**NAME - VIKRAM SAHAI SAXENA**

COURSE - BIG DATA AND HADOOP DEVELOPER

REGISTERED BATCH - Jun 13 Mon - Jun 28 Tue, 20:00 - 23:00 (IST)

**Big Data Hadoop—Real World Project—Social Media**

**Problem Statement :**

Analyze data set from Stack Exchange

As part of a recruiting exercise of the biggest social media company, they asked candidates to analyze data set from Stack Exchange. We will be using similar data set to arrive at certain key insights.

Download the data set from the following link:

http://www.ics.uci.edu/~duboisc/stackoverflow/answers.csv

The data set contains the following attributes:

qid: Unique question id

i: User id of questioner

qs: Score of the question

qt: Time of the question (in epoch time)

tags: a comma-separated list of the tags associated with the question. Examples of

tags are ``html'', ``R'', ``mysql'', ``python'', and so on; often between two and six tags are

used on each question.

qvc: Number of views of this question (at the time of the datadump)

qac: Number of answers for this question (at the time of the datadump)

aid: Unique answer id

j: User id of answerer

as: Score of the answer

at: Time of the answer (in epoch time)

We need to arrive at following results:

- Top 10 most commonly used tags in this data set.

- Average time to answer questions.

- Number of questions which got answered within 1 hour.

- Tags of questions which got answered within 1 hour.

**Solution: (Using Pig)**

**1. Top 10 most commonly used tags in this data set.**

Code:

data = load 'answers.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',','YES\_MULTILINE');

tokens = foreach data generate FLATTEN(TOKENIZE($5));

grp = group tokens by $0;

cnt = foreach grp generate $0, COUNT($1);

ord = order cnt by $1 desc;

li = limit ord 10;

dump li;

Approach:

1. Load the data set 'answers.csv'.

2. Generate individual tuples of all tags.

3. Group tuples by tag.

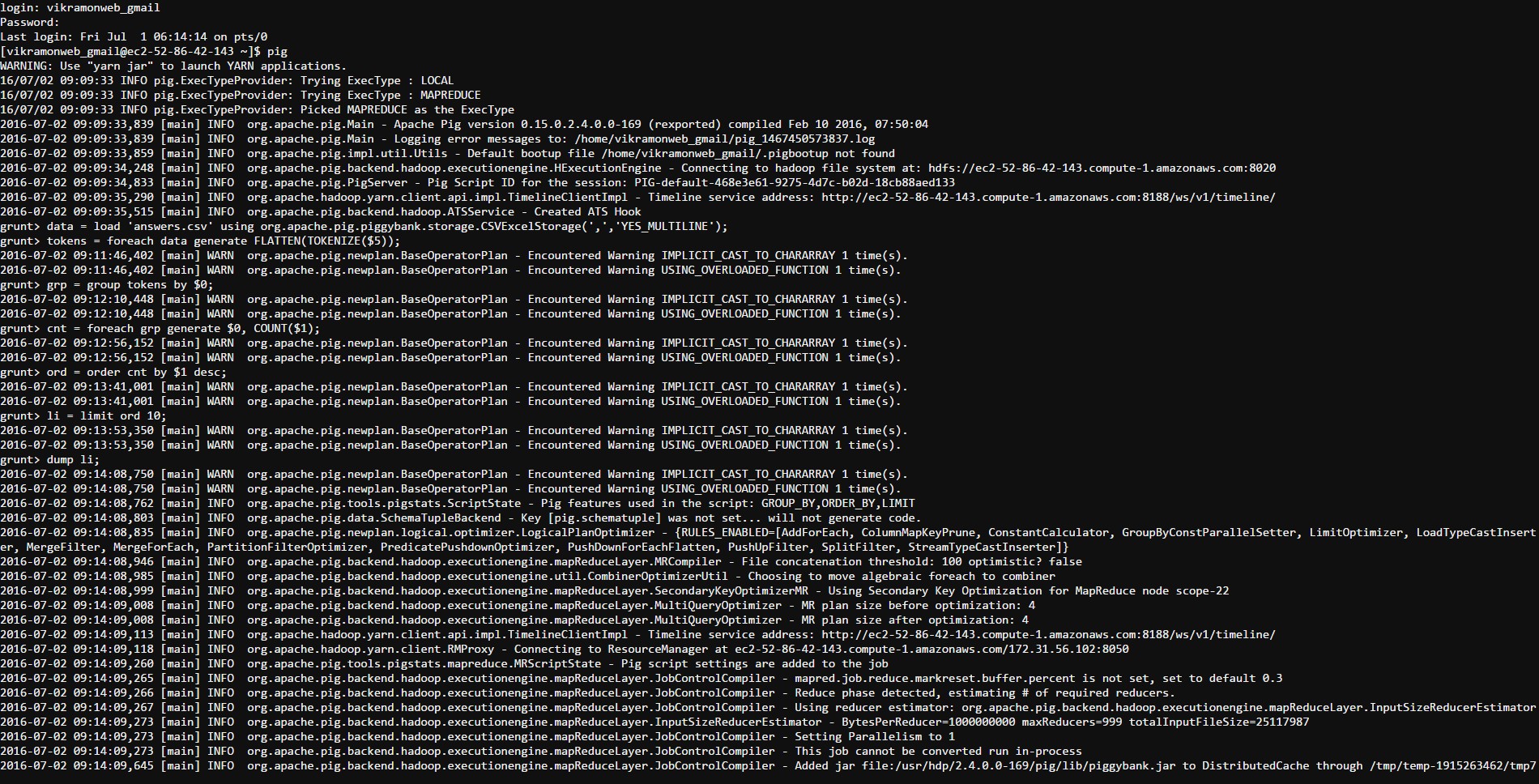
4. Generate tag along with its count of occurrence.

