CSE417: WEB ENGINEERING

Daffodil International University

Learning Outcomes

You will learn

- SQL
- To access mySQL database
- To create a basic mySQL database
- To use some basic queries
- To use PHP and mySQL
- Session & Cookie

Introduction to SQL

SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating databases.

- SQL stands for Structured Query Language
- using SQL can you can
  - access a database
  - execute queries, and retrieve data
  - insert, delete and update records
- SQL works with database programs like **MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, mySQL**, etc.

Unfortunately, there are many different versions. But, they must support the same major keywords in a similar manner such as SELECT, UPDATE, DELETE, INSERT, WHERE, etc.

Most of the SQL database programs also have their own proprietary extensions!

The University of Liverpool CS department has a version of mySQL installed on the servers, and it is this system that we use in this course. Most all of the commands discussed here should work with little (or no) change to them on other database systems.

SQL Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

For example, a table called "Persons":

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

The table above contains three records (one for each person) and four columns (**LastName, FirstName, Address, and City**).

SQL Queries

With SQL, you can query a database and have a result set returned.

A query like this:

```
SELECT LastName FROM Persons;
```

gives a result set like this:

LastName
Hansen
Svendson
Pettersen

The mySQL database system requires a semicolon at the end of the SQL statement!

Inserting Data

Using `INSERT INTO` you can insert a new row into your table. For example,

```
mysql> INSERT INTO students
    -> VALUES(NULL,'Russell','Martin',396310,'martin@csc.liv.ac.uk');
```

```
Query OK, 1 row affected (0.00 sec)
```

Using `SELECT FROM` you select some data from a table.

```
mysql> SELECT * FROM students;
```

```
+-----+---------+--------+------------+----------------------+
| num | f_name  | l_name | student_id | email                |
+-----+---------+--------+------------+----------------------+
|   1 | Russell | Martin |     396310 | martin@csc.liv.ac.uk |
+-----+---------+--------+------------+----------------------+
1 row in set (0.00 sec)
```

Inserting Some More Data

You can repeat inserting until all data is entered into the table.

```
mysql> INSERT INTO students
    -> VALUES(NULL,'James','Bond',007,'bond@csc.liv.ac.uk');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM students;
+-----+---------+--------+------------+----------------------+
| num | f_name  | l_name | student_id | email                |
+-----+---------+--------+------------+----------------------+
|   1 | Russell | Martin |     396310 | martin@csc.liv.ac.uk |
|   2 | James   | Bond   |          7 | bond@csc.liv.ac.uk   |
+-----+---------+--------+------------+----------------------+
2 rows in set (0.00 sec)
```

Note: The value “NULL” in the “num” field is automatically replaced by the SQL interpreter as the “auto_increment” option was selected when the table was defined.

Getting Data Out of the Table

- The SELECT command is the main way of getting data out of a table, or set of tables.

  SELECT * FROM students;

  Here the asterisk means to select (i.e. return the information in) all columns.

  You can specify one or more columns of data that you want, such as

  SELECT f_name,l_name FROM students;

```
+---------+--------+
| f_name  | l_name |
+---------+--------+
| Russell | Martin |
| James   | Bond   |
+---------+--------+
2 rows in set (0.00 sec)
```

Getting Data Out of the Table (cont.)

- You can specify other information that you want in the query using the **WHERE** clause.

```
SELECT * FROM students WHERE l_name='Bond';
```

```
+-----+---------+--------+------------+---------------------+
| num | f_name  | l_name | student_id | email               |
+-----+---------+--------+------------+---------------------+
|   2 | James   | Bond   |          7 | bond@csc.liv.ac.uk  |
+-----+---------+--------+------------+---------------------+
1 row in set (0.00 sec)
```

```
SELECT student_id, email FROM students WHERE l_name='Bond';
```

```
+------------+---------------------+
| student_id | email               |
+------------+---------------------+
|          7 | bond@csc.liv.ac.uk  |
+------------+---------------------+
1 row in set (0.00 sec)
```

