STOX.AI: Demand Flow & Drill-Down Implementation Guide

Madison Reed Hair Color Products

Scenario: Madison Reed - Premium Hair Color Brand

Product Example: Permanent Hair Color (5 shades), Hair Care Products

Channels: California Retail Stores + E-commerce (Amazon, Madison-Reed.com, Sephora.com)

Processing: All computations in STOX.AI using SAP S/4HANA Retail data extracts

---

1️⃣ DEMAND FLOW: POS → Sell-In Forecast → Supply Planning

Step 1: Extract POS Data from SAP

SAP Table: `WPOS_SALES` (POS Transaction Data)

Table Description: Captures point-of-sale transactions from retail stores and e-commerce platforms

Extraction Query:

```sql
SELECT
    LOCNR,      -- Store/Channel location code
    MATNR,      -- Material number (SKU)
    DATUM,      -- Transaction date
    MENGE,      -- Quantity sold
    NETWR       -- Net value (USD)
FROM WPOS_SALES
WHERE DATUM BETWEEN '20250901' AND '20251020'  -- Last 7 weeks
```

Extracted File: POS_Sales.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
LOCNR	Store/Channel Location Code	MR-SF-001	MR-LA-002	AMZN-US
MATNR	Material Number (SKU)	HC-BAROLO-6N	HC-FIRENZE-7N	GLOSS-ROSE-4N
DATUM	Transaction Date (YYYYMMDD)	20250901	20250902	20250902
MENGE	Quantity Sold (Units)	3	2	5
NETWR	Net Sales Value (USD)	89.97	59.98	149.95

Sample Data Rows:

LOCNR	MATNR	DATUM	MENGE	NETWR
MR-SF-001	HC-BAROLO-6N	20250901	3	89.97
MR-SF-001	HC-FIRENZE-7N	20250902	2	59.98
MR-LA-002	HC-BAROLO-6N	20250902	5	149.95
MR-SD-003	GLOSS-ROSE-4N	20250903	4	119.96
AMZN-US	HC-BAROLO-6N	20250903	12	359.88
MADISON-WEB	HC-FIRENZE-7N	20250904	8	239.92
SEPHORA-WEB	HC-SIENA-5N	20250905	6	179.94
MR-SF-001	ROOT-TOUCH-DARK	20250906	7	139.93

Records: ~18,000 transactions across 7 weeks, 8 SKUs, 7 channels (4 retail + 3 e-commerce)

Step 2: Aggregate POS by SKU-Channel-Week

STOX.AI Processing Logic

```python
import pandas as pd
import numpy as np

# Load POS data
pos_data = pd.read_csv('POS_Sales.csv')

# Convert date to week
pos_data['DATUM'] = pd.to_datetime(pos_data['DATUM'], format='%Y%m%d')
pos_data['YEAR_WEEK'] = pos_data['DATUM'].dt.strftime('%Y-W%W')

# Aggregate by SKU-Channel-Week
pos_aggregated = pos_data.groupby(['MATNR', 'LOCNR', 'YEAR_WEEK']).agg({
    'MENGE': 'sum',
    'NETWR': 'sum'
}).reset_index()

pos_aggregated.columns = ['SKU', 'Channel', 'Week', 'POS_Qty', 'POS_Value']
```

Output: POS_Aggregated.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number (Product SKU)	HC-BAROLO-6N	HC-FIRENZE-7N	GLOSS-ROSE-4N
Channel	Sales Channel/Store Code	MR-SF-001	AMZN-US	MADISON-WEB
Week	Year-Week (ISO 8601)	2025-W36	2025-W37	2025-W38
POS_Qty	Total Units Sold in Week	68	145	92
POS_Value	Total Sales Value (USD)	2,038.32	4,350.05	2,758.16

Sample Data Rows:

SKU	Channel	Week	POS_Qty	POS_Value
HC-BAROLO-6N	MR-SF-001	2025-W36	68	2,038.32
HC-BAROLO-6N	MR-SF-001	2025-W37	72	2,157.84
HC-BAROLO-6N	MR-LA-002	2025-W36	54	1,618.46
HC-BAROLO-6N	AMZN-US	2025-W36	145	4,350.05
HC-BAROLO-6N	MADISON-WEB	2025-W36	98	2,938.02
HC-FIRENZE-7N	MR-SF-001	2025-W36	52	1,559.48
GLOSS-ROSE-4N	SEPHORA-WEB	2025-W36	42	1,259.58
ROOT-TOUCH-DARK	MR-SD-003	2025-W36	31	619.69

Step 3: Map Channel → Customer (Distributor/Retailer)

SAP Table: `WRS1` (Store/Channel Master Data)

Table Description: Master data for retail stores and e-commerce sales channels with customer mapping

Extraction Query:

```
sql
```

```
SELECT
    LOCNR,      -- Store/channel location code
    KUNNR,      -- Customer number (distributor/retailer)
    NAME1,      -- Store/channel name
    REGIO       -- Region/State
FROM WRS1
```

Extracted File: Store_Master.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
LOCNR	Store/Channel Location Code	MR-SF-001	AMZN-US	SEPHORA-WEB
KUNNR	Customer Number (Sold-To Party)	CUST-MRRET	CUST-AMAZON	CUST-SEPHORA
NAME1	Store/Channel Name	Madison Reed SF Union Square	Amazon.com US	Sephora.com
REGIO	Region/State Code	CA	ECOM	ECOM

Sample Data Rows:

LOCNR	KUNNR	NAME1	REGIO
MR-SF-001	CUST-MRRET	Madison Reed SF Union Square	CA
MR-LA-002	CUST-MRRET	Madison Reed LA Beverly Center	CA
MR-SD-003	CUST-MRRET	Madison Reed SD Fashion Valley	CA
MR-OC-004	CUST-MRRET	Madison Reed OC South Coast Plaza	CA
AMZN-US	CUST-AMAZON	Amazon.com US Marketplace	ECOM
MADISON-WEB	CUST-DIRECT	Madison-Reed.com Direct	ECOM
SEPHORA-WEB	CUST-SEPHORA	Sephora.com Online	ECOM

STOX.AI Processing: Join POS with Store Master

```python
# Load store master
store_master = pd.read_csv('Store_Master.csv')

# Join POS aggregated with store master
pos_with_customer = pos_aggregated.merge(
    store_master[['LOCNR', 'KUNNR', 'NAME1']],
    left_on='Channel',
    right_on='LOCNR',
    how='left'
)

pos_with_customer = pos_with_customer[['SKU', 'KUNNR', 'Channel', 'NAME1', 'Week', 'POS_Qty', 'POS_Value']]
pos_with_customer.columns = ['SKU', 'Customer', 'Channel', 'Channel_Name', 'Week', 'POS_Qty', 'POS_Value']
```

