



Introdução

Prof. Anderson Henrique

# Capítulo 6

# Integrando o MySQL com o PHP

Como vimos anteriormente, as variáveis armazenam dados de forma temporária, assim que o programa termina sua execução a informação é destruída da memória RAM. Para resolver este problema, podemos armazenar as informações em um BD através de SGDB confiável. Neste capítulo, iremos trabalhar com funções que nos auxiliam na integração da linguagem com o SGDB MySQL.

Função	Para que serve
<code>mysqli_connect( )</code>	Estabelece conexão com o servidor web.
<code>mysqli_select_db( )</code>	Seleciona um banco de dados específico.
<code>mysqli_error( )</code>	Retorna um erro, caso ocorra.
<code>mysqli_query( )</code>	Executa uma instrução do SQL (DDL, DML, DQL).
<code>mysqli_close( )</code>	Encerra conexão com o servidor web.
<code>mysqli_affected_rows( )</code>	Retorna quantidade de registros afetados.
<code>mysqli_num_rows( )</code>	Retorna quantidade de registros encontrados.
<code>mysqli_fetch_assoc( )</code>	Armazena registros em array associativo.
<code>mysqli_fetch_all( )</code>	Armazena registros em array numérico.
<code>mysqli_fetch_array( )</code>	Armazena registros em arrays (associativo, numérico, misto)
<code>mysqli_fetch_object( )</code>	Armazena registros em um objeto dinâmico.
<code>mysqli_fetch_field( )</code>	Retorna o próximo campo em um resultado.
<code>mysqli_fetch_fields( )</code>	Retorna um array de objetos representa os campos em um resultado.

Acesse o phpmyadmin, e na opção SQL, digite o código abaixo para criar o banco de dados e a tabela:

```
1 CREATE DATABASE php_integrado;
2
3 USE php_integrado;
4
5 CREATE TABLE pessoa(
6
7     id            int(11)      PRIMARY KEY AUTO_INCREMENT NOT NULL,
8     nome          varchar(20)  NOT NULL,
9     sobrenome     varchar(20)  NOT NULL,
10    data_nascimento date        NOT NULL,
11    genero         char(1)      NULL,
12    estado_civil   varchar(10)  NULL
13 )
```

**Obs.:** Aumentar o tamanho do campo **estado\_civil** para 15.

# Arquivo de conexão com o servidor

Para estabelecermos a conexão com o servidor, precisamos saber quais os parâmetros de configuração necessários para que isto aconteça. Vamos utilizar a função `mysqli_connect( )` e a função `mysqli_select_db( )`.

Parâmetro	Valor
HOST (Servidor)	localhost ou 127.0.0.1
USER (Usuário)	root
PASS (Senha)	-----
PORT (Porta)	3306
DBNAME (Nome do Banco de Dados)	php_integrado

Com estas informações, vamos criar um arquivo PHP chamado **conecta.php**, guardar esses parâmetros dentro das constantes, estabelecer a conexão e selecionar o banco de dados.

```
12  define("HOST", "localhost");
13  define("USER", "root");
14  define("PASS", "");
15  define("PORT", "3306");
16  define("DBNAME", "php_integrado");
```

Linha código que estabelece conexão com o servidor, abaixo:

```
18  $conn = mysqli_connect(HOST, USER, PASS, NULL, PORT, NULL)
19                                     or die(mysqli_error());
```

Linha código que seleciona um banco de dados, abaixo:

```
21 mysqli_select_db($conn, DBNAME) or die(mysqli_error());
```

## Comando DML (INSERT INTO) com o PHP

Agora que nossa conexão foi estabelecida e o banco de dados selecionado, teremos que criar um arquivo (**form\_inserir.php**) que terá um formulário para o cadastro, assim utilizaremos o DML **INSERT INTO** na tabela pessoa:



```
1 <h3>Cadastro</h3>
2 <form method="post" action="insert.php">
3     Nome<br>
4     <input type="text" name="nome" required placeholder="Seu nome:"><br><br>
5     Sobrenome<br>
6     <input type="text" name="sobrenome" required placeholder="Seu sobrenome:"><br><br>
7     Data Nascimento<br>
8     <input type="date" name="data" required><br><br>
9     Gênero<br>
10    <select name="genero">
11        <option selected disabled></option>
12        <option value="M">Masculino</option>
13        <option value="F">Feminino</option>
14    </select><br><br>
15    Estado Civil<br>
16    <input type="radio" name="es" value="Solteiro(a)" required>Solteiro(a)
17    <input type="radio" name="es" value="Casado(a)" required>Casado(a)
18    <input type="radio" name="es" value="Divorciado(a)" required>Divorciado(a)
19    <input type="radio" name="es" value="Viúvo(a)" required>Viúvo(a) <br><br>
20    <input type="submit" value="Cadastrar">
21 </form>
```



## Cadastro

Nome

Sobrenome

Data Nascimento

Gênero

Estado Civil

☐ Solteiro(a) ☐ Casado(a) ☐ Divorciado(a) ☐ Viúvo(a)

Perceba que o atributo **action** na TAG form está chamando o arquivo **insert.php**. Este arquivo contém o código que utilizará o DML necessário para efetuar o cadastro na tabela do banco de dados.

