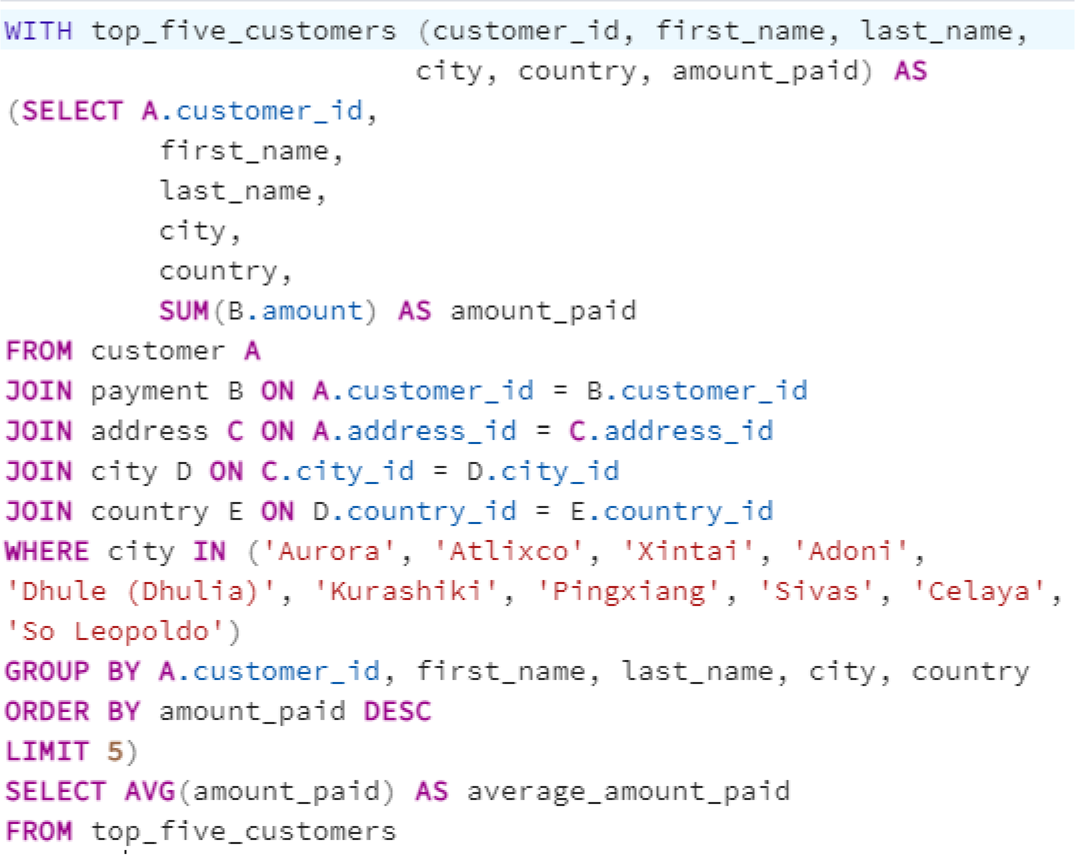
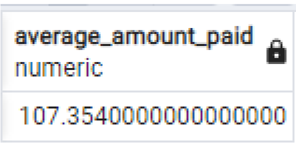
Task 3.9 – Common Table Expressions

Step 1: Rewrite your queries from steps 1 and 2 of task 3.8 as CTEs.

