

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TEMA:

**PRACTICA DE LABORATORIO Nro 03 - Consultas con  
Pivot y Grouping Sets**

**CURSO:**

BASE DE DATOS II

**DOCENTE(ING):**

Patrick Jose Cuadros Quiroga

Integrantes:

Wilfredo Vilca Chambilla

(2006028540)

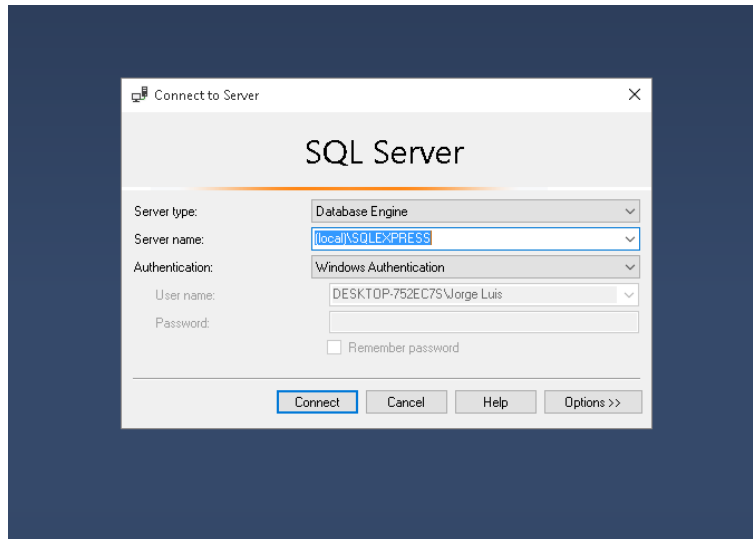
# Índice

1. Ejercicio 1: Escribiendo consultas con el operador PIVOT	1
2. Ejercicio 2: Escribiendo consultas que usan el operador UNPIVOT	12
3. Ejercicio 3: Escribiendo consultas que usan las clausulas JUEGOS DE GRUPO, CUBO, y ENROLLAR	16

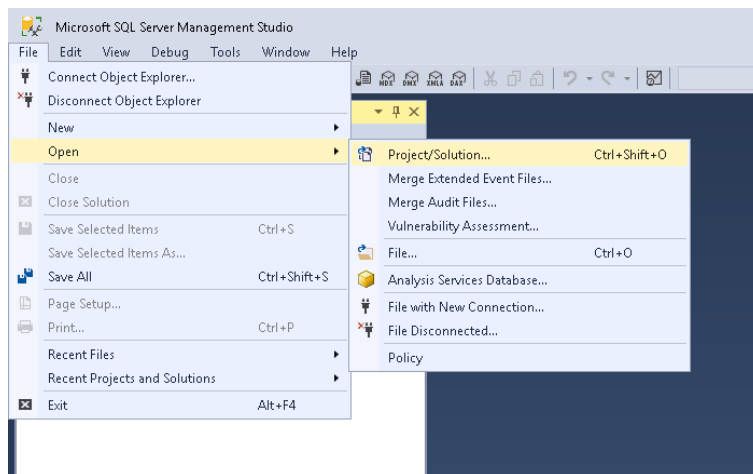
# 1. Ejercicio 1: Escribiendo consultas con el operador PIVOT

**Tarea 1:** Escriba una declaración **SELECT** para recuperar el número de clientes para un específico grupo de clientes

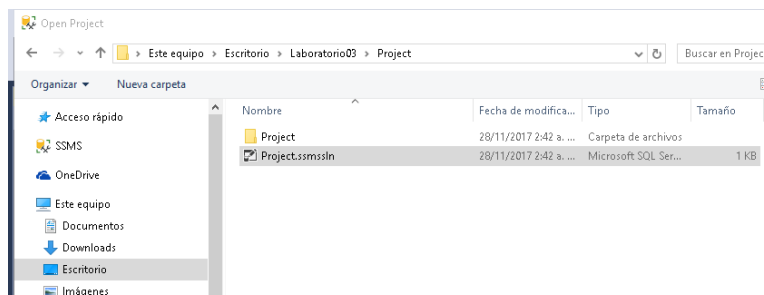
Paso 1. Inicie SQL Server Management Studio y conéctese al motor de base de datos (local) usando Windows autenticación.



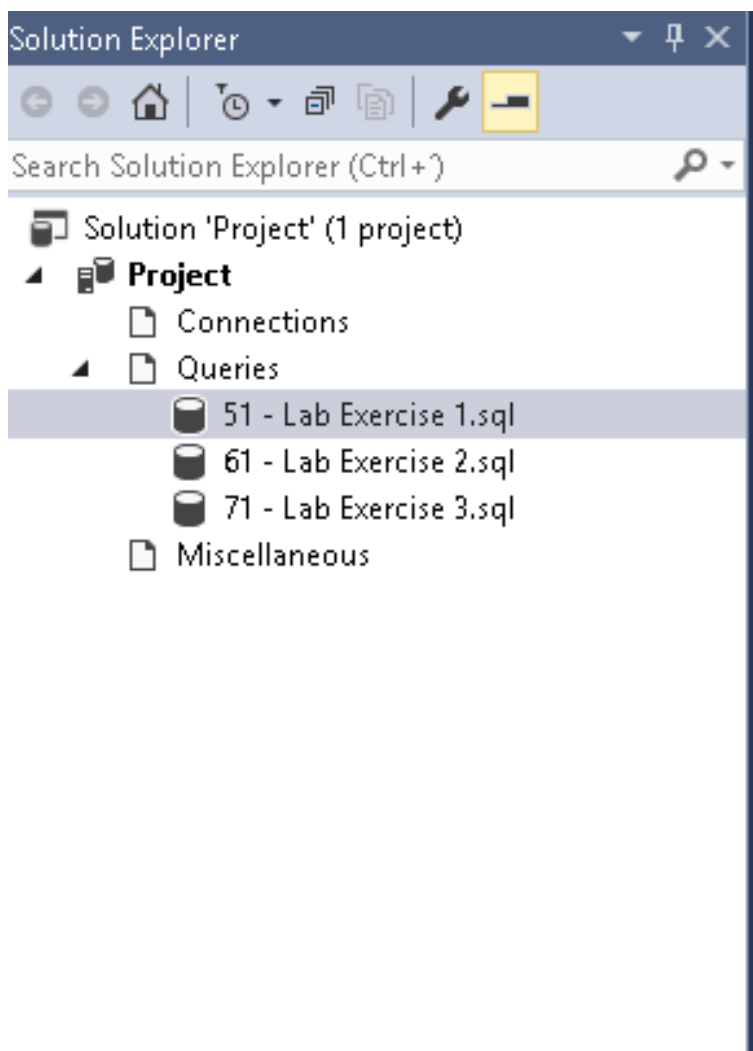
Paso 2. En el menú Archivo, haga clic en Abrir y haga clic en Proyecto / Solución.



