

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA



Linguagem SQL

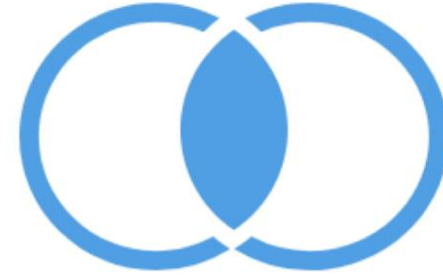
DQL - Junção de Tabelas



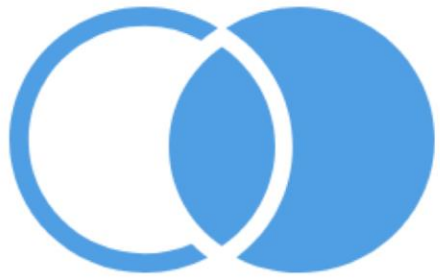
- A junção de várias tabelas permite consultar, em um único SELECT, informações contidas em diferentes tabelas.
- Para se fazer uma junção é necessário que as tabelas estejam relacionadas.
- A junção entre duas tabelas é realizada através da ligação entre a chave estrangeira de uma tabela e a respectiva chave primária referenciada.



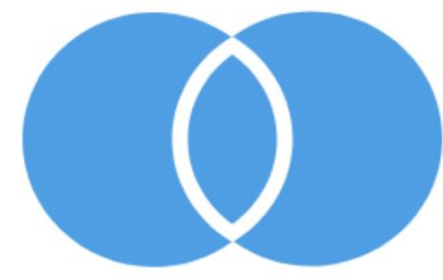
Left outer join



Inner join

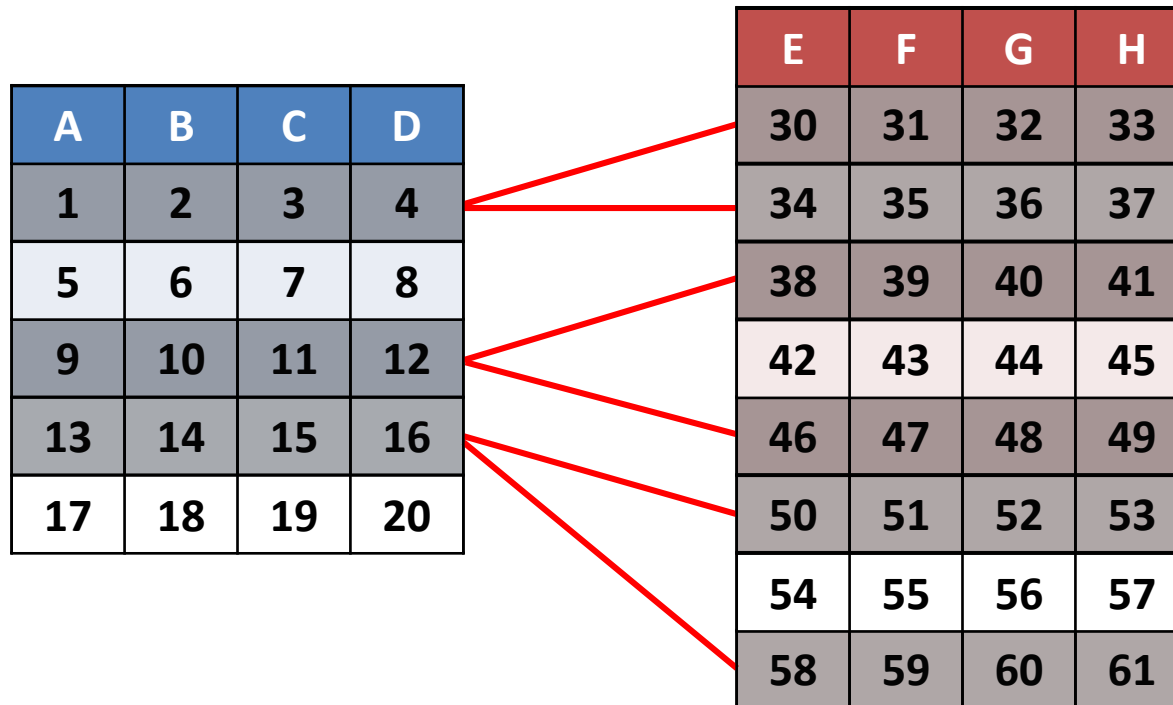


Right outer join



Full outer join

- **Junção Interna:** exibe apenas as linhas que participam do relacionamento, ou seja, apenas os registros em que exista ligação entre as tabelas serão exibidos.

