**Principles of Big Data Management**

**Phase-2 Report**

**Team Members:**

1. Rahul Dhamerla
2. Teja Devarapalli
3. Nikita Goyal

**Phase-2: Analyzing and Visualizing of twitter data**

**Applications:**

* Apache Spark SQL
* Python
* IntelliJ Idea IDE
* Twitter Developer Account

**Collecting tweets from Twitter:**

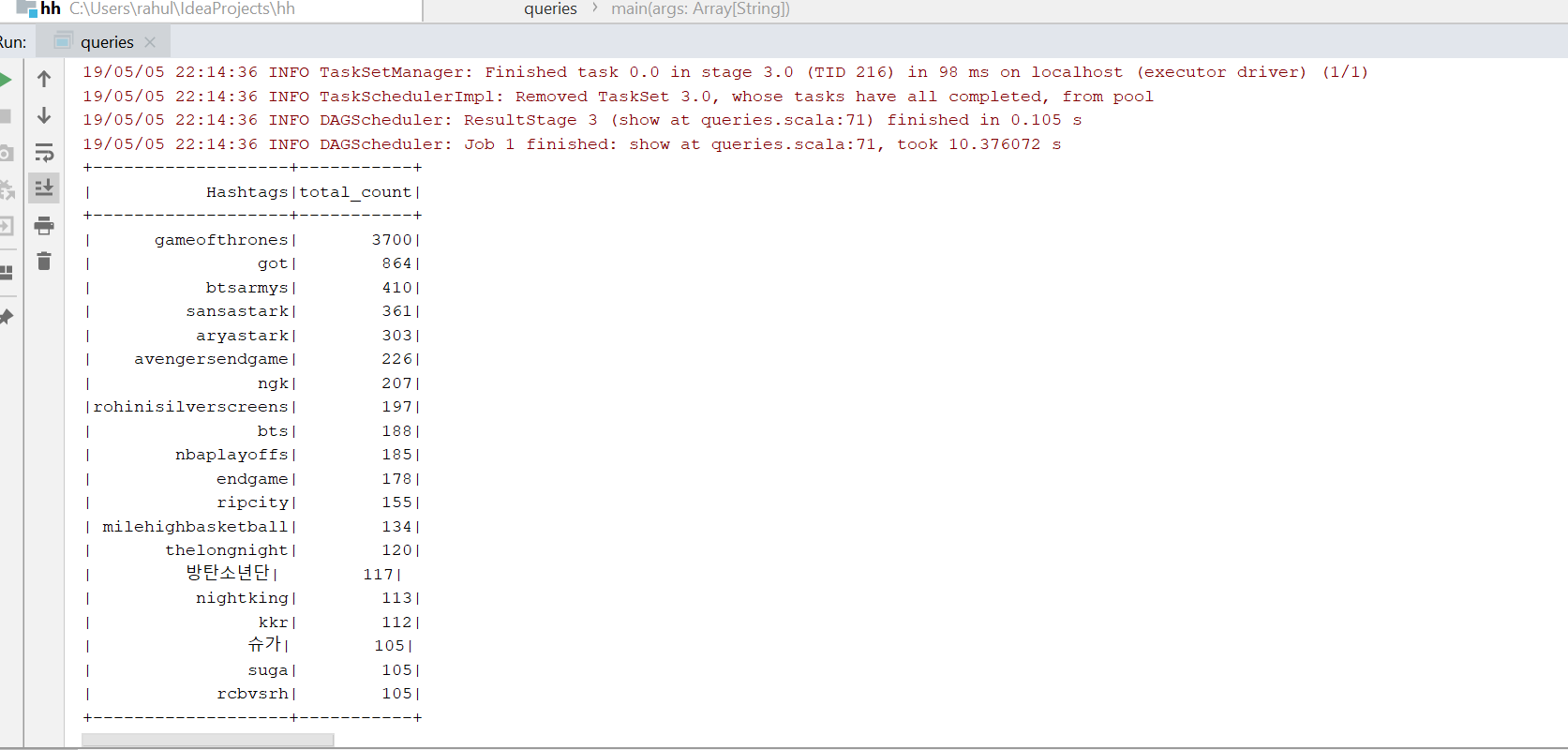
* Firstly, we have created a developer account in Twitter using below link. https://apps.twitter.com/
* Below are the variables that contains the user credentials to access Twitter API.
* access\_token = "1089997729219178496-BvhcwdDw6eGPbPJM5wJYqOvsq7rE7G"
* access\_token\_secret = "xZu9ILsesdPiRi5quCxx94mQTfrj00sMEedGrddvYOmsQ"
* consumer\_key = "f4qQNYSFaGxO4iB6SamtIZarf"
* consumer\_secret = "PTuSFEtZAeRP9nov16hJTWQzFNedKTSDKR9ZSS1Iu65JiSbXvO"
* We have written python program that is used to fetch tweets in JSON format. (Tweetscollection.py)
* The tweet data is collected on the concept based on to analyze and visualize the data regarding various characters in Game of Thrones a TV Show.
* We have written 10 analytic queries and performed visualization on them .

