June 1, 2022

Exercise 3.8

STEP 1

5.
SELECT AVG(total_amount_paid) AS average
FROM (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','Ambattur','Shanwei','So
Leopoldo','Teboksary','Tianjin','Cianjur')
GROUP BY A.customer_id, D.country, C.city
ORDER BY "total_amount_paid" DESC
LIMIT 5) AS "total_amount_paid"

Final Data Output
105.5540000000000000

STEP 2

SELECT DISTINCT(D.country),
COUNT(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 (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','Ambattur','Shanwei','So
Leopoldo','Teboksary','Tianjin','Cianjur')
GROUP BY A.customer_id, D.country, C.city
ORDER BY "total_amount_paid" DESC
LIMIT 5) AS top_5_customers
ON D.country=top_5_customers.country
GROUP BY D.country, top_5_customers.country
ORDER BY all_customer_count DESC
LIMIT 5

Rank	Country	all_customer_count	top_customer_count
1	"India"	60	1
2	"China"	53	1
3	"United States"	36	1
4	"Japan"	31	1
5	"Mexico"	30	1

STEP 3

Step 1 could have been done without a subquery as we were only seeking to find the average of the top 5 customers, no matter which country. Step 2 could only have been done with a subquery as we were looking for how many of the top 5 customers were based in each country, multiple tables.

Subqueries are useful when you would like something from multiple tables, especially when the tables are constantly being updated.