

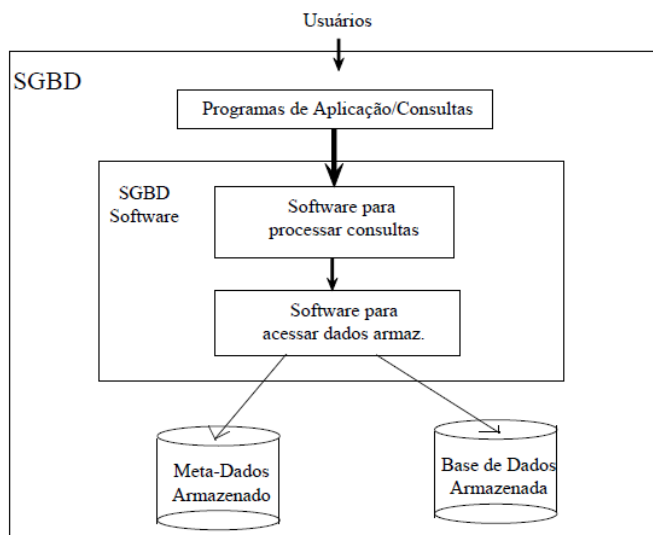
# PHP - Aula 9

## Tópicos

- 1 Alguns conceitos iniciais
- 2 Conceitos básicos sobre banco de dados relacional
- 3 Como criar um banco de dados para teste?
  - 3.1 Como criar tabelas em um banco de dados
  - 3.2 Inserindo um registro
- 4 Executando comando SQL em um Programa PHP
  - 4.1 Abrir uma conexão com o MySQL
  - 4.2 Criar um banco de dados MySQL
  - 4.3 Criar tabelas MySQL
  - 4.4 Inserindo dados em um banco de dados
  - 4.5 Inserir vários registros no MySQL Usando MySQLi
  - 4.6 Selecionar dados de um banco de dados MySQL
  - 4.7 Excluir dados do MySQL
  - 4.8 Atualizar dados em MySQL

### 1 Alguns conceitos iniciais

- **Dados:** são fatos que podem ser armazenados.
  - Exemplo: nomes, telefones, endereços, etc.
- **Base de dados:** é uma coleção de dados interrelacionados logicamente.
  - Exemplo: agenda de telefones.
- **Sistema de Gerência de Bases de Dados (SGBD):** coleção de programas que permite a criação e gerência de bases de dados ou Sistema de Banco de Dados.
  - Exemplos: PostgreSQL, Firebird, MySQL, Oracle, etc.



### 2 Conceitos básicos sobre banco de dados relacional

- **Tabelas:** um banco de dados relacional é formado por tabelas.

Exemplo:

| NumeroCliente | NomeCliente      | Endereço                            | CEP      |
|---------------|------------------|-------------------------------------|----------|
| 1             | Antônio José     | Rua 15 de Novembro, Nr 125, Centro  | 89022000 |
| 2             | Maria Ana        | Rua Gen Sampaio, Nr 1185, Várzea    | 56870320 |
| 3             | Erica Fernandes  | Rua 23, Nr 15, Cidade Nova          | 87032000 |
| 4             | Patrícia Ribeiro | Rua do Cemitério, S/N, Buraco Fundo | 15687320 |

- **Campo:** cada coluna da tabela armazena um tipo de dado e representa um campo do banco de dados.
- **Registro:** cada linha armazena os dados de um cliente e representa um registro.
- **Chave primária:** a tabela deve ter um campo que identifica o registro. Este campo é chamado de chave ou chave primária. Os dados da chave primária devem ser únicos e não devem haver dois ou mais registros que possuam os mesmos dados armazenados no campo que for a chave primária. Na nossa tabela de exemplo acima, a chave primária é o campo NumeroCliente.
- **Chave estrangeira:** uma tabela também pode fazer referência a chave de uma outra tabela, quando isto acontece a chave da outra tabela é chamada chave estrangeira. Nesta tabela, o campo NumeroPedido é a chave primária e o campo NumeroCliente é uma chave estrangeira. Neste caso, a chave estrangeira serve para referenciar qual o cliente que fez o pedido, por isso ela contém a chave primária da tabela dos clientes.

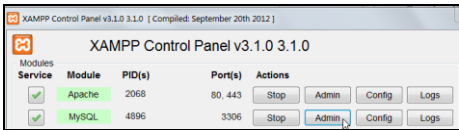
Exemplo:


| NumeroPedido | NumeroCliente | Valor  | Data   |
|--------------|---------------|--------|--------|
| 1            | 5             | 200,00 | 050603 |
| 2            | 3             | 50,00  | 050603 |
| 3            | 1             | 175,00 | 060603 |
| 4            | 2             | 300,00 | 060603 |


- **Relacionamentos:** as chaves estrangeiras representam um relacionamento entre as tabelas. Existem três tipos de relacionamentos: de um para um; de um para muitos; e de muitos para muitos.

