ETL use-case requirement

Souce

DB : create db (bikestore) ⇒ schema (general) ⇒ orders+order\_items (create empty table and use COPY to load csv directly inside)

DL : create folder (bikestore\_dim) ⇒ rest of files (folder with same name and file inside it)

EX : bikestore\_dim/staffs/staffs.csv

API

#########################################################################

Folder (Extraction): 3 notebooks (DB / DL / API) read all files + API : get rateexchange of only EGP with the timestamp and save it in landing under “exchange.csv”

**add new column “extract\_time” = now() timestamp**

**Add new column “source” = datalake / postgres / api**

and write it to new folder named (Landing) as csv EX : landing/staffs/staffs.csv

In landing you should have 10 csv files

def extract\_file(csv-file, source):

Read ⇒ path(csv-file)

Add the 2columns ⇒ source

Write ⇒ path(csv-file)

#########################################################################

Folder(DQcheck) : check :

Create notebook for each file READ FROM LANDING

For all 10 csv files ⇒ nulls in required field / duplicates rows /

For order ⇒ order date range (2000:2024)

For order item ⇒ price within normal range (200:10000) + item\_id is INT/float and available in products

DQ IS DONE WRITE ALL THE FILES WHETHER YOU APPLIED CHANGES OR NOT IN NEW FOLDER “staging\_1” FOLLOWING SAME “staging\_1/staffs/staffs.csv”

Folder(Transformation) :

-use pd.merge/new column to join the data coming from rateexchange (latest value for EGP column) with

Data coming from order\_items the add new columns “list\_price\_egp” as a calculated column using the list price and egp exchange rate

-in “orders” data add new column “late\_delivery” as a boolean type that would be TRUE if the item got delivery later than the required date or before it and FALSE it’

+add new column “latency days” to calculate how many days the order is delayed and if not leave it as NULL value

+add new column “weekend” as boolean type that would be TRUE if the delivery date is SUN/MON

-In “customers” data , add new columns to “local” as a boolean type that would be TRUE if the customer lives in a city where there’s a store and FALSE if not (use data coming from stores.csv)

\*\*You have an ambiguous column in “orders” called “order\_status” , after reaching out to the client he replied with an email containing this information :

*Order status: 1 = Pending; 2 = Processing; 3 = Rejected; 4 = Completed*

How can you add a new table as a lookup to represent the status? **USING PYTHON**

(Please don’t create a text file and write the data in it :) )

This table should be written in staging\_2 with the other files

TRANSFORMATION IS DONE WRITE ALL YOUR FILES WHETHER YOU APPLIED CHANGE OR NOT IN NEW FOLDER staging\_2 FOLLOWING SAME “staging\_2/staffs/staffs.csv”

#########################################################################

Folder(Modeling)

Merge the three dataset from (orders + order\_items + products) in one file named “orders\_products\_details.csv” in “Information Mart” folder

Use this file to create any sort of visualizations that makes sense to you (at least 3 plots) in “Vizualization” folder

**In this step you can choose any other data to merge as long as you can extract a useful information from it ,It’s up to you**

Good Luck

BONUS

