# BIG DATA TECHNOLOGIES LAB

## ASSIGNMENT 1
## TOPIC : LINUX COMMANDS

ASHISH DUSANE
M. TECH (ACDS)

{ PRN : 170847980003 }

1. Find out the username/userid on your machine.

**Ans: [root@localhost home]# whoami**

2. How to print the present working directory path?

Ans: PWD

3. How to go to a given directory (change the directory)?

**Ans: cd /directory_path**

4. What is the command to view the information about a Linux/Unix command? Try to know about the commands “ls”.

**Ans: Shows the listed files and folders in the current directory.**

**MAN used to see details about that commands**

5. View the man page for command ‘id’ and find the options for printing all the groups associated to a user with name.

**Ans: id -G**

6. Create a file called “ctest”. Modify the access permissions as: “user - rwx”, “group - rx”, and “others - nothing”.

**Ans: mkdir ctest**

7. Create a 3 level directory in your home directory as shown in the structure below with single command.

mydir

|

Cd /home

Mkdir mydir

Subdir1

Cd mydir

Mkdir subdir

Subdir2

Ans: cd subdir2

Mkdir subdir2

Now move the whole directory i.e. mydir to mydir1 and change the permissions as

owner-rwx, group-rx, others-x.

**Ans: mkdir –p  /mydir/subdir1/subdir2**

```
[root@localhost subdir]# ls -g

total 0

drwxrw----. 3 root 20 Dec 18 10:55 mydir

[root@localhost subdir]#

[root@localhost home]# ls -g acts.txt

-rw-r--r--. 1 root 195 Dec 18 11:05 acts.txt

[root@localhost home]# chmod 760 a.txt

[root@localhost home]# ls -g a.txt

-rwxrw----. 1 root 0 Dec 18 11:00 a.txt
```

```
[root@localhost home]# mv subdir mkdir1
```

```
[root@localhost home]# ls

acts.txt  a.txt  master  mkdir1

[root@localhost home]# ls -g mkdir

ls: cannot access mkdir: No such file or directory

[root@localhost home]# cd mkdir1

[root@localhost mkdir1]# ls -g

total 0

total 8

-rw-r--r--.  1 root    195 Dec 18 11:05 acts.txt

-rwxrw----.  1 root      0 Dec 18 11:00 a.txt

drwx------. 14 master 4096 Dec 15 14:24 master

drwxrw----.  3 root     19 Dec 18 11:32 mkdir1
```

8 Create a text file using an editor with content as:

Simple to understand and work.

Many os are based on LINUX.

All linux OS are open source and freely available.

a) From the file get the count of word “Linux” it must also include LINUX while counting.

Ans: [root@localhost home]# grep -ic 'Linux' acts.txt

```
3
```

b) Display lines that exactly have the word ‘OS’.

**Ans: [root@localhost home]# grep -c 'OS' acts.txt**

```
2
```

c) Display the line number where appears word ‘Linux’ irrespective of case

```
Ans: [root@localhost home]# grep -n 'Linux' acts.txt
2:First day is about Linux.
```

9. Create sample files ending with “.txt” in your directory. Now run a command to list all the files ending with “txt”.

```
Ans: [root@localhost home]# find *.txt
abc.txt
acts.txt
a.txt
```

10. Create a file “test” in the directory /home/cdac/mydir1/subdir1/test. Go to the home directory and now find the file “test” executing the appropriate command.

```
Ans: ind: ‘test’: No such file or directory
[root@localhost home]# find -name test
./cdac/mkdir1/subdir/mydir/subdir2/test
```

11. From home directory list all the files including the hidden files.

```
Ans: [root@localhost home]# find *
abc.txt
acts.txt
a.txt
cdac
cdac/mkdir1
cdac/mkdir1/subdir
cdac/mkdir1/subdir/mydir
cdac/mkdir1/subdir/mydir/subdir2
```

```
cdac/mkdir1/subdir/mydir/subdir2/test

master

master/.mozilla

master/.mozilla/firefox/profiles.ini

master/.bash_logout

master/.bash_profile

master/.bashrc

master/.config

master/.config/imsettings

master/.config/user-dirs.dirs

master/.config/user-dirs.locale

master/.config/gnome-session

master/.config/gnome-session/saved-session

master/.config/pulse

master/.config/pulse/cookie

master/.config/pulse/6e59fe2995ea46eba32300bb50cf2d42-device-volumes.tdb

master/.config/pulse/6e59fe2995ea46eba32300bb50cf2d42-stream-volumes.tdb

master/.local/share/evolution/mail
```

12. Create a file with the following content:

You absolutely must run these exercises in the bash shell, or results will not be as expected.

1. Display first 2 lines from the file.

**Ans: [root@localhost home]# sed -n '1,2p' acts.txt**

Welcome to ACTS courses.

First day is about Linux.

2. Display last 2 lines from the file.

```
Ans: [root@localhost home]# tail -2 demo.txt
expected.
```

3. Display exactly the tenth line in the file.

```
[root@localhost home]# head -n 10 d.txt | tail -n 1
hsfshsdf
[root@localhost home]# sed -n '10,10p' d.txt
hsfshsdf
```

13. Create a set of 3 files and tar them to myfirst.tar. Untar the tar file myfirst.tar and check the extracted files.

```
Ans: [root@localhost tar]# cd tarfiles
[root@localhost tarfiles]# ls
tar2  tar3
[root@localhost tarfiles]# mkdir tar 1
[root@localhost tarfiles]# ld
ld: no input files
[root@localhost tarfiles]# ld
ld: no input files
[root@localhost tarfiles]# ls
1  tar  tar2  tar3
[root@localhost tarfiles]# tar -zcvf TARFILES1 tarfiles
tar: tarfiles: Cannot stat: No such file or directory
```

```
tar: Exiting with failure status due to previous errors

[root@localhost tarfiles]# tar -zcvf TARFILES1 tar

tar/

[root@localhost tarfiles]# ls

1  tar  tar2  tar3  TARFILES1

[root@localhost tarfiles]#

[root@localhost tarfiles]# tar -zcvf TARFILES2.tar tar tar2 tar3

tar/

tar2/

tar3/

[root@localhost tarfiles]# ls

1  tar  tar2  tar3  TARFILES1  TARFILES2.tar
```

# BIG DATA TECHNOLOGIES LAB

## ASSIGNMENT 2
## TOPIC: HDFS COMMANDS

ASHISH DUSANE

M. TECH (ACDS)

{ PRN : 170847980003 }

To see the help page of hdfs commands, you may type `hdfs dfs -help`
To see help of any command on hdfs you may type `hdfs dfs -help comamnd_name`

