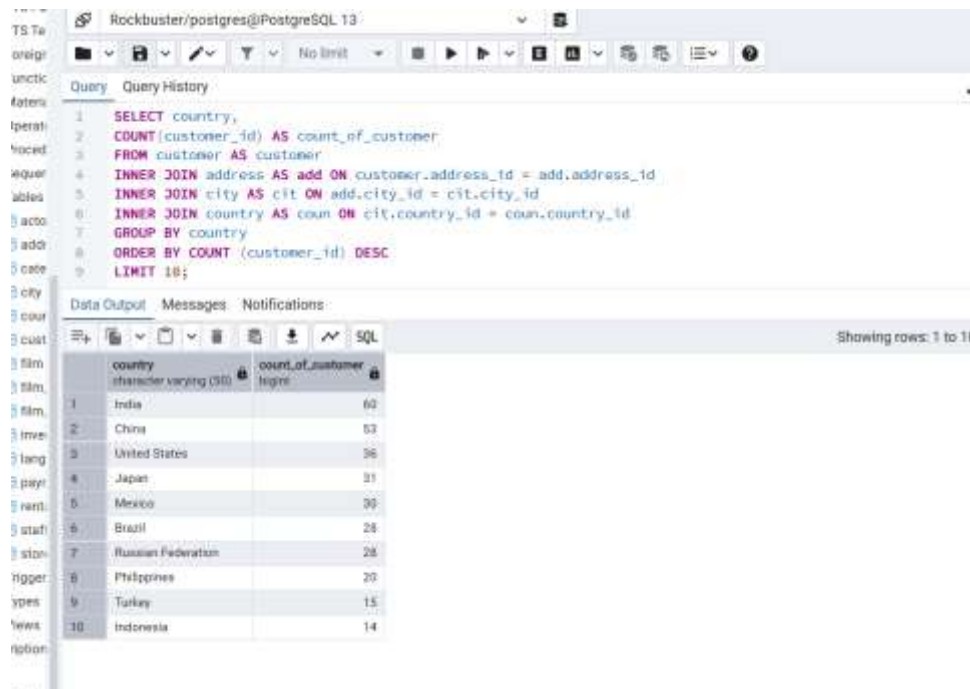


Ex 3.7

Prepared by: Nikita Kasrekar

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.)



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

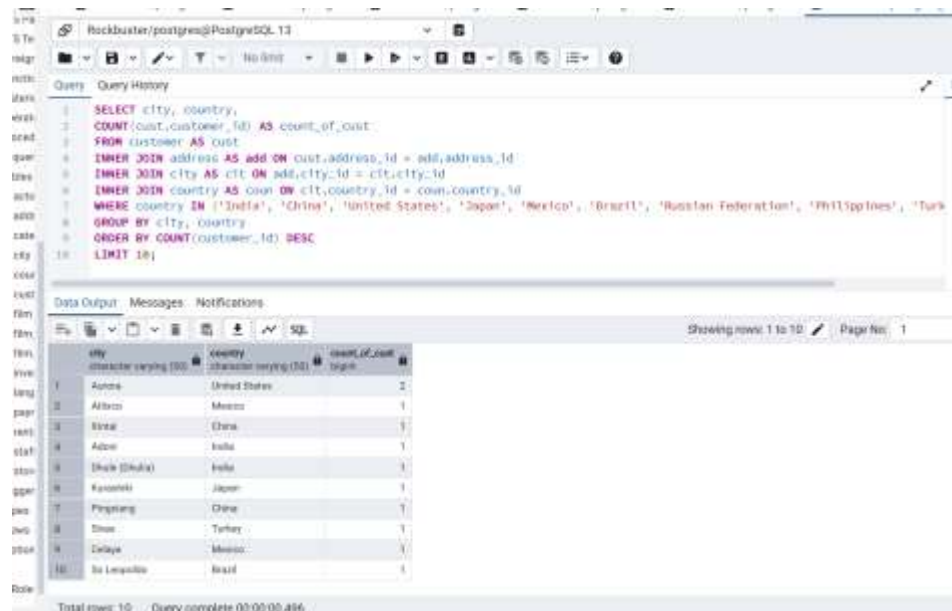
```
1 SELECT country,
2 COUNT(customer_id) AS count_of_customer
3 FROM customer AS customer
4 INNER JOIN address AS add ON customer.address_id = add.address_id
5 INNER JOIN city AS cit ON add.city_id = cit.city_id
6 INNER JOIN country AS coun ON cit.country_id = coun.country_id
7 GROUP BY country
8 ORDER BY COUNT (customer_id) DESC
9 LIMIT 10;
```

The query results are displayed in a table with the following data:

	country	count_of_customer
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

To identify the top 10 countries by customer numbers, I followed the relational path between tables: customer → address → city → country. I used INNER JOINs to connect these tables because customer locations are stored indirectly through address and city. After linking the correct tables, I grouped the results by country to count how many customers belong to each country. Finally, I ordered the results from highest to lowest and limited the output to the top 10. This approach ensures that I accurately capture customer distribution across countries while keeping the query efficient and easy to interpret.

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.)



