

LAPORAN PRAKTIKUM BASIS DATA

MINGGU 7



TI-2F

Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

1. Menulis Query Menggunakan Fungsi RANKING

Soal 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!

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 1
SELECT
    orderid,
    orderdate,
    val,
    ROW_NUMBER() OVER(ORDER BY orderdate) rowno
FROM Sales.OrderValues;
```

Berikut adalah hasil yang didapat pada 5 baris pertama:

	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

Dan berikut adalah hasil yang didapat pada 5 baris terakhir:

825	11072	2008-05-05 00:00:00.000	5218.00	825
826	11073	2008-05-05 00:00:00.000	300.00	826
827	11074	2008-05-06 00:00:00.000	232.09	827
828	11075	2008-05-06 00:00:00.000	498.10	828
829	11076	2008-05-06 00:00:00.000	792.75	829
830	11077	2008-05-06 00:00:00.000	1255.72	830

Fungsi SQL ROW_NUMBER() akan mendapatkan nilai urutan sebuah data urut dari 1 hingga paling akhir. Dengan tambahan fungsi OVER(), maka urutan ini bisa disesuaikan sesuai kolom tertentu. Pada kasus di atas pengurutan data berdasarkan tanggal (orderdate).

Soal 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!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 2
SELECT
   orderid,
    orderdate,
    val,
    ROW_NUMBER() OVER(ORDER BY orderdate) rowno,
    RANK() OVER(ORDER BY orderdate) rankno
FROM Sales.OrderValues;
```

Data yang dihasilkan sebagai berikut:

Results		Messages			
orderid	orderdate	val	rowno	rankno	
10248	2006-07-04 00:00:00.000	440.00	1	1	
10249	2006-07-05 00:00:00.000	1863.40	2	2	
10250	2006-07-08 00:00:00.000	1552.60	3	3	
10251	2006-07-08 00:00:00.000	654.06	4	3	
10252	2006-07-09 00:00:00.000	3597.90	5	5	

Pada kolom rankno(kolom terakhir) terdapat beberapa baris yang memiliki nilai rankno yang sama. Contohnya pada rowno 3 dan 4 memiliki rankno yang sama. Ini dikarenakan keduanya melakukan transaksi di tanggal yang sama sehingga keduanya dianggap sama.

Soal 3 - Apakah perbedaan antara fungsi RANK dan fungsi ROW_NUMBER?

Jawab: Fungsi ROW_NUMBER() akan mengurutkan data secara urut dari 1 hingga akhir baris. Sedangkan pada RANK() akan mengurutkan data juga secara urut, namun jika terdapat data yang sama, maka itu akan dianggap memiliki nilai rank yang sama.

Soal 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 rangking per pelanggan secara independen, berdasarkan pemesanan val dalam urutan menurun!

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 4
SELECT
   orderid,
    orderdate,
    custid,
    val,
    RANK() OVER(PARTITION BY custid ORDER BY val) orderrankno
FROM Sales.OrderValues;
```

Hasil yang didapat sebagai berikut:

	orderid	orderdate	custid	val	orderrankno
1	10702	2007-10-13 00:00:00.000	1	330.00	1
2	10952	2008-03-16 00:00:00.000	1	471.20	2
3	10643	2007-08-25 00:00:00.000	1	814.50	3
4	10835	2008-01-15 00:00:00.000	1	845.80	4
5	10692	2007-10-03 00:00:00.000	1	878.00	5
6	11011	2008-04-09 00:00:00.000	1	933.50	6
7	10308	2006-09-18 00:00:00.000	2	88.80	1
8	10759	2007-11-28 00:00:00.000	2	320.00	2
9	10625	2007-08-08 00:00:00.000	2	479.75	3
10	10926	2008-03-04 00:00:00.000	2	514.40	4
11	10682	2007-09-25 00:00:00.000	3	375.50	1

Kueri tersebut memanfaatkan PARTITION untuk mengkategorikan data yang sama berdasarkan kolom tertentu. Pada kasus tersebut, data akan dikelompokkan sesuai dengan custid. Kemudian akan diurutkan dengan sesuai nilai val dengan menggunakan RANK().

Soal 5 - Tuliskan pernyataan SELECT untuk mengambil kolom custid dan val dari view Sales.OrderValues. Tambahkan dua kolom berikut:

- orderyear sebagai tahun dari kolom orderdate
- orderrankno sebagai nomor urut, dipartisi berdasarkan pelanggan dan tahun pesanan, dan diurutkan berdasarkan nilai pesanan dalam urutan menurun!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 5
SELECT
    custid,
    val,
    YEAR(orderdate) AS orderyear,
    ROW_NUMBER() OVER (PARTITION BY custid, YEAR(orderdate) ORDER BY val) AS orderrankno
FROM Sales.OrderValues;
```

