**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?

**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 (R)**

*Definition:* 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 (F)**

*Definition:* 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 (M)**

*Definition:* The amount of money a customer has spent on purchases.

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

*Columns required:*CustomerID, SalesOrderID, OrderDate, 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]

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]),'2014-07-01') datedifference

INTO #numberofdays

FROM [Sales].[SalesOrderHeader]

GROUP BY [CustomerID]

ORDER BY MAX([OrderDate]) DESC

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 '2011-05-31' AND '2014-07-01'

GROUP BY [CustomerID],[OrderDate]

)

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

INTO #Sumfrq

FROM countfrq

GROUP BY [CustomerID]

order by Sumofpurchasecount desc

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([LineTotal]) total\_spent, sum([LineTotal]) Amt\_spent

INTO #totalamtspent

FROM #RFM

WHERE [OrderDate] BETWEEN '2011-05-31' AND '2014-07-01'

GROUP BY  [CustomerID]

ORDER BY [CustomerID]

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 < 100 THEN 'very recent'

WHEN datedifference between 101 and 180 THEN 'recent'

WHEN datedifference between 181 and 365 THEN 'fairly recent'

WHEN datedifference between 366 and 730 THEN 'not recent'

           ELSE 'Inactive'

         END AS Recency

         FROM #numberofdays

         ),

Recency2

AS       (

            SELECT [CustomerID], datedifference, Recency,

            CASE

WHEN Recency = 'very recent' THEN '1'

WHEN Recency = 'recent' THEN '2'

WHEN Recency = 'fairly recent' THEN '3'

WHEN Recency = 'not recent' THEN '4'

WHEN Recency = 'Inactive' THEN '5'

              END AS Recencyscore

              FROM Recency1

         ),

Frequency1

AS

(

    SELECT [CustomerID], Sumofpurchasecount,

CASE

    WHEN Sumofpurchasecount > =20 THEN 'Very High frequency'

    WHEN Sumofpurchasecount between 12 and 19 THEN 'High frequency'

WHEN Sumofpurchasecount between 5 and 11 THEN 'Medium frequency'

WHEN Sumofpurchasecount between 2 and 4 THEN 'Low frequency'

    ELSE 'Very Low frequency'

END AS frequency

FROM #Sumfrq

),

Frequency2

AS

(

SELECT [CustomerID], Sumofpurchasecount, frequency,

CASE

WHEN frequency = 'Very High frequency' THEN '1'

    WHEN frequency = 'High frequency' THEN '2'

    WHEN frequency= 'Medium frequency' THEN '3'

WHEN frequency = 'Low frequency' THEN '4'

    WHEN frequency= 'Very Low frequency' THEN '5'

END AS frequencyscore

FROM Frequency1

),

Monetaryvalue1

AS

(

    SELECT [CustomerID], total\_spent, Amt\_spent,

CASE

    WHEN total\_spent > 500000 THEN 'Very High spending'

    WHEN total\_spent between 100000 and 499999 THEN 'High spending'

WHEN total\_spent between 10000 and 99999 THEN 'Moderate spending'

WHEN total\_spent between 5000 and 9999 THEN 'Low spending'

    ELSE 'Very Low spending'

END AS monetaryvalue

FROM #totalamtspent

),

Monetaryvalue2

AS

(

SELECT [CustomerID], total\_spent, monetaryvalue,

CASE

    WHEN monetaryvalue = 'Very High spending' THEN '1'

    WHEN monetaryvalue= 'HIgh spending' THEN '2'

WHEN monetaryvalue= 'Moderate spending' THEN '3'

WHEN monetaryvalue= 'Low spending' THEN '4'

WHEN monetaryvalue= 'Very Low spending' THEN '5'

END AS monetaryspendingscore

FROM Monetaryvalue1

)

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

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 recencyscore=4 AND frequencyscore=4 AND (monetaryspendingscore=1 OR monetaryspendingscore=2 OR monetaryspendingscore=3 OR monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'At Risk'

            WHEN recencyscore=5 AND frequencyscore=4 AND (monetaryspendingscore=1 OR monetaryspendingscore=2 OR monetaryspendingscore=3 OR monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'Cant lose them'

