

## CSE311L : Database Systems Lab (Lab 10- PHP/MySQL)

Department of Electrical and Computer Engineering

School of Engineering and Physical Sciences

North South University, Bashundhara, Dhaka-1229, Bangladesh

---

### Objectives:

- Connecting Database with Webpage
- Insert, Edit and Delete Data Through a Webpage
- Retrieval of Data and Showing Them on a Webpage

To accomplish our lab tasks today, we have used PHP and MySQL.

Things you need to know before starting this lab:

- A basic understanding of PHP scripting language.

Tools that are needed to accomplish today's lab task are:

- XAMPP
- A Text Editor (Notepad++, Sublime, Brackets etc.)

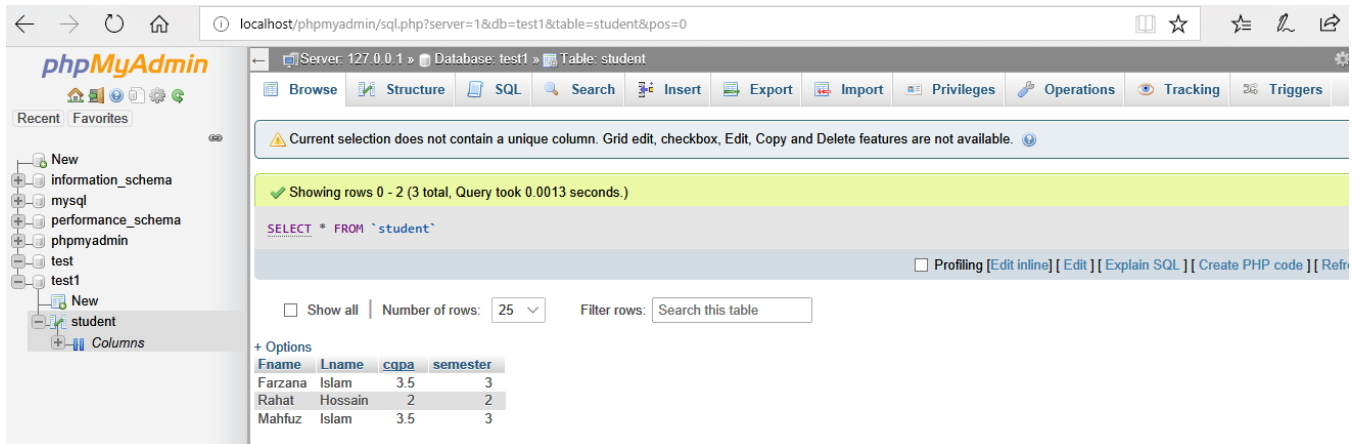
We have used some PHP functions to accomplish the tasks of inserting, deleting, editing and retrieving data. The functions are:

- `mysqli_connect()` : This function opens a new connection to the MySQL server.
- `mysqli_query()` : This function performs a query against the database.
- `mysqli_fetch_array()`: This function is used to fetch a result row as an associative array.

### **Example 1 on PHP: Insert a single record into student table through webpage**

Suppose you create a “test1” database into MySql. You also create a simple student file with four information student (Fname varchar(40), Fname varchar(40), cgpa double , semester int);

The screenshot from your XAMPP will be like this:



Let's make a SQL query using the INSERT INTO statement with appropriate values, after that we will execute this insert query through passing it to the PHP mysqli\_query() function to insert data in table. Here's an example, which insert a new row to the student table by specifying values for the first\_name, last\_name and cgpa and semester fields.

