DSLS 2023
MINI PROJECT DATA ENGINERING

SILVIA ASTRI RAHMANINGRUM

29/01/2023

—

DSLS 2023

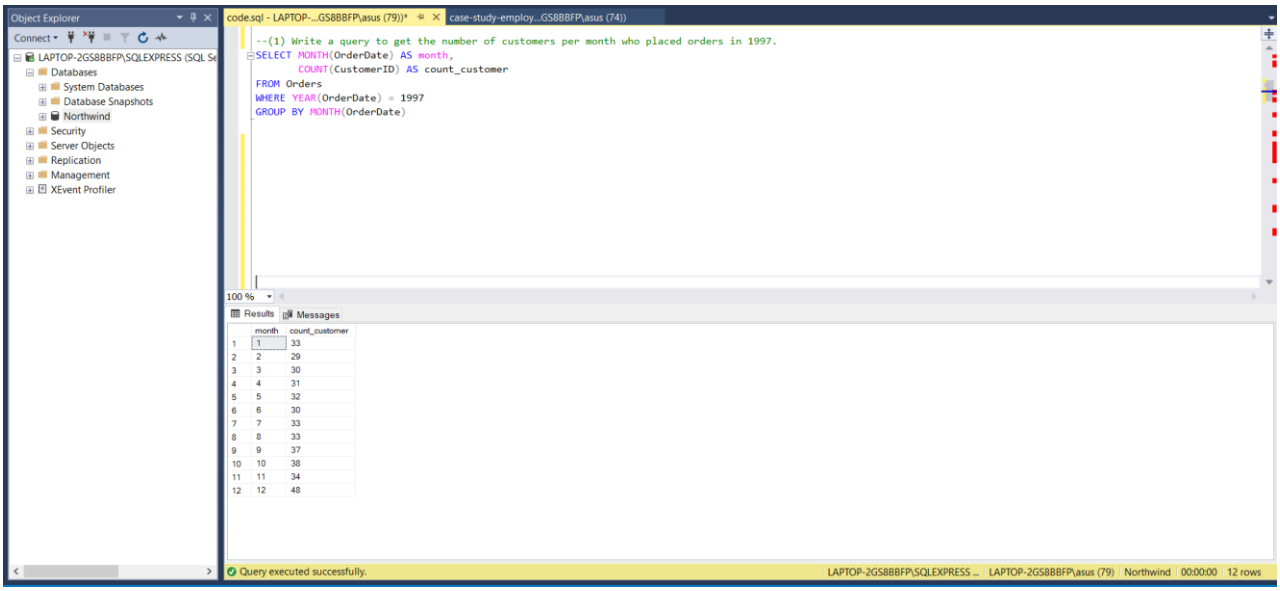
—

(2A) INTERMEDIATE QUERIES

In this assignment (2A) about Intermediate Queries has 9 point. To review your basic query skills, you are asked to work on this problem first:

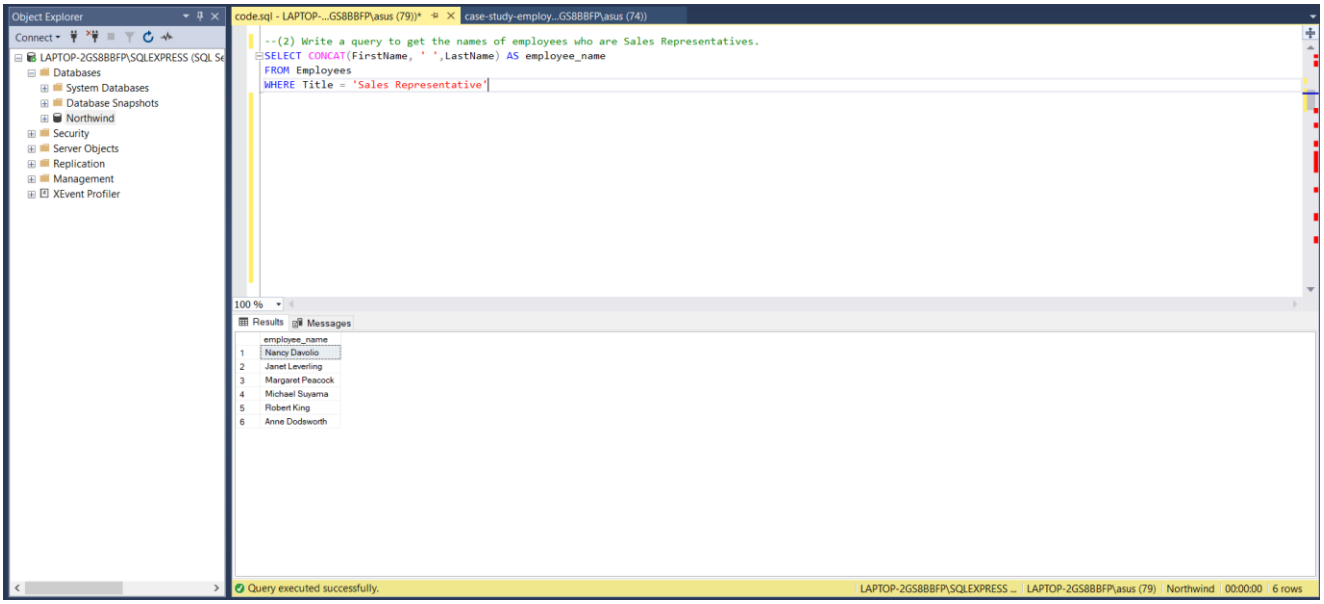
1. Write a query to get the number of customers per month who placed orders in 1997.

```
SELECT MONTH(OrderDate) AS month,
       COUNT(CustomerID) AS count_customer
FROM Orders
WHERE YEAR(OrderDate) = 1997
GROUP BY MONTH(OrderDate)
```



