

Join - Junção

O que são?

É o recurso do banco de dados que permite a busca de dados provenientes de mais de uma tabela.

Considere as tabelas abaixo:

BAIRRO

COD_BAIRRO	NOME_BAIRRO
1	Betânia
2	Floresta
3	Centro

ALUNO

COD_ALUNO	NOME_ALUNO	COD_BAIRRO
1	Homer Simpson	1
2	Manda Chuva	1
3	Fred Flintstone	2
4	Pepe Legal	2
5	Docinho	2
6	Sininho	

INNER JOIN – esse é o padrão SQL usado para buscar registros de tabelas distintas que contenham um campo em comum.

```
SELECT campo1, campo2, ..., campoN
```

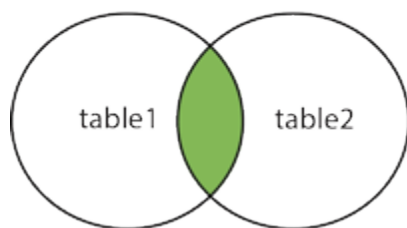
```
FROM tabela1
```

```
INNER JOIN tabela2
```

```
ON (tabela1.campo_comum=tabela2.campo_comum);
```

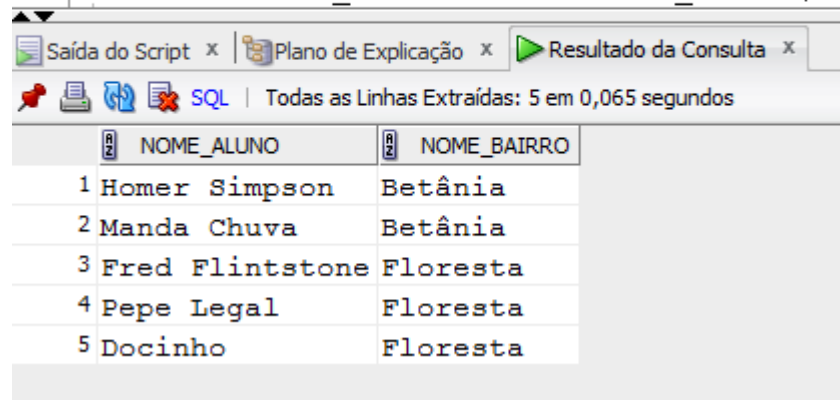
Esse comando equivale a intercessão de dois conjuntos:

INNER JOIN



Usando nossas tabelas teríamos:

```
21 SELECT nome_aluno, nome_bairro
22 FROM bairro
23 INNER JOIN aluno
24 ON bairro.cod_bairro = aluno.cod_bairro;
```



The screenshot shows a database query result window with the title 'Resultado da Consulta'. The query is an inner join between the 'ALUNO' and 'BAIRRO' tables. The result set contains 5 rows, showing the names of the students and their corresponding neighborhoods. The first two rows are from Betânia and the next three are from Floresta. The students 'Centro' and 'Sininho' from the ALUNO table are not shown because they do not have a matching record in the BAIRRO table.

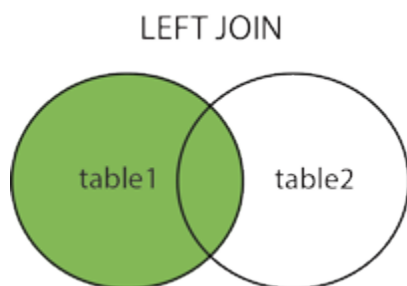
	NOME_ALUNO	NOME_BAIRRO
1	Homer Simpson	Betânia
2	Manda Chuva	Betânia
3	Fred Flintstone	Floresta
4	Pepe Legal	Floresta
5	Docinho	Floresta

Deve-se reparar que o registro Centro da tabela BAIRRO e o registro Sininho da tabela ALUNO que não tinham correspondência entre as tabelas não foram mostrados.

