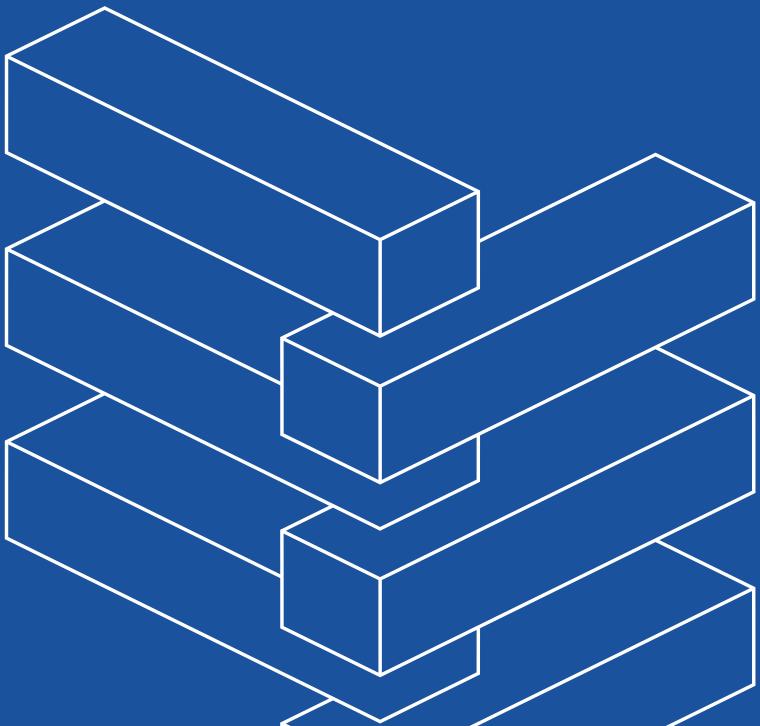


# Sales Analysis

Empower Businesses to make Data-Driven Decisions



Prepared By

Mazen Tamer  
Motasem Amr  
Philopateer Maher  
Ahmed Walid

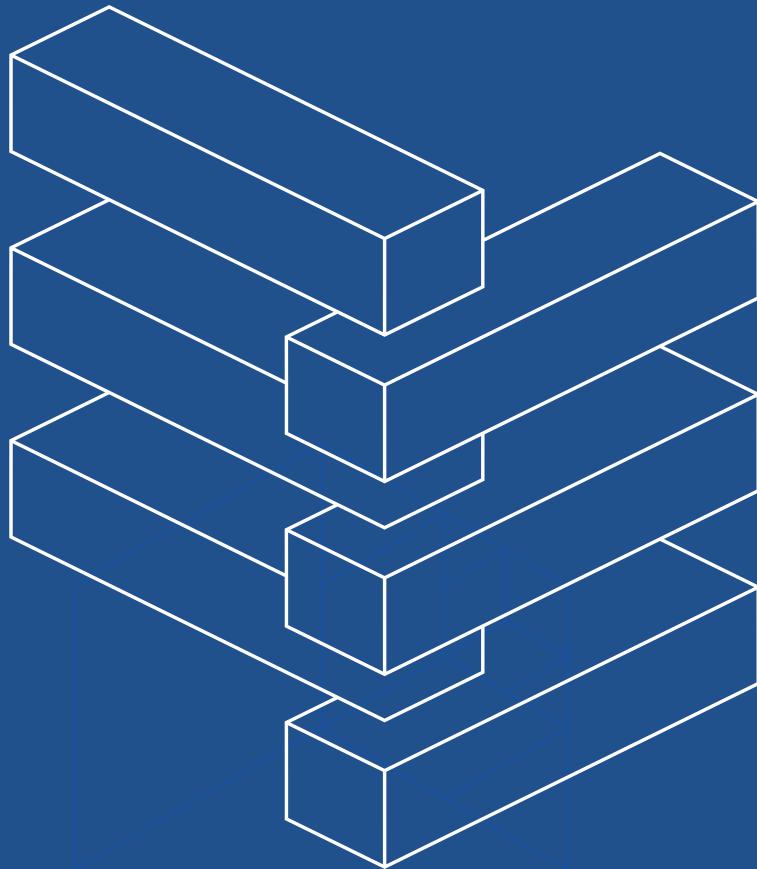
# Project Methodology

1- Data Understanding

2- *Data Modelling*

3- *DAX Measures*

4- Dashboard Building



# Data Understanding & Dataset Overview

analysis of a comprehensive Sales dataset. The data provides a 360-degree view of the sales lifecycle, focusing on the transition from Order placement to actual Delivery. It is designed to support strategic decision-making by identifying operational gaps and historical trends

PromotionKey	UnitCost	UnitPrice	SalesQuantity	ReturnQuantity	ReturnAmount	DiscountQuantity	DiscountAmount	TotalCost	SalesAmount	Delivery_Date
1	\$321.44	\$699	10	0	\$0	0	\$0	\$3,214.4	\$6,990	06/11/11 12:00:00 AM
1	\$321.44	\$699	10	0	\$0	0	\$0	\$3,214.4	\$6,990	08/02/13 12:00:00 AM
1	\$321.44	\$699	10	0	\$0	0	\$0	\$3,214.4	\$6,990	02/02/14 12:00:00 AM

- Financials:** Includes Sales Amount, Unit Price, and Total Cost for revenue and margin analysis.
- Logistics:** Features Delivery\_Date as a secondary date key to track fulfillment performance.
- Adjustments:** Captures Discount Amount and Return Quantity to measure the impact of promotions and cancellations.
- Dimensions:** Connected to PromotionKey and other unique identifiers for deep-dive slicing.

# Columns Characteristics & Data Types

The dataset is architected into two main categories: Fact Tables (capturing quantitative business events) and Dimension Tables (providing descriptive context). This separation ensures high granularity and enables multi-dimensional analysis of sales and marketing performance.

## Fact Tables

- Sales Fact: The primary table containing transactional records. Key columns: SalesAmount, TotalCost, UnitCost, and UnitPrice.
- Marketing & Adjustments (Fact-like): Handles promotion-related data and sales corrections. Key columns: PromotionKey, DiscountAmount, DiscountQuantity, and ReturnQuantity.

## Dimension Tables

- Snowflake Dimensions: Categorical hierarchies including Category (linked to Products), Product details, and Geography (Sub-region to Region).
- Date Dimensions: The DateTable for standard time intelligence and a specialized TooltipDate for independent historical comparison.

# Detailed Columns Definitions

- **SalesAmount:** Represents the total gross revenue generated per transaction before deductions. (Numeric - Currency).
- **TotalCost:** The cumulative cost of production or acquisition for the units sold, essential for gross profit calculation. (Numeric).
- **UnitPrice & UnitCost:** Individual price and cost points used to analyze pricing strategies and product margins. (Numeric).
- **SalesQuantity:** The number of units sold in a single transaction, used to track volume trends. (Numeric).
- **Delivery\_Date:** Critical for tracking fulfillment latency, operational efficiency, and delivery performance metrics. (Date).
- **PromotionKey:** A unique identifier linking sales to specific marketing campaigns to measure ROI and campaign effectiveness. (Key).
- **DiscountAmount & Quantity:** Tracks the financial impact and volume of promotional offers applied to orders. (Numeric).
- **ReturnQuantity & Amount:** Captures product reversals and refunds, vital for calculating "Net Sales" and identifying quality issues. (Numeric).

