

# **Junções em SQL**

**Magela López**

Curso de Analise e Desenvolvimento de Sistemas – Instituto Federal de Educação,  
Ciência e Tecnologia Sul-rio-grandense (IFSul) – Campus de Santana do Livramento –  
Santana do Livramento – RS – Brasil

Magegim4@gmail.com

## **1. Introdução**

O objetivo deste artigo é explicar o que são as junções em SQL, além de esclarecer quando e como utilizar os tipos junção INNER JOIN, LEFT JOIN e RIGHT JOIN.

Tanto Elmasri e Navathe (2010, p.105) como Ramakrishnan e Gehrke (2008, p.89) apontam a operação junção, da álgebra relacional, como uma das mais úteis e importantes para qualquer banco de dados relacional que precise combinar as informações de duas ou mais relações. Mesmo que as junções possam ser consideradas como uma operação produto cartesiano seguida por uma operação seleção e uma projeção, a junção é utilizada com maior frequência e tem um resultado mais específico e menor do que o resultado de um produto cartesiano normal.

Para Elmasri e Navathe (2010, p.105) A operação produto cartesiano “[...] em sua forma binária, [...] produz um novo elemento combinando cada membro (tupla) de uma relação (conjunto) com cada membro (tupla) da outra relação (conjunto).

Para atingir o objetivo do artigo será preciso utilizar alguns outros termos de álgebra relacional e da linguagem SQL, além de ver comandos e imagens que possam ilustrar o que será explicado.

## 2. Desenvolvimento

### 2.1 Álgebra Relacional

A álgebra relacional é uma forma de cálculo, além de uma linguagem procedural e um conjunto de operações, onde é utilizado como recurso de entrada uma ou mais relações, produzindo então, uma nova relação. A principal aplicação da mesma é extrair dados de um banco de dados e servir como base teórica das linguagens de consulta em banco de dados relacionais, entre as principais o SQL.

### 2.2 JOINS

É normal que em um banco de dados existam duas ou mais tabelas relacionadas, e quando se quer coletar informações de mais de uma relação é feito o uso de junções. Para isso usamos uma cláusula (que tem sua base na operação junção da álgebra relacional) conhecida como JOIN e é muito utilizada no SQL para fazer a ligação entre duas ou mais tabelas, a través da combinação dos valores das colunas. Geralmente são utilizadas a chave primaria de uma tabela e chave estrangeira da outra para concretizar essas ligações.

O resultado de uma join é uma nova relação, gerada somente para a execução da consulta, na qual poderão ser aplicadas outros tipos de operações. Essa nova relação contém as colunas e os valores das tabelas originais utilizadas para fazer a ligação, com o porem de que, o que será mostrado deve satisfazer a condição da junção, isso quer dizer que só vai aparecer no resultado a combinação de tuplas que satisfaçam a condição que foi especificada na hora de realizar a junção. Essa condição será declarada e avaliada nas duas tabelas que se pretende fazer a conexão. Ao mesmo tempo é importante saber que as linhas que vão aparecer na relação resultante dependem também do tipo de junção que será utilizada.

### 2.3 Tipos de Joins

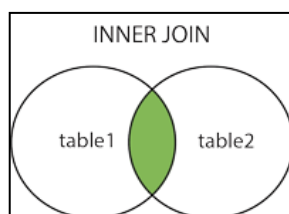
#### 2.3.1 INNER JOIN

É o tipo de junção que, faz com que no resultado, a nova relação tenha somente as linhas das tabelas de origem que cumprirem o critério disposto na clausula de junção. A consulta criada irá comparar cada linha da tabela1 com cada linha da tabela2 para encontrar os pares de linhas que satisfazem os critérios especificados, as linhas que vão entrar no resultado, além de cumprir com o predicado, devem ser não-nulos.

A sintaxe deste tipo de join é:

```
SELECT    nome_coluna1,    nome_coluna2,    ...    FROM    tabela1  
INNER JOIN tabela2 ON tabela1.nome_coluna = tabela2.nome_coluna;
```

Essa clausula vai permitir comparar os valores de colunas provenientes de tabelas associadas. Por meio do INNER JOIN, os registros de duas tabelas são usados para que sejam gerados os dados relacionados de ambas. Também são utilizadas as cláusulas WHERE e FROM para especificar esse tipo de associação.



