SLIIT
Discover Your Future

Lecture 08
PHP - Database handling

IT1100 Internet and Web technologies

SLIIT
FACULTY OF COMPUTING

Content

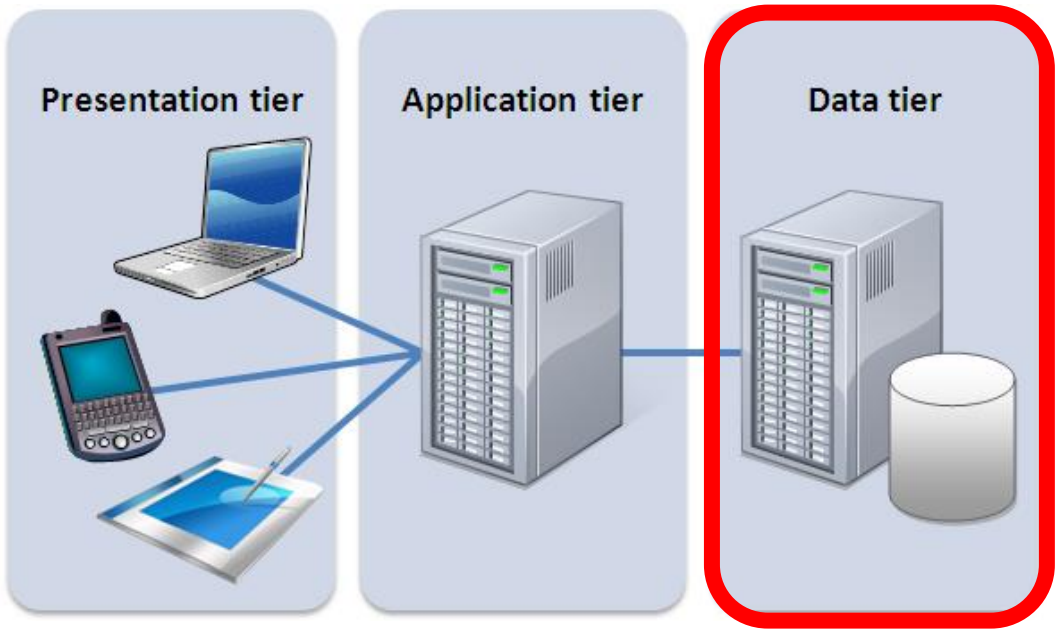
1. Introduction
2. The connection
3. **C**reate
4. **R**ead
5. **U**pdate
6. **D**elete

1. Introduction

- How to store data in software applications?
- What is the best method to store data in software applications?
- What is a Database?

1. Introduction

- **Database** is an external resource, hosted in a **database server**, and managed by a **DBMS**
- The database server is considered as a separate tire



1. Introduction

- **MySQL database server** is the de facto standard for PHP applications
- There are multiple ways to connect to a database using PHP
- PHP can perform **CRUD** operations on a database using SQL

1. Introduction

Ways to connect to DB using PHP

1. MySQL extension
   - Support only PHP versions before v7
   - Procedural

2. **MySQLi** (improved)
   - Support since PHP version 7
   - Support both procedural and **OOP**
   - Support prepared statements

3. PHP Data Objects (PDO)
   - A lightweight, consistent interface for accessing databases in PHP.
   - Support many DB servers
   - Only OOP
   - Support prepared statements

2. The connection
Configurations

- It is a good idea to keep the DB configurations in a dedicated file config.php

```
//The connection object
//$con= new mysqli(“Server”, “UN”, “PW”, “DB”);

$con=new mysqli("localhost","root",“123","test");
```

2. The connection Configurations

- Check for errors before continue

```
// Check connection
if ($con->connect_error)
{
   die("Connection failed: " . $con->connect_error);
}
```

The **connect_error** function returns the error description from the last connection error, if any. NULL if no error occurred.

