

LAPORAN PRAKTIKUM BASIS DATA

MINGGU 8



TI-2F

Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

A. UNION & UNION ALL

Soal 1 - Tulis sebuah SQL yang menampilkan hasil pada praktikum-1 langkah-1 & 2 secara sekaligus (gabungan) dengan menggunakan UNION!

Jawab: Kueri yang dibuat:

```
-- Soal 1
SELECT productid, productname
FROM Production.Products
WHERE categoryid = 4
UNION
SELECT p.productid, p.productname
FROM Production.Products p
INNER JOIN Sales.OrderDetails od ON p.productid = od.productid
GROUP BY p.productid, p.productname
HAVING SUM(od.qty * od.unitprice) > 50000;
```

Hasil kueri:

Results		Messages
	productid	productname
1	11	Product QMVUN
2	12	Product OSFNS
3	29	Product VJXYN
4	31	Product XWOXC
5	32	Product NUNAW
6	33	Product ASTMN
7	38	Product QDOMO
8	59	Product UKXRI
9	60	Product WHBYK
10	69	Product COAXA
11	71	Product MYMOI
12	72	Product GEEOO

Soal 2 - Serupa dengan langkah sebelumnya, kali ini tulislah sebuah SQL yang menampilkan hasil pada praktikum-1 langkah-1 & 2 secara sekaligus (gabungan) dengan menggunakan UNION ALL!

Jawab: Kueri yang dibuat:

```
-- Soal 2
SELECT productid, productname
FROM Production.Products
WHERE categoryid = 4
UNION ALL
SELECT p.productid, p.productname
FROM Production.Products p
INNER JOIN Sales.OrderDetails od ON p.productid = od.productid
GROUP BY p.productid, p.productname
HAVING SUM(od.qty * od.unitprice) > 50000;
```

Hasil kueri:

Results		Messages
	productid	productname
1	11	Product QMVUN
2	12	Product OSFNS
3	31	Product XWOXC
4	32	Product NUNAW
5	33	Product ASTMN
6	59	Product UKXRI
7	60	Product WHBYK
8	69	Product COAXA
9	71	Product MYMOI
10	72	Product GEEEO
11	29	Product VJXYN
12	38	Product QDOMO
13	59	Product UKXRI
14	60	Product WHBYK

Soal 3 - Apa bedanya UNION & UNION ALL?

Jawab: Kueri UNION akan mengambil data dari kueri namun tidak mendapatkan data yang sudah duplikat. Sedangkan UNION ALL akan mendapatkan dari semua data termasuk data yang duplikat.

Soal 4 - Tuliskan SQL untuk menampilkan 10 pelanggan dengan nilai pembelian tertinggi pada bulan Januari 2008 serta 10 tertinggi pada bulan Februari 2008.

Jawab: Kueri yang digunakan sebagai berikut:

```

-- Soal 4
SELECT *
FROM (
    SELECT TOP 10
        c.custid, c.contactname, o.orderdate, v.val
    FROM Sales.Orders o
    INNER JOIN Sales.OrderValues v ON v.orderid = o.orderid
    INNER JOIN Sales.Customers c ON o.custid = c.custid
    WHERE MONTH(o.orderdate) = 1 AND YEAR(o.orderdate) = 2008
    ORDER BY v.val DESC
) sales_january
UNION ALL
SELECT *
FROM (
    SELECT TOP 10
        c.custid, c.contactname, o.orderdate, v.val
    FROM Sales.Orders o
    INNER JOIN Sales.OrderValues v ON v.orderid = o.orderid
    INNER JOIN Sales.Customers c ON o.custid = c.custid
    WHERE MONTH(o.orderdate) = 2 AND YEAR(o.orderdate) = 2008
    ORDER BY v.val DESC
) sales_february;

```

Hasil yang diperoleh sebagai berikut:

Results		Messages		
	custid	contactname	orderdate	val
1	39	Song, Lolan	2008-01-06 00:00:00.000	10952.85
2	32	Krishnan, Venky	2008-01-06 00:00:00.000	8446.45
3	71	Navarro, Tom s	2008-01-22 00:00:00.000	4931.92
4	20	Kane, John	2008-01-16 00:00:00.000	4705.50
5	76	Gulbis, Katrin	2008-01-20 00:00:00.000	4581.00
6	63	Veronesi, Giorgio	2008-01-21 00:00:00.000	3812.70
7	89	Smith Jr., Ronaldo	2008-01-30 00:00:00.000	3523.40
8	65	Moore, Michael	2008-01-26 00:00:00.000	2984.00
9	20	Kane, John	2008-01-27 00:00:00.000	2966.50
10	46	Dressler, Marlies	2008-01-09 00:00:00.000	2826.00
11	63	Veronesi, Giorgio	2008-02-02 00:00:00.000	16387.50
12	65	Moore, Michael	2008-02-16 00:00:00.000	11380.00
13	37	Cr?ciun, Ovidiu V.	2008-02-19 00:00:00.000	10835.24
14	20	Kane, John	2008-02-18 00:00:00.000	6379.40
15	37	Cr?ciun, Ovidiu V.	2008-02-26 00:00:00.000	6200.55
16	39	Song, Lolan	2008-02-18 00:00:00.000	5502.11
17	34	Cohen, Shy	2008-02-13 00:00:00.000	3127.50
18	71	Navarro, Tom s	2008-02-18 00:00:00.000	2753.10
19	50	Mace, Donald	2008-02-17 00:00:00.000	2090.00
20	30	Shabalin, Rostislav	2008-02-05 00:00:00.000	2058.46

Kueri tersebut menggabungkan dua buah kueri yang mendapatkan data penjualan masing-masing untuk periode Januari 2008 dan Februari 2008. Data yang diambil adalah hanya data yang berada pada 10 baris teratas. Sebelumnya data sudah diurutkan sesuai dengan nilai val secara descending. Hasil gabungan antara kedua kueri adalah hasil data pada periode Januari 2008 berada pada 10 baris pertama sedangkan pada baris 11-20 diisi oleh data pada periode Februari 2008.

B. CROSS APPLY & OUTER APPLY

Soal 5 - Ubahlah CROSS APPLY pada Praktikum bagian-2.2 menjadi OUTER APPLY!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- Soal 5
SELECT c.custid, c.contactname, p.productid, p.productname, p.totalsalesamount
FROM Sales.Customers c
OUTER APPLY dbo.fnGetTop3ProductsForCustomer(c.custid) p
ORDER BY c.custid;
```

Hasil yang diperoleh sebagai berikut:

	custid	contactname	productid	productname	totalsalesamount
1	1	Allen, Michael	63	Product ICKNK	878,00
2	1	Allen, Michael	59	Product UKXRI	825,00
3	1	Allen, Michael	28	Product OFBNT	775,20
4	2	Hassall, Mark	72	Product GEEEO	348,00
5	2	Hassall, Mark	60	Product WHBYK	340,00
6	2	Hassall, Mark	32	Product NUNAW	320,00
7	3	Peoples, John	11	Product QMVUN	1453,20
8	3	Peoples, John	26	Product HLGZA	936,90
9	3	Peoples, John	59	Product UKXRI	825,00

Query executed success... | LAPTOP-V9Q55RPI (16.0 RTM) | LAPTOP-V9Q55RPI\KarlSc... | TSQL | 00:00:00 | 267 rows

Soal 6 - Modifikasilah SQL yang telah Anda buat dari bagian sebelumnya sehingga SQL tersebut HANYA menampilkan customer yang tidak pernah membeli produk.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- Soal 6
SELECT c.custid, c.contactname, p.productid, p.productname, p.totalsalesamount
FROM Sales.Customers c
OUTER APPLY dbo.fnGetTop3ProductsForCustomer(c.custid) p
WHERE p.productid IS NULL
ORDER BY c.custid;
```

Hasil yang diperoleh sebagai berikut:

Results		Messages			
	custid	contactname	productid	productname	totalsalesamount
1	22	Bueno, Janaina Burdan, Neville	NULL	NULL	NULL
2	57	Tollefsen, Bjørn	NULL	NULL	NULL

