A faint, dark gray background image of a printed circuit board (PCB) with various tracks, capacitors, and resistors.

A10. DDL: CREATE TABLE ALTER TABLE DROP TABLE

PROF. WILLIAM C. AUGUSTONELLI (BILLY)

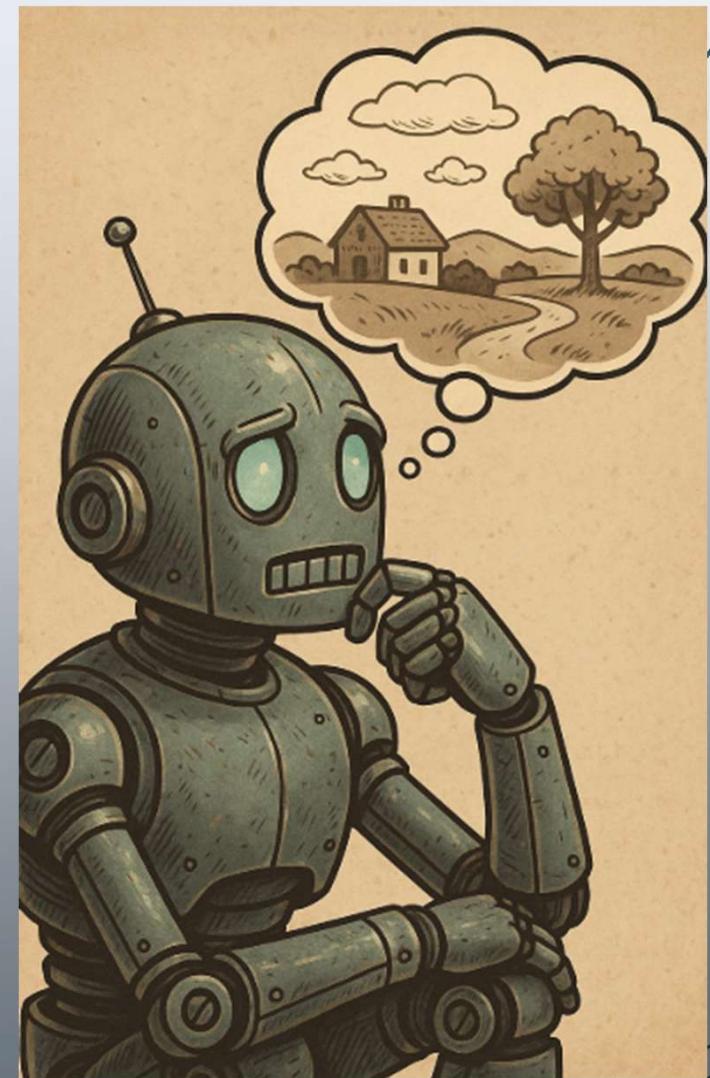
WILLIAM.AUGUSTONELLI@DOCENTE.SENAI.BR – 2S2025

OBJETIVO

- Ensinar a criar tabelas relacionais utilizando DDL
- Explicar como alterar a estrutura de tabelas existentes
- Mostrar como excluir tabelas e cuidados necessários
- Desenvolver a habilidade prática de criação e modificação de tabelas em um SGBD real

NA ÚLTIMA AULA...

- Introdução ao SQL
- Categorias da Linguagem
- DDL – Data Definition Language
 - CREATE DATABASE
 - DROP DATABASE
 - USE



NOSSA AULA DE HOJE...

- CREATE TABLE
 - Principais tipos de dados
 - Principais restrições
- ALTER TABLE
 - Ações comuns
- DROP TABLE
 - Boas Práticas

CREATE TABLE

O que é **CREATE TABLE**?

- É um comando DDL (Data Definition Language)
- Serve para **criar uma nova tabela** dentro de um banco de dados
- Define **colunas, tipos de dados e restrições**
- É a **base da estrutura de um banco relacional**

Estrutura básica do comando

```
create table nome_da_tabela (
    nome_coluna1 tipo_de_dado restrição,
    nome_coluna2 tipo_de_dado restrição,
    nome_coluna3 tipo_de_dado restrição,
    ...
);
```

CREATE TABLE – PRINCIPAIS TIPOS DE DADOS

- **Numéricos:** INT, BIGINT, DECIMAL(10, 2), FLOAT
- **Texto:** CHAR(n), VARCHAR(n), TEXT
- **Datas e Horas:** DATE, DATETIME, TIMESTAMP, TIME
- **Booleano:** BOOLEAN

```
create table alunos (
    id          bigint      not null auto_increment primary key,
    cpf         varchar(20)  not null,
    nome        varchar(120) not null,
    codMatricula varchar(10) not null unique,
    nomePai     varchar(120),
    nomeMae     varchar(120),
    responsavel varchar(120),
    cep         varchar(8)   not null,
    numeroEnd  varchar(20)  not null,
    complemento varchar(120)
);
```

