クライアントのバージョンを整数値で返す | △ | |
| 23 | mysqli::get_connection_stats | mysqli_get_connection_stats | クライアント接続に関する統計情報を返す(PHP5.3以降) | △ | |
| 24 | mysqli::\$host_info | mysqli_get_host_info | 使用している接続の型を文字列で返す | △ | <code>echo \$db->host_info;</code> |
| 25 | mysqli::\$protocol_version | mysqli_get_proto_info | 使用している MySQL プロトコルのバージョンを返す | △ | |
| 26 | mysqli::\$server_info | mysqli_get_server_info | MySQL サーバーのバージョンを返す | △ | |
| 27 | mysqli::\$server_version | mysqli_get_server_version | MySQL サーバーのバージョンを整数値で返す | △ | |
| 28 | mysqli::get_warnings | mysqli_get_warnings | SHOW WARNINGS の結果を取得する | △ | |
| 29 | mysqli::\$info | mysqli_info | 直近に実行されたクエリの情報を取得する | △ | |
| 30 | mysqli::init | mysqli_init | MySQLi を初期化し、mysqli_real_connect() で使用するリソースを返す | ○ | <code>\$db->init();</code> |
| 31 | mysqli::\$insert_id | mysqli_insert_id | 直近のクエリで使用した自動生成の ID を返す AUTO INCREMENT 属性を持つカラムがあるテーブル上でのクエリにより生成された ID を返します。 | | |
| 32 | mysqli::kill | mysqli_kill | サーバーに MySQL スレッドの停止を問い合わせる | △ | |
| 33 | mysqli::more_results | mysqli_more_results | マルチクエリからの結果がまだ残っているかどうかを調べる | | |
| 34 | mysqli::multi_query | mysqli_multi_query | データベース上でクエリを実行する。セミコロンで連結されたひとつまたは複数のクエリを実行します。 | | |

| | | | | | |
|----|----------------------------------|---------------------------------|--|---|---|
| 35 | mysqli::next_result | mysqli_next_result | 直近の mysqli_multi_query() コールから次の結果セットを用意します。これは mysqli_store_result() あるいは mysqli_use_result() で取得することが可能です。 | | |
| 36 | mysqli::options | mysqli_options | オプションを設定する | ○ | //MySQLサーバー接続後に実行するコマンド:オートコミットなし \$db->options(MYSQLI_INIT_COMMAND, 'SET AUTOCOMMIT = 0'); \$db->options(MYSQLI_OPT_CONNECT_TIMEOUT, 5); //タイムアウト秒数 |
| 37 | mysqli::ping | mysqli_ping | サーバーとの接続をチェックし、もし切断されている場合は再接続を試みる | △ | |
| 38 | mysqli::poll | mysqli_poll | 接続を問い合わせる(PHP5.3以降) | △ | |
| 39 | mysqli::prepare | mysqli_prepare | SQL クエリを準備し、後でそのステートメントを操作するために使用するステートメントハンドルを返します。クエリは、単一の SQL 文である必要があります。 | ◎ | \$sth = \$db->prepare('SELECT * FROM persons'); //クエリ準備 |
| 40 | mysqli::query | mysqli_query | データベース上でクエリを実行する。SELECT文等が成功した場合は、mysqli_result オブジェクトを返す。MYSQLI_ASYNC (mysqlind で使用可能) を使用すると、クエリを非同期実行できるようになり、クエリの結果を取得するには mysqli_poll() を使用。(PHP5.3以降: 非同期クエリ機能が追加) | ◎ | \$result = \$db->query('SELECT * FROM persons'); //クエリ実行 echo \$result->num_rows; |
| 41 | mysqli::real_connect | mysqli_real_connect | mysql サーバーとの接続をオープンする。mysqli_init() が作成した接続オブジェクトを必要とし、mysqli_options() 関数を使用して、さまざまな接続オプションを設定することが可能 | ○ | \$db = mysqli_init(); \$db->options(MYSQLI_INIT_COMMAND, 'SET AUTOCOMMIT = 0'); { \$db->real_connect('localhost', 'duser', 'password', 'dbname') { die('Connect Error :'.mysqli_connect_error()); } |