**Queries and Output:**

**Query-1: This query fetches the tweets on hashtags**

**val** hashtag = sqlContext.sql(**"SELECT LOWER(hashtags.text) As Hashtags, COUNT(\*) AS total\_count FROM tweets LATERAL VIEW EXPLODE(entities.hashtags) t1 AS hashtags GROUP BY LOWER(hashtags.text) ORDER BY total\_count DESC LIMIT 20"**)  
  
hashtag.show()

**Output:**

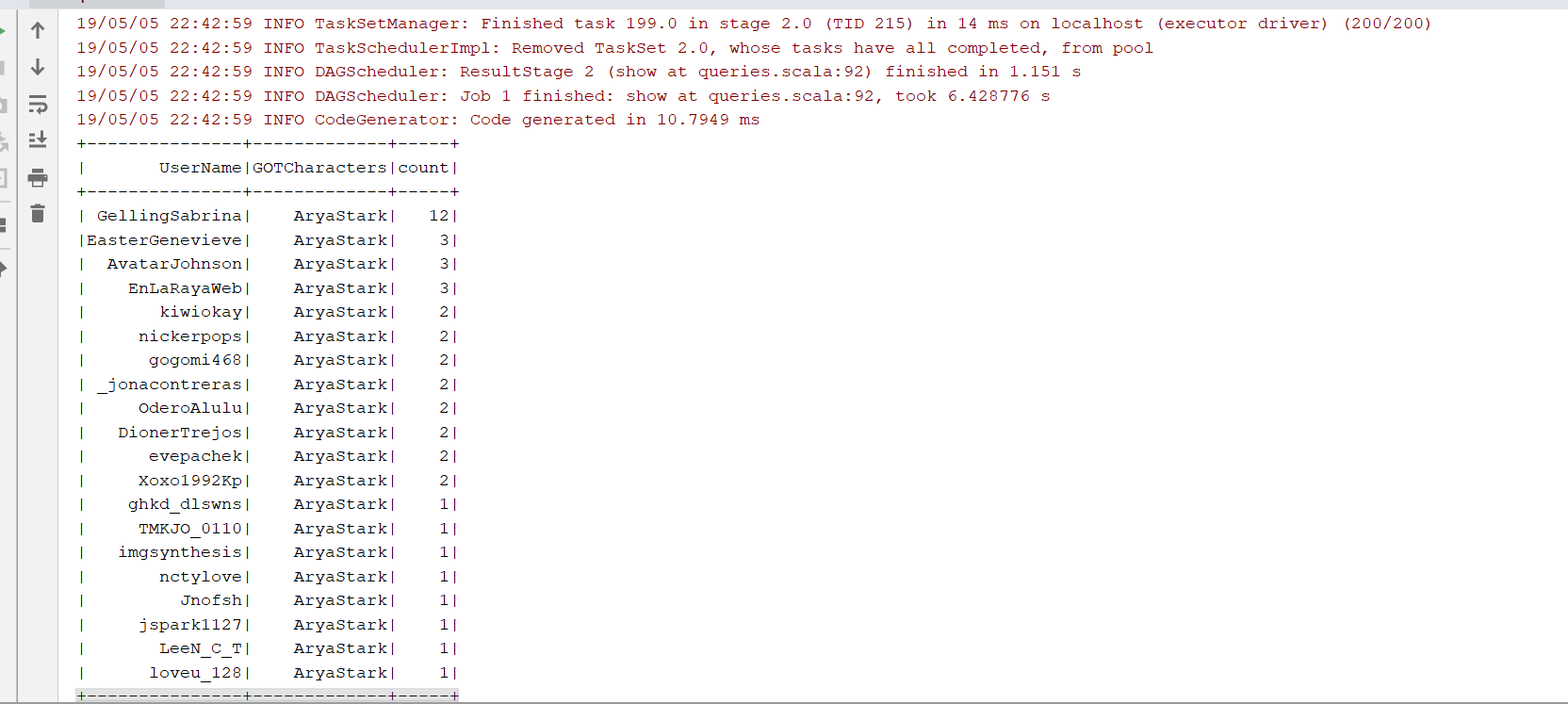


**Visualization:**

**Query-2:****Which user tweeted most about which GOTCharacters-**

**val** r1 = sqlContext.sql(**"SELECT UserName,GOTCharacters,count(\*) as count FROM disCat2 WHERE GOTCharacters in ('AryaStark') group by UserName,GOTCharacters order by count desc"**)

