# Testing ETL Framework with Sample Data in GitHub Repository

## Objective

To design a solution for testing the ETL framework (config.py logic) using sample data files stored in GitHub. The goal is to dynamically read sample data files from GitHub folders and run test cases automatically when code is pushed or a pull request is created, ensuring correctness before deployment to Google Cloud Platform (GCP).

## Key Requirements

- Dynamic Data Loading: Sample data files for the tables must be stored in GitHub folders.  
- Automated Testing: Validate each config.py file using the sample data.  
- GitHub Integration: Automatically trigger test cases on code pushes or pull requests.  
- Scalability: Support testing for multiple config.py files.  
- Environment Independence: No reliance on BigQuery or external databases for testing; use PySpark in local mode.

## Proposed Solutions

Here are the proposed solutions for the given requirements:

### Solution 1: Local Spark with Dynamic Data Loading

- Use PySpark local mode to process sample data files.  
- Dynamically read sample data files from GitHub folders and register them as temporary views.  
- Validate config.py logic against these views.  
- Advantages: Simple and works directly with GitHub Actions without requiring additional infrastructure.

### Solution 2: Dockerized Testing

- Package the test framework into a Docker container.  
- The container reads sample data files from the GitHub repository, processes them with PySpark, and runs test cases.  
- Advantages: Ensures consistent environments for testing; easily scalable.  
- Limitations: Additional setup required for Docker.

## Recommended Solution

Based on the requirements, the recommended approach is Solution 1: Local Spark with Dynamic Data Loading, combined with GitHub Actions for automation.

## Implementation Plan

### Dynamic Data Loading

create the test\_cases.py to dynamically load data files.

Each config.py can have its own run\_test.py, allowing testing.

### GitHub Actions Workflow

Create .github/workflows/test.yml

## Advantages of Recommended Approach

- Dynamic Testing: Automatically loads and validates all config.py files with associated sample data.  
- Automated CI/CD: Fully automated testing pipeline with GitHub Actions.  
- Scalable: Easily add more sample data files and config.py files.  
- Cost-Effective: Runs locally without requiring external infrastructure.