

ANSWERS 3.9

1. The average amount paid by the top 5 customers.

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the Rockbuster/postgres@PostgreSQL 15 database. The query calculates the average amount paid by the top 5 customers based on the total amount paid.

```

1 WITH Average_amount_paid_cte(customer_id, first_name, last_name, city, country, total_amount_paid) AS
2 (SELECT B.customer_id,
3    B.first_name,
4    B.last_name,
5    D.city,
6    E.country,
7    SUM(A.amount) as Total_Amount_Paid
8 FROM payment A
9 INNER JOIN customer B ON A.customer_id = B.customer_id
10 INNER JOIN address C ON B.address_id = C.address_id
11 INNER JOIN city D ON C.city_id = D.city_id
12 INNER JOIN country E ON D.country_id = E.country_id
13 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhulia)', 'Kurashiki', 'Pingxiang', 'Sivas')
14 GROUP BY B.customer_id,
15    B.first_name,
16    B.last_name,
17    D.city,
18    E.country
19 ORDER BY Total_Amount_Paid DESC
20 LIMIT 5)
21 SELECT AVG(total_amount_paid) AS Average_amount_paid
22 FROM Average_amount_paid_cte

```

The Data Output shows the result of the query:

average_amount_paid
107.3540000000000000

Successfully run. Total query runtime: 93 msec. 1 rows affected.

2. The top 5 customers are based within each country.

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the Rockbuster/postgres@PostgreSQL 15 database. The query finds the top 5 customers within each country based on the total amount paid.

```

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

```

The Data Output shows the result of the query:

country	all_customer_count	top_customer_count
Mexico	30	2
United States	36	1
India	60	1

Successfully run. Total query runtime: 133 msec. 5 rows affected.

ANSWERS 3.9

WITH top_customer_count_cte(customer_id, first_name, last_name, city, country, total_amount_paid) AS

(SELECT B.customer_id, B.first_name, B.last_name, D.city, E.country,

SUM(A.amount) AS total_amount_paid

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 city IN('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Duhlia)',

'Kursahiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')

GROUP BY B.customer_id, B.first_name, B.last_name, D.City, E.country

ORDER BY total_amount_paid DESC

LIMIT 5),

all_customer_count_cte AS

(SELECT E.country,

COUNT (DISTINCT B.customer_id) AS all_customer_count,

COUNT (DISTINCT E.country) AS top_customer_count

FROM country E

INNER JOIN city D ON E.country_id = D.country_id

INNER JOIN address C ON D.city_id = C.city_id

INNER JOIN customer B ON C.address_id = B.address_id

GROUP BY E.country)

SELECT E.country,

COUNT (DISTINCT B.customer_id) AS all_customer_count,

COUNT (DISTINCT top_customer_count_cte.customer_id) AS top_customer_count

FROM country E

INNER JOIN city D ON E.country_id = D.country_id

INNER JOIN address C ON D.city_id = C.city_id

INNER JOIN customer B ON C.address_id = B.address_id

ANSWERS 3.9

LEFT JOIN

top_customer_count_cte ON E.country = top_customer_count_cte.country

GROUP BY E.country

ORDER BY top_customer_count DESC

LIMIT 5

3. Write 2 to 3 sentences explaining how you approached this step, for example, what you did first, second, and so on.

First step: was to identify the relevant tables from our ERD so that we could work out the pattern we want to connect.

Second Step: copying the subqueries from previous task

Third Step: replacing the subqueries with CTE's using WITH at the beginning of the query and providing AS keyword.

Fourth Step: This was an additional step for query 2 where we wrote two CTE's one for top customer count and the other for all customer count.

Fifth Step: Finally after the CTE came the main statement. (as divided by the color scheme above)

Step 2: Compare the performance of your CTEs and subqueries.

1. Which approach do you think will perform better and why?
 - a. I feel using CTE is much better as these are used right at the beginning of the query and improves readability.
2. The EXPLAIN command gives you an estimated cost. To find out the actual speed of your queries.
 - a. CTE 1: 299 msec
 - b. Sub query 1: 176 msec
 - c. CTE 2: 148 msec
 - d. Sub query 2: 475 msec

ANSWERS 3.9

3. Compare the costs of all the queries by creating query plans for each one.

SUB QUERY 1

Query	Query History
1	EXPLAIN
2	SELECT AVG(total_amount_paid) AS average
3	FROM
4	(SELECT B.customer_id,
5	B.first_name,
6	B.last_name,
7	D.city,
8	E.country,
9	SUM(A.amount) as Total_Amount_Paid
10	FROM payment A
11	INNER JOIN customer B ON A.customer_id = B.c
12	INNER JOIN address C ON B.address_id = C.add
13	INNER JOIN city D ON C.city_id = D.city_id
14	INNER JOIN country E ON D.country_id = E.cou
15	WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'A
16	GROUP BY B.customer_id,
17	B.first_name
Data Output	Messages Explain × Notifications
+	QUERY PLAN
1	Aggregate (cost=64.45..64.46 rows=1 width=32)
2	-> Limit (cost=64.37..64.39 rows=5 width=67)
3	-> Sort (cost=64.37..64.98 rows=243 width=67)
4	Sort Key: (sum(a.amount)) DESC
5	-> HashAggregate (cost=57.30..60.34 rows=243 width=67)
6	Group Key: b.customer_id, d.city, e.country
7	-> Nested Loop (cost=18.16..54.87 rows=243 width=41)
8	-> Hash Join (cost=17.88..37.14 rows=10 width=35)
Total rows: 22 of 22	Query complete 00:00:00.176

