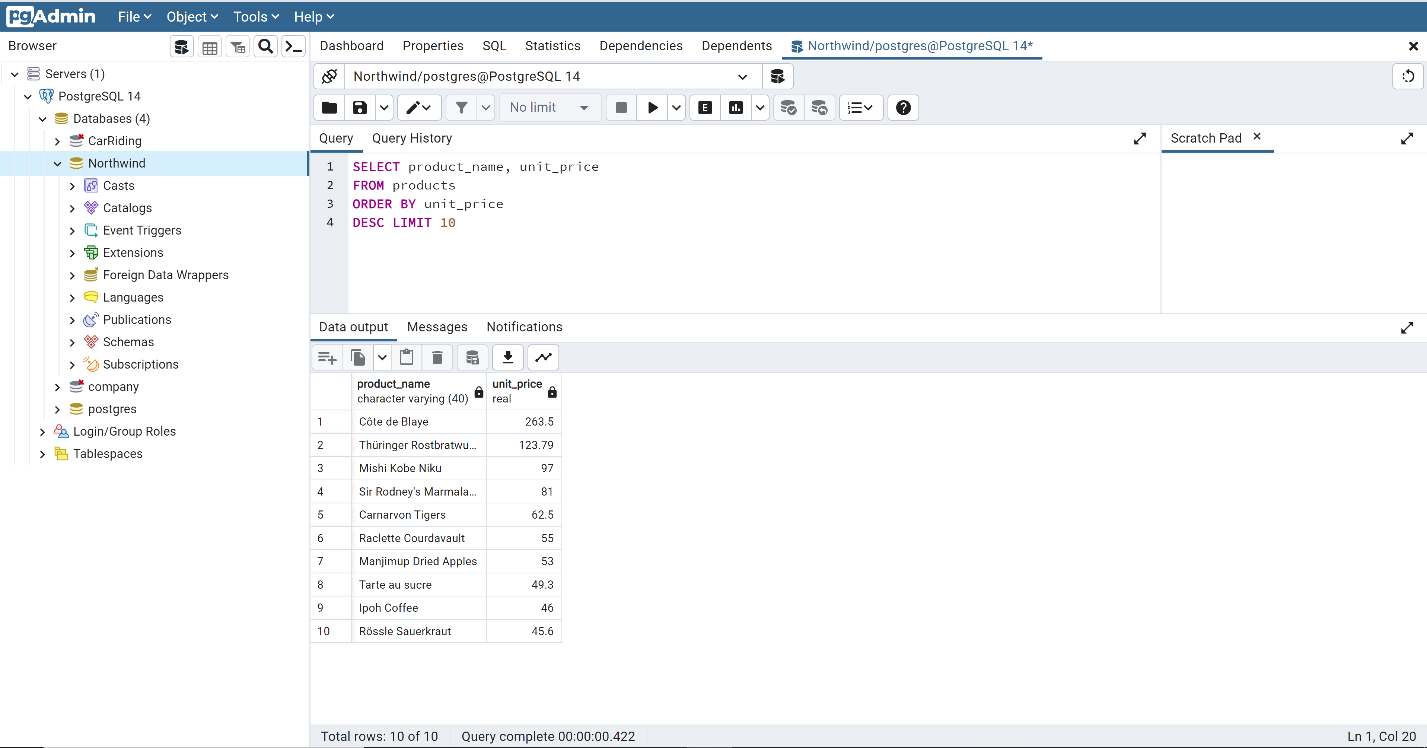
1. Write a query to get the ten most expensive Products (name and unit price).

