**CIS 612 Big Data & Para Database System**

Name: - Kaushal Prajapati

CSUID: - 2789553

Submitted to: - Dr. Sunnie S. Chung

Lab: - Lab-3\_3 Semi-Structured Data Processing with NoSQL Database Server Language Used: - Python

Interpreter: - PyCharm

Database: - Microsoft SQL , MongoDB

Description about project: -

1. Collected **349,765** tweets.
2. All data is regarding hashtag #Donald Trump, #Presidential Election, #Biden
3. Total Size of data 2.0GB.
4. Average size of each tweet: - 6.1kb
5. Used Python for parsing data.
6. Used packages: - Tweepy, pymongo, Json.
7. Used Twitter API.
8. Further Information is at last in conclusion.
9. Data fields parsed per tweet :- \_id,created\_at,id,id\_str,text,display\_text\_range,source,truncated,in\_reply\_to\_status\_id,in\_reply\_to\_status\_id\_str,in\_reply\_to\_user\_id,in\_reply\_to\_user\_id\_str,in\_reply\_to\_screen\_name,name,screen\_name,location,url,description,translator\_type,protected,verified,followers\_count,friends\_count,listed\_count,favourites\_count,statuses\_count,created\_at,time\_zone,geo\_enabled,lang,contributors\_enabled,is\_translator,profile\_background\_color,profile\_background\_image\_url,profile\_background\_image\_url\_https,profile\_background\_tile,profile\_link\_color,profile\_sidebar\_border\_color,profile\_sidebar\_fill\_color,profile\_text\_color,profile\_use\_background\_image,profile\_image\_url\_https,profile\_banner\_url,default\_profile,default\_profile\_image,following,follow\_request\_sent,notifications,geo,coordinates,place,contributors,is\_quote\_status,extended\_tweet,full\_text,display\_text\_range,entities,quote\_count,reply\_count,retweet\_count,favorite\_count,entities,hashtags,url,expanded\_url,display\_url,indices,user\_mentions,screen\_name,name,symbols,favorited,retweeted,filter\_level,lang,timestamp\_ms.

Code [Python][Interpreter – PyCharm] :-

from tweepy import Stream  
from tweepy import OAuthHandler #For handling authentication  
from tweepy.streaming import StreamListener #For streaming tweets  
import json  
from pymongo import MongoClient  
import pymongo

**# Credentials**

access\_token = '1240760950967480320-SKbpSG0JIJjEzEuvLYVyP6h3XxWJvr'  
access\_token\_secret = 'cvqKzSjXRwITluPal2Vem6z5BXu51pG8otf0KB7w3E6v1'  
consumer\_token = 'gM8jEQL88qaTJQC8mJzlknASy'  
consumer\_token\_secret = '2zv6zcJ2hPFUDZboFikmOifnOzueLvuMmZvBio6yM9emzVN0dd'

