Unit 6G4Z2101: Introduction to Web Design and Development

Laboratory Instructions - the MySQL tasks

**Important: For the final unit tests you are permitted to take in these instructions AND any notes you make on this laboratory sheet. You MUST also have the most up to date version of your laboratory attempts available at the final tests.**

**1.Learning materials**

The recommended text for this part of the unit is:   
  
"Learning PHP, MySQL and Javascript: A step by step guide to creating dynamic websites" by Robin Nixon and published by O'Reilly.

The key website for MySQL development and documentation is here: http://dev.mysql.com/   
  
What follows is an exercise in database management and its integration with a website.

A simple database is provided and once this is working you then have the task of inserting more data into the database. This will involve the use of SQL, PHPMyadmin and a spreadsheet. A detailed knowledge of SQL is not required but you will have to be careful about the data format.

Website administrators are often required to add a large number of new users to a website without going through a self-authentication process.

This laboratory has two stages, the first is one of creating and modifying a database (PHPMyadmin and a spreadsheet program are instruments of choice) and the second stage where PHP is used to talk to the database and drive a webpage.

**2.The PHPand MySQL web page**

All PHP,HTML and mysql should be contained in a single file called

"YourNameMySQL.php"..

e.g “FredBlogsMySQL,php”

At the end of the unit you will be required to upload this file – hence the file name structure.

**Always store a copy of your most recent work on your student area. In this way you have access to it and you will need access to it during the final tests.**

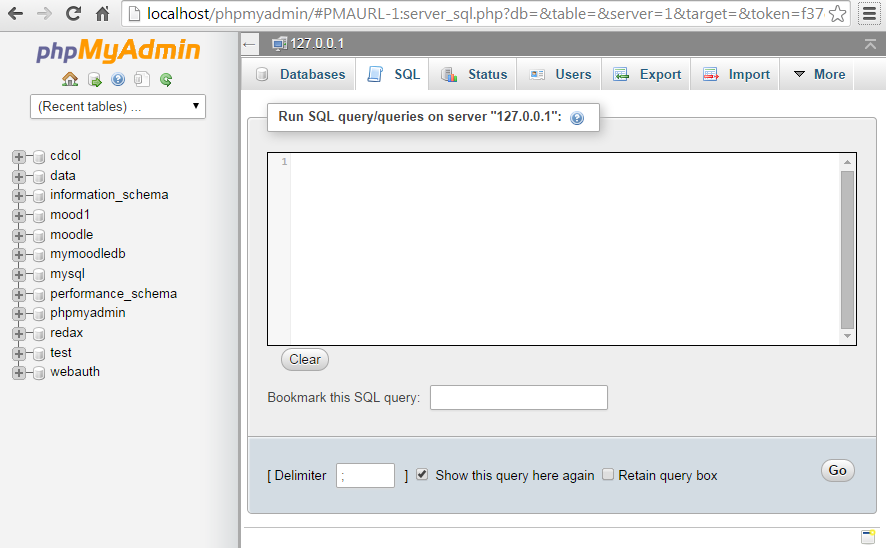
You will need to be familiar with a number of PHP and MySQL commands. These are listed below.

**3.Your Notes – you need these for the final unit test, make them good**

|  |  |
| --- | --- |
| **Commands** | Notes on the use of the |
| **PHP and MySQL** |  |
| $servername = "localhost";  $username = "username";  $password = "password";  // Create connection  $conn = mysqli\_connect($servername, $username, $password);  // Check connection  if (!$conn) {  die("Connection failed: " .  mysqli\_connect\_error()); | What does this code snippet do? |
| $sql = "INSERT INTO MyDataBase (firstname, lastname, email)  VALUES ('John', 'Doe', 'john@example.com')";  if (mysqli\_query($conn, $sql)) {  echo "New record created successfully";  } else {  echo "Error: " . $sql . "<br>" . mysqli\_error($conn); | This set of code embeds MySQL commands into PHP. What is the result of executing this code (once connected to a database)?  What is the SQL command which is actually executed on the database? |
| Mysqli\_insert\_id($conn) | What does this do? Assume $conn is the connection to the database as indicated above. |
| **MySQL** |  |
| SELECT column\_name(s) FROM table\_name | What does this do? How can this be used through PhPMyAdmin? |
| SELECT id | What does this do? How can this be used through PhPMyAdmin? |
| DELETE FROM table\_name  WHERE some\_column = some\_value | What does this do? |
| UPDATE table\_name  SET column1=’value’,  WHERE id=3 | What does this do? How can this be used through PhPMyAdmin? |
| REGEX |  |
| COUNT |  |
| WHERE xxxx LIKE |  |
| Other commands found useful |  |
| Other commands found useful |  |

**4. Setting up the database**

Create a database "lab2" and then populate it using the SQL given below. This can by done either through the consol or (more easily) by using phpMyAdmin directly. Look for the ”SQL” tab in the phpMyAdmin tool. Copy and paste the SQL into the box and click “Go”.