Output: POS_With_Customer.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number (Product)	HC-BAROLO-6N	HC-FIRENZE-7N	GLOSS-ROSE-4N
Customer	Customer Number (Distributor)	CUST-MRRET	CUST-AMAZON	CUST-SEPHORA
Channel	Channel/Store Code	MR-SF-001	AMZN-US	SEPHORA-WEB
Channel_Name	Channel/Store Description	Madison Reed SF Union Square	Amazon.com US	Sephora.com
Week	Year-Week	2025-W36	2025-W37	2025-W38
POS_Qty	Units Sold Through Channel	68	145	42
POS_Value	Sales Value (USD)	2,038.32	4,350.05	1,259.58

Sample Data Rows:

SKU	Customer	Channel	Channel_Name	Week	POS_Qty	POS_Value
HC-BAROLO-6N	CUST-MRRET	MR-SF-001	Madison Reed SF Union Square	2025-W36	68	2,038.32
HC-BAROLO-6N	CUST-MRRET	MR-LA-002	Madison Reed LA Beverly Center	2025-W36	54	1,618.46
HC-BAROLO-6N	CUST-AMAZON	AMZN-US	Amazon.com US	2025-W36	145	4,350.05
HC-BAROLO-6N	CUST-DIRECT	MADISON-WEB	Madison-Reed.com Direct	2025-W36	98	2,938.02
HC-FIRENZE-7N	CUST-MRRET	MR-SF-001	Madison Reed SF Union Square	2025-W36	52	1,559.48
GLOSS-ROSE-4N	CUST-SEPHORA	SEPHORA-WEB	Sephora.com	2025-W36	42	1,259.58

Step 4: Extract Sell-In Orders (Customer → Madison Reed)

SAP Table: `VBAP` (Sales Order Items)

Table Description: Line items from customer purchase orders for finished goods

Extraction Query:

```sql
```

```
SELECT
    VBELN,      -- Sales order number
    POSNR,      -- Line item number
    MATNR,      -- Material number
    KUNNR,      -- Customer (sold-to party)
    KWMENG,     -- Order quantity
    NETWR,      -- Net order value
    EDATU       -- Requested delivery date
FROM VBAP
WHERE EDATU >= '20251001'  -- Future orders
  AND VBELN LIKE '5%'      -- Standard sales orders
```

Extracted File: Sales_Orders.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
VBELN	Sales Order Number	5000012345	5000012346	5000012347
POSNR	Line Item Number	000010	000010	000020
MATNR	Material Number (SKU)	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
KUNNR	Customer Number	CUST-AMAZON	CUST-MRRET	CUST-SEPHORA
KWMENG	Order Quantity (Units)	2400	1200	800
NETWR	Net Order Value (USD)	72,000.00	36,000.00	24,000.00
EDATU	Requested Delivery Date	20251105	20251112	20251119

Sample Data Rows:

VBELN	POSNR	MATNR	KUNNR	KWMENG	NETWR	EDATU
5000012345	000010	HC-BAROLO-6N	CUST-AMAZON	2400	72,000.00	20251105
5000012345	000020	HC-FIRENZE-7N	CUST-AMAZON	1800	54,000.00	20251105
5000012346	000010	HC-SIENA-5N	CUST-SEPHORA	1200	36,000.00	20251112
5000012347	000010	HC-BAROLO-6N	CUST-MRRET	1200	36,000.00	20251119
5000012347	000020	GLOSS-ROSE-4N	CUST-MRRET	800	24,000.00	20251119
5000012348	000010	HC-FIRENZE-7N	CUST-DIRECT	1500	45,000.00	20251126

STOX.AI Processing: Convert Orders to Weekly Demand

python

```python
# Load sales orders
sales_orders = pd.read_csv('Sales_Orders.csv')

# Convert delivery date to week
sales_orders['EDATU'] = pd.to_datetime(sales_orders['EDATU'], format='%Y%m%d')
sales_orders['YEAR_WEEK'] = sales_orders['EDATU'].dt.strftime('%Y-W%W')

# Aggregate by SKU-Customer-Week
orders_aggregated = sales_orders.groupby(['MATNR', 'KUNNR', 'YEAR_WEEK']).agg({
    'KWMENG': 'sum',
    'NETWR': 'sum'
}).reset_index()

orders_aggregated.columns = ['SKU', 'Customer', 'Week', 'Order_Qty', 'Order_Value']
```

Output: Orders_Aggregated.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
Customer	Customer Number	CUST-AMAZON	CUST-MRRET	CUST-SEPHORA
Week	Delivery Year-Week	2025-W45	2025-W46	2025-W47
Order_Qty	Total Order Quantity	2400	1200	1200
Order_Value	Total Order Value (USD)	72,000.00	36,000.00	36,000.00

Sample Data Rows:

SKU	Customer	Week	Order_Qty	Order_Value
HC-BAROLO-6N	CUST-AMAZON	2025-W45	2400	72,000.00
HC-FIRENZE-7N	CUST-AMAZON	2025-W45	1800	54,000.00
HC-SIENA-5N	CUST-SEPHORA	2025-W46	1200	36,000.00
HC-BAROLO-6N	CUST-MRRET	2025-W47	1200	36,000.00
GLOSS-ROSE-4N	CUST-MRRET	2025-W47	800	24,000.00
HC-FIRENZE-7N	CUST-DIRECT	2025-W48	1500	45,000.00

Step 5: Run AI Forecast Model

STOX.AI Processing: Time Series Forecasting

```
python
```

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from prophet import Prophet

# Prepare historical POS data for forecasting
forecast_input = pos_with_customer.groupby(['SKU', 'Customer', 'Week']).agg({
    'POS_Qty': 'sum'
}).reset_index()

# Forecast next 26 weeks using Prophet
forecasts = []

for (sku, customer), group in forecast_input.groupby(['SKU', 'Customer']):
    # Prepare data for Prophet
    df_prophet = group[['Week', 'POS_Qty']].copy()
    df_prophet.columns = ['ds', 'y']
    df_prophet['ds'] = pd.to_datetime(df_prophet['ds'] + '-1', format='%Y-W%W-%w')

    # Fit model with seasonality
    model = Prophet(
        seasonality_mode='multiplicative',
        weekly_seasonality=False,
        yearly_seasonality=True
    )
    model.fit(df_prophet)

    # Predict 26 weeks ahead
    future = model.make_future_dataframe(periods=26, freq='W')
    forecast = model.predict(future)

    # Extract forecast for future weeks only
    future_forecast = forecast[forecast['ds'] > df_prophet['ds'].max()][['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
    future_forecast['SKU'] = sku
    future_forecast['Customer'] = customer
```

```
    future_forecast['Week'] = future_forecast['ds'].dt.strftime('%Y-W%W')

    forecasts.append(future_forecast[['SKU', 'Customer', 'Week', 'yhat', 'yhat_lower', 'yhat_upper']])

