# Step 1: Datapreparation - Dataset and setup for diving into DAX

Data preprocessing for ShipDate

# Step 2: Requirement of Date Table for creating measures

Create a Blank Query and have the following code:

let fnDateTable = (StartDate as date, EndDate as date, FYStartMonth as number) as table =>

let

DayCount = Duration.Days(Duration.From(EndDate - StartDate)),

Source = List.Dates(StartDate,DayCount,#duration(1,0,0,0)),

TableFromList = Table.FromList(Source, Splitter.SplitByNothing()),

ChangedType = Table.TransformColumnTypes(TableFromList,{{"Column1", type date}}),

RenamedColumns = Table.RenameColumns(ChangedType,{{"Column1", "Date"}}),

InsertYear = Table.AddColumn(RenamedColumns, "Year", each Date.Year([Date]),type text),

InsertYearNumber = Table.AddColumn(RenamedColumns, "YearNumber", each Date.Year([Date])),

InsertQuarter = Table.AddColumn(InsertYear, "QuarterOfYear", each Date.QuarterOfYear([Date])),

InsertMonth = Table.AddColumn(InsertQuarter, "MonthOfYear", each Date.Month([Date]), type text),

InsertDay = Table.AddColumn(InsertMonth, "DayOfMonth", each Date.Day([Date])),

InsertDayInt = Table.AddColumn(InsertDay, "DateInt", each [Year] \* 10000 + [MonthOfYear] \* 100 + [DayOfMonth]),

InsertMonthName = Table.AddColumn(InsertDayInt, "MonthName", each Date.ToText([Date], "MMMM"), type text),

InsertCalendarMonth = Table.AddColumn(InsertMonthName, "MonthInCalendar", each (try(Text.Range([MonthName],0,3)) otherwise [MonthName]) & " " & Number.ToText([Year])),

InsertCalendarQtr = Table.AddColumn(InsertCalendarMonth, "QuarterInCalendar", each "Q" & Number.ToText([QuarterOfYear]) & " " & Number.ToText([Year])),

InsertDayWeek = Table.AddColumn(InsertCalendarQtr, "DayInWeek", each Date.DayOfWeek([Date])),

InsertDayName = Table.AddColumn(InsertDayWeek, "DayOfWeekName", each Date.ToText([Date], "dddd"), type text),

InsertWeekEnding = Table.AddColumn(InsertDayName, "WeekEnding", each Date.EndOfWeek([Date]), type date),

InsertWeekNumber= Table.AddColumn(InsertWeekEnding, "Week Number", each Date.WeekOfYear([Date])),

InsertMonthnYear = Table.AddColumn(InsertWeekNumber,"MonthnYear", each [Year] \* 10000 + [MonthOfYear] \* 100),

InsertQuarternYear = Table.AddColumn(InsertMonthnYear,"QuarternYear", each [Year] \* 10000 + [QuarterOfYear] \* 100),

ChangedType1 = Table.TransformColumnTypes(InsertQuarternYear,{{"QuarternYear", Int64.Type},{"Week Number", Int64.Type},{"Year", type text},{"MonthnYear", Int64.Type}, {"DateInt", Int64.Type}, {"DayOfMonth", Int64.Type}, {"MonthOfYear", Int64.Type}, {"QuarterOfYear", Int64.Type}, {"MonthInCalendar", type text}, {"QuarterInCalendar", type text}, {"DayInWeek", Int64.Type}}),

InsertShortYear = Table.AddColumn(ChangedType1, "ShortYear", each Text.End(Text.From([Year]), 2), type text),

AddFY = Table.AddColumn(InsertShortYear, "FY", each "FY"&(if [MonthOfYear]>=FYStartMonth then Text.From(Number.From([ShortYear])+1) else [ShortYear]))

in

AddFY

in

fnDateTable

# Step 3: Connect “Dates.Date” to “Orders.OrderDate” on the Model view.

# Step 4: Create a custom table, “MeasuresTable” in PowerBI Model

Step 5: Create DAX query for “TotalSales”

TotalSales = SUMX(Orders, Orders[Amount]\*Orders[SellingPrice])

Step 5: Create DAX query for “TotalCosts”

TotalCosts = SUMX(Orders, Orders[PurchasingPrice]\*Orders[Amount])

Step 5: Create DAX query for “TotalProfit”

TotalProfit = ([TotalSales]-[TotalCosts])

Step 6: Time intelligence – CummulativeSales

TotalSalesMTD = TOTALMTD([TotalSales],Dates[Date])

TotalSalesQTD = TOTALQTD([TotalSales],Dates[Date])

TotalSalesYTD = TOTALYTD([TotalSales],Dates[Date])

Step 6: Get Past Sales

PreviousDaySales = CALCULATE([TotalSales],PREVIOUSDAY(Dates[Date]))

