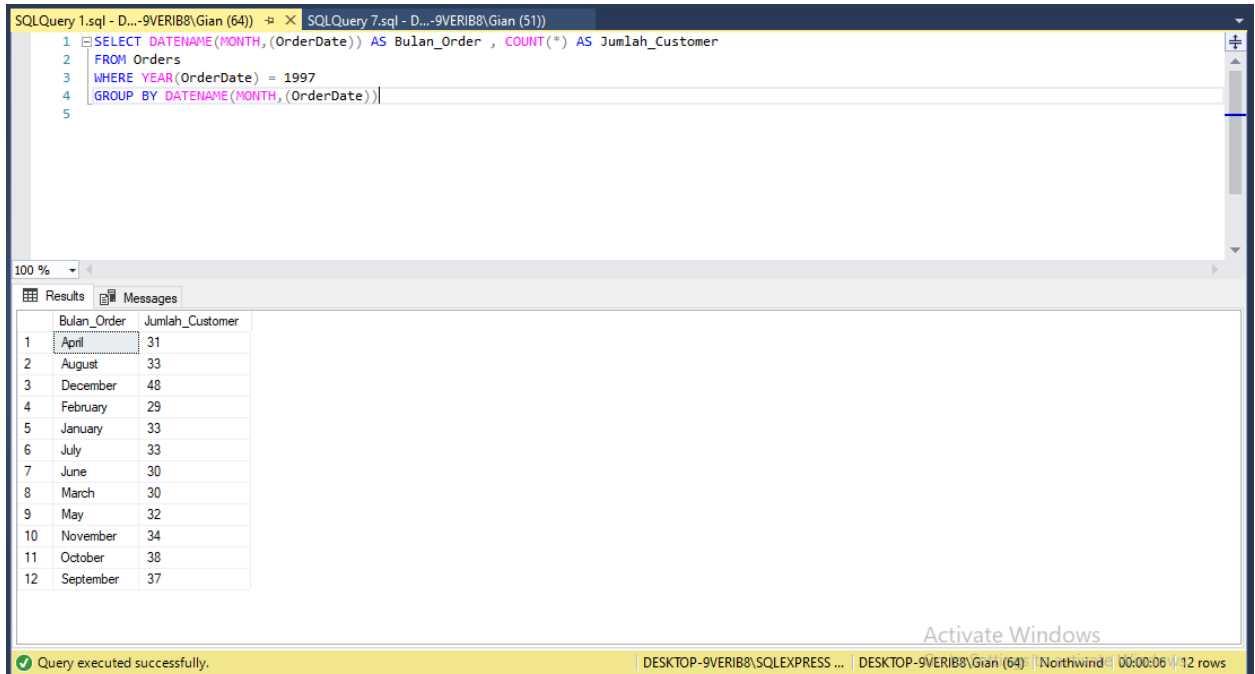


## Bagian 2a Transform : Intermediate Queries

1. Tulis query untuk mendapatkan jumlah customer tiap bulan yang melakukan order pada tahun 1997.



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

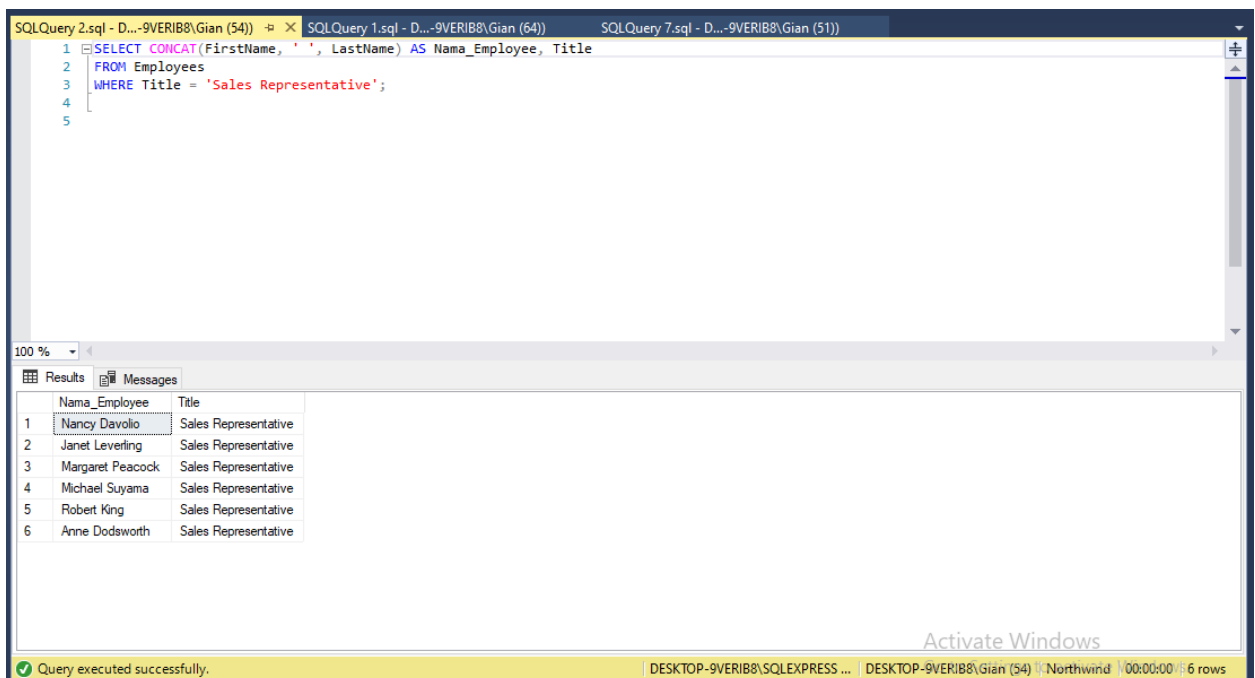
```
1 SELECT DATENAME(MONTH, OrderDate) AS Bulan_Order, COUNT(*) AS Jumlah_Customer
2 FROM Orders
3 WHERE YEAR(OrderDate) = 1997
4 GROUP BY DATENAME(MONTH, OrderDate);
5
```

The query results are displayed in a table with two columns: Bulan\_Order and Jumlah\_Customer. The results are as follows:

Bulan_Order	Jumlah_Customer
April	31
August	33
December	48
February	29
January	33
July	33
June	30
March	30
May	32
November	34
October	38
September	37

The status bar at the bottom indicates: Query executed successfully. DESKTOP-9VERIB8\SQLEXPRESS ... DESKTOP-9VERIB8\Gian (64) | Northwind | 00:00:06 | 12 rows

2. Tulis query untuk mendapatkan nama employee yang termasuk Sales Representative.



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

```
1 SELECT CONCAT(FirstName, ' ', LastName) AS Nama_Employee, Title
