

# Tipos de JOIN (Uniones) en SQL- Explicación.

Por: Inst. Carlos Enrique Navia Torres.

## ▼ Concepto.

La unión de tablas en SQL implica combinar dos conjuntos de datos, representados por tablas, para obtener un conjunto resultante que contiene filas de ambas tablas. Este proceso se asemeja a las operaciones de conjuntos, donde se pueden realizar uniones e intersecciones.

Es esencial que las tablas involucradas tengan un esquema compatible, es decir, el mismo número de columnas con tipos de datos compatibles. El resultado de la unión hereda los encabezados de la primera tabla, y las filas de una tabla se presentan seguidas de las filas correspondientes de la otra tabla.

Existen diversos tipos de joins, cada uno con sus propias características, como INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, entre otros, que afectan la manera en que se combinan las filas de las tablas.

## ▼ 1. Inner Join.

Esta es la forma mas fácil de seleccionar información de diferentes tablas, es tal vez la que mas usas a diario en tu trabajo con bases de datos. Esta union retorna todas las filas de la tabla A que coinciden en la tabla B. Es decir aquellas que están en la tabla A Y en la tabla B, si lo vemos en conjuntos la intersección entre la tabla A y la B.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
INNER JOIN Tabla_B B
ON A.pk = B.pk
```

### ▼ Ejemplo

```
SELECT
    p.IdProducto AS NombreCategoria,
    p.Nombre AS NombreProducto,
    p.PrecioCompra,
    p.PrecioVentaSugerido,
    c.IdCategoria AS NombreCategoria
FROM
    productos p
INNER JOIN
    categorias c ON p.IdCategoria = c.IdCategoria;
```

Esta consulta seleccionará columnas específicas de las tablas **productos** y **categorias**, y las combinará en función de la condición especificada en el **INNER JOIN**. En este caso, la condición es que el **IdCategoria** en la tabla **productos** coincida con el **IdCategoria** en la tabla **categorias**.

Esta consulta devolverá una lista de productos con información detallada, incluyendo el nombre de la categoría a la que pertenecen.

## ▼ 2. Left Join.

Esta consulta retorna todas las filas que están en la tabla A **y además** si hay coincidencias de filas en la tabla B también va a traer esas filas.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
LEFT JOIN Tabla_B B
ON A.pk = B.pk
```

#### ▼ Ejemplo:

```
SELECT
    c.IdCategoria,
    p.IdProducto,
    p.Nombre AS NombreProducto,
    p.PrecioCompra,
    p.PrecioVentaSugerido
FROM
    categorias c
LEFT JOIN
    productos p ON c.IdCategoria = p.IdCategoria;
```

En esta consulta, estamos seleccionando todas las filas de la tabla **categorias** (lado izquierdo del **LEFT JOIN**) y coincidiendo las filas correspondientes de la tabla **productos** (lado derecho del ) en función de la condición especificada. Si no hay coincidencias para una categoría en la tabla de productos, las columnas de productos mostrarán valores nulos.

Este tipo de consulta es útil cuando deseas obtener todos los registros de la tabla de la izquierda (en este caso, todas las categorías) y los registros coincidentes de la tabla de la derecha (en este caso, productos).

### ▼ 3. Right Join.

Esta consulta retorna todas las filas de la tabla B **y además** si hay filas en la tabla A que coinciden también va a traer estas filas de la tabla A.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
RIGHT JOIN Tabla_B B
ON A.pk = B.pk
```

#### Ejemplo

```
SELECT
    c.*,
    p.IdProducto,
    p.Nombre AS NombreProducto,
    p.PrecioCompra,
    p.PrecioVentaSugerido
FROM
    productos p
RIGHT JOIN
    categorias c ON p.IdCategoria = c.IdCategoria;
```

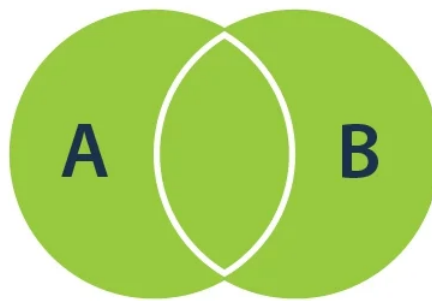
En esta consulta, estamos seleccionando todas las filas de la tabla **productos** (lado derecho del **RIGHT JOIN**) y coincidiendo las filas correspondientes de la tabla **categorias** (lado izquierdo del **RIGHT JOIN**) en función de la condición especificada.

Si no hay coincidencias para un producto en la tabla de categorías, las columnas de categorías mostrarán valores nulos.

Ten en cuenta que en muchos casos, se prefiere usar **LEFT JOIN** en lugar de **RIGHT JOIN**, ya que es más común y fácil de entender. Puedes lograr resultados similares cambiando el orden de las tablas en un **LEFT JOIN**.

## ▼ 4. Outer Join.

Este join retorna TODAS las filas de las dos tablas. Hace la union entre las filas que coinciden entre la tabla A y la tabla B.

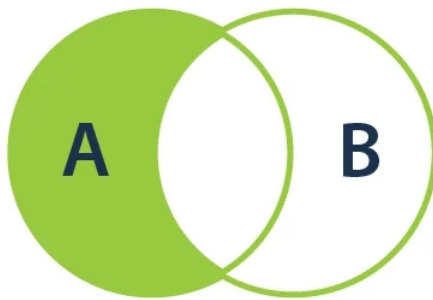


Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
FULL OUTER JOIN Tabla_B B
ON A.pk = B.pk
```