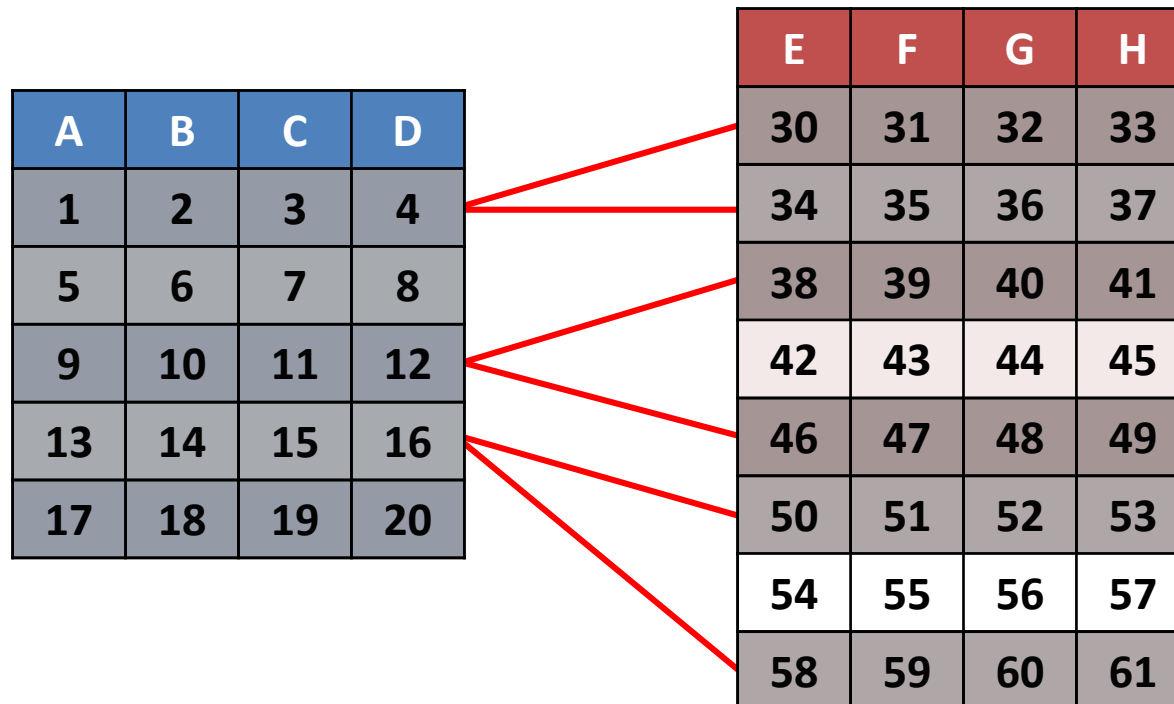


Junção Interna: Resultado

A	B	C	D	E	F	G	H
1	2	3	4	30	31	32	33
1	2	3	4	34	35	36	37
9	10	11	12	38	39	40	41
9	10	11	12	46	47	48	49
13	14	15	16	50	51	52	53
13	14	15	16	58	59	60	61



- **Junção Externa Esquerda:** exibe todas as linhas da tabela da esquerda e apenas os registros correspondentes na tabela da direita. Para as linhas onde não haja correspondência será exibido NULL.