SELECT product\_name, unit\_price

FROM products

ORDER BY unit\_price

DESC LIMIT 10



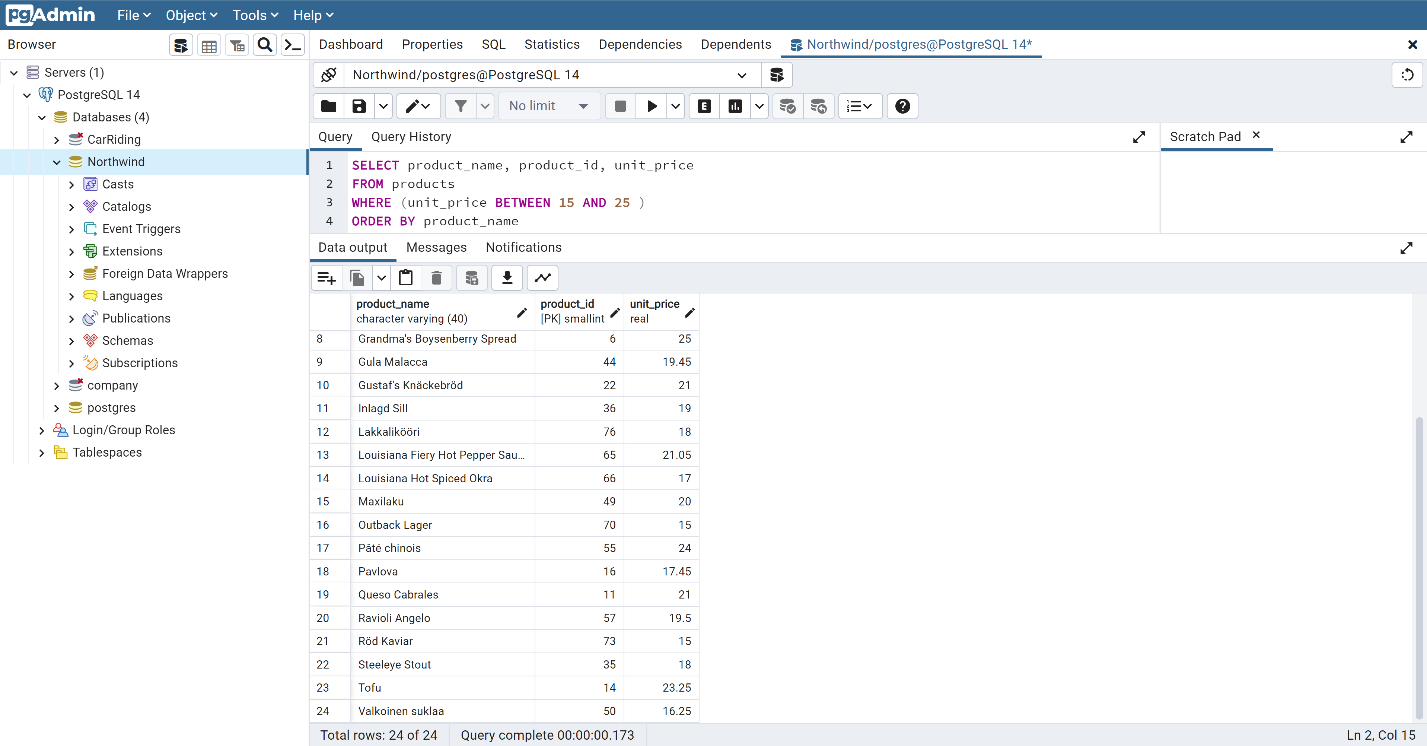
2. Write a query to get Product list (id, name, unit price) where products cost between $15 and $25

SELECT product\_name, product\_id, unit\_price

FROM products

WHERE (unit\_price BETWEEN 15 AND 25 )

