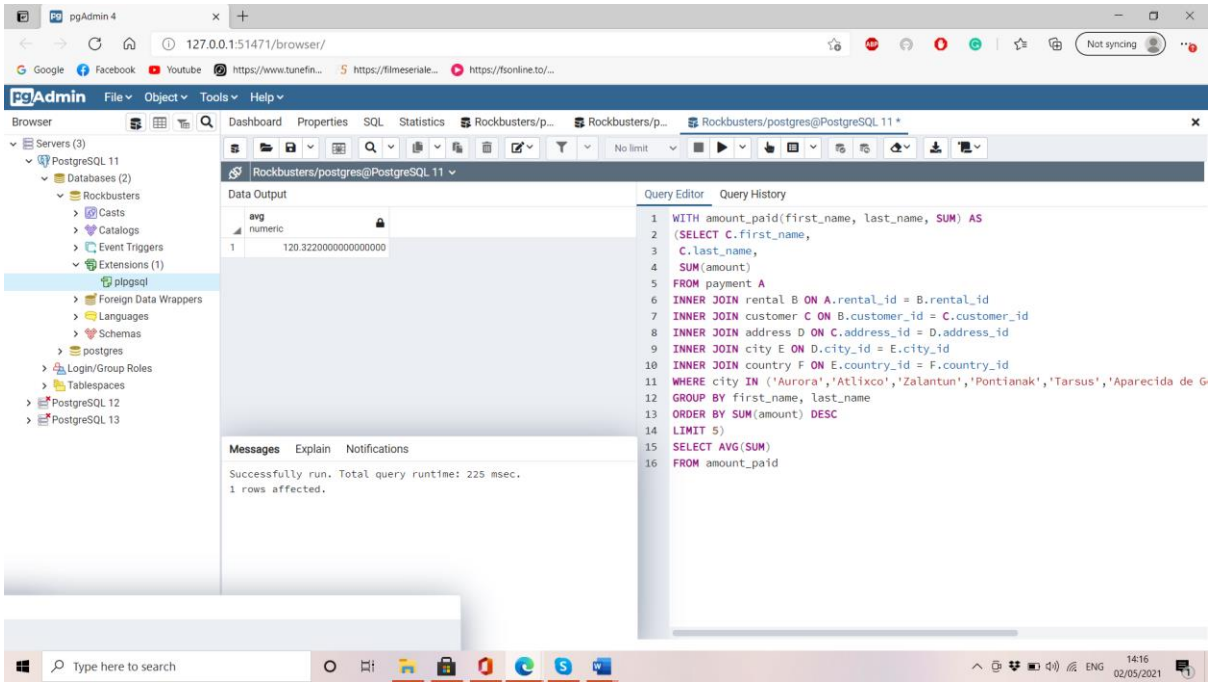
Step 1: Answer the business questions from step 1 and 2 of task 3.8 using CTEs

