```
import sqlite3
import csv

# Connect to SQLite database
conn = sqlite3.connect("shipment_database.db")
cursor = conn.cursor()

# Helper: Get or insert product and return its ID
def get_or_create_product(product_name):
    cursor.execute("SELECT id FROM product WHERE name = ?", (product_name,))
    result = cursor.fetchone()
    if result:
        return result[0]
    cursor.execute("INSERT INTO product (name) VALUES (?)", (product_name,))
    conn.commit()
    return cursor.lastrowid

# PART 1: Insert data from shipping_data_0.csv
with open("shipping_data_0.csv", newline='') as file:
    reader = csv.DictReader(file)
    for row in reader:
        product_id = get_or_create_product(row["product"])
        quantity = int(row["product_quantity"])
        origin = row["origin_warehouse"]
        destination = row["destination_store"]

        cursor.execute(
            "INSERT INTO shipment (product_id, quantity, origin, destination) VALUES (?, ?, ?, ?)",
            (product_id, quantity, origin, destination)
        )

# PART 2: Handle shipping_data_1.csv and shipping_data_2.csv (which must be joined on
shipment_identifier)
# Step 1: Load shipment locations from shipping_data_2.csv
shipment_locations = {}
with open("shipping_data_2.csv", newline='') as file:
    reader = csv.DictReader(file)
    for row in reader:
        shipment_id = row["shipment_identifier"]
        shipment_locations[shipment_id] = {
            "origin": row["origin_warehouse"],
            "destination": row["destination_store"]
        }

# Step 2: Process shipping_data_1.csv and insert joined results
with open("shipping_data_1.csv", newline='') as file:
    reader = csv.DictReader(file)
    shipment_product_counts = {}

    # Aggregate quantity by shipment and product
    for row in reader:
        shipment_id = row["shipment_identifier"]
        product = row["product"]
        key = (shipment_id, product)
        shipment_product_counts[key] = shipment_product_counts.get(key, 0) + 1

    for (shipment_id, product), quantity in shipment_product_counts.items():
        if shipment_id not in shipment_locations:
            continue  # Skip if shipment info is missing

        origin = shipment_locations[shipment_id]["origin"]
        destination = shipment_locations[shipment_id]["destination"]
        product_id = get_or_create_product(product)

        cursor.execute(
            "INSERT INTO shipment (product_id, quantity, origin, destination) VALUES (?, ?, ?, ?)",
            (product_id, quantity, origin, destination)
        )

# Finalize and close
conn.commit()
conn.close()
print("Data inserted successfully.")
```