Fundamentos de Programación con SQL Server



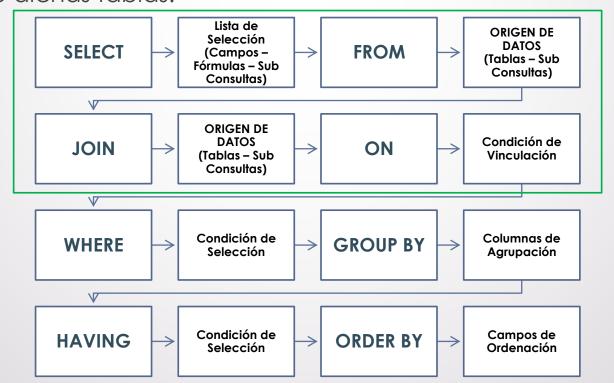


octavio.robleto@gmail.com

https://orobleto.github.io/octaviorobleto.github.io/

JOIN

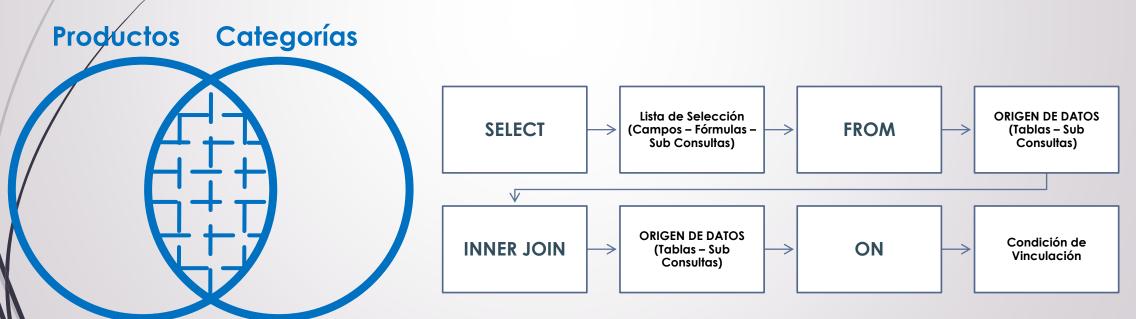
Esta instrucción se usa para vincular los datos de tablas relacionadas a través de campos en común que usualmente suelen ser las FOREIGN KEY y así mostrar como resultado filas que combinan la información procedente de dichas tablas.





INNER JOIN

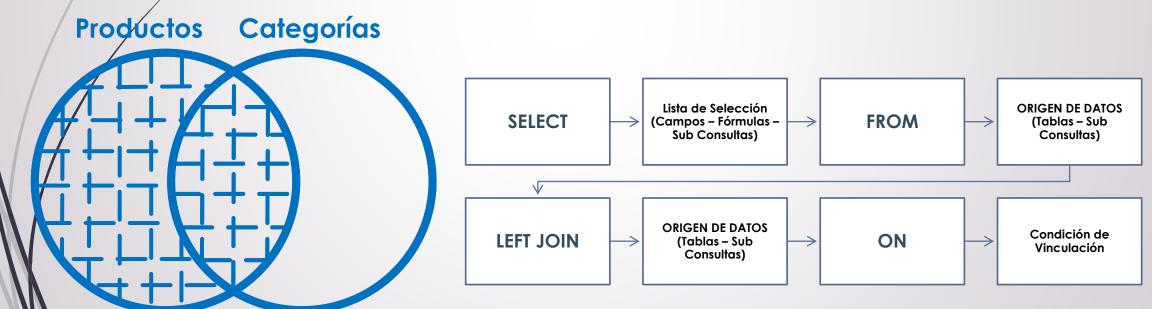
Devuelve sólo aquellas filas donde los valores de los campos de la Tabla Izquierda (Productos) que se utilizan para hacer el JOIN concuerden con los valores de los campos de la Tabla Derecha (Categorías).





LEFT JOIN

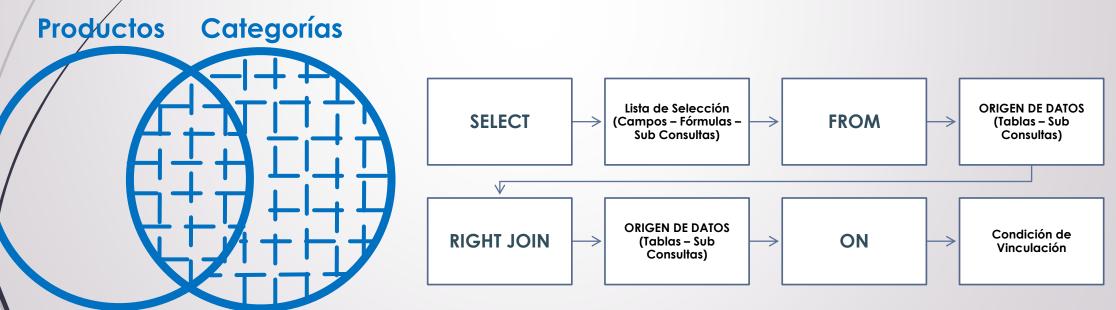
Devuelve todas las filas de la Tabla Izquierda (Productos) y además las filas de la Tabla Derecha (Categorías) que concuerden con los valores utilizados al hacer el LEFT JOIN. El resultado es NULL desde el lado derecho, si no hay coincidencia.