### 3 Como criar um banco de dados para teste?

Veremos agora como criar um banco de dados no MySQL, e na sequência criaremos as tabelas e manipularemos dados dentro desse banco de dados. Utilizaremos o MySQL por ser aquele usualmente utilizado nas instalações do XAMPP, WAMP ou MAMP.

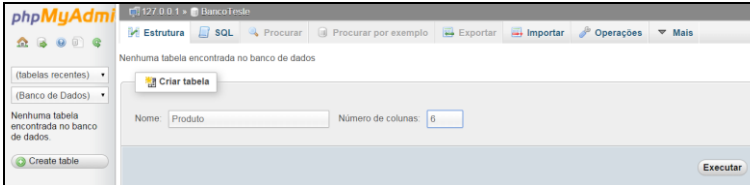
- 

The screenshot shows the XAMPP Control Panel v3.1.0.3.1.0. The MySQL service is running, indicated by a green checkmark and the 'Admin' button being highlighted.
- 

The screenshot shows the phpMyAdmin interface. The 'General Settings' tab is selected, and the 'Server connection collation' is set to 'utf8\_general\_ci'.
- 

The screenshot shows the 'Criar banco de dados' (Create database) screen in phpMyAdmin. The database name 'BancoTeste' is entered, and the 'Criar' (Create) button is highlighted.

#### 3.1 Como criar tabelas em um banco de dados

- 

The screenshot shows the 'Criar tabela' (Create table) screen in phpMyAdmin. The table name 'Produto' is entered, and the number of columns is set to 6. The 'Executar' (Execute) button is highlighted.

2.

Nome da Tabela:  Add  column(s)

| Nome                 | Tipo | Tamanho/Definir*     | Padrão | Colação | Atributos |
|----------------------|------|----------------------|--------|---------|-----------|
| <input type="text"/> | INT  | <input type="text"/> | None   |         |           |
| <input type="text"/> | INT  | <input type="text"/> | None   |         |           |
| <input type="text"/> | INT  | <input type="text"/> | None   |         |           |
| <input type="text"/> | INT  | <input type="text"/> | None   |         |           |
| <input type="text"/> | INT  | <input type="text"/> | None   |         |           |
| <input type="text"/> | INT  | <input type="text"/> | None   |         |           |

Código (int)  
 Nome (varchar 50)  
 Descrição (varchar 200)  
 Preço (double)  
 Categoria (varchar 50)  
 Subcategoria (varchar 50)

3.

Nome da Tabela:  Add  column(s)

| Nome         | Tipo    | Tamanho/Definir* | Padrão | Colação | Atributos |
|--------------|---------|------------------|--------|---------|-----------|
| Código       | INT     |                  | None   |         |           |
| Nome         | VARCHAR | 50               | None   |         |           |
| Descrição    | VARCHAR | 200              | None   |         |           |
| Preço        | DOUBLE  |                  | None   |         |           |
| Categoria    | VARCHAR | 50               | None   |         |           |
| Subcategoria | VARCHAR | 50               | None   |         |           |

4. Clicar em salvar

### 3.2 Inserindo um registro

1.

phpMyAdmin 127.0.0.1 - Banco teste

Tabela: **produto**

2.

phpMyAdmin 127.0.0.1 - Banco teste - produto

| Coluna       | Tipo         | Função | Nulo | Valor                |
|--------------|--------------|--------|------|----------------------|
| Código       | int(11)      |        |      | <input type="text"/> |
| Nome         | varchar(50)  |        |      | <input type="text"/> |
| Descrição    | varchar(200) |        |      | <input type="text"/> |
| Preço        | double       |        |      | <input type="text"/> |
| Categoria    | varchar(50)  |        |      | <input type="text"/> |
| Subcategoria | varchar(50)  |        |      | <input type="text"/> |

3.

phpMyAdmin 127.0.0.1 - Banco teste - produto

Mostrando registros 0 - 1 (-2 total) - Consulta levou 0.0003 segundos

```
SELECT * FROM produto LIMIT 0, 30
```

Mostrar: Linha inicial: 0 Number of rows: 30 Cabeçalho a cada 100 linhas

|  | Código | Nome       | Descrição                      | Preço | Categoria  | Subcategoria |
|--|--------|------------|--------------------------------|-------|------------|--------------|
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover | 1010   | Mouse      | apontador digital              | 50    | periférico | entrada      |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Remover | 2020   | impressora | impressora jato de tinta epson | 450   | periférico | saída        |

---

## 4 Executando comando SQL em um Programa PHP

PHP MySQLi = MySQL PHP Melhor!

