

Aula 11

Constraints e Relacionamentos (DDL)

(Slide 1 de 4) - Lembrando da Aula Anterior

- **O que fizemos?**
 - Nos conectamos ao banco pelo Workbench.
 - Usamos `CREATE DATABASE` para criar nosso espaço de trabalho.
 - Usamos `CREATE TABLE` para construir nossa primeira tabela: `autores`.
- **Resultado:** Temos uma tabela "solta", uma peça do nosso quebra-cabeça.
- **O objetivo de hoje:**
 - Criar a tabela `livros` e conectá-la à tabela `autores`, montando as primeiras peças.
 - Adicionar regras para garantir a qualidade dos nossos dados.

(Slide 2 de 4) - Garantindo a Qualidade: Constraints

- **Constraints** são **regras** que aplicamos às colunas para garantir a integridade dos dados.
- **NOT NULL**
 - Regra: "Este campo é obrigatório".
 - Impede que um registro seja inserido com essa coluna vazia (nula).
 - Exemplo: Um livro **precisa** ter um título.
- **UNIQUE**
 - Regra: "O valor nesta coluna não pode se repetir em toda a tabela".
 - Útil para campos que não são a chave primária, mas que também devem ser únicos.
 - Exemplo: E-mail de usuário, CPF, ou o código ISBN de um livro.

(Slide 3 de 4) - A Conexão: Chave Estrangeira (FOREIGN KEY)

- É o conceito **mais importante** para criar relacionamentos.
- **O que é?** Uma coluna em uma tabela que "aponta" para a PRIMARY KEY de outra tabela.
- **Para que serve?** Para garantir a **Integridade Referencial**.
 - O sistema não permitirá que você insira um livro com `autor_id = 99` se não existir um autor com `autor_id = 99` na tabela `autores`.
 - Impede que existam dados "órfãos" no banco.

(Slide 4 de 4) - A Sintaxe do Relacionamento

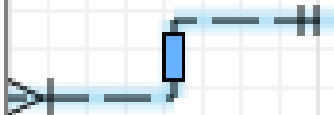
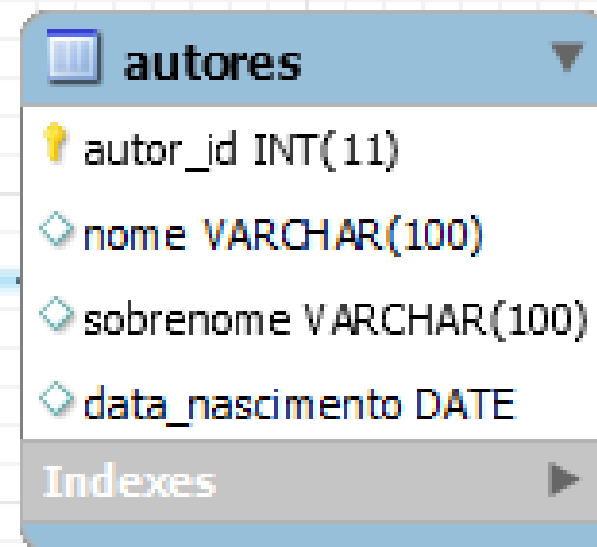
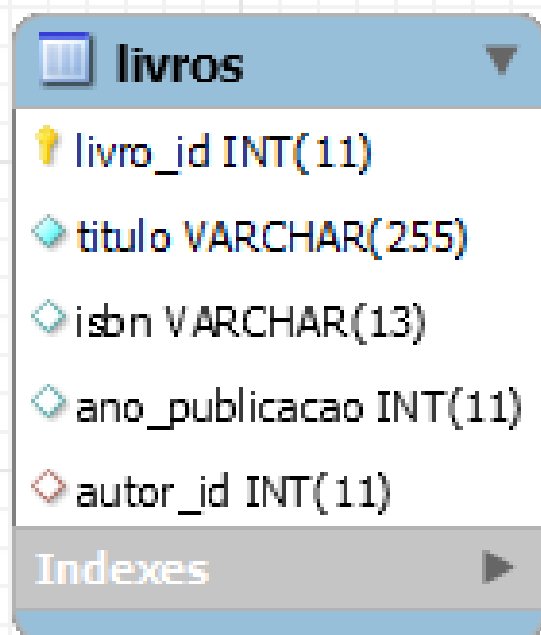
- Para criar a conexão, seguimos dois passos dentro do `CREATE TABLE` :
 - **1. Criar a coluna:**
 - `autor_id INT`
 - Cria o campo que vai armazenar o ID do autor. O tipo de dado deve ser o mesmo da chave primária que ele referencia.
 - **2. Definir a Constraint:**
 - `CONSTRAINT nome_da_regra`
 - É uma boa prática nomear suas constraints.
 - `FOREIGN KEY (coluna_dest_a_tabela)`
 - Diz qual coluna será a chave estrangeira.
 - `REFERENCES outra_tabela(coluna_da_outra_tabela)`
 - Diz para qual tabela e qual chave primária ela está apontando.

