**Week 3 - Project**

Complete & Continue

Enable Fullscreen

Case Study

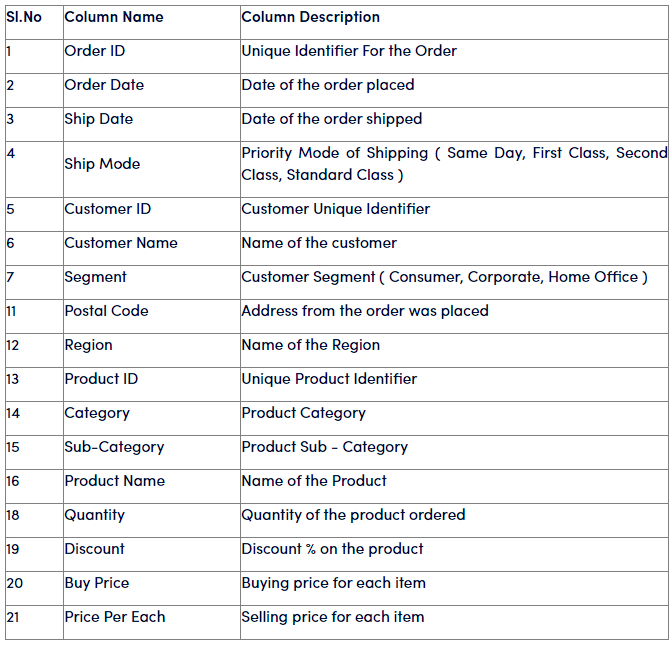
A superstore retail business is a large, multi-department store that sells various products, including groceries, electronics, home goods, clothing, and more. These stores are often designed to be a one-stop shop for customers, offering a wide range of products and services under one roof. Superstores are typically larger than traditional retail stores and may have a larger product selection. Superstores are often part of a larger chain and have multiple locations in a region or country.

A new store manager needs your help to better understand his/her Data Operations Team. You are provided with part of the sales data that a Business Intelligence Analyst encounters daily. Design the dashboard to analyse and interpret the data to help provide valuable insights to the store manager.

Dataset

A Superstore dataset typically includes information about the products, customers, and sales associated with a retail store. It may include the following columns:

**Table Description:**



**Problem Statement:**

Data Extraction, Cleaning, Loading and Transformation

1.  Desk representatives at the stores are not tech savvy hence they directly share the data in the single excel file. As a Power BI Developer, read the data directly from the excel file.

2.  The data coming from the source is in raw form in the flat file; hence clean and prepare (transform) the dataset for efficient use. Delete the empty columns and rows, change the fields to appropriate data types and split the fields and rename the columns appropriately.

3.  Standardize the values in the column Ship mode [Hint: Replace FC with first class]

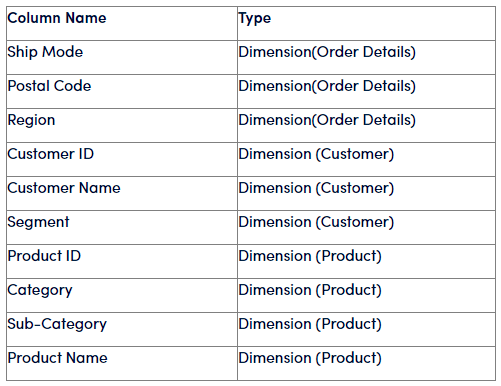
4.  Split the address column to City, State, Country and Pin code

Data Modelling:

1.  Tracking sales in the retail business is a weekly task; hence setting up the data model will be crucial for this. Convert a flat file into STAR schema for better performance of the analysis. The schema shall have a central Fact table, ‘Orders’ and three dimension tables, ‘Order details’, ‘Customer’ and ‘Product’. Refer to the table below for creating appropriate columns in each table.

