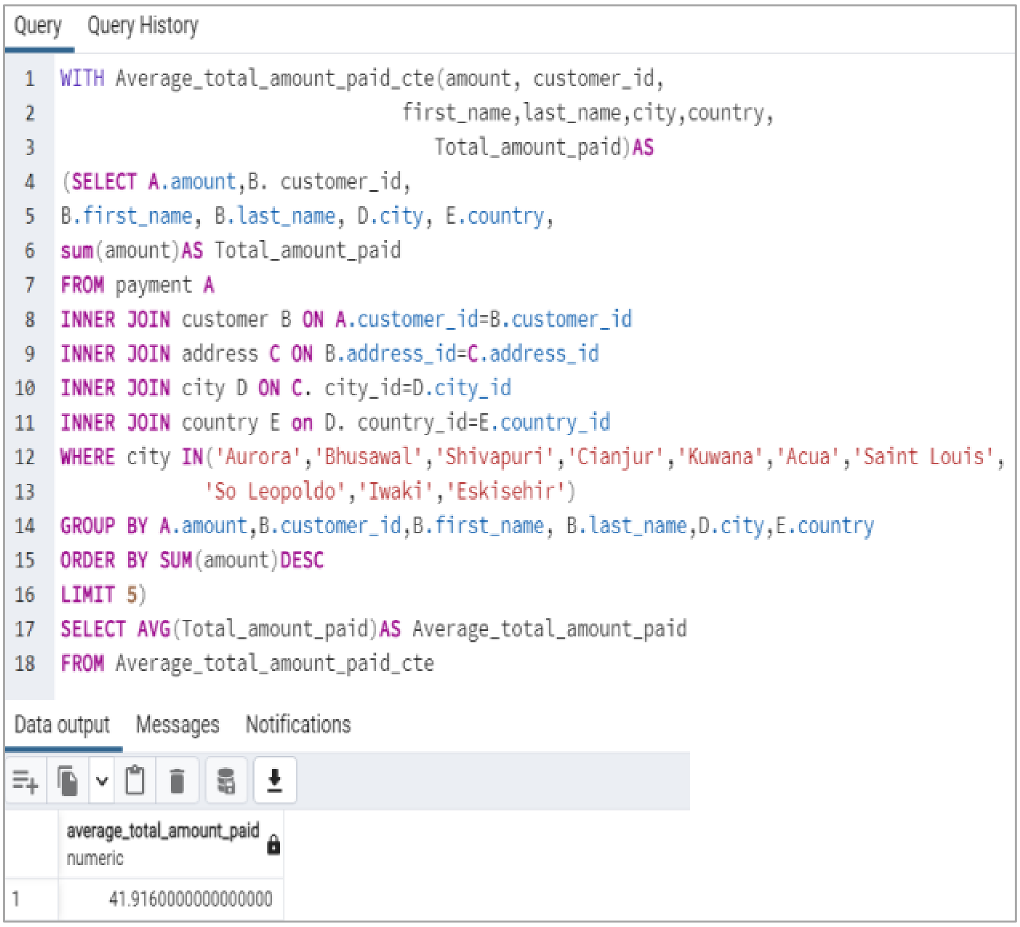
Exercise 3.9 - Common Table Expressions

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, what you did first, second, and so on.

