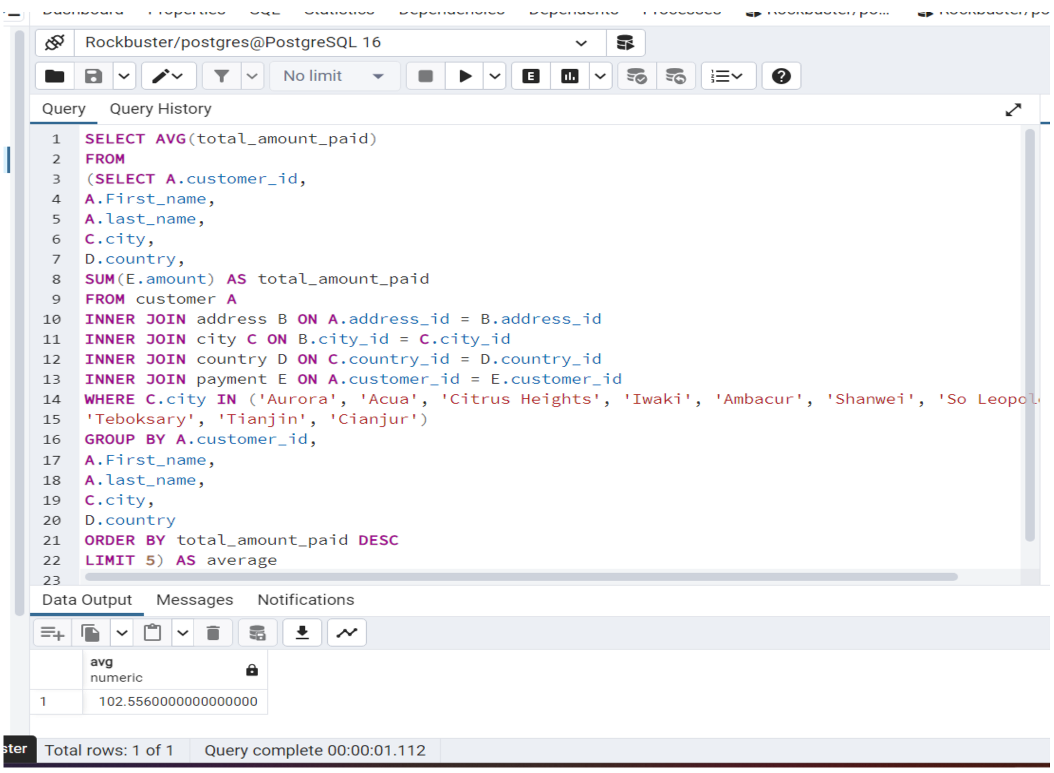
Olusola Olaniyi

Task 3.8 Performing Subqueries

Step 1.


Rockbuster/postgres@PostgreSQL 16

```
SELECT AVG(total_amount_paid)
FROM
(SELECT A.customer_id,
A.First_name,
A.last_name,
C.city,
D.country,
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 C.city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambacur', 'Shanwei', 'So Leopol
'Teboksary', 'Tianjin', 'Cianjur')
GROUP BY A.customer_id,
A.First_name,
A.last_name,
C.city,
D.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS average
```

