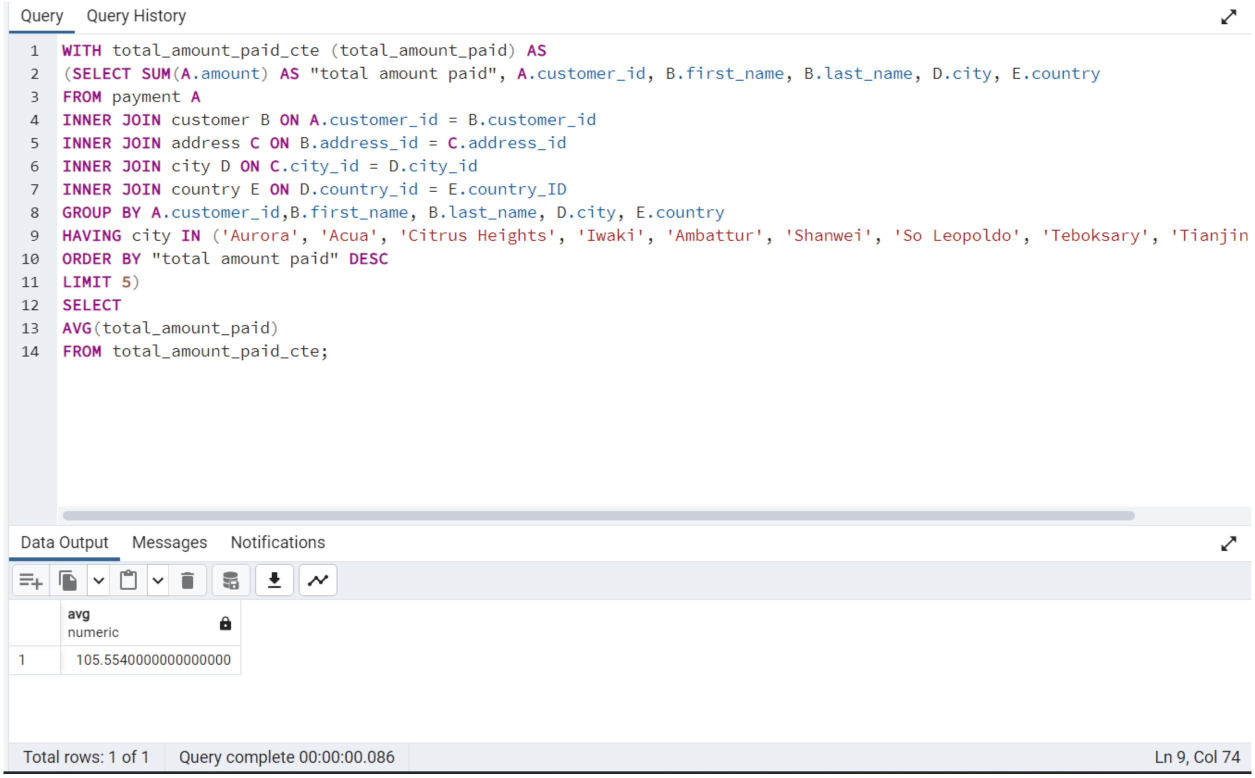
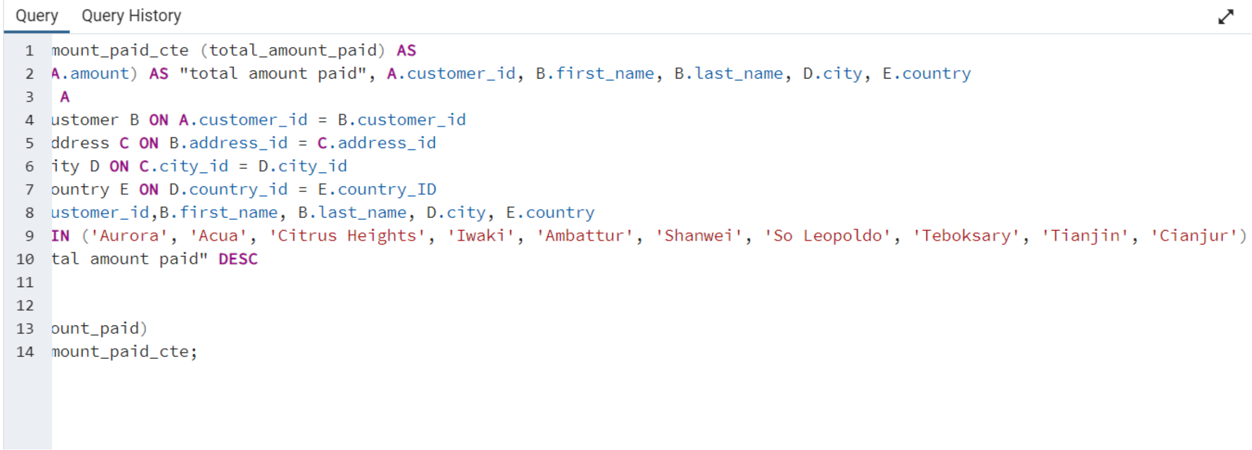
1



```
WITH total_amount_paid_cte (total_amount_paid) AS
(SELECT SUM(A.amount) AS "total amount paid", A.customer_id, B.first_name, B.last_name, D.city, E.country
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
GROUP BY A.customer_id,B.first_name, B.last_name, D.city, E.country
HAVING city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary', 'Tianjin
ORDER BY "total amount paid" DESC
LIMIT 5)
SELECT
AVG(total_amount_paid)
FROM total_amount_paid_cte;
```

