How to Insert Form Data into Database using PHP ?

Requirements:

- **XAMPP Server (Web Server)**
- **HTML**
- **PHP**
- **MySQL**

HTML form: First we create an HTML form that need to take user input from keyboard. HTML form is a document which stores information of a user on a web server using interactive controls. An HTML form contains different kind of information such as username, password, contact number, email id etc.
The elements that are used in an HTML form are check box, input box, radio buttons, submit buttons etc. With the help of these elements, the information of an user is submitted on the web server. The ***form*** tag is used to create an HTML form.

Syntax:

```
<form> Form Elements... </form>
```

or

To pass the values to next page, we use the page name with the following syntax. We can use either GET or POST method to sent data to server.

```
<form action=other_page.php  method= POST/GET>

    Form Elements...

</form>
```

What is a database?

Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

The basic steps to create MySQL database using PHP are:

- Establish a connection to MySQL server from your PHP script .
- If the connection is successful, write a SQL query to create a database and store it in a string variable.
- Execute the query.

Database Connection: The collection of related data is called a database. XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl. It is among the simple

light-weight local servers for website development. In PHP, we can connect to database using localhost XAMPP web server.
Syntax:

```php
<?php

$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "database_name";

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

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

$sqlquery = "INSERT INTO table VALUES
    ('John', 'Doe', 'john@example.com')"

if ($conn->query($sql) === TRUE) {
    echo "record inserted successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

How to get form data: We are going to collect the form data submitted through HTML form. PHP **$_REQUEST** method is a PHP super global variable which is used to collect data after submitting the HTML form.
Syntax:

```php
<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // collect value of input field
    $data = $_REQUEST['val1'];

    if (empty($data)) {
        echo "data is empty";
    } else {
        echo $data;
```

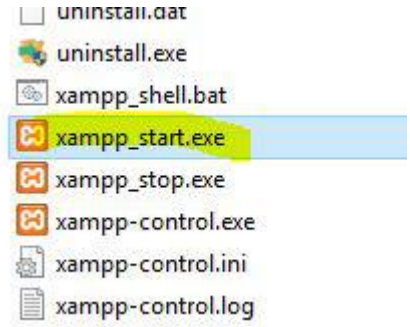
```
    }
}
?>

// Closing the connection.
$conn->close();

