**Report: Supply Chain and Inventory Management**

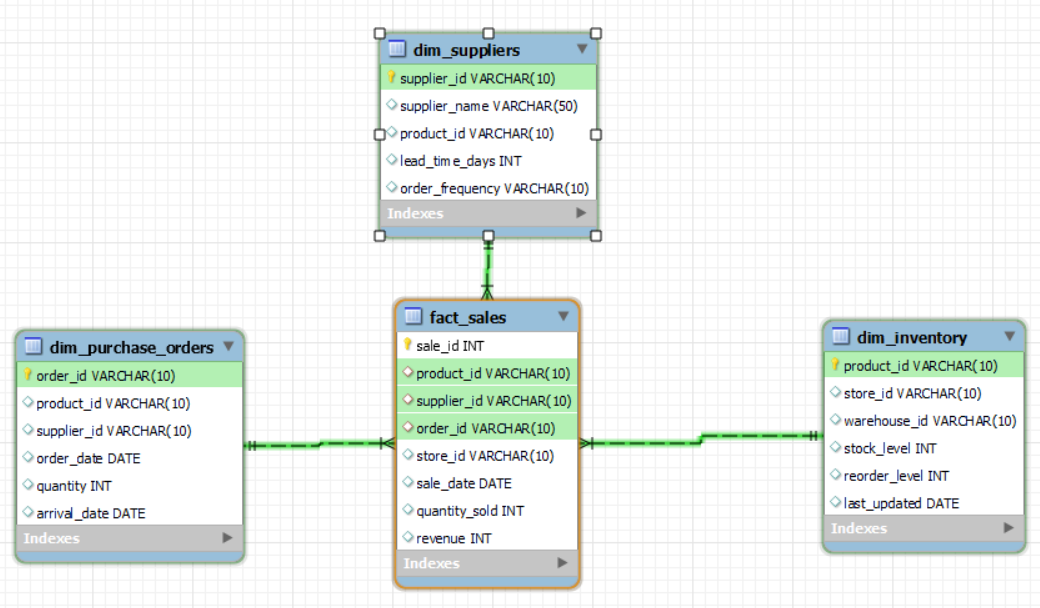
**Background:**

ABC Supermarket is a retail chain with multiple stores across different cities. The company has been struggling with stock imbalances—some products run out too quickly (stockouts), while others remain unsold for months (overstocking). These inefficiencies are increasing costs and reducing profits.

**Work Flow**

**Database Creation**:

* Created a database ‘supply\_chain’ in MYSQL Workbench.
* **Star Schema Design –** Build a warehouse schema to optimize queries for inventory and supply chain management.
* I have created tables in the database using ‘star schema’ design
* I created total 4 tables – 3 dimension tables and 1 fact table.

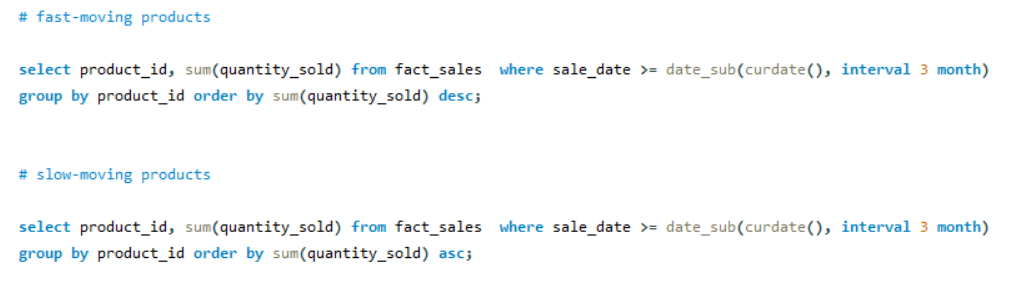


* Note: To connect all dimension tables to the fact table I have inserted the data of order id and supplier id data to fact table.
* dim\_inventory, dim\_suppliers, dim\_purchase\_orders and fact\_sales are the tables.
* For inserting the data I have made some assumptions.

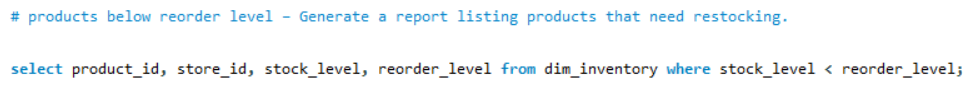
1. Products – 10
2. Stores – 5
3. Warehouses – 3
4. Suppliers – 10

**SQL Tasks (Inventory Analytics & Reporting)**

* Identify slow-moving and fast-moving products – Query products with the highest and lowest sales in the past 3 months.

