Task 3.9

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

- **Rewrite your queries from steps 1 and 2 of task 3.8 as CTEs.**
- **Copy-paste your CTEs and their outputs into your answers document.**
- **Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.**

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History

```
WITH average_amount_paid_cte AS
(SELECT A.customer_id, B.first_name, B.last_name, D.city, E.country,
SUM(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 E.country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
                    'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
AND D.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', '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 ROUND( AVG (total_amount_paid), 2) AS average_amount_paid
FROM average_amount_paid_cte;
```

Data Output | Explain | Messages | Notifications

	average_amount_paid numeric
1	107.35

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History

```
WITH top_five_customers_cte AS
(SELECT A.customer_id, B.first_name, B.last_name, D.city, E.country,
SUM(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 E.country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
                    'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
AND D.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', '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_five_customer
FROM top_five_customers_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 COUNT (country.country) DESC
LIMIT 5
```

Data Output | Explain | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_five_customer bigint
1	India	60	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1

I changed my inner query to a CTE by using the “WITH” clause, gave it a name. Then wrote the “SELECT” statement and added the joins.

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

- **Which approach do you think will perform better and why?**

I think CTEs might perform better due to its accessibility, and readability.

- **Compare the costs of all the queries by creating query plans for each one.**

Subquery 1

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Aggregate (cost=29.22..29.23 rows=1 width=32)

CTE Query 1

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Aggregate (cost=29.22..29.24 rows=1 width=32)

Subquery 2

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=344.72..344.74 rows=5 width=33)
2	[...] -> Sort (cost=344.72..345.00 rows=109 width=33)
3	[...] Sort Key: (count(country.country)) DESC
4	[...] -> GroupAggregate (cost=304.38..342.91 rows=109 width=33)

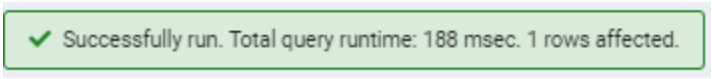
CTE Query 2

Data Output | Explain | Messages | Notifications

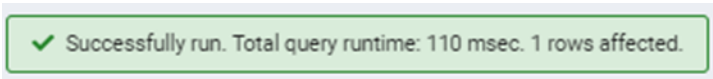
	QUERY PLAN text
1	Limit (cost=344.72..344.74 rows=5 width=33)
2	[...] -> Sort (cost=344.72..345.00 rows=109 width=33)
3	[...] Sort Key: (count(country.country)) DESC
4	[...] -> GroupAggregate (cost=304.38..342.91 rows=109 width=33)

- **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.**

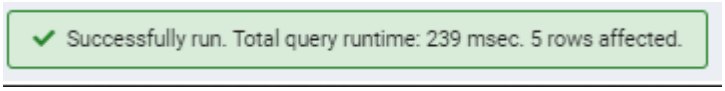
Subquery 1



CTE Query 1



Subquery 2



CTE Query 2

- **Did the results surprise you? Write a few sentences to explain your answer.**

The cost is same for subquery and CTE. CTEs has taken less time as compared to the subqueries. I assumed that CTEs would perform better because their accessibility & readability, also, they take less time.

Step 3: Write 1 to 2 paragraphs on the challenges you faced when replacing your subqueries with CTEs.

For me, the challenge was to understand the concept as its still very new to me. I had difficulties replacing the inner and outer queries.