

Advanced Database Window, Ranking, Offset, and Aggregate



Name

Dicha Zelianivan Arkana

NIM

2241720002

Class

2i

Department

Information Technology

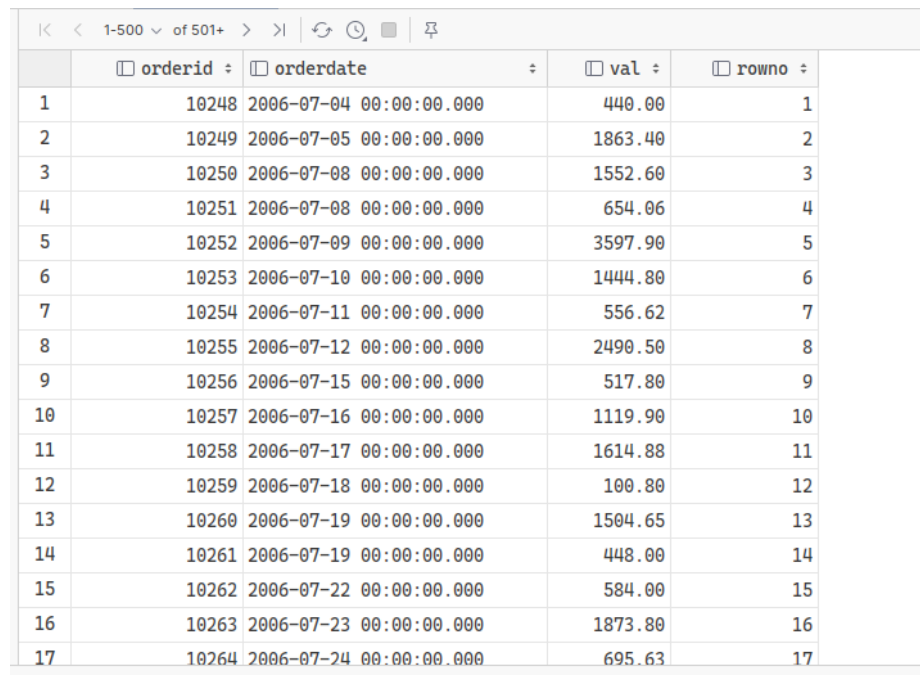
Study Program

D4 Informatics Engineering

1 Practicum

1. Tulislah pernyataan `SELECT` untuk mengambil kolom `orderid`, `orderdate`, dan `val` serta kolom hasil perhitungan bernama `rowno` dari view `Sales.OrderValues`! Gunakan fungsi `ROW_NUMBER` untuk mengembalikan `rowno`, urutkan nomor baris berdasarkan kolom `orderdate`!

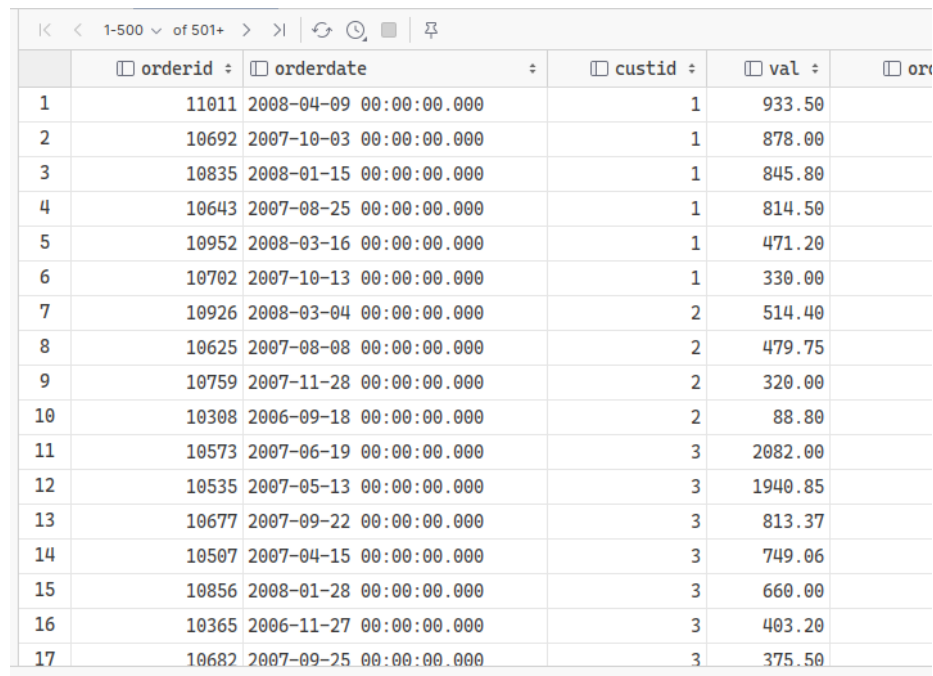
```
SELECT
    orderid,
    orderdate,
    val,
    rowno = ROW_NUMBER() OVER (ORDER BY orderdate)
FROM
    Sales.OrderValues;
```



