ONLINE RETAIL

Analytical Case Study

Meran Ehab

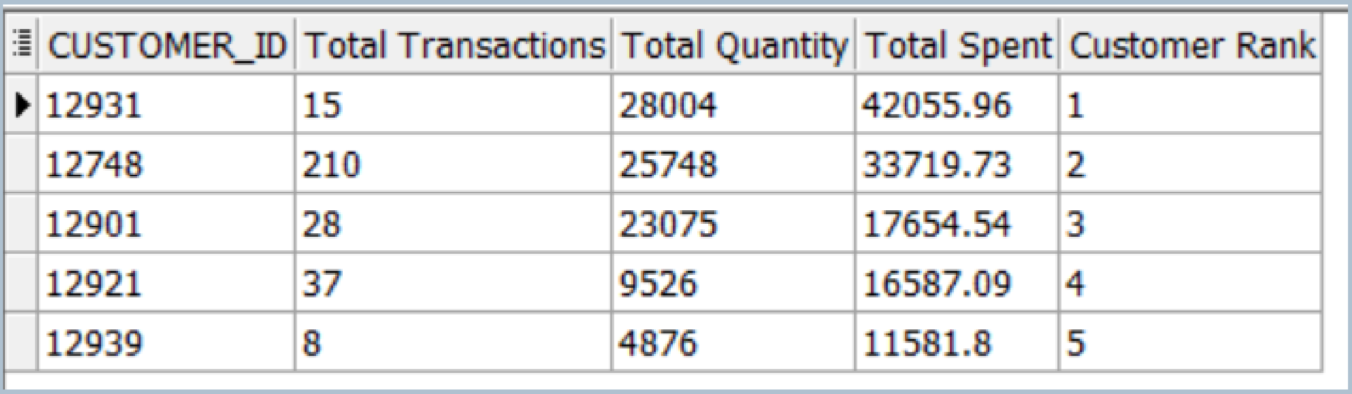
Question 1:

1. Top 5 Customers

```
select Customer_ID, ”Total Transactions”,
   ”Total Quantity”, ”Total Spent”, ”Customer Rank”
from (select Customer_ID,
            count(Distinct Invoice) ”Total Transactions”,
            sum(Quantity) ”Total Quantity”,
            sum(Quantity * Price) ”Total Spent”,
            row_number() over (order by sum(Quantity * Price) desc) ”Customer Rank”
      from tableRetail
      group by Customer_ID
      order by ”Total Spent” desc)
where ”Customer Rank” <= 5;
```

Business Description:

This query identifies the top five customers based on their total spending. It provides insights into which customers contribute the most revenue to the business.



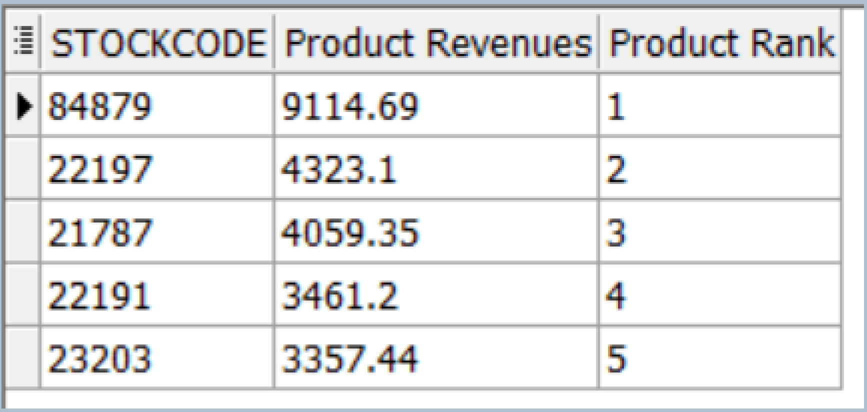
CUSTOMER_ID	Total Transactions	Total Quantity	Total Spent	Customer Rank
12931	15	28004	42055.96	1
12748	210	25748	33719.73	2
12901	28	23075	17654.54	3
12921	37	9526	16587.09	4
12939	8	4876	11581.8	5

2. Top 5 Selling Products:

```
select distinct StockCode, ”Product Revenues”, ”Product Rank”
from (select distinct StockCode,
             sum(quantity * price) ”Product Revenues”,
             rank() over(order by sum(quantity * price) desc  ) ”Product Rank”
       from tableRetail
       group by StockCode)
where ”Product Rank” <= 5;
```

Business Description:

This query identifies the top five selling products based on their total revenues. It helps understand which products are the most popular and profitable, informing inventory management and marketing strategies.