	avg numeric
1	105.5540000000000000

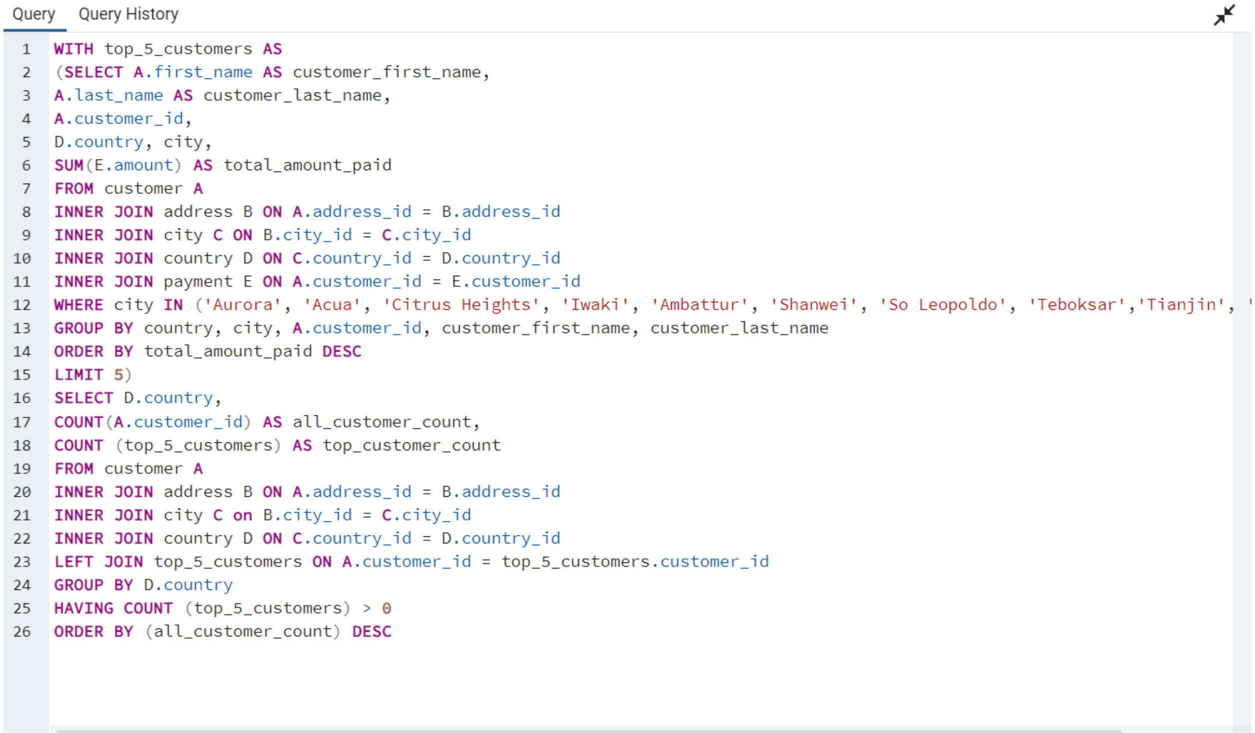
Total rows: 1 of 1 Query complete 00:00:00.086 Ln 9, Col 74