2. The connection Configurations

- The configuration file can be linked when needed

- index.php (or any other page/file)

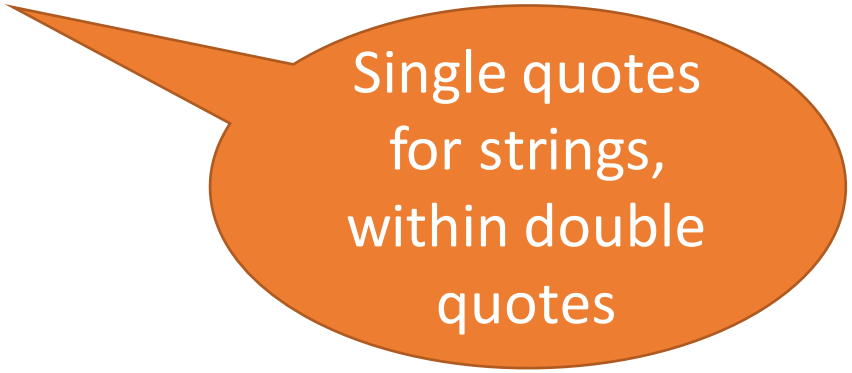
```
//Linking the configuration file
require 'config.php';
```

- require is identical to include except upon failure it will also produce a fatal E_COMPILE_ERROR level error.
- It will halt the script whereas include only emits a warning (E_WARNING) which allows the script to continue.
- The require_once statement is identical to require except PHP will check if the file has already been included, and if so, not include (require) it again

3. Create
The INSERT statement

- To create data, an insert SQL statement is used

$sql= "INSERT INTO myTable(ID, Name) VALUES (1, ‘SLIIT’)“;



3. Create
The statement

- Execute the statement

- $con->query($sql)
  - This returns a Boolean value to indicate the (un)successful execution of the statement in the DB server

3. Create
The statement

```
if ($con->query($sql))
{
    echo “Inserted successfully";
}
else
{
    echo “Error: ”. $con->error;
}
```

3. Create The statement

- Do not forget to close the connection
  - After executing any operation

```
$con->close();
```

Complete Code

```php
<?php
//Linking the configuration file
require 'config.php';
$sql= "INSERT INTO myTable(ID,Name)VALUES(11111,'SLIIT')";
   if($con->query($sql)){
            echo "Inserted successfully";
   }
   else{
            echo "Error:". $con->error;
   }
$con->close();
?>
```

config.php

```php
<?php
//The connection object

$con=new mysqli("localhost","root","","MyDB");
// Check connection
            if($con->connect_error){
                        die("Connection failed: " . $con->connect_error);
            }
?>
```

Problems in data INSERT method

- Can insert One Record at a time
- User need access rights to internal .PHP pages stored in webserver (ex. /htdocs/…)

Solutions

- Use a HTML Form
- Use a PHP Form

Solution1
Use a HTML Form

```
<?php
//Linking the configuration file
require 'config.php';
$ID = $_POST["stuID"];
$Name = $_POST["stuName"];
$sql= "INSERT INTO myTable(ID,Name)VALUES($ID,$Name)";
                if($con->query($sql)){
                        echo "Inserted successfully";
                }
                else{
                        echo "Error:". $con->error;
                }
$con->close();
?>
```

```
<!doctype html>
<html>
    <head>  </head>

    <body>
            <form method="post" action="form_process.php">
            <h3>Input Student Data </h3>
                    Student ID :<input type="text" name="stuID"><BR />
                    Student Name :<input type="text" name="stuName"><BR />
                    <input type="submit" value="Submit">
                    <input type="reset" value="Reset">
            </form>
    </body>
</html>
```

```php
<?php
//Linking the configuration file
require 'config.php';
?>
    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
            <h3>Input Student Data </h3>
            Student ID :<input type="text" name="stuID"><BR />
            Student Name :<input type="text" name="stuName"><BR />
            <input type="submit" value="Submit" name="btnSubmit">
            <input type="reset" value="Reset">
    </form>
<?php
if(isset($_POST["btnSubmit"])){
    $stuID = $_POST["stuID"];
    $stuName = $_POST["stuName"];

    $sql = "INSERT INTO myTable(ID,Name)VALUES($stuID,'$stuName')";
            if($con->query($sql)){
                    echo "Inserted successfully";
            }
            else{
                    echo "Error:". $con->error;
            }
}
$con->close();
?>
```

1 2 3 4 5 6 7 8

Solution 2
Use a PHP Form

4. Read

Select statement

- When reading data from a DB, we use a select statement, which returns a dataset as the result.

```
$sql = “select ID, name from myTable”
```

