

Web Programming (203105353)

Mr. Dheeraj Kumar Singh, Assistant Professor
Information Technology Department





CHAPTER-7

PHP AND MYSQL



Introduction PHP and MySQL

- PHP is the most popular scripting language for web development.
- It is free, open source and server-side language.
- It is powerful enough to be at the core of the biggest blogging system on web - WordPress.
- It is deep enough to run the largest social network - Facebook.
- MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).
- It is also free and open source.
- **MySQL is the widely used database system with PHP language.**

MySQL Characteristics

- Fast, secure and scalable.
- Operates on a wide variety of operating systems including UNIX or Linux, Microsoft Windows, Apple Mac OS X and others.
- Follows the working of a client/server architecture
- MySQL was developed by Sun Microsystems Now (Oracle Corporation).
- The data in a MySQL database are stored in tables.



Basic PHP Examples

- A PHP script can be placed anywhere in the Web document.
- A PHP script starts with `<?php` and ends with `?>`.

```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>
<? php
echo "Hello World!";
?>
</body>
</html>
```



Connection to server

- PHP 5 and later can work with a MySQL database using:
 - **MySQLi (MySQL improved)**
 - **PDO (PHP Data Objects)**
- Earlier versions of PHP used the MySQL extension.
- However, this extension was deprecated in 2012.
- PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.

Connection to server : MySQLi

MySQLi can use either **object-oriented** or **procedural** API.

Syntax: Object Oriented way

```
$mysqli = new mysqli("servername", "username", "password", "database");
```

Syntax: Procedural way

```
$link = mysqli_connect("servername", "username", "password", "database");
```





Connection to server : MySQLi Example

```
<?php
$link = mysqli_connect("localhost", "root", ""); //user 'root' with no password
if($link === false) // Check connection
{
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
// Print host information
echo "Connect Successfully. Host info: " . mysqli_get_host_info($link);
?>
```




Creating MySQL Database Using PHP : Example

```
<?php
$link = mysqli_connect("localhost", "root", ""); //user 'root' with no password
if($link === false)                               // Check connection
{
    die("ERROR: Could not connect. " . mysqli_connect_error()); }
$sql = "CREATE DATABASE demo";                    // Create database query

if(mysqli_query($link, $sql)){
    echo "Database created successfully";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link); }
// Close connection
mysqli_close($link); ?>
```

Selecting a Database using MySQLi

- To perform any operation on the database like create table, delete table, etc. first we have to select the database on which we want to do the operation.
- **mysql_list_dbs()** — List databases available on a MySQL server. We can also able to list the database by using Show database command.

- **Select database Statement Example**

```
$link = mysqli_connect("localhost", "root", "", "demo");
```



Listing Database: Example

```
<?php
// Usage without mysql_list_dbs()
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
$res = mysql_query("SHOW DATABASES");
while ($row = mysql_fetch_assoc($res)) {
    echo $row['Database'] . "\n";
}
// Deprecated as of PHP 5.4.0
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
$db_list = mysql_list_dbs($link);
while ($row = mysql_fetch_object($db_list)) {
    echo $row->Database . "\n";
}
?>
```

Listing Tables

- **mysql_list_tables()** — List tables in a MySQL database Description

`mysql_list_tables (string $database [, resource $link_identifier = NULL]) :`
`resource`

- A list of table names from a MySQL database is retrieved.
- That function is being deprecated.
- It is preferable to use `mysql_query()` instead to state the SQL `SHOW TABLES [FROM db_name] [LIKE 'pattern']`.

Listing Table: Example

```
<?php
$dbname = 'mysql_dbname';
if (!mysql_connect('mysql_host', 'mysql_user', 'mysql_password')) {
    echo 'Could not connect to mysql';
    exit; }
$sql = "SHOW TABLES FROM $dbname";
$result = mysql_query($sql);
if (!$result) {
    echo "DB Error, could not list tables\n";
    echo 'MySQL Error: ' . mysql_error();
    exit;
}
while ($row = mysql_fetch_row($result)) {
    echo "Table: {$row[0]}\n";
} mysql_free_result($result); ?>
```

Create Tables and Insert data

- The statement SQL CREATE TABLE is used for creating a table in the database.
- PHP mysqli query() function is used to execute the query and create table.
- The statement INSERT INTO is used to insert new rows into a table of databases.

