1)

Rockbuster/postgres@PostgreSQL 13

Query Editor | Query History

```
SELECT avg(sum)
FROM
            (SELECT E.country, first_name, last_name, Sum(A.amount)
            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 city IN ('Aurora','Pingxiang','Silvas','Dhule','Kurashiki','Xintai','Adoni','Celaya','Nezahualcyoti','Atilixco')
            GROUP BY e.country, first_name,last_name
            ORDER BY SUM desc
            LIMIT 5) AS top_customers

```

	avg numeric
1	95.1760000000000000

2)

Query Editor | Query History

```
SELECT E. country,
        COUNT(DISTINCT A.customer_id) AS total_customer_count,
        COUNT(DISTINCT top_customers.country) AS top_customer_count
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
LEFT JOIN (SELECT A.customer_id,
            B.first_name,
            B.last_name,
            D.city,
            E. country,
            SUM(A.amount)
            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 D.city IN ('Aurora',
                'Cape Coral',
                'Tanza',
                'Santa Brbara dOeste',
                'Balikesir',
                'Shimonoseki',
                'Bijapur',
                'Tanauan',
                'Liaocheng',
                'Fuyu')
            GROUP BY A.customer_id,
            B.first_name,
            B.last_name,
            D.city,
            E. country) AS top_customers ON E.country = top_customers.country
GROUP BY E. Country
ORDER BY top_customer_count DESC
```

Data Output | Explain | Messages | Notifications

	country character varying (50)	total_customer_count bigint	top_customer_count bigint
1	United States	36	1
2	Turkey	15	1
3	Brazil	28	1
4	China	53	1
5	Japan	31	1
6	Philippines	20	1
7	India	60	1
8	Austria	3	0
9	Azerbaijan	2	0
10	Bahrain	1	0
11	Bangladesh	3	0
12	Belarus	2	0
13	Bolivia	2	0
14	Brunei	1	0
15	Bulgaria	2	0
16	Cambodia	2	0
17	Cameroon	2	0
18	Canada	5	0
19	Chad	1	0
20	Chile	3	0
21	Colombia	6	0
22	Congo, The Democratic Republic of the	2	0
23	Czech Republic	1	0
24	Dominican Republic	3	0
25	Ecuador	3	0
26	Egypt	6	0
27	Estonia	1	0
28	Ethiopia	1	0
29	Faroe Islands	1	0
30	Finland	1	0
31	France	4	0
32	French Guiana	1	0
33	French Polynesia	2	0
34	Gambia	1	0
35	Germany	7	0

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.