Performing Subqueries

Step 1

s Help

Dashboard Properties SQL Statistics Dependencies Dependents Processes Task 3.8 Performing Subqueries.sql*

Rockbuster/postgres@PostgreSQL 15

No limit

Query Query History

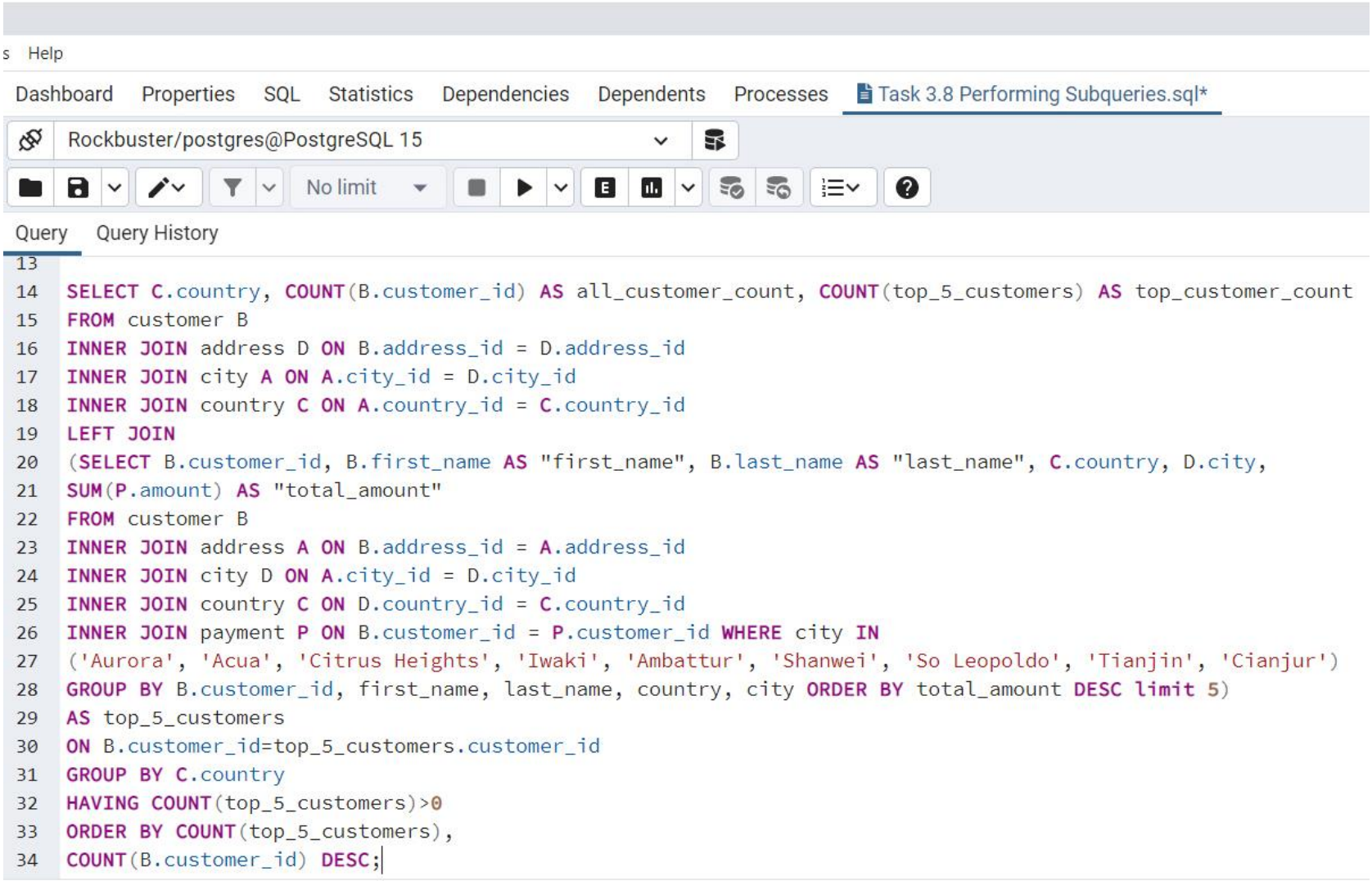
```
1  SELECT avg(total_amount) AS "average"
2  FROM
3  (SELECT B.customer_id, B.first_name AS "first_name", B.last_name AS "last_name", C.country, D.city,
4  SUM(P.amount) AS "total_amount"
5  FROM customer B
6  INNER JOIN address A ON B.address_id = A.address_id
7  INNER JOIN city D ON A.city_id = D.city_id
8  INNER JOIN country C ON D.country_id = C.country_id
9  INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
10 ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
11 GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount
12 DESC LIMIT 5) AS total_amount_paid;
13
```