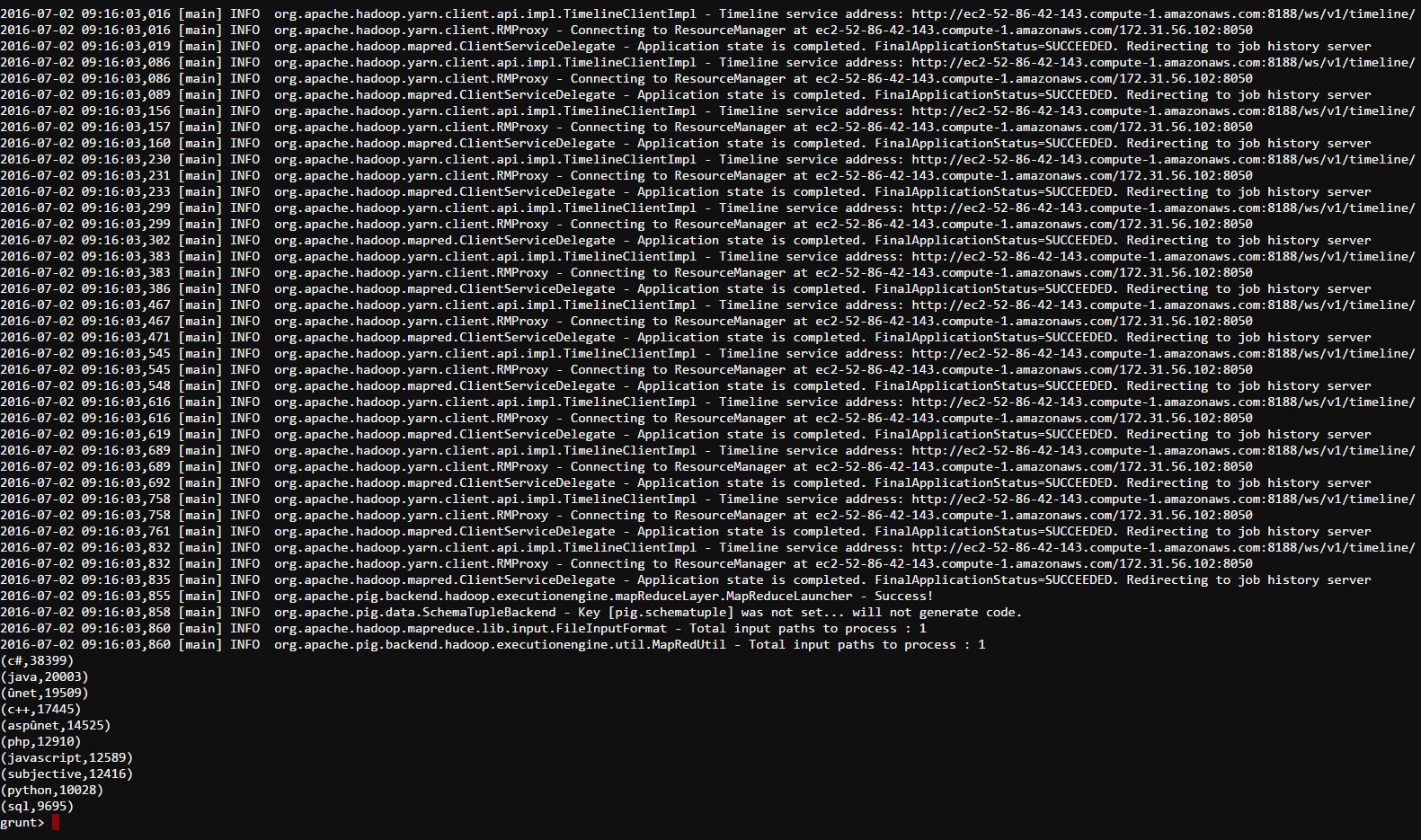
5. Order tags in descending order, ordered by count of occurrence.

