

Procedimientos para agrupar y resumir datos

Contenido

Introducción	1
Presentación de los primeros n valores	2
Uso de funciones de agregado	4
Fundamentos de GROUP BY	8
Generación de valores de agregado dentro de los conjuntos de resultados	13
Uso de las cláusulas COMPUTE y COMPUTE BY	22

Notas para el instructor

Este módulo proporciona a los alumnos los conocimientos necesarios para agrupar y resumir datos mediante funciones de agregado. Entre dichos conocimientos se incluye cómo usar las cláusulas `GROUP BY` y `HAVING` para resumir y agrupar datos, y los operadores `ROLLUP` y `CUBE` con la función `GROUPING` para agrupar datos y resumir valores para dichos grupos. Este módulo describe también la forma de usar las cláusulas `COMPUTE` y `COMPUTE BY` para generar informes de resumen y para mostrar los n primeros valores de un conjunto de resultados.

Al final de este módulo, los alumnos serán capaces de:

- Utilizar la palabra clave `TOP n` para obtener una lista de los primeros n valores especificados de una tabla.
- Generar un único valor de resumen mediante el uso de funciones de agregado.
- Organizar los datos de resumen de una columna mediante el uso de funciones de agregado con las cláusulas `GROUP BY` y `HAVING`.
- Generar datos de resumen de una tabla mediante el uso de funciones de agregado con la cláusula `GROUP BY` y los operadores `ROLLUP` o `CUBE`.
- Generar informes con secciones mediante el uso de cláusulas `COMPUTE` y `COMPUTE BY`.

Introducción

Objetivo del tema

Exponer un breve resumen de los temas tratados en este módulo.

Explicación previa

Puede ser interesante agrupar o resumir los datos cuando se obtienen.

- Presentación de los primeros n valores
- Uso de funciones de agregado
- Fundamentos de GROUP BY
- Generación de valores de agregado dentro de los conjuntos de resultados
- Uso de las cláusulas COMPUTE y COMPUTE BY

Puede ser interesante agrupar o resumir los datos cuando se obtienen.

Este módulo proporciona a los alumnos los conocimientos necesarios para agrupar y resumir datos mediante funciones de agregado. Entre dichos conocimientos se incluye cómo usar las cláusulas GROUP BY y HAVING para resumir y agrupar datos, y los operadores ROLLUP y CUBE con la función GROUPING para agrupar datos y resumir valores para dichos grupos. Este módulo describe también la forma de usar las cláusulas COMPUTE y COMPUTE BY para generar informes de resumen y para mostrar los n primeros valores de un conjunto de resultados.

Después de finalizar este módulo, el alumno será capaz de:

- Utilizar la palabra clave TOP n para obtener una lista de los primeros valores especificados de una tabla.
- Generar un único valor de resumen mediante el uso de funciones de agregado.
- Organizar los datos de resumen de una columna mediante el uso de funciones de agregado con las cláusulas GROUP BY y HAVING.
- Generar datos de resumen de una tabla mediante el uso de funciones de agregado con la cláusula GROUP BY y los operadores ROLLUP o CUBE.
- Generar informes con secciones mediante el uso de cláusulas COMPUTE y COMPUTE BY.

Presentación de los primeros n valores

Objetivo del tema

Describir cómo se presentan los primeros n valores de un resumen.

Explicación previa

Utilice la palabra clave TOP n para presentar sólo las n primeras filas de un conjunto de resultados.

- Presenta sólo las n primeras filas de un conjunto de resultados
- Especifica el intervalo de valores con la cláusula ORDER BY
- Devuelve las filas iguales si se utiliza WITH TIES

Ejemplo 1

```
USE northwind
SELECT TOP 5 orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
GO
```

Ejemplo 2

```
USE northwind
SELECT TOP 5 WITH TIES orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
GO
```

Nota del instructor

El uso de unos índices adecuados puede aumentar la eficacia de las operaciones de ordenación y agrupamiento. Este curso no trata en detalle acerca de los índices. Para obtener más información acerca del uso de índices, consulte el curso 2329A, *Programación de una base de datos Microsoft SQL Server 2000*.

Utilice la palabra clave TOP n para presentar sólo las n primeras filas o el n por ciento de un conjunto de resultados. Aunque la palabra clave TOP n no es un estándar ANSI, resulta útil, por ejemplo, para presentar los productos más vendidos de una compañía.

Cuando utilice la palabra clave TOP n o TOP n PERCENT, considere los hechos e instrucciones siguientes:

- Especifique el intervalo de valores en la cláusula ORDER BY. Si no utiliza una cláusula ORDER BY, Microsoft® SQL Server™ 2000 devuelve las filas que cumplen la cláusula WHERE sin ningún orden concreto.
- Utilice un entero sin signo a continuación de la palabra clave TOP.
- Si la palabra clave TOP n PERCENT produce un número no entero de filas, SQL Server redondea la cantidad no entera al siguiente valor entero.
- Utilice la cláusula WITH TIES para incluir las filas iguales en el conjunto de resultados. Las filas iguales se producen cuando hay dos o más filas con valores iguales a los de la última fila devuelta según la cláusula ORDER BY. Por lo tanto, el conjunto de resultados puede incluir cualquier número de filas.

