

# Bases de Dados

**Módulo 7: Modelo Relacional – Definição de Restrições na Linguagem SQL**

**Prof. André Bruno de Oliveira**

Data 14/12/2023 19:54

# Tópicos

- Aula teórica e prática
- Definição de Restrição com uso do SQL
  - **Chaves primárias**
  - Campos **não nulos** (NOT NULL – NN),
  - **Integridade referencial** (chaves estrangeiras – FK)
  - Restrições **CHECK** (CK)
  - Aprender como atualizar tabelas respeitando estas restrições
  - Exercícios

# Definição de Restrições na Linguagem SQL

## OBJETIVO

- Esta aula tem por objetivo apresentar a forma de definir esquemas considerando quatro tipos diferentes de restrição: **chaves primárias** (PK), campos **não nulos** (NOT NULL – NN), **integridade referencial** (**chaves estrangeiras** – FK) e restrições **CHECK** (CK).
- A prática também demonstra a forma correta (e a incorreta!) de utilizar os comandos manipulação de dados INSERT, DELETE e UPDATE em um banco de dados onde as tabelas são definidas com estas restrições.

# Definição de Restrições na Linguagem SQL

## EXECUÇÃO

### PASSO 1 – Criar o banco de dados “Cinema.db”

- Execute o programa SQLiteStudio e crie um novo banco de dados chamado “Cinema.db”. A seguir efetue a conexão com o banco recém-criado (caso você esteja em dúvida sobre como executar esta tarefa, consulte a Aula 03).

# Definição de Restrições na Linguagem SQL

## EXECUÇÃO

### PASSO 2 – Conhecendo as Tabelas e suas Restrições

- Nesta prática, criaremos 6 tabelas para um BD sobre filmes: *País*, *Filme*, *Artista*, *Genero*, *FilmeGenero* e *FilmeElenco*. Cada tabela será criada respeitando as restrições descritas no quadro a seguir:

Tabela	Restrição
<i>País</i>	A sigla e o nome devem ser obrigatórios. A sigla deverá representar a PK.
<i>Filme</i>	O título, ano e resumo devem ser obrigatórios. O título e ano e representarão a chave primária. A tabela só deve aceitar um novo filme caso o país contenha o valor nulo ou sua sigla esteja cadastrada na tabela <i>País</i> . Ou seja: deve existir FK para a tabela <i>País</i> .
<i>Artista</i>	O nome e o sexo ('F' ou 'M') devem ser obrigatórios. O nome representará a PK.
<i>Genero</i>	O nome deve ser obrigatório e representará a PK.
<i>FilmeGenero</i>	O título, ano e nome do gênero são obrigatórios. Os três campos representarão a PK. Um filme só poderá ser aceito nesta tabela se estiver cadastrado na tabela <i>Filme</i> . De maneira análoga, apenas gêneros cadastrados na tabela <i>Genero</i> serão aceitos. Ou seja: deverão existir FKs para as tabelas <i>Filme</i> e <i>Genero</i> .
<i>FilmeElenco</i>	O título, ano e nome do artista são obrigatórios. Os três campos representarão a PK. Um filme só poderá ser aceito nesta tabela se estiver cadastrado na tabela <i>Filme</i> . De maneira análoga, apenas artistas cadastrados na tabela <i>Artista</i> serão aceitos. Ou seja: deverão existir FKs para as tabelas <i>Filme</i> e <i>Artista</i> .

# Definição de Restrições na Linguagem SQL

## PASSO 3 – Definindo a Ordem de Criação das Tabelas

- Em um BD deste tipo, as tabelas precisam ser criadas de acordo com uma ordem adequada.
  - Basicamente, se uma tabela *A* possui uma chave estrangeira que referencia a chave primária de uma tabela *B*, então *B* precisa ser definida antes de *A*.
  - Segundo os requisitos definidos na tabela anterior, *Filme* possui chave estrangeira referenciando *Pais*. Sendo assim, é necessário definir (criar a tabela) *Pais* antes de definir (criar a tabela) *Filme*. De maneira análoga, a tabela *FilmeGenero* depende das tabelas *Filme* e *Genero* e a tabela *FilmeElenco* depende das tabelas *Filme* e *Artista*.

