



**DALHOUSIE
UNIVERSITY**

FACULTY OF
COMPUTER SCIENCE

**CSCI 5408 Data Management,
Warehousing, Analytics**

Assignment 3

November 12, 2020

Submitted by:

Samkit Shah[B00852292]

Table of Contents

Table of Figures	3
Cluster Setup Steps:	4
Tweet Extraction Steps:	5
Search data extraction:	5
Stream Data Extraction:	5
Output Screenshot for twitter data:	6
Cleaning Process:	9
Output Screenshot for Cleaning:	9
News Articles data extraction:	11
Output for news articles data extraction:	11
Map Reduce to perform count:	13
Output of MapReduce Program:	14
Data Visualization using Graph Database:	15
References:	18

Table of Figures

Figure 1 Compute Engine - Apache spark Cluster.....	4
Figure 2 Initial Status of RawDb Database.....	6
Figure 3 Search tweet execution on GCP.....	6
Figure 4 RawDb database	7
Figure 5 Stream tweet data execution.....	7
Figure 6 RawDb after added stream data.....	8
Figure 7 Initial ProcessedDb.....	9
Figure 8 Execution of cleaning process.....	10
Figure 9 ProcessedDb after cleaning	10
Figure 10 Initial state of ReuterDb.....	11
Figure 11 Reuter data extraction	12
Figure 12 ProcessedDb after data added.....	12
Figure 13 MapReduce Output-1	14
Figure 14 MapReduce Output-2	14
Figure 15 Graph of Nodes	17

GitLab Repository : <https://git.cs.dal.ca/ssshah/assignment-3>

Cluster Setup Steps:

1. Created an account on <https://cloud.google.com/>
2. Go to console of GCP by clicking on the console button.
3. Created a new project named "Assignment-3"
4. Created a new Compute Engine of Ubuntu E2- Medium Tier instance.
5. Installed Scala, Java and git to satisfy the requirements of spark.
6. Installed Apache Spark using command: `wget https://downloads.apache.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop2.7.tg`
7. After that installed apache spark and configured path.
8. Started the master node for apache spark using command: `start master.sh`
9. Started the apache slave for the above apache master using command: `start-slave.sh<url>`
10. Open spark dashboard in browser to verify the working cluster.
11. Installed pyspark, tweepy and mongodb for future use.

Figure 1 shows the compute engine with apache spark cluster.

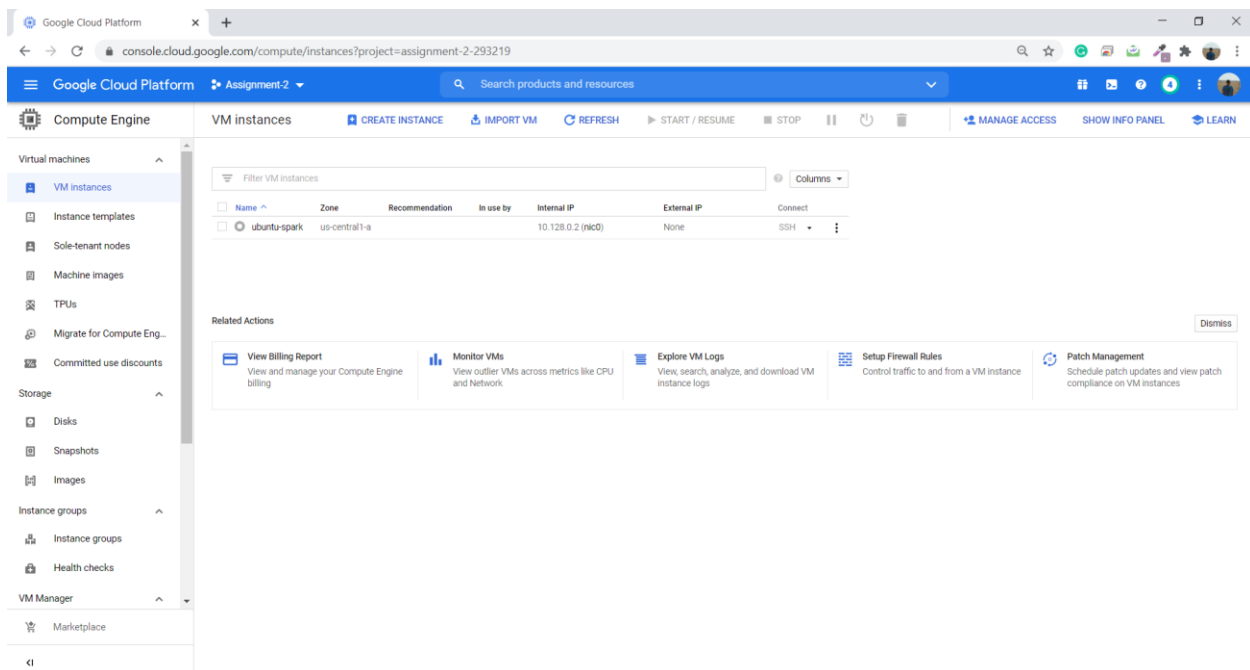


Figure 1 Compute Engine - Apache spark Cluster

Tweet Extraction Steps:

For data extraction from twitter, I have used tweepy library's search and stream API. Following is the approach used for data collection for search and stream of tweets.

1. Create a twitter development account.
2. Once Approved, generate consumer keys and access tokens to access twitter data using search and stream APIs.

After performing above two steps there are two part such as Search data extraction and Stream data Extraction. I have developed two python scripts to fetch the data from search API and stream API.

Along with that, tweets are stored in the MongoDB Database. I have created one database named RawDb using mongodb atlas. It has one collection named "tweets", where all tweets will be stored.

Search data extraction:

1. Authenticated identity of user using consumer API key and secret key.
2. Connect the MongoDB database using pymongo library.
3. The search API will fetch tweets based on the keywords mentioned in assignment.
4. A search API has returned the list of tweet object based on query. The tweet is of the four types such as Normal, Extended, Retweet, and Quoted Tweet.
5. Extracted approximately 2000 tweets.
6. Stored the JSON tweet object into the form of dictionary.
7. The tweets has different attributes according to type of tweet. The list of attributes for the tweet is as follow:
Normal: Timestamp, tweet id, Content, username, user.screen_name, user.location.
Extended: Normal tweet attributes & truncated, full_text.
Retweet: Normal and extended tweet attribute, timestamp, id, text, user details.
Quoted Tweet: Normal tweet attributes, id, text, user details of the original tweet.
8. Once the tweet object is composed then it will be stored in the RawDb.
9. Store tweet object into RawDb.

Stream Data Extraction:

1. Twitter_stream is a python script to fetch the live tweets data using stream API.
2. TwitterStreamListener is a class which is responsible to listen live tweets.
3. Connect the MongoDB database using pymongo library.
4. The user is authenticated using consumer key and access tokens.
5. On_data method is responsible to listen tweets and provide the tweets data in data variable.
6. In the response, it will check the type of the tweet attributes and compose the tweet according to type.
7. Compose_streaming_tweet_row function is responsible to create a dictionary of attributes of tweet.
8. The tweet object will be stored in the MongoDB database named RawDb.
9. This process will continue till 1000 tweets.

Output Screenshot for twitter data:

1. Initial Status of RawDB database:

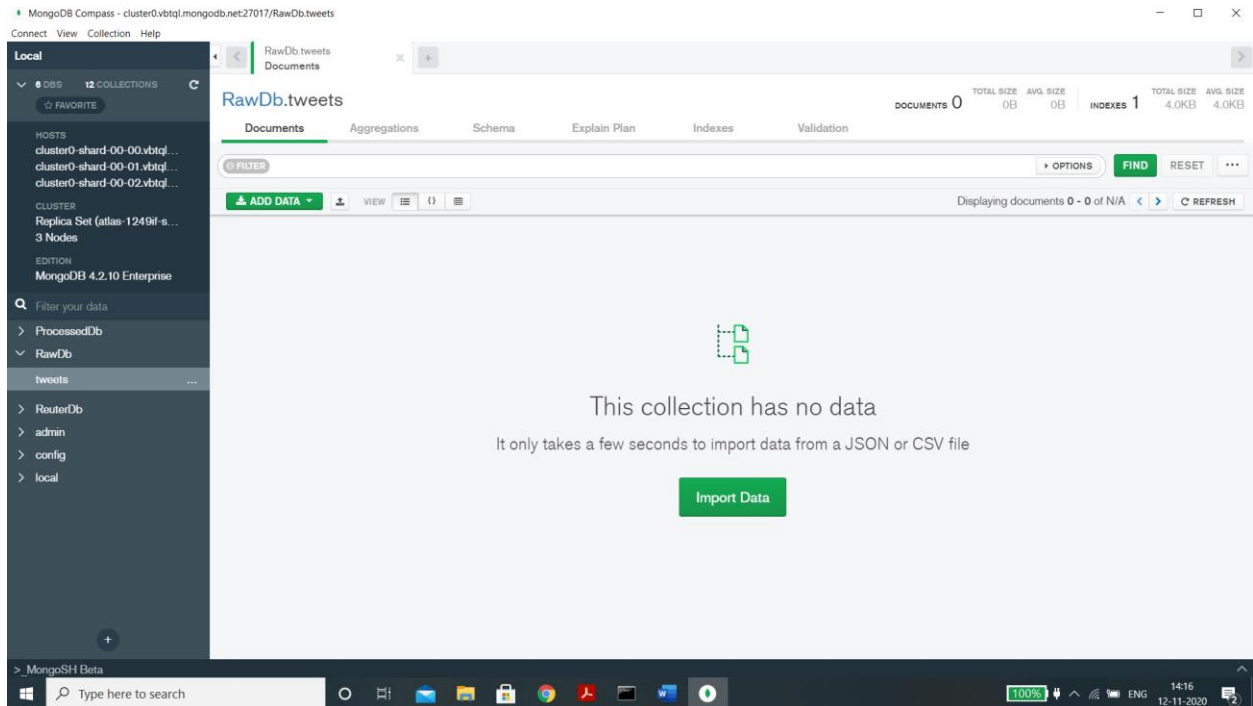


Figure 2 Initial Status of RawDb Database

2. Run python Script for search data extraction:

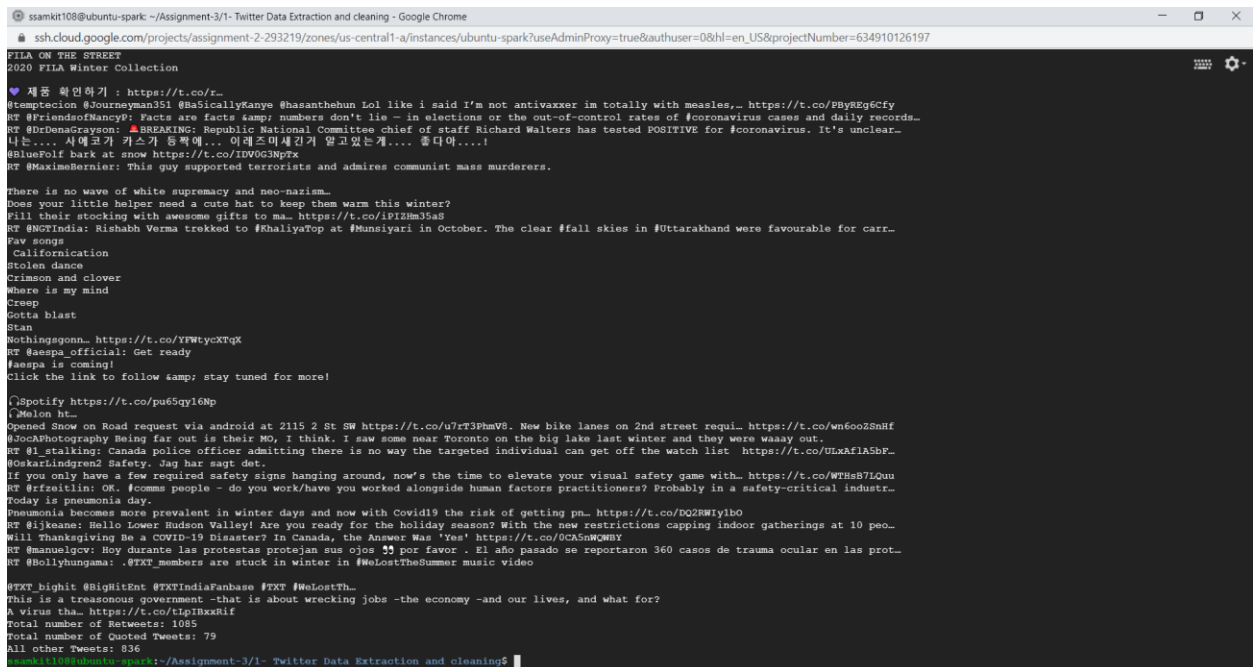


Figure 3 Search tweet execution on GCP

3. After Performing Search Data Extraction:

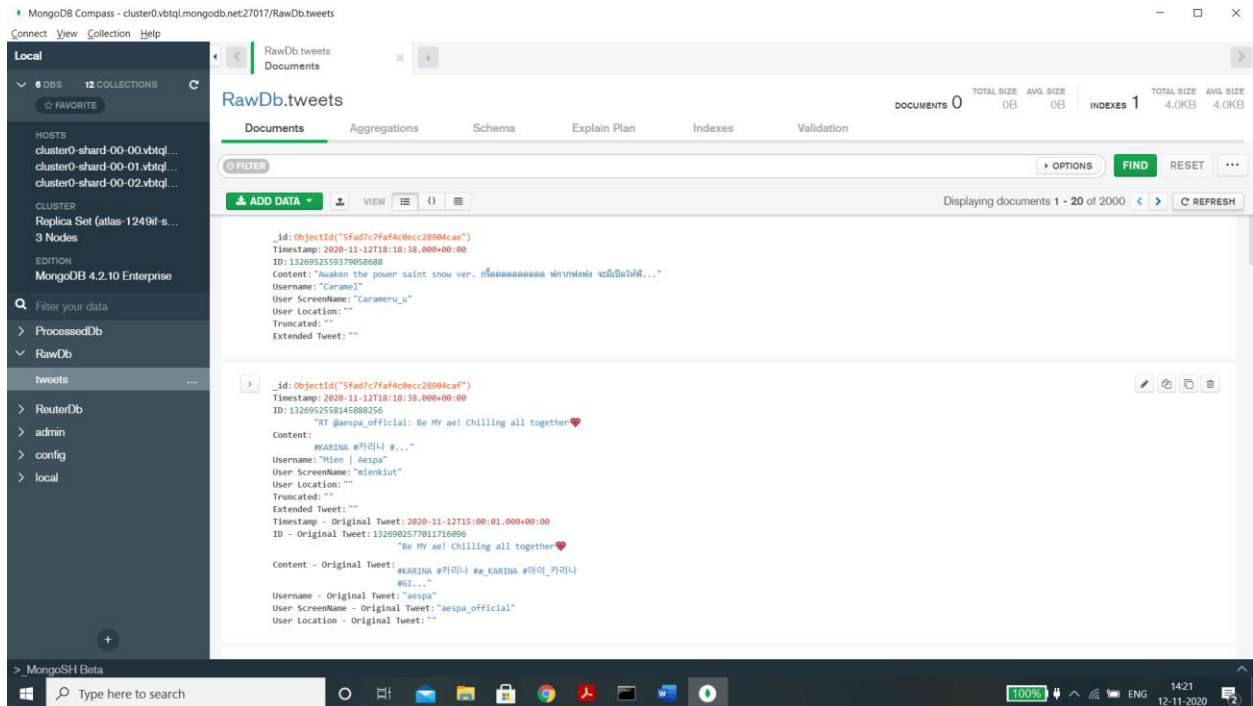


Figure 4 RawDb database

4. Run the python script to fetch the streaming data

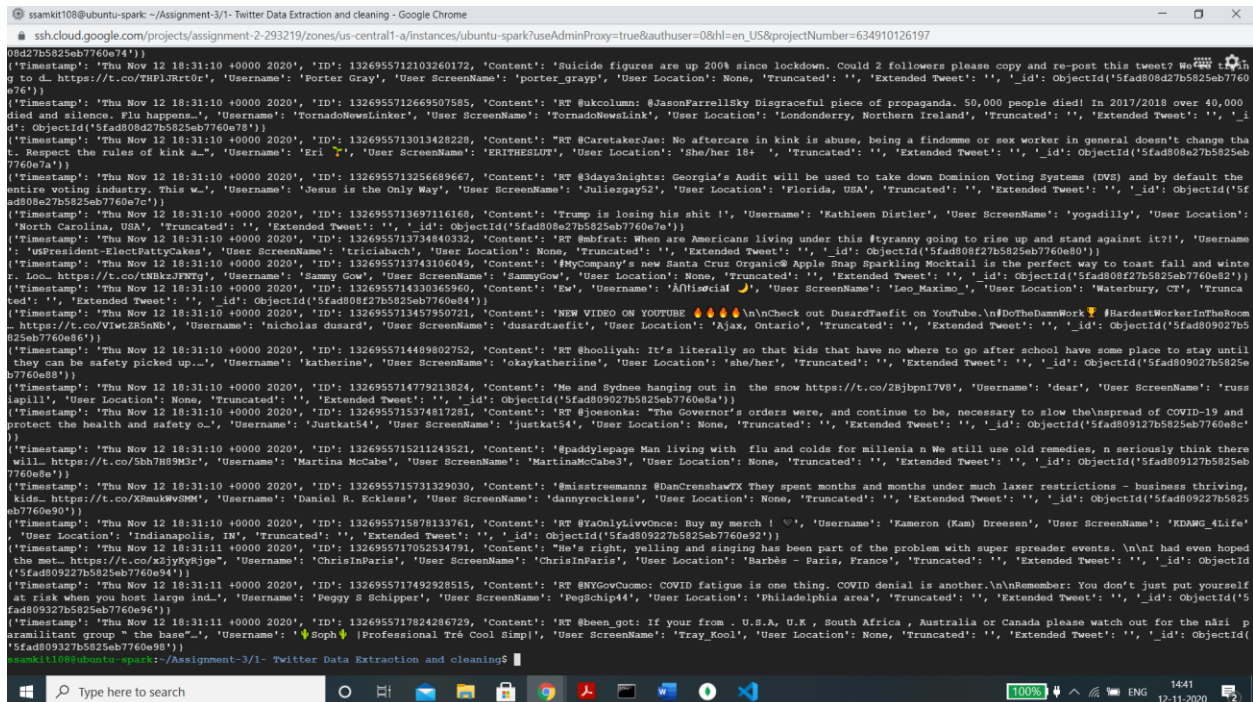


Figure 5 Stream tweet data execution

5. RawDb after adding stream data:

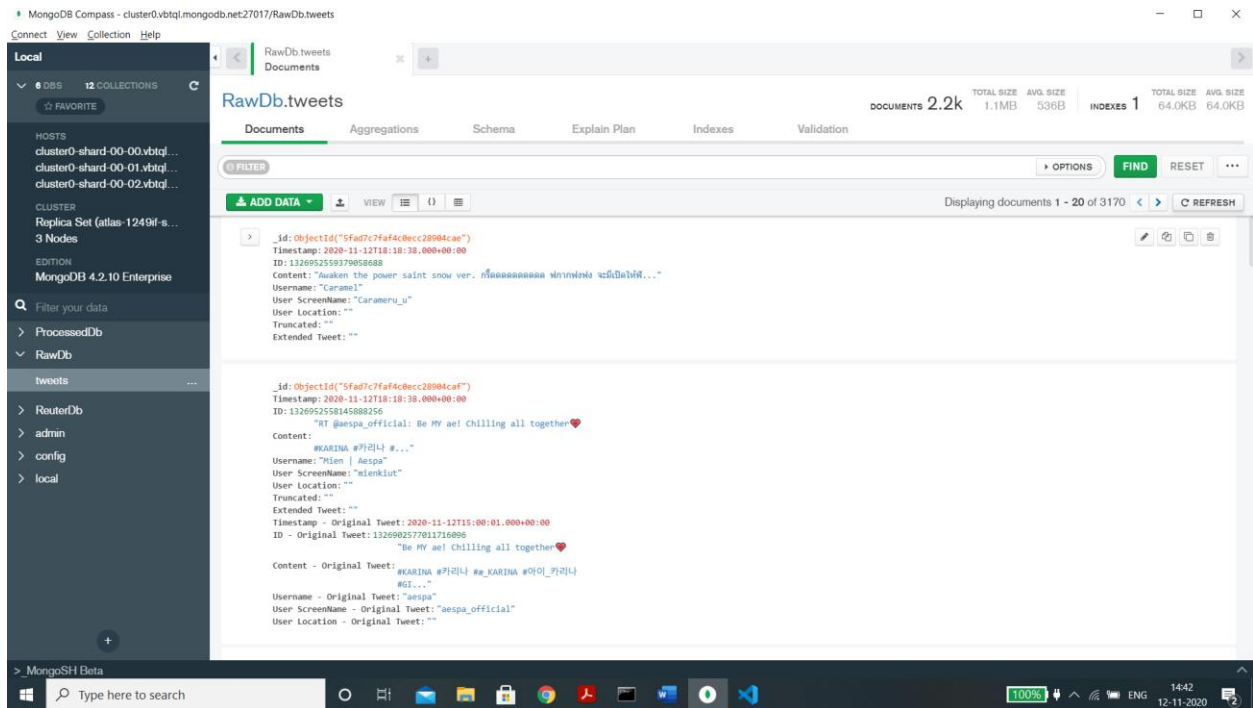


Figure 6 RawDb after added stream data

Cleaning Process:

All the tweets are stored in the RawDb. Now the task is to clean the tweets present in the RawDb and store it into the ProcessedDb. For