



## **The Handbook**

### **Codex**

[http://codex.wordpress.org/Function\\_Reference/wpdb\\_Class](http://codex.wordpress.org/Function_Reference/wpdb_Class)

### **Version Date**

9 August 2005

## Function Reference/wpdb Class

### *Interfacing With the Database*

WordPress provides a class of functions for all database manipulations. The class is called `wpdb` and is based on the [ezSQL](http://www.jvmultimedia.com/home/articles.php?articleId=2) (<http://www.jvmultimedia.com/home/articles.php?articleId=2>) class written and maintained by [Justin Vincent](http://www.jvmultimedia.com) (<http://www.jvmultimedia.com>). Though the WordPress class is slightly different than the `ezSQL` class, their use is essentially the same. Please see the [ezSQL documentation](http://justinvincent.com/home/docs/ezsql/ez_sql_help.htm) ([http://justinvincent.com/home/docs/ezsql/ez\\_sql\\_help.htm](http://justinvincent.com/home/docs/ezsql/ez_sql_help.htm)) for more information.

### *query - Run Any Query on the Database*

The `query` function allows you to execute *any* query on the WordPress database. It is best to use a more specific function, however, for SELECT queries.

```
<?php $wpdb->query('query'); ?>
```

`query`

(string) The query you wish to run.

If there are any query results, the function will return an integer corresponding to the number of rows affected and the query results will be cached for use by other `wpdb` functions. If there are no results, the function will return (int) 0. If there is a MySQL error, the function will return FALSE. (**Note:** since both 0 and FALSE can be returned, make sure you use the correct comparison operator: equality `==` vs. identity `===`).

### Examples

Add Post 13 to Category 2:

```
$wpdb->query("
    INSERT INTO $wpdb->post2cat (post_id, category_id)
    VALUES (13, 2)");
```

Delete the 'gargle' meta key and value from Post 13.

```
$wpdb->query("
    DELETE FROM $wpdb->postmeta WHERE post_id = '13'
    AND meta_key = 'gargle'");
```

*Performed in WordPress by `delete_post_meta()`.*

Set the parent of [Page](#) 15 to Page 7.

```
$wpdb->query("
```

```
UPDATE $wpdb->posts SET post_parent = 7
WHERE post_ID = 15 AND post_status = 'static');
```

## ***get\_var - SELECT a Variable***

The `get_var` function returns a single variable from the database. Though only one variable is returned, the entire result of the query is cached for later use.

```
<?php $wpdb->get_var('query',column_offset,row_offset); ?>
```

**query**

(string) The query you wish to run. Setting this parameter to `null` will return the specified variable from the cached results of the previous query.

**column\_offset**

(integer) The desired column (**0** being the first). Defaults to **0**.

**row\_offset**

(integer) The desired row (**0** being the first). Defaults to **0**.

## **Examples**

Retrieve the name of Link Category 4.

```
$name = $wpdb->get_var("SELECT cat_name FROM $wpdb->linkcategories WHERE
cat_id=4");
echo $name;
```

*Performed in WordPress by `get_linkcatname()`.*

## ***get\_row - SELECT a Row***

To retrieve an entire row from a query, use `get_row`. The function can return the row as an object, an associative array, or as a numbered array. If more than one row is returned by the query, only the specified row is returned by the function, but all rows are cached for later use.

```
<?php $wpdb->get_row('query', output_type, row_offset); ?>
```

**query**

(string) The query you wish to run. Setting this parameter to `null` will return the specified row from the cached results of the previous query.

**output\_type**

One of three pre-defined constants. Defaults to `OBJECT`.

- `OBJECT` - result will be output as an object.
- `ARRAY_A` - result will be output as an associative array.
- `ARRAY_N` - result will be output as a numbered array.

**row\_offset**

(integer) The desired row (**0** being the first). Defaults to **0**.

## Examples

Get all the information about Link 10.

```
$mylink = $wpdb->get_row("SELECT * FROM $wpdb->links WHERE link_id = 10");
```

The properties of the `$mylink` object are the column names of the result from the SQL query (in this all of the columns from the `$wpdb->links` table).

```
echo $mylink->link_id; // prints "10"
```

In contrast, using

```
$mylink = $wpdb->get_row("SELECT * FROM $wpdb->links WHERE link_id = 10",  
ARRAY_A);
```

would result in an associative array:

```
echo $mylink['link_id']; // prints "10"
```

and

```
$mylink = $wpdb->get_row("SELECT * FROM $wpdb->links WHERE link_id = 10",  
ARRAY_N);
```

would result in a numbered array:

```
echo $mylink[1]; // prints "10"
```

## ***get\_col - SELECT a Column***

To SELECT a column, use `get_col`. This function outputs a dimensional array. If more than one column is returned by the query, only the specified column will be returned by the function, but the entire result is cached for later use.

```
<?php $wpdb->get_col('query',column_offset); ?>
```

**query**

(string) the query you wish to execute. Setting this parameter to `null` will return the specified column from the cached results of the previous query.

**column\_offset**

(integer) The desired column (**0** being the first). Defaults to **0**.

