* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/zip.html)
* [Database Reference](http://docs.google.com/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/examples.html)
  + [Database Configuration](http://docs.google.com/configuration.html)
  + [Connecting to a Database](http://docs.google.com/connecting.html)
  + [Running Queries](http://docs.google.com/queries.html)
  + [Generating Query Results](http://docs.google.com/results.html)
  + [Query Helper Functions](http://docs.google.com/helpers.html)
  + [Query Builder Class](http://docs.google.com/query_builder.html)
  + [Transactions](http://docs.google.com/transactions.html)
  + [Getting MetaData](http://docs.google.com/metadata.html)
  + [Custom Function Calls](http://docs.google.com/call_function.html)
  + [Query Caching](http://docs.google.com/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/forge.html)
  + [Database Utilities Class](http://docs.google.com/utilities.html)
  + [Database Driver Reference](#gjdgxs)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

Toggle Table of Contents

[CodeIgniter](http://docs.google.com/index.html)

* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/zip.html)
* [Database Reference](http://docs.google.com/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/examples.html)
  + [Database Configuration](http://docs.google.com/configuration.html)
  + [Connecting to a Database](http://docs.google.com/connecting.html)
  + [Running Queries](http://docs.google.com/queries.html)
  + [Generating Query Results](http://docs.google.com/results.html)
  + [Query Helper Functions](http://docs.google.com/helpers.html)
  + [Query Builder Class](http://docs.google.com/query_builder.html)
  + [Transactions](http://docs.google.com/transactions.html)
  + [Getting MetaData](http://docs.google.com/metadata.html)
  + [Custom Function Calls](http://docs.google.com/call_function.html)
  + [Query Caching](http://docs.google.com/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/forge.html)
  + [Database Utilities Class](http://docs.google.com/utilities.html)
  + [Database Driver Reference](#gjdgxs)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

[CodeIgniter](http://docs.google.com/index.html)

* [Docs](http://docs.google.com/index.html) »
* [Database Reference](http://docs.google.com/index.html) »
* DB Driver Reference

classic layout

DB Driver Reference[**¶**](#30j0zll)

This is the platform-independent base DB implementation class. This class will not be called directly. Rather, the adapter class for the specific database will extend and instantiate it.

The how-to material for this has been split over several articles. This article is intended to be a reference for them.

Important

Not all methods are supported by all database drivers, some of them may fail (and return FALSE) if the underlying driver does not support them.

*class* CI\_DB\_driver[¶](#1fob9te) initialize()[¶](#3znysh7)

|  |  |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Initialize database settings, establish a connection to the database.

db\_connect(*$persistent = TRUE*)[¶](#2et92p0)

|  |  |
| --- | --- |
| Parameters: | * **$persistent** (*bool*) – Whether to establish a persistent connection or a regular one |
| Returns: | Database connection resource/object or FALSE on failure |
| Return type: | mixed |

Establish a connection with the database.

Note

The returned value depends on the underlying driver in use. For example, a mysqli instance will be returned with the ‘mysqli’ driver.

db\_pconnect()[¶](#tyjcwt)

|  |  |
| --- | --- |
| Returns: | Database connection resource/object or FALSE on failure |
| Return type: | mixed |

Establish a persistent connection with the database.

Note

This method is just an alias for db\_connect(TRUE).

reconnect()[¶](#3dy6vkm)

|  |  |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Keep / reestablish the database connection if no queries have been sent for a length of time exceeding the server’s idle timeout.

db\_select([*$database = ''*])[¶](#1t3h5sf)

|  |  |
| --- | --- |
| Parameters: | * **$database** (*string*) – Database name |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Select / switch the current database.

db\_set\_charset(*$charset*)[¶](#4d34og8)

|  |  |
| --- | --- |
| Parameters: | * **$charset** (*string*) – Character set name |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Set client character set.

platform()[¶](#2s8eyo1)

|  |  |
| --- | --- |
| Returns: | Platform name |
| Return type: | string |

The name of the platform in use (mysql, mssql, etc…).