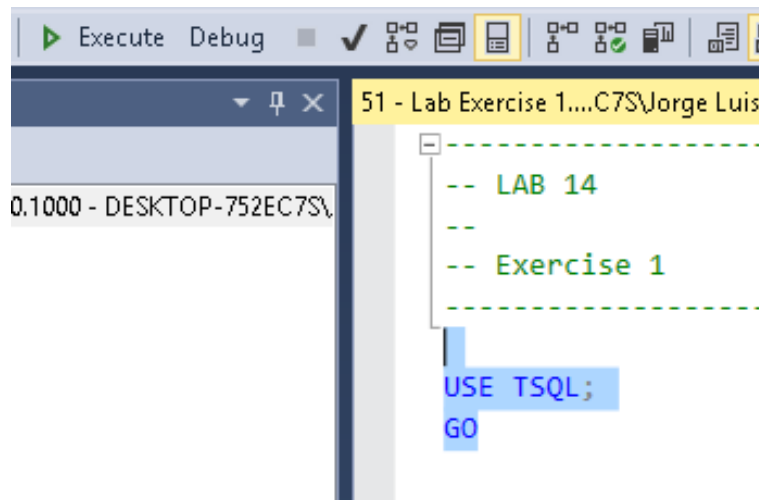
Paso 3. En la ventana Abrir proyecto, abra el proyecto Proyecto.ssmssl.n.



Paso 4. En el Explorador de soluciones, haga doble clic en la consulta 51 - Ejercicio de laboratorio 1.sql.



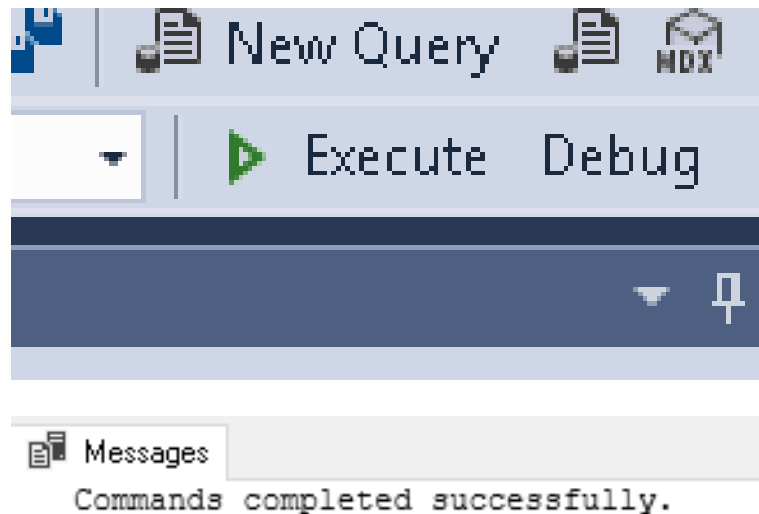
Paso 5. En la ventana de consulta, resalte la instrucción USE TSQL; y haga clic en Ejecutar.



Paso 6. Resalte el siguiente código T-SQL proporcionado:

```
-- Execute the modified statement and compare the results that
-----
CREATE VIEW Sales.CustGroups AS
SELECT
    custid,
    CHOOSE(custid % 3 + 1, N'A', N'B', N'C') AS custgroup,
    country
FROM Sales.Customers;
GO
```

Paso 7. Haga clic en Ejecutar. Este código crea una vista llamada Sales.CustGroups



Paso 8. En el panel de consulta, escriba la siguiente consulta después del código T-SQL proporcionado:

```

SELECT
custid,
custgroup,
country
FROM Sales.CustGroups;

```

Paso 9. Resalte la consulta escrita y haga clic en Ejecutar.

```

SELECT
custid,
custgroup,
country
FROM Sales.CustGroups;

```

Results		Messages	
	custid	custgroup	country
1	1	B	Germany
2	2	C	Mexico
3	3	A	Mexico
4	4	B	UK
5	5	C	Sweden
6	6	A	Germany
7	7	B	France
8	8	C	Spain
9	9	A	France
10	10	B	Canada
11	11	C	UK
12	12	A	Argentina
13	13	B	Mexico
14	14	C	Switzerland
15	15	A	Brazil
16	16	B	UK
17	17	C	Germany
18	18	A	France
19	19	B	UK

Paso 10. Modifique el código T-SQL escrito aplicando el operador PIVOT. La consulta debería verse así:

```

SELECT
country,
p.A,
p.B,
p.C
FROM Sales.CustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;

```

Paso 11. Resalte la consulta escrita y haga clic en Ejecutar.

```

SELECT
country,
p.A,
p.B,
p.C
FROM Sales.CustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;

```

	country	A	B	C
1	Argentina	0	1	0
2	Argentina	1	0	0
3	Argentina	1	0	0
4	Austria	0	0	1
5	Austria	0	0	1
6	Belgium	0	0	1
7	Belgium	0	1	0
8	Brazil	0	1	0
9	Brazil	0	1	0
10	Brazil	0	1	0
11	Brazil	0	1	0
12	Brazil	0	1	0
13	Brazil	0	0	1
14	Brazil	1	0	0
15	Brazil	1	0	0
16	Brazil	1	0	0
17	Canada	1	0	0
18	Canada	0	1	0
19	Canada	1	0	0

## Tarea 2: Especifique el elemento de agrupación para el operador PIVOT

Paso 1. Resalte el siguiente código T-SQL proporcionado después de la descripción de la Tarea 2:

```

ALTER VIEW Sales.CustGroups AS
SELECT
    custid,
    CHOOSE(custid % 3 + 1, N'A', N'B', N'C') AS custgroup,
    country,
    city,
    contactname
FROM Sales.Customers;
GO

