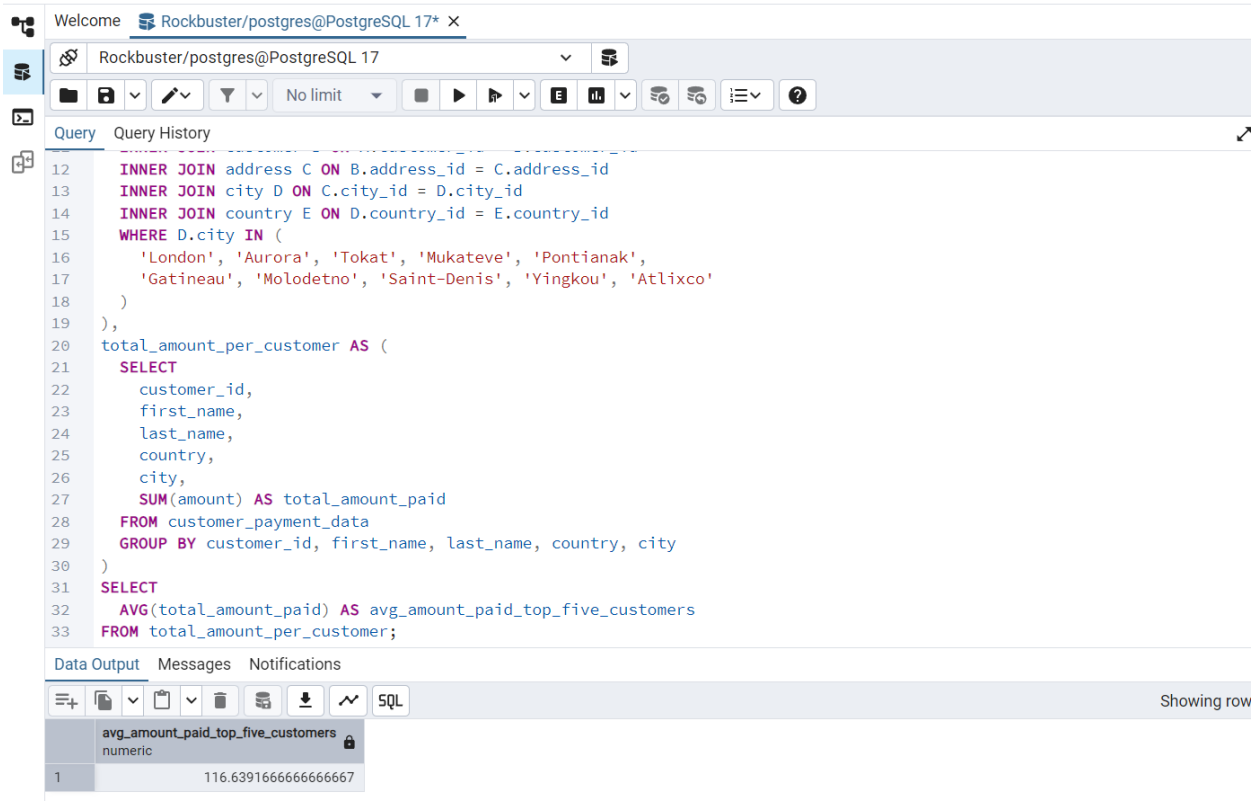
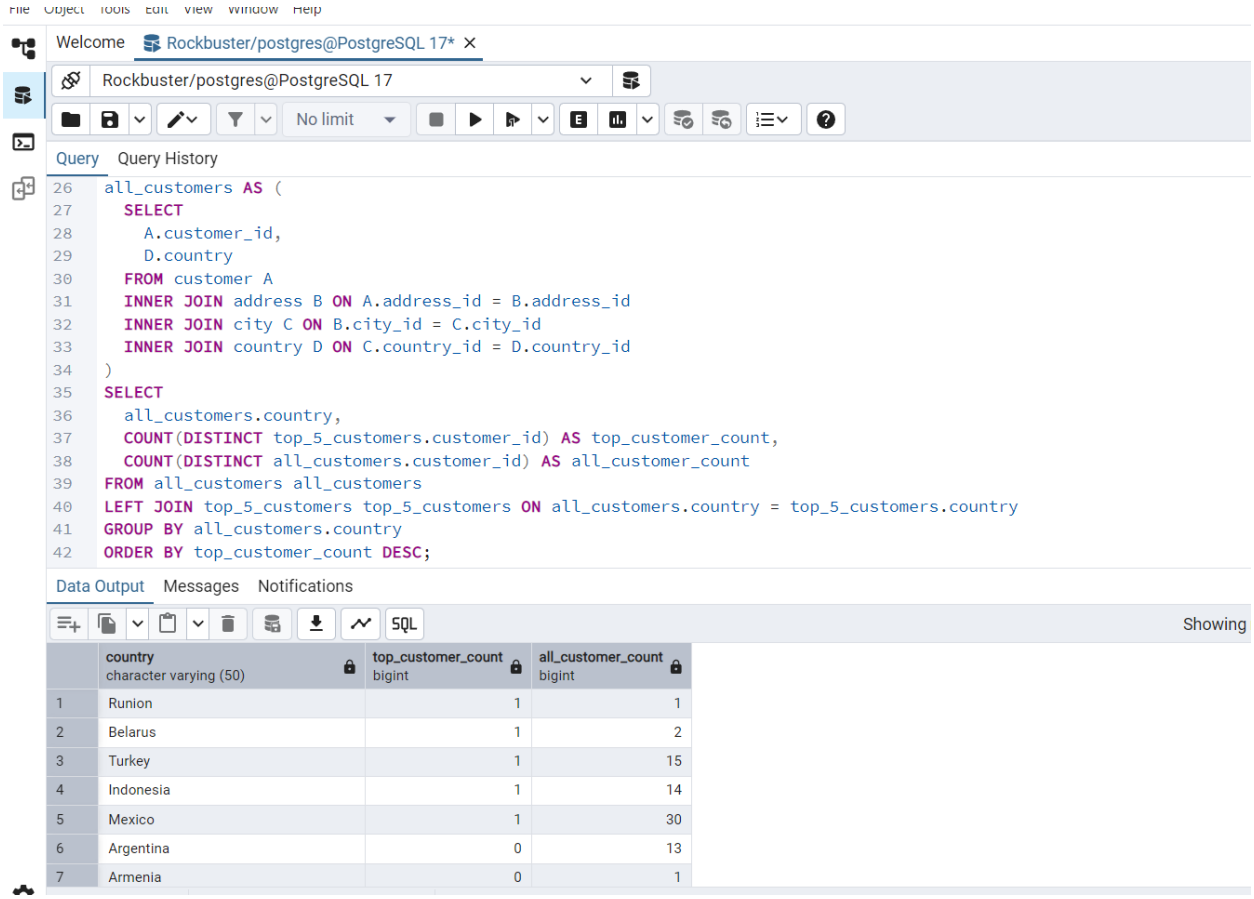
Find the average amount paid by the top 5 customers.




```
WITH customer_payment_data AS (
  SELECT
    A.amount,
    B.customer_id,
    B.first_name,
    B.last_name,
    C.address_id,
    D.city,
    E.country
  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 (    'London', 'Aurora', 'Tokat', 'Mukateve', 'Pontianak',
    'Gatineau', 'Molodetno', 'Saint-Denis', 'Yingkou', 'Atlixco' )),
total_amount_per_customer AS (
  SELECT
    customer_id,
    first_name,
    last_name,
    country,
    city,
    SUM(amount) AS total_amount_paid
  FROM customer_payment_data
  GROUP BY customer_id, first_name, last_name, country, city)
SELECT
  AVG(total_amount_paid) AS avg_amount_paid_top_five_customers
FROM total_amount_per_customer;
```

Find out how many of the top 5 customers you identified in step 1 are based within each country.




```
WITH customer_payment_data 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 (
```

```
    'London', 'Aurora', 'Tokat', 'Mukateve', 'Pontianak',
    'Gatineau', 'Molodetno', 'Saint-Denis', 'Yingkou', 'Atlixco'
  )
  GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city
),
top_5_customers AS (
  SELECT *
  FROM customer_payment_data
  ORDER BY total_amount_paid DESC
  LIMIT 5
),
all_customers AS (
  SELECT
    A.customer_id,
    D.country
  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
)
SELECT
  all_customers.country,
  COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count,
  COUNT(DISTINCT all_customers.customer_id) AS all_customer_count
FROM all_customers all_customers
LEFT JOIN top_5_customers top_5_customers ON all_customers.country = top_5_customers.country
GROUP BY all_customers.country
ORDER BY top_customer_count DESC;
```

One of the main challenges I faced was understanding how to logically break down the original query into modular, named blocks. Subqueries often work seamlessly within a query, but when converting them into CTEs, I had to ensure that each step produced the necessary intermediate data to be used in the next step. Another challenge was ensuring that the CTEs executed in the correct order and maintained the original query's logic.