

Joins

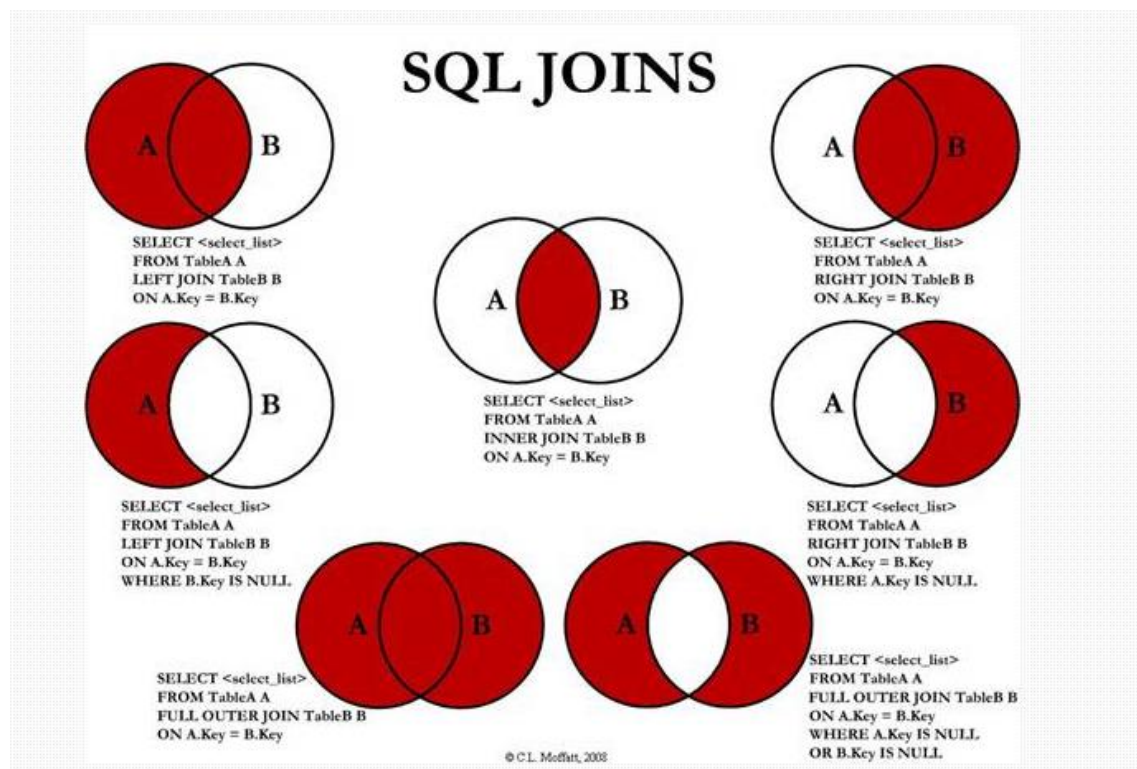
```
CREATE DATABASE JUNCAO;  
USE JUNCAO;
```

```
CREATE TABLE TABELA_A  
(NOME VARCHAR(50));
```

```
CREATE TABLE TABELA_B  
(NOME VARCHAR(50));
```

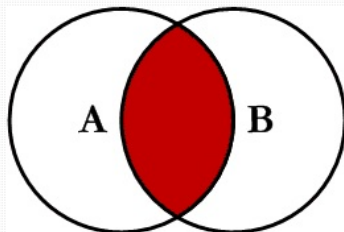
```
INSERT INTO TABELA_A VALUES ('Fernanda');  
INSERT INTO TABELA_A VALUES ('Zoraide');  
INSERT INTO TABELA_A VALUES ('Luiz');  
INSERT INTO TABELA_A VALUES ('Fernando');
```

```
INSERT INTO TABELA_B VALUES ('Carlos');  
INSERT INTO TABELA_B VALUES ('Evandro');  
INSERT INTO TABELA_B VALUES ('Luiz');  
INSERT INTO TABELA_B VALUES ('Fernando');
```



Inner Join

O Inner Join é o método de junção mais conhecido e, como ilustra a **Figura 2**, retorna os registros que são comuns às duas tabelas.



```
22 • SELECT a.Nome, b.Nome
23 FROM Tabela_A as A
24 INNER JOIN Tabela_B as B
25         on a.Nome = b.Nome;
```

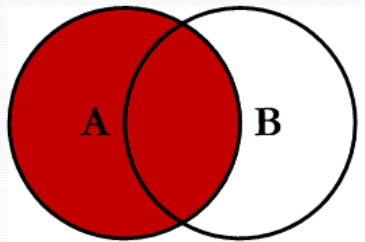
<

Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

	Nome	Nome
	Luiz	Luiz
	Fernando	Fernando

Left Join

O Left Join, cujo funcionamento é ilustrado na **Figura 4**, tem como resultado todos os registros que estão na tabela A (mesmo que não estejam na tabela B) e os registros da tabela B que são comuns à tabela A.

