

# COMP-SCI 5540 Principles of Big Data Management

University of Missouri-Kansas City

Department of Computer Science and Electrical Engineering

Project Report



**GitHub URL:** <https://github.com/bharathkumarna/Principles-of-BigData>

## Team – 6

Abhiram Reddy Nalla

Bharath Kumar Natesan Arumugam

Sai Kumar Ponnamaneni

Sibi Chakravarthy Ramesh

## Theme: (Wrestling)

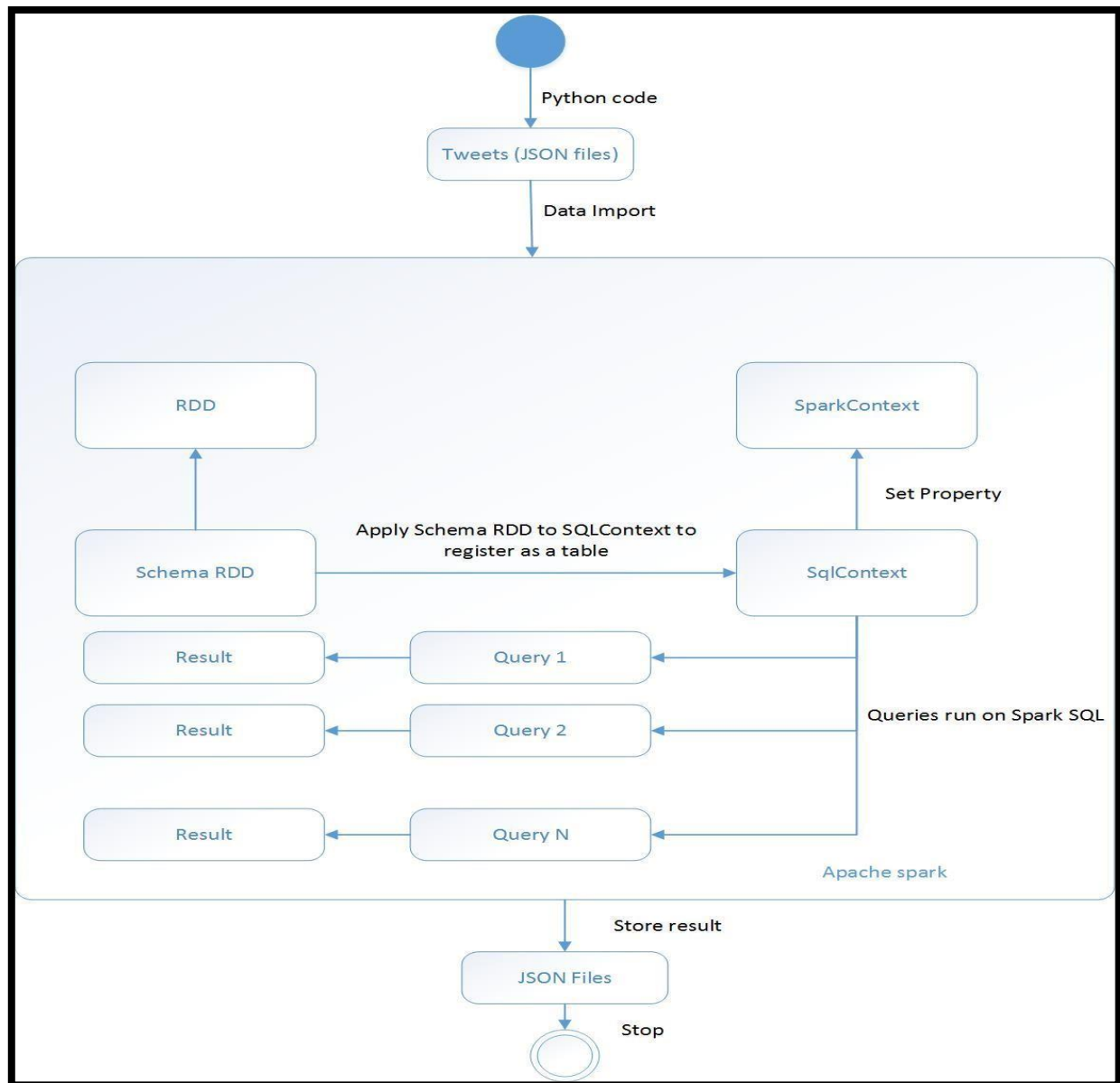
The Ultimate Fighting Championship (UFC) is the world's leading mixed martial arts (MMA) promoter and has held over 300 events to date. UFC is a combat sport abide by Unified Rules of Mixed Martial Arts where the outcomes are pre-determined and the matches are not choreographed. The UFC also connects with tens of millions of fans through its social media sites like Facebook, Instagram, and Twitter. The estimated tweets posted per hour (based on 1% sample) about #UFC is around 800.

World Wrestling Entertainment, Inc. (WWE) is an entertainment company that deals primarily in professional wrestling. WWE shows are purely entertainment based, featuring storyline-driven, scripted and choreographed matches. The estimated tweets posted per hour (based on 1% sample) about #WWE is around 750.

## References:

1. Wikipedia
2. Hashtags.org Analytics

## UML Diagram:



## Design Steps:

1. Collect social media data (tweets) using any theme as filter and store it as JSON files.
2. A Spark Context is created to establish connection to Spark Cluster.
3. SQL Context class is created which represents an entry point into all functionality in Spark SQL.
4. Data Frames are created based on content of JSON file and register it to tables.
5. Run SQL queries programmatically using SQL function on registered tables.
6. Store the returned results as JSON file.

## Libraries:

Spark Core contains the basic functionality of Spark and Spark SQL is Spark's package for working with Structured data.

1. org.apache.spark:spark-core\_2.11:2.0.02
2. org.apache.spark:spark-sql\_2.11:2.0.02

Signpost has been designed to work in conjunction with Apache HTTPComponents library for signing HTTP messages on the Scala platform in conformance with the OAuth Core 1.0 standard.

3. oauth.signpost:signpost-commonshttp4:1.2.1.22
4. org.apache.directory.studio:org.apache.httpcomponents.httpclient:4.02
5. signpost-core-1.2.1.22
6. org.apache.directory.studio:org.apache.httpcomponents.httpcore:4.02

Tweepy – An easy-to-use Python library for accessing the Twitter API.

7. tweepy-3.5.0

## APIs:

1. Twitter public REST APIs - GET followers/ids

Resource URL: <https://api.twitter.com/1.1/followers/ids.json>

Returns a collection of user IDs for every user following the specified user.

