SQL Programming

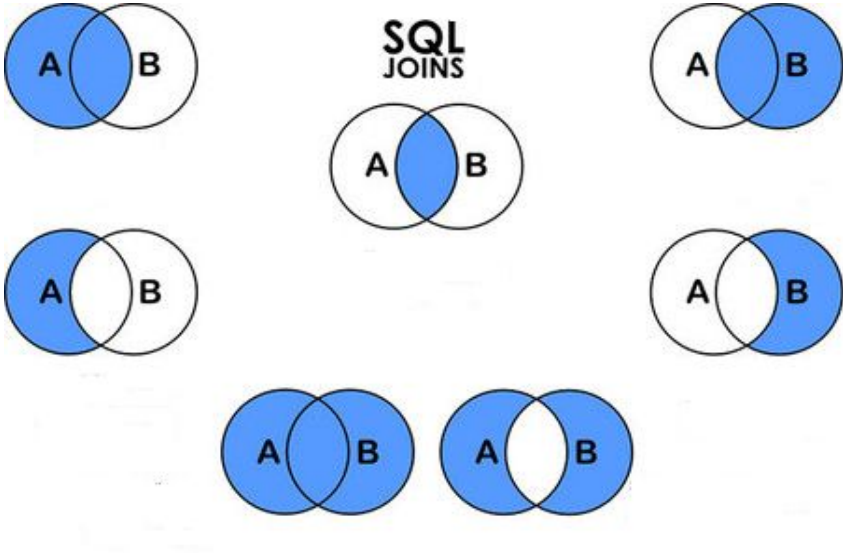
Módulo 3 - Relacionando Conjuntos de datos

Relacionando Conjuntos de datos

FROM

Especifica las tablas, vistas, tablas derivadas y tablas combinadas que se utilizan en las instrucciones DELETE, SELECT y UPDATE en SQL Server 2019. En la instrucción SELECT, la cláusula FROM es necesaria excepto cuando la lista de selección solo contiene constantes, variables y expresiones aritméticas (sin nombres de columna).

CONJUNTO DE DATOS A	CONJUNTO DE DATOS B
1	1
2	3
3	4



Tipo de combinación

Especifica el tipo de operación de combinación.

INNER

Especifica que se devuelvan todos los pares de filas coincidentes. Rechaza las filas no coincidentes de las dos tablas. Si no se especifica ningún tipo de combinación, éste es el valor predeterminado.

CONJUNTO DE DATOS A	CONJUNTO DE DATOS B
1	1
3	3

El siguiente ejemplo muestra los teléfonos de los empleados con código menor a 10.

Sintaxis

```
SELECT a.BusinessEntityID, a.FirstName, b.PhoneNumber
FROM Person.Person a INNER JOIN Person.PersonPhone b
ON a.BusinessEntityID = b.BusinessEntityID
WHERE a.BusinessEntityID < 10

---------------------------------------
BusinessEntityID    FirstName    PhoneNumber
1                   Ken           697-555-0142
2                   Terri   819-555-0175
3                   Roberto        212-555-0187
4                   Rob           612-555-0100
5                   Gail          849-555-0139
6                   Jossef  122-555-0189
7                   Dylan   181-555-0156
8                   Diane   815-555-0138
9                   Gigi          185-555-0186
```

Tipo de combinación

LEFT [OUTER]

Especifica que todas las filas de la tabla izquierda que no cumplan la condición de combinación se incluyan en el conjunto de resultados, con las columnas de resultados de la otra tabla establecidas en NULL, además de todas las filas devueltas por la combinación interna.

CONJUNTO DE DATOS A	CONJUNTO DE DATOS B
1	1
2	NULL
3	3

El siguiente ejemplo muestra la cantidad de órdenes de ventas para los productos 389 y 897.

Sintaxis

```
SELECT a.ProductID, b.OrderQty
FROM Production.Product a LEFT JOIN
Sales.SalesOrderDetail b
ON a.ProductID = b.ProductID
WHERE a.ProductID in (389,897)
----------------------------
ProductID   OrderQty
897           1
897           3
389           NULL
```

Tipo de combinación

RIGHT [OUTER]

Especifica que todas las filas de la tabla derecha que no cumplan la condición de combinación se incluyan en el conjunto de resultados, con las columnas de resultados de la otra tabla establecidas en NULL, además de todas las filas devueltas por la combinación interna.

