

Integrando PHP com banco de dados

- ▶ Um sistema dinâmico possui entre suas principais responsabilidades a integração com algum banco de dados, para armazenar, por exemplo:
 - ▶ Usuários e senhas, dados de produtos, informações cadastrais de clientes, funcionários, fornecedores, vendas, etc...
- ▶ A diferenciação dos seguintes termos é essencial e se faz necessária:

- ▶ **Banco de Dados**

"É uma coleção de dados logicamente coerente que possui um significado implícito cuja interpretação é dada por uma determinada aplicação"

No modelo relacional, os bancos de dados são compostos por tabelas e estas, por sua vez, armazenam os dados

- ▶ **SGBD (Sistema de Gerenciamento (ou Gerenciador) de Banco de Dados)**

"Software construído para facilitar as atividades de definição, construção e manipulação de bancos de dados"

Exemplos: MySQL, PostgreSQL, Oracle, SQL-Server, etc...

Integrando PHP com banco de dados

- ▶ Estudaremos a integração do PHP com bancos de dados do tipo MySQL
- ▶ O MySQL já é parte integrante do pacote XAMPP, mas também pode ser obtido gratuitamente em seu site:
 - ▶ <https://www.mysql.com/>
- ▶ O XAMPP também inclui o phpMyAdmin, que é um *front-end* para o MySQL. Ele pode ser obtido gratuitamente em:
 - ▶ <http://www.phpmyadmin.net/>
 - ▶ Uma vez instalado e configurado, o phpMyAdmin pode ser acessado pelo endereço:
 - ▶ <http://localhost/phpmyadmin>
- ▶ Existem ainda outros *front-ends* para o MySQL, por exemplo:
 - ▶ [MySQL-Front](#)
 - ▶ [HeidiSQL](#)
- ▶ Outro *software* de destaque para administração de bancos de dados MySQL é o [MySQL Workbench](#), que é muito mais que um simples *front-end*



Integrando PHP com banco de dados

- ▶ A manipulação de dados armazenados em um banco de dados utilizando PHP exige algumas etapas:
 1. Efetuar a conexão com o servidor e selecionar o banco de dados
 2. Executar o comando SQL para
 - ▶ Consultar os registros
 - ▶ Inserir novos registros
 - ▶ Alterar um registro existente
 - ▶ Excluir um ou mais registros
 3. Obter os resultados
 4. Visualizar os resultados
 5. Encerrar a Conexão

Integrando PHP com banco de dados

1. Efetuar a conexão com o servidor e selecionar o banco de dados

- ▶ Primeiramente, é necessário criar uma variável que conterà a conexão com o SGBD MySQL
- ▶ Utiliza-se a função `mysqli_connect`
 - ▶ Objetivo: abre uma conexão com um servidor MySQL
 - ▶ Retorno: um identificador de conexão MySQL em caso de sucesso na conexão, ou `FALSE` em caso de falha.

Sintaxe

```
$nome_variavel = mysqli_connect("end_servidor", "usuario", "senha",  
"banco_de_dados");
```

Exemplo

```
$conexao = mysqli_connect("localhost", "root", "admin",  
"meu_banco");
```

Integrando PHP com banco de dados

2. Executar o comando SQL

- ▶ Uma vez conectado ao banco de dados e definido sobre qual banco as instruções serão realizadas, é necessário montar uma instrução (*query*) SQL
- ▶ As instruções SQL são *strings* que podem ser guardadas em variáveis, para melhor organização do código

Sintaxe

```
$nome_variavel = "instrucao sql";
```

Exemplo

```
$sql = "SELECT * FROM clientes ORDER BY nome";
```

Integrando PHP com banco de dados

3. Obter os resultados

- ▶ Para obter os resultados de uma instrução SQL, devemos aplicá-la ao banco de dados e armazenar a resposta em uma variável
- ▶ Utiliza-se a função mysqli_query
 - ▶ Objetivo: envia uma consulta MySQL
 - ▶ Retorno:

Variável especial, que mantém uma referência a um recurso externo.

- ▶ Para comandos SELECT, SHOW, DESCRIBE ou EXPLAIN, retorna um conjunto de resultados do tipo mysqli_result em caso de sucesso, ou FALSE em caso de falha;
- ▶ Para outros tipos de *query* SQL, como UPDATE, DELETE, DROP, etc., retorna TRUE em caso de sucesso ou FALSE em caso de erro.

Sintaxe

```
$nome_variavel = mysqli_query($variavel_conexao, $variavel_query_sql);
```

Exemplo

```
$rs = mysqli_query($conexao, $sql);
```

Integrando PHP com banco de dados

Para *queries* SQL como UPDATE, DELETE, DROP, etc., este passo é desnecessário.

4. Visualizar os resultados

- ▶ Para exibir o resultado de uma consulta SQL (*SELECT*), é necessário separar os registros por linha
- ▶ Utiliza-se a função `mysqli_fetch_assoc`
 - ▶ Objetivo: obtém um linha do resultado como uma matriz associativa
 - ▶ Retorno: uma matriz associativa de *strings* que corresponde à linha obtida, ou `NULL` se não houverem mais linhas.

