Analytical SQL Case Study

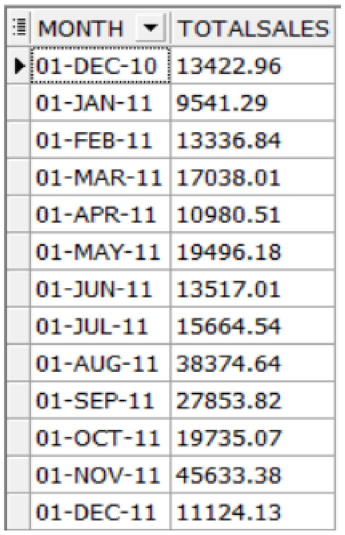
Customers Transaction Dataset

Q1:

Customers have purchasing transactions that we shall be monitoring to get intuition behind each customer behavior to target the customers in the most efficient and proactive way, to increase sales/revenue, to improve customer retention and decrease churn, so:

1: We will track the **Total Sales Over Time:**

```
SELECT
   DISTINCT TO_DATE(TO_CHAR(TO_DATE(INVOICEDATE, 'MM/DD/YYYY'),
'MM/YYYY'), 'MM/YYYY') AS MONTH,
   SUM(QUANTITY * PRICE) OVER (PARTITION BY
TO_CHAR(TO_DATE(INVOICEDATE, 'MM/DD/YYYY'), 'MM/YYYY')) AS TOTALSALES
FROM
   TABLERETAIL
ORDER BY
   MONTH;
```

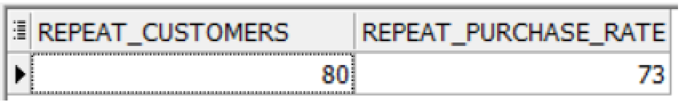


MONTH	TOTALSALES
01-DEC-10	13422.96
01-JAN-11	9541.29
01-FEB-11	13336.84
01-MAR-11	17038.01
01-APR-11	10980.51
01-MAY-11	19496.18
01-JUN-11	13517.01
01-JUL-11	15664.54
01-AUG-11	38374.64
01-SEP-11	27853.82
01-OCT-11	19735.07
01-NOV-11	45633.38
01-DEC-11	11124.13

This query helps us to examine the monthly sales trends, revealing fluctuations and peaks in sales over time. We observed a spike in sales during certain months, indicating potential seasonal patterns or marketing campaign effectiveness, and we can apply more discounts in the months with low sales to increase the revenues in it.

2: We can check the **Repeat Purchase Rate Analysis:**

```
WITH Customer_Purchase_Counts AS (
   SELECT
      COUNT(CASE WHEN Purchase_Count > 1 THEN Customer_ID END) AS Repeat_Customers,
      COUNT(*) AS Total_Customers,
      (COUNT(CASE WHEN Purchase_Count > 1 THEN Customer_ID END) * 100.0) / COUNT(*) AS
Repeat_Purchase_Rate
   FROM (

  SELECT
          Customer_ID,
          COUNT(DISTINCT Invoice) AS Purchase_Count
      FROM
          tableRetail
      GROUP BY
          Customer_ID
   )
)

SELECT
   Repeat_Customers,
   round(Repeat_Purchase_Rate) as Repeat_Purchase_Rate
FROM
   Customer_Purchase_Counts;
```



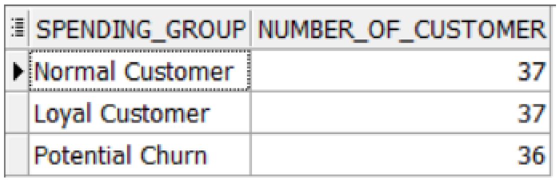
REPEAT_CUSTOMERS	REPEAT_PURCHASE_RATE
80	73

This query allows us to delve into customer behavior by analyzing the repeat purchase rate. This metric provides insights into customer loyalty and retention.

We found that a significant percentage of customers are repeat purchasers, suggesting strong customer engagement and satisfaction.