cleaning purpose, I have written one python script named process.py. This script will fetch the data from the RawDb then clean() function will clean data and then it will store cleaned tweets to the ProcessedDb.

Data is cleaned by using Regular Expression. There is no use of any library. Mainly following parameters are cleaned as a part of cleaning process.

1. Multiple white spaces are replaced with single spaces.
2. Removed URLs in the tweets.
3. Removed Emoticons.
4. Removed new line character.
5. & symbol is corrected in tweets.
6. All the special characters (! , @ , ' , " , &) are removed.

Output Screenshot for Cleaning:

1. Initial status of ProcessedDb database

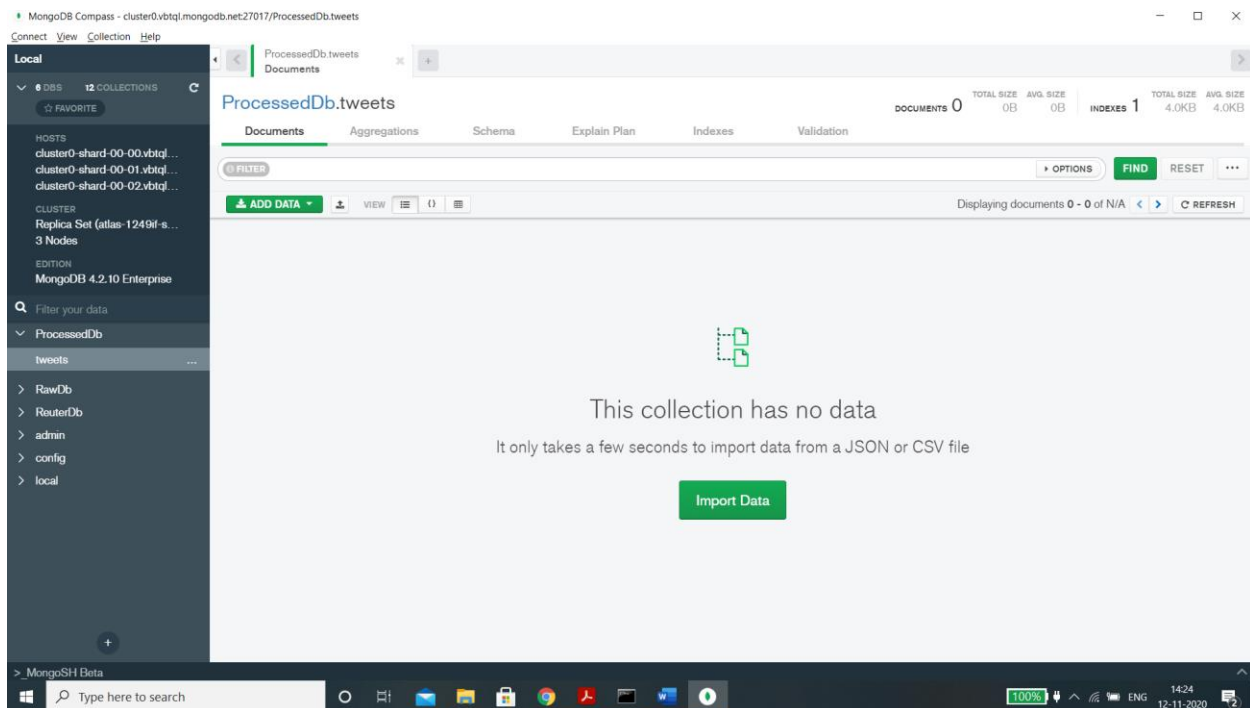
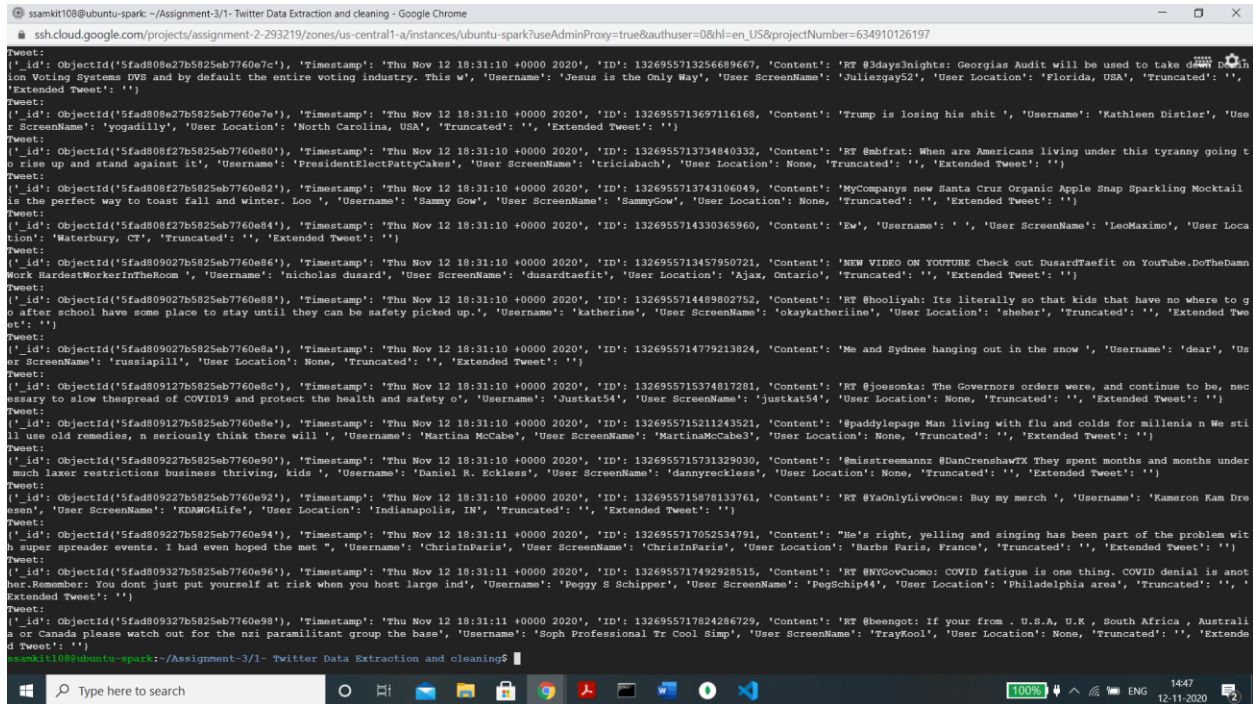


