

Bases de Dados

Módulo 10a: Junções Internas, Externas e outros tipos

Prof. André Bruno de Oliveira

01/08/2024 21:51

Junções de tabelas - Introdução

- As junções são utilizadas em instruções SELECT para recuperar dados de duas tabelas e “juntá-los” com o objetivo de produzir uma saída combinada.
 - Para que as junções sejam realizadas, as tabelas envolvidas devem compartilhar dados em comum através de um ou mais atributos. Na maioria das vezes (**embora não necessariamente**) estes atributos que “conectam” as tabelas estarão vinculados através de uma ligação Chave Primária-Chave Estrangeira.
 - Durante os módulos vistos anteriormente, aprendemos a trabalhar com o Produto Cartesiano e também com os dois tipos de junção oferecidos pela Álgebra Relacional básica: Junção Natural e Junção Theta. Todavia, a Linguagem SQL oferece outros tipos de junção ainda mais flexíveis e poderosas e, por isso, mas usadas na prática. São elas:
 - Junção Interna: INNER JOIN
 - Junções Externas: LEFT JOIN, RIGHT JOIN e FULL JOIN
 - NON-EQUIJOIN
 - SELF-JOIN

Junções de tabelas - Introdução

- As junções são utilizadas em instruções SELECT para recuperar dados de duas tabelas e “juntá-los” com o objetivo de produzir uma saída combinada.
 - Para que as junções sejam realizadas, as tabelas envolvidas devem compartilhar dados em comum através de um ou mais atributos. Na maioria das vezes (**embora não necessariamente**) estes atributos que “conectam” as tabelas estarão vinculados através de uma ligação Chave Primária-Chave Estrangeira.
 - Durante os módulos vistos anteriormente, aprendemos a trabalhar com o Produto Cartesiano e também com os dois tipos de junção oferecidos pela Álgebra Relacional básica: Junção Natural e Junção Theta. Todavia, a Linguagem SQL oferece outros tipos de junção ainda mais flexíveis e poderosas e, por isso, mas usadas na prática. São elas:
 - Junção Interna: INNER JOIN
 - Junções Externas: LEFT JOIN, RIGHT JOIN e FULL JOIN
 - NON-EQUIJOIN
 - SELF-JOIN

Junções de tabelas - Junção Interna x Junção Externa

- Começaremos este módulo abordando os conceitos de junção interna (inner join) e junção externa (outer join) e explicando o uso prático destas junções na SQL. Para tal, faremos uso de exemplos baseados em um BD composto por duas tabelas:

Profissao

| id | nome |
|----|--------------------|
| 1 | Engenheiro |
| 2 | Desenvolvedor |
| 3 | Analista de Dados |
| 4 | Minerador de Dados |
| 5 | Matemático |

Funcionario

| mat | nome | idade | sexo | id_prof |
|------------|-------------|-----------|----------|-------------|
| M01 | George | 58 | M | 5 |
| M02 | Jane | 32 | F | 3 |
| M03 | Aldous | 40 | M | 3 |
| M04 | Thomas | 28 | M | 1 |
| M05 | Mary | 43 | F | NULL |

- Neste BD, a coluna “id_prof” da tabela *Funcionario* possui uma referência para “id” da tabela *Profissao*. Ou seja, o “id da profissão” faz a ligação entre as duas tabelas, possivelmente atuando como chave primária em *Profissao* e chave estrangeira em *Funcionario*. Na instância apresentada, observe que existe uma funcionária sem profissão cadastrada (“M05-Mary”) e também duas profissões que não estão associadas a nenhum funcionário (“2-Desenvolvedor” e “4-Minerador de Dados”).

Junções de tabelas - Junção Interna x Junção Externa

- Com o uso do script abaixo, crie as tabelas e faça a inclusão dos registros.

```
CREATE TABLE Profissao(id INTEGER NOT NULL, nome TEXT NOT NULL, PRIMARY KEY(id));
```

```
CREATE TABLE Funcionario(mat TEXT NOT NULL, nome TEXT NOT NULL, idade  
INTEGER, sexo TEXT, id_prof INTEGER,  
PRIMARY KEY(mat), FOREIGN KEY(id_prof) REFERENCES Profissao(id));
```

```
INSERT INTO Profissao VALUES (1,'Engenheiro'), (2, 'Desenvolvedor'),  
(3, 'Analista de Dados'), (4, 'Minerador de Dados'), (5, 'Matemático');
```

```
INSERT INTO Funcionario VALUES  
( 'M01','George',58,'M',5),  
( 'M02','Jane',32,'F',3),  
( 'M03','Aldous',40,'M',3),  
( 'M04','Thomas',28,'M',1),  
( 'M05','Mary',43,'F',NULL);
```

Junções de tabelas – Junção interna (INNER JOIN)

- A junção interna (INNER JOIN) combina uma linha de uma **tabela A** com uma linha de uma **tabela B** apenas quando há coincidência nos valores das colunas especificadas na condição de junção. O INNER JOIN neste exemplo é utilizado para obter a lista de todos os funcionários e os nomes de suas profissões. **Veja que a há atributos com as mesmas características entre Funcionário e Profissão.**

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|----|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |

Junções de tabelas – Junção interna (INNER JOIN)

- Para realizar a junção interna entre duas tabelas em SQL, basta seguir sempre a mesma receita, que consiste em:
 - (1) Especificar os nomes das tabelas envolvidas na junção na cláusula FROM. Cada tabela poderá receber um “apelido”, utilizado para simplificar a referência às mesmas. Em nosso exemplo, as *Funcionario* recebeu o apelido “F” e *Profissao* o apelido “P”. Sem o uso do apelido, deveríamos usar os nomes reais das tabelas.
 - (2) Especificar o tipo de junção INNER JOIN entre as duas tabelas.
 - (3) Escrever a palavra-chave ON e depois especificar entre parênteses a condição de junção entre as duas tabelas. Isso significa simplesmente, indicar explicitamente qual é ou quais são o(s) atributo(s) que liga(m) as duas tabelas. No exemplo apresentado, o atributo em comum é o “id da profissão”, que se chama “id” em *Profissao* e possui o nome “id_prof” em *Funcionario*.

Ex.: FROM *Funcionario* F **INNER JOIN** *Profissao* P ON (F.id_prof = P.id)

Junções de tabelas – Junção interna (INNER JOIN)

- Vejamos alguns outros detalhes que envolvem a consulta:
 - A funcionária “M05-Mary” não aparece no resultado final, pois ela não possui um código de profissão associado (está com o valor NULL armazenado na coluna “id_prof”).

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|----|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |

Junções de tabelas – Junção interna (INNER JOIN)

- Vejamos alguns outros detalhes que envolvem a consulta:
 - Na primeira linha do comando SELECT, os campos selecionados foram precedidos pelo apelido de cada tabela, para indicar a que tabela pertence cada coluna (tabela “F” ou tabela “P”).
 - Os apelidos são obrigatórios apenas para remover ambiguidade que ocorre quando deseja-se projetar alguma coluna de uma tabela cujo nome seja igual ao de uma coluna da outra tabela. Neste SELECT apresentado, trata-se da coluna “nome”, que existe nas duas tabelas.
 - Veja que o atributo nome contido em ambas as tabelas possuem significados distintos: i)f.nome – é nome do funcionário; ii) p.nome - é nome da profissão.
 - O uso de apelidos para as tabelas em todos os campos (sejam ambíguos ou não) é considerado uma boa prática de programação.

