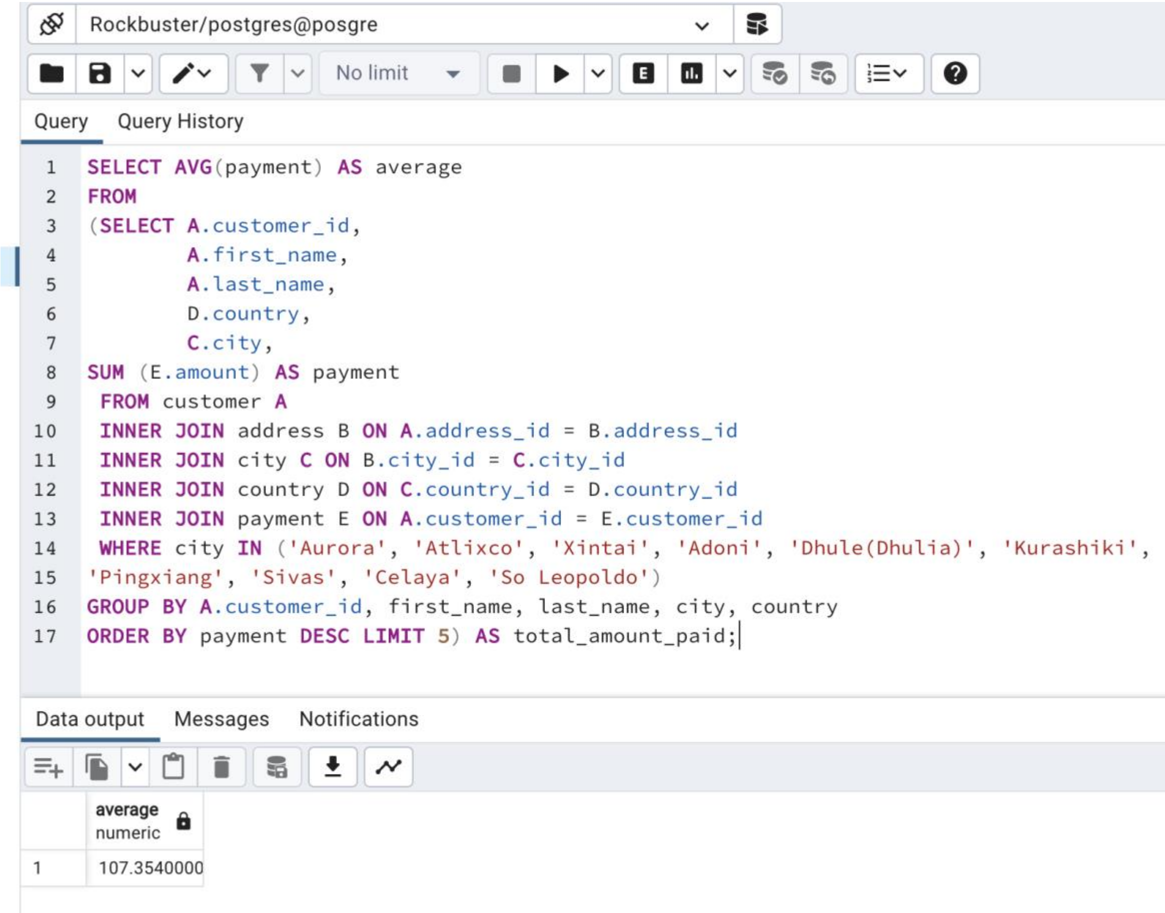
3.8: Performing Subqueries

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.

```
SELECT AVG(payment) AS average
FROM
(SELECT A.customer_id, A.first_name, A.last_name, D.country, C.city,
SUM (E.amount) AS payment
 FROM customer A
 INNER JOIN address B ON A.address_id = B.address_id
 INNER JOIN city C ON B.city_id = C.city_id
 INNER JOIN country D ON C.country_id = D.country_id
 INNER JOIN payment E ON A.customer_id = E.customer_id
 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, first_name, last_name, city, country
ORDER BY payment DESC LIMIT 5) AS total_amount_paid;
```



```sql
SELECT AVG(payment) AS average
FROM
(SELECT A.customer_id,
        A.first_name,
