

# LAPORAN PRAKTIKUM BASIS DATA

MINGGU 4



TI-2F

Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

## 1. Menulis Query Menggunakan Klausa GROUP BY

**Soal 1** - Tuliskan T-SQL SELECT yang akan menampilkan kelompok pelanggan yang melakukan pembelian. Klausa SELECT harus mencakup kolom custid dari tabel Sales.Orders dan kolom contactname dari tabel Sales.Customers. Kelompokkan kedua kolom tersebut, dan filter hanya pesanan dari sales employee yang memiliki empid sama dengan 5!

Jawab: Kueri yang digunakan sebagai berikut:

```
SELECT o.custid, c.contactname
FROM Sales.Orders o
INNER JOIN Sales.Customers c ON o.custid = c.custid
WHERE o.empid = 5
GROUP BY o.custid, c.contactname;
```

Dengan begitu maka dihasilkan sebuah output dengan 2 kolom 29 baris data.

| Results |        | Messages            |
|---------|--------|---------------------|
|         | custid | contactname         |
| 1       | 5      | Higginbotham, Tom   |
| 2       | 7      | Bansal, Dushyant    |
| 3       | 9      | Raghav, Amritansh   |
| 4       | 14     | Jelitto, Jacek      |
| 5       | 21     | Russo, Giuseppe     |
| 6       | 24     | San Juan, Patricia  |
| 7       | 25     | Carlson, Jason      |
| 8       | 30     | Shabalin, Rostislav |
| 9       | 34     | Cohen, Shy          |
| 10      | 41     | Litton, Tim         |
| 11      | 46     | Dressler, Marlies   |
| 12      | 47     | Lupu, Cornel        |
| 13      | 50     | Mace, Donald        |
| 14      | 52     | Dupont-Roc, Patrice |
| 15      | 58     | Fakhouri, Fadi      |
| 16      | 60     | Uppal, Sunil        |
| 17      | 62     | Misiec, Anna        |
| 18      | 63     | Veronesi, Giorgio   |
| 19      | 65     | Moore, Michael      |
| 20      | 66     | Voss, Florian       |
| 21      | 67     | Garden, Euan        |
| 22      | 71     | Navarro, Tom s      |
| 23      | 72     | Welcker, Brian      |
| 24      | 74     | O'Brien, Dave       |
| 25      | 76     | Gulbis, Katrin      |
| 26      | 85     | McLin, Nkenge       |
| 27      | 87     | Ludwig, Michael     |
| 28      | 89     | Smith Jr., Ronaldo  |
| 29      | 91     | Conn, Steve         |

Kueri tersebut akan menampilkan data dari kolom custid pada tabel Sales.Orders dan contactname pada tabel Sales.Customers. Agar kedua tabel bisa terhubung, maka bisa menggunakan INNER JOIN yang menghubungkan antara masing-masing kolom custid. Kemudian hasil tersebut akan difilter hanya pada data yang memiliki nilai

empid sama dengan 5. Kemudian untuk mencegah duplikasi data, maka bisa menggunakan klausa GROUP BY untuk menggabungkan data-data custid dan contactname yang sama.

**Soal 2** - Salin T-SQL jawaban soal 1. Kemudian modifikasi untuk menampilkan informasi tambahan kolom city dari tabel Sales.Customers pada klausa SELECT!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 2
SELECT o.custid, c.contactname, c.city
FROM Sales.Orders o
INNER JOIN Sales.Customers c ON o.custid = c.custid
WHERE o.empid = 5
GROUP BY o.custid, c.contactname;
```

Jika dijalankan maka akan muncul error sebagai berikut:

```
Messages
Msg 8120, Level 16, State 1, Line 10
Column 'Sales.Customers.city' is invalid in the select list because it is not contained in either an aggregate function or the
Completion time: 2024-09-28T14:18:33.8112119+07:00
```

**Soal 3** - Apakah terdapat pesan error pada jawaban soal-2? Apakah pesan errornya?

Kenapa pesan itu bisa terjadi?

Jawab: Error tersebut terjadi karena kueri yang dibuat akan berusaha mendapatkan seluruh data kota (city) dari tabel Sales.Customers tanpa ada proses grouping.

Sedangkan data-data lain seperti custid, contactname telah dilakukan proses grouping sehingga tidak ada duplikasi data di sana. Solusi untuk masalah ini adalah membuat kolom city juga dilakukan proses grouping.

**Soal 4** - Perbaiki error yang terjadi pada jawaban soal-2! Jika hasil eksekusinya sama dengan file 53 - Lab Exercise 1 - Task 2 Result.txt, maka T-SQL yang dibuat sudah tepat.!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 2
SELECT o.custid, c.contactname, c.city
FROM Sales.Orders o
INNER JOIN Sales.Customers c ON o.custid = c.custid
WHERE o.empid = 5
GROUP BY o.custid, c.contactname, c.city;
```

Dan akan menghasilkan data sebagai berikut:

|    | custid | contactname         | city            |
|----|--------|---------------------|-----------------|
| 1  | 5      | Higginbotham, Tom   | Lule            |
| 2  | 7      | Bansal, Dushyant    | Strasbourg      |
| 3  | 9      | Raghav, Amritansh   | Marseille       |
| 4  | 14     | Jelitto, Jacek      | Bern            |
| 5  | 21     | Russo, Giuseppe     | Sao Paulo       |
| 6  | 24     | San Juan, Patricia  | Br cke          |
| 7  | 25     | Carlson, Jason      | M nchen         |
| 8  | 30     | Shabalin, Rostislav | Sevilla         |
| 9  | 34     | Cohen, Shy          | Rio de Janeiro  |
| 10 | 41     | Litton, Tim         | Toulouse        |
| 11 | 46     | Dressler, Marlies   | Barquisimeto    |
| 12 | 47     | Lupu, Cornel        | I. de Margarita |
| 13 | 50     | Mace, Donald        | Bruxelles       |
| 14 | 52     | Dupont-Roc, Patrice | Leipzig         |
| 15 | 58     | Fakhouri, Fadi      | M xico D.F.     |
| 16 | 60     | Uppal, Sunil        | Lisboa          |
| 17 | 62     | Misieć, Anna        | Sao Paulo       |

| Results | Messages |
|---------|----------|
|---------|----------|

(29 rows affected)

Completion time: 2024-09-28T14:16:18.1918883+07:00

Kueri akhirnya bisa menampilkan data yang diinginkan setelah kolom city ditambahkan ke dalam klausa SELECT dan GROUP BY.

