



Disciplina: Sistemas de Banco de Dados 1	Turma: A
Professor: Vandor Rissoli	Período: 1º/2019
Aluno: Welison Lucas Almeida Regis	Matrícula: 17/0024121

## JUNÇÕES (JOINS) em SQL

**JOIN:** Uma junção (Join) SQL é uma instrução SQL (Structured Query Language) para combinar informações de dois conjuntos de dados (duas tabelas) com base em uma coluna relacionada entre as tabelas.

- **Vantagens do JOIN:**

- a) Velocidade no processamento: como se trata de uma atividade específica, as operações realizadas pelo JOIN são performadas mais rápidas que as operações mais genéricas, como filtros com cláusulas “WHERE”.
- b) combinar dados: possibilita combinar dados de duas tabelas com seus respectivos atributos de maneira fácil.

- **Desvantagens do JOIN:**

- a) Crescimento da quantidade de colunas: os JOINS devem ser feitos de maneira cautelosa a fim de evitar junções desnecessárias que causam o crescimento da tabela e prejudicam o desempenho das operações solicitadas.
- b) Acesso a múltiplas tabelas: o JOIN opera sobre tabelas em localidades distintas, logo, caso o banco não esteja devidamente normalizado, a operação não pode ser aplicada sobre o banco.

**\* TABELAS REFERENTES AOS JOINS ANEXAS AO FINAL DO DOCUMENTO.**

**INNER JOIN:** conforme figura 1, o INNER JOIN tem como resultado a **interseção entre duas tabelas**, isto é, os registros comuns entre os conjuntos de dados.

MySQL/SQL - obter uma lista dos vendedores que fizeram vendas e os detalhes da venda:

```
SELECT nome, dtVenda, total
FROM VENDEDOR V
INNER JOIN VENDA O
ON V.idVendedor = O.idVendedor;
```

#	nome	dtVenda	total
1	Ana Phelps	2019-02-11	300.12
2	Bruna Cavalcante	2019-02-19	490.32
3	Caio Stark	2019-02-14	192.29
4	Caio Stark	2019-02-22	70.00

**FULL JOIN (OUTER JOIN ou FULL OUTER JOIN):** resulta em **todos os registros que estão na tabela à esquerda e todos os registros da tabela à direita.**

O MySQL não disponibiliza método direto para o full join, porém, é possível realizá-lo da seguinte forma:

```
SELECT nome, dtVenda, total FROM VENDEDOR V
LEFT JOIN VENDA O ON V.idVendedor = O.idVendedor
UNION
SELECT nome, dtVenda, total FROM VENDEDOR V
RIGHT JOIN VENDA O ON V.idVendedor = O.idVendedor;
```

\* Comando **SQL** é idêntico ao inner join anterior, porém com as palavras “**FULL OUTER JOIN**”.

#	nome	dtVenda	total
1	Ana Phelps	2019-02-11	300.12
2	Bruna Cavalcante	2019-02-19	490.32

3	Caio Stark	2019-02-14	192.29
4	Caio Stark	2019-02-22	70.00
5	Diogo Stephen	NULL	NULL
6	Elena Rocha	NULL	NULL
7	NULL	2019-02-25	89.90



Figura 1 — Representação de INNER JOIN e FULL JOIN, sql.

**LEFT JOIN:** Ilustrado na figura 2, o LEFT JOIN retorna **todos os registros da tabela da esquerda** e aqueles da tabela direita que são comuns a tabela da esquerda (**interseção**).

MySQL/SQL - todos os vendedores e as respectivas vendas:

```
SELECT nome, dtVenda, total
FROM VENDEDOR V
LEFT JOIN VENDA O
ON V.idVendedor = O.idVendedor;
```

#	nome	dtVenda	total
1	Ana Phelps	2019-02-11	300.12
2	Bruna Cavalcante	2019-02-19	490.32
3	Caio Stark	2019-02-14	192.29

4	Caio Stark	2019-02-22	70.00
5	Diogo Stephen	NULL	NULL
6	Elena Rocha	NULL	NULL

**RIGHT JOIN:** semelhante ao LEFT JOIN, irá retornar **todos os registros da tabela da direita e também os registros da tabela da esquerda que são comuns a tabela da direita.**

MySQL/SQL - lista de todas as vendas e os respectivos vendedores:

```
SELECT nome, dtVenda, total
FROM VENDEDOR V
RIGHT JOIN VENDA O
ON V.idVendedor = O.idVendedor;
```

#	nome	dtVenda	total
1	Ana Phelps	2019-02-11	300.12
2	Bruna Cavalcante	2019-02-19	490.32
3	Caio Stark	2019-02-14	192.29
4	Caio Stark	2019-02-22	70.00
5	NULL	2019-02-25	89.90



Figura 2 — Representação do RIGHT JOIN e LEFT JOIN, sql

## **OUTROS TIPOS DE JOINS:**

- **CROSS JOIN:** cada tupla da **tabela da esquerda combina-se** com as tuplas da **tabela da direita**. Trata-se de um produto cartesiano.
- **JOINS EXCLUSIVOS:** utiliza-se dos mesmos procedimentos JOINS acima, porém, adiciona-se uma cláusula “WHERE”, exemplo “WHERE X.key is null” para filtrar o join realizado. São tipos excludentes: **left excluding join, right excluding join, outer excluding join**.

## TABELAS

### VENDEDOR

idVendedor	nome	email
1	Ana Phelps	ana.phelps@gmail.com
2	Bruna Cavalcante	brunacc1989@yahoo.com.br
3	Caio Stark	starkCaio9@hotmail.com
4	Diogo Stephen	stediogo@gmail.com
5	Elena Rocha	elenar19@gmail.com

### VENDAS

idVenda	idVendedor	dtVenda	total
1	1	2019-02-11	300.12
2	3	2019-02-14	192.29
3	2	2019-02-19	490.32
4	3	2019-02-22	70.00
5	NULL	2019-02-25	89.90