

# Module 2 Project

## Overview

In this project, as part of the data-engineering team, you are tasked to build an end-to-end data pipeline and analysis workflow for a company. You'll start with raw data files, load them into a data warehouse, perform ELT processes, ensure data quality, and conduct analysis in Python.

You will present the project solution, key findings, and data driven business recommendations to a mixed audience of business (e.g., CEO, CMO) and technical (e.g., CTO, VP of Engineering) executives. This presentation would translate the technical work and data analysis into a compelling business narrative.

## Project Brief

### 1. Data Ingestion

- Source data (pick any one of these):
  - [Brazilian E-Commerce Dataset by Olist](#)
  - [Instacart Market Basket Analysis Dataset](#)  
*(Unfortunately, the data dictionary is not available for this dataset, but most of the columns are easily interpretable)*
  - [London Bicycles dataset](#) *(Note that this BigQuery dataset is located in EU, hence your DBT project's location has to be set to EU as well)*
- Limit to the core datasets, you do not have to use all of them-
- You are not limited to what you learned in the course; you can use any database technology.
- Ingest the data into 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)

## 8. Executive Stakeholder Presentation

Present the project's architecture, insights, and business recommendations to business and technical executives. This presentation should use the slide deck and optionally any interactive aids.

### Recommended Components

- **Executive Summary:** Concise overview of the problem, solution, and business impact (2-3 minutes maximum)
- **Business Value Proposition:** Clear articulation of business value, efficiency improvements, and strategic alignment, driven by data and insights
- **Technical Solution Overview:** High-level system design with ability to explain key technical decisions without overwhelming detail
- **Risk and Mitigation:** Honest assessment of technical risks, limitations, and mitigation strategies
- **Q&A Handling:** Confident, concise responses that demonstrate deep understanding while remaining accessible

### Presentation Guidelines

- Duration: 10 minutes presentation + 5 minutes Q&A
- Audience: Assume mixed audience of technical executives (CTOs, Engineering Directors) and business executives (CFOs, COOs, Business Leaders)
- Delivery: Recommend all team members to present and be prepared to answer questions
- Visuals: Use executive-friendly visuals (clear charts, architecture diagrams, ROI/kpi metrics) avoiding overly technical details
- Language: Balance technical credibility with business accessibility—avoid excessive jargon but demonstrate technical competence

## 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