RIGHT JOIN

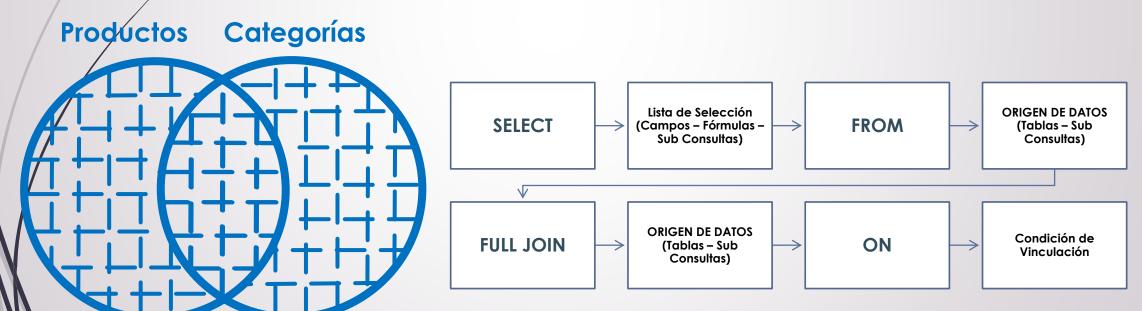
Devuelve todas las filas de la Tabla Derecha (Categoría) y además las filas de la Tabla Izquierda (Productos) que concuerden con los valores utilizados al hacer el RIGHT JOIN. El resultado es NULL desde el lado izquierdo, si no hay coincidencia.





FULL JOIN

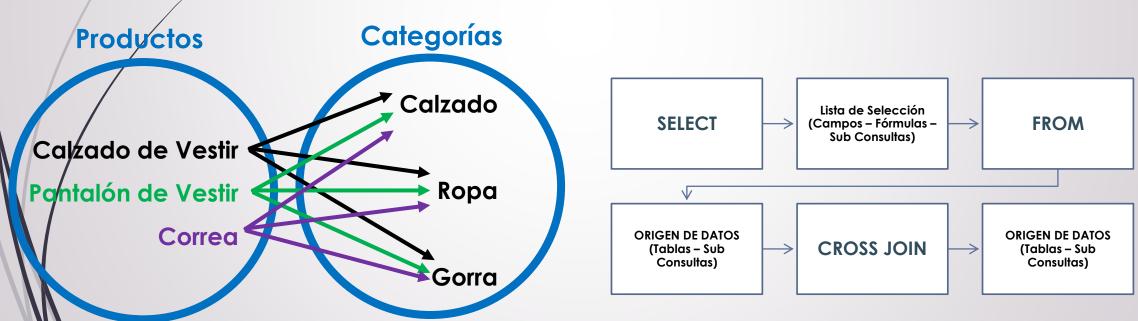
Devuelve todas las filas de la Tabla Izquierda (Productos), todas las filas de la Tabla Derecha (Categorías) concuerden o no con los valores utilizados al hacer el FULL JOIN. El resultado es NULL desde ambos lados, si no hay coincidencia.





CROSS JOIN

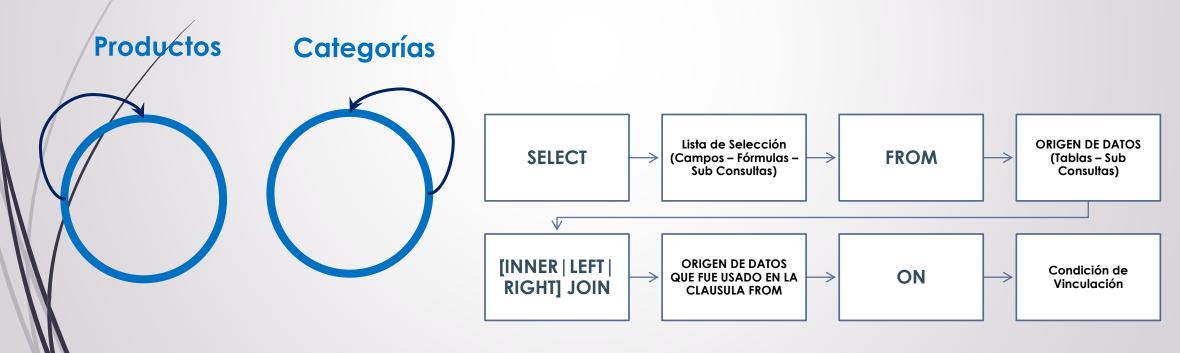
 Devuelve un producto cartesiano de las tablas involucradas, es decir, todas las combinaciones posibles.