## Programming Languages:

1. Scala – to run Spark Programs.
2. Python – to run Tweets collection program.

## Environment:

### Runtime Information:

<b>Name</b>	<b>Value</b>
Java Version	1.8.0_101 (Oracle Corporation)
Scala Version	version 2.11.8

### Spark Properties:

<b>Name</b>	<b>Value</b>
spark.sql.warehouse.dir	file:///c:/tmp/spark-warehouse
spark.scheduler.mode	FIFO
spark.master	local[2]
spark.executor.id	driver
spark.driver.port	55681
spark.driver.host	192.168.1.146
spark.app.name	CountSpark
spark.app.id	local-1478459427915

### System Properties:

<b>Name</b>	<b>Value</b>
file.encoding	UTF-8
hadoop.home.dir	C:\hadoop-2.3.0\bin\tweet
idea.launcher.bin.path	C:\Program Files (x86)\JetBrains\IntelliJ IDEA Community Edition 2016.2.5\bin
os.arch	amd64
os.name	Windows 10
os.version	10.0

## Queries:

### Query 1:

#### Description:

Query to display the top 10 users who tweeted the most times.

#### Code:

```
val Query1 = sqlcontext.sql("select user.name,user.screen_name, count(user.followers_count) as tweetsCount from querytable1 group by user.screen_name,user.name order by tweetsCount desc limit 10")
```

### Query 2:

#### Description:

Query to display the top 10 users with most Sensitive Tweet numbers.

#### Code:

```
val Query2 = sqlcontext.sql("select user.name,count(user.name) as no_of_sensitive_tweets from querytable1 where possibly_sensitive=true and user.lang='en' group by user.name order by no_of_sensitive_tweets desc limit 10")
```

### Query 3:

#### Description:

Query to display the top hashtags used in my collected tweets in conjunction with data in the HashtagsTopics.txt file posted on Blackboard.

#### Code:

```
val Query3 = sqlcontext.sql("select querytable3.name,count(querytable1.text) as count from querytable1 join querytable3 on querytable1.text like concat ('%',querytable3.name,'%') group by querytable3.name order by count desc limit 10 ")
```

#### Query 4:

##### Description:

Query to display cities from which most tweets and retweets are posted.

##### Code:

```
Query4=string.flatMap(x=>(x.split(",\\\\"))) .filter(line=>line.contains("location"))  
  
.flatMap(x=>(x.split("location\\":"))) .filter(x => x!="null").filter(x => x!="")  
  
.filter(line=>line.contains(", ")) .map(temp => (temp,1)) .reduceByKey(_+_).sortBy(_._2,false)  
  
.take(10) .foreach(println)
```

#### Query 5:

##### Description:

Query to display the most popular time zones.

##### Code:

```
Query5=string.flatMap(x =>(x.split(", "))) .filter(line=>line.contains("time_zone"))  
  
.flatMap(x =>(x.split("\\\"time_zone\\\":"))) .filter(x => x!="null") .filter(x => x!="") .map(temp =>  
(temp,1)).reduceByKey(_+_).sortBy(_._2,false).take(10).foreach(println)
```



## Runtime Measurements for Queries:

<i><b>Query</b></i>	<i><b>Total (sec)</b></i>
Query 1 – Data Frame	8
Query 2 – Data Frame	5
Query 3 – Data Frame	14
Query 4 – RDD	7
Query 5 – RDD	4.9

CountSpark - Spark Jobs x

localhost:4040/jobs/

CountSpark 2.0.0

JobsStagesStorageEnvironmentExecutorsSQL

CountSpark application UI

### Spark Jobs (?)

User: bn4n5  
Total Uptime: 6.3 min  
Scheduling Mode: FIFO  
Completed Jobs: 9  
[Event Timeline](#)

#### Completed Jobs (9)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
8	<a href="#">take at sample.scala:91</a>	2016/11/11 10:06:46	0.9 s	2/2 (1 skipped)	16/16 (15 skipped)
7	<a href="#">sortBy at sample.scala:91</a>	2016/11/11 10:06:43	4 s	2/2	30/30
6	<a href="#">take at sample.scala:85</a>	2016/11/11 10:04:58	1 s	2/2 (1 skipped)	16/16 (15 skipped)
5	<a href="#">sortBy at sample.scala:85</a>	2016/11/11 10:04:51	6 s	2/2	30/30
4	<a href="#">show at sample.scala:79</a>	2016/11/11 10:04:07	11 s	2/2	202/202
3	<a href="#">run at ThreadPoolExecutor.java:1142</a>	2016/11/11 10:04:04	3 s	1/1	4/4
2	<a href="#">show at sample.scala:67</a>	2016/11/11 10:03:20	5 s	2/2	204/204
1	<a href="#">show at sample.scala:50</a>	2016/11/11 10:02:23	8 s	2/2	204/204
0	<a href="#">json at sample.scala:28</a>	2016/11/11 10:02:06	6 s	1/1	15/15

Code:

```
Collecting Tweets: from tweepy.streaming
import StreamListener from tweepy import
OAuthHandler from tweepy import Stream

#Twitter Authentication
access_token = "1048610250-QQZ8D05FWBIon130QSgjjg0XGDN0dw3LXXhP7KFt"
access_token_secret = "RRiMG6c7mIY61apEJWSwoxMMaSVN8tQwIcuK627ugp46r"
consumer_key = "RRAnQIWfIUDBpJm940WgwmpEF"
consumer_secret = "uXj3hPKmkU931K8ye5FMZemBUky4UyEQxQCz2Ej5qyS4zp0Ddw"

class
StdOutListener(StreamListener):
    def on_data(self,
data):
        print(data)        with
open('fetched_tweet.json', 'a') as tf:
            tf.write(data)
return True
    if __name__ ==
'__main__':        l =
StdOutListener()
        auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)        stream
= Stream(auth, l)

#Filter Tweets according to theme
stream.filter(track=['UFC', 'WWE'])
```

