Data Platform Engineer
Take Home Test

Please complete this assignment within **5 days**
after the time this email has been sent.

Problem 1: Data Transformation via PySpark

You are required to write a Python script to perform two transformations on a PySpark DataFrame: "flatten all schema except array" and "rename every column to snake case." The transformation rules are as follows:

Transformation: Flatten all schema except array

- When you encounter a nested schema (StructType), flatten it into a first-level column.
- When you encounter an ArrayType, stop the flattening.
- The final result will contain a list of columns where all columns have a primitive type, except for arrays.
- The row count of this transformation should be equal to the raw DataFrame's count, as there is no array explode operation.

Transformation: Rename every column to snake case

- When you encounter a column at any level whose name is not in snake_case format, rename it to snake_case format. For example:
    1. driver_ID -> driver_id
    2. COORDINATES -> coordinates
    3. _Metrics -> _metrics

The template code file (item1.py) is provided. Please write your answer in the flatten_and_rename_to_snake_case function.

Deliverable

1. Create README.md and include guidelines on how to set up and execute the source code.
2. Screenshot of output of your program (capture the output of your function by using Apache Spark API df.printSchema())
3. The complete code for Item_1_answer.py.

Problem 2: Welcome to Data Platform

You are in the first phase of adopting a data platform for your organization. The end goal of this data platform is to support data usage at terabyte scale and also accommodate several query patterns from users. For example, Team A might execute queries for reports every day at 10 a.m., while Team B might need to execute queries to monitor sales performance every hour. Regarding user queries, assume that 100 queries will be sent to your data platform every time users require a report. This problem requires you to design and implement the data platform to serve these use cases. Please also be aware of the cost during the design. The constraints of this data platform are described below:

Constraints

1. The data platform must support Apache Iceberg as the main table format.
2. The Query Engine choice is up to you.
3. Please demonstrate one data pipeline that performs ETL of external data into this data platform and stores the data as an Iceberg table.
4. Please demonstrate how users will connect and query the data from your data platform.

Bonus for Those Who Want to Take A Challenge

1. Data platform is deployed on Kubernetes or data platform adopts the containerization technology.
2. How to monitor your data platform.
3. How security and governance can be done in your platform.
4. Any idea that can improve the data platform in any aspect is acceptable.

Deliverable

1. System architecture diagram, including reasons for choosing this architecture and cost awareness justification.
2. Guidelines on how to set up and deploy this architecture.
3. Guidelines on how to perform ETL.
4. Guidelines on how users can connect to your platform and query the data.

Submission

1. For each problem, please create a folder named problem_<i_answer> where <i_> is the problem number.
2. Place the deliverables of each problem into the corresponding problem_<i_answer> folder. For example, put README.md and item1_answer.py in the folder problem_1_answer.
3. If you have additional tools required to complete the problems, please put everything in a "miscellaneous" folder.
4. Zip all folders and name the resulting file as <Your_First_Name>_<Your_Last_Name>_LMWN_DPE_TakeHomeTest.zip.
5. Submit the zip file.