

LAPORAN BASIS DATA

MINGGU 2



TI-2F

Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Praktikum Bagian 1: Mengeksekusi Sebagian maupun keseluruhan script SQL

Soal 1 – Apa perbedaan hasil kueri berikut ?

Kueri 1:

```
SELECT * FROM Sales.Customers;
```

Kueri 2:

```
SELECT
    [custid],
    [companyname],
    [contactname],
    [contacttitle],
    [address],
    [city],
    [region],
    [postalcode],
    [country],
    [phone],
    [fax]
FROM
    [Sales].[Customers];
```

Jawab:

Kedua kueri tersebut memiliki hasil yang sama saja. Kueri 1 jauh lebih ringkas karena bisa menampilkan data dari seluruh kolom sekaligus hanya dengan satu karakter saja, yaitu bintang (*). Sedangkan kueri 2 mengharuskan untuk menuliskan nama-nama kolomnya sekaligus. Cara kedua lebih cocok ketika hanya menginginkan sebagian kolom saja yang diambil datanya.

Praktikum Bagian 2: Penggunaan Statement SELECT untuk kolom tertentu

Hasil dari kueri berikut:

```
SELECT
    contactname, address, postalcode, city, country
FROM Sales.Customers;
```

adalah gambar di bawah ini:

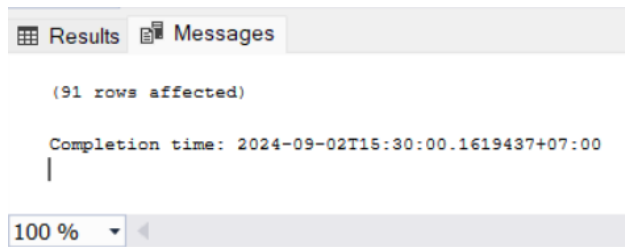
	contactname	address	postalcode	city	country
1	Allen, Michael	Obere Str. 0123	10092	Berlin	Germany
2	Hassall, Mark	Avda. de la Constituci n 5678	10077	M xico D.F.	Mexico
3	Peoples, John	Mataderos 7890	10097	M xico D.F.	Mexico
4	Arndt, Torsten	7890 Hanover Sq.	10046	London	UK
5	Higginbotham, Tom	Berguvsv gen 5678	10112	Lule	Sweden
6	Poland, Carole	Forsterstr. 7890	10117	Mannheim	Germany
7	Bansal, Dushyant	2345, place Kl ber	10089	Strasbourg	France
8	Ilyina, Julia	C/ Araquil, 0123	10104	Madrid	Spain
9	Raghav, Amritansh	6789, rue des Bouchers	10105	Marseille	France
10	Bassole, Pilar Coloma	8901 Teawassan Blvd	10111	Teawassan	Canada

Praktikum Bagian 3: Penggunaan statement SELECT untuk menampilkan data secara unique/DISTINCT

Soal 2 – Apakah ada data yang terduplikasi? Jika YA mengapa?

Jawab: Iya, dengan kueri `SELECT country FROM Sales.Customers;` maka akan menampilkan keseluruhan data pada tiap-tiap baris dari kolom country.

	country
1	Germany
2	Mexico
3	Mexico
4	UK
5	Sweden
6	Germany
7	France
8	Spain
9	France
10	Canada
11	UK
12	Argentina
13	Mexico
14	Switzerland

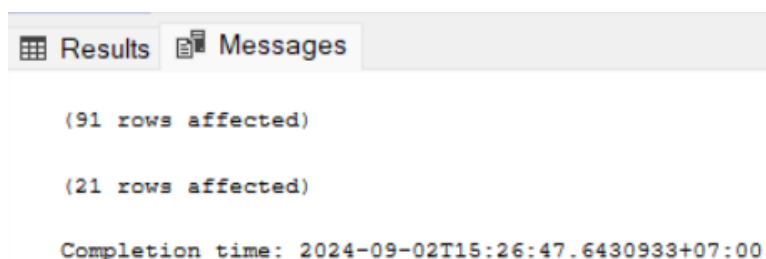


Bisa dilihat bahwa cukup banyak nama negara-negara yang sudah muncul di baris sebelumnya namun juga tetap dimunculkan di baris selanjutnya. Ini karena kueri yang telah dilakukan tidak mencoba untuk mengecek apakah data yang diambil sudah ada atau belum. Itu akan mencoba mengambil seluruh data apa adanya di keseluruhan baris.

Soal 3 – Apakah ada data yang terduplikasi? Jelaskan perbedaan hasil pada Langkah tahap 4 dan tahap 3! Apa manfaat dari perintah DISTINCT?

Jawab: Hasilnya sebagai berikut:

	country
13	Norway
14	Poland
15	Portugal
16	Spain
17	Sweden
18	Switzerl...
19	UK
20	USA
21	Venezu...



Total data yang diambil hanya sampai 21. Ini jauh berbeda dengan kueri yang sebelumnya yang bisa mencapai 91 baris. Dengan penambahan satu kata “DISTINCT”, maka data yang diambil sudah dipilih-pilih. Hanya data yang benar-benar belum ada sebelumnya yang bakal ditampilkan ke pengguna.

Praktikum Bagian 4: Penggunaan ALIAS untuk nama tabel dan nama kolom

Kueri 1

```
SELECT
    c.contactname,
    c.contacttitle
FROM Sales.Customers AS c;
```

Kueri 2

```
SELECT
    c.contactname AS Name,
    c.contacttitle AS Title,
    c.companyname AS "Company Name"
FROM Sales.Customers AS c;
```

Soal 4 – Apa yang membedakan hasil eksekusi dari kueri tahap 1 dan tahap 3 di atas ?
Apa manfaat dari perintah AS? Silakan jelaskan!

Jawab: Kueri 1 hanya menampilkan data dari 2 kolom sedangkan pada kueri 2 menampilkan 3 kolom. Selain itu, seluruh kolom pada kueri 2 diberikan ALIAS (AS). Berbeda dengan kueri 1 yang mana alias hanya digunakan pada nama tabel. Penggunaan alias ini bisa bermanfaat kemudahan kustomisasi nama kolom saat ditampilkan ke pengguna. Terkadang nama kolom dari database tidak melulu sesuai dengan apa yang diinginkan client. Maka dari itu, perlu kustomisasi nama table saat pengambilan data.

Results		Messages	
	Name	Title	Company Name
1	Allen, Michael	Sales Representative	Customer NRZBB
2	Hassall, Mark	Owner	Customer MLTDN
3	Peoples, John	Owner	Customer KBUDE
4	Arndt, Torsten	Sales Representative	Customer HFBZG
5	Higginbotham, Tom	Order Administrator	Customer HGVLZ
6	Poland, Carole	Sales Representative	Customer XHXJV
7	Bansal, Dushyant	Marketing Manager	Customer QXVLA
8	Ilyina, Julia	Owner	Customer QUHWH
9	Raghav, Amritansh	Owner	Customer RTXGC
10	Bassols, Pilar Colome	Accounting Manager	Customer EEALV
11	Jaffe, David	Sales Representative	Customer UBHAU

Praktikum Bagian 5: Penggunaan CASE

Kueri 1

```
SELECT p.categoryid, p.productname  
FROM Production.Products AS p;
```

Kueri 2

```
SELECT  
    p.categoryid, p.productname,  
    CASE  
        WHEN p.categoryid = 1 THEN 'Beverages'  
        WHEN p.categoryid = 2 THEN 'Condiments'  
        WHEN p.categoryid = 3 THEN 'Confections'  
        WHEN p.categoryid = 4 THEN 'Dairy Products'  
        WHEN p.categoryid = 5 THEN 'Grains/Cereals'  
        WHEN p.categoryid = 6 THEN 'Meat/Poultry'  
        WHEN p.categoryid = 7 THEN 'Produce'  
        WHEN p.categoryid = 8 THEN 'Seafood'  
        ELSE 'Other'  
    END AS categoryname  
FROM Production.Products p;
```

Soal 5 – Apa yang membedakan hasil eksekusi dari kueri tahap 1 dan tahap 3 di atas ?
Apa manfaat dari perintah CASE?