## Spark SQL Program:

```
import oauth.signpost.commonshhttp.CommonsHttpOAuthConsumer
import org.apache.commons.io.IOUtils import
org.apache.http.client.methods.HttpGet import
org.apache.http.impl.client.DefaultHttpClient import
org.apache.spark.{SparkConf, SparkContext} import
org.apache.spark.SparkConf import
org.apache.spark.SparkContext import
org.apache.spark.sql.SQLContext

object sample {
  //Twitter Authentication  val
  AccessToken = "1048610250-
QQZ8D05FWBIon130QSgJg0XGDN0dw3lXXhP7KFt";
  val AccessSecret = "RRiMG6c7mIY61apEJWSwoxMMaSVN8tQwIcuK627ugp46r";
  val ConsumerKey = "RRAnQIWfiuDBpJm940WgwpEF";  val ConsumerSecret =
"uXj3hPKmkU931K8ye5FMZemBUky4UyEQxCz2Ej5qyS4zp0Ddw";

  def main(args: Array[String]) {

    System.setProperty("hadoop.home.dir", "C:\\hadoop-
2.3.0\\bin\\tweet")
    val conf = new
SparkConf().setAppName("CountSpark").setMaster("local[2]").set("spark.
sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")  val sc = new
SparkContext(conf)  val sqlcontext = new SQLContext(sc)  import
sqlcontext.implicits._

    //Spark DataFrames
    val tweetsfile =
sqlcontext.read.json("C:\\Users\\bn4n5\\workspace\\Pbass\\mypackage\\fe
tched_tweet.json")  tweetsfile.registerTempTable("querytable1")
```

```

//Spark RDD's
val
string=sc.textFile("C:\\Users\\bn4n5\\workspace\\Pbass\\mypackage\\fetc
hed_tweet.json")
var a='Y'
while (a=='Y') {
//Menu Option
println("***** Analytical Queries using Apache Spark *****")
println("1=>Top Users who has Tweeted the most times")
println("2=>Users with Most Sensitive Tweet Numbers")
println("3=>Top Hashtags used in my collected data in conjunction with
Trending Hash tags Topics")
println("4=>Cities from which most Tweets and Retweets posted")
println("5=>Most Popular Time Zones")
println("Enter your choice:")    val
choice=readInt()    choice match {

    case 1 =>
        //Query 1 using Spark DataFrames
val Query1 = sqlcontext.sql("select
user.name,user.screen_name, count(user.followers_count) as tweetsCount
from querytable1 group by user.screen_name,user.name order by
tweetsCount desc limit 10")
Query1.write.json("C:\\Users\\bn4n5\\workspace\\Pbass\\mypackage\\Query
1")

        Query1.show()

        //Query 1 calling public API
val name = readLine("Enter screen name to find user IDs for
every user following the specified user:")
val consumer = new CommonsHttpOAuthConsumer(ConsumerKey,
ConsumerSecret)
        consumer.setTokenWithSecret(AccessToken, AccessSecret)
val request = new

```

```

HttpGet("https://api.twitter.com/1.1/followers/ids.json?cursor=-
1&screen_name=" + name)
consumer.sign(request)
    val client = new DefaultHttpClient()
val response = client.execute(request)
    println(IOUtils.toString(response.getEntity().getContent()))
println("Press Y to continue or N to exit:")    a = readChar()

```

```

case 2 =>

```

```

    //Query 2 using Spark DataFrames

```

```

val Query2 = sqlcontext.sql("select
user.name,count(user.name) as no_of_sensitive_tweets from querytable1
where possibly_sensitive=true and user.lang='en' group by user.name
order by no_of_sensitive_tweets desc limit 10")

```

```

Query2.write.json("C:\\Users\\bn4n5\\workspace\\Pbass\\mypack
age\\Query2")    Query2.show()
    println("Press Y to continue or N to exit:")
a = readChar()

```

```

case 3 =>

```

```

    //Query 3 using Spark DataFrames

```

```

    //Query 3 uses data in the PopularHahtagsAndTopics.txt file
    posted on Blackboard in conjunction with my collected data

```

```

val text =
sc.textFile("C:\\Users\\bn4n5\\workspace\\Pbass\\mypackage\\PopularHah
tagsAndTopics.txt").map(_._split("/n")).map(f rt =>
Text(frt(0))).toDF()
    text.registerTempTable("querytable")
    val Query=sqlcontext.sql("select querytable.name from
querytable where querytable.name like '%#UFC%' or querytable.name like
'%#WWE%' or querytable.name like '%#MMA%' ")
Query.registerTempTable("querytable3")
val Query3 = sqlcontext.sql("select

```

```
querytable3.name,count(querytable1.text) as count from querytable1 join
querytable3 on querytable1.text like concat ('%',querytable3.name,'%')
group by querytable3.name order by count desc limit 10 ")
```

```
Query3.write.json("C:\\Users\\bn4n5\\workspace\\Pbass\\mypack
age\\Query3")          Query3.show();
    println("Press Y to continue or N to exit:")
a = readChar()
```

```
case 4 =>
```

```
    //Query 4 using Spark RDD's
    val Query4=string.flatMap(x
=>(x.split(",\\\""))).filter(line=>line.contains("location")).flatMap(x=
>(x.split("location\\\":"))).filter(x => x!="null").filter(x => x!="")
    .filter(line=>line.contains(",")).map(temp => (temp,1))
    .reduceByKey(_+_).sortBy(_._2,false).take(10).foreach(println)
    println("Press Y to continue or N to exit:")
    a = readChar()
```

