**ATTIQ Retails Case Study**

**Tables used :**

1. Customers
2. Products
3. Stores
4. Regions
5. Calendar
6. Returns
7. Transactions including a folder combination of two tables for 1997 and 1998.

**Data Cleaning Process**

1. Customers Table : Checked data types of columns. Merged The first name and last name columns to get full name.Extracted Year to get the Year from birthdate. Added a conditional column to get if a person has children or not.
2. Products Table : Checked data types of columns. Applied statistic(distinct) function to get distinct product brands. Added a calculated column for discounted price as :

discounted\_price = Products[product\_retail\_price] \* 0.9. Replaced "*null*" values with zeros in both the "*recyclable*" and "*low-fat*" columns.

1. Stores Table : Checked data type of columns, Add a calculated column named "*full\_address*", by merging "*store\_city*", "*store\_state*", and "*store\_country*", separated by a comma and space.Add a calculated column named "*area\_code*", by extracting the characters before the dash ("-") in the "*store\_phone*" field.
2. Regions Table : Headers promoted. Checked Data Types.
3. Calendars : Headers promoted. Added custom columns, Start of week, name of day, name of month, start of month, Quarter of year, year columns.
4. Returns Table : Headers promoted. Checked Data Types.
5. Transactions : Added two tables via a folder value, Connect to the folder path, and choose "Edit" vs. Combine and Edit.

**Data Modelling Process**

1. Transaction Data to customers via customer\_id(PK in customers table & FK in transaction data)
2. Transaction Data to Products via product\_id.
3. Transaction Data to stores via store\_id.
4. Transaction Data with Date via Date(PK in calendar) with transaction\_date in TData.
5. Transaction Data with Date via inactive connection with
6. Return Data Fact table with Products via product id.
7. Return Data Fact table with Calendar via Date/ Return\_date.
8. Return data table with stores via store\_id.
9. Stores to regions as a snowflake schema via region\_id.

In the **Data view** adjusted the data types in the ribbon, checked out currency and values.

**DAX Calculated columns in Data View**

Data View added columns

1.1 In the **Calendar** table, added a column named "***Weekend***"

* + Equals "***Y***" for Saturdays or Sundays (otherwise "***N***")
  + Weekend = IF(WEEKDAY(Calendar[Date]) = 1 || WEEKDAY(Calendar[Date]) = 7, "Y", "N")

1.2. In the **Calendar** table, added a column named "***End of Month***"

* + Returns the last date of the current month for each row.
  + End of Month = EOMONTH('Calendar'[Date],0)

1.3 In the **Customers** table, added a column named "***Current Age***"

* + Calculates current customer ages using the "*birthdate*" column and the TODAY() function
  + Current Age = DATEDIFF(Customers[birthdate],TODAY(),YEAR)

1.4 In the **Customers** table, added a column named "***Priority***"

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

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

1.5 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))

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

Extracts all characters/numbers before the first space in the "*customer\_address*" column

House\_number =

VAR FirstspacePos = SEARCH(" ", Customers[customer\_address], 1, 0)

RETURN

IF(FirstspacePos > 0,LEFT(Customers[customer\_address], FirstspacePos - 1), Customers[customer\_address])

1.7 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"))

1.8 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

Year\_Since\_remodel = DATEDIFF(Stores[last\_remodel\_date],TODAY(),YEAR)

**Report view Measures**

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

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

Quantity Returned = SUM(Returns[quantity])

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

Total Transactions = COUNTROWS(Transaction\_Data)

Total Returns = COUNTROWS(Returns)

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

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

1. Measure named "**Weekend Transactions**" to calculate transactions on weekends

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

1. Measure named "**% Weekend Transactions**" to calculate weekend transactions as a percentage of total transactions(formatted as a % value)

% Weekend Transaction = DIVIDE([Weekend Transactions],[Total Transactions])

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

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

All Returns = CALCULATE(Returns[Total Returns],ALL(Returns))

1. Measure to calculate "**Total Revenue**" based on transaction quantity and product retail price, and format.

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

1. Measure to calculate "**Total Cost**" based on transaction quantity and product cost, and format

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

1. New measure named "**Total Profit**" to calculate total revenue minus total cost, and format

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

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

Profit Margin = DIVIDE([Total Profit],[Total Revenue])

1. Measure named "**Unique Products**" to calculate the number of unique product names in the **Products** table.

Unique Products = DISTINCTCOUNT(Products[product\_id])

1. measure named "**YTD Revenue**" to calculate year-to-date total revenue, and format.

