Shah, Subhradeep

SUBHRADEEP.SHAH@GMAIL.COM

Abstract

Extraction or Harvesting the data from World Wide Web and saving it into a database to perform analysis and gain insights from the data.

Web scrapper

USING GCP, PYTHON, MYSQL

REQUIREMENTS

Write a data-scrapper for Amazon which would do the following-  
  
• Manually input five products to your python program.  
• Collect all the relevant product information and store them into database table(s). The data collected should be refined, then structured properly into a well-designed schema.  
• You may choose to have multiple tables as necessary.  
  
Things to include:  
• You may include an ERD for the schema with the challenge submission.  
• Data must be collected through an automated process (i.e. No manual data collection).  
• Data must be efficiently stored and presentable.  
• Be creative in your design as you may.  
• Knowledge of proxy server is important as you may need that to bypass amazon network filter.

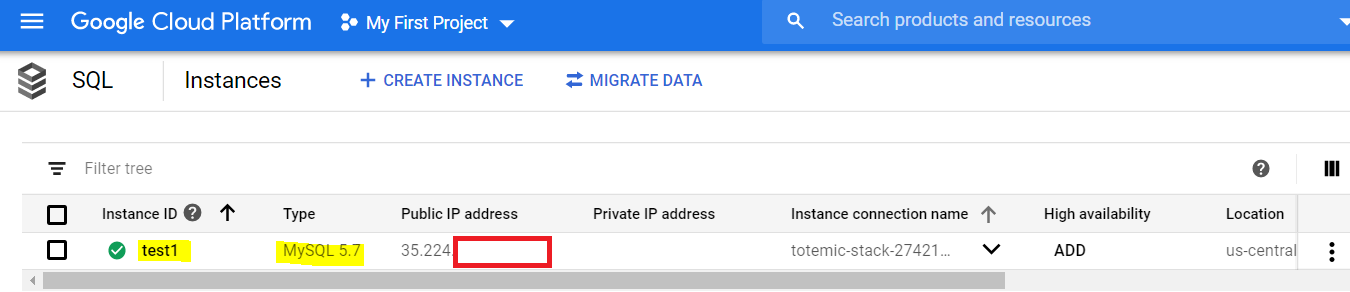
SOLUTION

Technologies used:

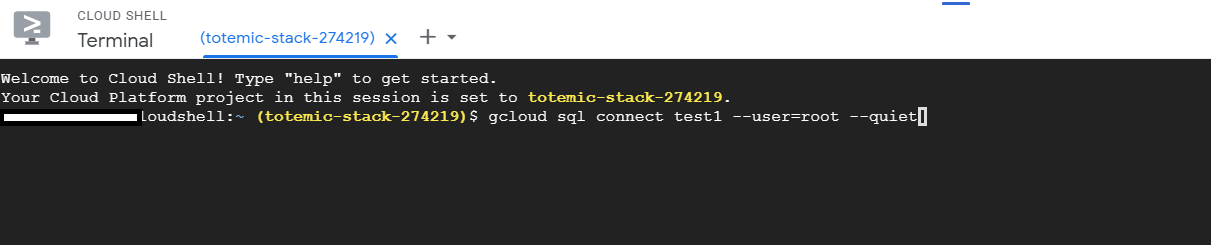
* PYTHON 3
* MYSQL
* GOOGLE CLOUD PLATFORM
  + STORAGE
  + SQL
  + VM INSTANCES (FOR COMPUTE ENGINE)

Execution process:

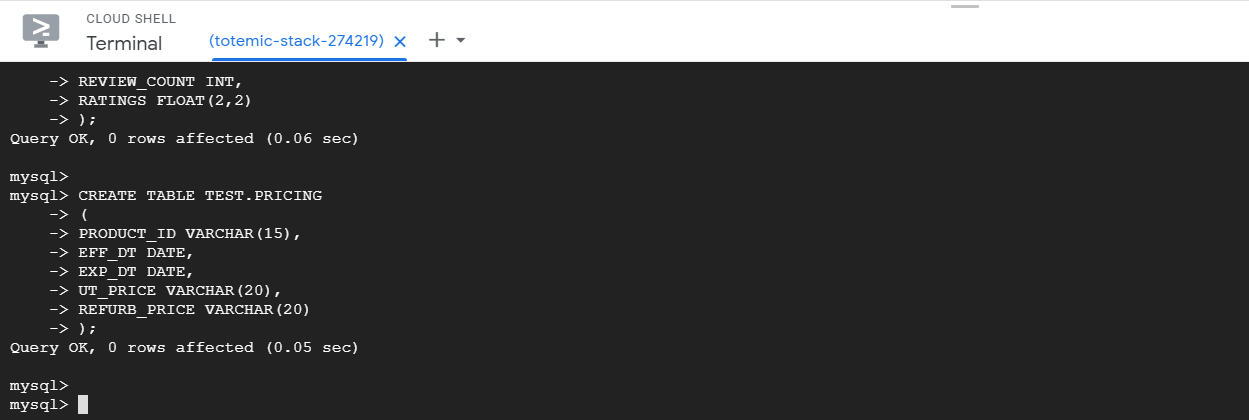
1. Created a gsuite account to login to Google cloud platform (console.cloud.google.com)
2. Created a bucket under storage in GCP to hold the scripts.
3. Create a SQL instance to connect to MYSQL



1. Connect SQL instance to MYSQL by creating user root



1. Created database test and created 5 tables under test



The tables created are

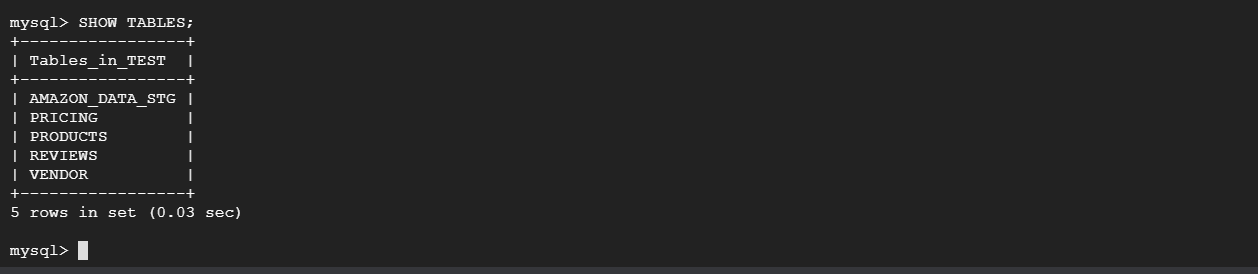
AMAZON\_DATA\_STG – to hold the raw extract of the data from the web scrapper

PRODUCTS – holds the product information, such as PRODUCT\_ID (amazon generated), PRODUCT\_NAME, BRAND, PRODUCT\_IMG, PRODUCT\_DESC