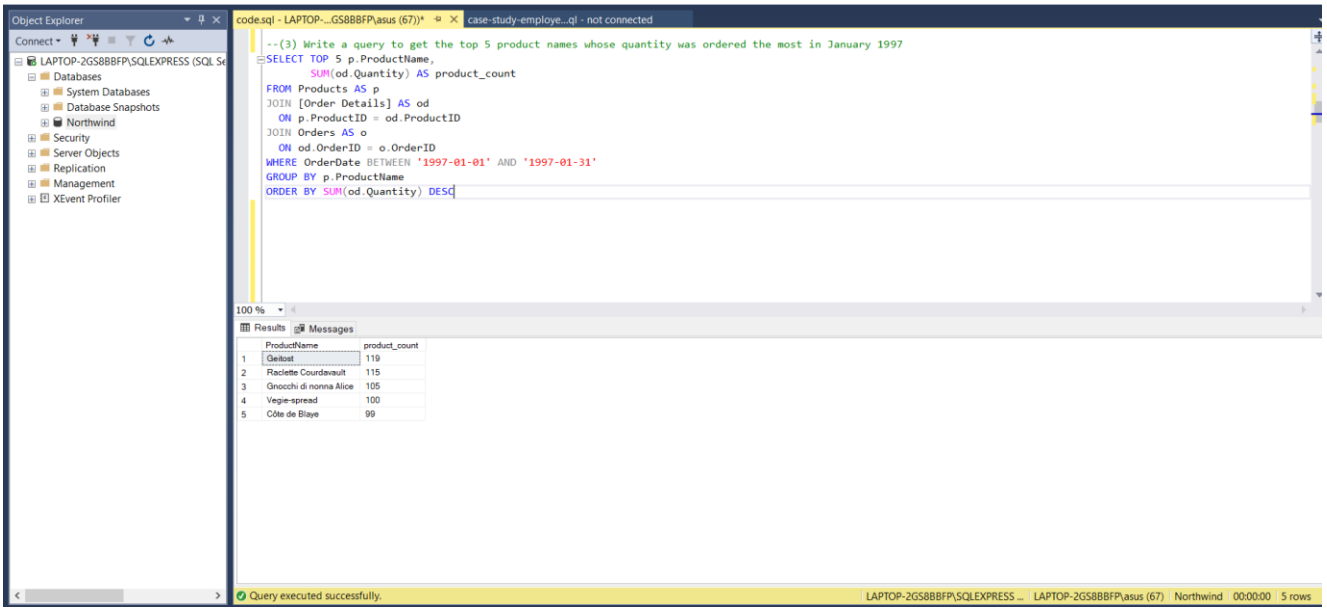
2. Write a query to get the names of employees who are Sales Representatives.

```
SELECT CONCAT(FirstName, ' ',LastName) AS employee_name
FROM Employees
WHERE Title = 'Sales Representative'
```



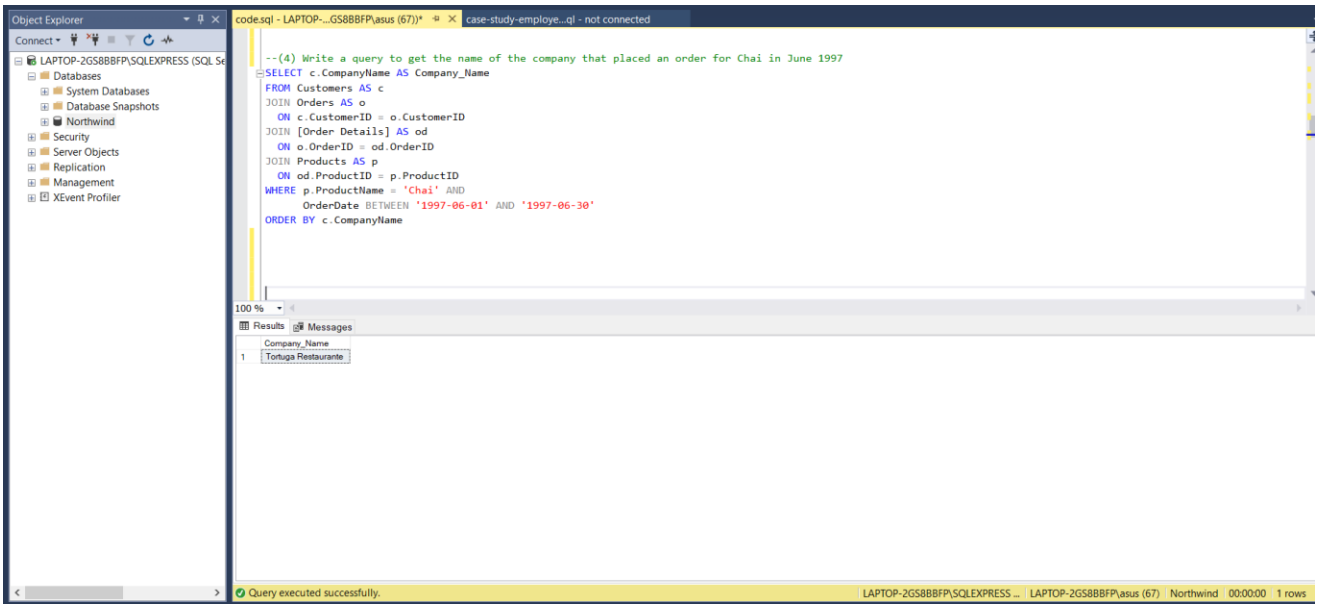
3. Write a query to get the top 5 product names whose quantity was ordered the most in January 1997.

```sql
SELECT TOP 5 p.ProductName,
           SUM(od.Quantity) AS product_count
FROM Products AS p
JOIN [Order Details] AS od
  ON p.ProductID = od.ProductID
JOIN Orders AS o
  ON od.OrderID = o.OrderID
WHERE OrderDate BETWEEN '1997-01-01' AND '1997-01-31'
GROUP BY p.ProductName
ORDER BY SUM(od.Quantity) DESC
```



