Exercise 3.9 Common Table Expressions

Step 1: Answer the business questions from step 1 and 2 of task 3.8 using CTEs

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

  Query from Step 1:

```
WITH top_5_customers(customer_id, first_name, last_name, country, city,
total_amount_paid)AS
(SELECT A.customer_id, A.first_name , A.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
```

Query Query History

```
WITH top_5_customers(customer_id, first_name, last_name, country, city,
total_amount_paid)AS
(SELECT A.customer_id, A.first_name , A.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
```

Data Output Messages Notifications

	average numeric
1	105.5540000000000000

Query from Step 2:

```
WITH top_5_customers (customer_id,first_name,last_name,country,city,total_amount_paid)AS
(SELECT A.customer_id, A.first_name , A.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
```

Data Output Messages Notifications

	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

- Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.

  First, I have defined the CTE using the WITH clause, then I gave CTE a name "top_5_customers" provided keyword "AS". Then I have added the query same from 3.8 in parentheses, rest of the logic is same only references slightly changed.

Step 2: Compare the performance of your CTEs and subqueries.

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

   CTE is better because for nesting multiple subqueries into hundreds of statements it will be difficult to handle the code. CTE is a temporary table we can reference in the main query that follows it.

2. Compare the costs of all the queries by creating query plans for each one.

3. 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.

   After using EXPAIN function I have found that there is slightly difference in the costs

4. Did the results surprise you? Write a few sentences to explain your answer.

   The results did not surprise me as using CTE in SQL means optimizing the queries which is more efficient. We just need to define CTE at the beginning then it will update automatically every time.

Step 3:

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

For me learning SQL itself a challenge which I am really enjoying. Replacing subqueries with CTEs is little complex and challenging though the logic was same. But the syntax is very clear just need to practice for it so we can use it easily for coming projects.