

## Big Data – cs522

### Lab 5 – Day 13 (Monday)

---

Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date. Follow the instructions completely.

---

This document is divided into two parts.

1. [Pig Practice Lab](#)

No need to submit this part.

2. [Pig Homework Lab](#)

Submit “.pig” script files for both the questions along with output files.

Paste screenshots wherever applicable in a separate document.

---

## Pig Practice Lab

---

### Part A – Running Word Count

- Try to run the word count example in Pig (both local and Hadoop mode).
- You can execute each command one by one or create a .pig script file and run it in grunt shell.
- Note that you'll need to modify the LOAD command to give proper path for your input file. Same is the case for the output file path in the STORE command as well.

```
input = LOAD 'input.txt' AS (line:chararray);
words = FOREACH input GENERATE FLATTEN(TOKENIZE (line)) AS word;
grpdc = GROUP words BY word;
cntdc = FOREACH grpdc GENERATE group, COUNT(words);
DUMP cntdc;
STORE cntdc INTO '/user/cloudera/pig/output/' USING PigStorage('\t');
```

### Part B – Running JOIN operator example

- If you find time, then try to execute this example from the lecture slides as well!
- Note that you'll need to create some sample data for "users.csv" and "pages.csv" files.

```
Users = LOAD 'users.csv' AS (name, age);
Fltrdc = FILTER Users BY age >= 18 and age <= 25;
Pages = LOAD 'pages.csv' AS (user, url);
Jnd = JOIN Fltrdc BY name, Pages by user;
Grpdc = GROUP Jnd BY url;
Smmd = FOREACH Grpdc GENERATE group, COUNT(Jnd) AS clicks;
Srtd = ORDER Smmd BY clicks DESC;
Top5 = LIMIT Srtd 5;
STORE Top5 INTO 'top5sites';
```

---

## Pig Homework Lab

---

You've been given a sample [Movies Data set](#). The details of the files and schema are as follows:

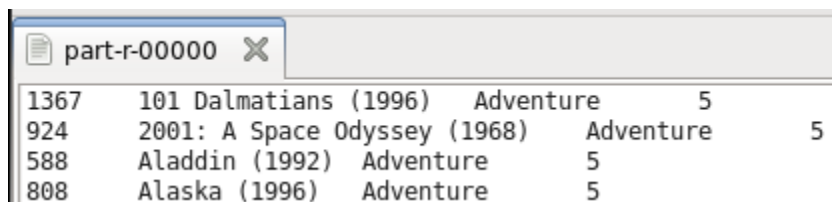
<b>movies.csv</b>	A list of 9000+ movies and their details	{movieId, title, genres}
<b>users.txt</b>	A list of 900+ users and their details	{userId, age, gender, occupation, zipCode}
<b>ratings.txt</b>	~2M file with movie rating details	{userId, movieId, rating, timestamp}

Note that in the movies.csv file, the column *genres* have multiple values in it for one movie which are separated by a pipe symbol (|)<sup>1</sup>.

Now let's do some analysis on this real-world data set using Pig in Hadoop mode (not local mode). But for testing purposes, you can try first in local mode as it's more faster.

1. Display a list of top 20 highest rated (rating=5) "Adventure" movies alphabetically sorted by title. The sample output file will look something like this:

movieId      title                      genres              rating



1367	101 Dalmatians (1996)	Adventure	5
924	2001: A Space Odyssey (1968)	Adventure	5
588	Aladdin (1992)	Adventure	5
808	Alaska (1996)	Adventure	5

2. Out of these highest rated 20 movies, how many male programmers have watched these movies?

Note: Your "part" file should show only one integer as this count.

You might want to take a look at [csvExcelStorage](#) for how to get rid of the first header line from the .csv file.

[More Help here!](#)

---

<sup>1</sup> Those with database experience will notice that this is a violation of the first normal form as defined by E.F. Codd. This intentional denormalization of data is very common in OLAP systems in general, and in large data-processing systems such as Hadoop in particular. RDBMS systems tend to make joins common and then work to optimize them. In systems such as Hadoop, where storage is cheap and joins are expensive, it is generally better to use nested data structures to avoid the joins.