# LAPORAN PRAKTIKUM BASIS DATA LANJUT

# JOBSHEET 13: PEMROGRAMAN T-SQL & ERROR HANDLING


Disusun Oleh:

MUHAMMAD ULIL FAHMI MA'RIFATULLOH (2341760194)

KELAS SIB-2F/20

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

JURUSAN TEKNOLOGI INFORMASI

**POLITEKNIK NEGERI MALANG**

Jl. Soekarno Hatta No. 9, Jattimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

Jurusan Teknologi Informasi Politeknik Negeri Malang
**Jobsheet Minggu ke-13: Pemrograman T-SQL & Error Handling Mata Kuliah BASIS DATA LANJUT**
Pengampu: Tim Ajar Basis Data

**Nama : Muhammad Ulil Fahmi Ma'rifatulloh**
**Kelas : SIB 2F**
**Absen : 20**
**NIM : 2341760194**

**Topik**

1. Menampilkan data dengan menggunakan perfomansi query
2. Pemrograman T-SQL
3. Error Handling

**Tujuan**

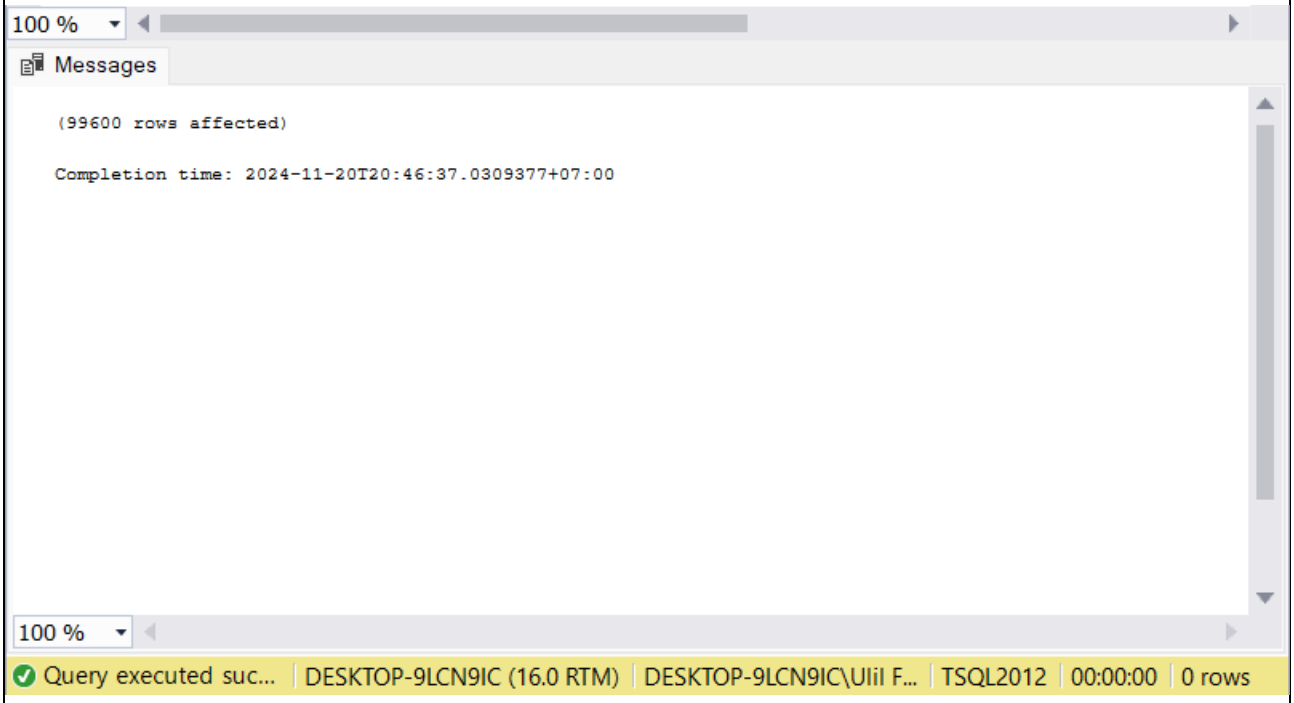
1. Memahami komponen dari query dengan performa yang baik
2. Memahami indexes dan statistics pada SQL Server
3. Mahasiswa memahami cara menggunakan elemen bahasa T-SQL dalam pemrograman dasar.
4. Mahasiswa memahami tentang batches dan bagaimana penanganannya dalam SQL Server.
5. Mahasiswa memahami cara mendeklarasikan & menugaskan variabel dan sinonim.
6. Mahasiswa memahami cara menggunakan blok IF dan WHILE dalam flow program.
7. Mahasiswa memahami bagaimana SQL Server menangani error yang muncul di kode T-SQL.
8. Mahasiswa memahami cara mengimplementasikan penanganan exception yang terstruktur dalam T-SQL.
9. Mahasiswa memahami cara mendapatkan informasi tentang error dari system objects.

**Petunjuk Umum**

1. Ikuti langkah-langkah pada bagian-bagian praktikum sesuai dengan urutan yang diberikan.
2. Anda dapat menggunakan SQL Server 2012 Standard Edition untuk mencoba praktikum pada jobsheet ini. Sesuaikan dengan kondisi komputer Anda.
3. Jawablah semua pertanyaan bertanda [Soal-X] yang terdapat pada langkah-langkah tertentu di setiap bagian praktikum.
4. Dalam setiap langkah pada praktikum terdapat penjelasan yang akan membantu Anda dalam menjawab pertanyaan-pertanyaan pada petunjuk nomor 3, maka baca dan kerjakanlah semua bagian praktikum dalam jobsheet ini.
5. Tulis jawaban dari soal-soal pada petunjuk nomor 3 pada sebuah laporan yang dikerjakan menggunakan aplikasi word processing (Word, OpenOffice, atau yang lain yang sejenis). Ekspor sebagai file **PDF** dengan format nama sebagai berikut:
   - **BDL_Tugas13_Kelas_2DigitNomorAbsen_NamaLengkapAnda**.pdf
   - Contoh:
     - **BDL_Tugas13_TI2Q_99_Suneo**.pdf
   - Perhatikan baik-baik format penamaannya.
   - Kumpulkan file PDF tersebut sebagai laporan praktikum kepada dosen pengampu.
   - Selain pada nama file, cantumkan juga identitas Anda pada halaman pertama laporan tersebut.

**TOPIK 13.1 – QUERY PERFORMANCE**

**Praktikum – Bagian 1: Percobaan Viewing Query Execution Plans**

| Langkah | Keterangan |
|---|---|
| **1** | **Pada bagian ini Anda akan melakukan percobaan untuk membuat dan menambahkan isi tabel Sales.TempOrders**<br><br>Untuk menjalankan instruksi-instruksi dibawah ini, pastikan anda tersambung dengan database TSQL2012 |
| **2** | Departemen IT akan membuat dan mengisi tabel sample bernama Sales.TempOrders dengan perintah berikut ini :<br><br>IF OBJECT_ID('Sales.TempOrders') IS NOT NULL DROP TABLE Sales.TempOrders; SELECT<br>orderid, custid, empid, orderdate, requireddate, shippeddate, shipperid, freight,<br>shipname, shipaddress, shipcity, shipregion, shippostalcode, shipcountry INTO Sales.TempOrders FROM<br>Sales.Orders AS o<br>CROSS JOIN dbo.Nums AS n WHERE<br>n.n <= 120; |
| **3** | [Soal-1] Jalankan dan catat hasilnya!<br>  <br> |
| **4** | [Soal-2] Buat perintah TSQL untuk mengembalikan kolom ordered, custid, dan orderdate dari table Sales.TempOrders<br>• Query<br>`SELECT orderid, custid, orderdate`<br>`FROM Sales.TempOrders;`<br>• Hasil Query |

Results | Messages

| | orderid | custid | orderdate |
|---|---|---|---|
| 1 | 10248 | 85 | 2006-07-04 00:00:00.000 |
| 2 | 10249 | 79 | 2006-07-05 00:00:00.000 |
| 3 | 10250 | 34 | 2006-07-08 00:00:00.000 |
| 4 | 10251 | 84 | 2006-07-08 00:00:00.000 |
| 5 | 10252 | 76 | 2006-07-09 00:00:00.000 |
| 6 | 10253 | 34 | 2006-07-10 00:00:00.000 |
| 7 | 10254 | 14 | 2006-07-11 00:00:00.000 |
| 8 | 10255 | 68 | 2006-07-12 00:00:00.000 |
| 9 | 10256 | 88 | 2006-07-15 00:00:00.000 |
| 10 | 10257 | 35 | 2006-07-16 00:00:00.000 |
| 11 | 10258 | 20 | 2006-07-17 00:00:00.000 |
| 12 | 10259 | 13 | 2006-07-18 00:00:00.000 |
| 13 | 10260 | 56 | 2006-07-19 00:00:00.000 |
| 14 | 10261 | 61 | 2006-07-19 00:00:00.000 |
| 15 | 10262 | 65 | 2006-07-22 00:00:00.000 |
| 16 | 10263 | 20 | 2006-07-23 00:00:00.000 |

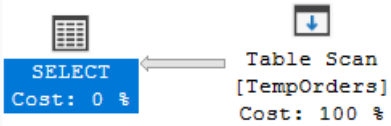
Query execute... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 99,600 rows

**5**

Highlight dari perintah nomor 4 dan tunjukkan perkiraan execution plan. Amati elemen-elemen yang ditampilkan.

Messages | Execution plan

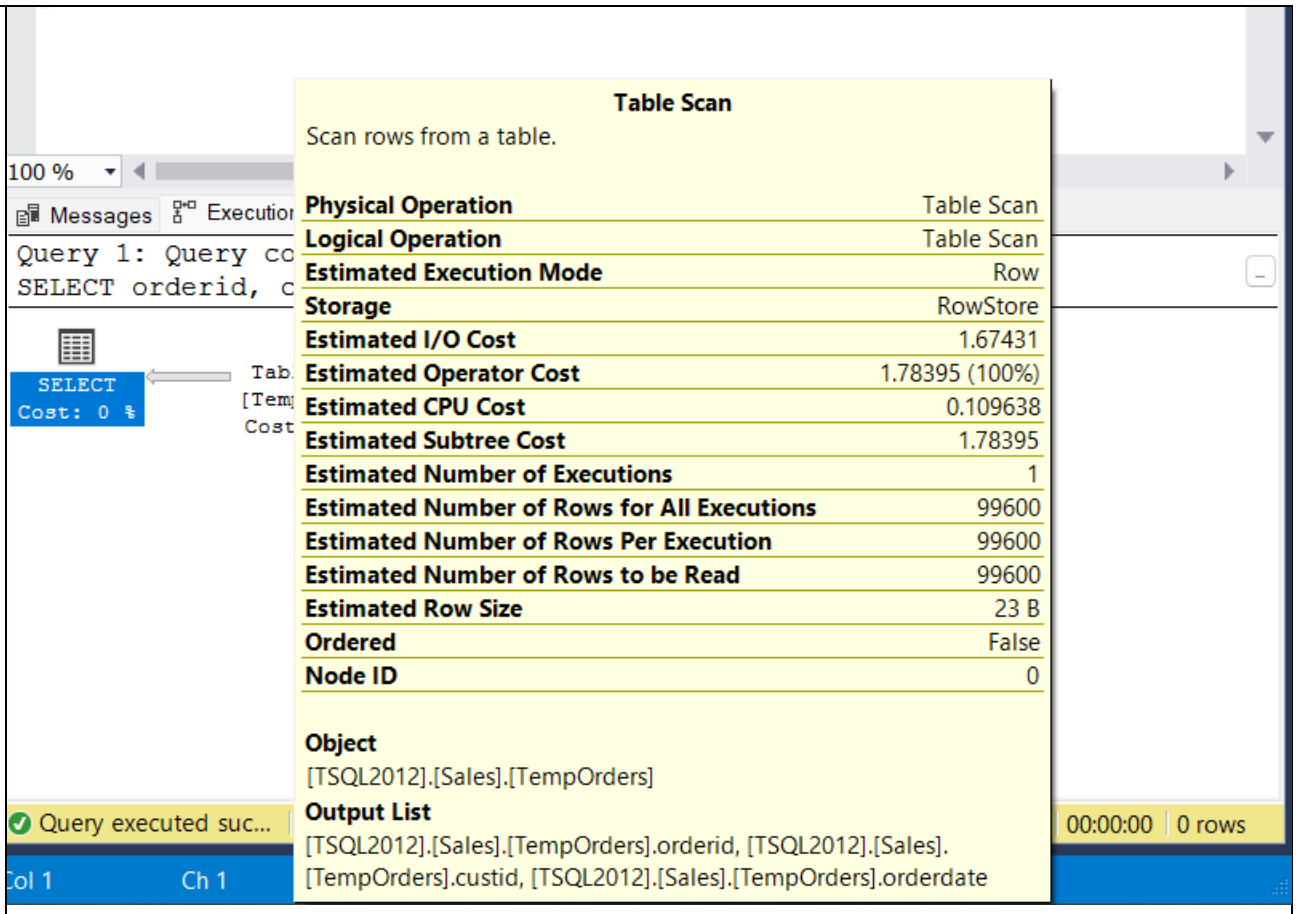
```
Query 1: Query cost (relative to the batch): 100%
SELECT orderid, custid, orderdate FROM Sales.TempOrders
```




Query executed suc... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

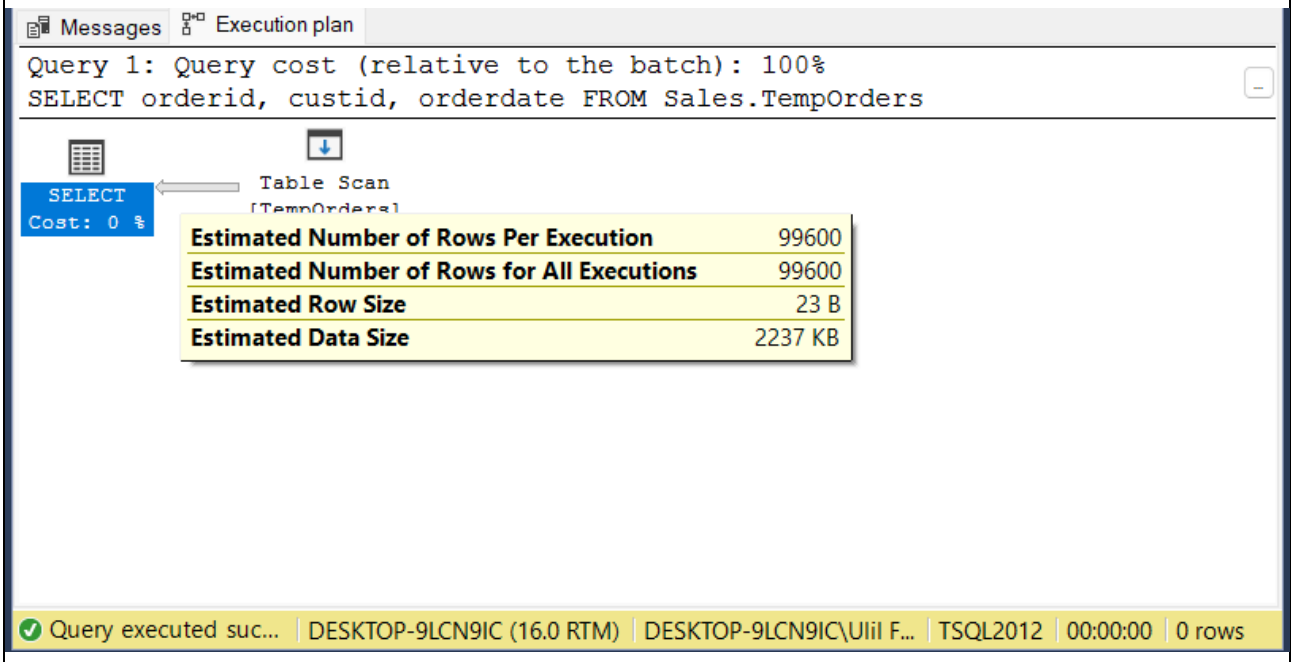
**6**

Arahkan kursor ke Table Scan pada Execution Plans dan lihat properti yang ditampilkan di Yellow tooltip box. Perhatikan kata “estimated” di berbagai properti.



