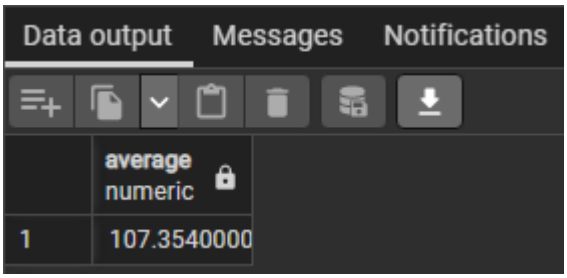
3.9 Common Table Expressions

1. **Step 1: Answer the business questions from step 1 and 2 of task 3.8 using CTEs**

   - Copy the Rewrite your queries from steps 1 and 2 of task 3.8 as CTEs.
   - Copy-paste your CTEs and their outputs into your answers document.
   - Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.

```
WITH total_amount_paid (payment) AS
   (SELECT SUM(pay.amount) AS payment
   FROM customer as cu
   INNER JOIN address AS ad ON cu.address_id = ad.address_id
   INNER JOIN city AS ci ON ad.city_id = ci.city_id
   INNER JOIN country AS co ON ci.country_id = co.country_id
   INNER JOIN payment AS pay ON cu.customer_id = pay.customer_id
   WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
            'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
   GROUP BY cu.customer_id
   ORDER BY payment DESC
   LIMIT 5)

SELECT AVG(payment) AS average
FROM total_amount_paid;
```



```
WITH top_5_customers (customer_id, first_name, last_name, city, country, total_amount_paid) AS
   (SELECT cu.customer_id,
         cu.first_name,
         cu.last_name,
         ci.city,
         co.country,
         SUM(pay.amount) AS total_amount_paid
   FROM customer as cu
   INNER JOIN address AS ad ON cu.address_id = ad.address_id
   INNER JOIN city AS ci ON ad.city_id = ci.city_id
   INNER JOIN country AS co ON ci.country_id = co.country_id
   INNER JOIN payment AS pay ON cu.customer_id = pay.customer_id
   WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki',
            'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
   GROUP BY cu.customer_id, first_name, last_name, city, country
   ORDER BY total_amount_paid DESC
   LIMIT 5)


SELECT co.country,
     COUNT(DISTINCT cu.customer_id) AS all_customer_count,
     COUNT(DISTINCT co.country) AS top_customer_count
FROM country AS co
INNER JOIN city AS ci ON co.country_id = ci.country_id
INNER JOIN address AS ad ON ci.city_id = ad.city_id
INNER JOIN customer AS cu ON ad.address_id = cu.address_id
LEFT JOIN top_5_customers ON co.country = top_5_customers.country
GROUP BY co.country, top_5_customers
ORDER BY all_customer_count DESC;
```

Data output | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1
6	Mexico	30	1
7	Brazil	28	1
8	Russian Federation	28	1
9	Philippines	20	1
10	Turkey	15	1

Total rows: 109 of 109 | Query complete 00:00:00.086

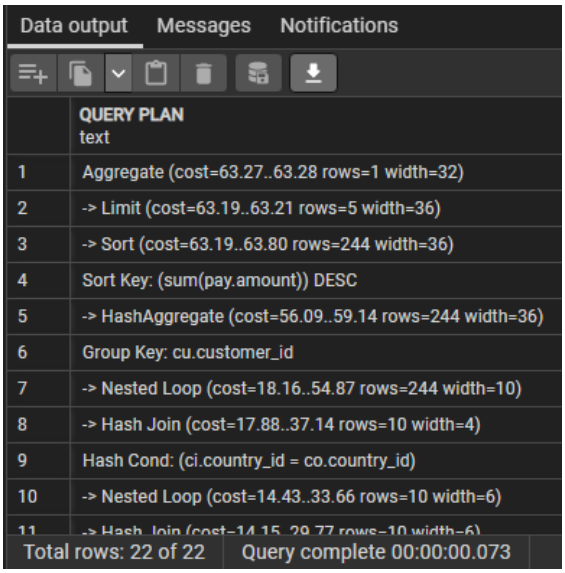
As the tables are already located from the previous task, I proceed with only converting it to CTE. It is easier for me to think from inside to outside, meaning I define firstly the deepest query (inside WITH clause) and then close it with parenthesis. Afterwards, think about the outer layer which is the main statement.

