Muhamad Fadhli Akbar

Cortana

ASSIGMENT 3 ABOUT SQL

SQL 3.1

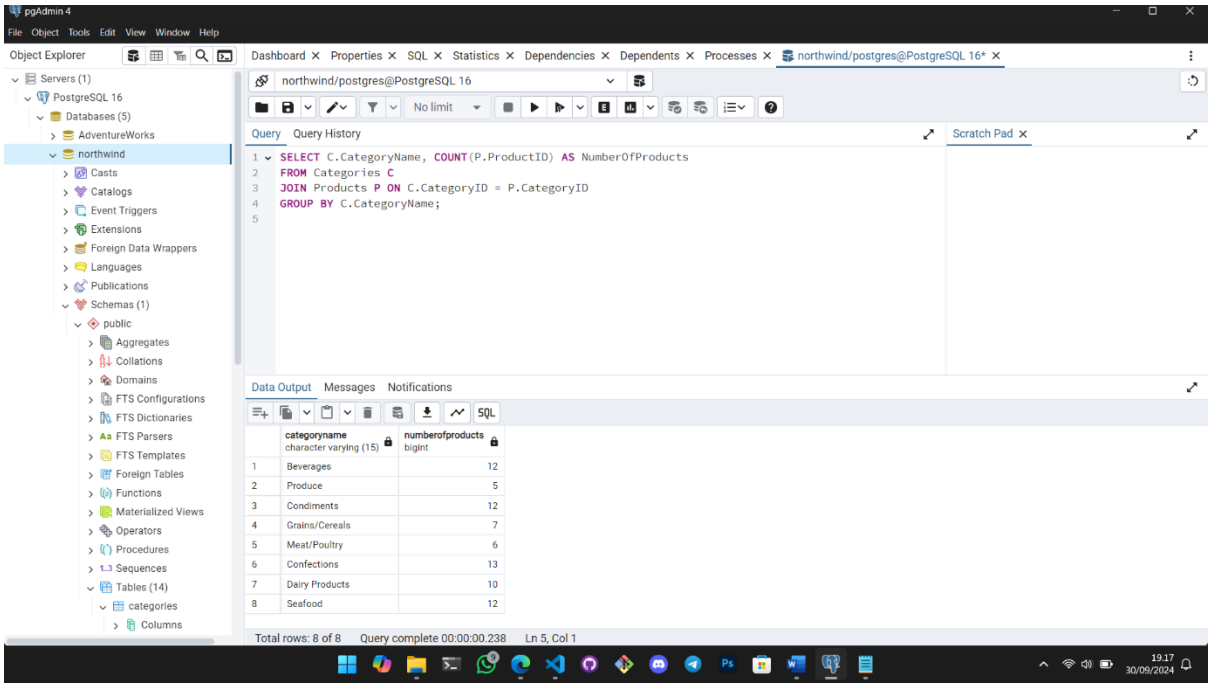
1. What is the number of products for each category (do it with joins)

SELECT C.CategoryName, COUNT(P.ProductID) AS NumberOfProducts

FROM Categories C

JOIN Products P ON C.CategoryID = P.CategoryID

GROUP BY C.CategoryName;



2. Total value of each product sold for year of 1997

SELECT p.ProductID, SUM(od.UnitPrice * od.Quantity) AS total_value

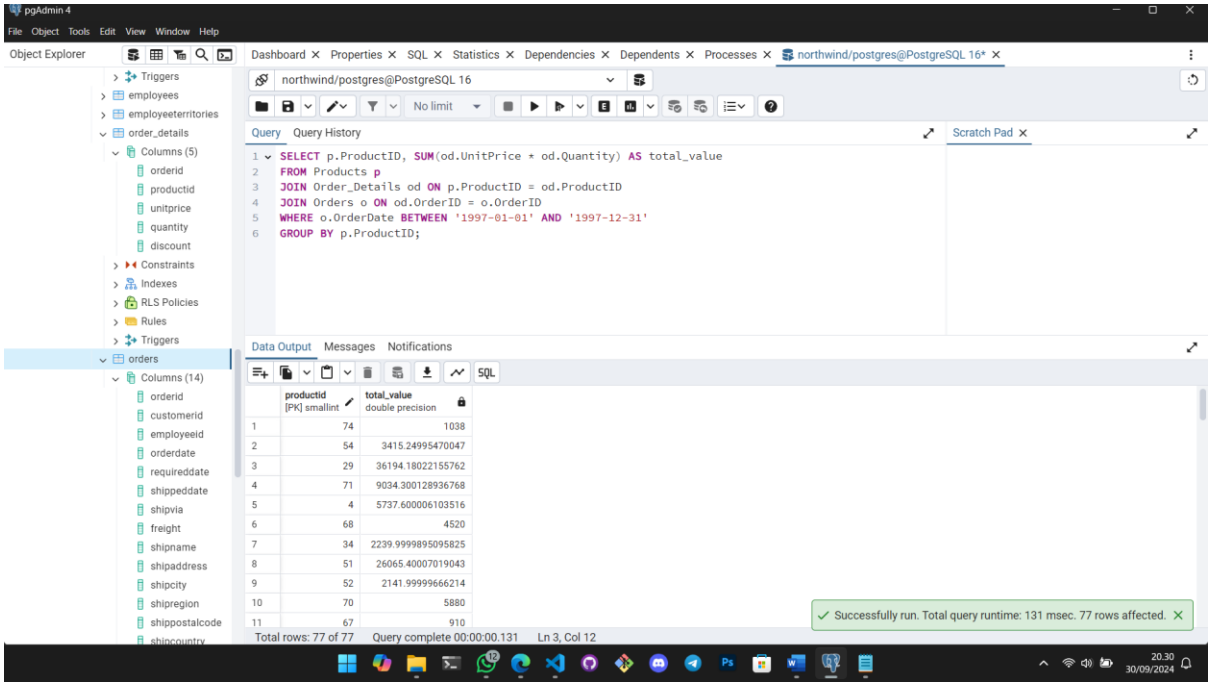
FROM Products p

JOIN Order_Details od ON p.ProductID = od.ProductID

JOIN Orders o ON od.OrderID = o.OrderID

WHERE o.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'

GROUP BY p.ProductID;



3. Customers that have bought more than $5000 of products

SELECT c.CustomerID, c.CompanyName, SUM(od.UnitPrice * od.Quantity) AS total_spent

