
3.8: Performing Subqueries

Innocent Bayai

13 June 2024

Step 1: Find the average amount paid by the top 5 customers.

Copy the query you wrote in step 3 of the task from Exercise 3.7: Joining Tables of Data into the Query Tool. This will be your subquery, so give it an alias, "total_amount_paid," and add parentheses around it.

Write an outer statement to calculate the average amount paid.

Add your subquery to the outer statement. It will go in either the SELECT, WHERE, or FROM clause. (Hint: When referring to the subquery in your outer statement, make sure to use the subquery's alias, "total_amount_paid".)

If you've done everything correctly, pgAdmin 4 will require you to add an alias after the subquery. Go ahead and call it "average".

Copy-paste your queries and the final data output from pgAdmin 4 into your answers document.

```
SELECT AVG (total_amount_paid) AS average
FROM
(SELECT B. customer_id,
      B. first_name,
      B. last_name,
      E. country,
      D. city,
      SUM (A. amount) AS total_amount_paid
FROM payment A
      JOIN customer B ON A. customer_id = B. customer_id
      JOIN address C ON B. address_id = C. address_id
      JOIN city D ON C. city_id = D. city_id
      JOIN country E ON D. country_id = E. country_id
WHERE
D. city IN (
SELECT D.city
FROM customer B
      JOIN address C ON B. address_id = C. address_id
      JOIN city D ON C. city_id = D. city_id
      JOIN country E ON D.country_id = E.country_id
GROUP BY D.city
ORDER BY COUNT(B. customer_id) DESC
LIMIT 10
)
GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city
ORDER BY total_amount_paid DESC
LIMIT 5)
```


two queries on the “country” column. You’ll need to add a LEFT JOIN after your outer query, followed by the subquery in parentheses.

4. Give your subquery an alias so you can refer to it in your outer query, for example, “top_5_customers”.
 5. Remember to specify which columns to join the two tables on using ON. Both ON and the column names should follow the alias.
 6. Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias “top_customer_count”.
 7. Copy-paste your query and the data output into your “Answers 3.8” document.
-

```
SELECT E. country,
       COUNT (DISTINCT B. customer_id) AS all_customer_count,
       COUNT (DISTINCT top_5_customers) AS top_customer_count
FROM customer B
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D. country_id = E. country_id
LEFT JOIN
(SELECT B. customer_id,B. first_name, B. last_name, E. country, D. city,
      SUM (A. amount) AS total_amount_paid
FROM payment A
JOIN customer B ON A. customer_id = B. customer_id
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D. country_id = E. country_id
WHERE D. city IN (
SELECT D.city
FROM customer B
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D.country_id = E.country_id
WHERE E.country      IN      (SELECT      E.country
FROM      customer      B
JOIN      address C      ON      B.address_id      =      C.address_id
JOIN      city      D      ON      C.city_id      =      D.city_id
JOIN      country E      ON      D.country_id      =      E.country_id
GROUP BY      E.country
ORDER BY      COUNT(B.customer_id) DESC
LIMIT 10)
GROUP BY      E.country,D.city
ORDER BY      COUNT(B.customer_id) DESC
LIMIT 10)
GROUP BY      B.customer_id,B.first_name,B.last_name,E.country, D.city
ORDER BY      SUM(A.amount) DESC
LIMIT 5) AS      top_5_customers
ON      B.customer_id =      top_5_customers. customer_id
GROUP BY      E.country
ORDER BY      all_customer_count      DESC
LIMIT 5
```

Query Query History

```

1  SELECT E. country,
2         COUNT (DISTINCT B. customer_id) AS all_customer_count,
3         COUNT (DISTINCT top_5_customers) AS top_customer_count
4  FROM customer B
5  JOIN address C ON B. address_id = C. address_id
6  JOIN city D ON C. city_id = D. city_id
7  JOIN country E ON D. country_id = E. country_id
8  LEFT JOIN
9  (SELECT B. customer_id,B. first_name, B. last_name, E. country, D. city,
10     SUM (A. amount) AS total_amount_paid
11  FROM payment A
12  JOIN customer B ON A. customer_id = B. customer_id
13  JOIN address C ON B. address_id = C. address_id
14  JOIN city D ON C. city_id = D. city_id
15  JOIN country E ON D. country_id = E. country_id
16  WHERE D. city IN (
17  SELECT D.city
18  FROM customer B
19  JOIN address C ON B. address_id = C. address_id
20  JOIN city D ON C. city_id = D. city_id
21  JOIN country E ON D.country_id = E.country_id
22  WHERE E.country IN (SELECT E.country
23  FROM customer B
24  JOIN address C ON B.address_id = C.address_id
25  JOIN city D ON C.city_id = D.city_id
26  JOIN country E ON D.country_id = E.country_id
27  GROUP BY E.country
28  ORDER BY COUNT(B.customer_id) DESC
29  LIMIT 10)
30  GROUP BY E.country,D.city

```

```

29  LIMIT 10)
30  GROUP BY E.country,D.city
31  ORDER BY COUNT(B.customer_id) DESC
32  LIMIT 10)
33  GROUP BY B.customer_id,B.first_name,B.last_name,E.country, D.city
34  ORDER BY SUM(A.amount) DESC
35  LIMIT 5) AS top_5_customers
36  ON B.customer_id = top_5_customers.customer_id
37  GROUP BY E.country
38  ORDER BY all_customer_count DESC
39  LIMIT 5

```

Data Output Messages Notifications

	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

Total rows: 5 of 5 Query complete 00:00:00.082

Step 3:

1. Write 1 to 2 short paragraphs on the following:
 - Do you think steps 1 and 2 could be done without using subqueries?
 - When do you think subqueries are useful?
-

I am a fresher with respect to SQL but I think that subqueries must be used when and where they are the only way out to get output, otherwise simpler queries must be adopted. The complexity associated with steps 1 and 2 took me lots of time to figure out and I made mistakes too that took long to resolve. This makes it difficult for non-technical individuals to understand the meaning imbedded in the syntax.

However, subqueries remain vital where data is required under certain conditions. Subqueries under these circumstances allows conditioned data to be selected. Also, where data keeps changing, it is important to use subqueries to pick up to date data.