Jawab: Kueri 1 hanya menampilkan data murni dari apa yang ada di table Production.Products. Sedangkan kueri 2 menampilkan data dengan tambahan satu kolom baru (categoryname) yang mana datanya diisi dari sesuatu yang belum ada di table. Nilai data ‘Beverages’, ‘Condiments’, ‘Confections’, ‘Dairy Products’, dll itu sebelumnya belum ada di tabel. Pemberian nilai ke kolom “categoryname” ini didasarkan pada nilai pada kolom “categoryid”. Jika memenuhi syarat, maka nilai tersebut akan diberikan.

Results		Messages	
	categoryid	productname	categoryname
1	1	Product HHYDP	Beverages
2	1	Product RECZE	Beverages
3	2	Product IMEHJ	Condiments
4	2	Product KSBRM	Condiments
5	2	Product EPEIM	Condiments
6	2	Product VAIIV	Condiments
7	7	Product HMLNI	Produce
8	2	Product WJFJP	Condiments
9	6	Product AOZBW	Meat/Poultry
10	8	Product YHXGE	Seafood
11	4	Product QMVUN	Dairy Products
12	4	Product OSFNS	Dairy Products
13	8	Product BOYEL	Seafood

Kueri 3

SELECT

p.categoryid, p.productname,

CASE

WHEN p.categoryid = 1 THEN 'Beverages'

WHEN p.categoryid = 2 THEN 'Condiments'

WHEN p.categoryid = 3 THEN 'Confections'

WHEN p.categoryid = 4 THEN 'Dairy Products'

WHEN p.categoryid = 5 THEN 'Grains/Cereals'

WHEN p.categoryid = 6 THEN 'Produce'

WHEN p.categoryid = 7 THEN 'Seafood'

ELSE 'Other'

END AS categoryname,

CASE

WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'

ELSE 'Non-Campaign Products'

END AS iscampaign

FROM Production.Products AS p;

Soal 6 – Silakan capture hasilnya, data apa yang didapatkan dari perintah kueri di atas? Jelaskan!

Jawab: Hasilnya sebagai berikut:

Results		Messages		
	categoryid	productname	categoryname	iscampaign
1	1	Product HHYDP	Beverages	Campaign Products
2	1	Product RECZE	Beverages	Campaign Products
3	2	Product IMEHJ	Condiments	Non-Campaign Products
4	2	Product KSBRM	Condiments	Non-Campaign Products
5	2	Product EPEIM	Condiments	Non-Campaign Products
6	2	Product VAIIV	Condiments	Non-Campaign Products
7	7	Product HMLNI	Seafood	Campaign Products
8	2	Product WJFJP	Condiments	Non-Campaign Products
9	6	Product AOZBW	Produce	Non-Campaign Products
10	8	Product YHXGE	Other	Campaign Products
11	4	Product QMVUN	Dairy Products	Non-Campaign Products
12	4	Product QSEMS	Dairy Products	Non-Campaign Products

Kueri tersebut dihasilkan dengan menambahkan kueri berikut:

CASE

WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'

ELSE 'Non-Campaign Products'

END AS iscampaign

Kueri tersebut digunakan untuk membuat kolom baru yang isinya berupa kategori apakah produk tersebut termasuk “Campaign Products” atau “Non-Campaign Products”.

Soal 7 – Menampilkan data berdasarkan kategori “seafood” saja dengan memanfaatkan perintah ALIAS.

Jawab:

Kode yang perlu ditambahkan adalah satu baris berikut:

WHERE p.categoryid = 8;

Dengan begitu data yang ditampilkan hanya berfokus pada kategori yang memiliki nilai 8.

Results		Messages		
	ID_KATEGORI	NAMA_PRODUK	NAMA_KATEGORI	iscampaign
1	8	Product YHXGE	Seafood	Campaign Products
2	8	Product POXFU	Seafood	Campaign Products
3	8	Product CKEDC	Seafood	Campaign Products
4	8	Product LYERX	Seafood	Campaign Products
5	8	Product GMKIJ	Seafood	Campaign Products
6	8	Product EVFFA	Seafood	Campaign Products
7	8	Product YZIXQ	Seafood	Campaign Products

Keseluruhan kueri sebagai berikut:

```

SELECT
    p.categoryid AS ID_KATEGORI,
    p.productname AS NAMA_PRODUK,
    CASE
        WHEN p.categoryid = 1 THEN 'Beverages'
        WHEN p.categoryid = 2 THEN 'Condiments'
        WHEN p.categoryid = 3 THEN 'Confections'
        WHEN p.categoryid = 4 THEN 'Dairy Products'
        WHEN p.categoryid = 5 THEN 'Grains/Cereals'
        WHEN p.categoryid = 6 THEN 'Meat/Poultry'
        WHEN p.categoryid = 7 THEN 'Produce'
        WHEN p.categoryid = 8 THEN 'Seafood'
        ELSE 'Other'
    END AS NAMA_KATEGORI,
    CASE
        WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
        ELSE 'Non-Campaign Products'
    END AS iscampaign
FROM Production.Products AS p
WHERE p.categoryid = 8;

```

Soal 8 – Tampilkan data employees dari tabel HR.Employees yang berasal dari negara USA dan kota ‘Seattle’

Jawab: Hasilnya sebagai berikut:

Results		Messages		
	FIRST_NAME	LAST_NAME	CITY	COUNTRY
1	Sara	Davis	Seattle	USA
2	Maria	Cameron	Seattle	USA

Dengan menggunakan kueri sebagai berikut maka bisa didapatkan hasil sesuai tabel di atas:

```
SELECT
    emp.firstname AS FIRST_NAME,
    emp.lastname AS LAST_NAME,
    emp.city AS CITY,
    emp.country AS COUNTRY
FROM HR.Employees AS emp
WHERE
    COUNTRY = 'USA' AND CITY = 'Seattle';
```

Memanfaatkan klausa WHERE maka data yang ditampilkan sudah bisa difokuskan ke data yang memiliki COUNTRY “USA” dan CITY “SEATTLE”.

Praktikum Bagian 6: Membuat Query Inner JOIN

Soal 9 – Menuliskan T-SQL SELECT yang menampilkan kolom productname dari tabel Production.Products (gunakan tabel alias “p”) dan kolom categoryname dari tabel Production.Categories (gunakan tabel alias “c”) menggunakan INNER JOIN!