# Definição de Restrições na Linguagem SQL

## PASSO 3 – Definindo a Ordem de Criação das Tabelas

- Considerando as dependências mencionadas, nesta prática definiremos a seguinte ordem para a criação das tabelas:
  - *Pais*,
  - *Filme*,
  - *Genero*,
  - *Artista*,
  - *FilmeGenero* e
  - *FilmeElenco*.
- Outras ordens seriam possíveis, contanto que respeitassem as relações de dependência. Por exemplo: criar primeiro *Pais*, *Genero* e *Artista* e depois *Filme*, *FilmeElenco* e *FilmeGenero*.

# Definição de Restrições na Linguagem SQL

## PASSO 4 – Criando e Preenchendo a Tabela *Pais* (**EXEMPLOS 2**)

O *script* abaixo realiza a criação da tabela *Pais*, respeitando as restrições definidas no quadro da página 1:

```
CREATE TABLE Pais (  
  sigla CHAR(2) NOT NULL,  
  nome VARCHAR(60) NOT NULL,  
  PRIMARY KEY (sigla)  
);
```

### Explicação:

- Este *script* define “sigla” e “nome” como campos obrigatórios (NOT NULL).
- Já a especificação PRIMARY KEY (sigla) define que o campo “sigla” representará a **chave primária** da tabela.



# Definição de Restrições na Linguagem SQL

- **PASSO 4 – Criando e Preenchendo a Tabela *Pais* (EXEMPLOS 3)**
- Vamos agora popular (preencher) a tabela criada. Demonstraremos diversos exemplos de comandos INSERT. Alguns deles falharão por violarem a restrição de NOT NULL e outros por violar a PK. Já alguns outros comandos executarão de forma bem-sucedida. Para cada comando, exibiremos a mensagem de status retornada pelo SQLite.

```
INSERT INTO PAIS VALUES('BR',NULL);
```

Error while executing SQL query on database 'cinema': NOT NULL constraint failed: Pais.nome

O comando acima falha porque houve uma tentativa de inserir um país sem “nome”, que é um campo NOT NULL.

```
INSERT INTO PAIS VALUES(NULL,'Brasil');
```

Error while executing SQL query on database 'cinema': NOT NULL constraint failed: Pais.sigla

O comando acima também falha por violação da *constraint* NOT NULL. Houve tentativa de inserir um país sem “sigla” (que além de tudo, é o campo PK da tabela!).

# Definição de Restrições na Linguagem SQL

- PASSO 4 – Criando e Preenchendo a Tabela *Pais* (EXEMPLOS 4)**

```
INSERT INTO PAIS VALUES('BR','Brasil');
```

Query finished in 0.204 second(s). Rows affected: 1

Comando executado com sucesso. A linha foi inserida com “sigla” e “nome”.

```
INSERT INTO PAIS VALUES('BR','xxxxxx');
```

Error while executing SQL query on database 'cinema': UNIQUE constraint failed: Pais.sigla

Ao tentar inserir pela segunda vez um país com a “sigla” 'BR', há violação da restrição de chave primária. Não podem existir dois registros com a mesma “sigla” na tabela *Pais*, mesmo que o valor do campo “nome” seja diferente em cada registro ('Brasil' e 'xxxxxx').

# Definição de Restrições na Linguagem SQL

- PASSO 4 – Criando e Preenchendo a Tabela *Pais* (EXEMPLOS 5)**

```
INSERT INTO Pais VALUES('US','Estados Unidos');  
INSERT INTO Pais VALUES('UK','Reino Unido');  
INSERT INTO Pais VALUES('FR','França');  
Query finished in 0.308 second(s). Rows affected: 3
```

Estes comandos são executados com sucesso. Nenhuma restrição é violada. O SELECT abaixo demonstrará que a tabela Pais possui agora 4 registros:

```
SELECT * FROM Pais;
```

	sigla	nome
1	BR	Brasil
2	US	Estados Unidos
3	UK	Reino Unido
4	FR	França

# Definição de Restrições na Linguagem SQL

- **PASSO 5 – Atualizando um Registro da Tabela *Pais* (EXEMPLO 1)**
- Tente agora executar o UPDATE a seguir:

```
UPDATE Pais SET sigla='FR' WHERE nome='Reino Unido';
```

Error while executing SQL query on database 'cinema': UNIQUE constraint failed: Pais.sigla

O comando falha, pois houve uma tentativa de modificar a “sigla” do país de “nome” Reino Unido de 'UK' para 'FR'. Isto provoca uma tentativa de violação de restrição de unicidade, uma vez que já existe outro registro nesta tabela com a sigla 'FR'.

