Quy trình xử lí database

Chú ý : Mọi nguồn xử lí trong này đều có trong source code

1. ***Kết nối với database thông qua Python:***

import pandas as pd  
from pytrends.request import TrendReq  
import psycopg2  
import xlsxwriter  
pytrends = TrendReq(tz=360)  
  
# Connect to sql  
con = psycopg2.connect(database="postgres", user="postgres", password="05092001thang", host="localhost", port="2001")  
cur = con.cursor()

1. ***Tạo một hàm tạo bảng để có thể tạo ra bảng mới trong database qua python :***

Trong đó table\_name chính là tên bảng muốn tạo

# Creat a new table in sql  
def create\_table(table\_name) :  
 cur.execute("CREATE TABLE {}(id SERIAL,keyword TEXT NOT NULL,date DATE NOT NULL, Value INT NOT NULL, trend\_type TEXT NOT NULL)".format(table\_name))  
 con.commit()

1. ***Insert các data vào trong từng bảng thông qua hàm trending***

Trong đó : +) geo là khu vực muốn tìm kiếm trending,

+) table\_name là bảng muốn dùng để lưu trữ data

+) excel\_name là tên file excel muốn dùng để lưu trữ dữ liệu(như ở phase 1)

# Get data from Google trends  
def trending(geo, table\_name, excel\_name):  
 writer = pd.ExcelWriter('{}'.format(excel\_name), engine='xlsxwriter')  
 for col in columns\_name:  
 df\_=df[col]  
 # Drop NULL and duplicate values in each column  
 df\_.dropna(inplace=True)  
 df\_.drop\_duplicates()  
  
 # Get list of keywords in each column  
 keywords = df\_.values.tolist()  
 datasets = []  
 for kw in keywords:  
 pytrends.build\_payload(  
 kw\_list=[kw],  
 cat=0,  
 timeframe='2020-01-01 2020-12-31',  
 geo=geo,  
 gprop=''  
 )  
 data = pytrends.interest\_over\_time()  
  
 # Change time format and drop isPartial column, NULL values  
 data.index = pd.to\_datetime(data.index)  
 if not data.empty:  
 data = data.drop(labels=['isPartial'], axis='columns')  
 data.dropna(inplace=True)  
 for i in range(data[kw].values.shape[0]):  
 cur.execute("INSERT INTO {}(keyword, date, value, trend\_type) VALUES('{}', '{}', {}, '{}')".format(table\_name,kw,data.index[i],data.values[i][0],col))  
 con.commit()  
 datasets.append(data)  
  
 # Make complete sheet and put it into excel  
 sheet = pd.concat(datasets, axis=1)  
 print(sheet)  
 sheet.to\_excel(writer, sheet\_name=col.replace('/', '\_'))  
 writer.save()

