Task 3.8 Performing Subqueries

By Lee Heng Chuah

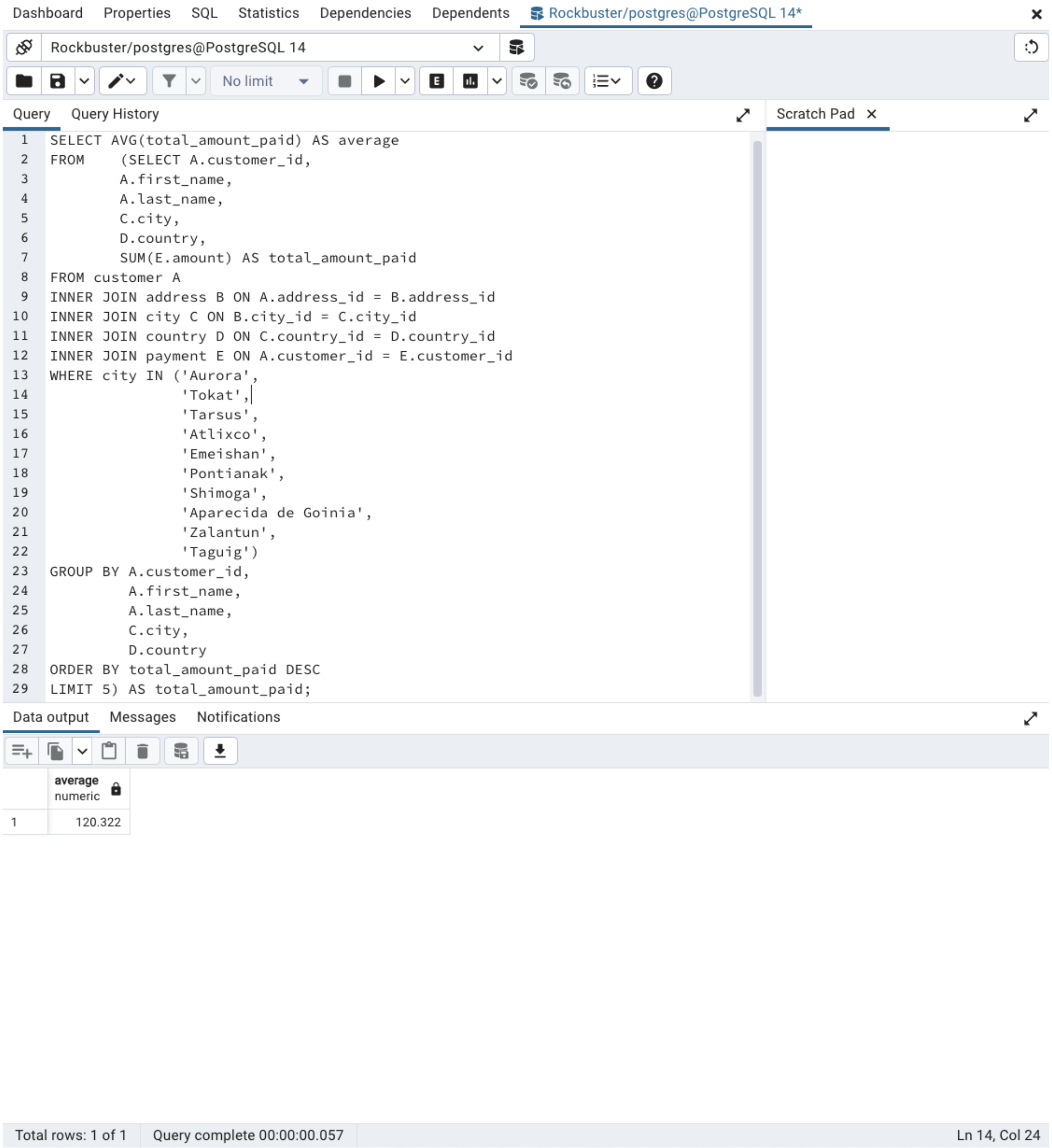
The previous task saw you write a complex query with multiple joins and filters. Now your manager wants you to analyse the results of that query. The only catch is that revisiting your query could take quite some time, not to mention the risk of breaking it. Instead, you decide to use it as a subquery (or inner query) to answer the business questions listed below.