Nota Sólo se puede utilizar la cláusula WITH TIES cuando existe una cláusula ORDER BY.

Ejemplo 1

Este ejemplo utiliza la palabra clave TOP n para buscar los cinco productos más vendidos en un mismo pedido. Los valores iguales siguientes quedan excluidos del conjunto de resultados.

```
USE northwind
SELECT TOP 5 orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
GO
```

Resultado

orderid	productid	quantity
10764	39	130
11072	64	130
10398	55	120
10451	55	120
10515	27	120

(5 filas afectadas)

Ejemplo 2

En este ejemplo se utiliza la palabra clave TOP n y la cláusula WITH TIES para buscar los cinco productos más vendidos dentro de un mismo pedido. El conjunto de resultados presenta un total de 10 productos, ya que se incluyen las filas adicionales que tienen los mismos valores que la última fila incluida. Compare el conjunto de resultados siguiente con el del ejemplo 1.

Sugerencia

Compare el conjunto de resultados siguiente con el del ejemplo 1.

```
USE northwind
SELECT TOP 5 WITH TIES orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
GO
```

Resultado

orderid	productid	quantity
10764	39	130
11072	64	130
10398	55	120
10451	55	120
10515	27	120
10595	61	120
10678	41	120
10711	53	120
10776	51	120
10894	75	120

(10 filas afectadas)

◆ Uso de funciones de agregado

Objetivo del tema

Demostrar el uso de funciones de agregado para producir datos de resumen.

Explicación previa

Utilice funciones de agregado para calcular valores a partir de columnas e incluir dichos valores en un conjunto de resultados.

<i>Función de agregado</i>	<i>Descripción</i>
AVG	Promedio de valores en una expresión numérica
COUNT	Número de valores en una expresión
COUNT (*)	Número de filas seleccionadas
MAX	Valor más alto en la expresión
MIN	Valor más bajo en la expresión
SUM	Valores totales en una expresión numérica
STDEV	Desviación estadística de todos los valores
STDEVP	Desviación estadística para la población
VAR	Varianza estadística de todos los valores
VARP	Varianza estadística de todos los valores para la población

Las funciones que calculan promedios y sumas se llaman funciones de agregado. Cuando se ejecuta una función de agregado, SQL Server resume los valores de toda una tabla o de grupos de columnas de una tabla, y produce un valor por cada conjunto de filas para las columnas especificadas:

- Las funciones de agregado se pueden utilizar en la instrucción **SELECT** o en combinación con la cláusula **GROUP BY**.
- Con la excepción de la función **COUNT(*)**, todas las funciones de agregado devuelven **NULL** si ninguna fila cumple la cláusula **WHERE**. La función **COUNT(*)** devuelve el valor cero si ninguna fila cumple la cláusula **WHERE**.

Sugerencia Para aumentar el rendimiento de las consultas, indice las columnas de agregado con frecuencia. Por ejemplo, si calcula frecuentemente la columna **quantity**, la indización de dicha columna mejora el rendimiento de las operaciones de agregado.

El tipo de datos de una columna determina las funciones que se pueden utilizar con ella. La tabla siguiente describe las relaciones entre las funciones y los tipos de datos.

	Función	Tipo de datos
	COUNT	COUNT es la única función de agregado que se puede utilizar en las columnas de los tipos de datos text , ntext o image .
	MIN y MAX	En las columnas con tipos de datos bit no se pueden utilizar las funciones MIN y MAX.
	SUM y AVG	Las funciones de agregado SUM y AVG sólo se pueden utilizar en las columnas con tipos de datos int , smallint , tinyint , decimal , numeric , float , real , money y smallmoney . Cuando se utiliza la función SUM o AVG, SQL Server trata los tipos de datos smallint o tinyint como un valor de tipo de datos int en el conjunto de resultados.
Sintaxis parcial	SELECT [ALL DISTINCT] [TOP <i>n</i> [PERCENT] [WITH TIES]] <listaSelección> [INTO <i>nuevaTabla</i>] [FROM <tablasOrigen >] [WHERE <condicionesBúsqueda>] [[GROUP BY [ALL] <i>expresiónAgrupación</i> [, ... <i>n</i>]] [HAVING <condicionesBúsqueda>] [WITH { CUBE ROLLUP }]] [ORDER BY { <i>columna</i> [ASC DESC] } [, ... <i>n</i>]] [COMPUTE { { AVG COUNT MAX MIN SUM } (<i>expresión</i>) } [, ... <i>n</i>] [BY <i>expresión</i> [, ... <i>n</i>]]	
Ejemplo 1	<p>En este ejemplo se calcula el precio promedio por unidad de todos los productos de la tabla products.</p> <pre>USE northwind SELECT AVG(unitprice) FROM products GO</pre>	
Resultado	<hr/> 28.8663 (1 fila afectada)	
Ejemplo 2	<p>En este ejemplo se suman todas las filas de la columna quantity de la tabla order details.</p> <pre>USE northwind SELECT SUM(quantity) FROM [order details] GO</pre>	
Resultado	<hr/> 51317 (1 fila afectada)	

Uso de funciones de agregado con valores nulos

Objetivo del tema

Explicar el comportamiento de los valores nulos cuando se utilizan con funciones de agregado.

