



SQL Server Desenvolvedor





Recuperando informações de mais de uma tabela ao mesmo tempo

```
SELECT
    Tab1.Campo1, Tab2.Campo2, Tab1.CampoN, Tab2.Campo1, Tab2.Campo2, Tab2.CampoN
FROM
    Tabela1 AS Tab1,
    Tabela2 AS Tab2
WHERE
    Tab1.Campo1 = Tab2.Campo1
```



Recuperando informações de mais de uma tabela ao mesmo tempo

```
SELECT
    F.Id AS FuncionarioId,
    F.Nome,
    F.Sobrenome,
    P.DataInicial,
    P.DataFinal
FROM
    Funcionarios AS F,
    PontosDeAcesso AS P
WHERE
    F.Id = P.FuncionarioId
```



Junções (joins)

O SQL possui um recurso chamado junção (join) para facilitar e melhorar a sintaxe de consultas que precisam relacionar várias tabelas. Com esse recurso, não precisamos declarar a segunda tabela na cláusula FROM nem escrever a condição no WHERE.

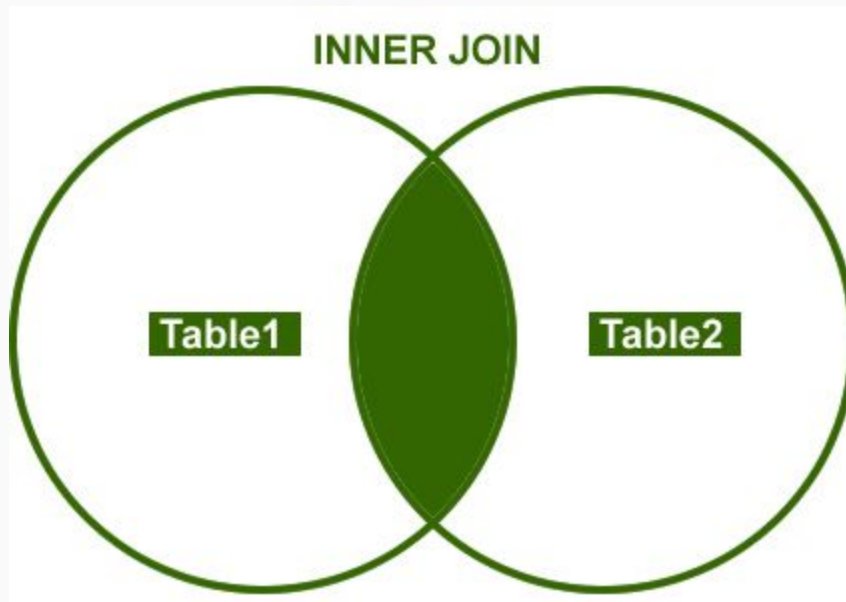


INNER JOIN

O INNER JOIN é o tipo de junção mais usado no dia a dia. Ele retorna apenas os dados que estão relacionados entre a tabela declarada no FROM e a declarada no JOIN. Na teoria de conjuntos, o INNER JOIN é representado pela intersecção. Veja a imagem abaixo para compreendê-lo melhor:



INNER JOIN





INNER JOIN



```
SELECT * FROM Tabela1 AS Apelido1  
INNER JOIN Tabela2 AS Apelido2  
ON Apelido1.Campo1 = Apelido2.Campo2
```



INNER JOIN

```
SELECT
    F.Id AS FuncionarioId,
    F.Nome,
    F.Sobrenome,
    P.DataInicial,
    P.DataFinal
FROM
    Funcionarios AS F
INNER JOIN
    PontosDeAcesso AS P
ON
    F.Id = P.FuncionarioId
```