```

Paso 2. Haga clic en Ejecutar. Este código modifica la vista agregando dos columnas adicionales.

```

ALTER VIEW Sales.CustGroups AS
SELECT
    custid,
    CHOOSE(custid % 3 + 1, N'A', N'B', N'C') AS custgroup,
    country,
    city,
    contactname
FROM Sales.Customers;
GO

```

Paso 3. Resalte la última consulta en la tarea 1. En la barra de herramientas, haga clic en Editar y luego en Copiar

```

SELECT
country,
p.A,
p.B,
p.C
FROM Sales.CustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;

```

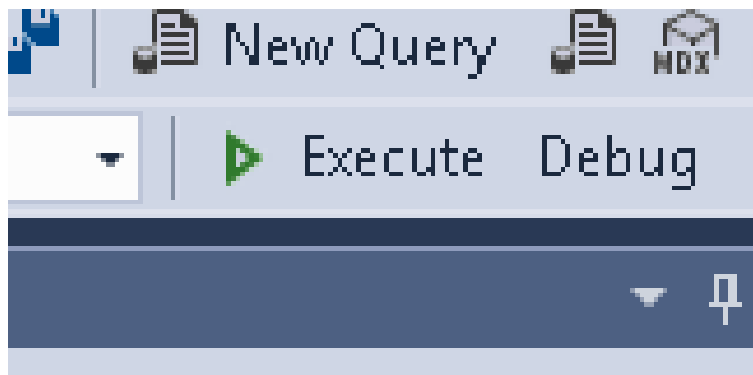
Paso 4. En la ventana de consulta, haga clic en la línea después del código T-SQL proporcionado. En la barra de herramientas, haga clic en Editar y luego en Pegar.

```

SELECT
country,
p.A,
p.B,
p.C
FROM Sales.CustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;

```

Paso 5. Resalte la consulta copiada y haga clic en Ejecutar.



Paso 6. Observa el resultado. ¿Es este resultado el mismo que el de la consulta en la tarea 1?

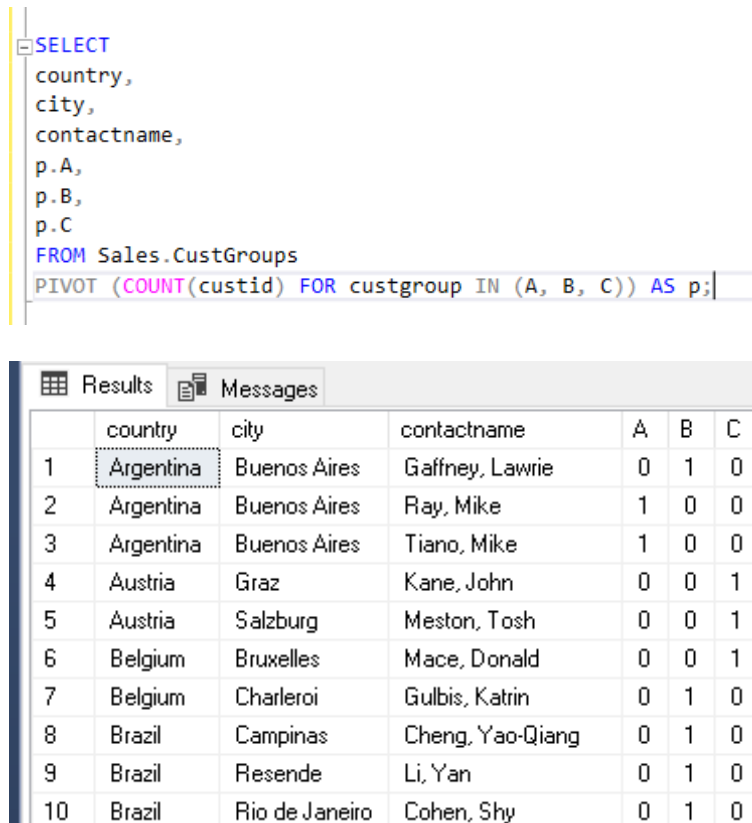


- El resultado no es el mismo. Más filas fueron devueltas después de que modificó la vista.

Paso 7. Modifique la sentencia T-SQL copiada para incluir columnas adicionales de la vista. La consulta debe se parece a esto:

```
SELECT
country,
city,
contactname,
p.A,
p.B,
p.C
FROM Sales.CustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;
```

Paso 8. Resalte la consulta escrita y haga clic en Ejecutar.



```
SELECT
country,
city,
contactname,
p.A,
p.B,
p.C
FROM Sales.CustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;
```

	country	city	contactname	A	B	C
1	Argentina	Buenos Aires	Gaffney, Lawrie	0	1	0
2	Argentina	Buenos Aires	Ray, Mike	1	0	0
3	Argentina	Buenos Aires	Tiano, Mike	1	0	0
4	Austria	Graz	Kane, John	0	0	1
5	Austria	Salzburg	Meston, Tosh	0	0	1
6	Belgium	Bruxelles	Mace, Donald	0	0	1
7	Belgium	Charleroi	Gulbis, Katrin	0	1	0
8	Brazil	Campinas	Cheng, Yao-Qiang	0	1	0
9	Brazil	Resende	Li, Yan	0	1	0
10	Brazil	Rio de Janeiro	Cohen, Shy	0	1	0

Paso 9. Observe que recibió el mismo resultado que la consulta anterior. ¿Por qué obtuviste el mismo número de filas?

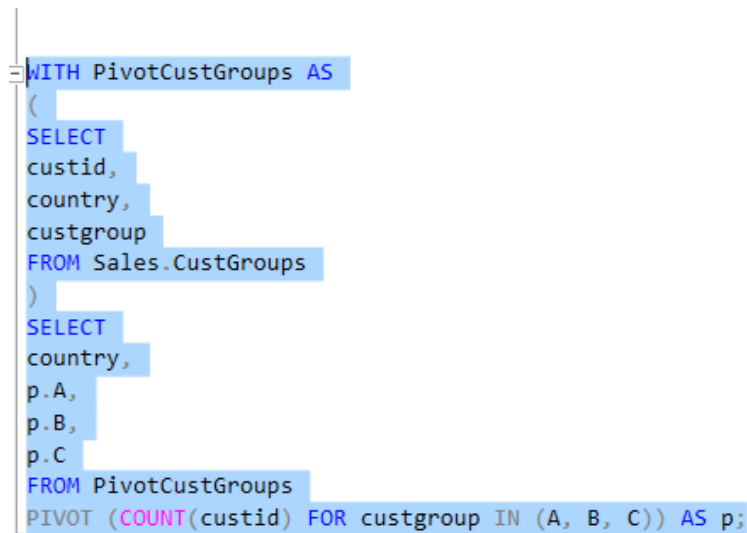
- El operador PIVOT asume que todas las columnas, excepto la agregación y la propagación Los elementos forman parte de las columnas de agrupación.

