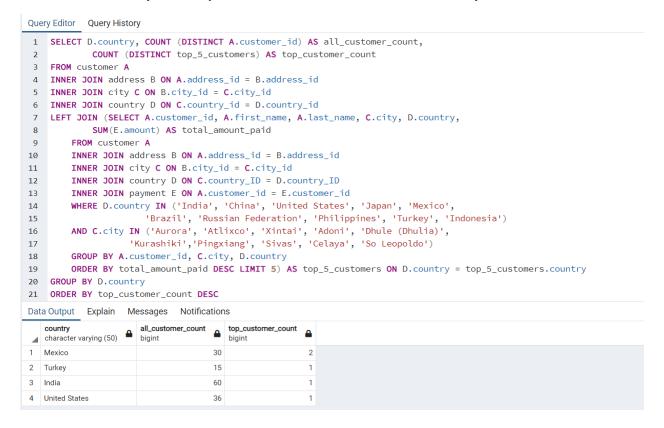
3.8: Performing Subqueries

1: Find the average amount paid by the top 5 customers.

```
Query Editor Query History
1 SELECT AVG(total_amount_paid)
2 FROM
3 (SELECT A.customer_id, A.first_name, A.last_name, C.city, D.country,
           SUM(E.amount) AS total_amount_paid
4
5 FROM customer A
6 INNER JOIN address B ON A.address_id = B.address_id
7 INNER JOIN city C ON B.city_id = C.city_id
8 INNER JOIN country D ON C.country_ID = D.country_ID
9 INNER JOIN payment E ON A.customer_id = E.customer_id
10 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
                   'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
12 GROUP BY A.customer_id, C.city, D.country
13 ORDER BY total_amount_paid DESC
14 LIMIT 5) AS average
Data Output Explain Messages Notifications
  avg
numeric A
1 107.354
```

2: Find out how many of the top 5 customers are based within each country.



3. Step 1 can be completed without a subquery by using an aggregate function. You need a subquery for step two because there are multiple tables you need to look at and the subquery allows you to do that without creating an additional table. Subqueries are useful when working with data that is continuously updated or tables that are reliant on each other with one of them, the inner, continuously being updated. You save time by having the subquery instead of having to run two separate queries.