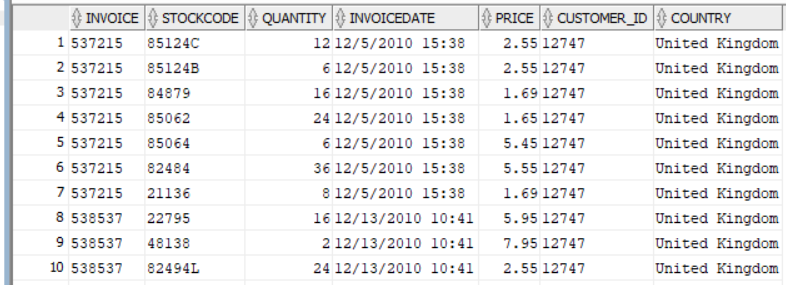
# **Analytical SQL project**

# **- First Question :**

Our data here is a retail transaction records which contains:

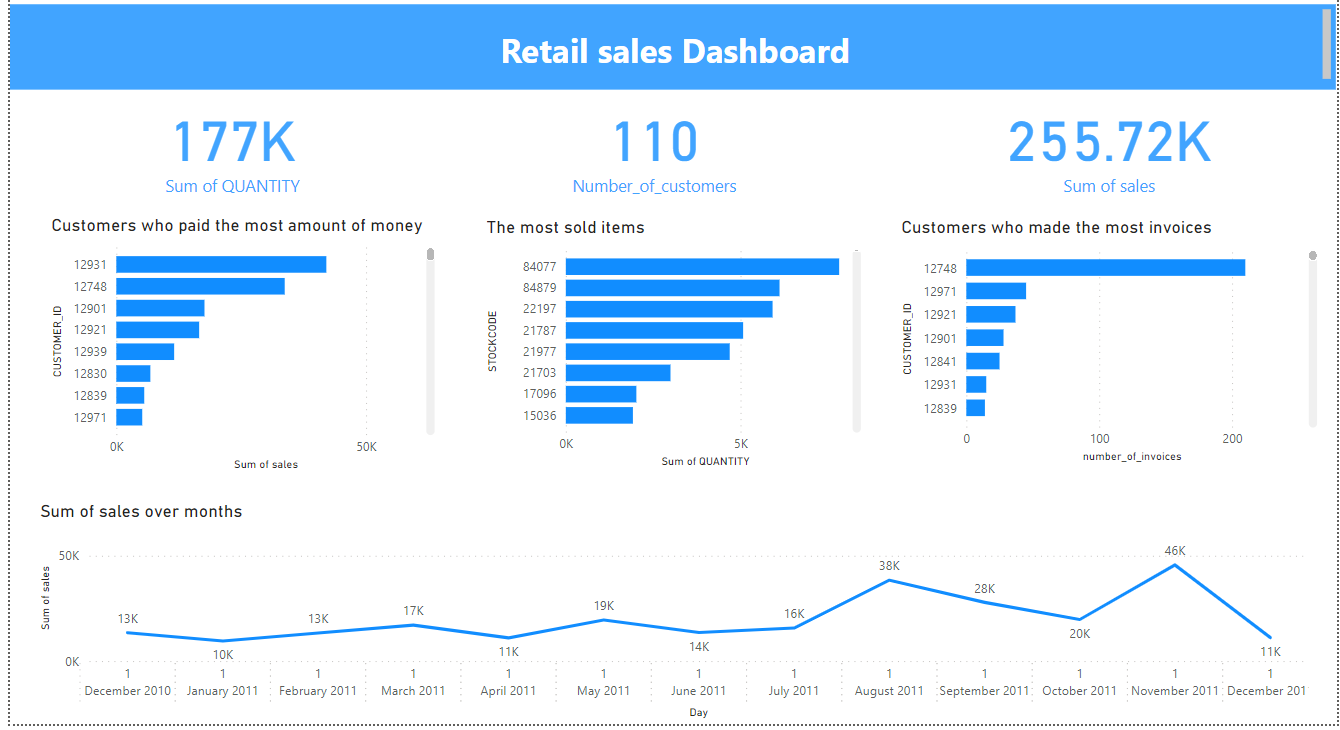
invoice\_id , stock\_code , quantity , Invoice\_date , price , customer\_id , country

