**Customer Information**

SELECT

    DISTINCT SOH.CustomerID,

    FirstName + ' ' + LastName AS CustomerName,

    PA.AddressLine1 + ', ' + PA.City + ', ' + SP.Name + ' ' + PA.PostalCode Address,

    CR.Name Country,

    PhoneNumber,

    EmailAddress

FROM Sales.Customer SC

LEFT JOIN Person.Person PP

ON SC.PersonID = PP.BusinessEntityID

LEFT JOIN Sales.SalesOrderHeader SOH

ON SC.CustomerID = SOH.CustomerID

LEFT JOIN Person.Address PA

ON SOH.BillToAddressID = PA.AddressID

LEFT JOIN Person.StateProvince SP

ON PA.StateProvinceID = SP.StateProvinceID

LEFT JOIN Person.PersonPhone PPP

ON SC.PersonID = PPP.BusinessEntityID

LEFT JOIN Person.EmailAddress EA

ON SC.PersonID = EA.BusinessEntityID

LEFT JOIN Person.CountryRegion CR

ON SP.CountryRegionCode = CR.CountryRegionCode

WHERE SC.PersonID IS NOT NULL;

**Customer Sales**

SELECT

    SC.CustomerID,

    SOD.SalesOrderID,

    SOD.ProductID,

    SOD.OrderQty,

    CAST(SOH.OrderDate AS date) OrderDate,

    SOD.OrderQty \* PP.StandardCost CostPrice,

    PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty AS SalesPrice,

    SOD.UnitPriceDiscount SalesPriceDiscount,

    (PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty) - (SOD.OrderQty \* PP.StandardCost) AS Profit

FROM Sales.Customer SC

INNER JOIN Sales.SalesOrderHeader SOH

ON SC.CustomerID = SOH.CustomerID

INNER JOIN Sales.SalesOrderDetail SOD

ON SOH.SalesOrderID = SOD.SalesOrderID

INNER JOIN Production.Product PP

ON SOD.ProductID = PP.ProductID;

**CLV**

WITH customersalesdetails AS (

    SELECT

        SC.CustomerID,

        SOD.SalesOrderID,

        SOD.ProductID,

        SOD.OrderQty,

        CAST(SOH.OrderDate AS date) OrderDate,

        SOD.OrderQty \* PP.StandardCost CostPrice,

        SOD.UnitPriceDiscount SalesPriceDiscount,

        PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty AS SalesPrice,

        (PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty) - (SOD.OrderQty \* PP.StandardCost) AS Profit

    FROM Sales.Customer SC

    INNER JOIN Sales.SalesOrderHeader SOH

    ON SC.CustomerID = SOH.CustomerID

    INNER JOIN Sales.SalesOrderDetail SOD

    ON SOH.SalesOrderID = SOD.SalesOrderID

    INNER JOIN Production.Product PP

    ON SOD.ProductID = PP.ProductID

)

SELECT

    CustomerID,

    COUNT(SalesOrderID) AS TotalOrders,

    SUM(SalesPrice) AS TotalSpent,

    DATEDIFF (DAY, MIN(OrderDate), MAX(OrderDate)) AS DaysAsCustomer,

    CASE

        WHEN COUNT(SalesOrderID) > 0 AND DATEDIFF(DAY, MIN(OrderDate), MAX(OrderDate)) > 0 THEN

        SUM(SalesPrice) / NULLIF(DATEDIFF(DAY, MIN(OrderDate), MAX(OrderDate)), 0)

    ELSE 0

    END AS AverageDailySpend,

    SUM(SalesPrice) / COUNT(SalesOrderID) AS AverageOrderValue,

    SUM(SalesPrice) \* 365 / NULLIF(DATEDIFF(DAY, MIN(OrderDate), MAX(OrderDate)), 0) AS AnnualCLV

FROM customersalesdetails

GROUP BY CustomerID

**Cohort Table**

SELECT

        SC.CustomerID,

        ST.CountryRegionCode,

        SOH.SalesOrderID,

        SOD.ProductID,

        OrderQty,

        LineTotal,

        CAST(OrderDate AS DATE) DateOrder

INTO #retention1

FROM Sales.Customer SC

LEFT JOIN Sales.SalesOrderHeader SOH

ON SC.CustomerID = SOH.CustomerID

LEFT JOIN Sales.SalesOrderDetail SOD

ON SOH.SalesOrderID = SOD.SalesOrderID

LEFT JOIN Sales.SalesTerritory ST

ON SC.TerritoryID = ST.TerritoryID

WHERE OrderDate IS NOT NULL

AND (OrderDate BETWEEN '2013-06-01' AND '2014-06-30');

