Data Engineer / Data Platform Engineer Test (Hands-on)

Please complete this assignment within **3 days**
after the time this email has been sent.

version 2023-07-10

Hands-on test

Problem Description

Source Data to process

1. order_detail.csv

Name	Type	Note
order_created_timestamp	timestamp	format YYYY-MM-DD HH:MM:SS
status	string	
price	integer	
discount	float	
id	string	
driver_id	string	
user_id	string	
restaurant_id	string	

2. restaurant_detail.csv

Name	**Type**	**Note**
id	string	
restaurant_name	string	
category	string	
esimated_cooking_time	float	
latitude	float	
longitude	float	

Business Requirements

- Create two tables in postgres database with the above given column types.
    - order_detail table using **order_detail.csv**
    - restaurant_detail table using **restaurant_detail.csv**
- Once we have these two tables in postgres DB, ETL the same tables to Hive with the same names and corresponding Hive data type using the below guidelines
    - Both the tables should be **external table**
    - Both the tables should have **parquet file format**
    - restaurant_detail table should be partitioned by a column name **dt** (type string) with a static value **latest**
    - order_detail table should be partitioned by a column named **dt** (type string) extracted from **order_created_timestamp** in the format **YYYYMMDD**

Example of dt column

```
order_created_timestamp: "2019-06-08 17:31:57"
dt: "20190608"
```

- After creating the above tables in Hive, create two new tables __order_detail_new__ and __restaurant_detail_new__ with their respective columns and partitions and add one new column for each table as explained below.

Table Name	**New Column Name**	**Logic**
order_detail	discount_no_null	replace all the NULL values of discount column with 0
restaurant_detail	cooking_bin	using esimated_cooking_time column and the below logic

esimated_cooking_time	**cooking_bin**
10-40	1
41-80	2
81-120	3
greater than 120	4

Final column count of each table (including partition column):
1. order_detail = 9
2. restaurant_detail = 7
3. order_detail_new = 10
4. restaurant_detail_new = 8

SQL requirements

- Get the average discount for each category
- Row count per each cooking_bin

CSV output requirements

Save the above query output to CSV files name discount.csv and cooking.csv.

Technical Requirements

- Use Apache Spark, Apache Sqoop or any other big data frameworks
- Use a scheduler tool to run the pipeline daily. Airflow is preferred
- Include a README file that explains how we can deploy your code
- (bonus) Use Docker or Kubernetes for up-and-running program

Question output

1. Source code
2. Docker, docker-compose, kubernetes files if possible.
3. README of how to test / run