Alooma's CSE Challenge

############################################################################################  
# File Name: Python Code Documentation.docx  
# Author: Sreeram Pulavarthi  
# Date: 05-22-2018  
# Compiler used: Python 3.6  
# Database used: MySQL 5.7  
# External packages Used: mysql, tkinter, requests  
############################################################################################

Install python3.6 package before running the code.

Install the packages using pip installer.

Step1:

Read the engagement API data from the URL below using python script “Data\_Capture.py”

“<https://api.hubapi.com/engagements/v1/engagements/paged?hapikey=demo>&limit=200&offset=”

Pass the offset of the page from the function extract\_data() in the class engagementsCapture().

This function recursively calls “URL” until there is not more data present in URL.

Step2 a:

Once after the response is received, the starts getting written into (.psv) file and into data base table concurrently.

Data base read and write are separately handled defined in the data abstraction layer (DAL) for the engagements information.

Step2 b:

Once after all the records been successfully loaded into the data base the getDataQuery method queries the data base and returns all the values from the records loaded in grouped be counts per engagement and broken by day.

This result set will be passed to the Rolling\_Average.py

The data will be loaded into the table “Engagement\_Data”, DDL is attached for it.

Step3:

MySQL code for the query asked,

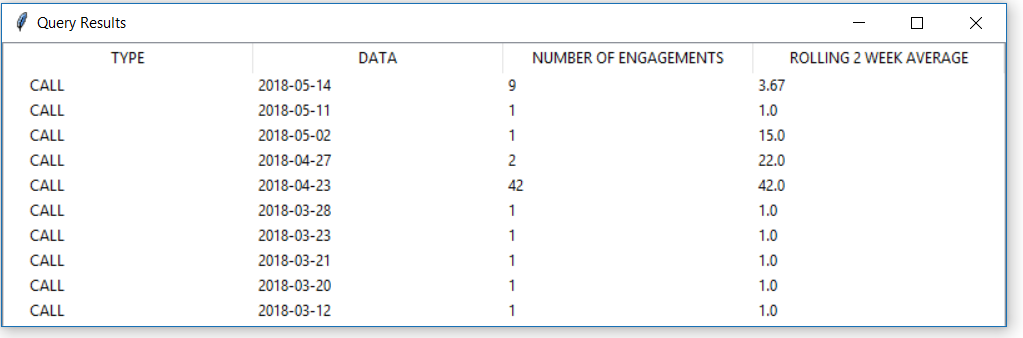
* Counts per day
* Rolling 2-week avg per each day.

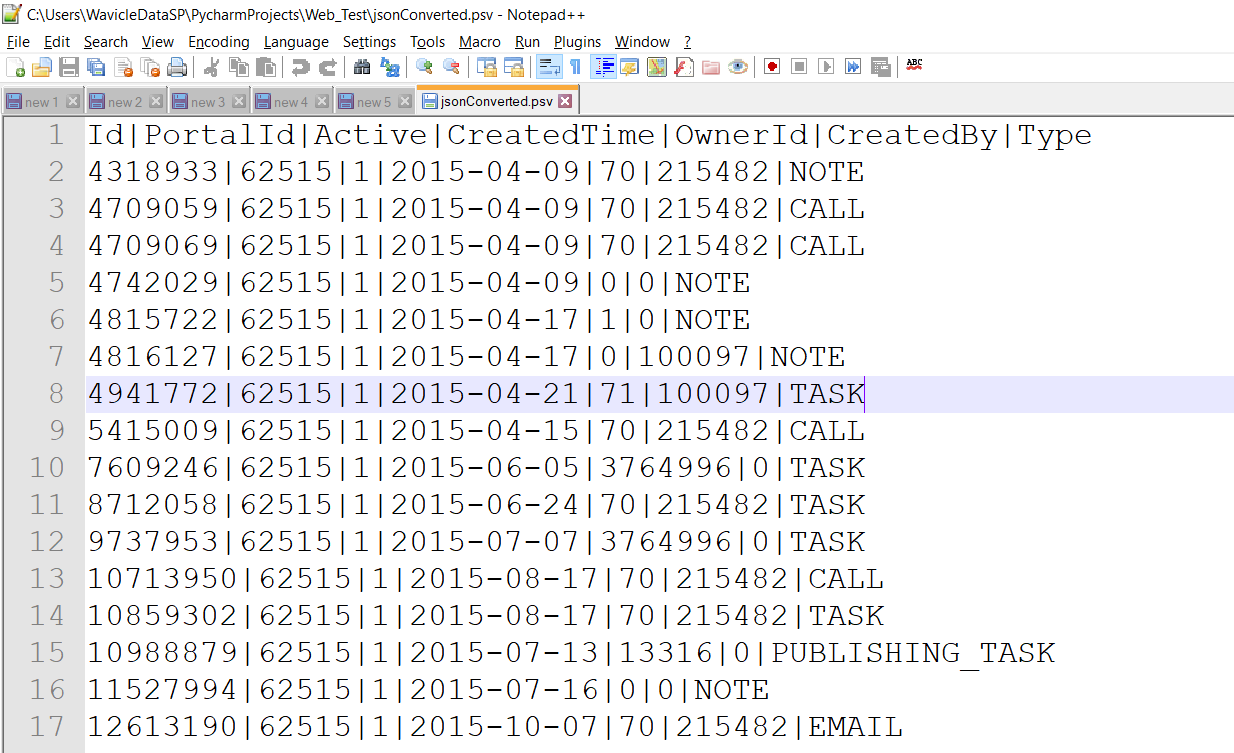
Above mentioned both the results are displayed on both the console and GUI.

Queries will be displayed in the tkinter window also, make sure that package in installed on the machine.



Below are the screen shots for file and GUI:





Python Code is below:



Please place all these 3 files in single folder and make sure the database table is created and then start the python code, python Data\_Capture.py