**Tarea 3: usar una expresión de tabla común (CTE) para especificar el elemento de agrupación para el operador PIVOT**

Paso 1. En el panel de consulta, escriba la siguiente consulta después de la descripción de la Tarea 3:

```
WITH PivotCustGroups AS
(
SELECT
custid,
country,
custgroup
FROM Sales.CustGroups
)
SELECT
country,
p.A,
p.B,
p.C
FROM PivotCustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;
```

Paso 2. Resalte la consulta escrita y haga clic en Ejecutar



```
WITH PivotCustGroups AS
(
SELECT
custid,
country,
custgroup
FROM Sales.CustGroups
)
SELECT
country,
p.A,
p.B,
p.C
FROM PivotCustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;
```

Paso 3. Observa el resultado. ¿Es el mismo que el resultado de la última consulta en la tarea 1? ¿Puedes explicar por qué?

- El resultado es el mismo. En esta tarea, el CTE ha proporcionado tres columnas posibles al operador PIVOT. En tarea 1, la vista también proporcionó tres columnas al operador PIVOT.

	country	A	B	C
1	Argentina	0	1	0
2	Argentina	1	0	0
3	Argentina	1	0	0
4	Austria	0	0	1
5	Austria	0	0	1
6	Belgium	0	0	1
7	Belgium	0	1	0
8	Brazil	0	1	0
9	Brazil	0	1	0
10	Brazil	0	1	0
11	Brazil	0	1	0
12	Brazil	0	1	0
13	Brazil	0	0	1
14	Brazil	1	0	0
15	Brazil	1	0	0
16	Brazil	1	0	0
17	Canada	1	0	0
18	Canada	0	1	0
19	Canada	1	0	0

	country	A	B	C
1	Argentina	2	1	0
2	Austria	0	0	2
3	Belgium	0	1	1
4	Brazil	3	5	1
5	Canada	2	1	0
6	Denmark	0	1	1
7	Finland	2	0	0
8	France	4	3	4
9	Germany	3	4	4
10	Ireland	0	1	0
11	Italy	2	1	0
12	Mexico	1	2	2
13	Norway	0	1	0
14	Poland	0	1	0
15	Portugal	1	1	0
16	Spain	2	1	2
17	Sweden	1	0	1
18	Switzerland	0	0	2
19	UK	1	3	3

Paso 4. ¿Por qué cree que es beneficioso utilizar un CTE cuando se usa el operador PIVOT?

Al usar el operador PIVOT, usted no puede especificar directamente el elemento de agrupación porque SQL Server automáticamente asume que todas las columnas deben usarse como elementos de agrupación, con la excepción de la propagación y elementos de agregación. Con un CTE, puede especificar las columnas exactas y, por lo tanto, controlar que Columnas utilizadas para la agrupación.

**Tarea 4: Escriba una instrucción SELECT para recuperar el monto total de ventas para cada uno de Categoría de cliente y producto**

Paso 1. En el panel de consulta, escriba la siguiente consulta después de la descripción de la Tarea 4.

```

WITH SalesByCategory AS
(
    SELECT
        o.custid,
        d.qty * d.unitprice AS salesvalue,
        c.categoryname
    FROM Sales.Orders AS o
    INNER JOIN Sales.OrderDetails AS d ON o.orderid = d.orderid
    INNER JOIN Production.Products AS p ON p.productid = d.productid
    INNER JOIN Production.Categories AS c ON c.categoryid = p.categoryid
    WHERE o.orderdate >= '20080101' AND o.orderdate < '20090101'
)
SELECT
    custid,
    p.Beverages,
    p.Condiments,
    p.Confections,
    p.[Dairy Products],
    p.[Grains/Cereals],
    p.[Meat/Poultry],
    p.Produce,
    p.Seafood
FROM SalesByCategory
PIVOT (SUM(salesvalue) FOR categoryname
IN (Beverages, Condiments, Confections, [Dairy Products], [Grains/Cereals],
[Meat/Poultry], Produce, Seafood)) AS p;

```

Paso 2. Resalte la consulta escrita y haga clic en Ejecutar.

```

WITH SalesByCategory AS
(
    SELECT
        o.custid,
        d.qty * d.unitprice AS salesvalue,
        c.categoryname
    FROM Sales.Orders AS o
    INNER JOIN Sales.OrderDetails AS d ON o.orderid = d.orderid
    INNER JOIN Production.Products AS p ON p.productid = d.productid
    INNER JOIN Production.Categories AS c ON c.categoryid = p.categoryid
    WHERE o.orderdate >= '20080101' AND o.orderdate < '20090101'
)
SELECT
    custid,
    p.Beverages,
    p.Condiments,
    p.Confections,
    p.[Dairy Products],
    p.[Grains/Cereals],
    p.[Meat/Poultry],
    p.Produce,
    p.Seafood
FROM SalesByCategory
PIVOT (SUM(salesvalue) FOR categoryname
IN (Beverages, Condiments, Confections, [Dairy Products], [Grains/Cereals],
[Meat/Poultry], Produce, Seafood)) AS p;

```

Results		Messages							
	custid	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood
1	1	NULL	426.00	NULL	1255.00	NULL	NULL	91.20	530.00
2	2	NULL	NULL	64.40	390.00	NULL	NULL	NULL	60.00
3	3	380.00	NULL	NULL	NULL	280.00	NULL	NULL	NULL
4	4	282.00	NULL	4440.00	812.50	NULL	NULL	NULL	304.00
5	5	850.50	300.00	2202.55	NULL	NULL	1237.90	1368.00	2151.60
6	6	NULL	114.00	283.00	714.00	NULL	NULL	424.00	625.00
7	7	NULL	NULL	NULL	437.50	292.50	NULL	NULL	NULL
8	8	NULL	NULL	NULL	NULL	280.00	NULL	NULL	NULL
9	9	533.00	1750.00	1515.10	556.80	665.00	624.00	705.00	837.00
10	10	1706.50	1290.10	4518.30	992.50	684.00	234.00	1872.00	930.00
11	11	1380.00	NULL	NULL	220.00	441.00	NULL	120.00	270.00
12	12	1037.00	NULL	NULL	25.00	NULL	NULL	364.80	150.00
13	14	570.00	1843.80	591.60	NULL	1216.00	NULL	NULL	208.00
14	15	NULL	405.75	NULL	NULL	NULL	NULL	NULL	108.00
15	16	NULL	640.50	NULL	NULL	NULL	291.00	NULL	NULL
16	17	NULL	NULL	374.76	1650.00	NULL	656.00	NULL	128.85
17	18	140.00	NULL	NULL	NULL	NULL	585.00	NULL	135.10
18	19	726.00	1082.55	374.76	3546.00	NULL	840.00	3000.00	NULL
19	20	4610.00	3667.50	2324.50	14141.50	7013.00	1560.00	3732.00	5550.40