SELFT JOIN

Es un JOIN común pero que se hace con la misma tabla





Tablas Temporales

 Son tablas que las crean temporalmente para resolver un tipo de problema. Poseen una vida útil definida por el usuario (desarrollador, programador, etc.)





Tipos

Locales: Las tablas temporales locales incluyen en el nombre el símbolo # como primer carácter. Se crean por cada usuario conectado y la tabla se elimina automáticamente cuando el usuario termina la sesión.

```
CREATE TABLE #TMP_Local
(
ID INT IDENTITY(1,1),
Descripcion VARCHAR(100)
)
```

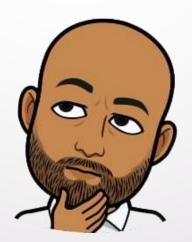
Globales: Las tablas temporales globales inician con dos símbolos ## en el nombre y son visibles por todos los usuarios conectados al servidor. Al desconectarse todos los usuarios, la tabla temporal global se elimina automáticamente.

```
CREATE TABLE ##TMP_Global
(
ID INT IDENTITY(1,1),
Descripcion VARCHAR(100)
)
```



Consideraciones

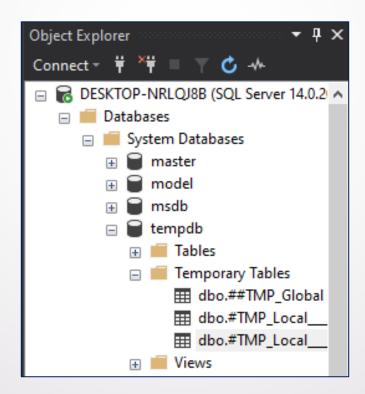
- No se pueden usar Foreign key.
- Una tabla temporal creada en un SP sólo está presente cuando se completa las transacciones del procedimiento almacenado.
- Las tablas temporales locales se pueden crear con el mismo nombre para diferentes usuarios, SQL Server le incluye un sufijo para diferenciarlas.
- Los dos tipos de tablas se pueden quitar usando Drop Table.





Almacenamiento

 Las tablas temporales se almacenan en la base de datos del sistema Tempolo.





CTE (Common Table Expression)

 También conocido como expresión de tabla común especifica un conjunto de resultados con nombre temporal al que puede hacer referencia dentro de una instrucción SELECT, INSERT, UPDATE o DELETE

```
;WITH CTE_Identificador (Campos) AS
(
    SELECT
    Campos
    FROM Tabla
)
SELECT Campos FROM CTE_Identificador
```



Consideraciones

- Un CTE debe ir seguido de una sola instrucción SELECT, INSERT, UPDATE o DELETE que haga referencia a algunas o todas las columnas CTE.
- Se pueden definir múltiples definiciones de consulta CTE en un CTE que sea no recursivo.
- Un CTE puede hacer referencia a sí mismo y a los CTE definidos previamente en la misma cláusula WITH
- Las instrucciones ORDER BY, INTO no se pueden usar en la definición de consulta CTE no recursiva
- Las subconsultas SELECT DISTINCT, GROUP BY, HAVING, Scalar aggregation, TOP, LEFT, RIGHT, OUTER JOIN (INNER JOIN está permitida) no se pueden utilizar en una definición de consulta CTE recursiva.





CTE Recursivo

Son ciclos o bucles de procedimiento repetidos, también conocidos como recursividad. La consulta recursiva se llama a sí misma hasta que la consulta satisfaga la condición. En un CTE recursivo, nosotros debemos proporcionar una condición where para terminar la recursión.

```
;WITH CTE_Identificador (Campos) AS
(
    SELECT
    1
    UNION ALL
    SELECT
    Campos + 1
    FROM CTE_Identificador
    WHERE Campos < 100
)
SELECT Campos FROM CTE_Identificador</pre>
```



Inserción Masiva

- Se pueden copiar los datos de una consulta masivamente a una nueva tabla con la instrucción SELECT INTO.
- La nueva tabla se creará con los nombres de columna y los tipos definidos en el origen de datos. Puede crear nuevos nombres de columna utilizando la cláusula AS.