# Data Modeling



## Snowflake Schema Design for Sales Analysis

A Snowflake Schema is a sophisticated data modeling technique used in this project to enhance data normalization and reporting accuracy. Unlike a traditional star schema, this model breaks down complex dimension tables into related sub-dimensions (e.g., Categories linked to Products, and Sub-regions linked to Regions). This hierarchical structure ensures maximum data integrity and optimizes query performance for multi-level sales analysis.

# Key Components of the Architectural

- Central Fact Table (Sales): The heart of the model, capturing transactional data, sales amounts, and quantities. It is linked to various dimensions via optimized surrogate keys.
- Hierarchical Dimensions (The Snowflake Effect): To achieve maximum granularity, the Product dimension is normalized into three levels: Product → ProductSubcategory → ProductCategory. This allows stakeholders to analyze performance from a broad category view down to specific individual items.
- Geographic & Operational Scaling: The model links Geography through Stores to the central sales data, enabling precise regional performance tracking.
- Multi-Calendar Logic: The model incorporates a primary Date table for core time-intelligence, alongside a New Date Table designed for specialized analytical comparisons (such as the 3-year historical tooltip).
- Strategic Dimensions: Includes Promotion, Channel, and Marketing tables to provide a 360-degree view of the factors driving sales growth.
- By implementing this Snowflake structure, the system maintains a 'Single Source of Truth,' allowing for complex 'Drill-Down' capabilities and ensuring that every data point is accurately attributed across the business hierarchy."

# DAX Measures

## Question 1 - Monthly Trend & Historical Benchmarking

### 1-Objective

The primary objective of this visual is to analyze the monthly efficiency of the sales cycle. By comparing 'Ordered Amount' against 'Delivered Amount' over a 3-year horizon, we can identify seasonal trends, delivery bottlenecks, and year-over-year growth pattern



# DAX Measures

## 2-Visual Implementation

- Visual Type: Multi-series Line Chart.
- X-Axis: Month (from DateTable).
- Y-Axis: Shared scale for both Ordered and Delivered metrics to ensure accurate comparison.
- Interactivity: Fully dynamic, responding to 'Single-Select' Year and Month slicers.

## 3-Advanced DAX Logic

- Ordered Selected: `CALCULATE([Ordered Amount], DATEADD('Date'[DateKey], 0, YEAR))`
- Ordered LY-1: `CALCULATE([Ordered Amount], SAMEPERIODLASTYEAR('Date'[DateKey]))`
- Ordered LY-2: `CALCULATE([Ordered Amount], DATEADD('Date'[DateKey], -2, YEAR))`  
(Note: Similar logic was applied to 'Delivered' metrics to ensure consistency)

# DAX Measures

## 4-Business Insights

- Trend Identification: This visual helps management understand if delivery capacity is scaling alongside order growth.
- Historical Benchmarking: Comparing the current month to the same month in previous years provides a realistic KPI for sales team

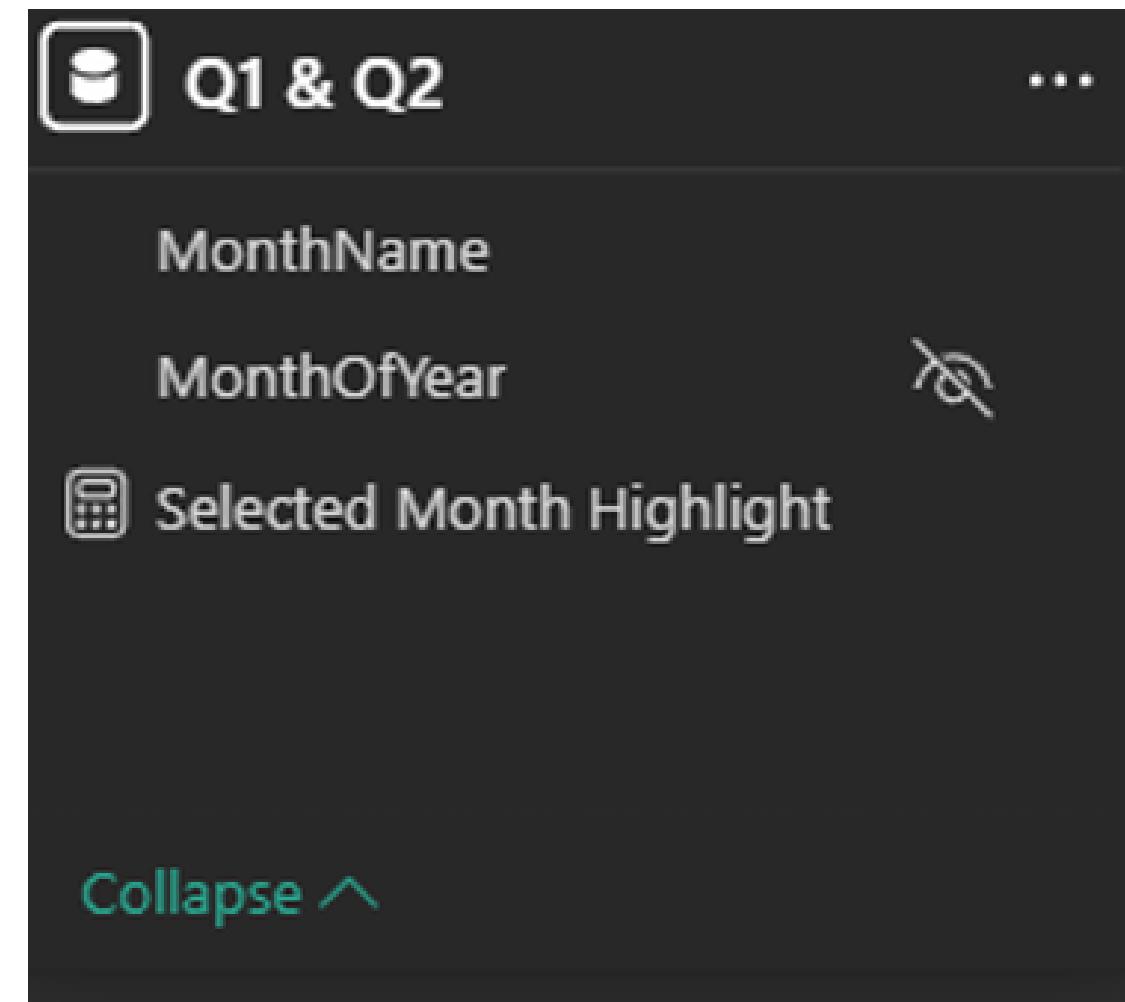


# DAX Measures

## Question 2: Strategic Month Highlighting within Annual Trends

### 1-Objective

The objective is to allow users to focus on a specific month's performance while maintaining the full-year context. Instead of filtering the chart to show only one month, we 'highlight' the selected month on the trend line, providing a clear comparison against previous and subsequent months



# DAX Measures

## 2-DAX Logic

Selected Month Highlight =

```
VAR Selected_Month = SELECTEDVALUE('Q1 & Q2'[MonthName])
```

```
VAR Current_Month = MAX('Date'[MonthName])
```

```
RETURN
```

```
IF(Selected_Month = Current_Month, [Ordered Amount], BLANK())
```

## 3-Visual Implementation

- Visual: The measure is added to the Line Chart as a secondary series.
- Markers: A distinctive color and marker (e.g., a bright dot) are applied to the 'Selected Month Highlight' series.
- User Experience: When a user selects a month from the slicer, a dot appears on the line chart at that specific month, while the full trend line remains visible for context.