As funções mysqli permitem a você acessar os servidores de banco de dados MySQL.

### Observação:

A extensão MySQL é projetado para trabalhar com o MySQL versão 4.1.13 ou mais recente.

---

### 4.1 Abrir uma conexão com o MySQL

Antes que possamos acessar os dados no banco de dados MySQL, precisamos ser capazes de se conectar ao servidor.

#### Exemplo:

```
<?php
$nome_servidor = "localhost";
$nome_usuario = "root";
$senha = "";
// Criar conexão
$conecta = new mysqli($nome_servidor, $nome_usuario, $senha);
// Verificar Conexão
if ($conecta->connect_error) {
    die("Conexão falhou: " . $conecta->connect_error."<br>");
}
echo "Conexão realizada com sucesso";
$conecta->close();
?>
```

---

### 4.2 Criar um banco de dados MySQL

A instrução CREATE DATABASE é utilizado para criar um banco de dados em MySQL. O exemplo a seguir cria um banco de dados chamado "BancoTeste".

```
<?php
$nome_servidor = "localhost";
$nome_usuario = "root";
$senha = "";
// Criar conexão
$conecta = new mysqli($nome_servidor, $nome_usuario, $senha);
// Verificar Conexão
if ($conecta->connect_error) {
    die("Conexão falhou: " . $conecta->connect_error."<br>");
}
echo "Conexão realizada com sucesso <br>";
// Cria banco de dados
$sql = "CREATE DATABASE BancoTeste";
if ($conecta->query($sql) === TRUE) {
    echo "Banco de dados criado com sucesso<br>";
} else {
    echo "Erro na criação do banco de dados: " . $conecta->error."<br>";
}
$conecta->close();
?>
```

### 4.3 Criar tabelas MySQL

A instrução CREATE TABLE é usado para criar uma tabela no MySQL. Vamos criar uma tabela chamada "participantes", com quatro colunas: "id", "nome", "sobrenome" e "e-mail".

```
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(30) NOT NULL,  
sobrenome VARCHAR(30) NOT NULL,  
email VARCHAR(50)
```

**Exemplo:**

```
<?php  
$nome_servidor = "localhost";  
$nome_usuario = "root";  
$senha = "";  
$nome_banco = "BancoTeste";  
// Criar conexão  
$conecta = new mysqli($nome_servidor, $nome_usuario, $senha, $nome_banco);  
// Verificar Conexão  
if ($conecta->connect_error) {  
    die("Conexão falhou: " . $conecta->connect_error."<br>");  
}  
echo "Conexão realizada com sucesso <br>";  
// Criar tabela  
$sql = "CREATE TABLE participantes(  
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(30) NOT NULL,  
    sobrenome VARCHAR(30) NOT NULL,  
    email VARCHAR(50)  
)";  
if ($conecta->query($sql) === TRUE) {  
    echo "Tabela participantes criada com sucesso<br>";  
} else {  
    echo "Erro na criação da tabela participantes: " . $conecta->error."<br>";  
}  
$conecta->close();  
?>
```

### 4.4 Inserindo dados em um banco de dados

Depois de um banco de dados e uma tabela terem sido criados, podemos começar a adicionar os dados em si. Cada linha de uma tabela de um BD é chamada de registro. Para inserir um ou mais registros usamos o comando SQL INSERT. Aqui estão algumas regras de sintaxe a seguir:

- A consulta SQL deve estar entre aspas duplas no PHP;
- Strings devem estar entre aspas simples dentro do comando SQL;
- Os valores numéricos não devem estar entre aspas;
- A palavra NULL não deve estar entre aspas.

A instrução INSERT INTO é usado para adicionar novos registros a uma tabela MySQL:

**Formato padrão:**

```
INSERT INTO <nome da tabela> VALUES (valor1, valor2, valor3...);
```

Ou

```
INSERT INTO <nome da tabela> (campo1, campo2, campo3,...) VALUES (valor1, valor2, valor3...);
```

**Exemplo:**

```
<?php
$nome_servidor = "localhost";
$nome_usuario = "root";
$senha = "";
$nome_banco = "BancoTeste";
// Criar conexão
$conecta = new mysqli($nome_servidor, $nome_usuario, $senha,$nome_banco);
// Verificar Conexão
if ($conecta->connect_error) {
    die("Conexão falhou: " . $conecta->connect_error."<br>");
}
echo "Conexão realizada com sucesso <br>";
// Inserir registro
$sql = "INSERT INTO participantes (nome, sobrenome, email)
VALUES ('Salvador', 'Melo', 'salvador.junior@projecao.br')";
if ($conecta->query($sql) === TRUE) {
    echo "Novo registro criado com sucesso";
} else {
    echo "Erro: " . $sql . "<br>" . $conecta->error."<br>";
}
$conecta->close();
?>
```