4. Read
Result set

- We execute the select SQL statement, then assign the result set into a variable

```
$result = $con->query($sql);
```

4. Read
Result set - availability

- If only there are results, we can read them

```
if ($result->num_rows > 0)
{
    //read data
}
else
{   echo “no results”;   }
```

4. Read
Result set – read data

- We read the dataset row by row using a loop
- There are multiple functions to fetch a row from a dataset

- fetch_all — Fetches all result rows as an associative array, a numeric array, or both
- fetch_array — Fetch a result row as an associative, a numeric array, or both
- fetch_assoc — Fetch a result row as an associative array
- fetch_field_direct — Fetch meta-data for a single field
- fetch_field — Returns the next field in the result set
- fetch_fields — Returns an array of objects representing the fields in a result set
- fetch_object — Returns the current row of a result set as an object
- fetch_row — Get a result row as an enumerated array

4. Read
Result set – read data

- Lets use fetch_assoc(), which return the row as an associative array

```
while($row = $result->fetch_assoc())
{
        //Read and utilize the row data
}
```

4. Read
Result set – read data

- Column names can be used as the indexes to read the cell data in the fetched row

```
echo $row["ID"]. " – " . $row["Name"] . "<BR />";
```

EX: show the data inside a table, on the page

4. Read
Complete function

- Column names can be used as the indexes to read the cell data in the fetched row

```
echo $row["ID"]. " – " . $row["Name"] . "<BR />";
```

EX: show the data inside a table, on the page

```php
<?php
//Linking the configuration file
require 'config.php';
$sql = "select ID, Name from myTable";
$result = $con->query($sql);


    if($result->num_rows > 0){
            //read data
            while($row = $result->fetch_assoc()){
            //Read and utilize the row data
                    echo $row["ID"]. " – " . $row["Name"] . "<BR />";
            }
    }
    else
    {
            echo "no results";
    }
$con->close();
?>
```

```php
<?php
//The connection object
$con=new mysqli("localhost","root","","MyDB");
// Check connection
        if($con->connect_error){
                die("Connection failed: " . $con->connect_error);
        }
?>
```

Complete Code

4. Read

Complete function

```php
<?php
require 'config.php';
function readData()
{
   global $con;
   $sql = "SELECT ID, Name FROM myTable";
   $result = $con->query($sql);
   if ($result->num_rows > 0)
   {
      while($row = $result->fetch_assoc())
      {
         echo "ID: " . $row["ID"]. " - Name: " . $row["Name"]. "<br>";
      }
   }
   else
   {
      echo "No results";
   }
   $con->close();
}
readData();
```

```php
<?php
$con=new mysqli("localhost","root","","test");

if($con->connect_error)
{
   die("Connection failed: ". $con->connect_error);
}
?>
```

Question 1

- Students details are stored in the table (“myTable”) inside the myDB database in MySQL server.
- Write a .php code to display all the records available in the table (“myTable”).
- Format the content using a table.
- Database details
  - DB server name = “localhost”
  - User name=“root”
  - Password=“”
  - Default database=“MyDB”

Table structure

#	Name	Type
1	stuID	int(11)
2	stuName	varchar(25)
3	stuMobile	int(11)
4	stuEmail	varchar(25)

Open database connection

```
<?php
//The connection object

$con=new mysqli("localhost","root","","MyDB");
// Check connection
        if($con->connect_error){
                die("Connection failed: " . $con->connect_error);
        }
?>
```

Display Records in a Table

```
$sql= "SELECT * FROM MyTable";


if($result=$con->query($sql)){
                        if($result->num_rows > 0){
                                    echo ("<table border='1'>");
                                    while($row = $result->fetch_assoc()){
                                                echo ("<tr>");
                                                echo ("<td>". $row['stuID']. "</td>");
                                                echo ("<td>" . $row['stuName'] . "</td>");
                                                echo ("<td>" . $row['stuMobile'] . "</td>");
                                                echo ("<td>" . $row['stuEmail'] . "</td>");
                                                echo ("</tr>");
                                    }
                                    echo ("</table>");
                        }else{
                                    echo "no results";
                        }
}
else{
            echo "Error:". $con->error;
}
```

Question 2

- Students details are stored in the table (“myTable”) inside the myDB database in MySQL server.
- Create a .php file perform following
  - User wants to get students details by entering student name fully or partially.
  - Write a .php code to display all the matching records(contain the input) in the table “myTable”.

