* [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
  + [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](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
  + [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](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) »
* Generating Query Results

classic layout

Generating Query Results[**¶**](#gjdgxs)

There are several ways to generate query results:

* [Result Arrays](#30j0zll)
* [Result Rows](#1fob9te)
* [Custom Result Objects](#3znysh7)
* [Result Helper Methods](#2et92p0)
* [Class Reference](#tyjcwt)

[**Result Arrays**](#3dy6vkm)[**¶**](#30j0zll)

**result()**

This method returns the query result as an array of **objects**, or **an empty array** on failure. Typically you’ll use this in a foreach loop, like this:

$query = $this->db->query("YOUR QUERY");  
  
foreach ($query->result() as $row)  
{  
 echo $row->title;  
 echo $row->name;  
 echo $row->body;  
}

The above method is an alias of result\_object().

You can also pass a string to result() which represents a class to instantiate for each result object (note: this class must be loaded)

$query = $this->db->query("SELECT \* FROM users;");  
  
foreach ($query->result('User') as $user)  
{  
 echo $user->name; // access attributes  
 echo $user->reverse\_name(); // or methods defined on the 'User' class  
}

**result\_array()**

This method returns the query result as a pure array, or an empty array when no result is produced. Typically you’ll use this in a foreach loop, like this:

$query = $this->db->query("YOUR QUERY");  
  
foreach ($query->result\_array() as $row)  
{  
 echo $row['title'];  
 echo $row['name'];  
 echo $row['body'];  
}

[**Result Rows**](#1t3h5sf)[**¶**](#1fob9te)

**row()**

This method returns a single result row. If your query has more than one row, it returns only the first row. The result is returned as an **object**. Here’s a usage example:

$query = $this->db->query("YOUR QUERY");  
  
$row = $query->row();  
  
if (isset($row))  
{  
 echo $row->title;  
 echo $row->name;  
 echo $row->body;  
}

If you want a specific row returned you can submit the row number as a digit in the first parameter:

$row = $query->row(5);

You can also add a second String parameter, which is the name of a class to instantiate the row with:

$query = $this->db->query("SELECT \* FROM users LIMIT 1;");  
$row = $query->row(0, 'User');  
  
echo $row->name; // access attributes  
echo $row->reverse\_name(); // or methods defined on the 'User' class

**row\_array()**

Identical to the above row() method, except it returns an array. Example:

$query = $this->db->query("YOUR QUERY");  
  
$row = $query->row\_array();  
  
if (isset($row))  
{  
 echo $row['title'];  
 echo $row['name'];  
 echo $row['body'];  
}

If you want a specific row returned you can submit the row number as a digit in the first parameter:

$row = $query->row\_array(5);

In addition, you can walk forward/backwards/first/last through your results using these variations:

**$row = $query->first\_row()**

**$row = $query->last\_row()**

**$row = $query->next\_row()**

**$row = $query->previous\_row()**

By default they return an object unless you put the word “array” in the parameter:

**$row = $query->first\_row(‘array’)**

**$row = $query->last\_row(‘array’)**

**$row = $query->next\_row(‘array’)**

**$row = $query->previous\_row(‘array’)**

Note

All the methods above will load the whole result into memory (prefetching). Use unbuffered\_row() for processing large result sets.

**unbuffered\_row()**

This method returns a single result row without prefetching the whole result in memory as row() does. If your query has more than one row, it returns the current row and moves the internal data pointer ahead.

$query = $this->db->query("YOUR QUERY");  
  
while ($row = $query->unbuffered\_row())  
{  
 echo $row->title;  
 echo $row->name;  
 echo $row->body;  
}

You can optionally pass ‘object’ (default) or ‘array’ in order to specify the returned value’s type:

$query->unbuffered\_row(); // object  
$query->unbuffered\_row('object'); // object  
$query->unbuffered\_row('array'); // associative array

[**Custom Result Objects**](#4d34og8)[**¶**](#3znysh7)

You can have the results returned as an instance of a custom class instead of a stdClass or array, as the result() and result\_array() methods allow. This requires that the class is already loaded into memory. The object will have all values returned from the database set as properties. If these have been declared and are non-public then you should provide a \_\_set() method to allow them to be set.

