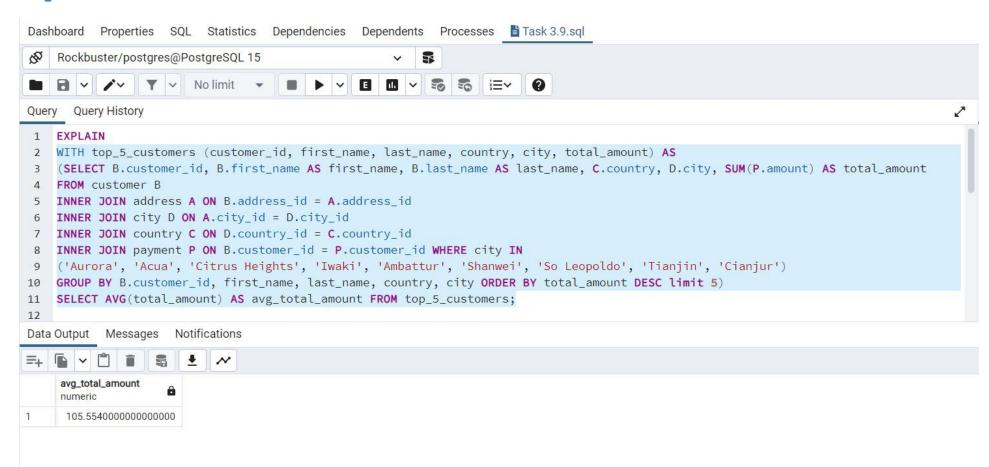
Common Table Expressions

Step 1

1-



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

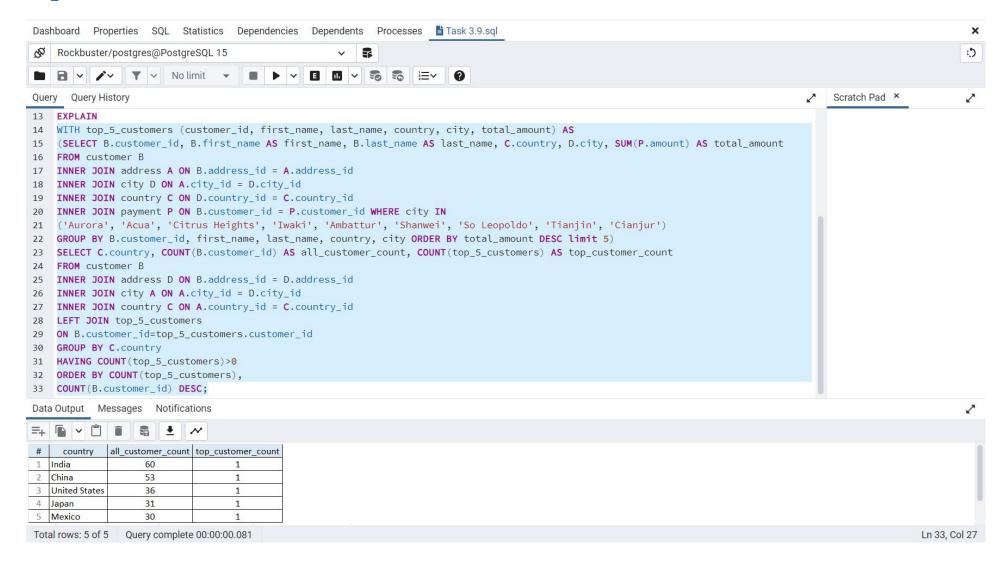
INNER JOIN country C ON D.country_id = C.country_id

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;



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;

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 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