# DAX Measures

## Question 3 - Row-Level Security (RLS) Implementation

### 1-Objective

The objective of implementing Row-Level Security (RLS) is to restrict data visibility dynamically based on the logged-in user.

The system ensures that:

- Each user can only see the Product Category and Sub-Category assigned to them.
- Data access is controlled automatically using the user's login email.
- Different users see different values in Cards, Tables, and Visuals without duplicating reports.

Example:

User A can see Car and Video categories and their related subcategories.

User B can see TV and Video categories and only specific subcategories like Car Video.

This guarantees secure, role-based, and dynamic data access.

User	Category	Sub-Category
A@abc.com	TV and Video	NULL
B@abc.com	TV and Video	Car Video

# DAX Measures

## Question 3 - Row-Level Security\_(RLS)\_Implementation

### 2-DAX Implementation

#### 2.1-User Matching (Security Table Filter)

`LOWER(TRIM(SecurityTable[User])) =  
LOWER(TRIM(USERPRINCIPALNAME()))`

#### Purpose

- Ensures case-insensitive and space-trimmed matching between the logged-in user and the Security Table to avoid authentication mismatches.

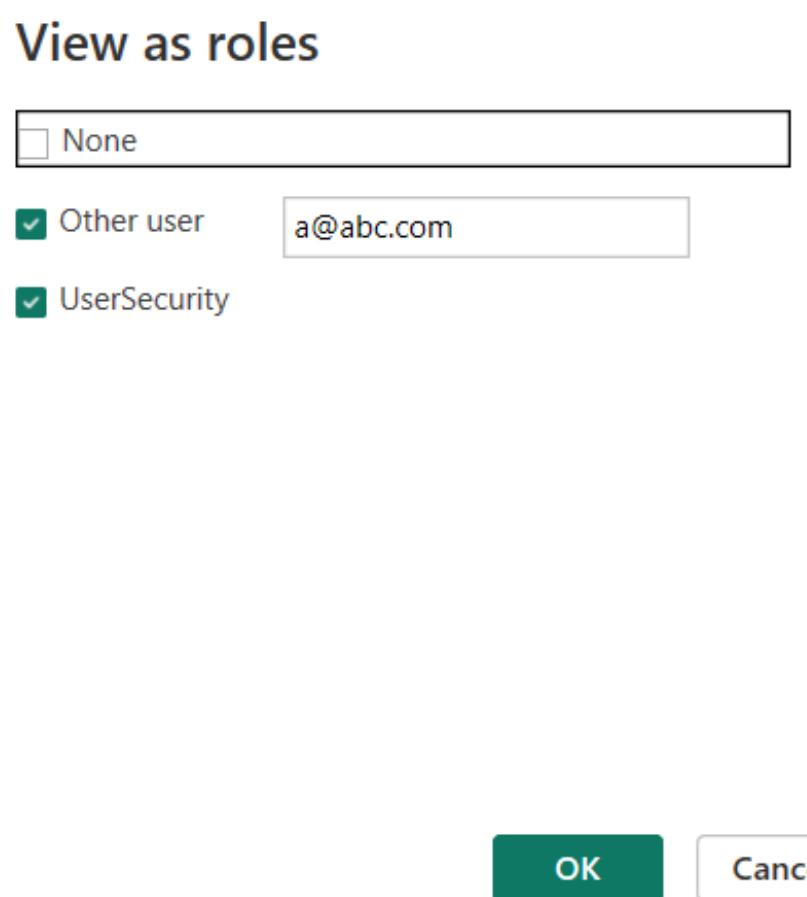
# DAX Measures

## 2.2-Category-Level Filtering

```
VAR CurrentUser = USERPRINCIPALNAME()
```

```
VAR UserCat =  
    MAXX(  
        FILTER(SecurityTable, SecurityTable[User] = CurrentUser),  
        SecurityTable[Ctaegory]  
    )
```

```
RETURN  
[ProductCategory] = UserCat
```



# DAX Measures

## 2.3-Sub-Category-Level Filtering

```
VAR CurrentUser = USERPRINCIPALNAME()
```

```
VAR UserSubCat =
    MAXX(
        FILTER(SecurityTable, SecurityTable[User] = CurrentUser),
        SecurityTable[Sub-Category]
    )
```

```
RETURN
IF(
    UserSubCat = "NULL",
    TRUE(),
    [ProductSubcategory] = UserSubCat
)
```

**View as roles**

None

Other user

UserSecurity

**OK** **Cancel**

The screenshot shows a Power BI report with two visualizations: a bar chart titled "Total Sales by ProductCategory" and another titled "Total\_Sales by ProductSubcategory". Both charts show a single blue bar representing sales figures. A date selector on the right allows users to choose between 2011, 2012, or 2013.

# DAX Measures

## Question 4 - Strategic KPI Analysis & Optimiz

### 1-Objective

The challenge was to monitor 5 core KPIs (Revenue, Profit, Margin, Orders, Quantity) across 5 different time-intelligence variations. Instead of the traditional, high-maintenance approach of writing 30 separate measures, we implemented Calculation Groups to achieve maximum efficiency and scalability."

### 2-Calculation Items (DAX Logic)

- KPI LY (Last Year):`CALCULATE(SELECTEDMEASURE(), SAMEPERIODLASTYEAR('Date'[Date]))`
- KPI LY YTD:`CALCULATE(TOTALYTD(SELECTEDMEASURE()), 'Date'[Date]), SAMEPERIODLASTYEAR('Date'[Date]))`
- Actual vs Target:`SELECTEDMEASURE() - (CALCULATE(SELECTEDMEASURE()), SAMEPERIODLASTYEAR('Date'[Date])) * 1.1`
- Growth %:`DIVIDE(SELECTEDMEASURE() - [LY_YTD_Value], [LY_YTD_Value], 0)`

# DAX Measures

## 3-Visual Implementation

- Visual: A Matrix or Multi-KPI Card visual showing the 5 base KPIs sliced by the Calculation Group items.
- Target Logic: Assumed a 10% strategic growth target over the previous year's performance.
- Result: A clean, professional table that compares current performance against historical and target benchmarks with minimal DAX footprint.



