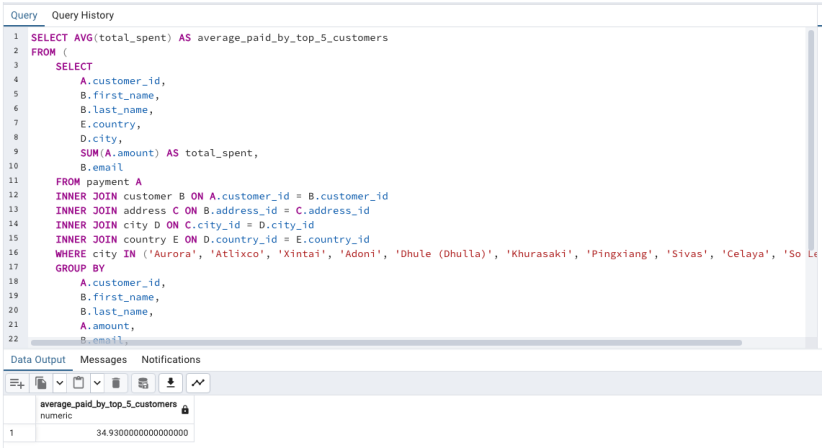
Katie Goyal

3.9 Common Table Expressions

Step 1:

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



2. Copy-paste your CTEs and their outputs into your answers document

3. I first broke down the original query into its component parts to better understand how it was handling data. I then reorganized these parts into a Common Table Expression (CTE) to make the query cleaner and easier to read. This restructuring not only made the query more logical but also improved its performance. After making these changes, I ran the revised query again to make sure it worked correctly and delivered the expected results efficiently.

Step 2:

I believe that using Common Table Expressions instead of subqueries will enhance performance, as they offer clearer readability. CTEs help to organize the structure of the query, potentially reducing its complexity and improving execution efficiency.

- Query 1 Task 3.8
  - Cost 34.90
  - Execution .030 SEC
- Query 1 Task 3.9
  - Cost 34.9
  - Execution .056 SEC
- Query 2 Task 3.8
  - Cost 270.33
  - Execution .075 SEC
- Query 2 task 3.9
  - Cost 107.35
  - Execution .069

I am not surprised by the results, as we observed that the queries using CTEs incurred higher cost and took longer to run.

Step 3:

Switching from subqueries to Common Table Expressions (CTEs) involved significant challenges, primarily in maintaining the integrity and complexity of the original query logic. Adapting to CTEs often required extensive restructuring of query flows and a deep understanding of the data, making the process time-consuming. Additionally, while CTEs enhance query readability and maintenance, they don't always improve performance; in fact, in large datasets, CTEs can lead to slower execution times. This necessitated rigorous performance testing and adjustments, including indexing and query hints, to ensure the system's efficiency was not compromised.