Figure 7 Initial ProcessedDb

2. Run the python script to clean data:



```
ssamkit108@ubuntu-spark: ~/Assignment-3/1- Twitter Data Extraction and cleaning - Google Chrome
ssh.cloud.google.com/projects/assignment-2-293219/zones/us-central1-a/instances/ubuntu-spark?useAdminProxy=true&authuser=0&hl=en_US&projectNumber=634910126197

Tweet:
{'_id': ObjectId('5fad808e27b5825eb7760e7c'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 132695571325689667, 'Content': 'RT @3days3nights: Georgias Audit will be used to take down the Voting Systems DWS and by default the entire voting industry. This w', 'Username': 'Jesus is the Only Way', 'User ScreenName': 'Juliezgay52', 'User Location': 'Florida, USA', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad808e27b5825eb7760e7e'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955713697116168, 'Content': 'Trump is losing his shit ', 'Username': 'Kathleen Distler', 'User ScreenName': 'yogadilly', 'User Location': 'North Carolina, USA', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad808f27b5825eb7760e80'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955713734840332, 'Content': 'RT @mbfrat: When are Americans living under this tyranny going to line up and stand against it', 'Username': 'PresidentElectPattyCakes', 'User ScreenName': 'triciabach', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad808f27b5825eb7760e82'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955713743106049, 'Content': 'MyCompanys new Santa Cruz Organic Apple Snap Sparkling Mocktail is the perfect way to toast fall and winter. Loo ', 'Username': 'Sammy Gow', 'User ScreenName': 'SammyGow', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad808f27b5825eb7760e84'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955714330365960, 'Content': 'Ew', 'Username': ' ', 'User ScreenName': 'LeoMaximo', 'User Location': 'Waterbury, CT', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809027b5825eb7760e86'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955713457950721, 'Content': 'NEW VIDEO ON YOUTUBE Check out DusardTaeFit on YouTube.DoTheDamn Work HardestWorkerInTheRoom ', 'Username': 'nicholas dusard', 'User ScreenName': 'dusardtaefit', 'User Location': 'Ajax, Ontario', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809027b5825eb7760e88'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955714489802752, 'Content': 'RT @hooliyah: Its literally so that kids that have no where to go after school have some place to stay until they can be safety picked up.', 'Username': 'katherine', 'User ScreenName': 'okaykatherine', 'User Location': 'sheher', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809027b5825eb7760e8a'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955714779213824, 'Content': 'Me and Sydnee hanging out in the snow ', 'Username': 'dear', 'User ScreenName': 'russiapiill', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809127b5825eb7760e8c'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955715374817281, 'Content': 'RT @joesonka: The Governors orders were, and continue to be, necessary to slow thespread of COVID19 and protect the health and safety o', 'Username': 'Justkat54', 'User ScreenName': 'justkat54', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809127b5825eb7760e8e'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955715211243521, 'Content': 'paddydylepage Man living with flu and colds for millenia n we still use old remedies, n seriously think there will ', 'Username': 'Martina McCabe', 'User ScreenName': 'MartinaMcCabe3', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809227b5825eb7760e90'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955715731329030, 'Content': '@misstremmannz @DanCrenshawTX They spent months and months under much laxer restrictions business thriving, kids ', 'Username': 'Daniel R. Eckless', 'User ScreenName': 'dannyeckless', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809227b5825eb7760e92'), 'Timestamp': 'Thu Nov 12 18:31:10 +0000 2020', 'ID': 1326955715878133761, 'Content': 'RT @YaOnlyLivvOnce: Buy my merch ', 'Username': 'Kameron Kam Dre esen', 'User ScreenName': 'KDWAG4Life', 'User Location': 'Indianapolis, IN', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809227b5825eb7760e94'), 'Timestamp': 'Thu Nov 12 18:31:11 +0000 2020', 'ID': 1326955717052534791, 'Content': 'He's right, yelling and singing has been part of the problem with super spreader events. I had even hoped the met ', 'Username': 'ChristaParis', 'User ScreenName': 'ChristaParis', 'User Location': 'Barba Paris, France', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809327b5825eb7760e96'), 'Timestamp': 'Thu Nov 12 18:31:11 +0000 2020', 'ID': 132695571492928515, 'Content': 'RT @NYGovCuomo: COVID fatigue is one thing. COVID denial is another.Remember: You dont put yourself at risk when you host large ind', 'Username': 'Peggy S Schipper', 'User ScreenName': 'PegSchip44', 'User Location': 'Philadelphia area', 'Truncated': '', 'Extended Tweet': ''}
Tweet:
{'_id': ObjectId('5fad809327b5825eb7760e98'), 'Timestamp': 'Thu Nov 12 18:31:11 +0000 2020', 'ID': 1326955717824286729, 'Content': 'RT @beengot: If your from . U.S.A, U.K , South Africa , Australia or Canada please watch out for the nri paramilitant group the base', 'Username': 'Soph Professional Tr Cool Simp', 'User ScreenName': 'TrayKool', 'User Location': None, 'Truncated': '', 'Extended Tweet': ''}
ssamkit108@ubuntu-spark:~/Assignment-3/1- Twitter Data Extraction and cleaning$
```

Figure 8 Execution of cleaning process

3. Cleaned data in ProcessedDb:

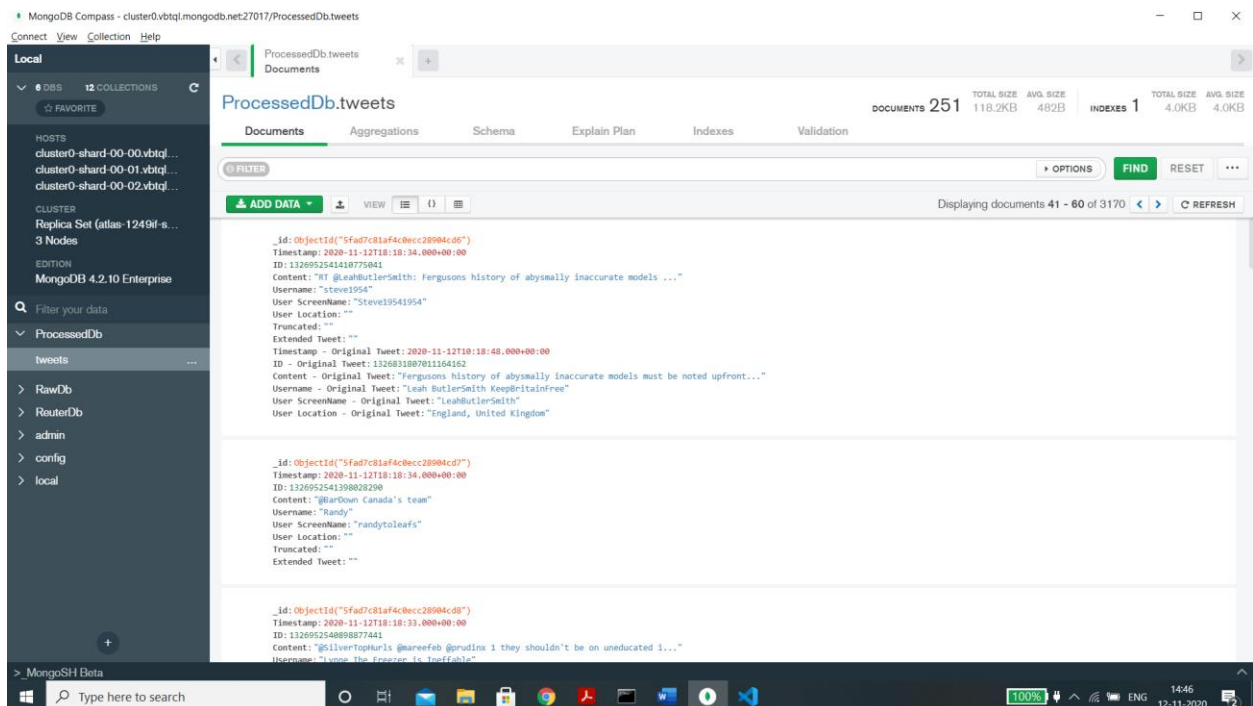


Figure 9 ProcessedDb after cleaning

News Articles data extraction:

Data extraction for news articles from the SGM file is done using python script. I have extracted data from the text tag. Basically it has mainly three tags such as title, dateline, and body. After reading both file I have cleaned it and stored in ReuterDb. The detailed steps are explained further.

1. Open and read both files.
2. Connect to MongoDB database named ReuterDb.
3. Create an instance of Articles collection for MongoDB.
4. Extract the content of the tag from the file using regular expression.
5. Clean the content of body, title and dateline.
6. Create an object of one article and store it in database.
7. Repeat this process until all files are not read.

One Document contains one news article. To clean the data of text tag, I have used regular expression.

Output for news articles data extraction:

1. Initial status of ReuterDb

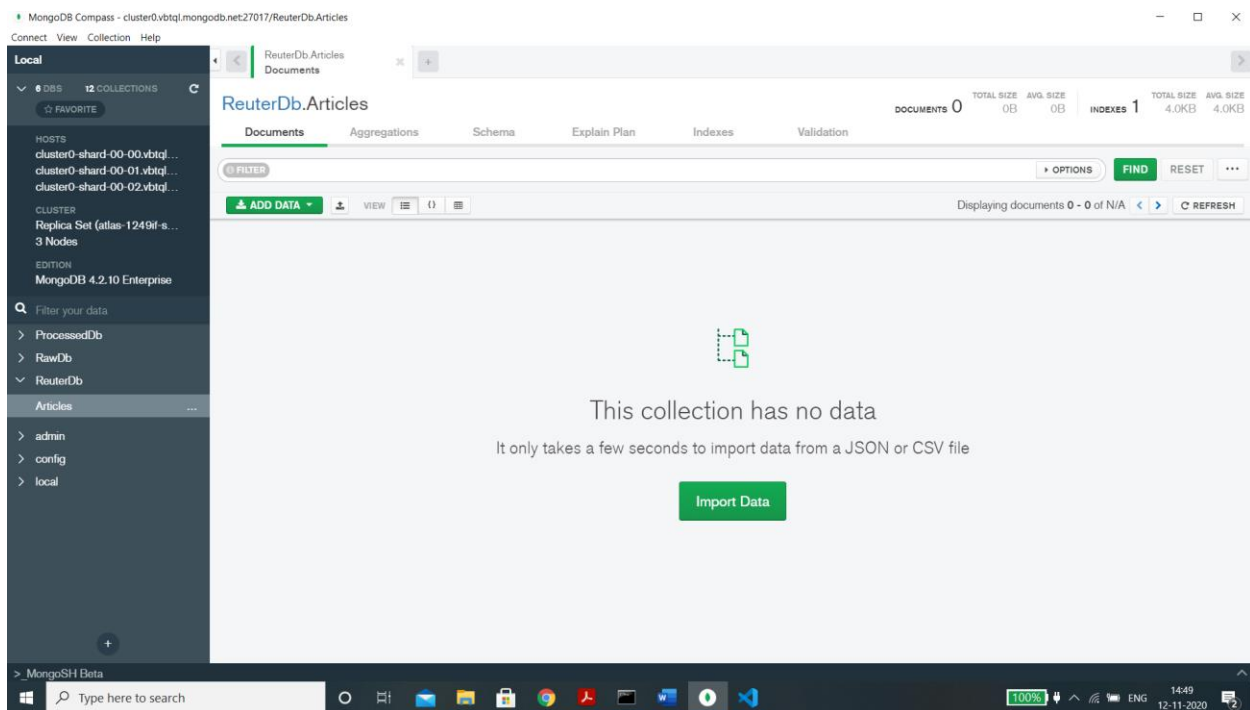


Figure 10 Initial state of ReuterDb

2. Executing news article data extraction:

```
ssamkit108@ubuntu-spark: ~/Assignment-3/2- Reuter News Article - Google Chrome
ssh.cloud.google.com/projects/assignment-2-293219/zones/us-central1-a/instances/ubuntu-spark?useAdminProxy=true&authuser=0&hl=en_US&projectNumber=634910126197

250 mln Swiss francs lead manager Credit Suisse said. REUTER 4#3: ", 'TITLE': 'JAPAN MINISTRY SAYS OPEN FARM TRADE
WOULD HIT U.S.', 'DATELINE': ' ' ZURICH April 8 - ' )
('BODY': 'Mead Corp said the outlook for its major paper markets looks strong for the second quarter and augurs well
l for its earnings in 1987. "The generally strong outlook bodes well for significantly improved earnings this y
ear" Burnell Roberts chairman and chief executive officer said. Earlier the company reported first quarter earn
ings of 34.2 mln dlrs or 1.09 dlrs a share versus 20.3 mln dlrs or 65 cts a share in last year\'s first quarter.
In 1986 the company reported earnings from continuing operations of 109.3 mln dlrs or 3.50 dlrs a share. Mead
said its first quarter benefitted from stronger market conditions and improved operations. "The combination of
capital improvement programs and more employee involvement has been paying off throughout our paper operations" Ro
berts said. He added that Mead\'s pulp and paperboard businesses are operating well as prices have improved and
strong demand has placed most products in a sold-out position through the middle of the year. Mead said sales
of its unleached coated paperboard was particularly strong up 13 pct versus the first quarter 1986. ', 'TITLE': 'A
MATIL PROPOSES TWO-FOR-FIVE BONUS SHARE ISSUE', 'DATELINE': ' ' DAYTON Ohio April 8 - ' )
('BODY': 'Endotronics Inc said because of its decisions to discontinue development of health care products reorgani
ze under Chapter 11 and establish adequate allowances for uncollectible receivables it expects to establish reserve
s totaling about 11 mln dlrs in the March 31 second quarter. The company said it plans to seek buyers for healt
h care technologies including proprietary rights to a Hepatitis B Vaccine and technologies related to LAK cell canc
er immunotherapy. ', 'TITLE': 'BOWATER 1986 PRETAX PROFITS RISE 15.6 MLN STG', 'DATELINE': ' ' MINNEAPOLIS April 8
- ' )
('BODY': 'Shr profit 20 cts vs loss three cts Net profit 849299 vs loss 82512 Revs 7929138 vs 3849224 ', 'T
ITLE': 'U.K. MONEY MARKET DEFICIT FORECAST AT 250 MLN STG', 'DATELINE': ' ' WOODSTOCK Ontario April 8 - ' )
('BODY': 'Comstock Group Inc said it has signed a letter of intent to sell about 20 mln dlrs of convertible preferre
d stock to <lt;Spie Batignolles SA> of Paris\' Spie Group Inc U.S. subsidiary. Details were not disclosed. ", '
TITLE': 'SOUTH KOREA MOVES TO SLOW GROWTH OF TRADE SURPLUS', 'DATELINE': ' ' DANBURY Conn. April 8 - ' )
('BODY': 'QVC Network Inc said its agreement in principle with Safeguard Scientifics Inc <lt;SFE> allows Safeguard
to name the majority of QVC\'s directors only if a seven mln dlr indebtedness to Safeguard is in default. Yester
day QVC said announced it entered into the agreement in principle with Safeguard under which QVC would receive 19 m
ln dlrs in financing including seven mln dlrs of QVC notes to be purchased by Safeguard and a six-mln-dlr revolving
credit facility to be provided by a local bank. QVC said as long as the seven mln dlr indebtedness to Safeguard
remains outstanding Safeguard will be able to name three of QVC\'s nine directors. Safeguard\'s ability to name a m
ajority of QVC\'s directors will be triggered only if the seven mln dlr indebtedness to Safeguard is in default the c
ompany said. ", 'TITLE': 'FINNS AND CANADIANS TO STUDY MTBE PRODUCTION PLANT', 'DATELINE': ' ' WESTCHESTER Pa. Apr
il 8 - ' )
('BODY': 'Shr primary 78 cts vs 68 cts Shr diluted 75 cts vs 68 cts Qtrly div six cts vs five cts Net 7
929000 vs 6569000 Revs 78.7 mln vs 61.9 mln NOTE: Pay date for the qtrly div is April 28 for shareholders o
f record April 20. ', 'TITLE': 'GERMAN ENGINEERING WAGE-ROUND TALKS BREAK DOWN', 'DATELINE': ' ' BALTIMORE April 8
- ' )
('BODY': 'Equitable Resources Inc said it filed with the Securities and Exchange Commission a registration statemen
t covering a 75 mln dlr issue of units. Each unit will consist of a 1000 dlr face amount 25-year debenture with
up to 21 five-year warrants to purchase the company\'s common stock. Each warrant will equal one share. The debentu
res will be non-refundable for 10 years. Proceeds will be used to repay short-term loans incurred to finance pa
rt of Equitable\'s 1986 capital expenditure program and the redemption of 9-5/8 pct and 10-1/2 pct first mortgage bo
nds of 1995. First Boston will manage the issue. ", 'TITLE': 'CRA SOLD FORREST GOLD FOR 76 MLN DLRS - WHIM CREEK',
'DATELINE': ' ' NEW YORK April 8 - ' )
('BODY': '4thh qtr Feb 28 Shr 46 cts vs 22 cts Net 2139034 vs 854182 Sales 30.8 mln vs 20.6 mln Avg
shrs 5280854 vs 4559646 Year Shr 1.34 dlrs vs 1.15 dlrs Net 5936117 vs 4156171 Sales 107.2 mln vs
71.6 mln Avg shrs 5281387 vs 3616183 NOTE: Town and Country Jewelry Manufacturing Corp. ', 'TITLE': 'MALAYS
IA SETS 100 MLN SWISS FRANC NOTES ISSUE', 'DATELINE': ' ' CHELSEA Mass. April 8 - ' )
Total: 1560
ssamkit108@ubuntu-spark:~/Assignment-3/2- Reuter News Article$
```

