Data Analytics Immersion
3.8: Performing Subqueries

Mehreen Becker

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

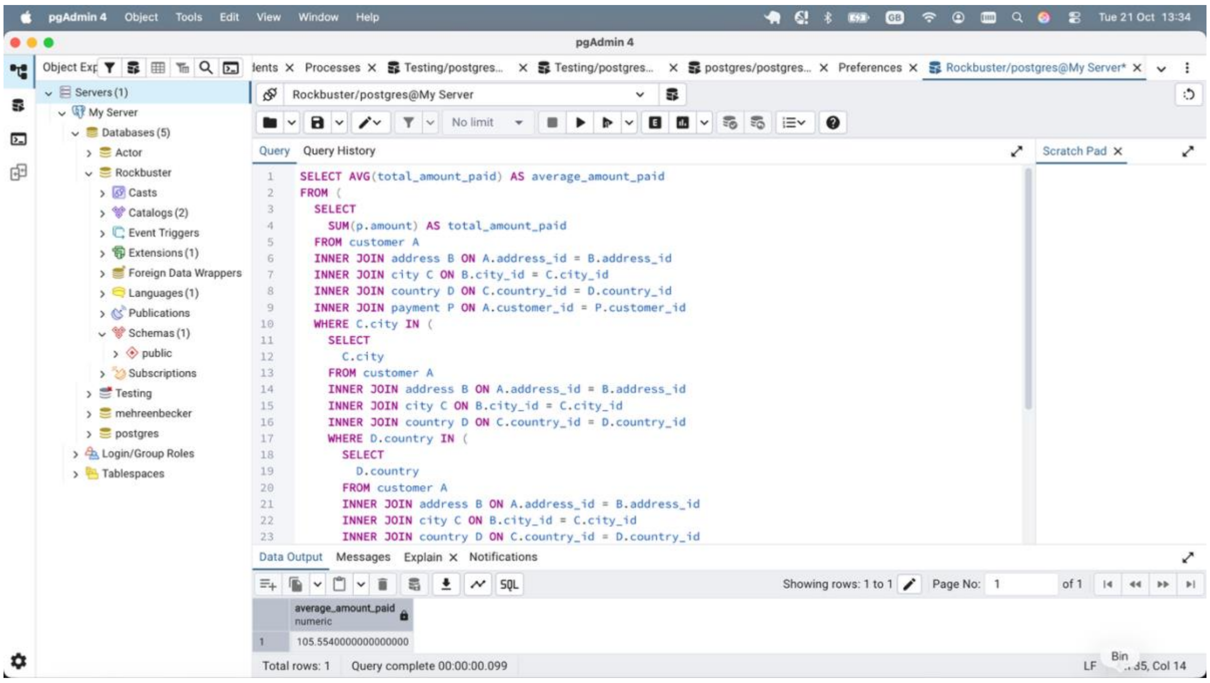
1. Copy the query you wrote in step 3 of the task from Exercise 3.7: Joining Tables of Data into the Query Tool. This will be your subquery, so give it an alias, “total_amount_paid,” and add parentheses around it.

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.



