

Step 1.

Question 1 from 3.8

-- CTE That Lists Total Payments from Top 5 Customers

WITH Top5_payments_cte (country, city, amount) AS

(SELECT A.customer_id,

d.country,

c.city,

SUM(amount) AS Top5_Payments

FROM customer A

INNER JOIN payment E ON A.customer_id = E.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(SELECT c.city

-- Finds Top 5 Cities in Top 5 Country

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

WHERE country IN (SELECT D.country

-- Finds Top 5 Country

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

ORDER BY COUNT(customer_id) DESC

LIMIT 10)

GROUP By d.country, c.city

ORDER BY COUNT(A.customer_id) DESC

LIMIT 10)

GROUP By A.customer_id, d.country, c.city

ORDER BY SUM(amount) DESC

LIMIT 5)

-- Finds Average of Total Payments from Top 5 Customers

SELECT AVG(Top5_Payments)

FROM Top5_payments_cte

;

```
630 -- CTE That Lists Total Payments from Top 5 Customers
631 WITH Top5_payments_cte ( country, city, amount) AS
632 (SELECT A.customer_id,
633         d.country,
634         c.city,
635         SUM(amount) AS Top5_Payments
636 FROM customer A
637 INNER JOIN payment E ON A.customer_id = E.customer_id
638 INNER JOIN address B ON A.address_id = B.address_id
639 INNER JOIN city C ON B.city_id = C.city_id
640 INNER JOIN country D ON C.country_id = D.country_id
641 WHERE c.city IN(SELECT c.city
642                 -- Finds Top 5 Cities in Top 5 Country
643                 FROM customer A
644                 INNER JOIN address B ON A.address_id = B.address_id
645                 INNER JOIN city C ON B.city_id = C.city_id
646                 INNER JOIN country D ON C.country_id = D.country_id
647                 WHERE country IN (SELECT D.country
648                                   -- Finds Top 5 Country
649                                   FROM customer A
650                                   INNER JOIN address B ON A.address_id = B.address_id
651                                   INNER JOIN city C ON B.city_id = C.city_id
652                                   INNER JOIN country D ON C.country_id = D.country_id
653                                   GROUP By d.country
654                                   ORDER BY COUNT(customer_id) DESC
655                                   LIMIT 10)
656                 GROUP By d.country, c.city
657                 ORDER BY COUNT(A.customer_id) DESC
658                 LIMIT 10)
659 GROUP By A.customer_id, d.country, c.city
660 ORDER BY SUM(amount) DESC
661 LIMIT 5)
662
663 -- Finds Average of Total Payments from Top 5 Customers
664 SELECT AVG(Top5_Payments)
665 FROM Top5_payments_cte
666 ;
667
668
```

Data Output Messages Notifications



	avg numeric	🔒
1	105.5540000000000000	

Question 2 From 3.8

```
-- CTE That Lists the Top 5 Payments of the Top 5 Customers
WITH Top5_CustomerPayments_CTE (customer_id, amount) AS(
    SELECT A.customer_id AS customer_id,
           SUM(amount) AS Top5_Payments
    FROM customer A
    INNER JOIN payment E ON A.customer_id = E.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(SELECT c.city
        -- Finds Top 5 Cities in Top 5 Country
        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
        WHERE country IN (SELECT D.country
            -- Finds Top 5 Country
            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
            ORDER BY COUNT(customer_id) DESC
            LIMIT 10)
        GROUP By d.country, c.city
        ORDER BY COUNT(A.customer_id) DESC
        LIMIT 10)
    GROUP By A.customer_id, A.first_name, A.last_name, d.country, c.city
    ORDER BY SUM(amount) DESC
    LIMIT 5)

SELECT d.country AS Country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count,
       COUNT(Top5_CustomerPayments_CTE.customer_id) AS top_5_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 Top5_CustomerPayments_CTE ON A.customer_id =
Top5_CustomerPayments_CTE.customer_id
GROUP BY d.country
ORDER BY top_5_customer_count DESC
```

```

629 -- CTE That Lists the Top 5 Payments of the Top 5 Customers
630 WITH Top5_CustomerPayments_CTE (customer_id, amount) AS(
631     SELECT A.customer_id AS customer_id,
632            SUM(amount) AS Top5_Payments
633     FROM customer A
634     INNER JOIN payment E ON A.customer_id = E.customer_id
635     INNER JOIN address B ON A.address_id = B.address_id
636     INNER JOIN city C ON B.city_id = C.city_id
637     INNER JOIN country D ON C.country_id = D.country_id
638     WHERE c.city IN(SELECT c.city
639                     -- Finds Top 5 Cities in Top 5 Country
640                     FROM customer A
641                     INNER JOIN address B ON A.address_id = B.address_id
642                     INNER JOIN city C ON B.city_id = C.city_id
643                     INNER JOIN country D ON C.country_id = D.country_id
644                     WHERE country IN (SELECT D.country
645                                     -- Finds Top 5 Country
646                                     FROM customer A
647                                     INNER JOIN address B ON A.address_id = B.address_id
648                                     INNER JOIN city C ON B.city_id = C.city_id
649                                     INNER JOIN country D ON C.country_id = D.country_id
650                                     GROUP By d.country
651                                     ORDER BY COUNT(customer_id) DESC
652                                     LIMIT 10)
653                     GROUP By d.country, c.city
654                     ORDER BY COUNT(A.customer_id) DESC
655                     LIMIT 10)
656     GROUP By A.customer_id, A.first_name, A.last_name, d.country, c.city
657     ORDER BY SUM(amount) DESC
658     LIMIT 5)
659
660 SELECT d.country AS Country,
661        COUNT(DISTINCT A.customer_id) AS all_customer_count,
662        COUNT(Top5_CustomerPayments_CTE.customer_id) AS top_5_customer_count
663 FROM customer A
664 INNER JOIN address B ON A.address_id = B.address_id
665 INNER JOIN city C ON B.city_id = C.city_id
666 INNER JOIN country D ON C.country_id = D.country_id
667 LEFT JOIN Top5_CustomerPayments_CTE ON A.customer_id = Top5_CustomerPayments_
668 GROUP BY d.country
669 ORDER BY top_5_customer_count DESC
670

