

### 3.7 Joining Tables of Data

1. Write a query to find the top 10 countries for Rockbuster in terms of customer numbers. (Tip: you'll have to use **GROUP BY** and **ORDER BY**, both of which follow the join.)
  - Copy-paste your query and its output into your answers document.
  - Write a few sentences on how you approached this query and why. It's important that you can explain your thought process when writing queries, especially for future interviews.

Query

Query History

1

SELECT D.country AS customer\_country,

2

COUNT (A.customer\_id) AS number\_of\_customer

3

FROM customer A

4

INNER JOIN address B ON B.address\_id = A.address\_id

5

INNER JOIN city C ON B.city\_id = C.city\_id

6

INNER JOIN country D ON C.country\_id = D.country\_id

7

GROUP BY country

8

ORDER BY number\_of\_customer DESC

9

LIMIT 10 |

Data output

Messages

Notifications

≡+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	customer_country character varying (50) 🔒	number_of_customer 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 00:00:00.094

1. Firstly, I identify what final information I want to get from the query.
2. From the ERD, I identify which table could provide me that final information from step 1.
3. Once done, it starts with the question on how? Do I only need one table or multiple tables?

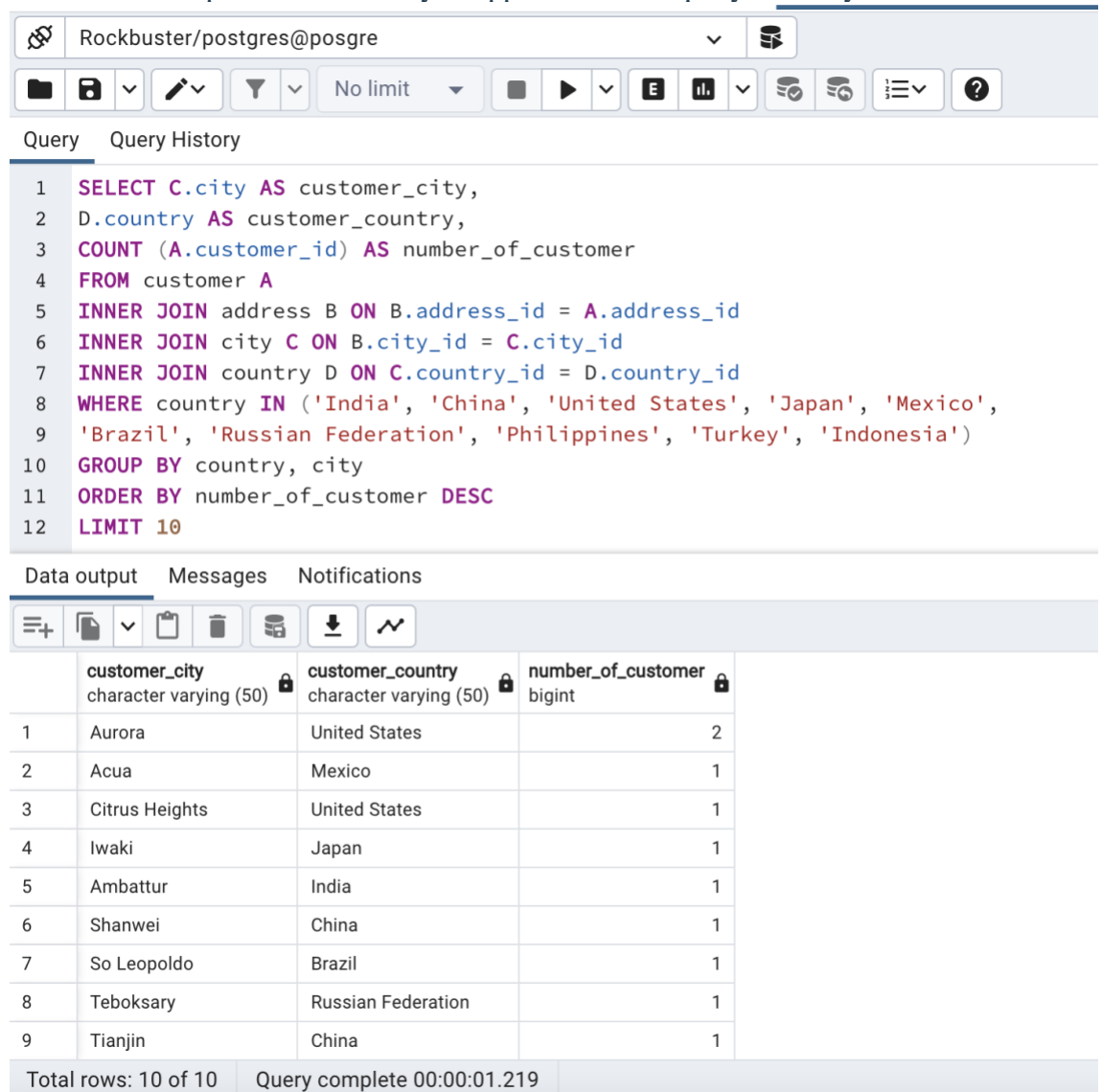
In case multiple tables:

- based on step 1 and 2, identify the joins (INNER JOIN, LEFT/RIGHT JOIN, FULL JOIN)
- identify the path to the desired table
- list the required table with their common keys.

4. Write the query from step 1

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

- o Copy-paste your query and its output into your answers document.
- o Write a short explanation of how you approached this query and why.



Rockbuster/postgres@posgre

Query Query History

```
1 SELECT C.city AS customer_city,
2 D.country AS customer_country,
3 COUNT (A.customer_id) AS number_of_customer
4 FROM customer A
5 INNER JOIN address B ON B.address_id = A.address_id
6 INNER JOIN city C ON B.city_id = C.city_id
7 INNER JOIN country D ON C.country_id = D.country_id
8 WHERE country IN ('India', 'China', 'United States', 'Japan', 'Mexico',
9 'Brazil', 'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
10 GROUP BY country, city
11 ORDER BY number_of_customer DESC
12 LIMIT 10
```

Data output Messages Notifications

	customer_city character varying (50)	customer_country character varying (50)	number_of_customer bigint
1	Aurora	United States	2
2	Acua	Mexico	1
3	Citrus Heights	United States	1
4	Iwaki	Japan	1
5	Ambattur	India	1
6	Shanwei	China	1
7	So Leopoldo	Brazil	1
8	Teboksary	Russian Federation	1
9	Tianjin	China	1

Total rows: 10 of 10 Query complete 00:00:01.219

Top 10 cities within the top 10 countries identified in step 1” is indeed a continuation of the query done in number 1. Here, I only need to go one layer deeper, which is: city from those country. Thus, WHERE is used.

1. 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!
  - Tip: After the join syntax, you'll 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.
  - Copy-paste your query and its output into your answers document.

The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT A.customer_id,
2 A.first_name,
3 A.last_name,
4 C.city,
5 D.country,
6 SUM(E.amount) AS payment
7 FROM customer A
8 INNER JOIN address B ON A.address_id = B.address_id
9 INNER JOIN city C ON B.city_id = C.city_id
10 INNER JOIN country D ON C.country_id = D.country_id
11 INNER JOIN payment E ON A.customer_id = E.customer_id
12 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
13 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
14 GROUP BY A.customer_id, first_name, last_name, city, country
15 ORDER BY payment DESC
16 LIMIT 5;
```

The query results are displayed in a table with the following columns: customer\_id, first\_name, last\_name, city, country, and payment. The results are ordered by payment in descending order, showing the top 5 customers.

	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	payment numeric
1	84	Sara	Perry	Atlixco	Mexico	128.7
2	518	Gabriel	Harder	Sivas	Turkey	108.75
3	587	Sergio	Stanfield	Celaya	Mexico	102.76
4	537	Clinton	Buford	Aurora	United States	98.76
5	367	Adam	Gooch	Adoni	India	97.8