Example:

class User {  
  
 public $id;  
 public $email;  
 public $username;  
  
 protected $last\_login;  
  
 public function last\_login($format)  
 {  
 return $this->last\_login->format($format);  
 }  
  
 public function \_\_set($name, $value)  
 {  
 if ($name === 'last\_login')  
 {  
 $this->last\_login = DateTime::createFromFormat('U', $value);  
 }  
 }  
  
 public function \_\_get($name)  
 {  
 if (isset($this->$name))  
 {  
 return $this->$name;  
 }  
 }  
}

In addition to the two methods listed below, the following methods also can take a class name to return the results as: first\_row(), last\_row(), next\_row(), and previous\_row().

**custom\_result\_object()**

Returns the entire result set as an array of instances of the class requested. The only parameter is the name of the class to instantiate.

Example:

$query = $this->db->query("YOUR QUERY");  
  
$rows = $query->custom\_result\_object('User');  
  
foreach ($rows as $row)  
{  
 echo $row->id;  
 echo $row->email;  
 echo $row->last\_login('Y-m-d');  
}

**custom\_row\_object()**

Returns a single row from your query results. The first parameter is the row number of the results. The second parameter is the class name to instantiate.

Example:

$query = $this->db->query("YOUR QUERY");  
  
$row = $query->custom\_row\_object(0, 'User');  
  
if (isset($row))  
{  
 echo $row->email; // access attributes  
 echo $row->last\_login('Y-m-d'); // access class methods  
}

You can also use the row() method in exactly the same way.

Example:

$row = $query->custom\_row\_object(0, 'User');

[**Result Helper Methods**](#2s8eyo1)[**¶**](#2et92p0)

**num\_rows()**

The number of rows returned by the query. Note: In this example, $query is the variable that the query result object is assigned to:

$query = $this->db->query('SELECT \* FROM my\_table');  
  
echo $query->num\_rows();

Note

Not all database drivers have a native way of getting the total number of rows for a result set. When this is the case, all of the data is prefetched and count() is manually called on the resulting array in order to achieve the same result.

**num\_fields()**

The number of FIELDS (columns) returned by the query. Make sure to call the method using your query result object:

$query = $this->db->query('SELECT \* FROM my\_table');  
  
echo $query->num\_fields();

**free\_result()**

It frees the memory associated with the result and deletes the result resource ID. Normally PHP frees its memory automatically at the end of script execution. However, if you are running a lot of queries in a particular script you might want to free the result after each query result has been generated in order to cut down on memory consumption.

Example:

$query = $this->db->query('SELECT title FROM my\_table');  
  
foreach ($query->result() as $row)  
{  
 echo $row->title;  
}  
  
$query->free\_result(); // The $query result object will no longer be available  
  
$query2 = $this->db->query('SELECT name FROM some\_table');  
  
$row = $query2->row();  
echo $row->name;  
$query2->free\_result(); // The $query2 result object will no longer be available

**data\_seek()**

This method sets the internal pointer for the next result row to be fetched. It is only useful in combination with unbuffered\_row().

It accepts a positive integer value, which defaults to 0 and returns TRUE on success or FALSE on failure.

$query = $this->db->query('SELECT `field\_name` FROM `table\_name`');  
$query->data\_seek(5); // Skip the first 5 rows  
$row = $query->unbuffered\_row();

Note

Not all database drivers support this feature and will return FALSE. Most notably - you won’t be able to use it with PDO.

[**Class Reference**](#17dp8vu)[**¶**](#tyjcwt)

*class* CI\_DB\_result[¶](#3rdcrjn) result([*$type = 'object'*])[¶](#26in1rg)

| Parameters: | * **$type** (*string*) – Type of requested results - array, object, or class name |
| --- | --- |
| Returns: | Array containing the fetched rows |
| Return type: | array |

A wrapper for the result\_array(), result\_object() and custom\_result\_object() methods.

Usage: see [Result Arrays](#30j0zll).

result\_array()[¶](#lnxbz9)

| Returns: | Array containing the fetched rows |
| --- | --- |
| Return type: | array |

Returns the query results as an array of rows, where each row is itself an associative array.

Usage: see [Result Arrays](#30j0zll).

result\_object()[¶](#35nkun2)

| Returns: | Array containing the fetched rows |
| --- | --- |
| Return type: | array |

Returns the query results as an array of rows, where each row is an object of type stdClass.

Usage: see [Result Arrays](#30j0zll).

custom\_result\_object(*$class\_name*)[¶](#1ksv4uv)

| Parameters: | * **$class\_name** (*string*) – Class name for the resulting rows |
| --- | --- |
| Returns: | Array containing the fetched rows |
| Return type: | array |

Returns the query results as an array of rows, where each row is an instance of the specified class.

row([*$n = 0*[, *$type = 'object'*]])[¶](#44sinio)

| Parameters: | * **$n** (*int*) – Index of the query results row to be returned * **$type** (*string*) – Type of the requested result - array, object, or class name |
| --- | --- |
| Returns: | The requested row or NULL if it doesn’t exist |
| Return type: | mixed |

A wrapper for the row\_array(), row\_object() and ``custom\_row\_object() methods.