version()[¶](#17dp8vu)

|  |  |
| --- | --- |
| Returns: | The version of the database being used |
| Return type: | string |

Database version number.

query(*$sql*[, *$binds = FALSE*[, *$return\_object = NULL*]])[¶](#3rdcrjn)

|  |  |
| --- | --- |
| Parameters: | * **$sql** (*string*) – The SQL statement to execute * **$binds** (*array*) – An array of binding data * **$return\_object** (*bool*) – Whether to return a result object or not |
| Returns: | TRUE for successful “write-type” queries, CI\_DB\_result instance (method chaining) on “query” success, FALSE on failure |
| Return type: | mixed |

Execute an SQL query.

Accepts an SQL string as input and returns a result object upon successful execution of a “read” type query.

Returns:

* Boolean TRUE upon successful execution of a “write type” queries
* Boolean FALSE upon failure
* CI\_DB\_result object for “read type” queries

simple\_query(*$sql*)[¶](#26in1rg)

|  |  |
| --- | --- |
| Parameters: | * **$sql** (*string*) – The SQL statement to execute |
| Returns: | Whatever the underlying driver’s “query” function returns |
| Return type: | mixed |

A simplified version of the query() method, appropriate for use when you don’t need to get a result object or to just send a query to the database and not care for the result.

affected\_rows()[¶](#lnxbz9)

|  |  |
| --- | --- |
| Returns: | Number of rows affected |
| Return type: | int |

Returns the number of rows *changed* by the last executed query.

Useful for checking how much rows were created, updated or deleted during the last executed query.

trans\_strict([*$mode = TRUE*])[¶](#35nkun2)

|  |  |
| --- | --- |
| Parameters: | * **$mode** (*bool*) – Strict mode flag |
| Return type: | void |

Enable/disable transaction “strict” mode.

When strict mode is enabled, if you are running multiple groups of transactions and one group fails, all subsequent groups will be rolled back.

If strict mode is disabled, each group is treated autonomously, meaning a failure of one group will not affect any others.

trans\_off()[¶](#1ksv4uv)

|  |  |
| --- | --- |
| Return type: | void |

Disables transactions at run-time.

trans\_start([*$test\_mode = FALSE*])[¶](#44sinio)

|  |  |
| --- | --- |
| Parameters: | * **$test\_mode** (*bool*) – Test mode flag |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Start a transaction.

trans\_complete()[¶](#2jxsxqh)

|  |  |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Complete Transaction.

trans\_status()[¶](#z337ya)

|  |  |
| --- | --- |
| Returns: | TRUE if the transaction succeeded, FALSE if it failed |
| Return type: | bool |

Lets you retrieve the transaction status flag to determine if it has failed.

compile\_binds(*$sql*, *$binds*)[¶](#3j2qqm3)

|  |  |
| --- | --- |
| Parameters: | * **$sql** (*string*) – The SQL statement * **$binds** (*array*) – An array of binding data |
| Returns: | The updated SQL statement |
| Return type: | string |

Compiles an SQL query with the bind values passed for it.

is\_write\_type(*$sql*)[¶](#1y810tw)

|  |  |
| --- | --- |
| Parameters: | * **$sql** (*string*) – The SQL statement |
| Returns: | TRUE if the SQL statement is of “write type”, FALSE if not |
| Return type: | bool |

Determines if a query is of a “write” type (such as INSERT, UPDATE, DELETE) or “read” type (i.e. SELECT).

elapsed\_time([*$decimals = 6*])[¶](#4i7ojhp)

|  |  |
| --- | --- |
| Parameters: | * **$decimals** (*int*) – The number of decimal places |
| Returns: | The aggregate query elapsed time, in microseconds |
| Return type: | string |

Calculate the aggregate query elapsed time.

total\_queries()[¶](#2xcytpi)

|  |  |
| --- | --- |
| Returns: | The total number of queries executed |
| Return type: | int |

Returns the total number of queries that have been executed so far.

last\_query()[¶](#1ci93xb)

|  |  |
| --- | --- |
| Returns: | The last query executed |
| Return type: | string |