**Output:**

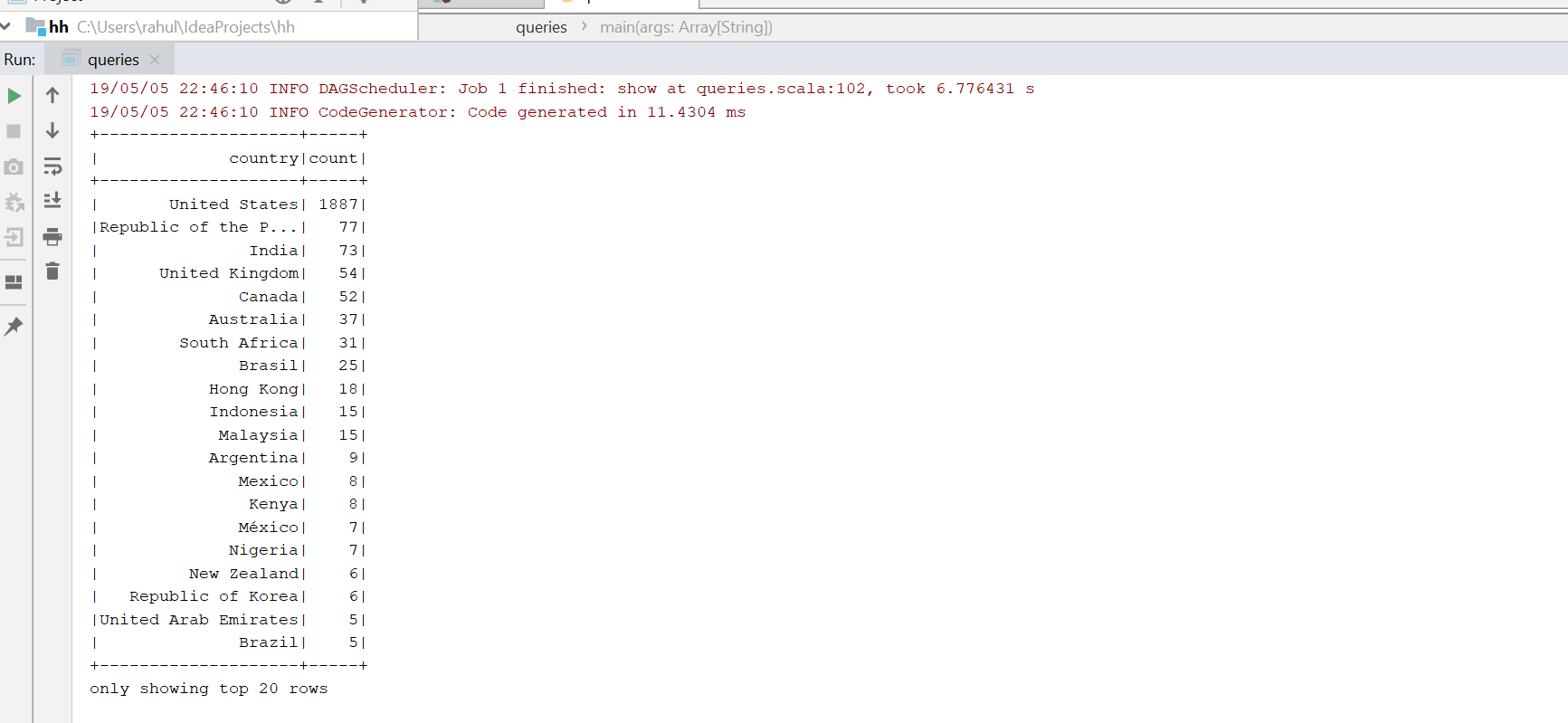


**Visualization:**

**Query-3: Tweets from different countries about GOT:**

**val** countrytweetscount=sqlContext.sql(**"SELECT distinct place.country, count(\*) as count FROM tweets where place.country is not null "** + **"GROUP BY place.country ORDER BY count DESC"**)  
countrytweetscount.createOrReplaceTempView(**"countrytweetscount"**)

