

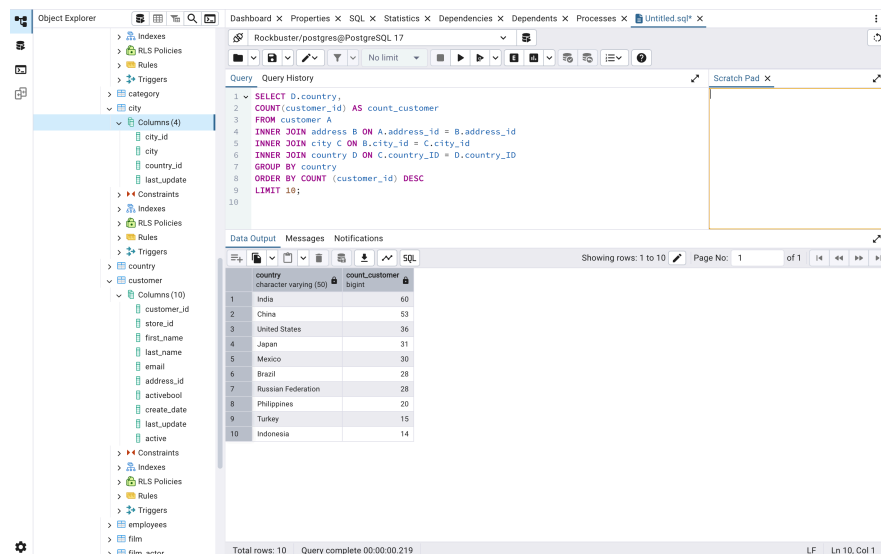
Join Tables for Data

Step 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.)

Query

```
SELECT D.country,  
COUNT(customer_id) AS count_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  
GROUP BY country  
ORDER BY COUNT (customer_id) DESC  
LIMIT 10;
```

Output



The screenshot shows a PostgreSQL query editor with the following SQL query:

```
1 SELECT D.country,  
2 COUNT(customer_id) AS count_customer  
3 FROM customer A  
4 INNER JOIN address B ON A.address_id = B.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 COUNT (customer_id) DESC  
9 LIMIT 10;
```

The output shows the top 10 countries by customer count:

country	count_customer
India	60
China	53
United States	36
Japan	31
Mexico	30
Brazil	28
Russian Federation	28
Philippines	20
Turkey	15
Indonesia	14

How and why approaching this query thinking of future interviews.

I selected first the columns I want to see and rename the count of customer_id. I begin the connection between the tables that are involved starting for costumer table (denominated table A). As the key for table A and B those tables are address_id, I use the INNER JOIN between those matching columns. And so on with B on C and Con D, until I arrive to country ID that is the last key between the last tables to join to get the country column.

Then I group them by country because that's the information I need and I order them in descent, just the top 10.

Step 2. Next, write a query to identify the top 10 cities that fall within the top 10 countries you identified in step 1. (Hint: the top 10 cities can be in any of the countries identified—you don't need to create a separate list for each country.)

Query

```
SELECT C.city, D.country,
COUNT(customer_id) AS count_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
WHERE D.country IN (SELECT D.country
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 D.country
ORDER BY COUNT(A.customer_id) DESC
LIMIT 10)
GROUP BY D.country,C.city
ORDER BY COUNT (customer_id) DESC
LIMIT 10;
```

The screenshot shows a database management tool interface. On the left is the 'Object Explorer' pane showing a tree view of database objects including 'city' and 'country'. The main area displays a SQL query in a text editor, which is the same query as shown in the previous block. Below the query editor is the 'Data Output' pane, which shows the results of the query in a table format. The table has three columns: 'city', 'country', and 'count_customer'. The results show the top 10 cities based on the number of customers, grouped by country.

city	country	count_customer
Aurora	United States	2
Acua	Mexico	1
Citrus Heights	United States	1
Iwaki	Japan	1
Ambattur	India	1
Shanwei	China	1
So Leopoldo	Brazil	1
Teboksary	Russian Federation	1
Tianjin	China	1
Cianjur	Indonesia	1