            WHEN (recencyscore=1 OR recencyscore=2) AND (frequencyscore=1 OR frequencyscore=2 OR frequencyscore=3) AND (monetaryspendingscore=1 OR monetaryspendingscore=2) THEN 'Champion'

            WHEN recencyscore=5 AND frequencyscore=5 AND (monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'Lost'

            WHEN (recencyscore=1 OR recencyscore=2) AND (frequencyscore=1 OR frequencyscore=2 OR frequencyscore=3) AND (monetaryspendingscore=3 OR monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'Loyal'

            WHEN (recencyscore=2 OR recencyscore=3) AND (frequencyscore=3 OR frequencyscore=4) AND (monetaryspendingscore=2 OR monetaryspendingscore=3) THEN 'Potential Loyalist'

            WHEN (recencyscore=2 OR recencyscore=3 OR recencyscore=4) AND (frequencyscore=3 OR frequencyscore=4) AND (monetaryspendingscore=1 OR monetaryspendingscore=2 OR monetaryspendingscore=3 OR monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'Promising'

            WHEN (recencyscore=1 OR recencyscore=2) AND (frequencyscore=4 OR frequencyscore=5) AND (monetaryspendingscore=2 OR monetaryspendingscore=3 OR monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'New Customer'

            WHEN (recencyscore=3 OR recencyscore=4) AND frequencyscore=5 AND (monetaryspendingscore=3 OR monetaryspendingscore=4 OR monetaryspendingscore=5) THEN 'One-off'

            ELSE 'Other'

        END AS customer\_segmentation

    FROM #RFManalysis

)

SELECT

    CS.customer\_segmentation,

    CS.[CustomerID],

    SC.[PersonID],

    P.[FirstName], P.[MiddleName], P.[LastName],

    ts.total\_spent,

    CS.monetaryspendingscore, CS.monetaryvalue, nd.datedifference, CS.Recencyscore,

    CS.Recency,

    sf.Sumofpurchasecount,

    CS.frequencyscore,

    CS.frequency

FROM CustomerSegments CS

JOIN #numberofdays nd ON CS.[CustomerID] = nd.[CustomerID]

JOIN #Sumfrq sf ON nd.[CustomerID] = sf.[CustomerID]

JOIN #totalamtspent ts ON sf.[CustomerID] = ts.[CustomerID]

JOIN [Sales].[Customer] SC ON CS.[CustomerID] = SC.[CustomerID]

JOIN [Person].[Person] P ON SC.[PersonID] = P.[BusinessEntityID]

ORDER BY customer\_segmentation

**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?

*Tables required*

1. [Sales].[SalesOrderHeader] & [sales].[customer]
2. [Sales].[SalesOrderHeader] & #RFM (from prev question)
3. [Sales].[Customer], [Person].[Person], [Person].[EmailAddress], [Person].[BusinessEntityAddress], [Person].[Address], [Person].[StateProvince], [Sales].[SalesTerritory]

**Solution: Customer Retention:**

1. **Which customers have not made a purchase in the last 12 months?**

WITH Purchase\_Last\_12\_Months

AS

(SELECT a.[CustomerID],  d.[FirstName], d.[MiddleName], d.[LastName], datediff(DAY,Max(a.[OrderDate]),'2014-07-01') datedifference

FROM [Sales].[SalesOrderHeader] a

    INNER JOIN [Sales].[Customer] c

    ON a.[CustomerID] = c.[CustomerID]

    INNER JOIN [Person].[Person] d

    ON c.[PersonID] = d.[BusinessEntityID]

GROUP BY a.[CustomerID] , d.[FirstName], d.[MiddleName], d.[LastName]

)

SELECT [CustomerID], [FirstName], [MiddleName], [LastName],

CASE

    WHEN datedifference > 366 THEN 'Over 12 Months'

    WHEN datedifference between 183 AND 366 THEN '6 to 12 Months'

    WHEN datedifference < 183 THEN 'Less than 6 Months'

END AS Purchase\_Recency

FROM Purchase\_Last\_12\_Months

GROUP BY [CustomerID], [FirstName], [MiddleName], [LastName], datedifference

1. **Which customers have shown a decline in their purchase frequency or monetary value?**

This code analyzes customer purchase behavior for the years 2013 and 2014. It calculates purchase frequency and monetary value for each customer in these years and identifies those who showed a decline in either spending or purchase frequency. The final results display customer IDs, spending for both years, the decline in spending, purchase frequency for both years, and the decline in purchase frequency, ordered by customer ID in descending order.