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

Junções de tabelas – Junção interna (INNER JOIN)

- Vejamos alguns outros detalhes que envolvem a consulta:
 - Tanto faz especificar ON (F.id_prof = P.id) ou ON (P.id = F.id_prof): de ambas as formas a junção interna será corretamente implementada.

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

ou

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (P.id= F.id_prof)
```

```
ORDER BY mat;
```

Junções de tabelas – Junção interna (INNER JOIN)

- Vejamos alguns outros detalhes que envolvem a consulta:
 - O ORDER BY do exemplo não é requerido para viabilizar a junção (ORDER BY não tem nada a ver com JOIN!!!!). Ele foi utilizado apenas para ordenar os resultados pela matrícula, algo que será mantido em todos os demais exemplos desta seção.

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|----|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |

Junções de tabelas – Junção interna (INNER JOIN)

- Vejamos alguns outros detalhes que envolvem a consulta:
 - A partir do exemplo apresentado, é possível concluir que o INNER JOIN possui o mesmo propósito do NATURAL JOIN, oferecendo porém a vantagem de não exigir que a coluna de ligação possua o mesmo nome nas duas tabelas.

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

Profissao

| id | nome |
|----|--------------------|
| 1 | Engenheiro |
| 2 | Desenvolvedor |
| 3 | Analista de Dados |
| 4 | Minerador de Dados |
| 5 | Matemático |

Funcionario

| mat | nome | idade | sexo | id_prof |
|------------|-------------|-----------|----------|-------------|
| M01 | George | 58 | M | 5 |
| M02 | Jane | 32 | F | 3 |
| M03 | Aldous | 40 | M | 3 |
| M04 | Thomas | 28 | M | 1 |
| M05 | Mary | 43 | F | NULL |

Junções de tabelas – Junção interna (INNER JOIN)

- Vejamos alguns outros detalhes que envolvem a consulta:
 - INNER JOIN é muito mais utilizado na prática, pois é mais flexível.
 - Veja que neste exemplo a operação de **NATURAL JOIN** entre *Funcionario* e *Profissao* inclusive falharia, produzindo um conjunto vazio como resultado, uma vez que a coluna que possui nome coincidente em ambas as tabelas é “nome”, que na verdade não representa um atributo de ligação entre as tabelas (em Funcionario “nome” é o nome do funcionário e em Profissao é o nome da profissão).

```
SELECT F.*, P.*
```

```
FROM Funcionario F INNER JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

Profissao

| id | nome |
|----|--------------------|
| 1 | Engenheiro |
| 2 | Desenvolvedor |
| 3 | Analista de Dados |
| 4 | Minerador de Dados |
| 5 | Matemático |

Funcionario

| mat | nome | idade | sexo | id_prof |
|-----|--------|-------|------|---------|
| M01 | George | 58 | M | 5 |
| M02 | Jane | 32 | F | 3 |
| M03 | Aldous | 40 | M | 3 |
| M04 | Thomas | 28 | M | 1 |
| M05 | Mary | 43 | F | NULL |

Junções de tabelas – Junção interna alternativa

- O exemplo a seguir apresenta uma sintaxe alternativa para o INNER JOIN.
 - O INNER JOIN é especificado como uma Junção Theta. A indicação das tabelas *Funcionario* e *Profissao* estão separadas por vírgula, o que força o produto cartesiano entre ambas as tabelas.
 - A condição $F.id_prof = P.id$ na cláusula WHERE faz com que apenas as linhas onde estas colunas possuam valores coincidentes sejam levadas para o resultado final. Esta sintaxe não é muito utilizada nos dias atuais, por isso o ideal é evitá-la.

```
SELECT F.*, P.*
```

```
FROM Funcionario F, Profissao P
```

```
WHERE F.id_prof = P.id
```

```
ORDER BY mat;
```

Junções de tabelas – INNER JOIN resumo

- O INNER JOIN possui o mesmo princípio de funcionamento do NATURAL JOIN.
- O INNER JOIN é mais flexível do que o NATURAL JOIN , pois não exige que o atributo de ligação possua o mesmo nome nas duas tabelas.
- O INNER JOIN combina duas linhas apenas se elas “casarem”. Ou seja, qualquer linha de **A** que não case com nenhuma linha de **B** e qualquer linha de **B** que não case com uma linha de **A** ficará de fora do resultado final.

Junções de tabelas – Junção externa (OUTER JOIN)

- As junções externas conectam linhas de duas tabelas de uma forma mais inclusiva:
 - As junções externas produzem os mesmos resultados do INNER JOIN acrescidos de linhas “não-casadas” de uma ou ambas as tabelas. Os tópicos seguintes descrevem os três tipos de junção externa: LEFT JOIN (a mais usada), RIGHT JOIN e FULL JOIN.

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- A operação de LEFT JOIN (ou LEFT OUTER JOIN) retorna todas as linhas da tabela especificada à esquerda da junção, mesmo que não exista casamento (valor equivalente) na tabela à direita. Para que o conceito fique claro, vamos retornar para o nosso exemplo envolvendo as tabelas *Funcionario* e *Profissao*.

```
SELECT F.*, P.*
```

```
FROM Funcionario F LEFT JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- Repare nesta query que a funcionária ‘M05-Mary’ não possui uma profissão cadastrada (o campo “id_prof” possui valor NULL para esse registro em Funcionario). Se desejarmos produzir um relatório contendo todos os funcionários com seus respectivos nomes de profissão, incluindo também os funcionários sem profissão cadastrada, é preciso fazer uso do LEFT JOIN como pode-se ver nesta query.

```
SELECT F.*, P.*
```

```
FROM Funcionario F LEFT JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- Para realizar a LEFT JOIN entre duas tabelas em SQL, a receita é quase idêntica à utilizada para implementar um INNER JOIN. O único detalhe a ser observado é que, na cláusula FROM, a tabela a qual desejamos todas as linhas no resultado final deve ser especificada à esquerda na definição da junção.

```
SELECT F.*, P.*
```

```
FROM Funcionario F LEFT JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- Por que usamos o LEFT JOIN ?
- Quando desejamos que os campos da tabela à esquerda sejam listados, mesmo que não haja casamento com a tabela à direita.
- Nesta query, estão todos os funcionários no resultado final, independentemente de terem profissão ou não, por isso a tabela *Funcionario* foi especificada à esquerda da cláusula LEFT JOIN, enquanto *Profissao* ficou à direita:

```
SELECT F.*, P.*
```

```
FROM Funcionario F LEFT JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- Com relação a condição de junção (ON) a ordem não faz diferença, ou seja, tanto faz especificar ON (F.id_prof = P.id) ou ON (P.id = F.id_prof).

```
SELECT F.*, P.*
```

```
FROM Funcionario F LEFT JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

Ou

```
SELECT F.*, P.*
```

```
FROM Funcionario F LEFT JOIN Profissao P ON (P.id = F.id_prof)
```

```
ORDER BY mat;
```

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- Para listar todos os cargos e funcionários, mesmo os cargos que não tenham um funcionário associado, é preciso especificar *Profissao* à esquerda.