**# Connecting to MongoDB database**

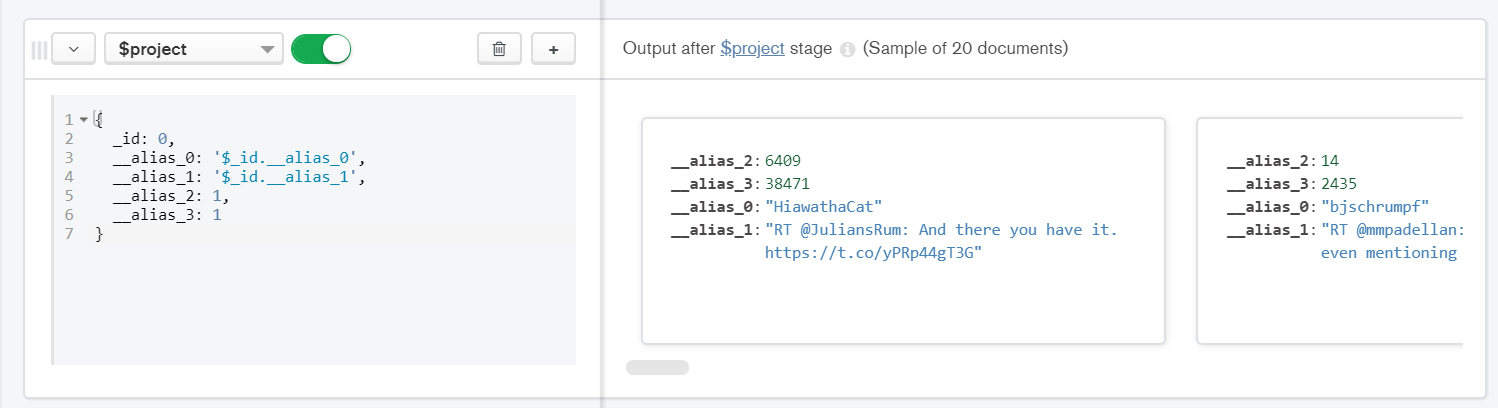
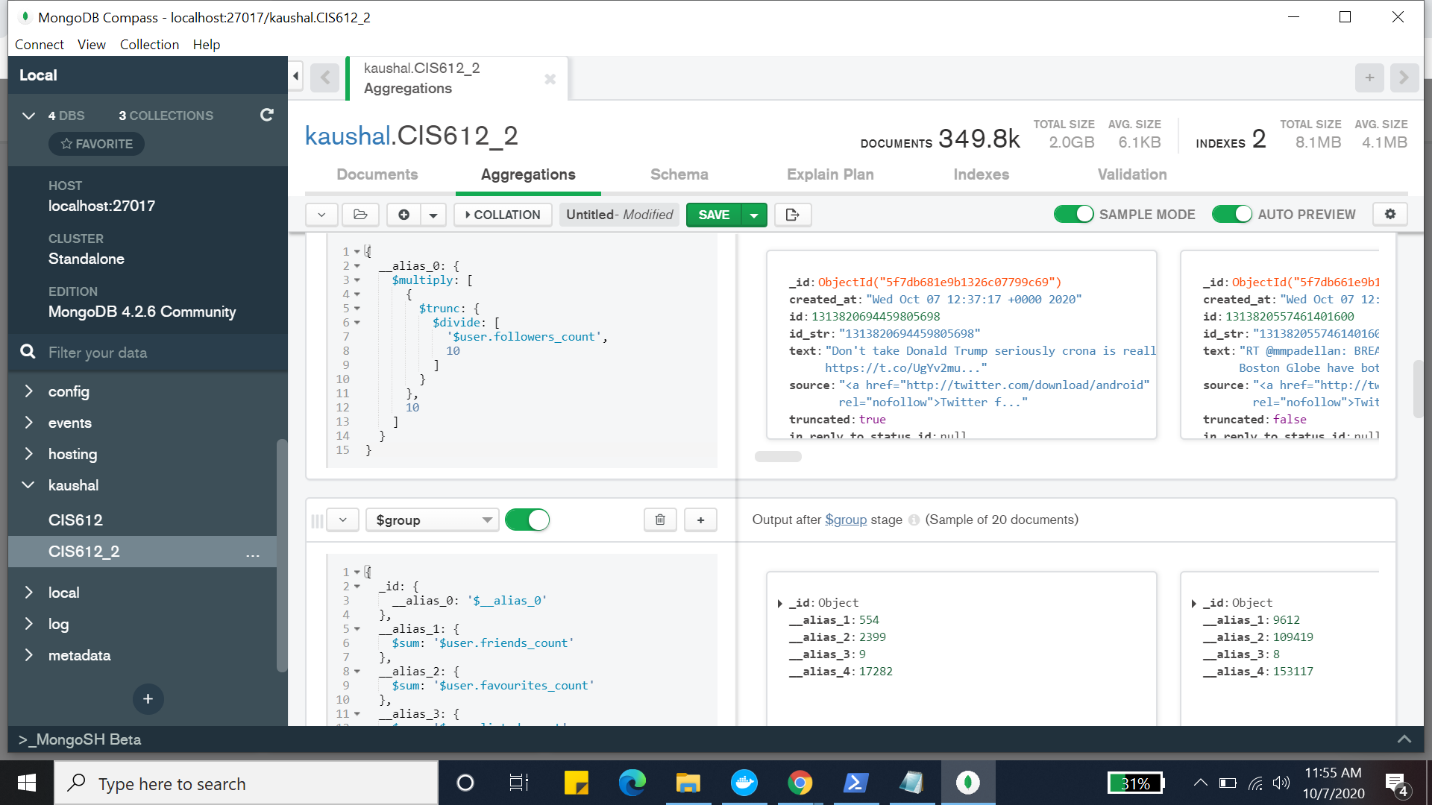
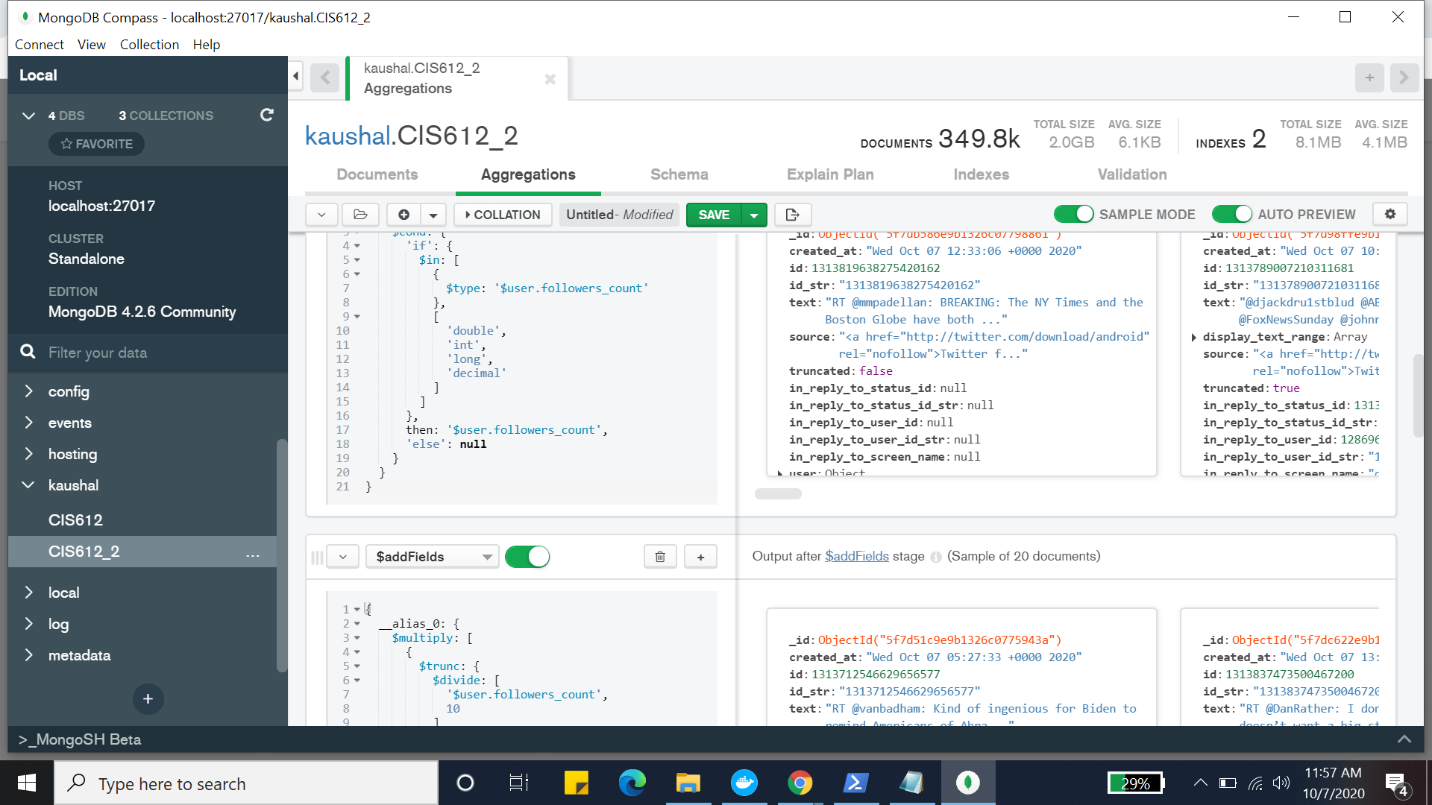
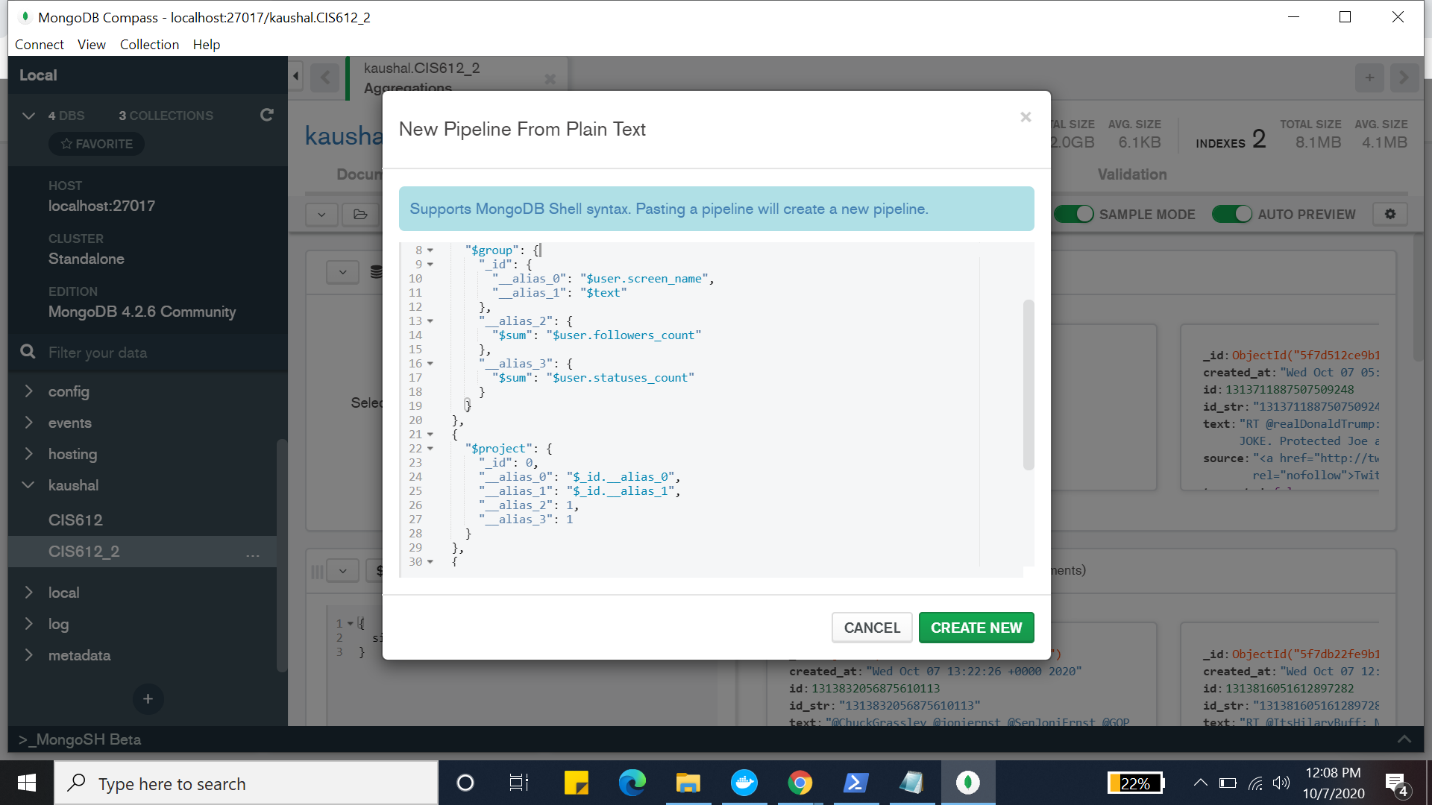
