

Advanced Database Pivot



Name

Dicha Zelianivan Arkana

NIM

2241720002

Class

2i

Department

Information Technology

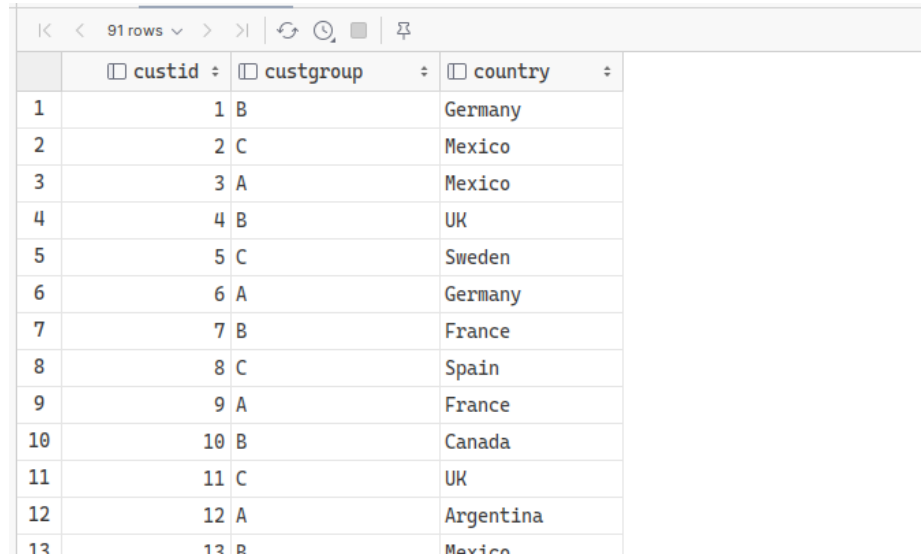
Study Program

D4 Informatics Engineering

1 Practicum

1. Dari view `Sales.CustGroups` yang sudah dibuat, buatlah sebuah query `SELECT` untuk menampilkan kolom `custid`, `custgroup`, dan `country`.

```
SELECT custid, custgroup, country
FROM Sales.CustGroups;
```



The screenshot shows a database interface with a query result table. The table has three columns: `custid`, `custgroup`, and `country`. There are 13 rows of data. The interface includes a toolbar with icons for navigation and a status bar indicating 91 rows.

	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

2. Modifikasilah kode T-SQL dari langkah no 3 di atas dengan menampilkan kolom `country`, lalu dengan menggunakan operator `PIVOT`, tambahkan 3 kolom tambahan yang berisi banyaknya customer dalam masing-masing grup (A, B, & C)

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

<div> <div> <div><</div> <div>></div> <div>21 rows</div> <div><</div> <div>></div> </div> <div> <div>↺</div> <div>⌚</div> <div>■</div> <div>📌</div> </div> </div>				
	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

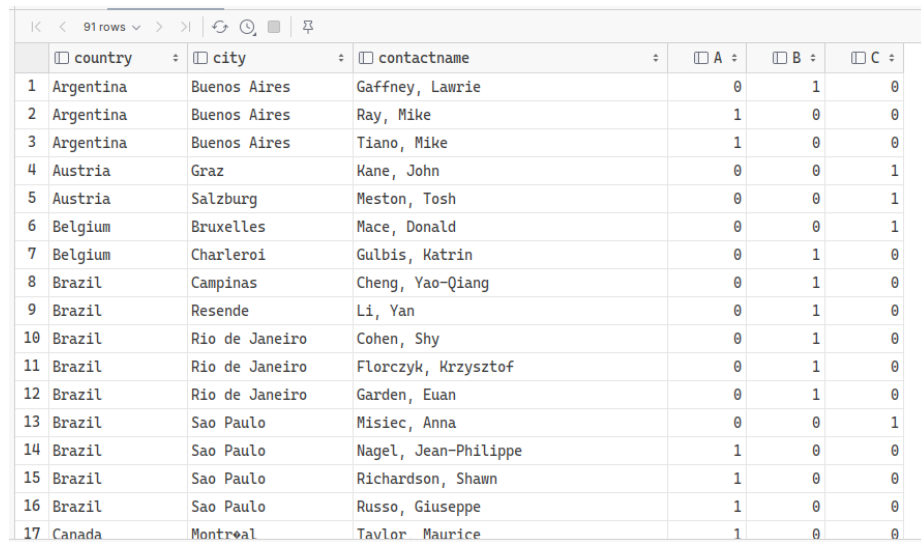
-
3. Salinlah statement SELECT dari Soal no 2 di atas, lalu jalankan kembali. Apakah hasil query ini sama dengan hasil pada Praktikum Bagian 1 no 4 di atas? Apakah jumlah baris yang dihasilkan sama persis?