ORDER BY product\_name

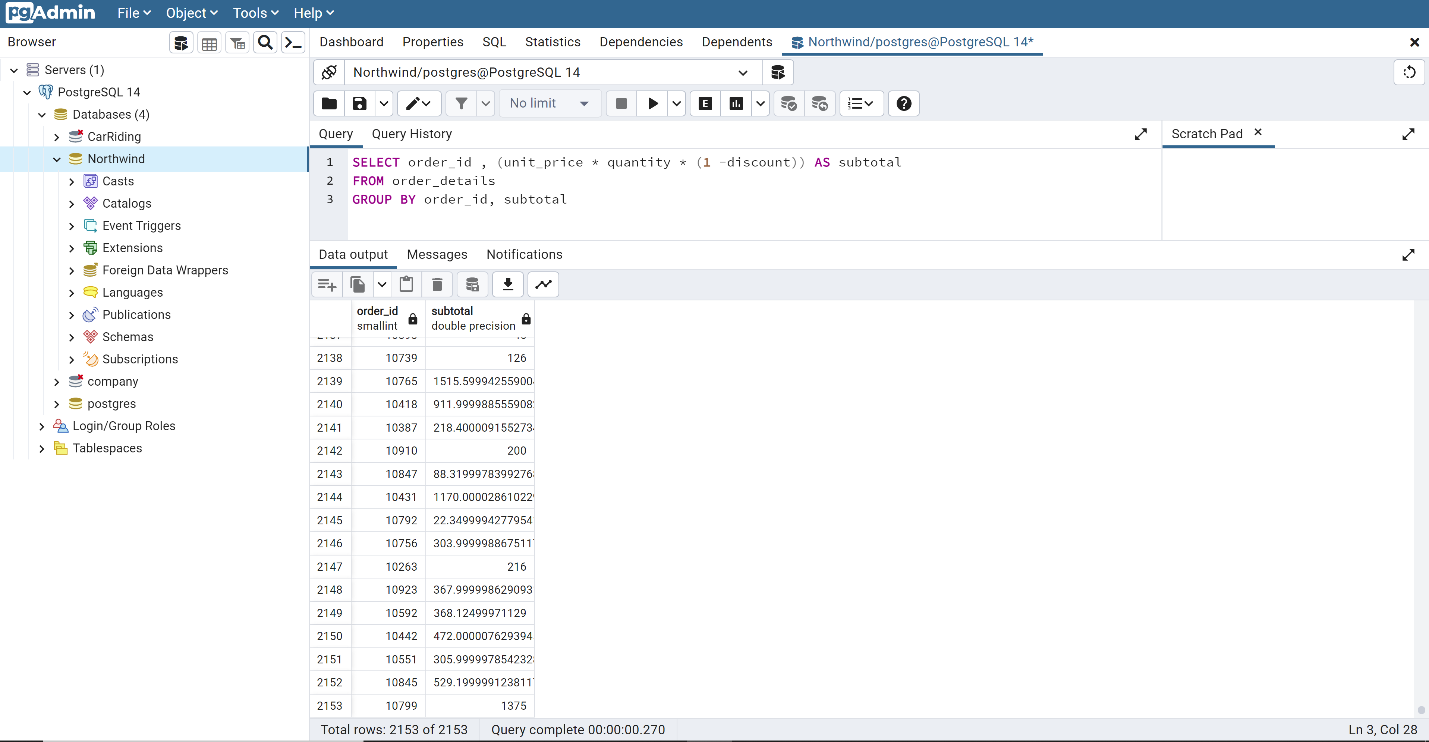


3. For each order, calculate a subtotal for each Order (identified by OrderID). This is a simple query using GROUP BY to aggregate data for each order. The formula to calculate the subtotal for each order is UnitPrice \* Quantity \* (1 - Discount).

SELECT order\_id , (unit\_price \* quantity \* (1 -discount)) AS subtotal

FROM order\_details

GROUP BY order\_id, subtotal



4. For each employee, get their sales amount, broken down by country name. We need to retrieve the country, last name, first name, shipped date, order id and the subtotal as sales\_amount (using this formula: Unit\_Price \* Quantity \* (1 - Discount)).

