

JOBSHEET

PRAKTIKUM BASIS DATA LANJUT

Jurusan Teknologi Informasi
POLITEKNIK NEGERI MALANG



PERTEMUAN 11

SQL SERVER – STORE PROCEDURE & TRIGGER

Team Teaching:

Dwi Puspitasari, S.Kom., M.Kom.

Yan Watequlis Syaifudin, ST., MMT., PhD.

Annisa Puspa Kirana, S. Kom, M.Kom

Dika Rizky Yuniarto, S.Kom, M.Kom

Muhammad Shulhan Khairy, S.Kom, M.Kom

Habibie Ed Dien, S.Kom., M.T.



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet Minggu-11: Stored Procedure & Trigger

Mata Kuliah Basis Data Lanjut (BDL)

Pengampu: Tim Ajar Basis Data Lanjut

November 2024

Topik

Stored Procedure & Trigger

Tujuan

1. Membuat stored procedure sederhana dengan SELECT statement.
2. Menggunakan input dan output parameter
3. Membuat TRIGGER (AFTER)
4. Membuat TRIGGER (INSTEAD OF)

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_Tugas14_Kelas_2DigitNomorAbsen_NamaLengkapAnda.pdf**
 - Contoh:
 - o **BDL_Tugas14_TI2Q_99_Ana.pdf**
 - Perhatikan baik-baik format penamaanya.
 - Kumpulkan file PDF tersebut sebagai laporan praktikum kepada dosen pengampu.
 - Selain pada nama file, cantumkan juga identitas Anda pada halaman pertama laporan tersebut.



Praktikum – Bagian 1: Menggunakan EXECUTE Statement Untuk Memanggil Stored Procedure

Langkah	Keterangan																																						
1	<p>Buat stored procedure bernama Sales.GetTopCustomers untuk menampilkan 10 pelanggan dengan pembelian tertinggi. Run SQL command berikut:</p> <pre>CREATE PROCEDURE Sales.GetTopCustomers AS SELECT TOP(10) C.CustomerID, C.ContactName, SUM(O.Value) AS SalesValue FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID GROUP BY C.CustomerID, C.ContactName ORDER BY salesvalue DESC;</pre>																																						
2	<p>[Soal-1] Tuliskan SQL command untuk menjalankan stored procedure yang telah dibuat pada langkah 1 dengan keyword EXECUTE. Pastikan output nya sama dengan gambar berikut</p> <div><div>Results Messages</div><table><tr><th></th><th>CustomerID</th><th>ContactName</th><th>SalesValue</th><th></th></tr><tr><td>1</td><td>63</td><td>Veronesi, Giorgio</td><td>110277.32</td><td rowspan="8"></td></tr><tr><td>2</td><td>20</td><td>Kane, John</td><td>104874.99</td></tr><tr><td>3</td><td>71</td><td>Navarro, Tomás</td><td>104361.96</td></tr><tr><td>4</td><td>65</td><td>Moore, Michael</td><td>51097.80</td></tr><tr><td>5</td><td>37</td><td>Cr?ciun, Ovidiu V.</td><td>49979.91</td></tr><tr><td>6</td><td>34</td><td>Cohen, Shy</td><td>32841.37</td></tr><tr><td>7</td><td>39</td><td>Song, Lolan</td><td>30908.39</td></tr><tr><td>8</td><td>24</td><td>San Juan, Patricia</td><td>29567.57</td></tr></table><div>RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 10 rows</div></div>		CustomerID	ContactName	SalesValue		1	63	Veronesi, Giorgio	110277.32		2	20	Kane, John	104874.99	3	71	Navarro, Tomás	104361.96	4	65	Moore, Michael	51097.80	5	37	Cr?ciun, Ovidiu V.	49979.91	6	34	Cohen, Shy	32841.37	7	39	Song, Lolan	30908.39	8	24	San Juan, Patricia	29567.57
	CustomerID	ContactName	SalesValue																																				
1	63	Veronesi, Giorgio	110277.32																																				
2	20	Kane, John	104874.99																																				
3	71	Navarro, Tomás	104361.96																																				
4	65	Moore, Michael	51097.80																																				
5	37	Cr?ciun, Ovidiu V.	49979.91																																				
6	34	Cohen, Shy	32841.37																																				
7	39	Song, Lolan	30908.39																																				
8	24	San Juan, Patricia	29567.57																																				



```
--1
CREATE PROCEDURE Sales.GetTopCustomers
AS
SELECT TOP(10) C.CustomerID,
C.ContactName, SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID
GROUP BY C.CustomerID, C.ContactName ORDER BY salesvalue DESC;

EXECUTE Sales.GetTopCustomers;
```

113 %

Results Messages

	CustomerID	ContactName	SalesValue
1	63	Veronesi, Giorgio	110277.32
2	20	Kane, John	104874.99
3	71	Navarro, Tomás	104361.96
4	65	Moore, Michael	51097.80
5	37	Cr?ciun, Ovidiu V.	49979.91
6	34	Cohen, Shy	32841.37
7	39	Song, Lolan	30908.39
8	24	San Juan, Patricia	29567.57
9	51	Taylor, Maurice	28872.19
10	89	Smith Jr., Ronaldo	27363.61

3

Modifikasi stored procedure Sales.GetTopCustomers dengan menjalankan script berikut:

```
ALTER PROCEDURE Sales.GetTopCustomers
AS
SELECT C.CustomerID,
       C.ContactName,
       SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O
INNER JOIN Sales.Customers AS C
ON O.CustomerID = C.CustomerID
GROUP BY C.CustomerID, C.ContactName
ORDER BY SalesValue DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;
```

4

Eksekusi kembali stored procedure Sales.GetTopCustomers seperti pada langkah 2
Pastikan outputnya sama dengan gambar berikut:



	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>CustomerID</th><th>ContactName</th><th>SalesValue</th><th></th></tr><tr><td>1</td><td>63</td><td>Veronesi, Giorgio</td><td>110277.32</td><td></td></tr><tr><td>2</td><td>20</td><td>Kane, John</td><td>104874.99</td><td></td></tr><tr><td>3</td><td>71</td><td>Navarro, Tomás</td><td>104361.96</td><td></td></tr><tr><td>4</td><td>65</td><td>Moore, Michael</td><td>51097.80</td><td></td></tr><tr><td>5</td><td>37</td><td>Cr?ciun, Ovidiu V.</td><td>49979.91</td><td></td></tr><tr><td>6</td><td>34</td><td>Cohen, Shy</td><td>32841.37</td><td></td></tr><tr><td>7</td><td>39</td><td>Song, Lolan</td><td>30908.39</td><td></td></tr><tr><td>8</td><td>24</td><td>San Juan, Patricia</td><td>29567.57</td><td></td></tr></table><div>RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 10 rows</div></div>		CustomerID	ContactName	SalesValue		1	63	Veronesi, Giorgio	110277.32		2	20	Kane, John	104874.99		3	71	Navarro, Tomás	104361.96		4	65	Moore, Michael	51097.80		5	37	Cr?ciun, Ovidiu V.	49979.91		6	34	Cohen, Shy	32841.37		7	39	Song, Lolan	30908.39		8	24	San Juan, Patricia	29567.57	
	CustomerID	ContactName	SalesValue																																											
1	63	Veronesi, Giorgio	110277.32																																											
2	20	Kane, John	104874.99																																											
3	71	Navarro, Tomás	104361.96																																											
4	65	Moore, Michael	51097.80																																											
5	37	Cr?ciun, Ovidiu V.	49979.91																																											
6	34	Cohen, Shy	32841.37																																											
7	39	Song, Lolan	30908.39																																											
8	24	San Juan, Patricia	29567.57																																											
5	<p>[Soal-2] Jika query pada stored procedure dimodifikasi, apakah command untuk mengeksekusi stored procedure tersebut perlu dimodifikasi juga?</p> <p>Tidak, command untuk mengeksekusi stored procedure tidak perlu dimodifikasi, kecuali terdapat perubahan dalam nama procedure atau parameter yang digunakan.</p>																																													