1. ***Hàm get\_top\_10 : dùng để tìm 10 từ khóa hot nhất :***

Trong đó : +) table\_name là tên của lưu trữ thông tin để tìm kiếm

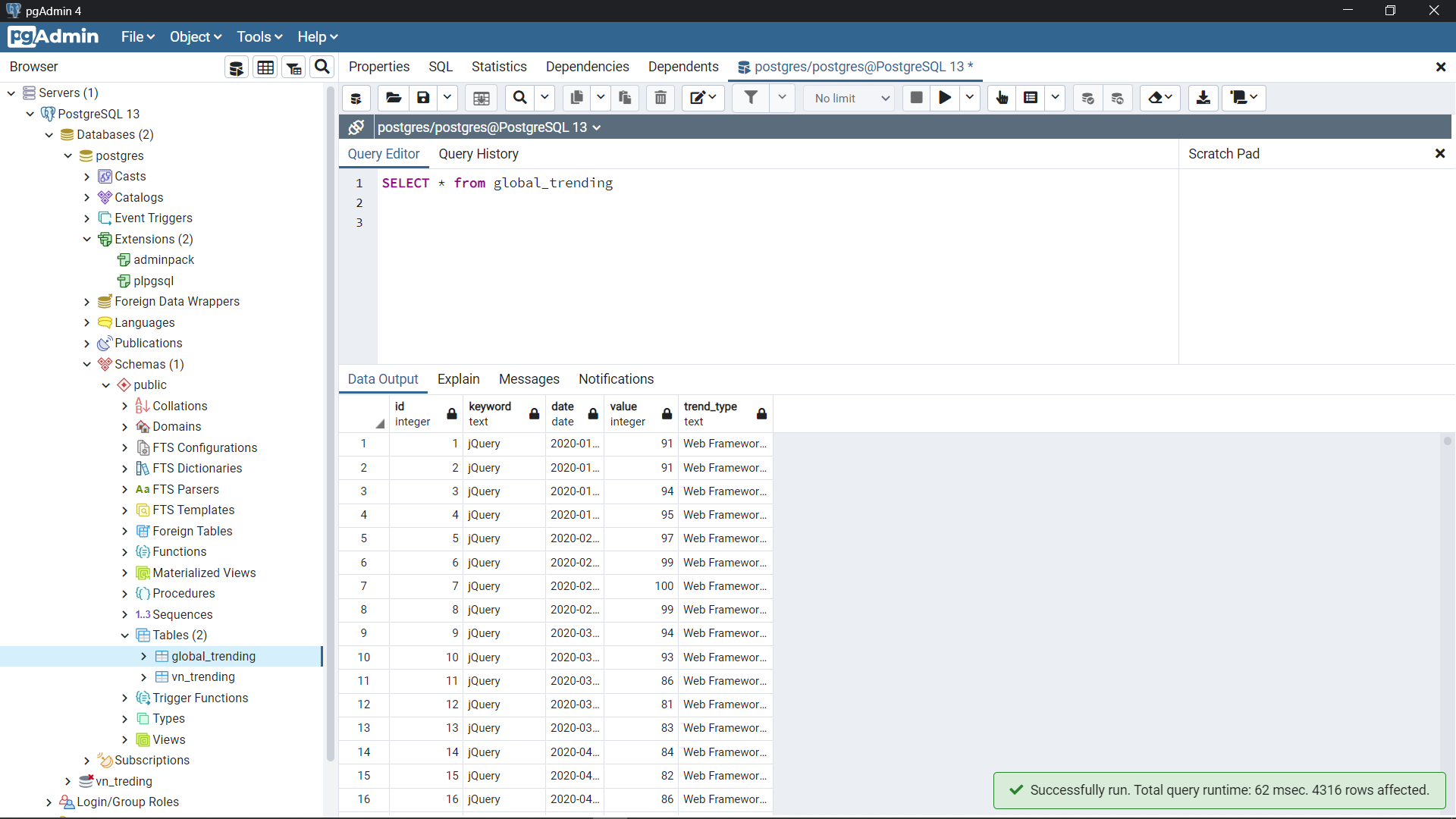
+) excel\_name là tên file excel muốn lưu thông tin top 10 từ khóa tìm được

# Query get top 10 records and put it to excel  
def get\_top\_10(table\_name, excel\_name):  
 cur.execute('''  
 select \*  
 FROM( SELECT   
 keyword,  
 SUM (value)  
 FROM {}  
 GROUP BY keyword ) AS n1  
 order by n1.sum desc  
 limit 10   
 '''.format(table\_name))  
 with xlsxwriter.Workbook('{}'.format(excel\_name)) as workbook:  
 worksheet = workbook.add\_worksheet()  
  
 for row\_num, data in enumerate(cur.fetchall()):  
 worksheet.write\_row(row\_num, 0, data)