Left Join – Retorna todos os registros da tabela A ESQUERDA do comando, mesmo aqueles que não têm correspondência na outra tabela.

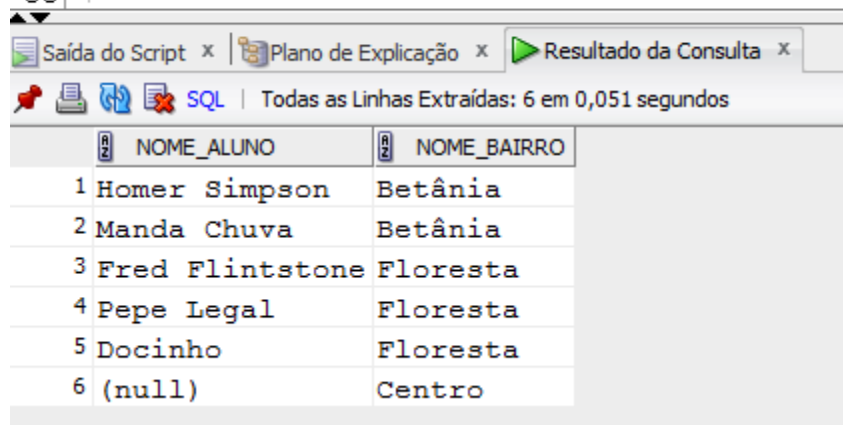
```
SELECT campo1, campo2, ..., campoN  
FROM tabela1  
LEFT JOIN tabela2  
ON (tabela1.campo_comum=tabela2.campo_comum);
```

Considerando os conjuntos teríamos:



Usando nossas tabelas teríamos:

```
26 SELECT nome_aluno, nome_bairro  
27 FROM bairro  
28 LEFT JOIN aluno  
29 ON bairro.cod_bairro = aluno.cod_bairro;  
30
```



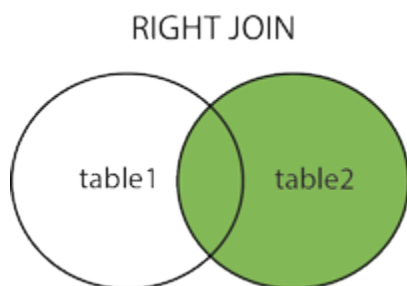
	NOME_ALUNO	NOME_BAIRRO
1	Homer Simpson	Betânia
2	Manda Chuva	Betânia
3	Fred Flintstone	Floresta
4	Pepe Legal	Floresta
5	Docinho	Floresta
6	(null)	Centro

Deve-se perceber que o bairro Centro aparece no resultado mesmo não havendo um aluno correspondente.

Right Join – Retorna todos os registros da tabela A DIREITA do comando, mesmo aqueles que não têm correspondência na outra tabela.

```
SELECT campo1, campo2, ..., campoN  
FROM tabela1  
RIGHT JOIN tabela2  
ON (tabela1.campo_comum=tabela2.campo_comum);
```

Considerando os conjuntos teríamos:



Usando nossas tabelas teríamos:

```
36 SELECT nome_aluno, nome_bairro  
37 FROM bairro  
38 RIGHT JOIN aluno  
39 ON bairro.cod_bairro = aluno.cod_bairro;  
40
```

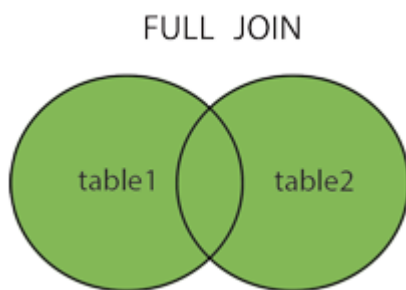
Saída do Script x Plano de Explicação x Resultado da Co... x	
SQL Todas as Linhas Extraídas: 6 em 0,045 segundos	
NOME_ALUNO	NOME_BAIRRO
1 Manda Chuva	Betânia
2 Homer Simpson	Betânia
3 Docinho	Floresta
4 Pepe Legal	Floresta
5 Fred Flintstone	Floresta
6 Sininho	(null)

Deve-se perceber que a aluna Sininho aparece no resultado mesmo não havendo um bairro correspondente.