Data Output Messages Notifications

	average numeric
1	105.5540000000000000

```
SELECT avg(total_amount) AS "average"
FROM
(SELECT B.customer_id, B.first_name AS "first_name", B.last_name AS "last_name", C.country, D.city,
SUM(P.amount) AS "total_amount"
FROM customer B
INNER JOIN address A ON B.address_id = A.address_id
INNER JOIN city D ON A.city_id = D.city_id
INNER JOIN country C ON D.country_id = C.country_id
INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount
DESC LIMIT 5) AS total_amount_paid;
```

Step 2




```
13
14  SELECT C.country, COUNT(B.customer_id) AS all_customer_count, COUNT(top_5_customers) AS top_customer_count
15  FROM customer B
16  INNER JOIN address D ON B.address_id = D.address_id
17  INNER JOIN city A ON A.city_id = D.city_id
18  INNER JOIN country C ON A.country_id = C.country_id
19  LEFT JOIN
20  (SELECT B.customer_id, B.first_name AS "first_name", B.last_name AS "last_name", C.country, D.city,
21  SUM(P.amount) AS "total_amount"
22  FROM customer B
23  INNER JOIN address A ON B.address_id = A.address_id
24  INNER JOIN city D ON A.city_id = D.city_id
25  INNER JOIN country C ON D.country_id = C.country_id
26  INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
27  ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
28  GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount DESC limit 5)
29  AS top_5_customers
30  ON B.customer_id=top_5_customers.customer_id
31  GROUP BY C.country
32  HAVING COUNT(top_5_customers)>0
33  ORDER BY COUNT(top_5_customers),
34  COUNT(B.customer_id) DESC;
```

	A	B	C
1	country	all_customer_count	top_customer_count
2	India	60	1
3	China	53	1
4	United States	36	1
5	Japan	31	1
6	Mexico	30	1

```
SELECT C.country, COUNT(B.customer_id) AS all_customer_count, COUNT(top_5_customers) AS top_customer_count
FROM customer B
INNER JOIN address D ON B.address_id = D.address_id
INNER JOIN city A ON A.city_id = D.city_id
INNER JOIN country C ON A.country_id = C.country_id
LEFT JOIN
(SELECT B.customer_id, B.first_name AS "first_name", B.last_name AS "last_name", C.country, D.city,
SUM(P.amount) AS "total_amount"
FROM customer B
INNER JOIN address A ON B.address_id = A.address_id
INNER JOIN city D ON A.city_id = D.city_id
INNER JOIN country C ON D.country_id = C.country_id
INNER JOIN payment P ON B.customer_id = P.customer_id WHERE city IN
('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Tianjin', 'Cianjur')
GROUP BY B.customer_id, first_name, last_name, country, city ORDER BY total_amount DESC limit 5)
AS top_5_customers
ON B.customer_id=top_5_customers.customer_id
GROUP BY C.country
HAVING COUNT(top_5_customers)>0
ORDER BY COUNT(top_5_customers),
COUNT(B.customer_id) DESC;
```

Step 3

- Answer A

Yes, I believe steps 1 and 2 could be completed using join function instead of subqueries. the dat can be searched, filtered, and sorted more efficiently when join is used.

- Answer B

The purpose of subqueries is to return data that will be used as a condition in a main query to further restrict the data retrieved. They are useful when dealing with multiple queries.