* [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
  + [Database Utilities Class](http://docs.google.com/utilities.html)
  + [Database Driver Reference](http://docs.google.com/db_driver_reference.html)
* [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
  + [Database Utilities Class](http://docs.google.com/utilities.html)
  + [Database Driver Reference](http://docs.google.com/db_driver_reference.html)
* [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) »
* Database Forge Class

classic layout

[**Database Forge Class**](#gjdgxs)[**¶**](#30j0zll)

The Database Forge Class contains methods that help you manage your database.

Table of Contents

* [Database Forge Class](#30j0zll)
  + [Initializing the Forge Class](#1fob9te)
  + [Creating and Dropping Databases](#3znysh7)
  + [Creating and Dropping Tables](#2et92p0)
    - [Adding fields](#tyjcwt)
    - [Adding Keys](#3dy6vkm)
    - [Creating a table](#1t3h5sf)
    - [Dropping a table](#4d34og8)
    - [Renaming a table](#2s8eyo1)
  + [Modifying Tables](#17dp8vu)
    - [Adding a Column to a Table](#3rdcrjn)
    - [Dropping a Column From a Table](#26in1rg)
    - [Modifying a Column in a Table](#lnxbz9)
  + [Class Reference](#35nkun2)

[**Initializing the Forge Class**](#1ksv4uv)[**¶**](#1fob9te)

Important

In order to initialize the Forge class, your database driver must already be running, since the forge class relies on it.

Load the Forge Class as follows:

$this->load->dbforge()

You can also pass another database object to the DB Forge loader, in case the database you want to manage isn’t the default one:

$this->myforge = $this->load->dbforge($this->other\_db, TRUE);

In the above example, we’re passing a custom database object as the first parameter and then tell it to return the dbforge object, instead of assigning it directly to $this->dbforge.

Note

Both of the parameters can be used individually, just pass an empty value as the first one if you wish to skip it.

Once initialized you will access the methods using the $this->dbforge object:

$this->dbforge->some\_method();

[**Creating and Dropping Databases**](#44sinio)[**¶**](#3znysh7)

**$this->dbforge->create\_database(‘db\_name’)**

Permits you to create the database specified in the first parameter. Returns TRUE/FALSE based on success or failure:

if ($this->dbforge->create\_database('my\_db'))  
{  
 echo 'Database created!';  
}

**$this->dbforge->drop\_database(‘db\_name’)**

Permits you to drop the database specified in the first parameter. Returns TRUE/FALSE based on success or failure:

if ($this->dbforge->drop\_database('my\_db'))  
{  
 echo 'Database deleted!';  
}

[**Creating and Dropping Tables**](#2jxsxqh)[**¶**](#2et92p0)

There are several things you may wish to do when creating tables. Add fields, add keys to the table, alter columns. CodeIgniter provides a mechanism for this.

[**Adding fields**](#z337ya)[**¶**](#tyjcwt)

Fields are created via an associative array. Within the array you must include a ‘type’ key that relates to the datatype of the field. For example, INT, VARCHAR, TEXT, etc. Many datatypes (for example VARCHAR) also require a ‘constraint’ key.

$fields = array(  
 'users' => array(  
 'type' => 'VARCHAR',  
 'constraint' => '100',  
 ),  
);  
// will translate to "users VARCHAR(100)" when the field is added.

Additionally, the following key/values can be used:

* unsigned/true : to generate “UNSIGNED” in the field definition.
* default/value : to generate a default value in the field definition.
* null/true : to generate “NULL” in the field definition. Without this, the field will default to “NOT NULL”.
* auto\_increment/true : generates an auto\_increment flag on the field. Note that the field type must be a type that supports this, such as integer.
* unique/true : to generate a unique key for the field definition.

$fields = array(  
 'blog\_id' => array(  
 'type' => 'INT',  
 'constraint' => 5,  
 'unsigned' => TRUE,  
 'auto\_increment' => TRUE  
 ),  
 'blog\_title' => array(  
 'type' => 'VARCHAR',  
 'constraint' => '100',  
 'unique' => TRUE,  
 ),  
 'blog\_author' => array(  
 'type' =>'VARCHAR',  
 'constraint' => '100',  
 'default' => 'King of Town',  
 ),  
 'blog\_description' => array(  
 'type' => 'TEXT',  
 'null' => TRUE,  
 ),  
);

After the fields have been defined, they can be added using $this->dbforge->add\_field($fields); followed by a call to the create\_table() method.