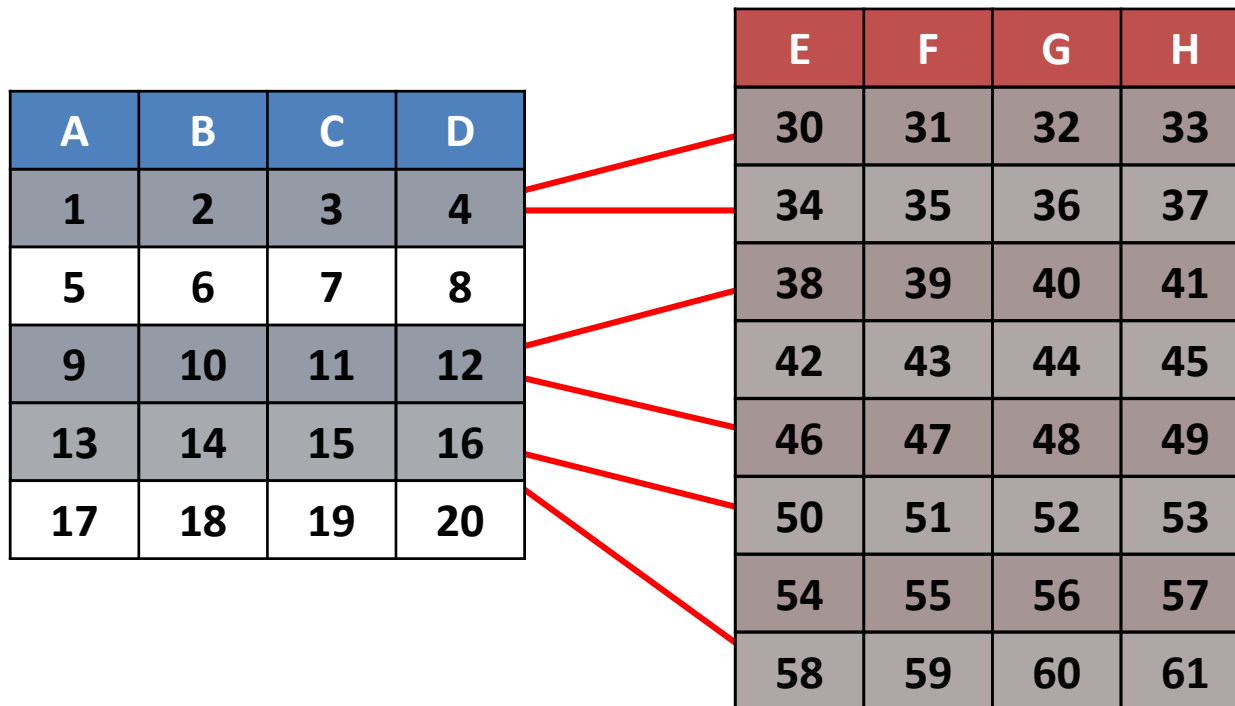


Junção Externa Esquerda: Resultado

A	B	C	D	E	F	G	H
1	2	3	4	30	31	32	33
1	2	3	4	34	35	36	37
5	6	7	8	N	N	N	N
9	10	11	12	38	39	40	41
9	10	11	12	46	47	48	49
13	14	15	16	50	51	52	53
13	14	15	16	58	59	60	61
17	18	19	20	N	N	N	N



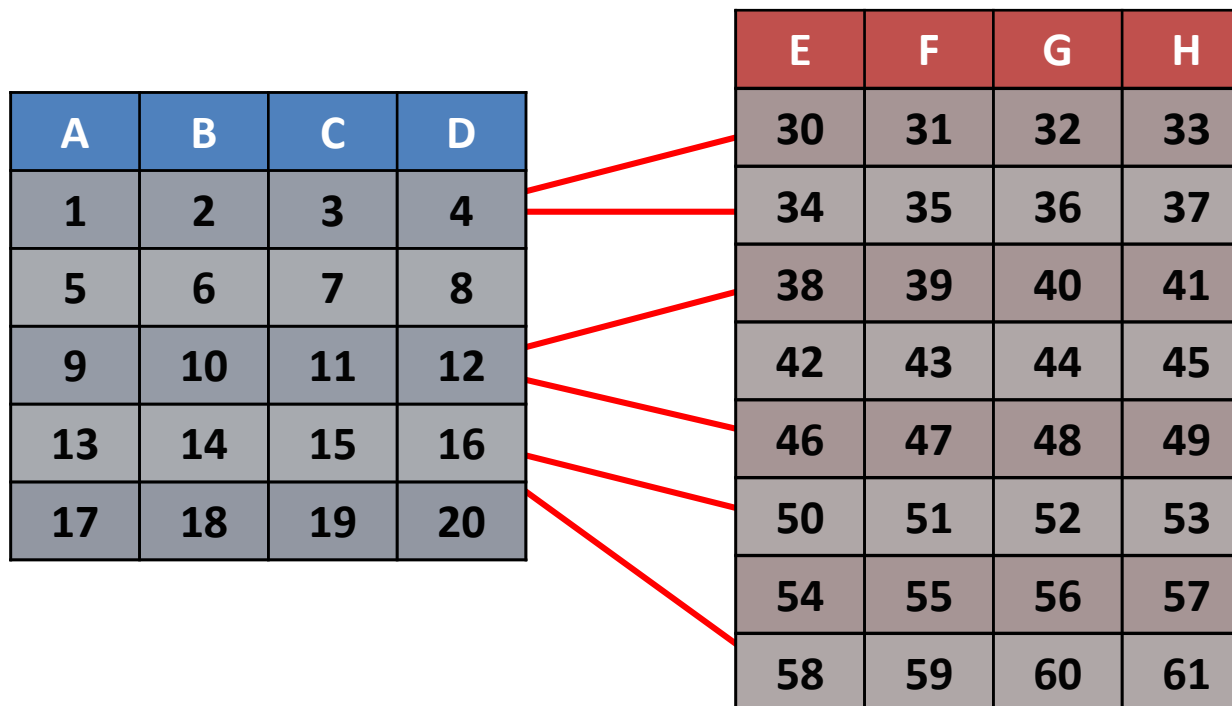
- **Junção Externa Direita:** exibe todas as linhas da tabela da direita e apenas os registros correspondentes na tabela da esquerda. Para as linhas onde não haja correspondência será exibido NULL.