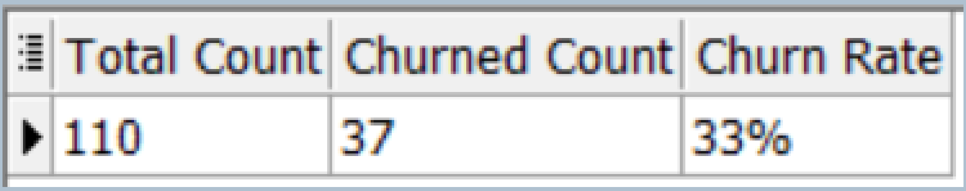
STOCKCODE	Product Revenues	Product Rank
84879	9114.69	1
22197	4323.1	2
21787	4059.35	3
22191	3461.2	4
23203	3357.44	5

3. Churn Rate by Customer Count:

```
with Churned_Customers as (
select count(Customer_ID) ”Churned Count”
from (select Customer_ID, trunc((”Churned” / 30)) ”Rate”
     from (select Customer_ID, (”Reference Date” - ”Last Value”) ”Churned”
           from (select Distinct Customer_ID,
                       last_value(to_date(invoicedate,’mm/dd/yyyy hh24:mi’))
over(order by to_date(invoicedate,’mm/dd/yyyy hh24:mi’) range between
unbounded preceding and unbounded following) ”Reference Date”,
                       last_value(to_date(invoicedate,’mm/dd/yyyy hh24:mi’))
over(partition by customer_id order by to_date(invoicedate,’mm/dd/yyyy
hh24:mi’) range between unbounded preceding and unbounded following) ”Last
Value”
                 from tableRetail))
where ”Churned” / 30 >= 3)
),
Total_Customers as (
  select count(distinct Customer_ID) ”Total Count”
  from tableRetail
)
select ”Total Count”, ”Churned Count” , trunc((”Churned Count” / ”Total
Count”)*100) ||’’|| ’%’ ”Churn Rate”
from Churned_Customers, Total_Customers;
```

Business Description:

This query calculates the churn rate, the rate at which customers stop purchasing, based on a specified period. It provides insights into customer retention and helps in devising strategies to reduce churn.



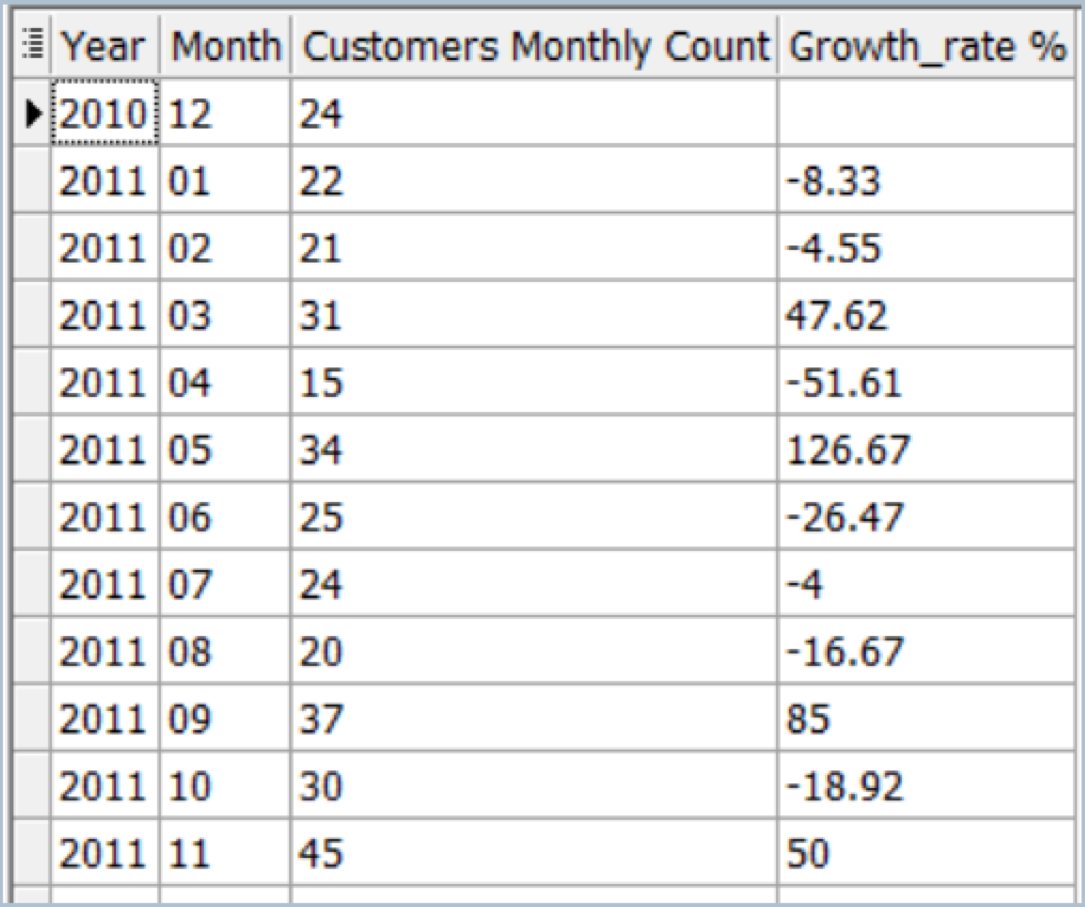
Total Count	Churned Count	Churn Rate
110	37	33%

