

## Server side scripting and databases

How Web Applications interact with server side databases

## mysql

mysqld	database process that runs in the background
mysqladmin	command line style admin control
mysql	SQL processor

Lots of other components exist, typically in `/bin`  
`/MAMP/Library/bin` in MAMP

mysql - <http://www.mysql.com/>

## mysql

Start the database process in the background

Use `mysqladmin` to create, change or drop databases

Use `mysql` to issue SQL statements at the command line - useful during debugging and initial application development

Use PHP built in functions to access live data from a web application

## mysql

Open Source database, issued under a dual commercial license as well

Rather than just being one program, actually consists of many separate components

Aim to download a complete pack containing apache, PHP and mysql (XAMPP pc or MAMP mac)

## mysqladmin

Example

```
./mysqladmin -u "root" -p create empdb
```

Creates a database called `empdb` (prompts for username and password - `root`, `root` in default set up)

Demo



## Connecting and using MySQL from PHP

PHP provides many MySQL specific functions

mysql_connect	Open a link/connection to a mysql database
mysql_select_db	Choose a specific database on a MySQL server
mysql_query	run an SQL statement on an opened database
mysql_fetch_array	process a result set
mysql_close	Close a MySQL connection

## Opening a connection to a mysql server

Use `mysql_connect`

mysql\_connect (PHP 4, PHP 5)

Open a connection to a MySQL Server

Description

```
resource mysql_connect ( [string $server [, string $username [, string $password [, bool $new_link [, int $client_flags]]]])
```

Opens or reuses a connection to a MySQL server.

## Opening a connection to a mysql server

Use `mysql_connect`

```
<?php
// we connect to example.com and port 3307
$link = mysql_connect('example.com:3307', 'user5', 'qwerty5');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
```

Opens a connection to the mysql server on **example.com:3307**, using **user5** with password **qwerty5**

## Local version

To connect to the local mysql database

```
<?php
// Create connection
$link = mysql_connect('localhost:8889', 'root', 'root');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
?>
```

Demo

## Local version

This example will fail to connect ...

```
<?php
// Create connection
$link = mysql_connect('localhost:8889', 'missing', 'root');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
?>
```

Demo

`$link` is only really used for testing the connection and for closing the connection - `$link` is of type **resource** - a built in PHP type for this kind of connection

## Select a particular database on a mysql server

Use `mysql_select_db`

mysql\_select\_db (PHP 4, PHP 5)

Select a MySQL database

Description

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

Sets the current active database on the server that's associated with the specified link identifier. Every subsequent call to `mysql_query()` will be made on the active database.

Opens a distinct connection to a particular named database on the previously opened mysql server

## Select a particular database on a mysql server

### Use `mysql_select_db`

```
<?php
// we connect to example.com and port 3307
$link = mysql_connect('example.com:3307', 'user5', 'qwerty5');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';

$db_selected = mysql_select_db('foo', $link);
if (!$db_selected) {
    die ("Can't use foo : " . mysql_error());
};

mysql_close($link);
```

Opens a connection to the database `foo`, using the `$link` resource

## Running a SQL statement against the database

### Use `mysql_query`

```
$result = mysql_query('insert into foo (field1, field2, field3) values (120,
2000, 22000)');
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

Runs the SQL query - if **DDL** or **DML** the result will indicate whether the query ran successfully or not (**bool**)

## Local version

### To open the `empdb` database

```
<?php
// we connect to localhost
$link = mysql_connect('localhost:8889', 'root', 'root');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';

$db_selected = mysql_select_db('empdb', $link);
if (!$db_selected) {
    die ("Can't use empdb : " . mysql_error());
}
else {
    print "Opened database empdb correctly";
};

mysql_close($link);
?>
```

Demo

## Local version

### Use `mysql_query`

```
$result = mysql_query('insert into emp (empno, ename, sal) values (120,
"WAHXYZ", 22000)');
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

This inserts a single record (or fails)

Demo

## Running a SQL statement against the database

### Use `mysql_query`

`mysql_query` (PHP 4, PHP 5)

Send a MySQL query

Description  
resource `mysql_query` ( string \$query [, resource \$link\_identifier ] )

`mysql_query()` sends a unique query (multiple queries are not supported) to the currently active database on the server that's associated with the specified `link_identifier`.

Runs an SQL statement against the opened **DB**

## Running a SQL statement against the database

### Use `mysql_query`

```
$result = mysql_query('select * from emp');
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

In this case `$result` is a resource which points to the result set (i.e. the select result) - structure that may have **many** records

So how can we process it?

## To process a returned results

### Use `mysql_fetch_array`

`mysql_fetch_array` (PHP 4, PHP 5)

Fetch a result row as an associative array, a numeric array, or both

Description  
array `mysql_fetch_array` ( resource `$result` [, int `$result_type` ] )

Returns an array that corresponds to the fetched row and moves the internal data pointer ahead.

Returns the next row in the structure, until there are none left (when it returns false)

## To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	23000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

```
$row = mysql_fetch_array($result)
```

empno	ename	sal	
\$row	121	JONES	23000

## To process a returned results

### Use `mysql_fetch_array`

```
$result = mysql_query("SELECT query ...");
```

```
while ($row = mysql_fetch_array($result) {  
    process the row  
}
```

As there are 0 or more rows in `$result`, use a while loop to extract each row, process it and then get the next row...

## To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	23000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

```
$row = mysql_fetch_array($result)
```

empno	ename	sal	
\$row	122	AVERY	22500

## To process a returned results

### `$row` becomes an associative array

```
$row = mysql_fetch_array($result)
```

empno	ename	sal
120	SMITH	22000
121	JONES	23000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

empno	ename	sal	
\$row	120	SMITH	22000

## To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	23000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

```
$row = mysql_fetch_array($result)
```

empno	ename	sal	
\$row	124	MITCHELL	26000

## To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	23000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

```
$row = mysql_fetch_array($result)
```

	empno	ename	sal
\$row	127	WEBB	22000

## To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	23000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

```
$row = mysql_fetch_array($result)
```

```
$row false ← $row becomes false when  
there are no more rows
```

## Local version

### Use `mysql_fetch_array`

```
$result = mysql_query("SELECT * from emp");  
  
while ($row = mysql_fetch_array($result)){  
    print $row["empno"]." ".$row["ename"]." ".$row["sal"]."<br />";  
};
```

Prints out the records

Demo

## To close a database connection

### Use `mysql_close`

`mysql_close` (PHP 4, PHP 5)

Close MySQL connection

Description  
bool `mysql_close` ( [resource \$link\_identifier] )

`mysql_close()` closes the non-persistent connection to the MySQL server that's associated with the specified link identifier. If `link_identifier` isn't specified, the last opened link is used.

Closes the connection and releases the resources

## Local version

### To close the `empdb` database

```
<?php  
// we connect to localhost  
$link = mysql_connect('localhost:8889', 'root', 'root');  
if (!$link) {  
    die('Could not connect: ' . mysql_error());  
}  
echo 'Connected successfully';  
  
$db_selected = mysql_select_db('empdb', $link);  
if (!$db_selected) {  
    die ("Can't use empdb : " . mysql_error());  
}  
else {  
    print "Opened database empdb correctly";  
};  
  
mysql_close($link);  
?>
```