**Praktikum – Bagian 2: Menambahkan Parameter ke Stored Procedure**

Langkah	Keterangan																																				
1	<p>Modifikasi stored procedure Sales.GetTopCustomers dengan menambahkan parameter tahun pesanan (@orderYear) :</p> <pre>ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT AS SELECT C.CustomerID, C.ContactName, SUM(O.Value) AS SalesValue FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID WHERE YEAR(O.OrderDate) = @orderYear GROUP BY C.CustomerID, C.ContactName ORDER BY SalesValue DESC OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;</pre>																																				
2	<p>[Soal-3] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure Sales.GetTopCustomers untuk tahun 2007</p> <p>Pastikan hasilnya sama dengan gambar berikut:</p> <div><div>ResultsMessages</div><table><tr><th></th><th>CustomerID</th><th>ContactName</th><th>SalesValue</th></tr><tr><td>1</td><td>63</td><td>Veronesi, Giorgio</td><td>61109.92</td></tr><tr><td>2</td><td>71</td><td>Navarro, Tomás</td><td>57713.58</td></tr><tr><td>3</td><td>20</td><td>Kane, John</td><td>48096.27</td></tr><tr><td>4</td><td>51</td><td>Taylor, Maurice</td><td>23332.31</td></tr><tr><td>5</td><td>37</td><td>Cr?ciun, Ovidiu V.</td><td>20454.41</td></tr><tr><td>6</td><td>65</td><td>Moore, Michael</td><td>19383.75</td></tr><tr><td>7</td><td>73</td><td>Gonzalez, Nuria</td><td>16232.41</td></tr><tr><td>8</td><td>5</td><td>Higginbotham, Tom</td><td>13849.02</td></tr></table><div>RTM)DESKTOP-47A9O7C\Zuraid...DailyMart00:00:0010 rows</div></div>		CustomerID	ContactName	SalesValue	1	63	Veronesi, Giorgio	61109.92	2	71	Navarro, Tomás	57713.58	3	20	Kane, John	48096.27	4	51	Taylor, Maurice	23332.31	5	37	Cr?ciun, Ovidiu V.	20454.41	6	65	Moore, Michael	19383.75	7	73	Gonzalez, Nuria	16232.41	8	5	Higginbotham, Tom	13849.02
	CustomerID	ContactName	SalesValue																																		
1	63	Veronesi, Giorgio	61109.92																																		
2	71	Navarro, Tomás	57713.58																																		
3	20	Kane, John	48096.27																																		
4	51	Taylor, Maurice	23332.31																																		
5	37	Cr?ciun, Ovidiu V.	20454.41																																		
6	65	Moore, Michael	19383.75																																		
7	73	Gonzalez, Nuria	16232.41																																		
8	5	Higginbotham, Tom	13849.02																																		



```
--3
ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT
AS
SELECT C.CustomerID,
C.ContactName, SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID
WHERE YEAR(O.OrderDate) = @orderYear GROUP BY C.CustomerID, C.ContactName ORDER BY SalesValue DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

EXECUTE Sales.GetTopCustomers @orderYear = 2007;
```

113 %

Results Messages

	CustomerID	ContactName	SalesValue
1	63	Veronesi, Giorgio	61109.92
2	71	Navarro, Tomás	57713.58
3	20	Kane, John	48096.27
4	51	Taylor, Maurice	23332.31
5	37	Cr?ciun, Ovidiu V.	20454.41
6	65	Moore, Michael	19383.75
7	73	Gonzalez, Nuria	16232.41
8	5	Higginbotham, Tom	13849.02
9	35	Langohr, Kris	13482.74
10	24	San Juan, Patricia	13314.67

3

[Soal-4] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure Sales.GetTopCustomers untuk tahun 2008. Pastikan hasilnya sama dengan gambar berikut:

Results Messages

	CustomerID	ContactName	SalesValue
1	20	Kane, John	41210.65
2	63	Veronesi, Giorgio	37217.32
3	71	Navarro, Tomás	36310.11
4	34	Cohen, Shy	23821.20
5	65	Moore, Michael	21238.27
6	37	Cr?ciun, Ovidiu V.	20402.12
7	39	Song, Lolan	19582.78
8	89	Smith Jr., Ronaldo	15278.90

RTM) DESKTOP-47A907C\Zuraid... DailyMart 00:00:00 10 rows

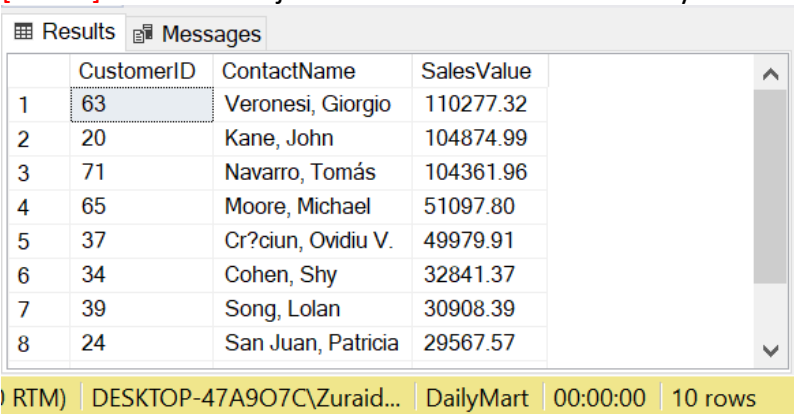
```
--4
ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT
AS
SELECT C.CustomerID,
C.ContactName, SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID
WHERE YEAR(O.OrderDate) = @orderYear GROUP BY C.CustomerID, C.ContactName ORDER BY SalesValue DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

EXECUTE Sales.GetTopCustomers @orderYear = 2008;
```

113 %

Results Messages

	CustomerID	ContactName	SalesValue
1	20	Kane, John	41210.65
2	63	Veronesi, Giorgio	37217.32
3	71	Navarro, Tomás	36310.11
4	34	Cohen, Shy	23821.20
5	65	Moore, Michael	21238.27
6	37	Cr?ciun, Ovidiu V.	20402.12
7	39	Song, Lolan	19582.78
8	89	Smith Jr., Ronaldo	15278.90
9	24	San Juan, Patricia	13644.07
10	76	Gulbis, Katrin	11644.60