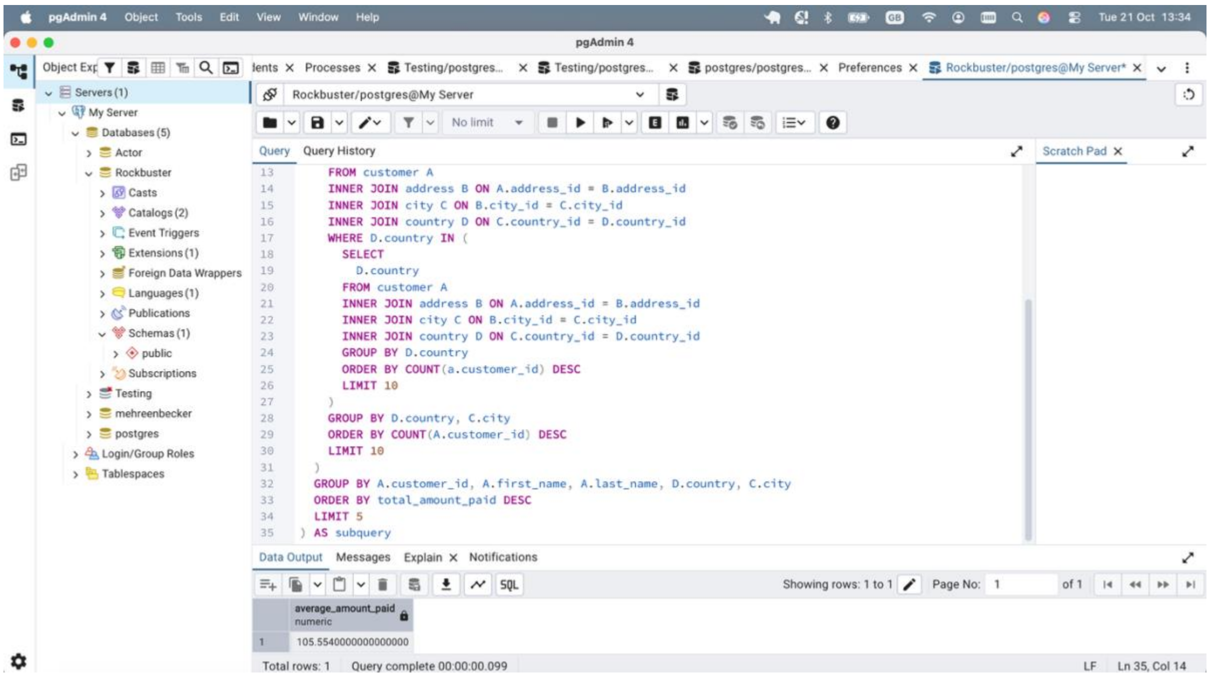
Rockbuster/postgres@My Server

```sql
SELECT AVG(total_amount_paid) AS average_amount_paid
FROM (
  SELECT
    SUM(p.amount) AS total_amount_paid
  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 P ON A.customer_id = P.customer_id
  WHERE C.city IN (
    SELECT
      C.city
    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
    WHERE D.country IN (
      SELECT
        D.country
      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
```

Showing rows: 1 to 1 Page No: 1 of 1

	average_amount_paid numeric
1	105.5540000000000000

Total rows: 1 Query complete 00:00:00.099



```
    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
    WHERE D.country IN (
      SELECT
        D.country
      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
      GROUP BY D.country
      ORDER BY COUNT(a.customer_id) DESC
      LIMIT 10
    )
    GROUP BY D.country, C.city
    ORDER BY COUNT(A.customer_id) DESC
    LIMIT 10
  )
  GROUP BY A.customer_id, A.first_name, A.last_name, D.country, C.city
  ORDER BY total_amount_paid DESC
  LIMIT 5
) AS subquery
```

```
   average_amount_paid
   numeric
1  105.5540000000000000

Total rows: 1    Query complete 00:00:00.099
```

Step 2: Find out how many of the top 5 customers you identified in step 1 are based within each country.

Your final output should include 3 columns:

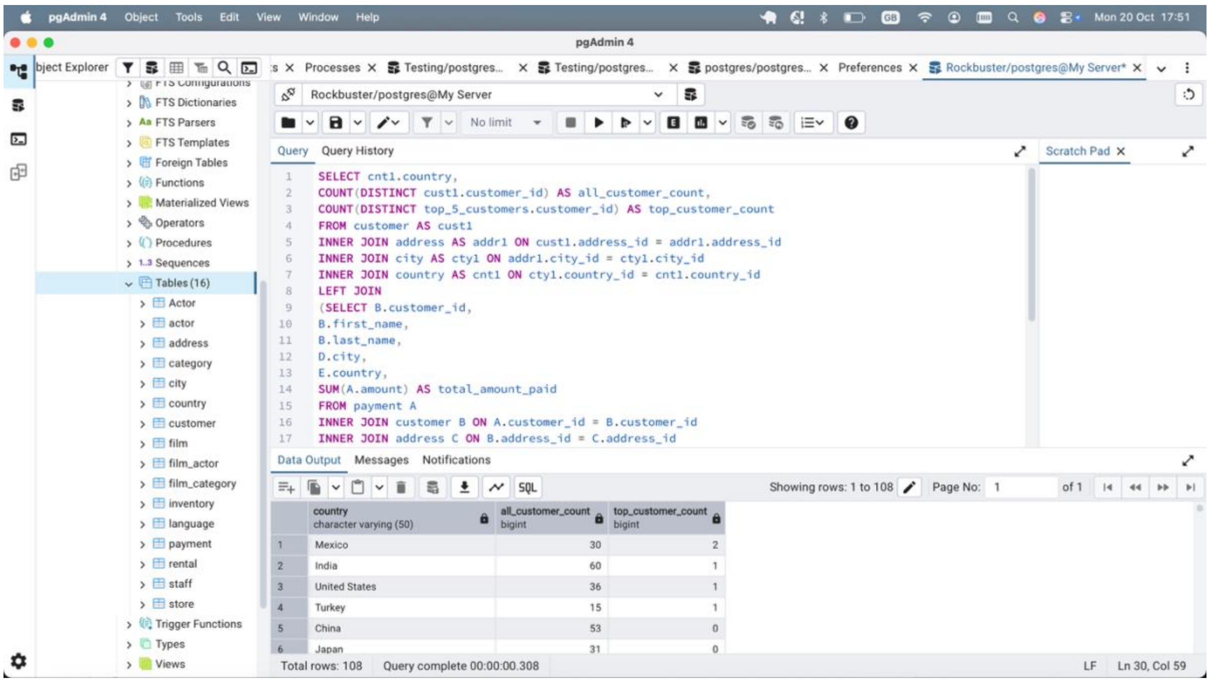
- “country”
- “all_customer_count” with the total number of customers in each country
- “top_customer_count” showing how many of the top 5 customers live in each country

