```
import csv
import sqlite3


connectionSQLite = sqlite3.connect('Task4_GitRepo/forage-walmart-task-4/shipment_database.db')
cursorSQLite = connectionSQLite.cursor()
cursorSQLite.execute("DELETE FROM shipment")
connectionSQLite.commit()
cursorSQLite.execute("DELETE FROM product")
connectionSQLite.commit()


class ShippingDataClass1:
    def __init__(self, data:  list):
        self.origin_warehouse = data[0]
        self.destination_store = data[1]
        self.product = data[2]
        self.on_time = data[3]
        self.product_quantity = data[4]
        self.driver_identifier = data[5]
        self.product_id = None


# CSV FILE 0
with open('Task4_GitRepo/forage-walmart-task-4/data/shipping_data_0.csv', newline='') as shippingDataFile1:
    reader = csv.reader(shippingDataFile1, delimiter=',', quotechar='|')
    skipFirstRow = True
    productId = 1000
    shipmentId = 1000
    for row in reader:
        if skipFirstRow is True:
```

```
            skipFirstRow = False
        else:
            data = ShippingDataClass1(row)
            # Check if the product already has an ID
            res = cursorSQLite.execute(f"SELECT id FROM product WHERE name='{data.product}'")
            resProductId = res.fetchone()
            # If it doesn't, create one and add it to the product table
            if resProductId is None:
                string = f"INSERT INTO product VALUES ({productId}, '{data.product}')"
                cursorSQLite.execute(string)
                connectionSQLite.commit()
                data.product_id = productId
                productId += 1
            # else add existing productId to the data storage
            else:
                data.product_id = resProductId[0]
            # Retrieve the productID
            string = f"INSERT INTO shipment VALUES ({shipmentId}, {data.product_id},
{data.product_quantity}, '{data.origin_warehouse}', '{data.destination_store}')"
            cursorSQLite.execute(string)
            connectionSQLite.commit()
            shipmentId += 1


class ShippingDataClass2:
    def __init__(self, data: list):
        self.shipment_identifier = data[0]
        self.origin_warehouse = data[1]
        self.destination_store = data[2]
        self.driver_identifier = data[3]
        self.product = None
```

```
        self.on_time = None
        self.product_id = None
        self.product_quantity = 0

    def addMoreData(self, data: list):
        self.product = data[1]
        self.on_time = data[2]
        self.product_quantity += 1

# CSV FILE 1 AND 2
# Store data from csv 2 in a dict
shippingDataDict = {}
with open('Task4_GitRepo/forage-walmart-task-4/data/shipping_data_2.csv', newline='') as shippingDataFile2:
    reader = csv.reader(shippingDataFile2, delimiter=',', quotechar='|')
    skipFirstRow = True
    for row in reader:
        if skipFirstRow is True:
            skipFirstRow = False
        else:
            data = ShippingDataClass2(row)
            shippingDataDict.update({data.shipment_identifier: data})

with open('Task4_GitRepo/forage-walmart-task-4/data/shipping_data_1.csv', newline='') as shippingDataFile3:
    reader = csv.reader(shippingDataFile3, delimiter=',', quotechar='|')
    skipFirstRow = True
    for row in reader:
        if skipFirstRow is True:
            skipFirstRow = False
```

```
        else:
            # Collect the data and merge the duplicates
            shipment_id = row[0]
            data: ShippingDataClass2 = shippingDataDict[shipment_id]
            data.addMoreData(row)


for key in shippingDataDict:
    data = shippingDataDict[key]
    res = cursorSQLite.execute(f"SELECT id FROM product WHERE name='{data.product}'")
    resProductId = res.fetchone()
    # If it doesn't, create one and add it to the product table
    if resProductId is None:
        string = f"INSERT INTO product VALUES ({productId}, '{data.product}')"
        cursorSQLite.execute(string)
        connectionSQLite.commit()
        data.product_id = productId
        productId += 1
    # else add existing productId to the data storage
    else:
        data.product_id = resProductId[0]
    # Retrieve the productID
    string = f"INSERT INTO shipment VALUES ({shipmentId}, {data.product_id},
{data.product_quantity}, '{data.origin_warehouse}', '{data.destination_store}')"
    cursorSQLite.execute(string)
    connectionSQLite.commit()
    shipmentId += 1
```