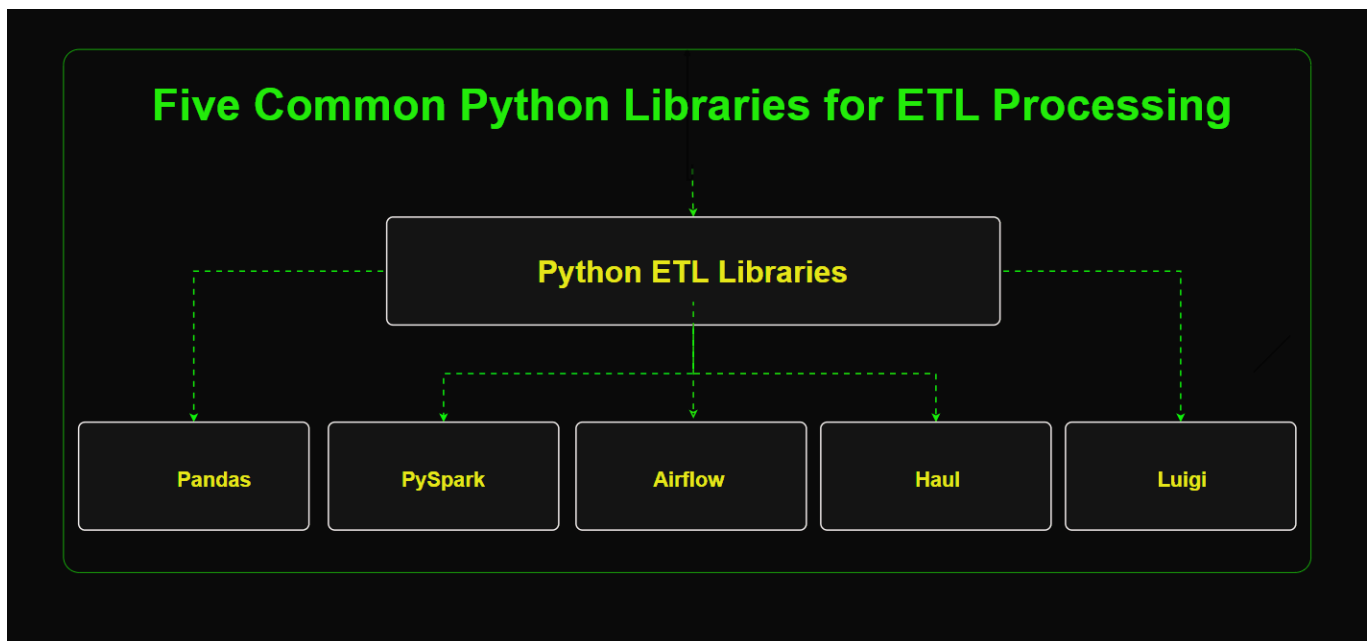


# Five Common Python Libraries for ETL Processing



Gen. David L. · [Follow](#)

5 min read · Dec 10, 2024



ETL is a critical concept in data warehousing technology, representing three main processing steps: Extract, Transform, and Load. ETL is commonly used for data integration, merging data from various sources into a unified view for analysis and reporting.

## ETL Detailed Process

### 1. Extract (E):

This step involves extracting data from various sources, which could include relational databases, file systems, cloud storage, or APIs. The extraction process may involve copying or moving data and recognizing and parsing its format.

### 2. Transform (T):

Once the data is extracted, it usually needs to be transformed into a format suitable for analysis. This process may include:

- Data cleansing (e.g., removing duplicates, correcting errors, and resolving inconsistencies).
- Data aggregation (e.g., calculating sums, averages, maximum/minimum values).
- Data normalization (e.g., converting data into a unified format or unit).
- Data enrichment (e.g., adding additional contextual information).

The transformation step is the most critical part of ETL as it directly affects the quality and usability of the data in the final data warehouse.

### 3. Load (L):

The loading step involves writing the transformed data into the target system, which could be a data warehouse, data lake, data mart, or another

form of data storage. During this step, the data may be indexed, partitioned, or archived to optimize query performance and storage efficiency.

ETL processes can be either batch-based or real-time. Batch ETL typically runs at scheduled intervals, such as daily, weekly, or monthly, whereas real-time ETL requires the system to continuously process and update data.