1. ***Kết quả khi chạy các câu lệnh :***

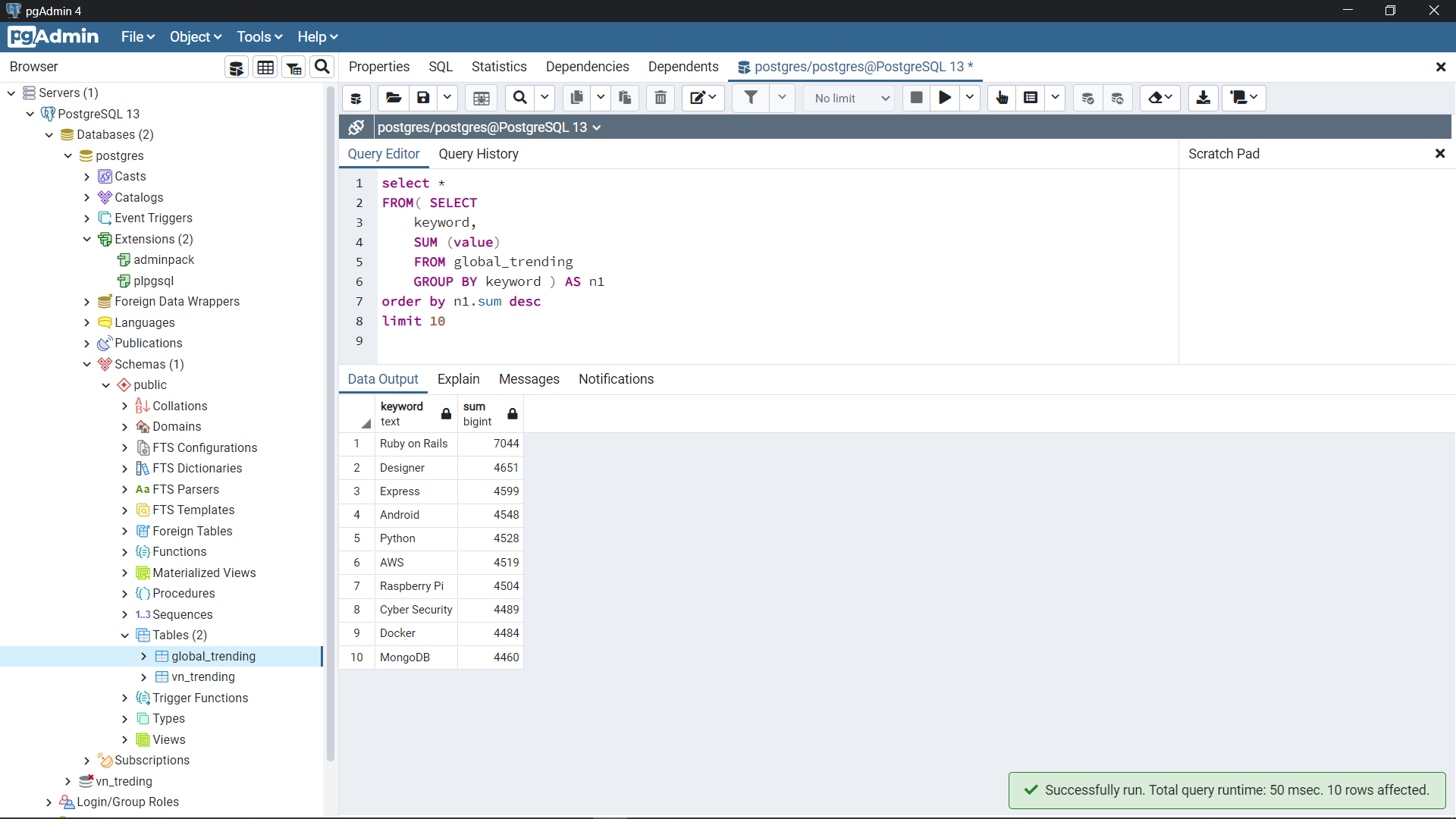
Tạo bảng global\_trending và lưu dữ liệu vào đó với geo =””

