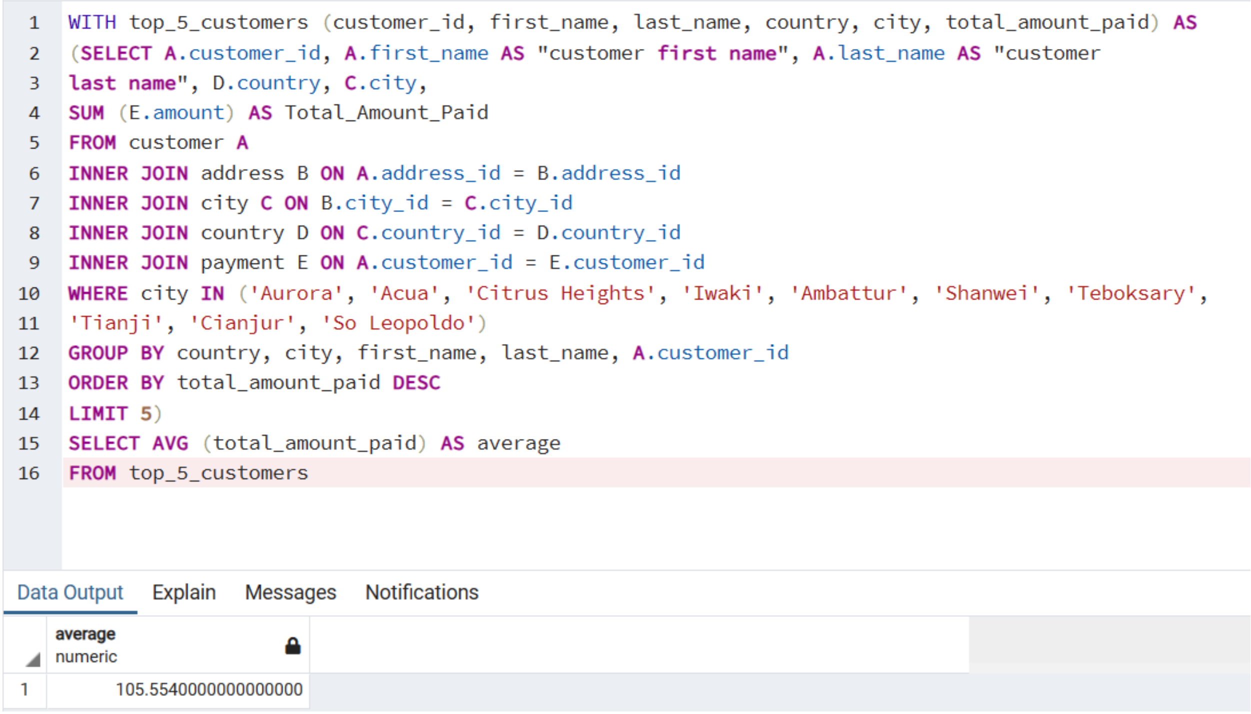
3.9: Common Table Expressions

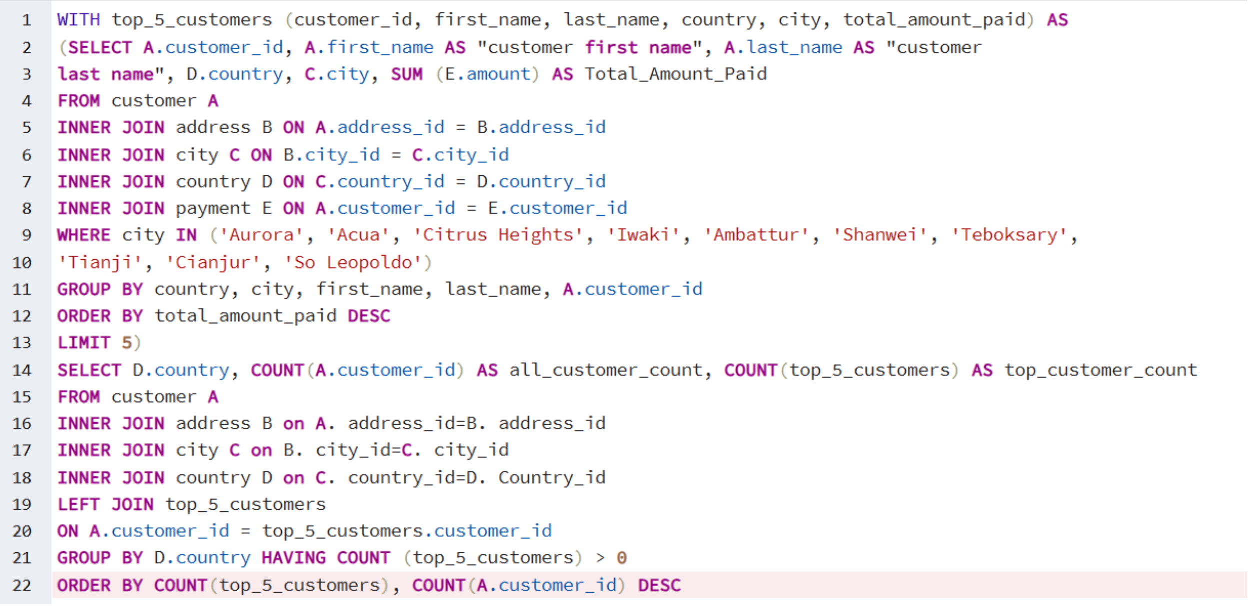
Rick Takeuchi

Question 1:

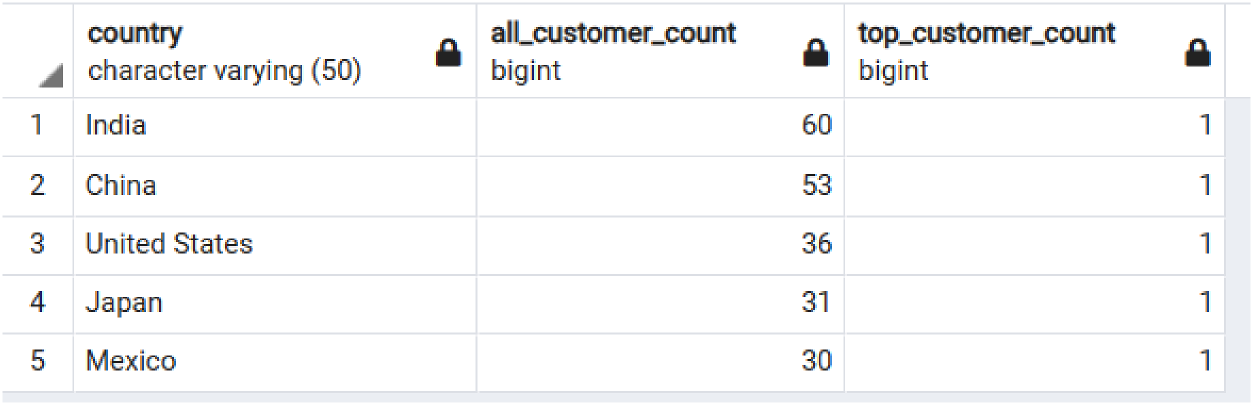


```sql
WITH top_5_customers (customer_id, first_name, last_name, country, city, total_amount_paid) AS
(SELECT A.customer_id, A.first_name AS "customer first name", A.last_name AS "customer
last name", D.country, C.city,
SUM (E.amount) AS Total_Amount_Paid
FROM customer A
INNER JOIN address B ON A.address_id = 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
INNER JOIN payment E ON A.customer_id = E.customer_id
WHERE city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'Teboksary',
'Tianji', 'Cianjur', 'So Leopoldo')
GROUP BY country, city, first_name, last_name, A.customer_id
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT AVG (total_amount_paid) AS average
FROM top_5_customers
```

	average numeric
1	105.5540000000000000



```sql
WITH top_5_customers (customer_id, first_name, last_name, country, city, total_amount_paid) AS
(SELECT A.customer_id, A.first_name AS "customer first name", A.last_name AS "customer
last name", D.country, C.city, SUM (E.amount) AS Total_Amount_Paid
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
INNER JOIN payment E ON A.customer_id = E.customer_id
WHERE city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'Teboksary',
'Tianji', 'Cianjur', 'So Leopoldo')
GROUP BY country, city, first_name, last_name, A.customer_id
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT D.country, COUNT(A.customer_id) AS all_customer_count, COUNT(top_5_customers) AS top_customer_count
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
LEFT JOIN top_5_customers
ON A.customer_id = top_5_customers.customer_id
GROUP BY D.country HAVING COUNT (top_5_customers) > 0
ORDER BY COUNT(top_5_customers), COUNT(A.customer_id) DESC
```



	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1

My methodology was referencing the two queries I wrote for steps 1 and 2 of the 3.8 exercise. I used the CTE table to reference the CTE syntax to determine columns needed for CTE. However, the rest of the query is same as I wrote main statement for both steps 1 and 2.

Question 2:

a) I feel both CTEs and subqueries will perform similarly because they provide the same output. However, they are executed in different ways.

b)

Using CTEs	Using Subqueries
Step 1 Cost: 64.49	Step 1 Cost: 64.49
Step 2 Cost: 136.93	Step 2 Cost: 136.93

c) The EXPLAIN command gives you an estimated cost. To find out the actual speed of your queries, run them in pgAdmin 4. After each query has been run, a pop-up window will display its speed in milliseconds

d) The results didn’t surprise me because CTEs and subqueries act very similarly when executed. There was no significant difference and you will only know which performs better or not through the comparison of costs with EXPLAIN commands. CTEs are more useful when it comes to accessibility and readability of the code.

Question 3:

I was extremely challenged replacing subqueries with CTEs because of the flow and structure of CTEs. You need to understand what subqueries will be replaced by CTEs. However, once you start to understand the structure, it is easier to catch on over time gradually. With more practice with SQL language, especially subqueries and CTEs, it will become easier to apply and understand over time.