Updating the Table

The UPDATE statement is used to modify data in a table.

```
mysql> UPDATE students SET date='2007-11-15' WHERE num=1;
```

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM students;
+-----+---------+--------+------------+----------------------+------------+
| num | f_name  | l_name | student_id | email                | date       |
+-----+---------+--------+------------+----------------------+------------+
|   1 | Russell | Martin |     396310 | martin@csc.liv.ac.uk | 2007-11-15 |
|   2 | James   | Bond   |          7 | bond@csc.liv.ac.uk   | NULL       |
+-----+---------+--------+------------+----------------------+------------+
2 rows in set (0.00 sec)
```

Note that the default date format is “YYYY-MM-DD” and I don’t believe this default setting can be changed.

Deleting Some Data

The DELETE statement is used to delete rows in a table.

```
mysql> DELETE FROM students WHERE l_name='Bond';
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM students;
+-----+---------+--------+------------+----------------------+------------+
| num | f_name  | l_name | student_id | email                | date       |
+-----+---------+--------+------------+----------------------+------------+
|   1 | Russell | Martin |     396310 | martin@csc.liv.ac.uk | 2006-11-15 |
+-----+---------+--------+------------+----------------------+------------+
1 row in set (0.00 sec)
```

Simulation - The Final Table

We'll first add another column, update the (only) record, then insert more data.

```
mysql> ALTER TABLE students ADD gr INT;
Query OK, 1 row affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0
mysql> SELECT * FROM students;
+-----+---------+--------+------------+----------------------+------------+------+
| num | f_name  | l_name | student_id | email                | date       | gr   |
+-----+---------+--------+------------+----------------------+------------+------+
|   1 | Russell | Martin |     396310 | martin@csc.liv.ac.uk | 2007-11-15 | NULL |
+-----+---------+--------+------------+----------------------+------------+------+
1 row in set (0.00 sec)
mysql> UPDATE students SET gr=3 WHERE num=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> SELECT * FROM students;
+-----+---------+--------+------------+----------------------+------------+------+
| num | f_name  | l_name | student_id | email                | date       | gr   |
+-----+---------+--------+------------+----------------------+------------+------+
|   1 | Russell | Martin |     396310 | martin@csc.liv.ac.uk | 2007-11-15 |    3 |
+-----+---------+--------+------------+----------------------+------------+------+
1 row in set (0.00 sec)

mysql> INSERT INTO students
VALUES(NULL,'James','Bond',007,'bond@csc.liv.ac.uk','2007-11-15', 1);
. . .
. . .
```

Simulation - The Final Table (cont.)

```
. . .
. . .
mysql> INSERT INTO students VALUES(NULL,'Hugh','Milner',75849789,'hugh@poughkeepsie.ny',
    CURRENT_DATE, 2);
```

Note: **CURRENT_DATE** is a built-in SQL command which (as expected) gives the current (local) date.

```
mysql> SELECT * FROM students;
+-----+---------+---------+-----------+---------------------------+------------+------+
| num | f_name  | l_name  | student_id | email                     | date       | gr   |
+-----+---------+---------+-----------+---------------------------+------------+------+
|   1 | Russell | Martin  |     396310 | martin@csc.liv.ac.uk      | 2007-11-15 |    3 |
|   5 | Kate    | Ash     |     124309 | kate@ozymandius.co.uk     | 2007-11-16 |    3 |
|   3 | James   | Bond    |          7 | bond@csc.liv.ac.uk        | 2007-11-15 |    1|
|   4 | Bob     | Jones   |      12190 | bob@nowhere.com           | 2007-11-16 |    3 |
|   6 | Pete    | Lofton  |         76 | lofton@iwannabesedated.com | 2007-11-17 |    2 |
|   7 | Polly   | Crackers |       1717 | crackers@polly.org        | 2007-11-17 |    1|
|   8 | Hugh    | Milner  |   75849789 | hugh@poughkeepsie.ny      | 2007-11-17 |    2 |
+-----+---------+---------+-----------+---------------------------+------------+------+
7 rows in set (0.00 sec)
mysql> exit
Bye
```

PHP: MySQL Database [CRUD Operations]

To do any operations, you need to Connect to your database first!

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password,
$dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

Inserting Data

