

Práctica A: Procedimientos para agrupar y resumir datos

Objetivos

Después de realizar esta práctica, el alumno será capaz de:

- Utilizar las cláusulas GROUP BY y HAVING para resumir datos por grupos.
- Utilizar los operadores ROLLUP y CUBE, y la función GROUPING para generar datos de resumen.
- Utilizar las cláusulas COMPUTE y COMPUTE BY para generar informes con secciones, totales y promedios.

Requisitos previos

- El archivo lab052.zip contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.

Ejercicio 1

Uso de la palabra clave TOP *n*

En este ejercicio, va a utilizar la palabra clave TOP *n* y la cláusula WITH TIES para obtener las primeras filas, o un porcentaje de ellas, de un conjunto de resultados. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para utilizar la palabra clave TOP *n* y presentar las primeras filas de un conjunto de resultados

En este procedimiento, debe modificar una secuencia de comandos para que devuelva las diez primeras filas de una consulta. La secuencia de comandos completa para este procedimiento es Answer_TopN1.sql.

Abra el Analizador de consultas de SQL Server y, si así se le pide, inicie una sesión en el servidor local con autenticación de Microsoft Windows®.

En la lista **BD**, haga clic en **northwind**.

Abra y examine la secuencia de comandos TopN.sql, que corresponde a una consulta que calcula el importe total de las ventas de cada pedido de la tabla **order details** y devuelve los resultados en orden decreciente.

Modifique la consulta descrita en el paso 4 para que sólo devuelva las diez primeras filas.

```
USE northwind
SELECT TOP 10
    orderid
    ,(unitprice * quantity) AS totalsale
FROM [order details]
ORDER BY (unitprice * quantity) DESC
GO
```

Ejecute la consulta para comprobar que sólo devuelve diez filas.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

orderid	totalsale
10865	15810.0000
10981	15810.0000
10353	10540.0000
10417	10540.0000
10889	10540.0000
10424	10329.2000
10897	9903.2000
10372	8432.0000
10540	7905.0000
10816	7905.0000

(10 filas afectadas)

✍ **Para presentar los primeros valores de un conjunto de resultados usando la palabra clave TOP *n***

En este procedimiento va a utilizar la palabra clave TOP *n* para presentar los primeros valores de un conjunto de resultados. La secuencia de comandos completa para este procedimiento es Answer_TopN2.sql.

Modifique la consulta descrita en el paso 5 del procedimiento anterior para que devuelva los diez primeros productos (incluidos los valores iguales) que tengan el mayor importe total.

```
USE northwind
SELECT TOP 10 WITH TIES
       orderid
       ,(unitprice * quantity) AS totalsale
FROM [order details]
ORDER BY (unitprice * quantity) DESC
GO
```

Ejecute la consulta para comprobar que devuelve once filas.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

orderid	totalsale
10865	15810.0000
10981	15810.0000
10353	10540.0000
10417	10540.0000
10889	10540.0000
10424	10329.2000
10897	9903.2000
10372	8432.0000
10540	7905.0000
10816	7905.0000
10817	7905.0000

(11 filas afectadas)

Ejercicio 2

Uso de las cláusulas GROUP BY y HAVING

En este ejercicio, va a utilizar las cláusulas GROUP BY y HAVING para resumir datos de la base de datos **Northwind**. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para utilizar la cláusula GROUP BY para resumir los datos

En este procedimiento va a abrir una secuencia de comandos que contiene una consulta con la cláusula GROUP BY. A continuación, modificará la consulta para obtener resultados diferentes.

Abra y examine la secuencia de comandos Groupby.sql. Se trata de una consulta que calcula la cantidad total de artículos pedidos de dos categorías distintas de artículos de la tabla **order details**.

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

categoryid	total_quantity
1	9532
2	5298

(2 filas afectadas)

✍ Para calcular la cantidad total para cada una de las categorías

Answer_Groupby1.sql es la secuencia de comandos completa de este procedimiento.

Modifique la secuencia de comandos descrita en el paso 1 del procedimiento anterior para resumir la cantidad por categoría para todos los productos, sea cual sea su categoría.

```
USE northwind
SELECT categoryid, SUM(quantity) AS total_quantity
FROM [order details] AS od
INNER JOIN products AS p
ON od.productid = p.productid
GROUP BY categoryid
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

categoryid	total_quantity
1h	9532
2	5298
3	7906
4	9149
5	4562
6	4199
7	2990
8	7681

(8 filas afectadas)

✍ Para calcular la cantidad total para cada pedido

En este procedimiento calculará la cantidad total para cada pedido.

Answer_Groupby2.sql es la secuencia de comandos completa de este procedimiento.

Modifique la secuencia de comandos descrita en el paso 1 del procedimiento anterior para resumir la cantidad por **orderid** (pedido) para todos los productos, sea cual sea su categoría.

