

Join

Muitos desenvolvedores têm a dificuldade de saber qual resultado é retornado por cada join no SQL e, portanto, quando devem utilizar cada um. Para facilitar esse entendimento, a **Figura 1** traz uma representação gráfica, baseada na Teoria dos Conjuntos, muito conhecida na matemática. Nessa imagem, temos a representação de duas tabelas (A e B) e o resultado esperado por cada tipo de join (a área em vermelho representa os registros retornados pela consulta).

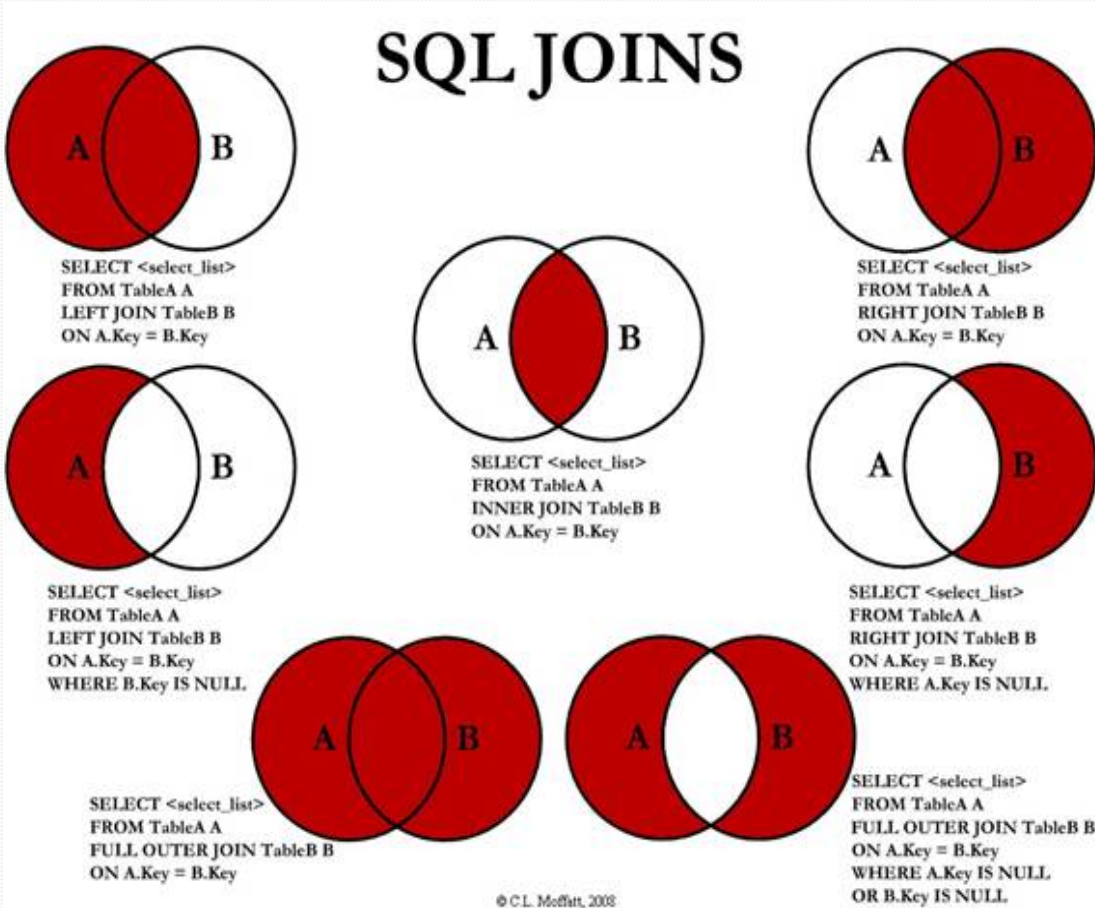


Figura 1. Representação gráfica dos joins

Preparando o ambiente de testes

Para demonstrar o funcionamento dos métodos de junção (joins), precisaremos criar duas tabelas entre as quais deve haver algum relacionamento para que possamos "cruzar" os dados. Como o objetivo aqui não é concentrar em boas práticas, modelagem ou normalização, criaremos apenas duas tabelas contendo uma coluna `Nome`, que será comum entre elas. O script da **Listagem 1** cria essa estrutura.

```
CREATE TABLE TabelaA(  
    Nome varchar(50) NULL  
)  
  
GO  
  
CREATE TABLE TabelaB(  
    Nome varchar(50) NULL  
)
```