It has 12858 row .. this is a sample of the data :



The purpose of the question was applying 5 analytical sql queries to extract useful business insights

Firstly I created a simple dashboard using power bi to show some insights about the data before analyzing it :



- Secondly I applied 2 sets of sql queries which contain 12 query :

the first set has **3** basic queries to explore the data and the second set has **9** analytical queries to extract business insights from the data

**- the first set :**

1 – displaying every customer\_id with his total number of invoices

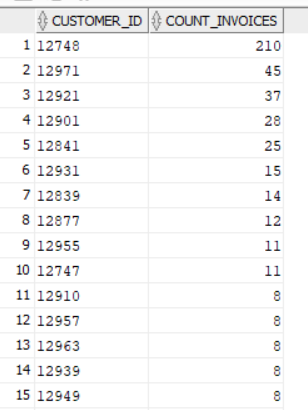
To know how many invoices that each customer made :

select customer\_id , count(distinct invoice) as count\_invoices

from tableRetail

group by customer\_id

order by count\_invoices desc ;



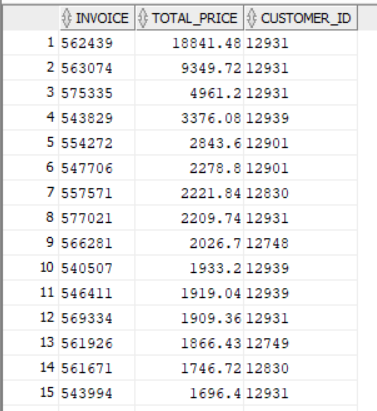
2- displaying each invoice id with its amount of money and the customer who made it to know who made the highest invoices :

select distinct invoice , sum(price \* quantity) as total\_price , customer\_id

from tableRetail

group by invoice , customer\_id

order by total\_price desc ;



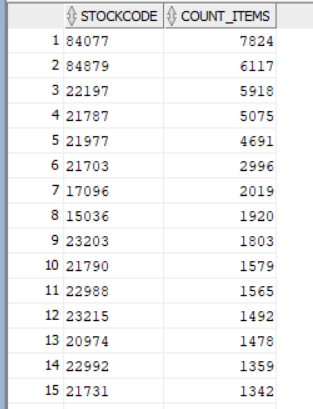
3- displaying each stock\_code with its sold quantity to know which stock items with has the most quantity sold :

select distinct stockcode , sum(quantity)as count\_items

from tableRetail

group by stockcode

order by count\_items desc ;



**- The second set :**

1 – displaying the top 10 customers ranked by every customer’s amount of money which has been spent in invoices

This will help us to know our loyal customers :

WITH RankedCustomers AS (

SELECT customer\_id, SUM(quantity \* price) AS total\_sales,

RANK() OVER (ORDER BY SUM(quantity \* price) DESC) AS customer\_rank

FROM tableRetail

GROUP BY customer\_id

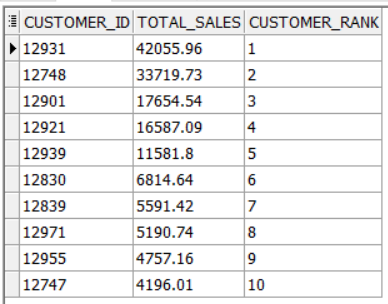
)

SELECT customer\_id, total\_sales, customer\_rank

FROM RankedCustomers

WHERE customer\_rank <= 10

ORDER BY customer\_rank;



2 – displaying the monthly spending amount for each customer and its accumulating sum to track the purchasing behavior of each customer over the months and notice if there any pattern in this behavior :

select customer\_id , invoiceday , customer\_sales , sum(customer\_sales) over (partition by customer\_id order by invoiceday) as total\_customer\_sales