Jawab: Kueri yang digunakan sebagai berikut:

```
SELECT
    p.productname,
    c.categoryname
FROM [Production].[Products] AS p
INNER JOIN
    [Production].[Categories] AS c ON p.categoryid = c.categoryid;
```

	productname	categoryname
1	Product HHYDP	Beverages
2	Product RECZE	Beverages
3	Product IMEHJ	Condiments
4	Product KSBRM	Condiments
5	Product EPEIM	Condiments
6	Product VAIIV	Condiments
7	Product HMLNI	Produce
8	Product WWJFP	Condiments
9	Product AQZBW	Meat/Poultry

Kueri tersebut akan mengambil data dua kolom dari tabel Production.Products dan Production.Categories. Data yang diambil ini didasarkan pada kesamaan categoryid yang dimiliki pada masing-masing tabel. Jika ditemukan memiliki pasangan di antar tabel maka data tersebut akan diambil dan dipasangkan dengan data di kolom sebelumnya.

Soal 10 – Kolom mana yang ditentukan sebagai predikat dalam klausa ON join ? Mengapa?

Jawab: Kolom yang digunakan adalah categoryid di masing-masing kedua kolom. Ini membuat jika masing-masing tabel memiliki nilai yang sama pada kolom categoryid, maka data tersebut akan diambil.

Praktikum Bagian 7: Membuat Query Inner Join pada Banyak Tabel

Soal 11 – Setelah percobaan tahap ke-2 dilakukan, maka akan muncul error. Apakah isi pesan error tersebut? Kenapa kesalahan tersebut dapat terjadi ? Jelaskan!

Jawab: Isi pesannya adalah:

```

Messages
Msg 209, Level 16, State 1, Line 23
Ambiguous column name 'custid'.

Completion time: 2024-09-03T11:08:02.4406654+07:00

```

Ini karena kolom yang ingin diambil custid ini berada di dua tabel yang berbeda. Jadi, jika tidak sebutkan nama tabel yang disebutkan, maka program akan bingung kolom custid mana yang harus diambil.

Soal 12 – Pada uji coba ke-4 ini lakukan perbaikan error yang terjadi pada uji coba tahap ke-3 yang menjelaskan jika semua nama tabel memiliki identitas tabel masing-masing!

Jawab: Untuk solusinya bisa dengan mengubah kueri menjadi sebagai berikut:

SELECT

Customers.custid, contactname,orderid

FROM Sales.Customers

INNER JOIN Sales.Orders ON Customers.custid = Orders.custid;

Hasilnya sebagai berikut:

	custid	contactname	orderid
1	1	Allen, Michael	10643
2	1	Allen, Michael	10692
3	1	Allen, Michael	10702
4	1	Allen, Michael	10835
5	1	Allen, Michael	10952
6	1	Allen, Michael	11011
7	2	Hassall, Mark	10308
8	2	Hassall, Mark	10695

Dari kueri tersebut sudah dinyatakan dengan jelas bahwa kolom custid diambil dari tabel Customers. Bisa juga diambil dari tabel Orders. Keduanya sama-sama benar karena kedua tabel memiliki isi kolom yang sama.

Soal 13 – Salin T-SQL pada uji coba tahap ke-4 dan modifikasi dengan menggunakan tabel alias “c” untuk tabel Sales.Customers dan “o” untuk tabel Sales.Orders.

Jawab: Kueri yang digunakan sebagai berikut:

SELECT

c.custid, contactname,orderid

FROM Sales.Customers AS c

INNER JOIN Sales.Orders AS o ON c.custid = o.custid;

	custid	contactname	orderid
1	1	Allen, Michael	10643
2	1	Allen, Michael	10692
3	1	Allen, Michael	10702
4	1	Allen, Michael	10835
5	1	Allen, Michael	10952
6	1	Allen, Michael	11011

Kueri pada tahap 8

```
SELECT
```

```
    Customers.custid, Customers.contactname, Orders.orderid
```

```
FROM Sales.Customers AS c
```

```
INNER JOIN Sales.Orders AS o ON c.custid = o.custid;
```

Soal 14 – Kenapa hasil eksekusi T-SQL tahap ke-8 mendapatkan hasil error ?

Jawab: Kueri tersebut error karena pemanggilan nama tabel tidak sesuai dengan nama alias. Di bagian klausa FROM hingga INNER JOIN dituliskan bahwa nama tabel yang digunakan adalah “c” untuk “Sales.Customers” dan “o” untuk “Sales.Orders”. Dengan demikian, seluruh pemanggilan tabel yang digunakan pada satu kueri tersebut harus sesuai dengan nama alias.

Soal 15 – Rubahlah prefix nama kolom pada T-SQL uji coba tahap ke-8 dengan nama aslinya, kemudian tampilkan hasil eksekusinya!

Jawab: Kueri yang digunakan sebagai berikut:

```
SELECT
```

```
    o.custid, c.contactname, o.orderid
```

```
FROM Sales.Customers AS c
```

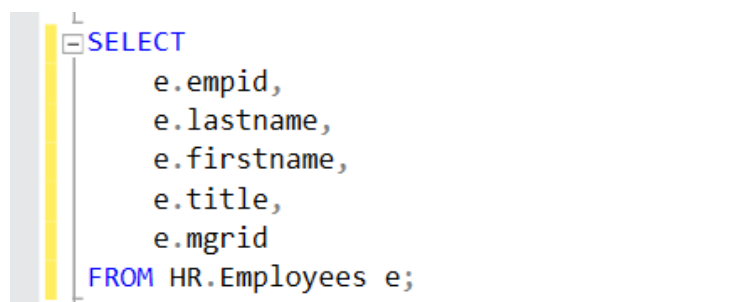
```
INNER JOIN Sales.Orders AS o ON c.custid = o.custid;
```

Perubahannya terletak pada pemanggilan nama tabel di bagian klausa SELECT menjadi sesuai dengan alias, yaitu “c” untuk “Sales.Customers” dan “o” untuk “Sales.Orders”.

Praktikum Bagian 8: Membuat Query Self-Join

Soal 16 – Tuliskan T-SQL menggunakan klausa SELECT untuk menampilkan kolom empid, lastname, firstname, title, dan mgrid pada tabel HR.Employees dengan memberikan nama alias “e” untuk tabel HR.Employees!

Jawab: Kueri dan hasil yang didapat:



```
SELECT
    e.empid,
    e.lastname,
    e.firstname,
    e.title,
    e.mgrid
FROM HR.Employees e;
```

110 %

	empid	lastname	firstname	title	mgrid
1	1	Davis	Sara	CEO	NULL
2	2	Funk	Don	Vice President, Sales	1
3	3	Lew	Judy	Sales Manager	2
4	4	Peled	Yael	Sales Representative	3
5	5	Buck	Sven	Sales Manager	2
6	6	Suurs	Paul	Sales Representative	5
7	7	King	Russell	Sales Representative	5
8	8	Cameron	Maria	Sales Representative	3
9	9	Dolgopyatova	Zoya	Sales Representative	5

Perintah kueri tersebut hanya akan mengambil seluruh data di kolom empid, lastname, firstname, mgrid dari tabel HR.Employees dengan memanfaatkan alias untuk nama tabel menjadi "e".

Soal 17 – Eksekusi uji coba tahap ke-2 dan bandingkan dengan 72 -Lab Exercise 3 – Task 1 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar!

Jawab: Hasil yang didapat sudah benar dengan kesamaan menampilkan tabel dengan terdiri dari 5 kolom dan dengan total baris sebanyak 9.