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 1)**
- O *script* abaixo realiza a criação da tabela *Filme*, respeitando as restrições definidas pelos requisitos do *slide* 4:

```
CREATE TABLE Filme (  
  titulo VARCHAR(100) NOT NULL,  
  ano INT NOT NULL,  
  resumo VARCHAR(2000) NOT NULL,  
  pais CHAR(2),  
  duracao INT,  
  avaliacao NUM,  
  PRIMARY KEY (titulo, ano),  
  FOREIGN KEY (pais) REFERENCES Pais(sigla)  
);
```

## Explicação:

- Neste script, definimos os campos “titulo”, “ano” e “resumo” como obrigatórios (NOT NULL).

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 1)**
- A especificação PRIMARY KEY (titulo, ano) é utilizada para definir que os campos “titulo” e “ano” serão utilizados como chave primária da tabela. Ou seja: trata-se de uma chave primária composta.
- **A maior novidade nessa tabela** é a especificação FOREIGN KEY (pais) REFERENCES Pais(sigla), que define uma restrição do tipo **chave estrangeira**. Ela determina a seguinte *constraint*: a coluna “pais” da tabela *Filme* (FOREIGN KEY (pais)) só poderá aceitar valores que estejam gravados na coluna “sigla” da tabela *Pais* (REFERENCES Pais(sigla)).

```
CREATE TABLE Filme (  
  titulo VARCHAR(100) NOT NULL,  
  ano INT NOT NULL,  
  resumo VARCHAR(2000) NOT NULL,  
  pais CHAR(2),  
  duracao INT,  
  avaliacao NUM,  
  PRIMARY KEY (titulo, ano),  
  FOREIGN KEY (pais) REFERENCES Pais(sigla)  
);
```

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 2)**
- Agora execute os comandos DML abaixo e observe cada resultado obtido:

```
INSERT INTO Filme (titulo) VALUES ('Cidade de Deus');
```

Error while executing SQL query on database 'cinema': NOT NULL constraint failed: Filme.ano

- O comando acima falha porque houve uma tentativa de inserir um filme sem “ano”, que além de ser um campo NOT NULL faz parte da chave primária. O comando também falharia por não ter “resumo” (que também é NOT NULL), mas o SQLite nem precisou fazer esse teste, já que a primeira *constraint* definida (“ano” deve ser NOT NULL) falhou.

# Definição de Restrições na Linguagem SQL

- PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 3)**

```
INSERT INTO Filme (titulo, ano, resumo)
VALUES ('Cidade de Deus',2002,'Two boys growing up in a violent neighborhood of Rio
de Janeiro take different paths: one becomes a photographer, the other a drug
dealer.');
```

Query finished in 0.133 second(s). Rows affected: 1

- Esta linha é inserida com sucesso, pois todos os campos obrigatórios foram especificados: “titulo”, “ano” e “resumo”. O campo “pais”, apesar de ser chave estrangeira, não possui a restrição NOT NULL.



# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO4)**

```
INSERT INTO Filme (titulo, ano, resumo)  
VALUES ('Cidade de Deus',2002,'bla bla bla');
```

Error while executing SQL query on database 'cinema': UNIQUE constraint failed:  
Filme.titulo, Filme.ano

- Falha! Já existe um filme com o “titulo”=‘Cidade de Deus’ e o “ano”=2002. Mesmo que o “resumo” seja diferente, ocorre uma violação de PK (não podem existir duas linhas com o mesmo título e ano na tabela *Filme*).

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 5)**

```
INSERT INTO Filme (titulo, ano, pais, duracao) VALUES ('A Dama de Ferro',2011,'UK',105);
```

Error while executing SQL query on database 'cinema': NOT NULL constraint failed: Filme.resumo

- Falha! Já existe um filme com o “titulo”=‘Cidade de Deus’ e o “ano”=2002. Mesmo que o “resumo” seja diferente, ocorre uma violação de PK (não podem existir duas linhas com o mesmo título e ano na tabela *Filme*).

