1)

Query Editor | Query History

```
WITH best_customers_cte (first_name, last_name, city, country) AS (SELECT first_name, last_name, city, country, sum(total_amount)
FROM (SELECT A. customer_id,
        B. first_name,
        B. last_name,
        D. city,
        E. country,
        SUM (A.amount) AS total_amount
FROM payment A
LEFT JOIN customer B on A.customer_id = B. customer_id
LEFT JOIN address C ON B.address_id = C.address_id
LEFT JOIN city D ON C.city_id = D.city_id
LEFT 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 A.customer_id,B.first_name,B.last_name,D.city,E.country
ORDER BY AVG(amount) DESC
LIMIT 5) AS total_amount_paid
GROUP BY first_name, last_name, city,total_amount_paid.country, total_amount_paid.*
ORDER BY total_amount_paid DESC)
SELECT AVG(sum)
FROM best_customers_cte
```

	avg numeric
1	95.1760000000000000

2)

Rockbuster/postgres@PostgreSQL 13

Query Editor | Query History

```
WITH best_customers_cte AS
        (SELECT A. customer_id,
        B. first_name,
        B. last_name,
        D. city,
        E. country,
        SUM (A.amount) AS total_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 A.customer_id,B.first_name,B.last_name,D.city,E.country
ORDER BY SUM(amount) DESC
LIMIT 5)
SELECT D.country,
COUNT (DISTINCT A.customer_id) AS all_customer_count,
 COUNT (DISTINCT best_customers_cte.customer_id) 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 best_customers_cte ON D.country =
best_customers_cte.country
GROUP BY D.country
ORDER BY all_customer_count DESC;
```

country	all_customer_count	top_customer_count
India	60	1
China	53	1
United States	36	1
Japan	31	1
Mexico	30	1
Brazil	28	0
Russian Federation	28	0
Philippines	20	0
Turkey	15	0
Indonesia	14	0
Nigeria	13	0
Argentina	13	0
South Africa	11	0
Taiwan	10	0
United Kingdom	9	0
Iran	8	0
Poland	8	0
Italy	7	0
Germany	7	0
Venezuela	7	0
Egypt	6	0
Ukraine	6	0
Vietnam	6	0
Colombia	6	0
Spain	5	0
Canada	5	0
Saudi Arabia	5	0
Netherlands	5	0
Pakistan	5	0
South Korea	5	0
Peru	4	0
France	4	0
Yemen	4	0
Israel	4	0
Algeria	3	0
Switzerland	3	0
Tanzania	3	0
United Arab Emirates	3	0
Morocco	3	0
Bangladesh	3	0
Chile	3	0
Thailand	3	0

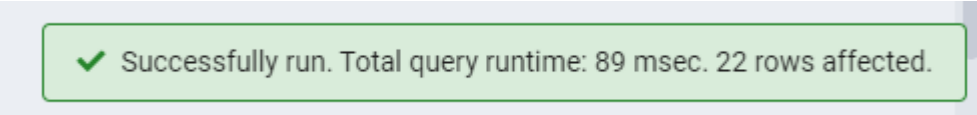
Malaysia	3	0
Austria	3	0
Paraguay	3	0
Mozambique	3	0
Ecuador	3	0
Dominican Republic	3	0
Sudan	2	0
Bolivia	2	0
Greece	2	0
Belarus	2	0
Bulgaria	2	0
Yugoslavia	2	0
Cambodia	2	0
Cameroon	2	0
Romania	2	0
Puerto Rico	2	0
Kazakstan	2	0
Kenya	2	0
Angola	2	0
Latvia	2	0
Azerbaijan	2	0
Congo, The Democratic Republic of the	2	0
Oman	2	0
Myanmar	2	0
French Polynesia	2	0
Zambia	1	0
American Samoa	1	0
Anguilla	1	0
Armenia	1	0
Bahrain	1	0
Brunei	1	0
Chad	1	0
Czech Republic	1	0
Estonia	1	0
Ethiopia	1	0
Faroe Islands	1	0
Finland	1	0
French Guiana	1	0
Gambia	1	0
Greenland	1	0
Holy See (Vatican City State)	1	0
Hong Kong	1	0

Hungary	1	0
Iraq	1	0
Kuwait	1	0
Liechtenstein	1	0
Lithuania	1	0
Madagascar	1	0
Malawi	1	0
Moldova	1	0
Nauru	1	0
Nepal	1	0
New Zealand	1	0
North Korea	1	0
Runion	1	0
Saint Vincent and the Grenadines	1	0
Senegal	1	0
Slovakia	1	0
Sri Lanka	1	0
Sweden	1	0
Tonga	1	0
Tunisia	1	0
Turkmenistan	1	0
Tuvalu	1	0
Virgin Islands, U.S.	1	0
Afghanistan	1	0

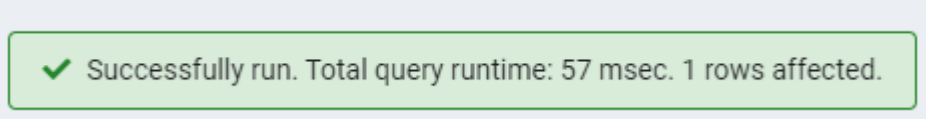
For both of these, it was a matter of making the subquery from 4.8 the CTE table instead of making it a subquery. This was easier for me personally as it meant I just needed to put the subquery at the beginning vs trying to figure out if I needed to put it in the “select”, “from”, or “where” column.

2)

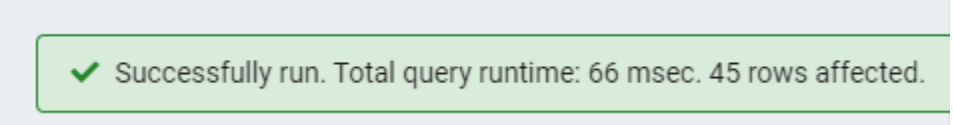
Top Customers with Subqueries



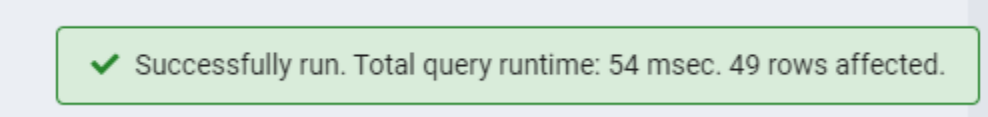
Top Customers with CTE



All Countries with Subqueries



All Countries with CTE



I am surprised at how much faster CTE was able to take care of the results. I can also say that I much prefer CTE as opposed to subqueries, but I'm sure my opinions will change as my career progresses.

3) My biggest challenge was actually in the previous chapter. I had a much more difficult time with subqueries than I did with CTE. CTE made a lot more sense cause you can give the tables aliases and then use those aliases in the outer query vs trying to copy and paste everything and have it all line up.