FROM Customers c

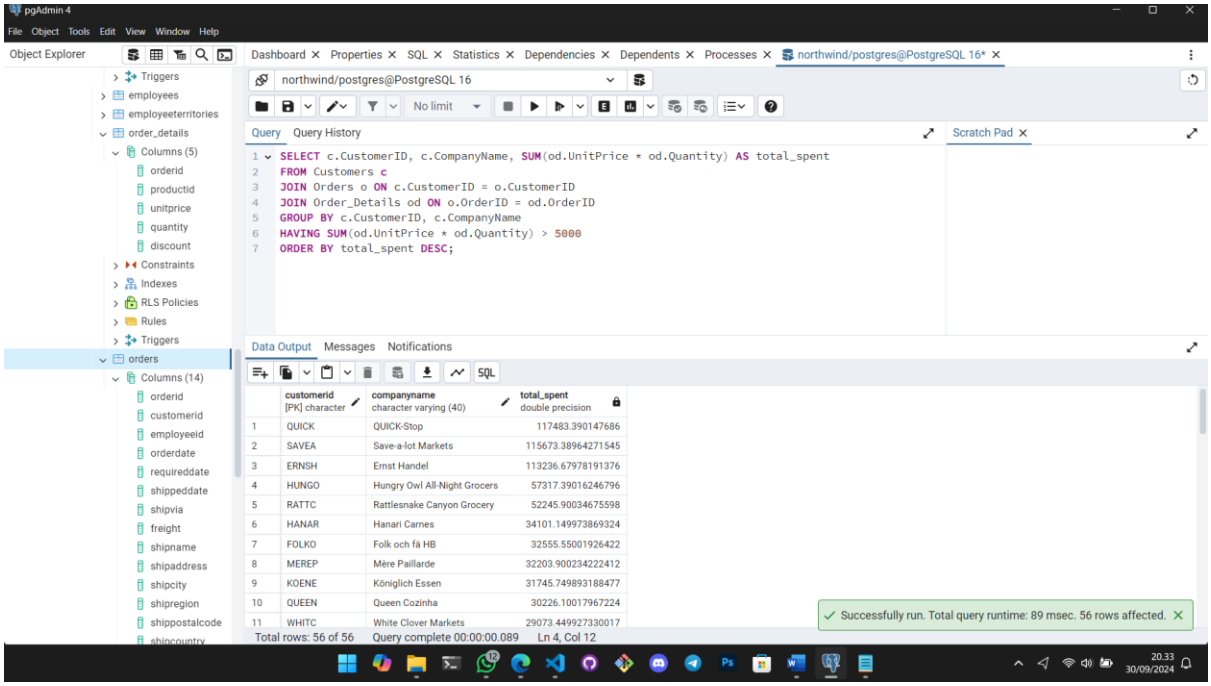
JOIN Orders o ON c.CustomerID = o.CustomerID

JOIN Order_Details od ON o.OrderID = od.OrderID

GROUP BY c.CustomerID, c.CompanyName

HAVING SUM(od.UnitPrice * od.Quantity) > 5000

ORDER BY total_spent DESC;



4. Customers that have bought more than $5000 of products with order date in first six month of the year of 1997 (see the effect of WHERE clause)

SELECT c.CustomerID, c.CompanyName, SUM(od.UnitPrice * od.Quantity) AS total_spent

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

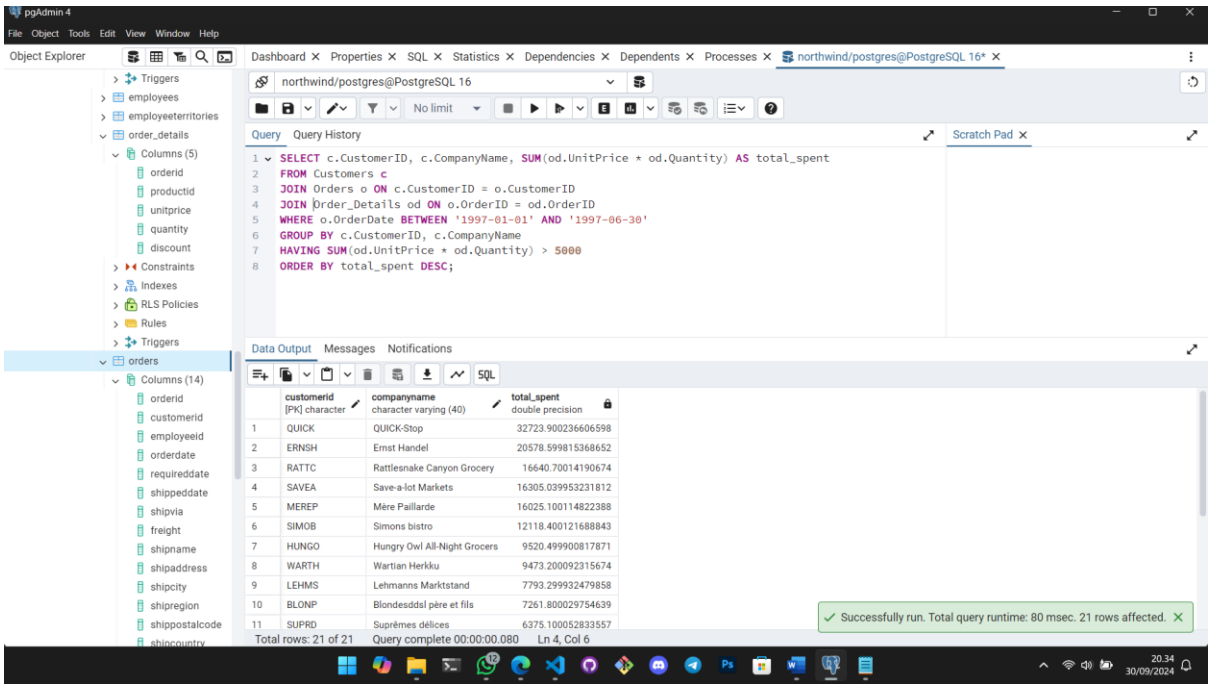
JOIN Order_Details od ON o.OrderID = od.OrderID

WHERE o.OrderDate BETWEEN '1997-01-01' AND '1997-06-30'

