```
/* Customer Sales Data Exploration

The purpose of this project: use SQL to explore the sales data and do RFM analysis
   for Customer segmentation
To do that first created a new database called 'ProjectSalesData'
*/

Use ProjectSalesData

Select * From dbo.sample_sales_data

--- count total columns from the table
Select COUNT(*)
From information_schema.columns
Where table_name = 'sample_sales_data';

---Questions: Check for unique values for some of columns /  features
Select distinct YEAR_ID  From sample_sales_data   ---status for the order have 6
  different status
Select distinct STATUS From sample_sales_data   ---the data are of for 3 different
   year, 2003, 2004 and 2005
Select distinct PRODUCTLINE From sample_sales_data  ---there are 7 differetn
  products are available
Select distinct COUNTRY From sample_sales_data   ---there are 19 ountries in the
  dataset
Select distinct TERRITORY From  sample_sales_data ---4 territiries are there
Select distinct DEALSIZE From sample_sales_data   ---there are 3 different deal
  size
Select distinct CUSTOMERNAME From sample_sales_data  ---There are total 92
  customers

---now do some analysis and calculations

--Question: check the total sale by product line
select PRODUCTLINE, ROUND(SUM(SALES),2) As Revenue
From sample_sales_data
Group by PRODUCTLINE
Order by 2 desc;  ---here ordering by Revenue and its column no. is 2, so used 2
  before desc
--- the first 3 revenue producting products are Classic Cars, Vintage cars and
  Motorcycles.

--- Question: check the total sale by each year
select YEAR_ID, ROUND(SUM(SALES),0) As Revenue
From sample_sales_data
Group by YEAR_ID
Order by 2 desc;
--- Highest sale is for 2004 and then sudden decline in the total revenue for the
```

```
  year 2005. So need to look at why that happened.

--- let's see the month wise sales for the year of 2005
Select distinct MONTH_ID From sample_sales_data Where YEAR_ID = 2003; --for 2003,
  the sales is for 12 months
Select distinct MONTH_ID From sample_sales_data Where YEAR_ID = 2004;  --for 2004,
   the sales is for 12 months
Select distinct MONTH_ID From sample_sales_data Where YEAR_ID = 2005;  --for 2005,
   the sales is for first 5 months only.
                                                          ----so it is obvious
                    that sales is very less compared to other years.

--- now check if dealsize is making any impact on the total sales
Select DEALSIZE, ROUND(SUM(SALES),0) As Revenue
From sample_sales_data
Group by DEALSIZE
Order by 2 desc
---Medium size deals have enerated more revenue(total sales). so we cn promote for
   small and large size deals to increase the sales

---now see which month have more generated revenue by specific year
Select MONTH_ID, ROUND(SUM(SALES), 0) As Revenue, COUNT(ORDERNUMBER) As Frequency
From sample_sales_data
Where YEAR_ID =  2003
Group by MONTH_ID
Order by 2 Desc;
--- the highest revenue generating months are November and then October for year
  2003 with the most order placed.

--now lets see what products are purchased most, it might be classic cars, but let
   us verify.
Select MONTH_ID, PRODUCTLINE, ROUND(SUM(SALES), 0) As Revenue, COUNT(ORDERNUMBER)
  As Frequency
From sample_sales_data
Where YEAR_ID = 2003 and MONTH_ID=11
Group by MONTH_ID, PRODUCTLINE
Order by 3 Desc;

-----What city has the highest number of sales in a specific country.
--Change country name to find most revenue generating city
select city, ROUND(SUM (sales),0) Revenue
from sample_sales_data
where country = 'USA'
group by city
order by 2 desc

---What is the best product in United States?
---Change country name accordingly
select country, YEAR_ID, PRODUCTLINE, ROUND(SUM(sales),0) Revenue
```

```
from sample_sales_data
where country = 'USA'
group by  country, YEAR_ID, PRODUCTLINE
order by 4 desc

---Are there more then one product ordered for same ordernumber?
Select ORDERNUMBER, Count(*) rownumber
From sample_sales_data
Where STATUS = 'Shipped'
Group By ORDERNUMBER
Order By rownumber Desc;
---we can say from the above query that some order numbers have more than one row, ↵
   means more than one product ordered for that Ordernumber

---What products are most often sold together?
select distinct ORDERNUMBER, stuff(

    (select ',' + ProductCode
    from sample_sales_data As p
    where ORDERNUMBER in
        (
            select ORDERNUMBER
            from (
                select ORDERNUMBER, count(*) rn
                FROM sample_sales_data
                where STATUS = 'Shipped'
                group by ORDERNUMBER
            )m
            where rn = 3
        )
        and p.ORDERNUMBER = s.ORDERNUMBER
        for xml path (''))

        , 1, 1, '') ProductCodes

from sample_sales_data As s
order by 2 desc


---finding the best customer for RFM analysis.
--Recency - last order date by customers /
--Frequency- Count of total orders by customers /
--Monetary - Total spending by customer /

--from below query, I am finding the last order date placed by the customers.      ↵
  means finding Recency
Select CUSTOMERNAME,
   ROUND(SUM(SALES),2) As TotalMonetaryValue,
   ROUND(AVG(SALES),2) As AvgMonetaryValue,
```

```
   COUNT(ORDERNUMBER) As Frequency,
   MAX(ORDERDATE) As LastOrderDate
From sample_sales_data
Group by CUSTOMERNAME;

---now let's see Who is the best customer.
