-- Step 1: Create a rolling 7-day average of units sold per SKU and store

WITH RollingAvg AS (

SELECT

SKU,

Store\_ID,

Date,

AVG(Units\_Sold) OVER (

PARTITION BY SKU, Store\_ID

ORDER BY Date

ROWS BETWEEN 6 PRECEDING AND CURRENT ROW

) AS RollingAvgUnitsSold

FROM sample\_sales\_data

)

-- Step 2: Join with inventory and supplier data

, InventoryContext AS (

SELECT

i.SKU,

i.Store\_ID,

i.Inventory\_Level,

i.Safety\_Stock,

s.Avg\_Lead\_Time\_Days,

s.Delivery\_Reliability,

r.RollingAvgUnitsSold

FROM inventory\_levels i

LEFT JOIN supplier\_lead\_times s ON i.SKU = s.SKU

LEFT JOIN RollingAvg r ON i.SKU = r.SKU AND i.Store\_ID = r.Store\_ID

)

-- Step 3: Forecast inventory need and classify risk

SELECT

SKU,

Store\_ID,

Inventory\_Level,

Safety\_Stock,

RollingAvgUnitsSold,

Avg\_Lead\_Time\_Days,

Delivery\_Reliability,

CASE

WHEN Inventory\_Level < Safety\_Stock AND Avg\_Lead\_Time\_Days > 7 AND Delivery\_Reliability < 0.85 THEN 'High'

WHEN Inventory\_Level < Safety\_Stock AND Avg\_Lead\_Time\_Days > 5 THEN 'Medium'

WHEN Inventory\_Level < Safety\_Stock THEN 'Low'

ELSE 'Stable'

END AS InventoryRiskScore,

CASE

WHEN Inventory\_Level < (RollingAvgUnitsSold \* Avg\_Lead\_Time\_Days) THEN 'Forecasted Stockout'

ELSE 'Sufficient'

END AS ForecastStatus

FROM InventoryContext

ORDER BY InventoryRiskScore DESC, ForecastStatus DESC;