# DAX Measures

## Question 5 - Dynamic Top 5 Subcategories for the Highest Sales Category

### **1-Objective**

- Identify the Category with the highest Sales dynamically.
- Display the Top 5 Subcategories within that Category.
- Ensure the result changes dynamically based on slicers (e.g., Year).
- Avoid static filtering (e.g., fixed on “Computers”)

### **2-DAX Implementation**

Category Rank =  
RANKX(  
ALLSELECTED(ProductCategory[ProductCategory]),  
[Total\_Sales],

,

DESC

)

Is Top Category =  
IF ( [Category Rank] = 1, 1, 0 )

Subcategory Rank =  
RANKX(  
ALLSELECTED(ProductSubcategory[ProductSubcategory]),  
[Total\_Sales],

,

DESC

)

Is Top 5 Subcategory =  
IF ( [Subcategory Rank] <= 5, 1, 0 )

# DAX Measures

## Question 5 - Dynamic Top 5 Subcategories for the Highest Sales Category

### 3-Visual Configuration

- Bar Chart
- Axis → Subcategory
- Values → Total Sales
- Filters applied:
  - Is Top Category = 1
  - Is Top 5 Subcategory = 1
  - Year slicer added for dynamic behavior

# DAX Measures

## Question 5 - Dynamic Top 5 Subcategories for the Highest Sales Category



# DAX Measures

## Question 6 - Country Sales Analysis with Drillthrough

### 1-Objective

- Analyze each country's Sales using Year and Month slicers.
- Implement Drillthrough functionality to navigate to a detailed page.
- Ensure that the Drillthrough page displays the full historical data of the selected country, regardless of the selected Year on the main page.

This guarantees separation between the main page filter context and the drillthrough historical analysis.

### 2-DAX Measures

```
Sales All Years =  
CALCULATE(  
    [Total Sales],  
    REMOVEFILTERS('Date'[Year])  
)
```

# DAX Measures

## Question 6 - Strategic Drill-through & Historical Overrides

### 3-DAX Logic Explanation

- CALCULATE() modifies the filter context.
- REMOVEFILTERS('Date'[Year]) removes the Year filter only.
- Month filter remains active.
- Country filter remains active.
- This allows the drillthrough page to show all historical years even if a specific year (e.g., 2012) is selected in the main page.

# DAX Measures

## Question 6 - Strategic Drill-through & Historical Overrides

The screenshot shows a Power BI report interface. On the right, there is a 'Select' button followed by two dropdown menus: 'Year' (with 2012 selected) and 'Month' (with August selected). Below these are two more dropdown menus: 'Year' (with 2012 selected) and 'Month' (with August selected). In the center, there is a table titled 'Sales All Years' with columns 'RegionCountryName' and 'Sales'. The table lists various countries with their total sales. A context menu is open over the 'China' row, with the following options: Copy, Show as a table, Include, Exclude, Drill through (selected), Summarize, New visual calculation (selected), and Set up a verified answer. The 'Drill through' option points to a new visual calculation named 'Q6\_details'.

RegionCountryName	Sales All Years
Armenia	\$26,084,935.2425
Australia	\$79,166,589.651
Bhutan	\$30,036,626.1635
Canada	\$175,942,752.728
China	\$1,063,856,268.9995
Denmark	909.4768
France	66.8798
Germany	81.7669
Greece	33.1011
India	462.133
Iran	19.4661
Ireland	51.2011
Italy	3103,475,755.747
Japan	\$8,341,224,364.8324
Total	

The screenshot shows a Power BI report interface. On the left, there is a table with columns 'Year', 'Total Sales', 'Total Quantity', and 'Total Cost'. The table data is as follows:

Year	Total Sales	Total Quantity	Total Cost
2011	\$308,391,585.3175	1140384	\$131,289,272.24
2012	\$358,389,358.0935	1549178	\$152,388,077.36
2013	\$397,075,325.5885	2298600	\$171,520,644.66
Total	\$1,063,856,268.9995	4988162	\$455,197,994.26

In the center, there is a card titled 'Selected Country' containing the text 'China'.

# DAX Measures

## Question 7 - Country Sales % Contribution Within Continent

### 1-Objective

1. The objective of this analysis is to calculate each country's Sales contribution as a percentage of the total Sales of its Continent.
2. The percentage should remain constant even when a Country slicer is applied.
3. This ensures that the calculation always reflects the country's share relative to the entire Continent, not the filtered selection.

### 2-The DAX Logic

```
% of Continent =  
    VAR CountrySales = [Total_Sales]  
    VAR ContinentTotal = CALCULATE([Total_Sales], ALL('Geography'[RegionCountryName]))  
    RETURN  
        DIVIDE(CountrySales, ContinentTotal)
```

# DAX Measures

## Question 7 - Country Sales % Contribution Within Continent

Region	Country Name	Total Sales	% of Continent
	United States	\$4,756,351,262.7945	57.02%
	China	\$1,063,856,268.9995	12.75%
	Germany	\$662,755,181.7669	7.95%
	France	\$433,928,266.8798	5.20%
	United Kingdom	\$220,900,594.1916	2.65%
	Canada	\$175,942,752.728	2.11%
	Japan	\$163,473,755.747	1.96%
	Australia	\$79,166,589.651	0.95%
	India	\$77,873,898.6695	0.93%
	Russia	\$70,604,855.0782	0.85%
	Italy	\$55,579,051.2011	0.67%
	Iran	\$52,360,462.133	0.63%
	Turkmenistan	\$51,536,297.1925	0.62%
	Syria	\$45,408,502.07	0.54%
	Pakistan	\$44,405,725.1295	0.53%
	Total	\$8,341,224,364.8324	100.00%

## Select

## Continent

## Country

- Asia
  - Europe
  - North America  
  - Armenia
  - Australia
  - Bhutan
  - Canada
  - China
  - Denmark
  - France
  - Germany
  - Greece
  - India

# DAX Measures

## Question 8 - Orders in Future Periods

### 1-Objective

I created a single-select Date slicer to allow the user to choose a specific day. Then I built calculations to count the number of orders scheduled for delivery within different future periods

### **2-The Dynamic Measure**

```
Total_Order = DISTINCTCOUNT(Sales[SalesKey])
```

```
Selected Date =  
SELECTEDVALUE('DateTable'[Date])
```

```
Orders Next Week =  
VAR SelectedDate =  
[Selected Date]  
RETURN  
CALCULATE(  
[Total_Order],  
FILTER (  
Sales,
```

```
Sales[Delivery_Date] > SelectedDate &&  
Sales[Delivery_Date] <= SelectedDate + 7  
)  
)
```

```
Orders Next 15 Days =  
VAR SelectedDate =  
[Selected Date]  
RETURN  
CALCULATE (  
[Total_Order],  
FILTER(
```

```
Sales,  
Sales[Delivery_Date] > SelectedDate &&  
Sales[Delivery_Date] <= SelectedDate + 15  
))
```

```
Orders Next 1 Month =  
VAR SelectedDate =  
[Selected Date]  
RETURN  
CALCULATE (  
[Total_Order],  
FILTER(
```

```
Sales,  
Sales[Delivery_Date] > SelectedDate &&  
Sales[Delivery_Date] <= SelectedDate + 30  
))
```