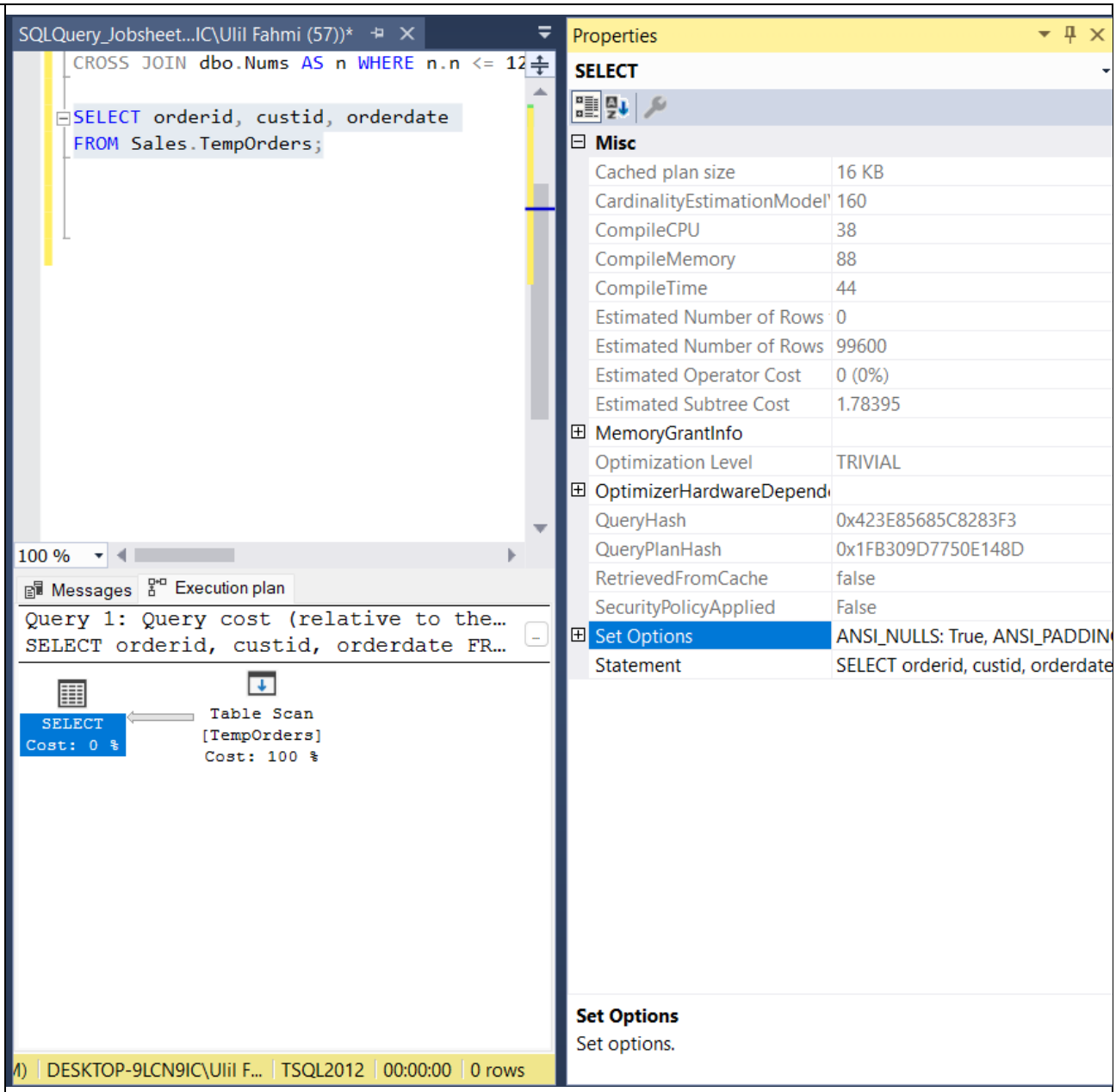

**7**

[Soal-3] Posisikan kursor mouse diatas panah antara operator SELECT dan operator Table Scan dalam execution plans. Properti mana yang akan ditampilkan ?



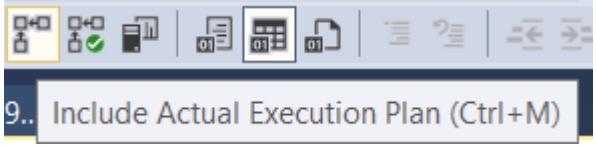

**8**

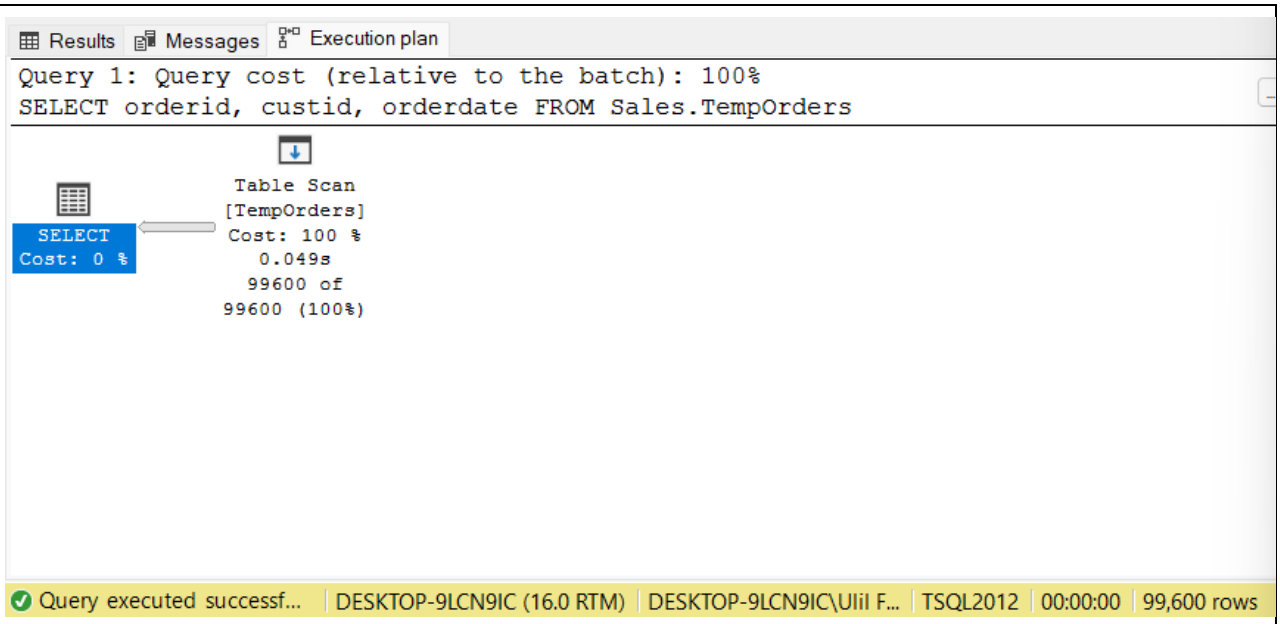
[Soal-4] Tampilkan semua properti operator SELECT dalam execution plans dengan mengklik kanan operator dan memilih properti dari context menu




**9**

[Soal-5] Klik tombol Include Actual Execution Plan pada SQL Editor toolbar (atau tekan ctrl+M pada keyboard) dan execute dengan query SELECT






**10**

Analisa actual execution plan dan jelaskan!

- **Table Scan**: Melakukan scan untuk seluruh tabel "TempOrders" karena tidak ada indeks yang digunakan dalam query ini.
- **Estimated Number of Rows**: Menunjukkan perkiraan jumlah baris yang akan diproses.
- **Actual Number of Rows**: Menampilkan jumlah baris yang sebenarnya diproses.
- **Cost**: Menunjukkan biaya yang digunakan untuk menjalankan operasi.

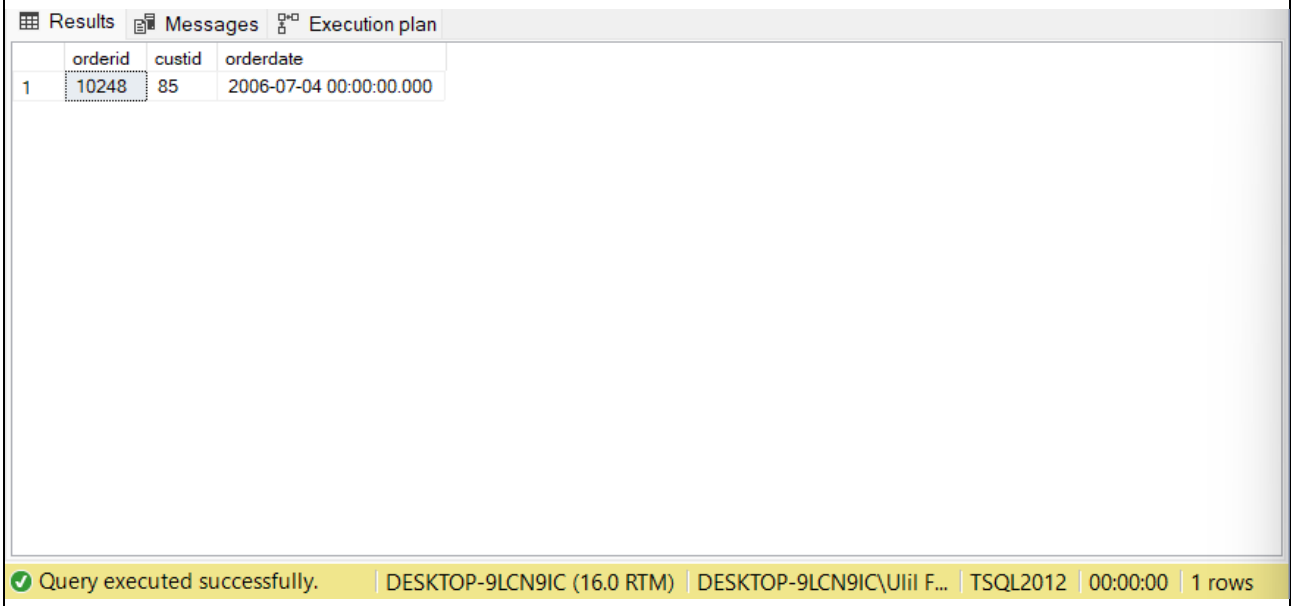
**11**

[Soal-6]Salin perintah SELECT sebelumnya dan lakukan modifikasi untuk mengambil satu baris dengan menggunakan klausa TOP. Tunjukan estimated execution plan

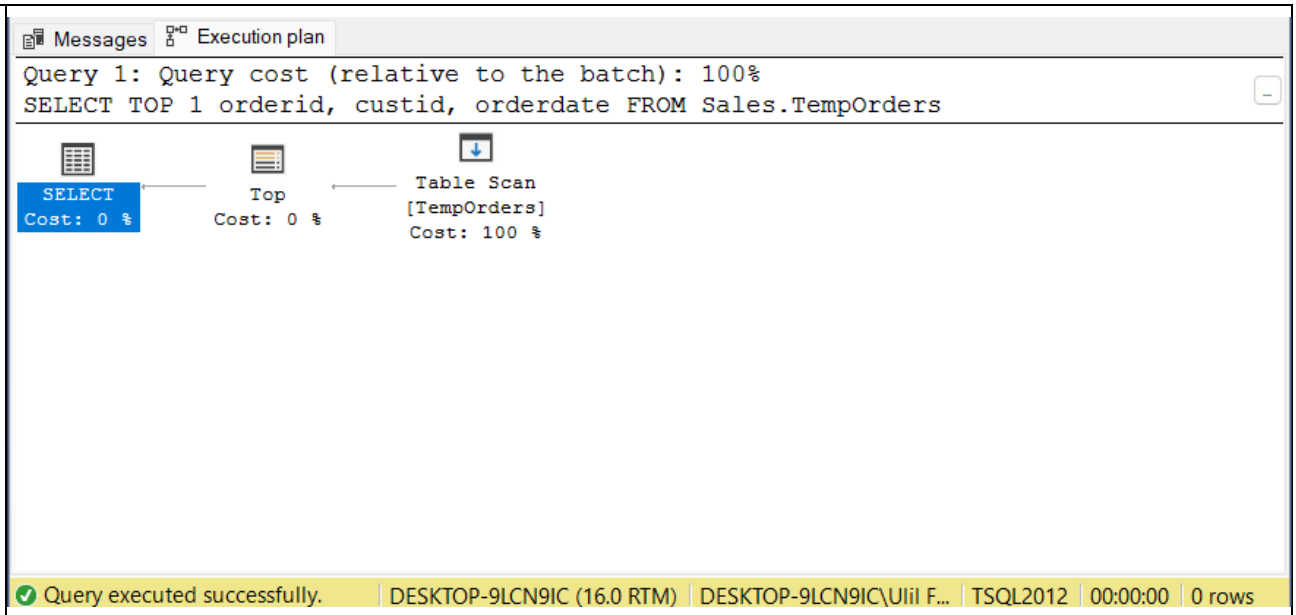
- Query

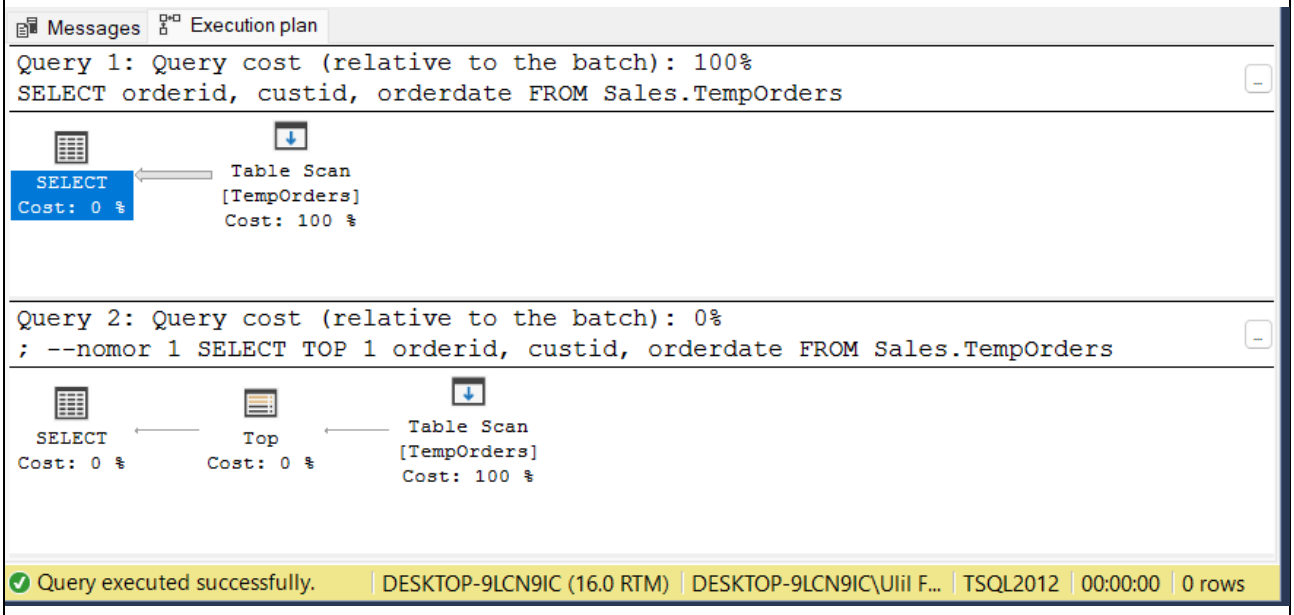
```
SELECT TOP 1 orderid, custid, orderdate
FROM Sales.TempOrders;
```