```
SELECT F.*, P.*
```

```
FROM Profissao P LEFT JOIN Funcionario F ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|------|--------|-------|------|---------|----|--------------------|
| 1 | NULL | NULL | NULL | NULL | NULL | 2 | Desenvolvedor |
| 2 | NULL | NULL | NULL | NULL | NULL | 4 | Minerador de Dados |
| 3 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 4 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 5 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 6 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |

Junções de tabelas – Junção externa (OUTER JOIN)

- LEFT JOIN

- O LEFT JOIN produz os mesmos dados do INNER JOIN, acrescidos das linhas da tabela A (tabela à esquerda) que não casam com nenhuma linha de B (tabela à direita).
- O LEFT JOIN é o tipo de junção externa mais utilizado.

```
SELECT F.*, P.*
FROM Profissao P LEFT JOIN Funcionario F
ON (F.id_prof = P.id) ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|------|--------|-------|------|---------|----|--------------------|
| 1 | NULL | NULL | NULL | NULL | NULL | 2 | Desenvolvedor |
| 2 | NULL | NULL | NULL | NULL | NULL | 4 | Minerador de Dados |
| 3 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 4 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 5 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 6 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |

```
SELECT F.*, P.*
FROM Funcionario F LEFT JOIN Profissao P ON
(F.id_prof = P.id) ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (OUTER JOIN)

- RIGHT JOIN
 - A junção do tipo **RIGHT JOIN** (ou **RIGHT OUTER JOIN**) retorna todas as linhas da tabela à direita, mesmo que não exista casamento (valor equivalente) na tabela à esquerda. Desta forma, para produzir um relatório com todos os funcionários com seus respectivos nomes de profissão, incluindo os funcionários sem profissão cadastrada, seria necessário montar o comando **SELECT** da forma indicada abaixo.

```
SELECT F.*, P.*
```

```
FROM Profissao P RIGHT JOIN Funcionario F ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (RIGHT JOIN)

- RIGHT JOIN
 - Como o LEFT JOIN e o RIGHT JOIN funcionam da mesma maneira (a única diferença é que no primeiro caso iremos especificar a tabela a qual desejamos todas as linhas no resultado à esquerda e no segundo caso ela ficará à direita), na prática o LEFT JOIN acabou se tornando mais adotado pelos programadores.

```
SELECT F.*, P.*
```

```
FROM Profissao P RIGHT JOIN Funcionario F ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|-----|--------|-------|------|---------|------|-------------------|
| 1 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 2 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 3 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 4 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 5 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Resumo (RIGHT JOIN)

- RIGHT JOIN
 - O **RIGHT JOIN** produz os mesmos dados do **INNER JOIN**, acrescidos das linhas da tabela **F** (tabela à direita) que não casam com nenhuma linha de **P** (tabela à esquerda).

```
SELECT P.*,F.*
```

```
FROM Profissao P RIGHT JOIN Funcionario F ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | id | nome | mat | nome:1 | idade | sexo | id_prof |
|---|------|-------------------|-----|--------|-------|------|---------|
| 1 | 5 | Matemático | M01 | George | 58 | M | 5 |
| 2 | 3 | Analista de Dados | M02 | Jane | 32 | F | 3 |
| 3 | 3 | Analista de Dados | M03 | Aldous | 40 | M | 3 |
| 4 | 1 | Engenheiro | M04 | Thomas | 28 | M | 1 |
| 5 | NULL | NULL | M05 | Mary | 43 | F | NULL |

Junções de tabelas – Junção externa (FULL JOIN)

- FULL JOIN

- A junção do tipo **FULL JOIN** (ou **FULL OUTER JOIN**) junta em uma única operação os resultados do **LEFT JOIN** e do **RIGHT JOIN**.

```
SELECT F.*, P.*
```

```
FROM Funcionario F FULL JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|------|--------|-------|------|---------|------|--------------------|
| 1 | NULL | NULL | NULL | NULL | NULL | 2 | Desenvolvedor |
| 2 | NULL | NULL | NULL | NULL | NULL | 4 | Minerador de Dados |
| 3 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 4 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 5 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 6 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 7 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (FULL JOIN)

- FULL JOIN

- Veja que o FULL JOIN retorna todas as linhas da tabela à esquerda e todas as linhas da tabela à direita devidamente combinadas, de acordo com a especificação da condição de junção.
- O FULL JOIN retorna as linhas da tabela à esquerda que não casam com qualquer linha da tabela à direita e mesmo assim a linha é exibida no resultado final.
- O FULL JOIN retorna as linhas da tabela à direita que não casam com qualquer linha da tabela à esquerda e mesmo assim a linha é exibida no resultado final.

SELECT P.*,F.*

FROM Profissao P RIGHT JOIN Funcionario F ON (F.id_prof = P.id)

ORDER BY mat;

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|------|--------|-------|------|---------|------|--------------------|
| 1 | NULL | NULL | NULL | NULL | NULL | 2 | Desenvolvedor |
| 2 | NULL | NULL | NULL | NULL | NULL | 4 | Minerador de Dados |
| 3 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 4 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 5 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 6 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 7 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (FULL JOIN)

- FULL JOIN

- Neste exemplo, veja que tanto a funcionária sem profissão (linha 5 do resultado), como as profissões que não estão associadas a nenhum funcionário (linhas 1 e 2 do resultado) foram listadas pelo SELECT.

```
SELECT F.*, P.*
```

```
FROM Funcionario F FULL JOIN Profissao P ON (F.id_prof = P.id)
```

```
ORDER BY mat;
```

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|---|------|--------|-------|------|---------|------|--------------------|
| 1 | NULL | NULL | NULL | NULL | NULL | 2 | Desenvolvedor |
| 2 | NULL | NULL | NULL | NULL | NULL | 4 | Minerador de Dados |
| 3 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 4 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 5 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 6 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 7 | M05 | Mary | 43 | F | NULL | NULL | NULL |

Junções de tabelas – Junção externa (FULL JOIN)

- FULL JOIN
 - O FULL JOIN produz os mesmos dados do INNER JOIN, acrescidos das linhas da tabela A (tabela à esquerda) que não casam com nenhuma linha de B (tabela à direita) e também das linhas de B que não casam com nenhuma linha de A.
 - O FULL JOIN não é um operador muito comum, há versões de banco de dados que não tem a sua implementação disponível para uso. Ele é suportado por todos os SGBD profissionais e também pelas linguagens Python (biblioteca pandas), SAS e diversas outras, pois ele é útil em muitas situações práticas.
 - Pode sugerir alternativas ao FULL JOIN como: i) Realizar a união (UNION) de dois LEFT JOIN, ou; ii) dois RIGHT JOIN; ou iii) Unir LEFT JOIN com RIGHT JOIN.

Junções de tabelas – Junção externa (NON-EQUIJOIN)

- NON-EQUIJOIN
 - Todas as junções apresentadas na seção anterior são do tipo equijoin. Isto quer dizer que as junções foram implementadas utilizando atributos comuns entre as duas tabelas e as linhas foram combinadas sempre que as colunas possuíam valores iguais para tais atributos..
 - Entretanto, também é possível combinar linhas de duas tabelas com base em algum outro tipo de critério diferente da igualdade.
Ex.: maior, menor ou igual. A junção com este tipo de operadores como critério de combinação é denominada non-equijoin.

Junções de tabelas – Junção externa (NON-EQUIJOIN)

- NON-EQUIJOIN

- Esta seção introduz o conceito de non-equijoin a partir do exemplo das duas tabelas baixo:

i) *T_Aluno*, que armazena os nomes e notas de 3 alunos em uma determinada avaliação; e

ii) *T_Conceito*, que contém informação sobre o critério adotado para atribuir conceitos para os alunos, por exemplo, quem tira nota entre 90 e 100 obtém conceito ‘A’.