1. Create a directory on HDFS in your home directory (Hadoop Distributed File System).

**Ans.**
```
[root@localhost /]#  hadoop fs -mkdir /Hadoopfs
```

2. Create two more directories in a single command in your home directory.

**Ans.**
```
[root@localhost /]# hadoop fs -mkdir /Hadp /hgg
```

3. List the directories created in HDFS and check in what sort order are the contents listed by default?

**Ans.**
```
[root@localhost /]# hadoop fs -ls /

    Found 8 items

    drwxr-xr-x   - root   supergroup        0 2017-12-19 10:09 /Hadoopfs

    drwxr-xr-x   - root   supergroup        0 2017-12-19 10:10 /Hadp

    drwxr-xr-x   - root   supergroup        0 2017-12-19 10:10 /hgg

    drwxr-xr-x   - root   supergroup        0 2017-12-16 11:54 /ip1

    drwxr-xr-x   - root   supergroup        0 2017-12-16 12:01 /ip2

    drwxr-xr-x   - mapred supergroup        0 2017-12-13 15:16 /mapred

    drwx------   - root   supergroup        0 2017-12-16 11:51 /usr

    drwxr-xr-x   - root   supergroup        0 2017-12-16 11:41 /wordcount
```

4. Create a sample file (eg: sample.txt) in any of the directories created above.

**Ans.**
```
[root@localhost /]# hadoop fs -touchz /Hadoopfs/sample.txt
```

5. Copy a file from local file system to one of the directories created on HDFS. (This process of copying file from local file system to HDFS called as Uploading files to HDFS).

**Ans.**
```
[root@master Desktop]# hadoop fs -put Installing\ Pig\ And\ Hive.pdf /Hadoopfs
```

6. View the uploaded file.

**Ans.**
```
[root@master Desktop]# hadoop fs -ls /Hadoopfs

    Found 2 items

    -rw-r--r--    1 root supergroup      884385 2017-12-19 10:30 /Hadoopfs/Installing Pig And
    Hive.pdf-rw-r--r--   1 root supergroup        0 2017-12-19 10:25 /Hadoopfs/sample.txt
```

7. Copy one more file from local file system to HDFS to another directory created.

```
Ans. [root@localhost /]# hadoop fs -put data.txt /Hadp
```

8. Copy a file from HDFS to local file system (This is called as Downloading a file from HDFS to local file system).

```
Ans. [root@localhost /]# hadoop fs -get 64\ bit\ -\ Hadoop\ Install\ Process.pdf /Hadp
```

9. Look at the contents in the file that is uploaded on HDFS.

```
Ans. [root@localhost /]# hadoop fs -cat /Hadp/data.txt

Hi..

MTech ACDS CDAC

Nashik
```

10. Copy the file from one directory to another directory in HDFS.

```
Ans. [root@localhost /]# hadoop fs -cp /Hadp/data.txt /Hadoopfs
```

11. Move the file from one directory to another directory in HDFS.

```
Ans. [root@localhost /]# hadoop fs -mv /Hadoopfs/data.txt /hgg
```

12. Copy a file from/To Local file system to HDFS. Use copyFromLocal and copyToLocal commands

```
Ans. [root@localhost /]# hadoop fs -copyFromLocal Labs.txt /

[root@localhost /]# hadoop fs -copyToLocal /Hadoopfs/sample.txt
```

13. Display last few lines from the file in HDFS.

```
Ans. [root@localhost /]# hadoop fs -tail /Labs.txt
```

***1. Install and configure pig and hive on a given Hadoop environment.***

***2. Write a java mapreduce program to count the number of word occurrence in a document.***

14. Display the size of the file in KB and MB in the HDFS.

```
Ans. [root@localhost /]# hadoop fs -du -h /Hadoopfs/Installing*.pdf

863.7 K  /Hadoopfs/Installing Pig And Hive.pdf
```

15. Append a file from Local File to system to file on HDFS

```
Ans. [root@localhost /]# hadoop fs -appendTofile /Labs.txt
```

16. Merge two file contents (files present on HDFS) in to one file (this file should be present on Local file system)

```
Ans. [root@localhost /]# hadoop fs -cat /Labs.txt /Labs.txt | hadoop fs -put - /dataopllll.txt
```

17. Get Access Control Lists (ACL's) of the files and directories created.

```
Ans. [root@localhost /]#hdfs dfs –getfacl [-R] /Labs.txt
```

18. Copy one directory structure to another.

```
Ans. [root@localhost /]# hadoop fs -cp /Hadp /hgg
```

19. Set the replication to the file created to 4

```
Ans. [root@localhost /]# hadoop fs -setrep 4 /Labs.txt
Replication 4 set: /Labs.txt

[root@localhost /]# hadoop fs -ls /Labs.txt
Found 1 items
-rw-r--r--   4 root supergroup       159 2017-12-19 11:13 /Labs.txt
```

20. Remove a file from the directory in HDFS.

```
Ans. [root@localhost /]# hadoop fs -rm /Labs.txt

Deleted /Labs.txt
```

21. Remove a directory in HDFS.

```
Ans. [root@localhost /]# hadoop fs -rmdir /Hdp
```

# BIG DATA TECHNOLOGIES LAB

## ASSIGNMENT 3

## TOPIC: PIG PRACTICE SESSION I

ASHISH DUSANE

M. TECH (ACDS)

{ PRN : 170847980003 }

## ~: PIG Practice Session I :~

## To Extract data from HDFS

```
student1 = LOAD '/home/student/Documents/Pig/student_details.txt' USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray);
```

```
grunt> dump student1;
```

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad)

(2,siddarth,Battacharya,22,9848022338,Kolkata)

(3,Rajesh,Khanna,22,9848022339,Delhi )

(4,Preethi,Agarwal,21,9848022330,Pune )

(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar )

(6,Archana,Mishra,23,9848022335,Chennai )

(7,Komal,Nayak,24,9848022334,trivendram )