- Hasil Query

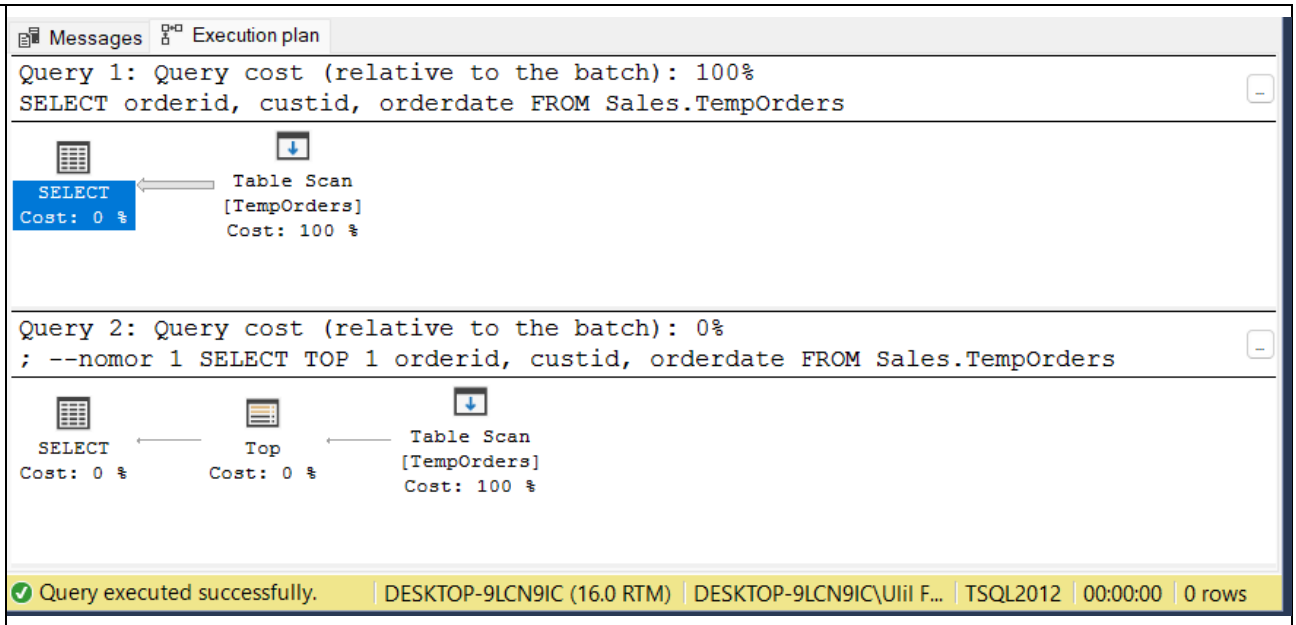



- Estimated execution plan

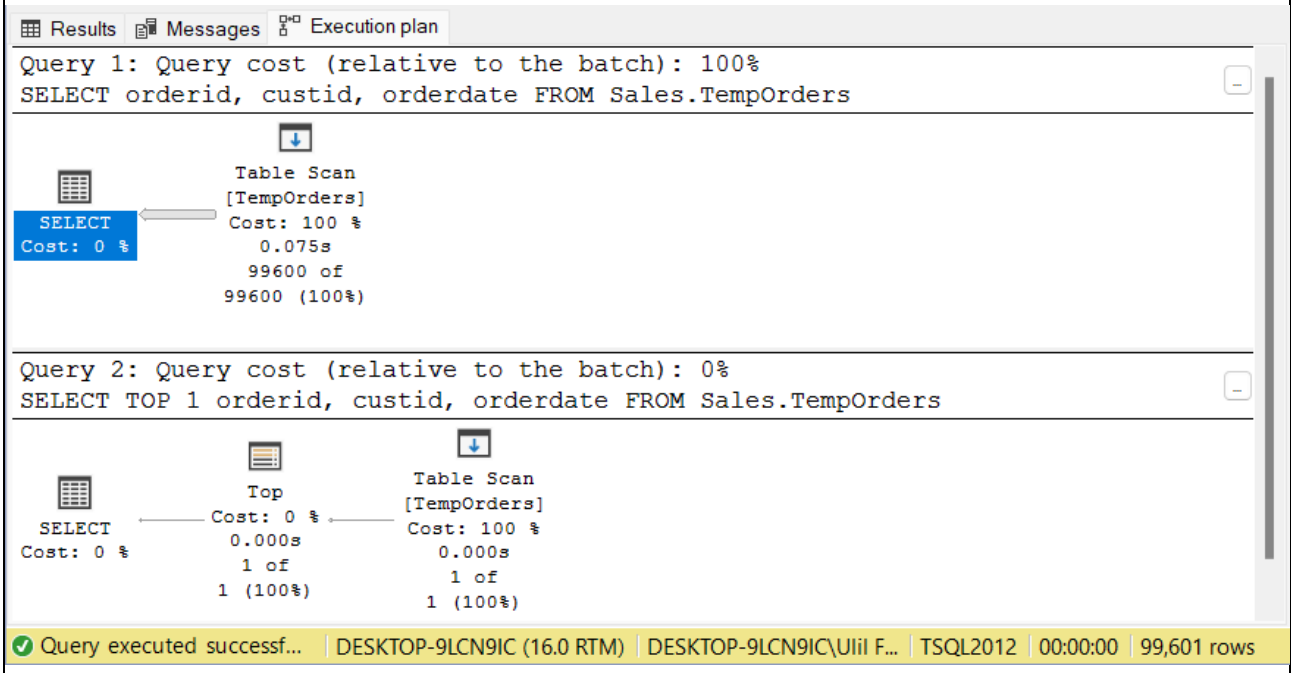
<table>
<tr><td></td><td>


</td></tr>
<tr><td>12</td><td>[Soal-7]Bandingkan execution plan dengan salah satu soal sebelumnya. Operator mana yang baru?<br>• Yang baru adalah operator “TOP”<br>


</td></tr>
<tr><td>13</td><td>Salin perintah SELECT pada no 4</td></tr>
<tr><td>14</td><td>Salin modifikasi SELECT pada no 11, letakkan setelah perintah select yang sudah di salin sebelumnya ( no 13 )</td></tr>
<tr><td>15</td><td>[Soal-8]Tunjukkan estimated execution plan kemudian tunjukkan actual execution plan<br>• Estimated execution plan</td></tr>
</table>




- Actual execution plan




**Setelah melakukan percobaan pada bagian 1, anda telah mampu menampilkan estimated dan actual execution plan**

## Praktikum – Bagian 2: Percobaan Viewing Index Usage dan Penggunaan perintah SET STATISTICS

| Langkah | Keterangan |
|---|---|
| **1** | **Pada modul sebelumnya, anda telah mempelajari bagaimana mengimplementasikan transaction, sekarang anda akan mempelajari tentang bagaimana cara mengaktifkan I/O statistic, menggunakan indexing dan mengimplementasikan perintah SET STATISTIC.**<br><br>Untuk menjalankan instruksi-instruksi dibawah ini, pastikan anda tersambung dengan database TSQL2012 |

| | |
|---|---|
| **2** | Eksekusi perintah dibawah ini<br>CREATE CLUSTERED INDEX CX_Sales_TempOrders_orderdate ON Sales.TempOrders (orderdate ASC); |
| **3** | [Soal-9]Tuliskan perintah SELECT untuk mengambil kolom orderid, cutid, dan orderdate dari tabel Sales.TempOrders. Tampilkan baris dengan tahun pemesanan 2017 dan bulan pemesanan sama dengan 6.<br>• Query<br>`SELECT orderid, custid, orderdate`<br>`FROM Sales.TempOrders`<br>`WHERE YEAR(orderdate) = 2017 AND MONTH(orderdate) = 6;`<br>• Hasil Query<br>Results \| Messages \| Execution plan<br>orderid \| custid \| orderdate<br>Query ex... \| DESKTOP-9LCN9IC (16.0 RTM) \| DESKTOP-9LCN9IC\Ulil F... \| TSQL2012 \| 00:00:00 \| 0 rows |
| **4** | Aktifkan I/O Statistic dengan mengeksekusi perintah SET STATISTIC IO<br>`SET STATISTICS IO ON;` |
| **5** | [Soal-10]Salin query pada [soal 9] kemudian eksekusi. Perhatikan jumlah logical reads yang ditampilkan pada message tab. Angka tersebut berdasarkan I/O Statistic.<br>• Hasil Query |

Results | Messages | Execution plan

```
(0 rows affected)
Table 'TempOrders'. Scan count 1, logical reads 2344, physical reads 0, page server reads 0,
read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0,
lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

(1 row affected)

Completion time: 2024-11-21T22:35:03.7501853+07:00
```

100 %

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

- Execution plan

Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT orderid, custid, orderdate FROM Sales.TempOrders WHERE YEAR...
```

SELECT
Cost: 0 %

Clustered Index Scan (Cluste...
[TempOrders].[CX_Sales_TempO...
Cost: 100 %
0.020s
0 of
1 (0%)

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

**6**

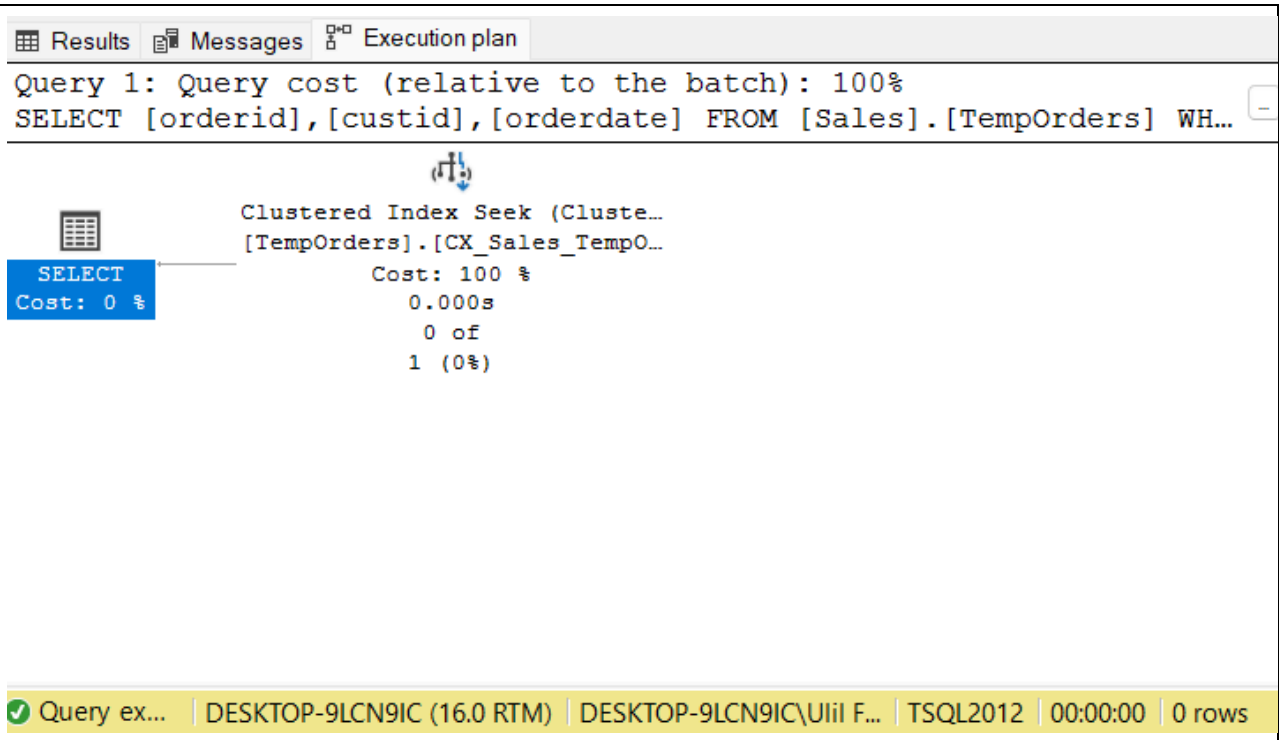
[Soal-11]Salin perintah SELECT pada [soal 9] dan modifikasi dengan mengganti klausa WHERE dengan range berdasarkan kolom orderdate. Tentu saja hasilnya pasti sama. Jalankan kemudian catat hasilnya.

- Query

```
SELECT orderid, custid, orderdate
FROM Sales.TempOrders
WHERE orderdate >= '2017-06-01' AND orderdate < '2017-07-01';
```

- Hasil Query

<table>
<tr>
<td></td>
<td>
Results | Messages | Execution plan<br>
<br>
(0 rows affected)<br>
Table 'TempOrders'. Scan count 1, logical reads 3, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.<br>
<br>
(1 row affected)<br>
<br>
Completion time: 2024-11-21T22:41:52.8592151+07:00<br>
<br>
100 %<br>
Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows<br>
<br>
• Execution plan<br>
<br>
Results | Messages | Execution plan<br>
Query 1: Query cost (relative to the batch): 100%<br>
SELECT [orderid],[custid],[orderdate] FROM [Sales].[TempOrders] WH...<br>
<br>
Clustered Index Seek (Cluste...<br>
[TempOrders].[CX_Sales_TempO...<br>
Cost: 100 %<br>
0.000s<br>
0 of<br>
1 (0%)<br>
<br>
SELECT<br>
Cost: 0 %<br>
<br>
Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows
</td>
</tr>
<tr>
<td>7</td>
<td>
Perhatikan angka pada logical reads pad message tab. Berikan analisa anda!<br>
• Meskipun hasil querynya sama, query pada [Soal-11] menggunakan Clustered Index Seek yang memindai data secara lebih efisien, sehingga jumlah logical reads lebih sedikit(hanya 3). Hal ini terjadi karena indeks memudahkan akses ke data berdasarkan rentang tanggal, berbeda dengan pemindaian seluruh tabel pada query pertama(jumlah logical readsnya 2344).
</td>
</tr>
<tr>
<td>8</td>
<td>Tampilkan query execution plan. Perhatikan operator baru diberi nama Clustered Index Seek.</td>
</tr>
</table>



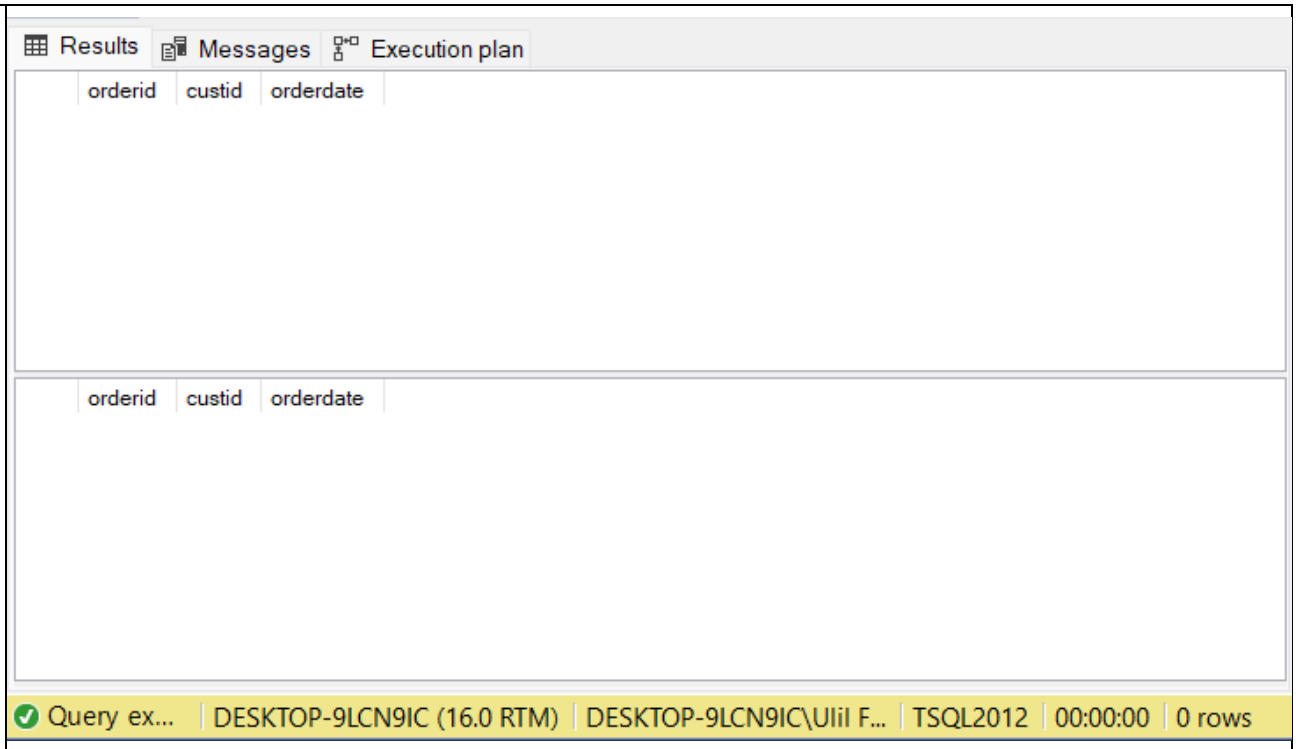

**10**

[Soal-12]Salin Perintah Select pada no 3 kemudian tambahkan SELECT Statement yang ada pada [soal 11]. Tandai kedua perintah tersebut kemudian Jalankan!

- Query

```
-- Query pertama
SELECT orderid, custid, orderdate
FROM Sales.TempOrders
WHERE YEAR(orderdate) = 2017 AND MONTH(orderdate) = 6;

