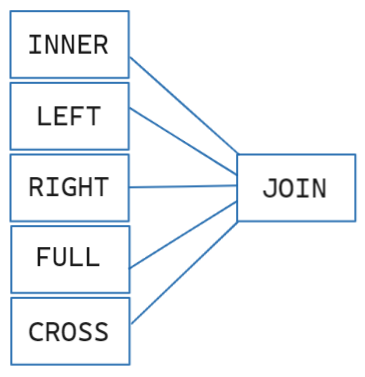
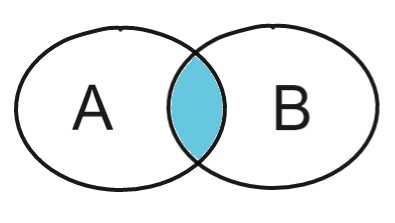
No SQL existem várias formas de realizar consultas no banco de dados utilizando desde queries simples a queries mais complexas. Um exemplo disso é a necessidade de realizar consultas em várias tabelas ao mesmo tempo. No banco de dados, existem várias tabelas diferentes onde os dados são armazenados e, muitas vezes, é necessário juntar os dados que estão em várias tabelas para se obter alguma informação muito importante.



Com esse intuito, foi criada a cláusula **JOIN** ou **junção** que é utilizada para realizar a combinação de colunas de uma ou mais tabelas em um única query a partir de uma coluna em comum entre as tabelas. Existem cinco tipos de JOIN, que realizam consultas de formas diferentes nas tabelas do banco de dados, o [**INNER JOIN**](https://www.alura.com.br/artigos/join-em-sql), **LEFT JOIN**, **RIGHT JOIN**, **FULL JOIN** e **CROSS JOIN**. Neste artigo, vamos abordar como cada um deles funciona.

**INNER JOIN**

O INNER JOIN é utilizado quando queremos retornar os registros que tenham correspondência nas duas tabelas presentes na junção. Observe que, na imagem a seguir, é utilizado o INNER JOIN. Nesse caso apenas os registros que estão na interseção do conjunto A com o conjunto B são retornados:



Trazendo para o SQL, vamos utilizar o seguinte exemplo: em um banco de dados de uma empresa, existem duas tabelas que se relacionam entre si, sendo elas a tabela de funcionários e a tabela de cargos. A primeira possui os campos **código do funcionário**, **nome** e **código do cargo**:



E a segunda possui os campos **código do cargo** e **descrição dos cargos**:



* **SELECT com INNER JOIN:**

SELECT <select\_list> FROM Tabela A INNER JOIN Tabela B ON A.Key = B.Key

SELECT A.nome, A.cod\_cargo, B.descrição FROM funcionario A INNER JOIN cargo B ON A.cod\_cargo = B.cod\_cargo;

* Informando no SELECT os campos que serão retornados no resultado:

SELECT A.nome, A.cod\_cargo, B.descrição

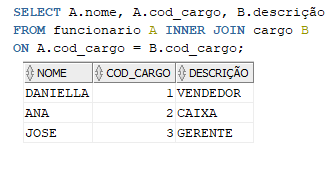
* Realizando a junção entre as tabelas:

FROM funcionario A INNER JOIN cargo B

* Condição que os registros precisam corresponder para serem retornados:

ON A.cod\_cargo = B.cod\_cargo;

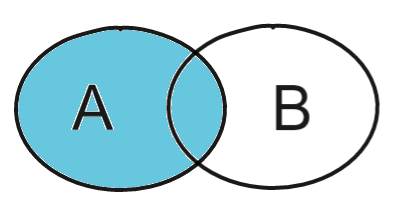
**Resultado**



Foram retornados todos funcionários, com exceção do funcionário Carlos que não tem um cargo associado a ele e do cargo entregador, pois não está associado a nenhum funcionário.

**LEFT JOIN**

O LEFT JOIN é utilizado quando queremos retornar apenas os registros da tabela da esquerda (tabela que está antes da cláusula *LEFT JOIN*) e os registros que tenham correspondência na tabela da direita. Observe que, na imagem a seguir, é utilizado o LEFT JOIN. Nesse caso, todos os registros do conjunto A e apenas os registros que estão na interseção do conjunto A com o conjunto B seriam retornados:

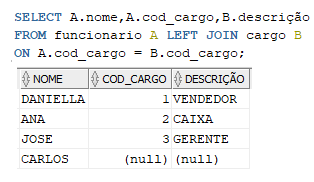


* **SELECT com LEFT JOIN:**

SELECT <select\_list> FROM Tabela A LEFT JOIN Tabela B ON A.Key = B.Key

SELECT A.nome, A.cod\_cargo, B.descrição FROM funcionario A LEFT JOIN cargo B ON A.cod\_cargo = B.cod\_cargo;

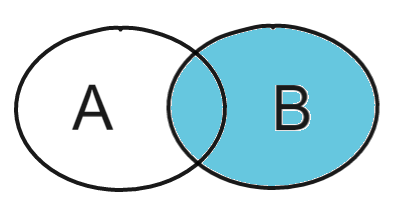
**Resultado**



Foram retornados todos funcionários, tem um cargo associado a ele, juntamente com todos os funcionários que não tem um cargo associado.