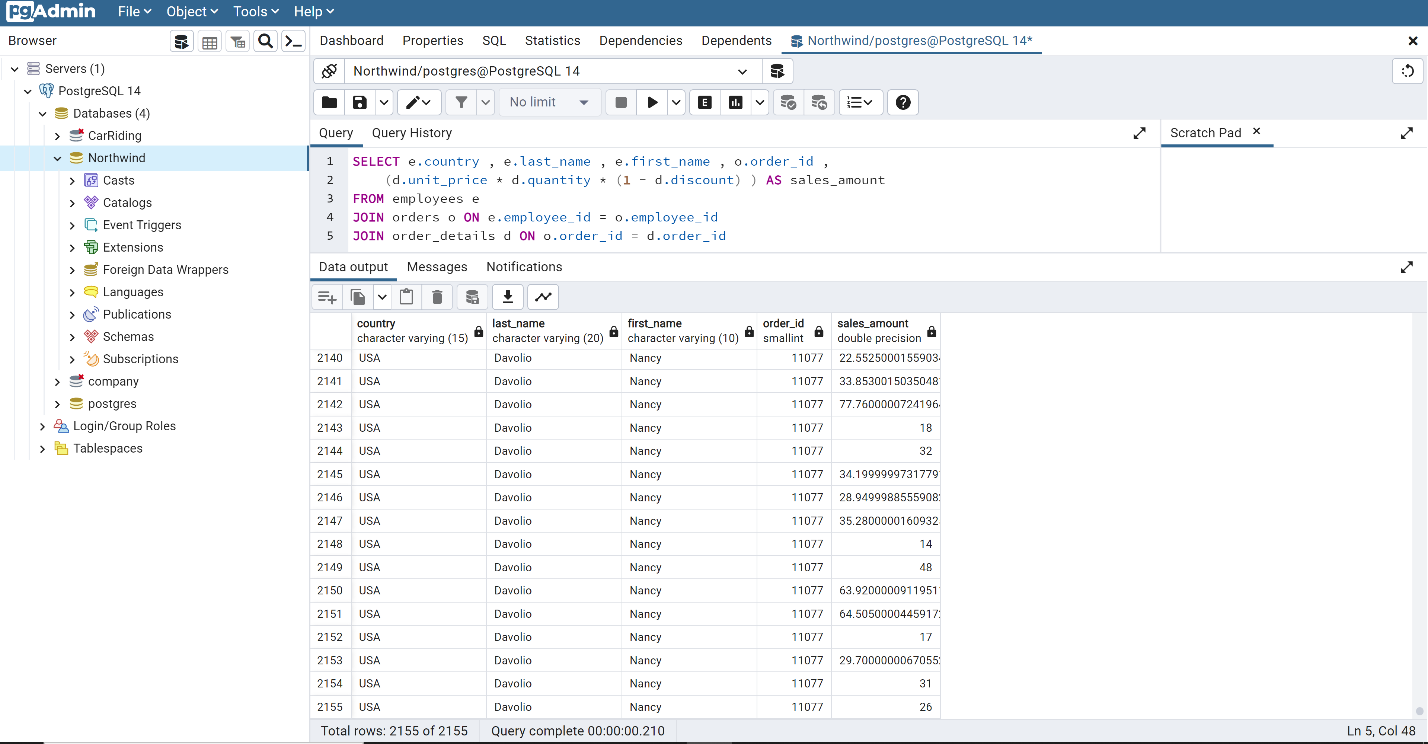
SELECT e.country , e.last\_name , e.first\_name , o.order\_id ,

(d.unit\_price \* d.quantity \* (1 - d.discount) ) AS sales\_amount

FROM employees e

JOIN orders o ON e.employee\_id = o.employee\_id

JOIN order\_details d ON o.order\_id = d.order\_id



5. List the company name, contact name from Customers and Suppliers by City. In the result of