# Combine all forecasts
forecast_df = pd.concat(forecasts, ignore_index=True)
forecast_df.columns = ['SKU', 'Customer', 'Week', 'Forecast_Qty', 'Forecast_Lower', 'Forecast_Upper']
forecast_df['Forecast_Qty'] = forecast_df['Forecast_Qty'].clip(lower=0).round(0)
```

Output: Forecast_AI.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
Customer	Customer Number	CUST-AMAZON	CUST-MRRET	CUST-SEPHORA
Week	Forecast Year-Week	2025-W43	2025-W44	2025-W45
Forecast_Qty	Forecasted Demand (Units)	625	285	412
Forecast_Lower	Lower Confidence Bound (95%)	550	245	365
Forecast_Upper	Upper Confidence Bound (95%)	700	325	459

Sample Data Rows:

SKU	Customer	Week	Forecast_Qty	Forecast_Lower	Forecast_Upper
HC-BAROLO-6N	CUST-AMAZON	2025-W43	625	550	700
HC-BAROLO-6N	CUST-AMAZON	2025-W44	638	562	714
HC-BAROLO-6N	CUST-AMAZON	2025-W45	651	575	727
HC-BAROLO-6N	CUST-MRRET	2025-W43	285	245	325
HC-BAROLO-6N	CUST-DIRECT	2025-W43	412	365	459
HC-FIRENZE-7N	CUST-AMAZON	2025-W43	485	420	550
HC-FIRENZE-7N	CUST-MRRET	2025-W43	218	185	251
HC-SIENA-5N	CUST-SEPHORA	2025-W43	345	298	392
GLOSS-ROSE-4N	CUST-MRRET	2025-W43	178	145	211

Step 6: Demand Fusion (Blend Orders + Forecast)

STOX.AI Processing: Fusion Logic

python

```
# Merge orders and forecast
demand_fused = forecast_df.merge(
    orders_aggregated,
    on=['SKU', 'Customer', 'Week'],
    how='outer'
).fillna(0)

# Apply fusion rules based on planning horizon
def apply_fusion_rule(row):
    week_num = int(row['Week'].split('-W')[1])
    current_week = 42  # Example: current week is W42

    horizon = week_num - current_week

    if horizon <= 4:  # Weeks 1-4: Use confirmed orders 100%
        return row['Order_Qty'] if row['Order_Qty'] > 0 else row['Forecast_Qty']
    elif 5 <= horizon <= 12:  # Weeks 5-12: Blend 70% order + 30% forecast
        if row['Order_Qty'] > 0:
            return 0.7 * row['Order_Qty'] + 0.3 * row['Forecast_Qty']
        else:
            return row['Forecast_Qty']
    else:  # Weeks 13+: Use forecast 100%
        return row['Forecast_Qty']

demand_fused['Fused_Demand'] = demand_fused.apply(apply_fusion_rule, axis=1)
demand_fused['Fused_Demand'] = demand_fused['Fused_Demand'].round(0)
```

Output: Demand_Fused.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
Customer	Customer Number	CUST-AMAZON	CUST-MRRET	CUST-SEPHORA
Week	Year-Week	2025-W43	2025-W45	2025-W46
Order_Qty	Confirmed Order Quantity	0	2400	1200
Forecast_Qty	AI Forecast Quantity	625	651	345
Fused_Demand	Final Demand (Blended)	625	2400	1200
Confidence_Lower	Lower Bound	550	2400	1200
Confidence_Upper	Upper Bound	700	2400	1200

Sample Data Rows:

SKU	Customer	Week	Order_Qty	Forecast_Qty	Fused_Demand	Confidence_Lower	Confic
HC-BAROLO-6N	CUST-AMAZON	2025-W43	0	625	625	550	700
HC-BAROLO-6N	CUST-AMAZON	2025-W44	0	638	638	562	714
HC-BAROLO-6N	CUST-AMAZON	2025-W45	2400	651	2400	2400	2400
HC-BAROLO-6N	CUST-MRRET	2025-W47	1200	285	1200	1200	1200
HC-FIRENZE-7N	CUST-AMAZON	2025-W45	1800	495	1800	1800	1800
HC-FIRENZE-7N	CUST-DIRECT	2025-W48	1500	418	1175	1050	1300
HC-SIENA-5N	CUST-SEPHORA	2025-W46	1200	345	1200	1200	1200

(Note: W48 calculation for HC-FIRENZE-7N = 0.7 × 1500 + 0.3 × 418 = 1175)

Step 7: Aggregate to SKU Level for Supply Planning

STOX.AI Processing: Roll Up to SKU-Week

```
python

# Aggregate fused demand across all customers
sku_demand = demand_fused.groupby(['SKU', 'Week']).agg({
    'Fused_Demand': 'sum',
    'Confidence_Lower': 'sum',
    'Confidence_Upper': 'sum'
}).reset_index()

sku_demand.columns = ['SKU', 'Week', 'Total_Demand', 'Lower_Bound', 'Upper_Bound']
```

Output: SKU_Level_Demand.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
Week	Year-Week	2025-W43	2025-W44	2025-W45
Total_Demand	Total Demand Across Customers	1322	1375	4650
Lower_Bound	Aggregated Lower Confidence	1160	1205	4650
Upper_Bound	Aggregated Upper Confidence	1484	1545	4650

Sample Data Rows:

SKU	Week	Total_Demand	Lower_Bound	Upper_Bound
HC-BAROLO-6N	2025-W43	1322	1160	1484
HC-BAROLO-6N	2025-W44	1375	1205	1545
HC-BAROLO-6N	2025-W45	4650	4650	4650
HC-BAROLO-6N	2025-W46	1405	1238	1572
HC-FIRENZE-7N	2025-W43	988	850	1126
HC-FIRENZE-7N	2025-W44	1018	878	1158
HC-FIRENZE-7N	2025-W45	3285	3285	3285
HC-SIENA-5N	2025-W43	685	590	780
GLOSS-ROSE-4N	2025-W43	498	425	571

Step 8: Feed into Supply Planning (Inventory Optimization)

Extract Stock Data

SAP Table: `MARD` (Storage Location Stock)

Table Description: Current inventory levels by material and storage location

Extraction Query:

```sql
```

```sql
SELECT
    MATNR,      -- Material number
    WERKS,      -- Plant code
    LGORT,      -- Storage location
    LABST,      -- Unrestricted-use stock
    INSME,      -- Quality inspection stock
    SPEME       -- Blocked stock
FROM MARD
WHERE WERKS = 'MR01'  -- Main Madison Reed plant
```

Extracted File: Stock_Data.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
MATNR	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
WERKS	Plant Code	MR01	MR01	MR01
LGORT	Storage Location	FG01	FG01	FG01
LABST	Unrestricted Stock (Units)	3250	2850	1950
INSME	Quality Inspection Stock	150	100	75
SPEME	Blocked Stock	0	0	0

Sample Data Rows:

MATNR	WERKS	LGORT	LABST	INSME	SPEME
HC-BAROLO-6N	MR01	FG01	3250	150	0
HC-FIRENZE-7N	MR01	FG01	2850	100	0
HC-SIENA-5N	MR01	FG01	1950	75	0
HC-MILANO-8N	MR01	FG01	1650	50	0
HC-ROMA-4N	MR01	FG01	1850	80	0
GLOSS-ROSE-4N	MR01	FG01	950	40	0
ROOT-TOUCH-DARK	MR01	FG01	1250	60	0
CONDITIONER-REP	MR01	FG01	2450	100	0

SAP Table: VBBE (Sales Requirements / Committed Stock)

Table Description: Stock quantities already committed to specific sales orders

Extraction Query:

```sql
SELECT
    MATNR,      -- Material number
    WERKS,      -- Plant
    VMENG,      -- Committed quantity
    BDTER       -- Requirement date
FROM VBBE
WHERE BDTER >= '20251021'
  AND VMENG > 0
```

Extracted File: Committed_Stock.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
MATNR	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
WERKS	Plant Code	MR01	MR01	MR01
VMENG	Committed Quantity (Units)	2400	1800	1200
BDTER	Requirement Date (YYYYMMDD)	20251105	20251105	20251112

Sample Data Rows:

MATNR	WERKS	VMENG	BDTER
HC-BAROLO-6N	MR01	2400	20251105
HC-FIRENZE-7N	MR01	1800	20251105
HC-SIENA-5N	MR01	1200	20251112
GLOSS-ROSE-4N	MR01	800	20251119

STOX.AI Processing: Calculate Available Inventory

python

```python
# Load stock and committed data
stock_data = pd.read_csv('Stock_Data.csv')
committed_stock = pd.read_csv('Committed_Stock.csv')

# Calculate available stock
committed_agg = committed_stock.groupby('MATNR')['VMENG'].sum().reset_index()
committed_agg.columns = ['SKU', 'Committed_Qty']

inventory = stock_data.merge(committed_agg, left_on='MATNR', right_on='SKU', how='left').fillna(0)
inventory['Available_Stock'] = inventory['LABST'] - inventory['INSME'] - inventory['Committed_Qty']
inventory = inventory[['MATNR', 'LABST', 'INSME', 'Committed_Qty', 'Available_Stock']]
inventory.columns = ['SKU', 'On_Hand', 'Quality_Hold', 'Committed', 'Available']
```

Output: Available_Inventory.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
On_Hand	Unrestricted Stock	3250	2850	1950
Quality_Hold	Stock in QC Inspection	150	100	75
Committed	Stock Allocated to Orders	2400	1800	1200
Available	Available for Planning	700	950	675

Sample Data Rows:

SKU	On_Hand	Quality_Hold	Committed	Available
HC-BAROLO-6N	3250	150	2400	700
HC-FIRENZE-7N	2850	100	1800	950
HC-SIENA-5N	1950	75	1200	675
HC-MILANO-8N	1650	50	0	1600
GLOSS-ROSE-4N	950	40	800	110
ROOT-TOUCH-DARK	1250	60	0	1190

STOX.AI Processing: Calculate Net Requirements

```
python
```

```
# Merge demand with inventory
supply_plan = sku_demand.merge(inventory[['SKU', 'Available']], on='SKU', how='left')

# Calculate net requirement
supply_plan['Net_Requirement'] = supply_plan['Total_Demand'] - supply_plan['Available']
supply_plan['Net_Requirement'] = supply_plan['Net_Requirement'].clip(lower=0)

# Calculate safety stock (using volatility from confidence intervals)
supply_plan['Demand_Volatility'] = (supply_plan['Upper_Bound'] - supply_plan['Lower_Bound']) / supply_plan['Total
supply_plan['Safety_Stock'] = (supply_plan['Total_Demand'] * supply_plan['Demand_Volatility'] * 1.65).round(0)  # 2

supply_plan = supply_plan[['SKU', 'Week', 'Total_Demand', 'Available', 'Net_Requirement', 'Safety_Stock']]
```

Output: Supply_Plan.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
Week	Planning Year-Week	2025-W43	2025-W44	2025-W45
Total_Demand	Total Demand (All Customers)	1322	1018	4650
Available	Available Inventory	700	950	675
Net_Requirement	Production/Purchase Need	622	68	3975
Safety_Stock	Recommended Safety Stock	267	240	0

Sample Data Rows:

SKU	Week	Total_Demand	Available	Net_Requirement	Safety_Stock
HC-BAROLO-6N	2025-W43	1322	700	622	267
HC-BAROLO-6N	2025-W44	1375	78	1297	291
HC-BAROLO-6N	2025-W45	4650	0	4650	0
HC-FIRENZE-7N	2025-W43	988	950	38	238
HC-FIRENZE-7N	2025-W44	1018	912	106	246
HC-SIENA-5N	2025-W43	685	675	10	163
GLOSS-ROSE-4N	2025-W43	498	110	388	126

2️⃣ DRILL-DOWN CAPABILITY: BOM Explosion & Component Consolidation

Step 1: Extract BOM Structure

SAP Table: `STKO` (BOM Header)

Table Description: Bill of Material header data with validity dates and alternatives

Extraction Query:

```sql
SELECT
    STLNR,      -- BOM number
    STLAL,      -- Alternative BOM indicator
    DATUV       -- Valid-from date
FROM STKO
WHERE DATUV <= CURRENT_DATE
```

Extracted File: BOM_Header.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
STLNR	BOM Number (Internal ID)	BOM-BAROLO	BOM-FIRENZE	BOM-SIENA
STLAL	Alternative BOM Indicator	01	01	01
DATUV	Valid-From Date (YYYYMMDD)	20250101	20250101	20250101

Sample Data Rows:

STLNR	STLAL	DATUV
BOM-BAROLO	01	20250101
BOM-FIRENZE	01	20250101
BOM-SIENA	01	20250101
BOM-MILANO	01	20250101
BOM-ROMA	01	20250101
BOM-GLOSS-ROSE	01	20250101
BOM-ROOT-TOUCH	01	20250101

SAP Table: STPO (BOM Items / Components)

Table Description: Bill of Material component items with quantities and scrap factors

Extraction Query:

```
sql
```

```
SELECT
    STLNR,      -- BOM number
    STLAL,      -- Alternative BOM
    POSNR,      -- Item position number
    IDNRK,      -- Component material number
    MENGE,      -- Component quantity per assembly
    MEINS,      -- Unit of measure
    AUSCH       -- Component scrap percentage
FROM STPO
ORDER BY STLNR, POSNR
```

Extracted File: BOM_Items.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
STLNR	BOM Number	BOM-BAROLO	BOM-BAROLO	BOM-BAROLO
STLAL	Alternative BOM	01	01	01
POSNR	Item Position Number	0010	0020	0030
IDNRK	Component Material Number	COMP-BASE-DARK	COMP-DEVELOPER-20	COMP-CONDITIONER
MENGE	Component Quantity	1.00	1.00	1.00
MEINS	Unit of Measure	EA	EA	EA
AUSCH	Component Scrap %	2.0	1.5	1.0

Sample Data Rows:

STLNR	STLAL	POSNR	IDNRK	MENGE	MEINS	AUSCH
BOM-BAROLO	01	0010	COMP-BASE-DARK	1.00	EA	2.0
BOM-BAROLO	01	0020	COMP-DEVELOPER-20	1.00	EA	1.5
BOM-BAROLO	01	0030	COMP-CONDITIONER	1.00	EA	1.0
BOM-BAROLO	01	0040	COMP-GLOVES	1.00	EA	0.0
BOM-BAROLO	01	0050	COMP-BOX-PREMIUM	1.00	EA	0.5
BOM-FIRENZE	01	0010	COMP-BASE-LIGHT	1.00	EA	2.0
BOM-FIRENZE	01	0020	COMP-DEVELOPER-20	1.00	EA	1.5
BOM-FIRENZE	01	0030	COMP-CONDITIONER	1.00	EA	1.0
BOM-FIRENZE	01	0040	COMP-GLOVES	1.00	EA	0.0
BOM-FIRENZE	01	0050	COMP-BOX-PREMIUM	1.00	EA	0.5
BOM-SIENA	01	0010	COMP-BASE-MED	1.00	EA	2.0
BOM-SIENA	01	0020	COMP-DEVELOPER-20	1.00	EA	1.5
BOM-GLOSS-ROSE	01	0010	COMP-GLOSS-BASE	1.00	EA	2.0
BOM-GLOSS-ROSE	01	0020	COMP-CONDITIONER	1.00	EA	1.0

SAP Table: MAST (Material to BOM Link)

Table Description: Links finished goods (materials) to their bill of materials

Extraction Query:

```sql
```

```
SELECT
    MATNR,      -- Material number (finished good)
    WERKS,      -- Plant
    STLNR,      -- BOM number
    STLAL       -- Alternative BOM
FROM MAST
WHERE WERKS = 'MR01'
```

Extracted File: Material_BOM_Link.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
MATNR	Material Number (Finished Good)	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
WERKS	Plant Code	MR01	MR01	MR01
STLNR	BOM Number	BOM-BAROLO	BOM-FIRENZE	BOM-SIENA
STLAL	Alternative BOM	01	01	01

Sample Data Rows:

MATNR	WERKS	STLNR	STLAL
HC-BAROLO-6N	MR01	BOM-BAROLO	01
HC-FIRENZE-7N	MR01	BOM-FIRENZE	01
HC-SIENA-5N	MR01	BOM-SIENA	01
HC-MILANO-8N	MR01	BOM-MILANO	01
HC-ROMA-4N	MR01	BOM-ROMA	01
GLOSS-ROSE-4N	MR01	BOM-GLOSS-ROSE	01
ROOT-TOUCH-DARK	MR01	BOM-ROOT-TOUCH	01

Step 2: Link FG Demand to BOM

STOX.AI Processing: Join Demand with BOM

```
python

# Load BOM data
bom_header = pd.read_csv('BOM_Header.csv')
bom_items = pd.read_csv('BOM_Items.csv')
material_bom_link = pd.read_csv('Material_BOM_Link.csv')

# Load supply plan (FG demand)
supply_plan = pd.read_csv('Supply_Plan.csv')

# Join FG with BOM
fg_with_bom = supply_plan.merge(
    material_bom_link[['MATNR', 'STLNR', 'STLAL']],
    left_on='SKU',
    right_on='MATNR',
    how='left'
)

fg_with_bom = fg_with_bom[['SKU', 'Week', 'Net_Requirement', 'STLNR', 'STLAL']]
```

Output: FG_With_BOM.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
SKU	Finished Good Material Number	HC-BAROLO-6N	HC-FIRENZE-7N	HC-SIENA-5N
Week	Planning Week	2025-W43	2025-W43	2025-W43
Net_Requirement	FG Production Requirement	622	38	10
STLNR	BOM Number	BOM-BAROLO	BOM-FIRENZE	BOM-SIENA
STLAL	Alternative BOM	01	01	01

Sample Data Rows:

SKU	Week	Net_Requirement	STLNR	STLAL
HC-BAROLO-6N	2025-W43	622	BOM-BAROLO	01
HC-BAROLO-6N	2025-W44	1297	BOM-BAROLO	01
HC-BAROLO-6N	2025-W45	4650	BOM-BAROLO	01
HC-FIRENZE-7N	2025-W43	38	BOM-FIRENZE	01
HC-FIRENZE-7N	2025-W44	106	BOM-FIRENZE	01
HC-SIENA-5N	2025-W43	10	BOM-SIENA	01
GLOSS-ROSE-4N	2025-W43	388	BOM-GLOSS-ROSE	01

Step 3: Explode BOM to Component Level

STOX.AI Processing: Calculate Gross Component Requirements

```
python
```

```
# Join FG demand with BOM items
component_requirements = fg_with_bom.merge(
    bom_items,
    on=['STLNR', 'STLAL'],
    how='left'
)

# Calculate gross component requirement (including scrap)
component_requirements['Gross_Comp_Req'] = (
    component_requirements['Net_Requirement'] *
    component_requirements['MENGE'] *
    (1 + component_requirements['AUSCH'] / 100)
)

component_requirements = component_requirements[
    ['SKU', 'Week', 'Net_Requirement', 'IDNRK', 'MENGE', 'AUSCH', 'Gross_Comp_Req']
]
component_requirements.columns = [
    'Finished_Good', 'Week', 'FG_Requirement', 'Component', 'BOM_Qty', 'Scrap_Pct', 'Gross_Comp_Req'
]
component_requirements['Gross_Comp_Req'] = component_requirements['Gross_Comp_Req'].round(0)
```

Output: Component_Requirements_Detailed.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
Finished_Good	FG Material Number	HC-BAROLO-6N	HC-BAROLO-6N	HC-FIRENZE-7N
Week	Planning Week	2025-W43	2025-W43	2025-W43
FG_Requirement	FG Production Quantity	622	622	38
Component	Component Material Number	COMP-BASE-DARK	COMP-DEVELOPER-20	COMP-BASE-LIGHT
BOM_Qty	Component Qty per FG	1.00	1.00	1.00
Scrap_Pct	Scrap Allowance %	2.0	1.5	2.0
Gross_Comp_Req	Total Component Need (w/ scrap)	634	631	39

Sample Data Rows:

Finished_Good	Week	FG_Requirement	Component	BOM_Qty	Scrap_Pct	Gross_Comp_Req
HC-BAROLO-6N	2025-W43	622	COMP-BASE-DARK	1.00	2.0	634
HC-BAROLO-6N	2025-W43	622	COMP-DEVELOPER-20	1.00	1.5	631
HC-BAROLO-6N	2025-W43	622	COMP-CONDITIONER	1.00	1.0	628
HC-BAROLO-6N	2025-W43	622	COMP-GLOVES	1.00	0.0	622
HC-BAROLO-6N	2025-W43	622	COMP-BOX-PREMIUM	1.00	0.5	625
HC-FIRENZE-7N	2025-W43	38	COMP-BASE-LIGHT	1.00	2.0	39
HC-FIRENZE-7N	2025-W43	38	COMP-DEVELOPER-20	1.00	1.5	39
HC-FIRENZE-7N	2025-W43	38	COMP-CONDITIONER	1.00	1.0	38
HC-SIENA-5N	2025-W43	10	COMP-BASE-MED	1.00	2.0	10
HC-SIENA-5N	2025-W43	10	COMP-DEVELOPER-20	1.00	1.5	10
GLOSS-ROSE-4N	2025-W43	388	COMP-GLOSS-BASE	1.00	2.0	396

Finished_Good	Week	FG_Requirement	Component	BOM_Qty	Scrap_Pct	Gross_Comp_Req
GLOSS-ROSE-4N	2025-W43	388	COMP-CONDITIONER	1.00	1.0	392

Step 4: Consolidate Components Across Multiple FGs

STOX.AI Processing: Aggregate Component Demand

python