21 rows				
	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

Hasil query ini sama dengan hasil pada Praktikum Bagian 1 no 4 di atas. Jumlah baris yang dihasilkan sama persis. Hal ini dikarenakan perintah alter digunakan untuk menambahkan kolom namun kita tidak melakukan SELECT terhadap kolom tersebut.

-
4. Modifikasi statement SELECT untuk menambahkan kolom city dan contactname!

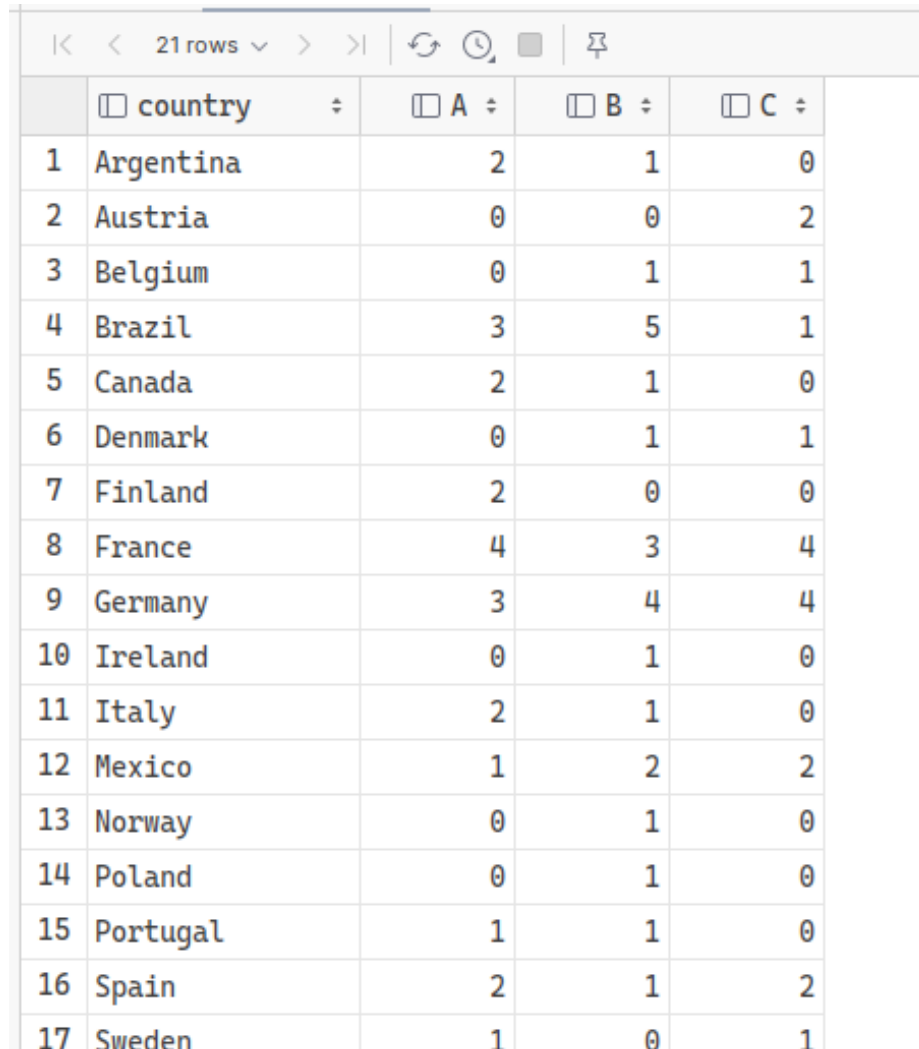
```
SELECT country, city, contactname, A, B, C
FROM (
    SELECT custid, custgroup, country, city, contactname
    FROM Sales.CustGroups
) AS SourceTable
PIVOT (
    COUNT(custid)
    FOR custgroup IN (A, B, C)
) AS PivotTable;
```