Now you write down some php file so you can insert a record directly into this table from a web browser, say Netscape, IE, Google Chrome. Your php (inset\_single.php) file will be like this:

```
<?php
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "", "test1");

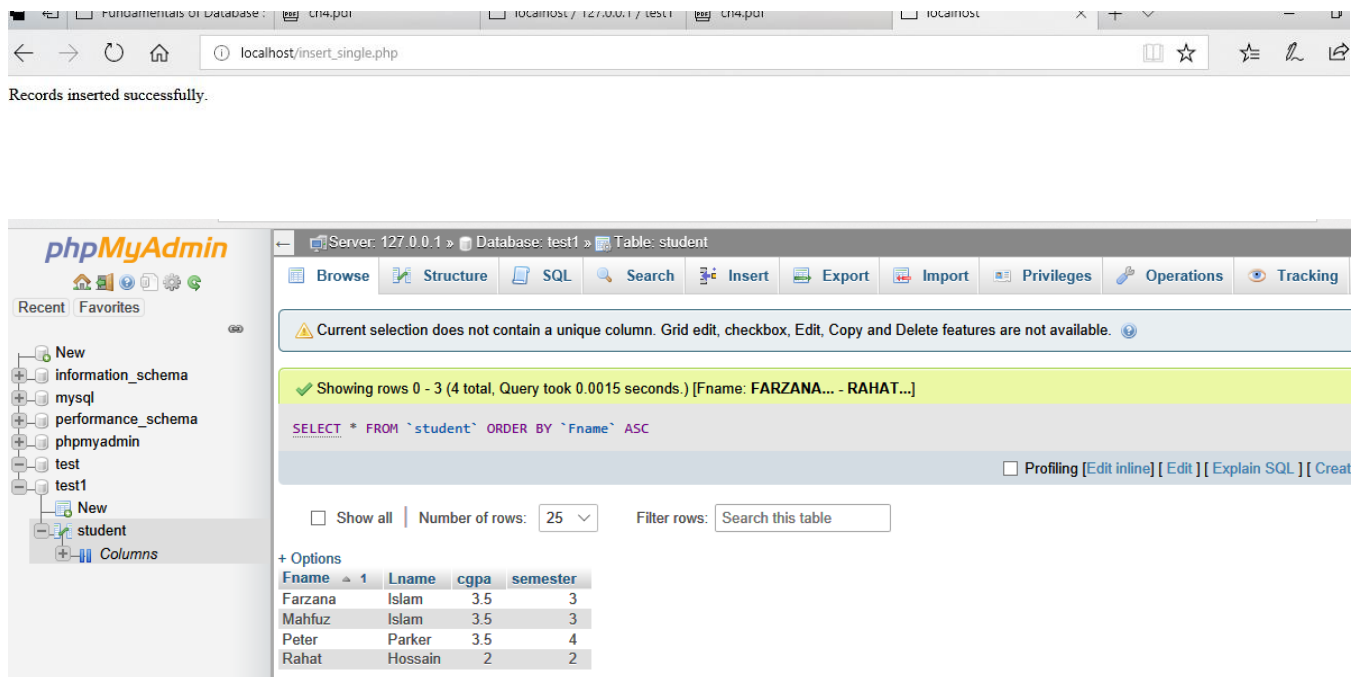
// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Attempt insert query execution
$sql = "INSERT INTO student (Fname, Lname, cgpa, semester) VALUES ('Peter', 'Parker', 3.5,4)";
if(mysqli_query($link, $sql)){
    echo "Records inserted successfully.";
}
else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
```

You will execute this from any of your web browser with the following address  
[http://localhost/insert\\_single.php](http://localhost/insert_single.php)

After running this, when you see your database again in PhpMyAdmin you will see something like this



### **Example 2: Insert multiple rows**

You can also insert multiple rows into a table with a single insert query at once. To do this, include multiple lists of column values within the INSERT INTO statement, where column values for each row must be enclosed within parentheses and separated by a comma.

### **Let's insert few more rows into the persons table, like this:**

```
• <?php
• /* Attempt MySQL server connection. Assuming you are running MySQL
• server with default setting (user 'root' with no password) */
• $link = mysqli_connect("localhost", "root", "", "demo");
•
• // Check connection
• if($link === false){
•     die("ERROR: Could not connect. " . mysqli_connect_error());
• }
•
• // Attempt insert query execution
• $sql = "INSERT INTO student (Fname, Lname, cgpa, semester) VALUES
•     ('John', 'Rambo',3.5,8),
•     ('Clark', 'Kent',4,5),
•     ('John', 'Carter',3.8,7),
```

- `('Harry', 'Potter',3,5)";`
- `if(mysqli_query($link, $sql)){`
- `echo "Records added successfully.";`
- `} else{`
- `echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);`
- `}`
- 
- `// Close connection`
- `mysqli_close($link);`
- `?>`

### **Example 3: To give input in a HTML Form**

In the previous section, we have learned how to insert data into database from a PHP script. Now, we'll see how we can insert data into database obtained from an HTML form. Let's create an HTML form that can be used to insert new records to student table.

#### **Step 1: Creating the HTML Form**

Here's a simple HTML form (input\_record\_form.html) that has three text `<input>` fields and a submit button.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Record Form</title>
</head>
<body>
<form action="insert_form.php" method="post">
    <p>
        <label for="firstName">First Name:</label>
        <input type="text" name="first_name" id="firstName">
    </p>
    <p>
        <label for="lastName">Last Name:</label>
        <input type="text" name="last_name" id="lastName">
    </p>
    <p>
        <label for="CGPA">CGPA:</label>
        <input type="text" name="cgpa" id="cgpa">
    </p>
    <p>
        <label for="Semester">Semester:</label>
        <input type="text" name="sem" id="sem">
    </p>
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

Output View of the Form:



First Name:

Last Name:

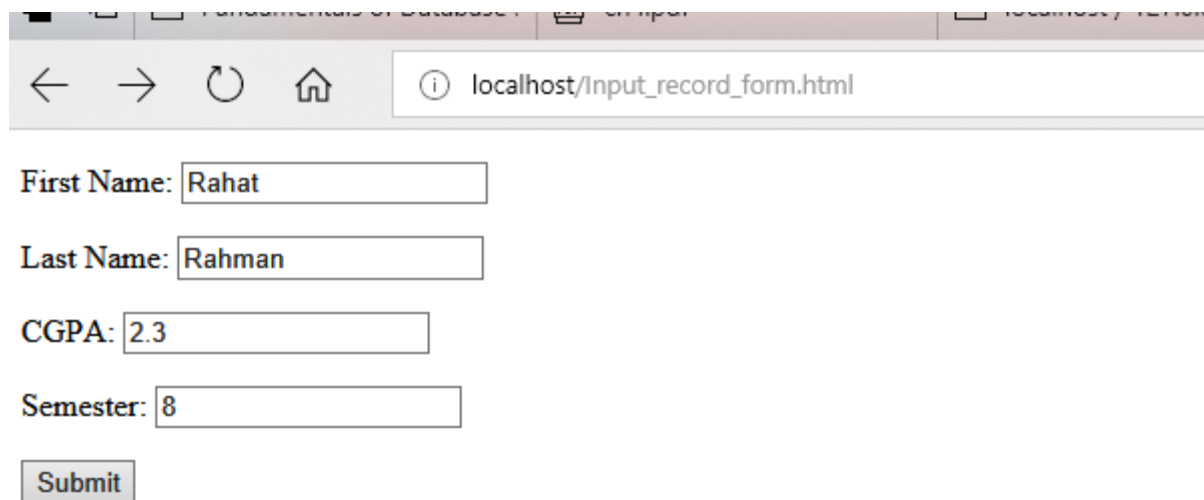
CGPA:

Semester:

## Step 2: Retrieving and Inserting the Form Data

When a user clicks the submit button of the add record HTML form, in the example above, the form data is sent to 'insert\_form.php' file. The 'insert\_form.php' file connects to the MySQL database server, retrieves forms fields using the PHP \$\_REQUEST variables and finally execute the insert query to add the records. Here is the complete code of our 'insert\_form.php' file:

**After filling up the information the data will be saved into database by (insert\_form.php)**



First Name:

Last Name:

CGPA:

Semester:

```
<?php
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "", "test1");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
```

```
// Escape user inputs for security
$first_name = mysqli_real_escape_string($link, $_REQUEST['first_name']);
$last_name = mysqli_real_escape_string($link, $_REQUEST['last_name']);
$cgpa = mysqli_real_escape_string($link, $_REQUEST['cgpa']);
$sem = mysqli_real_escape_string($link, $_REQUEST['sem']);

// attempt insert query execution
$sql = "INSERT INTO student (Fname, Lname, cgpa, semester) VALUES ('$first_name',
'$last_name', '$cgpa', '$sem')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// close connection
mysqli_close($link);
?>
```

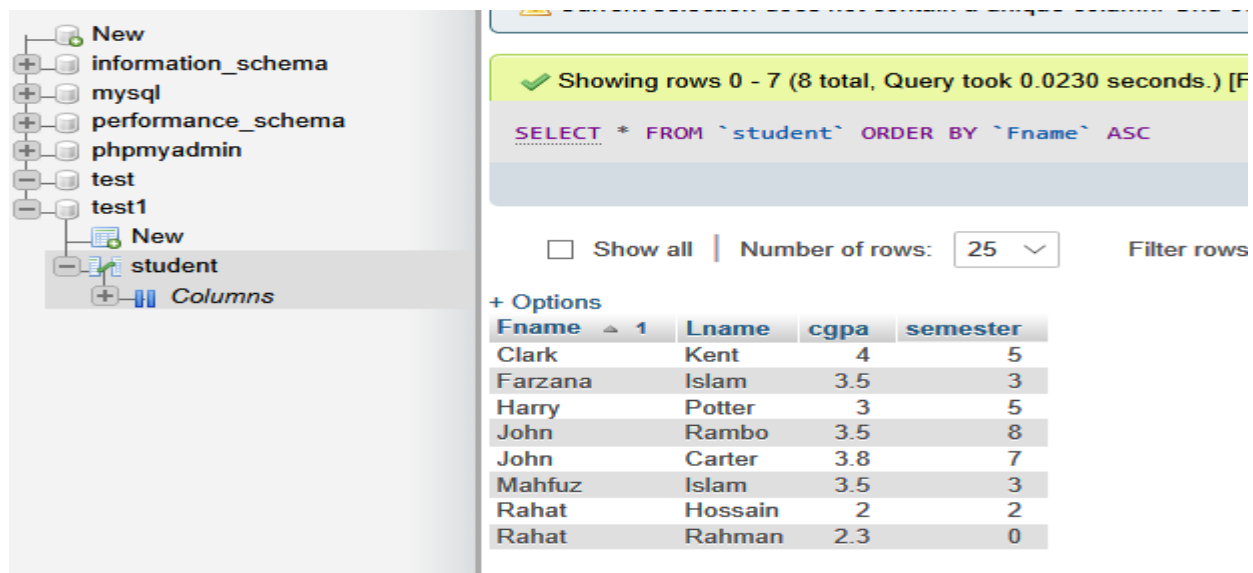
#### Example 4: Selecting data from a database and show them in HTML form