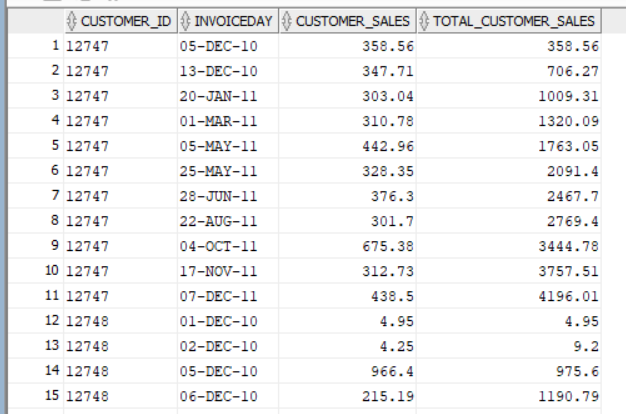
from

(select customer\_id , TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) as invoiceday , sum(quantity \* price) as customer\_sales

from tableRetail

group by customer\_id , TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') )

order by customer\_id , invoiceday )



3 – tracking count of items sold of each stock item with an accumulative sum of it over months to notice if there any pattern in purchasing some stock items :

select stockcode , day , count\_items\_sold ,

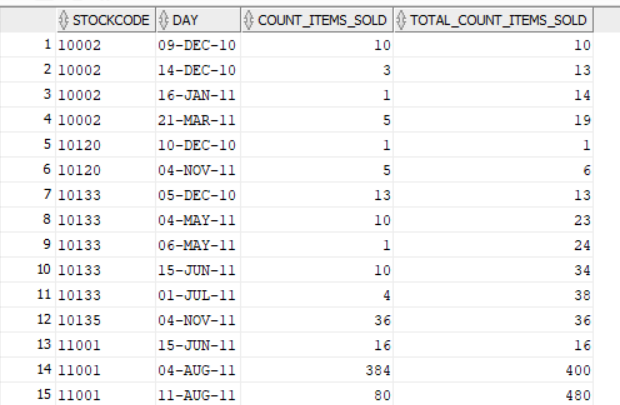
sum(count\_items\_sold) over (partition by stockcode order by day asc )as total\_count\_items\_sold

from

(select stockcode , TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') )as day , sum (quantity) as count\_items\_sold

from tableRetail

group by stockcode , TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') ) ) ;



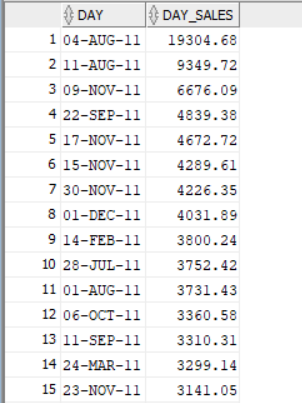
4 – displaying each day with its amount of sales to know the days with the highest amount of sales to detect if there is any pattern in them :

select distinct TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') ) as day ,

sum(quantity \* price) over (partition by TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') ) ) as day\_sales

from tableRetail

order by day\_sales desc ;



5 – displaying the count of customers who made transactions in each month and comparing it with the number of customers who made transactions in the previous month .. this help us to track the change in customers count over months and know if thsis change is positive or negative :

with months\_sales AS (select count (distinct customer\_id) as count\_customers , TO\_CHAR(TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')), 'MM/YYYY') as month

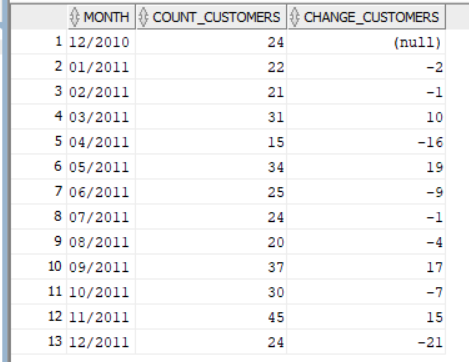
from tableRetail

group by TO\_CHAR(TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')), 'MM/YYYY'))

select month , count\_customers , (count\_customers -lag(count\_customers) over (order by to\_date(month , 'MM/YYYY'))) as change\_customers

from months\_sales

order by to\_date(month , 'MM/YYYY')asc ;



