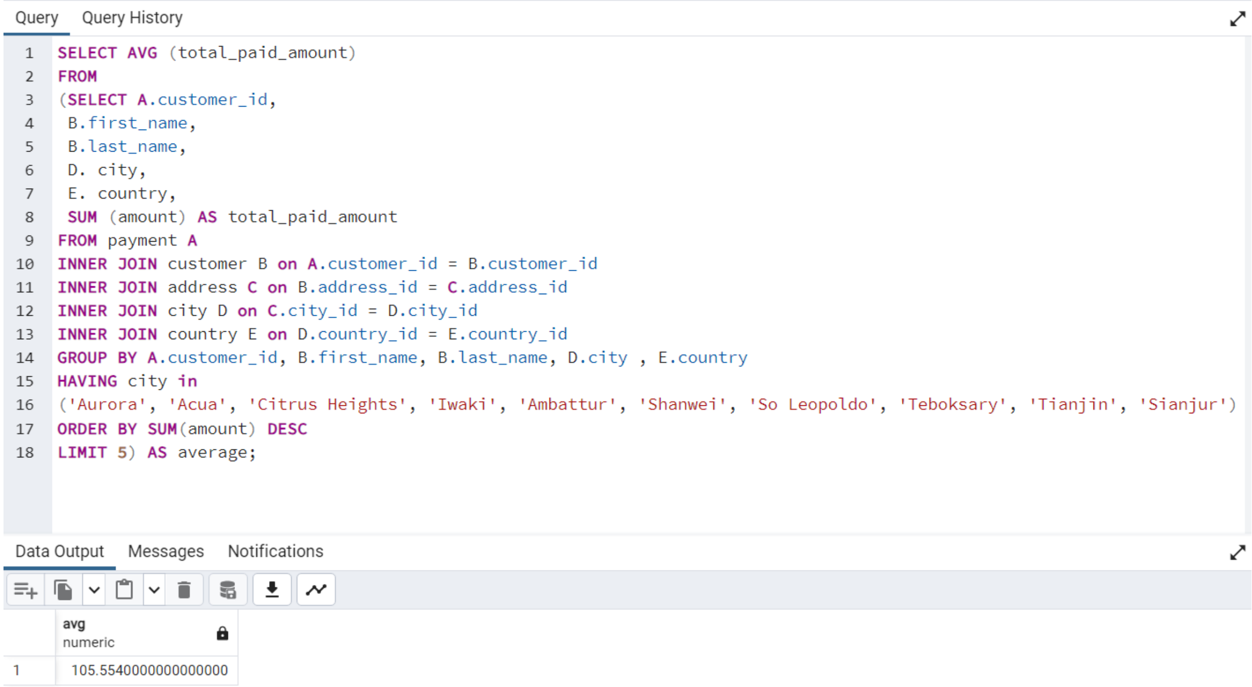
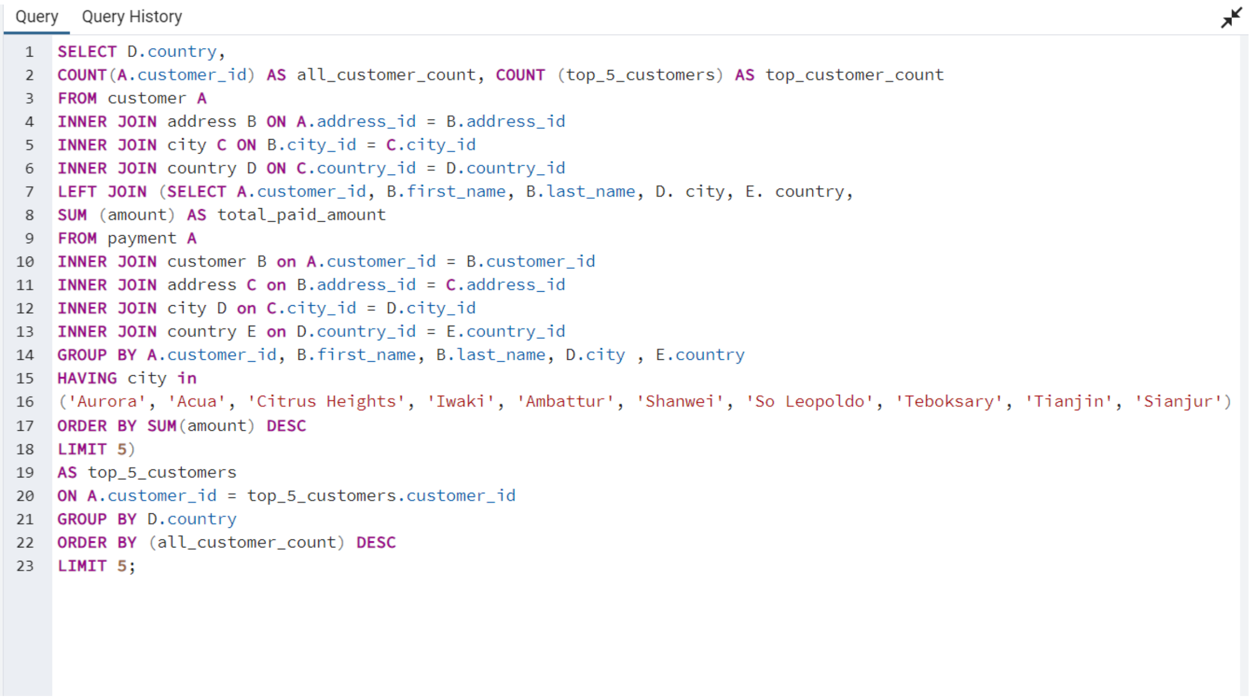
1.a



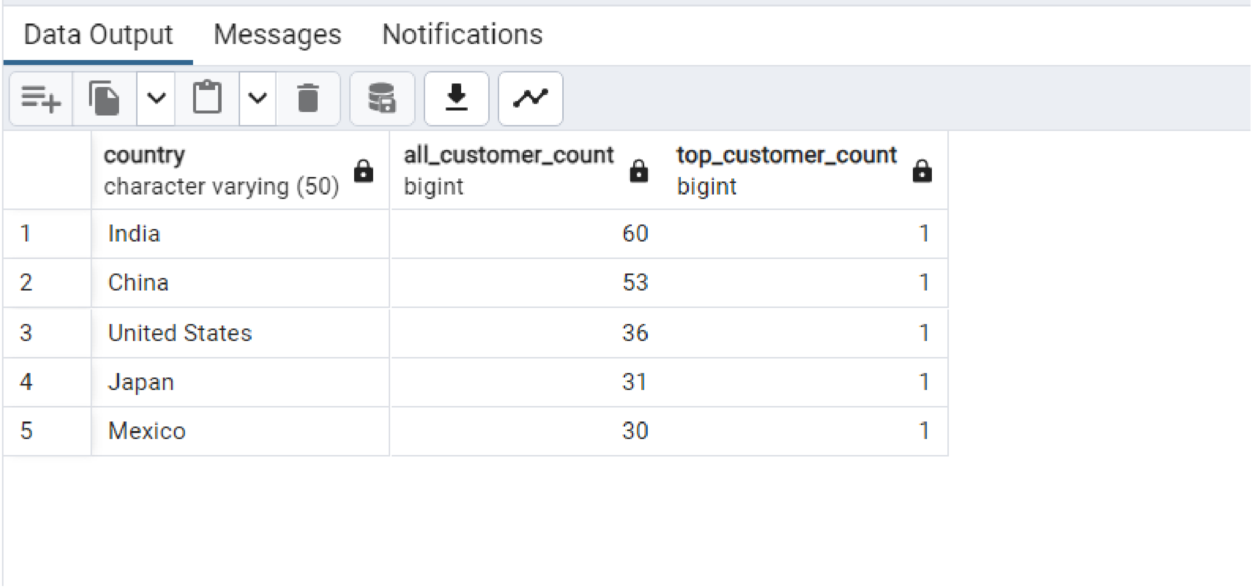
```sql
SELECT AVG (total_paid_amount)
FROM
(SELECT A.customer_id,
 B.first_name,
 B.last_name,
 D. city,
 E. country,
 SUM (amount) AS total_paid_amount
FROM payment A
INNER JOIN customer B on A.customer_id = B.customer_id
INNER JOIN address C on B.address_id = C.address_id
INNER JOIN city D on C.city_id = D.city_id
INNER JOIN country E on D.country_id = E.country_id
GROUP BY A.customer_id, B.first_name, B.last_name, D.city , E.country
HAVING city in
('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary', 'Tianjin', 'Sianjur')
ORDER BY SUM(amount) DESC
LIMIT 5) AS average;
```

```
  avg
  numeric
1 105.5540000000000000
```

2.a



```sql
SELECT D.country,
COUNT(A.customer_id) AS all_customer_count, COUNT (top_5_customers) AS top_customer_count
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
LEFT JOIN (SELECT A.customer_id, B.first_name, B.last_name, D. city, E. country,
SUM (amount) AS total_paid_amount
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
GROUP BY A.customer_id, B.first_name, B.last_name, D.city , E.country
HAVING city in
('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary', 'Tianjin', 'Sianjur')
ORDER BY SUM(amount) 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	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1

3.a I think this query could have been done without subqueries because subqueries slow the process down and we could have used a JOIN. If you are deciding between using or not using a subquery one of the top things you can ask yourself is the data changing frequently. If yes then use subqueries to bring in the latest data. If needing to consider this, then running an “EXPLAIN” could help to determine if using a subquery is cost effective.