Sintaxe

```
$nome_variavel = mysqli_fetch_assoc($variavel_resultado_query_sql);
```

Exemplo

```
while($reg = mysqli_fetch_assoc($rs)){  
    echo $reg['nome'] . '<br>' . $reg['cpf'];  
}
```

Integrando PHP com banco de dados

5. Encerrar a conexão

- ▶ Ao terminar as operações em uma tabela, convém fechá-la, assim como a conexão utilizada
- ▶ Utilizam-se as funções:
 - ▶ mysqli_free_result
 - ▶ Objetivo: Libera um resultado da memória
 - ▶ Retorno: *não há valor retornado*
 - ▶ mysqli_close
 - ▶ Objetivo: Fecha a conexão MySQL
 - ▶ Retorno: TRUE em caso de sucesso ou FALSE em caso de falha

Sintaxe

```
mysqli_free_result($variavel_resultado_query_sql);  
mysqli_close($variavel_conexao);
```

Exemplo

```
mysqli_free_result($rs);  
mysqli_close($conexao);
```


Integrando PHP com banco de dados

Resumo das Operações

1	Efetuar a conexão com o servidor MySQL e selecionar o banco de dados <code>\$nome_variavel = mysqli_connect("end_servidor", "usuario", "senha", "banco_de_dados");</code> Exemplo: <code>\$conexao = mysqli_connect("localhost", "root", "admin", "meu_banco");</code>
2	Criar a instrução SQL <code>\$nome_variavel = "instrucao sql";</code> Exemplo: <code>\$sql = "SELECT * FROM clientes ORDER BY nome";</code>
3	Guardar o resultado obtido em uma variável <code>\$nome_variavel = mysqli_query(\$variavel_conexao, \$variavel_query_sql);</code> Exemplo: <code>\$rs = mysqli_query(\$conexao, \$sql);</code>
4	Preparar os dados para serem exibidos <code>\$nome_variavel = mysqli_fetch_assoc(\$variavel_resultado_query_sql);</code> Exemplo: <pre>while(\$reg = mysqli_fetch_assoc(\$rs)){ echo \$reg['nome'] . '
' . \$reg['cpf']; }</pre>
5	Encerrar a conexão com o banco <code>mysqli_free_result(\$variavel_resultado_query_sql);</code> <code>mysqli_close(\$variavel_conexao);</code> Exemplo: <code>mysqli_free_result(\$rs);</code> <code>mysqli_close(\$conexao);</code>

IMPORTANTE:

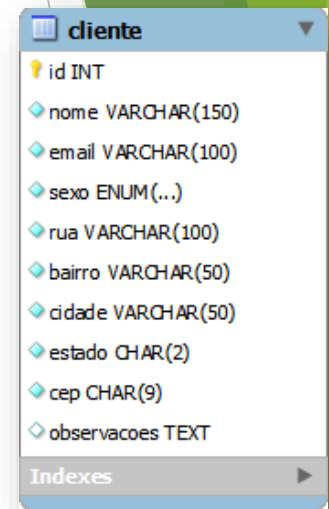
Insira pelo menos 10 registros na tabela, utilizando o formulário. Isso servirá para testar seu código e também para popular a tabela.

Exercício 34

- 34) Obtenha os arquivos `meu_formulario.php` e `exibe_dados.php`, utilizados no exercício 32. Modifique o arquivo `exibe_dados.php` para, **após validar e mostrar os dados submetidos pelo formulário** (isso já deve estar pronto), gravá-los no banco de dados, exibindo uma mensagem de sucesso ou falha no processo de gravação.

O SQL para criação do banco de dados `meubanco` e da tabela `cliente` é o seguinte:

```
CREATE DATABASE IF NOT EXISTS `meubanco`;
CREATE TABLE IF NOT EXISTS `meubanco`.`cliente` (
  `id` INT NOT NULL AUTO INCREMENT,
  `nome` VARCHAR(150) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `sexo` ENUM('masculino','feminino') NOT NULL,
  `rua` VARCHAR(100) NOT NULL,
  `bairro` VARCHAR(50) NOT NULL,
  `cidade` VARCHAR(50) NOT NULL,
  `estado` CHAR(2) NOT NULL,
  `cep` CHAR(9) NOT NULL,
  `observacoes` TEXT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB
```



O SQL para inserção de um registro na tabela `cliente` é o seguinte:

```
INSERT INTO `meubanco`.`cliente`
  (`nome`, `email`, `sexo`, `rua`, `bairro`, `cidade`, `estado`,
  `cep`, `observacoes`)
VALUES
  ('Marcos', 'marcos.vieira@ifgoiano.edu.br', 'masculino', 'Av.
  Pará', 'Centro', 'Iporá', 'GO', '76200-000', 'Sem observações no
  momento');
```

Exercício 35 e 36

- 35) Crie o arquivo `lista_clientes.php`, e mostre os dados de todos os clientes gravados na tabela `clientes`.

O SQL para esta consulta é o seguinte:

```
SELECT * FROM `meubanco`.`cliente` ORDER BY `nome`;
```

Exemplo:

Id: 1
Nome: Marcos
E-mail: marcos.vieira@ifgoiano.edu.br
Sexo: masculino
Rua: Av. Pará
Bairro: Centro
Cidade: Iporá
Estado: GO
CEP: 76200-000
Observações: Sem observações

Id: 2 (...)

DICA: Reproveite o código utilizado para exibir os dados do formulário, contido no arquivo `exibe_dados.php`, do exercício 34.

- 36) Modifique o arquivo `lista_clientes.php` para mostrar os dados de todos os clientes gravados na tabela `clientes` em forma de uma tabela, onde cada linha representa os dados de um registro da tabela.

ID	Nome	E-mail	Sexo	Rua	Bairro	Cidade	Estado	CEP	Observações
1	Marcos	marcos.vieira@ifgoiano.edu.br	masculino	Av. Pará	Centro	Iporá	GO	76200-000	Sem observações
2	(...)								

DICA: Faça o código HTML da tabela estática. Depois, coloque a linha do registro (`<tr>` até `</tr>`) dentro do `while` do `mysql_fetch_assoc`, fazendo então as alterações necessárias para exibição dos dados.