Dan hasil yang didapat sebagai berikut:

	custid	val	orderyear	orderrankno
1	1	330.00	2007	1
2	1	814.50	2007	2
3	1	878.00	2007	3
4	1	471.20	2008	1
5	1	845.80	2008	2
6	1	933.50	2008	3
7	2	88.80	2006	1
8	2	320.00	2007	1
9	2	479.75	2007	2
10	2	514.40	2008	1

```
Results Messages

(830 rows affected)

Completion time: 2024-10-19T22:16:10.4222199+07:00
```

Kueri SQL tersebut akan melakukan partisi pada dua kolom sekaligus yaitu custid dan orderdate yang telah dikonversi ke YEAR(). Hasil partisi tersebut akan didapatkan nilai ranking data menggunakan fungsi ROW_NUMBER().

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

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 6
WITH cte AS (
    SELECT
        custid,
        val,
        YEAR(orderdate) AS orderyear,
        ROW_NUMBER() OVER (PARTITION BY custid, YEAR(orderdate) ORDER BY val) AS orderrankno
    FROM Sales.OrderValues
)
SELECT *
FROM cte
WHERE cte.orderrankno IN (1, 2);
```

Maka hasil yang didapat sebagai berikut:

	custid	val	orderyear	orderrankno
1	1	330.00	2007	1
2	1	814.50	2007	2
3	1	471.20	2008	1
4	1	845.80	2008	2
5	2	88.80	2006	1
6	2	320.00	2007	1
7	2	479.75	2007	2
8	2	514.40	2008	1
9	3	403.20	2006	1

(418 rows affected)

Completion time: 2024-10-19T22:27:30.6929053+07:00

Kueri tersebut memanfaatkan CTE agar bisa melakukan filter menggunakan klausa WHERE pada kolom orderrrankno. Jika tanpa itu, maka pemfilteran tidak bisa dilakukan karena hasil window function tidak bisa dilakukan di luar klausa SELECT.

Msg 4108, Level 15, State 1, Line 57
 Windowed functions can only appear in the SELECT or ORDER BY clauses.

Completion time: 2024-10-19T22:30:55.9925902+07:00

2. Menulis Query Menggunakan Fungsi OFFSET

Soal 7 - Buatlah (common table expression) CTE dengan nama OrderRows berdasarkan query yang mengambil kolomorderid, 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!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 7
WITH OrderRows AS
(
    SELECT
        orderid, orderdate, val,
        ROW_NUMBER() OVER (ORDER BY orderdate, orderid) rowno
    FROM Sales.OrderValues
)
SELECT * FROM OrderRows;
```

Hasil yang didapat sebagai berikut:

Results		Messages		
	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

Kueri tersebut akan mendapatkan seluruh kolom dari hasil CTE yang sudah ditambahkan fungsi ROW_NUMBER() untuk mendapatkan nomor urut data.

Soal 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!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 8
WITH OrderRows AS
(
    SELECT
       orderid, orderdate, val,
        ROW_NUMBER() OVER (ORDER BY orderdate, orderid) rowno
    FROM Sales.OrderValues
)
SELECT
    o.orderid,
    o.orderdate,
    o.val,
    (p.val) AS prevval,
    (o.val - p.val) AS valdiff
    FROM OrderRows AS o
LEFT OUTER JOIN OrderRows AS p ON o.rowno = p.rowno + 1;
```

Hasil yang didapat sebagai berikut:

Results		Messages			
	orderid	orderdate	val	prevval	valdiff
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

Melanjutkan CTE yang sudah dibuat pada nomor 7. Hasil yang didapat dari CTE dijadikan acuan untuk kueri SELECT yang ada di bawahnya. Untuk mendapatkan data val sebelumnya dan nilai perbedaan dengan data sebelumnya maka bisa melakukan self join. Tabel yang pertama akan dianggap sebagai data terkini. Sedangkan tabel join kedua akan dianggap sebagai data sebelumnya. Ini bisa ditunjukkan pada join kedua tabel pada $o.rowno = p.rowno + 1$. Tabel o untuk data saat ini dan tabel p untuk data data sebelumnya.

Untuk mendapatkan data sebelumnya, maka bisa mengacu pada tabel p. Untuk mendapatkan selisih nilai maka bisa menggunakan perintah $(o.val - p.val)$.

Soal 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!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 9
SELECT
orderid, orderdate, val,
ROW_NUMBER() OVER (ORDER BY orderdate, orderid) rowno,
LAG(val) OVER(ORDER BY orderdate) prevval,
(val - LAG(val) OVER(ORDER BY orderdate)) diffprev
FROM Sales.OrderValues;
```

