

Task 3.9 Common Table Expressions

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

1. Question 1:

--AVG amount paid by top 5 customers

```
WITH top_5_customer_cte (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','Atlixco','xintai','Adoni','Dhule (Dhulla)','Kurashiki','Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
      C.city,
      D.country
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT AVG(total_amount_paid) AS average
FROM top_5_customer_cte
```

	average numeric
1	107.3540000000000000

Question 2:

--How many top 5 customer are based within each country?

```
WITH top_5_customer_cte (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','Atlixco','xintai','Adoni','Dhule (Dhulla)','Kurashiki','Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
      C.city,
      D.country
ORDER BY total_amount_paid DESC
```

```

LIMIT 5)
SELECT DISTINCT(A.country),
       COUNT(DISTINCT D.customer_id) AS all_customer_count,
       COUNT(DISTINCT A.country) AS top_customer_count
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
LEFT JOIN top_5_customer_cte E ON A.country = E.country
GROUP BY A.country
ORDER BY all_customer_count DESC
LIMIT 5;

```

	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

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

First step was to write out the subquery from the previous exercise. Second step was to place the query from step 1 within the CTE syntax and name the CTE. The final step was to write the main statement. For the first question this meant selecting the average from the results of the CTE table. The second questions main statement involved joining the CTE through a left join in order to input the results of the top 5 customers and their countries into the top_customer_count column.

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

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

The only way to find out which one will perform better is to compare using EXPLAIN, but since both the CTE and subquery are only referenced once in the statement I believe they would perform the same.

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

QUERY	COST
Step 1 subquery	64.49 ... 64.50
Step 2 subquery	189.52 ... 189.53
Step 1 CTE	64.49 ... 64.50
Step 2 CTE	168.14 ... 168.19

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.

4.

QUERY	COST
Step 1 subquery	35msec
Step 2 subquery	51msec
Step 1 CTE	40msec
Step 2 CTE	37msec

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

Yes, the results surprised me in that they were different for similar questions. I was also surprised that I got different costs for the same queries ran back-to-back. I wasn't surprised that the question 1 asking for a single column back (avg) was quicker to run than question 2.

Step 3:

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

I found writing out question 2's CTE more difficult to figure out how to add it to the left join but once figuring it out you see how much easier it is to go back and read the statements.