CareerFoundry Answers 3.8

Gabriel Pollicar

November 19, 2023

1.

```
-- Finds Average of Total Payments from Top 5 Customers
SELECT AVG(Top5_Payments)
FROM (
	-- Finds Top 5 Customers in Top 5 Cities
	SELECT A.customer_id,
			A.first_name,
			A.last_name,
			d.country,
			c.city,
			SUM(amount) AS Top5_Payments
	FROM customer A
	INNER JOIN payment E ON A.customer_id = E.customer_id
	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
	WHERE c.city IN(SELECT c.city
		-- Finds Top 5 Cities in Top 5 Country
		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
		WHERE country IN (SELECT D.country
			-- Finds Top 5 Country
			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
			GROUP By d.country
			ORDER BY COUNT(customer_id) DESC
			LIMIT 10)
		GROUP By d.country, c.city
		ORDER BY COUNT(A.customer_id) DESC
		LIMIT 10)
	GROUP By A.customer_id, A.first_name, A.last_name, d.country, c.city
	ORDER BY SUM(amount) DESC
	 LIMIT 5);
```

```sql
-- Finds Average of Total Payments from Top 5 Customers
SELECT AVG(Top5_Payments)
FROM (
    -- Finds Top 5 Customers in Top 5 Cities
    SELECT A.customer_id,
            A.first_name,
            A.last_name,
            d.country,
            c.city,
            SUM(amount) AS Top5_Payments
    FROM customer A
    INNER JOIN payment E ON A.customer_id = E.customer_id
    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
    WHERE c.city IN(SELECT c.city
        -- Finds Top 5 Cities in Top 5 Country
        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
        WHERE country IN (SELECT D.country
            -- Finds Top 5 Country
            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
            GROUP By d.country
            ORDER BY COUNT(customer_id) DESC
            LIMIT 10)
        GROUP By d.country, c.city
        ORDER BY COUNT(A.customer_id) DESC
        LIMIT 10)
    GROUP By A.customer_id, A.first_name, A.last_name, d.country, c.city
    ORDER BY SUM(amount) DESC
     LIMIT 5)
```

Data Output | Messages | Notifications

	avg numeric
1	105.5540000000000000

2.

```
SELECT d.country AS Country,
		COUNT(DISTINCT A.customer_id) AS all_customer_count,
		COUNT(top_5_customers) AS top_5_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,
			SUM(amount) AS Top5_Payments
	FROM customer A
	INNER JOIN payment E ON A.customer_id = E.customer_id
	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
	WHERE c.city IN(SELECT c.city
		-- Finds Top 5 Cities in Top 5 Country
		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
		WHERE country IN (SELECT D.country
			-- Finds Top 5 Country
			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
			GROUP By d.country
			ORDER BY COUNT(customer_id) DESC
			LIMIT 10)
		GROUP By d.country, c.city
		ORDER BY COUNT(A.customer_id) DESC
		LIMIT 10)
	GROUP By A.customer_id, A.first_name, A.last_name, d.country, c.city
	ORDER BY SUM(amount) DESC
	LIMIT 5
) AS top_5_customers ON A.customer_id = top_5_customers.customer_id
GROUP BY d.country
ORDER BY top_5_customer_count DESC
```

```sql
SELECT d.country AS Country,
        COUNT(DISTINCT A.customer_id) AS all_customer_count,
        COUNT(top_5_customers) AS top_5_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,
            SUM(amount) AS Top5_Payments
    FROM customer A
    INNER JOIN payment E ON A.customer_id = E.customer_id
    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
    WHERE c.city IN(SELECT c.city
        -- Finds Top 5 Cities in Top 5 Country
        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
        WHERE country IN (SELECT D.country
            -- Finds Top 5 Country
            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
            GROUP By d.country
            ORDER BY COUNT(customer_id) DESC
            LIMIT 10)
        GROUP By d.country, c.city
        ORDER BY COUNT(A.customer_id) DESC
        LIMIT 10)
    GROUP By A.customer_id, A.first_name, A.last_name, d.country, c.city
    ORDER BY SUM(amount) DESC
    LIMIT 5
) AS top_5_customers ON A.customer_id = top_5_customers.customer_id
GROUP BY d.country
ORDER BY top_5_customer_count DESC
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_5_customer_count bigint
1	Mexico	30	1
2	India	60	1
3	China	53	1
4	United States	36	1
5	Japan	31	1
6	Argentina	13	0
7	Armenia	1	0
8	Austria	3	0
9	Azerbaijan	2	0

STEP 3.

The first query required that we find the average of the top 5 customers, identified from the previous exercise. If we had already identified the customers by their name or customer_id we would be able to use the WHERE statement to locate the top 5 players. In this instance, we would only be required to use 1 subquery to find the sum of all top 5 player payments. The outer query would then be used to find the total average. However, I don't think we can do without any subqueries because you cannot nest aggregates, and this query requires two: the sum of all payments and the averaging of them.

For the second query, we specifically viewed the top 5 customers based on the top performing cities in each top country. If the query were different such as finding the top 5 customers overall, without the analysis of the region, we would use less subqueries. In that case, we would only use 1 subquery to find the top 5 customers in a LEFT JOIN of the customer table.

I think subqueries can be useful when you need to complete a query that requires multiple outputs in the process. For example, in the first problem you needed to find first the sum then the total average. Because you cannot nest aggregates, you would have to use a subquery.