C. EXCEPT & INTERSECT

Soal 7 - Buatlah sebuah statement SELECT yang menampilkan kolom 'custid' dari tabel 'Sales.Orders'. Saring hasilnya sehingga yang tampil hanyalah pelanggan yang berasal dari USA kecuali SEMUA pelanggan yang muncul pada hasil query pada praktikum bagian 3.1.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- Soal 7
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.Customers c ON o.custid = c.custid
WHERE c.country = 'USA'
EXCEPT
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT d.productid) > 20;
```

Hasil yang diperoleh sebagai berikut:

Results		Messages			
	custid				
1	32				
2	36				
3	43				
4	45				
5	48				
6	55				
7	75				
8	77				
9	78				
10	82				

Soal 8 - Salin SQL pada bagian 3.1 tambahkan operator INTERSECT dibelakangnya, kemudian salin-tempel SQL pada bagian 3.3 dibelakang operator INTERSECT tadi. Jalankan, dan perhatikan hasilnya!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- Soal 8
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING SUM(d.qty * d.unitprice) > 10000
INTERSECT
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT d.productid) > 20;
```

Hasil yang diperoleh sebagai berikut:

Results		Messages	
	custid		
1	4		
2	5		
3	7		
4	9		
5	10		
6	20		
7	24		
8	25		
9	30		
10	34		
11	35		
12	37		

Query execut... | LAPTOP-V9Q55RPI (16.0 RTM) | LAPTOP-V9Q55RPI\KarlSc... | TSQL | 00:00:00 | 32 rows

Soal 9 - Dapatkah anda menyimpulkan, customer yang bagaimana yang tampil pada hasil query bagian 3.4 ini ?

Jawab: Customer yang muncul adalah customer yang memiliki kriteria telah belanja lebih dari 20 produk dan memiliki total belanja lebih dari 10000.

Soal 10 - Salin keseluruhan query pada bagian-3.4, modifikasi SQL tersebut dengan cara mengapit statement SELECT sebelum operator INTERSECT dengan tanda kurung '(' dan ')'.
-- Soal 8
(
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING SUM(d.qty * d.unitprice) > 10000
INTERSECT
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT d.productid) > 20;
)

Jawab: Kueri yang digunakan sebagai berikut:

```
-- Soal 9
(
    SELECT o.custid
    FROM Sales.Orders o
    INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
    GROUP BY o.custid
    HAVING SUM(d.qty * d.unitprice) > 10000
)
INTERSECT
SELECT o.custid
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT d.productid) > 20;
```

Hasil yang diperoleh sebagai berikut:

Results		Messages
	custid	
1	4	
2	5	
3	7	
4	9	
5	10	
6	20	
7	24	
8	25	
9	30	
10	34	
11	35	
12	37	

LAPTOP-V9Q55RPI (16.0 RTM) | LAPTOP-V9Q55RPI\KarlSc... | TSQL | 00:00:00 | 32 rows

Soal 11 - Apakah hasilnya berbeda dengan SQL pada bagian-3.4? Mengapa demikian? Dapatkah Anda menjelaskan tentang urutan prioritas (precedence) operator yang digunakan pada SQL bagian ini?

Jawab: Hasilnya tetap sama seperti sebelumnya.

D. TRIGGER (AFTER)

Contoh kasus: Perhatikan tabel 'Sales.OrderDetails', pada tabel tersebut terdapat kolom 'productid' yang merupakan Foreign Key yang mengacu pada tabel 'Production.Products'. Misalkan kita ingin supaya: ketika sebuah 'productid' dihapus semuanya dari tabel 'OrderDetails' maka kolom 'discontinued' diubah nilainya menjadi '1', kita dapat menggunakan TRIGGER AFTER DELETE.

Soal 12 - Buatlah TRIGGER yang dapat menyelesaikan permasalahan pada contoh kasus diatas!