**Soal 5** - Tuliskan pernyataan SELECT yang akan menampilkan kelompok baris berdasarkan kolom custid dan akan dihitung oleh kolom orderyear mewakili tahun pesanan berdasarkan kolom orderdate dari tabel Sales.Orders. Kemudian filter hasilnya untuk memasukkan hanya pesanan dari karyawan penjualan yang empid nya sama dengan 5!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 5
SELECT custid, YEAR(orderdate) orderyear
FROM Sales.Orders
WHERE empid = 5
GROUP BY custid, YEAR(orderdate);
```

Hasil yang didapat adalah sebagai berikut:

|    | custid | orderyear |
|----|--------|-----------|
| 1  | 5      | 2007      |
| 2  | 5      | 2008      |
| 3  | 7      | 2006      |
| 4  | 9      | 2007      |
| 5  | 14     | 2006      |
| 6  | 21     | 2007      |
| 7  | 24     | 2006      |
| 8  | 25     | 2007      |
| 9  | 30     | 2008      |
| 10 | 34     | 2008      |
| 11 | 41     | 2006      |
| 12 | 46     | 2008      |
| 13 | 47     | 2008      |
| 14 | 50     | 2007      |
| 15 | 52     | 2007      |
| 16 | 58     | 2007      |
| 17 | 60     | 2006      |
| 18 | 60     | 2007      |
| 19 | 62     | 2006      |

| Results  | Messages |
|--|----------|
| (34 rows affected)                                 |          |
| Completion time: 2024-09-28T14:33:34.7435986+07:00 |          |

Kueri tersebut akan mendapatkan data custid dan orderyear dari tabel Sales.Orders. Hasil yang didapat sudah difilter lagi sehingga didapatkan data sejumlah 34 baris. Data tersebut hanya menampilkan seorang customer yang telah melakukan order di satu tahun yang ditentukan. Jika didapatkan seorang customer melakukan banyak order di tanggal berbeda namun masih berada di tahun yang sama, maka itu akan dianggap sebagai satu baris data. Hal ini bisa dilakukan dengan penggunaan GROUP BY pada kolom custid dan kolom orderdate yang sudah dikonversi menjadi YEAR.

**Soal 6** - Tuliskan pernyataan SELECT yang akan mengembalikan kelompok baris berdasarkan kolom categoryname di tabel Production.Categories. Kemudian filter hasilnya hanya untuk product categories yang dipesan pada tahun 2008!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 6
SELECT c.categoryid, categoryname
FROM Production.Categories c
INNER JOIN Production.Products p ON c.categoryid = p.categoryid
INNER JOIN Sales.OrderDetails d ON d.productid = p.productid
INNER JOIN Sales.Orders o ON o.orderid = d.orderid
WHERE YEAR(o.orderdate) = 2008
GROUP BY categoryname, c.categoryid;
```

Dengan begitu maka dihasilkan output tabel berupa:

| Results |            | Messages       |
|---------|------------|----------------|
|         | categoryid | categoryname   |
| 1       | 3          | Confections    |
| 2       | 6          | Meat/Poultry   |
| 3       | 7          | Produce        |
| 4       | 1          | Beverages      |
| 5       | 4          | Dairy Products |
| 6       | 5          | Grains/Cereals |
| 7       | 2          | Condiments     |
| 8       | 8          | Seafood        |

Kueri tersebut akan mendapatkan data-data categoryid dan categoryname dari tabel Production.Categories dengan hanya menampilkan data kategori produk yang ada di order pada tahun 2008. Dikarenakan pada tabel Production.Categories tidak memiliki kolom tahun untuk melakukan filter tahun, maka perlu ditelusuri lebih lanjut mengenai data order itu berasal.

Data order berasal dari tabel Sales.Orders. Untuk menuju ke tabel tersebut, maka perlu menghubungkan-hubungkan tabel Production.Categories ke Sales.Orders. Untuk menyambungkan antar tabel tersebut ternyata dibutuhkan data pada tabel Production.Products dan Sales.OrderDetails. Kemudian antar tabel dihubungkan dengan klausa INNER JOIN.

Setelah semua tersambung kemudian dilakukan filter untuk hanya mendapatkan data pada tahun 2008 dengan memanfaatkan klausa WHERE dan melakukan grouping dengan GROUP BY agar tidak ada data duplikasi. Dengan begitu semua maka bisa didapatkan tabel dengan menampilkan 8 baris data.

## 2. Menulis Query Menggunakan Fungsi Agregasi

**Soal 7 -** Tuliskan pernyataan SELECT yang akan mengembalikan kolom orderid, orderdate dari tabel Sales.Orders dan total sales amount per orderid (Petunjuk : Kalikan kolom qty dan unitprice dari tabel Sales.OderDetails) Gunakan alias salesamount untuk kolom yang dihitung. Kemudian urutkan hasilnya dengan total sales amount dalam urutan menurun!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 7
SELECT o.orderid, o.orderdate, SUM(d.qty * d.unitprice) salesamount
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
GROUP BY o.orderid, o.orderdate
ORDER BY salesamount DESC;
```

Kemudian berikut hasil tabel yang didapatkan:

|    | orderid | orderdate               | salesamount |
|----|---------|-------------------------|-------------|
| 1  | 10865   | 2008-02-02 00:00:00.000 | 17250,00    |
| 2  | 11030   | 2008-04-17 00:00:00.000 | 16321,90    |
| 3  | 10981   | 2008-03-27 00:00:00.000 | 15810,00    |
| 4  | 10372   | 2006-12-04 00:00:00.000 | 12281,20    |
| 5  | 10424   | 2007-01-23 00:00:00.000 | 11493,20    |
| 6  | 10817   | 2008-01-06 00:00:00.000 | 11490,70    |
| 7  | 10889   | 2008-02-16 00:00:00.000 | 11380,00    |
| 8  | 10417   | 2007-01-16 00:00:00.000 | 11283,20    |
| 9  | 10897   | 2008-02-19 00:00:00.000 | 10835,24    |
| 10 | 10353   | 2006-11-13 00:00:00.000 | 10741,60    |
| 11 | 10515   | 2007-04-23 00:00:00.000 | 10588,50    |
| 12 | 10479   | 2007-03-19 00:00:00.000 | 10495,60    |
| 13 | 10540   | 2007-05-19 00:00:00.000 | 10191,70    |
| 14 | 10691   | 2007-10-03 00:00:00.000 | 10164,80    |
| 15 | 11032   | 2008-04-17 00:00:00.000 | 8902,50     |
| 16 | 10816   | 2008-01-06 00:00:00.000 | 8891,00     |
| 17 | 10514   | 2007-04-22 00:00:00.000 | 8623,45     |
| 18 | 10912   | 2008-02-26 00:00:00.000 | 8267,40     |
| 19 | 10360   | 2006-11-22 00:00:00.000 | 7390,20     |
| 20 | 10776   | 2007-12-15 00:00:00.000 | 6984,50     |

| Results | Messages |
|---------|----------|
|---------|----------|

(830 rows affected)

Completion time: 2024-09-28T15:17:20.2557527+07:00

Kueri tersebut memanfaatkan agregasi SUM() yang digunakan untuk menjumlah suatu nilai. Dikarenakan ada fungsi agregasi dan perlu untuk menampilkan data selain hasil agregasi, maka perlu menggunakan klausa GROUP BY. Semua kolom kecuali hasil dari agregasi SUM (salesamount) perlu di grouping. Kemudian langkah terakhir adalah mengurutkan nilai salesamount secara DESCENDING. Nilai yang paling besar akan ditampilkan di posisi paling atas.

**Soal 8** - Salin pernyataan T-SQL pada jawaban soal-7 dan modifikasi dengan memasukkan jumlah order lines untuk setiap order dan nilai rata-rata sales amount per orderid sesuai pesanan. Gunakan nama alias masing- masing nooforderlines dan avgsalesamountperorderlines!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 8
SELECT
    o.orderid,
    o.orderdate,
    SUM(d.qty * d.unitprice) AS salesamount,
    COUNT(d.orderid) AS nooforderlines,
    AVG(d.qty * d.unitprice) AS avgsalesamountperorderline
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
GROUP BY o.orderid, o.orderdate
ORDER BY salesamount DESC;
```