WITH countfrq AS (

    SELECT

        [CustomerID],

        YEAR([OrderDate]) AS order\_year,

        COUNT([SalesOrderID]) AS purchase\_frequency

    FROM

        [Sales].[SalesOrderHeader]

    WHERE

        YEAR([OrderDate]) IN (2013, 2014)

    GROUP BY

        [CustomerID], YEAR([OrderDate])

),

monetary\_value AS (

    SELECT

        [CustomerID],

        SUM(CASE WHEN YEAR([OrderDate]) = 2013 THEN ([OrderQty] \* [UnitPrice])/12 ELSE 0 END) AS Avg\_Monthly\_Spend\_2013,

        SUM(CASE WHEN YEAR([OrderDate]) = 2014 THEN ([OrderQty] \* [UnitPrice])/6 ELSE 0 END) AS Avg\_Monthly\_Spend\_2014

    FROM

        #RFM

    WHERE

        YEAR([OrderDate]) IN (2013, 2014)

    GROUP BY

        [CustomerID]

),

sum\_purchase\_frequency

AS

(

SELECT

    cf.[CustomerID],

    SUM(CASE WHEN order\_year = 2013 THEN purchase\_frequency ELSE 0 END) AS Purchase\_Frequency2013,

    SUM(CASE WHEN order\_year = 2014 THEN purchase\_frequency ELSE 0 END) AS Purchase\_Frequency2014

FROM

    countfrq cf

GROUP BY

        [CustomerID]

)

SELECT spf.[CustomerID],

       mv.Avg\_Monthly\_Spend\_2013,

       mv.Avg\_Monthly\_Spend\_2014,

      (mv.Avg\_Monthly\_Spend\_2014 - mv.Avg\_Monthly\_Spend\_2013) Avg\_Monthly\_Spending\_Decline,

       spf.Purchase\_Frequency2013, spf.Purchase\_Frequency2014,

      (spf.Purchase\_Frequency2014 - spf.Purchase\_Frequency2013) Purchase\_Frq\_Decline

FROM sum\_purchase\_frequency spf

JOIN monetary\_value mv

    ON spf.[CustomerID] = mv.[CustomerID]

WHERE

    mv.Avg\_Monthly\_Spend\_2014 < mv.Avg\_Monthly\_Spend\_2013

OR  spf.[CustomerID] IN (SELECT [CustomerID]

FROM sum\_purchase\_frequency

WHERE Purchase\_Frequency2014 < Purchase\_Frequency2013)

ORDER BY

    spf.[CustomerID] DESC

1. **How can we reach out to these customers with personalized offers or interventions?**

This SQL query retrieves customer information including their ID, name, email, address, state/province, country/region, and postal code from the AdventureWorks database. By combining data from various tables to provide comprehensive customer information. It joins the**[Sales].[Customer], [Person].[Person], [Person].[EmailAddress], [Person].[BusinessEntityAddress], [Person].[Address], [Person].[StateProvince], [Sales].[SalesTerritory]** tables to retrieve the desired columns.

SELECT

    c.[CustomerID],

    e.[BusinessEntityID],

    p.[FirstName],

    p.[MiddleName],

    p.[LastName],

    e.[EmailAddress],

    a.[AddressLine1],

    a.[AddressLine2],

    a.[City],

    s.[Name] state\_province,

    st.[Name] country\_region,

    a.[PostalCode]

FROM

    [Sales].[Customer] c

INNER JOIN

    [Person].[Person] p

ON c.[PersonID] = p.BusinessEntityID

INNER JOIN

    [Person].[EmailAddress] e

ON p.BusinessEntityID = e.BusinessEntityID

INNER JOIN

    [Person].[BusinessEntityAddress] b

ON e.BusinessEntityID = b.[BusinessEntityID]

INNER JOIN

    [Person].[Address] a

ON b.[AddressID] = a.[AddressID]

INNER JOIN

    [Person].[StateProvince] s

ON a.[StateProvinceID] = s.[StateProvinceID]

INNER JOIN

    [Sales].[SalesTerritory] st

ON s.[TerritoryID] = st.[TerritoryID]

**Cross-Selling and Upselling:**

**Problem:** How can we identify opportunities for cross-selling or upselling to our existing customer base?

**Questions:** Which products are frequently purchased together? Can we recommend complementary

products to customers based on their previous purchases? Which customers have made high-value

purchases recently, indicating potential interest in premium upgrades?

