

**INSTITUTO  
FEDERAL**  
Santa Catarina

# Banco de Dados 2

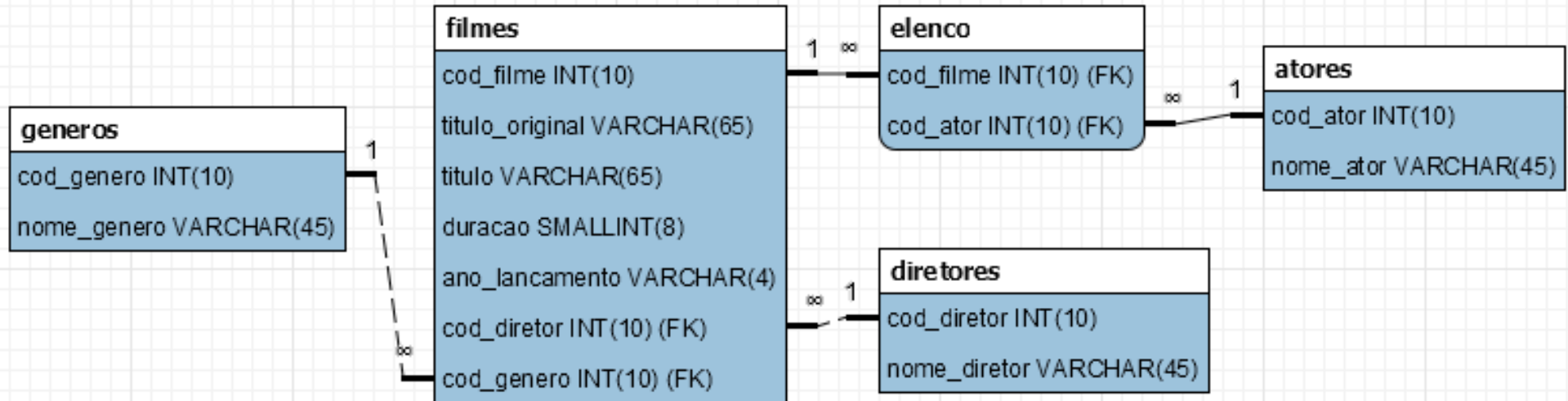


- 1) Junções
- 2) União
- 3) Interseção e diferença
- 4) Relações derivadas



# Materiais desta aula

Para esta aula precisaremos do BD “**filmes**”, cuja estrutura está representada abaixo.



1/4

# Junções

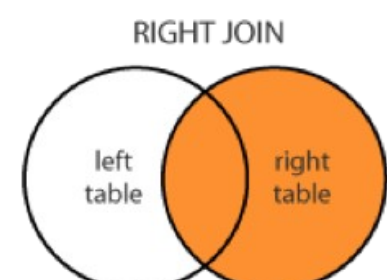
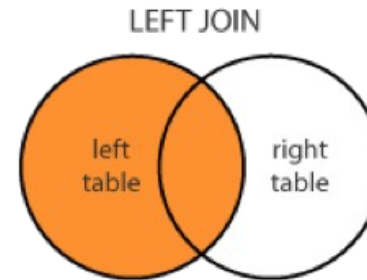
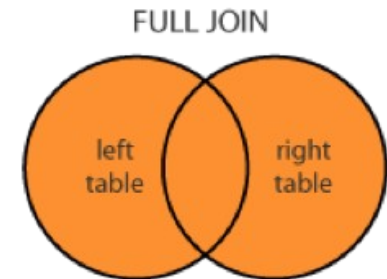
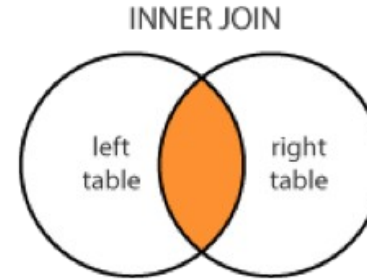


# Junções

- Uma forma mais elegante e poderosa de fazer seleções utilizando mais de uma tabela é através de junções.
- Operações de Junção tomam duas relações e retornam como resultado uma outra relação.
- Condições de Junção definem quais linhas das duas relações apresentam correspondência, e quais atributos serão apresentados no resultado de uma junção.