## 2. Ejercicio 2: Escribiendo consultas que usan el operador UNPIVOT

### Tarea 1: Crear y consultar la vista Sales.PivotCustGroups

Paso 1. En el Explorador de soluciones, haga doble clic en la consulta 611.

Paso 2. En la ventana de consulta, resalte la instrucción USE TSQL, y haga clic en Ejecutar.

Paso 3. Resalte el siguiente código T-SQL proporcionado:

```
CREATE VIEW Sales.PivotCustGroups AS
WITH PivotCustGroups AS
(
    SELECT
        custid,
        country,
        custgroup
    FROM Sales.CustGroups
)
SELECT
    country,
    p.A,
    p.B,
    p.C
FROM PivotCustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;
GO
```

Paso 4. Haga clic en Ejecutar. Este código crea una vista llamada Sales.PivotCustGroups.

```
CREATE VIEW Sales.PivotCustGroups AS
WITH PivotCustGroups AS
(
    SELECT
        custid,
        country,
        custgroup
    FROM Sales.CustGroups
)
SELECT
    country,
    p.A,
    p.B,
    p.C
FROM PivotCustGroups
PIVOT (COUNT(custid) FOR custgroup IN (A, B, C)) AS p;
GO
```

100 %

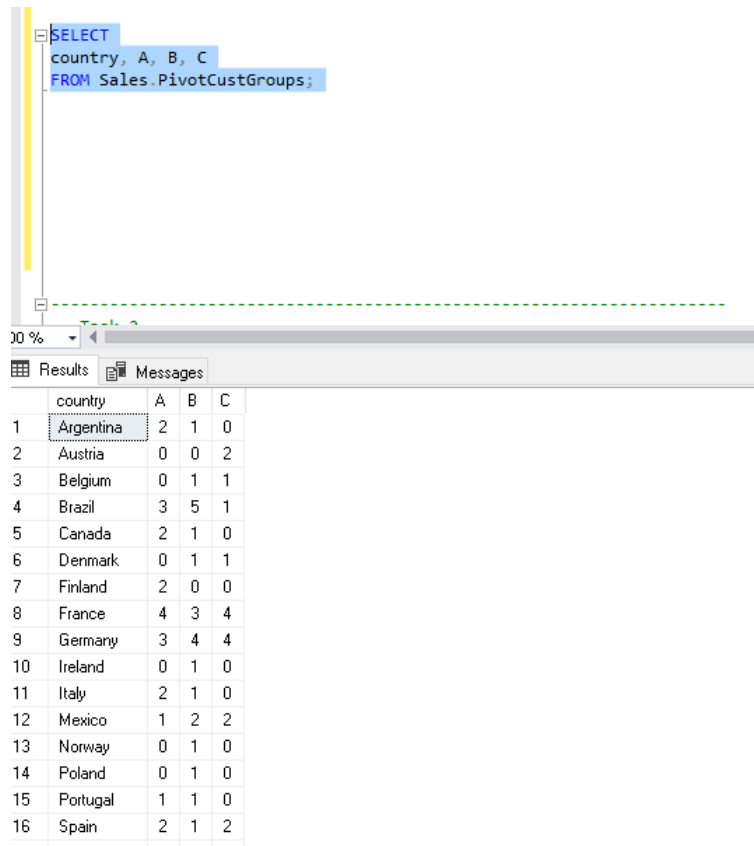
Messages

Commands completed successfully.

Paso 5. En el panel de consulta, escriba la siguiente consulta después del código T-SQL proporcionado:

```
SELECT  
country, A, B, C  
FROM Sales.PivotCustGroups;
```

Paso 6. Resalte la consulta escrita y haga clic en Ejecutar.



The screenshot shows a SQL Server Enterprise Manager interface. At the top, a query window displays the following T-SQL query, which is highlighted in blue:

```
SELECT  
country, A, B, C  
FROM Sales.PivotCustGroups;
```

Below the query window, the 'Results' tab is active, showing a grid of data. The grid has four columns: 'country', 'A', 'B', and 'C'. The data is as follows:

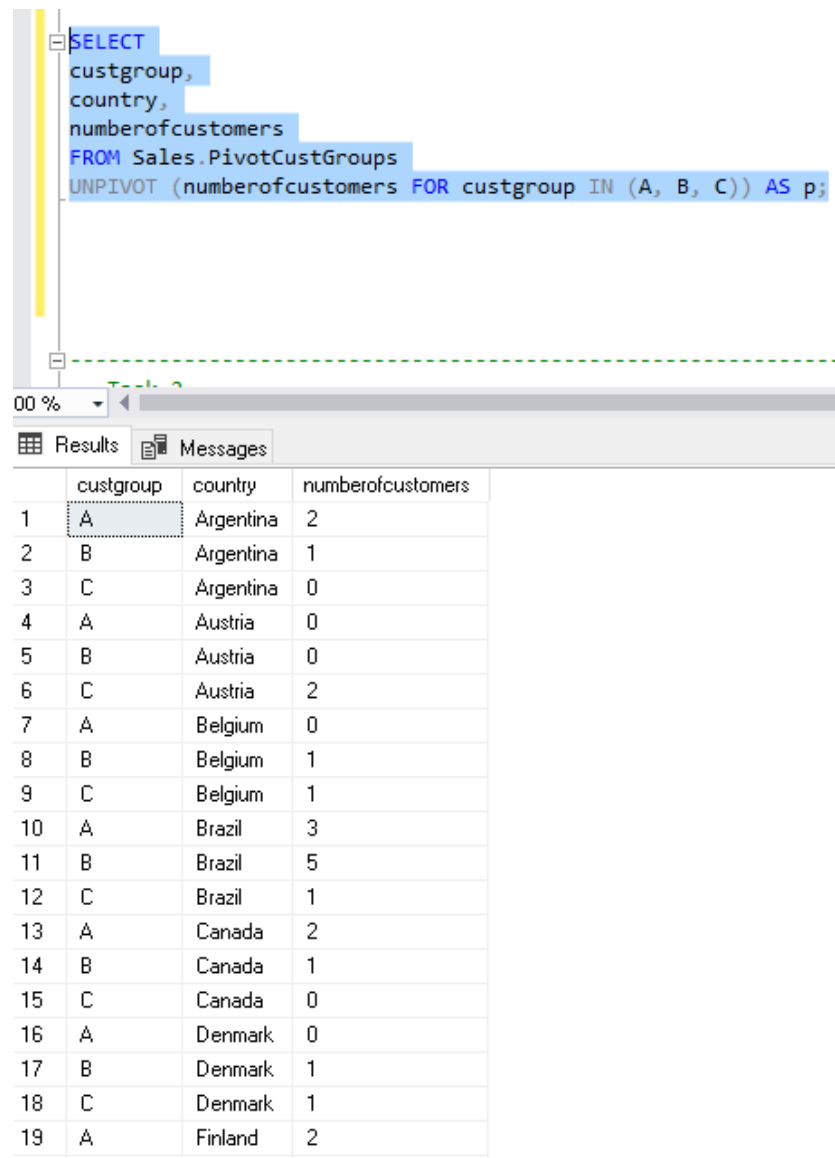
	country	A	B	C
1	Argentina	2	1	0
2	Austria	0	0	2
3	Belgium	0	1	1
4	Brazil	3	5	1
5	Canada	2	1	0
6	Denmark	0	1	1
7	Finland	2	0	0
8	France	4	3	4
9	Germany	3	4	4
10	Ireland	0	1	0
11	Italy	2	1	0
12	Mexico	1	2	2
13	Norway	0	1	0
14	Poland	0	1	0
15	Portugal	1	1	0
16	Spain	2	1	2

**Tarea 2:** escriba una instrucción SELECT para recuperar una fila para cada país y cliente grupo

Paso 1. En el panel de consulta, escriba la siguiente consulta después de los descriptores de la Tarea

```
SELECT
custgroup,
country,
numberofcustomers
FROM Sales.PivotCustGroups
UNPIVOT (numberofcustomers FOR custgroup IN (A, B, C)) AS p;
```

Paso 2. Resalte la consulta escrita y haga clic en Ejecutar.



The screenshot shows the SQL Server Enterprise Manager interface. The query editor at the top contains the following SQL query:

```
SELECT
custgroup,
country,
numberofcustomers
FROM Sales.PivotCustGroups
UNPIVOT (numberofcustomers FOR custgroup IN (A, B, C)) AS p;
```

Below the query editor, the Results pane displays the output of the query. The Results pane has tabs for "Results" and "Messages". The "Results" tab is active, showing a table with 19 rows and 3 columns: custgroup, country, and numberofcustomers. The first row is highlighted.

	custgroup	country	numberofcustomers
1	A	Argentina	2
2	B	Argentina	1
3	C	Argentina	0
4	A	Austria	0
5	B	Austria	0
6	C	Austria	2
7	A	Belgium	0
8	B	Belgium	1
9	C	Belgium	1
10	A	Brazil	3
11	B	Brazil	5
12	C	Brazil	1
13	A	Canada	2
14	B	Canada	1
15	C	Canada	0
16	A	Denmark	0
17	B	Denmark	1
18	C	Denmark	1
19	A	Finland	2

### Tarea 3: Eliminar las vistas creadas



- Resalte la instrucción T-SQL proporcionada después de la descripción de la Tarea 3 y haga clic en Ejecutar.

```
DROP VIEW Sales.CustGroups;  
DROP VIEW Sales.PivotCustGroups;
```

### 3. Ejercicio 3: Escribiendo consultas que usan las cláusulas JUEGOS DE GRUPO, CUBO, y ENROLLAR

Tarea 1: escriba una instrucción SELECT que use la subcláusula GROUPING SETS para devolver el número de Clientes para diferentes conjuntos de agrupación

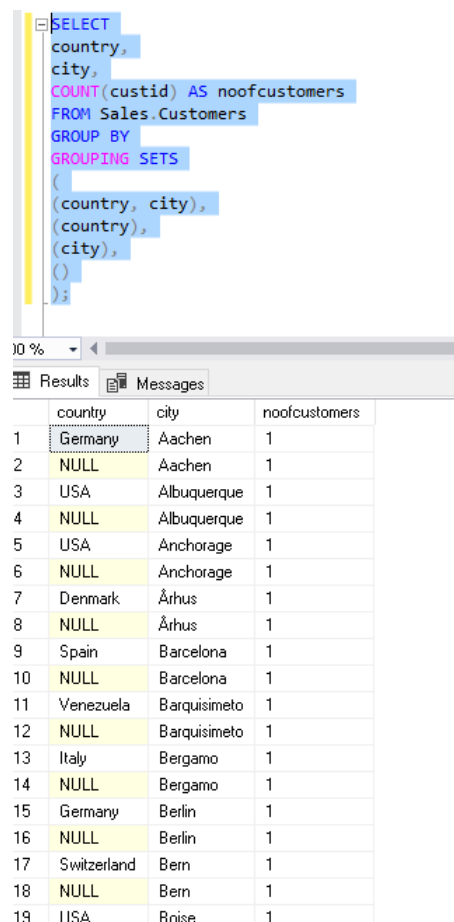
Paso 1. En el Explorador de soluciones, haga doble clic en la consulta 71 - Ejercicio de laboratorio 3.sql.

Paso 2. En la ventana de consulta, resalte la instrucción USE TSQL; y haga clic en Ejecutar.