-- Query kedua
SELECT orderid, custid, orderdate
FROM Sales.TempOrders
WHERE orderdate >= '2017-06-01' AND orderdate < '2017-07-01';
```

- Hasil Query

<table>
<tr><td></td><td>


</td></tr>
<tr><td>11</td><td>[Soal-13] Perhatikan perbedaan logical reads. Meskipun hasilnya sama , perintah SELECT pada [ soal 11 ] memindai data 25 kali lebih sedikit dibandingkan dari SELECT yang pertama. Analisa kenapa hal tersebut bisa terjadi !<br>• Meskipun hasil querynya sama, query pada [Soal-11] menggunakan Clustered Index Seek yang memindai data secara lebih efisien, sehingga jumlah logical reads lebih sedikit(hanya 3). Hal ini terjadi karena indeks memudahkan akses ke data berdasarkan rentang tanggal, berbeda dengan pemindaian seluruh tabel pada query pertama(jumlah logical readsnya 2344).</td></tr>
<tr><td>12</td><td>[Soal-14] Kembalikan tabel yang sudah dibuat dan menonaktifkan I/O Statistic dengan menjalankan T-SQL code yang sudah disediakan sebelumnya.<br>• Query<br><pre>DROP INDEX CX_Sales_TempOrders_orderdate ON Sales.TempOrders;<br>SET STATISTICS IO OFF;</pre>• Hasil Query</td></tr>
</table>

| | |
|---|---|
| | Messages<br>Commands completed successfully.<br><br>Completion time: 2024-11-21T23:02:52.0353757+07:00<br><br>100 %<br>Query ex... \| DESKTOP-9LCN9IC (16.0 RTM) \| DESKTOP-9LCN9IC\Ulil F... \| TSQL2012 \| 00:00:00 \| 0 rows |
| **13** | **Setelah melakukan percobaan bagian kedua, anda dapat memahami tentang cara mengatifkan opsi SET STATISTIC dan perlu diingat untuk menginvestasikan waktu dalam memahami index sehingga anda dapat menulis query yang efisien** |

**TOPIK 13.2 – T-SQL PROGAMMING**

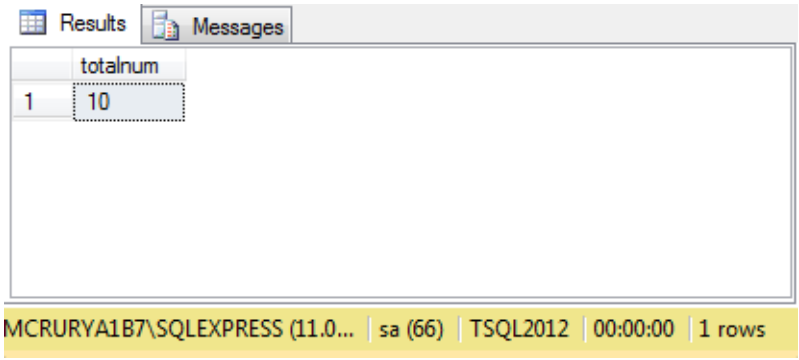
**Praktikum Bagian 3 – DEKLARASI VARIABEL & BATCH: Mendeklarasikan variabel dan mendapatkan nilai variabel**

| Langkah | Keterangan |
|---|---|
| 1 | Pastikan SSMS Anda terkoneksi ke database 'TSQL'. |
| 2 | [Soal-15] Buatlah sebuah kode T-SQL dengan mendeklarasikan sebuah variable bernama @num yang bertipe data integer bernilai 5. Tampilkan nilai variabel tersebut dengan menggunakan alias mynumber lalu eksekusi.<br><br>Hasil yang benar ditunjukkan pada tampilan berikut: xxxx<br><br>Results \| Messages<br>mynumber<br>1 5<br>MCRURYA1B7\SQLEXPRESS (11.0... \| sa (66) \| TSQL2012 \| 00:00:00 \| 1 rows<br><br>• Query<br>`DECLARE @num INT = 5;`<br>`SELECT @num AS mynumber;`<br>• Hasil Query<br>Results \| Messages<br>mynumber<br>1 5<br>Query ex... \| DESKTOP-9LCN9IC (16.0 RTM) \| DESKTOP-9LCN9IC\Ulil F... \| TSQL2012 \| 00:00:00 \| 1 rows |

**3**

[Soal-16] Dari skrip T-SQL **[Soal-1]** di atas, tambahkan batch delimiter (GO) setelahnya, lalu buatlah skrip T-SQL baru yang mendefinisikan 2 variabel bernama @num1 dan @num2 yang sama-sama bertipe data integer. Set nilainya masing-masing 4 dan 6. Tulis sebuah query SELECT yang menampilkan jumlah kedua variable tersebut sebagai totalnum, lalu eksekusi.

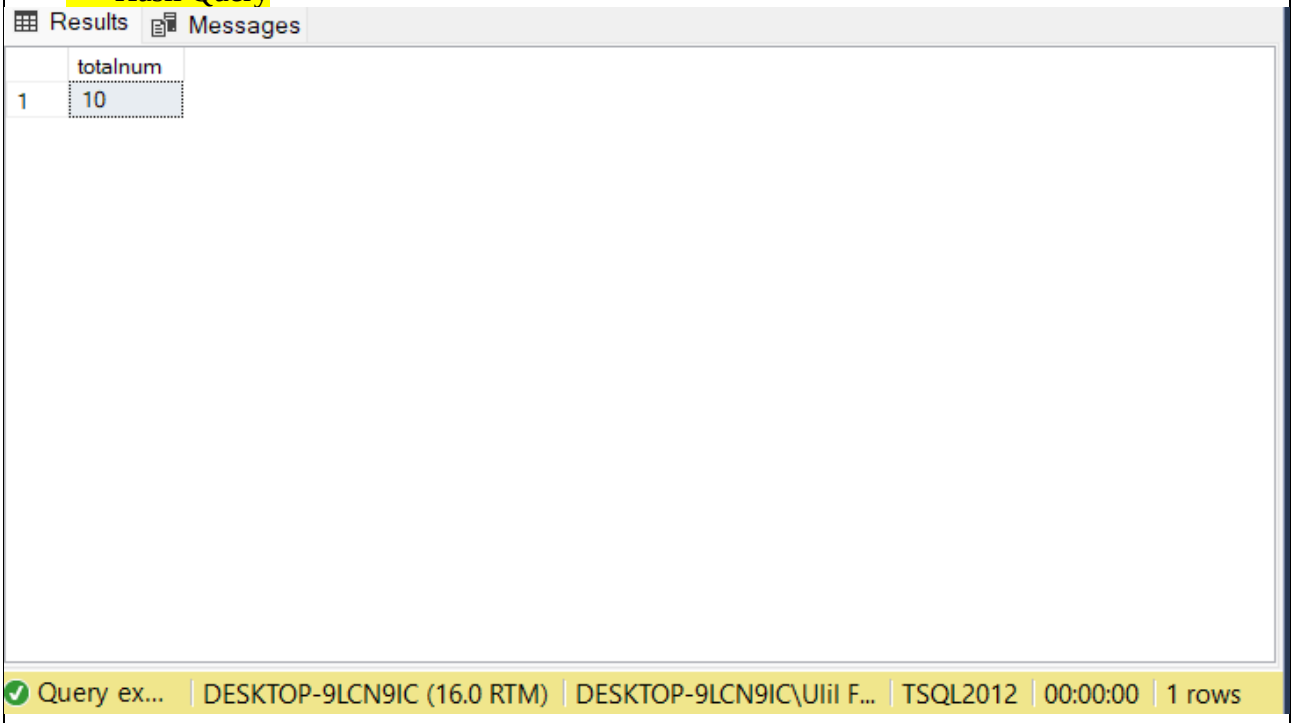
Hasil yang benar ditunjukkan pada tampilan berikut: xxxx



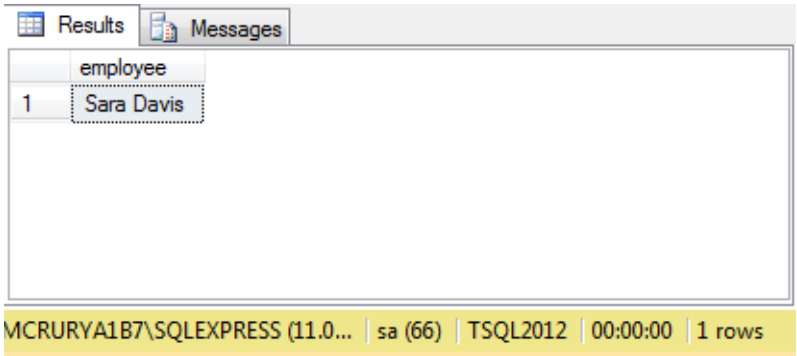

- Query

```
-- Skrip Soal-1
DECLARE @num INT = 5;
SELECT @num AS mynumber;
GO

-- Skrip baru
DECLARE @num1 INT = 4;
DECLARE @num2 INT = 6;
SELECT @num1 + @num2 AS totalnum;
```

- Hasil Query



**Praktikum Bagian 4 – DEKLARASI VARIABEL & BATCH: Memberi nilai terhadap variabel menggunakan query SELECT**

| Langkah | Keterangan |
|---|---|
| 1 | [Soal-17] Buatlah sebuah skrip T-SQL dengan mendefinisikan variabel @empname yang bertipe data nvarchar(30). Selanjutnya, set nilai variabel tersebut sebagai hasil query SELECT terhadap tabel **HR.Employees**, yang menggabungkan kolom *firstname* dan *lastname* dengan dipisahkan spasi, dimana nilai empid-nya sama dengan 1.<br><br>Terakhir, tampilkan nilai variabel @empname dengan menggunakan query SELECT dan beri nama alias sebagai employee. Eksekusi skrip tersebut.<br><br>Hasil yang benar ditunjukkan pada tampilan berikut:<br><br>  <br><br><br>• Query<br>`DECLARE @empname NVARCHAR(30);`<br><br>`-- Set nilai variabel dengan hasil query`<br>`SELECT @empname = firstname + ' ' + lastname`<br>`FROM HR.Employees`<br>`WHERE empid = 1;`<br><br>`-- Tampilkan nilai variabel`<br>`SELECT @empname AS employee;`<br>• Hasil Query |

Results | Messages | Execution plan

| | employee |
|---|---|
| 1 | Sara Davis |

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 1 rows

**2**

[Soal-18] Apakah yang terjadi apabila query SELECT yang dihasilkan lebih dari 1 baris? Lakukan uji coba misalnya dengan menghilangkan filter WHERE "empid = 1".

- Query

```
DECLARE @empname NVARCHAR(30);

-- Set nilai variabel dengan hasil query tanpa filter
SELECT @empname = firstname + ' ' + lastname
FROM HR.Employees;

-- Tampilkan nilai variabel
SELECT @empname AS employee;
```

- Hasil Query

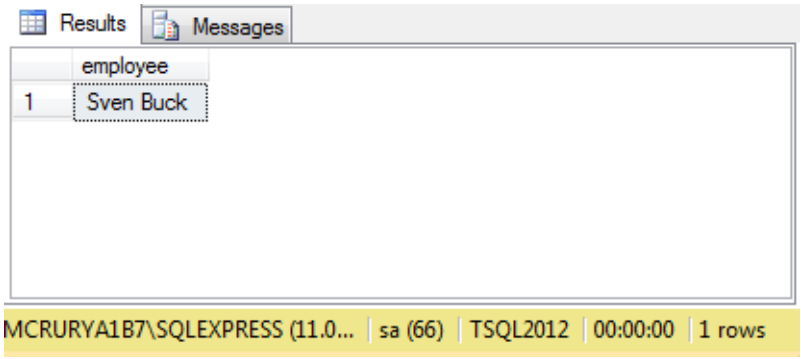
Results | Messages | Execution plan

| | employee |
|---|---|
| 1 | Zoya Dolgopyatova |

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 1 rows

| | |
|---|---|
| | • Analisis<br>○ Jika query di atas menghasilkan lebih dari satu baris, hanya nilai dari baris terakhir yang akan disimpan dalam variabel @empname.<br>○ Disini nilai terakhirnya adalah employee dengan nama "Zoya Dolgopyatova"<br>○ Semua nilai dari baris sebelumnya akan ditimpa oleh nilai dari baris berikutnya sampai baris terakhir. |

**Praktikum Bagian 5 – DEKLARASI VARIABEL & BATCH: Menggunakan sebuah variabel dalam klausa WHERE**

| Langkah | Keterangan |
|---|---|
| 1 | [Soal-19] Salinlah skrip T-SQL dari **[Soal- 3]** di atas dan lakukan modifikasi dengan mendefinisikan sebuah variabel baru bernama @empid yang bertipe data integer bernilai 5. Lalu, gunakan variabel baru ini dalam klausa WHERE sebagai nilai dari kolom *empid*. Eksekusi skrip tersebut.<br><br>Hasil yang benar ditunjukkan pada tampilan berikut:<br>  <br><br>• Query<br>`DECLARE @empid INT = 5;`<br><br>`-- Set nilai variabel dengan hasil query`<br>`DECLARE @empname NVARCHAR(30);`<br>`SELECT @empname = firstname + ' ' + lastname`<br>`FROM HR.Employees`<br>`WHERE empid = @empid;`<br><br>`-- Tampilkan nilai variabel`<br>`SELECT @empname AS employee;`<br>• Hasil Query |

Results | Messages | Execution plan

| | employee |
|---|---|
| 1 | Sven Buck |

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 1 rows

**2**

Sebagai uji coba, ubah nilai variabel @empid dari 5 ke 2 lalu eksekusi kembali. Perhatikan perubahan yang terjadi.

- Query

```
DECLARE @empid INT = 2;

-- Set nilai variabel dengan hasil query
DECLARE @empname NVARCHAR(30);
SELECT @empname = firstname + ' ' + lastname
FROM HR.Employees
WHERE empid = @empid;