| 42 | mysqli::real_escape_string | mysqli_real_escape_string | 接続の現在の文字セットを考慮して、SQL 文で使用する文字列の特殊文字をエスケープする | ○ | \$value = \$db->real_escape_string(\$_GET['value']); |
| 43 | mysqli::real_query | mysqli_real_query | データベースに対して単一のクエリを実行します。その結果を取得したり保存したりするには、関数 mysqli_store_result() あるいは mysqli_use_result() を使用します。 | △ | |
| 44 | mysqli::reap_async_query | mysqli_reap_async_query | 非同期クエリから結果を取得する(PHP5.3以降) | △ | |
| 45 | mysqli::refresh | mysqli_refresh | テーブルやキャッシュのフラッシュ、あるいはレプリケーションサーバー情報のリセットを行います。(PHP5.3以降) | △ | |
| 46 | mysqli::release_savepoint | mysqli_release_savepoint | 現在のトランザクションのセーブポイント群から、指定した名前のセーブポイントを削除する(PHP5.5以降) | × | |
| 47 | mysqli::rollback | mysqli_rollback | 現在のトランザクションをロールバックする | ○ | \$db->rollback(); |
| 48 | mysqli::rpl_query_type | mysqli_rpl_query_type | RPL クエリの型を返す(PHP5.3で削除) | × | |
| 49 | mysqli::savepoint | mysqli_savepoint | トランザクションのセーブポイントを設定する(PHP5.5以降) | × | |
| 50 | mysqli::select_db | mysqli_select_db | クエリを実行するためのデフォルトのデータベースを選択する | △ | |
| 51 | mysqli::send_query | mysqli_send_query | クエリを送信する(PHP5.3で削除) | × | |
| 52 | mysqli::set_charset | mysqli_set_charset | クライアントのデフォルト文字セットを設定する | △ | |
| 53 | mysqli::set_local_infile_default | mysqli_set_local_infile_default | load local infile コマンド用のユーザー定義ハンドラを削除する | △ | |
| 54 | mysqli::set_local_infile_handler | mysqli_set_local_infile_handler | LOAD DATA LOCAL INFILE コマンド用のコールバック関数を設定する | △ | |
| 55 | mysqli::\$sqlstate | mysqli_sqlstate | 直前の MySQL の操作での SQLSTATE エラーを返す | △ | |
| 56 | mysqli::ssl_set | mysqli_ssl_set | SSL を使用したセキュアな接続を確立する | △ | |
| 57 | mysqli::stat | mysqli_stat | 現在のシステム状態を取得する | △ | |
| 58 | mysqli::stmt_init | mysqli_stmt_init | ステートメントを初期化し、mysqli_stmt_prepare で使用するオブジェクトを返す | △ | \$sth = \$db->stmt_init(); |
| 59 | mysqli::store_result | mysqli_store_result | 直近のクエリから結果セットを転送する | | |
| 60 | mysqli::\$thread_id | mysqli_thread_id | 現在の接続のスレッド ID を返す。この ID を使用すると、mysqli_kill() 関数でセッションを切断することが可能です。接続が切断し mysqli_ping() で再接続した場合は、スレッド ID は別のものになります。 | △ | |
| 61 | mysqli::thread_safe | mysqli_thread_safe | スレッドセーフであるかどうかを返す | △ | |

| | | | | | |
|------------------------|------------------------------|---------------------------|--|---|--|
| 62 | mysqli::use_result | mysqli_use_result | mysqli_real_query() 関数を使用して実行した直近のクエリから、結果セットの取得を開始します | | |
| 63 | mysqli::\$warning_count | mysqli_warning_count | 指定した接続の直近のクエリから発生した警告の数を返す | △ | |
| mysqli_stmt クラス | | | | | |
