Mod - 2

Pig

Expl: loading & storing - from/to local file system from/to hdfs

⇒ $ pwd
o/p /home/cloudera

⇒ gedit datafile
↳ 1,a
2,b
3,c
4,d
5,e

$ ls
$ cat datafile

grunt // load & store from local file system.
to open grunt prompt in local mode.
$ pig -x local

// loading (no space) (no space)
grunt > bag1 = load '/home/cloudera/datafile' using PigStorage(',') as (id:int, name:chararray);
// to display the content:
grunt > dump bag1;

grunt > bag2 = bag1;

grunt > dump bag2;

new file name

// storing bag1
grunt > store bag1 into '/home/cloudera/output1'
using PigStorage(',');

grunt > quit

⇒ $ ls
$ cat ls output1
$ cat output1/part*

// load & store from/to hdfs

2nd file

ramya, ap, 20.
shree, tn, 22
siti, hyd, 18
mani, ap, 17
guru, blr, 23

⇒ $ gedit datafile2
⇒ $ hadoop fs -put datafile2 /user/cloudera
$ hadoop fs -ls /user/cloudera

$ hadoop fs -cat /user/cloudera/datafile2

⇒ $ pig -x mapreduce

⇒ grunt > student = load '/user/cloudera/datafile2'
using PigStorage(',') as (name:chararray,
city:chararray, age:int);

⇒ grunt > a = student;

to display
// ⇒ grunt > dump a;

store
⇒ grunt > store student into '/user/cloudera/output1'
using PigStorage(',');

⇒ grunt > fs -ls /user/cloudera

⇒ grunt > fs -cat /user/cloudera/output1/part*

rator

Exp2:- diagnostic op - dump, describe, explain, illustrate

```
grunt > describe student;

grunt > explain student;

grunt > illustrate Student;
```

Ex3:- filtering - filter operator, for each generate operator, distinct operator

```
//filter      grunt > A = filter student by city == 'bangalore';
//to display  grunt > dump A;
```

no space.

```
//for each    grunt > c = foreach student generate*;
              grunt > dump c;        //displays all columns

              grunt > c = foreach student generate city;
              grunt > dump c;        //displays only city column

//distinct    grunt > d = distinct c;
              grunt > dump d;        //displays distinct city

//group       grunt > g = group student by age;
              grunt > dump g;

//describe    grunt > describe g;
```

Exp 4:- Arithmetic operators, Comparision operator, Boolean operator

```
// Arithmetic   grunt > f = foreach student generate age, age+10;
                grunt > dump f;   // display age & age+10   Eg:- (20, 30)
                                                               (25, 35)

// comparision  grunt > F = filter student by age >25;
                grunt > dump F;

// boolean.     grunt > e = filter student by age >21 and age<24;
                grunt > dump e;
```