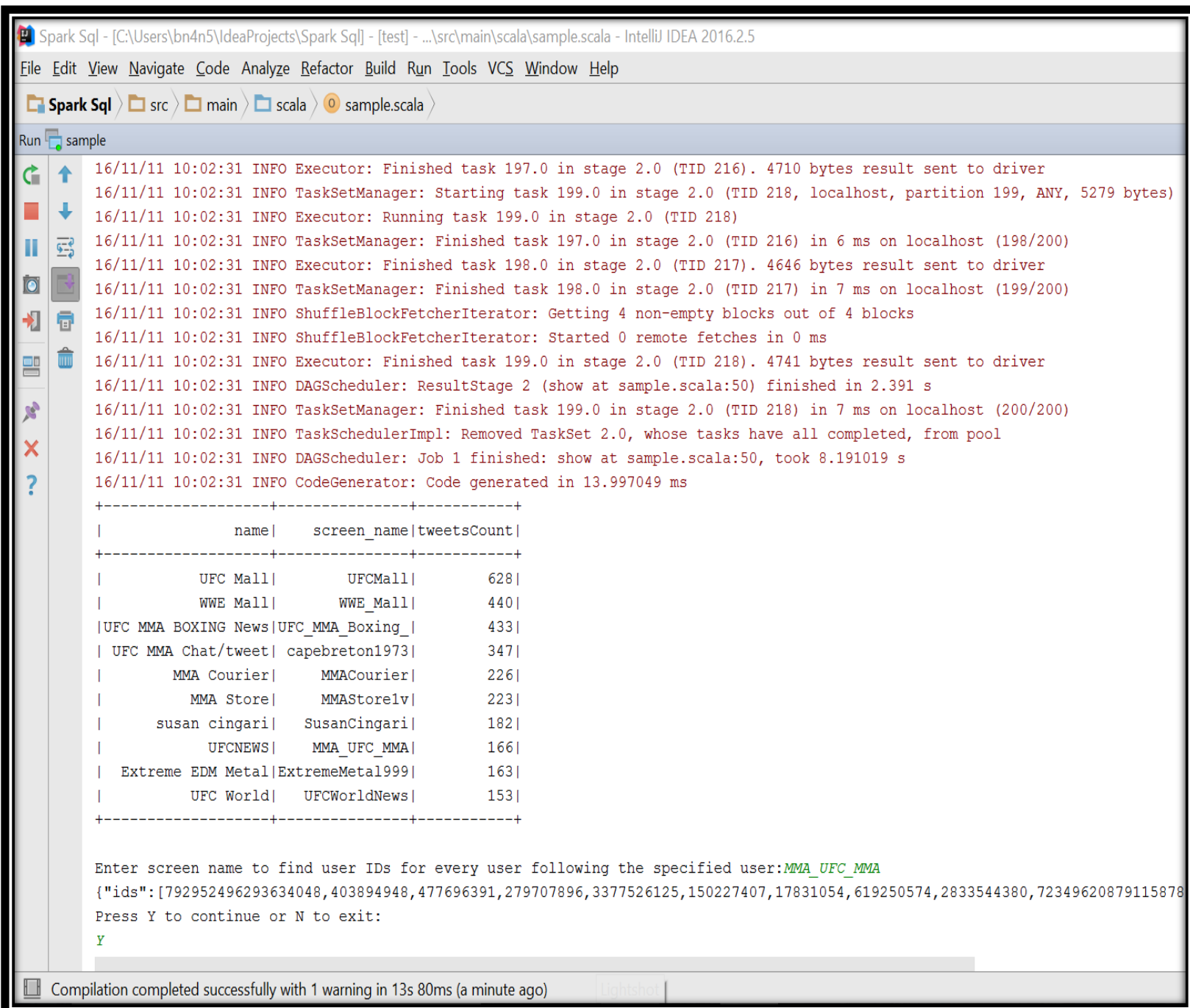
```
case 5 =>
```

```
    //Query 5 using Spark RDD's
    val Query5=string.flatMap(x
=>(x.split(","))).filter(line=>line.contains("time_zone")).flatMap(x
=>(x.split("\\\"time_zone\\\":"))).filter(x => x!="null").filter(x =>
x!="").map(temp => (temp,1)) .reduceByKey(_+_).sortBy(_._2,false)
    .take(10).foreach(println)
    println("Press Y to continue or N to exit:")
    a = readChar()
    }
    }
    }
}
```

```
case class Text(name: String)
```

## Output:

### Query 1: Top Users who has Tweeted the most times (Twitter API)



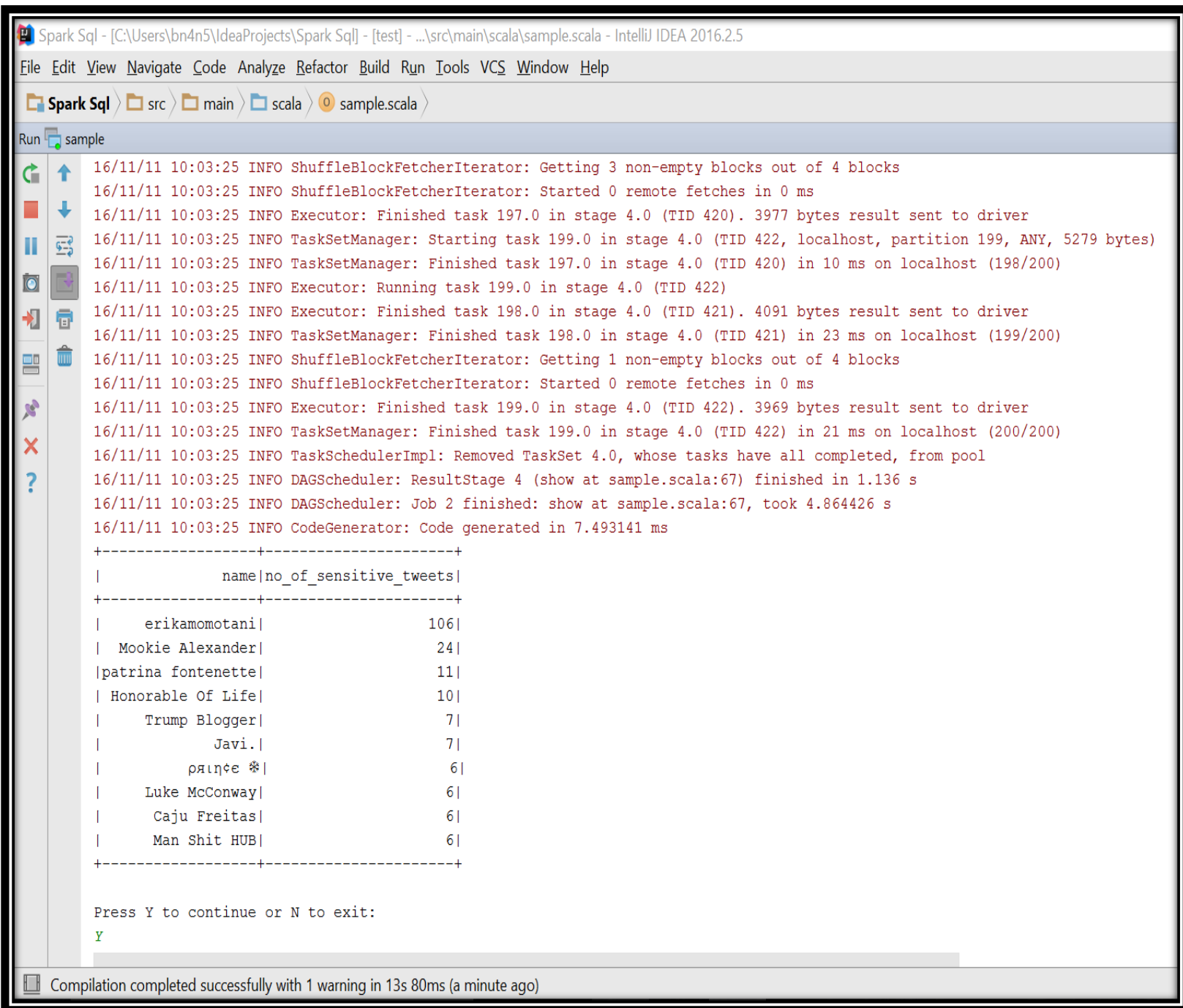
```
Spark Sql - [C:\Users\bn4n5\IdeaProjects\Spark Sql] - [test] - ...src\main\scala\sample.scala - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Spark Sql src main scala sample.scala
Run sample
16/11/11 10:02:31 INFO Executor: Finished task 197.0 in stage 2.0 (TID 216). 4710 bytes result sent to driver
16/11/11 10:02:31 INFO TaskSetManager: Starting task 199.0 in stage 2.0 (TID 218, localhost, partition 199, ANY, 5279 bytes)
16/11/11 10:02:31 INFO Executor: Running task 199.0 in stage 2.0 (TID 218)
16/11/11 10:02:31 INFO TaskSetManager: Finished task 197.0 in stage 2.0 (TID 216) in 6 ms on localhost (198/200)
16/11/11 10:02:31 INFO Executor: Finished task 198.0 in stage 2.0 (TID 217). 4646 bytes result sent to driver
16/11/11 10:02:31 INFO TaskSetManager: Finished task 198.0 in stage 2.0 (TID 217) in 7 ms on localhost (199/200)
16/11/11 10:02:31 INFO ShuffleBlockFetcherIterator: Getting 4 non-empty blocks out of 4 blocks
16/11/11 10:02:31 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:02:31 INFO Executor: Finished task 199.0 in stage 2.0 (TID 218). 4741 bytes result sent to driver
16/11/11 10:02:31 INFO DAGScheduler: ResultStage 2 (show at sample.scala:50) finished in 2.391 s
16/11/11 10:02:31 INFO TaskSetManager: Finished task 199.0 in stage 2.0 (TID 218) in 7 ms on localhost (200/200)
16/11/11 10:02:31 INFO TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
16/11/11 10:02:31 INFO DAGScheduler: Job 1 finished: show at sample.scala:50, took 8.191019 s
16/11/11 10:02:31 INFO CodeGenerator: Code generated in 13.997049 ms
+-----+
|          name|      screen_name|tweetsCount|
+-----+
|      UFC Mall|      UFCMall|      628|
|      WWE Mall|      WWE_Mall|      440|
|UFC MMA BOXING News|UFC_MMA_Boxing_|      433|
| UFC MMA Chat/tweet| capebreton1973|      347|
|      MMA Courier|      MMACourier|      226|
|      MMA Store|      MMAStorelv|      223|
|      susan cingari|      SusanCingari|      182|
|      UFCNEWS|      MMA_UFC_MMA|      166|
| Extreme EDM Metal|ExtremeMetal999|      163|
|      UFC World|      UFCWorldNews|      153|
+-----+

Enter screen name to find user IDs for every user following the specified user:MMA_UFC_MMA
{"ids": [792952496293634048, 403894948, 477696391, 279707896, 3377526125, 150227407, 17831054, 619250574, 2833544380, 72349620879115878]
Press Y to continue or N to exit:
Y

Compilation completed successfully with 1 warning in 13s 80ms (a minute ago) | Lightshot
```