---

## 4.5 Inserir vários registros no MySQL Usando MySQLi

Várias instruções SQL podem ser executadas com a função `mysqli_multi_query()`. O exemplo a seguir acrescentar três novos registros para a tabela "produto":

**Exemplo:**

```
<?php
$nome_servidor = "localhost";
$nome_usuario = "root";
$senha = "";
$nome_banco = "BancoTeste";
// Criar conexão
$conecta = new mysqli($nome_servidor, $nome_usuario, $senha,$nome_banco);
// Verificar Conexão
if ($conecta->connect_error) {
    die("Conexão falhou: " . $conecta->connect_error."<br>");
}
echo "Conexão realizada com sucesso <br>";
// Inserir vários registros
$sql = "INSERT INTO participantes (nome, sobrenome, email)
VALUES ('Maria', 'Dias', 'mary@example.com')";
$sql .= "INSERT INTO participantes (nome, sobrenome, email)
VALUES ('Julie', 'Farias', 'julie@example.com')";
if ($conecta->multi_query($sql) === TRUE) {
    echo "Novos registros criados com sucesso<br>";
} else {
    echo "Erro: " . $sql . "<br>" . $conecta->error."<br>";
}
$conecta->close();
?>
```

---

## 4.6 Selecionar dados de um banco de dados MySQL

A instrução SELECT é usada para selecionar dados de uma ou mais tabelas.

### Formato padrão:

SELECT nome\_coluna FROM nome\_tabela

ou podemos usar o caractere \* para selecionar todos os registros de uma tabela:

SELECT \* FROM nome\_tabela

O exemplo a seguir seleciona o id, nome e sobrenome colunas da tabela participantes e apresenta na página web.

### Exemplo:

```
<?php
$nome_servidor = "localhost";
$nome_usuario = "root";
$senha = "";
$nome_banco = "BancoTeste";
// Criar conexão
$conecta = new mysqli($nome_servidor, $nome_usuario, $senha,$nome_banco);
// Verificar Conexão
if ($conecta->connect_error) {
    die("Conexão falhou: " . $conecta->connect_error."<br>");
}
echo "Conexão realizada com sucesso <br>";
$sql = "SELECT id, nome, sobrenome FROM participantes";

$resultado = $conecta->query($sql);
if ($resultado->num_rows > 0) {
    // saída dos dados
    while($linha = $resultado->fetch_assoc()) {
        echo "id: " . $linha["id"]. " - Name: " . $linha["nome"]. " " . $linha["sobrenome"]. "<br>";
    }
} else {
    echo "0 results";
}
$conecta->close();
?>
```

### Observação:

fetch\_assoc() obtém uma linha do conjunto de resultados como uma matriz associativa.

---

## 4.7 Excluir dados do MySQL

A instrução DELETE é usado para excluir registros de uma tabela.

### Formato padrão:

DELETE FROM nome\_tabela

WHERE alguma\_coluna = algum\_valor

**Exemplo:**

```
<?php
    $nome_servidor = "localhost";
    $nome_usuario = "root";
    $senha = "";
    $nome_banco = "BancoTeste";
    // Criar conexão
    $conecta = new mysqli($nome_servidor, $nome_usuario, $senha,$nome_banco);
    // Verificar Conexão
    if ($conecta->connect_error) {
        die("Conexão falhou: " . $conecta->connect_error."<br>");
    }
    echo "Conexão realizada com sucesso <br>";

    // apagar registro
    $sql = "DELETE FROM participantes WHERE id=3";
    if ($conecta->query($sql) === TRUE) {
        echo "Registro apagado com sucesso<br>";
    } else {
        echo "Erro ao apagar o registro: " . $conecta->error."<br>";
    }
    $conecta->close();
?>
```

---

## 4.8 Atualizar dados em MySQL

A instrução UPDATE é usada para atualizar os registros existentes em uma tabela.

**Formato padrão:**

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

**Exemplo:**

```
<?php
    $nome_servidor = "localhost";
    $nome_usuario = "root";
    $senha = "";
    $nome_banco = "BancoTeste";
    // Criar conexão
    $conecta = new mysqli($nome_servidor, $nome_usuario, $senha,$nome_banco);
    // Verificar Conexão
    if ($conecta->connect_error) {
        die("Conexão falhou: " . $conecta->connect_error."<br>");
    }
    echo "Conexão realizada com sucesso <br>";
    $sql = "UPDATE participantes SET sobrenome='Galvao' WHERE id=2";
    if ($conecta->query($sql) === TRUE) {
        echo "Registro atualizado com sucesso<br>";
    } else {
        echo "Erro na atualização do registro: " . $conecta->error."<br>";
    }
    $conecta->close();
?>
```