# Junções

- Junções podem ser
  - Internas
  - Externas esquerda
  - Externas direita
  - Completas



# INNER JOIN

- Chamada de junção interna
- Resultado igual à junção usando a **cláusula WHERE**
- Exemplo considerando as tabelas filmes e diretores

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f INNER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

# Comandos equivalentes

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f, diretores d  
WHERE f.cod_diretor=d.cod_diretor
```

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f INNER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```



# Lógica separada

```
SELECT f.titulo, g.nome_genero  
FROM filmes f, generos g  
WHERE f.cod_genero=g.cod_genero  
AND g.nome_genero='Suspense'
```

```
SELECT f.titulo, g.nome_genero  
FROM filmes f INNER JOIN generos g  
ON f.cod_genero=g.cod_genero  
WHERE g.nome_genero='Suspense'
```

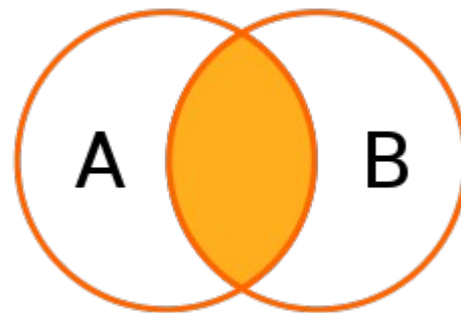
# INNER JOIN

- Na prática, este comando retornará uma relação contendo os nomes dos filmes e os nomes dos diretores SOMENTE dos filmes que possuem diretor.
- Filmes sem diretores (f.cod\_diretor=NULL) não aparecerão na lista.

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f INNER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

# INNER JOIN

- Ou seja, fará uma interseção entre as linhas de filmes e as linhas de diretores.



```
SELECT f.titulo, d.nome_diretor  
FROM filmes f INNER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

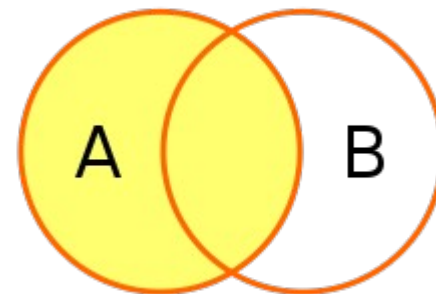
# LEFT JOIN

- Junção externa: LEFT OUTER JOIN
- Exemplo: como exibir todos filmes, e seus respectivos diretores, inclusive os que não possuem diretor?
- Processa a junção interna e adiciona as linhas da tabela da esquerda (filmes) que não estavam no resultado, associadas a valores **nulos** de diretores

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f LEFT JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

# LEFT JOIN

LEFT JOIN e LEFT OUTER JOIN é a mesma coisa.



```
SELECT f.titulo, d.nome_diretor  
FROM filmes f LEFT JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

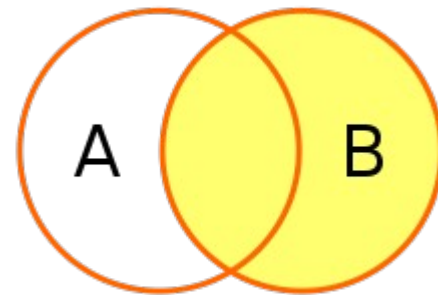
# RIGHT JOIN

- Junção externa: RIGHT OUTER JOIN
- Exemplo: como exibir os filmes, e seus respectivos diretores, inclusive os que diretores que não possuem filme?
- Processa a junção interna e adiciona as linhas da tabela da direita (diretores) que não estavam no resultado, associadas a valores **nulos** de filmes.

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f RIGHT JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

# RIGHT JOIN

Ou seja, se existe um diretor que ainda não possui um filme no catálogo, ele aparecerá associado a NULL filmes.