```

Data Output Messages Notifications

	country character varying (50)	all_customer_count bigint	top_5_customer_count bigint
1	Mexico	30	1
2	India	60	1
3	China	53	1
4	United States	36	1
5	Japan	31	1
6	Argentina	13	0
7	Armenia	1	0
8	Austria	3	0
9	Azerbaijan	2	0

Step 1C.)

I first wrote the WITH statement, using the Select outputs from the subquery to fill the column names. I then took the main subquery which includes the inner subqueries as well, and placed them into the CTE statement. In the main Query, I replaced the subquery with the name of the CTE and ran the codes. For part 2 there was an issue with the syntax of the LEFT JOIN statement, which I figured required me to remove the parenthesis around the CTE name in the main QUERY. Overall, the process was simple. However, I think there are more I could do to make the entire code simpler and more readable such as removing repeating lines of code.

Step 2.)

Question 1. I think that the CTE approach will perform better than the multiple Subqueries. This is because the CTE approach creates a table that can be used in the main Query versus running a subquery in the main Query which can run multiple times.

Question 2.

Total Cost for Question 1:

- **Using Subquery**

“Aggregate (cost=166.06..166.07 rows=1 width=32)”

“Successfully run. Total query runtime: 105 msec. 1 rows affected.”

- **Using CTE**

“Aggregate (cost=166.06..166.07 rows=1 width=32)”

“Successfully run. Total query runtime: 87 msec. 1 rows affected.”

Total Cost for Question 2:

- **Using Subquery**

“Sort (cost=270.11..270.38 rows=109 width=25)”

“Successfully run. Total query runtime: 86 msec. 108 rows affected.”

- **Using CTE**

“Sort (cost=270.11..270.38 rows=109 width=25)”

“Successfully run. Total query runtime: 62 msec. 108 rows affected.”

Question 4. Based on the outputs from the Explain Function and the message popup detailing the runtime, using the total cost on the system is the same for both, but runtime is significantly lower when using CTE than with subqueries. I’ve also found that the runtime changes with each Code Execute. However, despite differing runtimes, the CTE for both queries performs faster than using subqueries. This could be because a CTE is run once and is held as a table in memory, Compared to a subquery which may need to run multiple times during a query.

Step 3

Converting the Subqueries into CTE was simple because I had previously labeled all subqueries with comments detailing their purpose and what they output. To create the CTE, I moved the lines of code from the original main query and moved them into the WHERE statement of the CTE. The subquery in the main Query was changed to the title of the CTE. The code ran without much syntax change.

Problems that I faced were mostly syntax issues when replacing the subquery with CTE statements. For example, the LEFT JOIN statement for Query 2 required some syntax change, the parentheses surrounding the subquery had to be removed when replacing with the CTE name. Another syntax issue was changing the table references for identifying the relationship between the customer Table and the CTE table (this line: "A.customer_id = Top5_CustomerPayments_CTE.customer_id"). Overall, getting the code to run the correct output was doable.