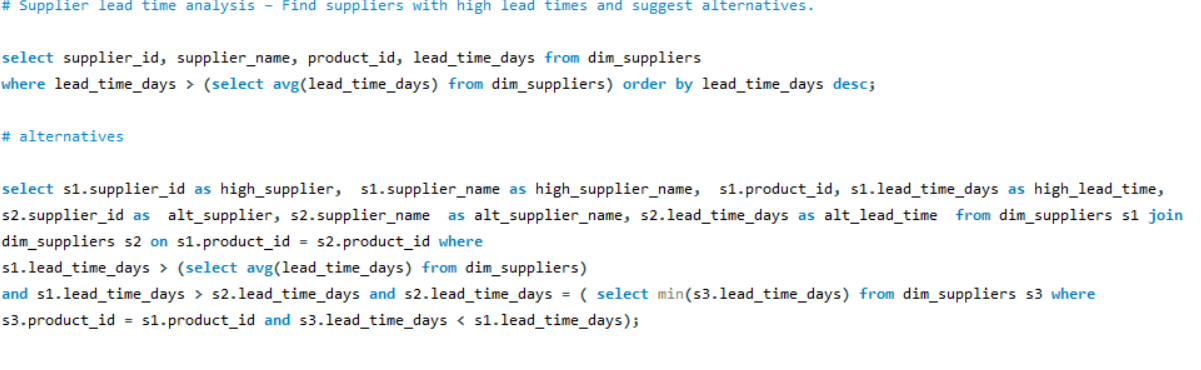


* Find products below reorder level – Generate a report listing products that need restocking



The products whose stock level is less than reorder level has to be restocked.

* Supplier lead time analysis – Find suppliers with high lead times and suggest alternatives.

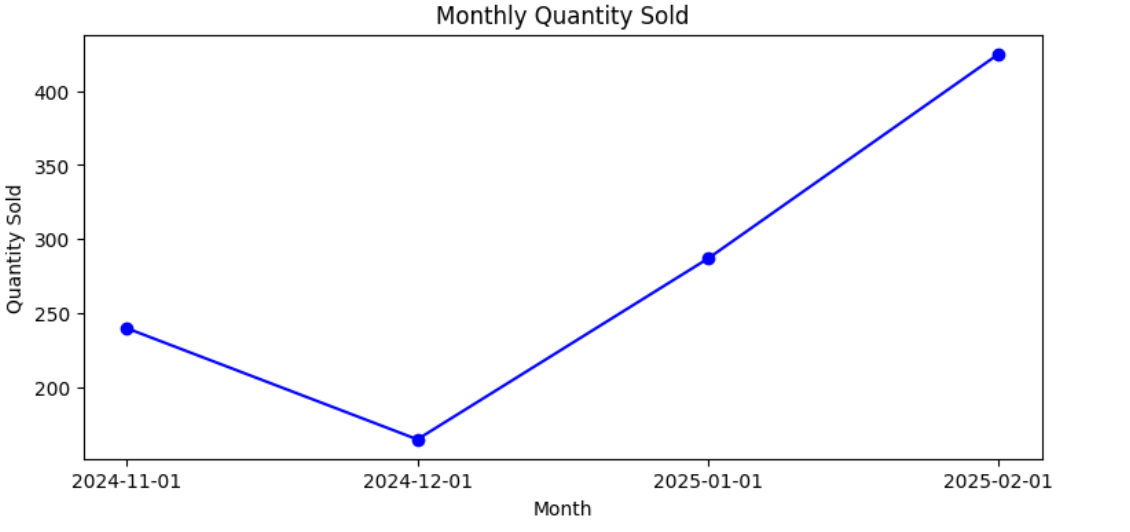


For high lead time I considered the suppliers whose lead time is more than the average lead time. For alternative I have considered the suppliers who sell the same products of high lead time suppliers but have lead time less than average lead time.

