Data Engineer Code Test - Python

Last Modified: Nov 21, 2023

Instructions:

This code test is meant to get an understanding of your python skills. The solution code files and any data output files need to be posted to a public GitHub repository owned by you. Please commit directly to the master branch. Commits can be made as often as needed. When complete, share a link to your repository with your recruiter.

You are not required to complete all modules of this test. You should select at least TWO modules to complete. Completing more than two is entirely optional.

Problem Statement:

Zomato, an online food-ordering platform, receives data files daily consisting of data pertaining to existing restaurants or new restaurants. The files are location specific. We would like to automate this process by designing a data pipeline that will ingest and process the files daily, check for quality, correct data issues and produce output files with both corrected data and the bad data.

Data File Layout

The data files are in csv format and named using this pattern:
`data_file_20210527182730.csv`.
Particular fields of interest are below:

Field Name	Data Type	Notes
url	char	
address	char	
name	char	Not null
rate	string	
votes	integer	
phone	integer	Not null

location	string	Not null
rest_type	string	
dish_liked	list	It is a comma-separated list in the data.
cuisines	string	It is a comma-separated list in the data.
reviews_list	array/list	

Areas_in_blore.csv Layout

If Module 4 is selected, the areas_in_blore.csv file should be used for validation. Fields included in this file are below.

Field Name	**Data Type**	**Notes**
Area	char	Not null
Taluk	char	Not null
District	char	Not null
State	char	Not null
Pincode	integer	Not null

Data Pipeline Modules

To complete this automated data pipeline, we think the following modules should be incorporated. For the purposes of this code test, select TWO modules to develop. Completing more than two modules is optional. Every module should produce output file(s) in cases where a check fails. See the section below this called, “Output Files”.

1 - File Check Module

The intention of this module is to verify that the files received daily are as expected, prior to processing them. It will read the files daily, from the given source location and check for the below requirements. Any file not meeting these requirements should not be processed and be placed in a separate directory.

1. Is this a new file? We do not want to reprocess already processed files. The module should also be able to process more than one unique file per day.
2. Is the file empty? Only process non-empty files.
3. Is the file extension .csv?

2 - Data Quality Check Module

The data inside of the daily files should be of acceptable quality before being processed. In cases where the data can be cleaned and validated, it should be. Records that cannot be corrected should be outputted in a separate file and not processed. This module should include the following data quality checks:

1. Data in the phone field can be validated for correct phone numbers.
    a. Any preceding “+” or spaces should be removed.
    b. Ensure phone numbers are correctly formatted.
2. For those fields that should not be null, check for null values. Consider records with nulls in these fields to be bad records and therefore should be removed.
3. Descriptive fields like address, reviews_list can be cleaned by removing special characters or junk characters, etc.
    a. The field data can be split and stored in two separate fields e.g. contact number 1 and contact number 2 for easy readability and access.

3 - Custom Data Quality Check Module

Create your own data quality checks/validation. Using any of the available fields in the data file, come up with at least two checks that are not listed above (i.e. check for duplicate data, etc.). Records that do not meet the criteria for the check should be outputted to a separate file.

4 - Location Validation Module

Validate the location field for correctness by doing a lookup against the Areas_in_blore.csv file. Records that do not validate should be outputted in a separate file.

Output Files

After the cleaning/validation operation, the data needs to be written into files having below output format.

- Capture all the clean records to a .out file.
- Capture all the bad records in a .bad file.
- (**Optional**) For the .bad file create a metadata file which will contain the following fields:
    1. Type_of_issue - this is a short keyword for the type of non-conformity.
    2. Row_num_list - list of all the row numbers which have the issue.

  For e.g. if there are null records found in the dataset then type_of_issue field will have the value “null” and row_num_list will contain the list of all the row numbers which have the issue.

Files You'll Need

- data_file_20210527182730.csv
- data_file_20210528182554.csv
- data_file_20210528182844.csv
- Areas_in_blore.csv