Answer 3.9

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.

```
WITH top_5_customers (customer_id, first_name, last_name, city, country, total_amo
unt) AS
(SELECT A.customer_id,
A.first_name,
A.last_name,
C.city,
D.country,
SUM(E.amount) AS total_amount
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D ON C.country_id = D.country_id
INNER JOIN payment E ON A.customer_id = E.customer_id
WHERE city IN('Aurora', 'Shimoga', 'Aparecida de Goinia', 'Emeishan', 'Pontianak',
'Tarsus', 'Atlixco', 'Zalantun', 'Rio Claro', 'Tokat')
GROUP BY A.customer_id, C.city, D.country
ORDER BY total_amount DESC
LIMIT 5)
SELECT AVG(total_amount) AS avg_amount_paid
FROM top_5_customers
```

avg_amount_paid
120.322

```
WITH top_5_customers (customer_id, first_name, last_name, city, country, total_amo
unt) AS
(SELECT customer.customer_id,
customer.first_name,
customer.last_name,
city.city,
country.country,
SUM(payment.amount) AS total_amount
FROM customer
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city ON address.city_id = city.city_id
INNER JOIN country ON city.country_id = country.country_id
INNER JOIN payment ON customer.customer_id = payment.customer_id
WHERE city IN('Aurora', 'Shimoga', 'Aparecida de Goinia', 'Emeishan', 'Pontianak',
'Tarsus', 'Atlixco', 'Zalantun', 'Rio Claro', 'Tokat')
```

```
GROUP BY customer.customer_id, city.city, country.country
ORDER BY total_amount DESC
LIMIT 5)
SELECT country.country,
COUNT (DISTINCT customer.customer_id) AS all_customer_count,
COUNT (DISTINCT top_5_customers.customer_id) AS top_customer_count
FROM customer
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city ON address.city_id = city.city_id
INNER JOIN country ON city.country_ID = country.country_ID
LEFT JOIN top_5_customers ON country.country = top_5_customers.country
GROUP BY country.country
ORDER BY top_customer_count DESC, all_customer_count DESC
LIMIT 5
```

country	all_customer_count	top_customer_count
China	53	1
United States	36	1
Mexico	30	1
Turkey	15	1
Indonesia	14	1

3. Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.
    a. I moved the subquery to the beginning of the query and prefaced it with the WITH AS statement. Then the remaining query just had to be adjusted to include the CTE reference.

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

1. Which approach do you think will perform better and why?
    a. I would assume for both to perform relatively similar, as they only include one subquery/CTE each and both the subquery and the CTE will run before the main query. If there had been multiple CTE and more nestled subqueries, it might be different. In that case I would assume the subquery heavy query to run slower but to require less memory capacity.
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.

a. Based on the query plans the CTE queries and the subqueries have the same estimated cost. When comparing the actual run time, the CTE version was faster on the first task and the subquery on the second. This follows from the remark that in terms of performance it cannot be assumed that one will be better than the other without having tested it before.

Step 3:

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

It was pretty straight forward and I wouldn’t say that there were any challenges for me in replacing the subqueries.