(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

## To Load data into HDFS

```
STORE student1 INTO '/home/student/Documents/Pig/pig_Output/' USING PigStorage (',');
```

# Output :

```
1,Rajiv,Reddy,21,9848022337,Hyderabad

2,siddarth,Battacharya,22,9848022338,Kolkata

3,Rajesh,Khanna,22,9848022339,Delhi

4,Preethi,Agarwal,21,9848022330,Pune

5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
```

```
6,Archana,Mishra,23,9848022335,Chennai

7,Komal,Nayak,24,9848022334,trivendram

8,Bharathi,Nambiayar,24,9848022333,Chennai
```

```
## Group Operator 1

## Student Details

hdfs dfs -put student_details.txt /pig

hdfs dfs -ls /pig

hdfs dfs -cat /pig/student_details.txt

student_details = LOAD 'hdfs://master:8020/pig/student_details.txt' USING PigStorage(',') as

(id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray);

group_data = GROUP student_details by age;

Dump group_data;
```

```
## Group Operator 2

group_multiple = GROUP student1 by (age, city);

dump group_multiple;

# Output :

((21,Pune ),{(4,Preethi,Agarwal,21,9848022330,Pune )})

((21,Hyderabad),{(1,Rajiv,Reddy,21,9848022337,Hyderabad)})

((22,Delhi ),{(3,Rajesh,Khanna,22,9848022339,Delhi )})

((22,Kolkata),{(2,siddarth,Battacharya,22,9848022338,Kolkata)})

((23,Chennai ),{(6,Archana,Mishra,23,9848022335,Chennai )})
```

```
((23,Bhuwaneshwar ),{(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
)})

((24,Chennai),{(8,Bharathi,Nambiayar,24,9848022333,Chennai)})

((24,trivendram ),{(7,Komal,Nayak,24,9848022334,trivendram )})
```

## Group Operator 3

```
group_all = GROUP student1  All;

dump group_all;
```

# Output :

```
(all,{(8,Bharathi,Nambiayar,24,9848022333,Chennai),(7,Komal,Nayak,24,984
8022334,trivendram ),(6,Archana,Mishra,23,9848022335,Chennai
),(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
),(4,Preethi,Agarwal,21,9848022330,Pune
),(3,Rajesh,Khanna,22,9848022339,Delhi
),(2,siddarth,Battacharya,22,9848022338,Kolkata),(1,Rajiv,Reddy,21,9848022
337,Hyderabad)})
```

## Cogroup

```
hdfs dfs -put employee_details.txt /pig

hdfs dfs -ls /pig/employee_details.txt

hdfs dfs -cat /pig/employee_details.txt
```

```
employee_details = LOAD
'/home/student/Documents/Pig/employee_details.txt' USING PigStorage(',') as
(id:int, name:chararray, age:int, city:chararray);

dump employee_details;
```

# Output :

```
(1,Robin,22,newyork)

(2,BOB,23,Kolkata)
```

```
(3,Maya,23,Tokyo)

(4,Sara,25,London)

(5,David,23,Bhuwaneshwar)

(6,Maggy,22,Chennai)
```

```
cogroup_data = COGROUP student1 by age, employee_details by age;

Dump cogroup_data;
```

# Output :

```
(21,{(4,Preethi,Agarwal,21,9848022330,Pune
),(1,Rajiv,Reddy,21,9848022337,Hyderabad)},{})

(22,{(3,Rajesh,Khanna,22,9848022339,Delhi
),(2,siddarth,Battacharya,22,9848022338,Kolkata)},{(6,Maggy,22,Chennai),(1,
Robin,22,newyork)})

(23,{(6,Archana,Mishra,23,9848022335,Chennai
),(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
)},{(5,David,23,Bhuwaneshwar),(3,Maya,23,Tokyo),(2,BOB,23,Kolkata)})

(24,{(8,Bharathi,Nambiayar,24,9848022333,Chennai),(7,Komal,Nayak,24,9848
022334,trivendram )},{})

(25,{},{(4,Sara,25,London)})
```

## Join Operator

```
hdfs dfs -put customers.txt /pig

hdfs dfs -put orders.txt /pig

customers = LOAD '/home/student/Documents/Pig/customers.txt' USING
PigStorage(',') as (id:int, name:chararray, age:int, address:chararray,
salary:int);

dump customers;
```

# Output :

```
(1,Ramesh,32,Ahmedabad,2000)
(2,Khilan,25,Delhi,1500)
(3,kaushik,23,Kota,2000)
(4,Chaitali,25,Mumbai,6500)
(5,Hardik,27,Bhopal,8500)
(6,Komal,22,MP,4500)
(7,Muffy,24,Indore,10000)
```

```
orders = LOAD '/home/student/Documents/Pig/orders.txt' USING PigStorage(',') as (oid:int, date:chararray, customer_id:int, amount:int);
dump orders;
# Output :
(102,2009-10-08 00:00:00,3,3000)
(100,2009-10-08 00:00:00,3,1500)
(101,2009-11-20 00:00:00,2,1560)
(103,2008-05-20 00:00:00,4,2060)
```

```
## Self-Join :
customers1 = LOAD '/home/student/Documents/Pig/customers.txt' USING PigStorage(',') as (id:int,name:chararray, age:int, address:chararray, salary:int);
dump customers1;
# Output :
(1,Ramesh,32,Ahmedabad,2000)
(2,Khilan,25,Delhi,1500)
```

```
(3,kaushik,23,Kota,2000)
(4,Chaitali,25,Mumbai,6500)
(5,Hardik,27,Bhopal,8500)
(6,Komal,22,MP,4500)
(7,Muffy,24,Indore,10000)
```

```
customers2 = LOAD '/home/student/Documents/Pig/customers.txt' USING
PigStorage(',') as (id:int, name:chararray, age:int, address:chararray,
salary:int);
```

```
# Output :
(1,Ramesh,32,Ahmedabad,2000)
(2,Khilan,25,Delhi,1500)
(3,kaushik,23,Kota,2000)
(4,Chaitali,25,Mumbai,6500)
(5,Hardik,27,Bhopal,8500)
(6,Komal,22,MP,4500)
(7,Muffy,24,Indore,10000)
```

```
customers3 = JOIN customers1 BY id, customers2 BY id;
Dump customers3;
```

```
# Output :
(1,Ramesh,32,Ahmedabad,2000,1,Ramesh,32,Ahmedabad,2000)
(2,Khilan,25,Delhi,1500,2,Khilan,25,Delhi,1500)
(3,kaushik,23,Kota,2000,3,kaushik,23,Kota,2000)
(4,Chaitali,25,Mumbai,6500,4,Chaitali,25,Mumbai,6500)
(5,Hardik,27,Bhopal,8500,5,Hardik,27,Bhopal,8500)
```

```
(6,Komal,22,MP,4500,6,Komal,22,MP,4500)
(7,Muffy,24,Indore,10000,7,Muffy,24,Indore,10000)
```

## Inner Join :

```
customer_orders = JOIN customers BY id, orders BY customer_id;
Dump customer_orders;
```

# Output :

```
(2,Khilan,25,Delhi,1500,101,2009-11-20 00:00:00,2,1560)
(3,kaushik,23,Kota,2000,100,2009-10-08 00:00:00,3,1500)
(3,kaushik,23,Kota,2000,102,2009-10-08 00:00:00,3,3000)
(4,Chaitali,25,Mumbai,6500,103,2008-05-20 00:00:00,4,2060)
```