T_aluno

| | nome | nota |
|---|--------|------|
| 1 | Rakesh | 95 |
| 2 | Dayse | 50 |
| 3 | Keith | 72 |

T_conceito

| | conceito | nota_min | nota_max |
|---|----------|----------|----------|
| 1 | A | 90 | 100 |
| 2 | B | 80 | 89 |
| 3 | C | 70 | 79 |
| 4 | D | 50 | 69 |
| 5 | E | 0 | 49 |

Junções de tabelas – Junção externa (NON-EQUIJOIN)

- NON-EQUIJOIN

- As tabelas podem ser criadas e populadas com o uso do script a seguir:

```
CREATE TABLE T_Aluno(nome TEXT PRIMARY KEY, nota INTEGER);
```

```
CREATE TABLE T_Conceito(conceito TEXT PRIMARY KEY, nota_min INTEGER,  
nota_max INTEGER);
```

```
INSERT INTO T_Aluno VALUES ('Rakesh', 95), ('Dayse', 50), ('Keith', 72);
```

```
INSERT INTO T_Conceito VALUES ('A', 90, 100),
```

```
('B', 80, 89), ('C', 70, 79),
```

```
('D', 50, 69), ('E', 0, 49);
```

Junções de tabelas – Junção externa (NON-EQUIJOIN)

- NON-EQUIJOIN

- O Query a seguir apresenta um SQL que realiza um non-equijoin entre tabelas para listar o nome, a nota e o conceito de cada aluno.

```
SELECT a.nome, a.nota, c.conceito
```

```
FROM T_Aluno a JOIN T_Conceito c
```

```
ON (a.nota BETWEEN c.nota_min AND c.nota_max);
```

| | nome | nota | conceito |
|---|--------|------|----------|
| 1 | Rakesh | 95 | A |
| 2 | Dayse | 50 | D |
| 3 | Keith | 72 | C |

Junções de tabelas – Junção externa (NON-EQUIJOIN)

- NON-EQUIJOIN

- Observe a sintaxe deste tipo de junção. Na segunda linha, a palavra-chave JOIN foi utilizada para indicar que desejávamos realizar a junção entre **T_Aluno** (apelidada de **a**) e **T_Conceito** (apelidada de **c**). Na linha 3, em vez de implementar um equijoin (condição de igualdade), indicamos que uma linha de **a** deveria ser combinada com uma linha de **c** apenas quando o valor da “nota” (coluna de **a**) estivesse compreendido entre os valores de “nota_min” e “nota_max” (colunas de **c**).

```
SELECT a.nome, a.nota, c.conceito
```

```
FROM T_Aluno a JOIN T_Conceito c
```

```
ON (a.nota BETWEEN c.nota_min AND c.nota_max);
```

| | nome | nota | conceito |
|---|--------|------|----------|
| 1 | Rakesh | 95 | A |
| 2 | Dayse | 50 | D |
| 3 | Keith | 72 | C |

Junções de tabelas – Junção externa (NON-EQUIJOIN)

- NON-EQUIJOIN

- Em essência, non-equijoin é apenas um nome mais “comercial” para a nossa conhecida Junção Theta e que, por essa razão, acabou por ser adotado pelos SGBD. Trata-se da operação combina linhas de duas tabelas com base em uma condição arbitrária, diferente da igualdade. É até mesmo possível implementar um non-equijoin utilizando-se a sintaxe da Junção Theta, conforme mostrado abaixo.

```
SELECT a.nome, a.nota, c.conceito
```

```
FROM T_Aluno a, T_Conceito c
```

```
WHERE a.nota BETWEEN c.nota_min AND c.nota_max;
```

| | nome | nota | conceito |
|---|--------|------|----------|
| 1 | Rakesh | 95 | A |
| 2 | Dayse | 50 | D |
| 3 | Keith | 72 | C |

Junções de tabelas – Resumo (NON-EQUIJOIN)

- NON-EQUIJOIN
 - A junção interna (INNER JOIN) e as junções externas (LEFT JOIN, RIGHT JOIN e FULL JOIN) são equijoins, pois utilizam a igualdade como critério de comparação.
 - Um **non-equijoin** é um tipo de junção baseado em qualquer critério de comparação que **não seja a igualdade**.
 - Um non-equijoin pode ser implementado com o uso da palavra-chave JOIN.
 - Um non-equijoin também pode ser implementado como uma Junção Theta. Tratam-se de conceitos quase que equivalentes (só não podemos dizer que são equivalentes porque a Junção Theta também permite a elaboração de equijoins, como foi mostrado nos tópicos anteriores).

Junções de tabelas – Junção Externa (SELF-JOIN)

- SELF-JOIN
 - Self-join é o nome dado para um tipo de junção onde as linhas de uma dada tabela são combinadas com outras linhas dela própria, tendo por base algum critério que compara duas diferentes colunas. Para que o conceito fique claro, considere a tabela *T_Filme*, cuja instância é apresentada abaixo.

T_Filme

| | id | nome | id_pred |
|---|----|------------------------|-------------|
| 1 | 1 | Guerra nas Estrelas | <i>NULL</i> |
| 2 | 2 | O Império Contra-Ataca | 1 |
| 3 | 3 | O Retorno de Jedi | 2 |
| 4 | 4 | Avatar | <i>NULL</i> |

Junções de tabelas – Resumo (Self-Join)

- O script para criar e popular a tabela *T_Filme* é apresentado abaixo. Veja que *T_Aluno* possui uma chave estrangeira no campo “id_pred”, que referencia o campo “id”, que é a chave primária da própria tabela.

```
CREATE TABLE T_Filme
```

```
(id INT, nome TEXT, id_pred INT,
```

```
PRIMARY KEY(id),
```

```
FOREIGN KEY (id_pred) REFERENCES T_Filme(id));
```

```
INSERT INTO T_Filme VALUES(1,'Guerra nas Estrelas', NULL);
```

```
INSERT INTO T_Filme VALUES(2,'O Império Contra-Ataca', 1);
```

```
INSERT INTO T_Filme VALUES(3,'O Retorno de Jedi', 2);
```

```
INSERT INTO T_Filme VALUES(4,'Avatar', NULL);
```

Junções de tabelas – Resumo (Self-Join)

- SELF-JOIN

- A tabela *T_Filme* possui três campos: “id” (id do filme), “nome” (nome do filme) e “id_pred” (id do filme predecessor, caso faça parte de alguma sequência).
- Veja que os filmes “Guerra nas Estrelas” e “Avatar” possuem o campo “id_pred” com valor NULL, pois eles não representam a sequência de nenhum outro filme.
- Já “O Império Contra-Ataca” possui “id_pred” com o valor 1, para indicar que ele é um sucessor de “Guerra nas Estrelas” (filme com id=1). De maneira análoga, “O Retorno de Jedi” possui “id_pred” com valor 2, para indicar que se trata de uma sequência de “O Império Contra-Ataca” (filme com id=2).