Paso 3. En el panel de consulta, escriba la siguiente consulta después de la descripción de la Tarea 1:

```
SELECT
country,
city,
COUNT(custid) AS noofcustomers
FROM Sales.Customers
GROUP BY
GROUPING SETS
(
(country, city),
(country),
(city),
()
);
```

Paso 4. Resalte la consulta escrita y haga clic en Ejecutar.



```
SELECT
country,
city,
COUNT(custid) AS noofcustomers
FROM Sales.Customers
GROUP BY
GROUPING SETS
(
(country, city),
(country),
(city),
()
);
```

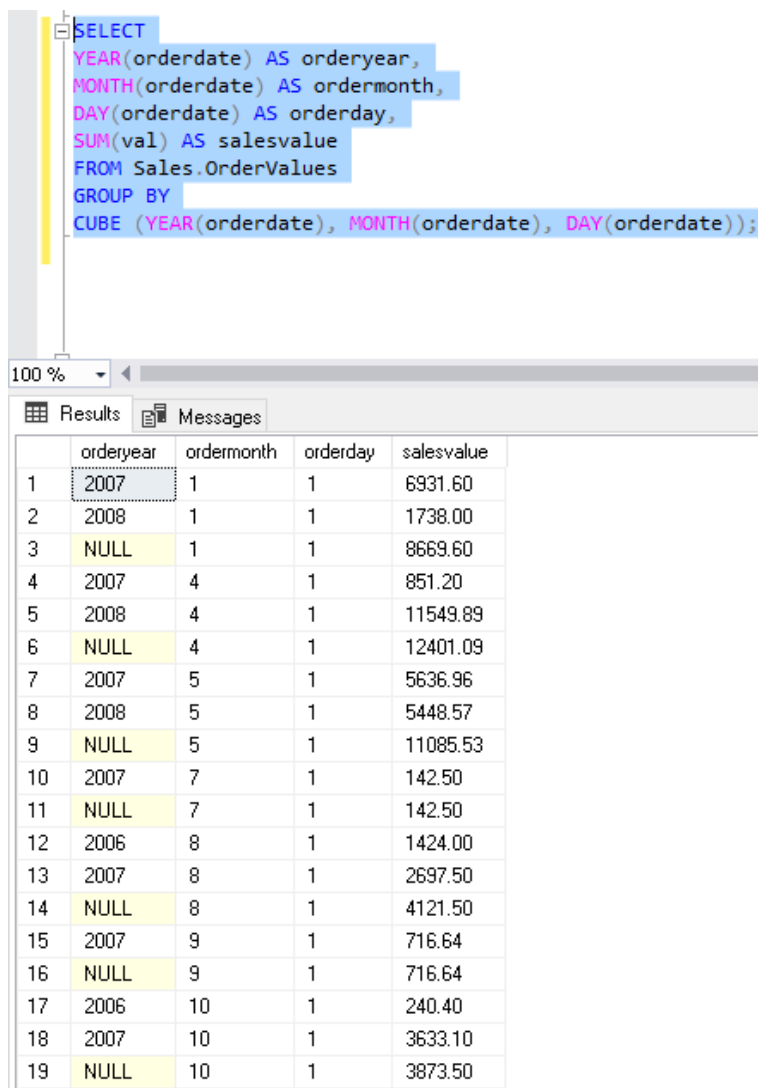
	country	city	noofcustomers
1	Germany	Aachen	1
2	NULL	Aachen	1
3	USA	Albuquerque	1
4	NULL	Albuquerque	1
5	USA	Anchorage	1
6	NULL	Anchorage	1
7	Denmark	Århus	1
8	NULL	Århus	1
9	Spain	Barcelona	1
10	NULL	Barcelona	1
11	Venezuela	Barquisimeto	1
12	NULL	Barquisimeto	1
13	Italy	Bergamo	1
14	NULL	Bergamo	1
15	Germany	Berlin	1
16	NULL	Berlin	1
17	Switzerland	Bern	1
18	NULL	Bern	1
19	USA	Boise	1

**Tarea 2:** escriba una instrucción **SELECT** que use la subcláusula **CUBE** para recuperar Grupos de agrupación basados en valores de ventas anuales, mensuales y diarios.

Paso 1. En el panel de consulta, escriba la siguiente consulta después de la descripción de la Tarea 2:

```
SELECT
YEAR(orderdate) AS orderyear,
MONTH(orderdate) AS ordermonth,
DAY(orderdate) AS orderday,
SUM(val) AS salesvalue
FROM Sales.OrderValues
GROUP BY
CUBE (YEAR(orderdate), MONTH(orderdate), DAY(orderdate));
```

Paso 2. Resalte la consulta escrita y haga clic en Ejecutar



The screenshot shows a SQL query window with the following query:

```
SELECT
YEAR(orderdate) AS orderyear,
MONTH(orderdate) AS ordermonth,
DAY(orderdate) AS orderday,
SUM(val) AS salesvalue
FROM Sales.OrderValues
GROUP BY
CUBE (YEAR(orderdate), MONTH(orderdate), DAY(orderdate));
```

Below the query window is a results grid with the following data:

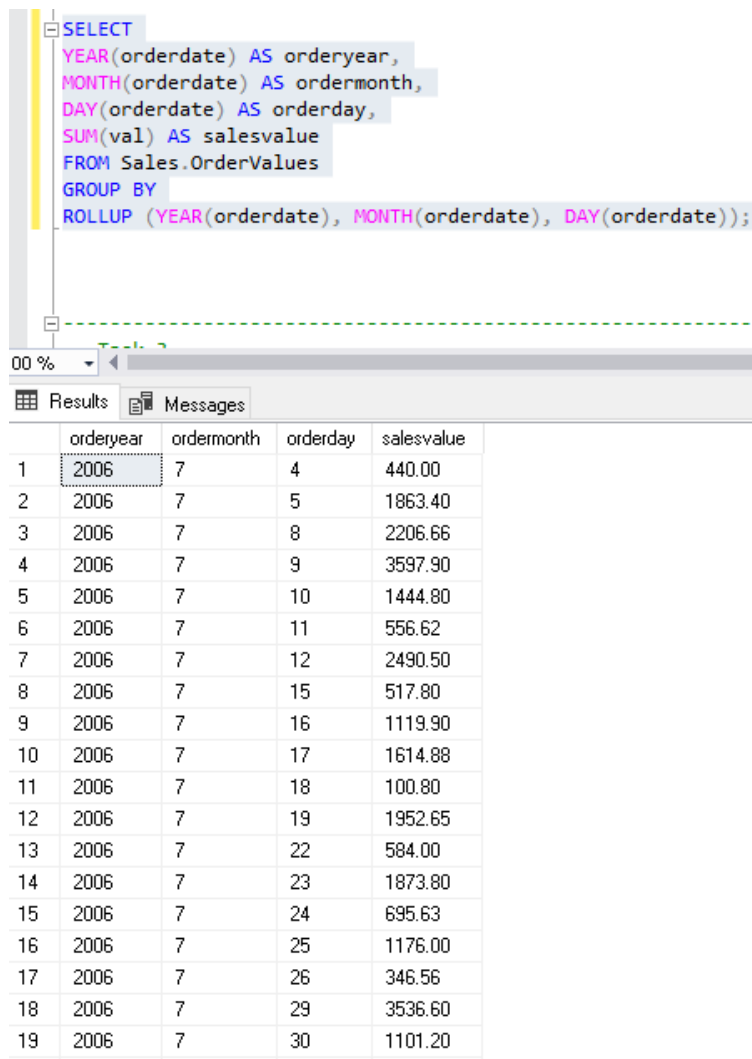
	orderyear	ordermonth	orderday	salesvalue
1	2007	1	1	6931.60
2	2008	1	1	1738.00
3	NULL	1	1	8669.60
4	2007	4	1	851.20
5	2008	4	1	11549.89
6	NULL	4	1	12401.09
7	2007	5	1	5636.96
8	2008	5	1	5448.57
9	NULL	5	1	11085.53
10	2007	7	1	142.50
11	NULL	7	1	142.50
12	2006	8	1	1424.00
13	2007	8	1	2697.50
14	NULL	8	1	4121.50
15	2007	9	1	716.64
16	NULL	9	1	716.64
17	2006	10	1	240.40
18	2007	10	1	3633.10
19	NULL	10	1	3873.50