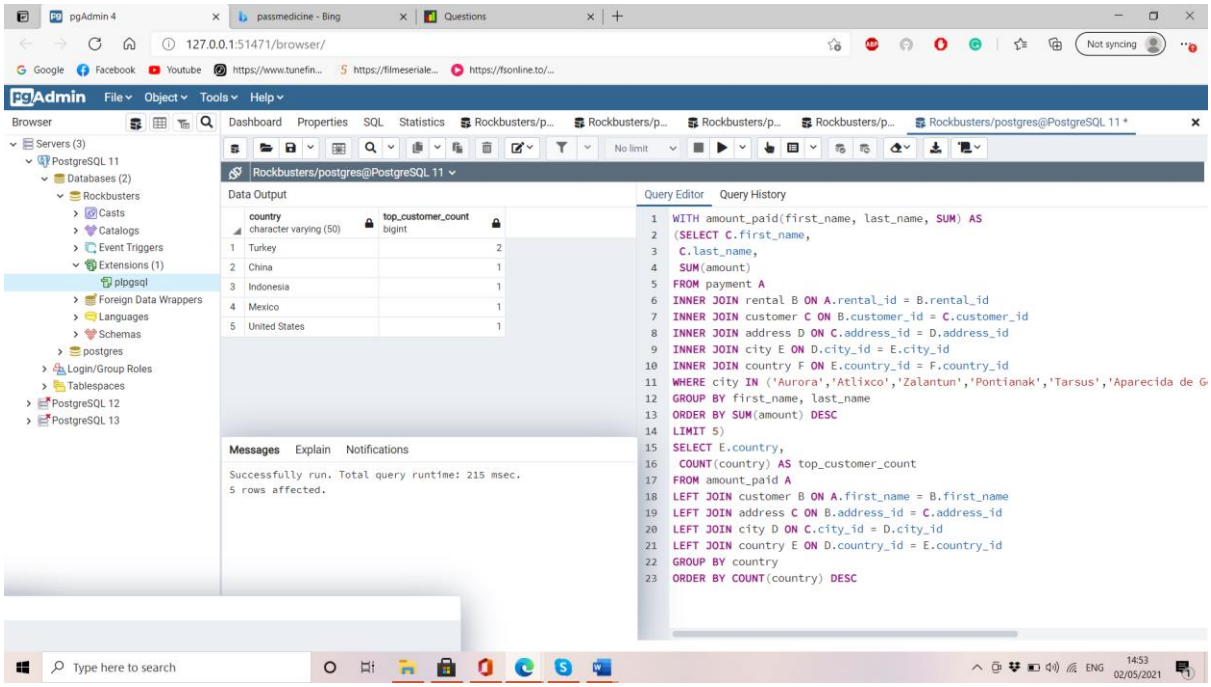
- Rewrite your queries from steps 1 and 2 of task 3.8 as CTEs.
- Copy-paste your CTEs and their outputs into your answers document.
- Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.

```
WITH amount_paid(first_name, last_name, SUM) AS

(SELECT C.first_name,

 C.last_name,

 SUM(amount)

FROM payment A

INNER JOIN rental B ON A.rental_id = B.rental_id

INNER JOIN customer C ON B.customer_id = C.customer_id

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

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

INNER JOIN country F ON E.country_id = F.country_id

WHERE city IN ('Aurora','Atlixco','Zalantun','Pontianak','Tarsus','Aparecida de Goinia','Emeishan','Rio Claro','Yingkou','Tokat')

GROUP BY first_name, last_name

ORDER BY SUM(amount) DESC

LIMIT 5)

SELECT AVG(SUM)

FROM amount_paid
```




```
WITH amount_paid(first_name, last_name, SUM) AS
(SELECT C.first_name,
 C.last_name,
 SUM(amount)
FROM payment A
INNER JOIN rental B ON A.rental_id = B.rental_id
INNER JOIN customer C ON B.customer_id = C.customer_id
INNER JOIN address D ON C.address_id = D.address_id
INNER JOIN city E ON D.city_id = E.city_id
INNER JOIN country F ON E.country_id = F.country_id
WHERE city IN ('Aurora','Atlixco','Zalantun','Pontianak','Tarsus','Aparecida de Goinia','Emeishan','Rio Claro','Yingkou','Tokat')
GROUP BY first_name, last_name
ORDER BY SUM(amount) DESC
LIMIT 5)
SELECT E.country,
 COUNT(country) AS top_customer_count
```

```
FROM amount_paid A
LEFT JOIN customer B ON A.first_name = B.first_name
LEFT JOIN address C ON B.address_id = C.address_id
LEFT JOIN city D ON C.city_id = D.city_id
LEFT JOIN country E ON D.country_id = E.country_id
GROUP BY country
ORDER BY COUNT(country) DESC
```



- I would compare the CTE with the English expression “Taking into account…” that it is often used in the beginning of sentences. Being an “a priori” statement it precedes your main statement.
- That is what I did. I started by writing the CTE and then I wrote the main statement observing that the references were correct.

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

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

➢ In my opinion, CTEs perform better than the subquery because the query looks a lot simple and way faster.

- Compare the costs of all the queries by creating query plans for each one.
- 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.
- Did the results surprise you? Write a few sentences to explain your answer.

Subquery

	QUERY PLAN text
1	Limit (cost=77.55..77.65 rows=5 width=38)
2	-> GroupAggregate (cost=77.55..77.75 rows=10 w...
3	Group Key: " Average"."First_Name"
4	-> Sort (cost=77.55..77.58 rows=10 width=38)
5	Sort Key: " Average"."First_Name"
6	-> Subquery Scan on " Average" (cost=77.2...
7	-> Limit (cost=77.26..77.29 rows=10 wi...
8	-> Sort (cost=77.26..77.93 rows=26...
9	Sort Key: payment.amount DESC
10	-> HashAggregate (cost=68.12..7...
11	Group Key: city.city, payment.c...

CTE cost

	QUERY PLAN text
1	Limit (cost=77.55..77.65 rows=5 width=38)
2	-> GroupAggregate (cost=77.55..77.75 rows=10 ...
3	Group Key: "Average"."First_Name"
4	-> Sort (cost=77.55..77.58 rows=10 width=38)
5	Sort Key: "Average"."First_Name"
6	-> Subquery Scan on "Average" (cost=77.2...
7	-> Limit (cost=77.26..77.29 rows=10 wi...
8	-> Sort (cost=77.26..77.93 rows=26...
9	Sort Key: payment.amount DESC
10	-> HashAggregate (cost=68.12.....
11	Group Key: city.city, payment.c...

- The estimated cost was the same, perhaps because it is calculated according to the output, and not the method. In practice the subquery was faster, therefore more efficient.
- I am not surprised because I did not have any expectations. But it would be relevant to know why there is such a difference and if it is possible to deduce that subqueries are always more efficient

Step 3:

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

- Considering what I wrote in step 1, the first challenge was to understand what a CTE is and how does it fit in the SQL language.
- During the implementation, it was a challenge to know what part to place inside the CTE parenthesis and to update the references that indicate which variables are we referring to.