

Linguagem Do Banco de Dados

Prof. Sergio Luiz

Conteúdo



1. Operações de Conjuntos;
2. Operação de Union;
3. Valores Nulo;
4. Composição de Relações;
5. SQL Joins;
6. Criando Banco de Dados – Joins;
7. Inner Join;
8. Left Join;
9. Rght Join;
10. Outer Join;
11. Praticando Uso de JOIN;
12. Junção por Nome de Coluna.

Os operadores SQL-92 union, intersect e except operam relações e correspondem às operações de **união** (\cup), **interseção** (\cap) e **diferença** ($-$) da álgebra relacional, e portanto, as relações participantes devem ser compatíveis, ou seja, apresentar o mesmo conjunto de atributos (ou esquema).

A SQL-89 possui diversas restrições para o uso de union, intersect e except.
Certos produtos não oferecem suporte para essas operações.

A Operação de União - (\cup)

Una todas as tuplas da relação **“Colaborador”** para as quais:

O valor do atributo **“Nr_Depto”** seja **igual a 120** com as tuplas da relação **“Colaborador”** cujo o valor do atributo **“Nr_Depto”** seja **igual a 600**.

A Operação de União - (U)

```
select Nome_Completo, Salario, Nr_Depto from  
Colaborador where Nr_Depto = 120  
union  
select Nome_Completo, Salario, Nr_Depto from  
Colaborador where Nr_Depto = 671
```

Nome_Completo	Salario	Nr_Depto
Reeves,Roger	22935.00	120
Stansbury,Willie	39224.00	120
Bennet, Ann	22935.00	120
Papadopoulos,Chris	89655.00	671
Fisher,Pele	81810.00	671

O valor *null* indica a ausência de informação sobre o valor de um atributo.

Sendo assim, pode-se usar a palavra-chave *null* como predicado para testar a existência de valores nulos.

Valores Nulos

```
select Nome_Completo from Colaborador where  
Nr_Depto is null
```

Nome Completo

O predicado **not null** testa a ausência de valores nulos.

```
select Nome_Completo from Colaborador where  
Nr_Depto is not null
```

Nome_Completo
Parker,Bill
Young,Bruce
Nordstrom,Carol
Papadopoulos,Chris
Sutherland,Claudia
Bishop,Dana
Glou,Jacques

Além de fornecer o mecanismo básico do produto cartesiano para a composição das tuplas de uma relação disponível, o **SQL também oferece** diversos outros mecanismos para composição de relações como as **junções condicionais e as junções naturais, assim como várias formas de junções externas**

Junção Interna (ou junção condicional):

inner join

Junção Externa à Esquerda:

left outer join

Junção Externa à Direita:

right outer join

Junção Externa Total:

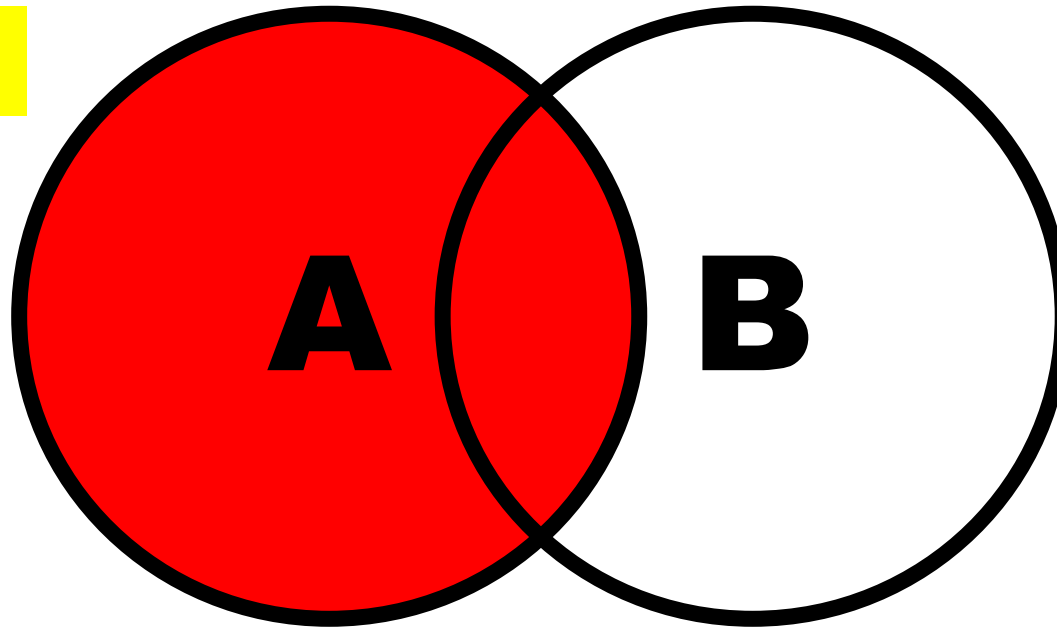
full outer join

Muitos desenvolvedores têm a dificuldade de saber qual resultado é retornado por cada **JOIN** no **SQL** e, portanto, quando devem utilizar cada um.

Para facilitar esse entendimento, os próximos SLIDES trarão uma **representação gráfica**, baseada na **Teoria dos Conjuntos**, muito conhecida na matemática.

Nas imagens, seguintes temos a representação de **duas tabelas (A e B)** e o resultado esperado por CADA TIPO DE JOIN (a **área em vermelho** representa os registros retornados pela consulta).

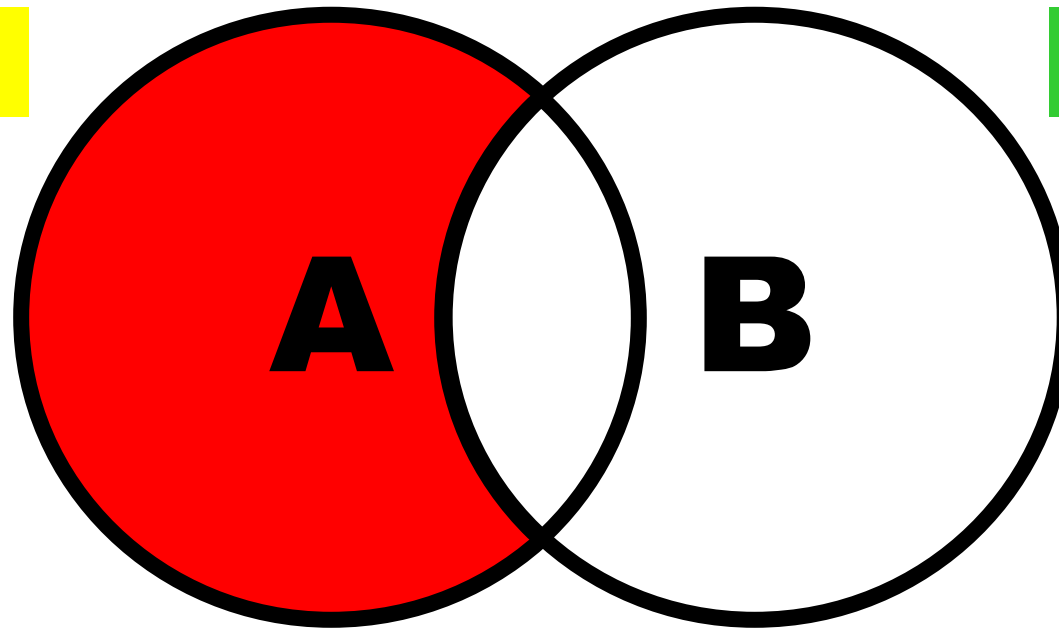
LEFT JOIN



```
SELECT <SELECT LIST>  
FROM TABELA_A  
LEFT JOIN TABELA_B  
ON A.Key = B.Key
```