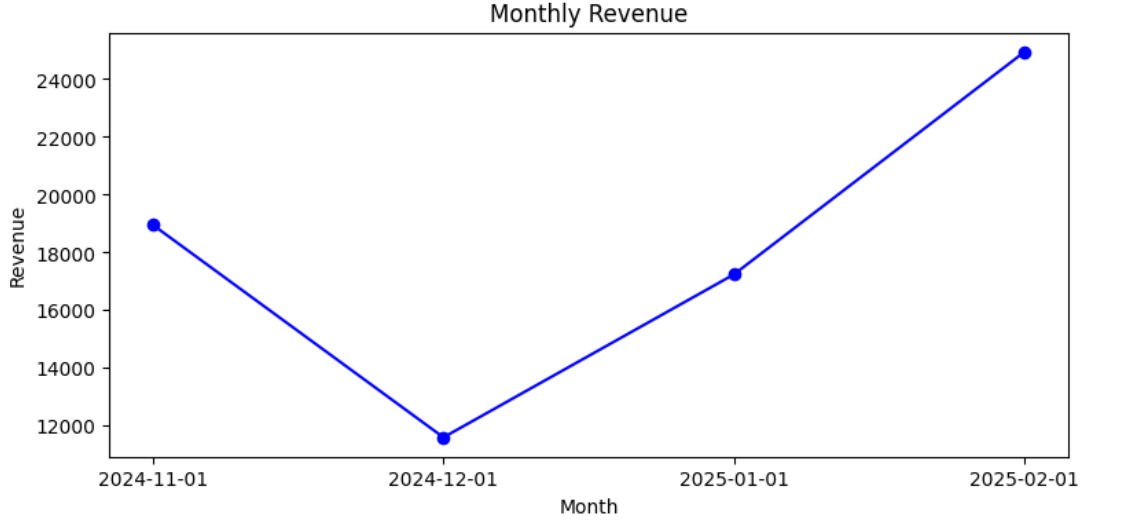
* I have saved the sql script and all the sql tasks performed. The file ‘rushitha\_supply\_chain.sql’ is attached in the folder.

**Python Tasks (Forecasting & Optimization):**

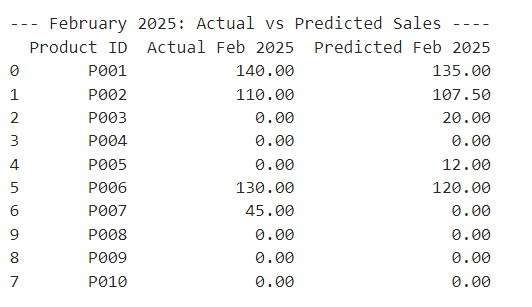
* Now I have connected the python and database tables using sqlalchemy library using engine.
* Load the database tables into Dataframes using pandas.
* Dispose the engine to close database connection.
* Consider the sales dataframe.
* Year and Month are extracted from the sale date.
* Monthly quantity sold trend is analysed using mathplotlib library.



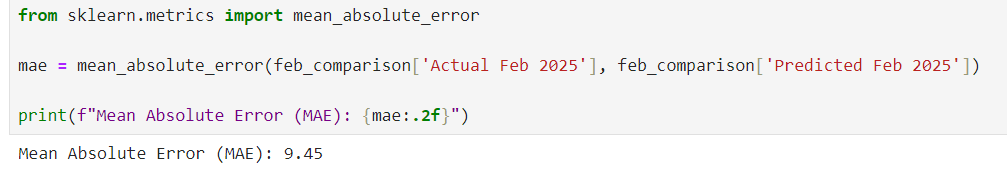
* Monthly revenue trend is analysed using mathplotlib library.