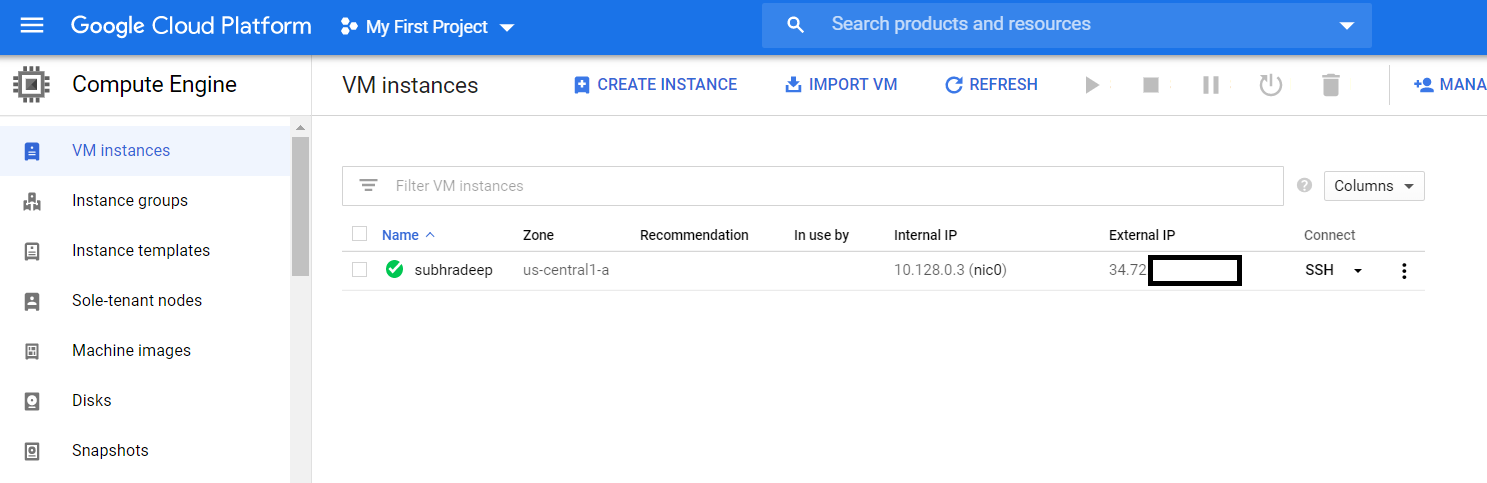
VENDOR – holds the product seller information, such as VENDOR\_STOCK, VENDOR\_NAME, DELIVERY\_MSG

REVIEWS – holds a product customer experience, such as RATINGS, REVIEW\_COUNT

PRICING – holds the pricing details of the product, such as UNIT\_PRICE, EFF\_DT (date from when price is effective), EXP\_DT (till when the file is effective, default 9999-12-31’), REFURB\_PRICE (if the product is second hand, the price would be less)

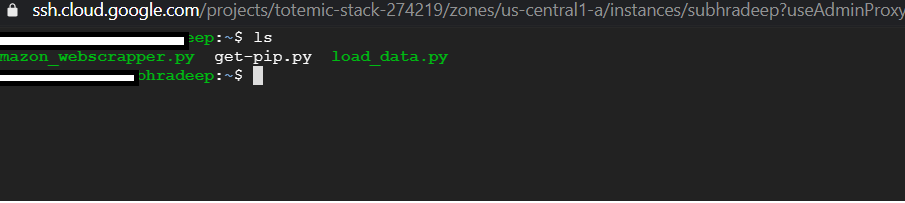


1. Create VM instance to run compute engine, herein I have run my python code.

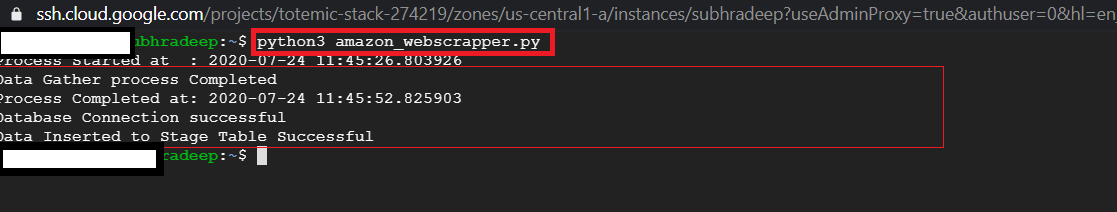


Installed Python libraries, MYSQL, GSUTIL within the VM instance.

Wrote the web scrapper program, most of the code was taken from my personal project of web scrapping (references mentioned within the code). Move the code to bucket in the storage and pull it within the VM.



