

Step 1

Query







Query History

```
1 WITH average_total_amount_paid_cte(customer_ID, first_name, last_name, city, country, total_amount_paid)
2   AS(SELECT a.customer_ID, a.first_name, a.last_name, c.city, d.country, SUM(e.amount)AS total_amount_paid
3 FROM customer a
4 INNER JOIN address B ON a.address_id = b.address_id
5 INNER JOIN city c ON b.city_id = c.city_id
6 INNER JOIN country d ON c.country_id = d.country_id
7 INNER JOIN payment e ON a.customer_id = e.customer_id
8 WHERE c.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulia)','Kurashiki',
9                 'Pingxiang','Sivas','Celaya','So Leopoldo')
10 GROUP BY a.customer_ID, c.city, d.country
11 ORDER BY SUM(e.amount) DESC
12 LIMIT 5)
13
14
15 SELECT AVG(total_amount_paid) AS average_total_amount_paid
16 FROM average_total_amount_paid_cte
```

Data output

Messages

Notifications



	average_total_amount_paid numeric	
1	107.3540000000000000	

Changing this subquery to a CTE was not difficult. A lot of the syntax stayed the same, and I only created one CTE, so it easily follows the syntax of the original subquery. I created the WITH statement first, and then added the SELECT statement at line 15.

Query	Query History
1	WITH top_customer_count_cte(amount, customer_ID, first_name, last_name, city, country, total_amount_paid)
2	AS (SELECT e.amount, a.customer_ID, a.first_name, a.last_name, c.city,
3	d.country, SUM(amount) AS total_amount_paid
4	FROM customer a
5	LEFT JOIN address B ON a.address_id = b.address_id
6	LEFT JOIN city c ON b.city_id = c.city_id
7	LEFT JOIN country d ON c.country_id = d.country_id
8	LEFT JOIN payment e ON a.customer_id = e.customer_id
9	WHERE city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulia)','Kurashiki',
10	'Pingxiang','Sivas','Celaya','So Leopoldo')
11	GROUP BY e.amount, a.customer_ID, a.first_name, a.last_name, c.city, d.country
12	ORDER BY SUM(amount) DESC LIMIT 5),
13	
14	customer_count_cte AS (SELECT d.country, COUNT(DISTINCT a.customer_id) AS all_customer_count,
15	COUNT(DISTINCT d.country) AS top_customer_count
16	FROM customer A
17	INNER JOIN address B ON a.address_id = b.address_id
18	INNER JOIN city C ON b.city_id = c.city_id
19	INNER JOIN country D ON c.country_id = d.country_id
20	INNER JOIN payment E ON a.customer_id = e.customer_id
21	GROUP BY d.country)
22	
23	SELECT d.country, COUNT(DISTINCT a.customer_ID) as all_customer_count,
24	COUNT(DISTINCT top_customer_count_cte.customer_ID) AS top_customer_count
25	FROM customer A
26	INNER JOIN address B ON a.address_id = b.address_id
27	INNER JOIN city C ON b.city_id = c.city_id
28	INNER JOIN country D ON c.country_id = d.country_id
29	INNER JOIN payment E ON a.customer_id = e.customer_id
30	LEFT JOIN top_customer_count_cte ON d.country= top_customer_count_cte.country
31	GROUP BY d.country
32	ORDER BY all_customer_count DESC LIMIT 5

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	1
2	China	53	0
3	United States	36	1
4	Japan	31	0
5	Mexico	30	2

Here, I created two CTEs, so the process was more difficult. I found it helpful to start with the syntax from the original subquery, and then change the first two lines of the CTE depending on what I want them to do. Then, I rewrote the SELECT statement from line 23.

Step 2

I thought that the CTEs would perform better because they are basically functions that you can call as you wish, and that allows you to not have to rewrite as much code if you want to make changes. Every query gave me a different runtime each time that I ran it. On average however, the CTEs had lower runtimes than the subqueries.

Step 3

At first, I tried to write the CTEs from scratch, which resulted in a lot of mistakes. For example, I didn't alias properly. When I copied the subqueries into the editor and then made changes, it was much easier. Another thing that helped was separating the CTEs from the main statement while I was working on them. Breaking the code down into sections helps it feel less overwhelming, and I can just focus on one piece at a time.