SQL JOINS

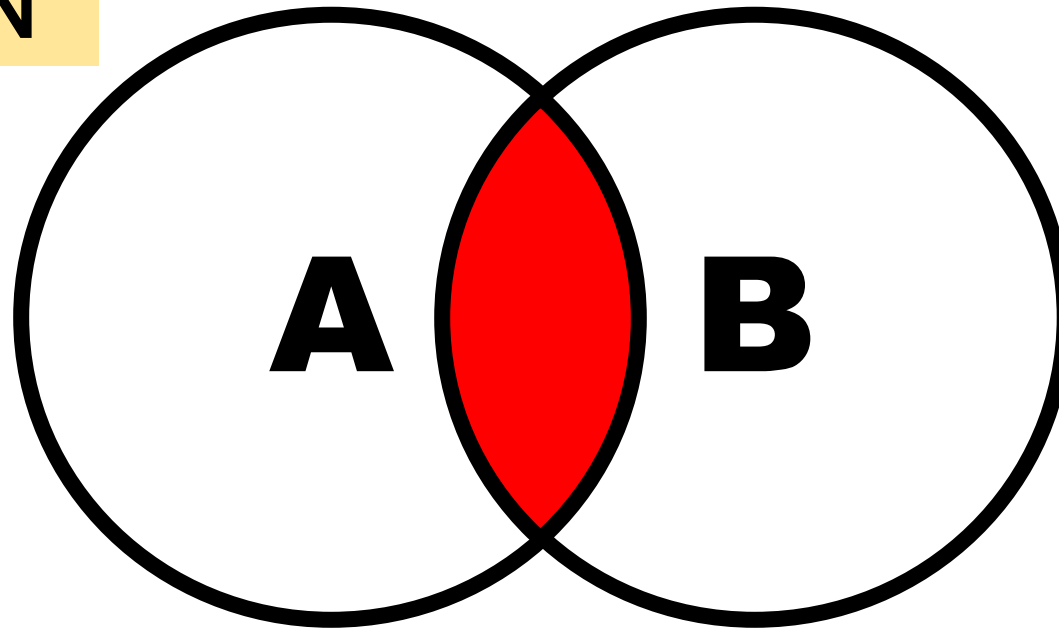
LEFT JOIN



IS NULL

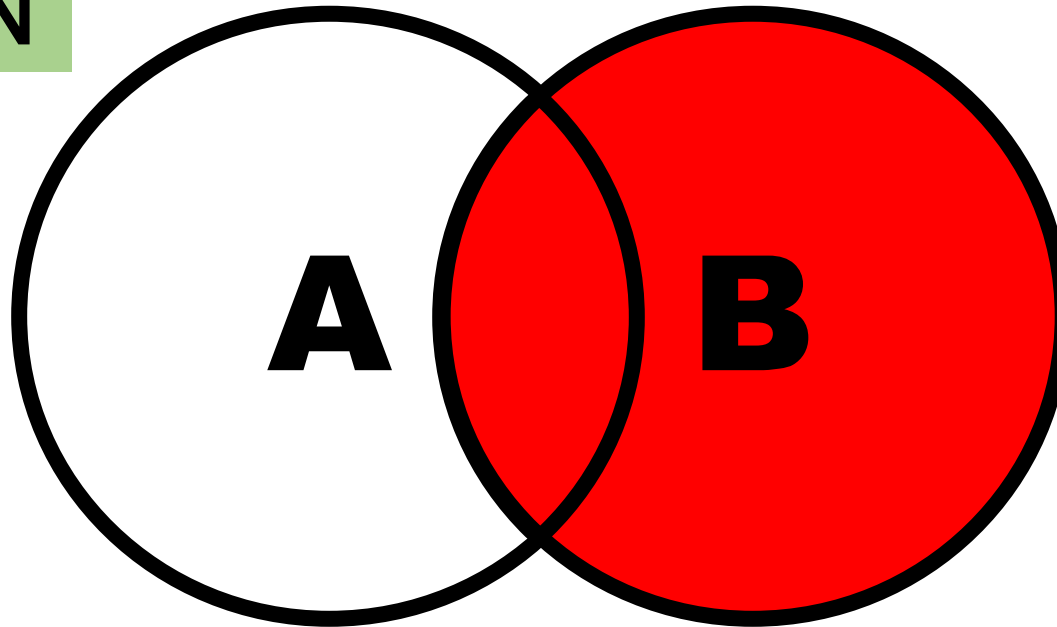
```
SELECT <SELECT LIST>  
FROM TABELA_A  
LEFT JOIN TABELA_B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```


INNER JOIN