# DAX Measures

## Question 8 - Orders in Future Periods

### 2-The Dynamic Measure

```
Orders Next 6 Months =  
VAR SelectedDate =  
    [Selected Date]  
    RETURN  
    CALCULATE (  
        [Total_Order],  
        FILTER (  
            Sales,  
            Sales[Delivery_Date] > SelectedDate &&  
            Sales[Delivery_Date] <= SelectedDate + 180  
        )  
    )
```

```
Delivery Periods =  
DATABASE (  
    "Period", STRING,  
    "SortOrder", INTEGER,  
    {  
        { "1 Week", 1 },  
        { "15 Days", 2 },  
        { "1 Month", 3 },  
        { "6 Months", 4 }  
    }  
)
```

```
Orders by Period =  
VAR P = SELECTEDVALUE ( 'Delivery Periods'[Period] )  
    RETURN  
    SWITCH (  
        TRUE(),  
        P = "1 Week", [Orders Next Week],  
        P = "15 Days", [Orders Next 15 Days],  
        P = "1 Month", [Orders Next 1 Month],  
        P = "6 Months", [Orders Next 6 Months]  
    )
```

# DAX Measures

## Question 8 - Orders in Future Periods

### 3-Visual Validation

Date Slicer: A single-select date slicer allows the user to choose explain

Future Period Analysis: A column chart displays the number of orders scheduled for delivery within predefined future periods (1 Week, 15 Days, 1 Month, and 6 Months) from the selected date.

Dynamic Filtering: The visual updates automatically based on the selected date, enabling flexible time-based analysis of upcoming deliveries.



# DAX Measures

## Question 9 - Active Promotions Over Time

### 1-Objective

I created a calculation to count how many promotions are active in each month between 2011 and 2014.

The logic considers promotions that do not have an end date as still active.

### 2-The Dynamic Measure

```
Active Promotions =  
VAR StartOf_Month =  
    MIN ( 'DateTable'[Date] )  
VAR EndOf_Month =  
    MAX ( 'DateTable'[Date] )  
    RETURN  
    CALCULATE (  
        DISTINCTCOUNT ( Promotion[PromotionKey] ),  
        Promotion[StartDate] <= EndOf_Month,  
        OR (  
            ISBLANK ( Promotion[EndDate] ),  
            Promotion[EndDate] >= StartOf_Month  
        )  
    )
```

# DAX Measures

## Question 9 - Active Promotions Over Time

### 3-Visual Validation

Time-Based Analysis: A visual shows the number of promotions active in each month within the selected time range.

Open-Ended Promotions Handling: Promotions without an end date are treated as still active to ensure accurate counting.

Trend Visibility: This implementation helps track how promotional activity changes over time.



# DAX Measures

## Question 10 - KPI Performance and Targets

### 1-Objective

I selected 10 KPIs from the Q10\_Measure table, including:  
Average Order Value, Delivery Rate, Discount Rate, Profit YTD, Profit Margin, Return  
Rate, Sales YTD, Total Cost, Total Orders, Total Profit, and Total Revenue.

Then I calculated the same KPIs for the previous year to enable a year-over-year  
comparison and analyze performance changes over time.

After that, I added a Target value for each KPI (defined manually) so that I could  
compare the actual value with the target and determine whether each KPI achieved its  
goal.

Next, I created a condition for each KPI so that if the Actual value is greater than or  
equal to the Target, the result returns 1, otherwise 0.

This approach allowed me to count how many KPIs achieved their targets and how  
many did not, and display them in Over Achieving and Under Achieving indicators.

# DAX Measures

## Question 10 - KPI Performance and Targets

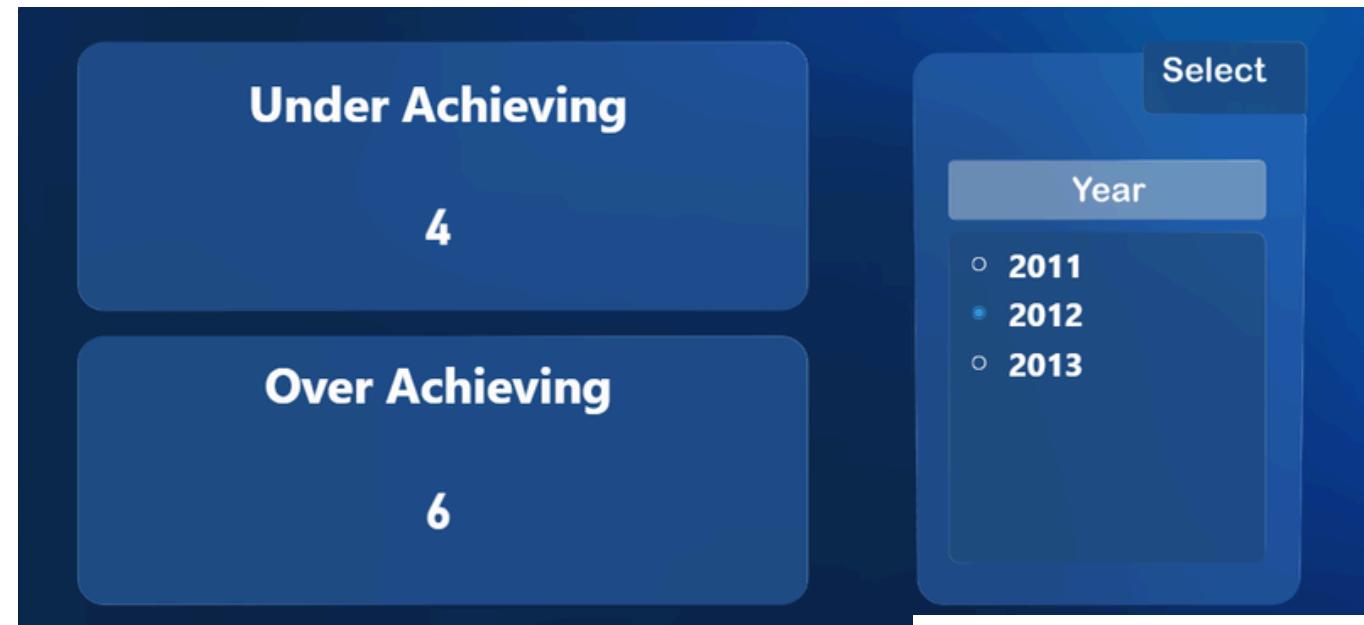
### 2-Visual Validation

Dynamic KPI Selection: A KPI structure was created to display multiple KPIs in a unified and flexible format.

Year-over-Year Comparison: The same KPIs were calculated for the previous year to support performance comparison.

Target Evaluation: A manually defined target was added for each KPI to measure performance against goals.