Junção Externa Direita: Resultado

A	B	C	D	E	F	G	H
1	2	3	4	30	31	32	33
1	2	3	4	34	35	36	37
9	10	11	12	38	39	40	41
N	N	N	N	42	43	44	45
9	10	11	12	46	47	48	49
13	14	15	16	50	51	52	53
N	N	N	N	54	55	56	57
13	14	15	16	58	59	60	61

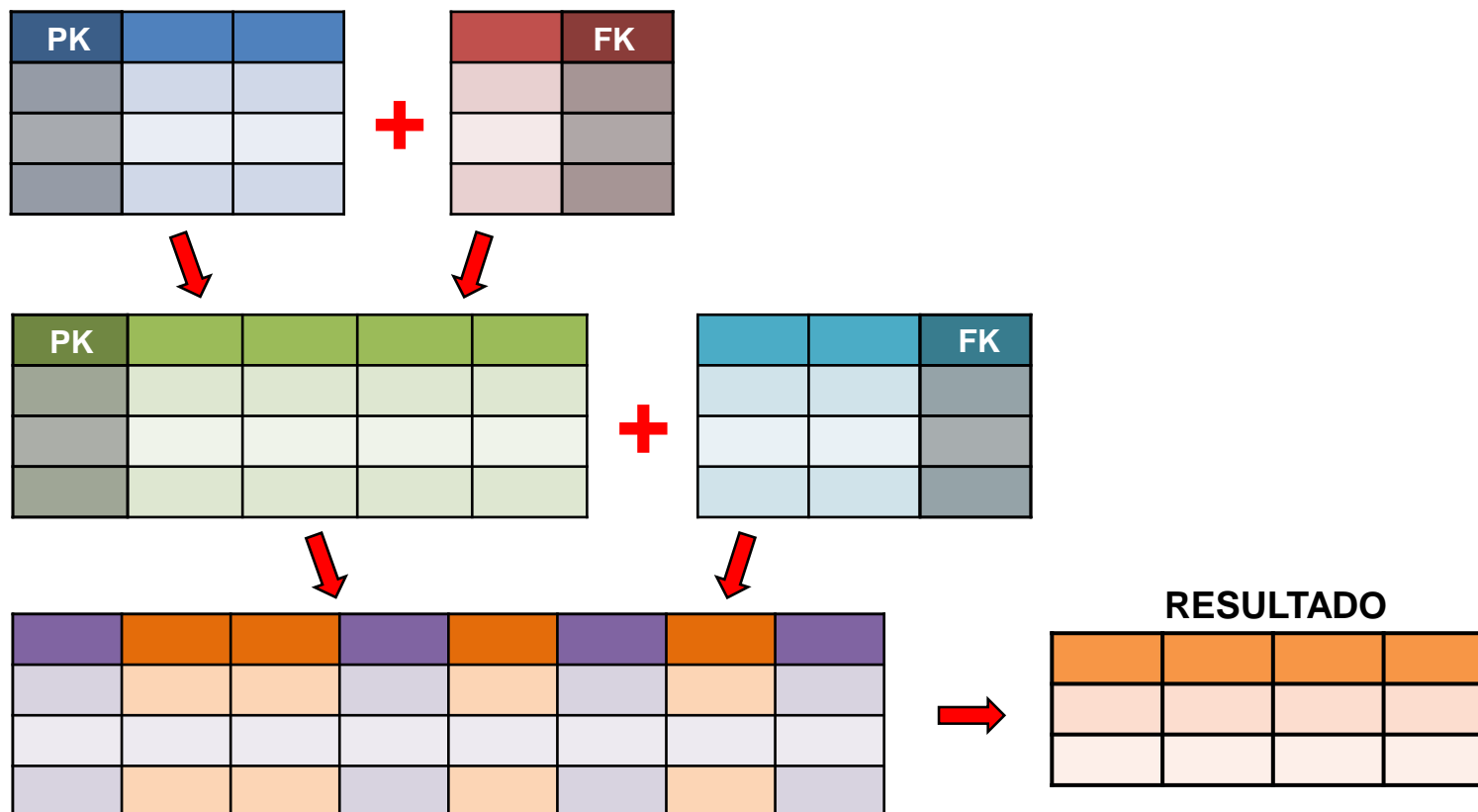
- **Junção Externa Completa:** exibe todas as linhas da tabela da direita e da esquerda, mesmo que existam linhas que não se relacionam. Para as linhas onde não haja correspondência será exibido NULL.



