**Customer Segmentation:**

**Problem:** How can we segment our customer base to better understand their behavior and needs?

**Questions:** Can you group customers based on their recency, frequency, and monetary value? What are

the different customer segments that emerge from the analysis?

Approach

1. Definition
2. Find tables required
3. Find columns required
4. Client segmentation
5. Visualize

**Solution: Customer Segmentation**

Segmenting a customer base based on recency, frequency, and monetary value is a commonly used technique in marketing and customer relationship management. This segmentation approach is often referred to as RFM analysis.

Recency

**Definition:**

**Recency (R)**: How recently a customer made a purchase or engaged with your business.

**Tables required:** Sales.SalesOrderHeader

**Columns required:**  CustomerID, and OrderDate

1. **Customer ID or Unique Identifier**: An identifier for each customer that allows you to link their recency, frequency, and monetary value to their individual profiles.
2. **Date of Last Interaction**: The date when the customer last interacted with your business, made a purchase, or engaged in any meaningful activity. This will be used to calculate the recency score.

Once you have these columns, the recency scores can be calculated based on the difference between the current date and the date of the last interaction for each customer.

Frequency

**Definition:**

**Frequency (F)**: How often a customer makes purchases or engages with your business.

**Tables required:** Sales.SalesOrderHeader

**Columns required:**  CustomerID, SalesorderID, and OrderDate

1. **Customer ID or Unique Identifier**: An identifier for each customer that allows you to link their recency, frequency, and monetary value to their individual profiles.
2. **Transaction/Purchase ID**: An identifier for each transaction or purchase made by the customer.
3. **Date of Transaction/Purchase**: The date when each transaction or purchase was made. This will be used to calculate the frequency score.

Once you have these columns, you can calculate the frequency score based on the number of transactions or purchases made by each customer within a specified time frame.

Monetary Value (still need to check the tables & colums)

**Definition:**

**Monetary Value (M)**: The amount of money a customer has spent on purchases.

**Tables required:** Sales.SalesOrderHeader & Sales.SalesOrderDetail

**Columns required:** CustomerID, SalesOrderID, OrderDate, Product ID, OrderQty, and Unit price

1. **Customer ID or Unique Identifier:** An identifier for each customer that allows you to link their recency, frequency, and monetary value to their individual profiles.
2. **Transaction/Purchase ID:** An identifier for each transaction or purchase made by the customer.
3. **Amount Spent:** The monetary value associated with each transaction or purchase made by the customer. This could be the total amount spent in that transaction.

Once you have these columns, you can calculate the monetary value score based on the total amount spent by each customer over a specified time frame or for their entire history as relevant to your analysis.

**QUERIES**

1. *General table for question 1 containing all columns needed*

This SQL code creates a temporary table **#RFM** by selecting specific columns from the **SalesOrderHeader** and **SalesOrderDetail** tables. It includes the customer ID, order date, sales order ID, product ID, order quantity, unit price, and line total for each sales order.

The **SalesOrderHeader** and **SalesOrderDetail** tables are joined based on the **SalesOrderID** to retrieve the required information for each sales order. The selected columns are then stored in the temporary table **#RFM.**

SELECT a.[CustomerID], a.[OrderDate], a.[SalesOrderID] , b.[ProductID], b.[OrderQty], b.[UnitPrice], b.[LineTotal]

INTO #RFM

FROM [Sales].[SalesOrderHeader] a

INNER JOIN [Sales].[SalesOrderDetail] b

ON a.[SalesOrderID] = b.[SalesOrderID]

NB: You can omit the b parts of the RFM Queries below. The 5th query was broken down in by RFM for better understanding. Everything is in the 5th QUERY

1. **Recency**
2. This SQL code calculates the difference in days between the maximum order date for each customer and the specified date ('2015-01-01'). It stores the customer ID and the date difference in a temporary table **#numberofdays**.

The **SalesOrderHeader** table is used to calculate the maximum order date for each customer. The **DATEDIFF** function computes the difference in days between the maximum order date and '2015-01-01'. The results are then stored in the temporary table **#numberofdays**, including the customer ID and the corresponding date difference.

SELECT [CustomerID], datediff(day, MAX([OrderDate]),'2015-01-01') datedifference

INTO #numberofdays

FROM [Sales].[SalesOrderHeader]

GROUP BY [CustomerID]

ORDER BY MAX([OrderDate]) DESC

1. This SQL code calculates the recency scores (**Recencyscore**) based on the date differences obtained from the **#numberofdays** temporary table. The recency is categorized as 'very recent', 'moderately recent', or 'not so recent', and corresponding scores are assigned. The results are then ordered by the recency scores in ascending order.

The **CASE** statement categorizes the recency, and then the subsequent **CASE** statement assigns scores accordingly. Finally, the results are ordered by the recency scores in ascending order.

