🧾 Executive Summary

Project: SQL-Based Inventory Monitoring & Optimization
Client: Urban Retail Co.
Date: June2025

🏢 Background

Urban Retail Co., a mid-sized retail chain with both physical and online operations, has been facing challenges with inventory management across its 5,000+ SKUs. Key issues include frequent **stockouts of fast-moving items**, **overstocking of slow-moving products**, and a general **lack of real-time visibility** across product categories and store regions.

Our team developed a SQL-based analytics framework to enable smarter, data-driven inventory decisions. This involved normalizing the raw inventory dataset, designing a relational database, and writing optimized queries to extract key performance indicators (KPIs) and actionable insights.

🔍 Key Insights

1. Stockout Risks Identified

- Products with sales exceeding forecasts (e.g., *Product P0096, Toys category*) are at risk of running out.

- High sales on holidays and in sunny weather show demand spikes not currently factored into reordering.

2. Overstock & Slow-Movers

- Items like *Product P0016 (Clothing)* and *P0159 (Electronics)* show excess inventory compared to demand forecasts.

- These overstocked items tie up capital and inflate warehouse costs.

3. Reorder Point Optimization

- Using historical sales, we calculated more accurate reorder points using a 20% buffer over average demand.
- This helps prevent future stockouts while avoiding over-purchasing.

4. Inventory Turnover Efficiency

- Fast-sellers (e.g., P0096, P0116) had strong turnover ratios (>1.5), while slow-movers lagged (<0.5).
- Low turnover signals inefficient use of shelf space and capital.

5. Impact of Seasonality and Weather

- Products like *P0070* and *P0171* showed stronger demand in certain weather and seasonal conditions (e.g., Winter, Cloudy).
- Competitor pricing did not always align with our promotions, leading to missed pricing opportunities.

📈 Recommendations

✅ 1. Implement Data-Driven Reordering

- Set dynamic reorder points using rolling average demand + seasonal/weather adjustments.
- Automate restocking triggers for high-demand SKUs nearing thresholds.

✅ 2. Reduce Overstock

- Apply markdowns or promotional strategies for slow-movers.
- Consider SKU rationalization for low-turnover products.

✅ 3. Align Promotions with Demand Patterns

- Schedule promotions for high-selling SKUs during holidays and favorable weather periods.
- Review pricing vs. competitors more frequently to remain competitive.

✅ 4. Build Real-Time Inventory Dashboard

- Use SQL queries to feed a Power BI/Tableau dashboard for store managers and procurement teams.
- Include alerts for stockouts, overstock, and reorder point breaches.

📊 Business Impact

With these changes, Urban Retail Co. can expect:

- 📉 **Reduced stockouts** by 20–30%
- 💰 **Lower holding costs** through better overstock management
- 🧠 **Improved decision-making** through real-time data visibility
- 😊 **Higher customer satisfaction** via better product availability

---

🔚 Conclusion

By transforming raw sales and inventory data into actionable insights using SQL, Urban Retail Co. can significantly improve its inventory management, reduce operational costs, and strengthen its competitive position. This project serves as a foundation for long-term data-driven growth.