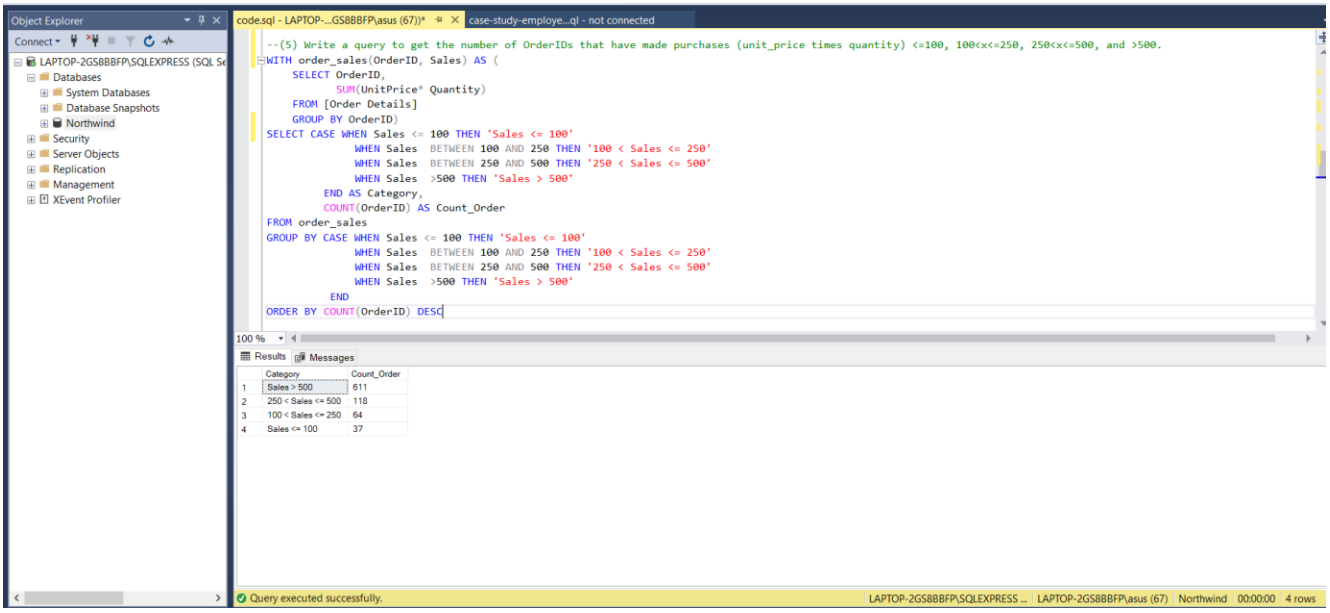
4. Write a query to get the name of the company that placed an order for Chai in June 1997.

```
SELECT c.CompanyName AS Company_Name
FROM Customers AS c
JOIN Orders AS o
  ON c.CustomerID = o.CustomerID
JOIN [Order Details] AS od
  ON o.OrderID = od.OrderID
JOIN Products AS p
  ON od.ProductID = p.ProductID
WHERE p.ProductName = 'Chai' AND
      OrderDate BETWEEN '1997-06-01' AND '1997-06-30'
ORDER BY c.CompanyName
```



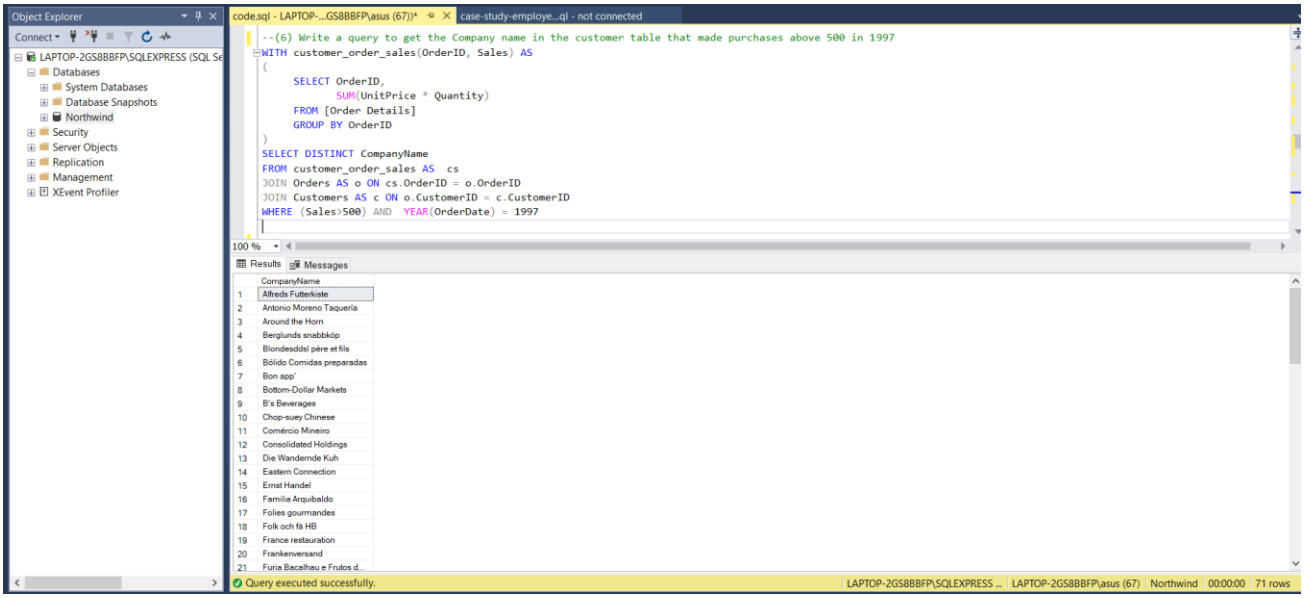
5. Write a query to get the number of OrderIDs that have made purchases (unit_price times quantity) <=100, 100<x<=250, 250<x<=500, and >500.

```
WITH order_sales(OrderID, Sales) AS (
       SELECT OrderID,
              SUM(UnitPrice* Quantity)
       FROM [Order Details]
       GROUP BY OrderID)
SELECT CASE WHEN Sales <= 100 THEN 'Sales <= 100'
            WHEN Sales  BETWEEN 100 AND 250 THEN '100 < Sales <= 250'
            WHEN Sales  BETWEEN 250 AND 500 THEN '250 < Sales <= 500'
            WHEN Sales  >500 THEN 'Sales > 500'
       END AS Category,
              COUNT(OrderID) AS Count_Order
FROM order_sales
GROUP BY CASE WHEN Sales <= 100 THEN 'Sales <= 100'
              WHEN Sales  BETWEEN 100 AND 250 THEN '100 < Sales <= 250'
              WHEN Sales  BETWEEN 250 AND 500 THEN '250 < Sales <= 500'
              WHEN Sales  >500 THEN 'Sales > 500'
         END
ORDER BY COUNT(OrderID) DESC
```