Directions:

Create a new text document and call it “Answers 3.8”. You will be copy-pasting your queries, outputs and written answers into this document, as you have 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 have done everything correctly, PgAdmin 4 will require you to add an alias after the subquery. Go ahead and call it “average”.
5. Copy-past your queries and the final data output from PgAdmin 4 into your answers document.



```
SELECT AVG(total_amount_paid) AS average
FROM    (SELECT A.customer_id,
        A.first_name,
        A.last_name,
        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 city IN ('Aurora',
               'Tokat',
               'Tarsus',
               'Atlixco',
               'Emeishan',
               'Pontianak',
               'Shimoga',
               'Aparecida de Goinia',
               'Zalantun',
               'Taguig')
GROUP BY A.customer_id,
         A.first_name,
         A.last_name,
         C.city,
         D.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid;
```

	average numeric
1	120.322

Total rows: 1 of 1   Query complete 00:00:00.057   Ln 14, Col 24

Step 2: Find out how many of the top 5 customers are based within each country

Your final output should include 3 columns:

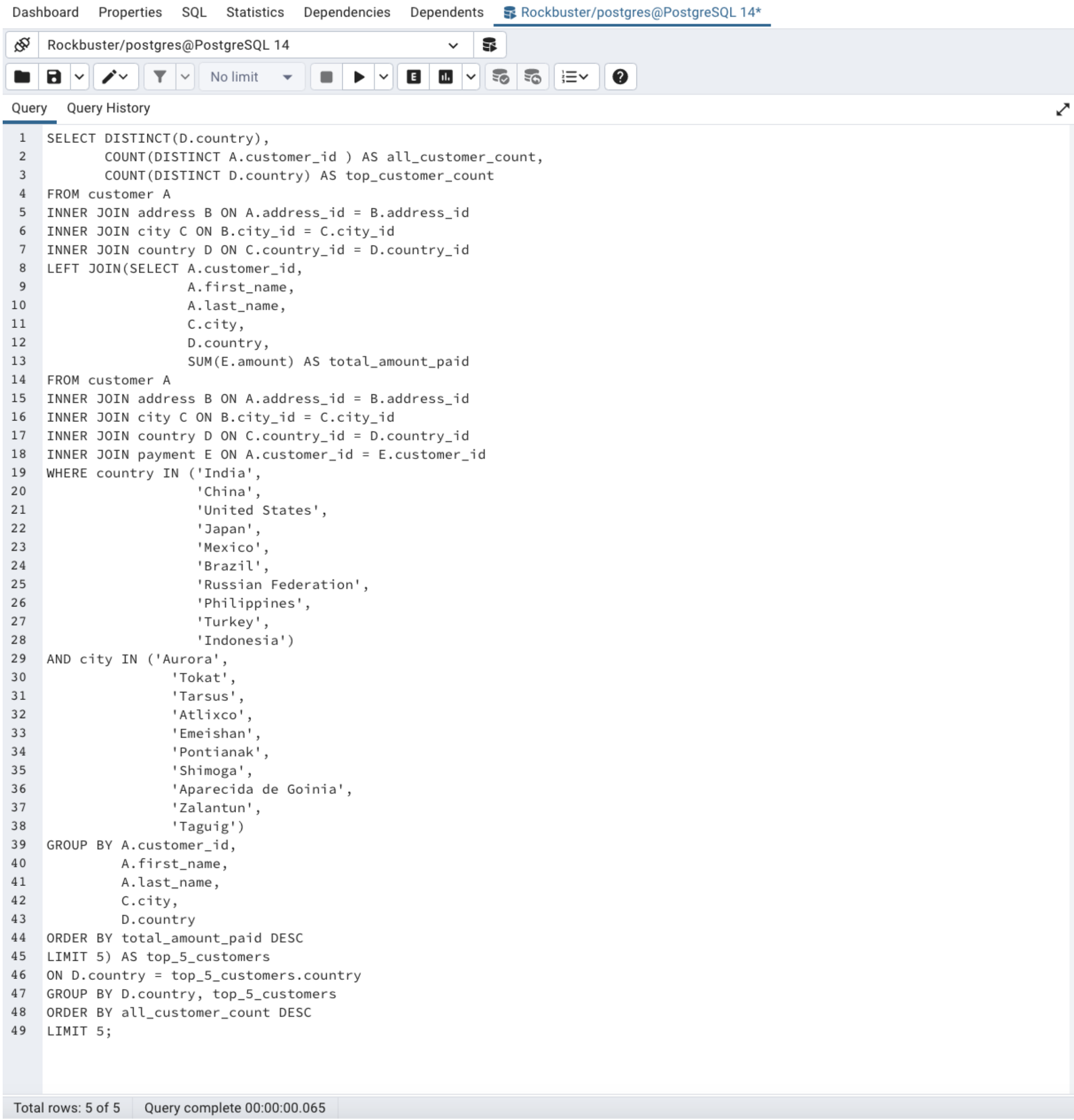
- “country”
- “all_customer_count” with the total number of customers in each country
- “top_customer_count” showing how many of the top 5 customers live in each country

