

```

In [ ]: import pandas as pd

# Load input data
products_df = pd.read_csv('products.csv')
sales_df = pd.read_csv('sales.csv')

# Merge product and sales data on 'sku'
merged_df = pd.merge(products_df, sales_df, on='sku', how='left')

# Ensure quantity_sold is numeric and fill missing with 0
merged_df['quantity_sold'] = pd.to_numeric(merged_df['quantity_sold'], errors='coerce').fillna(0)

def apply_pricing_rules(row):
    current_price = row['current_price']
    cost_price = row['cost_price']
    stock = row['stock']
    quantity_sold = row['quantity_sold']
    new_price = current_price

    # Rule 1 – Low Stock, High Demand
    if stock < 20 and quantity_sold > 30:
        new_price = current_price * 1.15
    # Rule 2 – Dead Stock
    elif stock > 200 and quantity_sold == 0:
        new_price = current_price * 0.70
    # Rule 3 – Overstocked Inventory
    elif stock > 100 and quantity_sold < 20:
        new_price = current_price * 0.90

    # Rule 4 – Minimum Profit Constraint
    min_allowed_price = cost_price * 1.2
    if new_price < min_allowed_price:
        new_price = min_allowed_price

    # Final rounding
    return round(current_price, 2), round(new_price, 2)

# Apply rules to each row
merged_df[['old_price', 'new_price']] = merged_df.apply(apply_pricing_rules, axis=1, result_type='expand')

# Add $ units
merged_df['old_price'] = merged_df['old_price'].apply(lambda x: f"${x:.2f}")
merged_df['new_price'] = merged_df['new_price'].apply(lambda x: f"${x:.2f}")

# Save to output CSV
output_df = merged_df[['sku', 'old_price', 'new_price']]
output_df.to_csv('updated_prices.csv', index=False)

print("updated_prices.csv has been created.")

```