

LAPORAN PRAKTIKUM BASIS DATA

MINGGU 5



TI-2F

Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

1. Menulis Sebuah Query SELECT untuk Mendapatkan Semua Produk Dalam Kategori Tertentu

Soal 1 - Tulislah sebuah query SELECT untuk menampilkan kolom productid, productname, supplierid, unitprice dan kolom discontinued dari tabel Productions.Product. Kemudian filter hasilnya agar hanya menampilkan produk yang ada di dalam kategori Beverages saja (categoryid = 1)

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 1
SELECT
    productid,
    productname,
    supplierid,
    unitprice,
    discontinued
FROM Production.Products
WHERE categoryid = 1;
```

Berikut adalah hasil dari kueri yang telah dijalankan:

Results		Messages			
	productid	productname	supplierid	unitprice	discontinued
1	1	Product HHYDP	1	18,00	0
2	2	Product RECZE	1	19,00	0
3	24	Product QOGNU	10	4,50	1
4	34	Product SWNJY	16	14,00	0
5	35	Product NEVTJ	16	18,00	0
6	38	Product QDOMO	18	263,50	0
7	39	Product LSOFL	18	18,00	0
8	43	Product ZZZHR	20	46,00	0
9	67	Product XLXQF	16	14,00	0
10	70	Product TOONT	7	15,00	0
11	75	Product BWRLG	12	7,75	0
12	76	Product JYGFE	23	18,00	0

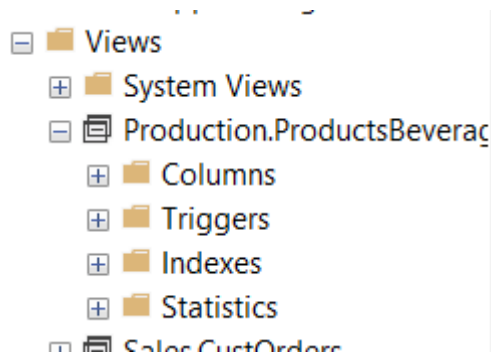
Soal 2 - Modifikasilah kode T-SQL dari no 2 di atas dengan menambahkan kode T-SQL berikut (letakkan sebelum query SELECT):

CREATE VIEW Production.ProductsBeverages AS

Jawab: Kueri yang dibuat sebagai berikut:

```
-- Soal 2
CREATE VIEW Production.ProductsBeverages
AS
SELECT
    productid,
    productname,
    supplierid,
    unitprice,
    discontinued
FROM Production.Products
WHERE categoryid = 1;
```

Dengan begitu akan dihasilkan sebuah view baru:



View yang telah dibuat tersebut bisa digunakan untuk kueri-kueri selanjutnya. Proses kueri selanjutnya juga jauh lebih ringkas karena beberapa perintah sudah ditulis di VIEW.

2. Menulis Query Select terhadap View yang Sudah Dibuat

Soal 3 - Buatlah sebuah query SELECT yang terdiri dari kolom productid dan productname dari VIEW Production.ProductsBeverages. Lalu filterlah hasilnya agar hanya menampilkan produk dengan supplierid = 1.

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 3
SELECT productid, productname
FROM Production.ProductsBeverages
WHERE supplierid = 1;
```

Maka dihasilkan dua baris data berikut:

Results		Messages
	productid	productname
1	1	Product HHYDP
2	2	Product RECZE

Kueri tersebut didapatkan dari hasil kueri yang disimpan pada VIEW bernama Production.ProductsBeverages. Pada kueri yang ada di view tersebut akan mendapatkan data productid, productname, supplierid, unitprice, dan discontinued. Data-data tersebut kemudian akan difilter lagi untuk mendapatkan data yang memiliki categoryid = 1.

Pada kueri yang selanjutnya (kueri di atas), prosesnya masih sama dengan meneruskan perintah yang ada di VIEW. Data yang akan diambil hanya productid dan productname. Kemudian akan difilter lagi agar mendapatkan data supplierid = 1. Pada akhirnya didapatkan 2 baris data yang menunjukkan data yang memiliki categoryid = 1 dan supplierid = 1.

3. Menambahkan Klausula ORDER BY pada VIEW

Soal 4 - Setelah mengeksekusi T-SQL di atas, apakah yang terjadi? Tuliskan pesan error yang ada dan jelaskan penyebab mengapa error tersebut muncul!