Maka hasil yang didapat sebagai berikut:

Results		Messages				
	orderid	orderdate	val	rowno	prevval	diffprev
1	10248	2006-07-04 00:00:00.000	440.00	1	NULL	NULL
2	10249	2006-07-05 00:00:00.000	1863.40	2	440.00	1423.40
3	10250	2006-07-08 00:00:00.000	1552.60	3	1863.40	-310.80
4	10251	2006-07-08 00:00:00.000	654.06	4	1552.60	-898.54
5	10252	2006-07-09 00:00:00.000	3597.90	5	654.06	2943.84
6	10253	2006-07-10 00:00:00.000	1444.80	6	3597.90	-2153.10
7	10254	2006-07-11 00:00:00.000	556.62	7	1444.80	-888.18
8	10255	2006-07-12 00:00:00.000	2490.50	8	556.62	1933.88
9	10256	2006-07-15 00:00:00.000	517.80	9	2490.50	-1972.70
10	10257	2006-07-16 00:00:00.000	1119.90	10	517.80	602.10
11	10258	2006-07-17 00:00:00.000	1614.88	11	1119.90	494.98

Fungsi LAG() tersebut akan mendapatkan nilai satu baris sebelumnya berdasarkan tanggal yang ada. Untuk mendapatkan nilai perbedaan (diffprev), maka bisa dilakukan dengan mengurangi nilai pada tabel val saat ini dengan tanggal sebelumnya.

Soal 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!

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 10
WITH SalesMonth2007 AS (
    SELECT
        MONTH(orderdate) AS monthno,
        SUM(val) AS total_sales
    FROM Sales.OrderValues
    WHERE YEAR(orderdate) = 2007
    GROUP BY MONTH(orderdate)
)
SELECT *
FROM SalesMonth2007
ORDER BY monthno;
```

Maka hasil yang didapat sebagai berikut:

Results		Messages
	monthno	total_sales
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

Kueri yang ada di CTE akan handle pengambilan data penjualan dari masing-masing bulan pada tahun 2007. Agar data bisa dilihat, maka perlu dilakukan klausa SELECT di luar CTE. Hasilnya didapatkan 12 baris yang menunjukkan penjualan pada tiap-tiap bulan.

Soal 11 - Tuliskan pernyataan SELECT yang akan mengambil kolom monthno dan val dari CTE

dan tambahkan 3 kolom untuk ditampilkan, yaitu :

- 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)

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 11
WITH SalesMonth2007 AS (
    SELECT
        MONTH(orderdate) AS monthno,
        SUM(val) AS val
    FROM Sales.OrderValues
    WHERE YEAR(orderdate) = 2007
    GROUP BY MONTH(orderdate)
)
SELECT
    monthno,
    val,
    (LAG(val, 3) OVER(ORDER BY monthno) / 3) avglast3months,
    (val - (FIRST_VALUE(val) OVER(ORDER BY monthno))) diffjanuary,
    LEAD(val) OVER(ORDER BY monthno) nextval
FROM SalesMonth2007
ORDER BY monthno;
```

Results		Messages			
	monthno	val	avglast3months	diffjanuary	nextval
1	1	61258.08	NULL	0.00	38483.64
2	2	38483.64	NULL	-22774.44	38547.23
3	3	38547.23	NULL	-22710.85	53032.95
4	4	53032.95	20419.360000	-8225.13	53781.30
5	5	53781.30	12827.880000	-7476.78	36362.82
6	6	36362.82	12849.076666	-24895.26	51020.86
7	7	51020.86	17677.650000	-10237.22	47287.68
8	8	47287.68	17927.100000	-13970.40	55629.27
9	9	55629.27	12120.940000	-5628.81	66749.23
10	10	66749.23	17006.953333	5491.15	43533.80
11	11	43533.80	15762.560000	-17724.28	71398.44
12	12	71398.44	18543.090000	10140.36	NULL

Kueri tersebut memanfaatkan 2 fungsi penting yaitu LAG() dan LEAD(). LAG digunakan untuk mendapatkan data-data terdahulu sedangkan LEAD() digunakan untuk mendapatkan data di masa mendatang.

3. Menulis Query Menggunakan Fungsi Agregasi Window

Tidak dikerjakan