WITH Recency1

AS

(SELECT [CustomerID], datedifference,

CASE

    WHEN datedifference < 200 THEN 'very recent'

    WHEN datedifference between 201 and 720 THEN 'moderately recent'

    ELSE 'not so recent'

END AS Recency

FROM #numberofdays

)

SELECT [CustomerID], datedifference, Recency,

CASE

    WHEN Recency = 'very recent' THEN '1'

    WHEN Recency = 'moderately recent' THEN '2'

    ELSE '3'

END AS Recencyscore

FROM Recency1

ORDER BY Recencyscore ASC

1. **Frequency**
2. This SQL code calculates the purchase frequency **(Sumofpurchasecount**) for each customer by counting the number of sales orders they made between '2013-01-01' and '2015-01-01'. The results are grouped by customer, and the total purchase frequency is obtained for each customer. Finally, the results are ordered in descending order based on the total purchase frequency.

The **countfrq** Common Table Expression (CTE) calculates the purchase frequency for each customer based on their distinct sales orders within the specified date range. The final SELECT statement aggregates the purchase frequency for each customer and stores the results in the **#Sumfrq** temporary table, ordered by the total purchase frequency in descending order.

WITH countfrq

AS

(

SELECT [CustomerID], [OrderDate], COUNT([SalesOrderID]) AS purchase\_frequency

FROM [Sales].[SalesOrderHeader]

WHERE [OrderDate] BETWEEN '2013-01-01' AND '2015-01-01'

GROUP BY [CustomerID],[OrderDate]

)

SELECT [CustomerID], sum(purchase\_frequency) Sumofpurchasecount

INTO #Sumfrq

FROM countfrq

GROUP BY [CustomerID]

order by Sumofpurchasecount desc

1. This SQL code calculates the frequency scores (**frequencyscore**) for each customer based on their purchase frequency (**Sumofpurchasecount**). The purchase frequency is categorized as 'High frequency', 'Medium frequency', or 'Low frequency', and corresponding scores are assigned. The results are then ordered by the frequency scores in ascending order.

The CASE statement categorizes the purchase frequency, and then the subsequent CASE statement assigns scores accordingly. Finally, the results are ordered by the frequency scores in ascending order.

WITH Frequency1

AS

(SELECT [CustomerID], Sumofpurchasecount,

CASE

    WHEN Sumofpurchasecount > 20 THEN 'High frequency'

    WHEN Sumofpurchasecount between 11 and 19 THEN 'medium frequency'

    ELSE 'low frequency'

END AS frequency

FROM Sumfrq

)

SELECT [CustomerID], Sumofpurchasecount, frequency,

CASE

    WHEN frequency = 'High frequency' THEN '1'

    WHEN frequency= 'moderate frequency' THEN '2'

    ELSE '3'

END AS frequencyscore

FROM Frequency1

ORDER BY frequencyscore ASC

1. **Monetary Value**
2. This SQL code calculates the total amount spent (**total\_spent**) by each customer within the date range of January 1, 2013, to January 1, 2015, by multiplying order quantity (**[OrderQty]**) with unit price (**[UnitPrice]**). The results are grouped by **[CustomerID]** and stored in a temporary table named **#totalamtspent**.

SELECT [CustomerID], sum([OrderQty] \* [UnitPrice]) total\_spent

INTO #totalamtspent

FROM #RFM

WHERE [OrderDate] BETWEEN '2013-01-01' AND '2015-01-01'

GROUP BY  [CustomerID]

ORDER BY [CustomerID]

1. This SQL code calculates spending scores (**monetaryspendingscore**) for each customer based on their total spending (**total\_spent**). The spending scores range from 1 to 3, with 'High spending' corresponding to 1, 'Moderate spending' to 2, and 'Low spending' to 3. The results are sorted in descending order of total spending.

WITH Monetaryvalue1

AS

(SELECT [CustomerID], total\_spent,

CASE

    WHEN total\_spent > 500000 THEN 'High spending'

    WHEN total\_spent between 210000 and 499999 THEN 'moderate spending'

    ELSE 'low spending'

END AS monetaryvalue

FROM #totalamtspent

)

SELECT [CustomerID], total\_spent, monetaryvalue,

CASE

    WHEN monetaryvalue = 'High spending' THEN '1'

    WHEN monetaryvalue= 'moderate spending' THEN '2'

    ELSE '3'

END AS monetaryspendingscore

FROM Monetaryvalue1

ORDER BY total\_spent desc

1. This SQL code calculates RFM (Recency, Frequency, Monetary Value) scores and corresponding segments for each customer. It begins by calculating recency scores (**Recencyscore**) based on date differences. It then determines frequency scores (**frequencyscore**) based on purchase counts and monetary spending scores (**monetaryspendingscore**) based on total spending. The final SELECT statement combines and displays these scores, segments, and relevant customer information, ordered by recency scores in ascending order.

