DATA IMMERSION | Databases & SQL for Analysts
3.8: Performing Subqueries

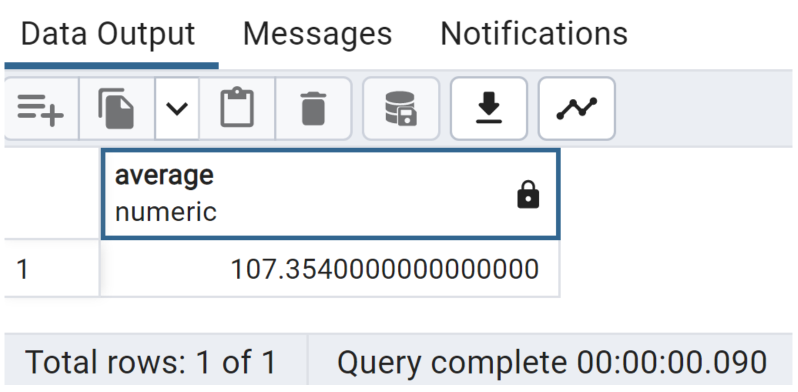
SUBMITTED BY:
Grace Skelley on 11.6.2022

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

QUERY

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

AVERAGE



```
Data Output    Messages    Notifications

     average
     numeric
1    107.3540000000000000

Total rows: 1 of 1    Query complete 00:00:00.090
```

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*

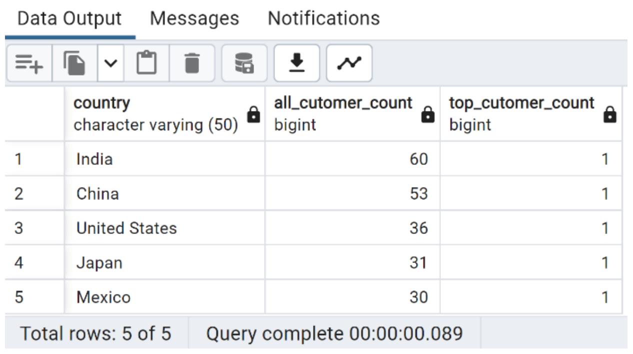
You'll notice that this step is quite difficult. We've broken down each part and provided you with some helpful hints below:

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.*

QUERY

```sql
SELECT DISTINCT(D.country)
      ,COUNT(DISTINCT A.customer_id) AS all_cutomer_count
      ,COUNT(DISTINCT D.country) AS top_cutomer_count
FROM country D
INNER JOIN city C ON C.country_id = D.country_id
INNER JOIN address B ON B.city_id = C.city_id
INNER JOIN customer A ON A.address_id = B.address_id
LEFT JOIN (SELECT A.customer_id,A.first_name ,A.last_name
      ,C.city ,D.country ,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 E.customer_id = A.customer_id
WHERE C. city IN ('Aurora' ,'Atlixco' ,'Xintai' ,'Adoni' ,'Dhule (Dhulia)'
,'Kurashiki' ,'Pingxiang' ,'Sivas' ,'Celaya' ,'So Leopoldo')
GROUP BY A.customer_id ,A.first_name ,A.last_name ,C.city ,D.country
ORDER BY SUM(E.amount) DESC
LIMIT 5) AS top_5_customers ON D.country=top_5_customers.COUNTRY
GROUP BY D.country ,top_5_customers
ORDER BY all_cutomer_count DESC
LIMIT 5;
```

TOP 5 CUSTOMER
(WITHIN EACH COUNTRY)



Data Output Messages Notifications

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

Total rows: 5 of 5 Query complete 00:00:00.089

3. Write 1 to 2 short paragraphs on the following:

➢ *Do you think steps 1 and 2 could be done without using subqueries?*

STEP 1: Data output can be achieved without subqueries as it only requires average amount paid. And so, `INNER JOIN` may not be optimal for this query as it can be simplified through adding `HAVING` clause with aggregate functions.

STEP 2: No, the links between tables is too complex that it will take much longer to generate without subqueries. That being said, not using subqueries may cost time inefficiencies.

➢ *When do you think subqueries are useful?*

Subqueries are best applied for when there are multiple tables are involved in achieving a targeted output. This demonstrates how nesting in inner statement in the `FROM` clause lets the outer statement use the results of the inner statement as its underlying table.