SUB

QUERY 2

Query	Query History
1	EXPLAIN
2	SELECT E.country,
3	COUNT (DISTINCT B.customer_id) AS all_custome
4	COUNT (DISTINCT top_5_customers.customer_id)
5	FROM country E
6	INNER JOIN city D ON E.country_id = D.country
7	INNER JOIN address C ON D.city_id = C.city_id
8	INNER JOIN customer B ON C.address_id = B.adc
9	LEFT JOIN
10	(SELECT B.customer_id, B.first_name, B.last_r
11	SUM(A.amount) AS total_amount_paid
12	FROM payment A
13	INNER JOIN customer B ON A.customer_id = B. c
14	INNER JOIN address C ON B.address_id = C.addr
15	INNER JOIN city D ON C.city_id = D.city_id
16	INNER JOIN country E ON D.country_ID = E.cour
17	WHERE city IN ('Aurora', 'Atlixco', 'Xintai')
Data Output	Messages Explain × Notifications
+	QUERY PLAN
1	Limit (cost=166.23..166.24 rows=5 width=25)
2	-> Sort (cost=166.23..166.50 rows=109 width=25)
3	Sort Key: (count(DISTINCT top_5_customers.customer_id)) DESC
4	-> GroupAggregate (cost=155.43..164.42 rows=109 width=25)
5	Group Key: e.country
6	-> Merge Left Join (cost=155.43..158.83 rows=599 width=17)
7	Merge Cond: ((e.country)::text = (top_5_customers.country)::text)
8	-> Sort (cost=90.94..92.44 rows=599 width=13)
Total rows: 46 of 46	Query complete 00:00:00.475

CTE 1

Query	Query History
1	EXPLAIN
2	WITH Average_amount_paid_cte(customer_id, first_name, last_name
3	(SELECT B.customer_id,
4	B.first_name,
5	B.last_name,
6	D.city,
7	E.country,
8	SUM(A.amount) as Total_Amount_Paid
9	FROM payment A
10	INNER JOIN customer B ON A.customer_id = B.customer_id
11	INNER JOIN address C ON B.address_id = C.address_id
12	INNER JOIN city D ON C.city_id = D.city_id
13	INNER JOIN country E ON D.country_id = E.country_id
14	WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule(Dhu
15	GROUP BY B.customer_id,
16	B.first_name,
17	B.last_name
Data Output	Messages Explain × Notifications
+	QUERY PLAN
1	Aggregate (cost=64.45..64.46 rows=1 width=32)
2	-> Limit (cost=64.37..64.39 rows=5 width=67)
3	-> Sort (cost=64.37..64.98 rows=243 width=67)
4	Sort Key: (sum(a.amount)) DESC
5	-> HashAggregate (cost=57.30..60.34 rows=243 width=67)
6	Group Key: b.customer_id, d.city, e.country
7	-> Nested Loop (cost=18.16..54.87 rows=243 width=41)
8	-> Hash Join (cost=17.88..37.14 rows=10 width=35)
Total rows: 22 of 22	Query complete 00:00:00.299

CTE 2

Query	Query History
1	EXPLAIN
2	WITH top_customer_count_cte(customer_id, first_
3	country, total_amount_paid) AS
4	(SELECT B.customer_id, B.first_name, B.last_nam
5	SUM(A.amount) AS total_amount_paid
6	FROM payment A
7	INNER JOIN customer B ON A.customer_id = B. cus
8	INNER JOIN address C ON B.address_id = C.address
9	INNER JOIN city D ON C.city_id = D.city_id
10	INNER JOIN country E ON D.country_ID = E.countr
11	WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'A
12	'Kursahiki', 'Pingxiang', 'Sivas', 'Celaya', 'S
13	GROUP BY B.customer_id, B.first_name, B.last_na
14	ORDER BY total_amount_paid DESC
15	LIMIT 5),
16	all_customer_count_cte AS
17	(SELECT E.country,
Data Output	Messages Explain × Notifications
+	QUERY PLAN
1	Limit (cost=166.23..166.24 rows=5 width=25)
2	-> Sort (cost=166.23..166.50 rows=109 width=25)
3	Sort Key: (count(DISTINCT top_customer_count_cte.customer_id)) DESC
4	-> GroupAggregate (cost=155.43..164.42 rows=109 width=25)
5	Group Key: e.country
6	-> Merge Left Join (cost=155.43..158.83 rows=599 width=17)
7	Merge Cond: ((e.country)::text = (top_customer_count_cte.country)::text)
8	-> Sort (cost=90.94..92.44 rows=599 width=13)
Total rows: 46 of 46	Query complete 00:00:00.148

ANSWERS 3.9

4. Did the results surprise you? Write a few sentences to explain your answer.

Yes indeed, I am. The cost from the CTEs are substantially different as well for the subqueries. Subquery 2 costed thrice as much of the CTE which clearly shows how much better it is to use CTEs.

Step 3: Write 1 to 2 paragraphs on the challenges you faced when replacing your subqueries with CTEs.

The replacing of subquery 1 with CTE was comparatively much simple and understandable. But it was a big challenge understanding the Subquery 2 and as a result the CTE 2.