2.  Remove duplicate rows from the newly created dimension tables, and ensure there are no empty rows

3.  Once the tables are created, ensure one- to - many relationships are created between dimensions and fact table.



Data Analysis:

Furthermore, to grow the footholds in the store and achieve an ambitious sales target, the store manager wants to track and visualize the following metrics;

1.  Create a new column ‘Sales’ or ‘Order value’ [Hint: Sales = Qty\*price per qty\*(1- % Discount)]. Create a card visual displaying the total sales.

2.  Similarly calculate the Sales from discounted products and display the total sales from discounted products.

3.  Since supermarkets sell bulk items, store managers want to know each order's cart value. Create a column “Cart Value” that categorizes the order value/sales as Low, medium, high or very high. 

**Cart Value**,

                  < 1000: Low

                  <3500: Medium

                  < 10000: High

                   > 10000: Very High

                [Hint: From Power Query editor->Add column-> Conditional column OR

                  DAX formula using nested IF() function]

Create a pie chart with Cart value as legend and Order value in Values field.

4.  Separately visualize the total sales just from the low cart value category (as mentioned, any value below 1000 can be considered as low value category).

5.  Using card visual, track the total sales coming from the low cart category and discount more than or equal to 50% to find out the contribution and cause.

[Hint: Create a new measure, sales using calculate () and sum() function with the two filters, Low cart category and discount>=50%   OR calculate sales using sumx() with if() and and()]

6.  Find out the number of days it takes to deliver for each shipment type (refer ship mode) so that delivery issues can be looked at on priority [Hint: No of days to deliver can be calculated from the difference between order\_date and shipping\_date]. Create a column chart that shows the average number of days it takes to deliver for each shipment type.

7.  So far, the store manager has managed to see the current snapshot of the sales based on various criteria. In the Retail business, do we see a spike in sales on special occasions like festivals? To achieve this, create a matrix visualization that displays order date as hierarchy, sales and sales year to date.

8.  Visualize the cumulative sales for each month for all the years to calculate Year on Year Sales Growth. Calculate YoY growth.

You are expected to provide a write-up explaining the data and any insights you can gather. You are encouraged to provide visuals and use basic statistics for this purpose.

**Requirements**:

1. A write-up of your analysis of the data.

2. Please choose an appropriate visual for better readability and interpretation.

3. Please avoid using jargon while explaining data and insights.

4. Please explain the calculation purpose and method.

**Deliverables:**

● A detailed write up of the analysis, capturing all the steps, methods and calculations (pdf or word file). There could be multiple ways of arriving at the solution. Detail any one method.

● PowerBI reports in .pbix format

ANSWERS: -

*Data Extraction – CLEANING, LOADING AND TRANSFORMATION*

1. At first appearance, reading the data at face value on excel feels a little overwhelming, as there is only a singular table that contains all the given columns as shown in the Table Contents (image 1). Going through it, it needs some Cleaning and transformations to be done. The steps that I took were, Selecting Excel workbook under home tab and adding the Superstore data to be worked with then import the data.
2. After Opening Power Query Editor (Clicking Transform Data under home tab), Look through the individual columns for their **respective datatypes**, and make changes wherever necessary, after that, check for empty data inside the Power Query Editor, select View and turn on Column Quality and Column Distribution, it shows all the respective columns and look for errors and blanks if present, then under Home, click on Remove Rows and select **remove duplicates** and **remove blank rows**, and also, select the downward arrow beside column name in each column and **uncheck null values**, this prevents errors in your data visualisation caused due to null values. In this particular dataset, **no files required any change like split or rename**. If any columns need to be split (e.g., a combined address field), we would use the Split column feature in the Transform tab. Rename the columns as needed by double-clicking the column headers.
3. Now to Standardize the values in the Ship Mode Column, we would go to the Power query Editor and select ’Ship Mode’ column and under ‘Transform’ tab, click on ‘Replace Values’, and then enter “FC” in ‘Value to Find’ field and “First Class” in the “Replace With” field. Then click on ‘OK’ to apply changes. These are the steps I took in Standardising Values.
4. As said in the question, ‘Postal Code’ gives the total entire address, but that is not present in the Dataset, what we have is a column named ‘Address’ that we need to split. Back to Power Query editor, select ‘Address’ and under ‘Transform’, click on ‘Split Column’, choose space for the first two times then choose ‘-’ to split the address into City, State, Country and Pin code. Rename the new columns as required.

