

## PDO Tutorial

### Banco de dados

Criar o banco de dados teste e importar para o mesmo as linhas abaixo:

```
DROP TABLE IF EXISTS countries;
CREATE TABLE countries(id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255), population INT);
```

```
INSERT INTO countries(name, population) VALUES('China', 1382050000);
INSERT INTO countries(name, population) VALUES('India', 1313210000);
INSERT INTO countries(name, population) VALUES('USA', 324666000);
INSERT INTO countries(name, population) VALUES('Indonesia', 260581000);
INSERT INTO countries(name, population) VALUES('Brazil', 207221000);
INSERT INTO countries(name, population) VALUES('Pakistan', 196626000);
INSERT INTO countries(name, population) VALUES('Nigeria', 186988000);
INSERT INTO countries(name, population) VALUES('Bangladesh', 162099000);
INSERT INTO countries(name, population) VALUES('Nigeria', 186988000);
INSERT INTO countries(name, population) VALUES('Russia', 146838000);
INSERT INTO countries(name, population) VALUES('Japan', 126830000);
```

### Conexão simples

```
<?php
```

```
$dsn = "mysql:host=localhost;dbname=testes";
$user = "root";
$password = "root";
```

```
$default_options = [
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
];
$options = array_replace($default_options, $options);
$pdo = new PDO($dsn, $user, $password, $options);
```

```
if($pdo){
    echo 'Conectou';
    exit;
}else{
    echo 'Não conectou';
    exit;
}
```

### Select simples com query()

PDO query() executa uma consulta SQL em uma chamada de função e retorna o resultado caso exista.

```
<?php
```

```
$dsn = "mysql:host=localhost;dbname=testes";
```

```

$user = "root";
$password = "root";

$pdo = new PDO($dsn, $user, $password);
$stmt = $pdo->query("SELECT 2*3");

$version = $stmt->fetch();
echo $version[0] . PHP_EOL;

```

### Função exec()

Executa uma consulta e retorna o número de registros afetados

```
<?php
```

```

$dsn = "mysql:host=localhost;dbname=testes";
$user = "root";
$password = "root";

$pdo = new PDO($dsn, $user, $password);
$id = 12;

$rows = $pdo->exec("DELETE FROM countries WHERE id IN (1, 2, 3)");

echo "The statement affected $rows rows\n";

```

### Alguns Estilos do fetch()

O parâmetro style do fetch() controla como o próximo registro deve retornar para quem o chamou.

PDO::FETCH\_ASSOC retorna um array indexado pelo nome da coluna

PDO::FETCH\_BOTH retorna um array indexado por ambos, nome da coluna e número da coluna, este é o default

PDO::FETCH\_OBJ retorna os registros no formato de objetos: \$campo->valor

### Exemplo de Estilo ASSOC

```

<?php
$dsn = "mysql:host=localhost;dbname=testes";
$user = "root";
$password = "root";
$pdo = new PDO($dsn, $user, $password);

$stmt = $pdo->query("SELECT * FROM countries");
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC); // Mostra os registros como arrays associativos

foreach($rows as $row) {
    print $row['id']. ' - ' . $row['name']. ' - ' . $row['population']. '<br>';
}

```

### Exemplo com estilo Objeto

```

<?php
$dsn = "mysql:host=localhost;dbname=testes";

```

```

$user = "root";
$password = "root";
$pdo = new PDO($dsn, $user, $password);

$stmt = $pdo->query("SELECT * FROM countries");
$objs = $stmt->fetchAll(PDO::FETCH_OBJ); // Mostra os registros como arrays associativos

foreach($objs as $obj) {
    print $obj->id.' - '. $obj->name.' - '. $obj->population.'<br>';
}

```

### Parâmetros tipo binding/vínculo

Com parâmetros tipo binding as consultas SQL são construídas dinamicamente. O usuário provê algumas entradas e estas entradas são montadas na consulta. Precisamos ser cautelosos ao lidar com entradas de usuários, elas contêm sérias implicações de segurança. A forma recomendada é construir consultas SQL dinamicamente usando parâmetros binding.

O PDO contém os métodos bindParam() e bindValue() para criar consultas parametrizadas. O PDO permite vincular/binding dados como ? (ponto de interrogação) e parâmetros nomeados.

#### Exemplos:

```

$id = 12;

$stmt = $pdo->prepare("SELECT * FROM countries WHERE id = ?");
$stmt->bindValue(1, $id);
$stmt->execute();
$row = $stmt->fetch(PDO::FETCH_ASSOC);

echo "Id: " . $row['id'] . PHP_EOL;
echo "Name: " . $row['name'] . PHP_EOL;
echo "Population: " . $row['population'] . PHP_EOL;

```

ou

```

$db->run("INSERT INTO users VALUES (null, ?,?,?)", [$name, $email, $password]);

```

### Nameds

```

$id = 12;

$stmt = $pdo->prepare("SELECT * FROM countries WHERE id = :id");
$stmt->bindParam(":id", $id, PDO::PARAM_INT);
$stmt->execute();

$row = $stmt->fetch(PDO::FETCH_ASSOC);

echo "Id: " . $row['id'] . PHP_EOL;
echo "Name: " . $row['name'] . PHP_EOL;
echo "Population: " . $row['population'] . PHP_EOL;

```

### Número de Registros

O número de registros de uma consulta/tabela é retornado por rowCount()

Exemplo:

```
<?php
$dsn = "mysql:host=localhost;dbname=testes";
$user = "root";
$password = "root";
$pdo = new PDO($dsn, $user, $password);

$stmt = $pdo->query("SELECT * FROM countries");

print $stmt->rowCount();
```

## Transações

Uma transação é uma operação atômica entre um ou mais bancos de dados. Ou executa todas as operações ou nenhuma.