You’ll notice that this step is quite difficult. We’ve broken down each part and provided you with some helpful hints:

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.

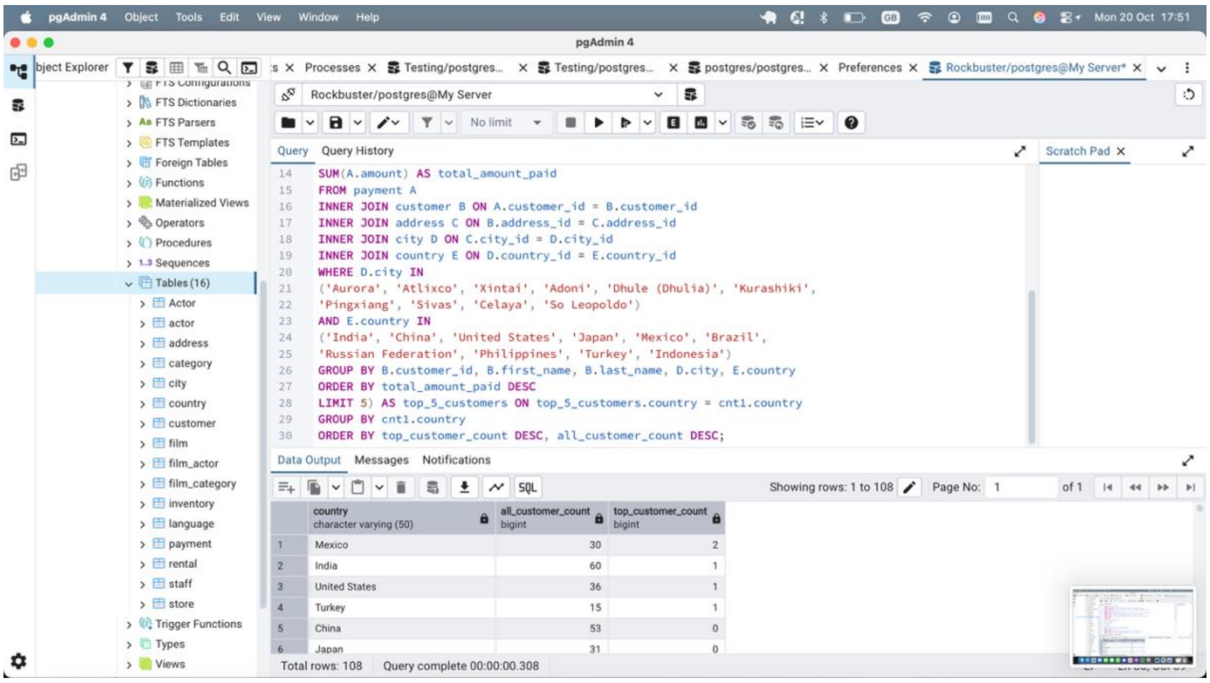


```sql
SELECT cntl.country,
COUNT(DISTINCT cust1.customer_id) AS all_customer_count,
COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
FROM customer AS cust1
INNER JOIN address AS addr1 ON cust1.address_id = addr1.address_id
INNER JOIN city AS cty1 ON addr1.city_id = cty1.city_id
INNER JOIN country AS cntl ON cty1.country_id = cntl.country_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
```

```
Showing rows: 1 to 108   Page No: 1 of 1

   country                   all_customer_count   top_customer_count
   character varying (50)    bigint               bigint
1  Mexico                                    30                    2
2  India                                     60                    1
3  United States                             36                    1
4  Turkey                                    15                    1
5  China                                     53                    0
6  Japan                                     31                    0

Total rows: 108    Query complete 00:00:00.308        LF   Ln 30, Col 59
```



```sql
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', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
AND E.country IN
('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
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_customers ON top_5_customers.country = cntl.country
GROUP BY cntl.country
ORDER BY top_customer_count DESC, all_customer_count DESC;
```

```
Showing rows: 1 to 108   Page No: 1 of 1

   country                   all_customer_count   top_customer_count
   character varying (50)    bigint               bigint
1  Mexico                                    30                    2
2  India                                     60                    1
3  United States                             36                    1
4  Turkey                                    15                    1
5  China                                     53                    0
6  Japan                                     31                    0

Total rows: 108    Query complete 00:00:00.308
```