## Query 2:

### Users with Most Sensitive Tweet Numbers



```
Spark Sql - [C:\Users\bn4n5\IdeaProjects\Spark Sql] - [test] - ...\src\main\scala\sample.scala - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Spark Sql > src > main > scala > sample.scala >
Run sample
16/11/11 10:03:25 INFO ShuffleBlockFetcherIterator: Getting 3 non-empty blocks out of 4 blocks
16/11/11 10:03:25 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:03:25 INFO Executor: Finished task 197.0 in stage 4.0 (TID 420). 3977 bytes result sent to driver
16/11/11 10:03:25 INFO TaskSetManager: Starting task 199.0 in stage 4.0 (TID 422, localhost, partition 199, ANY, 5279 bytes)
16/11/11 10:03:25 INFO TaskSetManager: Finished task 197.0 in stage 4.0 (TID 420) in 10 ms on localhost (198/200)
16/11/11 10:03:25 INFO Executor: Running task 199.0 in stage 4.0 (TID 422)
16/11/11 10:03:25 INFO Executor: Finished task 198.0 in stage 4.0 (TID 421). 4091 bytes result sent to driver
16/11/11 10:03:25 INFO TaskSetManager: Finished task 198.0 in stage 4.0 (TID 421) in 23 ms on localhost (199/200)
16/11/11 10:03:25 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 4 blocks
16/11/11 10:03:25 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:03:25 INFO Executor: Finished task 199.0 in stage 4.0 (TID 422). 3969 bytes result sent to driver
16/11/11 10:03:25 INFO TaskSetManager: Finished task 199.0 in stage 4.0 (TID 422) in 21 ms on localhost (200/200)
16/11/11 10:03:25 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
16/11/11 10:03:25 INFO DAGScheduler: ResultStage 4 (show at sample.scala:67) finished in 1.136 s
16/11/11 10:03:25 INFO DAGScheduler: Job 2 finished: show at sample.scala:67, took 4.864426 s
16/11/11 10:03:25 INFO CodeGenerator: Code generated in 7.493141 ms
+-----+-----+
|          name|no_of_sensitive_tweets|
+-----+-----+
|   erikamomotani|          106|
| Mookie Alexander|          24|
|patrina fontenette|          11|
| Honorable Of Life|          10|
|   Trump Blogger|           7|
|         Javi.|           7|
|      раянсе ✱|           6|
|   Luke McConway|           6|
|   Caju Freitas|           6|
|   Man Shit HUB|           6|
+-----+-----+

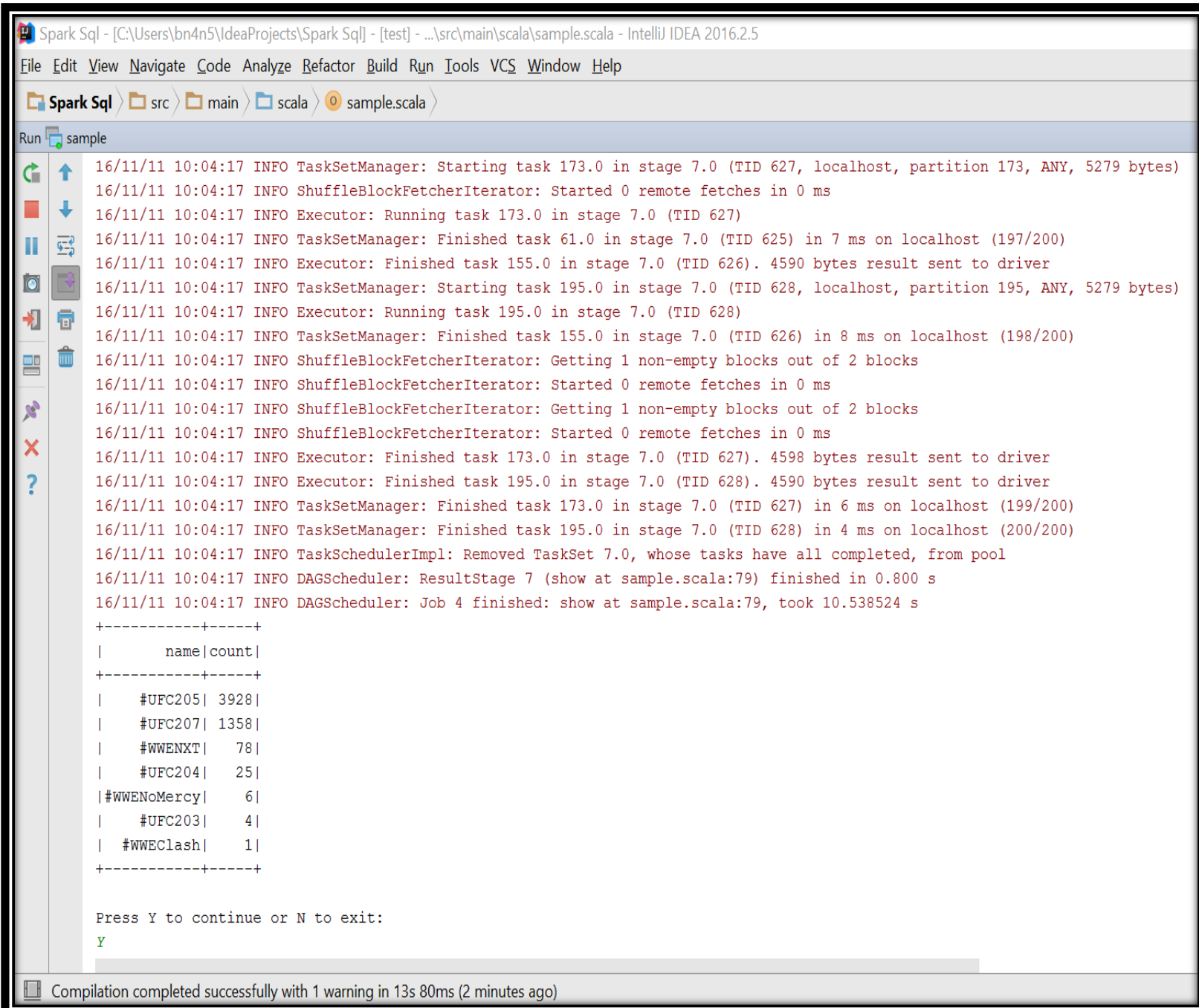
Press Y to continue or N to exit:
Y

Compilation completed successfully with 1 warning in 13s 80ms (a minute ago)
```



