# Internet Systems Development

# PHP and MySQL

## Introduction

The aim of the following exercises is to familiarise you with the use of MySQL as a data store that can be connected to your web site. You will be using the MySQL database that is already installed on the student server or on your local Xampp installation. To access and work with your database you will use a tool called phpMyAdmin. You can open phpMyAdmin in a browser window at the following addresses:

* Student server: <http://www.inn.leedsmet.ac.uk/phpMyAdmin> You will be prompted with a login box. If you have any problems logging in then ask for a password reset at the CA212 helpdesk.
* Local Xampp server: localhost/xampp/ Click on the phpMyAdmin link under tools in the left hand navigation panel.

### Resources

* Reading List
* ISD Lecture Series
* W3 Schools introduction to MySQL <http://www.w3schools.com/php/php_mysql_intro.asp>
* thePHPBasics Videos <http://thephpbasics.com/mysql-video-tutorials/>

Note: this is a different set of freely available videos. The phpAcademy also has similar videos to introduce MySQL but these have a different presenter and PD found the style less engaging. ThePHPBasics videos can also be found on You Tube.

* Zilmat Database Interaction <http://zilmat.com/int/foundations/database/selection.php>

## Exercise 1 – Create a table

1. Watch the The PHPBasics videos on an Introduction to MySQL and Database Creation, look at the Basic Data and Retrieval examples on Zilmat.
2. Open phpMyAdmin in the left hand panel on the student server click on the Databases dropdown and select the database that has your ID. If you are using Xampp you will need to create a database, and you should again use your ID for the name.
3. Create a table called Customer with 7 fields. Populate that table using the information below and click save. Note: the columns found on phpMyAdmin, but not shown below, are used for more advanced features of MySQL and will not be used for now, but feel free to do your own research.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Null | Index | Extra |
| CustomerID | INT | 8 | Not Null | Primary | Auto-Increment |
| FirstName | VARCHAR | 20 | Not Null |  |  |
| LastName | VARCHAR | 20 | Not Null |  |  |
| Email | VARCHAR | 50 | Not Null |  |  |
| Password | VARCHAR | 10 | Not Null |  |  |
| Gender | VARCHAR | 1 | Null |  |  |
| Age | INT | 3 | Null |  |  |

1. At present your table does not have any records. Click on the **Insert** tab and complete the **Value** column for a single record:

1, Jenny, Brown, [j.brown@leedsmet.ac.uk](mailto:j.brown@leedsmet.ac.uk), brown01, F, 23

Click **Go**. This will run an insert command and you will be shown the successful SQL statement. You can edit this statement to insert another record. Modify the statement to insert:

2, Ray, Green, [r.green@leedsmet.ac.uk](mailto:r.green@leedsmet.ac.uk), green01, M, 19

Click Go. Click the browse tab and view your two records.

1. If you are able to write SQL script you can simply run this in phpMyAdmin rather than use the wysiwyg interface. To complete your user table, click on the **SQL** tab and paste the SQL code below into the text box and click **Go**.

INSERT INTO `Customer` (`CustomerID`, `FirstName`, `LastName`, `Email`, `Password`, `Gender`, `Age`) VALUES ('3', 'Arjun', 'Patel', 'a.patel@leedsmet.ac.uk', 'patel01', 'M', '20');

INSERT INTO `Customer` (`CustomerID`, `FirstName`, `LastName`, `Email`, `Password`, `Gender`, `Age`) VALUES (4, 'Steve', 'Rich', 's.rich@leedsmet.ac.uk', 'rich01', 'M', '27');

INSERT INTO `Customer` (`CustomerID`, `FirstName`, `LastName`, `Email`, `Password`, `Gender`, `Age`) VALUES (5, 'Amy', 'Park', 'a.park@leedsmet.ac.uk', 'park01', 'F', '20');

INSERT INTO `Customer` (`CustomerID`, `FirstName`, `LastName`, `Email`, `Password`, `Gender`, `Age`) VALUES ('6', 'Rehana', 'Hashmi', 'r.hashmi@leedsmet.ac.uk', 'hashmi01', 'F', '22');

Browse your table to confirm that you now have 6 records.

1. Take care with this next step. It is useful to know how to mirror your site, this is easy for HTML and PHP files as they simply need copying to a new location, but a database must be exported and then imported. To experiment with this you will export the table you have currently, then delete your table and rebuild it with the exported code.

