Career Foundry

Data Analytics Immersion

A3.E7

Kendra Jackson

Step 1:

Tables to join: Customer (A) → Address (B) → City (C) → Country (D)

Query | Query History

```
1  SELECT D.country,
2  COUNT (customer_id) AS count_of_customers
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_of_customers DESC
9  LIMIT 10
```

Data Output | Messages | Notifications

	country character varying (50)	count_of_customers 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

Rationale:

Using the ERD document, in order to retrieve the information asked by management, you can see that in order to link a customer to a country, since there is no direct link, a multiple join must be used. In order to create the link a chain between the tables must be formed from customer to address to city then country. Management had asked for a information about which countries have the largest amount of customers. This means we need to count how many customers are in a country this is done by counting the customer_id column which I then aliased to make the results more understandable. Then, we must group by country to ensure the count of the customers is grouped into its appropriate country. Additionally, they want the top 10 countries which means we must order the results descending (from largest to smallest) and then limit the results by 10. This gives us the results they were looking for.

Step 2:

Query | Query History

```
SELECT D.country, C.city,
COUNT (A.customer_id) AS count_of_customers
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 ('India','China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russian Federation', 'Phil
GROUP BY D.country, C.city
ORDER BY count_of_customers DESC
LIMIT 10
```

Data Output | Messages | Notifications

	country character varying (50)	city character varying (50)	count_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

Rationale:

For this query, management is seeking to find the top 10 cities within the top 10 counties. As before, we will be using the same 4 tables customer, address, city, and country to form a link between the tables in order to find this information. Since the cities need to be within the top 10 counties, we can use our query from part 1 but need to add to it. We need to know the city, and country so these will appear in the select statement, and a count of customer_id will occur as we need to know the amount of customers in each place to find which is the largest. After joining the aforementioned tables, a where clause is used to distinguish which counties we want to look for these cities in. The data is then grouped by the city and country and the ordering by the count from largest to

smallest, then limited by 10. With this we reach the conclusion of our top 10 cities with the largest at the top.

Step 3:

Payment (a) → customer (b) → address (c) → city (d) → country (e)

Rockbuster/postgres@PostgreSQL 16

No limit

Query Query History

```
SELECT B.customer_id, B.first_name, B.last_name, D.city, E.country,
SUM (A.amount) AS total_amount_paid
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', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur','Shanwei', 'So Leopoldo', 'Teboksa
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
    ORDER BY total_amount_paid DESC
LIMIT 5
```

Data Output Messages Notifications

SQL

	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_amount_paid 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	Buford	Aurora	United States	98.76

Rationale:

Building on the previous queries, the payment table also needs to be added now. We need to know the customer.id, their first and last names as well as city and country so these will be in our SELECT statement. We also need to know who spends the most this means we will use the sum function on the amount column of the payment table to find this. Then we need to join all the tables by joining B to A, C to B so on and so forth. Since we are looking for the 5 highest paying customers in the top 10 cities, we can do

two things in which a WHERE statement will be needed – we can either write out a WHERE IN statement that has the top 10 cities listed or we can use the previous join statements from the prior exercise. Since join statements are expensive, it is better to write out the city results we found previously. Then we group by the columns in the select statement and order by DESC the aliased name of total_amount_paid. The top 5 customers should now appear.