*Self-Inference\_*

* After making all the necessary changes in the Power Query Editor, we click on ‘Close & Apply’.
* Certain issues and how I tackled them: product id was something like GC-13637 did I need to split this column as the question requires me to split wherever necessary but previously, I mentioned it wasn’t necessary? The answer is, no there is no need for splitting the product ID. If the ‘Product ID’ serves as a unique identifier and you don't need to analyse its components separately, we can leave it as is. However, if the prefix and number have distinct meanings or if we need to perform specific analyses, splitting the column can be beneficial.
* Now, since the data is Prepared, we can head into next step which is, Data Modeling.

*Data Modeling*

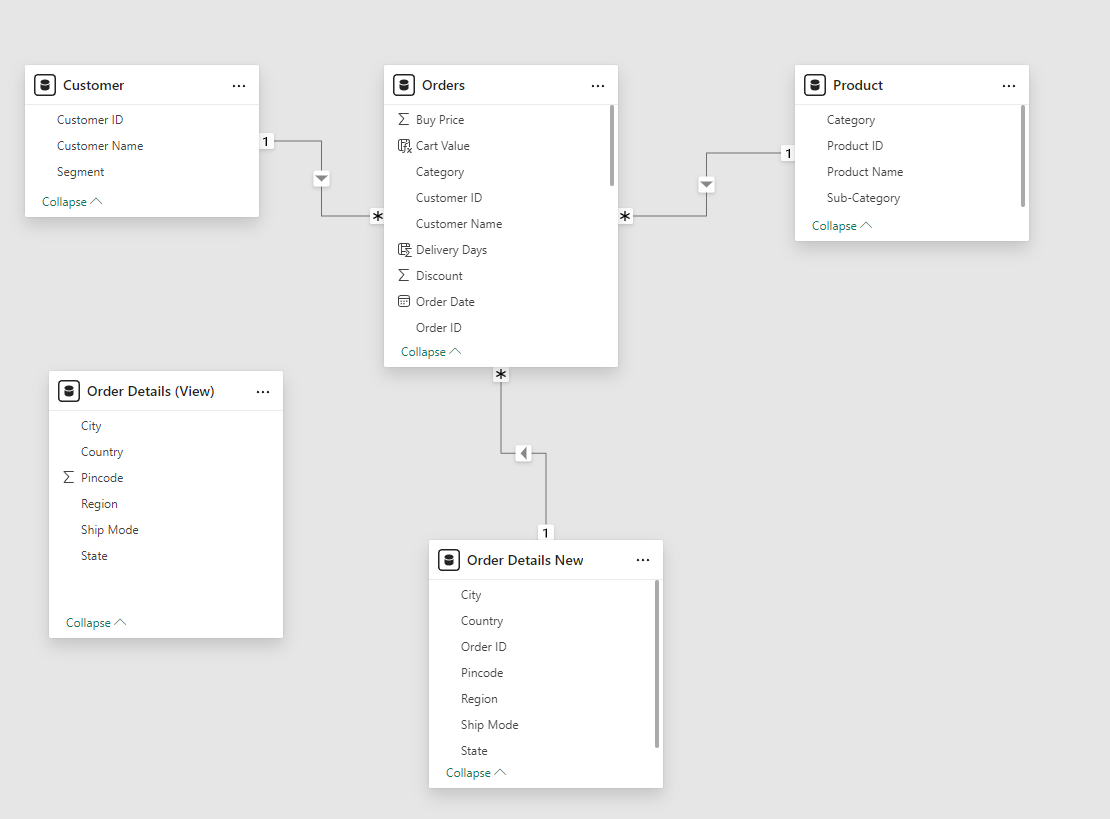
1. We are required to Convert this prepared flat file Database into a STAR schema, in order to do so we need to add the respected columns in the 3 different **Dimension tables**:  
   a. **Order Details Table**: Ship Mode, postal Code, Region  
   b. **Customer**: Customer ID, Customer Name, Segment  
   c. **Product**: Product ID, Category, Sub-Category, Product Name

To add these, the steps that I followed in the Power Query Editor was, click on the left pane of editor and select the initial orders database and duplicate it, after doing so, select the required columns in this duplicate table and right click and select ‘Remove Other Columns’ repeat the process for the next 2 tables.  
  
