Answers 3.8

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.

```
SELECT AVG(total_amount_paid.total_amount) AS avg_amount_paid
       FROM
       (SELECT A.customer_id,
A.first_name,
A.last_name,
C.city,
D.country,
SUM(E.amount) AS total_amount
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 city IN('Aurora', 'Shimoga', 'Aparecida de Goinia', 'Emeishan', 'Pontianak',
'Tarsus', 'Atlixco', 'Zalantun', 'Rio Claro', 'Tokat')
GROUP BY A.customer_id, C.city, D.country
ORDER BY total_amount DESC
LIMIT 5) AS total_amount_paid
```

avg_amount_paid
120.322

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.

```
SELECT country.country,
COUNT (DISTINCT customer.customer_id) AS all_customer_count,
COUNT (DISTINCT top_5_customer.customer_id) AS top_customer_count
FROM customer
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city ON address.city_id = city.city_id
INNER JOIN country ON city.country_ID = country.country_ID
LEFT JOIN  (SELECT customer.customer_id,
customer.first_name,
customer.last_name,
city.city,
country.country,
SUM(payment.amount) AS total_amount
```

```
FROM customer
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city ON address.city_id = city.city_id
INNER JOIN country ON city.country_id = country.country_id
INNER JOIN payment ON customer.customer_id = payment.customer_id
WHERE city IN('Aurora', 'Shimoga', 'Aparecida de Goinia', 'Emeishan', 'Pontianak',
'Tarsus', 'Atlixco', 'Zalantun', 'Rio Claro', 'Tokat')
GROUP BY customer.customer_id, city.city, country.country
ORDER BY total_amount DESC
LIMIT 5) AS top_5_customer ON country.country = top_5_customer.country
GROUP BY country.country
ORDER BY top_customer_count DESC, all_customer_count DESC
LIMIT 5
```

country	all_customer_count	top_customer_count
China	53	1
United States	36	1
Mexico	30	1
Turkey	15	1
Indonesia	14	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?
    - When do you think subqueries are useful?
        - Both step 1 and 2 require subqueries as their results depend on multiple queries and their output. For step 1 we need to know the top 5 customers and their total payments before being able to calculate the average and step two joins two temporary tables based on two different queries, that could not be achieve with only JOIN.
          Consequently subqueries are then useful and necessary when the result requires multiple queries and each of the subqueries provides part of the table for the overall query.