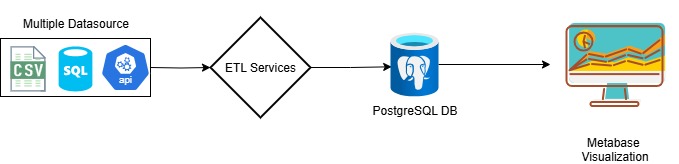
# Project Summary

This project delivers a complete data pipeline to analyze coal mining operations. It extracts data from files and an external API, transforms and validates it, stores it in a database, and visualizes insights in a dashboard using Metabase.

# Pipeline Architecture



**Data Sources and Tools:**

This project uses three main data sources: production\_logs.sql for historical mining logs, equipment\_sensors.csv for daily equipment activity and fuel usage, and the Open-Meteo API to retrieve daily rainfall data based on geographic coordinates. Instead of using the forecast API provided in the case, we use the archive (historical) version because it returns values for past dates. The tools used include Python for building the ETL process, PostgreSQL for storing processed data, Metabase for creating the dashboard, Docker for running services in containers, and Git for version control.

# ETL Process

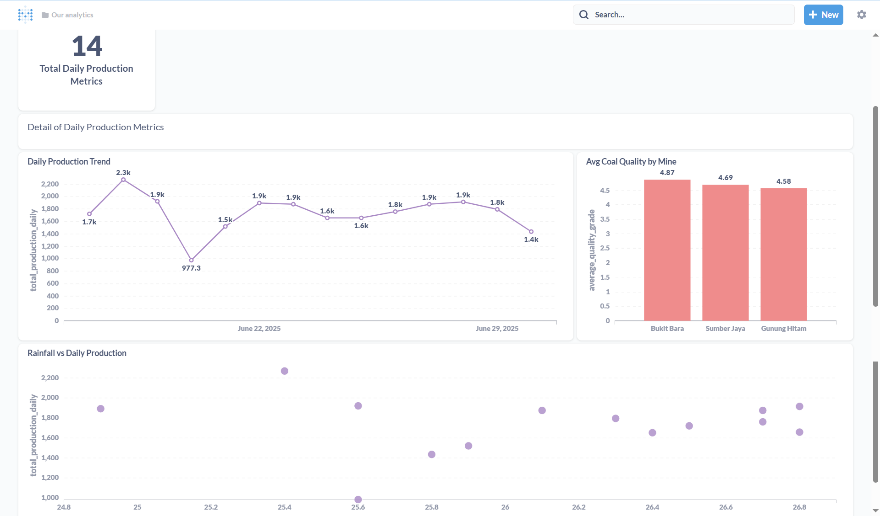
**Extract:**  
In the extraction phase, we load coal production data from a .sql file containing mining logs and sensor data from a .csv file that includes daily equipment activity, fuel consumption, and status. We also collect daily rainfall data using the Open-Meteo API. Instead of using the default forecast endpoint provided in the case, we use the historical archive endpoint because the forecast API does not return values for past dates, while the archive API includes complete historical data.

**Transform:**  
In the transformation phase, we replace negative values in the tons\_extracted column with zero and forward-fill missing equipment data using the previous day’s values for each equipment ID. We calculate several daily metrics, including total coal production, average quality grade, equipment utilization percentage, and fuel efficiency based on fuel consumption per ton mined. Rainfall data is added per day using values returned from the API.

**Load:**  
In the load phase, we store the cleaned and processed results into a PostgreSQL table named daily\_production\_metrics. Any invalid or anomalous records, such as negative production values before correction, are saved into a separate database table called production\_anomalies. We also log validation issues into a file (logs/validation\_errors.log) for reference. We do not use unit tests to handle or catch invalid data or processing errors in this project; instead, the negative production values are flagged and stored directly in the anomaly table.

# Dashboard Overview:

We use Metabase to visualize the final metrics stored in the database. The dashboard includes a line chart showing daily coal production over time, a bar chart comparing average coal quality by mine, and a scatter plot displaying the relationship between rainfall and daily production. A date filter is also added to the dashboard, allowing users to select a specific time range and apply it across all charts.



**Repository and Versioning:**

The code and data used in this project are version-controlled using Git. All components, including the ETL scripts, Docker configuration, and dataset files, are tracked in the repository. The full source code and Docker setup are available at: <https://github.com/enzeeeh/synapsis-challege>