**EXPERIMENT 5**

**PIG**

**Aim:** To understand Data Processing Tool – Pig and to execute pig latin commands.

**Objectives:**

* To read a file from HDFS into Pig environment
* To execute pig commands in grunt shell
* To be familiar with tuples and bags in Pig

**Key concept:**

Pig is an open-source high level data flow system. It provides a simple language called Pig Latin, for queries and data manipulation, which are then compiled in to MapReduce jobs that run on Hadoop.

**Q1: How to enter in grunt shell?**

[cloudera@quickstart Desktop]$ pig

**Q2: Create two data sets using gedit command in local?**

[cloudera@quickstart Desktop]$ cat pigfile.txt

1,2,3

4,5,6

5,7,0

[cloudera@quickstart Desktop]$ gedit pigfile1.txt

2,4,5

2,4,5

4,7,9

**Q3: Copy the above files in HDFS?**

[cloudera@quickstart Desktop]$ hadoop fs -put pigfile.txt /user/cloudera/

[cloudera@quickstart Desktop]$ hadoop fs -put pigfile1.txt /user/cloudera/

**Q3: How to read your (pigfile.txt and pigfile1.txt) data in PIG**

grunt> a = LOAD '/user/cloudera/pigfile.txt' using PigStorage(',');

grunt> dump a;

**OUTPUT**

(1,2,3)

(4,5,6)

(5,7,0)

grunt> b = LOAD '/user/cloudera/pigfile1.txt' using PigStorage(',');

grunt> dump b;

**OUTPUT**

2,4,5

2,4,5

4,7,9

**NOTE: If we want to specify schema, we can, but pig is flexible in that. The columns can be referred as $0 , $1 and so on. But even if you want to specify schema we can.**

**Q4: Specify the schema for above two tables?**

grunt> a = LOAD '/user/cloudera/pigfile.txt' using PigStorage(',') as (a1:int, a2:int, a3:int);

grunt> dump a;

grunt> b = LOAD '/user/cloudera/pigfile1.txt' using PigStorage(',') as (b1:int, b2:int, b3:int);

grunt> dump b;

**Q5: Check the schema of the two tables?**

grunt> describe a;

**OUTPUT**

a: {a1: int,a2: int,a3: int}

grunt> describe b;

**OUTPUT**

b: {b1: int,b2: int,b3: int}

**Q6: Combine the two tables**

grunt> c= union a,b;

grunt> dump c

**OUTPUT**

(2,4,5)

(2,4,5)

(4,7,9)

(1,2,3)

(4,5,6)

(5,7,0)

**Q7: Split the c data set into two different relations eg. d and e? E.g. I want one data set where $0 is having value 1 and other data set where value of $0 is 4**

grunt> SPLIT c INTO d IF $0 == 1 , e IF $0 == 4;

dump d;

**OUTPUT**

(1,2,3) :- here this is the row starting with 1.

dump e;

(4,7,9) :- here this is the row starting with 4.

(4,5,6) :- here this is the row starting with 4.

**Q8: Do filtering on data set c where $1 is greater than 6?**

grunt> f = FILTER c BY $1 > 6;

grunt> dump f

**OUTPUT**

(5,7,0) :: here $1 means column 1, so column 1>6

(4,7,9) :: here $1 means column 1, so column 1>6

**Q9: Group data set c by $2?**

grunt> g = GROUP c by $2;

grunt> dump g;

**OUTPUT**

(0,{(5,7,0)})

(3,{(1,2,3)})

(5,{(2,4,5),(2,4,5)})

(6,{(4,5,6)})

(9,{(4,7,9)})

**Q: 10: Select column 1 and 2 from dataset a ?**

grunt> s1 = foreach a generate $1,$2;

grunt> dump s1;

**OUTPUT**

(2,3)

(5,6)

(7,0)

**Q11: Store the above result in HDFS?**

grunt> store s1 into '/user/cloudera/pigresult';

**Q12: check the file written in HDFS**

[cloudera@quickstart Desktop]$ hadoop fs -ls /user/cloudera/pigresult/

**OUTPUT**

Found 2 items

-rw-r--r-- 1 clouderacloudera 0 2018-09-19 03:50 /user/cloudera/pigresult/\_SUCCESS

-rw-r--r-- 1 clouderacloudera 12 2018-09-19 03:50 /user/cloudera/pigresult/part-m-00000

**Now see what’s inside part-m-00000**

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/cloudera/pigresult/part-m-00000

2 3

5 6

7 0