# Geo = ""  
create\_table("global\_trending")  
trending("", "global\_trending",'global\_trending.xlsx')



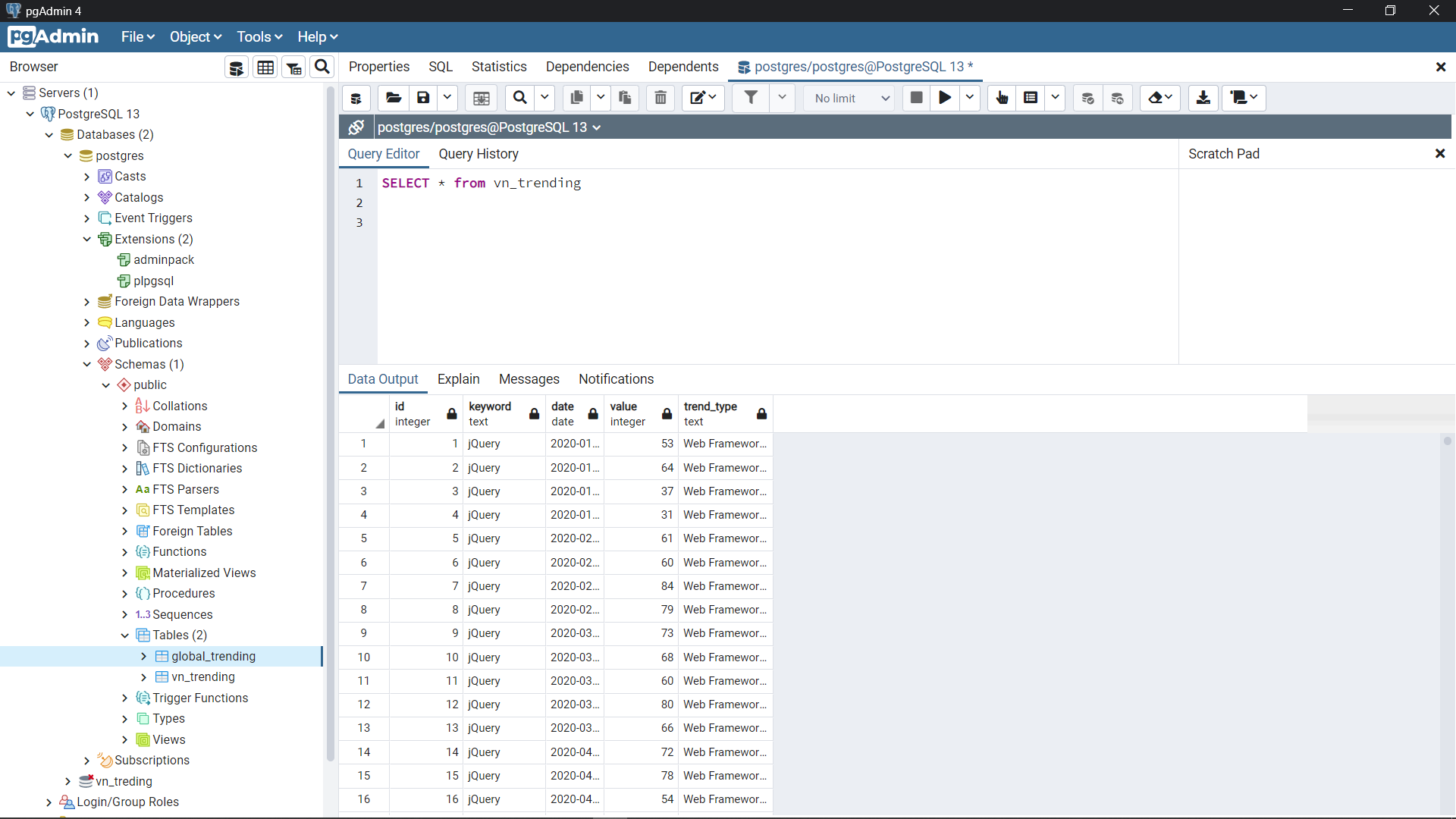
Sau đó gọi hàm get\_top\_10 lấy ra 10 records phổ biến nhất lưu vào vn\_trending\_result.xlsx :

get\_top\_10("global\_trending",'vn\_trending\_result.xlsx')