Explicación previa

Puede recibir resultados inesperados si utiliza funciones de agregado con valores nulos.

- La mayoría de las funciones de agregado pasan por alto los valores nulos
- La función COUNT(*) cuenta las filas con valores nulos

Ejemplo 1

```
USE northwind
SELECT COUNT (*)
FROM employees
GO
```

Ejemplo 2

```
USE northwind
SELECT COUNT(reportsto)
FROM employees
GO
```

Los valores nulos pueden hacer que las funciones de agregado produzcan resultados inesperados. Por ejemplo, si ejecuta una instrucción SELECT que incluye una función COUNT en una columna que contiene 18 filas, dos de las cuales contienen valores nulos, el conjunto de resultados devuelve un total de 16 filas. SQL Server pasa por alto las dos filas que contienen valores nulos.

Por tanto, preste atención cuando utilice funciones de agregado en columnas que contengan valores nulos, ya que puede que el conjunto de resultados no sea representativo de los datos. Sin embargo, si decide utilizar funciones de agregado con valores nulos, considere los hechos siguientes:

- Las funciones de agregado de SQL Server, con la excepción de la función COUNT (*), pasan por alto los valores nulos en las columnas.
- La función COUNT (*) cuenta todas las filas, incluso si todas las columnas contienen un valor nulo. Por ejemplo, si ejecuta una instrucción SELECT que incluya la función COUNT (*) en una columna que contenga un total de 18 filas, dos de las cuales contienen valores nulos, el conjunto de resultados devuelve un total de 18 filas.

Ejemplo 1

En este ejemplo se presenta el número de empleados de la tabla **employees**.

```
USE northwind
SELECT COUNT(*)
FROM employees
GO
```

Resultado

9

(1 fila afectada)

Ejemplo 2

Este ejemplo presenta el número de empleados de la tabla **employees** que no tienen un valor nulo en la columna **reportsto**, lo que indica que se ha definido un superior inmediato para ese empleado.

```
USE northwind
SELECT COUNT(reportsto)
FROM employees
GO
```

Resultado

8

(1 fila afectada)

◆ Fundamentos de GROUP BY

Objetivo del tema

Proporcionar una introducción a las cláusulas que resumen los valores de una columna.

Explicación previa

Las funciones de agregado se suelen utilizar junto con las cláusulas GROUP BY y HAVING.

- Uso de la cláusula GROUP BY
- Uso de la cláusula GROUP BY con la cláusula HAVING

Por sí mismas, las funciones de agregado sólo producen un valor de resumen para todas las filas de una columna.

Si desea generar valores de resumen para una sola columna, utilice funciones de agregado con la cláusula GROUP BY. Utilice la cláusula HAVING con la cláusula GROUP BY para restringir los grupos de filas devueltas en el conjunto de resultados.

Nota El uso de la cláusula GROUP BY no garantiza que se vaya a usar ningún criterio de ordenación. Si desea que los resultados se ordenen, debe incluir la cláusula ORDER BY.

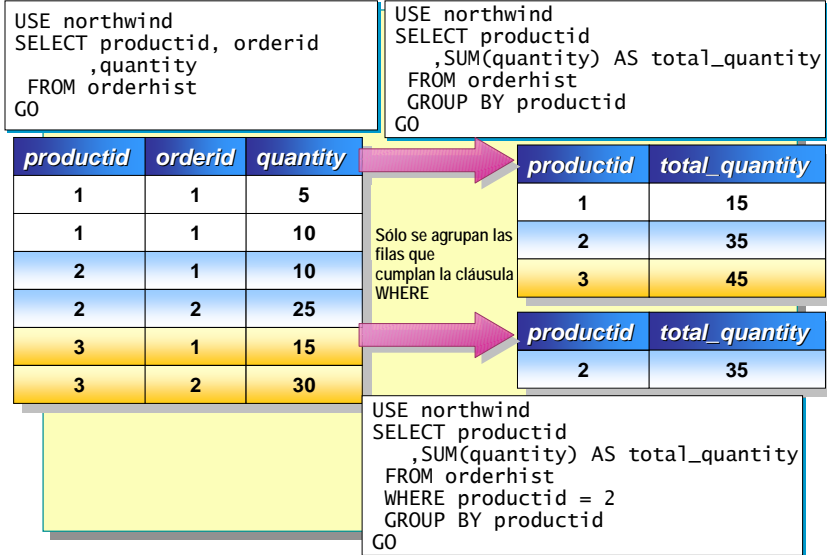
Uso de la cláusula GROUP BY

Objetivo del tema

Explicar cómo se utiliza la cláusula GROUP BY para resumir datos.

Explicación previa

Utilice la cláusula GROUP BY en columnas o expresiones para organizar filas en grupos y para resumir dichos grupos.



Utilice la cláusula GROUP BY en columnas o expresiones para organizar filas en grupos y para resumir dichos grupos. Por ejemplo, utilice la cláusula GROUP BY para determinar la cantidad de cada producto pedida en todos los pedidos.