You will notice that this step is quite difficult. We have 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 each country. You will need to refer to your entity relationship diagram or data dictionary in order to do this. The information you need is in

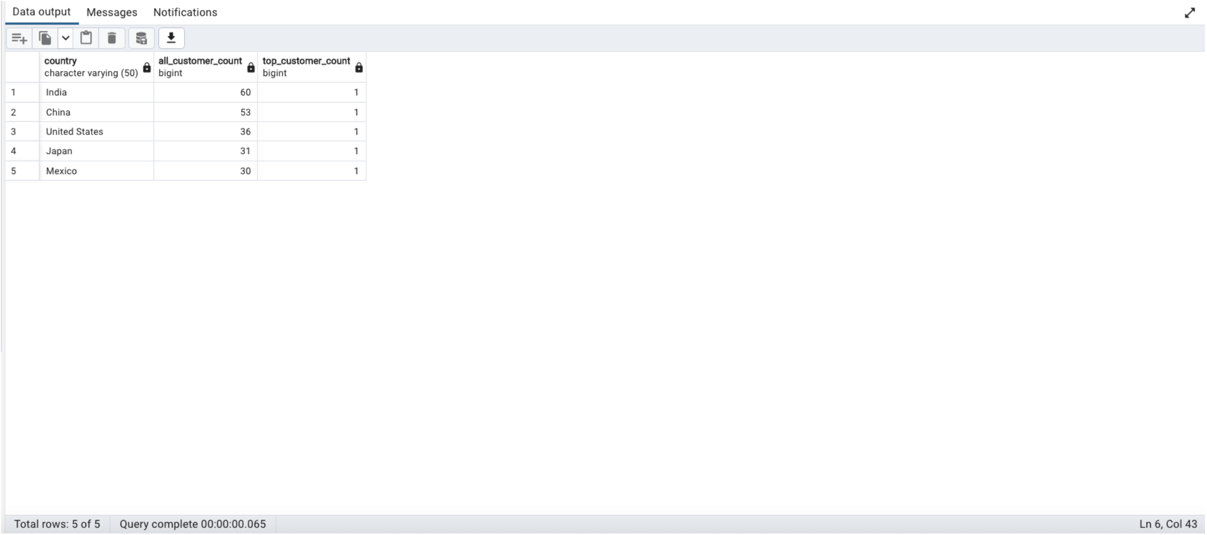
different tables, so you will have to use a join. To get the count for each country, use COUNT(DISTINCT) and GROUP BY. Give your second column the alias “all_customer_count” for readability.