Total rows: 10 Query complete 00:00:00.072

Output

Output 3.7 S2

city	country	customer_count
Aurora	United States	2
Atlixco	Mexico	1
Xintai	China	1
Adoni	India	1
Dhule (Dhulia)	India	1
Kurashiki	Japan	1
Pingxiang	China	1
Sivas	Turkey	1
Celaya	Mexico	1
So Leopoldo	Brazil	1

How and why approaching this query thinking of future interviews:

I have selected first the columns I need to see for my output, and made everything match to the first query, due the need of having the top 10 cities from the first output. So, to get that, I opened the clause, for the country column, WHERE and IN to filter and include the cities within the countries that I had in the first place, making the same INNER JOINS, GROUP and ORDER BY D.country, DESC, LIMIT 10. Closing the parenthesis of what I want to see in the country column to finish with the big picture and structure of the query: GROUP and ORDER BY C.city and D.country and again LIMIT 10 in DESC order.

Step 3. Now write a query to find the top 5 customers from the top 10 cities who've 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, and Total Amount Paid.

Query

```
SELECT B.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_payment
FROM payment A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.country_id
WHERE D.city IN
      (SELECT D.city
       FROM customer B
       INNER JOIN address C ON B.address_id = C.address_id
       INNER JOIN city D ON C.city_id = D.city_id
       INNER JOIN country E ON D.country_id = E.country_id
       WHERE E.country IN
            (SELECT E.country
             FROM customer B
             INNER JOIN address C ON B.address_id = C.address_id
             INNER JOIN city D ON C.city_id = D.city_id
             INNER JOIN country E ON D.country_id = E.country_id
             GROUP BY E.country
             ORDER BY COUNT(B.customer_id) DESC
             LIMIT 10)
      )
GROUP BY
       E.country,
       D.city
ORDER BY COUNT(B.customer_id) DESC
LIMIT 10)

GROUP BY E.country, D.city, B.customer_id, B.first_name, B.last_name
ORDER BY SUM(A.amount) DESC
LIMIT 5;
```

Output

The screenshot shows the DBeaver application window. On the left is the Object Explorer tree with the following structure:

- Database: postgres
- Schemas: public
 - Tables: postal_code, phone, last_update
 - Views: city
 - Constraints: (4)
 - Indexes: (10)
 - RLS Policies: (1)
 - Rules: (1)
 - Triggers: (1)

The main editor displays a SQL query titled "Rockbuster/postgres@PostgreSQL 17". The query is as follows:

```
SELECT B.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_payment
FROM   payment A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.country_id
WHERE  D.city IN ('Aurora', 'Bufoord', 'Talbert', 'Welch', 'Spurlock', 'Shanwei', 'Iwaki', 'Acua', 'Marlene', 'Glen', 'Harvey', 'Clinton')

(SELECT D.city
 FROM customer B
 INNER JOIN address C ON B.address_id = C.address_id
 INNER JOIN city D ON C.city_id = D.city_id
 INNER JOIN country E ON D.country_ID = E.country_ID
 WHERE E.country IN ('Canada'))
```

Below the query editor, the Data Output tab shows the results of the query. It indicates "Showing rows: 1 to 5" and "Page No: 1 of 1". The results are displayed in a table with 7 columns:

	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	225	Arlene	Harvey	Ambattur	India	111.76
2	424	Kyle	Spurlock	Shanwei	China	109.71
3	240	Marlene	Welch	Iwaki	Japan	106.77
4	486	Glen	Talbert	Acua	Mexico	100.77
5	537	Clinton	Bufoord	Aurora	United States	98.76

At the bottom of the window, it states "Total rows: 5" and "Query complete 00:00:00.098". The status bar at the very bottom shows "LF Ln 16, Col 11".

Output 3.7.S.3

customer_id	first_name	last_name	city	country	total_payment
225	Arlene	Harvey	Ambattur	India	111.76
424	Kyle	Spurlock	Shanwei	China	109.71
240	Marlene	Welch	Iwaki	Japan	106.77
486	Glen	Talbert	Acua	Mexico	100.77
537	Clinton	Buford	Aurora	United States	98.76