commonshare

Oussama LAHLIMI

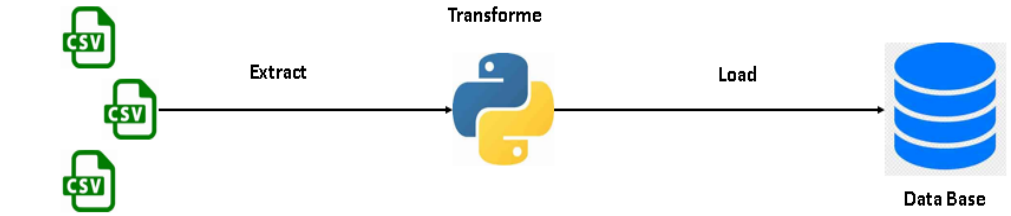
Documentation of Data Acquisition and ETL Processes

Date: 2024/08/31 **Makes**: Tanger

- **Definition of documentation:**

Focus on their ability to handle the entire ETL (Extract, Transform, and Load) process. This includes **data acquisition** from various sources (APIs, databases, files, web scraping), **preprocessing and cleaning** to manage missing values and inconsistencies, and data transformation to prepare data for analysis. Additionally, evaluate their skills in **designing an automated ETL pipeline, loading data into databases** (**Postgresql**) or data warehouses, and handling errors and optimizations to ensure efficient, reliable workflows.

- **ETL processes for this challenge:**



✓ **Objective of this challenge is**: Extract raw data from CSV files, transform it into a clean and usable format with python I do that, I verified I's transformation with my analysis the quality of this data like non-null value or duplication of the data I remove the fake data and the data (rows), and load it into a structured relational database **(postgresql**). The goal is to ensure that the data is properly formatted, cleansed, and stored in a way that makes it easy to query and analyze.

✓ The **library** with python I used the **pandas** for the extraction and the transformation and the load into DB I use **sqlalchemy** and **Pipeline**, **MinMaxScaler**, **FunctionTransformer** for Normalization of the data.

✓ We can schedule automation this process ETL with python to be extract transform and load into database real time (if the data it's the same resources).

- **Data Acquisition:**

  Explanation of CSV Loading Error Handling:

  1) File Not Found: Catches FileNotFoundError if the file path is incorrect or the file does not exist.
  2) Empty Data: Catches pd.errors.EmptyDataError if the file is empty.
  3) Parsing Errors: Catches pd.errors.ParserError if there are issues with the format of the CSV file.
  4) General Exceptions: Catches any other exceptions to avoid unexpected script termination.

  By implementing robust error handling in both API data acquisition and CSV file loading, you can ensure a reliable data acquisition process, which is essential for building robust ETL pipelines. This approach helps to manage potential errors gracefully and maintain data integrity.

- **Data Cleaning:**

  All the data frame non-null, Ensure data integrity and handle any potential data quality issues.
  The data is now cleaned and transformed into a usable format:
  - Missing Values are handled using appropriate strategies.
  - Duplicates are removed to avoid redundancy.
  - Inconsistencies are fixed by standardizing formats and units.
  - Data Types are converted to appropriate types for analysis or modeling.

- **Load data :**
  load_to_db(df, table name, engine) ; 'it's take three 3 argument the data frame in this case it's the df from all resources files and the table name it's the name of the table he create in our data base and the last argument it's just the connection between the data base.
  This is the function responsible for the loading the data into data base.

- **Schema of the table**