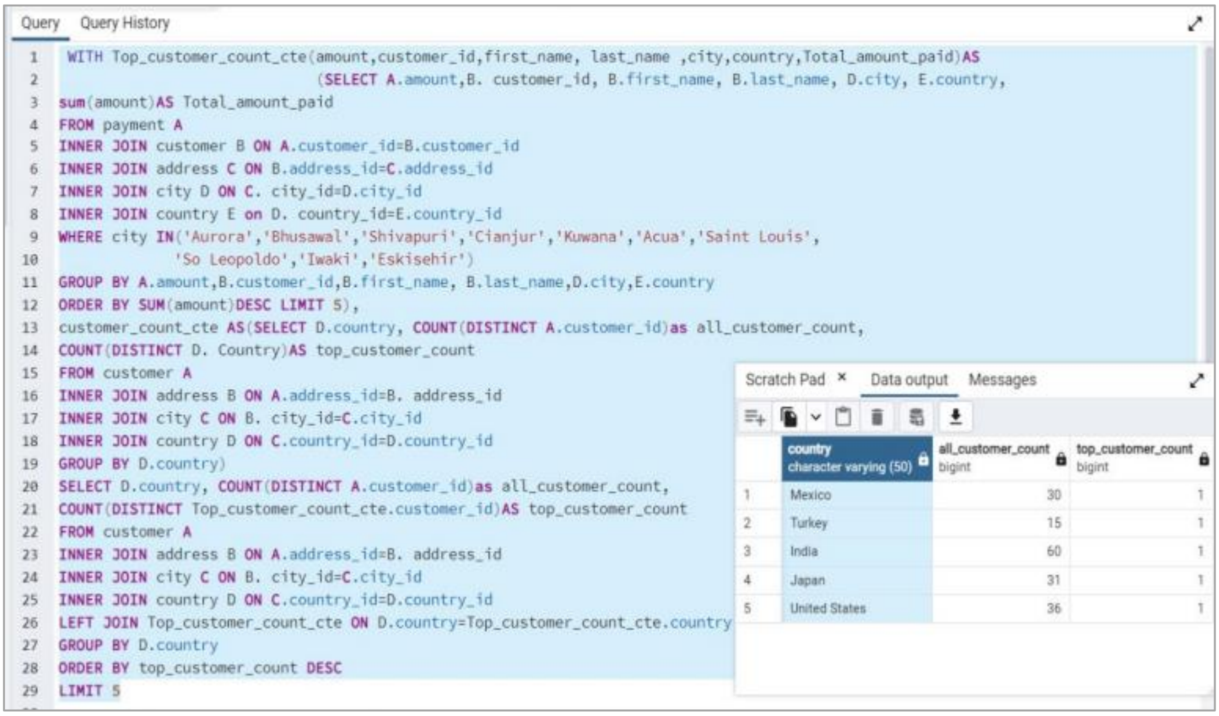
→First, find the average amount paid by the top 5 customers using CTE:



```sql
WITH Average_total_amount_paid_cte(amount, customer_id,
                                first_name,last_name,city,country,
                                   Total_amount_paid)AS
(SELECT A.amount,B. 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 city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','Acua','Saint Louis',
             'So Leopoldo','Iwaki','Eskisehir')
GROUP BY A.amount,B.customer_id,B.first_name, B.last_name,D.city,E.country
ORDER BY SUM(amount)DESC
LIMIT 5)
SELECT AVG(Total_amount_paid)AS Average_total_amount_paid
FROM Average_total_amount_paid_cte
```

	average_total_amount_paid numeric
1	41.9160000000000000

→Second, find out how many of the top 5 customers are based within each country (within the Top 10 cities from the top 10 countries:



```
 WITH Top_customer_count_cte(amount,customer_id,first_name, last_name ,city,country,Total_amount_paid)AS
                                    (SELECT A.amount,B. 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 city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','Acua','Saint Louis',
             'So Leopoldo','Iwaki','Eskisehir')
GROUP BY A.amount,B.customer_id,B.first_name, B.last_name,D.city,E.country
ORDER BY SUM(amount)DESC LIMIT 5),
customer_count_cte AS(SELECT D.country, COUNT(DISTINCT A.customer_id)as all_customer_count,
COUNT(DISTINCT D. Country)AS top_customer_count
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
GROUP BY D.country)
SELECT D.country, COUNT(DISTINCT A.customer_id)as all_customer_count,
COUNT(DISTINCT Top_customer_count_cte.customer_id)AS top_customer_count
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
LEFT JOIN Top_customer_count_cte ON D.country=Top_customer_count_cte.country
GROUP BY D.country
ORDER BY top_customer_count DESC
LIMIT 5
```

