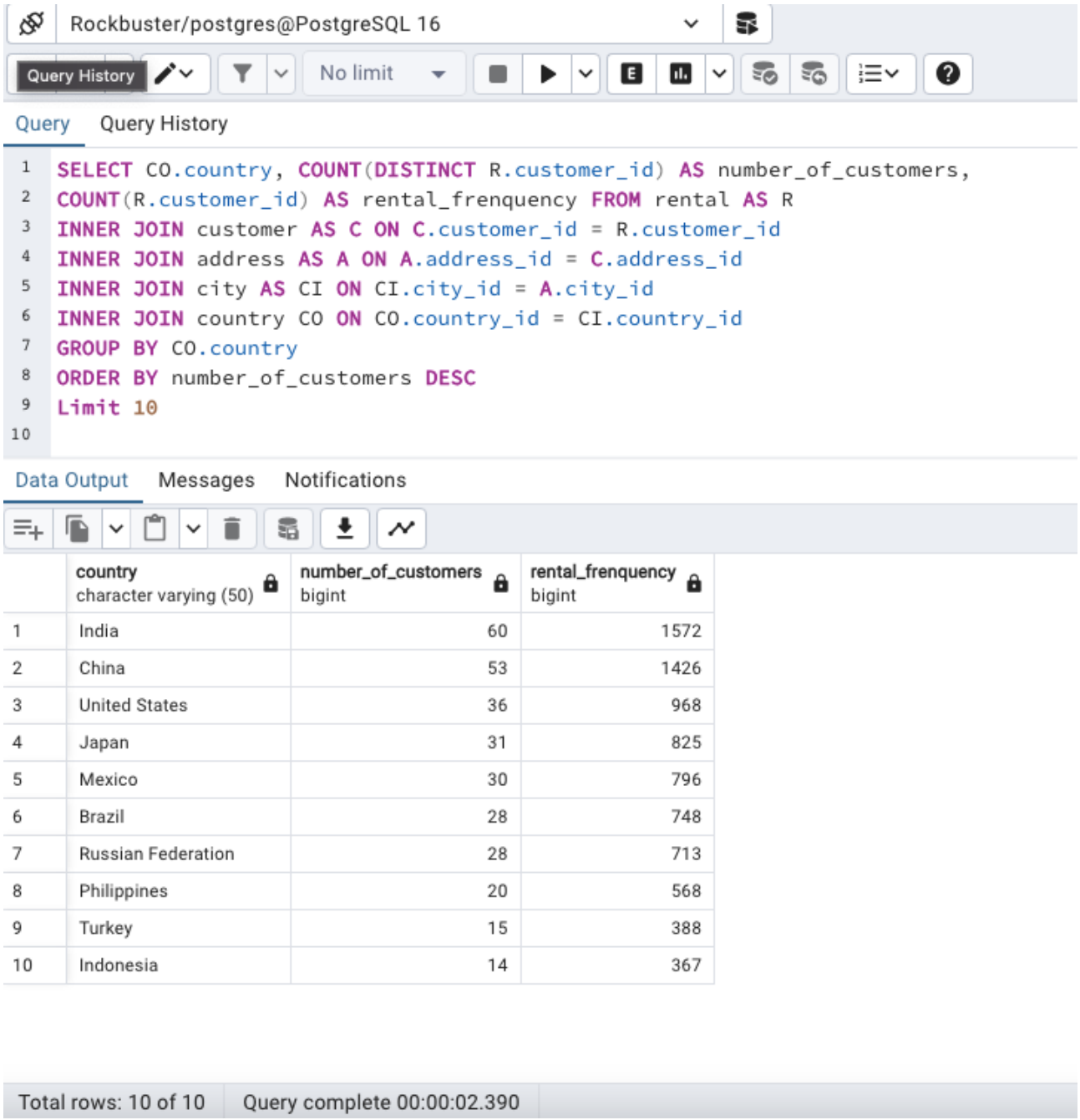
TASK 3.7

Joining Tables of Data

1:
To find the top 10 countries for Rockbuster in terms of customer numbers, I used SQL's GROUP BY and ORDER BY clauses. First, I joined the necessary tables (rental, customer, address, city, and country) to link customer information with their respective countries. Then, I grouped the results by country and counted the number of distinct customer IDs to determine the number of customers in each country. Finally, I ordered the results in descending order based on the number of customers and limited the output to the top 10 countries.

```
SELECT CO.country,
    COUNT(DISTINCT R.customer_id) AS number_of_customers
FROM rental AS R
INNER JOIN customer AS C ON C.customer_id = R.customer_id
INNER JOIN address AS A ON A.address_id = C.address_id
INNER JOIN city AS CI ON CI.city_id = A.city_id
INNER JOIN country CO ON CO.country_id = CI.country_id
GROUP BY CO.country
ORDER BY number_of_customers DESC
LIMIT 10;
```

