* INSPECT DATA
* check imported dataset
* SELECT \*
* FROM SalesDataSample..sales\_data\_sample
* total count of unique orders
* SELECT COUNT(DISTINCT ORDERNUMBER)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 307 UNIQUE ORDERS
* total count of orders
* SELECT COUNT(ORDERNUMBER)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 2823 ORDERS
* total sum of quantity order
* SELECT SUM(QUANTITYORDERED)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 99,067 QUANTITIES ORDERED
* NOTE 1: SALES = PRICEEACH \* QUANTITYORDERED
* NOTE 2: PRICEEACH IS NOT A CONSTANT VARIABLE WRT ORDERLINENUMBER
* total count of order line
* SELECT COUNT(DISTINCT ORDERLINENUMBER)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 18 ORDER LINE
* total sum of sales
* SELECT SUM(SALES)
* FROM SalesDataSample..sales\_data\_sample
* THE TOTAL SUM OF SALES IS 10,032,628.85
* total count of unique status
* SELECT COUNT(DISTINCT STATUS)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 6 UNIQUE STATUS
* date range
* SELECT DISTINCT YEAR\_ID
* FROM SalesDataSample..sales\_data\_sample
* ORDER BY YEAR\_ID
* THIS DATASET RANGES BETWEEN 2003 TO 2005
* check if each year has a complete month
* SELECT COUNT(DISTINCT MONTH\_ID)
* FROM SalesDataSample..sales\_data\_sample
* WHERE YEAR\_ID = 2003
* 2003 HAS 12 COMPLETE MONTHS
* SELECT COUNT(DISTINCT MONTH\_ID)
* FROM SalesDataSample..sales\_data\_sample
* WHERE YEAR\_ID = 2004
* 2004 HAS 12 CALENDER MONTHS
* SELECT COUNT(DISTINCT MONTH\_ID)
* FROM SalesDataSample..sales\_data\_sample
* WHERE YEAR\_ID = 2005
* 2005 HAS 5 CALENDER MONTHS
* SELECT DISTINCT MONTH\_ID
* FROM SalesDataSample..sales\_data\_sample
* WHERE YEAR\_ID = 2005
* ORDER BY MONTH\_ID
* VIZ 1, 2, 3, 4, 5
* NOTE 3: 2005 IS AN INCOMPLETE YEAR
* total count of distinct product line
* SELECT COUNT(DISTINCT PRODUCTLINE)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 7 PRODUCT LINE
* total count of distinct product code
* SELECT COUNT(DISTINCT PRODUCTCODE)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 109 PRODUCT CODE
* total count of distinct city of customers
* SELECT COUNT(DISTINCT CITY)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 73 CITIES
* total count of distinct country of customer
* SELECT COUNT(DISTINCT COUNTRY)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 19 COUNTRIES
* total count of distinct deal size
* SELECT COUNT(DISTINCT DEALSIZE)
* FROM SalesDataSample..sales\_data\_sample
* THERE ARE 3 DEALSIZE
* ANALYSIS
* order line by sum of sales and count of order
* SELECT ORDERLINENUMBER AS orderLineNumber, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY orderLineNumber
* ORDER BY totalSales DESC
* ORDER LINE NUMBER 1 HAS THE MAXIMUM TOTAL SALES OF 1,119,219.21 AND TOTAL ORDER OF 307 UNITS
* ORDER LINE NUMBER 18 HAS THE MINIMUM TOTAL SALES OF 24,155.62 AND TOTAL ORDER OF 10 UNITS
* status by sum of sale and count of order
* SELECT STATUS AS [status], SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY STATUS
* ORDER BY totalSales DESC
* SHIPPED HAS THE MAXIMUM TOTAL SALES OF 9291501.079 AND TOTAL ORDER OF 2617 UNITS
* DISPUTED HAS THE MINIMUM TOTAL SALES OF 72212.86 AND TOTAL ORDER OF 14 UNITS
* year by sum of sale and count of order
* SELECT YEAR\_ID AS [year], SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY YEAR\_ID
* ORDER BY totalSales ASC
* THERE WAS AN INCREASE IN SALE AND TOTAL ORDER BETWEEN 2003 AND 2004. HOWEVER DUE TO INCOMPLETE DATA MONTH FOR YEAR 205,
* IT WOULD BE DIFFICULT TO SAY IF THERE WOULD BE A DECREASE OR INCREASE IN SALES.
