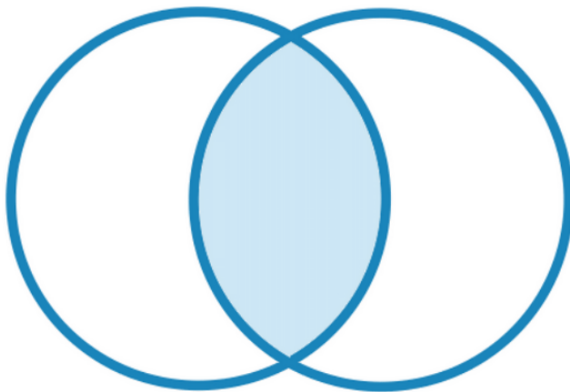


Tipos de Joins - Cheatsheet

Inner Join

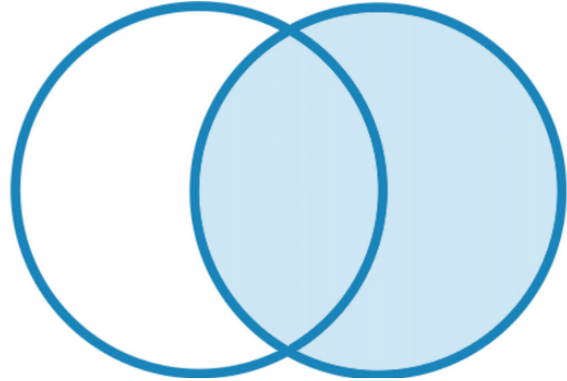
Cada registro de una tabla A es combinado con cada registro de la tabla B pero sólo si se satisface la condición que se especifique



```
SELECT * FROM TablaA AS A INNER JOIN  
TablaB AS B ON A.ColumnaID =  
B.ColumnaID
```

Right Join

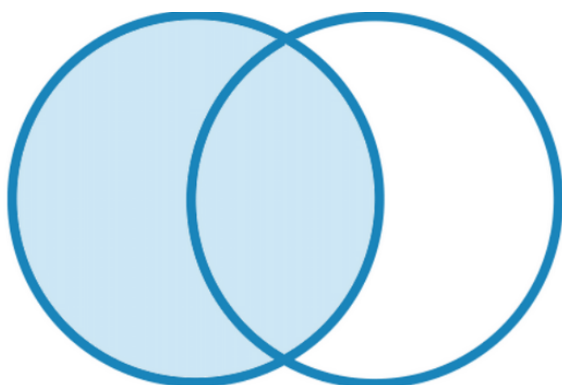
Contiene los registros de la tabla de la derecha, incluso si no existiera un registro correspondiente en la tabla de la izquierda para uno de la derecha. Los datos de la tabla de la izquierda que no pueda emparentar serán completados con NULL.



```
SELECT * FROM TablaA AS A RIGHT  
JOIN TablaB AS B ON A.ColumnaID =  
B.ColumnaID
```

Left Join

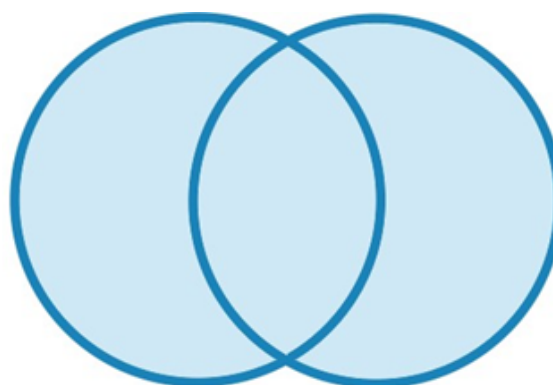
Contiene los registros de la tabla de la izquierda, incluso si no existiera un registro correspondiente en la tabla de la derecha para uno de la izquierda. Los datos de la tabla de la derecha que no pueda emparentar serán completados con NULL.



```
SELECT * FROM TablaA AS A LEFT JOIN  
TablaB AS B ON A.ColumnaID =  
B.ColumnaID
```