Soal 18 – Salin T-SQL pada tahap ke-2 kemudian modifikasi dengan menambahkan kolom mengenai informasi manajer yaitu lastname, firstname menggunakan SELF-JOIN. Gunakan nama alias mgrlastname dan mgrfirstname untuk membedakan nama manajer dan karyawan!

Jawab: Kueri dan hasil yang didapat sebagai berikut:

```

SELECT
    e.empid,
    e.lastname,
    e.firstname,
    e.title,
    e.mgrid,
    m.lastname mgrlastname,
    m.firstname mgrfirstname
FROM HR.Employees m
INNER JOIN HR.Employees e ON e.empid = m.empid
WHERE m.mgrid IS NOT NULL;

```

Results		Messages					
	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	2	Funk	Don	Vice President, Sales	1	Funk	Don
2	3	Lew	Judy	Sales Manager	2	Lew	Judy
3	4	Peled	Yael	Sales Representative	3	Peled	Yael
4	5	Buck	Sven	Sales Manager	2	Buck	Sven
5	6	Suurs	Paul	Sales Representative	5	Suurs	Paul
6	7	King	Russell	Sales Representative	5	King	Russell
7	8	Cameron	Maria	Sales Representative	3	Cameron	Maria
8	9	Dolgopyatova	Zoya	Sales Representative	5	Dolgopyatova	Zoya

Kueri tersebut akan mengambil seluruh data karyawan yang juga telah dilakukan proses INNER JOIN agar bisa dihubungkan dengan data manajernya. Data manajer yang bisa ditampilkan adalah manajer ide (mgrid), manajer lastname (mgrlastname), manajer firstname (mgrfirstname). Sebagai tambahan, bisa ditambahkan klausa WHERE agar hasil yang didapat tidak menampilkan data yang NULL.

Soal 19 – Eksekusi uji coba tahap ke-2 dan bandingkan dengan 73 – Lab Exercise 3 – Task 2 Result.txt

Jawab: Hasil yang didapat sudah sesuai dengan ditunjukkan tabel yang hasil kueri yang terdiri dari 6 kolom 8 baris.

Soal 20 – Apakah merupakan suatu keharusan untuk menuliskan nama alias tabel jika disaat melakukan perintah SELF-JOIN ? Apakah dapat digunakan nama tabel asli sebagai nama alias ? Jelaskan!

Jawab: Tidak bisa menggunakan tanpa pemberian nama alias. Pemberian alias untuk tabel yang akan dilakukan proses SELF-JOIN akan memudahkan pembedaan 2 tabel berbeda. Sehingga proses self join bisa dilakukan. Sesungguhnya proses self join ini seperti memerlukan dua tabel yang berbeda. Namun karena prosesnya self join, maka hanya perlu satu tabel dengan memberikan penanda bahwa itu seolah-olah tabel yang berbeda.

Jika dipaksakan dengan tanpa pemberian alias, maka hasilnya akan error.

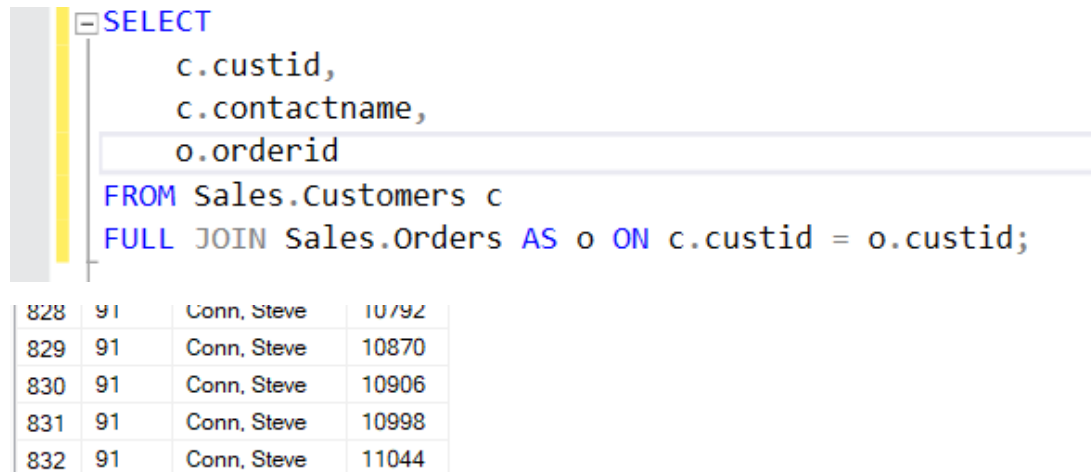
Msg 1013, Level 16, State 1, Line 55
The objects "HR.Employees" and "HR.Employees" in the FROM clause have the same exposed names. Us

Program akan bingung tabel mana yang digunakan.

Praktikum Bagian 9: Membuat Query Outer Join

Soal 21 – Tuliskan perintah T-SQL dengan klausa SELECT untuk mengambil kolom custid dan contactname dari tabel Sales.Customers dan kolom orderid dari tabel Sales.Orders tabel. Perintah yang dibuat harus mengambil semua baris dari tabel Sales.Customers.

Jawab: Kueri dan hasil yang didapat sebagai berikut:



```
SELECT
    c.custid,
    c.contactname,
    o.orderid
FROM Sales.Customers c
FULL JOIN Sales.Orders AS o ON c.custid = o.custid;
```

828	91	Conn, Steve	10/92
829	91	Conn, Steve	10870
830	91	Conn, Steve	10906
831	91	Conn, Steve	10998
832	91	Conn, Steve	11044

Klausa FULL JOIN ini sama artinya dengan kueri OUTER JOIN. Ini membuat seluruh data akan diambil dari masing-masing tabel. Jika didapatkan data yang tidak memiliki nilai pasangan di tabel lain, maka pasangan nilainya adalah NULL. Kebetulan saja data dari masing-masing tabel memiliki pasangan sehingga tidak ada NULL.

Soal 22 – Eksekusi uji coba tahap ke-2 dan bandingkan dengan 82 – Lab Exercise 4 – Task 1 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar.

Jawab: Hasil yang didapat sudah benar. Kueri yang saya buat bisa mendapatkan data dengan sebuah tabel yang terdiri dari 3 kolom dan 832 baris.

Soal 23 – Perhatikan nilai pada kolom orderid. Apakah terdapat nilai yang hilang (NULL) ? Kenapa ?

Jawab: Dari data yang ada, tidak ditemukan data yang NULL sama sekali. Jika menggunakan perintah outer join sebenarnya bisa saja dimungkinkan data yang NULL ditampilkan karena salah satu data tidak mempunyai pasangan. Jika tidak ditemukan NULL sama sekali, maka kedua tabel tersebut sudah memiliki pasangannya masing-masing sehingga tidak ada data NULL yang ditampilkan.

Praktikum Bagian 10: Membuat Query Cross Join

Soal 24 – Jalankan kode T-SQL di bawah task 1. Tampilkan outputnya!