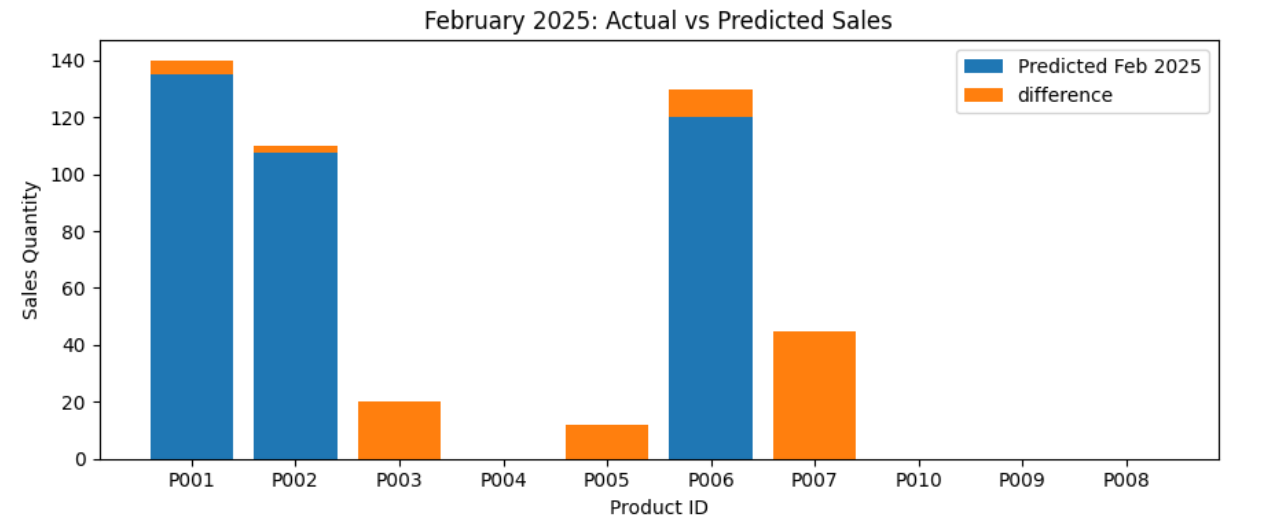
* Findings: Highest quantity & revenue generated in FEB 2025 where as the Lowest in DEC 2024.
* **Demand Forecasting:**
* Use Python (Pandas, NumPy, Matplotlib, Scikit-L) to perform time-series analysis and to predict demand for the next month.
* The data is from NOV 2024 – FEB 2025. I have divided the data into train and test data.
* Train data - sales data for the period from Nov 2024 to Jan 2025.
* Test data - sales data for the period of Feb 2025.
* Pivot table of monthly sales for each product.
* **Trend Calculation** : Month-to-month differences in sales for each product. If there are at least 3 months of data, the trend is calculated as the average of month-to-month differences of train data.
* If not, the trend is assumed to be 0.
* Predicts Feb sales as Jan sales + predicted trend.
* Compare actual February sales with predicted February sales.



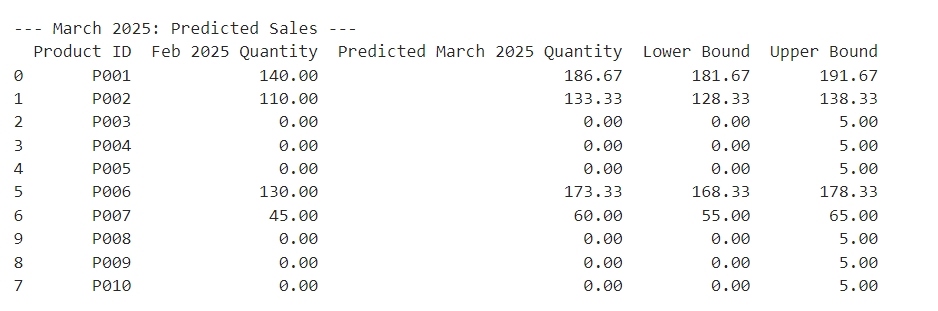
* Findings: The Actual Feb 2025 values very close to Predicted Feb 2025 values.
* **Model Evaluation**: Calculate Mean Absolute Error (MAE) for February predictions using mean\_absolute\_error from sklearn library.