	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
11	Brazil	Rio de Janeiro	Florczyk, Krzysztof	0	1	0
12	Brazil	Rio de Janeiro	Garden, Euan	0	1	0
13	Brazil	Sao Paulo	Misiec, Anna	0	0	1
14	Brazil	Sao Paulo	Nagel, Jean-Philippe	1	0	0
15	Brazil	Sao Paulo	Richardson, Shawn	1	0	0
16	Brazil	Sao Paulo	Russo, Giuseppe	1	0	0
17	Canada	Montreal	Taylor, Maurice	1	0	0

-
5. Buatlah sebuah CTE bernama PivotCustGroups yang mendapatkan kolom custid, country, dan custgroup dari view Sales.CustGroups. Kemudian, buatlah sebuah query SELECT terhadap CTE tersebut dan gunakan operator PIVOT, seperti halnya dalam query SELECT pada Praktikum Bagian sebelumnya.

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



	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

-
6. Apakah hasilnya sama persis dengan hasil yang ada pada Praktikum Bagian 1? Mengapa demikian?

Hal ini dikarenakan query hanya mengubah subquery menjadi common table expression (CTE).

7. Apakah keuntungan penggunaan CTE ketika membuat query yang menggunakan operator PIVOT?

Query dapat menjadi lebih ringkas karena tidak perlu menuliskan subquery lagi sehingga query terlihat lebih rapi.