	orderid	orderdate	val	rowno
1	10248	2006-07-04 00:00:00.000	440.00	1
2	10249	2006-07-05 00:00:00.000	1863.40	2
3	10250	2006-07-08 00:00:00.000	1552.60	3
4	10251	2006-07-08 00:00:00.000	654.06	4
5	10252	2006-07-09 00:00:00.000	3597.90	5
6	10253	2006-07-10 00:00:00.000	1444.80	6
7	10254	2006-07-11 00:00:00.000	556.62	7
8	10255	2006-07-12 00:00:00.000	2490.50	8
9	10256	2006-07-15 00:00:00.000	517.80	9
10	10257	2006-07-16 00:00:00.000	1119.90	10
11	10258	2006-07-17 00:00:00.000	1614.88	11
12	10259	2006-07-18 00:00:00.000	100.80	12
13	10260	2006-07-19 00:00:00.000	1504.65	13
14	10261	2006-07-19 00:00:00.000	448.00	14
15	10262	2006-07-22 00:00:00.000	584.00	15
16	10263	2006-07-23 00:00:00.000	1873.80	16
17	10264	2006-07-24 00:00:00.000	695.63	17

-
2. Salin T-SQL pada soal no 1. Kemudian modifikasi dengan memasukkan kolom tambahan bernama rankno. Untuk membuat rankno gunakan fungsi RANK dengan urutan peringkat berdasarkan kolom orderdate!

```
SELECT
   orderid,
    orderdate,
    val,
    rowno = ROW_NUMBER() OVER (ORDER BY orderdate),
    rankno = RANK() OVER (ORDER BY orderdate)
FROM
    Sales.OrderValues;
```

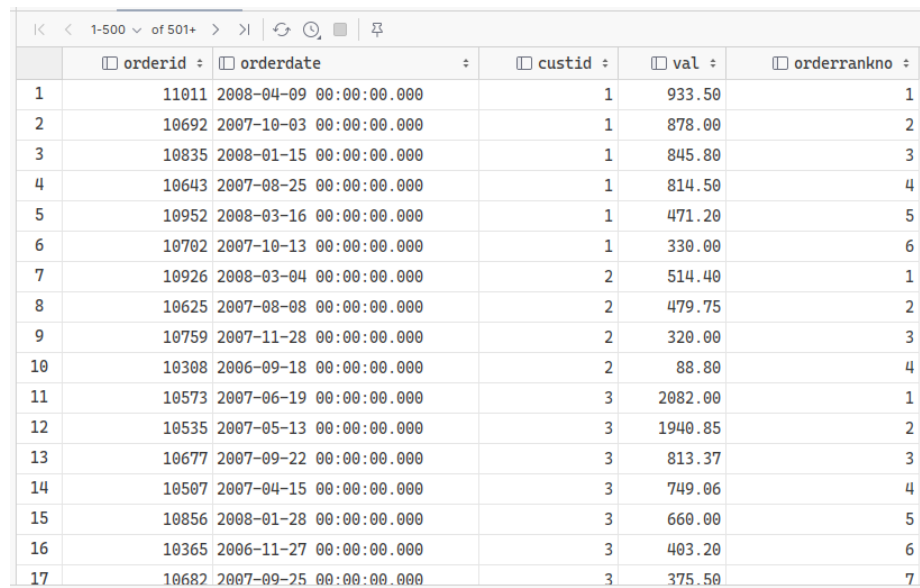


	orderid	orderdate	custid	val	rowno
1	11011	2008-04-09 00:00:00.000	1	933.50	
2	10692	2007-10-03 00:00:00.000	1	878.00	
3	10835	2008-01-15 00:00:00.000	1	845.80	
4	10643	2007-08-25 00:00:00.000	1	814.50	
5	10952	2008-03-16 00:00:00.000	1	471.20	
6	10702	2007-10-13 00:00:00.000	1	330.00	
7	10926	2008-03-04 00:00:00.000	2	514.40	
8	10625	2007-08-08 00:00:00.000	2	479.75	
9	10759	2007-11-28 00:00:00.000	2	320.00	
10	10308	2006-09-18 00:00:00.000	2	88.80	
11	10573	2007-06-19 00:00:00.000	3	2082.00	
12	10535	2007-05-13 00:00:00.000	3	1940.85	
13	10677	2007-09-22 00:00:00.000	3	813.37	
14	10507	2007-04-15 00:00:00.000	3	749.06	
15	10856	2008-01-28 00:00:00.000	3	660.00	
16	10365	2006-11-27 00:00:00.000	3	403.20	
17	10682	2007-09-25 00:00:00.000	3	375.50	

3. Apakah perbedaan antara fungsi RANK dan fungsi ROW_NUMBER?
- ROW_NUMBER akan mengembalikan nomor baris yang unik untuk setiap baris dalam partisi.
 - RANK akan mengembalikan peringkat baris dalam partisi dengan nilai yang sama. Nilai peringkat tidak akan berurutan jika ada nilai yang sama.

-
4. Tuliskan pernyataan **SELECT** untuk mengambil kolom **orderid**, **orderdate**, **custid**, dan **val** serta hitung kolom bernama **orderrankno** dari view **Sales.OrderValues**. Kolom **orderrankno** harus menampilkan ranking per pelanggan secara independen, berdasarkan pemesanan **val** dalam urutan menurun!

