Big Data Hadoop Stack Exchange project

GROUP B

Name: SHUBHAM KHULE EMP ID: 9188

Setting up SparkContext & importing necessary packages

val sqlContext = new org.apache.spark.sql.SQLContext(sc);
import sqlContext.implicits._
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
import org.apache.spark.sql.Column
import org.apache.spark.sql.types._
import org.apache.spark.sql.types._
import org.apache.spark.sql.types.{IntegerType, StringType, StructField, StructType,LongType};

Creation of Schema

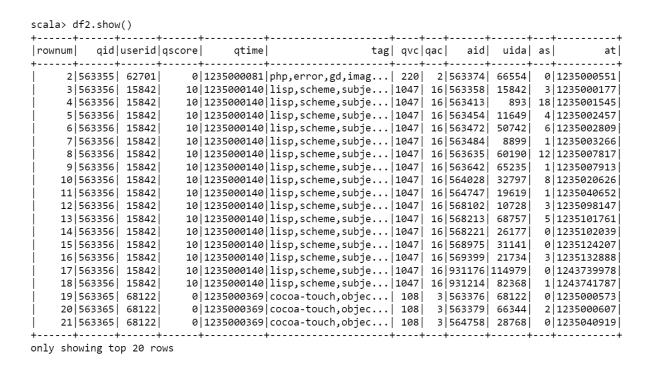
val schema =

StructType(Array(StructField("rownum",IntegerType,true),StructField("qid",IntegerType,true),StructField("qscore",IntegerType,true),StructField("qtime",LongType,true),StructField("tag",StringType,true),StructField("qvc",IntegerType,true),StructField("qac",IntegerType,true),StructField("aid",IntegerType,true),StructField("uida",IntegerType,true),StructField("aid",IntegerType,true),StructField("aid",IntegerType,true)))

Loading of Data

val df2 =sqlContext.read.format("com.databricks.spark.csv").schema(schema).option("header", "true").option("delimiter", "|").load("/user/maria_dev/data/project.txt");

df2.show()



Q1: Top 10 most commonly used tags in this data set.

df2.select(explode(split(col("tag"),","))).groupBy("col").count().orderBy(col("count").desc).show(1 0)

Q2. Average time to answer questions.

df2.registerTempTable("social")

val sql1=spark.sql("select avg(timetaken)/3600 from (select (at - qtime) as timetaken from social)")

Q3. Number of questions which got answered within 1 hour

val sql2=spark.sql("select count(qid) from (select (at - qtime) as timetaken,* from social) where timetaken <= 3600")

```
scala> sql2.show()
+-----
|count(qid)|
+-----
```

174608

sql2.show()

Q4. Tags of questions which got answered within 1 hour.

val sql3 = spark.sql("select tag from (select (at - qtime) as timetaken,* from social) where timetaken <= 3600")

```
scala> sql3.show(false)
ltag
+-----
|php,error,gd,image-processing
lisp, scheme, subjective, clojure
lisp, scheme, subjective, clojure
lisp,scheme,subjective,clojure
lisp, scheme, subjective, clojure
lisp,scheme,subjective,clojure
cocoa-touch,objective-c,design-patterns
cocoa-touch, objective-c, design-patterns
core-animation
aspûnet
|jquery,javascript,dom
|winforms,gridview,ûnet
python, http
python, http
python, http
python, http
python, http
|visualstudio
|visualstudio
visualstudio
only showing top 20 rows
```