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”.



```
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
LEFT JOIN(SELECT A.customer_id,
                 A.first_name,
                 A.last_name,
                 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 country IN ('India',
                  'China',
                  'United States',
                  'Japan',
                  'Mexico',
                  'Brazil',
                  'Russian Federation',
                  'Philippines',
                  'Turkey',
                  'Indonesia')
AND city IN ('Aurora',
               'Tokat',
               'Tarsus',
               'Atlixco',
               'Emeishan',
               'Pontianak',
               'Shimoga',
               'Aparecida de Goinia',
               'Zalantun',
               'Taguig')
GROUP BY A.customer_id,
         A.first_name,
         A.last_name,
         C.city,
         D.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers
ON D.country = top_5_customers.country
GROUP BY D.country, top_5_customers
ORDER BY all_customer_count DESC
LIMIT 5;
```

Total rows: 5 of 5   Query complete 00:00:00.065



	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

Total rows: 5 of 5 Query complete 00:00:00.065 Ln 6, Col 43

Step 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 could be carried out without using subqueries. However, Step 2 took sources from different tables, which it could be benefited to use subqueries.

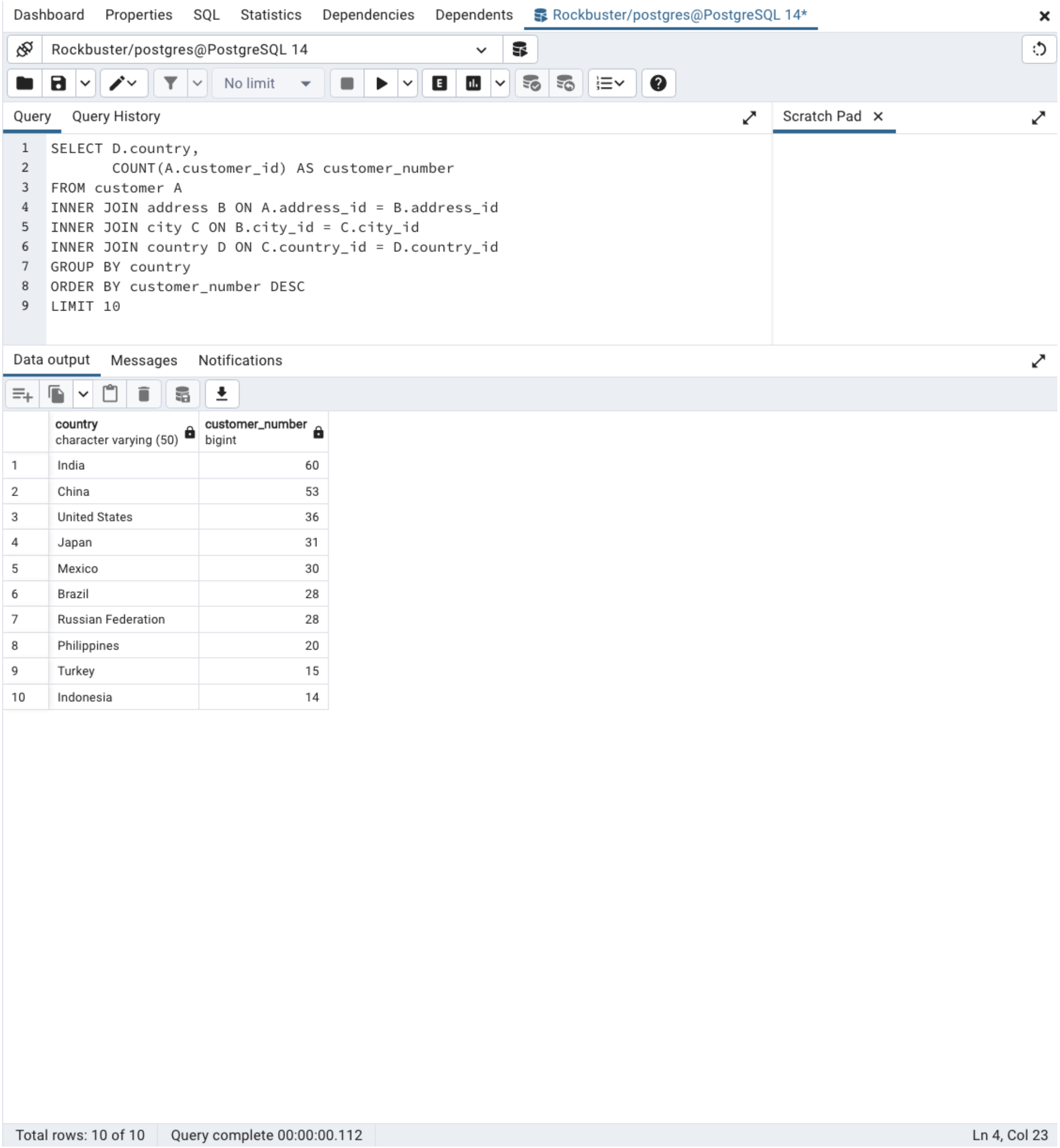
- When do you think subqueries are useful?