Important rules:

- In PHP you must quote the SQL query
- String values must be quoted inside the SQL query
- Not to quote numeric values
- Not to quote the word NULL



Create Table: Example

```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error()); }
// Attempt create table query execution
$sql = "CREATE TABLE persons(
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(30) NOT
    NULL, last_name VARCHAR(30) NOT NULL, email VARCHAR(70) NOT NULL UNIQUE )";
if(mysqli_query($link, $sql)){
    echo "Table created successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);}
mysqli_close($link);
?>
```




Insert Data: Example

```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error()); }
// Attempt insert query execution
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES ('Rahul',
'Sharma', rahulsharma@mail.com)";
if(mysqli_query($link, $sql)){
    echo "Records inserted successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}
mysqli_close($link);
?>
```



Inserting Multiple Rows into a Table: Example

```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());}
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES
    ('Raj', 'Mehta', 'rajmehta@mail.com'),
    ('Rahul', 'Joshi', 'rahuljoshi@mail.com'),
    ('Dipali', 'Patel', 'dipalipatel@mail.com'),
    ('Tejus', 'Pandya', 'tejuspandya@mail.com')";
if(mysqli_query($link, $sql)){ echo "Records added successfully."; }
// Close connection
mysqli_close($link);
?>
```

PHP MySQL Prepared Statements

- **What is Prepared Statement**

- A prepared statement (also known as parameterized statement) is simply a placeholder-containing SQL query template instead of the actual parameter values.
- These placeholders will be replaced by the actual values at the time of execution of the statement.
- MySQLi supports the use of anonymous positional placeholder (?).

`INSERT INTO persons (first_name, last_name, email) VALUES (?, ?, ?);`



PHP MySQL Prepared Statements

- The prepared statement execution consists of two stages:

1) Prepare - A SQL statement template is created during the Prepare stage and sent to the database server. The server parses the template for statements, performs an optimization of syntax checks and queries, and stores it for later use.

2) Execute - The parameter values are sent to the server during execution. To execute it, the server creates a statement from the template statement and those values.



PHP MySQL Prepared Statements: Example

```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");
if($link === false){ die("ERROR: Could not connect. " . mysqli_connect_error()); }
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES (?, ?, ?)";
if($stmt = mysqli_prepare($link, $sql)){
    // Bind variables to the prepared statement as parameters
    mysqli_stmt_bind_param($stmt, "sss", $first_name, $last_name, $email);
    $first_name = "Ram";
    $last_name = "Mehta";
    $email = "rammehta@mail.com";
    mysqli_stmt_execute($stmt);
```



PHP MySQL Prepared Statements: Example Con..

```
$first_name = "Rahul";  
$last_name = "Joshi";  
$email = "rahuljoshi@mail.com";  
mysqli_stmt_execute($stmt);  
echo "Records inserted successfully."  
} else{  
    echo "ERROR: Could not prepare query: $sql. " . mysqli_error($link);  
}  
// Close statement  
mysqli_stmt_close($stmt);  
// Close connection  
mysqli_close($link);  
?>
```

SELECT Query

- To select the records from database tables, the SQL SELECT statement is used.
`SELECT column1_name, column2_name, columnN_name FROM table_name;`
- The WHERE clause is used to extract only those records which satisfy a given condition.

`SELECT column_name(s) FROM table_name WHERE column_name operator value`

- Consider our persons database table with following records as example:

id	first_name	last_name	email
1	Ram	Mehta	ram@gmail.com
2	Rahul	Joshi	rahul@gmail.com
3	Tejus	Pandya	tejus@gmail.com



SELECT Query Example

```
<?php
$link = mysqli_connect("localhost",
"root", "", "demo");
if($link === false){
    die("ERROR: Could not connect. " .
mysqli_connect_error()); }
$sql = "SELECT * FROM persons";
if($result = mysqli_query($link, $sql)){
    if(mysqli_num_rows($result) > 0){
        echo "<table>";
        echo "<tr>";
        echo "<th>id</th>";
        echo "<th>first_name</th>";
        echo "<th>last_name</th>";
        echo "<th>email</th>";
        echo "</tr>";
```

```
while($row = mysqli_fetch_array($result)){
    echo "<tr>";
    echo "<td>" . $row['id'] . "</td>";
    echo "<td>" . $row['first_name'] . "</td>";
    echo "<td>" . $row['last_name'] . "</td>";
    echo "<td>" . $row['email'] . "</td>";
    echo "</tr>";
} echo "</table>";
mysqli_free_result($result);
} else{
    echo "No records matching your query were
found.";
}
} else{
    echo "ERROR: Could not able to execute $sql. " .
mysqli_error($link);
}
mysqli_close($link); ?>
```