Open database connection

```
<?php
//The connection object

$con=new mysqli("localhost","root","","MyDB");
// Check connection
        if($con->connect_error){
                die("Connection failed: " . $con->connect_error);
        }
?>
```

Search Form

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
                    <h3>Search Student Data </h3>
                    Student Name :<input type="text" name="stuName">
                    <input type="submit" value="Submit" name="btnSubmit">
                    <input type="reset" value="Reset">
</form>
```

Search Student Data

Student Name : Submit Reset

Display matching Records

```
<?php
if(isset($_POST["btnSubmit"])){
        $stuName = $_POST["stuName"];
        $sql= "SELECT * FROM MyTable where stuName LIKE '%$stuName%'";
            if($result=$con->query($sql)){
                if($result->num_rows > 0){
                    echo ("<table border='1'>");
                    while($row = $result->fetch_assoc()){
                        echo ("<tr>");
                        echo ("<td>". $row['stuID']. "</td>");
                        echo ("<td>" . $row['stuName'] . "</td>");
                        echo ("<td>" . $row['stuMobile'] . "</td>");
                        echo ("<td>" . $row['stuEmail'] . "</td>");
                        echo ("<tr>");
                    }
                    echo ("</table>");
                }else{
                    echo "no results";
                }
```

Close database connection

```
<?php
        $con->close();
?>
```

Complete code demo

Question 3

- Students details are stored in the table (“myTable”) inside the myDB database in MySQL server.
- Create a .php file perform following
  - Display all the available Records to the user.
  - User can input any ‘stuID’ in to available textbox press ‘DELETE’ button.
  - Write a .php code to delete the specified record
  - Display remaining Records to the user.

Open Database connection

```
<?php
//The connection object
$con=new mysqli("localhost","root","","MyDB");
// Check connection
        if($con->connect_error){
                die("Connection failed: " . $con->connect_error);
        }
?>
```

Delete Form

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
                    <h3>Delete Student Data </h3>
                    Student ID :<input type="text" name="stuID"><BR />
                    <input type="submit" value="Delete" name="btnSubmit">
                    <input type="reset" value="Reset">
</form>
```

Delete Student Data

Student ID :

Delete Reset

Display Records

```
//-----------------------------------------------function readData()-----------------------------------
function readData(){
global $con;
$sql = "SELECT * FROM myTable";
$result = $con->query($sql);
                if ($result->num_rows > 0) {
                                        echo ("<table border='1'>");
                        while($row = $result->fetch_assoc()) {
                                                echo ("<tr>");
                                                echo ("<td>". $row['stuID']. "</td>");
                                                echo ("<td>" . $row['stuName'] . "</td>");
                                                echo ("<td>" . $row['stuMobile'] . "</td>");
                                                echo ("<td>" . $row['stuEmail'] . "</td>");
                                                echo ("<tr>");
                        }
                        echo ("</table>");
                }else {
                    echo "No results <BR />";
                }
}
```

Record Delete

```
//----------------------function  deleteData()-------------------------

function deleteData($stuID){
global $con;
$sql = "delete from myTable where stuID='$stuID'";
if($con->query($sql)){
                    echo "Deleted successfully<BR />";
}else{
    echo "Error: ".$con->error;
}
}
```

Display remaining records

```
<?php
readData();
if(isset($_POST["btnSubmit"])){
        $stuID=$_POST["stuID"];
        if($stuID!=""){
        deleteData($stuID);
        readData();
        }

}
$con->close();
```

Update records
Complete function - Update

Steps

1. Create a form to search records
   - Ask user to input a unique field like a primary key.
2. Display the details of the matching record.
   - Display updatable values as editable fields.
3. Replace existing values with the input values
   - Write new values to the database

Update records
Complete function - Update

```
function updateData()
{
    global $con;
    $sql = "update myTable set ID='2', Name='SLIIT updated' where ID='1'";

    if($con->query($sql))
    {
        echo "Updated successfully";
    }
    else
    {
        echo "Error: ".$con->error;
    }
    $con->close();
}
```