Find the average amount paid by the top 5 customers



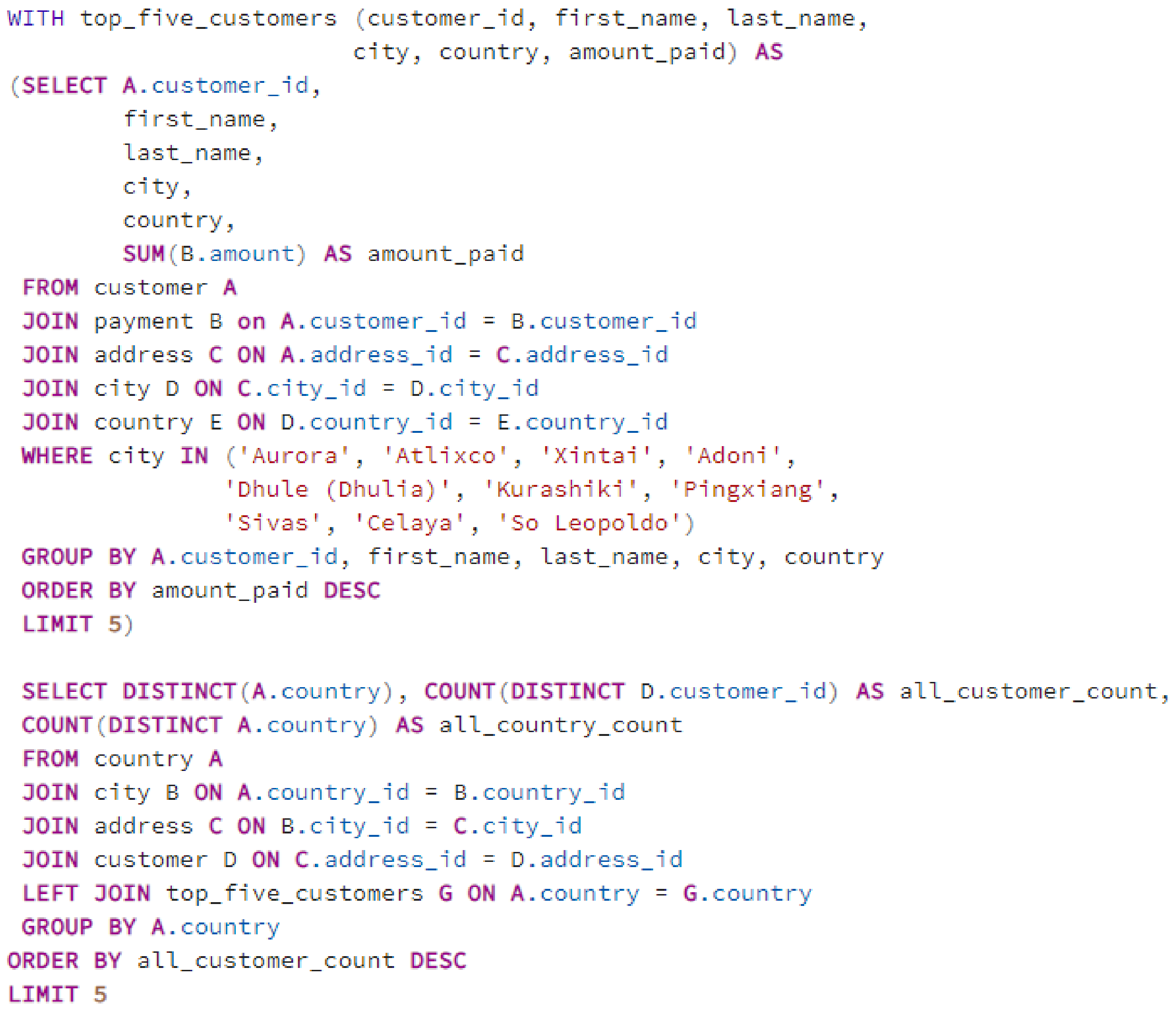
```
WITH top_five_customers (customer_id, first_name, last_name,
                          city, country, amount_paid) AS
(SELECT A.customer_id,
         first_name,
         last_name,
         city,
         country,
         SUM(B.amount) AS amount_paid
FROM customer A
JOIN payment B ON A.customer_id = B.customer_id
JOIN address C ON A.address_id = C.address_id
JOIN city D ON C.city_id = D.city_id
JOIN country E ON D.country_id = E.country_id
WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni',
'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya',
'So Leopoldo')
GROUP BY A.customer_id, first_name, last_name, city, country
ORDER BY amount_paid DESC
LIMIT 5)
SELECT AVG(amount_paid) AS average_amount_paid
FROM top_five_customers
```



