Instructions:

- Complete all questions.
- Use SQL and Python where applicable.
- Clearly explain your reasoning and assumptions.
- Submit your solutions as a Jupyter Notebook or script files.

Question 1: SQL Query Creation

You have the following tables in a **PostgreSQL** database:

Tables:

- `orders(order_id, customer_id, order_date, status)`
- `order_items(order_item_id, order_id, product_id, quantity, price, discount, tax)`
- `customers(customer_id, name, email, country, created_at)`
- `products(product_id, name, category, base_price)`

Write an **SQL query** to get the top 10 customers by **net total spending** in the last **6 months**, considering the following:

- Only include **completed** orders.
- Calculate the total amount spent per customer, including tax and discount.
- Display the customer's country and the number of distinct products they have purchased.

The output should include:

- `customer_id`
- `customer_name`
- `country`
- `total_spent`
- `unique_products_purchased`

Sample Data:

orders:

order_id	customer_id	order_date	status
1	101	2023-10-01	Completed
2	102	2024-01-15	Completed
3	103	2024-02-10	Pending

4	101	2024-02-20	Completed

order_items:

order_item_id	order_id	product_id	quantity	price	discount	tax
1	1	201	2	50	5	2
2	2	202	1	100	10	5
3	3	203	5	20	2	1
4	4	204	3	30	3	2

customers:

customer_id	name	email	country	created_at
101	John	john@email.com	USA	2022-03-10
102	Jane	jane@email.com	UK	2023-06-21
103	Mike	mike@email.com	Canada	2024-01-05

Expected Output:

customer_id	customer_name	country	total_spent	unique_products_purchased
101	John	USA	154	2
102	Jane	UK	95	1

Question 2: Data Modeling

You are designing a **data warehouse** for an e-commerce platform. The company wants to efficiently track and analyze the following:

- Orders, including order statuses, timestamps, and payment details.
- Customers, including their demographics and purchasing behavior.
- Products, including categories, pricing history, and supplier information.
- Shipments, including tracking status, delivery times, and logistics providers.

- Promotions and discounts applied to orders.

Tasks:

1. **Design a Star Schema:**
    - Identify and describe the **fact table(s)** and **dimension tables**.
    - Specify primary and foreign keys.
    - Include surrogate keys where applicable.
2. **Handle Slowly Changing Dimensions (SCD):**
    - Explain how you would track historical changes for product pricing and customer details.
    - Justify whether you would use **Type 1, Type 2, or Type 3 SCD** for each case.
3. **Optimize for Performance:**
    - Propose strategies for indexing and partitioning.
    - Discuss how to handle large-scale data growth efficiently.
4. **Provide an ERD (Entity Relationship Diagram) or Schema Diagram:**
    - Include relationships between tables.
    - Highlight key attributes and constraints.

Question 3: Data Pipeline Design (20 points)

You need to build a **batch ETL pipeline** that processes JSON files containing customer transactions. The pipeline should:

- **Extract:** Read JSON files from a local directory.
- **Transform:** Cleanse and normalize the data using Apache Spark, including:
    - Handling missing values and duplicate records.
    - Converting data types appropriately.
    - Aggregating transaction amounts per customer.
- **Load:** Write the transformed data into a **PostgreSQL database**.

Sample JSON Data:

```
[
  {
    "transaction_id": "T001",
    "customer_id": "101",
    "timestamp": "2024-03-01T12:34:56Z",
    "amount": 100.5,
    "currency": "USD",
```

```
  "status": "completed"
 },
 {
  "transaction_id": "T002",
  "customer_id": "102",
  "timestamp": "2024-03-02T15:20:30Z",
  "amount": 200.0,
  "currency": "USD",
  "status": "failed"
 },
 {
  "transaction_id": "T003",
  "customer_id": "101",
  "timestamp": "2024-03-03T18:45:00Z",
  "amount": 50.75,
  "currency": "USD",
  "status": "completed"
 }
]
```


Expected Output:

customer_id	total_transactions	total_amount
101	2	151.25
102	1	0.00

(Note: The failed transaction for customer 102 is excluded from the final aggregation.)

Tasks:

1. **Describe the entire ETL process, including tools used and justifications for design choices.**
2. **Provide a PySpark script** to perform the transformation steps, including:
    - Schema definition
    - Data cleaning operations
    - Aggregation logic
3. **Discuss error handling and monitoring strategies**, such as:
    - Handling corrupt/malformed JSON files.
    - Implementing logging and alerting mechanisms.
    - Ensuring idempotency and failure recovery.
