

BY : ACHALA PANDEY

---

# Streamlit Dashboard for Inventory Insights

GitHub Repo: [Streamlit dashboard1](#)

Live App: <https://inventorydash.streamlit.app/>

## Project Overview

This project is a data-driven dashboard built using **Streamlit**, powered by a **PostgreSQL** backend hosted on **Supabase**. It visualizes and analyzes inventory data to support smarter stock management decisions to help URBAN Retail Co..

The core of the project lies in its ability to:

- Display real-time inventory metrics
- Highlight low-stock items
- Recommend restocking actions based on usage trends

## Schema & Data Flow

The database schema was designed using an ER diagram that models key entities such as:

- `products`
- `Store`
- `Time`
- `Region`

SQL queries were optimized for performance and scalability, using joins and aggregations to extract actionable insights.

## Recommendation Logic

The recommendation engine is based on a **threshold-based heuristic**:

If `inventory_level` < `reorder_threshold`, then flag the item as **"Needs Restocking"**

1. URGENT Reorder: When inventory is completely out (`inventory_level == 0`)[STOCKOUT RATE]
2. Reduce Holding: When inventory is more than twice the expected demand — indicating potential overstock
3. Reorder Soon: When inventory is low but not depleted — it's below forecasted demand
4. Stock OK: When stock levels are healthy and aligned with expected needs

## Why This Works:

- It avoids overstocking by only recommending what's necessary
- It prevents stockouts by proactively identifying low inventory
- It's easy to explain and adjust for different business needs

## Key Insights Delivered

1. **Low Stock Alerts**  
The dashboard highlights products that are below their reorder threshold, helping managers act before stockouts occur.
2. **Top-Selling Items**  
A ranked list of frequently used or sold items helps prioritize procurement and marketing focus.
3. **Inventory Turnover Trends- FAST vs SLOW moving products.**  
Visualizations show how quickly items are moving, helping identify slow-moving stock that may need discounts or bundling.
4. **Category-Level Analysis**  
Aggregated views by product category help spot trends and imbalances across departments.



## Recommendations

- **Automate Restocking**: Integrate with a procurement system to auto-generate purchase orders for flagged items: **STOCK OK** , **REDUCE HOLDING** , **REORDER SOON** .
- **Dynamic Thresholds**: Use of seasonal trends to adjust reorder levels over time.
- **Export Reports**: Enable CSV or PDF export for offline analysis and sharing.

## Deployment & Tech Stack

- **Frontend:** Streamlit
  - **Backend:** PostgreSQL via Supabase
  - **Deployment:** Streamlit Cloud
  - **Data Access:** SQLAlchemy with connection pooling via Supabase's built-in PostgreSQL connection pooler.
-