Data Engineer Mini Project

There are 2 questions in this mini project in which you are supposed to submit a solution. Your solution must meet the following criteria:

1. The code must be available in a Github Repo and shared with us.
2. The code must contain documentation on how to run and where to start.
3. Make sure to include a requirements file if you use Python.
4. We strongly encourage using Boilerplates and templates like Cookiecutter etc.

# Notes

* **Time is an important factor. If you submit the answers sooner, you'll have an advantage.**
* We use Python but feel free to do it in whatever language you prefer.
* **Please make a new commit for each step.**
* When finished, please push to GitHub and send us a link

# Resources :

<https://drive.google.com/open?id=1mZW4cXH8VMog26cQE-TGCPv8nZHmHxGL>

# Question 1:

We have a toy OLTP database named **oltp\_db** which is presented as a set of CSV files. In **oltp\_db**, there are four tables including order\_line, order\_info, customer and product. Our team is in charge of creating a data warehouse in order to answer some business-related questions.

First, you should transfer data from **oltp\_db** to a data warehouse with the Star Schema in a staging environment. In the next steps, you should answer some business questions with efficient **SQL** queries on the data warehouse database.

Given the CSV files, you are expected to implement the following steps:

1. Load CSV files in a Postgresql Database named oltp\_db.
2. Create a Postgresql database named dwh\_db with stars schema using SQL queries.
3. Transfer data from oltp\_db to dwh\_db using SQL queries.

Given a data warehouse you should answer the following question:

1. What is the average order price per customer?
2. How many ‘Vacuum cleaners’ were ordered in New York? Ans who bought most of them in this city?
3. What product is the most popular in each city?

# Question 2:

In this question, you should build an ETL pipeline that extracts two datasets in JSON format from a data lake hosted on MINIO, then processes them using **Spark SQL**, and load the data back in MINIO as CSV files. The datasets are collected by crawling [tweets](https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object) and [users](https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object) from Twitter.

Note: Due to the code complexity, we provide the reading and writing code on MINIO in the Appendix section.

Given the user dataset, you are expected to implement the following steps:

1. Read JSON files from Minio.
2. Select the following fields; id, id\_str, name, screen\_name, location, description, url, protected, followers\_count, friends\_count, listed\_count, created\_at, favourites\_count, statuses\_count, lang, profile\_image\_url\_https, timestamp.
3. Remove duplicate users.
4. Remove space characters from description, name, location, and URL fields.
5. Convert created\_at field to DateTime with (year-month-day) format.
6. Load data in MINIO.

Given the tweet dataset, you are expected to implement the following steps:

1. Read JSON files from Minio.
2. Remove the user field.
3. Remove retweeted\_status and quoted\_status if they are available in JSON objects and add them to dataframe as new rows.
4. Remove duplicate tweets.
5. Remove space characters from text fields.
6. Convert created\_at field to DateTime with (year-month-day) format.
7. Partition dataframe based on created\_at date.
8. Load each partition in separate folders in MINIO. The name of folders should be set according to the partition name.

# Appendix:

For the second question, a python script file is attached to this document. You should change the transform function, MINIO and Spark connection information.

You should run MINIO service on Docker Compose using the following document, then build a specific bucket. After that copy the data to a raw folder and load the final result in a processed folder in MINIO.

<https://docs.min.io/docs/minio-docker-quickstart-guide.html>