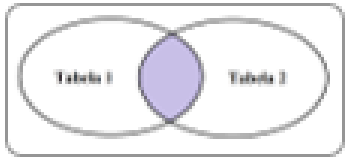
Junções (Joins)

Prof. Ramon Souza

Junções (Joins)

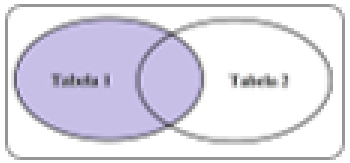
As **tabelas de junção**, que **combinam duas ou mais tabelas baseando-se em colunas relacionadas**. As tabelas de junção são especificadas segundo a cláusula **JOIN**.

Junções (Joins)



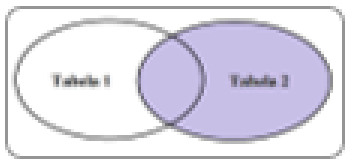
INNER JOIN (ou simplemente JOIN)

- Retorna somente os registros que possuem valores relacionados em ambas as tabelas, isto é, as intersecções.



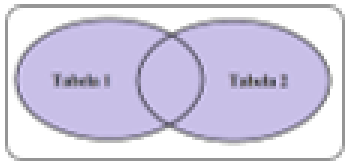
LEFT JOIN (ou LEFT OUTER JOIN)

- Retorna todos os registros da tabela da esquerda, e os registros relacionados da tabela da direita.
- Preenche campos não relacionados na tabela da direita com NULL.



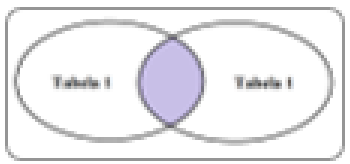
RIGHT JOIN (ou RIGHT OUTER JOIN)

- Retorna todos os registros da tabela da direita, e os registros relacionados da tabela da esquerda.
- Preenche campos não relacionados na tabela da esquerda com NULL



FULL OUTER JOIN

- Retorna todos os registros, independente de relação.
- Preenche campos não relacionados em qualquer das tabelas com NULL.



SELF JOIN

- União de uma tabela com ela mesma.

Junções (Joins)

A sintaxe básica para consultas com junções é:

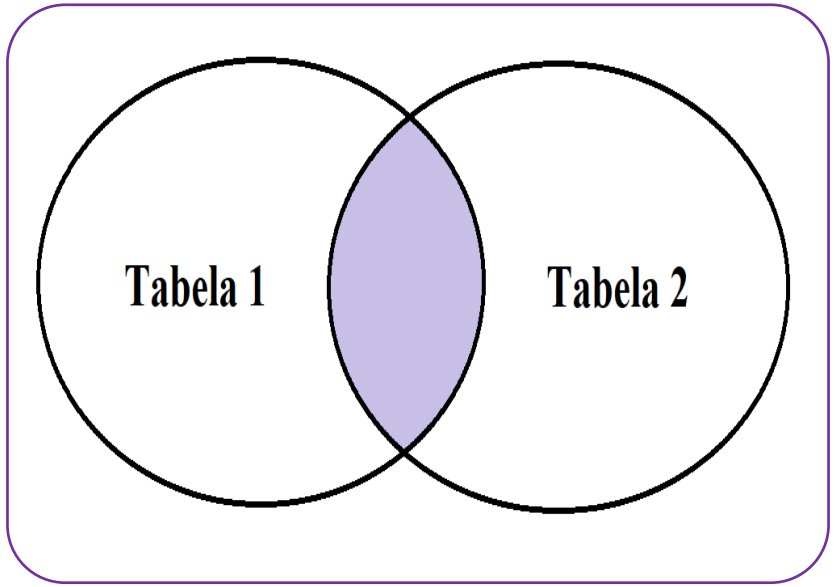
```
SELECT colunas FROM tabela1 JOIN tabela2 ON tabela1.coluna = tabela2.coluna;
```

Para o INNER JOIN, se as colunas em ambas as tabelas tiverem o mesmo nome, podemos usar a cláusula USING:

```
SELECT colunas FROM tabela1 INNER JOIN tabela2 USING (coluna);
```

INNER JOIN (ou simplesmente JOIN)

A cláusula **INNER JOIN** retorna somente os registros que possuem valores relacionados em ambas as tabelas, isto é, as intersecções.



