<u>Answers 3.8</u> **<u>Yaw Assensoh Opoku</u>**

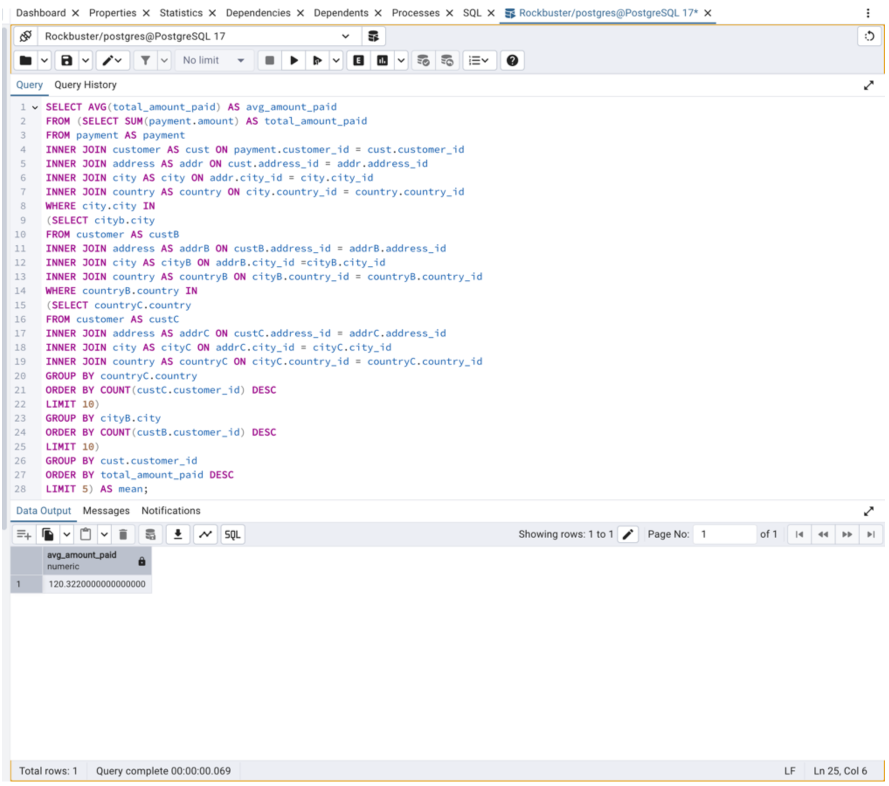
1. SELECT AVG(total_amount_paid) AS avg_amount_paid

```
FROM (SELECT SUM(payment.amount) AS total_amount_paid
FROM payment AS payment
INNER JOIN customer AS cust ON payment.customer_id = cust.customer_id
INNER JOIN address AS addr ON cust.address_id = addr.address_id
INNER JOIN city AS city ON addr.city_id = city.city_id
INNER JOIN country AS country ON city.country_id = country.country_id
WHERE city.city IN
(SELECT cityb.city
FROM customer AS custB
INNER JOIN address AS addrB ON custB.address_id = addrB.address_id
INNER JOIN city AS cityB ON addrB.city_id =cityB.city_id
INNER JOIN country AS countryB ON cityB.country_id = countryB.country_id
WHERE countryB.country IN
(SELECT countryC.country
FROM customer AS custC
INNER JOIN address AS addrC ON custC.address_id = addrC.address_id
INNER JOIN city AS cityC ON addrC.city_id = cityC.city_id
INNER JOIN country AS countryC ON cityC.country_id = countryC.country_id
GROUP BY countryC.country
ORDER BY COUNT(custC.customer_id) DESC
LIMIT 10)
GROUP BY cityB.city
ORDER BY COUNT(custB.customer_id) DESC
LIMIT 10)
```

GROUP BY cust.customer_id

ORDER BY total_amount_paid DESC

LIMIT 5) AS mean;



```
SELECT AVG(total_amount_paid) AS avg_amount_paid
FROM (SELECT SUM(payment.amount) AS total_amount_paid
FROM payment AS payment
INNER JOIN customer AS cust ON payment.customer_id = cust.customer_id
INNER JOIN address AS addr ON cust.address_id = addr.address_id
INNER JOIN city AS city ON addr.city_id = city.city_id
INNER JOIN country AS country ON city.country_id = country.country_id
WHERE city.city IN
(SELECT cityb.city
FROM customer AS custB
INNER JOIN address AS addrB ON custB.address_id = addrB.address_id
INNER JOIN city AS cityB ON addrB.city_id =cityB.city_id
INNER JOIN country AS countryB ON cityB.country_id = countryB.country_id
WHERE countryB.country IN
(SELECT countryC.country
FROM customer AS custC
INNER JOIN address AS addrC ON custC.address_id = addrC.address_id
INNER JOIN city AS cityC ON addrC.city_id = cityC.city_id
INNER JOIN country AS countryC ON cityC.country_id = countryC.country_id
GROUP BY countryC.country
ORDER BY COUNT(custC.customer_id) DESC
LIMIT 10)
GROUP BY cityB.city
ORDER BY COUNT(custB.customer_id) DESC
LIMIT 10)
GROUP BY cust.customer_id
ORDER BY total_amount_paid DESC
LIMIT 5) AS mean;
```

```
  avg_amount_paid
  numeric
1 120.3220000000000000
```

```
Total rows: 1    Query complete 00:00:00.069    LF    Ln 25, Col 6
```