2. **Step 2: Compare the performance of your CTEs and subqueries.**

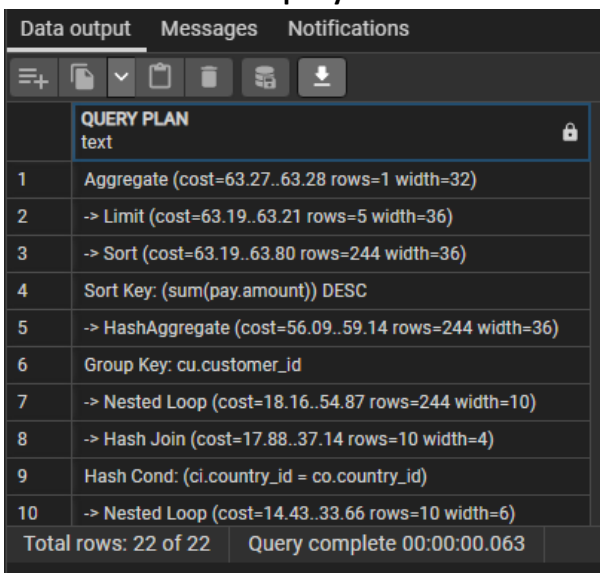
- Which approach do you think will perform better and why?
  CTE is indeed better in terms of readability and not cluttering which makes also easier for debugging. For me personally, I could think more systematically when using CTE compared to subquery.
- Compare the costs of all the queries by creating query plans for each one.

Question 1

CTE

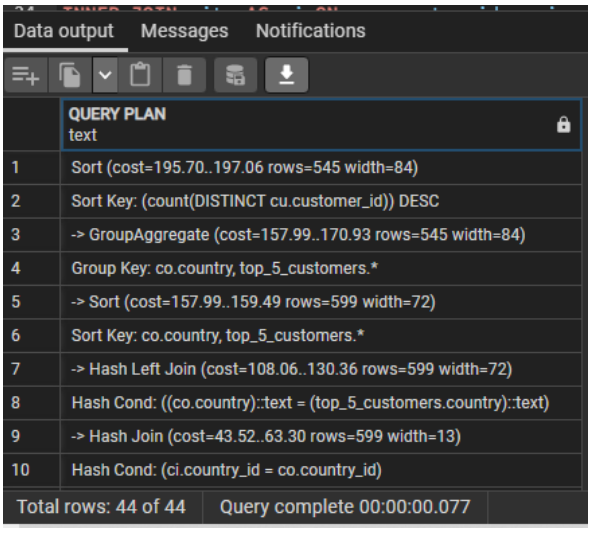


Subquery

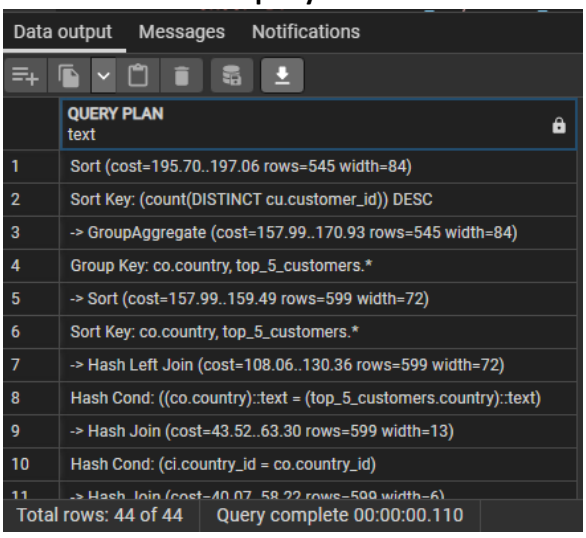


Question 2

CTE



Subquery



- The EXPLAIN command gives you an estimated cost. To find out the actual speed of your queries, run them in pgAdmin 4. After each query has been run, a pop-up window will display its speed in milliseconds.
- Did the results surprise you? Write a few sentences to explain your answer.
  Surprisingly, the cost for both CTE and subquery are the same. However, that is not always the case with the actual speed. Actual speed using subquery is faster for the question 1 and slower for the question 2 in comparison to the CTE. Not to mention, the first run takes always longer. As the queries cached, it actual time is reduced.