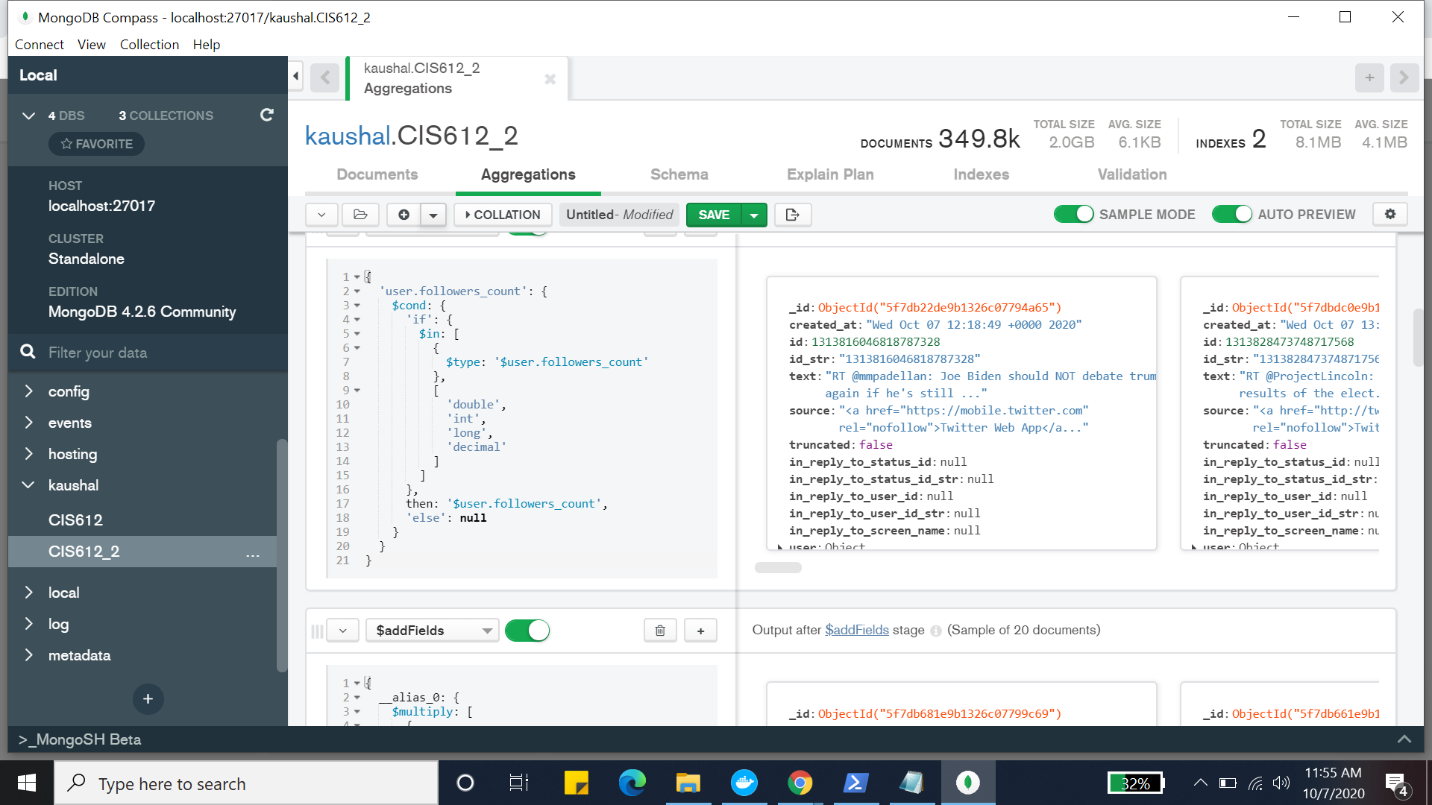
client = MongoClient()  
db = client.kaushal  
tweet\_collection = db.CIS612\_2  
tweet\_collection.create\_index([("id", pymongo.ASCENDING)], unique=True)