Execute the amazon\_webscrapper.py program

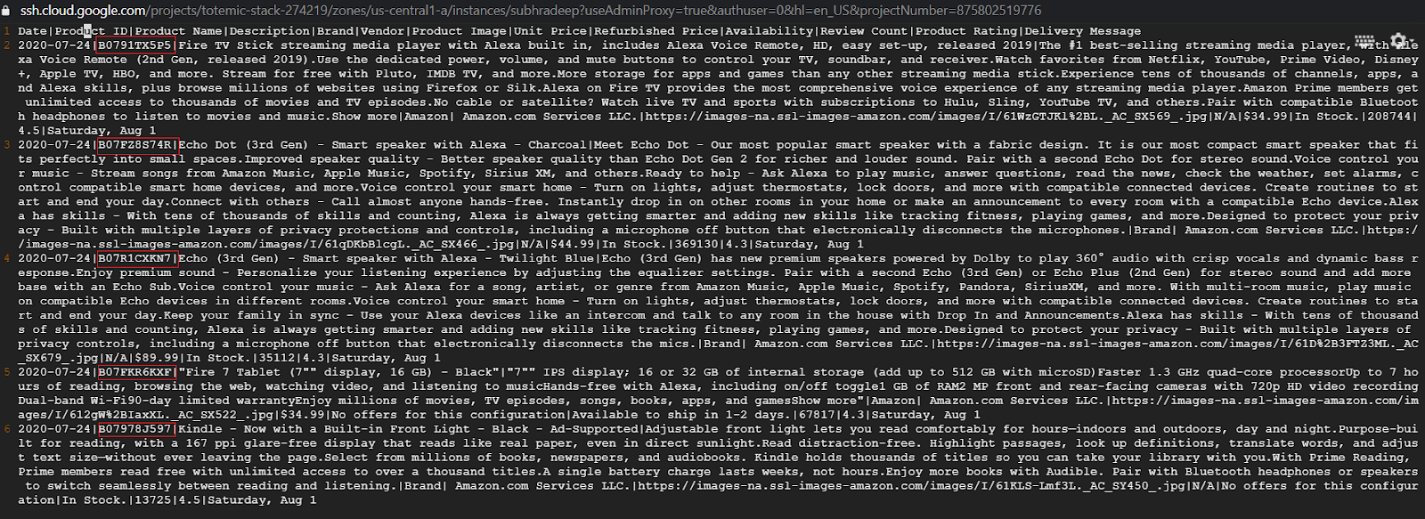


This program has 5 inputs – ECHO, ECHODOT, FIRESTICK, KINDLE, FIRE

These are 5 Amazon products, the code re-directes to [www.amazon.com/\*\*\*\*\*\*\*](http://www.amazon.com/*******), where \*\*\*\* = one of the 5 inputs.

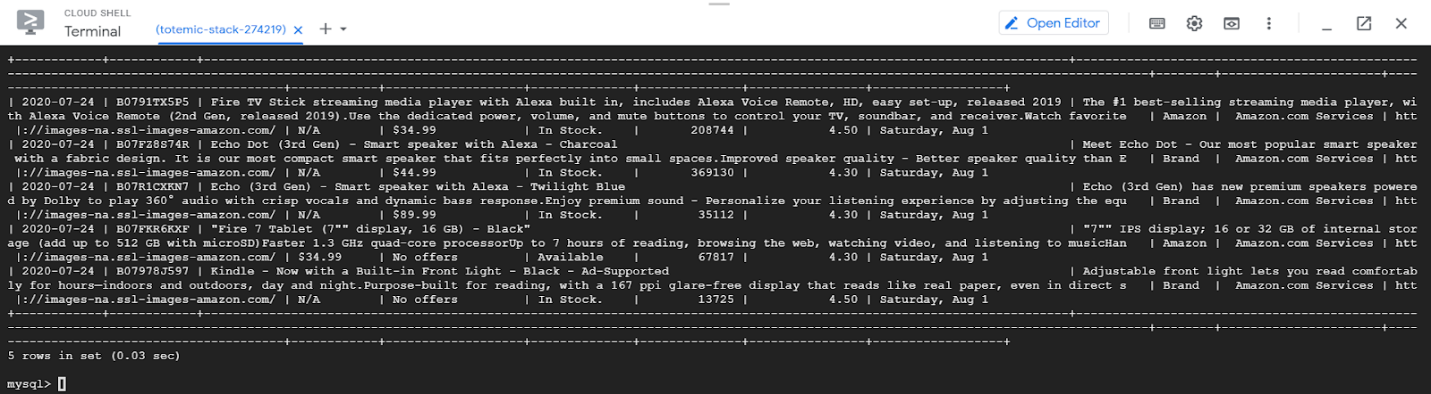
The web scrapper pulls product details from the page and stores it in a file called AMAZON\_PRODUCTS.csv. this file is then loaded to AMAZON\_LOAD\_STG table in MYSQL.

View of the sample file. The first line is the header giving details on the data pulled. The highlighted box points to the product ID used in by amazon for each of the product.



Screenshots of data load in MYSQL tables

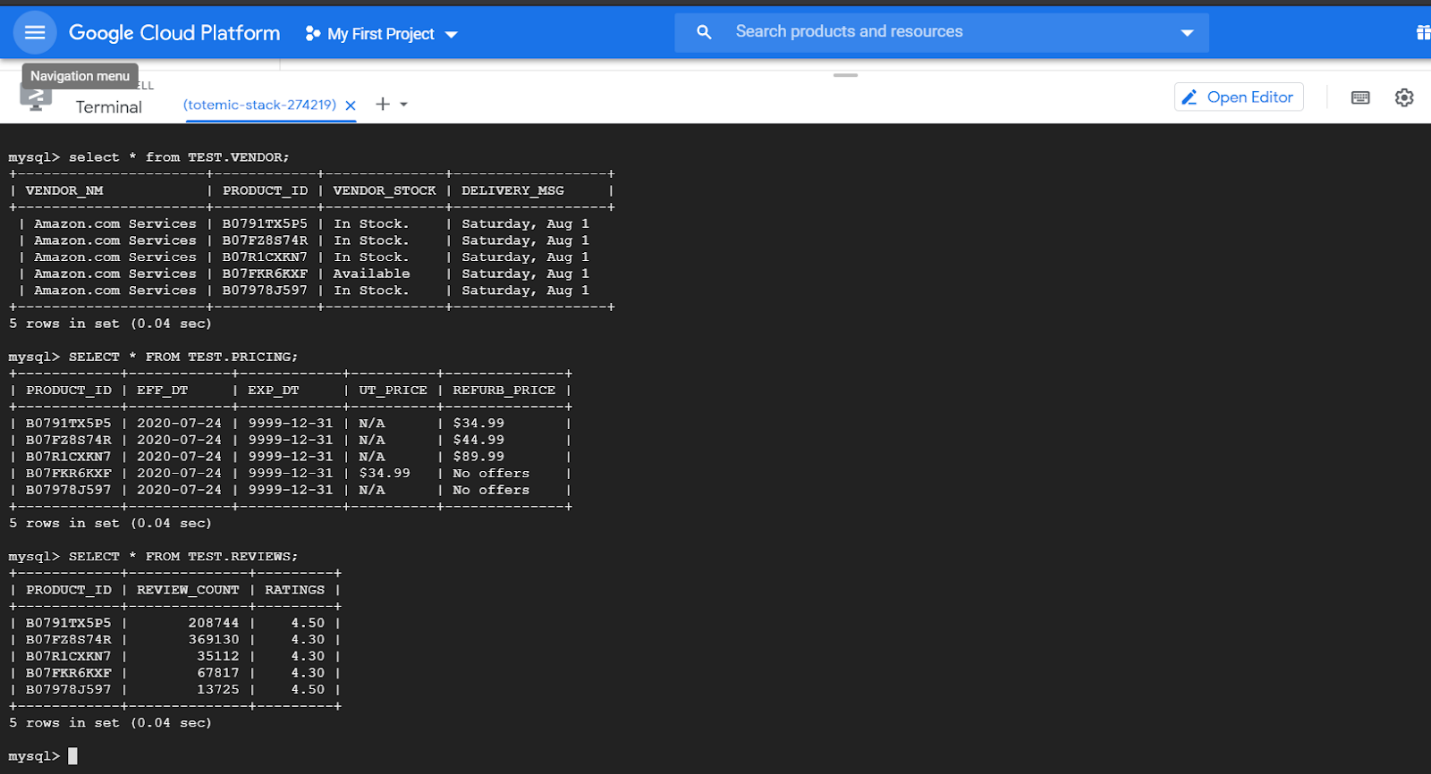
AMAZON\_DATA\_STG:



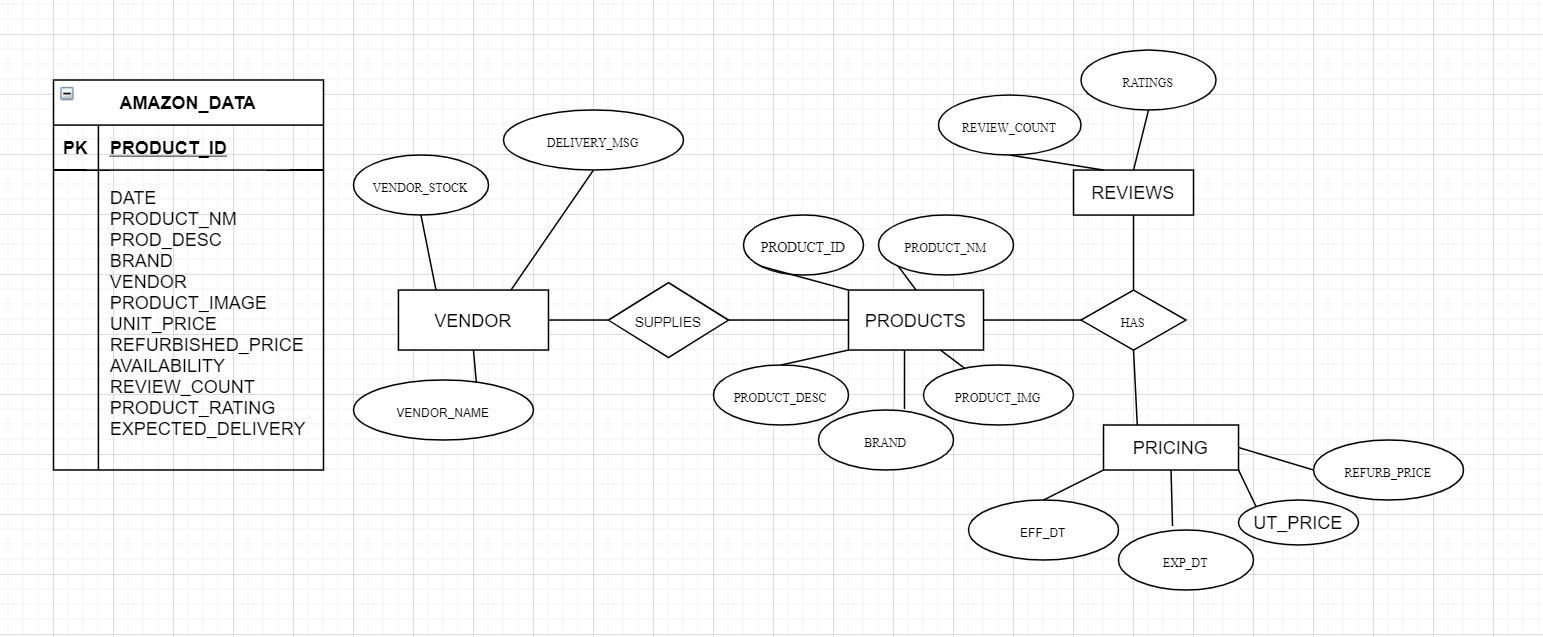
PRODUCTS:



Rest of the tables:



ERD diagram :



Future Enhancements:

1. Currently the 5 product pages are one to one, I would enhance my code to scrapper data from the web page regardless of ‘N’ number of products in single page. I would firstly get the web link of each product and run loop to each weblink to get the data.
2. In Vendor table, I would use VendorID and create another table called VENDOR\_TXT where I would only have VENDORID and VENDOR\_NM. This way if VENDOR\_NM is not redundant in the VENDOR table and would save space in the database.
3. Would better standardize the code to perform better and handle errors.
4. Change the batch mode to streamline it in real time.
5. Currently the code appends the data to file and tables, I would handle the processing better to avoid duplication of data.

