3.9 Common Table Expressions

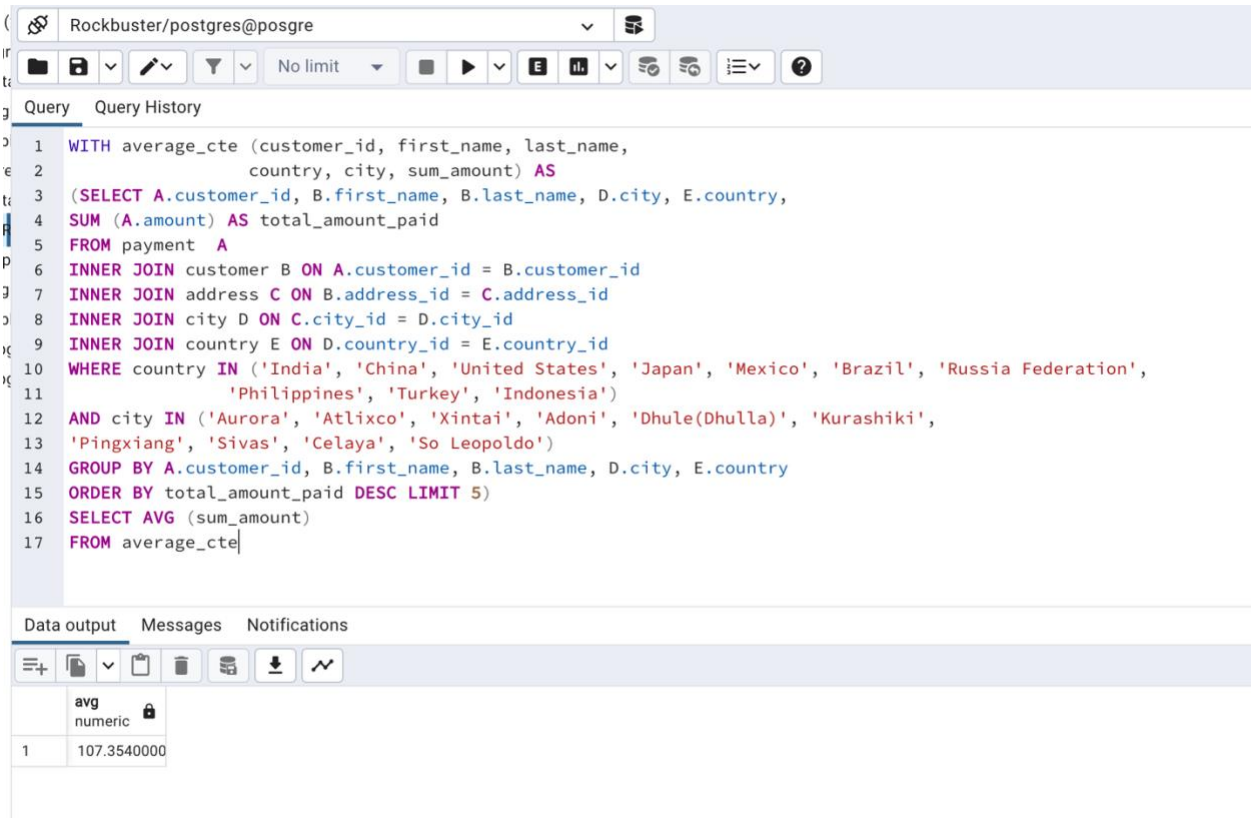
Step 1: Answer the business questions from step 1 and 2 of task 3.8 using CTEs

1. Rewrite your queries from steps 1 and 2 of task 3.8 as CTEs.
2. Copy-paste your CTEs and their outputs into your answers document.
3. Write 2 to 3 sentences explaining how you approached this step, for example, WITH customer_full_address (customer_id, first_name, last_name,

address, city_id, city, country)