Full Join

Contiene la combinación de los registros de la tabla A y de la tabla B que cumplen la condición especificada. Luego, incluirá todos los registros de la izquierda y la derecha para aquellos que no cumplan la condición, completando con valores NULL los datos que no se relacionan.



```
SELECT * FROM TablaA AS A FULL JOIN  
TablaB AS B ON A.ColumnaID =  
B.ColumnaID
```

Ejemplo simple con dos tablas en SQL

Tabla A
1
2
3
4

Tabla B
3
4
5
6

El siguiente ejemplo busca clarificar el uso de los JOINS en un caso práctico. A partir de dos tablas llamadas TablaA y TablaB tenemos 8 registros (cuatro registros en cada tabla). Nótese como los valores (3, 4) son comunes para A y B. Mientras que los valores (1, 2) sólo pertenecen a A y (5, 6) sólo pertenecen a B. Esta aclaración es importante ya que el listado resultante de cada join depende de los datos contenidos en las tablas a combinar. La única columna de la tabla A, que es su clave, se llama A y la única columna de la tabla B, que también es su clave, se llama B.

Inner Join

```
SELECT * FROM TablaA AS A INNER JOIN  
TablaB AS B ON A.A = B.B
```

Resultado:

A	B
3	3
4	4

Right Join

```
SELECT * FROM TablaA AS A RIGHT  
JOIN TablaB AS B ON A.A = B.B
```

Resultado:

A	B
3	3
4	4
NULL	5
NULL	6

Left Join

```
SELECT * FROM TablaA AS A LEFT JOIN  
TablaB AS B ON A.A = B.B
```

Resultado:

A	B
1	NULL
2	NULL
3	3
4	4

Full Join

```
SELECT * FROM TablaA AS A FULL JOIN  
TablaB AS B ON A.A = B.B
```

Resultado:

A	B
1	NULL
2	NULL
3	3
4	4
NULL	5
NULL	6

Elementos de A que no se encuentran en B

```
SELECT * FROM TablaA AS A LEFT JOIN  
TablaB AS B ON A.A = B.B WHERE B.B  
IS NULL
```

Resultado:

A	B
1	NULL
2	NULL

Elementos de B que no se encuentran en A

```
SELECT * FROM TablaA AS A RIGHT  
JOIN TablaB AS B ON A.A = B.B WHERE  
A.A IS NULL
```

Resultado:

A	B
NULL	5
NULL	6

Elementos de A que no se encuentran en B y elementos de B que no se encuentran en A

```
SELECT * FROM TablaA AS A FULL JOIN  
TablaB AS B ON A.A = B.B WHERE B.B  
IS NULL OR A.A IS NULL
```

Resultado:

A	B
1	NULL
2	NULL
NULL	5
NULL	6

Elementos de A que no se encuentran en B y elementos de B que no se encuentran en A

```
SELECT * FROM TablaA CROSS JOIN  
TablaB
```

Resultado:

A	B
1	3
2	3
3	3
4	3
1	4
2	4
3	4
4	4
1	5
2	5
3	5
4	5
1	6
2	6
3	6
4	6

Anexo: Creación de tablas SQL e inserts

```
Create Database EjemplosJoins
Go
Use EjemplosJoins
Go
Create Table TablaA(
    A tinyint not null primary key
);
Go
Create Table TablaB(
    B tinyint not null primary key
);
Go
-- Datos
Insert into TablaA(A) Values (1);
Insert into TablaA(A) Values (2);
Insert into TablaA(A) Values (3);
Insert into TablaA(A) Values (4);

Insert into TablaB(B) Values (3);
Insert into TablaB(B) Values (4);
Insert into TablaB(B) Values (5);
Insert into TablaB(B) Values (6);
```