Inner Join (exemplo)

Tabela PESSOAS		
NOME	CPF	ESTADO
Fernando	111.111.111-11	PR
Guilherme	222.222.222-22	SP

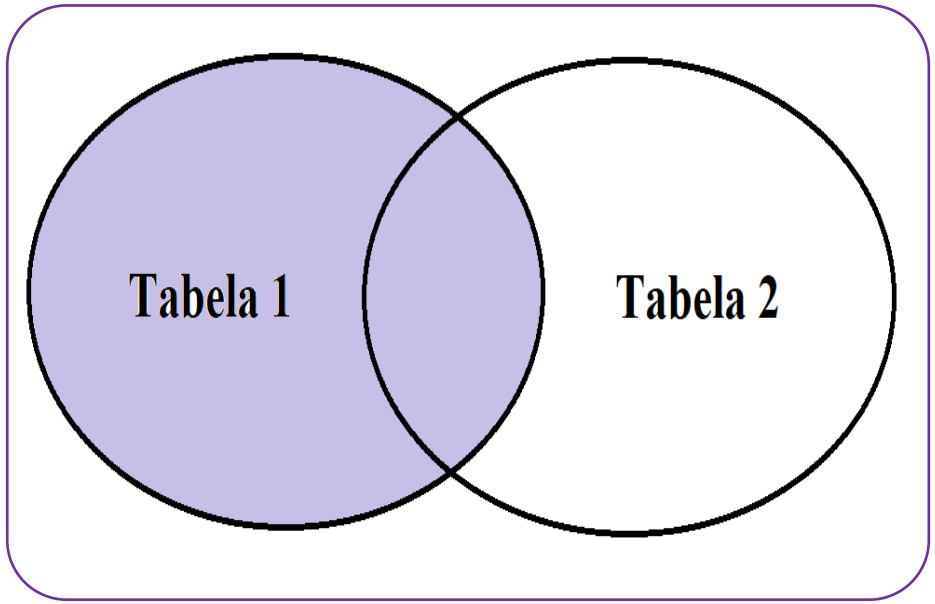
Tabela VEICULOS		
CPF	VEICULO	PLACA
111.111.111-11	Carro	SB-0001
NULL	Carro	SB-0002

```
SELECT * FROM PESSOAS INNER JOIN VEICULOS ON PESSOAS.CPF = VEICULOS.CPF; OU
SELECT * FROM PESSOAS JOIN VEICULOS ON PESSOAS.CPF = VEICULOS.CPF; OU
SELECT * FROM PESSOAS INNER JOIN VEICULOS USING(CPF); OU
SELECT * FROM PESSOAS JOIN VEICULOS USING(CPF);
```

Tabela RESULTADO					
NOME	CPF	ESTADO	CPF	VEICULO	PLACA
Fernando	111.111.111-11	PR	111.111.111-11	Carro	SB-0001

LEFT JOIN (ou LEFT OUTER JOIN)

A cláusula **LEFT JOIN** retorna todos os registros da tabela da esquerda, e os registros relacionados da tabela da direita. Caso não haja valores relacionados na tabela da direita, os seus campos serão preenchidos com NULL.



