## STEP 1

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

Average amount paid by TOP 5 customers.

```
24 WITH average_amount_cte (total_amount_paid,customer_id,first_name,last_name,city,country) AS
   (SELECT SUM(A.amount) AS total_amount_paid,
26
            B.customer_id,
27
            B.first_name,
28
            B.last_name,
29
            D.city,
30
            E.country
31 FROM payment A
32 INNER JOIN customer B on A.customer_id = B.customer_id
33 INNER JOIN address C on B.address_id = C.address_id
34 INNER JOIN city D on C.city_id = D.city_id
35 INNER JOIN country E on D.country_id = E.country_id
36 WHERE city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei',
37
                  'So Leopoldo', 'Teboksary', 'Tianjin', 'Cianjur')
   GROUP BY B.customer_id,
38
39
             first_name,
40
             last_name,
41
             city,
42
             country
43
   ORDER BY total_amount_paid DESC
44
   LIMIT 5)
45
   SELECT AVG(total_amount_paid) AS average_amount_paid
46
   FROM average_amount_cte
47
Data Output Messages Notifications
                       ♣ ~
   8
    average_amount_paid
    numeric
     105.55400000000000000
```

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

- 1.1: I took the subquery from 3.8 and used it to define the CTE. I then queried the main statement by looking for the average\_amount\_paid.
- 1.2: I proceeded similarly as in previous step. However, here we needed two CTEs: top5\_customer\_count\_cte and all\_customer\_count\_cte

```
Query Query History
48
   WITH
49
   top5_customer_count_cte AS
50
    (SELECT SUM(A.amount) AS total_amount_paid,
51
           B.customer_id,
52
           B.first name.
53
           B.last_name,
54
           D.city,
55
            E.country
56 FROM payment A
57
    INNER JOIN customer B on A.customer_id = B.customer_id
    INNER JOIN address C on B.address_id = C.address_id
    INNER JOIN city D on C.city_id = D.city_id
60 INNER JOIN country E on D.country_id = E.country_id
61 WHERE city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei',
62
                  'So Leopoldo','Teboksary','Tianjin','Cianjur')
63 GROUP BY B.customer_id,
64
             first_name,
65
             last_name,
66
             city,
67
             country
68
   ORDER BY total_amount_paid DESC
69
    LIMIT 5),
70
71
   all_customer_count_cte AS
72
    (SELECT E.country,
73
            COUNT(DISTINCT B.customer_id) AS all_customer_count,
74
            COUNT(DISTINCT E.country) AS top5_customer_count
75 FROM country E
76 INNER JOIN city D ON E.country_id = D.country_id
77
    INNER JOIN address C ON D.city_id = C.city_id
    INNER JOIN customer B ON C.address_id = B.address_id
79
    GROUP BY E.country)
80
81
   SELECT E.country,
82
           COUNT(DISTINCT B.customer_id) AS all_customer_count,
83
           COUNT(DISTINCT top5_customer_count_cte.customer_id) AS top5_customer_count
84 FROM country E
85 INNER JOIN city D ON E.country_id = D.country_id
86 INNER JOIN address C ON D.city_id = C.city_id
   INNER JOIN customer B ON C.address_id = B.address_id
87
   LEFT JOIN
89
    top5_customer_count_cte ON E.country = top5_customer_count_cte.country
90 GROUP BY E.country
91 ORDER BY all_customer_count DESC
92 LIMIT 5
Data Output Messages Notifications
=+ 6 ~ 6 1
                 S
                       all_customer_count
                                       top5_customer_count
     country
     character varying (50)
                       bigint
                                       bigint
     India
                                    60
                                                       1
2
     China
                                    53
                                                       1
     United States
                                                       1
3
                                    36
4
                                    31
                                                       1
     Japan
```

1

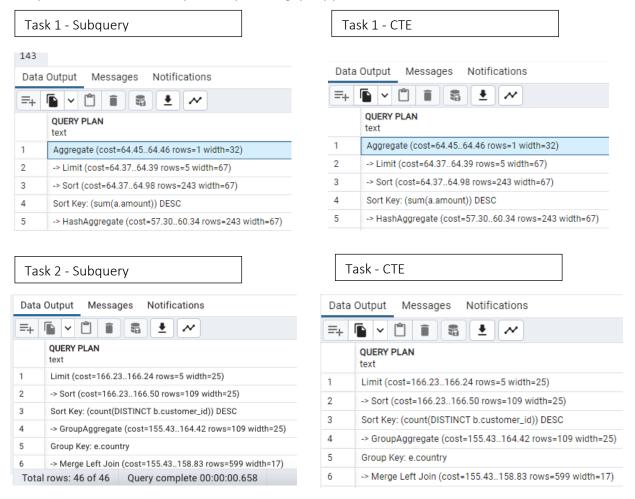
30

Mexico

## Which approach do you think will perform better and why?

I think that in this task, both CTE and subquery methods will perform similarly since the complexity of the queries overall seems rather even. Although CTE queries are more readable.

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



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.

The real time it took to process the queries varied every time I executed them. Sometimes it took a bit longer and sometimes a bit faster.

Overall, the results did not surprise me. The Query plans were similar in both cases.

## STEP 3

This task was one of the most challenging exercises so far. The first task was relatively simple with replacing the subquery for CTE statement. However, the second task was hard mainly because a second CTE had to be added which was not discussed in the reading part of the exercise.

I believe with more practice I will be more comfortable with the syntax and it will be more intuitive.