2. LECT country.country,

COUNT(DISTINCT customer.customer_id) AS all_customer_count,

COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count

FROM customer

INNER JOIN address ON customer.address_id = address.address_id

INNER JOIN city ON address.city_id = city.city_id

INNER JOIN country ON city.country_id = country.country_id

LEFT JOIN

(SELECT customer.customer_id, country.country

FROM payment

INNER JOIN customer ON payment.customer_id = customer.customer_id

INNER JOIN address ON customer.address_id = address.address_id

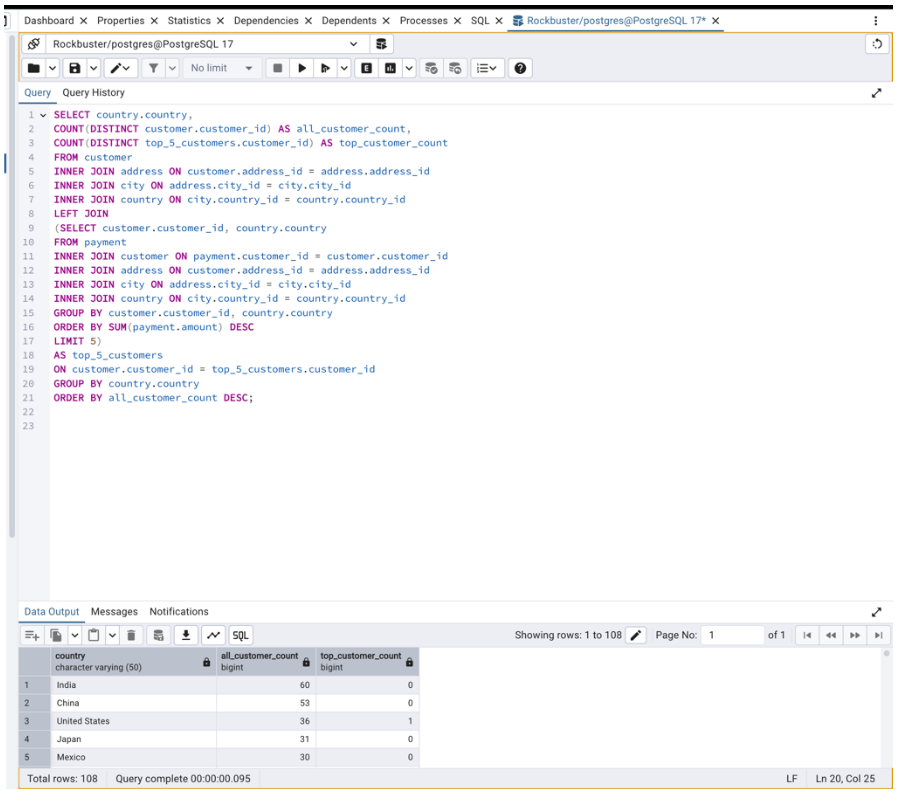
INNER JOIN city ON address.city_id = city.city_id

INNER JOIN country ON city.country_id = country.country_id

GROUP BY customer.customer_id, country.country

ORDER BY SUM(payment.amount) DESC

LIMIT 5)



```sql
SELECT country.country,
COUNT(DISTINCT customer.customer_id) AS all_customer_count,
COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
FROM customer
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city ON address.city_id = city.city_id
INNER JOIN country ON city.country_id = country.country_id
LEFT JOIN
(SELECT customer.customer_id, country.country
FROM payment
INNER JOIN customer ON payment.customer_id = customer.customer_id
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city ON address.city_id = city.city_id
INNER JOIN country ON city.country_id = country.country_id
GROUP BY customer.customer_id, country.country
ORDER BY SUM(payment.amount) DESC
LIMIT 5)
AS top_5_customers
ON customer.customer_id = top_5_customers.customer_id
GROUP BY country.country
ORDER BY all_customer_count DESC;
```

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

Total rows: 108 Query complete 00:00:00.095 LF Ln 20, Col 25

3.

Steps 1 and 2 technically be done without using subqueries, but it would make the query more complex and harder to manage. For example, instead of isolating the top 5 customers in a subquery, you'd have to join, sort, and filter the data all within a single main query. This can lead to long, nested JOINs and WHERE clauses that are more difficult to read. Subqueries help break down the logic into smaller, more understandable pieces, especially when you're working with aggregates data like totals and rankings.

Subqueries are particularly useful when you need to compute intermediate results like finding the top-paying customers, identifying the most common countries, or filtering data based on grouped values. They’re great for isolating and managing queries. Using subqueries can also improve performance in some cases by narrowing down the data earlier in the query process depending on how the database optimized.