WITH Recency1

AS

       (

        SELECT [CustomerID], datedifference,

         CASE

           WHEN datedifference < 200 THEN 'very recent'

           WHEN datedifference between 201 and 720 THEN 'moderately recent'

           ELSE 'not so recent'

         END AS Recency

         FROM #numberofdays

         ),

Recency2

AS       (

            SELECT [CustomerID], datedifference, Recency,

            CASE

              WHEN Recency = 'very recent' THEN '1'

              WHEN Recency = 'moderately recent' THEN '2'

              ELSE '3'

              END AS Recencyscore

              FROM Recency1

         ),

Frequency1

AS

(

    SELECT [CustomerID], Sumofpurchasecount,

CASE

    WHEN Sumofpurchasecount > 20 THEN 'High frequency'

    WHEN Sumofpurchasecount between 11 and 19 THEN 'moderate frequency'

    ELSE 'low frequency'

END AS frequency

FROM #Sumfrq

),

Frequency2

AS

(

SELECT [CustomerID], Sumofpurchasecount, frequency,

CASE

    WHEN frequency = 'High frequency' THEN '1'

    WHEN frequency= 'moderate frequency' THEN '2'

    ELSE '3'

END AS frequencyscore

FROM Frequency1

),

Monetaryvalue1

AS

(

    SELECT [CustomerID], total\_spent,

CASE

    WHEN total\_spent > 400000 THEN 'High spending'

    WHEN total\_spent between 210000 and 399999 THEN 'moderate spending'

    ELSE 'low spending'

END AS monetaryvalue

FROM #totalamtspent

),

Monetaryvalue2

AS

(

SELECT [CustomerID], total\_spent, monetaryvalue,

CASE

    WHEN monetaryvalue = 'High spending' THEN '1'

    WHEN monetaryvalue= 'moderate spending' THEN '2'

    ELSE '3'

END AS monetaryspendingscore

FROM Monetaryvalue1

)

SELECT R2.[CustomerID], R2.Recency, R2.Recencyscore, F1.frequency, F2.frequencyscore, M1.monetaryvalue, M2.monetaryspendingscore

INTO #RFManalysis

FROM Recency2 R2

INNER JOIN Frequency2 F2 ON R2.[CustomerID] = F2.[CustomerID]

INNER JOIN Monetaryvalue2 M2 ON F2.[CustomerID] = M2.[CustomerID]

INNER JOIN Frequency1 F1 ON M2.[CustomerID] = F1.[CustomerID]

INNER JOIN Monetaryvalue1 M1 ON F1.[CustomerID] = M1.[CustomerID]

order by R2.Recencyscore ASC

1. This SQL code creates customer segments based on RFM (Recency, Frequency, Monetary Value) criteria. It categorizes customers into segments such as 'High Value Customers', 'Big Spenders', 'Loyal Customers', 'Potential Loyalists', 'Churned Customers', and 'Other' based on their recency, frequency, and monetary value. The results are ordered according to the segmentation categories.

WITH CustomerSegments AS (

    SELECT

        [CustomerID], Recency, Recencyscore, frequency, frequencyscore, monetaryvalue, monetaryspendingscore,

        CASE

            WHEN (recency = 'very recent' OR recency = 'moderately recent') AND (frequency = 'high frequency' OR frequency = 'moderate frequency') AND monetaryvalue = 'high spending' THEN 'High Value Customers'

            WHEN recency = 'moderately recent' and frequency = 'low frequency' and monetaryvalue = 'high spending' THEN 'Big spenders'

            WHEN (recency = 'very recent' OR recency = 'moderately recent') AND (frequency = 'high frequency' OR frequency = 'moderate frequency') AND(monetaryvalue = 'low spending' OR monetaryvalue = 'moderate spending') THEN 'Loyal Customers'

            WHEN (recency = 'very recent' OR recency = 'moderately recent') AND frequency = 'low frequency' AND (monetaryvalue = 'low spending' OR monetaryvalue = 'moderate spending') THEN 'Potential Loyalists'

            WHEN recency = 'not so recent' AND (frequency = 'high frequency' OR frequency = 'moderate frequency') AND monetaryvalue = 'low spending' THEN 'Churned Customers'

            ELSE 'Other'

        END AS customer\_segmentation

    FROM #RFManalysis

)

SELECT [CustomerID], Recency, Recencyscore, frequency, frequencyscore, monetaryvalue, monetaryspendingscore, customer\_segmentation

FROM CustomerSegments

ORDER BY

    CASE

        WHEN customer\_segmentation = 'High Value Customers' THEN 1

        WHEN customer\_segmentation = 'Big spenders'THEN 2

        WHEN customer\_segmentation = 'Loyal Customers' THEN 3

        WHEN customer\_segmentation = 'Potential Loyalists' THEN 4

        WHEN customer\_segmentation = 'Churned Customers' THEN 5

        ELSE 6 -- 'Other'

    END;

**Customer Retention:**

**Problem:** How can we identify customers who are at risk of churning and take proactive measures to retain them?

**Questions:** Which customers have not made a purchase in the last 12 months? Which customers have shown a decline in their purchase frequency or monetary value? How can we reach out to these customers with personalized offers or interventions?