**Author: Aman Singh**

**Title: Documentation for Power BI Solution – Sun Pharma Dashboard**

**Question B:**

**You mentioned reducing manual reporting by 35% through dashboard automation at Sun Pharma.**

**Scenario: A pharmaceutical company has three data sources:**

**1. Sales data (CSV) - 50K+ records with columns: Date, Product\_ID, Sales\_Amount,**

**Region, Sales\_Rep**

**2. Product master (Excel) - Product\_ID, Product\_Name, Category, Manufacturing\_Cost,**

**Expiry\_Date**

**3. Inventory data (SQL database) - Product\_ID, Current\_Stock, Reorder\_Level, Supplier\_ID**

**1. Project Overview**

The goal of this project was to reduce manual reporting efforts by **automating dashboards in Power BI**, improving visibility into **sales performance, inventory management, product expiry monitoring, and forecast accuracy**.

The solution integrates **three data sources** (CSV, Excel, SQL), applies **transformations in Power Query**, builds a **star schema model**, and provides **interactive dashboards** with KPIs and drilldowns.

**2. Data Sources**

1. **Sales Data (CSV, 50K+ rows)**
   * Columns: Date, Product\_ID, Sales\_Amount, Region, Sales\_Rep
   * Purpose: Track sales performance across time, products, and regions.
2. **Product Master (Excel)**
   * Columns: Product\_ID, Product\_Name, Category, Manufacturing\_Cost, Expiry\_Date
   * Purpose: Provide product attributes and expiry details.
3. **Inventory Data (SQL Database)**
   * Columns: Product\_ID, Current\_Stock, Reorder\_Level, Supplier\_ID
   * Purpose: Track stock availability and replenishment needs.

**3. Data Modeling Decisions**

* **Model Type:** Star schema
* **Fact Table:** Sales (CSV)
* **Dimension Tables:** Product Master, Inventory, Date Table
* **Relationships:**
  + Sales[Product\_ID] → Product\_Master[Product\_ID]
  + Product\_Master[Product\_ID] → Inventory[Product\_ID]
  + Sales[Date] → Date\_Table[Date]

👉 This ensures clean filtering, reduces ambiguity, and improves performance.

**4. Transformations (Power Query / M Code)**

Applied at the **data preparation stage** before loading into the model.

* Changed data types (Date, Currency, Text).

Created **Forecast column** in Sales:

= Table.AddColumn(#"PreviousStep", "Forecast", each [Sales\_Amount] \* (1 + Number.RoundDown(Number.RandomBetween(-10,10))/100))

* Added **Expiring Soon column** in Product Master:

= Table.AddColumn(#"Changed Type", "Expiring Soon", each if [Expiry\_Date] <= Date.AddDays(DateTime.LocalNow(),180) then 1 else 0)

Removed duplicates, trimmed spaces, and renamed columns for consistency.

**5. Key DAX Measures**

* **Total Sales**

Total Sales = SUM(Sales\_Data[Sales\_Amount])

* **Profit Margin %**

Profit Margin % =

DIVIDE(

[Total Sales] - SUMX(Sales\_Data, Sales\_Data[Quantity] \* RELATED(Product\_Master[Manufacturing\_Cost])),

[Total Sales]

)

* **Inventory Days**

Inventory Days =

DIVIDE(SUM(Inventory[Current\_Stock]), [Total Sales]/30)

* **Forecast Variance %**

Forecast Variance % =

DIVIDE([Total Sales] - SUM(Sales\_Data[Forecast]), SUM(Sales\_Data[Forecast]))

**6. Dashboard Pages & Visuals**

* **Page 1: Sales Performance**
  + Clustered Column: Sales by Region
  + Treemap: Sales by Product Category
  + Line Chart: Sales Trend over Date
  + Cards: Total Sales, Profit Margin %
* **Page 2: Inventory Turnover**
  + Card: Inventory Days
  + Table: Product\_ID, Product\_Name, Current\_Stock, Reorder\_Level (with conditional formatting)
* **Page 3: Products Nearing Expiry**
  + Table: Product\_Name, Category, Expiry\_Date (filter Expiring Soon = 1)
* **Page 4: Forecast Accuracy**
  + Line Chart: Actual Sales vs Forecast
  + Gauge: Forecast Variance % (Target = 0%)