Jawab: Jika dieksekusi akan memunculkan error sebagai berikut:

```
Messages
Msg 1333, Level 15, State 1, Procedure ProductsBeverages, Line 6 (Batch Start Line 34)
The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified.
Completion time: 2024-10-07T10:28:47.4150892+07:00
```

Itu karena klausa ORDER BY tidak bisa digunakan pada pembuatan sebuah VIEW. Jika dipaksakan terdapat ORDER BY di sana, ini tidak akan menjamin hasil dari kueri yang ada saat memanggil VIEW ini memiliki urutan yang sama dengan yang sudah diatur di dalam VIEW. Urutan data bisa saja diubah lagi saat pemanggilan VIEW di kueri lain. Jika tetap mau menggunakan VIEW, maka perlu menambahkan opsi TOP atau OFFSET.

Messages

Commands completed successfully.

Completion time: 2024-10-07T10:39:06.3034366+07:00

Soal 5 - Jika sebuah query dijalankan terhadap VIEW Production.ProductsBeverages yang telah dimodifikasi, apakah baris yang dihasilkan dari VIEW tersebut akan selaluurut berdasarkan productname? Jelaskan!

Jawab: Hasil yang didapat sebagai berikut:

Results

Messages

	productid	productname	supplierid	unitprice	discontinued
1	1	Product HHYDP	1	18,00	0
2	2	Product RECZE	1	19,00	0
3	24	Product QOGNU	10	4,50	1
4	34	Product SWNJY	16	14,00	0
5	35	Product NEVTJ	16	18,00	0
6	38	Product QDOMO	18	263,50	0
7	39	Product LSOFL	18	18,00	0
8	43	Product ZZZHR	20	46,00	0
9	67	Product XLXQF	16	14,00	0
10	70	Product TOONT	7	15,00	0
11	75	Product BWRLG	12	7,75	0
12	76	Product JYGFE	23	18,00	0

Pada kueri yang sudah menjadi sebuah VIEW, yaitu Production.ProductsBeverages di situ terdapat klausa ORDER BY productname. Dikarenakan tidak ada keterangan diurutkan secara ascending atau descending, maka secara default akan diurutkan secara ascending. Namun, pada kenyataannya data yang diterima tidak ditampilkan secara urut.

Masalah ini terjadi karena penambahan klausa ORDER BY pada VIEW tidak akan dieksekusi oleh SQL Server. Berikut adalah perbandingan eksekusi langsung dari kueri dan menggunakan view:

```
]SELECT TOP(100) PERCENT
    productid, productname, supplierid, unitprice, discontinued
FROM Production.Products
WHERE categoryid = 1
ORDER BY productname;
```

```
]SELECT *
FROM Production.ProductsBeverages;
```

Results		Messages			
	productid	productname	supplierid	unitprice	discontinued
1	75	Product BWRLG	12	7,75	0
2	1	Product HHYDP	1	18,00	0
3	76	Product JYGFE	23	18,00	0
4	39	Product LSOFL	18	18,00	0
5	35	Product NEVTJ	16	18,00	0
6	38	Product QDOMO	18	263,50	0
7	24	Product QOGNU	10	4,50	1
8	2	Product RECZE	1	19,00	0
9	34	Product SWNJY	16	14,00	0
10	70	Product TOONT	7	15,00	0
11	67	Product XLXQF	16	14,00	0
12	43	Product ZZZHR	20	46,00	0

	productid	productname	supplierid	unitprice	discontinued
1	1	Product HHYDP	1	18,00	0
2	2	Product RECZE	1	19,00	0
3	24	Product QOGNU	10	4,50	1
4	34	Product SWNJY	16	14,00	0
5	35	Product NEVTJ	16	18,00	0
6	38	Product QDOMO	18	263,50	0
7	39	Product LSOFL	18	18,00	0
8	43	Product ZZZHR	20	46,00	0
9	67	Product XLXQF	16	14,00	0
10	70	Product TOONT	7	15,00	0
11	75	Product BWRLG	12	7,75	0
12	76	Product JYGFE	23	18,00	0

Tabel yang atas dihasilkan dari kueri langsung tanpa menggunakan VIEW. Sedangkan pada tabel yang kedua dihasilkan dari view. Jadi, kalau masih ingin mengurutkan data maka klausa ORDER BY dihapus saat pembuatan VIEW. Klausa ORDER BY bisa ditambahkan pada saat kueri pemanggilan VIEW.