```
SELECT <SELECT LIST>  
FROM TABELA_A  
INNER JOIN TABELA_B  
ON A.Key = B.Key
```

RIGHT JOIN

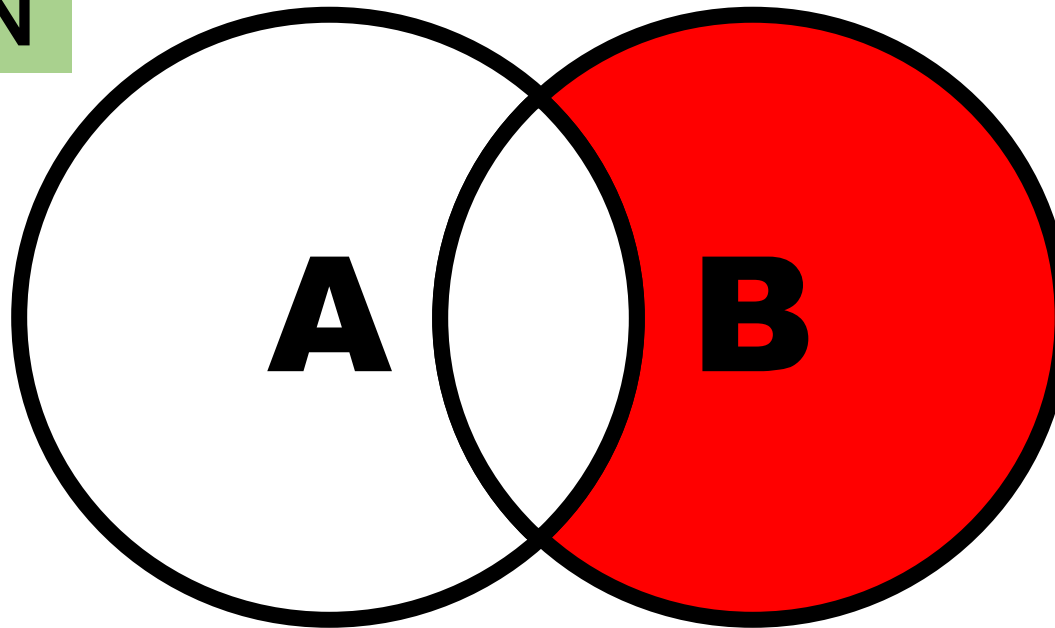


```
SELECT <SELECT LIST>  
FROM TABELA_A  
RIGHT JOIN TABELA_B  
ON A.Key = B.Key
```

SQL JOINS

RIGHT JOIN

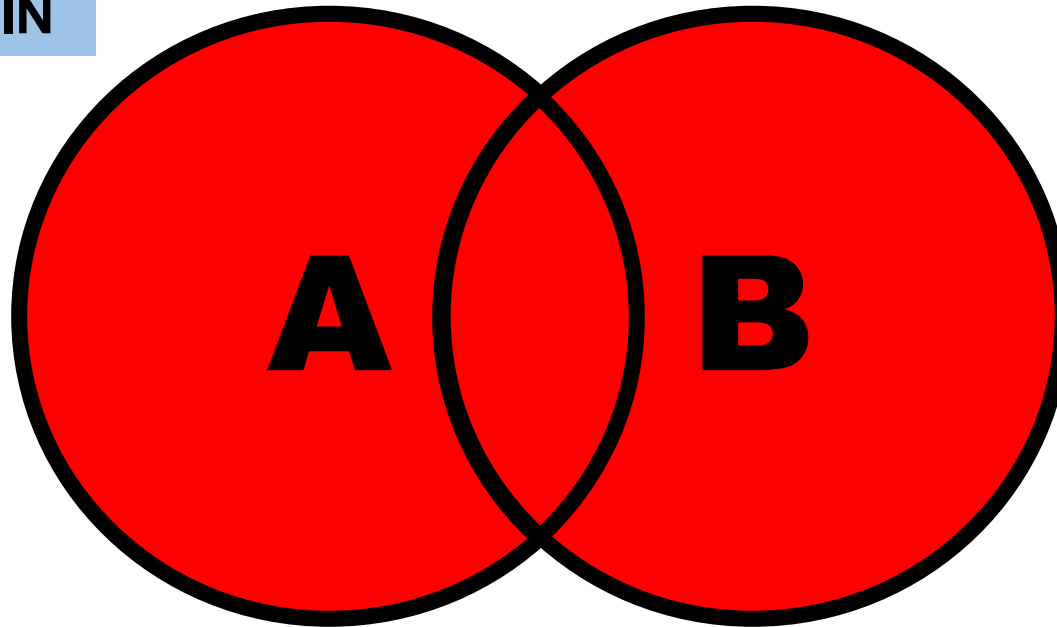
IS NULL