2 FROM Employees
3 WHERE Title = 'Sales Representative';
4
5
```

The query results are displayed in a table with two columns: Nama\_Employee and Title. The results are as follows:

Nama_Employee	Title
Nancy Davolio	Sales Representative
Janet Leverling	Sales Representative
Margaret Peacock	Sales Representative
Michael Suyama	Sales Representative
Robert King	Sales Representative
Anne Dodsworth	Sales Representative

The status bar at the bottom indicates: Query executed successfully. DESKTOP-9VERIB8\SQLEXPRESS ... DESKTOP-9VERIB8\Gian (64) | Northwind | 00:00:00 | 6 rows

3. Tulis query untuk mendapatkan top 5 nama produk yang quantitynya paling banyak diorder pada bulan Januari 1997.

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

```
1 SELECT P.ProductName, SUM(P.UnitsOnOrder) AS UnitsOnOrder
2 FROM Products AS P
3 INNER JOIN [Order Details] AS OD
4 ON P.ProductID = OD.ProductID
5 INNER JOIN Orders AS O
6 ON OD.OrderID = O.OrderID
7 WHERE (MONTH(O.OrderDate) = 1 AND YEAR(O.OrderDate) = 1997)
8 GROUP BY P.ProductName
9 ORDER BY SUM(P.UnitsOnOrder) DESC
10 OFFSET 0 ROWS FETCH FIRST 5 ROWS ONLY
11
```