-- Tampilkan nilai variabel
SELECT @empname AS employee;
```

- Hasil Query

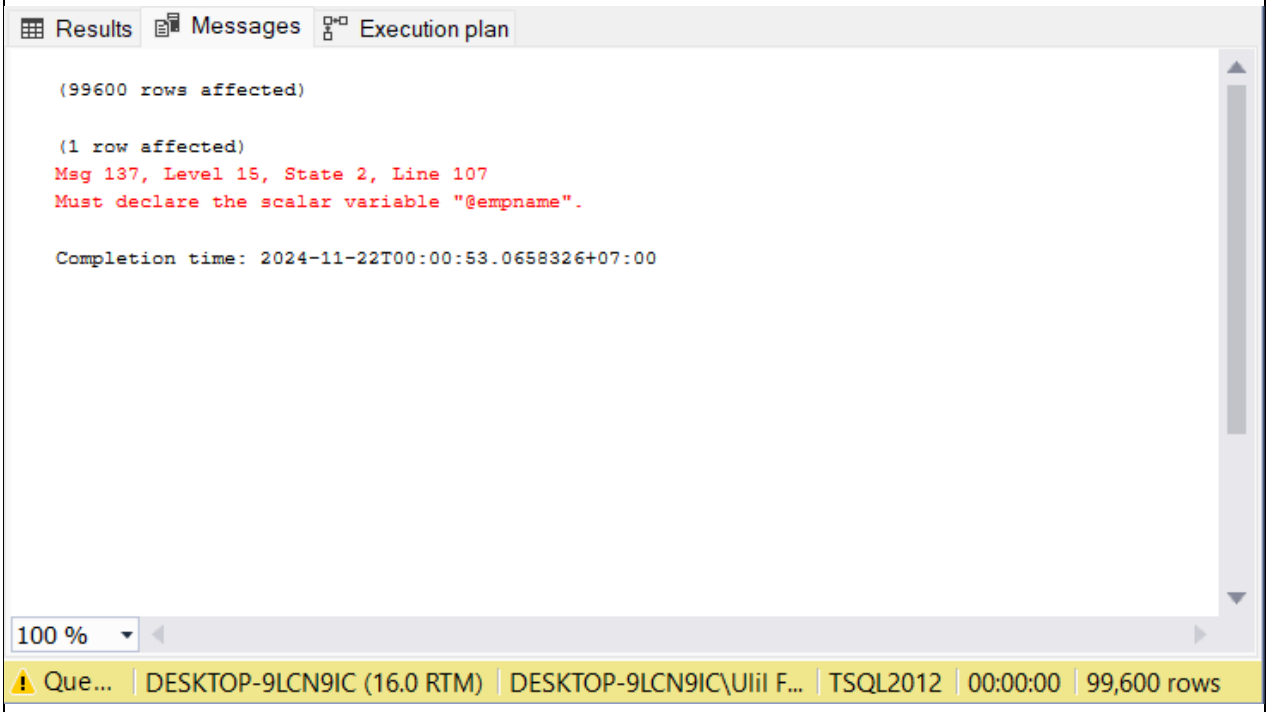
Results | Messages | Execution plan

| | employee |
|---|---|
| 1 | Don Funk |

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 1 rows

| | |
|---|---|
| | • Analisis<br>  ○ **Nilai @empid = 5**: Skrip akan menampilkan nama lengkap karyawan dengan empid 5.<br>  ○ **Nilai @empid = 2**: Skrip akan menampilkan nama lengkap karyawan dengan empid 2. |

**Praktikum Bagian 6 – DEKLARASI VARIABEL & BATCH: Menambah sebuah batch delimiter**

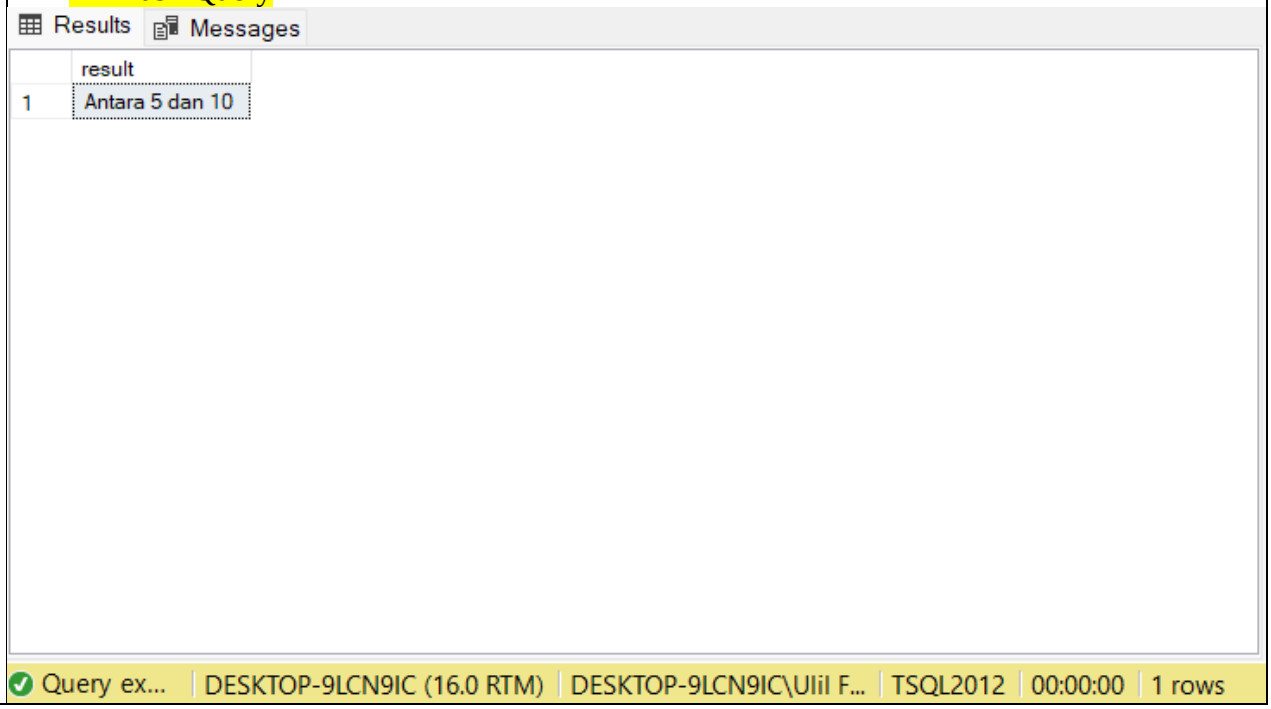
| Langkah | Keterangan |
|---|---|
| 1 | [Soal-20] Salinlah skrip T-SQL dari **[Soal-5]** di atas dengan menambahkan *batch delimiter* GO sebelum query SELECT seperti di bawah ini:<br><br>...<br><br>GO<br><br>SELECT @empname AS employee;<br><br>Setelah mengeksekusi skrip tersebut, apakah yang terjadi? Mengapa demikian?<br>• Query<br>`SELECT orderid, custid, orderdate`<br>`FROM Sales.TempOrders;`<br>`GO`<br>`SELECT @empname AS employee;`<br>• Hasil Query<br>  <br><br>• Analisis<br>○ GO adalah batch delimiter yang digunakan oleh SQL Server Management Studio (SSMS) dan alat klien lainnya untuk menandai akhir dari satu batch T-SQL dan awal dari batch berikutnya.<br>○ Ketika sql server menjalankan batch kedua, batch ini dieksekusi secara terpisah dan tidak memiliki akses ke variabel @empname yang dideklarasikan di batch sebelumnya, sehingga menghasilkan kesalahan karena variabel @empname tidak dideklarasikan dalam batch ini. |

**Praktikum Bagian 7 – CONTROL OF FLOW: Membuat conditional logic sederhana**

| Langkah | Keterangan |
|---|---|
| 1 | [Soal-21] Buatlah sebuah skrip T-SQL dengan mendeklarasikan variabel @result bertipe nvarchar(20) dan variabel @ibertipe integer bernilai 8. Tambahkan statement IF yang memenuhi *logic* di bawah ini:<br><br>• Jika variabel @ibernilai kurang dari 5, set nilai variabel @result menjadi "Kurang dari 5"<br>• Jika variabel @i bernilai antara 5 dan 10, set nilai variabel @result menjadi "Antara 5 dan 10"<br>• Jika variabel @ibernilai lebih dari 10, , set nilai variabel @result menjadi "Lebih dari 10"<br>• Selain dari itu, , set nilai variabel @resultmenjadi "Unknown"<br><br>Di bagian akhir, tambahkan sebuah query SELECT untuk menampilkan nilai variabel @result dengan memberi alias result.<br><br>• Query |

```
DECLARE @i INT = 8;
DECLARE @result NVARCHAR(20);

IF @i < 5
    SET @result = 'Kurang dari 5';
ELSE IF @i BETWEEN 5 AND 10
    SET @result = 'Antara 5 dan 10';
ELSE IF @i > 10
    SET @result = 'Lebih dari 10';
ELSE
    SET @result = 'Unknown';

SELECT @result AS result;
```

• Hasil Query



Eksekusi skrip yang sudah dibuat dan bandingkan dengan hasil berikut ini:

Results | Messages

| | result |
|---|---|
| 1 | Antara 5 dan 10 |

MCRURYA1B7\SQLEXPRESS (11.0... | sa (66) | TSQL2012 | 00:00:00 | 1 rows

**2**

[Soal-22] Modifikasilah skrip T-SQL dari [Soal-7] di atas dengan mengganti statement IF menjadi ekspresi CASE dan pastikan hasilnya sama.

- Query

```
DECLARE @i INT = 8;
DECLARE @result NVARCHAR(20);

SET @result = CASE
                 WHEN @i < 5 THEN 'Kurang dari 5'
                 WHEN @i BETWEEN 5 AND 10 THEN 'Antara 5 dan 10'
                 WHEN @i > 10 THEN 'Lebih dari 10'
                 ELSE 'Unknown'
              END;

SELECT @result AS result;
```

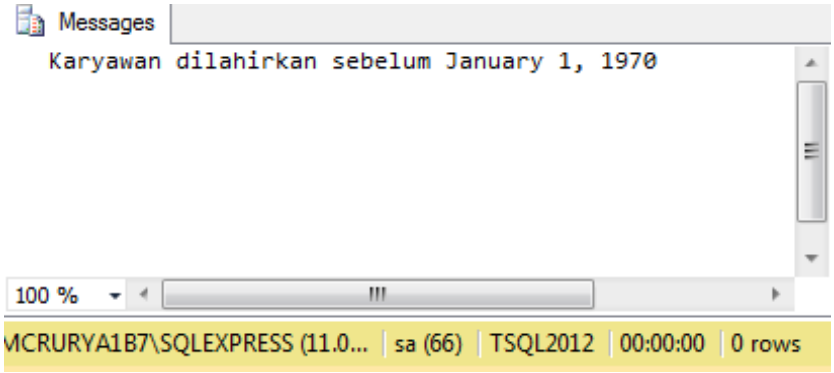
- Hasil Query

Results | Messages

| | result |
|---|---|
| 1 | Antara 5 dan 10 |

Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 1 rows

**Praktikum Bagian 8 – CONTROL OF FLOW: Mengecek tanggal lahir karyawan**

| Langkah | Keterangan |
|---|---|
| 1 | [Soal-23] Ikuti langkah berikut ini untuk membuat kode T-SQL yang mengecek tanggal lahir karyawan:<br>• Pertama, deklarasikan 2 variabel, yakni @birthdate dan @cmpdate (keduanya bertipe data date).<br>• Set nilai variabel @birthdate sebagai hasil dari query SELECT terhadap kolom birthdate dari tabel **HR.Employees**, dimana empid-nya adalah 5.<br>• Set variabel @cmpdate berisi tanggal January 1, 1970<br>• Buatlah pernyataan kondisional IF dengan membandingkan nilai @birthdate dan @cmpdate. Apabila @birthdate lebih kecil dari @cmpdate, gunakan perintah PRINT untuk menampilkan pesan "Karyawan dilahirkan sebelum Januari 1, 1970". Selain itu, tampilkan pesan "Karyawan dilahirkan pada atau setelah Januari 1, 1970".<br>• Eksekusi keseluruhan skrip T-SQL di atas.<br><br>Hasil yang benar ditunjukkan pada tampilan berikut:<br>  <br><br>• Query<br>`-- Deklarasi variabel`<br>`DECLARE @birthdate DATE;`<br>`DECLARE @cmpdate DATE;`<br><br>`-- Set nilai variabel @birthdate`<br>`SELECT @birthdate = birthdate`<br>`FROM HR.Employees`<br>`WHERE empid = 5;`<br><br>`-- Set nilai variabel @cmpdate`<br>`SET @cmpdate = '1970-01-01';`<br><br>`-- Pernyataan kondisional IF`<br>`IF @birthdate < @cmpdate`<br>`  PRINT 'Karyawan dilahirkan sebelum Januari 1, 1970';`<br>`ELSE`<br>`  PRINT 'Karyawan dilahirkan pada atau setelah Januari 1, 1970';` |

- Hasil Query

Messages | Execution plan

```
(1 row affected)
Karyawan dilahirkan sebelum Januari 1, 1970

Completion time: 2024-11-22T00:20:00.0030540+07:00
```

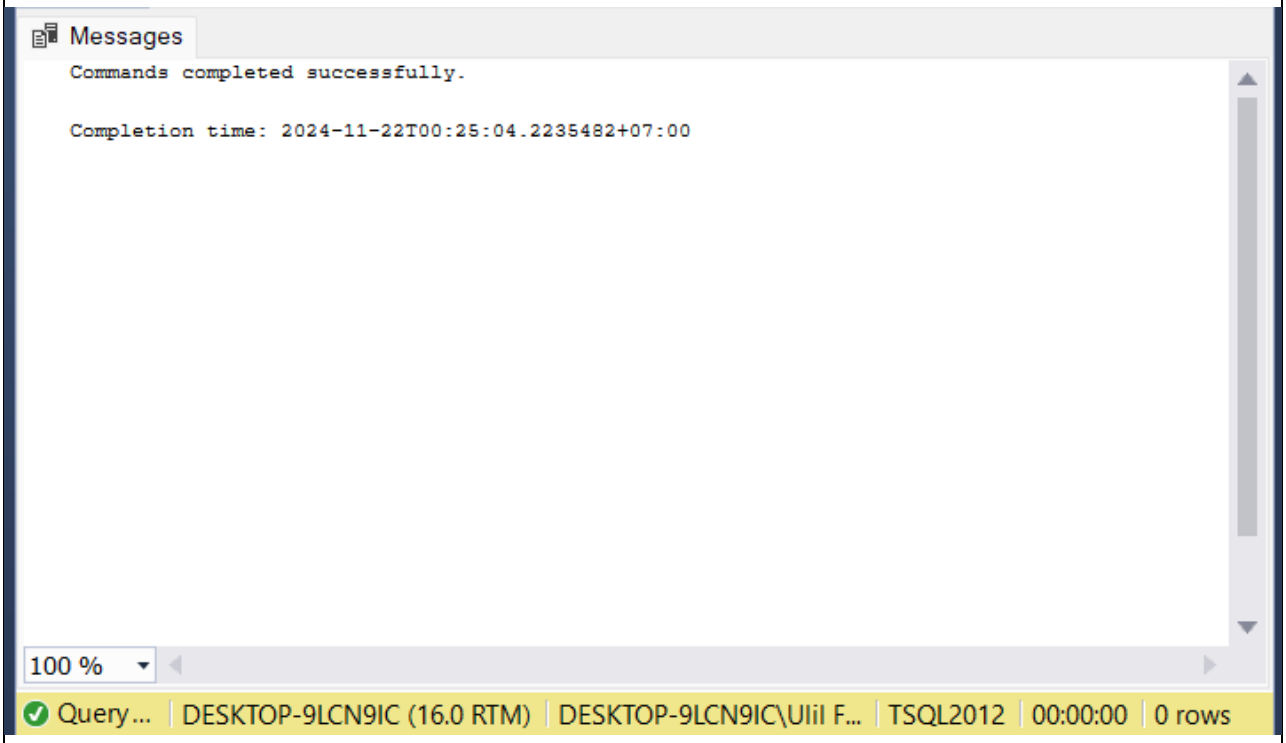
100 %

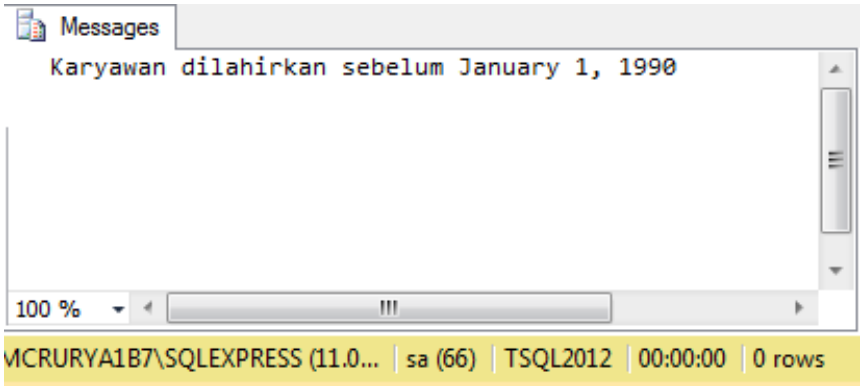
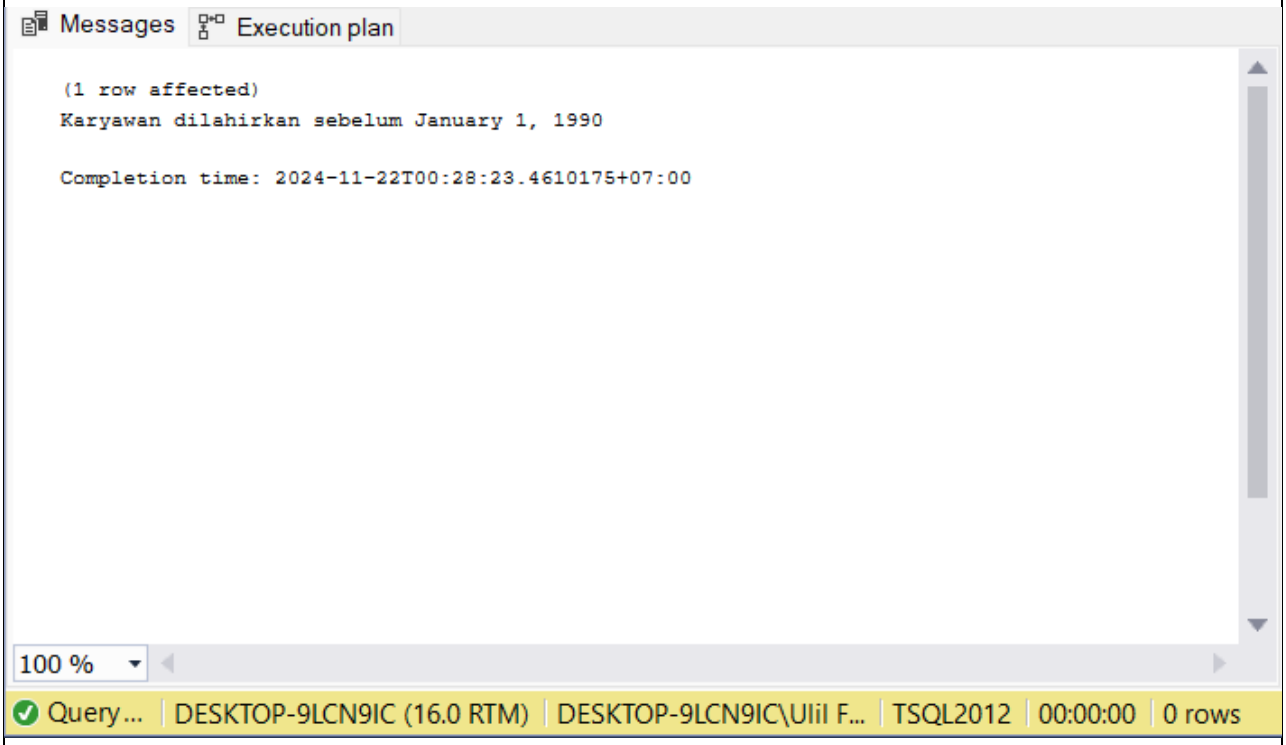
Query ex... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

**Praktikum Bagian 9 – CONTROL OF FLOW: Membuat dan mengeksekusi stored procedure**

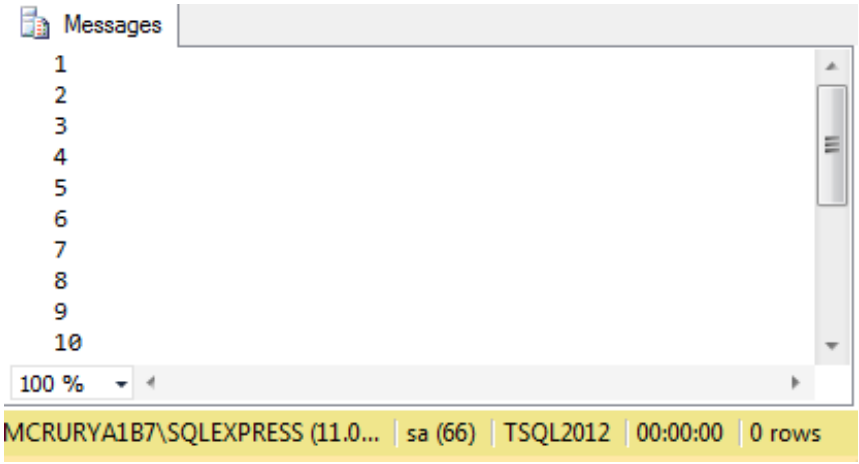
| Langkah | Keterangan |
|---|---|
| **1** | Salin & jalankan skrip T-SQL berikut ini: |

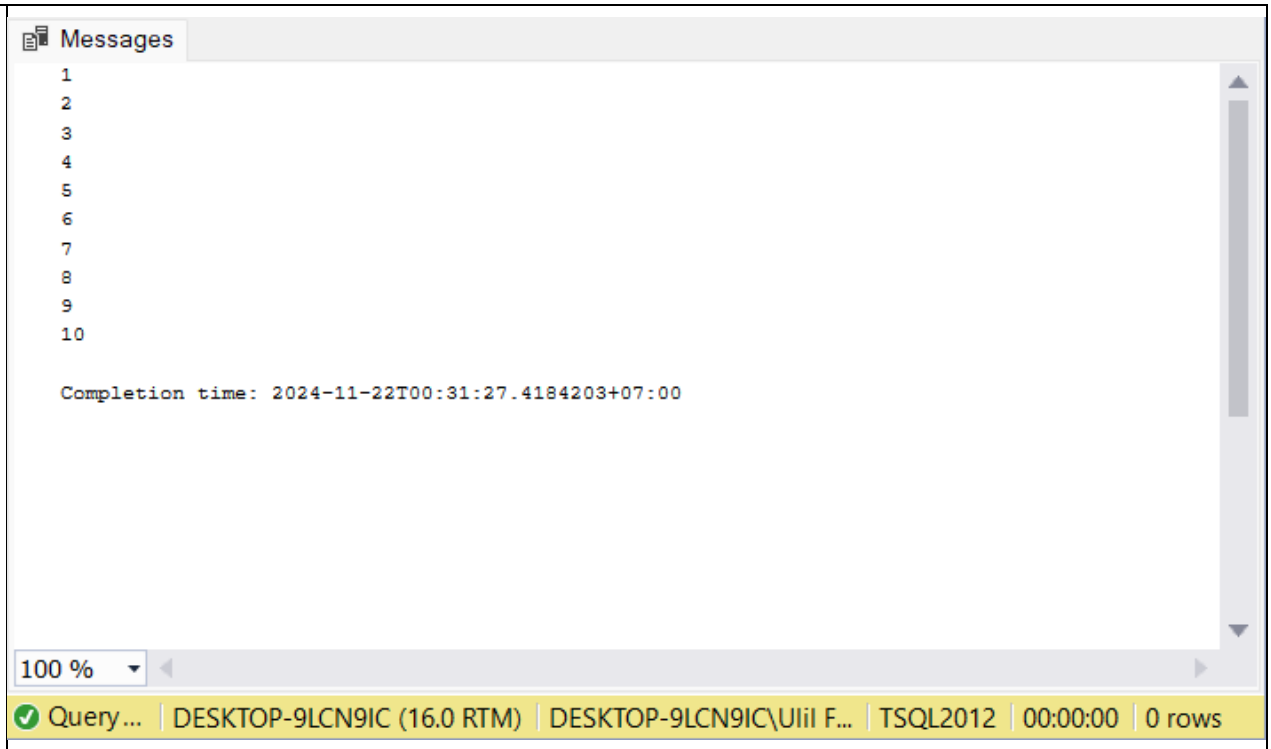
```
CREATE PROCEDURE Sales.CheckPersonBirthDate
       @empid int,
       @cmpdate date
AS