```
SELECT f.titulo, d.nome_diretor  
FROM filmes f RIGHT JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```



1. Qual o comando SQL que exibe todos os filmes/séries/shows de 2006 e seus respectivos diretores?
2. Qual o comando SQL que exibe todos os filmes/séries/shows que não possuem diretores?
3. Qual o comando SQL que exibe todos os diretores que não possuem nenhum filme?

## Dicas

- Use WHERE após a junção e ON
- Existe um operador para checar se um campo É NULO



- Como fazer uma junção com 3 tabelas?
- Exemplo: Exibir os títulos dos filmes e o nome dos atores.

```
SELECT f.titulo, a.nome_ator  
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme  
LEFT JOIN atores a ON e.cod_ator=a.cod_ator
```

2/4

# União



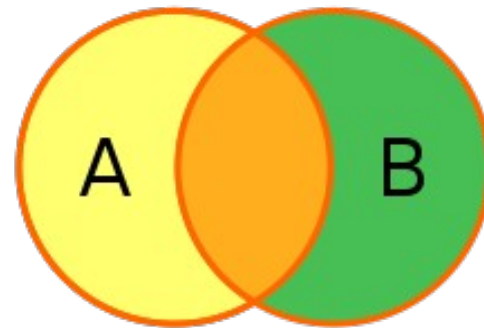
# FULL OUTER JOIN

- Junção externa completa
- Processa a junção interna e adiciona as linhas das duas tabelas que não estavam no resultado, associadas a valores nulos

```
SELECT f.titulo, d.nome_diretor  
FROM filmes f FULL OUTER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

# FULL OUTER JOIN

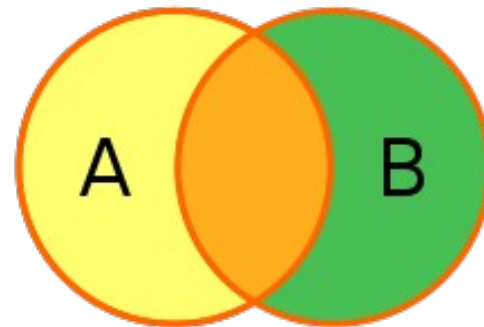
- Faz **união** dos resultados.
- No entanto, alguns SGDBs **NÃO** suportam, como por exemplo o **MySQL** e o **MariaDB**



```
SELECT f.titulo, d.nome_diretor  
FROM filmes f FULL OUTER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

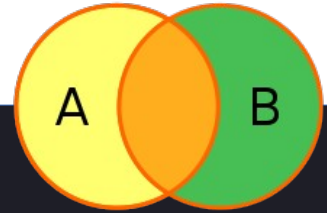
# FULL OUTER JOIN

- Então, como fazer uma união sem suporte do FULL OUTER JOIN?



```
SELECT f.titulo, d.nome_diretor  
FROM filmes f FULL OUTER JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

# FULL OUTER JOIN



```
SELECT f.titulo, d.nome_diretor, ano_lancamento  
FROM filmes f LEFT JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

→ **UNION**

```
SELECT f.titulo, d.nome_diretor, ano_lancamento  
FROM filmes f RIGHT JOIN diretores d  
ON f.cod_diretor=d.cod_diretor
```

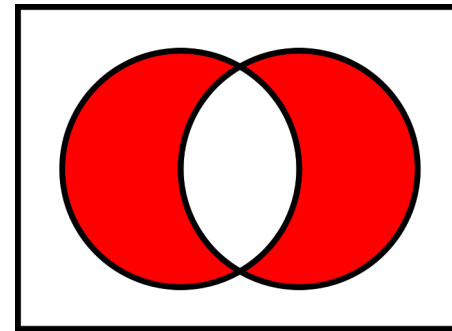
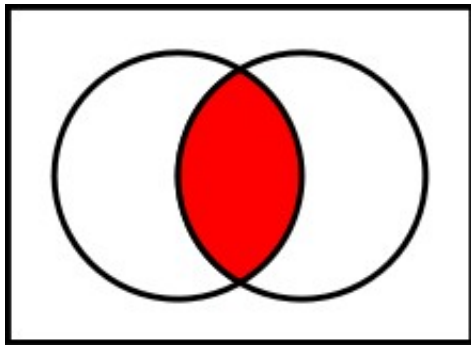
3/4