## Common Python ETL Libraries

### 1. Pandas

**Advantages:** Pandas provides data structures and analytical tools that are highly suitable for data cleaning and transformation. It simplifies data manipulation tasks in the ETL process, such as adding new columns and filtering data.

**Disadvantages:** Pandas may encounter performance bottlenecks when handling large datasets, as it primarily operates on in-memory data.

**Use Cases:** Suitable for small to medium-sized datasets and for rapid prototyping.

### Code Example:

```
import pandas as pd

# Read data from a CSV file
df = pd.read_csv('my_example.csv')

# Data transformation, such as adding a new column
df['new_A_column'] = df['existing_A_column'].apply(lambda x: x * 5.4)
```

```
# Save the transformed data to a new CSV file
df.to_csv('transformed_example.csv', index=False)
```

## 2. Apache Spark (PySpark)

**Advantages:** Apache Spark is a unified analytics engine designed for large-scale data processing. The PySpark API makes it simple to handle Spark jobs within Python workflows.

**Disadvantages:** Compared to Pandas, Spark has a steeper learning curve and requires more resources for setup and operation.

**Use Cases:** Ideal for processing large datasets, especially in distributed computing environments.

### Code Example:

```
from pyspark.sql import SparkSession

# Initialize a Spark session
spark = SparkSession.builder.appName('etl_example').getOrCreate()

# Read data from a CSV file
df = spark.read.csv('input_example_data.csv', header=True, inferSchema=True)

# Data transformation
df = df.withColumn('new_A_column', df['existing_A_column'] * 5.4)

# Write the data to a new CSV file
df.write.csv('output_example_data.csv')
```

### 3. Haul

**Advantages:** Haul is a lightweight and modular Python library designed specifically for ETL tasks. It focuses on simplifying data extraction, transformation, and loading by providing clean, reusable components that integrate seamlessly into modern workflows.

**Disadvantages:** Haul is relatively new and less feature-rich compared to more established ETL frameworks like Airflow or Luigi. It may not be suitable for highly complex ETL pipelines or large-scale distributed systems.

**Use Cases:** Ideal for small to medium-sized ETL workflows, rapid prototyping, and scenarios requiring flexibility and simplicity.

#### Code Example:

```
from haul import Extractor, Transformer, Loader

# Define an extractor to load data from a CSV file
extractor = Extractor.from_csv('input_example_data.csv')

# Define a transformer to add a new column
class MultiplyTransformer(Transformer):
    def transform(self, row):
        row['new_A_column'] = row['existing_A_column'] * 5.4
    return row

transformer = MultiplyTransformer()

# Define a loader to save the data to a new CSV file
loader = Loader.to_csv('output_example_data.csv')

# Execute the ETL process
etl_pipeline = extractor | transformer | loader
etl_pipeline.run()
```

## 4. Luigi

**Advantages:** Luigi is a Python library designed for building complex batch job pipelines. It handles dependency resolution, workflow management, visualization, and failure handling.

**Disadvantages:** Luigi does not automatically synchronize tasks to worker nodes and lacks built-in scheduling, alerting, or monitoring features.

**Use Cases:** Suitable for automating simple ETL workflows, such as log processing.

### Code Example:

```
from luigi import Task, ExternalTask, Parameter, LocalTarget
import pandas as pd

# Define a task to extract data from a source URL
class ExtractData(Task):
    source_url = Parameter() # Define a parameter for the source URL of the data

    def run(self):
        # Read data from the source URL into a pandas DataFrame
        df = pd.read_csv(self.source_url)
        # Write the DataFrame to a CSV file at the specified output location
        self.output().open('w').write(df.to_csv(index=False))

    def output(self):
        # Define the output target for this task (a local CSV file)
        return LocalTarget('output_data_example.csv')

# Define a task to transform the data
class TransformData(ExternalTask):
    source_url = Parameter() # Define a parameter for the source URL of the input data

    def run(self):
        # Read data from the source URL into a pandas DataFrame
        df = pd.read_csv(self.source_url)
```

```

# Perform a hypothetical transformation: adding a new column
df['new_A_column'] = df['existing_A_column'] * 5.4
# Write the transformed DataFrame to a CSV file at the specified output
self.output().open('w').write(df.to_csv(index=False))

def output(self):
    # Define the output target for this task (a transformed local CSV file)
    return LocalTarget('transformed_example_data.csv')

# Main entry point for Luigi tasks
if __name__ == '__main__':
    # Run the TransformData task, which depends on 'example_data.csv' as input
    luigi.build([TransformData('example_data.csv')])