# Definição de Restrições na Linguagem SQL

- PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 6)**

```
INSERT INTO Filme (titulo, ano, resumo, pais, duracao)
VALUES ('A Dama de Ferro',2011,
'An elderly Margaret Thatcher talks to the imagined presence of her recently deceased
husband while scenes from her past life intervene.',
'UK',105);
```

```
INSERT INTO Filme (titulo, ano, resumo, pais, duracao, avaliacao)
VALUES ('Um Sonho de Liberdade',1994,
'Two imprisoned men bond over a number of years, finding solace and eventual
redemption through acts of common decency.',
'US',142,9.3);
```

Query finished in 0.133 second(s). Rows affected: 2

- Registro inseridos com sucesso! Não há violação de PK (não são inseridos filmes com o mesmo “título” e “ano” de outro filme já gravado na tabela) e nem de FK (as “siglas” 'UK' e 'US' estão devidamente cadastradas na tabela Pais).

# Definição de Restrições na Linguagem SQL

- PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 7)**

```
INSERT INTO Filme (titulo, ano, resumo, avaliacao)
VALUES ('Intocáveis',
2011,
'After he becomes a quadriplegic from a paragliding accident, an aristocrat hires a
young man from the projects to be his caregiver.',
8.6);
```

Query finished in 0.133 second(s). Rows affected: 1

- Registro inserido com sucesso! Não há violação de PK (não há filme com o mesmo “título” e “ano” armazenado na tabela) e nem de FK (a “sigla” do país não foi especificada, porém o campo “pais” não foi definido como NOT NULL na tabela *Filme*) .

# Definição de Restrições na Linguagem SQL

- PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 8)**

```
INSERT INTO Filme (titulo, ano, resumo, pais)
VALUES ('O Filho da Noiva',
2001,
'At age 42, Rafael Belvedere is having a crisis.',
'AR');
```

Error while executing SQL query on database 'cinema': FOREIGN KEY constraint failed

- IMPORTANTE:** o comando acima falha porque houve uma tentativa de inserir um filme cuja “sigla” do país ('AR') não está cadastrada na tabela *Pais*. Este é um exemplo de INSERT que **violou uma regra de integridade referencial**. Para poder cadastrar este filme, é **necessário primeiro cadastrar um país de “sigla” 'AR' na tabela *Pais*** e só depois cadastrar o filme em questão na tabela *Filme*. **Isto é feito no exemplo a seguir:**

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO9)**

```
INSERT INTO Pais VALUES('AR','Argentina');  
INSERT INTO Filme (titulo,argentina ano, resumo, pais)  
VALUES ('O Filho da Noiva',  
2001,  
'At age 42, Rafael Belvedere is having a crisis.',  
'AR');  
Query finished in 0.210 second(s). Rows affected: 2
```

- Agora sim, a operação é bem-sucedida!

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 10)**

```
INSERT INTO Filme  
VALUES ('Noel: Poeta da Vila',  
2006,  
'The biopic of Noel Rosa, one of the best brazilian poets and composers.',  
'BR',  
99,  
6.8);
```

Query finished in 0.111 second(s). Rows affected: 1

- Operação bem-sucedida! Não há violação de PK, NOT NULL ou FK.

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 11)**

```
INSERT INTO Filme  
VALUES ('Noel: Poeta da Vila',  
2006,  
'The biopic of Noel Rosa, one of the best brazilian poets and composers.',  
'BR',  
99,  
6.8);
```

Query finished in 0.111 second(s). Rows affected: 1

- Operação bem-sucedida! Não há violação de PK, NOT NULL ou FK.



# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 12)**

```
UPDATE Filme SET pais='XX' WHERE titulo='O Filho da Noiva';
```

Error while executing SQL query on database 'cinema': FOREIGN KEY constraint failed

- Violação de FK! Houve uma tentativa de mudar o “país” do filme 'O Filho da Noiva' de 'AR' para 'XX'. Como a “sigla” 'XX' não está cadastrada na tabela *Pais*, o SGBD não permitiu a atualização.

