

A12. DML: **INSERT, UPDATE E DELETE**

PROF. WILLIAM C. AUGUSTONELLI (BILLY)

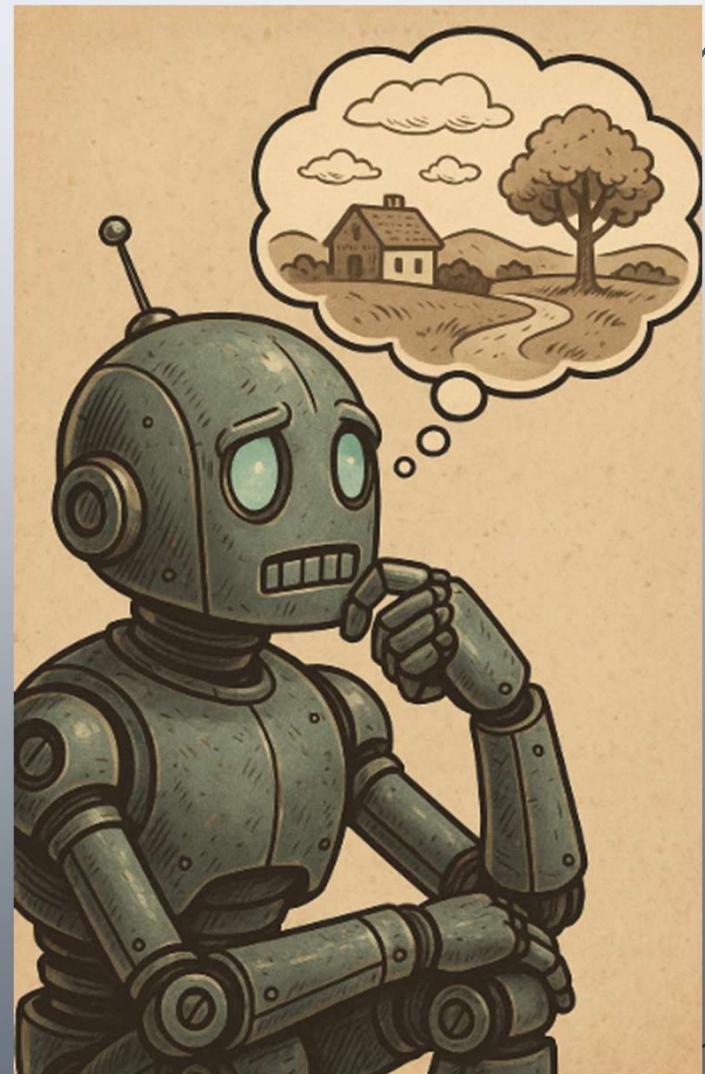
WILLIAM.AUGUSTONELLI@DOCENTE.SENAI.BR – 2S2025

OBJETIVO

- Compreender o papel da **DML** (*Data Manipulation Language*) em um SGBD relacional
- Executar comandos **INSERT**, **UPDATE** e **DELETE** no MySQL
- Manipular registros de tabelas previamente criadas

NA ÚLTIMA AULA...

- Índices em Banco de Dados
 - O que são e para que servem
 - Tipos de índices
- DDL – CREATE INDEX e DROP INDEX
- Migração de Dados
 - Exportação
 - Importação



NOSSA AULA DE HOJE...