GROUP BY c.CustomerID, c.CompanyName

HAVING SUM(od.UnitPrice * od.Quantity) > 5000

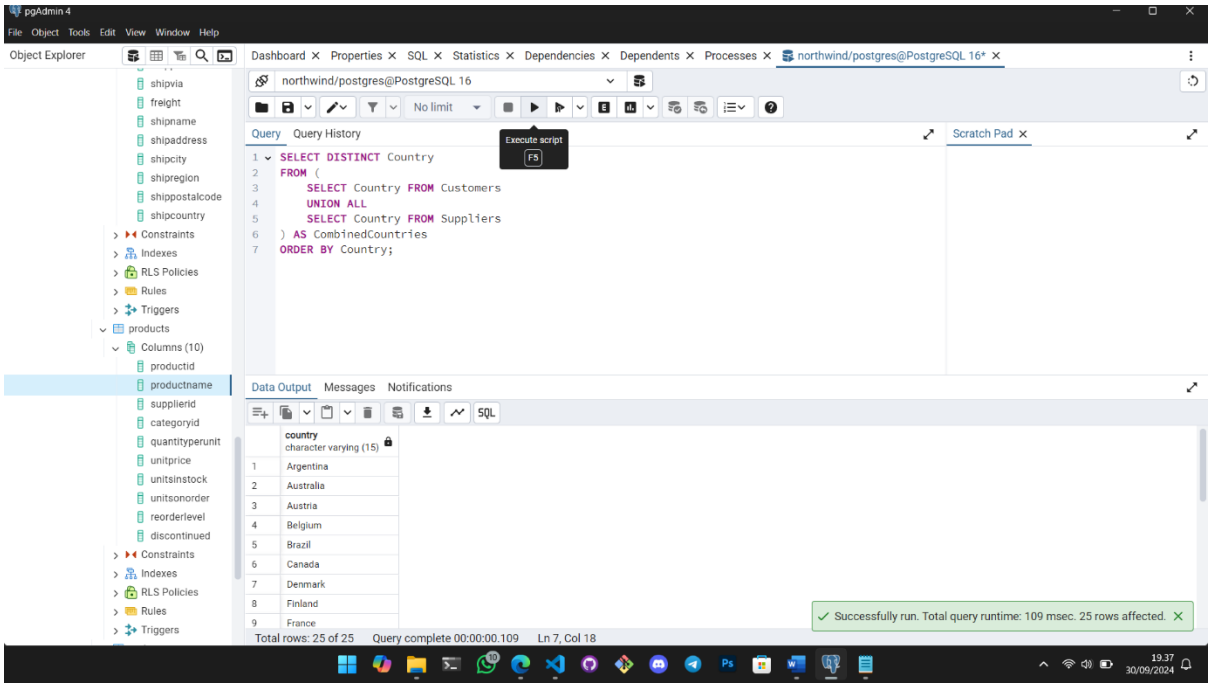
ORDER BY total_spent DESC;



SQL 3.2

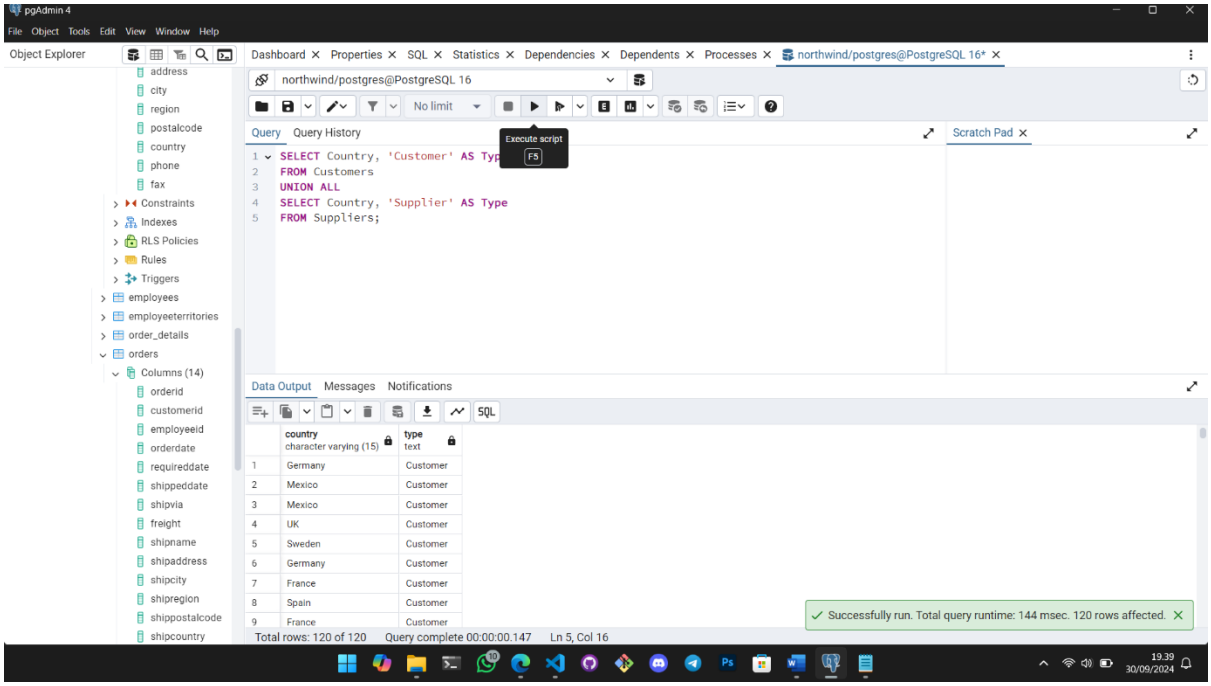
1. Distinct countries of all our customers and suppliers in alphabetical order

```
SELECT DISTINCT Country
FROM (
    SELECT Country FROM Customers
    UNION ALL
    SELECT Country FROM Suppliers
) AS CombinedCountries
ORDER BY Country;
```



2. All list of countries of our suppliers and customers, with a record for each one

```
SELECT Country, 'Customer' AS Type
FROM Customers
UNION ALL
SELECT Country, 'Supplier' AS Type
FROM Suppliers;
```

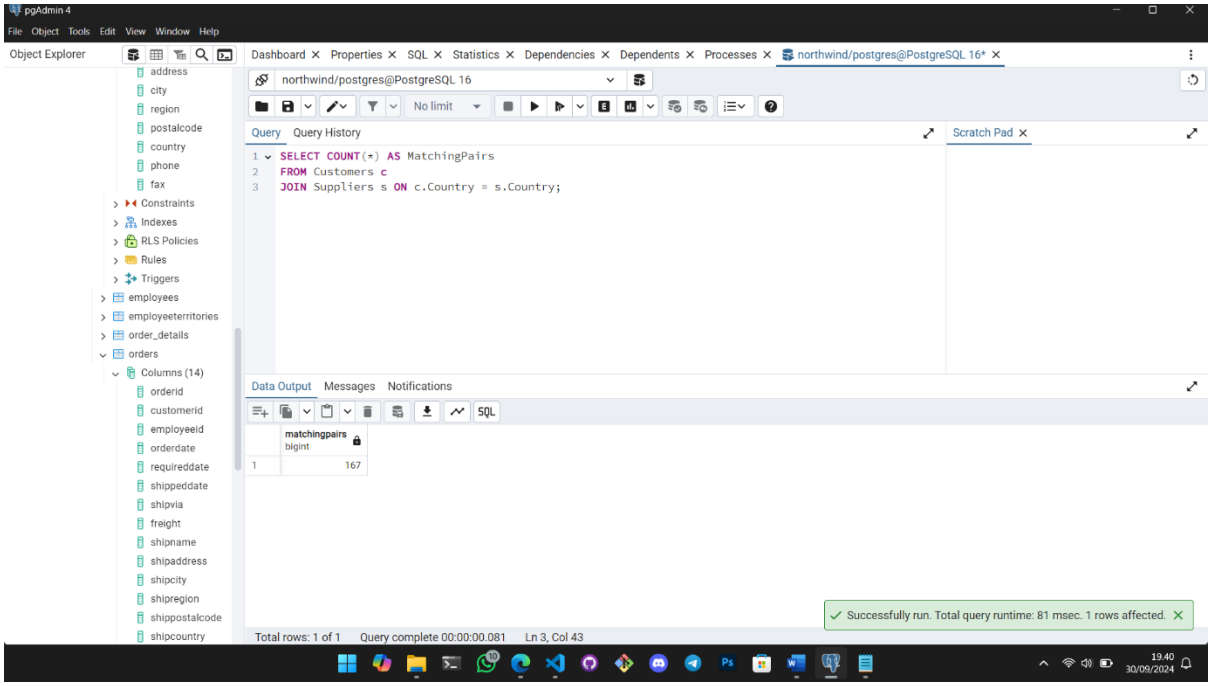


3. Find the number of customer and supplier pairs that are in the same country

SELECT COUNT(*) AS MatchingPairs

FROM Customers c

JOIN Suppliers s ON c.Country = s.Country;