Listagem 1. Criando as tabelas para teste

Em seguida, precisaremos adicionar nas tabelas recém criadas alguns dados que nos permitam colocar à prova as junções. Sendo assim, vamos inserir, com o script da **Listagem 2** alguns registros de forma que haja nomes que estão presentes apenas em uma tabela, e também nomes que sejam comuns às duas.

```
INSERT INTO TabelaA VALUES('Fernanda')
INSERT INTO TabelaA VALUES('Josefa')
INSERT INTO TabelaA VALUES('Luiz')
INSERT INTO TabelaA VALUES('Fernando')

INSERT INTO TabelaB VALUES('Carlos')
INSERT INTO TabelaB VALUES('Manoel')
INSERT INTO TabelaB VALUES('Luiz')
INSERT INTO TabelaB VALUES('Fernando')
```

Listagem 2. Inserindo registros para testes

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.

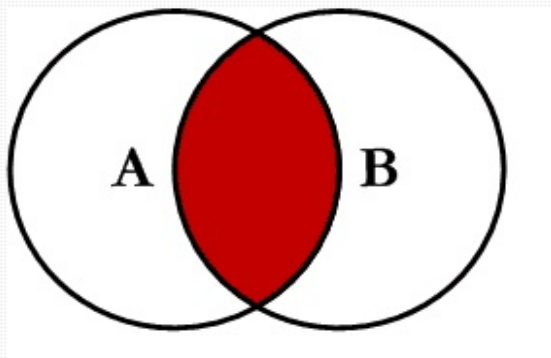
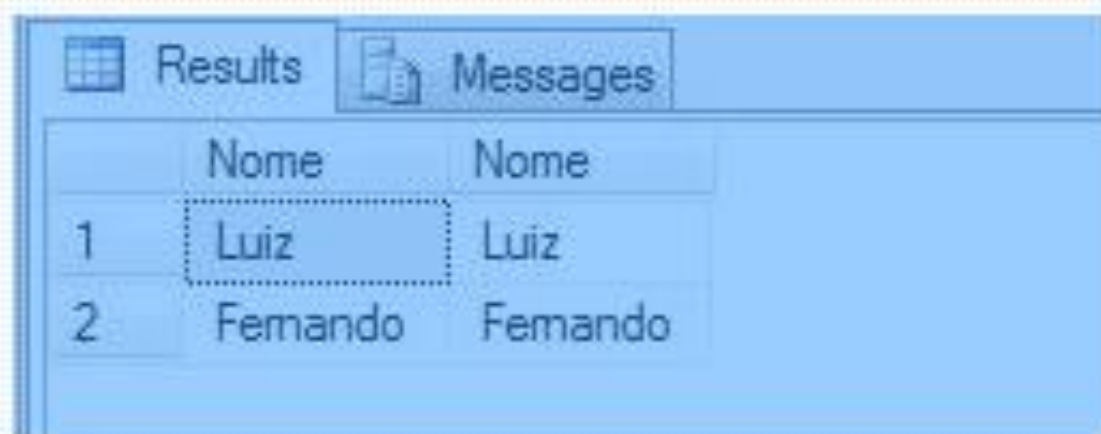


Figura 2. Representação do Inner Join

Na **Listagem 3** temos um exemplo de consulta com esse tipo de join, e na **Figura 3** podemos ver seu resultado.

```
SELECT a.Nome, b.Nome
FROM TabelaA as A
INNER JOIN TabelaB as B
      on a.Nome = b.Nome
```

Listagem 3. Usando o Inner Join



The image shows a screenshot of a database application window. At the top, there are two tabs: 'Results' (active) and 'Messages'. Below the tabs is a table with two columns, both labeled 'Nome'. The table contains two rows of data. The first row has the value 'Luiz' in both columns, and the second row has the value 'Femando' in both columns. The cell containing 'Luiz' in the first column is highlighted with a dashed border.

