Databases & SQL for Analysts

Task 3.8: Performing Subqueries

Ngawang Dhundup

LINK NAME:

Directions:

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.

```
(SELECT
A.customer_id,
A.first_name AS customer_first_name,
A.last_name AS customer_last_name,
A.email AS customer_email,
C.city,
D.country,
SUM(E.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 E ON A.customer_id = E.customer_id
WHERE C.city IN ('Aurora',
'Atlixco',
'Xintai',
'Adoni',
'Kurashiki',
'Dhule (Dhulia)',
'Pingxiang',
'Ozamis',
'Nezahualcyotl',
'So Leopoldo')
GROUP BY
A.customer_id,
A.first_name,
A.last_name,
A.email,
C.city, D.country
ORDER BY total_amount_paid desc
LIMIT 5) AS total_amount_paid
```

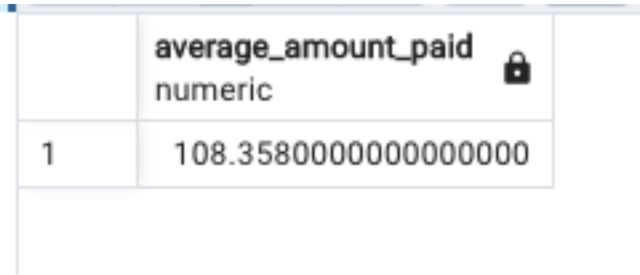
2. Write an outer statement to calculate the average amount paid.

```
SELECT AVG(total_amount_paid.total_paid_by_customer) AS average_amount_paid
FROM
```

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".)

```
SELECT AVG(total_amount_paid.total_paid_by_customer) AS average_amount_paid
FROM (SELECT
A.customer_id,
A.first_name AS customer_first_name,
A.last_name AS customer_last_name,
A.email AS customer_email,
C.city,
D.country,
SUM(E.amount) AS total_paid_by_customer
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 A.customer_id = E.customer_id
WHERE C.city IN ('Aurora',
'Atlixco',
'Xintai',
'Adoni',
'Kurashiki',
'Dhule (Dhulia)',
'Pingxiang',
'Ozamis',
'Nezahualcyotl',
'So Leopoldo')
GROUP BY
A.customer_id,
A.first_name,
A.last_name,
A.email,
C.city, D.country
ORDER BY total_paid_by_customer DESC
LIMIT 5) AS 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.



```
     average_amount_paid
     numeric
1    108.3580000000000000
```

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:

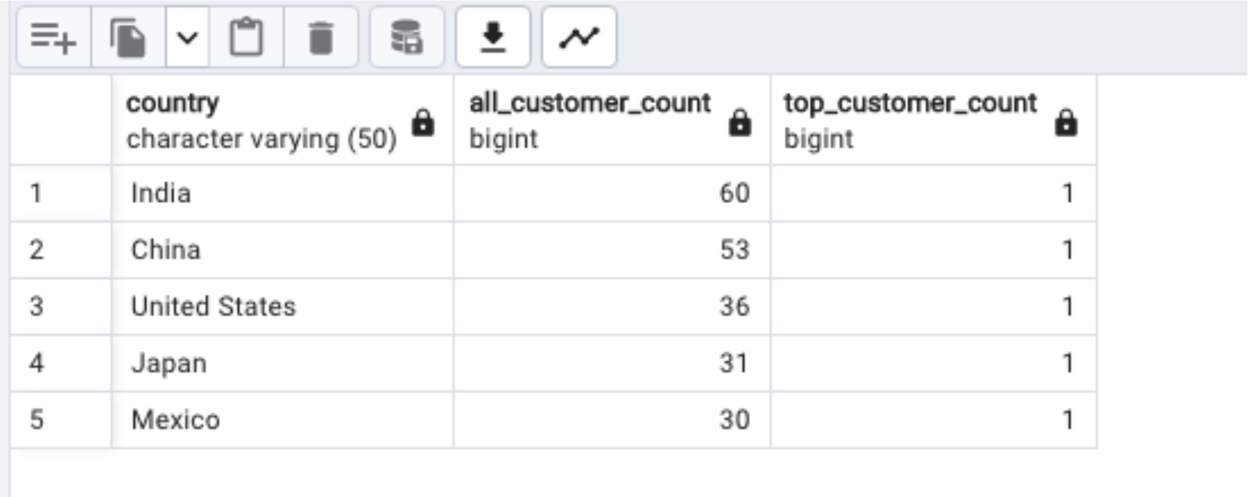
1. Copy the query from step 3 of task 3.7 into the Query Tool and add parentheses around it. This will be your inner query.
2. Write an outer statement that counts the number of customers living in each country. You’ll need to refer to your entity relationship diagram or data dictionary in order to do 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.
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.

```
--outer query--
SELECT
        DISTINCT D.country,
        COUNT(DISTINCT A.customer_id) AS all_customer_count,
        COUNT(DISTINCT D.country) 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
--joining inner query--
LEFT JOIN
        (SELECT
        A.customer_id,
        A.first_name AS customer_first_name,
        A.last_name AS customer_last_name,
        A.email AS customer_email,
        C.city,
        D.country,
        SUM(E.amount) AS total_paid_by_customer
        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 A.customer_id = E.customer_id
    WHERE C.city IN ('Aurora',
    'Atlixco',
    'Xintai',
    'Adoni',
    'Kurashiki',
    'Dhule (Dhulia)',
    'Pingxiang',
    'Ozamis',
    'Nezahualcyotl',
    'So Leopoldo')
    GROUP BY
    A.customer_id,
    A.first_name,
    A.last_name,
    A.email,
    C.city, D.country
    ORDER BY total_paid_by_customer DESC
    LIMIT 5) AS top_5_customers
ON D.country = top_5_customers.country
--grouping--
GROUP BY D.country, top_5_customers
ORDER BY all_customer_count DESC
LIMIT 5;
```



	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	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?
        - I believe the first step could have been done without a subquery, simply by using a having clause to do the average of amount from the payment table. However the second step needs the subquery since we need to reference the top 5 customers from the subquery to create another query counting the top 5 customers from each country.
    - When do you think subqueries are useful?
        - When you have to refer a query to another query. Or when you use a view or table as a reference to answer a different question. Similar to VLOOKUPS I believe.

Step 4:

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