Answers 3.8

Step 1

Query | Query History

```
1  SELECT AVG(total_payment) AS average
2  FROM (SELECT A.customer_id, A.first_name, A.last_name, B.address,
3        C.city, D.country, SUM(E.amount) AS total_payment
4  FROM customer A
5  INNER JOIN address B on A.address_id = B.address_id
6  INNER JOIN city C on B.city_id = C.city_id
7  INNER JOIN country D on C.country_id = D.country_id
8  INNER JOIN payment E on A.customer_id = E.customer_id
9  WHERE C.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulia)'
10                 ,'Kurashiki','Pingxang','Sivas','Celaya','So Leopoldo')
11 GROUP BY A.customer_id, A.first_name, A.last_name, B.address, C.city, D.country
12 ORDER BY total_payment DESC
13 LIMIT 5) AS total_amount_paid
```

Data output | Messages | Notifications

	average numeric
1	107.354

Step 2

Query | Query History

```
1  SELECT DISTINCT (A.country),
2  COUNT(DISTINCT D.customer_id) AS all_customer_count,
3  COUNT(distinct top_5_customers.customer_id) AS Top_customer_count
4  FROM country A
5  INNER JOIN city B on A.country_id = B.country_id
6  INNER JOIN address C on B.city_id = C.city_id
7  INNER JOIN customer D on C.address_id = D.address_id
8  LEFT JOIN
9  (SELECT A.customer_id, A.first_name, A.last_name, B.address,
10       C.city, D.country, SUM(E.amount) AS total_payment
11 FROM customer A
12 INNER JOIN address B on A.address_id = B.address_id
13 INNER JOIN city C on B.city_id = C.city_id
14 INNER JOIN country D on C.country_id = D.country_id
15 INNER JOIN payment E on A.customer_id = E.customer_id
16 WHERE C.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule (Dhulia)'
17                 ,'Kurashiki','Pingxang','Sivas','Celaya','So Leopoldo')
18 GROUP BY A.customer_id, A.first_name, A.last_name, B.address, C.city, D.country
19 ORDER BY total_payment DESC
20 LIMIT 5)
21 AS top_5_customers on A.country = top_5_customers.country
22 GROUP BY A.country, top_5_customers.country
23 ORDER BY all_customer_count DESC
```

Data output | Messages | Notifications

	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
6	Brazil	28	0
7	Russian Federation	28	0
8	Philippines	20	0
9	Turkey	15	1

Step 3

Do you think steps 1 and 2 could be done without using subqueries?

I believe step 1 could be done without a subquery as the use of an aggregate function could calculate this. Step 2 would likely not be able to be completed without a subquery as it required the creation of a whole new table for multiple steps along the way.

When do you think subqueries are useful?

When there are many steps required beyond the initial creation of a new table that would be too complex to complete without creating an entirely new table in the database, which can use unnecessary storage.