3.9 Common Table Expressions

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

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

**Subquery 1**

WITH average\_cte (customer\_id, first\_name, last\_name, city, country, Total\_amount\_table) AS

(SELECT A.customer\_id, A.first\_name, A.last\_name, C.city, D.country,

SUM (E.amount) AS Total\_amount\_table

FROM customer A

INNER JOIN address B ON B.address\_id = A.address\_id

INNER JOIN city C ON C.city\_id = B.city\_id

INNER JOIN country D ON D.country\_id = C.country\_id

INNER JOIN payment E ON E.customer\_id = A.customer\_id

WHERE C.city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary','Tianjin', 'Cianjur')

GROUP BY A.customer\_id, C.city, D.country

ORDER BY Total\_amount\_table desc

LIMIT 5)

SELECT AVG(Total\_amount\_table) AS Average\_amount

FROM average\_cte

Graphical user interface, text, application, chat or text message

Description automatically generated

**Subquery 2**

/\* A. customer B. addressC. city D. country E. payment, F.rental \*/

WITH Top\_customer\_count\_CTE (customer\_id, first\_name, last\_name, city, country, total\_amount\_paid) AS

(SELECT A.customer\_id, A.first\_name, A.last\_name, C.city, D.country,

SUM (E.amount) AS total\_amount\_paid

FROM customer A

INNER JOIN address B ON B.address\_id = A.address\_id

INNER JOIN city C ON C.city\_id = B.city\_id

INNER JOIN country D ON D.country\_id = C.country\_id

INNER JOIN payment E ON E.customer\_id = A.customer\_id

WHERE C.city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki',

'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary','Tianjin', 'Cianjur')

GROUP BY A.customer\_id, C.city, D.country

ORDER BY total\_amount\_paid DESC

LIMIT 5),

All\_customer\_count\_CTE AS

(SELECT D.country,

COUNT (DISTINCT A.customer\_id) AS all\_customer\_count

--COUNT (DISTINCT customer\_id) AS top\_customer\_count--

FROM country D

INNER JOIN city C ON D.country\_id = C.country\_id

INNER JOIN address B ON C.city\_id = B.city\_id

INNER JOIN customer A ON B.address\_id = A.address\_id

GROUP BY D.country)

SELECT D.country, COUNT(DISTINCT A.customer\_id) AS all\_customer\_count, COUNT(DISTINCT Top\_customer\_count\_CTE.customer\_id) 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\_customer\_count\_CTE ON D.country = Top\_customer\_count\_cte.country

GROUP BY D.country

ORDER BY top\_customer\_count DESC

LIMIT 5

1. **Copy-paste your CTEs and their outputs into your answers document.**
2. **Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.**
   * First thing was to define the statement that was going to be declared as the CTE. In this case I copied and pasted the inner statement done in the previous exercise.   
     After I defined the CTE by writing the function WITH … AS and giving a name to the CTE.
   * Finally I wrote the main statement by using the function select, and used the aggregation function AVG to get the average total amount.

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

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

I think CTE will be faster since it is a temporary table.

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

**Subquery**

* + Speed .038 milliseconds
  + Cost= 64.41..64.42 rows=1 width=32

**CTE**

* + Speed .039 milliseconds
  + Cost=64.41..64.42 rows=1 width=32

In both cases the cost is the same but the speed for the Subquery was a little bit faster.

1. **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.**
2. **Did the results surprise you? Write a few sentences to explain your answer.**

I thought the difference was going to be a little bigger between both. I was expecting to have a longer time of execution for the subquery since and a bigger cost considering that a CTE works as a view.

In conclusion, it might be the same in time and general performance, but I find it easier how to structure ideas using a CTE than a subquery

**Step 3:**

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

It was easier to understand and adapt it from the subquery, although I still have to practice the how to write it since the first time, I wrote it I didn’t add all the variables after the CTE name, just added one, and the program was marking an error for ambiguity in the variables set.

The first query was easy to adapt as a CTE but I the second one I faced challenges on how to write the main query after the CTEs.   
It was confusing for me how to connect the tables, I still have to practice.