6. Limit the order to 10 results.

7. Display the top 10 most commonly used tags in the data set.

Screenshots:





**2. Average time to answer questions.**

Code:

time = foreach data generate $1 as qid, $4 as qt, $11 as at;

grp\_data = group time by qid;

atime = foreach grp\_data generate $0 as qid, ((long)AVG(time.at))-((long)AVG(time.qt)) as avgtime;

dump atime;

Approach:

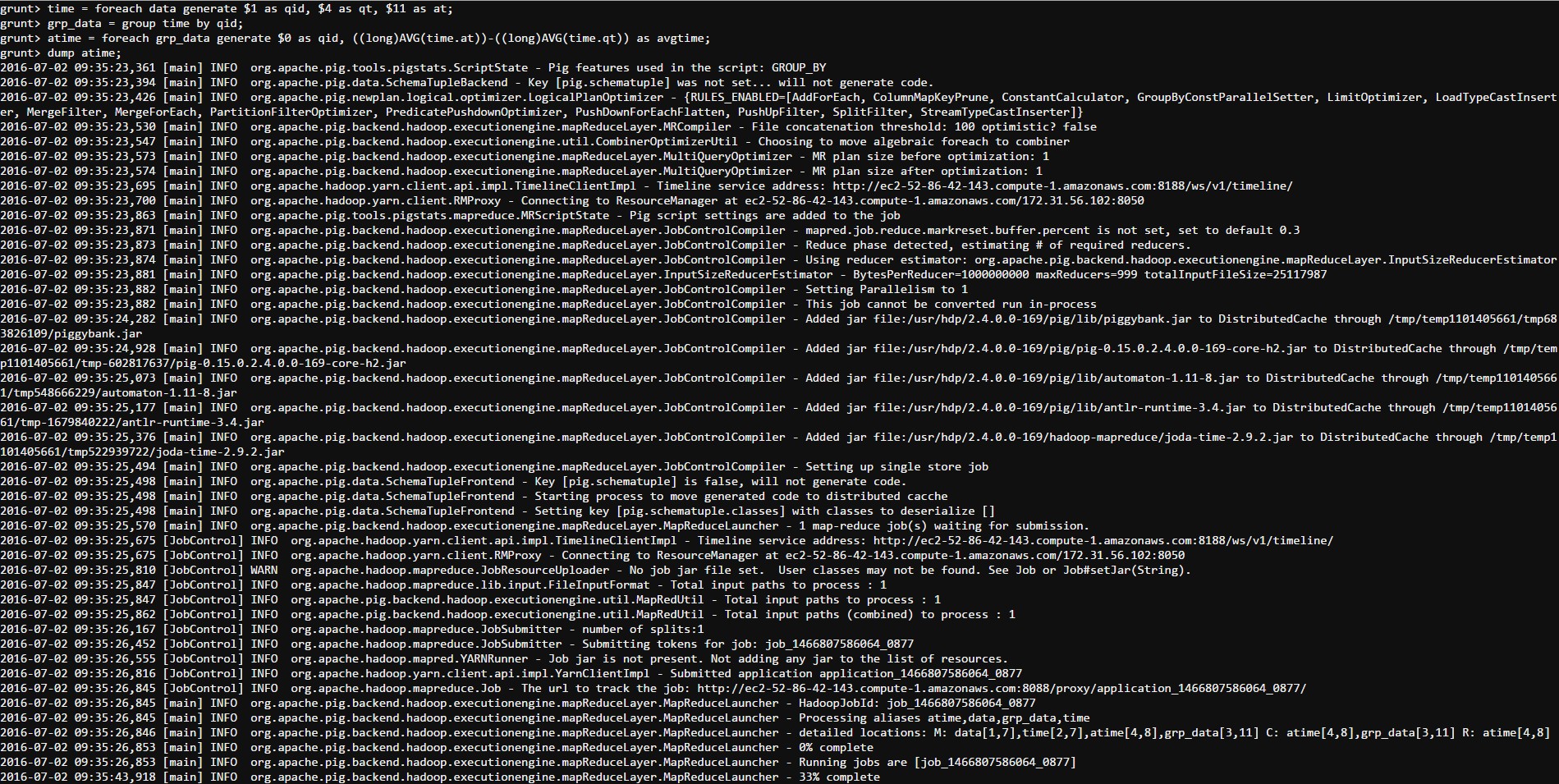
1. Generate columns qid, qt and at.