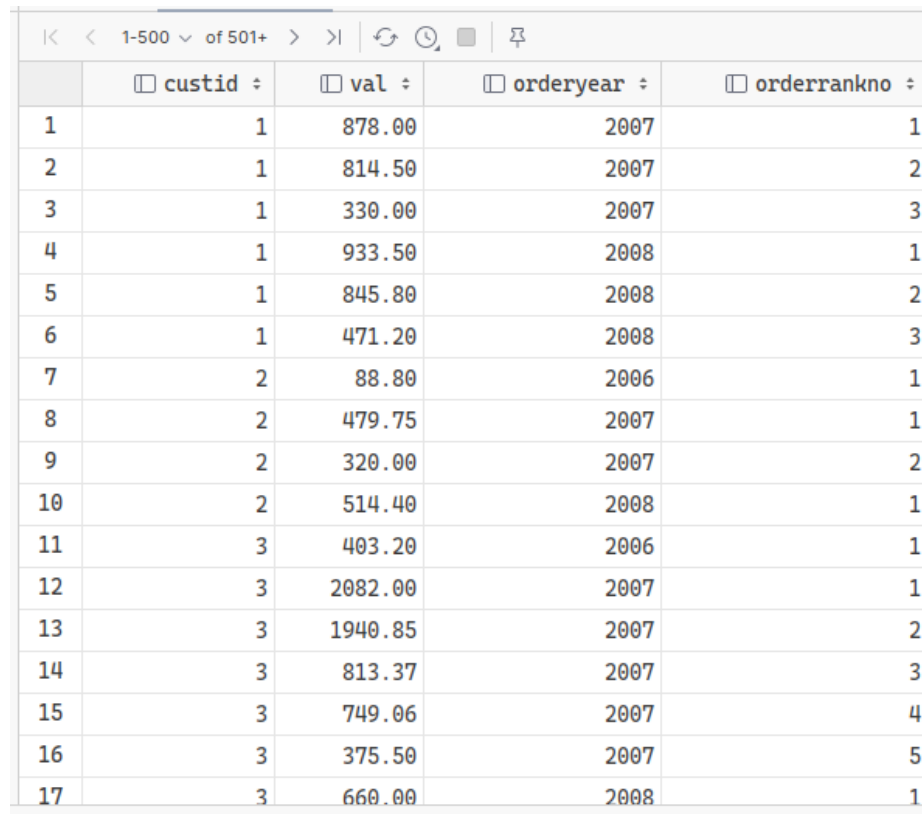
```
SELECT
    orderid,
    orderdate,
    custid,
    val,
    orderrankno = RANK() OVER (PARTITION BY custid ORDER BY val DESC)
FROM
    Sales.OrderValues;
```



	orderid	orderdate	custid	val	orderrankno
1	11011	2008-04-09 00:00:00.000	1	933.50	1
2	10692	2007-10-03 00:00:00.000	1	878.00	2
3	10835	2008-01-15 00:00:00.000	1	845.80	3
4	10643	2007-08-25 00:00:00.000	1	814.50	4
5	10952	2008-03-16 00:00:00.000	1	471.20	5
6	10702	2007-10-13 00:00:00.000	1	330.00	6
7	10926	2008-03-04 00:00:00.000	2	514.40	1
8	10625	2007-08-08 00:00:00.000	2	479.75	2
9	10759	2007-11-28 00:00:00.000	2	320.00	3
10	10308	2006-09-18 00:00:00.000	2	88.80	4
11	10573	2007-06-19 00:00:00.000	3	2082.00	1
12	10535	2007-05-13 00:00:00.000	3	1940.85	2
13	10677	2007-09-22 00:00:00.000	3	813.37	3
14	10507	2007-04-15 00:00:00.000	3	749.06	4
15	10856	2008-01-28 00:00:00.000	3	660.00	5
16	10365	2006-11-27 00:00:00.000	3	403.20	6
17	10682	2007-09-25 00:00:00.000	3	375.50	7

-
5. Tuliskan pernyataan **SELECT** untuk mengambil kolom **custid** dan **val** dari view **Sales.OrderValues**. Tambahkan dua kolom berikut: 1) **orderyear** sebagai tahun dari kolom **orderdate** 2) **orderrankno** sebagai nomor urut, dipartisi berdasarkan pelanggan dan tahun pesanan, dan diurutkan berdasarkan nilai pesanan dalam urutan menurun!