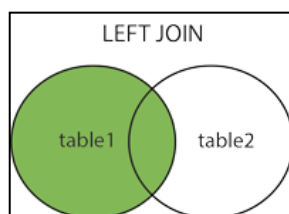
**Figura 1. Representação da instrução junção interna entre duas tabelas.**  
**Fonte: w3schools.com**

### 2.3.2 LEFT JOIN

No LEFT JOIN (que é uma junção externa), todas as linhas pertencentes a tabela da esquerda do termo LEFT JOIN e que cumpram com as especificações do predicado da junção, farão parte da relação resultante, independentemente se existem valores correspondentes na coluna de junção da tabela a direita. Quer dizer que, além de trazer os dados relacionados das tabelas, traz os dados não relacionados da tabela posicionada a esquerda da cláusula JOIN, nesse caso, os registros das tabelas a direita que não estiverem relacionados, terão valores nulos.

A sintaxe do LEFT JOIN é:

```
SELECT    nome_coluna1,    nome_coluna2,    ...    FROM    tabela1
LEFT JOIN tabela2 ON tabela1.nome_coluna = tabela2.nome_coluna;
```



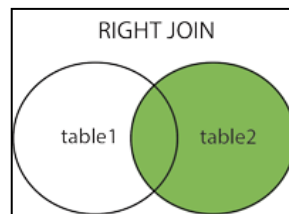
**Figura 2. Representação da instrução de junção externa a esquerda entre duas tabelas.**  
**Fonte: w3schools.com**

### 2.3.3 RIGHT JOIN

Muito similar a cláusula LEFT JOIN a diferença do RIGHT JOIN consta no fato de que, os dados que estarão na relação resultante, independente se existem valores relacionados ou não, são os que fazem parte da tabela da direita. Sendo assim, todos os registros encontrados na tabela a direita, serão mostrados no resultado da consulta, e quando não exista um valor correspondente na coluna de junção da tabela a esquerda, ele será nulo.

A sintaxe deste join é:

```
SELECT    nome_coluna1,    nome_coluna2,    ...    FROM    tabela1
RIGHT JOIN tabela2 ON tabela1.nome_coluna = tabela2.nome_coluna;
```



**Figura 2. Representação da instrução de junção externa a esquerda entre duas tabelas.**  
**Fonte: w3schools.com**

### 3. Conclusão

É muito importante conhecer os tipos de joins para ter o melhor desempenho na visualização do que você deseja. Usar o join certo para a situação correta será crucial para ter mais velocidade na mostra dos dados de forma precisa. Por isso que antes de escolher o tipo de junção a ser utilizado, tem que saber o que você quer ver para depois tomar a melhor decisão.

### Referências

- Navathe, S.B. Elmasri, R. Sistemas de Banco de Dados 6º edição, 2011.
- Ramakrishnan, R. Gehrke, J. Sistemas de Gerenciamento de Banco de Dados, 3º edição 2008.
- Carvalho Costa, R.L. SQL Guia Pratico, 2º edição, 2007.
- Microsoft. “Joins (SQL Server)”. Disponível: <https://docs.microsoft.com/pt-br/sql/relational-databases/performance/joins?view=sql-server-ver15> Acesso: novembro/2020.
- DevMedia. “Álgebra Relacional”, 2008. Disponível: <https://www.devmedia.com.br/algebra-relacional/9229> Acesso: novembro/2020.
- Macoratti, J.C. “SQL - Álgebra Relacional - Operações Fundamentais - Conceitos básicos”. Disponível: [http://www.macoratti.net/13/06/sql\\_arcb.htm#:~:text=A%20%C3%A1lgebra%20relacional%20%C3%A9%20uma,consulta%20em%20banco%20de%20dados](http://www.macoratti.net/13/06/sql_arcb.htm#:~:text=A%20%C3%A1lgebra%20relacional%20%C3%A9%20uma,consulta%20em%20banco%20de%20dados) Acesso: novembro/2020
- Camargo, W. DevMedia. “Cláusulas INNER JOIN, LEFT JOIN e RIGHT JOIN no SQL Server”, 2010. Disponível: <https://www.devmedia.com.br/clausulas-inner-join-left-join-e-right-join-no-sql-server/18930#:~:text=Ao%20contr%C3%A1rio%20do%20LEFT%20JOIN,JOIN%20%C2%ser%C3%A3o%20retornados%20valores%20nulos> Acesso: novembro/2020.
- w3schools.com. “SQL Joins” Disponível: [https://www.w3schools.com/sql/sql\\_join.asp](https://www.w3schools.com/sql/sql_join.asp) Acesso: novembro/2020.