










Answers 3.7

Query		Query History	
1	SELECT	D.country,	
2		COUNT(customer_id)	
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	

Data Output		Messages	Notifications
<div></div>			
	country		count
	character varying (50)		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

To reduce the cost of the query, I wanted to pull only the records that I needed, and nothing else. This is why I wanted only the country and the count of their customer IDs since this was the primary aim of the query. Since I didn't need multiple records or even whole tables, I used INNER JOIN to run this function. I used ORDER BY and DESC to get the top 10 countries in the correct order, and I used LIMIT to pull only those 10 countries. To my limited knowledge, this is the most efficient way to write this query.

Query

Query History

```
1 SELECT C.city,
2       D.country,
3       COUNT(customer_id) AS customer_count
4 FROM customer A
5 INNER JOIN address B ON A.address_id=B.address_id
6 INNER JOIN city C ON B.city_id=C.city_id
7 INNER JOIN country D ON C.country_id=D.country_id
8 WHERE D.country IN('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russian Federation', 'Phillippines', 'Turkey', 'Indonesia')
9 GROUP BY D.country, C.city
10 ORDER BY customer_count DESC
11 LIMIT 10
```

Data Output

Messages

Notifications

</

This query was more complex than the previous one, and there may be more efficient ways to perform this query. I had the most difficulty with the WHERE function and using it to include the top 10 countries. Once I had figured that part out, the rest of the query was easier. To keep it efficient I wanted to pull on the city and country values alongside the number of customers to present in the final table. I also decided to rename customer_id to customer_count to improve readability.

Query Query History

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

Data Output Messages Notifications

	total_amount_paid numeric	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)
1	111.76	225	Arlene	Harvey	Ambattur	India
2	109.71	424	Kyle	Spurlock	Shanwei	China
3	106.77	240	Marlene	Welch	Iwaki	Japan
4	100.77	486	Glen	Talbert	Acua	Mexico
5	98.76	537	Clinton	Buford	Aurora	United States

This query was the most difficult for me to figure out; perhaps by design. I knew I had to include an extra table (payment) so I needed to add another join as well. I made payment into the 'A' table so I shifted everything down to accommodate. For the WHERE function, I simply changed the country names from the previous query into the top 10 cities from the output. I made sure to GROUP BY the requested columns and limited it to the top 5 customers. I referenced previous assignments to see how other students approached this query, and I found significant heterogeneity in how students approached this query and the

output (specifically, the total amount paid by the customers). It makes me wonder if my query was run correctly or if I am missing something.