**Tecnológico Nacional de México**

**Instituto Tecnológico de Tijuana**

Subdirección Académica

Departamento de Sistemas y Computación

*Semestre Agosto-Diciembre 2015*

Materia: **Taller de Base de Datos**

Unidad IV

**Trabajo # 19**

Navarro Mancillas Guillermo 13211447

Maestro:  
Doc. Margarita Ramírez Ramírez

*Tijuana B.C. a 2 de noviembre de 2015*

USE AdventureWorks2008R2;

GO

/\* el GROUP BY simple devuelve un conjunto de resultados para comparar con los conjuntos de resultados. Estos ejemplos utilizan los operadores GROUP BY con la misma instrucción SELECT.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 290, 288)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY ROLLUP(

T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID)

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;

/\*En el ejemplo siguiente, el operador ROLLUP devuelve un conjunto de resultados que contiene las agrupaciones siguientes:

Region, Country, Store y SalesPersonID, Region, Country y Store, Region y Country, Region, total general\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 290, 288)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY ROLLUP(

H.SalesPersonID, S.Name, T.CountryRegionCode, T.[Group])

ORDER BY H.SalesPersonID, S.Name, T.CountryRegionCode, T.[Group];

/\*Las columnas de la lista ROLLUP son las mismas que las del ejemplo B, pero están en el orden inverso. Las columnas se resumen de derecha a izquierda; por consiguiente, el orden afecta a las agrupaciones. El número de filas en el conjunto de resultados podría variar en función del orden de las columnas.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,DATEPART(yyyy,OrderDate) AS 'Year'

,DATEPART(mm,OrderDate) AS 'Month'

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND DATEPART(yyyy,OrderDate) = '2006'

GROUP BY

ROLLUP(T.[Group], T.CountryRegionCode)

,ROLLUP(DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate))

ORDER BY T.[Group], T.CountryRegionCode

,DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate);

/\*En este ejemplo se obtiene el producto cruzado de las dos operaciones ROLLUP.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer AS C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 288, 290)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY CUBE(

T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID)

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;

/\*En el ejemplo siguiente, el operador CUBE devuelve un conjunto de resultados que tiene una agrupación para todas las combinaciones de columnas posibles de la lista CUBE y una agrupación de total general.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer AS C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 288, 290)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY CUBE(

T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID)

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;

/\*En el ejemplo siguiente, el operador GROUPING SETS tiene cuatro agrupaciones, una para cada columna de la lista SELECT. El operador devuelve una fila por cada valor único en las columnas Region, Country, Store, y SalesPersonID\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,DATEPART(yyyy,OrderDate) AS 'Year'

,DATEPART(mm,OrderDate) AS 'Month'

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND DATEPART(yyyy,OrderDate) = '2006'

GROUP BY CUBE(

(T.[Group], T.CountryRegionCode)

,(DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate)))

ORDER BY T.[Group], T.CountryRegionCode

,DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate);

/\*n el ejemplo siguiente, la lista GROUPING SETS contiene dos elementos compuestos, (T.[Group], T.CountryRegionCode) y (DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate)). Cada elemento compuesto se trata como una columna.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityId

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 288, 290)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY GROUPING SETS

(T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID)

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;

/\*En el ejemplo siguiente, la lista GROUPING SETS tiene cinco elementos. El conjunto de resultados tiene una fila para los elementos siguientes:

Cada combinación única de valores en las columnas Region y Country

Cada valor único en la columna Store

Cada combinación única de valores en las columnas SalesPersonID y Region

Cada valor único en la columna SalesPersonID

Un total general\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,DATEPART(yyyy,OrderDate) AS 'Year'

,DATEPART(mm,OrderDate) AS 'Month'

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND DATEPART(yyyy,OrderDate) = '2006'

GROUP BY GROUPING SETS(

(T.[Group], T.CountryRegionCode)

,(DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate)))

ORDER BY T.[Group], T.CountryRegionCode

,DATEPART(yyyy,OrderDate), DATEPART(mm,OrderDate);

/\*En el ejemplo siguiente, la lista GROUPING SETS incluye agrupaciones para las columnas T.[Group] y T.CountryRegionCode y un ROLLUP de las columnas S.Name y H.SalesPersonID.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 288, 290)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY GROUPING SETS(

(T.[Group], T.CountryRegionCode)

,(S.Name)

,(H.SalesPersonID,T.[Group])

,(H.SalesPersonID)

,())

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;

/\*En el ejemplo siguiente, la lista GROUPING SETS incluye agrupaciones para las columnas T.[Group] y T.CountryRegionCode y un CUBE de las columnas S.Name y H.SalesPersonID.\*/

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe'

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 288, 290)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY GROUPING SETS(

T.[Group], T.CountryRegionCode

,ROLLUP(S.Name, H.SalesPersonID))

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;

SELECT T.[Group] AS N'Region', T.CountryRegionCode AS N'Country'

,S.Name AS N'Store', H.SalesPersonID

,SUM(TotalDue) AS N'Total Sales'

FROM Sales.Customer C

INNER JOIN Sales.Store AS S

ON C.StoreID = S.BusinessEntityID

INNER JOIN Sales.SalesTerritory AS T

ON C.TerritoryID = T.TerritoryID

INNER JOIN Sales.SalesOrderHeader AS H

ON C.CustomerID = H.CustomerID

WHERE T.[Group] = N'Europe' /\*n sirve para indicar cual va a ser la excepcion del pais\*/

AND T.CountryRegionCode IN(N'DE', N'FR')

AND H.SalesPersonID IN(287, 288, 290)

AND SUBSTRING(S.Name,1,4)IN(N'Vers', N'Spa ')

GROUP BY GROUPING SETS(

T.[Group], T.CountryRegionCode

,CUBE(S.Name, H.SalesPersonID))

ORDER BY T.[Group], T.CountryRegionCode, S.Name, H.SalesPersonID;