

# Apostila de Banco de Dados

Professor: Marcos Brandão

Disciplina: Desenvolvimento Web/ Banco de Dados

Versão: 2026

---

## Sumário

1. Introdução aos Bancos de Dados
  2. Conceitos Fundamentais
  3. Instalando e Acessando o MariaDB
  4. Criando Bancos de Dados
  5. Tipos de Dados
  6. Criando Tabelas
  7. Inserindo Dados
  8. Consultas Básicas com SELECT
  9. Filtros e Ordenação
  10. Atualizando e Excluindo Dados
  11. Relacionamento entre Tabelas
  12. JOINS na Prática
  13. Funções e Operações
  14. Backup e Restauração
  15. Projeto Prático Guiado
  16. Exercícios Propostos
  17. Desafios Extras
- 

## 1. Introdução aos Bancos de Dados

### O que é um Banco de Dados?

Um banco de dados é um sistema utilizado para armazenar, organizar e recuperar informações.

Exemplos do dia a dia:

- Cadastro de alunos
- Sistema de vendas
- Redes sociais
- Aplicativos de mensagens
- Sistemas bancários
- Sites de e-commerce

Sem banco de dados, as informações seriam difíceis de organizar e consultar.

---

## O que é o MariaDB?

O MariaDB é um Sistema Gerenciador de Banco de Dados (SGBD).

Ele é:

- Gratuito
- Open source
- Compatível com MySQL
- Muito utilizado em aplicações web

Principais usos:

- Sites PHP
  - APIs
  - Sistemas administrativos
  - Aplicações web em geral
- 

## Estrutura Básica

### Banco de Dados

É o local onde as tabelas ficam armazenadas.

### Tabela

Organiza os dados em linhas e colunas.

### Registro

Cada linha da tabela.

### Campo

Cada coluna da tabela.

---

## 2. Conceitos Fundamentais

### Exemplo de Tabela

Tabela: usuarios

id	nom e	email
1	João	joao@email.com
2	Maria	maria@email.com

---

### Chave Primária (PRIMARY KEY)

É um campo que identifica unicamente cada registro.

Exemplo:

```
id INT PRIMARY KEY
```

---

### Auto Incremento

Faz o número aumentar automaticamente.

```
id INT AUTO_INCREMENT PRIMARY KEY
```

---

### Chave Estrangeira (FOREIGN KEY)

Usada para relacionar tabelas.

Exemplo:

- Uma tabela de pedidos ligada à tabela de clientes.
- 

## 3. Instalando e Acessando o MariaDB

# Instalação no Ubuntu Server

## Atualizar o sistema

```
sudo apt update
```

---

## Instalar o MariaDB

```
sudo apt install mariadb-server -y
```

---

## Verificar status

```
sudo systemctl status mariadb
```

Resultado esperado:

```
active (running)
```

---

## Acessar o MariaDB

```
sudo mariadb
```

Ou:

```
mysql -u root -p
```

---

# Comandos Básicos do Terminal MariaDB

## Mostrar bancos de dados

```
SHOW DATABASES;
```

---

## Selecionar um banco

```
USE nome_banco;
```

---

## Mostrar tabelas

```
SHOW TABLES;
```

---

## Sair do MariaDB

```
EXIT;
```

---

## 4. Criando Bancos de Dados

### Criando um Banco

```
CREATE DATABASE escola;
```

---

### Verificando

```
SHOW DATABASES;
```

---

### Selecionando o Banco

```
USE escola;
```

---

### Excluindo Banco

CUIDADO: apaga tudo.

```
DROP DATABASE escola;
```

---

## 5. Tipos de Dados

Tipo	Descrição
INT	Número inteiro
VARCHAR(100)	Texto curto
TEXT	Texto grande
DATE	Data
DATETIME	Data e hora
DECIMAL(10,2)	Valores monetários
BOOLEAN	Verdadeiro/Falso

---

### Exemplo

nome VARCHAR(100)  
idade INT  
salario DECIMAL(10,2)

---

## 6. Criando Tabelas

### Criando Tabela de Alunos

```
CREATE TABLE alunos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(100),  
idade INT,  
curso VARCHAR(100)  
);
```

---

### Verificar Estrutura

```
DESCRIBE alunos;
```

Resultado esperado:

Field	Type
id	int
nome	varchar(100)
idade	int
curso	varchar(100)

---

## 7. Inserindo Dados

### Inserir Um Registro

```
INSERT INTO alunos (nome, idade, curso)  
VALUES ('Carlos', 20, 'Desenvolvimento Web');
```

---

### Inserir Vários Registros

```
INSERT INTO alunos (nome, idade, curso)
VALUES
('Ana', 19, 'Redes'),
('Pedro', 22, 'Banco de Dados'),
('Marina', 18, 'Desenvolvimento Web');
```

---

## Visualizar Dados

```
SELECT * FROM alunos;
```

Resultado esperado:

id	nome	idade	curso
1	Carlos	20	Desenvolvimento Web
2	Ana	19	Redes

---

# 8. Consultas Básicas com SELECT

## Selecionar Tudo

```
SELECT * FROM alunos;
```

---

## Selecionar Colunas Específicas

```
SELECT nome, curso FROM alunos;
```

---

## Consultar Um Registro

```
SELECT * FROM alunos
WHERE id = 1;
```

---

## Consultar por Nome

```
SELECT * FROM alunos
WHERE nome = 'Carlos';
```

---

# 9. Filtros e Ordenação

## Operadores

Operador	Função
=	Igual
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual
<>	Diferente

---

## Exemplos

### Alunos maiores de idade

```
SELECT * FROM alunos  
WHERE idade >= 18;
```

---

### Ordenar por nome

```
SELECT * FROM alunos  
ORDER BY nome;
```

---

### Ordem decrescente

```
SELECT * FROM alunos  
ORDER BY idade DESC;
```

---

## LIKE

Usado para buscas.

### Nome iniciando com A

```
SELECT * FROM alunos  
WHERE nome LIKE 'A%';
```

---

### Nome terminando com a

```
SELECT * FROM alunos  
WHERE nome LIKE '%a';
```

---

## 10. Atualizando e Excluindo Dados

### Atualizar Dados

```
UPDATE alunos  
SET curso = 'Programação Web'  
WHERE id = 1;
```

---

### Excluir Registro

```
DELETE FROM alunos  
WHERE id = 2;
```

---

### Apagar Todos os Dados

```
DELETE FROM alunos;
```

---

## 11. Relacionamento entre Tabelas

### Exemplo Prático

Vamos criar:

- Tabela clientes
- Tabela pedidos

Cada pedido pertence a um cliente.

---

### Criando Tabela Clientes

```
CREATE TABLE clientes (
```

```
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(100),  
email VARCHAR(100)  
);
```

---

## Criando Tabela Pedidos

```
CREATE TABLE pedidos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
produto VARCHAR(100),  
valor DECIMAL(10,2),  
cliente_id INT,  
FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```

---

## Inserindo Clientes

```
INSERT INTO clientes (nome, email)  
VALUES  
( 'João', 'joao@email.com'),  
( 'Maria', 'maria@email.com');
```

---

## Inserindo Pedidos

```
INSERT INTO pedidos (produto, valor, cliente_id)  
VALUES  
( 'Notebook', 3500.00, 1),  
( 'Mouse', 80.00, 1),  
( 'Teclado', 120.00, 2);
```

---

# 12. JOINS na Prática

## O que é JOIN?

JOIN é utilizado para juntar informações de tabelas relacionadas.

---

## INNER JOIN

```
SELECT
```

```
clientes.nome,  
pedidos.produto,  
pedidos.valor  
FROM pedidos  
INNER JOIN clientes  
ON pedidos.cliente_id = clientes.id;
```

---

## Resultado Esperado

nome	produto	valor
João	Notebook	3500.00
João	Mouse	80.00
Maria	Teclado	120.00

---

# 13. Funções e Operações

## COUNT

Conta registros.

```
SELECT COUNT(*) FROM alunos;
```

---

## AVG

Média.

```
SELECT AVG(idade) FROM alunos;
```

---

## MAX

Maior valor.

```
SELECT MAX(idade) FROM alunos;
```

---

## MIN

Menor valor.

```
SELECT MIN(idade) FROM alunos;
```

---

## SUM

Soma.

