

## Kareemah Ashiru

### Answers 3.7

1. Write a query to find the top 10 countries for Rockbuster in terms of customer numbers.

Dashboard × Processes × Rockbuster/postgres@PostgreSQL 16\* ×

Rockbuster/postgres@PostgreSQL 16

Query Query History

```
1 SELECT co.country AS country_name, COUNT(c.customer_id) AS customer_count
2 FROM Customer c
3 JOIN Address a ON c.address_id = a.address_id
4 JOIN City ci ON a.city_id = ci.city_id
5 JOIN Country co ON ci.country_id = co.country_id
6 GROUP BY co.country
7 ORDER BY customer_count DESC
8 LIMIT 10;
```

Data Output Messages Notifications

	country_name character varying (50)	customer_count 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

a.

b. **Explanation**

- i. I used the Schema sheet to help me identify the tables, their contents and what table linked with the others. I was able to identify the multiple tables needed to produce the country list and the customer count: Customer - Address - City - Country. I used the count (c.customer\_id) as a sum to

rank the top countries based on the total number of customers. I grouped the query by country, ordered the customer count in a descending order and limited it to just 10 countries as instructed in this exercise. I chose INNER JOIN because what is needed is a limited number of countries (10) that can be found in both of the tables.

- ii. **JOIN Address a ON c.address\_id = a.address\_id:** Joins the **Customer** table with the **Address** table.
- iii. **JOIN City ci ON a.city\_id = ci.city\_id:** Joins the **Address** table with the **City** table.
- iv. **JOIN Country co ON ci.country\_id = co.country\_id:** Joins the **City** table with the **Country** table.
- v. **GROUP BY co.country:** Groups the results by country name.
- vi. **COUNT(c.customer\_id) AS customer\_count:** Counts the number of customers in each country.
- vii. **ORDER BY customer\_count DESC:** Sorts the results by the customer count in descending order.
- viii. **LIMIT 10:** Limits the output to the top 10 countries.

c.

2. **Write a query to identify the top 10 cities that fall within the top 10 countries you identified in step 1.**

Dashboard × Processes × Rockbuster/postgres@PostgreSQL 16\* ×

Rockbuster/postgres@PostgreSQL 16

Query Query History

```

1 WITH TopCountries AS (
2     SELECT co.country_id, co.country AS country_name,
3     COUNT (customer_id) AS number_of_customers
4     FROM Customer c
5     INNER JOIN Address a ON c.address_id = a.address_id
6     INNER JOIN City ci ON a.city_id = ci.city_id
7     INNER JOIN Country co ON ci.country_id = co.country_id
8     GROUP BY co.country_id, co.country
9     ORDER BY COUNT(c.customer_id) DESC
10    LIMIT 10
11 )
12
13 SELECT ci.city AS city_name, co.country AS country_name,
14 FROM Customer c
15 INNER JOIN Address a ON c.address_id = a.address_id
16 INNER JOIN City ci ON a.city_id = ci.city_id
17 INNER JOIN Country co ON ci.country_id = co.country_id
18 WHERE co.country_id IN (SELECT country_id FROM TopCountries)
19 GROUP BY ci.city, co.country
20 ORDER BY customer_count DESC
21 LIMIT 10;
22
23
24

```

Data Output Messages Notifications

	city_name character varying (50)	country_name character varying (50)	customer_count bigint
1	Aurora	United States	2
2	Atlixco	Mexico	1
3	Xintai	China	1
4	Adoni	India	1
5	Dhule (Dhulia)	India	1
6	Kurashiki	Japan	1
7	Pingxiang	China	1
8	Sivas	Turkey	1
9	Celaya	Mexico	1
10	So Leopoldo	Brazil	1

a.

b. Explanation

- i. I used **WITH TopCountries AS (...)**, a Common Table Expression that retrieves the top 10 countries based on customer count, similar to what we did in the first query. Then included the customer count and grouped it by country for one part of the column.
- ii. For the other part of the column I wanted to include the city and the associated country so I used **co.country\_id IN (SELECT country\_id FROM TopCountries)** that filters the results to include only customers from the top 10 countries.
- iii. **GROUP BY ci.city**: Groups the results by city name.

