Answers 3.8

Query | Query History

```
SELECT AVG(total_amount_paid) AS average
FROM
(SELECT SUM(A.amount) AS total_amount_paid,
       A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country
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
WHERE D.city IN('Aurora', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary', 'Tianjin', 'Cianjur')
GROUP BY A.amount, A.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	average numeric
1	41.3220000000000000

Query | Query History

```sql
SELECT E.country,
       COUNT(DISTINCT B.customer_id) AS all_customer_count,
       COUNT(DISTINCT top_5_customers) AS top_customer_count
FROM customer B
JOIN address C ON B.address_id=C.address_id
JOIN city D ON C.city_id=D.city_id
JOIN country E ON D.country_id=E.country_id
LEFT JOIN
(SELECT SUM(A.amount) AS total_amount_paid,
       B.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country
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
WHERE D.city IN(SELECT D.city
                FROM customer B
                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
               WHERE E.country IN (SELECT E.country
                                   FROM customer B
                                   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 E.country
                                  ORDER BY COUNT(B.customer_id) DESC
                                  LIMIT 10)
               GROUP BY E.country,
                        D.city
               ORDER BY COUNT(B.customer_id) DESC
               LIMIT 10)
GROUP BY B.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY SUM(A.amount) DESC
LIMIT 5) AS top_5_customers
ON B.customer_id=top_5_customers.customer_id
GROUP BY E.country
ORDER BY all_customer_count DESC
LIMIT 10
```

Data Output | Messages | Notifications

	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
6	Brazil	28	0
7	Russian Federation	28	0
8	Philippines	20	0
9	Turkey	15	0
10	Indonesia	14	0

Step 3:

My intuition tells me that there are many ways to perform the same function in SQL, and even though my experience is limited, I would say yes, the queries could have been performed without using subqueries. SQL is a great language in that it's set up to have redundancies, ie many paths to reach the same goal. However, I am not aware of all the different ways this function could be performed. Although I have yet to learn about CTEs, after doing some reading it seems they can provide another route to getting the same results.

Subqueries appear to have multiple uses; one is filtering results when the conditions set are complex and cannot be simply expressed. Another use case is for aggregations: when calculating aggregations for a subset of data, you can use subqueries. Also, subqueries can be helpful for the exact case above- when dealing with complex join conditions you can use subqueries to try to simplify things. I'm still learning, and I'm sure there are other use cases, but I think these are some of the main ways to use subqueries.