6 – displaying each month with its number of invoices and amount of sales and comparing it with the running maximum sales and the running minimum sales over months .. it will help us to assess the amount of achieved sales in each month :

with months\_sales AS (select sum(quantity \* price) as sales , TO\_CHAR(TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')), 'MM/YYYY') as month ,

count(distinct invoice) as count\_invoices

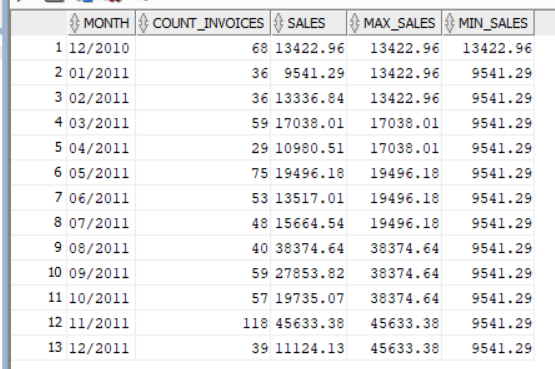
from tableRetail

group by TO\_CHAR(TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')), 'MM/YYYY'))

select month ,count\_invoices , sales , Max(sales) over(order by to\_date(month , 'MM/YYYY')asc ) as max\_sales , Min(sales) over(order by to\_date(month , 'MM/YYYY')asc ) as min\_sales

from months\_sales

order by to\_date(month , 'MM/YYYY')asc ;



7 – I noticed change in the price of some items so I wrote this query to track the change in prices .. if the item has only one price then it will be displayed .. if the item has multiple prices then each one of them will be displayed with the last date of applying each price before changing it :

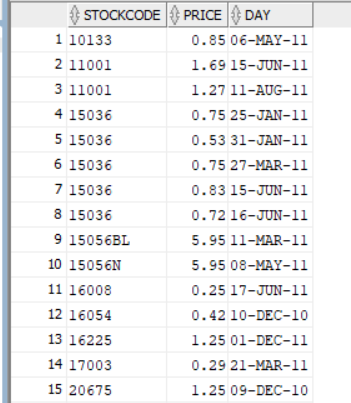
select stockcode , price , day from (

select distinct stockcode , price , lead (price) over(partition by stockcode order by TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) ) as coming\_price ,TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) as day

from tableRetail

order by stockcode , day asc )

where price <> coming\_price ;



8 – I noticed there are gaps among some dates so it means there are some without any transaction.. so I tried to get every gap among dates and ordered them to know the longest gaps to analyze why our sales stopped in these intervals :

select day , next\_day , (next\_day - day) as difference

from (

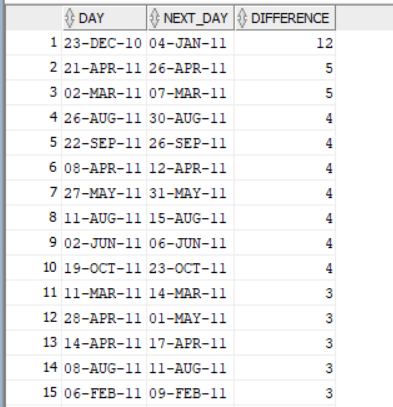
select distinct TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') ) as day ,

lead ( TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') ) ) over (order by TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI') ) asc ) as next\_day

from tableRetail )

where day <> next\_day

order by difference desc ;



9 – I wanted to classify the customers based on their max count of days without making any transactions .. I calculated average of gaps between purchasing days for all customers and I got 33

Then I compared it with the maximum of gap days for each customer and based on the result of this comparison I gave a value (consistent) or (inconsistent) to represent the status of each customer’s purchasing behavior :

WITH sequential\_days AS (

SELECT customer\_id, TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) AS day,

LEAD(TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI'))) OVER (PARTITION BY customer\_id ORDER BY TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI'))) AS next\_day

