

```

USE NORTHWIND
/* SQL JOINS */
/* SQL INNER JOIN
-----
Muestra valores que coinciden en las Tablas involucradas
JOIN con 2 Tablas
*/
-----
SELECT  nombre_columna(s)
FROM    Tabla1
INNER JOIN Tabla2
ON Tabla1.nombreColumna = Tabla2.nombreColumna
-----
-- Ejemplo: Mostrar Numero de Orden y Nombre del Cliente
--
SELECT o.OrderID, c.CompanyName
FROM Orders o
INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
-----

-- JOIN con 3 Tablas    (n-1 condiciones)
--  Mostrar Numero de Orden y Nombre del Cliente y Nombre del Transportista
--
SELECT o.OrderID, c.CompanyName,s.CompanyName
FROM ((Orders o
INNER JOIN Customers c
ON o.CustomerID = c.CustomerID )
INNER JOIN Shippers S
ON O.ShipVia = s.ShipperID )

-- los Parentesis no son necesarios y INNER es opcional

SELECT o.OrderID, c.CompanyName as Cliente,s.CompanyName as Transportista
FROM Orders o
INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
INNER JOIN Shippers S
ON O.ShipVia = s.ShipperID

--Otra Forma de Hacer los JOIN Multitablas

SELECT o.OrderID, c.CompanyName as Cliente,s.CompanyName as Transportista
FROM Orders o, Customers c, Shippers S
WHERE o.CustomerID = c.CustomerID and O.ShipVia = s.ShipperID

/* SQL LEFT JOIN
-----
Retorna TODOS las filas de la Tabla Izquierda del JOIN (Customers)
aunque NO EXISTAN coincidencias en la Tabla Derecha del JOIN (Orders)
*/
-- Ejemplo: Mostrar el Nombre de TODOS los Cliente, el Nro de Orden y la Fecha de
la Orden.
-- el Nombre del Cliente se debe mostrar aunque el Cliente no haya colocado
Ordenes.

SELECT c.CompanyName, o.OrderID, o.OrderDate
FROM Customers c LEFT JOIN Orders o
ON c.CustomerID = o.CustomerID
order by o.OrderDate

```

```

/* SQL LEFT JOIN    ''EXCLUDING''
*****

Retorna TODOS las filas de la Tabla Izquierda del JOIN (Customers)
que NO TENGAN coincidencias en la Tabla Derecha del JOIN (Orders)
*/
-- Ejemplo: Muestra solo los Clientes que No colocaron ORDENES

SELECT c.CompanyName, o.OrderID, o.OrderDate
FROM   Customers c LEFT JOIN Orders o
      ON c.CustomerID = o.CustomerID
WHERE  o.CustomerID IS NULL
order by o.OrderDate

/* SQL RIGHT JOIN
*****

Retorna TODOS las filas de la Tabla Derecha del JOIN (Employees)
aunque NO TENGAN coincidencias en la Tabla Izquierda del JOIN (Orders)
*/
-- Agregamos un Nuevo Empleado y no le creamos Ordenes
INSERT INTO Employees (FirstName, LastName) values ('Jorge', 'Benitez')

-- Mostramos A todos los Empleados que hayan colocado Ordenes y a los que no han
generado Ordenes

SELECT o.OrderID, o.OrderDate, e.FirstName, e.LastName
FROM   Orders o      RIGHT JOIN Employees e
      ON e.EmployeeID = o.EmployeeID
order by e.LastName, e.FirstName

/* SQL RIGHT JOIN    EXCLUDING

Retorna SOLAMENTE las filas de la Tabla Derecha del JOIN (Employees)
que NO TENGAN coincidencias en la Tabla Izquierda del JOIN (Orders)
*/
-- Ejemplo: Mostrar los empleados que no han generado Ordenes

SELECT o.OrderID, o.OrderDate, e.FirstName, e.LastName
FROM   Orders o      RIGHT JOIN Employees e
      ON e.EmployeeID = o.EmployeeID
WHERE  o.EmployeeID IS NULL
order by e.LastName, e.FirstName

/*  SQL FULL JOIN    */
-- *****

-- Ordenes y Empleados Pongo en null elCodigo de Empleado en las Ordenes del
Empleado 5
-- para que haya Ordenes sin Empleados
UPDATE Orders set EmployeeID = null where EmployeeID = 5  --43 registros

--select employeeID, count(*) from Orders Group by employeeID
--select * from Employees

--Ejemplo: Me muestra TODOS los Empleados y TODAS las Ordenes
-- incluyendo Empleados SIN ORDENES y Ordenes SIN EMPLEADOS

SELECT e.EmployeeID, e.FirstName, e.LastName, o.OrderID, o.OrderDate, o.ShipCountry
FROM   Employees e  FULL JOIN Orders o
      ON e.EmployeeID = o.EmployeeID
ORDER BY e.EmployeeID DESC

