3.9: Common Table Expressions

1.

Query Editor | Query History

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

Data Output | Explain | Messages | Notifications

	avg numeric
1	107.354

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History

```
WITH top_5_customers_cte (customer_id, first_name, last_name, country, total_amount_paid)
AS (SELECT A.customer_id, A.first_name, A.last_name, C.city, D.country,
        SUM(E.amount) AS total_amount_paid
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
INNER JOIN payment E ON A.customer_id = E.customer_id
WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
    AND country IN ('India', 'China', 'United States', 'Japan', 'Mexico',
                'Brazil', 'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
GROUP BY A.customer_id, C.city, D.country
ORDER BY total_amount_paid DESC
LIMIT 5)
SELECT D.country, COUNT(DISTINCT A.customer_id) AS all_customer_count, COUNT(DISTINCT D.country) 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
INNER JOIN payment E ON A.customer_id = E.customer_id
LEFT JOIN top_5_customers_cte ON D.country = top_5_customers_cte.country
GROUP BY D.country
ORDER BY all_customer_count DESC
LIMIT 5
```

Data Output | Explain | 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

First, I defined the cte with the WITH clause and gave it a name, then using AS I added my query from previously. Then I had to write a new select statement to get the information needed.

2. I think the CTE and subquery will perform about the same when the query is small but the CTE will perform better for larger queries. The speed of the first query without the CTE is 33 msec while with the

CTE its 32 msec. The speed of the second query without the CTE is 64 msec while with the CTE its 43 msec. I was surprised by the time difference for the second query because of what it took to write the query with the CTE.

3. The challenges I faced were mainly with the second query. I had difficulty figuring out what columns I needed and writing the new SELECT statement to properly reference the temporary table.