  Subqueries divide the complex query into small chunks queries followed a series of logical steps and thes isolated part of queries could be easy to understand and code maintenance is also easy to handle in future. Also, the subqueries also allow us to use the results of another query in the outer query.

Step 4:

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

a. Copy-paste your query and its output into your answers document



```
SELECT D.country,
       COUNT(A.customer_id) AS customer_number
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 country
ORDER BY customer_number DESC
LIMIT 10
```

	country character varying (50)	customer_number bigint
1	India	60
2	China	53
3	United States	36
4	Japan	31
5	Mexico	30
6	Brazil	28
7	Russian Federation	28
8	Philippines	20
9	Turkey	15
10	Indonesia	14

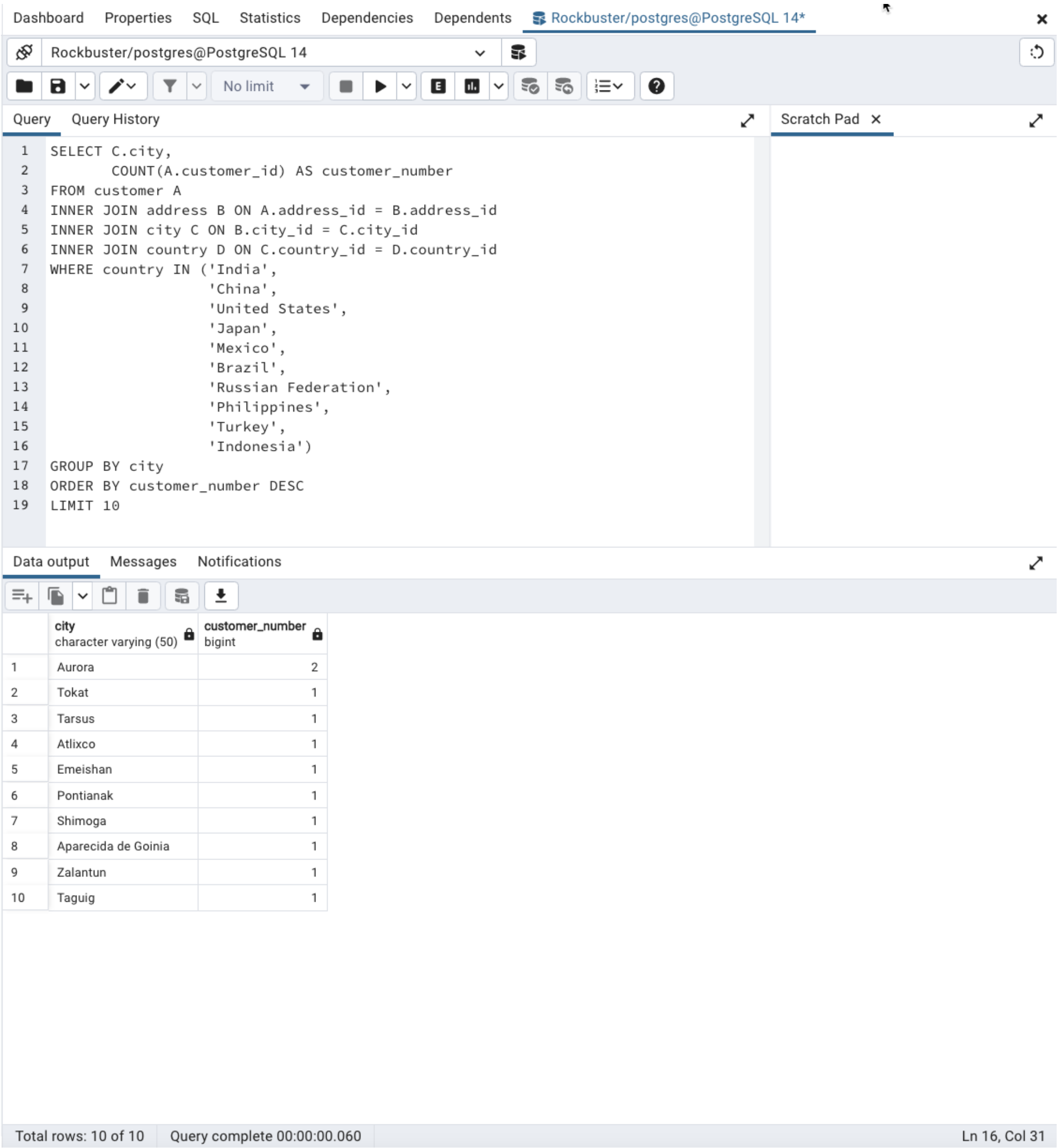
Total rows: 10 of 10 Query complete 00:00:00.112 Ln 4, Col 23