4. Growth Rate By Customer Count Every Month:

```
with Monthly_Customer_Count as (
select
      to_char(to_date(invoicedate ,’mm/dd/yyyy HH24:MI’), ’yyyy’) ”Year” ,
      to_char(to_date(invoicedate ,’mm/dd/yyyy HH24:MI’), ’mm’) ”Month”,
      count(Distinct customer_id) ”Customers Monthly Count”
      from tableRetail
 group by  to_char(to_date(invoicedate ,’mm/dd/yyyy HH24:MI’), ’yyyy’) ,
to_char(to_date(invoicedate ,’mm/dd/yyyy HH24:MI’), ’mm’)
                ) ,
Previous_Month_Customer_Count as (
select ”Year”, ”Month”, ”Customers Monthly Count”,
      lag(”Customers Monthly Count”) over (order by ”Year”, ”Month”) ”Previous
Month Customer Count”
from Monthly_Customer_Count
)
select ”Year”, ”Month”, ”Customers Monthly Count”, round((”Customers Monthly
Count” - ”Previous Month Customer Count”) / ”Previous Month Customer Count” *
100 , 2) as ”Growth_rate %”
from Previous_Month_Customer_Count;
```

Business Description:

This query calculates the monthly growth rate in the number of customers. It helps in understanding the business’s customer acquisition trends over time, which is crucial for forecasting and resource allocation.



