

UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Sistemas de Banco de Dados 1

Junções em SQL

Pedro Henrique Carvalho Campos,190036435

Brasília, DF
2022

1.Introdução

1.2. O que são **Junções em SQL** (*JOINS*).

As Junções em SQL ou Joins, correspondem a operações de junção em álgebra relacional, que combina colunas de uma ou mais tabelas em um banco de dados relacional. Sendo assim, operações Join criam um conjunto que pode ser salvo com uma tabela ou até mesmo usado da forma como está, criando um relacionamento entre tabelas.

Em um banco de dados relacional, os dados normalmente são distribuídos em várias tabelas lógicas. Normalmente é necessário que para obter um conjunto completo e mais significativo de dados seja preciso consultar dados de várias tabelas usando junções(Joins).

1.3. Vantagens na utilização de Junções em SQL.

1.4. Desvantagens que podem ser geradas na utilização de Junções em SQL.

Muitas vezes as pessoas acabam utilizando junções em SQL sem a devida necessidade. Normalmente a utilização do JOIN é recomendada para bancos normalizados, sendo que para bancos não normalizados podem acontecer alguns problemas. Em geral, buscar informações em lugares diferentes pode gerar prejuízos ao desempenho do banco de dados, sendo que ter que construir uma informação (o que acontece com o uso do JOIN) as vezes pode ser mais prejudicial do que ter uma informação pronta para consumo direto.

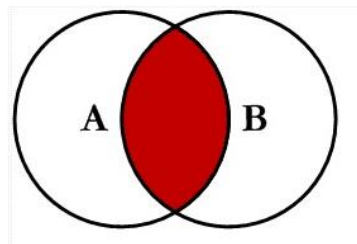
2. Tipos de Junções em SQL.

Os exemplos que serão utilizados abaixo terão como base as tabelas fictícias TabelaA e TabelaB onde os seguintes registros de nomes foram feitos:

```
1  INSERT INTO TabelaA VALUES('Fernanda')
2  INSERT INTO TabelaA VALUES('Josefa')
3  INSERT INTO TabelaA VALUES('Luiz')
4  INSERT INTO TabelaA VALUES('Fernando')
5
6  INSERT INTO TabelaB VALUES('Carlos')
7  INSERT INTO TabelaB VALUES('Manoel')
8  INSERT INTO TabelaB VALUES('Luiz')
9  INSERT INTO TabelaB VALUES('Fernando')
```

2.1. Inner Join.

O Inner Join pode ser considerado o método de junção em SQL mais conhecido, sendo que esse método de junção compara cada linha da tabela A com as linhas da tabela B para encontrar todos os pares de linhas que satisfazem a condição de junção retornando todos os registros que são comuns entre duas tabelas, como mostra a figura abaixo representado A e B.



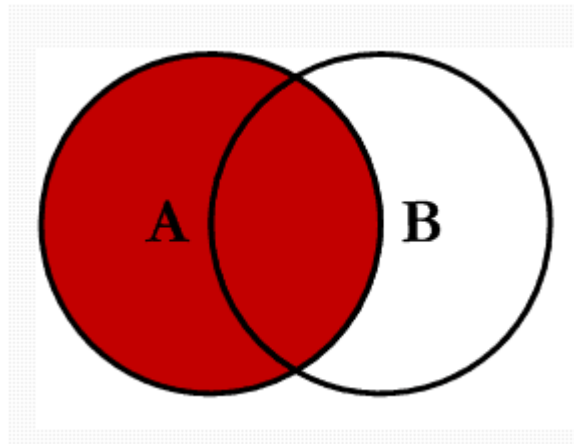
Um exemplo de consulta utilizando o Inner Join em SQL pode ser visualizado no trecho abaixo, sendo mostrado logo em seguida o resultado obtido em sql com base na Tabela A e TabalaB, destacadas anteriormente.

```
1 SELECT a.Nome, b.Nome
2 FROM TabelaA as A
3 INNER JOIN TabelaB as B
4      on a.Nome = b.Nome
```

	Nome	Nome
1	Luiz	Luiz
2	Fernando	Fernando

2.2. Left Join.

O Left Join retorna todos os registros da tabela esquerda e os registros correspondentes da tabela da direita. Sendo assim, para cada linha da tabela da esquerda (tabela A) a consulta compara todas as linhas da tabela B. Se um par de linhas fizer com que a condição seja verdadeira, os valores dessas linhas serão combinados para então formar uma nova linha. Em resumo, o Left Join tem como resultado todos os registros que estão na tabela A (mesmo que esses não estejam na tabela B) e os registros da tabela B que são comuns à tabela A, como representado na figura abaixo.



