Problem Statement:

A retail chain receives large volumes of sales data from various stores across different locations in .csv format daily. Your task is to automate the ETL process for this data using **Python**, store it in a **MySQL** database, and perform **data cleaning, transformation**, and **analysis** using **Pandas and NumPy**.

Dataset Description (you can generate mock data or use open datasets)

Each store sends a .csv file with the following fields:

- Store_ID (string)
- Date (YYYY-MM-DD)
- Product_ID (string)
- Product_Name (string)
- Quantity_Sold (integer)
- Unit_Price (float)
- Discount_Percent (float)
- Payment_Mode (Cash/Card/UPI/Wallet)

Tasks:

Extraction

- Write a Python script to read multiple CSV files (assume stored in a data/ folder).
- Combine all data into a single DataFrame.

Transformation

Perform the following transformations:

- Handle missing values (e.g., fill with default values or drop rows).

- Create a new column Total_Sale_Value = Quantity_Sold * Unit_Price * (1 - Discount_Percent/100)
- Convert all column names to lowercase and ensure consistent formatting.
- Convert Date column to proper datetime format.
- Remove duplicates based on Store_ID, Date, and Product_ID.
- Categorize sales into High, Medium, Low based on Total_Sale_Value using NumPy.

Load to MySQL

- Design a MySQL table retail_sales matching the DataFrame.
- Use mysql.connector or SQLAlchemy to insert data into the MySQL table.
- Ensure idempotency: running the script multiple times should not insert duplicates.

Analysis & Reporting (Bonus Task)

- Use Pandas to calculate:
    - Total sales per store.
    - Top 5 products with the highest total sales.
    - Daily total sales trend for each store.
- Export the analysis to .csv or .xlsx reports (e.g., store_sales_summary.csv).

NOTE:

- Use .bat or .sh script to schedule the pipeline daily (optional advanced task).
- Add logging to track data flow and errors.