The results pane shows the following data:

	ProductName	UnitsOnOrder
1	Maxilaku	120
2	Sir Rodney's Scones	120
3	Wimmers gute Semmelknödel	80
4	Aniseed Syrup	70
5	Chocolate	70

Query executed successfully. DESKTOP-9VERIB8\SQLEXPRESS ... DESKTOP-9VERIB8\Gian (55) Northwind 00:00:00 5 rows

4. Tulis query untuk mendapatkan nama company yang melakukan order Chai pada bulan Juni 1997.

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

```
1 SELECT C.CompanyName, P.ProductName
2 FROM Customers AS C
3 INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
4 INNER JOIN [Order Details] AS OD ON O.OrderID = OD.OrderID
5 INNER JOIN Products AS P ON OD.ProductID = P.ProductID
6 WHERE (P.ProductName = 'Chai') AND (MONTH(O.OrderDate) = 6 AND YEAR(O.OrderDate) = 1997)
7
```

The results pane shows the following data:

	CompanyName	ProductName
1	Tortuga Restaurante	Chai

Query executed successfully. DESKTOP-9VERIB8\SQLEXPRESS ... DESKTOP-9VERIB8\Gian (56) Northwind 00:00:00 1 rows

5. Tulis query untuk mendapatkan jumlah OrderID yang pernah melakukan sales (unit\_price dikali quantity)  $\leq 100$ ,  $100 < x \leq 250$ ,  $250 < x \leq 500$ , dan  $> 500$ .

```
SQLQuery 5.sql - D:\...-9VERIB8\Gian (57)  SQLQuery 4.sql - D:\...-9VERIB8\Gian (56)  SQLQuery 3.sql - D:\...-9VERIB8\Gian (55)  SQLQuery 2.sql - D:\...-9VERIB8\Gian (54)

1 SELECT OD.OrderID, SUM(OD.UnitPrice * OD.Quantity) AS Sales
2 FROM [Order Details] AS OD
3 INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
4 GROUP BY OD.OrderID
5 HAVING SUM(OD.UnitPrice * OD.Quantity) <= 100;
6
7 SELECT OD.OrderID, SUM(OD.UnitPrice * OD.Quantity) AS Sales
8 FROM [Order Details] AS OD
9 INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
10 GROUP BY OD.OrderID
11 HAVING SUM(OD.UnitPrice * OD.Quantity) BETWEEN 100 AND 250;
12
13 SELECT OD.OrderID, SUM(OD.UnitPrice * OD.Quantity) AS Sales
14 FROM [Order Details] AS OD
15 INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
16 GROUP BY OD.OrderID
17 HAVING SUM(OD.UnitPrice * OD.Quantity) BETWEEN 250 AND 500;
18
19 SELECT OD.OrderID, SUM(OD.UnitPrice * OD.Quantity) AS Sales
20 FROM [Order Details] AS OD
21 INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
22 GROUP BY OD.OrderID
23 HAVING SUM(OD.UnitPrice * OD.Quantity) > 500;
```

SQLQuery 5.sql - D:\...-9VERIB8\Gian (57) SQLQuery 4.sql - D:\...-9VERIB8\Gian (56) SQLQuery 3.sql - D:\...-9VERIB8\Gian (55) SQLQuery 2.sql - D:\...-9VERIB8\Gian (54)

100 %

Results Messages

	OrderID	Sales
1	10271	48.00
2	10281	86.50
3	10288	89.00
4	10308	88.80
5	10331	88.50

	OrderID	Sales
1	10259	100.80
2	10282	155.40
3	10295	121.60
4	10313	182.40
5	10318	240.40