Step 1

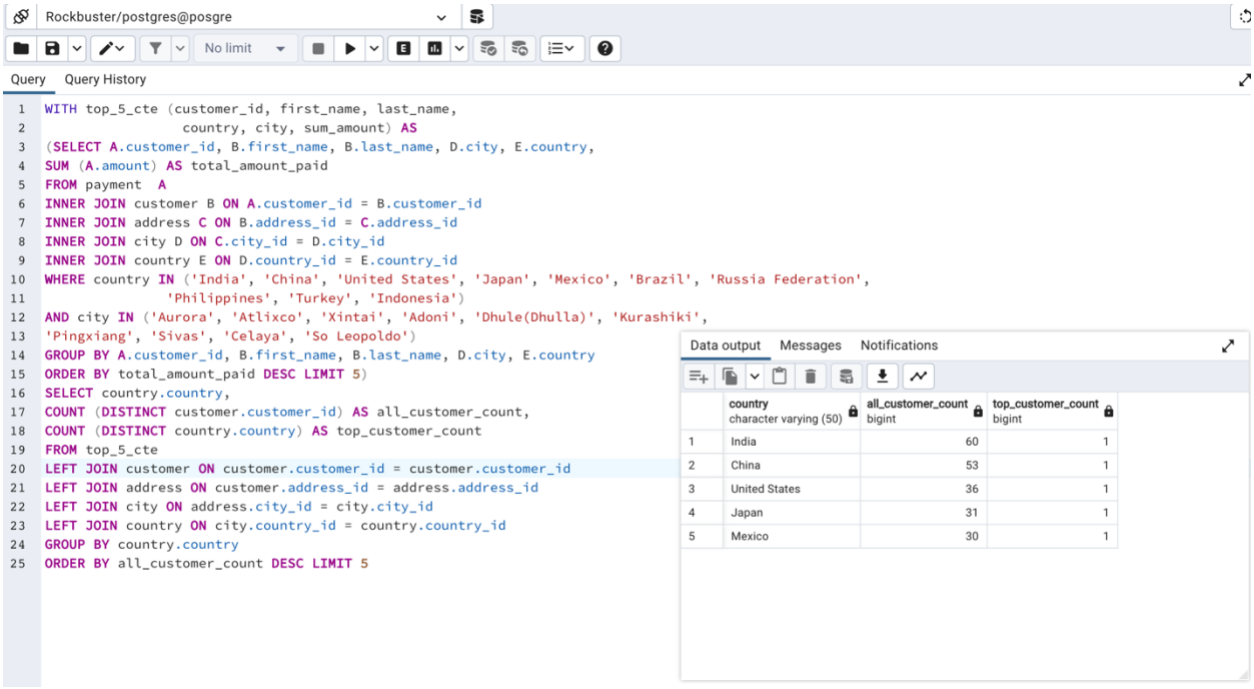
```
WITH average_cte (customer_id, first_name, last_name,
                                    country, city, sum_amount) AS
(SELECT A.customer_id, B.first_name, B.last_name, D.city, E.country,
SUM (A.amount) AS total_amount_paid
FROM payment  A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.country_id
WHERE country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russia Federation',
                        'Philippines', 'Turkey', 'Indonesia')
AND city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulla)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY total_amount_paid DESC LIMIT 5)
SELECT AVG (sum_amount)
FROM average_cte
```



Step 2

```
WITH top_5_cte (customer_id, first_name, last_name,
                                country, city, sum_amount) AS
(SELECT A.customer_id, B.first_name, B.last_name, D.city, E.country,
SUM (A.amount) AS total_amount_paid
FROM payment  A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.country_id
WHERE country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russia Federation',
                        'Philippines', 'Turkey', 'Indonesia')
AND city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulla)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY total_amount_paid DESC LIMIT 5)
SELECT country.country,
COUNT (DISTINCT customer.customer_id) AS all_customer_count,
COUNT (DISTINCT country.country) AS top_customer_count
FROM top_5_cte
LEFT JOIN customer ON customer.customer_id = customer.customer_id
LEFT JOIN address ON customer.address_id = address.address_id
```

LEFT JOIN city ON address.city_id = city.city_id
LEFT JOIN country ON city.country_id = country.country_id
GROUP BY country.country
ORDER BY all_customer_count DESC LIMIT 5



