
▮ Theme: Retail Inventory & Supply Chain Intelligence

▮ Dataset: `inventory_supply.csv`

```
ItemID,ItemName,Category,Warehouse,StockQty,ReorderLevel,LastRestocked,UnitPrice,Supplier
I001,LED TV,Electronics,WarehouseA,50,20,2024-03-15,30000,AVTech
I002,Laptop,Electronics,WarehouseB,10,15,2024-04-01,70000,TechWorld
I003,Office Chair,Furniture,WarehouseA,40,10,2024-03-25,6000,ChairCo
I004,Refrigerator,Appliances,WarehouseC,5,10,2024-02-20,25000,FreezeIt
I005,Printer,Electronics,WarehouseB,3,5,2024-03-30,8000,PrintFast
```

▮ New Pattern: Business Scenarios with Multi-Step Tasks

▮ Scenario 1: Inventory Alerting System

Tasks:

1. Load the data using PySpark.
 2. Create a new column `NeedsReorder = StockQty < ReorderLevel` .
 3. Create a view of all items that need restocking.
 4. Highlight warehouses with more than 2 such items.
-

▮ Scenario 2: Supplier Price Optimization

Tasks:

1. Group items by Supplier and compute average price.
 2. Find which suppliers offer items **below average price** in their category.
 3. Tag suppliers with `Good Deal` if >50% of their items are below market average.
-

▮ Scenario 3: Cost Forecasting

Tasks:

1. Calculate `TotalStockValue = StockQty * UnitPrice` .
 2. Identify top 3 highest-value items.
 3. Export the result as a Parquet file partitioned by `Warehouse` .
-

▮ Scenario 4: Warehouse Utilization

Tasks:

1. Count items stored per warehouse.
 2. Average stock per category in each warehouse.
 3. Determine underutilized warehouses (`total stock < 100`).
-

▮ Scenario 5: Delta Audit Trail

Tasks:

1. Save as Delta table `retail_inventory` .
 2. Update stock of 'Laptop' to 20.
 3. Delete any item with `StockQty = 0` .
 4. Run `DESCRIBE HISTORY` and query `VERSION AS OF` previous state.
-

▮ Scenario 6: Alerts from Restock Logs (Join Task)

▮ `restock_logs.csv` :

```
ItemID,RestockDate,QuantityAdded
I002,2024-04-20,10
I005,2024-04-22,5
I001,2024-04-25,20
```

Tasks:

1. Join with inventory table to update `StockQty`.
 2. Calculate new stock and flag `RestockedRecently = true` for updated items.
 3. Use `MERGE INTO` to update in Delta.
-

▮ Scenario 7: Report Generation with SQL Views

Tasks:

1. Create SQL view `inventory_summary` with:
 - `ItemName`, `Category`, `StockQty`, `NeedsReorder`, `TotalStockValue`
 2. Create view `supplier_leaderboard` sorted by average price
-

▮ Scenario 8: Advanced Filtering

Tasks:

1. Use `when / otherwise` to categorize items:
 - "Overstocked" (`>2x ReorderLevel`)
 - "LowStock"
 2. Use `.filter()` and `.where()` for the same and compare.
-

▮ Scenario 9: Feature Engineering

Tasks:

1. Extract `RestockMonth` from `LastRestocked` .
 2. Create feature: `StockAge = CURRENT_DATE - LastRestocked`
 3. Bucket `StockAge` into: New, Moderate, Stale
-

▮ Scenario 10: Export Options

Tasks:

1. Write full `DataFrame` to:

- CSV for analysts
- JSON for integration
- Delta for pipelines

2. Save with meaningful file and partition names like
`/export/inventory/stale_items/`
