

# Data Immersion Ach 03.08

Ryan Wick – 02/04/2025

## Directions

Create a new text document and call it “Answers 3.8.” You’ll be copy-pasting your queries, outputs, and written answers into this document, as you’ve done in previous tasks.

- **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.

## QUERY

Query Query History

```
1 SELECT AVG (total_amount_paid) AS average --average alias
2 FROM
3 (
4     SELECT SUM(amount) as total_amount_paid --total_amount_paid Alias
5     FROM customer A
6     INNER JOIN address B ON A.address_id = B.address_id
7     INNER JOIN city C ON B.city_id = C.city_id
8     INNER JOIN country D ON C.country_id = D.country_id
9     INNER JOIN payment E ON A.customer_id = E.customer_id
10    WHERE C.city IN
11    (
12        SELECT C.city
13        FROM customer A
14        INNER JOIN address B ON A.address_id = B.address_id
15        INNER JOIN city C ON B.city_id = C.city_id
16        INNER JOIN country D ON C.country_id = D.country_id
17        WHERE D.country IN
18        (
19            SELECT D.country
20            FROM customer A
21            INNER JOIN address B ON A.address_id = B.address_id
22            INNER JOIN city C ON B.city_id = C.city_id
23            INNER JOIN country D ON C.country_id = D.country_id
24            GROUP BY D.country
25            ORDER BY COUNT (A.customer_id) DESC
26            LIMIT 10
27        )
28    )
29    GROUP BY C.city
30    ORDER BY COUNT(A.customer_id) DESC
31    LIMIT 5
32 )
33 GROUP BY A.customer_id
34 ORDER BY total_amount_paid DESC
35 LIMIT 5
36 )
```

**OUTPUT**

Data Output	Messages	Noti
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🔒</div> </div>		
	average numeric	🔒
1	120.3220000000000000	

- **Step 2: Find out how many of the top 5 customers you identified in step 1 are based within each country.**
  1. 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
  2. You’ll notice that this step is quite difficult. We’ve broken down each part and provided you with some helpful hints:
    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.

## QUERY

Query Query History

```
1  SELECT D.country,
2      COUNT(DISTINCT A.customer_id) AS all_customer_count,
3      COUNT(DISTINCT top_customers.customer_id) AS top_customers_count
4  FROM customer A
5  INNER JOIN address B ON A.address_id = B.address_id
6  INNER JOIN city C ON B.city_id = C.city_id
7  INNER JOIN country D ON C.country_id = D.country_id
8  LEFT JOIN
9      (
10     SELECT A.customer_id,
11            A.first_name,
12            A.last_name,
13            C.city,
14            D.country,
15            SUM(amount) AS total_amount_paid
16     FROM customer A
17     INNER JOIN address B ON A.address_id = B.address_id
18     INNER JOIN city C ON B.city_id = C.city_id
19     INNER JOIN country D ON C.country_id = D.country_id
20     INNER JOIN payment E ON A.customer_id = E.customer_id
21     WHERE C.city IN
22         (
23         SELECT C.city
24         FROM customer A
25         INNER JOIN address B ON A.address_id = B.address_id
26         INNER JOIN city C ON B.city_id = C.city_id
27         INNER JOIN country D ON C.country_id = D.country_id
28         WHERE D.country IN
29             (
30             SELECT D.country
31             FROM customer A
32             INNER JOIN address B ON A.address_id = B.address_id
33             INNER JOIN city C ON B.city_id = C.city_id
34             INNER JOIN country D ON C.country_id = D.country_id
35             GROUP BY D.country
36             ORDER BY COUNT (A.customer_id) DESC
37             LIMIT 10
38             )
39         GROUP BY D.country, C.city
40         ORDER BY COUNT(A.customer_id) DESC
41         LIMIT 10
42         )
43     GROUP BY A.customer_id, A.first_name, A.last_name, C.City, D.country
44     ORDER BY total_amount_paid DESC
45     LIMIT 5
46     ) AS top_customers ON top_customers.country = D.country
47  GROUP BY D.country
48  ORDER BY top_customers_count DESC
49  LIMIT 5;
50
```

Total rows: 5 Query complete 00:00:00.086

## OUTPUT

Data Output Messages Notifications			
	country character varying (50)	all_customer_count bigint	top_customers_count bigint
1	Japan	31	1
2	Mexico	30	1
3	China	53	1
4	India	60	1
5	United States	36	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?
  - The short answer to whether you can complete steps 1 and 2 without subqueries is yes. BUT! You would quickly run into results that are cumbersome and difficult to understand, debug, and optimize. This issue is only exponentially compounded as the dataset gets larger. Now one might think they could simply create new tables and/or views but from what I've read this too brings cumbersome key management and possible duplicate code. Ultimately subqueries are the typical standard since you can break the problems into smaller sections. Lastly one must keep in mind that more times than not subqueries provide up-to-date information from changing data and uses very little resources.
- When do you think subqueries are useful?
  - While I touched on some of the reasons subqueries are useful in the previous question I can expand further. Subqueries are always praised for the ability to keep up to date with changing information and do it all while using very little resources. In environments such as banking, healthcare, etc. this can be crucial not only because the information is constantly being touched. There is also a huge payoff when it comes to the resource use part as many of these same environments work with massive datasets which can cost them an arm and a leg should queries not be efficient. Another key feature of subqueries over other methods is the fact that you can not only break down complex problems into smaller bite size pieces, but you can often reuse some of the queries. These two features alone are worth their weight in gold as once you can utilize the well-known code you just have to piece it together and determine what tweaks you may need to do to get the new results you are looking for.

- **Step 4:**

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