**Output:**

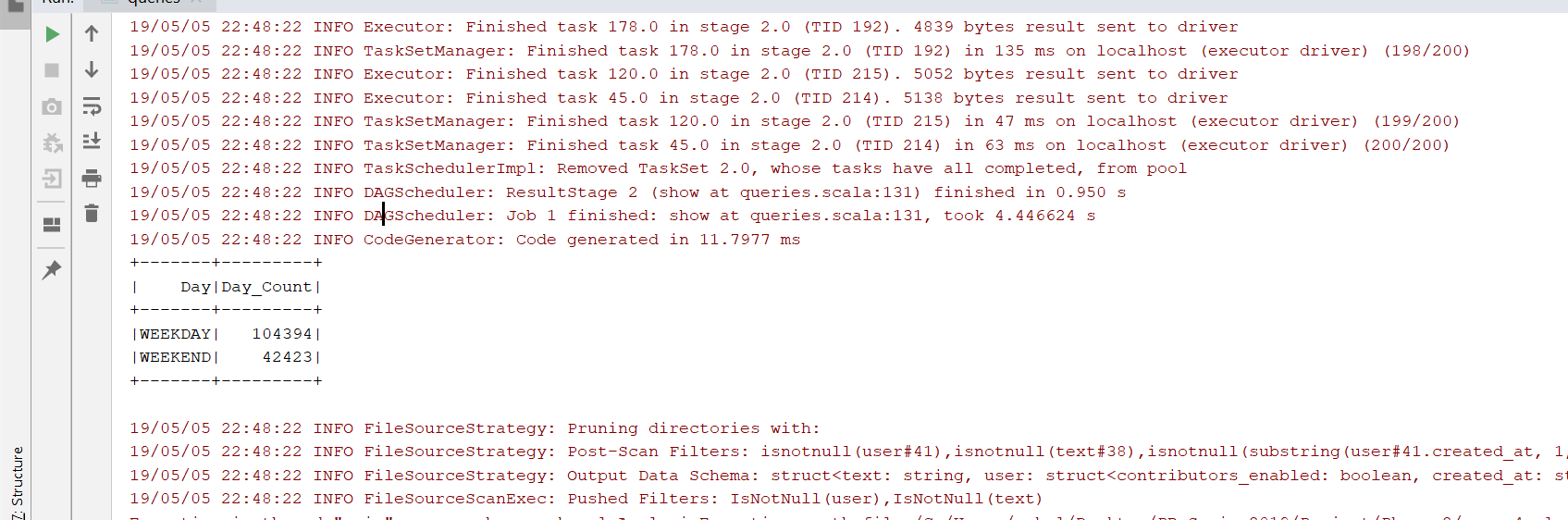


**Visualization:**

**Query-4: Tweets count on different days.**

**val** day\_data = sqlContext.sql(**"SELECT substring(user.created\_at,1,3) as day from tweets where text is not null"**)  
  
day\_data.createOrReplaceTempView(**"day\_data"**)  
  
**val** days\_final = sqlContext.sql(  
 **""" SELECT Case  
 |when day LIKE '%Mon%' then 'WEEKDAY'  
 |when day LIKE '%Tue%' then 'WEEKDAY'  
 |when day LIKE '%Wed%' then 'WEEKDAY'  
 |when day LIKE '%Thu%' then 'WEEKDAY'  
 |when day LIKE '%Fri%' then 'WEEKDAY'  
 |when day LIKE '%Sat%' then 'WEEKEND'  
 |when day LIKE '%Sun%' then 'WEEKEND'  
 | else  
 | null  
 | end as day1 from day\_data where day is not null"""**.stripMargin)  
  
days\_final.createOrReplaceTempView(**"days\_final"**)  
  
**val** res = sqlContext.sql(**"SELECT day1 as Day,Count(\*) as Day\_Count from days\_final where day1 is not null group by day1 order by count(\*) desc"**)