	Nome	Nome
1	Luiz	Luiz
2	Femando	Femando

Figura 3. Resultado do Inner Join

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.

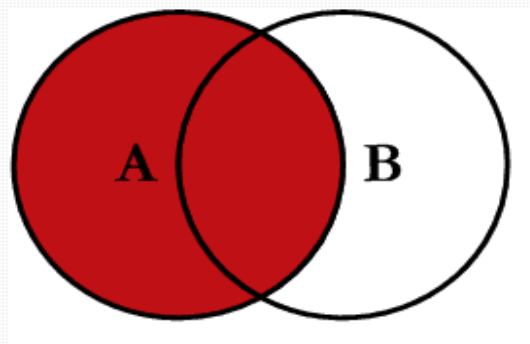
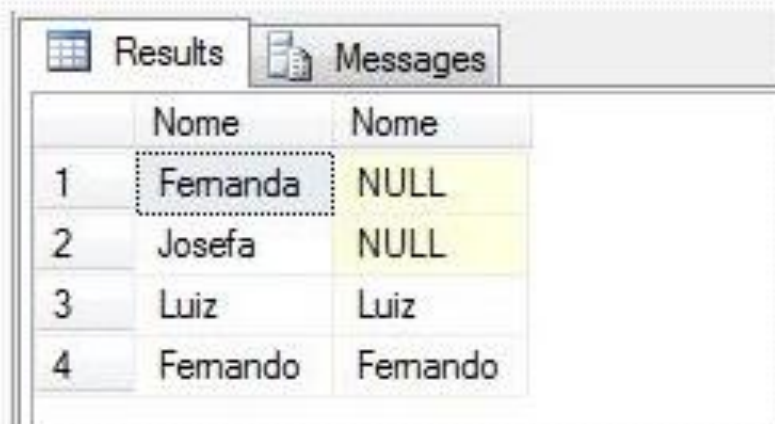


Figura 4. Representação do Left Join

Para compreender melhor seu uso, temos um exemplo na **Listagem 4**, cujo resultado é apresentado em seguida na **Figura 5**.

```
SELECT a.Nome, b.Nome
FROM TabelaA as A
LEFT JOIN TabelaB as B
      on a.Nome = b.Nome
```

Listagem 4. Usando o Left Join



	Nome	Nome
1	Femanda	NULL
2	Josefa	NULL
3	Luiz	Luiz
4	Femando	Femando

Figura 5. Resultado do Left Join

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.

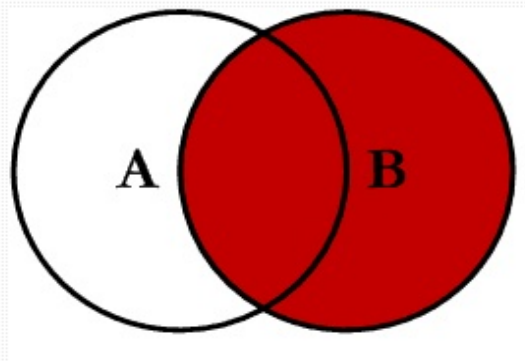


Figura 6. Representação do Right Join

Na **Listagem 5** temos um exemplo desse tipo de consulta, e na **Figura 7** vemos seu resultado.

```
SELECT a.Nome, b.Nome
FROM TabelaA as A
RIGHT JOIN TabelaB as B
on a.Nome = b.Nome
```

Listagem 5. Usando o Right Join

	Results	Messages
	Nome	Nome
1	NULL	Carlos
2	NULL	Manoel
3	Luiz	Luiz
4	Fernando	Fernando

Figura 7. Resultado do Right Join

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.