Sehingga didapatkan hasil sebagai berikut:

|   | orderid | orderdate               | salesamount | nooforderlines | avgsalesamountperorderline |
|---|---------|-------------------------|-------------|----------------|----------------------------|
| 1 | 10865   | 2008-02-02 00:00:00.000 | 17250,00    | 2              | 8625,00                    |
| 2 | 11030   | 2008-04-17 00:00:00.000 | 16321,90    | 4              | 4080,475                   |
| 3 | 10981   | 2008-03-27 00:00:00.000 | 15810,00    | 1              | 15810,00                   |
| 4 | 10372   | 2006-12-04 00:00:00.000 | 12281,20    | 4              | 3070,30                    |
| 5 | 10424   | 2007-01-23 00:00:00.000 | 11493,20    | 3              | 3831,0666                  |
| 6 | 10817   | 2008-01-06 00:00:00.000 | 11490,70    | 4              | 2872,675                   |
| 7 | 10889   | 2008-02-16 00:00:00.000 | 11380,00    | 2              | 5690,00                    |
| 8 | 10417   | 2007-01-16 00:00:00.000 | 11283,20    | 4              | 2820,80                    |
| 9 | 10897   | 2008-02-19 00:00:00.000 | 10835,24    | 2              | 5417,62                    |

(830 rows affected)

Completion time: 2024-09-28T16:14:08.7702199+07:00

Dari kueri yang sudah ada sebelumnya, ditambahkan 2 kolom baru yang didapatkan dari proses perhitungan jumlah order sesuai orderid dan juga nilai rata-rata dari tiap-tiap orderid. Untuk menghitung jumlah order sesuai dengan orderid, maka bisa menggunakan fungsi COUNT() yang mana akan menghitung jumlah baris dari kolom tersebut. Kemudian untuk menghitung nilai rata-rata bisa menggunakan fungsi AVG(). Isi di dalam fungsi AVG adalah perhitungan qty dengan unitprice. Ini sama dengan apa yang ada di dalam fungsi SUM().



**Soal 9** - Tuliskan pernyataan SELECT untuk mengambil jumlah penjualan total untuk setiap bulannya! Penggunaan klausa SELECT seharusnya menyertakan perhitungan kolom yearmonthno (notasi YYYYMM) berdasarkan kolom orderdate pada tabel Sales.Orders dan total jumlah penjualan (Perkalian kolom qty dengan unitprice dari tabel Sales.OrderDetils) yang diberi alias saleamountpermonth. Urutan hasilnya didasarkan pada perhitungan kolom yearmonthno.

Jawab: Kueri yang dibuat sebagai berikut:

```
-- 9
SELECT
    FORMAT(o.orderdate, 'yyyyMM') AS yearmonthno,
    SUM(d.qty * d.unitprice) saleamountpermonth
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
GROUP BY FORMAT(o.orderdate, 'yyyyMM')
ORDER BY yearmonthno ASC;
```

Dan hasil yang diperoleh adalah sebagai berikut:

|    | yearmonthno | saleamountpermonth |
|----|-------------|--------------------|
| 1  | 200607      | 30192,10           |
| 2  | 200608      | 26609,40           |
| 3  | 200609      | 27636,00           |
| 4  | 200610      | 41203,60           |
| 5  | 200611      | 49704,00           |
| 6  | 200612      | 50953,40           |
| 7  | 200701      | 66692,80           |
| 8  | 200702      | 41207,20           |
| 9  | 200703      | 39979,90           |
| 10 | 200704      | 55699,39           |
| 11 | 200705      | 56823,70           |

| Results  | Messages |
|--|----------|
| (23 rows affected)                                 |          |
| Completion time: 2024-09-28T16:22:08.7687225+07:00 |          |

Kueri di atas akan melakukan pembuatan sebuah kolom baru bernama yearmonthno dan juga saleamountpermonth. Kolom yearmonthno diisi dengan sebuah angka yang dihasilkan dari gabungan antara tahun dan bulan. Ini bisa dilakukan dengan memanfaatkan fungsi FORMAT(). Kemudian untuk kolom saleamountpermonth

didapatkan dari hasil penjumlahan dari kolom qty dan unitprice. Agar tidak ada duplikasi data dan telah menggunakan fungsi agregasi, maka perlu menggunakan klausa GROUP BY. Langkah terakhir adalah mengurutkan data berdasarkan dari kolom yearmonthno secara ASCENDING.

**Soal 10** - Tulislah perintah SELECT yang akan mengambil semua pelanggan/customer (termasuk yang tidak memiliki pesanan) dan jumlah penjualan, jumlah pesanan maksimum per baris, dan jumlah pesanan! Klausa SELECT harus memasukkan kolom custid dan contactname dari tabel Sales.Customers dan 4 (empat) kolom yang dikalkulasi berdasarkan fungsi agregasi sebagai berikut :

- totalsalesamount, adalah alias untuk jumlah penjualan total per pesanan
- 2) maxsalesamountperorderline, adalah alias untuk jumlah penjualan maksimum per baris pesanan
- 3) numberofrows, adalah alias untuk jumlah baris (gunakan \* dalam fungsi COUNT)
- 4) numberoforderlines, adalah alias untuk jumlah baris pesanan (gunakan kolom orderid di kolom fungsi COUNT)

Urutkan hasilnya berdasarkan kolom totalsalesamount!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 10
SELECT
    c.custid,
    contactname,
    SUM(d.qty * d.unitprice) totalsalesamount,
    MAX(d.unitprice * d.qty) maxsalesamountperorderline,
    COUNT(*) numberofrows,
    COUNT(d.orderid) numberoforderlines
FROM Sales.Customers c
LEFT JOIN Sales.Orders o ON c.custid = o.custid
LEFT JOIN Sales.OrderDetails d ON d.orderid = o.orderid
GROUP BY c.custid, contactname
ORDER BY totalsalesamount ASC;
```

Dan hasil yang didapat sebagai berikut:

Results

Messages

|    | custid | contactname                    | totalsalesamount | maxsalesamountperorderline | numberofrows | numberoforderlines |
|----|--------|--------------------------------|------------------|----------------------------|--------------|--------------------|
| 1  | 22     | Bueno, Janaina Burdan, Neville | NULL             | NULL                       | 1            | 0                  |
| 2  | 57     | Tollefsen, Bj rn               | NULL             | NULL                       | 1            | 0                  |
| 3  | 13     | Benito, Almudena               | 100,80           | 80,00                      | 2            | 2                  |
| 4  | 43     | Deshpande, Anu                 | 357,00           | 210,00                     | 2            | 2                  |
| 5  | 42     | Steiner, Dominik               | 522,50           | 154,00                     | 8            | 8                  |
| 6  | 53     | Mallit, Ken                    | 649,00           | 220,00                     | 6            | 6                  |
| 7  | 29     | Kolesnikova, Katerina          | 836,70           | 186,20                     | 8            | 8                  |
| 8  | 2      | Hassall, Mark                  | 1402,95          | 348,00                     | 10           | 10                 |
| 9  | 69     | Watters, Jason M.              | 1467,29          | 340,00                     | 14           | 14                 |
| 10 | 85     | McLin, Nkenge                  | 1480,00          | 344,00                     | 10           | 10                 |
| 11 | 33     | Sigurdarson, Hallur            | 1488,70          | 990,00                     | 4            | 4                  |
| 12 | 27     | Schm llerl, Martin             | 1545,70          | 530,00                     | 10           | 10                 |
| 13 | 82     | Veninga, Tjeerd                | 1571,20          | 493,00                     | 9            | 9                  |

Results

Messages

Warning: Null value is eliminated by an aggregate or other SET operation.

(91 rows affected)

Completion time: 2024-09-28T16:40:27.3742353+07:00

Kueri yang didapatkan melibatkan 3 kolom, yaitu Sales.Customers, Sales.Orders, dan Sales.OrderDetails. Data-data yang ada pada tabel Sales.Customers harus ditampilkan semuanya, maka dari itu perlu menggunakan LEFT JOIN. Terdapat 4 kolom baru yang perlu ditampilkan, yaitu 1) totalsalesamount yang didapatkan dengan menghitung jumlah penjualan pada tiap-tiap order; 2) maxsalesamountperorderline yang didapatkan dengan mencari nilai tertinggi penjualan dengan memanfaatkan fungsi MAX; 3) numberofrows dengan menghitung jumlah baris; 4) dan numberoforderlines yang didapatkan dengan menghitung berapa kali customer melakukan order dengan memanfaatkan fungsi COUNT().

### 3. Menulis Query Menggunakan Fungsi Agregasi Distinct

**Soal 11** - Berdasarkan hasil eksekusi T-SQL di bawah ini, Kenapa jumlah pesanan (nooforders) sama dengan jumlah pelanggan (noofcustomers)?

Jawab: Itu karena fungsi count menghitung keseluruhan baris tanpa memedulikan bahwa data tersebut sudah dihitung atau belum. Terkadang ada beberapa baris yang memiliki id customer yang sama, maka dari itu jumlah antara nooforders dan noofcustomers sama.

**Soal 12** - Perbaiki T-SQL pada soal-12 untuk menunjukkan jumlah pelanggan yang benar yang melakukan pemesanan setiap tahun!

Jawab: Pembaruan kueri yang digunakan:

```
-- 11
SELECT
    YEAR(orderdate) orderyear,
    COUNT(orderid) nooforders,
    COUNT(DISTINCT custid) noofcustomers
FROM Sales.Orders
GROUP BY YEAR(orderdate)
ORDER BY orderyear ASC;
```

Hasilnya adalah sebagai berikut:

|   | orderyear | nooforders | noofcustomers |
|---|-----------|------------|---------------|
| 1 | 2006      | 152        | 67            |
| 2 | 2007      | 408        | 86            |
| 3 | 2008      | 270        | 81            |

Satu kata yang perlu ditambahkan adalah pemberian kata DISTINCT di dalam fungsi COUNT(). Dengan begitu data yang dihitung untuk noofcustomers adalah data yang benar-benar unik.

**Soal 13** - Tuliskan pernyataan SELECT untuk mengambil jumlah pelanggan berdasarkan huruf pertama dari nilai pada kolom contactname dari tabel Sales.Customers. Tambahkan kolom yang menunjukkan jumlah pesanan yang dilakukan setiap grup pelanggan/customer. Gunakan alias masing-masing firstletter, noofcustomers dan nooforders. Urutkan hasilnya berdasarkan kolom firstletter!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 13
SELECT
    SUBSTRING(contactname, 1, 1) firstletter,
    COUNT(DISTINCT c.custid) noofcustomers,
    COUNT(orderid) nooforders
FROM Sales.Customers c
LEFT JOIN Sales.Orders o ON o.custid = c.custid
GROUP BY SUBSTRING(contactname, 1, 1)
ORDER BY firstletter ASC;
```

Dan dihasilkan data berikut:

|    | firstletter | noofcustomers | nooforders |
|----|-------------|---------------|------------|
| 1  | A           | 2             | 19         |
| 2  | B           | 6             | 37         |
| 3  | C           | 6             | 72         |
| 4  | D           | 5             | 35         |
| 5  | E           | 1             | 10         |
| 6  | F           | 3             | 26         |
| 7  | G           | 6             | 51         |
| 8  | H           | 2             | 22         |
| 9  | I           | 1             | 3          |
| 10 | J           | 3             | 24         |
| 11 | K           | 5             | 54         |
| 12 | L           | 8             | 100        |

| Results  | Messages |
|--|----------|
| <pre>Warning: Null value is eliminated by an aggregate or other SET operation.</pre> |          |
| <pre>(23 rows affected)</pre>  |          |
| <pre>Completion time: 2024-09-28T18:04:19.7164906+07:00</pre>                        |          |

Kueri di atas akan mendapatkan data-data customer beserta jumlah customer dan juga jumlah order. Untuk mendapatkan karakter pertama maka bisa menggunakan fungsi SUBSTRING(). Fungsi tersebut akan mengambil data dari kolom contactname dan hanya mengambil 1 karakter pertamanya saja. Kemudian untuk mendapatkan total customer yang memiliki karakter awal yang sama, maka bisa menggunakan fungsi COUNT dengan menambahkan opsi DISTINCT di dalamnya. Untuk kolom terakhir menghitung orderid bisa menggunakan fungsi COUNT. Dikarenakan data yang ingin didapat membutuhkan dua tabel, maka juga diperlukan klausa JOIN.

**Soal 14** - Salin T-SQL pada jawaban soal-6 kemudian modifikasi dengan memasukkan informasi tentang setiap category produk : jumlah penjualan, jumlah pesanan, dan jumlah penjualan rata-rata setiap pemesanan. Gunakan nama alias masing-masing, nooforders, dan avgsalesamountperorder.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 14
SELECT
    c.categoryid,
    categoryname,
    SUM(d.qty * d.unitprice) totalsalesamount,
    COUNT(DISTINCT o.orderid) nooforders,
    SUM(d.qty * d.unitprice) / COUNT(DISTINCT o.orderid) AS avgsalesamountperorder
FROM Production.Categories c
INNER JOIN Production.Products p ON c.categoryid = p.categoryid
INNER JOIN Sales.OrderDetails d ON d.productid = p.productid
INNER JOIN Sales.Orders o ON o.orderid = d.orderid
WHERE YEAR(o.orderdate) = 2008
GROUP BY categoryname, c.categoryid
ORDER BY c.categoryid ASC;
```

Dan berikut adalah output yang dihasilkan:

|   | categoryid | categoryname   | totalsalesamount | nooforders | avgsalesamountperorder |
|---|------------|----------------|------------------|------------|------------------------|
| 1 | 1          | Beverages      | 122223,75        | 128        | 954,873                |
| 2 | 2          | Condiments     | 34557,45         | 62         | 557,3782               |
| 3 | 3          | Confections    | 58359,73         | 89         | 655,7273               |
| 4 | 4          | Dairy Products | 82803,90         | 90         | 920,0433               |
| 5 | 5          | Grains/Cereals | 30422,25         | 55         | 553,1318               |
| 6 | 6          | Meat/Poultry   | 60275,57         | 43         | 1401,7574              |
| 7 | 7          | Produce        | 32415,85         | 42         | 771,8059               |
| 8 | 8          | Seafood        | 48712,84         | 101        | 482,3053               |

Dari kueri yang ada di nomor 6, terdapat beberapa tambahan. Beberapa di antaranya adalah ada agregasi SUM dan COUNT. SUM bisa digunakan untuk menghitung totalsalesamount. COUNT + DISTINCT bisa digunakan untuk nooforders. Sedangkan untuk menentukan average sale amount per order bisa menggabungkan kedua agregasi sehingga didapatkan nilai yang diharapkan.

#### 4. Menulis Query yang Melakukan Filter Group dengan Klausula HAVING

**Soal 15** - Tuliskan perintah T-SQL dengan klausa SELECT untuk mengambil 5 pelanggan teratas dengan penjualan total lebih dari \$10.000. Tampilkan kolom custid dari tabel order dan hitung kolom yang berisi jumlah penjualan berdasarkan kolom qty dan unitprice dari tabel Sales.OrderDetails. Gunakan alias totalsalesamount.

Jawab: Kueri yang digunakan:

```
-- 15
SELECT TOP 5
    o.custid,
    SUM(d.qty * d.unitprice) totalsalesamount
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
GROUP BY o.custid
HAVING SUM(d.qty * d.unitprice) > 10000
ORDER BY totalsalesamount DESC;
```

Hasil kueri sebagai berikut:

| Results |        | Messages         |
|---------|--------|------------------|
|         | custid | totalsalesamount |
| 1       | 63     | 117483,39        |
| 2       | 71     | 115673,39        |
| 3       | 20     | 113236,68        |
| 4       | 37     | 57317,39         |
| 5       | 65     | 52245,90         |

Untuk mendapatkan data customer dan totalsalesamount dengan nilai tertinggi maka diperlukan klausa HAVING dan TOP. Having digunakan untuk melakukan filter data setelah proses grouping. Fungsi HAVING mirip dengan WHERE. Bedanya hanya proses eksekusi keduanya. Where akan dieksekusi terlebih dahulu sebelum grouping sedangkan having sebaliknya. Untuk melakukan filter lebih lanjut dengan hanya mendapatkan 5 data teratas, maka bisa menggunakan klausa TOP 5.

**Soal 16** - Tuliskan perintah T-SQL dengan klausa SELECT untuk mengambil kolom empid,orderid dan kolom yang mempresentasikan perhitungan total penjualan (total sales amount) berdasarkan tabel Sales.Orders dan Sales.OrderDetails. Filter hasilnya menjadi grup baris data hanya untuk pesanan di tahun 2008!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 16
SELECT
    empid,
    o.orderid,
    SUM(d.qty * d.unitprice) totalsalesamount
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
WHERE YEAR(o.orderdate) = 2008
GROUP BY empid, o.orderid;
```

Dan hasil yang didapat sebagai berikut:

|   | empid | orderid | totalsalesamount |
|---|-------|---------|------------------|
| 1 | 2     | 10808   | 1660,00          |
| 2 | 7     | 10809   | 140,00           |
| 3 | 2     | 10810   | 187,00           |
| 4 | 8     | 10811   | 852,00           |
| 5 | 5     | 10812   | 1852,00          |
| 6 | 1     | 10813   | 648,00           |
| 7 | 3     | 10814   | 2070,00          |
| 8 | 2     | 10815   | 40,00            |

| Results  | Messages |
|--|----------|
| (270 rows affected)                                |          |
| Completion time: 2024-09-28T20:51:16.4074223+07:00 |          |

Kueri di atas akan mendapatkan data kolom empid, orderid, dan totalsalesamount dari dua tabel berbeda, yaitu Sales.Orders dan Sales.OrderDetails. Kemudian untuk menyeleksi data maka diperlukan klausa WHERE yang perlu menyeleksi orderdate pada tahun 2008.

**Soal 17** - Salin perintah T-SQL jawaban soal-16 dan modifikasi untuk menambahkan filter yang hanya mengambil baris yang memiliki jumlah penjualan lebih dari \$10.000!

Jawab: Kueri yang digunakan:

```
-- 17
SELECT
    empid,
    o.orderid,
    SUM(d.qty * d.unitprice) totalsalesamount
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
WHERE YEAR(o.orderdate) = 2008
GROUP BY empid, o.orderid
HAVING SUM(d.qty * d.unitprice) > 10000;
```



Hasil yang didapat:

|   | empid | orderid | totalsalesamount |
|---|-------|---------|------------------|
| 1 | 3     | 10817   | 11490,70         |
| 2 | 2     | 10865   | 17250,00         |
| 3 | 9     | 10889   | 11380,00         |
| 4 | 3     | 10897   | 10835,24         |
| 5 | 1     | 10981   | 15810,00         |
| 6 | 7     | 11030   | 16321,90         |

Pembaruan kueri dari sebelumnya adalah hanya perlu menambahkan klausa HAVING untuk filter data setelah proses grouping dengan hanya mendapatkan data yang memiliki totalsalesamount lebih dari 10000. Alasan menggunakan HAVING untuk proses filter daripada menggunakan WHERE adalah hal yang difilter adalah sesuatu dari hasil agregasi yang hanya bisa dilakukan oleh HAVING.

**Soal 18** - Salin perintah T-SQL jawaban soal-17 dan modifikasi untuk menambahkan filter yang hanya menampilkan pegawai dengan empid sama dengan 3(tiga)!

Jawab: Kueri terbaru yang digunakan:

```
-- 18
SELECT
    empid,
    o.orderid,
    SUM(d.qty * d.unitprice) totalsalesamount
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
WHERE
    YEAR(o.orderdate) = 2008 AND
    empid = 3
GROUP BY empid, o.orderid
HAVING SUM(d.qty * d.unitprice) > 10000;
```

Hasil yang didapat:

|   | empid | orderid | totalsalesamount |
|---|-------|---------|------------------|
| 1 | 3     | 10817   | 11490,70         |
| 2 | 3     | 10897   | 10835,24         |

Kueri terbaru adalah menambahkan filter tambahan berupa empid = 3. Dengan begitu, data yang ditampilkan hanya bisa menampilkan data dengan employee id bernilai 3.

**Soal 19** - Tuliskan perintah T-SQL dengan klausa SELECT untuk mengambil semua pelanggan yang memiliki lebih dari 25 order, dan tambahkan informasi mengenai tanggal pesanan terakhir dan jumlah penjualan. Tampilkan kolom custid dari tabel Sales.Orders table dan dua kolom perhitungan (lastorderdate berdasarkan kolom orderdate dan totalsalesamount berdasarkan kolom qty dan unitprice dari tabel Sales.OrderDetails!

Jawab: Kueri yang digunakan:

```
-- 19
SELECT
    o.custid,
    MAX(o.orderdate) lastorderdate,
    SUM(d.qty * d.unitprice) totalsalesamount
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT o.orderid) > 25
ORDER BY totalsalesamount DESC;
```

Hasil yang didapat:

|   | custid | lastorderdate           | totalsalesamount |
|---|--------|-------------------------|------------------|
| 1 | 63     | 2008-04-14 00:00:00.000 | 117483,39        |
| 2 | 71     | 2008-05-01 00:00:00.000 | 115673,39        |
| 3 | 20     | 2008-05-05 00:00:00.000 | 113236,68        |

Kueri tersebut memanfaatkan 2 agregasi, grouping, dan having clause. Agregasi yang digunakan adalah MAX dan SUM. Max saya gunakan untuk mendapatkan data orderdate paling terbaru dari keseluruhan data order. Agregasi SUM saya gunakan untuk menghitung totalsalesamount. Dikarenakan ada satu kolom yang tidak teragregasi, maka perlu dilakukan grouping. Kemudian akan dilakukan filter untuk mendapatkan data yang memiliki order lebih dari 25. Agar tidak ada duplikasi data yang dihitung, maka perlu klausa DISTINCT.