# Definição de Restrições na Linguagem SQL

- **PASSO 6 – Criando e Preenchendo a Tabela *Filme* (EXEMPLO 12)**
- Vamos verificar o conteúdo corrente das tabelas *Filme* e *Pais*.

```
SELECT * FROM Filme;
```

	titulo	ano	resumo	pais	duracao	avaliacao
1	Cidade de Deus	2002	Two boys growing up in a violent neighborhood of Rio de Janeiro take differen...	NULL	NULL	NULL
2	Um Sonho de Liberdade	1994	Two imprisoned men bond over a number of years, finding solace and eventu...	US	142	9.3
3	Intocáveis	2011	After he becomes a quadriplegic from a paragliding accident, an aristocrat hir...	NULL	NULL	8.6
4	O Filho da Noiva	2001	At age 42, Rafael Belvedere is having a crisis.	AR	NULL	NULL
5	Noel: Poeta da Vila	2006	The biopic of Noel Rosa, one of the best brazilian poets and composers.	BR	99	6.8
6	A Dama de Ferro	2011	An elderly Margaret Thatcher talks to the imagined presence of her recently ...	UK	105	NULL

```
SELECT * FROM Pais;
```

	sigla	nome
1	BR	Brasil
2	US	Estados Unidos
3	UK	Reino Unido
4	FR	França
5	AR	Argentina

# Definição de Restrições na Linguagem SQL

- **PASSO 7 – Excluindo e Atualizando Registros (EXEMPLO 1)**
- Analisaremos agora como as restrições afetam o comando DELETE. Execute o comando abaixo e veja o resultado:

```
DELETE FROM Pais WHERE sigla='US';
```

Error while executing SQL query on database 'cinema': FOREIGN KEY constraint failed

- Este comando falha, pois tentamos apagar uma linha da tabela *Pais* (neste caso, pais de “sigla”=‘US’) que **possui um registro vinculado** na tabela *Filme*. Em outras palavras: existe um filme na tabela *Filme* cujo valor da coluna “pais” é igual a ‘US’ (o filme em questão é “Um Sonho de Liberdade”). Conforme sabemos, a coluna “pais” de *Filme* é uma chave estrangeira que referencia a coluna “sigla” da tabela *Pais*. Quando *Filme* foi criada, esta regra de integridade referencial foi definida através da especificação FOREIGN KEY (pais) REFERENCES Pais(sigla) no comando CREATE TABLE.

# Definição de Restrições na Linguagem SQL

- **PASSO 7 – Excluindo e Atualizando Registros (EXEMPLO 2)**
- Analisaremos agora como as restrições afetam o comando DELETE. Execute o comando abaixo e veja o resultado:

```
DELETE FROM Pais WHERE sigla='FR';
```

Query finished in 0.080 second(s). Rows affected: 1

- Operação bem-sucedida! O país de “sigla” 'FR' (França) foi excluído da tabela Pais. Isso só foi permitido porque não existe nenhum filme em Filme, cujo valor da coluna “pais” é igual a 'FR' (até existe um filme francês nessa tabela - “Intocáveis”, porém ele foi cadastrado sem país!).

# Definição de Restrições na Linguagem SQL

- **PASSO 7 – Excluindo e Atualizando Registros (EXEMPLO 2)**
- Analisaremos agora como as restrições afetam o comando UPDATE. Execute o comando abaixo e veja o resultado:

```
UPDATE Filme SET pais='FR' WHERE titulo='Intocáveis';
```

Error while executing SQL query on database 'cinema': FOREIGN KEY constraint failed

- Violação de FK! Tentamos modificar o “pais” do filme 'Intocáveis' de NULL para 'FR'. Porém, a “sigla” 'FR' não consta mais na tabela Pais! Para que essa atualização seja possível, devemos inserir 'FR' novamente em Pais para depois fazer o UPDATE em Filme.

# Definição de Restrições na Linguagem SQL

- **PASSO 7 – Excluindo e Atualizando Registros (EXEMPLO 2)**
- Analisaremos agora como as restrições afetam o comando UPDATE. Execute o comando abaixo e veja o resultado:

```
INSERT INTO Pais VALUES('FR','França');  
UPDATE Filme SET pais='FR' WHERE titulo='Intocáveis';
```

Query finished in 0.382 second(s). Rows affected: 2

- Funcionou!