T_Filme

| | id | nome | id_pred |
|---|----|------------------------|---------|
| 1 | 1 | Guerra nas Estrelas | NULL |
| 2 | 2 | O Império Contra-Ataca | 1 |
| 3 | 3 | O Retorno de Jedi | 2 |
| 4 | 4 | Avatar | NULL |

Junções de tabelas – Resumo (Self-Join)

- SELF-JOIN

- A seguir apresenta-se um exemplo de self-join capaz de produzir uma listagem contendo nome de cada filme e o nome de seu predecessor, caso o mesmo exista.

```
SELECT a.nome AS nome_filme, b.nome AS nome_predecessor  
FROM T_Filme a LEFT JOIN T_Filme b ON (a.id_pred = b.id);
```

| | nome_filme | nome_predecessor |
|---|------------------------|------------------------|
| 1 | Guerra nas Estrelas | <i>NULL</i> |
| 2 | O Império Contra-Ataca | Guerra nas Estrelas |
| 3 | O Retorno de Jedi | O Império Contra-Ataca |
| 4 | Avatar | <i>NULL</i> |

Junções de tabelas – Resumo (Self-Join)

- SELF-JOIN

- Do exemplo apresentado, é possível notar que para implementar um self-join torna-se preciso:
 - (1) Especificar a tabela duas vezes na cláusula FROM, atribuindo apelidos diferentes para cada especificação (em nosso exemplo os apelidos foram a e b).
 - (2) Definir um critério de JOIN. No exemplo apresentada, utilizou-se o LEFT JOIN para que fosse possível listar tantos os filmes que possuem predecessor, como aqueles que não possuem.
 - (3) Especificar cuidadosamente a condição de junção entre a e b. Em nosso exemplo, utilizou-se a condição ON (a.id_pred = b.id) para combinar o id do predecessor de a com o id do filme b.

```
SELECT a.nome AS nome_filme, b.nome AS nome_predecessor
FROM T_Filme a LEFT JOIN T_Filme b ON (a.id_pred = b.id);
```

Junções de tabelas – Produto cartesiano

- CROSS JOIN

- Na linguagem SQL, CROSS JOIN é a mesma coisa que produto cartesiano. Ou seja, é uma operação que produz um conjunto de resultados onde cada linha de uma tabela é combinada com cada linha da outra tabela, sem que haja qualquer tipo de critério para realizar a combinação.
- Desta forma, o CROSS JOIN entre Funcionario e Profissao produz um conjunto de resultados contendo $5 \times 5 = 25$ linhas.

SELECT F.*, P.*

FROM Funcionario F

CROSS JOIN Profissao P;

| | mat | nome | idade | sexo | id_prof | id | nome:1 |
|----|-----|--------|-------|------|---------|----|--------------------|
| 1 | M01 | George | 58 | M | 5 | 1 | Engenheiro |
| 2 | M01 | George | 58 | M | 5 | 2 | Desenvolvedor |
| 3 | M01 | George | 58 | M | 5 | 3 | Analista de Dados |
| 4 | M01 | George | 58 | M | 5 | 4 | Minerador de Dados |
| 5 | M01 | George | 58 | M | 5 | 5 | Matemático |
| 6 | M02 | Jane | 32 | F | 3 | 1 | Engenheiro |
| 7 | M02 | Jane | 32 | F | 3 | 2 | Desenvolvedor |
| 8 | M02 | Jane | 32 | F | 3 | 3 | Analista de Dados |
| 9 | M02 | Jane | 32 | F | 3 | 4 | Minerador de Dados |
| 10 | M02 | Jane | 32 | F | 3 | 5 | Matemático |
| 11 | M03 | Aldous | 40 | M | 3 | 1 | Engenheiro |
| 12 | M03 | Aldous | 40 | M | 3 | 2 | Desenvolvedor |
| 13 | M03 | Aldous | 40 | M | 3 | 3 | Analista de Dados |
| 14 | M03 | Aldous | 40 | M | 3 | 4 | Minerador de Dados |
| 15 | M03 | Aldous | 40 | M | 3 | 5 | Matemático |
| 16 | M04 | Thomas | 28 | M | 1 | 1 | Engenheiro |
| 17 | M04 | Thomas | 28 | M | 1 | 2 | Desenvolvedor |
| 18 | M04 | Thomas | 28 | M | 1 | 3 | Analista de Dados |
| 19 | M04 | Thomas | 28 | M | 1 | 4 | Minerador de Dados |
| 20 | M04 | Thomas | 28 | M | 1 | 5 | Matemático |
| 21 | M05 | Mary | 43 | F | NULL | 1 | Engenheiro |
| 22 | M05 | Mary | 43 | F | NULL | 2 | Desenvolvedor |
| 23 | M05 | Mary | 43 | F | NULL | 3 | Analista de Dados |
| 24 | M05 | Mary | 43 | F | NULL | 4 | Minerador de Dados |
| 25 | M05 | Mary | 43 | F | NULL | 5 | Matemático |

Junções de tabelas – Produto cartesiano

- O script abaixo já foi usado nos slides anteriores, caso seja preciso crie as tabelas, e faça a inclusão dos registros.

```
CREATE TABLE Profissao(id INTEGER NOT NULL, nome TEXT NOT NULL, PRIMARY KEY(id));
```

```
CREATE TABLE Funcionario(mat TEXT NOT NULL, nome TEXT NOT NULL, idade  
INTEGER, sexo TEXT, id_prof INTEGER,  
PRIMARY KEY(mat), FOREIGN KEY(id_prof) REFERENCES Profissao(id));
```

```
INSERT INTO Profissao VALUES (1,'Engenheiro'), (2, 'Desenvolvedor'),  
(3, 'Analista de Dados'), (4, 'Minerador de Dados'), (5, 'Matemático');
```

```
INSERT INTO Funcionario VALUES  
( 'M01','George',58,'M',5),  
( 'M02','Jane',32,'F',3),  
( 'M03','Aldous',40,'M',3),  
( 'M04','Thomas',28,'M',1),  
( 'M05','Mary',43,'F',NULL);
```

Junções de tabelas – Produto cartesiano

- CROSS JOIN
 - Observações:
 - O SELECT equivale a: SELECT F.*, P.* FROM Funcionario F, Profissao P (syntaxe que havíamos apresentado para o produto cartesiano no módulo 10).
 - Dentro dos aplicativos e sistemas mais convencionais, o produto cartesiano é pouco utilizado, sendo seu uso limitado a situações especiais.
 - Um comando SELECT contendo uma condição de JOIN especificada de forma errada normalmente produzirá como resultado um produto cartesiano entre as duas tabelas envolvidas. Por este motivo, muitos desenvolvedores de sistema associam o termo “produto cartesiano” ao significado “junção errada”. Mas isso não é verdade! O produto cartesiano é uma operação básica da Álgebra Relacional que pode ser necessária em algumas situações especiais.

SELECT F.*, P.*

FROM Funcionario F CROSS JOIN Profissao P;

Junções de tabelas – Junção de Múltiplas Tabelas

- Todos os exemplos apresentados nesta aula envolveram a junção entre duas tabelas. Porém, é possível montar comandos SELECT contendo diversos INNER JOINS, LEFT JOINS, etc. Neste caso, o importante é saber que a operação de junção será sempre feita aos pares (resolvendo duas relações de cada vez), respeitando a ordem em que estes pares são especificados (diversos exemplos serão apresentados nas próximas aulas).
- Apresentaremos alguns exemplos utilizando a base do IMDb (“Cinema.db”). O esquema deste BD e um diagrama que indica a associação entre as tabelas é apresentado a seguir.

Filme(titulo, ano, resumo, pais, duracao, avaliacao)

Pais(sigla, nome)

Artista(nome, sexo)

Genero(nome)

FilmeGenero(titulofilme, ano, nomegenero)

FilmeElenco(titulofilme, ano, nomegenero)

Junções de tabelas – Junção de Múltiplas Tabelas

- Todos os exemplos apresentados nesta aula envolveram a junção entre duas tabelas. Porém, é possível montar comandos SELECT contendo diversos INNER JOINS, LEFT JOINS, etc. Neste caso, o importante é saber que a operação de junção será sempre feita aos pares (resolvendo duas relações de cada vez), respeitando a ordem em que estes pares são especificados (diversos exemplos serão apresentados nas próximas aulas).
- Apresentaremos alguns exemplos utilizando a base do IMDb (“Cinema.db”). O esquema deste BD e um diagrama que indica a associação entre as tabelas.