## Outer Joins

```
outer_left = JOIN customers BY id LEFT OUTER, orders BY customer_id;
Dump outer_left;
```

# Output :

```
(1,Ramesh,32,Ahmedabad,2000,,,,)
(2,Khilan,25,Delhi,1500,101,2009-11-20 00:00:00,2,1560)
(3,kaushik,23,Kota,2000,100,2009-10-08 00:00:00,3,1500)
(3,kaushik,23,Kota,2000,102,2009-10-08 00:00:00,3,3000)
(4,Chaitali,25,Mumbai,6500,103,2008-05-20 00:00:00,4,2060)
(5,Hardik,27,Bhopal,8500,,,,)
(6,Komal,22,MP,4500,,,,)
(7,Muffy,24,Indore,10000,,,,)
```

```
outer_right = JOIN customers BY id RIGHT, orders BY customer_id;

Dump outer_right;

# Output :

(2,Khilan,25,Delhi,1500,101,2009-11-20 00:00:00,2,1560)

(3,kaushik,23,Kota,2000,100,2009-10-08 00:00:00,3,1500)

(3,kaushik,23,Kota,2000,102,2009-10-08 00:00:00,3,3000)

(4,Chaitali,25,Mumbai,6500,103,2008-05-20 00:00:00,4,2060)
```

```
outer_full = JOIN customers BY id FULL OUTER, orders BY customer_id;

Dump outer_full;

# Output :

(1,Ramesh,32,Ahmedabad,2000,,,,)

(2,Khilan,25,Delhi,1500,101,2009-11-20 00:00:00,2,1560)

(3,kaushik,23,Kota,2000,100,2009-10-08 00:00:00,3,1500)

(3,kaushik,23,Kota,2000,102,2009-10-08 00:00:00,3,3000)

(4,Chaitali,25,Mumbai,6500,103,2008-05-20 00:00:00,4,2060)

(5,Hardik,27,Bhopal,8500,,,,)

(6,Komal,22,MP,4500,,,,)

(7,Muffy,24,Indore,10000,,,,)
```

```
## Joins by Multiple-Keys

employee = LOAD 'hdfs://master:8020/pig/employee.txt' USING PigStorage(',')

as (id:int, firstname:chararray, lastname:chararray, age:int, designation:chararray,
jobid:int);
```

```
employee_contact = LOAD 'hdfs://master:8020/pig/employee_contact.txt' USING PigStorage(',')as (id:int, phone:chararray, email:chararray, city:chararray, jobid:int);
```

## Cross operator

```
cross_data = CROSS customers, orders;
Dump cross_data;
```

# Output :

```
(7,Muffy,24,Indore,10000,103,2008-05-20 00:00:00,4,2060)
(7,Muffy,24,Indore,10000,101,2009-11-20 00:00:00,2,1560)
(7,Muffy,24,Indore,10000,100,2009-10-08 00:00:00,3,1500)
(7,Muffy,24,Indore,10000,102,2009-10-08 00:00:00,3,3000)
(6,Komal,22,MP,4500,103,2008-05-20 00:00:00,4,2060)
(6,Komal,22,MP,4500,101,2009-11-20 00:00:00,2,1560)
(6,Komal,22,MP,4500,100,2009-10-08 00:00:00,3,1500)
(6,Komal,22,MP,4500,102,2009-10-08 00:00:00,3,3000)
(5,Hardik,27,Bhopal,8500,103,2008-05-20 00:00:00,4,2060)
(5,Hardik,27,Bhopal,8500,101,2009-11-20 00:00:00,2,1560)
(5,Hardik,27,Bhopal,8500,100,2009-10-08 00:00:00,3,1500)
(5,Hardik,27,Bhopal,8500,102,2009-10-08 00:00:00,3,3000)
(4,Chaitali,25,Mumbai,6500,103,2008-05-20 00:00:00,4,2060)
(4,Chaitali,25,Mumbai,6500,101,2009-11-20 00:00:00,2,1560)
(4,Chaitali,25,Mumbai,6500,100,2009-10-08 00:00:00,3,1500)
(4,Chaitali,25,Mumbai,6500,102,2009-10-08 00:00:00,3,3000)
(3,kaushik,23,Kota,2000,103,2008-05-20 00:00:00,4,2060)
(3,kaushik,23,Kota,2000,101,2009-11-20 00:00:00,2,1560)
```

```
(3,kaushik,23,Kota,2000,100,2009-10-08 00:00:00,3,1500)
(3,kaushik,23,Kota,2000,102,2009-10-08 00:00:00,3,3000)
(2,Khilan,25,Delhi,1500,103,2008-05-20 00:00:00,4,2060)
(2,Khilan,25,Delhi,1500,101,2009-11-20 00:00:00,2,1560)
(2,Khilan,25,Delhi,1500,100,2009-10-08 00:00:00,3,1500)
(2,Khilan,25,Delhi,1500,102,2009-10-08 00:00:00,3,3000)
(1,Ramesh,32,Ahmedabad,2000,103,2008-05-20 00:00:00,4,2060)
(1,Ramesh,32,Ahmedabad,2000,101,2009-11-20 00:00:00,2,1560)
(1,Ramesh,32,Ahmedabad,2000,100,2009-10-08 00:00:00,3,1500)
(1,Ramesh,32,Ahmedabad,2000,102,2009-10-08 00:00:00,3,3000)
```

## Union Operators

```
student2 = LOAD '/home/student/Documents/Pig/student_details.txt' USING
PigStorage(',') as (id:int, firstname:chararray, lastname:chararray,
age:int,phone:chararray, city:chararray);


dump student2;
```

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(3,Rajesh,Khanna,22,9848022339,Delhi )
(4,Preethi,Agarwal,21,9848022330,Pune )
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar )
(6,Archana,Mishra,23,9848022335,Chennai )
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

```
student3 = LOAD '/home/student/Documents/Pig/student_details1.txt' USING
PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int,
phone:chararray, city:chararray);
```

```
# Output :
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

```
student = UNION student2, student3;
Dump student;
```

```
# Output :
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(3,Rajesh,Khanna,22,9848022339,Delhi )
(4,Preethi,Agarwal,21,9848022330,Pune )
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar )
(6,Archana,Mishra,23,9848022335,Chennai )
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

## Split Command

```
SPLIT student into student_details if age<23, student_details1 if (21<age and age>23);
```

**Dump student_details;**

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(3,Rajesh,Khanna,22,9848022339,Delhi )
(4,Preethi,Agarwal,21,9848022330,Pune )
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
```

Dump student_details1;

# Output :