| 64 | mysqli_stmt::\$affected_rows | mysqli_stmt_affected_rows | 直近に実行されたステートメントで変更・削除・あるいは追加された行の総数を返す。テーブルを更新するクエリに対してのみ働きます。SELECT クエリが返す行の数を知るには、mysqli_stmt_num_rows() 関数を代わりに使用します。 | △ | |
| 65 | mysqli_stmt::attr_get | mysqli_stmt_attr_get | ステートメントの属性の現在の値を取得する | △ | |
| 66 | mysqli_stmt::attr_set | mysqli_stmt_attr_set | プリペアドステートメントの振る舞いを変更する。複数の属性を設定するには、この関数を複数回コールします。 | △ | |
| 67 | mysqli_stmt::bind_param | mysqli_stmt_bind_param | プリペアドステートメントのパラメータに変数をバインドする。この変数は、mysqli_prepare() に渡されたもので、変数とパラメータの数が一致のこと。ひとつ又は複数の文字で、対応するバインド変数の型を指定→ i :変数の型はinteger、 d :変数の型はdouble、 s :変数の型はstring、 b :変数の型はblob ※動的なバインド項目(可変な項目数)を作成するためには、 call_user_func_array(array(\$stmt,"bind_param"), \$param_ary); を利用 | ○ | <pre>\$stmt = \$mysqli->prepare("INSERT INTO CountryLanguage VALUES(?, ?, ?)"); \$stmt->bind_param('ssd', \$code, \$language, \$percent);</pre> |
| 68 | mysqli_stmt::bind_result | mysqli_stmt_bind_result | 結果を保存するため、プリペアドステートメントに変数をバインドする。結果セットのカラムを変数にバインドします。 | ○ | <pre>\$sth = \$db->prepare('SELECT name, birth FROM persons'); //クエリ準備 \$sth->execute(); //クエリを実行 \$sth->bind_result(\$name, \$birth); //変数をバインド while (\$sth->fetch()) { echo \$name, ':', \$birth; }</pre> |
| 69 | mysqli_stmt::close | mysqli_stmt_close | プリペアドステートメントを閉じる。stmt が指すステートメントハンドルを開放します。 | ○ | <pre>\$sth->close();</pre> |
| 70 | mysqli_stmt::__construct | | Constructs a new mysqli_stmt object | △ | |
| 71 | mysqli_stmt::data_seek | mysqli_stmt_data_seek | ステートメントの結果セットの任意の行に移動する | △ | |
| 72 | mysqli_stmt::\$errno | mysqli_stmt_errno | 直近のステートメントのコールに関するエラーコードを返す | ○ | <pre>echo \$sth->errno;</pre> |
| 73 | mysqli_stmt::\$error_list | mysqli_stmt_error_list | 直近で実行したステートメントからのエラーの一覧を返す(PHP5.4以降) | △ | <pre>print_r(\$sth->error_list);</pre> |
| 74 | mysqli_stmt::\$error | mysqli_stmt_error | 直近のステートメントのエラー内容を文字列で返す | △ | <pre>echo \$sth->error;</pre> |
| 75 | mysqli_stmt::execute | mysqli_stmt_execute | 事前に mysqli_prepare() 関数で用意されたクエリを実行します。パラメータマーカーが存在する場合、その内容は自動的に適切なデータに置き換えられます。 | ◎ | <pre>\$sth = \$db->prepare('SELECT * FROM persons'); //クエリ準備 \$sth->execute(); //クエリを実行</pre> |
| 76 | mysqli_stmt::fetch | mysqli_stmt_fetch | プリペアドステートメントから結果を読み込み、mysqli_stmt_bind_result() でバインドした変数に格納します。 | ◎ | <pre>\$sth = \$db->prepare('SELECT name, birth FROM persons'); //クエリ準備 \$sth->execute(); //クエリを実行 \$sth->bind_result(\$name, \$birth); //変数をバインド while (\$sth->fetch()) { echo \$name, ':', \$birth; }</pre> |
| 77 | mysqli_stmt::\$field_count | mysqli_stmt_field_count | 指定したステートメントのフィールド数を返す | △ | <pre>echo \$sth->field_count;</pre> |