- Introdução ao DML
- Comando INSERT
- Comando UPDATE
- Comando DELETE

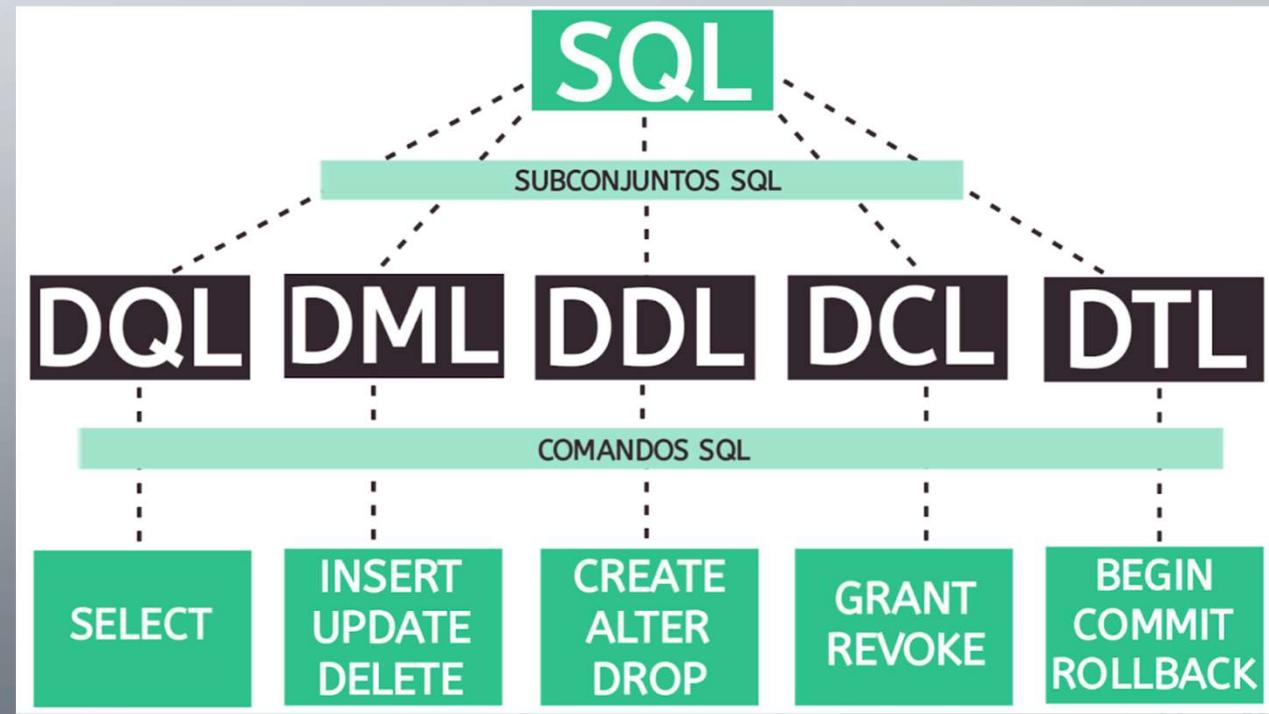


O QUE É DML?

- **DML** (*Data Manipulation Language*): conjunto de comandos SQL usados para manipular dados nas tabelas
- Diferente do **DDL** (que cria ou altera estruturas), o DML trabalha com **conteúdo**

Principais comandos

- **INSERT** – inserir registros
- **UPDATE** – atualizar registros
- **DELETE** – excluir registros



D1. DEMONSTRAÇÃO PRÁTICA



COMANDO INSERT

- O comando **INSERT** é usado para **adicionar novos registros** em uma tabela já existente
- Sempre respeita a **estrutura da tabela** (coluna e tipos de dados)
- Os valores devem estar na mesma **ordem** das colunas especificadas
- **Sintaxe básica**

```
INSERT INTO tabela (coluna1, coluna2, coluna3)
VALUES (valor1, valor2, valor3);
```

```
INSERT INTO clientes (id, nome, email)
VALUES (34, 'Maria', 'maria@aula.com');
```

- Inserindo em todas as colunas

```
INSERT INTO clientes values (34, 'Maria', 'maria@aula.com');
```

- Inserindo em colunas específicas

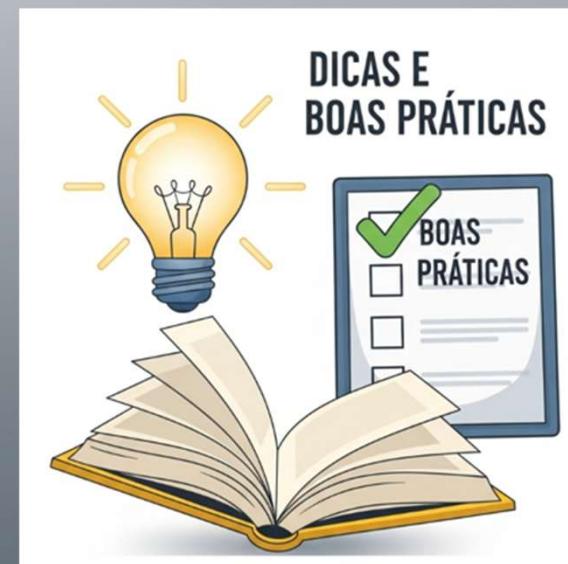
```
INSERT INTO clientes (nome, email)
VALUES ('Maria', 'maria@aula.com');
```

BOAS PRÁTICAS

- Sempre indicar os nomes das colunas – mais seguro em caso de mudanças futuras
- Usar tipos de dados corretos (‘texto’, números sem aspas, datas no formato ‘YYYY-MM-DD’)
- Inserir vários registros de uma vez quando possível – melhora o desempenho

Erros Comuns

- Tentar inserir em uma coluna que não existe
- Esquecer aspas em valores de texto
- Ordem dos valores não compreender à ordem das colunas



P1.
VAMOS PRATICAR....