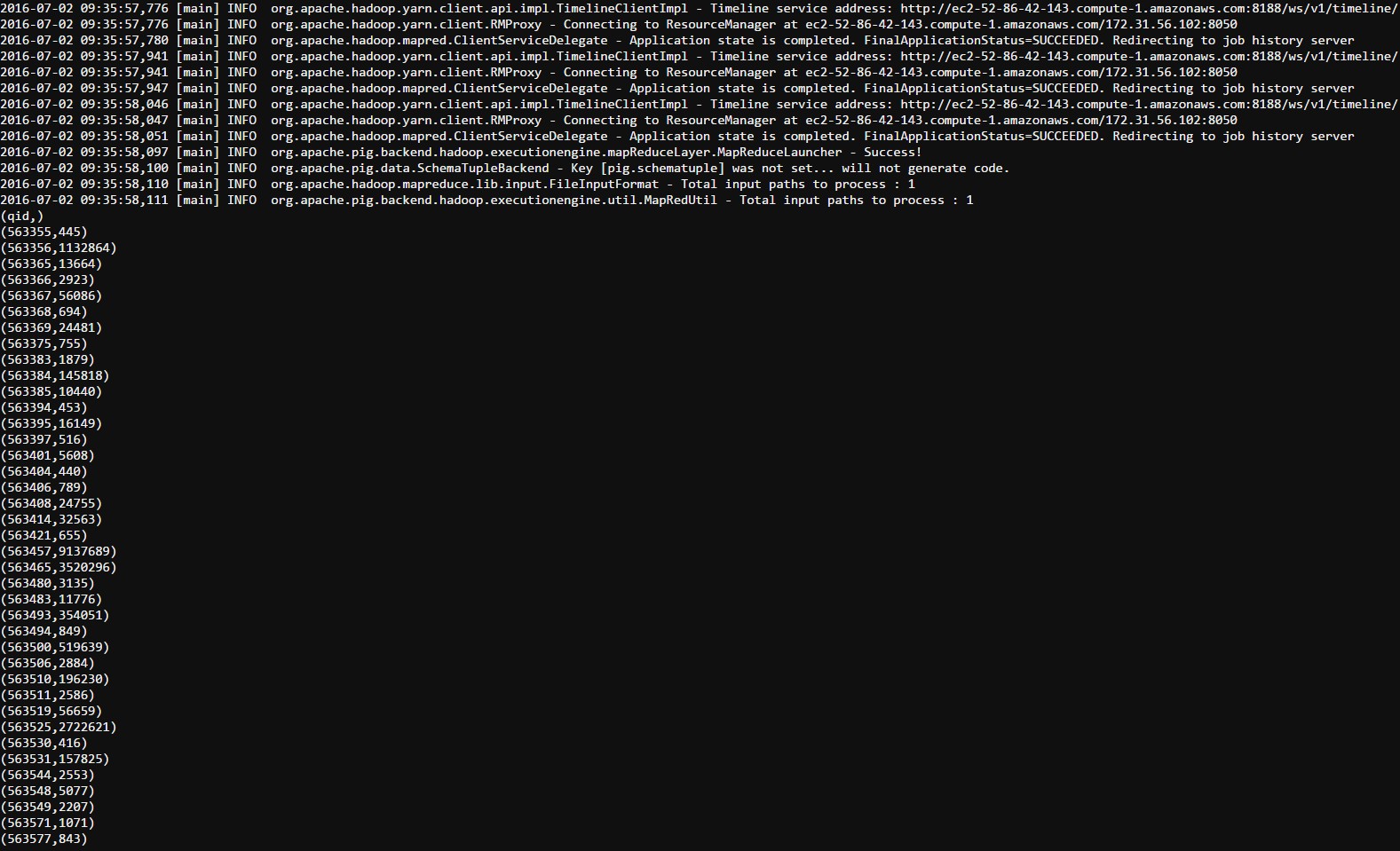
2. Group the column data by qid.

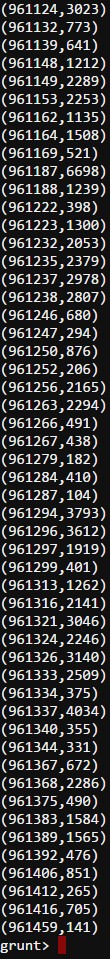
3. For each tuple of grouped data, generate qid along with the average time (in seconds) to answer the question.

4. Display the average time to answer questions.

Screenshots:





****

**3. Number of questions which got answered within 1 hour.**

Code:

time2 = foreach time generate $0 as qid, (long)(at-qt) as dt;

time3 = foreach time2 generate $0 as qid, ((long)dt/3600) as ht;

time4 = filter time3 by ht<=1;

time5 = DISTINCT time4;

gr = group time5 all;

count = foreach gr generate COUNT(time5.qid) as count;

dump count;

Approach:

1. For each tuple of time, generate qid and the time taken to answer a question (in seconds).

2. For each tuple of time2, generate qid and the time taken to answer a question (in hours).

3. Filter all tuples by time taken in hours <= 1 hour.

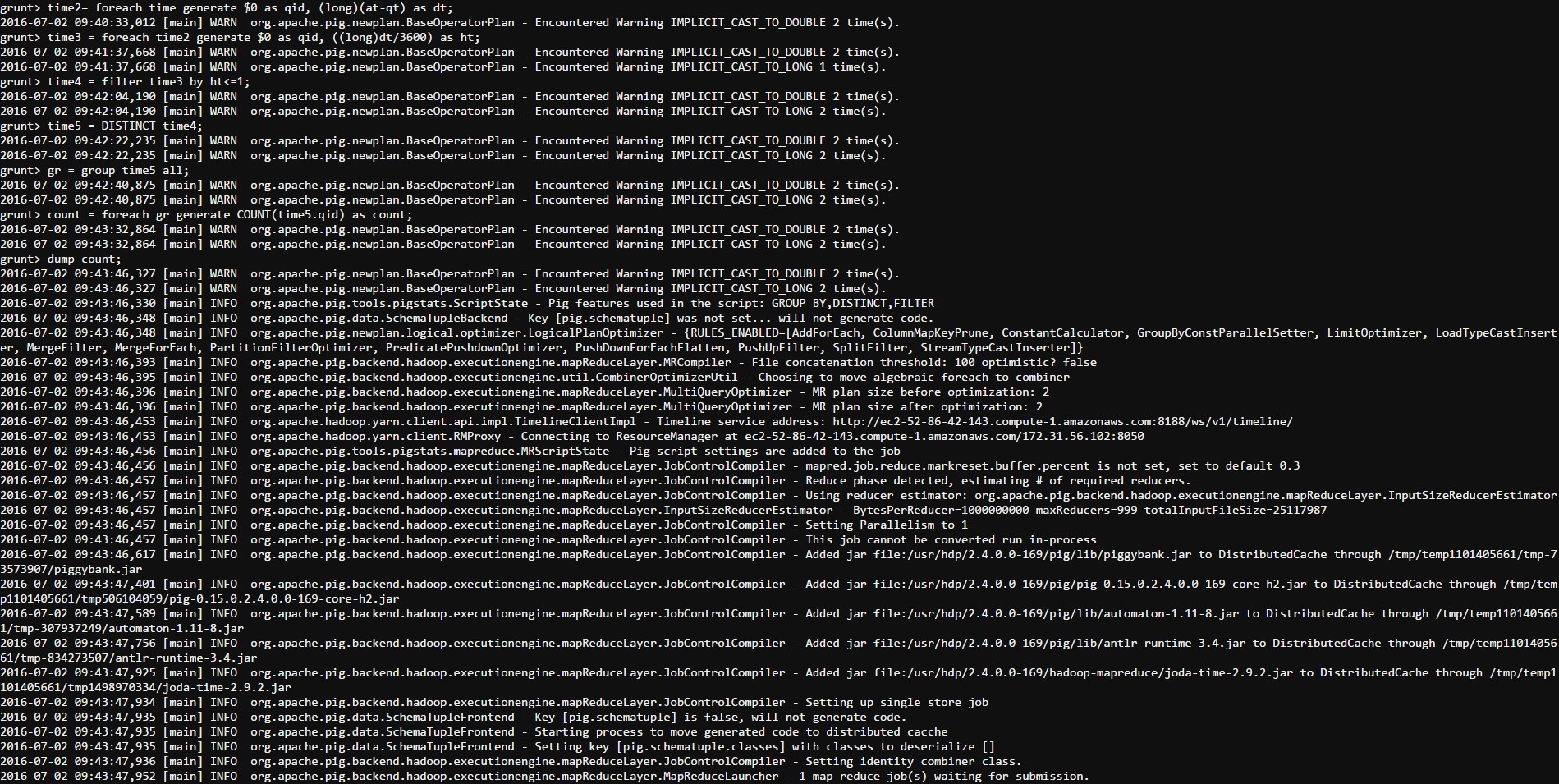
4. Remove duplicate tuples.

