**PART 3: Adding DAX Measures**

**1)** In the **DATA** view, add the following **calculated columns**:

1. In the **Calendar** table, add a column named "***Weekend***"

Equals "***Y***" for Saturdays or Sundays (otherwise "***N***")

Weekend = **IF** ('Calendar'[Day Name] = "Saturday" || 'Calendar'[Day Name] = "Sunday", "Y", "N")

1. In the **Calendar** table, add a column named "***End of Month***"

Returns the last date of the current month for each row

End of Month = **EOMONTH**('Calendar'[date],0)

1. In the **Customers** table, add a column named "***Current Age***"

Calculates current customer ages using the "*birthdate*" column and the TODAY () function

Current\_Age = **DATEDIFF**(Customers[birthdate],TODAY(),YEAR)

1. In the **Customers** table, add a column named "***Priority***"

Equals "***High***" for customers who own homes and have Golden membership cards (otherwise "***Standard***")

Priority = IF(Customers[member\_card] = "Golden" && Customers[homeowner] = "Y","High","Standard")

1. In the **Customers** table, add a column named "***Short\_Country***"

Returns the first three characters of the customer country, and converts to all uppercase

Short\_Country = **UPPER(LEFT**(Customers[customer\_country],3))

1. In the **Customers** table, add a column named "***House Number***"

Extracts all characters/numbers before the first space in the "*customer\_address*" column (***hint:****use SEARCH*)

House\_Number = **LEFT**(Customers[customer\_address], **SEARCH**(" ", Customers[customer\_address])-1)

1. In the **Products** table, add a column named "***Price\_Tier***"

Equals "***High***" if the retail price is >**$3**, "***Mid***" if the retail price is >**$1**, and "***Low***" otherwise

Price\_Tier = **IF**(Products[product\_retail\_price] >3, "High", **IF**(Products [product\_retail\_price] >1, "Mid","Low"))

1. In the **Stores** table, add a column named "***Years\_Since\_Remodel***"

Calculates the number of years between the current date (TODAY()) and the last remodel date

Years\_Since\_Remodel = **DATEDIFF**(Stores[last\_remodel\_date],Today(),YEAR)

**B)** In the **REPORT** view,

Add the following **measures**(*Assign to tables as you see fit, and use a matrix to match the "****spot check****" values*)

1. Create new measures named "**Quantity Sold**" and "**Quantity Returned**" to calculate the sum of quantity from each data table

***Spot check:****You should see total Quantity Sold =****833,489****and total Quantity Returned =****8,289***

Quantity Sold = Sum (Transaction\_Data[quantity])

Quantity Returned = SUM(Return\_Data[quantity])

1. Create new measures named "**Total Transactions**" and "**Total Returns**" to calculate the count of rows from each data table

***Spot check:****You should see****269,720****transactions and****7,087****returns*

Total Transactions = COUNT(Transaction\_Data[quantity])

Total Returns = COUNT(Return\_Data[quantity])

1. Create a new measure named "**Return Rate**" to calculate the ratio of quantity returned to quantity sold (format as %)

***Spot check:****You should see an overall return rate of****0.99%***

Return Rate = [Quantity Returned] / [Quantity Sold]

1. Create a new measure named "**Weekend Transactions**" to calculate transactions on weekends

***Spot check:****You should see****76,608****total weekend transactions*

Weekend Transactions = CALCULATE([Total Transactions], 'Calendar'[Weekend] = "Y")

1. Create a new measure named "**% Weekend Transactions**" to calculate weekend transactions as a percentage of total transactions (format as %)

***Spot check:****You should see****28.4%****weekend transactions*

% Weekend Transaction = [Weekend Transactions] / [Total Transactions]

1. Create new measures named "**All Transactions**" and "**All Returns**" to calculate grand total transactions and returns (regardless of filter context)

***Spot check:****You should see****269,720****transactions and****7,087****returns across all rows (test with product\_brand on rows)*

All Transactions = CALCULATE([Total Transactions], ALL(Transaction\_Data))

All Returns = CALCULATE([Total Returns],ALL(Return\_Data))

1. Create a new measure to calculate "**Total Revenue**" based on transaction quantity and product retail price, and format as $ (***hint:****you'll need an iterator*)

***Spot check:****You should see a total revenue of****$1,764,546***

Total Revenue = **SUMX**(Transaction\_Data,Transaction\_Data[quantity] \* **RELATED**(Products[product\_retail\_price]))

1. Create a new measure to calculate "**Total Cost**" based on transaction quantity and product cost, and format as $ (***hint:****you'll need an iterator*)

***Spot check:****You should see a total cost of****$711,728***

Total Cost = **SUMX**(Transaction\_Data,(Transaction\_Data[quantity] \* **RELATED**(Products[product\_cost])))

1. Create a new measure named "**Total Profit**" to calculate total revenue minus total cost, and format as $

***Spot check:****You should see a total profit of****$1,052,819***

Total Profit = [Total Revenue] - [Total Cost]

1. Create a new measure to calculate "**Profit Margin**" by dividing total profit by total revenue calculate total revenue (format as %)

***Spot check:****You should see an overall profit margin of****59.67%***

Profit Margin = [Total Profit] / [Total Revenue]

1. Create a new measure named "**Unique Products**" to calculate the number of unique product names in the **Products** table

***Spot check:****You should see****1,560****unique products*

Unique Products = **DISTINCTCOUNT**(Products[product\_name])

1. Create a new measure named "**YTD Revenue**" to calculate year-to-date total revenue, and format as $

***Spot check:****Create a matrix with "****Start of Month****" on rows; you should see****$872,924****in YTD Revenue in September 1998*

YTD Revenue = CALCULATE([Total Revenue], **DATESYTD**('Calendar'[date]))

1. Create a new measure named "**60-Day Revenue**" to calculate a running revenue total over a 60-day period, and format as $

***Spot check:****Create a matrix with "****date****" on rows; you should see****$97,570****in 60-Day Revenue on 4/14/1997*

60-Day Revenue = CALCULATE([Total Revenue], **DATESINPERIOD**('Calendar'[date], MAX('Calendar'[date]),-60,DAY))

1. Create new measures named  "**Last Month Transactions**", "**Last Month Revenue**", "**Last Month Profit**", and "**Last Month Returns**"

***Spot check:****Create a matrix with "****Start of Month****" on rows to confirm accuracy*

Last Month Profit = CALCULATE([Total Profit], DATEADD('Calendar'[date], -1,MONTH))

Last Month Transaction = CALCULATE([Total Transactions], DATEADD('Calendar'[date],-1,MONTH))

Last Month Revenue = CALCULATE([Total Revenue], DATEADD('Calendar'[date], -1,MONTH))

Last Month Return = CALCULATE([Total Returns], DATEADD('Calendar'[date], -1,MONTH))

1. Create a new measure named "**Revenue Target**" based on a 5% lift over the previous month revenue, and format as $

***Spot check:****You should see a Revenue Target of****$99,223****in March 1998*

Revenue Target = [Last Month Revenue] \* 1.05