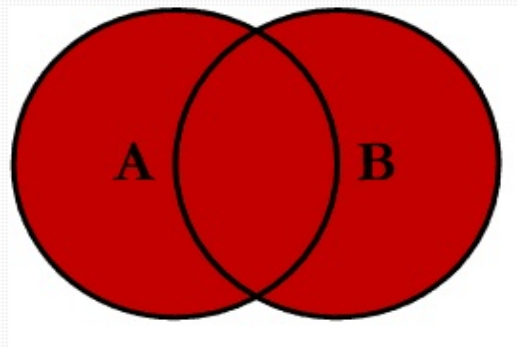


Figura 8. Representação do Outer Join

Para obter esse resultado, devemos executar a consulta seguinte a estrutura que é demonstrada na **Listagem 6**, e cujo retorno é apresentado na **Figura 9**.

```
SELECT a.Nome, b.Nome
FROM TabelaA as A
FULL OUTER JOIN TabelaB as B
    on a.Nome = b.Nome
```

Listagem 6. Usando o Outer Join

Results		Messages
	Nome	Nome
1	Fernanda	NULL
2	Josefa	NULL
3	Luiz	Luiz
4	Fernando	Fernando
5	NULL	Carlos
6	NULL	Manoel

Figura 9. Resultado do Outer Join

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.

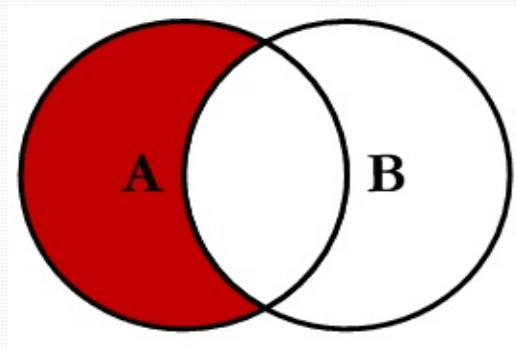


Figura 10. Representação do Left Excluding Join

Os comandos desse join podem ser vistos na **Listagem 7**, e seu resultado é apresentado na **Figura 11**.

```
SELECT a.Nome, b.Nome
FROM TabelaA as A
LEFT JOIN TabelaB as B
      on a.Nome = b.Nome
WHERE b.Nome is null
```

Listagem 7. Usando Left Excluding Join

Results		Messages
	Nome	Nome
1	Fernanda	NULL
2	Josefa	NULL

Figura 11. Resultado do Left Excluding Join

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**.

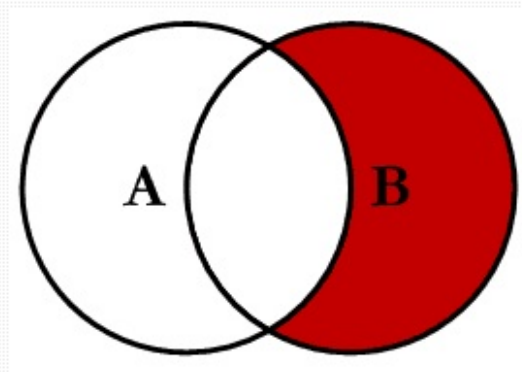


Figura 12. Representação do Right Excluding Join

```
SELECT a.Nome, b.Nome
FROM TabelaA as A
RIGHT JOIN TabelaB as B
    on a.Nome = b.Nome
WHERE a.Nome is null
```

Listagem 8. Utilizando Right Excluding Join

Results		Messages
	Nome	Nome
1	NULL	Carlos
2	NULL	Manoel

Figura 13. titulo

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.

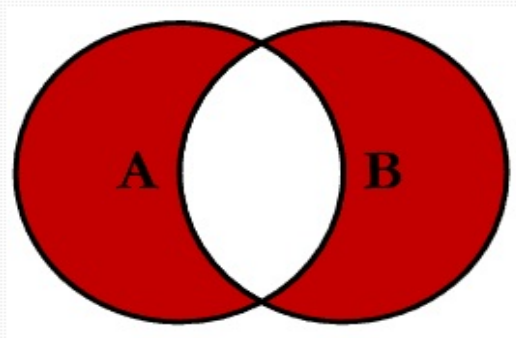


Figura 14. Representação do Outer Excluding Join

O código da **Listagem 9** mostra a sintaxe necessária para executar esse tipo de consulta. Nesse caso o termo `OUTER` é optativo e, se removido, deixando apenas `FULL JOIN`, o resultado será o mesmo.

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

Listagem 9. Usando Outer Excluding Join

Results		Messages
	Nome	Nome
1	Fernanda	NULL
2	Josefa	NULL
3	NULL	Carlos
4	NULL	Manoel

Figura 15. Resultado do Outer Excluding Join