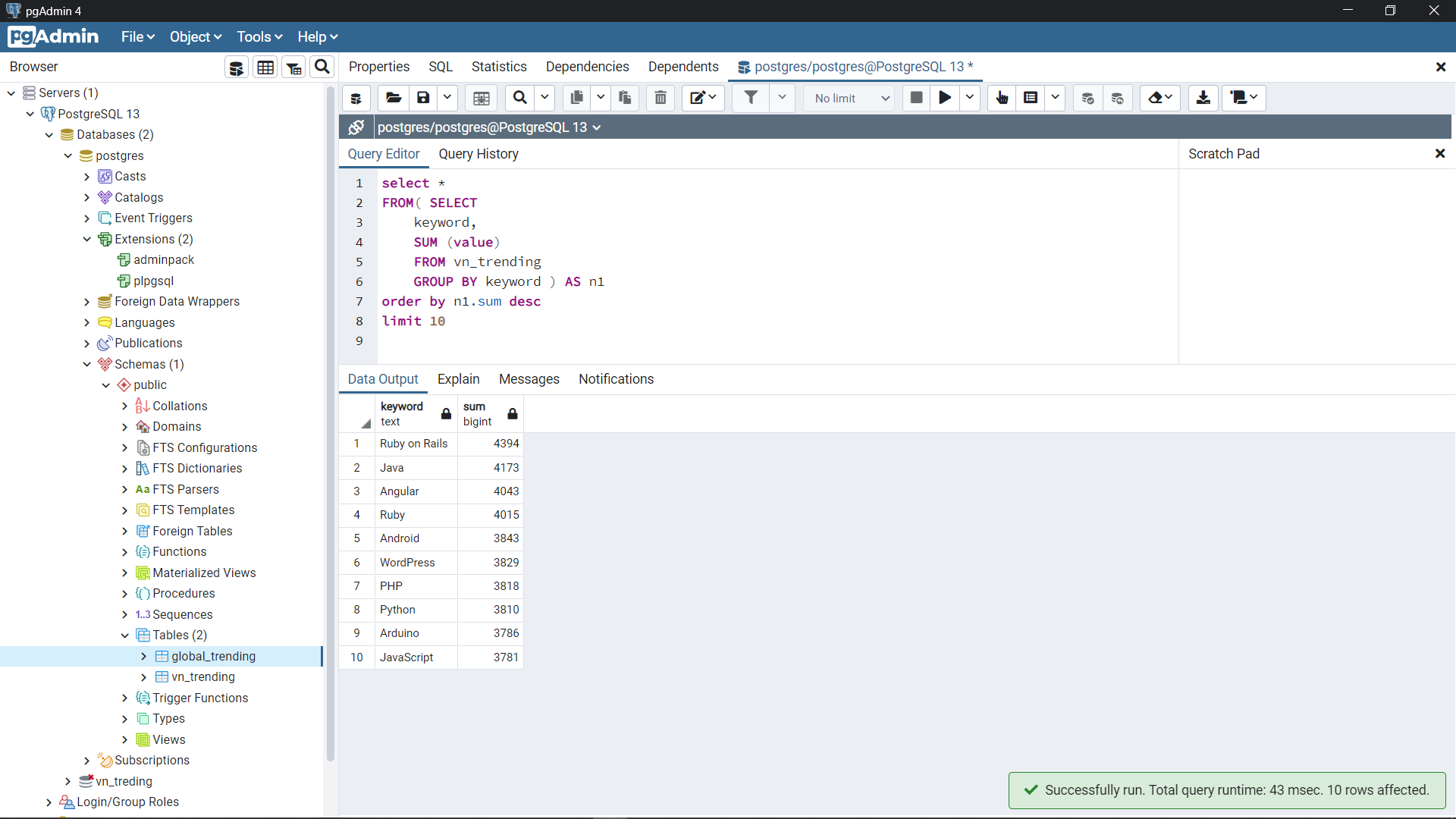
Tạo bảng vn\_trending và lưu dữ liệu vào đó với geo =”VN”