# Definição de Restrições na Linguagem SQL

- **PASSO 8 – Criando as Demais Tabela**
- O script SQL abaixo realiza a definição das demais tabelas de acordo com a ordem de criação que foi definida no PASSO 3 (*Genero*, *Artista*, *FilmeGenero* e *FilmeElenco*) e com as restrições definidas nas regras de negócio do slide 4.

```
CREATE TABLE Genero (  
  nome VARCHAR(15) NOT NULL,  
  PRIMARY KEY (nome)  
);
```

```
CREATE TABLE Artista (  
  nome VARCHAR(60) NOT NULL,  
  sexo CHAR(1) NOT NULL,  
  PRIMARY KEY (nome)  
  CHECK (sexo IN ('F', 'M'))  
);
```

# Definição de Restrições na Linguagem SQL

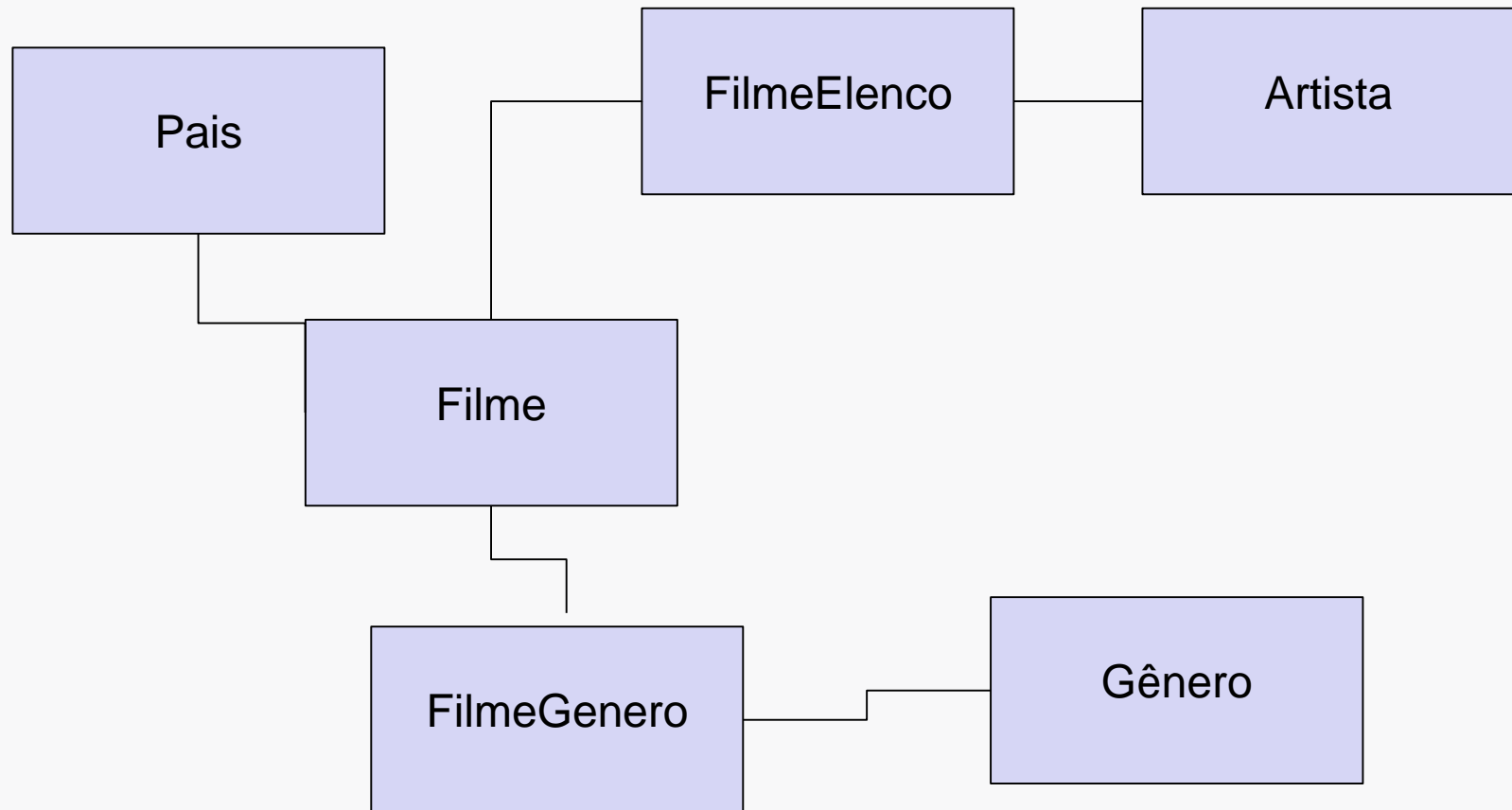
- PASSO 8 – Criando as Demais Tabela**

```
CREATE TABLE FilmeGenero (  
    titulofilme VARCHAR(100) NOT NULL,  
    ano INT NOT NULL,  
    nomegenero VARCHAR(15) NOT NULL,  
    PRIMARY KEY (titulofilme, ano, nomegenero),  
    FOREIGN KEY (titulofilme, ano) REFERENCES Filme(titulo,ano),  
    FOREIGN KEY (nomegenero) REFERENCES Genero(nome)  
);
```

```
CREATE TABLE FilmeElenco (  
    titulofilme VARCHAR(100) NOT NULL,  
    ano INT NOT NULL,  
    nomeartista VARCHAR(60) NOT NULL,  
    PRIMARY KEY (titulofilme, ano, nomeartista),  
    FOREIGN KEY (titulofilme, ano) REFERENCES Filme(titulo,ano),  
    FOREIGN KEY (nomeartista) REFERENCES Artista(nome)  
);
```