        A.last_name,
        D.country,
        C.city,
SUM (E.amount) AS payment
 FROM customer A
 INNER JOIN address B ON A.address_id = B.address_id
 INNER JOIN city C ON B.city_id = C.city_id
 INNER JOIN country D ON C.country_id = D.country_id
 INNER JOIN payment E ON A.customer_id = E.customer_id
 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, first_name, last_name, city, country
ORDER BY payment DESC LIMIT 5) AS total_amount_paid;
```

```
   average
   numeric
1  107.3540000
```

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.

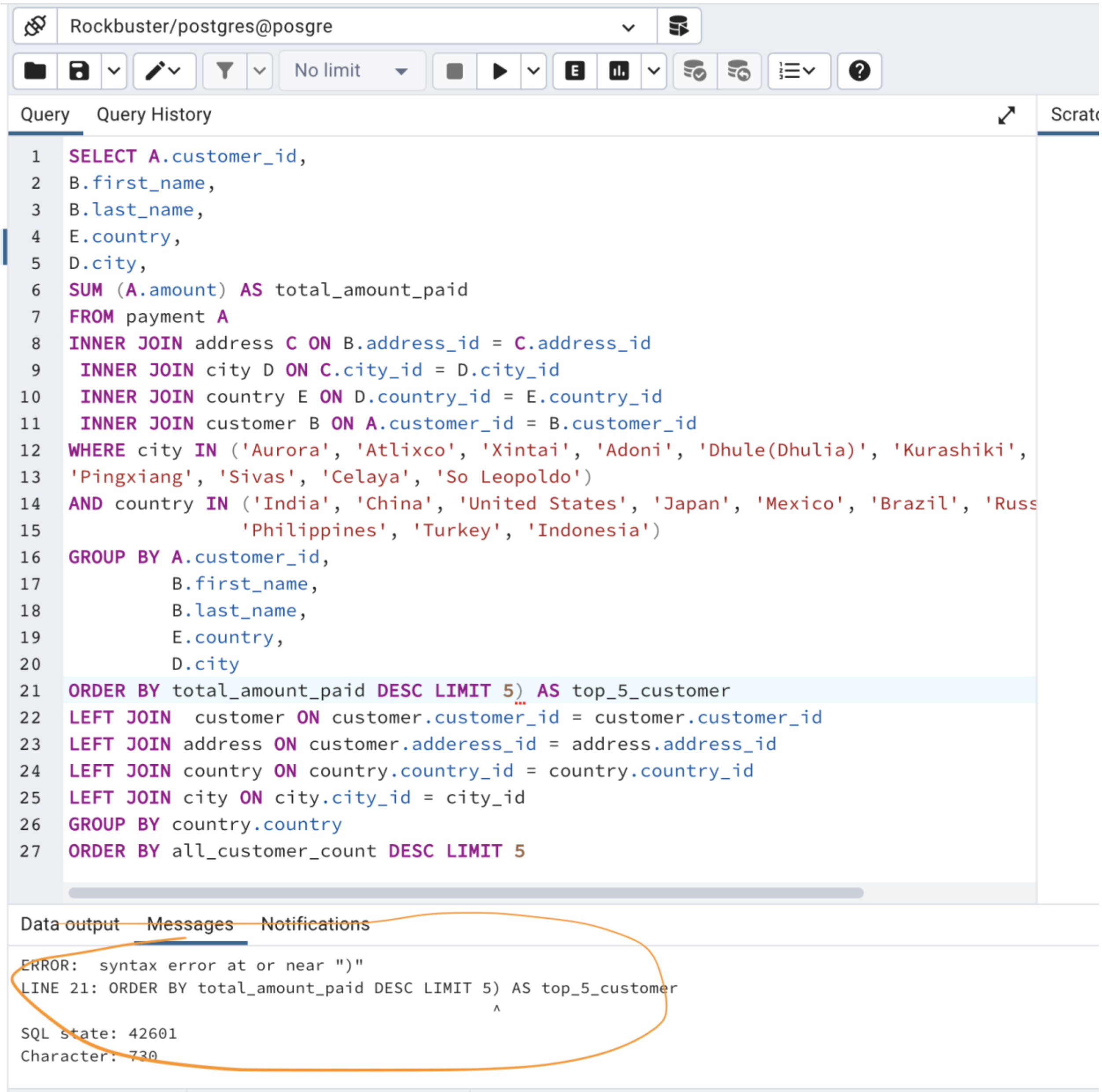
5. Give your subquery an alias so you can refer to it in your outer query, for example, “top_5_customers”.

