```
import sqlite3
import csv

class DatabaseConnector:

    def __init__(self, database_file):
        self.connection = sqlite3.connect(database_file)
        self.cursor = self.connection.cursor()

    def populate(self, folder):
        with open(f"{folder}/shipping_data_0.csv") as file_0:
            with open(f"{folder}/shipping_data_1.csv") as file_1:
                with open(f"{folder}/shipping_data_2.csv") as file_2:
                    reader_0 = csv.reader(file_0)
                    reader_1 = csv.reader(file_1)
                    reader_2 = csv.reader(file_2)

                    self.populate_shipping_data_1(reader_0)
                    self.populate_shipping_data_2(reader_1, reader_2)

    def populate_shipping_data_1(self, reader_0):
        for row_idx, row in enumerate(reader_0):
            if row_idx> 0:
                product_name = row[2]
                product_quantity = row[4]
                origin = row[0]
                destination = row[1]

                print(product_name, product_quantity, origin, destination)

                self.insert_product(product_name)
```

```
            self.insert_shipment(product_name, product_quantity, origin, destination)

    def populate_shipping_data_2(self, reader_1, reader_2):
        shipment_info = {}
        for row_idx, row in enumerate(reader_2):
            if row_idx > 0:
                shipment_identifier = row[0]
                origin = row[1]
                destination = row[2]

                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]
                product_name = row[1]

                products = shipment_info[shipment_identifier]["products"]
                products[product_name] = products.get(product_name, 0) + 1

        for shipment_identifier, shipment in shipment_info.items():
            origin = shipment_info[shipment_identifier]["origin"]
            destination = shipment_info[shipment_identifier]["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):
        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):
        query = '''
            SELECT id
            FROM product
            WHERE product.name = ?;
        '''
        self.cursor.execute(query, (product_name,))
        product_id = self.cursor.fetchone()[0]

        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(self):
        self.connection.close()

if __name__ == '__main__':
    db_connector = DatabaseConnector("shipment_database.db")

    db_connector.populate("./data")
```

```
db_connector.close()
```