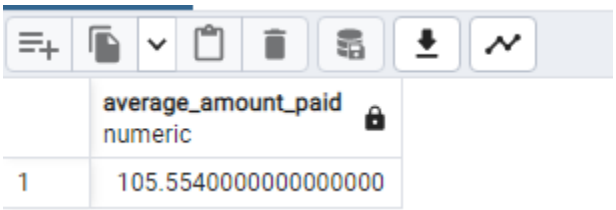
1. Copy the query you wrote in step 3 of the task from Exercise 3.7: Joining Tables of Data into the Query Tool. This will be your subquery, so give it an alias, “total_amount_paid,” and add parentheses around it.
2. Write an outer statement to calculate the average amount paid.
3. Add your subquery 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 (total_paid_amount) AS average_amount_paid
FROM (SELECT A.customer_id, E.country, D.city, B.first_name, B.first_name,
SUM (A.amount) AS total_paid_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 city IN ('Aurora', 'Acua', 'Citrus Heights','Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary
             , 'Tianjin', 'Cianjur' )
GROUP BY E.country, D.city, B.first_name, B.last_name, A.customer_id
ORDER BY total_paid_amount DESC
LIMIT 5) AS total_amount_paid
```



	average_amount_paid numeric
1	105.5540000000000000

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. 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 D.country,
COUNT (DISTINCT A.customer_id) AS total_customer_count,
COUNT (DISTINCT top_5_customers) AS top_customer_count
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
LEFT JOIN (SELECT A.customer_id, E.country, D.city, B.first_name, B.first_name,
SUM (A.amount) AS total_paid_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 city IN ('Aurora', 'Acua', 'Citrus Heights','Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary
             , 'Tianjin', 'Cianjur' )
GROUP BY E.country, D.city, B.first_name, B.last_name, A.customer_id
ORDER BY total_paid_amount DESC
LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
GROUP BY D.country
ORDER BY top_customer_count DESC
LIMIT 5
```

Data Output | Messages | Notifications

	country character varying (50)	total_customer_count bigint	top_customer_count bigint
1	Japan	31	1
2	Mexico	30	1
3	China	53	1
4	India	60	1
5	United States	36	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?

To do the queries without using subqueries would be very strenuous. Each time a query is run it generates a cost. Not using subqueries would be time and cost consuming for a company. Steps 1 and 2 contain multiple statements combined

into one. Separating would mean creating additional queries and possibly creating new tables based on the results to make running the remaining statements easier.

I think subqueries are useful when a magnitude of data is needed to produce specific results. I think breaking down the data into inner and outer queries identifies which data is necessary to produce certain results. Subqueries allow an analyst to have the most up to date information as opposed to creating completely new queries each time.