E-commerce Data Engineering Exercise Solution

1. Synthetic E-commerce Data Generator (Python)

```
import pandas as pd
import numpy as np
from faker import Faker
import random
from datetime import datetime, timedelta
import os

fake = Faker()
os.makedirs("data", exist_ok=True)

# Users
users = []
for i in range(200):
    users.append({
        "user_id": i+1,
        "name": fake.name(),
        "email": fake.email(),
        "signup_date": fake.date_between(start_date='-2y', end_date='today')
    })
pd.DataFrame(users).to_csv("data/users.csv", index=False)

# Products
products = []
for i in range(150):
    products.append({
        "product_id": i+1,
        "name": fake.word().title(),
        "price": round(random.uniform(5, 500), 2),
        "category": fake.word()
    })
pd.DataFrame(products).to_csv("data/products.csv", index=False)

# Orders
orders = []
for i in range(300):
    orders.append({
        "order_id": i+1,
        "user_id": random.randint(1, 200),
        "order_date": fake.date_between(start_date='-1y', end_date='today')
    })
pd.DataFrame(orders).to_csv("data/orders.csv", index=False)

# Order Items
order_items = []
for i in range(800):
    order_items.append({
        "order_item_id": i+1,
        "order_id": random.randint(1, 300),
        "product_id": random.randint(1, 150),
        "quantity": random.randint(1, 5)
    })
pd.DataFrame(order_items).to_csv("data/order_items.csv", index=False)

# Reviews
reviews = []
for i in range(200):
    reviews.append({
        "review_id": i+1,
        "user_id": random.randint(1, 200),
        "product_id": random.randint(1, 150),
        "rating": random.randint(1, 5),
        "review_date": fake.date_between(start_date='-1y', end_date='today')
    })
pd.DataFrame(reviews).to_csv("data/reviews.csv", index=False)

print("Synthetic data successfully generated.")
```

2. SQLite Ingestion Script

```
import sqlite3
import pandas as pd

conn = sqlite3.connect("ecom.db")
cur = conn.cursor()

cur.executescript("""
DROP TABLE IF EXISTS users;
DROP TABLE IF EXISTS products;
DROP TABLE IF EXISTS orders;
DROP TABLE IF EXISTS order_items;
DROP TABLE IF EXISTS reviews;

CREATE TABLE users (
    user_id INTEGER PRIMARY KEY,
    name TEXT,
    email TEXT,
    signup_date TEXT
);

CREATE TABLE products (
    product_id INTEGER PRIMARY KEY,
    name TEXT,
    price REAL,
    category TEXT
);

CREATE TABLE orders (
    order_id INTEGER PRIMARY KEY,
    user_id INTEGER,
    order_date TEXT,
    FOREIGN KEY(user_id) REFERENCES users(user_id)
);

CREATE TABLE order_items (
    order_item_id INTEGER PRIMARY KEY,
    order_id INTEGER,
    product_id INTEGER,
    quantity INTEGER,
    FOREIGN KEY(order_id) REFERENCES orders(order_id),
    FOREIGN KEY(product_id) REFERENCES products(product_id)
);

CREATE TABLE reviews (
    review_id INTEGER PRIMARY KEY,
    user_id INTEGER,
    product_id INTEGER,
    rating INTEGER,
    review_date TEXT,
    FOREIGN KEY(user_id) REFERENCES users(user_id),
    FOREIGN KEY(product_id) REFERENCES products(product_id)
);
""")
conn.commit()

for table in ["users","products","orders","order_items","reviews"]:
    df = pd.read_csv(f"data/{table}.csv")
    df.to_sql(table, conn, if_exists="append", index=False)
    print(f"Loaded {table}")

conn.close()
```

3. SQL Queries

```
-- Join query
SELECT
    u.user_id,
    u.name AS user_name,
    o.order_id,
    o.order_date,
    p.name AS product_name,
    oi.quantity,
    p.price,
    (oi.quantity * p.price) AS total_line_value
FROM orders o
JOIN users u ON o.user_id = u.user_id
JOIN order_items oi ON oi.order_id = o.order_id
JOIN products p ON p.product_id = oi.product_id
ORDER BY o.order_date DESC;

-- Revenue per user
SELECT
    u.user_id,
    u.name,
    SUM(oi.quantity * p.price) AS total_revenue
FROM users u
JOIN orders o ON u.user_id = o.user_id
JOIN order_items oi ON oi.order_id = o.order_id
JOIN products p ON p.product_id = oi.product_id
GROUP BY u.user_id
ORDER BY total_revenue DESC;
```

4. GitHub Push Instructions

```
git init
echo "__pycache__/
ecom.db" > .gitignore
git add .
git commit -m "Initial commit - ecom project"
git branch -M main
git remote add origin <your-repo-url>
git push -u origin main
```