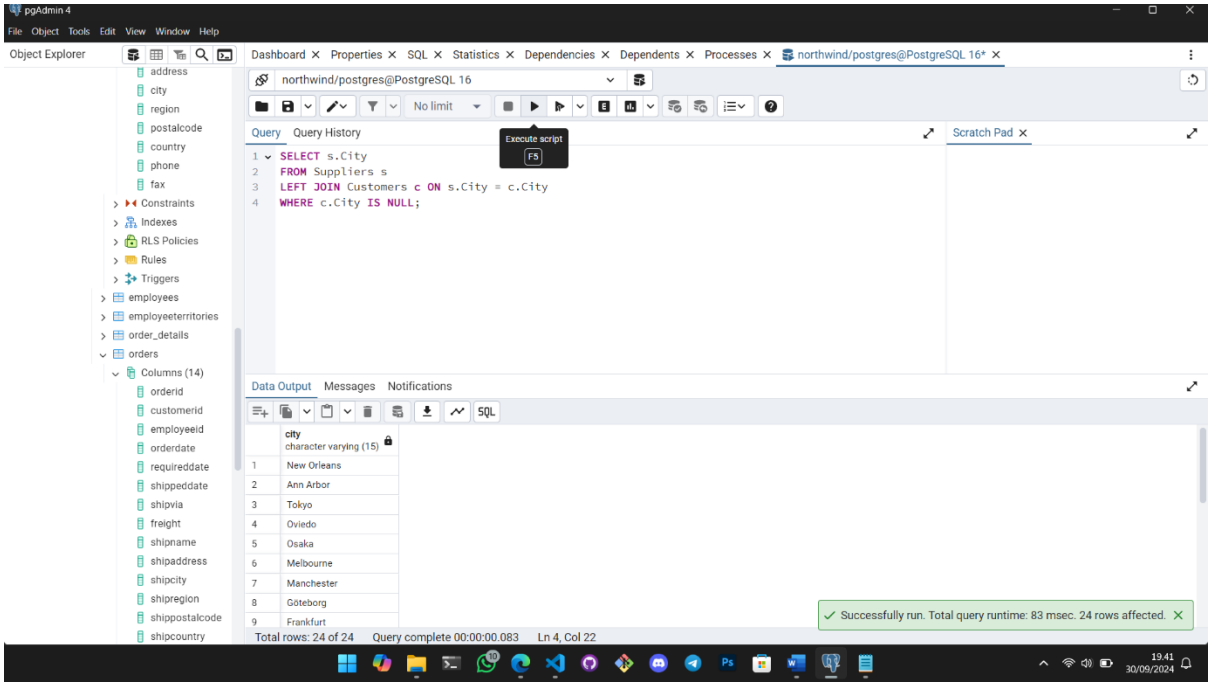
4. Cities that have a supplier with no customer

SELECT s.City

FROM Suppliers s

LEFT JOIN Customers c ON s.City = c.City

WHERE c.City IS NULL;



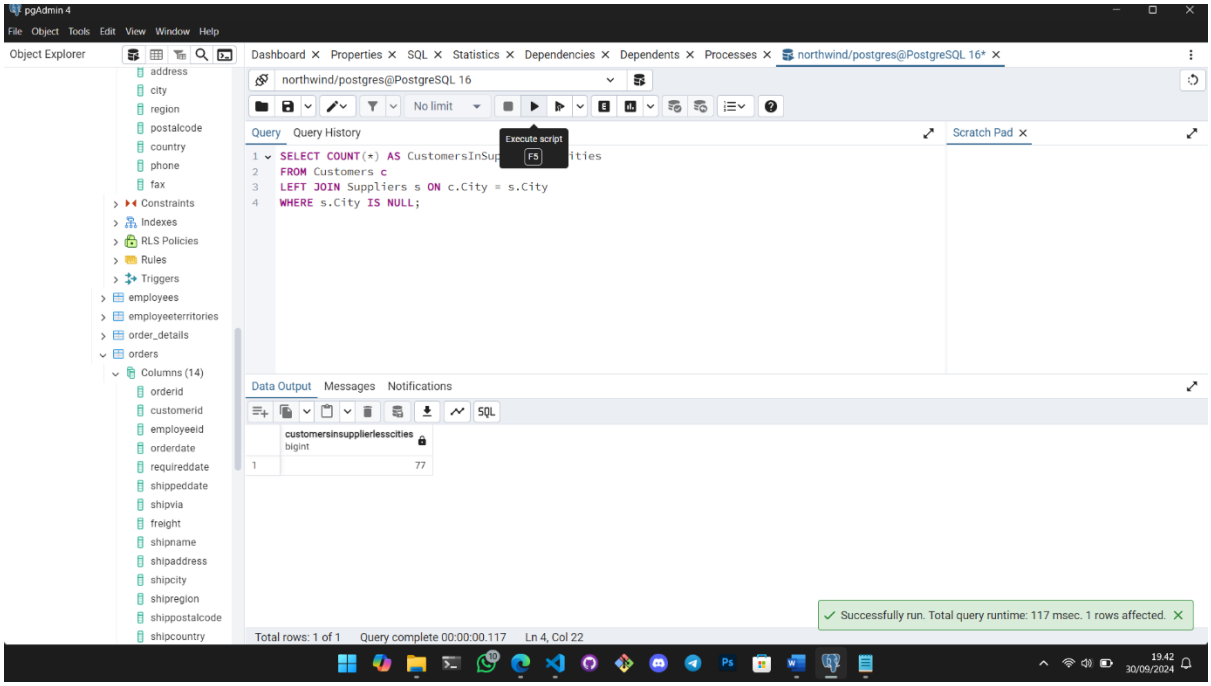
5. How many customers do we have in cities without suppliers

SELECT COUNT(*) AS CustomersInSupplierlessCities

FROM Customers c

LEFT JOIN Suppliers s ON c.City = s.City

WHERE s.City IS NULL;