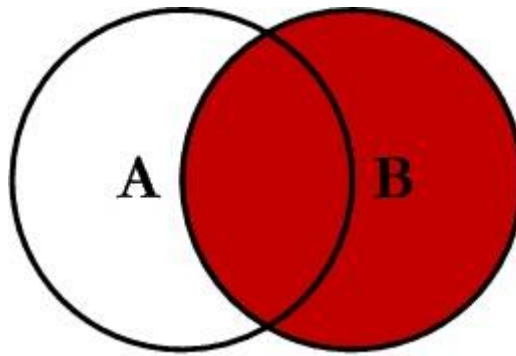
Abaixo é destacado um exemplo em SQL usando o Left Join e seu respectivo resultado em SQL.

```
1 SELECT a.Nome, b.Nome
2 FROM TabelaA as A
3 LEFT JOIN TabelaB as B
4      on a.Nome = b.Nome
```

	Nome	Nome
1	Fernanda	NULL
2	Josefa	NULL
3	Luiz	Luiz
4	Fernando	Fernando

2.3. Right Join

O Right Join, ao contrário do Left Join, retorna todos os registros da tabela da direita e os registros correspondentes da tabela da esquerda. Sendo assim, de maneira geral, o Right Join Combina dados de duas ou mais tabelas, começando a selecionar dados da tabela da direita (tabela B) e a corresponder linhas da tabela da esquerda, retornando um conjunto que inclui todas as linhas da tabela B com ou sem linhas correspondentes na tabela A. Em resumo, teremos 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 comuns à tabela B, como é destacado na ilustração abaixo.



O exemplo em SQL abaixo demonstra um uso do Right Join e seu respectivo resultado em SQL.

```

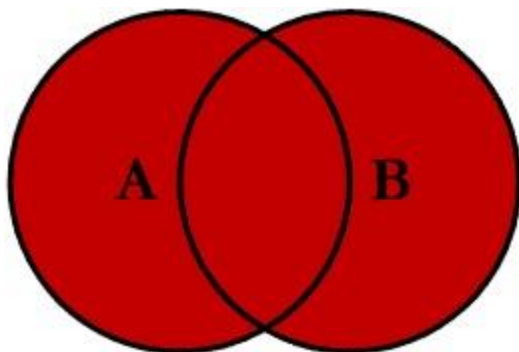
1 | SELECT a.Nome, b.Nome
2 | FROM TabelaA as A
3 | RIGHT JOIN TabelaB as B
4 | on a.Nome = b.Nome

```

	Nome	Nome
1	NULL	Carlos
2	NULL	Manoel
3	Luiz	Luiz
4	Fernando	Fernando

2.4. Outer Join/ Full Join

O Outer Join, que também pode ser conhecido por Full Outer Join ou até mesmo Full Join, retorna todos os registros quando houver uma correspondência na tabela esquerda ou direita. A Outer Join, de maneira geral, retorna todas as linhas das tabelas unidas, correspondidas ou não, sendo como o nome já diz, uma junção externa. Em resumo, o Outer join, tem como resultado todos os registros que estão na tabela A e todos os registros da tabela B, podendo ser considerado uma combinação das funções da Left Join e da Right Join.



Um exemplo do uso da Outer Join pode em SQL e seu respectivo resultado pode ser observado nas imagens abaixo:

```

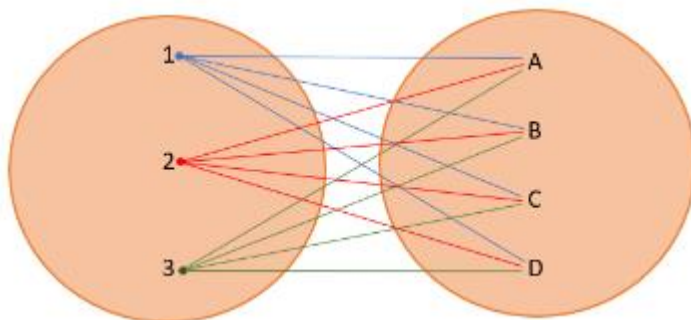
1 SELECT a.Nome, b.Nome
2 FROM TabelaA as A
3 FULL OUTER JOIN TabelaB as B
4     on a.Nome = b.Nome

```

	Nome	Nome
1	Fernanda	NULL
2	Josefa	NULL
3	Luiz	Luiz
4	Fernando	Fernando
5	NULL	Carlos
6	NULL	Manoel

2.5. Cross Join

A Cross Join retorna todas as linhas das tabelas por cruzamento, ou seja, para cada linha da tabela esquerda (Tabela A) queremos todas as linhas da tabela da direita (Tabela B), ou vice-versa. A Cross Join também é conhecida como produto cartesiano entre duas tabelas. Para que possa ser possível utilizar a Cross Join é necessário que aja pelo menos um campo em comum entre as tabelas, para que a ligação exista. Essa cláusula, normalmente deve ser utilizada com cuidado, pois pode gerar lentidão no uso do processamento dependendo do volume de dados. A representação do Cross Join pode ser visualizada melhor na figura abaixo.



Um exemplo em SQL que utiliza o Cross Join é apresentado abaixo, sendo que nesse exemplo a consulta iria retornar o cruzamento de todos os registros de funcionários com os registros de cargo em determinado banco SQL.

```

1. SELECT Func.NomeFuncionario
2.
3.     , Carg.DescCargo
4.
5. FROM FUNCIONARIO Func
6.
7. CROSS JOIN CARGO Carg

```