Junção Externa Completa: Resultado

A	B	C	D	E	F	G	H
1	2	3	4	30	31	32	33
1	2	3	4	34	35	36	37
5	6	7	8	N	N	N	N
9	10	11	12	38	39	40	41
9	10	11	12	46	47	48	49
N	N	N	N	42	43	44	45
13	14	15	16	50	51	52	53
13	14	15	16	58	59	60	61
17	18	19	20	N	N	N	N
N	N	N	N	54	55	56	57

- Pode-se unir várias tabelas em um único SELECT, mas a ligação é sempre feita entre duas tabelas de cada vez.





Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela1> [alias1]  
[INNER] JOIN <tabela2> [alias2]  
ON <condição de relacionamento>
```

```
SELECT <lista de colunas>  
FROM <tabela1> [alias1], <tabela2> [alias2]  
WHERE <condição de relacionamento>
```

- O formato da condição de relacionamento é:

PK da tabela1 = FK da tabela 2 ou

PK da tabela 2 = FK da tabela 1



Exemplo 1: Obter o código de cada pedido e o nome do cliente que realizou cada um deles, ordenados pelo código do pedido.

```
SELECT P.idpedido AS 'Pedido', C.nome AS 'Cliente'  
FROM Pedido P  
INNER JOIN Cliente C  
ON P.idcliente = C.idcliente  
ORDER BY P.idpedido;
```

```
SELECT P.idpedido AS 'Pedido', C.nome AS 'Cliente'  
FROM Pedido P, Cliente C  
WHERE P.idcliente = C.idcliente  
ORDER BY P.idpedido;
```



Exemplo 2: Obter, para cada funcionário, código e nome do funcionário, sigla e nome do setor onde trabalha, ordenados pela sigla do setor e nome do funcionário.

```
SELECT F.idfuncionario, F.nome AS 'Funcionário',  
       S.idsetor, S.nome AS 'Setor'  
FROM Funcionario F  
JOIN Setor S  
ON F.idsetor = S.idsetor  
ORDER BY S.idsetor, F.nome;
```

```
SELECT F.idfuncionario, F.nome AS 'Funcionário',  
       S.idsetor, S.nome AS 'Setor'  
FROM Funcionario F, Setor S  
WHERE F.idsetor = S.idsetor  
ORDER BY S.idsetor, F.nome;
```



Exemplo 3: Obter a quantidade, a média dos preços de venda e de custo dos produtos de cada tipo, mostrando o código e o nome do tipo do produto.

