

Describe the process of database Connection in PHP

Step-by-Step Database Connection Process

1. Define database connection details (host, username, password, and database name).
2. Use `mysqli_connect()` to establish a connection.
3. Check if the connection was successful.
4. Close the connection when done.

Q. program to connect database in php

```
<?php
// Step 1: Define database credentials
$host = "localhost"; // Server name (usually "localhost")
$user = "root"; // Database username
$pass = ""; // Database password (leave empty for XAMPP)
$dbname = "mydatabase"; // Database name

// Step 2: Connect to the database
$conn = mysqli_connect($host, $user, $pass, $dbname);

// Step 3: Check connection status
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully!";

// Step 4: Close the connection (optional but recommended)
mysqli_close($conn);
?>
```

- `mysqli_connect($host, $user, $pass, $dbname);` → Establishes the connection.
- `mysqli_connect_error();` → Returns an error message if the connection fails.



- `mysqli_close($conn);` → Closes the connection when it's no longer needed.

What is Mysql? Describe the database connection methods in php.

MySQL is an open-source **Relational Database Management System (RDBMS)** that allows you to store, manage, and retrieve structured data efficiently. It uses **SQL (Structured Query Language)** for querying and managing data.

Features of MySQL:

- Open-source and widely used in web applications.
- Supports large databases and high-speed transactions.
- Works well with PHP for dynamic web development.
- Provides security with user authentication and access control.

PHP provides several ways to connect to a MySQL database:

1. Using `mysqli_connect()` (Procedural Method)

The **`mysqli_connect()`** function is used to establish a database connection.

```
<?php
$host = "localhost"; // Server name
$user = "root"; // Database username
$pass = ""; // Password (empty for XAMPP)
$dbname = "mydatabase"; // Database name

// Create connection
$conn = mysqli_connect($host, $user, $pass, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully!";
?>
```

2. Using `mysqli` (Object-Oriented Method)

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$dbname = "mydatabase";

// Create connection
$conn = new mysqli($host, $user, $pass, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully!";
?>
```

Write any three database functions of PHP with example.

PHP provides several functions to interact with MySQL databases. Here are three commonly used ones:

1. mysqli_connect() – Connecting to a Database. This function is used to establish a connection between PHP and MySQL.

Example:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "mydatabase");

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully!";
?>
```

2. mysqli_query() – Executing SQL Queries. This function is used to execute SQL queries such as SELECT, INSERT, UPDATE, DELETE.

Example:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "mydatabase");

// Insert data into a table
$sql = "INSERT INTO users (name, email) VALUES ('John Doe', 'john@example.com')";
if (mysqli_query($conn, $sql)) {
    echo "Record inserted successfully!";
} else {
    echo "Error: " . mysqli_error($conn);
}

// Close the connection
mysqli_close($conn);
?>
```

3. mysqli_fetch_assoc() – Fetching Data as an Associative Array. This function is used to retrieve data from a result set in the form of an associative array.

Example:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "mydatabase");

// Fetch user data
$sql = "SELECT * FROM users";
$result = mysqli_query($conn, $sql);

while ($row = mysqli_fetch_assoc($result)) {
    echo "ID: " . $row["id"] . " – Name: " . $row["name"] . "<br>";
}

mysqli_close($conn);
?>
```

Three database methods used program in php that create table if already not exists.

```
<?php
// Step 1: Connect to the Database
$conn = mysqli_connect("localhost", "root", "", "mydatabase");

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully!<br>";

// Step 2: Create Table (if not exists)
$table_sql = "CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL
)";
if (mysqli_query($conn, $table_sql)) {
    echo "Table 'users' is ready!<br>";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

// Step 3: Insert Data into Table
$insert_sql = "INSERT INTO users (name, email) VALUES ('John Doe', 'john@example.com')";
if (mysqli_query($conn, $insert_sql)) {
    echo "Record inserted successfully!<br>";
} else {
    echo "Error inserting record: " . mysqli_error($conn);
}

// Step 4: Retrieve and Display Data
$select_sql = "SELECT * FROM users";
$result = mysqli_query($conn, $select_sql);

echo "<h3>User List:</h3>";
while ($row = mysqli_fetch_assoc($result)) {
    echo "ID: " . $row["id"] . " – Name: " . $row["name"] . " – Email: " . $row["email"] . "<br>";
}

// Step 5: Close Connection
mysqli_close($conn);
```

?>

What is session? How Sessions can be handled in PHP? Explain with example.

A session is used to store user data on the server across multiple pages. PHP assigns a unique session ID to each user.

