

Esse material foi produzido por Maria Luiza Mondelli e está licenciado com a licença Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)

Para mais informações sobre a licença, acesse: <https://br.creativecommons.org/licencas/>



Atribuição-NãoComercial

CC BY-NC

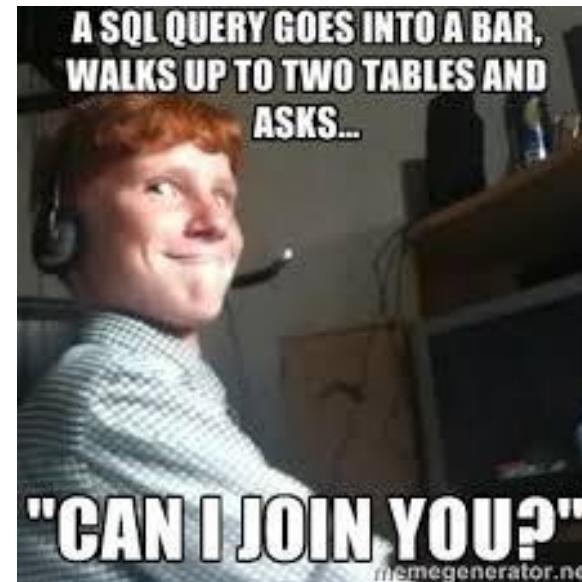
Esta licença permite que outros remixem, adaptem e criem a partir do seu trabalho para fins não comerciais, e embora os novos trabalhos tenham de lhe atribuir o devido crédito e não possam ser usados para fins comerciais, os usuários não têm de licenciar esses trabalhos derivados sob os mesmos termos.



SQL

Junções

- Em um banco de dados já vimos que podemos ter duas ou mais tabelas relacionadas entre si
- Muitas vezes as consultas precisam **recuperar** dados de **tabelas diferentes**
- Por isso, precisamos **definir critérios de junções** para obter estes dados



* Lembra da ideia de que uma consulta pode envolver mais de uma tabela?

E de como podemos pensar nas junções de tabelas como um **caminho a ser percorrido**?

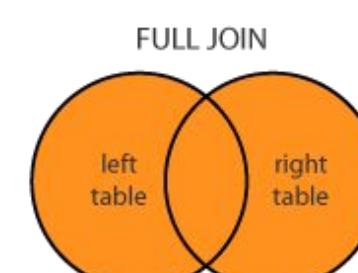
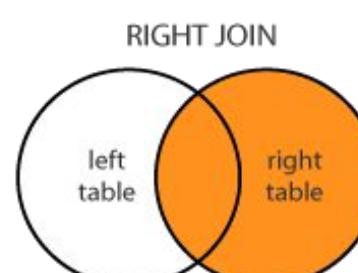
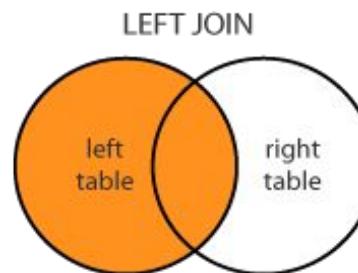
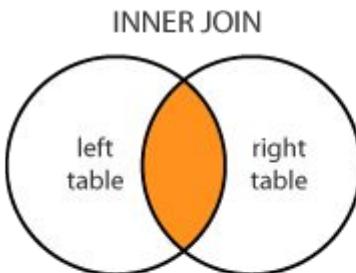


SQL

Junções

- Em um banco de dados já vimos que podemos ter duas ou mais tabelas relacionadas entre si
- Muitas vezes as consultas precisam **recuperar** dados de **tabelas diferentes**
- Por isso, precisamos definir **critérios** de **junções** para obter estes dados

Tipos de junções possíveis:

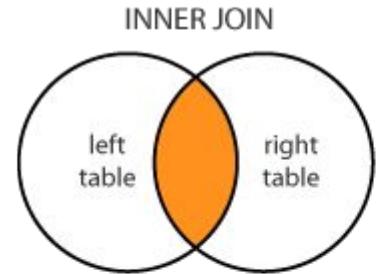


```
SELECT      nome_coluna(s)      FROM      tabela_1  
RIGHT      |      LEFT      |      FULL      JOIN      tabela_2  
          ON tabela_1.nome_coluna = tabela_2.nome_coluna;
```



SQL

Junções



INNER JOIN

Retorna todas as linhas quando há pelo menos um registro correspondente em ambas as tabelas.