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.

```
SELECT A.customer_id,
B.first_name,
B.last_name,
E.country,
D.city,
SUM (A.amount) AS total_amount_paid
FROM payment A
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
 INNER JOIN customer B ON A.customer_id = B.customer_id
WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
AND country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russian Federation',
                                'Philippines', 'Turkey', 'Indonesia')
GROUP BY A.customer_id,
        B.first_name,
                    B.last_name,
                    E.country,
                    D.city
ORDER BY total_amount_paid DESC LIMIT 5) AS top_5_customer
LEFT JOIN  customer ON customer.customer_id = customer.customer_id
LEFT JOIN address ON customer.adderess_id = address.address_id
LEFT JOIN country ON country.country_id = country.country_id
LEFT JOIN city ON city.city_id = city_id
GROUP BY country.country
ORDER BY all_customer_count DESC LIMIT 5
```



```sql
SELECT A.customer_id,
B.first_name,
B.last_name,
E.country,
D.city,
SUM (A.amount) AS total_amount_paid
FROM payment A
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
 INNER JOIN customer B ON A.customer_id = B.customer_id
WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
AND country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russ
               'Philippines', 'Turkey', 'Indonesia')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         E.country,
         D.city
ORDER BY total_amount_paid DESC LIMIT 5) AS top_5_customer
LEFT JOIN  customer ON customer.customer_id = customer.customer_id
LEFT JOIN address ON customer.adderess_id = address.address_id
LEFT JOIN country ON country.country_id = country.country_id
LEFT JOIN city ON city.city_id = city_id
GROUP BY country.country
ORDER BY all_customer_count DESC LIMIT 5
```