WHERE Clause Example

```
<?php
$link = mysqli_connect("localhost",
"root", "", "demo");
if($link === false){
    die("ERROR: Could not connect. ".
mysqli_connect_error());
}
// Attempt select query execution
$sql = "SELECT * FROM persons
WHERE first_name='Ram'";
if($result = mysqli_query($link,
$sql)){
    if(mysqli_num_rows($result) > 0){
        echo "<table>";
        echo "<tr>";
```

```
echo "<th>id</th>";
        echo "<th>first_name</th>";
        echo "<th>last_name</th>";
        echo "<th>email</th>";
        echo "</tr>";
while($row = mysqli_fetch_array($result)){
    echo "<tr>";
        echo "<td>" . $row['id'] . "</td>";
        echo "<td>" . $row['first_name'] .
"</td>";
        echo "<td>" . $row['last_name'] .
"</td>";
        echo "<td>" . $row['email'] . "</td>";
        echo "</tr>";
    }
}
```

WHERE Clause Example Con...

```
echo "</table>";  
    // Close 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);  
?>
```



PHP MySQL UPDATE Query

- The UPDATE statement is used to change or modify the existing records.
- The basic syntax of the UPDATE statement can be given with:

```
UPDATE table_name SET column1=value, column2=value2,... WHERE  
column_name=some_value
```



PHP MySQL Update Query: Example

```
<?php
$link =
mysqli_connect("localhost",
"root", "", "demo");

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

```
// Attempt update query execution
$sql = "UPDATE persons SET
email='ram_new@gmail.com' WHERE id=1";
if(mysqli_query($link, $sql)){
    echo "Records were updated successfully.";
} else {
    echo "ERROR: Could not able to execute $sql.
    ". mysqli_error($link);
}
```

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



PHP MySQL DELETE Query

- The SQL DELETE statement is used to delete records from a table.
- The basic syntax of the DELETE statement can be given with:

DELETE FROM table_name WHERE column_name=some_value



PHP MySQL DELETE Query: Example

```
<?php
$link = mysqli_connect("localhost",
"root", "", "demo");

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

```
// Attempt delete query execution
$sql = "DELETE FROM persons WHERE
first_name='Ram'";
if(mysqli_query($link, $sql)){
    echo "Records were deleted successfully.";
} else{
    echo "ERROR: Could not able to execute
$sql. " . mysqli_error($link);
}
```

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


PHP MySQL – Altering Table

- Existing tables can be altered using the `mysql_query()` function.
 - You can add or modify fields or columns in an existing table.
 - You can delete fields.
 - You can even delete an entire table and start over.
- **Add a Column or Field to a Table**
 - To add a column to an existing table the syntax would be:
`mysql_query("ALTER TABLE demo ADD street CHAR(30)");`
 - We can also specify where you want to add the field.
`mysql_query("ALTER TABLE demo ADD street CHAR(30) AFTER email");`
- **Modify a Column or Field**
 - `mysql_query("ALTER TABLE demo MODIFY COLUMN column_name datatype);`

PHP MySQL – Delete Table

- **DROP table**

To delete entire table from your database use **DROP** query of mysql.

Syntax for DROP query is **DROP TABLE** tablename

- **Example**

```
function DropTable($table)
{
    $query = "DROP TABLE ". $table;
    $result = ExecuteQuery($query,"Table deleted successfully");
    return $result;
}
```

PHP MySQL – Deleting Database

- The **DROP DATABASE** statement drops all tables in the database and deletes the database permanently.

- The following shows the syntax of the DROP DATABASE statement:

```
DROP DATABASE [IF EXISTS] database_name;
```

- The **DROP DATABASE** statement returns the number of tables that were deleted.