### Query 3:

## Top Hashtags used in my collected data in conjunction with Trending Hashtags Topics



```
Spark Sql - [C:\Users\bn4n5\IdeaProjects\Spark Sql] - [test] - ...src\main\scala\sample.scala - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Spark Sql src main scala sample.scala

Run sample

16/11/11 10:04:17 INFO TaskSetManager: Starting task 173.0 in stage 7.0 (TID 627, localhost, partition 173, ANY, 5279 bytes)
16/11/11 10:04:17 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:04:17 INFO Executor: Running task 173.0 in stage 7.0 (TID 627)
16/11/11 10:04:17 INFO TaskSetManager: Finished task 61.0 in stage 7.0 (TID 625) in 7 ms on localhost (197/200)
16/11/11 10:04:17 INFO Executor: Finished task 155.0 in stage 7.0 (TID 626). 4590 bytes result sent to driver
16/11/11 10:04:17 INFO TaskSetManager: Starting task 195.0 in stage 7.0 (TID 628, localhost, partition 195, ANY, 5279 bytes)
16/11/11 10:04:17 INFO Executor: Running task 195.0 in stage 7.0 (TID 628)
16/11/11 10:04:17 INFO TaskSetManager: Finished task 155.0 in stage 7.0 (TID 626) in 8 ms on localhost (198/200)
16/11/11 10:04:17 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 2 blocks
16/11/11 10:04:17 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:04:17 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 2 blocks
16/11/11 10:04:17 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:04:17 INFO Executor: Finished task 173.0 in stage 7.0 (TID 627). 4598 bytes result sent to driver
16/11/11 10:04:17 INFO Executor: Finished task 195.0 in stage 7.0 (TID 628). 4590 bytes result sent to driver
16/11/11 10:04:17 INFO TaskSetManager: Finished task 173.0 in stage 7.0 (TID 627) in 6 ms on localhost (199/200)
16/11/11 10:04:17 INFO TaskSetManager: Finished task 195.0 in stage 7.0 (TID 628) in 4 ms on localhost (200/200)
16/11/11 10:04:17 INFO TaskSchedulerImpl: Removed TaskSet 7.0, whose tasks have all completed, from pool
16/11/11 10:04:17 INFO DAGScheduler: ResultStage 7 (show at sample.scala:79) finished in 0.800 s
16/11/11 10:04:17 INFO DAGScheduler: Job 4 finished: show at sample.scala:79, took 10.538524 s

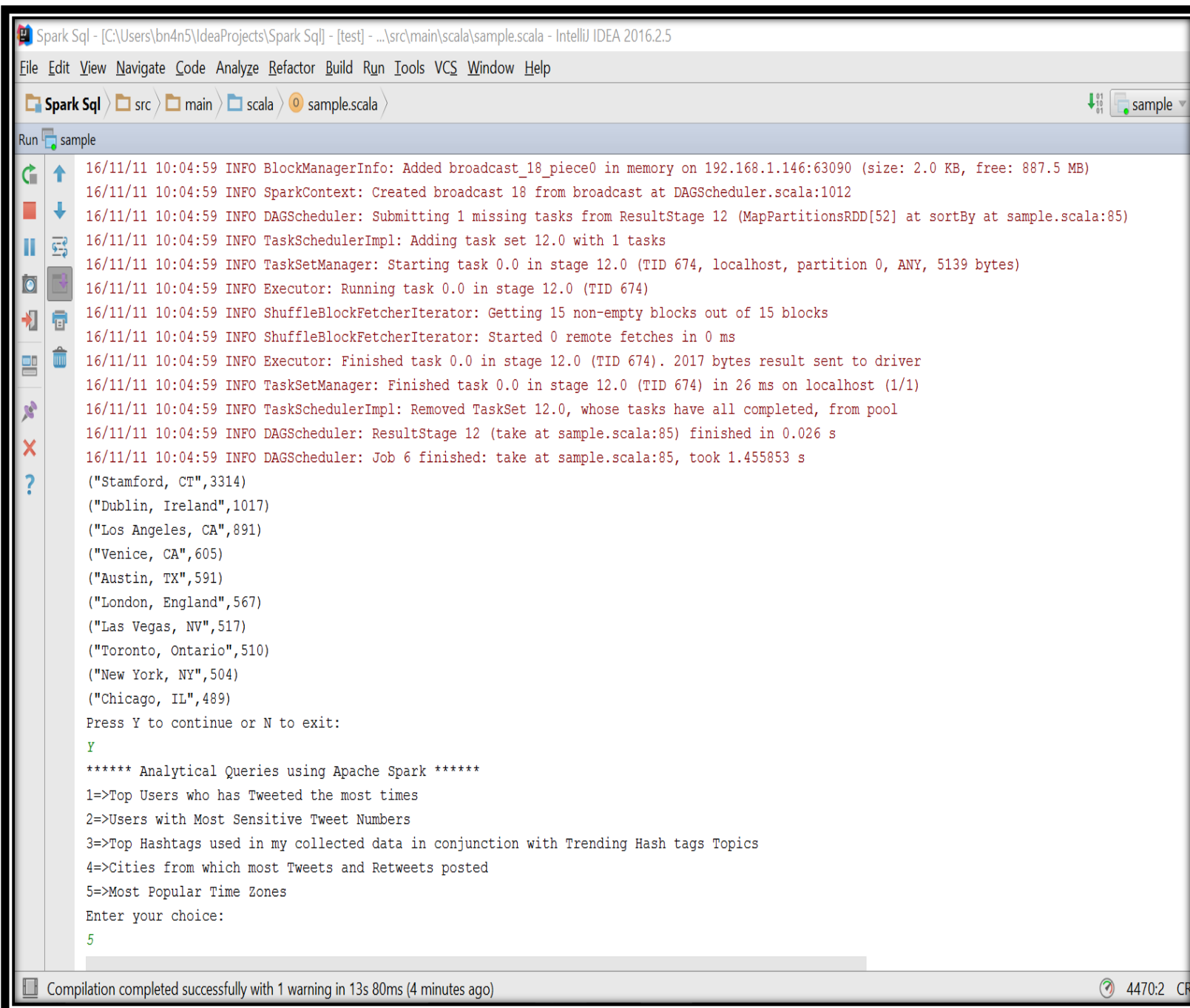
+-----+-----+
|      name|count|
+-----+-----+
|   #UFC205| 3928|
|   #UFC207| 1358|
|   #WWENXT|   78|
|   #UFC204|   25|
| #WWENoMercy|   6|
|   #UFC203|   4|
|  #WWEClash|   1|
+-----+-----+

Press Y to continue or N to exit:
Y

Compilation completed successfully with 1 warning in 13s 80ms (2 minutes ago)
```

