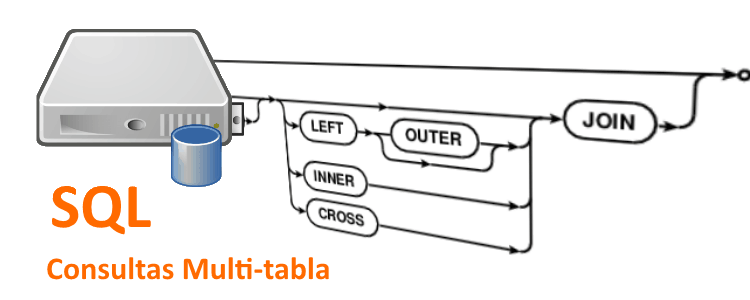
**Fundamentos de SQL: Consultas SELECT multi-tabla - Tipos de JOIN**

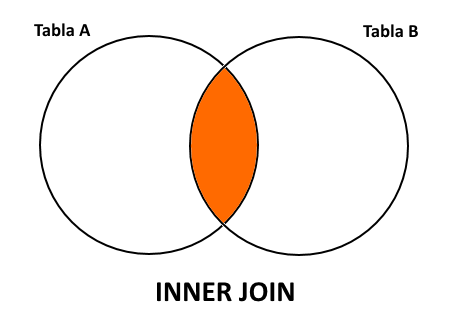


.

**Combinaciones internas - INNER JOIN**

Las combinaciones internas se realizan mediante la instrucción **INNER JOIN**. Devuelven únicamente aquellos registros/filas **que tienen valores idénticos en los dos campos**que se comparan para unir ambas tablas. Es decir, aquellas que tienen elementos en las dos tablas, identificados éstos por el campo de relación.

La mejor forma de verlo es con un [diagrama de Venn](http://es.wikipedia.org/wiki/Diagrama_de_Venn) que ilustre en qué parte de la relación deben existir registros:



En este caso se devuelven los registros que tienen nexo de unión en ambas tablas. Por ejemplo, en la relación entre las tablas de clientes y pedidos en [Northwind](https://www.campusmvp.es/recursos/post/VIDEO-Como-instalar-paso-a-paso-la-base-de-datos-Northwind.aspx" \t "_blank), se devolverán los registros de todos los clientes que tengan al menos un pedido, relacionándolos por el ID de cliente.

Esto **puede ocasionar la desaparición del resultado de filas** de alguna de las dos tablas, por tener valores nulos, o por tener un valor que no exista en la otra tabla entre los campos/columnas que se están comparando.

Su sintaxis es:

FROM Tabla1 [INNER] JOIN Tabla2 ON Condiciones\_Vinculos\_Tablas

Así, para seleccionar los registros comunes entre la *Tabla1* y la *Tabla2* que tengan correspondencia entre ambas tablas por el campo *Col1*, escribiríamos:

SELECT T1.Col1, T1.Col2, T1.Col3, T2.Col7  
FROM Tabla1 T1 INNER JOIN Tabla2 T2 ON T1.Col1 = T2.Col1

Por ejemplo, para obtener en Northwind los clientes que tengan algún pedido, bastaría con escribir:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C INNER JOIN Orders O ON C.CustomerID = O.CustomerID

Que nos devolverá 830 registros. Hay dos pedidos en la tabla de Orders sin cliente asociado, como puedes comprobar, y éstos no se devuelven por no existir la relación entre ambas tablas.

En realidad esto ya lo conocíamos puesto que en las combinaciones internas, el uso de la palabra **INNER** es opcional (por eso lo hemos puesto entre corchetes). Si simplemente indicamos la palabra JOIN y la combinación de columnas (como ya hemos visto en el artículo anterior) el sistema sobreentiende que estamos haciendo una combinación interna. Lo hemos incluido por ampliar la explicación y por completitud.

**Combinaciones externas (OUTER JOIN)**

Las **combinaciones externas** se realizan mediante la instrucción **OUTER JOIN**. Como enseguida veremos, devuelven todos los valores de la tabla que hemos puesto a la derecha, los de la tabla que hemos puesto a la izquierda o los de ambas tablas según el caso, devolviendo además valores nulos en las columnas de las tablas que no tengan el valor existente en la otra tabla.

Es decir, que **nos permite seleccionar algunas filas de una tabla aunque éstas no tengan correspondencia con las filas de la otra tabla** con la que se combina. Ahora lo veremos mejor en cada caso concreto, ilustrándolo con un diagrama para una mejor comprensión.

La sintaxis general de las combinaciones externas es:

FROM Tabla1 [LEFT/RIGHT/FULL] [OUTER] JOIN Tabla2 ON Condiciones\_Vinculos\_Tablas

Como vemos existen tres variantes de las combinaciones externas.

En todas estas combinaciones externas **el uso de la palabra OUTER es opcional**. Si utilizamos LEFT, RIGHT o FULL y la combinación de columnas, el sistema sobreentiende que estamos haciendo una combinación externa.

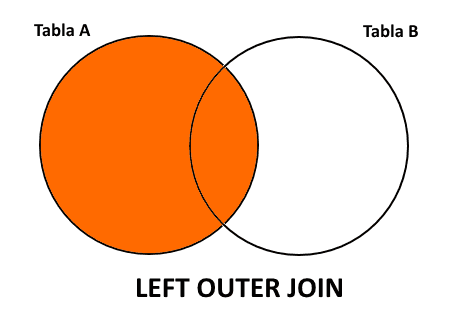
**Variante LEFT JOIN**

Se obtienen todas las filas de la tabla colocada a la izquierda, aunque no tengan correspondencia en la tabla de la derecha.