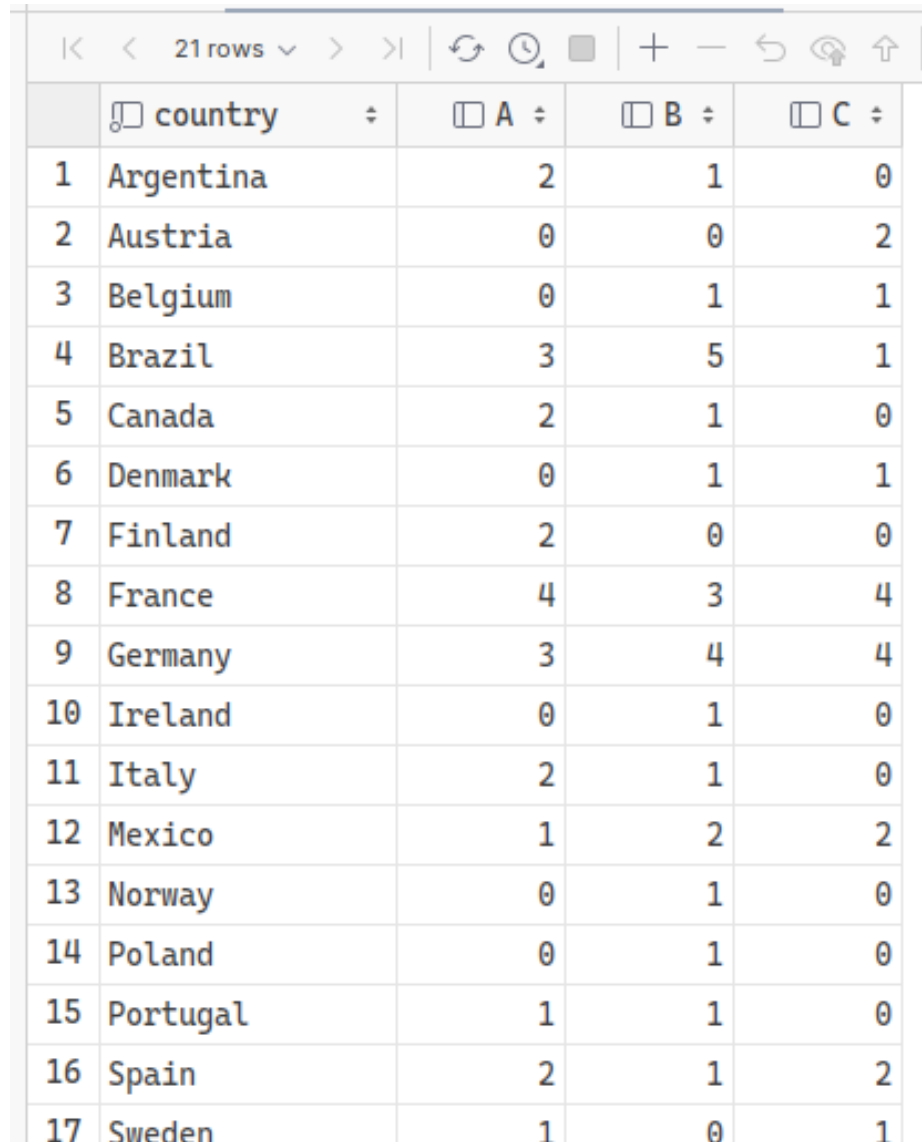
8. Buatlah sebuah query SELECT yang menampilkan data total jumlah penjualan untuk setiap kategori produk, untuk setiap customer. Tampilkan setiap kategori produk ke dalam kolom tersendiri, seperti pada tampilan di bawah ini.

```
SELECT
    custid,
    [Beverages],
    [Condiments],
    [Confections],
    [Dairy Products],
    [Grains/Cereals],
    [Meat/Poultry],
    [Produce],
    [Seafood]
FROM (
    SELECT * FROM SalesByCategory
) as SourceTable
PIVOT (
    SUM(SalesValue)
    FOR categoryname IN (
        [Beverages],
        [Condiments],
        [Confections],
        [Dairy Products],
        [Grains/Cereals],
        [Meat/Poultry],
        [Produce],
        [Seafood]
    )
) as PivotTable;
```

	custid	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood
1	1	<null>	426.0000	<null>	1255.0000	<null>	<null>	91.2000	530.0000
2	2	<null>	<null>	64.4000	390.0000	<null>	<null>	<null>	60.0000
3	3	380.0000	<null>	<null>	<null>	280.0000	<null>	<null>	<null>
4	4	282.0000	<null>	4440.0000	812.5000	<null>	<null>	<null>	304.0000
5	5	850.5000	300.0000	2202.5500	<null>	<null>	1237.9000	1360.0000	2151.6000
6	6	<null>	114.0000	283.0000	714.0000	<null>	<null>	424.0000	625.0000
7	7	<null>	<null>	<null>	437.5000	292.5000	<null>	<null>	<null>
8	8	<null>	<null>	<null>	<null>	280.0000	<null>	<null>	<null>
9	9	533.0000	1750.0000	1515.1000	556.8000	665.0000	624.0000	705.0000	837.0000
10	10	1706.5000	1290.1000	4518.3000	992.5000	684.0000	234.0000	1872.0000	930.0000
11	11	1380.0000	<null>	<null>	220.0000	441.0000	<null>	120.0000	270.0000
12	12	1037.0000	<null>	<null>	25.0000	<null>	<null>	364.0000	150.0000
13	14	570.0000	1843.8000	591.6000	<null>	1216.0000	<null>	<null>	200.0000
14	15	<null>	405.7500	<null>	<null>	<null>	<null>	<null>	100.0000
15	16	<null>	640.5000	<null>	<null>	<null>	291.0000	<null>	<null>
16	17	<null>	<null>	374.7600	1650.0000	<null>	656.0000	<null>	120.8500
17	18	140.0000	<null>	<null>	<null>	<null>	585.0000	<null>	135.1000

-
9. Buatlah query SELECT yang menghasilkan kolom country, kolom A, kolom B, dan kolom C dari view Sales.PivotCustGroups yang telah dibuat.