## Examples

Get all the Categories to which Post 103 belongs.

```
$postcats = $wpdb->get_col("SELECT category_id  
FROM $wpdb->post2cat  
WHERE post_id = 103  
ORDER BY category_id");
```

Performed in WordPress by `wp_get_post_cats()`.

## ***get\_results - SELECT Generic Results***

Generic, multiple row results can be pulled from the database with `get_results`. The function returns the entire query result as a dimensional array. Each element of this array corresponds to one row of the query result and, like `get_row` can be an object, an associative array, or a numbered array.

```
<?php $wpdb->get_results('query', output_type); ?>
```

`query`

(string) The query you wish to run. Setting this parameter to `null` will return the data from the cached results of the previous query.

`output_type`

One of three pre-defined constants. Defaults to `OBJECT`. See [SELECT a Row](#) and its examples for more information.

- `OBJECT` - result will be output as an object.
- `ARRAY_A` - result will be output as an associative array.
- `ARRAY_N` - result will be output as a numbered array.

## **Examples**

Get the IDs and Titles of all the Drafts by User 5 and echo the Titles.

```
$fivesdrafts = $wpdb->get_results("SELECT ID, post_title FROM $wpdb->posts  
    WHERE post_status = 'draft' AND post_author = 5");  
  
foreach ($fivesdrafts as $fivesdraft) {  
    echo $fivesdraft->post_title;  
}
```

## ***escape - Escape For INSERTing***

If you're inserting text into the database, make sure it is escaped properly first. This can be conveniently done with `escape`.

```
<?php $wpdb->escape('string'); ?>
```

`string`

(string) The string of text to be escaped for proper insertion.

## **Examples**

Add Meta key => value pair "Harriet's Adages" => "WordPress' database interface is like Sunday Morning: Easy." to Post 10.

```
$metakey = $wpdb->escape("Harriet's Adages");  
$metavalue = $wpdb->escape("WordPress' database interface is like Sunday
```

```
Morning: Easy.");

$wpdb->query("
    INSERT INTO $wpdb->postmeta
    (post_id,meta_key,meta_value)
    VALUES ('10','$metakey','$metavalue')");
```

*Performed in WordPress by `add_meta()`.*

## ***show/hide\_errors - Show and Hide SQL Errors***

You can turn error echoing on and off with the `show_errors` and `hide_errors`, respectively.

```
<?php $wpdb->show_errors(); ?>
<?php $wpdb->hide_errors(); ?>
```

You can also print the error (if any) generated by the most recent query with `print_error`.

```
<?php $wpdb->print_error(); ?>
```

## ***get\_col\_info - Getting Column Information***

You can retrieve information about the columns of the most recent query result with `get_col_info`. This can be useful when a function has returned an OBJECT whose properties you don't know. The function will output the desired information from the specified column, or an array with information on *all* columns from the query result if no column is specified.

```
<?php $wpdb->get_col_info('type', offset); ?>
```

**type**

(string) What information you wish to retrieve. May take on any of the following values (list taken from the [ezSQL docs](http://justinvincent.com/home/docs/ezsql/ez_sql_help.htm) ([http://justinvincent.com/home/docs/ezsql/ez\\_sql\\_help.htm](http://justinvincent.com/home/docs/ezsql/ez_sql_help.htm))). Defaults to **name**.

- **name** - column name. Default.
- **table** - name of the table the column belongs to
- **max\_length** - maximum length of the column
- **not\_null** - 1 if the column cannot be NULL
- **primary\_key** - 1 if the column is a primary key
- **unique\_key** - 1 if the column is a unique key
- **multiple\_key** - 1 if the column is a non-unique key
- **numeric** - 1 if the column is numeric
- **blob** - 1 if the column is a BLOB
- **type** - the type of the column
- **unsigned** - 1 if the column is unsigned
- **zerofill** - 1 if the column is zero-filled

**offset**

(integer) Specify the column from which to retrieve information (with **0** being the first column). Defaults to **-1**.

- -1 - Retrieve information from all columns. Output as array. Default.
- Non-negative integer - Retrieve information from specified column (0 being the first).

## ***flush - Clearing the Cache***

You can clear the SQL result cache with `flush`.

```
<?php $wpdb->flush(); ?>
```

This clears `$wpdb->last_result`, `$wpdb->last_query`, and `$wpdb->col_info`.

## ***Class Variables***

`$show_errors`

Whether or not [Error echoing](#) is turned on. Defaults to TRUE.

`$num_queries`

The number of queries that have been executed.

`$last_query`

The most recent query to have been executed.

`$queries`

You may save all of the queries run on the database and their stop times by setting the `SAVEQUERIES` constant to TRUE (this constant defaults to FALSE). If `SAVEQUERIES` is TRUE, your queries will be stored in this variable as an array.

`$last_results`

The most recent query results.

`$col_info`

The column information for the most recent query results. See [Getting Column Information](#).

## ***Tables***

The WordPress database tables are easily referenced in the `wpdb` class.

`$posts`

The table of Posts.

`$users`

The table of Users.

`$categories`

The table of Categories.

`$post2cat`

The table which defines which Posts are members of what Category.

`$comments`

The Comments table.

`$links`

The table of Links.

`$linkcategories`

The table of Link Categories.

\$options

The Options table.

\$postmeta

The Meta Content (a.k.a. [Custom Fields](#)) table.