* use mean to predict if sales and total order would increase in year 2005
* SELECT YEAR\_ID AS [year], AVG(SALES) AS meanSales, AVG(ORDERNUMBER) AS meanOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY YEAR\_ID
* ORDER BY meanSales ASC
* THERE ARE CHANCES THAT THERE WOULD ALSO BE AN INCREASE IN SALES AND TOTAL ORDER BETWEEN THE YEAR 2004 AND 2005
* months by sum of sale and count of order
* NOTE: IN ORDER TO AVIOD BIAS IN DATA WE WOULD EXCLUDE YEAR 2005
* SELECT MONTH\_ID AS MONTH, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* WHERE YEAR\_ID != 2005
* GROUP BY MONTH\_ID
* ORDER BY totalSales DESC
* MONTH 11 HAS THE MAXIMUM TOTAL SALES OF 2118885.67 AND TOTAL ORDER OF 597 UNITS
* MONTH 3 HAS THE MINIMUM TOTAL SALES OF 380238.63 AND TOTAL ORDER OF 106 UNITS
* FURTHER ANALYSIS ON THE PRODUCT LINE SOLD IN THE 11 MONTH
* SELECT MONTH\_ID AS MONTH, PRODUCTLINE AS productLine, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* WHERE YEAR\_ID != 2005 AND MONTH\_ID = 11
* GROUP BY MONTH\_ID, PRODUCTLINE
* ORDER BY totalSales DESC
* CLASSIC CARS HAS THE MAXIMUM TOTAL SALES OF 825156.26 AND TOTAL ORDER OF 219 UNITS
* TRAINS HAS THE MINIMUM TOTAL SALES OF 44794.63 AND TOTAL ORDER OF 15 UNITS
* product line by sum of sale and count of order
* SELECT PRODUCTLINE AS productLine, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY PRODUCTLINE
* ORDER BY totalSales DESC
* CLASSIC CARS HAS THE MAXIMUM TOTAL SALES OF 3919615.66 AND TOTAL ORDER OF 967 UNITS
* TRAINS HAS THE MINIMUM TOTAL SALES OF 226243.47 AND TOTAL ORDER OF 77 UNITS
* top 10 city by sum of sale and count of order
* SELECT TOP 10 CITY AS city, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY CITY
* ORDER BY totalSales DESC
* MADRID HAS THE MAXIMUM TOTAL SALES OF 1082551.44 AND TOTAL ORDER OF 304 UNITS
* BRICKHAVEN HAS THE MINIMUM TOTAL SALES OF 165255.2 AND TOTAL ORDER OF 47 UNITS
* country by sum of sale and count of order
* SELECT COUNTRY AS country, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY COUNTRY
* ORDER BY totalSales DESC
* USA HAS THE MAXIMUM TOTAL SALES OF 3627982.83 AND TOTAL ORDER OF 1004 UNITS
* IRELAND HAS THE MINIMUM TOTAL SALES OF 57756.43 AND TOTAL ORDER OF 16 UNITS
* dealsize by sum of sale and count of order
* SELECT DEALSIZE AS dealSize, SUM(SALES) AS totalSales, COUNT(ORDERNUMBER) AS totalOrder
* FROM SalesDataSample..sales\_data\_sample
* GROUP BY DEALSIZE
* ORDER BY totalSales DESC
* MEDIUM SIZE HAS THE MAXIMUM TOTAL SALES OF 6087432.24 AND TOTAL ORDER OF 1384 UNITS
* LERGE SIZE HAS THE MINIMUM TOTAL SALES OF 1302119.26 AND TOTAL ORDER OF 157 UNITS
* BEST CUSTOMER USING RFM
* The RFM in RFM analysis stands for recency, frequency and monetary value.
* An RFM model is built using three key factors:
* How recently a customer has transacted with a brand
* How frequently they’ve engaged with a brand
* How much money they’ve spent on a brand’s products and services
* DROP TABLE IF EXISTS #rfm;
* WITH
* rfm AS
* (
  + SELECT CUSTOMERNAME AS customerName, SUM(SALES) AS monetaryValue, COUNT(ORDERNUMBER) AS frequency,
    - * MAX(ORDERDATE) AS lastDateOrder, (SELECT MAX(ORDERDATE) FROM SalesDataSample..sales\_data\_sample) AS maxDateOrder,
      * DATEDIFF(DD, MAX(ORDERDATE), (SELECT MAX(ORDERDATE) FROM SalesDataSample..sales\_data\_sample)) AS recency
  + FROM SalesDataSample..sales\_data\_sample
  + GROUP BY CUSTOMERNAME
* ),
* rfm\_calc AS
* (
  + SELECT rfm.\*,
    - * NTILE(4) OVER (ORDER BY recency DESC) AS rfmRecency,
      * NTILE(4) OVER (ORDER BY frequency) AS rfmFrequency,
      * NTILE(4) OVER (ORDER BY monetaryValue) AS rfmMonetaryValue
  + FROM rfm
* )
* SELECT rfm\_calc.\*,
  + - rfmRecency + rfmFrequency + rfmMonetaryValue AS rfmCell,
    - CAST(rfmRecency AS VARCHAR) + CAST(rfmFrequency AS VARCHAR) + CAST(rfmMonetaryValue AS VARCHAR) AS rfmCellString
* INTO #rfm
* FROM rfm\_calc
* SELECT CUSTOMERNAME, rfmRecency, rfmFrequency, rfmMonetaryValue,
  + - CASE
      * WHEN rfmCell >= 10 THEN 'loyal customers'
      * WHEN rfmCell >= 8 THEN 'active customers'
      * WHEN rfmCell >= 6 THEN 'potential customers'
      * WHEN rfmCell >= 5 THEN 'new customers'
      * ELSE 'lost customers'
    - END rfmSegment
* FROM #rfm