COMANDO UPDATE

- O comando **UPDATE** é usado para **alterar os valores** de um ou mais registros em uma tabela existente
- **Sintaxe básica**

```
• UPDATE tabela  
    SET coluna1 = valor1, coluna2 = valor2  
    WHERE condição;  
  
-- Exemplo:  
  
• UPDATE clientes  
    SET cidade = 'Rio de Janeiro'  
    WHERE id = 3;
```

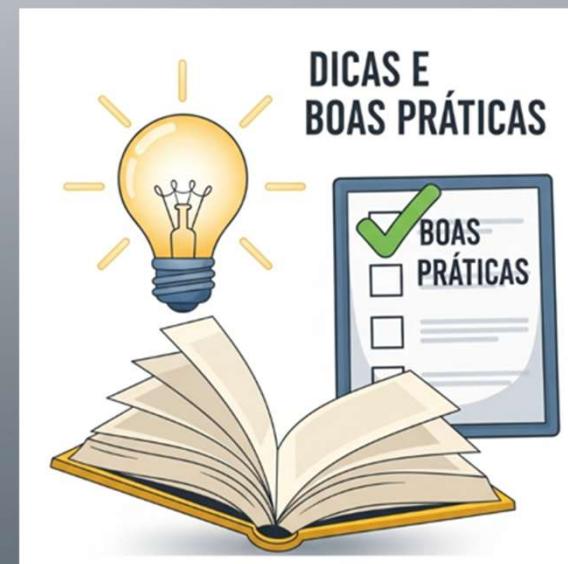
Atenção: NUNCA esquecer o WHERE, senão todos os registros são atualizados

BOAS PRÁTICAS

- Sempre usar **WHERE** para limitar os registros a serem atualizados
- Conferir antes com um **SELECT** a condição que será aplicada
- Usar transações (**BEGIN** , **ROLLBACK** , **COMMIT**) quando atualizar dados sensíveis

Erros Comuns

- Esquecer o **WHERE** – modifica a tabela inteira
- Atualizar coluna com valor incompatível com o tipo de dados
- Usar apelidos de tabela incorretas em **joins**



P2.
VAMOS PRATICAR...



COMANDO DELETE

- O comando **DELETE** remove registros de uma tabela
- Sempre afeta **linhas**
(linhas = registros) e não a estrutura da tabela)
- Combinado ao WHERE, pode remover **um registro específico ou vários**
- **Sintaxe básica**

```
DELETE FROM tabela WHERE condição;
```

-- Exemplo

```
DELETE FROM clientes WHERE id = 5;
```

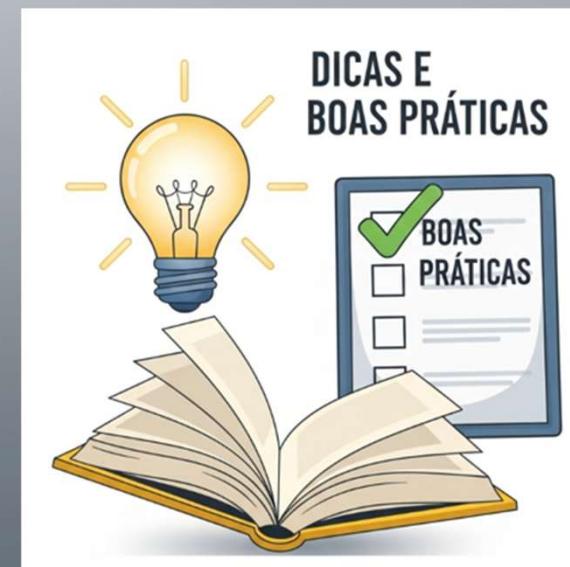
Atenção: Sem o WHERE, todos os registros serão apagados

BOAS PRÁTICAS

- Sempre usar WHERE ao excluir registros
- Testar antes com um SELECT usando a mesma condição
- Em dados críticos – usar transações (BEGIN/ ROLLBACK/COMMIT)

Erros Comuns

- Esquecer WHERE – apaga todos os registros
- Usar condição incorreta – excluir registros errados ou nenhum
- Confundir DELETE com DROP TABLE



D3.
VAMOS PRATICAR...



Perguntas?! Dúvidas?!?

