

## ❖❖ Roteiro de Módulo 4

### 1. Dados Estruturados vs. Não-estruturados

- **Estruturados**: armazenados em tabelas (linhas e colunas), exemplo: SQL, planilhas.
- **Não-estruturados**: não seguem um modelo fixo, exemplo: imagens, vídeos, documentos.

### 2. Sintaxe básica de uma consulta SQL

- **SELECT**: define quais colunas serão retornadas.
- **FROM**: define a tabela.
- **WHERE**: adiciona filtros (condições).
- **ORDER BY**: ordena os resultados.
- **GROUP BY**: agrupa registros para funções de agregação.
- **HAVING**: filtra resultados após o agrupamento.
- **LIMIT**: limita a quantidade de registros.

: Essa estrutura garante que as consultas sejam lógicas, legíveis e processadas corretamente pelo banco.

### 3. Consultas SQL – Exemplos

- **DISTINCT e LIMIT**

```
SELECT DISTINCT genero FROM filmes LIMIT 5;
```

Evita repetição de valores e limita o resultado.

- **ORDER BY**

```
SELECT titulo, ano_lancamento FROM filmes ORDER BY  
ano_lancamento DESC;
```

Ordena crescente (**ASC**) ou decrescente (**DESC**).

- **GROUP BY e COUNT**

```
SELECT genero, COUNT(*) FROM filmes GROUP BY genero;
```

Conta quantos registros existem por grupo.

- **GROUP BY e SUM**

```
SELECT genero, SUM(nota) FROM avaliacoes GROUP BY genero;
```

Soma valores numéricos agrupados.

- **Funções de agregação**

- **AVG()** → média
- **COUNT()** → quantidade
- **SUM()** → soma
- **MAX()** → maior valor
- **MIN()** → menor valor

- **GROUP BY e HAVING**

```
SELECT genero, SUM(nota)
FROM avaliacoes
GROUP BY genero
HAVING SUM(nota) > 10;
```

Filtra resultados de grupos após agregação.

- **Números inteiros e decimais**

- `INT` → valores inteiros (ex: idade, quantidade).
- `DECIMAL(5,2)` → valores com casas decimais (ex: preços, notas).

- **LIKE (busca por padrão)**

```
SELECT titulo FROM filmes WHERE titulo LIKE 'A%';
```

Retorna títulos que começam com "A".

- **BETWEEN (intervalos)**

```
SELECT titulo FROM filmes WHERE ano_lancamento BETWEEN
2000 AND 2010;
```

Busca dentro de um intervalo.

#### 4. Tipos de Dados e Restrições

- **Tipos comuns:**

- `INT`, `DECIMAL`, `VARCHAR`, `DATE`, `BOOLEAN`

- **Restrições:**

- `NOT NULL` (não pode ser nulo)

- **UNIQUE** (valor único)
- **PRIMARY KEY** (chave primária, identifica o registro)
- **FOREIGN KEY** (chave estrangeira, conecta tabelas)

#### Quando usar:

- **Chave primária (PK):** identifica unicamente um registro (ex: ID do cliente).
- **Chave estrangeira (FK):** faz referência à PK de outra tabela (ex: cliente\_id em pedidos).

### 5. Remoção de linhas no Banco de Dados

```
DELETE FROM filmes WHERE id = 10;
```

Importante: sem **WHERE**, remove todos os registros.

### 6. CRUD

- **Create (INSERT)**

```
INSERT INTO filmes (titulo, genero, ano_lancamento)
VALUES ('Matrix', 'Ficção', 1999);
```

- **Read (SELECT)** → consultas

- **Update (UPDATE)**

```
UPDATE filmes SET genero = 'Ação' WHERE id = 1;
```

- **Delete (DELETE)** → remoção de registros

### 💡💡 Mini-Projeto – Entrega

#### 💡💡 Objetivo:

Praticar o **CRUD completo** (Create, Read, Update, Delete) em banco de dados usando SQL.

#### 💡💡 Material:

- Notebook Colab: [Mini-projeto - Módulo 4 - Ciclo 14.ipynb](#)

#### ✅ O que deve ser entregue:

💡💡 Data de entrega: 28/09/2025 até 23h59