| 78 | mysqli_stmt::free_result | mysqli_stmt_free_result | 指定したステートメントハンドルの結果を格納したメモリを開放する。このメモリは mysqli_stmt_store_result() によって割り当てられたものです。 | △ | <pre>\$sth = \$db->prepare('SELECT * FROM persons'); //クエリ準備 \$sth->execute(); //クエリを実行 \$sth->store_result(); //結果を保存 \$sth->free_result(); //メモリを開放</pre> |
| 79 | mysqli_stmt::get_result | mysqli_stmt_get_result | resultクラスのオブジェクト生成 プリペアド・ステートメントから結果セットを取得する。 結果セットをレコード単位の処理(fetch_array,fetch_assoc,fetch_row)が可能 (PHP5.3以降) | ○ | <pre>\$sth = \$db->prepare(\$sql); \$sth->execute(); \$res = \$sth->get_result(); echo '件数: ', \$res->num_rows; while (\$ary = \$res->fetch_assoc()) { print_r(\$ary); }</pre> |
| 80 | mysqli_stmt::get_warnings | mysqli_stmt_get_warnings | SHOW WARNINGS の結果を取得する | △ | |
| 81 | mysqli_stmt::\$insert_id | mysqli_stmt_insert_id | 直近の INSERT 操作で生成した ID を取得する | △ | |
| 82 | mysqli_stmt::more_results | mysqli_stmt_more_results | 複数クエリで、次の結果が存在するかどうかを調べる(PHP5.3以降) mysqli_stmt::next_result() のみ使用可能です。 | △ | |
| 83 | mysqli_stmt::next_result | mysqli_stmt_next_result | 複数クエリから、次の結果を読み込む(PHP5.3以降) mysqli_stmt::next_result() のみ使用可能です。 | △ | |

| | | | | | |
|--------------------------|-----------------------------------|-----------------------------|---|---|--|
| 84 | mysqli_stmt::\$num_rows | mysqli_stmt_num_rows | ステートメントの結果セットの行数を返す。mysqli_stmt_store_result() を使用した場合は、すぐに mysqli_stmt_num_rows() をコールできます。 | ○ | <pre>\$sth = \$db->prepare(' SELECT * FROM persons '); //クエリ準備 \$sth->execute(); //クエリを実行 \$sth->store_result(); //結果を保存 echo \$sth->num_rows; //結果の行数</pre> |
| 85 | mysqli_stmt::\$param_count | mysqli_stmt_param_count | 指定したステートメントのパラメータ数を返す。プリペアドステートメント内のパラメーターマーカーの数を返します。 | ○ | |
| 86 | mysqli_stmt::prepare | mysqli_stmt_prepare | SQL ステートメントを実行するために準備する | ◎ | <pre>//プリペアドステートメントを作成 \$sth = \$db->prepare("INSERT INTO city VALUES(?, ?, ?)"); \$sth->bind_param(' iss', \$id, \$code, \$name);</pre> |
| 87 | mysqli_stmt::reset | mysqli_stmt_reset | クライアントおよびサーバーで、実行後のプリペアドステートメントをリセットする | ○ | <pre>\$sth->reset();</pre> |
| 88 | mysqli_stmt::result_metadata | mysqli_stmt_result_metadata | プリペアドステートメントから結果セットのメタデータを返す mysqli_prepare() に渡したステートメントが 結果セットを返すものであった場合、mysqli_stmt_result_metadata() はオブジェクトを返します。このオブジェクトは、結果のフィールド数や 各フィールドの情報などのメタ情報を取得するために使用可能です。 | △ | |
| 89 | mysqli_stmt::send_long_data | mysqli_stmt_send_long_data | データをブロックで送信する。パラメータのデータを、サーバーに分割して送信します。例えば blob のサイズが max_allowed_packet を越えてしまう場合などに使用します。この関数は、カラムに文字やバイナリのデータを送信するために複数回 コールすることが可能です。そのカラムの型は TEXT あるいは BLOB である 必要があります。 | △ | |
| 90 | mysqli_stmt::\$sqlstate | mysqli_stmt_sqlstate | 直近に実行されたプリペアドステートメントのエラーについて、SQLSTATE エラーコードを含む文字列を返します。エラーコードは 5 つの文字で構成されています。'00000' はエラーが発生しなかったことを意味します。 | △ | |
