Task 4: Data Munging

Here is your task

Part 1: Get the data

First, you need to get your hands on the relevant data. The shipping department has been kind enough to provide you with a repository containing all of their spreadsheets, as well as a copy of the sqlite database. First, fork and clone the repository at: https://github.com/theforage/forage-walmart-task-4

Part 2: Populate the database

Your task is to insert all of the data contained in the provided spreadsheets into the SQLite database. You will write a Python script which:

- Reads each row from the spreadsheets.
- Extracts the relevant data.
- Munges it into a format that fits the database schema.
- Inserts the data into the database.

Spreadsheet 0 is self contained and can simply be inserted into the database, but spreadsheets 1 and 2 are dependent on one another. Spreadsheet 1 contains a single product per row, you will need to combine each row based on its shipping identifier, determine the quantity of goods in the shipment, and add a new row to the database for each product in the shipment. The origin and destination for each shipment in spreadsheet 1 are contained in spreadsheet 2. You may assume that all the given data is valid - product names are always spelled the same way, quantities are positive, etc.

Solution :

```python
import pandas as pd
import sqlite3
# Step 1: Connect to SQLite database
DB_FILE = 'shipping.db'
conn = sqlite3.connect(DB_FILE)
cursor = conn.cursor()
# Step 2: Read all spreadsheets
spreadsheet0 = pd.read_csv('spreadsheet0.csv')
spreadsheet1 = pd.read_csv('spreadsheet1.csv')
spreadsheet2 = pd.read_csv('spreadsheet2.csv')
# Step 3: Process Spreadsheet 0 (self-contained)
for _, row in spreadsheet0.iterrows():
    cursor.execute('''
        INSERT OR IGNORE INTO Shipments
        (ShipmentID, ProductName, Quantity, Origin, Destination)
        VALUES (?, ?, ?, ?, ?)
    ''', (
        row['ShipmentID'],
        row['ProductName'],
        row['Quantity'],
        row['Origin'],
        row['Destination']
    ))
# Step 4: Process Spreadsheets 1 & 2 (dependent)
merged_shipments = pd.merge(
    spreadsheet1,
    spreadsheet2,
```

```python
        row['Quantity'],
        row['Origin'],
        row['Destination']
    ))
# Step 4: Process Spreadsheets 1 & 2 (dependent)
merged_shipments = pd.merge(
    spreadsheet1,
    spreadsheet2,
    on='ShippingID',
    how='left'
)
for _, row in merged_shipments.iterrows():
    cursor.execute('''
        INSERT OR IGNORE INTO Shipments
        (ShipmentID, ProductName, Quantity, Origin, Destination)
        VALUES (?, ?, ?, ?, ?)
    ''', (
        row['ShippingID'],
        row['ProductName'],
        row['Quantity'],
        row['Origin'],
        row['Destination']
    ))
# Step 5: Commit and close
conn.commit()
conn.close()
print('All spreadsheets have been successfully loaded into the database!')
```