We already have a Fact Table named Orders, this is our **Fact Table**.

1. For achieving this, we need to select **all** the columns in the table and remove the duplicate rows and make sure there are no empty rows (we already have done this in data preparation but yet again, check if it’s empty or not), for each of the 3, dimension tables, select all the columns (but for the table having primary key, we can select only that primary key column, it will give the same result) and right click and choose ‘Remove Duplicates’. If there are any empty rows, filter them out by using the ‘Remove Rows’ feature.
2. Now to create relationships between each of the dimension table and fact table, we need to go to ‘Model view’ and create one-to-many relationships between the dimension tables and the fact table, the method I used was to select Manage Relationships under Home tab in Model view then selecting Auto Detect Relationships. Only two of the tables were able to form due to presence of foreign keys namely ‘Customer ID’ and ‘Product ID’ in the ‘Orders’ Fact table.   
     
   The Third, Dimension Table, namely ‘Order Details’ only had City, State, Country, Pin-code, Region and Ship Mode, there weren’t any Distinct identifiers so relationship couldn’t be formed (instead of a postal code, Address was given so it was not possible to firm any link between the respective table and the fact table, Postal Code is also considered as an address but it’s not unique identifier to each of the unique addresses that we have in short, there was no presence of a column of a foreign key).

*Self-Inference\_*

* There was no mention in if we were supposed to delete the files in the fact table, that was present in the 3, dimension tables, one of the reasons of building a schema is to increase data readability and to save space. But as it was not mentioned, I deleted only the common columns in customer and product table and leaving fact table with the necessary foreign keys to link to the dimension tables (Customer ID, Product ID). There was no foreign key available for the ‘order details’ hence relationship could not be made so I decided to leave the Fact table with the values of the ‘order details’ table.
* Why did I select all columns in ‘order details’ instead of the foreign key? The reason to this, is that we select all columns when we want to ensure that each row is unique based on all its data (just like ‘order details’). We can do both (both = select primary key containing column or select all columns) for the tables having foreign keys. Because We Select specific columns when we want to ensure uniqueness based on specific key columns, such as foreign keys.
* For each dimension table, we will select all the relevant columns and then remove duplicates based on those columns. This ensures that each record in the dimension table is unique based on the combination of the relevant columns.
* ‘Order Details’ Dimension Table has no Foreign key or Primary Key mentioned in the list of tables but however, we could add ‘Order ID’ and create relationships, otherwise this table would be used for descriptive purposes.  
    
  The OrderDetails dimension table is a bit different from the typical dimension tables because it doesn't have a straightforward primary key like Customer ID or ProductID. Instead, it contains descriptive attributes related to the order, such as ShipMode, Postal Code, and Region.  
    
  In a STAR schema, the OrderDetails table can be used to provide additional context to the Orders fact table. However, it doesn't directly link to the Orders table via a foreign key in the same way that Customer and Product do.
* But I do think that Order ID should be included in Order Details but it’s just not mentioned.