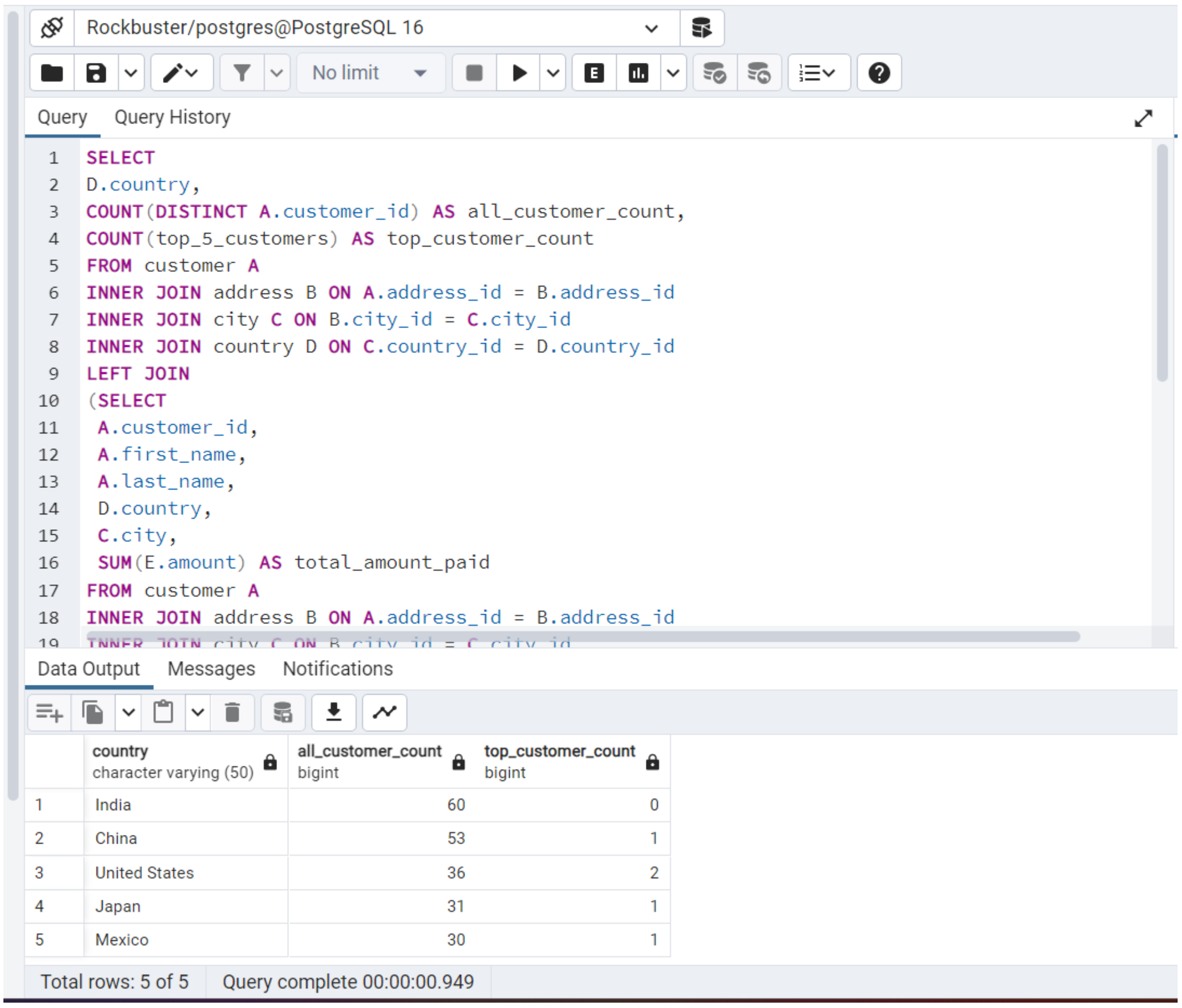
	avg numeric
1	102.5560000000000000

Total rows: 1 of 1   Query complete 00:00:01.112

Step 2.

SELECT
D.country,
COUNT(DISTINCT 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
(SELECT
 A.customer_id,
 A.first_name,
 A.last_name,
 D.country,
 C.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 C.city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambacur', 'Shanwei', 'So Leopoldo',
'Teboksary', 'Tianjin', 'Cianjur')
GROUP BY A.customer_id,
A.First_name,
A.last_name,
C.city,
D.country
ORDER BY total_amount_paid DESC
LIMIT 5) top_5_customers ON A.customer_id = top_5_customers.customer_id
GROUP BY D.country
ORDER BY all_customer_count DESC
LIMIT 5;



```
SELECT
D.country,
COUNT(DISTINCT 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
(SELECT
 A.customer_id,
 A.first_name,
 A.last_name,
 D.country,
 C.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
```

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	0
2	China	53	1
3	United States	36	2
4	Japan	31	1
5	Mexico	30	1

Total rows: 5 of 5 Query complete 00:00:00.949

Step 3.

Well, I think step 1 and 2 should have been done without using subqueries because we've been able to identify top 10 cities in the last task 3.7. Though subqueries are very useful in complex queries with multiple joins. I think subqueries are very useful where a query depends on the results of another query. Using only JOINS could have been less complex.