Jawab: Hasilnya sebagai berikut:

	calendardate
1	2024-01-01
2	2024-01-02
3	2024-01-03
4	2024-01-04
5	2024-01-05
6	2024-01-06
7	2024-01-07
8	2024-01-08
9	2024-01-09
10	2024-01-10
11	2024-01-11
12	2024-01-12
13	2024-01-13
14	2024-01-14
15	2024-01-15
16	2024-01-16
17	2024-01-17
18	2024-01-18
19	2024-01-19
20	2024-01-20

Soal 25 - Tuliskan perintah SELECT untuk mengambil nilai dari kolom empid, firstname, and lastname dari tabel HR.Employees dan kolom calendardate dari tabel HR.Calendar!

Jawab: Kueri dan hasil yang didapat sebagai berikut:

```
SELECT
    e.empid,
    e.firstname,
    e.lastname,
    c.calendardate
FROM HR.Employees e
CROSS JOIN HR.Calendar c;
```

	empid	firstname	lastname	calendardate
1	1	Sara	Davis	2024-01-01
2	1	Sara	Davis	2024-01-02
3	1	Sara	Davis	2024-01-03
4	1	Sara	Davis	2024-01-04
5	1	Sara	Davis	2024-01-05
6	1	Sara	Davis	2024-01-06
7	1	Sara	Davis	2024-01-07
8	1	Sara	Davis	2024-01-08
9	1	Sara	Davis	2024-01-09
10	1	Sara	Davis	2024-01-10
11	1	Sara	Davis	2024-01-11

Kueri di atas menggunakan klausa CROSS JOIN yang digunakan untuk saling memasangkan data dari tabel HR.Employees dengan data di HR.Calendar. Tabel tersebut akan menghasilkan 9 (jumlah keseluruhan pegawai) x 366 (jumlah keseluruhan hari dalam setahun di 2024) data.

Soal 26 – Eksekusi uji coba tahap ke-3 dan bandingkan dengan file 92 – Lab Exercise 5 – Task 2 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar!

Jawab: Hasil yang didapat sudah sesuai. Tabel yang didapat terdiri dari 4 kolom dan 3294 baris.

Praktikum Bagian 11: Menuliskan Query Yang Akan Melakukan Filter Data dengan Klausa WHERE

Soal 27 – Eksekusi uji coba tahap ke-2 dan bandingkan dengan file 52 – Lab Exercise 1 – task 1 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar!

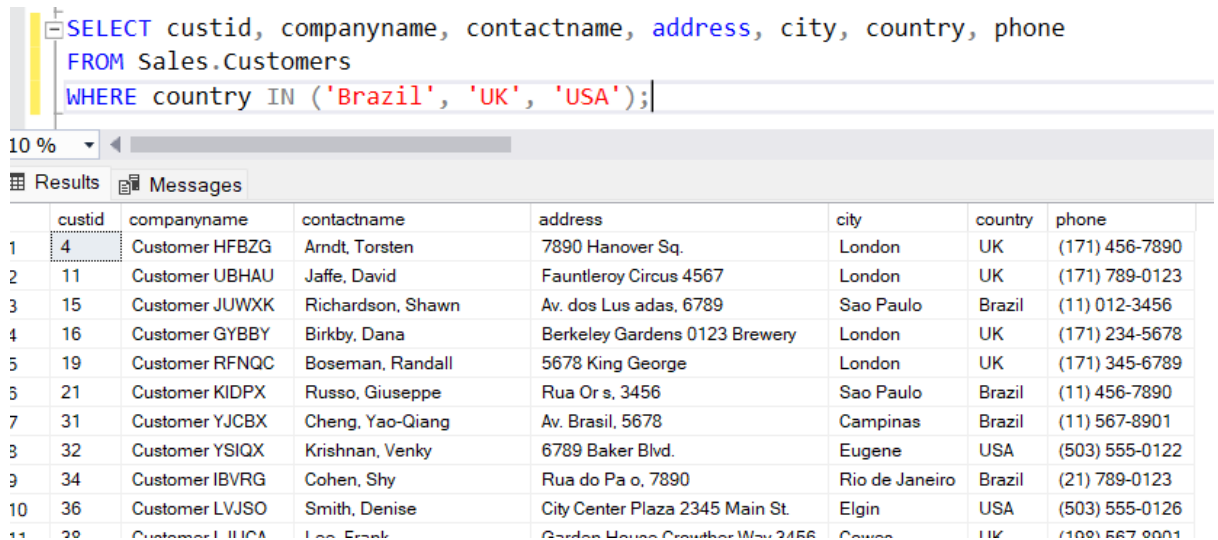
Jawab: Kueri dan hasil yang didapat sudah sesuai dengan file 52 - Lab Exercise 1 - Task 1 Result

```
SELECT custid, companyname, contactname, address, city, country, phone
FROM Sales.Customers
WHERE country = N'Brazil';
```

	custid	companyname	contactname	address	city	country	phone
1	15	Customer JUW XK	Richardson, Shawn	Av. dos Lus adas, 6789	Sao Paulo	Brazil	(11) 012-3456
2	21	Customer KIDPX	Russo, Giuseppe	Rua Ors, 3456	Sao Paulo	Brazil	(11) 456-7890
3	31	Customer YJCBX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
4	34	Customer IBVRG	Cohen, Shy	Rua do Pa o, 7890	Rio de Janeiro	Brazil	(21) 789-0123
5	61	Customer WULWD	Florczyk, Krzysztof	Rua da Panificadora, 1234	Rio de Janeiro	Brazil	(21) 678-9012
6	62	Customer WFIZJ	Misiec, Anna	Alameda dos Can rios, 1234	Sao Paulo	Brazil	(11) 901-2345
7	67	Customer QVEPD	Garden, Euan	Av. Copacabana, 6789	Rio de Janeiro	Brazil	(21) 345-6789
8	81	Customer YQQWW	Nagel, Jean-Philippe	Av. In s de Castro, 1234	Sao Paulo	Brazil	(11) 123-4567
9	88	Customer SRQVM	Li, Yan	Rua do Mercado, 4567	Resende	Brazil	(14) 234-5678

Soal 28 – Tulis perintah SELECT yang akan mengembalikan nilai pada kolom custid, companyname, contactname, address, city, country, and phone pada tabel Sales.Customers, kemudian filter hasilnya hanya untuk “Brazil, UK, dan USA” (Gunakan predikat IN dalam klausa WHERE)!

Jawab: Kueri dan hasil yang didapat:



The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT custid, companyname, contactname, address, city, country, phone FROM Sales.Customers WHERE country IN ('Brazil', 'UK', 'USA');`. The results table has 7 columns: custid, companyname, contactname, address, city, country, and phone. It contains 11 rows of data.

	custid	companyname	contactname	address	city	country	phone
1	4	Customer HFBZG	Arndt, Torsten	7890 Hanover Sq.	London	UK	(171) 456-7890
2	11	Customer UBHAU	Jaffe, David	Fauntleroy Circus 4567	London	UK	(171) 789-0123
3	15	Customer JUWXX	Richardson, Shawn	Av. dos Lus adas, 6789	Sao Paulo	Brazil	(11) 012-3456
4	16	Customer GYBBY	Birkby, Dana	Berkeley Gardens 0123 Brewery	London	UK	(171) 234-5678
5	19	Customer RFNQC	Boseman, Randall	5678 King George	London	UK	(171) 345-6789
6	21	Customer KIDPX	Russo, Giuseppe	Rua Or s, 3456	Sao Paulo	Brazil	(11) 456-7890
7	31	Customer YJCBX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
8	32	Customer YSIQX	Krishnan, Venky	6789 Baker Blvd.	Eugene	USA	(503) 555-0122
9	34	Customer IBVRG	Cohen, Shy	Rua do Pa o, 7890	Rio de Janeiro	Brazil	(21) 789-0123
10	36	Customer LVJSO	Smith, Denise	City Center Plaza 2345 Main St.	Elgin	USA	(503) 555-0126
11	39	Customer LJIUA	Lee, Frank	Garden House Grounds Way 2456	Campan	UK	(100) 567 9001