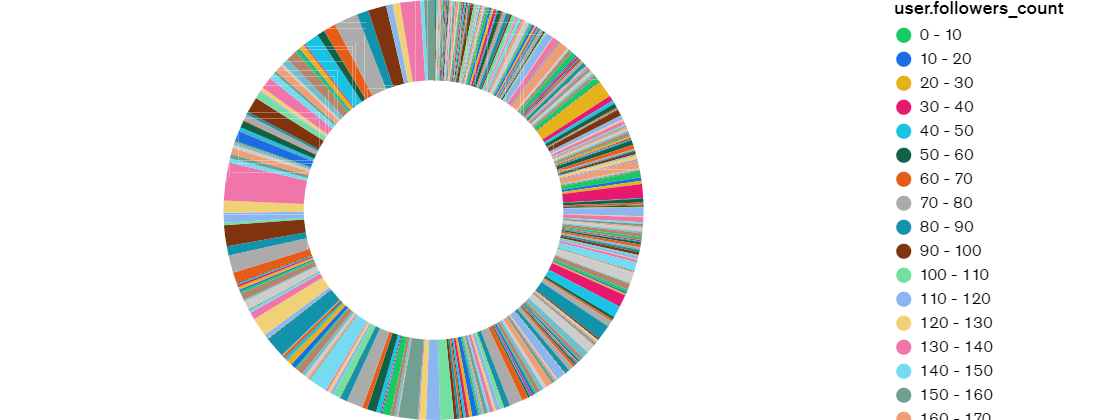
**# Tweets come in JSON format and then importing to MongoDB**class Listen(StreamListener):  
 def on\_data(self, data):  
 tweet = json.loads(data)  
 tweet\_collection.insert\_one(tweet)  
 return True

**# Main Function**if \_\_name\_\_ == "\_\_main\_\_":  
 listener = Listen() # listener object of Listener class  
 authentication = OAuthHandler(consumer\_token, consumer\_token\_secret)  
 authentication.set\_access\_token(access\_token, access\_token\_secret)  
 stream = Stream(authentication, listener) # stream object with authentication

stream.filter(track=["Presidential Election", "Donald Trump", "Joe Biden"]) # filtering the tweets with