```
SELECT <SELECT LIST>  
FROM TABELA_A  
RIGHT JOIN TABELA_B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```

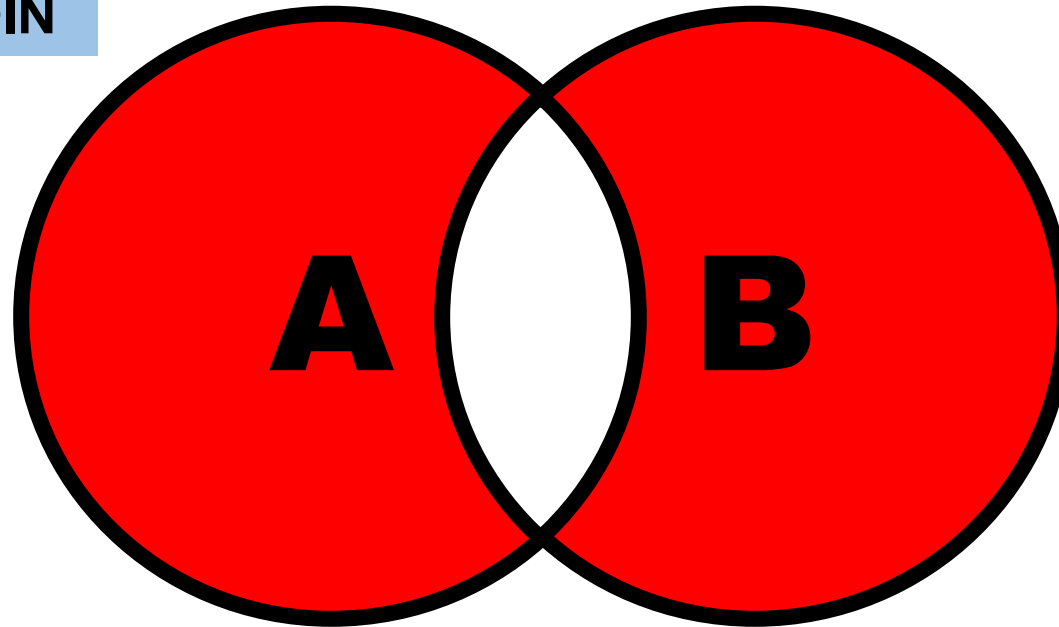
SQL JOINS

FULL OUTER JOIN



```
SELECT <SELECT LIST>  
FROM TABELA_A  
FULL OUTER JOIN TABELA_B  
ON A.Key = B.Key
```

FULL OUTER JOIN



```
SELECT <SELECT LIST>  
FROM TABELA_A  
FULL OUTER JOIN TABELA_B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

Nesses slides analisaremos cada **JOIN** individualmente, com base em exemplos, e veremos seus resultados.

Assim, poderemos comparar seu funcionamento e decidir quando devemos usar cada um.

Para demonstrar o funcionamento dos métodos de **junção (joins)**, precisaremos criar duas tabelas entre as quais deve haver algum relacionamento para que **possamos "cruzar" os dados.**

Como o objetivo criaremos apenas duas tabelas contendo uma coluna Nome, que será comum entre elas.

O script do próximo **SLIDE** **cria essa estrutura.**


```
CREATE TABLE Tabela_A  
(  
    Nome varchar(50) NULL  
)
```

```
CREATE TABLE Tabela_B  
(  
    Nome varchar(50) NULL  
)
```

Em seguida, precisaremos adicionar nas tabelas recém criadas alguns dados que nos permitam colocar à prova as junções.

Sendo assim, vamos inserir, com o script do próximo **SLIDE** alguns registros de forma que haja nomes que estão presentes apenas em uma tabela, e também nomes que sejam comuns às duas.