4	<p>[Soal-5] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure Sales.GetTopCustomers <i>tanpa parameter</i></p> <p><i>EXECUTE Sales.GetTopCustomers;</i></p>
5	<p>[Soal-6] Bagaimana output dari soal-5? Jika terdapat error, tuliskan pesan errornya</p> <p>Procedure or function 'GetTopCustomers' expects parameter '@orderYear', which was not supplied.</p>
6	<p>Run script berikut untuk memodifikasi stored procedure Sales.GetTopCustomers:</p> <pre> ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT = NULL AS SELECT C.CustomerID, C.ContactName, SUM(O.Value) AS SalesValue FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID WHERE YEAR(O.OrderDate) = @orderYear OR @orderYear IS NULL GROUP BY C.CustomerID, C.ContactName ORDER BY SalesValue DESC OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY; </pre>
7	<p>[Soal-7] Run kembali jawaban soal-5. Pastikan hasilnya sesuai dengan gambar berikut:</p> 



```
--7
ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT = NULL
AS
SELECT C.CustomerID,
C.ContactName, SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID
WHERE YEAR(O.OrderDate) = @orderYear OR @orderYear IS NULL GROUP BY C.CustomerID, C.ContactName
ORDER BY SalesValue DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;
EXECUTE Sales.GetTopCustomers;
```

113 %

Results Messages

	CustomerID	ContactName	SalesValue
1	63	Veronesi, Giorgio	110277.32
2	20	Kane, John	104874.99
3	71	Navarro, Tomás	104361.96
4	65	Moore, Michael	51097.80
5	37	Cr?ciun, Ovidiu V.	49979.91
6	34	Cohen, Shy	32841.37
7	39	Song, Lolan	30908.39
8	24	San Juan, Patricia	29567.57
9	51	Taylor, Maurice	28872.19
10	89	Smith Jr., Ronaldo	27363.61

8

[Soal-8] Jika stored procedure dimodifikasi dengan menambahkan parameter, apakah command untuk mengeksekusi stored procedure tersebut perlu dimodifikasi juga?

Jika parameter ditambahkan dengan nilai default, command untuk mengeksekusi procedure tidak perlu dimodifikasi kecuali untuk menggunakan parameter baru tersebut.

9

Tambahkan parameter @n ke stored procedure Sales.GetTopCustomers. Parameter tersebut digunakan untuk menentukan berapa banyak pelanggan yang akan diambil. Nilai defaultnya adalah 10. Lakukan dengan mengeksekusi script berikut:

```
ALTER PROCEDURE Sales.GetTopCustomers
@orderYear INT = NULL,
@n INT = 10
AS
SELECT C.CustomerID,
C.ContactName,
SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O
INNER JOIN Sales.Customers AS C
```



```
ON O.CustomerID = C.CustomerID
WHERE YEAR(O.OrderDate) = @orderYear OR @orderYear IS NULL
GROUP BY C.CustomerID, C.ContactName
ORDER BY SalesValue DESC
OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY;
```

10

[Soal-9] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure Sales.GetTopCustomers *tanpa parameter*. Pastikan hasilnya sesuai dengan gambar berikut:

Results		Messages	
	CustomerID	ContactName	SalesValue
1	63	Veronesi, Giorgio	110277.32
2	20	Kane, John	104874.99
3	71	Navarro, Tomás	104361.96
4	65	Moore, Michael	51097.80
5	37	Cr?ciun, Ovidiu V.	49979.91
6	34	Cohen, Shy	32841.37
7	39	Song, Lolan	30908.39
8	24	San Juan, Patricia	29567.57
9	51	Taylor, Maurice	28872.19
10	89	Smith Jr., Ronal...	27363.61

RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 10 rows

--9			
ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT = NULL,			
@n INT = 10 AS			
SELECT C.CustomerID,			
C.ContactName, SUM(O.Value) AS SalesValue			
FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C			
ON O.CustomerID = C.CustomerID			
WHERE YEAR(O.OrderDate) = @orderYear OR @orderYear IS NULL GROUP BY C.CustomerID, C.ContactName			
ORDER BY SalesValue DESC			
OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY;			
EXECUTE Sales.GetTopCustomers;			
113 %			
Results		Messages	
	CustomerID	ContactName	SalesValue
1	63	Veronesi, Giorgio	110277.32
2	20	Kane, John	104874.99
3	71	Navarro, Tomás	104361.96
4	65	Moore, Michael	51097.80
5	37	Cr?ciun, Ovidiu V.	49979.91
6	34	Cohen, Shy	32841.37
7	39	Song, Lolan	30908.39
8	24	San Juan, Patricia	29567.57
9	51	Taylor, Maurice	28872.19
10	89	Smith Jr., Ronaldo	27363.61

**11**

[Soal-10] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure Sales.GetTopCustomers agar menampilkan 20 pelanggan dengan pembelian tertinggi. Pastikan hasilnya sesuai dengan gambar berikut:

Results		Messages	
	CustomerID	ContactName	SalesValue
1	63	Veronesi, Giorgio	110277.32
2	20	Kane, John	104874.99
3	71	Navarro, Tomás	104361.96
4	65	Moore, Michael	51097.80
5	37	Cr?ciun, Ovidiu V.	49979.91
6	34	Cohen, Shy	32841.37
7	39	Song, Lolan	30908.39
8	24	San Juan, Patricia	29567.57
9	51	Taylor, Maurice	28872.19
10	89	Smith Jr., Ronaldo	27363.61

RTM | DESKTOP-47A907C\Zuraid... | DailyMart | 00:00:00 | 20 rows

```
--10
ALTER PROCEDURE Sales.GetTopCustomers @orderYear INT = NULL,
@n INT = 10 AS
SELECT C.CustomerID,
C.ContactName, SUM(O.Value) AS SalesValue
FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C
ON O.CustomerID = C.CustomerID
WHERE YEAR(O.OrderDate) = @orderYear OR @orderYear IS NULL GROUP BY C.CustomerID, C.ContactName
ORDER BY SalesValue DESC
OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY;
EXECUTE Sales.GetTopCustomers @n = 20;
```

100 %

Results		Messages	
	CustomerID	ContactName	SalesValue
3	71	Navarro, Tomás	104361.96
4	65	Moore, Michael	51097.80
5	37	Cr?ciun, Ovidiu V.	49979.91
6	34	Cohen, Shy	32841.37
7	39	Song, Lolan	30908.39
8	24	San Juan, Patricia	29567.57
9	51	Taylor, Maurice	28872.19
10	89	Smith Jr., Ronaldo	27363.61
11	25	Carlson, Jason	26656.57
12	62	Misiec, Anna	25717.50
13	5	Higginbotham, Tom	24927.58
14	76	Gulbis, Katrin	24088.78
15	59	Meston, Tosh	23128.86
16	35	Langohr, Kris	22768.76
17	9	Raghav, Amritansh	21963.25
18	10	Bassols, Pilar Colome	20801.61
19	68	Myrcha, Jacek	19343.78
20	44	Louverdis, George	19261.42

✓ Query executed successfully.

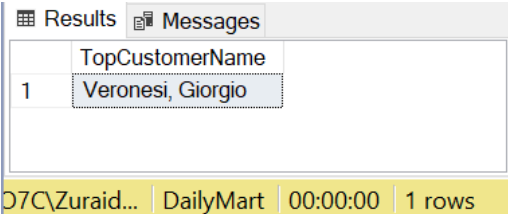
LAPTOP-5EG3DBVG\SMOKY (14.0... | LAPTOP-5EG3DBVG\SMOKY (14.0...)