?>
```

Complete Steps to Design Project:

- Start XAMPP Server.
- Open localhost/phpmyadmin in your web browser.
- Create database of name **staff** and table of name **college**.
- Write HTML and PHP code in your Notepad in a particular folder.
- Submit data through HTML Form.
- Verify the results.

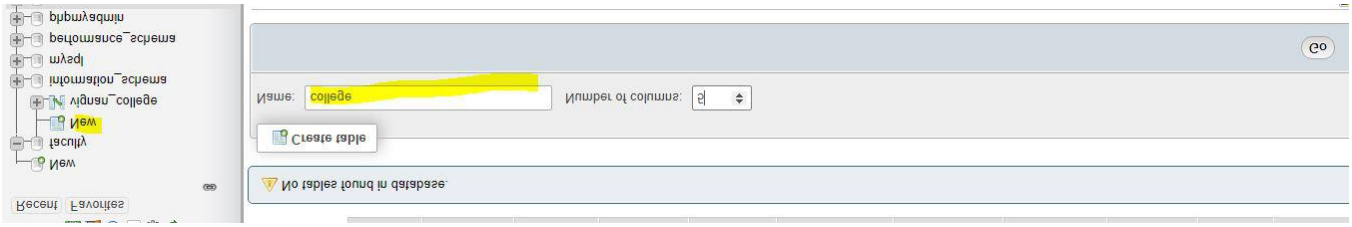
Steps In detail:

- Start XAMPP Server by opening XAMPP and click on XAMPP Start.

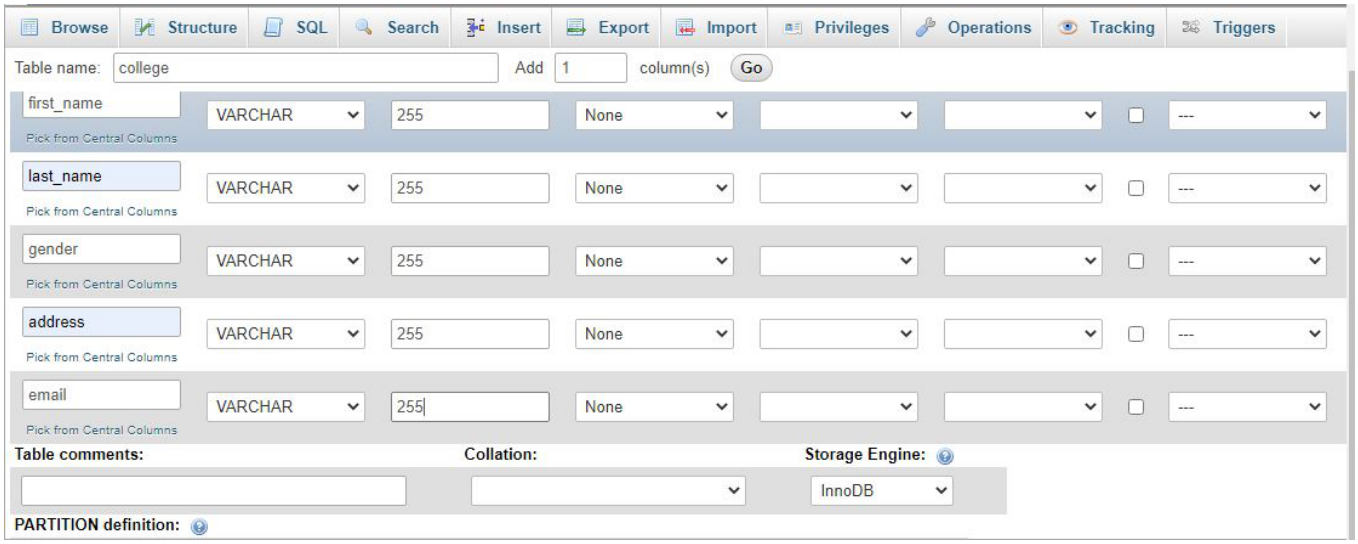


- Open ***localhost/phpmyadmin*** in your web browser and create database with database name as **staff** and click on create.

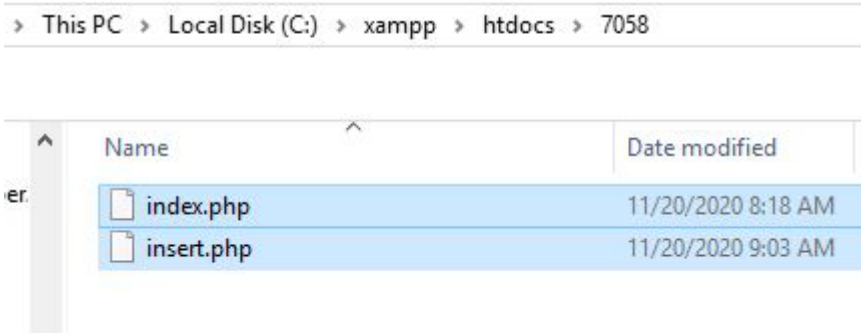
- Then create table name **college**.



- Enter columns and click on save



- Now open Notepad and start writing PHP code and save it as index.php and open other notepad and save it as insert.php Save both files in one folder under **htdocs**.



Filename: index.php

- PHP

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <title>GFG- Store Data</title>
   </head>
   <body>
      <center>
```

```
        <h1>Storing Form data in Database</h1>
        <form action="insert.php" method="post">

<p>
            <label for="firstName">First Name:</label>
            <input type="text" name="first_name" id="firstName">
        </p>


<p>
            <label for="lastName">Last Name:</label>
            <input type="text" name="last_name" id="lastName">
        </p>


<p>
            <label for="Gender">Gender:</label>
            <input type="text" name="gender" id="Gender">
        </p>


<p>
            <label for="Address">Address:</label>
            <input type="text" name="address" id="Address">
        </p>


<p>
            <label for="emailAddress">Email Address:</label>
            <input type="text" name="email" id="emailAddress">
        </p>

            <input type="submit" value="Submit">
        </form>
    </center>
  </body>
</html>
```