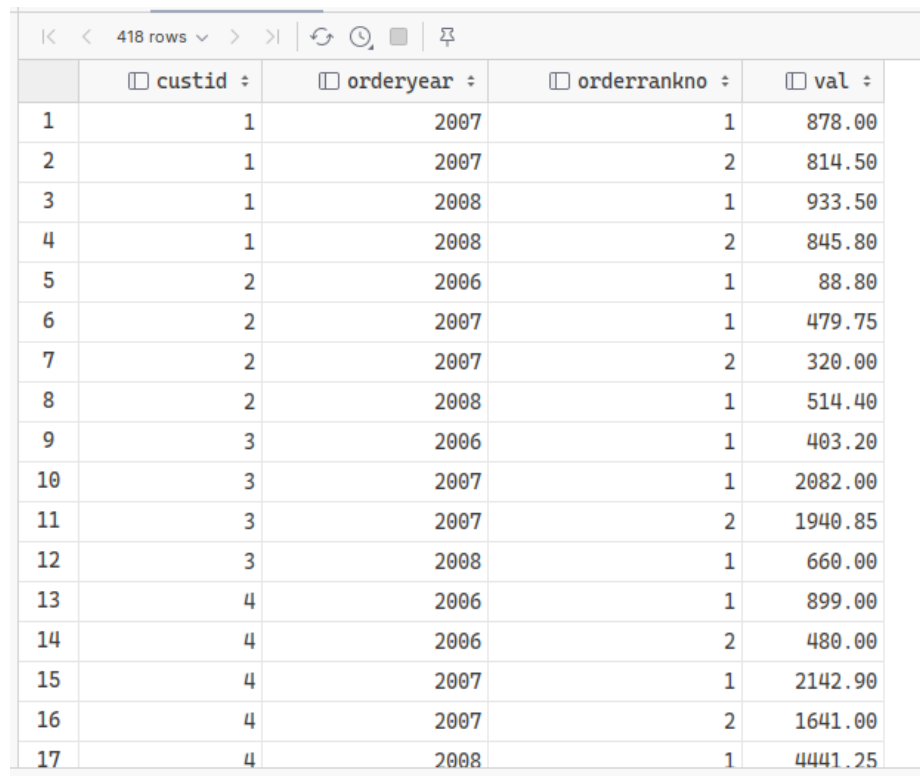
```
SELECT
    custid,
    val,
    orderyear = YEAR(orderdate),
    orderrankno = RANK() OVER (PARTITION BY custid,
                                YEAR(orderdate) ORDER BY val DESC)
FROM
    Sales.OrderValues;
```



	custid ÷	val ÷	orderyear ÷	orderrankno ÷
1	1	878.00	2007	1
2	1	814.50	2007	2
3	1	330.00	2007	3
4	1	933.50	2008	1
5	1	845.80	2008	2
6	1	471.20	2008	3
7	2	88.80	2006	1
8	2	479.75	2007	1
9	2	320.00	2007	2
10	2	514.40	2008	1
11	3	403.20	2006	1
12	3	2082.00	2007	1
13	3	1940.85	2007	2
14	3	813.37	2007	3
15	3	749.06	2007	4
16	3	375.50	2007	5
17	3	660.00	2008	1

-
6. Salin query jawaban soal nomor 6 dan modifikasi untuk memfilter hanya pesanan dengan dua peringkat paling awal berdasarkan kolom orderrankno!

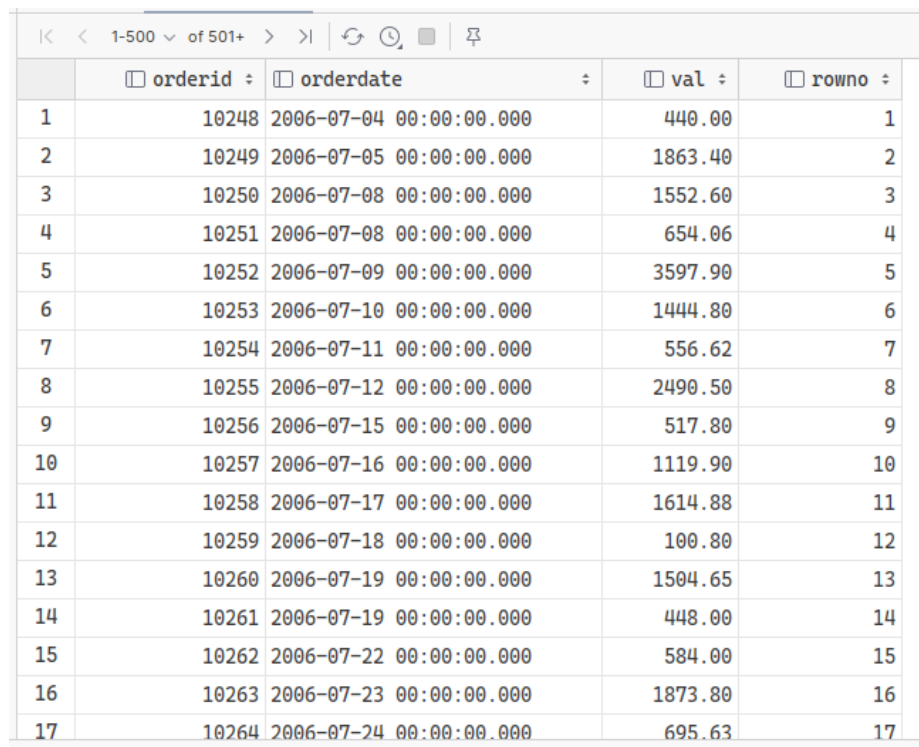
```
WITH RankedOrderValues AS (  
    SELECT custid,  
           val,  
           orderyear = YEAR(orderdate),  
           orderrankno = RANK() OVER (PARTITION BY custid,  
                                       YEAR(orderdate) ORDER BY val DESC)  
    FROM Sales.OrderValues  
)  
SELECT custid, orderyear, orderrankno, val  
FROM RankedOrderValues  
WHERE orderrankno <= 2
```