Steps to Handle Sessions in PHP

1. Start a Session: `session_start();`
2. Store Data: `$_SESSION["variable_name"] = "value";`
3. Access Data: `echo $_SESSION["variable_name"];`
4. Destroy Session:
`session_unset();` (Removes session variables)
`session_destroy();` (Ends the session)

1. Start Session and Store Data (login.php)

```
<?php
session_start();
$_SESSION["username"] = "Rajesh";
echo "Session started.";
?>
```

2. Access Session Data (profile.php)

```
<?php
session_start();
echo "Welcome, " . $_SESSION["username"];
?>
```

3. Destroy Session (logout.php)

```
<?php
session_start();
session_unset();
session_destroy();
```

```
echo "Logged out.";
?>
```

Advantages of Session: **Sessions are Secure, stores more data than cookies, maintains user authentication.**

Small project example to understand session .

index.php

```
<?php
session_start();
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST["username"];
    $password = $_POST["password"];

    // Simple authentication (Replace with database in real projects)
    if ($username == "admin" && $password == "1234") {
        $_SESSION["user"] = $username;
        $_SESSION["pass"] = $password;
        header("Location: dashboard.php"); // Redirect to dashboard
        exit();
    } else {
        echo "Invalid username or password!";
    }
}
?>

<!DOCTYPE html>
<html>
<head>
<title>Login</title>
</head>
<body>
<h2>Login</h2>
<form method="POST">
<input type="text" name="username" placeholder="Username" required><br><br>
<input type="password" name="password" placeholder="Password" required><br><br>
<button type="submit">Login</button>
</form>
</body>
</html>
```

dashboard.php

```
<?php
session_start();
if (!isset($_SESSION["user"]) || !isset($_SESSION["pass"])) {
header("Location: index.php"); // Redirect if not logged in
exit();
}
?>

<!DOCTYPE html>
<html>
<head>
<title>Dashboard</title>
</head>
<body>
<h2>Welcome, <?php echo $_SESSION["user"]; ?>!</h2>
<h2> In this page,you can use the session variable store value here like username: <?php echo $_SESSION["user"]."
and password:". $_SESSION['pass'];?> entered from login page.</h2>
<a href="logout.php">Logout</a>
</body>
</html>
```

logout.php

```
<?php
session_start();
session_unset();
session_destroy();
header("Location: index.php"); // Redirect to login page
exit();
?>
```

Write down a PHP code snippet for selecting a Eid, Ename, and salary from table "Employee" and display them in browse window. easy program


```
<?php
// Step 1: Connect to the database
$conn = new mysqli("localhost", "root", "", "your_database");

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

// Step 2: Fetch data from Employee table
$sql = "SELECT Eid, Ename, Salary FROM Employee";
$result = $conn->query($sql);

// Step 3: Display data in a table
echo "<table border='1'>
<tr>
<th>Eid</th>
<th>Ename</th>
<th>Salary</th>
</tr>";

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<tr>
        <td>{$row['Eid']}</td>
        <td>{$row['Ename']}</td>
        <td>{$row['Salary']}</td>
        </tr>";
    }
} else {
    echo "<tr><td colspan='3'>No records found</td></tr>";
}

echo "</table>";

// Step 4: Close the connection
$conn->close();
?>
```

Differentiate mysql_fetch_row() and mysql_fetch_array() function.

Feature	mysql_fetch_row()	mysql_fetch_array()
Return Type	Numeric indexed array	Associative array, numeric array, or both
Indexing	Uses only numeric indexes (0, 1, 2...)	Can use both column names and numeric indexes
Flexibility	Less flexible, only numeric indexes	More flexible, allows associative and numeric indexing
Usage	<code>\$row[0], \$row[1]</code>	<code>\$row['column_name']</code> or <code>\$row[0]</code>
Performance	Slightly faster (uses less memory)	Slightly slower (stores multiple index types)
Example Output	<code>["1", "John"]</code>	<code>["1", "John", "id" => "1", "name" => "John"]</code>
Syntax:	<code>\$row = mysql_fetch_row(\$result);</code>	<code>\$row = mysql_fetch_array(\$result, result_type);</code>
Alternative	Use <code>mysqli_fetch_row()</code>	Use <code>mysqli_fetch_array()</code>

What is Mysql_fetch_array()?

`mysql_fetch_array()` is a PHP function that retrieves a row from a MySQL query result and returns it as an associative array, a numeric array, or both.

Syntax:

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

Rajesh Parajuli

☎ +977-9847546279

✉ parajulirajesh2072@gmail.com



Useful Links

- > Home
- > Services
- > Contact Me
- > Terms & Conditions

Worked with

- > BidhyaTech
- > JK Arts
- > MastaSoftsolution
- > Ghodaghodi Multiple Campus

Location

📍 Ghodaghodi Municipality-1,
Kailali Nepal