## 5. Menulis Query Menggunakan Self-Contained Sub-Query

**Soal 20** - Tulislah pernyataan SELECT untuk menampilkan pemesanan orderdate maksimum dari tabel Sales.Orders.

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 20
SELECT MAX(orderdate) lastorderdate
FROM Sales.Orders;
```

Dan hasil yang didapat sebagai berikut:

| Results Messages |                         |
|------------------|-------------------------|
|                  | lastorderdate           |
| 1                | 2008-05-06 00:00:00.000 |

Dengan menggunakan fungsi agregasi MAX() maka bisa didapatkan waktu order paling akhir. Output yang dihasilkan berupa 1x1 data yang berisi waktu terakhir order.

**Soal 21** - Tulislah pernyataan SELECT untuk menampilkan kolom orderid, orderdate, empid, dan custid dari tabel Sales.Orders. Kemudian saring hasilnya dengan menyertakan hanya pesanan yang sesuai dengan waktu pesan paling akhir (Gunakan query pada jawaban soal-20 sebagai sub-query self-contained subquery)!  
Jawab: Kueri yang digunakan sebagai berikut:

```
-- 21
SELECT
    orderid,
    orderdate,
    empid,
    custid
FROM Sales.Orders
WHERE orderdate = (
    SELECT MAX(orderdate) lastorderdate
    FROM Sales.Orders);
```

Hasil yang diperoleh adalah sebagai berikut:

| Results Messages |         |                         |       |        |
|------------------|---------|-------------------------|-------|--------|
|                  | orderid | orderdate               | empid | custid |
| 1                | 11077   | 2008-05-06 00:00:00.000 | 1     | 65     |
| 2                | 11076   | 2008-05-06 00:00:00.000 | 4     | 9      |
| 3                | 11075   | 2008-05-06 00:00:00.000 | 8     | 68     |
| 4                | 11074   | 2008-05-06 00:00:00.000 | 7     | 73     |

Kueri tersebut akan mendapatkan seluruh data pada kolom orderid, orderdate, empid, dan custid dari tabel Sales.Orders. Data-data yang diterima kemudian akan difilter menggunakan klausa WHERE. Di dalam klausa WHERE terdapat SUBQUERY untuk mendapatkan nilai maksimal atau waktu terakhir order. Dengan begitu, dapat ditampilkan seluruh data order yang dilakukan pada waktu terakhir order.

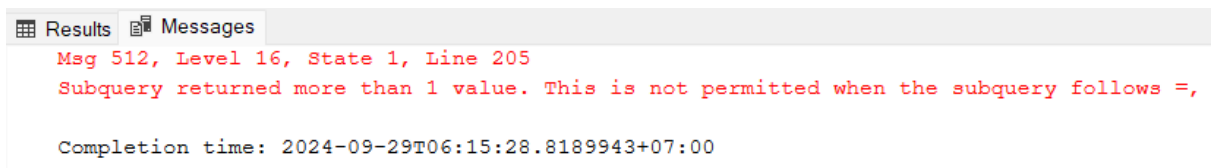
**Soal 22** - Eksekusi T-SQL di bawah ini, kemudian modifikasi dengan filter pelanggan berdasarkan contact name yang diawali dengan huruf B!

Jawab: Kueri yang dilaksanakan:

```
-- 22
SELECT
    orderid,
    orderdate,
    empid,
    custid
FROM Sales.Orders o
WHERE custid =
(
    SELECT custid
    FROM Sales.Customers
    WHERE contactname LIKE 'B%'
);
```

**Soal 23** - Apakah terjadi error pada hasil eksekusi soal-22? Kenapa?

Jawab: Kueri dari soal 22 akan memunculkan error sebagai berikut:

The image shows a screenshot of the SQL Server Enterprise Manager interface. At the top, there are two tabs: 'Results' and 'Messages'. The 'Messages' tab is active, displaying an error message in red text. The message reads: 'Msg 512, Level 16, State 1, Line 205 Subquery returned more than 1 value. This is not permitted when the subquery follows =,'. Below the error message, the completion time is shown as '2024-09-29T06:15:28.8189943+07:00'.

```
Results Messages
Msg 512, Level 16, State 1, Line 205
Subquery returned more than 1 value. This is not permitted when the subquery follows =,
Completion time: 2024-09-29T06:15:28.8189943+07:00
```

Itu terjadi karena hasil yang didapat pada subquery akan menghasilkan banyak nilai sekaligus yang mana itu tidak sesuai dengan operator sama dengan (=).

**Soal 24** - Perbaiki jawaban soal-23 sehingga hasilnya tidak error!

Jawab: Untuk mengatasi error pada soal 23, maka bisa mengganti operator '=' menjadi IN. Kueri lengkapnya sebagai berikut:

```
-- 22
SELECT
    orderid,
    orderdate,
    empid,
    custid
FROM Sales.Orders o
WHERE custid IN
(
    SELECT custid
    FROM Sales.Customers
    WHERE contactname LIKE 'B%'
);
```

Hasil yang didapat sebagai berikut:

|    | orderid | orderdate               | empid | custid |
|----|---------|-------------------------|-------|--------|
| 1  | 10259   | 2006-07-18 00:00:00.000 | 4     | 13     |
| 2  | 10265   | 2006-07-25 00:00:00.000 | 2     | 7      |
| 3  | 10297   | 2006-09-04 00:00:00.000 | 5     | 7      |
| 4  | 10360   | 2006-11-22 00:00:00.000 | 4     | 7      |
| 5  | 10364   | 2006-11-26 00:00:00.000 | 1     | 19     |
| 6  | 10389   | 2006-12-20 00:00:00.000 | 4     | 10     |
| 7  | 10400   | 2007-01-01 00:00:00.000 | 1     | 19     |
| 8  | 10410   | 2007-01-10 00:00:00.000 | 3     | 10     |
| 9  | 10411   | 2007-01-10 00:00:00.000 | 9     | 10     |
| 10 | 10431   | 2007-01-30 00:00:00.000 | 4     | 10     |

Results Messages

(37 rows affected)

Completion time: 2024-09-29T06:19:12.5644052+07:00

Dengan mengganti operator menjadi IN, maka operasi filter IN akan mencoba mencocokkan berbagai nilai yang didapatkan dari proses SUBQUERY. Berbeda dengan operator = yang mencoba mencocokkan nilai secara sama persis sehingga akan muncul error.

**Soal 25** - Tulislah pernyataan SELECT untuk mengambil kolom orderid dari tabel Sales.Orders dan juga kolom hasil perhitungan :

- totalsalesamount (berdasarkan kolom qty dan unitprice dari tabel Sales.OrderDetails)
- salespctoftotal (presentase total jumlah penjualan setiap pesanan dibagi jumlah total penjualan untuk semua pesanan dalam periode tertentu)

Filter hasilnya hanya untuk pemesanan pada bulan mei 2008!

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 25
SELECT
    o.orderid,
    SUM(d.qty * d.unitprice) totalsalesamount,
    SUM(d.qty * d.unitprice) /
    (
        SELECT SUM(qty * unitprice)
        FROM Sales.OrderDetails d
        INNER JOIN Sales.Orders o
            ON d.orderid = o.orderid
        WHERE
            MONTH(o.orderdate) = 5 AND
            YEAR(o.orderdate) = 2008
    ) * 100 salespctoftotal
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d
    ON d.orderid = o.orderid
WHERE
    MONTH(o.orderdate) = 5 AND
    YEAR(o.orderdate) = 2008
GROUP BY o.orderid;
```