b. Write a few sentences on how you approached this query and why. It is important that you can explain your thought process when writing queries, especially for future interviews.

I revisited and understood the customer, address, city, and country data. I then joined the customer data with the address, city, and country data with the inner join query. I put the GROUP BY query and counted the customer based on country. Later, I put the ORDER BY command based on customer numbers from highest to lowest and LIMIT to 10.

2. Write a query to find the top 10 cities within the top 10 countries identified in step 1.

a. Copy-paste your query and its output into your answers document



```
SELECT C.city,
       COUNT(A.customer_id) AS customer_number
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 country IN ('India',
                  'China',
                  'United States',
                  'Japan',
                  'Mexico',
                  'Brazil',
                  'Russian Federation',
                  'Philippines',
                  'Turkey',
                  'Indonesia')
GROUP BY city
ORDER BY customer_number DESC
LIMIT 10
```

	city character varying (50)	customer_number bigint
1	Aurora	2
2	Tokat	1
3	Tarsus	1
4	Atlixco	1
5	Emeishan	1
6	Pontianak	1
7	Shimoga	1
8	Aparecida de Goinia	1
9	Zalantun	1
10	Taguig	1

Total rows: 10 of 10   Query complete 00:00:00.060   Ln 16, Col 31

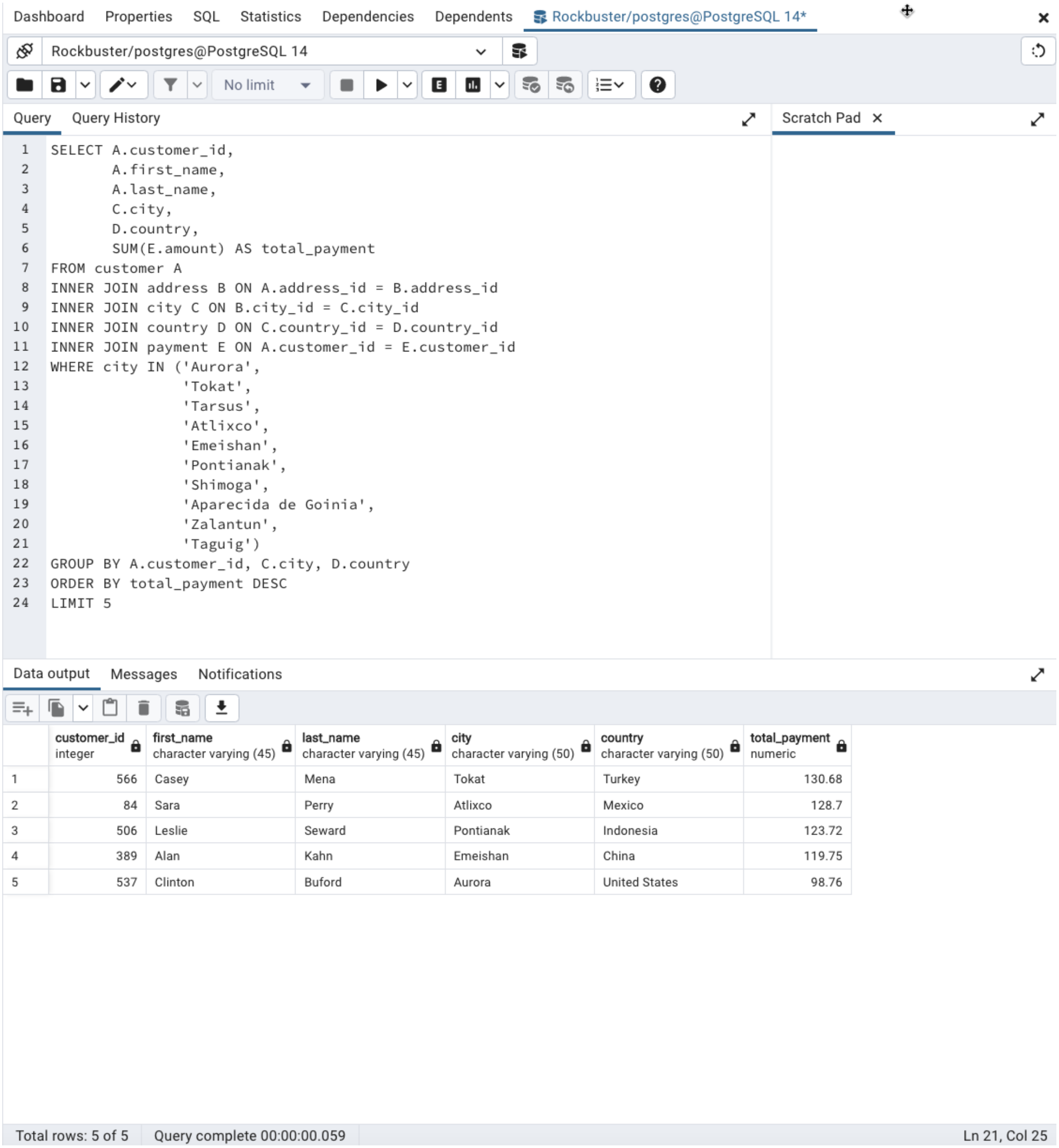
b. Write a short explanation of how you approached this query and why.