```
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

# BIG DATA TECHNOLOGIES LAB

## ASSIGNMENT 4

## TOPIC: PIG PRACTICE SESSION II

ASHISH DUSANE

M. TECH (ACDS)

{ PRN : 170847980003 }

## ~: Practice Session II :~

# Filter Operator :~

```
student_details3 = LOAD '/home/student/Documents/Pig/student_details.txt'
USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int,
phone:chararray, city:chararray);
```

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(3,Rajesh,Khanna,22,9848022339,Delhi )
(4,Preethi,Agarwal,21,9848022330,Pune )
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar )
(6,Archana,Mishra,23,9848022335,Chennai )
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

```
filter_data = FILTER student_details3 BY city == 'Chennai';
Dump filter_data;
```

# Output :

```
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

# Distinct Operator :~

```
distinct_data = DISTINCT student_details3;
Dump distinct_data;
```

(Duplicate Recors deleted)

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(3,Rajesh,Khanna,22,9848022339,Delhi )
(4,Preethi,Agarwal,21,9848022330,Pune )
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar )
(6,Archana,Mishra,23,9848022335,Chennai )
(7,Komal,Nayak,24,9848022334,trivendram )
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

# Foreach Operator :~

```
foreach_data = FOREACH student_details GENERATE id,age,city;
Dump foreach_data;
```

# Output :

```
(1,21,Hyderabad)
(2,22,Kolkata)
(3,22,Delhi )
(4,21,Pune )
(5,23,Bhuwaneshwar )
(6,23,Chennai )
(7,24,trivendram )
(8,24,Chennai)
```

# Order Operator :~

```
order_by_data = ORDER student_details BY age DESC;
Dump order_by_data;
```

# Output :

```
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
(7,Komal,Nayak,24,9848022334,trivendram )
(6,Archana,Mishra,23,9848022335,Chennai )
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar )
(3,Rajesh,Khanna,22,9848022339,Delhi )
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(4,Preethi,Agarwal,21,9848022330,Pune )
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
```

# Limit Operator :~

```
limit_data = LIMIT student_details 4;
Dump limit_data;
```

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(3,Rajesh,Khanna,22,9848022339,Delhi )
(4,Preethi,Agarwal,21,9848022330,Pune )
```

# Average Function Operator :~

```
student_details = LOAD '/home/student/Documents/Pig/student_details.txt'
USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int,
phone:chararray, city:chararray, gpa:int);
```

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad,89)
(2,siddarth,Battacharya,22,9848022338,Kolkata,78)
(3,Rajesh,Khanna,22,9848022339,Delhi,90)
(4,Preethi,Agarwal,21,9848022330,Pune,93)
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75)
(6,Archana,Mishra,23,9848022335,Chennai,87)
(7,Komal,Nayak,24,9848022334,trivendram,83)
(8,Bharathi,Nambiayar,24,9848022333,Chennai,72)
```

```
student_group_all = Group student_details All;
student_gpa_avg = foreach student_group_all GENERATE
(student_details.firstname, student_details.gpa), AVG(student_details.gpa);
Dump student_gpa_avg;
```

# Output :

```
(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Raji
v)},{(72),(83),(87),(75),(93),(90),(78),(89)}),83.375)
```

# BagToString Function :~

# Syntax : BagToString(vals:bag [, delimiter:chararray])

```
dob = LOAD '/home/student/Documents/Pig/dob.txt' USING PigStorage(',') as (day:int,month:int, year:int);

dump dob;
```

# Output :

```
(22,3,1990)
(23,11,1989)
(1,3,1998)
(2,6,1980)
(26,9,1989)
```

```
group_dob = Group dob All;

Dump group_dob;
```

# Output :

```
(all,{(26,9,1989),(2,6,1980),(1,3,1998),(23,11,1989),(22,3,1990)})
```

```
dob_string = foreach group_dob Generate BagToString(dob);

Dump dob_string;
```

## Output

```
(26_9_1989_2_6_1980_1_3_1998_23_11_1989_22_3_1990)
```

# Concat Function :~

```
student_name_concat = foreach student_details Generate CONCAT (firstname, lastname);
```

# Output :

```
(RajivReddy)
(siddarthBattacharya)
(RajeshKhanna)
(PreethiAgarwal)
(TrupthiMohanthy)
(ArchanaMishra)
(KomalNayak)
(BharathiNambiayar)
```

```
student_name_concat = foreach student_details Generate CONCAT(firstname, '_',lastname);

Dump student_name_concat;
```

# Output :

```
(Rajiv_Reddy)
(siddarth_Battacharya)
(Rajesh_Khanna)
(Preethi_Agarwal)
(Trupthi_Mohanthy)
(Archana_Mishra)
(Komal_Nayak)
(Bharathi_Nambiayar)
```

# Count Function :~

```
student_details1 = LOAD '/home/student/Documents/Pig/student_cgpa.txt' USING
PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int,
phone:chararray, city:chararray, gpa:int);
```

# Output :

```
(1,Rajiv,Reddy,21,9848022337,Hyderabad,89)
(2,siddarth,Battacharya,22,9848022338,Kolkata,78)
(3,Rajesh,Khanna,22,9848022339,Delhi,90)
(4,Preethi,Agarwal,21,9848022330,Pune,93)
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75)
(6,Archana,Mishra,23,9848022335,Chennai,87)
(7,Komal,Nayak,24,9848022334,trivendram,83)
(8,Bharathi,Nambiayar,24,9848022333,Chennai,72)
```

```
student_group_all = Group student_details1 All;
student_count = foreach student_group_all Generate
COUNT(student_details1.gpa);
```

# Output :

```
(8)
```

# Count_Star Function :~

```
student_count = foreach student_group_all Generate
COUNT_STAR(student_details1.gpa);
Dump student_count;
```

count NULL values also

# Output :

```
(8)
```

# Diff Function :~

```
hdfs dfs -put emp_sales.txt /pig

hdfs dfs -put emp_bonus.txt /pig


emp_sales = LOAD '/home/student/Documents/Pig/emp_sales.txt' USING
PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);


emp_bonus = LOAD '/home/student/Documents/Pig/emp_bonus.txt' USING
PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);


cogroup_data = COGROUP emp_sales by sno, emp_bonus by sno;

dump cogroup_data;
```

# Output :

```
(1,{(1,Robin,22,25000,sales)},{(1,Robin,22,25000,sales)})
(2,{(2,BOB,23,30000,sales)},{(2,Jaya,23,20000,admin)})
(3,{(3,Maya,23,25000,sales)},{(3,Maya,23,25000,sales)})
(4,{(4,Sara,25,40000,sales)},{(4,Alia,25,50000,admin)})
(5,{(5,David,23,45000,sales)},{(5,David,23,45000,sales)})
(6,{(6,Maggy,22,35000,sales)},{(6,Omar,30,30000,admin)})
```

