

# Aula 14

A Base da Consulta  
(DQL - SELECT, WHERE, ORDER BY)

## (Slide 1 de 4) - Lembrando da Aula Anterior

- **O que fizemos?**
  - Concluímos o ciclo básico do DML (Manipulação de Dados).
  - Aprendemos `UPDATE` para **modificar** registros.
  - Aprendemos `DELETE` para **remover** registros.
  - Dominamos a importância da cláusula `WHERE` como nosso filtro de segurança.
- **Resultado:** Sabemos construir, popular, atualizar e limpar um banco.
- **A pergunta de hoje:**
  - Como fazemos para **ler, consultar e fazer perguntas** ao nosso banco de dados?

**(Slide 2 de 4) - O Início do DQL:** `SELECT ... FROM ...`

- **DQL (Data Query Language):** A parte do SQL usada **exclusivamente** para ler dados. Ela não altera, não insere e não apaga nada.
- **Comando Principal:** `SELECT`
- **Sintaxe Básica:** `SELECT colunas FROM tabela;`
  - `SELECT` : O que você quer ver (quais colunas).
  - `FROM` : De onde você quer ver (de qual tabela).
- **Os dois modos principais:**
  - `SELECT * FROM tabela;`
    - O asterisco (`*`) é um curinga que significa "**todas as colunas**".
  - `SELECT coluna1, coluna2 FROM tabela;`
    - Lista apenas as colunas específicas que você deseja.

## (Slide 3 de 4) - Filtrando Linhas com WHERE

- É a **exata mesma** cláusula WHERE que já usamos!
- Em vez de dizer "atualize onde..." ou "apague onde...", agora dizemos "me mostre onde...".
- **Sintaxe:** SELECT colunas FROM tabela WHERE condicao;
- **Operadores de Comparação Comuns:**
  - = : Igual
  - != ou <> : Diferente
  - > : Maior que
  - < : Menor que
  - >= : Maior ou igual a
  - <= : Menor ou igual a

## (Slide 4 de 4) - Ordenando o Resultado com ORDER BY

- O comando ORDER BY é usado no final da consulta para **classificar** os resultados.
- Por padrão, o banco de dados não garante nenhuma ordem específica.
- **Sintaxe:**
  - ORDER BY coluna ASC;
    - ASC (Ascendente): É o padrão. Ordena de A-Z ou de 1-10.
  - ORDER BY coluna DESC;
    - DESC (Descendente): Ordena de Z-A ou de 10-1.
- Você pode ordenar por múltiplas colunas: ORDER BY coluna1 ASC, coluna2 DESC;

# Prática

## SQL

```
-- AULA 05 - SCRIPT PRATICO
```

```
-- Professor: [Seu Nome]
```

```
-- Data: 27/10/2025
```

```
-- 1. SETUP: Criando um banco de dados e uma tabela de produtos.
```

```
CREATE DATABASE IF NOT EXISTS aula5_exemplo;
```

```
USE aula5_exemplo;
```

```
CREATE TABLE produtos (
```

```
    produto_id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    nome VARCHAR(100) NOT NULL,
```

```
    preco DECIMAL(10, 2),
```

```
    categoria VARCHAR(50)
```

```
);
```

```
-- 2. POPULANDO A TABELA: Inserindo dados para podermos consultar.  
INSERT INTO produtos (nome, preco, categoria) VALUES  
('Notebook Gamer', 5500.00, 'Informática'),  
('Mouse Sem Fio', 120.00, 'Periféricos'),  
('Teclado Mecânico', 350.00, 'Periféricos'),  
('Monitor 27"', 1800.00, 'Monitores'),  
('Headset Gamer', 400.00, 'Periféricos');
```

### -- 3. PRÁTICA DE SELECT (DQL)

-- Consulta 1: Ver TUDO da tabela de produtos.

```
SELECT * FROM produtos;
```

-- Consulta 2: Ver apenas o NOME e o PREÇO dos produtos.

```
SELECT nome, preco FROM produtos;
```

-- Consulta 3: Ver apenas os produtos que custam menos de 500 reais.

```
SELECT * FROM produtos  
WHERE preco < 500.00;
```

-- Consulta 4: Ver apenas os produtos da categoria 'Periféricos'.

```
SELECT * FROM produtos  
WHERE categoria = 'Periféricos';
```

-- Consulta 5: Ver todos os produtos, ordenados pelo preço (do mais barato ao mais

```
SELECT * FROM produtos  
ORDER BY preco ASC; -- O ASC é opcional, pois é o padrão.
```

-- Consulta 6: Ver todos os produtos da categoria 'Periféricos',  
-- ordenados pelo preço (do mais caro ao mais barato).

```
SELECT * FROM produtos  
WHERE categoria = 'Periféricos'  
ORDER BY preco DESC;
```







# Exercícios

Altere o primeiro produto da  
tabela e exclua a última