

**Server side scripting and databases**

How Web Applications interact with server side databases

**mySQL**

<b>mysqld</b>	database process that runs in the background
<b>mysqladmin</b>	command line style admin control
<b>mysql</b>	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

## mysql

### Example

```
./mysql -u "root" -p empdb
```

**Start a database connection to empdb (prompts for username and password - root, root in default set up)**

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11 to server version: 5.0.19
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

## sql - creating a table using the DDL

### Syntax

```
create table tablename (
  fieldname type,
  fieldname type,
  ...);
```

**Here used to create a table called emp**

```
mysql> create table emp(empno int, ename char(255), sal int);
Query OK, 0 rows affected (0.06 sec)
```

```
mysql>
```

emp	empno	int
	ename	char
	sal	int

## sql - inserting records using the DML

### Syntax

```
insert into table [(columnname, columnname, ...)]
  values (value, value,...)
```

**Here used to insert a record into emp**

```
mysql> insert into emp (empno,ename,sal) values (120,
  "SMITH", 22000);
Query OK, 1 row affected, 1 warning (0.37 sec)
```

```
mysql>
```

## sql - querying the database using the DQL

### Syntax

```
select * or expression
from relations
[where expression]
```

**Here used to show all rows in emp**

```
mysql> select * from emp;
+-----+-----+-----+
| empno | ename | sal |
+-----+-----+-----+
| 120   | SMITH | 22000 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

## sql - running a SQL source file

### Example

```
source path/filename
```

**Load and execute a series of SQL statements in a single file (separated by semicolons)**

```
insert into emp (empno,ename,sal) values (121, "JONES", 23000);
insert into emp (empno,ename,sal) values (122, "AVERY", 22500);
insert into emp (empno,ename,sal) values (124, "MITCHELL", 26000);
insert into emp (empno,ename,sal) values (127, "WEBB", 22000);
insert into emp (empno,ename,sal) values (129, "TENNENT", 22000);
insert into emp (empno,ename,sal) values (130, "RUSSELL", 25000);
```

## sql - running a SQL source file

### Example

```
source /Applications/MAMP/htdocs/webscripting/php4/insertstaff.sql
```

**Load and execute a series of SQL statements in a single file (separated by semicolons)**

```
mysql> source /Applications/MAMP/htdocs/webscripting/php4/insertstaff.sql
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
```

## Connecting and using mySQL from PHP

PHP provides many mySQL specific functions

<code>mysql_connect</code>	Open a link/connection to a MySQL database
<code>mysql_select_db</code>	Choose a specific database on a MySQL server
<code>mysql_query</code>	run an SQL statement on an opened database
<code>mysql_fetch_array</code>	process a result set
<code>mysql_close</code>	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  
$link = mysql_connect('example.com:3307', 'user5', 'qwert5');  
if (!$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 qwert5

## 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`

```
$result = mysql_query('insert into emp (empno, ename, sal) values (120,
"WALKER", 22000)');
if (!$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 an 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

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

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

```
$row empno ename sal
```

```
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

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

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

```
$row empno ename sal
```

```
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

```
$row empno ename sal
```

```
120 SMITH 22000
```

### To process a returned results

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

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

```
$row empno ename sal
```

```
124 MITCHELL 26000
```

### To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	29000
122	AVERY	22500
124	MITCHELL	26000
127	WEBB	22000

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

empno	ename	sal
127	WEBB	22000

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

### To process a returned results

empno	ename	sal
120	SMITH	22000
121	JONES	29000
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

### 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);
?>
```