Hasil yang didapat sebagai berikut:

| Results |         | Messages         |                 |
|---------|---------|------------------|-----------------|
|         | orderid | totalsalesamount | salespctoftotal |
| 1       | 11064   | 4722,30          | 23,73           |
| 2       | 11065   | 252,56           | 1,26            |
| 3       | 11066   | 928,75           | 4,66            |
| 4       | 11067   | 86,85            | 0,43            |
| 5       | 11068   | 2384,80          | 11,98           |
| 6       | 11069   | 360,00           | 1,80            |
| 7       | 11070   | 1873,50          | 9,41            |
| 8       | 11071   | 510,00           | 2,56            |
| 9       | 11072   | 5218,00          | 26,22           |
| 10      | 11073   | 300,00           | 1,50            |
| 11      | 11074   | 244,30           | 1,22            |
| 12      | 11075   | 586,00           | 2,94            |
| 13      | 11076   | 1057,00          | 5,31            |
| 14      | 11077   | 1374,60          | 6,90            |

Kueri tersebut akan mendapatkan data orderid, totalsalesamonunt, dan persentase penjualan. Data orderid bisa didapatkan langsung dari tabel Sales.Orders. Sedangkan kolom totalsalesamount bisa didapatkan dari fungsi agregasi SUM(). Dan yang terakhir salespctoftotal didapatkan dari perhitungan total penjualan tiap order kemudian di persentasekan dengan memanfaatkan SUBQUERY di dalam klausa SELECT. Isi kueri yang menjadi SUBQUERY adalah perhitungan total penjualan secara keseluruhan pada tahun 2008 bulan Mei.

Hasil dari itu semua akan di filter lagi sesuai tanggal yang ditentukan dan juga dilakukan grouping pada orderid agar tidak ada duplikasi nilai.

## 6. Menulis Query yang Menggunakan Sub-Query Skalar dan Multi Nilai

**Soal 26** - Tulislah pernyataan SELECT untuk mengambil kolom productid dan productname dari tabel Production.Products. Kemudian filter hasilnya untuk menampilkan produk yang terjual dalam jumlah yang banyak (lebih dari 100 produk) untuk baris pesanan tertentu!

Jawab: Kueri yang digunakan sebagai berikut:

```

SELECT
    p.productid,
    p.productname
FROM Production.Products p
INNER JOIN Sales.OrderDetails d ON p.productid = d.productid
INNER JOIN Sales.Orders o ON d.orderid = o.orderid
GROUP BY p.productid, p.productname
HAVING SUM(d.qty) > 1000
ORDER BY SUM(d.qty) DESC;

```

Maka dihasilkan data sebagai berikut:

|    | productid | productname   |
|----|-----------|---------------|
| 1  | 60        | Product WHBYK |
| 2  | 59        | Product UKXRI |
| 3  | 31        | Product XWOXC |
| 4  | 56        | Product VKCMF |
| 5  | 16        | Product PAFRH |
| 6  | 75        | Product BWRLG |
| 7  | 24        | Product QOGNU |
| 8  | 40        | Product YZIXQ |
| 9  | 62        | Product WUXYK |
| 10 | 2         | Product RECZE |
| 11 | 71        | Product MYMOI |
| 12 | 21        | Product VJZZH |

Kueri tersebut akan mendapatkan data utama dari tabel Production.Products. Kemudian akan dilakukan sortir lebih lanjut hanya pada barang yang terjual lebih dari 1000. Operasi sortir bisa dilakukan menggunakan klausa HAVING.

**Soal 27** - Tulislah pernyataan SELECT untuk mengambil kolom custid dan contactname dari tabel Sales.Customers. Kemudian lakukan filter hanya untuk pelanggan/customer yang tidak memiliki pesanan apapun!

Jawab: Kueri yang digunakan sebagai berikut:



```
-- 27
SELECT c.custid, contactname
FROM Sales.Customers c
WHERE NOT EXISTS (
    SELECT * FROM Sales.Orders o
    WHERE o.custid = c.custid);
```

Dan dihasilkan data berikut:

| Results |        | Messages                       |
|---------|--------|--------------------------------|
|         | custid | contactname                    |
| 1       | 22     | Bueno, Janaina Burdan, Neville |
| 2       | 57     | Tollefsen, Bjorn               |

Kueri tersebut memanfaatkan penggunaan klausa EXISTS untuk melakukan filter pada data yang tidak ditemukan di tabel lain. Pada kasus tersebut, kueri di dalam NOT EXISTS akan mencari data yang sesuai antara data di tabel Sales.Customers dan Sales.Orders. Jika data dari Sales.Customers tidak ditemukan dari hasil kueri di dalam klausa NOT EXISTS, maka data tersebut yang menjadi hasil keseluruhan kueri.

**Soal 28** - Terdapat tambahan satu baris data pada tabel Sales.Orders dengan T-SQL sebagai berikut :

```
INSERT INTO Sales.Orders (
custid, empid, orderdate, requireddate, shippeddate, shipperid, freight,
shipname, shipaddress, shipcity, shipregion, shippostalcode, shipcountry)
VALUES
(NULL, 1, '20111231', '20111231', '20111231', 1, 0,
'ShipOne', 'ShipAddress', 'ShipCity', 'RA', '1000', 'USA');
```

Jawab: Data berhasil ditambahkan:

| Messages   |
|--|
| (1 row affected)                                   |
| Completion time: 2024-09-29T07:43:29.2036534+07:00 |

Jika dilakukan kueri yang pernah dilakukan pada soal nomor 27, maka hasilnya sebagai berikut:

| Results |        | Messages                       |
|---------|--------|--------------------------------|
|         | custid | contactname                    |
| 1       | 22     | Bueno, Janaina Burdan, Neville |
| 2       | 57     | Tollefsen, Bj rn               |

Tidak ada perubahan data yang diperoleh meski terdapat penambahan data pada Sales.Orders. Meski saat menambahkan data pada Sales.Orders, data untuk kolom custid bernilai NULL nyatanya tetap tidak mempengaruhi hasil kueri nomor 27. Ini karena pada kueri nomor 27, saya pastikan untuk mengambil data berdasarkan pada tabel Sales.Customers. Jadi, meski terdapat sebuah order yang tidak memiliki pasangan ke customer itu tidak akan didapatkan sebagai hasil kueri karena data NULL tidak ada di tabel Sales.Customers.

**Soal 29** - Modifikasi jawaban soal-27 (cara yang berbeda dengan output yang sama), dengan cara menghapus baris dengan nilai yang tidak diketahui pada kolom custid!  
Jawab: Kueri terbaru dari hasil modifikasi soal 27

```
-- 29
SELECT c.custid, contactname
FROM Sales.Customers c
LEFT JOIN Sales.Orders o ON o.custid = c.custid
WHERE o.orderid IS NULL;
```

Hasilnya sebagai berikut:

| Results |        | Messages                       |
|---------|--------|--------------------------------|
|         | custid | contactname                    |
| 1       | 22     | Bueno, Janaina Burdan, Neville |
| 2       | 57     | Tollefsen, Bj rn               |

Selain menggunakan klausa EXISTS, bisa juga menggunakan JOIN query. Hasil dari JOIN akan difilter dengan menyeleksi bahwa data pada kolom orderid bernilai NULL. Dengan demikian, bisa ditampilkan data yang sama persis seperti pada soal nomor 27.

## 7. Menulis Query yang Menggunakan Sub-Query yang berkorelasi dan Predikat EXISTS

**Soal 30** - Tulislah pernyataan SELECT untuk mengambil kolom custid dan contactname dari tabel Sales.Customers. Tambahkan kolom lastorderdate yang berisi tanggal terakhir dari tabel Sales.Orders untuk setiap pelanggan (Gunakan sub-query yang berkorelasi).