4. Menambahkan Kolom ke Dalam VIEW

Soal 6 - Setelah mengeksekusi T-SQL di atas, apakah yang terjadi? Tuliskan pesan error yang ada dan jelaskan penyebab mengapa error tersebut muncul!

Jawab: Kueri yang dijalankan akan memunculkan error sebagai berikut:

Messages

```
Msg 4511, Level 16, State 1, Procedure ProductsBeverages, Line 2 [Batch Start Line 59]
Create View or Function failed because no column name was specified for column 6.

Completion time: 2024-10-07T10:47:13.7276024+07:00
```

Error tersebut muncul karena view yang sudah dibuat tidak memiliki kolom ke-6. Kolom ke 6 sebenarnya ada namun masih belum memiliki nama. Maka dari itu, kolom yang digunakan untuk menyimpan nilai 'high' atau 'normal' perlu diberi alias agar kueri bisa berjalan dengan baik.

Soal 7 - Perbaiki skrip T-SQL di atas sehingga dapat dijalankan dengan benar!

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 7
GO
CREATE OR ALTER VIEW Production.ProductsBeverages AS
SELECT
    productid, productname, supplierid, unitprice, discontinued,
    CASE
        WHEN unitprice > 100 THEN N'high'
        ELSE N'normal'
    END AS price_category
FROM Production.Products
WHERE categoryid = 1;
GO
```

Dengan menambahkan kueri CREATE OR ALTER VIEW dan membuat alias untuk nama kolom kategori harga, maka sudah bisa dibuat sebuah VIEW dengan tambahan kolom baru.

5. Menghapus VIEW

```
-- Praktikum bagian 5
-- Hapus VIEW
IF OBJECT_ID(N'Production.ProductsBeverages', N'V') IS NOT NULL
DROP VIEW Production.ProductsBeverages;
```

Messages

Commands completed successfully.

Completion time: 2024-10-07T11:04:50.2282566+07:00

Jika dilihat pada bagian VIEW database TSQL, sudah ditemukan mengenai VIEW Production.ProductsBeverages.

6. Membuat Query SELECT dalam DERIVED TABLE

Soal 8 - Dengan menggunakan database TSQL2012, buatlah sebuah query SELECT terhadap derived table (tabel turunan) yang berisi kolom productid dan productname, dengan filter hanya menampilkan data yang 'pricetype'-nya adalah 'high'. Gunakan query SELECT yang ada pada Praktikum - Bagian 4 – Langkah 1 sebagai derived table-nya. Beri nama alias p terhadap derived table tersebut.

Jawab: Kueri yang dibuat:

```
-- 8
SELECT
    p.productid, p.productname
FROM (
    SELECT
        productid, productname, supplierid, unitprice, discontinued,
        CASE
            WHEN unitprice > 100 THEN N'high'
            ELSE N'normal'
        END AS price_type
    FROM Production.Products
) AS p
WHERE p.price_type = N'high';
```

Hasil yang didapat sebagai berikut:

Results			Messages	
	productid	productname		
1	29	Product VJXYN		
2	38	Product QDOMO		

7. Membuat Query SELECT untuk Mengetahui Total dan Rata-Rata Jumlah Order (nominal)

Soal 9 - Buatlah sebuah query SELECT untuk mendapatkan kolom custid dan 2 (dua) kolom kalkulasi, yakni totalsalesamount (total jumlah nominal order per customer) dan avgsalesamount (rata-rata jumlah nominal order per customer).

Jawab: Kueri yang digunakan sebagai berikut:


```

SELECT
    k.custid,
    SUM(k.totalsalesamountperorder) totalsalesamount,
    AVG(k.totalsalesamountperorder) avgtotalsalesamount
FROM (
    SELECT
        o.custid,
        SUM(d.qty * d.unitprice) totalsalesamountperorder
    FROM Sales.Orders o
    INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
    GROUP BY o.custid, o.orderid
) AS k
GROUP BY k.custid
ORDER BY k.custid ASC;

```

Hasil kueri sebagai berikut:

	custid	totalsalesamount	avgtotalsalesamount
1	1	4596,20	766,0333
2	2	1402,95	350,7375
3	3	7515,35	1073,6214
4	4	13806,50	1062,0384
5	5	26968,15	1498,2305
6	6	3239,80	462,8285
7	7	19088,00	1735,2727
8	8	5297,80	1765,9333
9	9	23850,95	1402,997
10	10	22607,70	1614,8357

8. Membuat Query SELECT untuk Mendapatkan Persentase Perkembangan Penjualan

Soal 10 - Tulislah sebuah query SELECT yang berisi kolom-kolom berikut ini:

- orderyear: tahun dari tanggal order
- curtotalsales: total jumlah penjualan pada tahun tersebut
- prevtotalsales: total jumlah penjualan pada tahun sebelumnya
- percentgrowth: prosentase perkembangan penjualan dari tahun yang sedang berjalan dibanding tahun sebelumnya

Jawab: Kueri yang digunakan:

```

SELECT
    YEAR(ov.orderdate) orderyear,
    SUM(ov.val) curtotalsales,
    prevorders.prevtotalsales,
    (
        (SUM(ov.val) - prevorders.prevtotalsales)
        / prevorders.prevtotalsales * 100
    ) percentgrowth
FROM Sales.OrderValues ov
LEFT JOIN (
    SELECT
        YEAR(ov.orderdate) orderyear,
        SUM(ov.val) prevtotalsales
    FROM Sales.OrderValues ov
    GROUP BY YEAR(ov.orderdate)
) AS prevorders ON prevorders.orderyear = YEAR(ov.orderdate) -1
GROUP BY YEAR(ov.orderdate), prevorders.prevtotalsales
ORDER BY orderyear ASC;

```

Maka tabel yang diperoleh sebagai berikut:

	orderyear	curtotalsales	prevtotalsales	percentgrowth
1	2006	208083.99	NULL	NULL
2	2007	617085.30	208083.99	196.555800
3	2008	440623.93	617085.30	-28.595900

Kueri tersebut memanfaatkan satu buah derived table untuk mendapatkan data penjualan sebelumnya. Untuk mendapatkan data penjualan terkini dan persentase penjualan maka bisa menggunakan kueri SQL biasa. Isi yang ada pada derived table adalah data orderyear dan total penjualan. Agar bisa mendapatkan data sebelumnya, maka tabel Sales.OrderValues bisa dilakukan proses join dengan dirinya sendiri. Keduanya dihubungkan apabila nilai orderyear saat ini sama dengan tahun sebelumnya.

Kemudian untuk melakukan kalkulasi persentase penjualan dari tahun ke tahun, maka bisa dilakukan perhitungan dengan rumus sederhana yaitu : (data penjualan saat ini - data penjualan tahun sebelumnya) / data penjualan tahun sebelumnya * 100. Proses ini bisa dilakukan di dalam kueri induk.

9. Membuat Query SELECT yang Menggunakan CTE

Soal 11 - Dengan tetap menggunakan database TSQL2012, buatlah query SELECT seperti di Praktikum - Bagian 6, akan tetapi dengan menggunakan Common Table Expressions (CTE). Beri nama alias query CTE tersebut sebagai ProductBeverages.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 11
WITH ProductsBeverages AS
(
    SELECT
        productid, productname, supplierid, unitprice, discontinued,
        CASE WHEN unitprice > 100 THEN N'high'
              ELSE N'normal'
        END AS pricetype
    FROM Production.Products
    WHERE categoryid = 1
)
SELECT productid, productname
FROM ProductsBeverages
WHERE pricetype = 'high';
```

Maka akan menghasilkan data berikut:

Results		Messages
	productid	productname
1	38	Product QDOMO

CTE yang telah dibuat berada di bagian dalam WITH. Dengan adanya CTE, logika untuk mendapatkan suatu data tertentu bisa dipisah-pisah lagi. Ini membuat kueri yang dibuat lebih mudah dibaca dan dikelola.

