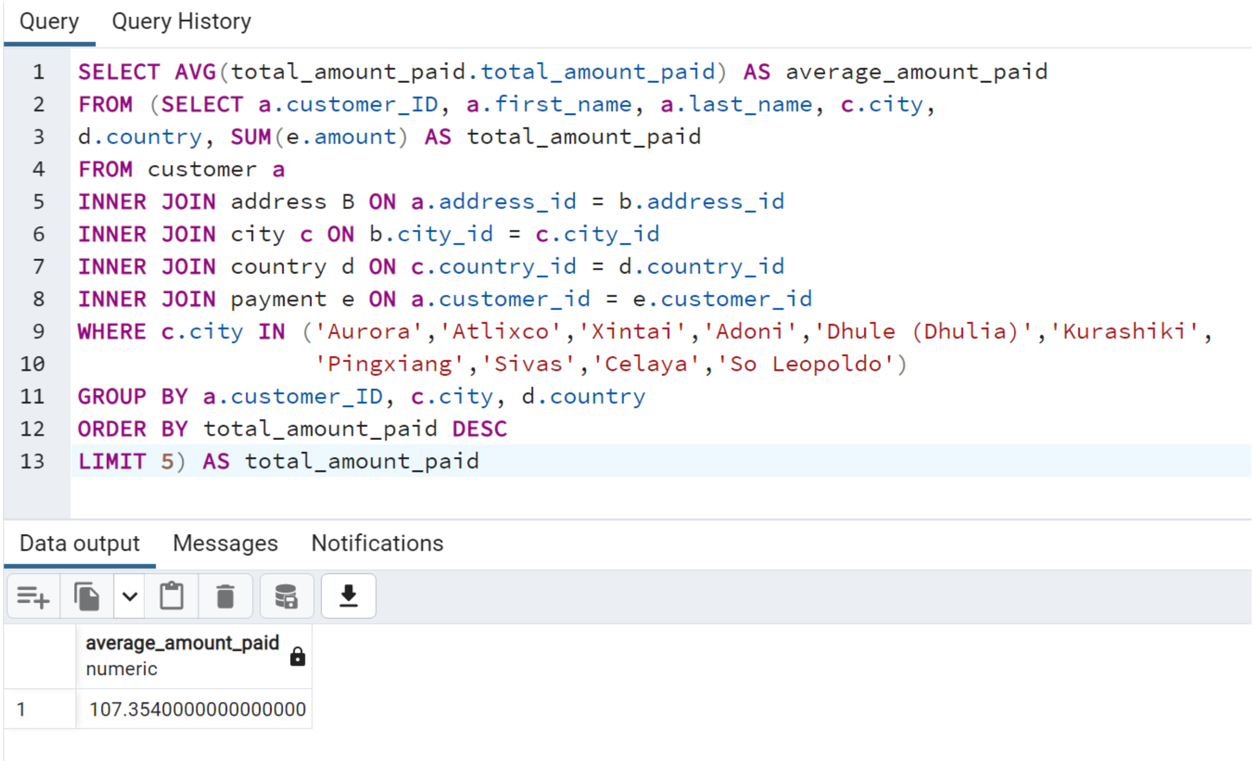
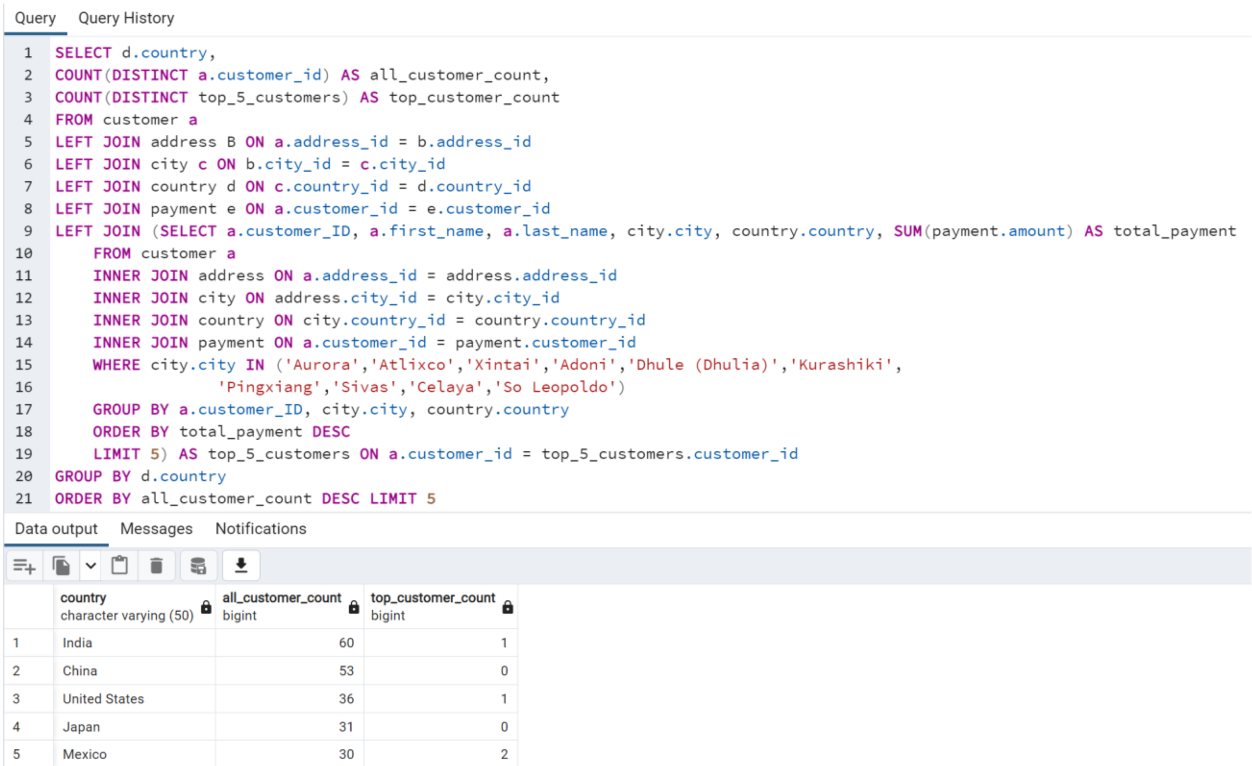
Step 1



```
SELECT AVG(total_amount_paid.total_amount_paid) AS average_amount_paid
FROM (SELECT a.customer_ID, a.first_name, a.last_name, c.city,
d.country, SUM(e.amount) AS total_amount_paid
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
INNER JOIN payment e ON a.customer_id = e.customer_id
WHERE c.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulia)','Kurashiki',
                 'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY a.customer_ID, c.city, d.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

	average_amount_paid numeric
1	107.3540000000000000

Step 2



```
SELECT d.country,
COUNT(DISTINCT a.customer_id) AS all_customer_count,
COUNT(DISTINCT top_5_customers) AS top_customer_count
FROM customer a
LEFT JOIN address B ON a.address_id = b.address_id
LEFT JOIN city c ON b.city_id = c.city_id
LEFT JOIN country d ON c.country_id = d.country_id
LEFT JOIN payment e ON a.customer_id = e.customer_id
LEFT JOIN (SELECT a.customer_ID, a.first_name, a.last_name, city.city, country.country, SUM(payment.amount) AS total_payment
    FROM customer a
    INNER JOIN address ON a.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
    INNER JOIN payment ON a.customer_id = payment.customer_id
    WHERE city.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulia)','Kurashiki',
                 'Pingxiang','Sivas','Celaya','So Leopoldo')
    GROUP BY a.customer_ID, city.city, country.country
    ORDER BY total_payment DESC
    LIMIT 5) AS top_5_customers ON a.customer_id = top_5_customers.customer_id
GROUP BY d.country
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

Step 3

Step 1 could have been done without a subquery using aggregate functions. Step 2 could not have been done without a subquery because we needed results from a different table. Subqueries are useful in this case because they allow you to retrieve results from another table without rewriting your inner query.