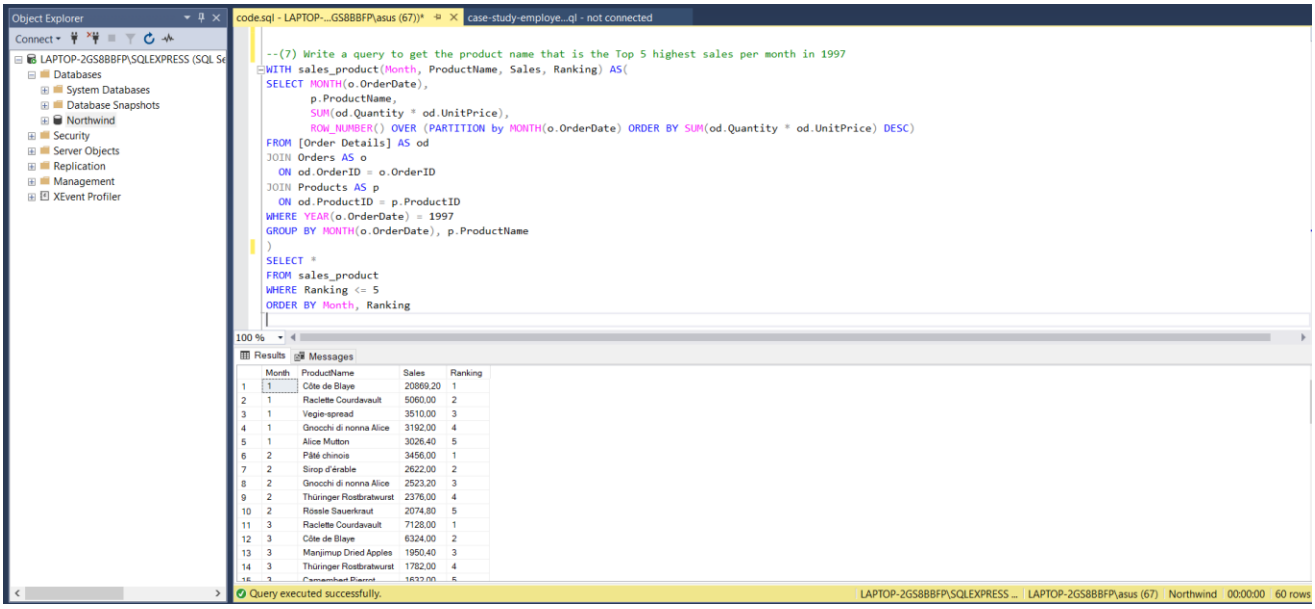
6. Write a query to get the Company name in the customer table that made purchases above 500 in 1997.

```
WITH customer_order_sales(OrderID, Sales) AS
(
        SELECT OrderID,
               SUM(UnitPrice * Quantity)
        FROM [Order Details]
        GROUP BY OrderID
)
SELECT DISTINCT CompanyName
FROM customer_order_sales AS  cs
JOIN Orders AS o ON cs.OrderID = o.OrderID
JOIN Customers AS c ON o.CustomerID = c.CustomerID
WHERE (Sales>500) AND  YEAR(OrderDate) = 1997
```



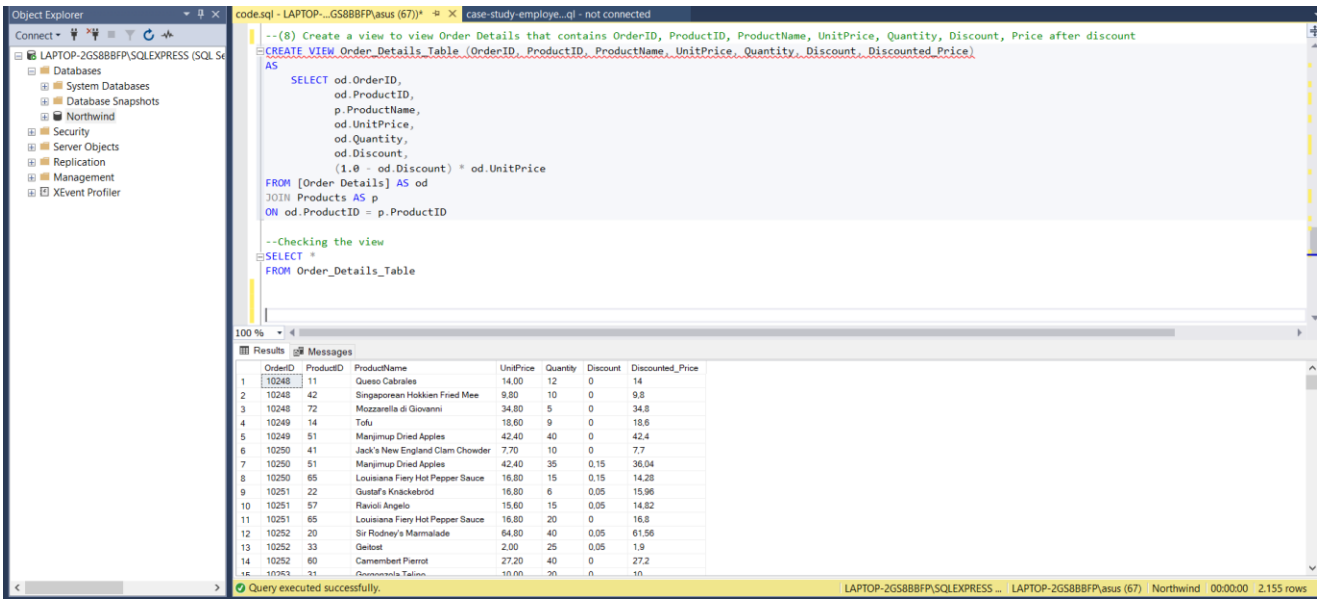
7. Write a query to get the product name that is the Top 5 highest sales per month in 1997.

```
WITH sales_product(Month, ProductName, Sales, Ranking) AS(
SELECT MONTH(o.OrderDate),
       p.ProductName,
       SUM(od.Quantity * od.UnitPrice),
       ROW_NUMBER() OVER (PARTITION by MONTH(o.OrderDate) ORDER BY SUM(od.Quantity *
                                                                       od.UnitPrice) DESC)
FROM [Order Details] AS od
JOIN Orders AS o
  ON od.OrderID = o.OrderID
JOIN Products AS p
  ON od.ProductID = p.ProductID
WHERE YEAR(o.OrderDate) = 1997
GROUP BY MONTH(o.OrderDate), p.ProductName
)
SELECT *
FROM sales_product
WHERE Ranking <= 5
ORDER BY Month, Ranking
```