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

```
SELECT city, country,
COUNT(customer_id) AS count_of_cust
FROM customer AS cust
INNER JOIN address AS add ON cust.address_id = add.address_id
INNER JOIN city AS cit ON add.city_id = cit.city_id
INNER JOIN country AS coun ON cit.country_id = coun.country_id
WHERE country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russian Federation', 'Philippines', 'Turkey')
GROUP BY city, country
ORDER BY COUNT(customer_id) DESC
LIMIT 10;
```

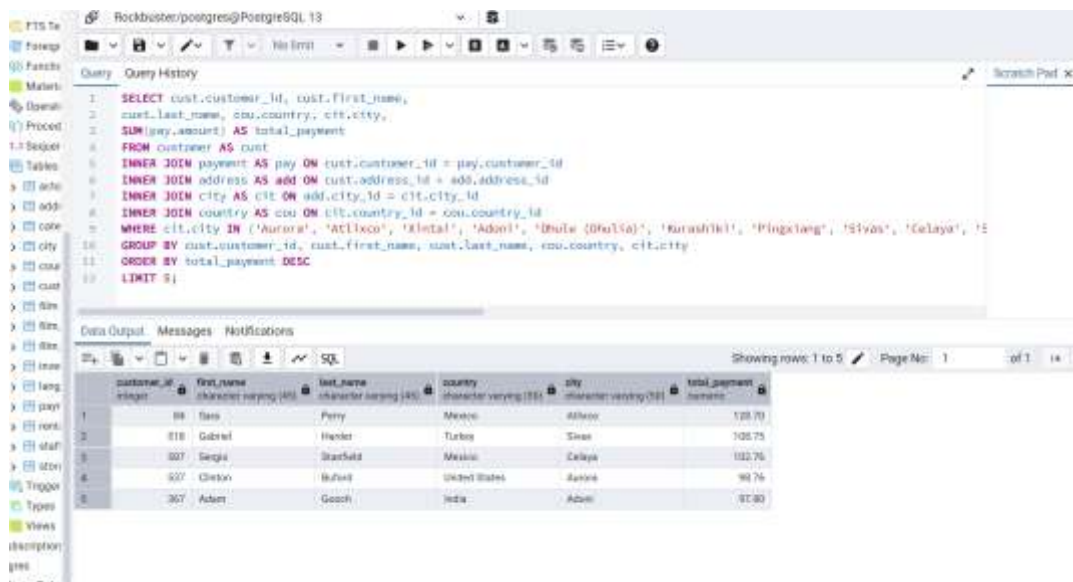
The results are displayed in a table with the following columns: city, country, and count_of_cust. The table shows the top 10 cities across the specified countries, ordered by the number of customers in descending order.

city	country	count_of_cust
Ahmedabad	India	1
Almaty	Kazakhstan	1
Beijing	China	1
Bombay	India	1
Buenos Aires	Argentina	1
Calcutta	India	1
Chennai	India	1
Delhi	India	1
London	United Kingdom	1
Mumbai	India	1

Total rows: 10 / Query complete 00:00:00.416

I already knew the top 10 countries from the previous query, so I used them directly in the WHERE clause to filter the cities. I then joined the customer, address, city, and country tables to connect each customer to their city and country. After filtering to the top countries, I grouped the results by city and country and counted the number of customers in each city. Finally, I ordered the results in descending order to find the top 10 cities overall. This approach is straightforward and avoids the need for a separate subquery.

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!



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

```
1 SELECT cust.customer_id, cust.first_name,
2        cust.last_name, cou.country, cit.city,
3        SUM(pay.amount) AS total_payment
4 FROM customer AS cust
5 INNER JOIN payment AS pay ON cust.customer_id = pay.customer_id
6 INNER JOIN address AS add ON cust.address_id = add.address_id
7 INNER JOIN city AS cit ON add.city_id = cit.city_id
8 INNER JOIN country AS cou ON cit.country_id = cou.country_id
9 WHERE cit.city IN ('Aurora', 'Atlixco', 'Xtetal', 'Adoni', 'Dhuie (Dha'isa)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Zelaya', 'S')
10 GROUP BY cust.customer_id, cust.first_name, cust.last_name, cou.country, cit.city
11 ORDER BY total_payment DESC
12 LIMIT 5;
```

The results of the query are displayed in a table with the following columns: customer_id, first_name, last_name, country, city, and total_payment. The table shows the top 5 customers with the highest total payments in the top 10 cities.

customer_id	first_name	last_name	country	city	total_payment
1	Ben	Perry	Mexico	Atlixco	128.70
2	Gabriel	Hender	Turkey	Sivas	106.75
3	George	Stanfield	Mexico	Zelaya	102.75
4	Clinton	Stewart	United States	Adoni	98.76
5	Adam	Good	India	Adoni	87.90

To find the top 5 customers with the highest total payments in the top 10 cities, I first used the list of top cities identified in the previous query. I joined the customer, payment, address, city, and country tables so that each customer could be linked to their city and country. Then I filtered the results to include only customers living in the top 10 cities and top 10 countries. I grouped the data by customer and summed up their total payments. Finally, I sorted the totals in descending order and selected the top 5 customers. This approach ensures that only the most valuable customers within the most important cities are included.