3.8-

1.

```
SELECT Round(AVG(total_paid),2) AS average
FROM(SELECT A.customer_id, A.first_name, A.last_name, E.country, B.city, SUM(C.amount) AS total_
    From customer A
Inner Join payment C ON A.customer_id = C.customer_id
Inner Join address D on A.address_id = D.address_id
Inner Join city B on D.city_id = B.city_id
Inner JOIN country E on B.country_id = E.country_id
Where city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulla)', 'Kurashiki',
              'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, E.country, B.city
ORDER BY total_paid DESC
LIMIT 5) AS total_amount_paid;
```

Query Editor | Query History

Data Output | Explain | Messages | Notifications

| | average numeric |
|---|---|
| 1 | 107.35 |

2.

```sql
SELECT DISTINCT(A.country),
    COUNT(DISTINCT D.customer_id) AS all_customer_count,
    COUNT(DISTINCT A.country) AS top_customer_count
FROM country A
Inner JOIN city B
    ON A.country_id = B.country_id
INNER JOIN address c
    ON b.city_id = C.city_id
INNER JOIN customer D
    ON C.address_id = D.address_id
LEFT JOIN (SELECT A.customer_id, A.first_name, A.last_name, E.country, B.city, SUM(C.amount) AS total_paid
    From customer A
Inner Join payment C ON A.customer_id = C.customer_id
Inner Join address D on A.address_id = D.address_id
Inner Join city B on D.city_id = B.city_id
Inner JOIN country E on B.country_id = E.country_id
Where E.country IN ('India', 'China', 'United States', 'Japan', 'Mexica', 'Brazil',
             'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
AND B.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulla)', 'Kurashiki',
            'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, E.country, B.city
ORDER BY total_paid DESC
LIMIT 5) AS top_5_customers
ON A.country = top_5_customers.COUNTRY
GROUP BY A.country, top_5_customers
ORDER BY all_customer_count DESC
LIMIT 5;
```

Data Output | Explain | Messages | Notifications

| | country<br>character varying (50) | all_customer_count<br>bigint | top_customer_count<br>bigint |
|---|---|---|---|
| 1 | India | 60 | 1 |
| 2 | China | 53 | 1 |
| 3 | United States | 36 | 1 |
| 4 | Japan | 31 | 1 |
| 5 | Mexico | 30 | 1 |

Step one could be used without the subquery using the aggregate function but step two would need results from another table so the subquery is required.

Subqueries are useful when needing to aggregate data from two different tables without having to create a new table. It also helps for when data is constantly updating.