**Output:**

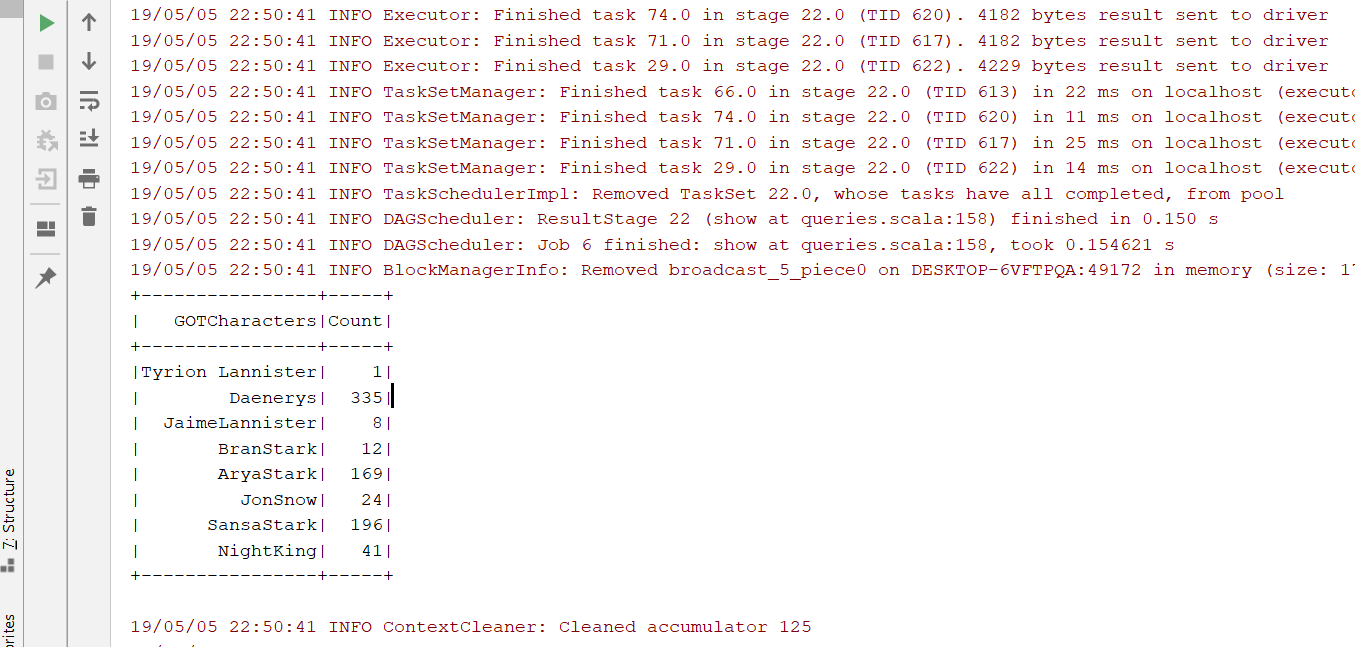


**Visualization:**

**Query-5: Tweets count for different types of Characters**

**val** r1 = sqlContext.sql(**"select loc,GotCharacters,count(\*) as count from disCat2 "** +  
 **"group by loc,GotCharacters order by count desc"**)