**12**

Run script berikut untuk memodifikasi stored procedure Sales.GetTopCustomers agar mengembalikan nama customer berdasarkan posisi tertentu dalam peringkat pembelian terbanyak. Gunakan parameter @customerPosition input dan parameter @customerName sebagai output.

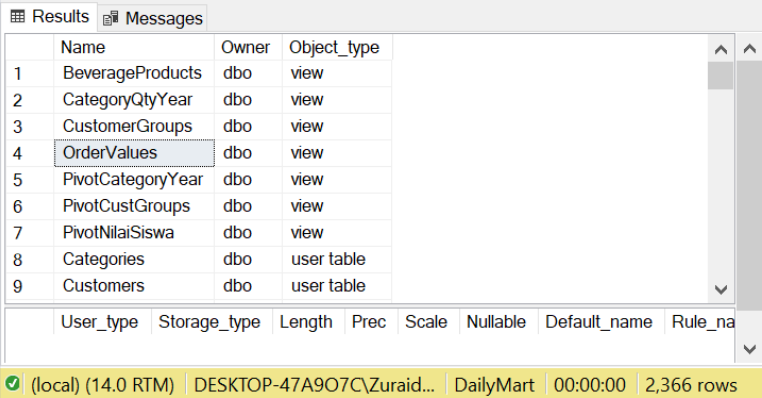
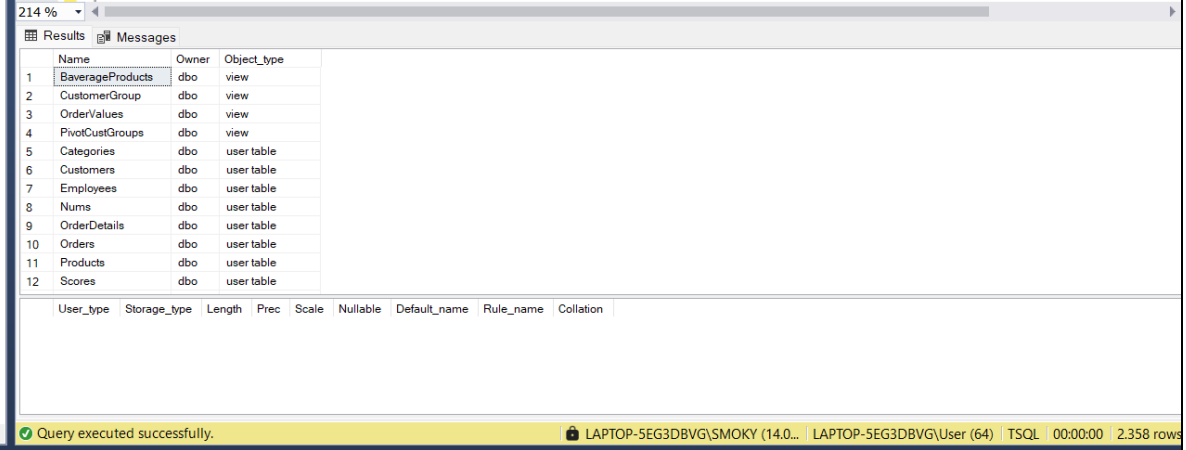
```
ALTER PROCEDURE Sales.GetTopCustomers  
@customerPosition INT = 1,  
@customerName NVARCHAR(30) OUTPUT
```



	<pre>AS SET @customerName = (SELECT C.ContactName FROM Sales.OrderValues AS O INNER JOIN Sales.Customers AS C ON O.CustomerID = C.CustomerID GROUP BY C.CustomerID, C.ContactName ORDER BY SUM(O.Value) DESC OFFSET @customerPosition - 1 ROWS FETCH NEXT 1 ROWS ONLY)</pre>
13	<p>Deklarasikan variabel baru @outCustomerName. Variabel ini digunakan untuk menampung output parameter dari stored procedure Sales.GetTopCustomers.</p> <pre>DECLARE @outCustomerName nvarchar(30);</pre>
14	<p>[Soal-11] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure Sales.GetTopCustomers untuk memperoleh nama customer di posisi teratas. Simpan nilai output parameter customerName ke dalam variable outCustomerName. Selanjutnya tampilkan nilai variable outCustomerName dengan statement SELECT.</p> <pre>-- Deklarasi variabel output DECLARE @customerName NVARCHAR(30); -- Eksekusi stored procedure dengan parameter input dan output EXECUTE Sales.GetTopCustomers @customerPosition = 1, @customerName = @customerName OUTPUT; -- Tampilkan hasil variabel output SELECT @customerName AS CustomerName;</pre>
15	<p>Run DECLARE, EXECUTE, dan SELECT statement pada langkah 13 dan 14 kemudian bandingkan hasilnya dengan gambar berikut:</p> 



Praktikum – Bagian 3: Executing System Stored Procedures

Langkah	Keterangan
1	<p>Skenario:</p> <p>Praktikum pada bagian ini akan mempelajari bagaimana menjalankan beberapa system stored procedure yang sering digunakan untuk memperoleh informasi terkait database.</p>
2	<p>[Soal-12] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure bernama sys.sp_help tanpa parameter dengan hasil seperti <u>contoh</u> di bawah ini. Apa kegunaan stored procedure tersebut?</p> <div><pre>--12 EXEC sys.sp_help;</pre></div> <p>Memberikan deskripsi khusus untuk objek tertentu, seperti tabel atau view.</p>



3

[Soal-13] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure sys.sp_help khusus untuk tabel Sales.Customer. Apa kegunaan stored procedure tersebut jika nama table digunakan sebagai nilai parameter?

Results Messages

Name	Owner	Type	Created_datetime
Customers	dbo	user table	2024-08-30 13:32:13.617

Column_name	Type	Computed	Length	Prec	Scale	Nullable	1
CustomerID	int	no	4	10	0	no	(
CompanyName	nvarchar	no	80			no	(
ContactName	nvarchar	no	60			no	(
ContactTitle	nvarchar	no	60			no	(
Address	nvarchar	no	120			no	(
City	nvarchar	no	30			no	(
Region	nvarchar	no	30			yes	(
PostalCode	nvarchar	no	20			yes	(

< >

Identity	Seed	Increment	Not For Replication
CustomerID	1	1	0

RowGuidCol
No rowguidcol column defined.

Data_located_on_filegroup
PRIMARY

index_name	index_description	index_keys
idx_nc_city	nonclustered located on PRIMARY	City
idx_nc_companyname	nonclustered located on PRIMARY	Company...