### Tarea 3: escriba la misma instrucción SELECT utilizando la subcláusula ROLLUP

Paso 1. En el panel de consulta, escriba la siguiente consulta después de la descripción de la Tarea 3:

```
SELECT
YEAR(orderdate) AS orderyear,
MONTH(orderdate) AS ordermonth,
DAY(orderdate) AS orderday,
SUM(val) AS salesvalue
FROM Sales.OrderValues
GROUP BY
ROLLUP (YEAR(orderdate), MONTH(orderdate), DAY(orderdate));
```

Paso 2. Resalte la consulta escrita y haga clic en Ejecutar.



The screenshot shows a SQL query window with the following query highlighted:

```
SELECT
YEAR(orderdate) AS orderyear,
MONTH(orderdate) AS ordermonth,
DAY(orderdate) AS orderday,
SUM(val) AS salesvalue
FROM Sales.OrderValues
GROUP BY
ROLLUP (YEAR(orderdate), MONTH(orderdate), DAY(orderdate));
```

Below the query window, the 'Results' tab is active, displaying the following data:

	orderyear	ordermonth	orderday	salesvalue
1	2006	7	4	440.00
2	2006	7	5	1863.40
3	2006	7	8	2206.66
4	2006	7	9	3597.90
5	2006	7	10	1444.80
6	2006	7	11	556.62
7	2006	7	12	2490.50
8	2006	7	15	517.80
9	2006	7	16	1119.90
10	2006	7	17	1614.88
11	2006	7	18	100.80
12	2006	7	19	1952.65
13	2006	7	22	584.00
14	2006	7	23	1873.80
15	2006	7	24	695.63
16	2006	7	25	1176.00
17	2006	7	26	346.56
18	2006	7	29	3536.60
19	2006	7	30	1101.20

Paso 3. Observa el resultado. ¿Cuál es la diferencia entre las subcláusulas ROLLUP y CUBE de la cláusula GROUP BY?

Me gusta La subcláusula CUBE, la subcláusula ROLLUP proporciona una forma abreviada de definir conjuntos de agrupación múltiples. Sin embargo, a diferencia de CUBE, ROLLUP no produce todos los conjuntos de agrupación posibles que pueden definirse en función de los miembros de entrada; Produce un subconjunto de esos. ROLLUP asume una jerarquía entre los miembros de entrada y produce todos los conjuntos de agrupación. Eso tiene sentido, teniendo en cuenta la jerarquía. En otras palabras, mientras CUBE (a, b, c) produce los ocho grupos posibles de los tres miembros de entrada, ROLLUP (a, b, c) produce solo cuatro conjuntos de agrupación, asumiendo la jerarquía a¿b¿c. ROLLUP (a, b, c) es el equivalente a especificar CONJUNTOS DE AGRUPACIÓN ((a, b, c), (a, b), (a), ()).

¿Cuál es la subcláusula más apropiada para usar en este ejemplo?

Desde año, mes y día forman un Jerarquía, la cláusula ROLLUP es más adecuada. Probablemente no hay mucho interés en mostrar se acumula durante un mes independientemente del año, pero al revés es interesante.

#### Tarea 4: Analizar el valor total de ventas por año y mes

Paso 1. En el panel de consulta, escriba la siguiente consulta después de la descripción de la Tarea 4:

```
SELECT
  GROUPING_ID(YEAR(orderdate), MONTH(orderdate)) AS groupid,
  YEAR(orderdate) AS orderyear,
  MONTH(orderdate) AS ordermonth,
  SUM(val) AS salesvalue
FROM Sales.OrderValues
GROUP BY
  ROLLUP (YEAR(orderdate), MONTH(orderdate))
ORDER BY groupid, orderyear, ordermonth;
```

Paso 2. Resalte la consulta escrita y haga clic en Ejecutar.

<pre> SELECT GROUPING_ID(YEAR(orderdate), MONTH(orderdate)) as groupid, YEAR(orderdate) AS orderyear, MONTH(orderdate) AS ordermonth, SUM(val) AS salesvalue FROM Sales.OrderValues GROUP BY ROLLUP (YEAR(orderdate), MONTH(orderdate)) ORDER BY groupid, orderyear, ordermonth; </pre>				
100 %				
Results Messages				
	groupid	orderyear	ordermonth	salesvalue
1	0	2006	7	27861.90
2	0	2006	8	25485.28
3	0	2006	9	26381.40
4	0	2006	10	37515.73
5	0	2006	11	45600.05
6	0	2006	12	45239.63
7	0	2007	1	61258.08
8	0	2007	2	38483.64
9	0	2007	3	38547.23
10	0	2007	4	53032.95
11	0	2007	5	53781.30
12	0	2007	6	36362.82
13	0	2007	7	51020.86
14	0	2007	8	47287.68
15	0	2007	9	55629.27
16	0	2007	10	66749.23
17	0	2007	11	43533.80
18	0	2007	12	71398.44
19	0	2008	1	94222.12