NOTE: If you are doing this task using an Apple Mac you may find that it will not work as it is (which creates the table an inserts the data in one go). If this happens, you can create the table manually and then run an ‘insert’ script to populate the table with the data.

-- phpMyAdmin SQL Dump  
-- version 3.3.9  
-- http://www.phpmyadmin.net  
-- SQL Code used to create the laboratory 2 database  
  
SET SQL\_MODE="NO\_AUTO\_VALUE\_ON\_ZERO";  
  
/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  
/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  
/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  
/\*!40101 SET NAMES utf8 \*/;  
--  
-- Database: `lab2`  
--  
-- --------------------------------------------------------  
--  
-- Table structure for table `lab2`  
--  
  
CREATE TABLE IF NOT EXISTS `lab2` (  
  `id` int(4) NOT NULL AUTO\_INCREMENT,  
  `category` varchar(10) NOT NULL DEFAULT '',  
  `name` varchar(50) NOT NULL DEFAULT '',  
  `username` varchar(50) NOT NULL DEFAULT '',  
  `regnum` varchar(50) NOT NULL DEFAULT '',  
  `password` varchar(50) NOT NULL DEFAULT '',  
  KEY `id` (`id`)  
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO\_INCREMENT=14 ;  
  
--  
-- Dumping data for table `lab2`  
--  
INSERT INTO `lab2` (`id`, `category`, `name`, `username`, `regnum`, `password`) VALUES  
(1, 'guest', 'Chung Kristina H', '32422', '1', '2c2c5fd01b61e3e0e687573af8f7e1fa'),  
(2, 'guest', 'Chen Paige H', '19731', '2', 'd5a92a68f9af953b26eafb935a054ce7'),  
(3, 'admin', 'Melton Sherri E', '7040', '3', '1def1713ebf17722cbe300cfc1c88558'),  
(4, 'guest', 'Hill Gretchen I', '27117', '4', '9aea3c940f8665926c97b3d3c64ace44'),  
(5, 'user', 'Puckett Karen U.', '14426', '5', 'fc5f86251458722c799d1830fa0c2c1f'),  
(6, 'user', 'Song Patrick O.', '1735', '6', 'c4492cbe90fbdbf88a5aec486aa81ed5'),  
(7, 'user', 'Hamilton Elsie A.', '21812', '7', '82aafca3e03771835c2e1acb19fde2d4'),  
(8, 'user', 'Bender Hazel E.', '9121', '8', '94fee470b43270a912c27d56c27b3211'),  
(9, 'moderator', 'Wagner Malcolm A.', '29198', '9', 'df584d514eb49beffdd8b37311204fc9'),  
(10, 'moderator', 'McLaughlin Dolores C.', '16507', '10', '800502ddc965d56e79f6545ee7c75f50'),  
(11, 'user', 'McNamara Francis C.', '3816', '11', 'c911241d00294e8bb714eee2e83fa475'),  
(12, 'moderator', 'Raynor Sandy A.', '23893', '12', '1f3ecb87f576752202975d3e0b868bdc'),  
(13, 'moderator', 'Moon  Marion O.', '11202', '13', 'fe998b49c41c4208c968bce204fa1cbb');

Examine the table using phpMyAdmin . Observe that the "id" and "regnumber" are the same for each user. The username (actually a number) is derived from seeding "rand" with the "regnum". The code snippet "idgenerator.php" shown below recreates the first thirteen usernames and passwords.

**5.Task: Manipulating data to a format which can be imported into the database. You need this for the final test**

A very common problem for database administrators is the need to covert data from one format to another. e.g a lot of company data may be stored in spreadsheets but – it is necessary to have all the data in a database .So a transfer has to take place. Reformatting data to import it in to the data base requires a number of skills. The following task will develop those skills.

**The task**: You have already put the first 13 names into the “lab2” database. You now discover that in fact, one thousand names have to be put in. Furthermore. When the system was setup, someone decided to:

1] Identify a “username” for each person with a number which itself was generated from using the “id” number found in the database to “seed” the random number generator “rand”.

2] The password for each person was then determined by taking the “md5” hash of the “username”.

Why would anyone do this? The answer is that in this manner every user gets a unique username (and hence password). The difficulty is that a separate PHP script is needed to create the “usernames” (actually a number) and passwords for all users.

The PHP script for producing the first 13 usernames is provided below. Get this running on your WAMP and then relate the output (seen on a browser) to the 13 people you have in your database “lab2”.

