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

   - Step 1: Find the average amount paid by the top 5 customers.

Rockbuster/postgres@PostgreSQL 13

Query Editor | Query History

```
WITH subquery_cte ("Customer_ID", "Customer_First_Name", "Customer_Last_Name", "Country", "City", "Total_Amount_Paid")
    AS (SELECT B.customer_id AS "Customer_ID",
            B.first_name AS "Customer_First_Name",
            B.last_name AS "Customer_Last_Name",
            E.country AS "Country",
            D.city AS "City",
        SUM(amount) AS "Total_Amount_Paid"
        FROM payment A
        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
        INNER JOIN country E ON D.country_id = E.country_id
        WHERE country IN ('India','China','United States','Japan','Mexico','Brazil','Russian Federation','Philippines','Turkey','Indonesia')
        GROUP BY "Customer_ID","Customer_First_Name","Customer_Last_Name","City","Country"
        ORDER BY "Total_Amount_Paid" DESC
        LIMIT 5)
SELECT "Customer_ID",
        "Customer_First_Name",
        "Customer_Last_Name",
        "City",
        "Country",
        "Total_Amount_Paid",
        ROUND(AVG("Total_Amount_Paid"), 2) AS "Average_Amount_Paid"
FROM subquery_cte
GROUP BY "Customer_ID","Customer_First_Name","Customer_Last_Name","City","Country","Total_Amount_Paid"
ORDER BY "Total_Amount_Paid" DESC
```

Data Output | Explain | Messages | Notifications

	Customer_ID integer	Customer_First_Name character varying (45)	Customer_Last_Name character varying (45)	City character varying (50)	Country character varying (50)	Total_Amount_Paid numeric	Average_Amount_Paid numeric
1	526	Karl	Seal	Cape Coral	United States	208.58	208.58
2	178	Marion	Snyder	Santa Brbara dOeste	Brazil	194.61	194.61
3	181	Ana	Bradley	Memphis	United States	167.67	167.67
4	236	Marcia	Dean	Tanza	Philippines	166.61	166.61
5	403	Mike	Way	Valparai	India	162.67	162.67

- Step 2: Find out how many of the top 5 customers are based within each country.

Rockbuster/postgres@PostgreSQL 13

Query Editor | Query History

```
WITH top_5_customers_cte ("Customer_ID", "Customer_First_Name", "Customer_Last_Name", "Country", "City", "Total_Amount_Paid")
    AS (SELECT B.customer_id AS "Customer_ID",
           B.first_name AS "Customer_First_Name",
           B.last_name AS "Customer_Last_Name",
           E.country AS "Country",
           D.city AS "City",
           SUM(amount) AS "Total_Amount_Paid"
        FROM payment A
           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
           INNER JOIN country E ON D.country_id = E.country_id
        WHERE country IN ('India','China','United States','Japan','Mexico','Brazil','Russian Federation','Philippines','Turkey','Indonesia')
        GROUP BY "Customer_ID","Customer_First_Name","Customer_Last_Name","City","Country"
        ORDER BY "Total_Amount_Paid" DESC
        LIMIT 5)
SELECT D.country AS "Country",
        COUNT(DISTINCT A.customer_id) AS "all_customer_count",
        COUNT (DISTINCT "top_5_customers_cte") 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 "top_customer_count" DESC, "all_customer_count" DESC
```

Data Output | Explain | Messages | Notifications

	Country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	United States	36	2
2	India	60	1
3	Brazil	28	1
4	Philippines	20	1
5	China	53	0
6	Japan	31	0
7	Mexico	30	0
8	Russian Federation	28	0
9	Turkey	15	0

- First, I copied my entire query from step 1 of task 3.8. Then I selected the subquery and pasted it into a new query window to create the CTE. Next, I pasted the remainder of my original query, pasted it below the CTE and wherever the subquery was referenced, I updated it to reference the CTE. I ran the query and checked the data output against the original query output. Then I followed the same steps for step 2 of task 3.8.

2. Compare the performance of your CTEs and subqueries:

- I expect the subquery approach to be slower because the general recommendation is to use CTEs instead of subqueries whenever possible.
- Query 1: amount paid by the top 5 customers.
    - Option 1: subquery
        - Estimated cost: 202.23
        - Total runtime: 77 msec.

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	GroupAggregate (cost=202.23..202.40 rows=5 width=99)
2	Group Key: subquery."Total_Amount_Paid", subquery."Customer_ID", subquery."Customer_First_Name", subquery."Customer_Last_Name", subquery."
3	-> Sort (cost=202.23..202.24 rows=5 width=67)
4	Sort Key: subquery."Total_Amount_Paid" DESC, subquery."Customer_ID", subquery."Customer_First_Name", subquery."Customer_Last_Name", su

- Option 2: CTE
  - Estimated cost: 203.23
  - Total runtime: 73 msec.

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	GroupAggregate (cost=202.23..202.40 rows=5 width=99)
2	Group Key: subquery_cte."Total_Amount_Paid", subquery_cte."Customer_ID", subquery_cte."Customer_First_Name", subquery_cte."Customer_Last_Nam
3	-> Sort (cost=202.23..202.24 rows=5 width=67)
4	Sort Key: subquery_cte."Total_Amount_Paid" DESC, subquery_cte."Customer_ID", subquery_cte."Customer_First_Name", subquery_cte."Customer_La

- Query 2: how many of the top 5 customers are based within each country.
  - Option 1: subquery
    - Estimated cost: 305.84
    - Total runtime: 87 msec.

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Sort (cost=305.84..306.11 rows=109 width=25)
2	Sort Key: (count(DISTINCT top_5_customers.*)) DESC, (count(DISTINCT a.customer_id)) DESC
3	-> GroupAggregate (cost=293.17..302.15 rows=109 width=25)
4	Group Key: d.country

  - Option 2: CTE
    - Estimated cost: 305.84
    - Total Runtime: 117 msec.

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Sort (cost=305.84..306.11 rows=109 width=25)
2	Sort Key: (count(DISTINCT top_5_customers_cte.*)) DESC, (count(DISTINCT a.customer_id)) DESC
3	-> GroupAggregate (cost=293.17..302.15 rows=109 width=25)
4	Group Key: d.country

  - I didn't expect these results. Estimated costs are identical and runtimes are similar between the CTE and subquery options.

3. Challenges replacing subqueries with CTEs:

   My main challenge was grasping the concept of CTEs vs. subqueries and understanding the CTE syntax. After that, everything went fairly smoothly. I also realized how important it was to check my entire query to ensure table and column names referenced the correct table and column names.