	OrderID	Sales
1	10248	440.00
2	10261	448.00
3	10266	364.80
4	10275	307.20
5	10276	420.00

	OrderID	Sales
1	10249	1863.40
2	10250	1813.00
3	10251	670.80
4	10252	3730.00
5	10253	1444.80
6	10254	625.20

Activate Windows

Query executed successfully. DESKTOP-9VERIB8\SQLEXPRESS ... DESKTOP-9VERIB8\Gian (57) Northwind 00:00:00 830 rows

6. Tulis query untuk mendapatkan Company name yang melakukan sales di atas 500 pada tahun 1997.

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

```
1 SELECT C.CompanyName, SUM(OD.UnitPrice * OD.Quantity) AS Sales
2 FROM Customers AS C
3 INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
4 INNER JOIN [Order Details] AS OD ON O.OrderID = OD.OrderID
5 WHERE YEAR(O.OrderDate) = 1997
6 GROUP BY C.CompanyName
7 HAVING SUM(OD.UnitPrice * OD.Quantity) > 500
8
```

The results window displays a table with two columns: CompanyName and Sales. The data is as follows:

CompanyName	Sales
Vaffeljernet	9710,50
Ottiles K�seladen	8398,60
Sant� Gourmet	700,00
Let's Stop N Shop	2039,42
Hungry Coyote Import Store	2283,20
Morgenstem Gesundkost	3596,40
HILARION-Abastos	14026,18
Ricardo Adocicados	4338,90
Pericles Comidas cl�sicas	2065,40
Wartian Herkku	13106,30
Rancho grande	1149,40
Lehmanns Marktstand	14433,17
Queen Cozinha	10937,25
Island Trading	2560,50
Blauer See Delikatessen	1079,80
Folies gourmandes	11666,90
QUICK-Stop	64238,00

The status bar at the bottom indicates "Query executed successfully." and "75 rows".

7. Tulis query untuk mendapatkan nama produk yang merupakan Top 5 sales tertinggi tiap bulan di tahun 1997.

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

```
1 SELECT P.ProductName, SUM(OD.UnitPrice * OD.Quantity) AS Sales
2 FROM Products AS P
3 INNER JOIN [Order Details] AS OD ON P.ProductID = OD.ProductID
4 INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
5 WHERE YEAR(O.OrderDate) = 1997
6 GROUP BY P.ProductName
7 ORDER BY SUM(OD.UnitPrice * OD.Quantity) DESC
8 OFFSET 0 ROWS FETCH FIRST 5 ROWS ONLY
9
```

The results window displays a table with two columns: ProductName and Sales. The data is as follows:

ProductName	Sales
C�te de Blaye	51962,20
Raclette Courdavault	37917,00
Th�ringer Rostbratwurst	36194,18
Gnocchi di nonna Alice	34754,80
Manjimup Dried Apples	26065,40

The status bar at the bottom indicates "Query executed successfully." and "5 rows".

8. Buatlah view untuk melihat Order Details yang berisi OrderID, ProductID, ProductName, UnitPrice, Quantity, Discount, Harga setelah diskon.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating a view named `Order_Details`. The script includes a `CREATE VIEW` statement followed by a `SELECT` query that joins the `Products` and `Orders` tables. The `SELECT` query calculates the price after discount using the formula  $(1 - \text{Discount}) * \text{UnitPrice}$ . The bottom pane shows the results of the view, displaying a table with columns: OrderID, ProductID, ProductName, UnitPrice, Quantity, Discount, and HargaSetelahDiskon. The table contains 11 rows of data.

```
1 CREATE VIEW Order_Details
2 AS
3 SELECT O.OrderID, P.ProductID, P.ProductName, OD.UnitPrice, OD.Quantity, OD.Discount, (1-OD.Discount)*OD.UnitPrice AS HargaSetelahDiskon
4 FROM Products AS P
5 INNER JOIN [Order Details] AS OD ON P.ProductID = OD.ProductID
6 INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
7 GO
8 -- Query the view
9 SELECT *
10 FROM Order_Details
```