```

```

/* SQL FULL JOIN (EXCLUDING) */
-- *****

-- Ejemplo: el Anterior pero SOLO MUESTRA: Empleados SIN ORDENES y Ordenes SIN
EMPLEADOS

SELECT e.EmployeeID, e.FirstName,e.LastName, o.OrderID,o.OrderDate,o.ShipCountry
FROM Employees e FULL JOIN Orders o
      ON e.EmployeeID = o.EmployeeID

where e.EmployeeID is null OR o.EmployeeID IS null

ORDER BY e.EmployeeID DESC

-----

/* SQL SELF JOIN */
-- *****

--Saber a cual Empleado reportan los Empleados
--select e.EmployeeID, e.FirstName+' '+e.LastName AS 'Empleado',e.ReportsTo FROM
Employees e WHERE e.ReportsTo IS NULL

SELECT e.EmployeeID, e.FirstName+' '+e.LastName AS 'Empleado',
      e.ReportsTo, e1.FirstName+' '+e1.LastName AS 'Supervisor'
FROM Employees e JOIN Employees e1
      ON e1.EmployeeID = e.ReportsTo

<<<< EJERCICIO DE INVESTIGACION >>>>
/* SQL SELF JOIN 'EXCLUDING' */

-- Que Empleados NO REPORTAN a NINGUN Empleado
--ASI NO FUNCIONA
SELECT e.EmployeeID, e.FirstName+' '+e.LastName AS 'Empleado',
      e.ReportsTo, e1.FirstName+' '+e1.LastName AS 'Supervisor'
FROM Employees e JOIN Employees e1
      ON e1.EmployeeID = e.ReportsTo
WHERE e.ReportsTo IS NULL

--Para que FUNCIONE necesitamos un LEFT JOIN
SELECT e.EmployeeID, e.FirstName+' '+e.LastName AS 'Empleado',
      e.ReportsTo, e1.FirstName+' '+e1.LastName AS 'Supervisor'
FROM Employees e LEFT JOIN Employees e1
      ON e1.EmployeeID = e.ReportsTo
WHERE e.ReportsTo IS NULL

```

```
/* SQL LEFT JOIN con GROUP BY */
```

```
/* Mostrar para Transportista (Shippers) cuantas Ordenes se enviaron para cada  
Uno de ellos. */
```

```
SELECT s.CompanyName, COUNT(o.OrderID) as CantidadOrdenes  
FROM Shippers s INNER JOIN Orders o  
ON s.ShipperID = o.ShipVia  
GROUP BY s.CompanyName
```

```
/*
```

```
Si no tiene Ordenes, tambien se debe mostrar el Nombre del Transportista
```

```
*/
```

```
SELECT s.CompanyName, COUNT(o.OrderID) as CantidadOrdenes  
FROM Shippers s LEFT JOIN Orders o  
ON s.ShipperID = o.ShipVia  
GROUP BY s.CompanyName
```