Step 1:

```
SELECT AVG(Total_amount_paid) AS average_amount_paid_by_the_top_5_customers

FROM (SELECT B.customer_id, B.first_name, B.last_name, E.country, D.city, SUM(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

   GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city

   HAVING D.city IN('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',

'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')

   ORDER BY SUM(amount) DESC

   LIMIT 5) AS Total_Amount_Paid
```

```
107.3540000000000000
```

Step 2:

```
SELECT B.country, B.All_customer_count, COUNT(A.country) AS Top_customer_count

FROM (SELECT B.customer_id, B.first_name, B.last_name, E.country, D.city, SUM(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

   GROUP BY B.customer_id, B.first_name, B.last_name, E.country, D.city

   HAVING D.city IN('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
   ORDER BY SUM(amount) DESC
   LIMIT 5) A
LEFT JOIN (SELECT D.country, COUNT(customer_id) AS All_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
   GROUP BY country
   ORDER BY COUNT(customer_id) DESC) B
         ON A.country = B.country
         GROUP BY B.country, B.All_customer_count
         ORDER BY COUNT(A.country) DESC
```

Country	All_customer_count	Top_customer_count
Mexico	30	2
India	60	1
Turkey	15	1
U.S.	36	1

Step: 3

From the limited knowledge I have about this, I think that both queries did need the use of a subquery to be used effectively. The first query was easy to understand on how to utilize the subquery, yet the same can’t be said for the second query. That query was very complex, and it took quite a while for me to wrap my head around what I was supposed to do.

It’s hard to say when a subquery is needed in SQL. My best answer for this is that whenever an analysis is stumped on how to get the info they want from SQL, a subquery might be what they need. But this does depend on the specific situation the analysis is in.