Step4, Step5 ReadMe of Guided capstone

Step 4: Analytical ETL

I ran all the steps in a Jupyter notebook in Databrick. **s4_Analytical_ETL.ipynb** shows each step and its result.

I was using Azure blob storage and have mounted the storage object so that we can seamlessly access data without requiring credentials and interact with object using directory and file semantics instead of storage URLs. **s0_mount-container.ipynb** show the script for mounting.

Finally, I created a class, Reporter, which contains all the steps in s4_Analytical_ETL.ipynb.

File	Description
s4_Analytical_ETL.ipynb	Shows the detail and result of each step
s0_mount-container.ipynb	Contains the scripts for mounting
reporter.py	

Step Five: Pipeline Orchestration

At this step, I did following work

1. Create job status table in Postgres database

```
CREATE TABLE job_status(
  job_id  text,
  status text,
  updated_time timestamp
);
```

2. Created class dbConfig and tracker. I also created the main program for ETL report. . It uses Reporter class to do ETL reporting and Tracker class to do the job management.

File	Description
dbconfig.py	class dbConfig contains the postgres database connection information.
tracker.py	It has class Tracker, which maintains the job status table.
ETL_report_main.py	It contains method run_report_etl , which calls class Tracker to populate job status table

3. Testing

   I ran Reporter on date '2020-08-06' twice..

   At first time it succeeded.

   The second time it failed as there was an error (when the scripts tried to create a temp table , the temp table already existed). The files in the ‘output’ folder shows the results:

File	Description
output\job_status.png’	Shows the results in ‘job_status’ table.
log.txt	Shows the log of running the program.