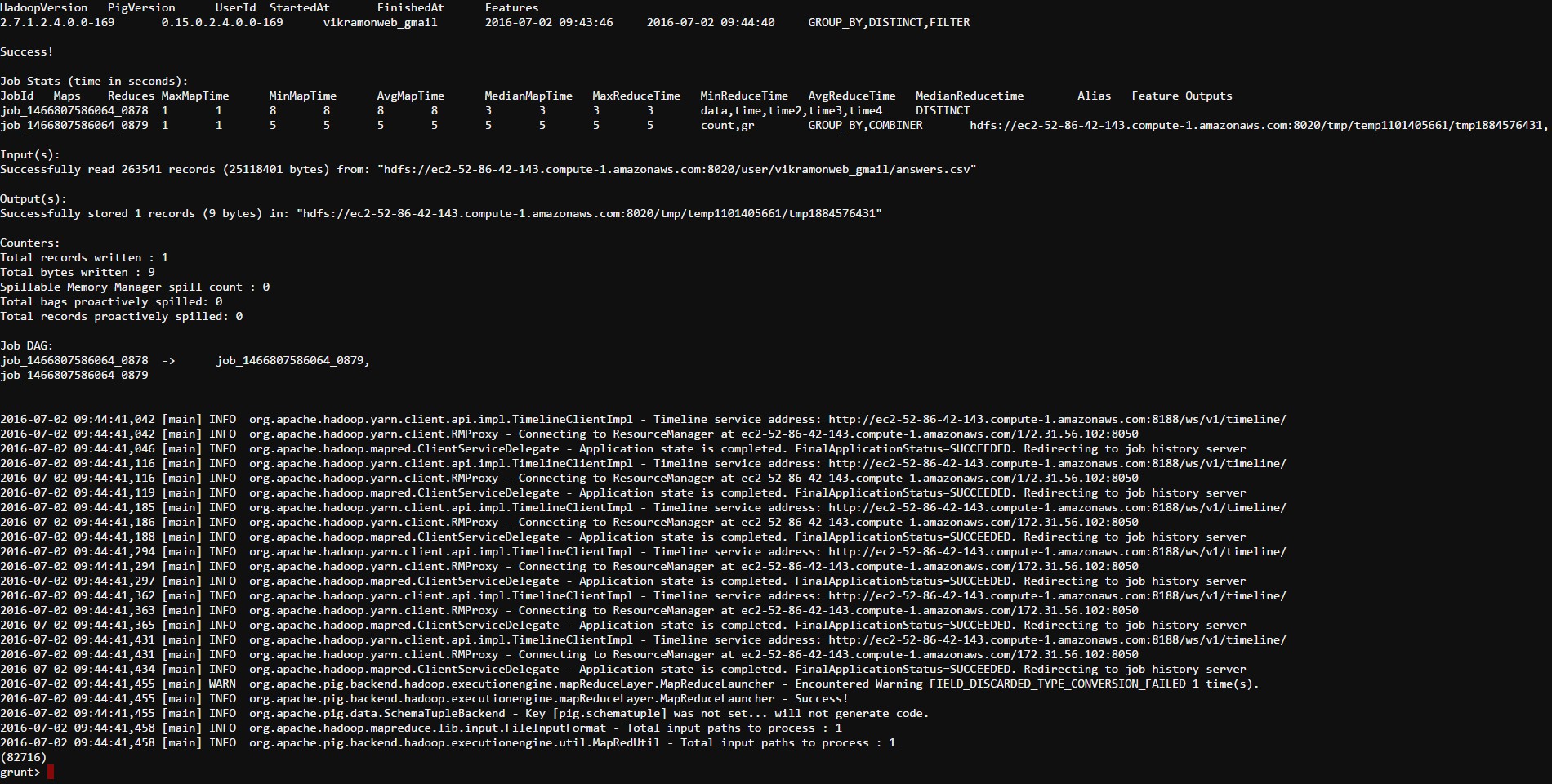
5. Group all the tuples in a bag.

6. Count the elements in the bag.

7. Display the number of questions which got answered within 1 hour.

Screenshots:





**4. Tags of questions which got answered within 1 hour.**

Code:

tag = foreach data generate $1 as qid, $5 as tags;

join\_data = JOIN time5 by $0, tag by qid;

join\_data2 = foreach join\_data generate $0 as qid, $3 as tags;

join\_data3 = DISTINCT join\_data2;

dump join\_data3;

Approach:

1. Generate qid and tags, and store them as tag.

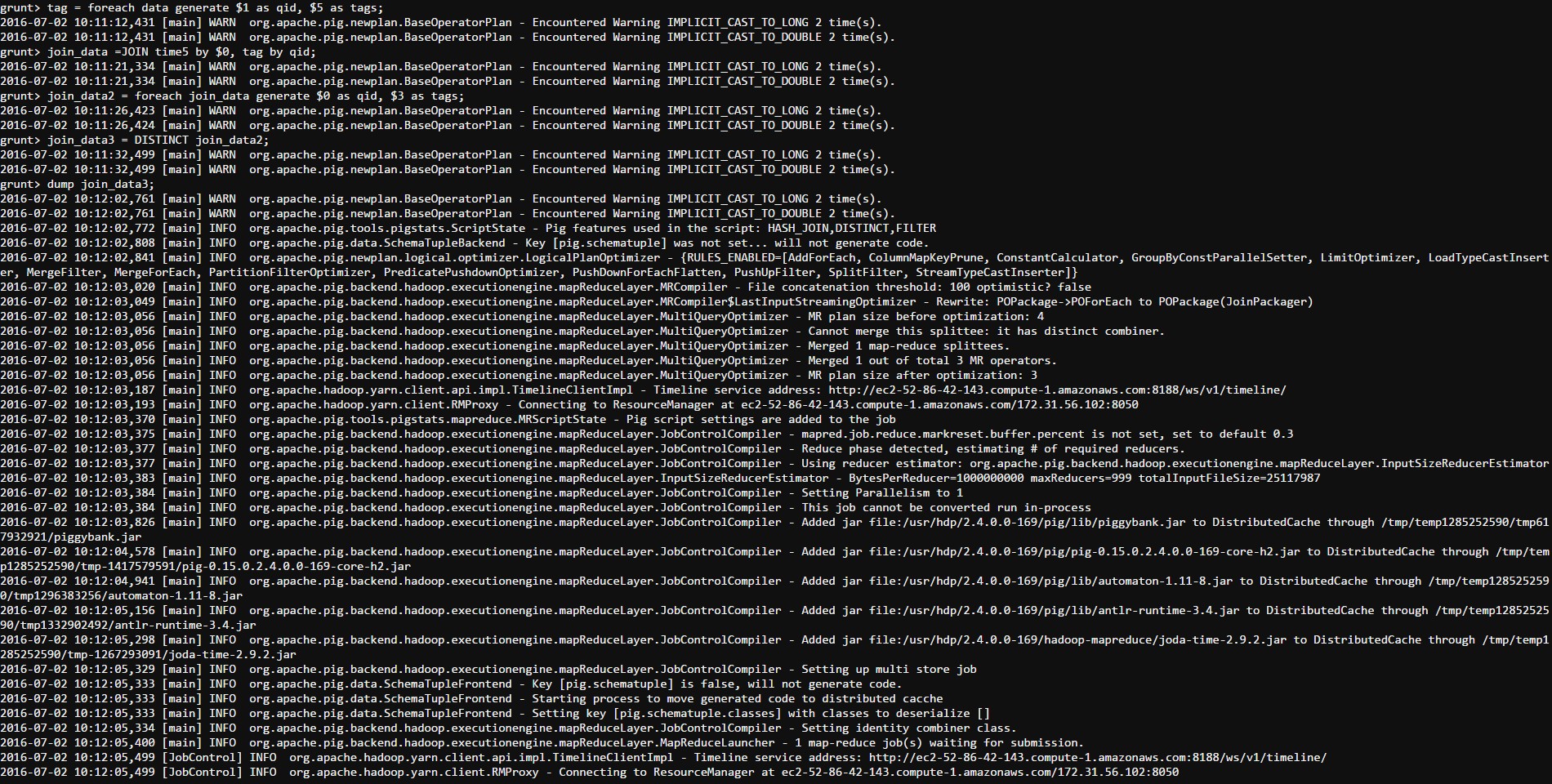
2. Join time5 with tag by using qid as key.

3. For each tuple of joined data, generate qid and tags.

4. Remove duplicate tuples.

5. Display tags of questions which got answered within 1 hour.

Screenshots:

****