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	1
2	Turkey	15	1
3	India	60	1
4	Japan	31	1
5	United States	36	1

Process:

- ➢ Step 1 was to copy the subquery made in exercise.
- ➢ Then took out the outer query from the subquery and replaced it with CTE syntax.
- ➢ Left the inner query as it is for the step 1 task.
- ➢ Then created 2 CTEs for step 2 task names for the two inner queries (one to get the total amount paid from top 5 customers in top 10 cities within the top 10 countries, and the other query focus on the customer counts).
- ➢ I finally wrote the main statement to query the information required from the CTE table created.

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

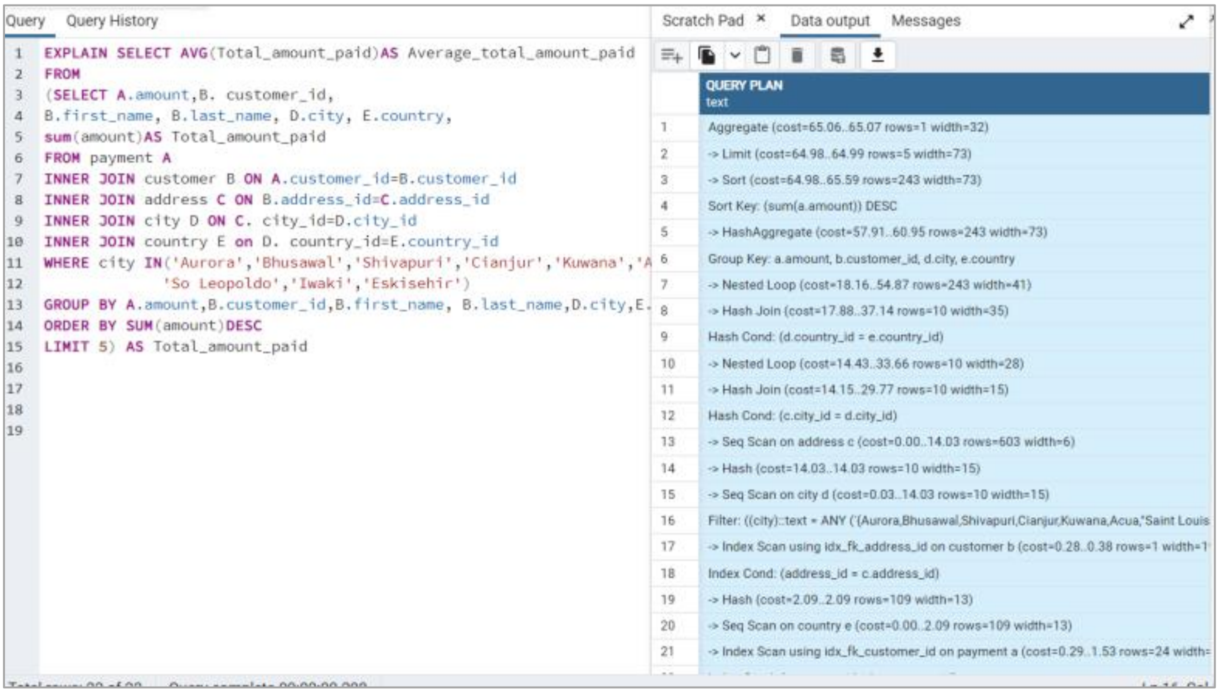
1. Which approach do you think will perform better and why?
2. Compare the costs of all the queries by creating query plans for each one.
3. The EXPLAIN command gives you an *estimated* cost. To find out the actual speed of your queries, run them in pgAdmin 4. After each query has been run, a pop-up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

→At the beginning, I didn’t have any idea which one will perform better. The subquery was shorter than the CTE query I wrote but knowing the CTE was easier to read and learned I used EXPLAIN to know which is faster and cost less.