Approaching this query, I first identified the need to link customer data with country information and count the number of customers in each country. I chose to use COUNT(DISTINCT R.customer_id) to ensure accurate customer counts. By sorting the results in descending order and limiting them to the top 10, I could easily identify the countries with the highest customer numbers, providing valuable insights for Rockbuster. This approach showcases my understanding of SQL's functionalities and my ability to efficiently retrieve the required information.



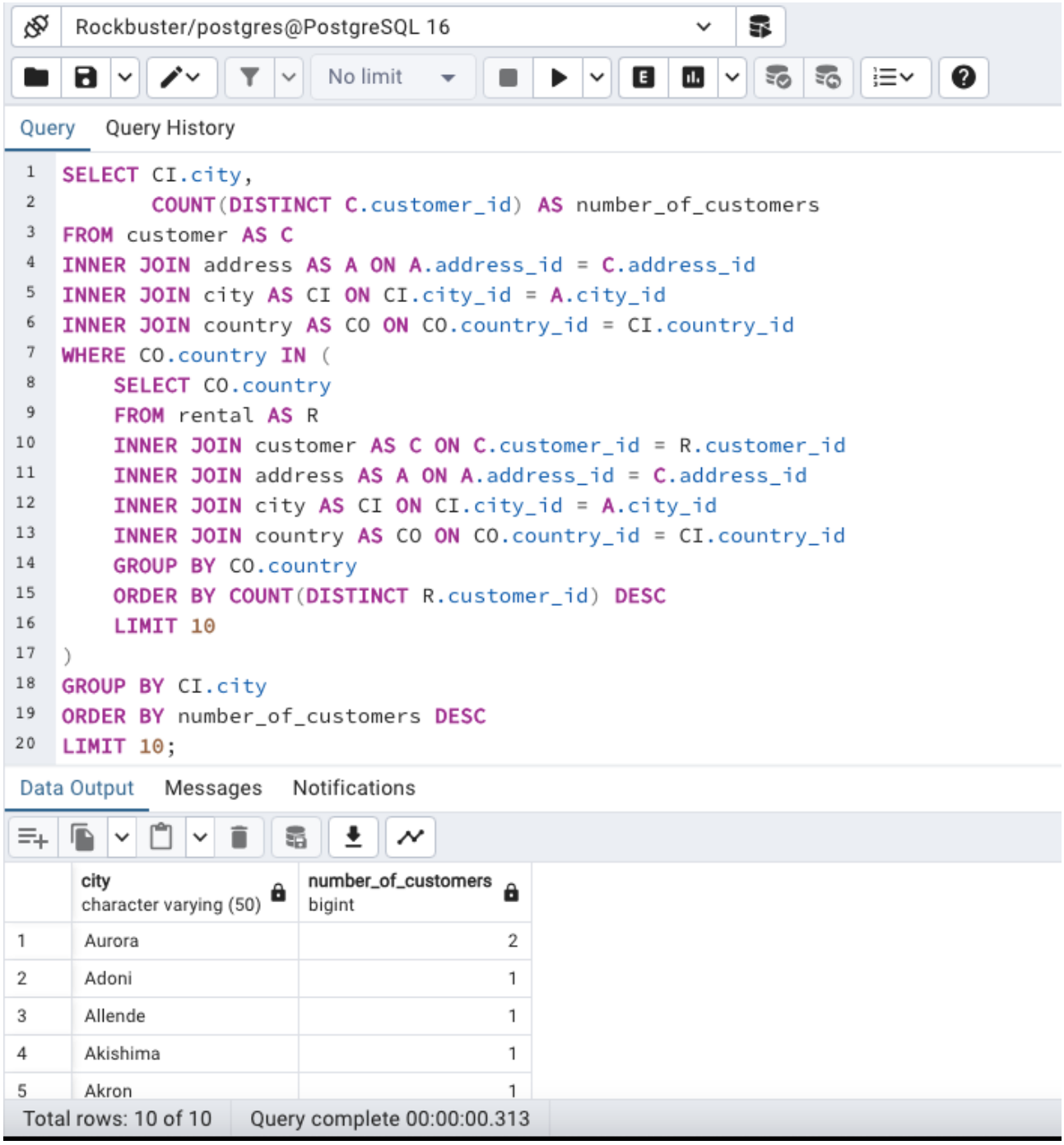
2:

To identify the top 10 cities within the top 10 countries identified in the previous step, I used a nested query to first retrieve the top 10 countries. Then, I joined the necessary tables (customer, address, city, and country) to link city information with their respective countries. Next, I grouped the results by city and counted the number of distinct customer IDs to determine the number of customers in each city. Finally, I ordered the results in descending order based on the number of customers and limited the output to the top 10 cities.

```
SELECT CI.city,
    COUNT(DISTINCT C.customer_id) AS number_of_customers
FROM customer AS C
```

```
INNER JOIN address AS A ON A.address_id = C.address_id
INNER JOIN city AS CI ON CI.city_id = A.city_id
INNER JOIN country AS CO ON CO.country_id = CI.country_id
WHERE CO.country IN (
   SELECT CO.country
   FROM rental AS R
   INNER JOIN customer AS C ON C.customer_id = R.customer_id
   INNER JOIN address AS A ON A.address_id = C.address_id
   INNER JOIN city AS CI ON CI.city_id = A.city_id
   INNER JOIN country AS CO ON CO.country_id = CI.country_id
   GROUP BY CO.country
   ORDER BY COUNT(DISTINCT R.customer_id) DESC
   LIMIT 10
)
GROUP BY CI.city
ORDER BY number_of_customers DESC
LIMIT 10;
```

Approaching this query, I first identified the need to find the top 10 countries from the previous step. I used a subquery to retrieve these countries. Then, I linked city data with customer information and filtered the cities based on the top 10 countries identified. By counting the number of customers in each city and sorting the results accordingly, I could easily identify the top 10 cities with the highest customer numbers across the top 10 countries. This approach efficiently provides insights into the most populous cities within the identified countries.



3:

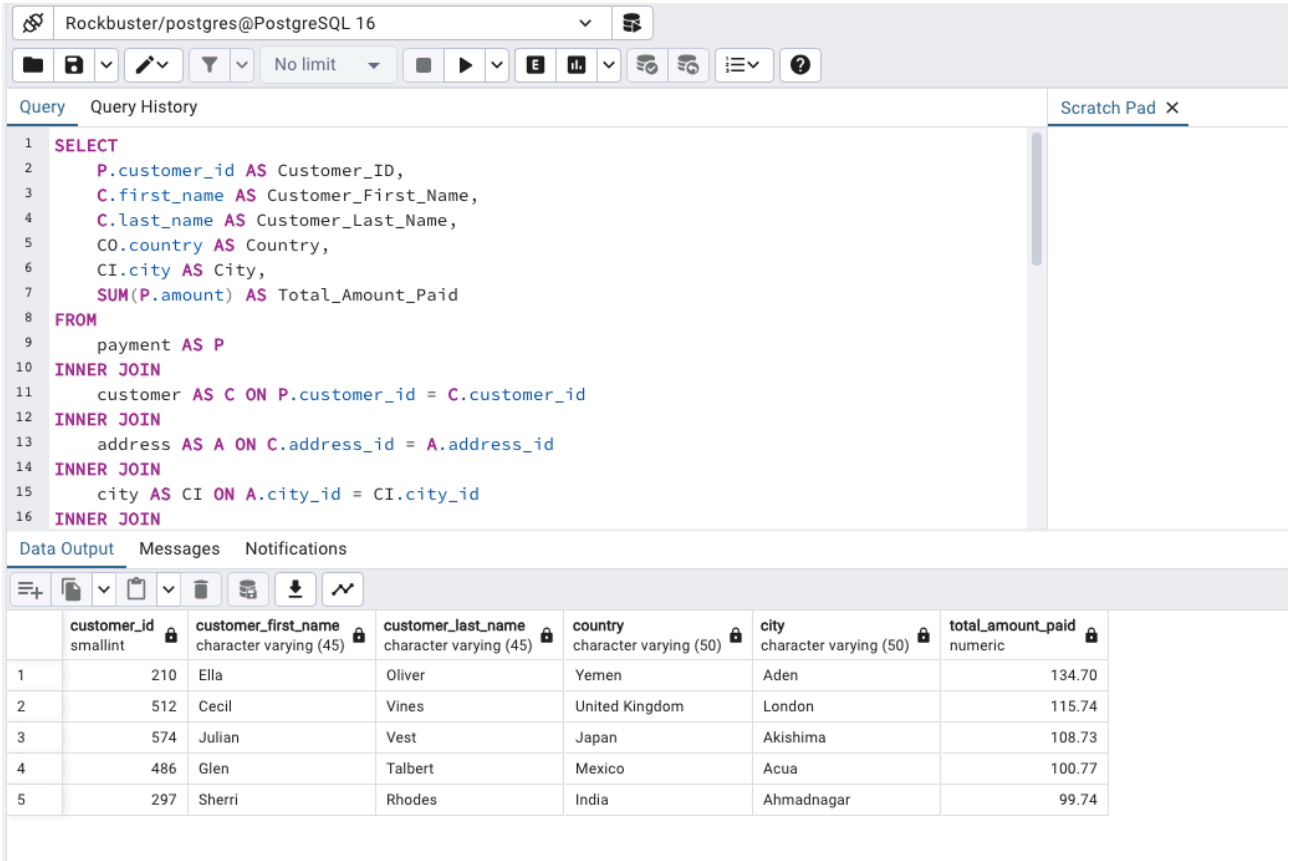
To find the top 5 customers from the top 10 cities who have paid the highest total amounts to Rockbuster, I employed a nested query to first identify the top 10 cities. Then, I joined the necessary tables (customer, address, city, country, and payment) to link customer payment information with their respective cities and countries. Next, I grouped the results by customer and summed the total amount paid by each customer. Finally, I ordered the results in descending order based on the total amount paid and limited the output to the top 5 customers.

SELECT

P.customer_id AS Customer_ID,

```
    C.first_name AS Customer_First_Name,
    C.last_name AS Customer_Last_Name,
    CO.country AS Country,
    CI.city AS City,
    SUM(P.amount) AS Total_Amount_Paid
FROM
    payment AS P
INNER JOIN
    customer AS C ON P.customer_id = C.customer_id
INNER JOIN
    address AS A ON C.address_id = A.address_id
INNER JOIN
    city AS CI ON A.city_id = CI.city_id
INNER JOIN
    country AS CO ON CI.country_id = CO.country_id
WHERE
    CI.city IN (
        SELECT
            CI.city
        FROM
            customer AS C
        INNER JOIN
            address AS A ON C.address_id = A.address_id
        INNER JOIN
            city AS CI ON A.city_id = CI.city_id
        INNER JOIN
            country AS CO ON CI.country_id = CO.country_id
        GROUP BY
            CI.city
        ORDER BY
            COUNT(DISTINCT C.customer_id) DESC
        LIMIT 10
    )
GROUP BY
    P.customer_id,
    C.first_name,
    C.last_name,
    CO.country,
    CI.city
ORDER BY
    Total_Amount_Paid DESC
LIMIT 5;
```

Approaching this query, I first identified the need to find the top 10 cities from the previous step. I used a subquery to retrieve these cities. Then, I linked payment data with customer information, filtering the customers based on the top 10 cities identified. By summing the total amount paid by each customer and sorting the results accordingly, I could easily identify the top 5 customers who have paid the highest total amounts to Rockbuster. This approach efficiently identifies loyal customers deserving of recognition and rewards.



	customer_id smallint	customer_first_name character varying (45)	customer_last_name character varying (45)	country character varying (50)	city character varying (50)	total_amount_paid numeric
1	210	Ella	Oliver	Yemen	Aden	134.70
2	512	Cecil	Vines	United Kingdom	London	115.74
3	574	Julian	Vest	Japan	Akishima	108.73
4	486	Glen	Talbert	Mexico	Acua	100.77
5	297	Sherri	Rhodes	India	Ahmadnagar	99.74