**Output:**

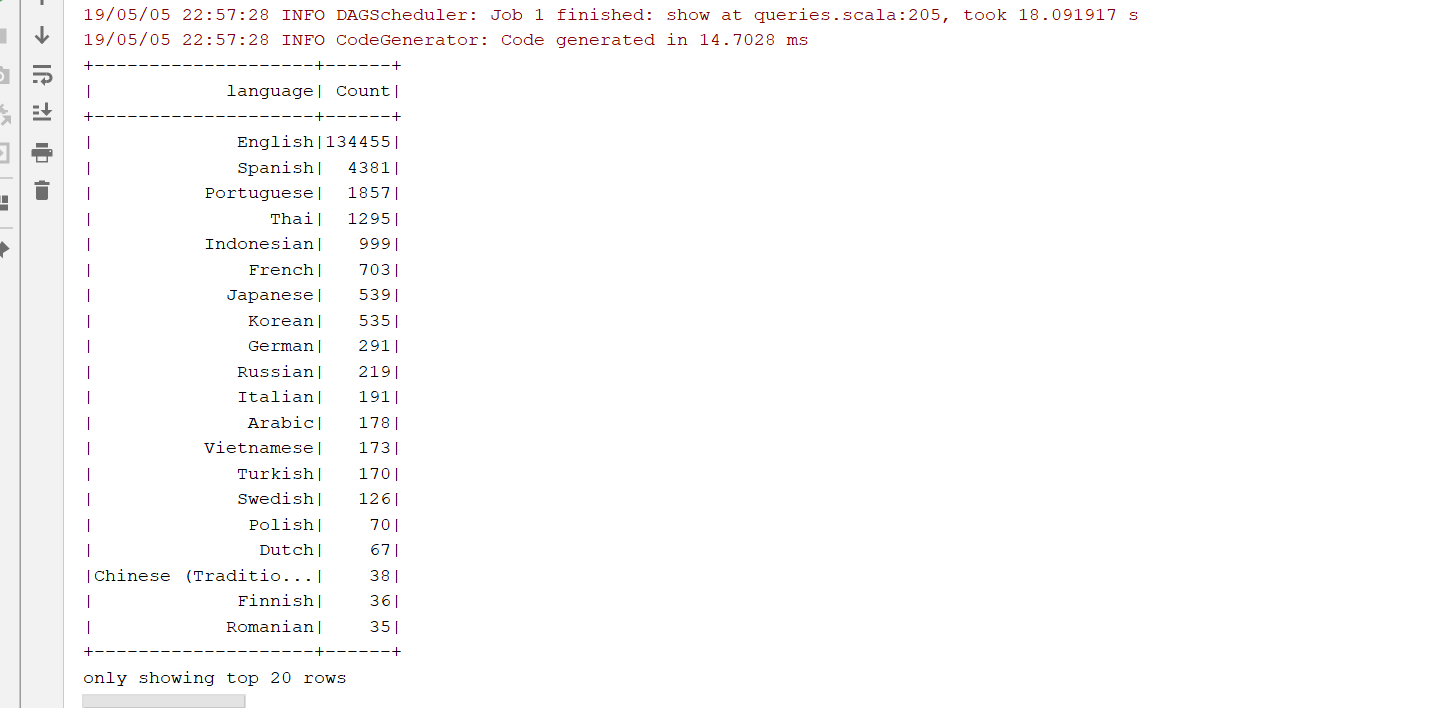


**Visualization:**

**Query-6: Popular languages used for tweets about GOT**

**val** langWstCount = sqlContext.sql(**"SELECT distinct id,"** +  
 **"CASE when user.lang LIKE '%en%' then 'English'"**+  
 **"when user.lang LIKE '%ja%' then 'Japanese'"**+  
 **"when user.lang LIKE '%es%' then 'Spanish'"**+  
 **"when user.lang LIKE '%fr%' then 'French'"**+  
 **"when user.lang LIKE '%vi%' then 'Vietnamese'"**+  
 **"when user.lang LIKE '%zh-cn%' then 'Chinese (Simplified)'"**+  
 **"when user.lang LIKE '%zh-tw%' then 'Chinese (Traditional)'"**+  
 **"END AS language from tweets where text is not null"**)  
langWstCount.createOrReplaceTempView(**"langWstCount"**)  
**var** langWstDataCount=sqlContext.sql(**"SELECT language, Count(language) as Count from langWstCount where id is NOT NULL and language is not null group by language order by Count DESC"**)

**Output:**

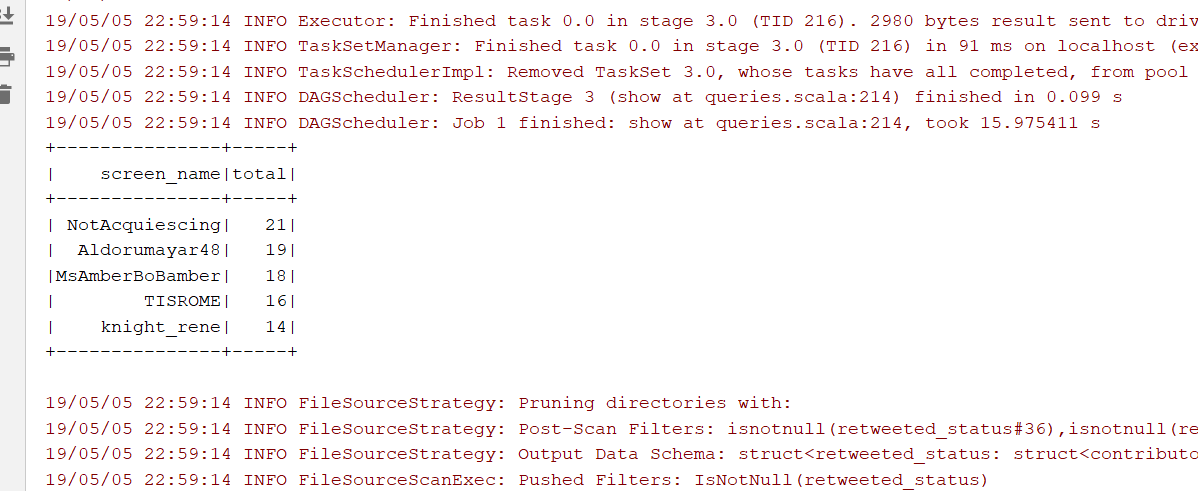


**Visualization:**

**Query-7: Retweet Count**

**val** retweetcount=sqlContext.sql(**"SELECT user.screen\_name, COUNT(\*) as total FROM tweets WHERE retweeted\_status.user is not null GROUP BY user.screen\_name ORDER BY total desc LIMIT 5"**)  
retweetcount.createOrReplaceTempView(**"retweetcount"**)