Sugerencia

La tabla **orderhist** se ha creado específicamente para los ejemplos de este módulo. También se incluye en el disco compacto Material del alumno.

Compare los conjuntos de resultados de la diapositiva. La tabla de la izquierda presenta todas las filas de la tabla **orderhist**.

La tabla de la parte superior derecha utiliza la cláusula GROUP BY para agrupar todos los datos de la columna **productid** y presentar la cantidad total pedida para cada uno de los grupos.

La tabla de la parte inferior derecha utiliza la cláusula GROUP BY y la cláusula WHERE para restringir el número de filas devuelto.

Cuando utilice la cláusula GROUP BY, considere los hechos e instrucciones siguientes:

- SQL Server produce una columna de valores por cada grupo definido.
- SQL Server sólo devuelve filas por cada grupo especificado; no devuelve información de detalle.
- Todas las columnas que se especifican en la cláusula GROUP BY tienen que estar incluidas en la lista de selección.
- Si incluye una cláusula WHERE, SQL Server sólo agrupa las filas que cumplen las condiciones de la cláusula WHERE.
- En la lista de columnas de la cláusula GROUP BY puede haber hasta 8.060 bytes.
- No utilice la cláusula GROUP BY en columnas que contengan varios valores nulos, porque los valores nulos se procesan como otro grupo.
- Utilice la palabra clave ALL con la cláusula GROUP BY para presentar todas las filas que tengan valores nulos en las columnas de agregado, independientemente de si las filas cumplen la condición de la cláusula WHERE.

Nota La tabla **orderhist** se ha creado específicamente para los ejemplos de este módulo. La secuencia de comandos 2317A_R04.sql, que se incluye en el disco compacto Material del alumno, se puede ejecutar para agregar dicha tabla a la base de datos **Northwind**.

Ejemplo 1

En este ejemplo se devuelve información acerca de los pedidos de la tabla **orderhist**. La consulta agrupa y presenta cada identificador de producto y calcula la cantidad total pedida. La cantidad total se calcula con la función de agregado SUM y presenta un valor para cada producto del conjunto de resultados.

```
USE northwind
SELECT productid, SUM(quantity) AS total_quantity
  FROM orderhist
  GROUP BY productid
GO
```

Resultado

productid	total_quantity
1	15
2	35
3	45

(3 filas afectadas)

Ejemplo 2

Este ejemplo agrega una cláusula WHERE a la consulta del ejemplo 1. Esta consulta restringe las filas al producto cuyo identificador es 2 y, después, agrupa dichas filas y calcula la cantidad total pedida. Compare este conjunto de resultados con el del ejemplo 1.

```
USE northwind
SELECT productid, SUM(quantity) AS total_quantity
  FROM orderhist
  WHERE productid = 2
  GROUP BY productid
GO
```

Resultado

productid	total_quantity
2	35

(1 fila afectada)

Ejemplo 3

En este ejemplo se devuelve información acerca de los pedidos de la tabla **order details**. Esta consulta agrupa y presenta los identificadores de los productos y, después, calcula la cantidad total pedida. La cantidad total se calcula con la función de agregado SUM y presenta un valor para cada producto del conjunto de resultados. En este ejemplo no se incluye una cláusula WHERE y, por tanto, se devuelve un total por cada identificador de producto.

```
USE northwind
SELECT productid, SUM(quantity) AS total_quantity
  FROM [order details]
  GROUP BY productid
GO
```

Resultado

productid	total_quantity
61	603
3	328
32	297
.	
.	
.	

(77 filas afectadas)

Uso de la cláusula GROUP BY con la cláusula HAVING

Objetivo del tema

Explicar cómo se utiliza la cláusula HAVING para resumir más los datos, en función de los grupos.

Explicación previa

Puede utilizar la cláusula HAVING para establecer condiciones en los grupos que se incluyen en un conjunto de resultados.

```
USE northwind
SELECT productid, orderid, quantity
FROM orderhist
GO
```

productid	orderid	quantity
1	1	5
1	1	10
2	1	10
2	2	25
3	1	15
3	2	30

```
USE northwind
SELECT productid, SUM(quantity)
AS total_quantity
FROM orderhist
GROUP BY productid
HAVING SUM(quantity)>=30
GO
```

productid	total_quantity
2	35
3	45

Utilice la cláusula HAVING en columnas o expresiones para establecer condiciones en los grupos incluidos en un conjunto de resultados. La cláusula HAVING establece condiciones en la cláusula GROUP BY de una forma muy similar a como interactúa la cláusula WHERE con la instrucción SELECT.

Cuando utilice la cláusula HAVING, considere los hechos e instrucciones siguientes:

- Utilice la cláusula HAVING sólo con la cláusula GROUP BY para restringir los agrupamientos. El uso de la cláusula HAVING sin la cláusula GROUP BY no tiene sentido.
- En una cláusula HAVING puede haber hasta 128 condiciones. Cuando utilice varias condiciones, tiene que combinarlas con operadores lógicos (AND, OR o NOT).
- Puede hacer referencia a cualquiera de las columnas que aparezcan en la lista de selección.
- No utilice la palabra clave ALL con la cláusula HAVING, porque la cláusula HAVING pasa por alto la palabra clave ALL y sólo devuelve los grupos que cumplen la cláusula HAVING.