```
USE northwind
SELECT orderid, SUM(quantity) AS total_quantity
FROM [order details] AS od
INNER JOIN products AS p
ON od.productid = p.productid
GROUP BY orderid
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados parciales.

orderid	total_quantity
10248	27
10249	49
10250	60
.	
.	
.	
11075	42
11076	50
11077	72

(830 filas afectadas)

✍ Para calcular el número de pedidos con más de 250 unidades pedidas

En este procedimiento calculará el número de pedidos con más de 250 unidades pedidas. `Answer_Groupby3.sql` es la secuencia de comandos completa de este procedimiento.

Modifique la secuencia de comandos descrita en el paso 1 del procedimiento anterior para resumir la cantidad por **orderid** (pedido) para todos los productos, sea cual sea su categoría, y devolver sólo los pedidos que tengan más de 250 unidades pedidas

```
USE northwind
SELECT orderid, SUM(quantity) AS total_quantity
FROM [order details] AS od
INNER JOIN products AS p
ON od.productid = p.productid
GROUP BY orderid
HAVING SUM(quantity) > 250
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

orderid	total_quantity
10515	286
10612	263
10658	255
10678	280
10847	288
10895	346
10990	256
11030	330

(8 filas afectadas)

Ejercicio 3

Uso de los operadores ROLLUP y CUBE

En este ejercicio, va a utilizar los operadores ROLLUP y CUBE para generar datos de resumen. También va a utilizar la función GROUPING para determinar las filas del resultado que son resúmenes. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✎ Para utilizar el operador ROLLUP con el fin de generar resultados de resumen

En este procedimiento va a utilizar el operador ROLLUP con las cláusulas GROUP BY y HAVING para generar resultados de resumen. La secuencia de comandos completa para este procedimiento es Answer_Rollup1.sql.

Abra y examine la secuencia de comandos Answer_Rollup1.sql. Se trata de una consulta que resume la cantidad de artículos pedidos por cada **productid** y **ordered**, y realiza un cálculo acumulativo.

Modifique la consulta descrita en el paso 1 para que en el resultado sólo aparezca el número de producto 50, con una cláusula WHERE, y después ejecute la consulta.

```
USE northwind
SELECT productid, orderid, SUM(quantity) AS total_quantity
FROM [order details]
WHERE productid = 50
GROUP BY productid, orderid
WITH ROLLUP
ORDER BY productid, orderid
GO
```

Ejecute la consulta para revisar los resultados. Cuento las filas que tienen valores nulos.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

productid	orderid	total_quantity
NULL	NULL	235
50	NULL	235
50	10350	15
50	10383	15
50	10429	40
50	10465	25
50	10637	25
50	10729	40
50	10751	20
50	10920	24
50	10948	9
50	11072	22

(12 filas afectadas)

4. ¿Qué significan los valores nulos en las columnas **productid** y **orderid**?

Los valores nulos en una fila indican que el valor que aparece en la columna **total_quantity** de esa fila es la suma de todos los valores **total_quantity** sin agrupar en la columna que tiene el valor nulo.

Por ejemplo, el valor **total_quantity** en la fila en que **productid** y **orderid** tienen valor nulo es la suma de todos los valores **total_quantity** de la tabla.

✍ Para utilizar el operador CUBE con el fin de generar resultados de resumen

En este procedimiento va a usar el operador CUBE y la función GROUPING para diferenciar entre las filas de resumen y detalles del conjunto de resultados. La secuencia de comandos completa para este procedimiento es `Answer_Cube1.sql`.

Abra y examine la secuencia de comandos `Answer_Rollup1.sql`. Se trata de una consulta que resume la cantidad de artículos pedidos por cada **productid** y **orderid**, y realiza un cálculo acumulativo.

Modifique la consulta descrita en el paso 1 para usar el operador CUBE en lugar del operador ROLLUP. Además, utilice la función GROUPING en las columnas **productid** y **orderid** para distinguir filas de resumen y filas de detalle dentro del conjunto de resultados, y, después, ejecute la consulta.