## ▼ 5. Left excluding join.

Esta consulta retorna todas las filas de la tabla de la izquierda, es decir la tabla A que no tienen ninguna coincidencia con la tabla de la derecha, es decir la tabla B.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
LEFT JOIN Tabla_B B
ON A.pk = B.pk
WHERE B.pk IS NULL
```

## ▼ 6. Right Excluding join.

Esta consulta retorna todas las filas de la tabla de la derecha, es decir la tabla B que no tienen coincidencias en la tabla de la izquierda, es decir la tabla A.



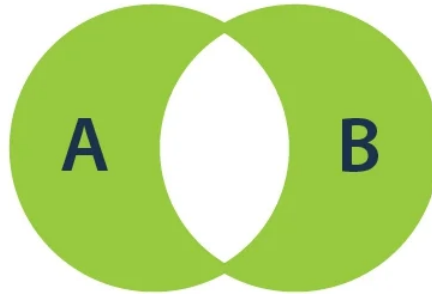
Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
RIGHT JOIN Tabla_B B
```

```
ON A.pk = B.pk  
WHERE A.pk IS NULL
```

## ▼ 7. Outer excluding join.

Esta consulta retorna todas las filas de la tabla de la izquierda, tabla A, y todas las filas de la tabla de la derecha, tabla B que no coinciden.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <select_list>  
FROM Table_A A  
FULL OUTER JOIN Table_B B  
ON A.Key = B.Key  
WHERE A.Key IS NULL OR B.Key IS NULL
```

## ▼ 8. Funciones de agregado.

Expresión	Descripción	Ejemplo
SUM	Calcula la suma del conjunto de datos determinado por la expresión, estos deben ser de tipo numérico y el resultado será del mismo tipo.	SELECT SUM (precio) FROM Ventas/* Calcula el total de las ventas */
MIN	Obtiene el menor de los valores correspondientes con el conjunto de datos determinado por la expresión.	SELECT MIN (precio) FROM Productos/* Obtiene el menor precio de los productos */

MAX	Obtiene el mayor de los valores correspondientes con el conjunto de datos determinado por la expresión.	SELECT MAX (precio) FROM Productos/* Obtiene el precio más alto de los productos*/
AVG	Calcula el valor promedio (media aritmética) del conjunto de datos determinado por la expresión, estos deben ser de tipo numérico y el resultado puede variar.	SELECT AVG (Precio) FROM Ventas /* Calcula el precio promedio de venta de los productos */
COUNT (campo)	Calcula el número de filas (registros) que contiene información en el campo especificado	SELECT COUNT (email) FROM Clientes /* Obtiene el número de clientes que registraron un email */
COUNT(*)	Cuenta el número de registros (filas) resultantes en una consulta.	SELECT COUNT (*) FROM Ventas /* Obtiene el número de ventas que se han hecho */

## ▼ 9.Ejemplos de Joins con Funciones de Agregado.

### ▼ ¿Cuántos productos ha comprado cada cliente y cuál es el total gastado por cada uno?

```
SELECT clientes.IdCliente, clientes.NombreCliente, COUNT(compras.IdCompra) AS TotalCompras, SUM(detallescompra.PrecioUnitario * detallescompra.CantidadOrdenada) AS TotalGastado
FROM clientes
INNER JOIN compras ON clientes.IdCliente = compras.IdCliente
INNER JOIN detallescompra ON compras.IdCompra = detallescompra.IdCompra
GROUP BY clientes.IdCliente, clientes.NombreCliente;
```

- **Explicación:** Esta consulta utiliza INNER JOIN para combinar la información de las tablas `clientes`, `compras`, y `detallescompra`. La función de agregado COUNT cuenta cuántas compras ha realizado cada cliente. La función SUM calcula el gasto total multiplicando el precio unitario por la cantidad ordenada en cada detalle de compra. El resultado muestra el ID del cliente, su nombre, el total de compras y el gasto total.

### ▼ ¿Cuántas compras ha realizado cada empleado y cuál es la fecha de la última compra que realizaron?



```
SELECT empleados.IdEmpleado, empleados.Nombre, COUNT(compras.IdCompra) AS TotalCompras, MAX(compras.FechaCompra) AS UltimaCompra
FROM empleados
INNER JOIN clientes ON empleados.IdEmpleado = clientes.IdEmpleado
INNER JOIN compras ON clientes.IdCliente = compras.IdCliente
GROUP BY empleados.IdEmpleado, empleados.Nombre;
```

- **Explicación:** En esta consulta, se utiliza INNER JOIN para combinar las tablas `empleados`, `clientes` y `compras`. La función de agregado COUNT cuenta cuántas compras ha realizado cada empleado. La función MAX obtiene la fecha de la última compra realizada por cada empleado. El resultado muestra el ID del empleado, su nombre, el total de compras y la fecha de la última compra.

### ▼ ¿Cuál es la cantidad total de productos vendidos en cada categoría y cuál es el precio de venta promedio por producto en cada categoría?

```
SELECT categorias.IdCategoria, categorias.Descripcion, SUM(detallescompra.CantidadOrdenada) AS TotalVendido, AVG(productos.PrecioVentaSugerido) AS PrecioPromedio
FROM categorias
INNER JOIN productos ON categorias.IdCategoria = productos.IdCategoria
INNER JOIN detallescompra ON productos.IdProducto = detallescompra.IdProducto
GROUP BY categorias.IdCategoria, categorias.Descripcion;
```

- **Explicación:** Esta consulta emplea INNER JOIN para combinar las tablas `categorias`, `productos` y `detallescompra`. La función de agregado SUM calcula la cantidad total de productos vendidos en cada categoría. La función AVG determina el precio de venta promedio por producto en cada categoría. El resultado muestra el ID de la categoría, su descripción, la cantidad total vendida y el precio promedio.