

Server side scripting and databases

How Web Applications interact with server side databases

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

The screenshot shows the MySQL website homepage. At the top, it says "MySQL. The world's most popular open source database". Below this is a navigation menu with links for Home, Products, Services, Partners, Community, Customers, Why MySQL?, News & Events, About, and How to Buy. The main content area is divided into several sections:

- News:** Includes articles like "Scale to New Heights at the 2007 MySQL Conference & Expo" and "MySQL Enterprise Unlimited: Site-Wide Agreements Now Available".
- Free White Papers:** Lists papers such as "MySQL Enterprise" and "MySQL Network Monitoring and Advisory Services".
- MySQL Embedded (OEM/MSV):** Promotes "Choosing an Embedded Database".
- Downloads:** Features "MySQL 5.2 Alpha - Falcon Preview".
- Documentation:** Provides links to various guides.
- MySQL Enterprise Unlimited:** A large central banner with the text "Deploy an unlimited number of MySQL Enterprise Servers for the cost of a single CPU of Oracle Enterprise Edition!" and a "Learn More" link.
- MySQL Enterprise:** Lists benefits like "Most Reliable, Secure, Up-To-Date" and "Achieve MySQL Best Practices".
- MySQL Training:** Lists upcoming seminars and training events, such as "MySQL High Availability" in Washington DC and "MySQL 5.0 for Developers" in Montreal.
- Free Web Seminars:** Promotes "MySQL Enterprise Monitoring and Advisory Services for Security, Replication, and Performance Tuning".
- Learn how MySQL is used in:** Lists industries like Web, Retail, Telecom, and Travel.
- High Availability:** Promotes a "Guide to High Availability" with a "Get the White Paper" link.

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

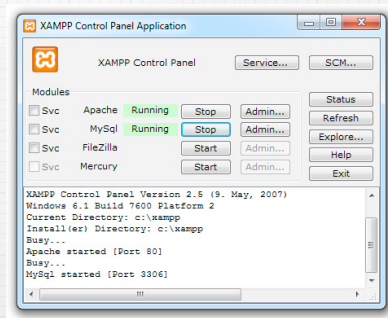
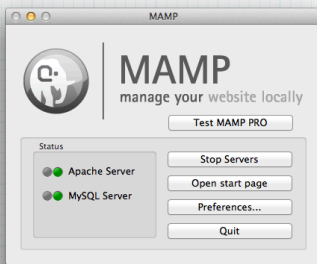
mysqld	database process that runs in the background
mysqladmin	command line style admin control
mysql	SQL processor

Lots of other components exist, typically in /bin

/MAMP/Library/bin in MAMP

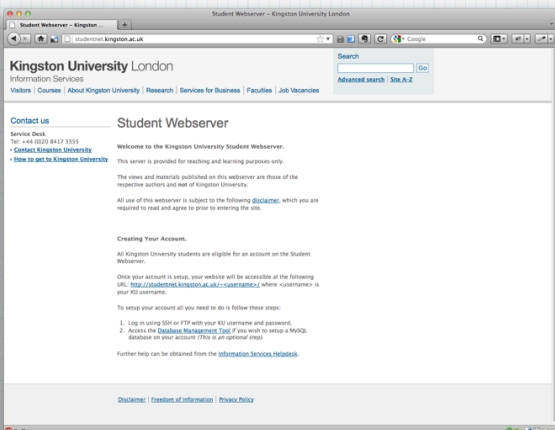
mysql - local

Start the database process in the background



mysql - dedicated server

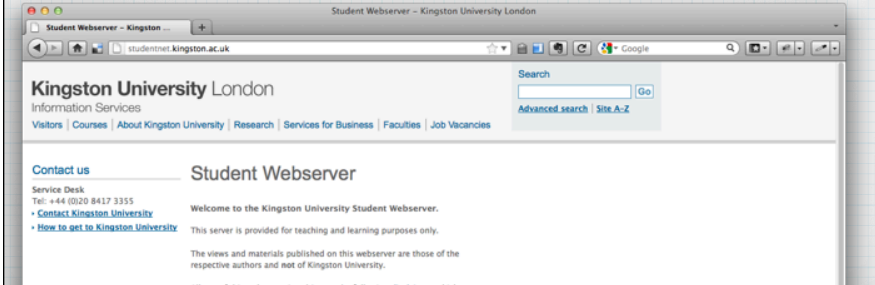
Use whatever interface is supplied to create a DB



mysql - dedicated server

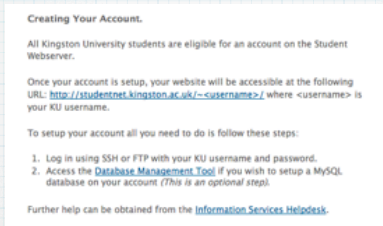
Use whatever interface is supplied to create a DB