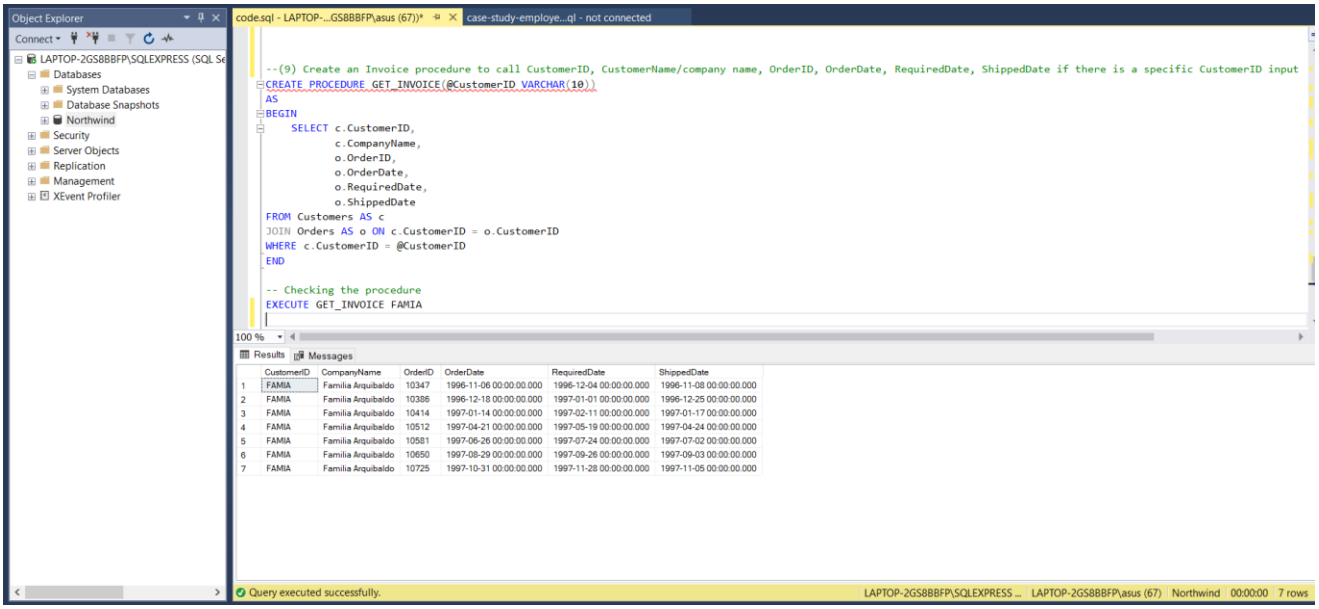
8. Create a view to view Order Details that contains OrderID, ProductID, ProductName, UnitPrice, Quantity, Discount, Price after discount.

```
CREATE VIEW Order_Details_Table (OrderID, ProductID, ProductName, UnitPrice, Quantity,
                                 Discount, Discounted_Price)
AS
       SELECT od.OrderID,
              od.ProductID,
              p.ProductName,
              od.UnitPrice,
              od.Quantity,
              od.Discount,
              (1.0 - od.Discount) * od.UnitPrice
FROM [Order Details] AS od
JOIN Products AS p
ON od.ProductID = p.ProductID

--Checking the view
SELECT *
FROM Order_Details_Table
```



9. Create an Invoice procedure to call CustomerID, CustomerName/company name, OrderID, OrderDate, RequiredDate, ShippedDate if there is a specific CustomerID input

```
CREATE PROCEDURE GET_INVOICE(@CustomerID VARCHAR(10))
AS
BEGIN
	SELECT c.CustomerID,
		c.CompanyName,
		o.OrderID,
		o.OrderDate,
		o.RequiredDate,
		o.ShippedDate
FROM Customers AS c
JOIN Orders AS o ON c.CustomerID = o.CustomerID
WHERE c.CustomerID = @CustomerID
END

-- Checking the procedure
EXECUTE GET_INVOICE FAMIA
```

