ICEYE

Senior Data Engineer Assignment

General Instructions

The assignment consists of two tasks. Task 1 is a programming assignment, while Task 2 focuses on system design and operations assessment, with no coding required.

Task 1:

- The final output dataset should be saved as a CSV file in a folder named ***etl_output***.
- You can choose any technology stack or programming language for the task.
- Please include sufficient documentation with your code to explain the rationale behind the transformations.
- The dataset needed for this task is provided with the assignment.

Task 2:

- Ensure that all aspects raised in the assignment are addressed. If you identify any missing dimensions, feel free to add them (bonus points for doing so).
- You are allowed to use GPT for the writing portion, but please indicate if you do.
- The goal is to provide a high-level overview of your solution, rather than extensive documentation. You may include links to additional resources to support your solution and reduce redundancy.

Expected Duration: 2-3 hours

Source Data

- britain_councils - containing the following four files:
    - district_councils.csv
    - london_boroughs.csv
    - metropolitan_districts.csv
    - unitary_authorities.csv

  Each file contains data on different types of local councils and has the same structure with two columns: council, county.
- property_avg_price.csv - Contains data on average property prices by council with columns:
    - local_authority, avg_price_dec_2023, avg_price_dec_2022, difference.
- property_sales_volume.csv- Contains data on sales volume by council with columns:
    - local_authority, sales_volume_nov_2023, sales_volume_nov_2022.

Task 1

Objective: Create a final dataset containing information about each council from the files in the england_councils directory, enriched with additional data from the property_avg_price.csv and property_sales_volume.csv files.

- Read all data from the britain_councils directory, combining data from all four files and add a new column, council_type, based on the file from which each row is taken. For example, rows from district_councils.csv should have council_type set to **"District Council"**.
- Read the data from property_avg_price.csv and property_sales_volume.csv.
- From the aforementioned datasets, perform the following tasks and write output under an output directory as csv files.
    - Create a dataset containing the top 10 authorities with the lowest change in average property prices from 2022 to 2023.
      Target Columns: council, county, council_type, avg_price_dec_2023, avg_price_dec_2022, difference
    - Create a dataset with following columns and rank the councils in descending order based on the percentage growth in sales volume from 2022 to 2023.
      Target Columns: council, county, council_type, sales_volume_nov_2023, sales_volume_nov_2022, growth(%), and rank.

Ensure that each council appears in the final output dataset, even if some columns have no data populated.

Task -2

This task builds on the previous one, focusing on design rather than implementation. **No code is required for this part.**

Now, imagine that the work completed in Task 1 needs to be implemented as a production ETL process.

- Outline the steps you would take to put this pipeline into production. Please include a sample diagram. You can choose a tech stack similar to what you used in Task 1. Your design should address the following aspects:
    - What kind of data quality checks would you implement, and how would they be integrated?
    - How would you design the system to handle larger volumes of data efficiently?
    - How would you ensure that the definitions of calculations are available to analysts, for instance, when visualizing data?
- If the data in property_avg_price.csv is near real-time, how would you design the ETL process to accommodate this while ensuring the final output remains consistent?