Returns the last query that was executed.

escape(*$str*)[¶](#3whwml4)

|  |  |
| --- | --- |
| Parameters: | * **$str** (*mixed*) – The value to escape, or an array of multiple ones |
| Returns: | The escaped value(s) |
| Return type: | mixed |

Escapes input data based on type, including boolean and NULLs.

escape\_str(*$str*[, *$like = FALSE*])[¶](#2bn6wsx)

|  |  |
| --- | --- |
| Parameters: | * **$str** (*mixed*) – A string value or array of multiple ones * **$like** (*bool*) – Whether or not the string will be used in a LIKE condition |
| Returns: | The escaped string(s) |
| Return type: | mixed |

Escapes string values.

Warning

The returned strings do NOT include quotes around them.

escape\_like\_str(*$str*)[¶](#qsh70q)

|  |  |
| --- | --- |
| Parameters: | * **$str** (*mixed*) – A string value or array of multiple ones |
| Returns: | The escaped string(s) |
| Return type: | mixed |

Escape LIKE strings.

Similar to escape\_str(), but will also escape the % and \_ wildcard characters, so that they don’t cause false-positives in LIKE conditions.

Important

The escape\_like\_str() method uses ‘!’ (exclamation mark) to escape special characters for *LIKE* conditions. Because this method escapes partial strings that you would wrap in quotes yourself, it cannot automatically add the ESCAPE '!' condition for you, and so you’ll have to manually do that.

primary(*$table*)[¶](#3as4poj)

|  |  |
| --- | --- |
| Parameters: | * **$table** (*string*) – Table name |
| Returns: | The primary key name, FALSE if none |
| Return type: | string |

Retrieves the primary key of a table.

Note

If the database platform does not support primary key detection, the first column name may be assumed as the primary key.

count\_all([*$table = ''*])[¶](#1pxezwc)

|  |  |
| --- | --- |
| Parameters: | * **$table** (*string*) – Table name |
| Returns: | Row count for the specified table |
| Return type: | int |

Returns the total number of rows in a table, or 0 if no table was provided.

list\_tables([*$constrain\_by\_prefix = FALSE*])[¶](#49x2ik5)

|  |  |
| --- | --- |
| Parameters: | * **$constrain\_by\_prefix** (*bool*) – TRUE to match table names by the configured dbprefix |
| Returns: | Array of table names or FALSE on failure |
| Return type: | array |

Gets a list of the tables in the current database.

table\_exists(*$table\_name*)[¶](#2p2csry)

|  |  |
| --- | --- |
| Parameters: | * **$table\_name** (*string*) – The table name |
| Returns: | TRUE if that table exists, FALSE if not |
| Return type: | bool |

Determine if a particular table exists.

list\_fields(*$table*)[¶](#147n2zr)

|  |  |
| --- | --- |
| Parameters: | * **$table** (*string*) – The table name |
| Returns: | Array of field names or FALSE on failure |
| Return type: | array |

Gets a list of the field names in a table.

field\_exists(*$field\_name*, *$table\_name*)[¶](#3o7alnk)

|  |  |
| --- | --- |
| Parameters: | * **$table\_name** (*string*) – The table name * **$field\_name** (*string*) – The field name |
| Returns: | TRUE if that field exists in that table, FALSE if not |
| Return type: | bool |

Determine if a particular field exists.

field\_data(*$table*)[¶](#23ckvvd)

|  |  |
| --- | --- |
| Parameters: | * **$table** (*string*) – The table name |
| Returns: | Array of field data items or FALSE on failure |
| Return type: | array |

Gets a list containing field data about a table.

escape\_identifiers(*$item*)[¶](#ihv636)

|  |  |
| --- | --- |
| Parameters: | * **$item** (*mixed*) – The item or array of items to escape |
| Returns: | The input item(s), escaped |
| Return type: | mixed |

Escape SQL identifiers, such as column, table and names.