## Query 4:

### Cities from which most Tweets and Retweets posted



```
Spark Sql - [C:\Users\bn4n5\IdeaProjects\Spark Sql] - [test] - ... \src\main\scala\sample.scala - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Spark Sql | src | main | scala | sample.scala

Run sample

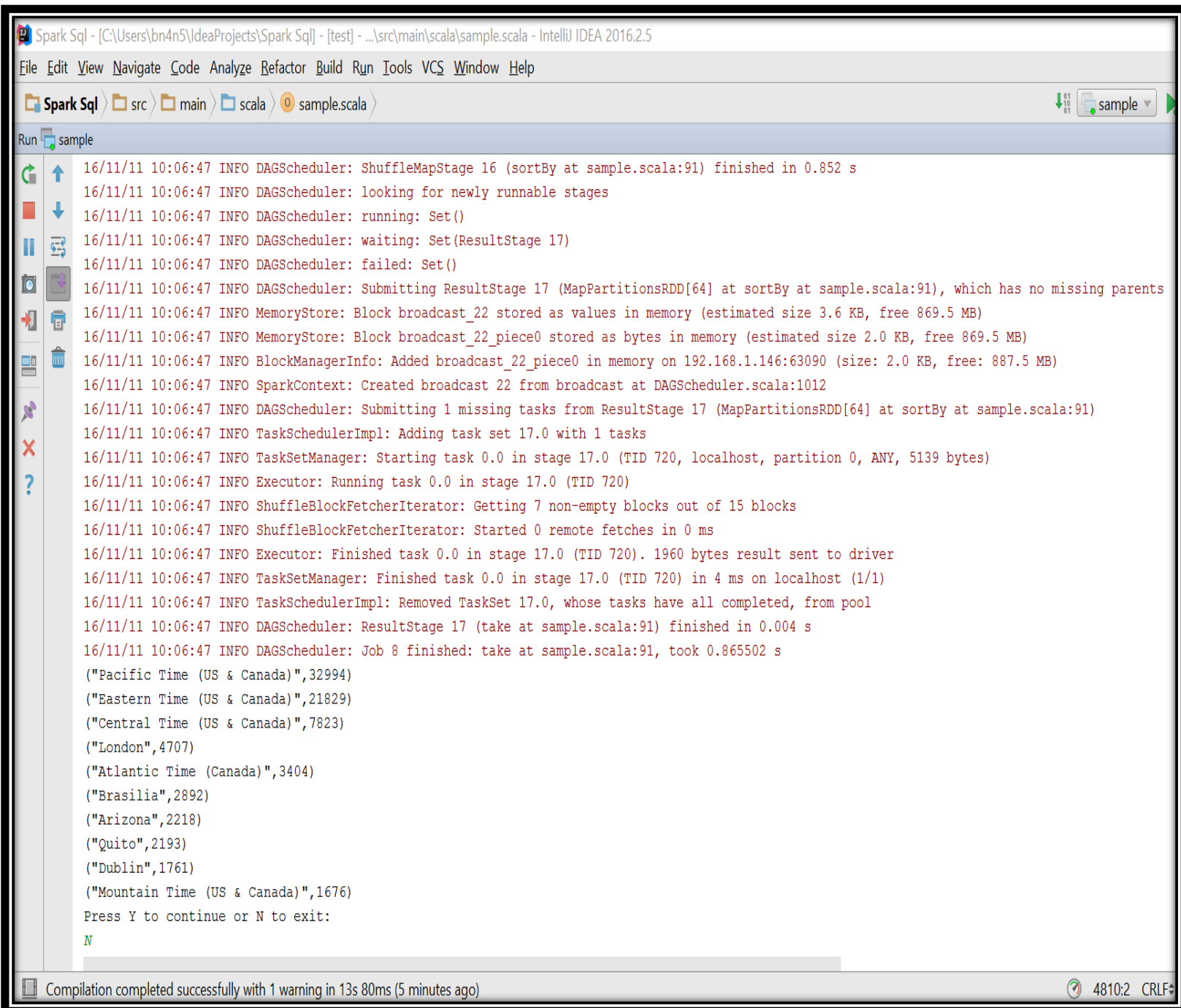
16/11/11 10:04:59 INFO BlockManagerInfo: Added broadcast_18_piece0 in memory on 192.168.1.146:63090 (size: 2.0 KB, free: 887.5 MB)
16/11/11 10:04:59 INFO SparkContext: Created broadcast 18 from broadcast at DAGScheduler.scala:1012
16/11/11 10:04:59 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 12 (MapPartitionsRDD[52] at sortBy at sample.scala:85)
16/11/11 10:04:59 INFO TaskSchedulerImpl: Adding task set 12.0 with 1 tasks
16/11/11 10:04:59 INFO TaskSetManager: Starting task 0.0 in stage 12.0 (TID 674, localhost, partition 0, ANY, 5139 bytes)
16/11/11 10:04:59 INFO Executor: Running task 0.0 in stage 12.0 (TID 674)
16/11/11 10:04:59 INFO ShuffleBlockFetcherIterator: Getting 15 non-empty blocks out of 15 blocks
16/11/11 10:04:59 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:04:59 INFO Executor: Finished task 0.0 in stage 12.0 (TID 674). 2017 bytes result sent to driver
16/11/11 10:04:59 INFO TaskSetManager: Finished task 0.0 in stage 12.0 (TID 674) in 26 ms on localhost (1/1)
16/11/11 10:04:59 INFO TaskSchedulerImpl: Removed TaskSet 12.0, whose tasks have all completed, from pool
16/11/11 10:04:59 INFO DAGScheduler: ResultStage 12 (take at sample.scala:85) finished in 0.026 s
16/11/11 10:04:59 INFO DAGScheduler: Job 6 finished: take at sample.scala:85, took 1.455853 s

("Stamford, CT",3314)
("Dublin, Ireland",1017)
("Los Angeles, CA",891)
("Venice, CA",605)
("Austin, TX",591)
("London, England",567)
("Las Vegas, NV",517)
("Toronto, Ontario",510)
("New York, NY",504)
("Chicago, IL",489)
Press Y to continue or N to exit:
Y
***** Analytical Queries using Apache Spark *****
1=>Top Users who has Tweeted the most times
2=>Users with Most Sensitive Tweet Numbers
3=>Top Hashtags used in my collected data in conjunction with Trending Hash tags Topics
4=>Cities from which most Tweets and Retweets posted
5=>Most Popular Time Zones
Enter your choice:
5

Compilation completed successfully with 1 warning in 13s 80ms (4 minutes ago) 4470:2 CF
```