# Diagrama do esquema Filme



# Definição de Restrições na Linguagem SQL

- **PASSO 9 – Inserindo um Filme e Todas as suas Informações**
  - Vamos agora inserir um filme, seus gêneros e elenco. Abaixo as informações do filme a ser incluído:
  - **Título:** “ENCE: O Filme”.
  - Ano: 2024;
  - **Resumo:** “Um grupo de estudantes desvenda os mistérios da Ciência de Dados.”.
  - País: BR
  - **Duração:** 120 minutos
  - Avaliação: 9.9
  - **Elenco:** Karl Pearson, Ronald Fisher, Alan Turing, Ada Lovelace.
  - **Gêneros:** Drama, Ação e Aventura.

# Definição de Restrições na Linguagem SQL

- **PASSO 9 – Inserindo um Filme e Todas as suas Informações**
  - Vamos agora inserir um filme, seus gêneros e elenco. Abaixo as informações do filme a ser incluído:
    - **Título:** “ENCE: O Filme”.
    - **Ano:** 2024;
    - **Resumo:** “Um grupo de estudantes desvenda os mistérios da Ciência de Dados.”.
    - **País:** BR
    - **Duração:** 120 minutos
    - **Avaliação:** 9.9
    - **Elenco:** Karl Pearson, Ronald Fisher, Alan Turing, Ada Lovelace.
    - **Gêneros:** Drama, Ação e Aventura.
  - Tudo tem que ser feito na ordem certa! Inicialmente iremos incluir um registro na tabela *Filme*. Só depois poderemos incluir o seu elenco e os gêneros. Ainda assim, os artistas e gêneros precisarão estar previamente cadastrados nas tabelas *Artista* e *Pais*, respectivamente.

# Definição de Restrições na Linguagem SQL

- **PASSO 9 – Inserindo um Filme e Todas as suas Informações**
  - Como o país do filme é “BR”, que já consta na tabela *Pais*, podemos realizar a inclusão dos dados diretamente em *Filme*.

```
INSERT INTO Filme VALUES('ENCE: O Filme',2023,  
'Um grupo de estudantes desvenda os mistérios da Ciência de Dados','BR',120,9.9);
```

# Definição de Restrições na Linguagem SQL

- **PASSO 9 – Inserindo um Filme e Todas as suas Informações**
  - Para incluir o elenco, precisamos inicialmente cadastrar os atores na tabela *Artista*. Caso isto não seja feito, ocorrerá um erro de chave estrangeira na hora de inserir em *FilmeElenco*.

```
INSERT INTO Artista VALUES('Karl Pearson', 'M');  
INSERT INTO Artista VALUES('Ronald Fisher', 'M');  
INSERT INTO Artista VALUES('Alan Turing', 'M');  
INSERT INTO Artista VALUES('Ada Lovelace', 'F');  
INSERT INTO FilmeElenco  
VALUES('ENCE: O Filme',2023,'Karl Pearson');  
INSERT INTO FilmeElenco  
VALUES('ENCE: O Filme',2023,'Ronald Fisher');  
INSERT INTO FilmeElenco  
VALUES('ENCE: O Filme',2023,'Alan Turing');  
INSERT INTO FilmeElenco  
VALUES('ENCE: O Filme',2023,'Ada Lovelace');
```