(local) (14.0 RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 22 rows

--13

```
EXEC sys.sp_help '[Sales].[Customers]';
```

214 %

Results Messages

Name	Owner	Type	Created_datetime						
Customers	dbo	user table	2024-08-30 13:32:13.617						
Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
CustomerID	int	no	4	10	0	no	(n/a)	(n/a)	NULL
CompanyName	nvarchar	no	80			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
ContactName	nvarchar	no	60			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
ContactTitle	nvarchar	no	60			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
Address	nvarchar	no	120			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
City	nvarchar	no	30			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
Region	nvarchar	no	30			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
PostalCode	nvarchar	no	20			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
Identity	Seed	Increment	Not For Replication						
CustomerID	1	1	0						
RowGuidCol									
No rowguidcol column defined.									
Data_located_on_filegroup									
PRIMARY									
index_name	index_description		index_keys						

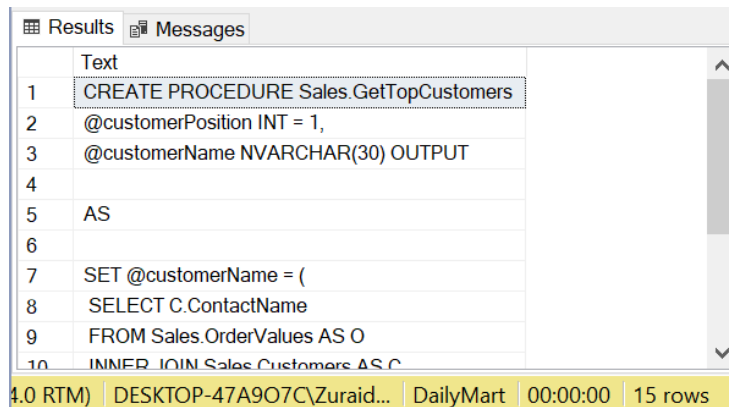
Query executed successfully.

LAPTOP-5EG3DBVG\SMOKY (14.0...LAPTOP-5EG3DBVG\User (64)TSQL00:00:0122 rows

Memberikan deskripsi khusus untuk objek tertentu, seperti tabel atau view.

4

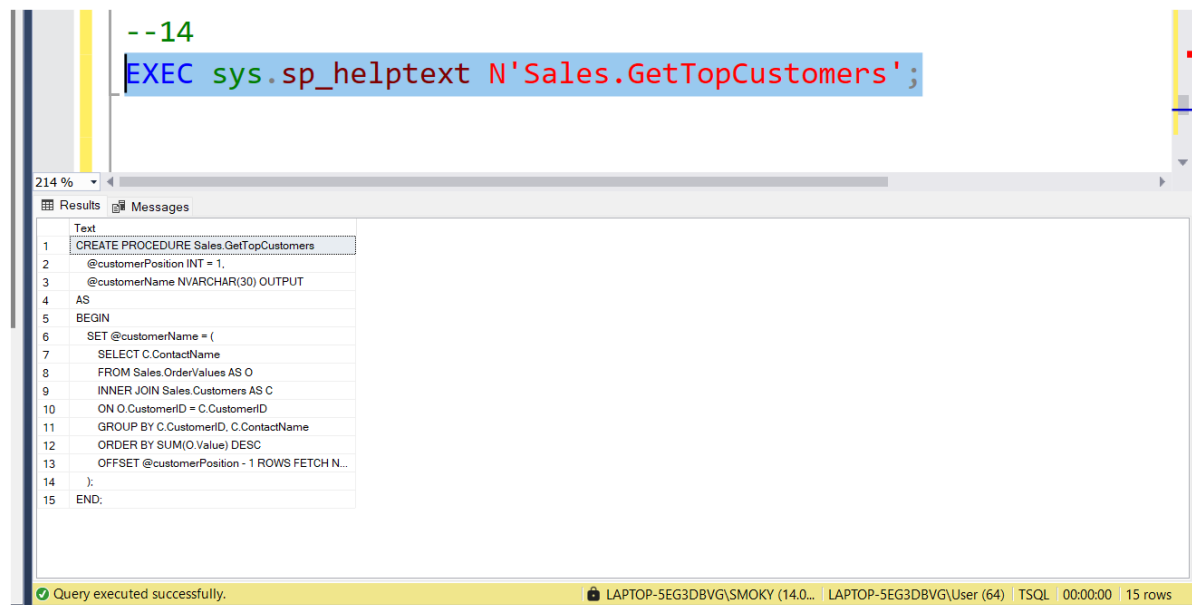
[Soal-14] Tuliskan EXECUTE statement untuk mengeksekusi stored procedure `sys.sp_helptext` dengan `N'Sales.GetTopCustomers'` sebagai nilai parameter. Cek contoh pada gambar berikut. Apa kegunaan stored procedure tersebut?



The screenshot shows the 'Results' pane of SQL Server Enterprise Manager. The 'Text' tab is selected, displaying the definition of the stored procedure `Sales.GetTopCustomers`. The code is as follows:

```
1 CREATE PROCEDURE Sales.GetTopCustomers
2   @customerPosition INT = 1,
3   @customerName NVARCHAR(30) OUTPUT
4
5 AS
6
7 SET @customerName = (
8   SELECT C.ContactName
9   FROM Sales.OrderValues AS O
10  INNER JOIN Sales.Customers AS C
```

The status bar at the bottom indicates: 4.0 RTM) | DESKTOP-47A907C\Zuraid... | DailyMart | 00:00:00 | 15 rows



The screenshot shows the execution of the `sys.sp_helptext` stored procedure. The command window displays the following SQL statement:

```
--14
EXEC sys.sp_helptext N'Sales.GetTopCustomers';
```

The 'Results' pane shows the definition of the stored procedure `Sales.GetTopCustomers` in a scrollable list:

```
1 CREATE PROCEDURE Sales.GetTopCustomers
2   @customerPosition INT = 1,
3   @customerName NVARCHAR(30) OUTPUT
4
5 AS
6 BEGIN
7   SET @customerName = (
8     SELECT C.ContactName
9     FROM Sales.OrderValues AS O
10    INNER JOIN Sales.Customers AS C
11    ON O.CustomerID = C.CustomerID
12    GROUP BY C.CustomerID, C.ContactName
13    ORDER BY SUM(O.Value) DESC
14    OFFSET @customerPosition - 1 ROWS FETCH N...
15  );
16 END;
```

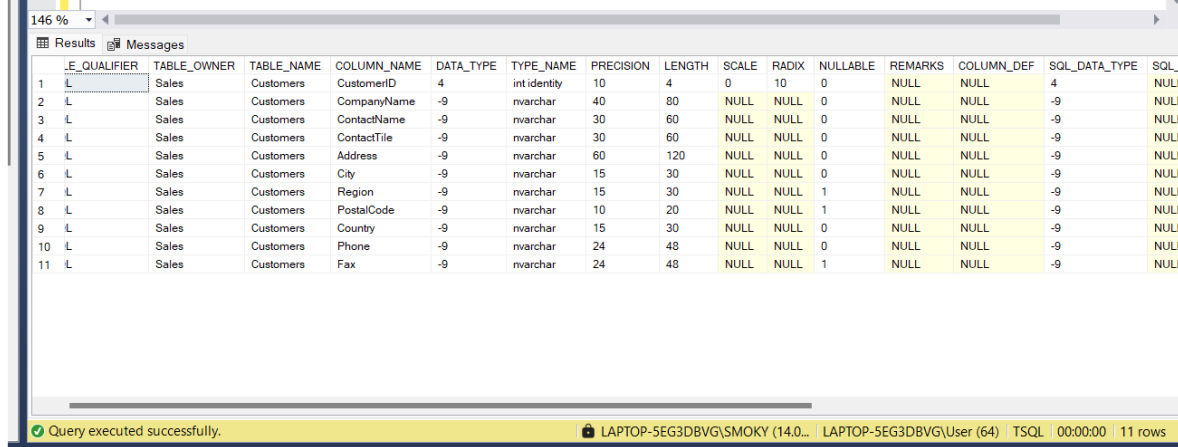
The status bar at the bottom indicates: Query executed successfully. | LAPTOP-5EG3DBVG\SMOKY (14.0... | LAPTOP-5EG3DBVG\User (64) | TSQL | 00:00:00 | 15 rows

Menampilkan definisi teks dari suatu stored procedure.