**$this->dbforge->add\_field()**

The add fields method will accept the above array.

Passing strings as fields[**¶**](#3j2qqm3)

If you know exactly how you want a field to be created, you can pass the string into the field definitions with add\_field()

$this->dbforge->add\_field("label varchar(100) NOT NULL DEFAULT 'default label'");

Note

Passing raw strings as fields cannot be followed by add\_key() calls on those fields.

Note

Multiple calls to add\_field() are cumulative.

Creating an id field[**¶**](#1y810tw)

There is a special exception for creating id fields. A field with type id will automatically be assigned as an INT(9) auto\_incrementing Primary Key.

$this->dbforge->add\_field('id');  
// gives id INT(9) NOT NULL AUTO\_INCREMENT

[**Adding Keys**](#4i7ojhp)[**¶**](#3dy6vkm)

Generally speaking, you’ll want your table to have Keys. This is accomplished with $this->dbforge->add\_key(‘field’). An optional second parameter set to TRUE will make it a primary key. Note that add\_key() must be followed by a call to create\_table().

Multiple column non-primary keys must be sent as an array. Sample output below is for MySQL.

$this->dbforge->add\_key('blog\_id', TRUE);  
// gives PRIMARY KEY `blog\_id` (`blog\_id`)  
  
$this->dbforge->add\_key('blog\_id', TRUE);  
$this->dbforge->add\_key('site\_id', TRUE);  
// gives PRIMARY KEY `blog\_id\_site\_id` (`blog\_id`, `site\_id`)  
  
$this->dbforge->add\_key('blog\_name');  
// gives KEY `blog\_name` (`blog\_name`)  
  
$this->dbforge->add\_key(array('blog\_name', 'blog\_label'));  
// gives KEY `blog\_name\_blog\_label` (`blog\_name`, `blog\_label`)

[**Creating a table**](#2xcytpi)[**¶**](#1t3h5sf)

After fields and keys have been declared, you can create a new table with

$this->dbforge->create\_table('table\_name');  
// gives CREATE TABLE table\_name

An optional second parameter set to TRUE adds an “IF NOT EXISTS” clause into the definition

$this->dbforge->create\_table('table\_name', TRUE);  
// gives CREATE TABLE IF NOT EXISTS table\_name

You could also pass optional table attributes, such as MySQL’s ENGINE:

$attributes = array('ENGINE' => 'InnoDB');  
$this->dbforge->create\_table('table\_name', FALSE, $attributes);  
// produces: CREATE TABLE `table\_name` (...) ENGINE = InnoDB DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci

Note

Unless you specify the CHARACTER SET and/or COLLATE attributes, create\_table() will always add them with your configured *char\_set* and *dbcollat* values, as long as they are not empty (MySQL only).

[**Dropping a table**](#1ci93xb)[**¶**](#4d34og8)

Execute a DROP TABLE statement and optionally add an IF EXISTS clause.

// Produces: DROP TABLE table\_name  
$this->dbforge->drop\_table('table\_name');  
  
// Produces: DROP TABLE IF EXISTS table\_name  
$this->dbforge->drop\_table('table\_name',TRUE);

[**Renaming a table**](#3whwml4)[**¶**](#2s8eyo1)

Executes a TABLE rename

$this->dbforge->rename\_table('old\_table\_name', 'new\_table\_name');  
// gives ALTER TABLE old\_table\_name RENAME TO new\_table\_name

[**Modifying Tables**](#2bn6wsx)[**¶**](#17dp8vu)

[**Adding a Column to a Table**](#qsh70q)[**¶**](#3rdcrjn)

**$this->dbforge->add\_column()**

The add\_column() method is used to modify an existing table. It accepts the same field array as above, and can be used for an unlimited number of additional fields.

$fields = array(  
 'preferences' => array('type' => 'TEXT')  
);  
$this->dbforge->add\_column('table\_name', $fields);  
// Executes: ALTER TABLE table\_name ADD preferences TEXT

If you are using MySQL or CUBIRD, then you can take advantage of their AFTER and FIRST clauses to position the new column.

Examples:

// Will place the new column after the `another\_field` column:  
$fields = array(  
 'preferences' => array('type' => 'TEXT', 'after' => 'another\_field')  
);  
  
// Will place the new column at the start of the table definition:  
$fields = array(  
 'preferences' => array('type' => 'TEXT', 'first' => TRUE)  
);

[**Dropping a Column From a Table**](#3as4poj)[**¶**](#26in1rg)

**$this->dbforge->drop\_column()**

Used to remove a column from a table.

