FACULTY OF ENGINEERING AND BASIC SCIENCES
ACADEMIC PROGRAM: DATA ENGINEERING AND ARTIFICIAL INTELLIGENCE

COURSE: ETL (G01)
Workshop-1: Data Engineer

1. Introduction

This workshop simulates a **real-world Data Engineer technical challenge**.

The objective is to design and implement a complete **ETL pipeline**, from raw data extraction to loading a Dimensional Data Model (Star Schema) into a Data Warehouse, and finally generating analytical KPIs directly from the DW.

This is not only a coding exercise — it evaluates:

- Data modeling decisions
- ETL logic
- Analytical thinking
- Professional documentation and communication

It will help you understand what companies expect in recruitment processes and allow you to create a **portfolio project** to showcase on GitHub for your future career.

2. Scenario

You have received a CSV file containing **50,000 rows of candidate application data** from technical recruitment processes (randomly generated). Each row represents **one candidate application**.

Your task is to:

1. Design a **Dimensional Data Model (Star Schema)**.
2. Implement an **ETL process in Python**.
3. Load the transformed data into a **Data Warehouse (DW)**.
4. Generate analytical queries and KPIs from the DW (not from the CSV).

3. Proposed System Architecture

The following diagram illustrates the high-level architecture expected for this workshop. It represents the complete ETL data flow, from raw data ingestion to analytical consumption.

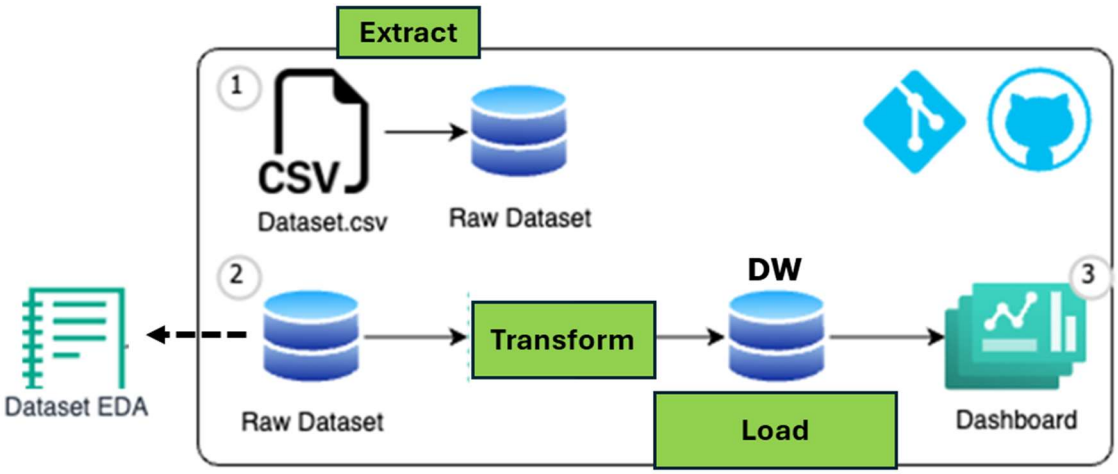

Figure 1. Proposed System Architecture

4. Technologies

You must use:

- Python
- Jupyter Notebook
- SQL
- A Data Warehouse of your choice (SQLite, PostgreSQL, BigQuery, Snowflake, etc.)
- BI Tool (Power BI, Tableau, etc.)

For this workshop, you must follow an **ETL approach**: All transformations must be performed in Python before loading into the DW.

5. Data Description

The CSV file contains 50k rows with the following fields:

- First Name
- Last Name
- Email
- Country
- Application Date
- Yoe (years of experience)
- Seniority
- Technology
- *Code Challenge Score*
- *Technical Interview Score*

Business Rule

A candidate is considered HIRED if: Code Challenge Score ≥ 7 AND Technical Interview Score ≥ 7

You must implement this logic during the transformation phase.