```
//Remember, you always connect to your database first
//Assumption: We have a Table named MyGuests
```

```
$sql = "INSERT INTO MyGuests (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);
}
```

Reading Data

```
//Remember, you always connect to your database first
//Assumption: We have a Table named MyGuests


$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " .
$row["firstname"]. " " . $row["lastname"]. "<br>";
}
} else {
    echo "0 results";
}
```

More..

- Did you close your connection?

```
mysqli_close($conn);
//put this at the end of operation
```

- Similarly you can update and delete data
-
- More here : PHP: MySQL Database

- There are slides hidden and supposed to help you doing more complex opertions
- These will not be covered in Class Lecture.
- Most of them are already explained to you in DATABASE MANAGEMENT SYSTEM course

Attention!

- There is easier way to do using PHPMyadmin accesses via localhost (after XAMPP installation)
- Remember, you will not always get flexibilty and,
- You love command line tools to write code!
- Cheers!

Exercise

- Show one example of each of CRUD operation
- Do two complex operations
- **READINGS/Practice**
  - M Schafer: Ch. 31
  - W3 schools
  - http://www.php-mysql-tutorial.com/
  - http://www.sitepoint.com/php-security-blunders/
  - http://php.net/manual/en/mysqli.quickstart.prepared-statements.php

Acknowledgement

- This module is designed and created with the help from following sources-
  - https://cgi.csc.liv.ac.uk/~ullrich/COMP519/
  - http://www.csc.liv.ac.uk/~martin/teaching/comp519/

Putting Content into Your Database with PHP

We can simply use PHP functions and mySQL queries together:

- Connect to the database server and login (this is the PHP command to do so)

```
mysql_connect("host","username","password");
```

- Choose the database

```
mysql_select_db("database");
```

Host:	mysql
Database:	martin
Username:	martin
Password:	<blank>

- Send SQL queries to the server to add, delete, and modify data

```
mysql_query("query");
```

(use the exact same query string as you would normally use in SQL, without the trailing semi-colon)

- Close the connection to the database server (to ensure the information is stored properly)

```
mysql_close();
```

Student Database: data_in.php

```
<html>
<head>
<title>Putting Data in the DB</title>
</head>
<body>
<?php
/*insert students into DB*/
if(isset($_POST["submit"]))  {

   $db = mysql_connect("mysql", "martin");
   mysql_select_db("martin");

   $date=date("Y-m-d");   /*  Get the current date in the right SQL format
   */

   $sql="INSERT INTO students  VALUES(NULL,'" . $_POST["f_name"] . "','" .
   $_POST["l_name"] . "'," . $_POST["student_id"] . ",'" . $_POST["email"] .
   "','" . $date . "'," . $_POST["gr"] . ")";       /*  construct the query
   */

   mysql_query($sql);   /*  execute the query  */
   mysql_close();

   echo"<h3>Thank you. The data has been entered.</h3> \n";

   echo'<p><a href="data_in.php">Back to registration</a></p>' . "\n";

   echo'<p><a href="data_out.php">View the student lists</a></p>' ."\n";

   }
```

Student Database: data_in.php

```
else  {
?>
<h3>Enter your items into the database</h3>
<form action="data_in.php" method="POST">
First Name: <input type="text" name=" f_name" /> <br/>
Last Name: <input type="text" name=" l_name" /> <br/>
ID: <input type="text" name=" student_id" /> <br/>
email: <input type="text" name=" email" /> <br/>
Group: <select name=" gr">
       <option value ="1">1</option>
       <option value ="2">2</option>
       <option value ="3">3</option>
</select><br/><br/>
<input type="submit" name=" submit" /> <input type="reset" />
</form>