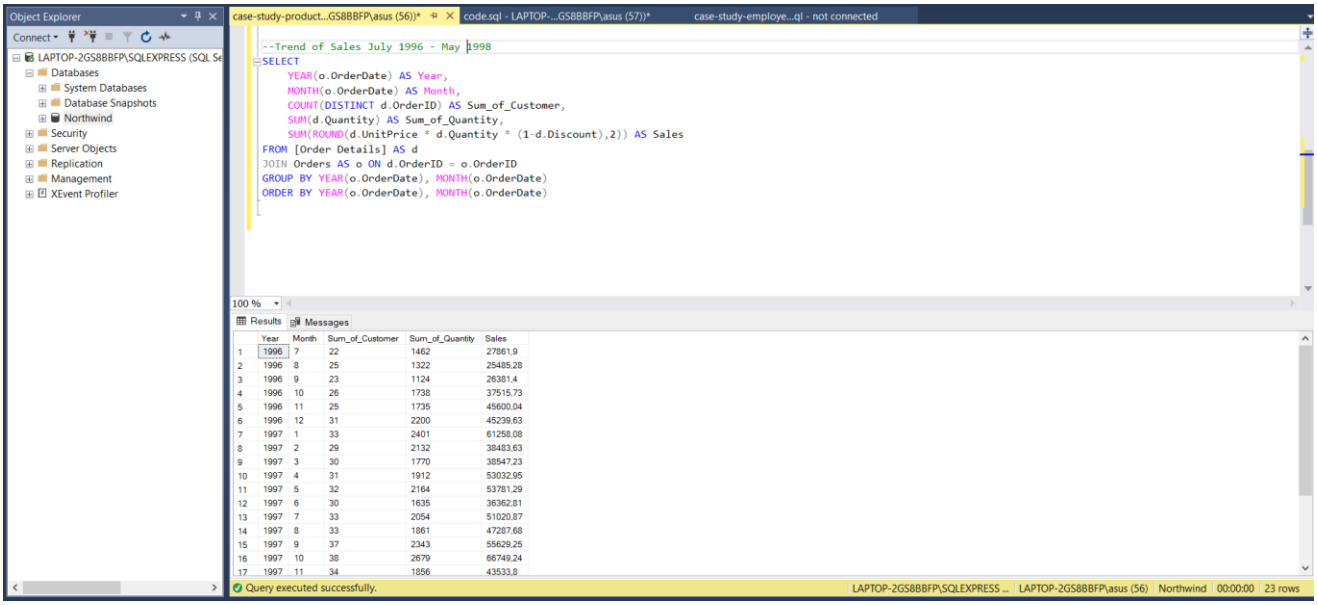


1. SALES ANALYSIS

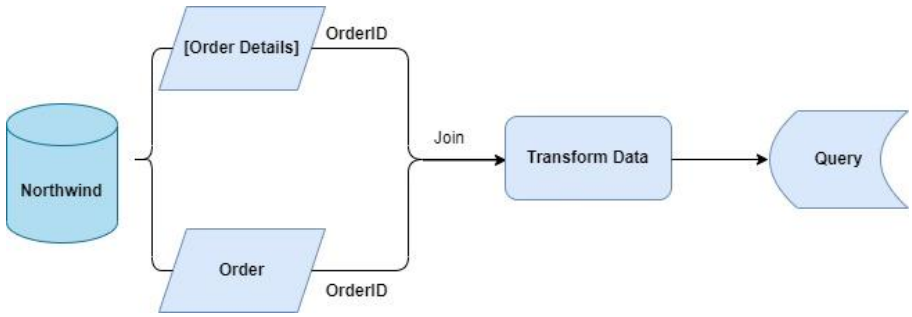
In this study case, the sales department want to find insight of the sales during July 1996 – May 1998.

SQL Query

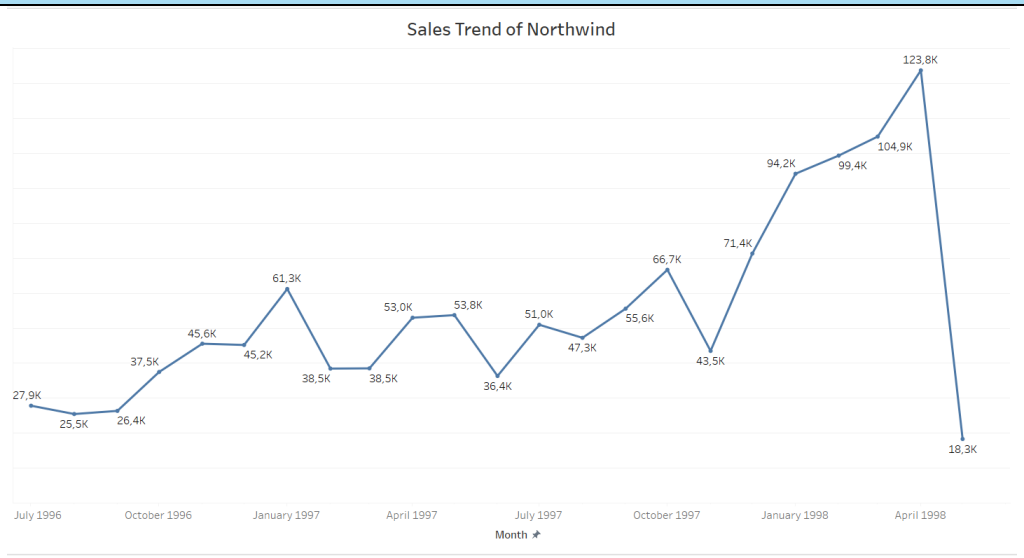
```
SELECT
       YEAR(o.OrderDate) AS Year,
       MONTH(o.OrderDate) AS Month,
       COUNT(DISTINCT d.OrderID) AS Sum_of_Customer,
       SUM(d.Quantity) AS Sum_of_Quantity,
       SUM(ROUND(d.UnitPrice * d.Quantity * (1-d.Discount),2)) AS Sales
FROM [Order Details] AS d
JOIN Orders AS o ON d.OrderID = o.OrderID
GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate)
ORDER BY YEAR(o.OrderDate), MONTH(o.OrderDate)
```



Flowchart



Visualizations

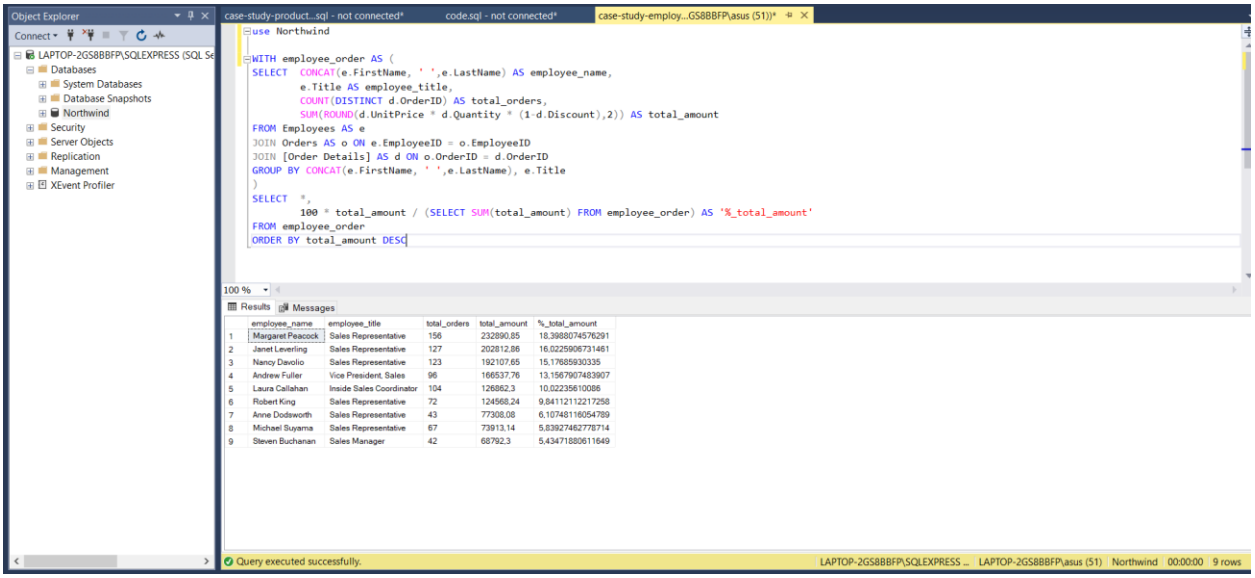


Based on the chart above, the trend of sales is up and the by month. The highest sales in April 1998 (123.8K) but in May the sales increase until (18.3K).