3: We can divide **Customer to Groups**:

```
select Spending_Group, count(*)
from(
SELECT CUSTOMER_ID, TOTAL_SPENDING,
      CASE SPENDINGRANK
          WHEN 1 THEN 'Loyal Customer'
          WHEN 2 THEN 'Normal Customer'
          WHEN 3 THEN 'Potential Churn'
          ELSE 'Unknown'
      END AS SPENDING_GROUP
FROM (

 SELECT CUSTOMER_ID,
          SUM(QUANTITY * PRICE) AS TOTAL_SPENDING,
          NTILE(3) OVER (ORDER BY SUM(QUANTITY * PRICE) DESC) AS SPENDINGRANK
    FROM TABLERETAIL
    GROUP BY CUSTOMER_ID
) )
GROUP BY SPENDING_GROUP;
```



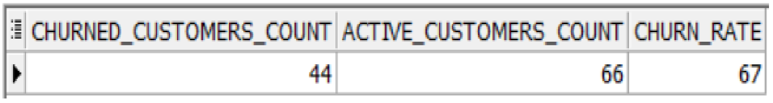
SPENDING_GROUP	NUMBER_OF_CUSTOMER
Normal Customer	37
Loyal Customer	37
Potential Churn	36

We use this query to explore customer behavior, we segmented customers into spending groups based on their total spending. This segmentation helps identify high-value customers (Loyal), regular spenders (Normal), and those showing signs of potential churn.

By understanding spending patterns, we can tailor marketing strategies to retain high-value customers and re-engage potential churners.

4: We can check the **Churn Rate**:

```
WITH ActiveCustomers AS (
    SELECT DISTINCT Customer_ID
    FROM tableRetail
    WHERE TO_DATE(InvoiceDate, 'MM/DD/YYYY') BETWEEN TO_DATE('01/01/2011', 'MM/DD/YYYY')
AND TO_DATE('06/30/2011', 'MM/DD/YYYY')),
ChurnedCustomers AS (
    SELECT DISTINCT Customer_ID
    FROM tableRetail
    WHERE TO_DATE(InvoiceDate,
'MM/DD/YYYY') BETWEEN
TO_DATE('07/01/2011', 'MM/DD/YYYY') AND
TO_DATE('12/31/2011', 'MM/DD/YYYY'))
SELECT
    COUNT(c.Customer_ID) AS Churned_Customers_Count,
    COUNT(a.Customer_ID) AS Active_Customers_Count,
    round(100 * COUNT(c.Customer_ID) / COUNT(a.Customer_ID)) AS Churn_Rate
FROM ActiveCustomers a LEFT JOIN ChurnedCustomers c ON a.Customer_ID = c.Customer_ID;
```

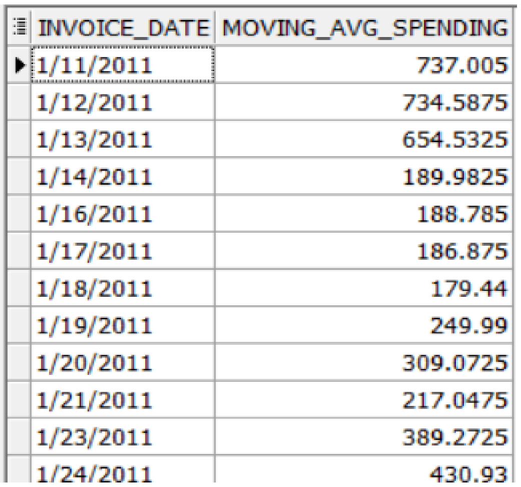


CHURNED_CUSTOMERS_COUNT	ACTIVE_CUSTOMERS_COUNT	CHURN_RATE
44	66	67

Understanding churn is crucial for business sustainability, Analyzing the churn rate revealed the percentage of customers lost over a specific period. We found that while most customers remained active, there was a notable churn rate.
Identifying the reasons behind churn and implementing retention strategies can mitigate customer loss and improve long-term profitability.

5: we can calculate the **Moving Average Spending:**

```
SELECT TO_DATE(INVOICEDATE, 'MM/DD/YYYY') AS INVOICE_DATE ,
    AVG(SUM(QUANTITY * PRICE)) OVER (ORDER BY TO_DATE(INVOICEDATE, 'MM/DD/YYYY') ROWS
    BETWEEN 3 PRECEDING AND CURRENT ROW) AS MOVING_AVG_SPENDING
FROM
        TABLERETAIL
GROUP BY
        INVOICEDATE
ORDER BY INVOICEDATE;
```



INVOICE_DATE	MOVING_AVG_SPENDING
1/11/2011	737.005
1/12/2011	734.5875
1/13/2011	654.5325
1/14/2011	189.9825
1/16/2011	188.785
1/17/2011	186.875
1/18/2011	179.44
1/19/2011	249.99
1/20/2011	309.0725
1/21/2011	217.0475
1/23/2011	389.2725
1/24/2011	430.93

Lastly, we analyzed the moving average spending to identify trends in customer spending behavior more effectively over time.

This smoothed out short-term fluctuations, providing insights into underlying spending patterns and potential changes in customer preferences.

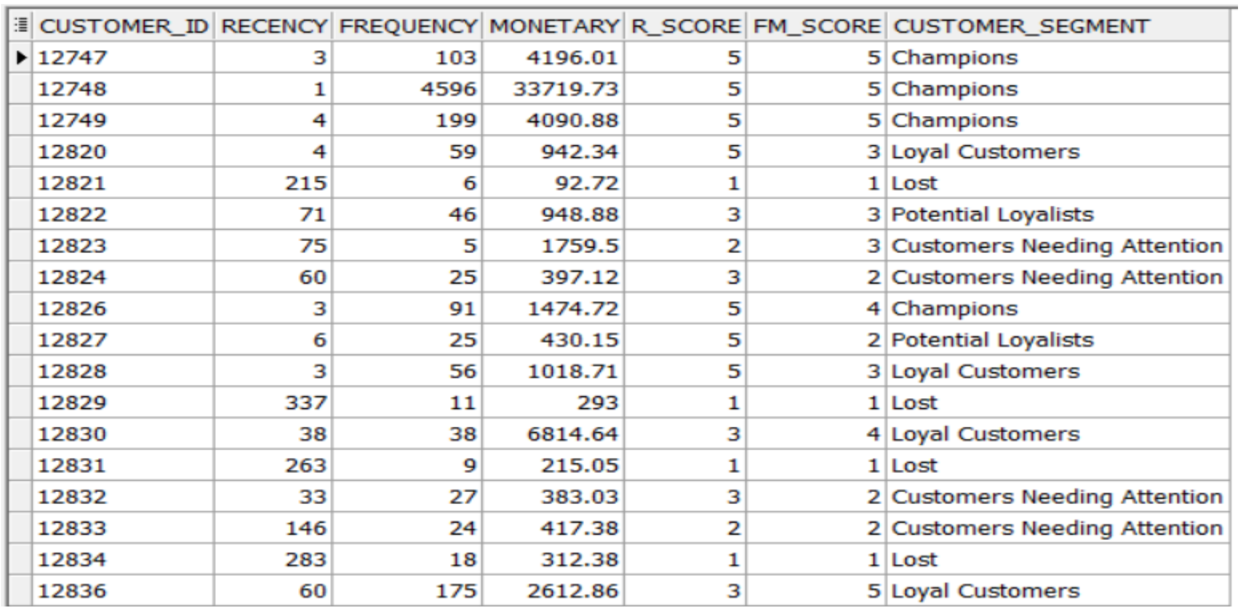
In conclusion, by leveraging insights from these analyses, businesses can develop proactive strategies to increase sales, improve customer retention, and reduce churn, ultimately driving long-term growth and success.