SELECT

    CustomerID,

    MIN(DateOrder) First\_Purchase\_Date,

    DATEFROMPARTS(YEAR(MIN(DateOrder)),MONTH(MIN(DateOrder)),1) Cohort\_Date

INTO #cohort

FROM #retention1

GROUP BY CustomerID;

WITH cte1 AS (

    SELECT

        r.\*,

        c.Cohort\_Date,

        YEAR(DateOrder) OrderYear,

        MONTH(DateOrder) OrderMonth,

        YEAR(Cohort\_Date) CohortYear,

        MONTH(Cohort\_Date) CohortMonth

    FROM #retention1 r

    LEFT JOIN #cohort c

    ON r.CustomerID = c.CustomerID

), cte2 AS (

    SELECT

        \*,

        Year\_Diff = OrderYear - CohortYear,

        Month\_Diff = OrderMonth - CohortMonth

    FROM cte1

)

SELECT

    \*,

    Year\_Diff \* 12 + Month\_Diff + 1 AS Cohort\_Index

INTO #cohort\_retention

FROM cte2;

SELECT \*

Into #cohort\_pivot

FROM (

    SELECT

        Distinct CustomerID,

        Cohort\_Date,

        Cohort\_Index

    FROM #cohort\_retention

    )tbl

PIVOT(

    COUNT(CustomerID)

    FOR Cohort\_Index IN ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13])

) AS Pivot\_Table

ORDER BY Cohort\_Date;

SELECT

    Cohort\_Date,

    CAST((1.0\*[1]/[1]) AS Decimal(5,3)) AS [1],

    CAST((1.0\*[2]/[1]) AS Decimal(5,3)) AS [2],

    CAST((1.0\*[3]/[1]) AS Decimal(5,3)) AS [3],

    CAST((1.0\*[4]/[1]) AS Decimal(5,3)) AS [4],

    CAST((1.0\*[5]/[1]) AS Decimal(5,3)) AS [5],

    CAST((1.0\*[6]/[1]) AS Decimal(5,3)) AS [6],

    CAST((1.0\*[7]/[1]) AS Decimal(5,3)) AS [7],

    CAST((1.0\*[8]/[1]) AS Decimal(5,3)) AS [8],

    CAST((1.0\*[9]/[1]) AS Decimal(5,3)) AS [9],

    CAST((1.0\*[10]/[1]) AS Decimal(5,3)) AS [10],

    CAST((1.0\*[11]/[1]) AS Decimal(5,3)) AS [11],

    CAST((1.0\*[12]/[1]) AS Decimal(5,3)) AS [12],

    CAST((1.0\*[13]/[1]) AS Decimal(5,3)) AS [13]

FROM #cohort\_pivot

ORDER BY cohort\_Date

**Customer Behavior**

SELECT

    C.CustomerID,

    CAST(MIN(OH.OrderDate) AS DATE)  AS FirstPurchaseDate,

    CAST(MAX(OH.OrderDate) AS DATE) AS LatestPurchaseDate,

    DATEDIFF(DAY, MIN(OH.OrderDate), MAX(OH.OrderDate)) AS DaysAsCustomer,

    DATEDIFF(MONTH, MAX(OH.OrderDate), '2014-06-30') AS MonthSinceLastPurchase,

    SUM(PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty) TotalSpent,

    CASE

        WHEN COUNT(OH.SalesOrderID) > 1 THEN 'Repeat Customer'

    ELSE 'Lost Customer'

    END AS CustomerType,

    CASE

        WHEN DATEDIFF(MONTH, MAX(OH.OrderDate), '2014-06-30') <= 6 THEN 'YES'

    ELSE 'NO'

    END AS LastSixMonthPurchase

FROM Sales.Customer C

LEFT JOIN Sales.SalesOrderHeader OH

ON C.CustomerID = OH.CustomerID

LEFT JOIN Sales.SalesOrderDetail SOD

ON OH.SalesOrderID = SOD.SalesOrderID

LEFT JOIN Production.Product PP

ON SOD.ProductID = PP.ProductID

WHERE PersonID IS NOT NULL

GROUP BY C.CustomerID

**Monetary Decline 2012/2013**

WITH table1 AS (

    SELECT

        CustomerID Repeat\_Customer

    FROM Sales.SalesOrderDetail a

    LEFT JOIN Sales.SalesOrderHeader b

    ON a.SalesOrderID = b.SalesOrderID

    WHERE DATEPART(YEAR, OrderDate) IN (2012, 2103)