SQL 3.3

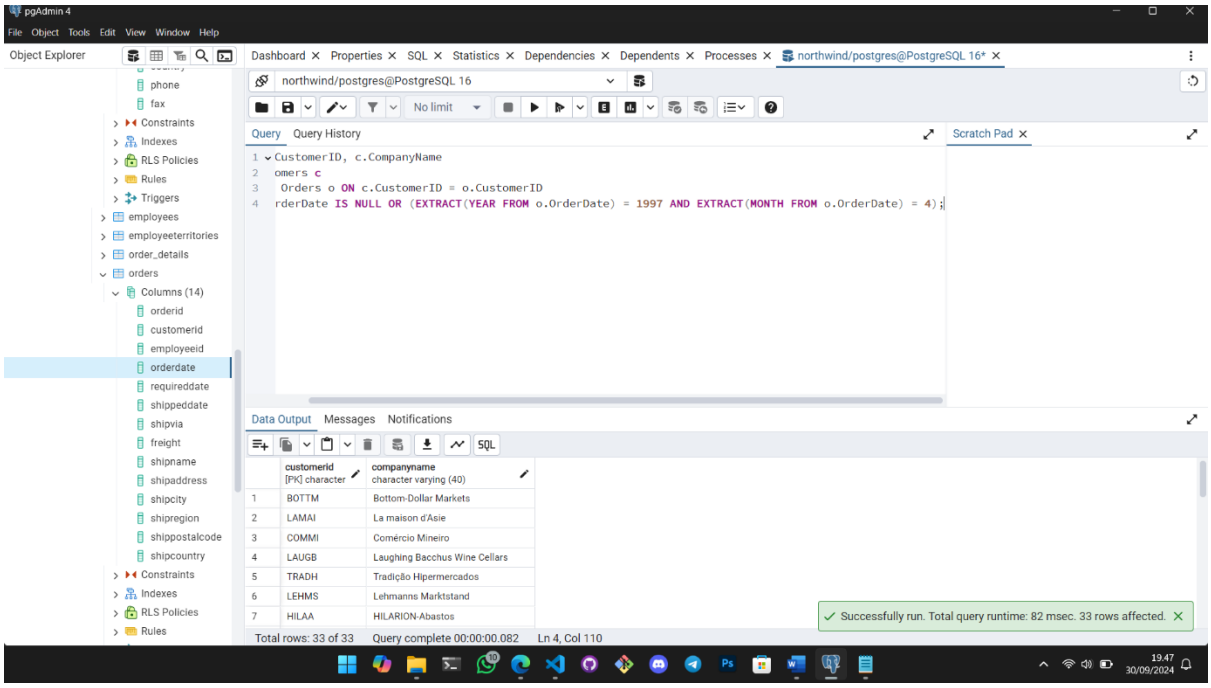
1. How do you find customers who didn't have an order in April, 1997

SELECT c.CustomerID, c.CompanyName

FROM Customers c

LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderDate IS NULL OR (EXTRACT(YEAR FROM o.OrderDate) = 1997 AND EXTRACT(MONTH FROM o.OrderDate) = 4);



2. Find all suppliers that have had an order with 1 item

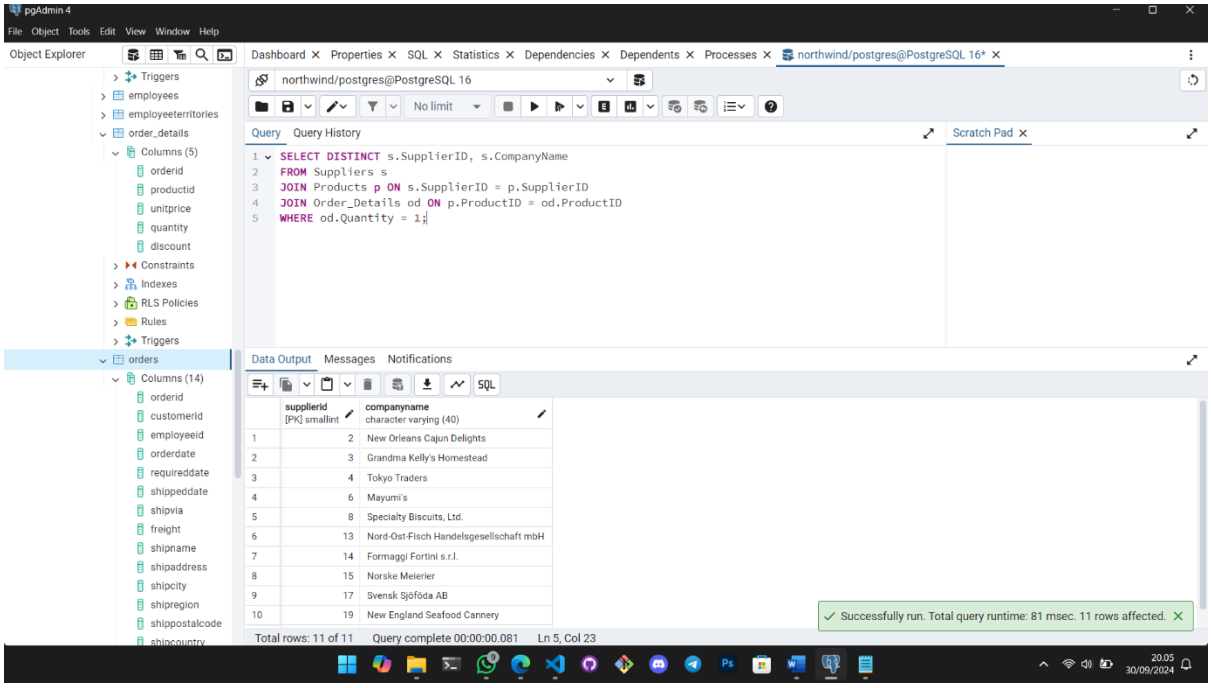
SELECT DISTINCT s.SupplierID, s.CompanyName

FROM Suppliers s

JOIN Products p ON s.SupplierID = p.SupplierID

JOIN Order_Details od ON p.ProductID = od.ProductID

WHERE od.Quantity = 1;



3. Find all distinct customers that ordered more in one item than the average orderamount per item of all customers

WITH AvgOrderItemQuantity AS (

  SELECT AVG(od.Quantity) AS AvgQuantity

  FROM Order_Details od

)