DECLARE
       @birthdate date;

SET @birthdate = (SELECT birthdate FROM HR.Employees WHERE empid = @empid);

IF @birthdate < @cmpdate
       PRINT 'Karyawan dilahirkan sebelum ' + FORMAT(@cmpdate, 'MMMM d, yyyy', 'en-US');
ELSE
       PRINT 'Karyawan dilahirkan pada atau setelah ' + FORMAT(@cmpdate, 'MMMM d, yyyy',
'en-US');
```



| | |
|---|---|
| **2** | [Soal-24] Stored procedure bernama **Sales.CheckPersonBirthDate** pada Langkah 1 di atas mempunyai 2 parameter, yakni @empid (untuk menentukan ID karyawan) dan @cmpdate (untuk perbandingan tanggal).<br><br>Lakukan perintah EXECUTE pada stored procedure tersebut dengan memasukkan parameter @empid = 3 dan @cmpdate yang di-set ke tanggal January 1, 1990. Hasil<br><br>yang benar ditunjukkan pada tampilan berikut:<br><br>  <br><br><br>• Query<br>`EXEC Sales.CheckPersonBirthDate @empid = 3, @cmpdate = '1990-01-01';`<br><br>• Hasil Query<br>  <br> |

**Praktikum Bagian 10 – CONTROL OF FLOW: Melakukan loop/ pengulangan menggunakan pernyataan WHILE**

| Langkah | Keterangan |
|---|---|
| 1 | [Soal-25] Buatlah sebuah skrip T-SQL yang berisi looping/ pengulangan dengan mengikuti langkah berikut:<br>• Pertama, deklarasikan sebuah variabel @iyang bertipe data integer bernilai 1<br>• Lalu buatlah sebuah pengulangan dengan menggunakan pernyataan WHILE, dimana selama nilai variabel @ikurang dari 10, tampilkan/ cetak variabel @idan tambahkan nilai @isecara incremental dengan menambah 1 (@i+1).<br><br>Hasil yang benar ditunjukkan pada tampilan berikut:<br>  <br><br>• Query<br>`-- Deklarasi variabel`<br>`DECLARE @i INT = 1;`<br><br>`-- Pengulangan dengan WHILE`<br>`WHILE @i <= 10`<br>`BEGIN`<br>`  -- Tampilkan nilai variabel @i`<br>`  PRINT @i;`<br><br>`  -- Tambahkan nilai @i secara incremental`<br>`  SET @i = @i + 1;`<br>`END;`<br>• Hasil Query |




**Praktikum Bagian 11 – DYNAMIC SQL: Membuat SQL dinamis tanpa parameter**

| Langkah | Keterangan |
|---|---|
| **1** | [Soal-26] Buatlah skrip T-SQL dengan mendeklarasikan variabel bernama @SQLstr bertipe data nvarchar(200). Lalu, isikan dengan string berupa statement SELECT yang mengambil kolom *empid*, *firstname*, dan *lastname* dari tabel **HR.Employees**.<br><br>• **Perhatikan** bahwa yang diambil adalah string statement-nya, bukan hasil query-nya. Contoh: SET @SQLstr = N'SELECT. ..... ';<br>Bukan SET @SQLstr = (SELECT..... );<br>• Lakukan perintah EXECUTE yang mengeksekusi pernyataan query yang sudah di-set dalam variabel @SQLstr. (Petunjuk: gunakan sp_executesql)<br><br>Hasil yang benar ditunjukkan pada tampilan berikut:<br><br>Results \| Messages<br>empid firstname lastname<br>1 1 Sara Davis<br>2 2 Don Funk<br>3 3 Judy Lew<br>4 4 Yael Peled<br>5 5 Sven Buck<br>6 6 Paul Suurs<br>7 7 Russell King<br>8 8 Maria Cameron<br>MCRURYA1B7\SQLEXPRESS (11.0... \| sa (66) \| TSQL2012 \| 00:00:00 \| 9 rows |

- Query

```
-- Deklarasi variabel
DECLARE @SQLstr NVARCHAR(200);

-- Isi variabel dengan pernyataan SELECT
SET @SQLstr = N'SELECT empid, firstname, lastname FROM HR.Employees';

-- Eksekusi pernyataan query
EXEC sp_executesql @SQLstr;
```

- Hasil Query

Results | Messages | Execution plan

| | empid | firstname | lastname |
|---|---|---|---|
| 1 | 1 | Sara | Davis |
| 2 | 2 | Don | Funk |
| 3 | 3 | Judy | Lew |
| 4 | 4 | Yael | Peled |
| 5 | 5 | Sven | Buck |
| 6 | 6 | Paul | Suurs |
| 7 | 7 | Russell | King |
| 8 | 8 | Maria | Cameron |
| 9 | 9 | Zoya | Dolgopyatova |

Query ... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 9 rows

**2** [Soal-27] Salinlah skrip T-SQL dari **[Soal-26]** di atas, lakukan modifikasi dengan mengikuti langkah berikut:

- Tambahkan filter WHERE dimana empid = @empidpada string pernyataan SELECT yang disimpan pada variabel @SQLstr
- Deklarasikan variabel baru bernama @SQLParambertipe data nvarchar(100)untuk menyimpan string definisi dari @empidyang bertipe data int
- Seperti halnya pada **[Soal-13]**, lakukan perintah EXECUTE dengan menambahkan parameter @empid = 5

Hasil yang benar ditunjukkan pada tampilan berikut:

Results | Messages

| | empid | firstname | lastname |
|---|---|---|---|
| 1 | 5 | Sven | Buck |

- Query

```
-- Deklarasi variabel
DECLARE @SQLstr NVARCHAR(200);
DECLARE @SQLParam NVARCHAR(100);
DECLARE @empid INT = 5;

-- Isi variabel dengan pernyataan SELECT dan filter WHERE
SET @SQLstr = N'SELECT empid, firstname, lastname FROM HR.Employees
WHERE empid = @empid';

-- Deklarasikan variabel untuk parameter
SET @SQLParam = N'@empid INT';

-- Eksekusi pernyataan query dengan parameter
EXEC sp_executesql @SQLstr, @SQLParam, @empid = @empid;
```

- Hasil Query

Results | Messages | Execution plan

| | empid | firstname | lastname |
|---|---|---|---|
| 1 | 5 | Sven | Buck |

Query... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 1 rows

**Praktikum Bagian 12 – SYNONYMS: Membuat dan menggunakan synonym**

| Langkah | Keterangan |
|---|---|
| 1 | [Soal-28] Tulislah sebuah skrip T-SQL yang membuat SYNONYM bernama **dbo.Pelanggan** dari tabel **Sales.Customers** dalam database TSQL. Eksekusi skrip tersebut.<br>Kemudian, buatlah query SELECT terhadap synonym **dbo.Pelanggan** yang mengambil kolom<br>*contacttitle* dan *contactname*. Eksekusi kembali skrip T-SQL ini.<br><br>Hasil yang benar ditunjukkan pada tampilan berikut: |

Results | Messages

| | contacttitle | contactname |
|---|---|---|
| 1 | Sales Representative | Allen, Michael |
| 2 | Owner | Hassall, Mark |
| 3 | Owner | Peoples, John |
| 4 | Sales Representative | Arndt, Torsten |
| 5 | Order Administrator | Higginbotham, Tom |
| 6 | Sales Representative | Poland, Carole |
| 7 | Marketing Manager | Bansal, Dushyant |
| 8 | Owner | Ilyina, Julia |
| 9 | Owner | Raghav, Amritansh |
| 10 | Accounting Manager | Bassols, Pilar Colome |
| 11 | Sales Representative | Jaffe, David |

Query executed successfully. | MCRURYA1B7\SQLEXPRESS (11.0... | sa (66) | TSQL2012 | 00:00:00 | 91 rows

- Query

```
CREATE SYNONYM dbo.Pelanggan FOR Sales.Customers;
GO

SELECT contacttitle, contactname
FROM dbo.Pelanggan;
```

- Hasil Query

Results | Messages | Execution plan

| | contacttitle | contactname |
|---|---|---|
| 1 | Sales Representative | Allen, Michael |
| 2 | Owner | Hassall, Mark |
| 3 | Owner | Peoples, John |
| 4 | Sales Representative | Arndt, Torsten |
| 5 | Order Administrator | Higginbotham, Tom |
| 6 | Sales Representative | Poland, Carole |
| 7 | Marketing Manager | Bansal, Dushyant |
| 8 | Owner | Ilyina, Julia |
| 9 | Owner | Raghav, Amritansh |
| 10 | Accounting Manager | Bassols, Pilar Colome |
| 11 | Sales Representative | Jaffe, David |
| 12 | Sales Agent | Ray, Mike |
| 13 | Marketing Manager | Benito, Almudena |
| 14 | Owner | Jelitto, Jacek |
| 15 | Sales Associate | Richardson, Shawn |
| 16 | Sales Representative | Birkby, Dana |

Quer... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 91 rows

| | |
|---|---|
| **2** | Untuk menghapus SYNONYM yang dibuat sebelumnya, jalankan skrip berikut:<br><br>DROP SYNONYM dbo.Pelanggan;<br><br>Messages<br>Commands completed successfully.<br><br>Completion time: 2024-11-22T00:42:35.5167264+07:00<br><br>100 %<br>Query... DESKTOP-9LCN9IC (16.0 RTM) DESKTOP-9LCN9IC\Ulil F... TSQL2012 00:00:00 0 rows |

**TOPIK 13.3 – ERROR HANDLING**

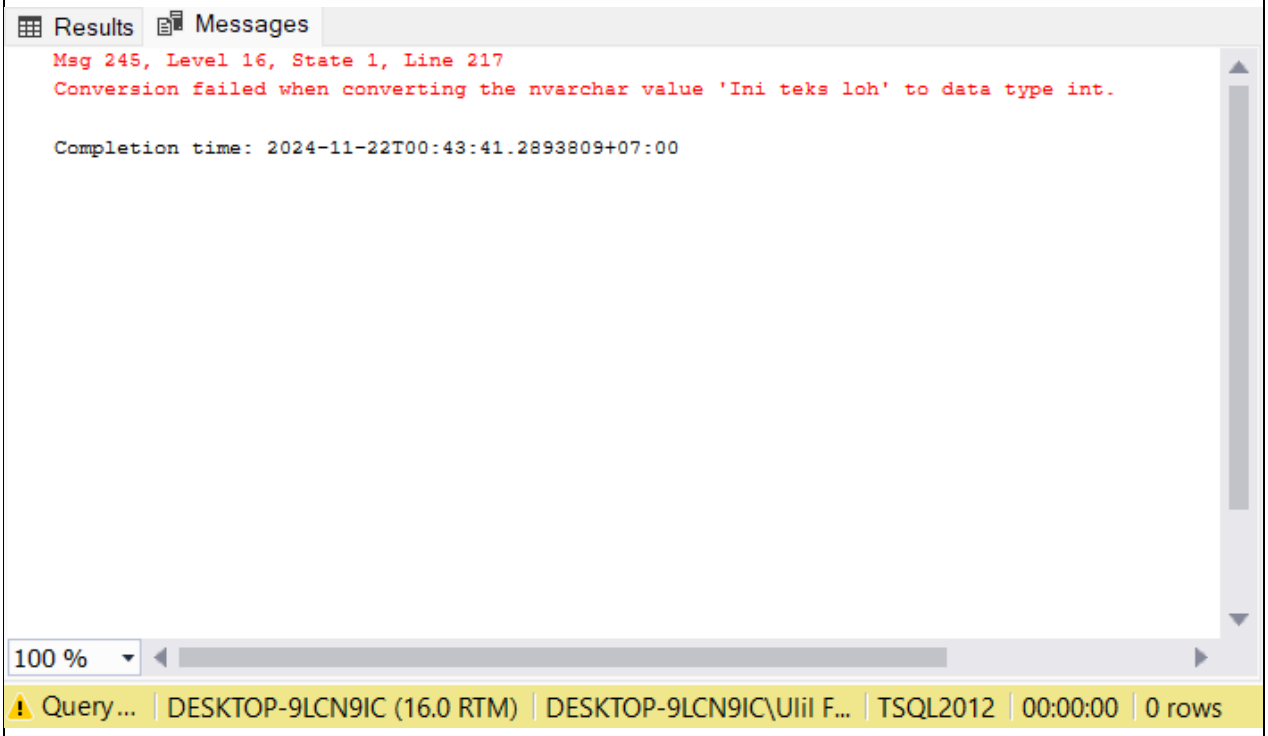
**Praktikum Bagian 13 – TRY / CATCH: Membuat blok TRY / CATCH sederhana**

<table>
<tr><th>Langkah</th><th>Keterangan</th></tr>
<tr><td>1</td><td>
Salin dan eksekusi pernyataan SELECT berikut ini:

```
SELECT CAST(N'Ini teks loh' AS int);
```

Perhatikan error yang terjadi saat dieksekusi.

- Hasil Query




- Analisis
  - Konversi ini tidak valid karena 'Ini teks loh' bukan angka dan tidak dapat dikonversi menjadi integer.
  - SQL Server tidak bisa mengonversi teks yang tidak mengandung angka menjadi tipe data numerik seperti integer.
</td></tr>
<tr><td>2</td><td>
[Soal-29] Buatlah konstruksi TRY / CATCH dengan menempatkan query pada Langkah 1 di atas dalam blok TRY. Sedangkan dalam blok CATCH, isikan perintah untuk menampilkan teks “Error”. Jalankan skrip T-SQL tersebut, lalu pada tab Messages, akan menampilkan pesan:

(0 row(s) affected) Error

- Query

```
BEGIN TRY
    -- Query yang akan menghasilkan error
    SELECT CAST(N'Ini teks loh' AS int);
END TRY
```
</td></tr>
</table>

```
BEGIN CATCH
    -- Pesan error yang akan ditampilkan
    PRINT 'Error';
END CATCH;
```

- Hasil Query

**Praktikum Bagian 14 – TRY / CATCH: Menampilkan kode & pesan error**

| Langkah | Keterangan |
|---|---|
| **1** | Salin dan eksekusi skrip T-SQL berikut ini dan perhatikan hasilnya:<br><br>`DECLARE @num varchar(20) = '0';`<br><br>`BEGIN TRY`<br>`  PRINT 5. / CAST(@num AS numeric(10,4));`<br>`END TRY`<br>`BEGIN CATCH`<br>`END CATCH;` |

<table>
<tr><td></td><td>

```
Messages
Commands completed successfully.