Sugerencia

Destaque la condición de búsqueda definida en la cláusula HAVING del ejemplo de la diapositiva.

La tabla de la derecha agrupa todos los datos de la columna **productid** pero sólo presenta la cantidad total pedida para los grupos que cumplen la condición de búsqueda de la cláusula HAVING.

Ejemplo 1

En este ejemplo se presentan todos los grupos de productos de la tabla **orderhist** que tienen pedidos de 30 unidades o más.

```
USE northwind
SELECT productid, SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid
HAVING SUM(quantity) >=30
GO
```

Resultado

productid	total_quantity
2	35
3	45

(2 filas afectadas)

Ejemplo 2

En este ejemplo se presenta el identificador de producto y la cantidad para todos los productos que tienen pedidos de más de 1.200 unidades.

```
USE northwind
SELECT productid, SUM(quantity) AS total_quantity
FROM [order details]
GROUP BY productid
HAVING SUM(quantity) > 1200
GO
```

Resultado

productid	total_quantity
59	1496
56	1263
60	1577
31	1397

(4 filas afectadas)

◆ Generación de valores de agregado dentro de los conjuntos de resultados

Objetivo del tema

Proporcionar una introducción a los valores de resumen de una tabla mediante los operadores ROLLUP y CUBE.

Explicación previa

Utilice la cláusula GROUP BY con los operadores ROLLUP y CUBE para generar valores de agregado dentro de los conjuntos de resultados. Si lo hace, utilizará la función GROUPING para interpretar el conjunto de resultados.

- Uso de la cláusula GROUP BY con el operador ROLLUP
- Uso de la cláusula GROUP BY con el operador CUBE
- Uso de la función GROUPING

Utilice la cláusula GROUP BY con los operadores ROLLUP y CUBE para generar valores de agregado dentro de los conjuntos de resultados. Los operadores ROLLUP o CUBE pueden ser útiles para obtener información de referencias cruzadas dentro de una tabla sin tener que escribir secuencias de comandos adicionales.

Cuando utilice los operadores ROLLUP o CUBE, use la función GROUPING para identificar los valores de detalle y de resumen dentro del conjunto de resultados.

Uso de la cláusula GROUP BY con el operador ROLLUP

Objetivo del tema

Explicar cómo se utiliza el operador ROLLUP para resumir los datos de una tabla.

Explicación previa

Utilice el operador ROLLUP para resumir los datos de una tabla.

```
USE northwind
SELECT productid, orderid, SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid, orderid
WITH ROLLUP
ORDER BY productid, orderid
GO
```

productid	orderid	total_quantity	Descripción
NULL	NULL	95	Total general
1	NULL	15	Resume sólo las filas para productid 1
1	1	5	Detalla el valor para productid 1, orderid 1
1	2	10	Detalla el valor para productid 1, orderid 2
2	NULL	35	Resume sólo las filas para productid 2
2	1	10	Detalla el valor para productid 2, orderid 1
2	2	25	Detalla el valor para productid 2, orderid 2
3	NULL	45	Resume sólo las filas para productid 3
3	1	15	Detalla el valor para productid 3, orderid 1
3	2	30	Detalla el valor para productid 3, orderid 2

Sugerencia

Explique que los valores NULL del ejemplo de la diapositiva indican que dichas filas concretas son el resultado del operador ROLLUP.

Utilice la cláusula GROUP BY con el operador ROLLUP para resumir valores de grupos. La cláusula GROUP BY y el operador ROLLUP proporcionan datos en un formato relacional estándar.

Por ejemplo, se puede generar un conjunto de resultados que incluya la cantidad pedida de cada producto en cada pedido, la cantidad total pedida por cada producto y el total final de todos los productos.

Cuando utilice la cláusula GROUP BY con el operador ROLLUP, considere los hechos e instrucciones siguientes:

- SQL Server procesa los datos de derecha a izquierda en la lista de columnas especificadas en la cláusula GROUP BY. Después, SQL Server aplica la función de agregado a cada grupo.
- SQL Server agrega al conjunto de resultados una fila que presenta los cálculos acumulados, como un total o un promedio acumulado. Dichos cálculos acumulados se indican en el conjunto de resultados con un valor NULL.
- Cuando utiliza el operador ROLLUP puede tener hasta 10 expresiones de agrupación.
- Con el operador ROLLUP no se puede utilizar la palabra clave ALL.
- Cuando utilice el operador ROLLUP, asegúrese de que las columnas que siguen a la cláusula GROUP BY tienen una relación significativa en su entorno de trabajo.

Ejemplo 1

En este ejemplo se presentan todas las filas de la tabla **orderhist** y los valores de cantidades de resumen de cada producto.

