

Common Table Expressions

Step 1

1-

Dashboard Properties SQL Statistics Dependencies Dependents Processes Task 3.9.sql

Rockbuster/postgres@PostgreSQL 15

No limit

Query Query History

```
1 EXPLAIN
2 WITH top_5_customers (customer_id, first_name, last_name, country, city, total_amount) AS
3 (SELECT B.customer_id, B.first_name AS first_name, B.last_name AS last_name, C.country, D.city, SUM(P.amount) AS total_amount
4 FROM customer B
5 INNER JOIN address A ON B.address_id = A.address_id
6 INNER JOIN city D ON A.city_id = D.city_id
7 INNER JOIN country C ON D.country_id = C.country_id
8 INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
9 ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
10 GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount DESC limit 5)
11 SELECT AVG(total_amount) AS avg_total_amount FROM top_5_customers;
12
```

Data Output Messages Notifications

	avg_total_amount numeric
1	105.5540000000000000

```
WITH top_5_customers (customer_id, first_name, last_name, country, city, total_amount) AS
(SELECT B.customer_id, B.first_name AS first_name, B.last_name AS last_name, C.country, D.city, SUM(P.amount) AS total_amount
FROM customer B
INNER JOIN address A ON B.address_id = A.address_id
INNER JOIN city D ON A.city_id = D.city_id
INNER JOIN country C ON D.country_id = C.country_id
INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount DESC limit 5)
SELECT AVG(total_amount) AS avg_total_amount FROM top_5_customers;
```

2-

Dashboard Properties SQL Statistics Dependencies Dependents Processes Task 3.9.sql

Rockbuster/postgres@PostgreSQL 15

No limit

Query Query History

```
13 EXPLAIN
14 WITH top_5_customers (customer_id, first_name, last_name, country, city, total_amount) AS
15 (SELECT B.customer_id, B.first_name AS first_name, B.last_name AS last_name, C.country, D.city, SUM(P.amount) AS total_amount
16 FROM customer B
17 INNER JOIN address A ON B.address_id = A.address_id
18 INNER JOIN city D ON A.city_id = D.city_id
19 INNER JOIN country C ON D.country_id = C.country_id
20 INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
21 ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
22 GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount DESC limit 5)
23 SELECT C.country, COUNT(B.customer_id) AS all_customer_count, COUNT(top_5_customers) AS top_customer_count
24 FROM customer B
25 INNER JOIN address D ON B.address_id = D.address_id
26 INNER JOIN city A ON A.city_id = D.city_id
27 INNER JOIN country C ON A.country_id = C.country_id
28 LEFT JOIN top_5_customers
29 ON B.customer_id=top_5_customers.customer_id
30 GROUP BY C.country
31 HAVING COUNT(top_5_customers)>0
32 ORDER BY COUNT(top_5_customers),
33 COUNT(B.customer_id) DESC;
```

Scratch Pad

Data Output Messages Notifications

#	country	all_customer_count	top_customer_count
1	India	60	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1

Total rows: 5 of 5 Query complete 00:00:00.081 Ln 33, Col 27

```
WITH top_5_customers (customer_id, first_name, last_name, country, city, total_amount) AS

(SELECT B.customer_id, B.first_name AS first_name, B.last_name AS last_name, C.country, D.city, SUM(P.amount) AS total_amount

FROM customer B

INNER JOIN address A ON B.address_id = A.address_id

INNER JOIN city D ON A.city_id = D.city_id

INNER JOIN country C ON D.country_id = C.country_id

INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN

('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')

GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount DESC limit 5)

SELECT C.country, COUNT(B.customer_id) AS all_customer_count, COUNT(top_5_customers) AS top_customer_count

FROM customer B

INNER JOIN address D ON B.address_id = D.address_id

INNER JOIN city A ON A.city_id = D.city_id

INNER JOIN country C ON A.country_id = C.country_id

LEFT JOIN top_5_customers

ON B.customer_id=top_5_customers.customer_id

GROUP BY C.country

HAVING COUNT(top_5_customers)>0

ORDER BY COUNT(top_5_customers),

COUNT(B.customer_id) DESC;
```

3-

To start with, I used the Entity relationship diagram that I created in the task 3.2 to search for what all tables and attributes I will be needing to answer the business question. At first, I copied the query that I wrote for task 3.8; then, I had to select the columns that I want to display for which I used the select command, and did some computations as required. Lastly, to sum it up I got the whole query together and ran it and examined the results.

At first, I copied the query that I wrote for task 3.8; then, to select the columns that I want to display I used the select command and to find the average of the total amount paid I used the function AVG() and renamed the column as average.; after this I decided on which columns I can group the table and selected the values to be in descending order and as asked I displayed 5 values.

Step 2

- 1- CTEs would perform better, because CTEs can be written once and can be called many times rather than subqueries
- 2- For the first query it was 61.33, as for the second query the cost was 133.77
- 3- After comparing the performance between Subqueries in Task 8 with CTE in Task 9, the cost is the same
61.33 for the first query and 133.77 for the second
- 4- The speed was surprising

Step 3

It took some time to replace sub-queries with CTE, I think the practice will solve this problem