JOIN en SQL

Valeria Beratto U.

JOIN

- SQL provee varias versiones de JOINS, las que permiten la combinación de filas de dos o más tablas, basado en la columna relacionada entre ellas.
- Optimiza las consultas, porque no realiza producto cartesiano
- Usaremos este esquema:

```
CLIENTE (NCLIENTE, NOMBRE)
PRODUCTO (COD, DESCRIPCIÓN, TIPO, PRECIO)
VENTA (COD, NCLIENTE, FECHA, CANTIDAD)
```

Cliente	
Ncliente	**Nombre**
1	Juan
2	Maria

Producto			
Cod	**Des**	**Tipo**	**Precio**
11	TV	electro	100000
12	PC	tecno	70000

Venta			
Cod	**Ncliente**	**Fecha**	**Cantidad**
11	1	10/10/2020	1
11	1	01/02/2021	3

INNER JOIN

- Permite **emparejar filas** de distintas tablas de forma **más eficiente** que con el producto cartesiano **cuando** una de las **columnas de emparejamiento** está **indexada.**

```
FROM TABLA1 INNER JOIN TABLA2 ON TABLA1.COL1 COMP TABLA2.COL2
```

- **Donde:**
- **COMP:** representa cualquier operador de **comparación** (=, <, >, <=, >=, o <>) y se utiliza para establecer la condición de emparejamiento.

```
SELECT C.NCLIENTE,V.COD FROM CLIENTE C INNER JOIN VENTA V ON
C.NCLIENTE=V.NCLIENTE
```

- Qué muestra??

Respuesta	
NCliente	**Cod**
1	11
1	11

INNER JOIN

- Cuando intervienen más de dos tabla

```
FROM (TABLA1 INNER JOIN TABLA2 ON TABLA1.COL1 COMP TABLA2.COL2)
INNER JOIN TABLA3 ON TABLAi.COLi ON TABLA3.COL3
```

- Ejemplos:

```
SELECT C.NCLIENTE,P.NOMBRE FROM (CLIENTE C INNER JOIN VENTA V ON
C.NCLIENTE=V.NCLIENTE) INNER JOIN PRODUCTO P ON V.COD=P.COD
```

```
SELECT C.NCLIENTE,P.NOMBRE FROM CLIENTE C INNER JOIN (VENTA V
INNER JOIN PRODUCTO P ON V.COD=P.COD) ON C.NCLIENTE=V.NCLIENTE
```

Respuesta	
NCliente	**Nombre**
1	TV
1	TV

LEFT JOIN

- Es una extensión del inner join, **que muestra tanto los datos que están emparejados y los datos de la tabla izquierda que no se encuentren relacionados con la tabla derecha.**

```
FROM TABLA1 LEFT JOIN TABLA2 ON TABLA1.COL1 COMP TABLA2.COL2
```

- **Donde:**
- **COMP:** representa cualquier operador de **comparación** (=, <, >, <=, >=, o <>) y se utiliza para establecer la condición de emparejamiento.

```
SELECT C.NCLIENTE,V.COD FROM CLIENTE C LEFT JOIN VENTA V ON
C.NCLIENTE=V.NCLIENTE
```

- Qué muestra??

Respuesta	
NCliente	**Cod**
1	11
1	11
2	NULL

RIGHT JOIN

- Es una extensión del inner join, **que muestra tanto los datos que están emparejados y los datos de la tabla derecha que no se encuentren relacionados con la tabla izquierda.**

```
FROM TABLA1 RIGHT JOIN TABLA2 ON TABLA1.COL1 COMP TABLA2.COL2
```

- **Donde:**
- **COMP:** representa cualquier operador de **comparación** (=, <, >, <=, >=, o <>) y se utiliza para establecer la condición de emparejamiento.

```
SELECT V.NCLIENTE,P.COD FROM VENTA V RIGHT JOIN PRODUCTO P ON
V.COD=P.COD
```

- Qué muestra??

Respuesta	
NCliente	**Cod**
1	11
1	11
	12

FULL OUTER JOIN

- Es una extensión del inner join, **que muestra tanto los datos que están emparejados y los datos de ambas tabla derecha que no se encuentren relacionados con la tabla izquierda.**

```
FROM TABLA1 FULL OUTER JOIN TABLA2 ON TABLA1.COL1 COMP TABLA2.COL2
```

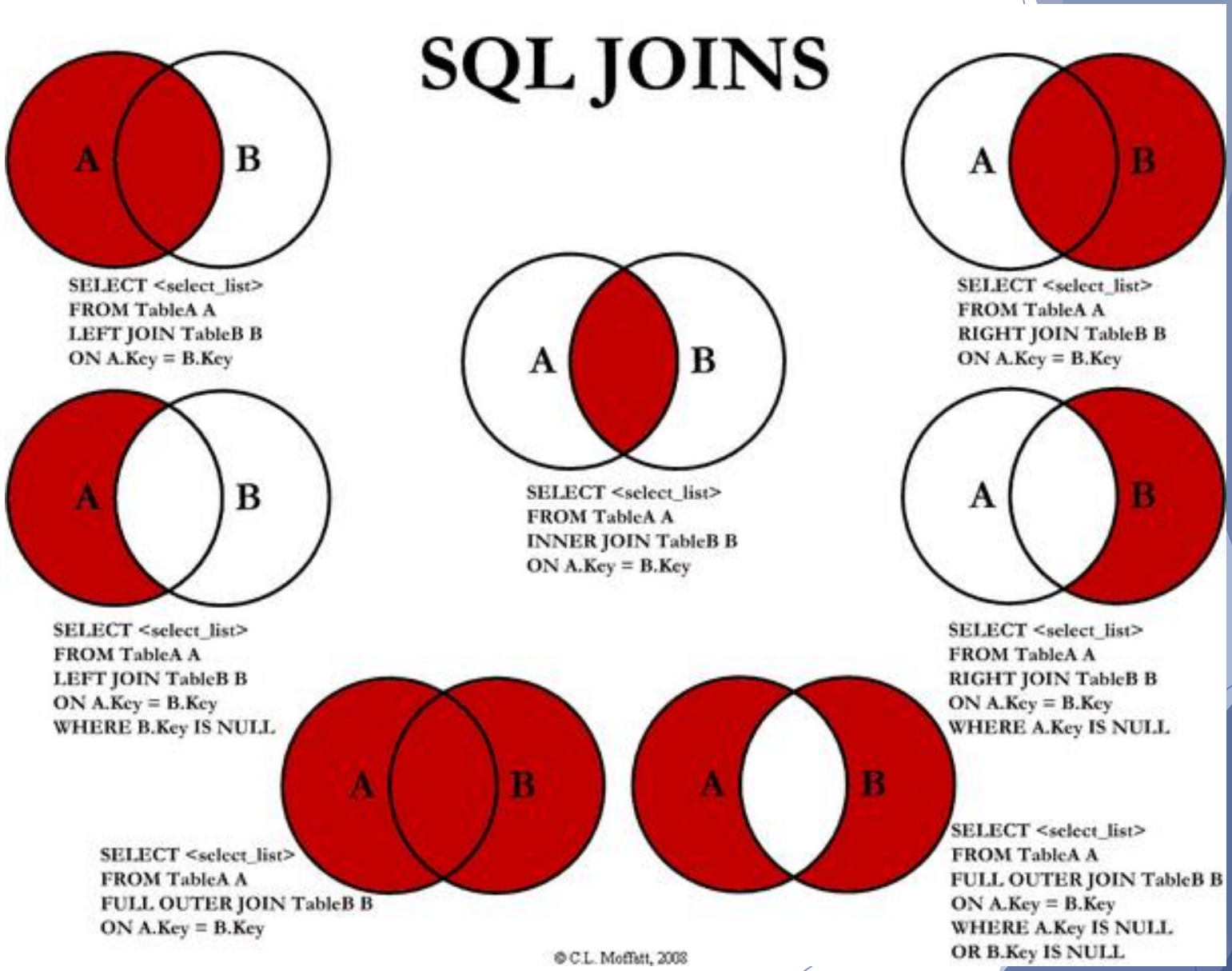
- **Donde:**
- **COMP:** representa cualquier operador de **comparación** (=, <, >, <=, >=, o <>) y se utiliza para establecer la condición de emparejamiento.

```
SELECT V.COD,V.NCLIENTE, P.DESCRIPCION FROM VENTA V FULL OUTER JOIN PRODUCTO P ON V.COD=P.COD
```

- Qué muestra??

Respuesta		
Cod	**Ncliente**	**Descripcion**
1	11	TV
1	11	TV
NULL	NULL	PC

En resumen


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008

Qué cálculan?

- SELECT V.COD, V.NCLIENTE, P.COD,P.DESCRIPCION, C.NCLIENTE,C.NOMBRE FROM (VENTA V INNER JOIN PRODUCTO P ON V.COD=P.COD) FULL OUTER JOIN CLIENTE C ON C.NCLIENTE=V.NCLIENTE;
- SELECT V.COD, V.NCLIENTE, P.COD,P.DESCRIPCION, C.NCLIENTE,C.NOMBRE FROM (VENTA V RIGHT JOIN PRODUCTO P ON V.COD=P.COD) FULL OUTER JOIN CLIENTE C ON C.NCLIENTE=V.NCLIENTE;

Ejercicios

Manteniendo el esquema donde los atributos en negrita representan las claves primarias:

*CLIENTE (**NCLIENTE**, NOMBRE)*

*PRODUCTO (**COD**, DESCRIPCIÓN, TIPO, PRECIO)*

*VENTA (**COD**, **NCLIENTE**, **FECHA**, CANTIDAD)*

Resuelvas siguientes consultas en SQL utilizando joins:

i. Listar el nombre de los clientes no tienen ventas asociadas.
ii. Listar los tipos de productos que se han vendido.
iii. Mostrar el nombre y tipo de productos que se tienen registrados, independiente si se han vendido o no.
iv. Mostrar la cantidad de clientes tienen asociadas ventas.
v. Muestre el nombre del cliente y total recibido de sus ventas asociadas (precio*cantidad)
vi. Calcule la tasa de inefectividad de los clientes (clientes sin ventas vs total de clientes).