

Aula 10

Conhecendo o Ambiente e
Criando Nosso Banco (DDL)

(Slide 1 de 4) - O Ponto de Partida

- **Na aula anterior...** (Conceitual)
 - Discutimos o Modelo Entidade-Relacionamento (MER).
 - O MER é o nosso **projeto lógico**, a planta baixa da casa.
- **O objetivo de hoje:**
 - Começar a **construir a casa**.
 - Transformar o projeto lógico (MER) em um **modelo físico** (tabelas reais em um banco de dados).
 - Usaremos a linguagem SQL para dar as ordens de construção.

(Slide 2 de 4) - Nosso Ambiente: MySQL Workbench

- É a nossa **caixa de ferramentas** (IDE) para interagir com o banco de dados MySQL.
- **Áreas principais que usaremos hoje:**
 - **Navigator (Schemas):** À esquerda, mostra os bancos de dados (`schemas`) e tabelas existentes.
 - **Editor de SQL:** Onde escrevemos nossos códigos (scripts).
 - **Botão de Execução (⚡):** O "raio" que executa nosso código.
 - **Painel de Saída (Output):** Mostra se os comandos deram certo ou errado.

(Slide 3 de 4) - DDL: Criando o Banco de Dados

- **DDL (Data Definition Language):** Parte do SQL usada para **definir** ou **modificar** a estrutura do banco.
- **Comandos de hoje:**
 - `CREATE DATABASE nome_do_banco;`
 - Cria o "container", o nosso terreno onde as tabelas serão construídas.
 - `USE nome_do_banco;`
 - "Entra" no banco de dados. Avisa o sistema em qual terreno vamos trabalhar.

(Slide 4 de 4) - DDL: Criando a Primeira Tabela

- `CREATE TABLE nome_da_tabela (...);`
 - Cria uma tabela dentro do banco de dados selecionado.
 - **Estrutura básica de uma coluna:** `nome_da_coluna TIPO_DE_DADO;`
 - **Tipos de Dados Essenciais:**
 - `INT` : Para números inteiros (IDs, quantidades).
 - `VARCHAR(tamanho)` : Para textos de tamanho variável (nomes, títulos).
 - `DATE` : Para datas (data de nascimento, data de publicação).
 - **A Chave Primária (`PRIMARY KEY`):**
 - É o "CPF" da linha, um identificador único que não pode se repetir.
 - Usamos `AUTO_INCREMENT` para que o MySQL numere os IDs automaticamente para nós.
-

Prática

```
1  -- AULA 01 - SCRIPT PRÁTICO
2
3  -- 1. Comando para criar nosso banco de dados da biblioteca.
4  --      Usamos "IF NOT EXISTS" como uma segurança para não dar erro se ele já existir.
5  ● CREATE DATABASE IF NOT EXISTS biblioteca_curso;
6
```

```
7      -- 2. Comando para selecionar o banco que acabamos de criar.
8      --      A partir daqui, tudo acontece dentro de 'biblioteca_curso'.
9  ●    USE biblioteca_curso;
10
```



```
11      -- 3. Comando para criar nossa primeira tabela: 'autores'.
12      --      Note a sintaxe: CREATE TABLE nome (colunas aqui dentro);
13  ● ⊖ CREATE TABLE autores (
14      -- 'autor_id' é a chave primária. INT para número, AUTO_INCREMENT para numerar sozinho.
15      autor_id INT PRIMARY KEY AUTO_INCREMENT,
16
17      -- 'nome' e 'sobrenome' são textos. VARCHAR é o tipo ideal.
18      nome VARCHAR(100),
19      sobrenome VARCHAR(100),
20
21      -- 'data_nascimento' armazenará uma data no formato 'AAAA-MM-DD'.
22      data_nascimento DATE
23  );
```

Exercícios

1. **Criar Tabela de Editoras:** Crie uma nova tabela chamada `editoras` com as seguintes colunas:

- `editora_id` (chave primária, inteiro, auto-incremento)
- `nome` (texto, até 150 caracteres)

2. **Criar Tabela de Gêneros:** Crie uma tabela chamada `generos` com as colunas:
- `genero_id` (chave primária, inteiro, auto-incremento)
 - `nome` (texto, até 100 caracteres)

3. **Criar Tabela de Usuários:** Crie uma tabela `usuarios` com:

- `usuario_id` (chave primária, inteiro, auto-incremento)
- `nome` (texto, até 100 caracteres)
- `email` (texto, até 150 caracteres)

4. **Escrever o Comando:** Apenas escreva o comando que seria usado para **apagar** a tabela `generos` (não precisa executar!).

Desafios de Complexidade Média:

1. **Tabela de Livros (Versão Inicial):** Crie uma tabela `livros`. Pense bem nos tipos de dados. Ela deve ter:
 - `livro_id` (chave primária, inteiro, auto-incremento)
 - `titulo` (texto, até 255 caracteres)
 - `ano_publicacao` (um número inteiro para o ano)
 - `isbn` (texto, 13 caracteres - por que não um número?)

2. **Tabela de Endereços:** Crie uma tabela `enderecos_usuarios` para armazenar os endereços dos usuários. Ela deve conter:

- `endereco_id` (chave primária, inteiro, auto-incremento)
- `rua` (texto, até 255 caracteres)
- `numero` (texto, até 10 caracteres - por que não um número?)
- `cidade` (texto, até 100 caracteres)
- `estado` (texto, 2 caracteres)
- `cep` (texto, 8 caracteres)