# Geo = "VN"  
create\_table("vn\_trending")  
trending("VN","vn\_trending", 'vn\_trending.xlsx')

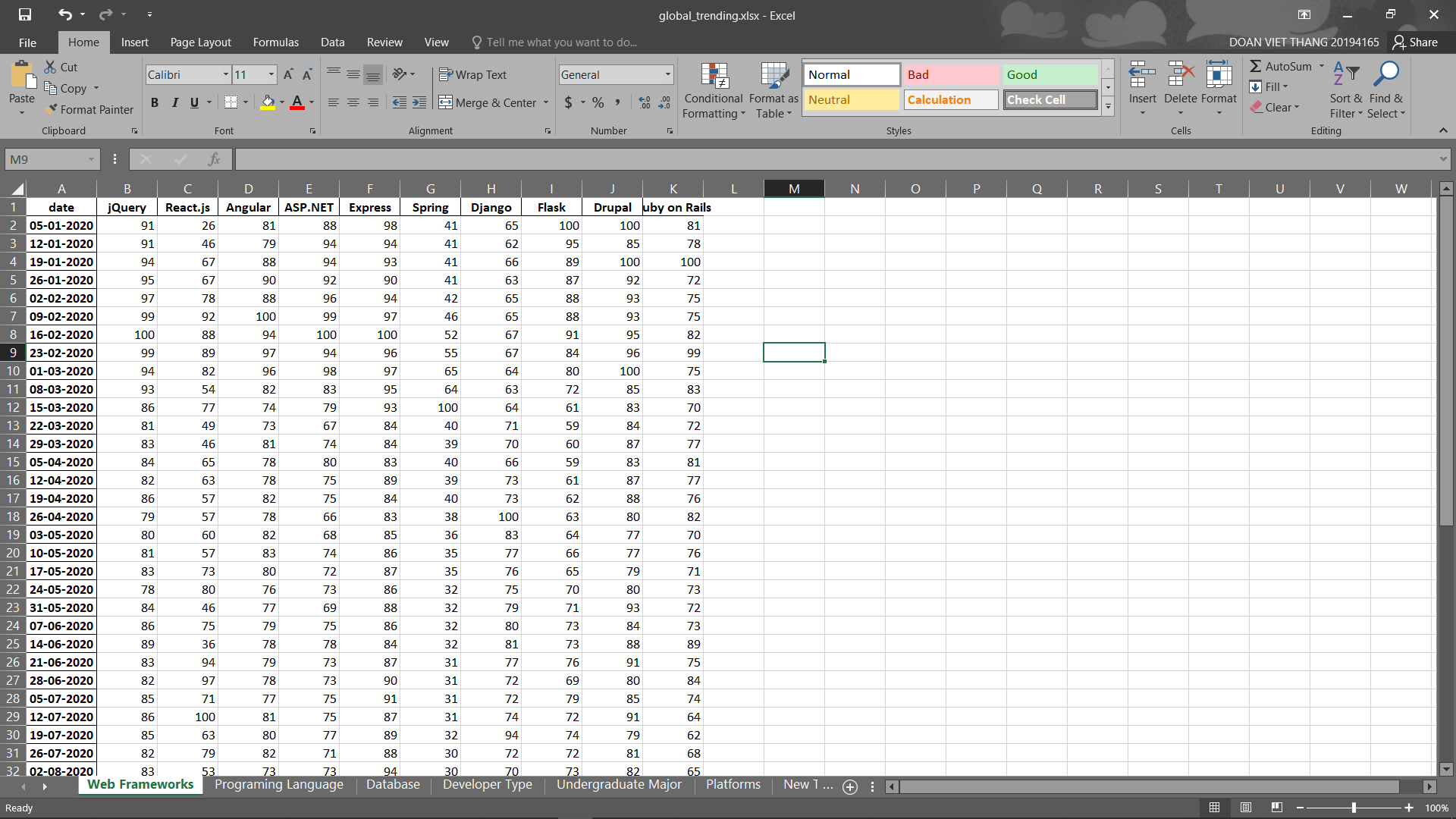


Sau đó gọi hàm get\_top\_10 lấy ra 10 records phổ biến nhất lưu vào vn\_top\_ten.xlsx :