Left Join (exemplo)

Tabela PESSOAS		
NOME	CPF	ESTADO
Fernando	111.111.111-11	PR
Guilherme	222.222.222-22	SP

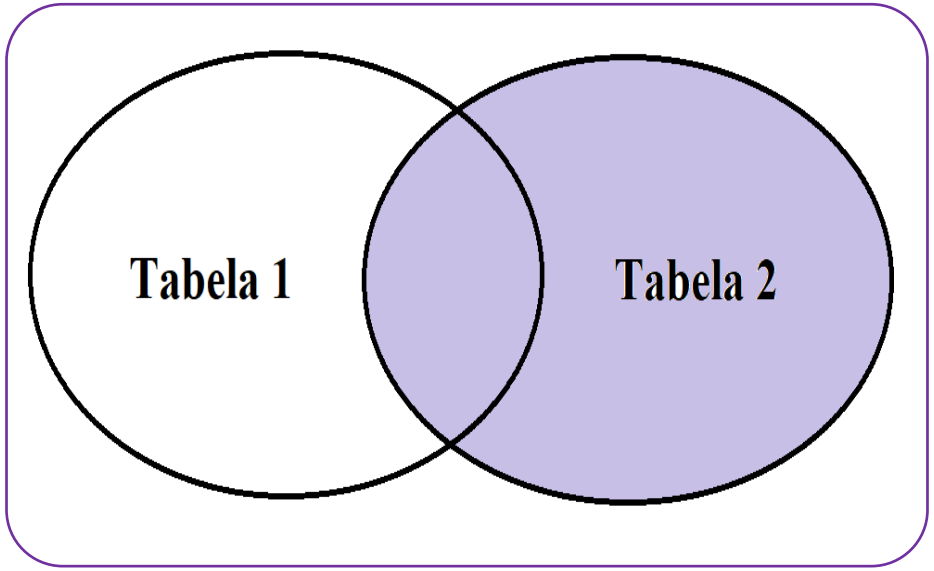
Tabela VEICULOS		
CPF	VEICULO	PLACA
111.111.111-11	Carro	SB-0001
NULL	Carro	SB-0002

```
SELECT * FROM PESSOAS LEFT JOIN VEICULOS ON PESSOAS.CPF = VEICULOS.CPF;
```

Tabela RESULTADO					
NOME	CPF	ESTADO	CPF	VEICULO	PLACA
Fernando	111.111.111-11	PR	111.111.111-11	Carro	SB-0001
Guilherme	222.222.222-22	SP	NULL	NULL	NULL

RIGHT JOIN (ou RIGHT OUTER JOIN)

A cláusula **RIGHT JOIN** retorna todos os registros da tabela da direita, e os registros relacionados da tabela da esquerda. Caso não haja valores relacionados na tabela da esquerda, os seus campos serão preenchidos com NULL.



Right Join (exemplo)

Tabela PESSOAS		
NOME	CPF	ESTADO
Fernando	111.111.111-11	PR
Guilherme	222.222.222-22	SP

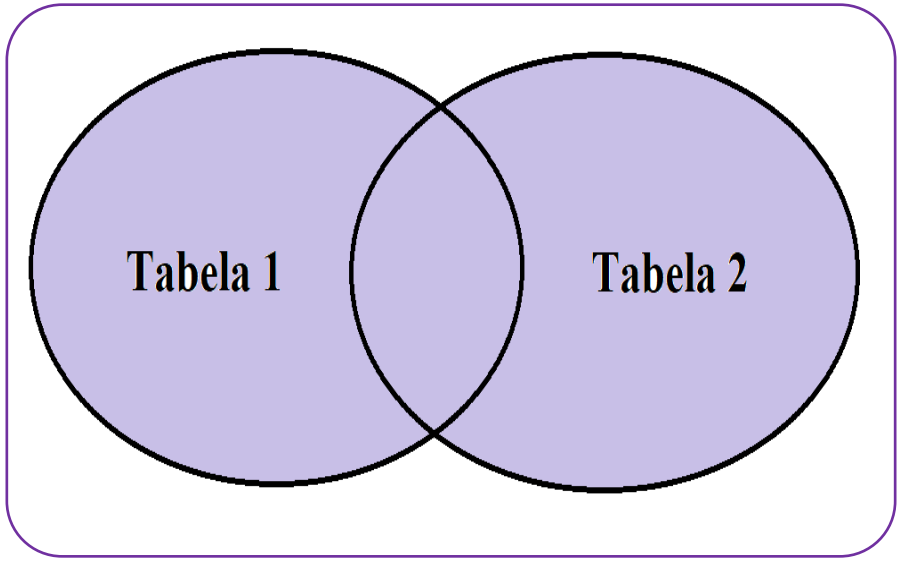
Tabela VEICULOS		
CPF	VEICULO	PLACA
111.111.111-11	Carro	SB-0001
NULL	Carro	SB-0002

```
SELECT * FROM PESSOAS RIGHT JOIN VEICULOS ON PESSOAS.CPF = VEICULOS.CPF;
```