```

## 5. Airflow

**Advantages:** Apache Airflow is a powerful workflow orchestration tool, allowing users to programmatically author, schedule, and monitor workflows. Its flexibility and modular architecture make it ideal for creating complex ETL pipelines that can handle dependencies and integrate with various data sources.

**Disadvantages:** Airflow has a steep learning curve, especially for users unfamiliar with its DAG (Directed Acyclic Graph) paradigm. Additionally, it requires careful setup and maintenance, as scaling can introduce performance and resource challenges.

**Use Cases:** Best suited for orchestrating ETL workflows in environments with multiple steps or dependencies. It is particularly effective for managing workflows in distributed or cloud-based systems.

**Code Example:**

```

from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime

# Define the ETL functions
def extract():
    print("Extracting data...")
    print("more code here to do extract job")

def transform():
    print("Transforming data...")
    print("more code here to do transform job")

def load():
    print("Loading data...")
    print("more code here to do load job")

# Initialize the DAG
default_args = {
    'owner': 'airflow',
    'start_date': datetime(2024, 1, 1),
    'retries': 1,
}
dag = DAG(
    'example_etl',
    default_args=default_args,
    schedule_interval='@daily',
)

# Define the tasks
extract_task = PythonOperator(task_id='extract', python_callable=extract, dag=dag)
transform_task = PythonOperator(task_id='transform', python_callable=transform, dag=dag)
load_task = PythonOperator(task_id='load', python_callable=load, dag=dag)

# Set task dependencies
extract_task >> transform_task >> load_task

```

## Summary

These libraries and tools each have their strengths, and the choice of which to use depends on the specific project requirements, data scale, and the



familiarity of the development team.

The ETL process forms the foundation of data management and analysis, ensuring data consistency, accuracy, and availability. This enables data analysis and business intelligence (BI) tools to deliver valuable insights and support decision-making.

With technological advancements, the ETL process continues to evolve, giving rise to variations like ELT (Extract, Load, Transform), where data is first loaded into the target system before transformation. This approach can improve efficiency when dealing with large-scale datasets.

Thanks for your reading.

Python Etl

Etl Tool

Etl Pipeline

Python Data Preprocessing



**Written by Gen. David L.**

453 Followers · 4 Following

Follow

AI practitioner & python coder to record what I learned in python project development

---

**Responses (3)**



What are your thoughts?

Respond



★ david libert ("dadoo")

29 days ago



Hello thank you. You forgot Mage.ai, very cool and strong solution 😊



4 [Reply](#)



★ Dinu Gherman

27 days ago



Missing prefect.io.



2 [Reply](#)



R. Ganesh

16 days ago

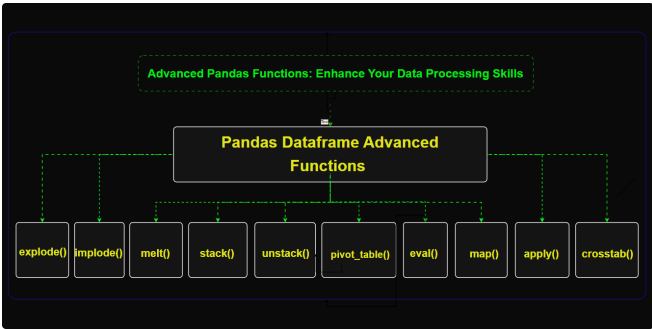


these frameworks not libraries



[Reply](#)

**More from Gen. David L.**



M Gen. David L.

## Advanced Pandas Features: Enhance Your Data Processing...

The essence of data analysis lies in uncovering the stories behind the data. In thi...

Dec 5, 2024 🖱 105



M Gen. David L.

## ETL-PIPES: An Efficient Python ETL Data Processing Library

ETL-pipes is a powerful and flexible Python library specifically designed for ETL (Extract...