Jawab: Kueri yang dipakai:

```
-- versi 2
SELECT
    c.custid,
    contactname,
    (
        SELECT MAX(orderdate)
        FROM Sales.Orders o
        WHERE o.custid = c.custid
    ) lastorderdate
FROM Sales.Customers c;
```

Hasil yang didapat:

| Results |        | Messages                       |                         |
|---------|--------|--------------------------------|-------------------------|
|         | custid | contactname                    | lastorderdate           |
| 1       | 1      | Allen, Michael                 | 2008-04-09 00:00:00.000 |
| 2       | 2      | Hassall, Mark                  | 2008-03-04 00:00:00.000 |
| 3       | 3      | Peoples, John                  | 2008-01-28 00:00:00.000 |
| 4       | 4      | Arndt, Torsten                 | 2008-04-10 00:00:00.000 |
| 5       | 5      | Higginbotham, Tom              | 2008-03-04 00:00:00.000 |
| 6       | 6      | Poland, Carole                 | 2008-04-29 00:00:00.000 |
| 7       | 7      | Bansal, Dushyant               | 2008-01-12 00:00:00.000 |
| 8       | 8      | Ilyina, Julia                  | 2008-03-24 00:00:00.000 |
| 9       | 9      | Raghav, Amritansh              | 2008-05-06 00:00:00.000 |
| 10      | 10     | Bassols, Pilar Colome          | 2008-04-24 00:00:00.000 |
| 11      | 11     | Jaffe, David                   | 2008-04-14 00:00:00.000 |
| 12      | 12     | Ray, Mike                      | 2008-04-28 00:00:00.000 |
| 13      | 13     | Benito, Almudena               | 2006-07-18 00:00:00.000 |
| 14      | 14     | Jelitto, Jacek                 | 2008-04-22 00:00:00.000 |
| 15      | 15     | Richardson, Shawn              | 2008-04-22 00:00:00.000 |
| 16      | 16     | Birkby, Dana                   | 2008-01-23 00:00:00.000 |
| 17      | 17     | Jones, TiAnna                  | 2008-05-04 00:00:00.000 |
| 18      | 18     | Rizaldy, Arif                  | 2008-02-16 00:00:00.000 |
| 19      | 19     | Boseman, Randall               | 2008-04-28 00:00:00.000 |
| 20      | 20     | Kane, John                     | 2008-05-05 00:00:00.000 |
| 21      | 21     | Russo, Giuseppe                | 2007-10-31 00:00:00.000 |
| 22      | 22     | Bueno, Janaina Burdan, Neville | NULL                    |
| 23      | 23     | Khanna, Karan                  | 2007-12-22 00:00:00.000 |

Results Messages

```
(91 rows affected)
```

Completion time: 2024-09-29T08:28:50.5081674+07:00

Kueri tersebut memanfaatkan subquery untuk mendapatkan data yang ada pada tabel lain. Pada kasus ini, tabel lain yang dimaksud adalah tabel Sales.Orders. Hasil dari subquery nantinya akan menjadi data pada kolom lastorderdate. Perlu juga menggunakan agregasi MAX untuk mendapatkan nilai waktu terbaru dari order customer.

Jika dilihat hasilnya, maka bisa dilihat bahwa hampir semua customer telah melakukan order. Hanya 2 data saja yang tidak pernah melakukan order sama sekali.

**Soal 31** - Tuliskan pernyataan SELECT untuk mengambil semua pelanggan yang tidak memiliki pesanan di tabel Sales.Orders. Gunakan predikat EXISTS untuk melakukan filter yang mencakup pelanggan yang tidak memiliki pesanan! (Tidak diperlukan pemeriksaan eksplisit kolom custid dari tabel Sales.Orders table berstatus not NULL)

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 31
SELECT c.custid, contactname
FROM Sales.Customers c
WHERE NOT EXISTS (
    SELECT * FROM Sales.Orders o
    WHERE o.custid = c.custid);
```

pelanggan yang memesan pada atau setelah 1 april 2008, dan melakukan pemesanan dengan harga tinggi di atas \$100!

Jawab: Kueri yang digunakan:

```
-- 32
SELECT
    c.custid,
    contactname
FROM Sales.Customers c
WHERE EXISTS (
    SELECT 1
    FROM Sales.Orders o
    INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
    WHERE
        o.orderdate >= '2008-04-01' AND
        d.unitprice > 100 AND
        c.custid = o.custid
);
```

Hasil yang didapat:

| Results |        | Messages           |
|---------|--------|--------------------|
|         | custid | contactname        |
| 1       | 24     | San Juan, Patricia |
| 2       | 32     | Krishnan, Venky    |
| 3       | 60     | Uppal, Sunil       |
| 4       | 71     | Navarro, Tom s     |
| 5       | 89     | Smith Jr., Ronaldo |

Kueri tersebut memanfaatkan klausa EXISTS di dalam WHERE untuk memastikan data yang akan diterima nanti sesuai dengan data yang ada di inner query. Inner query berfungsi untuk mendapatkan data order yang terjadi pada atau setelah 1 April 2008 dan memiliki harga unit lebih dari 100. Selain itu, perlu juga dihubungkan antara inner query dan outer query. Ini bisa dilihat pada kondisi WHERE di dalam inner query  $c.custid = o.custid$ .

**Soal 33** - Tulislah pernyataan SELECT yang akan mengambil informasi setiap tahun sebagai berikut :

- Tahun pesanan
- Jumlah total penjualan
- Jumlah total penjualan yang terjual selama bertahun-tahun (setiap tahun dikembalikan jumlah total penjualan sampai tahun tertentu, misalkan awal tahun 2006 mengembalikan jumlah total penjualan untuk tahun selanjutnya 2007)
- Pernyataan SELECT harus memiliki 3 kolom :

- orderyear, berasal dari kolom orderyear dari tabel Sales.Orders
- totalsales, berasal dari kolom qty dan unitprice dari tabel Sales.OrderDetails
- runsales, mewakili jumlah penjualan yang sedang terjadi. Kolom ini menggunakan sub-query yang berkorelasi

Jawab: Kueri yang digunakan sebagai berikut:

```
-- 33
SELECT
    YEAR(o.orderdate) AS orderyear,
    SUM(d.qty * d.unitprice) AS totalsales,
    (SELECT SUM(d1.qty * d1.unitprice)
     FROM Sales.OrderDetails d1
     INNER JOIN Sales.Orders oo ON d1.orderid = oo.orderid
     WHERE YEAR(oo.orderdate) <= YEAR(o.orderdate)
    ) AS runsales
FROM Sales.Orders o
INNER JOIN Sales.OrderDetails d ON o.orderid = d.orderid
GROUP BY YEAR(o.orderdate)
ORDER BY orderyear ASC;
```

Hasil yang diperoleh sebagai berikut:

|   | orderyear | totalsales | runsales   |
|---|-----------|------------|------------|
| 1 | 2006      | 226298,50  | 226298,50  |
| 2 | 2007      | 658388,75  | 884687,25  |
| 3 | 2008      | 469771,34  | 1354458,59 |

Kueri tersebut melakukan kueri SQL dengan cukup kompleks. Diperlukan penggabungan antar tabel. Untuk mendapatkan data total penjualan pada tiap tahun bisa langsung menggunakan agregasi SUM. Kemudian untuk mendapatkan total penjualan secara keseluruhan, maka bisa menggunakan SUBQUERY. Subquery ini berfungsi untuk menghitung total sales pada tahun saat ini dan pada tahun-tahun sebelumnya.