Tabela RESULTADO					
NOME	CPF	ESTADO	CPF	VEICULO	PLACA
Fernando	111.111.111-11	PR	111.111.111-11	Carro	SB-0001
NULL	NULL	NULL	NULL	Carro	SB-0002

FULL OUTER JOIN

A cláusula **FULL OUTER JOIN** retorna todos os registros, relacionando aqueles que tiverem relação. Caso não haja valores relacionados na tabela da esquerda ou da direita, os seus campos serão preenchidos com NULL



Full Outer Join (exemplo)

Tabela PESSOAS		
NOME	CPF	ESTADO
Fernando	111.111.111-11	PR
Guilherme	222.222.222-22	SP

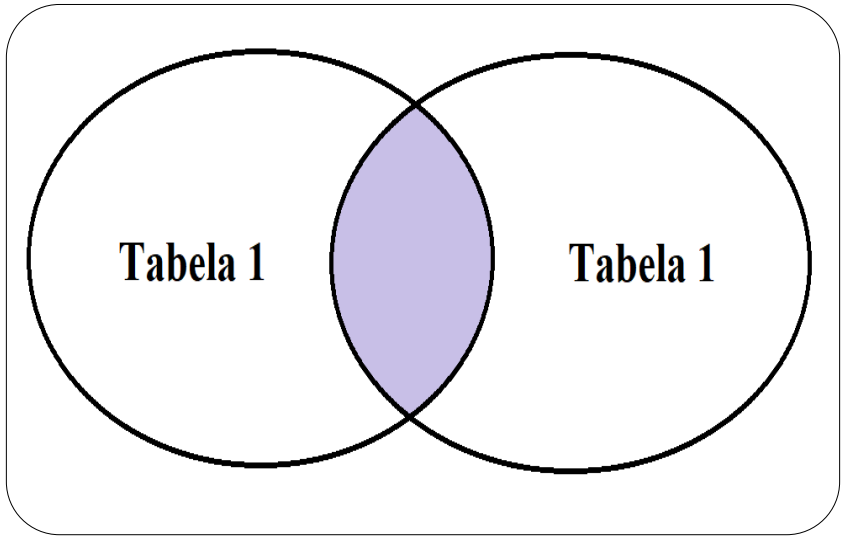
Tabela VEICULOS		
CPF	VEICULO	PLACA
111.111.111-11	Carro	SB-0001
NULL	Carro	SB-0002

```
SELECT * FROM PESSOAS FULL OUTER JOIN VEICULOS ON PESSOAS.CPF = VEICULOS.CPF;
```

Tabela RESULTADO					
NOME	CPF	ESTADO	CPF	VEICULO	PLACA
Fernando	111.111.111-11	PR	111.111.111-11	Carro	SB-0001
Guilherme	222.222.222-22	SP	NULL	NULL	NULL
NULL	NULL	NULL	NULL	Carro	SB-0002

SELF JOIN

A cláusula **SELF JOIN** é similar a um JOIN, contudo relaciona uma tabela com ela mesma.



Self Join (exemplo)

Tabela PESSOAS			
NOME	CPF	ESTADO	INDICADO
Fernando	111.111.111-11	PR	NULL
Guilherme	222.222.222-22	SP	111.111.111-11

```
SELECT A.NOME, B.NOME AS INDICADO_POR FROM PESSOAS A JOIN PESSOAS B ON
A.INDICADO = B.CPF;
```

Tabela RESULTADO	
NOME	INDICADO_POR
Guilherme	Fernando

(CESPE - 2019 - TJ-AM - Analista Judiciário - Analista de Sistemas)

A respeito de bancos de dados relacionais, julgue o item a seguir.

