



CENTRO UNIVERSITÁRIO SENAC

Disciplina : Banco de Dados

Prof.Me Rômulo Maia

Email :

romulo.fsmaia@sp.senac.br

Group By

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
ORDER BY column_name(s);
```

Criar o Ambiente

```
CREATE TABLE Vendas (  
    ID int Primary Key,  
    Nome_Vendedor Varchar(20),  
    Quantidade Int,  
    Produto Varchar(20),  
    Cidade Varchar(20) );
```

```
INSERT INTO Vendas (ID, Nome_Vendedor, Quantidade, Produto,  
Cidade) VALUES  
(10,'Jorge',1400,'Mouse','São Paulo'),  
(12,'Tatiana',1220,'Teclado','São Paulo'),  
(14,'Ana',1700,'Teclado','Rio de Janeiro'),  
(15,'Rita',2120,'Webcam','Recife'),  
(18,'Marcos',980,'Mouse','São Paulo'),  
(19,'Carla',1120,'Webcam','Recife'),  
(22,'Roberto',3145,'Mouse','São Paulo');
```

Usando Group By

```
SELECT SUM(Quantidade) As TotalMouses FROM Vendas  
WHERE Produto = 'Mouse';
```

TotalMouses
5525

```
SELECT Cidade, SUM(Quantidade) As Total FROM Vendas  
GROUP BY Cidade;
```

Cidade	Total
Recife	3240
Rio de Janeiro	1700
São Paulo	6745

```
SELECT Cidade, COUNT(*) As Total FROM Vendas GROUP BY Cidade;
```

Cidade	Total
Recife	2
Rio de Janeiro	1
São Paulo	4