So far you have learnt how to create database and table as well as inserting data. Now it's time to retrieve data what have inserted in the preceding tutorial. The SQL SELECT statement is used to select the records from database tables. Its basic syntax is as follows:

```
SELECT column1_name, column2_name, columnN_name FROM table_name;
```

Let's make a SQL query using the SELECT statement, after that we will execute this SQL query through passing it to the PHP `mysqli_query()` function to retrieve the table data.

Consider our student database table has the following records:



The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, with the 'student' table selected under the 'test' database. The main panel displays the query results for the SQL statement: `SELECT * FROM `student` ORDER BY `Fname` ASC`. The results show 8 rows of data, ordered by first name. A status bar at the top indicates 'Showing rows 0 - 7 (8 total, Query took 0.0230 seconds.)'. Below the query, there are options to 'Show all' or 'Number of rows: 25' and a 'Filter rows' button. The table data is as follows:

Fname	Lname	cgpa	semester
Clark	Kent	4	5
Farzana	Islam	3.5	3
Harry	Potter	3	5
John	Rambo	3.5	8
John	Carter	3.8	7
Mahfuz	Islam	3.5	3
Rahat	Hossain	2	2
Rahat	Rahman	2.3	0

The PHP code (Select\_Data.php) in the following example selects all the data stored in the student table (using the asterisk character (\*) in place of column name selects all the data in the table).

```
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "", "test1");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Attempt select query execution
$sql = "SELECT * FROM student";
if($result = mysqli_query($link, $sql)){
    if(mysqli_num_rows($result) > 0){
        echo "<table>";
        echo "<tr>";
        echo "<th>First Name</th>";
        echo "<th>Last Name</th>";
        echo "<th>CGPA</th>";
        echo "<th>Semester</th>";
        echo "</tr>";
        while($row = mysqli_fetch_array($result)){
            echo "<tr>";
            echo "<td>" . $row['Fname'] . "</td>";
            echo "<td>" . $row['Lname'] . "</td>";
            echo "<td>" . $row['cgpa'] . "</td>";
            echo "<td>" . $row['semester'] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
        // Free result set
        mysqli_free_result($result);
    } else{
        echo "No records matching your query were found.";
    }
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
```

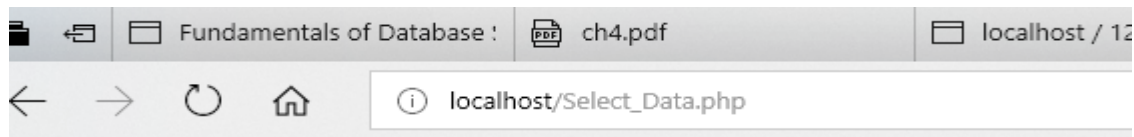
### **Explanation of Code (Procedural style)**

In the example above, the data returned by the `mysqli_query()` function is stored in the `$result` variable. Each time `mysqli_fetch_array()` is invoked, it returns the next row from the result set as an array. The while loop is used to loops through all the rows in the result set. Finally the value of individual field can be accessed from the row either by passing the field index or field name to the `$row` variable like

`$row['Fname']` or `$row[0]`, `$row['Lname']` or `$row[1]`, `$row['cgpa']` or `$row[2]`, and `$row['semester']` or `$row[3]`.

If you want to use the for loop you can obtain the loop counter value or the number of rows returned by the query by passing the `$result` variable to the `mysqli_num_rows()` function. This loop counter value determines how many times the loop should run.

Finally you will see the following output in the html document



First Name	Last Name	CGPA	Semester
Farzana	Islam	3.5	3
Rahat	Hossain	2	2
Mahfuz	Islam	3.5	3
John	Rambo	3.5	8
Clark	Kent	4	5
John	Carter	3.8	7
Harry	Potter	3	5
Rahat	Rahman	2.3	0

**Lab Work:** You could run any example from php/mysql book or run from this website (<https://www.tutorialrepublic.com/php-tutorial>) to the assigned problem given by the lab instructor.