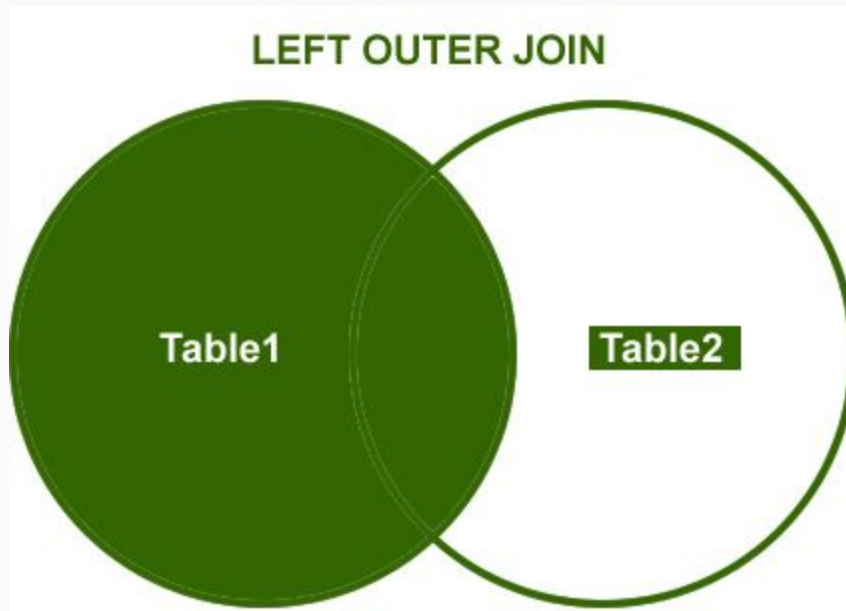



LEFT e RIGHT JOIN

A junção INNER JOIN realiza a união retornando apenas os registros que possuem dados relacionados, mas existem casos onde é necessário recuperar dados de registros que não estão relacionados. Por exemplo, em um relacionamento, pode ser necessário retornar todos os dados de um lado da relação, algo que não pode ser obtido com o INNER JOIN. Para este tipo de cenário existem mais dois tipos de junção: LEFT JOIN e RIGHT JOIN.



LEFT e RIGHT JOIN





LEFT e RIGHT JOIN



```
SELECT * FROM Tabela1 AS Apelido1  
LEFT JOIN Tabela2 AS Apelido2  
ON Apelido1.Campo1 = Apelido2.Campo2
```



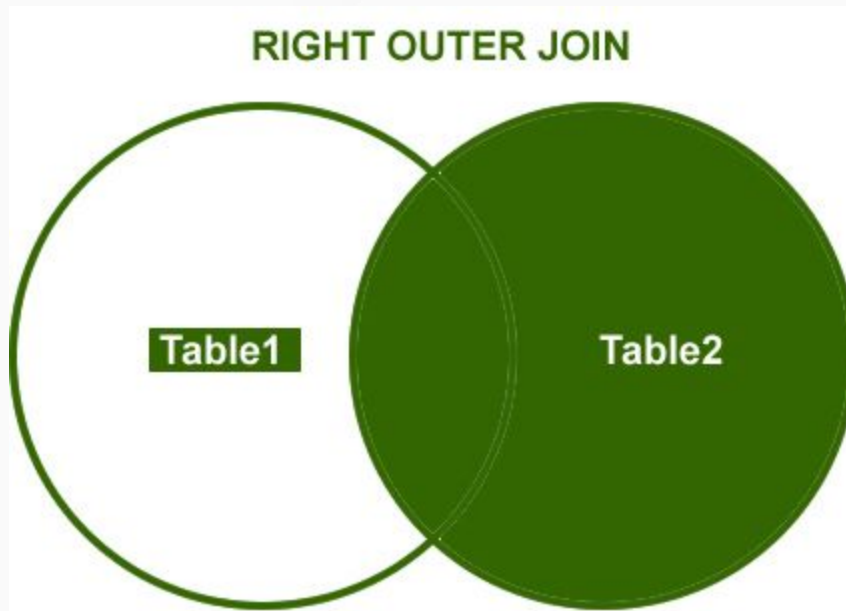
LEFT e RIGHT JOIN



```
SELECT
    F.Id AS FuncionarioId,
    F.Nome,
    F.Sobrenome,
    P.DataInicial,
    P.DataFinal
FROM
    Funcionarios AS F
LEFT JOIN
    PontosDeAcesso AS P
ON
    F.Id = P.FuncionarioId
```



LEFT e RIGHT JOIN





LEFT e RIGHT JOIN



```
SELECT * FROM Tabela1 AS Apelido1  
RIGHT JOIN Tabela2 AS Apelido2  
ON Apelido1.Campo1 = Apelido2.Campo2
```



LEFT e RIGHT JOIN



```
SELECT
    F.Id AS FuncionarioId,
    F.Nome,
    F.Sobrenome,
    P.DataInicial,
    P.DataFinal
FROM
    Funcionarios AS F
RIGHT JOIN
    PontosDeAcesso AS P
ON
    F.Id = P.FuncionarioId;
```



Por que utilizar JOINS?

- **Clareza:** A sintaxe de **JOIN** torna a intenção da consulta mais clara. Usar **JOIN** indica explicitamente que você está combinando duas tabelas, e o tipo de **JOIN** (por exemplo, **INNER JOIN**) especifica como os dados devem ser combinados. Isso facilita a leitura e manutenção do código.
- **Legibilidade:** Com **JOINS** explícitos, a consulta é mais organizada e fácil de entender, especialmente em queries complexas com várias tabelas e condições.
- **Melhor Performance e Otimização:** A maioria dos bancos de dados é otimizada para lidar com **JOINS** explícitos. Em alguns casos, o otimizador de consultas do banco de dados pode gerar planos de execução melhores para **JOINS**, resultando em um desempenho superior.
- **Evita Ambiguidade:** Em consultas mais complexas, especialmente quando há várias tabelas e condições, **JOINS** explícitos ajudam a evitar erros e ambiguidade no código.