In MySQL, the schema is the synonym for the database, therefore, you can use them interchangeably:

```
DROP SCHEMA [IF EXISTS] database_name;
```



PHP MySQL – Deleting Database: Example

```
<?php
$dbhost = 'localhost:3306';
$dbuser = 'root';
$dbpass = '<Password Here>';
$conn = mysqli_connect($dbhost, $dbuser, $dbpass);
if(! $conn ){
    echo 'Connected failure<br>'; }
echo 'Connected successfully<br>';
$sql = "DROP DATABASE demo";
if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn); }
mysqli_close($conn); ?>
```

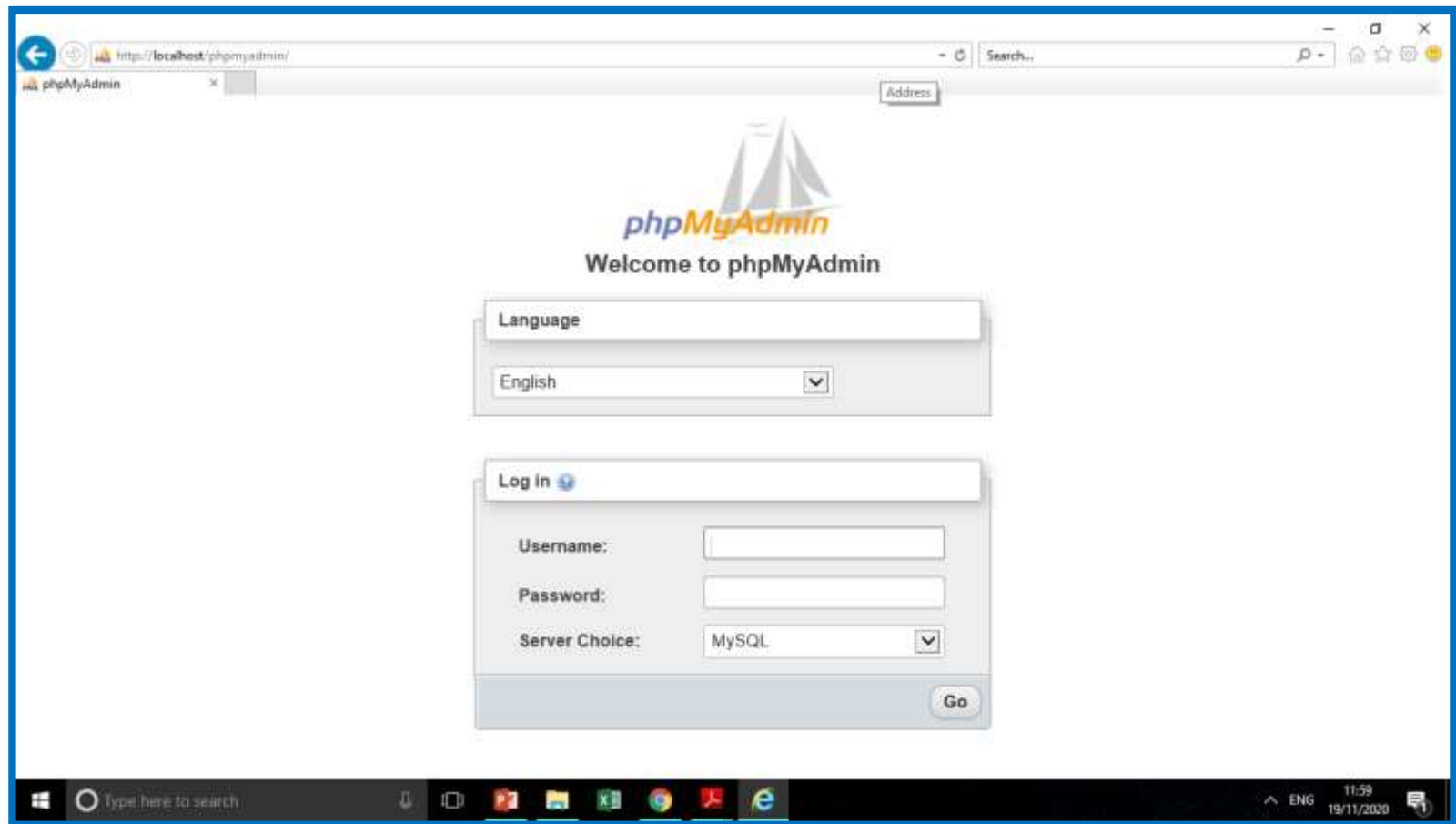


PHP MyAdmin

- PhpMyAdmin is an open-source software tool that was introduced in PHP on 9 September 1998.
- phpMyAdmin supports different types of MariaDB and MySQL operations.
- PhpMyAdmin's primary purpose is to handle MySQL administration over web.
- It is the most popular MySQL data base management application.
- Using this software we are able to create, update, drop, alter, delete, import and export MySQL database tables.



