**SKIP CODE SEGMENT FOR PARAGRAPHED EXPLANATION**

import mysql.connector **import of sql python library**

import sys **unused**

from collections import defaultdict **to work on dictionary**

from geopy.geocoders import Nominatim **importing library required to translate location names into lat lon**

**Creating of MySQL Connection**

cnx = mysql.connector.connect(host="localhost",

user="root",

database='test')

**Declaration of SQL cursor. We cannot execute SQL statements through connection. We need to do it through cursors.**

cur = cnx.cursor()

**Preparation of SQL statements**

sqlStatement = "INSERT INTO STARHUB(S\_ID, UNIX\_TIME, SESSION, LOCATION) VALUES('%s',%s,%s,'%s');"

precheck = "SELECT \* FROM STARHUB WHERE S\_ID = %s AND UNIX\_TIME = %s;"

query = ("SELECT COORD FROM coordinates WHERE LOCATION = '%s';")

insert = "INSERT INTO coordinates(LOCATION, COORD) VALUES('%s','%s');"

**Count variables.**

countRead = 0

countAdded = 0

**See paragraphed explanation**

locationWeHave = []

**Function to translate string location name into lat lon through Python Geopy**

def geopy(location):

geolocator = Nominatim()

location = geolocator.geocode('%s Singapore'%(location),timeout=30) **Timeout period for the location lookup.**

if(location is None): **Nominatim will return object type None if location is unknown**

return('%s,%s'%(0,0))

else:

lat = location.latitude

lon = location.longitude

return('%s,%s'%(lat,lon))

**Function to check if the location was encountered and recorded**

def getLatLon(loca):

if(loca in locationWeHave): **See paragraph explanation for detailed flow**

return

else:

prep = query%loca

cur.execute(prep)

rowCount = 0;

for COORD in cur:

rowCount = rowCount + 1

locationWeHave.append(loca)

if(rowCount == 1):

return

else:

loc = geopy(loca)

prep = insert%(loca,loc)

cur.execute(prep)

cnx.commit()

alreadyhave = True

with open('DATA\_FOR\_RPMASTER.txt') as f:

for line in f:

countRead = countRead + 1

if(countRead % 1000 == 0):

print("read: " + str(countRead))

temp = line.rstrip('\n').split(',')

if(alreadyhave):

cur.execute(precheck,[temp[0],temp[1]])

rowcount = 0;

for r in cur:

rowcount = rowcount + 1;

if(rowcount > 0):

countAdded = countAdded + 1

if(countAdded % 1000 == 0):

print('Line added %s'%(countAdded))

continue

else:

alreadyhave = False

temp[3] = temp[3].replace("'","\\'")

getLatLon(temp[3])

try:

cur.execute(sqlStatement%tuple(temp))

cnx.commit()

countAdded = countAdded + 1

if(countAdded % 1000 == 0):

print('Line added %s'%(countAdded))

except IOError:

cnx.rollback()

print(sys.exc\_info()[0])

cur.close() **Closing the cursor**

cnx.close() **Closing the connection. Do not leave excessive connections open.**

print("total lines read: " + str(countRead) + " Total lines added: " + str(countAdded))

**Paragraphed explanation:**

Keep in mind throughout that there are two conditions which this script will run under

1. The data had not been loaded at all. This is the first run.
2. Partial data was loaded into SQL table and script was stopped. This is a resuming run.

Script will read Master.txt. And for each row of text translates to one row in our database.

Read count will be displayed in python console for every 1000 lines.

The line will then be stripped of the ending \n (new line literal), and then splitted into their individual components at each comma.

Up to this point it is assumed that this run is a resuming run. A precheck will run and determine if the current row was indeed inserted before. If inserted before it will add 1 count to countAdded, and then skip the rest of the iteration (continue). Else not inserted before the rest of the iteration will continue to proceed with add.

Alreadyhave is the boolean variable that will become false once the txt file was read down to the row last inserted, thereafter with future loop iterations this portion of the insertion script will be skipped.

The reason to escape the single quote is to avoid SQL syntax error, as single quotes are string delimiter in SQL statements. Certain locations (ie People’s Park) contain single quotes.

The script will then check if the location had been inserted before. Why the check was not done in precheck is because if the row was inserted the location check would had already been done for this location.

**getLatLon(loca)**

The location that are detected to have been inserted in this run of the script is cached in locationWeHave list.

Thus if the location exists in this list we can skip the rest.

If not a check will be done against the ‘coordinate’ table. Once we finds it we will store it into the list and skip the rest of the code. If not found still it will **translate the location through geopy function** and insert into ‘coordinate’ table.

**Main flow**

Once all these have been done, the script will proceed and try insert into ‘starhub’ table.