the query you need to specify who are the customers and who are the suppliers.

SELECT c.company\_name AS customer, c.contact\_name AS customer\_contact\_name, c.city AS customer\_city,

s.company\_name AS supplier, s.contact\_name AS supplier\_contact\_name, s.city AS supplier\_city

FROM customers c

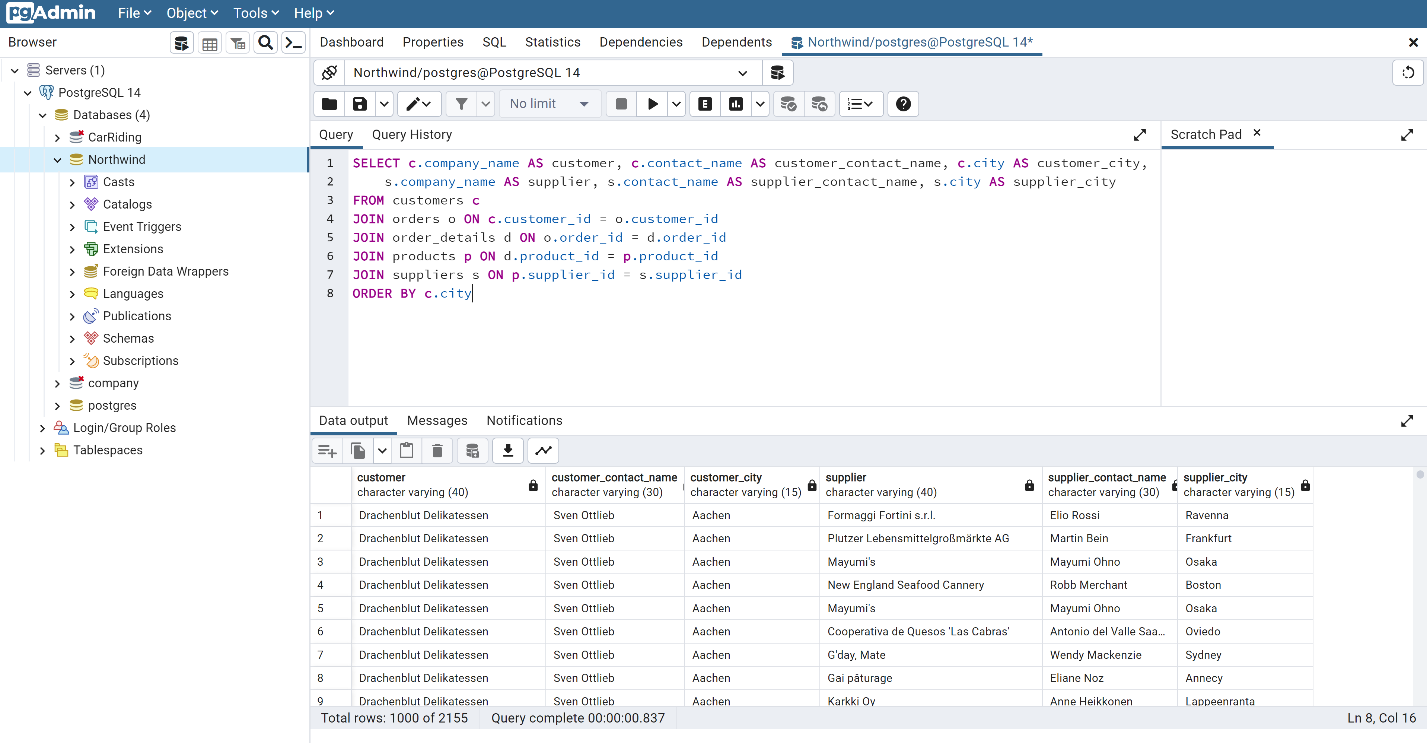
JOIN orders o ON c.customer\_id = o.customer\_id