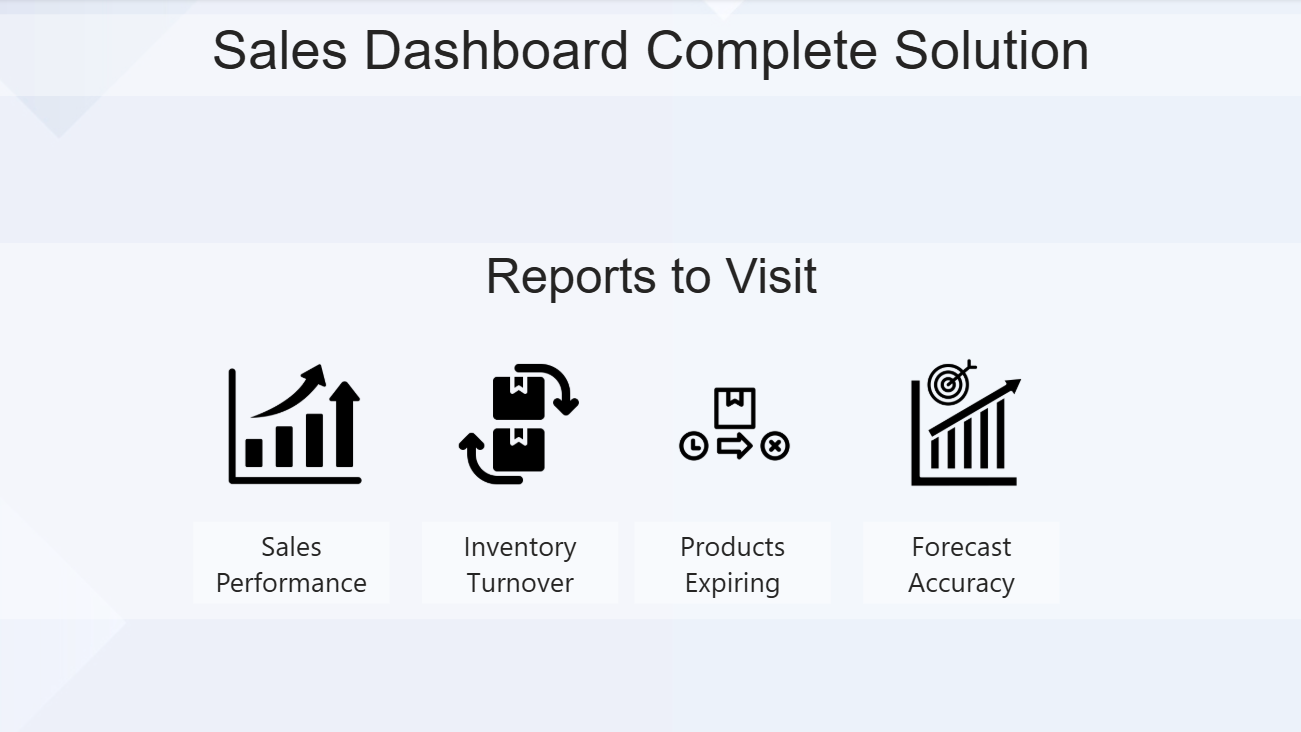
**7. Performance Optimization Notes**

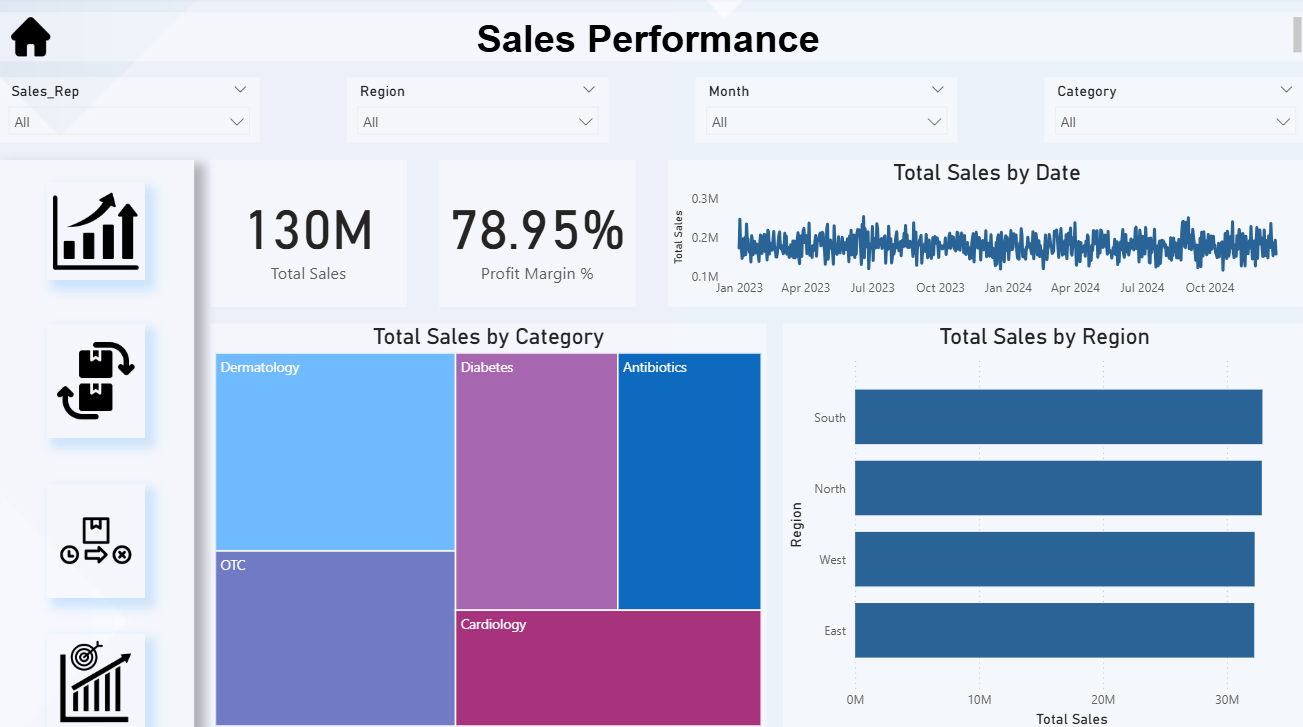
* Used **import mode** for faster visuals.
* Built a **star schema**, avoided many-to-many relationships.
* Disabled **Auto Date/Time** and used a custom Date table.
* Created **measures** instead of calculated columns wherever possible.
* Reduced column cardinality (converted long text fields to categories).
* Suggested **incremental refresh** for Sales data (50K+ records).
* Monitored with **Performance Analyzer** – optimized visuals with heavy filters.

