

Using Mysql - PostgreSQL with PHP

System Software Division
NIC

Topics Covered

- Requirements
 - How to check if mysql is configured with PHP
 - Basic Mysql commands
 - How to connect to mysql from PHP
 - How to use PostgreSQL with php
-
-

Prerequisites :

In order to have these database functions available, you must compile PHP with MySQL/PostgreSQL support.

- To find out if the MySQL / PostgreSQL interface is available in your copy of PHP, call the `phpinfo()` function and look for a section titled “MySQL Support” / “PostgreSQL Support.”
 - In order to enable PostgreSQL support, `--with-pgsql[=DIR]` is required when you compile PHP. DIR is the PostgreSQL base install directory, defaults to `/usr/local/pgsql`.
-
-

Prerequisites :

- use the `--with-mysql[=DIR]` configuration option where the optional `[DIR]` points to the MySQL installation directory.

Eg: #./configure --prefix=/usr/local/php --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql=/usr/local/mysql (enter)

- In order to enable PostgreSQL support, `--with-pgsql[=DIR]` is required when you compile PHP. `DIR` is the PostgreSQL base install directory, defaults to `/usr/local/pgsql`.
-
-

Prerequisites :

- Configuring PHP to work with MySQL / PostgreSQL
 - Open up your php.ini file and find the line
 - ;extension=php_pgsql.dll
 - ;extension=php_mysql.dll

To enable the MySQL /PostgreSQL extension, delete the semi-colon at the beginning of that line.



Connecting to MySQL from PHP

An example CODE to connect to mysql :

```
<?php
// Connecting, selecting database
$link = mysql_connect('mysql_host', 'mysql_user', 'mysql_password')
    or die('Could not connect: ' . mysql_error());
echo 'Connected successfully';
mysql_select_db('my_database') or die('Could not select database');
// Performing SQL query
$query = 'SELECT * FROM my_table';
$result = mysql_query($query) or die('Query failed: ' . mysql_error());
```

----> **continued**

Connecting to MySQL from PHP

```
// Printing results in HTML
echo "<table>\n";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";
// Free resultset and closing connection
mysql_free_result($result); mysql_close($link); ?>
```

MySql Functions

mysql_connect -- Open a connection to a MySQL Server

Syntax : resource mysql_connect ([string server [, string username [, string password [, bool new_link [, int client_flags]]]])

Eg:

- \$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
 - \$link = mysql_connect('example.com:3307', 'mysql_user', 'mysql_password');
 - \$link = mysql_connect('localhost:/tmp/mysql.sock', 'mysql_user', 'mysql_password');
-
-

MySql Functions

- **mysql_select_db** -- Select a MySQL database

Syntax: `bool mysql_select_db (string database_name [, resource link_identifier])`

Eg: `$db_selected = mysql_select_db('foo', $link);`

- **mysql_query** -- Send a MySQL query

Syntax: `:resource mysql_query (string query [, resource link_identifier])`

Eg: `$result = mysql_query('SELECT * WHERE 1=1');`

```
if (!$result) {  
    die('Invalid query: ' . mysql_error());  
}
```



MySql Functions

- **Functions to read result set :**

- string `mysql_result` (resource result, int row [, mixed field])

- array `mysql_fetch_row` (resource result)

- array `mysql_fetch_array` (resource result [, int result_type]) - result type

MYSQL_ASSOC, MYSQL_NUM, and the default value of
MYSQL_BOTH.

- **mysql_error** -- Returns the text of the error message from previous
MySQL operation

Syntax: string `mysql_error` ([resource link_identifier])

related functions : `mysql_errno()`

MySql Functions

- `mysql_affected_rows` -- Get the number of affected rows by the last `INSERT`, `UPDATE`, `REPLACE` or `DELETE` query associated with `link_identifier`.

Eg: `int mysql_affected_rows ([resource link_identifier])`

- `mysql_free_result()` will free all memory associated with the result identifier result.

`bool mysql_free_result (resource result)`

- `mysql_close` -- Close MySQL connection

`bool mysql_close ([resource link_identifier])`

Connecting to PostgreSQL from PHP

```
<?php
// Connecting, selecting database
$dbconn = pg_connect("host=localhost dbname=publishing user=www
    password=foo")
    or die('Could not connect: ' . pg_last_error());

// Performing SQL query
$query = 'SELECT * FROM authors';
$result = pg_query($query) or die('Query failed: ' . pg_last_error());
// Printing results in HTML
echo "<table>\n";
```

--> **Cont.**

Connecting to PostgreSQL from PHP

```
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC)) {  
    echo "\t<tr>\n";  
    foreach ($line as $col_value) {  
        echo "\t\t<td>$col_value</td>\n";  
    }  
    echo "\t</tr>\n";  
}  
echo "</table>\n";  
// Free resultset and closing  
pg_free_result($result); pg_close($dbconn);  
?>
```

PostgreSQL functions

- `pg_connect` -- Open a PostgreSQL connection

Syntax: `resource pg_connect (string connection_string [, int connect_type])`

Eg: `<?php`

```
$dbconn = pg_connect("dbname=mary");
```

```
$dbconn2 = pg_connect("host=localhost port=5432  
dbname=mary");
```

```
$dbconn3 = pg_connect("host=sheep port=5432 dbname=mary  
user=lamb password=foo");
```

```
$conn_string = "host=sheep port=5432 dbname=test user=lamb  
password=bar";
```

```
$dbconn4 = pg_connect($conn_string);
```

PostgreSQL functions

- `pg_execute` -- Sends a request to execute a prepared statement with given parameters, and waits for the result.

Syntax : `resource pg_execute (resource connection, string stmtname, array params)`

eg: `<?php`

```
// Prepare a query for execution
```

```
$result = pg_prepare($dbconn, 'my_query', 'SELECT * FROM  
shops WHERE name = $1');
```

```
// Execute the prepared query.
```

```
$result = pg_execute($dbconn, "my_query", array("Joe's  
Widgets"));
```

PostgreSQL functions

- `pg_query` -- Execute a query

Syntax:

```
resource pg_query ( string query )
```

```
resource pg_query ( resource connection, string query )
```

```
eg: $result = pg_query($conn, "SELECT author, email FROM
    authors");
if (!$result) {
    echo "An error occurred.\n";
    exit;
}
```

PostgreSQL functions

- `pg_fetch_array` -- Fetch a row as an array

Syntax: `array pg_fetch_array (resource result [, int row [, int result_type]])`

Eg:

```
$arr = pg_fetch_array($result, 0, PGSQL_NUM);
```

```
echo $arr[0] . " <- Row 1 Author\n";
```

```
echo $arr[1] . " <- Row 1 E-mail\n";
```

```
$arr = pg_fetch_array($result, NULL, PGSQL_ASSOC);
```

```
echo $arr["author"] . " <- Row 2 Author\n";
```

```
echo $arr["email"] . " <- Row 2 E-mail\n";
```