Filename: insert.php

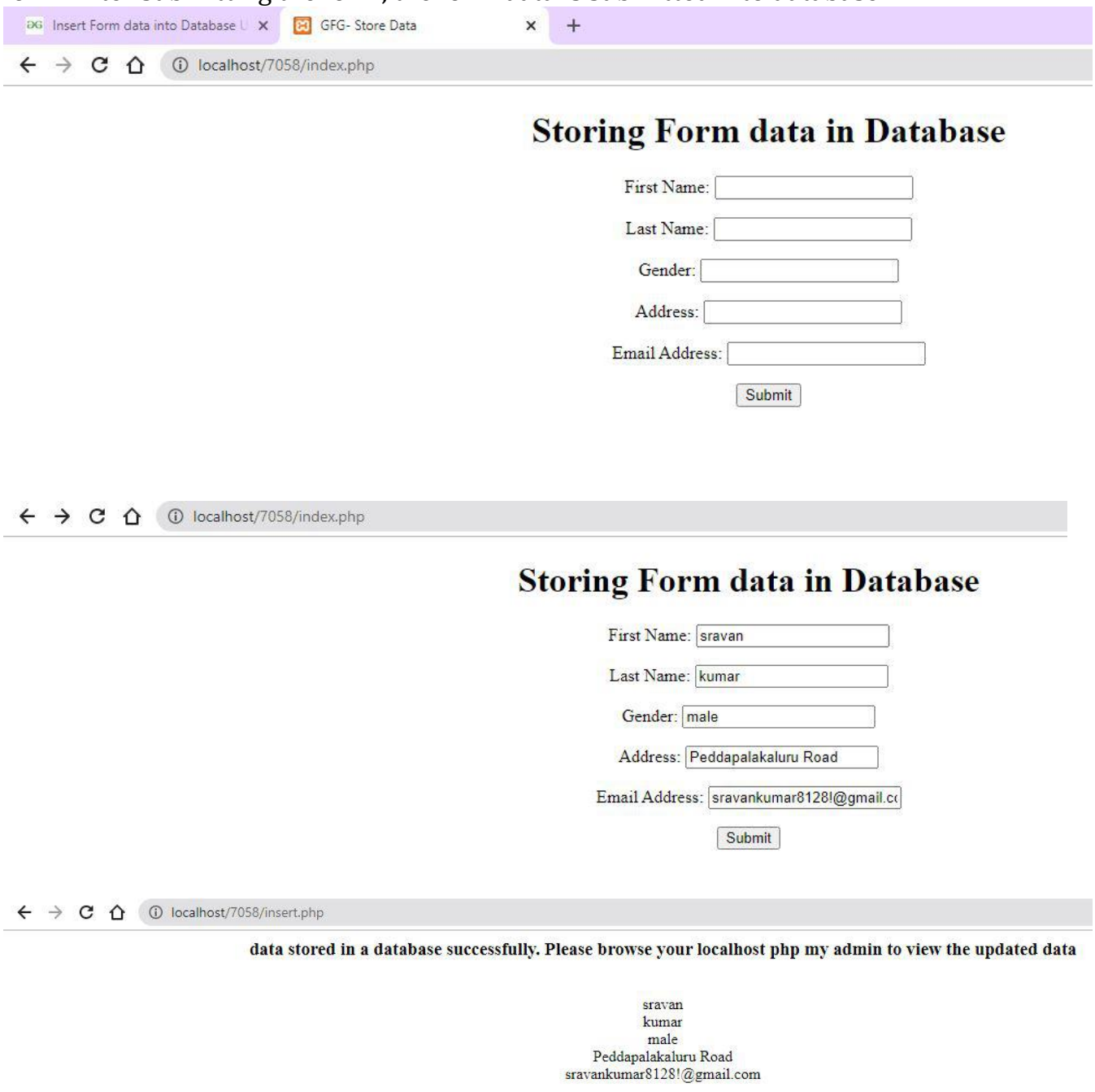
- PHP

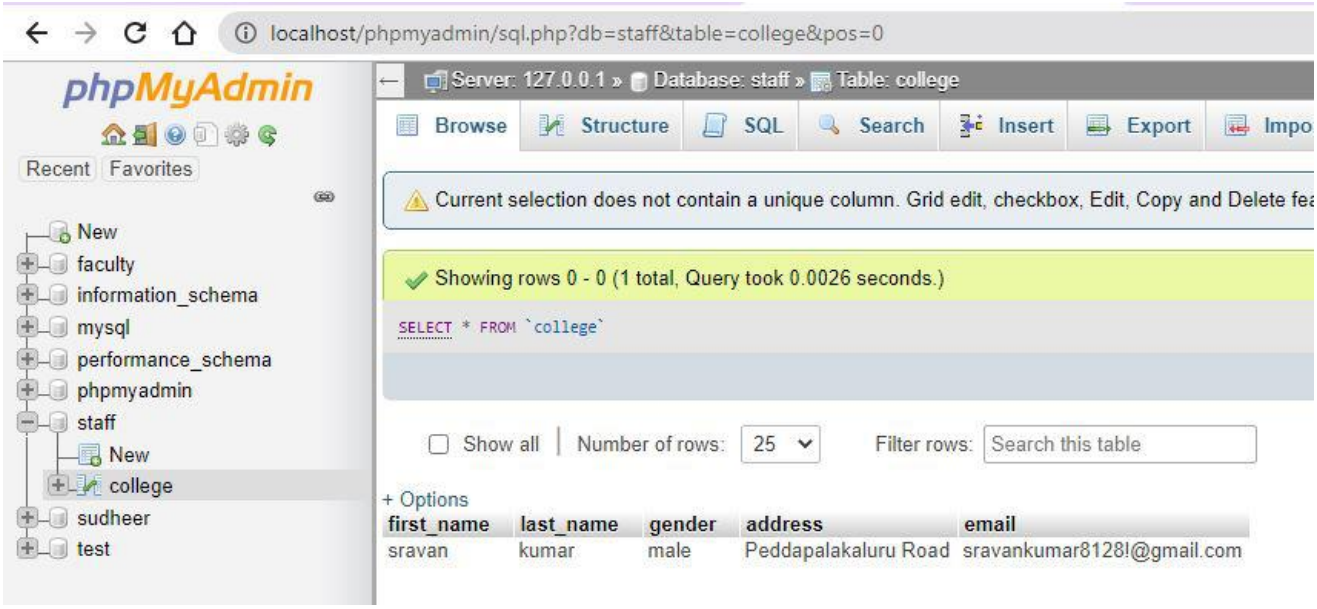
```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Insert Page page</title>
</head>

<body>
    <center>
        <?php

        // servername => localhost
        // username => root
        // password => empty
        // database name => staff
        $conn = mysqli_connect("localhost", "root", "", "staff");

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

        // Taking all 5 values from the form data(input)
        $first_name =  $_REQUEST['first_name'];
        $last_name = $_REQUEST['last_name'];
        $gender =  $_REQUEST['gender'];
        $address = $_REQUEST['address'];
        $email = $_REQUEST['email'];

        // Performing insert query execution
        // here our table name is college
        $sql = "INSERT INTO college  VALUES ('$first_name',
            '$last_name','$gender','$address','$email')";

        if(mysqli_query($conn, $sql)){
            echo "<h3>data stored in a database successfully."
                . " Please browse your localhost php my admin"
                . " to view the updated data</h3>";

            echo nl2br("\n$first_name\n $last_name\n "
                . "$gender\n $address\n $email");
        } else{
            echo "ERROR: Hush! Sorry $sql. "
                . mysqli_error($conn);
        }

        // Close connection
        mysqli_close($conn);
```

```
        ?>
    </center>
</body>

</html>
```

Output: Type ***localhost/7058/index.php*** in your browser, it will display the form. After submitting the form, the form data is submitted into database.





PHP is a server-side scripting language designed specifically for web development. You can learn PHP from the ground up by following this PHP Tutorial and PHP Examples.

PHP | MySQL (Creating Table)

What is a table?

In relational databases, and flat file databases, a table is a set of data elements using a model of vertical columns and horizontal rows, the cell being the unit where a row and column intersect. A table has a specified number of columns, but can have any number of rows.

Creating a MySQL Table Using MySQLi and PDO

We have already learned about creating databases in MySQL from PHP. The steps to create table are similar to creating databases. The difference is instead of creating a new database we will connect to existing database and create a table in that database. To connect to an existing database we can pass an extra variable "database name" while connecting to MySQL.

The CREATE TABLE statement is used to create a table in MySQL.

In this article, a table named "employees", with four columns: "id", "firstname", "lastname" and "email" will be created.

The data types that will be used are :

1. **VARCHAR:**Holds a variable length string that can contain letters, numbers, and special characters. The maximum size is specified in parenthesis.