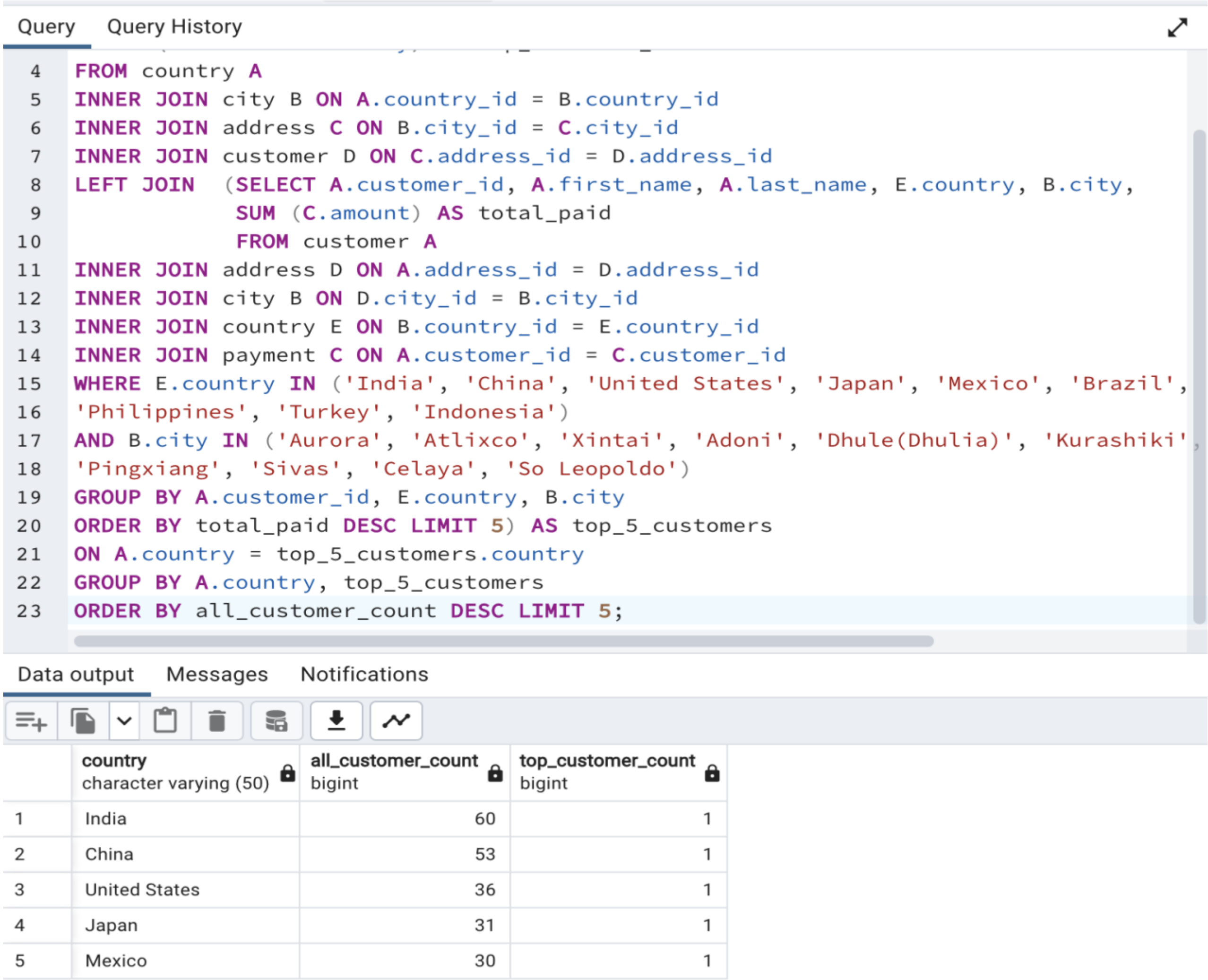
```
ERROR:  syntax error at or near ")"
LINE 21: ORDER BY total_amount_paid DESC LIMIT 5) AS top_5_customer
                                                ^

SQL state: 42601
Character: 730
```

I tried to follow your advice but it didn't work

BUT here I copied the query from your example

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 A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
LEFT JOIN (SELECT A.customer_id, A.first_name, A.last_name, E.country, B.city,

```
                        SUM (C.amount) AS total_paid
                               FROM customer A
INNER JOIN address D ON A.address_id = D.address_id
INNER JOIN city B ON D.city_id = B.city_id
INNER JOIN country E ON B.country_id = E.country_id
INNER JOIN payment C ON A.customer_id = C.customer_id
WHERE E.country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russian
Federation',
'Philippines', 'Turkey', 'Indonesia')
AND B.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, E.country, B.city
ORDER BY total_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;
```



```sql
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
LEFT JOIN  (SELECT A.customer_id, A.first_name, A.last_name, E.country, B.city,
            SUM (C.amount) AS total_paid
            FROM customer A
INNER JOIN address D ON A.address_id = D.address_id
INNER JOIN city B ON D.city_id = B.city_id
INNER JOIN country E ON B.country_id = E.country_id
INNER JOIN payment C ON A.customer_id = C.customer_id
WHERE E.country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
'Philippines', 'Turkey', 'Indonesia')
AND B.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, E.country, B.city
ORDER BY total_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;
```

	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

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?

Step 1 could have been done without a subquery by using HAVE syntax along with aggregation functions, but it would have been just as entailed.
Step 2 could not have been done unless you created an entirely new table in the database.
Subqueries are most useful for combining two steps together, such as an aggregation along with a join without having to actually create a new table in the database., Or when you need a query to automatically adjust itself to changing data and new averages