```
SELECT T.idtipo, T.nome AS 'Tipo',  
       COUNT(P.idproduto) AS 'Quantidade',  
       AVG(P.venda) AS 'Venda',  
       AVG(P.custo) AS 'Custo'  
FROM Produto P  
JOIN Tipo T  
ON P.idtipo = T.idtipo  
GROUP BY T.idtipo, T.nome;
```


INNER JOIN (com mais de duas tabelas)



Sintaxe:

```
SELECT <lista de colunas>
FROM <tabela1> [alias1]
[INNER] JOIN <tabela2> [alias2]
ON <condição de relacionamento1>
[INNER] JOIN <tabela3> [alias3]
ON <condição de relacionamento2> ...
[INNER] JOIN <tabelaN> [aliasN]
ON <condição de relacionamentoN>
```

```
SELECT <lista de colunas>
FROM <tabela1> [alias1], <tabela2> [alias2],
     <tabela3> [alias3], <tabelaN> [aliasN]
WHERE <condição de relacionamento1> AND
      <condição de relacionamento2> AND
      <condição de relacionamentoN>
```

INNER JOIN (com mais de duas tabelas)



Exemplo 4: Obter o código dos pedidos, a via de transporte de cada um deles, o nome do cliente e o nome do vendedor, ordenados pelo código do pedido.

```
SELECT P.idpedido 'Pedido', P.via,  
       C.nome 'Cliente', F.nome 'Vendedor'  
FROM Pedido P  
JOIN Cliente C ON P.idcliente = C.idcliente  
JOIN Funcionario F ON P.idvendedor = F.idfuncionario  
ORDER BY P.idpedido;
```

```
SELECT P.idpedido 'Pedido', P.via,  
       C.nome 'Cliente', F.nome 'Vendedor'  
FROM Pedido P, Cliente C, Funcionario F  
WHERE P.idcliente = C.idcliente AND  
       P.idvendedor = F.idfuncionario  
ORDER BY P.idpedido;
```

INNER JOIN (com mais de duas tabelas)



Exemplo 5: Obter o nome do funcionário, o nome da cidade onde reside, o nome da cidade e o nome do país onde nasceu, ordenados pelo nome do funcionário em ordem ascendente.

```
SELECT F.nome AS 'Funcionário', C1.nome AS 'Reside',  
       C2.nome AS 'Nasceu', P.nome AS 'País'  
FROM Funcionario F  
JOIN Cidade C1 ON F.idreside = C1.idcidade  
JOIN Cidade C2 ON F.idnatural = C2.idcidade  
JOIN Pais P ON C2.idpais = P.idpais  
ORDER BY F.nome;
```

INNER JOIN (com mais de duas tabelas)



Exemplo 6: Obter o nome do país e a quantidade de clientes existente em cada um deles.

```
SELECT P.nome, COUNT(Cl.idcliente) AS 'Quantidade'  
FROM Cliente Cl  
JOIN Cidade Ci ON Cl.idcidade = Ci.idcidade  
JOIN Pais P ON P.idpais = Ci.idpais  
GROUP BY P.nome;
```

INNER JOIN (com mais de duas tabelas)



Exemplo 7: Obter os nomes dos países estrangeiros e a quantidade de pedidos realizados pelos clientes de cada um deles, mostrando apenas os países para os quais houve mais de 10 pedidos.

```
SELECT Pa.nome, COUNT(Pe.idpedido) AS 'Quantidade'
FROM Pedido Pe
JOIN Cliente Cl ON Pe.idcliente = Cl.idcliente
JOIN Cidade Ci ON Cl.idcidade = Ci.idcidade
JOIN Pais Pa ON Ci.idpais = Pa.idpais
WHERE Pa.nome <> 'Brasil'
GROUP BY Pa.nome
HAVING COUNT(Pe.idpedido) > 10;
```



- A junção externa exhibe linhas que não participam de relacionamentos, diferentemente da ligação interna, onde só são exibidas as linhas que se relacionam.
- É útil para exibir dados de relacionamentos opcionais.
- Permite que todas as linhas de uma tabela sejam exibidas, ainda que existam linhas que não se relacionam com a outra tabela ligada pela junção.
- Há três tipos de junções externas:
 - Junção externa esquerda (LEFT OUTER JOIN)
 - Junção externa direita (RIGHT OUTER JOIN)
 - Junção externa completa (FULL OUTER JOIN)



Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela1> [alias1]  
LEFT [OUTER] JOIN <tabela2> [alias2]  
ON <condição de relacionamento>
```