***MAJOR FIX: -***

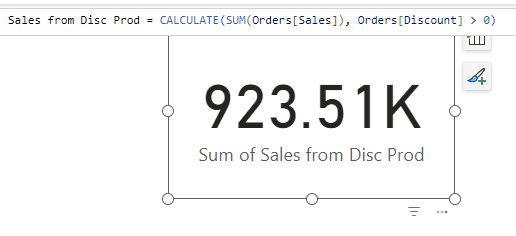
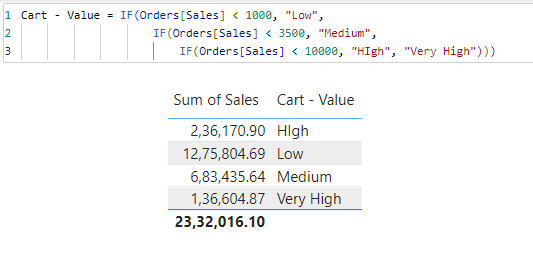
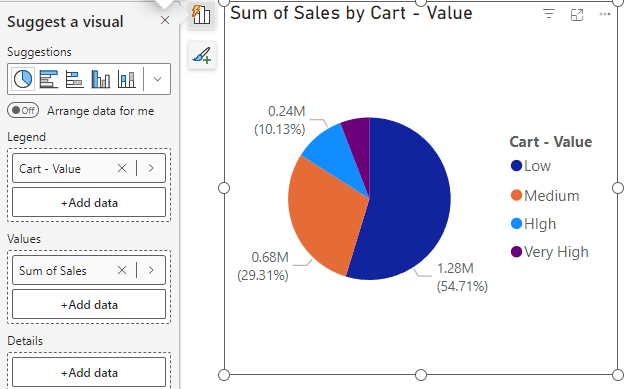
* Successfully formed relationships between all 3, dimension tables. I had an error, I had not loaded the ‘Order ID’, due to macros enabled I was unable to edit the file. The file was (xlsm) I disabled macros and turned it into (xlsx).
* And loaded that new source through the Load New Source Option in Power Query Editor.
* I saw that the Order ID was available on this one also another file which was being previously created named (col. 18) with all null values was showing up, I must’ve deleted order ID along with this col. 18 null column last time.
* The question even though doesn’t mention order ID to be a part of the ‘Order Details’ but still we can use it as a foreign key, it’s obvious it is supposed to also be a part of Order details.
* Also changed the datatype and cleared blanks and etc., and it started showing up in table view.
* Everything goes on as follows as mentioned above. All the steps.
* Created a duplicate of Orders table and removed all but Ship mode, Region, Orders ID, City, State, Country and Pin-code.
* I deleted the Ship mode, region, City, State, Country and Pin-code from ‘Orders Table’. As I had a one-to-many relationship showing on the model view.
* Hence, we established 3 relationships with the dimension tables, using ‘Order ID’, ‘Customer ID’ and ‘Product ID’.
* And I used the Order Details View as a descriptive purpose table.
* Now it works!

*Data Analysis*

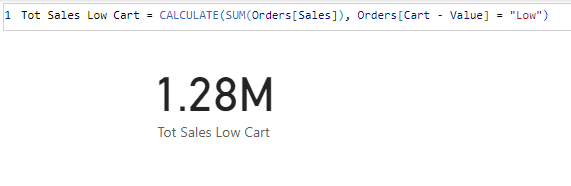
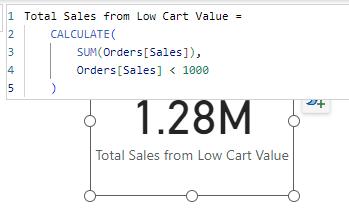
1. Create a new column by Going to ‘Data’ view and clicking on ‘Data’ icon on the left sidebar. Select the ‘Orders’ table from the Fields pane, in the ‘Table Tools’ ribbon, click on ‘New Column’ and Enter the Following DAX to create the ‘Sales’ or ‘Order Value’:  
     
    Sales = Orders[Quantity] \* Orders[Price Per Each] \* (1 - Orders[Discount])

Now, switch to Report View by clicking on the left sidebar to switch to the Report view. In the visualisations pane, click on the ‘Card’ visual icon to add a card visual to the report canvas. Drag the newly created ‘Sales’ column from the Fields pane to the ‘Values’ field well of the card visual.