Em SQL, o comando RIGHT OUTER JOIN exibe a união entre duas tabelas, apresentando as linhas da segunda tabela que também existem na primeira tabela, descartando-se as demais situações.

(CESPE - 2018 - MPE-PI - Analista Ministerial - Tecnologia da Informação)

A respeito do Maven e do SQL, julgue o próximo item.

Situação hipotética: Determinada consulta foi feita mediante a execução da seguinte sequência de comandos SQL.

```
CREATE TABLE prova (codprocesso integer,
descricao varchar(10),
codprocessopai integer);

INSERT INTO prova VALUES (1, "A", 3);
INSERT INTO prova VALUES (2, "B", 1);
INSERT INTO prova VALUES (3, "C", null);
INSERT INTO prova VALUES (4, "D", 3);
```

```
SELECT p.descricao, pp.descricao as descricaopai
FROM prova p
INNER JOIN prova pp on p.codprocesso=pp.codprocessopai
WHERE pp.codprocessopai =3;
```

Assertiva: Nessa situação, o resultado da consulta será o que se segue.

descrição	descricaopai
C	A
C	D

(CESPE - 2018 - STJ - Técnico Judiciário - Desenvolvimento de Sistemas) Julgue o seguinte item, relativo a métrica de qualidade de software, JUnit, SQL, Delphi e desenvolvimento mobile.

A sentença SQL seguinte produzirá como resultado a lista de todos os funcionários de uma empresa. Para aqueles em que seja verdadeira a condição Funcionarios.CodigoDep = Departamentos.CodigoDep, será apresentado também o nome do departamento.

```
SELECT Funcionarios.Nome, Departamentos.NomeDep
FROM Funcionarios
INNER JOIN Departamentos ON
Funcionarios.CodigoDep =
Departamentos.CodigoDep
ORDER BY Funcionarios.Nome;
```

(**CESPE - 2015 - MEC - Arquiteto de Sistemas**) A manipulação dos dados armazenados em bancos de dados é realizada, muitas vezes, por linguagens com essa finalidade específica. Essas linguagens podem ser próprias do SGBD utilizado, embora muitas vezes representem a implementação de linguagem padrão de acesso a dados. Acerca desse assunto, julgue o seguinte item.

Na linguagem SQL, o comando FULL OUTER JOIN combina os resultados dos comandos LEFT JOIN e RIGHT JOIN.

(CESPE - 2015 - TRE-GO - Técnico Judiciário - Programação de Sistemas) Julgue o item a seguir, a respeito da linguagem SQL.

Um dos qualificadores para o OUTER JOIN previstos na sintaxe SQL ANSI é o FULL OUTER JOIN, em que são incluídas todas as linhas que satisfaçam a expressão tanto da primeira, quanto da segunda tabela.

(FCC - 2018 - SEFAZ-SC - Auditor-Fiscal da Receita Estadual - Auditoria e Fiscalização (Prova 3))

Atenção: As informações a seguir devem ser utilizadas para responder à questão.

O Auditor digitou um comando que exibiu os dados abaixo.

idPed	idItePed	idPro	nomePro	qtdItePed	PreUnitPro	Total
1	1	1	GPU NVIDIA 1080Ti	3	3456.00	10368.00
1	2	2	MEMORIA KINGSTON 16GB 1866MHZ	1	1078.45	1078.45
1	3	2	MEMORIA KINGSTON 16GB 1866MHZ	2	1078.45	2156.90
2	1	4	PROCESSADOR INTEL CORE I7 1151 3.7GHZ	5	2185.00	10925.00
2	2	3	HD SEAGATE 1TB SATA III	4	437.49	1749.96
3	1	6	PEN DRIVE SANDISK CRUZER32GB USB 2.0	4	55.00	220.00