Completion time: 2024-11-22T00:51:51.0468217+07:00
```

100 %

Query... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

</td></tr>
<tr><td>2</td><td>

[Soal-30] Pada langkah 1 di atas, jika memperhatikan nilai variabel @num, semestinya dihasilkan error “division by zero”, tetapi nyatanya tidak. Mengapa demikian?

- Analisis
  - Pada langkah di atas, meskipun nilai variabel @num adalah '0', tidak terjadi error "division by zero" karena blok CATCH tidak memiliki pernyataan untuk menangkap dan menampilkan error. Error terjadi, tetapi tidak ada cara untuk melihatnya tanpa pernyataan PRINT dalam blok CATCH.

</td></tr>
<tr><td>3</td><td>

[Soal-31] Modifikasilah skrip T-SQL pada Langkah 1 di atas dengan menambahkan 2 (dua) pernyataan PRINT pada bagian blok CATCH. Pernyataan yang pertama untuk menampilkan nomer error dengan menggunakan fungsi ERROR_NUMBER dan pernyataan kedua untuk menampilkan pesan error dengan memakai fungsi ERROR_MESSAGE.

Untuk memperjelas, tambahkan string label “Error Number:” pada pesan pertama dan string label “Error Message:” pada pesan kedua.

Eksekusi dan bandingkan hasilnya dengan tampilan berikut:

</td></tr>
<tr><td></td><td>

```
Messages
Error Number: 8134
Error Message: Divide by zero error encountered.
```

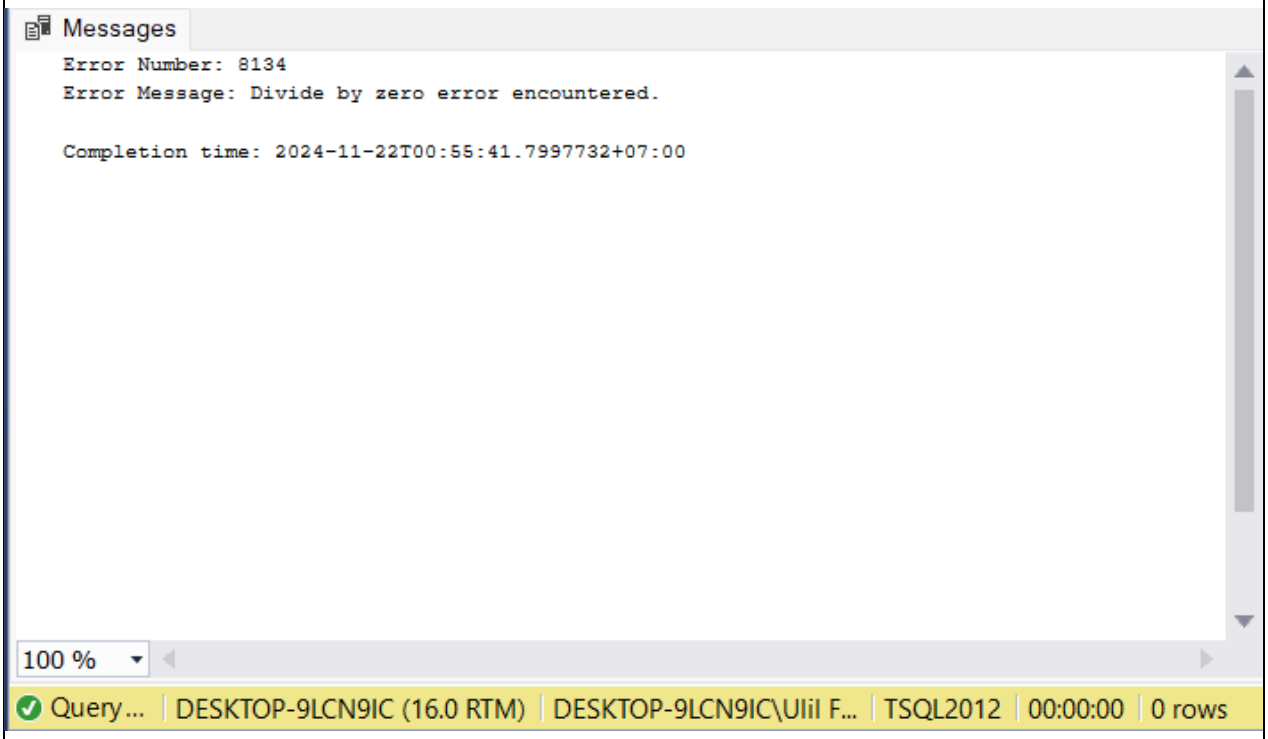
- Query

```
DECLARE @num varchar(20) = '0';

BEGIN TRY
    PRINT 5. / CAST(@num AS numeric(10,4));
END TRY
```

</td></tr>
</table>

```
BEGIN CATCH
    PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS varchar);
    PRINT 'Error Message: ' + ERROR_MESSAGE();
END CATCH;
```

- Hasil Query



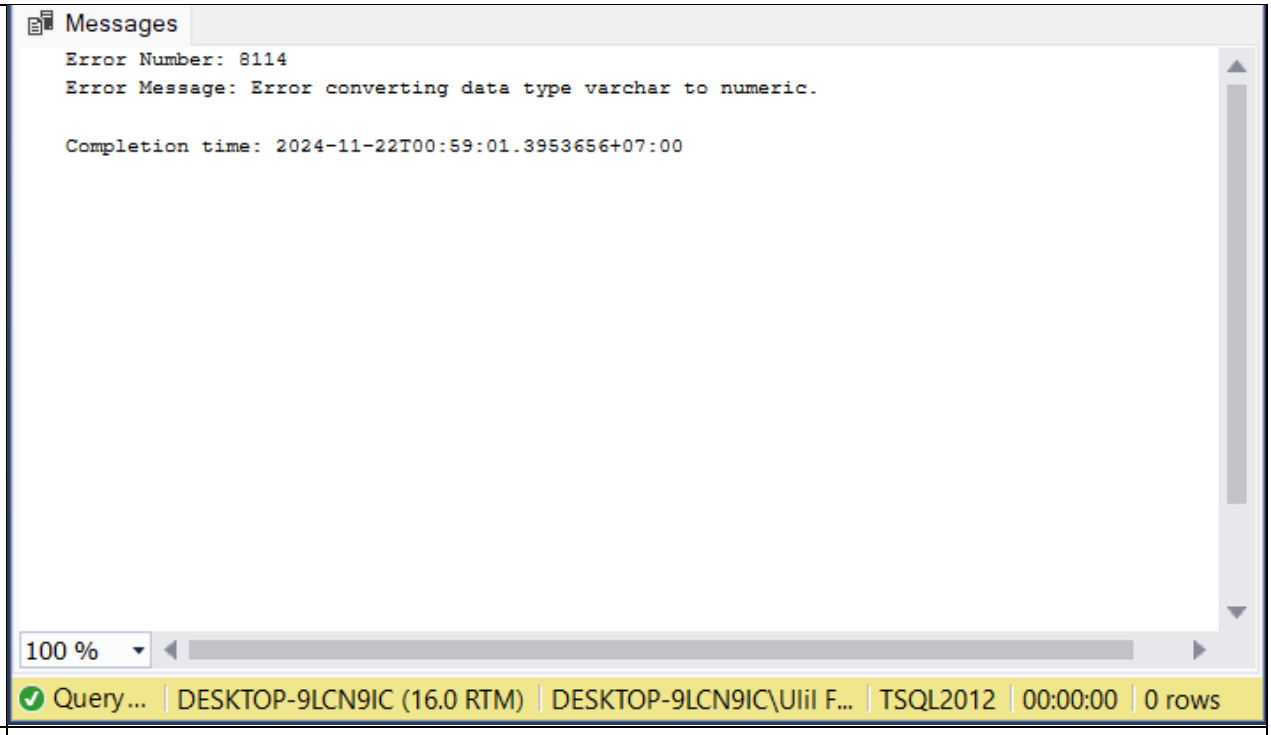

**4**

Sebagai uji coba, masih dengan skrip T-SQL yang sama, ubah nilai variabel @numdari 0 menjadi 'A'.

- Query

```
DECLARE @num varchar(20) = 'A';

BEGIN TRY
    PRINT 5. / CAST(@num AS numeric(10,4));
END TRY
BEGIN CATCH
    PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS varchar);
    PRINT 'Error Message: ' + ERROR_MESSAGE();
END CATCH;
```

- Hasil Query



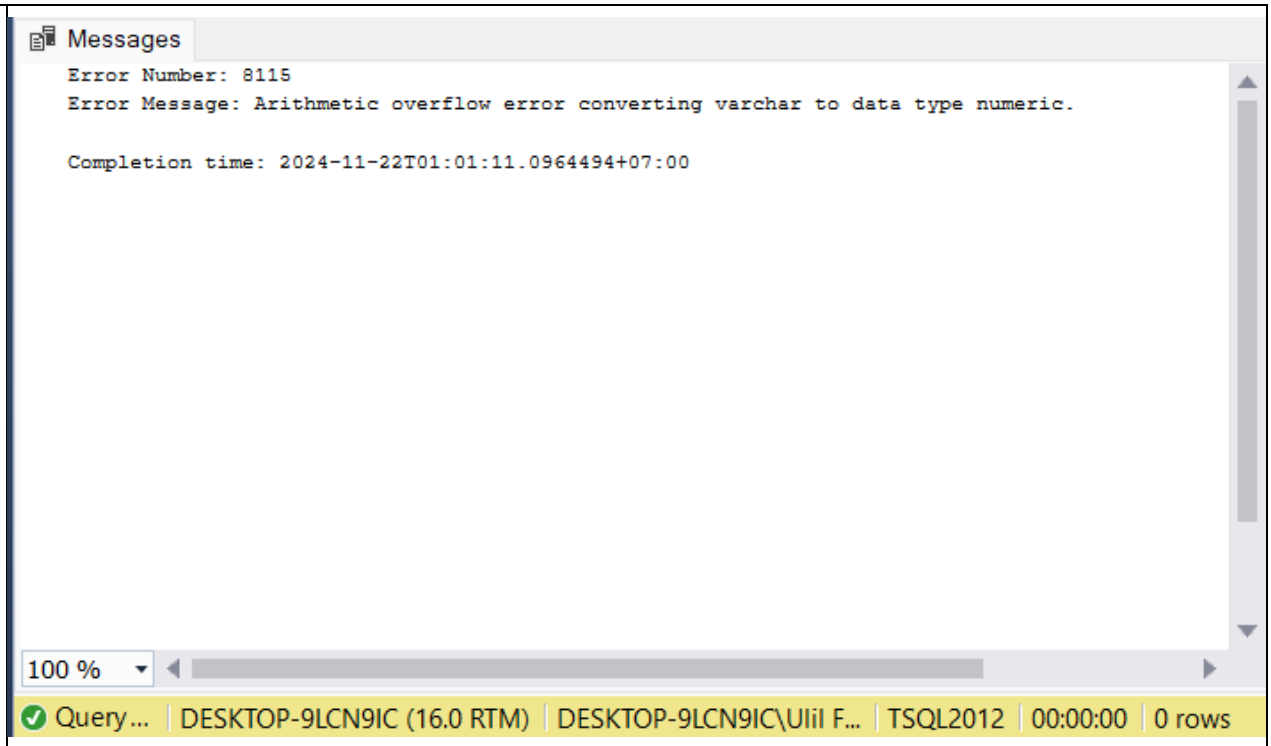

**5**

Sebagai uji coba, masih dengan skrip T-SQL yang sama, ubah nilai variabel @num dari ‘A’
menjadi 1000000000.

- Query

```
DECLARE @num varchar(20) = '1000000000';

BEGIN TRY
    PRINT 5. / CAST(@num AS numeric(10,4));
END TRY
BEGIN CATCH
    PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS varchar);
    PRINT 'Error Message: ' + ERROR_MESSAGE();
END CATCH;
```

- Hasil Query



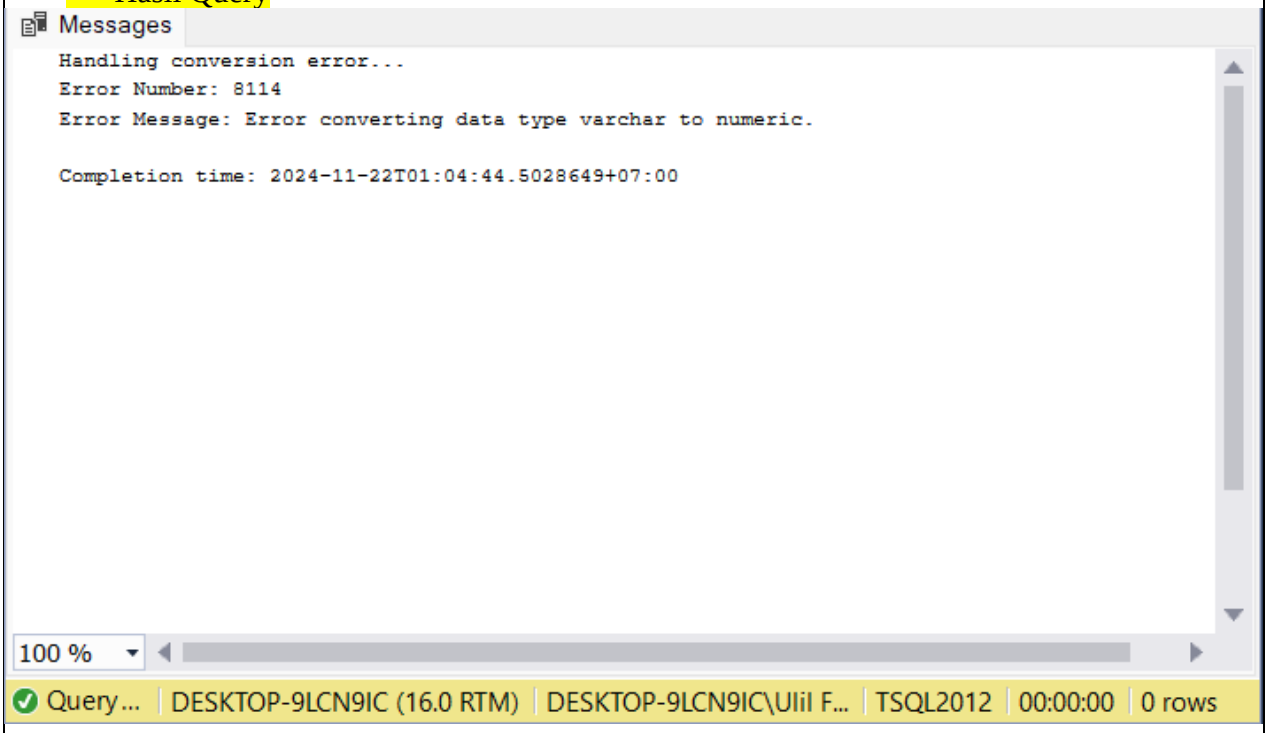

**Praktikum Bagian 15 – TRY / CATCH: Menambahkan conditional logic pada blok CATCH**

| Langkah | Keterangan |
|---|---|
| **1** | [Soal-32] Dengan tetap menggunakan skrip T-SQL pada Bagian 12 Langkah 1, lakukan modifikasi dengan menambahkan pernyataan IF pada bagian blok CATCH sebelum pernyataan PRINT.<br>• Pernyataan IF tersebut untuk mengecek apakah nomer error = 245 atau 8114.<br>  ○ Apabila kondisi ini terpenuhi, tampilkan pesan “Handling conversion error...” dengan perintah PRINT.<br>• Jika tidak sama dengan 245 atau 8114,<br>  ○ tampilkan pesan “Handling NON conversion error...”.<br>• Terakhir, set nilai variabel @numsebagai ‘A’, lalu eksekusi skrip T-SQL tersebut.<br><br>Bandingkan hasilnya dengan tampilan berikut:<br><br>Messages<br>Handling conversion error...<br>Error Number: 8114<br>Error Message: Error converting data type varchar to numeric.<br><br>• Query |