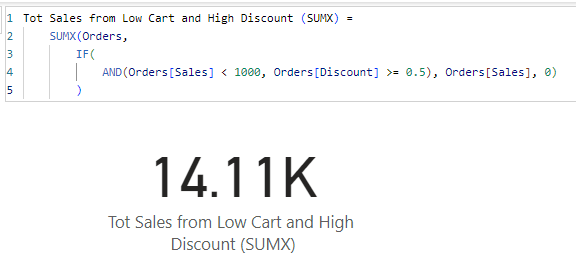


1. Following the similar steps of Number 1, we will proceed to create a new column in under ‘Column Tools’ and Enter the DAX:  
     
    Sales from Disc Prod = CALCULATE(SUM(Orders[Sales]), Orders[Discount] > 0)  
     
   
2. There are hinted two ways to do this problem, through Power Query Editor and the other Through coding in DAX, I chose the latter. As the first one is pretty much self-explanatory.  
     
   So, in order to create a nested IF, for Cart Value, select the ‘New Column’ and write the following DAX Statement:  
     
     
     
   Now, select Pie Chart in Visualisation panel in Report View and Add Cart Value in Legend and Sales/Order values in Value:  
     
   

Cart – Value = IF(Order[Sales] < 1000, “Low”,  
 IF(Order[Sales] < 3500, “Medium”,  
 IF(Order[Sales] < 10000, “High”, “Very High”)

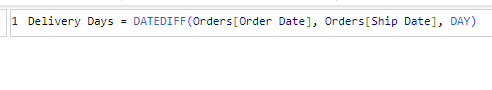
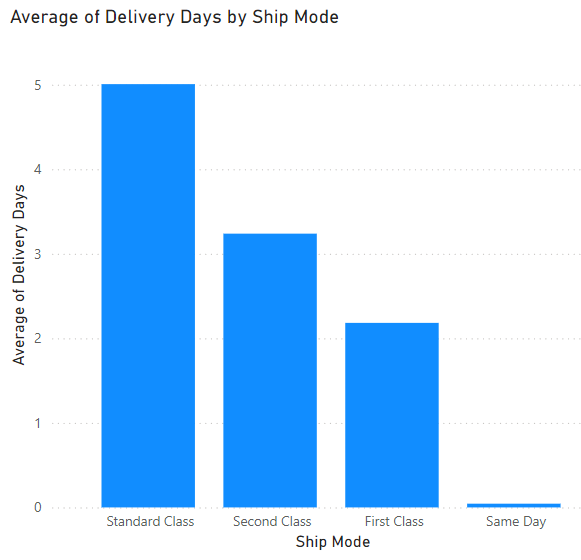
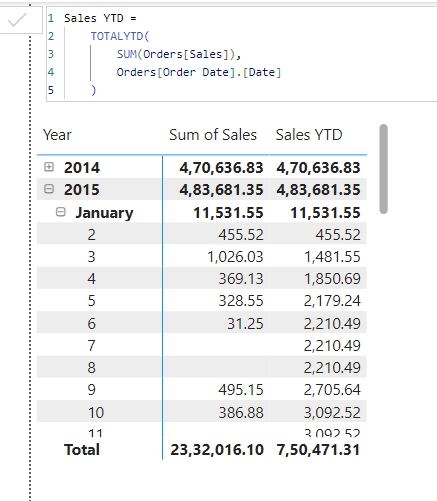
1. In this Case we will be creating a measure, following same steps as while creating the New Column under Table tools. Select New Measure under Table Tools if you’re in Data View or New Measures under Home in Report View. We can solve this using two methods:  
     
   a. using previously made column ‘Cart – Value’:  
     
      
     
   b. using the hint (Sales < 1000) directly from the question:  
     
    Total Sales from Low Cart Value = CALCULATE(SUM(Orders[Sales]), Orders[Sales] < 1000)  
     
   

Tot Sales Low Cart = CALCULATE(SUM(Orders[Sales]), Orders[Cart - Value] = “Low”)