```
diff_data = FOREACH cogroup_data GENERATE DIFF(emp_sales,emp_bonus);

Dump diff_data;
```

# Output :

```
({})
({(2,BOB,23,30000,sales),(2,Jaya,23,20000,admin)})
({})
({(4,Sara,25,40000,sales),(4,Alia,25,50000,admin)})
({})
({(6,Maggy,22,35000,sales),(6,Omar,30,30000,admin)})
```

# isEmpty Function :~

```
cogroup_data = COGROUP emp_sales by age, emp_bonus by age;
dump cogroup_data;
```

# Output :

```
(23,{(5,David,23,45000,sales),(3,Maya,23,25000,sales),(2,BOB,23,30000,sales)},
{(5,David,23,45000,sales),(3,Maya,23,25000,sales),(2,Jaya,23,20000,admin)})
(25,{(4,Sara,25,40000,sales)},{(4,Alia,25,50000,admin)})
(30,{},{(6,Omar,30,30000,admin)})
```

```
isempty_data = filter cogroup_data by IsEmpty(emp_sales);
Dump isempty_data;
```

# Output :

```
(30,{},{(6,Omar,30,30000,admin)})
```

# Max Function and Min Function :~

```
student_cgpa = LOAD '/home/student/Documents/Pig/student_cgpa.txt' USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray, gpa:int);

student_group_all = Group student_cgpa All;

student_gpa_max = foreach student_group_all Generate (student_details.firstname, student_details.gpa), MAX(student_details.gpa);

student_gpa_min = foreach student_group_all Generate (student_details.firstname, student_details.gpa), MIN(student_details.gpa);



Dump student_gpa_max;
```

# Output :

```
(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Rajiv) } ,{ (72) , (83) , (87) , (75) , (93) , (90) , (78) , (89) }) ,93)
```

```
Dump student_gpa_min;
```

# Output :

```
(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Rajiv) } ,{ (72) , (83) , (87) , (75) , (93) , (90) , (78) , (89) }) ,72)
```

# Size :~

## *Syntax*

```
employee_data = LOAD '/home/student/Documents/Pig/employee.txt' USING PigStorage(',') as (id:int, name:chararray, workdate:chararray, aily_typing_pages:int);


size = FOREACH employee_data GENERATE SIZE(name);
```

Dump size;

# Output :

```
(4)
(3)
(4)
(4)
(4)
(4)
(4)
```

# Subtract Function :~

emp_sales = LOAD '/home/student/Documents/Pig/emp_sales.txt' USING PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);

emp_bonus = LOAD '/home/student/Documents/Pig/emp_bonus.txt' USING PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);

cogroup_data = COGROUP emp_sales by sno, emp_bonus by sno;

Dump cogroup_data;

# Output :

```
(1,{(1,Robin,22,25000,sales)},{(1,Robin,22,25000,sales)})
(2,{(2,BOB,23,30000,sales)},{(2,Jaya,23,20000,admin)})
(3,{(3,Maya,23,25000,sales)},{(3,Maya,23,25000,sales)})
(4,{(4,Sara,25,40000,sales)},{(4,Alia,25,50000,admin)})
(5,{(5,David,23,45000,sales)},{(5,David,23,45000,sales)})
```

```
(6,{(6,Maggy,22,35000,sales)},{(6,Omar,30,30000,admin)})
```

```
sub_data1 = FOREACH cogroup_data GENERATE SUBTRACT(emp_sales, emp_bonus);
sub_data2 = FOREACH cogroup_data GENERATE SUBTRACT(emp_bonus, emp_sales);
```

# Output :

```
Dump sub_data1;
({})
({(2,BOB,23,30000,sales)})
({})
({(4,Sara,25,40000,sales)})
({})
({(6,Maggy,22,35000,sales)})

Dump sub_data2;
({})
({(2,Jaya,23,20000,admin)})
({})
({(4,Alia,25,50000,admin)})
({})
({(6,Omar,30,30000,admin)})
```

# SUM Function :~

```
employee_data = LOAD '/home/student/Documents/Pig/employee.txt' USING
PigStorage(',') as (id:int, name:chararray, workdate:chararray,
daily_typing_pages:int);
```

# Output :

```
(1,John,2007-01-24,250)
(2,Ram,2007-05-27,220)
(3,Jack,2007-05-06,170)
(3,Jack,2007-04-06,100)
(4,Jill,2007-04-06,220)
(5,Zara,2007-06-06,300)
(5,Zara,2007-02-06,350)
```

```
employee_group = Group employee_data ALL;
student_workpages_sum = foreach employee_group Generate
(employee_data.name,employee_data.daily_typing_pages),SUM(employee_data.daily_typing_pages);
Dump student_workpages_sum;
```

# Output :

```
(({(Zara),(Zara),(Jill),(Jack),(Jack),(Ram),(John)},{(350),(300),(220),(100),(170),(220),(250)}),1610)
```

# TextLoader Function :~

```
details = LOAD '/home/student/Documents/Pig/hadoop_logs.txt' USING
TextLoader();
dump details;
```

# Output :

```
(2014-10-20 14:50:08,367 INFO
org.apache.hadoop.hdfs.server.namenode.NNStorageRetentionManager:
Going to retain 2 images with txid >= 0)

(2014-10-20 14:50:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:50:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 0 millisecond(s).)

(2014-10-20 14:50:53,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:50:53,091 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 1 millisecond(s).)

(2014-10-20 14:51:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:51:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 1 millisecond(s).)

(2014-10-20 14:51:53,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:51:53,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 0 millisecond(s).)

(2014-10-20 14:52:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)
```

```
(2014-10-20 14:52:23,091 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 1 millisecond(s).)

(2014-10-20 14:52:53,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:52:53,111 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 21 millisecond(s).)

(2014-10-20 14:53:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:53:23,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 0 millisecond(s).)

(2014-10-20 14:53:53,090 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30001 milliseconds)

(2014-10-20 14:53:53,091 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 0 millisecond(s).)

(2014-10-20 14:54:23,091 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)

(2014-10-20 14:54:23,184 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 94 millisecond(s).)

(2014-10-20 14:54:53,091 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Rescanning after 30000 milliseconds)
```

```
(2014-10-20 14:54:53,142 INFO
org.apache.hadoop.hdfs.server.blockmanagement.CacheReplicationMonitor:
Scanned 0 directive(s) and 0 block(s) in 51 millisecond(s).)