JOIN order\_details d ON o.order\_id = d.order\_id

JOIN products p ON d.product\_id = p.product\_id

JOIN suppliers s ON p.supplier\_id = s.supplier\_id

ORDER BY c.city

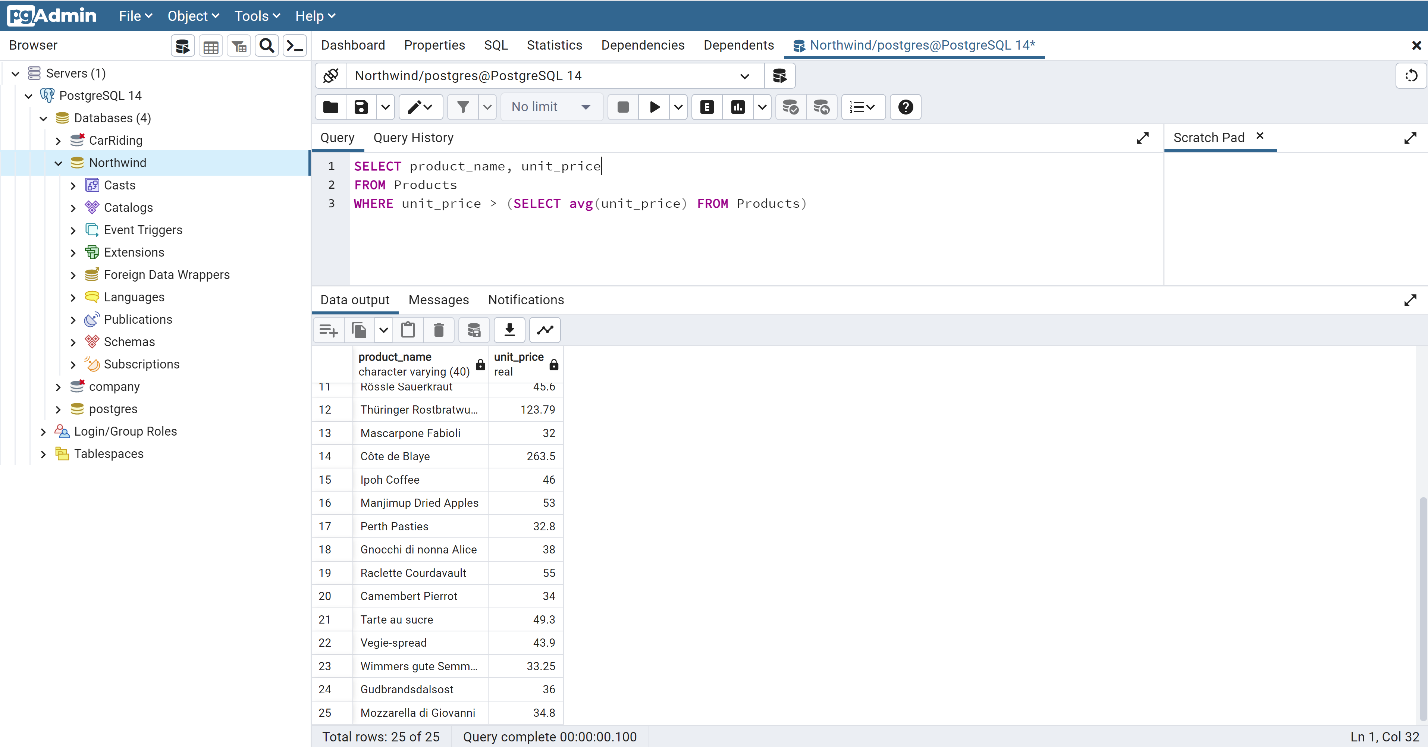


6. List the product name and unit price for products above the average price

SELECT product\_name, unit\_price

FROM Products

WHERE unit\_price > (SELECT avg(unit\_price) FROM Products)



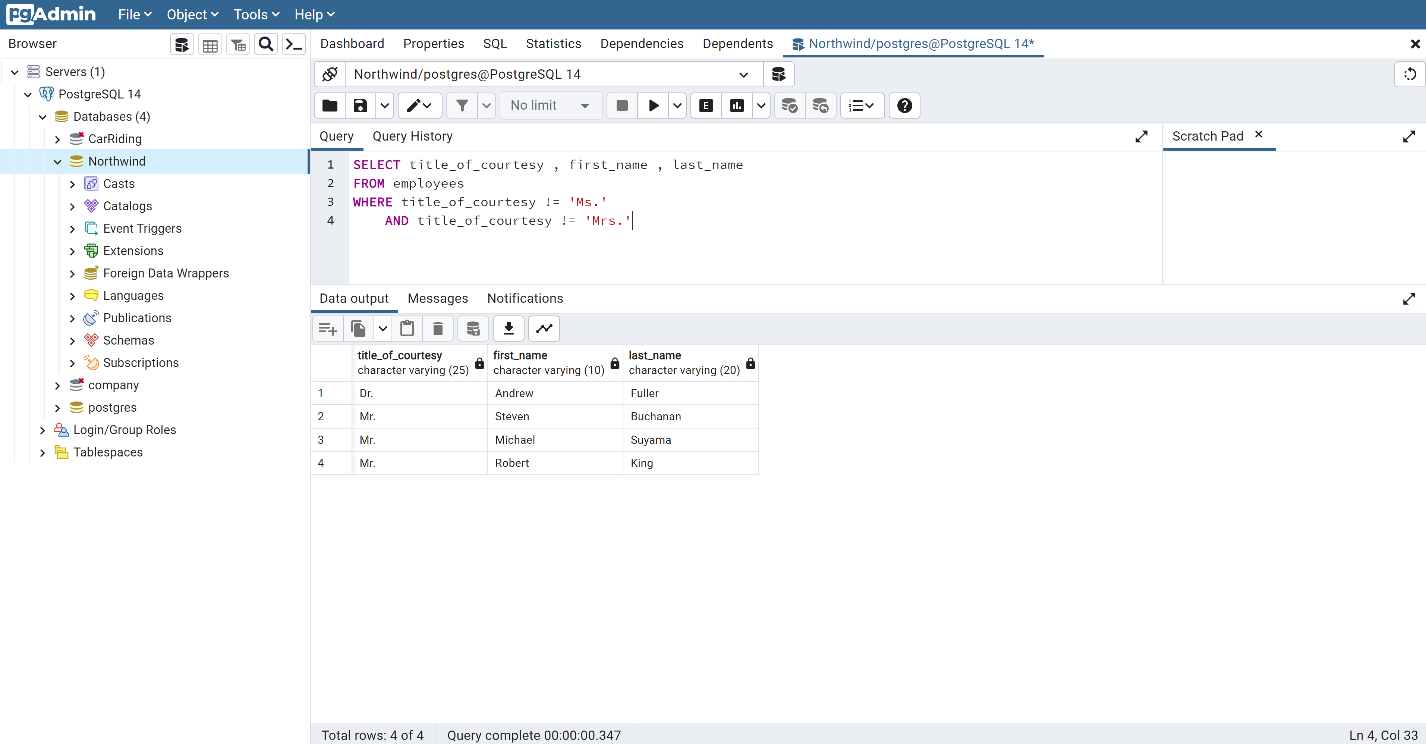
7. Create a report showing the title of courtesy and the first and last name of all employees whose title of courtesy is not "Ms." or "Mrs.".

