

PHP+PDO+MySQL

- × Database;**
- × Conexão;**
- × Criando um banco;**
- × Criando uma tabela;**
- × Inserindo dados;**
- × Recuperando o ultimo ID;**
- × Inserindo múltiplos dados;**
- × Proteção conta SQL injections**

PHP+PDO+MySQL

× Database:

- MySQL é a mais popular banco de dados usado com PHP;
- Usado na WEB;
- Roda no lado do servidor;
- Suporta tanto pequenas quanto grandes aplicações;
- Rápido, e fácil de usar
- Suporta comandos SQL padrão;
- Compila em várias plataformas;

PHP+PDO+MySQL

× Database:

- Gratuito para download e para uso;
- Desenvolvido, distribuído e mantido pela Oracle Corporation;
- Os dados são armazenados em tabelas;



PHP+PDO+MySQL

× Conexão:

- A partir do PHP5 podemos trabalhar com MySQL usando, MySQLi Extension ou PDO (PHP Data Objects), que será a forma adotada no curso;
- Ambas as formas tem vantagens, porem PDO suporta 12 diferentes tipos de banco s de dados, dessa forma, a migração de uma banco para outra fica muito mais simples usando PDO. AS duas formas também possuem proteção de SQL injections, fator, muito importante segurança dentro de aplicações WEB.

PHP+PDO+MySQL

× Conexão:

```
<?php
```

```
$servername = "localhost";
```

```
$username = "username";
```

```
$password = "password";
```

```
try {
```

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

```
    // set the PDO error mode to exception
```

```
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
    echo "Connected successfully";
```

```
} catch(PDOException $e) {
```

```
    echo "Connection failed: " . $e->getMessage();
```

```
}
```

```
?>
```

PHP+PDO+MySQL

× Criando um banco:

- É necessário ter privilégios de ADM para criar ou deletar um BD;

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

try {
    $conn = new PDO("mysql:host=$servername", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

PHP+PDO+MySQL

x Criando tabela:

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

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to create table
    $sql = "CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    )";
```

PHP+PDO+MySQL

x Criando tabela:

```
// use exec() because no results are returned
$conn->exec($sql);
echo "Table MyGuests created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}
```

```
$conn = null;
```

```
?>
```


PHP+PDO+MySQL

x Inserindo dados:

- Após a tabela ser criada é possível inserir os dados;

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

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "New record created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

PHP+PDO+MySQL

x Recuperando último dado:

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

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    $last_id = $conn->lastInsertId();
    echo "New record created successfully. Last inserted ID is: " . $last_id;
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

PHP+PDO+MySQL

x Inserindo Múltiplos Dados:

```
<?php
```

```
$servername = "localhost";
```

```
$username = "username";
```

```
$password = "password";
```

```
$dbname = "myDB";
```

```
// Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
```

```
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
```

```
}
```

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('John', 'Doe', 'john@example.com');";
```

```
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('Mary', 'Moe', 'mary@example.com');";
```

```
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('Julie', 'Dooley', 'julie@example.com');";
```

```
if ($conn->multi_query($sql) === TRUE) {  
    echo "New records created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}
```

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

```
?>
```

PHP+PDO+MySQL

x **Proteção contra SQL Injection:**

- Prepared Statement (declarações preparadas) são uteis contra SQL Injections;
- Primeiro passo: Criar um template de como será a declaração a ser enviada pro BD
- Segundo passo: o BD analisa, compila e executa a consulta no modelo do template e armazena o resultado sem executá-lo;
- Terceiro passo: a aplicação vincula os valores aos parâmetros e a instrução é executada;

PHP+PDO+MySQL

xProteção contra SQL Injection:

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

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // prepare sql and bind parameters
    $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (:firstname, :lastname, :email)");
    $stmt->bindParam(':firstname', $firstname);
    $stmt->bindParam(':lastname', $lastname);
    $stmt->bindParam(':email', $email);
```

PHP+PDO+MySQL

xProteção contra SQL Injection:

```
// insert a row
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

// insert another row
$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();

// insert another row
$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();

echo "New records created successfully";
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
?>
```

PHP+PDO+MySQL

x Referências:

- https://www.w3schools.com/php/php_mysql_intro.asp