

CÓMO CONECTAR PHP CON BASES DE DATOS MYSQL

PASO I: CREAR UNA BASE DE DATOS

Crearemos una base de datos en Hostinger. Para ello:

1. Busca el menú Bases de datos MySQL en la sección Bases de datos.
2. Completa los campos necesarios y presiona Crear.

⊕ Crear Nueva Base de Datos MySQL y Usuario de la Base de Datos

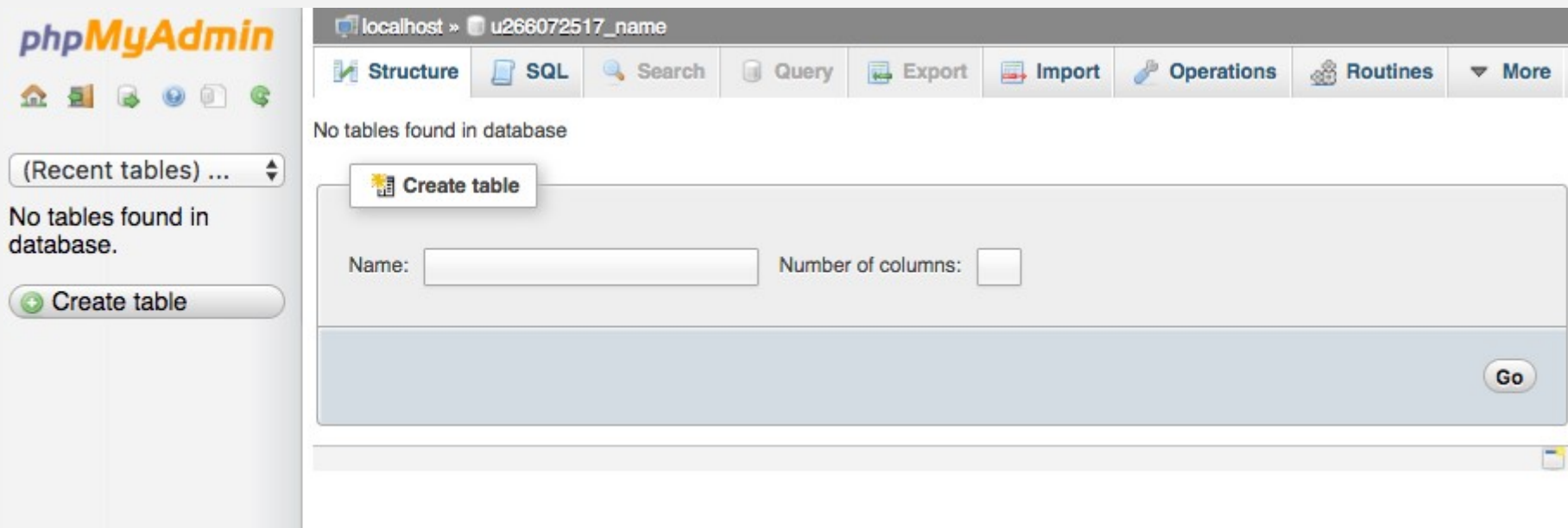
NOMBRE DE BASE DE DATOS MYSQL	<input type="text" value="u309565199_ name"/>
USUARIO MYSQL	<input type="text" value="u309565199_ user"/>
CONTRASEÑA	<input type="password" value="....."/> ⚙ Generar <p>Fortaleza de Contraseña</p> <div style="width: 100%; height: 10px; background-color: green; border-radius: 5px;"></div>

✓ Crear

PASO I: CREAR UNA BASE DE DATOS

Ahora procederemos a crear una tabla. Para ello:

1. Para crear una tabla usaremos la opción phpMyAdmin, que encuentras en tu panel de control de hosting
2. Click en la base de datos que hemos creado anteriormente y que usaremos con nuestro formulario.
3. Creamos la tabla



The screenshot displays the phpMyAdmin web interface. The browser address bar shows 'localhost » u266072517_name'. The top navigation bar includes tabs for 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', 'Routines', and 'More'. The main content area shows 'No tables found in database' and a 'Create table' button. Below this, there is a form with two input fields: 'Name:' and 'Number of columns:'. A 'Go' button is located at the bottom right of the form. On the left sidebar, there is a '(Recent tables) ...' dropdown menu, a 'No tables found in database.' message, and a 'Create table' button.

PASO 2: CREAR UNA TABLA

4. Damos el formato necesario a la tabla, es decir, definimos las columnas y tipos de dato.

Table name: Add column(s)

Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comment
<input type="text" value="ID"/>	<input type="text" value="INT"/>	<input type="text" value="4"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="PRIMARY"/>	<input checked="" type="checkbox"/>	<input type="text"/>
<input type="text" value="name"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="50"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text" value="lastname"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="50"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text" value="email"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="50"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>	<input type="text"/>

Table comments:

Storage Engine:

Collation:

PASO 3: CREAR EL FORMULARIO

Hay dos métodos que puedes usar para **INSERTAR** datos en tu base de datos **MySQL**. El método **PHP MySQLi** y el método **PHP Data Object o PDO**.

