Using PHP to Access a Database

Overview

PHP can interact with databases like MySQL, PostgreSQL, and SQLite to perform operations such as querying, inserting, updating, and deleting data. The most common way to connect to a database in PHP is using **MySQLi** or **PDO**.

Key Concepts

1. **MySQLi**: MySQL Improved extension (supports only MySQL).
2. **PDO**: PHP Data Objects (supports multiple databases).

Connecting to a Database

Example: Connecting Using MySQLi

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_db";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

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

Example: Connecting Using PDO

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_db";

try {
```

```
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch (PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

Querying a Database with PHP

Overview

Once connected to a database, you can execute SQL queries to retrieve, insert, update, or delete data.

Key Functions

- **MySQLi**:
  - `query()` : Executes a SQL query.
  - `fetch_assoc()` : Fetches a result row as an associative array.
- **PDO**:
  - `query()` : Executes a SQL query.
  - `fetch()` : Fetches a result row.

Example: Querying Data Using MySQLi

```
<?php
$sql = "SELECT id, name, email FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " . $row["e
    }
} else {
    echo "0 results";
}
?>
```

Example: Querying Data Using PDO

```php
<?php
$sql = "SELECT id, name, email FROM users";
$stmt = $conn->query($sql);

while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " . $row["email
}
?>
```

CRUD Operations Using Forms

Overview

CRUD stands for **Create, Read, Update, and Delete**. These are the four basic operations for managing data in a database.

1. Create (Insert Data)

- Use an HTML form to collect data and insert it into the database.

Example: Inserting Data

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    $email = $_POST['email'];

    // Insert data using MySQLi
    $sql = "INSERT INTO users (name, email) VALUES ('$name', '$email')";
    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
?>
<form method="POST">
    Name: <input type="text" name="name"><br>
    Email: <input type="email" name="email"><br>
    <input type="submit" value="Submit">
</form>
```

2. Read (Retrieve Data)

- Retrieve data from the database and display it in an HTML table.

Example: Retrieving Data

```
<?php
$sql = "SELECT id, name, email FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<table border='1'>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Email</th>
            </tr>";
    while ($row = $result->fetch_assoc()) {
        echo "<tr>
                <td>" . $row["id"] . "</td>
                <td>" . $row["name"] . "</td>
                <td>" . $row["email"] . "</td>
              </tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
?>
```

3. Update (Modify Data)

- Use an HTML form to update existing data in the database.

Example: Updating Data

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $email = $_POST['email'];

    // Update data using MySQLi
    $sql = "UPDATE users SET name='$name', email='$email' WHERE id=$id";
    if ($conn->query($sql) === TRUE) {
        echo "Record updated successfully";
    } else {
        echo "Error updating record: " . $conn->error;
    }
}
?>
```

```
<form method="POST">
    ID: <input type="number" name="id"><br>
    Name: <input type="text" name="name"><br>
    Email: <input type="email" name="email"><br>
    <input type="submit" value="Update">
</form>
```

4. Delete (Remove Data)

- Use an HTML form to delete data from the database.

Example: Deleting Data

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];

    // Delete data using MySQLi
    $sql = "DELETE FROM users WHERE id=$id";
    if ($conn->query($sql) === TRUE) {
        echo "Record deleted successfully";
    } else {
        echo "Error deleting record: " . $conn->error;
    }
}
?>
<form method="POST">
    ID: <input type="number" name="id"><br>
    <input type="submit" value="Delete">
</form>
```

Summary of Key Points

Operation**Description****Create**Insert data into the database using an HTML form and SQL `INSERT` query.**Read**Retrieve data from the database and display it using an HTML table.**Update**Modify existing data in the database using an HTML form and SQL `UPDATE` query.**Delete**Remove data from the database using an HTML form and SQL `DELETE` query.

Practical Questions

1. Create a form to insert a new user into the `users` table.
2. Write a PHP script to display all users from the `users` table in an HTML table.

3. Create a form to update a user's email address based on their ID.
4. Write a PHP script to delete a user from the `users` table based on their ID.