PHP MyAdmin





Advantage of phpMyAdmin

- We can easily create, delete, and edit database and can manage all elements using the graphical interface of phpMyAdmin, which is much easier than MySQL command-line editor.
- phpMyAdmin helps us to control the user's permission and operate several servers at the same time.
- We can also backup our database and export the data into different formats like XML, CSV and Spreadsheet.
- We can execute complex SQL statements and queries, create and edit functions, triggers, and events using the graphical interface of phpMyAdmin.

Disadvantage of phpMyAdmin

- phpMyAdmin is difficult to install as it needs three more software tools before installation, which is- Apache server, PHP, and MySQL.
- We have to install all these software tools individually. XAMPP is the easiest way to get phpMyAdmin.
- It has no schema visualization.
- phpMyAdmin is a web-based software tool which runs only on the browser.
- It does not have auto-compilation capability.

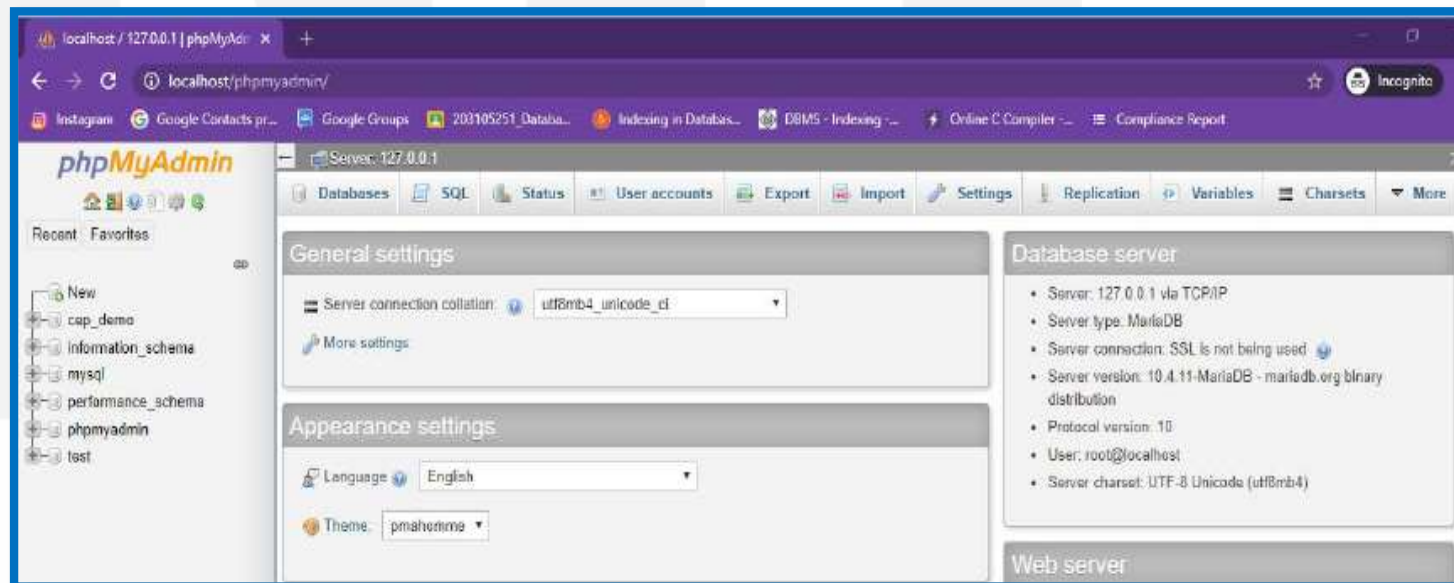
Data Backup problem with phpMyAdmin

- **Scheduling** - There is no way to export the data of the database in phpMyAdmin automatically.
- **Storage media support** - phpMyAdmin is web-based software, so it runs only on browser. We can take backups only to local system drives.
- **Compression, Encryption, and other option** - The files which are exported with phpMyAdmin are saved as common text files. Whereas storing these files in the original form usually takes a lot of disk storage.



Prerequisite for PhpMyAdmin

- Web server – Such as Apache, IIS
- PHP
- Database - MySQL
- Web Browser



Difference between phpMyAdmin and MySQL

- **MySQL** is a database server, on the other hand, phpMyAdmin is a server tool to access the database on MySQL server.
- **phpMyAdmin** provides a user interface to execute query within SQL
- We have to learn database query to create, delete, alter, update, and drop to execute on MySQL console, whereas in phpMyAdmin we can do it using graphical user interface which automatically executes these queries in background.

× ○ DIGITAL LEARNING CONTENT



Parul[®] University



www.paruluniversity.ac.in