```
SELECT SUM(valor) FROM pedidos;
```

---

# 14. Backup e Restauração

## Fazer Backup

No terminal Linux:

```
mysqldump -u root -p escola > escola_backup.sql
```

---

## Restaurar Backup

```
mysql -u root -p escola < escola_backup.sql
```

---

# 15. Projeto Prático Guiado

## Sistema de Biblioteca

---

### Objetivo

Criar um pequeno banco de dados para controlar livros.

---

### Passo 1 — Criar Banco

```
CREATE DATABASE biblioteca;
```

---

## Passo 2 — Usar Banco

USE biblioteca;

---

## Passo 3 — Criar Tabela Livros

```
CREATE TABLE livros (  
id INT AUTO_INCREMENT PRIMARY KEY,  
titulo VARCHAR(150),  
autor VARCHAR(100),  
ano INT,  
disponivel BOOLEAN  
);
```

---

## Passo 4 — Inserir Dados

```
INSERT INTO livros (titulo, autor, ano, disponivel)  
VALUES  
( 'Dom Casmurro', 'Machado de Assis', 1899, true),  
( '1984', 'George Orwell', 1949, true),  
( 'O Hobbit', 'J.R.R. Tolkien', 1937, false);
```

---

## Passo 5 — Consultar Livros

### Todos os livros

```
SELECT * FROM livros;
```

---

### Livros disponíveis

```
SELECT * FROM livros  
WHERE disponivel = true;
```

---

### Livros por autor

```
SELECT * FROM livros  
WHERE autor = 'George Orwell';
```

---

## Passo 6 — Atualizar Livro

```
UPDATE livros  
SET disponivel = false  
WHERE id = 1;
```

---

## Passo 7 — Remover Livro

```
DELETE FROM livros  
WHERE id = 3;
```

---

# 16. Exercícios Propostos

## Exercício 1 — Cadastro de Produtos

### Objetivo

Criar uma tabela de produtos.

Campos:

- id
- nome
- preco
- quantidade

### Tarefas

1. Criar banco loja
  2. Criar tabela produtos
  3. Inserir 5 produtos
  4. Mostrar todos os produtos
  5. Mostrar produtos acima de R\$ 100
  6. Atualizar o preço de um produto
  7. Excluir um produto
- 

## Exercício 2 — Sistema Escolar

### Objetivo

Criar tabela de professores.

Campos:

- id
- nome
- disciplina
- salario

### Tarefas

1. Inserir 5 professores
  2. Mostrar professores ordenados por nome
  3. Mostrar salários acima de R\$ 3000
  4. Atualizar disciplina
  5. Excluir professor
- 

## Exercício 3 — Relacionamento

### Objetivo

Criar:

- tabela autores
- tabela livros

Relacionar usando chave estrangeira.

### Tarefas

1. Criar tabelas
  2. Inserir dados
  3. Fazer INNER JOIN
  4. Mostrar livros e autores
- 

## 17. Desafios Extras

### Desafio 1

Crie um sistema de:

- locadora
- hospital
- mercado

- academia
- restaurante

Com pelo menos:

- 3 tabelas
  - relacionamento entre tabelas
  - INSERT
  - SELECT
  - UPDATE
  - DELETE
  - JOIN
- 

## Desafio 2

Crie um sistema de login.

Tabela usuarios:

- id
- nome
- email
- senha

Faça consultas:

```
SELECT * FROM usuarios  
WHERE email = 'admin@email.com';
```

---

## Boas Práticas

### Utilize nomes claros

Bom:

clientes  
produtos  
pedidos

Ruim:

tab1  
xpto

abc

---

## Sempre use WHERE em UPDATE e DELETE

Perigoso:

```
DELETE FROM alunos;
```

Seguro:

```
DELETE FROM alunos  
WHERE id = 1;
```

---

## Faça backups regularmente

```
mysqldump -u root -p banco > backup.sql
```

---

# Integração com Desenvolvimento Web

O banco de dados normalmente é conectado a:

- PHP
- JavaScript
- Node.js
- Python
- APIs REST

Fluxo comum:

Usuário → Aplicação Web → Banco de Dados

---

## Conclusão

Nesta apostila você aprendeu:

- Conceitos básicos de banco de dados
- Criação de bancos e tabelas
- Inserção de dados
- Consultas SQL
- Relacionamentos

- JOINS
- Atualização e remoção de dados
- Projeto prático
- Exercícios

Esses conhecimentos são fundamentais para qualquer desenvolvedor web.