	custid ÷	orderyear ÷	orderrankno ÷	val ÷
1	1	2007	1	878.00
2	1	2007	2	814.50
3	1	2008	1	933.50
4	1	2008	2	845.80
5	2	2006	1	88.80
6	2	2007	1	479.75
7	2	2007	2	320.00
8	2	2008	1	514.40
9	3	2006	1	403.20
10	3	2007	1	2082.00
11	3	2007	2	1940.85
12	3	2008	1	660.00
13	4	2006	1	899.00
14	4	2006	2	480.00
15	4	2007	1	2142.90
16	4	2007	2	1641.00
17	4	2008	1	4441.25

-
7. Buatlah (common table expression) CTE dengan nama OrderRows berdasarkan query yang mengambil kolom `orderid`, `orderdate`, and `val` dari view `Sales.OrderValues`. Tambahkan kolom hasil perhitungan dengan nama `rowno` menggunakan fungsi `ROW_NUMBER` yang diurutkan berdasarkan kolom `orderdate` dan `orderid`!

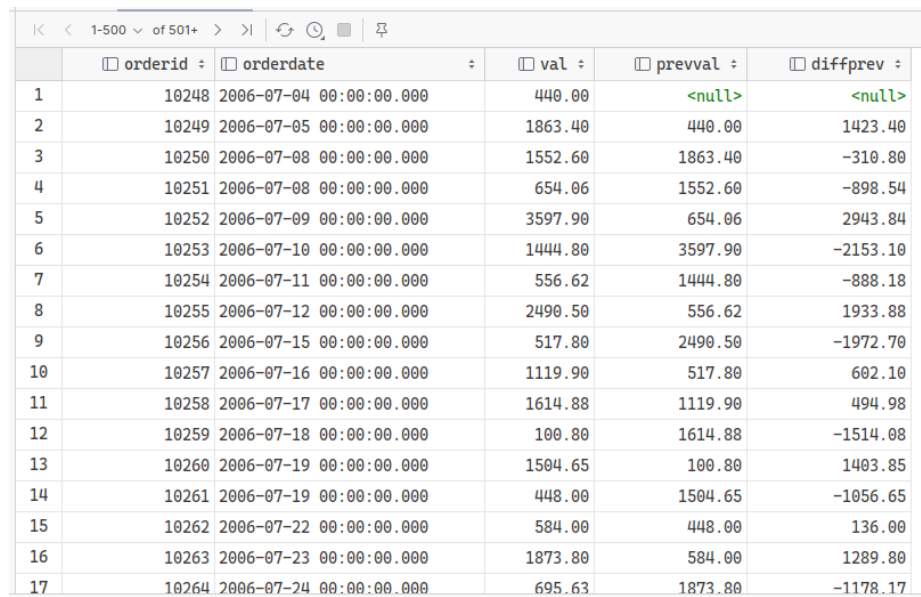
```
WITH OrderRows AS (  
    SELECT  
        orderid,  
        orderdate,  
        val,  
        rowno = ROW_NUMBER() OVER (ORDER BY orderdate, orderid)  
    FROM  
        Sales.OrderValues  
)  
SELECT * FROM OrderRows;
```



	orderid	orderdate	val	rowno
1	10248	2006-07-04 00:00:00.000	440.00	1
2	10249	2006-07-05 00:00:00.000	1863.40	2
3	10250	2006-07-08 00:00:00.000	1552.60	3
4	10251	2006-07-08 00:00:00.000	654.06	4
5	10252	2006-07-09 00:00:00.000	3597.90	5
6	10253	2006-07-10 00:00:00.000	1444.80	6
7	10254	2006-07-11 00:00:00.000	556.62	7
8	10255	2006-07-12 00:00:00.000	2490.50	8
9	10256	2006-07-15 00:00:00.000	517.80	9
10	10257	2006-07-16 00:00:00.000	1119.90	10
11	10258	2006-07-17 00:00:00.000	1614.88	11
12	10259	2006-07-18 00:00:00.000	100.80	12
13	10260	2006-07-19 00:00:00.000	1504.65	13
14	10261	2006-07-19 00:00:00.000	448.00	14
15	10262	2006-07-22 00:00:00.000	584.00	15
16	10263	2006-07-23 00:00:00.000	1873.80	16
17	10264	2006-07-24 00:00:00.000	695.63	17

-
8. Tuliskan pernyataan SELECT terhadap CTE dan gunakan LEFT JOIN dengan CTE yang sama untuk mengambil baris saat ini (current row) dan baris sebelumnya (previous row) berdasarkan kolom rowno. Kembalikan kolom `orderid`, `orderdate`, and `val` untuk baris saat ini dan kolom `val` untuk baris sebelumnya sebagai `prevval`. Tambahkan kolom hasil perhitungan dengan nama `diffprev` yang menunjukkan perbedaan antara `val` saat ini dengan sebelumnya!