- iv. **COUNT(c.customer\_id) AS customer\_count**: Counts the number of customers in each city.
- v. **ORDER BY customer\_count DESC**: Sorts the cities by customer count in descending order.
- vi. **LIMIT 10**: Limits the results to the top 10 cities.

The Inner Join is used here because of the desired result of this task which is to provide a limited number of information that can be found in both tables. This query will give you the top 10 cities, regardless of which top 10 country they are in, based on customer count.

### 3. 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!

Dashboard X Processes X Rockbuster/postgres@PostgreSQL 16 X

Rockbuster/postgres@PostgreSQL 16

No limit

Query Query History

```

1 SELECT
2   cu.customer_id,
3   cu.first_name,
4   cu.last_name,
5   co.country,
6   ci.city,
7   SUM(pay.amount) AS total_amount_paid
8 FROM payment AS pay
9 INNER JOIN customer AS cu ON pay.customer_id = cu.customer_id
10 INNER JOIN address AS ad ON cu.address_id = ad.address_id
11 INNER JOIN city AS ci ON ad.city_id = ci.city_id
12 INNER JOIN country AS co ON ci.country_id = co.country_id
13 WHERE co.country IN(
14
15 SELECT
16   co.country
17 FROM customer AS cu
18 INNER JOIN address AS ad ON cu.address_id = ad.address_id
19 INNER JOIN city AS ci ON ad.city_id = ci.city_id
20 INNER JOIN country AS co ON ci.country_id = co.country_id
21 GROUP BY co.country
22 ORDER BY COUNT(cu.customer_id) DESC
23 LIMIT 10
24 )
25 AND ci.city IN(
26
27 SELECT
28   ci.city
29 FROM customer AS cu
30 INNER JOIN address AS ad ON cu.address_id = ad.address_id
31 INNER JOIN city AS ci ON ad.city_id = ci.city_id
32 INNER JOIN country AS co ON ci.country_id = co.country_id
33 WHERE co.country IN(
34
35 SELECT
36   co.country
37 FROM customer AS cu
38 INNER JOIN address AS ad ON cu.address_id = ad.address_id
39 INNER JOIN city AS ci ON ad.city_id = ci.city_id
40 INNER JOIN country AS co ON ci.country_id = co.country_id
41 GROUP BY co.country
42 ORDER BY COUNT(cu.customer_id) DESC
43 LIMIT 10
44 )
45 GROUP BY ci.city
46 ORDER BY COUNT(cu.customer_id) DESC
47 LIMIT 10
48 )
49 GROUP BY cu.customer_id, cu.first_name, cu.last_name, co.country, ci.city
50 ORDER BY total_amount_paid DESC
51 LIMIT 5

```

Data Output Messages Notifications

	customer_id integer	first_name character varying (45)	last_name character varying (45)	country character varying (50)	city character varying (50)	totalAmount_paid numeric
1	566	Casey	Mena	Turkey	Tokat	130.68
2	84	Sara	Perry	Mexico	Atlixco	128.70
3	506	Leslie	Seward	Indonesia	Pontianak	123.72
4	389	Alan	Kahn	China	Emeishan	119.75
5	537	Clinton	Buford	United States	Aurora	98.76

- a.
- b. Explanation

- i. The Inner Join is used here as explained in 1 and 2.
- ii. I wanted the columns customer\_id, first\_name, last\_name, country, city, and the sum of payment to show. So I used SELECT cu.customer\_id, cu.first\_name, cu.last\_name, co.country, ci.city, sum (pay.amount) as total\_amount\_paid.
- iii. I used the WHERE co.country IN( function as a subquery to get the top 10 countries by number of customers.
- iv. I used the AND ci.city IN( subquery to get the top 10 cities by number of customers.
- v. Because #3 question only wants the top 5 paying customers, I used LIMIT 5.