3.8: Performing Subqueries

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

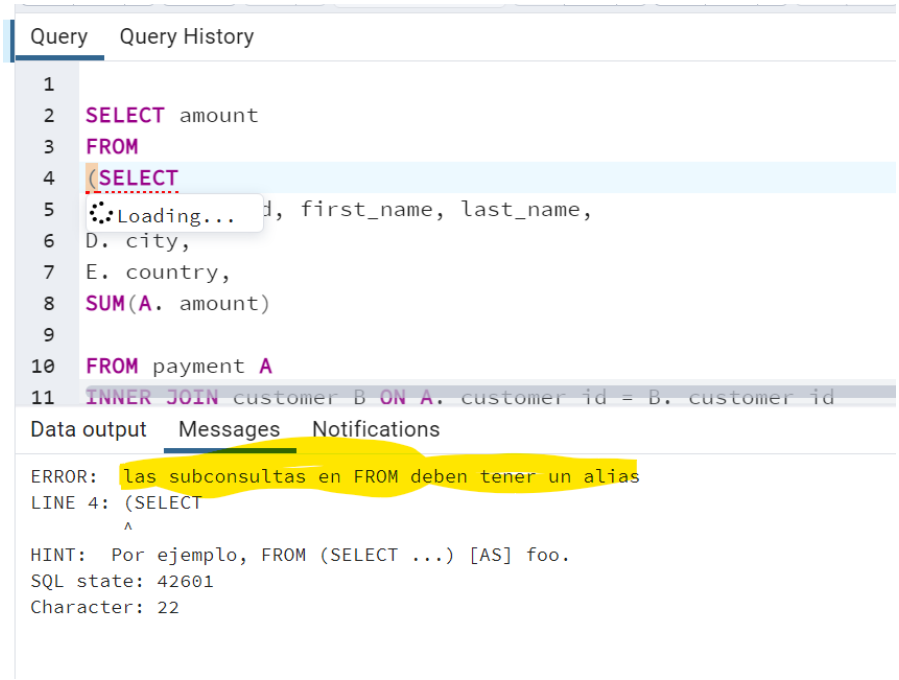
Copy the query you wrote in step 3 of the task from Exercise 3.7: Joining Tables of Data into the Query Tool. This will be your subquery, so give it an alias, "total_amount_paid," and add parentheses around it.

```
(SELECT
B. customer_id, first_name, last_name,
D. city,
E. country,
SUM(A. amount)

FROM payment A
INNER JOIN customer B ON A. customer_id = B. customer_id
INNER JOIN address C ON B. address_id = C. address_id
INNER JOIN city D ON C. city_id = D. city_id
INNER JOIN country E ON D. country_id = E. country_id
WHERE D. city IN ('Aurora', 'So Leopoldo', 'Tianjin', 'Ambattur', 'Shanwei', 'Citrus Heights',
'Iwaki', 'Hami', 'Laredo', 'Acua' )

GROUP BY B. customer_id, first_name, last_name, D. city, E. country
ORDER BY SUM(A. amount) DESC LIMIT 5)
```

Write an outer statement to calculate the average amount paid.

```
SELECT AVG (amount)
FROM payment
```

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".




```
SELECT AVG (total_amount_paid)
FROM
(SELECT
B. customer_id, first_name, last_name,
D. city,
E. country,
SUM(A. amount) AS total_amount_paid

FROM payment A
INNER JOIN customer B ON A. customer_id = B. customer_id
INNER JOIN address C ON B. address_id = C. address_id
INNER JOIN city D ON C. city_id = D. city_id
INNER JOIN country E ON D. country_id = E. country_id
WHERE D. city IN ('Aurora', 'So Leopoldo', 'Tianjin', 'Ambattur', 'Shanwei', 'Citrus Heights',
'Iwaki', 'Hami', 'Laredo', 'Acua' )

GROUP BY B. customer_id, first_name, last_name, D. city, E. country
ORDER BY SUM(A. amount) DESC LIMIT 5) AS total_amount_paid
```

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

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:

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.

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.

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.

Add a left join after your outer query, followed by the subquery in parentheses.
Give your subquery an alias so you can refer to it in your outer query, for example, "top_5_customers".

Remember to specify which columns to join the two tables on using ON. Both ON and the column names should follow the alias.

Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias "top_customer_count".

Copy-paste your query and the data output into your "Answers 3.8" document.

```
SELECT DISTINCT (A. country),
COUNT (DISTINCT D. customer_id) AS all_customer_count,
COUNT (DISTINCT A. country) AS top_customer_count

FROM country A
INNER JOIN city B ON B.country_id = A.country_id
INNER JOIN address C ON C.city_id = B.city_id
INNER JOIN customer D ON D.address_id = C.address_id
LEFT JOIN (SELECT A. customer_id, A. first_name, A. last_name, E. country, B. city,
                SUM (C. amount) AS total_amount_paid
FROM customer A
INNER JOIN address D ON D.address_id = A.address_id
INNER JOIN city B ON B.city_id = D.city_id
INNER JOIN country E ON E.country_id = B.country_id
INNER JOIN payment C ON C.customer_id = A.customer_id

WHERE B. city IN ('Aurora', 'So Leopoldo',
                                'Tianjin', 'Ambattur',
                                'Shanwei', 'Citrus Heights',
                                'Iwaki', 'Hami', 'Laredo', 'Acua' )

GROUP BY A. customer_id, E. country, B.city
```

```
ORDER BY total_amount_paid DESC LIMIT 5) AS top_5_customers
ON A. country=top_5_customers.COUNTRY
GROUP BY A. country, top_5_customers
ORDER BY all_customer_count DESC LIMIT 5
```

Step 3:

Write 1 to 2 short paragraphs on the following:

Do you think steps 1 and 2 could be done without using subqueries?

Maybe the first step we could try to do it without subquery, just with joins because at the end we are looking for a concrete average, but I don´t think we can do the second one without using subqueries or another tool/function as there are many steps on it (for me it´s been the most difficult exercise of the course)

When do you think subqueries are useful?

When you want to make a query inside another one. It´s a way to join two different clauses and it allows you to filter in two levels the information you are asking for

Step 4:

Save your "Answers 3.8" document as a PDF and upload it here for your tutor to review.