For Task 1(Find the average amount paid by the top 5 customers within the Top 10 cities from the top 10 countries))

	SUBQUERY	**CTE**
COST	cost=65.06..65.07 rows=1 width=32	cost=65.06. 65.07 rows=1 width=32
TIME	Total query runtime: 298 msec. 22 rows affected.	Total query runtime: 267 msec. 22 rows affected.

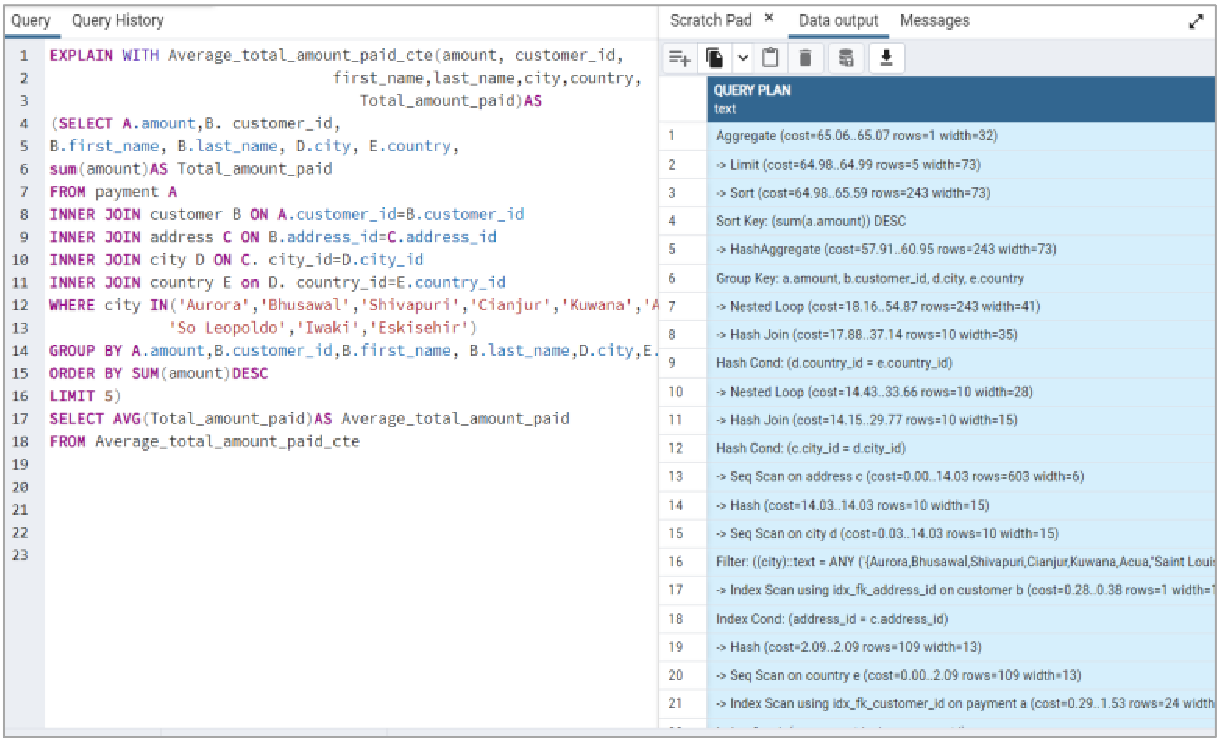
Subquery



```sql
EXPLAIN SELECT AVG(Total_amount_paid)AS Average_total_amount_paid
FROM
(SELECT A.amount,B. 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 city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','A
              'So Leopoldo','Iwaki','Eskisehir')
GROUP BY A.amount,B.customer_id,B.first_name, B.last_name,D.city,E.
ORDER BY SUM(amount)DESC
LIMIT 5) AS Total_amount_paid
```

```
QUERY PLAN
text
1  Aggregate (cost=65.06..65.07 rows=1 width=32)
2  -> Limit (cost=64.98..64.99 rows=5 width=73)
3  -> Sort (cost=64.98..65.59 rows=243 width=73)
4  Sort Key: (sum(a.amount)) DESC
5  -> HashAggregate (cost=57.91..60.95 rows=243 width=73)
6  Group Key: a.amount, b.customer_id, d.city, e.country
7  -> Nested Loop (cost=18.16..54.87 rows=243 width=41)
8  -> Hash Join (cost=17.88..37.14 rows=10 width=35)
9  Hash Cond: (d.country_id = e.country_id)
10 -> Nested Loop (cost=14.43..33.66 rows=10 width=28)
11 -> Hash Join (cost=14.15..29.77 rows=10 width=15)
12 Hash Cond: (c.city_id = d.city_id)
13 -> Seq Scan on address c (cost=0.00..14.03 rows=603 width=6)
14 -> Hash (cost=14.03..14.03 rows=10 width=15)
15 -> Seq Scan on city d (cost=0.03..14.03 rows=10 width=15)
16 Filter: ((city)::text = ANY ('{Aurora,Bhusawal,Shivapuri,Cianjur,Kuwana,Acua,"Saint Louis
17 -> Index Scan using idx_fk_address_id on customer b (cost=0.28..0.38 rows=1 width=1
18 Index Cond: (address_id = c.address_id)
19 -> Hash (cost=2.09..2.09 rows=109 width=13)
20 -> Seq Scan on country e (cost=0.00..2.09 rows=109 width=13)
21 -> Index Scan using idx_fk_customer_id on payment a (cost=0.29..1.53 rows=24 width=
```