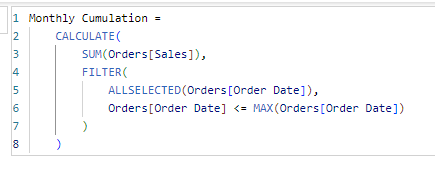
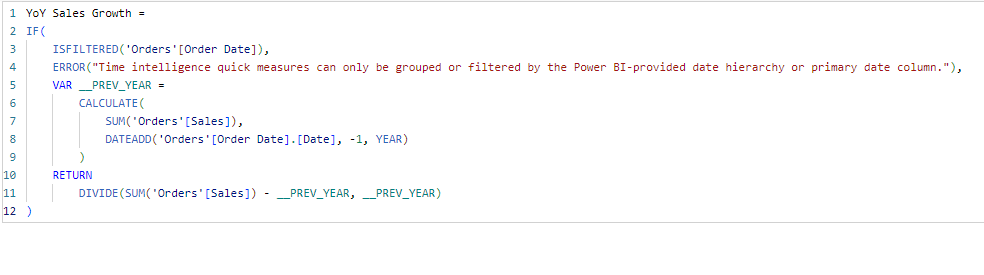
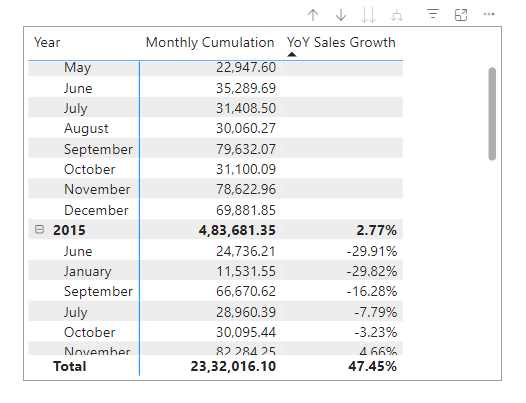
1. To solve this, we will be creating a measure once again, this problem can be achieved by two hinted steps:  
     
   a. Create a new measure, sales using calculate() and sum() function with the two filters, Low cart category and discount>=50%:  
     
     
     
   b. and the second hinted method to calculate sales using sumx() with if() and and():  
     
     
   SumX calculates row level operation and inculcating IF and And is a clever way to filter out the required rows, the IF statement proves true only if both the conditions are satisfied, and adds the respective sales data which is below 1000 and has equal or more than 50% discount, if the IF statement is false, it returns 0 to be added.  
     
   

Tot Sales from Low Cart and High Discount (SUMX) =   
 SUMX(Orders,  
 IF(AND  
 AND(Orders[Sales]) < 1000, Orders[Discount] >= 0.5), Orders[Sales], 0  
 )

Tot Sales from Low Cart and High Discount =   
 CALCULATE(  
 SUM(Orders[Sales]),   
 Orders[Cart - Value] = “Low”,  
 Orders[Discount] >= 0.5  
 )

1. To solve this problem, we can use a Date time Function called DATEDIFF, that gives us the difference between two given dates here specifically, order\_date and shipping\_date:  
     
   Delivery Days = DATEDIFF(Orders[Order Date], Orders[Ship Date], Day)  
     
     
     
     
     
   Now, we will create a column chart as directed on the question, in order to do that, we will select the respective visualisation (column chart) under visualisation model tabs which is under home, and select ‘clustered column chart’ and add it to our visualisation and then put the no of days (Delivery Days) on the y column (Y-axis) and Shipment type (here Ship mode) in (X-axis). But we won’t get the solution yet, we are getting the total days of each shipment type, we need to convert the aggregation to average from sum as by default it is always showing sum (to do this, there are multiple ways but the way I used is, I clicked on the visualisation and then two icons appear, click the top icon that has a thunder symbol over column chart; this is the building visual options and under it, we can see that the visual on y axis has an arrow for additional settings pointing on the right, click on it. And click on summarisation options and select Average, which is the required aggregation).  
     