---Count total spend as Monetary value, Frequency of orders and last order date as
   Recency
Select CUSTOMERNAME,
   ROUND(SUM(SALES),2) As TotalMonetaryValue,
   ROUND(AVG(SALES),2) As AvgMonetaryValue,
   COUNT(ORDERNUMBER) As Frequency,
   MAX(CAST(ORDERDATE As DATE)) As LastOrderDate,   ---use cast to remove
     timestamp from last oorderdate
   (Select MAX(CAST(ORDERDATE As DATE)) From sample_sales_data) As MaxOrderDate,
   DATEDIFF(DD, MAX(CAST(ORDERDATE AS DATE)), (Select MAX(CAST(ORDERDATE As DATE))
      From sample_sales_data)) As Recency
From sample_sales_data
Group by CUSTOMERNAME
Order by Recency Asc;

--- now create a CTE 'rfm' with above query and then call rfm as alias r using CTE
With rfm as
(
    Select CUSTOMERNAME,
       ROUND(SUM(SALES),2) As TotalMonetaryValue,
       ROUND(AVG(SALES),2) As AvgMonetaryValue,
       COUNT(ORDERNUMBER) As Frequency,
       MAX(CAST(ORDERDATE As DATE)) As LastOrderDate,   ---use cast to remove
         timestamp from last oorderdate
       (Select MAX(CAST(ORDERDATE As DATE)) From sample_sales_data) As
         MaxOrderDate,
       DATEDIFF(DD, MAX(CAST(ORDERDATE AS DATE)), (Select MAX(CAST(ORDERDATE As
         DATE)) From sample_sales_data)) As Recency
    From sample_sales_data
    Group by CUSTOMERNAME
)
Select r. *          ---- Select all columns from call CTE rfm as r
From rfm As r;

---now do buckating using NTILE() for above query. The NTILE() function is a
  window function that distributes rows of
--an ordered partition into a specified number of approximately equal -sized
  groups, or "tiles." This can be particularly useful
--for ranking or categorizing data.

With rfm as
(
    Select CUSTOMERNAME,
```

```
        ROUND(SUM(SALES),2) As TotalMonetaryValue,
        ROUND(AVG(SALES),2) As AvgMonetaryValue,
        COUNT(ORDERNUMBER) As Frequency,
        MAX(CAST(ORDERDATE As DATE)) As LastOrderDate,    ---use cast to remove
          timestamp from last oorderdate
        (Select MAX(CAST(ORDERDATE As DATE)) From sample_sales_data) As
          MaxOrderDate,
        DATEDIFF(DD, MAX(CAST(ORDERDATE AS DATE)), (Select MAX(CAST(ORDERDATE As
          DATE)) From sample_sales_data)) As Recency
    From sample_sales_data
    Group by CUSTOMERNAME
)
Select r. * ,
    NTILE(4) OVER (Order By Recency desc) rfm_recency,
    NTILE(4) OVER (Order By Frequency) rfm_frequency,
    NTILE(4) OVER (Order By AvgMonetaryValue) rfm_monetary
From rfm As r;
-- here I have divided data in 4 groups represented by 1, 2, 3 and 4. where 4 is a
   higher value and 1 is lowest value
--- for Recency - 4 means recent transaction and 1 means older transaction



------
Drop Table If Exists #rfm
;With rfm as
(
    Select CUSTOMERNAME,
        ROUND(SUM(SALES),2) As TotalMonetaryValue,
        ROUND(AVG(SALES),2) As AvgMonetaryValue,
        COUNT(ORDERNUMBER) As Frequency,
        MAX(CAST(ORDERDATE As DATE)) As LastOrderDate,    ---use cast to remove
          timestamp from last oorderdate
        (Select MAX(CAST(ORDERDATE As DATE)) From sample_sales_data) As
          MaxOrderDate,
        DATEDIFF(DD, MAX(CAST(ORDERDATE AS DATE)), (Select MAX(CAST(ORDERDATE As
          DATE)) From sample_sales_data)) As Recency
    From sample_sales_data
    Group by CUSTOMERNAME
),
rfm_calc As
(
    Select r. * ,
        NTILE(4) OVER (Order By Recency desc) rfm_recency,
        NTILE(4) OVER (Order By Frequency) rfm_frequency,
        NTILE(4) OVER (Order By TotalMonetaryValue) rfm_monetary
    From rfm r
)
select
```

```
    c.*, rfm_recency+ rfm_frequency+ rfm_monetary as rfm_cell,
    cast(rfm_recency as varchar) + cast(rfm_frequency as varchar) + cast
      (rfm_monetary  as varchar)rfm_cell_string
into #rfm
from rfm_calc c

select CUSTOMERNAME , rfm_recency, rfm_frequency, rfm_monetary,
    case
        when rfm_cell_string in (111, 112 , 121, 122, 123, 132, 211, 212, 114,
          141) then 'Lost customers'  --lost customers
        when rfm_cell_string in (133, 134, 143, 234, 244, 334, 343, 344, 144) then
           'Slipping away, cannot lose' -- (Big spenders who haven't purchased
          lately) slipping away
        when rfm_cell_string in (311, 411, 412, 331) then 'New customers'
        when rfm_cell_string in (221, 222, 223, 232, 233, 322) then 'Potential
          churners'
        when rfm_cell_string in (323, 333,321, 423, 422, 421, 332, 432) then
          'Active' --(Customers who buy often & recently, but at low price points)
        when rfm_cell_string in (433, 434, 443, 444) then 'Loyal'
    end rfm_segment

from  #rfm



---What products are most often sold together?

Select ORDERNUMBER, Count(*) rownumber
From sample_sales_data
Where STATUS = 'Shipped'
Group By ORDERNUMBER
Order By rownumber Desc;
---- we can say from the above query that some order numbers have more than one
  row, means more than one product ordered for that Ordernumber
```