Server side scripting and databases

How Web Applications interact with server side databases

mySQL - http://www.mysql.com/

mySQL

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

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

... if doing this on your own machine...

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>SQL processor</td>
</tr>
<tr>
<td>mysqladmin</td>
<td>command line style admin control</td>
</tr>
<tr>
<td>mysqld</td>
<td>database process that runs in the background</td>
</tr>
</tbody>
</table>

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

/MAMP/Library/bin in MAMP

**mySQL - local**

Start the database process in the background

**mySQL - dedicated server**

Use whatever interface is supplied to create a DB
**MySQL - dedicated server**

*Use whatever interface is supplied to create a DB*

studentnet.kingston.ac.uk
mySQL - dedicated server

Use whatever interface is supplied to create a DB

Database Management

Through this interface you can manage the mySQL database associated with your account on the StudentWeb Web Server:

Create MySQL: Database

Database Name:

db_ku12492

Username:

ku12492

Database Password:

Confirm Password:

Create Database

Keep a note of these values

DON’T use your normal password

mySQL - dedicated server

Use whatever interface is supplied to create a DB

Database Management

The database db_ku12492 has been successfully created.

You can now access the database using the following connection parameters:

- Database Host: studentnet.kingston.ac.uk
- Port: 3306
- Username: ku12492
- Password: As provided in the previous form

You can now:

1. Go back to the Database Management Tool
2. Manage your database with phpMyAdmin
3. Visit your Student Website! (You will need to have logged in to the Student Webserver for this to be available)

Keep a note of these values

phpMyAdmin

Use a web interface to setup DBs, tables
**phpMyAdmin**

Use a web interface to set up DBs, tables

---

**phpMyAdmin - database setup at kingston**

One database created by the setup process

```
db_kxxxxxxxxx
```

On your own machine create a database and keep a note of the name (for later)

---

**To create a table**

Click on structure, name table and number of fields
To create a table

Add fields...

Alternative - creating a table using the DDL/SQL

Syntax

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

Here used to create a table called emp

create table emp(empno int, ename char(255), sal int);
Query OK, 0 rows affected (0.06 sec)
sql - inserting records using the DML

**Syntax**

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

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

```sql
insert into emp (empno, ename, sal) values (120, "SMITH", 22000);
```

sql - querying the database using the DQL

**Syntax**

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

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

```sql
select * from emp;
```

```
+-------+-------+-------+
<table>
<thead>
<tr>
<th>empno</th>
<th>ename</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>SMITH</td>
<td>22000</td>
</tr>
</tbody>
</table>
+-------+-------+-------+
```

1 row in set (0.00 sec)

sql - running many SQL lines

**SQL statements separated by semicolons**

```sql
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);
```
Connecting and using mySQL from PHP

PHP provides many mySQL specific functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql_connect</td>
<td>Open a link/connection to a mysql database</td>
</tr>
<tr>
<td>mysql_select_db</td>
<td>Choose a specific database on a mySQL server</td>
</tr>
<tr>
<td>mysql_query</td>
<td>Run an SQL statement on an opened database</td>
</tr>
<tr>
<td>mysql_fetch_array</td>
<td>Process a result set</td>
</tr>
<tr>
<td>mysql_close</td>
<td>Close a mySQL connection</td>
</tr>
</tbody>
</table>

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
To connect to the local mysql database - mamp

```php
<?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

To connect to the local mysql database - mamp

```php
<?php
// Create connection
$link = mysql_connect('studentnet.kingston.ac.uk:3306', 'kxxxx', 'password');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
?>
```

Local version

This example will fail to connect ...

```php
<?php
// Create connection
$link = mysql_connect('studentnet.kingston.ac.uk:3306', 'kxxxx', 'missing');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
?>
```

$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 connection to the database foo, using the $link resource

```php
<?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);
?>
```

Demo
To open the empdb database

```php
<?php
// we connect to localhost
$link = mysql_connect('studentnet.kingston.ac.uk:3306', 'kxxxx', 'missing');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
$db_selected = mysql_select_db('db_kxxxxxxx', $link);
if (!$db_selected) {
    die ('Can\'t use database : ' . mysql_error());
} else {
    print "Opened database correctly";
}
mysql_close($link);
?>
```

Running a SQL statement against the database

Use `mysql_query`

`mysql_query` (PHP 4, PHP 5)

Send a MySQL query

Description

```php
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

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

Running a SQL statement against the database

Use `mysql_query`

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

---

**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

Use mysql_fetch-array

```php
$result = mysql_query("SELECT query ...");
while ($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 becomes an associative array

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

To process a returned results

```
$empno  $ename    $sal
120     SMITH    22000
```

To process a returned results

```
$empno  $ename    $sal
121     JONES    23000
```
To process a returned results

<table>
<thead>
<tr>
<th>empno</th>
<th>ename</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>SMITH</td>
<td>22000</td>
</tr>
<tr>
<td>121</td>
<td>JONES</td>
<td>23000</td>
</tr>
<tr>
<td>122</td>
<td>AVERY</td>
<td>22500</td>
</tr>
<tr>
<td>124</td>
<td>MITCHELL</td>
<td>26000</td>
</tr>
<tr>
<td>127</td>
<td>WEBB</td>
<td>22000</td>
</tr>
</tbody>
</table>

$row = mysql_fetch_array($result)

<table>
<thead>
<tr>
<th>empno</th>
<th>ename</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>AVERY</td>
<td>22500</td>
</tr>
</tbody>
</table>

$row = mysql_fetch_array($result)

<table>
<thead>
<tr>
<th>empno</th>
<th>ename</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>124</td>
<td>MITCHELL</td>
<td>26000</td>
</tr>
</tbody>
</table>

$row = mysql_fetch_array($result)

<table>
<thead>
<tr>
<th>empno</th>
<th>ename</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>127</td>
<td>WEBB</td>
<td>22000</td>
</tr>
</tbody>
</table>
To process a returned results

<table>
<thead>
<tr>
<th>empno</th>
<th>ename</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>SMITH</td>
<td>22000</td>
</tr>
<tr>
<td>121</td>
<td>JONES</td>
<td>23000</td>
</tr>
<tr>
<td>122</td>
<td>AVERY</td>
<td>22500</td>
</tr>
<tr>
<td>124</td>
<td>MITCHELL</td>
<td>26000</td>
</tr>
<tr>
<td>127</td>
<td>WEBB</td>
<td>22000</td>
</tr>
</tbody>
</table>

$row = mysql_fetch_array($result)

$row false

$row becomes false when there are no more rows

Local version

Use **mysql_fetch-array**

```php
$result = mysql_query("SELECT * FROM emp");
while ($row = mysql_fetch_array($result)) {
    print $row["empno"] . " " . $row["ename"] . " " . $row["sal"] . "\n";
}
```

Prints out the records

To close a database connection

Use **mysql_close**

```php
mysql_close (PHP 4, PHP 5)
Close MySQL connection
```

Description

`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
<?php
// we connect to localhost
$link = mysql_connect('localhost:8889', 'root', 'root');

....

mysql_close($link);
?>
```