(2014-10-20 20:26:49,988 INFO
org.apache.hadoop.hdfs.server.namenode.NameNode: STARTUP_MSG: )
```

# TOBAG Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/employee_details.txt' USING
PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);
```

# Output :

```
(1,Robin,22,newyork)
(2,BOB,23,Kolkata)
(3,Maya,23,Tokyo)
(4,Sara,25,London)
(5,David,23,Bhuwaneshwar)
(6,Maggy,22,Chennai)
```

```
tobag = FOREACH emp_data GENERATE TOBAG (id,name,age,city);
Dump tobag;
```

# Output :

```
({(1),(Robin),(22),(newyork)})
({(2),(BOB),(23),(Kolkata)})
({(3),(Maya),(23),(Tokyo)})
({(4),(Sara),(25),(London)})
({(5),(David),(23),(Bhuwaneshwar)})
```

```
({(6),(Maggy),(22),(Chennai)})
```

## TOP Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/employee_details.txt' USING PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);

emp_group = Group emp_data BY age;

Dump emp_group;
```

# Output :

```
(22,{(6,Maggy,22,Chennai),(1,Robin,22,newyork)})
(23,{(5,David,23,Bhuwaneshwar),(3,Maya,23,Tokyo),(2,BOB,23,Kolkata)})
(25,{(4,Sara,25,London)})
```

```
data_top = FOREACH emp_group
{top = TOP(2, 0, emp_data);
GENERATE top;}
```

In the above example we are retrieving the top 2 tuples of a group having greater id. Since we are retrieving top 2 tuples based on the id, we are passing the index of the column name "id" as second parameter of TOP() function. (index of id is 0)

#TOTUPLE Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/employee_details.txt' USING PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);

totuple = FOREACH emp_data GENERATE TOTUPLE (id,name,age);

Dump totuple;
```

# Output :

```
((1,Robin,22))
((2,BOB,23))
((3,Maya,23))
((4,Sara,25))
((5,David,23))
((6,Maggy,22))
```

# TOMAP Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/employee_details.txt' USING PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);
tomap = FOREACH emp_data GENERATE TOMAP(name, age);
dump tomap;
```

# Output :

```
([Robin#22])
([BOB#23])
([Maya#23])
([Sara#25])
([David#23])
([Maggy#22])
```

# ENDSWITH and STARTSWITH Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/employee_details.txt' USING PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);
```

```
emp_endswith = FOREACH emp_data GENERATE (id,name),ENDSWITH ( name, 'n' );
dump emp_endswith;
```

# Output :

```
((1,Robin),true)
((2,BOB),false)
((3,Maya),false)
((4,Sara),false)
((5,David),false)
((6,Maggy),false)
```

```
startswith_data = FOREACH emp_data GENERATE (id,name), STARTSWITH (name,’Ro’);
dump startswith_data;
```

# Output :

```
((1,Robin),true)
((2,BOB),false)
((3,Maya),false)
((4,Sara),false)
((5,David),false)
((6,Maggy),false)
```

# SUBSTRING Function :~

```
substring_data = FOREACH emp_data GENERATE (id,name), SUBSTRING (name, 0, 2);

Dump substring_data;
```

# Output :

```
((1,Robin),Ro)

((2,BOB),BO)

((3,Maya),Ma)

((4,Sara),Sa)

((5,David),Da)

((6,Maggy),Ma)
```

```
substring_data = FOREACH emp_data GENERATE (id,name), SUBSTRING (name, 0, 3);

Dump substring_data;
```

# Output :

```
((1,Robin),Rob)

((2,BOB),BOB)

((3,Maya),May)

((4,Sara),Sar)

((5,David),Dav)

((6,Maggy),Mag)
```

# EqualsIgnoreCase Function :~

```
equals_data = FOREACH emp_data GENERATE (id,name), EqualsIgnoreCase(name, 'Robin');
```

```
Dump equals_data;
```

# Output :

```
((1,Robin),true)
((2,BOB),false)
((3,Maya),false)
((4,Sara),false)
((5,David),false)
((6,Maggy),false)
```

# IndexOf Function :~

```
indexof_data = FOREACH emp_data GENERATE (id,name),
INDEXOF(name,'r',0);
Dump indexof_data;
```

# Output :

```
((1,Robin),-1)
((2,BOB),-1)
((3,Maya),-1)
((4,Sara),2)
((5,David),-1)
((6,Maggy),-1)
```

The above statement parses the name of each employee and returns the index value at which the letter ‘r’ occurred for the first time. If the name doesn’t contain the letter ‘r’ it returns the value -1

# Last_Index_of :~

```
last_index_data = FOREACH emp_data GENERATE (id,name),
LAST_INDEX_OF(name, 'g');
```

```
Dump last_index_data;
```

# Output :

```
((1,Robin),-1)
((2,BOB),-1)
((3,Maya),-1)
((4,Sara),-1)
((5,David),-1)
((6,Maggy),3)
```

The above statement parses the name of each employee from the end and returns the index value at which the letter ‘g’ occurred for the first time. If the name doesn’t contain the letter ‘g’ it returns the value −1

# LCFIRST, UCFIRST, LOWER, UPPER :~

```
Lcfirst_data = FOREACH emp_data GENERATE (id,name), LCFIRST(name);
ucfirst_data = FOREACH emp_data GENERATE (id,city), UCFIRST(city);
upper_data = FOREACH emp_data GENERATE (id,name), UPPER(name);
lower_data = FOREACH emp_data GENERATE (id,name), LOWER(name);
Dump Lcfirst_data;
```

# Output :

```
((1,Robin),robin)
((2,BOB),bOB)
```

```
((3,Maya),maya)
((4,Sara),sara)
((5,David),david)
((6,Maggy),maggy)
```

```
Dump ucfirst_data;
# Output :
((1,newyork),Newyork)
((2,Kolkata),Kolkata)
((3,Tokyo),Tokyo)
((4,London),London)
((5,Bhuwaneshwar),Bhuwaneshwar)
((6,Chennai),Chennai)
```

```
Dump lower_data;
# Output :
((1,Robin),robin)
((2,BOB),bob)
((3,Maya),maya)
((4,Sara),sara)
((5,David),david)
((6,Maggy),maggy)
```

```
Dump upper_data;
# Output :
```

```
((1,Robin),ROBIN)
((2,BOB),BOB)
((3,Maya),MAYA)
((4,Sara),SARA)
((5,David),DAVID)
((6,Maggy),MAGGY)
```

# Replace Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/employee_details.txt' USING
PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);

replace_data = FOREACH emp_data GENERATE
(id,city),REPLACE(city,'Bhuwaneshwar','Bhuw');

Dump replace_data;
```

# Output :

```
((1,newyork),newyork)
((2,Kolkata),Kolkata)
((3,Tokyo),Tokyo)
((4,London),London)
((5,Bhuwaneshwar),Bhuw)
((6,Chennai),Chennai)
```

# STRSPLIT Function :~