$this->dbforge->drop\_column('table\_name', 'column\_to\_drop');

[**Modifying a Column in a Table**](#1pxezwc)[**¶**](#lnxbz9)

**$this->dbforge->modify\_column()**

The usage of this method is identical to add\_column(), except it alters an existing column rather than adding a new one. In order to change the name you can add a “name” key into the field defining array.

$fields = array(  
 'old\_name' => array(  
 'name' => 'new\_name',  
 'type' => 'TEXT',  
 ),  
);  
$this->dbforge->modify\_column('table\_name', $fields);  
// gives ALTER TABLE table\_name CHANGE old\_name new\_name TEXT

[**Class Reference**](#49x2ik5)[**¶**](#35nkun2)

*class* CI\_DB\_forge[¶](#2p2csry) add\_column(*$table*[, *$field = array()*[, *$\_after = NULL*]])[¶](#147n2zr)

| Parameters: | * **$table** (*string*) – Table name to add the column to * **$field** (*array*) – Column definition(s) * **$\_after** (*string*) – Column for AFTER clause (deprecated) |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Adds a column to a table. Usage: See [Adding a Column to a Table](#3rdcrjn).

add\_field(*$field*)[¶](#3o7alnk)

| Parameters: | * **$field** (*array*) – Field definition to add |
| --- | --- |
| Returns: | CI\_DB\_forge instance (method chaining) |
| Return type: | CI\_DB\_forge |

Adds a field to the set that will be used to create a table. Usage: See [Adding fields](#tyjcwt).

add\_key(*$key*[, *$primary = FALSE*])[¶](#23ckvvd)

| Parameters: | * **$key** (*array*) – Name of a key field * **$primary** (*bool*) – Set to TRUE if it should be a primary key or a regular one |
| --- | --- |
| Returns: | CI\_DB\_forge instance (method chaining) |
| Return type: | CI\_DB\_forge |

Adds a key to the set that will be used to create a table. Usage: See [Adding Keys](#3dy6vkm).

create\_database(*$db\_name*)[¶](#ihv636)

| Parameters: | * **$db\_name** (*string*) – Name of the database to create |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Creates a new database. Usage: See [Creating and Dropping Databases](#3znysh7).

create\_table(*$table*[, *$if\_not\_exists = FALSE*[, *array $attributes = array()*]])[¶](#32hioqz)

| Parameters: | * **$table** (*string*) – Name of the table to create * **$if\_not\_exists** (*string*) – Set to TRUE to add an ‘IF NOT EXISTS’ clause * **$attributes** (*string*) – An associative array of table attributes |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Creates a new table. Usage: See [Creating a table](#1t3h5sf).

drop\_column(*$table*, *$column\_name*)[¶](#1hmsyys)

| Parameters: | * **$table** (*string*) – Table name * **$column\_name** (*array*) – The column name to drop |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Drops a column from a table. Usage: See [Dropping a Column From a Table](#26in1rg).

drop\_database(*$db\_name*)[¶](#41mghml)

| Parameters: | * **$db\_name** (*string*) – Name of the database to drop |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Drops a database. Usage: See [Creating and Dropping Databases](#3znysh7).

drop\_table(*$table\_name*[, *$if\_exists = FALSE*])[¶](#2grqrue)

| Parameters: | * **$table** (*string*) – Name of the table to drop * **$if\_exists** (*string*) – Set to TRUE to add an ‘IF EXISTS’ clause |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Drops a table. Usage: See [Dropping a table](#4d34og8).

modify\_column(*$table*, *$field*)[¶](#vx1227)

| Parameters: | * **$table** (*string*) – Table name * **$field** (*array*) – Column definition(s) |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Modifies a table column. Usage: See [Modifying a Column in a Table](#lnxbz9).

rename\_table(*$table\_name*, *$new\_table\_name*)[¶](#3fwokq0)

| Parameters: | * **$table** (*string*) – Current of the table * **$new\_table\_name** (*string*) – New name of the table |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Renames a table. Usage: See [Renaming a table](#2s8eyo1).

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

© Copyright 2014 - 2016, British Columbia Institute of Technology. Last updated on Oct 28, 2016.

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).