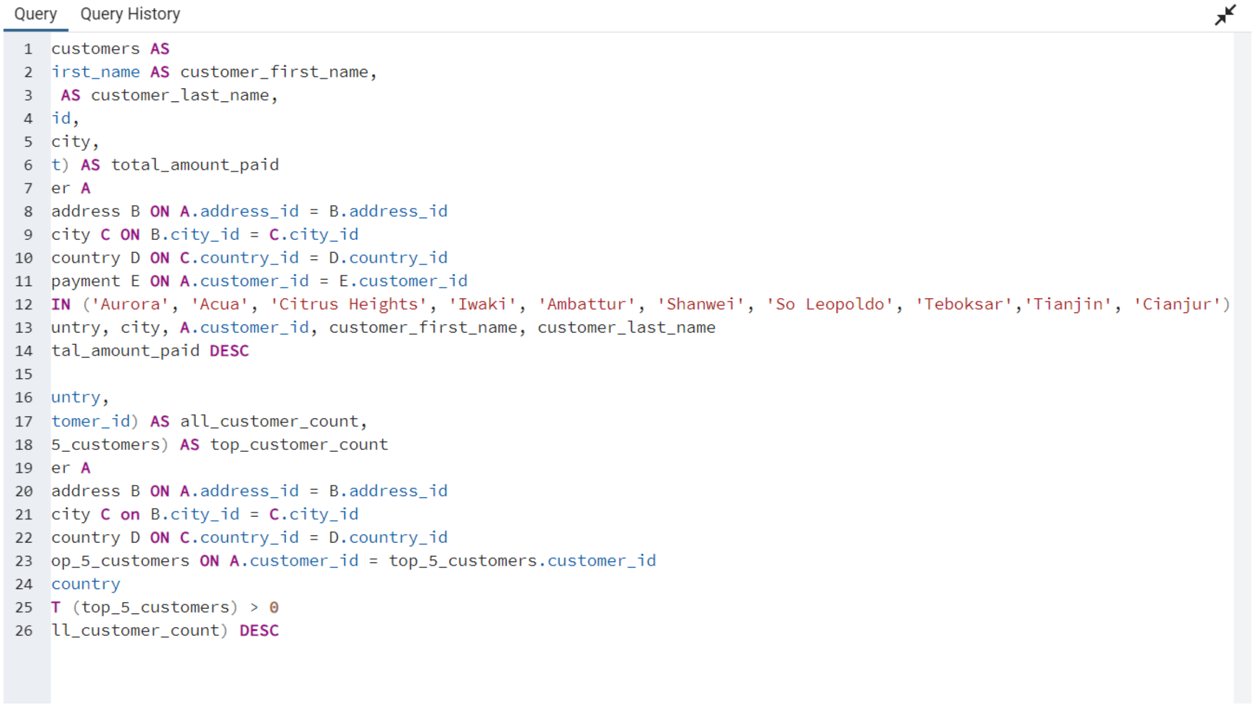
```
mount_paid_cte (total_amount_paid) AS
A.amount) AS "total amount paid", A.customer_id, B.first_name, B.last_name, D.city, E.country
 A
ustomer B ON A.customer_id = B.customer_id
ddress C ON B.address_id = C.address_id
ity D ON C.city_id = D.city_id
ountry E ON D.country_id = E.country_ID
ustomer_id,B.first_name, B.last_name, D.city, E.country
IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary', 'Tianjin', 'Cianjur')
tal amount paid" DESC


ount_paid)
mount_paid_cte;
```