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

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

60-Day Rolling revenue =

CALCULATE(

[Total Revenue],

DATESINPERIOD(

'Calendar'[Date],

MAX(

'Calendar'[Date]),

-60, DAY

)

)

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

Last Month Transactions =

CALCULATE([Total Transactions],

DATEADD('Calendar'[Date],

-1,

MONTH)

)

Last month revenue =

CALCULATE([Total Revenue],

DATEADD('Calendar'[Date],

-1,

MONTH)

)

Last Month Profit =

CALCULATE([Total Profit],

DATEADD('Calendar'[Date],

-1,

MONTH)

)

Last month Returns =

CALCULATE([Total Returns],

DATEADD('Calendar'[Date],

-1,

MONTH)

)

1. Measure named "**Revenue Target**" based on a 5% lift over the previous month revenue, and format .

Revenue Target = [Last month revenue] \* 1.05.

**Visual Tools and Techniques**

1. Rename the tab "**Topline Performance**" and insert the shop logo.

**2)** Insert a **Matrix** visual to show **Total Transactions**, **Total Profit**, **Profit Margin**, and **Return Rate** by **Product\_Brand** (*on rows*)

* Add conditional formatting to show **data bars** on the Total Transactions column, and **colour scales** on Profit Margin (*White to Green*) and Return Rate (*White to Red*)
* Add a visual level **Top N** filter to only show the top 30 product brands, then sort descending by Total Transactions

**3)** Add a **KPI Card** to show **Total Transactions**, with **Start of Month** as the trend axis and **Last Month Transactions** as the target goal

* Update the title to "***Current Month Transactions***", and format as you see fit
* Create two more copies: one for **Total Profit** (*vs. Last month Profit*) and one for **Total Returns** (*vs. Last Month Returns*)
  + Make sure to update titles, and change the Returns chart to colour coding to "*Low is Good*"

**4)** Add a **Map** visual to show **Total Transactions** by store city

* Add a slicer for store country
  + Under the "selection controls" menu in the formatting pane, activate the "***Show Select All***" option
  + **Pro Tip:** Change the orientation in the "General" formatting menu to **horizontal** and resize to create a *vertical* stack (rather than a list)

**5)** Next to the map, add a **Treemap** visual to break down **Total Transactions** by store country

* Pull in **store\_state** and **store\_city** beneath **store\_country** in the "Group" field to enable drill-up and drill-down functionality

**6)** Beneath the map, add a **Column Chart** to show **Total Revenue** by week, and format as you see fit

* Add a **report level filter** to only show data for 1998
* Update the title to "***Weekly Revenue Trending***"

**7)** In the lower right, add a **Gauge Chart** to show **Total Revenue** against **Revenue Target** (*as either "target value" or "maximum value"*)

* Add a visual level **Top N** filter to show the latest **Start of Month**
* Remove data labels, and update the title to "***Revenue vs. Target***"

**8)** Select the Matrix and activate the **Edit interactions** option to prevent the Treemap from filtering

**9)** Select "*USA"* in the country slicer, and drill down to select "*Portland*" in the Treemap

* Add a new bookmark named "***Portland 1000 Sales***"
* Add a new report page, named "***Notes***"
* Insert a text box and write something along the lines of "***Portland hits 1,000 sales in December***"
* Add a button (your choice) and use the "***Action***" properties to link it to the bookmark you created
* Test the bookmark by CTRL-clicking the button
* Find 2-3 additional insights from the Topline Performance tab and add new bookmarks and notes linking back