Figure 11 Reuter data extraction

3. After running script of data extraction:

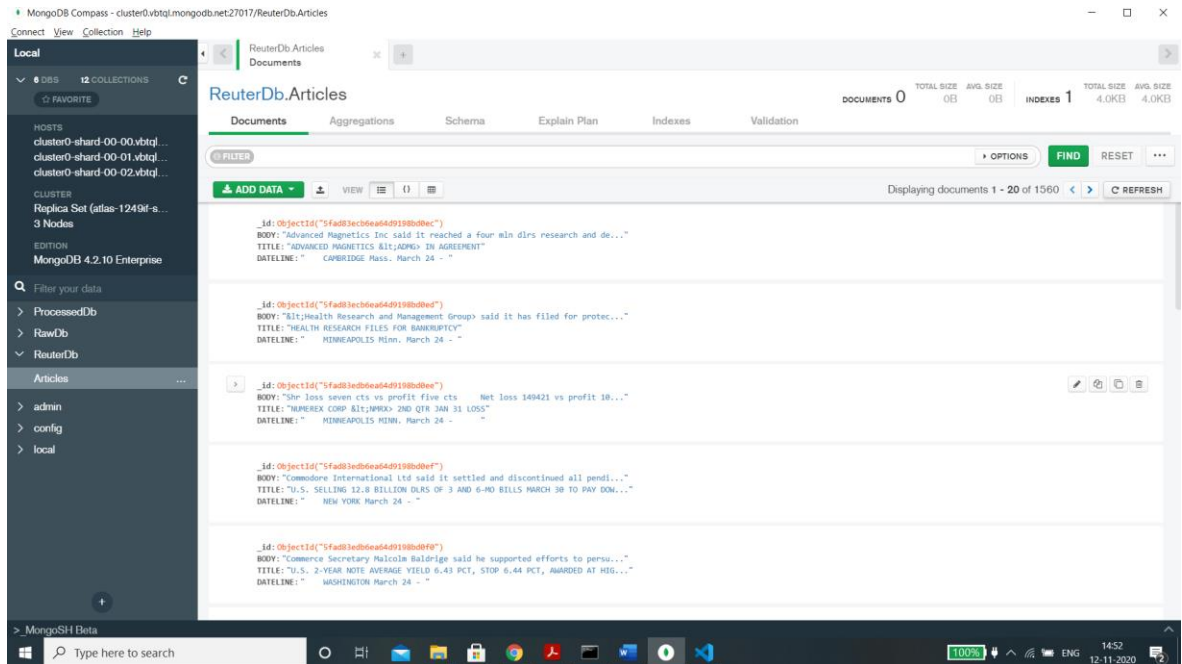


Figure 12 ProcessedDb after data added

Map Reduce to perform count:

Refer the MapReduce python script which is responsible to perform the task using Apache Spark. I have used pyspark library to achieve the distributed computation. Implemented MapReduce approach to count the frequency of keywords in tweets and news articles. The detailed steps are explained further.

1. Create SparkContext to allocate the spark cluster.
2. Sparkcontext is responsible to manage jobs and distributed processing.
3. Fetched tweets from the ProcessedDb database.
4. Splitted content of tweets using single space and extracted all words of tweets.
5. Created a list containing all words of tweets.
6. Filtered the words on the basis of given keywords and remove unnecessary words from the list.
7. Used lambda function to filter the words and stored relevant words in the list.
8. Fetched news articles from the ReuterDb database.
9. Splitted body of news article and extracted all words of news article.
10. Filtered the words on the basis of given keywords and removed unnecessary words from the list.
11. Used lambda function to filter the words and stored in separate list.
12. Merged two list of tweets and news article.
13. Applied MapReduce Approach to count the frequency of words. There are mainly two components one is Map and second is Reduce.
14. Map function takes a set of data in terms of list and convert it into the set of data in terms of key-value pair. Initially the value is always 1 because we are considering single word as a key.
15. Converted the conventional list into the distributed dataset (RDD) using `parallelize()` method of sparkcontext. This method will convert a collection to RDD for distributed processing.
16. Reduce function takes output of map function and combines those data into a set and incrementing value based on occurrence.
17. Reduce function will give word with the occurrences.
18. Displayed the word count as an output.

The Mapreduce program is executed on GCP in Apache spark cluster. To execute that I have used Spark-submit command which executes python script in apache spark. It is shown in figure 13 and 14.