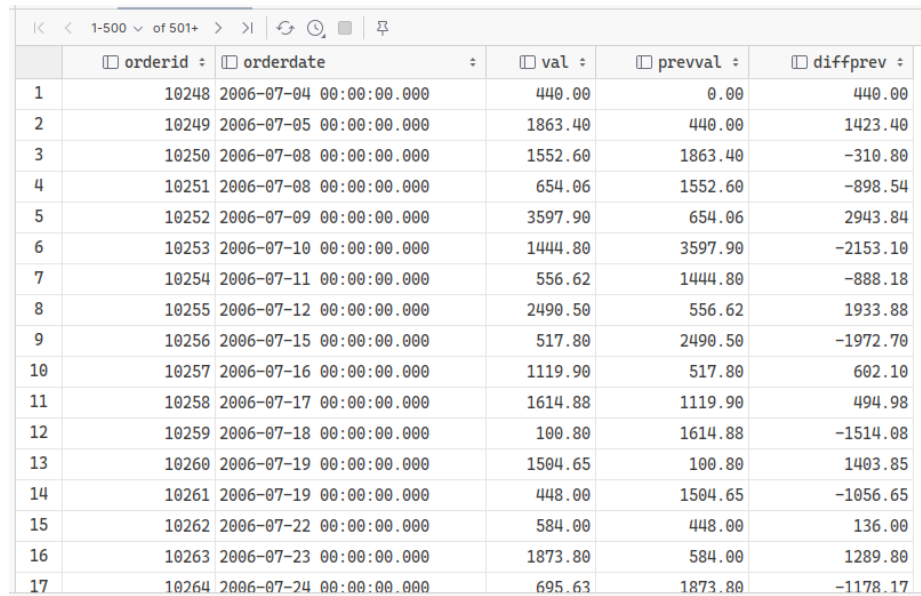
```
WITH OrderRows AS (  
    SELECT  
        orderid,  
        orderdate,  
        val,  
        rowno = ROW_NUMBER() OVER (ORDER BY orderdate, orderid)  
    FROM  
        Sales.OrderValues  
)  
SELECT  
    curr.orderid,  
    curr.orderdate,  
    curr.val,  
    prev.val AS prevval,  
    diffprev = curr.val - prev.val  
FROM  
    OrderRows curr  
LEFT JOIN OrderRows prev ON curr.rowno = prev.rowno + 1;
```



	orderid	orderdate	val	prevval	diffprev
1	10248	2006-07-04 00:00:00.000	440.00	<null>	<null>
2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40
3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80
4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54
5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84
6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10
7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18
8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88
9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70
10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10
11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98
12	10259	2006-07-18 00:00:00.000	100.80	1614.88	-1514.08
13	10260	2006-07-19 00:00:00.000	1504.65	100.80	1403.85
14	10261	2006-07-19 00:00:00.000	448.00	1504.65	-1056.65
15	10262	2006-07-22 00:00:00.000	584.00	448.00	136.00
16	10263	2006-07-23 00:00:00.000	1873.80	584.00	1289.80
17	10264	2006-07-24 00:00:00.000	695.63	1873.80	-1178.17

-
9. Tuliskan pernyataan **SELECT** menggunakan fungsi **LAG** untuk mendapat hasil yang sama dengan query pada soal no.2! Query tidak yang dibuat pada soal ini tidak menggunakan **CTE**.

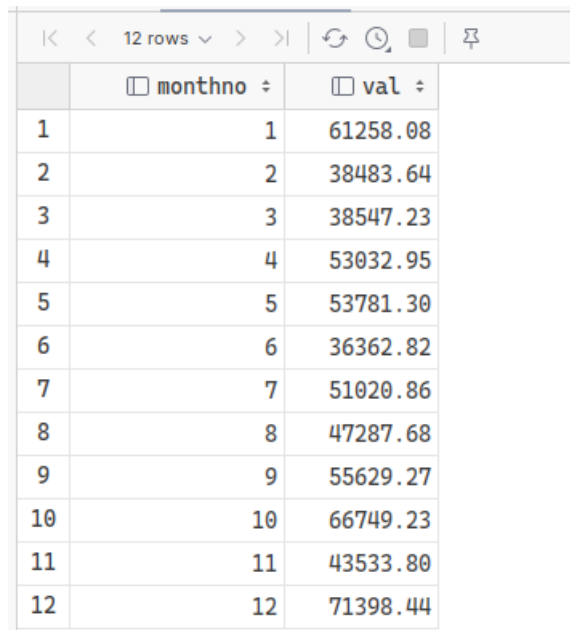
```
SELECT
   orderid,
    orderdate,
    val,
    prevval = LAG(val, 1, 0) OVER (ORDER BY orderdate),
    diffprev = val - LAG(val, 1, 0) OVER (ORDER BY orderdate)
FROM
    Sales.OrderValues;
```



	orderid	orderdate	val	prevval	diffprev
1	10248	2006-07-04 00:00:00.000	440.00	0.00	440.00
2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40
3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80
4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54
5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84
6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10
7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18
8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88
9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70
10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10
11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98
12	10259	2006-07-18 00:00:00.000	100.80	1614.88	-1514.08
13	10260	2006-07-19 00:00:00.000	1504.65	100.80	1403.85
14	10261	2006-07-19 00:00:00.000	448.00	1504.65	-1056.65
15	10262	2006-07-22 00:00:00.000	584.00	448.00	136.00
16	10263	2006-07-23 00:00:00.000	1873.80	584.00	1289.80
17	10264	2006-07-24 00:00:00.000	695.63	1873.80	-1178.17