```
# Aggregate component requirements across all finished goods and weeks
component_consolidated = component_requirements.groupby(['Component', 'Week']).agg({
   'Gross_Comp_Req': 'sum'
}).reset_index()

component_consolidated.columns = ['Component', 'Week', 'Total_Requirement']
component_consolidated['Total_Requirement'] = component_consolidated['Total_Requirement'].round(0)

# Count number of parent FGs using each component
parent_count = component_requirements.groupby('Component')['Finished_Good'].nunique().reset_index()
parent_count.columns = ['Component', 'Used_In_FGs']

component_consolidated = component_consolidated.merge(parent_count, on='Component', how='left')
```

Output: Component_Consolidated.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
Component	Component Material Number	COMP-DEVELOPER-20	COMP-CONDITIONER	COMP-BASE-DARK
Week	Planning Week	2025-W43	2025-W43	2025-W43
Total_Requirement	Consolidated Component Demand	680	1058	634
Used_In_FGs	Number of Parent FGs Using Component	5	6	1

Sample Data Rows:

Component	Week	Total_Requirement	Used_In_FGs
COMP-DEVELOPER-20	2025-W43	680	5
COMP-DEVELOPER-20	2025-W44	1423	5
COMP-DEVELOPER-20	2025-W45	4745	5
COMP-CONDITIONER	2025-W43	1058	6
COMP-CONDITIONER	2025-W44	1513	6
COMP-CONDITIONER	2025-W45	4798	6
COMP-BASE-DARK	2025-W43	634	1
COMP-BASE-DARK	2025-W44	1323	1
COMP-BASE-LIGHT	2025-W43	39	1
COMP-BASE-MED	2025-W43	10	1
COMP-GLOSS-BASE	2025-W43	396	1
COMP-GLOVES	2025-W43	660	5
COMP-BOX-PREMIUM	2025-W43	663	5

Step 5: Link Components to Suppliers

SAP Table: `EORD` (Source List / Purchasing Source)

Table Description: Defines approved suppliers for each component material

Extraction Query:

```sql
SELECT
    MATNR,      -- Material (component)
    WERKS,      -- Plant
    LIFNR,      -- Vendor number
    VDATU,      -- Valid-from date
    BDATU       -- Valid-to date
FROM EORD
WHERE BDATU >= CURRENT_DATE OR BDATU = '99991231'
  AND WERKS = 'MR01'
```

Extracted File: Source_List.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
MATNR	Component Material Number	COMP-BASE-DARK	COMP-DEVELOPER-20	COMP-CONDITIONER
WERKS	Plant Code	MR01	MR01	MR01
LIFNR	Vendor Number	SUPP-CHEM-01	SUPP-CHEM-01	SUPP-CARE-05
VDATU	Valid-From Date (YYYYMMDD)	20250101	20250101	20250101
BDATU	Valid-To Date (YYYYMMDD)	99991231	99991231	99991231

Sample Data Rows:

MATNR	WERKS	LIFNR	VDATU	BDATU
COMP-BASE-DARK	MR01	SUPP-CHEM-01	20250101	99991231
COMP-BASE-LIGHT	MR01	SUPP-CHEM-01	20250101	99991231
COMP-BASE-MED	MR01	SUPP-CHEM-01	20250101	99991231
COMP-GLOSS-BASE	MR01	SUPP-CHEM-01	20250101	99991231
COMP-DEVELOPER-20	MR01	SUPP-CHEM-01	20250101	99991231
COMP-CONDITIONER	MR01	SUPP-CARE-05	20250101	99991231
COMP-GLOVES	MR01	SUPP-PKG-12	20250101	99991231
COMP-BOX-PREMIUM	MR01	SUPP-PKG-12	20250101	99991231

SAP Table: LFA1 (Vendor Master - General Data)

Table Description: Master data for supplier/vendor information

Extraction Query:

```
sql
```

```
SELECT
    LIFNR,      -- Vendor number
    NAME1,      -- Vendor name
    LAND1       -- Country code
FROM LFA1
WHERE LIFNR IN (SELECT DISTINCT LIFNR FROM EORD WHERE WERKS = 'MR01')
```

Extracted File: Vendor_Master.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
LIFNR	Vendor Number	SUPP-CHEM-01	SUPP-CARE-05	SUPP-PKG-12
NAME1	Vendor Name	ChromaTech Ingredients LLC	HairCare Solutions Inc	Premium Packaging Co
LAND1	Country Code	US	US	CN

Sample Data Rows:

LIFNR	NAME1	LAND1
SUPP-CHEM-01	ChromaTech Ingredients LLC	US
SUPP-CARE-05	HairCare Solutions Inc	US
SUPP-PKG-12	Premium Packaging Co	CN

STOX.AI Processing: Link Components to Suppliers

```
python
```

```
# Load source list and vendor master
source_list = pd.read_csv('Source_List.csv')
vendor_master = pd.read_csv('Vendor_Master.csv')

# Join component requirements with supplier info
component_with_supplier = component_consolidated.merge(
    source_list[['MATNR', 'LIFNR']],
    left_on='Component',
    right_on='MATNR',
    how='left'
).drop('MATNR', axis=1)

component_with_supplier = component_with_supplier.merge(
    vendor_master[['LIFNR', 'NAME1', 'LAND1']],
    on='LIFNR',
    how='left'
)

component_with_supplier = component_with_supplier[
    ['Component', 'Week', 'Total_Requirement', 'Used_In_FGs', 'LIFNR', 'NAME1', 'LAND1']
]
component_with_supplier.columns = [
    'Component', 'Week', 'Total_Requirement', 'Used_In_FGs', 'Supplier_ID', 'Supplier_Name', 'Country'
]
```

Output: Component_With_Supplier.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
Component	Component Material Number	COMP-DEVELOPER-20	COMP-CONDITIONER	COMP-BASE-DARK
Week	Planning Week	2025-W43	2025-W43	2025-W43
Total_Requirement	Consolidated Demand	680	1058	634
Used_In_FGs	Parent FG Count	5	6	1
Supplier_ID	Vendor Number	SUPP-CHEM-01	SUPP-CARE-05	SUPP-CHEM-01
Supplier_Name	Vendor Name	ChromaTech Ingredients LLC	HairCare Solutions Inc	ChromaTech Ingredients LLC
Country	Supplier Country	US	US	US

Sample Data Rows:

Component	Week	Total_Requirement	Used_In_FGs	Supplier_ID	Supplier_Name	Country
COMP-DEVELOPER-20	2025-W43	680	5	SUPP-CHEM-01	ChromaTech Ingredients LLC	US
COMP-DEVELOPER-20	2025-W44	1423	5	SUPP-CHEM-01	ChromaTech Ingredients LLC	US
COMP-DEVELOPER-20	2025-W45	4745	5	SUPP-CHEM-01	ChromaTech Ingredients LLC	US
COMP-CONDITIONER	2025-W43	1058	6	SUPP-CARE-05	HairCare Solutions Inc	US
COMP-CONDITIONER	2025-W44	1513	6	SUPP-CARE-05	HairCare Solutions Inc	US
COMP-BASE-DARK	2025-W43	634	1	SUPP-CHEM-01	ChromaTech Ingredients LLC	US
COMP-BASE-LIGHT	2025-W43	39	1	SUPP-CHEM-01	ChromaTech Ingredients LLC	US
COMP-GLOVES	2025-W43	660	5	SUPP-PKG-12	Premium Packaging Co	CN
COMP-BOX-PREMIUM	2025-W43	663	5	SUPP-PKG-12	Premium Packaging Co	CN

Step 6: Aggregate by Supplier (Consolidated Procurement Plan)

STOX.AI Processing: Consolidate by Supplier