country	all_customer_count	top_customer_count
Mexico	30	2
India	60	1

United States	36	1
Turkey	15	1
China	53	0
Japan	31	0
Brazil	28	0
Russian Federation	28	0
Philippines	20	0
Indonesia	14	0
Argentina	13	0
Nigeria	13	0
South Africa	11	0
Taiwan	10	0
United Kingdom	9	0
Poland	8	0
Iran	8	0
Venezuela	7	0
Germany	7	0
Italy	7	0
Egypt	6	0
Ukraine	6	0
Vietnam	6	0
Colombia	6	0
Spain	5	0
Canada	5	0
Saudi Arabia	5	0
Netherlands	5	0
Pakistan	5	0
South Korea	5	0
Peru	4	0
France	4	0
Yemen	4	0
Israel	4	0
Algeria	3	0
Switzerland	3	0
Tanzania	3	0
United Arab Emirates	3	0
Morocco	3	0
Bangladesh	3	0
Chile	3	0
Thailand	3	0
Malaysia	3	0

Austria	3	0
Paraguay	3	0
Mozambique	3	0
Ecuador	3	0
Dominican Republic	3	0
Sudan	2	0
Bolivia	2	0
Greece	2	0
Belarus	2	0
Bulgaria	2	0
Yugoslavia	2	0
Cambodia	2	0
Cameroon	2	0
Romania	2	0
Puerto Rico	2	0
Kazakstan	2	0
Kenya	2	0
Angola	2	0
Latvia	2	0
Azerbaijan	2	0
Congo, The Democratic Republic of the	2	0
Oman	2	0
Myanmar	2	0
French Polynesia	2	0
Zambia	1	0
American Samoa	1	0
Anguilla	1	0
Armenia	1	0
Bahrain	1	0
Brunei	1	0
Chad	1	0
Czech Republic	1	0
Estonia	1	0
Ethiopia	1	0
Faroe Islands	1	0
Finland	1	0
French Guiana	1	0
Gambia	1	0
Greenland	1	0
Holy See (Vatican City State)	1	0

Hong Kong	1	0
Hungary	1	0
Iraq	1	0
Kuwait	1	0
Liechtenstein	1	0
Lithuania	1	0
Madagascar	1	0
Malawi	1	0
Moldova	1	0
Nauru	1	0
Nepal	1	0
New Zealand	1	0
North Korea	1	0
Runion	1	0
Saint Vincent and the Grenadines	1	0
Senegal	1	0
Slovakia	1	0
Sri Lanka	1	0
Sweden	1	0
Tonga	1	0
Tunisia	1	0
Turkmenistan	1	0
Tuvalu	1	0
Virgin Islands, U.S.	1	0
Afghanistan	1	0

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?

In principle, steps 1 and 2 could have been done without using subqueries. However, the benefit of using a subquery is that one can use the original query in our own query to filter out the data that we are looking for. Subqueries, in this instance, saves us having to run multiple queries to end up at the same result.

Since we are able to nestle the inner statement in the outer statement, subqueries are useful when we want to be flexible. They can also make complex queries easier to read, and thus, I can imagine probably are also more efficient.