   
2. In order to display the required criteria, we have to calculate sales year-to-date (Sales YTD is a financial metric which measures the total sales revenue generated by a business from the beginning of the current fiscal year to the present date), we will be using the function called TOTALYTD() which calculates sales; the year-to-date total for a given expression (in this case, the sum of sales up to the specified date):  
     
     
     
     
     
     
     
     
     
     
   My inference after failing to output the desired result; the .[Date] is absolutely necessary, because the date column is in a hierarchy, we need to mention the specific ‘date’, and if not, we won’t get the desired result, and it will work the same way as sum where it displays the summation aggregation of the entire sum of the sales – we don’t want that. The notation Orders[Order Date].[Date] suggests that Order Date is a hierarchy, and you're specifically referencing the Date level of that hierarchy. This is common in tools like Power BI when working with date dimensions that have been set up as hierarchies.  
     
     
     
     
   After getting the Sales YTD, we need to add matrix visualisation to our dashboard from under Home, and then drag and drop order Date (Hierarchy) and then Sales and Sales YTD on the Matrix, to have the desired output.  
     
   We could have done it using quick measures as well. But we used measures and wrote the code. There are multiple ways to do it.

Sales YTD =  
 TOTALYTD(  
 SUM(Orders[Sales]),  
 Orders[Order Date].[Date]  
 )

1. In order to solve this question, there are a handful of things to keep in mind, the cumulative sales for each month can be displayed by just adding the ‘sales’ (calculated) column on the visualisation of the order dates, or we can make a summation measure of it.  
     
   Monthly Cumulation = SUM(Order[Sales])   
     
   **There was no need** for using FILTER function or ALLSELECTED or CALCULATE, but just for my knowledge I had added them and they serve a purpose as well.  
     
     
     
   Components of the Function:  
     
   CALCULATE: This function changes the context in which data is evaluated. In this case, it calculates the sum of sales while applying a filter.  
     
   SUM(): This part sums up the sales from the Orders table. It gives you the total sales amount for the specified context.  
     
   FILTER(): The FILTER function is used to create a table that only includes rows that meet certain conditions. In this case, it filters the dates based on the condition specified.  
     
   ALLSELECTED(): This function is crucial for understanding how the context is modified. ALLSELECTED returns all the values of Orders[Order Date] that are currently selected in the report or visual, while ignoring any filters that might have been applied to that column. This means that if you have slicers or filters in your report that affect the date, ALLSELECTED will consider those selections but still allow the cumulative calculation to include all relevant dates up to the current date in context.  
     
   Orders[Order Date] <= MAX(Orders[Order Date]): This condition checks if the order date is less than or equal to the maximum order date in the current context. Essentially, it means "include all orders up to and including the maximum date being evaluated  
     
   It was absolutely unnecessary but I just did it the extra way and I am explaining the steps involved – Putting it all together, this formula calculates the cumulative sales for the current context (like a month or year by); Summing the sales from the ‘orders’ table, Filtering the dates to include only those that are less than or equal to the maximum date currently being evaluated and using ‘ALLSELECTED’ allows the calculation to respect any filters applied in the report while still considering all relevant dates up to that point.  
     
   Now, moving on, we can calculate the YoY (Year over Year Growth) by using a quick measure:  
     
     
     
   Select Quick Measure >> select a calculation >> Under Time Intelligence Functions >> Year over Year Change >> Add Base Value as Orders[Sales] and Date as Orders[Order Date] >> Rename it to “YoY Sales Growth”.  
     
   Finally Now, Add a visualisation; Matrix and Add the ‘Order Date’, but remove ‘Quarter’ and ‘Date’ hierarchy, as our calculation only requires us to see the Monthly Cumulative sales, we do this in order to provide the viewer with necessary insights which are relevant.  
     
   And Hence, it looks like this:  
     
   

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Xtra – Self:

Cart - Value = IF(Orders[Sales] < 1000, "Low",

                    IF(Orders[Sales] < 3500, "Medium",

                        IF(Orders[Sales] < 3500, "High",

                            IF(Orders[Sales] < 10000, "HIgh", "Very High"))))