Q2:

After exploring the data now, you are required to implement a Monetary model for customers behavior for product purchasing and segment each customer based on the below groups:

Champions - Loyal Customers - Potential Loyalists – Recent Customers – Promising - Customers Needing Attention - At Risk - Can't Lose Them – Hibernating – Lost

```
WITH RFM_CTE AS (
    SELECT DISTINCT
        CUSTOMER_ID,
        TO_DATE('2011-12-10', 'YYYY-MM-DD') - MAX(TO_DATE(INVOICEDATE,'MM/DD/YYYY')) OVER
(PARTITION BY CUSTOMER_ID) AS RECENCY,
        COUNT(INVOICE) OVER (PARTITION BY CUSTOMER_ID) AS FREQUENCY,
        SUM(PRICE * QUANTITY) OVER (PARTITION BY CUSTOMER_ID) AS MONETARY
    FROM
        TABLERETAIL
)

SELECT
    CUSTOMER_ID,
    RECENCY,
    FREQUENCY,
    MONETARY,
    R_SCORE,
    FM_SCORE,
    CASE
        WHEN R_SCORE = 5 AND FM_SCORE = 5 THEN 'Champions'
        WHEN R_SCORE = 5 AND FM_SCORE = 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 = 2 THEN 'Potential Loyalists'
        WHEN R_SCORE = 3 AND FM_SCORE = 3 THEN 'Potential Loyalists'
        WHEN R_SCORE = 4 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 = 5 THEN 'Loyal Customers'
        WHEN 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 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 = 3 THEN 'Customers Needing Attention'
        WHEN R_SCORE = 2 AND FM_SCORE = 2 THEN 'Customers Needing Attention'

        WHEN R_SCORE = 2 AND FM_SCORE = 5 THEN 'At Risk'
        WHEN R_SCORE = 2 AND FM_SCORE = 4 THEN 'At Risk'
        WHEN R_SCORE = 1 AND FM_SCORE = 3 THEN 'At Risk'
        WHEN R_SCORE = 1 AND FM_SCORE = 5 THEN 'Cant Lose Them'
        WHEN R_SCORE = 1 AND FM_SCORE = 4 THEN 'Cant Lose Them'

        WHEN R_SCORE = 1 AND FM_SCORE = 2 THEN 'Hibernating'

        WHEN R_SCORE = 1 AND FM_SCORE = 1 THEN 'Lost'

        ELSE 'Unknown'
    END AS CUSTOMER_SEGMENT

FROM (
    SELECT
        CUSTOMER_ID,
        RECENCY,
        FREQUENCY,
        MONETARY,
        R_SCORE,
        NTILE(5) OVER (ORDER BY ROUND( (F_SCORE + M_SCORE) / 2)) AS FM_SCORE
    FROM (
        SELECT
            CUSTOMER_ID,
            RECENCY,
            FREQUENCY,
            MONETARY,
            NTILE(5) OVER (ORDER BY RECENCY DESC) AS R_SCORE,
            NTILE(5) OVER (ORDER BY FREQUENCY) AS F_SCORE,
            NTILE(5) OVER (ORDER BY MONETARY) AS M_SCORE
        FROM
            RFM_CTE
    )
)
 ORDER BY CUSTOMER_ID
```

