JOINS EN SQL SERVER

JOIN	Devuelve	SQL Server
INNER JOIN	Coincidencias	✔
LEFT JOIN	Todo de izquierda + coincidencias	✔
RIGHT JOIN	Todo de derecha + coincidencias	✔
FULL JOIN	Todo de ambas tablas	✔
CROSS JOIN	Producto cartesiano	✔
SELF JOIN	Una tabla consigo misma	✔
CROSS APPLY	Como INNER JOIN con función	✔
OUTER APPLY	Como LEFT JOIN con función	✔
NATURAL JOIN	Automático por columnas comunes	✘
USING	JOIN por columnas comunes	✘

1) INNER JOIN

Devuelve **solo las filas que coinciden** en ambas tablas.

Obtener pedidos con su cliente:

```
SELECT p.IdPedido, c.Nombre
FROM Pedidos p
INNER JOIN Clientes c ON p.IdCliente = c.IdCliente;
```

Qué devuelve: Solo pedidos que tienen cliente.

2) LEFT JOIN (LEFT OUTER JOIN)

Devuelve **todas las filas de la tabla izquierda**, aunque no haya coincidencia en la derecha.

Clientes con o sin pedidos:

```
SELECT c.Nombre, p.IdPedido
FROM Clientes c
LEFT JOIN Pedidos p ON c.IdCliente = p.IdCliente;
```

Qué devuelve:

- Clientes con pedidos
- Clientes sin pedidos (p.IdPedido = NULL)

3) RIGHT JOIN (RIGHT OUTER JOIN)

Devuelve **todas las filas de la tabla derecha**, aunque no haya coincidencia en la izquierda.

Pedidos con o sin cliente (poco habitual):

SELECT c.Nombre, p.IdPedido

FROM Clientes c

RIGHT JOIN Pedidos p ON c.IdCliente = p.IdCliente;

Qué devuelve:

- Pedidos con cliente
- Pedidos sin cliente (c.Nombre = NULL)

4) FULL JOIN (FULL OUTER JOIN)

Devuelve **todas las filas de ambas tablas**, coincidan o no.

Comparar dos catálogos:

SELECT a.IdProducto, b.IdProducto

FROM CatalogoA a

FULL JOIN CatalogoB b ON a.IdProducto = b.IdProducto;

Qué devuelve:

- Productos que están en A
- Productos que están en B
- Productos que están en ambos

5) CROSS JOIN

Producto cartesiano: combina **todas las filas de A con todas las de B**.

Generar combinaciones de tallas y colores:

```
SELECT t.Talla, c.Color
FROM Tallas t CROSS JOIN Colores c;
```

6) SELF JOIN

Una tabla se une consigo misma.

Obtener empleados con su jefe:

```
SELECT e.Nombre AS Empleado, j.Nombre AS Jefe
FROM Empleados e
LEFT JOIN Empleados j ON e.IdJefe = j.IdEmpleado;
```

7) CROSS APPLY

Se usa cuando la tabla derecha es una **función que depende de la fila izquierda**. Es como un INNER JOIN dinámico.

Obtener los pedidos de cada cliente usando una TVF:

```
SELECT c.Nombre, p.*
FROM Clientes c
CROSS APPLY dbo.PedidosPorCliente(c.IdCliente) p;
```

Solo devuelve clientes que tienen pedidos.

8) OUTER APPLY

Igual que CROSS APPLY, pero devuelve también filas sin coincidencia (como LEFT JOIN).

Clientes con o sin pedidos:

```
SELECT c.Nombre, p.*
FROM Clientes c
OUTER APPLY dbo.PedidosPorCliente(c.IdCliente) p;
```

9) NATURAL JOIN

NO existe en SQL Server (Es de Oracle y PostgreSQL).

10) USING

NO existe en SQL Server (Es de MySQL, PostgreSQL, Oracle).