1.b

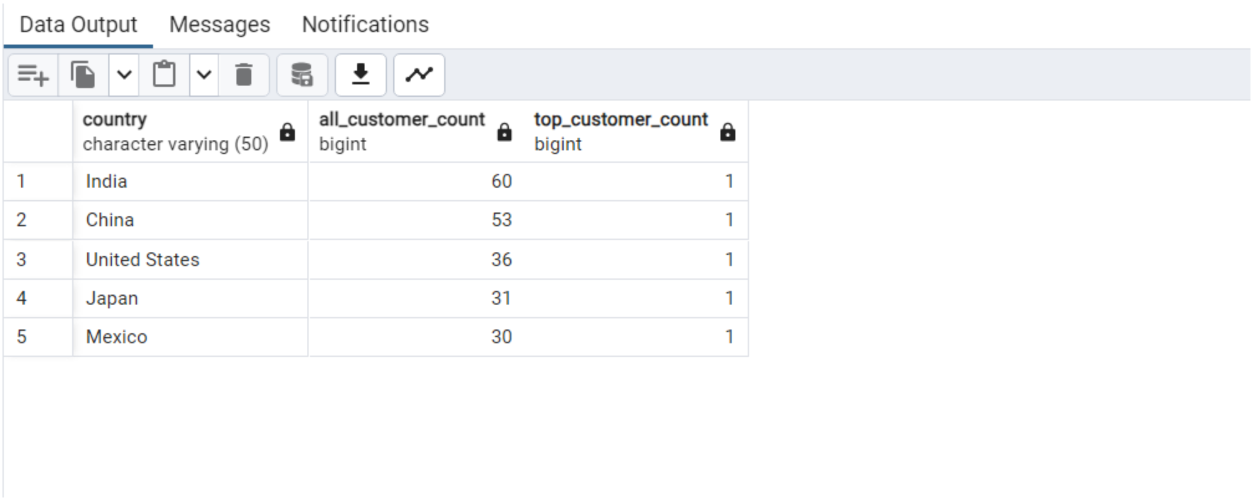


```
WITH top_5_customers AS
(SELECT A.first_name AS customer_first_name,
A.last_name AS customer_last_name,
A.customer_id,
D.country, city,
SUM(E.amount) AS total_amount_paid
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D ON C.country_id = D.country_id
INNER JOIN payment E ON A.customer_id = E.customer_id
WHERE city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksar','Tianjin', '
GROUP BY country, city, A.customer_id, customer_first_name, customer_last_name
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT D.country,
COUNT(A.customer_id) AS all_customer_count,
COUNT (top_5_customers) 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_5_customers ON A.customer_id = top_5_customers.customer_id
GROUP BY D.country
HAVING COUNT (top_5_customers) > 0
ORDER BY (all_customer_count) DESC
```



```
customers AS
irst_name AS customer_first_name,
 AS customer_last_name,
id,
city,
t) AS total_amount_paid
er A
address B ON A.address_id = B.address_id
city C ON B.city_id = C.city_id
country D ON C.country_id = D.country_id
payment E ON A.customer_id = E.customer_id
IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksar','Tianjin', 'Cianjur')
untry, city, A.customer_id, customer_first_name, customer_last_name
tal_amount_paid DESC

untry,
tomer_id) AS all_customer_count,
5_customers) AS top_customer_count
er A
address B ON A.address_id = B.address_id
city C on B.city_id = C.city_id
country D ON C.country_id = D.country_id
op_5_customers ON A.customer_id = top_5_customers.customer_id
country
T (top_5_customers) > 0
ll_customer_count) DESC
```



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

1.c I used the Common Table Expression (CTE) to generate a temporary table denominated "top_5_customers," which encompasses the information of the highest 5 spending customers. By utilizing CTE, I enhanced the organization and comprehensibility of the code. Subsequently, I alluded to the CTE table in the primary query to ascertain the number of customers residing in each country and the count of customers from the top 5 list residing in each country.

2.a I feel like CTE is quicker and more organized. However when deciding which process to use the overall cost of the project needs to be considered.

2.b

	CTE	Subquery
Average Paid Top 5 Customers	"Aggregate (cost=70.10..70.11 rows=1 width=32)" Query time:00:00:00.080	"Aggregate (cost=70.10..70.11 rows=1 width=32)" 00:00:00.00.072
Top 5 Customers in each Country	"Sort (cost=140.85..140.90 rows=36 width=25)" 00:00:00.086	"Sort (cost=140.85..142.90 rows=36 width=25)" 00:00:00.110

2.c The results of the EXPLAIN Query was surprising because it appears the subquery was faster when searching for the average, but the CTE was faster with the top 5 customers in each country.

3.a I find it hard to perform CTEs. Renaming the queries and looking at them individually was the hardest part. But, like subqueries I know with more practice the concept and ability to perform both will become easier.