

Exercise 3.7
Katherine Lecce

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

The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is to 'Rockbuster/postgres@PostgreSQL 14'. The query editor contains the following SQL query:

```
1 SELECT
2 D.country,
3 COUNT(A.customer_id) AS number_of_customers
4 FROM
5 customer A
6 JOIN address B ON A.address_id = B.address_id
7 JOIN city C ON B.city_id = C.city_id
8 JOIN country D ON C.country_id = D.country_id
9 GROUP BY
10 D.country
11 ORDER BY
12 number_of_customers DESC
13 LIMIT
14 10
```

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table format. The table has two columns: 'country' (character varying (50)) and 'number_of_customers' (bigint). The results show the top 10 countries by customer count.

	country	number_of_customers
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

At the bottom of the interface, a status bar indicates 'Total rows: 10 of 10' and 'Query complete 00:00:00.153'.

How did you approach this query and why? You must be able to explain your thought process when writing queries, especially for future interviews.

1. We want to connect or **SELECT** the **country column** from the **country table** which is represented by 'D.'
2. Then we want to count the # of '**customer_ids**' from the **customer** table which is represented by 'A' and will change the column name to "number of customers" on our output.
3. Then by using the **FROM** function, we use the **customer table** represented as 'A'
4. Using the **JOIN** function we connect **address table** to retrieve the 'address_id' field
5. We also want to **JOIN** the **city's table** 'C.' with the **address table** with the '**city_id**' field
6. Lastly use the **JOIN** function to join the **country table** represented as 'D' with the **city table** and the **country_id** field
7. **GROUP BY** 'D.country': This groups the results by the **country column** from **the country table**.
8. Use **ORDER BY** function to sort the number of **customer column** in descending order (**DESC**)
9. **LIMIT** function (**by 10**) tells the Quarry how many rows of results you want - top 10 results.

```

1  SELECT
2  D.country,
3  C.city,
4  COUNT(A.customer_id) AS number_of_customers
5  FROM
6  customer A
7  JOIN address B ON A.address_id = B.address_id
8  JOIN city C ON B.city_id = C.city_id
9  JOIN country D ON C.country_id = D.country_id
10 WHERE
11 D.country IN (
12 SELECT
13 D.country
14 FROM
15 customer A
16 JOIN address B ON A.address_id = B.address_id
17 JOIN city C ON B.city_id = C.city_id
18 JOIN country D ON C.country_id = D.country_id
19 GROUP BY
20 D.country
21 ORDER BY
22 COUNT(A.customer_id) DESC
23 LIMIT
24 10
25 )
26 GROUP BY
27 D.country, C.city
28 ORDER BY
29 COUNT(A.customer_id) DESC
30 LIMIT
31 10

```

Total rows: 10 of 10 Query complete 00:00:00.135

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



Data Output Messages Notifications			
	country character varying (50)	city character varying (50)	number_of_customers bigint
1	United States	Aurora	2
2	Mexico	Acua	1
3	United States	Citrus Heights	1
4	Japan	Iwaki	1
5	India	Ambattur	1
6	China	Shanwei	1
7	Brazil	So Leopoldo	1
8	Russian Federation	Teboksary	1
9	China	Tianjin	1
10	Indonesia	Cianjur	1

1. We first want to use the **SELECT** Function:
 - Select the **country column** from the **country table** that aligns with 'D.'
 - Select the **city column** from the **city table** that aligns with the 'c.'
 - **COUNT** function will then count the **customer_id** values from the **customer table** aligned with 'A.' resulting in the count as **number_of_customers**.
2. **FROM** clause:
 - Then using the from clause select the **customer table** that aligns with 'A.'
 - Join the **address table** (B) with the **address_id field**.
 - Join the city table (c.) with the **address table** with the **city_id**.
 - Join the **country table** with (D), with the **city table** using the **country_id** (D.)
3. **WHERE** clause:
 - This allows us to filter the data to only include information where the **country values** are aligned with quarry D (country table)
4. **GROUP BY** clause:
 - allows the **Country column** from the **country table** and the **city column** from the **city table**.
5. **ORDER BY** clause:
 - This orders the result set by the **number_of_customer** column in descending order.
6. **LIMIT** clause:
 - This limits the number of rows returned by the query to **10**.

Write a short explanation of how you approached this query and why

I like to start by writing out what I need to retrieve from where before I start to think about how the quarry will be set up. This allows me to organize where I need to look to retrieve the data from the set since all of the columns are marked by numerical letters from where in the set I want to retrieve them and how the Query will function when trying to combined the different parts of the query in longer queries like these. I also like to stop and make sure that I don't write the whole Query all at once and check as I go along before adding more. This allows for checking mistakes along the way so it doesn't all have to be done at the end.

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!



```
1  SELECT
2  A.customer_id,
3  A.first_name,
4  A.last_name,
5  D.country,
6  C.city,
7  SUM(B.amount) AS total_amount_paid
8  FROM
9  customer A
10 JOIN
11 payment B ON A.customer_id = B.customer_id
12 JOIN
13 address E ON A.address_id = E.address_id
14 JOIN
15 city C ON E.city_id = C.city_id
16 JOIN
17 country D ON C.country_id = D.country_id
18 WHERE
19 C.city IN (
20 SELECT city
21 FROM customer A
22 JOIN address E ON A.address_id = E.address_id
23 JOIN city C ON E.city_id = C.city_id
24 JOIN country D ON C.country_id = D.country_id
25 GROUP BY C.city
26 ORDER BY COUNT(A.customer_id) DESC
27 LIMIT 10
28 )
29 GROUP BY
30 A.customer_id, A.first_name, A.last_name, D.country, C.city
```

```
30 A.customer_id, A.first_name, A.last_name, D.country, C.city
31 ORDER BY
32 total_amount_paid DESC
33 LIMIT
34 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)	total_amount_paid numeric
1	148	Eleanor	Hunt	Runion	Saint-Denis	211.55
2	144	Clara	Shaw	Belarus	Molodetno	189.60
3	566	Casey	Mena	Turkey	Tokat	130.68
4	84	Sara	Perry	Mexico	Atlixco	128.70
5	506	Leslie	Seward	Indonesia	Pontianak	123.72