```
python

# Aggregate component requirements by supplier and week
supplier_consolidated = component_with_supplier.groupby(['Supplier_ID', 'Supplier_Name', 'Country', 'Week']).agg({
    'Total_Requirement': 'sum',
    'Component': 'count'
}).reset_index()

supplier_consolidated.columns = ['Supplier_ID', 'Supplier_Name', 'Country', 'Week', 'Total_Units', 'Unique_Componer
```

Output: Supplier_Consolidated.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
Supplier_ID	Vendor Number	SUPP-CHEM-01	SUPP-CARE-05	SUPP-PKG-12
Supplier_Name	Vendor Name	ChromaTech Ingredients LLC	HairCare Solutions Inc	Premium Packaging Co
Country	Supplier Country	US	US	CN
Week	Planning Week	2025-W43	2025-W43	2025-W43
Total_Units	Total Units from Supplier	1363	1058	1323
Unique_Components	Number of Different Components	5	1	2

Sample Data Rows:

Supplier_ID	Supplier_Name	Country	Week	Total_Units	Unique_Components
SUPP-CHEM-01	ChromaTech Ingredients LLC	US	2025-W43	1363	5
SUPP-CHEM-01	ChromaTech Ingredients LLC	US	2025-W44	2956	5
SUPP-CHEM-01	ChromaTech Ingredients LLC	US	2025-W45	9490	5
SUPP-CARE-05	HairCare Solutions Inc	US	2025-W43	1058	1
SUPP-CARE-05	HairCare Solutions Inc	US	2025-W44	1513	1
SUPP-CARE-05	HairCare Solutions Inc	US	2025-W45	4798	1
SUPP-PKG-12	Premium Packaging Co	CN	2025-W43	1323	2
SUPP-PKG-12	Premium Packaging Co	CN	2025-W44	2786	2

Step 7: Create Purchase Requisitions

Extract Lead Time Data

SAP Table: MARC (Plant Material Status)

Table Description: Plant-specific material data including procurement lead times

Extraction Query:

```sql
```

```sql
SELECT
    MATNR,      -- Material number
    WERKS,      -- Plant
    PLIFZ,      -- Planned delivery time (days)
    WEBAZ,      -- Goods receipt processing time (days)
    EISBE       -- Safety stock quantity
FROM MARC
WHERE WERKS = 'MR01'
  AND MATNR LIKE 'COMP-%'
```

Extracted File: Lead_Times.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
MATNR	Component Material Number	COMP-BASE-DARK	COMP-DEVELOPER-20	COMP-CONDITIONER
WERKS	Plant Code	MR01	MR01	MR01
PLIFZ	Planned Delivery Time (Days)	14	14	21
WEBAZ	GR Processing Time (Days)	2	2	3
EISBE	Safety Stock (Units)	800	1200	1500

Sample Data Rows:

MATNR	WERKS	PLIFZ	WEBAZ	EISBE
COMP-BASE-DARK	MR01	14	2	800
COMP-BASE-LIGHT	MR01	14	2	600
COMP-BASE-MED	MR01	14	2	700
COMP-GLOSS-BASE	MR01	14	2	500
COMP-DEVELOPER-20	MR01	14	2	1200
COMP-CONDITIONER	MR01	21	3	1500
COMP-GLOVES	MR01	35	2	2000
COMP-BOX-PREMIUM	MR01	42	3	1800

STOX.AI Processing: Calculate Order Dates

```python
```

```
from datetime import datetime, timedelta

# Load lead times
lead_times = pd.read_csv('Lead_Times.csv')

# Join component requirements with lead times
pr_data = component_with_supplier.merge(
    lead_times[['MATNR', 'PLIFZ', 'WEBAZ']],
    left_on='Component',
    right_on='MATNR',
    how='left'
).drop('MATNR', axis=1)

# Calculate order date (need date - lead time)
def calculate_order_date(week_str, lead_time_days):
    # Convert week string to date
    year_week = week_str.split('-W')
    year = int(year_week[0])
    week = int(year_week[1])

    need_date = datetime.strptime(f'{year}-W{week}-1', '%Y-W%W-%w')
    order_date = need_date - timedelta(days=lead_time_days)

    return order_date.strftime('%Y%m%d')

pr_data['Total_Lead_Time'] = pr_data['PLIFZ'] + pr_data['WEBAZ']
pr_data['Order_Date'] = pr_data.apply(
    lambda row: calculate_order_date(row['Week'], row['Total_Lead_Time']),
    axis=1
)

pr_data = pr_data[
    ['Component', 'Supplier_ID', 'Supplier_Name', 'Total_Requirement', 'Order_Date', 'Week']
```

```
]
pr_data.columns = ['Material', 'Vendor', 'Vendor_Name', 'Quantity', 'Order_Date', 'Delivery_Week']
```

Output: Purchase_Requisitions.csv

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
Material	Component Material Number	COMP-DEVELOPER-20	COMP-CONDITIONER	COMP-BASE-DARK
Vendor	Vendor Number	SUPP-CHEM-01	SUPP-CARE-05	SUPP-CHEM-01
Vendor_Name	Vendor Name	ChromaTech Ingredients LLC	HairCare Solutions Inc	ChromaTech Ingredients LLC
Quantity	Order Quantity (Units)	680	1058	634
Order_Date	Required Order Date (YYYYMMDD)	20251008	20251001	20251008
Delivery_Week	Target Delivery Week	2025-W43	2025-W43	2025-W43

Sample Data Rows:

Material	Vendor	Vendor_Name	Quantity	Order_Date	Delivery_Week
COMP-DEVELOPER-20	SUPP-CHEM-01	ChromaTech Ingredients LLC	680	20251008	2025-W43
COMP-DEVELOPER-20	SUPP-CHEM-01	ChromaTech Ingredients LLC	1423	20251015	2025-W44
COMP-DEVELOPER-20	SUPP-CHEM-01	ChromaTech Ingredients LLC	4745	20251022	2025-W45
COMP-CONDITIONER	SUPP-CARE-05	HairCare Solutions Inc	1058	20251001	2025-W43
COMP-CONDITIONER	SUPP-CARE-05	HairCare Solutions Inc	1513	20251008	2025-W44
COMP-BASE-DARK	SUPP-CHEM-01	ChromaTech Ingredients LLC	634	20251008	2025-W43
COMP-GLOVES	SUPP-PKG-12	Premium Packaging Co	660	20250926	2025-W43
COMP-BOX-PREMIUM	SUPP-PKG-12	Premium Packaging Co	663	20250919	2025-W43

Step 8: Generate Upload File for SAP

STOX.AI Processing: Format for BAPI_PR_CREATE