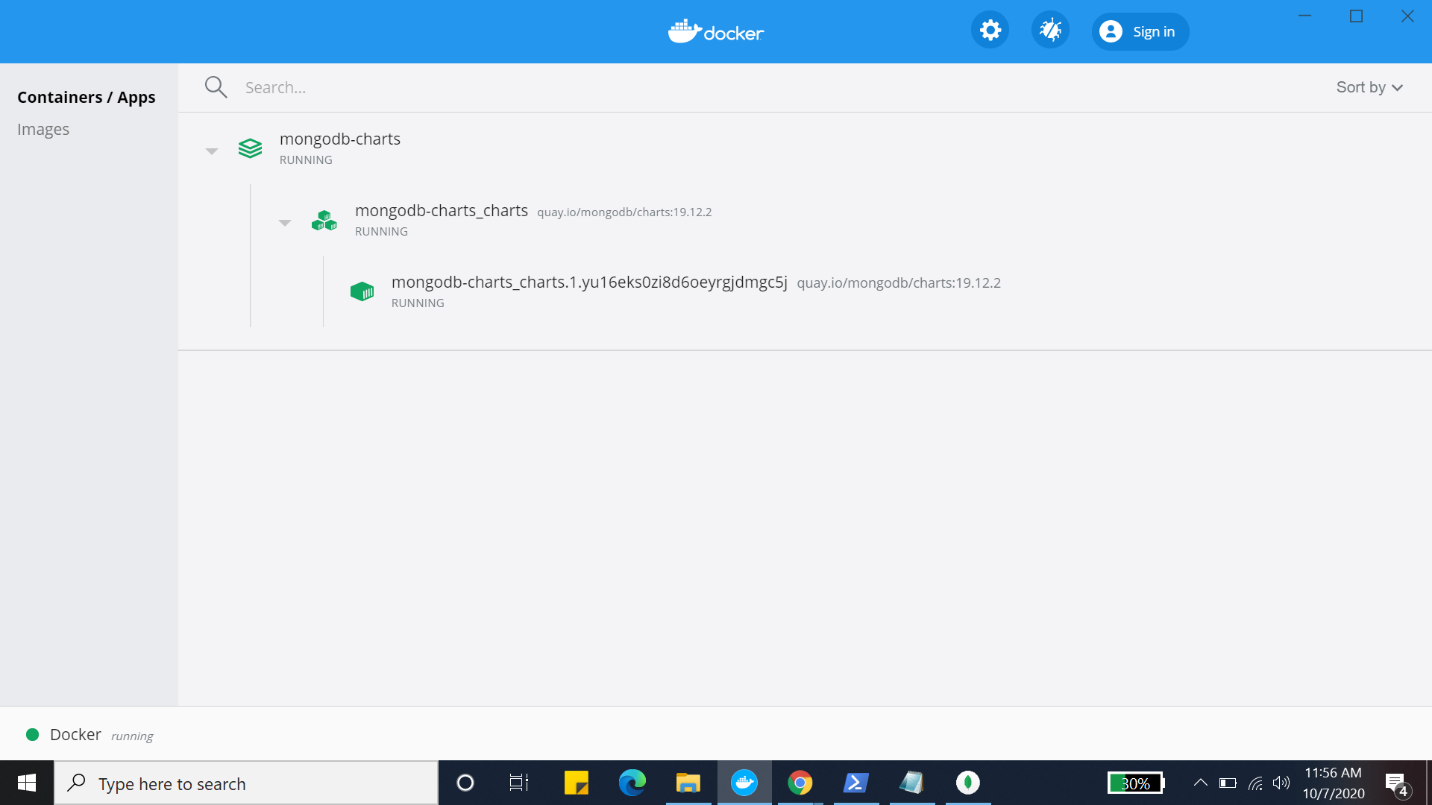
MongoDB Queries: -



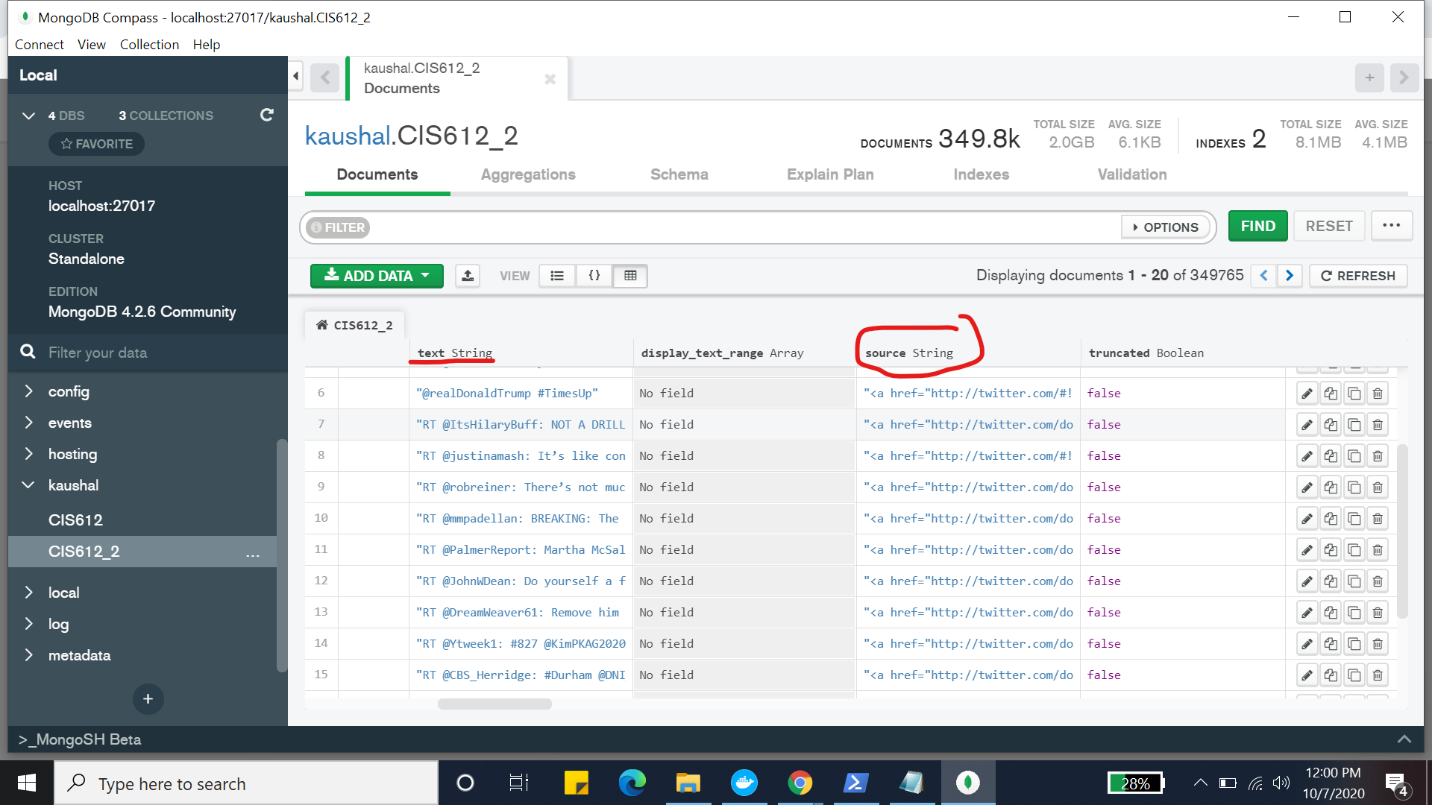
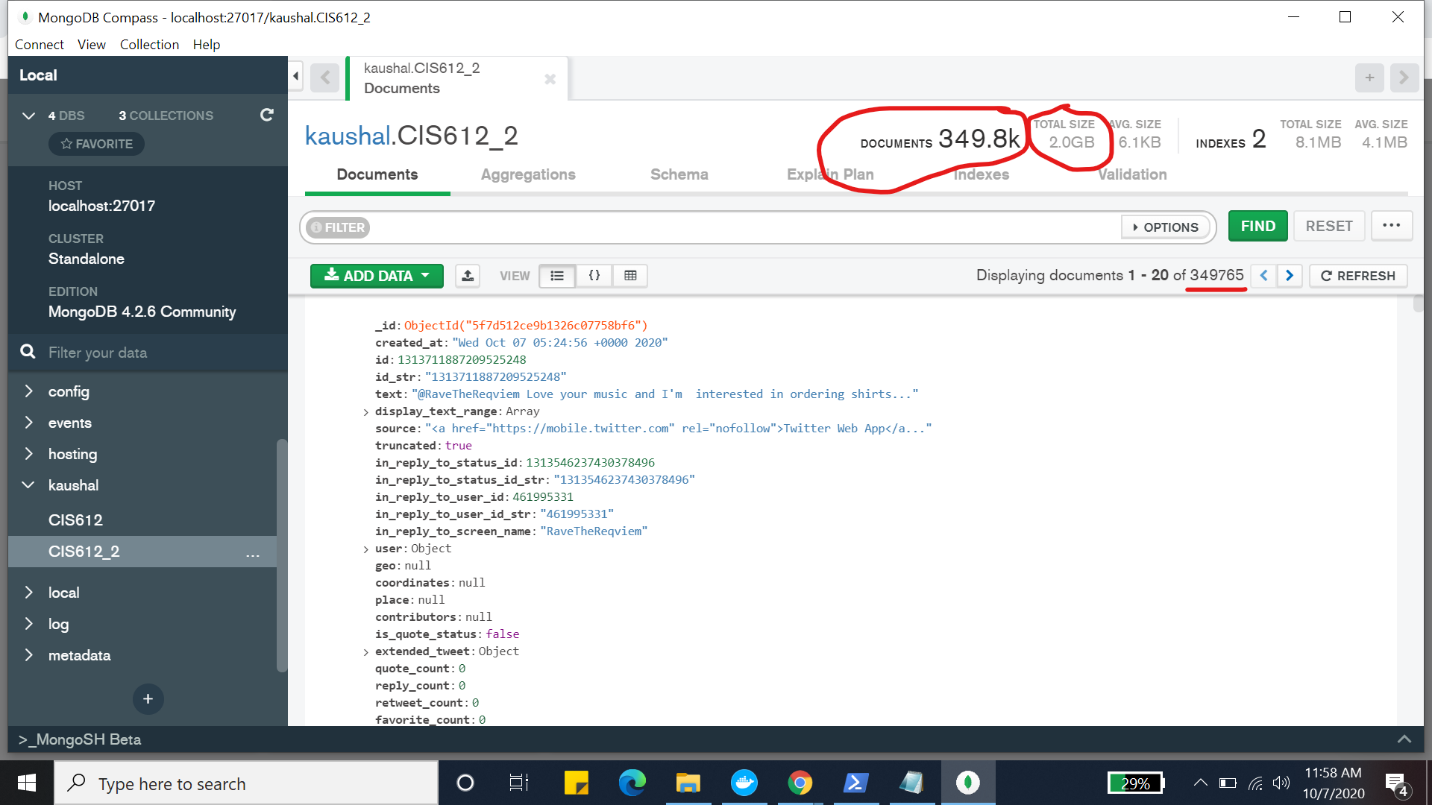
MongoDB Charts :- 