Requisição do arquivo de conexão

```
6 include_once '11 - integrando_mysql_e_php.php';
```

Utilizando a função `filter_input()` para pegar os valores digitados e enviados através do método POST, o PHP recebe como atributos deste método (METODO, NOME\_CAMPO), nome do campo está no atributo name da TAG input.

```
15 $nome      = filter_input(INPUT_POST, "nome");
16 $sobrenome = filter_input(INPUT_POST, "sobrenome");
17 $data      = filter_input(INPUT_POST, "data");
18 $genero    = filter_input(INPUT_POST, "genero");
19 $es        = filter_input(INPUT_POST, "es");
```

Armazena o DML INSERT INTO dentro da variável sqlInsert.

```
24 $sqlInsert = "INSERT INTO pessoa(nome,sobrenome,data_nascimento,
25      genero,estado_civil)VALUES('$nome','$sobrenome',
26      '$data','$genero','$es')";
```

A função `mysqli_query` retorna valor booleano, portanto estamos utilizando uma estrutura condicional composta, se retornar `TRUE` o cadastro foi efetuado, senão algum erro ocorreu e não foi possível efetuar o cadastro.

```
31  if(mysqli_query($conn, $sqlInsert)){  
32      echo "Cadastro efetuado com sucesso!";  
33  }else{  
34      echo "Erro ao efetuar o cadastro";  
35  }
```

Encerramos a conexão com o servidor.

```
40  mysqli_close($conn);
```

Vamos efetuar alguns cadastros, pois em seguida criaremos uma página que irá listar todas as pessoas da tabela. Abaixo segue a imagem dos cadastros que foram efetuados na tabela **pessoa** do banco de dados **php\_integrado**.

<div><div>←</div><div>↕</div><div>→</div></div>					id	nome	sobrenome	data_nascimento	genero	estado_civil		
<input type="checkbox"/>		Editar		Copiar		Remover	1	Anderson	Henrique	1978-09-28	M	Solteiro(a
<input type="checkbox"/>		Editar		Copiar		Remover	2	Fabiana	de Souza	1982-03-12	F	Solteiro(a
<input type="checkbox"/>		Editar		Copiar		Remover	3	Diogo	da Silva	1999-01-15	M	Casado(a)
<input type="checkbox"/>		Editar		Copiar		Remover	4	Viviane	Mourão	1974-08-22	F	Casado(a)

## Comando DQL (SELECT FROM) com o PHP

Iremos criar uma página chamada **select.php** onde utilizaremos o DQL SELECT FROM para listar todos os registros da tabela. Segue o código abaixo:



## Tabela que armazenará os resultados

```
1  <h3>Listar</h3>
2
3  <table style="width: 95%; text-align: center">
4      <tr style="background-color: #333; color: #FFF">
5          <td>Nome</td>
6          <td>Sobrenome</td>
7          <td>Data Nascimento</td>
8          <td>Gênero</td>
9          <td>Estado Civil</td>
10     </tr>
11     <?php
```

## Requisição do arquivo de conexão

```
13  include_once '11 - integrando_mysql_e_php.php';
```

**Armazena o DQL SELECT FROM na variável sqlSelect**

```
16      $sqlSelect = "SELECT * FROM pessoa";
```

**Executa a instrução SQL e armazena o resultado, caso ocorra algum erro, o código finaliza a execução nessa linha**

```
19      $result = mysqli_query($conn, $sqlSelect) or die(mysqli_error());
```

**Retorna a quantidade de registros (linhas) afetadas pela instrução SQL**

```
22      echo "Total de Registros Afetados: " . mysqli_affected_rows($conn)
23      . "<br>";
```

**Retorna o total de registros (linhas) encontradas**

```
26      echo "Total de registros: " . mysqli_num_rows($result) . "<br>";
```



## Retorna registro (linha) em um array associativo

```
35 | | print_r(mysqli_fetch_assoc($result));
```

```
Array
(
    [id] => 1
    [nome] => Anderson
    [sobrenome] => Henrique
    [data_nascimento] => 1978-09-28
    [genero] => M
    [estado_civil] => Solteiro(a
)
```