SELECT title\_of\_courtesy , first\_name , last\_name

FROM employees

WHERE title\_of\_courtesy != 'Ms.'

AND title\_of\_courtesy != 'Mrs.'



8. Create a report showing the Order ID, the name of the company that placed the order,and the first and last name of the associated employee. Only show orders placed after January 1, 1998 that shipped after they were required. Sort by Company Name.

SELECT order\_id, company\_name, first\_name, last\_name,shipped\_date, required\_date

FROM orders

LEFT JOIN employees on employees.employee\_id = orders.employee\_id

LEFT JOIN customers on customers.customer\_id = orders.customer\_id

WHERE order\_date > (

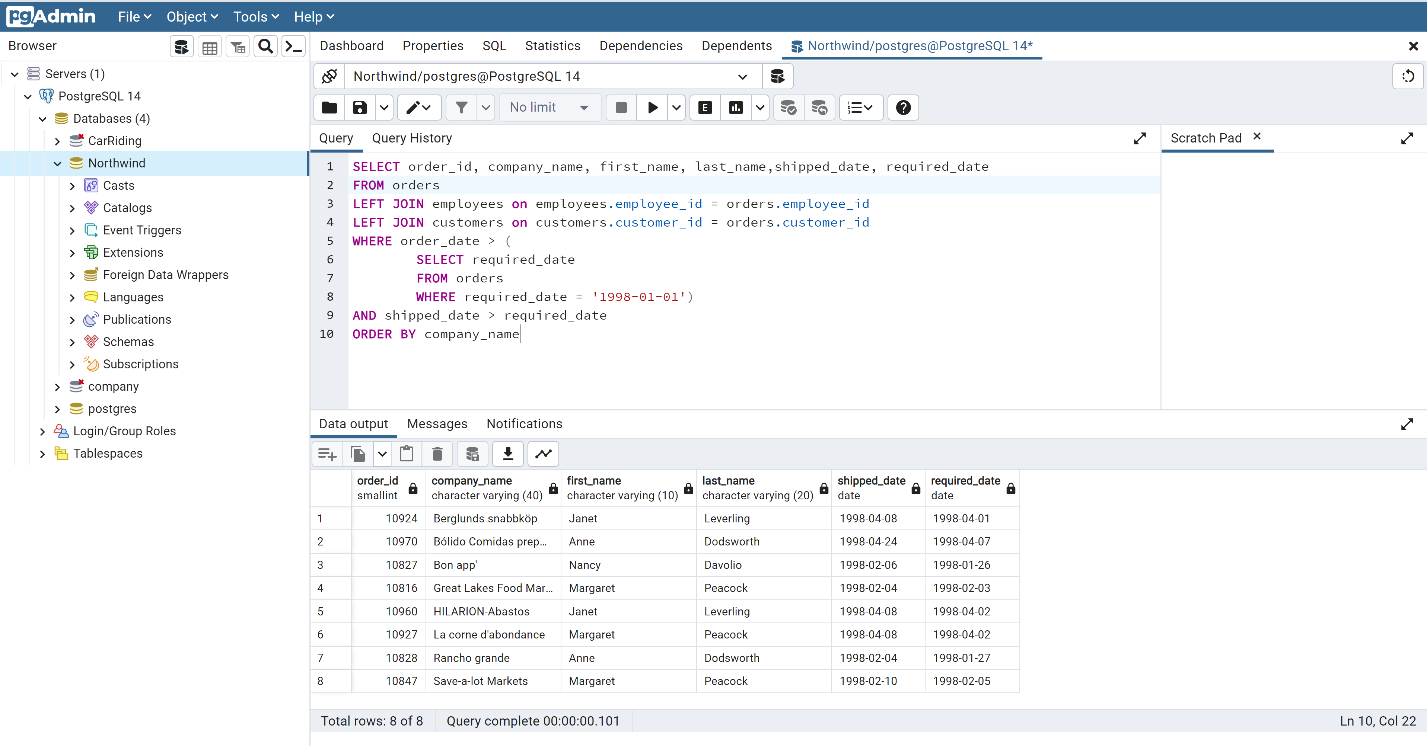
SELECT required\_date

FROM orders

WHERE required\_date = '1998-01-01')

AND shipped\_date > required\_date

ORDER BY company\_name



9. Create a report showing the customer ID and company name, employee id, firstname and lastname, and the order id and a conditional column called "Shipped" that displays "On Time" if the order was shipped on time and "Late" if the order was shipped late.