CTE



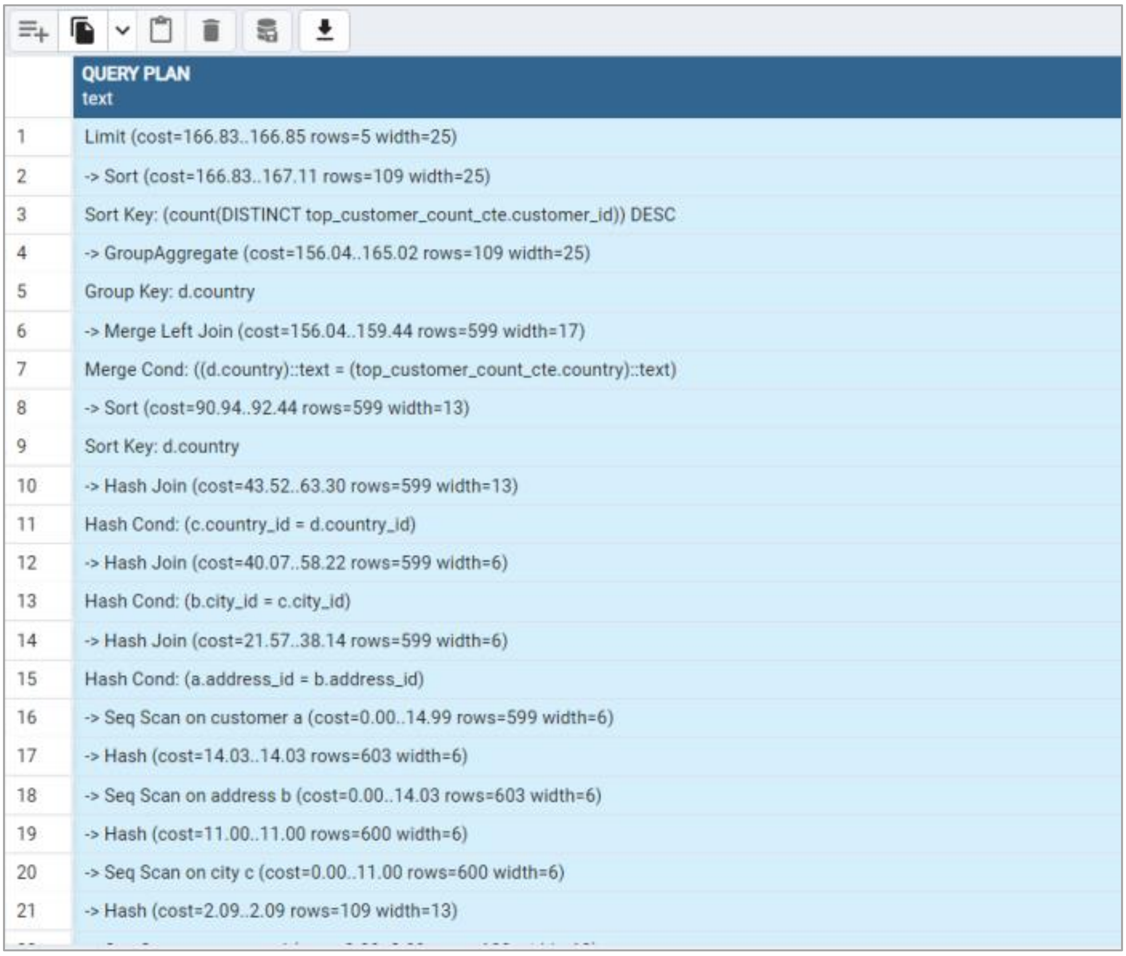
```sql
EXPLAIN WITH Average_total_amount_paid_cte(amount, customer_id,
                                first_name,last_name,city,country,
                                  Total_amount_paid)AS
(SELECT A.amount,B. customer_id,
B.first_name, B.last_name, D.city, E.country,
sum(amount)AS Total_amount_paid
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 city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','A
              'So Leopoldo','Iwaki','Eskisehir')
GROUP BY A.amount,B.customer_id,B.first_name, B.last_name,D.city,E.
ORDER BY SUM(amount)DESC
LIMIT 5)
SELECT AVG(Total_amount_paid)AS Average_total_amount_paid
FROM Average_total_amount_paid_cte
```

```
QUERY PLAN
text
1  Aggregate (cost=65.06..65.07 rows=1 width=32)
2  -> Limit (cost=64.98..64.99 rows=5 width=73)
3  -> Sort (cost=64.98..65.59 rows=243 width=73)
4  Sort Key: (sum(a.amount)) DESC
5  -> HashAggregate (cost=57.91..60.95 rows=243 width=73)
