MPivec_Exercise 3.9: Common Table Expressions

Step 1: Answer the business questions from task 3.8 as CTEs

1. Average amount paid by top 5 customers

```
Query Query History
          WITH average_total_paid_cte(customer_id,first_name,last_name,city,countyr,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
       8 FROM customer A
       9 INNER JOIN address B ON A.address_id = B.address_id
      10 INNER JOIN city C ON B.city_id = c.city_id
      11 INNER JOIN country D on C.country_id = D.country_id
      12 INNER JOIN payment E ON A.customer_id = E.customer_id
      13 WHERE C.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'S
      14 GROUP BY A.customer_id, A.first_name, A.last_name, C.city, D.country
      15 ORDER BY total_amount_paid DESC
      16 LIMIT 5)
      17 SELECT ROUND (AVG(total_amount_paid),2) AS average_amount_paid
      18 FROM average_total_paid_cte;
      Data output Messages Notifications
      =+ □ ∨ □ ■ ■ ■ ★ /
           average_amount_paid
                      107.35
a.
```

2. Top 5 customers based in each country

```
Query Query History
 1 EXPLAIN
 2 WITH top_customer_count_cte(amount,customer_id,first_name,last_name,city,country,total_amount_paid)AS
 3 (SELECT A.amount, B.customer_id, B.first_name, B.last_name, D.city, E.country,
         SUM(amount) AS total_amount_paid
 5 FROM payment A
 6 INNER JOIN customer B ON A.customer_id = B.customer_id
 7 INNER JOIN address C ON B.address_id = C.address_id
 8 INNER JOIN city D ON C.city_id = D.city_id
 9 INNER JOIN country E ON D.country_id = E.country_id
10 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
11 GROUP BY A.amount, B.customer_id, B.first_name, B.last_name, D.city, E.country
12 ORDER BY SUM(amount) DESC),
13 customer_count_cte AS(SELECT D.country,
14
       COUNT(DISTINCT A.customer_id) AS all_customer_count, COUNT(DISTINCT D.country) AS top_customer_count
15 FROM customer A
16 INNER JOIN address B ON A.address_id = B.address_id
17 INNER JOIN city C ON B.city_id = C.city_id
18  INNER JOIN country D ON C.country_id = D.country_id
19 GROUP BY D.country)
20 SELECT D.country,
21
        COUNT(DISTINCT A.customer_id) AS all_customer_count, COUNT (DISTINCT top_customer_count_cte.customer_id) AS top_customer_count
22 FROM customer A
23 INNER JOIN address B ON A.address_id = B.address_id
24 INNER JOIN city C ON B.city_id = C.city_id
25 INNER JOIN country D ON C.country_id = D.country_id
26 LEFT JOIN top_customer_count_cte ON D.country=top_customer_count_cte.country
27 GROUP BY D.country
28 ORDER BY top_customer_count DESC
29 LIMIT 5;
```

3. How did you approach this step?

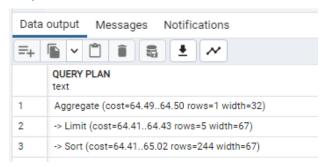
a.

a. From the subquery statements, I replaced the outer statement (first SELECT statement/last AS statement) that chooses what to do with the inner statement. Keeping the inner statement, I insert this into the WITH newtable_cte, listing the columns, and give it an alias. After closing the subquery, I ended the query with a

SELECT statement to show the final result, with an alias, and referencing FROM newtable_cte, as well as grouping, ordering, and limiting to fine tune the results.

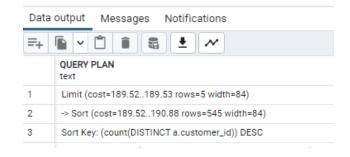
Step 2: Compare the performance of your CTEs and Subqueries

- 1. Which do you think will perform better and why?
 - a. I would have expected the subquery statements to perform better as it seems there are less processing steps involved. Sure enough, the estimated cost and actual run time are less for both subquery statements compared to the CTE statements.
- 2. Compare the costs of all queries by creating query plans for each one
 - a. Subquery cost:
- i. Step 1:

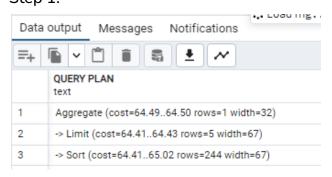


b. CTE cost

ii. Step 2:

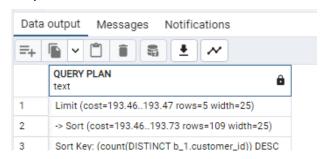


i. Step 1:



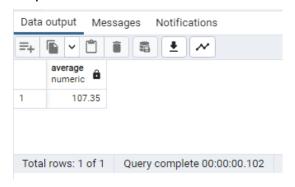
- 3. What is the actual speed of your queries
 - a. Subquery cost:

ii. Step 2:



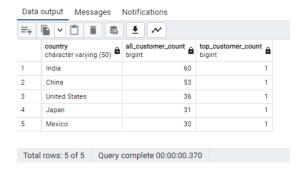
b.

i. Step 1:



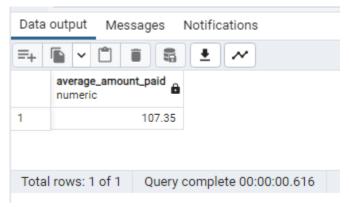
c. CTE cost:

ii. Step 2:

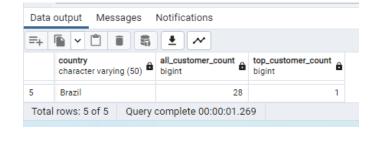


C. CIL COSC

i. Step 1:



i. Step 2:



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

1. The second CTE task was one of the most difficult. In order to have enough information, a lot of extra studying needed to go into this lesson, including reviewing previous exercises and reading all of the resources provided. I don't think we were given enough information to fully grasp what was required in order to accomplish the task. I did have to use previously submitted work to compare my own and help build my understanding. I ended up starting over and restructuring my tables/columns/primary keys. I'm still not certain that I understand it completely, but have a mentor call scheduled where I will review this.