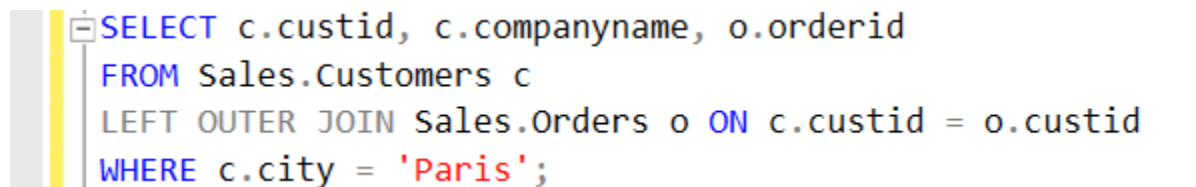
Kueri tersebut hanya akan menampilkan data baris yang memiliki nilai “country” Brazil, UK, atau USA sesuai dengan tanda kurung di dalam IN.

Soal 29 – Eksekusi uji coba tahap ke-3 dan bandingkan dengan file 53 – Lab Exercise Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar.

Jawab: Hasil yang didapat sudah sesuai dengan contoh. Tabel yang dihasilkan berupa 7 kolom dan 29 baris.

Soal 30 – Salin kode T-SQL pada tahap ke-7 kemudian modifikasi dengan operator perbandingan untuk kolom city pada klausa WHERE. Setelah itu eksekusi kode tersebut, tunjukkan hasilnya!

Jawab: Kueri dan hasil yang didapat:



The screenshot shows a SQL query in a text editor: `SELECT c.custid, c.companyname, o.orderid FROM Sales.Customers c LEFT OUTER JOIN Sales.Orders o ON c.custid = o.custid WHERE c.city = 'Paris';`

Results		Messages	
	custid	companyname	orderid
1	57	Customer WVAXS	NULL
2	74	Customer YSHXL	10738
3	74	Customer YSHXL	10907
4	74	Customer YSHXL	10964
5	74	Customer YSHXL	11043

Kueri tersebut akan menampilkan sebuah tabel dengan 3 kolom 5 baris. Selain itu, data yang ada diseleksi hanya data yang memiliki city bernilai “Paris”.

Praktikum Bagian 12: Menuliskan Query yang Akan Mengurutkan (SORT) Data dengan Klausa ORDER BY

Soal 31 – Tuliskan perintah SELECT untuk mengambil kolom custid, custname, dari tabel Sales.Customers dan kolom orderid, orderdate, dari tabel Sales.Orders! Filter hasilnya hanya untuk pesanan pada atau setelah 1 April 2008. Kemudian urutkan hasilnya berdasarkan orderdate secara descending (menurun) dan custid ascending (menaik)!

Jawab: Kueri dan hasil yang didapat sebagai berikut:

```

SELECT
    c.custid,
    c.contactname,
    o.orderid,
    o.orderdate
FROM Sales.Customers c
RIGHT JOIN Sales.Orders o ON c.custid = o.custid
WHERE o.orderdate >= '2008-04-01'
ORDER BY
    o.orderdate DESC,
    c.custid ASC;

```

Results		Messages		
	custid	contactname	orderid	orderdate
1	9	Raghav, Amritansh	11076	2008-05-06 00:00:00.000
2	65	Moore, Michael	11077	2008-05-06 00:00:00.000
3	68	Myrcha, Jacek	11075	2008-05-06 00:00:00.000
4	73	Gonzalez, Nuria	11074	2008-05-06 00:00:00.000
5	20	Kane, John	11072	2008-05-05 00:00:00.000
6	44	Louverdis, George	11070	2008-05-05 00:00:00.000
7	46	Dressler, Marlies	11071	2008-05-05 00:00:00.000
8	58	Fakhouri, Fadi	11073	2008-05-05 00:00:00.000
9	17	Jones, TiAnna	11067	2008-05-04 00:00:00.000
10	62	Misiec, Anna	11068	2008-05-04 00:00:00.000
11	80	Gopakumar, Jena	11069	2008-05-04 00:00:00.000

Arti dari kueri tersebut adalah berusaha untuk mengambil 4 kolom dari dua tabel sekaligus, yaitu dari Sales.Customers dan Sales.Orders. Saya menggunakan perintah RIGHT JOIN terhadap Sales.Orders agar keseluruhan data pada tabel Sales.Orders diambil semuanya baru dipasangkan dengan tabel Sales.Customers. Setelah itu hasilnya akan difilter hanya berdasarkan orderdate lebih dari 1 April 2008. Langkah terakhir adalah berusaha untuk mengurutkan orderdate dari yang paling terbaru terlebih dahulu (DESCENDING) baru kemudian diurutkan berdasarkan customerid secara menaik (ASCENDING).

Soal 32 – Eksekusi uji coba tahap ke-2 dan bandingkan dengan file 62 – Lab Exercise 2 – Task 1 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar.

Jawab: Hasil yang didapat sudah sesuai dengan apa yang ada di file 62 – Lab exercise 2 – Task 1 Result.txt. Bisa dilihat bahwa menampilkan tabel dengan kolom sebanyak 4 dan baris sebanyak 88.

Soal 33 – Eksekusi perintah T-SQL pada tahap 3! Apakah terjadi kesalahan? Apa pesan errornya ? Menurut Anda, apakah penyebabnya?

Jawab: Kesalahan tersebut terjadi karena nama kolom “mgrlastname” belum dikenali di satu pernyataan kueri tersebut. Di SQL, terdapat urutan eksekusi perintah. Jadi, meski sudah ditulis di bagian awal, belum tentu program akan mengeksekusinya terlebih dahulu. Tabel berikut akan menjadi gambaran mengenai urutan eksekusi klausa SQL:

ORDER	CLAUSE	FUNCTION
1	from	Choose and join tables to get base data.
2	where	Filters the base data.
3	group by	Aggregates the base data.
4	having	Filters the aggregated data.
5	select	Returns the final data.
6	order by	Sorts the final data.
7	limit	Limits the returned data to a row count.

(Sumber: sisense.com)

Dari situ terlihat bahwa klausa WHERE akan dieksekusi terlebih dahulu daripada klausa SELECT. Ini jadinya nama alias yang dibuat di dalam klausa SELECT, maka tidak akan dikenali oleh klausa WHERE. Maka dari itu, agar kueri bisa berjalan dengan lancar, nama kolom di dalam klausa WHERE harus menggunakan nama asli.

Soal 34 – Lakukan perubahan perintah T-SQL untuk memperbaiki kesalahan pada uji coba ke-3, kemudian lakukan eksekusi! Bandingkan hasil eksekusi dengan file 63 – Lab Exercise 2 – Task 2 Result.txt. Jika sama, maka hasil uji coba sudah benar..