```
INSERT INTO Tabela_A  
VALUES
```

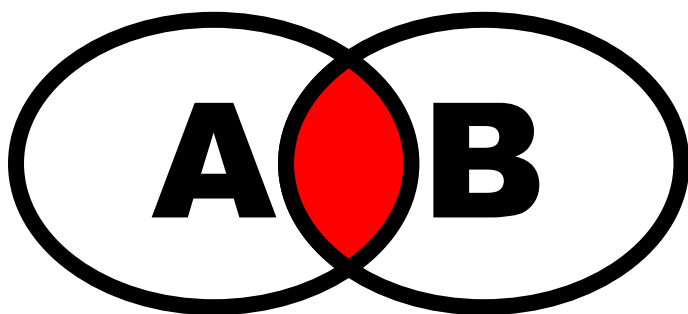
```
    ('Fernanda'),  
    ('Josefa'),  
    ('Luiz'),  
    ('Fernando');
```

```
INSERT INTO Tabela_B  
VALUES
```

```
    ('Carlos'),  
    ('Manoel'),  
    ('Luiz'),  
    ('Fernando');
```

INNER JOIN

O **INNER JOIN** é o método de junção mais conhecido e, como ilustra a figura abaixo, **retorna os registros que são comuns às duas tabelas.**



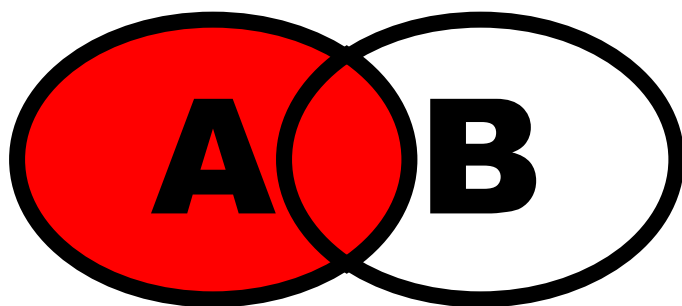
Exemplo de consulta com esse tipo de join

```
SELECT a.Nome, b.Nome  
FROM Tabela_A as A  
INNER JOIN Tabela_B as B  
on a.Nome = b.Nome
```

Resultado	
Nome	Nome
Luiz	Luiz
Fernando	Fernando

LEFT JOIN

O **LEFT JOIN**, tem como resultado todos os registros que estão na **tabela A** (mesmo que não estejam na **tabela B**) e os registros da **tabela B** que são comum a **tabela A**, veja como ilustra a figura abaixo.



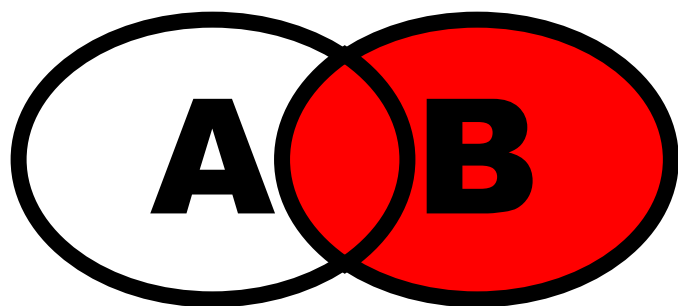
Ex. de consulta com esse tipo de left join