6  Group Key: a.amount, b.customer_id, d.city, e.country
7  -> Nested Loop (cost=18.16..54.87 rows=243 width=41)
8  -> Hash Join (cost=17.88..37.14 rows=10 width=35)
9  Hash Cond: (d.country_id = e.country_id)
10 -> Nested Loop (cost=14.43..33.66 rows=10 width=28)
11 -> Hash Join (cost=14.15..29.77 rows=10 width=15)
12 Hash Cond: (c.city_id = d.city_id)
13 -> Seq Scan on address c (cost=0.00..14.03 rows=603 width=6)
14 -> Hash (cost=14.03..14.03 rows=10 width=15)
15 -> Seq Scan on city d (cost=0.03..14.03 rows=10 width=15)
16 Filter: ((city)::text = ANY ('{Aurora,Bhusawal,Shivapuri,Cianjur,Kuwana,Acua,"Saint Loui
17 -> Index Scan using idx_fk_address_id on customer b (cost=0.28..0.38 rows=1 width=1
18 Index Cond: (address_id = c.address_id)
19 -> Hash (cost=2.09..2.09 rows=109 width=13)
20 -> Seq Scan on country e (cost=0.00..2.09 rows=109 width=13)
21 -> Index Scan using idx_fk_customer_id on payment a (cost=0.29..1.53 rows=24 width
```