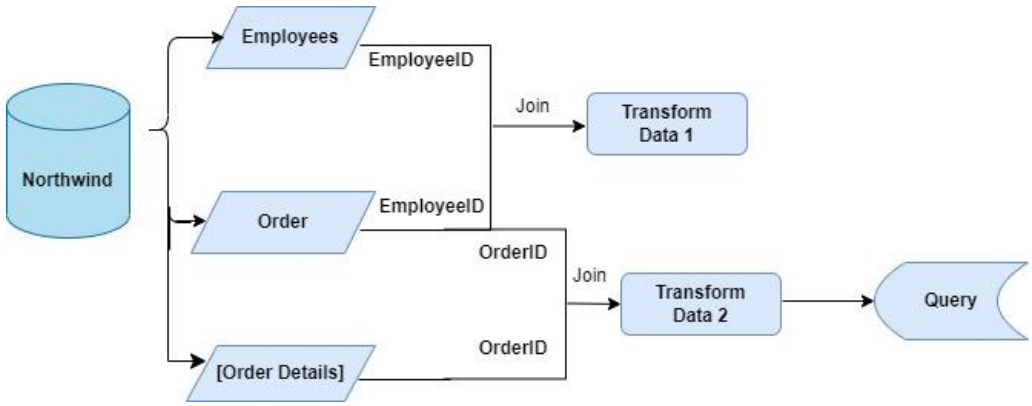
2. EMPLOYEE ANALYSIS

SQL Query

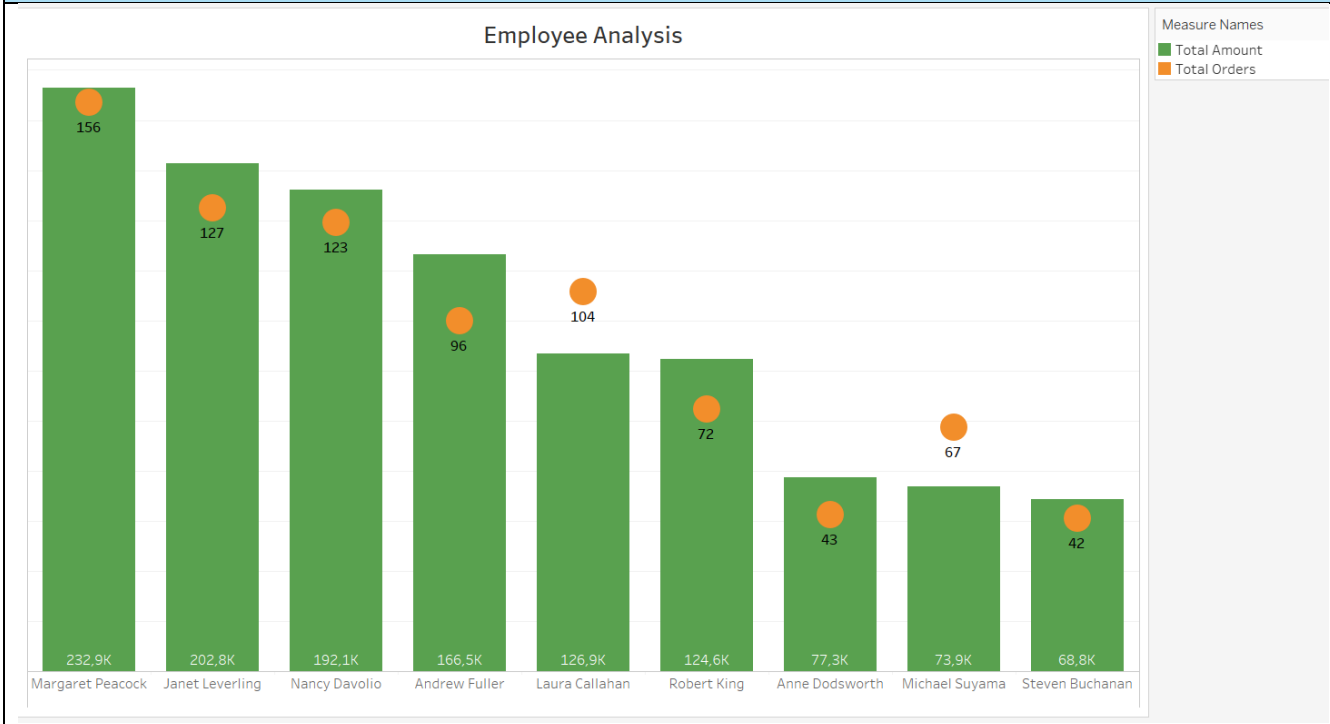
```
WITH employee_order AS (
SELECT  CONCAT(e.FirstName, ' ',e.LastName) AS employee_name,
        e.Title AS employee_title,
        COUNT(DISTINCT d.OrderID) AS total_orders,
        SUM(ROUND(d.UnitPrice * d.Quantity * (1-d.Discount),2)) AS total_amount
FROM Employees AS e
JOIN Orders AS o ON e.EmployeeID = o.EmployeeID
JOIN [Order Details] AS d ON o.OrderID = d.OrderID
GROUP BY CONCAT(e.FirstName, ' ',e.LastName), e.Title
)
SELECT  *,
       100 * total_amount / (SELECT SUM(total_amount) FROM employee_order) AS '%_total_amount'
FROM employee_order
ORDER BY total_amount DESC
```



