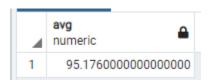
Rockbuster/postgres@PostgreSQL 13 ➤ Query Editor Query History 1 SELECT avg(sum) 2 FROM 3 $(\textbf{SELECT} \ \texttt{E.country}, \ \texttt{first_name}, \ \texttt{last_name}, \ \textbf{Sum}(\texttt{A.amount})$ 4 5 FROM payment A INNER JOIN customer B on A.customer_id = B. customer_id INNER JOIN customer 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 city IN ('Aurora', 'Pingxiang', 'Silvas', 'Dhule', 'Kurashiki', 'Xintai', 'Adoni', 'Celaya', 'Nezahualcyoti', 'Atilixco') GOULP BY a country first name last name 6 7 8 GROUP BY e.country, first_name,last_name ORDER BY SUM desc 10 11 LIMIT 5) AS top_customers 12 13



Query Editor Query History

```
SELECT E. country,
 2
            COUNT(DISTINCT A.customer_id) AS total_customer_count,
3
            COUNT(DISTINCT top_customers.country) AS top_customer_count
 4
   FROM payment A
 5 INNER JOIN customer B ON A.customer_id = B. customer_id
 6 INNER JOIN address C ON B. address_id = C. address_id
   INNER JOIN city D ON C. city_id = D. city_id
 7
   INNER JOIN country E ON D. country_id = E. country_id
   LEFT JOIN (SELECT A.customer_id,
9
                B.first_name,
10
                B.last_name,
11
12
                D.city,
                E. country,
13
14
                SUM(A.amount)
15
                FROM payment A
                INNER JOIN customer B ON A.customer_id = B.customer_id
16
                INNER JOIN address C ON B.address_id = C.address_id
17
                INNER JOIN city D ON C.city_id = D.city_id
18
19
                INNER JOIN country E ON D.country_id = E.country_id
20
                WHERE D.city IN ('Aurora',
21
                    'Cape Coral',
                    'Tanza',
22
                    'Santa Brbara dOeste',
23
24
                    'Balikesir',
25
                    'Shimonoseki',
                    'Bijapur',
26
                    'Tanauan',
27
                    'Liaocheng',
28
29
                    'Fuyu')
                GROUP BY A.customer_id,
30
31
                B.first_name,
                B.last_name,
32
33
                D.city,
34
                E. country) AS top_customers ON E.country = top_customers.country
   GROUP BY E. Country
   ORDER BY top_customer_count DESC
```

3) I know that the problems can be solved using CTE tables, which was the preferential way of doing it for me. Subqueries are helpful if you want to make sure your info is up to date. The top customers can change at any given time, so its useful to have an up-to-the-minute result table to pull from.