Day 7 Assignment

PostgreSQL

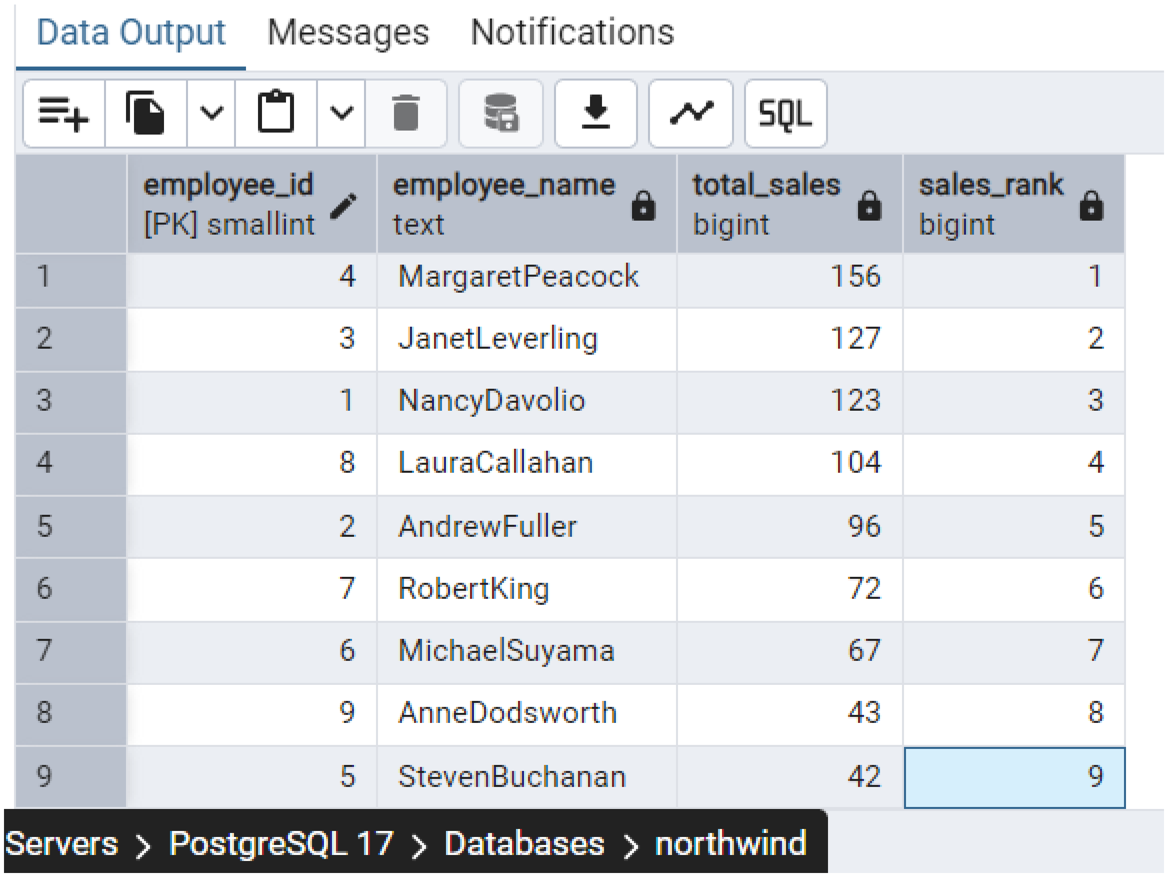
1. Rank employees by their total sales

(Total sales = Total no of orders handled, JOIN employees and orders table)

Query:

```
SELECT e.employee_id, e.first_name ||''|| e.last_name As employee_name,
    COUNT(o.order_id) AS total_sales,
    RANK() OVER (ORDER BY COUNT(o.order_id) DESC) AS sales_rank
FROM employees e
JOIN  orders o ON e.employee_id = o.employee_id
GROUP BY e.employee_id, employee_name
ORDER BY total_sales DESC;
```

OUTPUT:


Data Output Messages Notifications

	employee_id [PK] smallint	employee_name text	total_sales bigint	sales_rank bigint
1	4	MargaretPeacock	156	1
2	3	JanetLeverling	127	2
3	1	NancyDavolio	123	3
4	8	LauraCallahan	104	4
5	2	AndrewFuller	96	5
6	7	RobertKing	72	6
7	6	MichaelSuyama	67	7
8	9	AnneDodsworth	43	8
9	5	StevenBuchanan	42	9

Servers > PostgreSQL 17 > Databases > northwind

2. Compare current order's freight with previous and next order for each customer.

(Display order_id, customer_id, order_date, freight,

Use lead(freight) and lag(freight).