PreviousMonthSales = CALCULATE([TotalSales],PREVIOUSMONTH(Dates[Date]))

PreviousQuarterSales = CALCULATE([TotalSales],PREVIOUSQUARTER(Dates[Date]))

PreviousYearSales = CALCULATE([TotalSales],PREVIOUSYEAR(Dates[Date]))

# Step 7: Compare timeframe

SalesDatesInPeriod = CALCULATE([TotalSales],DATESINPERIOD(Dates[Date],DATE(2015,10,1),2,DAY))

%ChangeDatesInPeriod = CALCULATE([TotalSales], DATESINPERIOD(Dates[Date],DATE(2016,10,1),2,DAY))

/

CALCULATE([TotalSales], DATESINPERIOD(Dates[Date],DATE(2015,10,1),2,DAY))

# Step 8: How much did we sell on the first day last day and in between?

FirstDate = FIRSTDATE(Dates[Date])

Salesfirstday = CALCULATE([TotalSales], FIRSTDATE(Orders[OrderDate]))

Saleslastday = CALCULATE([TotalSales], LASTDATE(Orders[OrderDate]))

Salesbetweenfirstandlastdayofmonth = CALCULATE([TotalSales],DATESBETWEEN(Dates[Date], FIRSTDATE(Orders[OrderDate])+1,LASTDATE(Orders[OrderDate])-1))

# Step 9: Moving average

MovingAverageofSales = AVERAGEX(DATESINPERIOD(Dates[Date],LASTDATE(Dates[Date]),-20,DAY),[TotalSales])

# Step 10: Selected country comparison

SalesselectedCountry = CALCULATE([TotalSales],ALLSELECTED(Suppliers[Country]))

NotselectedCountries = CALCULATE([TotalSales],ALL(Suppliers[Country])) - CALCULATE([TotalSales],ALLSELECTED(Suppliers[Country]))

# Step 11: Best selling day on average using DAX (not the easy way)

AverageSales = AVERAGEX(Dates,[TotalSales])

BestsellingDayName = MAXX(TOPN(1,SUMMARIZE(Dates,Dates[DayOfWeekName],"WeekDay", [AverageSales]),[Weekday],Desc),Dates[DayOfWeekName])

# Step 12: Top employees based on sales

Top3EmployeesContribution = SUMX(TOPN(3,SUMMARIZE(Employees,Employees[Employee Name],"Salesperson",[TotalSales]),[Salesperson],Desc),[TotalSales])

# Step 13: Lost customers

LostCustomers = var Customers = All(Orders[CustomerCompany])

var Timeframe = 90

return COUNTROWS

(

FILTER

(

Customers,

CALCULATE

(

COUNTROWS(Orders),

FILTER

(

ALLSELECTED(Dates[Date]),

Dates[Date]>MIN(Dates[Date])

- Timeframe

&&

Dates[Date] < MIN(Dates[Date])

)

)=0

)

)

Step 14: Monthly Customers - *Can also be done through Drag-Drop*

MonthlyCustomers =

var Customerlist = VALUES(Orders[CustomerCompany])

return

CALCULATE

(

COUNTROWS(Customerlist),

FILTER

(

Customerlist,

CALCULATE

(

COUNTROWS(Orders),

FILTER

(

ALLSELECTED(Dates[Date]),

Dates[Date] < MIN(Dates[Date])

)

) = 0

)