Achievement Tracking: A logical condition returns 1 when a KPI meets its target and 0 otherwise, allowing counting of achieved vs. non-achieved KPIs for performance monitoring.



KPI Name	KPI Actual	KPI Target	%
AOV	3,654.45	3,837.18	95.2%
Margin %	0.57	0.70	82.0%
Orders	2,282,482.00	2,510,730.20	90.9%
Profit	4,788,816,884.44	5,746,580,261.33	83.3%
Revenue	8,341,224,364.83	9,592,408,019.56	87.0%
Total			

KPI Name	KPI Actual	KPI Target	%
Delivery Rate	1.00	0.90	111.1%
Discount Rate	0.02	0.15	10.1%
Return Rate	0.01	0.05	23.3%
Total			

# DAX Scripting & Logic:

## Question 11

### Total Impression :

Total Impressions = SUM(Marketing[Clicks])

### Current Impression :

```
Selected Impressions =
VAR LastDataDate = CALCULATE(MAX('Marketing'[Date]), REMOVEFILTERS())

RETURN
SWITCH(TRUE(),
ISFILTERED('Period 2 Dates'[DateKey]),
CALCULATE([Total Impressions],
TREATAS(VALUES('Period 2 Dates'[DateKey]), 'New Date Table'[Date])
),

ISFILTERED('TimeRanges'[Range]),
VAR Days = SELECTEDVALUE('TimeRanges'[Days], 30)
RETURN CALCULATE([Total Impressions],
DATESBETWEEN('New Date Table'[Date], LastDataDate - Days, LastDataDate)
),

CALCULATE([Total Impressions],
DATESYTD('New Date Table'[Date]),
'New Date Table'[Date] <= LastDataDate
)
)
```

# DAX Scripting & Logic:

## Question 11

To Change The Arrow Direction :

```
Arrow Direction =  
IF([Impression Change %] > 0, 1,  
 IF([Impression Change %] < 0, -1, 0)  
)
```

To Change The Arrow Direction :

```
Arrow Icon =  
IF([Arrow Direction] = 1, "▲",  
 IF([Arrow Direction] = -1, "▼", "-"))  
)
```

Previous Year Impression :

```
Reference Impressions =  
// 1. Find the last date in your data (e.g. Dec 31, 2014)  
VAR LastDataDate = CALCULATE(MAX('Marketing'[Date]), REMOVEFILTERS())  
  
// 2. Calculate "Same Day Last Year" safely using EDATE  
VAR LastDataDateLY = EDATE(LastDataDate, -12)  
  
VAR RangeSelect = SELECTEDVALUE('TimeRanges'[Days], 30)  
  
RETURN  
SWITCH(TRUE(),  
 // Manual Mode  
 ISFILTERED('Period 2 Dates'[DateKey]),  
 CALCULATE([Total Impressions],  
 TREATAS(VALUES('Period 1 Dates'[DateKey]), 'New Date Table'[Date]))  
,  
  
// Time Range Mode  
ISFILTERED('TimeRanges'[Range]),  
 CALCULATE([Total Impressions],  
 SAMEPERIODLASTYEAR(  
 DATESBETWEEN('New Date Table'[Date], LastDataDate - RangeSelect, LastDataDate))  
,  
,  
  
// Default YTD (Previous Year)  
CALCULATE([Total Impressions],  
 DATESYTD(DATEADD('New Date Table'[Date], -1, YEAR)),  
 'New Date Table'[Date] <= LastDataDateLY  
)  
)
```

## Visualizations & Insights:

- Metrics Displayed: [Impressions YTD], [Impressions LY], [Change %].
  - The "Smart" Element: A dynamic Directional Arrow that changes color (Green/Red) based on performance.
- . Impressions Trend Over Time (Dynamic Line Chart)
- Visual: Line Chart with Zoom Slider.
  - X-Axis: Date (or Month).
  - Y-Axis: [Selected Impressions] vs [Reference Impressions].

**Range**

- Last 15 Days
- Last 30 Days
- Last 45 Days

[YTD View](#)[Time Range View](#)[Manual View](#)**18.36M**

Total Impression

**654.25K**

Previous Year impression

**687.85K**

Current Impression

**5.14%**

Impression Change %



# Dashboard 1: C-Level Executive View (Strategic)

**Target Audience:** CEO, CFO, and Board Members.

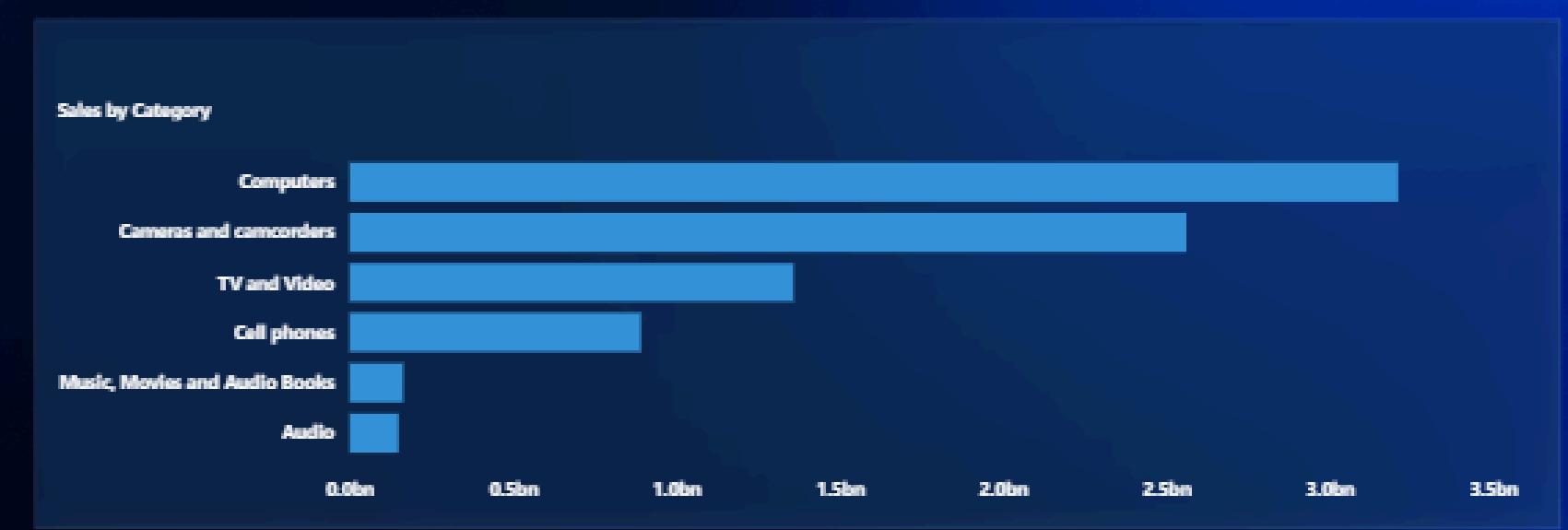
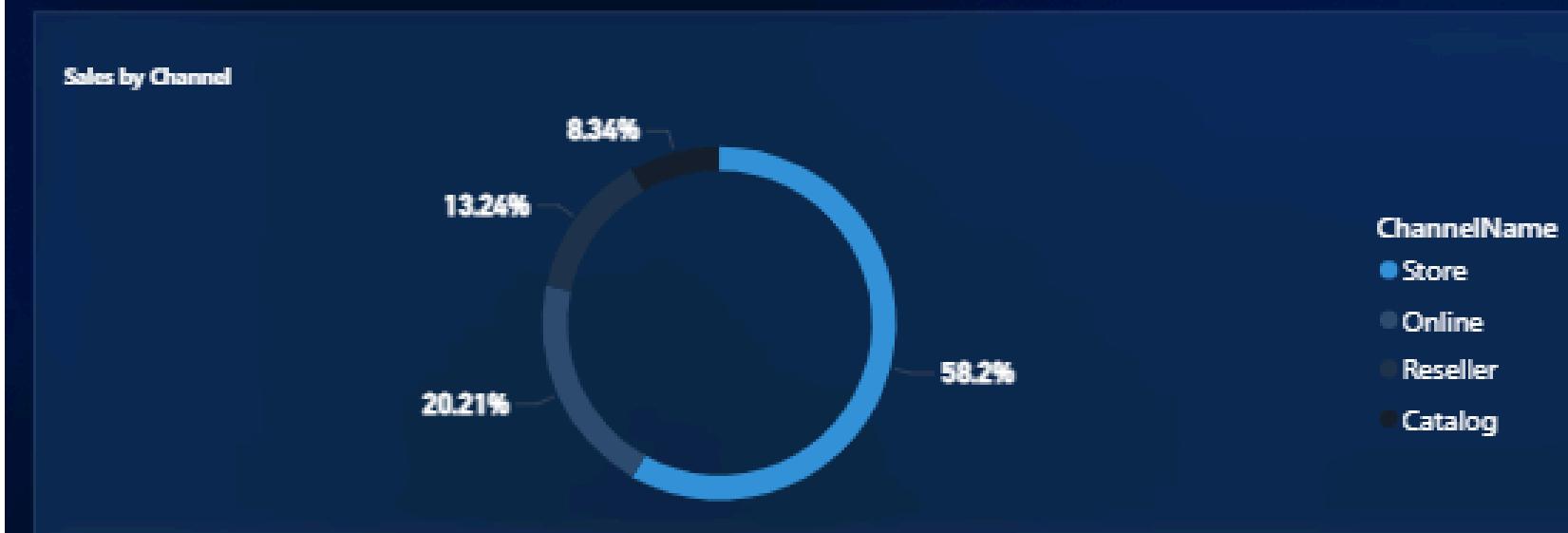
**Focus:** Profitability, yearly trends, and high-level business growth. This dashboard supports strategic decision-making by providing an overview of financial performance, revenue sustainability, and market expansion.

## Key Performance Indicators (KPIs)

- **Total Revenue:** Represents total sales value and indicates overall market performance.
- **Net Profit:** Measures earnings after deducting all operational and product costs.
- **Profit Margin %:** Shows how efficiently revenue converts into profit.
- **Total Orders:** Indicates total number of transactions and customer demand level

Year: All

Country: All



# Visualizations & Insights:

- **Revenue Trend (Line Chart):** Monthly performance tracking across years with historical comparison.
- **Sales by Country (Map Chart):** Geographic distribution of revenue with drill-through analysis.
- **Top 5 Subcategories (Dynamic Bar Chart):** Best-performing product subcategories based on filters.
- **Revenue by Channel (Donut Chart):** Revenue mix between Store, Online, and Reseller channels.

# DAX Scripting & Logic:

## 1. Strategic Dashboard Measures (C-Level)

Core financial metrics used for high-level analysis.

### Total Revenue

Description: Calculates the sum of sales amounts from the transaction table.

Total Revenue =  $\text{SUM}(\text{Sales}[\text{SalesAmount}])$

### Net Profit

Description: Derived by subtracting total costs from total revenue.

Net Profit =  $[\text{Total Revenue}] - \text{SUM}(\text{Sales}[\text{TotalCost}])$

### Profit Margin %

Description: The percentage of revenue that turns into profit.

Profit Margin % =  $\text{DIVIDE}([\text{Net Profit}], [\text{Total Revenue}], 0)$

### Total Orders

Description: Counts the distinct number of orders (or transactions).

