SQL TASK 9: CTEs

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.

a) Average payment by Top5 customers

```sql
WITH top5_customers AS (
  SELECT
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    city.city,
    country.country,
    SUM(payment.amount) AS total_payment_amount
  FROM payment
  INNER JOIN customer ON payment.customer_id = customer.customer_id
  INNER JOIN address  ON customer.address_id = address.address_id
  INNER JOIN city     ON address.city_id = city.city_id
  INNER JOIN country  ON city.country_id = country.country_id
  WHERE city.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni',
                      'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang',
                      'Sivas', 'Celaya', 'So Leopoldo')
    AND country.country IN ('India', 'China', 'United States', 'Japan',
                            'Mexico', 'Brazil', 'Russian Federation',
                            'Philippines', 'Turkey', 'Indonesia')
  GROUP BY customer.customer_id, customer.first_name, customer.last_name,
           city.city, country.country
  ORDER BY total_payment_amount DESC
  LIMIT 5
)
SELECT AVG(total_payment_amount) AS avg_paid_top5
FROM top5_customers;
```

Data Output | Messages | Notifications

	avg_paid_top5 numeric
1	107.3540000000000000

b) How many of top5 customers are based in each country

```sql
WITH top_5_customers AS (
  SELECT
    customer.customer_id,
    country.country
  FROM payment
  INNER JOIN customer ON payment.customer_id = customer.customer_id
  INNER JOIN address  ON customer.address_id = address.address_id
  INNER JOIN city     ON address.city_id     = city.city_id
  INNER JOIN country  ON city.country_id     = country.country_id
  WHERE city.city IN ('Aurora','Atlixco','Xintai','Adoni',
                      'Dhule (Dhulia)','Kurashiki','Pingxiang',
                      'Sivas','Celaya','So Leopoldo')
    AND country.country IN ('India','China','United States','Japan',
                            'Mexico','Brazil','Russian Federation',
                            'Philippines','Turkey','Indonesia')
  GROUP BY customer.customer_id, country.country
  ORDER BY SUM(payment.amount) DESC
  LIMIT 5
)
SELECT
  country.country,
  COUNT(DISTINCT customer.customer_id) AS all_customer_count,
  COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
FROM customer
INNER JOIN address ON customer.address_id = address.address_id
INNER JOIN city    ON address.city_id     = city.city_id
INNER JOIN country ON city.country_id     = country.country_id
LEFT JOIN top_5_customers
       ON top_5_customers.country = country.country
GROUP BY country.country
ORDER BY top_customer_count DESC, all_customer_count DESC
LIMIT 5;
```

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	2
2	India	60	1
3	United States	36	1
4	Turkey	15	1
5	China	53	0

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

   I tried to rewrite the query step by step based on the lesson learned in this task. I started with creating the CTE command and then copy-pasted most of the lines of the previous

subquery. At the end, I wrote the main query, which used the (referred to) to fetch the desired data.

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

1. Which approach do you think will perform better and why?
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 you've run each query, a popup window will display its speed in milliseconds.
4. Did the results surprise you? Write a few sentences to explain your answer.

Query NO.	Subquery	CTE
1	Cost: 24.64	Cost: 24.64
	Time: 0.429	Time: 0.457
2	Cost: 126.04	Cost: 126.98
	Time: 1.775	Time: 3.130

I have assumed that CTEs are optimal in any case, but it seems that I was wrong. Obviously the time is less in longer queries in the subquery version. However, as a junior analyst, I prefer CTEs because of higher readability and the capability of being referred to through the main query.

Step 3:

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

I had to spend some time to decide which part I should put in the CTE, especially in the second query because it was longer and it had two nested join parts. All in all, it reminded me of CLASS feature in programming, which is an integral and basic part of it.