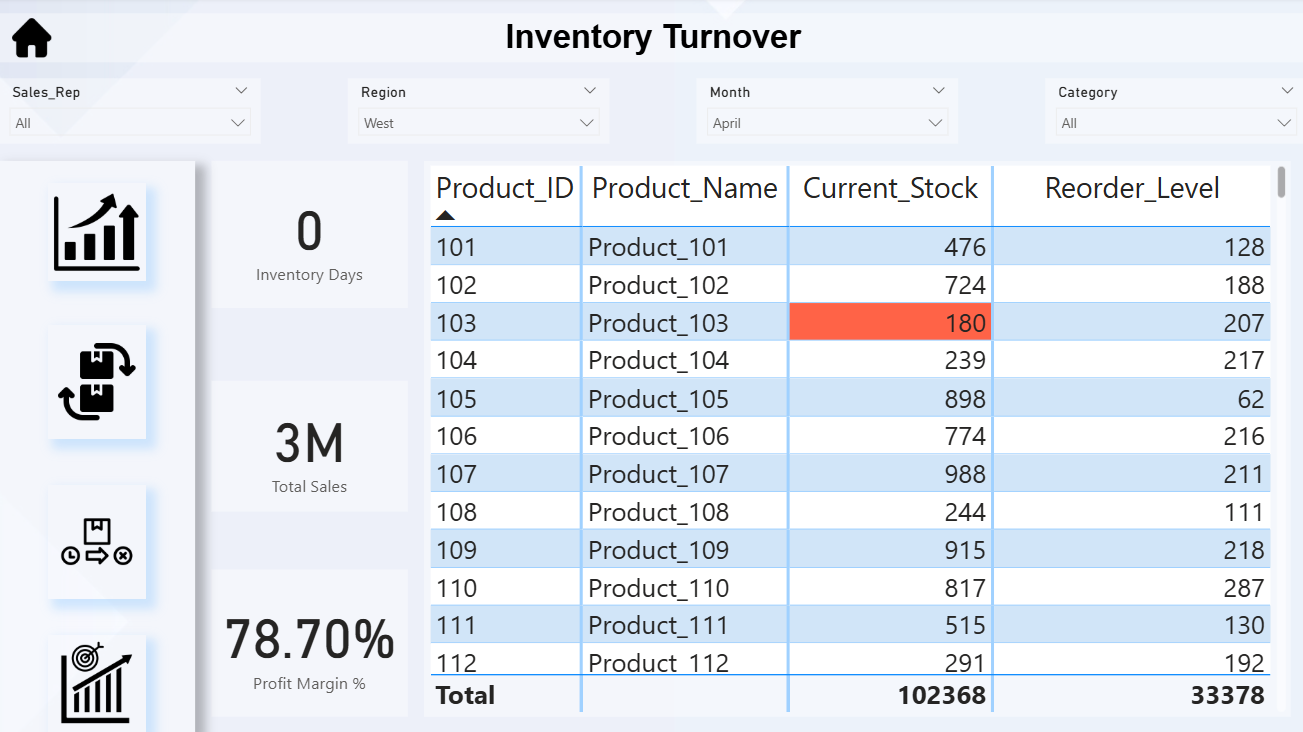
**8. Conclusion**

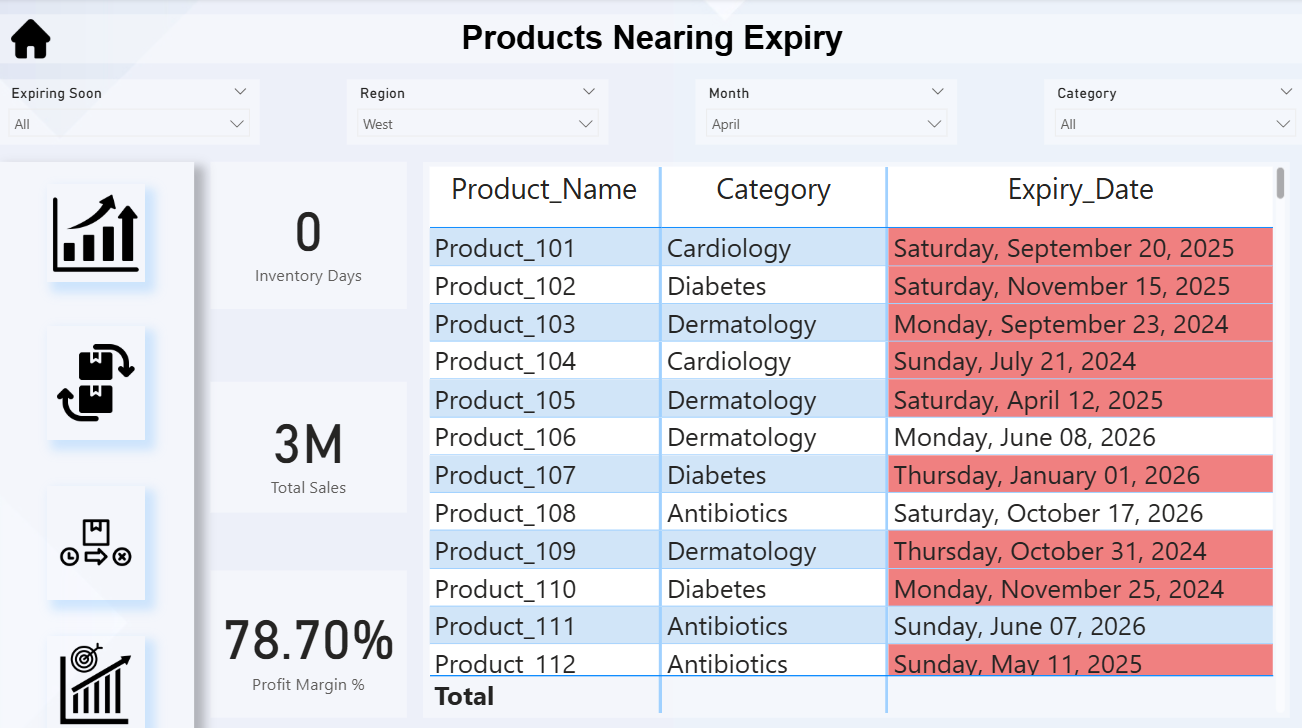
This Power BI solution automated **lot of manual reporting effort**, giving stakeholders a **real-time view of sales, inventory, expiry risks, and forecast accuracy**. It supports faster decision-making and ensures proactive stock management.