FROM tableRetail ),

difference\_between\_days AS (

SELECT customer\_id, day, next\_day,

MAX(next\_day - day) over (partition by customer\_id ) AS max\_gap,

round(AVG(next\_day - day) OVER () )AS avg\_gap

FROM sequential\_days

WHERE day <> next\_day )

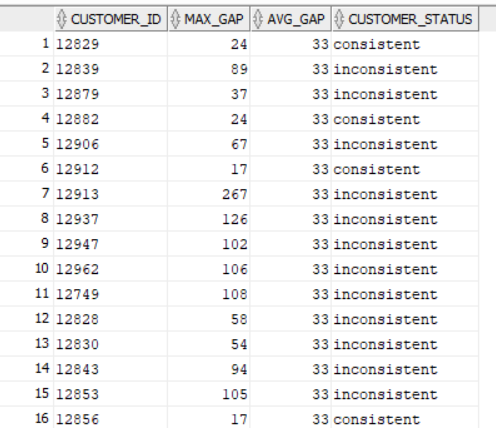
SELECT distinct customer\_id, max\_gap, avg\_gap ,

CASE WHEN max\_gap > avg\_gap THEN 'inconsistent'

ELSE 'consistent'

END AS customer\_status

FROM difference\_between\_days ;



# **- Second Question :**

Our data here is the same dataset of the first question

I calculated the recency , frequency and monetary for each customer to categorize customers into different segmentations to know what is the suitable approach in dealing with each customer based on his segment :

With customer\_sales as (select customer\_id , sum(price \* quantity) as sum\_sales from tableRetail

group by customer\_id) ,

RFM as (select customer\_id ,

round ( ( (select max( TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) ) as last\_date from tableRetail ) - max( TRUNC(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) ) ) , 0 ) as recency ,

count(customer\_id) as frequency , round( percent\_rank() over ( order by sum (quantity \* price )) , 2) as monetary from tableRetail

group by customer\_id ),

RFM\_scores as (

select customer\_id, recency, frequency, monetary, ntile(5) over (order by recency) as r\_score,

ntile(5) over (order by frequency + monetary) as fm\_score

from RFM )

select customer\_id, recency, frequency, monetary, r\_score, fm\_score ,

case

when (r\_score >= 5 and fm\_score >= 5)

or (r\_score >= 5 and fm\_score = 4)

or (r\_score = 4 and fm\_score >= 5) then 'champions'

when (r\_score >= 5 and fm\_score = 2)

or (r\_score = 4 and fm\_score = 2)

or (r\_score = 3 and fm\_score = 3)

or (r\_score = 4 and fm\_score >= 3) then 'potential

loyalists'

when (r\_score >= 5 and fm\_score = 3)

or (r\_score = 4 and fm\_score = 4)

or (r\_score = 3 and fm\_score >= 5)

or (r\_score = 3 and fm\_score >= 4) then 'loyal

customers'

when r\_score >= 5 and fm\_score = 1 then 'recent

customers'

when (r\_score = 4 and fm\_score = 1)

or (r\_score = 3 and fm\_score = 1) then 'promising'

when (r\_score = 3 and fm\_score = 2)

or (r\_score = 2 and fm\_score = 3)

or (r\_score = 2 and fm\_score = 2) then 'customers

needing attention'

when (r\_score = 2 and fm\_score >= 5)

or (r\_score = 2 and fm\_score = 4)

or (r\_score = 1 and fm\_score = 3) then 'at risk'

when (r\_score = 1 and fm\_score >= 5)

or (r\_score = 1 and fm\_score = 4) then 'cant lose

them'