Total Orders =  $\text{DISTINCTCOUNT}(\text{Sales}[\text{OrderNumber}])$

# Dashboard 2: Sales Team View (Operational)

**Target Audience:** Sales Managers and Regional Heads.

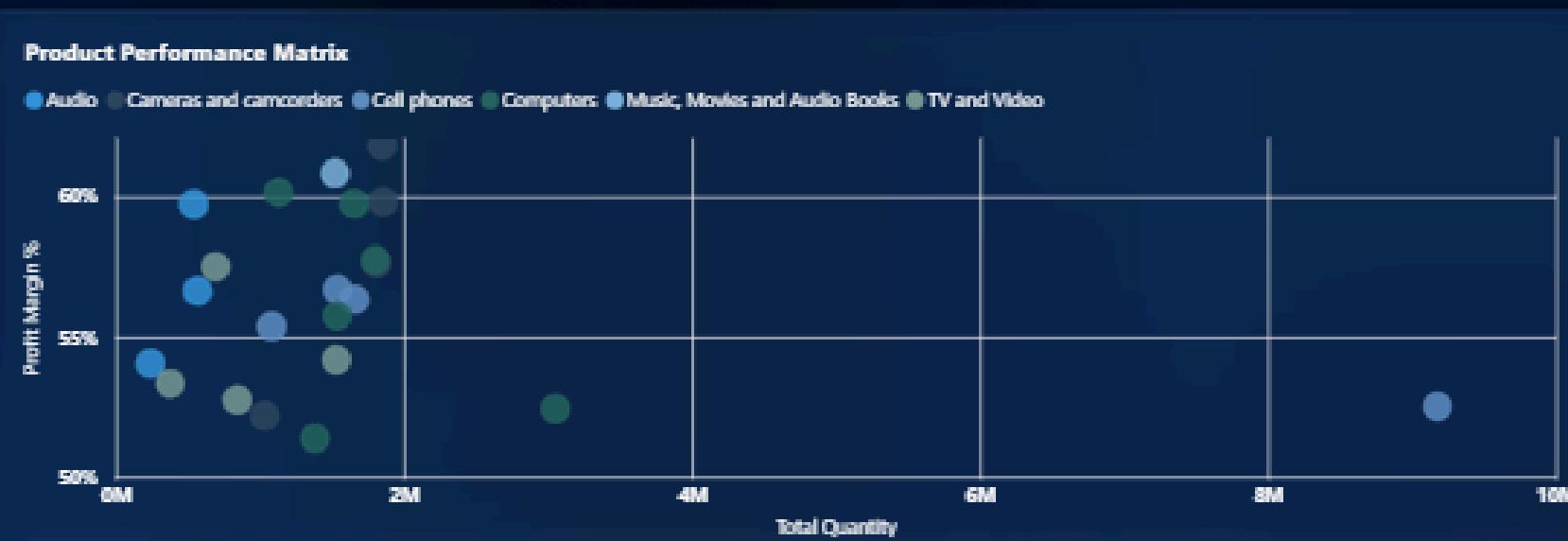
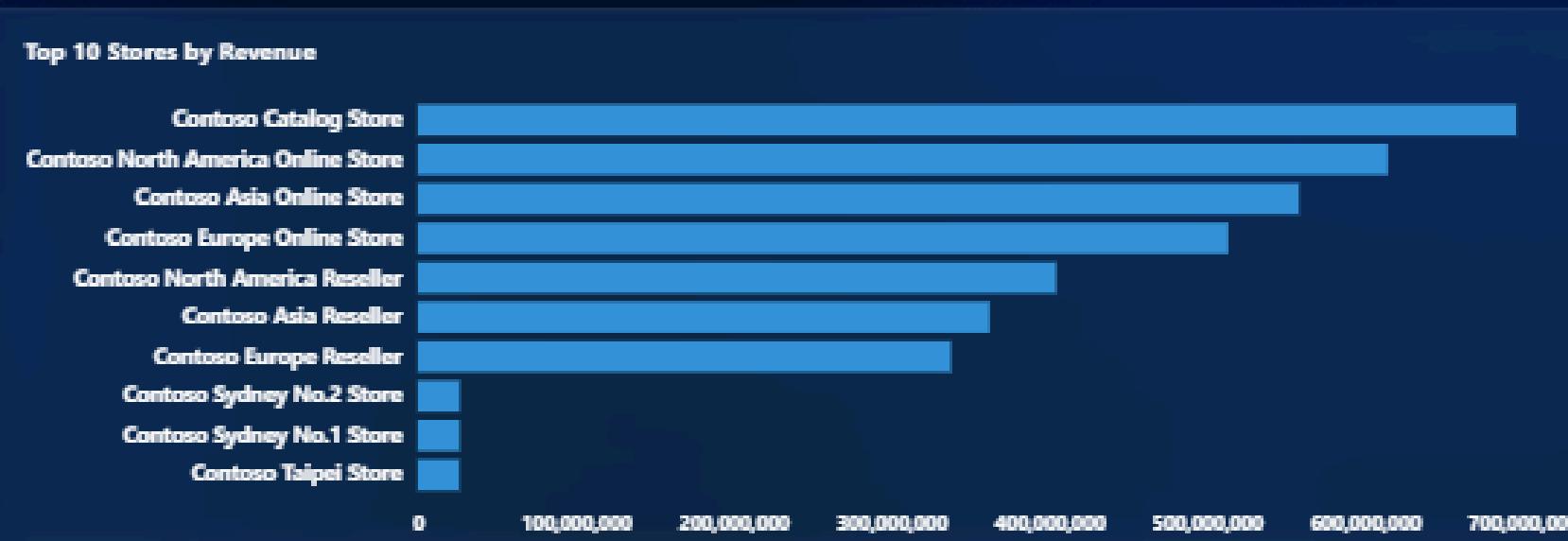
**Focus:** Sales volume, operational efficiency, returns monitoring, and store performance.

## Key Performance Indicators (KPIs)

- Sales Quantity: Total units sold reflecting product demand.
- Return Rate %: Indicates product quality and customer satisfaction level.
- Average Order Value (AOV): Average revenue per order.
- Discount %: Measures reliance on discounts to drive sales.

Year: All

StoreName: All



Product Details

ProductName	ProductCategory	Total Quantity	Total Returns	Return Rate %
Contoso Phone Tough Skin Case E140 White	Cell phones	120600	91	0.03%
Contoso Phone Tough Skin Case E140 Pink	Cell phones	121960	93	0.03%
Contoso Phone Tough Skin Case E140 Silver	Cell phones	108600	90	0.03%
Contoso Phone Tough Skin Case E140 Black	Cell phones	108120	92	0.03%
Headphone Adapter for Contoso Phone E130 Black	Cell phones	282140	116	0.04%
Headphone Adapter for Contoso Phone E130 White	Cell phones	241480	101	0.04%
Headphone Adapter for Contoso Phone E130 Silver	Cell phones	265800	112	0.04%
Contoso Touch Stylus Pen E150 Black	Cell phones	356340	172	0.05%
Contoso Touch Stylus Pen E150 White	Cell phones	349780	190	0.05%
Contoso Touch Stylus Pen E150 Red	Cell phones	343040	189	0.06%
Contoso Original K1m Li-Ion Standard Battery E170 White	Cell phones	370320	208	0.06%
Total		3690410	326208	0.88%

# Visualizations & Insights:

- **Top 10 Stores (Bar Chart):** Ranking stores by revenue.
- **Returns by Category (Bar Chart):** Highlights high-return categories using conditional formatting.
- **Product Performance Matrix (Scatter Chart):** Volume vs profitability classification.

**Detailed Product Table:** Granular operational data with visual indicators.

# DAX Scripting & Logic:

## 2. Operational Dashboard Measures (Sales Team)

Operational metrics focusing on volume, efficiency, and quality control.

### Sales Quantity

Description: Total number of units sold.

Total Quantity =  $\text{SUM}(\text{Sales}[\text{SalesQuantity}])$

### Return Rate %

Description: The percentage of items returned relative to total items sold. Used for quality alerts.

Total Returns =  $\text{SUM}(\text{Sales}[\text{ReturnQuantity}])$

Return Rate % =  $\text{DIVIDE}([\text{Total Returns}], [\text{Total Quantity}], 0)$

### Average Order Value (AOV)

Description: Average revenue generated per single order.

AOV =  $\text{DIVIDE}([\text{Total Revenue}], [\text{Total Orders}], 0)$

### Discount %

Description: The ratio of discount given vs. the original price (Gross Sales).

Total Discount =  $\text{SUM}(\text{Sales}[\text{DiscountAmount}])$

Discount % =  $\text{DIVIDE}([\text{Total Discount}], ([\text{Total Revenue}] + [\text{Total Discount}]), 0)$

# Final Conclusion & Acknowledgments

## 1-Strategic Impact

This Sales Decision Support System (DSS) is more than just a collection of charts; it is a comprehensive tool designed for data-driven growth. Through this project, we have successfully implemented:"

- Operational Clarity: Bridging the gap between orders and fulfillment.
- Advanced Governance: Securing data through Dynamic RLS logic.
- Technical Efficiency: Reducing maintenance by 80% using Calculation Groups.
- Strategic Benchmarking: Providing a 3-year historical context for every metric.

## 2-Lessons Learned

Working on this project has deepened our expertise in advanced Business Intelligence workflows, specifically in Snowflake schema optimization, complex DAX context manipulation (CALCULATE & REMOVEFILTERS), and user-centric dashboard design."

# Thank You!