```
SELECT a.Nome, b.Nome
FROM Tabela_A as A
LEFT JOIN Tabela_B as B
on a.Nome = b.Nome
```

Nome	Nome
Luiz	Luiz
Fernando	Fernando
Fernanda	NULL
Josefa	NULL

RIGHT JOIN

O **RIGHT JOIN**, conforme mostra a figura abaixo, têm como resultado todos os registros que estão na **tabela B** (mesmo que não estejam na **tabela A**) e os registros da **tabela A** que são comum a **tabela B**.



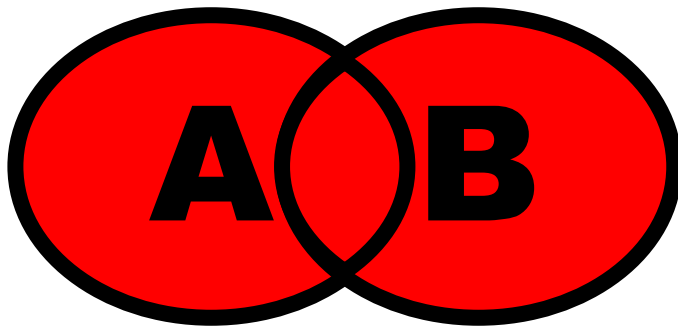
Ex. de consulta com esse tipo de right join

```
SELECT a.Nome, b.Nome
FROM Tabela_A as A
RIGHT JOIN Tabela_B as B
on a.Nome = b.Nome
```

Nome	Nome
Luiz	Luiz
Fernando	Fernando
NULL	Carlos
NULL	Manoel

OUTER JOIN

O **OUTER JOIN**, (também conhecido por Full Outer Join ou Full Join), conforme mostra a figura abaixo, têm como resultado todos os registros que estão na **tabela A** e todos os registros da **tabela B**.



Ex. de consulta com esse tipo de right join

```
SELECT a.Nome, b.Nome  
FROM Tabela_A as A  
FULL OUTER JOIN Tabela_B as B  
on a.Nome = b.Nome
```


PRATICANDO o APRENDIDO

Junção por Nome de Coluna

Digamos que você seja um fabricante de jogos de xadrez que tem uma **tabela de estoque** na qual monitora seu estoque de **peças brancas** e outra que monitora as **peças pretas**.

Junção por Nome de Coluna

As tabelas contem os dados a seguir:

BRANCA		
Peca	Qtde	Material
Rei	502	Carvalho
Rainha	398	Carvalho
Torre	1020	Carvalho
Bispo	985	Carvalho
Cavaleiro	950	Carvalho
Peão	431	Carvalho

PRETA		
Peca	Qtde	Material
Rei	502	Ébano
Rainha	397	Ébano
Torre	1020	Ébano
Bispo	985	Ébano
Cavaleiro	950	Ébano
Peão	453	Ébano

Para cada tipo de peça, o numero de **peças brancas** deve corresponder ao numero de **peças pretas**.

Se não houver correspondência, significa que algumas peças foram roubadas ou perdidas e você precisa aperfeiçoar a segurança.

Uma **junção natural** compara todas as colunas com o mesmo nome para verificar a igualdade.

Nesse caso, é produzida uma tabela resultante sem linhas, pois **nenhuma linha da coluna Material na tabela Branca** corresponde a **uma linha na coluna Material da tabela Preta**.

Essa tabela resultante não ajuda a determinar se esta faltando algum produto.

Em vez de usar uma junção natural ,
use uma junção por nome de coluna
para excluir a coluna Material do exame.
Ela pode ter a forma a seguir:

DESAFIO!

Resulta nessa consulta

Peca	Qtde	Material	Material
Rei	502	Carvalho	Ébano
Torre	1020	Carvalho	Ébano
Bispo	985	Carvalho	Ébano
Cavaleiro	950	Carvalho	Ébano

Junção por Nome de Coluna

Peca	Qtde	Material	Material
Rei	502	Carvalho	Ébano
Torre	1020	Carvalho	Ébano
Bispo	985	Carvalho	Ébano
Cavaleiro	950	Carvalho	Ébano

O que você percebeu nessa consulta?

Está faltando a **RAINHA** e os **PEÕES**, indicando a falta desses tipos de peças em algum lugar.

FIM