Flowchart



Visualizations

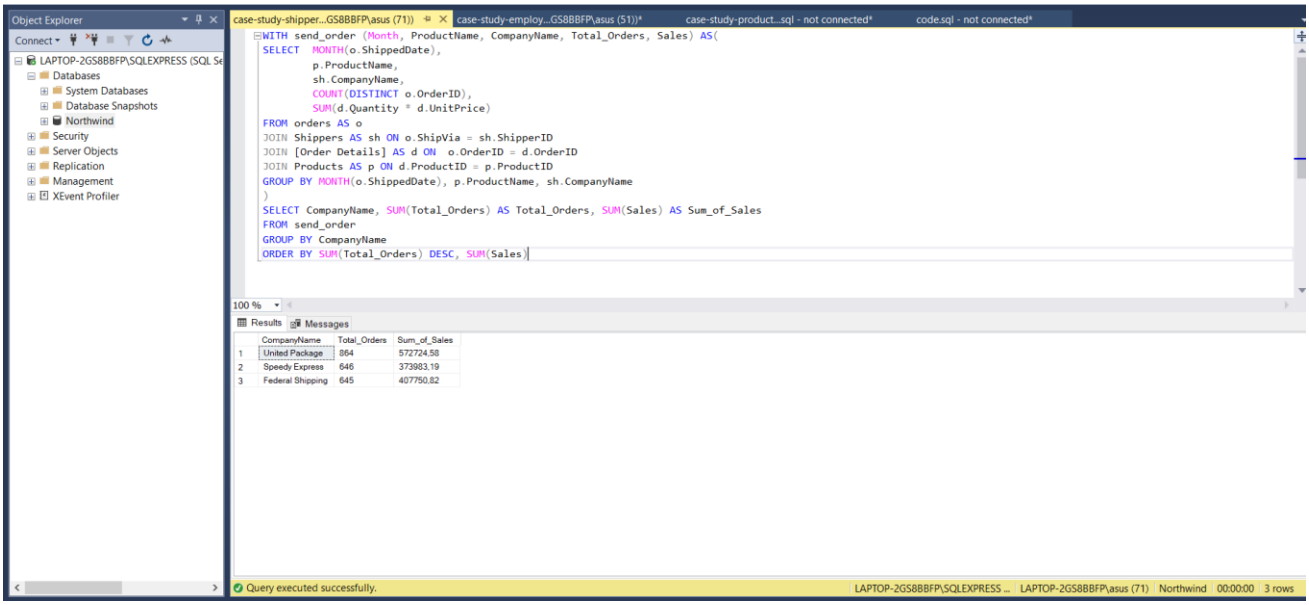


Based on the chart above, Margaret Peacock is the Most Valuable Employee who handle the most orders with a total sales 232.9K and total orders 156 for customer purchase.

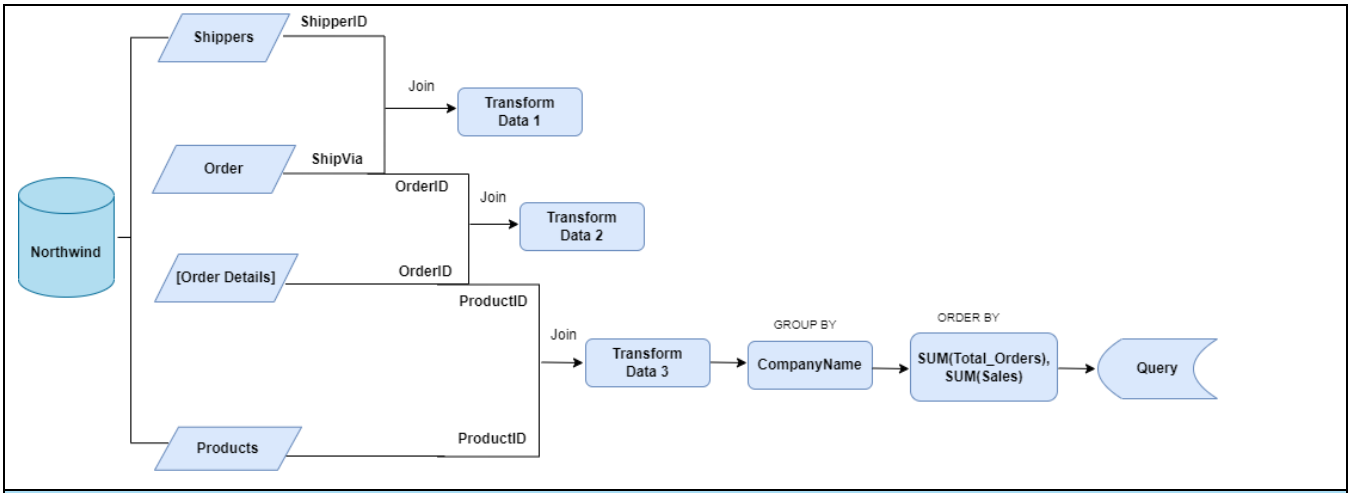
3. SHIPPER ANALYSIS

SQL Query

```
WITH send_order (Month, ProductName, CompanyName, Total_Orders, Sales) AS(
SELECT  MONTH(o.ShippedDate),
        p.ProductName,
        sh.CompanyName,
        COUNT(DISTINCT o.OrderID),
        SUM(d.Quantity * d.UnitPrice)
FROM orders AS o
JOIN Shippers AS sh ON o.ShipVia = sh.ShipperID
JOIN [Order Details] AS d ON  o.OrderID = d.OrderID
JOIN Products AS p ON d.ProductID = p.ProductID
GROUP BY MONTH(o.ShippedDate), p.ProductName, sh.CompanyName
)
SELECT CompanyName, SUM(Total_Orders) AS Total_Orders, SUM(Sales) AS Sum_of_Sales
FROM send_order
GROUP BY CompanyName
ORDER BY SUM(Total_Orders) DESC, SUM(Sales)
```

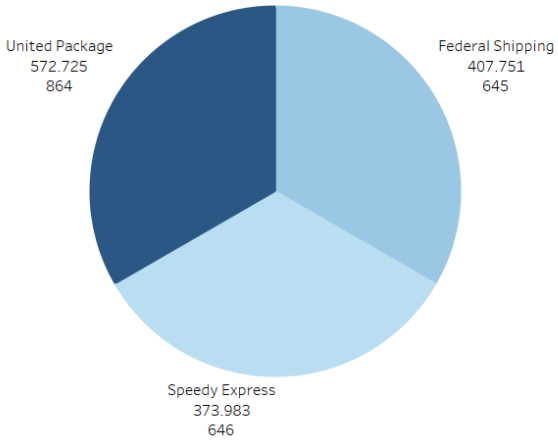


Flowchart



Visualizations

Shipper Analysis



The shipping company name United Package is the favorite by customer to shipping their order. For July 1996 – May 1998 has delivered around 864.