Sugerencia

Estos ejemplos están generados uno a partir de otro, para que se pueda comprender cómo ROLLUP complementa a GROUP BY.

```
USE northwind
SELECT productid, orderid, SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid, orderid
WITH ROLLUP
ORDER BY productid, orderid
GO
```

Resultado

productid	orderid	total_quantity
NULL	NULL	95
1	NULL	15
1	1	5
1	2	10
2	NULL	35
2	1	10
2	2	25
3	NULL	45
3	1	15
3	2	30

(10 filas afectadas)

Ejemplo 2

En este ejemplo se devuelve información acerca de los pedidos de la tabla **order details**. Esta consulta contiene una instrucción SELECT con una cláusula GROUP BY sin el operador ROLLUP. El ejemplo devuelve la lista de las cantidades totales de cada producto solicitadas en cada pedido, para los pedidos cuyo **orderid** sea menor que 10250.

```
USE northwind
SELECT orderid, productid, SUM(quantity) AS total_quantity
FROM [order details]
WHERE orderid < 10250
GROUP BY orderid, productid
ORDER BY orderid, productid
GO
```

Resultado

orderid	productid	total_quantity
10248	11	12
10248	42	10
10248	72	5
10249	14	9
10249	51	40

(5 filas afectadas)

Ejemplo 3

En este ejemplo se agrega el operador ROLLUP a la instrucción del ejemplo 2. El conjunto de resultados incluye la cantidad total para:

- Cada producto en cada pedido (también devuelto por la cláusula GROUP BY sin el operador ROLLUP).
- Todos los productos de cada pedido.
- Todos los productos de todos los pedidos (total final).

Observe en el conjunto de resultados que la fila que contiene NULL en las columnas **productid** y **orderid** representa la cantidad total final de todos los pedidos para todos los productos. Las filas que contienen NULL en la columna **productid** representan la cantidad total de un producto en el pedido de la columna **orderid**.

```
USE northwind
SELECT orderid, productid, SUM(quantity) AS total_quantity
FROM [order details]
WHERE orderid < 10250
GROUP BY orderid, productid
WITH ROLLUP
ORDER BY orderid, productid
GO
```

Resultado

orderid	productid	total_quantity
NULL	NULL	76
10248	NULL	27
10248	11	12
10248	42	10
10248	72	5
10249	NULL	49
10249	14	9
10249	51	40

(8 filas afectadas)

Uso de la cláusula GROUP BY con el operador CUBE

Objetivo del tema

Explicar cómo se utiliza el operador CUBE para resumir los datos de una tabla.

Explicación previa

El operador CUBE difiere del operador ROLLUP en que el primero crea todas las combinaciones posibles de los grupos en función de la cláusula GROUP BY y, después, aplica las funciones de agregado.

```
USE northwind
SELECT productid, orderid, SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid, orderid
WITH CUBE
ORDER BY productid, orderid
GO
```

El operador CUBE produce dos resúmenes más de valores que el operador ROLLUP

productid	orderid	total_quantity	Descripción
NULL	NULL	95	Total general
NULL	1	30	Resume todas las filas para orderid 1
NULL	2	65	Resume todas las filas para orderid 2
1	NULL	15	Resume sólo las filas para productid 1
1	1	5	Detalla el valor para productid 1, orderid 1
1	2	10	Detalla el valor para productid 1, orderid 2
2	NULL	35	Resume sólo las filas para productid 2
2	1	10	Detalla el valor para productid 2, orderid 1
2	2	25	Detalla el valor para productid 2, orderid 2
3	NULL	45	Resume sólo las filas para productid 3
3	1	15	Detalla el valor para productid 3, orderid 1
3	2	30	Detalla el valor para productid 3, orderid 2

Sugerencia

Señale que los valores NULL del conjunto de resultados del ejemplo de la diapositiva indican que dichas filas concretas se crean como resultado del operador CUBE.

Utilice la cláusula GROUP BY con el operador CUBE para crear y resumir todas las combinaciones posibles de los grupos en función de la cláusula GROUP BY. Utilice la cláusula GROUP BY con el operador ROLLUP para proporcionar datos en un formato relacional estándar.

Cuando utilice la cláusula GROUP BY con el operador CUBE, considere los hechos e instrucciones siguientes:

- Si tiene n columnas o expresiones en la cláusula GROUP BY, SQL Server devuelve las 2^{n-1} combinaciones posibles en el conjunto de resultados.
- Los valores NULL del conjunto de resultados indican que dichas filas concretas son el resultado del operador CUBE.
- Cuando utilice el operador CUBE, puede incluir hasta 10 expresiones de agrupamiento.
- Con el operador CUBE no se puede utilizar la palabra clave ALL.
- Cuando utilice el operador CUBE, asegúrese de que las columnas que siguen a la cláusula GROUP BY tienen una relación significativa en su entorno de trabajo.

Ejemplo

En este ejemplo se devuelve un resultado que proporciona la cantidad de cada producto en cada pedido, la cantidad total de todos los productos de cada pedido, la cantidad total de cada producto en todos los pedidos y la cantidad total final de todos los productos en todos los pedidos.

```
USE northwind
SELECT productid,orderid,SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid,orderid
WITH CUBE
ORDER BY productid,orderid
GO
```

Resultado

productid	orderid	total_quantity
NULL	NULL	95
NULL	1	30
NULL	2	65
1	NULL	15
1	1	5
1	2	10
2	NULL	35
2	1	10
2	2	25
3	NULL	45
3	1	15
3	2	30

(12 filas afectadas)

Uso de la función GROUPING

Objetivo del tema

Explicar cómo funciona la función GROUPING.