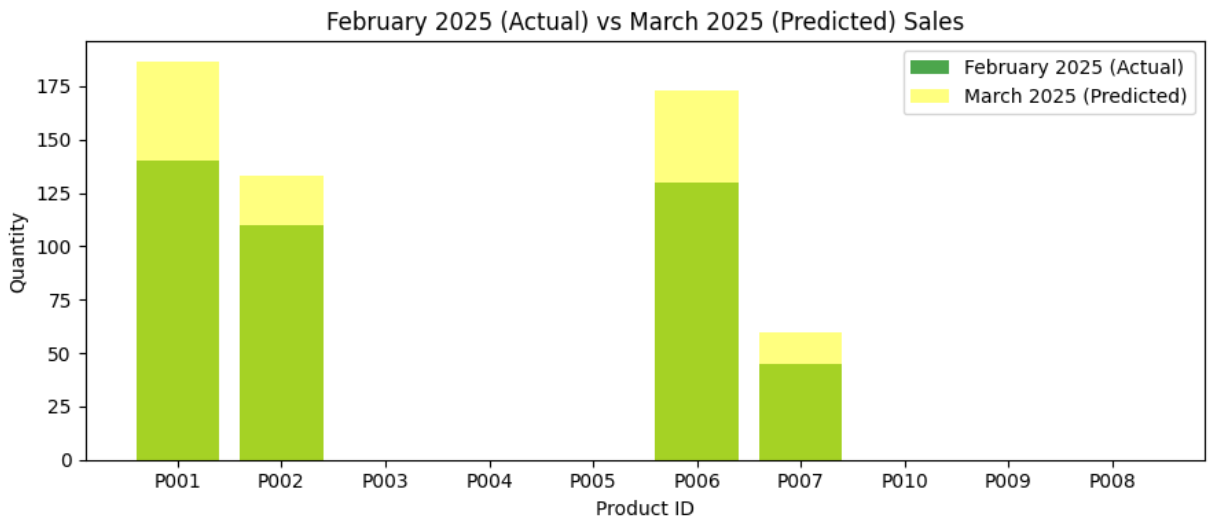
* Findings: 9.45 is considered as good value for accuracy which indicates that the predictions are correct.



* Findings: From above stacked bar graph we can spot the difference is very minimal.
* To handle this difference for next month prediction which is March 2025 we are going to calculate the “**lower bound and upper bound values**” to the prediction to give a range in which the actual value will reside.
* Predicting March 2025 Sales: For each product uses the Feb 2025 sales as the base.



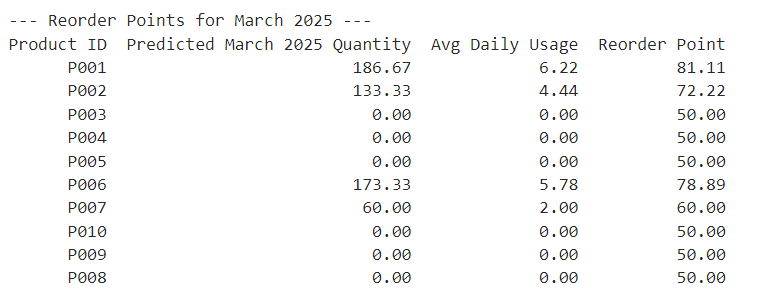
* Findings: Shows predicted sales for March 2025, along with lower and upper bounds gives a good range for prediction.



* Findings: Compares actual February sales with predicted March sales.
* **Reorder Point Calculation:**

Implement a formula to calculate the optimal reorder point for each product

* I calculated reorder point for March 2025 prediction.
* lead\_time: The time (in days) it takes to receive new stock after placing an order.
* extra\_stock: Safety stock added to account for variability in demand or delays.
* Assumed: lead time as 5 days as it is the average of the lead time of data (this is a finding from SQL query). Extra stock is assumed as 50.
* Average daily usage of each product, calculated as Predicted March 2025 Quantity / 30.
* Reorder Point: The inventory level at which a new order should be placed, calculated as (Avg Daily Usage \* Lead Time) + Extra Stock



* Findings: reorder point is calculated for each product.
* **Supplier Performance Analysis:**

Use clustering (K-Means or Hierarchical) to classify suppliers based on lead time and order frequency.

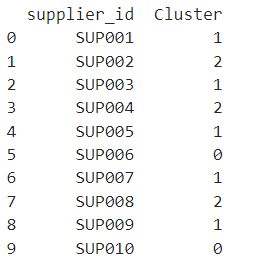
* Consider supplier dataframe.



