Olusola Olaniyi

Task 3.8 Performing Subqueries

Step 1.

A screenshot of a computer program

Description automatically generated

Step 2.

SELECT

D.country,

COUNT(DISTINCT A.customer\_id) AS all\_customer\_count,

COUNT(top\_5\_customers) 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

(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 C.city IN ('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambacur', 'Shanwei', 'So Leopoldo',

'Teboksary', 'Tianjin', 'Cianjur')

GROUP BY A.customer\_id,

A.First\_name,

A.last\_name,

C.city,

D.country

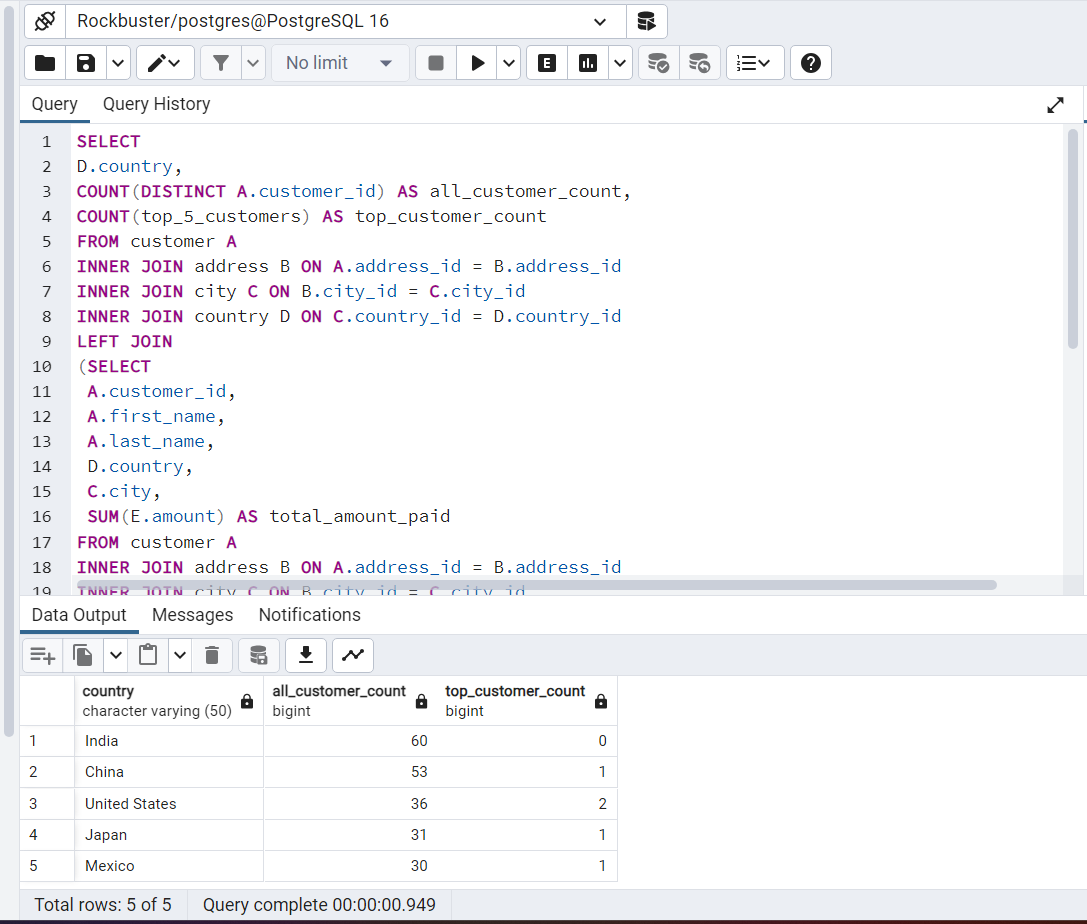
ORDER BY total\_amount\_paid DESC

LIMIT 5) top\_5\_customers ON A.customer\_id = top\_5\_customers.customer\_id

GROUP BY D.country

ORDER BY all\_customer\_count DESC

LIMIT 5;



Step 3.

Well, I think step 1 and 2 should have been done without using subqueries because we've been able to identify top 10 cities in the last task 3.7. Though subqueries are very useful in complex queries with multiple joins. I think subqueries are very useful where a query depends on the results of another query. Using only JOINS could have been less complex.