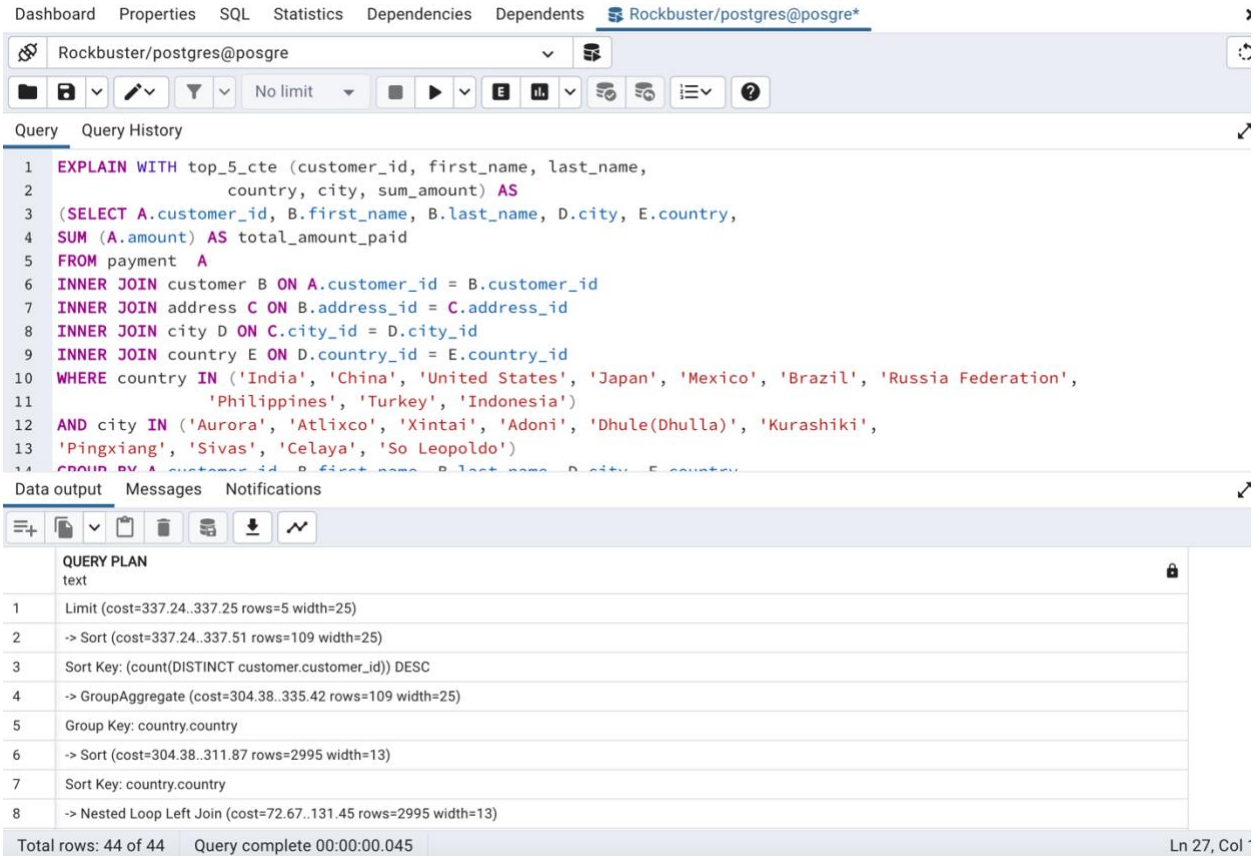
Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.