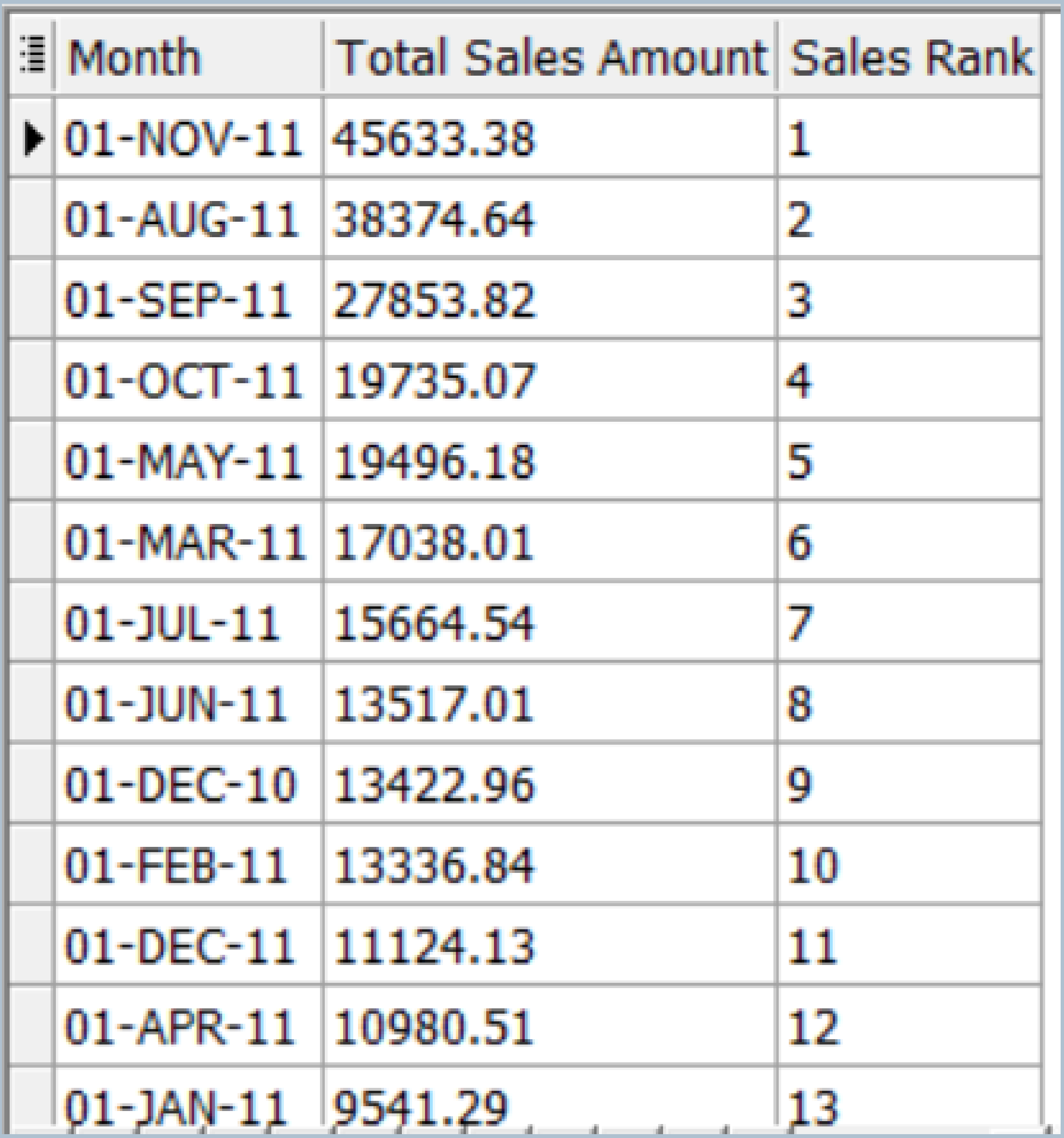
Year	Month	Customers Monthly Count	Growth_rate %
2010	12	24	
2011	01	22	-8.33
2011	02	21	-4.55
2011	03	31	47.62
2011	04	15	-51.61
2011	05	34	126.67
2011	06	25	-26.47
2011	07	24	-4
2011	08	20	-16.67
2011	09	37	85
2011	10	30	-18.92
2011	11	45	50

5. Time Series Analysis By Month:

```
select
   trunc(to_date(InvoiceDate, ’mm/dd/yyyy hh24:mi’), ’month’) ”Month”,
   sum(Price * Quantity) ”Total Sales Amount”,
   rank() over (order by sum(Price * Quantity) desc) ”Sales Rank”
from tableRetail
group by trunc(to_date(InvoiceDate, ’mm/dd/yyyy hh24:mi’), ’month’)
order by ”Sales Rank”;
```

Business Description:

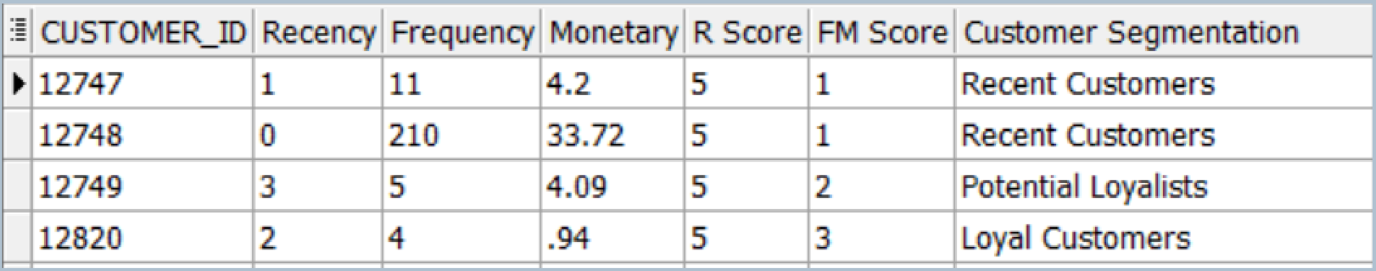
This query performs a time series analysis of total sales amount by month. It helps identify sales trends over time, such as monthly fluctuations or growth patterns, enabling informed decision-making and forecasting.