Query:

OUTPUT:

3. Show products and their price categories, product count in each category, avg price:

(HINT:

· Create a CTE which should have price_category definition:

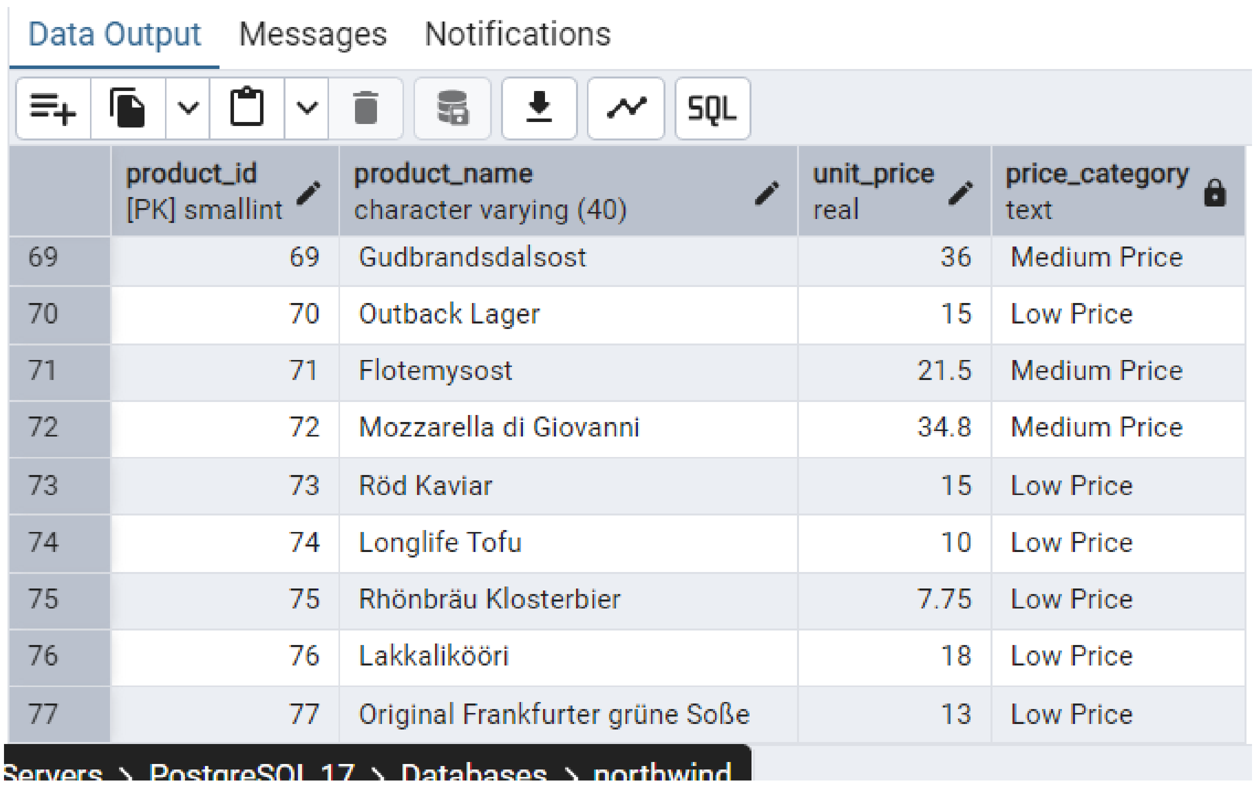
```
        WHEN unit_price < 20 THEN 'Low Price'
          WHEN unit_price < 50 THEN 'Medium Price'
          ELSE 'High Price'
```

- In the main query display: price_category, product_count in each price_category, ROUND(AVG(unit_price)::numeric, 2) as avg_price)

Query:

```
WITH categorized_products AS (
   SELECT product_id, product_name, unit_price,
      CASE
         WHEN unit_price < 20 THEN 'Low Price'
         WHEN unit_price < 50 THEN 'Medium Price'
         ELSE 'High Price'
      END AS price_category
   FROM products
)
SELECT price_category, COUNT(*) AS product_count,
ROUND(AVG(unit_price)::numeric, 2) AS avg_price
FROM categorized_products
GROUP BY price_category
ORDER BY price_category;
```

OUTPUT:



	product_id [PK] smallint	product_name character varying (40)	unit_price real	price_category text
69	69	Gudbrandsdalsost	36	Medium Price
70	70	Outback Lager	15	Low Price
71	71	Flotemysost	21.5	Medium Price
72	72	Mozzarella di Giovanni	34.8	Medium Price
73	73	Röd Kaviar	15	Low Price
74	74	Longlife Tofu	10	Low Price
75	75	Rhönbräu Klosterbier	7.75	Low Price
76	76	Lakkalikööri	18	Low Price
77	77	Original Frankfurter grüne Soße	13	Low Price

Servers › PostgreSQL 17 › Databases › northwind