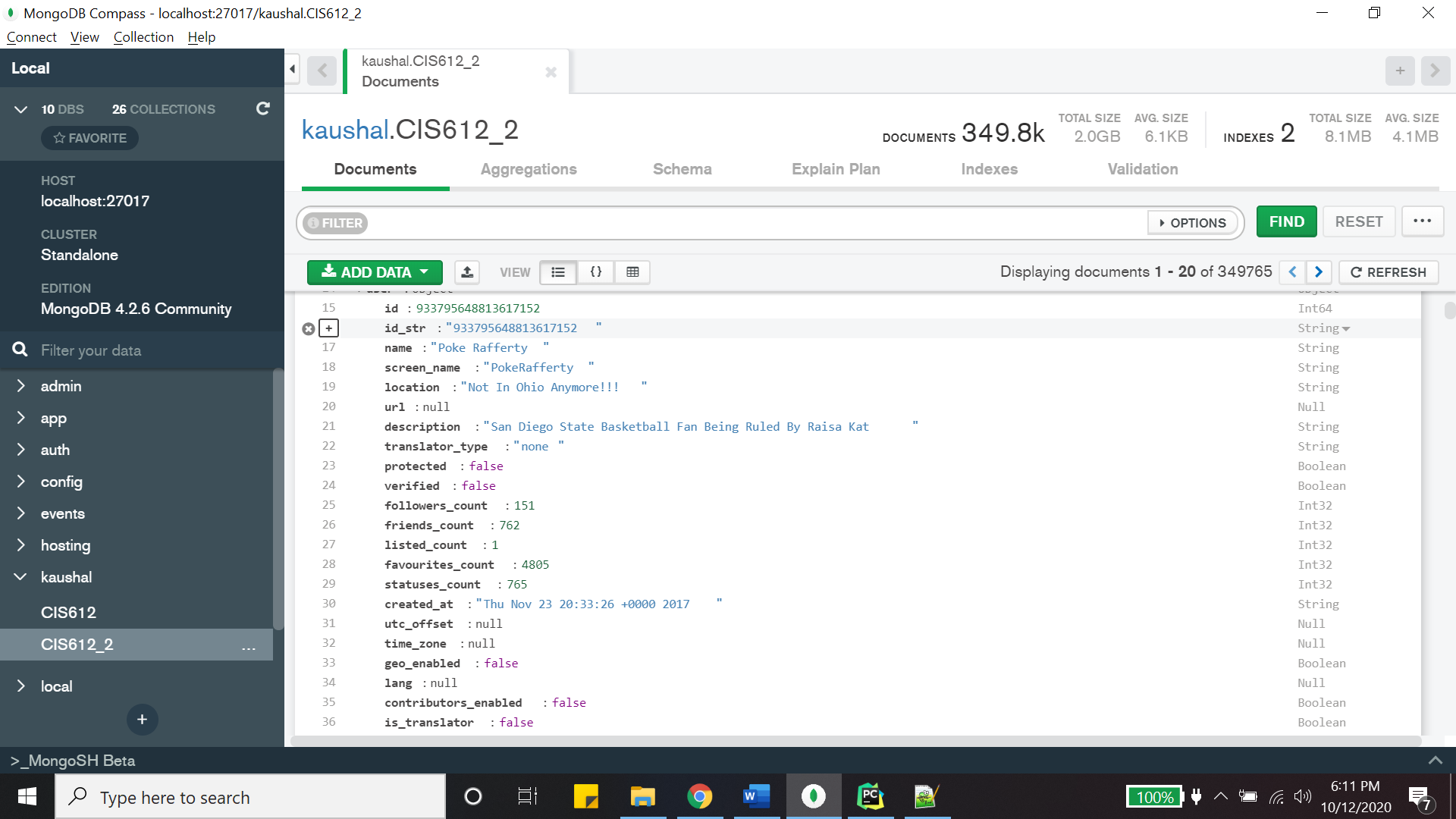


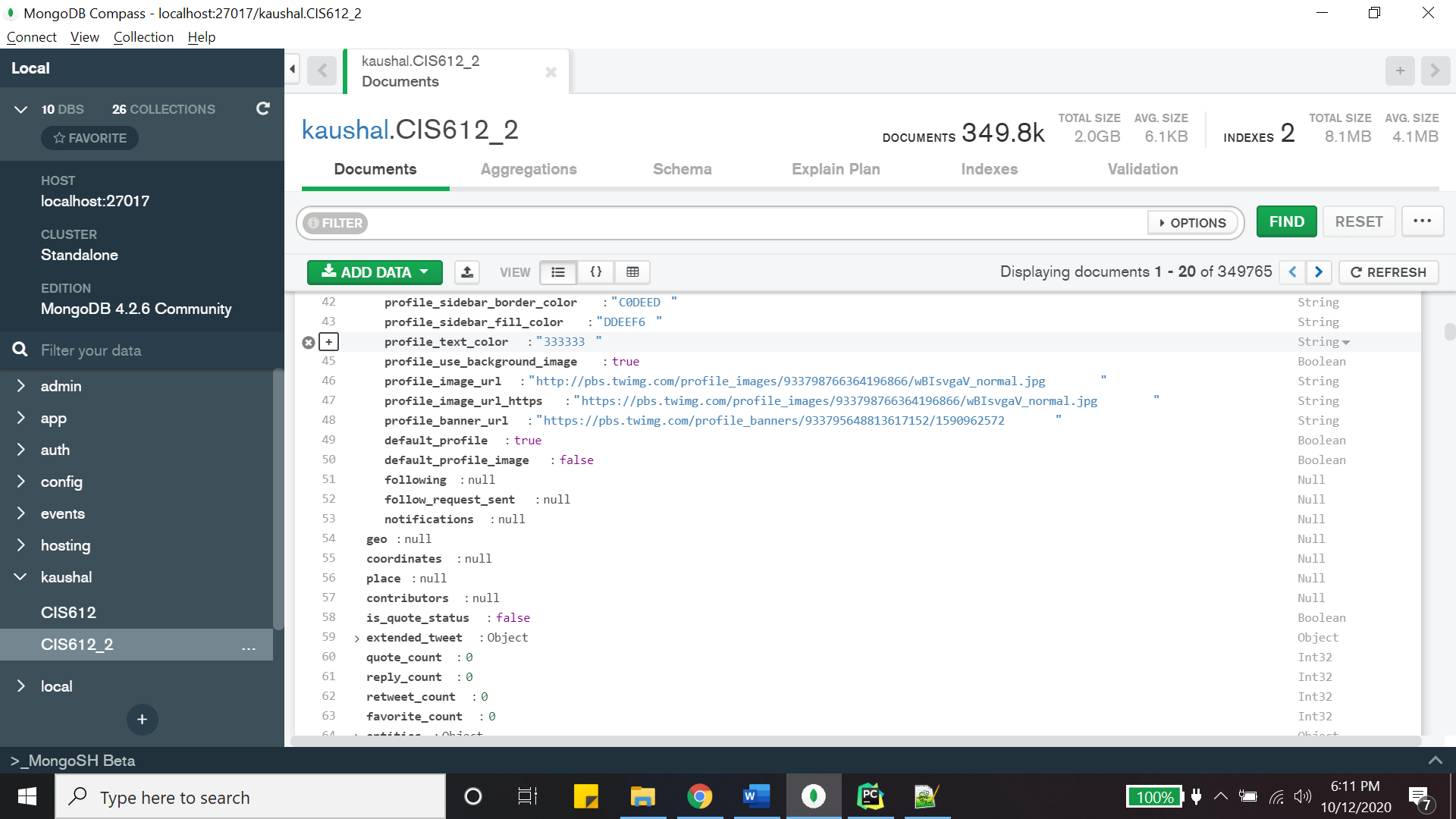
I Used Docker for MongoDB Charts: -



Some Screenshots of Database: -

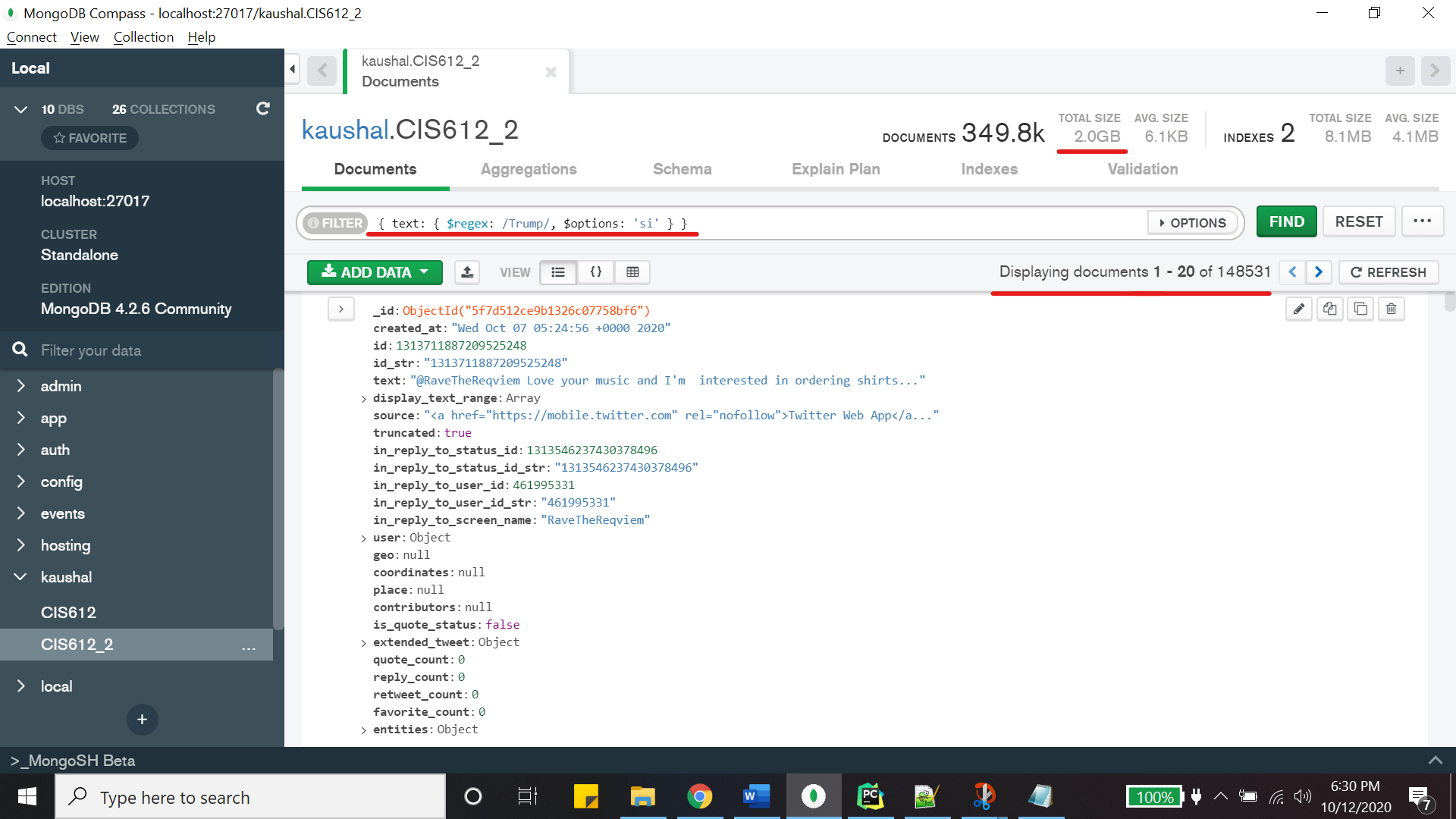