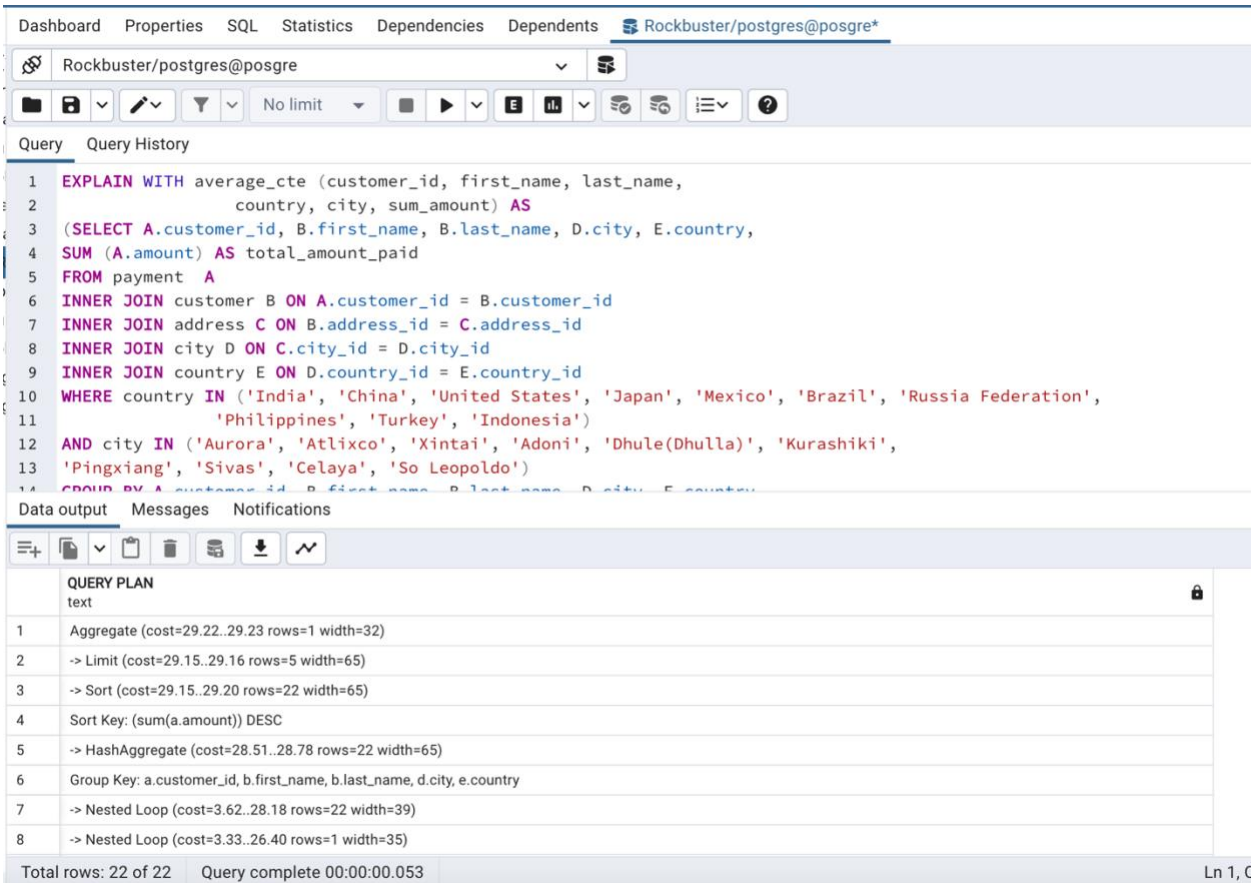
I define the CTE with the 'WITH' clause and gave it an appropriate expression name. Then, I listed columns that will be listed in the CTE and used the AS keyword. Then used SELECT again to display the result.

Step 2: Compare the performance of your CTEs and subqueries.

1. Which approach do you think will perform better and why?
2. Compare the costs of all the queries by creating query plans for each one.
3. The EXPLAIN command gives you an *estimated* cost. To find out the actual speed of your queries, run them in pgAdmin 4. After each query has been run, a pop-up window will display its speed in milliseconds.
4. Did the results surprise you? Write a few sentences to explain your answer.

When joining the customer table with city and country, only 2 columns are added each time, whereas when the payment table is joined with the customer data, it becomes much, much larger. Doing aggregations with a larger table is the biggest contributor to that high cost as well. Also, when the large table of customer data was compiled via JOINs, there was no filter,
I am surprised that using subqueries instead of CTE is such a lower cost. I imagine it's because each subquery is only identifying exactly what it needs and because the scope of each subquery is small, aggregations don't cost as much. The time it took to run each query varied; this is because the database is stored locally and it's dependent on the other processes being executed by my computer





Step 3:

Write 1 to 2 paragraphs on the challenges you faced when replacing your subqueries with CTEs.

The problems I've run into include determining what is needed to create a CTE. I know that unlike subrequests, CTEs are defined at the beginning of the request. Also, writing an entirely new SELECT statement that queries a temporary table created with a CTE seems odd at first. However, if you spend more time reading the syntax, everything will be much easier to understand.