<?php
   }
?>

</body>
</html>
```

view the output page

Getting Content out of Your Database with PHP

Similarly, we can get some information from a database:

- Connect to the server and login, choose a database

```
mysql_connect("host","username","password");
mysql_select_db("database");
```

  - Send an SQL query to the server to select data from the database into an array

```
$result=mysql_query("query");
```

  - Either, look into a row and a fieldname

```
$num=mysql_numrows($result);
$variable=mysql_result($result,$i,"fieldname");
```

    - Or, fetch rows one by one

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

- Close the connection to the database server

```
mysql_close();
```

Student Database: data_out.php

```
<html>
<head>
<title>Getting Data out of the DB</title>
</head>
<body>
<h1> Student Database </h1>
<p> Order the full list of students by
<a href="data_out.php?order=date">date</a>,
<href="data_out.php?order=student_id">id</a>, or
by <a href="data_out.php?order=l_name">surname</a>.
</p>

<p>
<form action="data_out.php" method="POST">
Or only see the list of students in group
<select name="gr">
  <option value ="1">1</option>
  <option value ="2">2</option>
  <option value ="3">3</option>
</select>
<br/>
<input type="submit" name="submit“ />
</form>
</p>
```

Student Database: data_out.php

```
<?php
/*get students from the DB */
$db = mysql_connect("mysql",“martin");
mysql_select_db(“martin", $db);

switch($_GET["order"]){
case  'date':    $sql = "SELECT * FROM students ORDER BY date"; break;
case  ‘student_id':      $sql = "SELECT * FROM students ORDER BY student_id";
   break;
case  ‘l_name': $sql = "SELECT * FROM students ORDER BY l_name"; break;

default: $sql = “SELECT * FROM students”;  break;
}
if(isset($_POST["submit"])){
  $sql = “SELECT * FROM students WHERE gr=“ . $_POST["gr"];
}

$result=mysql_query($sql);       /*  execute the query  */
while($row=mysql_fetch_array($result)){
  echo "<h4> Name: “ .  $row["l_name"]  . ', ‘ . $row["f_name"]  . "</h4> \n";
  echo "<h5> ID: “ .  $row[“student_id"]  . "<br/> Email: “ .  $row["email"]  .
   "<br/> Group: “ .  $row["gr"]  . "<br/> Posted: “ .  $row["date"]  . "</h5>
   \n";
}
mysql_close();
?>
</body>
</html>
```

view the output page

Using several tables

- mySQL (like any other database system that uses SQL) is a relational database, meaning that it's designed to work with multiple tables, and it allows you to make queries that involve several tables.
- Using multiple tables allows us to store lots of information without much duplication.
- Allows for easier updating (both insertion and deletion).
- We can also perform different types of queries, combining the information in different ways depending upon our needs.

Advanced Queries

- We can link these tables together by queries of this type:

```
myql> SELECT * from clients, purchases WHERE clients.client_id=purchases.client_id ORDER BY purchase_id;
+----------+---------+--------+-----------------------+---------------+----------+------------+-----------+------------+
|client_id | f_name  | l_name | address               | city          | postcode | purchase_id | client_id | date       |
-----------+---------+--------+-----------------------+---------------+----------+------------+-----------+------------+
|        1 | Russell | Martin | Dept of Computer Science | Liverpool     | L69 3BX  |          1 |         1 | 2007-11-09 |
|        1 | Russell | Martin | Dept of Computer Science | Liverpool     | L69 3BX  |          2 |         1 | 2007-11-10 |
|        2 | Bob     | Milnor | 12 Peachtree Ln       | Liverpool     | L12 3DX  |          4 |         2 | 2007-11-20 |
|        4 | Larry   | Vance  | 76 Jarhead Ln         | Liverpool     | L12 4RT  |          5 |         4 | 2007-11-20 |
|        3 | Sarah   | Ford   | 542b Jersey Rd        | West Kirby    | L43 8JK  |          6 |         3 | 2007-11-21 |
|        5 | Paul    | Abbott | 90 Crabtree Pl        | Leamingotn Spa | CV32 7YP |          7 |         5 | 2007-11-25 |
|        3 | Sarah   | Ford   | 542b Jersey Rd        | West Kirby    | L43 8JK  |          8 |         3 | 2007-11-25 |
+----------+---------+--------+-----------------------+---------------+----------+------------+-----------+------------+
7 rows in set (0.01 sec)

