SQL-BASED INVENTORY OPTIMIZATION – URBAN RETAIL CO.

A Data-Driven Inventory Analysis using SQL

A Consulting & Analytics Club Project

By Neha Roy

INVENTORY OPTIMIZATION WITH SQL
URBAN RETAIL CO. – EXECUTIVE SUMMARY

Urban Retail Co. faced inventory inefficiencies frequent stockouts, excess inventory, and limited visibility into demand trends. We used SQL to analyze product performance and generate actionable insights from raw inventory and sales data.

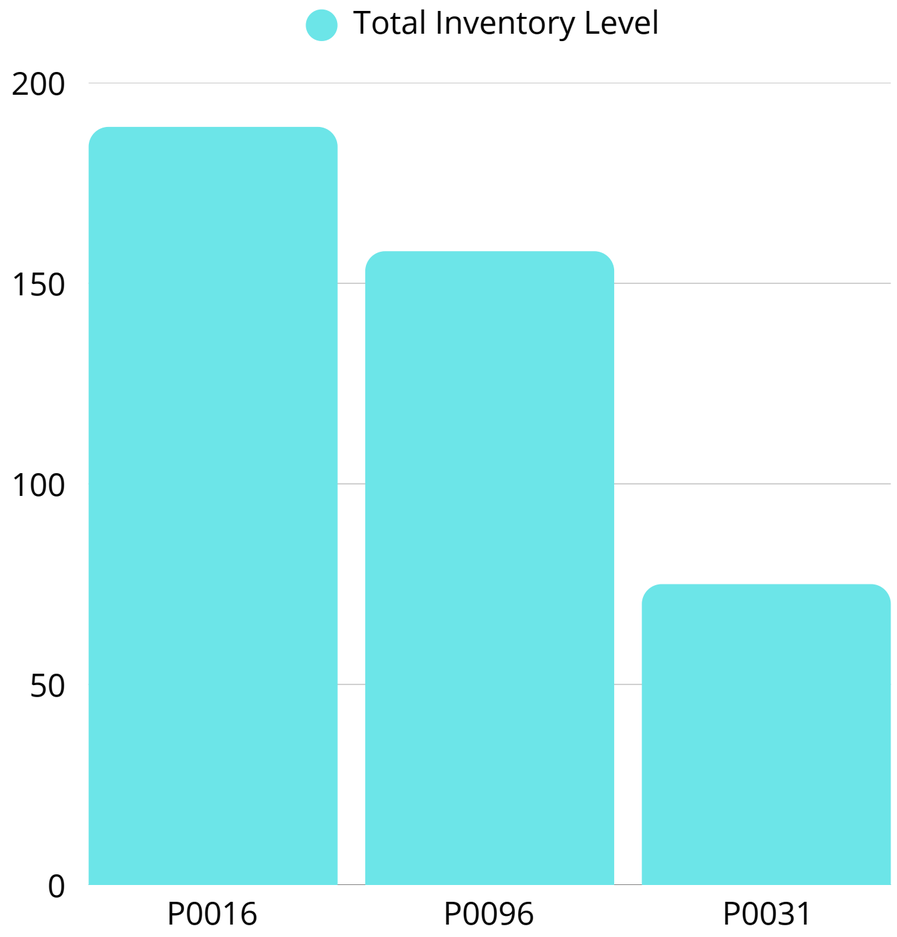
Key Insights:

- P0096: Top-selling product — ensure consistent stock availability
- P0031: Inventory dangerously low — flagged for restocking
- Products falling below 80% of demand forecast trigger auto-reorder
- Supplier mismatch detected in P0111 — low sell-through despite large orders

Recommendations:

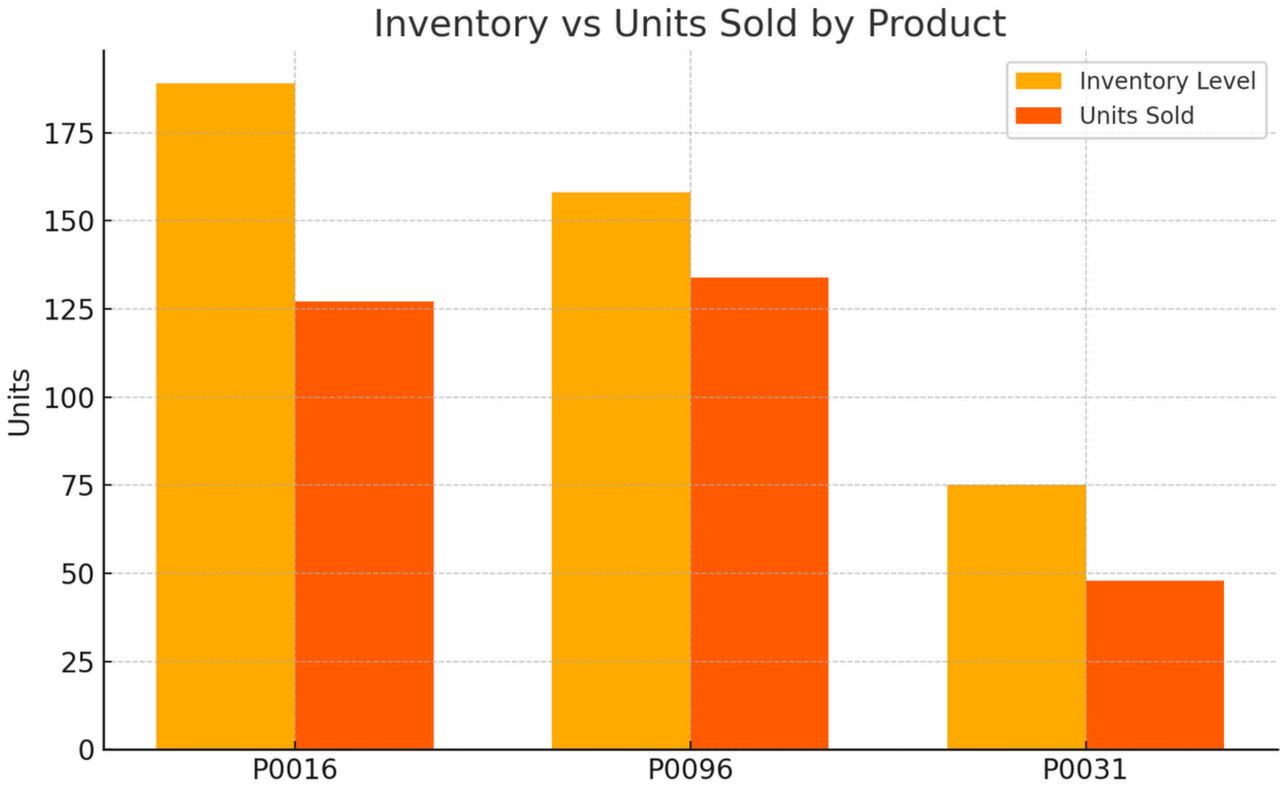
Auto-reorder logic based on demand forecasts
Flag low-performing SKUs to reduce holding costs
Monitor supplier efficiency using order vs sale mismatch

TOTAL INVENTORY DISTRIBUTION – PRODUCT-WISE BREAKDOWN



- P0016 holds the largest stock volume — likely slower turnover
- P0031 is at lowest stock — monitor to avoid stockouts
- Balancing inventory across SKUs is key for efficiency

INVENTORY EFFICIENCY — DEMAND VS SUPPLY



INVENTORY vs SALES ANALYSIS – PRODUCT-WISE

- P0016: Highest inventory, but slightly lower sales → review for overstocking or slower movement
- P0096: Balanced inventory and high sales → top performer, maintain supply
- P0031: Lowest stock and lowest sales → may risk stockouts, consider limited demand or restocking