Month	Total Sales Amount	Sales Rank
01-NOV-11	45633.38	1
01-AUG-11	38374.64	2
01-SEP-11	27853.82	3
01-OCT-11	19735.07	4
01-MAY-11	19496.18	5
01-MAR-11	17038.01	6
01-JUL-11	15664.54	7
01-JUN-11	13517.01	8
01-DEC-10	13422.96	9
01-FEB-11	13336.84	10
01-DEC-11	11124.13	11
01-APR-11	10980.51	12
01-JAN-11	9541.29	13

Question 2:

```
select Customer_ID, ”Recency”, ”Frequency”, ”Monetary”, ”R Score”, ”FM Score”,
     case
     when ”R Score” = 5 and ”FM Score” in (5, 4) then ’Champions’
     when ”R Score” = 4 and ”FM Score” = 5 then ’Champions’
     when ”R Score” = 5 and ”FM Score” = 2 then ’Potential Loyalists’
     when ”R Score” = 4 and ”FM Score” in (2 , 3) then ’Potential Loyalists’
     when ”R Score” = 3 and ”FM Score” = 3 then ’Potential Loyalists’
     when ”R Score” = 5 and ”FM Score” = 3 then ’Loyal Customers’
     when ”R Score” = 4 and ”FM Score” = 4 then ’Loyal Customers’
     when ”R Score” = 3 and ”FM Score” in (4 , 5) then ’Loyal Customers’
     when ”R Score” = 5 and ”FM Score” = 1 then ’Recent Customers’
     when ”R Score” = 4 and ”FM Score” = 1 then ’Promising’
     when ”R Score” = 3 and ”FM Score” = 1 then ’Promising’
     when ”R Score” = 3 and ”FM Score” = 2 then ’Customers Needing Attention’
     when ”R Score” = 2 and ”FM Score” in (2, 3) then ’Customers Needing
Attention’
     when ”R Score” = 1 and ”FM Score” = 3 then ’At Risk’
     when ”R Score” = 2 and ”FM Score” in (4, 5) then ’At Risk’
     when ”R Score” = 1 and ”FM Score” = 2 then ’Hibernating’
     when ”R Score” = 1 and ”FM Score” in (4, 5) then ’Cant Lose Them’
     when ”R Score” = 1 and ”FM Score” = 1 then ’Lost’
     else ’Undefined’
   end ”Customer Segmentation”
from (select Customer_ID, ”Recency”, ”Frequency”, ”Monetary”, ntile(5)
over(order by ”Recency” desc) ”R Score”, round(( ”Frequency Ntile” + ”Monetary
Ntile”)/2) ”FM Score”
        from (select Customer_ID, ”Recency”, ”Frequency”, ”Monetary”, ntile(5)
over(order by ”Frequency” desc) ”Frequency Ntile”, ntile(5) over(order by
”Monetary” desc) ”Monetary Ntile”
              from (select Customer_ID, trunc(”Reference Date” - ”Last Value”)
”Recency”, count(distinct invoice) ”Frequency”,
                        round(sum(quantity * price) / 1000, 2) ”Monetary”
                     from (select Customer_ID, Price, Quantity, Invoice,
                              last_value(to_date(invoicedate,’mm/dd/yyyy
hh24:mi’)) over(order by to_date(invoicedate,’mm/dd/yyyy hh24:mi’) range
between unbounded preceding and unbounded following) ”Reference Date”,
                              last_value(to_date(invoicedate,’mm/dd/yyyy
hh24:mi’)) over(partition by customer_id order by
to_date(invoicedate,’mm/dd/yyyy hh24:mi’) range between unbounded preceding
and unbounded following) ”Last Value”
                          from tableRetail)
                     group by customer_id, ”Last Value”, ”Reference Date”))
        order by customer_id);
```