**RIGHT JOIN**

O RIGHT JOIN é utilizado quando queremos retornar apenas os registros da tabela da direita(tabela que está após a cláusula *RIGHT JOIN*) e os registros que tenham correspondência na tabela da esquerda. Observe que, na imagem a seguir, é utilizado o RIGHT JOIN. Nesse caso, todos os registros do conjunto B e apenas os registros que estão na interseção do conjunto A com o conjunto B seriam retornados:

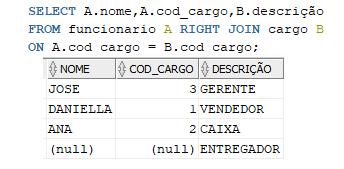


* **SELECT com RIGHT JOIN:**

SELECT <select\_list> FROM Tabela A RIGHT JOIN Tabela B ON A.Key = B.Key

SELECT A.nome,A.cod\_cargo,B.descrição FROM funcionario A RIGHT JOIN cargo B ON A.cod\_cargo = B.cod\_cargo;

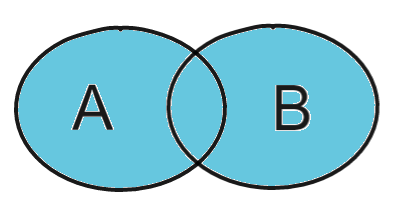
**Resultado**



Foram retornados todos os cargos que estão associados a um funcionário, juntamente com todos os cargos que não estão associados a funcionário.

**FULL JOIN**

O FULL JOIN é utilizado quando queremos retornar registros que tenham correspondência em qualquer uma das tabelas presentes na junção. Observe que, na imagem a seguir, é utilizado o FULL JOIN. Nesse caso, todos os registros do conjunto B e todos os registros do conjunto A seriam retornados:

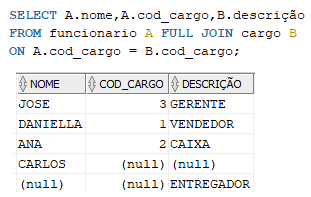


* **SELECT com FULL JOIN:**

SELECT <select\_list> FROM Tabela A FULL JOIN Tabela B ON A.Key = B.Key

SELECT A.nome, A.cod\_cargo, B.descrição FROM funcionario A FULL JOIN cargo B ON A.cod\_cargo = B.cod\_cargo;

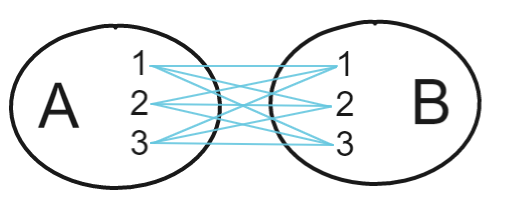
**Resultado**



Foram retornados todos os funcionários, mesmo os que não estão associados a um cargo e todos os cargos, mesmo os que não estão associados a um funcionário.

**CROSS JOIN**

O CROSS JOIN é utilizado quando queremos retornar os registros realizando um cruzamento entre os dados das tabelas presentes na junção. Observe que, na imagem a seguir, é utilizado o CROSS JOIN. Nesse caso, todos os registros do conjunto A realizam um cruzamento com todos os registros do conjunto B para serem retornados:

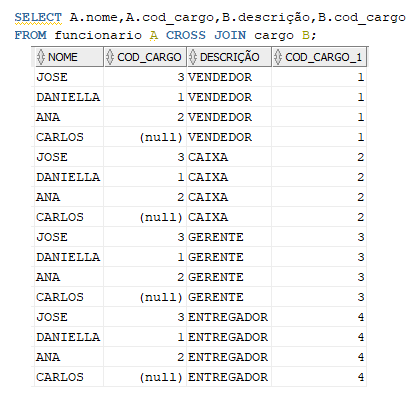


* **SELECT com CROSS JOIN:**

SELECT <select\_list> FROM Tabela A CROSS JOIN Tabela B

SELECT A.nome, A.cod\_cargo, B.descrição, B.cod\_cargo FROM funcionario A CROSS JOIN cargo B;

**Resultado**



Foi realizado o cruzamento entre todos os registros da tabela de funcionários com todos os registros da tabela de cargos.

**CONCLUSÃO**

Neste artigo podemos ver como os joins são de suma importância para a linguagem SQL. Ao utilizar os JOINS, podemos realizar a combinação de colunas de uma ou mais tabelas para realizar consultas complexas nos bancos de dados de forma simples.

Gostou deste artigo e quer conhecer ainda mais sobre banco de dados e SQL?

A [Formação Modelagem de Dados](https://cursos.alura.com.br/formacao-modelagem-dados) foi feita para você! Acesse a formação e descubra o que mais você pode fazer com banco de dados e SQL