    GROUP BY CustomerID

    HAVING COUNT(DISTINCT a.SalesOrderID) > 1

), table2 AS (

SELECT

    Repeat\_Customer,

    DATEPART(YEAR, SOH.OrderDate) Year,

    SUM(PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty) YearlySpent

FROM table1 T

INNER JOIN Sales.SalesOrderHeader SOH

ON T.Repeat\_Customer = SOH.CustomerID

INNER JOIN Sales.SalesOrderDetail SOD

ON SOH.SalesOrderID = SOD.SalesOrderID

INNER JOIN Production.Product PP

ON SOD.ProductID = PP.ProductID

WHERE DATEPART(YEAR, OrderDate) IN (2012, 2013)

GROUP BY Repeat\_Customer, DATEPART(YEAR, OrderDate)

HAVING COUNT(SOH.SalesOrderID) > 1

), table3 AS (

SELECT

    \*,

    LAG(YearlySpent) OVER (PARTITION BY Repeat\_Customer ORDER BY Year) PreviousYearSpent,

    YearlySpent -  LAG(YearlySpent) OVER (PARTITION BY Repeat\_Customer ORDER BY Year) MonetryValue

FROM table2

)

SELECT

    \*,

    CASE

        WHEN MonetryValue > 0 THEN 'Improve'

    ELSE 'Decline'

    END AS MonetryStatus

FROM table3

WHERE PreviousYearSpent IS NOT NULL

**Popular product among best 50 customers**

SELECT TOP 50

    SC.CustomerID,

    COUNT(OrderQty) TotalOrderQuantity,

    SUM(PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty) TotalOrderAmount

INTO #y

FROM Sales.Customer SC

LEFT JOIN Sales.SalesOrderHeader SOH

ON SC.CustomerID = SOH.CustomerID

LEFT JOIN Sales.SalesOrderDetail SOD

ON SOH.SalesOrderID = SOD.SalesOrderID

LEFT JOIN Production.Product PP

ON SOD.ProductID = PP.ProductID

GROUP BY SC.CustomerID

ORDER BY TotalOrderAmount DESC;

WITH cte AS (

SELECT CustomerID, ProductID, SUM(OrderQty) ProductQuantity

from Sales.SalesOrderDetail sod

LEFT JOIN Sales.SalesOrderHeader soh

ON sod.SalesOrderID = soh.SalesOrderID

GROUP BY CustomerID, ProductID

)

SELECT

    z.ProductID,

    SUM(ProductQuantity) Quantity,

    RANK() OVER (ORDER by SUM(ProductQuantity) DESC) AS Rank

FROM #y y

LEFT JOIN cte z

ON y.CustomerID = z.CustomerID

GROUP BY z.ProductID

**Product bought together**

SELECT

    PP.ProductID,

    PP.Name,

    SOD.SalesOrderID,

    SOD.SalesOrderDetailID,

    SOD.OrderQty

INTO #table1

FROM Production.Product PP

LEFT JOIN Sales.SalesOrderDetail SOD

ON PP.ProductID = SOD.ProductID

WITH cte\_cs1 AS

(

    SELECT

        a.ProductID ProductID\_1,

        a.Name Product\_1,

        b.ProductID ProductID\_2,

        b.Name Product\_2,

        COUNT(1) TotalCount

    FROM #table1 a

    FULL OUTER JOIN #table1 b

    ON a.SalesOrderID = b.SalesOrderID

    AND a.ProductID <> b.ProductID

    AND a.SalesOrderDetailID < b.SalesOrderDetailID

    WHERE a.ProductID IS NOT NULL AND b.ProductID IS NOT NULL

    GROUP BY a.ProductID, a.Name, b.ProductID, b.Name

    order by TotalCount DESC

)