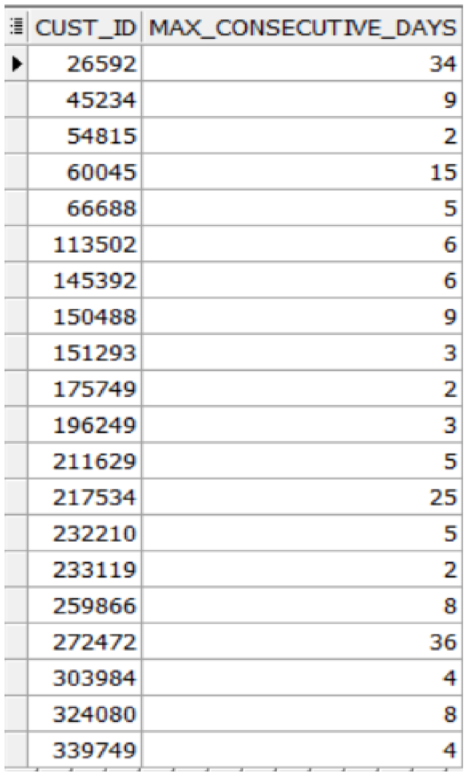


CUSTOMER_ID	RECENCY	FREQUENCY	MONETARY	R_SCORE	FM_SCORE	CUSTOMER_SEGMENT
12747	3	103	4196.01	5	5	Champions
12748	1	4596	33719.73	5	5	Champions
12749	4	199	4090.88	5	5	Champions
12820	4	59	942.34	5	3	Loyal Customers
12821	215	6	92.72	1	1	Lost
12822	71	46	948.88	3	3	Potential Loyalists
12823	75	5	1759.5	2	3	Customers Needing Attention
12824	60	25	397.12	3	2	Customers Needing Attention
12826	3	91	1474.72	5	4	Champions
12827	6	25	430.15	5	2	Potential Loyalists
12828	3	56	1018.71	5	3	Loyal Customers
12829	337	11	293	1	1	Lost
12830	38	38	6814.64	3	4	Loyal Customers
12831	263	9	215.05	1	1	Lost
12832	33	27	383.03	3	2	Customers Needing Attention
12833	146	24	417.38	2	2	Customers Needing Attention
12834	283	18	312.38	1	1	Lost
12836	60	175	2612.86	3	5	Loyal Customers

Q3:

a. What is the maximum number of consecutive days a customer makes purchases?

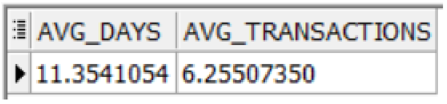
```
WITH PURCHASE_DAYS AS (
    SELECT
        CUST_ID,
        CALENDAR_DT,
        ROW_NUMBER() OVER (PARTITION BY CUST_ID ORDER BY CALENDAR_DT) AS RN,
        CALENDAR_DT - ROW_NUMBER() OVER (PARTITION BY CUST_ID ORDER BY CALENDAR_DT) AS
GRP
    FROM
        CUSTOMERS
    WHERE
        AMT_LE > 0
)
SELECT DISTINCT CUST_ID, MAX(CONSECUTIVE_DAYS) OVER (PARTITION BY CUST_ID) AS
MAX_CONSECUTIVE_DAYS
FROM
(SELECT
    CUST_ID, GRP,
    ROW_NUMBER() OVER (PARTITION BY CUST_ID, GRP ORDER BY CALENDAR_DT) AS
CONSECUTIVE_DAYS
FROM
    PURCHASE_DAYS)
ORDER BY CUST_ID;
```



CUST_ID	MAX_CONSECUTIVE_DAYS
26592	34
45234	9
54815	2
60045	15
66688	5
113502	6
145392	6
150488	9
151293	3
175749	2
196249	3
211629	5
217534	25
232210	5
233119	2
259866	8
272472	36
303984	4
324080	8
339749	4

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

```
WITH TOTAL_SPENDING AS(
   SELECT  CUST_ID,
           CALENDAR_DT,
         SUM(AMT_LE ) OVER(PARTITION BY CUST_ID ORDER BY CALENDAR_DT) AS TOTAL,
         ROW_NUMBER() OVER(PARTITION BY CUST_ID ORDER BY CALENDAR_DT) AS
TRANSACTION_NUMBER,
         MIN(CALENDAR_DT) OVER(PARTITION BY CUST_ID) FIRST_DATE
   FROM CUSTOMERS
)

SELECT AVG(CALENDAR_DT- FIRST_DATE) AS AVG_DAYS,
          AVG(TRANS) AS AVG_TRANSACTIONS
FROM
 (
   SELECT  CUST_ID, MIN(CALENDAR_DT) CALENDAR_DT, FIRST_DATE, MIN(TRANSACTION_NUMBER)
AS TRANS
   FROM
      TOTAL_SPENDING
   WHERE
      TOTAL IN (SELECT MIN(TOTAL) FROM TOTAL_SPENDING WHERE TOTAL >=250 GROUP BY
CUST_ID )
   GROUP BY
      CUST_ID, FIRST_DATE
 )
```



AVG_DAYS	AVG_TRANSACTIONS
11.3541054	6.25507350