```python
import sqlite3
import csv

class DataPopulator:
    def __init__(self, database_file):
        # Connect to SQLite3 database
        self.connection = sqlite3.connect(database_file)  # Create a connection object
        self.cursor = self.connection.cursor()            # Creates a cursor to interact with the database

    def populate_database(self, folder):
        # Specify the path to each spreadsheet
        file_0_path = f"{folder}/shipping_data_0.csv"
        file_1_path = f"{folder}/shipping_data_1.csv"
        file_2_path = f"{folder}/shipping_data_2.csv"

        # Read the file and process the data
        with open(file_0_path, newline='') as file_0:
            reader_0 = csv.reader(file_0)   # Create a reader object to read shipping_data_0.csv
            self._populate_shipping_data_1(reader_0)  # Call a method to process the data in Spreadsheet 1

        with open(file_1_path, newline='') as file_1, open(file_2_path, newline='') as file_2:
            reader_1 = csv.reader(file_1)   # Create a reader object to read shipping_data_1.csv
            reader_2 = csv.reader(file_2)   # Create a reader object to read shipping_data_2.csv
            self._populate_shipping_data_2(reader_1, reader_2)  # Call a method to process the data in Spreadsheet 2

    def _populate_shipping_data_1(self, reader_0):
        # Process the data in Spreadsheet 1
        for row_idx, row in enumerate(reader_0):
            if row_idx > 0:
                product_name = row[2]   # Extract the product name
                product_quantity = row[4]   # Extract product quantity
                origin = row[0]          # Extract the origin of shipping
                destination = row[1]     # Extract the destination

                # Output the extracted data
                print(product_name, product_quantity, origin, destination)

                # Insert the product into the database
                self._insert_product(product_name)
                # Insert shipment data into the database
                self._insert_shipment(product_name, product_quantity, origin, destination)

    def _populate_shipping_data_2(self, reader_1, reader_2):
        # Process the data in Spreadsheet 2
        shipment_info = {}
        for row_idx, row in enumerate(reader_2):
            if row_idx > 0:
                shipment_identifier = row[0]   # Extract the shipment identifier
                origin = row[1]                # Extract the origin of shipping
                destination = row[2]           # Extract the destination

                # Store shipping information in dictionary
                shipment_info[shipment_identifier] = {
                    "origin": origin,
                    "destination": destination,
                    "products": {}
                }

        for row_idx, row in enumerate(reader_1):
            if row_idx > 0:
                shipment_identifier = row[0]   # Extract the shipment identifier
                product_name = row[1]            # Extract product name

                products = shipment_info[shipment_identifier]["products"]
                # Increase product quantity by adding into the list
                products[product_name] = products.get(product_name, 0) + 1

        # Populate the database based on shipping information
        for shipment_identifier, shipment in shipment_info.items():
            origin = shipment["origin"]
            destination = shipment["destination"]
            for product_name, product_quantity in shipment["products"].items():
                self._insert_product(product_name)
                self._insert_shipment(product_name, product_quantity, origin, destination)

    def _insert_product(self, product_name):
        # Insert product into database
        query = '''
            INSERT OR IGNORE INTO product(name)
            VALUES(?);
        '''
        self.cursor.execute(query, (product_name,))
        self.connection.commit()

    def _insert_shipment(self, product_name, product_quantity, origin, destination):
        # Insert shipment data into the database
        # Search for the product in the database to get the product ID
        query = '''
            SELECT id
            FROM product
            WHERE name = ?;
        '''
        self.cursor.execute(query, (product_name,))
        # Gets the ID of the searched product
        product_id = self.cursor.fetchone()[0]

        # Insert product and shipment data into the database.
        query = '''
            INSERT OR IGNORE INTO shipment(product_id, quantity, origin, destination)
            VALUES(?,?,?,?);
        '''
        self.cursor.execute(query, (product_id, product_quantity, origin, destination))
        self.connection.commit()

    def close_connection(self):
        # Close the database connection
        self.connection.close()

if __name__ == '__main__':
    # Set the path to the folder containing database files and spreadsheet data
    data_populator = DataPopulator("C:/Users/Heeju/Documents/GitHub/forage-walmart-task-4/shipment_database.db")
    # Populate the database
    data_populator.populate_database("C:/Users/Heeju/Documents/GitHub/forage-walmart-task-4/data")
    # Close the database connection after all processes are complete
    data_populator.close_connection()
```