# Interseção e diferença



# Conectivos IN e NOT IN

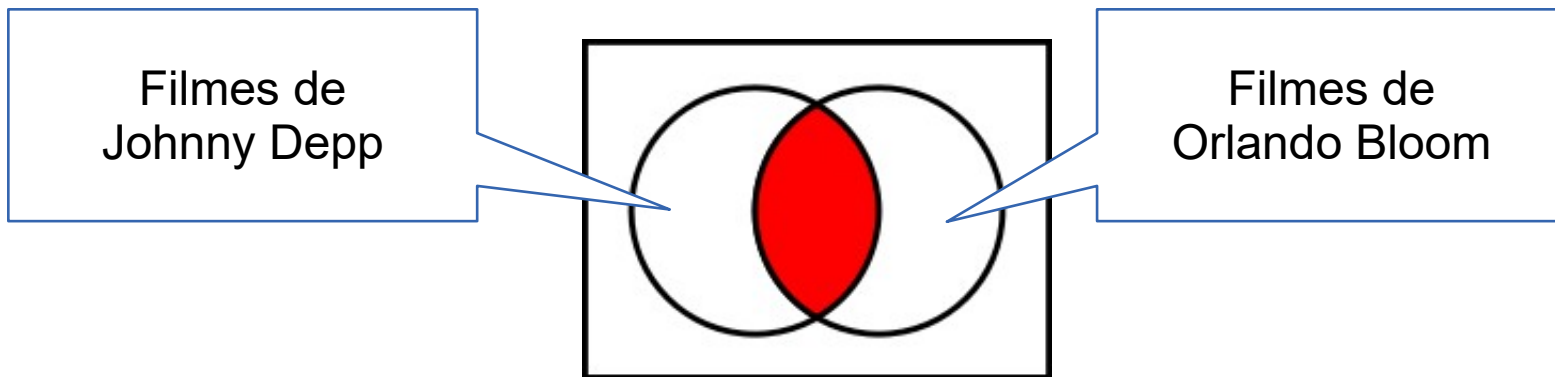
- **IN**: testa os membros de um conjunto, sendo o conjunto produzido por uma cláusula SELECT
- **NOT IN**: verifica a ausência de membros de um conjunto
- Permitem fazer a **interseção** ou a **diferença** entre conjuntos





# Conectivos IN e NOT IN

- Considere que você precisa da lista de **filmes** que **Johnny Depp** e **Orlando Bloom** atuaram **juntos**



# Conectivos IN e NOT IN

- Primeiro passo: encontrar os filmes do **Johnny Depp**

```
SELECT f.titulo  
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme  
LEFT JOIN atores a ON e.cod_ator=a.cod_ator  
WHERE a.nome_ator='Johnny Depp'
```

# Conectivos IN e NOT IN

- Segundo passo: encontrar os filmes do **Orlando Bloom**

```
SELECT f.titulo  
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme  
LEFT JOIN atores a ON e.cod_ator=a.cod_ator  
WHERE a.nome_ator='Orlando Bloom'
```