5

[Soal-15] Tulislah EXECUTE statement untuk mengeksekusi stored procedure sys.sp_columns untuk tabel Sales.Customers. Set parameter @table_name dan @table_owner agar output sesuai dengan contoh gambar di bawah ini. Apa kegunaan stored procedure tersebut?

```
--15
EXEC sys.sp_columns @table_name = 'Customers', @table_owner = 'Sales';
```



LINE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	PRECISION	LENGTH	SCALE	RADIX	NULLABLE	REMARKS	COLUMN_DEF	SQL_DATA_TYPE	SQL_DATETIME
1	L	Sales	Customers	CustomerID	int identity	10	4	0	10	0	NULL	NULL	4	NULL
2	L	Sales	Customers	CompanyName	nvarchar	40	80	NULL	NULL	0	NULL	NULL	-9	NULL
3	L	Sales	Customers	ContactName	nvarchar	30	60	NULL	NULL	0	NULL	NULL	-9	NULL
4	L	Sales	Customers	ContactTitle	nvarchar	30	60	NULL	NULL	0	NULL	NULL	-9	NULL
5	L	Sales	Customers	Address	nvarchar	60	120	NULL	NULL	0	NULL	NULL	-9	NULL
6	L	Sales	Customers	City	nvarchar	15	30	NULL	NULL	0	NULL	NULL	-9	NULL
7	L	Sales	Customers	Region	nvarchar	15	30	NULL	NULL	1	NULL	NULL	-9	NULL
8	L	Sales	Customers	PostalCode	nvarchar	10	20	NULL	NULL	1	NULL	NULL	-9	NULL
9	L	Sales	Customers	Country	nvarchar	15	30	NULL	NULL	0	NULL	NULL	-9	NULL
10	L	Sales	Customers	Phone	nvarchar	24	48	NULL	NULL	0	NULL	NULL	-9	NULL
11	L	Sales	Customers	Fax	nvarchar	24	48	NULL	NULL	1	NULL	NULL	-9	NULL

Query executed successfully. LAPTOP-5EG3DBVG\SMOKY (14.0...) LAPTOP-5EG3DBVG\User (64) TSQL : 00:00:00 | 11 rows

Memberikan informasi detail tentang kolom-kolom pada tabel tertentu.

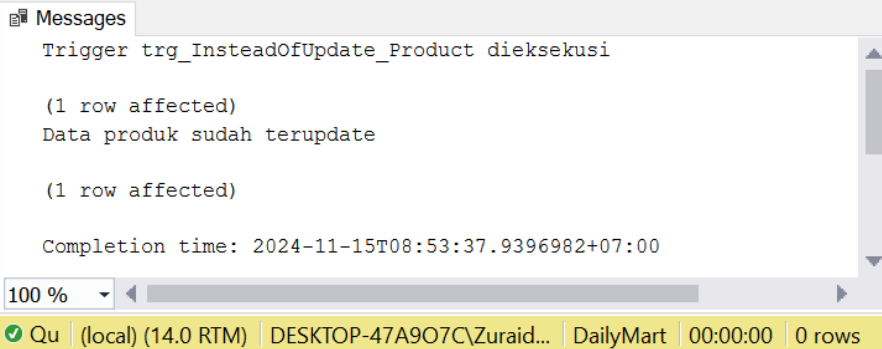
6

Eksekusi T-SQL statement berikut untuk menghapus stored procedure Sales.GetTopCustomers:

DROP PROCEDURE Sales.GetTopCustomers;

**Praktikum – Bagian 4: TRIGGER**

Langkah	Keterangan																																																																								
1	<p>Trigger adalah stored procedure khusus yang secara otomatis dijalankan saat sebuah event terjadi dalam database.</p> <p>Berdasarkan waktu eksekusinya, terdapat 2 jenis TRIGGER:</p> <ul style="list-style-type: none">- INSTEAD OF Trigger: trigger yang menggantikan suatu event- AFTER Trigger: trigger yang dilakukan setelah suatu event																																																																								
2	<p>Tambahkan kolom Stock bertipe INT pada table Production.Products dan beri nilai 100 untuk semua produk.</p> <pre>ALTER TABLE Production.Products ADD Stock INT; GO UPDATE Production.Products SET Stock = 100;</pre> <div><div>ResultsMessages</div><table><tr><th></th><th>ProductID</th><th>ProductName</th><th>SupplierID</th><th>CategoryID</th><th>UnitPrice</th><th>Discontinued</th><th>Stock</th><th></th></tr><tr><td>1</td><td>1</td><td>Product HHYDP</td><td>1</td><td>1</td><td>18.00</td><td>0</td><td>100</td><td></td></tr><tr><td>2</td><td>2</td><td>Product RECZE</td><td>1</td><td>1</td><td>19.00</td><td>0</td><td>100</td><td></td></tr><tr><td>3</td><td>3</td><td>Product IMEHJ</td><td>1</td><td>2</td><td>10.00</td><td>0</td><td>100</td><td></td></tr><tr><td>4</td><td>4</td><td>Product KSB RM</td><td>2</td><td>2</td><td>22.00</td><td>0</td><td>100</td><td></td></tr><tr><td>5</td><td>5</td><td>Product EPEIM</td><td>2</td><td>2</td><td>21.35</td><td>1</td><td>100</td><td></td></tr><tr><td>6</td><td>6</td><td>Product VAIIV</td><td>3</td><td>2</td><td>25.00</td><td>0</td><td>100</td><td></td></tr><tr><td>7</td><td>7</td><td>Product HMLNI</td><td>3</td><td>7</td><td>30.00</td><td>0</td><td>100</td><td></td></tr></table><div>Que... (local) (14.0 RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 77 rows</div></div>		ProductID	ProductName	SupplierID	CategoryID	UnitPrice	Discontinued	Stock		1	1	Product HHYDP	1	1	18.00	0	100		2	2	Product RECZE	1	1	19.00	0	100		3	3	Product IMEHJ	1	2	10.00	0	100		4	4	Product KSB RM	2	2	22.00	0	100		5	5	Product EPEIM	2	2	21.35	1	100		6	6	Product VAIIV	3	2	25.00	0	100		7	7	Product HMLNI	3	7	30.00	0	100	
	ProductID	ProductName	SupplierID	CategoryID	UnitPrice	Discontinued	Stock																																																																		
1	1	Product HHYDP	1	1	18.00	0	100																																																																		
2	2	Product RECZE	1	1	19.00	0	100																																																																		
3	3	Product IMEHJ	1	2	10.00	0	100																																																																		
4	4	Product KSB RM	2	2	22.00	0	100																																																																		
5	5	Product EPEIM	2	2	21.35	1	100																																																																		
6	6	Product VAIIV	3	2	25.00	0	100																																																																		
7	7	Product HMLNI	3	7	30.00	0	100																																																																		
3	<p>Kasus: Jika terdapat produk yang Discontinued-nya bernilai 1 (true), maka Stock hanya boleh berkurang (dijual), tidak boleh bertambah lagi (disupply).</p> <p>Buatlah INSTEAD OF UPDATE trigger pada table Production.Products dengan mengeksekusi script berikut.</p> <pre>CREATE TRIGGER trg_InsteadOfUpdate_Product ON Production.Products INSTEAD OF UPDATE AS BEGIN PRINT 'Trigger trg_InsteadOfUpdate_Product dieksekusi'; DECLARE @productId INT; DECLARE @discontinued BIT; DECLARE @newStock INT; DECLARE @oldStock INT; SELECT @productId = ProductID, @discontinued = Discontinued, @newstock = Stock FROM inserted; SELECT @oldStock = Stock FROM Production.Products WHERE ProductID = @productId;</pre>																																																																								