# Definição de Restrições na Linguagem SQL

- **PASSO 9 – Inserindo um Filme e Todas as suas Informações**
  - Por fim, cadastraremos os gêneros do filme. Para isso, esses gêneros precisam estar previamente cadastrados em *Genero*.

```
INSERT INTO Genero VALUES('Drama');
```

```
INSERT INTO Genero VALUES('Aventura');
```

```
INSERT INTO Genero VALUES('Ação');
```

```
INSERT INTO FilmeGenero VALUES('ENCE: O Filme',2023,'Drama');
```

```
INSERT INTO FilmeGenero VALUES('ENCE: O Filme',2023,'Aventura');
```

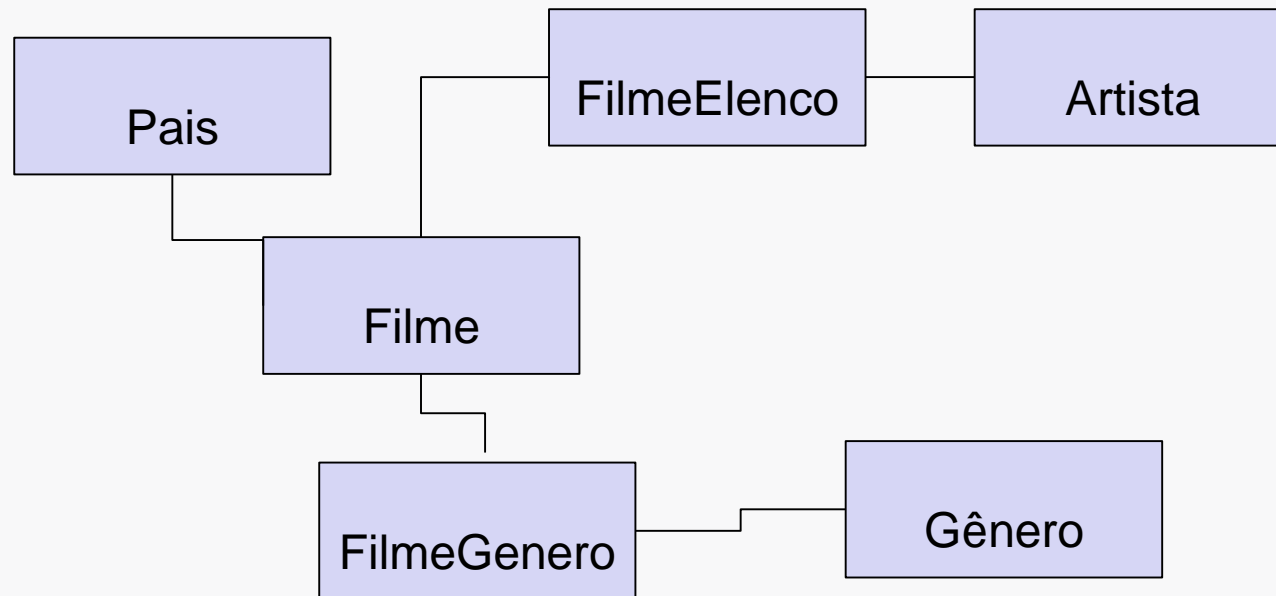
```
INSERT INTO FilmeGenero VALUES('ENCE: O Filme',2023,'Ação');
```

# EXERCÍCIO

- (1) Faça `SELECT` em todas as tabelas do BD. Examine o conteúdo das mesmas e tenha certeza de que você entendeu as razões pelas quais as chaves primárias e estrangeiras foram definidas para cada tabela.

# EXERCÍCIO

- (2) Uma boa forma de verificar se você compreendeu é escrever o que você ouviu e leu, assim você consegue se certificar que o seu entendimento está correto. Assim, responda as perguntas a abaixo:
- Qual ordem de script DELETE que deve ser seguida para excluir um gênero?
- Qual ordem de comando INSERT deve ser seguido para incluir uma nova tupla na tabela Filme contendo no elenco um artista que ainda não consta neste banco de dados?





Obrigado