**Solution : Cross-Selling and Upselling**

1. **Which products are frequently purchased together?**

The query identifies pairs of products (identified by **ProductID**) that are frequently bought together from the **[Sales].[SalesOrderDetail]** table. It counts the occurrences of each pair, excluding cases where a product is paired with itself, and presents this information as **original\_purchase** (the first product), **bought\_with** (the product it was bought with), and **number\_of\_times** (how many times they were bought together). The results are sorted in descending order of frequency.

   SELECT SOD1.[ProductID] AS product\_1, SOD2.[ProductID] AS product\_2, COUNT(\*) AS times\_bought\_together

   FROM [Sales].[SalesOrderDetail] SOD1

   JOIN [Sales].[SalesOrderDetail] SOD2

   ON SOD1.[SalesOrderID] = SOD2.[SalesOrderID]

WHERE SOD1.[ProductID] > SOD2.[ProductID]

   GROUP BY  SOD1.[ProductID], SOD2.[ProductID]

   ORDER BY times\_bought\_together DESC

1. **Can we recommend complementary products to customers based on their previous purchases?**
2. **Which customers have made high-value purchases recently, indicating potential interest in premium upgrades?**

This SQL code identifies high-value customers based on their total spending in the last 3 months. It calculates the total spent by each customer in this period, categorizing them as either 'High-value customer' if they spent more than $100,000 or 'Regular customer' if they spent less. The result is ordered by the total spent in descending order.

WITH CustomerHighValuePurchases AS (

    SELECT

        SOH.[CustomerID],

        SUM(sod.[OrderQty] \* sod.[UnitPrice]) AS total\_spent

    FROM

        [Sales].[SalesOrderHeader] SOH

    INNER JOIN

        [Sales].[SalesOrderDetail] SOD ON SOH.[SalesOrderID] = SOD.[SalesOrderID]

    WHERE

        SOH.[OrderDate] >= DATEADD(month, -3, 2014/07/01) -- Consider purchases from the last 3 months

    GROUP BY

        SOH.[CustomerID]

)

SELECT

    chv.[CustomerID],

    chv.total\_spent,

    CASE

        WHEN chv.total\_spent > 100000 THEN 'High-value customer'

        ELSE 'Regular customer'

    END AS customer\_category

FROM

    CustomerHighValuePurchases chv

ORDER BY

    chv.total\_spent DESC;

OR

WITH CustomerHighValuePurchases AS (

    SELECT

        SOH.[CustomerID],

        SUM(sod.[OrderQty] \* sod.[UnitPrice]) AS total\_spent

    FROM

        [Sales].[SalesOrderHeader] SOH

    INNER JOIN

        [Sales].[SalesOrderDetail] SOD ON SOH.[SalesOrderID] = SOD.[SalesOrderID]

    WHERE

        SOH.[OrderDate] >= DATEADD(month, -3, 2014/07/01) -- Consider purchases from the last 3 months

    GROUP BY

        SOH.[CustomerID] )

SELECT

    chv.[CustomerID],

    chv.total\_spent

FROM

    CustomerHighValuePurchases chv

    WHERE total\_spent >=100000

ORDER BY chv.total\_spent DESC;

**Marketing Campaign Optimization:**

**Problem:** How can we optimize our marketing campaigns to target the right customers at the right

time?

**Questions:** Which customer segments should we prioritize for specific marketing campaigns? What is

the optimal timing for sending promotional offers based on the recency of customers&#39; last purchases?

Which channels are most effective in reaching different customer segments?

**Solution: Marketing Campaign Optimization**