kaggle

Data Source

Zipped File

Unzip File

Pre-processed Files

Note:
Using Python script to process the files to ensure it's formatted correctly for parsing, so that each line stays intact and doesn't get split into multiple lines.

Manually import the csv files

supabase

Hosted

PostgresSQL Database

EXTRACT

meltano

Add Postgres Extractor

Configure an extractor to fetch data from Postgres

Add BigQuery Loader

Configure loader to load Postgres data into BigQuery

LOAD

dbt

Data Modelling

Galaxy Schema

Fact and Dimension Tables

Workflow

- dbt debug
- dbt clean
- dbt deps
- dbt seed
- dbt run
- dbt test
- dbt docs generate
- dbt docs serve

DATA QUALITY TESTING

great expectations

LOG

Local File

Extract

TRANSFORMED & MATERIALIZED MODELS

Google Cloud BigQuery

DATA WAREHOUSE

dagster

ORCHESTRATION

EXPLORATORY DATA ANALYSIS & VISUALIZATIONS

1. Data Ingestion

- Ingest the data to your database/data warehouse
  - E.g Write Python scripts to load the CSV and Excel files into the database tables.
  - Or you can use any **"ingestion" method to ingest the data**

2. Data Warehouse Design

- Design a **star schema** for the e-commerce data
- Create dimension tables (e.g., **DimCustomer, DimProduct, DimDate**) and **fact tables** (e.g., FactSales)
- Implement the schema in your chosen database

3. ELT Pipeline

- You can **use dbt to transform the raw data** into the star schema. (not limited to DBT)
- Implement **data cleaning** and **validation steps**
- **Create derived columns** (e.g., **total_sale_amount, customer_lifetime_value**)

4. Data Quality Testing

- **Use Great Expectations** or custom SQL queries to define and test data quality rules
- Implement **tests for null values, duplicates, referential integrity, and business logic**

5. Data Analysis with Python

When designing a data pipeline, it is essential to consider the end goal: **making the data in your data warehouse accessible and usable for BI analysts, data scientists, and business stakeholders to extract valuable insights.**

- Connect to the data warehouse using **SQLAlchemy**
- Perform simple exploratory **data analysis using pandas**
- Calculate key metrics like:
  - **Monthly sales trends**
  - **Top-selling products**
  - **Customer segmentation by purchase behavior**

6. Pipeline Orchestration (Optional)

Use any orchestration framework to orchestrate the entire pipeline.
Schedule regular runs of the ELT process and data quality checks.
You can use any technology that allows scheduled runs.
This is not limited to:

- **Orchestration** tools (**dagster**, airflow..etc)
- Managed service (e.g. Google Cloud Composer)
- Cron jobs
- CICD via Github actions

7. Documentation

- Document your **code, data lineage, and pipeline architecture** using tools like **DRAW.IO**, EXCALIDRAW to illustrate the architecture of your data pipeline system
- Prepare **a report summarizing the technical approach and your findings / insights, include relevant tables / charts / graphs**
  - **Explain why certain tools were chosen over others**…etc
  - **Explain why you decided to use your particular schema design and how it supports efficient querying** (schema design justification)

Deliverables

1. GitHub repository in a single main branch with all code and documentation
2. Jupyter notebooks with basic analysis
3. Slide deck to present executive summary and key findings

Evaluation Criteria

Focus:

- Accuracy and integrity of the Data Pipeline
- Quality of code and adherence to best practices
- Overall architecture and scalability of the solution
- Good documentation of your overall system - why certain designs and tools are considered

Good to have:

- Depth of data analysis and insights generated

Dagster
├── Step 1: Download & unzip Kaggle dataset
├── Step 2: Run Meltano extract
├── Step 3: Load to BigQuery (or Postgres → BigQuery)
├── Step 4: Run dbt transformations
└── Step 5: Notify / trigger analysis tools

Dagster's role in the pipeline

Pipeline Step	Dagster Role
Data acquisition	Run Python function to download/unzip
Meltano EL	Shell out to `meltano run` or use API
Load to warehouse	Run loaders (Meltano, scripts, etc.)
Run dbt	Use built-in Dagster-dbt integration to run models
Orchestration	Schedule runs, track assets, retry failures
Monitoring	Alert on errors, retry failed steps