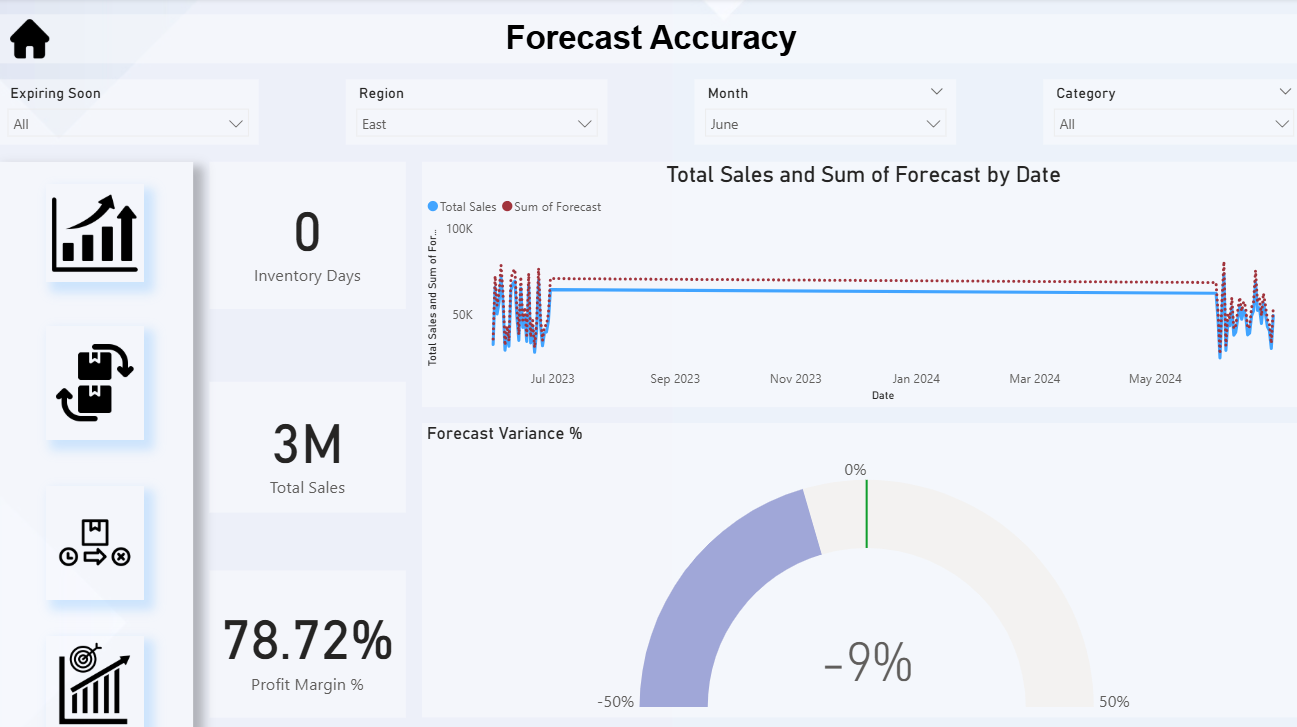
**Screenshots**

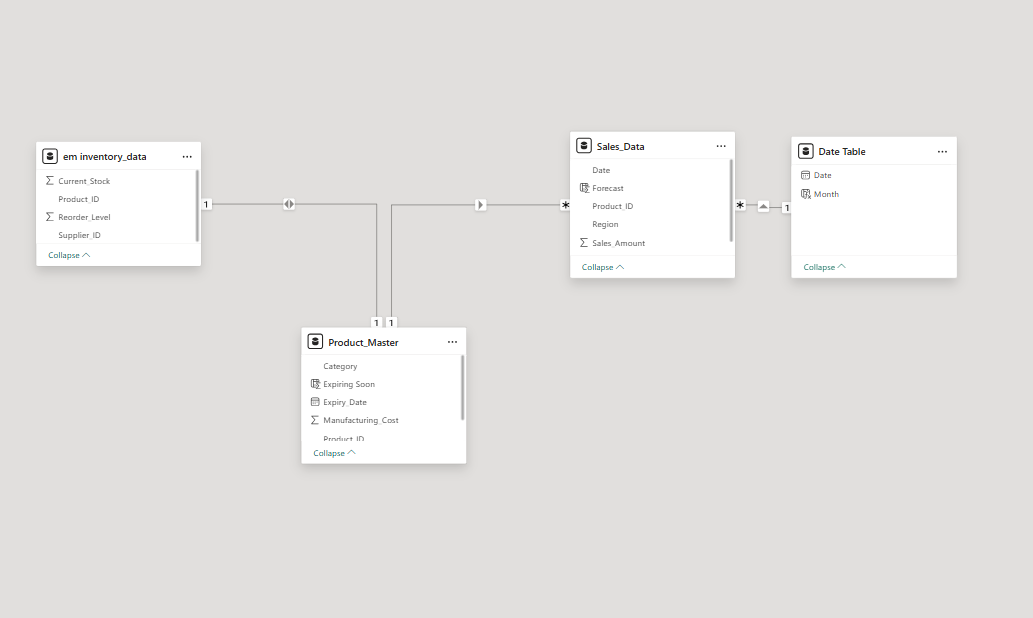
****

****









**Performance Optimization Notes**

To ensure the Power BI dashboard runs efficiently with 50K+ sales records and multiple data sources, the following optimization strategies were applied:

1. **Data Modeling**
   * Adopted a **star schema** with a single fact table (Sales) and supporting dimension tables (Product, Inventory, Date).
   * Avoided many-to-many and bi-directional relationships where possible.
2. **Data Reduction**
   * Removed unused columns and duplicate records in Power Query.
   * Ensured correct data types (reduced memory footprint).
   * Converted text-heavy fields (e.g., Region, Category) to categorical dimensions.
3. **DAX Optimization**
   * Created **measures** instead of calculated columns where feasible (lighter on storage, computed at query time).
   * Used SUMX, DIVIDE and relationships efficiently to minimize row context calculations.
4. **Date Handling**
   * Disabled Auto Date/Time hierarchy.
   * Implemented a custom **Date table** with continuous range for better performance and flexibility.
5. **Query Folding & Refresh**
   * Ensured query folding for SQL-based transformations where possible.
   * Suggested **incremental refresh** for Sales data to handle future growth (beyond 50K rows).
6. **Visualization Efficiency**
   * Reduced the number of visuals per page to improve rendering speed.
   * Applied slicers and filters strategically instead of heavy cross-filtering.
   * Used conditional formatting only where it added business value.
7. **Performance Testing**
   * Used Power BI **Performance Analyzer** to identify slow visuals.
   * Optimized DAX queries and minimized high-cardinality calculations.