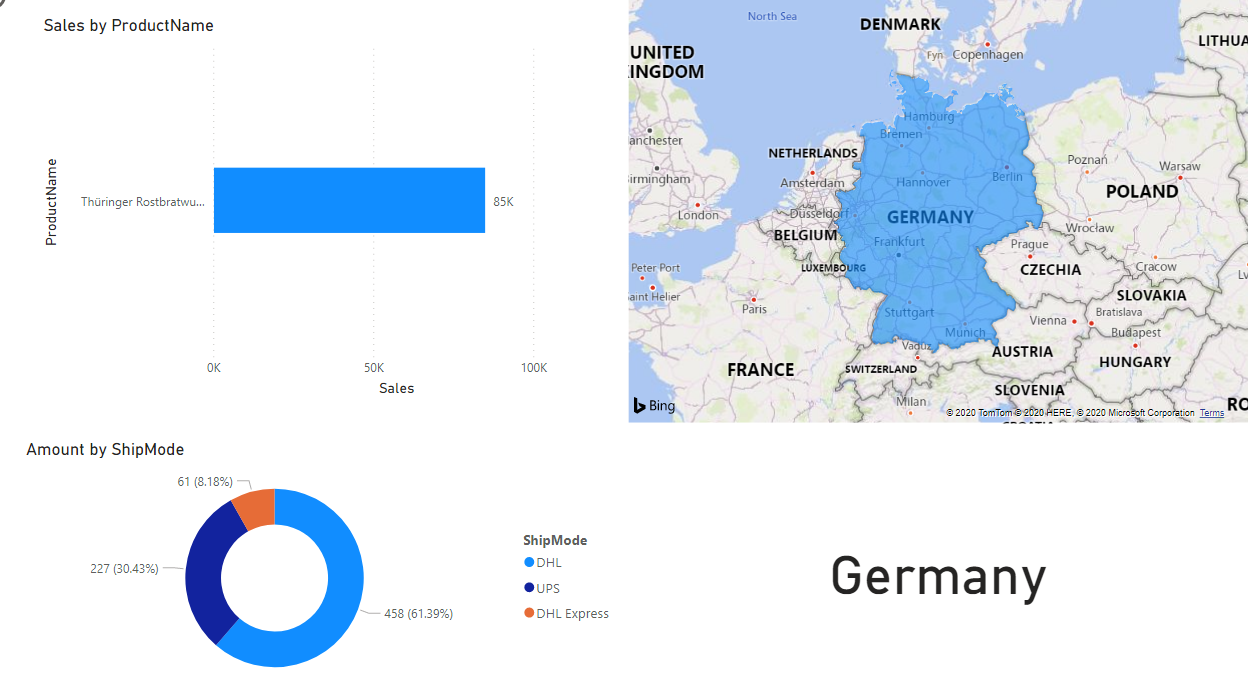
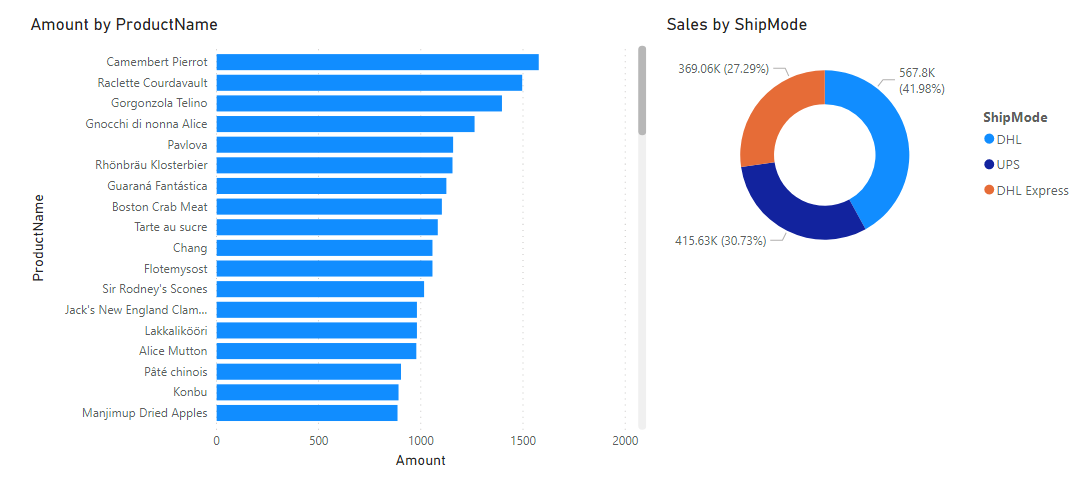
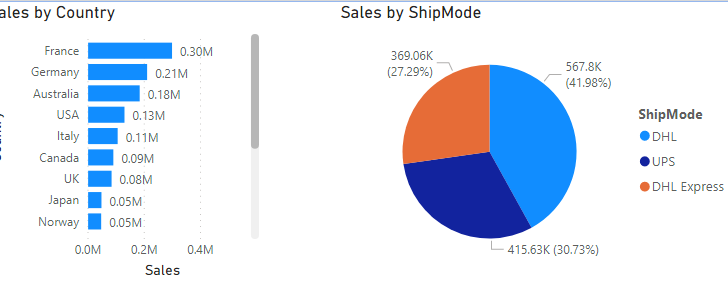
)

Step 15: Different views on data

Step 16: Drillthrough Filters

Drillthrough on Tab page.

Name of country: SelectedCountry = SELECTEDVALUE(Suppliers[Country])



Step 17: Specific date to date comparison

LastYearSalesComparision =

var LastDay = LASTNONBLANK(Dates[Date],CALCULATE([TotalSales],ALLEXCEPT(Orders,Orders[OrderDate])))

var currentrange = DATESBETWEEN(Dates[Date],MIN(Dates[Date]),LastDay)

var previousrange = SAMEPERIODLASTYEAR(currentrange)

return If(LastDay >= MIN(Dates[Date]), CALCULATE([TotalSales], previousrange))

Step 18: Visualize the data