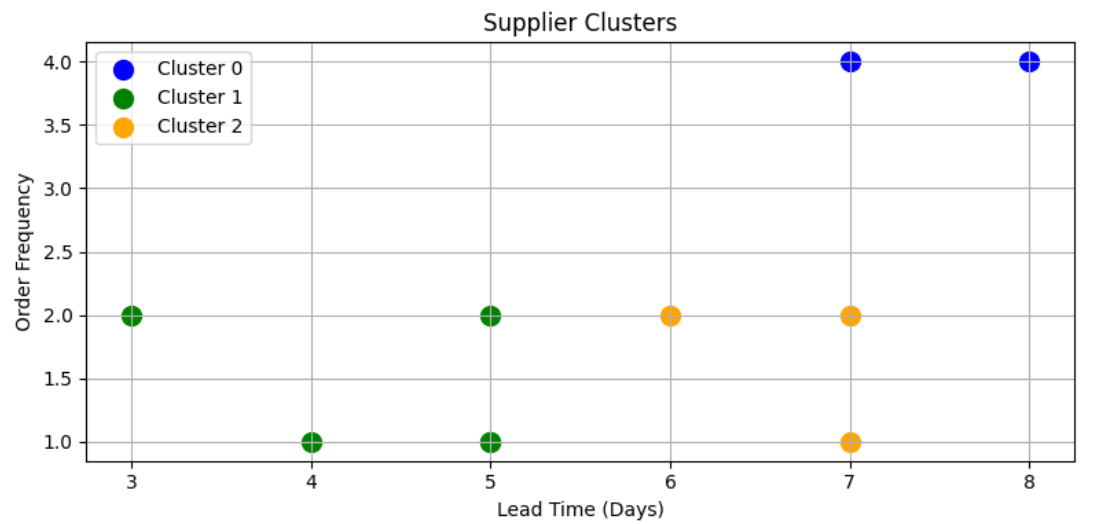
* Findings: order frequency is a text data to implement kmeans algorithm we need to convert text to numerical.
* Maps order frequency "Weekly": 1, "Biweekly": 2, "Monthly": 4.



* Findings: Mapping is done correctly.
* Using Kmeans from sklearn library cluster the suppliers based on lead time and order frequency.



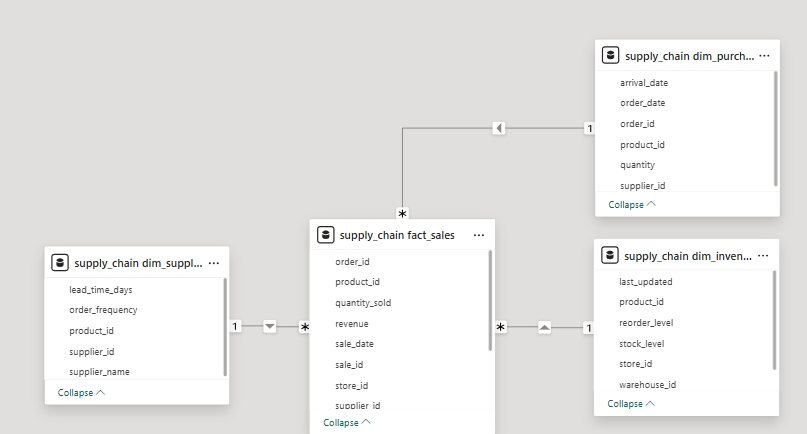
* Findings: total 3 clusters are formed.



* Findings: total 3 clusters are formed.
* I have saved the python notebook and all the python tasks performed. The file ‘rushitha\_supply\_chain.ipynb’ is attached in the folder.

**Power BI Dashboard:**

* The 4 Tables in MYSQL Workbench are loaded into power query for transformation.
* The data types are changed for ID columns from number to text as there is no need of performing mathematical calculations.
* After transformation data is loaded into report.
* Goto Model view and check the relationships.



* Findings: I have checked the relations they satisfy ‘star schema’.
* I added title ‘Supply chain & Inventory management’ using text box.
* **'supply\_chain fact\_sales' table Measures:**

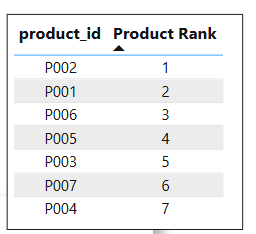
1. Created measure total revenue. Displayed using card visual.

 Findings: total revenue generated is 73 thousand.

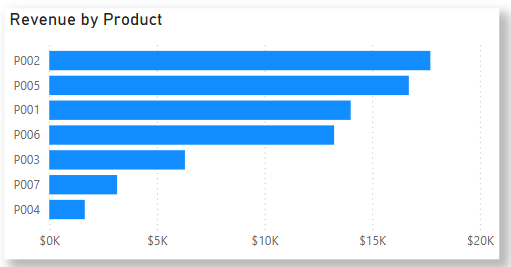
1. Created measure total Sales. Displayed using card visual.

 Findings: total number of sales done is 16.

1. Created measure total Sales quantity.
2. Created measure Product Rank. this is calculated based on total Sales quantity.

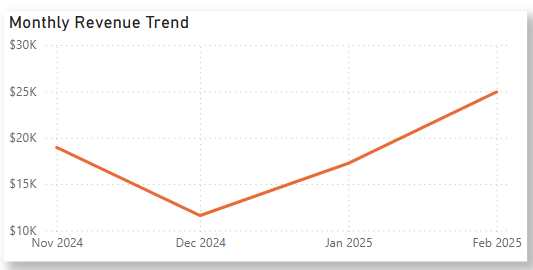
 Findings: Only 7 products are getting sold and P002 is highest sold product by quantity.

1. Plotted Revenue by Product using clustered bar chart.



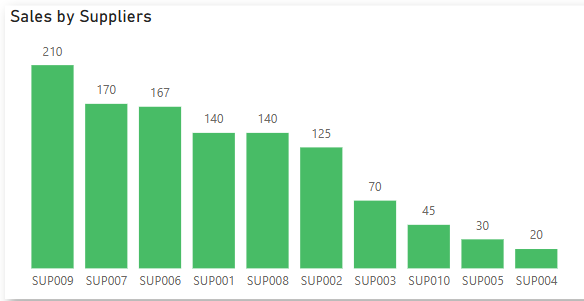
Findings: P002 is the highest revenue generating product.

1. Plotted Monthly revenue trend using line chart.



Findings: Feb 2025 has highest revenue and Dec 2024 has lowest revenue.

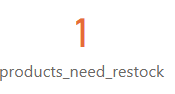
1. Plotted Sales by suppliers using clustered column chart. this is calculated based on total Sales quantity.



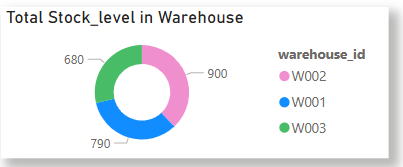
Findings: SUP009 is the biggest supplier.

* **supply\_chain dim\_inventory table Measures:**

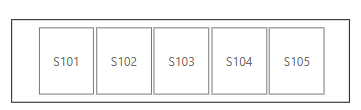
1. Created measure Product need restock. Based on if stock level less than reorder level. Displayed using card visual.

Findings: There is 1 product that needed to be restocked.

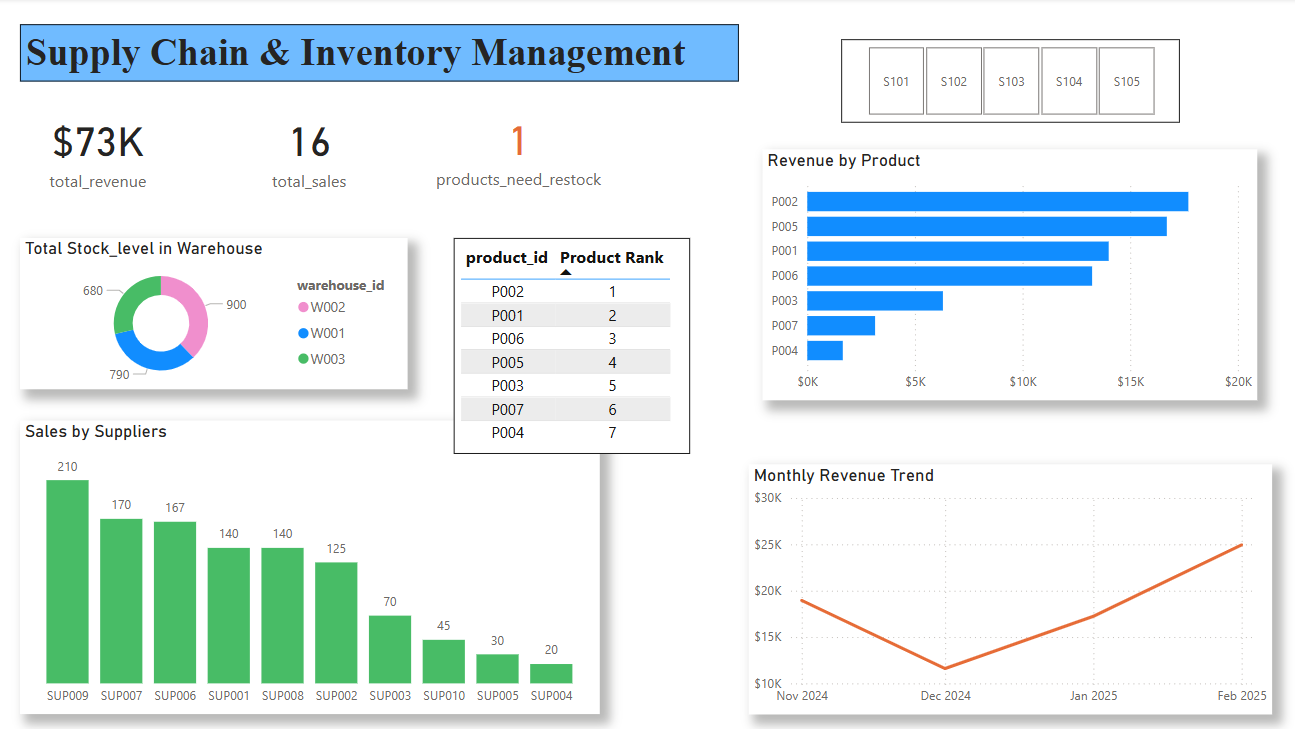
1. Plotted Total stock level in warehouse using donut chart.

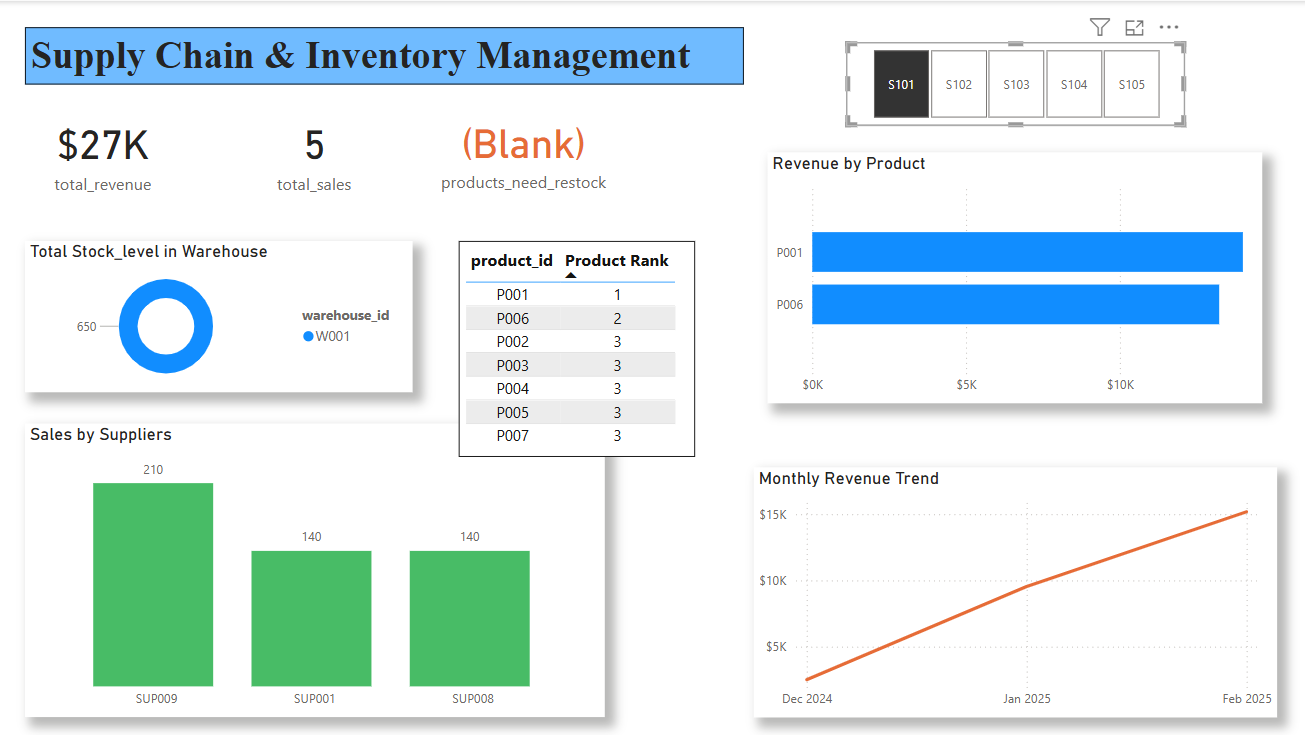
 Findings: W002 has the highest stock.

1. Created Stores slicer to filter all the visuals to get each store data.



**Complete Dashboard**.





* I have saved the power bi The file ‘Rushitha\_supply\_chain.pbix’ is attached in the folder.