**Output:**

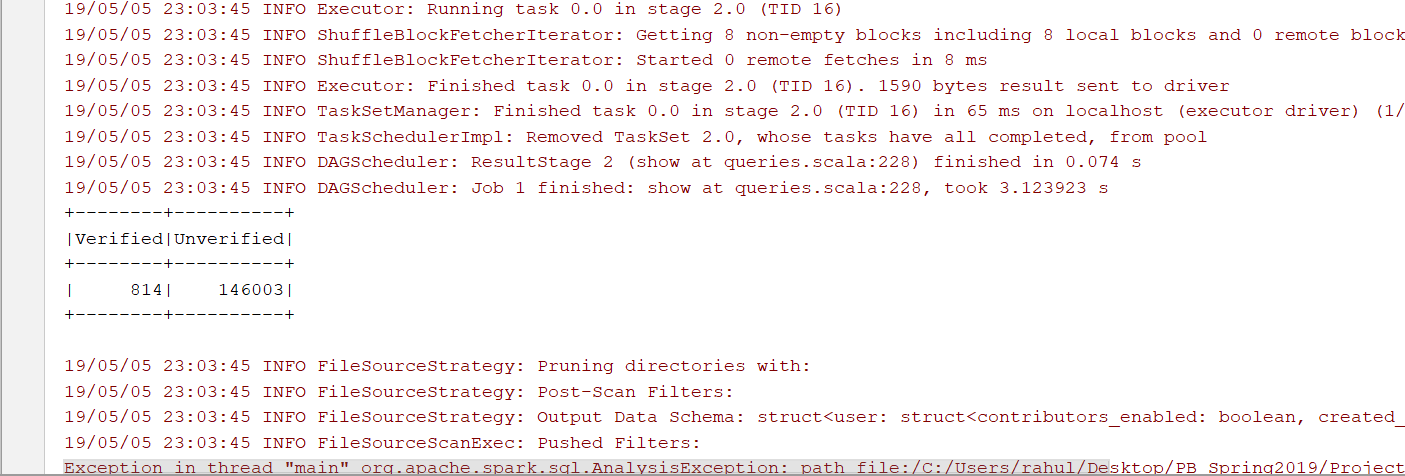


**Visualization:**

**Query-8: Account Verification**

**var** acctVerifydata=sqlContext.sql(**"select sum(case when user.verified = true then 1 else 0 end)Verified,sum(case when user.verified = false then 1 else 0 end)Unverified from tweets"**)

