

## PHP / MYSQL - PART ONE

### INTRODUCTION

If you haven't already done so create a MySQL user account. Access the phpmyadmin application at:

<http://homepages.shu.ac.uk/mysql>

### PHPMYADMIN

In phpMyAdmin create yourself a table called 'movies'. Add fields to the table as follows:

	Browse	Structure	SQL	Search	Insert	Export	Import	Operations	
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
<input type="checkbox"/>	1 <b>filmID</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop	
<input type="checkbox"/>	2 <b>filmCertificate</b>	varchar(5)	utf8_unicode_ci		No	None		Change  Drop	
<input type="checkbox"/>	3 <b>filmName</b>	varchar(70)	utf8_unicode_ci		No	None		Change  Drop	
<input type="checkbox"/>	4 <b>filmDescription</b>	text	utf8_unicode_ci		No	None		Change  Drop	
<input type="checkbox"/>	5 <b>filmImage</b>	varchar(50)	utf8_unicode_ci		No	None		Change  Drop	
<input type="checkbox"/>	6 <b>filmPrice</b>	decimal(5,2)			No	None		Change  Drop	
<input type="checkbox"/>	7 <b>filmReview</b>	int(11)			No	None		Change  Drop	
<input type="checkbox"/>	8 <b>releaseDate</b>	date			No	None		Change  Drop	

Download the files from Blackboard or Github . Populate your *movies* table with data from the zip in the *data/movies.csv* file via the 'import' tab.

Once you have the data in MySQL use the export tab to create an SQL file backup of your data.

Take time to look at the data. Knowing your data is extremely useful when building an application.

### CONNECTING TO MYSQL

Open the PHP file in the 'includes' folder called *conn.inc.php*. You will need to edit this file to match your own database name, username and password. For the purposes of this module you should only need to create this file once and then reference it in each of the pages on `public_html` that requires database access.

## CREATING A SIMPLE QUERY - SINGLE RECORD

Open the file *select-single.php*. Use this to build a simple query of your data to extract one unique value. First you need to include the connection file.

```
<?php
require_once('../includes/conn.inc.php');
?>
```

As the include file contains the login credentials for the database this should be moved above the public visible directory thus the `../includes` path above.

- Build a SQL query.
- Use the mysqli extension to send your query to MySQL.
- Create an array to hold the results of your query.

Your code should now appear:

```
<?php
require('../includes/conn.inc.php');
$queryFilms = "SELECT filmName, filmDescription FROM movies
WHERE filmID = 10";
$resultFilms = $mysqli->query($queryFilms);
$rowFilms = $resultFilms->fetch_assoc();
?>
```

In the body of the HTML file now use `echo` to output the value of *filmName* within the `<section>` element.

```
<?php
echo "<h2>{$rowFilms['filmName']}</h2>";
?>
```

Amend the above to output the *filmDescription* field in a `<p>` element underneath the *filmName*.

## CREATING A SIMPLE QUERY - MULTIPLE RECORDS

Open the file *select-multiple.php*. Use this to build a simple query of your data to extract multiple records. First you need to include the connection file.

```
<?php
require('../includes/conn.inc.php');
?>
```

- Build a SQL query.
- Use the mysqli extension to send your query to MySQL.

Your code should now appear:

```
<?php
require('../includes/conn.inc.php');
$queryFilms = "SELECT * FROM movies";
$resultFilms = $mysqli->query($queryFilms);
?>
```

In the `<section>` element of the HTML file now use a `while` loop to `echo` around each value of `filmName` from the associate array produced by `fetch_assoc()`;

```
<ul>
<?php
while($rowFilms = $resultFilms->fetch_assoc()){
    echo "<li>{$rowFilms['filmName']}</li>";
}
?>
</ul>
```

## CREATING A SEARCH/RESULTS PAGE - SINGLE RECORD

Open the file *prepare-single.php*. Here you will find a HTML form. In this file build a prepared statement to query of your data based on user input from a form.

- Add your connection include
- Build a `prepare` statement
- Bind a `$_GET` value to the query
- Execute the query and bind the results
- Fetch the results and close the query.

Your code should now appear:

```
<?php
require('../includes/conn.inc.php');
$stmt = $mysqli->prepare("SELECT filmName FROM movies WHERE
filmID = ?");
$stmt->bind_param('i', $_GET['filmID']);
$stmt->execute();
$stmt->bind_result($filmName);
$stmt->fetch();
$stmt->close();
?>
```

This code queries the database by the *filmID* field which is the primary key in the *movies* table.

In the body of the HTML file use conditional logic to see if the form has been submitted and then output the results of the query.

```
<?php
if(isset($_GET['filmID'])) {
    echo "<ul>";
    echo "<li>{$filmName}</li>";
    echo "</ul>";
}
?>
```

## CALCULATING THE NUMBER OF RETURNED RECORDS

If no record is found we want to display a message. To do so we will use the `num_rows` property of the `Mysqli` object.

Amend the PHP code that performs the query as follows:

```
<?php
require('../includes/conn.inc.php');
$stmt = $mysqli->prepare("SELECT filmName FROM movies WHERE
filmID = ?");
$stmt->bind_param('i', $_GET['filmID']);
$stmt->execute();
$stmt->bind_result($filmName);
$stmt->store_result();
$numRows = $stmt->num_rows;
$stmt->fetch();
$stmt->close();
?>
```

The value of the `num_rows` object has been stored in the variable `$numRows`.

Amend your code to use conditional logic to display a message if no records match the primary key used.

## DISPLAYING DATES

Amend the SQL to extract the `releaseDate` field from the database. Your code should appear as follows:

```
<?php
require('../includes/conn.inc.php');
$stmt = $mysqli->prepare("SELECT filmName, filmDescription,
releaseDate FROM movies WHERE filmID = ?");
$stmt->bind_param('i', $_GET['filmID']);
$stmt->execute();
$stmt->bind_result($filmName, $filmDescription, $releaseDate);
$stmt->store_result();
$numRows = $stmt->num_rows;
$stmt->fetch();
$stmt->close();
?>
```

If the `'releaseDate'` is echoed then the default format of `YYYY-MM-DD` is used.

To output the date in a more user friendly format use the PHP `date()` method.

```
if(isset($_GET['filmID'])) {
    $timestampDate = strtotime($releaseDate);
    $displayDate = date("D d M Y", $timestampDate);
    echo "<ul>";
    echo "<li>{$filmName} - {$displayDate}</li>";
    echo "</ul>";
}
```

## CREATING A SEARCH/RESULTS PAGE - MULTIPLE RECORDS

Open the file *prepare-multiple.php*. Here you will again find a HTML form. In this file build a prepared statement to query the database based on user input from a form.

- Add your connection include
- Build a prepare statement
- Create a variable that concatenates the SQL wildcard characters to the value received from the form
- Bind your newly created value to the query
- Execute the query and bind the results

In the body of the HTML file use conditional logic to see if the form has been submitted. Then use a while loop with the `mysqli_fetch()` method to retrieve and echo the results of the query. Output the film's name and release date, formatting the release date using the PHP `date()` function.