Filme(titulo, ano, resumo, pais, duracao, avaliacao)

Pais(sigla, nome)

Artista(nome, sexo)

Genero(nome)

FilmeGenero(titulofilme, ano, nomegenero)

FilmeElenco(titulofilme, ano, nomegenero)

Junções de tabelas – Junção de Múltiplas Tabelas

- Nos exemplos apresentados, será considerada a seguinte instância :

Pais

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

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 different ... | NULL | NULL | NULL |
| 2 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and eventual ... | US | 142 | 9.3 |
| 3 | Intocáveis | 2011 | After he becomes a quadriplegic from a paragliding accident, an aristocrat hires a ... | 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 |
| 7 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | BR | 120 | 9.9 |

Artista

| | nome | sexo |
|---|---------------|------|
| 1 | Karl Pearson | M |
| 2 | Ronald Fisher | M |
| 3 | Alan Turing | M |
| 4 | Ada Lovelace | F |

FilmeElenco

| | titulofilme | ano | nomeartista |
|---|---------------|------|---------------|
| 1 | ENCE: O Filme | 2023 | Karl Pearson |
| 2 | ENCE: O Filme | 2023 | Ronald Fisher |
| 3 | ENCE: O Filme | 2023 | Alan Turing |
| 4 | ENCE: O Filme | 2023 | Ada Lovelace |

Genero

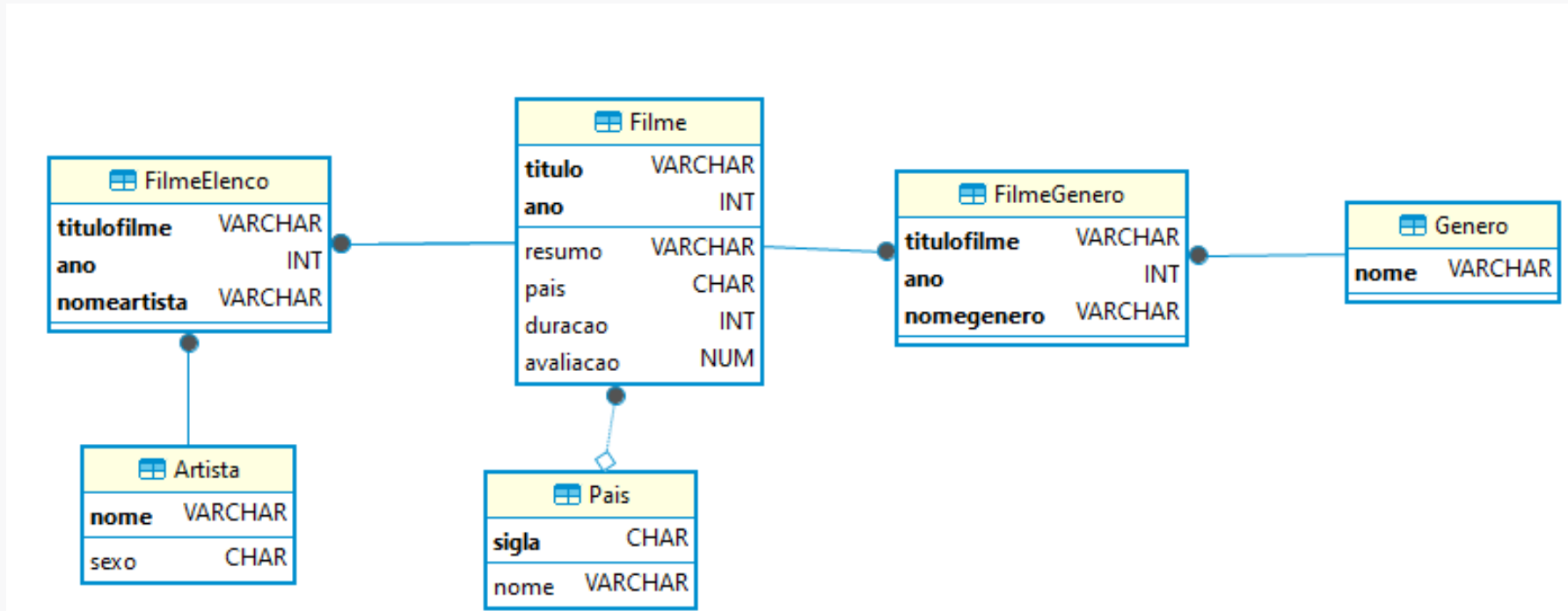
| | nome |
|---|----------|
| 1 | Drama |
| 2 | Aventura |
| 3 | Ação |

FilmeGenero

| | titulofilme | ano | nomegenero |
|---|---------------|------|------------|
| 1 | ENCE: O Filme | 2023 | Drama |
| 2 | ENCE: O Filme | 2023 | Aventura |
| 3 | ENCE: O Filme | 2023 | Ação |

Junções de tabelas – Junção de Múltiplas Tabelas

- Diagrama do banco de dados Cinema



Junções de tabelas – Junção de Múltiplas Tabelas

- A seguir apresentaremos uma série de exemplos de instruções SQL comentadas. O primeiro exemplo mostra o comando SELECT que recupera o título, ano, resumo, sigla do país e nome do país dos filmes:

```
SELECT F.titulo, F.ano, F.resumo, F.pais, P.nome
```

```
FROM Filme F INNER JOIN Pais P ON (F.pais = P.sigla);
```

| | titulo | ano | resumo | pais | nome |
|---|-----------------------|------|---|------|----------------|
| 1 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and eventual ... | US | Estados Unidos |
| 2 | O Filho da Noiva | 2001 | At age 42, Rafael Belvedere is having a crisis. | AR | Argentina |
| 3 | Noel: Poeta da Vila | 2006 | The biopic of Noel Rosa, one of the best brazilian poets and composers. | BR | Brasil |
| 4 | A Dama de Ferro | 2011 | An elderly Margaret Thatcher talks to the imagined presence of her recently ... | UK | Reino Unido |
| 5 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | BR | Brasil |

Junções de tabelas – Junção de Múltiplas Tabelas

- O resultado desta query é obtido através da junção entre as tabelas Filme (que contém o título, ano, resumo e sigla do país) e Pais (que contém a sigla e o nome do país).
- Veja que o filme “Cidade de Deus” não possui país cadastrado, assim não foi levado para o conjunto de resultados, já que utilizamos uma junção do tipo INNER JOIN.

| | titulo | ano | resumo | pais | duracao | avaliacao |
|---|----------------|------|--|------|---------|-----------|
| 1 | Cidade de Deus | 2002 | Two boys growing up in a violent neighborhood of Rio de Janeiro take different ... | NULL | NULL | NULL |

```
SELECT F.titulo, F.ano, F.resumo, F.pais, P.nome
FROM Filme F INNER JOIN Pais P ON (F.pais = P.sigla);
```

| | titulo | ano | resumo | pais | nome |
|---|-----------------------|------|---|------|----------------|
| 1 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and eventual ... | US | Estados Unidos |
| 2 | O Filho da Noiva | 2001 | At age 42, Rafael Belvedere is having a crisis. | AR | Argentina |
| 3 | Noel: Poeta da Vila | 2006 | The biopic of Noel Rosa, one of the best brazilian poets and composers. | BR | Brasil |
| 4 | A Dama de Ferro | 2011 | An elderly Margaret Thatcher talks to the imagined presence of her recently ... | UK | Reino Unido |
| 5 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | BR | Brasil |