O comando `beginTransaction()` inicia uma nova transação

O comando `commit()` conclui uma transação

O comando `rollback()` cancela as transações até o momento

## Exemplo

```
<?php

$dsn = "mysql:host=localhost;dbname=mydb";
$user = "user12";
$password = "12user";

$pdo = new PDO($dsn, $user, $password);

try {

    $pdo->beginTransaction();
    $stmt = $pdo->exec("INSERT INTO countries(name, population) VALUES ('Iraq', 38274000)");
    $stmt = $pdo->exec("INSERT INTO countries(name, population) VALUES ('Uganda',
37673800)");

    $pdo->commit();

} catch(Exception $e) {

    $pdo->rollback();
    throw $e;
}
```

## Recebendo metadados

Metadados são informações sobre os dados no banco de dados. Um exemplo seria receber a quantidade de campos de uma tabela.

## Exemplo recebendo o número de colunas da tabela

```
<?php
$dsn = "mysql:host=localhost;dbname=testes";
```

```

$user = "root";
$password = "root";
$pdo = new PDO($dsn, $user, $password);

$stmt = $pdo->query("SELECT * FROM countries");

print $stmt->columnCount();

```

O método `getAttribute()` recebe os atributos de uma conexão com o banco

### Exemplo

```

<?php

$dsn = "mysql:host=localhost;dbname=mydb";
$user = "user12";
$password = "12user";

$pdo = new PDO($dsn, $user, $password);

$driver = $pdo->getAttribute(PDO::ATTR_DRIVER_NAME);
$server_version = $pdo->getAttribute(PDO::ATTR_SERVER_VERSION);
$autocommit_mode = $pdo->getAttribute(PDO::ATTR_AUTOCOMMIT);

echo "Driver: $driver\n";
echo "Server version: $server_version\n";
echo "Autocommit mode: $autocommit_mode\n";

```

O método `getColumnMeta()` recebe dados sobre as colunas em uma consulta.

### Exemplo

```

<?php
$dsn = "mysql:host=localhost;dbname=testes";
$user = "root";
$password = "root";
$pdo = new PDO($dsn, $user, $password);

$stmt = $pdo->query("SELECT * FROM countries");

$col_meta = $stmt->getColumnMeta(0);

echo "Table name: {$col_meta["table"]}\n<br>";
echo "Column name: {$col_meta["name"]}\n<br>";
echo "Column length: {$col_meta["len"]}\n<br>";
echo "Column flags: {$col_meta["flags"][0]} {$col_meta["flags"][1]} \n<br>";

```

### Códigos de Erro

Para capturar o código de erro do nome do banco errado

```

<?php
$dns = "mysql:host=localhost;dbname=testes2";
$user = "root";
$password = "root";

try {
    $pdo = new PDO($dns, $user, $password);
    $stm = $pdo->query("SELECT * FROM countries");
} catch (Exception $e) {
    echo "Código do erro: ", $e->getCode();
    echo "<br>Mensagem: ", $e->getMessage();
}

```

### **Recebendo registros com fetch()**

fetch() recebe um registro

fetchAll() recebe todos os registros da tabela/consulta

```

<?php
$dns = "mysql:host=localhost;dbname=testes";
$user = "root";
$password = "root";
$pdo = new PDO($dns, $user, $password);

$stm = $pdo->query("SELECT * FROM countries");

$result = $stm->fetchAll();

foreach($result as $reg) {
    print $reg['id']. ' - '. $reg['name']. '<br>';
}

$total = $stm->rowCount();

print "<br>Total de registros: $total";

```

### **Uso do try/catch**

É recomendado o uso do try/catch nas operações de alteração: insert, delete e update.

### **Exemplo**

```

<?php
try {
    $pdo = new PDO('mysql:host=localhost;dbname=meuBancoDeDados', $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $pdo->prepare('INSERT INTO minhaTabela (nome) VALUES(:nome)');
    $stmt->execute(array(
        ':nome' => 'Ricardo Arrigoni'
    ));
}

```

```
    echo $stmt->rowCount();  
} catch(PDOException $e) {  
    echo 'Error: ' . $e->getMessage();
```

```
?>
```

<https://www.devmedia.com.br/crud-com-php-pdo/28873>

## Referências

<https://www.php.net/manual/en/book.pdo.php>

<https://phpdelusions.net/pdo>

<https://zetcode.com/php/pdo/>

<https://www.phptutorial.net/php-pdo/>