insert\_string(*$table*, *$data*)[¶](#32hioqz)

|  |  |
| --- | --- |
| Parameters: | * **$table** (*string*) – The target table * **$data** (*array*) – An associative array of key/value pairs |
| Returns: | The SQL INSERT statement, as a string |
| Return type: | string |

Generate an INSERT statement string.

update\_string(*$table*, *$data*, *$where*)[¶](#1hmsyys)

|  |  |
| --- | --- |
| Parameters: | * **$table** (*string*) – The target table * **$data** (*array*) – An associative array of key/value pairs * **$where** (*mixed*) – The WHERE statement conditions |
| Returns: | The SQL UPDATE statement, as a string |
| Return type: | string |

Generate an UPDATE statement string.

call\_function(*$function*)[¶](#41mghml)

|  |  |
| --- | --- |
| Parameters: | * **$function** (*string*) – Function name |
| Returns: | The function result |
| Return type: | string |

Runs a native PHP function , using a platform agnostic wrapper.

cache\_set\_path([*$path = ''*])[¶](#2grqrue)

|  |  |
| --- | --- |
| Parameters: | * **$path** (*string*) – Path to the cache directory |
| Return type: | void |

Sets the directory path to use for caching storage.

cache\_on()[¶](#vx1227)

|  |  |
| --- | --- |
| Returns: | TRUE if caching is on, FALSE if not |
| Return type: | bool |

Enable database results caching.

cache\_off()[¶](#3fwokq0)

|  |  |
| --- | --- |
| Returns: | TRUE if caching is on, FALSE if not |
| Return type: | bool |

Disable database results caching.

cache\_delete([*$segment\_one = ''*[, *$segment\_two = ''*]])[¶](#1v1yuxt)

|  |  |
| --- | --- |
| Parameters: | * **$segment\_one** (*string*) – First URI segment * **$segment\_two** (*string*) – Second URI segment |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Delete the cache files associated with a particular URI.

cache\_delete\_all()[¶](#4f1mdlm)

|  |  |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Delete all cache files.

close()[¶](#2u6wntf)

|  |  |
| --- | --- |
| Return type: | void |

Close the DB Connection.

display\_error([*$error = ''*[, *$swap = ''*[, *$native = FALSE*]]])[¶](#19c6y18)

|  |  |
| --- | --- |
| Parameters: | * **$error** (*string*) – The error message * **$swap** (*string*) – Any “swap” values * **$native** (*bool*) – Whether to localize the message |
| Return type: | void |
| Returns: | Displays the DB error screensends the application/views/errors/error\_db.php template |
| Return type: | string |

Display an error message and stop script execution.

The message is displayed using the *application/views/errors/error\_db.php* template.

protect\_identifiers(*$item*[, *$prefix\_single = FALSE*[, *$protect\_identifiers = NULL*[, *$field\_exists = TRUE*]]])[¶](#3tbugp1)

|  |  |
| --- | --- |
| Parameters: | * **$item** (*string*) – The item to work with * **$prefix\_single** (*bool*) – Whether to apply the dbprefix even if the input item is a single identifier * **$protect\_identifiers** (*bool*) – Whether to quote identifiers * **$field\_exists** (*bool*) – Whether the supplied item contains a field name or not |
| Returns: | The modified item |
| Return type: | string |

Takes a column or table name (optionally with an alias) and applies the configured *dbprefix* to it.

Some logic is necessary in order to deal with column names that include the path.

Consider a query like this:

SELECT \* FROM hostname.database.table.column AS c FROM hostname.database.table

Or a query with aliasing:

SELECT m.member\_id, m.member\_name FROM members AS m

Since the column name can include up to four segments (host, DB, table, column) or also have an alias prefix, we need to do a bit of work to figure this out and insert the table prefix (if it exists) in the proper position, and escape only the correct identifiers.

This method is used extensively by the Query Builder class.

[Next](http://docs.google.com/helpers/index.html)   [Previous](http://docs.google.com/utilities.html)

© Copyright 2014 - 2018, British Columbia Institute of Technology. Last updated on Jun 12, 2018.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/snide/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).