### Task 1: Conditional Column – Order Priority

**Question:**Create a new column **Order Priority** using this logic:

* If Sales ≥ 500 and Discount = 0 → "High"
* If Sales between 200 and 500 → "Medium"
* Else → "Low"

**Answer:**

Use Invoke custom column,

**if ([Sales] >= 500 and [Discount] = 0) then "High" else if ([Sales] > 200 and [Sales] < 500) then "Medium" else "Low"**

### Task 2: Column from Examples – Brand Name

**Question:**From Product Name, create a new column with the **first two words** (brand).

** Go to Power Query Editor.**

** Select your Orders table.**

** Go to the "Add Column" tab.**

** Click on "Column from Examples" -> "From Selection" (if you select the Product Name column first) or "From All Columns".**

** In the new column, for the first few rows, type the expected output (the first two words of the Product Name). Power Query will automatically detect the pattern.**

** Rename the new column (e.g., Brand Name).**

**Task 3: Create a DimDate Table**

**Question:**

**Create a DimDate table including:**

* **Year**
* **Month Name**
* **ISO Week**

**Answer (DAX Calculated Table):**

**DimDate =**

**ADDCOLUMNS (**

**CALENDAR ( MIN ( Orders[Order Date] ), MAX ( Orders[Order Date] ) ),**

**"Year", YEAR ( [Date] ),**

**"Month Number", MONTH ( [Date] ), // Added for correct sorting of Month Name**

**"Month Name", FORMAT ( [Date], "MMMM" ),**

**"ISO Week", WEEKNUM ( [Date], 21 ) // 21 for ISO 8601 standard (week starts Monday, first week has 4+ days)**

**)**

**Explanation:** This DAX calculated table dynamically generates a DimDate table covering the full range of Order Date values from your Orders table. It includes essential time intelligence attributes such as Year, Month Name, and ISO Week. The Month Number column is added to ensure that Month Name can be sorted chronologically in visuals. This DimDate table is crucial for enabling robust time-based analysis and DAX time intelligence functions.

**Task 4: Customer Retention**

**Question:**

**Count customers who purchased in more than one distinct year.**

**Answer (DAX Measure):**

**Customers\_Purchased\_Multiple\_Years =**

**COUNTROWS (**

**FILTER (**

**SUMMARIZE (**

**Orders,**

**Orders[Customer ID], // Group by Customer ID**

**"Distinct\_yr", CALCULATE ( DISTINCTCOUNT ( 'DimDate'[Year] ) ) // Count distinct years for each customer**

**),**

**[Distinct\_yr] > 1 // Filter for customers with more than one distinct year**

**)**

**)**

**Explanation:** This DAX measure calculates the total number of customers who have made purchases in more than one unique year. It first creates a virtual table summarizing each Customer ID and counting their Distinct\_yr of purchase (from the DimDate table). It then filters this virtual table to include only customers where Distinct\_yr is greater than 1, and finally counts the rows (customers) in the filtered table.

**Task 5: Profit Margin for Technology**

**Question:**

**Calculate Profit Margin (Profit ÷ Sales) specifically for the "Technology" category.**

**Answer (DAX Measure):**

**Profit\_Margin\_by\_Tech =**

**CALCULATE (**

**DIVIDE ( SUM ( Orders[Profit] ), SUM ( Orders[Sales] ), 0 ), // Calculate Profit / Sales**

**Orders[Category] = "Technology" // Apply filter for Technology category**

**)**

**Explanation:** This DAX measure calculates the overall profit margin for the "Technology" product category. The CALCULATE function modifies the filter context, ensuring that the SUM of Profit and SUM of Sales are computed only for orders where the Category is "Technology". The DIVIDE function safely handles any potential division by zero. After creating this measure, remember to format it as a **Percentage** in Power BI's "Measure tools" to display it correctly (e.g., 17.40%).

**Task 6: Classification on Discount & Profit**

**Question:**

Create a column to classify orders into:

* “Discounted Loss”
* “Healthy Sale”
* “Risky Sale”
* “Others”

**Answer (DAX Calculated Column):**

**Classify\_orders =**

**IF (**

**Orders[Profit] < 0 && Orders[Discount] > 0,**

**"Discounted\_Loss",**

**IF (**

**Orders[Profit] > 0 && Orders[Discount] = 0,**

**"Healthy\_sale",**

**IF (**

**Orders[Profit] > 0 && Orders[Discount] > 0,**

**"Risky\_sale",**

**"Others"**

**)**

**)**

**)**

**Explanation:** This DAX calculated column categorizes each order based on its profit and discount values. "Discounted Loss" indicates a negative profit with a discount applied. "Healthy Sale" signifies a positive profit with no discount. "Risky Sale" denotes a positive profit despite a discount, suggesting a potential compromise on margin. All other scenarios fall under "Others".

**Task 7: Clean Product Name**

**Question:**

**Remove all spaces and convert to uppercase from the Product Name column.**

**Answer (DAX Calculated Column):**

**Cleaned Product Name = UPPER(SUBSTITUTE(Orders[Product Name], " ", ""))**

**Explanation:** This DAX calculated column Cleaned Product Name takes the original Product Name from the Orders table. It first uses SUBSTITUTE to replace all spaces (" ") with an empty string (""), effectively removing them. Then, UPPER converts the entire resulting string to uppercase, ensuring a standardized format for analysis.

**Task 8: Logical Match Column**

**Question:**

Return TRUE if Ship Mode = “Second Class” AND Segment = “Corporate”

**Answer (DAX Calculated Column):**

**ShipMode\_Category =**

**IF (**

**Orders[Ship Mode] = "Second Class" && Orders[Segment] = "Corporate",**

**"True", // Returns the string "True"**

**"False" // Returns the string "False"**

**)**

**Explanation:** This DAX calculated column ShipMode\_Category evaluates each row in the Orders table. If both conditions (Ship Mode is "Second Class" AND Segment is "Corporate") are true, it returns the string "True"; otherwise, it returns "False". This column helps in quickly identifying specific order characteristics.