| 91 | mysqli_stmt::store_result | mysqli_stmt_store_result | プリペアドステートメントから結果を転送する。結果セットをクライアントのバッファに格納するだけであり、データを取得するには続けて mysqli_stmt_fetch() をコールします。 mysqli_stmt_store_result() をコールするとバッファを使用し、パフォーマンスが減少します | ○ | <pre>\$sth = \$db->prepare(' SELECT * FROM persons '); //クエリ準備 \$sth->execute(); //クエリを実行 \$sth->bind_result(\$ary[0], ...); //変数をバインド \$sth->store_result(); //結果を保存 while (\$sth->fetch()) { print_r(\$ary); }</pre> |
| mysqli_result クラス | | | | | |
| 92 | mysqli_result::\$current_field | mysqli_field_tell | 直近の mysqli_fetch_field() のコールで使用した フィールドカーソルの位置を返します。この値は mysqli_field_seek() の引数として使用可能です。 | | |
| 93 | mysqli_result::data_seek | mysqli_data_seek | 結果セットの任意の行 offset にポインタを移動します。 | | |
| 94 | mysqli_result::fetch_all | mysqli_fetch_all | 結果のすべての行を連想配列・数値添字配列あるいはその両方の形式で取得(すべての行を配列にまとめて一括で返す) (PHP5.3以降) | △ | <pre>\$ary = \$result->fetch_all(MYSQLI_BOTH); //連想と添字配列両方</pre> |
| 95 | mysqli_result::fetch_array | mysqli_fetch_array | 結果の行を 連想配列・数値添字配列あるいはその両方 の形式で取得する。指定可能な値は 定数 MYSQLI_ASSOC :mysqli_fetch_assoc() と同じ、 MYSQLI_NUM : mysqli_fetch_row()と同じ、 MYSQLI_BOTH : 両方の属性のいずれかです。 | ◎ | <pre>\$ary = \$result->fetch_array(MYSQLI_ASSOC); //連想配列 \$ary = \$result->fetch_array(MYSQLI_NUM); //添字配列 \$ary = \$result->fetch_array(MYSQLI_BOTH); //連想と添字配列両方</pre> |
| 96 | mysqli_result::fetch_assoc | mysqli_fetch_assoc | 取得した行に対応する 連想配列 を返します。もしも行がない場合には NULL を返します。 | ◎ | <pre>while (\$ary = \$result->fetch_assoc()) { print_r(\$ary); } \$result->free(); //結果セットを開放</pre> |
| 97 | mysqli_result::fetch_field_direct | mysqli_fetch_field_direct | 単一のフィールドのメタデータを取得する。指定した結果セットから、フィールド定義情報を含むオブジェクトを返します。 | | |
| 98 | mysqli_result::fetch_field | mysqli_fetch_field | 結果セットの次のフィールドを返す。結果セットから ひとつのカラムの情報をオブジェクトとして返します。この関数を 繰り返しコールすることで、結果セットのすべてのカラムについての情報が 取得可能です。 | | |
| 99 | mysqli_result::fetch_fields | mysqli_fetch_fields | 結果セットのフィールド情報をオブジェクトの配列で返す。複数のカラムの情報をオブジェクトの配列で返します。 | | |
| 100 | mysqli_result::fetch_object | mysqli_fetch_object | 結果セットの現在の行をオブジェクトとして返す。このオブジェクトは、結果セットのフィールド名を 属性として持ちます。 | | |
| 101 | mysqli_result::fetch_row | mysqli_fetch_row | 結果セットからデータを 1 行取得し、それを 数値添字の配列 で返します。各カラムは 0(ゼロ)から始まる添字に格納されます。mysqli_fetch_row() 関数を続けてコールすると、結果セットの次の行を返します。もしも行がない場合には NULL を返します。 | ◎ | <pre>while (\$ary = \$result->fetch_row()) { print_r(\$ary); } \$result->free(); //結果セットを開放</pre> |
| 102 | mysqli_result::\$field_count | mysqli_num_fields | 指定した結果セットからフィールドの数を返します。 | | |
| 103 | mysqli_result::field_seek | mysqli_field_seek | フィールドカーソルを指定したオフセットに設定します。mysqli_fetch_field() を次にコールした際に、設定したオフセットに関連するカラムのフィールド定義を取得します。 | | |