## Retorna todos os registros (linhas) em um array numérico

```
38 | | print_r(mysqli_fetch_all($result));
```

```
Array
(  
  [0] => Array  
    (  
      [0] => 1  
      [1] => Anderson  
      [2] => Henrique  
      [3] => 1978-09-28  
      [4] => M  
      [5] => Solteiro(a  
    )  
  
  [1] => Array  
    (  
      [0] => 2  
      [1] => Fabiana  
      [2] => de Souza  
      [3] => 1982-03-12  
      [4] => F  
      [5] => Solteiro(a  
    )  
)
```

**Retorna registro (linha) em arrays (associativo, numérico, misto)**

```
41 | print_r(mysqli_fetch_array($result, MYSQLI_ASSOC));
```

**Retorna registro (linha) em objeto dinâmico**

```
44 | print_r(mysqli_fetch_object($result));
```

```
stdClass Object
(
    [id] => 1
    [nome] => Anderson
    [sobrenome] => Henrique
    [data_nascimento] => 1978-09-28
    [genero] => M
    [estado_civil] => Solteiro(a
)
```

Optamos retornar os registros em um array associativo, para isto, utilizamos um laço de repetição para buscar todos os registros e armazenar em linhas da nossa tabela. Cada volta que o laço dá, cria uma nova linha na tabela e exibe os valores das colunas da tabela em uma TAG <td>.

Para exibir o resultado, utilizamos a sintaxe do php conhecida como modo resultado <?= ?>

```
21 while($linha = mysqli_fetch_assoc($result)) {
22     ?>
23     <tr>
24         <td><?=$linha["nome"]?></td>
25         <td><?=$linha["sobrenome"]?></td>
26         <td><?=$linha["data_nascimento"]?></td>
27         <td><?=$linha["genero"]?></td>
28         <td><?=$linha["estado_civil"]?></td>
29     </tr>
30     <?php } ?>
31 </table>
```

## Listar

Nome	Sobrenome	Data Nascimento	Gênero	Estado Civil
Anderson	Henrique	1978-09-28	M	Solteiro(a
Fabiana	de Souza	1982-03-12	F	Solteiro(a
Diogo	da Silva	1999-01-15	M	Casado(a)
Viviane	Mourão	1974-08-22	F	Casado(a)

## Comando DML (DELETE) com o PHP

Vamos alterar o arquivo **select.php** adicionando uma coluna de título na nossa tabela e coluna no resultado contendo um link para excluir cada registro.

```
32 <td><?=$linha["estado_civil"]?></td>
33 <td>
34 <a href="delete.php?id=<?=$linha["id"]?>">excluir</a>
35 </td>
36 </tr>
```

Observe que o link chama o arquivo **delete.php** sendo que inserimos o sinal de (?), o que indica a passagem de um parâmetro na URL, esse parâmetro armazenará o valor do id (chave\_primária) de cada registro (linha) na nossa tabela.

Gênero	Estado Civil	#
M	Solteiro(a)	<a href="#">excluir</a>
F	Solteiro(a)	<a href="#">excluir</a>
M	Casado(a)	<a href="#">excluir</a>
F	Casado(a)	<a href="#">excluir</a>

Se o ponteiro do mouse estiver sobre o link **excluir**, a barra de status do navegador apresentará na URL o nome do parâmetro seguido do valor que este recebe dinamicamente do array. Esta passagem de parâmetro se dá pela URL, portanto, o método que utilizamos para recuperar esse valor é o **GET**.

localhost/curso\_php/delete.php?id=4

Abaixo temos o código do arquivo **delete.php**, utilizando a conexão com o servidor, buscando o valor do parâmetro id, executando a instrução sqlDelete e retornando o resultado:

### Requisição do arquivo de conexão

```
4 include_once '11 - integrando_mysql_e_php.php';
```

Pega o valor do parâmetro 'id' da URL e armazena

```
7 $id = filter_input(INPUT_GET, "id");
```

Armazena o DML DELETE FROM na variável sqlDelete

```
10 $sqlDelete = "DELETE FROM pessoa WHERE id='$id' ";
```

A função **mysqli\_query** retorna valor booleano, portanto estamos utilizando uma estrutura condicional composta, se retornar TRUE o registro foi excluído, senão algum erro ocorreu e não foi possível excluir o registro.

```
13  if(mysqli_query($conn, $sqlDelete)){
14      echo "Registro excluído com sucesso!";
15  }else{
16      echo "Erro ao excluir o registro";
17  }
```

Excluímos o registro do Diogo da Silva

Nome	Sobrenome	Data Nascimento	Gênero
Anderson	Henrique	1978-09-28	M
Fabiana	de Souza	1982-03-12	F
Viviane	Mourão	1974-08-22	F