**Output:**

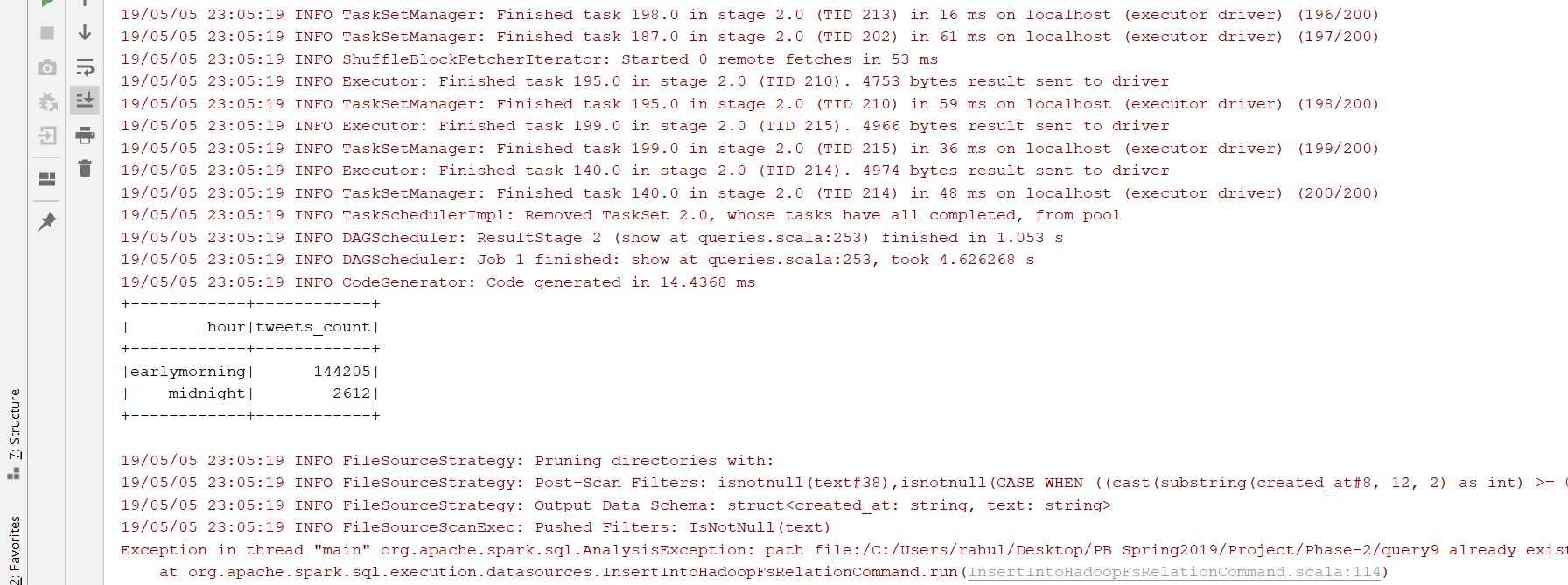


**Visualization:**

**Query-9: On which hours tweets flow is high**

**val** timehour = sqlContext.sql(**"SELECT SUBSTRING(created\_at,12,2) as hour from tweets where text is not null"**)  
  
timehour.createOrReplaceTempView(**"timehour"**)

**Output:**

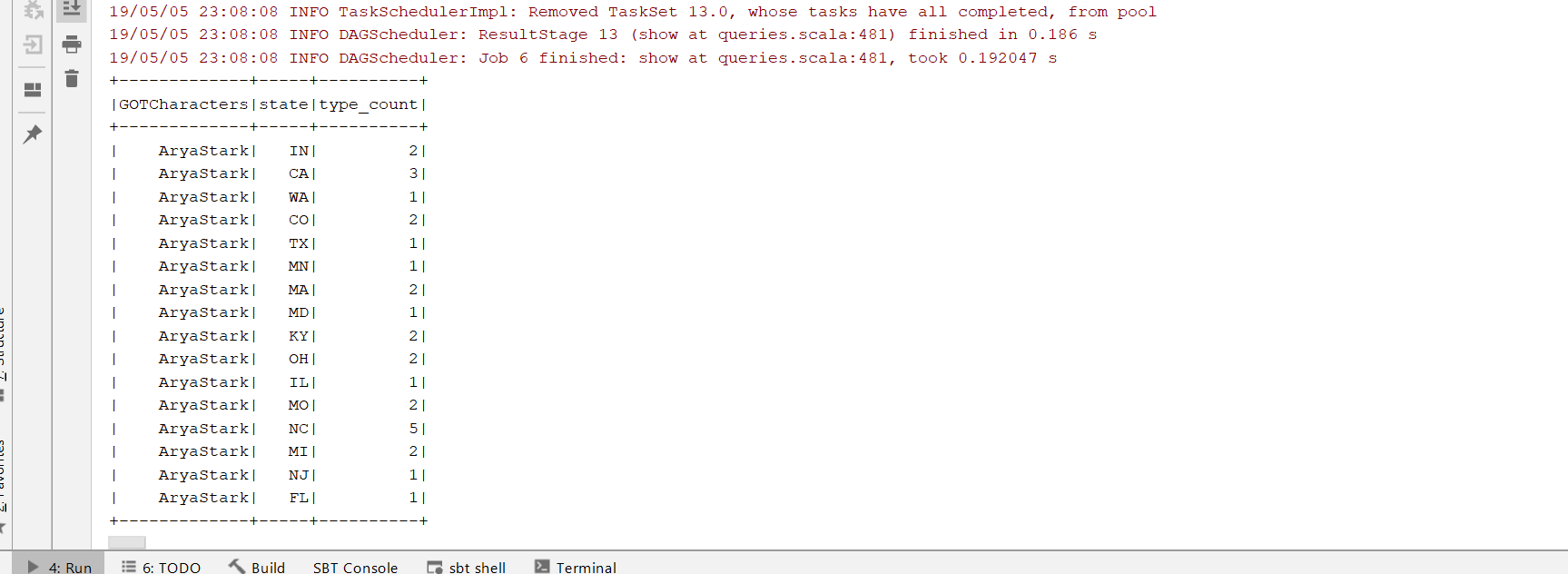


**Visualization:**

**Query-10: Tweets from different states about a particular GOT character**

**val** AryaStarkRDD = sqlContext.sql(**""" SELECT 'AryaStark' as GOTCharacters, user.location as loc from tweets where text LIKE '%#AryaStark%' """**)

**Output:**



**Visualization:**