

## Projet ETL Dagster – S7 ISIBD ENSAB

Réalisé par : Hamza Elmourabit

Encadré par : Pr .Lamia Karim

15 décembre 2025



## **Table des matières**

<b>1 Présentation du projet</b>	<b>2</b>
<b>2 Structure du projet</b>	<b>2</b>
<b>3 Code Python (assets/etl<code>assets.py</code>)</b>	<b>2</b>
<b>4 Configuration (workspace.yaml)</b>	<b>2</b>
<b>5 Instructions d'exécution</b>	<b>3</b>
<b>6 Résultats</b>	<b>3</b>

# 1 Présentation du projet

Ce projet consiste à construire un pipeline ETL (Extract, Transform, Load) avec **Dagster**. L'objectif est de lire un fichier CSV (input.csv), effectuer des transformations et sauvegarder le résultat dans output.csv. Le job peut être exécuté manuellement ou via un schedule automatique.

## 2 Structure du projet

```
dagster_etl_scheduled/
  .venv/           # Environnement virtuel
  assets/
    etl_assets.py  # Code du pipeline
  workspace.yaml   # Config Dagster
  input.csv        # Données sources
  output.csv       # Résultat transformé
```

## 3 Code Python (assets/etl\_assets.py)

```
1 import pandas as pd
2 from dagster import asset, job, materialize
3
4 # Extract : charger les données depuis input.csv
5 @asset
6 def load_data():
7     df = pd.read_csv('input.csv')
8     print('Data loaded:\n', df.head())
9     return df
10
11 # Transform : ajouter colonne 'total'
12 @asset
13 def transform_data(load_data):
14     df = load_data.copy()
15     df['total'] = df['quantity'] * df['price']
16     print('Data transformed:\n', df.head())
17     return df
18
19 # Load : sauvegarder les données transformées dans output.csv
20 @asset
21 def save_data(transform_data):
22     df = transform_data
23     df.to_csv('output.csv', index=False)
24     print('Data saved to output.csv')
25     return 'Success'
```

## 4 Configuration (workspace.yaml)

```

load_from :
- python_file :
  relative_path : assets/etl_assets.py

```

## 5 Instructions d'exécution

- Activer l'environnement virtuel :

```
1 & .\venv\Scripts\Activate.ps1
```

- Définir DAGSTER\_HOME :

```
1 $env:DAGSTER_HOME = "$HOME\dagster_home"
```

- Lancer le Webserver Dagster :

```
1 dagster -webserver -w workspace.yaml -p 3001
```

- Lancer le daemon pour le schedule :

```
1 dagster -daemon run -w workspace.yaml
```

- Vérifier le fichier output.csv après exécution.

## 6 Résultats

Après exécution du pipeline, le fichier output.csv contient les données transformées avec la nouvelle colonne total.

TIMESTAMP	OP	EVENT TYPE	INFO
22:28:14,165	save_data	LOADED_INPUT	Loaded input "transform_data" using input manager "io_manager", from output "result" of step "transform_data"
22:28:14,184	save_data	STEP_INPUT	Got input "transform_data" of type "Any". (Type check passed).
22:28:14,210	save_data	STEP_OUTPUT	Yielded output "result" of type "Any". (Type check passed).
22:28:14,264	save_data	HANDLED_OUTPUT	Handled output "result" using IO manager "io_manager" path   C:\Users\khadi\storage\c0565df4-cd4f-4c55-963c-0bf510b31016\save_data\result
22:28:14,283	save_data	STEP_SUCCESS	Finished execution of step "save_data" in 168ms.
22:28:18,456	-	ENGINE_EVENT	Multiprocess executor: parent process exiting after 21.19s (pid: 29992) pid   29992
22:28:18,473	-	RUN_SUCCESS	Finished execution of run for "etl_job".
22:28:18,583	-	ENGINE_EVENT	Process for run exited (pid: 29992).

**dagster**

Overview

**Runs**

Catalog

Jobs

Automation

Lineage

Deployment

Search

Hide navigation

Settings

Support

dagster

Jobs / etl\_job Job in \_\_repository\_\_etl\_job@etl\_assets.py Latest run: 15 déc., 22:27

Overview Launchpad Runs

Search and filter ops

Highlight...

Info Types

Job **etl\_job**

Description

No description provided

Resources

Metadata

Tags

The image shows two screenshots of the Dagster UI. The top screenshot displays the 'Runs' page, listing seven runs for the 'etl\_job'. Each run is identified by its ID (e.g., c0565df4, 0adc5fa5, f91e7caa, 66c2436c, eba47ed5, 57ed3da2), target (etl\_job), launch status (Manually launched or Queued), status (Success or Queued...), creation time (e.g., 15 déc., 22:27, 14 déc., 20:07), duration (e.g., 0:00:21, 0:00:16), and a 'View' button. The bottom screenshot shows the 'Launchpad' page for the 'etl\_job'. It features a graph of ops: 'load\_data' (with an 'Any' input), which connects to 'transform\_data' (also with an 'Any' input), which then connects to 'save\_data' (with an 'Any' input). The right side of the screen provides detailed information about the 'etl\_job', including its description (empty), resources, metadata, and tags.