CONJUNTO DE DATOS A	CONJUNTO DE DATOS B
1	1
3	3
NULL	4

El siguiente ejemplo muestra la cantidad de órdenes de compra para los clientes 701 y 11000.

Sintaxis

```
SELECT s.SalesOrderID, c.CustomerID
FROM Sales.SalesOrderHeader AS s RIGHT JOIN
Sales.Customer AS c
ON s.CustomerID = c.CustomerID
WHERE C.CustomerID IN (701, 11000)
------------------------------
SalesOrderID      CustomerID
NULL               701
43793              11000
51522              11000
57418              11000
```

Tipo de combinación

FULL [OUTER]

Especifica que una fila de la tabla de la derecha o de la izquierda, que no cumpla la condición de combinación, se incluya en el conjunto de resultados y que las columnas que correspondan a la otra tabla se establezcan en NULL. De esta forma se agregan más filas a las que se suelen devolver con INNER JOIN.

CONJUNTO DE DATOS A	CONJUNTO DE DATOS B
1	1
1	NULL
2	1
NULL	2
2	1
2	2

Tipo de combinación

FULL [OUTER]

El siguiente ejemplo muestra la cantidad de órdenes de compra para los clientes 701 y 11000

Sintaxis

```
SELECT c.CurrencyRateID, s.SalesOrderID
FROM Sales.SalesOrderHeader AS s
FULL OUTER JOIN Sales.CurrencyRate AS c
ON c.CurrencyRateID = s.CurrencyRateID
WHERE (c.CurrencyRateID = 2705 OR s.SalesOrderID IN (56358, 56359));

------------------------------
CurrencyRateID    SalesOrderID
NULL                                                    56358
9615                                                    56359
2705               NULL
```

Tipo de combinación

CROSS

Se utiliza en los casos que se quiere hacer el producto cartesiano entre dos tablas.

CONJUNTO DE DATOS A	CONJUNTO DE DATOS B
A	1
B	2
PRODUCTO CARTESIANO	
A	1
A	2
B	1
B	2

Sintaxis

```
SELECT sd.SalesOrderID, sh.SalesOrderID
FROM Sales.SalesOrderDetail sd
CROSS JOIN Sales.SalesOrderHeader sh
WHERE sd.SalesOrderID = 43665 AND sh.SalesOrderID IN
(43662, 43668)
-----------------------------------
SalesOrderID      SalesOrderID
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43662
43665                  43668
...
```

Tipo de combinación

SELF

El **self join** correlaciona diferentes registros que se encuentran en la misma tabla.

Es comúnmente utilizado en consultas que comparten la misma información.

Cuando se utiliza más de una vez la misma tabla para construir un JOIN es necesario utilizar ALIAS para poder identificarlas.

La siguiente consulta realiza un reporte de los empleados de la empresa y el nombre de su supervisor asignado.

Como el supervisor también es un empleado de la empresa se debe acceder a la misma tabla “Empleados” pero con un alias distinto. Para poder resolver esta consulta es necesario hacer un SELF JOIN. Esto es puramente conceptual, ya que lo que se está aplicando es un INNER JOIN entre la misma tabla “Empleados”.

Tipo de combinación

SELF

Empleados			
Código	**Nombre**	**Puesto**	**Supervisor**
1	Juan	1	NULL
2	Pedro	1	1
3	Maria	2	NULL
4	Martin	2	3
5	Matias	3	3

Sintaxis

```
CREATE TABLE Empleados(
Codigo INT,
Nombre VARCHAR(50),
Puesto VARCHAR(50),
Supervisor INT
);

INSERT INTO dbo.Empleados
VALUES
(1, 'Juan', 1, NULL),
(2, 'Pedro', 1, 1),
(3, 'Maria', 2, NULL),
(4, 'Martin', 2, 3),
(5, 'Matias', 3, 3);
```

Tipo de combinación

SELF

El siguiente ejemplo muestra a los empleados junto a sus supervisores.

Sintaxis

```
SELECT p2.Puesto, p2.Nombre, p2.Codigo, p1.Nombre AS Supervisor
FROM Empleados p1 INNER JOIN Empleados p2
ON p1.Codigo = p2.Supervisor;
-----------------------------------------------
Puesto      Nombre      Codigo      Supervisor
1            Pedro        2           Juan
2            Martin       4           Maria
3            Matias       5           Maria
```

¡Muchas gracias!

¡Sigamos trabajando!