2. **INT :**he INTEGER data type accepts numeric values with an implied scale of zero. It stores any integer value between -2147483648 to 2147483647.

The attributes that are used along with data types in this article are:

1. **NOT NULL:** Each row must contain a value for that column, null values are not allowed.
2. **PRIMARY KEY:** Used to uniquely identify the rows in a table. The column with PRIMARY KEY setting is often an ID number.

Creating tables in three different versions are described below:

Creating table using MySQLi Object-oriented Procedure

Syntax :

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


// checking connection
$conn = new mysqli($servername, $username, $password,
  $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql code to create table
$sql = "CREATE TABLE employees(
        id INT(2)  PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50)
        )";

if ($conn->query($sql) === TRUE) {
    echo "Table employees created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```

Output :

Table employees created successfully

Creating table using MySQLi Procedural procedure

Syntax :

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

// Checking connection
$conn = mysqli_connect($servername, $username, $password,
 $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// sql code to create table
$sql = "CREATE TABLE employees (
        id INT(2)  PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50)
        )";

if (mysqli_query($conn, $sql)) {
    echo "Table employees created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

Output :

Table employees created successfully

Creating table using PDO procedure

Syntax :

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

```
$dbname = "newDB";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname",
                                                $username,
  $password);


    // setting the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE,
  PDO::ERRMODE_EXCEPTION);

    // sql code to create table
    $sql = "CREATE TABLE employees (
            id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            firstname VARCHAR(30) NOT NULL,
            lastname VARCHAR(30) NOT NULL,
            email VARCHAR(50)
            )";

    // using exec() because no results are returned
    $conn->exec($sql);
    echo "Table employees created successfully";
    }
catch(PDOException $e)
    {
    echo $sql . "
  " . $e->getMessage();
    }

$conn = null;
?>
```

Output :

Table employees created successfully

PHP | MySQL Select Query

The SQL **SELECT** statement is used to select the records from database tables.

Syntax :

The basic syntax of the select clause is –

```
SELECT column1_name,column2_name,columnN_name FROM table_name;
```

To select all columns from the table, the character is used.

```
SELECT * FROM table_name
```

Implementation of the Select Query :
Let us consider the following table ' Data ' with three columns ' FirstName ', ' LastName ' and ' Age '.

Firstname	Lastname	Age
Ram	Singh	26
Lakshman	Sharma	29
Priya	Mathur	21
Ram	Agarwal	35
Sidharth	Goyal	45

To select all the data stored in the ' Data ' table, we will use the code mentioned below.

SELECT Query using Procedural Method :

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

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

$sql = "SELECT * FROM Data";
if ($res = mysqli_query($link, $sql)) {
    if (mysqli_num_rows($res) > 0) {
        echo "<table>";
        echo "<tr>";
        echo "<th>Firstname</th>";
        echo "<th>Lastname</th>";
        echo "<th>age</th>";
        echo "</tr>";
        while ($row = mysqli_fetch_array($res)) {
            echo "<tr>";
            echo "<td>".$row['Firstname']."</td>";
            echo "<td>".$row['Lastname']."</td>";
            echo "<td>".$row['Age']."</td>";
            echo "</tr>";
        }
        echo "</table>";
```

```
        mysqli_free_result($res);
    }
    else {
        echo "No matching records are found.";
    }
}
else {
    echo "ERROR: Could not able to execute $sql. "
                                .mysqli_error($link);
}
mysqli_close($link);
?>
```

Output :

Firstname	Lastname	Age
Ram	Singh	26
Lakshman	Sharma	29
Priya	Mathur	21
Ram	Agarwal	35
Sidharth	Goyal	45

Code Explanation:

1. The "res" variable stores the data that is returned by the function *mysql_query()*.
2. Everytime *mysqli_fetch_array()* is invoked, it returns the next row from the *res()* set.
3. The while loop is used to loop through all the rows of the table "data".

SELECT Query using Object Oriented Method :

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

if ($mysqli === false) {
    die("ERROR: Could not connect. "
                            .$mysqli->connect_error);
}

$sql = "SELECT * FROM Data";
if ($res = $mysqli->query($sql)) {
    if ($res->num_rows > 0) {
        echo "<table>";
        echo "<tr>";
        echo "<th>Firstname</th>";
```

```
        echo "<th>Lastname</th>";
        echo "<th>Age</th>";
        echo "</tr>";
        while ($row = $res->fetch_array())
        {
            echo "<tr>";
            echo "<td>".$row['Firstname']."</td>";
            echo "<td>".$row['Lastname']."</td>";
            echo "<td>".$row['Age']."</td>";
            echo "</tr>";
        }
        echo "</table>";
        $res->free();
    }
    else {
        echo "No matching records are found.";
    }
}
else {
    echo "ERROR: Could not able to execute $sql. "
                                            .$mysqli->error;
}
$mysqli->close();
?>
```