Output of MapReduce Program:

```
ssamkit108@ubuntu-spark: ~/Assignment-3/3- Data processing using MapReduce - Google Chrome
ssh.cloud.google.com/projects/assignment-2-293219/zones/us-central1-a/instances/ubuntu-spark?useAdminProxy=true&authuser=0&hl=en_US&projectNumber=634910126197

ssamkit108@ubuntu-spark:~/Assignment-3/3- Data processing using MapReduce$ nano MapReduce.py
ssamkit108@ubuntu-spark:~/Assignment-3/3- Data processing using MapReduce$ spark-submit MapReduce.py
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.0.1-jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/13 00:26:11 WARN NativeCodeLoader: Unable to load native-heapoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
20/11/13 00:26:12 INFO SparkContext: Running Spark version 3.0.1
20/11/13 00:26:12 INFO ResourceUtils: =====
20/11/13 00:26:12 INFO ResourceUtils: Resources for spark.driver:
20/11/13 00:26:12 INFO ResourceUtils: =====
20/11/13 00:26:12 INFO SparkContext: Submitted application: word count
20/11/13 00:26:12 INFO SecurityManager: Changing view acls to: ssamkit108
20/11/13 00:26:12 INFO SecurityManager: Changing modify acls to: ssamkit108
20/11/13 00:26:12 INFO SecurityManager: Changing view acls groups to:
20/11/13 00:26:12 INFO SecurityManager: Changing modify acls groups to:
20/11/13 00:26:12 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(ssamkit108); groups with view permissions: Set(); users with
modify permissions: Set(ssamkit108); groups with modify permissions: Set()
20/11/13 00:26:13 INFO SparkEnv: Registering MapOutputTracker
20/11/13 00:26:13 INFO SparkEnv: Registering BlockManagerMaster
20/11/13 00:26:13 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
20/11/13 00:26:13 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
20/11/13 00:26:13 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
20/11/13 00:26:13 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-3fd5df96-12ce-401d-b994-2d8f2d380793
20/11/13 00:26:13 INFO MemoryStore: MemoryStore started with capacity 434.4 MiB
20/11/13 00:26:13 INFO SparkEnv: Registering OutputCommitCoordinator
20/11/13 00:26:13 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/11/13 00:26:13 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:4040
20/11/13 00:26:14 INFO Executor: Starting executor ID driver on host ubuntu-spark-us-central1-a.c.assignment-2-293219.internal
20/11/13 00:26:14 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 41967.
20/11/13 00:26:14 INFO NettyBlockTransferService: Server created on ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:41967
20/11/13 00:26:14 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/11/13 00:26:14 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, ubuntu-spark-us-central1-a.c.assignment-2-293219.internal, 41967, None)
20/11/13 00:26:14 INFO BlockManagerMasterEndpoint: Registering block manager ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:41967 with 434.4 MiB RAM, BlockManagerId(driver, ubuntu-spark
-us-central1-a.c.assignment-2-293219.internal, 41967, None)
20/11/13 00:26:14 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, ubuntu-spark-us-central1-a.c.assignment-2-293219.internal, 41967, None)
20/11/13 00:26:14 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ubuntu-spark-us-central1-a.c.assignment-2-293219.internal, 41967, None)
Frequency counts:
20/11/13 00:26:16 INFO DAGScheduler: Registering RDD 2 (reduceByKey at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:16) as input to shuffle 0
20/11/13 00:26:16 INFO DAGScheduler: ShuffleMapStage 0 (reduceByKey at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:16) finished in 1.572 s
20/11/13 00:26:16 INFO DAGScheduler: looking for newly runnable stages
20/11/13 00:26:16 INFO DAGScheduler: running: Set()
20/11/13 00:26:16 INFO DAGScheduler: waiting: Set(ResultStage 1)
20/11/13 00:26:16 INFO DAGScheduler: failed: Set()
20/11/13 00:26:16 INFO DAGScheduler: Submitting ResultStage 1 (PythonRDD[5] at toLocalIterator at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:18), which has no m
issing parents
20/11/13 00:26:16 INFO MemoryStore: Block broadcast_1 stored as values in memory (estimated size 9.8 KiB, free 434.4 MiB)
20/11/13 00:26:16 INFO MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 5.7 KiB, free 434.4 MiB)
20/11/13 00:26:16 INFO BlockManagerInfo: Added broadcast_1_piece0 in memory on ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:41967 (size: 5.7 KiB, free: 434.4 MiB)
20/11/13 00:26:16 INFO SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1223
20/11/13 00:26:16 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (PythonRDD[5] at toLocalIterator at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py
:18) (first 15 tasks are for partitions Vector(0))
20/11/13 00:26:16 INFO TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
20/11/13 00:26:16 INFO TaskSetManager: Starting task 0.0 in stage 1.0 (TID 1, ubuntu-spark-us-central1-a.c.assignment-2-293219.internal, executor driver, partition 0, NODE_LOCAL, 7143 bytes)
20/11/13 00:26:16 INFO Executor: Running task 0.0 in stage 1.0 (TID 1)
20/11/13 00:26:16 INFO ShuffleBlockFetcherIterator: Getting 1 (109.0 B) non-empty blocks including 1 (109.0 B) local and 0 (0.0 B) host-local and 0 (0.0 B) remote blocks
20/11/13 00:26:16 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 18 ms
20/11/13 00:26:16 INFO PythonRunner: Times: total = 14, boot = -626, init = 640, finish = 0
20/11/13 00:26:16 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 2030 bytes result sent to driver
20/11/13 00:26:16 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 196 ms on ubuntu-spark-us-central1-a.c.assignment-2-293219.internal (executor driver) (1/1)
20/11/13 00:26:16 INFO DAGScheduler: ResultStage 1 (toLocalIterator at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:18) finished in 0.219 s
20/11/13 00:26:16 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
20/11/13 00:26:16 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
20/11/13 00:26:16 INFO TaskSchedulerImpl: Killing all running tasks in stage 1: Stage finished
canada -> 348
winter -> 132
safety -> 59
snow -> 51
cold -> 43
flu -> 32
storm -> 24
rain -> 15
hot -> 13
indoor -> 10
ice -> 8
20/11/13 00:26:18 INFO SparkContext: Invoking stop() from shutdown hook
20/11/13 00:26:18 INFO SparkUI: Stopped Spark web UI at http://ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:4040
20/11/13 00:26:18 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
20/11/13 00:26:18 INFO MemoryStore: MemoryStore cleared
20/11/13 00:26:18 INFO BlockManager: BlockManager stopped
20/11/13 00:26:18 INFO BlockManagerMaster: BlockManagerMaster stopped
20/11/13 00:26:18 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
20/11/13 00:26:18 INFO SparkContext: Successfully stopped SparkContext
20/11/13 00:26:18 INFO ShutdownHookManager: Shutdown hook called
20/11/13 00:26:18 INFO ShutdownHookManager: Deleting directory /tmp/spark-d61f3265-0870-448b-9ebc-cd1e1cfb005d
20/11/13 00:26:18 INFO ShutdownHookManager: Deleting directory /tmp/spark-636e5be7-a69e-49a6-a68d-1b54fa15b552/pyspark-3b59309a-2e1d-46f1-be46-b2641e81bf32
20/11/13 00:26:18 INFO ShutdownHookManager: Deleting directory /tmp/spark-636e5be7-a69e-49a6-a68d-1b54fa15b552
ssamkit108@ubuntu-spark:~/Assignment-3/3- Data processing using MapReduce$
```

Figure 13 MapReduce Output-1

```
ssamkit108@ubuntu-spark: ~/Assignment-3/3- Data processing using MapReduce - Google Chrome
ssh.cloud.google.com/projects/assignment-2-293219/zones/us-central1-a/instances/ubuntu-spark?useAdminProxy=true&authuser=0&hl=en_US&projectNumber=634910126197

20/11/13 00:26:18 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port: 41337
20/11/13 00:26:18 INFO DAGScheduler: ShuffleMapStage 0 (reduceByKey at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:16) finished in 1.572 s
20/11/13 00:26:18 INFO DAGScheduler: looking for newly runnable stages
20/11/13 00:26:18 INFO DAGScheduler: running: Set()
20/11/13 00:26:18 INFO DAGScheduler: waiting: Set(ResultStage 1)
20/11/13 00:26:18 INFO DAGScheduler: failed: Set()
20/11/13 00:26:18 INFO DAGScheduler: Submitting ResultStage 1 (PythonRDD[5] at toLocalIterator at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:18), which has no m
issing parents
20/11/13 00:26:18 INFO MemoryStore: Block broadcast_1 stored as values in memory (estimated size 9.8 KiB, free 434.4 MiB)
20/11/13 00:26:18 INFO MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 5.7 KiB, free 434.4 MiB)
20/11/13 00:26:18 INFO BlockManagerInfo: Added broadcast_1_piece0 in memory on ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:41967 (size: 5.7 KiB, free: 434.4 MiB)
20/11/13 00:26:18 INFO SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1223
20/11/13 00:26:18 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (PythonRDD[5] at toLocalIterator at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py
:18) (first 15 tasks are for partitions Vector(0))
20/11/13 00:26:18 INFO TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
20/11/13 00:26:18 INFO TaskSetManager: Starting task 0.0 in stage 1.0 (TID 1, ubuntu-spark-us-central1-a.c.assignment-2-293219.internal, executor driver, partition 0, NODE_LOCAL, 7143 bytes)
20/11/13 00:26:18 INFO Executor: Running task 0.0 in stage 1.0 (TID 1)
20/11/13 00:26:18 INFO ShuffleBlockFetcherIterator: Getting 1 (109.0 B) non-empty blocks including 1 (109.0 B) local and 0 (0.0 B) host-local and 0 (0.0 B) remote blocks
20/11/13 00:26:18 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 18 ms
20/11/13 00:26:18 INFO PythonRunner: Times: total = 14, boot = -626, init = 640, finish = 0
20/11/13 00:26:18 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 2030 bytes result sent to driver
20/11/13 00:26:18 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 196 ms on ubuntu-spark-us-central1-a.c.assignment-2-293219.internal (executor driver) (1/1)
20/11/13 00:26:18 INFO DAGScheduler: ResultStage 1 (toLocalIterator at /home/ssamkit108/Assignment-3/3- Data processing using MapReduce/MapReduce.py:18) finished in 0.219 s
20/11/13 00:26:18 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
20/11/13 00:26:18 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
20/11/13 00:26:18 INFO TaskSchedulerImpl: Killing all running tasks in stage 1: Stage finished
canada -> 348
winter -> 132
safety -> 59
snow -> 51
cold -> 43
flu -> 32
storm -> 24
rain -> 15
hot -> 13
indoor -> 10
ice -> 8
20/11/13 00:26:18 INFO SparkContext: Invoking stop() from shutdown hook
20/11/13 00:26:18 INFO SparkUI: Stopped Spark web UI at http://ubuntu-spark-us-central1-a.c.assignment-2-293219.internal:4040
20/11/13 00:26:18 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
20/11/13 00:26:18 INFO MemoryStore: MemoryStore cleared
20/11/13 00:26:18 INFO BlockManager: BlockManager stopped
20/11/13 00:26:18 INFO BlockManagerMaster: BlockManagerMaster stopped
20/11/13 00:26:18 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
20/11/13 00:26:18 INFO SparkContext: Successfully stopped SparkContext
20/11/13 00:26:18 INFO ShutdownHookManager: Shutdown hook called
20/11/13 00:26:18 INFO ShutdownHookManager: Deleting directory /tmp/spark-d61f3265-0870-448b-9ebc-cd1e1cfb005d
20/11/13 00:26:18 INFO ShutdownHookManager: Deleting directory /tmp/spark-636e5be7-a69e-49a6-a68d-1b54fa15b552/pyspark-3b59309a-2e1d-46f1-be46-b2641e81bf32
20/11/13 00:26:18 INFO ShutdownHookManager: Deleting directory /tmp/spark-636e5be7-a69e-49a6-a68d-1b54fa15b552
ssamkit108@ubuntu-spark:~/Assignment-3/3- Data processing using MapReduce$
```