average_amount_paid numeric
107.3540000000000000

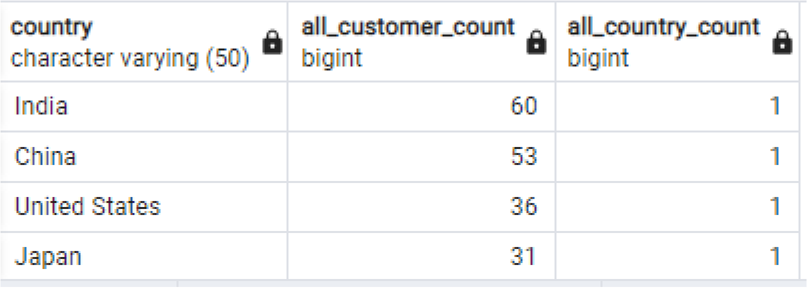
The query in the parentheses was my answer for Step 1 in Task 3.8. Based on the previous tasks, I knew that this answer produced a table displaying the top 5 customers already. With the steps involved to create a CTE, the hard part was cut out and I just had to place it within the CTE syntax. Using examples from the notes, I placed the WITH clause at the beginning, and matched the number of columns in the query below it. I then wrote the main statement to query it.

Find out how many of the top 5 customers are based within each country



```
WITH top_five_customers (customer_id, first_name, last_name,
                        city, country, amount_paid) AS
(SELECT A.customer_id,
        first_name,
        last_name,
        city,
        country,
        SUM(B.amount) AS amount_paid
 FROM customer A
 JOIN payment B on A.customer_id = B.customer_id
 JOIN address C ON A.address_id = C.address_id
 JOIN city D ON C.city_id = D.city_id
 JOIN country E ON D.country_id = E.country_id
 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni',
               'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang',
               'Sivas', 'Celaya', 'So Leopoldo')
 GROUP BY A.customer_id, first_name, last_name, city, country
 ORDER BY amount_paid DESC
 LIMIT 5)

 SELECT DISTINCT(A.country), COUNT(DISTINCT D.customer_id) AS all_customer_count,
 COUNT(DISTINCT A.country) AS all_country_count
 FROM country A
 JOIN city B ON A.country_id = B.country_id
 JOIN address C ON B.city_id = C.city_id
 JOIN customer D ON C.address_id = D.address_id
 LEFT JOIN top_five_customers G ON A.country = G.country
 GROUP BY A.country
ORDER BY all_customer_count DESC
LIMIT 5
```



country character varying (50)	all_customer_count bigint	all_country_count bigint
India	60	1
China	53	1
United States	36	1
Japan	31	1

With the same logic I had in the previous example, I already know where the top 5 customers reside. I basically took the subquery and placed it at the beginning as a CTE instead. From there, I ran my main query which LEFT JOINED the CTE to produce the same result I had in 3.8.

Step 2: Compare the performance of your CTEs and subqueries

FIND THE AVERAGE AMOUNT PAID BY THE TOP 5 CUSTOMERS	
CTE	**SUBQUERY**
Cost = 64.37 to 64.39	**Cost = 64.37 to 64.39**
Runtime = 156 ms	**Runtime = 157 ms**

FIND OUT HOW MANY OF THE TOP 5 CUSTOMERS ARE BASED WITHIN EACH COUNTRY	
CTE	**SUBQUERY**
Cost = 168.10 to 168.15	**Cost = 189.48 to 189.49**
Runtime = 178 ms	**Runtime = 177 ms**

a. Which approach do you think will perform better and why?

Though CTEs and subqueries are very similar, a CTE is more readable, and they can be reusable since it only needs to be written once. I can keep referring to it as it’s in the beginning before my main query. Therefore, I would ultimately choose the CTE in the long run.

b. Did the results surprise you?

I’m not sure if I did it correctly, but the runtime between both queries were very similar. I even ran it a couple times just to make sure, but I got results within the same time. They are running the same query, so I’m not too surprised considering my answers didn’t differ that much.

Step 3: Write 1 to 2 paragraphs on the challenges you faced when replacing subqueries with CTEs.

First off, I don’t know if I did it correctly, as my answers are just rearranging the format of my query. I understand that both CTEs and subqueries are essentially running the same query, but “worded” differently, however I didn’t change anything from Task 3.7. I’m not sure if that’s correct or not, but I was just following the notes and how to create a CTE.

At first, I struggled because I was attempting to rewrite the queries but couldn’t get the same output. I then applied the same logic I had from the first question to my second question and realized that the output was the same. Had I seen a different answer, I may be extracting the incorrect columns or my CTEs would’ve gone wrong somewhere.