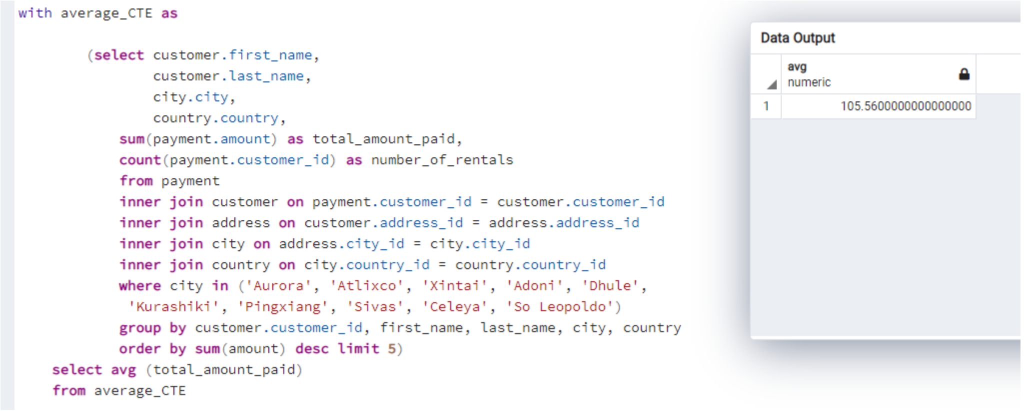
3.9 Common Table Expressions

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

1. Rewrite your queries from steps 1 and 2 of task 3.8 as CTEs.
2. Copy-paste your CTEs and their outputs into your answers document.

Average:



```
with average_CTE as

        (select customer.first_name,
                customer.last_name,
                city.city,
                country.country,
            sum(payment.amount) as total_amount_paid,
            count(payment.customer_id) as number_of_rentals
            from payment
            inner join customer on payment.customer_id = customer.customer_id
            inner join address on customer.address_id = address.address_id
            inner join city on address.city_id = city.city_id
            inner join country on city.country_id = country.country_id
            where city in ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule',
             'Kurashiki', 'Pingxiang', 'Sivas', 'Celeya', 'So Leopoldo')
            group by customer.customer_id, first_name, last_name, city, country
            order by sum(amount) desc limit 5)
    select avg (total_amount_paid)
    from average_CTE
```

Data Output

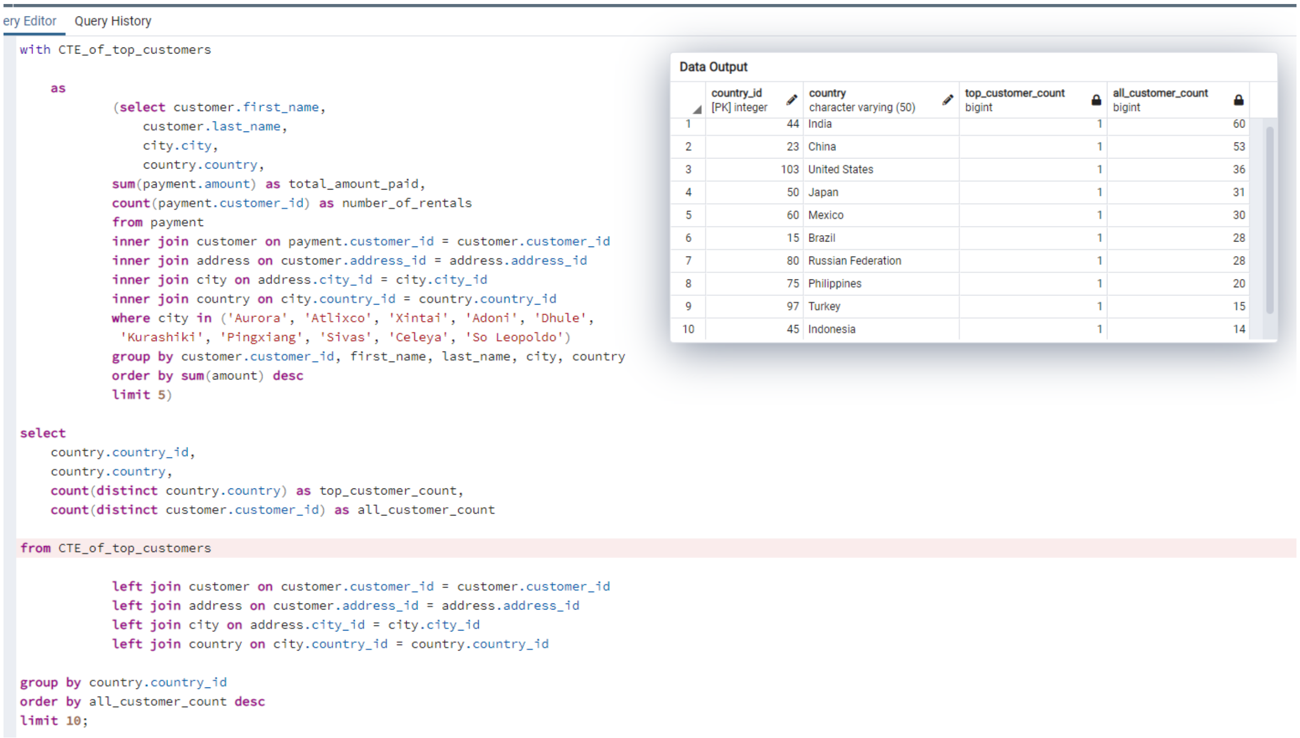
	avg numeric
1	105.5600000000000000

```
with average_CTE as

            (select customer.first_name,
                    customer.last_name,
                    city.city,
                    country.country,
            sum(payment.amount) as total_amount_paid,
            count(payment.customer_id) as number_of_rentals
            from payment
            inner join customer on payment.customer_id = customer.customer_id
            inner join address on customer.address_id = address.address_id
            inner join city on address.city_id = city.city_id
            inner join country on city.country_id = country.country_id
            where city in ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule',
             'Kurashiki', 'Pingxiang', 'Sivas', 'Celeya', 'So Leopoldo')
```

```
                    group by customer.customer_id, first_name, last_name, city, country
                    order by sum(amount) desc limit 5)
        select avg (total_amount_paid)
        from average_CTE
```