The following figure shows a small sample of the data.

candidates.csv

First Name	Last Name	Email	Application Date	Country	YOE	Seniority	Technology	Code Challenge Score	Technical Interview Score
Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway	2	Intern	Data Engineer	3	3
Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama	10	Intern	Data Engineer	2	10
Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus	4	Mid-Level	Client Success	10	9
Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea	25	Trainee	QA Manual	7	1
Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar	13	Mid-Level	Social Media Community Management	9	7
Alec	Abbott	juanita_hansen@gmail.com	2019-08-17	Zimbabwe	8	Junior	Adobe Experience Manager	2	9
Allison	Jacobs	alba_rolfson27@yahoo.com	2018-05-18	Wallis and Futuna	19	Trainee	Sales	2	9
Nya	Skiles	madisen.zulauf@gmail.com	2021-12-09	Myanmar	1	Lead	Mulesoft	2	5
Mose	Lakin	dale_murazik@hotmail.com	2018-03-13	Italy	18	Lead	Social Media Community Management	7	10
Terrance	Zieme	dustin31@hotmail.com	2022-04-08	Timor-Leste	25	Lead	DevOps	2	0
Aiyana	Goodwin	vallie.damore@yahoo.com	2019-09-22	Armenia	24	Intern	Development - CMS Backend	4	9

Figure 2. Small sample of the data.

6. Task Breakdown

Task 1: Dimensional Data Model (DDM)

Design a **Star Schema** including:

- One Fact Table
- Dimension Tables
- Surrogate Keys for each dimension
- Clear definition of the **grain** of the Fact Table

You must provide:

- A diagram (image)
- A written explanation justifying your design decisions

Important:

- Do NOT use natural keys from the CSV as primary keys in the DW.
- You must explicitly define the grain (e.g., one row per candidate application).

Task 2: ETL Process

You must implement the full ETL pipeline in Python.

Extract

- Load the CSV file.
- Validate data types.

Transform

- Apply the "HIRED" rule.
- Handle null values or inconsistencies (if applicable).
- Create dimension tables.
- Generate surrogate keys.
- Build the Fact Table aligned with your defined grain.

Briefly describe any assumptions or data quality validations applied.

Load

- Insert dimension tables into the Data Warehouse.
- Insert the Fact Table.
- Ensure referential integrity.

Task 3: KPIs & Visualizations

All reports must be generated **from the Data Warehouse**, not from the CSV.

You must create SQL queries and visualizations for:

1. Hires by Technology
2. Hires by Year
3. Hires by Seniority
4. Hires by Country over Years (Focus on USA, Brazil, Colombia, Ecuador)
5. Two additional KPIs of your choice, examples:
    - Hire rate (%)
    - Average scores
    - Hires by experience range
    - Seniority distribution
    - Technology success rate
    - Etc.

Visualizations must clearly represent the information.

Task 4: GitHub Deliverables

Your repository should include:

- **ETL Code** (code for Extract, Transform, Load).
- **Star Schema Diagram** (image + explanation).
- **Visualizations** (e.g., exported as screenshots).
- **README.md** explaining your project, setup instructions, and key decisions.
- **.gitignore** file.

README Requirements

Your README must explain:

- Project objective
- Star Schema design decisions
- Grain definition
- ETL logic
- Data quality assumptions
- How to run the project
- Example outputs

7. Recommended Project Structure

```
etl-workshop-1/
│
├── data/
│   ├── raw/
│   │   └── candidates.csv
│   └── processed/
│
├── notebooks/
│   └── eda.ipynb
│
├── sql/
│   ├── create_tables.sql
│   ├── load_tables.sql
│
├── diagrams/
│   └── star_schema.png
│
├── src/
│   ├── extract.py
│   ├── transform.py
│   ├── load.py
│   └── main.py
│
├── README.md
├── requirements.txt
└── .gitignore
```

8. Evaluation Criteria

Workshop: 80%

Item	Description	Weight
GitHub Repo Setup	Clean, organized structure	0.4
Readme	Clear documentation	1.0
Gitignore	Proper file exclusions	0.2
Dimensional Data Model	Correct Star Schema + justification	1.0
ETL Implementation	Correct transformation & loading	0.6
Data Warehouse Loading	Referential integrity & correct insertion	0.4
Extracting from DW	Queries pull data from DW, not CSV	0.4
KPIs & Visualizations	Correct metrics, accurate, clear charts	1.0

Presentation (Clarity and Structure, Communication and Professionalism): 20%