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.

Code:

```
import csv
import sqlite3

def tableCreations(c):      #function to create the required tables
    c.execute("""
                CREATE TABLE IF NOT EXISTS shipping_data_0 (
                origin_warehouse TEXT,
                destination_store TEXT,
                product TEXT,
                on_time TEXT,
                product_quantity INTEGER,
                driver_identifier TEXT
            )
            """)

    c.execute("""
                CREATE TABLE IF NOT EXISTS shipping_data_1_2 (
                shipment_identifier TEXT,
                product TEXT,
                on_time TEXT,
                origin_warehouse TEXT,
                destination_store TEXT,
                driver_identifier TEXT
            )
            """)

def shippingData0Table(c):                  #function to create Data
Table 0

    with open('Python\\shipping_data_0.csv', 'r') as csv_file:
        csv_reader = csv.reader(csv_file)

        next(csv_reader)    #to not print the name of the field

        for line in csv_reader:
```

```
            origin_warehouse, destination_store, product, on_time,
product_quantity, driver_identifier = line   #breaking down the line
into 6 variables
            c.execute("INSERT INTO shipping_data_0 (origin_warehouse,
destination_store, product, on_time, product_quantity,
driver_identifier) VALUES (?, ?, ?, ?, ?, ?)",
                              (origin_warehouse, destination_store,
product, on_time, product_quantity, driver_identifier))

def shippingData1and2Table(cursor):         #function to create Data
Table 0

    shipping_data_2_dict = {}  # Creating a dictionary for storing

    with open('Python\\shipping_data_2.csv', 'r') as file:
        csv_reader = csv.reader(file)
        next(csv_reader)

        for line in csv_reader:
            shipment_identifier, origin_warehouse, destination_store,
driver_identifier = line
            shipping_data_2_dict[shipment_identifier] =
(origin_warehouse, destination_store, driver_identifier)

    with open('Python\\shipping_data_1.csv', 'r') as file:
        csv_reader = csv.reader(file)
        next(csv_reader)

        for line in csv_reader:
            shipment_identifier, product, on_time = line

            if shipment_identifier in shipping_data_2_dict:
#Finding shipment identifier in the dictionary
                origin_warehouse, destination_store,
driver_identifier = shipping_data_2_dict[shipment_identifier]
```

```
                cursor.execute("INSERT INTO shipping_data_1_2
(shipment_identifier, product, on_time, origin_warehouse,
destination_store, driver_identifier) VALUES (?, ?, ?, ?, ?, ?)",
                           (shipment_identifier, product, on_time,
origin_warehouse, destination_store, driver_identifier))

#----Main
conn = sqlite3.connect('shipment.db')    #Establsihing a conncection

c = conn.cursor()  #Creating a cursor

tableCreations(c)  #Creates the tables

shippingData0Table(c)        #Inserts into table 0
shippingData1and2Table(c)    #Inserts into table 1_2

c.execute("Select * from shipping_data_0")  #To view Table Contents
print(c.fetchall())

c.execute("Select * from shipping_data_1_2") #To view Table Contents
print(c.fetchall())

conn.commit()
conn.close()
```