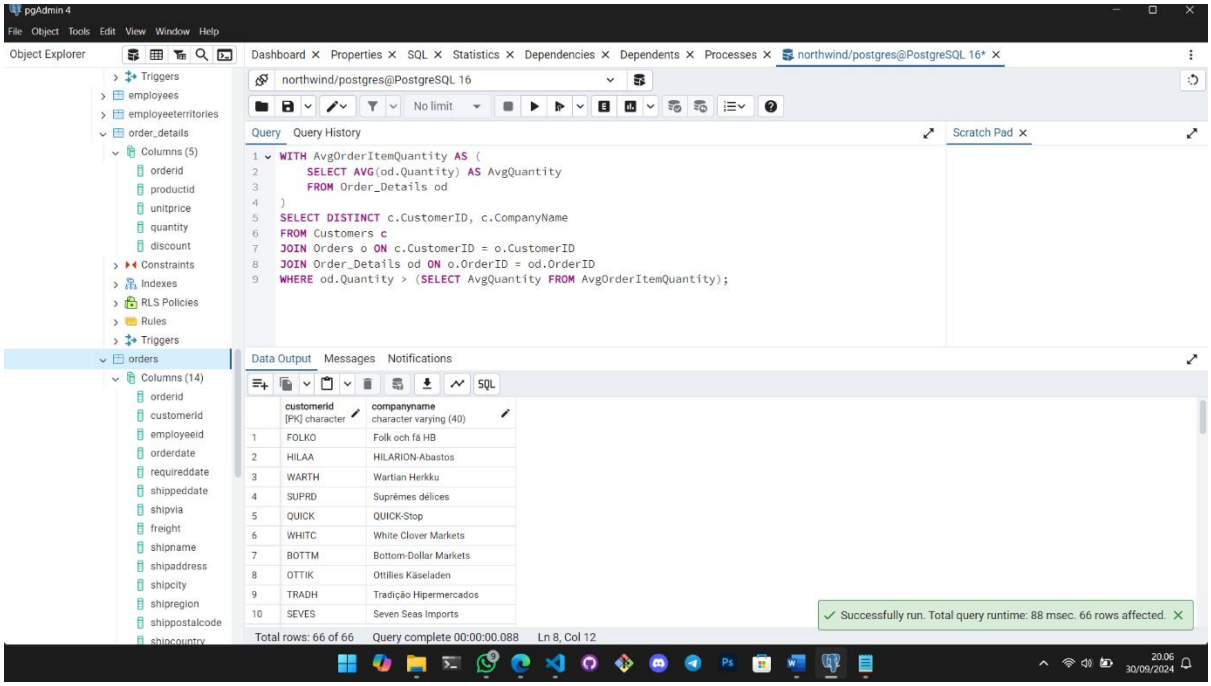
SELECT DISTINCT c.CustomerID, c.CompanyName

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

JOIN Order_Details od ON o.OrderID = od.OrderID

WHERE od.Quantity > (SELECT AvgQuantity FROM AvgOrderItemQuantity);