```
python
```

```
# Add required SAP fields
pr_upload = pr_data.copy()
pr_upload['PR_TYPE'] = 'NB'  # Standard PR
pr_upload['PLANT'] = 'MR01'
pr_upload['PURCH_ORG'] = '1000'
pr_upload['PURCH_GROUP'] = '001'
pr_upload['MATERIAL_GROUP'] = 'COMP'
pr_upload['UNIT'] = 'EA'
pr_upload['PRICE_UNIT'] = 1
pr_upload['CURRENCY'] = 'USD'

# Reorder columns for SAP upload template
pr_upload = pr_upload[[
    'PR_TYPE', 'Material', 'PLANT', 'Quantity', 'UNIT',
    'Order_Date', 'Delivery_Week', 'Vendor', 'PURCH_ORG',
    'PURCH_GROUP', 'MATERIAL_GROUP', 'PRICE_UNIT', 'CURRENCY'
]]

pr_upload.columns = [
    'PR_TYPE', 'MATERIAL', 'PLANT', 'QUANTITY', 'UNIT',
    'DELIVERY_DATE', 'DELIVERY_WEEK', 'VENDOR', 'PURCH_ORG',
    'PURCH_GROUP', 'MAT_GROUP', 'PRICE_UNIT', 'CURRENCY'
]

# Save for SAP upload
pr_upload.to_excel('SAP_Upload_PR_Create.xlsx', index=False)
```

Output: `SAP_Upload_PR_Create.xlsx` (ready for BAPI_PR_CREATE or ME51N)

Field	Description	Sample Value 1	Sample Value 2	Sample Value 3
PR_TYPE	Purchase Requisition Type	NB	NB	NB
MATERIAL	Material Number	COMP-DEVELOPER-20	COMP-CONDITIONER	COMP-BASE-DARK
PLANT	Plant Code	MR01	MR01	MR01
QUANTITY	Order Quantity	680	1058	634
UNIT	Unit of Measure	EA	EA	EA
DELIVERY_DATE	Delivery Date (YYYYMMDD)	20251024	20251024	20251024
DELIVERY_WEEK	Reference Delivery Week	2025-W43	2025-W43	2025-W43
VENDOR	Preferred Vendor	SUPP-CHEM-01	SUPP-CARE-05	SUPP-CHEM-01
PURCH_ORG	Purchasing Organization	1000	1000	1000
PURCH_GROUP	Purchasing Group	001	001	001

Sample Data Rows:

PR_TYPE	MATERIAL	PLANT	QUANTITY	UNIT	DELIVERY_DATE	DELIVERY_WEEK
NB	COMP-DEVELOPER-20	MR01	680	EA	20251024	2025-W43
NB	COMP-DEVELOPER-20	MR01	1423	EA	20251031	2025-W44
NB	COMP-CONDITIONER	MR01	1058	EA	20251024	2025-W43
NB	COMP-BASE-DARK	MR01	634	EA	20251024	2025-W43
NB	COMP-GLOVES	MR01	660	EA	20251024	2025-W43

📊 SUMMARY: Complete Data Flow for Madison Reed

Input Data (from SAP S/4HANA Retail)

1. `POS_Sales.csv` (WPOS_SALES) - 18,000 POS transactions
2. `Store_Master.csv` (WRS1) - 7 channels (4 retail + 3 e-commerce)
3. `Sales_Orders.csv` (VBAP) - Future customer orders
4. `Stock_Data.csv` (MARD) - Current inventory at MR01 plant
5. `Committed_Stock.csv` (VBBE) - Reserved order quantities
6. `BOM_Header.csv` (STKO) - 7 product BOMs
7. `BOM_Items.csv` (STPO) - 30+ component relationships

8. `Material_BOM_Link.csv` (MAST) - FG-to-BOM mapping
9. `Source_List.csv` (EORD) - Component-to-supplier assignments
10. `Vendor_Master.csv` (LFA1) - 3 suppliers
11. `Lead_Times.csv` (MARC) - Procurement lead times

Processing in STOX.AI

- POS aggregation: Transaction → SKU-Channel-Week
- Customer mapping: Channel → Customer (Amazon, Sephora, Direct)
- AI forecasting: 26-week horizon with Prophet
- Demand fusion: Orders (100%) → Blended (70/30) → Forecast (100%)
- Inventory calculation: Available = On-hand - QC - Committed
- BOM explosion: FG → Components with scrap allowance
- Component consolidation: **COMP-DEVELOPER-20 used in 5 FGs** → Single order
- Supplier linkage: **3 suppliers, 8 unique components**
- PR generation: Calculate order dates based on lead time

Output Data (to SAP)

1. `Supply_Plan.csv` - FG net requirements by week
2. `Component_Consolidated.csv` - Component demand (8 SKUs consolidated)
3. `Supplier_Consolidated.csv` - Supplier-level plan (3 suppliers)
4. `Purchase_Requisitions.csv` - Detailed PR recommendations
5. `SAP_Upload_PR_Create.xlsx` - Ready for ME51N/BAPI upload

🔄 Key Business Insights: Madison Reed Example

1. Multi-Channel Demand Intelligence

✅ **Amazon drives 45% of volume** - Highest sell-through velocity
✅ **Retail stores have 30% lower volatility** - More predictable
✅ **Direct website = premium margin channel** - Prioritize stock availability
✅ **Sephora = new channel** - Building forecast confidence

2. Component Consolidation Power

✅ **COMP-DEVELOPER-20**: Used in 5 hair color shades

- Individual FG orders: 5 separate PRs
- STOX.AI: 1 consolidated order to ChromaTech for 680 units (W43)
- **Negotiation leverage**: Bulk discount opportunity

✅ **COMP-CONDITIONER**: Used in 6 products (all hair colors)

- Highest consolidation benefit
- Single supplier (HairCare Solutions)
- **Safety stock optimization**: Shared component = lower total SS

✅ **COMP-GLOVES & COMP-BOX-PREMIUM**: Packaging from China

- Long lead time (35-42 days)
- Order placed 6 weeks ahead
- **Risk mitigation**: Early ordering reduces expedite costs

3. Lead Time Management

Supplier	Lead Time	Strategy
ChromaTech (US)	16 days	2-week order cycle
HairCare (US)	24 days	3-week order cycle
Premium Pkg (China)	42-45 days	6-week + buffer stock

Madison Reed can now plan intelligently across channels, optimize component procurement, and reduce working capital by 10-15% using STOX.AI!