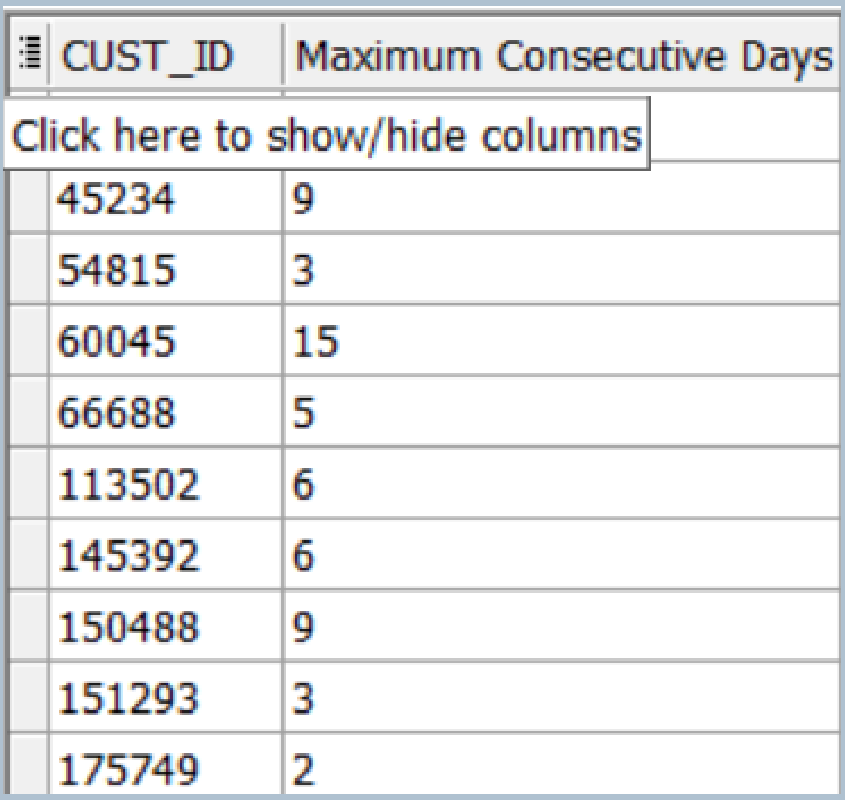


CUSTOMER_ID	Recency	Frequency	Monetary	R Score	FM Score	Customer Segmentation
12747	1	11	4.2	5	1	Recent Customers
12748	0	210	33.72	5	1	Recent Customers
12749	3	5	4.09	5	2	Potential Loyalists
12820	2	4	.94	5	3	Loyal Customers

Question 3:

a.

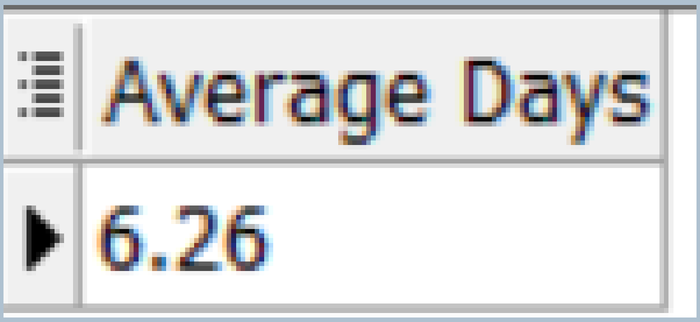
```
with Difference_Date as(
select Cust_Id,  Calendar_Dt, (Calendar_Dt  - ”Rank”) ”Date Difference”
from (select Cust_Id , Calendar_Dt ,
                row_number() over(partition by cust_id  order by
calendar_dt) ”Rank”
        from CustomerDailyTransactions)
)

select Cust_Id , max(”Consecutive Days”) ”Maximum Consecutive Days”
from(
       select Cust_Id , count(”Date Difference”) ”Consecutive Days”
       from Difference_Date
       group by Cust_Id , ”Date Difference”
       )
group by Cust_Id
order by Cust_Id;
```



CUST_ID	Maximum Consecutive Days
45234	9
54815	3
60045	15
66688	5
113502	6
145392	6
150488	9
151293	3
175749	2

b.

```
select Round(avg(”Count Days”),2) ”Average Days”
   from  (
   select Cust_Id, min(case when ”Total Amount” >= 250 then ”Rank” end )
”Count Days”
      from (
      select Cust_Id,”Total Amount” , row_number() over(partition by
Cust_Id order by ”Total Amount”) ”Rank”
         from (
         select Cust_Id, sum(Amt_LE) over(partition by Cust_Id order by
Calendar_Dt) ”Total Amount”
            from CustomerDailyTransactions
            ))
    group by Cust_Id ) ;
```



Average Days
6.26