I revisited my existing query. I changed the SELECT from country to city. To find the top 10 cities within the top 10 countries, I listed out the top 10 countries by the WHERE query. Lastly, I changed the GROUP BY from country to city.

3. Write a query to find the top 5 customers in the top 10 cities who have paid the highest total amounts to Rockbuster. The customer team would like to reward them for their loyalty.

a. Tip: After the join syntax, you will need to use the WHERE clause with an operator, followed by GROUP BY and ORDER BY. Your output should include the following columns: Customer ID, Customer First Name and Last Name, Country, City, Total Amount Paid

b. Copy-past your query and its output into your answers document



```sql
SELECT A.customer_id,
       A.first_name,
       A.last_name,
       C.city,
       D.country,
       SUM(E.amount) AS total_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 A.customer_id = E.customer_id
WHERE city IN ('Aurora',
               'Tokat',
               'Tarsus',
               'Atlixco',
               'Emeishan',
               'Pontianak',
               'Shimoga',
               'Aparecida de Goinia',
               'Zalantun',
               'Taguig')
GROUP BY A.customer_id, C.city, D.country
ORDER BY total_payment DESC
LIMIT 5
```

	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_payment numeric
1	566	Casey	Mena	Tokat	Turkey	130.68
2	84	Sara	Perry	Atlixco	Mexico	128.7
3	506	Leslie	Seward	Pontianak	Indonesia	123.72
4	389	Alan	Kahn	Emeishan	China	119.75
5	537	Clinton	Buford	Aurora	United States	98.76

Total rows: 5 of 5   Query complete 00:00:00.059   Ln 21, Col 25

4. Save your “Answer 3.7” document as a PDF and upload it here for your tutor to review.