

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

The screenshot shows the pgAdmin 4 Query Tool interface. The 'Query' tab is active, displaying a SQL query. The 'Data output' tab is also visible, showing the results of the query.

Query:

```
1 SELECT AVG(Total_amount_paid) AS Average_total_amount_paid
2 FROM
3 (SELECT A.amount, B. customer_id,
4 B.first_name, B.last_name, D.city, E.country,
5 sum(amount) AS Total_amount_paid
6 FROM payment A
7 INNER JOIN customer B ON A.customer_id=B.customer_id
8 INNER JOIN address C ON B.address_id=C.address_id
9 INNER JOIN city D ON C. city_id=D.city_id
10 INNER JOIN country E on D. country_id=E.country_id
11 WHERE city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','Acua','Saint Louis',
12 'So Leopoldo','Iwaki','Eskisehir')
13 GROUP BY A.amount, B.customer_id, B.first_name, B.last_name, D.city, E.country
14 ORDER BY SUM(amount) DESC
15 LIMIT 5) AS Total_amount_paid
```

Data output:

	average_total_amount_paid
1	41.916000000000000

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.

Query Query History

```

1 SELECT D.country, COUNT(DISTINCT A.customer_id)as all_customer_count,
2 COUNT(DISTINCT D. Country)AS top_customer_count
3 FROM customer A
4 INNER JOIN address B ON A.address_id=B. address_id
5 INNER JOIN city C ON B. city_id=C.city_id
6 INNER JOIN country D ON C.country_id=D.country_id
7 LEFT JOIN (SELECT A.amount,B. customer_id,
8 B.first_name, B.last_name, D.city, E.country,
9 sum(amount)AS Total_amount_paid
10 FROM payment A
11 INNER JOIN customer B ON A.customer_id=B.customer_id
12 INNER JOIN address C ON B.address_id=C.address_id
13 INNER JOIN city D ON C. city_id=D.city_id
14 INNER JOIN country E on D. country_id=E.country_id
15 WHERE city IN('Aurora','Bhusawal','Shivapuri','Cianjur','Kuwana','Acua','Saint Louis',
16 'So Leopoldo','Iwaki','Eskisehir')
17 GROUP BY A.amount,B.customer_id,B.first_name, B.last_name,D.city,E.country
18 ORDER BY SUM(amount)DESC
19 LIMIT 5)AS top_5_customers
20 ON D.country = top_5_customers.country
21 GROUP BY D.country,top_5_customers
22 ORDER BY all_customer_count DESC
23 LIMIT 5;
24

```

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 00:00:00.170

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?

❖ I believe step 1 can be done without subqueries but by aggregated functions. I will need help with this as I am not able to get the answer to the total average for the top 5 customers. Below is what I can do as I was only able to get an average for individuals of the 5 top customers. I will appreciate any help on these.

Query Query History

```

1 SELECT AVG (amount) AS AVG_amount
2 FROM payment
3 GROUP BY amount, customer_id
4 HAVING amount = SUM(amount)
5 order by SUM (amount)DESC
6 LIMIT 5

```

Data output Messages Notifications

	avg_amount numeric
1	11.990000000000000000
2	11.990000000000000000
3	11.990000000000000000
4	11.990000000000000000
5	11.990000000000000000

- ❖ I don't think step 2 can be done without subqueries because of the complexity of getting answers from different tables.

- **When do you think subqueries are useful?**

- ❖ Subqueries are useful to select rows from a table with a condition that depends on the data in the same or another table. It is useful to filter out information from a complex query that gives the information you need