Full Join – Retorna todos os registros das tabelas envolvidas mesmo aqueles que não têm correspondência na outra tabela.

```
SELECT campo1, campo2, ..., campoN  
FROM tabela1  
FULL JOIN tabela2  
ON (tabela1.campo_comum=tabela2.campo_comum);
```

Considerando os conjuntos teríamos:



Usando nossas tabelas teríamos:

```
46 SELECT nome_aluno, nome_bairro  
47 FROM aluno  
48 FULL OUTER JOIN bairro  
49 ON bairro.cod_bairro = aluno.cod_bairro;  
50
```

Saída do Script x Plano de Explicação x Resultado da Consulta x	
SQL Todas as Linhas Extraídas: 7 em 0,044 segundos	
NOME_ALUNO	NOME_BAIRRO
1 Homer Simpson	Betânia
2 Manda Chuva	Betânia
3 Fred Flintstone	Floresta
4 Pepe Legal	Floresta
5 Docinho	Floresta
6 Sininho	(null)
7 (null)	Centro

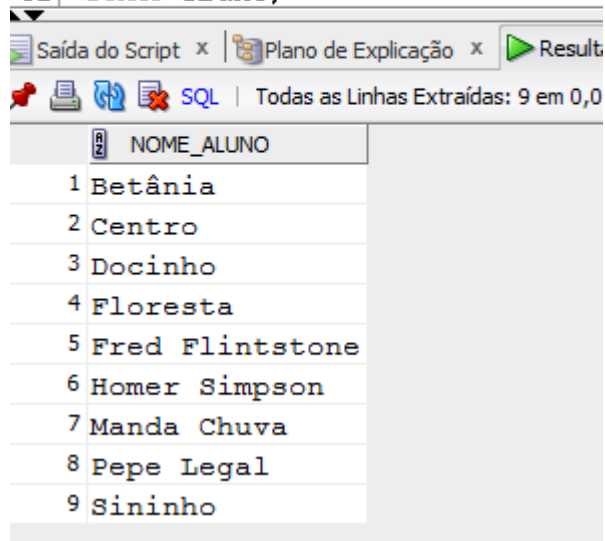
Deve-se perceber que a aluna Sininho e o bairro Centro aparecem no resultado mesmo não havendo um correspondente na outra tabela.

Union Join – Retorna todos os registros das tabelas envolvidas "concatenados". Deve-se observar que os campos envolvidos nas tabelas devem ser do mesmo tipo.

```
SELECT campo1, campo2, ..., campoN  
FROM tabela1  
UNION  
SELECT campo1, campo2, ..., campoN  
FROM tabela2
```

Usando nossas tabelas teríamos:

```
58 SELECT nome_bairro  
59 FROM bairro  
60 UNION  
61 SELECT nome_aluno  
62 FROM aluno;
```



The screenshot shows a database query result in a window titled 'Saída do Script x | Plano de Explicação x | Resultado'. The query is a UNION of two SELECT statements. The first SELECT statement selects 'nome_bairro' from the 'bairro' table, and the second SELECT statement selects 'nome_aluno' from the 'aluno' table. The result is a single column labeled 'NOME_ALUNO' containing 9 rows of data, which are the names of the neighborhoods concatenated with the names of the students.

NOME_ALUNO
1 Betânia
2 Centro
3 Docinho
4 Floresta
5 Fred Flintstone
6 Homer Simpson
7 Manda Chuva
8 Pepe Legal
9 Sininho

Deve-se perceber o resultado trouxe todos os registros de BAIRRO e depois todos os registros ALUNO.

OBSERVAÇÃO: o comando UNION pode ser:

UNION – retorna somente os valores distintos (que não repetem);

UNION ALL – retorna todos os valores mesmo que duplicados;