

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 AS Customer_ID,
  B.first_name AS Customer_First_Name,
  B.last_name AS Customer_Last_Name,
  D.city AS City,
  E.country AS 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=C.city_id
INNER JOIN country E on D.country_id=E.country_id

WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki', 'Pingxinag', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY Total_Amount_Paid DESC
LIMIT 5 ) AS total_amount_paid
```

```
(SELECT
  B.customer_id AS Customer_ID,
  B.first_name AS Customer_First_Name,
  B.last_name AS Customer_Last_Name,
  D.city AS City,
  E.country AS 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=C.city_id
INNER JOIN country E on D.country_id=E.country_id

WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki', 'Pingxiang',
'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY Total_Amount_Paid DESC
LIMIT 5 ) AS total_amount_paid
```

- **Write an outer statement to calculate the average amount paid.**

```
SELECT AVG ( amount)
```

```
FROM payment A
```

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

The screenshot shows the pgAdmin 4 interface with a SQL query entered in the query editor. The query is as follows:

```

1 SELECT AVG (total_amount_paid) AS average_amount_paid
2 FROM
3     (SELECT B.customer_id, B.first_name, B.last_name, D.city, E.country,
4      SUM(A.amount) AS total_amount_paid
5 FROM payment A
6 INNER JOIN customer B on A.customer_id=B.customer_id
7 INNER JOIN address C on B.address_id=C.address_id
8 INNER JOIN city D on C.city_id=D.city_id
9 INNER JOIN country E on D.country_id=E.country_id
10 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
11 GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
12 ORDER BY total_amount_paid DESC
13 LIMIT 5 ) AS total_amount_paid
14

```

Below the query editor, the 'Data Output' tab is active, showing the result of the query:

average_amount_paid
107.354

```

SELECT AVG (total_amount_paid) AS average_amount_paid
FROM
    (SELECT B.customer_id, B.first_name, B.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 city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas',
'Celaya', 'So Leopoldo')
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY total_amount_paid DESC
LIMIT 5 ) AS total_amount_paid

```

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. You’ll need to 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.

Rockbuster/postgres@PostgreSQL 15

Query Query History

```

1 SELECT E.country,
2       COUNT(DISTINCT B.customer_id) AS all_customer_count,
3       COUNT(DISTINCT top_5_customer.customer_id) AS top_customer_count
4 FROM country E
5       INNER JOIN city D ON E.country_id = D.country_id
6       INNER JOIN address C ON D.city_id = C.city_id
7       INNER JOIN customer B ON C.address_id = B.address_id
8       LEFT JOIN (SELECT B.customer_id, B.first_name, B.last_name, D.city, E.country,
9                     SUM(A.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', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
16                                'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
17                  GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
18                  ORDER BY total_amount_paid DESC
19                  LIMIT 5) AS top_5_customer
20       ON E.country = top_5_customer.country
21 GROUP BY E.country
22 ORDER BY top_customer_count DESC
23 LIMIT 5

```

Data Output Messages Notifications

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	2
2	United States	36	1
3	India	60	1
4	Turkey	15	1
5	American Samoa	1	0

```

SELECT E.country,
       COUNT(DISTINCT B.customer_id) AS all_customer_count,
       COUNT(DISTINCT top_5_customer.customer_id) AS top_customer_count
FROM country E
       INNER JOIN city D ON E.country_id = D.country_id
       INNER JOIN address C ON D.city_id = C.city_id
       INNER JOIN customer B ON C.address_id = B.address_id
       LEFT JOIN (SELECT B.customer_id, B.first_name, B.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 city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
              'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So
              Leopoldo')
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customer
ON E.country = top_5_customer.country

GROUP BY E.country
ORDER BY top_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?**
They certainly could be done without the use of subqueries but this would be more convoluted.
 - **When do you think subqueries are useful?**
Subqueries are useful when the data is dynamically-changing as well as when performing an analysis on many disparate tables with a set of prescribed conditions with which we want to filter the data.