Data Integration and Automation Challenge: Merging Sales with Economic Indicators

Objective

The primary objective of this challenge is to assess your ability to manipulate sales data, merge it with external economic variables, and implement basic data integrity checks. This will require skills in data gathering, processing, merging datasets, and ensuring data quality.

Requirements

Data Creation:

- Generate a synthetic dataset for aggregated weekly sales data for a retail store, covering one year. You can assume any category, like soft drinks, furniture, shampoos. Incorporate seasonality , discounting effect on sales.
- Fields should include:
  - Week Start Date (Sunday to Saturday),
  - Product ID (create for 50 products)
  - Product Name
  - Units Sold
  - Price ($)
  - Discount Percentage (if applicable for that week)
  - Revenue ($)
  - Region == USA (can be fixed value)

External Data Integration:

- Use the FRED API to integrate weekly average gas prices, the Consumer Price Index (CPI)
- Details on the FRED API are available here.
  - Average Gas Price (US): https://fred.stlouisfed.org/series/GASREGW
  - Consumer Price Index (CPI): https://fred.stlouisfed.org/series/CPIAUCSL

Data Merging:

- Merge the sales data with the economic indicators by week dates, enriching each sales record with relevant economic data.

Data Refresh and Automation:

- Write a script that automates the fetching and updating of the sales and economic data each month.
- The script should handle:
  - Downloading new sales data (simulated for the purpose of this challenge).
  - Fetching the latest economic indicators from FRED.
  - Merging these datasets.

- Saving or updating the combined dataset in a designated location (e.g., a local file or a database).

Data Integrity Checks and Alerts:

- Implement data checks that:
  - Ensure no missing values in critical fields.
  - Detect anomalies in sales trends such as unexpected spikes or drops or missing products
  - Confirm correct alignment of merged data by date.
- Develop a system to alert stakeholders (e.g., via email or a log file) if issues are detected during the data checks.

Documentation:

- Provide documentation with:
  - Execution instructions.
  - An overview of your approach.
  - Assumptions made during the challenge.
- Comment your code adequately to explain critical sections and logic.

Submission:

- Submit your code, preferably as a Python or SQL scripts as files or Jupyter notebooks.
- Include sample csv files for testing.
- Ensure your code is executable and can reproduce the results described.

Evaluation Criteria

- Automation Efficiency: Robustness and reliability of the automation script.
- Data Handling: Accuracy of data merging and updating processes.
- Alert System: Effectiveness of the alert system in catching and reporting issues.
- Code Quality: Clarity, readability, and organisation of code.
- Documentation: Detail and clarity in documentation.