CREATE TABLE – PRINCIPAIS RESTRIÇÕES

- **primary key** – identifica cada registro de forma única
- **foreign key** – relaciona tabelas
- **not null** – não permite valor nulo
- **unique** – não permite valores repetidos
- **default** – define um valor padrão
- **check** – impõe condição para os valores

```
create table alunos (
    id          bigint      not null auto_increment primary key,
    cpf         varchar(20)  not null,
    nome        varchar(120) not null,
    codMatricula varchar(10) not null unique,
    nomePai     varchar(120),
    nomeMae     varchar(120),
    responsavel varchar(120),
    cep         varchar(8)   not null,
    numeroEnd   varchar(20)  not null,
    complemento varchar(120)
);
```

ALTER TABLE

O que é **ALTER TABLE**?

- É um comando **DDL** usado para **modificar a estrutura de uma tabela existente**
- Permite **adicionar, remover ou alterar** colunas e restrições

Sintaxe Geral

```
alter table nome_tabela
    ação;
```

ALTER TABLE – AÇÕES COMUNS

- **Adicionar coluna**

- Adiciona uma nova coluna na tabela
- Pode ter restrições como **not null**, **default**, etc.

- **Remover coluna**

- Remove a coluna da tabela
- Atenção: todos os dados dessa coluna serão perdidos!

- **Modificar tipo de coluna**

- Altera informações (tipo, tamanho) da coluna

- **Adicionar restrição**

- Cria uma nova regra para a coluna

- **Remover restrição**

- Remove uma regra existente, em MySQL é necessário conhecer o nome da constraint

```
CREATE TABLE Clientes (
    idCliente INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(150) UNIQUE,
    dataCadastro DATE DEFAULT (CURRENT_DATE),
    idade INT CHECK (idade >= 18)
);
```

```
alter table clientes add telefone varchar(20);
```

```
alter table clientes drop column telefone;
```

```
alter table clientes modify nome varchar(150);
```

```
alter table clientes add valorCredito decimal(18,2);
alter table clientes add constraint valorCredito check (valorCredito > 0.0);
```

```
alter table clientes drop check valorCredito;
```

DROP TABLE

O que é *DROP TABLE*?

- É um comando **DDL** utilizado para **excluir completamente uma tabela do banco de dados**
- Apaga a **estrutura da tabela** e todos os **dados armazenados nela**
- Diferente do **DELETE**, que remove apenas os registros, mantendo a estrutura

Sintaxe Básica

```
drop table nome_tabela;
```

DROP TABLE – BOAS PRÁTICAS

1. Verifique dependências

- Se a tabela tem **chaves estrangeiras** ligadas a outras tabelas
- Ex.: não é possível dropar Clientes se Pedidos ainda depende dela

2. Fazer backup

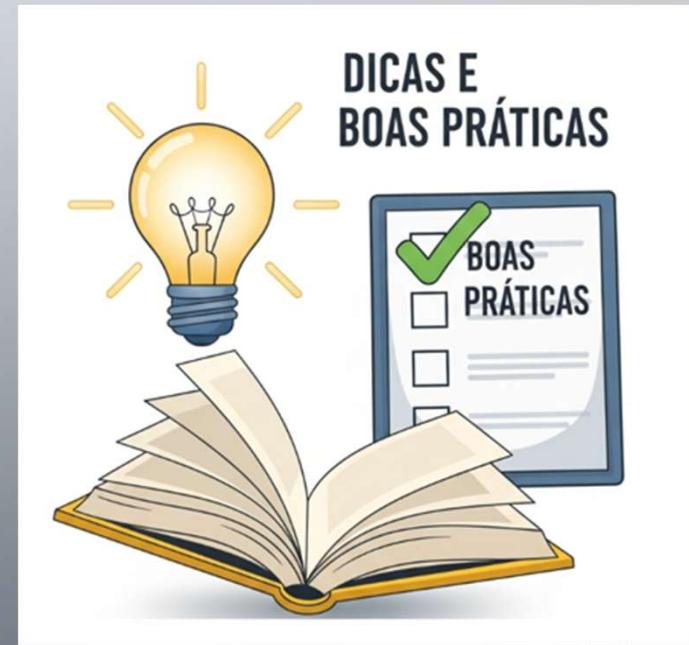
- Exportar dados: via mysqldump, pg_dump, ou Workbench

3. Usar com cautela em produção

- Evite executar direto sem revisão
- Prefira ambientes de **teste** primeiro

4. Confirmar se não basta TRUNCATE ou DELETE

- Muitas vezes, só queremos limpar os dados, não apagar a tabela inteira



Perguntas?! Dúvidas?!?