Output :

Firstname	Lastname	Age
Ram	Singh	26
Lakshman	Sharma	29
Priya	Mathur	21
Ram	Agarwal	35
Sidharth	Goyal	45

SELECT Query using PDO Method :

```
<?php
try {
    $pdo = new PDO("mysql:host = localhost;
                    dbname=mydb", "root", "");
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
                       PDO::ERRMODE_EXCEPTION);
}
catch (PDOException $e) {
```

```
    die("ERROR: Could not connect. ".$e->getMessage());
}
try {
    $sql = "SELECT * FROM Data";
    $res = $pdo->query($sql);
    if ($res->rowCount() > 0) {
        echo "<table>";
        echo "<tr>";
        echo "<th>Firstname</th>";
        echo "<th>Lastname</th>";
        echo "<th>Age</th>";
        echo "</tr>";
        while ($row = $res->fetch()) {
            echo "<tr>";
            echo "<td>".$row['Firstname']."</td>";
            echo "<td>".$row['Lastname']."</td>";
            echo "<td>".$row['Age']."</td>";
            echo "</tr>";
        }
        echo "</table>";
        unset($res);
    }
    else {
        echo "No matching records are found.";
    }
}
catch (PDOException $e) {
    die("ERROR: Could not able to execute $sql. "
                                .$e->getMessage());
}
unset($pdo);
?>
```

Output :

Firstname	Lastname	Age
Ram	Singh	26
Lakshman	Sharma	29
Priya	Mathur	21
Ram	Agarwal	35
Sidharth	Goyal	45

DATABASE CREATION

We can execute the query from our PHP script in 3 different ways as described below:

1. **Using MySQLi Object-oriented procedure**: If the MySQL connection is established using Object-oriented procedure then we can use the query() function of mysqli class to execute our query as described in the below syntax.
   Syntax:

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


// Creating a connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Creating a database named newDB
$sql = "CREATE DATABASE newDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully with the name
newDB";
} else {
    echo "Error creating database: " . $conn->error;
}

// closing connection
$conn->close();
?>
```

   Note: Specify the three arguments servername, username and password to the mysqli object whenever creating a database.
   Output:

```
Database created successfully with the name newDB
```

2. **Using MySQLi Procedural procedure**: If the MySQL connection is established using procedural procedure then we can use the mysqli_query() function of PHP to execute our query as described in the below syntax.
   Syntax:

```
<?php
```

```
$servername = "localhost";

$username = "username";

$password = "password";


// Creating connection
$conn = mysqli_connect($servername, $username, $password);
// Checking connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Creating a database named newDB
$sql = "CREATE DATABASE newDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully with the name
newDB";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}

// closing connection
mysqli_close($conn);
?>
```

Output:

Database created successfully with the name newDB

3. **Using PDO procedure**: If the MySQL connection is established using PDO procedure then we can execute our query as described in the below syntax.

Syntax:

```
<?php

$servername = "localhost";

$username = "username";

$password = "password";


try {

    $conn = new PDO("mysql:host=$servername;dbname=newDB",
$username, $password);

    // setting the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE newDB";
```

```
    // using exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully with the name
newDB";
    }
catch(PDOException $e)
    {
    echo $sql . "
" . $e->getMessage();
    }
$conn = null;
?>
```

Note: The exception class in PDO is used to handle any problems that may occur in our database queries. If an exception is thrown within the try{ } block, the script stops executing and flows directly to the first catch(){ } block.

Output:

```
Database created successfully with the name newDB
```

PHP | MySQL UPDATE Query

The MySQL **UPDATE** query is used to update existing records in a table in a MySQL database.

- It can be used to update one or more field at the same time.
- It can be used to specify any condition using the WHERE clause.

Syntax :

The basic syntax of the Update Query is –