For task 2(Task 2: Find out how many of the top 5 customers are based within each country(Within the Top 10 cities from the top 10 countries))

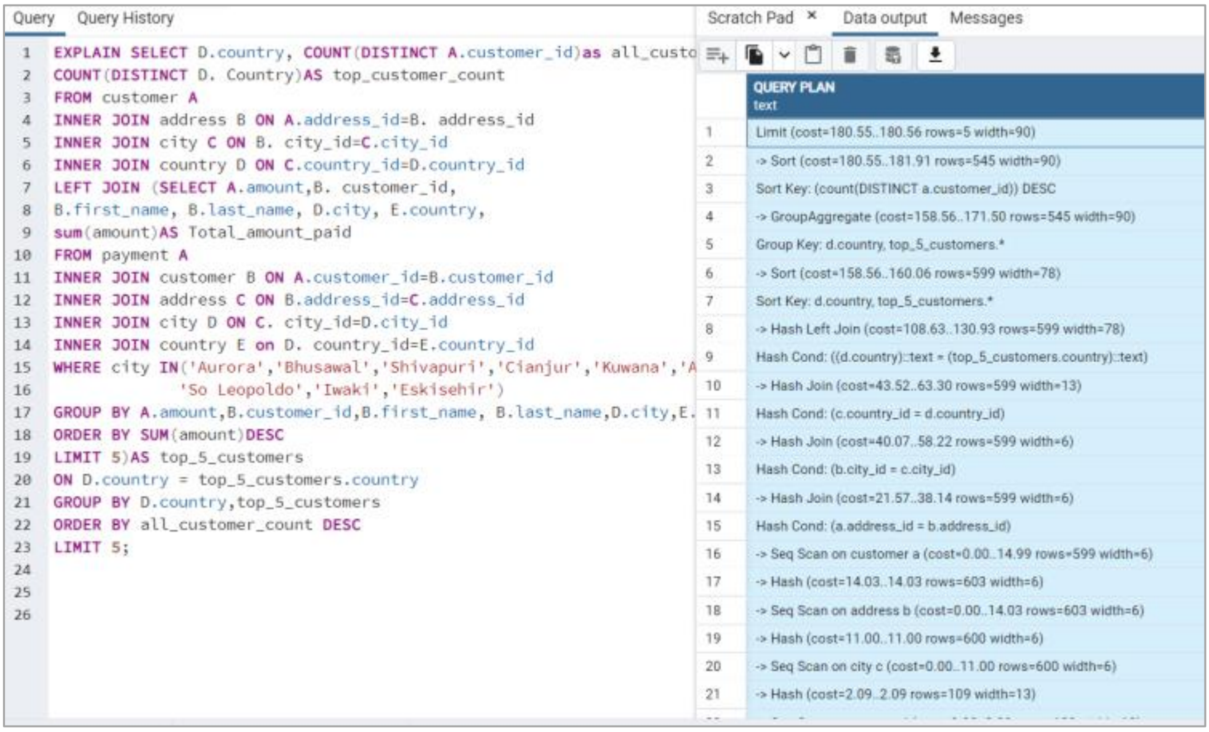
	SUBQUERY	**CTE**
COST	cost=180.55..180.56 rows=5 width=90	cost=166.83. 166.85 rows=5 width=25
TIME	Total query runtime: 262 msec. 45 rows affected.	Total query runtime: 157 msec. 46 rows affected.