| | | | | | |
|--------------------------|--------------------------------------|------------------------------|--|---|---|
| 104 | mysqli_result::free | mysqli_free_result | 結果に関連付けられたメモリを開放する(結果オブジェクトがなくなつた場合は、常に mysqli_free_result() でメモリを開放すべき) | ◎ | \$result->free(); //結果セットを開放 |
| 105 | mysqli_result::\$lengths | mysqli_fetch_lengths | 結果セットにおける現在の行の列の長さを返す。result が指す結果セットの現在の行について、すべての列の長さを含む 配列を返しま | | |
| 106 | mysqli_result::\$num_rows | mysqli_num_rows | 結果の行数を取得する。 mysqli_stmt_store_result()を用いて結果をステートメントハンドルにバッファリングしている必要あり。 | ○ | \$result = \$db->query("SELECT * FROM persons WHERE id = \$id"); \$cnt_nums = \$result->num_rows; //クエリ結果の行数 \$result->free(); //結果セットを開放 |
| mysqli_driver クラス | | | | | |
| 107 | mysqli_driver::embedded_server_end | mysqli_embedded_server_end | 組み込みサーバーを停止する | | |
| 108 | mysqli_driver::embedded_server_start | mysqli_embedded_server_start | 組み込みサーバーを初期化して開始する | | |
| 109 | mysqli_driver::\$report_mode | mysqli_report | 内部のレポート機能を有効あるいは無効にする | | |

【mysqli_optionsの利用可能なオプション】

| | | |
|---------------------------------|--|--|
| MYSQLI_OPT_CONNECT_TIMEOUT | | 接続のタイムアウト秒数 (PHP 5.3.1 以降、Windows 上での TCP/IP 接続をサポートします)。 |
| MYSQLI_OPT_LOCAL_INFILE | | LOAD LOCAL INFILE の使用可/不可。 |
| MYSQLI_INIT_COMMAND | | MySQL サーバーへの接続後に実行するコマンド。 |
| MYSQLI_READ_DEFAULT_FILE | | my.cnf の代わりに、指定した名前のファイルから 設定を読み込みます。 |
| MYSQLI_READ_DEFAULT_GROUP | | my.cnf の指定した名前のグループ、あるいは MYSQLI_READ_DEFAULT_FILE で指定したファイルから 設定を読み込みます。 |
| MYSQLI_SERVER_PUBLIC_KEY | | RSA 公開鍵ファイル。SHA-256 ベースの認証で使います。 |
| MYSQLI_OPT_NET_CMD_BUFFER_SIZE | | 内部のコマンド/ネットワークバッファのサイズ。mysqlnd でのみ有効です。 |
| MYSQLI_OPT_NET_READ_BUFFER_SIZE | | MySQL コマンド/パケットのボディを読み込むときの、読み込みチャンクの最大バイト数。mysqlnd でのみ有効です。 |

【mysqli_stmt_attr_setの属性値】

| | | |
|------------------------------------|--|---|
| MYSQLI_STMT_ATTR_UPDATE_MAX_LENGTH | | TRUE に設定すると、mysqli_stmt_store_result() は メタデータ MYSQL_FIELD->max_length の値を上書きします。 |
| MYSQLI_STMT_ATTR_CURSOR_TYPE | | mysqli_stmt_execute() が起動された際にステートメントをオープンするためのカーソル型。mode は、MYSQLI_CURSOR_TYPE_NO_CURSOR (デフォルト) あるいは MYSQLI_CURSOR_TYPE_READ_ONLY となります。 |
| MYSQLI_STMT_ATTR_PREFETCH_ROWS | | カーソルの使用時にサーバーからいちどに取得する行数。mode に指定できる値の範囲は 1 から unsigned long の最大値までで、デフォルトは 1 です。 |

※動的なバインド項目(可変な項目数)を作成する

```
$param_ary = []; // stmp_bind_param()関数の引数用
// 対応するバインド変数の型を連想配列($ary)から作成
$param_ary[0] = BindParamMake($ary); // BindParamMakeはユーザー関数
// 可変変数項目の配列作成
foreach($ary as $key=>$value){
    $param_ary[$key] = &$ary[$key]; // 参照渡しで、可変項目の配列に追加
}
call_user_func_array ( array($stmt,"bind_param"), $param_ary );
```