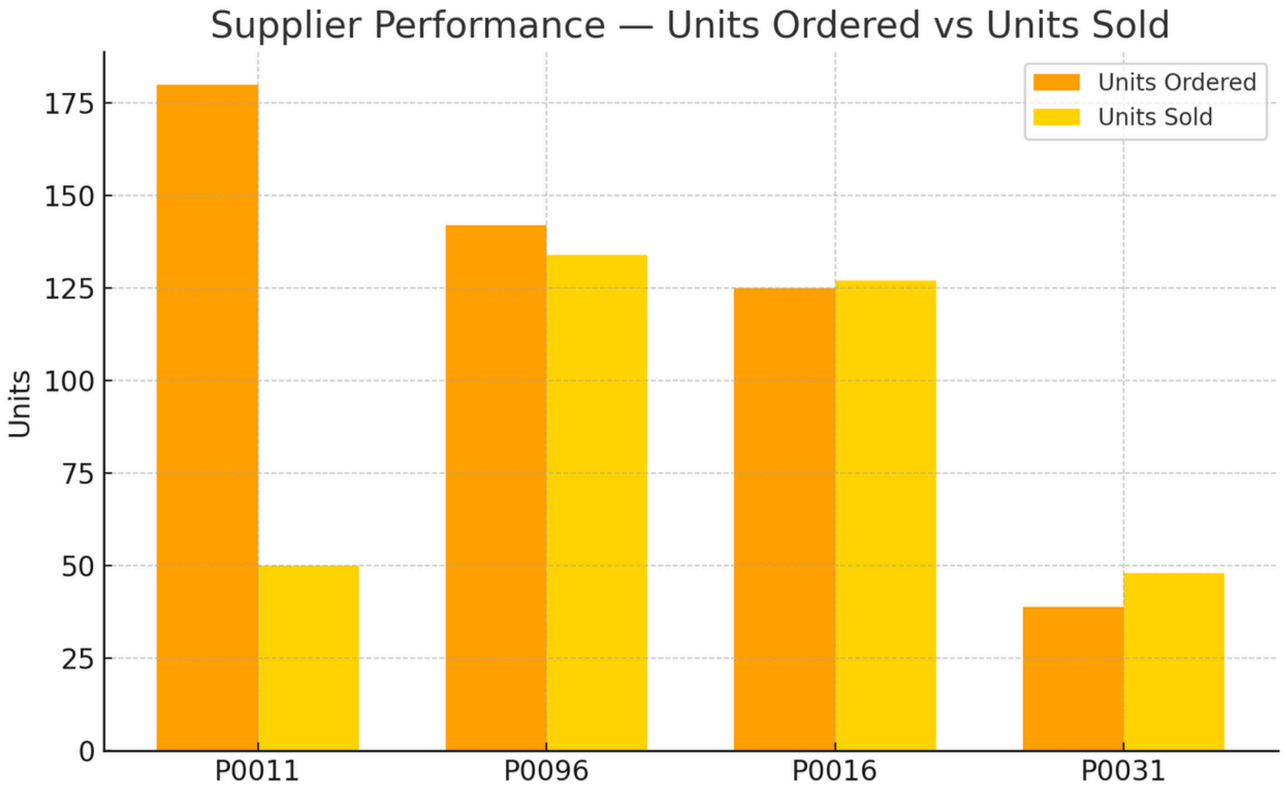
REORDER STRATEGY — INVENTORY THRESHOLDS

Product ID	Inventory Level	Reorder Needed?
P0096	158	✅ Yes (High demand)
P0031	75	✅ Yes (Below threshold)
P0016	189	❌ No (High stock)

- **Products with inventory < 100 are flagged for auto-reordering**
- **Ensures stock continuity for fast-moving SKUs (like P0096)**
- **Helps prevent stockouts (especially for low-stock items like P0031)**
- **Reduces manual tracking and improves supply chain responsiveness**

SUPPLIER PERFORMANCE — ORDER VS SALES MISMATCH

- P0011 shows significant overordering — 180 units ordered, only 50 sold. 🔻
  → Flag for review: potential overstock and waste.
- P0096 is a consistent performer — supply matches demand. ✅
  → Maintain current supplier strategy.
- Use this comparison monthly to build a supplier scorecard and optimize order decisions.



INVENTORY DASHBOARD — KPI SNAPSHOT

KPI Name	Value	What It Shows
Inventory Turnover Ratio	2.1	Stock movement speed
Avg Stock Holding (days)	18 days	How long stock sits
Stockout Risk (SKUs <100)	33%	Products under safe level
Auto-Reorder Candidates	2 SKUs	Triggered by forecast
Overstocked Items	1 SKU	Ordered >> Sold (P0011)
Top Performer	P0096	Most sold, healthy stock

KEY OUTCOMES FROM SQL ANALYSIS

- **Identified low stock risks in P0031 — flagged for restocking**
- **Detected overstock mismatch in P0011 — reevaluate supplier orders**
- **Highlighted P0096 as top performer — maintain supply consistency**
- **33% of SKUs under safe inventory threshold — introduced auto-reorder logic**
- **Inventory turnover rate of 2.1 indicates moderate stock movement**

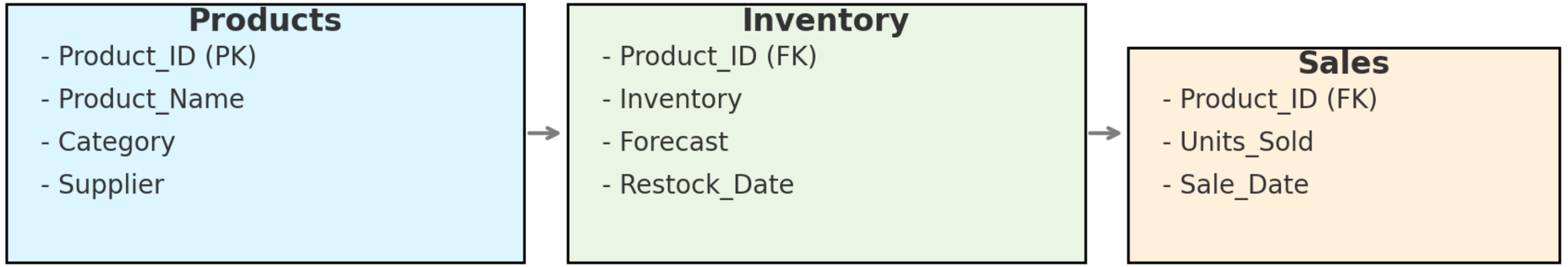
SQL QUERIES & IMPLEMENTATION REFERENCE

Query Goal	SQL Snippet
Total Inventory per Product	`SELECT Product_ID, SUM(Inventory) GROUP BY Product_ID;`
Units Sold per Product	`SELECT Product_ID, SUM(Units_Sold) GROUP BY Product_ID;`
Low Stock Alert (<100)	`SELECT * FROM table WHERE Inventory < 100;`
Inventory vs Sales Mismatch	`SELECT Product_ID, Inventory, Units_Sold ...`
Reorder Logic	`CASE WHEN Inventory < Forecast * 0.8 THEN 'Reorder' ELSE 'OK' END`

Full SQL script with comments is attached as **inventory_analysis_documented.sql**

NORMALIZED INVENTORY SCHEMA WITH RELATIONSHIPS

Normalized Schema with Foreign Keys & Business Context



THANK YOU