## Query 5:

### Most Popular Time Zones



```
Spark Sql - [C:\Users\bn4n5\IdeaProjects\Spark Sql] - [test] - ...\src\main\scala\sample.scala - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Spark Sql > src > main > scala > sample.scala >

Run sample

16/11/11 10:06:47 INFO DAGScheduler: ShuffleMapStage 16 (sortBy at sample.scala:91) finished in 0.852 s
16/11/11 10:06:47 INFO DAGScheduler: looking for newly runnable stages
16/11/11 10:06:47 INFO DAGScheduler: running: Set()
16/11/11 10:06:47 INFO DAGScheduler: waiting: Set(ResultStage 17)
16/11/11 10:06:47 INFO DAGScheduler: failed: Set()
16/11/11 10:06:47 INFO DAGScheduler: Submitting ResultStage 17 (MapPartitionsRDD[64] at sortBy at sample.scala:91), which has no missing parents
16/11/11 10:06:47 INFO MemoryStore: Block broadcast_22 stored as values in memory (estimated size 3.6 KB, free 869.5 MB)
16/11/11 10:06:47 INFO MemoryStore: Block broadcast_22_piece0 stored as bytes in memory (estimated size 2.0 KB, free 869.5 MB)
16/11/11 10:06:47 INFO BlockManagerInfo: Added broadcast_22_piece0 in memory on 192.168.1.146:63090 (size: 2.0 KB, free: 887.5 MB)
16/11/11 10:06:47 INFO SparkContext: Created broadcast 22 from broadcast at DAGScheduler.scala:1012
16/11/11 10:06:47 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 17 (MapPartitionsRDD[64] at sortBy at sample.scala:91)
16/11/11 10:06:47 INFO TaskSchedulerImpl: Adding task set 17.0 with 1 tasks
16/11/11 10:06:47 INFO TaskSetManager: Starting task 0.0 in stage 17.0 (TID 720, localhost, partition 0, ANY, 5139 bytes)
16/11/11 10:06:47 INFO Executor: Running task 0.0 in stage 17.0 (TID 720)
16/11/11 10:06:47 INFO ShuffleBlockFetcherIterator: Getting 7 non-empty blocks out of 15 blocks
16/11/11 10:06:47 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/11/11 10:06:47 INFO Executor: Finished task 0.0 in stage 17.0 (TID 720). 1960 bytes result sent to driver
16/11/11 10:06:47 INFO TaskSetManager: Finished task 0.0 in stage 17.0 (TID 720) in 4 ms on localhost (1/1)
16/11/11 10:06:47 INFO TaskSchedulerImpl: Removed TaskSet 17.0, whose tasks have all completed, from pool
16/11/11 10:06:47 INFO DAGScheduler: ResultStage 17 (take at sample.scala:91) finished in 0.004 s
16/11/11 10:06:47 INFO DAGScheduler: Job 8 finished: take at sample.scala:91, took 0.865502 s

("Pacific Time (US & Canada)",32994)
("Eastern Time (US & Canada)",21829)
("Central Time (US & Canada)",7823)
("London",4707)
("Atlantic Time (Canada)",3404)
("Brasilia",2892)
("Arizona",2218)
("Quito",2193)
("Dublin",1761)
("Mountain Time (US & Canada)",1676)
Press Y to continue or N to exit:
N

Compilation completed successfully with 1 warning in 13s 80ms (5 minutes ago) 4810:2 CRLF
```