**Question 3**

Query 1:

with

consecutive\_days AS (

SELECT

Cust\_Id,

Calendar\_Dt,

Amt\_LE,

(ROW\_NUMBER() OVER (PARTITION BY Cust\_Id ORDER BY Calendar\_Dt)) as seq,

DATE\_ADD(Calendar\_Dt, INTERVAL - (ROW\_NUMBER() OVER (PARTITION BY Cust\_Id ORDER BY Calendar\_Dt)) DAY) AS grp

FROM transactions

),

max\_consecutive\_days AS (

SELECT

Cust\_Id,

Calendar\_Dt,

Amt\_LE,

seq,

grp,

COUNT(grp) over(partition by Cust\_Id,grp) AS consecutive\_days\_count

FROM consecutive\_days

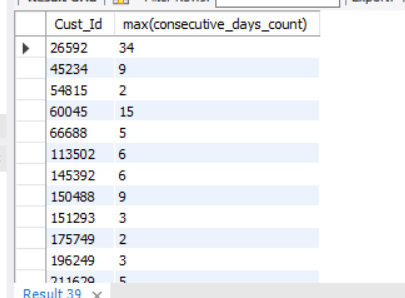
WHERE Amt\_LE > 0

)

SELECT Cust\_Id, max(consecutive\_days\_count)

FROM max\_consecutive\_days

group by Cust\_Id;

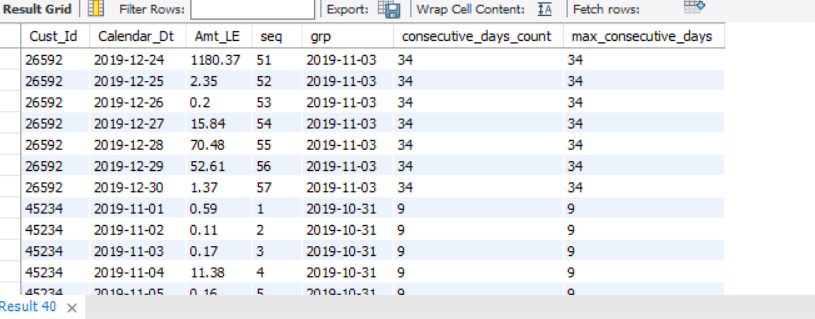


The CTE, **consecutive\_days**, groups the purchases by customer and creates a new grouping column **grp** based on the difference between the row number and the calendar date. This grouping column is used to calculate the number of consecutive days each customer made purchases.

the **max\_consecutive\_days** CTE counts the number of distinct groups of consecutive days each customer made purchases, filters out zero purchase amounts, and groups by customer.

The outer query selects the maximum count of consecutive days made by each customer.

This is an output for all columns which I used to calculate this final result:



Query 2:

with cte as(

select cust\_id,calendar\_dt,amt\_le,sum(amt\_le) over(partition by Cust\_Id order by Calendar\_Dt) as total\_amount

from transactions)

, cte2 as(

select \*,case when total\_amount >= 250 then 1 else 0 end as reached\_threshold

from cte

where total\_amount >= 250

),

cte22 as (SELECT cust\_id, min(cte2.calendar\_dt) final\_date

FROM cte2

group by cust\_id),

cte1 as(

select \*,case when total\_amount >= 250 then 1 else 0 end as reached\_threshold

from cte

where total\_amount <250

),cte11 as(SELECT cust\_id, min(cte1.calendar\_dt) first\_date

FROM cte1

group by cust\_id

),

final as (select c1.cust\_id, datediff(final\_date , first\_date) as days\_before\_reaching\_threshold

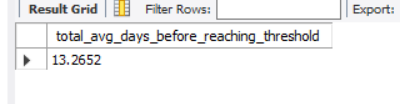
from cte11 c1 ,cte22 c2

where c1.cust\_id = c2.cust\_id)

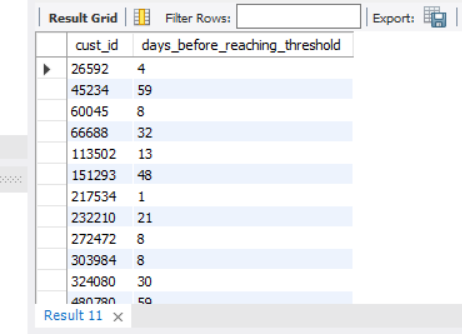
select avg(days\_before\_reaching\_threshold) as total\_avg\_days\_before\_reaching\_threshold

from final

;



Before taking the average …here is every count of days each customer takes to reach the threshold



I decided to count the days not the count of transactions as customers may make a few transaction in a long time period

But if u want count of transactions of each customer not to count the days

I made this query here is every count of transactions each customer takes to reach the threshold

Another answer:

with cte as(

select cust\_id,calendar\_dt,amt\_le,sum(amt\_le) over(partition by Cust\_Id order by Calendar\_Dt) as total\_amount

from transactions)

, cte2 as(

select \*,case when total\_amount >= 250 then 1 else 0 end as reached\_threshold

from cte

where total\_amount >= 250

),

cte1 as(

select \*,case when total\_amount >= 250 then 1 else 0 end as before\_reached\_threshold

from cte

where total\_amount <250

),

final as (select c1.cust\_id, count(before\_reached\_threshold) as Transaction\_before\_reaching\_threshold

from cte1 c1

where c1.cust\_id in (select cust\_id from cte2)

group by c1.cust\_id)

select \*

from final

;

Here is the average:

