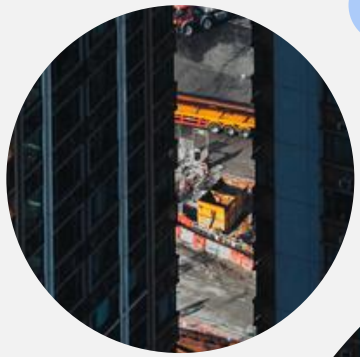
Consulting & Analytics Club

Solving Inventory Inefficiencies Using SQL

IIT GUWAHATI

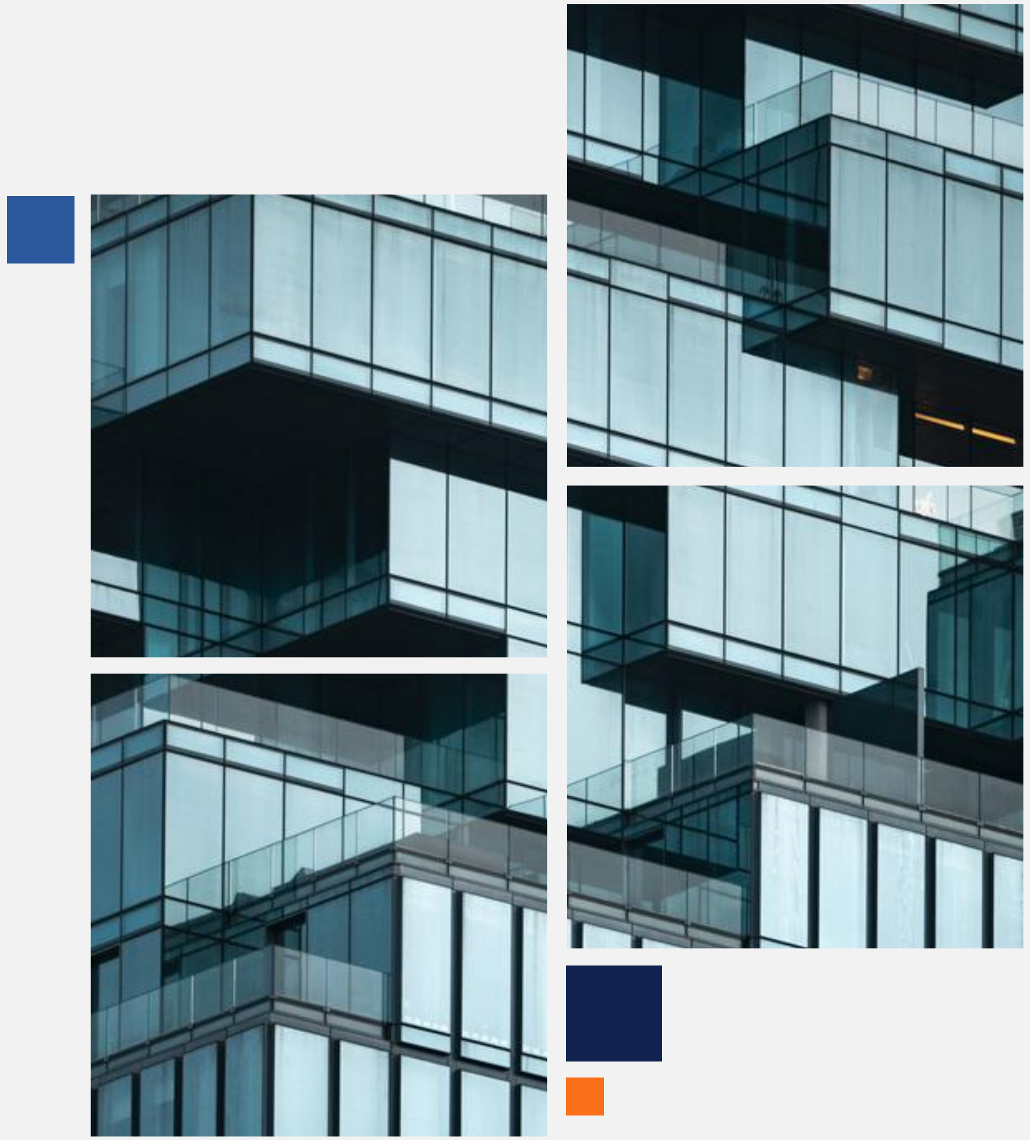
Agenda

- Project Overview
- Problem Statement
- Approach
- Analytical Findings
- Power BI Dashboard
- Conclusion



Project Overview

To analyze inventory inefficiencies and uncover actionable insights for improving demand forecasting, stock allocation, pricing, and promotion strategies across different products, categories, stores, and seasons.



Problem Statement

Urban Retail Co. is a mid-sized omnichannel retailer managing thousands of stock-keeping units (SKUs) across regional distribution centers and brick-and-mortar outlets. Today, inventory decisions are made reactively—often leading to:

Frequent stock-outs of fast-moving items (lost sales, dissatisfied customers)

Chronic overstocks of slow-moving items (excess carrying costs, tied-up capital)

Lack of integrated analytics for demand forecasting, safety-stock setting, and performance monitoring

As a result, the company faces reduced profitability, sub-optimal supply-chain efficiency, and missed revenue opportunities.

Objective: Design a SQL-based solution to analyze and optimize inventory operations through schema normalization, KPI tracking, and advanced analytics.

Approach

Data Exploration & Cleaning

- Analyzed raw transactional data containing store, product, sales, forecast, and external factors (weather, seasonality, promotion).
- Identified redundancy in the raw dataset requiring normalization.

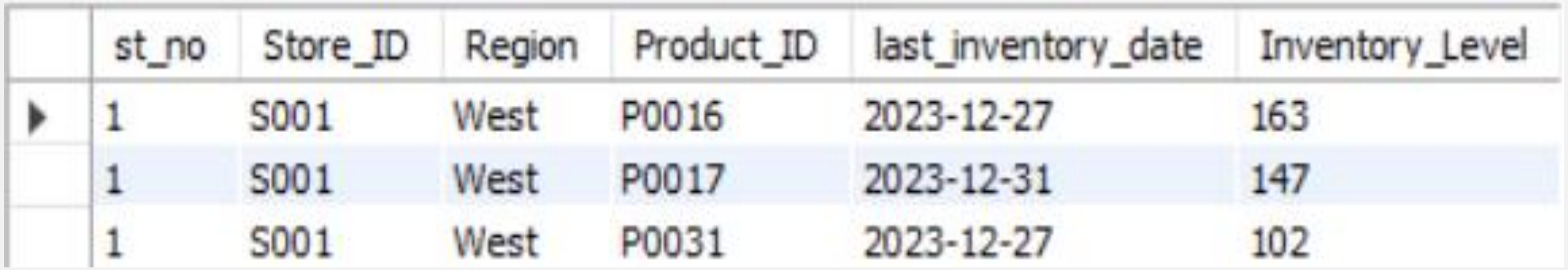
Schema Normalization

- To improve query efficiency and data integrity, the dataset was normalized into the following structure:
- 1. stores - Store_ID and Region
- 2. products - Product_ID and Category
- 3. inventory_facts - Daily transactions (Inventory, Orders, Forecasts, Weather, etc.
- Primary and foreign keys were applied, and data was populated from the raw table using INSERT INTO ... SELECT DISTINCT queries.

Key SQL Implementations

A. Stock Level Calculations

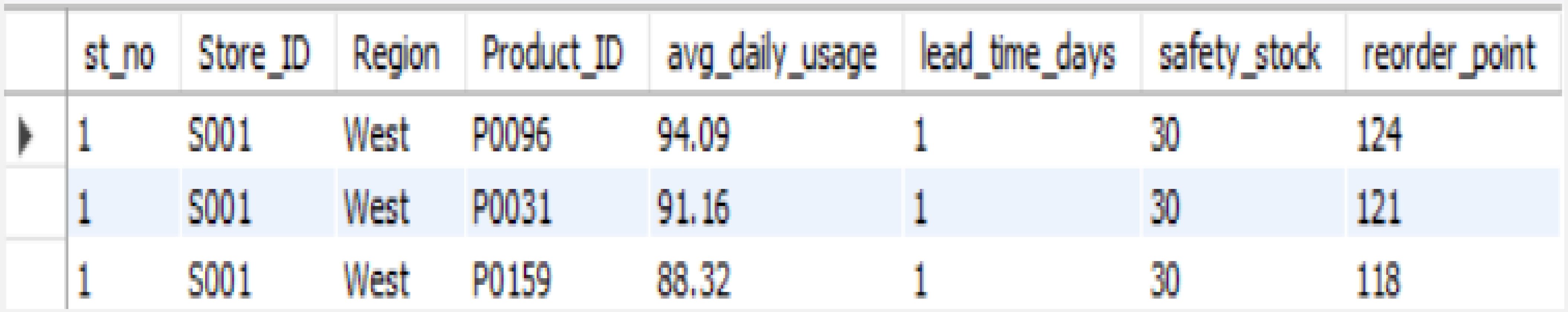
- **Captured the latest inventory levels** for each product in every store by combining store–product data with the most recent inventory date using a Common Table Expression (CTE) and MAX(Date) grouped by store and product.
- **Joined the latest date back to the main facts table** to retrieve the most recent Inventory_Level, giving a real-time snapshot of stock across all store locations.
- Calculating the latest inventory helps identify the **current stock level** of each product at every store, which is essential for **restocking decisions**, avoiding **overstock or stockouts**, and aligning with **forecasted demand**.



	st_no	Store_ID	Region	Product_ID	last_inventory_date	Inventory_Level
▶	1	S001	West	P0016	2023-12-27	163
	1	S001	West	P0017	2023-12-31	147
	1	S001	West	P0031	2023-12-27	102

B. Reorder Point (ROP) Analysis

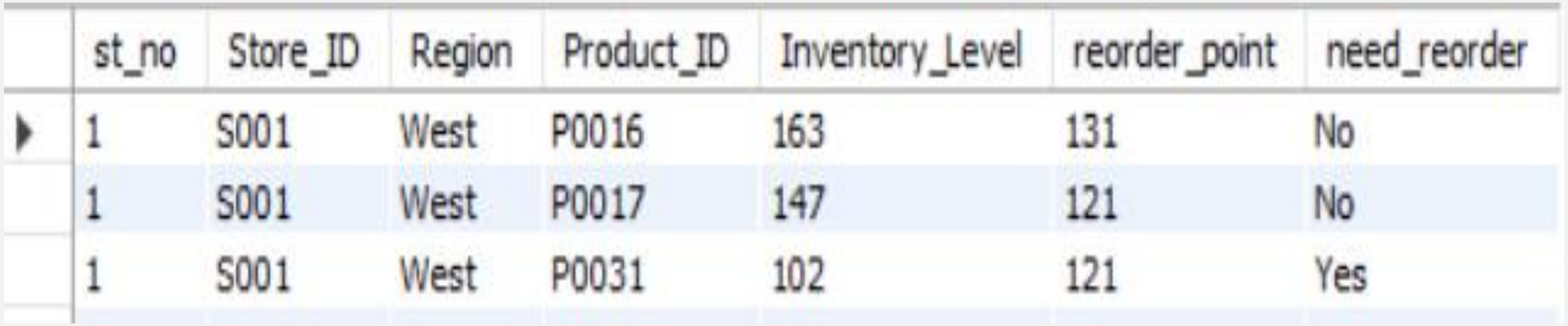
- Formula used: ROP = **(Average Daily Usage * Lead Time) + Safety Stock**. Assumed Lead Time as 1 day and Safety Stock as 30 items.
- Created a query to calculate ROP per store-product and inserted it into reorder_estimations.
- Calculating ROP (Reorder Point) is important to ensure products are **reordered before stock runs out**, helping maintain **continuous availability** and avoid **lost sales due to stockouts**.
- ROP also helps in business analysis by identifying **which products need urgent restocking**, enabling better **inventory planning**, reducing **holding costs**, and improving **supply chain efficiency**.



st_no	Store_ID	Region	Product_ID	avg_daily_usage	lead_time_days	safety_stock	reorder_point
1	S001	West	P0096	94.09	1	30	124
1	S001	West	P0031	91.16	1	30	121
1	S001	West	P0159	88.32	1	30	118

C. Low Inventory Detection based on reorder points

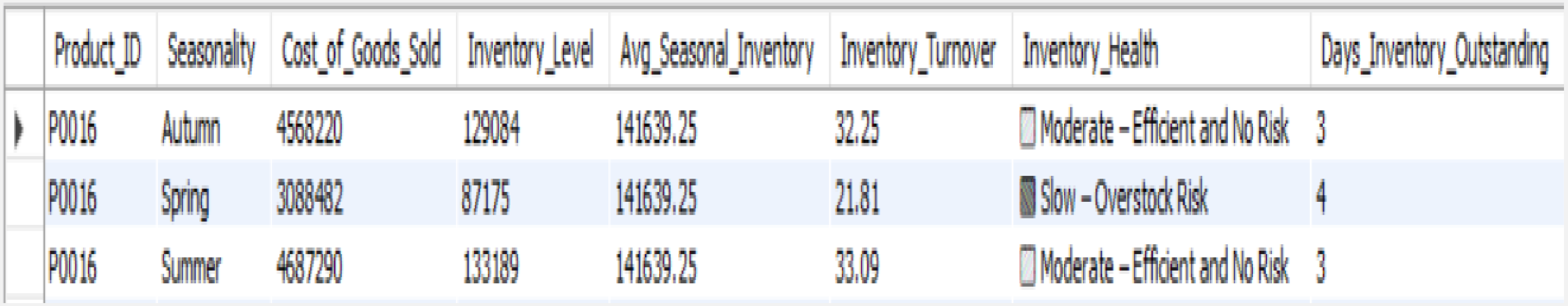
- Identified low stock situations by **comparing current inventory levels against the reorder point** for every product–store combination.
- Flagged products needing restocking using a clear **Yes/ No indicator (need_reorder)**, supporting proactive inventory replenishment decisions.
- **Immediately reorder products** that are below their ROP to avoid stockouts.
- **Adjust inventory planning** based on demand patterns to maintain optimal stock levels.



st_no	Store_ID	Region	Product_ID	Inventory_Level	reorder_point	need_reorder
1	S001	West	P0016	163	131	No
1	S001	West	P0017	147	121	No
1	S001	West	P0031	102	121	Yes

D. Inventory Turnover Analysis

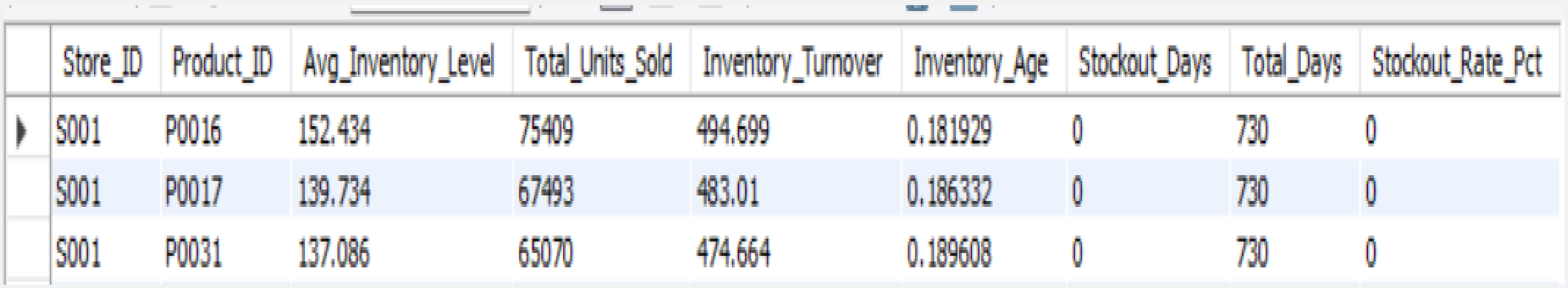
- Formula used : **Inventory Turnover = Cost of Goods Sold (COGS) / Avg Seasonal Inventory** .
- We have used the final Inventory Turnover Classification : **< 25 Slow** (Overstock risk) , **25 - 50 Moderate** (Acceptable) **,** **> 50 Fast moving** (Excellent).
- **Measures how quickly inventory is sold and replaced**, indicating product movement efficiency.
- **Helps identify slow-moving or excess stock**, allowing better inventory control and reduced holding costs.
- **Supports smarter purchasing and stocking decisions** based on actual product demand and sales speed.



Product_ID	Seasonality	Cost_of_Goods_Sold	Inventory_Level	Avg_Seasonal_Inventory	Inventory_Turnover	Inventory_Health	Days_Inventory_Outstanding
P0016	Autumn	4568220	129084	141639.25	32.25	☐ Moderate – Efficient and No Risk	3
P0016	Spring	3088482	87175	141639.25	21.81	▨ Slow – Overstock Risk	4
P0016	Summer	4687290	133189	141639.25	33.09	☐ Moderate – Efficient and No Risk	3

E. Summary Reports with KPIs

- Average Inventory Level: Calculated using AVG(Inventory_Level) - gives the average stock available for each product in a store over time.
- Inventory Turnover: Calculated as Total units sold / Average inventory — measures how many times inventory is sold and replaced during the period.
- Inventory Age: Computed as 90 / Inventory Turnover - estimates the average number of days inventory sits before being sold.
- **Stockout Days** refer to the number of days a product had **zero inventory available**, meaning it was completely out of stock and unavailable for sale.
- Stockout Rate (%):Calculated using (Stockout Days / Total Days)*100 - shows the percentage of days a product was out of stock.



Store_ID	Product_ID	Avg_Inventory_Level	Total_Units_Sold	Inventory_Turnover	Inventory_Age	Stockout_Days	Total_Days	Stockout_Rate_Pct
S001	P0016	152.434	75409	494.699	0.181929	0	730	0
S001	P0017	139.734	67493	483.01	0.186332	0	730	0
S001	P0031	137.086	65070	474.664	0.189608	0	730	0

Analytical Findings:

Optimal Discount Range

Identify the **optimal discount range** that balances both **demand stimulation** and **profitability**.

Price vs Competitor Impact

Compares product prices with competitor prices by category and checks how it affects sales. It helps find if products are overpriced, underpriced, or well-priced based on how much customers are buying.

Product Profitability Status

Checks each product's price vs competitor pricing and demand forecast to decide its business status. It helps identify products that are profitable, overpriced, underpriced, or not selling well - so the company can take action like adjusting prices or removing poor performers.

Promotion Effectiveness

Compares product sales during holidays vs regular days to measure how much promotions increase demand.It helps identify which products benefit most from promotions and which ones show little or no effect - useful for planning discounts.

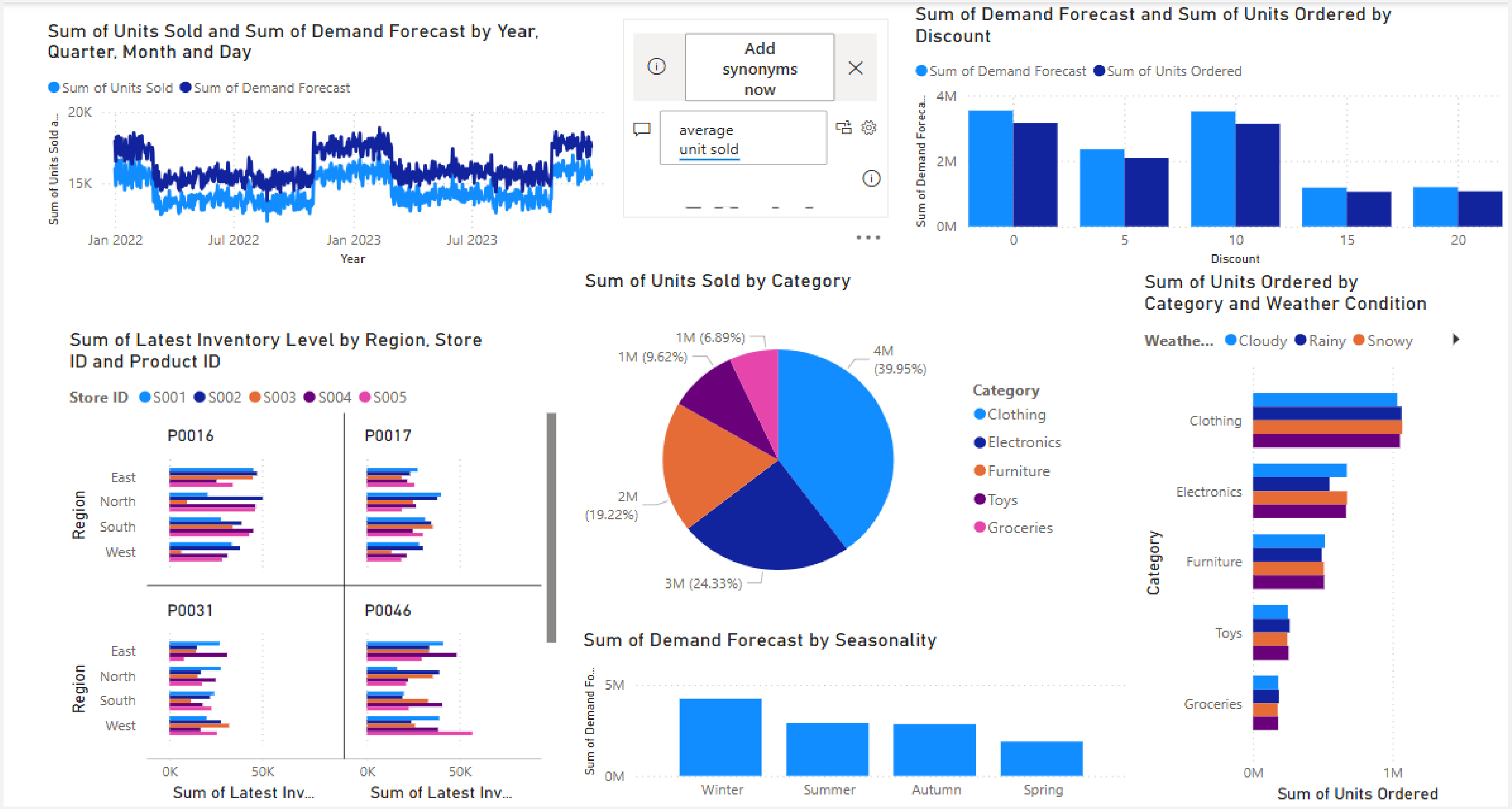
Weather-Based Demand Analysis

How different weather conditions affect sales for each product. It helps find which weather (like rainy or sunny) boosts demand, so stock and marketing can be planned accordingly.

Seasonal Performance Evaluation

How each product performs across different seasons based on sales, forecast, price, and competition. It helps decide if a product is profitable, overpriced, underperforming, or needs review - guiding pricing, stocking, and seasonal planning.

Power BI Dashboard


Sum of Units Sold and Sum of Demand Forecast by Year, Quarter, Month and Day
● Sum of Units Sold ● Sum of Demand Forecast
Sum of Units Sold a...
20K
15K
Jan 2022
Jul 2022
Jan 2023
Jul 2023
Year
Add synonyms now
average unit sold
Sum of Demand Forecast and Sum of Units Ordered by Discount
● Sum of Demand Forecast ● Sum of Units Ordered
Sum of Demand Foreca...
4M
2M
0M
0
5
10
15
20
Discount
Sum of Units Sold by Category
1M (6.89%)
1M (9.62%)
4M (39.95%)
2M (19.22%)
3M (24.33%)
Category
● Clothing
● Electronics
● Furniture
● Toys
● Groceries
Sum of Units Ordered by Category and Weather Condition
Weathe... ● Cloudy ● Rainy ● Snowy
Clothing
Electronics
Furniture
Toys
Groceries
Category
0M
1M
Sum of Units Ordered
Sum of Latest Inventory Level by Region, Store ID and Product ID
Store ID ● S001 ● S002 ● S003 ● S004 ● S005
P0016
P0017
P0031
P0046
Region
East
North
South
West
0K
50K
Sum of Latest Inv...
Sum of Demand Forecast by Seasonality
Sum of Demand Fo...
5M
0M
Winter
Summer
Autumn
Spring

Conclusion

In this project, we analyzed and optimized inventory and pricing strategies for Urban Retail Co. using structured SQL queries. A comprehensive KPI summary was generated per store and product, including metrics like average inventory, units sold, inventory turnover, inventory age, stockout days, and stockout rate (%). These helped identify slow-moving products, overstock risks, and potential stockout issues.

Drive Link for SQL Queries, ER Diagram and Dashboards.

https://drive.google.com/drive/folders/13Y-2fxllKmeJpoJ2U-xuG-F98xhWfudQ

Team Name: Data Decoders

Team Members: K Ramya

V Sai Kiran Reddy

R Kiran Kumar