Nosotros usaremos el método **PHP MySQLi**. Para ello, se debe establecer una conexión con la base de datos. Una vez hecho esto, podemos proceder con la consulta **MySQL INSERT**.

La diferencia entre ambos métodos consiste en que **PHP MySQLi** está diseñado específicamente para MySQL, mientras que **PDO** es una capa de abstracción que puede trabajar con diferentes bases de datos.

A continuación, veamos un ejemplo de formulario sencillo y el archivo PHP con el código de ejemplo completo con la conexión básica y los métodos de inserción.

PASO 3: CREAR EL FORMULARIO

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Formulario de contacto</title>
</head>
<body>
  <form action="procesar_formulario.php" method="post">
    <label for="nombre_apellidos">Nombre y apellidos:</label>
    <input type="text" id="nombre_apellidos" name="nombre_apellidos"
required>
    <br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>
    <br>
    <label for="mensaje">Mensaje:</label>
    <textarea id="mensaje" name="mensaje" required></textarea>
    <br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

PASO 4: CREAR EL ARCHIVO PHP

```
<?php
// Configuración de la conexión a la base de datos
$servername = "localhost";
$username = "tu_usuario";
$password = "tu_contraseña";
$dbname = "bdform";

// Crear conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar conexión
if ($conn->connect_error) {
    die("Error de conexión: " . $conn->connect_error);
}
```

```
// Obtener los datos del formulario
$nombre_apellidos = $_POST['nombre_apellidos'];
$email = $_POST['email'];
$mensaje = $_POST['mensaje'];
// Preparar la consulta para insertar los datos en la base de datos
$sql = "INSERT INTO Base_de_Datos (nombre_apellidos, email, mensaje) VALUES (?, ?, ?)";
// Preparar la declaración
$stmt = $conn->prepare($sql);
// Vincular los parámetros
$stmt->bind_param("sss", $nombre_apellidos, $email, $mensaje);
// Ejecutar la declaración
if ($stmt->execute()) {
    echo "Datos enviados correctamente.";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
// Cerrar la declaración y la conexión
$stmt->close();
$conn->close();
?>
```

PASO 4: CREAR EL ARCHIVO PHP

La primera parte del código (líneas 3 a 14) tiene como objetivo la conexión a la base de datos.

En la línea 21 encontramos la parte más importante del código **PHP**, ya que es la que logra insertar datos en la base de datos **MySQL**. **INSERT INTO** es una declaración que agrega datos a la tabla de la base de datos especificada.

```
// Preparar la consulta para insertar los datos en la base de datos
```

```
$sql = "INSERT INTO Base_de_Datos(nombre_apellidos, email, mensaje)  
VALUES (?, ?, ?)";
```

Algo más que vale la pena destacar es que acabamos de ejecutar una consulta **SQL** (**SQL query**, por su nombre en inglés) usando código **PHP**, las consultas **SQL** deben establecerse entre comillas. En nuestro ejemplo, todo lo que está entre comillas y después de **\$sql =** es una consulta **SQL**.

PASO 4: CREAR EL ARCHIVO PHP

La siguiente parte del código (líneas 20 a 22) verifica si nuestra consulta fue exitosa o no:

```
// Ejecutar la declaración
if ($stmt->execute()) {
    echo "Datos enviados correctamente.";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Por último, la línea 30 cierra la conexión con la base de datos, lo que es necesario para liberar recursos y garantizar una gestión adecuada de la conexión con la base de datos.

```
// Cerrar la conexión
$conn->close();
```

PASO 5: CONECTAR

Por último, debemos crear la conexión remota con nuestro servidor MySQL desde la siguiente pantalla. Desde ahí también podemos obtener el hostname MySQL...

MySQL remoto

🏠 - Hosting - fusionaweb.es - Bases de datos - MySQL remoto

+ Crear conexión de base de datos remota

Use esta función para conectarte a nuestro servidor MySQL desde tu computadora o desde algún otro servidor. Simplemente agrega la dirección IP desde la que te conectarás a nuestro servidor MySQL y elige en la lista el nombre de la base de datos a la que deseas conectarte.

El hostname de nuestro servidor MySQL es: srv965.hstgr.io o puedes usar esta IP como tu hostname: **89.116.147.154**

IP (IPv4 o IPv6) *

Cualquier Host

Base de datos *

Elige una opción ▼

✓ Crear