Top Customers:



```
with CTE_of_top_customers

    as
            (select customer.first_name,
                customer.last_name,
                city.city,
                country.country,
            sum(payment.amount) as total_amount_paid,
            count(payment.customer_id) as number_of_rentals
            from payment
            inner join customer on payment.customer_id = customer.customer_id
            inner join address on customer.address_id = address.address_id
            inner join city on address.city_id = city.city_id
            inner join country on city.country_id = country.country_id
            where city in ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule',
             'Kurashiki', 'Pingxiang', 'Sivas', 'Celeya', 'So Leopoldo')
            group by customer.customer_id, first_name, last_name, city, country
            order by sum(amount) desc
            limit 5)

select
    country.country_id,
    country.country,
    count(distinct country.country) as top_customer_count,
    count(distinct customer.customer_id) as all_customer_count

from CTE_of_top_customers

            left join customer on customer.customer_id = customer.customer_id
            left join address on customer.address_id = address.address_id
            left join city on address.city_id = city.city_id
            left join country on city.country_id = country.country_id

group by country.country_id
order by all_customer_count desc
limit 10;
```

	country_id [PK] integer	country character varying (50)	top_customer_count bigint	all_customer_count bigint
1	44	India	1	60
2	23	China	1	53
3	103	United States	1	36
4	50	Japan	1	31
5	60	Mexico	1	30
6	15	Brazil	1	28
7	80	Russian Federation	1	28
8	75	Philippines	1	20
9	97	Turkey	1	15
10	45	Indonesia	1	14

```
with CTE_of_top_customers
        as
                    (select customer.first_name,
                            customer.last_name,
                            city.city,
                            country.country,
                    sum(payment.amount) as total_amount_paid,
                    count(payment.customer_id) as number_of_rentals
                    from payment
                    inner join customer on payment.customer_id = customer.customer_id
```

```
            inner join address on customer.address_id = address.address_id
            inner join city on address.city_id = city.city_id
            inner join country on city.country_id = country.country_id
            where city in ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule',
             'Kurashiki', 'Pingxiang', 'Sivas', 'Celeya', 'So Leopoldo')
            group by customer.customer_id, first_name, last_name, city, country
            order by sum(amount) desc
            limit 5)

select
    country.country_id,
    country.country,
    count(distinct country.country) as top_customer_count,
    count(distinct customer.customer_id) as all_customer_count

from CTE_of_top_customers

            left join customer on customer.customer_id = customer.customer_id
            left join address on customer.address_id = address.address_id
            left join city on address.city_id = city.city_id
            left join country on city.country_id = country.country_id

group by country.country_id
order by all_customer_count desc
limit 10;
```

3. Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.

a. Since this process is about rearranging an existing subquery into a CTE, the procedure is extremely similar. In fact, the only difference is the 'with cte as' statement at the beginning and the 'from cte' statement at the end. Everything else exactly the same as the subquery process. In terms of doing a CTE de novo, the building procedure is extremely similar to building a subquery.

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

1. Which approach do you think will perform better and why?

2. Compare the costs of all the queries by creating query plans for each one.

3. 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.

4. Did the results surprise you? Write a few sentences to explain your answer.

    a. Subqueries run a tad faster. In the top customers query and cte, the former runs in 208ms and the latter in 250ms. As far as "better" – I cannot tell any advantage to one over the other. Both have an identical output and each is as complicated/simple as the other.

Step 3:

Write 1 to 2 paragraphs on the challenges you faced when replacing your subqueries with CTEs.

I personally don't see any benefit in replacing one for the other. Subqueries are a bit faster and the same basic structure is applied to both with minor exceptions. At best, I find the CTE to be somewhat neater in that there is the 'with cte as' followed by the inner query followed by the outer query with all joins, groupings, and orders where the subquery approach breaks up the select and from before the inner query and then the joins, groupings and orders follow. But on the complexity of constructing either a CTE or a subquery, I find that there's no significant difference between the two and it comes down, ultimately, to preference. If the ratios speed differences continue to expand as the dataset one analyzes becomes increasingly larger and more complex, it may simply be a case of computing power to stick to subqueries. At the end of the day, however, I don't think the company or people one would report to will really care *how* you got the answer as long as you got the answer to the question they were looking for (Who's the top paying customer? We want to reward them!); as long as whatever you're writing – codewise – makes coherent sense and someone can follow your logic.

Step 4:

Save your "Answers 3.9" document as a PDF and upload it here for your tutor to review.