```
DECLARE @num varchar(20) = 'A';

BEGIN TRY
    PRINT 5. / CAST(@num AS numeric(10,4));
END TRY
BEGIN CATCH
    IF ERROR_NUMBER() = 245 OR ERROR_NUMBER() = 8114
    BEGIN
```

```
        PRINT 'Handling conversion error...';
    END
    ELSE
    BEGIN
        PRINT 'Handling NON conversion error...';
    END

    PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS varchar);
    PRINT 'Error Message: ' + ERROR_MESSAGE();
END CATCH;
```

- Hasil Query




**2**

Sebagai uji coba, masih dengan skrip T-SQL yang sama, ubah nilai variabel @num dari 'A'
menjadi 0.

- Query

```
DECLARE @num varchar(20) = '0';

BEGIN TRY
    PRINT 5. / CAST(@num AS numeric(10,4));
END TRY
BEGIN CATCH
    IF ERROR_NUMBER() = 245 OR ERROR_NUMBER() = 8114
    BEGIN
        PRINT 'Handling conversion error...';
    END
    ELSE
    BEGIN
        PRINT 'Handling NON conversion error...';
    END
```

```
    PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS varchar);
    PRINT 'Error Message: ' + ERROR_MESSAGE();
END CATCH;
```

- Hasil Query

```
Messages
Handling NON conversion error...
Error Number: 8134
Error Message: Divide by zero error encountered.

Completion time: 2024-11-22T01:06:36.7002906+07:00
```

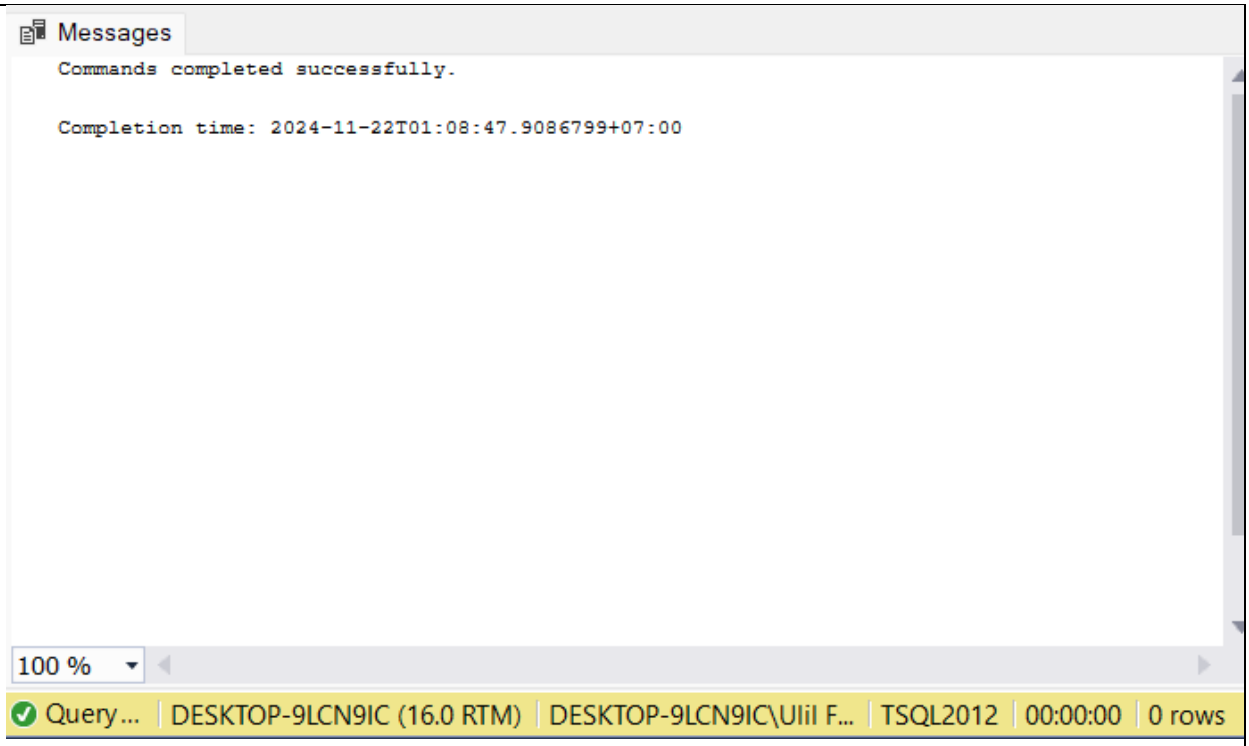
100 %

Query... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

**Praktikum Bagian 16 – TRY / CATCH: Mengeksekusi stored procedure pada blok CATCH**

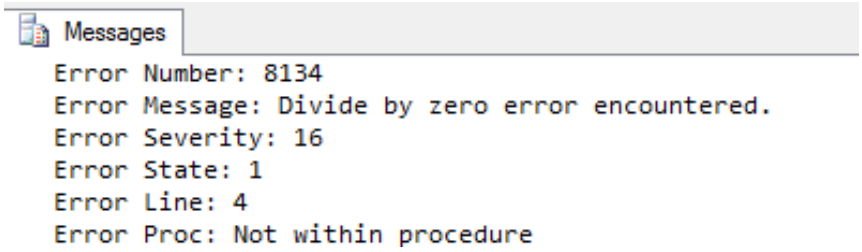
| | |
|---|---|
| **1** | Salin dan eksekusi skrip T-SQL yang membuat sebuah stored procedure dbo.GetErrorInfo di bawah ini. |

```
CREATE PROCEDURE dbo.GetErrorInfo AS
PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS varchar(10));
PRINT 'Error Message: ' + ERROR_MESSAGE();
PRINT 'Error Severity: ' + CAST(ERROR_SEVERITY() AS varchar(10));
PRINT 'Error State: ' + CAST(ERROR_STATE() AS varchar(10));
PRINT 'Error Line: ' + CAST(ERROR_LINE() AS varchar(10));
PRINT 'Error Proc: ' + COALESCE(ERROR_PROCEDURE(), 'Not within procedure');
```




**2**

[Soal-33] Buatlah sebuah konstruksi TRY / CATCH, dimana pada bagian blok CATCH, lakukan eksekusi stored procedure yang telah dibuat pada Langkah 1 di atas, lalu jalankan.




- Query

```
BEGIN TRY


    SELECT 1 / 0;
END TRY
BEGIN CATCH

    EXEC dbo.GetErrorInfo;
END CATCH;
```

- Hasil Query

```
Results | Messages

(0 rows affected)
Error Number: 8134
Error Message: Divide by zero error encountered.
Error Severity: 16
Error State: 1
Error Line: 4
Error Proc: Not within procedure

Completion time: 2024-11-22T01:16:05.7195596+07:00

100 %
Query... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows
```

**Praktikum Bagian 17 – THROW: Menggunakan THROW untuk mengirimkan kembali pesan error**

**1**

[Soal-34] Modifikasi skrip T-SQL dari Soal-19 di atas dengan menambahkan perintah THROW yang ditempatkan setelah pernyataan eksekusi stored procedure.

```
Messages
Error Number: 8134
Error Message: Divide by zero error encountered.
Error Severity: 16
Error State: 1
Error Line: 5
Error Proc: Not within procedure
Msg 8134, Level 16, State 1, Line 5
Divide by zero error encountered.
```

- Query

```
BEGIN TRY


      -- Menyebabkan kesalahan pembagian dengan nol
    SELECT 1 / 0;
END TRY
BEGIN CATCH
    -- Memanggil prosedur error info
    EXEC dbo.GetErrorInfo;

    -- Menambahkan perintah THROW
    THROW;
END CATCH;
```

- Hasil Query

Results | Messages

```
(0 rows affected)
Error Number: 8134
Error Message: Divide by zero error encountered.
Error Severity: 16
Error State: 1
Error Line: 5
Error Proc: Not within procedure
Msg 8134, Level 16, State 1, Line 326
Divide by zero error encountered.

Completion time: 2024-11-22T01:23:35.6184431+07:00
```

100 %

Query... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

**2**

[Soal-35] Modifikasi skrip T-SQL dari Soal-20 di atas dengan mengganti perintah THROW dengan pernyataan IF.

- Pernyataan IF tersebut untuk mengecek apakah nomer error = 8114.
- Apabila kondisi ini terpenuhi, tampilkan pesan "Handling division by zero..." dengan perintah PRINT.
- Jika tidak, tampilkan pesan "Throwing original error...".
- Terakhir, set nilai variabel @numsebagai 'A', lalu eksekusi skrip T-SQL tersebut.

Bandingkan hasilnya dengan tampilan berikut:

Messages

```
Error Number: 8114
Error Message: Error converting data type varchar to numeric.
Error Severity: 16
Error State: 5
Error Line: 6
Error Proc: Not within procedure
Throwing original error
Msg 8114, Level 16, State 5, Line 6
Error converting data type varchar to numeric.
```

- Query

```
DECLARE @num varchar(20) = 'A';

BEGIN TRY


    PRINT 5. / CAST(@num AS numeric(10,4));
END TRY
BEGIN CATCH
```

```
    -- Memanggil prosedur error info
    EXEC dbo.GetErrorInfo;

    -- Menambahkan pernyataan IF untuk mengecek error number
    IF ERROR_NUMBER() = 8114
    BEGIN
        PRINT 'Handling division by zero';
    END
    ELSE
    BEGIN
        PRINT 'Throwing original error';
    END

    -- Mengatur nilai variabel @num sebagai 'A'
    SET @num = 'A';


    THROW;
END CATCH;
```

- Hasil Query

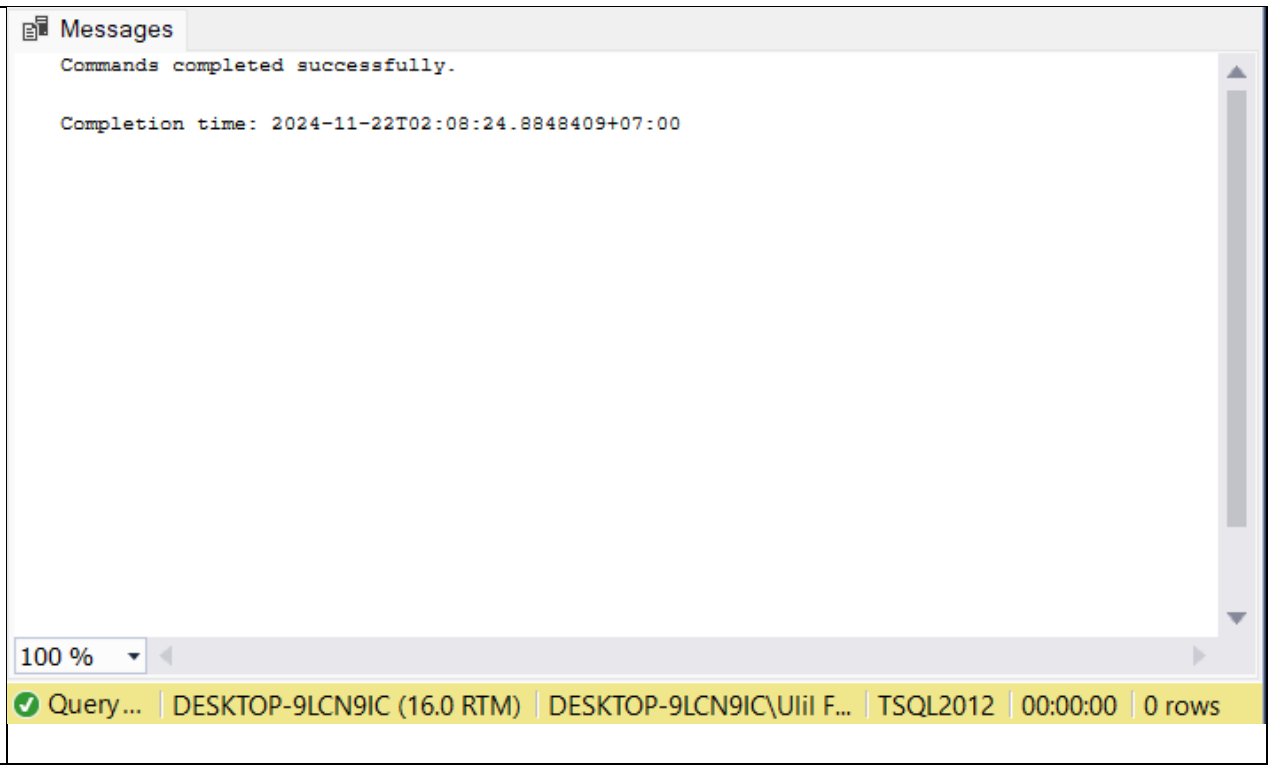
```
Messages
Error Number: 8114
Error Message: Error converting data type varchar to numeric.
Error Severity: 16
Error State: 5
Error Line: 6
Error Proc: Not within procedure
Handling division by zero
Msg 8114, Level 16, State 5, Line 343
Error converting data type varchar to numeric.

Completion time: 2024-11-22T01:51:59.3450704+07:00
```

100 %

Query... | DESKTOP-9LCN9IC (16.0 RTM) | DESKTOP-9LCN9IC\Ulil F... | TSQL2012 | 00:00:00 | 0 rows

- Tambahan
  - Dicontoh output yang tampil “Throwing original error” saat nomor error 8114
  - Sementara di soalnya diminta jika nomor error = 8114 maka akan tampil “Handling division by zero”
  - Jadinya saya ikut yang ada di soal saja

| | |
|---|---|
| **3** | [Soal-36] Salinlah skrip T-SQL berikut ini:<br><br>`DECLARE @msg AS varchar(2048);`<br>`SET @msg = 'Anda sedang mengerjakan Jobsheet 14 pada ' +`<br>`FORMAT(CURRENT_TIMESTAMP, 'MMMM d, yyyy', 'en-US') + '. Ini bukan error, bosque';`<br><br>Lalu, tambahkan pernyataan THROW setelah skrip di atas. Sebagai argumen pertama, isikan 50001, untuk argumen kedua, isikan variabel @msg,sedangkan argumen ketiga, isikan nilai 1.<br><br>Bandingkan hasilnya dengan tampilan berikut:<br><br>Messages<br>Msg 50001, Level 16, State 1, Line 4<br>Anda sedang mengerjakan Jobsheet 14 pada January 1, 2018. Ini bukan error, bosque<br><br>• Query<br>`DECLARE @msg AS varchar(2048);`<br>`SET @msg = 'Anda sedang mengerjakan Jobsheet 14 pada ' +`<br>`FORMAT(CURRENT_TIMESTAMP, 'MMMM d, yyyy', 'en-US') + '. Ini bukan error, bosque';`<br>`THROW 50001, @msg, 1;`<br><br>• Hasil Query<br>Messages<br>Msg 50001, Level 16, State 1, Line 366<br>Anda sedang mengerjakan Jobsheet 14 pada November 22, 2024. Ini bukan error, bosque<br><br>Completion time: 2024-11-22T02:05:57.9996743+07:00<br><br>100 %<br>Query... \| DESKTOP-9LCN9IC (16.0 RTM) \| DESKTOP-9LCN9IC\Ulil F... \| TSQL2012 \| 00:00:00 \| 0 rows |
| **5** | Untuk menghapus stored procedure yang dibuat sebelumnya, jalankan skrip berikut:<br><br>DROP PROCEDURE Sales.CheckPersonBirthDate; -- Praktikum 7 - Langkah 1 DROP PROCEDURE dbo.GetErrorInfo; |




***--- Selamat Mengerjakan ----***