Figure 14 MapReduce Output-2

Highest Frequency: Canada → 348

Lowest Frequency: Ice → 8

Data Visualization using Graph Database:

Created Node of various keywords. The cypher query for creating nodes are given below:

```
CREATE (storm:Storm {
  title: "Storm"
})
CREATE (winter:Winter {
  title: "Winter"
})
CREATE (canada:Canada {
  title: "Canada"
})
CREATE (hot:Hot {
  title: "Hot"
})
CREATE (cold:Cold {
  title: "Cold"
})
CREATE (flu:Flu {
  title: "Flu"
})
CREATE (snow:Snow {
  title: "Snow"
})
CREATE (indoor:Indoor {
  title: "Indoor"
})
CREATE (safety:Safety {
  title: "Safety"
})
CREATE (rain:Rain {
  title: "Rain"
})
CREATE (ice:Ice {
  title: "Ice"
})
```

After creating node, I have added properties to each node. These properties are identified from the relevant tweets and some properties are added by critical thinking. All the cypher queries are mentioned below:

```
MATCH (n:Canada {
  title:"Canada"
}) set n.coronaCases=277061
RETURN n
```

```
MATCH (n:Canada {
  title:"Canada"
}) set n.Capital='Ottawa'
RETURN n
```

```
MATCH (n:Canada {
  title:"Canada"
```

```

}) set n.Currency= "Canadian Dollor", n.Area='9.98 millllion sq km',
n.languages='English, French'
RETURN n

MATCH (n:Cold {
  title:"Cold"
}) set n.Temperature="<10 Celcius"
RETURN n

MATCH (n:Strom {
  title:"Strom"
}) set n.location="Halifax", n.Country="Canada", n.Date="07/11/2019",
n.Time="18:13", n.Type="Strom Surge/Tide", n.Property_Damage="68.100 million
CAD"
RETURN n

MATCH (n:Snow {
  title:"Snow"
}) set n.Density="0.1-0.8 g/cm^3", n.Strength="2.5 kPa", n.melting_temp="0
degree Celcius"
RETURN n
MATCH (n:hot {
  title:"hot"
}) set n.Temperature=">10 Celcius"
RETURN n

MATCH (n:Flu {
  title:"Flu"
}) set n.Symptoms=["Fever", "runny nose", "sore throat", "headache"],
n.duration="1 week"
RETURN n

```

There are various relations between two nodes. Relationship are added through cypher query language. Some nodes do not have any logical relations but some have. All the relations are shown in the figure 14 and cypher query is mentioned below:

```

MATCH (a:Winter), (b:Snow) CREATE (b)-[r:FALL_IN]->(a)
MATCH (a:Strom), (b:rain) Create (a)-[r:CAUSES]->(b)
MATCH (a:rain), (b:flu) Create (a)-[r:GET_SICKED]->(b)
MATCH (a:Canada), (b:Strom) Create (a)-[r:ISSUED_ALERTS]->(b)
MATCH (a:Strom), (b:Safety) Create (a)-[r:INCREASES_RISK]->(b)
MATCH (a:Strom), (b:Indoor) Create (a)-[r:STAY]->(b)
MATCH (a:Cold), (b:Flu) Create (a)-[r:IS_SYMPTON]->(b)
MATCH (a:Winter), (b:Flu) Create (b)-[r:STRIKES_IN]->(a)
MATCH (a:Indoor), (b:hot) Create (a)-[r:IS]->(b)
MATCH (a:rain), (b:Flu) Create (a)-[r:IS_A_SEASON]->(b)
MATCH (a:Canada), (b:Cold) Create (a)-[r:WEATHER]->(b)
MATCH (a:Snow), (b:ice) Create (a)-[r:CONVERTS_INTO]->(b)

```

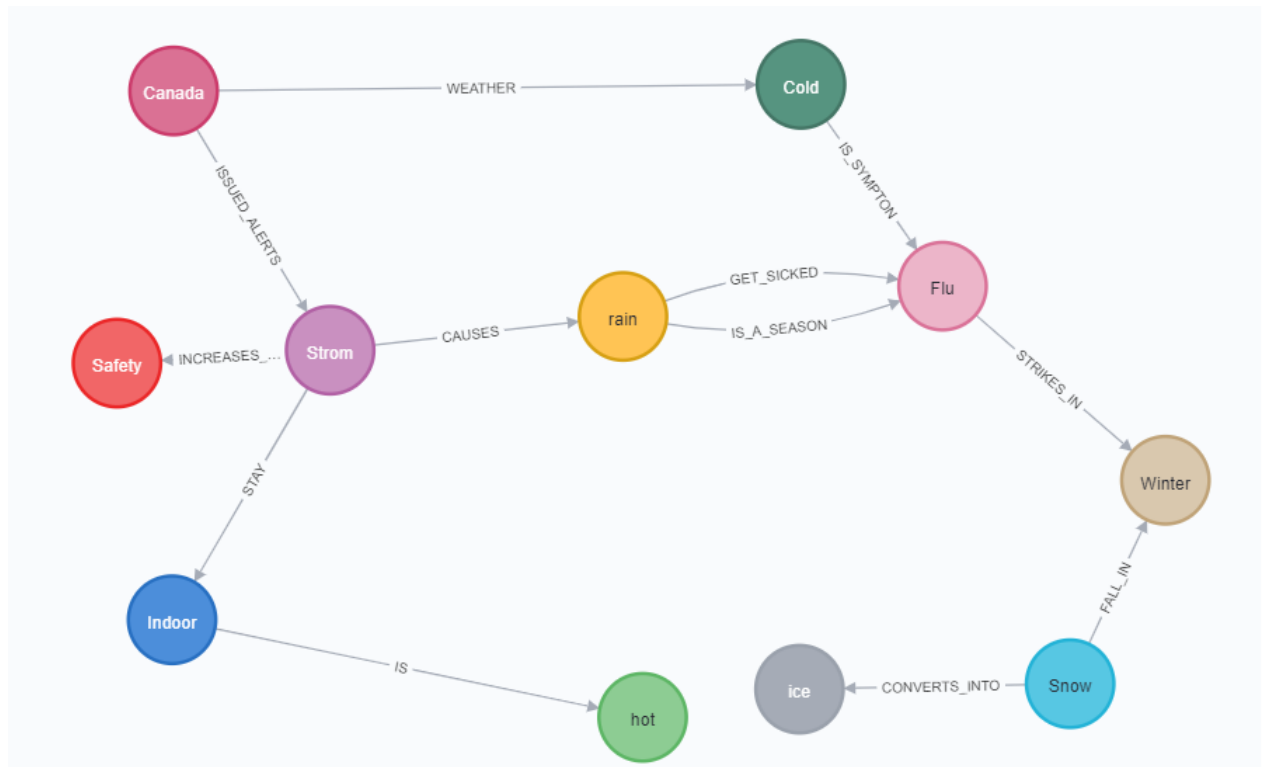



Figure 15 Graph of Nodes

References:

1. "Managed MongoDB Hosting | Database-as-a-Service", MongoDB, 2020. [Online]. Available: <https://www.mongodb.com/cloud/atlas>. [Accessed: 10- Nov- 2020]
2. "Cloud Computing Services | Google Cloud", Google Cloud, 2020. [Online]. Available: <https://cloud.google.com/>. [Accessed: 6- Nov- 2020]
3. "Querying with Cypher - Neo4j Graph Database Platform", Neo4j Graph Database Platform, 2020. [Online]. Available: <https://neo4j.com/developer/cypher/querying/>. [Accessed: 6- Nov- 2020]
4. "Tweepy", Tweepy.org, 2020. [Online]. Available: <https://www.tweepy.org/>. [Accessed: 6- Nov- 2020]
5. "Neo4j Graph Platform – The Leader in Graph Databases", Neo4j Graph Database Platform, 2020. [Online]. Available: <https://neo4j.com/>. [Accessed: 6- Nov- 2020]
6. "Welcome to Spark Python API Docs" [Online].Available : <https://spark.apache.org/docs/latest/api/python/index.html> [Accessed 08-Nov-2020]