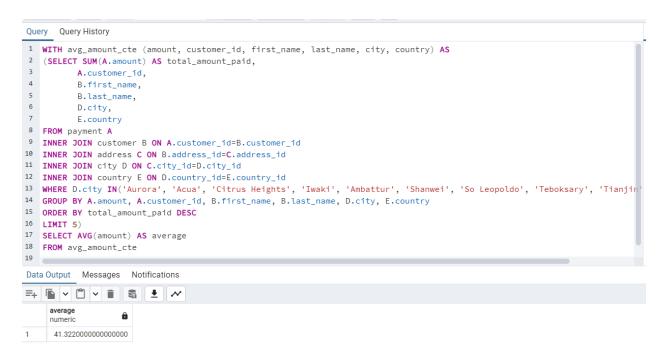
Answers 3.9



```
1 WITH top_customer_cte (amount, customer_id, first_name, last_name, city, country) AS
   (SELECT SUM(A.amount) AS total_amount_paid,
3
          B.customer_id,
4
          B.first_name,
5
          B.last name,
6
          D.city,
          E.country
  FROM payment A
9 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
3 WHERE D.city IN(SELECT D.city
4
                   FROM customer B
.5
                   INNER JOIN address C ON B.address_id=C.address_id
6
                   INNER JOIN city D ON C.city_id=D.city_id
.7
                   INNER JOIN country E ON D.country_id=E.country_id
8
                  WHERE E.country IN (SELECT E.country
9
                                       FROM customer B
10
                                       INNER JOIN address C ON B.address_id=C.address_id
1
                                       INNER JOIN city D ON C.city_id=D.city_id
2
                                       INNER JOIN country E ON D.country_id=E.country_id
13
                                      GROUP BY E.country
4
                                      ORDER BY COUNT(B.customer id) DESC
5
                                      LIMIT 10)
16
                  GROUP BY E.country,
.7
                           D.city
8
                  ORDER BY COUNT(B.customer_id) DESC
19
                  LIMIT 10)
10
  GROUP BY B.customer_id,
1
            B.first_name,
12
            B.last_name,
13
            D.city,
14
            E.country
ORDER BY SUM(A.amount) DESC
6 LIMIT 5)
7 SELECT E.country,
8
          COUNT(DISTINCT B.customer_id) AS all_customer_count,
19
          COUNT(DISTINCT top_customer_cte) AS top_customer_count
PROM customer B
JOIN address C ON B.address_id=C.address_id
JOIN city D ON C.city_id=D.city_id
3 JOIN country E ON D.country_id=E.country_id
4 LEFT JOIN top_customer_cte ON B.customer_id=top_customer_cte.customer_id
5 GROUP BY E.country
ORDER BY all_customer_count DESC
7 LIMIT 10
Data Output Messages Explain x Notifications
                        * ~
                      all_customer_count
                                       top_customer_count
    country
    character varying (50)
                       bigint
                                       bigint
     India
                                    60
                                                      1
     China
                                    53
                                                      1
     United States
                                    36
                                                      1
```

21

lanan

Explanation of Approach

 First I went back and copied the original subqueries. From there, I isolated the subquery and cut out the rest of the query. Next, I put in the WITH function and named the CTE in the top line. I kept referencing the previous subqueries to ensure I contained the necessary information.

Performance Comparison

- First one: the subquery took 77 milliseconds while the CTE took 54 milliseconds.
- Second one: the subquery took 85 milliseconds while the CTE took 95 milliseconds.
- These results were a bit surprising- at least, for the second query, as the subquery took less time than the CTE. However, we are dealing with much less data compared to databases at medium-to-large companies; in those cases, the differences would be magnified by orders of magnitude. In other words, I don't think the subquery versus CTE makes much of a difference in the Rockbuster database, but in the real world I think the advantages of the CTE would be much more apparent.

Challenges Faced When Replacing Subqueries with CTEs

The first subquery was easier for me to replace with a CTE. It was far less involved, meaning there were a lot fewer lines, so it wasn't as difficult to isolate the nested query and then rewrite the beginning and end with the correct CTE syntax.

However, the second query posed some issues. There were a lot of JOINs and more columns being pulled, so it took more time to read through line by line and ensure I kept the necessary information. I initially met some errors when trying to run the new CTE as I did not include JOINs that came outside the nested query, and I had to add those back in. Once I had the correct aliases as well, I was ready to run the query, and that time it came back correctly.