10. Membuat Query SELECT untuk Mendapatkan Total Jumlah Penjualan (nominalnya) untuk Setiap Customer
11. Membuat Query SELECT untuk Membandingkan Jumlah Total Penjualan untuk Setiap Customer dengan Tahun Sebelumnya
12. Membuat Query SELECT untuk Mendapatkan Total Jumlah Penjualan (nominal) untuk Setiap Customer

Soal 14 - Dengan menggunakan database TSQL2012, buatlah query SELECT terhadap view Sales.OrderValues yang berisi kolom custid dan kolom totalsalesamount (total dari kolom val). Filter hasilnya agar hanya menampilkan order pada tahun 2017 saja.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 14
SELECT
    custid,
    SUM(val) AS totalsalesamount
FROM Sales.OrderValues
WHERE YEAR(orderdate) = 2007
GROUP BY custid;
```

Hasil yang didapat sebagai berikut:

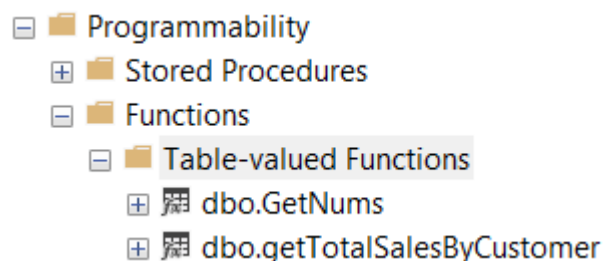
Results		Messages
	custid	totalsalesamount
1	1	2022.50
2	2	799.75
3	3	5960.78
4	4	6406.90
5	5	13849.02
6	6	1079.80
7	7	7817.88
8	8	3026.85

Soal 15 - Buatlah sebuah inline TVF/ Table-Valued Function dengan menambahkan baris berikut ini dan diletakkan sebelum query SELECT pada Langkah 1 di atas !

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 15
GO
CREATE FUNCTION getTotalSalesByCustomer
(@year AS INT) RETURNS TABLE
AS
RETURN
    SELECT
        custid,
        SUM(val) AS totalsalesamount
    FROM Sales.OrderValues
    WHERE YEAR(orderdate) = 2007
    GROUP BY custid;
GO
```

Fungsi bisa dicek pada bagian Programmability > Functions > Table-Valued Functions



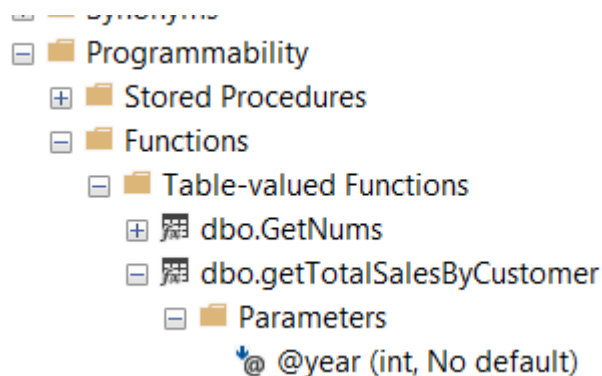
Fungsi ini bisa bermanfaat untuk mendapatkan hasil data berdasarkan kueri yang ada pada sebuah fungsi yang dibuat. Untuk cara penggunaannya maka tinggal panggil nama fungsi dan parameter jika diperlukan.

Soal 16 - Modifikasilah query tersebut dengan mengganti nilai konstanta tahun 2007 pada klausa WHERE, dengan parameter @orderyear.!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 16
GO
ALTER FUNCTION getTotalSalesByCustomer
(@year AS INT) RETURNS TABLE
AS
RETURN
    SELECT
        custid,
        SUM(val) AS totalsalesamount
    FROM Sales.OrderValues
    WHERE YEAR(orderdate) = @year
    GROUP BY custid;
GO
```

Fungsi yang telah dibuat bisa dicek pada bagian berikut:



Bisa terlihat sudah terdapat parameter bernama year pada fungsi getTotalSalesByCustomer. Ini membuat kueri yang ada lebih fleksibel karena bisa melakukan kueri dengan filter tahun sesuai nilai parameter.

13. Membuat Query SELECT yang Beroperasi pada Inline Table-Valued Function

Soal 17 - Buatlah sebuah query SELECT yang berisi kolom custid dan totalsalesamount terhadap inline TVF dbo.fnGetSalesByCustomer. Masukkan nilai 2007 sebagai parameternya.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 17
SELECT
    custid, totalsalesamount
FROM getTotalSalesByCustomer(2007);
```

Maka hasil yang didapat sebagai berikut:

Results		Messages
	custid	totalsalesamount
1	1	2022.50
2	2	799.75
3	3	5960.78
4	4	6406.90
5	5	13849.02
6	6	1079.80
7	7	7817.88
8	8	3026.85
9	9	11208.36

14. Menghapus Inline Table-Valued Function