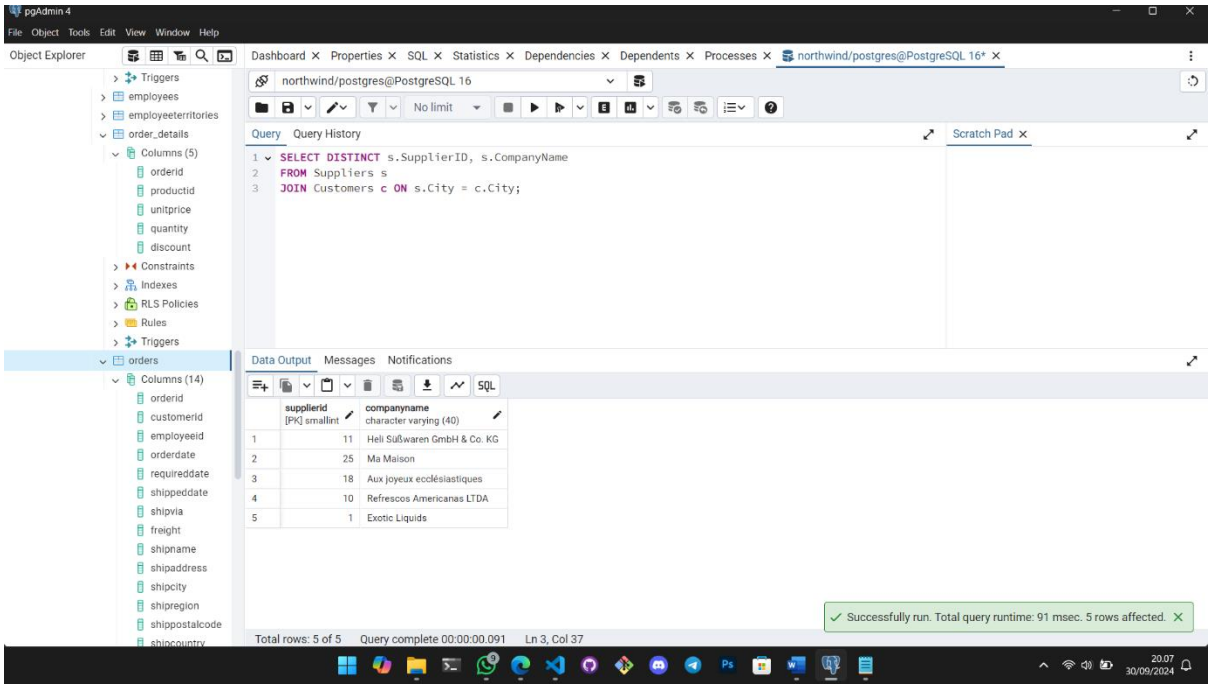


4. Find all suppliers that are in the same city as a customer

SELECT DISTINCT s.SupplierID, s.CompanyName

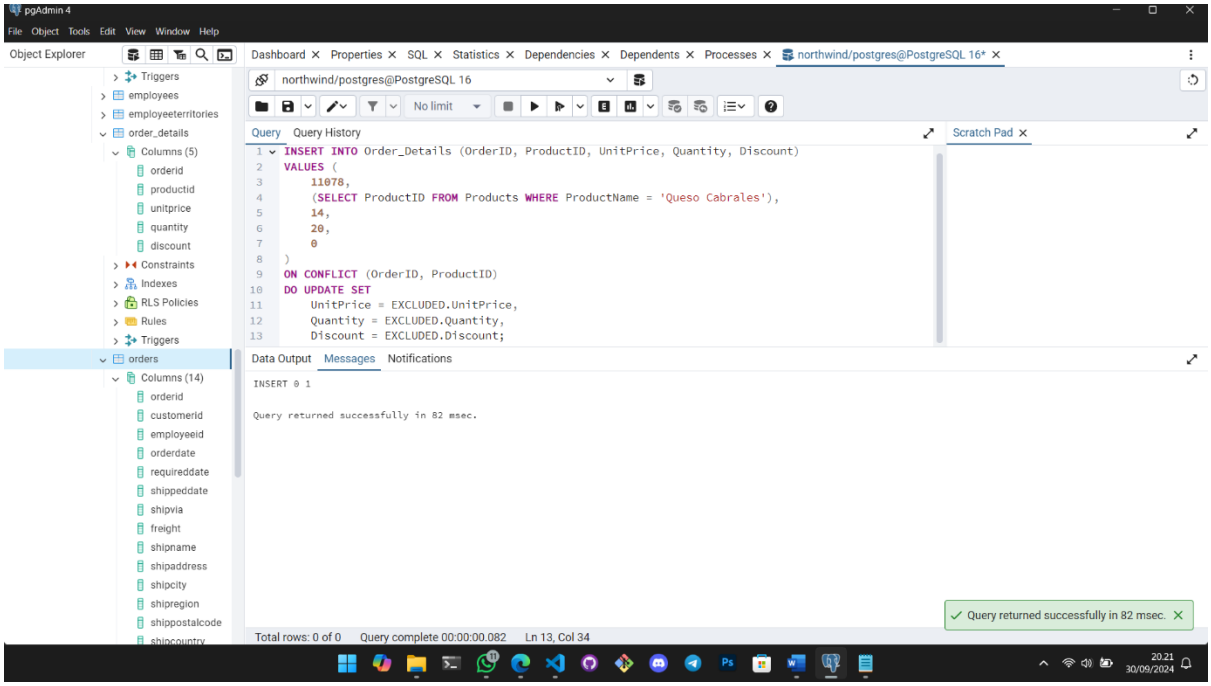
FROM Suppliers s

JOIN Customers c ON s.City = c.City;

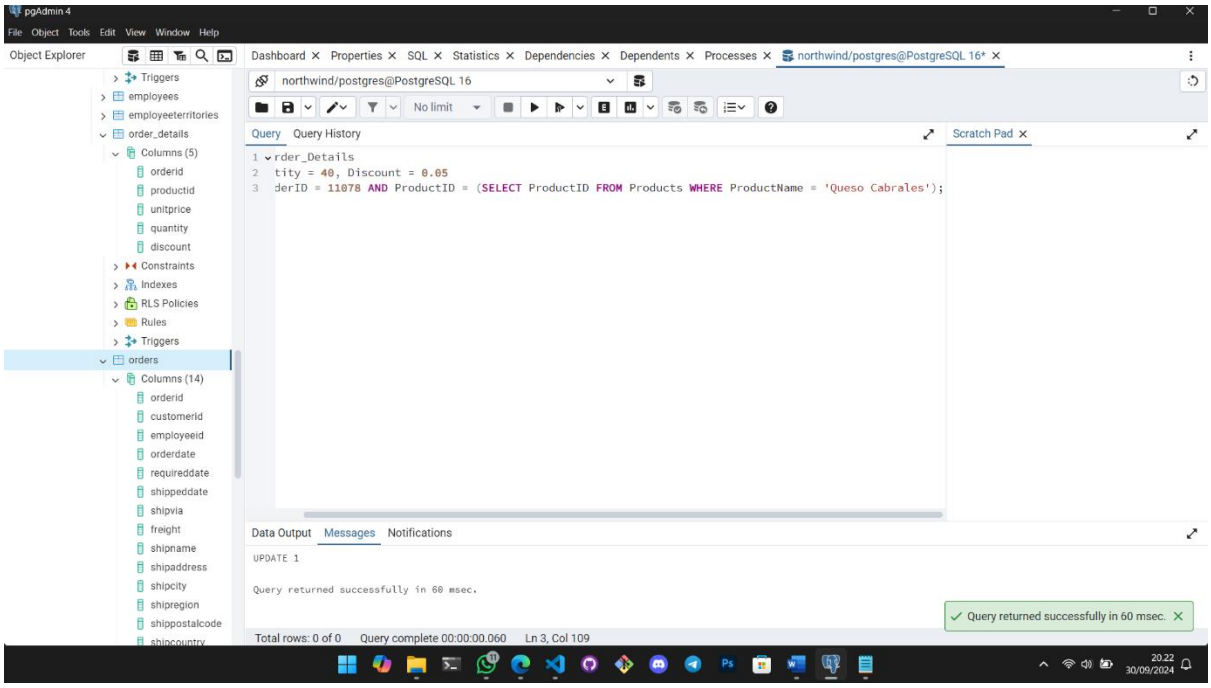


SQL 3.4

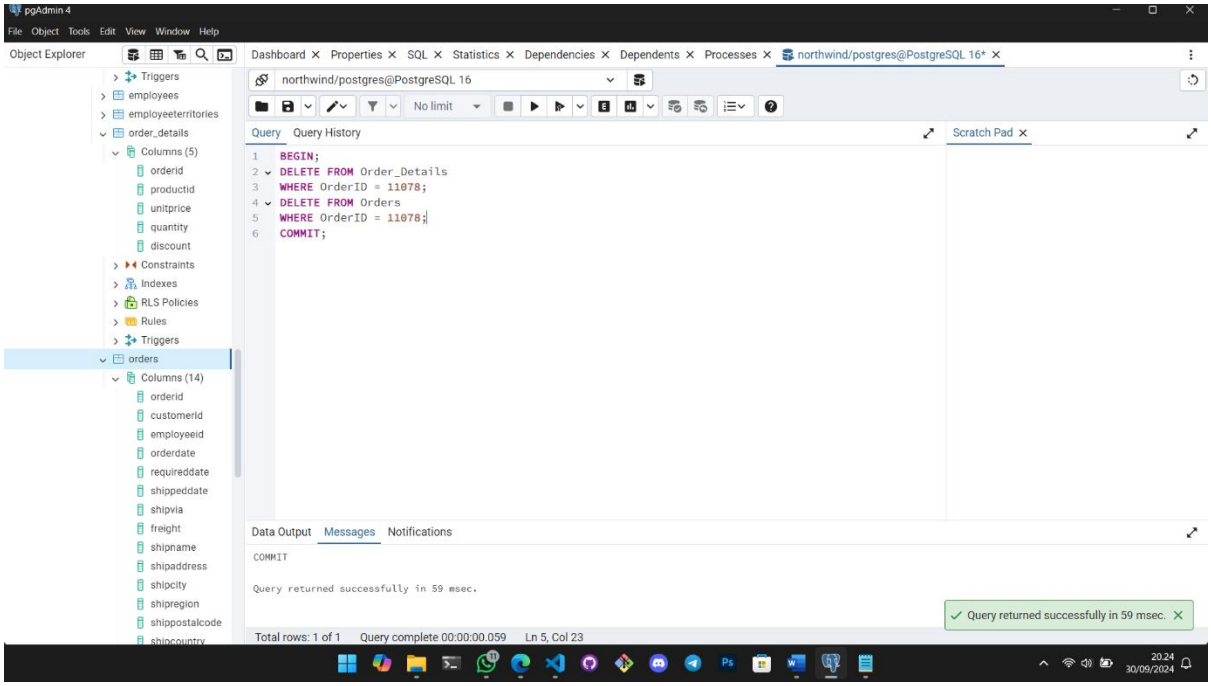
1. Insert an order detail for order we just created. Make it a quantity of 20 of Queso Cabrales (you will have to look up id) with a price of $14