Usage: see [Result Rows](#1fob9te).

unbuffered\_row([*$type = 'object'*])[¶](#2jxsxqh)

| Parameters: | * **$type** (*string*) – Type of the requested result - array, object, or class name |
| --- | --- |
| Returns: | Next row from the result set or NULL if it doesn’t exist |
| Return type: | mixed |

Fetches the next result row and returns it in the requested form.

Usage: see [Result Rows](#1fob9te).

row\_array([*$n = 0*])[¶](#z337ya)

| Parameters: | * **$n** (*int*) – Index of the query results row to be returned |
| --- | --- |
| Returns: | The requested row or NULL if it doesn’t exist |
| Return type: | array |

Returns the requested result row as an associative array.

Usage: see [Result Rows](#1fob9te).

row\_object([*$n = 0*])[¶](#3j2qqm3)

| Parameters: | * **$n** (*int*) – Index of the query results row to be returned |
| --- | --- |
| Returns: | The requested row or NULL if it doesn’t exist |
| Return type: | stdClass |

Returns the requested result row as an object of type stdClass.

Usage: see [Result Rows](#1fob9te).

custom\_row\_object(*$n*, *$type*)[¶](#1y810tw)

| Parameters: | * **$n** (*int*) – Index of the results row to return * **$class\_name** (*string*) – Class name for the resulting row |
| --- | --- |
| Returns: | The requested row or NULL if it doesn’t exist |
| Return type: | $type |

Returns the requested result row as an instance of the requested class.

data\_seek([*$n = 0*])[¶](#4i7ojhp)

| Parameters: | * **$n** (*int*) – Index of the results row to be returned next |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Moves the internal results row pointer to the desired offset.

Usage: see [Result Helper Methods](#2et92p0).

set\_row(*$key*[, *$value = NULL*])[¶](#2xcytpi)

| Parameters: | * **$key** (*mixed*) – Column name or array of key/value pairs * **$value** (*mixed*) – Value to assign to the column, $key is a single field name |
| --- | --- |
| Return type: | void |

Assigns a value to a particular column.

next\_row([*$type = 'object'*])[¶](#1ci93xb)

| Parameters: | * **$type** (*string*) – Type of the requested result - array, object, or class name |
| --- | --- |
| Returns: | Next row of result set, or NULL if it doesn’t exist |
| Return type: | mixed |

Returns the next row from the result set.

previous\_row([*$type = 'object'*])[¶](#3whwml4)

| Parameters: | * **$type** (*string*) – Type of the requested result - array, object, or class name |
| --- | --- |
| Returns: | Previous row of result set, or NULL if it doesn’t exist |
| Return type: | mixed |

Returns the previous row from the result set.

first\_row([*$type = 'object'*])[¶](#2bn6wsx)

| Parameters: | * **$type** (*string*) – Type of the requested result - array, object, or class name |
| --- | --- |
| Returns: | First row of result set, or NULL if it doesn’t exist |
| Return type: | mixed |

Returns the first row from the result set.

last\_row([*$type = 'object'*])[¶](#qsh70q)

| Parameters: | * **$type** (*string*) – Type of the requested result - array, object, or class name |
| --- | --- |
| Returns: | Last row of result set, or NULL if it doesn’t exist |
| Return type: | mixed |

Returns the last row from the result set.

num\_rows()[¶](#3as4poj)

| Returns: | Number of rows in the result set |
| --- | --- |
| Return type: | int |

Returns the number of rows in the result set.

Usage: see [Result Helper Methods](#2et92p0).

num\_fields()[¶](#1pxezwc)

| Returns: | Number of fields in the result set |
| --- | --- |
| Return type: | int |

Returns the number of fields in the result set.

Usage: see [Result Helper Methods](#2et92p0).

field\_data()[¶](#49x2ik5)

| Returns: | Array containing field meta-data |
| --- | --- |
| Return type: | array |

Generates an array of stdClass objects containing field meta-data.

free\_result()[¶](#2p2csry)

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

Frees a result set.

Usage: see [Result Helper Methods](#2et92p0).

list\_fields()[¶](#147n2zr)

| Returns: | Array of column names |
| --- | --- |
| Return type: | array |

Returns an array containing the field names in the result set.

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