	OrderID	ProductID	ProductName	UnitPrice	Quantity	Discount	HargaSetelahDiskon
1	10248	11	Queso Cabrales	14,00	12	0	14
2	10248	42	Singaporean Hokkien Fried Mee	9,80	10	0	9,8
3	10248	72	Mozzarella di Giovanni	34,80	5	0	34,8
4	10249	14	Tofu	18,60	9	0	18,6
5	10249	51	Manjimup Dried Apples	42,40	40	0	42,4
6	10250	41	Jack's New England Clam Chowder	7,70	10	0	7,7
7	10250	51	Manjimup Dried Apples	42,40	35	0,15	36,04
8	10250	65	Louisiana Fiery Hot Pepper Sauce	16,80	15	0,15	14,28
9	10251	22	Gustaf's Knäckebröd	16,80	6	0,05	15,96
10	10251	57	Ravioli Angelo	15,60	15	0,05	14,82
11	10251	65	Louisiana Fiery Hot Pepper Sauce	16,80	20	0	16,8

9. Buatlah procedure Invoice untuk memanggil CustomerID, CustomerName, OrderID, OrderDate, RequiredDate, ShippedDate jika terdapat inputan CustomerID tertentu.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating a procedure named `Invoice_Procedure`. The script includes a `CREATE PROCEDURE` statement followed by a `SELECT` query that joins the `Customers` and `Orders` tables. The `SELECT` query retrieves the `CustomerID`, `CompanyName`, `OrderID`, `OrderDate`, `RequiredDate`, and `ShippedDate` for a given `CustomerID`. The bottom pane shows the results of the procedure, displaying a table with columns: CustomerID, CompanyName, OrderID, OrderDate, RequiredDate, and ShippedDate. The table contains 11 rows of data.

```
1 CREATE PROCEDURE Invoice_Procedure
2 AS
3 BEGIN
4 SET NOCOUNT ON
5
6 SELECT C.CustomerID, C.CompanyName, O.OrderID, O.OrderDate, O.RequiredDate, O.ShippedDate
7 FROM
8 Customers AS C
9 INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
10 END
11
12 EXEC Invoice_Procedure
```

	CustomerID	CompanyName	OrderID	OrderDate	RequiredDate	ShippedDate
1	VINET	Vins et alcools Chevalier	10248	1996-07-04 00:00:00.000	1996-08-01 00:00:00.000	1996-07-16 00:00:00.000
2	TOMSP	Toms Spezialitäten	10249	1996-07-05 00:00:00.000	1996-08-16 00:00:00.000	1996-07-10 00:00:00.000
3	HANAR	Hanari Carnes	10250	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000	1996-07-12 00:00:00.000
4	VICTE	Victuals en stock	10251	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000	1996-07-15 00:00:00.000
5	SUPRD	Suprêmes délicies	10252	1996-07-09 00:00:00.000	1996-08-06 00:00:00.000	1996-07-11 00:00:00.000
6	HANAR	Hanari Carnes	10253	1996-07-10 00:00:00.000	1996-07-24 00:00:00.000	1996-07-16 00:00:00.000
7	CHOPS	Chop-suey Chinese	10254	1996-07-11 00:00:00.000	1996-08-08 00:00:00.000	1996-07-23 00:00:00.000
8	RICSU	Richter Supermarkt	10255	1996-07-12 00:00:00.000	1996-08-09 00:00:00.000	1996-07-15 00:00:00.000
9	WELLI	Wellington Importadora	10256	1996-07-15 00:00:00.000	1996-08-12 00:00:00.000	1996-07-17 00:00:00.000
10	HILAA	HILARION-Abastos	10257	1996-07-16 00:00:00.000	1996-08-13 00:00:00.000	1996-07-22 00:00:00.000
11	ERNSH	Ernst Handel	10258	1996-07-17 00:00:00.000	1996-08-14 00:00:00.000	1996-07-23 00:00:00.000