SELECT

    \*,

    RANK() OVER (ORDER BY TotalCount DESC) AS Rank\_Sold\_Together

FROM cte\_cs1

**Product Inventory**

SELECT

    PP.ProductID,

    PP.Name,

    PP.Color,

    PS.Name SubCategory,

    PC.Name Category,

    PP.ListPrice,

    SUM(Quantity) TotalQuantity

FROM Production.Product PP

LEFT JOIN Production.ProductInventory PI

ON PP.ProductID = PI.ProductID

LEFT JOIN Production.ProductSubcategory PS

ON PP.ProductSubcategoryID = PS.ProductSubcategoryID

LEFT JOIN Production.ProductCategory PC

ON PS.ProductCategoryID = PC.ProductCategoryID

GROUP BY PP.ProductID, PP.Name, PS.Name, PC.Name, PP.Color, PP.ListPrice

ORDER BY TotalQuantity DESC;

**Recency, Frequency, Monetary (RFM)**

WITH cte\_rfm AS (

    SELECT

        SC.CustomerID,

        ST.CountryRegionCode,

        DATEDIFF(DAY, MAX(OrderDate), '2014-06-30 00:00:00.000') DaySinceLastOrder,

        COUNT(SOH.SalesOrderID) OrderCount,

        SUM(PP.ListPrice \* (1 - SOD.UnitPriceDiscount) \* SOD.OrderQty) TotalSpent

    FROM Sales.Customer SC

    LEFT JOIN Sales.SalesOrderHeader SOH

    ON SC.CustomerID = SOH.CustomerID

    LEFT JOIN Sales.SalesOrderDetail SOD

    ON SOH.SalesOrderID = SOD.SalesOrderID

    INNER JOIN Production.Product PP

    ON SOD.ProductID = PP.ProductID

    LEFT JOIN Sales.SalesTerritory ST

    ON SC.TerritoryID = ST.TerritoryID

    WHERE PersonID IS NOT NULL

    GROUP BY SC.CustomerID, ST.CountryRegionCode

),

cte\_rfm\_calc AS (

    SELECT \*,

        NTILE(3) OVER (ORDER BY DaySinceLastOrder DESC) Recency,

        CASE

            WHEN OrderCount <= 1 THEN 1

            WHEN OrderCount BETWEEN 2 AND 5 THEN 2

            WHEN OrderCount > 5 THEN 3

        END AS Frequency,

        CASE

            WHEN TotalSpent < 1000 THEN 1

            WHEN TotalSpent BETWEEN 1000 AND 10000 THEN 2

            WHEN TotalSpent > 10000 THEN 3

        END AS Monetery

    FROM cte\_rfm

),

cte\_rfmtable AS (

    SELECT

        \*,

        CONCAT(Recency, Frequency, Monetery) RFM\_Score

    FROM cte\_rfm\_calc

)

SELECT

    \*,

    CASE

        WHEN RFM\_Score IN (311) THEN 'Low Customers'

        WHEN RFM\_Score IN (111, 121, 131, 122, 133, 113, 112, 132, 123) THEN 'Lost Customers'

        WHEN RFM\_Score IN (212, 221, 211, 232, 223, 222, 213, 231, 321) THEN 'Regular Customers'

        WHEN RFM\_Score IN (322, 331, 312, 233, 313) THEN 'Loyal Customers'

        WHEN RFM\_Score IN (333, 332, 323) THEN 'Premium Customers'

    END RFM\_Segment

FROM cte\_rfmtable

**Cross-Selling**

SELECT

    PP.ProductID,

    PP.Name,

    SOD.SalesOrderID,

    SOD.SalesOrderDetailID,

    SOD.OrderQty

INTO #table1

FROM Production.Product PP

LEFT JOIN Sales.SalesOrderDetail SOD

ON PP.ProductID = SOD.ProductID;

WITH cte\_cs1 AS

(

    SELECT

        a.ProductID ProductID\_1,

        a.Name Product\_1,

        b.ProductID ProductID\_2,

        b.Name Product\_2,

        COUNT(1) TotalCount

    FROM #table1 a

    FULL OUTER JOIN #table1 b

    ON a.SalesOrderID = b.SalesOrderID

    AND a.ProductID <> b.ProductID

    AND a.SalesOrderDetailID < b.SalesOrderDetailID

    WHERE a.ProductID IS NOT NULL AND b.ProductID IS NOT NULL

    GROUP BY a.ProductID, a.Name, b.ProductID, b.Name

), cte\_cs2 AS (

    SELECT

        \*,

        RANK() OVER (ORDER BY TotalCount DESC) AS Rank\_Sold\_Together

    FROM cte\_cs1

)

SELECT \*

FROM cte\_cs2

WHERE Rank\_Sold\_Together <= 50