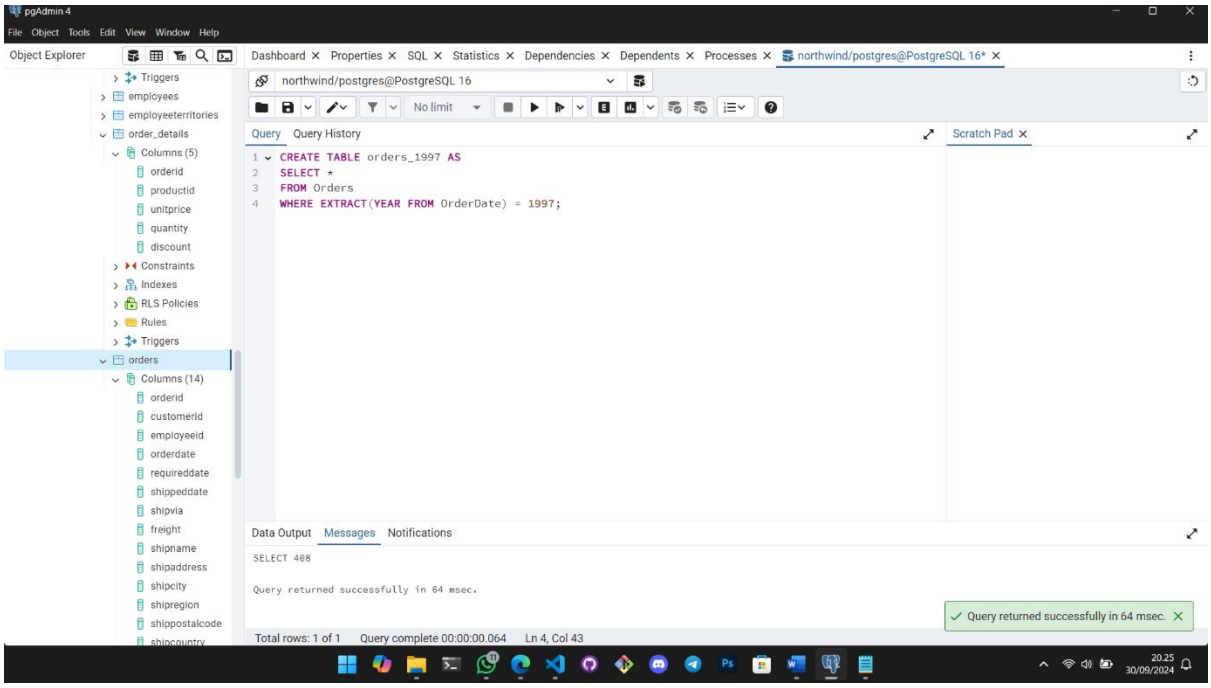
2. They also want 40 Queso Cabrales instead of 20 and we are giving a discount of 0.05. (trick is the WHERE clause to make sure we update the right order_details since there is no order detail id field)



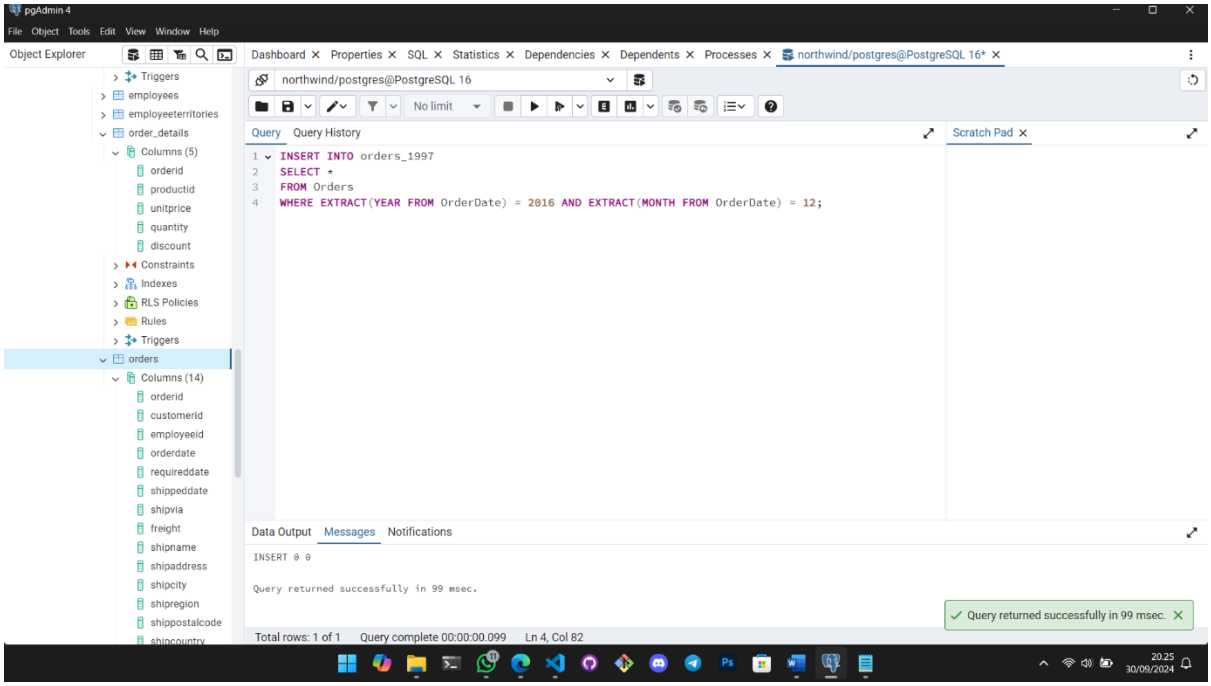
3. Delete the order for the customer using the order id 11078



4. Backup orders in the year 1997 to a new table orders_1997



5. Add orders from December 2016 to table orders_1997


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
Dashboard Properties SQL Statistics Dependencies Dependents Processes northwind/postgres@PostgreSQL 16*
northwind/postgres@PostgreSQL 16
No limit
Query Query History
Scratch Pad
INSERT INTO orders_1997
SELECT *
FROM Orders
WHERE EXTRACT(YEAR FROM OrderDate) = 2016 AND EXTRACT(MONTH FROM OrderDate) = 12;
Data Output Messages Notifications
INSERT 0 0
Query returned successfully in 99 msec.
Query returned successfully in 99 msec.
Total rows: 1 of 1 Query complete 00:00:00.099 Ln 4, Col 82
Triggers
employees
employeeterritories
order_details
Columns (5)
orderid
productid
unitprice
quantity
discount
Constraints
Indexes
RLS Policies
Rules
Triggers
orders
Columns (14)
orderid
customerid
employeeid
orderdate
requireddate
shippeddate
shipvia
freight
shipname
shipaddress
shipcity
shipregion
shippostalcode
shipcountry
20.25
30/09/2024