Junções de tabelas – Junção de Múltiplas Tabelas

- Para que o filme “Cidade de Deus” seja incluído, é preciso mudar o tipo de junção para LEFT JOIN. Colocando a tabela Filme à esquerda do LEFT JOIN, todas as linhas dessa tabela serão incluídas no resultado independente de possuírem ou não país.

```
SELECT F.titulo, F.ano, F.resumo, F.pais, P.nome
```

```
FROM Filme F INNER JOIN Pais P ON (F.pais = P.sigla);
```

Mudar para

```
SELECT F.titulo, F.ano, F.resumo, F.pais, P.nome
```

```
FROM Filme F LEFT JOIN Pais P ON (F.pais = P.sigla);
```

Junções de tabelas – Junção de Múltiplas Tabelas

- Filme “Cidade de Deus” incluído na seleção. A lógica usada lista as linhas de Filme mesmo que não o critério usado não seja verdade.
- Veja que nas linhas sem casamento entre *Filme* e *Pais* os campos de Pais ficam com NULO (NULL) para indicar que esta informação não pode ser determinada.

```
SELECT F.titulo, F.ano, F.resumo, F.pais, P.nome
```

```
FROM Filme F LEFT JOIN Pais P ON (F.pais = P.sigla);
```

| | titulo | ano | resumo | pais | nome |
|---|-----------------------|------|--|------|----------------|
| 1 | Cidade de Deus | 2002 | Two boys growing up in a violent neighborhood of Rio de Janeiro take different ... | NULL | NULL |
| 2 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and eventual ... | US | Estados Unidos |
| 3 | Intocáveis | 2011 | After he becomes a quadriplegic from a paragliding accident, an aristocrat hires a ... | NULL | NULL |
| 4 | O Filho da Noiva | 2001 | At age 42, Rafael Belvedere is having a crisis. | AR | Argentina |
| 5 | Noel: Poeta da Vila | 2006 | The biopic of Noel Rosa, one of the best brazilian poets and composers. | BR | Brasil |
| 6 | A Dama de Ferro | 2011 | An elderly Margaret Thatcher talks to the imagined presence of her recently ... | UK | Reino Unido |
| 7 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | BR | Brasil |

Junções de tabelas – Junção de Múltiplas Tabelas

- Suponha agora que desejamos listar o título, ano, resumo, nome do país e elenco de todos os filmes com elenco cadastrado. Para obter este resultado, será preciso incluir a tabela FilmeElenco em uma nova junção.
- Veja que os valores de nomeArtista estão listados no resultado. Foi usado PK de *Filme* com a FK de *FilmeElenco*.

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
```

```
FROM Filme F
```

```
INNER JOIN Pais P ON (F.pais = P.sigla)
```

```
INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)
```

| | titulo | ano | resumo | nome_pais | nomeArtista |
|---|---------------|------|--|-----------|---------------|
| 1 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Karl Pearson |
| 2 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Ronald Fisher |
| 3 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Alan Turing |
| 4 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Ada Lovelace |

Junções de tabelas – Junção de Múltiplas Tabelas

- A seguir a explicação sobre como este SELECT é resolvido:
- (1) Conforme mencionado no início desta seção, as junções entre tabelas são sempre feitas aos pares. Sendo assim, primeiro realiza-se o INNER JOIN entre *Filme* e *Pais* gerando um resultado intermediário (relação temporária).

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
```

```
FROM Filme F
```

```
INNER JOIN Pais P ON (F.pais = P.sigla)
```

```
INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)
```

Junções de tabelas – Junção de Múltiplas Tabelas

- O resultado intermediário inclui um conjunto contendo todos os filmes acrescentados do nome do país de cada filme. No resultado dessa operação, o filme “Cidade de Deus” já ficará de fora, pois a operação foi de INNER JOIN e este filme contém NULL no atributo “país”.

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
FROM Filme F
```

```
INNER JOIN Pais P ON (F.pais = P.sigla)
```

```
INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)
```

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 different ... | NULL | NULL | NULL |
| 2 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and eventual ... | US | 142 | 9.3 |
| 3 | Intocáveis | 2011 | After he becomes a quadriplegic from a paragliding accident, an aristocrat hires a ... | 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 |
| 7 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | BR | 120 | 9.9 |

Junções de tabelas – Junção de Múltiplas Tabelas

- (2) O conjunto de resultados intermediário será então utilizado no INNER JOIN com a tabela *FilmeElenco*. Veja que desta vez a condição de junção envolve uma chave composta (**titulo + ano do filme**).

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
FROM Filme F
INNER JOIN Pais P ON (F.pais = P.sigla)
INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)
```

Resultado intermediário

| | titulo | ano | resumo | nome_pais |
|---|-----------------------|------|---|----------------|
| 1 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and eventual ... | Estados Unidos |
| 2 | O Filho da Noiva | 2001 | At age 42, Rafael Belvedere is having a crisis. | Argentina |
| 3 | Noel: Poeta da Vila | 2006 | The biopic of Noel Rosa, one of the best brazilian poets and composers. | Brasil |
| 4 | A Dama de Ferro | 2011 | An elderly Margaret Thatcher talks to the imagined presence of her recently ... | Reino Unido |
| 5 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil |

FilmeElenco

| | titulofilme | ano | nomeartista |
|---|---------------|------|---------------|
| 1 | ENCE: O Filme | 2023 | Karl Pearson |
| 2 | ENCE: O Filme | 2023 | Ronald Fisher |
| 3 | ENCE: O Filme | 2023 | Alan Turing |
| 4 | ENCE: O Filme | 2023 | Ada Lovelace |

Junções de tabelas – Junção de Múltiplas Tabelas

- (2) Nota-se que o resultado final deve conter somente o filme sobre a Ence.

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
FROM Filme F
```

```
INNER JOIN Pais P ON (F.pais = P.sigla)
```

```
INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)
```

Resultado intermediário

| | titulo | ano | resumo | nome_pais |
|---|-----------------------|------|---|----------------|
| 1 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace and ... | Estados Unidos |
| 2 | O Filho da Noiva | 2001 | At age 42, Rafael Belvedere is having a crisis. | Argentina |
| 3 | Noel: Poeta da Vila | 2006 | The biopic of Noel Rosa, one of the best brazilian poets and composers. | Brasil |
| 4 | A Dama de Ferro | 2011 | An elderly Margaret Thatcher talks to the imagined presence of her recently ... | Reino Unido |
| 5 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil |

FilmeElenco

| | titulofilme | ano | nomeartista |
|---|---------------|------|---------------|
| 1 | ENCE: O Filme | 2023 | Karl Pearson |
| 2 | ENCE: O Filme | 2023 | Ronald Fisher |
| 3 | ENCE: O Filme | 2023 | Alan Turing |
| 4 | ENCE: O Filme | 2023 | Ada Lovelace |

Junções de tabelas – Junção de Múltiplas Tabelas

- (2) O resultado final contém apenas os dados de “ENCE: O Filme”, pois trata-se do único filme que possui elenco cadastrado no BD.
- Para cada artista do elenco (são quatro no total), o título, ano e resumo do filme será repetido (se você tiver alguma dúvida sobre o motivo, simule a execução deste comando em papel, da mesma forma que apresentamos diversas vezes nos exemplos da Álgebra Relacional).

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
FROM Filme F
INNER JOIN Pais P ON (F.pais = P.sigla)
INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)
```