studentnet.kingston.ac.uk



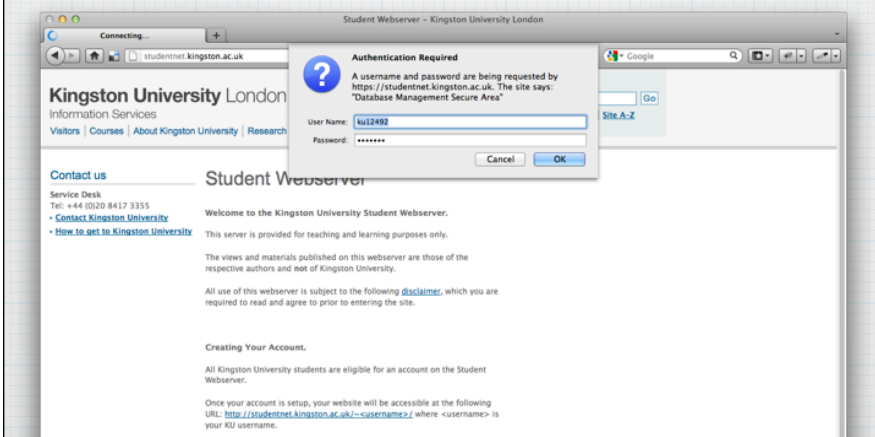
mysql - dedicated server

Use whatever interface is supplied to create a DB



mysql - dedicated server

Use whatever interface is supplied to create a DB



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 Studentnet Web Server.

Create MySQL Database

Database Name:

Username:

Database Password:

Confirm Password:

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

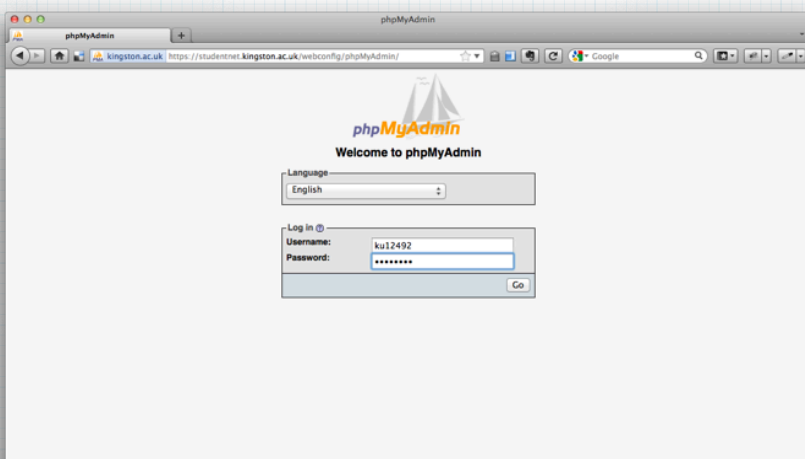
You can now:

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

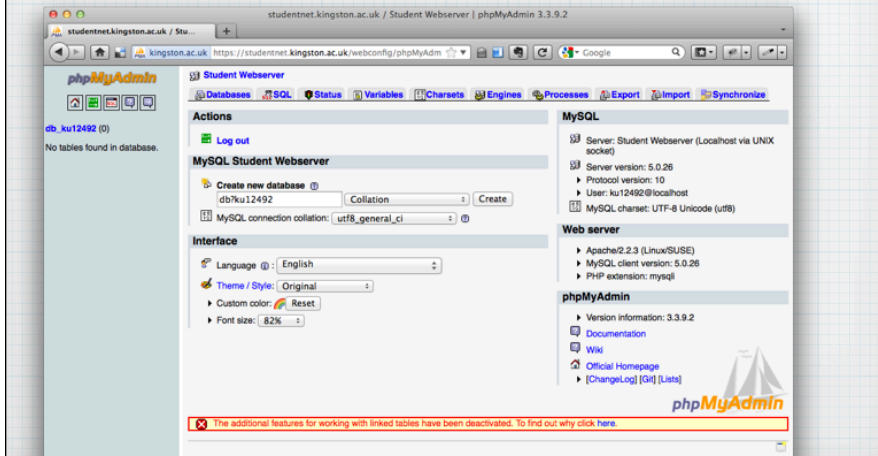
phpMyAdmin

Use a web interface to setup DBs, tables



phpMyAdmin

Use a web interface to setup DBs, tables



phpMyAdmin - database setup at kingston

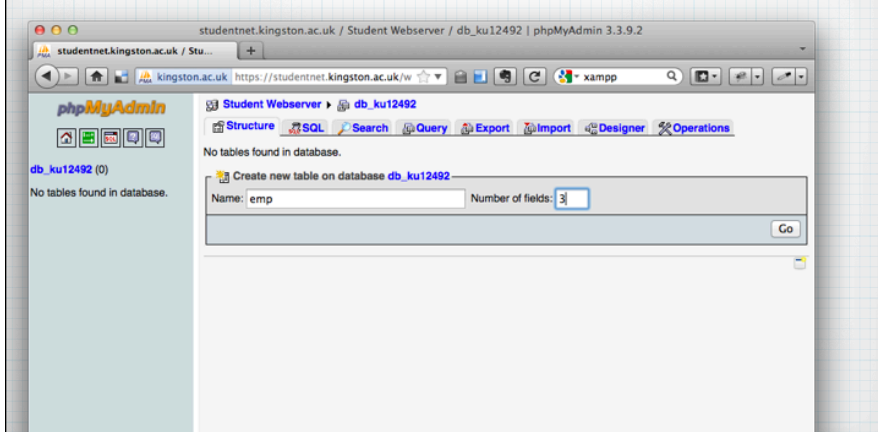
One database created by the setup process