Se o Auditor usasse join, o comando correto para exibir os dados seria

a) SELECT JOIN ip.idPed, ip.idItePed, ip.idPro, pro.nomePro, ip.qtdItePed, pro.PreUnitPro, (ip.qtdItePed * pro.PreUnitPro) as Total FROM Produto pro, ItemPedido ip WHERE ip.idPro = pro.idPro AND ip.idPed <=3 ORDER BY ip.idPed;

b) JOIN ip.idPed, ip.idItePed, ip.idPro, pro.nomePro, ip.qtdItePed, pro.PreUnitPro, (ip.qtdItePed * pro.PreUnitPro) as Total ON Produto pro, ItemPedido ip WHERE ip.idPro = pro.idPro AND ip.idPed <=3 ORDER BY ip.idPed;

c) SELECT idPed, idItePed, dPro, nomePro, qtdItePed, reUnitPro, (qtdItePed * PreUnitPro) as Total FROM Produto pro JOIN ItemPedido ip WHERE ItemProduto.idPro = Produto.idPro AND ItemProduto.idPed <=3 ORDER BY idPed;

d) JOIN ip.idPed, ip.idItePed, ip.idPro, pro.nomePro, ip.qtdItePed, pro.PreUnitPro, (ip.qtdItePed * pro.PreUnitPro) as Total FROM Produto pro, ItemPedido ip WHERE ip.idPro = pro.idPro AND ip.idPed <=3 ORDER BY ip.idPed;

e) SELECT ip.idPed, ip.idItePed, ip.idPro, pro.nomePro, ip.qtdItePed, pro.PreUnitPro, (ip.qtdItePed * pro.PreUnitPro) as Total FROM Produto pro JOIN ItemPedido ip ON ip.idPro = pro.idPro AND ip.idPed <=3 ORDER BY ip.idPed;

(FCC - 2015 - TRE-RR - Analista Judiciário - Análise de Sistemas) Considere a instrução SQL a seguir:

SELECT Clientes.NomeCliente, Pedidos.PedidoID FROM Clientes

...I....

ON Clientes.ClienteID=Pedidos.ClienteID

ORDER BY Clientes.NomeCliente;

Esta instrução seleciona todas as linhas de ambas as tabelas, desde que haja uma correspondência entre as colunas ClienteID. Se houver linhas na tabela Clientes que não tem correspondentes na tabela Pedidos, esses clientes não serão listados.

Para que a instrução dê o resultado descrito, a lacuna I deve ser preenchida com

a) INNER JOIN Pedidos

b) LEFT JOIN

c) RIGHT OUTER JOIN

d) FULL OUTER JOIN

e) LEFT OUTER JOIN

(FGV - 2018 - Banestes - Analista em Tecnologia da Informação - Suporte e Infraestrutura) Considere um banco de dados com duas tabelas, R e S, contendo 4 e 3 registros, respectivamente. Em R, os valores da coluna A são 1, 2, 5 e 7. Em S, os valores da coluna B são 2, 4 e 7.

Excetuando-se a linha de títulos, o número de linhas no resultado do comando SQL

select * from R full outer join S on A=B

é:

a) 3

b) 4

c) 5

d) 7

e) 12

(FGV - 2016 - SEE-PE - Professor de Desenvolvimento de Sistemas) Na questão, considere as tabelas T1 e T2 exibidas a seguir com suas respectivas instâncias.

T1

a	b
10	A
11	B
12	C
14	D

T2

a	c
10	7
11	9
15	NULL

De acordo com a definição e instâncias das tabelas T1 e T2 definidas acima, analise o comando SQL a seguir.

```
select t2.a, t2.c
from T1 left join T2
on t1.a = t2.a
```

Ao ser executado, esse comando produz um resultado contendo quatro linhas, além da linha de títulos.

Assinale a opção que indica o número de células que aparecem nesse resultado contendo "NULL".

a) 0

b) 1

c) 2

d) 3

e) 4

Questão de Concurso

a	b	a	c
10	A	10	7
11	B	11	9
12	C	NULL	NULL
14	D	NULL	NULL

a	c
10	7
11	9
NULL	NULL
NULL	NULL

Junções (Joins)

Prof. Ramon Souza