```
SELECT Nome_Vendedor, SUM(Quantidade) FROM Vendas  
GROUP BY Nome_Vendedor;
```

Nome_Vendedor	SUM(Quantidade)
Ana	1700
Carla	1120
Jorge	1400
Marcos	980
Rita	2120
Roberto	3145
Tatiana	1220

SQL JOIN

Preparando o Bando de dados, criando as tabelas abaixo :

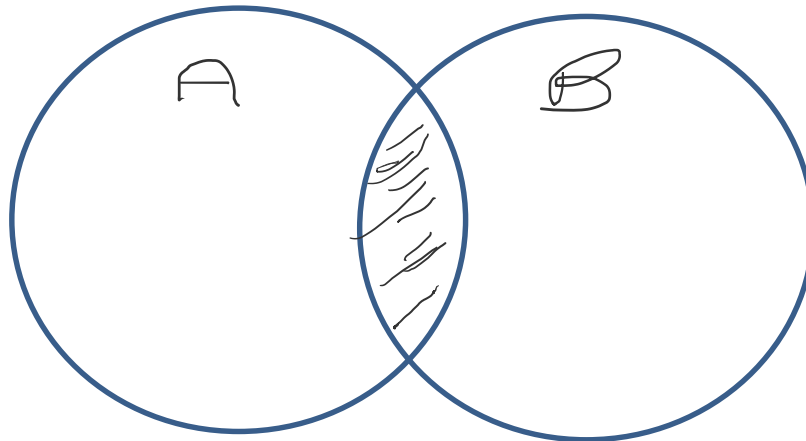
Arquivo1	
Codigo	Nome (Varchar(50))
1	Fernanda
2	Josefa
3	Luiz
4	Fernando
5	Romulo

Arquivo2	
Codigo	Nome
1	Carlos
2	Manoel
3	Luiz
4	Fernando
5	Romulo

Inner Join

O Inner Join é o método de junção mais conhecido e retorna os registros que são comuns às duas tabelas.

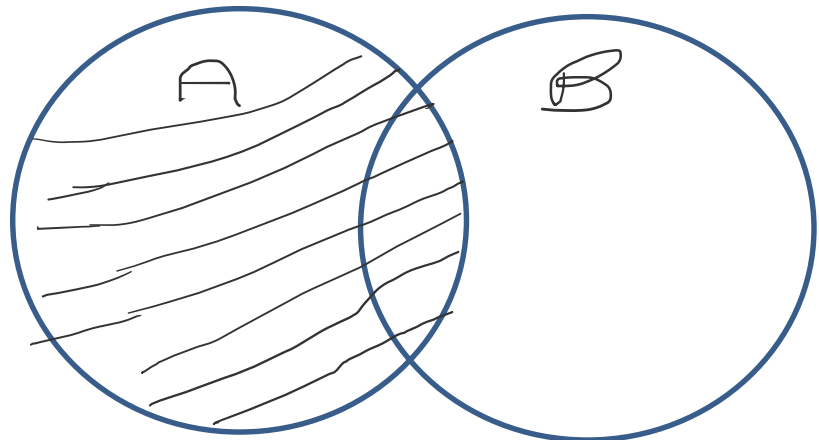
```
SELECT a.Nome, b.Nome  
FROM Arquivo1 as a  
INNER JOIN Arquivo2 as b  
on a.Nome = b.Nome
```



Left Join

O Left Join, 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.

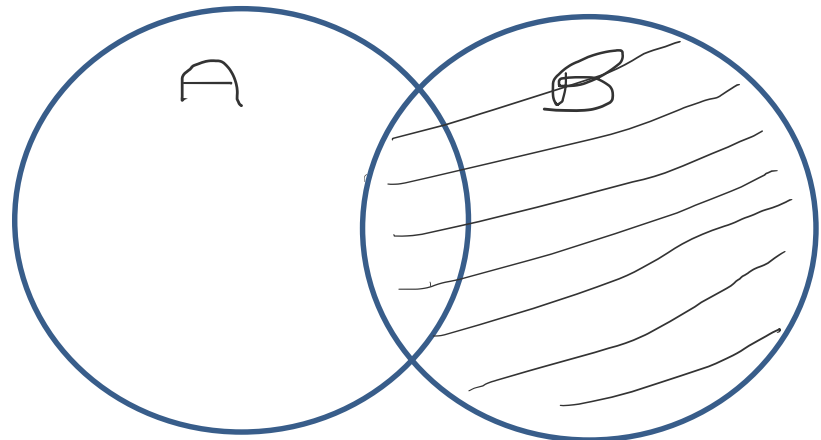
```
SELECT a.Nome, b.Nome  
FROM Arquivo1 as a  
LEFT JOIN Arquivo2 as b  
on a.Nome = b.Nome
```



Right Join

Usando o Right Join, 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.

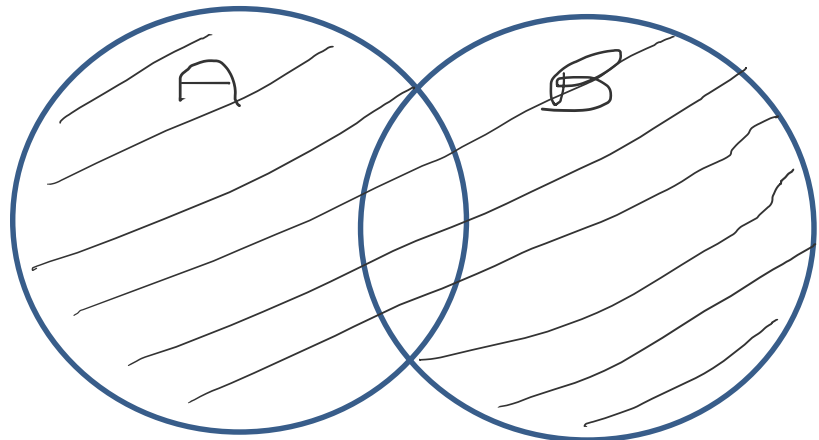
```
SELECT a.Nome, b.Nome  
FROM Arquivo1 as a  
RIGHT JOIN Arquivo2 as b  
on a.Nome = b.Nome
```



Outer Join

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

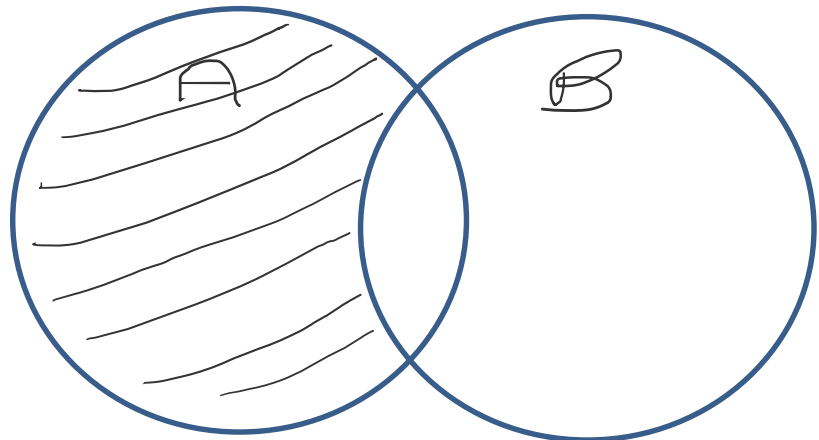
```
SELECT a.Nome, b.Nome  
FROM Arquivo1 as a  
Left/Right OUTER JOIN Arquivo2 as b  
on a.Nome = b.Nome
```



Left Excluding Join

Retorna como resultado todos os registros que estão na tabela A e que não estejam na tabela B.

```
SELECT a.Nome, b.Nome  
FROM Arquivo1 as a  
LEFT JOIN Arquivo2 as b  
      on a.Nome = b.Nome  
WHERE b.Nome is null
```



```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

```
SELECT  
    YEAR(orderDate) AS year,  
    SUM(quantityOrdered * priceEach) AS total  
FROM  
    orders  
    INNER JOIN  
    orderdetails USING (orderNumber)  
WHERE  
    status = 'Shipped'  
GROUP BY YEAR(orderDate);
```



```
SELECT
    YEAR(orderDate) AS year,
    SUM(quantityOrdered * priceEach) AS total
FROM
    orders
    INNER JOIN
    orderdetails USING (orderNumber)
WHERE
    status = 'Shipped'
GROUP BY year
HAVING year > 2003;
```