SELECT C.customer\_id, C.company\_name, E.employee\_id, E.first\_name,

E.last\_name, orders.order\_id, orders.required\_date, orders.shipped\_date,

CASE

WHEN shipped\_date > required\_date THEN 'Late'

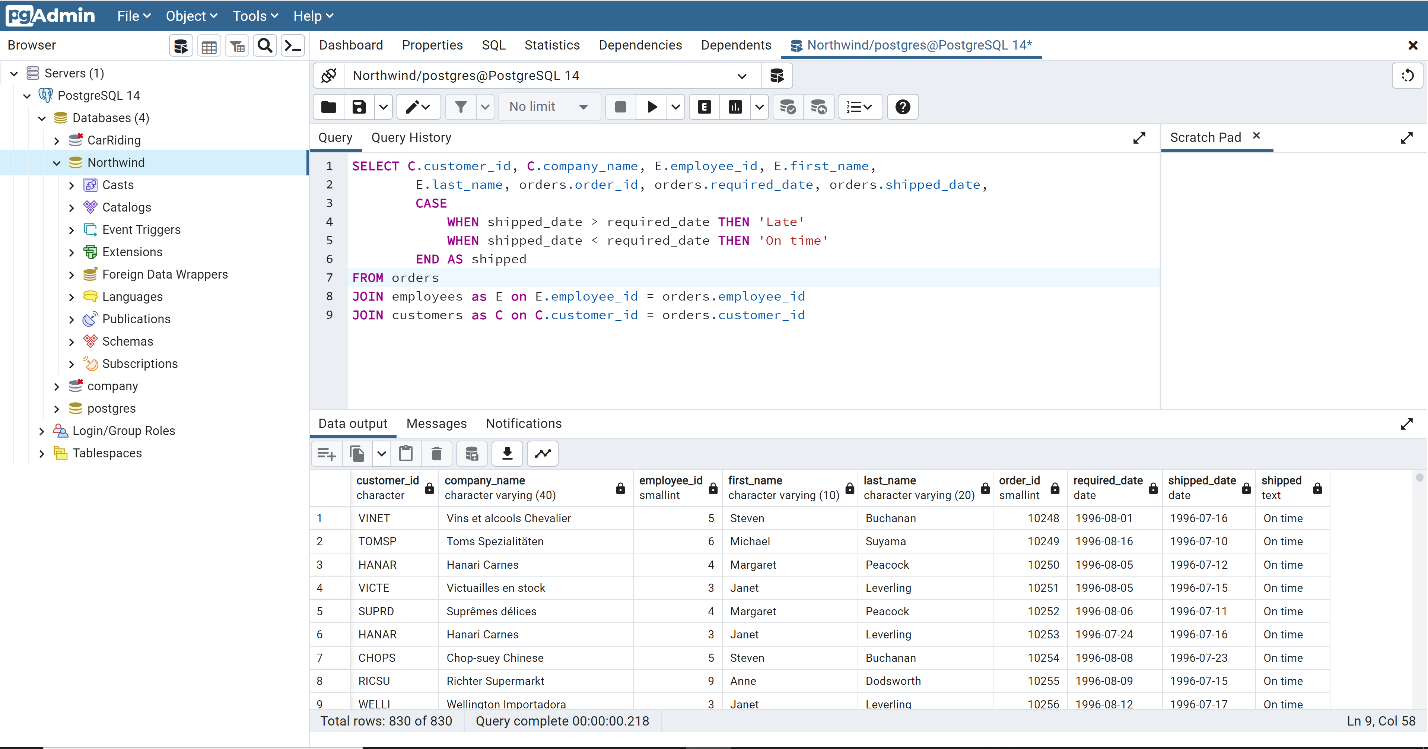
WHEN shipped\_date < required\_date THEN 'On time'

END AS shipped

FROM orders

JOIN employees as E on E.employee\_id = orders.employee\_id

JOIN customers as C on C.customer\_id = orders.customer\_id



10. Raise the unit price of all products in the Products table by 10% for all products that are out of stock. This should affect 5 rows.

SELECT unit\_price + (unit\_price \*.10) as out\_of\_stock

FROM products

WHERE units\_in\_stock = 0