-
10. Buatlah sebuah CTE bernama `SalesMonth2007` yang membuat dua kolom yaitu, `monthno` (jumlah bulan dari kolom `orderdate`) dan `val` (agregat dari kolom `val`)! Kemudian filter hasilnya hanya untuk tahun pesanan 2007 dan dikelompokkan berdasarkan `monthno`!

```
WITH SalesMonth2007 AS (  
    SELECT  
        monthno = MONTH(orderdate),  
        val = SUM(val)  
    FROM  
        Sales.OrderValues  
    WHERE  
        YEAR(orderdate) = 2007  
    GROUP BY  
        MONTH(orderdate)  
)  
SELECT * FROM SalesMonth2007;
```



	monthno	val
1	1	61258.08
2	2	38483.64
3	3	38547.23
4	4	53032.95
5	5	53781.30
6	6	36362.82
7	7	51020.86
8	8	47287.68
9	9	55629.27
10	10	66749.23
11	11	43533.80
12	12	71398.44

11. Tuliskan pernyataan **SELECT** yang akan mengambil kolom **monthno** dan **val** dari CTE dan tambahkan 3 kolom untuk ditampilkan, yaitu :

- (a) **avglast3months** (jumlah penjualan rata-rata tiga bulan terakhir)
- (b) **diffjanuary** (perbedaan antara val saat ini dengan val pada bulan januari, gunakan fungsi **FIRST_VALUE**)
- (c) **nextval** (nilai dari kolom val pada bulan selanjutnya)

Informasi : Jumlah rata-rata untuk tiga bulan terakhir tidak dihitung dengan benar karena jumlah total 2 bulan pertama dibagi dengan 3.

```
WITH SalesMonth2007 AS (
    SELECT
        monthno = MONTH(orderdate),
        val = SUM(val)
    FROM
        Sales.OrderValues
    WHERE
        YEAR(orderdate) = 2007
    GROUP BY
        MONTH(orderdate)
)
SELECT
    monthno,
    val,
    avglast3months = ISNULL(AVG(val) OVER (ORDER BY monthno ROWS
        BETWEEN 3 PRECEDING AND 1 PRECEDING), 0),
    diffjanuary = val - FIRST_VALUE(val) OVER (ORDER BY monthno),
    nextval = LEAD(val, 1, 0) OVER (ORDER BY monthno)
FROM
    SalesMonth2007;
```

	monthno	val	avglast3months	diffjanuary	nextval
1	1	61258.08	0.000000	0.00	38483.64
2	2	38483.64	61258.080000	-22774.44	38547.23
3	3	38547.23	49870.860000	-22710.85	53032.95
4	4	53032.95	46096.316666	-8225.13	53781.30
5	5	53781.30	43354.606666	-7476.78	36362.82
6	6	36362.82	48453.826666	-24895.26	51020.86
7	7	51020.86	47725.690000	-10237.22	47287.68
8	8	47287.68	47054.993333	-13970.40	55629.27
9	9	55629.27	44890.453333	-5628.81	66749.23
10	10	66749.23	51312.603333	5491.15	43533.80
11	11	43533.80	56555.393333	-17724.28	71398.44
12	12	71398.44	55304.100000	10140.36	0.00

12. Tuliskan pernyataan `SELECT` untuk mengambil kolom `custid`, `orderid`, `orderdate`, dan `val` dari view `Sales.OrderValues`. Tambahkan kolom bernama `percoftotalcust` yang berisi persentase nilai masing-masing jumlah pesanan penjualan dibandingkan dengan jumlah penjualan untuk pelanggan tersebut!

```
SELECT
    custid,
    orderid,
    orderdate,
    val,
    percoftotalcust = val / SUM(val) OVER (PARTITION BY custid)
FROM
    Sales.OrderValues;
```

	custid	orderid	orderdate	val	percoftotalcust
1	1	10643	2007-08-25 00:00:00.000	814.50	0.19061549262813011935408378
2	1	10692	2007-10-03 00:00:00.000	878.00	0.20547624619705125204774163
3	1	10702	2007-10-13 00:00:00.000	330.00	0.07722911303533816990404867
4	1	10835	2008-01-15 00:00:00.000	845.80	0.19794055698572431546922536
5	1	10952	2008-03-16 00:00:00.000	471.20	0.11027381230985256260238708
6	1	11011	2008-04-09 00:00:00.000	933.50	0.21846477884390358062251345
7	2	10926	2008-03-04 00:00:00.000	514.40	0.36665597491001104814854413
8	2	10759	2007-11-28 00:00:00.000	320.00	0.22809080865319505328058733
9	2	10625	2007-08-08 00:00:00.000	479.75	0.34195801703553227128550554
10	2	10308	2006-09-18 00:00:00.000	88.80	0.06329519940126162728536298
11	3	10365	2006-11-27 00:00:00.000	403.20	0.05740335251524064704056674
12	3	10507	2007-04-15 00:00:00.000	749.06	0.10664324215046170404813225
13	3	10535	2007-05-13 00:00:00.000	1940.85	0.27631770022124208781915666
14	3	10677	2007-09-22 00:00:00.000	813.37	0.11579901992887223482982582
15	3	10573	2007-06-19 00:00:00.000	2082.00	0.29641314468435274587911696
16	3	10682	2007-09-25 00:00:00.000	375.50	0.05345971941833547362036907
17	3	10856	2008-01-28 00:00:00.000	660.00	0.09396382108149510676283246