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	1	1996-09-19
10310	1	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

```

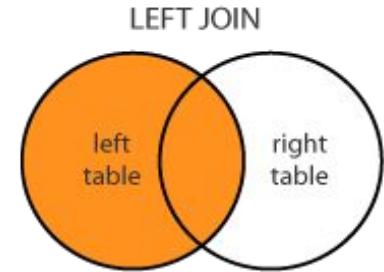
SELECT Orders.OrderID,
       Customers.CustomerName,
       Orders.OrderDate
  FROM Orders
 INNER JOIN Customers
    ON Orders.CustomerID=Customers.CustomerID
  
```

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados	1996-09-18
10309	Alfreds Futterkiste	1996-09-19
10310	Alfreds Futterkiste	1996-09-20



SQL

Junções



LEFT JOIN

Retorna todas as linhas da tabela à esquerda e os registros correspondentes da tabela direita

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	1	1996-09-19
10310	1	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

```

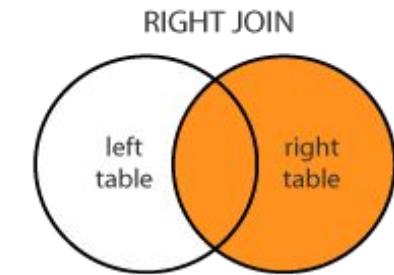
SELECT Orders.OrderID,
       Customers.CustomerName,
       Orders.OrderDate
  FROM Orders
 LEFT JOIN Customers
        ON Orders.CustomerID=Customers.CustomerID
    
```

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados	1996-09-18
10309	Alfreds Futterkiste	1996-09-19
10310	Alfreds Futterkiste	1996-09-20



SQL

Junções



RIGHT JOIN

Retorna todas as linhas da tabela à direita, e os registros correspondentes da tabela esquerda..

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	1	1996-09-19
10310	1	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

```

SELECT Orders.OrderID,
       Customers.CustomerName,
       Orders.OrderDate
  FROM Orders
 RIGHT JOIN Customers
    ON Orders.CustomerID=Customers.CustomerID
  
```

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados	1996-09-18
10309	Alfreds Futterkiste	1996-09-19
10310	Alfreds Futterkiste	1996-09-20
	Antonio Moreno Taquería	



SQL

Subconsulta

- É uma instrução SELECT dentro de outra instrução SQL, que pode ser:
 - SELECT, INSERT, UPDATE ou DELETE
- Pode ser extremamente útil ou complicar consultas que poderiam ser facilmente resolvidas apenas com junções, por exemplo.
- Podemos pensar na declaração de subconsultas como uma 'combinação de blocos', exemplo:

```
select .... from .... where ... in
```

Bloco externo

```
    select .... from .... where ....
```

Bloco interno



SQL

Formas de usar subconsultas

1. Como nova coluna da consulta

```
externo   SELECT P.titulo,  
interno   (SELECT COUNT(C.id_projeto)  
           FROM comentarios C  
           WHERE C.id_projeto = P.id ) AS qtd_com  
           FROM projetos P  
           GROUP BY P.id
```



SQL

Formas de usar subconsultas

2. Como filtro de uma consulta externa (usando operadores de comparação, in ou exists)

```
externo   SELECT P.id,  
              P.titulo,  
              P.data  
        FROM projetos P  
 WHERE P.id IN  
           (SELECT C.id_projeto  
              FROM comentarios C  
             WHERE P.id = C.id_projeto  
           );  
  
interno
```



SQL

Formas de usar subconsultas

3. Como fonte de dados de uma consulta externa (ou seja, fazendo o papel de uma nova tabela)

```

externo
SELECT F.titulo,
       F.Quantidade_Comentarios
  FROM
    internos
      (SELECT P.titulo,
              (SELECT COUNT(C.id_projeto)
                 FROM comentarios C
                WHERE C.id_projeto = P.id ) AS Quantidade_Comentarios
             FROM projetos P
           ) as F
 WHERE F.Quantidade_Comentarios > 2
  
```

Você pode fazer uma infinidade de subconsultas.
 Só precisa avaliar: é realmente necessário?
 Vai ajudar ou complicar mais?



SQL

Subconsultas com outras instruções SQL

INSERT

```
externo   INSERT INTO EMPREGADOS(CODIGO, NOME, SALARIO, SECAO)
interno    SELECT CODIGO, NOME, SALARIO, SECAO
                  FROM EMPREGADOS_FILIAL
                     WHERE DEPARTAMENTO = 2
```

UPDATE

```
externo   UPDATE EMPREGADOS
              SET SALARIO = SALARIO * 1.1
              WHERE SALARIO = (
interno      SELECT MIN(SALARIO)
                      FROM EMPREGADOS
                    )
```

DELETE

```
externo   DELETE FROM EMPREGADOS
              WHERE SALARIO = (
interno      SELECT MIN(SALARIO)
                      FROM EMPREGADOS
                    )
```