```
USE northwind
SELECT productid
      ,GROUPING(productid)
      ,orderid
      ,GROUPING(orderid)
      ,SUM(quantity) AS total_quantity
FROM [order details]
WHERE productid = 50
GROUP BY productid, orderid
WITH CUBE
ORDER BY productid, orderid
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

productid		orderid		total_quantity
NULL	1	NULL	1	235
NULL	1	10350	0	15
NULL	1	10383	0	15
NULL	1	10429	0	40
NULL	1	10465	0	25
NULL	1	10637	0	25
NULL	1	10729	0	40
NULL	1	10751	0	20
NULL	1	10920	0	24
NULL	1	10948	0	9
NULL	1	11072	0	22
50	0	NULL	1	235
50	0	10350	0	15
50	0	10383	0	15
50	0	10429	0	40
50	0	10465	0	25
50	0	10637	0	25
50	0	10729	0	40
50	0	10751	0	20
50	0	10920	0	24
50	0	10948	0	9
50	0	11072	0	22

(22 filas afectadas)

¿Qué filas son de resumen?

Las filas que tienen el número 1 en las columnas de la función GROUPING.

¿Qué filas son el resumen por producto? ¿Y por pedido?

Si el número 1 está presente en la columna generada por la función GROUPING para la columna productid, la fila es un resumen por pedido. La columna productid de dicha fila es NULL porque es un resumen, no una fila de detalle que contiene un valor NULL. La fila que tiene un número 1 en la columna GROUPING es un resumen para el producto número 50. La fila que tiene un número 1 en las dos columnas GROUPING es un total global.

Ejercicio 4

Uso de las cláusulas COMPUTE y COMPUTE BY

En este ejercicio va a utilizar las cláusulas COMPUTE y COMPUTE BY para generar informes con secciones y totales, y promedios finales.

La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para utilizar la cláusula COMPUTE con el fin de generar informes

En este procedimiento va a modificar una consulta existente al agregar las cláusulas COMPUTE y COMPUTE BY para generar subtotales y totales finales. Answer_Compute1.sql es la secuencia de comandos completa de este procedimiento.

Abra y examine la secuencia de comandos Answer_Compute1.sql. Se trata de una consulta que devuelve el número de pedido (**orderid**) y la cantidad (**quantity**) pedida para todos los pedidos cuyo **orderid** > 11070.

Modifique la consulta descrita en el paso 1 para generar un total final de la columna cantidad (**quantity**) mediante el uso de la cláusula COMPUTE.

```
USE northwind
SELECT orderid, quantity
FROM [order details]
WHERE orderid >= 11070
COMPUTE SUM(quantity)
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados parciales.

orderid	quantity
11070	40
11070	20
11070	30
.	
.	
.	
11077	24
11077	4
11077	1
	Sum
	=====
	543

(45 filas afectadas)

✍ Para utilizar la cláusula COMPUTE BY con el fin de generar informes

En este procedimiento, va a modificar una consulta existente. Para ello, agregará la cláusula COMPUTE BY con el fin de generar totales finales. Answer_Compute2.sql es la secuencia de comandos completa de este procedimiento.

Abra y examine la secuencia de comandos Answer_Compute2.sql. Se trata de una consulta que devuelve el número de pedido (**orderid**) y la cantidad (**quantity**) pedida para todos los pedidos cuyo **orderid** > 11070.

Modifique la consulta descrita en el paso 1 para generar un informe con secciones que proporcione la cantidad total de los pedidos 11075 y 11076.

```
USE northwind
SELECT orderid, quantity
FROM [order details]
WHERE orderid in ( 11075, 11076 )
ORDER BY orderid
COMPUTE SUM(quantity) BY orderid
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados.

orderid	quantity
11075	10
11075	30
11075	2
	Sum
	=====
	42
11076	20
11076	20
11076	10
	Sum
	=====
	50

(8 filas afectadas)

✍ Para agregar la cantidad total y cantidad promedio al final del informe con secciones

En este procedimiento va a agregar la cantidad total y cantidad promedio al final del informe con secciones. Answer_Compute3.sql es la secuencia de comandos completa de este procedimiento.

Modifique la consulta descrita en el paso 1 del procedimiento anterior para agregar la cantidad total y cantidad promedio al final del informe con secciones.

```
USE northwind
SELECT orderid, quantity
FROM [order details]
WHERE orderid in ( 11075, 11076 )
ORDER BY orderid
COMPUTE SUM(quantity) BY orderid
COMPUTE SUM(quantity)
COMPUTE AVG(quantity)
GO
```

Ejecute la consulta para revisar los resultados.

Resultado

Su resultado será similar al siguiente conjunto de resultados. Observe que este conjunto de resultados es similar al del paso 3 del procedimiento anterior, con la adición de los totales finales (cantidad total y cantidad promedio).

orderid	quantity
11075	10
11075	30
11075	2
	Sum
	=====
	42
11076	20
11076	20
11076	10
	Sum
	=====
	50
	Sum
	=====
	92
	Avg
	=====
	15

(10 filas afectadas)