	<pre> IF @discontinued = 1 AND @newStock > @oldStock BEGIN PRINT 'Produk sudah discontinued, tidak dapat disupply kembali.'; END ELSE BEGIN UPDATE Production.Products SET Stock = @newStock WHERE ProductID = @productId; PRINT 'Data produk sudah terupdate'; END END </pre>
4	<p>Lakukan percobaan untuk mengupdate jumlah stock salah satu produk yang discontinued, misalnya produk dengan ID = 5</p> <pre> UPDATE Production.Products SET UnitPrice = 25, Stock = 120 WHERE ProductID = 5 </pre> <p>Trigger akan dieksekusi dengan output sebagai berikut:</p> 
5	<p>Lakukan percobaan untuk mengupdate jumlah stock salah satu produk yang tidak discontinued, misalnya produk dengan ID = 4</p> <pre> UPDATE Production.Products SET UnitPrice = 25, Stock = 120 WHERE ProductID = 4 </pre> <p>Trigger akan dieksekusi dengan mengubah nilai stock menjadi 120</p> 
6	<p>[Soal-16] Pada langkah no 5, mengapa nilai UnitPrice untuk produkID 4 tidak terupdate? Modifikasi trigger agar semua kolom dapat terupdate dengan benar.</p> <p>Karena trigger hanya memproses kolom Stock, sehingga perubahan pada UnitPrice</p>



	diabaikan. Untuk memperbaikinya, trigger harus mencakup semua kolom.
--	--



[Soal-17] Tuliskan dan jalankan command untuk men-disable TRIGGER yang dibuat pada langkah 3

```
--17
DISABLE TRIGGER trg_InsteadOfUpdate_Product ON Production.Products;
```

100 %

Messages

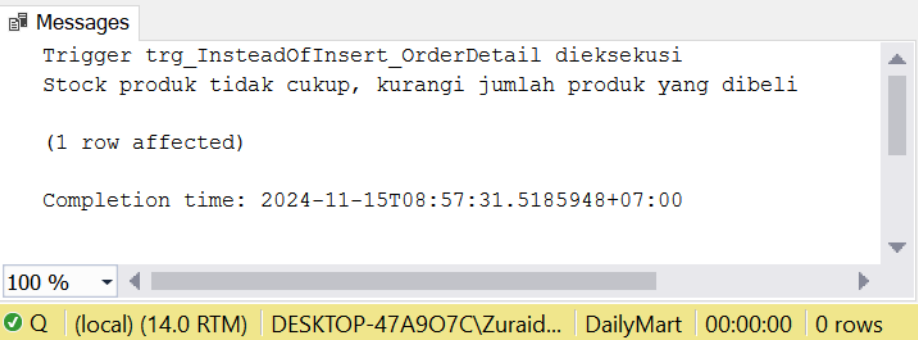
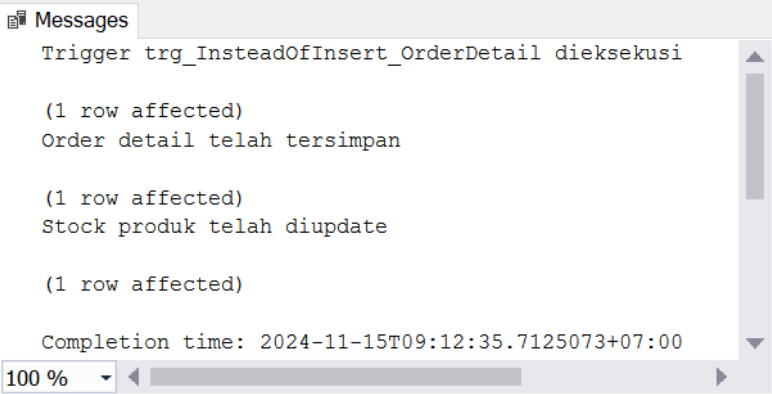
Commands completed successfully.

Completion time: 2024-11-16T03:24:06.3425616+07:00



7	<p>Kasus: Jika terdapat OrderDetail baru, periksa apakah Quantity nya lebih besar daripada Stock. Jika iya, batalkan insertion. Jika tidak, insert OrderDetail tersebut dan update stock produk pada table Production.Products. Print pesan untuk kedua kondisi.</p> <p>Buatlah INSTEAD OF INSERT trigger untuk table Sales.OrderDetails dengan mengeksekusi script berikut:</p> <pre>CREATE TRIGGER trg_InsteadOfInsert_OrderDetail ON Sales.OrderDetails INSTEAD OF INSERT AS BEGIN PRINT 'Trigger trg_InsteadOfInsert_OrderDetail dieksekusi'; DECLARE @productId INT DECLARE @quantity INT; DECLARE @stock INT; SELECT @productId = ProductID, @quantity = Quantity FROM inserted; SELECT @stock = Stock FROM Production.Products WHERE ProductID = @productId; IF @stock < @quantity BEGIN PRINT 'Stock produk tidak cukup, kurangi jumlah produk yang dibeli'; END ELSE BEGIN INSERT INTO Sales.OrderDetails (OrderID, ProductID, UnitPrice, Quantity, Discount) SELECT OrderID, ProductID, UnitPrice, Quantity, Discount FROM inserted; PRINT 'Order detail telah tersimpan'; UPDATE Production.Products SET Stock = Stock - @quantity WHERE ProductID = @productId; PRINT 'Stock produk telah diupdate'; END END</pre>
8	Lakukan percobaan dengan insert baris data baru pada table Sales.OrderDetails seperti <u>contoh</u> berikut:



	<p><code>INSERT INTO Sales.OrderDetails (OrderID, ProductID, UnitPrice, Quantity, Discount)</code> <code>VALUES (10625, 1, 18, 150, 0)</code></p> <p>Execute SQL command maka akan muncul warning karena stock ProductID = 1 hanya 100.</p> 
9	<p>Lakukan percobaan insert lain dengan quantity lebih kecil dari stock, <u>misalnya</u>:</p> <p><code>INSERT INTO Sales.OrderDetails (OrderID, ProductID, UnitPrice, Quantity, Discount)</code> <code>VALUES (10625, 1, 18, 50, 0)</code></p> <p>Output yang muncul adalah pesan berikut:</p> 

Baris baru telah masuk ke dalam table Sales.OrderDetails dan nilai Stock untuk ProductID = 1 sudah berkurang menjadi 50.



10

[Soal-18] Kasus: Jika terdapat penghapusan OrderDetail, tambahkan kembali jumlah stock produk tersebut pada table Production.Products. Buatlah **AFTER DELETE TRIGGER** untuk table Sales.OrderDetails.

