# 1. Introduction

The **DEX Ingestion Framework** is a Python-based automation tool that converts data from an Excel file into YAML configuration files. It automates the process of generating YAML files for data ingestion tasks, such as loading data from a source database to Redshift. The framework also interacts with JSON files to dynamically update column names, add additional columns (ETL columns, OGG columns), and ensure data consistency based on specific conditions. By reducing manual effort and minimizing errors, the framework enhances efficiency in large-scale data ingestion workflows.

# 2. Project Overview

The **DEX Ingestion Framework** is designed to automate and streamline the ingestion of structured data from source systems into a target database (Redshift). This framework reads configuration details from an Excel template, processes metadata from JSON files, and generates YAML configurations and SQL queries for seamless data transfer.

# 3. Key Features

**Excel-Based Configuration**: Extracts metadata from an Excel template.  
**JSON Schema Processing:** Reads JSON schema definitions for table structures.  
**Automated Column Renaming:** Identifies SQL reserved keywords and renames columns to avoid conflicts.  
**YAML Generation:** Produces YAML configuration files (DBtoRedshift.yml & OGGToRedshift.yml) for data ingestion.  
**SQL Query Generation:** Creates SELECT queries with support for PII / SPII masking.  
**Secret Management Integration:** Fetches database credentials from a secret\_name.json file.  
**Additional Columns:** ETL-related columns are added to the JSON data if the database type is not 'sybase'.  
**OGG Handling:** Special handling for OGG-related columns if the task is OGGToRedshift.  
**Historical Tables:** Support for handling archived/historical tables.  
**Partitioning:** Automatic partitioning based on table size and reliable date columns.

# 4. Benefits

This framework offers several advantages:

* **Automation:** Eliminates manual YAML & SQL file creation, reducing human errors.
* **Scalability:** Easily adaptable to new tables and source systems.
* **Data Security:** Supports masking of PII/SPII data through SQL transformations.
* **Standardization:** Ensures consistent data transformation across ingestion pipelines.
* **Flexibility:** Allows customization based on source system types and ingestion needs.

# 5. How to Use the Framework

**Step 1: Prepare the Input Files**

* Place the ingestion metadata Excel file (DEX-Table\_Ingestion\_Template-V1.xlsx) in the working directory.
* Ensure that JSON schema files are available in the srcl folder.
* Verify that the secret\_name.json file contains correct database credentials.

**Step 2: Execute the Framework**

1. Run the main script:
2. The script will:

* Read the Excel file and extract table details.
* Process JSON schema files to extract column metadata.
* Generate YAML configurations (DBtoRedshift.yml, OGGToRedshift.yml).
* Generate SQL queries and store them in the srcl\_vw folder.

**Step 3: Review the Outputs**

* YAML Files: Used for configuring ETL jobs.
* SQL Files: Used for running transformations and secure data selection.
* Logs: Console logs help track processing steps and errors

# 6. Folder Structure

Excel to YAML Converter/

├── dex\_ingestion/ # Main Python package

│ ├── \_\_init\_\_.py # Makes the folder a Python package

│ ├── constants.py # Contains constants like KEYWORDS\_TO\_RENAME

│ ├── excel\_processor.py # Processes the Excel file

│ ├── main.py # Main script to run the project

│ ├── sql\_generator.py # Generates SQL files

│ ├── utils.py # Utility functions (e.g., rename\_column\_if\_keyword)

│ └── yaml\_generator.py # Generates YAML files

├── requirements.txt # Lists all dependencies

├── secret\_name.json # Contains secret names for source databases

├── srcl/ # Folder for input JSON files

│ └── ebt\_rm\_data.json # Example JSON file

├── srcl\_vw/ # Folder for generated SQL files (created automatically)

├── temp/ # Folder for updated JSON files (created automatically)

└── DEX-Table\_Ingestion\_Template-V1.xlsx # Input Excel file

### ****Explanation of Each File/Folder****

1. **Excel to YAML Converter/**:
   * This is the root folder of the project. It contains all the files and subfolders.
2. **dex\_ingestion/**:
   * This is the main Python package for the project. It contains all the Python scripts.
3. **\_\_init\_\_.py**:
   * This file makes the dex\_ingestion folder a Python package, allowing you to use relative imports (e.g., from .excel\_processor import process\_excel).
4. **constants.py**:
   * Contains constants like KEYWORDS\_TO\_RENAME, which are used to rename columns that match SQL keywords.
5. **excel\_processor.py**:
   * Reads and processes the Excel file (DEX-Table\_Ingestion\_Template-V1.xlsx) to extract metadata.
6. **main.py**:
   * The main script that orchestrates the entire process. It calls functions from other modules to process the Excel file, generate YAML files, and create SQL files.
7. **sql\_generator.py**:
   * Generates .sql files based on the JSON files and saves them in the srcl\_vw folder.
8. **utils.py**:
   * Contains utility functions like rename\_column\_if\_keyword, which renames columns if they match SQL keywords.
9. **yaml\_generator.py**:
   * Generates YAML files (DBtoRedshift.yml and OGGToRedshift.yml) based on the processed data.
10. **requirements.txt**:
    * Lists all the Python packages required to run the project. Install them using pip install -r requirements.txt.
11. **secret\_name.json**:
    * Contains a list of secret names for source databases. Used to populate the source\_secret\_name field in the YAML files.
12. **srcl/**:
    * Contains the input JSON files (e.g., ebt\_rm\_data.json). These files define the structure of the tables being ingested.
13. **srcl\_vw/**:
    * This folder is created automatically and contains the generated .sql files.
14. **temp/**:
    * This folder is created automatically and contains the updated JSON files with the schemaName changed to "temp".
15. **DEX-Table\_Ingestion\_Template-V1.xlsx**:
    * The input Excel file that contains metadata about the data ingestion process.