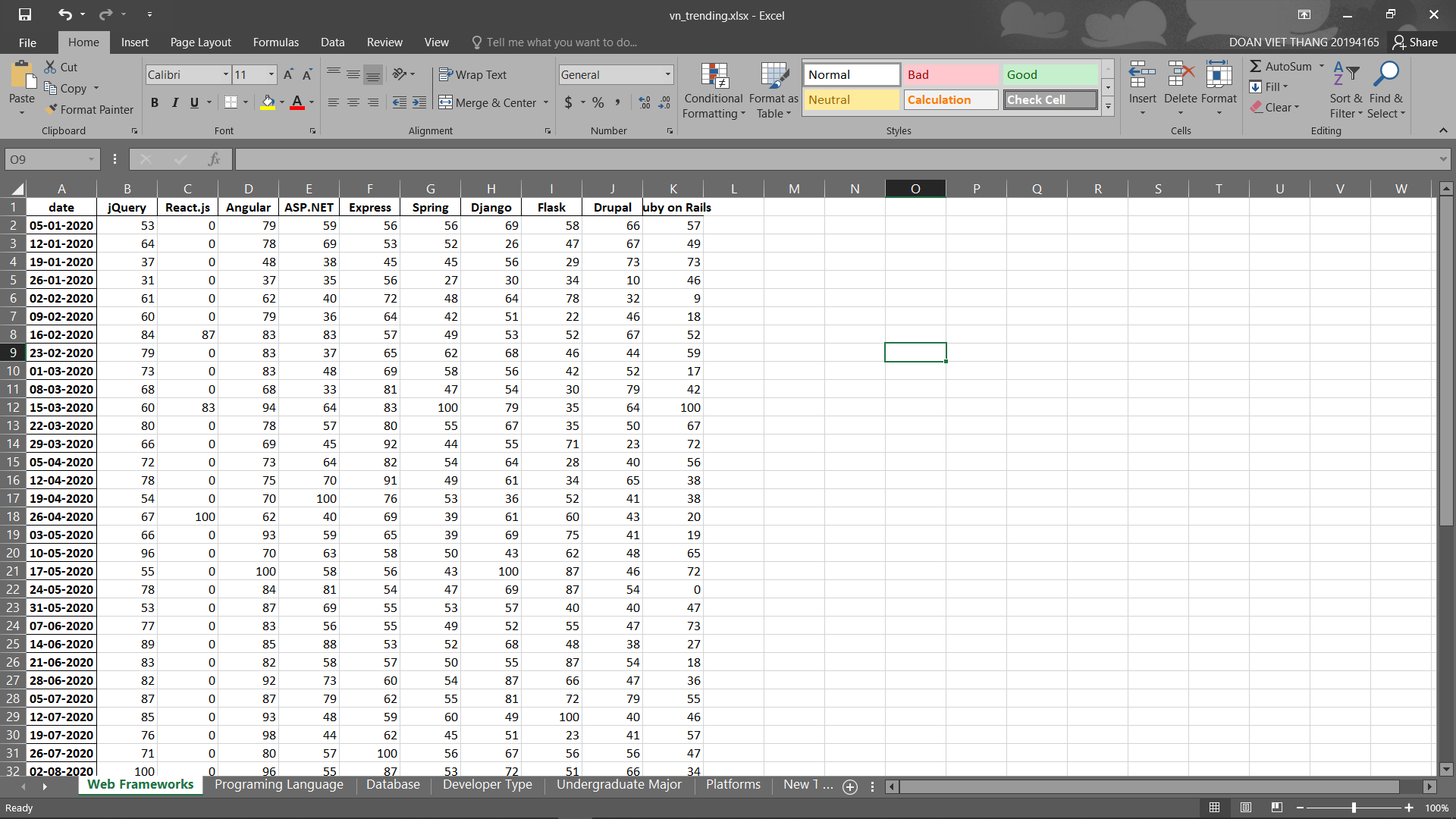
get\_top\_10("vn\_trending", 'vn\_top\_ten.xlsx')



