3.9: Common Table Expressions

Innocent Bayai

15 June, 2024

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

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

Step 1 of task 3.8

```
WITH average_amount_cte (customer_id, first_name, last_name, country, city) AS

(SELECT B. customer_id,
        B. first_name,
        B. last_name,
        E. country,
        D. city,
        SUM (A. amount) AS total_amount_paid
FROM payment A
        JOIN customer B ON A. customer_id = B. customer_id
        JOIN address C ON B. address_id = C. address_id
        JOIN city D ON C. city_id = D. city_id
        JOIN country E ON D. country_id = E. country_id
WHERE
D. city IN (
SELECT D.city
FROM customer B
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D.country_id = E.country_id
GROUP BY D.city
ORDER BY COUNT(B. customer_id) DESC
LIMIT 10
)
GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT AVG (total_amount_paid)
FROM average_amount_cte
```

Query Query History

```
WITH average_amount_cte (customer_id, first_name, last_name, country, city) AS
(SELECT B. customer_id,B. first_name, B. last_name,E. country,D. city,
                    SUM (A. amount) AS total_amount_paid
FROM payment A
                   JOIN customer B ON A. customer_id = B. customer_id
                   JOIN address C ON B. address_id = C. address_id
                   JOIN city D ON C. city_id = D. city_id
                   JOIN country E ON D. country_id = E. country_id WHERE
D. city IN (SELECT D.city
FROM customer B
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D.country_id = E.country_id
GROUP BY D.city
ORDER BY COUNT(B. customer_id) DESC
LIMIT 10 )
GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT AVG (total_amount_paid)
FROM average_amount_cte
```

Data Output Messages Notifications

	avg numeric
1	156.8500000000000000

Total rows: 1 of 1 Query complete 00:00:00.060

Step 2 of task 3.8

```
WITH top_5_count_cte (customer_id, first_name, last_name, country, city) AS
(SELECT B. customer_id,B. first_name, B. last_name, E. country, D. city,
 SUM (A. amount) AS total_amount_paid
FROM payment A
JOIN customer B ON A. customer_id = B. customer_id
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D. country_id = E. country_id
WHERE D. city IN (
SELECT D.city
FROM customer B
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D.country_id = E.country_id
WHERE E.country        IN      (SELECT        E.country
FROM   customer        B
JOIN   address C       ON      B.address_id   =      C.address_id
JOIN   city    D       ON      C.city_id      =      D.city_id
JOIN   country E       ON      D.country_id   =      E.country_id
GROUP BY       E.country
ORDER BY       COUNT(B.customer_id)  DESC
LIMIT  10)
GROUP BY       E.country,D.city
```

```
ORDER  BY        COUNT(B.customer_id)   DESC
LIMIT     10)
GROUP  BY        B.customer_id,B.first_name,B.last_name,E.country, D.city
ORDER  BY        SUM(A.amount)  DESC
LIMIT     5)
SELECT
E.country,
COUNT (DISTINCT B.customer_id) AS all_customer_count,
COUNT (DISTINCT top_5_count_cte) AS top_customer_count
FROM customer B
JOIN address C ON B.address_id = C.address_id
JOIN city D ON C.city_id = D.city_id
JOIN country E ON D.country_id = E.country_id
LEFT JOIN top_5_count_cte ON B.customer_id = top_5_count_cte.customer_id
GROUP BY
E.country
ORDER BY
all_customer_count DESC
LIMIT 5
```

Query Query History

```
WITH top_5_count_cte (customer_id, first_name, last_name, country, city) AS
(SELECT B. customer_id,B. first_name, B. last_name, E. country, D. city,
 SUM (A. amount) AS total_amount_paid
FROM payment A
JOIN customer B ON A. customer_id = B. customer_id
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D. country_id = E. country_id
WHERE D. city IN (
SELECT D.city
FROM customer B
JOIN address C ON B. address_id = C. address_id
JOIN city D ON C. city_id = D. city_id
JOIN country E ON D.country_id = E.country_id
WHERE   E.country   IN  (SELECT E.country
FROM    customer    B
JOIN    address C   ON  B.address_id    =   C.address_id
JOIN    city    D   ON  C.city_id   =   D.city_id
JOIN    country E   ON  D.country_id    =   E.country_id
GROUP   BY  E.country
ORDER   BY  COUNT(B.customer_id)    DESC
LIMIT   10)
GROUP   BY  E.country,D.city
ORDER   BY  COUNT(B.customer_id)    DESC
LIMIT   10)
GROUP   BY  B.customer_id,B.first_name,B.last_name,E.country, D.city
ORDER   BY  SUM(A.amount)   DESC
LIMIT   5)
SELECT
E.country,
COUNT (DISTINCT B.customer_id) AS all_customer_count,
```

```sql
SELECT
E.country,
COUNT (DISTINCT B.customer_id) AS all_customer_count,
COUNT (DISTINCT top_5_count_cte) AS top_customer_count
FROM customer B
JOIN address C ON B.address_id = C.address_id
JOIN city D ON C.city_id = D.city_id
JOIN country E ON D.country_id = E.country_id
LEFT JOIN top_5_count_cte ON B.customer_id = top_5_count_cte.customer_id
GROUP BY
E.country
ORDER BY
all_customer_count DESC
LIMIT 5
```

Data Output Messages Notifications

	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

Total rows: 5 of 5 Query complete 00:00:00.446

I took queries from Ex 3.8 and made sure that I removed all irrelevant sections of the queries. I placed the CTE and treated the query from Ex. 3.8 as subquery of the CTE. The resultant query failed to run as it had some unnecessary queries especially at the end. After running into dead ends, I checked both AI and some model answers for guidance. And finally I got it correct!!

My experience so far tells me that the process is still complex and long and my intuition is asking if there is a way of using two or three CTEs as a way of reducing the complexity of the query. I will try it outside this exercise.

2.	**How did the queries you produced compare with those suggested by ChatGPT?**
	My comparison is based on the cost aspect associated with sub-queries vs CTE-based queries. My major take home is that, whilst CTEs might simply the manner of coming up with queries to solve any task, there might be little if no different at all with respect to the costs incurred by both methods. I use EXPLAIN to get the costs linked to both methods.
	Did you find any errors in your own work?
	Yes, I found errors mostly because some sections of the queries from 3.8 were no longer necessary once a CTE has been adopted. After cleaning the query, everything went well.
	Did you find any errors in ChatGPT's suggestions?
	Yes, ChatGPT's suggestions require to be customed to the case being investigated. Otherwise, the queries are a good starting point but one must be informed about the direction of the query so that they don't have to rely 100% on ChatGPT.

4. Reflect on your experience using AI for this task.
    1. Did it save you time or aid your learning?
    2. Would you use it again in the future?

AI remains pivotal as it guides the thinking process. However, it cannot undo the complexities around the generation of appropriate queries. It certainly saves time and aids the learning process too. I definitely will use it again and again whilst taking note of the fact that AI cannot do the work, but guide my thinking process.