Jawab: Trigger yang dibuat:

```
CREATE TRIGGER trgAutoUpdateDiscontinuedCol ON Sales.OrderDetails
AFTER DELETE
AS
UPDATE Production.Products
SET discontinued = 1
WHERE productid IN (SELECT productid FROM deleted);
```

Kueri yang diujicobakan:

```
DELETE FROM Sales.OrderDetails
WHERE productid = 10;
```

```
SELECT * FROM Production.Products WHERE productid = 10;
```

Hasil yang diperoleh:

Results		Messages				
	productid	productname	supplierid	categoryid	unitprice	discontinued
1	10	Product YHXGE	4	8	31,00	1

Trigger tersebut akan terjadi saat pengguna melakukan delete pada salah satu atau banyak baris pada tabel Sales.OrderDetails. Karena dimungkinkan banyak, maka filter yang ada di WHERE perlu diberikan klausa IN agar bisa memenuhi beberapa opsi baris yang dipilih. Jika menggunakan tanda sama dengan, maka akan terjadi error dan proses tidak bisa dijalankan.

E. TRIGGER (INSTEAD OF)

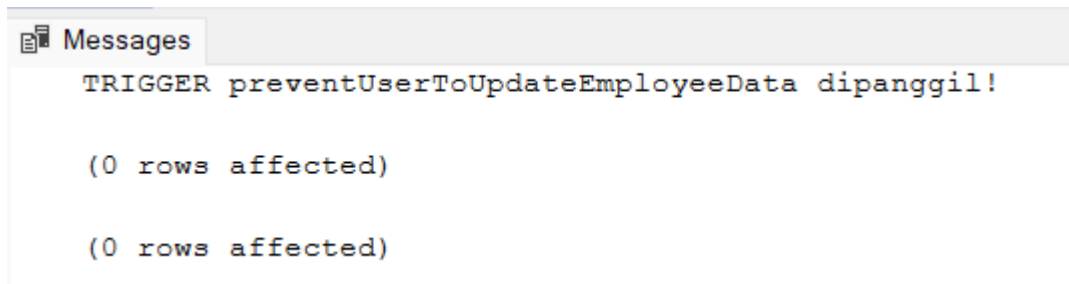
Soal 13 - Dengan cara yang serupa dengan langkah sebelumnya, buatlah TRIGGER yang mencegah user melakukan UPDATE ke table 'HR.Employee'. Ketika ada UPDATE yang terjadi, terapkan hasilnya ke tabel 'HR.EmployeesBackup'!

Jawab: Trigger yang dibuat:

```
]CREATE TRIGGER preventUserToUpdateEmployeeData ON HR.Employees
INSTEAD OF UPDATE
AS
    PRINT 'TRIGGER preventUserToUpdateEmployeeData dipanggil!';

]    INSERT INTO HR.EmployeesBackup
        (LastName, FirstName, Title, TitleOfCourtesy, BirthDate, HireDate,
        Address, City, Region, PostalCode, Country, Phone, MGrid)
    SELECT LastName, FirstName, Title, TitleOfCourtesy, BirthDate, HireDate,
        Address, City, Region, PostalCode, Country, Phone, MGrid
    FROM inserted;
```

Hasil uji coba melakukan pembaruan data pada tabel HR.Employees:



The screenshot shows the 'Messages' window in SQL Server. It displays the following text:

```
TRIGGER preventUserToUpdateEmployeeData dipanggil!

(0 rows affected)

(0 rows affected)
```

Soal 14 - Buatlah TRIGGER yang mencegah user melakukan DELETE ke table 'HR.Employee'. Ketika ada DELETE yang terjadi, jangan biarkan ada data pada tabel tersebut yang hilang! Hapus data yang sama 'HR.EmployeesBackup'!

Jawab: Trigger yang dibuat:

```
]CREATE TRIGGER preventUserToDeleteEmployeeData ON HR.Employees
INSTEAD OF DELETE
AS
    PRINT 'TRIGGER preventUserToDeleteEmployeeData dipanggil!';

]    DELETE FROM HR.EmployeesBackup
    WHERE empid = (SELECT empid deleted);
```

Hasil ketika dijalankan:

 Results  Messages

TRIGGER preventUserToDeleteEmployeeData dipanggil!

(10 rows affected)

(1 row affected)

(0 rows affected)

Completion time: 2024-11-02T17:58:02.3979991+07:00