Prática

Revisando aula 1

```
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 );
```

```
1  -- Garante que estamos usando o banco de dados correto.
2  ●  USE biblioteca_curso;
3
4  -- Vamos criar a tabela 'livros' e já aplicar as constraints.
5  ● ⊖ CREATE TABLE livros (
6      -- Chave primária, igual à da tabela autores.
7      livro_id INT PRIMARY KEY AUTO_INCREMENT,
8
9      -- O título é obrigatório, por isso usamos a constraint NOT NULL.
10     titulo VARCHAR(255) NOT NULL,
11
12     -- O ISBN (International Standard Book Number) deve ser único por livro.
13     isbn VARCHAR(13) UNIQUE,
14
15     ano_publicacao INT,
16
17     -- 1. Criamos a coluna que vai guardar a referência ao autor.
18     -- Ela precisa ter o mesmo tipo de dado da coluna 'autor_id' da tabela 'autores' (INT).
19     autor_id INT,
20
21     -- 2. Criamos a constraint que estabelece a relação de fato.
22     -- Damos um nome para a nossa regra: fk_livro_autor
23     CONSTRAINT fk_livro_autor
24     -- Definimos qual coluna em 'livros' é a chave estrangeira
25     FOREIGN KEY (autor_id)
26     -- E informamos que ela referencia a coluna 'autor_id' na tabela 'autores'
27     REFERENCES autores(autor_id)
28 );
```

Exercícios

1. **Criar Tabela de Editoras:** Crie uma tabela chamada `editoras` com as seguintes colunas:
 - `editora_id` (chave primária, inteiro, auto-incremento)
 - `nome` (texto, até 150 caracteres, e **não pode ser nulo**)
 - `telefone` (texto, até 20 caracteres)

2. **Criar Tabela de Leitores:** Crie uma tabela `leitores` com as seguintes colunas:

- `leitor_id` (chave primária, inteiro, auto-incremento)
- `nome` (texto, até 100 caracteres, **não pode ser nulo**)
- `email` (texto, até 150 caracteres, e deve ser **único**)

3. **Ligar o Livro à Editora:** Crie uma tabela chamada `livro_editora`. O objetivo é apenas registrar qual livro (`livro_id` da tabela `livros`) foi publicado por qual editora (`editora_id` da tabela `editoras`). A tabela deve conter:
- `livro_id` (inteiro, que será uma **chave estrangeira** para a tabela `livros`)
 - `editora_id` (inteiro, que será uma **chave estrangeira** para a tabela `editoras`)

4. **Escrever o Comando:** Apenas escreva o comando `CREATE TABLE` para uma tabela `enderecos`, que se relaciona com a tabela `leitores`. A tabela deve ter as colunas `endereco_id` (PK), `rua`, `cidade` e `leitor_id` (sendo esta última a **chave estrangeira** para a tabela `leitores`).