```
UPDATE table_name SET column1=value1,column2=value2, WHERE column_n
```

Implementation of Where Update Query :

Let us consider the following table "Data" with four columns 'ID', 'FirstName', 'LastName' and 'Age'.

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
201	Priya	Mathur	21
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

To update the “Age” of a person whose “ID” is 201 in the “Data” table, we can use the following code :

Update Query using Procedural Method :

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

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

$sql = "UPDATE data SET Age='28' WHERE id=201";
if(mysqli_query($link, $sql)){
    echo "Record was updated successfully.";
} else {
    echo "ERROR: Could not able to execute $sql. "
                              . mysqli_error($link);
}
mysqli_close($link);
?>
```

Output :
Table After Updation –

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
201	Priya	Mathur	28
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

The output on Web Browser :

```
Record was updated successfully.
```

Update Query using Object Oriented Method :

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

if($mysqli === false){
    die("ERROR: Could not connect. "
            . $mysqli->connect_error);
}

$sql = "UPDATE data SET Age='28' WHERE id=201";
```

```
if($mysqli->query($sql) === true){
    echo "Records was updated successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. "
                                        . $mysqli->error;
}
$mysqli->close();
?>
```

Output :
Table After Updation –

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
201	Priya	Mathur	28
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

The output on Web Browser :

```
Record was updated successfully.
```

Update Query using PDO Method :

```
<?php
try{
    $pdo = new PDO("mysql:host=localhost;
                    dbname=Mydb", "root", "");
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
                        PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e){
    die("ERROR: Could not connect. "
                    . $e->getMessage());
}

try{
    $sql = "UPDATE data SET Age='28' WHERE id=201";
    $pdo->exec($sql);
    echo "Records was updated successfully.";
} catch(PDOException $e){
    die("ERROR: Could not able to execute $sql. "
                                . $e->getMessage());
}
unset($pdo);
?>
```

Output :
Table After Updation –

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
201	Priya	Mathur	28
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

The output on Web Browser :

```
Record was updated successfully.
```

PHP | MySQL Delete Query

The **DELETE** query is used to delete records from a database table.
It is generally used along with the "Select" statement to delete only those records that satisfy a specific condition.

Syntax :
The basic syntax of the Delete Query is –

```
DELETE FROM table_name WHERE column_name=colname_value
```

Let us consider the following table "Data" with four columns ' ID ', ' FirstName ', ' LastName ' and ' Age '.

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
201	Priya	Mathur	21
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

To delete the record of the person whose ID is 201 from the ' Data ' table, the following code can be used.

Delete Query using Procedural Method :

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

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

```
}

$sql = "DELETE FROM Data WHERE ID=201";
if(mysqli_query($link, $sql)){
    echo "Record was deleted successfully.";
}
else{
    echo "ERROR: Could not able to execute $sql. "
                                    . mysqli_error($link);
}
mysqli_close($link);
?>
```

Output :
Table after updation –

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

The output on Web Browser :

```
Record was deleted successfully.
```

Delete Query using Object Oriented Method :

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

if($mysqli === false){
    die("ERROR: Could not connect. " . $mysqli->connect_error);
}

$sql = "DELETE FROM Data WHERE ID=201";
if($mysqli->query($sql) === true){
    echo "Record was deleted successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. "
                                        . $mysqli->error;
}

$mysqli->close();
?>
```

Output :
Table After Updation –

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

The output on Web Browser :

```
Record was deleted successfully.
```

Delete Query using PDO Method :

```
<?php
try{
    $pdo = new PDO("mysql:host=localhost;
                        dbname=Mydb", "root", "");
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
                           PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e){
    die("ERROR: Could not connect. " . $e->getMessage());
}

try{
    $sql = "DELETE FROM Data WHERE ID=201";
    $pdo->exec($sql);
    echo "Record was deleted successfully.";
} catch(PDOException $e){
    die("ERROR: Could not able to execute $sql. "
                                  . $e->getMessage());
}
unset($pdo);
?>
```

Output :
Table After Updation –

ID	Firstname	Lastname	Age
101	Lakshman	Sharma	29
301	Ram	Singh	26
401	Ram	Agarwal	35
501	Sidharth	Goyal	45

The output on Web Browser :

```
Record was deleted successfully.
```