db_kxxxxxxxxx

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

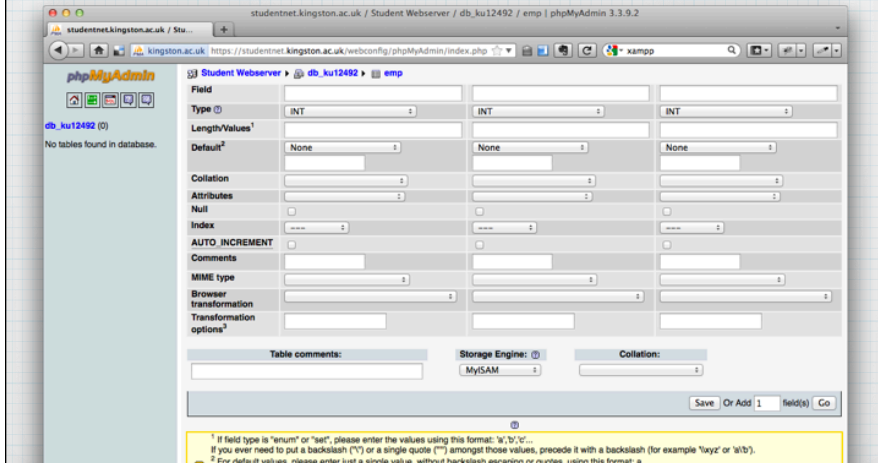
To create a table

Click on structure, name table and number of fields

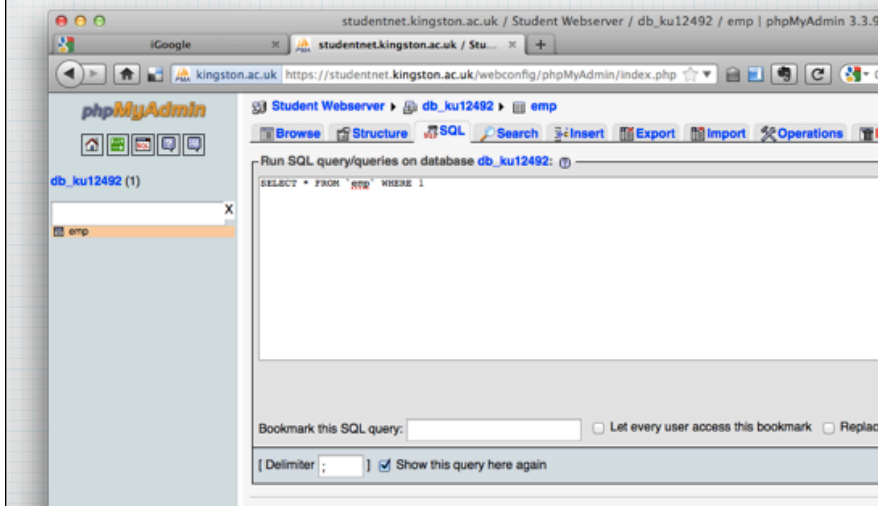


To create a table

Add fields ...



Alternative - creating a table using the DDL/SQL



Alternative - creating a table using the DDL

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)
```

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

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

sql - querying the database using the DQL

Syntax

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

Here used to show all rows in emp

```
select * from emp;
```

```
+-----+-----+-----+
| empno |  ename |   sal   |
+-----+-----+-----+
|  120  | SMITH  | 22000  |
+-----+-----+-----+
1 row in set (0.00 sec)
```

sql - running many SQL lines

SQL statements 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);
```

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
$link = mysql_connect('example.com:3307', 'user5', 'qwerty5');
if (!$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 - mamp

```
<?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

Kingston version

To connect to the local mysql database - mamp

```
<?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
// 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 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

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

mysql_close($link);
?>
```

Demo

Kingston version

To open the **empdb** database

```
<?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

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**

```
$result = mysql_query('insert into foo (field1, field2, field3) values (120,
2000, 22000)');
if (!$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

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`

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

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

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

	empno	ename	sal
<code>\$row</code>	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

	empno	ename	sal
<code>\$row</code>	121	JONES	23000

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

empno	ename	sal
122	AVERY	22500

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

empno	ename	sal
124	MITCHELL	26000

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

empno	ename	sal
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

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');

....

mysql_close($link);
?>
```