13. Salin pernyataan **SELECT** sebelumnya dan modifikasi dengan menambahkan kolom yang baru dihitung bernama **runval**! Kolom ini harus berisi total penjualan yang sedang terjadi untuk setiap pelanggan berdasarkan tanggal pemesanan, menggunakan **orderid** sebagai tiebreaker.

```
SELECT
    custid,
    orderid,
    orderdate,
    val,
    percoftotalcust = val / SUM(val) OVER (PARTITION BY custid),
    runval = SUM(val) OVER (PARTITION BY custid ORDER BY orderdate, orderid)
FROM
    Sales.OrderValues;
```

	custid	orderid	orderdate	val	percoftotalcust	runval
1	1	10643	2007-08-25 00:00:00.000	814.50	0.19061549262813011935408378	814.50
2	1	10692	2007-10-03 00:00:00.000	878.00	0.20547624619705125204774163	1692.50
3	1	10702	2007-10-13 00:00:00.000	330.00	0.07722911303533816990404867	2022.50
4	1	10835	2008-01-15 00:00:00.000	845.80	0.19794055698572431546922536	2868.30
5	1	10952	2008-03-16 00:00:00.000	471.20	0.11027381230985256260238708	3339.50
6	1	11011	2008-04-09 00:00:00.000	933.50	0.21846477884390358062251345	4273.00
7	2	10308	2006-09-18 00:00:00.000	88.80	0.06329519940126162728536298	88.80
8	2	10625	2007-08-08 00:00:00.000	479.75	0.34195001703553227128550554	568.55
9	2	10759	2007-11-28 00:00:00.000	320.00	0.22809080865319505328058733	888.55
10	2	10926	2008-03-04 00:00:00.000	514.40	0.36665597491001104814854413	1402.95
11	3	10365	2006-11-27 00:00:00.000	403.20	0.05740335251524064704056674	403.20
12	3	10507	2007-04-15 00:00:00.000	749.06	0.10664324215046170404813225	1152.26
13	3	10535	2007-05-13 00:00:00.000	1940.85	0.27631770022124208781915666	3093.11
14	3	10573	2007-06-19 00:00:00.000	2082.00	0.29641314468435274587911696	5175.11
15	3	10677	2007-09-22 00:00:00.000	813.37	0.11579901992887223482982582	5988.48
16	3	10682	2007-09-25 00:00:00.000	375.50	0.05345971941833547362036907	6363.98
17	3	10856	2008-01-28 00:00:00.000	660.00	0.09396382108149510676283246	7023.98

14. Salin CTE SalesMonth2007 dalam percobaan 2. Tuliskan pernyataan SELECT untuk mengambil kolom monthno dan val. Tambahkan dua kolom yang dihitung:

- (a) avglast3months. Kolom ini harus berisi jumlah penjualan rata-rata untuk tiga bulan terakhir sebelum bulan saat ini menggunakan fungsi window agregat. Asumsikan bahwa tidak ada missing months.
- (b) ytdval Kolom ini harus berisi nilai penjualan kumulatif sampai dengan bulan saat ini.

```
WITH SalesMonth2007 AS (
    SELECT
        monthno = MONTH(orderdate),
        val = SUM(val)
    FROM
        Sales.OrderValues
    WHERE
        YEAR(orderdate) = 2007
    GROUP BY
        MONTH(orderdate)
)
SELECT
    monthno,
    val,
    avglast3months = ISNULL(AVG(val) OVER (ORDER BY monthno ROWS
                                           BETWEEN 3 PRECEDING AND 1 PRECEDING), 0),
    ytdval = SUM(val) OVER (ORDER BY monthno)
FROM
    SalesMonth2007;
```

	monthno	val	avglast3months	ytdval
1	1	61258.08	0.000000	61258.08
2	2	38483.64	61258.080000	99741.72
3	3	38547.23	49870.860000	138288.95
4	4	53032.95	46096.316666	191321.90
5	5	53781.30	43354.606666	245103.20
6	6	36362.82	48453.826666	281466.02
7	7	51020.86	47725.690000	332486.88
8	8	47287.68	47054.993333	379774.56
9	9	55629.27	44890.453333	435403.83
10	10	66749.23	51312.603333	502153.06
11	11	43533.80	56555.393333	545686.86
12	12	71398.44	55304.100000	617085.30