

3.8 Performing subqueries

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

1. 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.
2. 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”.)
3. If you've done everything correctly, pgAdmin 4 will require you to add an alias after the subquery. Go ahead and call it “average”.
4. Copy-paste your queries and the final data output from pgAdmin 4 into your answers document.

The screenshot shows the pgAdmin 4 interface with the following components:

- Top Bar:** Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, and a connection dropdown showing 'Rockbuster/postgres@PostgreSQL 14 *'.
- Toolbar:** A row of icons for file operations, search, and execution.
- Query Editor:** Contains the following SQL query:

```
1 SELECT AVG(total_amount_paid.total_amount_paid) AS average
2 FROM
3 (SELECT A.customer_id,
4      B.first_name,
5      B.last_name,
6      E.country,
7      D.city,
8      SUM(A.amount) AS total_amount_paid
9 FROM payment A
10 INNER JOIN customer B ON A.customer_id = B.customer_id
11 INNER JOIN address C ON B.address_id = C.address_id
12 INNER JOIN city D ON C.city_id = D.city_id
13 INNER JOIN country E ON D.country_id = E.country_id
14 WHERE country IN ('India','China','United States','Japan','Mexico','Brazil',
15                  'Russian Federation','Philippines','Turkey','Indonesia')
16 AND city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulla)','Kurashiki',
17              'Pingxiang','Sivas','Celaya','So Leopoldo')
18 GROUP BY A.customer_id,
19          B.first_name,
20          B.last_name,
21          E.country,
22          D.city
23 ORDER BY total_amount_paid DESC LIMIT 5) AS total_amount_paid;
```
- Data Output:** A table with one column 'average' and one row containing the value '107.3540000000000000'.

average
107.3540000000000000

Step 2 Find out how many of the top 5 customers are based within each country.

Your final output should include 3 columns:

- “country”
- “all_customer_count” with the total number of customers in each country
- “top_customer_count” showing how many of the top 5 customers live in each country

You'll notice that this step is quite difficult. We've broken down each part and provided you with some helpful hints below:

1. Copy the query from step 3 of task 3.7 into the Query Tool and add parentheses around it. This will be your inner query.
2. Write an outer statement that counts the number of customers living in each country. You'll need to refer to your entity relationship diagram or data dictionary in order to do this. The information you need is in different tables, so you'll have to use a join. To get the count for each country, use COUNT(DISTINCT) and GROUP BY. Give your second column the alias “all_customer_count” for readability.
3. Place your inner query in the outer query. Since you want to merge the entire output of the outer query with the information from your inner query, use a left join to connect the two queries on the “country” column.
4. Add a left join after your outer query, followed by the subquery in parentheses.
5. Give your subquery an alias so you can refer to it in your outer query, for example, “top_5_customers”.
6. Remember to specify which columns to join the two tables on using ON. Both ON and the column names should follow the alias.
7. Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias “top_customer_count”.
8. Copy-paste your query and the data output into your “Answers 3.8” document

Rockbuster/postgres@PostgreSQL 14

Query Editor Query History

```

6      B.first_name,
7      B.last_name,
8      E.country,
9      D.city,
10     SUM(A.amount) AS total_amount_paid
11 FROM payment A
12 INNER JOIN customer B ON A.customer_id = B.customer_id
13 INNER JOIN address C ON B.address_id = C.address_id
14 INNER JOIN city D ON C.city_id = D.city_id
15 INNER JOIN country E ON D.country_id = E.country_id
16 WHERE country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
17                  'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
18 AND city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulla)', 'Kurashiki',
19             'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
20 GROUP BY A.customer_id,
21          B.first_name,
22          B.last_name,
23          E.country,
24          D.city
25 ORDER BY total_amount_paid DESC LIMIT 5) AS top_5_customer
26 LEFT JOIN customer ON customer.customer_id = customer.customer_id
27 LEFT JOIN address ON customer.address_id = address.address_id
28 LEFT JOIN city ON address.city_id = city.city_id
29 LEFT JOIN country ON city.country_id = country.country_id
30 GROUP BY country.country
31 ORDER BY all_customer_count DESC LIMIT 5;

```

Data Output Explain Messages Notifications

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	55	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1

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?

Step 1 can be performed without subquery using aggregate function while Step 2 will require subquery because result from different table is needed.

- When do you think subqueries are useful?

Subqueries are useful when you need to summarise results from different tables that are constantly changing. Also when the result from a table (inner statement) is needed for a main query (outer statement)