With your Customer table open, click the **Export** tab. Click **Go** (if given the option then chose **Custom** and set the **Output** option to **View as Text**). The SQL code to create the table will be displayed, copy this code into a text file and save it as customer.sql.

Click on the database in the breadcrumb navigation then click **Drop** (or **X** in some versions of phpMyAdmin) to remove your table. Confirm that the table has gone. Click on the **SQL** tab and paste a copy of customer.sql into the window and click **Go**. You should have your table back.

Take some time to look at customer.sql and take note of the code structure. You could potentially edit this and use it to create a different table.

## Exercise 2 – INSERT records from a form

1. To display records from a database you need to establish a connection to that database using PHP. You should watch the ThePHPBasics videos on Connecting to a Database and MYSQL Insert, plus review the Basic Data Retrieval and Display examples on Zilmat. You will note that they use slightly different approaches to the connection; you should produce your own connection script using the MySQLi technique used on Zilmat. The script should connect to your ‘localhost’ server using your login details (these will be user=’root’ password=’’ for an Xampp installation or your standard login details for the student server). You should select the cXXXXXX database. Save your connection script as connection.php and run it in the browser to confirm no error is generated. Introduce an error into your login details temporarily and run the script again. Correct your script.
2. Create a new HTML page and before the HTML code open some PHP tags and include your connection script. Call the page customer.php.

Between the body tags open up some more PHP tags and using the mysqli\_query() function, write code to INSERT a new record into your customer table. To help with the syntax of the SQL statement you could revisit your customer.sql file. Choose your own customer details.

Edit the INSERT statement in your code to add yet another customer, but remove the reference to CustomerID and do not include a value for the CustomerID. When you run this file your new record should be added, with the ID being auto incremented.

This has merely been an experiment what we really want to do is to INSERT a new record using a form. Remove all the PHP code from customer.php and add a form with boxes to accommodate each field except customerID. Set the method for the form to POST and the action to dbwrite.php.

1. The overall structure of the dbwrite.php file which you will produce is given below. After that are some detailed instructions on the steps to take.

**<?php**

//Make connection to database

//Gather from $\_POST[]all the data submitted and store in variables

//Construct INSERT query using variables holding data gathered

//Temporarily echo $query for debugging purposes

//run $query

**?>**

1. Add code to collect all the information that is and held in the $\_POST[] array after being submitted by the form in customer.php. Save each element in the $\_POST[] array to appropriately named variables. Declare another variable called $query and make it equal to the INSERT statement. The line of code will be of the form below:

$query="INSERT INTO customer (field1, field2, …..) VALUES ('$var1', '$var2', …..)";

Take note that the variables in the query are declared inside quotes.

Add a line to echo $query to the screen then run customer.php and submit the form. You should see the query that has been put together by your code, this is a useful debugging technique that you can use at any time to check the queries that you build. If all is looking good then comment out your echo line of code.

Finally add a line of code to run the query using the mysqli\_query() function. Test your code by entering information into the form, submitting and checking your database to see if it has been added.

1. When implementing such a system for real, you would need to apply all the validation and security techniques learnt previously to data entered into the form, before constructing a query.

## Exercise 3 – Displaying records

In the following exercises you will explore the use of the SELECT, WHERE and ORDER BY statements in SQL. You should watch the ThePHPBasics videos for Database Select, Database Where and Database Order By. Also refer to the examples on Zilmat for Database Integration.

1. Add PHP code just after your form in customer.php, to include a file called select.php. Start select.php, the initial structure of which is shown below:

**<?php**

//Make connection to database

//Display heading

//run query to select all records from customer table

//store the result of the query in a variable called $result

//Use a while loop to iterate through your $result array and display

//the first name, last name and email for each record

**?>**

1. You will note that ThePHPBasics video uses a function mysql\_fetch\_array(), whilst Zilmat uses mysqli\_fetch\_assoc(). You can do your own research on the difference between \_array and \_assoc, you should use the approach used on Zilmat.
2. Run customer.php and the output below the form should be something like you see below:



1. You should now repeat the structure above four times (except the connection) but using a different query each time. The four queries should be to select records defined below:
   * + SELECT records only WHERE age is greater than or equal to 22
     + SELECT records WHERE females have an age greater than or equal to 22
     + SELECT all records WHERE male and ORDER BY age descending (DESC)
     + Use LIKE to SELECT all records with an ‘a’ in the first name, ORDER BY age DESC and LIMIT the number of records displayed to 3.

You will need to do a little research beyond ThePHPBasics and Zilmat, but the clues are in the use of CAPITALS.