Explicación previa

Utilice la función GROUPING con los operadores ROLLUP o CUBE para distinguir los valores de detalle y de resumen en un conjunto de resultados.

```
USE northwind
SELECT productid, GROUPING (productid)
,orderid, GROUPING (orderid)
,SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid, orderid
WITH CUBE
ORDER BY productid, orderid
GO
```

	productid		orderid		total_quantity
	NULL	1	NULL	1	95
	NULL	1	1	0	30
		1	2	0	65
1 representa los valores de resumen en la columna precedente	1	0	NULL	1	15
	1	0	1	0	5
		0	2	0	10
0 representa los valores de detalle en la columna precedente	2	0	NULL	1	35
	2	0	1	0	10
	2	0	2	0	25
	3	0	NULL	1	45
	3	0	1	0	15
	3	0	2	0	30

Sugerencia

Señale que el conjunto de resultados del ejemplo de la diapositiva es similar al de la diapositiva anterior con una excepción importante: se utiliza la función GROUPING y se incluyen dos columnas adicionales en el conjunto de resultados. El 1 representa valores de resumen y el 0 representa valores de detalle en la columna precedente.

Utilice la función GROUPING con los operadores ROLLUP o CUBE para distinguir los valores de detalle y de resumen en un conjunto de resultados. El uso de la función GROUPING ayuda a determinar si los valores NULL que aparecen en un conjunto de resultados son realmente valores nulos de las tablas base o si la fila que los contiene ha sido generada por los operadores ROLLUP o CUBE.

Cuando utilice la función GROUPING, considere los hechos e instrucciones siguientes:

- SQL Server produce nuevas columnas en el conjunto de resultados para cada columna especificada en la función GROUPING.
- SQL Server devuelve el valor 1 para representar los valores de resumen de ROLLUP o CUBE del conjunto de resultados.
- SQL Server devuelve el valor 0 para representar los valores de detalle del conjunto de resultados.
- Sólo se puede especificar la función GROUPING en las columnas que aparezcan en la cláusula GROUP BY.
- Utilice la función GROUPING para facilitar las referencias a los conjuntos de resultados desde los programas.

Ejemplo 1

En este ejemplo se devuelve un resultado que proporciona la cantidad de cada producto en cada pedido, la cantidad total de todos los productos de cada pedido, la cantidad total de cada producto en todos los pedidos y la cantidad total final de todos los productos en todos los pedidos. La función **GROUPING** distingue las filas del conjunto de resultados que han sido generadas por el operador **CUBE**.

```
USE northwind
SELECT productid, GROUPING (productid)
      ,orderid, GROUPING (orderid)
      ,SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid, orderid
WITH CUBE
ORDER BY productid, orderid
GO
```

Resultado

productid		orderid		total_quantity
NULL	1	NULL	1	95
NULL	1	1	0	30
NULL	1	2	0	65
1	0	NULL	1	15
1	0	1	0	5
1	0	2	0	10
2	0	NULL	1	35
2	0	1	0	10
2	0	2	0	25
3	0	NULL	1	45
3	0	1	0	15
3	0	2	0	30

(12 filas afectadas)

Ejemplo 2

En este ejemplo se utiliza la función **GROUPING** en las columnas **productid** y **orderid** que aparecen en la cláusula **GROUP BY**. El conjunto de resultados tiene una columna adicional después de las columnas **productid** y **orderid**. La función **GROUPING** devuelve un 1 cuando los valores de dicha columna concreta han sido agrupados por el operador **CUBE**. El conjunto de resultados incluye la cantidad total de cada producto en cada pedido, de cada producto en todos los pedidos, de todos los productos en cada pedido y la cantidad total de todos los productos en todos los pedidos.

Observe en el conjunto de resultados que las filas que contienen valores **NULL** en las columnas **productid** y **orderid** representan la cantidad total final de todos los productos en todos los pedidos. Las filas que contienen **NULL** en la columna **productid** representan la cantidad total de todos los productos en cada pedido. Las filas que contienen **NULL** en la columna **orderid** representan la cantidad total de un producto en todos los pedidos.

```
USE northwind
SELECT orderid, GROUPING(orderid), productid
      ,GROUPING(productid), SUM(quantity) AS total_quantity
FROM [order details]
WHERE orderid < 10250
GROUP BY orderid, productid
WITH CUBE
ORDER BY orderid, productid
GO
```

Resultado

	orderid		productid		total_quantity
	NULL	1	NULL	1	76
	NULL	1	11	0	12
	NULL	1	14	0	9
	NULL	1	42	0	10
	NULL	1	51	0	40
	NULL	1	72	0	5
	10248	0	NULL	1	27
	10248	0	11	0	12
	10248	0	42	0	10
	10248	0	72	0	5
	10249	0	NULL	1	49
	10249	0	14	0	9
	10249	0	51	0	40

(13 filas afectadas)

Uso de las cláusulas COMPUTE y COMPUTE BY

Objetivo del tema

Explicar el propósito de las cláusulas COMPUTE y COMPUTE BY.