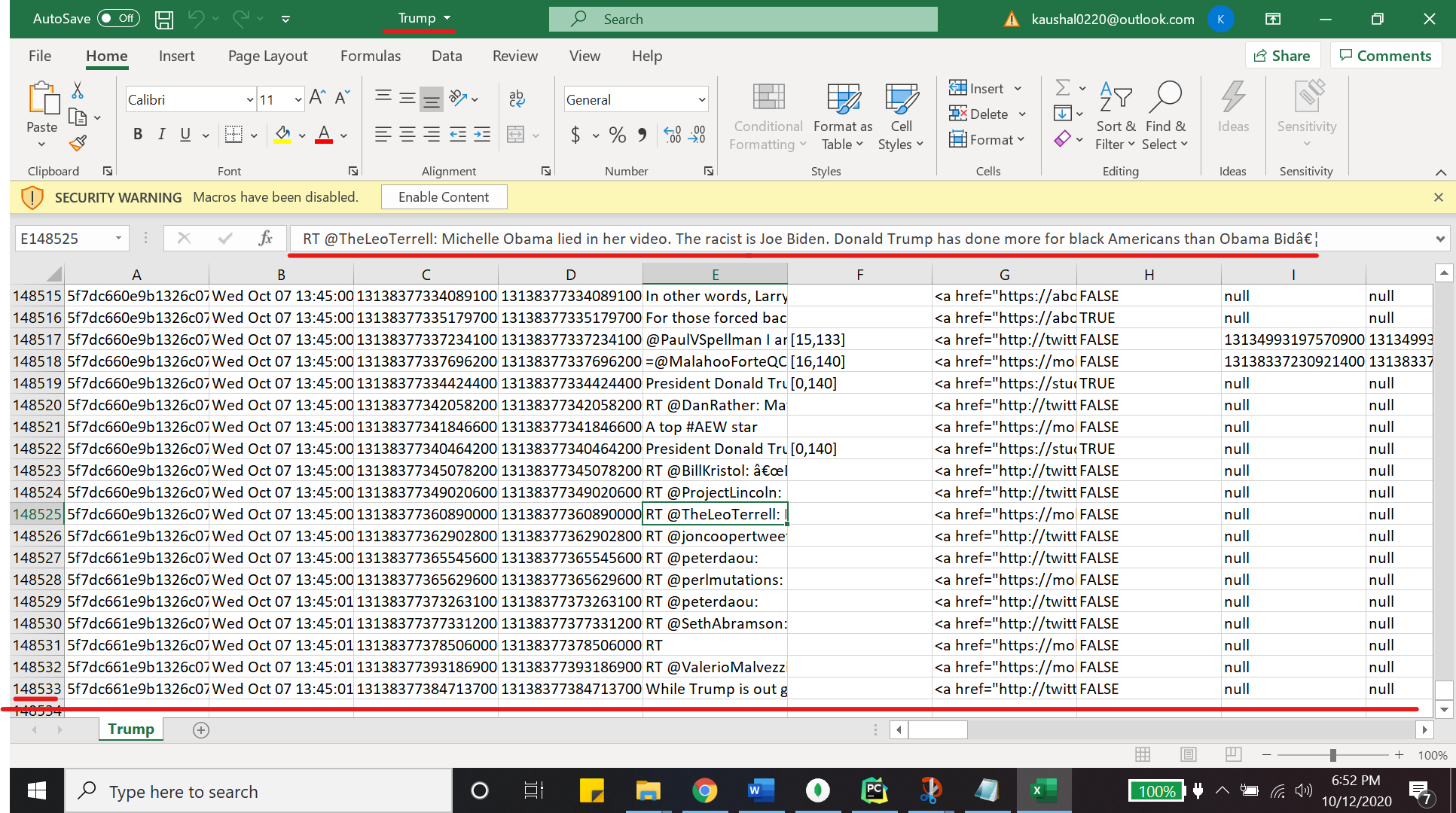


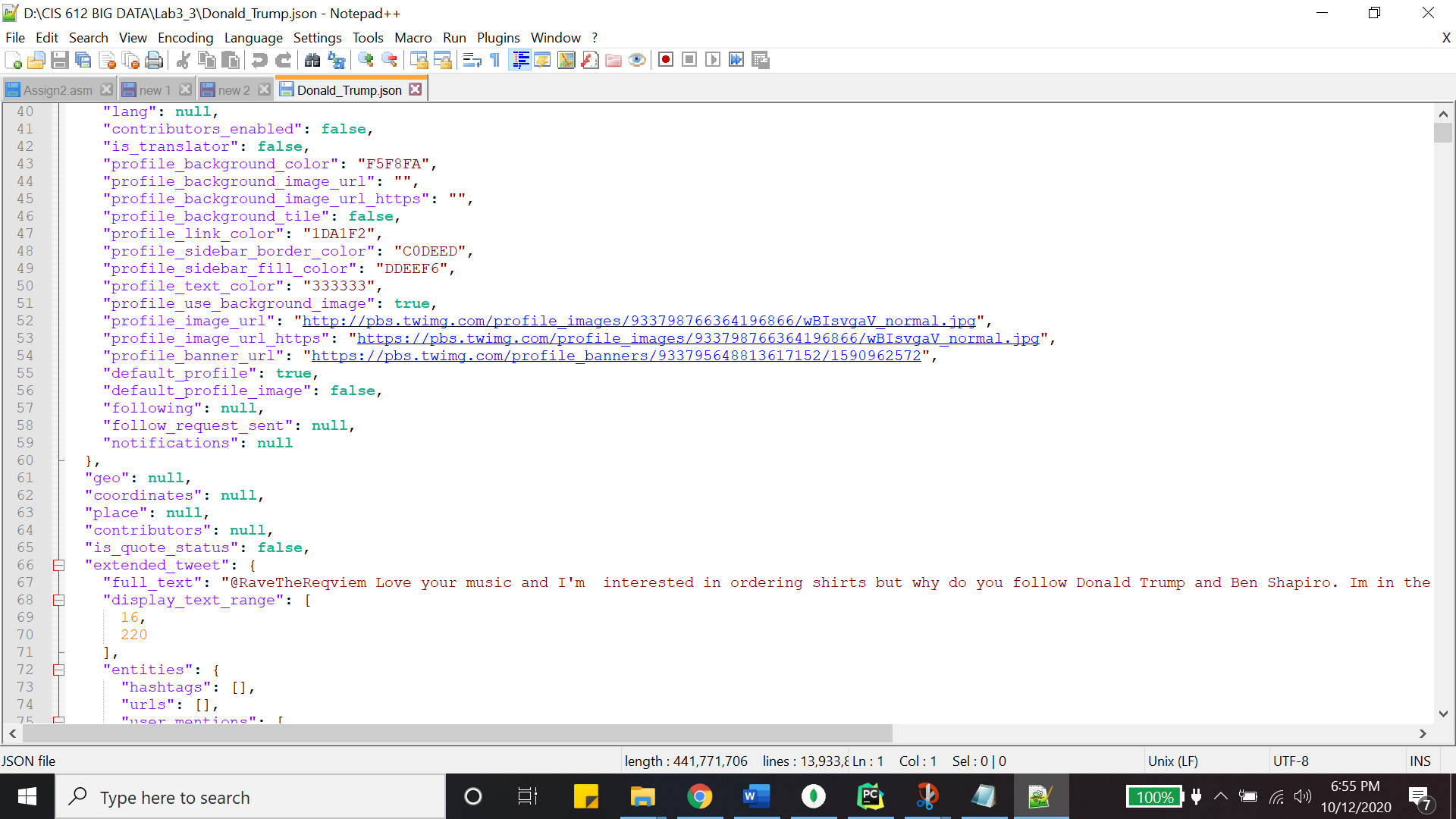


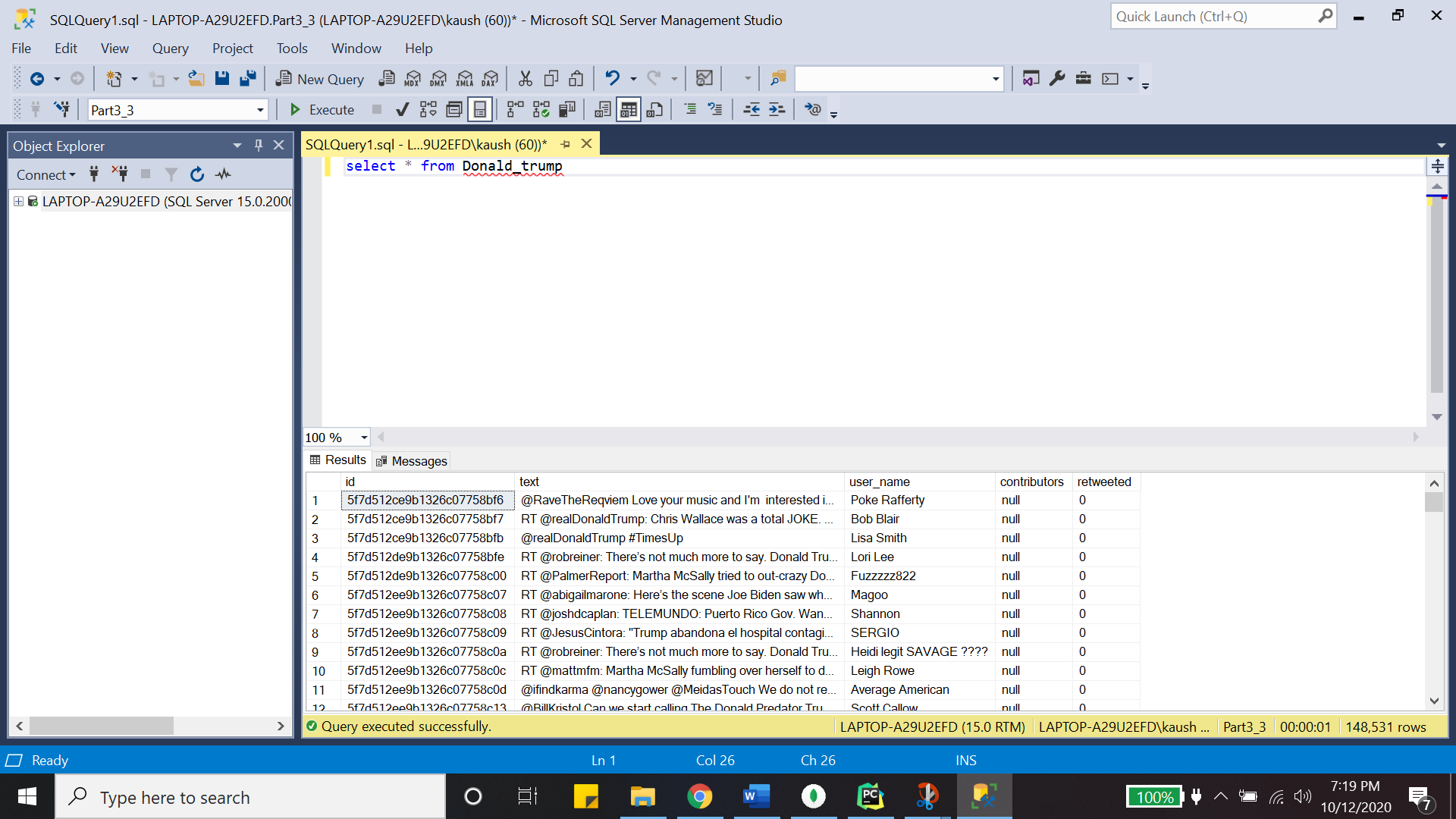
Dividing Tweets including text: - Trump , Biden.

