

Lab Session 6: Introduction to MySQL

Working with MySQL using PHP involves several steps:

1. Opening a connection to MySQL and selecting the database
2. Sending a query to the database
3. Processing the results of the query
4. Closing the connection to MySQL

Connecting to MySQL

```
<?php
    $server = "localhost";
    $username = "root";
    $password = "";
    $database = "unebd";

    @mysql_connect($server, $username, $password);?>
```

Example 1: Create a Database and Two Tables

Instructions:

1. Create a database named **unebd** with MySQL.
2. Create two tables: **Enfants** (Children) and **Parents**, and populate them with data.

```
<?php
// 1. Connect to MySQL
$server = "localhost";
$username = "root";
$password = "";
$database = "unebd";

$db = @mysql_connect($server, $username, $password);
// 2. Select the database
mysql_select_db($database, $db);
// 3. Create the SQL query
$sql = 'SELECT * FROM Enfants;';
// 4. Send the query
$req = mysql_query($sql) or die('SQL
Error!<br>'.$sql.'<br>'.mysql_error());
// 5. Loop through each record
while($data = mysql_fetch_assoc($req)) {
    echo '<b>'.$data['NomEnf'].' '.$data['Cle'].'</b>
('.$data['PreEnf'].')<br>';
}
// 6. Close the connection
mysql_close();
?>
```

Example 2: Inserting Data from a Form

HTML Form (form.html):

```
<html>
<form method="POST" action="add.php">
<center>
    <input type="text" name="nom" size="20" placeholder="Name"
maxlength="35">
    <input type="text" name="prenom" size="20"
placeholder="First Name" maxlength="35"><br>
    <input type="text" name="email" size="20"
placeholder="Email" maxlength="70">
    <input type="text" name="adr" size="20"
placeholder="Address" maxlength="50"><br>
    <input type="submit" value="Submit" name="envoyer">
</center>
</form>
</html>
```

PHP Script (add.php):

```
<?php
// Retrieve form fields
$nom = $_POST['nom'];
$prenom = $_POST['prenom'];
$email = $_POST['email'];
$adr = $_POST['adr'];
// Check if any required fields are empty
if(empty($nom) || empty($prenom) || empty($email)) {
    echo '<font color="red">Warning: All fields except
<b>Address</b> must be filled!</font>';
} else {
// Connect to the database
$db = mysql_connect('localhost', 'root', '') or
die('Connection error: '.mysql_error());
// Select the database
mysql_select_db('unebd', $db) or die('Database selection
error: '.mysql_error());
// Create the SQL insert query
$sql = "INSERT INTO Parents(id, nom, prenom, email, adr)
VALUES(' ', '$nom', '$prenom', '$email', '$adr)";
// Execute the query
mysql_query($sql) or die('SQL Error:
' . $sql . '<br>'.mysql_error());
// Confirmation message
echo 'Your information has been added.';
// Close the connection
mysql_close(); }?>
```

Your Turn! E-Learning Management System

You will create a database for managing pupils, teachers, parents, and modules. You will build forms to add entries and write PHP scripts to search for pupils and parents by name.

Part 1: Create a New Database and Four Tables

1. Create a database named: elearning_db.
2. Create the following four tables:

Table Name	Columns
Pupils	id (INT, AUTO_INCREMENT, PRIMARY KEY), first_name (VARCHAR(50)), last_name (VARCHAR(50)), email (VARCHAR(100))
Teachers	id (INT, AUTO_INCREMENT, PRIMARY KEY), first_name (VARCHAR(50)), last_name (VARCHAR(50)), email (VARCHAR(100))
Parents	id (INT, AUTO_INCREMENT, PRIMARY KEY), first_name (VARCHAR(50)), last_name (VARCHAR(50)), phone (VARCHAR(20))
Modules	id (INT, AUTO_INCREMENT, PRIMARY KEY), module_name (VARCHAR(100)), description (TEXT)

Part 2: Create Forms to Add Records

Create four HTML forms: Add a new Pupil, Add a new Teacher, Add a new Parent, Add a new Module

Each form will submit the data to a corresponding add_*.php script that will insert the data into the appropriate table.

Part 3: Create Search Functions

1. **Create a PHP script** find_mohamed.php to find all pupils whose first name is "Mohamed" and display them.
2. **Create another PHP script** find_ali.php to find all parents whose first name is "Ali" and display them.

Part 4: Create a Dynamic Search Form

- Create a general search form where the user enters a name.
- Then create a PHP script search_pupil.php to search for pupils with the given name entered by the user and display the results.

Structure Summary

Part	What to Create	File Names Example
1	Database + Tables	SQL script or phpMyAdmin
2	Forms and Insert Scripts	add_pupil.php, add_teacher.php, add_parent.php, add_module.php
3	Fixed Search Scripts	find_mohamed.php, find_ali.php
4	Dynamic Search Form	search_form.html, search_pupil.php

Tips:

- Remember to connect to your database at the beginning of each PHP script.
- Use basic HTML forms with method="POST".
- Validate that all required fields are filled.
- Close the database connection at the end of your scripts.

Advanced Exercise 1: Display Pupils with Their Parents**Objective:**

- Create a **relationship** between Pupils and Parents (i.e., each pupil has a parent).
- Display a list of all pupils along with their parent's information.

Instructions:

1. Modify the Pupils table to add a column:
2. ALTER TABLE Pupils ADD COLUMN parent_id INT;
3. When adding a pupil, allow selection of their parent (dropdown list populated from Parents table).
4. Create a PHP script list_pupils_parents.php:
 - It will join the Pupils and Parents tables using parent_id.
 - It will display:
 - Pupil's full name, Pupil's email
 - Parent's full name, Parent's phone

SQL Example (Join Query):

```
SELECT Pupils.first_name AS pupil_first, Pupils.last_name AS pupil_last,
       Pupils.email,
       Parents.first_name AS parent_first, Parents.last_name AS
parent_last,
       Parents.phone
FROM Pupils
LEFT JOIN Parents ON Pupils.parent_id = Parents.id;
```

Advanced Exercise 2: Assign Teachers to Modules and List Them**Objective:**

- Create a **many-to-many relationship** between Teachers and Modules (because one teacher can teach multiple modules, and one module can be taught by multiple teachers).
- List all modules with their teachers.

Instructions:

1. Create an association table called Teacher_Module:

```
CREATE TABLE Teacher_Module (
    id INT AUTO_INCREMENT PRIMARY KEY,
    teacher_id INT,
    module_id INT);
```
2. Create a PHP form where: you can assign a teacher to one or multiple modules.
3. Create a PHP script list_modules_teachers.php:
4. It will display a list of modules with the names of the assigned teachers.

SQL Example (Join Query):

```
SELECT Modules.module_name, Teachers.first_name, Teachers.last_name
FROM Teacher_Module
JOIN Teachers ON Teacher_Module.teacher_id = Teachers.id
JOIN Modules ON Teacher_Module.module_id = Modules.id
ORDER BY Modules.module_name;
```