Inventory Analysis SQL Documentation

This document describes various SQL queries used in analyzing stock levels, inventory performance, and sales trends using a relational schema across multiple tables: stores, inventory, external_factor, seasonality, weather, and category.

1. Stock Summary by Store

Query Purpose: Calculate the total available stock for each product across different stores and regions.

- **Inputs: stores, inventory**
- **Metric: SUM(inventory_level)**
- **Grouped By: store_id, region, product_id**
- **Sorted By: Descending total_stock**

2. Low Inventory Alert System

2.1 Reorder Point Calculation

Purpose: Compute product-level reorder point using a 3-day lead time and standard deviation.

- **Inputs: stores, inventory**
- **Metric: (AVG(units_sold) * 3) + STDDEV(units_sold)**

2.2 Inventory Status Flagging

Purpose: Classify current stock as Low Stock or Sufficient.

- **Logic: Compare inventory_level against reorder_point**
- **Output: Classification label (Low Stock, Sufficient)**
- **Ordered By: status ASC**

3. Historical Reorder Point Estimation

Purpose: Provide reorder estimates per store-product combination based on historical sales.

- **Metric: estimated_reorder_point = (AVG(units_sold) * 3) + STDDEV(units_sold)**
- **Sorted By: Highest reorder needs**

4. Inventory Turnover Ratio

Purpose: Analyze inventory turnover per product per month.

- **Formula: turnover = total_units_sold / avg_inventory_level**

- Time Bucket: month (YYYY-MM)
- Use Case: Identify slow/fast moving SKUs and restocking effectiveness

5. Key Inventory KPIs

5.1 Stockout Rate by Season and Weather

- Join Tables: stores, inventory, external_factor, seasonality, weather
- Reorder Threshold: Based on product and season
- Output:
  - stockout_count
  - total_records
  - stockout_rate_percent

5.2 Average Inventory Age

Purpose: Measure how long inventory has been sitting in stores.

- Metric: DATEDIFF(current_date, record_date)
- Condition: Only items with inventory_level > 0

5.3 Average Stock by Weather and Season

Purpose: Understand how average inventory levels vary with external conditions.

- Groupings: By product, season, and weather
- Metric: AVG(inventory_level)

6. Product Movement Classification

Goal: Categorize products into:

- Fast-Selling: Above market average daily sales
- Slow-Moving: Below market average
- Calculation:
  - avg_daily_units_sold = SUM(units_sold) / number_of_days
  - Use a subquery to compare against global average

7. Seasonal Demand Forecast Analysis

Goal: Compare forecasted demand with actual sales.

- Inputs: inventory.demand_forecast vs inventory.units_sold
- Group By: season, product_id, category
- Sorted By: avg_seasonal_demand DESC

8. Overstock vs Stockout Day Count

Purpose: Identify days with either excessive or insufficient stock levels.

8.1 Preprocessing

- **Step 1: Compute reorder point and average inventory per product**
- **Step 2: Derive overstock multiplier (avg_inventory / reorder_point)**

8.2 Daily Flags

- **Conditions:**
  - **is_stockout = inventory_level < reorder_point**
  - **is_overstock = inventory_level > reorder_point * multiplier**

8.3 Final Output

- **Metrics:**
  - **stockout_days: # of unique days flagged as understocked**
  - **overstock_days: # of unique days flagged as overstocked**
- **Grouped By: store_id, product_id**

Entity Relationships Overview

Table Name	Description
stores	Store-wise product mapping and inventory linkage
inventory	Daily inventory levels, units sold, demand
external_factor	Season and weather reference for a store
seasonality	Season names per season ID

weather	**Weather type per weather ID**
category	**Product category mapping**