```
emp_data = LOAD '/home/student/Documents/Pig/emp_split.txt USING
PigStorage(',') as (id:int, name:chararray, age:int, city:chararray);

strsplit_data = FOREACH emp_data GENERATE (id,name), STRSPLIT
(name,'_',2);
```

# Output :

```
((1,Robin),(Robin))
((2,BOB),(BOB))
((3,Maya),(Maya))
((4,Sara),(Sara))
((5,David),(David))
((6,Maggy),(Maggy))
```

# Date Functions :~

```
date_data = LOAD '/home/student/Documents/Pig/date.txt' USING PigStorage(',')
as (id:int,date:chararray);
todate_data = foreach date_data generate ToDate(date,'yyyy/MM/dd HH:mm:ss')
as (date_time:DateTime >);
Dump todate_data;

currenttime_data = foreach todate_data generate CurrentTime();
Dump currenttime_data;
todate_data = foreach date_data generate ToDate(date,'yyyy/MM/dd HH:mm:ss')
as (date_time:DateTime );
Dump todate_data;

getday_data = foreach todate_data generate(date_time), GetDay(date_time);
Dump getday_data;
todate_data = foreach date_data generate ToDate(date,'yyyy/MM/dd HH:mm:ss')
as (date_time:DateTime );
```

```
gethour_data = foreach todate_data generate (date_time), GetHour(date_time);
```

# Mathematical Functions :~

ABS Function :

```
math_data = LOAD '/home/student/Documents/Pig/math.txt' USING PigStorage(',')
as (data:float);
```

# Output :

```
(5.0)
(16.0)
(9.0)
(2.5)
(5.9)
(3.1)
```

```
abs_data = foreach math_data generate (data), ABS(data);
Dump abs_data;
```

# Output :

```
(5.0,5.0)
(16.0,16.0)
(9.0,9.0)
(2.5,2.5)
(5.9,5.9)
(3.1,3.1)
```

# Cube and Square Root Function :~

```
cbrt_data = foreach math_data generate (data), CBRT(data);
```

# Output :

```
(5.0,1.709975946676697)
(16.0,2.5198420997897464)
(9.0,2.080083823051904)
(2.5,1.3572088082974532)
(5.9,1.8069688790571206)
(3.1,1.4580997208745365)
```

```
sqrt_data = foreach math_data generate (data), SQRT(data);
```

# Output :

```
(5.0,2.23606797749979)
(16.0,4.0)
(9.0,3.0)
(2.5,1.5811388300841898)
(5.9,2.4289915799292987)
(3.1,1.76068165908337)
```

# Trigometric Functions :~

```
acos_data = foreach math_data generate (data), ACOS(data);
```

# Output :

```
(5.0,NaN)
```

```
(16.0,NaN)
(9.0,NaN)
(2.5,NaN)
(5.9,NaN)
(3.1,NaN)
```

```
asin_data = foreach math_data generate (data), ASIN(data);
```

# Output :

```
(5.0,NaN)
(16.0,NaN)
(9.0,NaN)
(2.5,NaN)
(5.9,NaN)
(3.1,NaN)
```

```
atan_data = foreach math_data generate (data), ATAN(data);
```

# Output :

```
(5.0,1.373400766945016)
(16.0,1.5083775167989393)
(9.0,1.460139105621001)
(2.5,1.1902899496825317)
(5.9,1.4029004062076729)
(3.1,1.2587541962439153)
```

```
cos_data = foreach math_data generate (data), COS(data);
```

```
# Output :
(5.0,0.28366218546322625)
(16.0,-0.9576594803233847)
(9.0,-0.9111302618846769)
(2.5,-0.8011436155469337)
(5.9,0.9274784663996888)
(3.1,-0.9991351463078334)
```

```
cosh_data = foreach math_data generate (data), COSH(data);
```

```
# Output :
(5.0,74.20994852478785)
(16.0,4443055.260253992)
(9.0,4051.5420254925943)
(2.5,6.132289479663686)
(5.9,182.52012106128686)
(3.1,11.121499185584959)
```

```
sin_data = foreach math_data generate (data), SIN(data);
```

```
# Output :
(5.0,-0.9589242746631385)
(16.0,-0.2879033166650653)
(9.0,0.4121184852417566)
(2.5,0.5984721441039564)
```

```
(5.9,-0.3738765763789988)
(3.1,0.04158075771824354)
```

```
sinh_data = foreach math_data generate (data), SINH(data);
```

```
# Output :
(5.0,74.20321057778875)
(16.0,4443055.26025388)
(9.0,4051.54190208279)
(2.5,6.0502044810397875)
(5.9,182.51738161672935)
(3.1,11.076449978895173)
```

```
tan_data = foreach math_data generate (data), TAN(data);
```

```
# Output :
(5.0,-3.380515006246586)
(16.0,0.3006322420239034)
(9.0,-0.45231565944180985)
(2.5,-0.7470222972386603)
(5.9,-0.4031107890087444)
(3.1,-0.041616750118239246)
```

```
tanh_data = foreach math_data generate (data), TANH(data);
```

```
# Output :
```

```
(5.0,0.9999092042625951)
(16.0,0.9999999999999747)
(9.0,0.999999969540041)
(2.5,0.9866142981514303)
(5.9,0.9999849909996685)
(3.1,0.9959493584508665)
```

```
ceil_data = foreach math_data generate (data), CEIL(data);
```

# Output :

```
(5.0,5.0)
(16.0,16.0)
(9.0,9.0)
(2.5,3.0)
(5.9,6.0)
(3.1,4.0)
```

```
floor_data = foreach math_data generate (data), FLOOR(data);
```

# Output :

```
(5.0,5.0)
(16.0,16.0)
(9.0,9.0)
(2.5,2.0)
(5.9,5.0)
```

```
(3.1,3.0)

round_data = foreach math_data generate (data), ROUND(data);

# Output :
(5.0,5)
(16.0,16)
(9.0,9)
(2.5,3)
(5.9,6)
(3.1,3)
```

```
# Logarithmic Functions :~

log_data = foreach math_data generate (data),LOG(data);

# Output :

(5.0,1.6094379124341003)
(16.0,2.772588722239781)
(9.0,2.1972245773362196)
(2.5,0.9162907318741551)
(5.9,1.774952367075645)
(3.1,1.1314020807274126)
```

```
log_data1 = foreach math_data generate (data),LOG10(data);
```

```
# Output :
```

```
(5.0,0.6989700043360189)
(16.0,1.2041199826559248)
(9.0,0.9542425094393249)
(2.5,0.3979400086720376)
(5.9,0.7708520186620678)
(3.1,0.4913616804737727)
```