Explicación previa

Aunque las cláusulas COMPUTE y COMPUTE BY no son un estándar ANSI, puede que le interese utilizarlas para presentar informes básicos o para comprobar los resultados de las aplicaciones que esté desarrollando.

COMPUTE			COMPUTE BY		
<pre>USE northwind SELECT productid, orderid, quantity FROM orderhist ORDER BY productid, orderid COMPUTE SUM(quantity) GO</pre>			<pre>USE northwind SELECT productid, orderid, quantity FROM orderhist ORDER BY productid, orderid COMPUTE SUM(quantity) BY productid COMPUTE SUM(quantity) GO</pre>		
productid	orderid	quantity	productid	orderid	quantity
1	1	5	1	1	5
1	2	10	1	2	10
2	1	10	sum		15
2	2	25	2	1	10
3	1	15	2	2	25
3	2	30	sum		35
sum		95	3	1	15
			3	2	30
			sum		45
			sum		95

Sugerencia

Estas cláusulas no se recomiendan para generar aplicaciones. Sin embargo, pueden ser útiles para probarlas.

Las cláusulas COMPUTE y COMPUTE BY generan filas de resumen adicionales en un formato no relacional que no se adaptan al estándar ANSI. Aunque presentan datos útiles, su salida no está adaptada para generar conjuntos de resultados que se vayan a utilizar en otras aplicaciones.

Por ejemplo, puede que le interese utilizar COMPUTE y COMPUTE BY para presentar informes básicos y rápidos, o para comprobar los resultados de las aplicaciones que esté desarrollando. Sin embargo, otras herramientas, como Crystal Reports o Microsoft Access, ofrecen características más completas para la generación de informes.

Si utiliza las cláusulas COMPUTE y COMPUTE BY, considere los hechos siguientes:

- En las cláusulas COMPUTE o COMPUTE BY no se pueden incluir tipos de datos **text**, **ntext** o **image**.
- No se puede ajustar el formato del conjunto de resultados. Por ejemplo, si utiliza la función de agregado SUM, SQL Server presenta la palabra *sum* en el conjunto de resultados. No se puede cambiar para que presente *resumen*.

Generación de informes con valores de detalle y de resumen para una columna

La cláusula COMPUTE produce filas de detalle y un solo valor de agregado para una columna. Cuando utilice la cláusula COMPUTE, considere los hechos y directrices siguientes:

- En una misma instrucción se pueden utilizar varias cláusulas COMPUTE con la cláusula COMPUTE BY.
- SQL Server requiere que se especifiquen las mismas columnas en la lista de selección que en la cláusula COMPUTE.
- No utilice la instrucción SELECT INTO en la misma instrucción que la cláusula COMPUTE, puesto que las instrucciones que incluyen COMPUTE no generan una salida relacional.

Ejemplo 1

En este ejemplo se presentan todas las filas de la tabla **orderhist** y se genera el total final de todos los productos pedidos.

```
USE northwind
SELECT productid, orderid, quantity
FROM orderhist
ORDER BY productid, orderid
COMPUTE SUM(quantity)
GO
```

Resultado

productid	orderid	total_quantity
1	1	5
1	2	10
2	1	10
2	2	25
3	1	15
3	2	30
		sum
		=====
		95

(7 filas afectadas)

Generación de informes con valores de detalle y de resumen para subconjuntos de grupos

La cláusula **COMPUTE BY** genera filas de detalle y varios valores de resumen. Los valores de resumen se generan cuando cambian los valores de la columna. Utilice la cláusula **COMPUTE BY** con datos que se puedan clasificar con facilidad. Cuando utilice la cláusula **COMPUTE BY**, considere los hechos e instrucciones siguientes:

- Para que las filas estén agrupadas, con la cláusula **COMPUTE BY** tiene que utilizar una cláusula **ORDER BY**.
- Especifique los nombres de las columnas después de la cláusula **COMPUTE BY** para determinar qué valores de resumen son generados por SQL Server.
- Las columnas que aparezcan después de la cláusula **COMPUTE BY** tienen que ser idénticas a un subconjunto de las columnas que aparezcan después de la cláusula **ORDER BY**. Tienen que estar en el mismo orden (de izquierda a derecha), comenzar con la misma expresión y no saltar ninguna de las expresiones.

Ejemplo 2

En este ejemplo se presentan todas las filas de la tabla **orderhist**, se genera el total pedido para cada producto y el total final de todos los productos pedidos.

```
USE northwind
SELECT productid, orderid, quantity
FROM orderhist
ORDER BY productid, orderid
COMPUTE SUM(quantity) BY productid
COMPUTE SUM(quantity)
GO
```

Resultado

productid	orderid	total_quantity
1	1	5
1	2	10
		sum
		=====
		15
2	1	10
2	2	25
		sum
		=====
		35
3	1	15
3	2	30
		sum
		=====
		45
		sum
		=====
		95
(10 filas afectadas)		