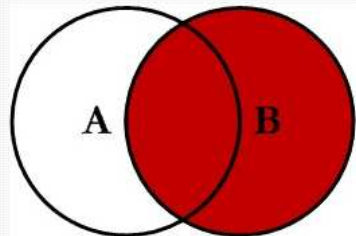


```
26
27 • SELECT a.Nome, b.Nome
28 FROM Tabela_A as A
29 LEFT JOIN Tabela_B as B
30      on a.Nome = b.Nome;
```

Nome	Nome
Luiz	Luiz
Fernando	Fernando
Fernanda	NULL
Zoraide	NULL

Right Join

Usando o Right Join, conforme mostra a **Figura 6**, 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.



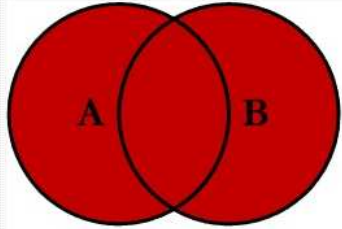
```
31
32 • SELECT a.Nome, b.Nome
33 FROM Tabela_A as A
34 RIGHT JOIN Tabela_B as B
35     on a.Nome = b.Nome
36
```

result Grid | Filter Rows: | Export: | Wrap Cell Content:

Nome	Nome
Luiz	Luiz
Fernando	Fernando
NULL	Carlos
NULL	Evandro

Outer Join

O Outer Join (também conhecido por Full Outer Join ou Full Join), conforme mostra a **Figura 8**, tem como resultado todos os registros que estão na tabela A e todos os registros da tabela B.



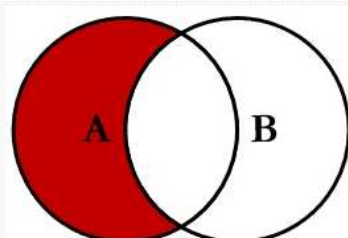
```
SELECT *  
FROM tabela_a  
LEFT OUTER JOIN tabela_b ON tabela_a.nome = tabela_b.nome
```

UNION

```
SELECT *  
FROM tabela_a  
RIGHT OUTER JOIN tabela_b ON tabela_a.nome = tabela_b.nome;
```


Left Excluding Join

Na **Figura 10** temos a representação gráfica do Left Excluding Join, que retorna como resultado todos os registros que estão na tabela A e que não estejam na tabela B.



```
52
53
54 • SELECT a.Nome, b.Nome
55 FROM Tabela_A as A
56 LEFT JOIN Tabela_B as B
57         on a.Nome = b.Nome
58 WHERE b.Nome is null
59
```

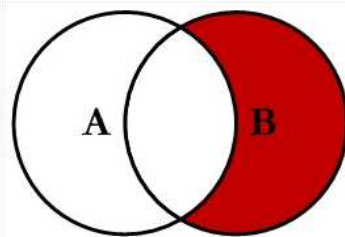
<

Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

	Nome	Nome
	Fernanda	NULL
	Zoraide	NULL

Right Excluding Join

O Right Excluding Join, como ilustra a **Figura 12**, retorna como resultado todos os registros que estão na tabela B e que não estejam na tabela A. Para vermos isso na prática, podemos executar os comandos da **Listagem 8**. Como resultado, teremos os mesmos registros apresentados na **Figura 13**.

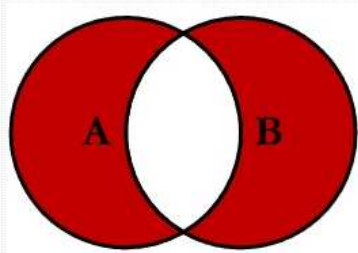


```
60  
61  
62 • SELECT a.Nome, b.Nome  
63 FROM Tabela_A as A  
64 RIGHT JOIN Tabela_B as B  
65     on a.Nome = b.Nome  
66 WHERE a.Nome is null  
67
```

result Grid			Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
	Nome	Nome			
	NULL	Carlos			
	NULL	Evandro			

Outer Excluding Join

Usando o Outer Excluding Join, conforme mostra a **Figura 14**, teremos como resultado todos os registros que estão na tabela B, mas que não estejam na tabela A, e todos os registros que estão na tabela A, mas que não estejam na tabela B.



Junção FULL OUTER JOIN NÃO EXISTENTE NO MYSQL.

```
SELECT a.Nome, b.Nome
FROM Tabela_A as A
FULL OUTER JOIN TabelaB as B
  on a.Nome = b.Nome
WHERE a.Nome is null or b.Nome is null;
```