Jawab: Perubahan yang dilakukan sebagai berikut:

```

SELECT
    e.empid, e.lastname, e.firstname, e.title, e.mgrid,
    m.lastname AS mgrlastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE
    m.lastname = N'Buck';

```

0 %

Results Messages

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
	6	Suurs	Paul	Sales Representative	5	Buck	Sven
:	7	King	Russell	Sales Representative	5	Buck	Sven
:	9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

Menghapus nama kolom lastname yang sebelumnya ditulis berdasarkan nama alias diubah dengan menggunakan nama aslinya.

Soal 35 - Salin perintah T-SQL pada uji coba 4, dan modifikasi sehingga mengashilkan semua karyawan ORDER BY nama depan manajer. Pada awalnya uji coba dengan menggunakan nama asal tabel, kemudian lakukan uji coba menggunakan nama alias tabel! Eksekusi T-SQL tersebut dan bandingkan hasilnya dengan file 64 - Lab Exercise 2 - Task 3 Result.txt. Jika Hasilnya sama, maka uji coba sudah benar.

Jawab: Kueri dan hasil yang didapat sebagai berikut:

```
-- Cara 1
SELECT
    e.empid, e.lastname, e.firstname, e.title, e.mgrid,
    m.lastname AS mgrlastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
LEFT JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE e.mgrid IS NOT NULL
ORDER BY
    m.firstname ASC,
    e.lastname DESC;

-- Cara 2
SELECT
    e.empid, e.lastname, e.firstname, e.title, e.mgrid,
    m.lastname AS mgrlastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
LEFT JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE e.mgrid IS NOT NULL
ORDER BY
    mgrfirstname ASC,
    e.lastname DESC;
```

10 %

Results Messages

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	3	Lew	Judy	Sales Manager	2	Funk	Don
2	5	Buck	Sven	Sales Manager	2	Funk	Don
3	4	Peled	Yael	Sales Representative	3	Lew	Judy
4	8	Cameron	Maria	Sales Representative	3	Lew	Judy
5	2	Funk	Don	Vice President, Sales	1	Davis	Sara
6	6	Suurs	Paul	Sales Representative	5	Buck	Sven
7	7	King	Russell	Sales Representative	5	Buck	Sven
8	9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

Perubahan yang dilakukan dari kueri pada percobaan sebelumnya adalah 1) Mengganti yang semula INNER JOIN menjadi LEFT JOIN agar seluruh karyawan bisa ditampilkan datanya; 2) Mengubah filter menjadi tidak akan menampilkan data yang memiliki mgrid bernilai NULL; 3) Menambahkan klausa ORDER BY untuk kolom mgrfirstname secara ASCENDING dan e.lastname secara DESCENDING.

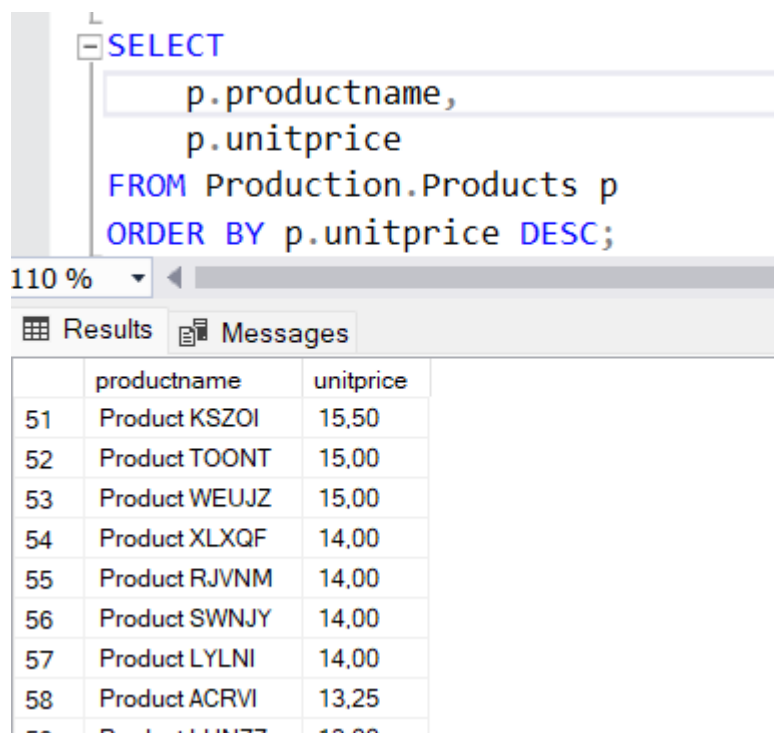
Soal 36 - Kenapa kita dapat menggunakan nama kolom sesuai nama asli tabel ataupun menggunakan nama alias tabel?

Jawab: Ini kembali lagi pada proses urutan eksekusi perintah SQL. Klausa ORDER BY ini cenderung dieksekusi di bagian cukup akhir. Bahkan paling terakhir di antara klausa SELECT dan WHERE. Keadaan ini membuat nama alias yang dibuat di dalam klausa SELECT ataupun WHERE bisa digunakan di dalam klausa ORDER BY karena sebelumnya sudah mengenalinya.

Praktikum Bagian 13: Menuliskan Query yang akan Melakukan Filter Data dengan Klausa TOP

Soal 37 – Tuliskan perintah SELECT untuk menampilkan kolom productname and unitprice pada tabel Production.Products yang diurutkan secara menurun berdasarkan unitprice! Tampilkan hasil eksekusinya!

Jawab: Kueri dan hasil yang didapat sebagai berikut:



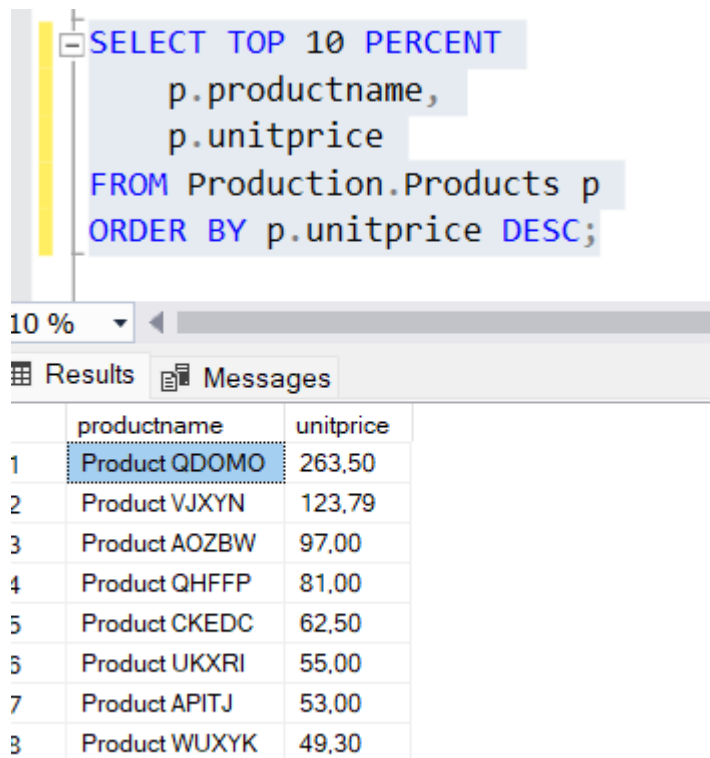
```
SELECT
    p.productname,
    p.unitprice
FROM Production.Products p
ORDER BY p.unitprice DESC;
```

	productname	unitprice
51	Product KSZOI	15,50
52	Product TOONT	15,00
53	Product WEUJZ	15,00
54	Product XLXQF	14,00
55	Product RJVNM	14,00
56	Product SWNJY	14,00
57	Product LYLNI	14,00
58	Product ACRVI	13,25
59	Product LUM77	13,00

Hasil yang didapat adalah data sebanyak 77 baris dengan isi productname dan unitprice.