CTE



	QUERY PLAN text
1	Limit (cost=166.83..166.85 rows=5 width=25)
2	-> Sort (cost=166.83..167.11 rows=109 width=25)
3	Sort Key: (count(DISTINCT top_customer_count_cte.customer_id)) DESC
4	-> GroupAggregate (cost=156.04..165.02 rows=109 width=25)
5	Group Key: d.country
6	-> Merge Left Join (cost=156.04..159.44 rows=599 width=17)
7	Merge Cond: ((d.country)::text = (top_customer_count_cte.country)::text)
8	-> Sort (cost=90.94..92.44 rows=599 width=13)
9	Sort Key: d.country
10	-> Hash Join (cost=43.52..63.30 rows=599 width=13)
11	Hash Cond: (c.country_id = d.country_id)
12	-> Hash Join (cost=40.07..58.22 rows=599 width=6)
13	Hash Cond: (b.city_id = c.city_id)
14	-> Hash Join (cost=21.57..38.14 rows=599 width=6)
15	Hash Cond: (a.address_id = b.address_id)
16	-> Seq Scan on customer a (cost=0.00..14.99 rows=599 width=6)
17	-> Hash (cost=14.03..14.03 rows=603 width=6)
18	-> Seq Scan on address b (cost=0.00..14.03 rows=603 width=6)
19	-> Hash (cost=11.00..11.00 rows=600 width=6)
20	-> Seq Scan on city c (cost=0.00..11.00 rows=600 width=6)
21	-> Hash (cost=2.09..2.09 rows=109 width=13)

Subquery



```sql
EXPLAIN SELECT D.country, COUNT(DISTINCT A.customer_id)as all_custo
COUNT(DISTINCT D. Country)AS top_customer_count
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
LEFT JOIN (SELECT A.amount,B. 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 city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','A
               'So Leopoldo','Iwaki','Eskisehir')
GROUP BY A.amount,B.customer_id,B.first_name, B.last_name,D.city,E.
ORDER BY SUM(amount)DESC
LIMIT 5)AS top_5_customers
ON D.country = top_5_customers.country
GROUP BY D.country,top_5_customers
ORDER BY all_customer_count DESC
LIMIT 5;
```

	QUERY PLAN text
1	Limit (cost=180.55..180.56 rows=5 width=90)
2	-> Sort (cost=180.55..181.91 rows=545 width=90)
3	Sort Key: (count(DISTINCT a.customer_id)) DESC
4	-> GroupAggregate (cost=158.56..171.50 rows=545 width=90)
5	Group Key: d.country, top_5_customers.*
6	-> Sort (cost=158.56..160.06 rows=599 width=78)
7	Sort Key: d.country, top_5_customers.*
8	-> Hash Left Join (cost=108.63..130.93 rows=599 width=78)
9	Hash Cond: ((d.country)::text = (top_5_customers.country)::text)
10	-> Hash Join (cost=43.52..63.30 rows=599 width=13)
11	Hash Cond: (c.country_id = d.country_id)
12	-> Hash Join (cost=40.07..58.22 rows=599 width=6)
13	Hash Cond: (b.city_id = c.city_id)
14	-> Hash Join (cost=21.57..38.14 rows=599 width=6)
15	Hash Cond: (a.address_id = b.address_id)
16	-> Seq Scan on customer a (cost=0.00..14.99 rows=599 width=6)
17	-> Hash (cost=14.03..14.03 rows=603 width=6)
18	-> Seq Scan on address b (cost=0.00..14.03 rows=603 width=6)
19	-> Hash (cost=11.00..11.00 rows=600 width=6)
20	-> Seq Scan on city c (cost=0.00..11.00 rows=600 width=6)
21	-> Hash (cost=2.09..2.09 rows=109 width=13)

For task 1, the cost is the same, but CTE run time is faster than the subquery statement For Task 2, subquery cost and time are higher than CTE. I think this has to do with my system because anytime I rerun the queries, I get a different run time for the same query.

Step 3:

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

→For the task number 1, it was straightforward with the help of the example given in the study note. But on the other hand, task 2 much more difficult as I forgot to rename the second inner query with another CTE name only that I won't start with the "WITH" statement. Moreover, combining the two CTEs was difficult which makes time-consuming in order to get the output I wanted. I need to improve my SQL logic in and train in order to get to do these more efficiently.