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



	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

-
10. Buatlah sebuah query SELECT terhadap view Sales.PivotCustGroups yang menghasilkan data seperti tampilan berikut:

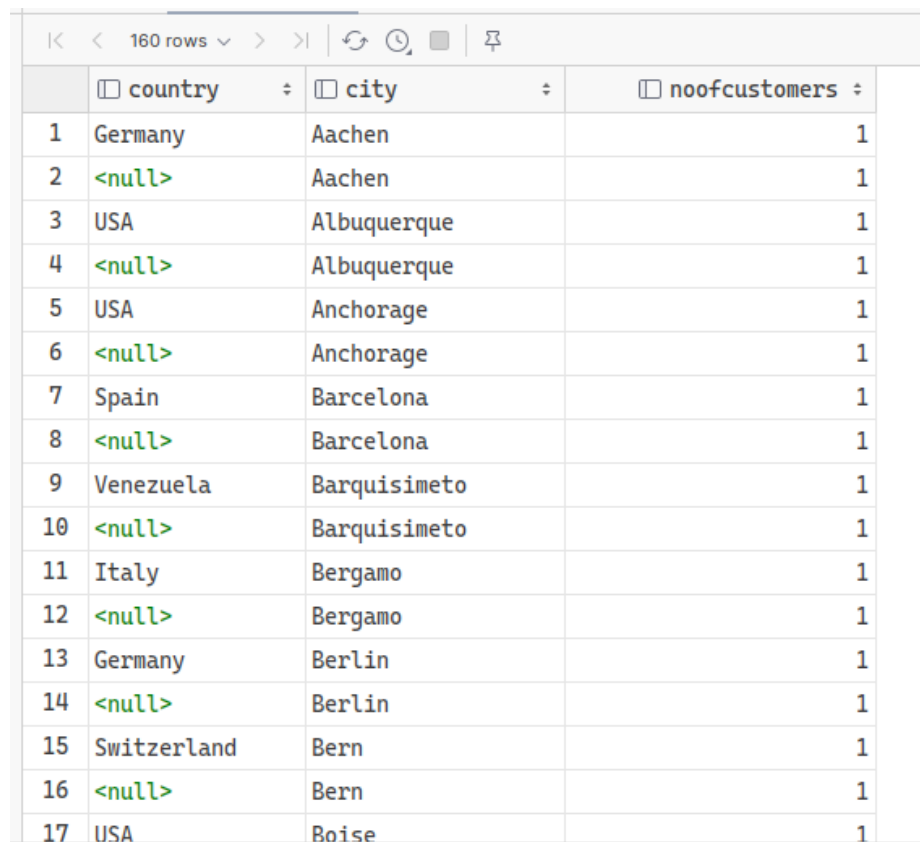
```
SELECT custgroup, country, SUM(numberofcustomers) AS numberofcustomers
FROM Sales.PivotCustGroups
UNPIVOT (
    numberofcustomers
    FOR custgroup IN (
        A, B, C
    )
) AS unpivoted
GROUP BY custgroup, country
ORDER BY country;
```

	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

11. Buatlah query SELECT terhadap tabel Sales.Customers yang terdiri dari kolom contry, city, dan kolom kalkulasi yang menghitung banyaknya customer bernama noofcustomers. Dapatkan pengelompokan (grouping set) berdasarkan:

- kolom country dan city
- kolom country
- kolom city
- dan sebuah kolom tanpa kelompok

```
SELECT country, city, COUNT(*) 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	Spain	Barcelona	1
8	<null>	Barcelona	1
9	Venezuela	Barquisimeto	1
10	<null>	Barquisimeto	1
11	Italy	Bergamo	1
12	<null>	Bergamo	1
13	Germany	Berlin	1
14	<null>	Berlin	1
15	Switzerland	Bern	1
16	<null>	Bern	1
17	USA	Boise	1

12. Buatlah sebuah query SELECT terhadap view Sales.OrderValues yang berisi kolom:

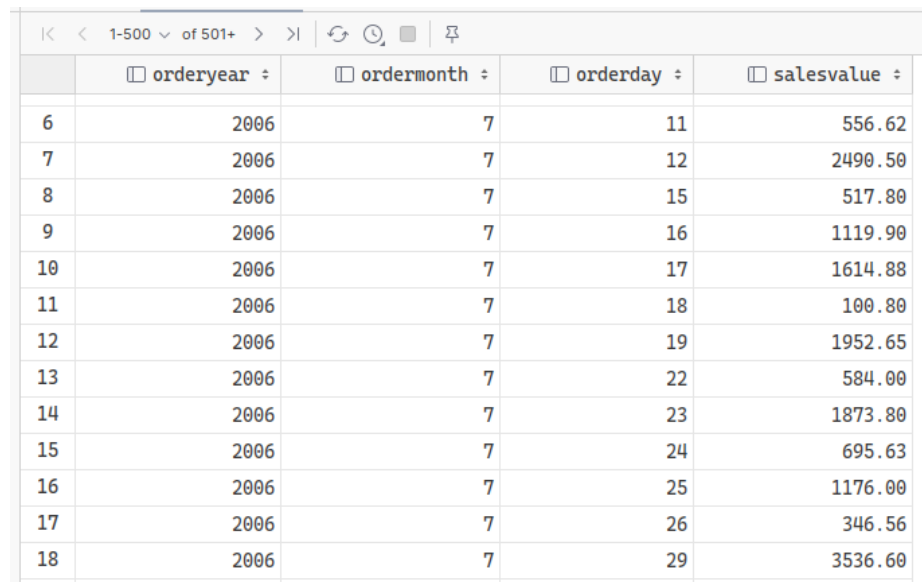
- orderyear: tahun dari kolom orderdate
- ordermonth: bulan dari kolom orderdate
- orderday: hari dari kolom orderdate
- salesvalue: total jumlah penjualan dari kolom val

```
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)
);
```

	orderyear	ordermonth	orderday	salesvalue
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

-
13. Salinlah query dari Soal no 12 di atas dan ubah sub klausa CUBE menjadi ROLLUP, lalu jalankan query tersebut.

```
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)
);
```



	orderyear	ordermonth	orderday	salesvalue
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

-
14. Apakah perbedaan antara sub klausa ROLLUP dan CUBE? Manakah yang lebih tepat digunakan untuk query pada langkah 1 di atas?

Perbedaan antara sub klausa ROLLUP dan CUBE adalah ROLLUP hanya menghasilkan subtotal untuk kolom yang ditentukan sedangkan CUBE menghasilkan subtotal untuk semua kolom.

15. Buatlah query SELECT terhadap view Sales.OrderValues dan dapatkan kolom berikut ini:

- kolom kalkulasi dengan nama alias groupid (gunakan fungsi GROUPING_ID dengan tahun order dan bulan order sebagai parameter inputnya)
- orderyear: tahun dari kolom orderdate
- ordermonth: bulan dari kolom orderdate
- salesvalue: total nilai penjualan dari kolom val
- oleh karena tahun dan bulan berbentuk hierarki, dapatkan semua pengelompokan/ grouping set berdasarkan kolom orderyear dan ordermonth, lalu urutkan berdasarkan groupid, orderyear, dan ordermonth

```
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;
```

27 rows				
	groupid	orderyear	ordermonth	salesvalue
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