STEP 1

```sql
WITH average_amount_paid_cte (customer_id,
    first_name,
        last_name,
        city,
        total_amount_paid)
AS
(SELECT A.customer_id,
        A.first_name,
        A.last_name,
        C.city,
        SUM(E.amount) AS total_amount_paid
FROM payment E
INNER JOIN customer A ON E.customer_id=A.customer_id
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
WHERE C.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
GROUP BY A.customer_id, D.country, C.city
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT  AVG (total_amount_paid)
FROM average_amount_paid_cte
```

Data Output | Messages | Explain | Notifications

	avg numeric
1	107.3540000000000000

```sql
WITH top_customer_count_cte AS  (SELECT A.customer_id,
        A.first_name,
        A.last_name,
        C.city,
        D. country_id,
        SUM(E.amount) AS total_amount
FROM payment E
INNER JOIN customer A ON E.customer_id=A.customer_id
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
WHERE C.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY A.customer_id, D.country_id, C.city
ORDER BY total_amount DESC
LIMIT 5),
all_customer_count_cte AS(SELECT  D.country,
        COUNT (DISTINCT A.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 D.country)

SELECT  D.country,
        COUNT (DISTINCT A.customer_id) AS all_customer_count,
        COUNT (DISTINCT top_customer_count_cte.customer_id) AS top_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
LEFT JOIN top_customer_count_cte ON D.country_id = top_customer_count_cte.country_id
GROUP BY D.country
ORDER BY top_customer_count DESC
LIMIT 5;
```

I first approached this CTE using the WITH command and defined the necessary column and made a descriptive name for the CTE. You write the statements by rewriting queries from the previous task in parenthesis using the AS command. You then adjust the order and group accordingly.

STEP 2.
I don't think CTEs and subqueries will make much of a difference. I think they will perform the same.

3.8 part 1
Aggregate (cost=29.22..29.23 rows=1 width=32)
3.9 part 1
Aggregate (cost=64.45..64.46 rows=1 width=32)

3.8 part 2
Aggregate (cost=29.22..29.23 rows=1 width=32)
3.9 part 2
Limit (cost=166.84..166.86 rows=5 width=25)

The results did end up surprising me a bit. What surprised me the most is the cost of certain queries that you run whether it be a CTE or a subquery they can still be very costly. The speed does not differ that much among the queries run. This can also depend on the computer that somebody has.

For me personally writing subqueries was a lot easier compared to writing CTEs. For CTEs if you make one mistake you have to figure out where you placed the parentheses wrong and what name you have to correctly put for the CTE. The rearranging for the CTE was also a bit confusing for me because looking at all of that code written then having to rearrange it to me was super overwhelming and I did not want to mess anything up.