



JANUARY 9, 2026

3.9 COMMON TABLE EXPRESSIONS

JANELLA VALENCIA

Table of Contents

<u>STEP 1: REWRITE YOUR QUERIES FROM STEP 1 AND STEP 2 IN 3.8 TASK.....</u>	2
3.8 STEP 1 CTE	2
STEP 1 CTE OUTPUT	2
3.8 STEP 2 CTE	3
STEP 2 CTE OUTPUT	3
EXPLANATION.....	3
<u>STEP 2: COST AND TIME</u>	4
COST COMPARISON	4
EXPLANATION.....	4
<u>STEP 3: CHALLENGES</u>	4

Step 1: Rewrite your queries from Step 1 and Step 2 in 3.8 Task

3.8 Step 1 CTE

```
WITH Top5Customers AS
(SELECT B.customer_id,
    B.first_name,
    B.last_name,
    E.country,
    D.city,
    SUM(A.amount) AS total_amount_paid
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', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
    AND E.country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT AVG(total_amount_paid) AS Average_amount_paid
FROM Top5Customers
```

Step 1 CTE Output

	average_amount_paid
	numeric
1	107.354000000000000000

3.8 Step 2 CTE

```

WITH Top5WithinCountries AS
(SELECT B.customer_id, B.first_name, B.last_name, E.country, D.city,
SUM(A.amount) AS total_amount_paid
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', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
AND E.country IN ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT cnt1.country, |
COUNT(DISTINCT cust1.customer_id) AS all_customer_count,
COUNT(DISTINCT Top5WithinCountries.customer_id) AS top_customer_count
FROM customer AS cust1
INNER JOIN address AS addr1 ON cust1.address_id=addr1.address_id
INNER JOIN city AS cty1 ON addr1.city_id=cty1.city_id
INNER JOIN country AS cnt1 ON cty1.country_id=cnt1.country_id
LEFT JOIN Top5WithinCountries ON Top5WithinCountries.country=cnt1.country
GROUP BY cnt1.country
ORDER BY top_customer_count DESC, all_customer_count DESC
LIMIT 5;

```

Step 2 CTE Output

	country character varying (50) 	all_customer_count bigint 	top_customer_count bigint 
1	Mexico	30	2
2	India	60	1
3	United States	36	1
4	Turkey	15	1
5	China	53	0

Explanation

I started by giving my CTE a name, I choose **Top5Customers** (since this is what the SQL calculates). I used my subquery from exercise 3.8 and inserted it after my **AS** statement. After my subquery parenthesis I added the command to calculate the **AVG** and assigned it an alias of **average_amount_paid** and for my **FROM** I used **Top5Customers** (the CTE). We wanted to know the average paid by the top 5 customers.

I then built on the previous statement to find the top 5 customers that were identified in step 1, based in which countries. I named my CTE **Top5withincountries** and inserted my subquery as **AS** statement. I moved the COUNT for alias **all_customers_count** and **top_customer_count** after the subquery. I needed to join my new CTE (

Top5WithinCountries) with my statement with a **LEFT JOIN**. I **GROUP BY** country and **ORDER BY top_customer_count DESC, all_customers_count DESC**. Lastly the **LIMIT** clause to limit the results to top 5 customers.

Step 2: Cost and Time

Which approach do you think will perform better and why?

- I think the CTE method may perform better because we're distinguishing the CTE in the beginning of the statement and then using the CTE name later on in the LEFT JOIN. Rather than, having the whole subquery embedded within the LEFT JOIN like it was in 3.8 step 2 exercise. I think the CTE method also makes it easier to understand, higher readability, and easier updating when needed. For this same reason I think the cost will be higher for the subquery method.

Cost Comparison

Query	Subquery	CTE
Step 1	Cost: 24.65/ Time: 0.510 ms	Cost: 24.65/ Time: 1.017
Step 2	Cost: 127.04/ Time: 5.203	Cost: 127.04/ Time: 5.451

Explanation

Did the results surprise you?

- I notice the difference is in the time not in the cost. Yes, I was surprised to see that in both cases the time is quicker for Subqueries. Maybe, creating a CTE view and then the main statement could take longer to produce a result. In the Subquery method, the subquery was embedded in the outer statement.

Step 3: Challenges

Some of the challenges I faced was wrapping my head around how to organize my statement to get the CTE and correct output. For Step 1 CTE it was a little more straight forward since I wasn't using the COUNT aggregation yet. When it came to Step 2 CTE it felt a lot more complicated, I had to place close attention to replace some alias with the CTE table name, and also joining it to the rest of the statement correctly. Remembering to Order both COUNTS was also tricky.