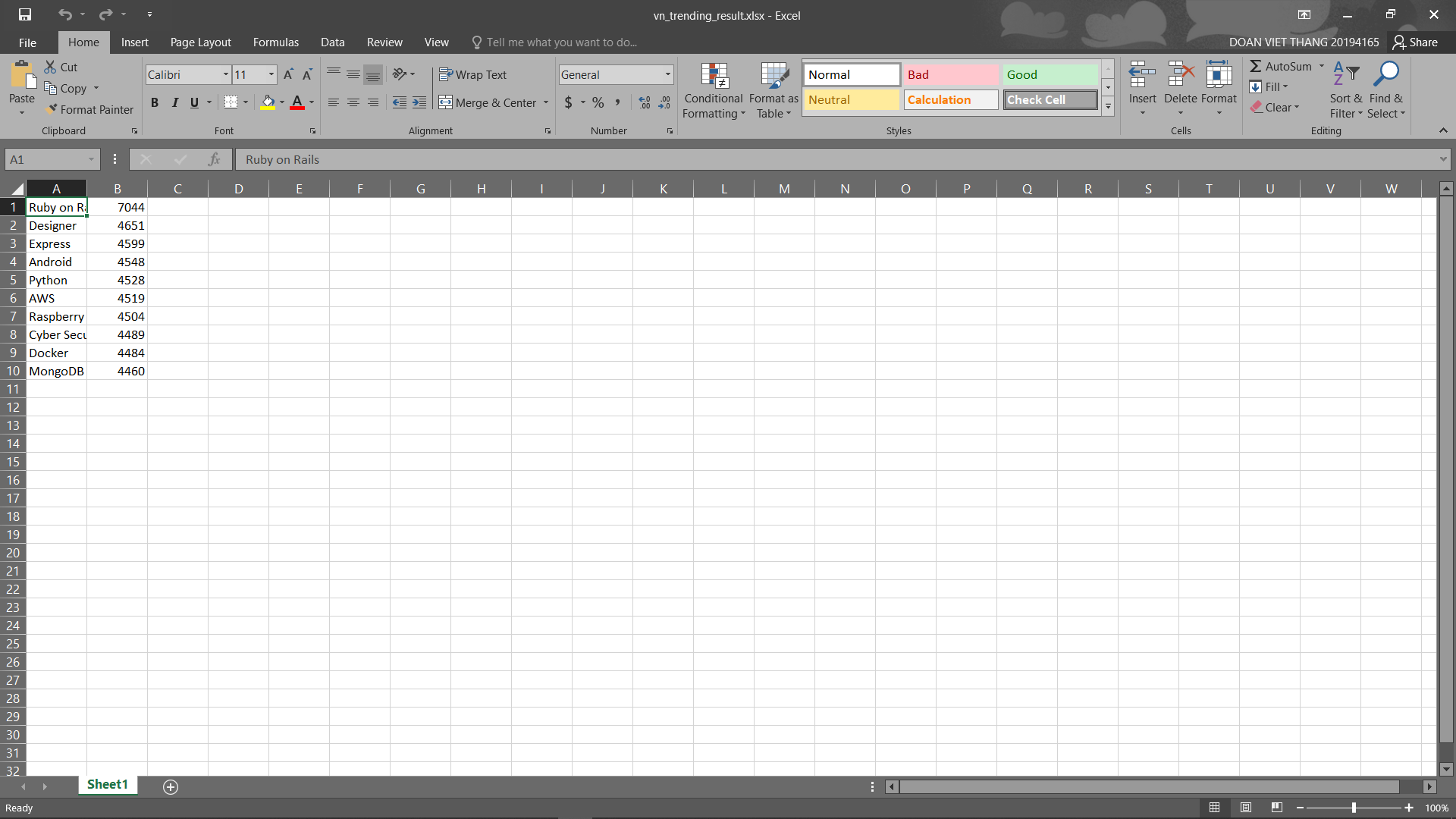
1. ***Các file excel được tạo ra từ python trong quá trình thực hiện***
2. global\_trending.xlsx



1. vn\_trending.xlsx



1. vn\_trending\_result.xlsx



1. vn\_top\_ten.xlsx