Nov 30, 2024 🖱 45 💬 6

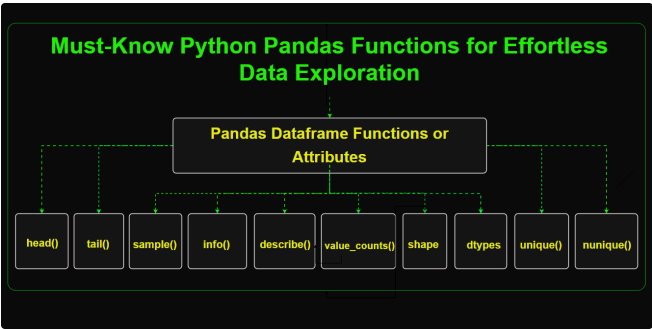


M Gen. David L.

## How to Map Column Values in a Pandas DataFrame?

Mapping column values refers to replacing specific values in a column with other values,...

Dec 24, 2024 🖱 81



M Gen. David L.

## Must-Know Python Pandas Functions for Effortless Data...

The key point of data analysis lies in uncovering the stories behind the data. To...

Nov 27, 2024 🖱 19




See all from Gen. David L.

# Recommended from Medium



**Top 25 Python Scripts To Automate Your Daily Tasks**

www.vastites.ca +16474916566

 Harold Finch

## Top 25 Python Scripts To Automate Your Daily Tasks

Python is an excellent tool for automating daily tasks, thanks to its simplicity and a wid...

Nov 19, 2024  337  5  




 Vijay Gadhave

## Must-Have Data Engineering Certifications

Note: If you're not a medium member, [CLICK HERE](#)


 Dec 3, 2024  300  5  

### Lists



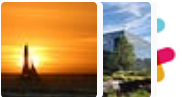
**Staff picks**

798 stories · 1568 saves




**Self-Improvement 101**

20 stories · 3214 saves



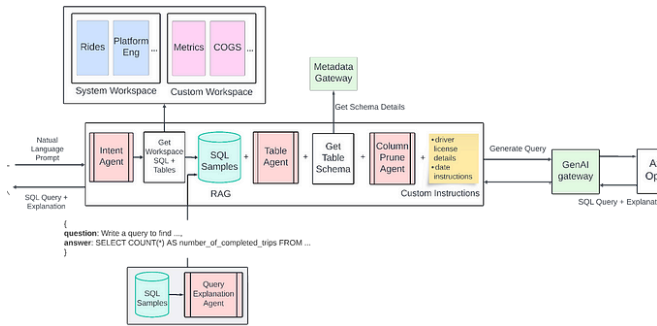
**Stories to Help You Level-Up at Work**


19 stories · 916 saves



**Productivity 101**

20 stories · 2716 saves



 In Wren AI by Howard Chi

## How Uber is Saving 140,000 Hours Each Month Using Text-to-SQL — ...

Discover how Uber's Text-to-SQL technology streamlines data queries and learn how to...

Jan 2  405  7  



 Sai Parvathaneni

## Data Quality Checks (DQCs): A Guide for Data Engineers

As a data engineer in financial services, I've learned one critical lesson: the quality of you...

 Dec 8, 2024  62  2  

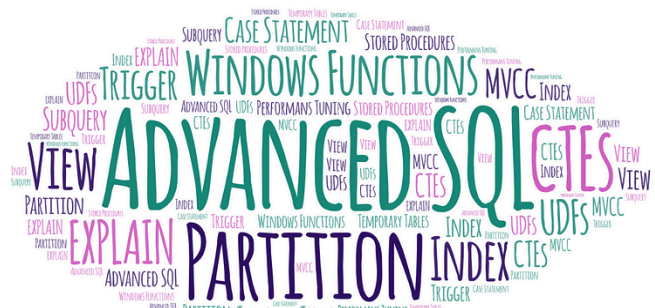


 Varun Singh

## Python 3.14 Released—Top 5 Features You Must Know

Faster Annotations & Mind-Blowing Updates You NEED to Know!

 Dec 31, 2024  613  4  



 Esra Soylu

## Advanced SQL Techniques

In this article, we will focus on advanced SQL techniques. I will try to explain the methods...

Nov 23, 2024  263  3  

See more recommendations