Así, para seleccionar todas las filas de la Tabla1, aunque no tengan correspondencia con las filas de la Tabla2, suponiendo que se combinan por la columna Col1 de ambas tablas escribiríamos:

SELECT T1.Col1, T1.Col2, T1.Col3, T2.Col7  
FROM Tabla1 T1 LEFT [OUTER] JOIN Tabla2 T2 ON T1.Col1 = T2.Col1

Esto se ilustra gráficamente de la siguiente manera:

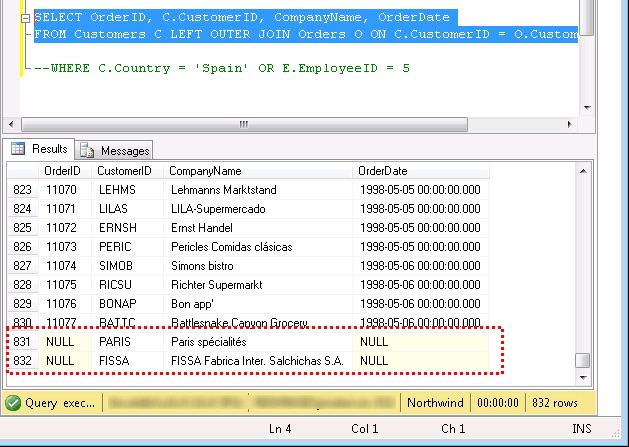


De este modo, volviendo a Northwind, si escribimos la siguiente consulta:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C **LEFT JOIN** Orders O ON C.CustomerID = O.CustomerID

Obtendremos 832 registros ya que se incluyen **todos los clientes y sus pedidos, incluso aunque no tengan pedido alguno**. Los que no tienen pedidos carecen de la relación apropiada entre las dos tablas a partir del campo *CustomerID*. Sin embargo se añaden al resultado final dejando la parte correspondiente a los datos de la tabla de pedidos con valores nulos, como se puede ver en esta captura de SQL Server:



**Variante RIGHT JOIN**

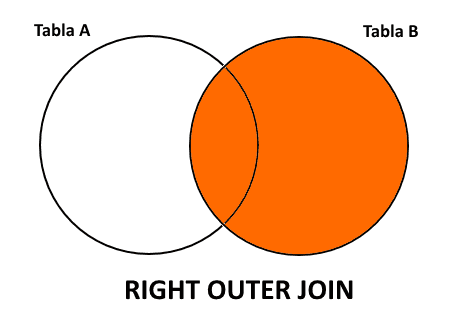
Análogamente, usando RIGHT JOIN se obtienen todas las filas de la tabla de la derecha, aunque no tengan correspondencia en la tabla de la izquierda.

Así, para seleccionar todas las filas de la Tabla2, aunque no tengan correspondencia con las filas de la Tabla1 podemos utilizar la cláusula RIGHT:

SELECT T1.Col1, T1.Col2, T1.Col3, T2.Col7

FROM Tabla1 T1 RIGHT [OUTER] JOIN Tabla2 T2 ON T1.Col1 = T2.Col1

El diagrama en este caso es complementario al anterior:



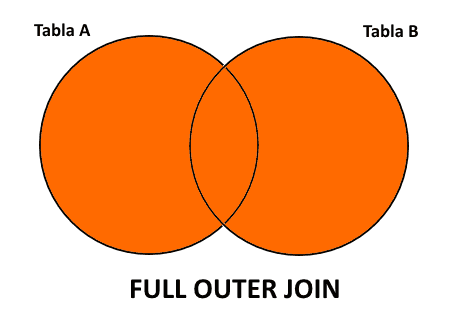
Si en nuestra base de datos de ejemplo queremos obtener **todos los pedidos aunque no tengan cliente asociado**, junto a los datos de dichos clientes, escribiríamos:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate  
FROM Customers C **RIGHT JOIN** Orders O ON C.CustomerID = O.CustomerID

En este caso se devuelven 830 registros que son todos los pedidos. Si hubiese algún pedido con el CustomerID vacío (nulo) se devolvería también en esta consulta (es decir, órdenes sin clientes), aunque en la base de datos de ejemplo no se da el caso.

**Variante FULL JOIN**

Se obtienen todas las filas en ambas tablas, aunque no tengan correspondencia en la otra tabla. Es decir, todos los registros de A y de B aunque no haya correspondencia entre ellos, rellenando con nulos los campos que falten:



Es equivalente a obtener los registros comunes (con un INNER) y luego añadirle los de la tabla A que no tienen correspondencia en la tabla B, con los campos de la tabla vacíos, y los registros de la tabla B que no tienen correspondencia en la tabla A, con los campos de la tabla A vacíos.

Su sintaxis es:

SELECT T1.Col1, T1.Col2, T1.Col3, T2.Col7

FROM Tabla1 T1 FULL [OUTER] JOIN Tabla2 T2 ON T1.Col1 = T2.Col1

Por ejemplo, en Northwind esta consulta:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C FULL JOIN Orders O ON C.CustomerID = O.CustomerID

nos devuelve nuevamente 832 registros: los clientes y sus pedidos, los clientes sin pedido (hay 2) y los pedidos sin cliente (que en este caso son 0).