Walmart Data Munging Script - Task 4

Author: S. Rahul
Purpose: Populate SQLite database with shipping data from CSV files

This script performs ETL operations:

• Extract: Read data from three CSV files
• Transform: Clean, merge, and aggregate data
• Load: Insert processed data into SQLite database

Python Script:

```
"""
Walmart Data Munging Script - Task 4
====================================
Author: S. Rahul
Purpose: Populate SQLite database with shipping data from CSV files

This script performs ETL operations:
- Extract: Read data from three CSV files
- Transform: Clean, merge, and aggregate data
- Load: Insert processed data into SQLite database
"""

import pandas as pd
import sqlite3

def main():
    """
    Main function to execute the data munging pipeline
    """
    print("■ Starting Walmart Data Munging Process...")

    # --- Step 1: Extract - Read CSV files ---
    print("■ Reading CSV files...")
    sheet0 = pd.read_csv('data/shipping_data_0.csv')
    sheet1 = pd.read_csv('data/shipping_data_1.csv')
    sheet2 = pd.read_csv('data/shipping_data_2.csv')

    print(f"   - shipping_data_0.csv: {len(sheet0)} records")
    print(f"   - shipping_data_1.csv: {len(sheet1)} records")
    print(f"   - shipping_data_2.csv: {len(sheet2)} records")

    # --- Step 2: Connect to SQLite database ---
    print("■ Connecting to SQLite database...")
    conn = sqlite3.connect('shipment_database.db')
    cursor = conn.cursor()

    # --- Step 3: Transform & Load - Process shipping_data_0 (self-contained) ---
    print("■ Processing shipping_data_0 (self-contained dataset)...")

    # Map column names to match expected schema
    sheet0_mapped = sheet0.rename(columns={
        'origin_warehouse': 'origin',
        'destination_store': 'destination',
        'product': 'product_name',
        'product_quantity': 'quantity'
    })

    # Select only the columns we need for the shipments table
    sheet0_clean = sheet0_mapped[['origin', 'destination', 'product_name', 'quantity', 'on_t

    # Insert data into database
    sheet0_clean.to_sql('shipments', conn, if_exists='append', index=False)
    print(f"   ■ Inserted {len(sheet0_clean)} records from shipping_data_0")

    # --- Step 4: Transform - Combine shipping_data_1 + shipping_data_2 ---
```

```
    print("■ Merging shipping_data_1 and shipping_data_2...")
    merged = pd.merge(sheet1, sheet2, on='shipment_identifier')
    print(f"   ■ Merged data: {len(merged)} records")

    # --- Step 5: Transform - Group and calculate totals ---
    print("■ Aggregating merged data...")

    # Map column names for consistency
    merged_mapped = merged.rename(columns={
        'origin_warehouse': 'origin',
        'destination_store': 'destination',
        'product': 'product_name'
    })

    # Add a quantity column (assuming 1 for each product entry)
    merged_mapped['quantity'] = 1

    # Group by origin, destination, and product to aggregate quantities
    shipment_summary = (
        merged_mapped.groupby(['origin', 'destination', 'product_name'])
        .agg({'quantity': 'sum', 'on_time': 'first'})
        .reset_index()
    )

    # Add driver_identifier (placeholder since not available in merged data)
    shipment_summary['driver_identifier'] = 'unknown'

    # Reorder columns to match the shipments table schema
    shipment_summary = shipment_summary[['origin', 'destination', 'product_name', 'quantity'

    print(f"   ■ Aggregated data: {len(shipment_summary)} unique shipments")

    # --- Step 6: Load - Insert merged data into the same database ---
    print("■ Inserting aggregated data into database...")
    shipment_summary.to_sql('shipments', conn, if_exists='append', index=False)
    print(f"   ■ Inserted {len(shipment_summary)} aggregated records")

    # --- Step 7: Finalize - Close database connection ---
    print("■ Committing changes and closing database connection...")
    conn.commit()
    conn.close()

    # --- Success Message ---
    total_records = len(sheet0_clean) + len(shipment_summary)
    print(f"\n■ Database successfully populated with {total_records} total records!")
    print("   - Self-contained records:", len(sheet0_clean))
    print("   - Aggregated records:", len(shipment_summary))
    print("■ Data munging process completed successfully!")

if __name__ == "__main__":
    main()
```