Databases & SQL for Analysts

Task 3.9: Common Table Expressions

Ngawang Dhundup

LINK NAME:

Directions:

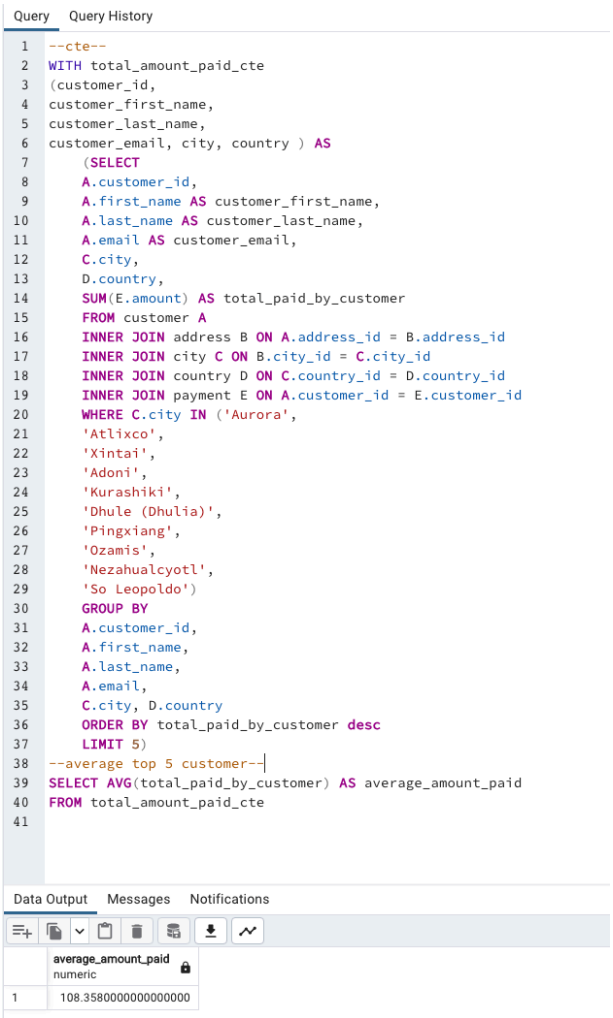
Step 1: Answer the business questions from step 1 and 2 of task 3.8 using CTEs

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

STEP 1

```
WITH total_amount_paid_cte (customer_id, customer_first_name, customer_last_name,
customer_email, city, country ) AS
        (SELECT
        A.customer_id,
        A.first_name AS customer_first_name,
        A.last_name AS customer_last_name,
        A.email AS customer_email,
        C.city,
        D.country,
        SUM(E.amount) AS total_paid_by_customer
        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 C.city IN ('Aurora',
        'Atlixco',
        'Xintai',
        'Adoni',
        'Kurashiki',
        'Dhule (Dhulia)',
        'Pingxiang',
        'Ozamis',
```

'Nezahualcyotl',
'So Leopoldo')
GROUP BY
A.customer_id,
A.first_name,
A.last_name,
A.email,
C.city, D.country
ORDER BY total_paid_by_customer desc
LIMIT 5)
SELECT AVG(total_paid_by_customer) AS average_amount_paid
FROM total_amount_paid_cte



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

STEP 2

```
--first cte = top 5 customers--
WITH top_5_customers_cte AS
                (SELECT
                A.customer_id,
                A.first_name AS customer_first_name,
                A.last_name AS customer_last_name,
                A.email AS customer_email,
                C.city,
                D.country,
                SUM(E.amount) AS total_paid_by_customer
                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 C.city IN ('Aurora',
                'Atlixco',
                'Xintai',
                'Adoni',
                'Kurashiki',
                'Dhule (Dhulia)',
                'Pingxiang',
                'Ozamis',
                'Nezahualcyotl',
                'So Leopoldo')
                GROUP BY
                A.customer_id,
                A.first_name,
                A.last_name,
                A.email,
                C.city, D.country
                ORDER BY total_paid_by_customer DESC
                LIMIT 5)
        SELECT D.country,
```

COUNT(DISTINCT A.customer_id) AS all_customer_count,
COUNT(DISTINCT D.country) AS top_customer_count
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
LEFT JOIN
top_5_customers_cte
ON D.country =
top_5_customers_cte.country
GROUP BY D.country
ORDER BY
all_customer_count DESC
LIMIT 5

Query | Query History

```sql
--first cte = top 5 customers--
WITH top_5_customers_cte AS
        (SELECT
        A.customer_id,
        A.first_name AS customer_first_name,
        A.last_name AS customer_last_name,
        A.email AS customer_email,
        C.city,
        D.country,
        SUM(E.amount) AS total_paid_by_customer
        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 C.city IN ('Aurora',
        'Atlixco',
        'Xintai',
        'Adoni',
        'Kurashiki',
        'Dhule (Dhulia)',
        'Pingxiang',
        'Ozamis',
        'Nezahualcyotl',
        'So Leopoldo')
        GROUP BY
        A.customer_id,
        A.first_name,
        A.last_name,
        A.email,
        C.city, D.country
        ORDER BY total_paid_by_customer DESC
        LIMIT 5)
--counting by country--
    SELECT D.country,
    COUNT(DISTINCT A.customer_id) AS all_customer_count,
    COUNT(DISTINCT D.country) AS top_customer_count
    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
    LEFT JOIN
    top_5_customers_cte
    ON D.country = top_5_customers_cte.country
    GROUP BY D.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

3. Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.
   - I copied the subquery into the pg4. Then wrote a WITH clause to create a CTE View. After adding all the columns used in the subquery, I named the CTE an appropriate name(total_amount_paid). After that I conducted. Simple SELECT function to calculate the average, FROM the CTE I created.
   - For the second step, I created 2 CTES. The first one was for the subquery of top 5 customers. Then the second CTE was selecting the counts after left joining with the first CTE.

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

1. Which approach do you think will perform better and why?
   - CTES have easier readability and seem to be more useful if referencing the same table or query multiple times. However the cost I believe will be the same, since the lines of code are similar.
2. Compare the costs of all the queries by creating query plans for each one.
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.
   - 64.41 step 1 CTE
   - 179.98 step 2 CTE
   - 64.41 step 1 subquery
   - 189.52 step 2 subquery
4. Did the results surprise you? Write a few sentences to explain your answer.
   - The results speed and cost wise were about the same. Makes sense since the lines of code were about the same.

Step 3:

Write 1 to 2 paragraphs on the challenges you faced when replacing your subqueries with CTEs.
The challenges I had were primarily with the second step. Figuring out that I needed two CTEs took a while. And then figuring out how to join the tables using the first CTE was hard. Knowing all the links and how the tables and columns were connected was difficult.
I still don't understand how it all works. Especially the highlighted part;

```
LEFT JOIN
        top_5_customers_cte
        ON D.country = top_5_customers_cte.country
*/ why the cte.country part??
        GROUP BY D.country, top_5_customers_cte
        ORDER BY all_customer_count DESC
        LIMIT 5)
-- selecting results after combining 2 ctes--
SELECT *
FROM count_of_country_cte
```

Step 4:

Save your “Answers 3.9” document as a PDF and upload it here for your tutor to review.