mysql>
```

So you can see that this query basically gives us all of the purchase orders that have been placed by the clients (but not the number of items, or the items themselves).

More Complex Queries

- We can create most any type of query that you might think of with a (more complicated) "WHERE" clause:

```
mysql> SELECT purchases.purchase_id, f_name, l_name, date
   FROM purchases, clients WHERE
   purchases.client_id=clients.client_id;
+-------------+---------+--------+------------+
| purchase_id | f_name  | l_name | date       |
+-------------+---------+--------+------------+
|           1 | Russell | Martin | 2007-11-09 |
|           2 | Russell | Martin | 2007-11-10 |
|           4 | Bob     | Milnor | 2007-11-20 |
|           5 | Larry   | Vance  | 2007-11-20 |
|           6 | Sarah   | Ford   | 2007-11-21 |
|           7 | Paul    | Abbott | 2007-11-25 |
|           8 | Sarah   | Ford   | 2007-11-25 |
+-------------+---------+--------+------------+
7 rows in set (0.00 sec)

mysql>
```

More Complex Queries (cont.)

- Find the purchases by the person named “Ford”.

```
mysql> SELECT purchases.purchase_id, f_name, l_name, date FROM
   purchases, clients
        WHERE (purchases.client_id=clients.client_id) AND
   (l_name='Ford');
+-------------+--------+--------+------------+
| purchase_id | f_name | l_name | date       |
+-------------+--------+--------+------------+
|           6 | Sarah  | Ford   | 2007-11-21 |
|           8 | Sarah  | Ford   | 2007-11-25 |
+-------------+--------+--------+------------+
2 rows in set (0.01 sec)

mysql>
```

Cookie Workings

`setcookie(name,value,expire,path,domain)` **creates cookies.**

```
<?php
setcookie("uname", $_POST["name"], time()+36000);
?>
<html>
<body>
<p>
Dear <?php echo $_POST["name"] ?>, a cookie was set on this
page! The cookie will be active when the client has sent the
cookie back to the server.
</p>
</body>
</html>
```

NOTE:
setcookie() **must appear BEFORE** <html> **(or any output) as it's part of the header information sent with the page.**

```
<html>
<body>
<?php
if ( isset($_COOKIE["uname"]) )
echo "Welcome " . $_COOKIE["uname"] . "!<br
/>";
else
echo "You are not logged in!<br />";
?>
</body>
</html>
```

`$_COOKIE`
contains all COOKIE data.

`isset()`
finds out if a cookie is set

use the cookie name as a variable

Session

- When you work with an application, you open it, do some changes, and then you close it. This is much like a Session.
  - HTTP address doesn't maintain state.
  - Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc).
  - By default, session variables last until the user closes the browser.
  - Session variables hold information about one single user
  - available to all pages in one application. ie, logged in

Example

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

Exercise

- Design a form and validate its data.
- Design a user registration system
- **READINGS/Practice**
  - M Schafer: Ch. 29-32
  - https://www.w3schools.com/php/php_form_validation.asp
  - Designing a Sign-up/Log-in page

Acknowledgement

- This module is designed and created with the help from following sources-
  - https://cgi.csc.liv.ac.uk/~ullrich/COMP519/
  - http://www.csc.liv.ac.uk/~martin/teaching/comp519/
  -