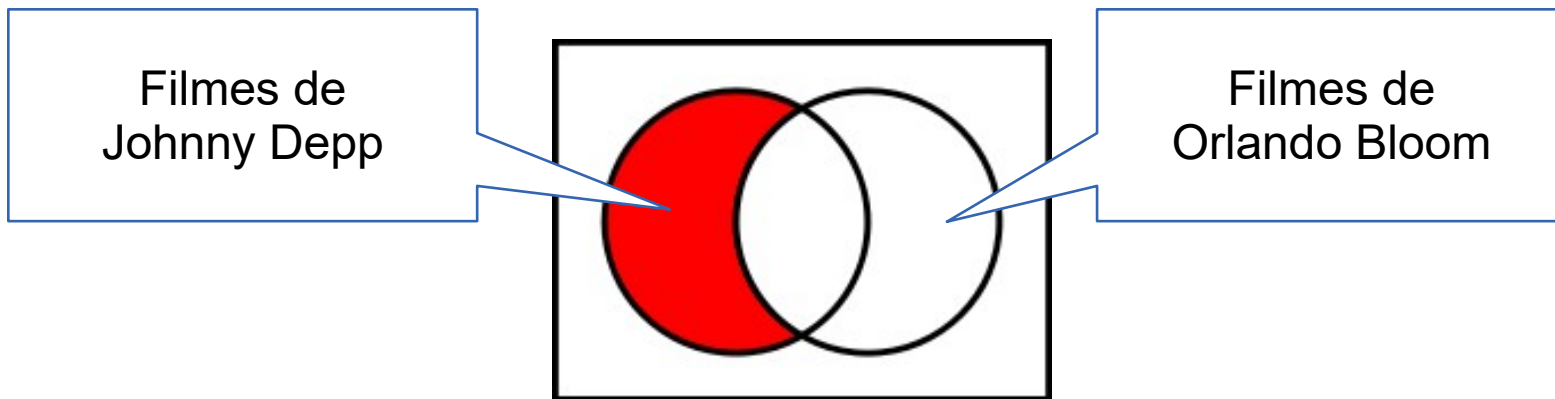
# Conectivos IN e NOT IN

- Terceiro passo: juntar os conjuntos.
- Para isso, verificamos se o título do filme está contido no outro conjunto usando IN

```
SELECT f.titulo
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme
LEFT JOIN atores a ON e.cod_ator=a.cod_ator
WHERE a.nome_ator='Johnny Depp' AND
→ f.titulo IN
(SELECT f.titulo
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme
LEFT JOIN atores a ON e.cod_ator=a.cod_ator
WHERE a.nome_ator='Orlando Bloom')
```

# Conectivos IN e NOT IN

- Considere que você precisa da lista de filmes que **Johnny Depp** atuou **SEM** o **Orlando Bloom**.



# Conectivos IN e NOT IN

- Terceiro passo: juntar os conjuntos.
- Para isso, verificamos se o título do filme está contido no outro conjunto usando IN

```
SELECT f.titulo
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme
LEFT JOIN atores a ON e.cod_ator=a.cod_ator
WHERE a.nome_ator='Johnny Depp' AND
→ f.titulo NOT IN
(SELECT f.titulo
FROM filmes f LEFT JOIN elenco e ON f.cod_filme=e.cod_filme
LEFT JOIN atores a ON e.cod_ator=a.cod_ator
WHERE a.nome_ator='Orlando Bloom')
```

4/4

# Relações derivadas



# Relações Derivadas

- Recurso que permite gerar uma relação através de uma subconsulta, atribuindo nome a esta e a seus atributos.
- Funciona como uma tabela temporária, ou seja, é possível fazer uma consulta sobre o resultado de outra consulta.




# Relações Derivadas


- Exemplo: Selecionar os filmes do gênero **Romance** **ou** **Drama** e exibindo seus títulos.

# Relações Derivadas

- **Primeiro passo:** selecionamos os gêneros e damos um nome ao resultado, tornando-o uma tabela temporária.
- Perceba que retornamos apenas a coluna `cod_genero`
- Este resultado servirá como uma nova tabela temporária com nome “preferidos”

```
(SELECT cod_genero  
FROM generos  
WHERE nome_genero='Drama'  
OR nome_genero='Romance') AS preferidos
```



cod_genero	
	6
	10

# Relações Derivadas

- **Segundo passo:** selecionamos os títulos dos filmes e fazemos a junção com a “tabela” preferidos.

```
SELECT filmes.titulo FROM filmes,  
(SELECT cod_genero  
FROM generos  
WHERE nome_genero='Drama'  
OR nome_genero='Romance') AS preferidos  
WHERE filmes.cod_genero=preferidos.cod_genero
```

# Testando seus conhecimentos

## Lista 2