Soal 38 – Salin dan modifikasi perintah T-SQL pada uji coba 2 dengan batasan hanya 10 persen produk yang akan ditampilkan berdasarkan pemesanan unitprice! Eksekusi perintah tersebut, dan bandingkan apakah sudah sesuai dengan file 73 – Lab Exercise 3 – Task 2 Result.txt

Jawab: Kueri dan hasil yang didapat sebagai berikut:



The screenshot shows a SQL query window with the following text:

```
SELECT TOP 10 PERCENT
    p.productname,
    p.unitprice
FROM Production.Products p
ORDER BY p.unitprice DESC;
```

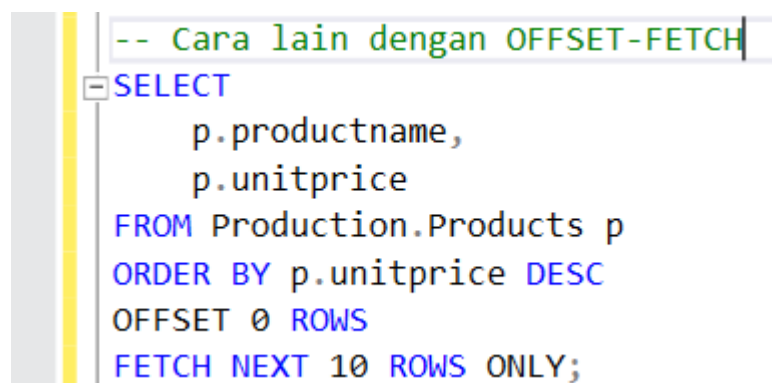
Below the query window, the 'Results' tab is active, displaying a table with 10 rows. The first row is highlighted. The table has two columns: 'productname' and 'unitprice'.

	productname	unitprice
1	Product QDOMO	263,50
2	Product VJXYN	123,79
3	Product AOZBW	97,00
4	Product QHFFP	81,00
5	Product CKEDC	62,50
3	Product UKXRI	55,00
7	Product APITJ	53,00
3	Product WUXYK	49,30

Hanya dengan menambahkan teks “TOP 10 PERCENT” setelah klausa SELECT, maka hanya akan mendapatkan data 10 persen dari keseluruhan data.

Soal 39 – Apakah memungkinkan mengimplementasikan perintah T-SQL uji coba 5 menggunakan klausa OFFSET-FETCH?

Jawab: Mungkin saja. Jika menggunakan klausa OFFSET-FETCH, maka contohnya sebagai berikut:



The screenshot shows a SQL query window with the following text:

```
-- Cara lain dengan OFFSET-FETCH
SELECT
    p.productname,
    p.unitprice
FROM Production.Products p
ORDER BY p.unitprice DESC
OFFSET 0 ROWS
FETCH NEXT 10 ROWS ONLY;
```

Results Messages		
	productname	unitprice
1	Product QDOMO	263,50
2	Product VJXYN	123,79
3	Product AOZBW	97,00
4	Product QHFFP	81,00
5	Product CKEDC	62,50
6	Product UKXRI	55,00
7	Product APITJ	53,00
8	Product WUXYK	49,30
9	Product ZZZHR	46,00
10	Product OFBNT	45,60

Jika menggunakan klausa OFFSET-FETCH, maka caranya mungkin sedikit lebih panjang daripada hanya menggunakan klausa TOP. Jika data yang ingin diambil adalah sebanyak 10 data teratas saja, maka bisa dengan mengatur OFFSET 0 ROWS agar tidak melakukan proses skipping data. Kemudian FETCH NEXT 10 ROWS ONLY agar bisa mendapatkan 10 baris data.

Praktikum Bagian 14: Menuliskan Query yang akan Melakukan Filter Data dengan klausa OFFSET-FETCH

Soal 40 - Tuliskan perintah SELECT untuk menampilkan kolom custid, orderid, and orderdate pada tabel Sales.Orders. Urutkan baris berdasarkan orderdate dan orderid. Ambil 20 baris pertama. Eksekusi perintah tersebut dan bandingkan hasilnya dengan the file 82 - Lab Exercise 4 - Task 1 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar.

Jawab: Kueri dan hasil yang didapat sebagai berikut:

```
SELECT TOP 20
    custid,
    orderid,
    orderdate
FROM Sales.Orders
ORDER BY
    orderdate ASC,
    orderid ASC;
```

Results		Messages	
	custid	orderid	orderdate
1	85	10248	2006-07-04 00:00:00.000
2	79	10249	2006-07-05 00:00:00.000
3	34	10250	2006-07-08 00:00:00.000
4	84	10251	2006-07-08 00:00:00.000
5	76	10252	2006-07-09 00:00:00.000
6	34	10253	2006-07-10 00:00:00.000
7	14	10254	2006-07-11 00:00:00.000
8	68	10255	2006-07-12 00:00:00.000
9	88	10256	2006-07-15 00:00:00.000
10	35	10257	2006-07-16 00:00:00.000
11	20	10258	2006-07-17 00:00:00.000
12	13	10259	2006-07-18 00:00:00.000

Kueri tersebut memanfaatkan klausa TOP agar bisa mendapatkan data sebanyak 20 baris teratas saja. Hasil yang didapat sudah sesuai dengan file Lab Exercise 4 – Task 1 Result.txt.

Soal 41 - Tuliskan perintah SELECT untuk menampilkan hasil yang sama dengan soal no. 43, lewati 20 baris awal, dan lanjutkan dengan 20 baris selanjutnya menggunakan klausa OFFSETFETCH! Eksekusi perintah tersebut dan bandingkan 83 - Lab Exercise 4 - Task 2 Result.txt. Jika hasilnya sama, maka uji coba Anda sudah benar.

Jawab: Kueri dan hasil yang didapat sebagai berikut:

```

SELECT
    custid,
    orderid,
    orderdate
FROM Sales.Orders
ORDER BY
    orderdate ASC,
    orderid ASC
OFFSET 0 ROWS
FETCH NEXT 20 ROWS ONLY;

```

Results		Messages	
	custid	orderid	orderdate
1	85	10248	2006-07-04 00:00:00.000
2	79	10249	2006-07-05 00:00:00.000
3	34	10250	2006-07-08 00:00:00.000
4	84	10251	2006-07-08 00:00:00.000
5	76	10252	2006-07-09 00:00:00.000
6	34	10253	2006-07-10 00:00:00.000
7	14	10254	2006-07-11 00:00:00.000
8	68	10255	2006-07-12 00:00:00.000

Kali ini memanfaatkan klausa OFFSET-FETCH. Saya mengatur OFFSET 0 ROWS agar tidak ada data yang terlewat/skip. Kemudian FETCH NEXT 20 ROWS ONLY untuk mendapatkan data sebanyak 20.