Exemplo 8: Obter o código e o nome da função com o nome do funcionário que exerce cada uma delas, mostrando também as funções que não tenham funcionário, ordenando pelo nome da função e pelo nome do funcionário em ordem ascendente.

```
SELECT Fn.idfuncao, Fn.nome AS 'Função',  
       Fu.nome AS 'Funcionário'  
FROM Funcao Fn  
LEFT JOIN Funcionario Fu  
ON Fn.idfuncao = Fu.idfuncao  
ORDER BY Fn.nome, Fu.nome;
```




Sintaxe:

```
SELECT <lista de colunas>  
FROM <tabela1> [alias1]  
RIGHT [OUTER] JOIN <tabela2> [alias2]  
ON <condição de relacionamento>
```



Exemplo 9: Obter o código e o nome dos tipos de produto com os respectivos nomes dos produtos existentes para cada tipo, mostrando também os tipos que não tenham produto, ordenando pelo nome do tipo e nome do produto em ordem ascendente.

```
SELECT T.idtipo, T.nome AS 'Tipo',  
       P.nome AS 'Produto'  
FROM Produto P  
RIGHT JOIN Tipo T  
ON P.idtipo = T.idtipo  
ORDER BY T.nome, P.nome;
```



- O MySQL não possui uma instrução específica para realizar o **FULL OUTER JOIN**, entretanto isso pode ser resolvido aplicando-se **LEFT** e **RIGHT JOIN** com as tabelas e unindo os dois resultados com a instrução **UNION**.

Sintaxe:

```
SELECT <lista de colunas>
FROM <tabela1> [alias1]
LEFT JOIN <tabela2> [alias2]
ON <condição de relacionamento>
UNION
SELECT <lista de colunas>
FROM <tabela1> [alias1]
RIGHT JOIN <tabela2> [alias2]
ON <condição de relacionamento>
```



Exemplo 10: Obter o código e o nome das funções com o nome do funcionário que exercem cada uma delas, mostrando também as funções sem funcionário e os funcionários sem função, ordenando pelo nome da função e pelo nome do funcionário em ordem ascendente.

```
SELECT Fn.idfuncao, Fn.nome, Fu.nome  
FROM Funcao Fn  
LEFT JOIN Funcionario Fu ON Fn.idfuncao = Fu.idfuncao  
  
UNION  
  
SELECT Fn.idfuncao, Fn.nome, Fu.nome  
FROM Funcao Fn  
RIGHT JOIN Funcionario Fu ON Fn.idfuncao = Fu.idfuncao  
  
ORDER BY 2,3;
```



Exemplo 11: Obter a sigla e o nome dos setores da empresa com o nome do setor hierarquicamente superior, ordenando pela sigla do setor em ordem ascendente. Mostrar também os setores que não tenham setor superior.

```
SELECT S1.idsetor,  
       S1.nome AS 'Setor',  
       S2.nome AS 'Superior'  
FROM Setor S1  
LEFT JOIN Setor S2  
ON S1.idsuperior = S2.idsetor  
ORDER BY S1.idsetor;
```



Exemplo 12: Obter a sigla e o nome dos setores da empresa com o nome do chefe do setor e o nome do setor hierarquicamente superior, ordenando pela sigla do setor em ordem ascendente. Mostrar também os setores que não tenham setor superior.

```
SELECT S1.idsetor,  
       S1.nome AS 'Setor',  
       F.nome AS 'Chefe',  
       S2.nome AS 'Setor Superior'  
FROM Setor S1  
LEFT JOIN Setor S2  
ON S1.idsuperior = S2.idsetor  
INNER JOIN Funcionario F  
ON S1.idchefe = F.idfuncionario  
ORDER BY S1.idsetor;
```



Exemplo 13: Obter a sigla e o nome dos setores da empresa com o nome do chefe do setor, o nome do setor hierarquicamente superior e o nome do chefe do setor superior, ordenando pela sigla do setor em ordem ascendente.

```
SELECT S1.idsetor ,  
       S1.nome AS 'Setor' ,  
       F1.nome AS 'Chefe' ,  
       S2.nome AS 'Setor Superior' ,  
       F2.nome AS 'Chefe do Setor Superior'  
FROM Setor S1  
JOIN Setor S2 ON S1.idsuperior = S2.idsetor  
JOIN Funcionario F1 ON S1.idchefe = F1.idfuncionario  
JOIN Funcionario F2 ON S2.idchefe = F2.idfuncionario  
ORDER BY S1.idsetor;
```