Uji trigger yang dibuat dengan menjalankan SQL command berikut:

DELETE FROM Sales.OrderDetails **WHERE** OrderID = 10625 **AND** ProductID = 1

Output yang muncul sebagai berikut:

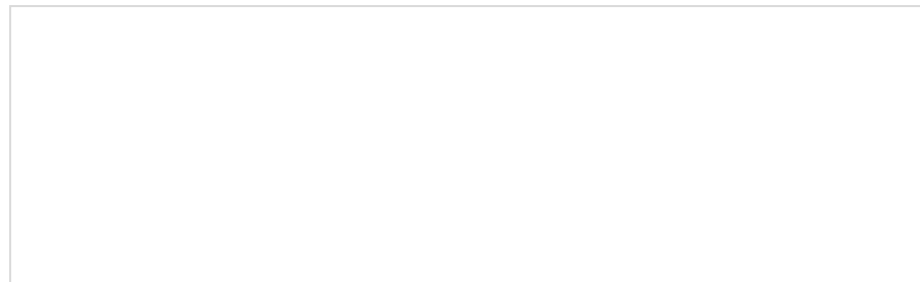
```
Messages
Trigger trg_AfterDelete_OrderDetail dieksekusi

(1 row affected)
Order detail sudah terhapus dan stok produk telah ditambahkan kembali

(1 row affected)

Completion time: 2024-11-15T08:33:46.0816789+07:00
100 %
Query ex... (local) (14.0 RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 0 rows
```

Perhatikan bahwa stock ProductID 1 seharusnya kembali menjadi 100



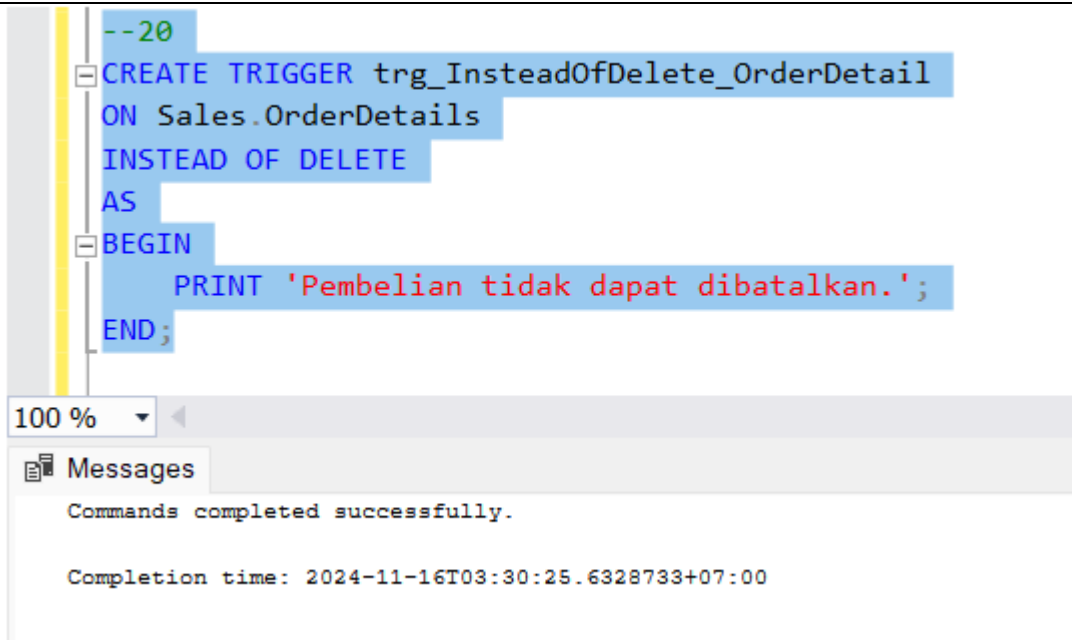
Catatan:

Uji coba di atas hanya contoh, silakan menggunakan OrderDetail dan Produk lain sebagai data uji.



	<pre>--18 CREATE TRIGGER trg_AfterDelete_OrderDetail ON Sales.OrderDetails AFTER DELETE AS BEGIN DECLARE @productId INT, @quantity INT; SELECT @productId = ProductID, @quantity = Quantity FROM deleted; UPDATE Production.Products SET Stock = Stock + @quantity WHERE ProductID = @productId; PRINT 'Stock produk telah dikembalikan.'; END;</pre> <p>100 %</p> <p>Messages</p> <p>Msg 2714, Level 16, State 2, Procedure trg_AfterDelete_OrderDetail, Line 2 [Batch Start Line 179] There is already an object named 'trg_AfterDelete_OrderDetail' in the database.</p> <p>Completion time: 2024-11-16T03:27:32.6195054+07:00</p>
11	<p>[Soal-19] Tuliskan dan jalankan command untuk men-disable TRIGGER yang dibuat pada soal-18</p> <pre>--19 DISABLE TRIGGER trg_AfterDelete_OrderDetail ON Sales.OrderDetails;</pre> <p>100 %</p> <p>Messages</p> <p>Commands completed successfully.</p> <p>Completion time: 2024-11-16T03:29:54.1420935+07:00</p>
12	<p>[Soal-20] Buatlah INSTEAD OF DELETE TRIGGER untuk menghalangi penghapusan baris pada table Sales.OrderDetail. Print pesan bahwa pembelian yang sudah dilakukan tidak dapat dibatalkan.</p>



	
13	<p>[Soal-21] Cobalah menghapus 1 baris data pada table Sales.OrderDetail.</p> <pre>DELETE FROM Sales.OrderDetails WHERE OrderID = 10625 AND ProductID = 14</pre> <p>Bandingkan outputnya dengan gambar berikut</p>



	<div><div>Messages</div><div>Trigger trg_InsteadOfDelete_OrderDetail dieksekusi Order detail tidak dapat dihapus Pembelian yang sudah dilakukan tidak dapat dibatalkan (1 row affected) Completion time: 2024-11-15T08:39:58.9078108+07:00 100 % Q... (local) (14.0 RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 0 rows</div></div> <div><div>--21</div><div>DELETE FROM Sales.OrderDetails WHERE OrderID = 10625 AND ProductID = 14</div><div>100 %</div><div>Messages</div><div>Pembelian tidak dapat dibatalkan. (1 row affected) Completion time: 2024-11-16T03:31:17.2927817+07:00</div></div>
14	<p>[Soal-22] Apa perbedaan INSTEAD OF TRIGGER dan AFTER TRIGGER? Perbedaan INSTEAD OF Trigger dan AFTER Trigger:</p> <p>INSTEAD OF Trigger: Menggantikan operasi asli, mencegah eksekusi event utama (misalnya DELETE tidak dilakukan sama sekali).</p> <p>AFTER Trigger: Dieksekusi setelah event utama selesai diproses (misalnya, setelah baris dihapus, trigger berjalan).</p>
15	<p>[Soal-23] Tuliskan dan jalankan command untuk menghapus trigger yang sudah dibuat pada table Sales.OrderDetail</p>



--23

DROP TRIGGER IF EXISTS trg_AfterDelete_OrderDetail;

.00 %

Messages

Commands completed successfully.

Completion time: 2024-11-16T03:33:57.8258542+07:00

--- Selamat Mengerjakan ----