Resultado final

| | titulo | ano | resumo | nome_pais | nomeArtista |
|---|---------------|------|--|-----------|---------------|
| 1 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Karl Pearson |
| 2 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Ronald Fisher |
| 3 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Alan Turing |
| 4 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Ada Lovelace |

Junções de tabelas – Junção de Múltiplas Tabelas

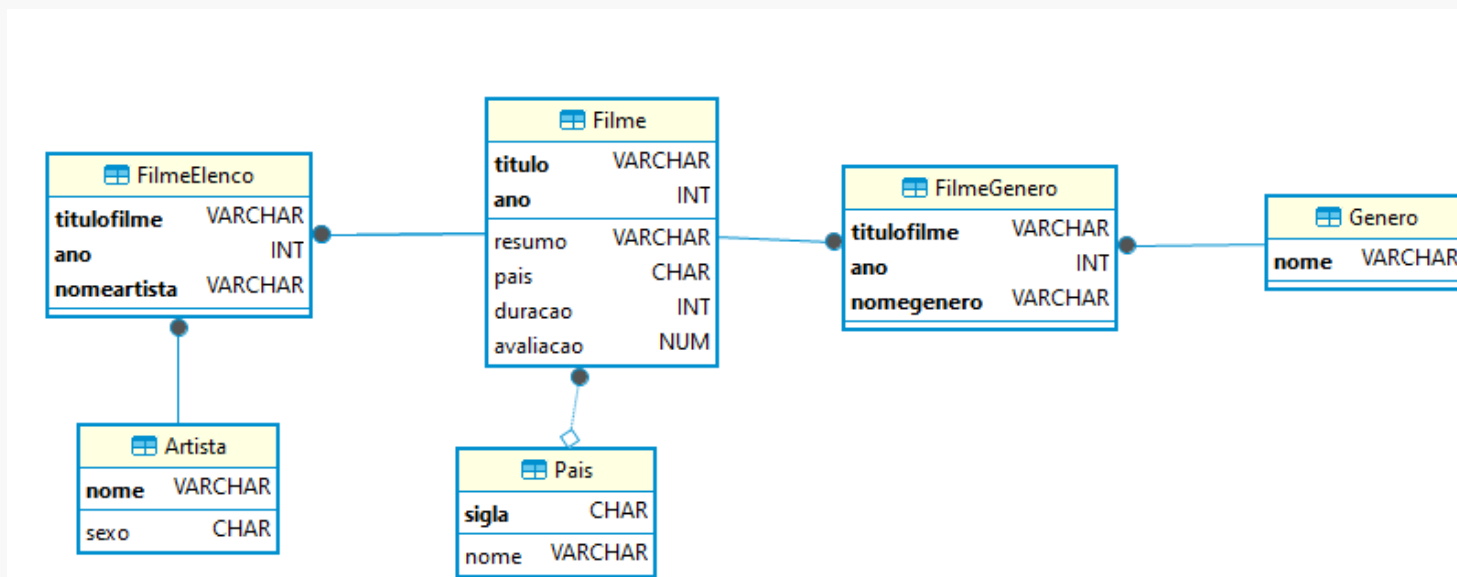
- É possível elaborar comandos que misturam INNER JOIN e LEFT JOIN. Por exemplo: listar os dados dos filmes com país cadastrado e apresentar o elenco dos mesmos caso exista.

```
SELECT F.titulo, F.ano, F.resumo, P.nome AS nome_pais, E.nomeArtista
FROM Filme F
INNER JOIN Pais P ON (F.pais = P.sigla)
LEFT JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano);
;
```

| | titulo | ano | resumo | nome_pais | nomeArtista |
|---|-----------------------|------|---|----------------|---------------|
| 1 | Um Sonho de Liberdade | 1994 | Two imprisoned men bond over a number of years, finding solace ... | Estados Unidos | NULL |
| 2 | O Filho da Noiva | 2001 | At age 42, Rafael Belvedere is having a crisis. | Argentina | NULL |
| 3 | Noel: Poeta da Vila | 2006 | The biopic of Noel Rosa, one of the best brazilian poets and ... | Brasil | NULL |
| 4 | A Dama de Ferro | 2011 | An elderly Margaret Thatcher talks to the imagined presence of her... | Reino Unido | NULL |
| 5 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Ada Lovelace |
| 6 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Alan Turing |
| 7 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Karl Pearson |
| 8 | ENCE: O Filme | 2023 | Um grupo de estudantes desvenda os mistérios da Ciência de Dados | Brasil | Ronald Fisher |

Junções de tabelas – Junção de Múltiplas Tabelas

- Considere agora a seguinte consulta: **listar o nome e sexo dos artistas e os nomes dos países de todos os filmes** onde estes artistas atuaram. Observe a figura abaixo e note que não há uma ligação direta entre as tabelas Pais (onde estão armazenados os nomes dos países) e Artista (onde está o nome e sexo de cada artista).
- O “caminho” que existe entre essas duas tabelas passa pelas tabelas Filme e FilmeElenco. Em situações como esta, **é preciso incluir todas as tabelas do caminho no SELECT a ser elaborado**. Entre cada par de tabelas, é incluído um JOIN para possibilitar a geração dos resultados intermediários.



Junções de tabelas – Junção de Múltiplas Tabelas

- O SELECT fez uso de três INNER JOIN: i) Pais e Filme; ii) Film e FilmeElenco; iii) FilmeEleco e Artista.

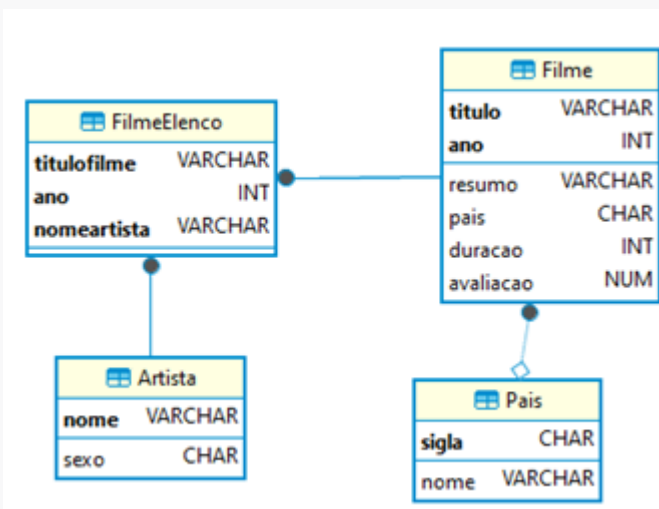
SELECT P.nome AS nome_pais, A.nome AS nome_artista, A.sexo

FROM Pais P

INNER JOIN Filme F ON (F.pais = P.sigla)

INNER JOIN FilmeElenco E ON (F.titulo = E.tituloFilme AND F.ano = E.ano)

INNER JOIN Artista A ON (E.nomeArtista = A.nome);



| | nome_pais | nome_artista | sexo |
|---|-----------|---------------|------|
| 1 | Brasil | Karl Pearson | M |
| 2 | Brasil | Ronald Fisher | M |
| 3 | Brasil | Alan Turing | M |
| 4 | Brasil | Ada Lovelace | F |

Junções de tabelas – Junção de Múltiplas Tabelas

- **Atenção:** Como observação final perceba que um SELECT que utiliza 4 tabelas precisa de 3 junções, já que as junções são feitas aos pares. Desta forma, um SELECT com n tabelas, precisará de n-1 junções.

Obrigado