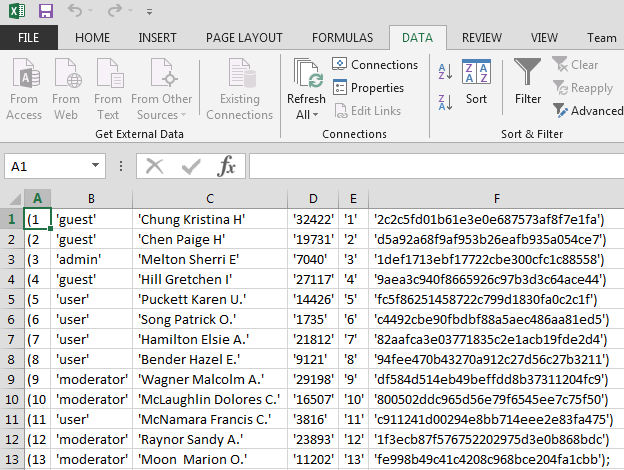
<?php   
//creates a unique id  username(renum)  and password using rand and md5  
  
  
//creates a longer unique id with the 'about' prefix  
$a=0;  
echo " id &nbsp &nbsp regnum &nbsp &nbsp password <br>";  
while($a<13){  
$a=$a+1;  
srand($a);  
$b = rand();  
$c = strval($b);  
$d = md5($c);  
echo $a, "&nbsp &nbsp &nbsp",$b,"&nbsp &nbsp &nbsp", $d, "<br>";  
echo "<br>";  
}

Additionally, each new user must be registered as a "guest" under "category"

1] Alter the “username/password” generation script above to generate all 1000 usernames and passwords.

2] use a spreadsheet to “merge” the 1000 names found in “namesfilewrongformatfordatabase.txt” with the output from the PHP script above. The PHP output is a web page, but the output can be cut and pasted into notepad++, stored, and then imported into the spreadsheet.

3] Once you have all 1000 names correctly formatted in the spreadsheet it should look like this:



3] (cont) At this point you can then export the data in the spreadsheet into a text file.

4] Using Notepad++ open up the SQL code provided above which created lab2. Replace the SQL which follows after this line:

INSERT INTO `lab2` (`id`, `category`, `name`, `username`, `regnum`, `password`) VALUES

With a complete list of all 1000 students.

5] You should now have a modified version of the SQL command that instead of placing 13 names into a database called “lab2”, puts in 1000 names.

6] Before you execute this new SQL command with 1000 names, Drop (i.e delete) the existing “lab2” database using phpMyAdmin.

7] When you execute your SQL successfully, you should have a new database with 1000 names, usernames and passwords in it.

Populating the database is a substantial task. It should take you at least one session to get this component of the laboratory completed.

**6. The Laboratory task for MySQL  - four weeks**

When the database is set up with all one thousand names you will need to create PHP code which connects to and interrogates that database

Use the table structure outlined in the PHP Laboratory (5 rows 2 columns -ten rectangles in total) and serve your pages from an "WAMP" development system. **YOU MUST USE MySQL command through PHP to populate the rectangles and NOT do it manually.**

Apart from 1, 2 and 6 each rectangle on the webpage should be derived from a PHP script making an SQL call to the database containing all 1000 names, regnumbers and passwords.

**As in the PHP laboratory one, a web page should be built around a table structure (5 rows and 2 columns). The following should appear in each rectangle in the table. These need to be working for the final test.**

|  |  |
| --- | --- |
| Rectangle 1: Your name and student registration number and CURRENT date and t The version number of Mysql you have using "mysql\_get\_server\_info" | Rectangle 2: .The id number of the person with the ‘username’ of 13022 |
| Rectangle 3: .The id number of the person with the ‘password’ of:  5f4f7141b65a730b4efb0e0d51f63e94 | Rectangle 4: The password of  McKee Heidi C. |
| Rectangle 5: The list of people who have a username which has a numeric value of less than 100 | Rectangle 6: The person who has the following string as **part** of their password (the middle part)  b42c3e0713278dd |
| Rectangle 7: A list of all the “Kelly”s in the database | Rectangle 8: The number of “Penny”s in the database |
| Rectangle 9 : The number of people with a ‘username’ less than 600. | Rectangle 10: The number of people with a ‘username’ of between 700 and 1500. |

**All PHP,HTML and mysql should be contained in a single file called**

**"YourNameMySQL.php"..**

**e.g “FredBlogsMySQL,php”**

**At the end of the unit you will be required to upload this file – hence the file name structure.**

**Always store a copy of your most recent work on your student area. In this way you have access to it and you will need access to it during the final tests.**