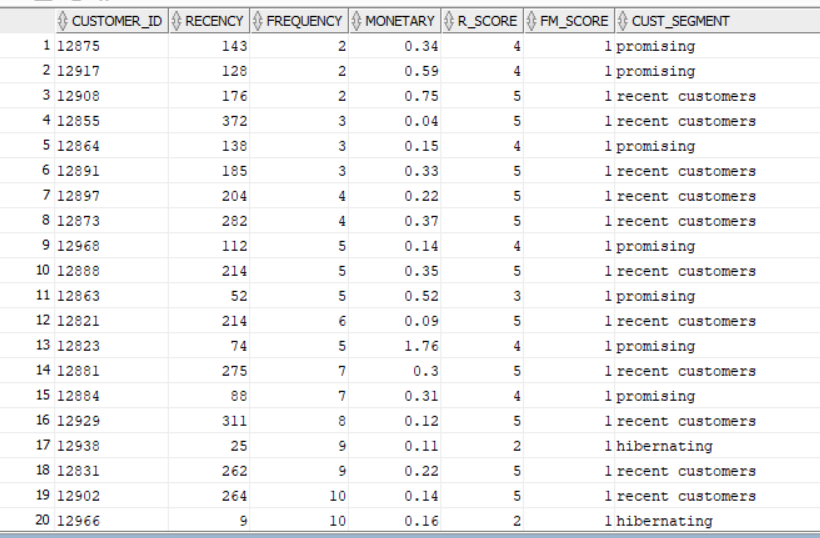
when (r\_score = 1 and fm\_score = 2)

or (r\_score = 2 and fm\_score = 1) then 'hibernating'

when r\_score = 1 and fm\_score <= 1 then 'lost'

else 'other'

end as cust\_segment from RFM\_scores ;



# **-Third Question :**

Our data here is transactions data which has 574396 row ,.. it contains : cust\_id , calendar\_dt , amt\_l

The question contains 2 parts :

1 - What is the maximum number of consecutive days a customer made purchases?

WITH customer\_dates AS (

SELECT cust\_id,

TO\_DATE(calendar\_dt, 'YYYY-MM-DD') AS date\_value,

ROW\_NUMBER() OVER (PARTITION BY cust\_id ORDER BY TO\_DATE(calendar\_dt, 'YYYY-MM-DD')) AS rn

FROM test

) ,

customer\_intervals as (

SELECT cust\_id, MIN(date\_value) AS start\_date,

MAX(date\_value) AS end\_date,

COUNT(\*) AS sequential\_days

FROM (

SELECT cust\_id, date\_value, date\_value - rn AS grp

FROM customer\_dates

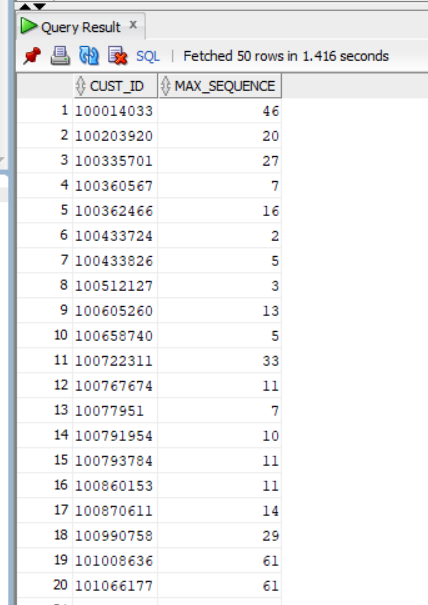
)

GROUP BY cust\_id, grp

ORDER BY cust\_id, start\_date )

select cust\_id , max(sequential\_days) as max\_sequence from customer\_intervals

group by cust\_id ;



2 -On average, How many days/transactions does it take a customer to reach a spent threshold of 250 L.E?

with customers\_sales as (select cust\_id , calendar\_dt ,count(calendar\_dt) over (partition by cust\_id order by TO\_DATE(calendar\_dt, 'YYYY-MM-DD')) as count\_days , amt\_l ,

sum(amt\_l) over (partition by cust\_id order by TO\_DATE(calendar\_dt, 'YYYY-MM-DD')) as total\_amount

from test ) ,

high\_amounts as (

select cust\_id , count\_days ,total\_amount from customers\_sales

where total\_amount >=250 )

select round ( avg (count\_days) )as avg\_days from high\_amounts

where (cust\_id, total\_amount) IN (select cust\_id , min(total\_amount) from high\_amounts

group by cust\_id)

order by cust\_id ;