***This Tweets Includes: - Donald Trump***

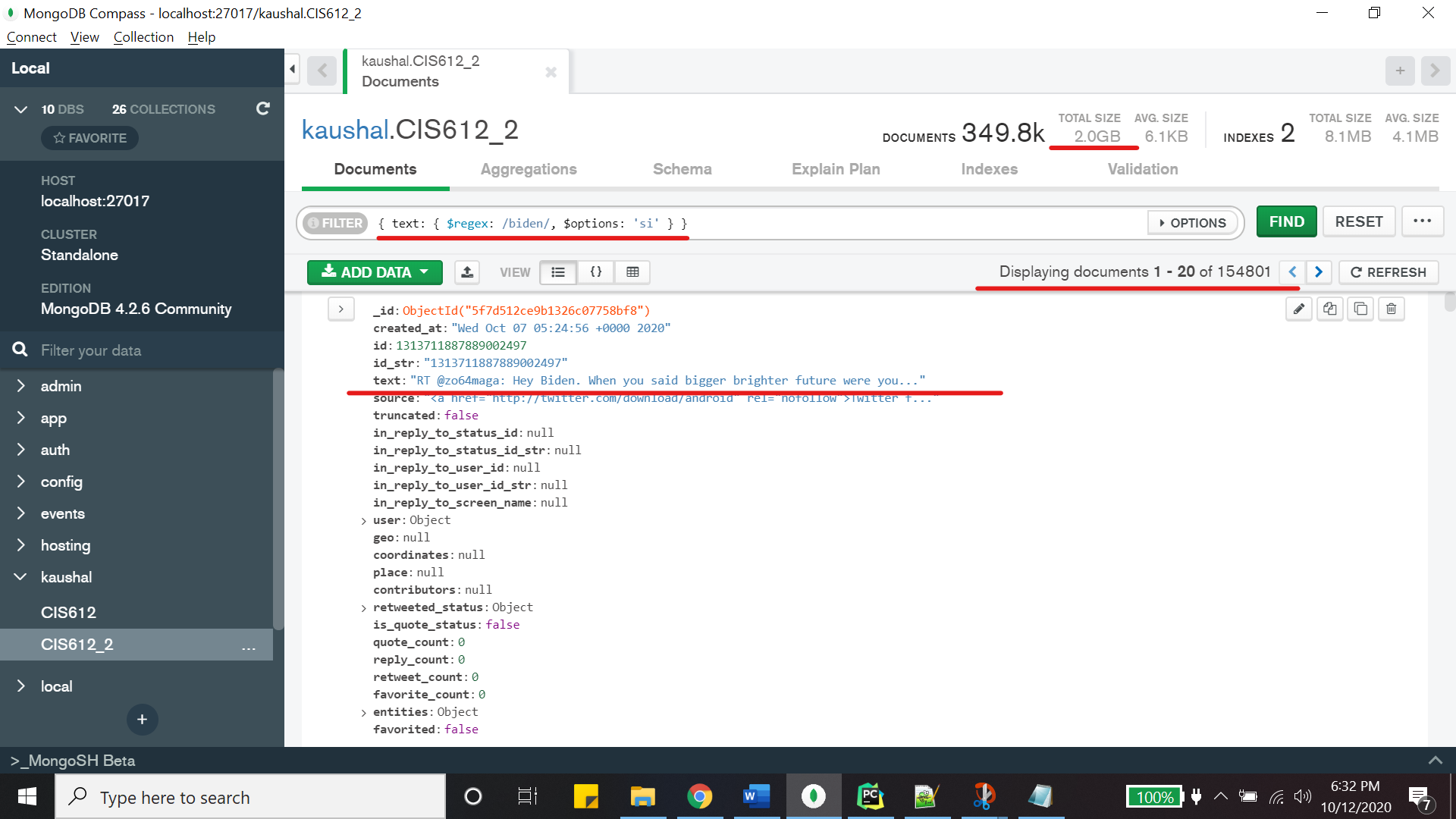


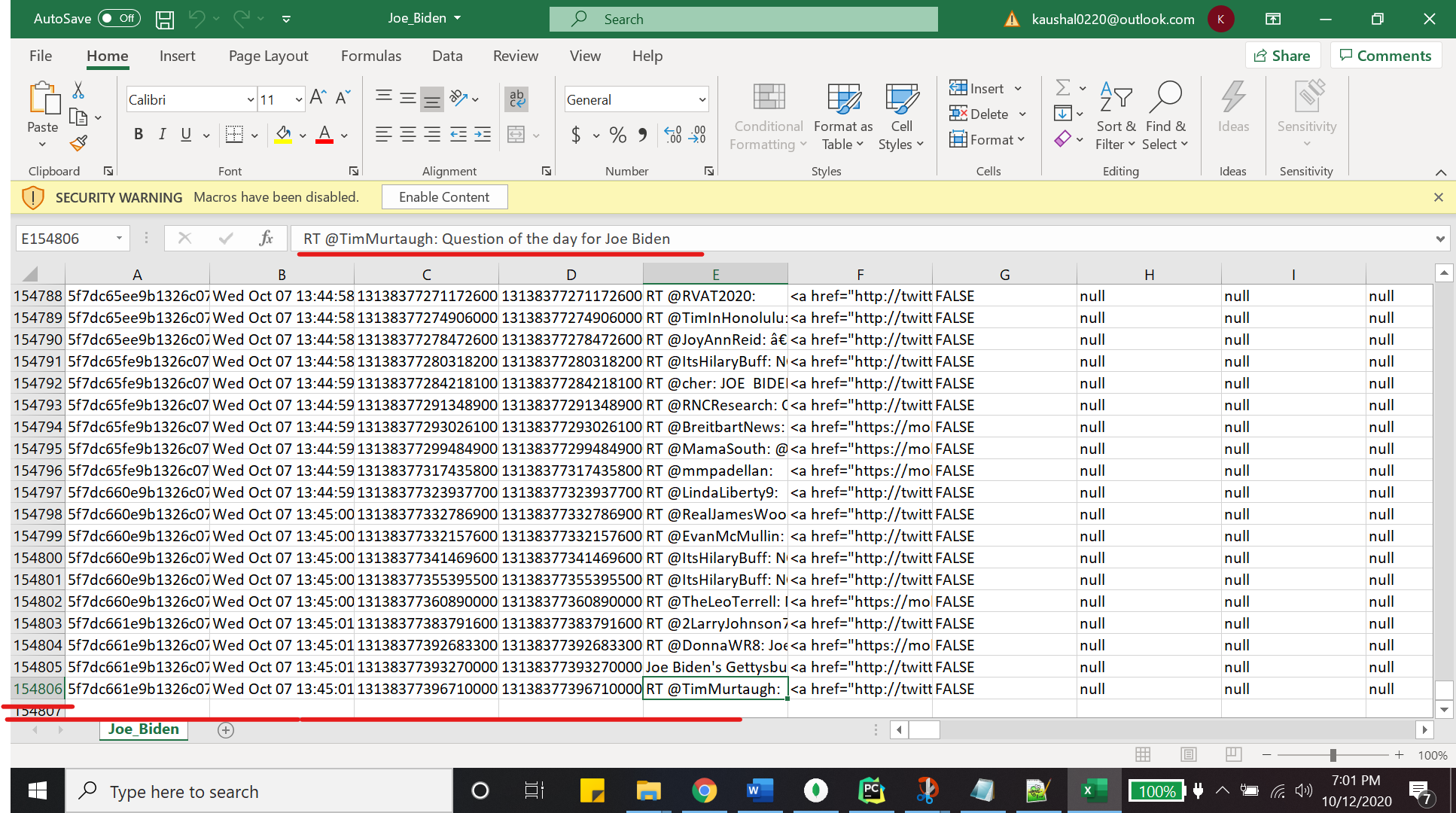


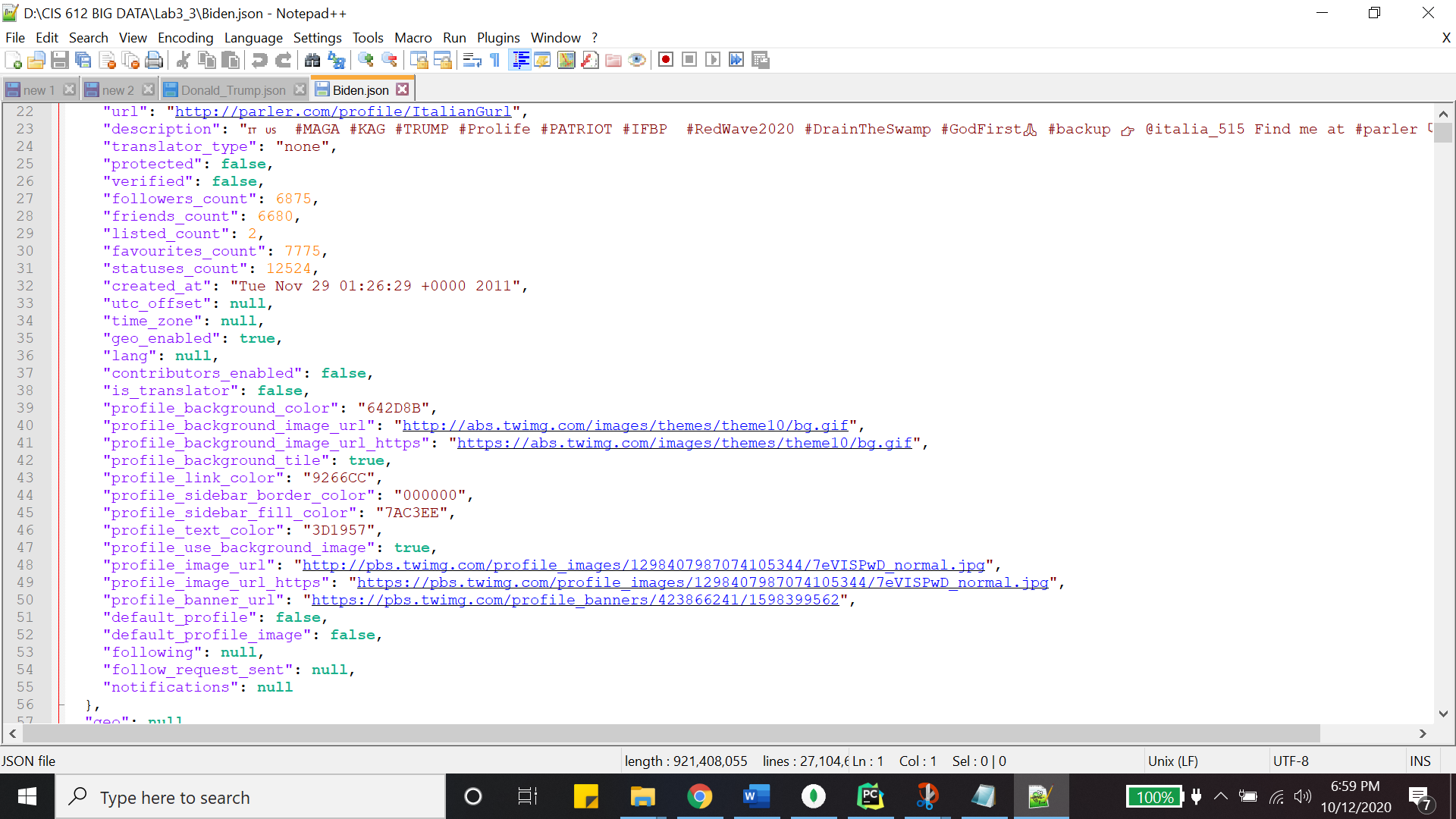


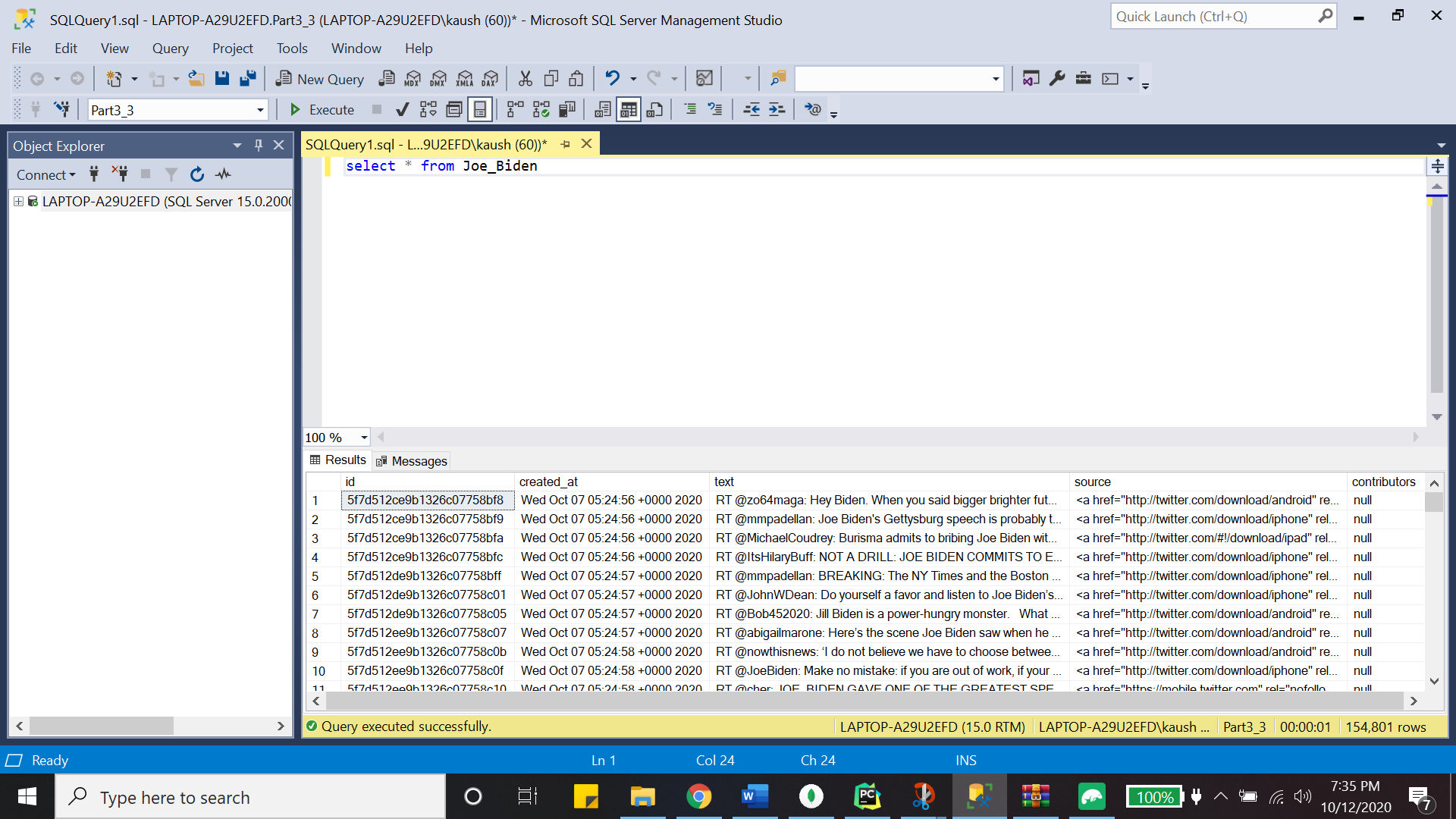


***This Tweets Include: - Joe Biden***









Mongo Queries: -

use kaushal;

db.getCollection("CIS612\_2").find(

{

"user.followers\_count" : {

"$gte" : NumberInt(87)

},

"user.friends\_count" : {

"$gt" : NumberInt(117)

},

"user.listed\_count" : NumberInt(1)

}

);

(2)

use kaushal;

db.getCollection("CIS612\_2").find(

{

"user.profile\_background\_color" : "F5F8FA",

"user.profile\_text\_color" : "333333"

}

);

Conclusion: -

* Made the application as a REST API application in Python to make the process automatic for big data processing.
* Retrieve all the messages related to Donald Trump in MongoDB Queries and saved them.
* Transform each twit in JSON structure to CSV/TSV to Create a Relational table in a SQL Server. [ All Screenshots are included]
* Found the basic info like how many messages with $Match, Aggregation pipelining, and $Group [Screenshots included in this report above].
* Collected all the data in MongoDB.
* Used and Learned Docker for MongoDB Charts.
* Coding Language – Python, Interpreter – PyCharm.
* Project Completely done with all the requirements and Extra Credit Part.